Routing and Scheduling Problem of N-Side

Fen Zhou, Haitao Wu, Min Ju, Jérémy Omer, Imane Sefrioui,
Luis Flores, Bruno Rosa, Samuel Deleplanque, Marcos de Melo da Silva

July 15, 2016

1 Problem description

The N-Side company presented a vehicle routing and scheduling problem for the transportation of patients to hospital using a heterogeneous fleet of vehicles. One specificity of the problem is that the duration of the consultations is uncertain. The deterministic version of the problem is a close variant of the dial-a-ride problem (DARP), which is the subject of several recent research projects (see e.g. [1, 2, 3, 4]). The growing interest for this topic has been motivated by the great difficulty of this combinatorial optimization problem.

In the next sections, we describe the parameters, constraints and objective of the problem.

1.1 Given parameters:

- A set of vehicles V with heterogeneous transportation capacities (e.g., 4 seats)
- A set of patients P with their location, the location of the health center they're going to, their appointment time, t_A, and the duration of the appointment, δ_A.

1.2 Constraints:

- A patient P_i should arrive at a hospital before his appointment time t_A^i
- If a patient goes to the hospital by a vehicle, he must go back home on a vehicle also but not necessarily the same one.
- A patient should not wait more than t_H minutes at the health before and after his appointment (so this can lead to a total $2t_H$ minutes wait at the health center).
- The ride time of a patient P_i is upper bounded by t_R^i
- The working time of a driver is upper bounded by T

1.3 Objective

The objective is to find the vehicles' plannings that maximize the number of transported patients while respecting all the time constraints (departure time, rendez-vous time, and arrival time to home) as well as the vehicle capacity constraint.

2 Work organization

Sébastien Mouthuy (Head of Innovation & Development Group) presented the studied problem. There are 11 people who are interested at this problem and participated in the discussion. Finally, we have decided to form the following working groups

- Group 1 (Global Routing Heuristic Algorithm): Fen Zhou (UAPV), Haitao Wu (UAPV), and Min Ju (UAPV)
- Group 2 (Global MIP solution): Jérémy Omer (INSA Rennes)
- Group 3 (MIP based Routes Selection): Imane Sefrioui (Univ Abdelmalek Essaadi) Luis Flores (UAPV), Bruno Rosa (UAPV)
- Group 4 (Insertion Heuristic Algorithm): Samuel Deleplanque (ULB/GOM)
- Group 5 (Data Analysis): Marcos de Melo da Silva (Univ Paris 13)

3 Global routing heuristic algorithm

3.1 Spatial-temporal Graph modeling

To route vehicles, each vehicle is represented by a path starting from the beginning of the day in the depot and visiting other locations (home of patients and the hospitals) until the end of the service in the same day. Such a path is basically characterized by its traces in two dimensions: space and time. Therefore, it is natural to use a Spatial-Temporal graph. We discretize the the daily service by a set of time period T. We model the VRPS problem with a Spatial-Temporal graph $G(V, A, C, R)$ where:

- The set of vertex $V = \{P_t^i : t \in T, i \in N\} \cup \{H =_t^i\} \cup \{v_s, v_d\}$ is composed of the vertex P_t^i for patients and the vertex for hospitalH_t^i, where i is a location and t is a period of time, in addition of two vertex v_s and v_d which respectively represents the start and the end of the service of a driver.
- The set of arcs A is the union of three sets of arcs $A = A_1 \cup A_2 \cup A_3$:
  - $A_1 = \{(v_t^i, v_{t+1}^i), (v_s, v_i^1), (v_i^{max_t}, v_d) : i \in N, t \in T\}$. Each arc (v_t^i, v_{t+1}^i) represents a link within the same location between two consecutive periods of time, which allows the path crossing this arc to model a vehicle staying in the same location;
  - $A_2 = \{(v_{t'}^k, v_t^i) : i \in N, k \in N, t \in T, t' \in T, (t = t' + d_{ki} \leq d_k \wedge i \neq k)\}$. Each arc $(v_{t'}^k, v_t^i)$ represents a link between two different locations i and k, $t = t' + d_{ki}$ ensures that a vehicle respects the

travel time from location i to k and respects the time window (the departure time and the arrival time);

  - $A_3 = \{(v_{t'}^k, v_t^i) : i \in N, k \in N, t \in T\}$. Each arc $(v_{t'}^k, v_t^i)$ represents a patient request to go from location k at t' to hospital i at $t - Tr_{ki}$;

