Exact and Metaheuristic Approaches for a Bi-Objective School Bus Scheduling Problem

Xiaopan Chen¹,², Yunfeng Kong¹*, Lanxue Dang¹,², Yane Hou¹,², Xinyue Ye³

¹ Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng, Henan 475004, China, ² College of Computer and Information Engineering, Henan University, Kaifeng, Henan 475004, China, ³ Department of Geography, Kent State University, Kent, OH 44242, United States of America

* yfkong@henu.edu.cn

Abstract

As a class of hard combinatorial optimization problems, the school bus routing problem has received considerable attention in the last decades. For a multi-school system, given the bus trips for each school, the school bus scheduling problem aims at optimizing bus schedules to serve all the trips within the school time windows. In this paper, we propose two approaches for solving the bi-objective school bus scheduling problem: an exact method of mixed integer programming (MIP) and a metaheuristic method which combines simulated annealing with local search. We develop MIP formulations for homogenous and heterogeneous fleet problems respectively and solve the models by MIP solver CPLEX. The bus type-based formulation for heterogeneous fleet problem reduces the model complexity in terms of the number of decision variables and constraints. The metaheuristic method is a two-stage framework for minimizing the number of buses to be used as well as the total travel distance of buses. We evaluate the proposed MIP and the metaheuristic method on two benchmark datasets, showing that on both instances, our metaheuristic method significantly outperforms the respective state-of-the-art methods.

Introduction

Due to the large scale of school merge and closures in China since the late 1990s, more and more primary and secondary schools have begun providing school bus service. According to the authors’ survey, in 2013, there are already thousands of school buses in service in cities such as Shanghai, Guangzhou, Shenzhen, Changchun, Xiamen and Qingdao. However, most of the school buses are operated by schools respectively. Recognizing the importance of school bus safety and operation efficiency, some local authorities, such as Binzhou, Chifeng, Wuli, Youxian and Zhuzhou, have proposed to provide regional level bus service for all schools. The Ministry of Education of China has launched a statewide school bus pilot program in six counties in 2011, aiming to explore the best practice of school bus operations in China. For local government and transportation company, school bus route planning is already a difficult task; still, it is crucial to serve with quality, equality and efficiency.
The school bus routing problem (SBRP) is concerned with how to transport students to and from schools in the safest, the most economical and convenient manner [1]. SBRP research has shown that the cost of school bus service can be reduced greatly by optimizing the school bus routes while satisfying service quality at the same time. SBRP consists of five sub-problems[2]: data preparation, bus stop selection, bus route generation, school bell time adjustment, and route scheduling. The bus route generation problem is to construct bus trips for each school. The route scheduling is also called bus scheduling wherein trips for each school are given. By bus scheduling, many sequences of bus trips are constructed and make each trip be finished within the time windows of designated school. The bus utilization can be improved by school bus routing and scheduling. Based on the bus service designed for a single-school or a multi-school system, SBRP can be classified into two general categories: single-school problem and multi-school problem[2].

Most existing SBRP researches focus on the single-school routing problem. Considering the limited capacity of each bus, the bus routing problem is usually modeled as the capacitated vehicle routing problem (CVRP). In a CVRP, a fleet of vehicles (the buses) serve a set of clients (the bus stops). The objective is to minimize the total travel distance and/or the number of buses required. For satisfying school bus service regulations, various constraints must be satisfied: bus capacity, student’s maximum riding time in a bus, and time windows of schools. Such problems are usually solved by exact methods [3–7] or (meta)heuristics such as ant colony optimization (ACO)[8, 9], time savings heuristic and sweep method [10], genetic algorithm (GA) [11–13], and tabu search [14–16]. Since CVRP is a NP-hard problem in strong sense, only the small-size problems can be solved by exact methods such as branch-and-cut and column generation[17, 18].

The multi-school SBRP is a special case of the pickup and delivery problem with time windows (PDPTW) [19]. In a PDPTW, goods or passengers (students) have to be transported from different origins (bus stops) to different destinations (schools). PDPTW is more difficult than CVRP [17]. Braca et al. developed a randomized location-based heuristic (RLBH) to construct bus routes by inserting a pair of bus stop and school into route solution sequentially [20]. Dang et al. proposed a PDPTW-based neighborhood search algorithm that is able to find much better solution for SBRP, though it would require redundant computation time [21]. Campbell et al. developed a three-phase heuristic algorithm for multi-school SBRP[22]. Since the size of multi-school SBRP is usually large in terms of the number of students, the number of bus stops and the size of bus fleet, it is difficult to be solved efficiently using existing PDPTW algorithms. Therefore, in most of the existing literature, the multi-school SBRP is separated into two sub-problems: the single-school routing problem and bus scheduling problem[19, 23–27]. When the bus trips for each single school are given, the bus scheduling problem is usually optimized by modeling it as a VRP with time windows (VRPTW)[25].

This paper proposes two approaches for solving the bi-objective school bus scheduling problem: an exact method of mixed integer programming (MIP) and a metaheuristic method. Both the homogeneous and heterogeneous fleet of buses are considered in this research. We rewrite the MIP formulations for bi-objective bus scheduling problem and solve it using the MIP optimizer IBM ILOG CPLEX. We also propose a simulated annealing algorithm framework to minimize the number of buses and the total travel distance. The performance of the proposed models and algorithms is tested on two sets of benchmark instances.

This paper is organized as follows. The next section defines the school bus scheduling problem. A bus type-based formulation for heterogeneous fleet school bus scheduling problem is introduced in this section. Section 3 proposes a generic simulated annealing framework. Based on the framework, three algorithms are implemented to solve single-school SBRP, homogeneous fleet school bus scheduling problem and heterogeneous fleet school bus scheduling
problem respectively. Section 4 shows the computation results for benchmark instances. The optimization performance of exact method and metaheuristic algorithm are compared in detail. The last section offers the research findings and conclusions.

**Problem Definition and Modeling**

**2.1 Problem definition**

In SBRP, a number of prerequisites, such as a set of schools, a set of student stops, a set of depots, a fleet of school buses and their geographic background, must be available for school bus route planning. The SBRP dataset should include both geographic and attribute information. The bus travel distances and times between schools, stops and depots can be prepared by geospatial analysis of the transportation network in GIS. The objective of school bus route planning is to minimize the number of school buses and the total bus travel distance while satisfying service qualities such as the student maximum riding time in a bus, the earliest and latest arrival times to their schools, and the maximum bus capacity.

Fig 1 illustrates two bus route plans for an example of school district. There are 2 schools and 8 student stops in the school district (Fig 1a). The student stops are filled with the same color as their schools. For school bus service in this district, Fig 1b shows a single-school bus route plan, in which there are two trips for school $s_1$ and $s_2$ respectively. Each trip consists of the depot, a sequence of bus stops and their destination school. One bus starts from the depot, picks up students of each bus stop in its trip, delivers them to their school, and then returns to the depot. When two trips can be served by the same bus in sequence without violating service constraints, they are assigned to the same bus (Fig 1c). This is a school bus scheduling problem, which may reduce the number of buses required. In Fig 1c, the dotted line represents the empty travel trip between two connected trips.

This paper focuses on the bus scheduling problem. In following sections, the MIP models for school bus scheduling problems are formulated in detail. Note that the bus trip in this paper refers to a bus route for a single school. The bus scheduling is to assign buses to trips, thus one bus can serve one or more bus trips.

