AN ILS APPROACH TO SOLVE THE BIOMEDICAL SAMPLE TRANSPORTATION PROBLEM IN THE PROVINCE OF QUEBEC

ANA MARIA ANAYA-ARENAS[1], CAROLINE PRODHON[2], HASAN MURAT ASFAR[2], CHRISTIAN PRINS[2]

[1] CIRRELT AND DEPARTEMENT OPERATIONS ET SYSTEMES DE DECISION, FSA, UNIVERSITE LAVAL
Pavillon Palasis-Prince, 2325 rue de la Terrasse, Québec (G1V 0A6), Canada
AnaMaria.AnayaArenas@cirrelt.ca

[2] INSTITUT CHARLES DELAUNAY CNRS UMR 6281- LOSI, UNIVERSITE DE TECHNOLOGIE DE TROYES
12 rue Marie Curie, CS 4206, 10004 Troyes Cedex, France
caroline.prodhon@utt.fr, afsar@utt.fr, christian.prins@utt.fr

Résumé -
Le but de cet article est de résoudre le problème de transport d'échantillons biomédicaux pour le réseau de laboratoires de Québec. Les prélèvements sont effectués dans un établissement de soins de santé (ou centre de collecte d'échantillons — SCC). Ensuite, les échantillons doivent être expédiés vers un laboratoire externe (Lab) où ils seront analysés. Dans ce contexte, nous cherchons à définir une planification efficace de l'ensemble de routes ainsi que des heures d'ouverture à chaque SCC afin de minimiser le nombre de routes et le temps facturable (durée totale des routes), tout en garantissant qu'aucun des échantillons ne périsse, ni au SCC ni pendant le transport. Nous proposons d'aborder le problème par une procédure itérative de recherche locale (ILS). L'idée principale est d'explorer à chaque itération le voisinage local d'une solution perturbée. Nous avons testé la méthode proposée sur les instances inspirées du réseau de laboratoires de la province du Québec, Canada. Les résultats préliminaires obtenus semblent très prometteurs puisqu'en quelques secondes la méthode trouve la solution optimale pour les petites et moyennes instances, et des solutions de bonne qualité pour les plus grandes. Toutefois, nous croyons que la performance de l'heuristique ILS peut encore être amélioré.

Abstract -
The aim of this paper is to solve the biomedical sample transportation problem for the laboratory's network in the province of Québec. Biomedical tests are performed in a healthcare facility (or specimen collection center - SCC). Then, the collected specimens have to be analyzed, in most of the cases, at an external laboratory (Lab) to which the samples have to be transported. In this context, we seek to define an efficient set of routes to perform the optimal number of pick-ups at each SCC, coordinating SCC's opening hours, to minimize billable time (total route duration) and to warrant that none of the samples perish, neither at the SCCs nor during transportation. We propose to approach the problem heuristically, through an iterated local search (ILS) procedure. The main idea is to explore at each iteration a different neighborhood of a perturbed solution. The obtained preliminary results are very promising, achieving in few seconds optimal solution for the small and medium-size instances, and good-quality solutions for the larger ones. Further analysis on the ILS performance is planned, seeking a better performance.

***Mots clés* - Transport d'échantillons biomédicaux, Tournées de véhicules, Collectes multiples-interdépendantes.**
***Keywords* - Biomedical sample transportation; Vehicle routing; Multiple-interdepend pick-ups.**

1 INTRODUCTION

The biomedical sample transportation problem (BSTP) rises in the context of healthcare logistics to support accurate and in time diagnosis. Specimens are collected in a healthcare facility (or specimen collection center - SCC) that subsequently have to be analyzed. In most of the cases, the SCCs do not own the proper equipment to perform the analysis, thus the samples have to be sent to an external laboratory (Lab). This analysis has to be made inside the samples' lifespan, or the specimens will perish and tests have to be performed again. Therefore, due to samples' lifespan, an SCC might request several pick-ups per day to preserve samples' quality. Evidently, this transportation task requires an efficient planning to ensure service quality, to avoid delays and loss of samples and to reduce operation costs. This research work is an extension of our partnership with the Ministry of Health and Social Services (Ministère de la santé et des services sociaux - MSSS) of the Canadian province of Quebec. The MSSS is responsible for supporting and overseeing Quebec's health network, and one of its current priorities in is the optimization of the laboratories' network services (embedded in a supply chain optimization project named Optilab). In a previous work, [Anaya-Arenas et al., 2015] performed the formalization of the BSTP faced by the MSSS. The authors considered a first version of the problem where the SCCs state several transportation requests that have to be performed inside strict time windows, including also other time constraints related to the perishable nature of specimens and Quebec's laboratory network. The problem was modeled as a multi-trip vehicle routing problem with time windows (MTVRP-TW), two mathematical formulations were proposed, and MSSS's real instances were solved with a combination of fast heuristics and a commercial branch and bound software. In the current phase of the project, some key aspects on the network's structure are reconsidered, and an efficient metaheuristic is proposed, in order to seek a more productive transportation plan for the MSSS. Two primary hypotheses are lifted to include in our problem the optimization of the SCCs' operation decisions. First of all, we take a deeper look in the opening hours of the SCCs. Even if the collection periods are defined by the particular demand and workload of a SCC, in a crowed network, like the one in Montreal's downtown, many SCCs can have a similar (even the same) schedule. This fact can easily result in multiple and simultaneous transportations request at different locations and finally incompatibles SCCs for a transportation plan, increasing the transportation costs. A more interesting approach could be to define the right opening hours of the SCCs, keeping into account their demand and specific needs, in order to get a simpler and more synchronized schedule for the entire set of SCCs. Secondly, we include in our analysis the SCC's decisions concerning the number and frequency of the transportation requests. Until now, each SCC provided the number of pick-ups to be performed during the workday with strict time windows. However, these parameters represent a major restriction in the transportation planning, and can lead ultimately to inefficient use of resources. We thus consider that SCCs have a limited stocking capacity, which must be respected in order to avoid that samples perish. In other words, we can no longer talk about a fixed time window for a transportation request, but a maximum time span between two consecutives pick-ups that must be respected. This creates interdependency between pick-ups, and requires a different treatment in the routing planning. Route's schedule has now a direct influence in the feasibility of a given frequency of visits and, thus, other routes planned in the current solution. On the other hand, due to the geographic disposition and the time-constraints of a network, an extra pick-up to a particular client can result in greater flexibility in the transportation plan and ultimately a better global solution.

