European Journal of Operational Research 255 (2016) 58–67

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

A practical vehicle routing problem with desynchronized arrivals to depot

Zahra Naji-Azimi[a], Majid Salari[b], Jacques Renaud[c,d,*], Angel Ruiz[c,d]

[a] Department of Management, Ferdowsi University of Mashhad, P.O. Box 91779-48951, Mashhad, Iran
[b] Department of Industrial Engineering, Ferdowsi University of Mashhad, P.O. Box 91779-48951, Mashhad, Iran
[c] Département opérations et systèmes de décision, Faculté des sciences de l'administration, Pavillon Palasis-Prince, Université Laval, 2325, rue de la Terrasse, Québec, G1V 0A6, Canada
[d] Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), Québec, Canada

A R T I C L E I N F O

Article history:
Received 12 June 2015
Accepted 7 April 2016
Available online 13 April 2016

Keywords:
Routing
Multiple routes
Time windows
Desynchronized arrivals

A B S T R A C T

The transportation of biomedical samples is a key component of healthcare supply chains. The samples are collected, consolidated into cooler boxes, and then transported to be analyzed in a specialized laboratory. Since many hospitals and samples' collection points are assigned to the same laboratory, it is important to manage the flow of samples arriving to the laboratory to avoid congestion. In other words, it is preferable to try to desynchronize the samples' arrivals by managing the vehicles' departure times and the routes ordering. We propose a mathematical model and a multi-start heuristic to minimize the route duration times and the maximum number of samples' boxes arriving at the laboratory within a given time period. Based on real data, we demonstrated that both the model and the heuristic are very efficient in solving real size instances.

1. Introduction

Vehicle routing and scheduling are key components of the efficiency of modern supply chains, where quantities of goods and raw materials should be continuously exchanged in the seamless way possible. In its most common version, the *vehicle routing problem* (VRP) is used to plan the distribution of goods from a depot to a set of customers, (Koç, Bektas, Jabali, & Laporte, 2016, Laporte, 2009, Semet, Toth, & Vigo, 2014) and subject to some constraints, like time windows (Bräysy & Gendreau, 2005a, 2005b), vehicle restrictions (Semet, 1995) and many other practical and industrial considerations (Coelho, Renaud, & Laporte, 2016). In other situations, products are exchanged between the customers, which leads to the pickup and delivery VRP (Berbeglia, Cordeau, & Laporte, 2010, Gschwind, 2015; Berbeglia, Cordeau, Gribkovskaia, & Laporte, 2007). Vehicles can also be used to bring back the customers' goods to a central depot, or consolidation point, like in the waste collection problem (Ghiani, Laganà, Manni, Musmanno, & Vigo, 2014). These practical routing problems deal with many real-world constraints and are often referred as *rich* VRP (Lahyani, Khemakhem, & Semet, 2015).

In this article, we focus on a situation arising in the healthcare supply chain context which corresponds to the daily transportation of biomedical samples from hospitals or clinics, which will be referred to as *collection points* (CP), to a single *laboratory* (Lab) where they will be analyzed. Transportation is done by a fleet of vehicles performing multiple routes during the day. As the lifespan of these samples is limited, CPs often require multiple collection requests on a given day, and each collection request is generally bounded by a time window. Samples are consolidated in cold boxes that, once collected, must arrive to the Lab within a limited specified time to preserve samples' integrity. The boxes are opened at the Lab, and each sample is manually registered in the system and bar-coded for tracking purposes during the analysis process. These manual tasks are time consuming and according to our partner's experts, the Quebec's *Ministère de la santé et des services sociaux* (Ministry of Health and Social Services – MSSS), they constitute a bottleneck of the samples' supply chain. In fact, if too many boxes arrive in a short period of time, samples are queued and may suffer long wait, which might exceed their remaining lifetime making them unsuitable to be analyzed. Our observations on the ground confirmed that some days or periods are more popular for sample collection. Thereby, reducing the maximum number of samples boxes arriving within a time period is desirable in order to normalize the Lab workload and minimize sample losses. Thus, the objective of

* Corresponding author at: Département opérations et systèmes de décision, Faculté des sciences de l'administration, Pavillon Palasis-Prince, Université Laval, 2325, rue de la Terrasse, Québec, G1V 0A6, Canada. Tel.: +1 418 656 7029.

E-mail addresses: znajiazimi@um.ac.ir (Z. Naji-Azimi), msalari@um.ac.ir (M. Salari), jacques.renaud@fsa.ulaval.ca (J. Renaud), angel.ruiz@fsa.ulaval.ca (A. Ruiz).

http://dx.doi.org/10.1016/j.ejor.2016.04.007
0377-2217/

this *vehicle routing problem with desynchronized arrivals* at the depot (VRP-DA) can be defined as minimizing the sum of the routes' traveling time as well as the maximum number of boxes arriving at the Lab within any time period of the considered planning day. This work is motivated by the reengineering of the health supply chain performed by the Quebec's government under the name of the Optilab project (MSSS, 2012). One of the first objective was to centralize the transportation needs of the collection points to obtain better tariffs from the carriers (Anaya Arenas, Chabot, Renaud, & Ruiz, 2016). Now the MSSS seeks at reducing the number of laboratories in order to better use the strategic ones. In order to do so, it clearly appears that a good management of samples arriving to the Lab would be an important issue. To the best of our knowledge, our contribution is the first one considering desynchronizing the vehicles' arrivals to the depot as this is closely related to samples' limited lifetime, which is not the case for most of classical or industrial goods.

Instead of regularizing the arrivals, many works have tried to balance the vehicles' workload. Jozefowiez, Semet, and Talbi (2009) considered a bi-objective VRP which minimized the total length of routes and the balance of routes, which is defined as the difference between the maximal route length and the minimal route length. Kritikos and Ioannou (2010) studied a VRP with time windows (VRPTW) where they balanced the load carried by each active vehicle. Baños, Ortega, Gil, Márquez, and de Toro (2013) also considered a multi-objective VRPTW with both distance and load imbalances. López-Sánchez, Hernández-Díaz, Vigo, Caballero, and Molina (2014) solved a balanced open VRP where the maximum time spent on the vehicle must be minimized. We note that, if all vehicles depart at the same time, balancing the routes will indeed concentrate the vehicle arrival times at the depot.

Closer to our context, Doerner, Gronalt, Hartl, Kiechle, and Reimann (2008) and Doerner and Hartl (2008) studied a blood collection problem which shares some characteristics with our problem. However, in their study they considered that the blood deterioration process begins right after the donation, and they used interdependent time windows. They also did not consider arrival times at the depot. Liu, Xie, Augusto, and Rodríguez (2013) considered the pickup and delivery of four demand types in a home health care system. They dealt with many specific constraints, but they did not consider a time limit for the samples' delivery to the depot, nor did they manage vehicles' arrival times. Şahinyazan, Kara, and Taner (2015) studied a system composed of bloodmobiles and shuttles which brought the collected blood to the depot to prevent spoilage. In order to maximize the collected blood and minimize the transportation cost, they managed the activities of the bloodmobiles and shuttles over a time horizon. However, they did not consider a time limit for the blood's return to the depot. Finally, and as stated before, this particular context was introduced by Anaya Arenas et al. (2016) which minimized traveling distance, but did not consider desynchronizing the arrivals at the Lab.

