European Journal of Operational Research 229 (2013) 518–528

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

A metaheuristic for the school bus routing problem with bus stop selection

Patrick Schittekat [a,d,*], Joris Kinable [b,e], Kenneth Sörensen [a], Marc Sevaux [c], Frits Spieksma [b], Johan Springael [a]

[a] ANT/OR, University of Antwerp, Operations Research Group, Belgium
[b] ORSTAT, KULeuven, Belgium
[c] Lab-STICC Laboratory, University of South Brittany, France
[d] SINTEF ICT, Department of Applied Mathematics, Norway
[e] CODeS, KU Leuven, KAHO Sint Lieven, Belgium

ARTICLE INFO

Article history:
Received 19 December 2011
Accepted 14 February 2013
Available online 27 February 2013

Keywords:
School bus routing problem
Bus stop selection
Metaheuristic
Matheuristic

ABSTRACT

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 joint problem of bus route generation and bus stop selection – two important sub-problems – in its most basic form. To this end, this paper defines 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 (s)he should walk to, and (3) determine routes that lie along the chosen stops, so that the total traveled distance is minimized. An MIP model of this basic problem is developed.

To increase the practical usefulness and to solve large instances of the SBRP, an efficient parameter-free GRASP + VND metaheuristic is developed. This method is a *matheuristic* since it uses an exact algorithm to optimally solve the sub-problem of assigning students to stops when routes are given. The results of this matheuristic approach on 112 artificially generated instances are compared to solutions found by a sequential method, to solutions obtained by implementing a MIP model in a commercial solver, and to a lower bound obtained by a dedicated column generation approach. Using appropriate statistical techniques, a neighborhood analysis is performed to test the design of the metaheuristic. Similarly, the characteristics of the problem instance that determine the computing time of the metaheuristic are discovered using statistical analysis. Finally, the importance of integrating all decisions in a single model is shown experimentally by comparing the metaheuristic to a sequential method.

Experiments show that the matheuristic exhibits excellent performance and finds optimal or close-to-optimal solutions of large instances of the SBRP in very limited computing times.

1. Introduction

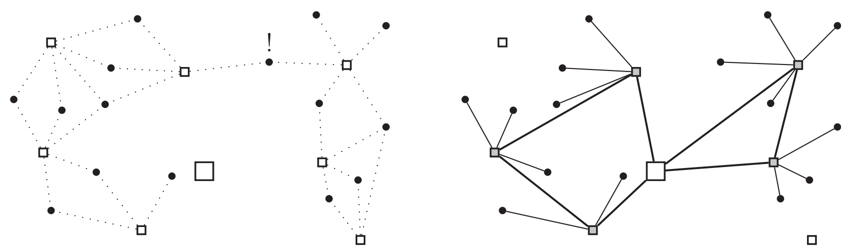
In the Flemish region of Belgium, students that live within certain minimum and maximum distances of their school are entitled to free transport to and from school. The transport is organized by the Flemish transportation company, which uses school buses that drive fixed routes. An additional requirement is that a bus stop should be located at a distance of not more than 750 meters 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 reach. To this end, a set of *potential* stops is determined first in such a way that each student lives within 750 meters of at least one stop. Routes are then determined for the school buses so that all students are picked up at a stop they can reach, while assuring 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 seven 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. Thus, 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,

* Corresponding author at: ANT/OR, University of Antwerp, Operations Research Group, Belgium. Tel.: +47 92014840.

E-mail addresses: patrick.schittekat@ua.ac.be (P. Schittekat), joris.kinable@econ.kuleuven.be (J. Kinable), kenneth.sorensen@ua.ac.be (K. Sörensen), marc.sevaux@univ-ubs.fr (M. Sevaux), frits.spieksma@econ.kuleuven.be (F. Spieksma), johan.springael@ua.ac.be (J. Springael).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.02.025



(a) An instance (b) A possible solution

Fig. 1. Example school bus routing problem.

(2) determine for each student which stop (s)he should move to, and (3) determine routes that visit the selected stops, such that the total distance travelled by the buses is minimized. Fig. 1a shows an example of this type of problem, which we call the *school bus routing problem* or SBRP. In this figure, dots represent students, small squares represent potential stops and a large square represents the school. Dotted lines indicate which stops a student is able to reach.

If, for example, the capacity of each bus equals 8, a possible (but not necessarily optimal) solution to this problem is shown in Fig. 1b.

In the problem discussed in this paper, it is assumed that all students represent a unit to be transported and that the capacity of the buses can be expressed as an integral number of units. Students who can reach the school are not taken into account.

As mentioned earlier, the SBRP under consideration consists of three interrelated sub-problems. Two of these sub-problems, (1) and (3), have a direct impact on the total traveled distance by all vehicles whereas sub-problem (2), the allocation of students to selected bus stops, only indirectly affects the objective function, because it merely determines whether a combination of routes and selected bus stops is feasible or not. Therefore, it seems intuitively logical to treat sub-problem (1) and (3) on the same level.

When the stops to use and the routes that visit these stops have been determined, students need to be assigned to stops. When a student can be assigned to multiple stops along the same route, the allocation of this student to a particular stop is arbitrary. This is not the case if a student can be assigned to multiple stops in *different* routes. All students that can be assigned to multiple routes need to be distributed over those routes in such a way that the capacity of the buses is not exceeded. In Fig. 1a and b, there is one student who can reach a stop in both of the routes. However, given that the capacity of each bus is 8, this student needs to be assigned to the route on the right (Fig. 1b). Compared to traditional vehicle routing problems, the possibility to assign students to different stops offers the possibility to incur potential savings; at the same time, it introduces an extra decision level that makes the problem much more difficult to solve.

Apart from school bus routing applications, 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 (such as the European distriution network of Kiala) 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 are asked to specify more than one drop-off point and that the parcel delivery company will then choose among the ones selected 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.

Metaheuristics have been proven successful on large instances for the basic Vehicle Routing Problem, but have not attracted much attention to solve SBRPs (Park and Kim, 2010). In the present paper, the emphasis lies on developing a practical, parameter-free metaheuristic. A good metaheuristic should generate high-quality solutions in little time without the need for excessive parameter tuning. Furthermore, it should be possible to incorporate the generated solution into existing routing software without much effort. This is one of the main reasons why tabu search, although very successful in academic literature, is practically non-existent in commercial vehicle routing software (Sörensen et al., 2008). Commercial routing software use simple diversification strategies such as Iterated Local Search or apply multiple neighborhoods to overcome the myopic behavior of a single neighborhood. These local search neighborhoods are almost never accompanied by intricate, complex diversification techniques with a lot of parameters. The additional complexity would render it difficult to maintain the software, or to adapt it to different problems.