Blood collection and clinical specimen collection were first addressed in the 1970s [McDonald, 1972]. However, due to the specific constraints attached to the context, as well as the practical impact in social welfare, biomedical sample logistics has regained attention in recent years, and some related works had been added to the literature. [Yi, 2003] was one of the first to approach the blood transportation as a variant of the VRP with time windows (VRPTW). Their collection points need to be visited (once) inside its operation time window to bring blood back to a central depot to be treated. As there is only one pick-up that can be done to each center, the later the pick-up is done, more blood can be brought back to the depot. The objective is to maximize the quantity of blood that is collected and treated, respecting transportation's time constraints and minimizing logistics' cost. More recently, [Sahinyazan et al., 2015] present a tour mobile collection system for the Red Cross in Turkey. Their main concern is to define the tours for the mobile collection units (vehicles) selecting the nodes to be used as base for the collection. Then, a second tour needs to be planned each day to pick-up collected blood through the planning horizon. As in the case of [Yi, 2003], there is only one pick-up to perform at each client per day. This is a different and simpler approach that the one stated in our problem because there is no interdependency between pick-ups. On their side, [Yücel et al., 2013] present what they denominated as the collection for processing problem. Their contribution is including the laboratory's processing rate in the tour's planning. Contrary to the BSTP, routes are not restricted by any time constraints, but are rather formed to seek balance between the number of samples processed in a day and the transportation costs. Finally, [Doerner et al., 2008] present a problem from the blood collection process in the Austrian Red Cross, with the particularity that the SCCs' sites are temporal collection tends. This consideration implies that no proper samples' conservation can be assured, and represent multi-interdependent pick-ups for a single customer. The authors propose a complex MIP model to the problem (which demanded days to solve even small size instances), a construction heuristic and a branch-and-bound algorithm to find a schedule to a specific number of pick-ups. Notice that even if there are common features with [Doerner et al., 2008]'s work and ours, in their Red Cross case, the operation hours of each SCCs were already defined. Moreover, authors' approach to decide on the number of pick-ups for a given SCC was an exhaustive search in a limited number of neighbourhoods, resulting in a higher computational effort. On our side, we propose to embed these two aspects in the same resolution approach, seeking an integral solution to the global optimization of the BSTP.

The rest of this article is organized as follows. Section 2 describes the BSTP version approached in this work and Section 3 presents the resolution method proposed to solve it. Section 4 presents preliminary results from our numerical experiments, and Section 5 draws main conclusions and research perspectives of our work.

2 PROBLEM STATEMENT

The biomedical sample collection network in Quebec province is composed by a set of SCCs and laboratories (labs). Each SCC g has a specific operation timeframe in which samples are collected at a known and constant rate λ_g. This operation timeframe is denoted in the following as the SCC's *collection period* (O_g). Once collected, the specimens are pre-treated and prepared to be sent to the lab. In the current structure of the MSSS, SCCs are assigned to a specific lab, according to their geographical position, having thus a single-depot (single-lab) network for our transportation problem. Now, a SCC can collect hundreds of different specimens, and pre-treatment operations vary from a sample type to another. Even so, after preparation, all the samples are consolidated and stocked in standard refrigerated sample boxes, which are transported. A typical box can content up to 80 different samples. Due to the size of the boxes, and the small number of boxes send by SCCs at each pick-up, we consider that each SCC states a number of *transportation requests* during their work day and it will never overpass vehicles' capacity. A major determinant of the samples' lifespan is the capacity of the SCC to pre-treat the samples and stock them in optimal conditions before sending them to the lab. In fact, we assume that as long as samples are kept at SCCs, their deterioration is slowed down. However, each SCC has limited conservation capacity C_g that in many cases is smaller than the capacity required for the entire operation day. This means that, inside its operation hours, an SCC might need to call several transportation requests to liberate its stocking capacity. We can therefore estimate a maximal timespan inter-pick-ups to ensure that SCC's capacity is always respected as follows: $\Delta^g_{max} = C_g/\lambda_g$. This is the first particularity of the BSTP and a major challenge in transportation planning. The second one is the travel time limit. In fact, once the boxes leave the SCC, the optimal conditions of temperature cannot be warranted, so deterioration process is accelerated. Therefore, as soon as samples are out of the SCC facility, they must arrive to the lab within a *maximal transportation time*. This time limit can vary from a sample to another. Hence, each transportation request p has a particular time limit for transportation, noted T^p_{max}, deduced by the most urgent sample transported in it. To summarize, a vehicle must visit the SCCs and recover the samples' boxes (completely full or not) before a maximum time from its last visit, and return to the lab in time to avoid that any of transported samples (in the route) perish.

In addition to the time-constraints described before, the BSTP has other particularities related to the SCCs. First of all, SCCs opening time can be modified, inside a small window, in order to contribute to a better global transportation plan. Let us thus define a_g as the opening time for SCC g, which is inside its opening time window $[e_g, l_g]$. Let e_g be the earliest and latest hour that SCC g can open (respectively). However, remember that the collection period is fixed for each SCC, so once a_g has been decided, the collection period start and last exactly O_g hours. In that moment (b_g) the center closes its service to patients so no more samples are collected (i.e. $b_g = a_g + O_g$). Moreover, all samples collected must be treated in the same day (the length of the time horizon). Therefore, once the collection period has passed (at b_g), a last pick-up must be scheduled within the next φ_g minutes to transport the last collected samples before the SCC's employees leave. (Figure 1) illustrate how the opening decision and the other parameters interrelate. It also shows how once a pick-up decision has been made (here noted u_1), Δ^g_{max} fixes a limit in time to revisit the SCC.

