DEPARTMENT OF ENVIRONMENT,
TECHNOLOGY AND TECHNOLOGY MANAGEMENT

A matheuristic for the school bus routing problem

Patrick Schittekat, Kenneth Sörensen, Marc Sevaux & Johan Springael

UNIVERSITY OF ANTWERP
Faculty of Applied Economics

EARNED EXCELLENCE
AACSB ACCREDITED
EST. 1916
THE BEST BUSINESS SCHOOLS IN THE WORLD

Stadscampus
Prinsstraat 13, B.213
BE-2000 Antwerpen
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
http://www.ua.ac.be/tew

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENVIRONMENT,
TECHNOLOGY AND TECHNOLOGY MANAGEMENT

A matheuristic for the school bus routing problem

Patrick Schittekat, Kenneth Sörensen, Marc Sevaux & Johan Springael

RESEARCH PAPER 2009-010
OCTOBER 2009

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.213
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@ua.ac.be

D/2009/1169/010

A matheuristic for the school bus routing problem

Patrick Schittekat*[1,2], Kenneth Sörensen[1], Marc Sevaux[3], and Johan Springael[1]

[1]Faculty of Applied Economics, University of Antwerp
patrick.schittekat@ua.ac.be, kenneth.sorensen@ua.ac.be, johan.springael@ua.ac.be
[2]ORTEC
[3]Lab-STICC Laboratory, University of South Brittany
marc.sevaux@univ-ubs.fr

Existing literature on routing of school buses has focused mainly on building intricate models that attempt to capture as many real-life constraints and objectives as possible. In contrast, the focus of this paper is on understanding the problem in its most basic form. To this end, we define the school bus routing problem (SBRP) as a variant of the vehicle routing problem in which three simultaneous decisions have to be made: (1) determine the set of stops to visit, (2) determine for each student which stop he should walk to and (3) determine routes that visit the chosen stops, so that the total traveled distance is minimized. We develop an MIP model of this basic problem.

To efficiently solve large instances of the SBRP we develop an efficient GRASP+VND metaheuristic. Our method can be called a *matheuristic* because it uses an exact algorithm to optimally solve the subproblem of assigning students to stops and to routes. The results of our matheuristic approach on 112 artificially generated instances are compared to those obtained by implementing the MIP model in a commercial solver and solving it using a specially developed cutting plane procedure. Experiments show that our matheuristic outperforms the exact method in terms of speed and matched the exact method in terms of solution quality.

*Corresponding author

1 Introduction

In the Flemish region of Belgium, students that live within certain minimum and maximum distances of their school have the right to free transport to and from the school. The transport is organized by the Flemish transportation company, which uses school buses that drive fixed routes. An additional requirement is that a bus stop should be located at a distance of at most 750 metres from the home of each student. Each school term, the Flemish transportation company determines which routes its buses will follow, and where they should stop so that each student has at least one stop he or she can walk to. To this end, a set of *potential* stops is determined first in such a way that each student lives within 750m of at least one stop. 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. The Flemish transportation company is faced with problems where up to 3000 students have to be picked up and brought to 7 different schools.

Contrary to most vehicle routing formulations, in which a set of stops is given and routes need to be determined that visit each stop, this paper discusses a vehicle routing problem 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 student which stop he should move to and (3) determine routes that visit the chosen stops, so that the total distance traveled by the buses is minimized. Figure 1 shows an example of this problem, that we call the *school bus routing problem* or SBRP. In this figure, dots represent students, small squares represent potential stops and large square represents the school. Dotted lines indicate which stops a student is allowed to walk to.

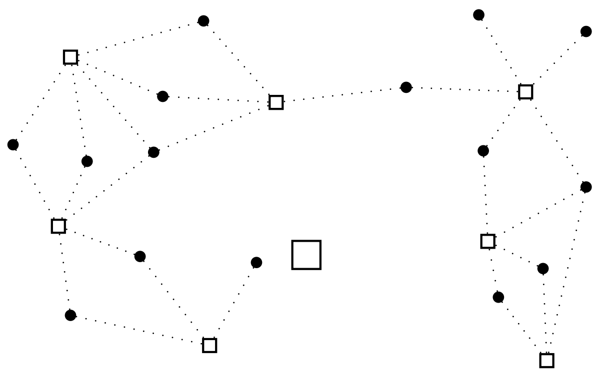


Figure 1: Unsolved problem

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

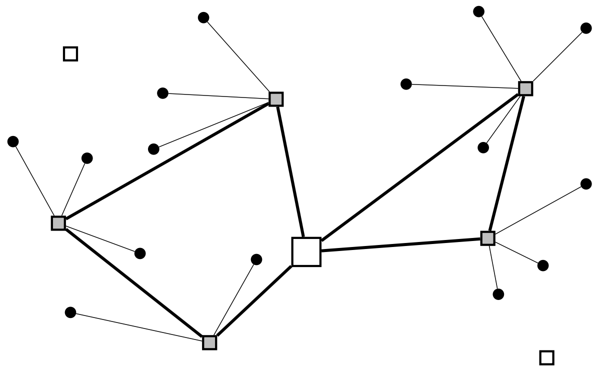


Figure 2: A possible solution

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 an integer number of units. Students that can walk directly to school are not taken into account.

Even when the stops to use and the routes that visit these stops have been determined, the sub-problem of allocating students to stops used by the routes is not trivial. When students can be assigned to multiple stops in the same route, the allocation to a stop is arbitrary. This is not usually the case if a student can be assigned to multiple stops in different routes. In this case, students should be assigned to stops in such a way that the capacity of the buses 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. While the possibility to assign students to different stops offers the possibility to incur potential savings, it introduces an extra decision level that makes the problem much more difficult to solve.