**2.2 MIP formulation for homogeneous fleet school bus scheduling**

If the bus trips for each school are given, the bus scheduling problem is to assign buses to trips, and find the optimal solution with minimum number of buses and minimum bus travel distance. Let $V'$ be the set of single-school bus trips, and the number of trips is $n$. Each trip is composed of a sequence of bus stops and their destination school. The travel distance, $d_{ij}$, from trip $i$ to trip $j$, can be determined by the travel time from the designated school of trip $i$, to the first stop of trip $j$. Figure 1 shows two bus route plans for an example of school district.
The non-linear constraints can be transformed to linear inequalities Eq (5) by using the big M approach:

\[
\text{Eq (5)} \quad \text{subject to: } \sum_{j \in V'} x_{ij} = 1, \forall j \in V'
\]

\[
\sum_{j \in V'} x_{ij} = 1, \forall i \in V'
\]

The objective Eq (1) is to minimize the number of buses and total travel distance. The first part of Eq (1) denotes the number of buses. The parameter \( M_0 \) is a big number that ensures the number of buses as the primary objective and minimizes the two objectives in a lexicographic manner. Eqs (2) and (3) ensure that each trip is served by exactly one bus, and the bus must leave to another trip or the depot. Eq (4) assures that the bus serving trip \( i \) should arrive to school \( s(i) \) at any time within the school time window. Eq (5) states the accumulation of bus travel time: \((y_i + p_j + t_{ij} - y_j)x_{ij} = 0 \). If the arc \( (i,j) \) is traversed by a bus \( x_{ij} = 1 \), then \( y_i + p_j + t_{ij} = y_j \). The non-linear constraints can be transformed to linear inequalities Eq (5) by using the big M parameter. Eqs (6) and (7) are the constraints on decision variables.

\[
\text{Minimize: } M_0 \sum_{i \in V'} x_{0i} + \sum_{i \in V'} \sum_{j \in V'} d_{ij} x_{ij} + \sum_{i \in V'} x_{0i} d_{0i} + \sum_{j \in V'} x_{ji} d_{0j} \quad (1)
\]

\[
\text{Subject to: } \sum_{j \in V'} x_{ij} = 1, \forall j \in V' \quad (2)
\]

\[
\sum_{j \in V'} x_{ij} = 1, \forall i \in V' \quad (3)
\]

\[
e(s(i)) \leq y_i \leq l(s(i)), \forall i \in V' \quad (4)
\]

\[
y_i + t_{ij} + p_j - y_j \leq (1 - x_{ij})M, \forall i \in V, j \in V'|i \neq j \quad (5)
\]

\[
x_{ij} \in \{0, 1\}, \forall i, j \in V'|i \neq j \quad (6)
\]

\[
y_i \in \{0, 1, 2, \ldots\}, \forall i \in V' \quad (7)
\]
2.3 MIP formulation for heterogeneous fleet school bus scheduling

Unlike the assumption of a set of identical buses in the homogenous fleet model, a fleet of buses usually have different capacities and purchase cost in the heterogeneous fleet problem. There are two approaches of defining the three-index decision variables $x_{ijk}$ in problem formulation, where $k$ presents a vehicle [17, 28] or a type of vehicles [29]. Kim et al. developed a bus-based MIP model (MIPHE) for heterogeneous fleet school bus scheduling problem [25], and solved a set of problem instances by CPLEX. We rewrite the model by changing the bus-based decision variables to the bus type-based variables, aiming to reduce the model complexity. In the bus-based formulation, $k$ in the decision variables $x_{ijk}$ denotes the bus $k$ to serve arc $(i,j)$ in which  $i$ and $j$ denotes the single bus trip. But, in the bus type-based formulation, the $k$ denotes the bus of type $k$ to serve arc $(i,j)$. In addition, the fixed cost for each type of buses is also introduced in our objective function, which is different from that defined in MIPHE. Most parameters of the formulation are the same as the homogeneous problem defined in Table 1; additional parameters and decision variables are shown in Table 2.

The bi-objective MIP formulation for heterogeneous fleet school bus scheduling problem (MIPHE-B) is shown as follows:

Minimize : \[ \sum_{k \in K} \sum_{j \in V'} F_k x_{ijk} + \sum_{i \in V} \sum_{j \in V, j \neq i} d_i x_{ijk} + \sum_{i \in V} x_{0ik} d_{0i} + \sum_{j \in V} x_{ijk} d_{0j} \] \hspace{1cm} (8)

Subject to : \[ \sum_{k \in K} \sum_{i \in V} x_{ijk} = 1, \forall i, j \in V'|j \neq i \] \hspace{1cm} (9)

\[ \sum_{i \in V} x_{ijk} - \sum_{j \in V} x_{ijk} = 0, \forall u \in V', k \in K \] \hspace{1cm} (10)

\[ \sum_{i \in V} x_{0ik} \leq m_k, \forall k \in K \] \hspace{1cm} (11)

\[ c_i x_{ijk} \leq Q_k x_{ijk}, \forall i \in V, j \in V|j \neq i, k \in K \] \hspace{1cm} (12)

\[ c_j x_{ijk} \leq Q_k x_{ijk}, \forall j \in V, i \in V|i \neq i, k \in K \] \hspace{1cm} (13)

Table 2. Parameters and decision variables for heterogeneous fleet school bus scheduling problem.

| Parameters          | Description                                           |
|---------------------|-------------------------------------------------------|
| $K$                 | The set of bus types                                   |
| $F_k$               | The fixed cost of a bus of type $k, k \in K$          |
| $Q_k$               | The capacity of a bus of type $k, k \in K$            |
| $m_k$               | The number of buses of type $k, k \in K$              |
| $c_i$               | The number of students serviced by trip $i, j \in V$, let $c_0 = 0$ |
| $E_{ik}$            | If a bus of type $k$ can serve route $i, j \in V$, $i \neq j, k \in K$ |

Decision variables:

- $x_{ijk}$: If a bus of type $k$ serves trip $i$ and leaves to serve trip $j$ immediately, $x_{ijk} = 1$; otherwise $x_{ijk} = 0, i, j \in V, i \neq j, k \in K$.
- $y_{0}$: The arrival time at the destination school of trip $i$ by a bus of type $k, i \in V', k \in K$.

Table 2. Parameters and decision variables for heterogeneous fleet school bus scheduling problem.
The objective Eq (8) is to minimize the total fixed cost of buses and total travel distance. Eqs (9) and (10) ensure that each trip is served by only one bus, and the bus shall then leave to another trip or the depot. Eq (11) ensure that the number of used buses of type \( k \) does not exceed the total number of type \( k \) buses. Eqs (12) and (13) assure that a bus of type \( k \) must have enough capacity to serve a trip. Eq (14) assure that the bus of type \( k \) serving route \( i \) should arrive to school \( s(i) \) at any time within the school time window. Eq (15) state the accumulation of bus travel time: 
\[
y_{ik} + t_{ij} + p_j - y_{jk} \leq (1 - x_{ijk})M, \forall i \in V', j \in V', k \in K
\]
Eq (16) and (17) are the constraints on decision variables.

The bus type-based formulation for the school bus scheduling problem can reduce the model complexity significantly. Comparing to the bus based model MIPHE, the number of decision variables is reduced from \((n^2 + n)m\) to \((n^2 + n)k\), where \( n \), \( m \) and \( k \) represent the number of bus stops, the number of buses and the number of bus types. For an example of problem instance with 50 bus stops and a fleet of 20 buses, we need to define 51,000 decision variables in the bus-based formulation. If the buses are categorized into 3 types, we need to define 7,650 decision variables in bus type-based formulation, where 85% of decision variables are reduced. The parameters \( F_k \) in the objective function encourage the model solver to select small buses with smaller fixed costs. In model building for a problem instance, the model can be simplified by introducing parameters \( E_{ik} \). If \( E_{ik} = 0 \) (or \( E_{ik} = 0 \)), the decision variable \( x_{ijk} = 0 \) (or \( y_{ik} = 0 \)) can be deleted from the model, and the related elements in the objective function and constraints can also be deleted. In fact, all the Eqs (12) and (13) can be removed from the model. In addition, if \( e(s(i)) + t_{ij} + p_j > l(s(j)) \), the connection between trips \( i \) and \( j \) is infeasible. In such a case, the decision variable \( x_{ijk} = 0 \); thus the related elements in the objective function and constraints can also be deleted, and some Eq (15) will also be deleted.