2. Literature review

Contrary to the literature on the ordinary vehicle routing problem (VRP) and several of its extensions (e.g. time windows), only a limited amount of research has considered the routing of school buses.

In their book on the traveling salesman problem (TSP), Applegate et al. (2007), mention schoolbus routing as one of the early applications motivating the TSP. However, in their context selecting stops is not part of the problem.

Typically, school bus 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) minimize total travel time of all children. 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. Li and Fu (2002) present a multi-objective approach in which both the number of buses, the travel time of the buses and the travel times of the students are minimized. Recently, some research has been done on the mixed-load school bus routing problem, allowing students from different schools to travel on the same buses (e.g., Park et al., 2012).

In their recent survey on the School Bus Routing problem, Park and Kim (2010) mention five different sub-problems, which are often treated separately in the SBPR literature: data preparation, bus stop selection, bus route generation, school bell time adjustment, and bus scheduling. The authors emphasize that there is a strong need to integrate these sub-problems. This work treats the selection of bus stops and bus route generation simultaneously, making it one of few works (a recent exception being Riera-Ledesma and Salazar-Gonzlez, 2013) which treats 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, then stops are selected so that as many students as possible can reach a stop. The school bus routing problem discussed in Bowerman et al. (1995) includes a maximum walking distance for a student to his or her 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. Other references incorporating bus stop selection and routing are Bodin and Berman (1979) and Desrosiers et al. (1980). Most solution techniques developed for the SBRP with bus stop selection are characterized by a sequential approach. In other words, a bus stop selection procedure and a routing procedure are performed one after another. Either the bus stops are selected first after which a routing procedure takes place or the other way around. These two strategies are called Location–Allocation–Routing (LAR) and Allocation–Routing–Location (ARL) strategies respectively and are closely related to similar strategies developed for location–routing problems (Laporte et al., 1988). Location, allocation and routing represent sub-problems (1)–(3) described in Section 1 respectively. The LAR strategy has the unfortunate drawback that it generates solutions with an excessive number of routes since it fails to take into account the effect of student allocation on the capacity of the routes. By reversing the order, ARL strategies attempt to overcome this obstacle. In the metaheuristic developed in this paper both sub-problems are treated in an integrated fashion. Although, metaheuristics are successfully applied to solve VRPs, the survey of Park and Kim (2010) also shows that it is surprisingly not the case for the SBRP. In their opinion, metaheuristics for the SBRP are an excellent research avenue.

Similar problems outside the SBRP context exist. 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. The capacitated *m*-ring star problem (Baldacci et al., 2004) differs from the school bus routing problem described in this paper 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), instead an assignment cost is given. Moreover, the number of rings (tours) is pre-specified. Baldacci and Dell'Amico (2010) apply a tabu search metaheuristic to solve larger sized instances, but for the solution technique to work a rather large number of parameters have to be set.

In summary, previous work on SBRP can be improved by developing a metaheuristic which enables a strong integration between the bus stop selection sub-problem and the bus route generation sub-problem. This is exactly the aim of this paper. Moreover, the aim of this paper is to develop an effective metaheuristic which is simple to implement and does not require any parameters to be specified. For small-sized random benchmark instances, the effectiveness of the proposed metaheuristic is evaluated against outcoms of solving a MIP (see Schittekat et al. (2006)). Also, results for larger-sized instances obtained by the metaheuristic are given, and they are compared to lowerbounds found by applying a column generation approach to a set-partitioning formulation of the problem. We consider instances containing up to 80 stops and 800 students.

The resulting *matheuristic* consists of two phases. The construction phase uses ideas from GRASP (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 sub-problem is solved exactly by modeling the problem as a transportation problem (see Section 4.3).

3. Problem formulation

In this section we present an integer programming formulation of the SBRP. Recall that the SBRP as presented here contains a single school, one type of students, and identical buses, each with fixed capacity. The standard vehicle routing criterion is optimized: the total distance traveled by all vehicles. Clearly, the school bus routing problem as described here is a generalization of the basic vehicle routing problem and therefore also NP-hard. It is assumed that the problem is defined on a directed graph. The following formulation builds on the formulation of Toth and Vigo (2001, p. 15). Table 1 discusses the symbols used in the model.

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

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

$$\text{s.t.} \quad \sum_{j\in V} x_{ijk} = \sum_{j\in V} x_{jik} = y_{ik} \quad \forall i \in V,\ k=1,\ldots,n \tag{2}$$

$$\sum_{i,j\in Q} x_{ijk} \leqslant |Q|-1 \quad \forall Q \subseteq V\setminus\{v_0\}, \quad \forall k \tag{3}$$

$$\sum_{k=1}^{n} y_{ik} \leqslant 1 \quad \forall i \in V\setminus\{0\} \tag{4}$$

$$\sum_{k=1}^{n} z_{ilk} \leqslant s_{il} \quad \forall l \in S, \quad \forall i \in V \tag{5}$$

$$\sum_{i\in V}\sum_{l\in S} z_{ilk} \leqslant C \quad k=1,\ldots,n \tag{6}$$

$$z_{ilk} \leqslant y_{ik} \quad \forall i,l,k \tag{7}$$

$$\sum_{i\in V}\sum_{k=1}^{n} z_{ilk} = 1 \quad \forall l \in S \tag{8}$$

$$y_{ik} \in \{0,1\} \quad \forall i \in V,\ k=1,\ldots,n \tag{9}$$

$$x_{ijk} \in \{0,1\} \quad \forall i,j \in V, i\neq j,\ k=1,\ldots,n \tag{10}$$

$$z_{ilk} \in \{0,1\} \quad \forall i,j \in V, i\neq j,\ l \in S \tag{11}$$

The objective function (1) minimizes the total distance travelled by all buses. Constraints (2) enforce that if stop i is visited by bus k,

Table 1
Symbols used in the mathematical model.