The remainder of this article is as follows. In Section 2 we propose the problem formulation and some valid inequalities designed to improve its solvability. A multi-start heuristic is developed in Section 3. The efficiency of this formulation and of the heuristic and the impact of desynchronized arrivals are evaluated in Section 4, based on a set of real instances obtained from the MSSS of Quebec, Canada. We also demonstrate how this new formulation and heuristic improve upon those of Anaya Arenas et al. (2016), where desynchronized arrivals were not considered. Our conclusions are presented in Section 5.

2. Problem definition and formulation

In order to formulate the vehicle routing problem with desynchronized arrivals, we need to identify both the locations of the CPs and the collections requests. The n CPs are represented by $V' = \{v'_1, \ldots, v'_n\}$ and each CP l requires Q_l collection requests, leading to a total of $p = \sum_{l=1}^{n} Q_l$ requests. Each request is composed of one box that contains several samples. Then we define a complete graph $G = \{V, A\}$, where $V = \{v_0, v_1, \ldots, v_p, v_{p+1}\}$ is the set of nodes in the network, which includes the laboratory as nodes $\{v_0, v_{p+1}\}$ where every route must start and end, and the set $P = \{v_1, v_2, \ldots, v_p\}$, being the p transportation requests. Also, we note P_l as the set of request nodes in V which corresponds to the same CP location. We consider the arc set $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j, i = 0, \ldots, p, j = 1, \ldots p+1\}$ and a travel time (t_{ij}) and a travel distance (d_{ij}) are assigned to each arc (v_i, v_j). Clearly, t_{ij} and d_{ij} are equal to zero for every (v_i, v_j) if i and $j \in P_l$ (i.e., i and j correspond to two requests from the same CP). In addition, each request needs to be served within a time window $[a_j, b_j]$. Finally, any two requests related to the same collection point cannot be on the same route.

K uncapacited vehicles are available for satisfying the transportation requests, and each vehicle can perform multiple routes ($r = 1, \ldots, R$) within a work shift, but a limit on the length of the working day (T_k) must be respected. In addition, we need to consider a loading time (τ_i) for each transportation request, as well as the vehicle's unloading time (τ_0) at the Lab before a new route can be started. Furthermore, let T^i_{max} be the maximal transportation time for the samples associated to request i. The objective is to minimize the total traveling time of the vehicle, including the waiting times, plus a weighted penalty θ associated to the maximum number of boxes arriving at the depot during the most visited time period. In the following sections we present the VRP-DA formulation followed by some valid inequalities to strengthen it.

2.1. VRP-DA formulation

The following decision variables are needed to define the VRP-DA:

- x_{ijkr} Binary variable equal to 1 if vehicle k travels from request i to request j in its route r; 0 otherwise.
- u_{ikr} Continuous variable that indicates the visit time (start of loading) of transportation request i by vehicle k in route r.
- y_{itkr} Binary variable equal to 1 if the request i performed by the r^{th} route of the vehicle k, arrives at the Lab within the t^{th} time period; 0 otherwise.
- w Highest number of boxes arriving to the Lab within any given time period (the busiest one).

The model VRP-DA reads as follows.

$$Min \sum_{k=1}^{K} \sum_{r=1}^{R} \left(u_{p+1kr} - u_{0kr}\right) + \theta w \quad (1)$$

Subject to:

$$\sum_{k=1}^{K} \sum_{r=1}^{R} \sum_{i=0}^{p} x_{ijkr} = 1 \quad j = 1, \ldots, p \quad (2)$$

$$\sum_{j \in P_l} \sum_{i=0}^{p} x_{ijkr} \leq 1 \quad l = 1, \ldots, n; \; k = 1, \ldots, K; \; r = 1, \ldots, R \quad (3)$$

$$\sum_{i=0}^{p} x_{ijkr} - \sum_{l=1}^{p+1} x_{jlkr} = 0 \quad j = 1, \ldots, p; \; k = 1, \ldots, K; \; r = 1, \ldots, R \quad (4)$$

$$\sum_{j=1}^{p} x_{0jkr} \leq 1 \quad k = 1, \ldots, K; \quad r = 1, \ldots, R \quad (5)$$

$$\sum_{j=1}^{p} x_{0jkr} - \sum_{j=1}^{p} x_{j,p+1kr} = 0 \quad k = 1,\dots,K; \quad r = 1,\dots,R \tag{6}$$

$$a_j - T_k\left(1 - \sum_{i=0}^{p} x_{ijkr}\right) \le u_{jkr} \le b_j + T_k\left(1 - \sum_{i=0}^{p} x_{ijkr}\right)$$
$$j = 1,\dots,p; \quad k = 1,\dots,K; \quad r = 1,\dots,R \tag{7}$$

$$u_{ikr} - u_{jkr} + (b_i + \tau_i + t_{ij} - a_j)x_{ijkr} \le b_i - a_j$$
$$i = 0,\dots,p; \quad j = 1,\dots,p+1;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{8}$$

$$u_{0kr} \ge u_{p+1,k,r-1} \quad k = 1,\dots,K; \quad r = 2,\dots,R \tag{9}$$

$$u_{p+1,kr} - u_{jkr} \le T^j_{max} + T_k\left(1 - \sum_{i=0}^{p} x_{ijkr}\right)$$
$$j = 1,\dots,p; \; k = 1,\dots,K; \quad r = 1,\dots,R \tag{10}$$

$$u_{p+1,kr} - u_{0,k1} \le T_k \quad k = 1,\dots,K; \quad r = 1,\dots,R \tag{11}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{t=0}^{T} y_{itkr} = 1 \qquad i = 1,\dots,p \tag{12}$$

$$y_{itkr} \le \sum_{j=1}^{p+1} x_{ijkr} \quad i = 1,\dots,p; \quad t = 0,\dots,T;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{13}$$

$$u_{p+1,kr} < \omega(t+1) + M(1 - y_{itkr}) \quad i = 1,\dots,p; \quad t = 0,\dots,T;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{14}$$

$$u_{p+1,kr} \ge \omega t - M(1 - y_{itkr}) \quad i = 1,\dots,p; \quad t = 0,\dots,T;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{15}$$

$$w \ge \sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{i=1}^{p} y_{itkr} \quad t = 0,\dots, T \tag{16}$$

$$x_{ijkr} \in \{0,1\} \quad i = 0,\dots,p; \quad j = 1,\dots,p+1;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{17}$$

$$u_{ikr} \in \mathbb{R}_+ \quad i = 0,\dots,p; \quad k = 1,\dots,K; \quad r = 1,\dots,R \tag{18}$$

$$y_{itkr} \in \{0,1\} \quad i = 0,\dots,p; \quad t = 0,\dots,T;$$
$$k = 1,\dots,K; \quad r = 1,\dots,R \tag{19}$$