Since the difference of capacity and cost between a fleet of buses is considered in the heterogeneous fleet problem, it is more complex than the homogenous fleet problem. Technically, the MIP model for the homogeneous fleet problem is a special case of the heterogeneous fleet problem, where all the buses are assumed to be identical. In addition, compared with the bus-based model MIPHE, the bus type-based model MIPHE-B reduces the number of decision variables significantly; thus, its computational complexity can be reduced.

### Metaheuristic Algorithm

#### 3.1 Algorithm framework

In this paper, the multi-school SBRP is decomposed into single school bus routing problem and school bus scheduling problem. The former problem is equivalent to open CVRP with a constraint of maximum riding time of students. The latter can be converted to VRPTW by treating each trip as a virtual bus stop [25]. Based on the neighborhood search operators for VRP, we propose a two-stage algorithm framework for both problems in the following steps:

1. DataInput();
2. BuildInitialSolution();
3. RouteMinimize_ILS_SA();
4. LengthMinimize_ILS_SA();
5. OutputSolution();

Step 1 reads the data of a SBRP instance. Step 2 generates a set of feasible routes as the initial solution. The feasible routes must meet specific constraints. Step 3 is to minimize the number of routes using neighborhood operators coupled with simulated annealing. Since the multi-objective function in step 3 encourages reducing the number of bus routes, additional step 4 is used to minimize the total travel distance. The output solution in step 3 is fed as the initial solution of step 4.

There are two general objectives for SBRP in this research: minimizing the number of buses and minimizing the total travel distance. Since the capital cost per bus is significantly higher than the travel cost per distance unit [30], the number of routes is of higher priority than the travel distance. However, it is very difficult to obtain a solution that all the objectives are optimal simultaneously. The objective function of total travel distance commonly used in VRP may lead the search to the solutions with a small travel distance and makes it impossible to remove routes[31]. Therefore, we use a lexicographic multi-objective function to evaluate the neighbourhood solutions in step 3:

\[
\text{Cost}(S) = \alpha |R| - \beta \sum_{r \in R} |r|^2 + \gamma \text{Dist}(R)
\]  

In the function, the $|R|$, Dist$(R)$ and $|r|^2$ are the number of routes, the total length of all routes, and the sum of squared route stop numbers respectively. Given the parameters $\alpha \gg \beta \gg \gamma$, the neighbourhood moves prefer to delete some routes if possible. In step 4, only the function of Dist$(R)$ is used to minimize the total travel distance.

After the construction of an initial route solution by a simple heuristic algorithm, the key steps of this algorithm are to improve the solution by exploring the neighborhood solutions. Local search is a metaheuristic method for solving combinatorial optimization. Various local search algorithms iteratively improves the solution of a problem by applying local changes to current solution until optimal solution is deemed to be found or a stop criterion is met. In the process of searching a better solution, once a feasible neighborhood solution is identified and its objective is better than the current solution, the current solution will be replaced by the new one. Local search has proven to be an effective technique for generating good solutions to instances of the classical VRP as well as variants such as VRPTW[32].

A general local search operator exchanges $\mu$ nodes from one route with $\lambda$ nodes from another. The local search operators such as one-point move, two-point move and 2-opt move can work on a single route or different routes, while cross-exchange is an inter-route operators that only work on two routes simultaneously[33]. One-point move attempts to relocate an existing node to a new position, which can reduce the number of routes and create more opportunities for other operators. Two-point move swaps two nodes to reduce route distance. 2-opt move tries to improve a solution by removing two edges from the solution and replace them with two new edges. Cross-exchange move tries to remove two edges respectively from two different routes and replace them with four new edges. The repeat of the four local search operators will improve the route solution gradually.

The local search algorithm has drawbacks: the quality of solution is closely related with the initial solution and the spatiotemporal structure of the problem instance. Meanwhile, the procedure is very easily trapped into local optima. In general, such a local optimum is not a global optimum[34]. Even worse, there is usually no guarantee that the value of the objective function
at an arbitrary local optimum is close to the optimal value. Some methods have been proposed to solve the problem of local optimum, such as simulated annealing and tabu search. In this paper, the technique of simulated annealing[35] is combined with local search operators. In the local search operations, we accept some solutions that make route length longer by simulated annealing, but may escape from local optima and realize the search in depth and diversification. The general local search procedure coupled with simulated annealing is designed in the following steps:

**Algorithm 1: The local search with simulated annealing**

**Input:** the SBRP instance \((S)\), the route solution \((sol)\), the initial temperature \((t_0)\), the cooling rate \((\alpha)\), the acceptance criterion for neighborhood solutions \((rule)\), and the maximum loops of local search \((maxloop)\)

1. \(t \leftarrow t_0;\)
2. \(sol_{best} \leftarrow sol;\)
3. for \((i = 1; i \leq maxloop; i++)\) {
   4. \(perm = \text{NodePermutation}(S)\)
   5. \(OPM(S, perm, t, rule);\)
   6. \(TPM(S, perm, t, rule);\)
   7. \(TOM(S, perm, t, rule);\)
   8. \(CE(S, t, rule);\)
   9. \(t \leftarrow \alpha \times t;\)
4. }

**Output:** the best solution.

This procedure can be implemented for school bus routing and bus scheduling for the purpose of minimizing the number of routes and minimizing the total travel distance. Line 4 permutes the nodes randomly, which will change the node searching sequence in each iteration for solution diversification. From line 5 to line 8 in the loop, one-point move search, two-point move search, 2-opt move search, and cross-exchange search shall be executed sequentially. In each local search function, according to feasibility check, objective evaluation and acceptance criterion; the neighborhood solution will either be accepted as the current best solution so far or rejected. In line (9) the temperature drops, and the possibility of accepting a new solution with longer travel distance in next loops will be decreased gradually.

### 3.2 Local search operators

The route solution \(sol\) is gradually improved by four local search operators: one-point move, two-point move, 2-opt move and cross-exchange move. Algorithm 2 shows the details of the one-point move search. The search procedures for other three operators are similar to one-point move search.

**Algorithm 2: OPM \((S, perm, rule)\)**

**Input:** the SBRP instance \((S)\), the sequential or randomized node list \((perm)\), and the criteria to accept a new solution \((rule)\). Note that the instance \(S\) includes the current best route solution \((sol_{best})\), the current temperature \((t)\) and other information.
1. FOR EACH \((p \text{ in } \text{perm})\) {
2. \(\text{Nlist} \leftarrow \text{OPM\_neighborhood}(p, S)\);
3. FOR EACH \((p^* \text{ in } \text{Nlist})\) {
4. IF \((\text{check\_feasibility}(p, p^*) = \text{false})\) CONTINUE;
5. \(\text{sol}^* \leftarrow \text{OPM\_neighborhood\_solution}(p, p^*)\);
6. IF \((\text{is\_better}(\text{sol}_{\text{best}}, \text{sol}^*) = \text{false})\) CONTINUE;
7. IF \((\text{rule} = \text{FA})\) \(\{\text{sol}_{\text{best}} \leftarrow \text{sol}^*; \text{BREAK};\}\)
8. IF \((\text{rule} = \text{BA})\) {
9. IF \((\text{is\_better2}(\text{sol}_{\text{best}}, \text{sol}^*) = \text{false})\) CONTINUE;
10. \(\text{sol}_{\text{best}} \leftarrow \text{sol}^*;\)
11. } 
12. IF \((\text{rule} = \text{BA})\) \(\text{sol}_{\text{best}} \leftarrow \text{sol}^*\) 
13. }

