

A mathematical formulation for a school bus routing problem

Patrick Schittekat¹, Marc Sevaux², Kenneth Sørensen^{3*}

¹University of Antwerp, Belgium (patrick.schittekat@ua.ac.be)

²University of South Brittany, France (marc.sevaux@univ-ubs.fr)

³University of Antwerp, Belgium (kenneth.sorensen@ua.ac.be)

ABSTRACT

The school bus routing problem discussed in this paper, is similar to the standard vehicle routing problem, but has several interesting additional features. In the standard VRP all stops to visit are given. In our school bus routing problem, we assume that a set of *potential* stops is given, as well as a set of students that can walk to one or more of these potential stops. The school buses used to pick up the students and transport them to their schools have a finite capacity. The goal of this routing problem is to select a subset of stops that will actually be visited by the buses, determine which stop each student should walk to and develop a set of tours that minimize the total distance travelled by all buses. We develop an integer programming formulation for this problem, as well as a problem instance generator. We then show how the problem can be solved using a commercial integer programming solver and discuss some of our results on small instances.

Keywords: Schoolbus, routing, stop selection, mathematical programming, exact algorithm

1. INTRODUCTION

This research is motivated by a real-life school bus routing problem. In the Flemish region, transport of children to their school is organized by the Flemish transportation society. Each pupil that has the right to be transported to its school, also has the right to a bus stop located at a distance of less than 750 metres from its residence. To efficiently design the routes, a set of potential stops is determined, so that each pupil has at least one stop he/she can walk to. Routes are then determined for the school buses so that all students are picked up at a stop they are allowed to use, while making sure that the capacity of the buses is not exceeded.

Contrary to most vehicle routing formulations, in which a set of stops is given and routes need to be determined that visit each stop, we discuss in this paper a vehicle routing formulation in which a set of *potential* stops is given, but in which determining the set of stops to actually visit is a part of the problem formulation. The objective of this problem is to simultaneously (1) find the set of stops to visit, (2) determine for each customer which stop he should move to and (3) determine routes that visit the chosen stops, so that the total travelled distance is minimized.

Figure 1 shows an example of such a school bus routing problem. In this figure, dots represent students, the small squares represent potential stops, the large square represents the school. Dotted lines indicate which stop a student is allowed to walk to.

Assuming that the capacity of the buses is 8, a possible (but not necessarily optimal) solution to this problem is shown in figure 2.

In the problem discussed in this paper, we assume that all students represent a unit to be transported and that the capacity of the buses can be expressed as a whole number of units. In a more general version of this problem, closer to the classical capacitated vehicle routing problem, each cus-

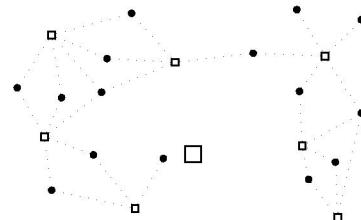


Fig. 1: Unsolved problem

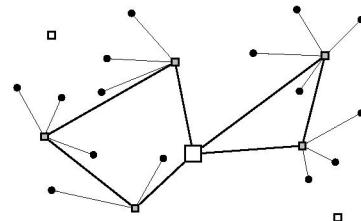


Fig. 2: A possible solution

tomer has a certain demand and the vehicles have a certain capacity, that cannot be exceeded.

We assume that the students cannot walk directly to the school. Students that can walk to school do not have to be taken into account and may consequently be removed from the problem data.

The assignment of students to stops used by the routes is by itself a difficult problem. When students can be assigned to more than one stop in the same route, the assignment to a stop is arbitrary. This is not necessarily the case if a student can be assigned to more than one stop in different routes. In this case, students should be assigned to stops in such a way that the capacity of the vehicles is not violated. In figures 1 and 2, there is one student that can move to a stop in both of the routes. However, given that the capacity of the buses is 8, this student needs to be assigned to the route on the right.

Next to the obvious school bus routing application, this

problem formulation has other applications. For example, some parcel delivery services nowadays offer the option of delivering at a set of pre-defined drop-off points. This has obvious cost-saving advantages over delivering at any location specified by the customer. Customers have to decide beforehand at which drop-off point they wish to pick up their items. It can be envisaged that customers may be asked to specify more than one drop-off point and that the parcel delivery company will then choose among the ones chosen by at least one customer in such a way that routing costs are minimized but every customer can pick up his parcel at one of the drop-off points he specified. Customers may e.g. be notified by a mobile phone message of the drop-off points their package will be delivered. In a more complex setting, the price of the delivery may depend on the number of drop-off points specified by the customer.