- The set of costs C is the union of three sets of costs $C = C_1 \cup C_2 \cup C_3$:
  - $C_1 = \{c_{(v_{t'}^k, v_t^i)} : (v_{t'}^k, v_t^i) \in A_1\}$. Each $c_{(v_{t'}^k, v_t^i)} \in C_1$ represents the cost for a patient to stay at the station i between the two periods of time t' and t.
  - $C_2 = \{c_{(v_{t'}^k, v_t^i)} : (v_{t'}^k, v_t^i) \in A_2\}$. Each $c_{(v_{t'}^k, v_t^i)} \in C_1$ represents the cost for a vehicle to go a location k at t' to the location i at t.
  - $C_3 = \{c_{(v_{t'}^k, v_t^i)} : (v_{t'}^k, v_t^i) \in A_3\}$. A negative cost $c_{(v_{t'}^k, v_t^i)}$ is affected to each arc $(v_{t'}^k, v_t^i) \in A_3$, which represents a client demand and is used to direct the shortest paths into client demand arcs A_3. The negative cost value allows attraction in the process of calculating shortest paths to satisfy customer demands. Indeed, over a course of vehicle, the more customers are satisfied, the more cost decreases through traversing arcs of negative cost.

3.2 Heuristic algorithm

INPUT: the set of vehicles and the data of patients.
OUTPUT: the schedule of the vehicles.

- **Step 1.** Construct the Spatial-temporal Graph $G(V, A, C, R)$ according to its definition.
- **Step 2.** Arrange the schedule for each vehicle by the following principles:
  - Each vehicle starts at the node v_s of G.
  - Each vehicle has two properties:
    * Capacity: represents the residual capacity of the vehicle.
    * Compatibility: represents the vehicle can serve these patients who now are already on this vehicle by keeping all the time constraints.
  - At current time period t, starting from t_0, the vehicle search the next moving by following principle:
    * If the capacity of the vehicle is 0 which means the vehicle can not take any more patients, the vehicle deliveries these patients to corresponding hospitals or homes according to the strategies.
    * If the capacity is not 0, search and pick up these waiting service patients (waiting going to hospital or back to home) which the vehicle can pick up such that the compatibility is kept, i.e., there is a routing strategy to serve all the patients on the vehicle and keep all the time constraints.
    * If the capacity is not 0 and no waiting service patients can be picked up, the vehicle deliveries these patients to corresponding hospitals or homes according to the strategies.

* $*$ One vehicle is finished arrangement when current time period is t_{final}

- **Step 3.** Select these patients which are fully serviced, i.e.,going to hospital and back to home by vehicles as our final served patients and output corresponding schedule for vehicles.

Remark: The routing strategy in step 2 can be implemented by greedy algorithm or enumeration.

3.3 Illustrative Example

We give an example in this section. There are three patient locations denoted by P_1, P_2, P_3 and the number of patients in each location are 4, 2 and 2 respectively. The Spatial-temporal Graph is shown in Figure 1.

We discrete the service time of one day into serial time periods from t_0 to t_{final}. We assume that t_1, t_2 and t_3 are the time periods of leaving for patients at P_1, P_2 and P_3 and the objective hospitals of patients at P_1, P_2 and P_3 are H_1 H_2 and H_1 respectively. Supposes there are two vehicles whose capacity is 4 and the view time of each patient costs two time periods.

First we arrange the schedule of one vehicle denoted by the red line. According to the heuristic algorithm, the vehicle starts from (v_s, t_0) to (P_1, t_1). As the number of patients at P_1 is 4, vehicle picks up the 4 patients and its capacity is changed to 0. Therefore vehicle directly deliveries the 4 patients to H_1 and its capacity is changed to 4 at (H_1, t_2). Then there are two patients waiting to leave at (P_3, t_3). But it will take two time periods from H_1 to P_3 thus the vehicle can not serve the patients at (P_3, t_3). The next task is at (H_1, t_4) to delivery the 4 patients back to P_1. Following the similar process, we finally arrange the schedule for the two vehicles indicated by the red line and green lines.