\textbf{Output}: the current best solution \((\text{sol}_{\text{best}})\)

For each node \(p\), the algorithm constructs its neighborhood list \((\text{Nlist})\) at first (line 2). For each neighbor node \(p^*\) in the neighborhood list, if the node \(p\) can be inserted after node \(p^*\) without violation of the constraints, the objective function for this feasible move will be evaluated. Otherwise, the operator will try another neighbor position (line 4). If the neighborhood solution (line 5) is better than the current best solution \(\text{sol}_{\text{best}}\) or, at least acceptable with the simulated annealing criteria (line 6), the best solution will be updated according to acceptance criterion \((\text{rule})\): first acceptance (FA) or best acceptance (BA) (line 7–12). The function \(\text{is\_better}(\text{sol}_{\text{best}}, \text{sol}^*)\) evaluates the solution quality of \(\text{sol}^*\) compared with that current best solution \(\text{sol}_{\text{best}}\). The function \(\text{is\_better2}(\text{sol}_{\text{best}}, \text{sol}^*)\) is used to select the best neighborhood solution. For minimizing the number of routes, the three elements in objective Eq (18) are in a lexicographic order.

### 3.3 Solvers for school bus routing and scheduling

Based on the algorithm framework and local search operators discussed in Section 3.1 and 3.2, we developed three solvers for single-school SBSP, homogeneous fleet school bus scheduling problem and heterogeneous fleet school bus scheduling problem respectively. The solvers are used to plan bus routes. Firstly, the bus trips for each school in a multi-school system is generated by the single-school solver. Secondly, the buses are assigned to the trips by using the homogeneous fleet or heterogeneous fleet school bus scheduling solver. The output data from the first solver is fed into the second solver. In each solver, a simple savings-based heuristic algorithm is used to construct an initial solution. A two-stage metaheuristic algorithm is then used to improve the solution. The first stage minimizes the number of routes, which reduces the total number of buses needed for students. The second stage optimizes the total bus travel distance, which will reduce the total bus travel cost. The procedures for the two stages are designed in the same manner, except for the evaluation of solution objectives.

The VRP with a heterogeneous fleet is considered much harder than corresponding problems with a homogeneous fleet[37]. The heterogeneous fleet school bus scheduling problem is also much more difficult to solve than the homogeneous fleet problem. For initial solution
construction, we divide the heterogeneous fleet problem into multiple homogeneous fleet sub-problems according to the types of buses. First, the trips and the types of buses are sorted in an ascending order. Second, the trips are divided into multiple groups according to the capacities of buses; a type of buses with minimum capacity is assigned to each group of trips as well. Third, the saving-based heuristic algorithm is used to construct the route solution for each group of trips. In case the number of one type of buses is not enough to serve the sub-problem, the types of buses for all routes need to be adjusted appropriately. Fourth, the route solutions of each group are merged into an initial route solution for the whole heterogeneous fleet problem. The initial solution is then improved by the two-stage algorithm. Comparing with the neighborhood point moves in solving homogeneous fleet problem, it is needed to check the bus types for any possible point moves.

The solver programs are implemented by C++ programming and compiled as 32-bit command-line executables with parameter setting options. The parameters includes the maximum number of loops (maxloop), the initial temperature for simulated annealing ($t_0$), the cooling rate of temperature ($\alpha$), the length of neighborhood list (NList), the acceptance criteria for solution updating (rule), and the randomized node search strategy (random).

### Computational Results

#### 4.1. Results for HOMO and HETERO instances

The HOMO and HETERO benchmark instances are collected from literature [25]. There are 15 instances for homogeneous fleet problem and heterogeneous fleet problem respectively. Table 3 shows the size of all instances. For each instance, the school data includes the school ID, the spatial coordinates $x$ and $y$ and the school time window. The bus trips for each school are given, which include the trip ID, the coordinates of the first stop in the trip, destination school ID, and the bus service time. The bus data for heterogeneous fleet instances include the bus ID and the bus capacity.

We build models for all the instances according to the MIP formulations proposed in Section 2.2 and 2.3. The model parameters are the same as literature [25]: the travel distance $d_{ij}$ is

| Table 3. The school bus scheduling instances HOMO and HETERO. |
|---------------------------------|
| Homogeneous fleet instances      | Heterogeneous fleet instances |
| Instance | #schools | #trips | #students | #buses | Instance | #schools | #trips | #students | #buses |
| homo1    | 1        | 3      | 143       | 3       | hetero1  | 1        | 7      | 335       | 7       |
| homo2    | 2        | 13     | 611       | 13      | hetero2  | 2        | 13     | 596       | 13      |
| homo3    | 3        | 20     | 975       | 20      | hetero3  | 3        | 21     | 1012      | 21      |
| homo4    | 4        | 23     | 1154      | 23      | hetero4  | 4        | 24     | 1199      | 24      |
| homo5    | 5        | 29     | 1448      | 29      | hetero5  | 5        | 32     | 1610      | 32      |
| homo10   | 10       | 49     | 2405      | 49      | hetero10 | 10       | 58     | 3002      | 58      |
| homo20   | 20       | 106    | 5097      | 106     | hetero20 | 20       | 120    | 6012      | 120     |
| homo30   | 30       | 161    | 8155      | 161     | hetero30 | 30       | 187    | 9415      | 187     |
| homo40   | 40       | 234    | 11528     | 234     | hetero40 | 40       | 239    | 12136     | 239     |
| homo50   | 50       | 309    | 15238     | 309     | hetero50 | 50       | 286    | 14352     | 286     |
| homo60   | 60       | 372    | 18551     | 372     | hetero60 | 60       | 345    | 17350     | 345     |
| homo70   | 70       | 430    | 21554     | 430     | hetero70 | 70       | 408    | 20723     | 408     |
| homo80   | 80       | 461    | 23066     | 461     | hetero80 | 80       | 469    | 23686     | 469     |
| homo90   | 90       | 514    | 25736     | 514     | hetero90 | 90       | 514    | 25736     | 514     |
| homo100  | 100      | 562    | 28175     | 562     | hetero100| 100      | 562    | 28175     | 562     |

doi:10.1371/journal.pone.0132600.t003
an integer number calculated by Euclidean distance with a truncated fraction; the bus speed is assumed to be 1 distance unit per time unit, and thus $t_{ij} = d_{ij}$. In the objective function, the travel distance from the bus depot to the first trip and from the last trip to the depot are not considered in the total distance travelled by all the buses. In the model MIPHE-B, the bus type data are counted from the bus data. There are three types of buses in the heterogeneous fleet instances; and the fixed costs for each type of buses are assumed to be 100000, 90000 and 80000 for buses with capacity of 70, 50 and 40 respectively. The parameters $M$ and $M_0$ are set to $10^9$ and $10^5$ respectively.

All the models are solved by the MIP solver IBM ILOG CPLEX 12.6 (64bits). CPLEX is installed on a personal computer with the following hardware configuration: Intel(R) Core2 Duo E7600 3.06G CPU, 4GB RAM and 64-bit Windows 7 operating system. The parameters for CPLEX solver are set to its default values, except that the maximum computation time is set to 3600s and the MIPGap is set to $10^{-12}$.

