

A Heuristic Algorithm for School Bus Routing with Bus Stop Selection

Monique Sciortino, Rhyd Lewis, and Jonathan Thompson

School of Mathematics, Cardiff University, Cardiff, CF24 4AG, Wales
{sciortinom, lewisr9, thompsonjm1}@cardiff.ac.uk

Abstract. In this paper a heuristic algorithm is proposed for a school bus routing problem which is formulated as a capacitated and time-constrained open vehicle routing problem with a homogeneous fleet and single loads. The algorithm determines the selection of bus stops from a set of potential stops, the assignment of students to the selected bus stops, and the routes along the selected bus stops. Its goals are to minimize the number of buses used, the total route journey time and the student walking distances. It also aims at balancing route journey times between buses. The performance of the algorithm is evaluated on a set of twenty real-world problem instances and compared against solutions achieved by a mixed integer programming model. Reported results indicate that the heuristic algorithm finds high-quality solutions in very short amounts of computational time.

Keywords: School Bus Routing · Bus Stop Selection · Local Search · Heuristics · Set Covering · Mixed Integer Programming

1 Introduction

The school bus routing problem (SBRP) is a combinatorial optimization problem which was first investigated over 40 years ago [18]. In various countries, school bus transportation forms part of the government's administrative mechanism and is funded through local taxes. Students who live at least a certain distance from the school they attend are entitled to free or subsidized transport to and from school. In Malta, for example, school transport is provided free of charge to all state school students residing at least 1km from their school. Additional restrictions are also typically imposed on the distance that students are expected to walk between their homes and their designated bus stops.

For the academic year 2019-2020, the Maltese government announced that €27 million was to be allocated to provide free transport for over 26,000 students. Given the large amount of funds being invested, it is crucial that governments make efforts to minimize the total cost required to provide these services. One of the highest priorities is to limit the number of buses used, since each bus has an acquisition cost and a driver employment cost. Moreover, it is critical to minimize operational costs by ensuring that route journey times are kept as short as possible. This also promotes positive well-being of students, particularly younger

ones. In Wales, for example, a maximum 45 minute and 60 minute journey time is recommended for primary and secondary school pupils, respectively.

The SBRP falls into a larger class of problems called vehicle routing problems (VRPs). These involve designing optimal delivery or collection routes from one or more depots to a set of geographically scattered customers, subject to a variety of side constraints [15]. Typical constraints in VRPs include maximum capacity restrictions on vehicles (capacitated VRP (CVRP)) and maximum time/distance restrictions on routes. The CVRP first appeared over six decades ago in the seminal paper by Dantzig and Ramser [9]. Sariklis and Powell [21] proposed the open VRP (OVRP) in which routes do not start and end at a depot (as in the classical VRP), but rather either start or end at a depot. The SBRP can be modelled as a capacitated and time/distance-constrained OVRP (e.g. [4]). A taxonomic review of the VRP and its variants is presented by Braekers et al. [6].

Desrosiers et al. [10] decompose the SBRP into five subproblems. In the first subproblem, *data preparation*, a network containing the schools, student residences, potential bus stop locations and bus depots is generated. Information on the number of students at each residence, the school destination of each student, the number of buses available and their capacities is also specified. The second subproblem, *bus stop selection*, seeks to select a subset of bus stops from a set of potential bus stops and assign students to these stops. *Route generation* deals with designing routes that optimize operational efficiency without sacrificing bus safety and service quality. These objectives are often conflicting in nature since an improvement in the level of service quality can increase the cost of provision. The last two subproblems, *school bell time adjustment* and *route scheduling*, adjust the schools' opening/closing times to allow buses to service multiple schools and establish chains of routes that can be executed by the same vehicle.

In this paper, we focus on the single-school SBRP in which a series of routes is constructed for each school. This is because mixed loads (students from different schools travelling on the same bus simultaneously) are not permitted in the locations considered. The majority of the publications on school bus routing also deal with the single-school SBRP (e.g. [20, 22, 24]). Here, we cover the first three subproblems stated above. Park and Kim [19] and Ellegood et al. [13] note that bus stop selection is often omitted in the literature; however, they classify the solution approaches developed for the SBRP with bus stop selection. Here, we employ the location-allocation-routing (LAR) strategy, in which bus stops are first selected, students are assigned to stops, and then route generation follows. Park and Kim [19] and Ellegood et al. [13] also observe that most studies assume a homogeneous (same capacity) fleet. We take the same assumption here and dedicate our research to the morning problem, whereby students are picked up from stops and dropped off at school. A solution to the afternoon problem, whereby students are picked up from school and dropped off at stops, can be found by reversing the routes.

The remainder of the paper is organized as follows. Section 2 defines our SBRP whereas Section 3 describes our heuristic algorithm developed for this problem. Section 4 presents the set of real-world problem instances considered

as well as the computational results. Finally, Section 5 provides the concluding remarks.

2 Problem Definition

In our SBRP, we define a parameter m_w which indicates the maximum walking distance that a student is expected to walk to get to a bus stop. We also define parameter m_e which specifies the minimum walking distance that students should live from the school to be eligible for school transportation. As in [16], our problem can be represented via two sets of vertices, V_1 and V_2 , and two sets of edges, E_1 and E_2 . The vertex set V_1 consists of one school v_0 and n potential bus stops v_1, v_2, \dots, v_n and the edge set E_1 contains all $n(n+1)$ directed edges (u, v) with $u, v \in V_1$ and $u \neq v$. Each edge (u, v) in the complete directed graph (V_1, E_1) is weighted by the shortest driving time $t(u, v)$ from u to v . Meanwhile, the vertex set V_2 consists of eligible student addresses, with each address $w \in V_2$ being weighted by the number $s(w)$ of students living at address w who require school transportation. E_2 is the set $\{\{w, v\} : w \in V_2 \wedge v \in V_1 \setminus \{v_0\} \wedge d(w, v) \leq m_w\}$, where $d(w, v)$ gives the shortest walking distance from address w to bus stop v .

