Computers & Industrial Engineering 192 (2024) 110196

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Iterative matheuristic for the biomedical sample transportation problem

Chiara Mazzanti [a], Ana María Anaya-Arenas [b,f,*], Valérie Bélanger [c,f], Ettore Lanzarone [a,d], Angel Ruiz [e,f]

[a] Institute for Applied Mathematics and Information Technology (IMATI), Consiglio Nazionale delle Ricerche (CNR), Milan, Italy
[b] Department of Analytics, Operations and IT, ESG-UQAM, Montreal, Canada
[c] Department of Operations Management and Logistics, HEC Montreal, Montreal, Canada
[d] Department of Management, Information and Production Engineering, University of Bergamo, Dalmine (Bg), Italy
[e] Department of Operations and Decisions Systems, Faculty of Administration Sciences, Universite Laval, Quebec City, Canada
[f] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal and Quebec City, Canada

ARTICLE INFO

Keywords:
Biomedical samples transportation problem
Healthcare logistics
Routing with interdependency
Synchronization
Blood transportation
Fix-&-Optimize (F-&-O)

ABSTRACT

This paper proposes an iterative matheuristic for solving the biomedical sample transportation problem (BSTP), which is a routing problem with multiple and interdependent visits in the context of healthcare services. In this problem, the biomedical samples are collected from individuals at a set of healthcare or specimen collection centers and must be transported to designated laboratories to be analyzed. The perishable nature of the specimens forces to visit the collection centers more than once a day to ensure that the time from the moment they are drawn to the arrival at the laboratory do not exceed the samples lifespan. Also, a visit to one center imposes (1) a limit on the duration of the route that transports its samples to the laboratory, and (2) a limit on the latest time at which the same center must be visited again, creating an interdependency between visits, routes and the decision concerning the centers' opening times. This paper first proposes a mathematical formulation to model the BSTP. Since this formulation is not able to solve medium or large sized instances efficiently, it also proposes an iterative matheuristic, which includes two main steps. The first step produces an approximated solution to the BSTP by a decomposition approach that splits the problem into a series of smaller subproblems that are solved by the proposed mathematical formulation. In the second step, two fix-&-optimize strategies are used with the mathematical formulation to perform a local search around the solutions produced by the decomposition method. The matheuristic has demonstrated its efficiency solving a rich set of real-life instances corresponding to the needs of several regions in the province of Quebec, Canada, in a fraction of the time required to solve the exact mathematical formulation.

1. Introduction

Healthcare providers must usually deploy large facility networks to ensure proximity to patients. However, the increasing number of facilities leads to several challenges from a managerial perspective, especially when involving high-tech and expensive equipment, because demand consolidation and return to scale opportunities are reduced. Therefore, healthcare managers must carefully design services to balance the requirements of proximity and the system costs.

The management of biomedical samples face these challenges. They intend to multiply the collection points over the territory to facilitate service access for individuals. Moreover, they intend to minimize the cost of laboratory equipment and maximize its usage. Both goals are achievable by planning adequate networks that collect and transport samples from the various service points to a reduced number of laboratories in which they are analyzed. However, having a decentralized collection and centralized analysis requires complex logistics that must maintain costs as low as possible and respect a set of practical restrictions, the short lifetime of biomedical samples being one, if not the most difficult restriction, among them.

The Biomedical Sample Transportation Problem (BSTP) arising in the Province of Quebec, Canada inspired this contribution, although similar problems have been studied in other healthcare systems like in Italy, France, Colombia, Austria, the United States or Morocco (e.g., Ait Haddadene et al., 2016; Bonadies et al., 2020; Doerner et al., 2008; Haitam et al., 2021; Pirabán-Ramírez et al., 2022; Zabinsky et al., 2020). The BSTP was first introduced in Anaya-Arenas et al. (2016) and it studies the planning of sample transportation for Quebec's healthcare network, where the Ministry of Healthcare and Social Services (i.e. the Ministry) intended to rationalize the analysis of biomedical samples.

* Corresponding author at: Department of Analytics, Operations and IT, ESG-UQAM, Montreal, Canada.
E-mail address: anaya-arenas.ana_maria@uqam.ca (A.M. Anaya-Arenas).

https://doi.org/10.1016/j.cie.2024.110196
Received 26 September 2023; Received in revised form 30 April 2024; Accepted 5 May 2024
Available online 8 May 2024

Important investment from the Ministry was granted to fit up a small number of laboratories (ideally a lab per administrative region) with state-of-the-art equipment. These "central" laboratories should receive and analyze all samples from the different healthcare facilities in the region (i.e., local clinics, elderly centers, etc.), named henceforward specimen collection centers (SCCs), where samples are drawn from patients.

In this context, the BSTP can be seen as a tactical transportation problem, where a daily transportation plan needs to be proposed for each each region and each working day. These plans are used as detailed estimates of global transportation efforts. A plan includes the set of daily routes serving each region, and for each route, the set of SCCs they visit and the estimated schedule (i.e., arrival/departure time at each SCC) in order to warrant that none of the samples perish. Moreover, it considers that each SCC offers a certain flexibility in setting its opening time, while still having a fixed total collection time known a priori. The collection time in a SCC can be as short as a few hours, or as long as a full 24-h day. During the collection period, samples are drawn from patients on a continuous basis, then accumulated at the SCC until a transport collects them and takes them to the lab. Notice that (1) since the Lab's capacity is sufficient, the BSTP is not concerned with the operations inside the Lab, and (2) according to the Ministry, sample transportation is not part of their core business, it should therefore be outsourced to private carriers. Solutions to the BSTP should help the Ministry assessing the feasibility and the cost/effectiveness of their centralization strategy. For this reason, the Ministry is interested in the total traveled distance or route duration, the key aspect in the negotiation of contracts with carriers, rather than fleet size or number of routes. This tactical transportation plan is then executed daily and reoptimized at the operational level only in case new information becomes available.

Because several SCCs are intended to be visited sequentially in a route, the short lifespan of samples (usually 3 to 4 h) implies that (1) a single SCC requires more than one visit on a day, and (2) there is a link and a limit between the time that samples can be accumulated at the SCC, waiting to be transported, and the time required by a vehicle to transport them to a Lab.

Previous research related to sample transportation, or more generally, to the logistics of fast perishing products, handles the limited lifespan of samples by either defining time windows to control the lifespan of the perishable items at their origin (e.g., setting frequency of visits, time of visits, etc.), or assuming that the deterioration process starts only during transportation, and then restricting route length. This, however, forces mangers to set proper transportation and accumulation time limits. In this paper, we propose a different and more realistic approach. We assume that a sample starts to deteriorate as soon as it is drawn from the patient. We explicitly define lifespan constraints on samples from the time of their draw to their arrival at the Lab. This approach will result in a global and more flexible, but also more difficult, optimization problem. Finally, the flexibility granted to opening hours improves coordination between SCCs, resulting in a more efficient overall transportation plan.

This paper makes two contributions. First, it contributes to the intertwined routing and scheduling decisions of the BSTP, incorporating a novel and more realistic accountability to samples' lifespan. Second, it proposes a new and efficient iterative two-step matheuristic for solving real instances of the problem, seeking to tackle the interdependency in a smart way. The first step produces, at each iteration, a solution to the BSTP by applying a decomposition heuristic inspired by the *rolling horizon* strategy to divide the pickup decisions in smaller groups. In the second step, a mathematical formulation for the BSTP in combination with two effective *fix*-&-*optimize* (F&O) strategies are used to perform a local search around each solution provided by the first step. This method provides a practical decision support tool for determining routes in a tactical setting when facing strong time interdependencies.

The remainder of this paper is organized as follows. Section 2 summarizes relevant studies and discusses the main contributions of this research. Section 3 identifies the problem description and modeling approach for the biomedical context studied here. Then, Section 4 adapts the mathematical formulation of BTSP to address such aspects. Section 5 describes the matheuristic, and Section 6 reports the numerical results. Finally, Section 7 summarizes the main contributions of this study and suggests future research directions.

2. Literature review

The transportation of blood, and other specimens, is a logistics problem related to the delivery of health services at the local, regional, and national levels (Brailsford & Vissers, 2011). Strong time restrictions and/or precedence constraints characterize these problems, which are at least partially also present in other healthcare logistics problems. As introduced in Section 1, the lifespan of samples imposes that some SCCs may need to be visited more than once a day. Furthermore, it creates strong links between transportation decisions. The time at which a SCC is visited by a route determines two things: (1) the route length, i.e., the time left to visit other SCCs while respecting the maximum time to arrive at the Lab, and (2) the latest time at which the same SCC must be (eventually) visited again by a further route. Having these aspects simultaneously involved in the planning of daily operations is what separates the BSTP from other transportation problems. The next paragraphs review contributions in healthcare logistics that address synchronization and/or time interdependencies on visits (the first challenge of the BSTP). The second part of this section reviews the contributions to specimen and blood transportation with short lifespan. Finally, we review a few papers that are closer to our contribution by including the precedence and perishability constraint simultaneously.

Interdependency in the visit schedule is appropriate for many service applications in which some type of synchronization (or temporal precedence) constraint must be imposed. In logistics, several problems analyze the schedule of visits when a given interdependency is imposed, by a frequency of visits required, by service consistency or by inventory management, in a planning horizon of several days or weeks (Vidal et al., 2020). For instance, the tramp ship routing and scheduling problem sets a link between the departure of one or several vessels, forcing a minimum and a maximum number of days between departures (e.g., Vilhelmsen et al., 2017). Other works, like the inventory routing problem (IRP, see Coelho et al., 2014; Coelho & Laporte, 2013; Mor & Speranza, 2022, for a review), plan visits to a set of customers imposing a link between different periods of the planning horizon (e.g., the client has to be visited every two or three days). Contrary to the BSTP, problems like the IRP require a synchronization for each customer between periods in the planning horizon (and not inside a single period). Hence, routes can be planned independently for each day and there is no interdependency inside routing decisions of a day schedule.

Synchronization of visits within a single-day planning horizon has been studied before, motivated by several real-world applications in transportation. Consequently, there has been a significant growth in the literature on routing problems that incorporate synchronization. A recent review by Soares et al. (2023) presents a framework, defining two types of synchronization (operations or movement). A problem with *operations synchronization* arises when a series of tasks present interdependencies, linking several routes together in a temporal manner. *Movement synchronization* requires inter-route dependencies between two vehicles (Soares et al., 2023). Over 40% of the papers reviewed by Soares et al. (2023) define a *schedule synchronization*, as it is the case of the BSTP, in contexts such as forestry, the technician routing problem, and home healthcare, among others (e.g., Ali et al., 2021; Bredström & Rönnqvist, 2008; Euchi et al., 2021; Rousseau et al., 2013). Previous studies explicitly define a set of time windows for each requested visit and set synchronization constraints between the visits. This problem has commonly been studied under the name *vehicle routing problem with time windows and synchronization (VRPTW-Syn)* (see Bredström & Rönnqvist, 2008; Dohn et al., 2011; Drexl, 2012, for more details). Here, synchronization restrictions are used to either limit or exceed the time between two visits to the same node (e.g., a patient must be visited by the cleaning service, and this, one hour before the medical service). The BSTP shares with the VRPTW-Syn the interdependency between visits. However, our contribution does not set any time windows, making the problem more flexible but also more difficult. Furthermore, the VRPTW-Syn does not include any route length restriction, which is a key element in the BSTP. Nonetheless, due to its relevance and similarity with our problem, we provide a short review of papers in healthcare that covers this topic.

An important healthcare application for the operations synchronization problems can be found in the routing of resources for home healthcare (HHC). For example, Ait Haddadene et al. (2016) proposed a GRASPxILS metaheuristic that includes budgetary restrictions. Kergosien et al. (2014) analyzed the routing of nurses considering drop-off of samples with a specific time limit. Liu et al. (2013) proposed the pickup and delivery of goods, but the lifespan was longer than a day (single visit per customer without a time limit). Decerle et al. (2018) proposed a memetic algorithm for solving an HHC problem in France, and Frifita and Masmoudi (2020) proposed metaheuristics to solve the problem, including several specialties in the scheduling. Melachrinoudis et al. (2007), in particular, set a dial-a-ride problem for a healthcare organization in Boston (USA), and proposed a tabu search algorithm to solve real-life instances of up to five (independent) transportation requests. Owing to the complexity of the problem, past contributions of the VRPTW-Syn have primarily proposed heuristic algorithms whose performance have often been tested over the test set of Bredström and Rönnqvist (2008). This test set contains up to 80 visits instances with fixed and independent time windows, and 10% of synchronization (i.e., if there are 20 visits to schedule, exactly two of those visits need to be synchronized in time). Even if the BSTP shares the synchronization challenge of said problems, our case is more difficult. Indeed, in the case of BSTP, a visit to one SCC impacts its next visit, as well as the whole collection route, i.e., it will impose the latest arrival time to the lab. Moreover, our problem does not consider any time windows for the visits, which increases the complexity and interdependency in the decision-making. Finally, the real instances of the Ministry have a similar size in number of visits than the test set that has been usually solved in the past, but the Ministry case requires a percentage of synchronization of at least 25%, and up to 100% in many cases, making it much harder to solve (see details in Section 6.1).

