CIRRELT
Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport
Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Biomedical Sample Transportation: A Case Study Based on Québec's Healthcare Supply Chain

Ana Maria Anaya-Arenas
Thomas Chabot
Jacques Renaud
Angel Ruiz

October 2014

CIRRELT-2014-51

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2014-009.

Biomedical Sample Transportation: A Case Study Based on Québec's Healthcare Supply Chain

Ana Maria Anaya-Arenas, Thomas Chabot, Jacques Renaud*, Angel Ruiz

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

Abstract. Biomedical sample management plays a central role in an efficient healthcare system and requires important resources. This article describes and models the challenging context of biomedical sample transportation in the Canadian province of Quebec as a variant of the multi-trip vehicle routing problem with time windows. In this case, several routes need to be planned from a laboratory to satisfy the multiple pick-up requests of each sample collection center (SCC) under some practical constraints. We propose two alternative mathematical formulations as well as fast heuristics to minimize total transportation distances. The performance of the proposed methods is assessed over a large case study based on the network of laboratories in the province of Quebec. Results helped Quebec's Ministry of Health and Social Services to increase its service quality and to reduce its transportation costs.

Keywords. Biomedical sample transportation; vehicle routing; multi-trips; time windows

Acknowledgements. This research was partially supported by Grants OPG 0293307 and OPG 0172633 from the Natural Sciences and Engineering Research Council of Canada (NSERC) this support is gratefully acknowledged. We also express our gratitude to Dominic Morand, « conseiller à la direction de la logistique et des équipements », of the Ministère de la Santé et des Services sociaux (MSSS) for his collaboration in this project.

---

* Corresponding author: Jacques.Renaud@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1. Introduction

Biomedical tests are of great importance in helping physicians make accurate diagnoses. To accomplish such tests, thousands of biomedical samples or "specimens" need to be transported daily from different facilities of the healthcare network where they are collected (hospitals, private clinics or other *samples collection centers,* SCCs) to laboratories where they are analyzed. Although large hospitals often own on-site laboratories where collected specimens can be analyzed, most of the SCCs are not equipped to analyze the samples they collect, so they must ensure adequate transportation of samples to laboratories.

This research focuses on the organization of these transportation activities and is based on the real situation observed in Quebec, where the Ministry of Health and Social Services (*Ministère de la santé et des services sociaux* – MSSS) has the role of supporting and overseeing the health network in the province in order to ensure the well-being of Quebecers. Aiming at improving the services offered to the population, a special focus has recently been put on the optimization of the existing laboratories and, in particular, on the transportation aspects, giving birth to a supply chain optimization project named *Optilab.* Optilab seeks to enhance the quality of the services given by the network of laboratories in terms of security, accessibility, efficiency and efficacy (MSSS, 2012). Until now, the transportation of biomedical specimens in Quebec has been done in a rather decentralized manner, each laboratory and SCC deciding independently the transportation services needed and used. There is no formal structure ruling the specimens' transportation on this network, nor are there directives. Also, the MSSS would apply closer control to transportation costs and take advantage of potential economies of scale due to high transportation volume generated by samples' transportation. In this context, this research is a first step to model, quantify and optimize the daily transportation operations of biomedical samples.

This paper presents two major contributions. First, it defines the *biomedical sample transportation problem* (BSTP) and proposes two alternative formulations. Second, it proposes a solving approach able to cope with real-size daily instances. This article is structured as follows. Section 2 describes the practical context of the biomedical sample transportation problem. Section 3 reviews related literature. Section 4 presents two mathematical formulations to optimize the BSTP and Section 5 introduces several heuristics to solve it. Real data and computational results are presented in Section 6. Section 7 closes this article and offers some research perspectives.

2. Problem statement

The network considered here is composed of *sample collection centers* (SCCs) and *laboratories* (Labs). SCCs are health facilities that patients visit in order to produce the requested biomedical samples. Then, the specimens are pre-treated and prepared to be sent to a Lab for analysis. Following the structure imposed by the MSSS, the Quebec province is divided into territories, and each SCC in a territory is affected *a priori* to a given Lab. Hence, each transportation network is composed of a set of SCCs (clients) affected to a Lab (that we identify as a depot). Finally, even if hundreds of different specimens are collected, they are all grouped and transported within standard refrigerated sample boxes. As a box may contain up to 80 samples, an SCC rarely requests transportation for more than one or two boxes at a time. Due to the small size of the sample boxes, vehicle capacity is irrelevant. Therefore, we only need to consider the *transportation requests* of each SCC. These transportation requests are performed by public carriers which are under contract with the SCCs and paid on the basis of the traveled distance. Each driver may perform multiple routes during the working day. When visiting an SCC, the driver exchanges empty boxes for new ones. No constraint on the number of vehicles is considered because the carrier will adjust the number of drivers to satisfy regular planned transportation requests as well as emergency calls. Thus, drivers' routes must begin at the Lab, where they will get empty boxes to exchange with full boxes at every visited SCC. The driver has to return to the Lab to deliver the samples before starting any other route. Finally, drivers' schedule must respect the maximum number of driving hours per day fixed by the Quebec province regulation.

The biomedical sample transportation from SCCs to the Lab has some characteristics that make it a challenging optimization problem. The major constraints are the samples' maximum transportation time and the multiple transportation requests at each SCC. Maximal transportation time is due to the samples' lifespan. After collection, each sample box must arrive at the Lab within a given time frame to be treated. Otherwise, samples deteriorate and become unusable, increasing tremendously both the Lab's costs and the quality of the service. In fact, an unusable sample forces the patient to make a second collection; it delays the analysis and doubles the operations costs of the entire process (collecting, transporting and analyzing). In order to respect sample lifespans, SCC cannot keep the collected specimens for a long time; this is why each SCC may make a different number of sample transportation requests, depending on its daily opening hours. For example, if an SCC is open from 9h 00 to 13h 00, it cannot hold samples during the entire morning and then make a single transportation request at the end of the day. Instead, it will require at least two different pick-up appointments. Moreover, these two appointments cannot be too close to one another (e.g., it would not be useful to make a visit at 12h 00 and another one at 13h 00) because this can be as bad and as risky as just making one pick-up. This is why SCCs propose strict time windows for each transportation request; for example, the first pick-up must be done between 10h 45 and 11h 15 and the last one between 13h 00 and 13h 30. These multiple pick-ups are also needed to avoid problems with sample storage capacity at the SCC and provide the Lab with a more continuous supply of sample boxes and thus help to smooth the workload.

