Heuristics for a school bus routing problem

Tutors:

Gregorio Tirado

Javier Martín

Students:

Marta Artalejo

Javier Gómez

Ignacio Leguey

Ana Mateos Jaime

Giulia Palagi

Miguel Ángel Rodríguez

Content

1. Introduction

The problem approached during the Modelling Week, proposed by Goal Systems Company, arises from the need to develop a method to deal with the transportation of students to the university by bus, including the selection of the stops and the design of the bus lines.

This problem, known as School Bus Routing Problem (SBRP), seeks to plan an efficient schedule for a fleet of buses, picking up the students from the different bus stops and delivering them to their schools. Several constraints related to the time window to arrive at school, the maximum distance that a person can walk to reach the bus stop and the maximum bus capacity must be verified.

Desrosiers et al. (1981) solved the problem in five steps: data preparation, bus stop selection (student assignment to stops), bus route generation, school bell time adjustment, and route scheduling. In most existing approaches, although they are not independent, these sub-problems are considered separately and sequentially due to the complexity and size of the problem. The bus stop selection problem is often omitted in the literature and many studies assume that the locations of bus stops are given.

The solution approach for SBRP may differ depending on whether the surroundings of the service are urban or rural. It is assumed that students in urban areas walk from their homes to the stops to take a bus. However, in rural areas the number of students is small, and it is common to pick them up at their homes. Therefore, bus stop selection is not necessary for rural surroundings. Many researchers have pointed out the difference between the surroundings of urban and rural school bus systems.

The bus route generation sub-problem is very similar to the Vehicle Routing Problem (VRP), which is an extensively studied application of operations research. VRP seeks to generate efficient routes for a fleet of vehicles in order to deliver (or collect) products from depots to a set of customers (Toth and Vigo, 2002).

The combined problem of bus stop selection and bus route generation falls into the class of location-routing problems (LRPs). LRP includes determining the location of the facilities (in SBRP, bus stops) serving more than one customer and the optimal set of routes for a fleet of vehicles (Min et al., 1998; Laporte, 1988; Nagy and Salhi, 2007).

In most studies, the starting and ending time of schools are constraints. However, there are many works that consider the times as decision variables and attempt to find the optimal starting and ending times in order to maximize the number of routes that can be served sequentially by the same bus and to reduce the number of buses used.

The problem could also deal with a heterogeneous fleet of buses, which assumes that each bus has different characteristics such as capacity, maximum allowable riding time, fixed cost, and per unit distance variable cost. The problem with a heterogeneous fleet of buses is similar to the heterogeneous fleet VRP (HVRP), a variant of traditional VRP. The work of Newton and

Thomas (1974) was the first in assuming that all buses have the same capacity. Bowerman et al. (1995) also studied a problem on routing a set of buses having the same capacity.

SBRP can be decomposed in two subproblems, regarding the morning and the afternoon. In the morning, a bus picks up students from bus stops and then drops them off at their school. On the contrary, students are picked up at schools and delivered to their starting stops in the afternoon. Braca et al. (1997) claimed that the morning problem is more difficult than the afternoon one due to a couple of reasons: ill-dispersed school time windows and heavier traffic. As a result, the afternoon problem receives less attention than the morning one. Many studies are devoted to the morning problem and the afternoon problem is only mentioned briefly. Notice that the afternoon school bus problem for a school can be converted into a morning problem with little modifications (Li and Fu, 2002).

For a recent survey on the school bus routing problem we refer the reader to Park and Kim (2010).

In our case, as real data were not available, we had to deal with the generation of instances to test the solution methods developed. Then, we have worked on three interrelated different problems: the generation of the instance data, the location of the bus stops and the assignment to the students and, finally, the design of a good set of routes for the buses.

The rest of the report is organized as follows. In Section 2 the problem considered is described with detail. In Section 3 the generation of data is explained, while the selection and assignment of bus stops is approached in Section 4 and the routing problem is detailed in Section 5. The final section comprises the results obtained, possible improvements and extensions.

2. Description of the problem

The size of the original problem is 7500 students and 500 teachers for 10 schools/universities. Due to the lack of time and the available tools we have considered a reduced problem: 800 students and one school/university; from now on university. However, if we consider that each bus is assigned to one university, we can solve the original problem considering each university and the students going there as independent problems.

The problem is defined on a network composed of nodes representing the students' homes, the potential bus stops that could be used and the location of the university. The students will walk from their houses to one of the stops, where they will be picked up by a bus and dropped off at the university. The problem consists of designing a set of bus routes to ensure that every student can reach a bus stop within a walking distance, and arrive at the university on time.

There are several important constraints that must be verified:

- Maximum walking distance (d_{max}): the longest distance that a student is allowed to walk to arrive at a bus stop.
- Maximum duration of each route (t_{max}): the longest time that a bus is allowed to spend to arrive at the university starting from the headboard of a route. This is mostly related to the maximum time that any student can spend in the bus before reaching the university.
- Maximum capacity of each bus (c_{max}): every bus has an assigned payload i.e. it can contain a certain number of students that we cannot exceed due to safety reasons.

The following constants were also used:

- Average speed of the buses (v).
- Fixed costs for using one bus (f): cost per bus.
- Costs related to the covered distances (c): cost per kilometer.

Any solution will be given by a matrix that contains in each row the sequence of stops followed by each bus. The solution also returns the timetable of the routes, the number of the students picked up at each stop of the different routes and the distances between two consecutive bus stops at each route.

Besides, it is reasonable to assume that all buses have the same capacity, and that all students must be picked up to have a final feasible solution. Under all these conditions, the goal is to find the optimal bus routes attempting to reduce the total cost, which is the sum of fixed and variable costs for the use of the buses. Therefore, the definition of the objective function is as follows:

$$f_{obj} = f \cdot NB + \sum_{i=1}^{NR} c \cdot KR(i)$$

Where NB is the number of the buses, NR is the number of the routes and KR(i) is the number of kilometers of the route (i).

3. Data generation

To generate the **home locations** (x_i, y_i)we simulate them under a uniform distribution, prefixing a rectangle (a, b) x(c, d) and generating them randomly inside it. The **university location** (x_0, y_0) was also randomly created in the same rectangle (a, b) x(c, d), following also a uniform distribution.

For the generation of the **potential bus stops**(α_i, β_i)**,**we created a squared grid on the same rectangle, where the diagonal length of each square is equal to the maximum walking distance d_{max}. This is done in this way so that any student whose home is inside one small square of the grid could walk to any bus stop located inside that square. Once this is done, one stop is generated uniformly on each square of the grid, ensuring that each student can walk to at least one stop.