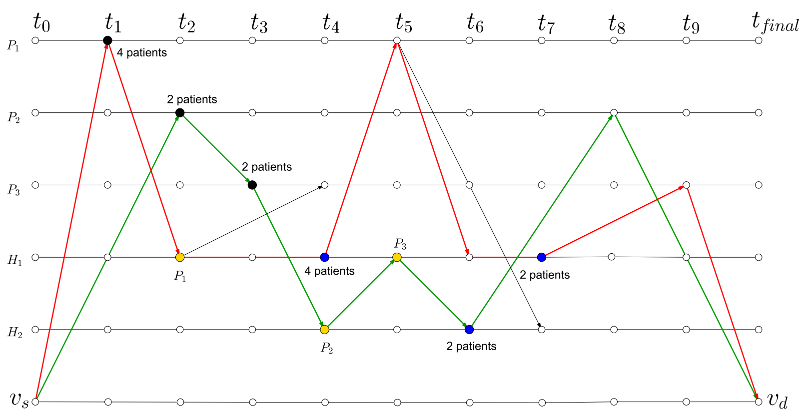

Figure 1: Example of heuristic algorithm in subsection 3.2

4 Mixed integer linear programming approach

The classical compact formulation of the DARP can be found for instance in [3]. The formulation considers a graph of tasks $(\mathcal{V}, \mathcal{E})$. Denoting n the number of patients, the transport of a patient P_i is associated with two pairs of vertices (v_i, v_{i+2n}) and (v_{i+n}, v_{i+3n}) that correspond to the origin and destinations of the way to the health and of the way back home. The edges include link all the pairs of vertices $(v_i, v_j) \in \mathcal{V}^2$ except those that do not respect the chronological succession $v_i \to v_{i+2n} \to v_{i+n} \to v_{i+3n}$. Each vertex v_{i+2n} has a time window $[t_A^i - t_H; t_A^i]$ stating that patient i must arrive at the health center before his appointment and cannot wait more than t_H minutes before. A similar time window $[t_A^i + \delta_A^i; t_A^i + \delta_A^i + t_H]$ is associated with vertex v_{i+n} to express the time constraints of the pickup at the health center after the appointment.

To adapt the formulation to our specific problem, we had to consider a set of variable $(z_i^k, z_{i+n}^k) \in \{0,1\}^2$ for all patients P_i and all vehicles k, such that $z_i^k = 1$ if and only if vehicle k takes P_i to his health center. As a consequence, the two specificities of our problem with respect to the classical DARP is that we maximize

$$\sum_{i,k} z_i^k$$

, and add the constraint

$$z_i^k = z_{i+n}^k, \forall i, k$$

to express that a patient has to do the round-trip if he is taken care of.

We implemented the model using the mathematical programming modeling language AMPL. The model files are attached to this report with a ".mod" extension. Since AMPL is very expressive, we refer directly to the code for the detailed formulation of the model.

4.1 Preprocessing the edges of the graph of tasks

To accelerate the solution of the model, we preprocess the graph of tasks to delete all the useless edges, based on the time windows of the vertices.

For this, we first compute time windows for the nodes of pickup at home v_i and that of return at home v_{i+4n}. Let $t_{DR}^{i,j}$ be the direct ride time from vertex v_i to v_j. Then, patient i cannot leave home after $t_A^i - 2t_S - t_{DR}^{i,j}$ if he needs to be at his appointment at t_A^i. Likewise, he cannot leave home before $t_A^i - t_H - 2t_S - t_R^i$ if his ride cannot last more than t_R^i. As a consequence the time window of v_i is

$$[t_A^i - t_H - 2t_S - t_R^i; t_A^i - 2t_S - t_{DR}^{i,j}].$$

Similarly, v_{i+4n} has a time window

$$[t_A^i + \delta_A^i + t_S + t_{DR}^{i,j}; t_A^i + \delta_A^i + t_H + t_S + t_R^i].$$

Now that every node v_i has a relatively tight time window, denoted $[l_i; u_i]$, we can use the following rules to simplify the graph of tasks:

1. *Arrive before the end:* if $l_i + t_S + t_{DR}^{i,j} > u_j$, then delete (v_i, v_j).
2. *Respect max ride time:* if v_i is a pickup node and $l_j - t_S > u_i + t_S + t_R^i$, then delete (v_i, v_j).

3. *Triangle max ride time:* if v_i is a pickup node and $t_{DR}^{i,j} + t_{DR}^{j,i+2n} + t_S > t_R^i$, delete (v_i, v_j) and (v_{i+2n}, v_j).

We can also add a heuristic rule to delete extra edges. The rules formulates that an efficient route should not lead to long waits at some node. Denoting t_W, a chosen threshold, the rules can be expressed as: if $u_i + 2t_S + t_{DR}^{i,j} < l_j + t_W$, delete (v_i, v_j).

All the deletion rules are coded in the AMPL script file, "NSIDE.run", attached to this report.

4.2 Alternative formulations

The original formulation that we implemented contains so-called *big-M* constraints to compute the starting service time at each node and the capacity after each node. As an illustration, we consider the continuous variables τ_i^k, the minimum starting service time of patient i by vehicule k, and the binary variable $x_{i,j}^k$ corresponding to the flow value of vehicule k on the edge (i,j). Then, the propagation of the starting service time on edge (i,j) for vehicule k is given by:

$$\tau_j^k \geq \tau_i^k + t_S + t_{DR}^{i,j} - M(1 - x_{i,j}^k), \forall (i,j), \forall k$$

where M is the classical "big-M" value, that can optimally be set to $(u_i - l_j)$.

It is well-known that *big-M* formulations usually fail to provide good upper bounds based on the continuous relaxation. As a consequence, we considered two slightly different formulations were studied to improve these constraints.