$$w \in \mathbb{R}_+ \tag{20}$$

The first part of the objective function (1) minimizes the routes' total duration (end time minus starting time) which includes the necessary waiting time to respect the time windows. In our case, the total duration of the routes may be different from the total operating time of a vehicle as waiting times are allowed at the Lab between routes. The second applies a penalty factor θ to the maximum number of boxes arriving within a time period (w). In preliminary computations we found that minimizing traveling distance was inefficient, as the model simply added useless waiting times within the route to span the arrivals over the time periods. We found that minimizing the route's duration, including the waiting time, was a more logical objective, even if it is more complicated to optimize. Constraints (2) assure that each request is serviced by exactly one route. Since some of the original collection points are duplicated, constraints (3) ensure that a route visits only one original point at a time. Flow conservation is ensured by constraints (4). Constraints (5) state that truck k can start a route r or not, and if a route is started, it has to come back to the depot by constraints (6). If vehicle k in its route r performs request j, the time windows must be respected by constraints (7). Constraints (8) are the sub-tour elimination constraints. Constraints (9) state that route r of vehicle k starts later than the arrival of its route r-1. The time needed to return to the depot after visiting a request node is bounded by constraints (10) to satisfy sample lifetime. Constraints (11) set the maximum duration on any route. Constraints (12) state that the request i will arrive at the depot within a given time period t. Constraints (13) ensure that, if request i has been visited by the r^{th} route of vehicle k, the variable y_{itkr} can take the value one. Constraints (14) and (15) force the relation between the flow and y variables and discretize time into periods of ω units of time. When using these constraints in the case where $y_{itkr} = 1$, we have $\omega t \le u_{p+1kr} < \omega(t+1)$, meaning that the depot must be visited within the tth^{th} time period, thus within time ωt and $\omega(t+1)$. Constraints (16) calculate the Lab's maximum workload during the available time periods. As expressed by (16), the number of boxes arriving during a given time period corresponds to the number of locations visited by the routes which return to the depot during this period. Assuming that a sample box is collected at each request location, the workload then corresponds to the number of boxes arriving at the depot during this period. Domains of the variables are given by (17)–(20).

2.2. Valid inequalities

The solvability of the model (1)–(20) can be improved by the addition of the following groups of inequalities.

$$\sum_{j=1}^{p} x_{0jkr} - \sum_{j=1}^{p} x_{0jkr-1} \le 0 \quad k = 1,\dots,K; \quad r = 2,\dots,R \tag{21}$$

$$u_{ikr} - \sum_{j=0\&i\neq j}^{p} (a_j + \tau_j - a_i + t_{ji})x_{jikr} \ge a_i$$
$$i = 1,\dots,p+1; \quad k = 1,\dots,K; \quad r = 1,\dots,R \tag{22}$$

$$\sum_{j=1}^{p} x_{0jk1} - \sum_{j=1}^{p} x_{0jk-1,1} \le 0 \quad k = 2,\dots,K \tag{23}$$

$$\sum_{r=1}^{R}\sum_{i=1}^{p} x_{ijkr} - \sum_{l=1}^{j-1}\sum_{r=1}^{R}\sum_{i=1}^{p} x_{ilk-1,r} \le 0 \quad j = 1,\dots,p; \quad k = 2,\dots,K \tag{24}$$

$$\sum_{i=1}^{p}\sum_{r=1}^{R}\sum_{t=0}^{T} y_{itkr} \le M \sum_{i=1}^{p}\sum_{t=1}^{T} y_{itk-1,1} \qquad k = 2,\dots, p \tag{25}$$

$$\sum_{i=1}^{p}\sum_{t=0}^{T} y_{itkr} \le M \sum_{i=1}^{p}\sum_{t=1}^{T} y_{itk,r-1} \quad k = 1,\dots,K; \quad r = 2,\dots,R \tag{26}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijkr} \le |S| - 1 \quad S \subseteq \{1,2,\dots,p\}, \; |S| = 2 \text{ or } 3 \tag{27}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{R} x_{ijkr} = 0 \quad \forall i,j \in V\backslash\{0,p+1\} | (a_i + \tau_i + t_{ij} > b_j) \tag{28}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{R} x_{ijkr} = 0 \quad \forall i,j \in V\backslash\{0,p+1\} | (a_j - b_i + \tau_j + t_{j,p+1} > T_{max}) \tag{29}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{l=1}^{t-1} y_{ilkr} = 0 \quad \forall i \in V\backslash\{0, p+1\} | \left(a_i + \tau_i + t_{i,p+1} \geq \omega t\right);$$
$$t = 1, \ldots, T-1 \tag{30}$$

To avoid symmetry, active routes are ordered by constraint (21). Constraint (22) puts a lower bound on the minimum value of each variable u. Essentially, when visiting the arc $(i,\ j)$ by the r^{th} route of vehicle k we have $x_{ijkr} = 1$ and consequently, $u_{ikr} + \tau_i + \ t_{ij} \leq u_{jkr}$. In case that $x_{ijkr} = 0$ we have $u_{ikr} - \ u_{jkr} \leq b_i - a_j$ which is always valid. Constraints (23) and (24) are adaptations of the symmetry breaking constraints proposed by Coelho and Laporte (2014). Essentially, they eliminate many symmetric solutions of same values by ordering the use of the vehicles and the assignment of requests to vehicles. Constraints (25) and (26) are symmetry defeating constraints and are only introduced to enhance the model by breaking the symmetry caused by variables y. In particular, none of the requests can be performed by routes of the vehicle k, whenever the first route of the vehicle $k-1$ has not already been used. In addition, (26) states that route r of the vehicle k can be used to deliver a request to the Lab only if route $r-1$ of the same vehicle has already been used. Constraint (27) is a classical sub-tours elimination constraint generated for subsets of two and three requests. Constraints (28) to (30) are generated only if specific conditions are respected. Constraint sets (28) and (29) state that we cannot visit arc (i, j) by different routes of available vehicles when $a_i + \tau_i + t_{ij} > b_j$ and $a_j + \tau_j + t_{j,p+1} - b_i > T_{max}$, respectively. In fact, (28) and (29) respectively remove redundant arcs violating the time window and maximum sample travel time. Constraint (30) shows the relation between the earliest visit time of a request and the possible time periods where its corresponding sample can be delivered at the Lab. Essentially, when $a_i + \tau_i + t_{i,p+1} \geq \omega t$, request i cannot be delivered to the Lab sooner than the $t\text{th}^{th}$ time period.

Model (1)–(30) extends and strengthens the one proposed by Anaya Arenas et al. (2016) who used constraints (2)–(7), (9)–(11), (21), (23), (24) and the following connectivity constraint:

$$u_{ikr} + \tau_i + \ t_{ij} - u_{jkr} \leq T_k(1 - x_{ijkr}) \quad \begin{array}{c} i = 0, \ldots, p;\ j = 1, \ldots, p+1; \\ k = 1, \ldots, K;\ r = 1, \ldots, R \end{array} \tag{31}$$

We improved this constraint by using (8) and (22). In their model, they also minimized the total traveled distance which is a different objective function than (1). Nonetheless, the formulation is only able to solve efficiently small to medium sized instances, as it will be shown in the section devoted to numerical experiments. The development of a fast and efficient solving method was therefore necessary to deal with the larger real-life instances provided by our partner.

3. Heuristic algorithm

In this section, we develop a multi-start algorithm for the introduced problem. In the next sections we first describe the algorithm, followed by a detailed description of how visiting times are updated efficiently to handle the time windows and the maximal sample transportation time constraints.