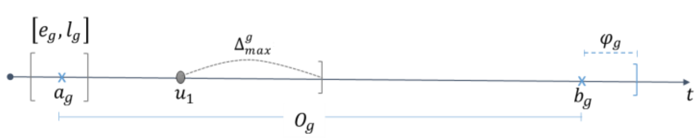

Figure 1. Example of opening decision and collection period

Finally, the collection period and the maximum timespan between pick-ups, allow us to fix for SCC g, a minimum number of transportation requests $P_g = \lceil O_g/\Delta^g_{max} \rceil$. However, there is no guarantee that this minimum number of pick-ups is optimal, and thus it needs to be included as a decision in the BSTP. (Figure 2) gives an example of the BSTP network as well as a feasible solution to the problem. Two routes (R1 and R2) that start and end at the lab visit the nine SCCs. Notice that SCC 2 and SCC 6 requested two pick-ups and are thus visited by both R1 and R2.

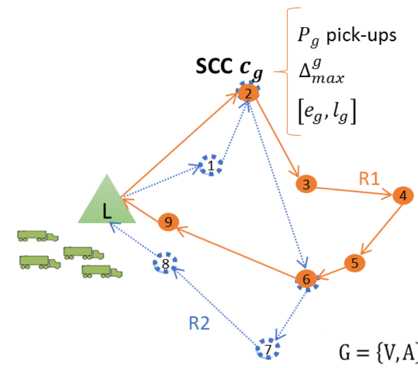

Figure 2. Example of a BSTP network configuration

It is worth to mention that the interdependency between the pick-ups also implies that pick-ups have to be sufficiently separated in time to ensure an efficient use of resources. Otherwise, a SCC might be visited an excessive number of times. This efficiency principle might lead to create waiting times at an SCC node. Consider, for instance, an SCC g that has a collect period from $7{:}00$ to $10{:}00$ a.m. and a maximum time span of $1{:}30$. If a vehicle arrives at $8{:}25$ a.m and makes the first pick-up immediately, a second pick-up has to be done before $9{:}55$ a.m to respect the SCC's capacity. This means that SCC g has to request a third pick-up after closing in order to send the samples collected in the last five minutes of operation. It might be more interesting to ask the vehicle to wait before performing the first pick-up (respecting still the Δ^g_{max}), get the samples at $8{:}30$ a.m. and then a last pick-up is requested at $10{:}00$. Hence, even if there are no time windows fixed to the transportation requests, a vehicle might have to wait at a SCC before the pick-up to assure than the next pick-up will not exceed Δ^g_{max}. Following the MSSS objectives, an external carrier will be selected to perform the samples transportation; hence, there are no limits in the available fleet of vehicles. In addition, as time constraints play a major role in the transportation planning for the BSTP, it will also be considered as the optimization objective, considering that carrier services are charged by total service time. This consideration allows us to avoid (minimize) waiting time for service, and it adapts to the MSSS operations context. Notice that there are no practical constraints forbidding that a given SCC is served more than once by the same route, even if this could rarely happen in practice.

We seek thus to define a set of routes to perform the optimal number of pick-ups at each SCC, minimizing billable time (total routes duration) and warranting that none of the samples perish, neither in the SCCs nor during transportation.

3 AN ITERATED LOCAL SEARCH FOR THE BSTP

We propose to approach the problem heuristically, through an iterated local search (ILS) procedure. The main idea is to explore at each iteration a different neighborhood through the local search (LS) of a perturbed solution, generated by removing a set of SCCs and reinserting them in the incumbent solution, until a stopping criterion is reached. In addition, this perturbation process allows us to explore neighborhoods where the SCCs are visited more

times than the minimum number requested. (Figure 3) presents a general schema of the three main stages of our ILS.
The three main stages in our ILS can be summarized as follows.

3.1 Initialization

To initialize the ILS, a first feasible solution is obtained routing the minimum number of transportation requests stated by each SCC. Due to the multiple time constraints and the interdependency between the pick-ups, construct a valid solution of good quality is quite challenging. Therefore, we propose a mixed integer program that includes all the practical constraints of the BSTP to be solved for a given number of transportation requests. The model proposed is formalized as follows:

3.1.1 MIP for a fixed number of pick-ups

We define $N = \{c_1, c_2, \dots, c_n\}$ as the set of the n SCCs assigned to the lab. We consider that each SCC c_g (with $g = \{1, \dots n\}$) requires a fixed number of transportation requests (set P_g), and all the transportation requests of all SCCs form a set P ($P = \cup_g P_g$). The BSTP can be modeled over a complete digraph $G = \{V, A\}$, where the set of nodes $V = \{v_0, v_1, v_2, \dots, v_{|P|}\}$ is composed by the $|P|$ transportation requests of all SCCs, and the *laboratory* ($\{v_0\}$) where the routes must start and end. We consider and unlimited fleet of vehicles available at the lab to perform the routes. Without loss of generality, we label the pick-ups so that $\{v_1, v_2, \dots v_{|P_1|}\}$ are the transportation requests of SCC 1, $\{v_{|P_1|+1}, \dots, v_{|P_1|+|P_2|}\}$ are requests of SCC 2 and so on. More precisely, we define I_g as the set of index for the requests of a center c_g where $I_g = \{\sum_{h=1}^{g-1}|P_h| + 1, \dots, \sum_{h=1}^{g}|P_h|\}$.
In addition, we define an arc set $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j, i, j = 0, \dots, |P|\}$ and for each arc (v_i, v_j) a fixed transportation time (t_{ij}) is known. Evidently, t_{ij} is fixed to zero to all v_i and v_j belonging to the same SCC ($v_i, v_j \in P_g$). In addition to the notation defined in Section 2, we need to consider the loading time at c_g (τ_g) and the unloading time of the vehicle at the lab before a new route can be started (τ_0). The objective is to define a transportation plan (set of routes) defining the service time for each one of the transportation request of each SCC, as well as the opening hours of the SCCs, in order to warrant that none of the samples perish and to minimize total route duration. The BSTP can be modeled as a MIP as follows:

Decisions variables

x_{ij}: binary variable equal to one if node i is visited before node j.
u_i: continuous variable that indicates the time when pick-up i is performed.
d_i: continuous variable that indicates the duration of the route that starts at node i.
a_g: continuous variable that indicates the opening hour of SCC g.
b_g: continuous variable that indicates the end of collection period hour of SCC g.
f_i^p: the maximum remaining time at node i, to bring request p to the lab before perishing.

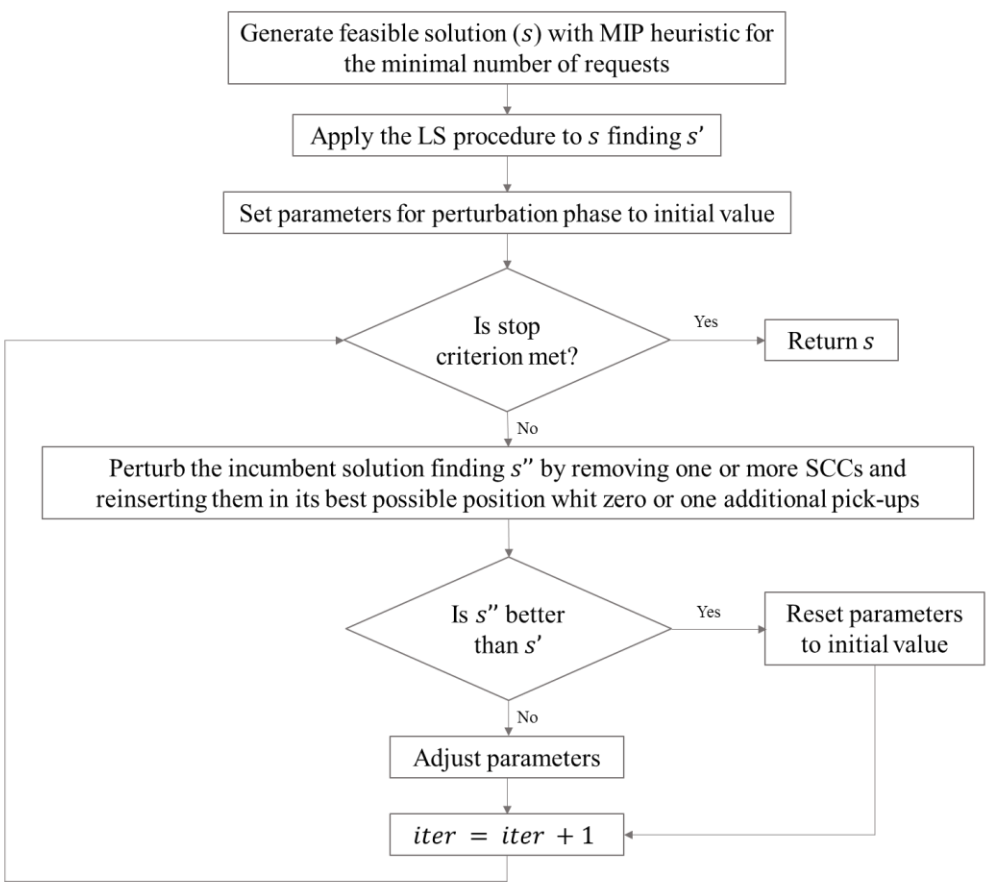

Figure 3. Basic structure our ILS approach

BSTP for a fixed number of pick-ups

$$Min\ \sum_{i=1}^{|P|} d_i \qquad (1)$$

Subject to:

$\sum_{i=0}^{\lvert P\rvert} x_{ij} - \sum_{i=0}^{\lvert P\rvert} x_{ji} = 0$	$j = 0, \dots, \lvert P\rvert$	(2)
$\sum_{i=0}^{\lvert P\rvert} x_{ij} = 1$	$j = 1, \dots, \lvert P\rvert$	(3)
$u_j \geq u_i + t_{ij} + \tau_i - M(1 - x_{ij})$	$i = 0, \dots, \lvert P\rvert;\ j = 1, \dots \lvert P\rvert;\ (i \neq j)$	(4)
$e_g \leq a_g \leq l_g$	$g = 1, \dots, n$	(5)
$a_g + O_g = b_g$	$g = 1, \dots, n$	(6)
$u_k - a_g \leq \Delta_{max}^g$	$g = 1, \dots, n$ where $\lvert P_g\rvert > 1; k = \sum_{h=1}^{g-1}\lvert P_h\rvert + 1$	(7)
$u_k - u_{k-1} \leq \Delta_{max}^g$	$g = 1, \dots, n$ where $\lvert P_g\rvert > 2; k = \sum_{h=1}^{g-1}\lvert P_h\rvert + 2, \dots, \sum_{h=1}^{g}\lvert P_h\rvert - 1$	(8)
$b_g - u_k \leq \Delta_{max}^g$	$g = 1, \dots, n$ where $\lvert P_g\rvert > 1; k = \sum_{h=1}^{g-1}\lvert P_h\rvert - 1$	(9)
$b_g \leq u_k \leq b_g + \varphi_g$	$g = 1, \dots n; k = \sum_{h=1}^{g}\lvert P_h\rvert$	(10)
$f_0 - f_i + M\left(1 - x_{ij}\right) \geq t_{i0} + \tau_i$	$i = 1, \dots, \lvert P\rvert$	(11)
$f_j - f_i + M\left(1 - x_{ij}\right) \geq u_j - u_i$	$i, j = 1, \dots, \lvert P\rvert\ \ (i \neq j)$	(12)
$f_0^i = T_{max}^i$	$i = 1, \dots, \lvert P\rvert$	(13)
$f_i^i \geq 0$	$i = 1, \dots, \lvert P\rvert$	(14)
$d_i \geq T_{max}^i - f_i^i + t_{0i} + \tau_0 - M(1 - x_{0i})$	$i = 1, \dots, \lvert P\rvert$	(15)
$u_i, a_g, b_g, d_i, f_i^i \geq 0$	$i = 0, \dots, \lvert P\rvert\ ; g = 1, \dots, n$	(16)
$x_{ij} = \{0,1\}$	$i \neq j = 0, \dots, \lvert P\rvert$	(17)

