View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Aston Publications Explorer

A Strategy for Clustering Students Minimizing the Number of Bus Stops for Solving the School Bus Routing Problem

João F. M. Sarubbi*, Caio M. R. Mesquita*, Elizabeth F. Wanner*, Vinicius F. Santos* and Cristiano M. Silva†
*Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte-MG, Brazil
Emails: joao@decom.cefetmg.br, caioboninho@gmail.com, efwanner@decom.cefetmg.br, vinicius.santos@gmail.com
†Departamento de Tecnologia
Universidade Federal de São João Del Rey, Ouro Branco-MG, Brazil
Email: cristiano@ufsj.edu.br

***Abstract*—In this work we tackle the bus stop selection step for the School Bus Routing Problem (SBRP). Our goal is to minimize the number of bus stops in order to assign all students to a bus stop respecting a home-to-bus-stop walking distance constraint. Our strategy creates a large number of possible bus stops points in a road network and uses a pseudo-random constructive heuristic algorithm to assign students to a bus stops. Our approach is tested on a real georeferenced data of a brazilian city and is compared with a different methodology. Results demonstrate that the proposed approach is able to find good solutions for this optimization problem. Besides,the higher the number of possible points to install bus stops, the smaller is the number of bus stops required to attend all students.**

***Index Terms*—School Bus Routing Problem; Bus Stop Selection; Constructive Greedy Heuristic;**

I. Introduction

The School Bus Routing Problem (SBRP) [1]–[5] is a classical combinatorial optimization problem which, given a set of roads, schools, students, vehicles and garages, consists of generating an efficient schedule for a fleet of school buses. Each bus picks up students from various bus stops and delivers them to their designated schools while satisfying various constraints such as the maximum capacity of a bus, the maximum riding time of a student in a bus and the time window of a school. According to [2] the SBRP can be divided into five sub-problems: *Data Preparation*; *Bus Stop Selection*;*Routes Generation*; *School Bell Time Adjustment* and *Route Scheduling*. As far as we are concerned, only Desrosiers et al. [2,6] address all five steps of the SBRP. Normally, the research works deal with the *Route Generation* and the *Route Scheduling* steps.

In this work we will address a different version of this problem dealing with just with the *Data Preparation* and the *Bus Stop Selection* steps. The *Data Preparation* step [7] consists of generating a single network containing students, residences, schools and garages. The *Bus Stop Selection* step considers the road network and the students location in order to determine the location of the bus stops. Some constraints may also be employed, such as a maximum home-to-bus-stop distance [8]. However, the *Bus Stop Selection* step is often omitted in literature because most studies assume that the locations of bus stops are known before hand.

Our goal is to minimize the number of bus stops in a road network respecting a maximal home-to-bus-stop distance constraint. With less bus stops it can be easier to the planning authorities to choose the best bus routes using any software or even manually. For instance, if we are able to assign all students within a same bus stop, no computation is need to find the best routes. Instead of choosing bus stops just in the road intersections as Faraj et al. [9], our strategy discretizes the network points enabling the creation of bus stops in a large number of points. Furthermore, we use a pseudo-random algorithm based on a GRASP [10] Construction Phase to tackle the problem.

Our results demonstrate that our approach was able to find good solutions for this optimization problem. Increasing the number of possible points to install bus stops, we are able to decrease substantially the number of bus stops to attend all students.

In our study we have used a real georeferenced data of a brazilian city and our sample consists of 944 students. Figure 1 presents the road network of the city being studied. The blue points represent the student's homes.

This work is organized as follows: Section II presents the definition of the problem; Section III presents the related work; Section IV presents our baseline algorithm; Section V presents our proposed algorithm; Section VI presents our results; and Section VII concludes our work.