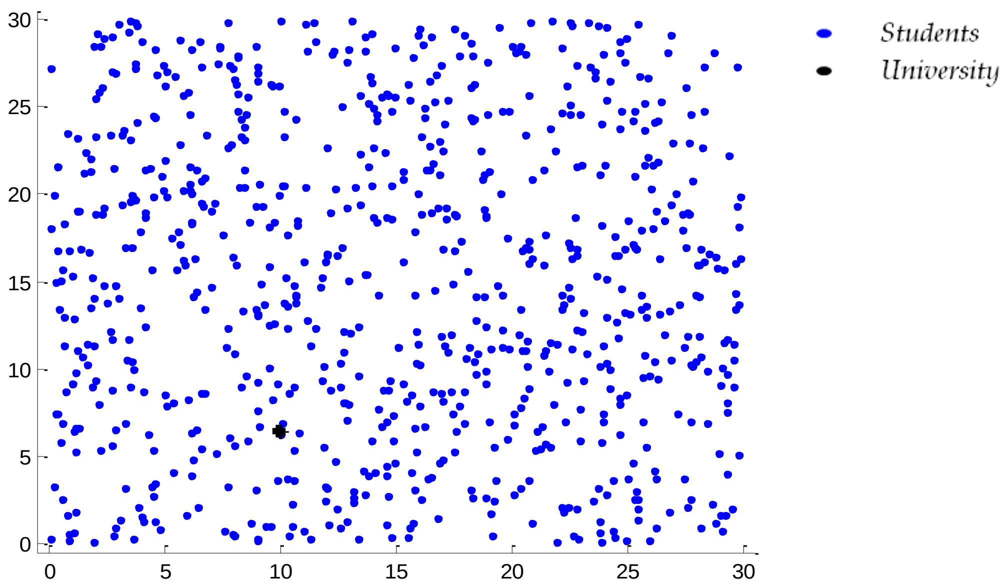

Figure 1: Home locations and University

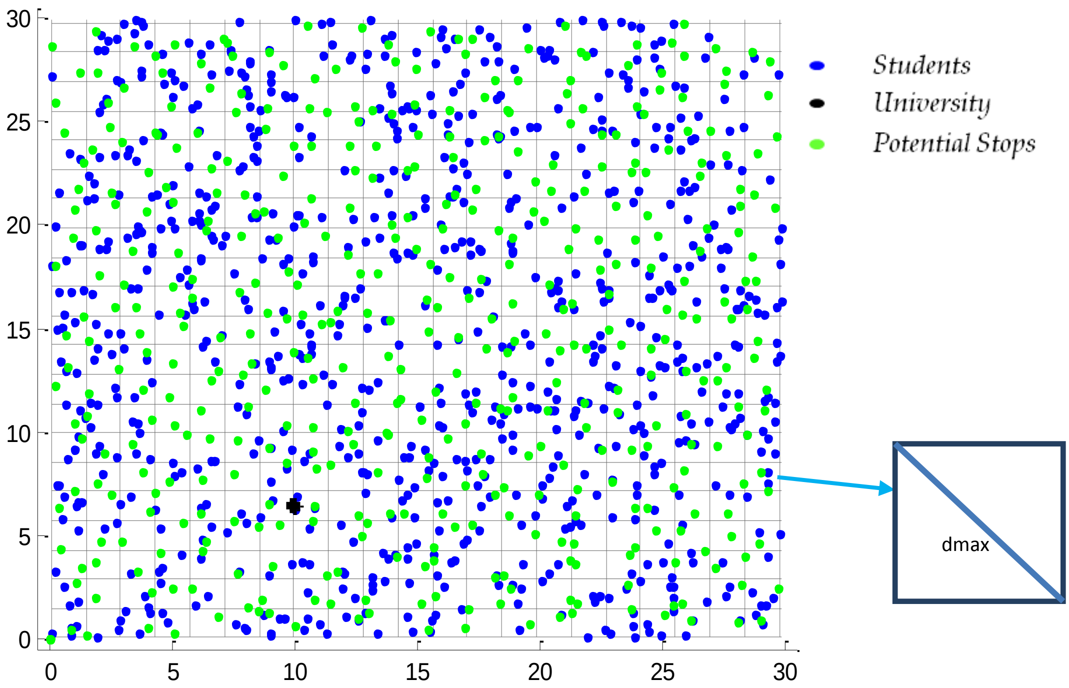

Figure 2: Grid to generate the potential bus stops

4. Selection and assignment of bus stops

As a first step, we have to assign students to the stops. To tackle this problem, we consider the maximum walking distance and the circle whose center is the bus stop and its radius the maximum walking distance. People who live inside the circle are assigned to that stop.

Then we select the bus stops with the maximum number of students assigned to it. Since we could have more students than the capacity of the stop, we sort the students according to their appearance in other stops; and select the stop with the highest number of students.

Once we have found the stop, we include it in the set of selected ones and as we might have the same person assigned to more than one stop we delete those students from the rest of stops.

The process is iterated until we have assigned all the students to the stops.

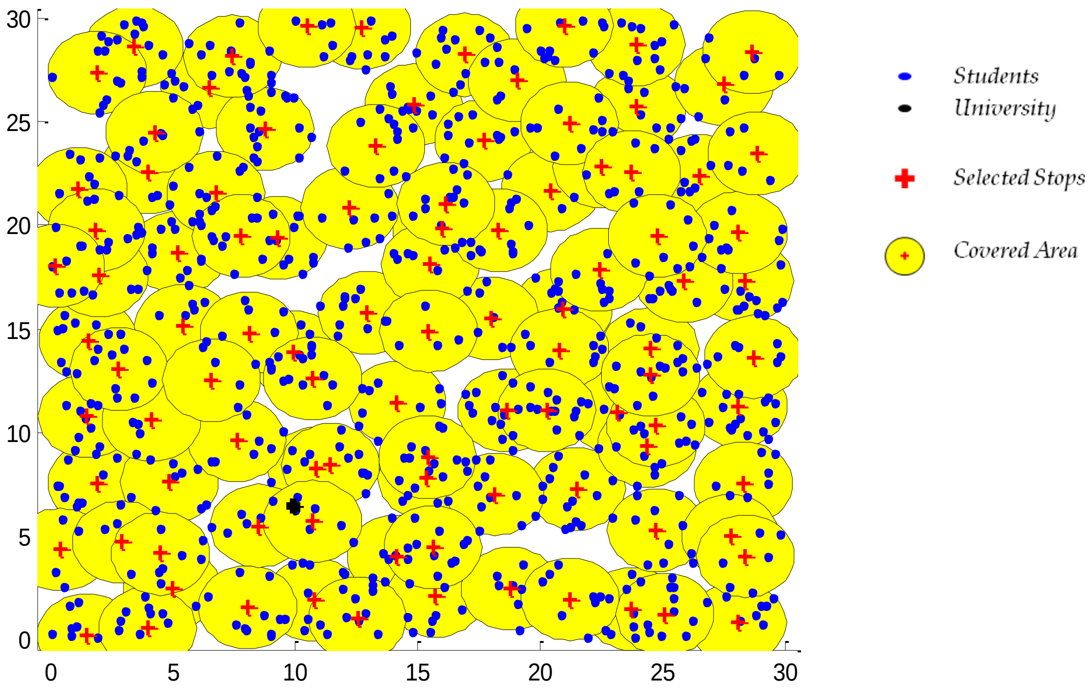

Figure 3: Area covered by the stops

In this process we can check the existence of compulsory stops. A bus stop is defined as compulsory if and only if at least one of the students can reach only that stop.