Therefore, the BSTP aims at finding the minimum distance set of routes in order to satisfy all the transportation requests of each SCC while respecting the imposed time windows, the maximum transportation time of any sample and the vehicles' maximum working hours.

3. Related literature

The BSTP is characterized by multiple visits to each customer, a time window on each visit time, multiple routes (trips) for each vehicle and other practical constraints. Therefore, it is clearly grounded in the vehicle routing literature and particularly related to the vehicle routing problem with time windows (VRPTW) and the multi-trip vehicle routing problem (MTVRP). Because the literature on these subjects is vast, the next paragraphs aim solely at outlining some key reviews or the most recent works on these topics.

Concerning VRPTW, the wide review of Cordeau et al. (2002) and those by Bräysy and Gendreau focused on local search algorithms (2005a) and metaheuristics (2005b) are key contributions to the field. More recently, Kallehauge (2008) and Baldacci et al. (2011; 2012) proposed exact algorithms that are among the best available for solving the classical capacitated VRPTW.

Taillard et al. (1996) and Brandão and Mercer (1997; 1998) were among the first to deal with the multi-trip vehicle routing problem (MTVRP). They proposed tabu search algorithms for solving an MTVRP with different constraint types, including time windows. Olivera and Viera (2007) used an adaptive memory programming algorithm to solve the classical MTVRP, leading to better results than those produced by Brandão and Mercer (1998) and Taillard et al. (1996). Prins (2002) considers a MTVRP with two objectives (the total distance and the number of required vehicles) and reported results from a French manufacturer of furniture. Mingozzi et al. (2013) proposed an exact algorithm and solved benchmark instances with up to 120 customers. Battarra et al. (2009) solved an MTVRP-TW with multiple incompatible commodities with the objective of minimizing the number of vehicles. Cattaruzza et al. (2014) proposed an iterated local search algorithm for MTVRP-TW. Azi et al. (2010) addressed a variant of the MTVRP where a time window and revenue are associated with each customer. Martínez and Amaya (2013) used a tabu search algorithm for solving an MTVRP-TW with loading constraints. Azi et al. (2014) developed an adaptive large neighborhood search algorithm for the MTVRP where the objective is first to maximize the number of served customers and then to minimize the total distance traveled by the vehicles. Wang et al. (2014) proposed a metaheuristic based on a pool of routes to solve the MTVRP-TW.

Three particular works seem to be the closest to the BSTP. Hernández et al. (2011) proposed a MTVRP-TW where a limit on the total duration of the routes is imposed. They developed a two-phase exact algorithm to solve the problem and tested it on Solomon’s benchmark instances. Liu et al. (2013) studied a routing problem where biomedical samples need to be collected and delivered to laboratories. In their case, four types of deliveries and pick-up requirements were considered. As in the BSTP, visits must respect time windows, but in their case each node requires only one visit and each vehicle performs a single route. Finally, Doerner et al. (2008) dealt with a blood collection problem and proposed several variants of a construction heuristic and a branch-and-bound algorithm. As in the BSTP, transportation time is limited to preserve the blood’s quality, and multiple pick-ups can be planned at each customer location. However, in their case, time windows concern the opening hours of SCCs, instead of the transportation requests. Also, the number of pick-ups is not fixed but is one of the decisions.

4. Mathematical formulations

This section proposes two formulations for the BSTP. The first one deals explicitly with the multiple transportation requests at each SCC, while the second one duplicates the SCC, such as each node has only one transportation request.

4.1. Model 1: Multiple transportation requests per SCC (BSTP-MP)

The BSTP is modeled over a complete graph $G = \{V, A\}$, where $V = \{v_0, v_1, \dots, v_n, v_{n+1}\}$ is the set of nodes in the network, composed by the n SCCs (set $N = \{v_1, v_2, \dots, v_n\}$) that generates transportation requests and the Lab $\{v_0, v_{n+1}\}$ where every route must start and end. We define the arc set $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j, i = 0, \dots, n, j = 1, \dots n+1\}$, and to each arc (v_i, v_j) are associated a travel time (t_{ij}) and a travel distance (d_{ij}). K uncapacited vehicles are available for satisfying SCCs transportation requests. Each vehicle can perform multiple routes ($r = 1, \dots R$) inside a work shift, but it has to respect a limit on the length of its working day (T_k).

Each SCC i requires a specific number of transportation requests ($q = 1, \dots, Q_i$). For SCC i, its q^{th} pick-up has to be done inside a time window $[a_{iq}, b_{iq}]$, where a_{iq} is the earliest time the transport may arrive (otherwise, he has to wait) to perform the pick-up q of SCC i, and b_{iq} is the latest accepted arrival time. Time windows are considered to be hard constraints. In addition, we need to consider a loading time (τ_i) at each SCC and an unloading time (τ_0) of the vehicle at the Lab before a new route can be started. Furthermore, let T_{max} be the maximal transportation time for any sample.

In order to define a transportation plan that respects the practical constraints of our problem and minimize transportation costs, we define the following decision variables.

x_{ijkr} binary variable equal to 1 if the arc (i, j) is used by vehicle k in its route r; 0 otherwise.

y_{jqkr} binary variable equal to 1 if the q^{th} request of SCC j is done by vehicle k in its route r; 0 otherwise.

u_{ikr} continuous variable that indicates the visit time of SCC i by vehicle k in route r.