Data	
C	Capacity of a bus
V	Set of potential stops, with $\lvert V\rvert = n$
E	Set of arcs between stops
S	Set of students
c_{ij}	Cost of traversing the arc from stop i to stop j
s_{il}	1 if student l can reach stop i and 0 otherwise
$i = 0$	Index for the school
Decision variables	
x_{ijk}	1 if bus k traverses the arc from i to j, 0 otherwise
y_{ik}	1 if bus k visits stop i, 0 otherwise
z_{ilk}	1 if student l is picked up by bus k at stop i, 0 otherwise

then an arc should be traversed by bus k entering stop i and leaving stop i, while constraints (3) impose the connectivity of the route performed by bus k. These constraints serve as subtour elimination constraints. Constraints (4) guarantee that each stop is visited at most once, except for the stop corresponding to the school. Constraints (5) ensure that each student is picked up at a stop he or she can reach. Constraints (6) make sure that the capacity of the buses is not exceeded. Constraints (7) impose that student l is not picked up at stop i by bus k if bus k does not visit stop i. Constraints (8) enforce that each student is picked up once. Finally, constraints (9)–(11) require that all decision variables are binary.

By using this formulation, a number of assumptions are implicitly made. 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 implies 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, 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, a hybrid exact/metaheuristic procedure is developed to solve large instances of the school bus routing problem. This *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 sub-problem 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. Selecting elements and updating 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 the metaheuristic is based on the well-known Clarke–Wright savings heuristic (Clarke 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 Clarke–Wright heuristic, the GRASP procedure starts from a solution in which each stop is visited in a separate route. After this initial setup, students are assigned to these stops by solving the student allocation sub-problem (see Section 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 sub-problem algorithm).

To generate different solutions, the GRASP construction heuristic adopts a method which is free of parameters 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. A similar roulette wheel procedure was successfully applied by Drexl (1991) in the context of project network scheduling. 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 sub-problems 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

Algorithm 1: GRASP construction phase for the SBRP

input : initial solution with one route per stop
Calculate Clark-Wright savings matrix $\sigma_{ij} = c_{i0} + c_{0j} - c_{ij}$;
Create list of stop pairs L containing all pairs (i, j);
repeat
  Calculate probability of selecting stop pair $(i, j) \in L$ as $p_{ij} = \frac{\sigma_{ij}}{\sum_{i,j} \sigma_{ij}}$;
  Roulette wheel selection: select stop pair $(i, j) \in L$ with probability p_{ij};
  if *Connecting stops i and j yields a feasible solution* **then**
    Connect stops i and j;
  Remove pair (i, j) from L;
until $L = \emptyset$;

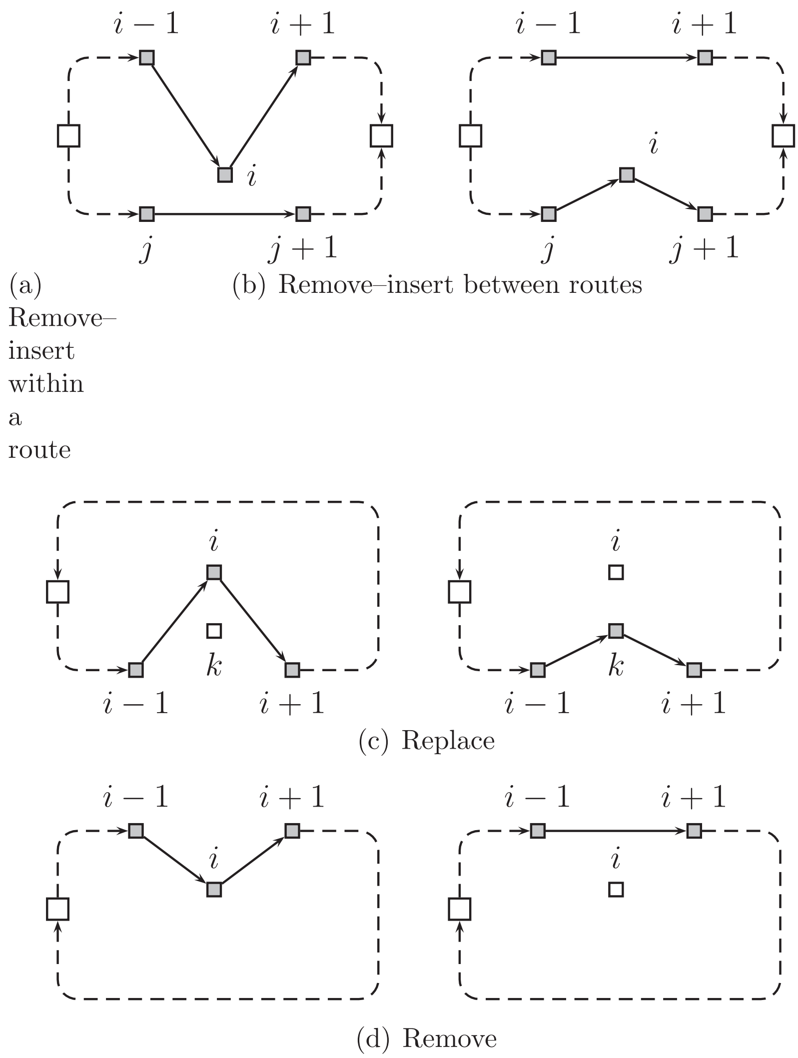


Fig. 2. The figures on the left correspond to an original solution while the figures on the right visualize a new solution after performing a move of the given move type. The depot is represented by one or two white squares.

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. Diversity is introduced by the different starting solutions generated by the GRASP construction phase and a perturbation phase is not needed. The variable neighborhood *descent* or VND variant is therefore used. Pseudo-code for the VND improvement phase is given in Algorithm 2.

Algorithm 2. Variable neighborhood descent for the SBRP

Algorithm 2: Variable neighborhood descent for the SBRP

input : Solution x obtained by GRASP, k_{max} move types
$k \leftarrow 1$;
repeat
 Local Search: perform local search using neighborhood $\mathcal{N}_k$, starting from solution x until it cannot be improved and local optimum x' is reached;
 if *x' is better than x* **then**
  $x \leftarrow x'$ (center the search around the new solution) and $k \leftarrow 1$ (search again using the first neighborhood);
 else
  $k \leftarrow k + 1$;
until $k = k_{max}$;

Our VND improvement phase uses four neighborhood structures. The neighborhoods are used in the order presented below: Fig. 2 visualize all four different moves.