We now review papers that cover a series of logistics problems in the blood supply chain (BSC). The main challenge in this field is the strict temporal restrictions imposed by the perishability of products (see Baş et al., 2016; Osorio et al., 2015; Pirabán et al., 2019 for comprehensive reviews). Since the first studies in the 1960's, over 200 papers have been published in the field (Pirabán et al., 2019). Recent studies have analyzed the design of the entire BSC network, considering the multi-echelon aspect of the problem, as it includes the collection, production, inventory, and distribution of blood products with a lifespan of a few days (e.g., Araújo et al., 2020; Baş et al., 2018; Ghandforoush & Sen, 2010; Yousefi Nejad Attari et al., 2019). However, such contributions rarely include transportation planning, or (if planned) it uses shuttles that execute several trips to a single collection point without routing. A recent contribution by Wolfinger et al. (2023) proposed a Location Routing Problem (LRP) to design a contagious disease testing network. Inspired by the challenges of COVID-19 pandemics, authors decide on the location of temporal test-centers and the routing of mobile test-teams, while allocating suspected cases to either test-centers or mobile teams. Here, no routing is done between the temporal test-centers and the labs. Moreover, the authors use different time slots for generation of cases (samples) and analysis of the laboratory. This allows a break of the interdependency, by coordinating the routing decisions with the time slots for the labs using time windows.

Considering the collection stage of the BSC, some studies aim to plan collection routes to maximize the number of processed samples and minimize transportation costs (e.g., Gunpinar & Centeno, 2016; Pirabán-Ramírez et al., 2022; Yücel et al., 2013), whereas others coordinate the appointment schedule with the transportation planning to maximize production (Mobasher et al., 2015). Şahinyazan et al. (2015) determine the schedules of mobile clinics over a week and Zahiri et al. (2018) maximize freshness of the collected samples. The main aspect that separates these contributions from the BSTP is that all these studies assume the lifespan of products is one day or longer, removing the need for more than one visit per day and, therefore, the dependency between daily routes.

The following contributions study the routing of samples removing the interdependency between visits. Anaya-Arenas et al. (2016) introduced the BSTP inspired by the needs of the Ministry of Health and Social Services of Quebec to transport samples from the SCCs to the labs. The short lifespan of the samples was addressed using independent hard time windows and a limited route duration. Naji-Azimi et al. (2016) planned the de-synchronization of trucks arrivals to the lab in the same context, and Zabinsky et al. (2020) presented a multi-trip VRP to minimize completion time for each “product” applied to a case study from Washington Medical Center. Similarly, recent contributions like Benini et al. (2019, 2022) and Detti et al. (2021) proposed a VRP with multiple independent time windows for a super laboratory in Italy. Authors proposed to consider batches of samples that need to be transported (named transportation requests) and they fixed a time window for collection and a maximum time to arrive to the Lab. Moreover, the possibility of delivery to an intermediate facility allows the samples to extend their lifespan of 90 min. Different scenarios with up to 100 batches (transportation request) are solved using a Hybrid Adaptive Large Neighborhood Search (H-ALNS). In these six studies, temporal constraints were present in the routes, due to short lifespan of samples, but the interdependency in the visits is removed using time windows for each visit. Therefore, no precedence or synchronization is necessary between visits and there is no link between the visits and the routes limits.

There are only a few contributions that tackle both aspects of our problem simultaneously. Doerner et al. (2008) were one of the first to explicitly present the interdependency created by the deterioration of the samples, including the time restrictions and the precedence constraints between the multiple and interdependent pickup time windows and verifying that no sample perishes during transportation. In their study, a savings and greedy construction heuristic was proposed to solve instances involving up to 15 customers with multiple pickups. Elalouf et al. (2018) studied a similar problem and solved cases of up to 11 customers. Finally, Anaya-Arenas et al. (2021) proposed an iterative local search algorithm (ILS) to solve the BSTP with interdependency, solving instances of up to 17 SCCs and 50 visits. In their study, the lifespan of the samples was divided by the maximum waiting time at each SCC (Δ_{max}) and maximum routing time (T_{max}). Moreover, the ILS algorithm of Anaya-Arenas et al. (2021) uses the Δ_{max} parameter to calculate fictive time windows to solve the problem efficiently. However, Anaya-Arenas et al. (2021)'s ILS algorithm is inapplicable to the generalization of the problem presented here. The difference in the lifespan calculation makes it impossible to use fictive time windows as they will overlap. Moreover, in our study, the route length is not a predetermined restriction (i.e., a known parameter) to each route, but it will depend on the other visits and routes. Hence, the lifespan calculation in Anaya-Arenas et al. (2021) and ours are different, are formulated differently, and require different solution methods. Moreover, to the best of our understanding, in all the reviewed papers, when interdependency is considered, the current state-of-the-art heuristics focus on solving instances of less than 20 customers with more than a single visit. The size of our real-life instances (up to 74 visits) and their synchronization requirements (over 50% of synchronization) are harder than what has been solved in the literature. For these reasons, we also propose an iterative matheuristic that address the challenges of timing and routing interdependency in a new way.

3. Problem description

This section describes the characteristics of the BTSP and formalizes the problem, emphasizing how the short lifespans of samples require the planning of several visits to the same SCC and the resulting intertwined links between them and the entire transportation plan.

The addressed BSTP is defined over a region that contains a set $C = \{c_1, c_2, \dots, c_n\}$ with n SCCs in which samples are drawn from individuals and accumulated. SCCs need to be visited by vehicles that collect the samples and transport them to the corresponding Lab to be analyzed. Different parameters characterize every SCC c_g ($g \in [1, \dots, n]$). Each SCC c_g is open for a collection period of fixed length O_g, from a_g until closing time $b_g = a_g + O_g$. The opening time can be chosen within a time window $[e_g, l_g]$ to grant more flexibility. Concerning logistics activities, τ_g denotes the service time required by the truck driver to pick up the samples at c_g. Unloading time for all the samples at the Lab is noted by τ_0.

We now examine the samples. As it is sought by the Ministry, the samples have a maximum time allowed from the moment a sample is drawn until its arrival at the Lab. This maximum time is specific to each sample (its lifespan). However, since each center c_g accumulates the samples it draws to form a batch that is sent to the Lab, the batch must respect the lifespan of its most urgent sample. We hence define, according to the types of samples that can be drawn at SCC c_g, parameter T^g_{max} as the shortest useful lifetime of the samples that are collected at SCC c_g.

The limitation imposed by T^g_{max} may require visiting some SCCs several times during their opening hours. The **minimum number** of visits required by c_g can be estimated as $|P_g| = \left\lceil \frac{O_g}{\hat{T}_g} \right\rceil$, where P_g denotes the set of visits to perform, and $\hat{T}_g$ is the longest time allowed between two consecutive visits at c_g, assuming the samples are brought directly to the Lab. Given t_{g0}, the traveling time from c_g to the Lab, $\hat{T}_g$ can be computed as $\hat{T}_g = T^g_{max} - t_{g0} - \tau_g$. As no sample can stay at the SCC overnight, the last visit must be planned to every SCC after closing time. The staff of each center define φ_g, which is the time they can wait for the last visit after closing.

Despite previous calculations on the minimum number of visits required by an SCC, in reality, the number of times that c_g must be visited depends on a_g and b_g, but also on the decisions made, particularly on the time at which c_g receives the first visit. Indeed, as soon as the first sample is drawn, it sets (1) the latest time at which c_g must be visited next, and (2) the latest samples' arrival time at the Lab. Similarly, when a second visit is planned (no later than the latest time set by the previous visit), a bound is set on the next visit, and so on, until the last visit is planned after the closing time of c_g. Note that the bound on the latest time for the next visit is set based on the assumption that the vehicle will travel directly and immediately to the Lab right after leaving c_g. Therefore, this bound on the latest time imposes a restriction on the end of the route for the vehicle that performs the next visit. Consequently, a single visit at c_g affects both the rest of the first vehicle's route and the end of the route of the vehicle performing the next visit to c_g. The BSTP requires that these scheduling decisions be carefully planned to ensure that a feasible solution is found to perform the $|P_g|$ visits to each SCC.

As explained in Section 1, the Ministry does not have a fleet of vehicles and relies on third-party services to execute the transportation plan for this problem. In addition, there are no physical capacity limits to consider in the transportation plan. Therefore, we do not characterize any vehicle in particular but approach the problem in terms of *routes* that need to be created. A route is a sequence of visits to one or many SCCs, starting and ending at the Lab. The return to the Lab of a given route must be done exactly in time to ensure that none of the samples picked up along the route perish. We then assume that the logistics partner can assign one or several routes of the BSTP plan to the same vehicle, according to the availability of its fleet, but this is external to our decision-making problem. The BSTP will provide as a solution a set of routes that will perform all the visits to all the SCCs, collect all the samples and bring them to the laboratory within their lifetime.

To better illustrate the intertwined relationships between routes visiting the same SCC, and between SCCs, consider Fig. 1 which sketches a part of a feasible solution for a given SCC.

In this example, c_1 has a collection period O_1 = 8 h, a time window to open between 6:00 and 6:30, a maximum time T^1_{max} = 4 h, and a maximum time to pick up after closing φ_1 = 1 h. In this solution, SCC c_1 is opened at 6:10 and the collection period starts. Then, a first visit v_1 is scheduled for 8:45 (with route K_1) to pick up all accumulated samples until then. Hence, route K_1 has to bring the samples to the Lab by 10:10 so they will not perish.

The timing decision of v_1 sets that the second visit at SCC c_1 must be scheduled in time to bring the new accumulated samples to the Lab before 12:45 (i.e., 8:45 $+T^1_{max}$). Then, v_2 is scheduled for 11:30 with route K_2, which has one hour and 15 min to arrive at the Lab, and a last visit v_3 is done at 14:40 after closing. Samples picked up at 14:40 have to be back to the Lab by 15:30, so the vehicle performing route K_3 has only 50 min to visit other SCCs and return to the Lab. This example also demonstrates the link between the opening decision and the first visit. Setting the opening time 10 min earlier (the earliest possible) means that the latest arrival of K_1 to the lab would be at 10:00. This will force a shorter route time while keeping the first visit v_1 at 8:45, or it will force that v_1 is done earlier to keep a route of 1:25 h (or a mix of both).

This planning process is challenging. Planning visits too close to their latest time reduces the flexibility to construct routes. However, planning visits too early pulls the timing of future visits, which must be performed earlier, and eventually more visits than necessary might be performed at the same SCC, leading to an unfeasible solution. This shows the importance of optimizing jointly visit times and opening decisions. Moreover, Fig. 1 illustrates a forward calculation, from opening to first visit and so on. An equivalent calculation could be done backwards, from setting the closing time, to the pickup time of last visit and so on, visiting the SCCs as early or as late as possible. However, in the global problem there is no way to predict whether it is better to use the earliest or latest strategy, nor estimating the solution in a forward or backward manner. It is critical to optimize these decisions in a single mathematical model.

The next section proposes a formulation adapted from Anaya-Arenas et al. (2021) to address this new and explicit way of considering the perishability of samples. The following formulation is the base of the algorithm proposed in Section 5.

4. Mixed integer linear programming formulation for the BSTP

The BSTP is modeled on an extended graph $G = (V, A)$ in which each SCC c_g is associated with a set of nodes P_g representing the visits c_g requires during the day. Let P be the set of all visits to perform over all SCCs, and V the set of nodes, $V = \{v_0, \dots, v_{|P|}\}$, composed by the node v_0 (Lab) and nodes $\{v_1, \dots, v_{|P|}\}$ (the visits). Without loss of generality, and to reduce symmetries, visits are labeled in such a manner that the first $|P_1|$ nodes in V, after the Lab, correspond to the visits to c_1, $\{v_1, \dots, v_{|P_1|}\}$; then, $\{v_{|P_1|+1}, \dots, v_{|P_1|+|P_2|}\}$ are the ones visiting c_2, and so on. Specifically, defining the set of indexes is possible for visits at c_1 as $I_1 = \{1, 2, \dots, |P_1|\}$ and are analogous for a general SCC c_g $I_g = \left\{1 + \sum_{h=0}^{g-1} |P_h|, 2 + \sum_{h=0}^{g-1} |P_h|, \dots, |P_g| + \sum_{h=0}^{g-1} |P_h|\right\}$, where $|P_0|$ is set to 0.