3.1. Multi-start algorithm

The multi-start algorithm is based on three procedures, namely *Construction*, *Extraction-Reinsertion* and *Swap*. To ensure that different executions lead to different solutions, the algorithm sets a level of randomization which modifies two important parameters of the problem: the maximal sample transportation time (T_{max}) and the maximum length of a vehicle working day (T_k). These parameters impact the *Construction* algorithm, which may provide different initial feasible or unfeasible solutions. Parameters are adjusted according to the feasibility of the solution produced at the previous execution. Only feasible solutions are passed to the improvement steps.

Construction procedure

Initial solutions are built by a constructive method in which nodes are sequentially added to the routes. We use the following rules to select n_1, the first node to be visited by a route.

- $N_1 = \text{argmax}_{i \in P}\{t_{0i}\}$ i.e., the set of nodes $i \in P = \{\nu_1,\ \nu_2, \ldots, \nu_p\}$ whose travel time from the Lab is the greatest.
- $N_2 = \text{argmin}_{i \in N_1}\{b_i\}$ i.e., the set of nodes $i \in N_1$ whose time window upper bound is the lowest.
- n_1 = Choose a node from N_2 randomly.

At each step, to add a new node to the set of visited nodes of the working route, we verify the possibility of adding each unvisited node to every insertion place of the current route and select the insertion leading to the smallest increase in the route's total travel time. For instance, if node i is inserted between nodes j and k, the detour is computed as $t_{ji} + t_{ik} - t_{ij}$. The general framework of the construction phase is sketched in Algorithm 1, where K is the number of vehicles and R the maximum number of routes per vehicle.

Extraction-Reinsertion procedure

In this procedure, the goal is to reduce the value of the objective function (1) by repositioning some of the nodes. Thus, the length of the time period ω must be considered in the calculation of this cost. To try improving the solution, the procedure extracts a node from its location and reinserted into its best feasible position. Starting from the first route of the first vehicle, all the nodes are repositioned in all possible locations. A move is accepted as soon as it leads to an improvement, and the whole procedure is repeated until no improvement can be reached. During this procedure, we are allowed to create a new route or close an already existing one.

Swap

Following their order in the current solution, we consider each pair of nodes and their corresponding positions are swapped. This swap is applied to all possible combinations of two nodes over all the vehicles' routes. As soon as a move improves the solution's cost, it is accepted and the procedure stops whenever the swapping of all available nodes offers no more improvement.

Multi-start heuristic algorithm

The general framework of the multi-start algorithm is provided in Algorithm 2 and consists of two loops. During the inner loop's execution, the goal is to construct an initial feasible solution. Within this loop, we run the *Construction* procedure by setting different temporary values for the maximum vehicle travel time ($\overline{T_k}$) and the sample travel time ($\overline{T_{max}}$) parameters until a feasible solution is obtained (the *Repeat - Until* loop). Essentially, the algorithm is initialized at the first iteration by setting $Temp_{T_{max}} = T_{max}$ and $Temp_{T_k} = T_k$, which are the initial feasible parameters of the instance. For the other iterations, if the generated solution produced by applying the *Construction* procedure is feasible, the values of $Temp_{T_k}$ and $Temp_{T_{max}}$ are decreased by a factor α which is an input parameter (set to 0.1 in our computational experiments):

$$Temp_{T_k} = Temp_{T_k} - \alpha T_k$$
$$Temp_{T_{max}} = Temp_{T_{max}} - \alpha T_{max}$$

Otherwise, in order to increase the chance of obtaining a feasible solution, the corresponding values of the parameters are

Algorithm 1 The construction procedure.

```
For (i = 1 to K) do
    For (r = 1 to R) do
        Initialize the route r by visiting the first node n_1
        While (there is an un-routed node to be added in a feasible position) do
            Add to the route r, the node having the smallest detour in time
        End While
        r = r + 1
    End For
    k = k + 1
End For
```

Algorithm 2 The multi-start algorithm.

```
BestSolution = Ø
Temp_{T_k} = T_k
Temp_{T_max} = T_max
For (Iter=1 to Max_iter) do
  Repeat
    T̄_k = rand(0, T_k − Temp_{T_k}) + Temp_{T_k}
    T̄_max = rand(0, T_max − Temp_{T_max}) + Temp_{T_max}
    CurrentSolution = Construction (T̄_k, T̄_max)
    If (CurrentSolution is feasible)
      Temp_{T_max} = Temp_{T_max} − αT_max
      Temp_{T_k} = Temp_{T_k} − αT_k
    Else
      Temp_{T_max} = Temp_{T_max} + αT_max
      If (Temp_{T_max} > T_max) Then Temp_{T_max} = T_max
      Temp_{T_k} = Temp_{T_k} + αT_k
      If (Temp_{T_k} > T_k) Then Temp_{T_k} = T_k
    End If
  Until CurrentSolution is feasible
  CurrentSolution = Extraction-Reinsertion (CurrentSolution)
  CurrentSolution = Swap (CurrentSolution)
  If (CurrentSolution improves the cost of the best known solution) BestSolution = CurrentSolution
End For
```

increased by applying the updates:

$$Temp_{T_k} = Min\left\{Temp_{T_k} + \alpha T_k; T_k\right\}$$
$$Temp_{T_{max}} = Min\{Temp_{T_{max}} + \alpha T_{max}; T_{max}\}$$

Finally, the values to be used as the maximum travel and sample times when running the *Construction* procedure are set by applying the following relations in which $rand(0, x)$ is a random integer number between 0 and x.

$$\overline{T_k} = rand\left(0, T_k - Temp_{T_k}\right) + Temp_{T_k}$$
$$\overline{T_{max}} = rand(0, T_{max} - Temp_{T_{max}}) + Temp_{T_{max}}$$

Upon obtaining an initial feasible solution, we apply the *Extraction-Reinsertion* and *Swap* procedures to try to improve the quality of the initial solution. To do so, we use the original values for the sample travel time (i.e. T_{max}) and vehicle travel time (i.e. T_k). The algorithm stops after a given number of iterations (i.e. Max_{iter}).

3.2. Information update

In order to efficiently manage the time windows and the maximum sample transportation time constraints, we need rules to update the earliest and the latest start time of nodes in the solution. In the following, we offer an adaptation of the method proposed by Campbell and Savelsbergh (2004) for the VRP with time windows.

We represent the rth route of vehicle k by $k_r = \{c_0^{k_r}, c_1^{k_r}, \ldots, c_i^{k_r}, c_{i+1}^{k_r}, \ldots, c_{n_{k_r}}^{k_r}, c_{n_{k_r}+1}^{k_r}\}$ in which n_{k_r} is the number of nodes visited by the r^{th} route of vehicle k. We denote by $E_{c_j^{k_r}}$ and $L_{c_j^{k_r}}$, the earliest and the latest time at which node $c_j^{k_r}$ can be visited by the r^{th} route of vehicle k, respectively.