Next to the obvious school bus routing application, this problem formulation has other applications. For example, large companies that want to organize common transport for their employees are faced with the same problem. A related but different problem can be found in some parcel delivery services which 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 selected 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 specific drop-off point their package will be delivered at. In a more complex setting, the price of the delivery may depend on the number of drop-off points specified by the customer. Note that the capacity constraints in this case may have to be replaced by the more typical vehicle routing constraints, in which each order has a certain size and the sum of all order sizes in a route may not exceed the vehicle capacity.

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.

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 (1979), Braca et al. (1997), and Desrosiers et al. (1980), add a maximum travel-time constraint for each student and/or a time window for arrival at the school. Bennett and Gazis (1972) add the total travel time of all children as an objective.

Thangiah et al. (2005) 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 it.

Some papers exist that focus on the selection of stops as an integral part of the optimization problem. In Dulac et al. (1980), 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 is solved. In Chapleau et al. (1985), 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. (1995) includes a maximum walking distance for a student to his/her assigned bus stop. The authors develop a multi-objective optimization problem, one of the objectives being the minimization of the total walking distance of all students.

There exist similar problems outside the school bus routing context. The capacitated m-ring star problem (Baldacci et al., 2004) differs from our school bus routing problem in that there are no restrictions on which students can be assigned to which stops (or in the case of the m-ring star problem, which customers can be assigned to which transition points), rather an assignment cost is given. Moreover, the number of rings (tours) is pre-specified. In the multi-vehicle covering tour problem (Hachicha et al., 2000) the total route length and the number of stops that can be visited in a route is limited, instead of the total capacity in a route.

Previous research has focused on building intricate multi-objective models of school bus routing problems, attempting to capture as many real-life constraints and objectives as possible. The aim of this paper is different: to study the problem in its most basic form and develop both a mathematical programming model and an efficient metaheuristic that uses some of the mathematical properties of the problem. More specifically, we find that the problem can be decomposed in a master problem and a subproblem. The master problem is an integer programming problem and consists of the selection of stops and the routing decisions. The subproblem decides on the allocation of students to stops. Because of the mathematical properties of the subproblem, it can be efficiently solved using an exact linear programming method, that we integrate into our metaheuristic.

The resulting *matheuristic* consists of two phases. The construction phase uses ideas from GRASP or greedy randomized adaptive search procedure (Feo and Resende, 1989, 1995), a constructive metaheuristic that attempts to balance greediness and randomness. The improvement phase is a variable neighborhood descent (VND) method, a variant of variable neighborhood search (VNS) (Mladenović, 1995; Hansen and Mladenović, 1997, 1999). VNS is one of the dominant paradigms in vehicle routing metaheuristics, and a large number of successful applications has been reported (Hansen and Mladenović, 2001a,b). The student allocation subproblem is solved exactly by the primal-dual labeling method of Ford and Fulkerson (1962) initially developed for the transportation problem. Section 4.3 explains how the student allocation subproblem can be transformed into a special case of the transportation problem.

3 Problem formulation

As mentioned, this paper focuses on a basic version of the problem of routing school buses. Special attention is given to the stop selection aspect, that distinguishes this problem from more traditional vehicle routing problems. We attempt to uncover the relationship between allocation, selection, and routing decisions and use this information to build a powerful metaheuristic to solve large instances quickly. We restrict ourselves in this paper to 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 traveled by all vehicles. The basic school bus routing problem (SBRP) as described here is a generalization of the basic vehicle routing problem (VRP) and therefore also NP-hard. The SBRP 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 on the formulation of Toth and Vigo (2001, p. 15). Table 1 discusses the symbols used in the model.

Table 1: Symbols used in the mathematical model

Data	
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
c_{ij}	Cost of traversing arc from stop i to stop j
s_{li}	1 if student l can walk to stop i and 0 otherwise
$i = 0$	Index for the school
Decision variables	
x_{ijk}	1 if vehicle k traverses arc from i to j, 0 otherwise
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

The mathematical programming formulation of the school bus routing problem (SBRP) is the following.

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

s.t.

$$\sum_{k=1}^{K} y_{0k} \leq K \qquad k = 1, \ldots, K \tag{2}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \qquad \forall i \in V, k = 1, \ldots, K \tag{3}$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk} \qquad \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \ldots, K \tag{4}$$

$$\sum_{k=1}^{K} y_{ik} \leq 1 \qquad \forall i \in V \setminus \{0\} \tag{5}$$

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

$$\sum_{i \in V} \sum_{l \in S} z_{ilk} \leq C \qquad k = 1, \ldots, K \tag{7}$$

$$z_{ilk} \leq y_{ik} \qquad \forall i, l, k \tag{8}$$

$$\sum_{i \in V} \sum_{k=1}^{K} z_{ilk} = 1 \qquad \forall l \in S \tag{9}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in V, k = 1, \ldots, K \tag{10}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in V | i \neq j \tag{11}$$

$$z_{ilk} \in \{0, 1\} \qquad \forall i, j \in V | i \neq j \tag{12}$$

The objective function (1) minimizes the total traveled distance by all buses. Constraints (2) ensure that all buses start from the school. The maximum number of buses K obviously cannot exceed the number of stops. Constraints (3) enforce that if stop i is visited by vehicle k, then an arc should be traversed by vehicle k entering stop i and leaving stop i. Capacity cut constraints (4) check that each cut $(V \setminus S, S)$ defined by a student set S is crossed by a number of arcs not smaller than $r(S)$, the minimum number of buses needed to serve set S. These constraints serve as subtour elimination constraints. Constraints (5) guarantee that all stops are visited no more than once, except the stop corresponding to the school. Constraints (6) ensure that each student walks to a single stop he or she is allowed to walk to. Constraints (7) make sure that the capacity of the buses is not exceeded. Constraints (8) impose that student l is not picked up at stop i by vehicle k if vehicle k does not visit stop i. Constraints (9) enforce that all students are picked up once. Finally, constraints (10), (11), and (12) require 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. Thirdly, one bus can only perform one route. Finally, as mentioned, we assume that each student counts as one unit. These assumptions may be relaxed in future research.