II. Definition of the Problem

Given a network N composed of a set of roads R, $R = \{1, 2, ..., r\}$, a set of students S, $S = \{1, 2, ..., i\}$, a set of equally spaced points P, $P = \{1, 2, ..., j\}$ for installing bus stops and a maximal home-to-bus-stop walking distance λ, our goal is to find the minimal set of bus stops B, $B \subseteq P$, which is able to attend all students. We also need to define a

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

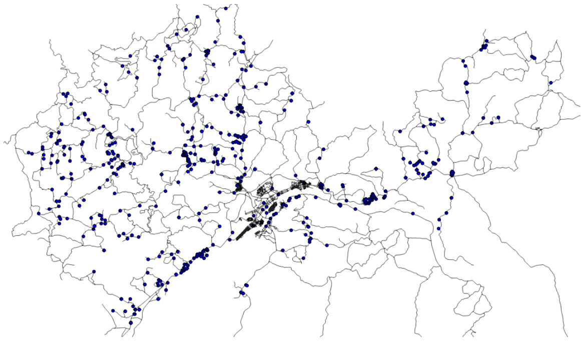

Figure 1. Roads and Students

d_{ij} parameter that represents the distance between the student i and the possible bus stop j. It is important to notice that for this case, since we are dealing with georefenced data, the distance between the students and the bus stops should be calculated using the network roads and is different from the Euclidian distance.

Figure 2 shows an example of assignment of students to bus stops: circles represent stops, squares represent students and the color represents 'assignment': students assigned to a given stop share the same color of the stop.

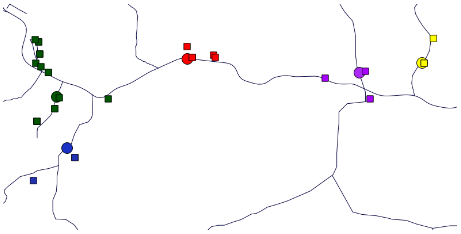

Figure 2. Example of an assignment of students to bus stops

For this problem, an Integer Linear Programming (ILP) formulation M with the following sets of variables is defined:

$$y_j = \begin{cases} 1, & \text{if point } j \text{ is chosen to receive a bus stop} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if student } i \text{ is assigned to bus stop } j \\ 0, & \text{otherwise.} \end{cases}$$

and the following set of parameters can be also defined:

- d_{ij}: distance between student i and point j.
- λ: maximal home-to-bus-stop walking distance.

In order to model this problem we must create the following set:

$$\Theta = \{(i,j) \in S \times P \mid d_{ij} \leq \lambda\}$$

The Θ set avoids creating variables x_{ij} in which the distance between student i and possible bus stop j are less than λ.

The bus stop allocation is modeled as follows:

$$\min \sum_{j=1}^{|P|} y_j \qquad (1)$$

subject to:

$$\sum_{(i,j)\in\Theta} x_{ij} = 1 \qquad \forall i \in S \qquad (2)$$

$$x_{ij} \leq y_j \qquad \forall i \in S, j \in P \mid (i,j) \in \Theta \qquad (3)$$

$$y_j \in \{0,1\} \qquad \forall j \in P \qquad (4)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \Theta \qquad (5)$$

Objective function (1) minimizes the number of select points to install bus stops. Constraint (2) ensures that every students will be assignment to just one bus stop. Constraint (3) ensures that, if one point is chosen to receive a bus stop, at least one student will be assignment to this bus stop, and, if one student is assignment to a point, this point must be a bus stop. Constraints (4) and (5) are the integrality constraints.

III. Related work

After the work of Bodin and Berman [11], only few works deal with the bus stop selection step of the SBRP. Although the bus stop selection and the bus route generation are highly interrelated, in most of previous works both problems are considered independently. The combination of the two steps has already been approached from some location-allocation-routing (LAR) heuristics [2,11,12] and allocation-routing-location (ARL) [13,14] strategies.

More recently, Schittekat et al. [5] present a GRASP coupled with a VND heuristic for the SBRP with only one school. The authors aim to minimize the total travel distance of all school buses determining the set of bus stops, the best assignment of students to a bus stop and the routes that lie along the chosen stops. Different from most works that the student-bus stop assignment step is made before the route generation step, the authors compute a set of potential stops to each students, generate the routes, and then, assign the students to the best possible bus stop in order to minimize the total travel distance.