At the beginning, for each k and r, we set $E_{c_0^{k_r}} = E_{c_{n_{k_r}+1}^{k_r}} = a_0$ and $L_{c_0^{k_r}} = L_{c_{n_{k_r}+1}^{k_r}} = b_0$. In case of visiting node $c_j^{k_r}$ between nodes $c_i^{k_r}$ and $c_{i+1}^{k_r}$ we apply the relations (R1) and (R2) to update the earliest and latest visit time of $c_j^{k_r}$ as follows:

$$E_{c_j^{k_r}} = max\left\{a_{c_j^{k_r}}, E_{c_i^{k_r}} + \tau_{c_i^{k_r}} + t_{c_i^{k_r} c_j^{k_r}}\right\} \quad \text{(R1)}$$

$$L_{c_j^{k_r}} = min\left\{b_{c_j^{k_r}}, L_{c_{i+1}^{k_r}} - \tau_{c_j^{k_r}} - t_{c_j^{k_r} c_{i+1}^{k_r}}\right\} \quad \text{(R2)}$$

If $E_{c_j^{k_r}} \le L_{c_j^{k_r}}$, inserting $c_j^{k_r}$ between $c_i^{k_r}$ and $c_{i+1}^{k_r}$ will not violate the time windows, but it is still necessary to verify that both the maximum duration of the vehicle length and the maximum sample travel time are satisfied. To do so, we use the relations (R3) and (R4) to update the earliest and latest start time of nodes visited after and before $c_j^{k_r}$ as follows:

$$E_{c_s^{k_r}} = max\left\{a_{c_s^{k_r}}, E_{c_{s-1}^{k_r}} + \tau_{c_{s-1}^{k_r}} + t_{c_{s-1}^{k_r}, c_s^{k_r}}\right\} \quad s = i+1, \ldots, n_{k_r}+1 \quad \text{(R3)}$$

$$L_{c_s^{k_r}} = min\left\{b_{c_s^{k_r}}, L_{c_{s+1}^{k_r}} - \tau_{c_s^{k_r}} - t_{c_s^{k_r} c_{s+1}^{k_r}}\right\} \quad s = 0, \ldots, i \quad \text{(R4)}$$

Finally, the minimum length of the r^{th} route of vehicle k requires serving the route at the latest feasible time, by setting it to $vt_{c_{n_{k_r}+1}^{k_r}} - vt_{c_0^{k_r}}$. In this expression, $vt_{c_i^{k_r}}$ is the real visit time of

Table 1
Performance of the mathematical models on small and medium instances.

	Small instances				Medium instances		
#	Dist	BSTP-EG Second	VRP-DA Second	#	Dist.	BSTP-EG Second	VRP-DA Second
1	193.9	0.05	0.04	13	754.4	2.76	0.26
2	125.3	0.05	0.02	14	230.3	0.49	0.21
3	311.8	0.06	0.08	15	234.0	1.32	0.28
4	235.8	0.07	0.07	16	126.0	1.30	0.59
5	324.2	0.13	0.08	17	193.0	0.78	0.52
6	270.7	0.19	0.18	18	193.0	0.87	0.36
7	279.9	0.51	0.30	19	284.7	4.44	0.39
8	267.9	0.27	0.17	20	301.3	3.71	0.78
9	184.0	0.23	0.16	21	154.9	15.87	1.70
10	556.8	0.14	0.12	22	230.1	10.08	0.69
11	618.9	0.07	0.14	23	931.3	883.61	0.72
12	199.4	0.29	0.17	24	995.3	286.93	1.51
				25	991.3	66.68	4.51
Avg.		0.17	0.13	Avg.		98.37	0.96

node $c_i^{k_r}$ by the rth route of vehicle k and can be calculated as $vt_{c_i^{k_r}} = max\{a_{c_i^{k_r}},\ vt_{c_{i-1}^{k_r}} + \tau_{c_{i-1}^{k_r}} + t_{c_{i-1}^{k_r}, c_i^{k_r}}\}$, $i = 1, \ldots, n_{k_r} + 1$. In addition, $vt_{c_0^{k_r}} = L_{c_0^{k_r}}$.

Satisfaction of the maximum vehicle travel time and the maximum sample travel time are ensured by Eqs. R5 and R6, respectively:

$$vt_{c^{k_R}_{n_{k_R}+1}} - vt_{c_0^{k_1}} \leq T_k \tag{R5}$$

$$vt_{c^{k_r}_{n_{k_r}+1}} - vt_{c_1^{k_r}} \leq T_{max} \quad r = 1, \ldots, R \tag{R6}$$

In addition, $vt_{c_0^{k_r}} = vt_{c^{k_{r-1}}_{n_{k_{r-1}}+1}}$ for $r = 2, \ldots, R$.

4. Computational results

In this section we evaluate both the efficiency of the model and the heuristic to solve the set of instances in Anaya Arenas et al. (2016). These instances were obtained from the Quebec's MSSS and correspond to real biomedical sample transportation problems in four administrative regions in the province of Quebec.

The heuristic algorithms were implemented in C using Microsoft Visual Studio 2010 and executed on an Intel Core I-7 with a 3.4 gigahertz processor and 32 gigabyte of RAM. The mathematical formulations were solved using Cplex 12.6. As the algorithms developed in this article can also solve the instances in Anaya Arenas et al. (2016), our experiments will be separated in two parts. First, we compare our formulation and heuristic to those in Anaya Arenas et al. (2016). Then, we evaluate the model's behavior and the heuristic in the context of desynchronized arrivals.

4.1. Results on biomedical sample transportation instances

The proposed model VRP-DA of Section 2 can be used to solve the BSTP if we use constraints (2)–(15) and replace the objective function (1) by the distance minimization objective $\sum_{i=0}^{n}\sum_{j=1}^{n+1}\sum_{k=1}^{K}\sum_{r=1}^{R} d_{ij}x_{ijkr}$. These results are reported under columns VRP-DA in the following tables. Columns BSTP-EG correspond to the Anaya Arenas et al. (2016) model. Table 1 reports data on the small and medium instances. These instances have up to 10 collection points and 20 transportation requests.

As can be observed, both formulations solved all the instances to optimality. For medium instances, the BSTP-EG proposed by Anaya Arenas et al. (2016) required 98 seconds in averages, while our formulation required only 0.96.

Table 2 reports the results for the larger instances having up to 20 collection points and 50 requests. We ran the BSTP-EG model, but it was never able to produce any feasible solution after 10 800 seconds of computing. This behavior was also observed by Anaya Arenas et al. (2016) and this is why they initialized their model with the best solution given by their heuristics. Thus, column BSTP-EG in Table 2 reports their original results (distance and final Cplex gap in percentage after 10 800 seconds). Column VRP-DA reports our results after 3 600 and 10 800 seconds of computing time, respectively. Values marked by an asterisk indicate instances for which Cplex ran out of memory and the best feasible solutions for them are reported.

The new model clearly offers a better performance, obtaining better solutions for all the cases except instances 33 and 38. The average solution cost was reduced from 1388 for the previous model to respectively 1322 and 1290 after 3600 and 10,800 seconds of computing time with the new VRP-DA model. Average optimality gap was also reduced from 21.51 percent for BSTP-EG to 8.77 percent for VRP-DA. Thus, the new formulation is shown to be faster and more effective.

Table 3 reports heuristics' results for all the 38 instances. Column "Anaya-Arenas" reports their best found results over different heuristic combinations. Columns "Heuristic" report the results of the algorithm developed in Section 4 after 1 and 100 iterations, respectively. Computing times are not reported, as they are negligible. Our multi-start algorithm can be applied to the BSTP by using the travel time as a cost function for the Extraction-Reinsertion and Swap procedures and by setting the value of $\theta = 0$. For each heuristic, we report the distance (Dist) and the gap (Gap), with

Table 2
Performance of the models on large instances.