The model BSTP-MR is stated as follows:

$$Min \sum_{i=0}^{n} \sum_{j=1}^{n+1} \sum_{k=1}^{K} \sum_{r=1}^{R} d_{ij} x_{ijkr} \tag{1.1}$$

Subject to:

$$\sum_{i=0}^{n} x_{ijkr} \leq 1 \qquad j = 1, \dots, n; k = 1, \dots, K; r = 1, \dots, R \tag{1.2}$$

$$\sum_{i=0}^{n} x_{ijkr} - \sum_{l=1}^{n+1} x_{jlkr} = 0 \qquad j = 1, \dots, n; k = 1, \dots, K; r = 1, \dots, R \tag{1.3}$$

$$\sum_{j=1}^{n} x_{0jkr} \leq 1 \qquad k = 1, \dots, K; r = 1, \dots, R \tag{1.4}$$

$$\sum_{j=1}^{n} x_{0jkr} - \sum_{j=1}^{n} x_{j,n+1kr} = 0 \qquad k = 1, \dots, K\,; r = 1, \dots, R \qquad (1.5)$$

$$\sum_{j=1}^{n} x_{0jkr} - \sum_{j=1}^{n} x_{0jkr-1} \leq 0 \qquad k = 1, \dots, K\,; r = 2, \dots, R \qquad (1.6)$$

$$\sum_{k=1}^{K} \sum_{r=1}^{R} y_{jqkr} = 1 \qquad j = 1, \dots, n;\; q = 1, \dots Q_j \qquad (1.7)$$

$$\sum_{q=1}^{Q_j} y_{jqkr} - \sum_{i=0}^{n} x_{ijkr} = 0 \qquad j = 1, \dots, n\; k = 1, \dots, K\,; r = 1, \dots, R \qquad (1.8)$$

$$u_{ikr} + \tau_i + t_{ij} - u_{jkr} \leq T_k\left(1 - x_{ijkr}\right) \qquad i = 0, \dots, n; j = 1, \dots n+1; \; k = 1, \dots, K\,; r = 1, \dots, R \qquad (1.9)$$

$$a_{jq} - T_k\left(1 - y_{jpkr}\right) \leq u_{jkr} \leq b_{jq} + T_k\left(1 - y_{jpkr}\right) \qquad j = 1, \dots, n; k = 1, \dots, K \; r = 1, \dots, R\,; q = 1, \dots, Q_j \qquad (1.10)$$

$$u_{0kr} \geq u_{n+1,k,r-1} \qquad k = 1, \dots, K; r = 2, \dots, R \qquad (1.11)$$

$$u_{n+1kr} - u_{jkr} \leq T_{max} + \left(1 - \sum_{i=0}^{n} x_{ijkr}\right)T_k \qquad j = 1, \dots, n, k = 1, \dots, K\,; r = 1, \dots, R \qquad (1.12)$$

$$u_{n+1,kr} - u_{0,k1} \leq T_k \qquad k = 1, \dots, K\,; r = 1, \dots, R \qquad (1.13)$$

Objective (1.1) is to find a transportation plan that minimizes the total traveled distance. Constraints (1.2) assure that an SCC j is visited at most one time by route r of vehicle k. Constraints (1.3) force the flow of each vehicle k for each of its routes r to be balanced at each SCC of the network. This means that if an arc enters to node j in a route r of vehicle k, there must be an arc that leaves the same node for the same (k, r) combination. Constraints (1.4) state that a truck k can start a route r or not, but (1.5) if the vehicle starts a route, it must come back to depot (node $n + 1$). Constraints (1.6) make an order on the routes; thus, route r is started if and only if a route $r - 1$ has been already created. Constraints (1.7) and (1.8) verify the pick-up request satisfaction. Constraints (1.7) state that each pick-up q for each SCC j is done by one and only one vehicle route (k, r) combination; constraints (1.8) link the arc to the pick-up variables, saying that if a pick-up is done by the route r of the truck k is because there is an arc of this combination that entered to the node j. Constraints (1.9) to (1.13) handle the time constraints. Constraints (1.9) have two main purposes: First, it estimates the arrival time at every node (clients or dummy depot), and second, it forces the sub-tours' elimination. Then, constraint (1.10) fixes the upper and lower bound of the time windows, forcing that if the pick-up q of client j is done with vehicle k in its route r ($y_{jqkr} = 1$), it must be inside the time window of the pick-up request; otherwise, the constraints are irrelevant (when $y_{jqkr} = 0$). Here T_k is used as a Big M value. Constraints (1.11) ensure that the starting time of route r is later than the arriving time of route $r - 1$ at node (n+1). Constraints (1.12) impose the maximum transportation length time limit (returning time to the Lab minus the pick-up time at any j is less than the limit if j is visited by vehicle k on its route r). Constraints (1.13) force respect for the total work shift length for a vehicle k.

4.2. *Model 2: Extended graph (BSTP-EG)*

In the BSTP-EG, each transportation request is represented by a specific node, so if SCC i requires Q_i pick-ups, i is represented by Q_i nodes located at the same place, each needing one request. Therefore, the original set N is extended into a set P of p nodes ($p = \sum_{i=1}^{n} Q_i$).

We define a complete graph $G_2 = \{V_2, A\}$, where $V_2 = \{v_0, v_1, \dots, v_p, v_{p+1}\}$ is the set of nodes in the network, which includes the laboratory as nodes $\{v_0, v_{p+1}\}$ and the set $P = \{v_1, v_2, \dots, v_p\}$, with the p transportation requests of the SCCs. We define also P_n as the set of nodes representing the pick-ups demanded by the original SCC n. Therefore, node set P is composed of a set of pick-ups originating from different SCCs (i.e., $P = \bigcup_n P_n$). Finally, we consider the arc set $A = \{(v_i, v_j): v_i, v_j \in V_2, i \neq j, i = 0, \dots, p, j = 1, \dots p+1\}$. Clearly, t_{ij} and d_{ij} are equal to zero for every (v_i, v_j) if i and $j \in P_n$, i.e., nodes i and j correspond to two requests from the same SCC. In addition, each request needs to be performed inside its original time window $[a_j, b_j]$. Finally, no more than one node from each P_n can be visited on any route. The rest of the notation of model BSTP-MP is also valid for model BSTP-EG. The following decisions variables are used:

x_{ijkr} binary variable equal to 1 if the arc (i,j) is used by vehicle k in its route r; 0 otherwise.
u_{ikr} continuous variable that indicates the visit time of pick-up i by vehicle k in route r.

