A Metaheuristic Algorithm for Routing School Buses With Mixed Load

YAN-E HOU1,2, LANXUE DANG1,2, WEICHUAN DONG3, AND YUNFENG KONG4
1Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China
2College of Computer and Information Engineering, Henan University, Kaifeng 475004, China
3Department of Geography, Kent State University, Kent, OH 44240, USA
4Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng 475004, China
Corresponding author: Lanxue Dang (danglx@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 41801310, and in part by the Technology Development Plan Project of Henan Province, China, under Grant 202102210160.

ABSTRACT Designing a system to solve school bus routing problems (SBRP), especially in a large school district, is very complex and expensive. One of the challenges resides in designing routes for the school buses when they have mixed loads, where each bus transports students for one or more schools at the same time to save the total number of buses required. This article aims to explore whether the algorithm originally developed for pickup and delivery problem with time windows (PDPTW) can be employed for solving mixed load SBRP. We present a PDPTW-based algorithm to address the mixed load SBRP focusing on minimizing the number of buses required. Our algorithm combines a record-to-record travel framework with three neighborhood operators, single paired insertion, swapping pairs between routes, and within route insertion, to improve solution iteratively. Results from implementing this algorithm show that our PDPTW-based algorithm is feasible for mixed load SBRP. Moreover, we found that guided strategy is better than random for permuting nodes that would be selected to relocate positions in a route or among routes. In addition, the results also show that the spatiotemporal connectivity index used in our algorithm can reduce the computation time needed for searching for solutions without affecting the quality of the solutions.

INDEX TERMS School bus routing problem, mixed load, pickup and delivery problem with time windows, record to record travel.

I. INTRODUCTION

The increasing demand for school bus service has prompted more and more primary and secondary schools to make better plans for student transportation using their private buses in China. While attempting to balance safety, service quality, and operational efficiency, regulating school bus services has become a mandate and a challenge for local governments and school districts. To address this challenge, the Chinese government launched pilot programmes in six counties in 2011, encouraging local governments to explore innovative operations in providing school bus services. Five of the selected counties attempted to provide school bus services at regional level and the other county devoted its effort to developing a public transportation system that would also meet the demand. In practice, all six counties recognized that an efficient school bus service system could save cost by reducing the number of buses required and by minimizing the total distances travelled by the buses. With the increasing scale of school bus services and the complexity of constraints to the way services are provided, it is very difficult to route school buses manually while considering bus capacity, school starting time, and maximum riding time of students simultaneously.

Planning an efficient schedule for school buses is a combinatorial optimization problem in operations research. It is a well-known problem called the school bus routing problem (SBRP). It is a challenging variant of vehicle routing problems (VRP), which aim at finding an optimal set of routes for a fleet of school buses, with each bus transporting students to or from their schools while satisfying various constraints [1], [2].

For illustrating this type of routing problems, Figure 1 shows three types of school bus routes: single school route, single load route, and mixed load route. The general activities of a school bus include driving empty, picking up students, driving loaded, delivering students, and waiting. In Figure 1,
hexagons represent the depots that school buses start from and return to after a series of student stops (the triangles) or school stops (the squares). Different shading patterns represent different school destinations for the corresponding student stops and school stops.

There are two types of SBRP while regarding the number of schools involved. In a single-school SBRP, each bus serves only one school, such as the top route in Figure 1. While in a multi-school SBRP, each bus can serve more than one school. For the multi-school problem, one approach is to schedule each school bus to serve multiple schools in a certain order [3]–[5], such as the middle route in Figure 1. An alternative approach is to allow mixed load on the route, known as the mixed load SBRP [6], [7], so that at a specific stop each bus picks up all students who have their destination schools on the route and drops off them at their respective schools successively. This approach is illustrated as the bottom route in Figure 1.

Operational practices and research in this area [6], [7] have demonstrated that allowing mixed load would potentially reduce the need for vehicles in quantity and the operational cost of school bus service. In a mixed load school bus route, each student stop is considered a pickup location and each school is considered a delivery location. From this perspective, there is a similarity between the mixed load SBRP and the pickup and delivery problem with time windows (PDPTW) in that both problems involve a set of pickup locations and a set of delivery locations. In view of this, this research aims at exploring whether algorithms developed for solving PDPTW can be applied to mixed load SBRP. If this is the case, we will explore whether these algorithms can be further developed to minimize the number of buses required in a school bus service.