The objective of this paper is to define the school bus routing problem as a mixed integer linear programming problem and solve it using both exact and heuristics methods. As will become clear from the literature review, previous research has focused mainly on building intricate multi-objective models of similar problems. The focus on this paper is on understanding the problem in its simple form and building a powerful meta-heuristic to solve large instances quickly.

2. LITERATURE REVIEW

Contrary to the literature on the ordinary vehicle routing problem and several of its extensions (e.g. time windows), only a limited amount of research has considered the routing of school buses. In the literature on routing of school buses, only a small number of papers considers the simultaneous selection of stops.

Most school bus vehicle routing formulations focus on formulating extra constraints and/or objectives to take some student-related factors into account. Bodin and Berman [2], Braca et al. [4], and Desrosiers et al. [6], add a maximum travel-time constraint for each student and/or a time window for arrival at the school. Bennett and Gazis [1] add the total travel time of all children as an objective.

Thangiah et al. [8] discuss the routing of school buses in rural areas. They develop a system that is able to solve large-scale routing problems with a large number of complex constraints and several objectives. Interestingly, the authors note that local government subsidizing policies may result in very ineffective routings, e.g. maximizing the time that students spend on a bus instead of minimizing this. The authors do not consider automatic stop selection in their problem formulation.

Only a limited number of papers on school bus routing considers the automatic selection of bus stops to be a part of the optimization problem. In Dulac et al. [7], students are assigned to an intersection of streets adjacent to the street of their residence. A subset of these potential bus stops is then selected and a VRP solved. In Chapleau et al. [5], potential stops are first clustered, after which stops are selected so that a maximum number of students has a stop within walking distance. The school bus routing problem discussed in Bowerman et al. [3] includes a maximum walking distance for a student to his/her assigned bus stop. The authors de-

velop a multi-objective optimization problem, one of the objectives being the minimization of the total walking distance by all students.

3. PROBLEM FORMULATION

As should be clear from the previous discussion, most papers on the routing of school buses have focussed on capturing as many aspects of reality as possible, building intricate multi-objective models with many complex constraints. In this paper, we focus on a very basic version of this problem to develop efficient methods that take the structure of the problem into account. We focus specifically on the stop selection aspect. Our problem formulation has only a single school, one type of student and one type of bus, with fixed capacity. We optimize the standard vehicle routing criterion: the total distance travelled by all vehicles.

The schoolbus routing problem can be expressed as an integer linear programming problem. We assume that the graph on which the problem is defined, is directed. The following formulation builds upon the one discussed on page 15 of Toth and Vigo [9].

3.1. Data

c_{ij}	Cost of traversing arc ij
K	Number of buses
C	Capacity of the buses
V	Set of all potential stops
E	Set of all arcs between stops
S	Set of all students
s_{li}	Binary variable that indicates whether student l can walk to stop i or not

3.2. Decision variables

x_{ijk}	number of times vehicle k traverses arcs from i to j
y_{ik}	1 if vehicle k visits stop i , 0 otherwise
z_{ilk}	1 if student l is picked up by vehicle k at stop i , 0 otherwise

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (1)$$

$$\sum_{k=1}^K y_{0k} \leq K, \quad k = 1, \dots, K \quad (2)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \quad \forall i \in V, k = 1, \dots, K \quad (3)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk}, \quad \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \dots, K \quad (4)$$

$$\sum_{k=1}^K y_{ik} \leq 1, \quad \forall i \in V \setminus \{0\} \quad (5)$$

$$\sum_{k=1}^K z_{ilk} \leq s_{li}, \quad \forall l \in S, \forall i \in V \quad (6)$$

$$\sum_{i \in V} \sum_{l \in S} z_{ilk} \leq C, \quad \forall k = 1, \dots, K \quad (7)$$

$$z_{ilk} \leq y_{ik}, \quad \forall i, l, k \quad (8)$$

$$\sum_{i \in V} \sum_{k=1}^K z_{ilk} = 1, \quad \forall l \in S \quad (9)$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in V, k = 1, \dots, K \quad (10)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in V | i \neq j \quad (11)$$