4 A GRASP+VND matheuristic for the school bus routing problem

In this section, we develop a hybrid exact/metaheuristic procedure to solve large instances of the school bus routing problem. Our *matheuristic* uses a GRASP construction phase followed by a variable neighborhood descent (VND) improvement phase. These two phases are executed sequentially and the resulting procedure is iterated n_{max} times, after which the best solution is selected as the final solution. As mentioned, the student allocation subproblem is solved by an exact method.

4.1 GRASP construction phase

GRASP, or greedy randomized adaptive search procedure, is a well-known constructive metaheuristic, that starts from an empty solution and builds a complete solution by adding one element at a time. Most GRASP implementations use a *restricted candidate list* (RCL), which is a subset of all candidate elements selected in a greedy fashion. Assuming a minimization problem, the RCL contains the elements whose incorporation into the partially built solution would yield the smallest increase (or largest decrease) in objective function value. From the RCL, an element is then selected at random, after which the RCL is updated to reflect the fact that a new element was added to the solution and is no longer available for selection. Selection of an element and update of the RCL are repeated until a complete solution has been built. The size of the RCL, α, is a parameter of the GRASP algorithm that controls the balance between greediness and randomness. If α is small, the construction is relatively greedy. If α is large, it is relatively random. In the extreme cases, $\alpha = 1$ causes a completely deterministic greedy construction. If α is equal to the number of elements in the solution, the construction is completely random.

The GRASP construction phase in our metaheuristic is based on the well-known Clark-Wright savings heuristic (Clark and Wright, 1964) for the vehicle routing problem (VRP). This heuristic starts from a solution in which all stops are visited in separate routes. The heuristic builds a *savings matrix* that contains for each pair of stops the decrease in cost (or "saving") that would result from connecting the stops, thereby merging the two routes that contain the stops. For two stops to be "connectable", they have to be in different routes. Moreover, one of the stops has to be the first stop in a route and the other one the last. Also, the total capacity required by the two routes containing the stops cannot be larger than the capacity of the vehicle. In each iteration, the original Clarke-Wright heuristic greedily selects the pair of stops to connect.

Like the original Clark-Wright heuristic, our GRASP procedure starts from a solution in which each stop is used and visited in a separate route. After this initial setup, students are assigned to these stops by solving the student allocation subproblem (see 4.3). Obviously, if no feasible allocation can be found, no feasible solution for the SBRP instance exists. If a feasible assignment of students to stops can be found, the algorithm proceeds using a randomized variant of the Clarke-Wright heuristic connecting two stops (and merging two routes) in each iteration. Unlike the Clarke-Wright heuristic for the VRP, the feasibility of a solution after connecting two stops is more difficult to determine, as it might involve reallocating the students over the different routes (using the student allocation subproblem algorithm).

To generate different solutions, our GRASP construction heuristic adopts a parameter-free method to balance randomness and greediness. Instead of using a restricted candidate list, a *roulette wheel selection* procedure is introduced which selects candidate stop pairs with a probability proportional to the saving that would result from connecting them. To save time, the roulette wheel mechanism does not take into account the feasibility of the solution after connecting the selected pair of stops, as this would involve solving many student allocation subproblems before selecting a pair of stops to connect. If a pair of stops is selected that results in an infeasible solution when connected, the move is not executed and removed from the list of stop pairs.

Pseudo-code for the GRASP construction phase is shown in algorithm 1. After each iteration of the GRASP construction phase, a feasible solution is found. This solution is then subjected to the VND improvement phase.

Algorithm 1: GRASP construction phase for the SBRP

```
Input: initial solution with one route per stop
Calculate Clark-Wright savings matrix σ_ij = c_i0 + c_0j − c_ij;
Create list of stop pairs L containing all pairs (i, j);
repeat
    Calculate probability of selecting stop pair (i, j) ∈ L as p_ij = σ_ij / Σ_{i,j} σ_ij;
    roulette wheel selection: select stop pair (i, j) ∈ L with probability p_ij;
    if connecting stops i and j yields a feasible solution then
        Connect stops i and j;
    end
    Remove pair (i, j) from L;
until L is empty ;
```

4.2 VND improvement phase

Variable neighborhood descent (VND) is a deterministic variant of the well-known variable neighborhood search (VNS) metaheuristic. Most implementations of VNS use a sequence of nested neighborhoods, $\mathcal{N}_1$ to $\mathcal{N}_{k_{\max}}$, in which each neighborhood in the sequence is "larger" than its predecessor, *i.e.* $\mathcal{N}_k \subset \mathcal{N}_{k+1}$. VNS typically uses a perturbation move for diversification purposes. In our algorithm, diversity is introduced by the different starting solutions generated by the GRASP construction phase and a perturbation phase is not needed. We therefore use the variable neighborhood *descent* or VND variant. Pseudo-code for the VND improvement phase is given in algorithm 2.