The main contributions of this research are described in the following. First, we model the mixed load SBRP problem as an m-1 pickup and delivery problem with time windows (PDPTW) in that both problems involve a set of pickup locations and a set of delivery locations. In view of this, this research aims at exploring whether algorithms developed for solving PDPTW can be applied to mixed load SBRP. If this is the case, we will explore whether these algorithms can be further developed to minimize the number of buses required in a school bus service.

The main contributions of this research are described in the following. First, we model the mixed load SBRP problem as an m-1 pickup and delivery problem with time windows (PDPTW), where m pickup locations correspond to one delivery location. Then we design a record-to-record travel (RRT) algorithm that is coupled with three neighborhood operators: single paired insertion (SPI), swapping pairs between routes (SBR), and within route insertion (WRI). Second, we narrow the neighborhood search space by using a spatiotemporal connectivity index. When a stop is moved in the local search, the neighborhood solutions search space is reduced by restricting the length of neighborhood list, which can reduce computation time spent on searching the neighborhood solutions. Third, a guided node permutation strategy and a lexicographic neighborhood solution evaluation function are adopted in the process of local search. The routes are sorted in ascending order by the number of stops, and the stops in the short routes have the priority to be moved. The new obtained neighborhood solution is evaluated by the evaluation function in order to guide the algorithm turn towards the direction with optimization objective as far as possible. Finally, we test and analyse the impact of the parameters of the algorithm, which including number of loops, neighborhood size, node permutation strategies and deviation coefficient and so on.

The rest of this article is organized as follows. Section II presents a brief review of existing literature. The description of the mixed load SBRP is presented in Section III. Section IV describes the design and implementation of the proposed algorithm. Numerical results from testing the effectiveness of our algorithm followed by a discussion are reported in Section V. Finally, the conclusion of our work is given in Section VI.

II. LITERATURE REVIEW

SBRP has received considerable attention in the research community of vehicle routing problems since it was introduced by [8]. A detailed review on was provided in [1], [2]. The front is mainly about SBRP literatures prior to 2010, and the latest review about SBRP from 2010 to now have provided in [2]. They all summarized the SBRP research publications from multiple perspectives, such as problem description, problem characteristics, and different solution approaches. The majority of SBRP research has focused on single school problems similar to the capacitated vehicle routing problems or their variants. These can be solved using exact or heuristic algorithms as demonstrated in [9] and [10].

Our goal here is to address mixed load SBRP by using a PDPTW-based algorithm. For this purpose, this review of relevant literature mainly focuses on the mixed load SBRP and closely related algorithms for solving PDPTW.

The first paper to discuss mixed load SBRP was in [11], pointing out that mixed loading problems can occur in rural
school districts, but they did not provide a method for generating mixed load routes. Then, a computer-assisted system was presented in [12], which could be used to generate mixed load routes, but the system still relied on some manual processes. A constructive heuristic approach based on the Location Based Heuristic [13] was developed in [6] for solving city school bus routing problem in New York. Then a two-phase heuristic algorithm that addressed mixed load SBRP with a focus on evaluating service quality by student ride times was described in [4]. Their approach started with generating an initial feasible solution and then used heuristic strategies for updating and improvement. Another heuristic algorithm proposed by [14] concerned government regulations and reimbursement policies for school bus services in a rural area in Pennsylvania. Later, the algorithm was adopted suggested by [6] to solve mixed load SBRP in Parana, Brazil [15]. In addition, the genetic algorithm also was developed for the mixed load school bus routing problem with bus stop selection sub-problem [16].

Reference [7] proposed a posterior improvement algorithm that they tested on artificial data sets and real-world instances. In addition to the algorithm, another great contribution of their study is the benchmark data sets and results that are freely downloadable to evaluate any new approaches. Many papers have used the same benchmark data sets to test new algorithms, including [17], [18]. Reference [17] presented a discrete algorithm using continuous approximation modelling, with the focus on minimizing the total travel distance. For a rural mixed load school bus routing problem, four metaheuristic algorithms were developed in [18] to seek the best routes. Because of without considering student maximum riding time constraint, the mixed load SBRP was simplified as capacitated vehicle routing problem to be solved. Further, the mixed load school bus routing problems with multi-objects or heterogeneous fleet were studied in [18], [19]. Recently, new SBRP research give emphasis to the more complex real world mixed load SBRP problems [20], [21], which addresses the contemporary trends of SBRP.