$$z_{ilk} \in \{0, 1\}, \quad \forall i, j \in V | i \neq j \quad (12)$$

The objective function (1) minimizes the total travelled distance by all vehicles. Constraints (2) ensure that all vehicles start from the depot. Constraints (3) ensure that if stop i is visited by vehicle k , then one arc should be traversed by vehicle k entering stop i and leaving stop i . Capacity cut constraints (4) ensure that each cut $(V \setminus S, S)$ defined by a customer set S is crossed by a number of arcs not smaller than $r(S)$, the minimum number of vehicles needed to serve set S . These constraints serve as subtour elimination constraints. The following constraints are added or adjusted in comparison with the original formulation to cope with the special features of this specific vehicle routing problem. Constraints (5) ensure that all are visited no more than once, except the stop corresponding to the depot. Constraints (6) ensure that each student walks to a single stop he or she is allowed to walk to. Constraints (7) ensure that the capacity of the vehicles is not exceeded. Constraints (8) ensure that student l is not picked up at stop i by vehicle k if vehicle k does not visit stop i . Constraints (9) ensure that all students are picked up once. Finally, constraints (10), (11), and (12) ensure that all decision variables are binary. This corresponds to ensuring respectively that a vehicle k either visits a stop i or it does not, a vehicle k either drives from one stop i to another stop j or it does not, and a vehicle k either picks up a student l at stops i or it does not.

By using this formulation, we implicitly make a number of assumptions. One assumption is that a stop is only visited by one bus. This means that the number of students per stop may not exceed the capacity of the bus. It also means that the students that go to a bus stop may not be divided into groups which may then each take a different bus. A second assumption is that all buses have equal capacity. Finally, as mentioned, we assume that each student counts as one unit. These assumptions may be relaxed in future research.

4. PROBLEM INSTANCE GENERATION

We have designed and implemented an instance generator for this problem that can generate random problem instances of any size. The generator requires 5 parameters per instance: n_p (the number of potential stops), n_s (the number of students per stop), x_d , y_d (the x and y -coordinates of the depot) and d_{\max} (the maximum walking distance).

The instances are generated on the Euclidean square defined by $(0, 0)$ and (x_{\max}, y_{\max}) . It first generates n_p stops in this square. The coordinates (x_i, y_i) of stop i are uniformly distributed in the intervals $[w_{\max}, x_{\max} - w_{\max}]$ and $[w_{\max}, y_{\max} - w_{\max}]$ respectively. In this way, no student is ever generated outside the boundaries $(0, 0)$ and

$$(x_{\max}, y_{\max}).$$

For each generated stop, n_s student positions are generated at a distance of maximum w_{\max} from the stop. This is done by first generating for each student j an angle $\alpha_j \in [0, 2\pi]$ and a distance w_j from the stop. The student is then put at (x, y) -coordinates equal to $(x_i + w_j \cos \alpha_j, y_i + w_j \sin \alpha_j)$.

5. BENCHMARK EXACT SOLUTIONS

An exact algorithm using the IP formulation proposed in this paper has been implemented in a commercial IP solver and is currently used to solve (small) benchmark instances. Preliminary results are reported using a Pentium 4, 3.20 GHz running linux. All the source code is written using the commercial ILP modelling language Xpress-Mosel 1.6.0 and solved with the optimizer 16.10.07 from Dash Associates. Only three instances have been generated for this first experimental part.

As usual, the IP model is first solved ignoring the breaking subtour constraints (4). Once an optimal solution of the subproblem is obtained, i.e. a lower bound of the global problem, breaking subtour constraints are added one by one and the subproblem with these additional constraints is solved again. This procedure is repeated until no illegal subtours are found. The solution obtained after all these iterations is one of the optimal solution of the problem.

A typical execution is shown on an example with 10 stops and 50 students given on Figure 3. As in Figure 1, the depot is represented with a large square (centered in the figure), stops are represented with small squares and circles locate the students. Dotted lines indicate to which bus stop a student can walk.