Algorithm 2: Variable neighborhood descent for the SBRP

```
Input: Solution x obtained by GRASP
Initialize: k ← 1;
repeat
    local search: perform local search using neighborhood N_k, starting from
    solution x until it cannot be improved;
    if x' is better than x then
        x ← x' (center the search around the new solution) and k ← 1 (search
        again using the first neighborhood);
    else
        k ← k + 1;
    end
until k = k_max ;
```

Our VND improvement phase uses four neighborhood structures that are applied in the order presented here. Neighborhood structures can be classified by the type of moves they allow. The four move types defining the implemented neighborhood structures We first describe the different move types and then elaborate on the search strategy that we use in these neighborhoods. The four move types are presented graphically in figure 3.

The first two are *remove-insert within a route* and *remove-insert between routes*. In these typical VRP neighborhoods a stop is removed from its current location and inserted at another location in the solution. The distinction between relocating a stop within a route or between routes is important because of the student allocation subproblem. When a remove-insert move is applied within a single route no student reallocation or capacity check has to take place. When a stop is moved to another route, the assignment of students to stops is initially left unchanged. A simple capacity check shows whether the addition of the extra stop to the second route violates the bus capacity of this route. If this is the case students are reallocated to the visited stops of the proposed solution. If a feasible reallocation is found, the move is executed, otherwise it is discarded.

A third move type is called *replace* and is specific to the SBRP. This move removes a visited stop from a route and adds another (unvisited) one. The move only attempts to remove stops that are not obligatory. An obligatory stop is one that needs to be visited in each feasible solution because there exists at least one student for which this stop is the only one he can walk to. The student allocation subproblem is always solved after a replace move.

Finally, the *remove* move type reduces the total distance of the current solution by removing a stop from a route. To check the feasibility of the solution after a remove operation, the student allocation subproblem is solved.

To save time generating solutions in a neighborhood, we adopt the following strategy. When local search using a specific neighborhood structure is started from a given initial solution all possible moves that form this neighborhood are sorted in descending order according to their respective savings. Only moves with a positive saving are considered. The list of improving moves is then traversed in decreasing order of saving and moves are executed as they appear on the list if (1) they result in a decrease in objective function, (2) they can be executed and (3) the resulting solution is feasible. Remark that some moves might yield a different saving than the one initially predicted or become impossible because of the prior execution of other moves on the list. However, we found that the effort of updating the list of savings after each move does not outweigh the additional benefits of increased accuracy. If a move becomes non-improving after some other move(s), this move is simply discarded. This procedure ends when there are no improving moves left in this list. The fact that there are no more improving moves on the list does not imply that the resulting solution is a local optimum with respect to the current neighborhood. However, the structure of the VND ensures that the final solution found is a local optimum in all four neighborhoods.

4.3 Solving the student allocation subproblem exactly

In our metaheuristic solution method, the SBRP is decomposed in a master problem and a subproblem. The master problem is a vehicle routing problem with stop selection, the objective of which is to minimize the total traveled distance. Once the stops have been selected and the routes have been fixed, a subproblem remains of allocating students to stops in such a way that the capacity of the buses is not exceeded (see figure 4). This subproblem is a constraint satisfaction problem in that it does not have an objective function. The existence of a feasible solution to this problem however implies that the corresponding solution of the master problem is valid. A solution to the master problem fixes both the stops that are used and the routes that are performed, *i.e.* it fixes the values of variables y_{ik} and x_{ijk}. Thus, only the z_{ilk} variables need to be determined for given values of y_{ik} and x_{ijk}. The subproblem can be written as an optimization problem as follows:

$$\min \sum_{l \in S} \sum_{k=1}^{K} t_{kl} z'_{kl} \tag{13}$$

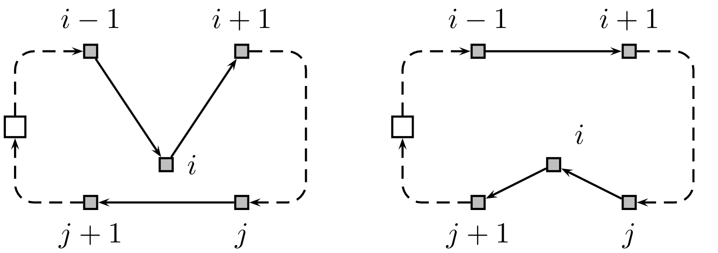
s.t.

$$\sum_{k=1}^{K} z'_{kl} = 1 \qquad \forall l \in S \tag{14}$$

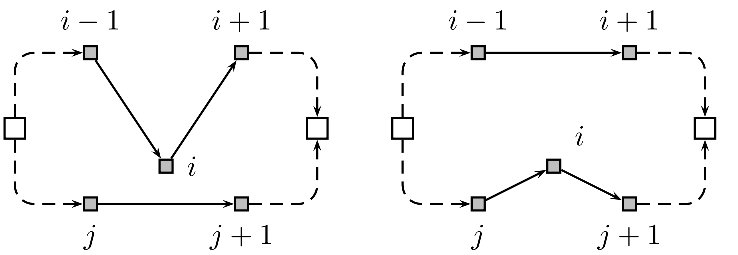
$$\sum_{l \in S} z'_{kl} \leq C \qquad \forall k = 1, \ldots, K \tag{15}$$

$$z'_{kl} \in \{0, 1\} \qquad \forall k = 1, \ldots, K, l \in S \tag{16}$$

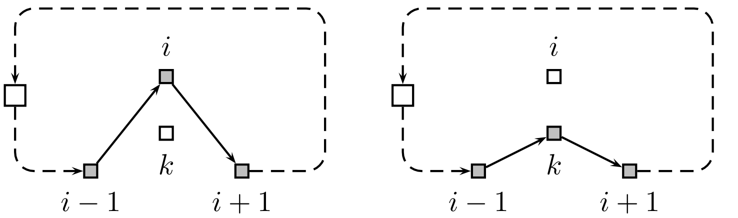
In this formulation $z'_{kl} = \sum_{i \in V} y_{ik} z_{ilk}$ is a decision variable equal to 1 if student l is