For this problem we can assume that the undirected bipartite graph $(V_2, V_1 \setminus \{v_0\}, E_2)$ has no isolated vertices. Otherwise, either an address has no bus stop within walking distance m_w (and therefore a new bus stop must be added to V_1), or a bus stop has no address within walking distance m_w and can thus be removed from V_1 . Moreover, a bus stop $v \in V_1 \setminus \{v_0\}$ for which there exists an address $w \in V_2$ with just one incident edge $\{w, v\}$ shall be referred to as a *compulsory stop*. This is because such a stop v is the only stop within walking distance m_w to students living in address w and must therefore be present in a solution.

A feasible solution to our SBRP is given by a set of routes $\mathcal{R} = \{R_1, R_2, \dots\}$, as illustrated in Fig. 1. Each route $R \in \mathcal{R}$ uses one bus of capacity C which visits a subset of bus stops and terminates at the school v_0 . The subset of bus stops traversed by all routes is denoted by $V_1' \subseteq V_1 \setminus \{v_0\}$. This set should *cover* each address $w \in V_2$ at least once, meaning that students in each address w will have at least one bus stop in V_1' within walking distance m_w . Such a covering shall be referred to as a *complete covering* of V_2 whereas a covering which does not satisfy this property shall be referred to as an *incomplete covering* of V_2 . In addition, the total number $s(R)$ of students boarding the bus on route R should not exceed the capacity C , and the journey time $t(R)$ of route R should not exceed the maximum journey time m_t . These constraints can be expressed as follows:

$$\bigcup_{R \in \mathcal{R}} R = V_1' \quad (1)$$

$$\forall w \in V_2, \exists v \in V_1' \mid \{w, v\} \in E_2 \quad (2)$$

$$s(R) \leq C \quad \forall R \in \mathcal{R} \quad (3)$$

$$t(R) \leq m_t \quad \forall R \in \mathcal{R}. \quad (4)$$

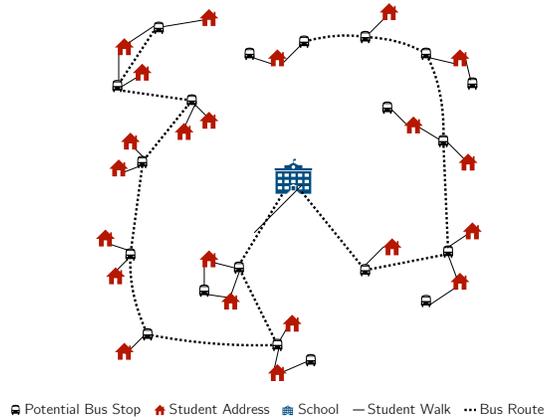


Fig. 1. A feasible solution with $|\mathcal{R}| = 2$, $|V_1| = 21$, $|V'_1| = 14$ and $|V_2| = 22$.

It is important to note that bus stops in V'_1 are not restricted to feature in exactly one route in \mathcal{R} . For example, there may not be enough spare capacity in a bus to serve all students waiting at a bus stop $v \in V'_1$. In that case, bus stop v must be visited by more than one bus and we call such a stop a *multistop*. In VRP literature, this characteristic is referred to as the allowance of split deliveries [11, 12]. Each student boarding at a multistop is only permitted to board one specific route serving that stop since, otherwise, a bus stopping at that multistop may possibly be too full to serve a subsequent stop in its path.

As in [16], the calculation of the journey time $t(R)$ of route $R \in \mathcal{R}$ is composed of two components; the total *bus travel time* and the total *bus dwell time*. Each dwell time within a route captures the time spent servicing a designated bus stop, i.e. the time spent to halt the bus, open the doors, board the students and merge back into traffic. In our case, we estimate the dwell time at stop v in route R by $d(v, R) = d_1 + d_2 s(v, R)$, where $s(v, R)$ represents the number of boarding students at stop v onto route R , d_2 represents the boarding time per student, and d_1 is a parameter which accounts for the remaining servicing time. Here, d_1 and d_2 are taken to be 15 and 5 seconds, respectively. Therefore, given a route $R = (v_1, v_2, \dots, v_l, v_0)$, the route journey time $t(R)$ is given by

$$t(R) = \left(\sum_{i=1}^{l-1} t(v_i, v_{i+1}) + t(v_l, v_0) \right) + \left(\sum_{i=1}^l (d_1 + d_2 s(v_i, R)) \right), \quad (5)$$

where the first component gives the total bus travel time and the second component gives the total bus dwell time.

As previously mentioned, the primary objective of our SBRP is to identify an appropriate subset of bus stops V'_1 in order to minimize the number $|\mathcal{R}|$ of routes (buses) included in a solution. In our case, this is achieved by attempting to produce feasible solutions that use the lower bound of $\lceil \sum_{w \in V_2} s(w)/C \rceil$ routes needed to serve all students. A solution satisfying constraints (1)-(3) and

meeting this lower bound of $|\mathcal{R}|$ is always guaranteed since multistops are allowed; however, any one of the routes could potentially violate the maximum riding time constraint (4). Thus, there may be cases where additional routes are needed. In the generation of feasible solutions, two secondary objectives and a tertiary objective are also considered. The first secondary objective deals with the effectiveness of the school transportation service, whereby we seek to minimize the total student walking distance. Through the other secondary objective, we aim to target efficiency of the service by minimizing the total route journey time in a solution. The tertiary objective is employed whenever multiple feasible solutions have the same minimum total route journey time. Here, we encourage equity of service by minimizing the discrepancy between the longest and shortest routes.

3 Algorithm Description

Our heuristic algorithm uses the following overall strategy. Initially, a subset of bus stops is selected and a nearest neighbour heuristic is employed to construct an initial solution using a fixed number $|\mathcal{R}|$ of routes. As mentioned, $|\mathcal{R}|$ is initially taken to be the lower bound stated in the previous section. Note also that the initial assignment of stops to routes allows the violation of (4). A local search routine involving six improvement heuristics is then invoked on this solution to try and shorten the routes using the current subset of bus stops. After this routine has completed, a procedure is performed whereby the current subset of selected bus stops is altered, the current solution is repaired, and the local search routine is re-applied. This is repeated until a time limit is reached, leading to an approach similar to iterated local search. If no solution satisfying (1)-(4) is achieved at this limit, the number $|\mathcal{R}|$ of routes is increased by one and the algorithm is restarted.