The objective (1) is to define a transportation plan that minimizes the total routes' duration. Constraints (2) ensure flow conservation on every node of the graph, while constraints (3) ensure that every transportation request j is satisfied. Constraints (4)-(10) are time constraints. Constraints (4) estimates the service time at request j (time in which pick-up j is performed) and eliminates the subtours between pick-ups. Constraints (5) assure that SCC g opens inside its given time window and constraints (6) fixes the end of the collecting period. Constraints (7) assure that the first pick-up of SCC g (pick-up $k = \sum_{h=1}^{g-1}|P_h| + 1$) is performed before Δ_{max}^g units of time after the SCC g opens. Constraints (8) check that all pair of consecutives pick-ups of SCC g (k and $k-1$, with $k = \sum_{h=1}^{g-1}|P_h| + 2, \dots, \sum_{h=1}^{g}|P_h| - 1$) respect as well the maximum timespan Δ_{max}^g, and constraints (9) ensure that the SCC g'collection period ends before Δ_{max}^g minutes after penultimate pick-up. Evidently, constraints (7) and (9) are imposed exclusively over the SCCs demanding more than a single pick-up during its collection period, and constraints (8) is only needed if the SCC request more than two pick-ups. In addition, constraints (10) state that the last pick-up of SCC g ($k = \sum_{h=1}^{g}|P_h|$) is performed after the end of the collection period but before the center closes. Constraints (11) to (14) control the flow of time restriction over all pick-ups. Please notice that our problem fixes for pick-up i a limited time to return to the lab (T_{max}^i) from the moment the pick-up is done (no from the moment the truck start a route). That is why our "time left" resource is consumed from lab to customers in the opposite direction of the route. This time consumption is directly related to the service time at each request. Therefore, constraints (11) and (12) ensure the coherence between the time resource variables and the service time variables for any pair of nodes (i, j), saying that if the arc (i, j) is included in the route, the difference between its respective service time variables must match the resource consumption difference. Then, constraints (13) state the available time at the lab for pick-up i at its limit T_{max}^i. Constraints (14) force the time resource to return pick-up i to be non-negative at i. Finally, constraints (15) estimate the duration of routes, saying that if the route start with transportation request i, its duration is greater or equal to the time consumed from i to be back to the depot, plus the travel time of arc $(0, i)$ and the loading time at the depot. If the route does not start with pick-up i, the duration will be fixed to 0. Constraints (16) and (17) define the decision variables' domain.

This model was implemented in Gurobi (v.6.0) and the solver provides feasible solutions in less than a second for all the proposed instances. This is already an encouraging and interesting result compared to other modeling approaches previously proposed for related problems ([Anaya-Arenas et al., 2014] and [Doerner et al., 2008]). Indeed, eliminate the vehicle index to define routes as we propose proves to be very efficient finding a feasible solution. Seeking to start our heuristic with a high-quality solution, we set a time limit which depends on the size of the instance to solve before stopping Gurobi and calling the LS procedure.

3.2 Local search procedure (LS)

In this phase we explore two basic neighbourhoods of the best solution produced by Gurobi. The major challenge in this case is to handle the interdependency between the transportation requests of a single SCC. Consider for instance a SCC g that has a total collection period of four hours and a half, but its capacity limit forces a pick-up every hour and a quarter ($\Delta_{max}^{g}= 1:15$). In addition, let us assume that SCC g has an opening time window between $[7:00 - 7:30]$ a.m. Then, SCC g have to be visited at least four times and no more than $1:15$ later than the last visit or the opening period. Suppose a feasible solution where SCC g have a collection period from $7:00$ to $11:30$ a.m. and vehicle one visits SCC g at $8:10$, vehicle two at $9:25$, vehicle three at $10:40$, and finally, vehicle 4 at $11:30$ to get any remaining samples. A possible neighbour of this solution could be to modify route two in such a way that the arrival time at SCC g would be advanced to $9:15$. In order to check if this change leads to a feasible solution, vehicle three has to visit SCC g before $10:30$, which might lead to change the pick-up schedule for all pick-ups (and thus the SCCs) that are included in route three. In other words, the whole solution should then be almost completely inspected to assure that the desired modification produces a feasible solution. Evidently, a complete inspection is not computationally efficient. To cope this difficulty, we introduce time windows to each transportation request that depend exclusively on the current solution (service time for each request) in order to reduce this interdependency. In addition, we implement concatenation techniques from [Vidal et al., 2014] to evaluate efficiently cost and feasibility of the movement explored. This two aspects are explained in the two following sub-sections.