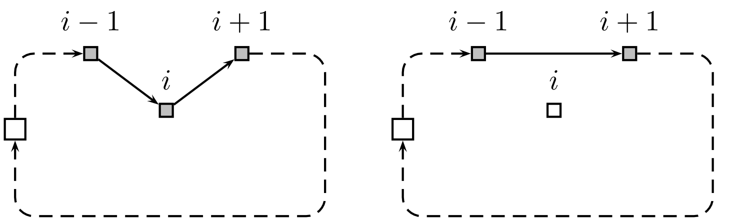
(a) Remove–insert within a route



(b) Remove–insert



(c) Replace



(d) Remove

Figure 3: Different move types

picked up in route k. The variable t_{kl} indicates the "cost" of assigning a student to a route. That cost is 0 if student l can walk to at least one stop in k and 1 otherwise. Constraints (14) ensure that each student is assigned to exactly one route. Constraints (15) ensure that the capacity of the buses is not exceeded. As already noted in original formulation of the SBRP, a bus cannot perform more than one route. The number of routes can be lower than the maximum number of buses K. This means that less than K buses have to be used when this solution is implemented in practice.

This problem is a special case of the transportation problem. Because of the structure of the cost matrix (which is totally unimodular) and the integer right-hand-sides of the constraints, we can relax the integrality constraints (16). Any feasible solution of the relaxed subproblem is guaranteed to be integer. The subproblem can therefore be solved to optimality by any algorithm for the transportation problem.

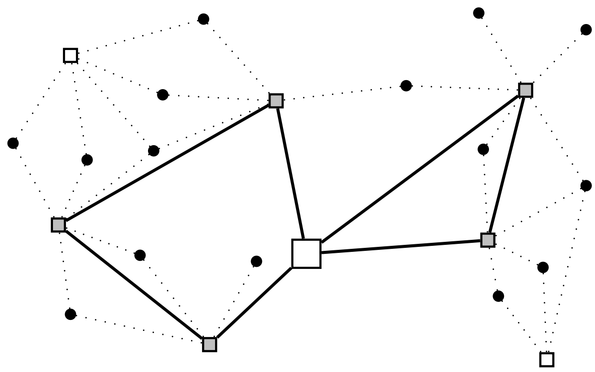


Figure 4: When the routes have been fixed, the allocation of students to routes is a special case of the transportation problem

The objective function (13) minimizes the cost of assigning all students to a route. If there exists an allocation of all the students to the routes, the objective function will equal 0, indicating that all students can be assigned to the current solution of the master routing problem.

In our matheuristic, we solve the transportation problem using the well-known primal-dual labeling method of Ford and Fulkerson (1962). For the details of this method, we refer to the cited paper.

5 Experiments

5.1 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 school) and $w_{\max}$ (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 $(x_i + w_j \cos \alpha_j, y_j + w_j \sin \alpha_j)$.

The 112 instances considered for the experiments in this paper are available from the authors upon request. The instance names are SSSS-sα-uβ-cγ-wδ for an instance with α stops, β students, a bus capacity of γ and a walking distance of δ. For example, the instance of which the best solution found appears in figure 5 is called `SSSS-s40-u200-c25-w10`.

5.2 Exact benchmark solutions

An exact algorithm using the MIP formulation proposed in this paper has been implemented in a commercial MIP solver and used to solve (medium-sized) benchmark instances. 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 and executed on a Pentium 4, 3.20 GHz running linux.

To solve these instances we have implemented a cutting plane procedure. The MIP model is solved by initially relaxing the subtour elimination constraints (4) and solving the relaxed problem to optimality. Then, subtour elimination constraints are added for each subtour encountered in the relaxed solution and the problem is solved again.

Table 2: Instance 1: details of the iterations

It.	Subtours	Cost	CPU (s)
1	(5,8) (2,6) (4,9)	223.90	10686
2	(2,9)	235.17	19790
3	(2,4,9)	236.24	10900
4	(2,9,4)	236.25	10722
5	(3,9) (2,4)	243.81	6858
6	(6,9) (2,7)	252.57	10406
7	(3,6)	253.75	7892
8	(4,6)	254.33	9961
9	(3,9,4)	255.90	45045
10	(3,4,9)	255.90	27594
11	(2,6,4)	257.10	26829
12	(2,4,6)	257.10	5712
13	Manually stopped after 37369s		
Unfinished after about 64 hours of CPU time			

Table 3: Instance 2: details of the iterations

It.	Subtours	Cost	CPU (s)
1	(6,10) (1,9) (5,8)	294.67	84
2	–	307.44	143
Optimal solution		307.44	227

This procedure is repeated until no more subtours are found, at which time an optimal solution has been found.

The performance of the solver using our cutting plane procedure is unpredictable, which is exemplified by the following two test cases, both with 10 potential stops and 50 students.