	BSTP-EG (10 800 seconds)		VRP-DA (max 3 600 seconds)			VRP-DA (max 10 800 seconds)		
#	Cost	Gap	Cost	Gap	second	Cost	Gap	second
26	1193.2	26.8	1193.0	0.0	58.7	1193.0	0.0	58.7
27	1832.8	1.2	1832.8	0.0	502.2	1832.8	0.0	502.2
28	2108.9	56.2	1932.9	20.1	3 600	1932.9	11.5	10 800
29	497.0	8.3	469.5	4.1	3 600	468.9	3.9	10 800
30	523.7	10.1	484.8	5.1	3 600	484.3	3.6	10 800
31	636.8	13.7	578.8	6.9	3 600	569.6*	6.1	5560
32	1700.7	23.3	1602.5	15.2	3 600	1551.4	11.4	10 800
33	586.7	6.9	586.9*	8.1	OOM	586.9*	8.1	OOM
34	1787.0	29.7	1660.7	22.3	3 600	1560.3	15.7	10 800
35	1883.1	28.5	1725.8	17.8	3 600	1692.9	14.1	10 800
36	1888.3	29.6	1867.9	21.4	3 600	1689.6	12.2	10 800
37	2022.3	29.0	1929.0	18.7	3 600	1928.5	18.6	OOM
38	445.1	16.3	*	*	*	*	*	*
Avg.	1388.4	21.5 percent	1322.0	11.6 percent		1290.9	8.7 percent	

Table 3
Heuristics' performance for the BSTP instances.

#	Best Cplex	Anaya-Arenas Dist	Anaya-Arenas Gap	Heuristic (1 it.) Dist	Heuristic (1 it.) Gap	Heuristic (100 it.) Dist	Heuristic (100 it.) Gap
1	193.9	193.9	0.00	193.9	0.00	193.9	0.00
2	125.3	125.3	0.00	125.3	0.00	125.3	0.00
3	311.8	311.8	0.00	311.8	0.00	311.8	0.00
4	235.8	235.8	0.00	235.8	0.00	235.8	0.00
5	324.2	324.2	0.00	324.2	0.00	324.2	0.00
6	270.7	285.8	5.58	270.7	0.00	270.7	0.00
7	279.9	292.6	4.54	279.9	0.00	279.9	0.00
8	267.9	286.8	7.05	267.9	0.00	267.9	0.00
9	184.0	184.0	0.00	184.0	0.00	184.0	0.00
10	556.8	556.8	0.00	556.8	0.00	556.8	0.00
11	618.9	618.9	0.00	618.9	0.00	618.9	0.00
12	199.4	206.4	3.51	206.1	3.36	199.4	0.00
13	754.4	754.4	0.00	754.4	0.00	754.4	0.00
14	230.3	246.8	7.16	230.3	0.00	230.3	0.00
15	234	234.0	0.00	237.9	1.67	234.0	0.00
16	126	131.0	3.97	126.0	0.00	126.0	0.00
17	193	193.0	0.00	193.0	0.00	193.0	0.00
18	193	193.0	0.00	223.3	15.70	193.0	0.00
19	284.7	284.7	0.00	284.7	0.00	284.7	0.00
20	301.3	330.4	9.66	301.3	0.00	301.3	0.00
21	154.9	160.0	3.29	163.1	5.29	154.9	0.00
22	230.1	244.8	6.39	269.3	17.04	230.1	0.00
23	931.3	949.6	1.96	931.3	0.00	931.3	0.00
24	995.3	1003.2	0.79	995.3	0.00	995.3	0.00
25	990.5	1031.2	4.11	990.5	0.00	990.5	0.00
Avg.	367.5	375.1	2.32 percent	371.0	1.72 percent	367.8	0.00 percent
26	1193.0	1229.3	3.04	1257.3	5.39	1193.0	0.00
27	1832.8	1923.3	4.94	1832.8	0.00	1832.8	0.00
28	1932.9	2108.9	9.11	1932.9	0.00	1932.9	0.00
29	468.9	497.0	5.99	468.9	0.00	468.9	0.00
30	484.3	523.7	8.14	511.0	5.51	484.3	0.00
31	569.6	636.8	11.80	612.4	7.51	572.7	0.54
32	1551.4	1700.7	9.62	1582.5	2.00	1552.5	0.07
33	586.9	638.4	8.77	626.8	6.80	575.2	−1.99
34	1560.3	1787.0	14.53	1658.6	6.30	1560.3	0.00
35	1692.9	1883.1	11.24	1879.1	11.00	1683.2	−0.57
36	1689.6	1888.3	17.76	1886.1	11.63	1688.2	−0.08
37	1928.5	2022.3	4.86	2011.3	4.29	1793.7	−6.99
38	445.1	460.7	3.50	446.1	0.22	432.6	−2.81
Avg.	1225.86	1330.73	8.25 percent	1285.06	4.67 percent	1213.10	−0.91 percent

respect to the best solution obtained by Cplex (solution to instances 1 to 27 are proven optimal). For small and medium size instances 1 to 25, the proposed heuristic with 100 iterations produced all of the optimal solutions. For the larger instances, we improved the average gap produced by Anaya Arenas et al. (2016) heuristic, 8.25 percent, to 4.67 percent when only 1 iteration of the heuristic is run, and to −0.91 percent after 100 iterations. Even more, the multi-start heuristic produced solutions that improved the best known ones (produced by Cplex) in 5 times.

4.2. Results for the VRP with desynchronized arrivals

In this section we analyze the ability of the VRP-DA formulation and the multi-start heuristic to minimize the largest number of boxes arriving to the Lab during any time period. Our computational experiments were still based on the Anaya Arenas et al. (2016) set of instances, as they are the practical foundation of this work. However, for each instance, we set the opening and closing times of the laboratory and then divided the working hours into time periods of ω minutes. Clearly the length of these periods not only influences the size of the model, but also the maximal number of boxes arriving within a period. Managing the time periods adds a new complexity level to the problem and impacts the size of the instances that can be solved to optimality. This is why the instances in this section are divided into to two sets, with instances 1 to 19 in the first one and instances 20 to 38 in the second one. The VRP-DA model can only be solved to optimality by Cplex for the latter set.

Let us first analyze the behavior of model VRP-DA when the synchronization factor is not considered. To this end, we set the penalty factor $\theta = 0$ in the objective function (1). The model optimizes only the sum of the route's durations and we observe for each instance the number of boxes arriving to the Lab at the busiest period.

Table 4 reports the numerical results produced for these experiments when ω was set to 60 and 30 minutes. Columns $\overline{RD}$ provide the optimal duration of the routes. The routes' duration produced by the heuristic are reported in columns *RD*. Column *w* reports the number of boxes arriving during the busiest period, giving an idea of the values that one should obtain if desynchronization is not consider in the optimization process. Finally, columns *Second* report the Cplex computing time in seconds. We do not report the heuristic computing time as it is always below a second.