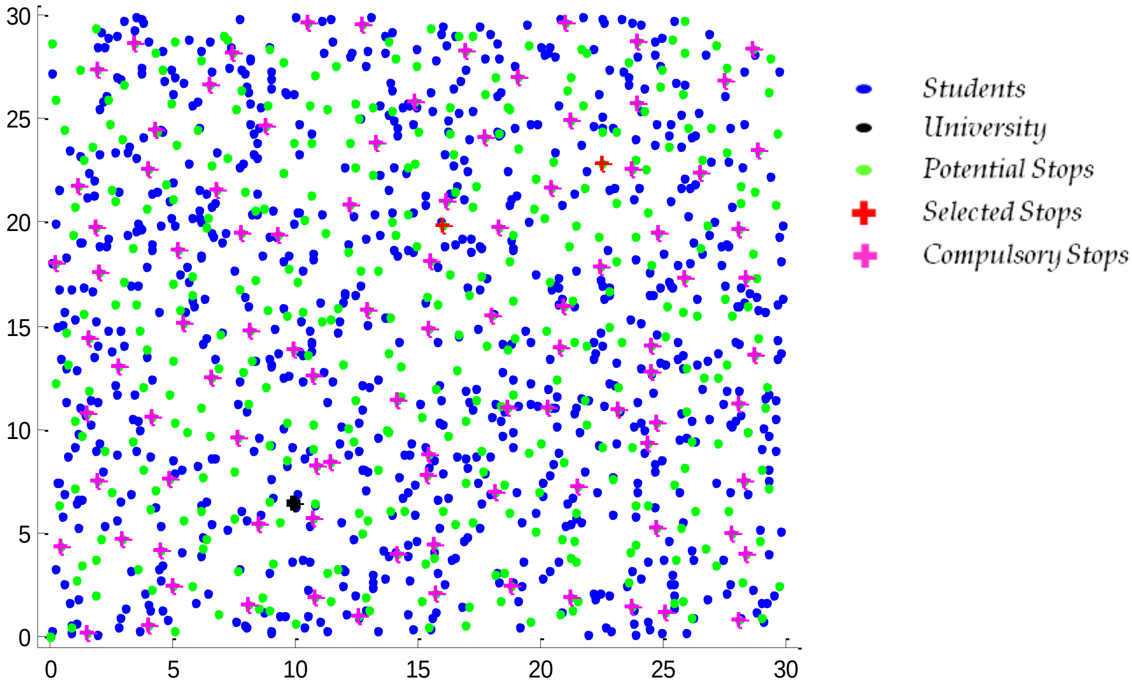

Figure 4: Selected and compulsory stops

5. Routing

This part of the approach is focused on the routes generation. A constructive algorithm and three local search procedures are presented in the following.

5.1 Initial routing generation

The constructive method is aimed to generate an initial feasible solution to the routing problem in order to improve it in the following steps of the work. The main idea is to build a route by connecting some of the stops that have not been already visited. For this purpose, we consider the distance between them, the capacity of the bus and the maximum time that a bus would need to arrive at the university as criteria to decide how to connect the nodes. From the current stop we select the closest stop verifying the capacity and maximum duration constraints to continue creating the route, but only if it is possible to pick up all the students waiting there. If the number of students assigned to one stop is greater that the capacity of the bus, we create a route only visiting that stop with a bus that leaves completely full.

We create the routes starting from the university and reaching as many stops as possible, taking into account the bus capacity and the remaining time, so as to minimize the number of buses needed.

In order to select the stop at each step we sort all of them from closest to farthest, and start considering the first available node of the list (i.e. the closest one), picking up as many people as possible and updating the remaining time and the capacity of the bus. Once this is done, we continue with the choice of the next stop if there is time and capacity remaining.

When we cannot select any other nodes from the one that we are visiting (in the case we do not have enough time to reach any other stop or if the bus is full), we close the route and start creating a new one by using the same method. If we have already visited all the stops and picked up all the customers, the solution is finished.

In the figure 5 we present the corresponding pseudocode.

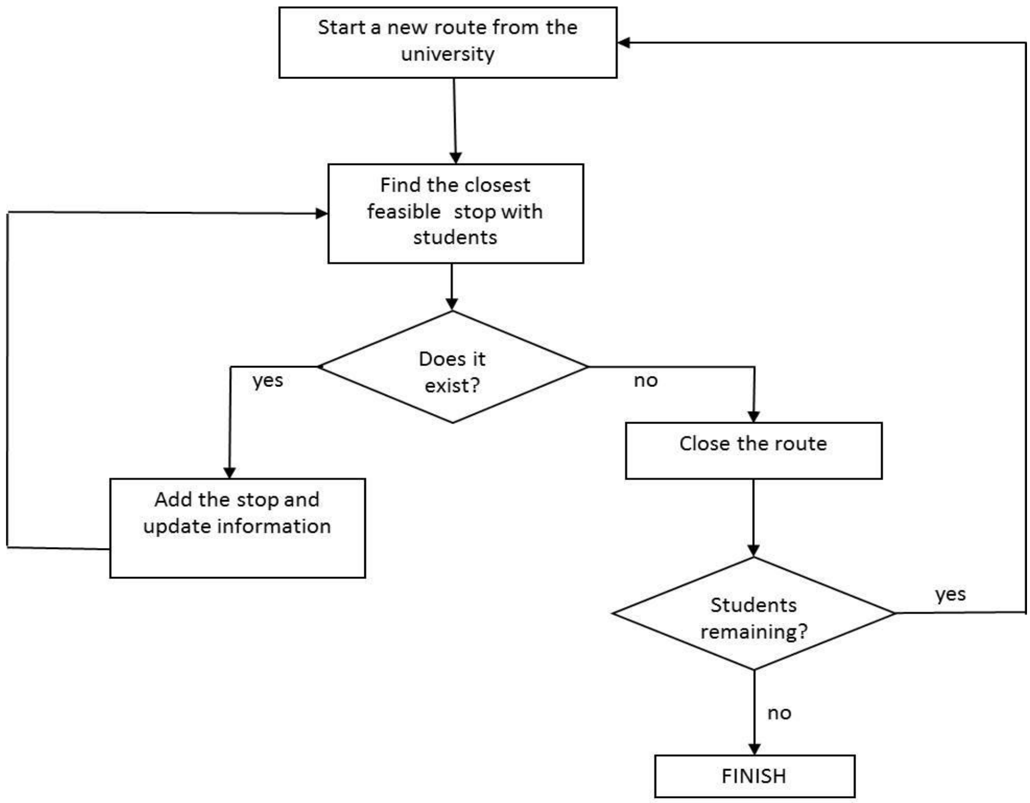

Figure 5:pseudocode

The figure 6 gives an example of the initial solution created by the constructive algorithm for one instance with 800 students.