The first alternative formulation, referred to as the "reduced" formulation, does not deal with the *big-M* constraints but reduces the number of variables and constraints. We actually observed that it is not necessary to use one variable for the starting service time per vehicule. Instead, it is sufficient to consider the variables τ_i corresponding to the starting time of service at node i. With this, the propagation constraints become:

$$\tau_j \geq \tau_i + t_S + t_{DR}^{i,j} - M(1 - \sum_k x_{i,j}^k), \forall (i,j).$$

Finally, we followed the exact opposite process to build an extended formulation. For this we consider additional starting time variables for each arc and each vehicule $\tau_{i,j}^k$ to avoid the use of the "big-M" value. The variable $\tau_{i,j}^k$ takes a value equal to the starting time of service at node i if and only if vehicule k goes through edge (i,j). We obtain the following constraints:

$$\tau_j^k \geq \tau_{i,j}^k + (t_S + t_{DR}^{i,j}) \times x_{i,j}^k, \forall (i,j), \forall k \quad (1)$$

$$\tau_i^k = \sum_{i,j} \tau_{i,j}^k, \forall i, \forall k \quad (2)$$

$$\tau_{i,j}^k \leq u_i \times x_{i,j}^k \quad (3)$$

4.3 Preliminary results

Preliminary tests were done on DARP instances of Cordeau[1]. These tests were chosen, because some instances are smaller than those provided by NSIDE, and

[1] The instances are available online `http://neumann.hec.ca/chairedistributique/data/darp/branch-and-cut/`

	a2-16	a5-40	a5-60
complete graph	982	6298	14248
preprocessed graph	168	882	1412

Table 1: Number of edges in the task graph

	complete graph			preprocessed graph		
	a2-16	a5-40	a5-60	a2-16	a5-40	a5-60
big-M	0.38	1112	> 3600	0.08	7.6	2731
reduced	0.24	200	> 3600	0.09	38	470
extended	13.11	> 3600	> 3600	0.14	400	> 3600

Table 2: Runtime to solve the compact MILP formulation

we do not expect the MILP approach to scale up. The test instances are "a2-16", "a5-40" and 'a5-60". They correspond to the problem proposed by NSIDE with 8 (2), 20 (5) and 30 (5) patients (vehicles). The format of Cordeau was slightly modified to fit the AMPL requirements. The corresponding AMPL data files are attached to this report.

In these tests, we always used CPLEX 12.6.3 to solve the MILPs, because it provided he best results in our first tests. Table 4.3 provides the number edges in the complete graph and that in the preprocessed graph, and in Table 4.3, we present the CPU times for the three formulation ("big-M", "reduced", "extended"), with and without preprocessing the graph.

The analysis of every result is not straightforward. Clearly, preprocessing the graph is absolutely necessary and the extended formulation has no added value. However, the choice between the reduced and the initial formulations would require additional experimental tests, even if the reduced formulation seems better at this stage.

4.4 Using the compact formulation for large instances

The results presented in the previous clearly indicate that the MILP approach will not scale up to solve the real instances NSIDE has to deal with. However, it might be used in a heuristic approach based on clustering the patients into subgroups that are likely to be served together. Clustering was not studied in these three days, but it should be emphasized that it is also a difficult issue, because it must consider the geographical and temporal aspects altogether.

Nevertheless, results from the literature suggest that en extended formulation based on column generation will be more adapted for a mathematical programming-based approach. The one restriction is that it requires an intensive effort in development. Such an approach can be found in [5, 3]. A heuristic approach that would be also based on the generation of valid routes for one vehicle, dynamic programming has also shown to be efficient (see e.g. [4]).

5 MIP based Routes Selection

A randomized insertion heuristic was used to generate a set of feasible candidate routes. The routes are generated under some constraints, either by adding nodes or links. The constraints include the number of routes, the maximal duration of a route, the travel time, the time window of each patient, and the capacity of each vehicle. Once a number of routes are generated, a route selection is performed to select the best ones. Route selection requires a list of feasible candidate routes and assemble a solution from the candidates. The idea is to select the subsets that maximize the number of patients served.

Figure 2 shows a set of 4 possible routes generated for a problem of 6 patients. Figure 3 shows the selected routes that maximizes the number of patients served. We note that only patients 1 and 3 are not served when selecting these 2 routes.