3.2.1 Interdependency reduction

Notice that the time window of the opening decision, together with the maximum time span between pick-ups, define an initial approximation for the time windows for each transportation request. This first rough calculation of the possible service time can be very large and it might lead to an infeasibility for the SCC's capacity. However, once a service time decision is made, time windows are shrunk and reflect the "real flexibility" of the pick-ups, without affecting the other request of that center. Hence, to reduce the interdependency between pick-ups during the local search evaluation process, we estimate time windows for each transportation request based on the current solution. For index notation simplicity, we present an example of the time window computation for SCC c_1. We define $[\alpha_k - \beta_k]$ as the earliest and latest possible time at which pick-up k can be done without affecting the other pick-ups of SCC c_1. Let u_k (with $k = 1, \ldots, |P_1|$) be the service time of the transportation requests of SCC c_1 in a solution s. First, the opening and closing time windows are reduced by the service time of the first and penultimate requests. The service time for the first requested pick-up of an SCC (i.e. u_1) might delay the earliest opening hour (to find a new limit, named e_1') and service time of penultimate pick-up ($u_{|P_1|-1}$) might advance the latest possible closing (thus opening) hour defining l_1'. Precisely, for SCC c_1 one can define $e_1' = max\{e_1; u_1 - \Delta_{max}^1\}$ (17) and $l_1' = min\{l_1; u_{|P_1|-1} + \Delta_{max}^1 - O_1\}$ (18). The new time windows for opening and closing of SCC c_1, affects the time windows for the first, penultimate and last pick-up as follows:

$$\beta_1 = l_1' + \Delta_{max}^1 \quad (19)$$

$$\alpha_{|P_1|-1} = e_1' + O_1 - \Delta_{max}^1 \quad (20)$$

$$\alpha_{|P_1|} = e_1' + O_1 \quad (21)$$

$$\beta_{|P_1|} = l_1' + O_1 + \varphi_1 \quad (22)$$

Needless to say, the service time of pick-up j (u_j) sets the latest possible service time for $j + 1$ and the earliest possible service time for $j - 1$, i.e. $\alpha_{j-1} = u_j - \Delta_{max}^1$ (23) and $\beta_{j+1} = u_j + \Delta_{max}^1$ (24). For an SCC that requests only one pick-up, time windows are always the same (see equations 21 and 22). This approach eliminates the interdependency for any two consecutives pick-ups during the evaluation of moves. However, due to the fact that the opening and closing times are also flexible inside a window, there is still interdependency between the first, penultimate and last pick-up of each client. Consider again the example of SCC g, but this time, a move in route 1 could delay the arrival to SCC g to $8:45$. In order to validate if this delay still leads to a feasible solution, the earliest opening hour must be delayed to 7:30 (to respect the Δ_{max}^{g}). However, this will imply that the last pick-up is no longer feasible, because it is performed before the end of the collection period. Likewise, an advance in service time of penultimate pick-up might result in infeasibility for the first pick-up. This last dependency cannot be avoided and it is checked for every move considered that affects any of those three requests (for every SCC). If infeasible, the movement will be refused.

3.2.2 Cost and feasibility check

It is well known that a key component of any efficient LS is the cost and feasibility check procedure. In order to evaluate efficiently a move we implement the sequence concatenation technique proposed by [Vidal et al., 2014]. The basic idea is to define any solution (i.e. set of routes) as the combination of different sequence of visits. Moreover, edge exchanges and node relocations are finally a recombination of known sequences. Estimating significant information to describe the sequences and evaluate its concatenation allows us to evaluate all moves in constant time. However, once a move is implemented, the entire solution information has to be updated. The significant information that is used to characterize any sub-sequence of the BSTP are:

- the total service time $T(\sigma)$, i.e. the sum of travel and service time,
- the earliest finish time of a sequence $E(\sigma)$,
- the latest start time $L(\sigma)$,
- the minimal duration of the sequence $D(\sigma)$,
- and the feasibility statement $F(\sigma)$.

In addition, in order to control the return of all the samples to the lab before they perish, we compute $Tlim(\sigma)$ as the maximum transportation time available when the vehicle leaves the last SCC in the sequence σ, so all its samples arrive to lab on time. For a sequence of a single visit i ($\sigma_0 = p_i$) we define:

- $T(\sigma_0) = D(\sigma_0) = \tau_i,$ (25)
- $E(\sigma_0) = \alpha_i + \tau_i,$ (26)
- $L(\sigma_0) = \beta_i,$ (27)
- $Tlim(\sigma_0) = T_{max}^{i} - \tau_i$ (28)
- and $F(\sigma_0) = true.$ (29)

We refer the interested reader on the equations used to compute the information for a concatenation of sequence for $T(\sigma), E(\sigma)$, $L(\sigma)$ to [Vidal et al., 2014]. Now, remember that waiting time is allowed before serving a customer. Its estimation (ΔWT), as well as the one for its minimum duration $D(\sigma)$, are defined in [Vidal et al., 2013]. On its side, (30) and (31) show how $Tlim(\sigma_1 \oplus \sigma_2)$ and $F(\sigma_1 \oplus \sigma_2)$ are computed.

$$Tlim(\sigma_1 \oplus \sigma_2) = Min\{Tlim(\sigma_1) - \Delta WT - t_{\sigma_1(|\sigma_1|),\sigma_2(1)} - D(\sigma_2), Tlim(\sigma_2)\} \quad (30)$$

$$F(\sigma_1 \oplus \sigma_2) \equiv F(\sigma_1) \wedge F(\sigma_2) \wedge (E(\sigma_1) + t_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2)) \wedge (Tlim(\sigma_1) - \Delta WT - t_{\sigma_1(|\sigma_1|),\sigma_2(1)} - D(\sigma_2) \geq t_{\sigma_2(|\sigma_2|),0}) \quad (31)$$

Well-known neighbourhoods designed for the VRP, like 2-opt, have a high chance of leading to infeasibility solutions when applied to the BSTP, due to the time constraints in service and transportation. We thus limit our current application to two basic *relocate* neighbourhoods, incorporated in a variable neighborhood descent (VND) scheme. First, we explore the possibility of relocate a visit inside its route (**N1 :** intra-route). As our objective is to find the minimal duration for a route, for every neighbor solution explored, its feasibility is checked and then, if feasible, its minimal duration is computed and compared to the current solution. **N1** is explored completely to find the best possible improvement. Then, if an improvement is found, the movement is implemented, time windows and sequence information are updated, and we restart the LS of the incumbent solution. If no improvement is found, an inter-route relocate neighborhood is explored (**N2 :** inter-route) and we proceed as done in **N1**. When a move is performed in **N2**, the LS goes back to **N1** to explore the intra-route of the routes affected by the inter-route. The procedure is repeated until no further improvement can be found.