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 sub-problem. 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 sub-problem 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 sub-problem is solved.

To save time generating solutions in a neighborhood, the following strategy is adopted. 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. It is important to 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, it was 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.

These neighborhoods, which guide both the selection of bus stops and the routing of the selected bus stops, are all equally important to find promising solutions and as a consequence the bus stop selection sub-problem and the routing sub-problem are treated on the same level. The student allocation problem is used to check for feasibility of a potential solution.

4.3. Solving the student allocation sub-problem exactly

In the metaheuristic solution method, the SBRP is decomposed in a Master Problem and a sub-problem. The Master Problem is a school bus routing problem with bus stop selection, the objective of which is to minimize the total travelled distance. Once the stops have been selected and the routes have been fixed, a sub-problem remains of allocating students to stops in such a way that the capacity of the buses is not exceeded. This sub-problem 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 sub-problem can be written as an optimization problem as follows:

$$\min \quad \sum_{l\in S}\sum_{k=1}^{n} t_{lk} z'_{lk} \tag{12}$$

$$\text{s.t.} \quad \sum_{k=1}^{n} z'_{lk} = 1 \quad \forall l \in S \tag{13}$$

$$\sum_{l\in S} z'_{lk} \leqslant C \quad \forall k = 1,\ldots,n \tag{14}$$

$$z'_{lk} \in \{0,1\} \quad \forall k = 1,\ldots,n,\ l \in S \tag{15}$$

In this formulation $z'_{lk} = \sum_{i\in V} y_{ik} z_{ilk}$ is a decision variable equal to 1 if student l is picked up in route k. The parameter t_{lk} 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 route k and 1 otherwise. Constraints (13) ensure that each student is assigned to exactly one route. Constraints (14) ensure that the capacity of the buses is not exceeded.

The objective function (12) 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. The assignment of students to routes has the structure of a transportation problem in which each student is a supply source (with a capacity of one) and each route is a demand sink (with a demand equal to the bus capacity). If a solution with a cost of zero exists, a feasible assignment of students to stops also exists. A well-known property of the transportation problem is that – if supply and demand are integer – all basic solutions are integer too. This means that the binary constraints can be relaxed (15) when solving the assignment of students to routes.

In the matheuristic, the transportation problem is solved using the well-known primal–dual labeling method of Ford and Fulkerson (1962).

5. Experiments

5.1. Problem instance generation

An instance generator for the SBRP has been developed that can generate random problem instances of any size. The generator requires five 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 plane defined by (0,0) and (x_{max}, y_{max}). It first generates n_p stops in this rectangle. 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 $(x_i + w_j\cos\alpha_j,\ y_j + w_j\sin\alpha_j)$.

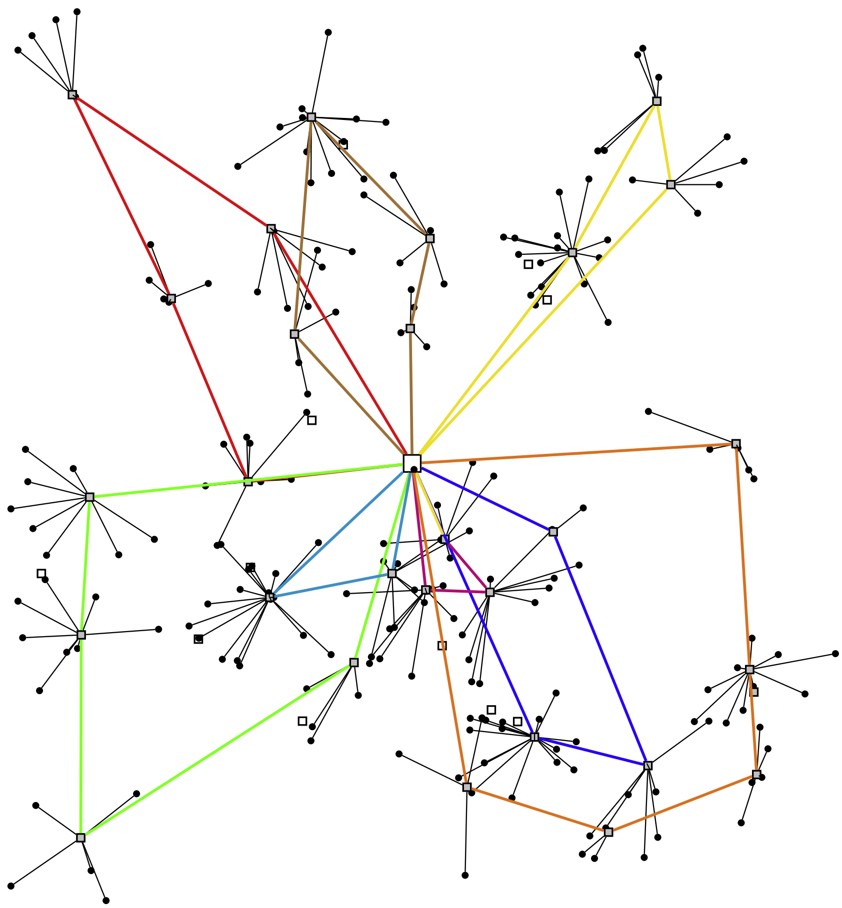


Fig. 3. Best solution for 40 stops, 200 students, capacity 25 and maximum walking distance 10 (instance SSSS-s40-u200-c25-w10).

The 112 instances considered for the experiments in this paper are available can be downloaded from http://antor.ua.ac.be/school-bus-routing. 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 Fig. 3 is called SSSS-s40-u200-c25-w10.

In order to assess the quality of the solutions produced by the metaheuristic, two approaches are used. In a first approach, the formulation developed in Section 3 is solved exactly using IBM ILOG CPLEX (Section 5.2). Since only the easiest instances can be solved in this way, a dedicated column generation approach was also developed (Section 5.3). Finally, computational results are discussed in Section 5.4.