Tables 4 and 5 show the CPLEX results of the homogeneous fleet and heterogeneous fleet instances respectively. The columns N, D and T provide the number of buses required, the deadhead bus travel distance, and the CPU time that are used by the CPLEX solver. The deadhead bus travel distance is defined as the idle distance covered by the bus between the destination school of the first trip to the first bus stop of the next trip without carrying any students, when a bus can serve the two trips in sequence. The column Gap indicates the CPLEX solution status: optimal or the optimality gap. The instances of homo1~homo60 can be solved with optimal solutions; while the instances of homo70~homo100 can be solved with suboptimal solutions. Since the new version of CPLEX is used in our research, we get better solutions with the less number of buses for large instances of homo70~homo100. For the last two instances (homo90 and homo100), we get longer travel distances. However, we provide the solutions with less buses, which will reduce the total cost of school bus service. Note that our optimal solution for homo60 is different from that in literature [25].

Table 5 also shows the number of each type of buses used for the instances. The numbers in bracket denote the number of buses with capacity of 40, 50 and 70 respectively.

| Instance | MIPHO[25] | MIPHO-B |
|----------|-----------|---------|
| homo1    | N = 3     | D = 0   | T(s) = 0.33 | opt. | N = 3 | D = 0 | T(s) = 0.02 | opt. |
| homo2    | 10        | 119    | 0.06    | opt. | 10    | 119    | 0.06    | opt. |
| homo3    | 10        | 418    | 0.08    | opt. | 10    | 418    | 0.05    | opt. |
| homo4    | 10        | 441    | 0.08    | opt. | 10    | 441    | 0.03    | opt. |
| homo5    | 15        | 380    | 0.08    | opt. | 15    | 380    | 0.03    | opt. |
| homo10   | 17        | 806    | 0.17    | opt. | 17    | 806    | 0.09    | opt. |
| homo20   | 45        | 1265   | 0.38    | opt. | 45    | 1265   | 0.19    | opt. |
| homo30   | 57        | 2132   | 25.72   | opt. | 57    | 2132   | 7.88    | opt. |
| homo40   | 63        | 3091   | 91.31   | opt. | 63    | 3091   | 7.86    | opt. |
| homo50   | 88        | 3421   | 241.31  | opt. | 88    | 3421   | 22.75   | opt. |
| homo60   | 90        | 4260   | 1303.55 | opt. | 90    | 4255   | 192.16  | opt. |
| homo70   | 101       | 4962   | 1 hour  | -     | 99    | 4921   | 1 hour  | 1.01  |
| homo80   | 109       | 5296   | 1 hour  | -     | 106   | 5252   | 1 hour  | 5.43  |
| homo90   | 124       | 5212   | 1 hour  | -     | 117   | 5864   | 1 hour  | 11.23 |
| homo100  | 135       | 5425   | 1 hour  | -     | 127   | 6211   | 1 hour  | 16.52 |

doi:10.1371/journal.pone.0132600.t004
results show that our model MIPHE-B for the heterogeneous fleet school bus scheduling problem outperforms the model MIPHE significantly. Since the number of buses is rather large compared to the types of buses, the bus-based formulation of MIPHE is computationally more complex than the bus type-based formulation. Thus, the CPLEX can only solve the small instances (hetero1~hetero10). Our bus type-based formulation MIPHE-B reduces the model complexity in terms of the number of decision variables, the number of constraints, and the model file size. Modeling the instances with MIPHE-B, CPLEX can solve larger instances (hetero20~hetero50) optimally; the optimal problem size is increased from 58 (hetero10) trips to 286 trips (hetero50). For the large instances (hetero60~hetero100), the CPLEX can provide feasible solutions, and the results for some instances (hetero60~hetero80) are near to the optimal solutions. At the same time, since the difference between the fixed costs of buses is not considered in MIPHE model, the optimal solutions for a instance obtained from MIPHE and MIPHE-B may be different and cannot be compared directly. For example, the bus travel distances for hetero5 and hetero10 from MIPHE-B are longer than that from MIPHE. However, we provide the optimal configuration of the three types of buses. The parameter of fixed bus cost ($F_k$) in MIPHE-B model can guide the MIP solver to select smaller, lower cost buses as many as possible.

The heuristic results for HOMO and HETERO instances are listed in Tables 6 and 7 respectively. Since simulated annealing is a generic probabilistic metaheuristic, the solver proposed in Section 3 was executed 10 times over each instance. For both homogeneous fleet and heterogeneous fleet instances, the size of neighborhood list (NList) is set to n/2, where n is the number of bus trips. The initial temperature for simulated annealing is set to 5.0/n, and the cooling rate $\alpha$ is 0.995. The maximum number of loops (maxloop) is set to 200. The best, average and standard deviation of the number of buses and the total distance among the 10 solutions are listed in columns N_best, N_ave, N_dev, or D_best, D_ave, D_dev, which are useful to verify the effectiveness of the algorithm.

Table 6 shows the heuristic results for the homogeneous fleet instances. The columns N_best, N_ave and N_dev, represent the best, average and standard deviation of the number of buses
among the 10 solutions respectively. The columns $D_{best}$, $D_{ave}$ and $D_{dev}$ represent the best, average and standard deviation of the distance among the 10 solutions respectively, and the column $Ngap$ and $D_{gap}$ represents the percentage difference of the number of buses and distance between the best heuristic solution and the CPLEX solution for each instance in Table 4, respectively. The column $T$ denotes the solver time in seconds.

Compared with CPLEX, the heuristic approach to homogeneous fleet school bus scheduling can find the optimal solutions for small-size instances (homo1~homo10), sub-optimal solutions for medium-size instances (homo20~homo60), and better solutions for large-size instances (homo90 and homo100). In addition, compared with the solution results in literature [25], our SA-heuristic algorithm outperforms Kim’s heuristic algorithm in terms of the number of buses and the total travel distance; for large instances homo60~homo100, on average, the number of buses and the total travel distance are reduced by 2.35% and 2.83% respectively.

Table 7 shows the heuristic results for the heterogeneous fleet instances. Compared with the solutions in literature [25], we provide the number of buses by bus type. In addition, the objective differences between the heuristic solutions and CPLEX solutions are shown in column Gap1, Gap2 and Gap3, which denote the percentage gaps of the number of buses, the fixed cost of buses and the travel distance respectively. The solution comparison between exact and heuristic approach shown in Fig 2 is interesting. For small-size instances (hetero1~hetero5, the number of trips between 7 and 32), both the two approaches can provide optimal solutions in terms of the number of buses, the total fixed cost and the total travel distance. For medium-size instances (hetero10~hetero50, the number of trips between 58 and 286), CPLEX can find the optimal solutions, and our algorithm can find good enough results. For example, our algorithm can find the optimal number of buses on instances hetero10 and hetero20. For large-size instances (hetero60~hetero100, the number of trips between 342 and 562), CPLEX is able to find sub-optimal solution with small objective gaps for instance hetero60~hetero80, whereas high quality solutions for instances hetero90 and hetero100 are difficult to find. However, the
heuristic approach has obvious advantage in solving the larger instances in both the solution quality and computation time.

4.2. Results for RSRB and CSCB instances

Another set of sixteen benchmark instances are selected from literature [27]. The instances shown in Table 8 are classified into two groups: the random spatial distribution of schools and bus stops (RSRB) and the clustered distribution (CSCB). These instances have different numbers of students, bus stops and schools. The capacity of the vehicle is assumed to be 66, and its average speed is 20 miles per hour. For the purpose of comparison, the service time of picking up or delivering students at each stop is calculated by the regression model developed in [20], the same as in literature [27]. The maximum riding time (MRT) of students in a bus is set to 2700 seconds (45 minutes) or 5400 seconds (90 minutes). Since there is no bus depot

![Figure 2. Objective difference between metaheuristic and CPLEX solutions for the heterogeneous fleet instances.](image)
information in the benchmark instances, we assume the bus depot is located at coordinates (105600, 105600).