In 2014, Kinable et al. [15] present an exact branch-and-price algorithm for the SBRP. The authors aim to find a feasible assignment of students to stops and find routes for the vehicles, minimizing the total length of the routes and respecting the vehicle capacity. Although the authors present an exact approach for the SBRP and also concern about the school bus stop assignment, they use a simpler version of the problem with just one school, single load, homogeneous vehicles and without time windows.

Also in 2014, Faraj et al. [9] solve a single load version of the SBRP using real georeferenced data with information about the students, roads and schools. They deal with the data preparation, bus stops selection and routes generation steps. For the bus stop selection, the authors implement a heuristic based in the Dominating Set Problem in order to find the minimum number of bus stops to assign all students.

A fundamental difference between the work of Faraj et al. [9] and our proposal is related to the location of the bus stop assignment. In Faraj et al. [9], the bus stop can only be allocated to the same position of one student, while in this work the bus stop can be allocated to any position along the roads as long as this position being a vertex in the graph. In this way, much more possible points where a bus stop can be allocated will be analyzed and the results will be compared with the baseline algorithm [9].

IV. Baseline Algorithm

In this section the baseline algorithm proposed by Faraj et al. [9] is presented. We choose this algorithm as our baseline algorithm because Faraj et al. [9] algorithm also aims to minimize the number of bus stops. Moreover, the authors use the same real georeferenced data making easier to compare the results.

In Faraj et al. [9] algorithm the bus stop selection is divided in two phases: (a) generation of buses stops; and (b) assignment of students to their buses stops. The students are divided in two groups: the isolated ones, and the groupable ones. The isolated are the ones which there is no other student closer enough to form a group. A group is formed if there is more than one student whose distance between them is less or equal to the maximum allowed walk distance for each student. In the exact position of each isolated student a bus stop must be created.

For each group, one or more geographical points is chosen which might become a bus stop. In other to choose these points, the authors implement a heuristic for the Dominating Set Problem [16]. The heuristic for **Dominating Set Problem**, proposed by Faraj et al. [9], consists of computing a priority P for each vertex of a graph $G = (V, E)$, in which V represents the set of student residences and E represents the road network. Priority is directly proportional to square of the vertex degree and is inversely proportional to the sum of the degrees of neighbor vertex. Vertices are interactively inserted into the dominating set DS according to its priority (highest to lowest). At each step, all redundancies are removed from DS. The computational cost of this heuristic is given by $\Theta(|Students| \times |Stops|)$.

V. Proposed Algorithm

In this section we present our algorithm to solve the proposed problem. We use the PostgreSQL software with some geographical[1] extensions which deals with spacial data and routing functions respectively.

Given a network N with a set of roads R, a set of students S and the parameters λ and σ - distance between the equidistant points - our algorithm has 5 steps. Firstly, we create equidistant points along the road network. Then we create a topology and project the students to a vertex of the network. Secondly, we compute the travelled distance between each student to each vertex storing the ones that respect the home-to-bus-stop walking distance constraint. At last we select the bus stops using a pseudo-random greedy function. Algorithm 1 presents our algorithm to solve the problem.

Algorithm 1: General Algorithm

```
Input: $R, S, \lambda, \alpha$
Output: Bus Stops
$R' \leftarrow$ Equidistant_Points($R, \lambda$);
Create_Topology($R', S$);
Students_Projection($R', S$);
Distance_Calculation($R', S$);
Bus_Stop_Selection($R', S, \alpha$);
```

A. Creation of Equidistant Vertices in the Road Network

In this section we present the procedure that creates the $Equidistant\_Points$. The original network N has few vertices. The only vertices are the starting points, the ending points of the roads and also the road intersection. The $Equidistant\_Points$ Procedure creates vertices in the original road network at each λ kilometers. For instance, if $\lambda = 0.5$, at each 500 meters of each road, the $Equidistant\_Points$ procedure creates a new vertex. Of course, the higher the number of different points (possible bus stops) created along the road network, the more points must be analyzed and, probably, better solutions will be achieved. In an ideal model, all the points belonging to the roads should be analyzed in order to find the best one which can attend the students. However this is not feasible since the number of points in a road is infinite. In spite of that fact, a good solution will be the one that the distance between each point to another, along the road is small, creating a lot of possible bus stops to be analyzed. However, that distance can not be too small creating a large number of points that can cripple the algorithm. Algorithm 2 presents our procedure to create equidistant points in the road network.