3.1 Construction of Initial Solution

In our approach, the initial subset of bus stops V'_1 is selected as follows. First, all compulsory stops are included in V'_1 . The non-compulsory stops are arranged in non-increasing order according to the number of currently uncovered addresses they serve. The stop with the largest such value is then added to V'_1 , breaking ties randomly. This ordering and selection procedure is repeated until a complete covering of V_2 is obtained. Each address in V_2 is then assigned to the closest bus stop in V'_1 . The assignment of addresses to stops determines the number $s_{V'_1}(v)$ of boarding students at each stop $v \in V'_1$. It may also be the case that some stops have no boarding students, in which case they are removed from V'_1 . Next, each bus stop in V'_1 is assigned to one of the $|\mathcal{R}|$ routes such that each bus is not overloaded. This assignment follows a parallel backward implementation of the nearest neighbour constructive heuristic. To start, $|\mathcal{R}|$ empty routes are defined and the remaining capacity c_i of each route $R_i \in \mathcal{R}$ is set to C . The $|\mathcal{R}|$ closest stops to the school are then added at the front of the routes, one in each route.

Closeness to school is measured by the dwell time at the stop plus the shortest driving time from the stop to the school. In order to calculate the dwell time at stop $v \in V'_1$ in route R_i , the minimum of c_i and $s_{V'_1}(v)$ is considered as there may be more than c_i students boarding at stop v . In this case, a multistop is created since the remaining $s_{V'_1}(v) - c_i$ students boarding at stop v will need to be assigned to a different route R_j . The remaining capacities c_i are then updated accordingly. This iterative procedure of determining the closest stop to the most recently added stop in route R_i , adding it to the front of the route and updating the remaining capacity c_i is repeated until all stops in V'_1 are assigned to a route. On completion, an initial solution \mathcal{R} will have been generated and can be evaluated according to the cost function described presently.

3.2 Cost Function

A solution $\mathcal{R} = \{R_1, R_2, \dots\}$ is evaluated according to the cost function

$$f(\mathcal{R}) = \sum_{R \in \mathcal{R}} t'(R), \quad (6)$$

where

$$t'(R) = \begin{cases} t(R) & \text{if } t(R) \leq m_t, \\ m_t + m_t(1 + t(R) - m_t) & \text{otherwise.} \end{cases} \quad (7)$$

This means that if the journey time of a route $R \in \mathcal{R}$ exceeds m_t , then this journey time is scaled up heavily via a penalty. Otherwise, it is unaltered. The addition of the value 1 in the second case of (7) guarantees that two routes both with journey time at most m_t are always preferred over one route with journey time exceeding m_t .

3.3 Local Search Routine

As mentioned, the intention of our local search routine is to shorten the journey times of routes in \mathcal{R} while maintaining the satisfaction of (1)-(3). Our routine uses a combination of three intra-route and three inter-route operators, with the former being applied to a single route $R_1 \in \mathcal{R}$ and the latter being applied to a pair of routes $R_1, R_2 \in \mathcal{R}$. Without loss of generality, assume that $R_1 = (v_1, v_2, \dots, v_{l_1}, v_0)$ and $R_2 = (u_1, u_2, \dots, u_{l_2}, v_0)$. Note that this local search acts on a solution using a fixed subset of bus stops V'_1 . The six operators considered are the following:

- *Exchange*: Choose two stops v_i, v_j in R_1 , where $1 \leq i < j \leq l_1$, and swap their position.
- *Two-Opt*: Choose two stops v_i, v_j in R_1 , where $1 \leq i < i + 3 \leq j \leq l_1$, and invert sub-route v_i, \dots, v_j .
- *Generalized Or-Opt*: Choose stops v_i, v_j, v_k in R_1 , where $1 \leq i \leq j \leq l_1$ and $(1 \leq k < i$ or $j + 1 < k \leq l_1 + 1)$. Remove sub-route v_i, \dots, v_j and transfer it before stop v_k , possibly also inverting the sub-route if this yields a better cost. If $k = l_1 + 1$, then the sub-route is transferred before school v_0 .

- *Or-Exchange*: Choose stops v_i, v_j in R_1 , where $1 \leq i \leq j \leq l_1$, and stop u_k in R_2 , where $1 \leq k \leq l_2 + 1$. Remove sub-route v_i, \dots, v_j from R_1 and transfer it before stop u_k in R_2 , possibly also inverting the sub-route if this yields a better cost. If $k = l_2 + 1$, then the sub-route is transferred before school v_0 .
- *Cross-Exchange*: Choose stops v_{i_1}, v_{j_1} in R_1 , where $1 \leq i_1 \leq j_1 \leq l_1$, and stops u_{i_2}, u_{j_2} in R_2 , where $1 \leq i_2 \leq j_2 \leq l_2$. Swap sub-routes v_{i_1}, \dots, v_{j_1} and u_{i_2}, \dots, u_{j_2} , possibly inverting either sub-route if this yields a better cost.
- *Creating Multistops*: If routes R_1, R_2 satisfy $t(R_1) > m_t$ and $s(R_2) < C$, then choose stop v_i in R_1 , where $1 \leq i \leq l_1$, for which $s(v_i, R_1) \geq 2$. If v_i is not already in R_2 , then insert a copy of v_i in R_2 before the stop u_k , where $1 \leq k \leq l_2$, (or school v_0) which causes the smallest increase in $t(R_2)$. Next transfer $z = \min\{s(v_i, R_1) - 1, C - s(R_2)\}$ students from the occurrence of stop v_i in R_1 to the occurrence of stop v_i in R_2 . Here, the value z gives the maximum number of students who can be transferred (hence, decreasing $t(R_1)$ as much as possible) such that both occurrences of v_i have at least one boarding student and both routes R_1 and R_2 satisfy (3).