For school bus scheduling, the bus trips for each school are solved by the metaheuristic algorithm proposed in Section 3. The travel distance $d_{ij}$ between any two nodes is a float number in inches and is calculated by Manhattan distance. The travel time $t_{ij}$ between any two nodes is an integer number in seconds, it is estimated by the travel distance and the average bus speed, i.e. $t_{ij} = d_{ij}/29.33333$. The pickup time at student stop is an integer number estimated by the formula $19+2.6/n$ ($n$ is the number of students at the stop); and the delivery service time at school stop is also an integer number, which is estimated by the formula $29+1.9/n$ ($n$ is the number of students served by the trip). The output trip data includes these features: the trip ID, the first student stop ID and its coordinates, the destination school ID and its coordinates and time windows, the bus travel distance, the trip service time, and the number of students served by the trip. The service time for each trip includes the bus travel time, the students’ pickup time and unloading time at the destination school. The column #trips1 in Table 8 presents the total number of trips for each instance when the MRT is 2700 seconds; and column #trips2 presents the total number of trips for each instance when the MRT is 5400 seconds.

Based on the trips for each school, we build all the models for the instances according to the MIPHO-B formulations proposed in Section 2.2. The parameter $M$ in objective function is set to $10^9$. Parameters such as bus travel distance and time are also calculated, same as the settings in single-school problem. The models are solved by CPLEX 12.6. The CPLEX parameters are set to default values, except that the maximum computation time is set to 3600s and the MIP-Gap is set to $10^{-15}$.

All the instances in Table 8 are also solved by our solver for homogeneous fleet school bus scheduling problem. The solver was executed 10 times over each instance. The initial temperature for simulated annealing is set to $1.2/n$, where $n$ is the number of bus trips of each instance. The cooling rate $\alpha$ is set to 0.99. The size of neighborhood list (NList) is set to $n/2$; and the maximum number of loops (maxloop) is 200.

Tables 9 and 10 exhibit the computational results for RSRB and CSCB instances respectively. The 2700 and 5400 in column Instance denote the students’ maximum riding time in

| Instance | #schools | #stops | #students | #trips1 | #trips2 |
|----------|----------|--------|-----------|---------|---------|
| RSRB01   | 6        | 250    | 3409      | 60      | 55      |
| RSRB02   | 12       | 250    | 3670      | 76      | 62      |
| RSRB03   | 12       | 500    | 6794      | 131     | 112     |
| RSRB04   | 25       | 500    | 6805      | 158     | 120     |
| RSRB05   | 25       | 1000   | 13765     | 251     | 225     |
| RSRB06   | 50       | 1000   | 12201     | 293     | 215     |
| RSRB07   | 50       | 2000   | 26912     | 486     | 437     |
| RSRB08   | 100      | 2000   | 31939     | 658     | 542     |
| CSCB01   | 6        | 250    | 3907      | 69      | 64      |
| CSCB02   | 12       | 250    | 3204      | 76      | 57      |
| CSCB03   | 12       | 500    | 6813      | 152     | 112     |
| CSCB04   | 25       | 500    | 7541      | 193     | 128     |
| CSCB05   | 25       | 1000   | 16996     | 332     | 277     |
| CSCB06   | 50       | 1000   | 18232     | 410     | 309     |
| CSCB07   | 50       | 2000   | 27594     | 614     | 449     |
| CSCB08   | 100      | 2000   | 27945     | 745     | 484     |

doi:10.1371/journal.pone.0132600.t008

Table 8. The school bus routing and scheduling instances RSRB and CSCB.
seconds. The columns N, D and T denote the total number of buses, the total bus travel distance and the solver time in seconds. The column Gap indicates the solution status for CPLEX solver; the columns Ngap and Dgap refer to the percentage difference of the number of buses and the travel distance between CPLEX solutions and the heuristic solutions. For the

Table 9. Results for the RSRB benchmark instances.

| Instance     | N     | D     | Gap(%) | T(s) | N_best | N_dev | N_ga | D_best | D_dev | D_ga | T(s) |
|--------------|-------|-------|--------|------|--------|-------|------|--------|-------|------|------|
| RSRB01_2700 | 26    | 6914083 | 0.06   | 2.80 | 26     | 0.00  | 0.00 | 6914092.46 | 0.00  | 0.00  | 2.80 |
| RSRB02_2700 | 27    | 10028708 | 0.09   | 4.18 | 27     | 0.00  | 0.00 | 10120784.23 | 0.00  | 0.02  | 4.18 |
| RSRB03_2700 | 48    | 15761527 | 0.42   | 19.86| 48     | 0.00  | 0.00 | 15761542.48 | 0.00  | 0.00  | 19.86 |
| RSRB04_2700 | 58    | 16209789 | 0.49   | 25.28| 58     | 0.00  | 0.00 | 16284162.52 | 0.00  | 0.06  | 25.28 |
| RSRB05_2700 | 94    | 26414196 | 1.44   | 25.93| 94     | 0.00  | 0.00 | 26414264.91 | 0.00  | 0.14  | 25.93 |
| RSRB06_2700 | 87    | 25484524 | 1.76   | 123.03| 87     | 0.00  | 0.00 | 25572260.46 | 0.00  | 0.04  | 123.03 |
| RSRB07_2700 | 149   | 42328760 | 9.92   | 109.58| 149    | 0.00  | 0.00 | 42509078.51 | 0.43  | 0.34  | 109.58 |
| RSRB08_2700 | 136   | 46767310 | 1785.92| 1652.50| 139    | 0.52  | 2.21 | 46560688.64 | 175338.4 | 0.43 | 1652.50 |
| RSRB09_2700 | 27    | 6304173 | 0.05   | 2.79 | 27     | 0.00  | 0.00 | 6304182.28 | 1361.2 | 0.00  | 2.79 |
| RSRB10_2700 | 23    | 8665826 | 0.00   | 2.67 | 23     | 0.00  | 0.00 | 8665846.47 | 0.00  | 0.00  | 2.67 |
| RSRB11_2700 | 72    | 19492477 | 175338.4 | 1652.50 | 140    | 0.67  | 0.72 | 42749715.12 | 141358.9 | 0.40 | 1652.50 |

Average 74.63 21993413.9 - 132.57 74.88 0.07 0.18 22030248.82 39518.5 0.18 316.21

doi:10.1371/journal.pone.0132600.t009

Table 10. Results for the CSCB benchmark instances.