$$Min \sum_{i=0}^{p} \sum_{j=1}^{p+1} \sum_{k=1}^{K} \sum_{r=1}^{R} d_{ij} x_{ijkr} \tag{2.1}$$

Subject to:

$$\sum_{k=1}^{K} \sum_{r=1}^{R} \sum_{i=0}^{p} x_{ijkr} = 1 \qquad j = 1, \dots, p \tag{2.2}$$

$$\sum_{j \in P_n} \sum_{i=0}^{p} x_{ijkr} \leq 1 \qquad n = 1, \dots, N; k = 1, \dots, K; r = 1, \dots, R \tag{2.3}$$

$$\sum_{i=0}^{p} x_{ijkr} - \sum_{l=1}^{p+1} x_{jlkr} = 0 \qquad j = 1, \dots, p; k = 1, \dots, K; r = 1, \dots, R \tag{2.4}$$

$$\sum_{j=1}^{p} x_{0jkr} \leq 1 \qquad k = 1, \dots, K; r = 1, \dots, R \tag{2.5}$$

$$\sum_{j=1}^{p} x_{0jkr} - \sum_{j=1}^{p} x_{j,p+1kr} = 0 \qquad k = 1, \dots, K; r = 1, \dots, R \tag{2.6}$$

$$\sum_{j=1}^{p} x_{0jkr} - \sum_{j=1}^{p} x_{0jkr-1} \leq 0 \qquad k = 1, \dots, K; r = 2, \dots, R \tag{2.7}$$

$$u_{ikr} + \tau_i + t_{ij} - u_{jkr} \leq T_k(1 - x_{ijkr}) \qquad i = 0, \dots, p; j = 1, \dots p+1; \; k = 1, \dots, K; r = 1, \dots, R \tag{2.8}$$

$$a_j - T_k\left(1 - \sum_{i=0}^{p} x_{ijkr}\right) \leq u_{jkr} \leq b_j + T_k\left(1 - \sum_{i=0}^{p} x_{ijkr}\right) \qquad j = 1, \dots, p; k = 1, \dots, K; r = 1, \dots, R \tag{2.9}$$

$$u_{0kr} \geq u_{p+1,k,r-1} \qquad k = 1, \dots, K; r = 2, \dots, R \tag{2.10}$$

$$u_{p+1kr} - u_{jkr} \leq T_{max} + \left(1 - \sum_{i=0}^{p} x_{ijkr}\right) T_k \qquad j = 1, \dots, p, k = 1, \dots, K; r = 1, \dots, R \tag{2.11}$$

$$u_{p+1,kr} - u_{0,k1} \leq T_k \qquad k = 1, \dots, K; r = 1, \dots, R \tag{2.12}$$

Objective (2.1) minimizes the traveled distance. Constraints (2.2) ensure that every pick-up p (every node of P) is performed by a vehicle route (r, k). Constraints (2.3) assure that a truck k in its route r visits at the most one node of the original SCC n. Constraints (2.4) force the flow of each truck k for each of its routes r to be balanced for each node j of the network. Constraints (2.5) state that a truck k can start a route r or not, but (2.6) if the vehicle starts a route, it must come back to the depot (node $p+1$). Constraints (2.7) order the routes; thus, route r is started if and only if a route $r-1$ has been already done. Constraints (2.8) to (2.12) handle the time constraints. Their explanation is similar to constraints (1.9) to (1.13).

As will be discussed in Section 6, the two formulations presented here are very difficult to solve. Aiming at improving their solvability, we added the following sub-tour elimination constraints to model BSTP-MR:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijkr} \leq |S|-1 \qquad \begin{array}{c}\forall\, S \subseteq N\backslash\{0\};\ |S| = \{2,3\} \\ k=1,\dots,K\,;\, r=1,\dots,R\end{array} \qquad (1.14)$$

Similarly, we added the following constraints to BSTP-EG:

$$\sum_{i=1}^{p}\sum_{j=1}^{p} x_{ijkr} \leq |S|-1 \qquad \begin{array}{c}\forall\, S \subseteq P\backslash\{0\};\ |S| = \{2,3\} \\ k=1,\dots,K\,;\, r=1,\dots,R\end{array} \qquad (2.13)$$

These constraints eliminate all infeasible tours of two or three nodes not connected to the depot. However, after executing extensive numerical experiments, we were not able to identify a positive impact of those constraints. Therefore, in order to provide a daily transportation plan for the MSSS, an alternative, faster approach to solving the BSTP needed to be developed. The next section presents some simple but efficient heuristics for solving the BSTP.

5. Solving approach

This section presents a two-stage heuristic for solving the BSTP. In the first stage, a pool of feasible solutions is generated using a construction procedure. Then, a local improvement procedure is applied to all the generated solutions in the second stage. Two alternative construction procedures (H1 and H2) are presented before describing the improvement stage.

5.1. Route first and then schedule (H1)

The first construction heuristic is an iterative procedure composed of two steps that are executed sequentially until all the requests are assigned to routes. We note U_{jr} as the arrival time at node j (in minutes) with route r. We also define D_r and F_r as the departure and finishing time of route r, respectively. Transportation requests are sorted in increasing order of their earliest time window (a_j) in the ordered set TR. Since each transportation request j is associated to a specific SCC, *request* and *node* are used indistinctly.

Step 1 Routes construction

Let $r = 1$ the first route.

<u>*Step 1.1 Route initialization*</u>*:* Select a transportation request $i \in TR$. Start route r with request i. The departure time of this route is fixed so that the vehicle will arrive at i at the beginning of its time window (a_i), i.e., $D_r = a_i - t_{0i} - \tau_0$. The visit time of node i is fixed ($U_{ir} = a_i$). Delete i from TR.