As expected, the routes' durations are not impacted by the length of the considered period. However, the average computing time of Cplex rises from 269 seconds to 331 seconds when the time period is reduced from 60 to 30 minutes. We can observe the excellent performance of the heuristic (with 100 iterations), which was able to find the optimal route's duration for all instances. It is also worth to mention that both methods led to the

Table 4
Results for the VRP-DA with no penalty on the number of arrivals ($\theta = 0$).

	Time period $\boldsymbol{\omega} = 60$					Time period $\boldsymbol{\omega} = 30$				
	Cplex			Heuristic		Cplex			Heuristic	
#	$\overline{RD}$	w	Second	RD	w	$\overline{RD}$	w	Second	RD	w
1	270.0	1	0.2	270.0	1	270.0	1	0.2	270.0	1
2	179.0	2	0.1	179.0	2	179.0	2	0.1	179.0	2
3	378.0	2	0.4	378.0	2	378.0	2	0.9	378.0	2
5	444.0	2	2.1	444.0	3	444.0	2	1.8	444.0	2
6	444.0	4	5.6	444.0	2	444.0	2	10.2	444.0	2
7	426.0	4	3.9	426.0	4	426.0	2	8.2	426.0	2
8	431.0	2	5.0	431.0	2	431.0	2	11.9	431.0	2
9	423.0	3	9.1	423.0	3	423.0	3	27.2	423.0	3
12	311.6	3	4.8	311.6	3	311.6	3	8.0	311.6	3
13	853.0	4	77.2	853.0	4	853.0	3	392.0	853.0	4
14	347.6	7	285.0	347.6	7	347.6	4	577.0	347.6	4
15	515.0	3	9.5	515.0	3	515.0	3	28.6	515.0	2
16	306.6	4	3600.0	306.6	4	306.6	4	3600.0	306.6	4
17	330.6	4	83.0	330.6	4	330.6	4	331.0	330.6	4
18	337.6	4	11.2	337.6	4	337.6	4	19.8	337.6	4
19	599.0	4	207.9	599.0	4	599.0	3	279.1	599.0	3
Avg.	412.2	3.3	269.1	412.2	3.25	412.2	2.8	331.0	412.2	2.75

Table 5
Results for the VRP-DA with penalty on the number of box arrivals ($\theta = 100$).

	Time period $\omega = 60$							Time period $\omega = 30$					
	Cplex					Heuristic		Cplex				Heuristic	
#	$\overline{RD}$	$\bar{w}$	RD	w	Second	RD	w	$\bar{w}$	RD	W	Second	RD	w
1	270.0	1	270.0	1	0.1	270.0	1	1	270.0	1	0.2	270.0	1
2	179.0	2	179.0	2	0.1	179.0	2	1	179.0	2	0.1	179.0	2
3	378.0	1	384.0	1	0.3	378.0	2	1	384.0	1	0.7	378.0	2
5	444.0	1	525.0	1	2.9	444.0	3	1	512.0	1	4.8	444.0	2
6	444.0	1	470.0	1	6.7	444.0	2	1	470.0	1	13.2	444.0	2
7	426.0	1	471.0	1	5.9	448.0	2	1	458.0	1	14.0	426.0	2
8	431.0	1	456.0	1	7.2	431.0	2	1	456.0	1	13.7	431.0	21
9	423.0	1	486.0	1	22.2	439.0	2	1	486.0	1	33.1	486.0	
12	311.6	2	330.1	2	15.3	351.6	2	1	330.1	2	57.4	330.1	2
13	853.0	1	892.0	2	202.3	880.0	3	1	894.0	1	319.1	867.0	3
14	347.6	1	390.1	2	1598.5	415.1	2	1	472.1	1	3600.0	375.1	2
15	515.0	2	522.0	2	14.7	558.0	2	1	515.0	2	103.1	515.0	2
16	306.6	1	346.3	2	3600.0	359.3	2	1	412.0	1	3600.0	342.6	2
17	330.6	1	339.6	2	145.2	387.6	3	1	330.6	2	1077.8	483.3	2
18	337.6	2	349.6	2	15.6	377.6	3	1	337.6	2	56.4	394.0	2
19	599.0	2	623.0	2	285.7	607.0	3	1	599.0	2	231.9	623.0	2
Avg.	412.2	1.3	439.6	1.5	370	435.6	2.3	1	444.1	1.3	571	436.7	1.9
Gap(percent)			6.5			6.3			8.2			6.8	

same w value in almost all but four cases. This can be explained by the presence of an equivalent solution with respect to the route duration.

Table 5 reports the results when we set the penalty factor to $\theta = 100$, which means that the model reaches a compromise between minimizing the route's duration and the number of arrivals during the busiest period. In column $\overline{RD}$, we report the lower bound on the route duration as per Table 4, which is valid for both values of ω. In order to obtain a lower bound on the number of box arrivals in the busiest period, we ran the model with a very high penalty factor ($\theta = 100\,000$). These results are reported in columns $\bar{w}$.

When we set $\omega = 60$ minutes and $\theta = 100$, Cplex produced optimal solutions having in average a route duration of 439.6 instead of 412.2 units (lower bound in Table 4). However, the average number of boxes arriving during the busiest period is reduced from 3.3 to 1.5 boxes. The heuristic produced average values of 435.6, and 2.3 for the same indicators. When we set $\omega = 30$ minutes, the average number of boxes arriving during the busiest period produced by Cplex decreases to 1.3, whereas the lower bound ($\bar{w}$) is equal to 1. For this case, and comparing to the results in Table 4, we can say that a better desynchronization has allowed reducing the maximal number of boxes received within the busiest period from 2.8 to 1.3 at the cost of an additional 8.2 percent in the total route duration the reduction.

Table 6 reports the results produced for the larger instances. Since Cplex was unable to produce a feasible solution after 3 600 seconds of computing time, only heuristic results are reported. Experiments were run for penalty factor $\theta = 100$ and $\omega = 60$ minutes. To evaluate the heuristic's robustness and how the number of iterations influences the quality of the solutions, we report the results produced right after the construction phase of the heuristic and after 1, 10 and 500 iterations. Computing times are only reported for 500 iterations, as they are otherwise negligible. Finally, since we do not have bounds on the route duration and the number of arrivals, we ran 10 000 iterations of the heuristic to try and get "good" bounds. To this end, first we set $\theta = 0$ to minimize the route's duration (column RD'), and then we set $\theta = 100\,000$ to minimize the number of arrivals during the busiest period (column w').

Table 6
Heuristic performance for $\theta = 100$, $\omega = 60$.