The neighbourhood sizes corresponding to the above operators are $\mathcal{O}(|V'_1|^2)$, $\mathcal{O}(|V'_1|^2)$, $\mathcal{O}(|V'_1|^3)$, $\mathcal{O}(|V'_1|^3)$, $\mathcal{O}(|V'_1|^4)$, and $\mathcal{O}(|V'_1|^3)$, respectively. These operators are the same as those used in [16]. The exchange, two-opt and cross-exchange operators are also used in a similar context in [8], while the generalized Or-opt and Or-exchange are extensions (case $i \neq j$) of operators used in [8] and [22]. Note that some Or-exchange and cross-exchange moves can lead to a violation of (3). Such moves are therefore not evaluated. Moreover, these two operators can result in duplicate stops in the same route, which are removed as follows. Without loss of generality, assume that sub-route v_i, \dots, v_j is being transferred from route R_1 to route R_2 and that one stop $v_h, i \leq h \leq j$, is already present in R_2 . Then stop v_h is removed from the sub-route and the students boarding this occurrence of v_h are all transferred to the occurrence of v_h in R_2 .

Our local search routine follows the direction of steepest descent. In each iteration, all moves in the union of the six neighbourhoods are evaluated and the move which gives the largest reduction in cost is performed. If multiple moves give the largest reduction in cost, the one which yields the smallest discrepancy between the longest and shortest routes in the solution is performed. Such a breakage of ties aims at balancing the journey times between buses. The local search routine terminates when a solution whose neighbourhoods contain no improving moves is reached.

3.4 Generation of Alternative Solutions

Recall that the subset of bus stops V'_1 is fixed during our local search routine. For this reason, our algorithm also contains an operator that generates a new subset of bus stops V''_1 , assigns students to these bus stops, and then creates a set of routes that use these stops. We designed four variants of the algorithm, which differ in the way they generate V''_1 . These are:

- (I) Generating V_1'' from scratch;
- (II) Generating V_1'' from the subset V_1' used in the previous iteration;
- (III) Generating V_1'' from the most recent subset V_1' that yields a feasible solution with the lowest cost found so far;
- (IV) Generating V_1'' via a trade-off between Variants II and III, whereby V_1'' has 50% chance of being generated according to Variant II and 50% chance of being generated according to Variant III.

Note that in Variant III, the subset of stops generated in the previous iteration is used if no subset has yielded a feasible solution so far.

In Variant I, the generation of V_1'' follows the same selection strategy as that discussed in Section 3.1 and new routes are again produced via nearest neighbour construction. In Variants II to IV, the non-compulsory stops in V_1' are identified and a random selection of these is removed. Assuming a total number α of non-compulsory stops, in our case the number of removals is selected according to a Binomial distribution with parameters α and $3/\alpha$ so that three stops are removed on average. Upon removal, if we have an incomplete covering of V_2 , then additional stops must be added to V_1' . If all addresses not covered by the stops in V_1' are covered by stops which were not originally in V_1' , then, at each stage, a stop from the latter set of stops which serves the largest number of uncovered addresses is added, breaking ties randomly. If, on the other hand, some address is also uncovered by the stops which were not originally in V_1' , then at least one of the removed stops must be added back. The same selection strategy is applied in this case and the whole procedure is repeated until a new complete covering V_1'' of V_2 is achieved. Each address is then reassigned to the closest stop in V_1'' and stops with no addresses assigned to them are removed from V_1'' .

Having determined a new subset of bus stops, repairs are then made to \mathcal{R} so that only bus stops in V_1'' feature in the solution. To do this, all occurrences of stops in $V_1' \setminus V_1''$ are first removed from \mathcal{R} . For stops $v \in V_1'' \cap V_1'$ for which $s_{V_1''}(v) < s_{V_1'}(v)$, $s_{V_1'}(v) - s_{V_1''}(v)$ students are removed from occurrences of v in \mathcal{R} . If this results in an occurrence of v with no boarding students, then this occurrence is removed from \mathcal{R} . For stops $v \in V_1'' \cap V_1'$ for which $s_{V_1''}(v) > s_{V_1'}(v)$, an attempt is made to add students to occurrences of v in \mathcal{R} . If not all $s_{V_1''}(v) - s_{V_1'}(v)$ students can be added, then a new occurrence of v must be added to \mathcal{R} . Stops $v \in V_1'' \setminus V_1'$ must also be added to the solution. A new stop is inserted in a route having the lowest load, at the position which causes the least increase in the route journey time. If this insertion does not cater for all students boarding that stop, then the procedure is repeated.

Having repaired solution \mathcal{R} (or generated completely new routes in the case of Variant I), the local search routine is then re-invoked. This repair-and-improve process is repeated until the time limit is reached.

4 Computational Experiments

A total of twenty real-world problem instances are considered here, summarized in Table 1. The problem instances pertaining to the UK and Australia originate

Table 1. Summary statistics for the twenty real-world problem instances, listed in increasing order of $|V_1|$. The number S represents the total number of students, calculated as $\sum_{w \in V_2} s(w)$. Distances m_e and m_w are given in km.