As described by [7], the mixed load SBRP can be modelled as a PDPTW problem where each student stop and the corresponding school stop are considered a pair of pickup and delivery request. For multi-vehicle static PDPTW [22] [23], the former developed a reactive tabu search method that uses SPI, SBR, and WRI operators to search for neighborhood solutions. A tabu embedded simulated annealing approach was proposed in [23] to address the same problem, where shift, exchange, and rearrange operators were used in neighborhood search. Although mixed load SBRP is different from normal PDPTW in the problem definition, such as the m-to-1 request pairs, the hard school time window, and the maximum riding time constraint, we expect local neighborhood operators for the static PDPTW to have potential to solve mixed load SBRP problem.

In this study, we developed a record-to-record travel metaheuristic to solve mixed load SBRP problem. RRT is a variant of simulated annealing (SA) algorithm. RRT use a deviation value to determine to accept a worse solution or not, which is not use probability acceptance rules. SA has been successfully applied to solve various optimization problems [24]–[26], especially for vehicle routing problem [27], [28]. Its performance and practicality prompt us to solve the mixed school bus routing problem by the variant of SA.

III. PROBLEM DESCRIPTION

Now we define notations and present a mathematical formulation of the mixed load SBRP. In the model described in [7], there is a one-to-one relationship between student stop and school stop. They proposed a method that duplicates the school into certain virtual schools during routing of school buses. Here we modelled this problem as an m-1 PDPTW problem where m pickup locations correspond to one school according to the actual situation.

Suppose a given number of students need to be transported to their schools by bus. Let \( P^+ \) represent the set of stops where a bus picks up students, and \( P^- \) represent the set of schools where buses drop off students. Every stop \( i \) in \( P^+ \) has a corresponding school in \( P^- \), noted as \( s(i) \). Let \( D \) be the set of depots that each bus starts from and returns to. For any two stops \( i \) and \( j \) there are two weights \( c_{ij} \) and \( t_{ij} \) denoting the distance and the travel time between them. \( n_i \) denotes the demand of each stop corresponding to the number of students to be picked up by the bus, while \( t_i \) denotes the service time at the node. The total number of buses is \( K \) and each bus \( k = \{1, 2, ..., K\} \) has a capacity \( Q_k \) and located at the depot \( d(k) \in D \).

In general PDPTW, each stop has an earliest and a latest possible arrival time, denoted by \( e_i \) and \( l_i \), and the vehicle must arrive at stop \( i \) between \( e_i \) and \( l_i \). Whereas in the SBRP, there are only specific time windows for schools, but no clear time windows for the student stops. However, there is a maximum riding time \( R \) for every student, so the student stop has an implied time window. Given a student stop \( m \) corresponding to the school \( i \) that has a time window \([e_i, l_i] \), we define \([e_i - R, l_i - t_{mi}] \) as the time window for the student stop. Here \( t_{mi} \) is the time cost between stop \( m \) and school \( i \). In order to satisfy the time window constraint, the bus may need to wait at specific school stops. Meanwhile, several other constraints must be honoured. For example, every student stop must be visited by a bus before arriving at its school and the number of students on any bus must not exceed the capacity of the bus at any time. Having fewer buses can significantly reduce the overall cost than otherwise. The objective of mixed load SBRP is usually to minimize the number of school buses required to serve a given number of students. All parameters and decision variables used to describe our problem are shown in TABLE 1.

With these parameters, the mixed load SBRP can be formulated as follows.

\[
\min \sum_{j \in P^+} \sum_{k \in K} x_{d(k)j}
\]

\[
\text{s.t. } \sum_{k \in K} y_{ik} = 1, \quad i \in P^+
\]
TABLE 1. Parameters and decision variables for the mixed load school bus routing problem.

| Parameters | Description |
|------------|-------------|
| $P^*$      | The set of student stops |
| $P^+$      | The set of schools |
| $P$        | $P = P^* \cup P^+$ |
| $D$        | The set of depots |
| $V$        | $V = P\setminus D$ |
| $s(i)$     | The school that students at stop $i$ will go to, $i \in P^*$, $s(i) \in P^*$ |
| $s_j$      | The relationship between student stop $i$ and school $j$. If students at stop $i$ are from school $j$, $s_j = 1$; otherwise $s_j = 0$ |
| $K$        | The set of school buses |
| $d(k)$     | The depot that bus $k$ located at, $k \in K$ |
| $Q_k$      | The capacity of bus $k$, $k \in K$ |
| $R$        | Max riding time for each student |
| $c_{ij}$   | Travel distance between stop $i$ and stop $j$, $i, j \in V$ |
| $t_{ij}$   | Travel time between stop $i$ and stop $j$, $i, j \in V$ |
| $n_i$      | The number of students at student stop $i$, $i \in P^+$ |
| $t_i$      | The service time costed at stop $i$, $i \in P$ |