Additionally, the set of arcs can be specified as $A = \left\{(v_i, v_j) : v_i, v_j \in V,\ i \neq j,\ i, j \in \{0, \dots, |P|\}\right\}$ where each arc (v_i, v_j) is

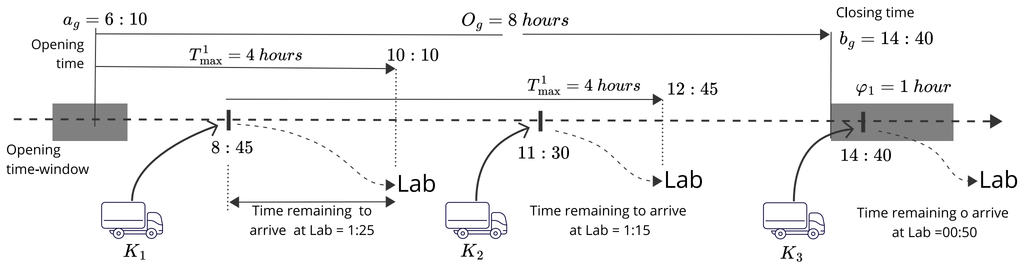

Fig. 1. Feasible BSTP solution for SCCs c_1. Opening time and visits are decided by the model depending on SCC's collection period and samples lifespan (T^1_{max}).

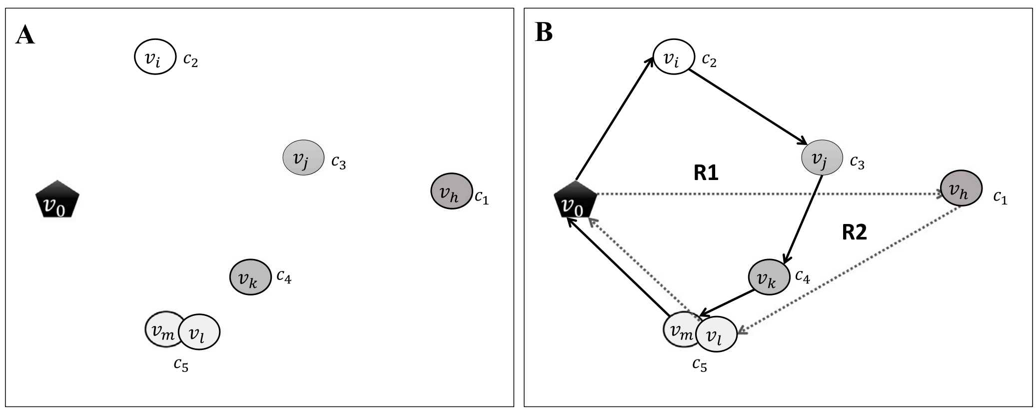

Fig. 2. Graph for the BSTP with six visits and five SCCs (A) and a feasible solution for this instance (B).

characterized by transportation time t_{ij}. As the graph models real routes, we assume that $t_{ij} \neq t_{ji}, \forall i, j \in P, i \neq j$.

Fig. 2(A) shows the graph for a small instance of the BSTP with five SCCs ($C = \{c_1, \dots, c_5\}$). Assuming that SCC c_5 requires two visits during the planning horizon, then six visits need to be performed: $\{v_h, v_i, v_j, v_k\}$, which correspond respectively to SCCs c_1 to c_4, and the two visits $\{v_l, v_m\}$ to c_5. Fig. 2(B) shows a potential solution that performs the six visits using two routes: route R_1 visiting v_i, v_j, v_k, v_m, and goes to the lab, and R_2 that visits v_h, v_l, and then goes the lab.

We propose a new formulation based on the one proposed by Anaya-Arenas et al. (2021). Taking advantage of the fact that there is no capacity constraints or other vehicle-specific features to account for, we formulate the problem as an arc-based vehicle routing problem with a two-index formulation. This significantly reduces the size of the problem (number of binary variables) and eliminates the symmetry that can be generated by the vehicle/route index in the arc variables. The formulation uses the following variables. Binary variables x_{ij} take value 1 if the arc $(v_i, v_j) \in A$ is included in the solution. Continuous variables u_i set the time to visit node v_i. Continuous variables a_g and b_g set opening and closing times of SCC c_g, respectively. Continuous variables f_i, which calculates the remaining time at node v_i to complete the route (arrive at the Lab), are timing variables used to deal with the interdependency as it will be explained in constraints (11)–(13). Finally, note that even if no time windows are fixed for each visit, the visits cannot be performed too early to avoid that more pickups than necessary will be done. Hence, waiting time before or after a collection at a node is possible, and needs to be minimized. To this end, we introduce variable d_i that counts the total duration of a route, if and only if the route starts at node v_i. Total duration will be minimized in the objective function.

In addition to the parameters introduced in Section 3, like T^g_{max}, O_g and $[e_g, l_g]$, we also set a bound on the longest lifespan of the samples, named $T_{max} = \max_{g\in[1,\dots,n]} T^g_{max}$. T_{max} is then defined as the time limit of the less restricted SCC. Given this, we can consider for each SCC the difference $\delta_g = T_{max} - T^g_{max}$, which ranges between $[0, T_{max}]$. Parameter δ_g is introduced to control the specific lifespan and time limit set by each SCC. It also ensures that the calculation of the lifespan is defined over both the accumulation and the routing time. Parameters δ_g and T_{max}, together with variables f_i and u_i, allow the model to track the specific lifespan of samples transported and the interdependency among them, without including a vehicle index for individual tracking. The proposed formulation to control the samples' lifespan is performed by constraints (12) and (13).

The sets, parameters, and decision variables to formulate the model, along with their domains and meanings, are grouped in Table 1. The notation $(A)_n$ is used to indicate the nth element of the ordered set A.

The formulation can be stated as follows:

$$\min \sum_{i=1}^{|P|} d_i \tag{1}$$

s.t.

$$\sum_{i=0}^{|P|} x_{ij} - \sum_{i=0}^{|P|} x_{ji} = 0 \qquad j = 0, \dots, |P| \tag{2}$$

$$\sum_{i=0}^{|P|} x_{ij} = 1 \qquad j = 1, \dots, |P| \tag{3}$$

$$e_g \le a_g \le l_g \qquad g = 1, \dots, n \tag{4}$$

$$a_g + O_g = b_g \qquad g = 1, \dots, n \tag{5}$$

$$u_j \ge u_i + \tau_i + t_{ij} - M(1 - x_{ij}) \qquad i = 0, \dots, |P|,\ j = 1, \dots, |P|,\ (i \neq j) \tag{6}$$

$$u_k \ge a_g \qquad g = 1, \dots, n,\ k = (I_g)_1 \tag{7}$$

$$u_k \ge u_{k-1} \qquad g = 1, \dots, n \text{ s.t. } |P_g| > 2,\ k \in I_g \text{ s.t. } (I_g)_1 < k \tag{8}$$

$$b_g \le u_k \le b_g + \varphi_g \qquad g = 1, \dots, n,\ k = (I_g)_{|P_g|} \tag{9}$$

$$T_{max} - f_i + M(1 - x_{i0}) \ge t_{i0} + \tau_i \qquad i = 1, \dots, |P| \tag{10}$$

$$f_j - f_i + M(1 - x_{ij}) \ge u_j - u_i \qquad i, j = 1, \dots, |P|,\ (i \neq j) \tag{11}$$

$$f_i - (u_i - a_g) \ge \delta_g \qquad g = 1, \dots, n,\ i = (I_g)_1 \tag{12}$$

Table 1
Notation.

Sets		
C		Specimen Collection Centers (SCC)
P		Visits to schedule for all SCCs
P_g		Visits required by the single SCC c_g
I_g		Indices for visits required by SCC c_g
V		Set of nodes in the network ($\|P\|$ visits and the Lab)
A		Set of arcs in the network
Parameters		
n		Total number of SCCs in the network
O_g		Length of the collection period at SCC c_g
e_g		Earliest opening of SCC c_g
l_g		Latest opening of SCC c_g
φ_g		Time limit after closing, set by the staff, to perform the last pickup at SCC c_g
T^g_{max}		Maximum time allowed to bring samples of SCC c_g to the Lab
T_{max}		Maximum T^g_{max} over all SCC
δ_g		Difference among T_{max} and T^g_{max}
τ_g		Loading time at SCC c_g
τ_0		Unloading time at the Lab
t_{ij}		Traveling time between centers v_i and v_j
$\hat{T}_g$		Maximum time between two consecutive pickups at SCC c_g
Decision variables		
x_{ij}	$\{0,1\}$	Takes value 1 if node v_i is visited immediately before node v_j, and 0 otherwise;
u_0	$\mathbb{R}^+$	Start time of the earliest route at the Lab v_0;
u_i	$\mathbb{R}^+$	Time at which node v_i is visited for $v_i \in P$;
d_i	$\mathbb{R}^+$	Duration of the route starting with visit v_i, for $v_i \in P$;
f_i	$\mathbb{R}^+$	Remaining time at node v_i to complete the route (arrive to Lab) for $v_i \in P$
a_g	$\mathbb{R}^+$	Opening time of SCC c_g
b_g	$\mathbb{R}^+$	Closing time of SCC c_g

$$f_i - (u_i - u_{i-1}) \geq \delta_g \qquad g = 1,\dots,n, \ \text{s.t. } |P_g| \geq 2, \ i \in I_g \text{ s.t. } i > (I_g)_1 \tag{13}$$

$$d_i \geq T_{max} - f_i + t_{0i} + \tau_0 - M(1 - x_{0i}) \qquad i = 1,\dots,|P| \tag{14}$$

$$u_0, u_i, f_i, a_g, b_g, d_i \in \mathbb{R}^+ \qquad i = 1,\dots,|P|, \ g = 1,\dots,n \tag{15}$$

$$x_{ij} \in \{0,1\} \qquad i,j = 1,\dots,|P|, (i \neq j) \tag{16}$$

The objective function (1) aims to minimize the total duration of all routes in the solution. This objective function assumes that transportation is routed by private carriers such that there is no fixed cost of vehicles. Constraints (2) ensure flow conservation in every node of the graph, whereas constraints (3) ensure that all required visits are performed while forcing the route sequence (i.e., the visit to node v_j only has one predecessor). Constraints (4) and (5) control the flexible opening window of the SCCs; in particular, constraints (4) ensure that each SCC c_g opening time is within its time opening window $[e_g, l_g]$, and constraints (5) define the closing hours of the SCCs.

Constraints (6)–(9) define the time at which each visit is performed and ensure coherency between the variables and parameters related to time. Constraints (6) set u_j, which is the time of each visit v_j, to u_i (the time of the visit to its predecessor in the route v_i such that $x_{ij} = 1$) plus the loading time at v_i (τ_i) and the traveling time from v_i to v_j (t_{ij}). Constraints (6) also force sub-tour elimination. Note that constraints (6) set arrival times for all the nodes that are visited in the network ($j = 1,\dots,|P|$) and consider all origins including the Lab ($i = 0,\dots|P|$). As all the routes start from the Lab, constraints (6) set u_0 to show the departure of the first route of the transportation plan (earliest start). Constraints (7) state that no visit can be performed at any SCC c_g before its opening time. Constraints (8) require visits to be performed in chronological order and avoid visit symmetries: this constraint is only enforced if the SCC requires more than two visits, otherwise symmetry is broken by constraints (7) and (9). Finally, constraints (9) state that the last visit at each SCC c_g is performed within φ_g units of time after its closing time b_g, with $k = \sum_{h=1}^{g} |P_h|$, or, equivalently, $k = (I_g)_{|P_g|}$.

Constraints (10)–(14) also relate to time, but model time as a *resource* consumed along the route. In particular, variables f_i are the "remaining time" resource, which allows to compute the time consumed from the Lab to each node, in the opposite direction of the vehicle. Constraints (10) state that for every visit v_i, which is performed immediately before the vehicle returns to the Lab (that is, $x_{i0} = 1$), its resource variable (f_i) is T_{max} reduced by the service time at visit v_i and transportation times between v_i and v_0. Constraints (11) link the resource and visit time variables for any pair of consecutive nodes (v_i, v_j). Indeed, if arc (v_i, v_j) is included in the route (that is, $x_{ij} = 1$), then the difference in time consumption must match the difference in the time variables of the visits. Hence, the remaining time variables at f_i calculates the time consumed in a route from the node v_i to the Lab as $T_{max} - f_i$. Moreover, f_i let us warrant that samples will not perish at the SCC nor in the route. We introduce constraints (12) and (13) to link the available time to accumulate samples in a SCC, the time that will take to deliver them in its route (according to f_i) and warrant that it will respect the time of parameter δ_i. Constraints (12) and (13) are the precedence constraints between visits. They ensure that the samples do not perish by requiring that the resource of remaining time f_i, is sufficient to cover the time the samples waited at the SCC between the opening and the first visit (sonstraints (12)) or between two consecutive visits (constraints (13)), respecting their lifespan. In other words, the sum of the time the samples stay at the SCC, plus the time required to bring them to the Lab, cannot exceed the lifespan of the samples. Constraints (10)–(13) are the set of constraints that allow us to model and control the interdependency of the problem, linking both the precedence constraints of the visits and the routing decisions. For a detailed numerical example of how to calculate constraints (10)–(13), please refer to Appendix A. Constraints (14) define the duration of a route starting at node i. Indeed, if arc $(0, i)$ is in the solution, a route starting from node i exists, and its duration equals the travel time t_{0i} plus the unloading time at the laboratory τ_0 and the time consumed from v_i back to the Lab, that is (T_{max} - f_i). If no route starts from i, i.e., $x_{0i} = 0$, d_i is set to zero owing to the objective function structure. Finally, constraints (15) and (16) state the domains of the decision variables, as they are reported in Table 1.