Location	Country/State	$ V_1 $	$ V_2 $	S	m_e	m_w	C
Mġarr	Malta	60	110	190	1.0	1.0	40
Mellieħa	Malta	83	98	171	1.0	1.0	40
Porthcawl	Wales	153	42	66	3.2	1.6	70
Qrendi	Malta	161	150	255	1.0	1.0	40
Suffolk	England	174	123	209	4.8	1.6	70
Senglea	Malta	186	158	266	1.0	1.0	40
Victoria	Gozo	292	99	171	1.0	1.0	40
Handaq	Malta	298	170	285	1.0	1.0	40
Pembroke	Malta	329	200	335	1.0	1.0	40
Canberra	ACT	331	296	499	4.8	1.0	70
Valletta	Malta	419	159	268	1.0	1.0	40
Birkirkara	Malta	422	181	306	1.0	1.0	40
Hamrun	Malta	519	321	192	1.0	1.0	40
Cardiff	Wales	552	90	156	4.8	1.6	70
Milton Keynes	England	579	149	274	4.8	1.6	70
Bridgend	Wales	633	221	381	4.82	1.6	70
Edinburgh-2	Scotland	917	190	320	1.6	1.6	70
Edinburgh-1	Scotland	959	409	680	1.6	1.6	70
Adelaide	South Australia	1118	342	565	1.6	1.6	70
Brisbane	Queensland	1817	438	757	3.2	1.6	70

from [16] and can be downloaded at [1]. The remainder were generated by us using the Bing Maps API and can be downloaded at [2]. Each problem instance was generated as follows. The location of the school was first identified and a number of random student addresses were selected on/outside a circle of radius m_e from the school. The number of students living at each address was generated randomly according to the following distribution: 1, 2, 3 and 4 with probabilities 0.45, 0.4, 0.14 and 0.01, respectively. As mentioned in [16], this distribution approximates the relevant statistics in the locations considered. Potential bus stops were then identified through public records such that each stop has at least one address within walking distance m_w and each address has at least one stop within walking distance m_w . Shortest driving times between each bus stop pair and shortest walking distances between each bus stop and address pair were then determined. Here, we use $C \in \{40, 70\}$, depending on the location of the problem instance under study, and $m_t = 2700$ seconds (45 minutes).

Our heuristic algorithm was coded in C++ and run on a 3.6GHz 8-Core Intel Core i9 processor with 8GB RAM. Variants I to IV were each run 25 times on each instance. The time limit for each run was taken to be five minutes. Overall, we found that feasible solutions using the lower bound of $\lceil \sum_{w \in V_2} s(w)/C \rceil$ routes were achieved in nineteen of the twenty instances in all runs. The only instance

which required one additional route was the rural-based Suffolk instance. This was also observed in [16].

Statistics on the results achieved by our algorithm are summarized in Table 2. Columns 3 to 6 display the average number of iterations performed in each algorithm variant. For each variant, this number was compared with the average number of iterations that resulted in solutions satisfying (1)-(4). For Bridgend, approximately 74%, 61%, 26% and 32% of the iterations performed in Variants I to IV, respectively, produced infeasible solutions, on average. For Brisbane, the four average proportions of infeasible solutions were all less than 0.2%. Some infeasible solutions were also produced in Variants II and IV for Porthcawl and Variants I, II and IV for Suffolk. The average proportions were both less than 0.04% for Porthcawl and all less than 0.7% for Suffolk. All remaining instances saw (1)-(4) satisfied in all runs.

According to Table 2, Variant II performed the highest average number of iterations for all instances except Edinburgh-2. On the contrary, Variant I performed the lowest average number of iterations for all instances except Porthcawl and Edinburgh-2. This was expected given that this variant does not use information from previous iterations when altering the current subset of bus stops. Consequently, applications of local search take longer in each iteration. It is also evident that Variant IV performed a higher average number of iterations than Variant III for all instances except Edinburgh-2.

Table 2. Number of iterations performed by our algorithm. All figures are averaged across the 25 runs, rounded to the nearest integer, plus/minus the standard deviation.

Location	$ \mathcal{R} $	Variant I	Variant II	Variant III	Variant IV
Mgarr	5	12777 \pm 112	32131 \pm 367	25137 \pm 360	27482 \pm 248
Mellieħa	5	1323 \pm 9	6951 \pm 53	5359 \pm 107	5848 \pm 74
Porthcawl	1	91916 \pm 1091	96843 \pm 1210	85351 \pm 2465	92171 \pm 3915
Qrendi	7	9602 \pm 68	28176 \pm 230	22880 \pm 656	23901 \pm 621
Suffolk	4	3384 \pm 37	8076 \pm 95	6436 \pm 193	6660 \pm 191
Senglea	7	16026 \pm 105	45776 \pm 317	42425 \pm 1177	43792 \pm 896
Victoria	5	2124 \pm 31	6956 \pm 90	5135 \pm 206	5367 \pm 254
Handaq	8	2439 \pm 30	8438 \pm 120	7433 \pm 495	7731 \pm 362
Pembroke	9	4208 \pm 29	16109 \pm 160	14170 \pm 621	14702 \pm 468
Canberra	8	2431 \pm 42	10073 \pm 167	8038 \pm 290	8456 \pm 247
Valletta	7	6639 \pm 6	21976 \pm 78	20915 \pm 577	21405 \pm 403
Birkirkara	8	4691 \pm 70	17519 \pm 276	16039 \pm 635	16466 \pm 491
Hamrun	9	3233 \pm 54	13681 \pm 242	13210 \pm 258	13404 \pm 259
Cardiff	3	16241 \pm 102	25386 \pm 181	25330 \pm 185	25345 \pm 131
Milton Keynes	4	8601 \pm 7	20039 \pm 40	18699 \pm 637	19196 \pm 272
Bridgend	6	3932 \pm 57	11066 \pm 106	9944 \pm 119	10277 \pm 85
Edinburgh-2	5	5681 \pm 66	5692 \pm 55	5709 \pm 52	5680 \pm 60
Edinburgh-1	10	1792 \pm 20	6489 \pm 83	6365 \pm 141	6417 \pm 128
Adelaide	9	1745 \pm 24	5943 \pm 72	5530 \pm 173	5668 \pm 134
Brisbane	11	742 \pm 18	3304 \pm 71	3241 \pm 70	3262 \pm 70