The solution process for our procedure on the two instances is shown in tables 2 and 3. In these tables, the first column shows the iteration number. The Subtours column shows the subtours present after that iteration. The column CPU (s) shows the number of CPU seconds that were used for that iteration. The Cost column contains the cost of the solution at this iteration. Whereas the second example is solved to optimality in about three minutes after only two iterations, the first example cannot be solved to optimality in 64 hours.

5.3 GRASP+VND matheuristic results

The experimental results on medium-sized instances (10 stops, 50 students) in the previous section show that our exact method exhibits a large variability in computation time, ranging from 2 minutes to more than 64 hours. One may argue that SBRPs are tactical problems (school bus routes are only determined once every school term) and that computational performance is not a major issue. However, managers and decision makers generally want to have the opportunity to quickly assess solutions for different scenarios. Moreover, next to the obvious school bus routing application, this problem formulation has other applications (*e.g.* parcel delivery service) in which fast running times are required. Therefore, for large instances as we encounter in practice (50 stops, 500 students per school) exact methods are not expected to result in viable solution techniques.

To test our matheuristic, a larger experiment of 112 instances was set up with problem sizes ranging from 5 stops and 25 students to 80 stops and 800 students. Also, four maximum walking distances were considered: 5, 10, 20, and 40. The maximum walking distance determines to a large extent how many stops the average student is able to walk to. Clearly, the larger the maximum walking distance, the more degrees of freedom there are in the student allocation subproblem. Below, the results are summarized by maximum walking distance. The vehicle capacity is either 25 or 50. For every instance, the metaheuristic was stopped after 25 runs of the GRASP+VND, but only solutions found within one hour were considered and reported.

Every summary consists of 5 columns indicating the number of stops, the number of students, the vehicle capacity, the execution time of the matheuristic and the total travel distance of the best solution found respectively. Two columns were also added: the execution time and travel distance of the optimal solution produced by the exact algorithm described here above, for those instances which could be solved to optimality. Maximum computing time of the exact algorithm was also one hour. The exact procedure was able to find 32 optimal solutions within the time limit of one hour. However, the lower bounds obtained after one hour were generally very weak and are consequently not reported. The largest instance that could be solved optimally, was an instance with 20 stops and 400 students. The best solution for a 40 stops, 200 students instance with vehicle capacity of 25 and maximum walking distance of 10 can be found in figure 5.

The matheuristic experiments were conducted on a different computer than the exact algorithm experiments, more specifically on a Pentium Centrino 2.20 GHz. The reported CPU Time is scaled according to Dongarra (2009) such that CPU times can be adequately compared.

The results of our experiments show that for every instance where the optimal solution was found by the exact algorithm, the matheuristic also gives the optimal solution and is always clearly faster than the exact algorithm. The matheuristic is 1.25 to 800 times faster, than the exact algorithm for those instances. Notwithstanding the fact that the problem difficulty increases rapidly when the number of stops and students increases, the matheuristic can generate high quality solutions for instances up to 80 stops and 800 students as opposed to the exact method.

6 Conclusions and future research

In this paper, we have proposed an MIP formulation for a school bus routing problem in which selection of stops from a set of potential stops and allocation of students to stops are additional decision variables. We propose a GRASP+VND matheuristic that uses an exact linear programming procedure to solve the subproblem of assigning students to stops. Experiments on 112 instances show that the proposed GRASP+VND matheuristic finds all known optimal solutions and this up to 800 times faster than an exact algorithm applied to the MIP formulation. The matheuristic can also produce very good solutions within 1 hour for realistic instances of 80 stops and 800 students. Our research efforts are now aimed in three directions. First, we are working on a cutting plane algorithm to obtain good lower bounds of the problem for larger instances. Secondly, we are investigating ways to exploit the problem structure of the school bus routing problem even more, *e.g.* to find out whether partial re-optimizations of the student allocation problem after certain moves are possible and to use student allocation problem information in specifically adapted neighborhoods. Thirdly, additional features may be added to the formulation to increase its realism. Such features include multiple buses visiting a single stop, time window constraints, multiple schools, and buses that do not start at the school.