<u>*Step 1.2 Adding visits to the current route:*</u> Consider the next transportation request $j \in TR$ and verify the next three conditions:

1. SCC demanding request *j* has not been visited in the current route r.
2. It is possible to arrive at *j* before the end of its time window ($U_{ir} + \tau_i + t_{ij} \leq b_j$).
3. After visiting j at time $U_{jr} = max\{U_{ir} + \tau_i + t_{ij}; a_j\}$, it is possible to return to the Lab, respecting the maximal travel time T_{max} of all the requests in the route.

If all three are satisfied, *j* is added to route r, U_{jr} is fixed as the earliest possible service time to point j ($U_{jr} = max\{U_{ir} + \tau_i + t_{ij}; a_j\}$), and j is erased from *TR*. Node j becomes the current position in route *r*, and the next potential visit is evaluated. When none of the transportation requests in *TR* is eligible, the route is closed, and the vehicle goes back to the Lab. Then, we can calculate F_r as the finish time of route r ($F_r = U_{jr} + \tau_j + t_{j0}$). If *TR* is not empty, go to Step 1.1 to create route $r = r + 1$; otherwise, the algorithm goes to Step 2.

Step 2 Trucks' assignment

Let *R* be the set of feasible routes sorted in ascending order of their departure time D_r. The routes are assigned to vehicles in order to construct the carriers' schedule. This is done by assigning a subset of routes to a specific truck *k*. Truck k departure and finishing times are U_k^d and U_k^f, respectively. To initialize this phase, we set $k = 1$. The first route $r \in R$ is selected, and we set $U_k^d = D_r$ and $U_k^f = F_r$. Route *r* is deleted from *R*. Then we elect the next route $r' \in R$ and evaluate the two following conditions:

1. The departure time of route r', $D_{r'}$, is later (greater) than U_k^f.
2. The schedule of vehicle k, $F_{r'} - U_k^d$, respects the daily work shift limit T_k.

If the two conditions are respected, the route r' is added to the schedule of vehicle k, we set $U_k^d = D_{r'}$ and $U_k^f = F_{r'}$ and we erase r' from the list R. Otherwise, the next route is considered. The process is repeated until no route in R can be assigned to vehicle k, in which case k's schedule is finished. If there are still routes that are not schedule to any vehicle, a new vehicle $k = k + 1$ is created, and Step 2 is repeated until all routes are assigned.

5.2 Schedule construction (H2)

This heuristic produces the schedule of the vehicles directly. This means that vehicles are activated one at a time, and the transportation requests are assigned to them one by one in order to compose its routes. Let $k = 1$ the first vehicle.

Step 1 Vehicle initialization

Select a transportation request $i \in TR$. The vehicle k is started with request i and *TR* is updated. U_k^d is set in such a way that the vehicle collects empty boxes at the Lab and leaves in time to be at the beginning of the time window i. The arrival time to node i is set ($U_{ik} = a_i$).

Step 2 Schedule construction

Let i be the last node visited in the current route. We define a subset J' of "feasible" destinations that could be visited from *i*. A feasible node satisfies all three of the conditions in Step 1.2 of H1 and allows the vehicle to visit node *j* and to go back to the Lab without exceeding the daily shift duration T_k. Then, a destination $j \in J'$ is selected according to a selection criterion (either the closest request to *i*, or the one having the earliest time window) and is added to the vehicle. U_{jk} is updated, j is erased from TR, and this step is re-executed. If none of the requests in TR can be added to J', the vehicle returns to Lab and U_k^f is set. Then, a new route for vehicle k is initiated. The first visit j in this new route will be the first request in TR satisfying the next two conditions:

1. k is able to arrive to j before the end of the node's time window (b_j).
2. k is able to go to j and return to the Lab before the end of the drivers' shift.

If the two conditions are assured, j is added, *TR* is updated, and Step 2 is re-executed. Otherwise, a new truck $k = k + 1$ is activated, and the heuristic goes to Step 1. The procedure is repeated until TR is empty.

Multi-start versions of H1 and H2

H1 and H2 are deterministic procedures. In order to generate a set of feasible solutions, they are executed several times, but each time, we force them to choose a different request from *TR* during the initialization process.

Notice that all the schedules produced by H1 and H2 visit the first node in the first route at the beginning of its time window. However, departing as early as possible might force a premature return to the depot to satisfy constraint T_{max}. Figure 1 illustrates a route $\{Lab - i - j - v - Lab\}$ but considers two different departure times. The upper route a) arrives to i at 8h 00 (the beginning of i's time window), which sets $T_{max} = $ 11h 00. Even if visits to nodes *j* and *v* are scheduled as early as possible, the route is infeasible because its arrival to Lab is 11h 05, exceeding T_{max}. In the lower case, arrival to i has been delayed to 8h 05, so T_{max} =11h 05, which allows an on-time return to the Lab.

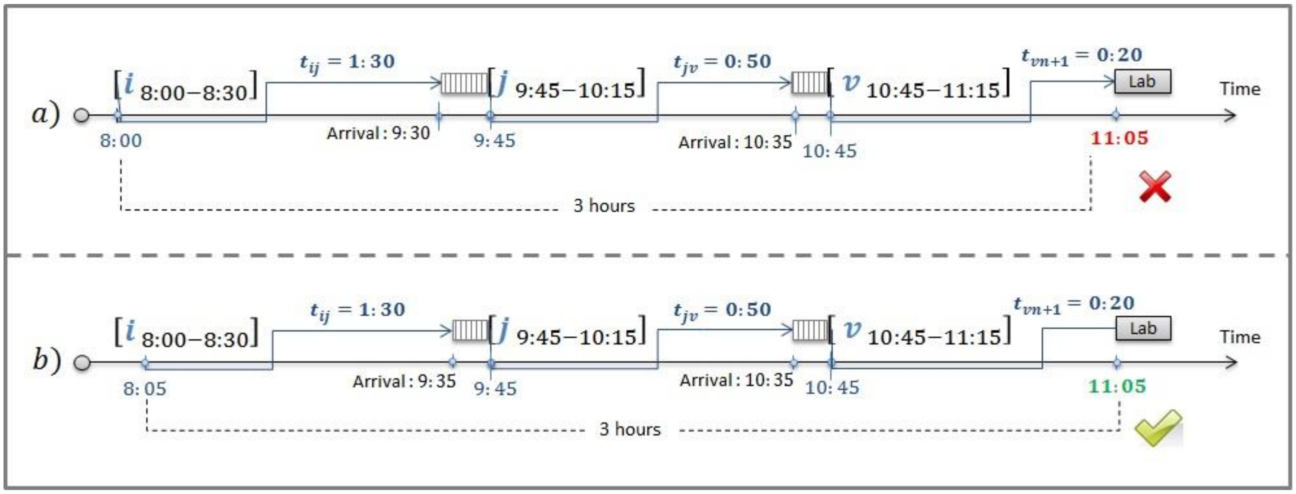

Figure 1: a) Infeasible schedule due to earliest start; b) Feasible schedule due to later start.