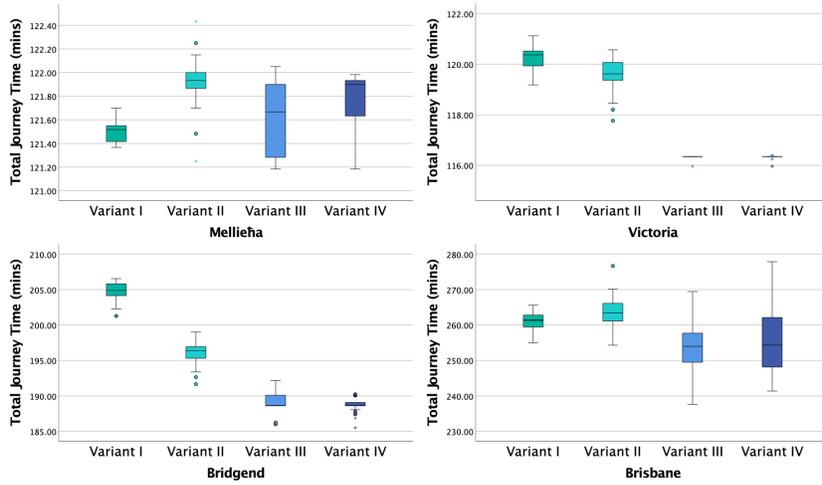


Fig. 2. Boxplots displaying the performance of the variants on the Mellieha, Victoria, Bridgend and Brisbane instances.

For each instance, a Kruskal-Wallis test on the 100 total journey times (in minutes) reached by our heuristic algorithm revealed statistically significant differences between the algorithm variants ($p < 0.001$). Post-hoc Bonferroni-adjusted pairwise comparison tests indicated that, for 16 instances, the total journey times of Variants I and II are significantly different at the 0.05 level than those of Variants III and IV. Half of these instances also saw a significant difference between Variants I and II. Two instances (Porthcawl and Edinburgh-2) saw significant differences between Variant III and all other variants, whereas the Mğarr instance saw significant differences between Variant I and all other variants. For the Mellieha instance, significant differences were found between Variants I and II, I and IV, and II and III. Boxplots displaying the performance of the variants on four selected instances are displayed in Fig. 2.

Moving to Table 3, Columns 4 to 7 display the best total journey times for the different algorithm variants. Each instance’s best reported result across all variants is displayed in bold and the number of runs giving that result is shown in brackets. According to Table 3, Variants I to IV produced best reported results in 2, 3, 16 and 14 instances, respectively. Moreover, Variants I to IV produced best total journey times that are at most 14.37%, 8.76%, 4.28% and 2.14% (respectively) worse than the best reported results. It is also evident that our best reported result for eight instances was achieved by only one run. For the other twelve instances, multiple runs reached the best reported result. Some or all multiple runs for all these instances except Porthcawl and Edinburgh-2 have different corresponding subsets of bus stops. The total number of alternative subsets of bus stops is given in Column 3 of Table 3.

The best reported results from our heuristic algorithm are also compared with those of Lewis and Smith-Miles [16], in Column 2. The algorithm used in [16] is

Table 3. Best (across the 25 runs) total route journey times (in minutes) achieved by our heuristic algorithm. Column 2 gives the best reported results in [16] for the instances pertaining to the UK and Australia. Column 8 gives the total route journey time of the best incumbent solution achieved by Gurubi. Column 9 displays the percentage improvement between the best reported result from our heuristic algorithm and the result in Column 8. The last column gives the relative gap between the result in Column 8 and the best known lower bound on the optimal total route journey time.

Location	Best [16]	Subsets	Heuristic Algorithm				MIP (Gurubi)		
			Variant I	Variant II	Variant III	Variant IV	Incumbent	Improvement	MIP Gap
Mgarr	-	3	90.70	87.77 (2)	87.77 (2)	87.77 (3)	87.42	0.40%	8.35%
Mellicha	-	2	121.37	121.25	121.18 (2)	121.18 (1)	120.95	0.19%	11.65%
Porthcawl	26.87	1	28.02	27.35	28.02	26.87 (2)	26.87	-	44.35%
Qrendi	-	7	97.67	96.63	95.80 (6)	95.80 (5)	95.80	-	40.52%
Suffolk	113.43	12	118.77	116.08	112.75 (20)	112.75 (21)	112.75	-	55.54%
Senglea	-	4	84.85	84.57	83.18 (7)	83.18 (7)	81.87	1.58%	63.36%
Victoria	-	4	119.18	117.77	115.97 (2)	115.97 (4)	115.97	-	38.66%
Handaq	-	12	142.47	140.28	138.18 (11)	138.18 (7)	-	-	-
Pembroke	-	1	132.82	133.37	128.80 (1)	129.48	128.80	-	52.89%
Canberra	190.60	1	198.27	193.13	188.25 (1)	188.37	188.25	-	57.00%
Valletta	-	1	125.28	122.50	117.82	117.28 (1)	117.28	-	66.12%
Birkirkara	-	4	116.27	113.87	108.25 (1)	108.25 (3)	108.25	-	55.40%
Hamrun	-	2	118.68	108.75	104.27 (1)	104.27 (1)	104.27	-	62.12%
Cardiff	57.52	3	57.52 (13)	57.52 (16)	57.52 (25)	57.52 (25)	57.52	-	69.17%
Milton Keynes	63.48	1	68.57	65.45	61.63 (1)	62.95	61.10	0.87%	54.58%
Bridgend	185.52	1	201.28	191.67	186.00	185.48 (1)	-	-	-
Edinburgh-2	56.08	1	55.87 (25)	55.87 (25)	55.90	55.87 (25)	-	-	-
Edinburgh-1	151.40	1	174.50	165.95	152.58 (1)	153.60	-	-	-
Adelaide	136.80	1	147.83	143.23	135.47 (1)	136.22	-	-	-
Brisbane	235.15	1	254.97	254.30	237.63 (1)	241.38	-	-	-

similar to Variant II of our algorithm, but makes use of a first-fit-decreasing bin-packing heuristic for the assignment of bus stops to routes rather than a nearest neighbour heuristic. Our best reported results for two instances (Porthcawl and Cardiff) match those in [16]. Additionally, our best reported results for six of the remaining eight instances are better than those in [16].