5.2. Exact solution

The MIP formulation presented in Section 3 has been implemented using the commercial MIP solver IBM ILOG CPLEX version 12.2. Since in the original formulation an exponential number of subtour elimination constraints ($O(2^n)$) are required, those constraints have been replaced by the Miller–Tucker–Zemlin constraints (Miller et al., 1960) in the actual implementation (u_i represents the rank order in which stop i is visited):

$$u_i + u_j + nx_{ij} \leqslant n - 1 \quad \forall i,j = 2,\ldots,n,\ i \neq j \tag{16}$$
$$u_1 = 1 \tag{17}$$

Although this formulation yields a weaker LP relaxation (Langevin et al., 1990) in comparison with the LP relaxation of the formulation given in Section 3, only n extra constraints are required which makes this formulation suitable for larger instances. Notwithstanding the fact that there exist better ways to deal with subtour elimination using sophisticated cutting and separation techniques, developing a state-of-the art exact algorithm is beyond the scope of this paper. Moreover, it is not expected that improving the exact algorithm will result in different conclusions with regard to the effectiveness and performance of the developed metaheuristic.

The results of the metaheuristic in comparison with the exact solution are given in Table 5 in Appendix A.

5.3. Lower bound

When given a maximum runtime of 2 hours, the exact MIP algorithm (Section 5.2) is only able to solve the easiest 45 instances. The largest instance solved to optimality within the given timespan has 10 stops and 200 students. To evaluate the metaheuristic's performance on larger instances, strong lower bounds need to be obtained on the optimal solutions of these instances. To this end a dedicated column generation approach has been developed.

Recall the MIP formulation given in Section 3. This MIP can be reformulated to the following equivalent formulation (Master Problem):

$$\min \quad \sum_{p \in P} \delta_p z_p \tag{18}$$
$$\text{s.t.} \quad \sum_{p \in P} r_{vp} z_p \leqslant 1 \quad \forall v \in V \tag{19}$$
$$\sum_{p \in P} t_{sp} z_p = 1 \quad \forall s \in S \tag{20}$$
$$z_p \in \{0,1\} \quad \forall p \in P \tag{21}$$

Table 2
Symbols used in the mathematical model.

Data	
δ_p	Cost induced by schedule p
t_{sp}	1 if student s is picked up in bus schedule p, 0 otherwise
r_{vp}	1 if stop v is part of bus schedule p, 0 otherwise
Decision variables	
z_p	1 if bus schedule p is used, 0 otherwise

Table 2 discusses the symbols used in this formulation. Here, p is an index which corresponds to a bus route. More precisely, p defines a complete, valid, *bus schedule*: a sequence of stops the bus driver should visit, and the specific students that should be picked up at the corresponding stops. The set of all feasible bus schedules is denoted by P. The cost δ_p associated with each bus schedule is the travel distance required to visit all stops on the schedule. Furthermore, similar to constraints (8) and (4) in the original formulation, constraints (19) and (21) respectively ensure that each student is picked up at some stop, and that no stop is visited more than once. When the integrality constraints are replaced by the weaker constraints $z_p \geqslant 0$, a Linear Program relaxation of the Master Problem is obtained, called LPM (see Vanderbeck (1994), Wolsey (1998)). Solving LPM gives a lower bound on the optimal value of the problem. Solving LPM directly would be impractical (if not impossible) due to the exponential number of possible bus schedules p. Therefore, a simpler problem, commonly referred to as the Restricted Master Problem (RMP) is considered instead. Contrary to the LPM, the Restricted Master Problem only utilizes a small subset of bus schedules $P' \subseteq P$. The resulting linear problem is small and can be solved by any MIP solver.

Following a standard approach (see e.g. Wolsey (1998), Chvátal (1983)), a pricing problem needs to be solved to establish whether a solution of the RMP is an optimal one of LPM. The pricing problem searches for columns with negative reduced cost, i.e. bus schedules for which,

$$\delta_p + \sum_{v \in V} r_{vp} w_v - \sum_{s \in S} t_{sp} u_s \tag{22}$$

yields a negative value, where w_v and u_s are the dual variables corresponding with constraints (19) and (21) respectively. Each time a column with negative reduced cost is found, the column is added to the subset P' and the RMP is recomputed. This procedure is repeated until no more columns price out. Once this point is reached, by the Strong Duality Theorem, an optimal solution to the linear relaxation of the Master Problem is obtained (see e.g. Chvátal (1983)).

Notice that the primary interest here is computing the bounds that arise from solving LPM, and to compare these bounds with the solutions found by the heuristic. Thus, only a high-level description of the column generation approach is given. Full details of the procedure can be found in Kinable et al. (submitted for publication).

The pricing problem is solved using an enumerative procedure. When bus schedules with negative reduced costs are detected, the corresponding variables are added to the LP, after which the RMP is re-solved. A starting solution is found by solving a transportation problem similar to the one described in Section 4.3, where students are represented by demand nodes with demand 1, stops are represented by supply nodes with supply equal to the bus capacity, and there is an arc between a student-node and a stop-node. Next, a feasible solution is constructed by having a bus schedule of the form: school-stop-school for each populated stop.

5.4. Computational results

To test the GRASP + VND matheuristic, experiments were conducted on 112 instances. The instance sizes range from 5 stops and 25 students to 80 stops and 800 students. Also, four maximum walking distances are 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 exist in the student allocation sub-problem. In Table 5 (Appendix A), the results of the experiments are reported.

Table 3
Model CPU time: summary of fit.

Measure	Value
R^2	0.946214
R^2 adj	0.940889
Root mean square error	274.0761
Mean of response	595.4923
Observations (or sum Wgts)	112

Table 4
Model CPU time.

Source	Nparm	DF	DFDen	*F* Ratio	Prob > *F*
Stops	4	4	6651525	22.1370	<.0001
Students	5	5	52542237	139.8931	<.0001
Capacity	1	1	980123	13.0478	0.0005

The table provides details on the problem instances and shows for every instance 11 columns: number of stops (column *stop*), number of students (column *stud*), bus capacity (column *cap*) and maximum walking distance (column *wd*). The table also reports the results of the GRASP + VND matheuristic (column *MH*), and an overview of the exact solutions (column *MIP*) and lower bounds (column *LB*). Furthermore, for each lower bound value, a (*i*) in the column column *LB* indicates whether the corresponding LB solution is an integer solution. When this is the case the optimal solution of the relaxed Master Problem LPM coincides with the optimal solution of the Master Problem. Finally, the column t_{MH} gives for each instance the runtime (in seconds) of the matheuristic.