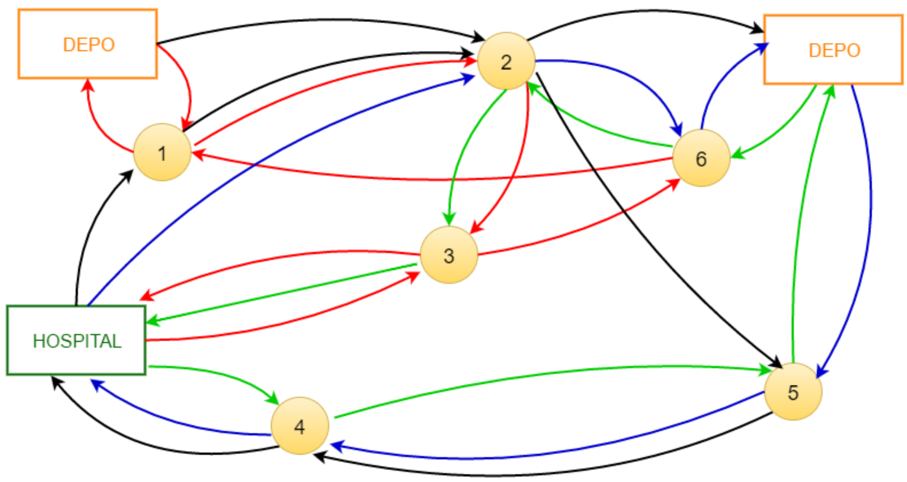

Figure 2: Example of 4 possible routes generated for the problem of 6 patients.

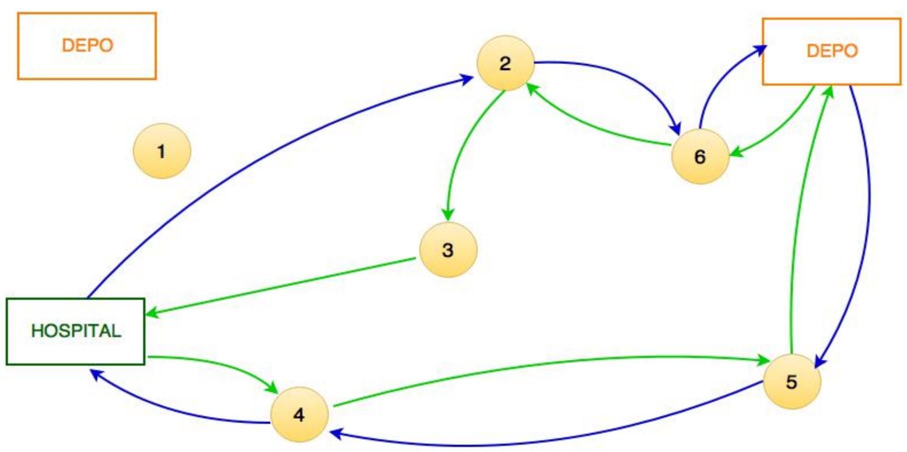

Figure 3: Example of the 2 selected routes that serve 4 patients.

The Integer Linear Programming model for the problem of the route selection is proposed below.

5.1 Model

Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of the patients, $R = \{r_1, r_2, \dots, r_m\}$ the set of feasible routes and $V = \{v_1, v_2 \dots, v_p\}$ the set of vehicles. We assume to know which are the patients that are picked up at home and taken to the hospital as well as the patients that are picked up at hospital and taken back home by each route $r_j \in R$. So, given a patient $p_i \in P$ and a route $r_j \in R$, the constant $a_{ij} = 1$ iff route r_j takes the patient p_i to the hospital and $a_{ij} = 0$ otherwise. Similarly, the constant $b_{ij} = 1$ iff route r_j takes the patient p_i to home and $b_{ij} = 0$ otherwise.

5.1.1 Decision Variables

The binary decision variables are defined as follows:

$$x_j = \begin{cases} 1 & \text{if the route } r_j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \qquad l_i = \begin{cases} 1 & \text{if the patient } p_i \text{ is served} \\ 0 & \text{otherwise} \end{cases}$$

5.1.2 Constraints

The goal is to serve as many patients as possible, i.e., we can maximize the following objective function:

$$maximize \quad \sum_{i \in P} l_i \tag{4}$$

subject to:

$$\sum_{j \in R} a_{ij} \cdot x_j \geq l_i \quad \forall i \in P \tag{5}$$

$$\sum_{j \in R} b_{ij} \cdot x_j \geq l_i \quad \forall i \in P \tag{6}$$

$$\sum_{j \in R} x_j \leq |V| \tag{7}$$

$$x_j, \quad l_i \in \{0, 1\} \qquad \forall j \in R \quad \forall i \in P \tag{8}$$

- The objective function (1) expresses the number of patients served (pick up and delivery) that should be maximized.
- Constraint (2) and (3) guarantee that the patient is served if he is picked up to go to the hospital and brought back home after consultation.
- Constraint (4) states that the number of routes chosen does not exceed the number of vehicles.
- Constraint (5) imposes the binary variables.

5.2 Experiments

We used CPLEX to solve the model. Experiments are done on 2 GHz Intel Core i5 processor with 4 GO RAM. Results are summarized in table 3.