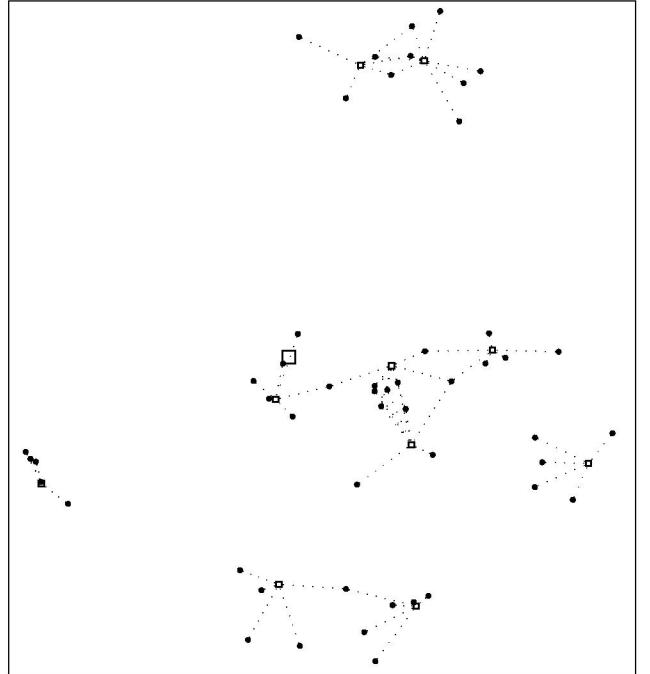


Fig. 3: A medium size problem

The model without any breaking subtour constraints (Constraints (4)) is solved and the result is the one obtained on Figure 4. Clearly, this solution does not satisfy the global

problem constraints and 3 illegal subtours are built (red dashed arcs on the figure). The value of the objective function (1) is 245.74. This value is a lower bound of the optimal objective function value.

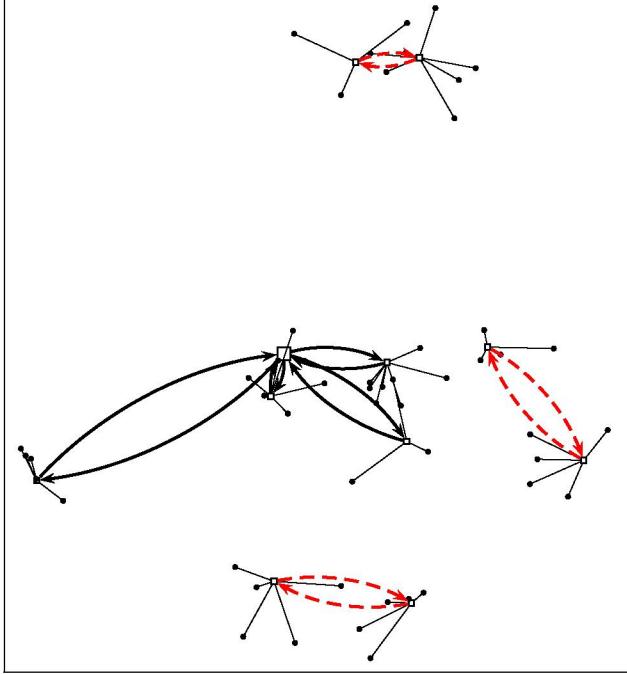


Fig. 4: Solution after the first iteration

Before starting a new iteration, the illegal subtours are automatically detected and the corresponding constraints are added to the model. The problem is solved again with these additional constraints. The results after the second iteration are shown on Figure 5. Again two illegal subtours are created and should be removed during the next iteration.

The objective function value is now 321.23, an improved lower bound of the optimal objective function value. The two illegal subtours correspond to two new constraints that are added to the model and the problem is solved again. The result of the third iteration is presented on Figure 6.

Now only one illegal subtour remains and the objective function value is 327.26. After adding the corresponding constraint to the model, the problem is solved and the optimal solution is presented in Figure 7. The objective function value is 330.87 and this is the optimal cost for the considered problem.

Details of the iterations are provided in Table 1. The first column gives the iteration counter. The second column displays the illegal subtours observed at the end of the iteration; (5,9) for example means that an illegal subtour for stop 5 to stop 9 and back to stop 5 was found. Column “Cost” gives the current value of the objective function. The last two columns give the CPU time and the number of Integer Programming iterations of this subtour elimination iteration. Only four iterations of the subtour elimination procedure are necessary to find the optimal solutions. But each of these iterations costs a lot since an ILP has to be optimally solved. Of course such a methodology can be applied only to small or medium size instances.