5. The iterative matheuristic

To address the interdependency challenges raised by the BSTP and the fact that no time windows can be used to simplify the problem, we conceived an iterative matheuristic that encompasses two steps. The first step uses a decomposition method, referred to as *DM*. Our *DM* is inspired by the rolling horizon procedure. It separates the BSTP into a sequence of linked subproblems. Then, each subproblem is solved using the formulation proposed in Section 4. Once the final subproblem is solved, we obtain a complete feasible solution to the BSTP. Since this solving procedure is an approximated one, the second step performs a local search. To this end, we use again the formulation proposed in the previous section, but this time combined with two fix-&-optimize (F-&-O) techniques that we propose. Two exploring strategies are alternated in this local search to form an efficient scheme: the *keep groups* strategy aims to form groups of visits that should be performed together (i.e. in the same route), while the *change arcs* strategy requires that, in the new solution, at least a certain number of arcs are taken from a subset of "promising arcs" which are kept in a dynamic memory as the search advances. After a certain number of iterations without improvements on the best solution found so far, the matheuristic executes a diversification and the two steps are repeated until a stop criterion is met or until a time limit is reached. Sections 5.1 and 5.2 present respectively *DM*, the method to efficiently generate solutions to the BSTP, and the proposed local search strategies. Section 5.3 presents the overall matheuristic and describes its diversification and parameter updating mechanisms.

5.1. A decomposition method to generate solutions to the BSTP

Rolling horizon techniques are commonly used to approach complex problems that can be decomposed over time. However, in our case, there is no clear indication on the adequate strategy to "split" the original problem, the number of subproblems to create, nor their length or size. Therefore, we propose a decomposition procedure *DM* that, instead of separating the planning horizon into subperiods, separates the visits to perform to create a sequence of linked subproblems that will be solved in an ordered manner. In our implementation, when solving a subproblem Q_i, a set of R visits is planned, while also considering the subproblems already solved. However, a given number of visits of the previous subproblems are "frozen" and become parameters (i.e., we set the variables associated to them to the values already found). Freezing is dosed by parameter $\alpha \in [0,1]$, so the number of frozen visits is $\lceil \alpha \times R \rceil$. Similarly, when solving a subproblem Q_i, it is also possible to consider a number of visits belonging to the next subproblem Q_{i+1}. This "looking ahead" strategy potentially improves the solutions produced by the solver, but increases the computational effort. Parameter β controls the extend of the look ahead, and the number of forthcoming visits to be considered is set by $\lceil \beta \times R \rceil$. The three parameters R, α, and β grant a great flexibility to the decomposition method *DM* as it allows to simultaneously plan the R visits, reevaluate the decisions made in previous subproblems, and anticipate the needs related to forthcoming visits.

To illustrate the procedure *DM*, let assume a BSTP problem with 14 visits to perform, as shown in Fig. 3, and let also assume that parameters R, α, and β are set to values 4, 0.75, and 0.25, respectively. This means that in each subproblem four visits are added. Three visits ($\lceil \alpha \times R \rceil$) from the previous subproblem are frozen and one ($\lceil \beta \times R \rceil$) will be included as the look ahead. To apply the procedure, we need to establish a list of the SCCs to visit, ordered by the time at which their visits are done. Since the moments at which SCCs are visited depend on the actual routes, we solve the BSTP formulation until a first integer solution is generated by the solver. However, as it will be seen later, any feasible solution can be used by the procedure DM. The time at which visits are performed constitutes the ordered list of visits, as illustrated in the upper part of Fig. 3, where the 14 visits are organized in six routes and where circles and squares represent visits to SCCs and to the Lab, respectively.

To form the first subproblem, denoted *Q1*, we consider the first R visits (visits a to d) and we add to them the next $R \times \beta = 1$ visit (visit e, colored in black). Then, subproblem *Q1* is solved, producing *Sol. Q1*. Notice that the values produced for the variables associated to the fifth visit (that does not belong to the first subproblem) will be reevaluated when solving the second subproblem.

To form the second subproblem *Q2*, we consider the first subproblem (visits a to d), we add the next R visits (visits e to h), and one more visit corresponding to the look ahead (visit i, in black). However, the first $R \times \alpha = 3$ visits (visits a, b and c, in white) are frozen or, in other words, the variables associated to them are set to the values already produced when solving *Q1*. Subproblem *Q2* is then solved to produce *Sol Q1+Q2*. The third subproblem *Q3* includes the visits in *Q1* and *Q2*, the next four visits (visits i, j, k, and l), and an additional visit (visit m). When solving *Q3*, variables associated to visits a to d, g and f are frozen (set to the values produced when solving the previous subproblems). In this manner, the algorithm progress until a subproblem containing all the visits, subproblem Q_4 in Fig. 3, is solved. The procedure stops and the solution becomes the initial solution for the next step of the matheuristic.

Freezing is easily handled in the mathematical formulation. Although we mentioned that we fixed the information concerning the visits to freeze, we relax part of this information (i.e., the time at which visits are performed represented by variables u_i) to focus exclusively on frozing the arcs forming the routes (variables x_{ij}). For instance, let us consider the example illustrated in Fig. 3. When solving subproblem *Q2*, information on visits a to d is available in the solution produced to subproblem *Q1*. To freeze visits a, b, and c when solving *Q2*, we use the following set of equations:

$$x_{ij}^{Q2} \geq x_{ij}^{Q1} \quad \forall i, j \in 0, a, b, c \tag{17}$$

Eq. (17) shows, for our given example, that the arcs between visits a, b, c, and the depot in the second subproblem Q_2 need to be the same as the ones from Q_1, hence the route structure for these visits is frozen in Q_2. Relaxing the visit time decisions u_i grants even more flexibility to the approach because it allows to reconsider the time at which visits were performed, without increasing much the difficulty of the problems to solve. The structure of the already formed routes is kept.

Coming back to the parameters of *DM*, the impact of R is easy to understand. As R increases towards $|P|$, the decomposition tends to the initial problem so one might expect the solver to produce better solutions. However, the computational effort to solve each subproblem Q increases quickly with R. Parameter α affects the size of the solution space explored at each iteration. Setting α to a small value fixes only the departure of the earliest routes and allows the new subproblem to reconsider the majority of the decisions, having then a larger subproblem each time. As α increases, more and more information is kept from the previous subproblems and, if α is set to 1, previous decisions will not be reevaluated. Parameter β doses, as mentioned before, the "looking ahead" strategy . The values of the parameters are adjusted from one iteration to the next to produce different sets of subproblems, as will be explained in Section 5.3. It is important to note that the value of parameters α and β are independent in the interval [0,1]. Furthermore, the method does not require solving each subproblem to optimality. Instead, the best solution found inside a 90-s time limit is reported for each subproblem, until a full feasible solution to the BSTP is obtained.

Finally, it is worth mentioning that the formulation proposed in Section 4 must be adapted to the case where it is applied to a subproblem in order to ensure the feasibility and continuity of the routes over the entire horizon. See Appendix B for a detailed description of the modified formulation.

5.2. A local search procedure based on fix-&-optimize strategies

Since the *DM* procedure provides an approximated solution to the BSTP, the second step of the matheuristic performs a local search on the solution produced by the first step. In this paper, we introduce a set of cuts based on a F-&-O technique to explore efficiently the neighborhood of the obtained solution. This approach takes advantage of what the MILP formulation can efficiently do: set the schedule of the visits. More precisely, we solve the BSTP using the formulation proposed in Section 4 but fixing a part of the formulation variables to the values they have in the solution produced by the *DM*. By doing so, a large number of variables become parameters and a part of the solution is "fixed" while the rest remain free and can be explored by the solver. This idea was implemented in two different manners or strategies referred to as *Keep groups* and *Change arcs*.

Given a current solution s, the *Keep groups* strategy imposes that at least a given number of the visits must be performed by the same route as in s, although the sequence of visits in the route and the times at which the visits are performed may change. Mathematically, this strategy can be expressed by constraints such as :

$$\sum_{v_i \in r} \sum_{v_j \in r\, v_j \neq v_i} x_{ij} \geq |r| - 1 \quad \forall\, route\; r \tag{18}$$

where r represents the route, v_i and $v_j \in r$ the nodes to be visited by route r, and $|r|$ the cardinality of the route (i.e., the number of visits in route r).

The *Change arcs* strategy is inspired by probabilistic approaches such as *granular tabu search*. It requires that, a certain number of arcs Δ taken from a subset of "promising arcs" $\bar{A}$ will be used in the new solution. Subset $\bar{A}$ is built and managed as the research advances, keeping in

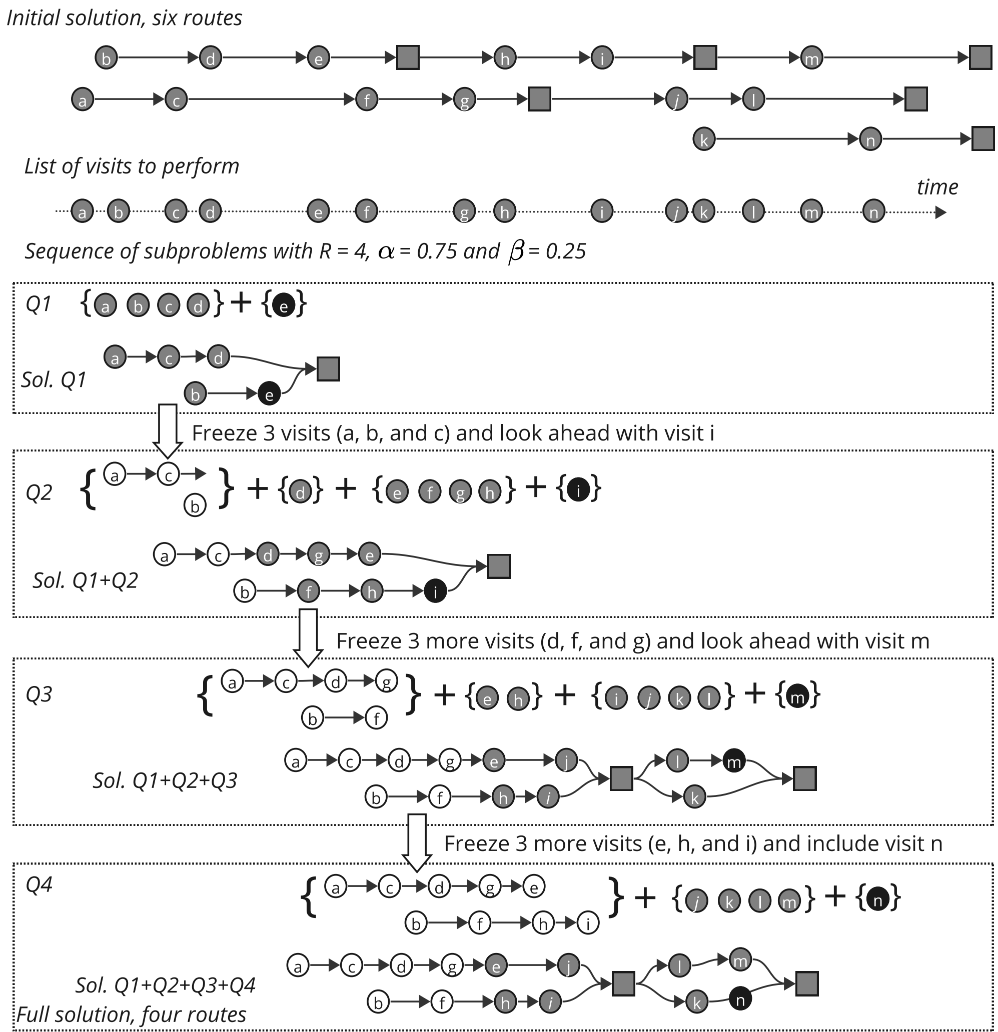

Fig. 3. Example of problem decomposition: an initial solution is used to generate a list of visits which are then progressively grouped to form a sequence of four subproblems using $R = 4$, $\alpha = 0.75$, and $\beta = 0.25$.

memory the number of times that each arc has been used to form past solutions. Parameter Δ is used in this strategy to control the solver's freedom choosing the arcs in the solution. We express such restrictions by constraints such as:

$$\sum_{(i,j)\in\bar{A}} (1 - x_{ij}) + \sum_{(i,j)\notin\bar{A}} x_{ij} \geq \Delta \tag{19}$$

where $\bar{A} = \left\{(i,j) : \bar{x}_{ij} = 1\right\}$ is the set of promising arcs.