3.3 Perturbation

The LS procedure seeks to improve the routing decisions based on the estimated time windows calculated for a given solution. However, as it was explained before, this procedure ignores the flexibility that is added by the time window of the opening decisions, or the possibility of making extra visits to one or several SCCs. Indeed, considering the operation decisions for the SCC has a great complexity that can hardly be coped directly with the time windows estimation and relocate neighbourhoods. Hence, we propose to explore this upper-level decision through a perturbation of the current solution. The basic idea of the perturbation stage is to remove all the transportation requests of one or several SCCs from the solution, and then to reinsert them in the best possible position. Removing an SCC from the routes gives us the opportunity to reset the opening decisions and the service time decisions at the same time. Different criteria can be defined to select the SCCs to remove. However, seeking to guide the search, we limit the selection criterion to remove the SCC with the pick-up that produces the highest increase on total duration. Once the selected SCCs have been removed, the remaining routes are improved by LS, seeking to shake the solution. On its side, the time windows on the transportation request removed from the solution are reset to the first approximation and reinserted one by one. Evidently, at each request insertion, the time windows must be properly updated in order to avoid infeasibilities. Once all the requests have been reinserted, a new solution is obtained and its basic neighbourhoods are explored by the LS. Nonetheless, nothing can assure that the solution will be improved. In this case, the solution found is still accepted (implanted), but no improvement successive iterations are limited to a small number (*Fails'*) before a diversification is applied, defining a farther neighborhood.

There are two main diversification parameters in our ILS. First, we define ϑ_1 as the number of SCCs that are removed from the incumbent solution. Second, when an SCC is removed, it can be reinserted requesting the same numbers of pick-ups as before, or more. Although in a particular configuration making more than the minimum pick-up can result in a better result, it would not probably be the case for all the SCCs and for no more than one pick-up per SCC [Doerner et al., 2008]. The objective is thus to find the right set of customers (if any) that by adding one pick-up, lead to a better solution. To handle this aspect, let us define ϑ_2 as the number of additional pick-ups inserted to the request of the removed SCC. These two parameters are initially set to $\vartheta_1 = 1$ and $\vartheta_2 = 0$, and are increased gradually through the ILS if no improvement is reached. In addition, when an SCC has been removed during the perturbation procedure, it is tagged as *forbidden* to guide the search to a new part of the solution space. At the beginning, a single SCC is removed and reinserted (the same number of pick-ups as requested before). Then, if no improvement is found for a certain number of iterations, ϑ_1 is increased, and the process is repeated until a certain limit ϑ_1'. Finally, if no improvement is possible by removing ϑ_1' SCCs, we restart ϑ_1=1 and the SCCs removed will now be reinserted but with an additional pick-up (each). The forbidden list is restarted when ϑ_1 or ϑ_2 are incremented. Pseudo-code of the ILS is summarized by Algorithm 1: ILS.

Algorithm 1: ILS for the BSTP

1. $s \leftarrow$ MIP heuristic for minimum number of transportation request.
2. $s \leftarrow LS(s)$
3. **Fix** ϑ_1=1; $\vartheta_2 = 0$; *nbFails* = 0; *forbidden* = $\emptyset$; $iter = 0$
4. $s' \leftarrow s$
5. **Do until** *iter*= $iter'$ ***or*** all the SCCs are tagged as forbidden.
    - 5.1. Remove ϑ_1 SCCs from s' producing a partial solution $\hat{s}$ and tagged them as forbidden.
    - 5.2. $\hat{s} \leftarrow LS(\hat{s})$
    - 5.3. Reinsert the $|P_g| + \vartheta_2$ pick-ups of the removed SCCs to $\hat{s}$ obtaining s''.
    - 5.4. $s'' \leftarrow LS(s'')$
    - 5.5. **If** s'' isNot better than s' **then**
        - 5.5.1. *nbFails*= *nbFails* +1
        - 5.5.2. **If** *nbFails* = *Fails'* **then**
            - **If** ϑ_1<ϑ_1' andAlso ϑ_2=0 **then**
                - *forbidden*=$\emptyset$; *nbFails*=0;
                - $\vartheta_1 = \vartheta_1 + 1$
            - **elseIf** $\vartheta_2 = 0$ **then**
                - *forbidden*=$\emptyset$; *nbFails*=0; $\vartheta_1 = 1$; $\vartheta_2 = \vartheta_2 + 1$
            - **elseIf** $\vartheta_1 < \vartheta_1''$ **then**
                - *forbidden*=$\emptyset$; *nbFails*=0; $\vartheta_1 = \vartheta_1 + 1$
            - **else**
                - *nbFails*=0; $\vartheta_1 = 1$; $\vartheta_2 = 0$
    - **else**
        - *nbFails*=0; $\vartheta_1 = 1$; $\vartheta_2 = 0$
        - $s' \leftarrow s''$

5.6. $iter = iter + 1$
5.7. **If** s' is better than s **then** $s \leftarrow s'$
6. **return** s

4 NUMERICAL EXPERIMENTS

In order to assess the efficiency of the proposed approach, we solved the set of 38 instances proposed in [Anaya-Arenas et al., 2014]. Instances are arbitrarily divided into 12 *small instances* (four SCCs for a total of around a total of 10 "minimum request" to schedule), 13 m*edium* (up to 10 SCCs, around 20 "minimum requests") and 13 l*arge* (up to 20 SCCs, up to 50 "minimum requests") sets. In this section, we present the preliminary results obtained with our ILS and compare its performance to the best-known solutions (BKS), obtained by Gurobi after 30 minutes of computing (1800 sec.)