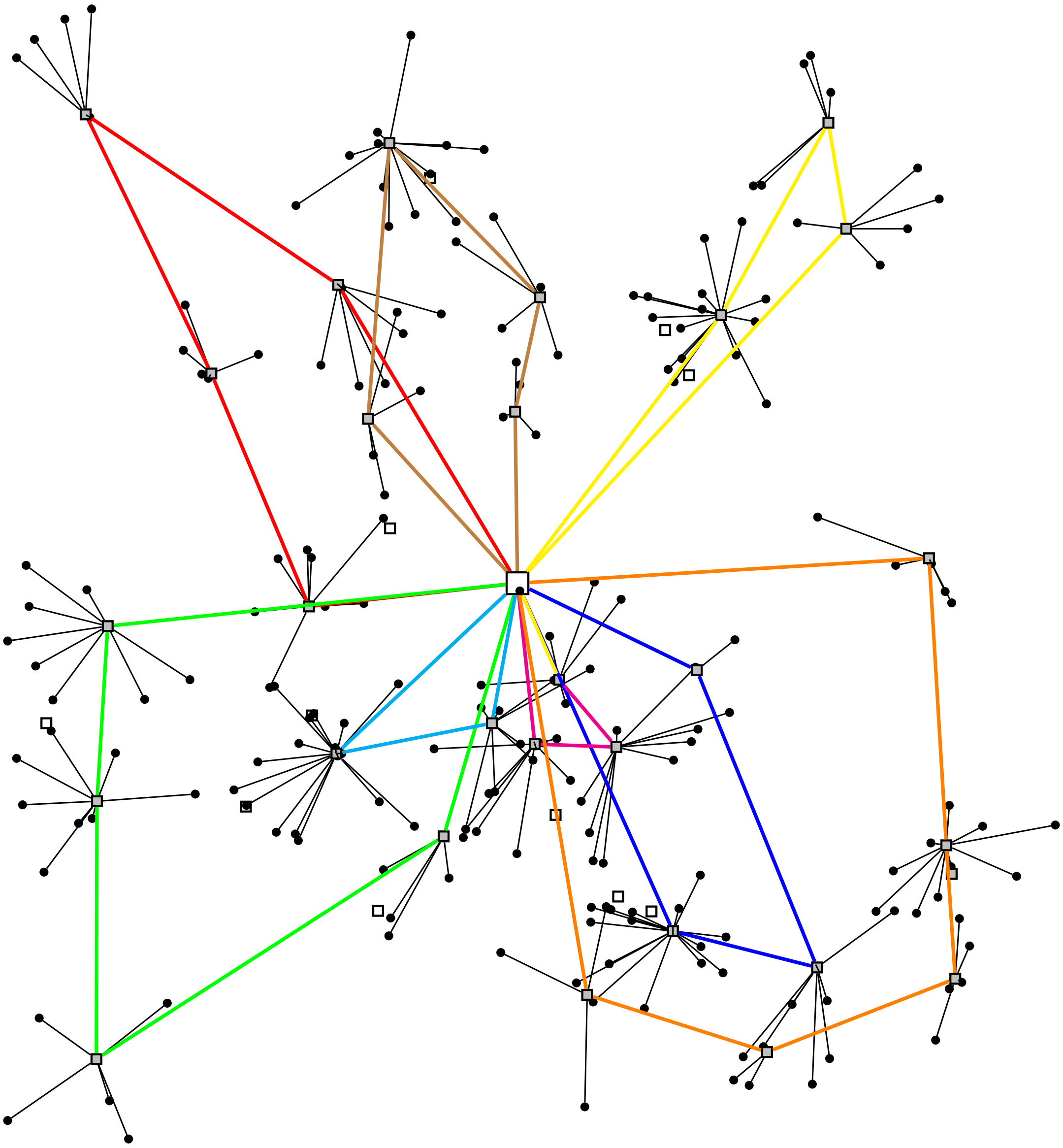


Figure 5: Best solution for 40 stops, 200 students, capacity 25 and maximum walking distance 10 (instance `SSSS-s40-u200-c25-w10`)

References

R. Baldacci, M. DellAmico, and J.J. Salazar Gonzalez. The capacitated m-ring star problem. Technical report, DISMI, University of Modena and Reggio Emilia, Italy, 2004.

B. Bennett and D. Gazis. School bus routing by computer. *Transportation Research*, 6: 317–326, 1972.

L. Bodin and L. Berman. Routing and scheduling of school buses by computer. *Transportation Science*, 13:113–129, 1979.

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.

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.

L. Chapleau, J. Ferland, and J. Rousseau. Clustering for routing in densely populated areas. *European Journal of Operational Research*, 20:48–57, 1985.

G. Clark and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.

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.

J. J. Dongarra. Performance of various computers using standard linear equations software, (Linpack benchmark report). Technical report CS-89-85, University of Tennessee, Knoxville, 2009. URL `http://www.netlib.org/benchmark/performance.ps`.

G. Dulac, J. Ferland, and P.-A. Fogues. School bus routes generator in urban surroundings. *Computers and Operations Research*, 7:199–213, 1980.

T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. 8:67–71, 1989.

T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. 8: 109–133, 1995.

L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.

M. Hachicha, J.M. Hodgson, G. Laporte, and F. Semet. Heuristics for the multi-vehicle covering tour problem. *Computers and Operations Research*, 27:29–42, 2000.

P. Hansen and N. Mladenović. Industrial applications of the variable neighbourhood search metaheuristic. In *Decisions and Control in Management Science*, pages 261–274, Boston, 2001a. Kluwer.

P. Hansen and N. Mladenović. Variable neighbourhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467, 2001b.

P. Hansen and N. Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5:207–226, 1997.

P. Hansen and N. Mladenović. An introduction to variable neighborhood search. In S. Voss, S. Martello, I. Osman, and C. Roucairol, editors, *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 433–458, Boston, 1999. Kluwer.

N. Mladenović. A variable neighborhood algorithm – a new metaheuristic for combinatorial optimization. In *Abstracts of Papers Presented at Optimization Days*, page 112, 1995.

S.R. Thangiah, B. Wilson, A. Pitluga, and W. Mennell. School bus routing in rural school districts. Working paper, Computer Science Department, Slippery Rock University, Slippery Rock, PA, USA, 2005.

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.