Exact solutions are only reported whenever the optimal solution was found within 2 hours. Similarly, lower bounds obtained via the column generation approach are reported whenever the Master Problem could be solved to optimality within 2 hours. For the largest instances, hardware limitations such as lack of sufficient memory prevented running the column generation procedure to completion. Finally, two gaps are reported. The first gap (column Δ_{LB}) denotes the percentage difference between the lower bound *LB* and the optimal solution of the MIP formulation. The second gap (column Δ_{MH}) shows the difference between the result of the matheuristic and the exact solution. In case the exact solution is unavailable, the lower bound *LB* is used instead. In this case, the gap is due to either the difference between the solution of the matheuristic and the optimal solution, between the lower bound and the optimal solution, or both.

When comparing the results of the GRASP + VND matheuristic with the exact solutions (instances 1–42, 44), one can observe that for all except two instances the optimal solution has been found. Continuing this comparison for the matheuristic and the instances which could not be solved by the exact approach, it can be observed that the average gap between the lower bounds and the matheuristic results is 1.4%. Consequently, we conclude that the heuristic finds most known optimal solutions and finds solutions within a few percent of the lower bound in those cases where the optimal solution could not be found. In addition, it follows from the results in Table 5 that the bounds resulting from solving the school bus routing problem by column generation are quite close to the integer optimum. This latter finding should be

Table 5
Computational results.

ID	*stop*	*stud*	*cap*	*wd*	*MH*	*MIP*	*LB*	Δ_{LB}	Δ_{MH}	t_{MH}
1	5	25	25	5	141.01	141.01	141.01(*i*)	0	0	0.16
2	5	25	50	5	161.62	161.62	161.62(*i*)	0	0	0.26
3	5	25	25	10	182.14	182.14	182.14(*i*)	0	0	0.39
4	5	25	50	10	195.80	195.8	195.8(*i*)	0	0	0.29
5	5	25	25	20	111.65	111.65	111.65(*i*)	0	0	0.49
6	5	25	50	20	103.18	103.18	103.18(*i*)	0	0	0.52
7	5	25	25	40	7.63	7.63	7.63(*i*)	0	0	0.29
8	5	25	50	40	25.64	25.64	25.64(*i*)	0	0	0.25
9	5	50	25	5	286.68	286.68	281.49	1.84	0	0.39
10	5	50	50	5	197.20	197.2	197.2(*i*)	0	0	0.35
11	5	50	25	10	193.55	193.55	181.02	6.92	0	0.43
12	5	50	50	10	215.86	215.86	215.86(*i*)	0	0	0.74
13	5	50	25	20	130.53	130.53	130.53(*i*)	0	0	1.68
14	5	50	50	20	96.26	96.26	96.26(*i*)	0	0	1.69
15	5	50	25	40	12.89	12.89	12.89(*i*)	0	0	1.38
16	5	50	50	40	30.24	30.24	30.24(*i*)	0	0	1.17
17	5	100	25	5	360.35	360.35	360.35(*i*)	0	0	1.15
18	5	100	50	5	304.23	304.23	290.67	4.67	0	0.9
19	5	100	25	10	294.21	294.21	255.93	14.96	0	2.08
20	5	100	50	10	229.41	229.41	229.41(*i*)	0	0	1.67
21	5	100	25	20	134.95	134.95	134.95	0	0	2.89
22	5	100	50	20	144.41	144.41	139.87	3.25	0	1.34
23	5	100	25	40	58.95	58.95	58.95	0	0	4.24
24	5	100	50	40	39.44	39.44	39.44	0	0	2.89
25	10	50	25	5	242.85	242.85	242.85(*i*)	0	0	1.55
26	10	50	50	5	282.12	282.12	282.12(*i*)	0	0	1.32
27	10	50	25	10	244.54	244.54	244.54(*i*)	0	0	2.45
28	10	50	50	10	288.33	288.33	288.33(*i*)	0	0	1.6
29	10	50	25	20	108.98	108.98	108.95	0.03	0	2.86
30	10	50	50	20	157.48	157.48	157.48(*i*)	0	0	2.28
31	10	50	25	40	32.25	32.25	32.25	0	0	2.84
32	10	50	50	40	36.66	36.66	36.66(*i*)	0	0	2.76
33	10	100	25	5	403.18	403.18	400.54	0.66	0	0.90
34	10	100	50	5	296.53	296.53	294.11	0.82	0	0.54
35	10	100	25	10	388.87	388.87	369.62	5.21	0	3.82
36	10	100	50	10	294.80	294.8	294.8(*i*)	0	0	4.18
37	10	100	25	20	178.28	178.28	178.28	0	0	5.58
38	10	100	50	20	175.96	175.96	175.41	0.31	0	7.98
39	10	100	25	40	57.50	57.5	57.5	0	0	7.38
40	10	100	50	40	31.89	31.89	31.89	0	0	5.9
41	10	200	25	5	735.27	735.27	735.27(*i*)	0	0	8.49
42	10	200	50	5	512.16	506.06	506.06(*i*)	0	1.21	4.45
43	10	200	25	10	513.00		463.78		10.61	27.17
44	10	200	50	10	475.21	475.21	458.17	3.72	0	12.09
45	10	200	25	20	347.29		331.49		4.77	25.61
46	10	200	50	20	217.46		194.66		11.71	20.58
47	10	200	25	40	102.93		102.93		0	33.35
48	10	200	50	40	55.05		55.05		0	13.05
49	20	100	25	5	520.24		507.81(*i*)		2.45	8.85
50	20	100	50	5	420.64		406.65(*i*)		3.44	3.40
51	20	100	25	10	422.21		404.78		4.31	7.84
52	20	100	50	10	360.86		356.52		1.22	3.88
53	20	100	25	20	245.17		245.17		0	10.32
54	20	100	50	20	185.06		181.3		2.07	5.60
55	20	100	25	40	52.52		52.52		0	10.40
56	20	100	50	40	19.05		19.05		0	23.73
57	20	200	25	5	903.84		851.98		6.09	10.60
58	20	200	50	5	485.65		473.89		2.48	29.27
59	20	200	25	10	616.93		589.89		4.58	28.85
60	20	200	50	10	462.31		451.09		2.49	18.48
61	20	200	25	20	373.21		366.1		1.94	50.39
62	20	200	50	20	250.75		246.49		1.73	26.94
63	20	200	25	40	93.01		93.01		0	67.73
64	20	200	50	40	45.40		45.4		0	33.50
65	20	400	25	5	1323.35		1247.65		6.07	234.66
66	20	400	50	5	733.54		709.87		3.33	37.64
67	20	400	25	10	975.12		911.06		7.03	139.12
68	20	400	50	10	614.67		599.16		2.59	73.23
69	20	400	25	20	763.76		756.04		1.02	132.47
70	20	400	50	20	298.47		298.05		0.14	90.54
71	20	400	25	40	239.58		239.58		0	307.2
72	20	400	50	40	84.49		84.49		0	127.08
73	40	200	25	5	831.94		787.14		5.69	60.10
74	40	200	50	5	593.35					40.00