All the tests were run on a 64 bits Intel Core i7-4770 CPU @3.4 GHz. PC with 32 Gb. of RAM. For the initialization stage, the MIP was solved by Gurobi v6.0, allowing $\lceil |P|/4 \rceil$ seconds to solve small and medium size instances and $\lceil |P|/2 \rceil$ seconds for the larger ones. On the other side, we allowed the perturbation to remove up to 10% of the SCCs (i.e. $\vartheta_1' = \lfloor 10\% |N| \rfloor$), if there is no improvement found we explore to add up to one pick-up per SCC removed, and this to a maximum of two SCCs ($\vartheta_2' = 1; \vartheta_1'' = 2$). Finally, the maximum number of iterations was set to the double of the number of SCCs in the instance ($iter' = 2 \times |N|$) and the number of iterations allowed without improvement was set at each iteration between four and one, according to the number of removed SCCs ($Fails' = \lceil 4/\vartheta_1 \rceil$).

Table 1 presents the preliminary results of our ILS. Column *BKS Avg.* gives, for each set of instances, the average value of the objective function. Column *ILS Avg.* reports the average results produced by the ILS. Notice that (*) symbol in column *BKS Avg.* indicates that Gurobi was able to prove optimality of the solution reported. Column *%ToBKS* gives the average gap of our method with respect to BKS (in percentage). Finally, columns *Gurobi sec.* and *ILS sec.* reports the average CPU time of Gurobi and the ILS procedure (respectively) to solve an instance of each group.

Table 1: Average results over the three set of instances

	BKS Avg.	**ILS Avg.**	**%To BKS**	**Gurobi sec.**	**ILS sec.**
Small	389*	389	0	0,2	0
Medium	606*	606	0	7,2	1
Large	1586	1624	2,4	1547,7	18

These first computational results are quite encouraging. Our ILS achieves optimal solutions for all small and medium instances. Solutions for the small and medium sized instances were obtained in less than a second. For the set of large instances, ILS produced good-quality solutions (only 2.4% to BKS) in less than 20 seconds in average, which is very good compared to the average of 26 minutes reported by Gurobi. This shows how a metaheuristic approach is suitable to solve the BSTP.

Based on our empirical study, a deeper analysis is currently being held to improve the perturbation procedure. In addition, different parameters tuning (on the maximum number of SCCs to remove, additional visits to reinsert, or number of iterations) and other selection criteria for the removed SCCs are presently being tested to perform a greater shake of the solution and hopefully obtain optimal solutions for all of our instances.

5 CONCLUSION AND RESEARCH PERSPECTIVES

We present in this paper an efficient ILS to solve the biomedical sample transportation problem for Quebec's network of laboratories. This paper extends the version of the problem that was presented in [Anaya-Arenas et al., 2014] to cope with some key tactical decisions on the operation of SCCs. In fact, the problem approached in this paper aligns with the objectives of the MSSS to continue the analysis and improvement of their laboratories' network. More precisely, the version of the problem here described encompasses decisions on the number and the frequency of the SCCs' pick-ups, as well as their opening hours. We propose a mathematical formulation for the BSTP with a fixed number of transportation request that proved to be efficient in finding a feasible solution in a less than a second with a commercial solver. We develop an ILS procedure that aims at improving the initial solution produced by the MIP. The ILS explores routing decisions through relocate moves, and then a perturbation phase of the incumbent solution. The perturbation procedure seeks to adjust the SCCs' operation and pick-ups schedule to minimize total route's duration. Preliminary results are very promising. Our ILS achieves optimal solutions for the small and medium-size instances, and a solution in average around 2.4% behind the best known solutions for the larger ones, and this in 1% of the computational time required by the solver. Further analysis on the ILS performance is planned, seeking a better parameters' tuning and thus a better performance.

ACKNOWLEDGEMENTS

This research was mainly supported by the scholarship for international research internship attributed to Ana Maria Anaya-Arenas from the program for doctoral students of the *Fonds de recherche du Québec - Nature et technologies* (FRQNT). This financial support is gratefully acknowledged.

REFERENCES

Anaya-Arenas, A. M., Chabot, T., Renaud, J., and Ruiz, A. (2015). Biomedical sample transportation in the province of Quebec: a case study. *International Journal of Production Research*, (ahead-of-print), pp. 1-14.

Doerner, K. F., Gronalt, M., Hartl, R. F., Kiechle, G., and Reimann, M. (2008). Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. *Computers & Operations Research*, *35*(9), pp. 3034-3048.

McDonald, JJ. (1972) Vehicle Scheduling-A Case Study. *Operational Research Quarterly*. JSTOR, pp. 433–44.

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*(1), pp. 22-34.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, *40*(1), pp. 475-489.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, *234*(3), pp. 658-673.

Yi, J. (2003). Vehicle routing with time windows and time-dependent rewards: A problem from the American Red Cross. *Manufacturing & Service Operations Management*, *5*(1), pp. 74-77.

Yücel, E., Salman, F. S., Gel, E. S., Örmeci, E. L., and Gel, A. (2013). Optimizing specimen collection for processing in clinical testing laboratories. *European Journal of Operational Research*, *227*(3), pp. 503-514.