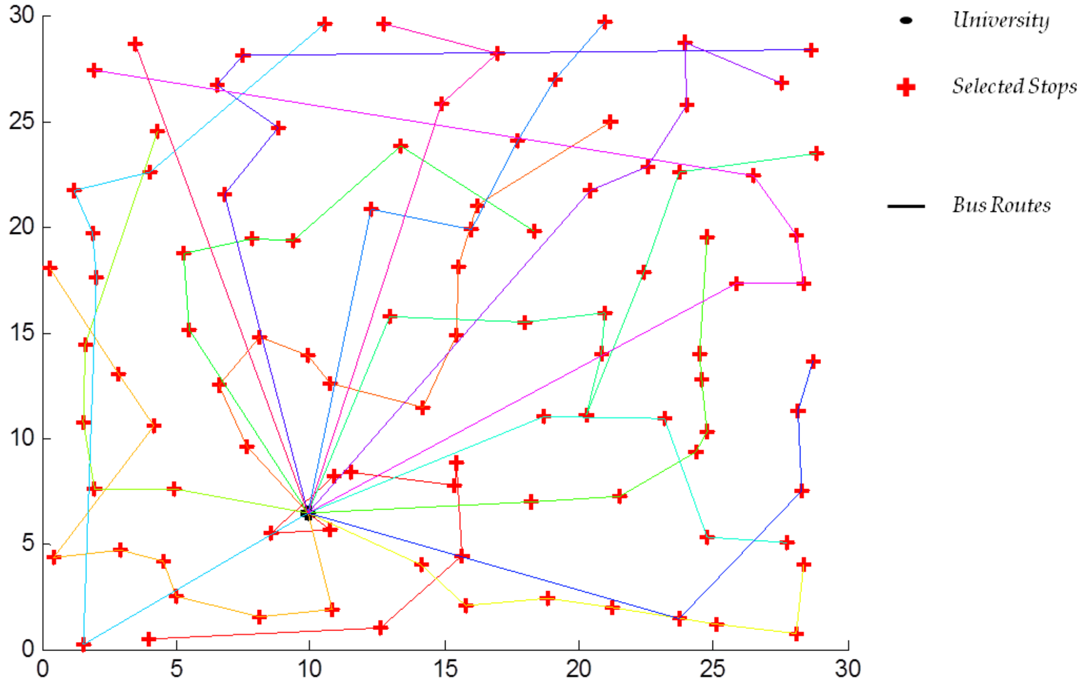

Figure 6: Initial solution

5.2 Local search

As the initial solution obtained is not expected to be good, we try to improve it with different local search methods, as explained in what follows.

5.2.1 Exchanges in the same route

With this method we try to improve the initial solution by exchanging stops from the same route. We start with the first route and check all the possible moves. We evaluate each move and calculate the associated costs. Then we choose the best cost, that means the lowest one, and we keep that solution and start again in the same route. If none of the costs obtained is better than the one we already have, we change to the next route. The process finishes when we have checked all the routes.

In the figure 7a the initial route is drawn, whereas the figure 7b presents the route after the algorithm is applied (old arcs are drawn using dashed lines).

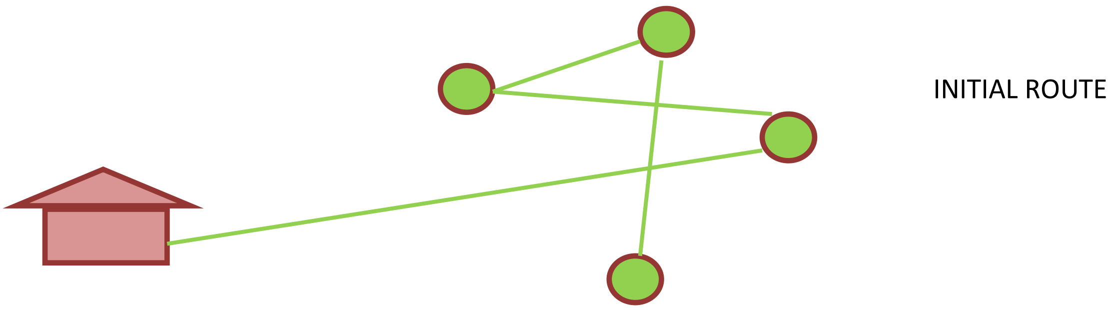

Figure 7a: Initial route

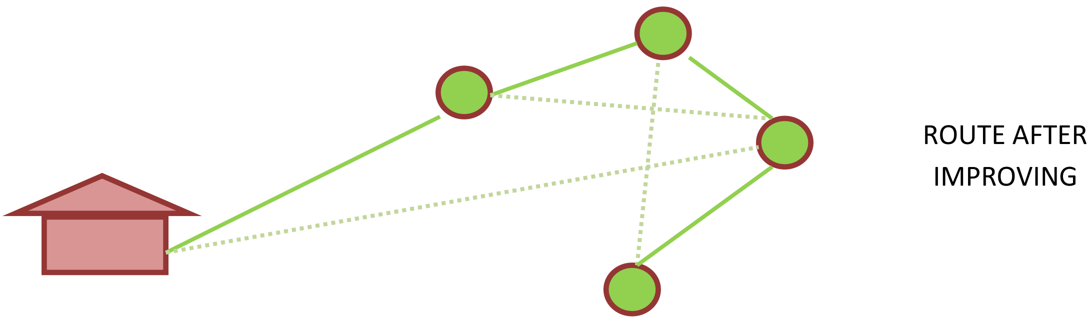

Figure 7b: Route After Removing

The figure 8 shows the corresponding pseudocode.