| Instance     | N     | D     | Gap(%) | T(s) | N_best | N_dev | N_ga | D_best | D_dev | D_ga | T(s) |
|--------------|-------|-------|--------|------|--------|-------|------|--------|-------|------|------|
| CSCB01_2700 | 32    | 8238295 | 0.06   | 4.03 | 32     | 0.00  | 0.00 | 8238320.96 | 0.00  | 0.00  | 4.03 |
| CSCB02_2700 | 26    | 9047666 | 0.19   | 4.09 | 26     | 0.00  | 0.00 | 9049453.10 | 5413.7 | 0.02  | 4.09 |
| CSCB03_2700 | 58    | 15180749 | 0.62   | 23.81| 58     | 0.00  | 0.00 | 15180775.18 | 0.00  | 0.00  | 23.81 |
| CSCB04_2700 | 62    | 20111639 | 0.58   | 42.66| 62     | 0.00  | 0.00 | 20111678.69 | 60773.4 | 0.25  | 42.66 |
| CSCB05_2700 | 113   | 38837810 | 5.94   | 543.31| 141    | 0.00  | 0.00 | 38916850.74 | 45393.8 | 0.20  | 543.31 |
| CSCB06_2700 | 139   | 42579350 | 312.00 | 1305.45| 140    | 0.67  | 0.72 | 42749715.12 | 141358.9 | 0.40  | 1305.45 |
| CSCB07_2700 | 29    | 7429868 | 1.22   | 54.66| 29     | 0.00  | 0.00 | 7429873.97 | 0.00  | 0.25  | 54.66 |
| CSCB08_2700 | 22    | 6934233 | 0.08   | 2.32 | 22     | 0.00  | 0.00 | 6934243.00 | 0.00  | 0.00  | 2.32 |
| CSCB09_2700 | 42    | 10125783 | 0.19   | 345.40| 42     | 0.00  | 0.00 | 10125802.50 | 495.0  | 0.00  | 345.40 |
| CSCB10_2700 | 45    | 13752851 | 2.32   | 758.36| 45     | 0.29  | 0.00 | 13752859.94 | 95146.0 | 0.36  | 758.36 |
| CSCB11_2700 | 99    | 23805471 | 9.19   | 98.58| 100    | 0.00  | 0.00 | 24087785.53 | 0.00  | 1.19  | 98.58 |
| CSCB12_2700 | 104   | 23525340 | 3.46   | 172.86| 104    | 0.00  | 0.00 | 23552967.39 | 32467.6 | 0.12  | 172.86 |
| CSCB13_2700 | 146   | 32561350 | 10.83  | 445.88| 146    | 0.00  | 0.00 | 32620032.80 | 2473.6  | 0.18  | 445.88 |
| CSCB14_2700 | 130   | 37933090 | 46.97  | 758.36| 130    | 0.63  | 0.00 | 38707240.90 | 162638.3 | 2.04  | 758.36 |

Average 84.38 22805047.9 - 435.40

doi:10.1371/journal.pone.0132600.t010
Table 11. Comparison of our algorithm with existing algorithm.

| Instances      | \( Z_{park}^{[27]} \) | \( Z_{SA} \) | Gap(%) | Instances      | \( Z_{park}^{[27]} \) | \( Z_{SA} \) | Gap(%) |
|----------------|------------------------|-------------|--------|----------------|------------------------|-------------|--------|
| RSRB01_2700   | 35                     | 26          | 34.62  | CSCB01_2700    | 39                     | 32          | 17.95  |
| RSRB02_2700   | 32                     | 27          | 18.52  | CSCB02_2700    | 33                     | 26          | 21.21  |
| RSRB03_2700   | 66                     | 48          | 37.50  | CSCB03_2700    | 66                     | 58          | 12.12  |
| RSRB04_2700   | 68                     | 58          | 17.24  | CSCB04_2700    | 72                     | 62          | 13.89  |
| RSRB05_2700   | 124                    | 94          | 31.91  | CSCB05_2700    | 135                    | 114         | 15.56  |
| RSRB06_2700   | 103                    | 87          | 18.39  | CSCB06_2700    | 138                    | 120         | 13.04  |
| RSRB07_2700   | 185                    | 149         | 24.16  | CSCB07_2700    | 201                    | 173         | 13.93  |
| RSRB08_2700   | 178                    | 139         | 28.06  | CSCB08_2700    | 183                    | 154         | 15.85  |
| RSRB01_5400   | 31                     | 27          | 14.81  | CSCB01_5400    | 35                     | 29          | 17.14  |
| RSRB02_5400   | 30                     | 23          | 30.43  | CSCB02_5400    | 27                     | 22          | 18.52  |
| RSRB03_5400   | 61                     | 46          | 32.61  | CSCB03_5400    | 52                     | 42          | 19.23  |
| RSRB04_5400   | 56                     | 41          | 36.59  | CSCB04_5400    | 57                     | 45          | 21.05  |
| RSRB05_5400   | 106                    | 80          | 32.50  | CSCB05_5400    | 115                    | 100         | 13.04  |
| RSRB06_5400   | 82                     | 72          | 13.89  | CSCB06_5400    | 123                    | 104         | 15.45  |
| RSRB07_5400   | 159                    | 141         | 12.77  | CSCB07_5400    | 164                    | 146         | 10.98  |
| RSRB08_5400   | 158                    | 140         | 12.86  | CSCB08_5400    | 156                    | 130         | 16.67  |
| Average       | 92.13                  | 74.89       | 24.80  | Average        | 99.75                  | 84.81       | 15.98  |

doi:10.1371/journal.pone.0132600.t011

There are several findings from Tables 9 and 10. First, the number of buses can be reduced significantly by bus scheduling. Comparing to the single-school based bus route plan, only about 30% of buses are needed for both RSRB and CSCB instances. Second, the homogeneous fleet bus scheduling problem, which is modeled as VRPTW without capacity constraints, can be efficiently solved by CPLEX with optimal solutions. Third, the local search algorithm coupling with simulated annealing is also effective to solve homogeneous fleet bus scheduling problem. The algorithm proposed in this research can also find the minimum number of buses for the benchmark instances except a few instances.

Table 11 shows the comparison of our algorithm with the heuristic algorithm in [27]. \( Z_{park} \) and \( Z_{SA} \) denote the number of required buses obtained from the two algorithms. The Gap column shows the percentage improvement of our algorithm, which is computed as \((Z_{park} - Z_{SA}) / Z_{park}\). On average, our route solutions reduce the number of required buses by 24.8% and 15.98% for RSRB and CSCB instances respectively. It is mainly due to Park’s algorithm is a construction heuristic algorithm with simple improvement operators only.

**Conclusions**

For a multi-school system, the school bus scheduling problem seeks to optimize bus schedules to serve all the given trips considering the school time windows. Two approaches to solve the bi-objective school bus scheduling problem are discussed in this paper. This research manages to make two contributions: the rewriting of MIP formulation for heterogeneous-fleet school bus scheduling, and the design of a local search algorithm coupling with simulated annealing to minimize the number of buses and the total travel distance.

This research also reports three findings. First, the school bus scheduling problem can be efficiently solved with optimal or sub-optimal solution by MIP solver CPLEX. School bus scheduling problems is a class of combinatorial optimization problem, and is also NP-complete...
problem. However, along with the great progress in integer programming, especially the introduction of hybrid heuristics in MIP solver design, it is possible to solve large-size real-world school bus scheduling problems by MIP optimizer. For homogenous fleet problem, the CSCB and RSRB instances with up to 745 trips and the HOMO instances with up to 372 trips are solved optimally. For heterogeneous fleet problem, the HETERO instances with up to 286 trips can be solved optimally.

Second, the bus type-based integer programming formulation for heterogeneous fleet problem is more useful in solving real-world problems. Such formulation can reduce the model complexity in terms of the number of decision variables, the number of constraints, and the size of model file. Our benchmark test shows the optimally solvable problem size is increased from 58 trips to 286 trips. At the same time, the fixed costs for each type of buses are considered in the model formulation, which will guide the solver to select as many low priced buses as possible.

Third, we propose a general local search algorithm coupling with simulated annealing to minimize the number of buses and total travel distance. The VRP local search operators, such as one-point move, two-point move, 2-opt move and cross-exchange, are used to improve the route solution gradually. Based on this algorithm, the two-stage metaheuristic solvers for bi-objective school bus routing problem or school bus scheduling problem are designed. The test on the RSRB and CSCB instances shows that our algorithm outperforms the existing mixed load heuristic significantly. Compared to the optimal solutions obtained by CPLEX, our algorithm can also provide the optimal numbers of buses for most of the RSRB and CSCB instances. The test on HETERO instances shows that our metaheuristic approach is of obvious advantages for large-size heterogeneous fleet problem in terms of the solution quality and computation efficiency. In addition, the algorithm can be easily integrated with GIS for solving real-world school bus routing and scheduling problem.