An attempt was also made to improve each instance’s best reported result from our heuristic algorithm. For this purpose, a mixed integer programming (MIP) model was formulated as shown in the appendix. This model was executed using Gurobi 9.0 with a run time limit of one hour per bus stop subset. Each run was also seeded with the best feasible solution found by our heuristic algorithm for that bus stop subset. The MIP results are presented in Columns 8 to 10 of Table 3. Column 8 gives the total route journey time of the best incumbent solution reached. Column 9 displays the percentage improvement between the best reported result from our heuristic algorithm and the best incumbent result. Finally, Column 10 gives the relative MIP optimality gap between the best incumbent result and the best known lower bound on the optimal total route journey time. Note that Gurobi was able to find a solution having a better total route journey time for four instances and the percentage improvements of these range between 0.19% and 1.58%. On the other hand, Gurobi was not able to improve the heuristic algorithm solution of ten instances within the time limit. For the remaining six instances, the solver did not provide any results before the time limit was reached. The optimality gaps for the achieved MIP results range between 8.35% and 69.17%.

5 Conclusions and Future Developments

In this paper a real-world SBRP has been studied, which incorporates several features found in the literature such as bus capacities, student eligibility, maximum student riding time, maximum student walking distance, multistops (multiple buses visiting a single bus stop) and bus dwell times. A heuristic algorithm has been developed which encompasses the first three subproblems of the SBRP, as defined in [10].

Experiments conducted on twenty problem instances from Malta, the UK and Australia demonstrate the success of the heuristic algorithm on a variety of real-sized instances. For all instances, our algorithm was able to find high-quality solutions in a very short computational time. It also copes with large-scale instances of more than 1800 potential bus stops and 750 students. Through different variants, our algorithm has provided multiple subsets of bus stops yielding the best reported total journey time for ten instances. This extension to what has been done in [16] is beneficial since government administrators can liaise with bus operators to identify the most appropriate subset of bus stops based on factors such as bus depot locations and bus stop accessibility.

The performance of our heuristic algorithm was also compared against solutions achieved through a branch-and-cut method. Only four of the twenty best reported results from the heuristic algorithm turned out to be slightly worse (by

less than two minutes) than the results achieved by MIP. The MIP optimality gaps turned out to be higher than expected. These may be improved by allowing a longer time limit.

The proposed heuristic algorithm as well as the MIP model formulation presented in this paper can be extended to other SBRP variants. One such variant is the heterogeneous fleet, in which buses are characterized by different capacities, as studied in [17, 20, 24] amongst others. Another potential future development is the consideration of multi-tripping where several routes, possibly pertaining to different schools, are merged so that buses are able to perform multiple routes successively. Recent work in this area is discussed in [5, 23]. It is also suggested that future work should address uncertainty in the bus travel times as this will make the heuristic algorithm more applicable in real-world settings (e.g. [3, 7, 25]).

Acknowledgement The research work disclosed in this publication is supported by the Tertiary Education Scholarships Scheme (TESS, Malta).

References

1. <http://rhydlewislew.eu/resources/busprobs.zip>.
2. <https://github.com/MoniqueSciortino/sbrpMaltaInstances>.
3. Babaei, M., Rajabi-Bahaabadi, M.: School bus routing and scheduling with stochastic time-dependent travel times considering on-time arrival reliability. *Computers and Industrial Engineering*, **138**, 106125 (2019).
4. Bektaş, T., Elmastaş, S.: Solving school bus routing problems through integer programming. *Journal of the Operational Research Society*, **58**(12), 1599–1604 (2007).
5. Bertsimas, D., Delarue, A., Martin, S.: Optimizing schools’ start time and bus routes. *Proceedings of the National Academy of Sciences*, **116**(13), 5943–5948 (2019).
6. Braekers, K., Ramaekers, K., Nieuwenhuysse, I.V.: The vehicle routing problem: state of the art classification and review. *Computers and Industrial Engineering*, **99**, 300–313 (2016).
7. Caceres, H., Batta, R., He, Q.: School bus routing with stochastic demand and duration constraints. *Transportation Science*, **51**(4), 1349–1364 (2017).
8. Chen, X., Kong, Y., Dang, L., Hou, Y., Ye, X.: Exact and metaheuristic approaches for a bi-objective school bus scheduling Problem. *PLoS ONE*, **11**(4), e0153614 (2015).
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management Science*, **6**(1), 80–91 (1959).
10. Desrosiers, J., Ferland, J.A., Rousseau, J.M., Lapalme, G., Chapleau, L.: An overview of a school busing system. *Scientific Management of Transport Systems*, 235–243 (1981).
11. Dror, M., Trudeau, P.: Savings by split delivery routing. *Transportation Science*, **23**(2), 141–145 (1989).
12. Dror, M., Trudeau, P.: Split delivery routing. *Naval Research Logistics*, **37**(3), 383–402 (1990).
13. Ellegood, W.A., Solomon, S., North, J., Campbell, J.F.: School bus routing problem: contemporary trends and research directions. *Omega*, **95**, 1–18 (2020).

14. Kek, A.G.H., Cheu, R.L., Meng, Q.: Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling*, **47**(1-2), 140–152 (2008).
15. Laporte, G., Nobert, Y., Taillefer, S.: Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, **22**(3), 161–172 (1988).
16. Lewis, R., Smith-Miles, K.: A heuristic algorithm for finding cost-effective solutions to real-world school bus routing problems. *Journal of Discrete Algorithms*, **52-53**, 2–17 (2018).
17. Lima, F.M., Pereira, D.S., Conceição, S.V., Ramos Nunes, N.T.: A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context. *Expert Systems With Applications*, **56**, 320–334 (2016).
18. Newton, R.M., Thomas, W.H.: Design of school bus routes by computer. *Socio Economic Planning Sciences*, 75–85 (1969).
19. Park, J., Kim, B.I.: The school bus routing problem: A review. *European Journal of Operational Research*, **202**, 311–319 (2010).
20. Sales, L.D., Melo, C.S., Bonates, T.D., Prata, B.D.: Memetic algorithm for the heterogeneous fleet school bus routing problem. *Journal of Urban Planning and Development*, **144**(2), 04018018 (2018).
21. Sariklis, D., Powell, S.: A heuristic method for the open vehicle routing problem. *The Journal of the Operational Research Society*, **51**(5), 564–573 (2000).
22. Schittkat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., Springael, J.: A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, **229**(2), 518–528 (2013).
23. Shafahi, A., Wang, Z., Haghani, A.: Solving the school bus routing problem by maximizing trip compatibility. *Transportation Research Record: Journal of the Transportation Research Board*, **2667**(1), 17–27 (2017).
24. Siqueira, V.S., Silva, E.N., Silva, R.V., Rocha, M.L.: Implementation of the metaheuristic GRASP applied to the school bus routing problem. *International Journal of E-Education, E-Business, E-Management and E-Learning*, **6**(2), 137–145 (2016).
25. Sun, S., Duan, Z., Xu, Q.: School bus routing problem in the stochastic and time-dependent transportation network. *PLoS ONE*, **13**(8), e0202618 (2018).