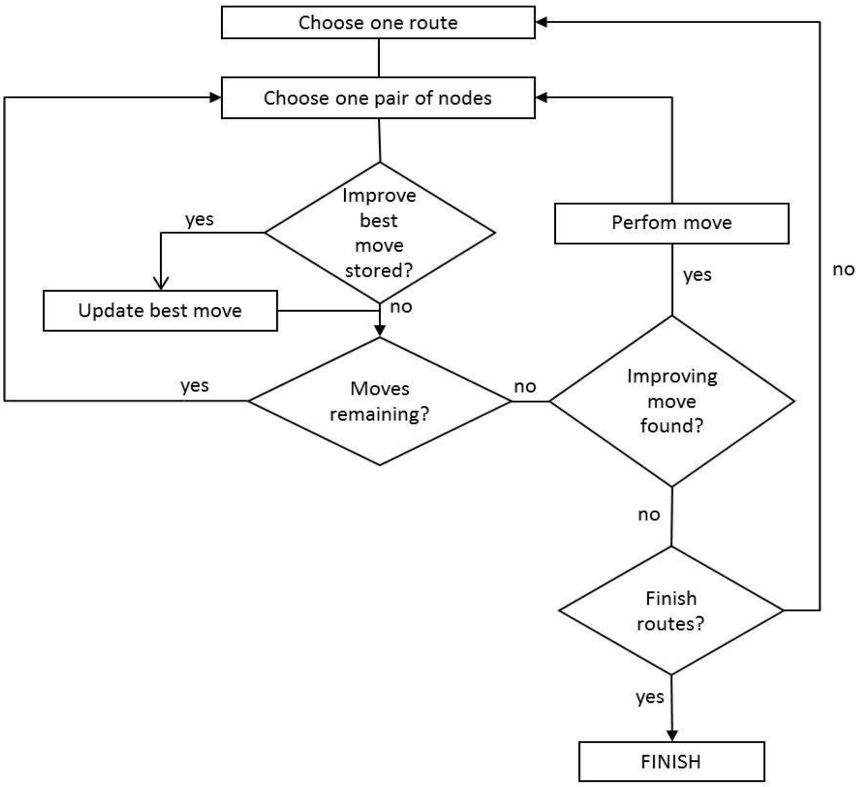

Figure 8: Pseudocode

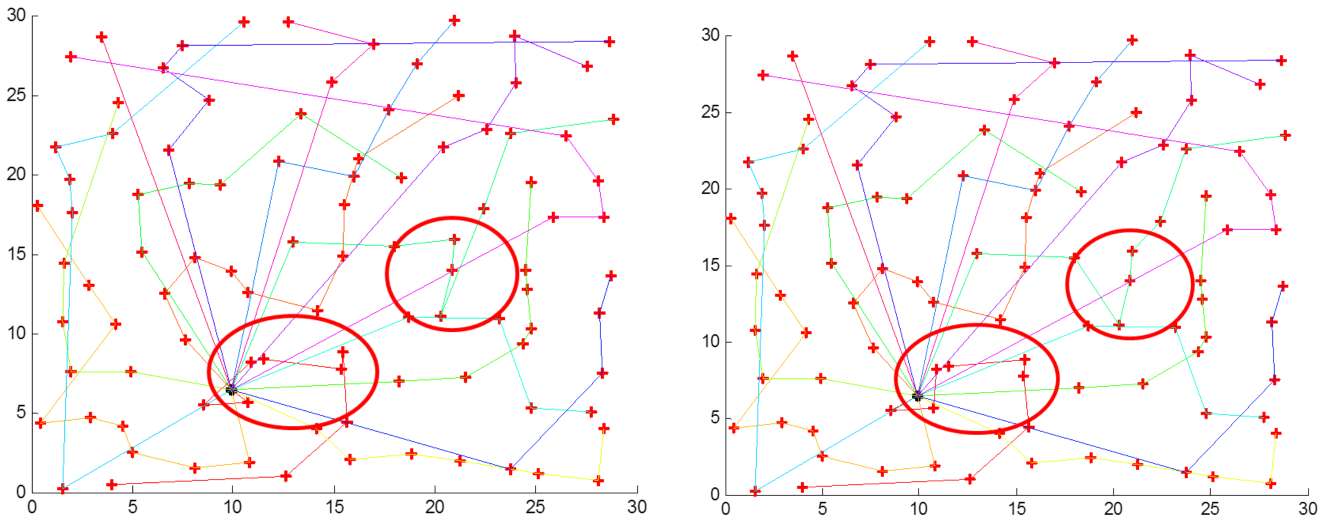

Figure 9: Exchanges in the same route

The figure 9 shows an example of the effects of applying the method explained. The left picture presents the original solution and the right one shows the changes. With this method we avoid complex routes, reducing the covered distance.

5.2.2 Exchanges between two routes

The previous method worked with only one route, whereas the next two methods try to improve the solution by mixing different routes.

5.2.2.1 Exchange of two nodes

This first method is intended to improve the routes by exchanging nodes from different routes. We initially select two different routes and one bus stop in each route randomly, as shown in the figure 10.

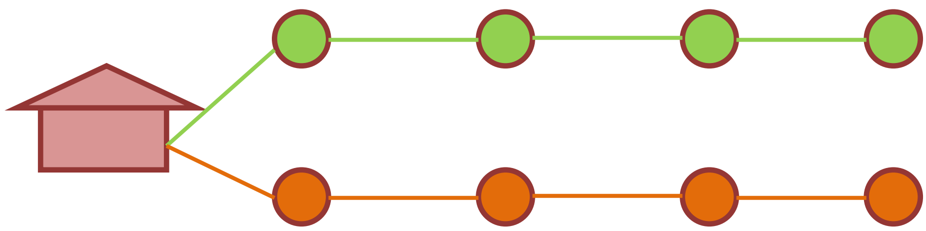

Figure 10

This can produce three different situations for the algorithm, as follows. In each figure, the arcs removed are indicated with a red cross and the new arcs reconnecting the routes are drawn using dashed lines.

- One or both nodes selected are the last stop of the route before arriving to the university.

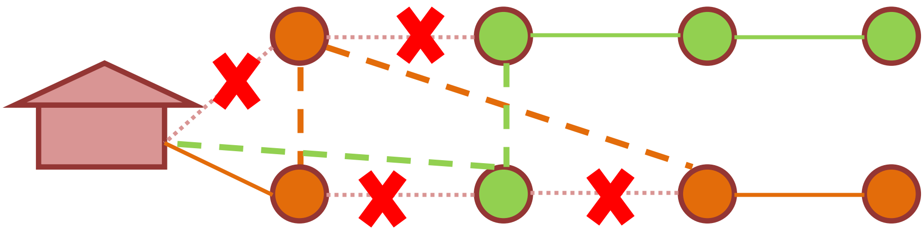

Figure 11a

- One or both nodes are the headboard of the route.

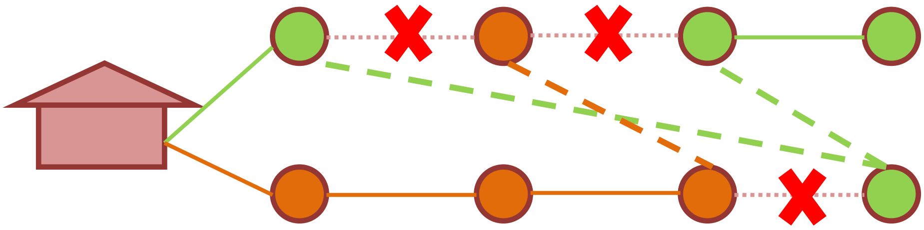

Figure 11b

- One or both nodes are in a different position as the described above.

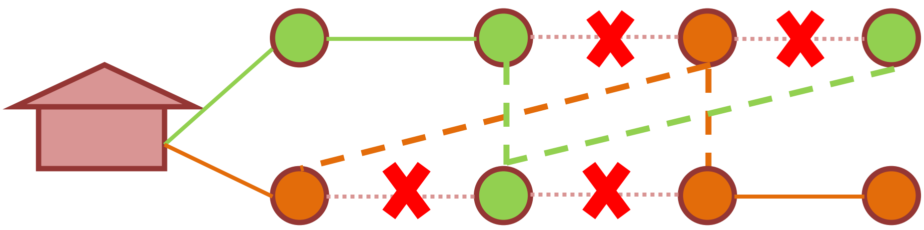

Figure 11c

We check that the constraints of maximum capacity and maximum distance for each route are satisfied after the exchange between nodes. If the constraints are satisfied and the objective function is improved, we keep the changes. In other case the move is discarded and the process is iterated from the beginning for a certain number of iterations.

The pseudocode for the process is illustrated in the figure 12.