However, later departures are not always better. Visiting the first SCC at the end of its time window can prevent potential SCCs having similar (or the same) time windows from being included in the route. Unfortunately, there isn't an *a priori* "best route construction" policy.

To sum up, we executed both heuristics H1 and H2 several times, choosing at each execution a different request in the initialization process (independently of its order in *TR*), and for each considered initialization request j, we ran the heuristics, fixing the departure time at a_j, b_j and at the middle of j's time window ($(b_j - a_j)/2$).

5.3. Local improvement

An iterative local improvement procedure is applied to all the solutions obtained by the previously described heuristics. A feasible solution S is composed of K vehicles, each vehicle performing multiple routes. A neighborhood of a given solution is obtained by moving a request v assigned to a vehicle k to a later position in any of k's routes. If the move leads to a distance reduction, the feasibility of the neighbor solution is checked. Starting with the first vehicle, its complete neighborhood is evaluated, and the best feasible move is implemented. The procedure is repeated until no improvement is found. Then the procedure is applied to the following vehicle, until all the other vehicles schedules have been considered.

6. Computational results

This section presents the computational results over a set of 38 real instances provided by the MSSS. It presents the instances and then reports the results produced by the two formulations and compares them to those produced by the developed heuristics.

6.1 Instances

We conducted with the MSSS a detailed survey from June to August 2013 to determine the transportation needs of SCCs. The 149 SCCs of four administrative regions[1] were required to provide opening hours, the number of transportation requests and their associated time windows for each working period (a.m., p.m., night) of weekdays, weekend days and holidays. Hence, each SCC provided demand data concerning 18 different periods. Because some SCCs have the same demand in several periods, we obtained 38 different instances. For example, if the total demand of Monday a.m. is the same as that of Wednesday a.m., both periods were considered the same instance. The workload is higher Monday through Wednesday, leading to larger (more requests) instances. As fewer SCCs are open on weekends and only a few are open on holidays, the related instances are rather small. We arbitrarily divide instances into *Small* (four SCCs, around 10 requests), *Medium* (up to 10 SCCs, around 20 requests) and *Large* (up to 20 SCCs, up to 50 requests) sets.

Experts in the MSSS set the loading and unloading time to 10 minutes ($\tau_i = 10$ minutes), and $T_{max} = 180$ minutes, which means that a sample will never travel more than 180 minutes. The working shift's maximal length was fixed to $T_k = 480$ minutes. All travel times and distances were defined using GoogleMaps.

6.2 Results produced by BSTP-MR and BSTP-EG

Both formulations were solved using the commercial software Gurobi v5.5, running on a PC with two Intel Xeon X5650 2.66GHz 6 Core and 72Go de RAM. Computational time was limited to 3 600 sec. Table 1 reports the distance of the best feasible solution (*Dist.*), its gap in percentage with respect to the best lower bound (*Gap*) and the computational time (*Sec.*). Table 1 does not contain results for the larger instances because Gurobi was not able to find any integer feasible solution except for instance I-38. For this particular instance, BSTP-MR and BSTP-EG found solutions of equal distance (1 183), but the optimality gap was higher than 30%.

[1] Saguenay-Lac-Saint-Jean, Capitale-Nationale, Mauricie and Montérégie.

Table 1: Results produced by the two formulations (time limit = 3 600 sec.)

		BSTP-MR			BSTP-EG		
	Inst.	**Dist.**	**Gap**	**Sec.**	**Dist.**	**Gap**	**Sec.**
Small	I-1	199	0	0	199	0	0
	I-2	557	0	0	557	0	0
	I-3	619	0	0	619	0	0
	I-4	199	0	0	199	0	0
	I-5	324	0	0	324	0	0
	I-6	271	0	0	271	0	0
	I-7	280	0	0	280	0	0
	I-8	268	0	0	268	0	0
	I-9	312	0	0	312	0	0
	I-10	236	0	0	236	0	0
	I-11	194	0	0	194	0	0
	I-12	125	0	0	125	0	0
	Avg. :	*299*	*0*	*0*	*299*	*0*	*0*
Medium	I-13	995	8	3600	995	0	249
	I-14	991	9	3600	991	3	3600
	I-15	931	29	3600	931	0	256
	I-16	159	0	199	159	0	26
	I-17	230	0	34	230	0	23
	I-18	301	0	9	301	0	3
	I-19	126	0	0	126	0	1
	I-20	193	0	0	193	0	4
	I-21	193	0	0	193	0	2
	I-22	285	0	1	285	0	4
	I-23	754	0	1	754	0	0
	I-24	230	0	0	230	0	0
	I-25	234	0	0	234	0	0
	Avg.:	*432*	*4*	*850*	*432*	*0*	*321*

All the *Small* instances were solved to optimality in negligible time. Considering *Medium* instances, formulation BSTP-EG reached 12 optimal solutions, while BSTP-MR gave proof of optimality in 10 cases. Nonetheless, both models report the same total average distance. BSTP-EG computing times are smaller, 321 seconds on average, compared to 850 seconds for BSTP-MR.

We ran again the Large instances extending the limit on the computational time to 10 800 seconds but Gurobi was not able to find any integer feasible solution other than I-38. Still, gaps reported for I-38 remain at 29%.