Algorithm 2: Equidistant Points Procedure

```
Input: Roads of the data set $R$, $\lambda$
Output: Set of equidistant points along the roads $R'$
$R' \leftarrow 0$;
for each $r \in R$ do
    $newPoint \leftarrow$ StartPoint(r);
    $R' \leftarrow$ EndPoint(r);
    $numPoints \leftarrow (\lfloor roadLength(r)/\lambda \rfloor)$;
    for $it \leftarrow 0$ to $numPoints$ do
        $newPoint \leftarrow newPoint + \lambda$;
        $R' \leftarrow R' \cup newPoint$;
    end
end
```

The Figure 3 shows a created graph using the original points associated with the vertices of the edges while the Figure 4 shows the created graph for the same data, using the equidistant points created by the Algorithm 2. It is easy to notice that the graph of Figure 4 has more vertices when compared with the graph of Figure 3, showing that there are more bus stops possibilities to be analyzed.

[1] PostGis and PgRouting - postgis.net and pgrouting.org

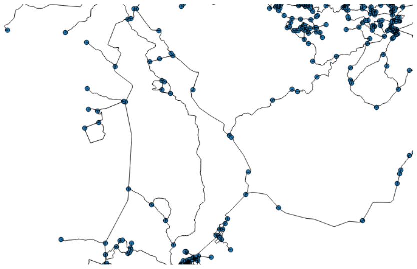

Figure 3. Graph with Original Points

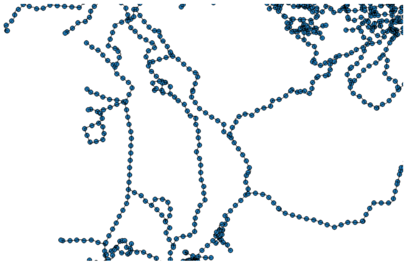

Figure 4. Graph with Equidistant Points

B. Creation of the Topology of the Graph

In order to create the graph topology for the problem we use the *CreateTopology* function provided by PgRouting. It is responsible to create a edge table, a vertex table and to connect all the edges to the respective vertex of the graph. The topology of the graph is based on the geometry information of the georeferenced data. After the creation of the graph, it is possible to use all the routing functions of PgRouting.

This function associates to each edge a vertex corresponding to the point of the beginning of the edge, the point of the end of the edge and the point related with the intersection of edges. Furthermore all the road segments are broken at the intersections.

C. Projection of the Students in a Road Network

In this section we present our projection strategy. Such issue arises since our source database contains the exact georeferenced location of each entity, and we have to bring these entities to the road network. After projecting entities, we separate the street (edge) considering the point of projection.

For this step it is used a function which return the minimum distance between two geometries. So for each coordinate of a student, the closest vertex is associated to that coordinate. In that way, each student is projected on the closest vertex in the graph.

Fig. 5 shows an example of the projection. The green points represents the exact location of entities. The red ones represent their projection to the nearest road.

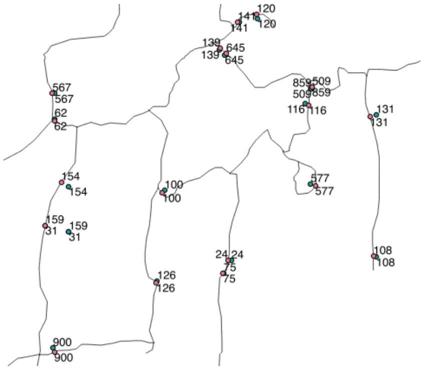

Figure 5. Projection of Students

D. Distance Calculation Procedure

Having all the possible points that could be a bus stop represented by the set of neighbors vertex and the same mapped vertex of the students, it is possible to calculate the distance of each vertex of this set to all of the mapped students.