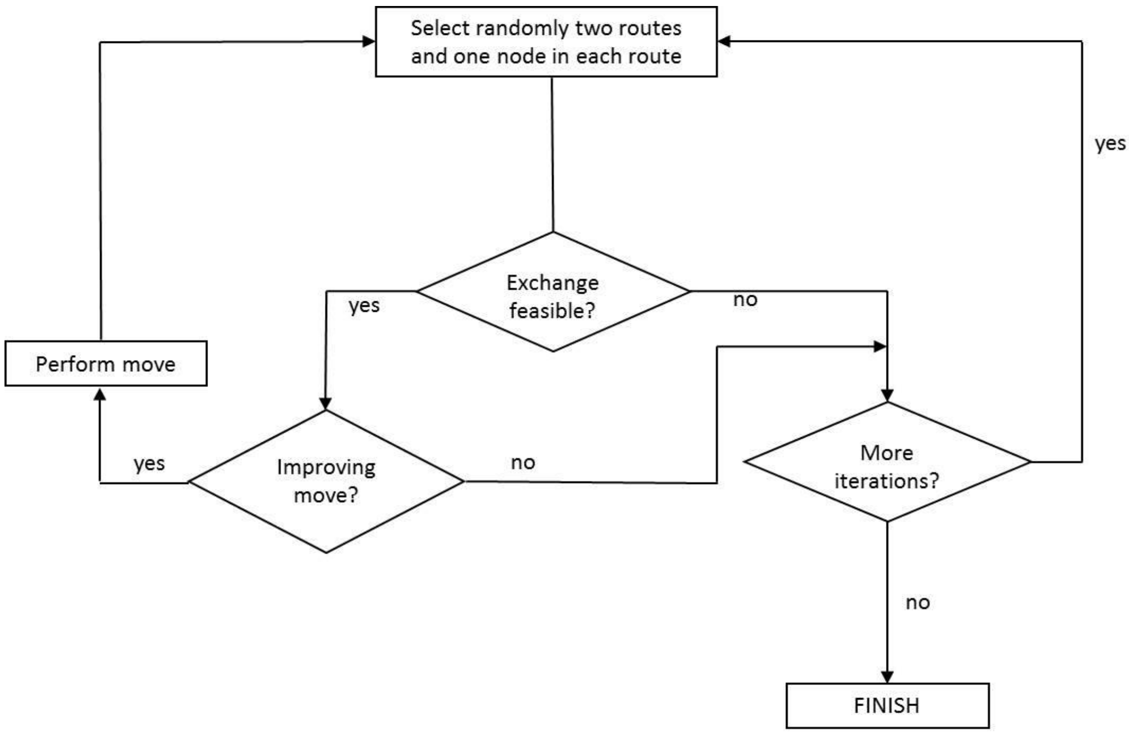

Figure 12: Pseudocode

In the figure 13 there is an example of the changes that appear after the method explained is applied. The left picture presents the original solution and the right one shows the changes. As we can see, most of the longest arcs are replaced by shorter ones.

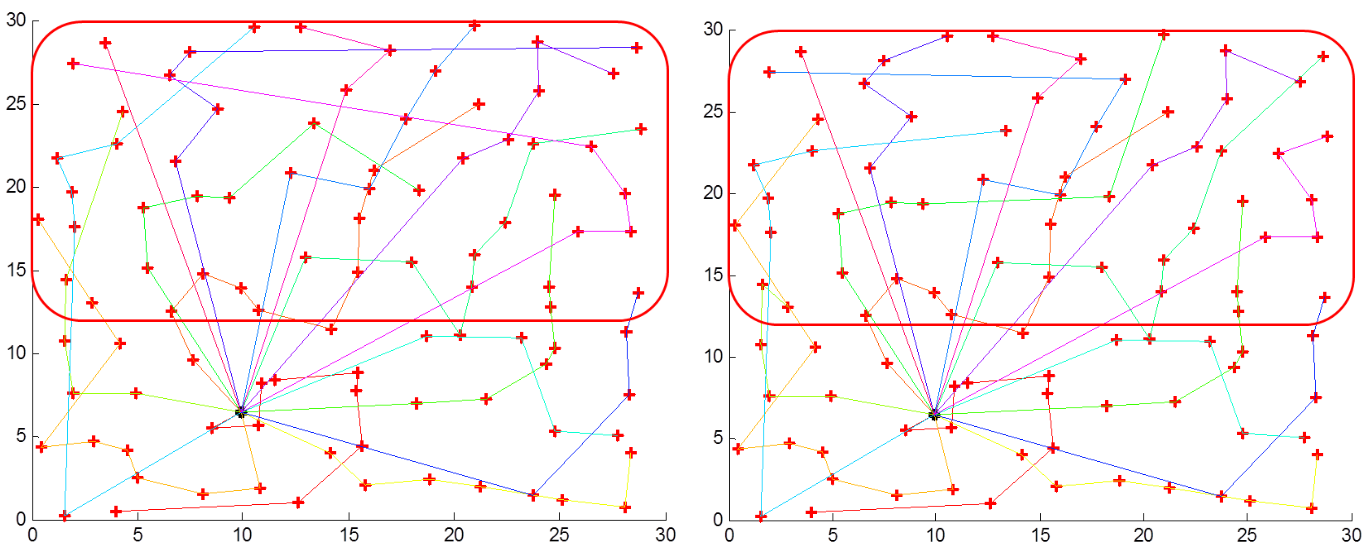

Figure 13: Exchange of two nodes

1.2.2.2 Exchange of part of the routes

In the previous local search method we focused on the nodes. However, in this local search we work with arcs in order to try to improve the solution of the problem. The objective consists of combining two routes by deleting an arc of each one and joining the remaining pieces of the routes.

To do so, we choose randomly one route and one arc in that route, discarding the last one reaching the University. We considered choosing directly the longest arc, in order to try to achieve largest savings, but this restricted the search too much and many other potentially good moves were not considered.

Once the first arc to be removed is determined, we have to calculate the duration and the number of students picked up on each piece of the considered route. It is important to keep under control the times and the number of people getting on the bus in each stop in order to hold feasibility.

After that, we check in the other routes with at least 2 arcs, if it is possible to mix each piece of the first route with one of the pieces of the second route by deleting again one arc. This move is illustrated in the figure 14a.

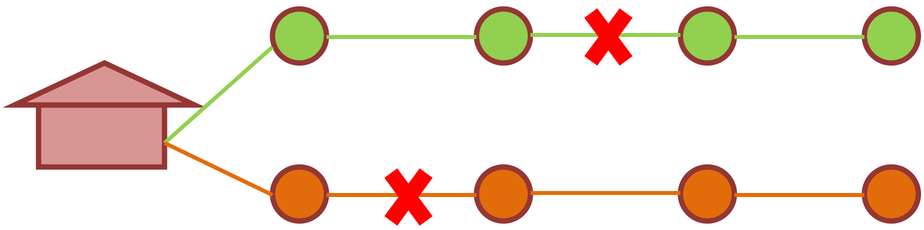

Figure 14a

The initial routes are drawn in green and orange and the selected arcs are marked with a cross. After removing the selected arcs and rejoining the routes we obtain the figure 14b.