Given the difficulty shown by Gurobi to find an integer solution, we decided to provide the solver with an initial feasible solution. To this end, we used the best solutions found by the heuristics presented in Section 5. Table 2 reports the results of these experiments.

Table 2: Results produced for large instances using an initial heuristic solution.

	Best Heuristic	BSTP-MR				BSTP-EG			
		3 600 sec		10 800 sec		3 600 sec		10 800 sec	
Inst.	**Dist.**	**Dist.**	**Gap**	**Dist.**	**Gap**	**Dist.**	**Gap**	**Dist.**	**Gap**
I-26	462	449	24	**433**	21	448	22	445	22
I-27	2023	1982	36	1961	36	2022	32	**1943**	29
I-28	1894	1838	35	1811	34	1851	31	**1785**	29
I-29	1899	1802	34	**1741**	31	1854	31	1845	31
I-30	1787	1721	39	1656	37	1717	31	**1633**	27
I-31	645	600	15	**582**	13	622	16	622	14
I-32	1701	1641	32	1641	32	1640	27	**1619**	24
I-33	536	494	7	**488**	6	490	7	**488**	5
I-34	500	469	5	**469**	4	475	8	**469**	5
I-35	500	469	5	**469**	4	475	8	**469**	5
I-36	2109	1937	64	**1933**	64	1933	63	**1933**	62
I-37	1923	1833	7	**1833**	7	1833	5	**1833**	3
I-38	1285	1183	21	**1183**	12	1183	20	**1183**	9
Avg:	*1328*	*1263*	*25*	*1246*	*23*	*1273*	*23*	*1251*	*20*

Column *Best Heuristic Dist.* reports the best solution found by the heuristics to each instance. We used these solutions as starting solutions for Gurobi, and we allotted 3 600 and 10 800 sec. of computational time. Both formulations were able to improve the provided initial solution. In particular, the average distance produced by the heuristics (1 328) was reduced after one hour to 1 263 and 1 273 by *BSTP-MR* and *BSTP-EG*, respectively, which represents an improvement of around 4%. Within the 10 800 seconds limit, distances were reduced to 1246 and 1251, which represents an additional improvement of around 2%. Formulations show a rather poor performance closing the optimality gap, which ranges from 3 to 64%. We conclude that even if *BSTP-MR* seems to perform slightly better than *BSTP-EG* in the context of this particular experiment, they produce quite similar results.

6.3 Results produced by the heuristics

Table 3 reports the results produced by the multistart versions of heuristics H1 and H2. In fact, for each instance, H1 was executed $3*|TR|$ times, with each execution using a different request in the initialization phase and, for each request, using the three arrival times strategies (at the beginning, at the middle and at the end of the time window). As per H2, each instance was executed $2*3*|TR|$ times because two options for the selection criterion were available (choose the closest request or the one having the earliest time window).

The left part of Table 3 reports the results of H1 and H2 while the right part shows the results produced after applying the Local improvement procedure. An asterisk * by the instance number indicates that the best-known solution is optimal. Columns under header *Deviation % g*ive the difference between the heuristic solutions and the best-known solutions in percentage.

Corresponding best distances are also reported. For those instances where both H1 and H2 *without local improvement* were not optimal, column Δ reports the difference between columns *Min.* without and with local improvement. Computational times were always under one second; therefore, they are not reported.

Table 3: Heuristics performance

	Without local improvement				**With local improvement**				
	Deviation %			**Best H1-H2**	**Deviation%**			**Best Heuristic**	
Inst.	**H1**	**H2**	**Min.**	**Dist**	**H1**	**H2**	**Min.**	**Dist**	**Δ**
I-1*	0	0	0	619				619	
I-2*	0	0	0	199				199	
I-3*	0	0	0	557				557	
I-4*	5.5	3.5	3.5	206	5.5	0	0	199	3.5
I-5*	0	0	0	324				324	
I-6*	0	0	0	268				268	
I-7*	0	0	0	271				271	
I-8*	0	0	0	280				280	
I-9*	0	0	0	236				236	
I-10*	0	0	0	312				312	
I-11*	0	0	0	194				194	
I-12*	0	0	0	125				125	
Avg:	*0.5*	*0.3*	*0.3*	*299*	*0.5*	*0*	*0*	*299*	
I-13*	4.4	0.8	0.8	1003	4.4	0.8	0.8	1003	0
I-14*	8	4.1	4.1	1031	7.8	4.1	4.1	1031	0
I-15*	2	2	2	950	2	2	2	950	0
I-16*	4.7	4.6	4.6	166	4.6	0.7	0.7	160	3.9
I-17*	6.4	6.4	6.4	245	6.4	6.4	6.4	245	0
I-18*	12.6	11.4	11.4	336	12.6	11.4	11.4	336	0
I-19*	0	0	0	285				285	
I-20*	7.9	4	4	131	7.9	4	4	131	0
I-21*	15.7	15.7	15.7	223	0	15.7	0	193	15.7
I-22*	15.7	2.6	2.6	198	0	2.6	0	193	2.6
I-23*	0	0	0	234				234	
I-24*	0	0	0	754				754	
I-25*	8.5	7.2	7.2	247	8.5	7.2	7.2	247	0
Avg:	*11.1*	*8.3*	*4.5*	*446*	*10.4*	*8.2*	*2.8*	*443*	
I-26	20.9	6.6	6.6	462	20.8	6.4	6.4	461	0.2
I-27	4.1	6.7	4.1	2023	4.1	6.7	4.1	2022	0
I-28	8.2	6.1	6.1	1894	6.8	5.5	5.5	1883	0.6
I-29	11.2	9.1	9.1	1899	9.8	8.5	8.5	1888	0.6
I-30	13.9	9.4	9.4	1787	13.3	9.4	9.4	1787	0
I-31	16.8	10.8	10.8	645	16.8	9.6	9.6	638	1.2
I-32	9.2	5	5	1701	9.2	5	5	1701	0
I-33	14.9	9.9	9.9	536	10.2	7.4	7.4	524	2.5
I-34	9.6	6.7	6.7	500	8.8	6	6	497	0.7
I-35	9.6	6.7	6.7	500	8.8	6	6	497	0.7
I-36	9.1	11.4	9.1	2109	9.1	10.7	9.1	2109	0
I-37	4.9	5.3	4.9	1923	4.9	5.3	4.9	1923	0
I-38	8.6	8.9	8.6	1285	8.6	8.4	8.4	1283	0.2
Avg:	*12.6*	*9.1*	*7.5*	*1328*	*11.7*	*8.5*	*6.9*	*1324*	*0.5*

Both heuristics H1 and H2 produced optimal solutions for all the *Small* instances but one. For I-4, H2 produced a better solution, only 3.5% worse than the optimal one. We applied the Local Improvement procedure to I-4, and the procedure was able to improve solution H2 to optimality.