The strategies described show two different levels of intensification, which are complementary. Indeed, while the *Keep groups* strategy seeks to group visits without regarding the structure of the routes, the *Change arcs* strategy intends to encourage the presence of parts of routes that were deemed appealing. We therefore decided to use them alternately, starting by *Keep groups* and using it until no improvement on the incumbent solution is reached. Then, *Change arcs* is used. Anytime an improvement is reached, the local search comes back to the *Keep groups* strategy. Algorithm 1 formalizes the use of the F-&-O strategies. In Algorithm 1, s_0 and s^* refer to the initial solution produced by the decomposition method DM, and the best solution found so far, respectively, while s' is used to record temporarily a solution. The modulus of a solution, for instance $|s^*|$, indicates the value of the objective function for s^*. We defined it as the counter of iterations performed without improvement, while it_{max} is the parameter limiting the allowed number of iterations without improvement.

It is worth noting the relevance of parameter it_{max}, which allows us to dose the manner in which the search effort is invested if we assume

Algorithm 1 Receives s_0, performs a Local search around it, and returns s^*

$s^* \leftarrow s_0$
$it \leftarrow 0$
while $it < it_{max}$ **do**
  $s' \leftarrow KeepGroups(s_0)$
  if $|s'| < |s^*|$ **then**
    $s^* \leftarrow s'$
    $it \leftarrow 0$
  else
    $it = it + 1$
    $s' \leftarrow ChangeArcs(s_0)$
    if $|s'| < |s^*|$ **then**
      $s^* \leftarrow s'$
      $it \leftarrow 0$
    end if
  end if
end while
return s^*

a restriction on the computational time allowed to the matheuristic. A large value of it_{max} allows for a longer local search (more iterations without improvement), and therefore a more thorough exploration of the neighborhood around each of the solutions produced by the DM method. Consequently, less initial solutions will be generated and explored within the same total time. On the other hand, if it_{max} is set to a small value, the local search will consume shorter time, allowing for more (different) initial solutions to be generated and explored. Section 5.3 discusses how this balance is set.

5.3. The complete matheuristic and the diversification mechanisms

Algorithm 2 formalizes the structure of the matheuristic. Using an initial solution s, it uses the decomposition method *DM*, described in 5.1 with initial parameters (R_0, α_0, β_0) to produce a solution s_0 on which the local search procedure is applied.

The two steps (i.e., generate an initial solution by *DM* and then applying the local search) are repeated for a number of iterations, which is controlled by counter It and limited to the value of parameter It_{max}. At each repetition, the current values of parameters $\{R, \alpha, \beta\}$ of *DM* are adjusted according to the following idea: if solving all the subproblems reaches a total time limit of 350 s, then R is reduced to $R-2$ visits while α and β remained unchanged. Otherwise, α is reduced to $\alpha = \alpha_0 - 0.1$ and β is increased to $\beta = \beta_0 + 0.1$, aiming to provide more freedom to variables in the problem and to look ahead farther, respectively.

Although the progressive update of parameters R, α, and β helps producing different initial solutions from one iteration to the next, the matheuristic needs to implement a set of mechanisms to force the algorithm to explore different regions of the solution space. Said mechanisms seek to improve the robustness of the matheuristic. In our case, we implemented two simple yet effective methods *DM1* and *DM2* to move the search to other regions of the solution space. The first method *DM1* is based on the analysis of a dynamic memory that tracks the previously explored solutions and identifies the times at which their routes end (arrival time at the lab). The most frequent end times are selected and used as cutting points to split the best solution found so far into subproblems instead of using parameter R. The initial values of parameters α_0 and β_0 are used to produce a new initial solution. The second mechanism, *DM2*, simply splits the best solution found so far into R_0 subproblems. Subproblems are formed in such a manner that all but the last subproblem have the same number of visits and α_0 and β_0 are used to produce a new initial solution. The described procedure is repeated until counter Id reaches the value of parameter Id_{max}, the matheuristic stopping criterion.

6. Numerical results

The aim of this section is twofold. First, it assesses the performance of the proposed matheuristic in terms of the quality of the solutions it produces and the required computational time. Second, it analyzes the results produced by the matheuristic to highlight the specific contribution and added value of the different mechanisms and algorithmic strategies.

6.1. Description of the test instances and algorithm's parameters

We first describe the characteristics of the 32 instances used in our numerical experiments. These instances correspond indeed to the real sample collection needs of different regions in the Province of Quebec, Canada, and offer a very rich and diverse testbed for our experiments, as explained in the following paragraphs. Instances were categorized into 17 *medium* and 15 *large* size instances, according to the cardinality of the set of collection centers to serve $|C|$ and the number of visits to perform $|P|$. Instances are also characterized by the synchronization percentage of their visits ($\%P_{sync}$) and the interdependency percentage ($\%SCC_{sync}$). Bredström and Rönnqvist (2008) defined $\%P_{sync}$ as the percentage of visits that required some sort of synchronization. In their testbed, $\%P_{sync}$ was always 10%, so in an instance with 80 visits to schedule, eight of them required some sort of synchronization. We computed for all our instances $\%P_{sync} = P_{sync}/|P|$. In the BSTP case, this synchronization is required between visits of the same SCC, causing interdependency in the routes and the SCCs' visits. Therefore, in addition to the synchronization percentage, we defined for each instance their *interdependency percentage* or $\%SCC_{sync}$ as the percentage of SCCs that required two or more visits in a single day ($\%SCC_{sync} = SCC_{sync}/|SCC|$). As an example, consider Instance 24, which has 17 SCCs ($|SCC| = 17$) and 29 visits ($|P| = 29$). Of these 29 visits, 21 require some sort of synchronization (i.e. $P_{sync} = 21$). Therefore, $\%P_{sync} = 21/29 = 72,4\%$. Moreover, from the 17 SCCs, nine SCCs require two or more visits, which means that $\%SCC_{sync} = 9/17 = 52.9\%$. Instance 4 has a $\%SCC_{sync} = \%P_{sync} = 100\%$, because all of its eight SCCs require two or more visits in the planning horizon, then the 24 visits required synchronization.

Algorithm 2 Matheuristic

Generate feasible solution s
$Id \leftarrow 0$
$R \leftarrow R_0,\ \alpha \leftarrow \alpha_0,\ \beta \leftarrow \beta_0$
while $Id < Id_{max}$ **do**
  $It \leftarrow 0$
  while $It < It_{max}$ **do**
    $s_0 \leftarrow DM(s, R, \alpha, \beta)$ ▷ First step: generate solution s_0
    $s_1 \leftarrow LocalSearch(s_0)$ ▷ Second step: local search on s_0, Algo. 1
    if $|s_1| < |s|$ **then**
      $s \leftarrow s_1$
    end if
    $Update(R, \alpha, \beta)$ ▷ Update according to Section 5.3
    $I_t \leftarrow I_t + 1$
  end while
  $s_0 \leftarrow DM1(s, \alpha_0, \beta_0)$ ▷ Perform diversification DM1
  $s_1 \leftarrow LocalSearch(s_0)$
  if $|s_1| < |s|$ **then**
    $s \leftarrow s_1$
  else
    $s_0 \leftarrow DM2(s, \alpha_0, \beta_0)$ ▷ Perform diversification DM2
    $s_1 \leftarrow LocalSearch(s_0)$
    if $|s_1| < |s|$ **then**
      $s \leftarrow s_1$
    end if
  end if
  $Id \leftarrow Id + 1$
end while

In the group of *medium* instances, $|SCC|$ ranges from four to 24, and the number of visits $|P|$ varies from 11 to 28. Medium instances present a synchronization percentage $\%P_{sync}$ between 27% and 100%, an interdependency percentage $\%SCC_{sync}$ between 16% and 100%, and ten instances have interdependency above 50%. In the group of *large* instances, $|SCC|$ ranged from 11 to 50, and the number of visits $|P|$ varied between 29 and 74, while $\%P_{sync}$ ranges from 25% to 94%, $\%SCC_{sync}$ between 14% and 88%, and there are nine instances with interdependency above 50%.

Moreover, the demographic and topological aspects are as relevant as the size of the instance to understand the richness and diversity of the testbed. Indeed, some instances consider vast territories with a low population density and a rather light road network. We refer to the 17 instances matching this description as "rural" (11 medium and six large instances). The 15 remaining instances are much denser; thus, the distances between SCCs are shorter, and the number of arcs connecting them is much higher. We refer to these as "urban" instances (six medium and nine large instances).

The matheuristic has parameters that must be set adequately. To this end, preliminary tests were conducted to help us select appropriate values. The tests consisted in running the algorithm using several combinations of parameters, seeking to balance the computational effort and the quality of solutions. According to these preliminary tests, we fixed $it_{max} = 2$, $It_{max} = 5$, and $Id_{max} = 7$, to ensure that various decompositions were considered for each instance, while also being able to provide good quality solutions in less than an hour. We also limited the computational time required to solve each subproblem in the first step of the matheuristic to 90 s. With this time limit, almost all subproblems were solved to optimality, although the method does not require it. The preliminary experiments also allowed us to set the initial values for the parameters of matheuristic first step to (R_0, α_0, β_0) = (7, 0.5, 0.5) and the time limit to adapt the decomposition parameters to 350 s total. Finally, we set parameter Δ, which defines the number of arcs to be selected from the promising arc set $\bar{A}$ to $\Delta = \lceil 0.8 \times |\bar{A}| \rceil$. Moreover, the value of M in the model has been bounded to the length of a day in minutes (1440 min).

All tests were executed on a multi-user server with 64 GB of RAM and am Intel(R) Xeon(R) Gold 6130 CPU @ 2.10 GHz, with 8 GB of RAM. The MILP formulation was solved using CPLEX 12.8.

6.2. Numerical results

To assess the performance of the proposed matheuristic, we compare the results it produced to the ones reached by mathematical formulation presented in Section 4 with a computational time limit of 36,000 s (10 h).

The results are reported in Table 2. The leftmost part lists the instances and describes their main characteristics, starting with the number of collection centers $|SCC|$, the number of visits to perform $|P|$, the synchronization percentage $\%P_{sync}$, and the interdependency percentage $\%SCC_{sync}$. Finally, under header R/U, we show the type of instance with respect to its topology (rural = R or urban = U). The results produced by the solver to the mathematical formulation of Section 4 are reported in the columns under header $MILP$, including the value of the objective function (column OF) of the best solution found by the solver in the time/memory limit, the computational time in seconds (column $Sec.$), and optimality gap reported by the solver ($\%GAP$). Note that in several cases, the optimality gap was not closed before exhausting the allotted computational time (36,000 s). In other cases, however, the computer memory limit (8 GB) was reached, stopping the search. An asterisk ($*$) in the computational time column identifies such cases for which the search was aborted. The last three columns in Table 2 reports the objective value of the best solution produced by the matheuristic (OF_{Math}); then, $\%BKS$ reports the difference in percentage between the solution produced by the proposed method (OF_{Math}) and the best-known solution produced by the solver (OF), so that $\%BKS = (OF_{Math} - OF)/OF$. Notice that negative values of $\%BKS$ indicate that the matheuristic produced a better solution than MILP inside the time limit of our experiment. Finally, we report the CPU time (in seconds) it took to produce a solution ($Sec.$).

Let us first look at the results produced by the MILP for the *medium* size instances (instances 1 to 17). Table 2 confirms the difficulty of solving this problem. Indeed, CPLEX was able prove optimality only for instances 2, 3, and 13, and, in the case of instance 13, doing such required more than 26, 000 s. For the five cases in which the allotted computational time was exhausted, the optimality gaps were greater than 40%. Finally, in nine cases, the search was aborted after the computer's memory limit was reached.

The matheuristic was able to produce solutions with the same objective value as the MILP in ten of 17 instances, including the three instances having proven optimal solutions. Moreover, it improved the MILP's best solution in five of 17 instances, although the MILP performed better in two cases. Regarding the computational time, most instances required between 1500 and 3300 s, with an average computational time of 2167 s, confirming that the matheuristic was able to reach solutions as good as CPLEX in less than 40 min, a fraction of the time required by the solver.

If we examine the results produced for *large* size instances (instances 18 to 32), CPLEX produced solutions that, in the best case, showed an optimality gap of 71%. In 11 cases, the search tree exhausted the memory of the available computer, aborting the search. This computational limit confirms the need for efficient solution methods to get good quality solutions. Even if the BSTP does not need to be solved daily, managers still need to perform new experiments regularly to cope with events or new situations. Filling this gap, the matheuristic improved the results of the MILP in eight of 15 cases and produced the same objective value in two more cases, whereas the MILP was better in five cases. The matheuristic produced an average improvement of 0.43% over the *large* size instances, which is encouraging when considering the computational time required to reach these solutions. Indeed, the computational times required by the matheuristic to solve *large* instances remained within the same order of magnitude as those for *medium* instances, ranging from 2768 to 4750 s (46 to 80 min). This confirms the good scalability of the matheuristic and its potential for efficiently handling even larger instances. To summarize, the proposed matheuristic is able to produce good quality solutions in much shorter time than the solver, providing an efficient tool for managers to test different scenarios when creating their tactical transportation plans.