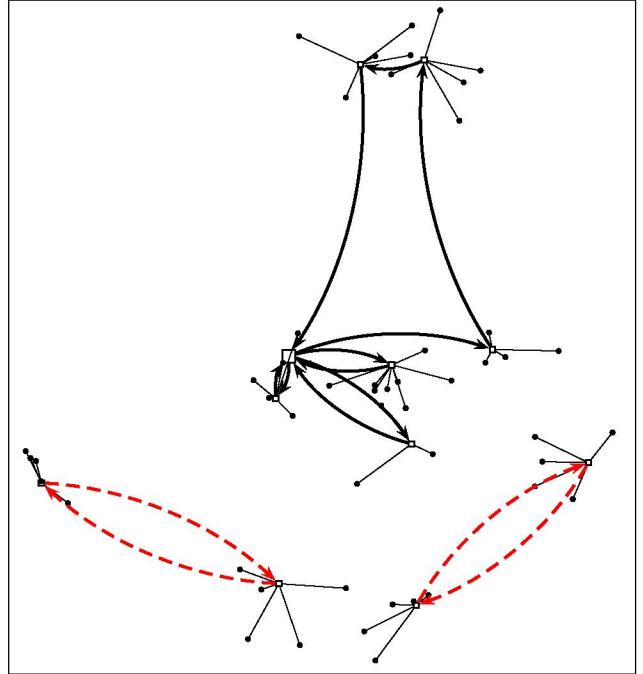


Fig. 5: Solution after the second iteration

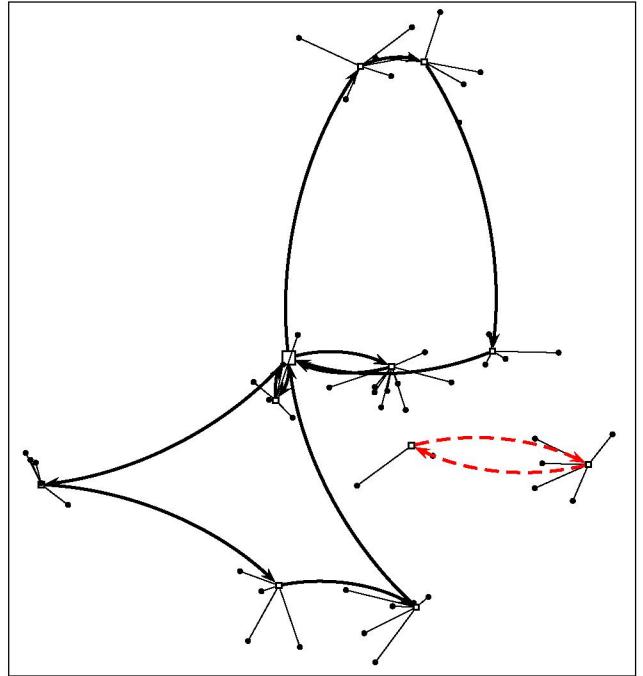


Fig. 6: Solution after the third iteration

It.	Subtours	Cost	CPU (s)	ILP it.
1	(5,9) (2,10) (4,6)	245.74	1424	129427
2	(2,8) (4,10)	321.23	2966	229393
3	(4,7)	327.26	2303	134987
4	—	330.87	5676	313358
Optimal solution		330.87	12369	—