Table 5 (*continued*)

ID	stop	stud	cap	wd	MH	MIP	LB	Δ_{LB}	Δ_{MH}	t_{MH}
75	40	200	25	10	728.44		696.04		4.65	709.28
76	40	200	50	10	481.05					91.71
77	40	200	25	20	339.75		328.19		3.52	153.04
78	40	200	50	20	273.88		273.05		0.3	53.84
79	40	200	25	40	76.77		76.77		0	132.52
80	40	200	50	40	58.46		58.46		0	77.92
81	40	400	25	5	1407.05		1307.52		7.61	353.09
82	40	400	50	5	858.80					585.98
83	40	400	25	10	891.02		869.38		2.49	496.35
84	40	400	50	10	757.42					413.29
85	40	400	25	20	586.29		575.66		1.85	739.56
86	40	400	50	20	395.95					242.91
87	40	400	25	40	195.33		195.33		0	1186.56
88	40	400	50	40	70.77		70.77		0	549.07
89	40	800	25	5	2900.14		2801.05		3.54	3529.15
90	40	800	50	5	1345.70		1280.51		5.09	1257.96
91	40	800	25	10	2200.57		2153.76		2.17	3495.62
92	40	800	50	10	1025.16		978.88		4.73	3600.03
93	40	800	25	20	1404.16		1404.16		0	3600.18
94	40	800	50	20	616.58		613.72		0.47	3600.12
95	40	800	25	40	396.92					3600.81
96	40	800	50	40	200.94					3074.14
97	80	400	25	5	1546.23					958.12
98	80	400	50	5	1048.56					471.89
99	80	400	25	10	1216.74					1833.44
100	80	400	50	10	760.61					576.26
101	80	400	25	20	565.49					1224.64
102	80	400	50	20	372.05					878.86
103	80	400	25	40	131.75					1116.28
104	80	400	50	40	95.84					3600.05
105	80	800	25	5	2527.96					3433.78
106	80	800	50	5	1530.58					3600.03
107	80	800	25	10	1809.90					3600.05
108	80	800	50	10	1187.51					3600.04
109	80	800	25	20	1110.44					3600.10
110	80	800	50	20	623.03					3600.62
111	80	800	25	40	311.41					3600.21
112	80	800	50	40	126.06					3600.05

contrasted with the bounds obtained via the LP relaxation of the original problem (Section 3) as reported in Schittekat (2010): not surprisingly, these bounds are much weaker.

5.5. Algorithm analysis

5.5.1. Determinants of CPU time

A multi-way ANOVA model was constructed in order to determine which problem characteristics (number of stops, number of students, capacity of the bus, maximum walking distance) have a strong influence on the CPU time of the algorithm. The final model data is summarized in Tables 3 and 4. The R^2 statistic of the model, i.e. the proportion of variation around the mean solution quality that can be explained by the model, is a good measure of the accuracy of the model. In this case, the model explains approximately 95% of total variation around the mean. Variables with a significant effect are the number of stops, the number of students and the capacity of the buses. The effect on the CPU time of the maximum walking distance is not significant.

5.5.2. The importance of integrated decision-making

As mentioned, the matheuristic developed in this paper is an *integrated* method, in which all decisions are taken simultaneously. To quantify the advantages of such an approach a *sequential* method was developed in which the assignment and routing decisions are taken in two distinct steps executed in sequence.

In the first step, students are assigned to stops. The aim of the heuristic assignment procedure is to minimize the number of stops used. The procedure starts creating a list of all students, sorted in increasing order of the number of allowable stops (i.e., stops they are allowed to walk to). It then proceeds by iterating the following two steps until all students have been assigned. First, students that have only one allowable stop are selected and assigned to this stop. Secondly, the stop with the largest number of allowable students is selected and as many students are assigned to this stop as possible. Students are assigned in increasing order of the number of stops that they can go to. Finally, the list of unassigned students is updated to reflect the new situation.

When students have been assigned to stops, the remaining stops are closed and the resulting vehicle routing problem is solved using the same GRASP + VND procedure as the one used in the integrated method, except that a student is only allowed to go to the stop to which (s)he has been assigned to in the first phase. Similar to the integrated method, the sequential method was also given a maximum of 1 hour of CPU time.

Results show that the sequential method performs, on average, 23% worse than the integrated method. A paired *t*-test shows that this result is highly significant (*p*-value of <0.001).

The average difference between the results produced by both methods increases from 1.4% to 63.4% when the maximum walking distance is increased from 5 to 40. Even though the difference remains significant even for a maximum walking distance of 5, this result shows that the benefits of using an integrated method increase when the flexibility in one of the decisions (in this case, the assignment decision) is larger.

5.5.3. Neighborhood analysis

In this section, 4 extra test are runs over the 112 instances to examine the contribution of each neighborhood to the final solution quality. In each of the test runs, one neighborhood is removed, keeping the order of the remaining neighborhoods the same. In this case, all 25 solutions are considered regardless of whether or not the CPU time is more than 1 hour. The corresponding paired t-tests examine the hypothesis that the solution obtained using the algorithm without the respective neighborhood is larger (worse) than the solution obtained when all neighborhoods are included.

When the *remove*, *replace*, and *remove–insert within a route* neighborhoods are removed, the average solution becomes 19.7%, 5% and 0.6% worse respectively. All these differences are highly statistically significant (with p-values of .0013 or smaller).