From each vertex it is possible to pass, as an external parameter, the distance of the restriction of the problem and, therefore, to seek only for the result of the mapped students. In this way it is possible to create a set composed of each vertex and the number of students it can attend using the constrained distance.

E. Bus Stops Selection Procedure

In the section we present our bus stops selection procedure. We build a constructive heuristic that at each iteration chooses a new bus stop. The procedure repeats until all students are assigned.

Our strategy works as follow: for each possible point i, to install bus stops we compute all the unattended students that can be assign to this bus stop respecting the maximal walk-to-bus-stop constraint. Then, instead of always selecting the best solution (the point that attended the most number of unattended students), we build a Restricted Candidate List (RCL) of good elements, and one element (not necessarily the best candidate) is randomly selected. A RCL parameter α determines the level of greediness or randomness in this pseudo-random procedure. In the algorithm, α is the RCL size. When $\alpha = 1$, the RCL has one element representing a greedy solution. When $\alpha = 2$, the RCL has two elements, that means that the RCL will have the two vertex with more unattended students, and so forth. For instance, when $\alpha = |R|$, a full random solution is returned. Algorithms 3 and 4 present the proposed Build_RCL and Select_Bus_Stop procedures.

VI. Computational Results

The methodology described in the previously section is applied in the real georeferenced data set related to a Brazilian city, and the results are compared with the ones found in the Faraj and al. [9]. We analyze our algorithm using different values for λ and different values for α.

```
Algorithm 3: Build_RCL Procedure
  Data: $N, \alpha$
  RCL ← ∅;
  for $i \leftarrow 1$ to $\alpha$ do
      Γ ← Point_With_Max_Unattend_Students($N$);
      RCL ← RCL ∪ Γ;
  end
```

```
Algorithm 4: Select_Bus_Stops Procedure
  Data: $G, \eta, \alpha$
  busStops ← ∅;
  attendedStudents ← 0;
  while attendedStudents $< |S|$ do
      Build_RCL($G$,$\alpha$);
      selectedPoint ← Randomly_Choose_Element($RCL$);
      busStops ← busStops ∪ selectedPoint;
      attendedStudents ← Update_Attended_Students($S$);
  end
```

A. Comparing different values for λ

Our first set of experiments compares different values for λ when $\alpha = 1$. In this case our algorithm is fully greedy, i.e., the Select_Bus_Stops procedure always selects the point that attend the most number of unattended students.

Table I shows the number of bus stops found for the same number of students, using the presented methodology for different λ values. When $\lambda = Original$, it means that the algorithm does not create the equidistant points and it is only possible to install bus stops in the end points, start points and intersection points of the original road network. The field *Number Edges*, *Number Vertices*, *Number Stops* represent the number of edges of the road network graph, the number of vertices, i.e., the number of possible points to install bus stops and the number of bus stops found by our algorithm. The field *Original Stops* represents the number of bus stops that intersects the Original set, i.e., the number of bus stops that are the same ones when $\lambda = Original$.

λ	Number Edges	Number Vertices	Number Stops	Original Stops
Original	3640	1754	128	128
2km	4181	2258	125	51
1km	5124	3106	121	36
0.5km	7166	4901	117	22
0.25km	11561	8570	111	3
0.1km	25449	10855	67	1
0.05km	49183	13899	46	2