Table 3: Experiments

Patients	Routes	Patients served	Time(s)	GAP (%)
100	100	53	12	0
100	200	56	46	0
100	300	58	100	0
100	400	58	360	0
100	500	56	900	23.63
120	100	63	15	0
120	200	64	87	0
120	300	65	536	0
120	400	65	900	21.96
120	500	67	900	25.05
150	100	77	21	0
150	200	79	105	0
150	300	80	655	0
150	400	78	900	28.98
150	500	80	900	30.90
200	100	94	74	0
200	200	98	900	2.96
200	300	99	900	30.79
200	400	99	900	35.89
200	500	102	900	35.99

In particular, we built some generated scenarios by varying the number of patients in the set $\{100, 120, 150, 200\}$, the number of routes in the set $\{100, 200, 300, 400, 500\}$. In total we generated 20 different instances. We note that all of them were run for a time limit set to 900 seconds (15 min). We report:

- Patients served: The value of the objective function (the number of patients that were satisfied).
- Time: CPU time required to solve the instance (seconds).
- GAP: the percentage between the feasible integer solution found and the optimal (%).

From Table 3, we can see that the model is able to serve more than a half of the patients. For 12 instances, we were able to find optimality within a short time. For example, for 100 patients and 400 routes, the optimality was found in a reasonable time (6 minutes). In only 8 cases, the optimality was not found within 15 minutes, but a solution was returned and a GAP was not greater than

35.99% for 200 patients and 500 routes. In average, for these cases, solutions were found in 15 minutes for an average GAP of 28.37%.

After that a set of routes are selected, we can apply improvement to repair the solutions by removing the patients that are present in more than one route.

We can serve per day an average of 200 patients and it is necessary for the program to return a solution in a maximum of half an hour when receiving a total of 1000 routes. Choosing the optimal solution depends on the quality and quantity of route selection.

6 Insertion Heuristic Algorithm

7 Data Analysis

The data provided by the representative of N-Side concerns the service provided in the city of Liege, Belgium. It contains the patient records of one year grouped by week (among 100 and 300 entries per week). The service includes more than 25 medical centers, a central depot and a heterogeneous fleet of vehicles (capacity among 5 and 7 places). Additionally, a patient appointment also contains his home coordinates, the respective hospital coordinates, the rendez-vous time, the type of medical care, and a expected visit duration.

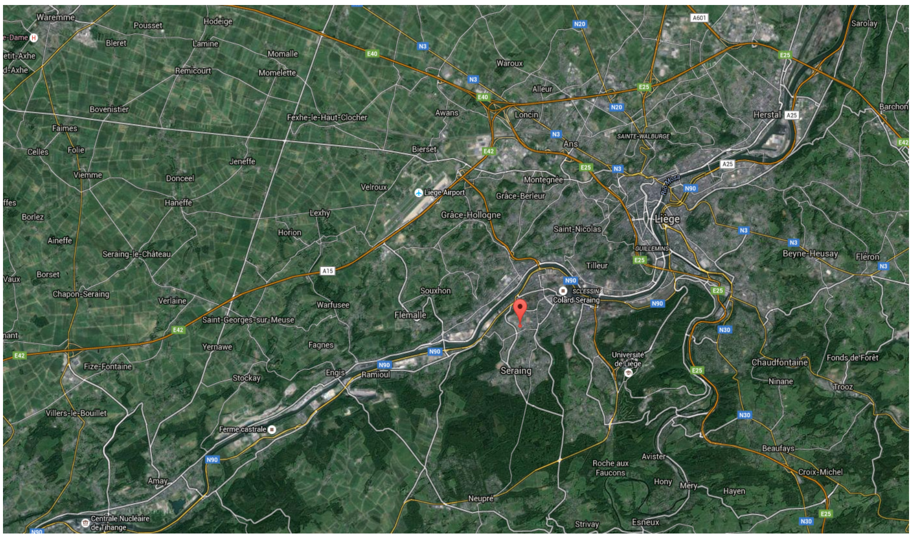

Figure 4: Satellite view of the city of Liege, Belgium.

In order to use the real data during the workshop, the provided files needed to be clean so that inconsistent entries and visits that last more than one day (e.g., hospitalization) were removed. The GPS coordinates were converted to Cartesian coordinates with the depot at the origin. Figure 5 depicts the appointments for one week. The cleaned files were then transformed to the format used in the DARP instances of Cordeau.