Tab. 1: Instance 1: details of the iterations

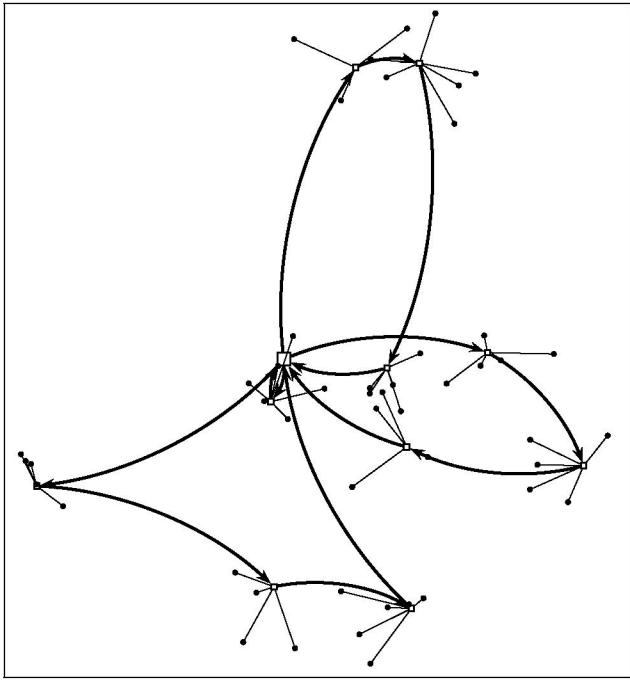


Fig. 7: One optimal solution

It.	Subtours	Cost	CPU (s)	ILP it.
1	(5,8) (2,6) (4,9)	223.90	10686	639683
2	(2,9)	235.17	19790	1324610
3	(2,4,9)	236.24	10900	590679
4	(2,9,4)	236.25	10722	547891
5	(3,9) (2,4)	243.81	6858	482269
6	(6,9) (2,7)	252.57	10406	628056
7	(3,6)	253.75	7892	426228
8	(4,6)	254.33	9961	563522
9	(3,9,4)	255.90	45045	2173264
10	(3,4,9)	255.90	27594	1314457
11	(2,6,4)	257.10	26829	1311801
12	(2,4,6)	257.10	5712	278011
13	Manually stopped after 37369s			
Unfinished after about 64 hours of CPU time				

Tab. 2: Instance 2: details of the iterations

Tests have been conducted on other similar instances with different results. A second instance with the same number of students and stops ran for hours and was stopped before giving a feasible solution. Partial results are reported in Table 2.

Approximate solutions can be found by truncating the search. This is done by setting a stopping condition in the ILP solver (limit of CPU time between two integer solutions). Since one integer solution corresponds to several bus tours (including illegal subtours), we can stop the search and check for illegal subtours. After adding the corresponding constraints, we can solve the augmented problem. The cost value in that case has no signification (in terms of bounding) but if a feasible solution is obtained at the end (with no illegal subtours), this is an upper bound of the optimal solution. We have proceed like this for instances 2 and 3.

We arbitrarily set the time limit to 600s between two in-

It.	Subtours	Cost	CPU (s)	ILP it.
1	(6,10) (1,9) (5,8)	294.67	84	9665
2	—	307.44	143	7575
Optimal solution		307.44	227	—

Tab. 3: Instance 3: details of the iterations

teger solutions. For instance 2, the result obtained gives a total cost of 262.21 which is in fact only 1.98% from the bound provided at the last complete iteration of Table 2. 15 subtour elimination iterations and 9780 seconds of CPU time were necessary to reach the end of the search and obtain the first feasible solution. Concerning instance 3, we were lucky enough to obtain the optimal solution in 227 seconds of CPU time and after only one subtour elimination iteration. The optimal solution of the subproblem at the first iteration of the subtour elimination procedure was found in only 84 seconds of CPU time. Results are reported in Table 3.

6. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have proposed an IP formulation for a schoolbus routing problem in which selection of stops from a set of potential stops and allocation of students to stops were additional decision variables. Our research efforts are now aimed in two directions: a cutting plane algorithm to obtain lower bounds of the problem and a metaheuristic optimization approach to solve large scale instances. In the future, additional features may be added to the formulation, such as allowing more buses to visit a stop, adding time window constraints, etc.

REFERENCES

- [1] B. Bennett and D. Gazis. “School bus routing by computer”. *Transportation Research*, 6:317–326, 1972.
- [2] L. Bodin and L. Berman. “Routing and scheduling of school buses by computer”. *Transportation Science*, 13:113–129, 1979.
- [3] R. Bowerman, B. Hall, and P. Calamai. “A multiobjective optimization approach to urban school bus routing: formulation and solution method”. *Transportation Research Part A: Policy and Practice*, 29A:107–123, 1995.
- [4] J. Braca, J. Bramel, B. Posner, and Simchi-Levi D. “A computerized approach to the New York City school bus routing problem”. *IIE Transactions*, 29:693–702, 1997.
- [5] L. Chapleau, J. Ferland, and J. Rousseau. “Clustering for routing in densely populated areas”. *European Journal of Operational Research*, 20:48–57, 1985.
- [6] J. Desrosiers, J. Ferland, J.-M. Rousseau, G. Lapalme, and L. Chapleau. “An overview of a school busing system”. In N. Jaiswal, editor, *Internal Conference on Transportation*, volume IX of *Scientific Management of Transportation*, pages 235–243, New Delhi, India, 1980.
- [7] G. Dulac, J. Ferland, and P.-A. Fogues. “School bus routes generator in urban surroundings”. *Computers and Operations Research*, 7:199–213, 1980.
- [8] S.R. Thangiah, B. Wilson, A. Pittluga, and W. Mennell. “School bus routing in rural school districts”. Working

paper, Computer Science Department, Slippery Rock University, Slippery Rock, PA, USA, 2005.

- [9] P. Toth and D. Vigo, editors. “*The vehicle routing problem*”. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.