To analyze further the performance of our matheuristic, in the rural instances, with large distances between nodes and less potential of routing, the matheuristic tends to perform well, providing as good solutions as the solver in less time in 13 of 17 rural instances, and in average to less than 2% of the four instances that are not improved. For urban cases, where longer routes are possible with a strong combinatorial challenge, the matheuristic is also a great tool, as it achieves or improved the solution found by the solver in 12 out of 15 urban instances, with an improvement of over 2%, showing how the decomposition and F-&-O techniques successfully create good quality routes.

6.3. Assessing the contribution of the matheuristic components to its performance

The previous experiments demonstrated the effectiveness and the efficiency of the proposed matheuristic. This section analyzes how the different mechanisms and search strategies that form the matheuristic contribute to its performance. To this end, we solved the BSTP instances using the two main components of the matheuristic (i.e., the *DM*, and the local search) independently. In this section, we will first compare and discuss these results. Then, we will look in detail at the contribution of the two F-&-O strategies. Finally, we will assess the ability of the diversification mechanisms to explore more effectively the solution space.

To assess the contribution of the matheuristic components to its performance, we proceed in the following manner. First, we executed the matheuristic without step 2 (the local search), limiting the total running time to 10 h. Then, we run only the local search using the first integer solution produced by the solver to the mathematical formulation presented in Section 4 as the initial solution. A time limit of 10 h was also set for each experiment. Table 3 reports the results produced by these experiments. Header Features presents again the main characteristics of each instance and Header Matheuristic recalls the results of the algorithm in $\%BKS$ and $Sec.$, as in the previous table. Finally, headers *DM* and *LS* present the $\%BKS_{xx}$ and $Sec.$ obtained when using only the decomposition method, or only the local search, respectively. In all results, $\%BKS$ is computed with respect to the best solution found by CPLEX after 10 h.

Table 3 shows that, for medium instances of type *rural*, both components achieve, in general, results as good as the whole matheuristic. Only in two cases (instances four and seven) the components were able to improve slightly the best solution found by the complete matheuristic. When we consider the *urban* instances, *DM* reached again a very good performance producing solutions as good as the matheuristic. For the worst case (instance 15) the difference between the $\%BKS$ produced by the matheuristic and *DM* with respect to the one produced by

Table 2
Numerical results produced by the mathematical formulation and the matheuristic for the *medium* and *large* instances.

Inst.	Features					MILP			Matheuristic		
	$\|C\|$	$\|P\|$	$\%SCC_{sync}$	$\%P_{sync}$	R/U	OF	$Sec.$	$\%GAP$	OF_{Math}	$\%BKS$	$Sec.$
1	4	17	100%	100%	R	1870	*	100%	1869	−0.05%	1535
2	6	11	67%	82%	R	502	5	**0%**	502	**0.00%**	59
3	7	14	57%	79%	R	1137	467	**0%**	1137	**0.00%**	279
4	8	24	100%	100%	R	2606	*	100%	2619	0.49%	2632
5	9	16	44%	69%	R	1252	36 000	40%	1252	0.00%	981
6	9	26	67%	88%	R	1847	36 000	100%	1847	0.00%	2901
7	12	18	25%	50%	R	1407	*	77%	1407	0.00%	1973
8	17	24	24%	46%	R	2009	*	100%	2006	−0.15%	2906
9	19	22	16%	27%	R	1581	*	90%	1581	0.00%	2961
10	23	27	17%	30%	R	2020	*	89%	2020	0.00%	2838
11	24	28	17%	29%	R	2019	*	91%	2019	0.00%	2044
12	8	15	63%	80%	U	516	36 000	49%	516	0.00%	1602
13	9	14	33%	57%	U	439	26 112	**0%**	439	**0.00%**	1570
14	10	24	90%	96%	U	817	36 000	100%	799	−2.20%	3195
15	11	25	82%	92%	U	811	*	100%	828	2.10%	2875
16	12	26	75%	88%	U	913	*	100%	907	−0.66%	3297
17	13	28	77%	89%	U	952	36 000	100%	926	−2.71%	3188
						1335		73%	**1334**	**−0.19%**	**2167**
18	11	29	64%	86%	R	2490	*	100%	2487	−0.12%	2768
19	14	36	86%	94%	R	2453	*	100%	2389	−2.60%	3751
20	26	31	19%	32%	R	2193	*	93%	2193	0.00%	3292
21	40	63	53%	70%	R	3987	36 000	100%	4179	4.82%	4126
22	46	70	48%	66%	R	5017	36 000	100%	5057	0.79%	4750
23	50	74	44%	62%	R	5289	*	100%	5378	1.68%	4002
24	17	29	53%	72%	U	1905	*	89%	1781	−6.51%	2900
25	17	33	88%	94%	U	924	*	100%	913	−1.19%	3301
26	18	35	61%	80%	U	1860	*	100%	1863	0.16%	3708
27	19	33	58%	76%	U	1898	*	100%	1895	−0.16%	3646
28	19	33	58%	76%	U	1898	*	90%	1882	−0.84%	3500
29	19	35	63%	80%	U	2208	36 000	94%	2166	−1.90%	3422
30	27	32	19%	31%	U	3095	*	71%	3069	−0.85%	3095
31	28	33	18%	30%	U	3131	*	76%	3131	0.00%	3332
32	35	40	14%	25%	U	3746	36 000	87%	3756	0.27%	3960
						2806		93%	**2809**	**−0.43%**	**3570**

the solver after 10 h were of 2.10% and 2.47%, respectively. However, the local search alone produced solutions up to 10.72% and 14.54% worse than the matheuristic (instances 15 and 17), demonstrating that the initial solution produced by the MILP was not a good one thus the need for a mechanism to generate several initial solutions. The results produced for the large instances point on the same direction. Although in three cases over 15 large instances the *DM* produced slightly better results that the matheuristic, on average, *DM* produced solutions that were 1.03% worse than CPLEX and 1.46% worse than the matheuristic. As per the local search, its dependence with respect to the quality of the initial solution becomes even clearer. Indeed, the results produced by the local search procedure were on average 12.64% worse than the matheuristic. We understand that the LS performs poorly, as the interdependency of the visits are difficult to solve with classic VRP neighborhoods. We conclude then that the proposed hybrid matheuristic balances adequately the intensification/diversification paradigm at the base of approximated methods. On the one hand, the solutions produced by the rolling horizon inspired *DM* show that the decomposition is effective and contributes to the robustness, and, on the other hand, local search allows the method to improve the average quality of the initial solutions in most of the cases.

Although the previous analysis demonstrated the efficiency of the *DM*, it is noteworthy understanding the extent to which its performance lays on the progressive adaptation of the method's parameters (R, α, and β) or the use of the diversification mechanisms *DM1* and *DM2*. To this end, we identified, for each of the 32 instances, the phase of the search during which the best solution was reached and, particularly if it happened exploring an initial solution produced by the first configuration (R_0, α_0, and β_0), the adjusted values of *DM* or the diversification mechanisms *DM1* and *DM2*. The results are summarized in Fig. 4 reports the number of times each decomposition method led, after applying the local search step, to the best solution for rural and urban instances.

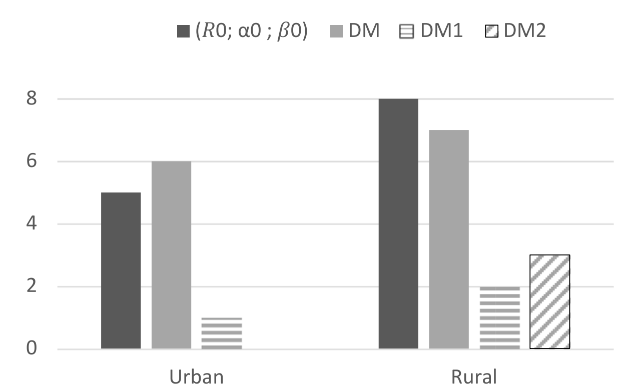

Fig. 4. Number of times each decomposition method led to the best solution for rural and urban instances.

The initial solution produced with parameters ($R = 7, \alpha = 0.5, \beta = 0.5$) was at the base of 13 best solutions, confirming that the empirically chosen values were quite effective. Note that, once the values of the parameters have been adjusted, *DM* produced solutions that led to 13 more best solutions. Finally, the last two splitting approaches, *DM1* and *DM2*, contributed to six best solutions, although *DM2* did not produce any best solution for urban instances. We conclude that all the strategies used contributed to the effectiveness of the matheuristic, and that their different principles enhance its robustness.

Table 3
Numerical results produced by the components of the matheuristic (DM = decomposition approach, LS = Local search) for the medium and large sized instances.

	Features			Matheuristic		DM		LS	
Ins	$\|SCC\|$	$\|P\|$	R/U	$\%BKS$	$Sec.$	$\%BKS_{DM}$	$Sec.$	$\%BKS_{LS}$	$Sec.$
1	4	17	R	−0.05%	1535	−0.05%	824	−0.05%	18 105
2	6	11	R	0.00%	59	0.00%	28	0.00%	6
3	7	14	R	0.00%	279	0.00%	132	0.00%	2 059
4	8	24	R	0.49%	2632	−0.26%	1456	−0.26%	2 665
5	9	16	R	0.00%	981	0.00%	530	0.00%	10 811
6	9	26	R	0.00%	2901	0.00%	1377	0.05%	22 132
7	12	18	R	0.00%	1973	0.00%	983	0.00%	10 812
8	17	24	R	−0.15%	2906	−0.15%	1459	−0.20%	4 858
9	19	22	R	0.00%	2961	0.00%	1501	0.00%	3 500
10	23	27	R	0.00%	2838	0.00%	1422	0.00%	1 939
11	24	28	R	0.00%	2044	0.00%	1289	0.00%	1 669
12	8	15	U	0.00%	1602	0.00%	1042	0.19%	10 808
13	9	14	U	0.00%	1570	0.00%	864	0.00%	10 809
14	10	24	U	−2.20%	3195	−1.48%	1544	−0.49%	13 160
15	11	25	U	2.10%	2875	2.47%	1557	12.82%	*
16	12	26	U	−0.66%	3297	−0.33%	1466	−0.22%	2 836
17	13	28	U	−2.71%	3188	2.20%	1731	16.52%	*
				−0.19%	2167	0.14%	1130	1.67%	7 745
18	11	29	R	−0.12%	2768	−0.12%	1420	0.72%	8 501
19	14	36	R	−2.60%	3751	−3.13%	2207	20.24%	*
20	26	31	R	0.00%	3292	0.00%	1727	0.00%	5 164
21	40	63	R	4.82%	4126	11.11%	2780	62.26%	*
22	46	70	R	0.79%	4750	9.10%	2573	37.44%	*
23	50	74	R	1.68%	4002	7.32%	2245	48.87%	*
24	17	29	U	−6.51%	2900	−5.83%	1817	−2.78%	470
25	17	33	U	−1.19%	3301	0.65%	1455	17.42%	*
26	18	35	U	0.16%	3708	0.22%	2134	0.75%	16 464
27	19	33	U	−0.16%	3646	−0.63%	1918	0.26%	11 998
28	19	33	U	−0.84%	3500	−0.32%	2209	−0.21%	6 338
29	19	35	U	−1.90%	3422	−2.49%	2210	−1.81%	11 786
30	27	32	U	−0.85%	3095	−0.85%	1993	−0.85%	1 401
31	28	33	U	0.00%	3332	0.13%	1891	0.00%	1 755
32	35	40	U	0.27%	3960	0.27%	2011	0.80%	10 825
				−0.43%	3570	1.03%	2039	12.21%	7 470

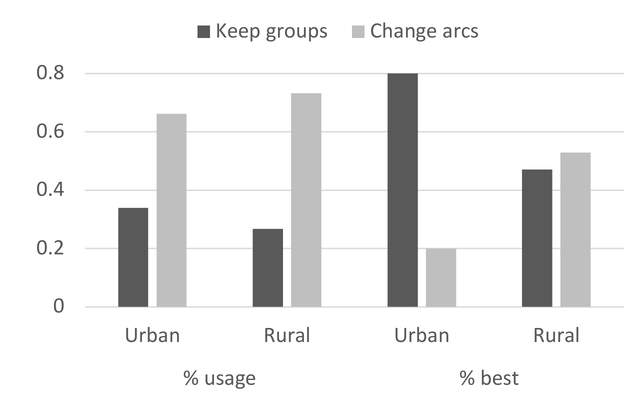

Fig. 5. Percentage of the time the local search uses each F-&-O strategy, and percentage of times each F-&-O strategy led to the best solution (separated by rural and urban instances).

Let us now look at the efficiency of the two F-&-O strategies that form the local search stage: the *Keep Groups* and the *Change arcs*. To assess the extent to which these strategies work in an intertwined manner, the left part of Fig. 5 shows the usage of the two strategies, in percentage of time with respect to the local search total time, whereas the right part shows the number of times over 32 instances that the best solution was produced using each strategy. Because the considered instances show distinct topological natures (i.e., rural vs. urban), our analysis explicitly considered this.