			GRASP+VND		MIP solver	
stops	students	capacity	CPU (s)	distance	CPU (s)	distance
5	25	25	0.16	141.01	0.987	141.01
5	50	25	0.39	286.68	7.567	286.68
5	100	25	1.15	360.35	31.54	360.35
10	50	25	1.55	242.85	-	-
10	100	25	2.93	407.20	-	-
10	200	25	8.49	735.27	-	-
20	100	25	8.85	520.24	-	-
20	200	25	26.39	915.71	-	-
20	400	25	234.66	1323.35	-	-
40	200	25	62.04	862.33	-	-
40	400	25	545.92	1433.20	-	-
40	800	25	3529.15	2900.14	-	-
80	400	25	946.21	1573.68	-	-
80	800	25	3433.78	2527.96	-	-
5	25	50	0.26	161.62	0.83	161.62
5	50	50	0.35	197.20	9.33	197.20
5	100	50	0.90	304.23	20.25	304.23
10	50	50	1.32	282.12	14.43	282.12
10	100	50	2.95	296.53	-	-
10	200	50	4.45	512.16	-	-
20	100	50	8.10	441.26	-	-
20	200	50	15.03	499.40	-	-
20	400	50	61.74	742.66	-	-
40	200	50	42.18	615.87	-	-
40	400	50	148.40	870.78	-	-
40	800	50	1933.24	1370.39	-	-
80	400	50	471.89	1048.56	-	-
80	800	50	3051.47	1545.51	-	-

Table 4: Summary results for instances with a maximum walking distance of 5

			GRASP+VND		MIP solver	
stops	students	capacity	CPU (s)	distance	CPU (s)	distance
5	25	25	0.39	182.14	3.572	182.14
5	50	25	0.43	193.55	20.21	193.55
5	100	25	2.08	294.21	148.95	294.21
10	50	25	2.45	244.54	1952.73	244.54
10	100	25	3.82	388.87	-	-
10	200	25	27.17	513.00	-	-
20	100	25	12.20	432.23	-	-
20	200	25	40.49	620.56	-	-
20	400	25	139.12	975.12	-	-
40	200	25	69.27	734.83	-	-
40	400	25	496.35	891.02	-	-
40	800	25	3495.62	2200.57	-	-
80	400	25	1647.16	1222.34	-	-
80	800	25	3245.85	1811.43	-	-
5	25	50	0.29	195.80	0.37	195.80
5	50	50	0.74	215.86	9.12	215.86
5	100	50	1.67	229.41	19.47	229.41
10	50	50	1.60	288.33	47.26	288.33
10	100	50	4.18	294.80	2005.70	294.80
10	200	50	12.09	475.21	-	-
20	100	50	15.73	365.82	-	-
20	200	50	28.76	462.77	-	-
20	400	50	73.23	614.67	-	-
40	200	50	51.57	489.55	-	-
40	400	50	173.45	779.77	-	-
40	800	50	2417.50	1037.46	-	-
80	400	50	576.26	760.61	-	-
80	800	50	3437.53	1189.03	-	-

Table 5: Summary for instances with a maximum walking distance of 10

			GRASP+VND		MIP solver	
stops	students	capacity	CPU (s)	distance	CPU (s)	distance
5	25	25	0.49	111.65	0.92	111.65
5	50	25	1.68	130.53	17.38	130.53
5	100	25	2.89	134.95	85.72	134.95
10	50	25	2.86	108.98	-	-
10	100	25	5.58	178.28	-	-
10	200	25	25.61	347.29	-	-
20	100	25	19.25	248.19	-	-
20	200	25	50.39	373.21	-	-
20	400	25	132.47	763.76	-	-
40	200	25	88.60	351.04	-	-
40	400	25	569.74	599.36	-	-
40	800	25	3389.94	1409.39	-	-
80	400	25	2143.93	587.04	-	-
80	800	25	3271.80	1119.10	-	-
5	25	50	0.52	103.18	3.43	103.18
5	50	50	1.69	96.26	13.83	96.26
5	100	50	2.88	144.42	17.96	144.42
10	50	50	2.28	157.48	129.70	157.48
10	100	50	7.98	175.96	-	-
10	200	50	20.58	217.46	-	-
20	100	50	13.82	185.88	-	-
20	200	50	36.83	257.57	-	-
20	400	50	90.54	298.47	-	-
40	200	50	68.02	274.24	-	-
40	400	50	242.91	395.95	-	-
40	800	50	2744.55	618.06	-	-
80	400	50	878.86	387.03	-	-
80	800	50	3327.46	637.16	-	-

Table 6: Summary for instances with a maximum walking distance of 20

			GRASP+VND		MIP solver	
stops	students	capacity	CPU (s)	distance	CPU (s)	distance
5	25	25	0.29	7.63	0.60	7.63
5	50	25	1.38	12.89	121.55	12.89
5	100	25	4.24	58.95	2655.31	58.95
10	50	25	2.84	32.25	-	-
10	100	25	7.38	57.50	-	-
10	200	25	33.35	102.93	59.78	102.93
20	100	25	20.17	53.19	-	-
20	200	25	67.73	93.01	-	-
20	400	25	307.20	239.58	1556.67	239.58
40	200	25	158.36	87.17	-	-
40	400	25	777.28	213.97	-	-
40	800	25	3506.44	399.77	-	-
80	400	25	2555.72	149.74	-	-
80	800	25	3454.54	349.65	-	-
5	25	50	0.25	25.64	6.80	25.64
5	50	50	1.17	30.24	122.58	30.24
5	100	50	2.89	39.44	1359.96	39.44
10	50	50	2.76	36.66	-	-
10	100	50	5.90	31.89	1679.45	31.89
10	200	50	18.50	56.61	-	-
20	100	50	23.73	19.05	-	-
20	200	50	46.04	46.66	-	-
20	400	50	127.08	84.49	-	-
40	200	50	139.33	63.28	-	-
40	400	50	382.78	76.58	-	-
40	800	50	2127.67	207.31	-	-
80	400	50	1734.78	98.39	-	-
80	800	50	3520.24	147.14	-	-

Table 7: Summary results for instances with a maximum walking distance of 40