#	RD'	w'	Construction		Heuristic (1 it.)		Heuristic (10 it.)		Heuristic (500 it.)		
			RD	w	RD	w	RD	w	RD	w	Second
20	496.0	2	606.0	6	560.0	2	551.0	2	551	2	0.2
21	460.0	3	592.0	4	479.0	3	479.0	3	471.0	3	0.2
22	538.0	2	575.0	4	538.0	3	538.0	3	567.0	2	0.2
23	1077.0	3	1216.0	4	1214.0	3	1105.0	3	1105.0	3	0.3
24	1210.0	3	1273.0	6	1272.0	5	1249.0	3	1249.0	3	0.3
25	1260.0	3	1341.0	6	1285.0	3	1285.0	3	1285.0	3	0.4
26	1404.0	3	1584.0	8	1520.0	5	1422.0	4	1473.0	3	0.6
27	2055.0	3	2207.0	4	2088.0	4	2104.0	3	2104.0	3	1.1
28	2271.0	5	2445.0	8	2321.0	6	2284.0	6	2328.0	5	1.3
29	957.3	4	988.3	10	985.3	9	1014.5	4	987.5	4	2.0
30	1028.3	4	1105.3	12	1060.3	6	1062.3	5	1052.3	5	1.7
31	1167.5	5	1249.3	13	1212.3	7	1241.3	6	1214.3	5	4.5
32	1865.0	5	2176.0	10	2028.0	8	1998.0	5	1998.0	5	2.3
33	1192.5	5	1280.3	12	1267.3	7	1270.5	5	1265.5	5	4.8
34	1978.0	5	2264.0	12	2263.0	7	2084.0	6	2137.0	5	3.6
35	2088.0	6	2497.0	12	2302.0	7	2176.0	7	2153.0	7	3.4
36	2110.0	6	2525.0	12	2407.0	7	2269.0	7	2199.0	6	3.2
37	2217.0	6	2627.0	11	2450.0	7	2300.0	7	2352.0	6	3.6
38	1224.9	6	1344.4	15	1277.1	7	1267.5	7	1300.5	6	4.8
Avg	1400.0	4.2	1573.5	8.9	1501.5	5.6	1457.9	4.7	1462.7	4.3	1.9
Gap(percent)			12.4	111	7.3	33.3	4.1	11.9	4.5	2.4	

Table 6 shows that the lowest route duration was achieved with 10 iterations with a deviation of 4.1 percent from the minimum (obtained with $\theta = 0$). With 500 iterations, the heuristics improve the maximum number of arrivals to 4.3, while the minimum (obtained with $\theta = 100\,000$) was 4.2, however the routes duration was increased slightly. Thus, running the heuristic with 500 iterations seems a good compromise while the computing time remains below five seconds.

5. Conclusions

This article deals with an important practical transportation problem encountered in the health system of the Quebec province. This problem requires the transport of biomedical samples from collection points to a laboratory where they are analyzed. As the laboratory is the bottleneck of the system, it is important to avoid congestion by balancing the arrival of samples. To this end, we discretize the working hours into periods and try to minimize the number of samples boxes arriving to the laboratory in the busiest period. We modeled the problem as a vehicle routing problem with desynchronized arrivals (VRP-DA). To the best of our knowledge, it is the first time that this problem has been addressed and modeled. We formulated it as a MIP and developed a heuristic to solve it. The formulation and the heuristic were adapted to deal with a similar routing problem in which desynchronization was not considered, and both outperformed existing methods. Our computational results demonstrated also that the problem with desynchronized arrivals is much more difficult to solve to optimality. For medium size real instances, the formulation was solved to optimality within few seconds. However, for larger instances, the Cplex was unable to obtain any feasible solution, although the heuristic proved to be efficient in minimizing both the route's duration and the number of arrivals during the busiest period.

The algorithms developed in this research have been applied to four administrative regions in the Quebec province. On an annual basis, more than 2.1 million kilometers are involved in the new routes configurations (Renaud, Ruiz, Chabot, Anaya Arenas, & Zue Ntoutoume, 2014). We are currently working with the Ministry to reorganize transportation operations of the other 13 administrative regions as part of the global laboratories optimization project. From a broader perspective, we believe that this practical situation, where the route planner might avoid congestion at depots or other facilities, justifies additional research.

Acknowledgments

This research was partially supported by grants OPG 0293307 and OPG 0172633 from the Canadian Natural Sciences and Engineering Research Council (NSERC), by grant 37348 from the Research Deputy of Ferdowsi University of Mashhad and by the Ministry of Health and Social Services of Quebec under contract GQ104976. This support is gratefully acknowledged. We express our gratitude to Dominic Morand, conseiller à la direction de la logistique et des équipements, of the MSSS for his collaboration on this project and for providing us with real data. The authors are grateful to the three anonymous referees for their useful and constructive comments and suggestions.

References

Anaya Arenas, A.-M., Chabot, T., Renaud, J., & Ruiz, A. (2016). Biomedical sample transportation in the province of Quebec: A case study. *International Journal of Production Research, 54*, 602–615.

Baños, R., Ortega, J., Gil, G., Márquez, A. L., & de Toro, F. (2013). A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering, 65*, 286–296.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP, 15*, 1–31.

Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research, 202*, 8–15.

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

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

Campbell, A. M., & Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science, 38*, 369–378.

Coelho, L. C., & Laporte, G. (2014). Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics, 155*, 391–397.

Coelho, L. C., Renaud, J., & Laporte, G. (2016). Road-based goods transportation : A survey of real-world logistics applications from 2000 to 2015. *INFOR : Information Systems and Operations Research* in press http://dx.doi.org/10.1080/03155986.2016.1167357.

Doerner, K. F., Gronalt, M., Hartl, R. F., Kiechle, G., & Reimann, M. (2008). Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. *Computers & Operations Research, 35*, 3034–3048.

Doerner, K. F., & Hartl, R. (2008). Health care logistics, emergency preparedness, and disaster relief: New challenges for routing problems with a focus on the Austrian Situation. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem* (pp. 527–550). Springer.

Ghiani, G., Laganà, D., Manni, E., Musmanno, R., & Vigo, D. (2014). Operations research in solid waste management: A survey of strategic and tactical issues. *Computers and Operations Research, 44*, 22–32.

Gschwind, T. (2015). A comparison of column-generation approaches to the synchronized pickup and delivery problem. *European Journal of Operational Research, 247*, 60–71.

Jozefowiez, N., Semet, F., & Talbi, E.-G. (2009). An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research, 195*, 761–769.

Koç, C., Bektas, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research, 249*, 1–21.

Kritikos, M. N., & Ioannou, G. (2010). The balanced cargo vehicle routing problem with time windows. *International Journal of Production Economics, 123*, 42–51.

Lahyani, R., Khemakhem, M., & Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research, 241*, 1–14.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*, 408–416.

Liu, R., Xie, X., Augusto, V., & 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*, 475–486.

López-Sánchez, A. D., Hernández-Díaz, A. G., Vigo, D., Caballero, R., & Molina, J. (2014). A multi-start algorithm for a balanced real-world open vehicle routing problem. *European Journal of Operational Research, 238*, 104–113.

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*.

Renaud, J., Ruiz, A., Chabot, T., Anaya Arenas, A. A., & Zue Ntoutoume, J. L. (2014). *Modélisation et optimisation du transport d'échantillons en biologie médicale*. Technical report for the. Ministry of Health and Social Services, Gouvernement du Québec.

Şahinyazan, F. G., Kara, B. Y., & Taner, M. R. (2015). Selective vehicle routing for a mobile blood donation system. *European Journal of Operational Research, 245*, 22–34.

Semet, F. (1995). A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research, 61*, 45–66.

Semet, F., Toth, P., & Vigo, D. (2014). Classical exact algorithms for the capacitated vehicle routing problem. In P. Toth, & D. Vigo (Eds.), *Vehicle routing* (pp. 37–57). Philadelphia: SIAM.