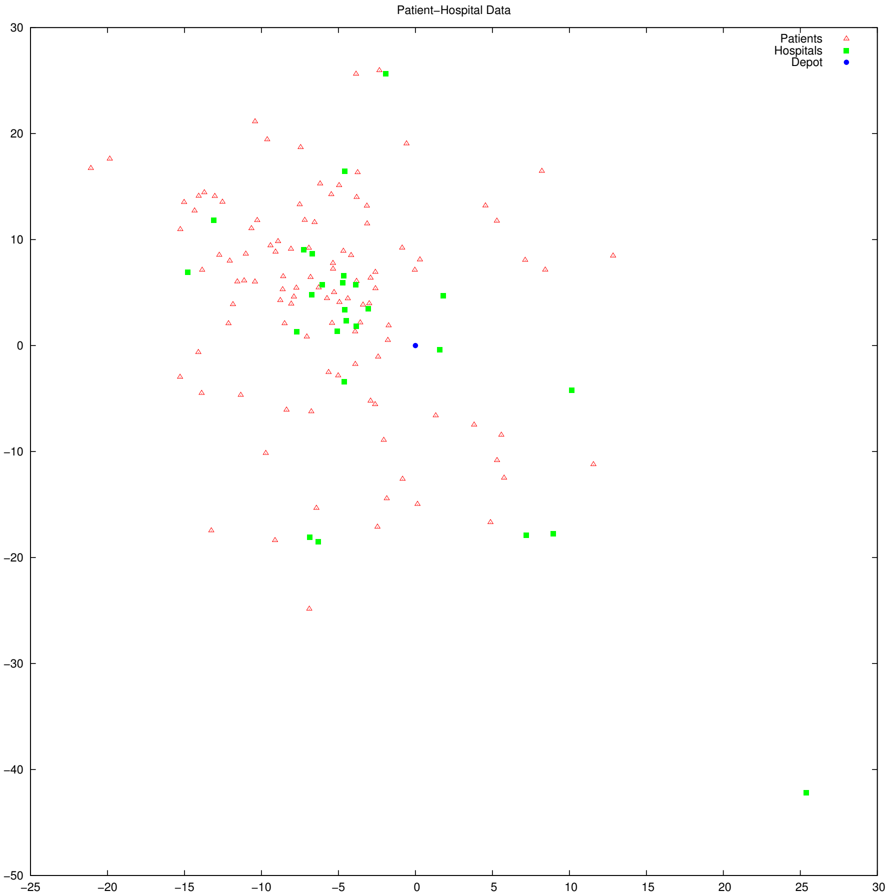

Figure 5: One week data after cleaning and coordinates conversion.

An initial data analysis and clustering has been performed. Figures 6 and 7 shows some of the links between patients and the respective medical centers. We know that is not always possible to schedule all the clients, and the analysis of geographical distribution of centers and patients can indicate appointments that will present some difficulties to be covered; for example, in Figure 7, only one patient has appointment in center 9 and they are far away from each other.

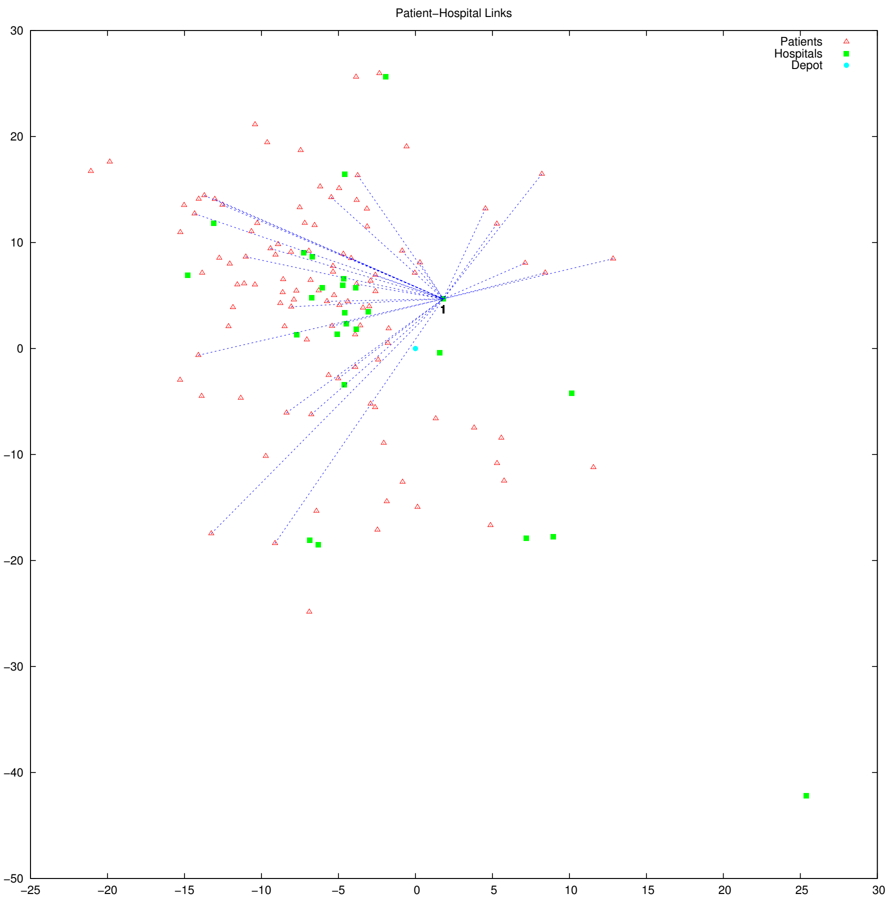

Figure 6: Patients with appointments in medical center 1.