The left part of Fig. 5 (over title %usage) shows that although the *Change arcs* strategy is used more frequently than the *Keep groups* strategy, the latter is still used more than 1/3 of the times. Evidently, the use of these strategies depends on the type of instance. Denser urban instances perform fewer *Change arcs* iterations (approximately 66% of the total) than rural instances (74% of the total iterations). This suggests that the dynamic mechanism allowing to switch between them works adequately. Moreover, the right part of Fig. 5 (over title %best) reports the portion of times that the matheuristic produced its best solution during an iteration using the *Keep groups* or the *Change arcs* strategy. In the case of urban instances, the best solution was produced in 80% of the cases using the *Keep groups* strategy, and this percentage reduces to only 47% when solving rural instances. This makes sense; as the routes produced for urban instances have more visits, *Keep groups* forces the solver to focus on the route configuration to determine the best timing for the visits. The complementarity of the proposed strategies and their contribution to the effectiveness of the matheuristic are confirmed.

Finally, we inquired about the time at which the best solutions were found. In an attempt to demonstrate the value of the time invested and the strategies used during the search, we normalized the computational time across the instances to present in Fig. 6 the number of times that the best solution was reached during the first 20% of the total computational time, between the 20% and the 40% of the total computational time, and so on.

The largest number of best solutions (20) was produced during the first part of the search within 40% of the time, which is consistent with the success of the initial splitting strategy, as previously discussed.

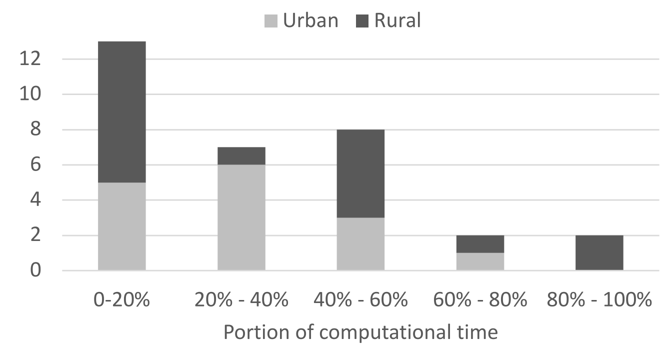

Fig. 6. Number of best solutions produced with respect to the normalized computational time.

Nevertheless, up to 10 best solutions were reached between the 40% to 80% of the total computation time. Unsurprisingly, only two best solutions were obtained during the final part of the experiments.

To summarize, the matheuristic proposes an adequate combination of strategies and mechanisms that allow it to effectively tackle the difficulties and challenges raised by the BSTP, as the decomposition is not straightforward. In particular, the iterative approach that builds different initial solutions produced encouraging results and reveals as a promising approach for addressing routing problems with dependency or synchronization constraints between routes. Nonetheless, the local search mechanism brings added value to the method and complements the strengths of the rolling horizon inspired heuristic. Our detailed analysis demonstrates the contribution of each aspect of the matheuristic and its capacity to solve interdependent routing problems efficiently.

7. Conclusions

This paper presents an efficient iterative matheuristic for solving a complex routing problem arising during the transport of products with short lifespans, which results in visits to customers that are interdependent. The problem is inspired by the biomedical sample transportation problem (BSTP) faced by managers of the healthcare system in the province of Quebec, Canada. The problem studied relates to VRPTW-Syn and blood transportation problems defined in the literature, as it shows a strong interdependence in the decisions concerning visits to customers and route schedules. This paper builds on previous formulations for the BSTP, but proposing a new and more realistic approach to sample lifespan. This allows to integrate and optimize decisions on samples accumulation in SCCs and routing. Until now, these two decisions were set as parameters according to managers' expertise, which produced a suboptimal approach. We now offer managers an integrated decision tool for deciding opening times, pickup times, and routing times. This tool allows managers to better exploit the reality of each region when planning sample collection processes.

In addition to the extension in the modeling approach, this paper proposes an efficient and novel matheuristic to tackle the complex MILP formulation and shortcomings of commercial solvers. The efficient matheuristic combines a decomposition approach and a local search stage that uses two F-&-O techniques. The iterative matheuristic reduces interdependency by applying different strategies to divide the problem into smaller subproblems with different visit structures. Moreover, the local search is performed on each solution produced by the decomposition method by means of two F-&-O techniques, increasing the efficiency of the algorithm. We tested the matheuristic on a set of rich and diverse real-life instances with up to 50 customers, 74 visits, and a visit synchronization average of 68%, which is much larger than the synchronization percentage requested in previous studies. For these instances, the matheuristic provided good quality solutions, as good or better than CPLEX, but in only a fraction of the time, making it suitable for implementation in a decision support tool for planners in healthcare logistics.

Besides its contribution to a real and practical problem, this paper makes a significant contribution to the understanding and resolution capacity of routing problems with high interdependency. We proposed and tested several mechanisms to decompose the problem, as well as cuts to explore the solution space in an efficient and fast way, proving the interest of having decomposition approaches interacting with MILPs solvers.

From a practical perspective, the reduction in the computational times achieved by the proposed algorithm with respect to the mathematical formulation allows managers to run a variety of what-if studies that open new perspectives. First of all, the heterogeneity among the regions made that the one-size-fits-all approach had to be discarded. New simulations with structures having two or even three labs have been tested and their performance compared. The ability to solve even the largest instances in a reasonable time has allowed to produce more robust solutions. Each instance we considered corresponds to an "average" day for each region, but some collection centers are closed some days, and in other cases, their service times are different. We can now compute solutions for variants of the "average instances". Furthermore, although the current computational times (under one hour) do not allow to elaborate the routes in "real time", managers can compare solution scenarios in the case of special events such as, for instance, a snowstorm that forces some centers to remain close and reduces drastically the speed on the routes.

Despite the contributions achieved, our research still has limitations. From a scientific perspective, we are now considering the use of heuristic approaches for solving the subproblems generated by the decomposition, seeking to improve the effectiveness of the method. Moreover, we explore how to incorporate more randomness to the algorithm to better escape local optima and enhance robustness. Finally, from a practical standpoint, we believe that further research should be devoted to including laboratory operations in the design of the collection routes. We assume that the laboratory is able to handle any number of samples arriving at any rate. However, in practice, the first steps (opening the boxes, identifying the samples and preparing the samples for analysis) can be very time consuming and might provoke queues, exceeding in some cases the lifetime of samples that must wait too long for treatment at the lab. A promising avenue should be to consider the average rate of samples production at each collection sample to limit somehow the number of samples arriving with each vehicle to the lab.

CRediT authorship contribution statement

Chiara Mazzanti: Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Conceptualization. **Ana María Anaya-Arenas:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Valérie Bélanger:** Validation, Supervision, Funding acquisition, Formal analysis, Conceptualization. **Ettore Lanzarone:** Validation, Supervision, Software, Resources, Methodology, Investigation, Conceptualization. **Angel Ruiz:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

The data that has been used is confidential.

Acknowledgments

The authors thank the Natural Sciences and Engineering Research Council (NSERC) of Canada [RGPIN-2018-04609 and RGPIN-2020-07055] and the *Fonds de Recherche du Quebec en Nature et Techonologies* [2021-NC-282544 and 2020-NC-271332] for their financial support.

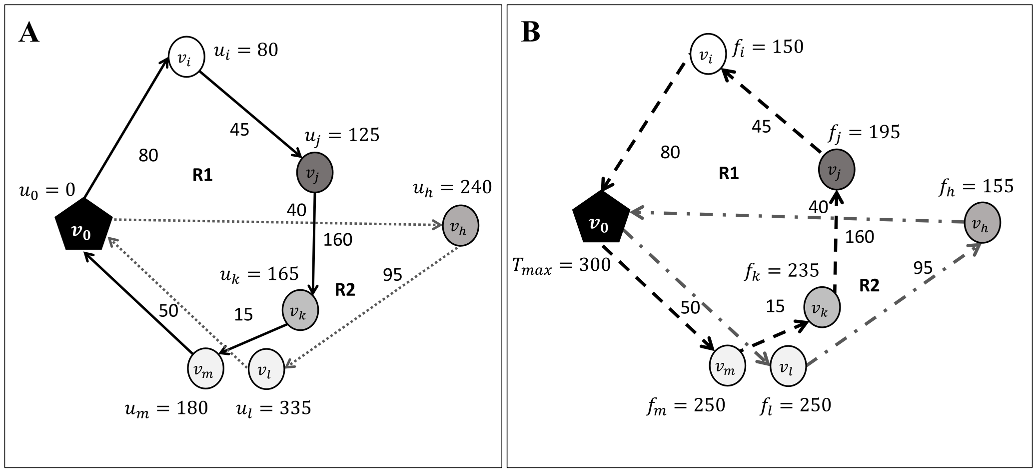

Fig. A.7. Routing time calculation (left) and remaining time consumption (right) for routes R_1 and R_2.

Appendix A. Time resource constraints

Section 4 presents the formulation for the BSTP explored in this paper. This appendix goes into more detail on how the lifespan of the sample is controlled in our formulation. As T_{max} is the maximum value over all SCCs, it points to the less restrictive SCC, which samples have the longest lifespan. Then, a given SCC c_g with a T^g_{max} lower T_{max} (a more restrictive SCC) will have a larger δ_g, as $\delta_g = T_{max} - T^g_{max}$. Then, samples' lifespan is warranted by the introduced constraints (12)–(13) using T^g_{max}, T_{max}, δ_g and variables f_i. As the samples' lifespan is computed from the moment samples are drawn from the patients until it arrives to the lab, the restriction must cover accumulation and transportation time. In other words, the sum of the time between two visits (the accumulation time) at SCC c_g plus the time needed to pick up the samples (i.e., to make a visit) and take said samples to the lab (in a constructed route) has to be less than the lifespan of the sample. This means that using a constraint to limit total route duration to the lifespan of the samples, as is usually done in time restricted-VRPs, will be incorrect.

This is the motivation behind variables f_i and d_i, and parameters δ_g and T_{max}. As no explicit time windows are set, and no vehicle or route index is included in the model, the use of variables **f**, **d** and **u** allows us to calculate the time of the route, and the resource constraint for all the routes created.

We will now illustrate the use of variables f_i and parameters T_{max} and δ_i using a numerical example. Fig. A.7, shows a solution for a small instance to perform six visits ($P = \{v_h, v_i, v_j, v_k, v_l, v_m\}$) and to take samples to the Lab (noted as v_0). Hence, $V = \{v_0, v_h, v_i, v_j, v_k, v_l, v_m\}$. To simplify the example, we consider only five SCCs ($C = \{c_1, \dots, c_5\}$), where only c_5 requires more than one visit during the planning horizon (only two visits require synchronization). We consider that $\{v_h, \dots, v_k\}$ are visits requested by SCCs $\{c_1, \dots, c_4\}$ and visits $\{v_l, v_m\}$ are requested by SCC c_5.

The left side of Fig. A.7 shows how the time calculation is done with the use of variables $\boldsymbol{u}$, and this for two routes R_1 and R_2. Assuming the route R_1 starts at the Lab at time $u_0 = 0$, then the visit v_i is done at time 80 ($u_i = 80$). If no service or waiting time is necessary, then $u_j = 125$, $u_k = 165$ and finally $u_m = 180$. This means that the vehicle will return to the Lab by time 230 ($u_m + t_{m0} = 180 + 50 = 230$). Likewise, R_2 reaches its first node v_h at time 240 ($u_h = 240$), then visit v_l is done at time 335 ($u_l = 335$) and the vehicle gets to the Lab by time 385. To arrive to node v_h at 240, with $t_{0h} = 160$, the R_2 must start at minute 80 and it last 305 min in total ($305 = 160 + 95 + 50$). Here, Constraints (6) in the model is still respected, as $u_0 = 0$, then $u_h \geq u_0 + \tau_0 + t_{0h} - M(1 - x_{0h})$. On the other hand, the variables $\boldsymbol{f}$ will calculate the consumption of remaining time, as it is shown in the right side of Fig. A.7. As explained before and in Section 4, the remaining time resource is consumed in the opposite direction to the route. The maximum remaining time is at the Lab, starting with $T_{max} = 300$ min, $f_m = 300 - 50 = 250$, $f_k = 235$ $f_j = 195$ and $f_i = 150$. The duration of the route is R_1 still 230 min. This can be calculated by the routing time (left side of Fig. A.7) or by Constraints (10), (11) and (14) as $d_i \geq T_{max} - f_i + t_{0i} + \tau_i = 300 - 150 + 80 + 0 = 230$. Likewise, in route R_2 with $T_{max} = 300$ min, then $f_l = 250$ and $f_h = 155$. Then, as v_h is the first node in R_2, then $d_h \geq T_{max} - f_h + t_{0h} + \tau_h = 300 - 155 + 160 + 0 = 305$.