For *Medium* instances, H1 and H2 produced three solutions proven as optimal. The Local Improvement procedure achieved four more optimal solutions and, in average, reduced the deviation from 4.5 to 2.8%.

Finally, for *Large* instances, the Local Improvement procedure reduced the distances in 8 out of 13 cases. In average, H1 and H2 with local improvement produced solutions 11.7 and 8.5% longer that the best-known solutions. Therefore, H2 seems to dominate H1, but since H1's solutions are better than H2's for three instances, the wise thing to do is to keep both heuristics to ensure better results. In average, Best Heuristic Dist. is 6.9% longer that the best-know solutions.

7. Conclusions and research perspectives

This article presents and formalizes the biomedical sample transportation problem faced by the health ministry of Quebec (MSSS). Although this problem is close to the multi-trip vehicle routing problem with time windows, it has particular constraints related to the perishable nature of the samples and the work organization in the network of laboratories of Quebec. We proposed two mathematical formulations and some fast heuristics to tackle this problem. Since commercial branch and bound software have shown to be unable to find integer solutions to several of our instances, we used the heuristic solutions as initial solutions for the solver. This strategy produced interesting results, but optimality gaps remain high.

Nonetheless, this first phase of the *Optilab* project has allowed the MSSS to get a precise idea of the needs for transportation and logistics related to the biomedical sample collection and analysis. Transportation schedules of good quality have been produced. These schedules can be used as references to evaluate the transportation "effort", in number of vehicles, driving time and traveling distance, required to adequately satisfy the requirements of the current biomedical analysis system in Quebec. In other words, the MSSS can now express its transportation requirements in a clear and detailed manner to interested 3PL or carriers willing to provide transportation services.

Optilab offers new and challenging perspectives. Among them, we aim to extend our experiments to the other 13 administrative regions of the Quebec province, some of which are larger than the ones study up to now, requiring the development of even more efficient solving approaches. Also, we feel that the existing network needs to be reconsidered to include, for example, optimized opening hours at certain SCCs or the SCC-to-Lab allocation.

Acknowledgments

This research was partially supported by Grants OPG 0293307 and OPG 0172633 from the Canadian Natural Sciences and Engineering Research Council (NSERC). This support is gratefully acknowledged. We also express our gratitude to Dominic Morand, *conseiller à la direction de la logistique et des équipements*, of the MSSS for his collaboration in this project.

References

Azi, N., Gendreau, M. and Potvin, J.-Y., 2014. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41, pp.167–173.

Azi, N., Gendreau, M. and Potvin, J.-Y., 2010. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3), pp.756–763.

Baldacci, R., Mingozzi, A. and Roberti, R., 2011. New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, 59(5), pp.1269–1283.

Baldacci, R., Mingozzi, A. and Roberti, R., 2012. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1), pp.1–6.

Battarra, M., Monaci, M. and Vigo, D., 2009. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research*, 36(11), pp.3041–3050.

Brandão, J. and Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100(1), pp.180–191.

Brandão, J. and Mercer, A., 1998. The multi-trip vehicle routing problem. *Journal of the Operational research society*, pp.799–805.

Bräysy, O. and Gendreau, M., 2005a. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science*, 39(1), pp.104–118.

Bräysy, O. and Gendreau, M., 2005b. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation science*, 39(1), pp.119–139.

Cattaruzza, D., Absi, N., Feillet, D. and Vigo, D., 2014. An Iterated Local Search for the Multi Commodity Multi Trip Vehicle Routing Problem with Time Windows. *Computers & Operations Research*, 51, pp.257–267.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M. and Soumis, F., 2002. VRP with time windows. In *The vehicle routing problem*. SIAM Philadelphia, pp. 157–193.

Doerner, K.F., Gronalt, M., Hartl, R.F., Kiechle, G. and Reimann, M., 2008. Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows.

Computers & Operations Research, 35(9), pp.3034–3048.

Hernández, F., Feillet, D., Giroudeau, R. and Naud, O., 2011. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR*, pp.1–25.

Kallehauge, B., 2008. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7), pp.2307–2330.

Liu, R., Xie, X., Augusto, V. and Rodríguez, C., 2013. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3), pp.475–486.

Martínez, L. and Amaya, C., 2013. A vehicle routing problem with multi-trips and time windows for circular items. *Journal of the Operational Research Society*, 64(11), pp.1630–1643.

Ministère de Santé et des Services Sociaux, 2012. Démarche d'optimisation des services offerts par les laboratoires de biologie médicale du Québec. (m.D. Québec, éd.) *Optilab express*.

Mingozzi, A., Roberti, R. and Toth, P., 2013. An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, 25(2), pp.193–207.

Olivera, A. and Viera, O., 2007. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, 34(1), pp.28–47.

Prins, C., 2002. Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1(2), pp.135–150.

Taillard, E.D., Laporte, G. and Gendreau, M., 1996. Vehicle routeing with multiple use of vehicles. *Journal of the Operational research society*, pp.1065–1070.

Wang, Z., Liang, W. and Hu, X., 2014. A metaheuristic based on a pool of routes for the vehicle routing problem with multiple trips and time windows. *Journal of the Operational Research Society*, 65(1), pp.37–48.