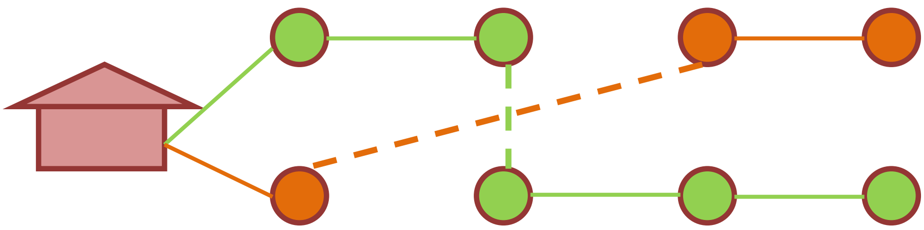

Figure 14b

We only choose moves that verify all capacity and time constraints, so that no student arrives late to the University and the number of people in a bus does not exceed its capacity.

If the move is feasible, we calculate the costs associated to the new solution. If the total cost is reduced, it means that we have obtained a better solution; in that case, we implement the move and save this new solution. On the other hand, if the cost is increased, the move is discarded.

After all possible arcs are considered for the second route, the process is iterated from the beginning for a certain number of iterations.

The pseudocode for the process is illustrated in the figure 15.

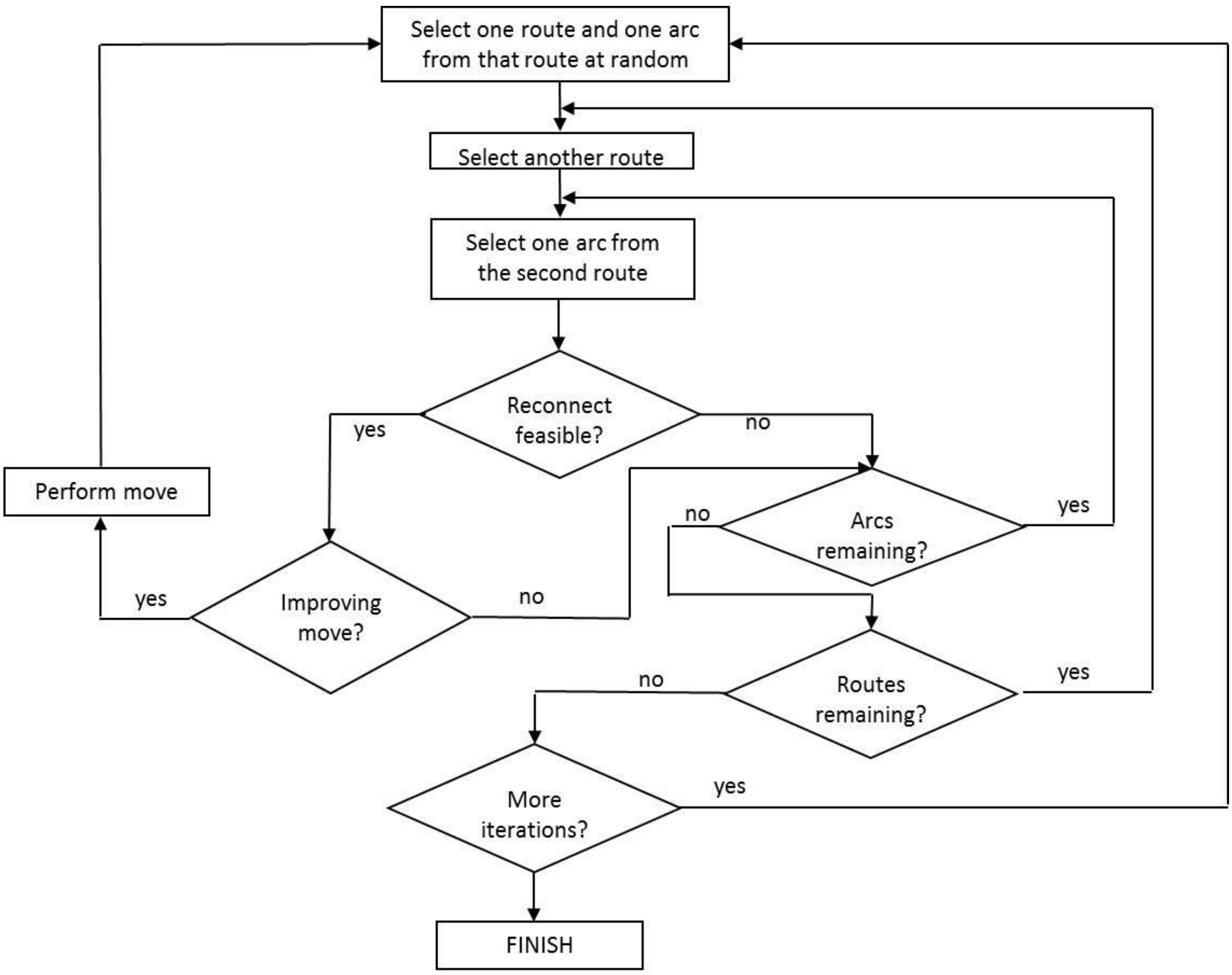

Figure 15: Pseudocode

The figure 16 shows an example of how the local search improves some routes. This method allows replacing long and unnecessary arcs with shorter and more useful ones.

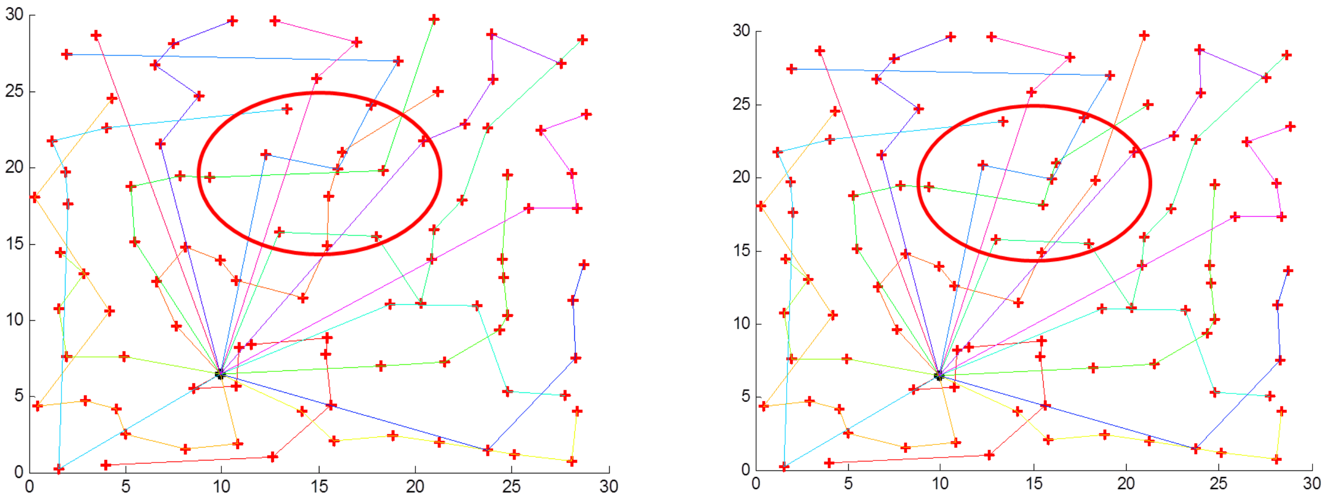

Figure 16: Exchange of part of the routes

6. Results

With the purpose of getting the best possible solution of the routing, we have applied iteratively the three methods presented.