Moreover, these constraints, interacting with constraints (12) and (13), force the samples' lifespan to be respected. In the same example, consider that v_m is the first visit of SCC c_5. Then, if $T_{max} = 300$ min, and SCC c_5 has lifespan limit of only 220 min ($T^g_{max} = 220$), then $\delta_5 = 300 - 220 = 80$. This solution is only feasible if the time elapsed between the opening of c_5 and its first visit u_m is less than 170 min (i.e., $u_m - a_5 \leq 170$), as the accumulation time at c_5 plus transportation time to the Lab has to respect samples lifespan of 220 min. Applying constraint (12) to v_m, and assuming opening at time 10 ($a_1 = 10$), one can find $f_m - (u_m - a_5) \geq \delta_5$ as $250 - (180 - 10) \geq 80$. Finally, consider that v_l is the second visit of c_5. Then again, constraints (12)–(13) are respected, as $f_l - (u_l - u_m) \geq \delta_5$ is $250 - (335 - 180) \geq 80$. This is equivalent to calculate the accumulation time at c_5 of 155 min ($335 - 180 = 155$) and travel time from c_5 to the lab of 50 min (or $300 - 250 = 50$). Hence, lifespan is always respected, even if the routes last more than $T_{max} = 300$ min as it is the case of the route R_2 that has a total driving time of 305 min.

Appendix B. Linking the subproblems

The interdependency between routes makes it inappropriate to solve a part of the problem without considering those previous and forthcoming. Moreover, in a given decomposition, a route may span two or more subproblems. Therefore, when solving a subproblem, extending the original formulation proposed in Section 4 is necessary to ensure the feasibility and continuity of the routes over the entire horizon.

Let us know look at how the formulation given in Section 4 needs to be adapted to handle subproblems rather than the complete BSTP. Using the same notation as in Section 4, a formalization of a subproblem is first given before the mathematical formulation.

Formalization of subproblems

As anticipated, when excluding a part of the instance, not all requests of the original problem are contained. To address this, we can redefine all sets described in Section 4, those marked with a tilde, to state that they are related to this subproblem.

In particular,

$$\tilde{C} = \{c_1, c_2, \dots, c_{\tilde{n}}\} \tag{B.1}$$

is the set of SCCs with at least one request included in the subproblem.

Moreover,

$$\tilde{P}_g \subset P_g \tag{B.2}$$

is the set of requests for SCC g that are part of the subproblem.

Given this, it suffices to define

$$\tilde{P} = \bigcup_g \tilde{P}_g \tag{B.3}$$

to obtain the set of all requests related to the considered Lab.

Following the same reasoning, it is possible to define:

$$\tilde{V} = \{v_0, v_1, \dots, v_{|\tilde{P}|}\} \tag{B.4}$$

$$\tilde{A} = \left\{(v_i, v_j) : v_i, v_j \in V,\ i \neq j,\ i, j \in \{0, \dots, |\tilde{P}|\}\right\} \tag{B.5}$$

$$\tilde{I}_g = \left\{1 + \sum_{h=0}^{g-1} |\tilde{P}_h|,\ 2 + \sum_{h=0}^{g-1} |\tilde{P}_h|, \dots, |\tilde{P}_g| + \sum_{h=0}^{g-1} |\tilde{P}_h|\right\} \tag{B.6}$$

where $|\tilde{P}_0|$ is set to 0. Their meaning is absolutely analogous to the that for the entire problem, as presented in 4.

Mixed integer programming formulation of subproblems

Once the new sets and parameters are defined, the considered subproblem can be formulated. The formulation uses the same variables presented in Table 1 and inherits Constraints (2)–(8) and (10)–(16), with the only difference being that for each set or parameter, its analogy with the tilde must be used.

Also concerning the objective function, it is totally analogous to (1).

$$\min \sum_{i=0}^{|\tilde{P}|} d_i \tag{B.7}$$

In terms of constraints, in addition to the inherited constraints, these two must be added:

$$b_g \leq u_k \leq b_g + \varphi_g; \quad g = 1, \dots, \tilde{n};\ k = (\tilde{I}_g)_{|\tilde{P}_g|} \text{ only if } |\tilde{P}_g| = |P_g| \tag{B.8}$$

$$u_k \geq b_g - (|P_g| - |\tilde{P}_g|)\hat{T}_g; \quad g = 1, \dots, \tilde{n};\ k = (\tilde{I}_g)_{|\tilde{P}_g|} \tag{B.9}$$

In particular, Constraint (B.8) substitutes (9) of the standard formulation. It states that in every SCC g, the last pickup is performed after the closing hour, b_g, within φ_g units of time. Evidently, when dealing with a subproblem, this must be stated only if in the considered subproblem the "real" last pickup of the SCC is included.

Constraint (B.9) is not included in the original formulation. It is added to ensure that all solutions of the subproblem present partial, feasible solutions for the original subproblem. Therefore, this constraint excludes the feasible region of the subproblem all solutions that are impossible to complete to obtain a full feasible solution. In particular, it is required that, for each center c_g, the last pickup included in the subproblem ($u_{|\tilde{P}_g|}$) is performed not before the closing time b_g is reduced by the maximum time that can elapse between two consecutive pickups at c_g ($\hat{T}_g$) multiplied by the number of pickups excluded from the subproblem ($|P_g| - |\tilde{P}_g|$). Hence, Constraint (B.9) provides a ***lower limit*** for the pickup visit time k. If this limit is not satisfied, a way to build the next routes is nonexistent without allowing some specimens collected by SCC g to perish.

References

Ait Haddadene, S. R., Labadie, N., & Prodhon, C. (2016). A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications*, *66*, 1339–1351.

Ali, O., Côté, J. F., & Coelho, L. C. (2021). Models and algorithms for the delivery and installation routing problem. *European Journal of Operational Research*, *291*, 162–177.

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.

Anaya-Arenas, A. M., Prodhon, C., Renaud, J., & Ruiz, A. (2021). An iterated local search for the biomedical sample transportation problem with multiple and interdependent pickups. *Journal of the Operational Research Society*, *72*, 367–382.

Araújo, A. M., Santos, D., Marques, I., & Barbosa-Povoa, A. (2020). Blood supply chain: a two-stage approach for tactical and operational planning. *OR Spectrum*, *42*, 1023–1053.

Baş, S., Carello, G., Lanzarone, E., Ocak, Z., & Yalçındağ, S. (2016). Management of blood donation system: Literature review and research perspectives. In *Springer proceedings in mathematics and statistics*: *vol. 169*, (pp. 121–132).

Baş, S., Carello, G., Lanzarone, E., & Yalçındağ, S. (2018). An appointment scheduling framework to balance the production of blood units from donation. *European Journal of Operational Research*, *265*, 1124–1143.

Benini, M., Detti, P., & de Lara, G. Z. M. (2019). A MILP model for biological sample transportation in healthcare. In *Advances in optimization and decision science for society, services and enterprises: ODS, Genoa, Italy, September 4-7, 2019* (pp. 81–94).

Benini, M., Detti, P., & de Lara, G. Z. M. (2022). Mathematical programming formulations and metaheuristics for biological sample transportation problems in healthcare. *Computers & Operations Research*, *146*, Article 105921.

Bonadies, A., Mancini, R., Maria Petrini, A., Gibertoni, C., & Maci, M. (2020). Laboratorio unico metropolitano: innovazione e alta tecnologia per un nuovo paradigma di medicina di laboratorio. (pp. 79–94).

Brailsford, S., & Vissers, J. (2011). OR in healthcare: A European perspective. *European Journal of Operational Research*, *212*, 223–234.

Bredström, D., & Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, *191*, 19–31.

Coelho, L. C., Cordeau, J. F., & Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, *48*, 1–19.

Coelho, L. C., & Laporte, G. (2013). The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, *40*, 558–565.

Decerle, J., Grunder, O., Hajjam El Hassani, A., & Barakat, O. (2018). A memetic algorithm for a home health care routing and scheduling problem. *Operations Research for Health Care*, *16*, 59–71.

Detti, P., Lara, G. Z. M. d., & Benini, M. (2021). A metaheuristic approach for biological sample transportation in healthcare. In *Graphs and combinatorial optimization: from theory to applications: CTW2020 proceedings* (pp. 293–306).

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.

Dohn, A., Rasmussen, M. S., & Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, *58*, 273–289.

Drexl, M. (2012). Synchronization in vehicle routing—A survey of VRPs with multiple synchronization constraints. *Transportation Science*, *46*, 297–316.

Elalouf, A., Tsadikovich, D., & Hovav, S. (2018). Optimization of blood sample collection with timing and quality constraints. *International Transactions in Operational Research*, *25*, 191–214.

Euchi, J., Zidi, S., & Laouamer, L. (2021). A new distributed optimization approach for home healthcare routing and scheduling problem. *Decision Science Letters*, *10*, 217–230.

Frifita, S., & Masmoudi, M. (2020). VNS methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. *International Transactions in Operational Research*, *27*, 291–313.

Ghandforoush, P., & Sen, T. K. (2010). A DSS to manage platelet production supply chain for regional blood centers. *Decision Support Systems*, *50*, 32–42.

Gunpinar, S., & Centeno, G. (2016). An integer programming approach to the blood-mobile routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *86*, 94–115.

Haitam, E., Najat, R., & Abouchabaka, J. (2021). A vehicle routing problem for the collection of medical samples at home: Case study of Morocco. *International Journal of Advanced Computer Science and Applications*, *12*, 345–351.

Kergosien, Y., Ruiz, A., & Soriano, P. (2014). A routing problem for medical test sample collection in home health care services. In *Springer proceedings in mathematics and statistics*, (pp. 29–46). Springer.

Liu, R., Xie, X., Augusto, V., & Rodriguez, 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.

Melachrinoudis, E., Ilhan, A. B., & Min, H. (2007). A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, *34*, 742–759.

Mobasher, A., Ekici, A., & Özener, O. Ö. (2015). Coordinating collection and appointment scheduling operations at the blood donation sites. *Computers & Industrial Engineering*, *87*, 260–266.

Mor, A., & Speranza, M. G. (2022). Vehicle routing problems over time: a survey. *Annals of Operations Research*, *314*, 255–275.

Naji-Azimi, Z., Salari, M., Renaud, J., & Ruiz, A. (2016). A practical vehicle routing problem with desynchronized arrivals to depot. *European Journal of Operational Research*, *255*, 58–67.

Osorio, A. F., Brailsford, S. C., & Smith, H. K. (2015). A structured review of quantitative models in the blood supply chain: A taxonomic framework for decision-making. *International Journal of Production Research*, *53*, 7191–7212.

Pirabán, A., Guerrero, W. J., & Labadie, N. (2019). Survey on blood supply chain management: Models and methods. *Computers & Operations Research*, *112*.

Pirabán-Ramírez, A., Guerrero-Rueda, W. J., & Labadie, N. (2022). The multi-trip vehicle routing problem with increasing profits for the blood transportation: An iterated local search metaheuristic. *Computers & Industrial Engineering*, *170*, Article 108294.

Rousseau, L. M., Gendreau, M., & Pesant, G. (2013). The synchronized dynamic vehicle dispatching problem. *INFOR. Information Systems and Operational Research*, *51*, 76–83.

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.

Soares, R., Marques, A., Amorim, P., & Parragh, S. N. (2023). Synchronisation in vehicle routing: Classification schema, modelling framework and literature review. *European Journal of Operational Research*.

Vidal, T., Laporte, G., & Matl, P. (2020). A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, *286*, 401–416.

Vilhelmsen, C., Lusby, R. M., & Larsen, J. (2017). Tramp ship routing and scheduling with voyage separation requirements. *OR Spectrum*, *39*, 913–943.

Wolfinger, D., Gansterer, M., Doerner, K. F., & Popper, N. (2023). A large neighbourhood search metaheuristic for the contagious disease testing problem. *European Journal of Operational Research*, *304*, 169–182.

Yousefi Nejad Attari, M., Pasandideh, S. H. R., & Akhavan Niaki, S. T. (2019). A hybrid robust stochastic programming for a bi-objective blood collection facilities problem (Case study: Iranian blood transfusion network). *Journal of Industrial and Production Engineering*, *36*, 154–167.

Yücel, E., Salman, F. S., Gel, E. S., Örmeci, E. L., & Gel, A. (2013). Optimizing specimen collection for processing in clinical testing laboratories. *European Journal of Operational Research*, *227*, 503–514.

Zabinsky, Z. B., Dulyakupt, P., Zangeneh-Khamooshi, S., Xiao, C., Zhang, P., Kiatsupaibul, S., & Heim, J. A. (2020). Optimal collection of medical specimens and delivery to central laboratory. *Annals of Operations Research*, *287*, 537–564.

Zahiri, B., Torabi, S. A., Mohammadi, M., & Aghabegloo, M. (2018). A multi-stage stochastic programming approach for blood supply chain planning. *Computers & Industrial Engineering*, *122*, 1–14.