Table I
COMPARISON BETWEEN DIFFERENT λ VALUES

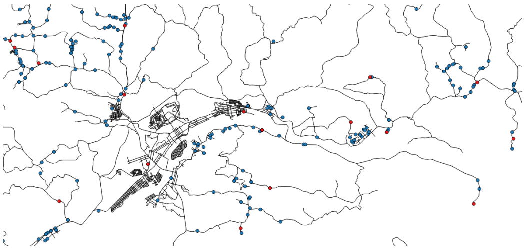

Figure 6. Students and Bus Stops Assignment

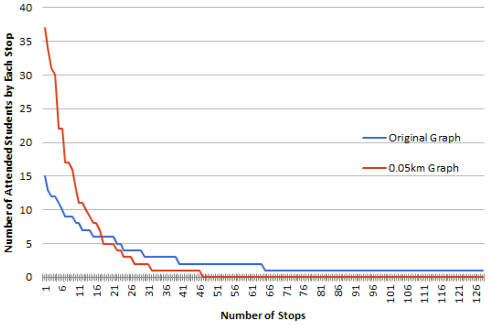

Figure 7. Comparison Between Different λ Values

It is possible to note that for all λ value our algorithm find less bus stops then Faraj et al. [9] work. In our worst case, using only the original vertices, 128 bus stops were found while in the best case, using $\lambda = 0.05$, it was found only 46 bus stops. Also it is possible to observe that as long as the number of vertices in the graph grows, the solution tends to be better. The last column in the Table I shows that as long as the number of vertices grows, the number of bus stops found, that corresponds with the same ones found for the original case, is smaller.

Figure 6 illustrates the students (blue points) and the bus stops (red points) for the best case, using the graph of 0.05 kilometer equidistant vertices.

The Figure 7 illustrates the comparison between the number of students attended by each bus stop. We compared the two approaches. The first one we use the original graph. In the second one $\lambda = 0.05$. Analyzing the figure, is possible to notice that, when $\lambda = 0.05$, the best bus stop for the graph attended 37 students while the best bus stop found for the graph using the original vertices attended only 15 students. Comparing both curves of the two different graphs, it is possible to see why the first one found 128 bus stops while the second one found 46.

B. Comparing different values for α

Our second set of experiments compare four different values for α when $\lambda = 1$, i.e., when the distance between the equidistant points is 1 kilometer. When $\alpha = 1$ we have a fully greedy solution. For each α value we execute the algorithm 33 times in order to present some statistic results.

The Figure 8 illustrates the comparison among the four different λ values. The graphic represent the number of bus stops found for each execution of the algorithm.

Analyzing the Figure 8, it is noticeable that the number

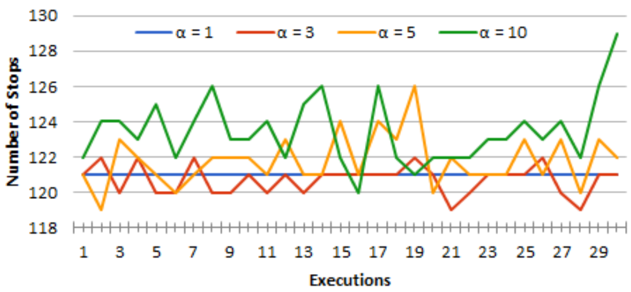

Figure 8. Comparing Different α Values

of bus stops found when $\alpha \neq 1$ was less than the greedy approach ($\alpha = 1$), specially when $\alpha = 3$. However all times that a better solution was found, it was only by one, and, at most, two bus stops less than the greedy approach. So, in order to see how good the fully greedy approach is compared with the pseudo-random approach, the confidence interval was calculated.

A confidence interval is used to indicate whether an estimate is reliable . It represents an estimated range of a population parameter given a range of likely estimates [17]. For this case, it was used the sample average, the standard deviation and the sample size to have a 95% of confidence. It is possible to calculate the confidence interval using the following formula: $-Zc \leq \left(\frac{Media-\mu}{\sigma \div \sqrt{\eta}}\right) \leq Zc$, where Zc = 1,96.

Analyzing the Table II it is possible to observe that when $\alpha = 1$ the algorithm presented a better solution compared with $\alpha = 5$ and $\alpha = 10$. When $\alpha = 1$ the algorithm found always 121 stops which is smaller than the confidence interval of the other two instances. However, for the case of the random candidate, the fully greedy approach present a result that was inside the confidence interval. This demonstrates that in 95% of the times, the fully greedy approach found a solution that was smaller or inside the confidence interval of the average when $\alpha \neq 1$.