6. Conclusions and future research

The school bus routing problem consists of a student to stop assignment problem and a routing problem. A GRASP + VND matheuristic has been proposed that solves both problems in an iterative fashion, thereby improving the solution at each consecutive iteration. To assess the quality of the GRASP + VND matheuristic, an exact MIP model has been developed, as well as a column generation approach which computes tight lower bounds on the optimal solution. Experiments conducted on 112 instances show that the proposed GRASP + VND matheuristic finds most known optimal solutions much faster than the exact procedure and can handle instances that are much larger: the matheuristic can produce very good solutions within 1 hour for realistic instances of 80 stops and 800 students. Future research efforts are now aimed in three directions. First, ways are being investigated 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. It may also be interesting to test the algorithm against various realistic student distribution patterns (rural, urban, etc.). Secondly, 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. Thirdly, even stronger formulations and better exact procedures should lead to more optimal solutions and stronger bounds.

Appendix A. Detailed computational results

See Table 5.

References

Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J., 2007. The Traveling Salesman Problem: A Computational Study. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ.
Baldacci, R., Dell'Amico, M., 2010. Heuristic algorithms for the multi-depot ring-star problem. European Journal of Operational Research 203 (1), 270–281.
Baldacci, R., Dell'amico, M., SalazarGonzález, J.J., 2004. The Capacitated m-Ring Star Problem. Technical Report, DISMI, University of Modena and Reggio Emilia, Italy.
Bennett, B., Gazis, D., 1972. School bus routing by computer. Transportation Research 6, 317–326.
Bodin, L., Berman, L., 1979. Routing and scheduling of school buses by computer. Transportation Science 13, 113–129.
Bowerman, R., Hall, B., Calamai, P., 1995. A multiobjective optimization approach to urban school bus routing: formulation and solution method. Transportation Research Part A: Policy and Practice 29A, 107–123.
Braca, J., Bramel, J., Posner, B., Simchi-Levi, D., 1997. A computerized approach to the New York City school bus routing problem. IIE Transactions 29, 693–702.
Chapleau, L., Ferland, J., Rousseau, J., 1985. Clustering for routing in densely populated areas. European Journal of Operational Research 20, 48–57.
Chvátal, V., 1983. Linear Programming. W.H. Freeman.
Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581.
Desrosiers, J., Ferland, J., Rousseau, J.-M., Lapalme, G., Chapleau, L., 1980. An overview of a school busing system. In: Jaiswal, N. (Ed.), Internal Conference on Transportation. Scientific Management of Transportation, vol. IX. New Delhi, India, pp. 235–243.
Drexl, A., 1991. Scheduling of project networks by job assignment. Management Science 37, 1590–1602.
Dulac, G., Ferland, J., Fogues, P.-A., 1980. School bus routes generator in urban surroundings. Computers and Operations Research 7, 199–213.
Feo, T.A., Resende, M.G.C., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8 (2), 67–71.
Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133.
Ford, L.R., Fulkerson, D.R., 1962. Flows in Networks. Princeton University Press, Princeton, NJ.
Hachicha, M., Hodgson, J.M., Laporte, G., Semet, F., 2000. Heuristics for the multi-vehicle covering tour problem. Computers and Operations Research 27, 29–42.
Hansen, P., Mladenović, N., 1997. Variable neighborhood search for the p-median. Location Science 5, 207–226.
Hansen, P., Mladenović, N., 1999. An introduction to variable neighborhood search. In: Voss, S., Martello, S., Osman, I., Roucairol, C. (Eds.), Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston, pp. 433–458.
Hansen, P., Mladenović, N., 2001a. Industrial applications of the variable neighbourhood search metaheuristic. In: Decisions and Control in Management Science. Kluwer, Boston, pp. 261–274.
Hansen, P., Mladenović, N., 2001b. Variable neighbourhood search: Principles and applications. European Journal of Operational Research 130, 449–467.
Kinable, J., Spieksma, F., van den Berghe, G., submitted for publication. School Bus Routing – A Column Generation Approach.
Langevin, A., Soumis, F., Desrosiers, J., 1990. Classification of travelling salesman problem formulations. Operations Research Letters 9 (2), 127–132.
Laporte, G., Nobert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location–routing problems. Transportation Science 22 (3), 161–172.
Li, LYO, Fu, Z., 2002. The school bus routing problem: a case study. Journal of the Operational Research Society, 552–558.
Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. Journal of the ACM 7 (4), 326–329.
Mladenović, N., 1995. A variable neighborhood algorithm – a new metaheuristic for combinatorial optimization. In: Abstracts of Papers Presented at Optimization Days, p. 112.
Park, J., Kim, B.I., 2010. The school bus routing problem: a review. European Journal of Operational research 202 (2), 311–319.
Park, J., Tae, H., Kim, B.-I., 2012. A post-improvement procedure for the mixed load school bus routing problem. European Journal of Operational Research 217 (1), 204–213.
Riera-Ledesma, J., Salazar-Gonzlez, J.J., 2013. A column generation approach for a school bus routing problem with resource constraints. Computers and Operations Research 40 (2), 566–583.
Schittekat, P., Sevaux, M., Sörensen, K., 2006. A mathematical formulation for a school bus routing problem. In: Proceedings of the IEEE 2006 International Conference on Service Systems and Service Management, Troyes, France.
Schittekat Patrick, 2010. Metaheuristics for Planning in Logistics From Theory to Industrial Applications. PhD Thesis, Universiteit Antwerpen.
Sörensen, K., Sevaux, M., Schittekat, P., 2008. Multiple neighbourhood search in commercial VRP packages: evolving towards self-adaptive methods. Lecture Notes in Economics and Mathematical Systems, vol. 136. Springer, London, pp. 239–253 (Chapter Adaptive, self-adaptive and multi-level metaheuristics).
Thangiah, S.R., Wilson, B., Pitluga, A., Mennell, W., 2005. School Bus Routing in Rural School Districts, Working Paper, Computer Science Department, Slippery Rock University, Slippery Rock, PA, USA.
Toth, P., Vigo, D. (Eds.), 2001. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
Vanderbeck François, 1994. Decomposition and Column Generation for Integer Programs. PhD Thesis, Louvain-La-Neuve.
Wolsey, Laurence A., 1998. Integer Programming. Wiley-Interscience.