Appendix

The MIP model presented here produces solutions consisting of cycles that start and end at the school. The arc from the school to the first bus stop in each route is then excluded. This is possible by assuming that the driving time from the school to any stop is zero.

The decision variables of our model are as follows. Binary variable x_{uvR} indicates whether route $R \in \mathcal{R}$ travels from $u \in V_1$ to $v \in V_1 \setminus \{u\}$. Binary variable y_{vR} indicates whether route $R \in \mathcal{R}$ visits $v \in V_1$. Also, binary variable z_{wv} indicates whether students in address $w \in V_2$ walk to stop $v \in V_1 \setminus \{v_0\}$. Variable $s_{vR} \in \{0, 1, \dots, C\}$ gives the number of students boarding route $R \in \mathcal{R}$ from stop $v \in V_1 \setminus \{v_0\}$. Moreover, variable $l_{vR} \in \{0, 1, \dots, C\}$ gives the total load of route $R \in \mathcal{R}$ just after visiting stop $v \in V_1 \setminus \{v_0\}$. Finally, variable $t_R \in [0, m_t]$ specifies the total journey time of route $R \in \mathcal{R}$. The MIP formulation

is as follows:

$$\min \sum_{R \in \mathcal{R}} t_R \quad (8)$$

$$\text{s.t. } \sum_{u \in V_1} x_{uvR} = y_{vR} \quad \forall v \in V_1, R \in \mathcal{R} \quad (9)$$

$$\sum_{u \in V_1} x_{vuR} = y_{vR} \quad \forall v \in V_1, R \in \mathcal{R} \quad (10)$$

$$y_{v_0R} \geq y_{vR} \quad \forall v \in V_1 \setminus \{v_0\}, R \in \mathcal{R} \quad (11)$$

$$\sum_{\substack{v \in V_1 \setminus \{v_0\} \\ d(w,v) \leq m_w}} z_{wv} = 1 \quad \forall w \in V_2 \quad (12)$$

$$\sum_{R \in \mathcal{R}} y_{vR} \geq z_{wv} \quad \forall v \in V_1 \setminus \{v_0\}, w \in V_2 \quad (13)$$

$$\sum_{w \in V_2} s(w)z_{wv} - \sum_{R \in \mathcal{R}} s_{vR} = 0 \quad \forall v \in V_1 \setminus \{v_0\} \quad (14)$$

$$y_{vR} \leq s_{vR} \quad \forall v \in V_1 \setminus \{v_0\}, R \in \mathcal{R} \quad (15)$$

$$C y_{vR} \geq s_{vR} \quad \forall v \in V_1 \setminus \{v_0\}, R \in \mathcal{R} \quad (16)$$

$$l_{uR} + s_{vR} - C(1 - x_{uvR}) \leq l_{vR} \quad \forall u, v \in V_1 \mid v \neq v_0, R \in \mathcal{R} \quad (17)$$

$$l_{uR} + s_{vR} + C(1 - x_{uvR}) \geq l_{vR} \quad \forall u, v \in V_1 \mid v \neq v_0, R \in \mathcal{R} \quad (18)$$

$$\sum_{u, v \in V_1} t(u, v)x_{uvR} + \sum_{v \in V_1 \setminus \{v_0\}} (d_1 y_{vR} + d_2 s_{vR}) = t_R \quad \forall R \in \mathcal{R}. \quad (19)$$

Objective function (8) minimizes the total journey time of all routes. Constraints (9)-(11) relate to stop and school visits. Constraints (9)-(10) guarantee that if route $R \in \mathcal{R}$ visits $v \in V_1$, then route R should enter and leave v exactly once. Next, Constraints (11) force each route $R \in \mathcal{R}$ to visit school v_0 whenever it visits at least one stop $v \in V_1 \setminus \{v_0\}$. Constraints (12)-(14) relate to student walks and pickups. Constraints (12) ensure that students living in each address $w \in V_2$ walk to exactly one stop within walking distance m_w . Constraints (13) assure that no student walks to an unvisited stop, while Constraints (14) guarantee that the total number of students boarding from stop $v \in V_1 \setminus \{v_0\}$ is equal to the total number of students walking to that stop. Constraints (15)-(16) relate to student boardings. These constraints force the number of students boarding route $R \in \mathcal{R}$ from stop $v \in V_1 \setminus \{v_0\}$ to be 0 if route R does not visit stop v . If route R visits stop v , then (15) also updates the lower bound on the number of boarding students to 1. In addition, Constraints (17)-(18) relate to route loads and also serve as subtour elimination constraints as proposed in [14]. Note that $l_{v_0R} = 0 \forall R \in \mathcal{R}$. These constraints guarantee that no route contains a subtour disconnected from school v_0 and that each route load increases in accordance to the number of students boarding the bus on that route. In fact, if route $R \in \mathcal{R}$ goes from $u \in V_1$ to stop $v \in V_1 \setminus \{u, v_0\}$, then the load of route R just after visiting stop v is set equal to the sum of the load of route R just after visiting u and the number of students boarding route R from stop v . Finally, Constraints (19) calculate the total journey time of each route $R \in \mathcal{R}$.