	Average	Standard Deviation	Confidence Interval
Best 3	120.73	0.83	$120.43 \leq \mu \leq 121.03$
Best 5	121.80	1.45	$121.20 \leq \mu \leq 122.3$
Best 10	123.47	1.87	$122.76 \leq \mu \leq 124.10$

Table II
CONFIDENCE INTERVAL WHEN $\lambda = 1$

VII. CONCLUSION

In this work we presented a solution for the allocation problem of bus stops using a real and georeferenced data set of the students and roads of a city located in the country side of Brazil. The results found by our algorithm were, in some cases, more than 80% better than the one found in Faraj et al. [9]. We also noticed that the less is the λ value, the better is the solution. Also, it was possible to notice that the fully greedy approach ($\alpha = 1$) was satisfactory for the problem. That fact can be showed analyzing the confidence interval for different α values. All results presented a better solution or a solution that was inside the confidence interval compared with $\alpha \neq 1$.

Finally, this work presented an efficient approach to solve the optimization problem that can be implemented in different georeferenced cities. Is expected that this work can help in the country side scholar transportation. As future works we intend to measure the influence of parameter λ in the total distance traveled by the school buses.

REFERENCES

[1] R. M. Newton and W. H. Thomas, “Design of school bus routes by computer. socio-economic planning sciences,” *Socio-Economic Planning Sciences*, vol. 3, no. 1, pp. 75–85, 1969.

[2] J. Desrosiers, J. Ferland, J.-M. Rousseau, G. Lapalme, and L. Chapleau, *An Overview of a School Busing System*, pp. 235–243. North-Holland, 1981.

[3] J. Park and B.-I. Kim, “The school bus routing problem: A review,” *European Journal of Operational Research*, vol. 202, pp. 311–319, April 2010.

[4] B.-I. Kim, S. Kim, and J. Park, “A school bus scheduling problem,” *European Journal of Operational Research*, vol. 218, pp. 577–585, 2012.

[5] P. Schittekat, J. Kinable, K. Sörensen, F. Spieksma, and J. Springael, “A metaheuristic for the school bus routing problem with bus stop selection,” *European Journal of Operational Research*, vol. 229, no. 2, pp. 518–528, 2013.

[6] J. Desrosiers, J. Ferland, J.-M. Rousseau, G. Lapalme, and L. Chapleau, *TRANSCOL - a Multiperiod School Bus Routing and Scheduling System*, pp. 47–71. North-Holland, 1986.

[7] B.-I. Kim and S. Jeong, “A comparison of algorithms for origin-destination matrix generation on real road networks and an approximation approach.,” *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 70–76, 2009.

[8] L. Chapleau, J.-A. Ferland, and J.-M. Rousseau, “Clustering for routing in densely populated areas,” *European Journal of Operational Research*, vol. 20, no. 1, pp. 48–57, 1985.

[9] M. F. Faraj, J. F. M. Sarubbi, C. M. Silva, M. F. Porto, and N. T. R. Nunes, “A real geographical application for the school bus routing problem,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE*, (Qingdao, China), pp. 2762 – 2767, October 2014.

[10] T. A. Feo and M. G. Resende, “Greedy randomized adaptive search procedures,” *Journal of Global Optimization*, vol. 6, pp. 109–133, 1995.

[11] L. Bodin and L. Berman, “Routing and scheduling of school buses by computer,” *Transportation Science*, vol. 13, no. 2, pp. 113–129, 1979.

[12] G. Dulac, J. A. Ferland, and P. A. Forgues, “School bus routes generator in urban surroundings,” *Computers & Operations Research*, vol. 7, no. 3, pp. 199–213, 1980.

[13] R. Bowerman, B. Hall, and P. Calamai, “A multi-objective optimization approach to urban school bus routing: formulation and solution method,” *Transportation Res A*, vol. 29, pp. 107–123, 1995.

[14] B. Schittekat, M. Sevaux, and K. Sorensen, “A mathematical formulation for a school bus routing problem,” in *IEEE 2006 international conference on service systems and service management*, (Troyes, France), pp. 1552 – 1557, 2006.

[15] J. Kinable, F. Spieksma, and G. Vanden Berghe, “School bus routing—a column generation approach,” *International Transactions in Operational Research*, vol. 21, no. 3, pp. 37–44, 2014.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 3rd ed., 2009.

[17] S. M. Ross, *Introduction to Probability Models*. Academic Press, 2003.