8 Conclusions and future work

N-Side has presented a routing and scheduling problem in ESGI, which is similar to the famous dial-a-ride problem (DARP). To this end, several possible approaches have been proposed, for instance global MILP formulation, MIP

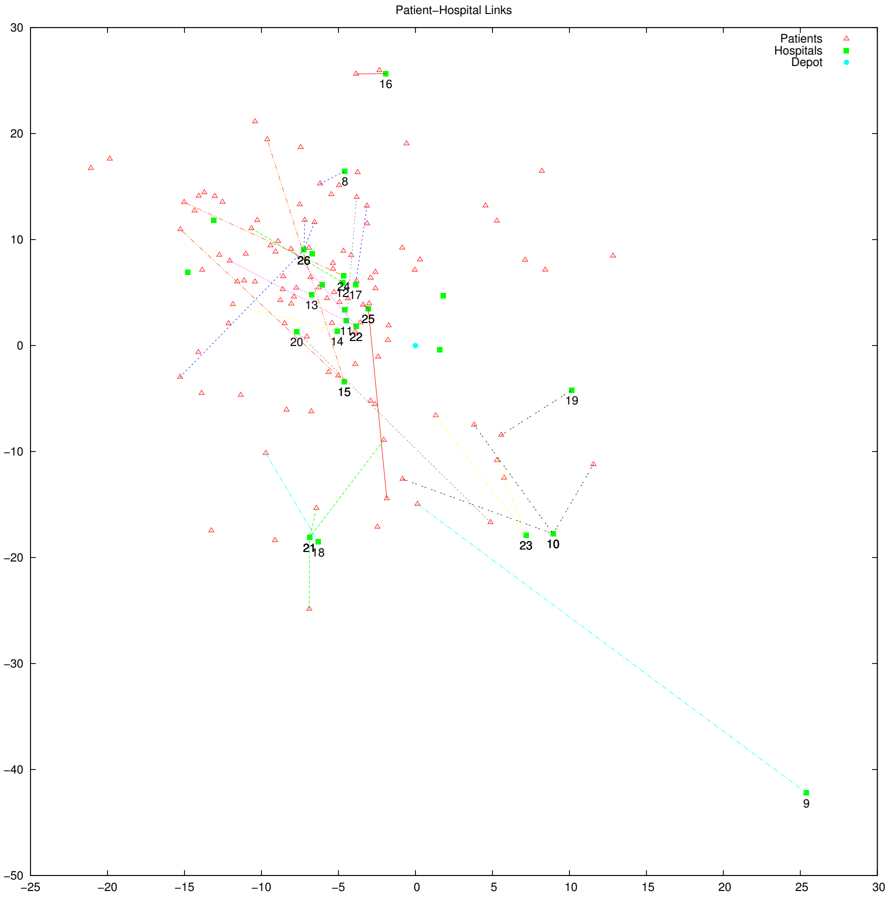

Figure 7: Patients with appointments in medical centers 8 to 26.

based routes selection method, global incremental routing heuristic algorithm. Due to the limitation of working time, only the main ideas and some preliminary results have been presented in this report. For future work, the proposed methods may be implemented and applied to solve the real instances provided by N-Side.

References

[1] G. Heilporn, J.-F. Cordeau, and G. Laporte, "An integer L-shaped algorithm for the Dial-a-Ride Problem with stochastic customer delays," *Discrete Applied Mathematics*, vol. 159, no. 9, pp. 883–895, jun 2011. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2011.01.021http://linkinghub.elsevier.com/retrieve/pii/S0166218X11000473

[2] I. Tensen, "Stochastic optimization of the dial-a-ride problem," Master thesis, University of Twente, 2015.

[3] N. Rahmani, "Planning and routing via decomposition approaches," PhD thesis, Université de Bordeaux 1, 2016.

[4] U. Ritzinger, J. Puchinger, and R. F. Hartl, "Dynamic programming based metaheuristics for the dial-a-ride problem," *Annals of Operations Research*, vol. 236, no. 2, pp. 341–358, jan 2016. [Online]. Available: http://dx.doi.org/10.1007/s10479-014-1605-7http://link.springer.com/10.1007/s10479-014-1605-7

[5] S. Ropke and J.-F. Cordeau, "Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows," *Transportation Science*, vol. 43, no. 3, pp. 267–286, aug 2009. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/trsc.1090.0272