As a first step, we have exchanged nodes in the same route, followed by a node exchange in different routes. Doing that, we have eliminated most of the longest arcs and replaced them with shorter ones. However, these routes might go over the same place so we have applied again the node exchange in the same route, which is the fastest method of the three local search processes.

After that we have mixed different routes with the arcs exchange method, as mentioned before, to get even shorter routes. Finally we have used the node exchange in the same route method.

This process is iterated for a certain number of iterations, after which the algorithm stops, outputting the best solution found so far. The general pseudocode of the algorithm is shown in the figure 17.

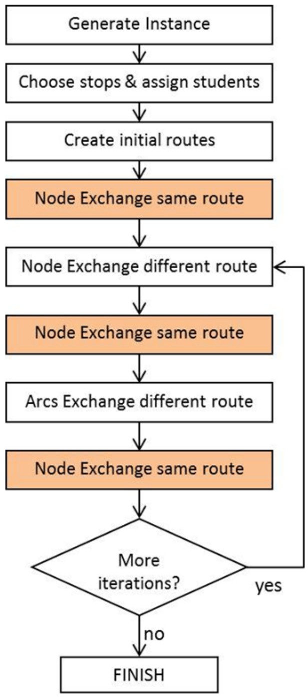

Figure 17: General pseudocode

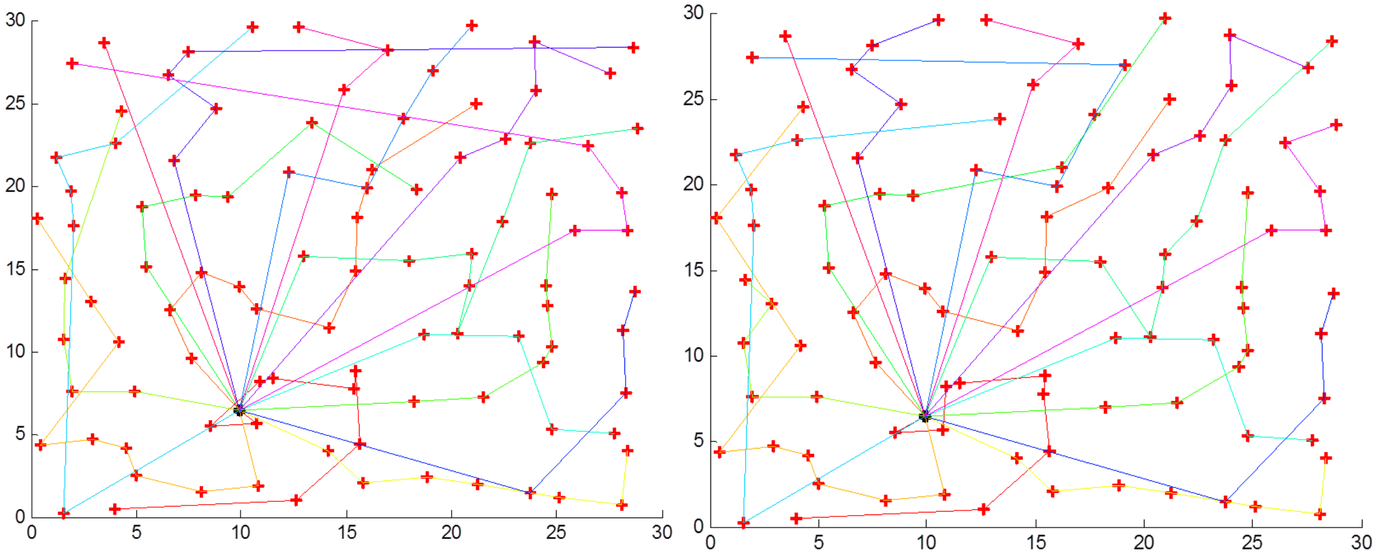

Figure 18: Initial and Final Solution

The figure 18 shows an example of the result of applying the general pseudocode (right figure)to the initial solution (left figure). We have achieved a reduction of 5% on total costs.

7. Improvements and extensions.

The solution methods proposed in this work could be improved in several ways, and the problem considered could also be extended to better represent the real life situation; here we are presenting some ideas which could be useful for these purposes.

7.1 Possible improvements

- Implement a metaheuristic method based on the proposed local search procedures, such as simulated annealing, tabu search or a genetic algorithm, in order to escape from local optima and obtain better solutions.
- Try to use the final routing information to improve the selection of stops and the assignment of students to them. For example, once we know some initial routes it would be possible to add more stops in the areas with a high number of routes, increase the area of influence of a bus stop in areas with a low density or vary the maximum walking distance.

- Check how the algorithm behaves with different population distributions. We have only worked with a uniform distribution, but Normal and Poisson distributions, among others, could be chosen.

7.2 Possible extensions

- One of our hypothesis was that the fleet of buses was homogeneous, which is far away from real-life problems. Areas with a low number of students can be covered with minibuses and highly populated areas with longer buses.
- We have reduced the problem to only one school/university. So, another possible extension is considering several schools/universities, and the same bus assigned to different universities.
- Considering different shifts, people may want to arrive to the school/university in the morning, at noon or in the afternoon.
- Taking into account that more than one person comes from the same home and they have to travel together.

8. References

Bowerman, R., Hall, B., Calamai, P., 1995. A multi-objective optimization approach to urban school bus routing: formulation and solution method. Transportation Research Part A 29 (2), 107–123.

Braca, J., Bramel, J., Posner, B., Simchi-Levi, D., 1997. A computerized approach to the New York City school bus routing problem.IIE Transactions 29, 693–702.

Desrosiers, J., Ferland, J.A., Rousseau, J.-M., Lapalme, G., Chapleau, L., 1981. An overview of a school busing system. In: Jaiswal, N.K. (Ed.), Scientific Management of Transport Systems. North-Holland, Amsterdam, pp. 235–243.

Laporte, G., 1988. Location-routing problems. In: Golden, B.L., Assad, A.A. (Eds.),Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 163–198.

Li, L., Fu, Z., 2002. The school bus routing problem: a case study. Journal of the Operational Research Society 53, 552–558.

Min, H., Jayaraman, V., Srivastava, R., 1998. Combined location-routing problems: synthesis and future research directions. European Journal of Operational Research 108 (1), 1–15.

Nagy, G., Salhi, S., 2007. Location-routing: issues, models and methods. European Journal of Operational Research 177 (2), 649–672.

Newton, R.M., Thomas, W.H., 1974. Bus routing in a multi-school system. Computers and Operations Research 1 (2), 213–222.

Park, J., Kim, B.,2009. The school bus routing problem: A review. European Journal of Operational Research 202 (2010) 311–319

Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. SIAM, Philadelphia, PA.