For school bus scheduling problem, this paper and most of existing literature focus on the problems with fixed school time windows. However, the benefit of bus scheduling depends largely on the settings of school bell times. For a multi-school system, if the bell times are adjusted properly, the solution for school bus routing and scheduling problems can be improved substantially. Future works might have considered the problems of school bus scheduling and bell time adjustment simultaneously.

Supporting Information
S1 File. The benchmark instances files of school bus scheduling problem(SBSP). (RAR)
S2 File. The benchmark instances files of school bus routing problem(SBRP). (RAR)
S3 File. The single school route solution files for SBRP benchmark instances. (RAR)

Author Contributions
Conceived and designed the experiments: XPC YFK. Performed the experiments: XPC YFK LXD YEH. Analyzed the data: XPC YFK. Contributed reagents/materials/analysis tools: XPC YFK LXD YEH. Wrote the paper: XPC YFK XYY.
References

1. Corberán A, Fernández E, Laguna M, Marti R. Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of the operational research society. 2002; 53(4):427–35.
2. Park J, Kim B-I. The school bus routing problem: A review. European Journal of operational research. 2010; 202(2):311–9.
3. Bektaş T, Elmagatı S. Solving school bus routing problems through integer programming. Journal of the Operational Research Society. 2007; 58(12):1599–604.
4. Riera-Ledesma J, Salazar-González J-J. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. Computers & Operations Research. 2012; 39(2):391–404.
5. Riera-Ledesma J, Salazar-González JJ. A column generation approach for a school bus routing problem with resource constraints. Computers & Operations Research. 2013; 40(2):566–83.
6. Schittekat P, Sevaux M, Sorensen K, editors. A mathematical formulation for a school bus routing problem. Service Systems and Service Management, 2006 International Conference on; 2006 Oct. 2006. pp. 1552–7.
7. Martínez LM, Viegas JM. Design and deployment of an innovative school bus service in Lisbon. Procedia-Social and Behavioral Sciences. 2011; 20:120–30.
8. Huo L, Yan G, Fan B, Wang H, Gao W, editors. School bus routing problem based on ant colony optimization algorithm. Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo; 2014 Aug. 31 2014-Sept. 3 2014; Beijing IEEE. pp. 1–5.
9. Arias-Rojas JS, Jiménez JF, Montoya-Torres JR. Solving of school bus routing problem by ant colony optimization. Revista EIA. 2012; 9(17):193–208.
10. Spasovic L, Chien S, Kelinhofer-Fealey C, Wang Y, Hu Q, editors. A methodology for evaluating school bus routing-a case study of riverdale, New Jersey. transportation research board 80th annual meeting, Washington, DC; 2001. pp. 7–11.
11. Zhang M. The Research on Multi-objective School Bus Vehicle Routing Problems Based On Bi-level Programming. Chengdu, China: Southwest Jiaotong University; 2008.
12. Ben Sghaier S, Ben Guedria N, Mraihi R, editors. Solving School Bus Routing Problem with genetic algorithm. Advanced Logistics and Transport (ICALT), 2013 International Conference on; 2013: IEEE. pp. 7–12.
13. Minocha B, Tripathi S, editors. Solving School Bus Routing Problem Using Hybrid Genetic Algorithm: A Case Study. Proceedings of the Second International Conference on Soft Computing for Problem Solving (SoCProS 2012), December 28–30, 2012; 2014: Springer. pp. 93–103.
14. Pacheco J, Caballero R, Laguna M, Molina J. Bi-objective bus routing: an application to school buses in rural areas. Transportation Science. 2013; 47(3):397–411.
15. Ripplinger D. Rural school vehicle routing problem. Transportation Research Record: Journal of the Transportation Research Board. 2005; 1922(1):105–10.
16. Aparicio Ruiz P, Muñozuri Sanz J, Guadix Martín J. Mixed Trips in the School Bus Routing Problem with Public Subsidy. In: Cortés P, Maeso-González E, Escudero-Santana A, editors. Enhancing Synergies in a Collaborative Environment. Lecture Notes in Management and Industrial Engineering: Springer International Publishing; 2015. pp. 105–13.
17. Toth P, Vigo D. An overview of vehicle routing problems. In: Toth P, Vigo D, editors. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications. 9. Philadelphia: siam; 2002. pp. 1–26.
18. Laporte G. Fifty Years of Vehicle Routing. Transportation Science. 2009; 43(4):408–16.
19. Addas M. Analyzing the Performance of a Two-Stage Framework for the School Bus Routing and Scheduling Problem [PhD Thesis]. Halifax, Nova Scotia: Dalhousie University; 2009.
20. Braca J, Bramel J, Posner B, Simchi-Levi D. A computerized approach to the New York City school bus routing problem. IIE Transactions. 1997; 29(8):693–702.
21. Dang L, Wang Z, Kong Y, Liu Q. Heuristic Algorithm for Solving Mixed Load School Bus Routing Problem. Computer Science. 2013; 40(7):248–53.
22. Campbell J, North J, Ellegood W. Modeling Mixed Load School Bus Routing. In: Sebastian H-J, Kaminsky P, Müller T, editors. Quantitative Approaches in Logistics and Supply Chain Management. Lecture Notes in Logistics: Springer International Publishing; 2015. pp. 3–30.
23. Fügenschuh A. Solving a school bus scheduling problem with integer programming. European Journal of Operational Research. 2009; 193(3):867–84.
24. Fügenschuh A, Martin A. A multicriteria approach for optimizing bus schedules and school starting times. Annals of Operations Research. 2006; 147(1):199–216.

25. Kim B-I, Kim S, Park J. A school bus scheduling problem. European Journal of Operational Research. 2012; 218(2):577–85.

26. Bögl M, Doerner KF, Parragh SN. The school bus routing and scheduling problem with transfers. Networks. 2015; 65(2):180–203.

27. Park J, Tae H, Kim B-I. A post-improvement procedure for the mixed load school bus routing problem. European Journal of Operational Research. 2012; 217(1):204–13.

28. Repoussis PP, Tarantilis CD. Solving the Fleet Size and Mix Vehicle Routing Problem with Time Windows via Adaptive Memory Programming. Transportation Research Part C: Emerging Technologies. 2010; 18(5):695–712.

29. Baldacci R, Battarra M, Vigo D. Routing a heterogeneous fleet of vehicles. In: Golden B, Raghavan S, Wasil E, editors. The vehicle routing problem: latest advances and new challenges. operations research/computer science interfaces. New York: Springer; 2008. pp. 3–27.

30. Bowerman R, Hall B, Calamai P. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transportation Research Part A: Policy and Practice. 1995; 29(2): 107–23.

31. Bent R, Van Hentenryck P. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. Transportation Science. 2004; 38(4):515–30.

32. Bräysy O, Gendreau M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transportation science. 2005; 39(1):104–18.

33. Groër C, Golden B, WASIL E. A library of local search heuristics for the vehicle routing problem. Mathematical Programming Computation. 2010; 2(2):79–101.

34. Crama Y, Kolen AWJ, Pesch EJ. Local search in combinatorial optimization. In: Braspennin PJ, Thuijsman F, Weijters AJMM, editors. Artificial Neural Networks. Lecture Notes in Computer Science. 931: Springer Berlin Heidelberg; 1995. pp. 157–74.

35. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. science. 1983; 220(4598): 671–80. PMID: 17813860

36. Clarke G, Wright JW. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research. 1964; 12(4):568–81.

37. Subramanian A, Penna PHV, Uchoa E, Ochi LS. A hybrid algorithm for the Heterogeneous Fleet Vehicle Routing Problem. European Journal of Operational Research. 2012; 221(2):285–95.