Decision variables

- $x_{ik}$: If stop $j$ is served by bus $k$ immediately after stop $i$, $x_{ik} = 1$; otherwise $x_{ik} = 0$
- $y_{ij}$: If stop $i$ is served by bus $k$, $y_{ij} = 1$; otherwise $y_{ij} = 0$
- $T_k$: The time that bus $k$ arrives at stop $i$
- $L_k$: The accumulated number of students when bus $k$ leaves from stop $i$

\[
\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \quad i \in P, \quad k \in K
\] (3)

\[
\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{j(i)k} = 0, \quad i \in P^+, \quad k \in K
\] (4)

\[
\sum_{i \in P^+} x_{d(k)i} - \sum_{j \in P^-} x_{jd(k)} \leq 1, \quad k \in K
\] (5)

\[
x_{ijk}(L_{ik} + n_j - L_{jk}) = 0, \quad i \in P, \quad j \in P^+, \quad k \in K
\] (6)

\[
x_{ijk}(L_{ik} - \sum_{m \in P^+} s_{mj}y_{mk}h_m - L_{jk}) = 0, \quad i \in P, \quad j \in P^-, \quad k \in K
\] (7)

\[
n_i \leq L_{ik} \leq Q_k, \quad i \in P^+, \quad k \in K
\] (8)

\[
T_{ik} + t_i + t_{i(i)} \leq T_{d(k)}
\] (9)

\[
T_{ik} + t_i + t_{ij} \leq T_{jk} + M(1 - x_{ijk}), \quad i \in V, \quad j \in V, \quad k \in K
\] (10)

\[
T_{d(k)} - T_{ik} \leq R, \quad i \in P^+
\] (11)

\[
e_{ij} \leq T_i \leq l_i, \quad i \in V
\] (12)

\[
x_{ijk} \in \{0, 1\}, \quad i, j \in P, \quad k \in K
\] (13)

\[
y_{ij} \in \{0, 1\}, \quad i, j \in P
\] (14)

\[
T_{ik}, \quad L_{ik} \geq 0, \quad i \in P, \quad k \in K
\] (15)

In this formulation, function (1) is the objective function that focuses on minimizing the number of school buses required. Equation (2) ensures that every student stop would be visited exactly once by a school bus. Equation (3) states that the bus $k$ would leave stop $i$ after visiting stop $i$. Equation (4) ensures that the student stop and its corresponding school are visited by the same bus $k$. Equation (5) states that each bus would not start from and return to its depot more than once. Equations (6), (7) and (8) are load constraints. Equation (6) states that if bus $k$ visits student stop $i$, all of students at this stop must be picked up. Equation (7) states that if bus $k$ visits school $j$, all students from school $j$ should be dropped off. Equation (8) ensures that the number of students in bus $k$ would not exceed bus capacity $Q_k$, at any time. Equations (9), (10), (11) and (12) are time constraints. Equation (9) ensures all student stops must be visited before arriving at the corresponding schools. Equation (10) states that if stop $j$ is visited immediately after stop $i$ by the same bus $k$, the time difference between stops $i$ and $j$ should not be less than the sum of service time at stop $i$ and travel time $t_{ij}$. Equation (11) describes the maximum riding time for each student while Equation (12) states that any stop $i$ must be serviced at its time window $[e_i, l_i]$. Equations (13), (14) and (15) are constraints on decision variables.

IV. ROUTING ALGORITHM

A. NEIGHBOURHOOD SEARCH OPERATORS

Generally, most heuristic algorithms designed for route optimization are iterative processes of relocating stops in one or among several routes from an initial solution quickly constructed with a simple strategy. All possible solutions to a
specific problem form a search space from which the optimal solution can be found. Choosing a new solution at random from the entire search space and checking if it satisfies all the acceptance conditions would be time-consuming and unwise. So, neighborhood operators that guide the search process from the current solution to new solutions are needed to avoid wasting resources when conducting blind searches. Solutions that can be reached from current one by specific neighborhood operators are called neighborhood solutions of current solution. Therefore, searching for neighborhood solutions is a critical aspect in our algorithm for routing optimization.

For any proposed route changes, the constraints on routing, such as a prescribed sequence of stops, the predefined and fixed bus capacity, the maximum riding time, and the allowed time windows, must be checked at all or part of the route stops. In addition, the bus waiting times at certain stops need to be checked and adjusted properly to reflect different demands at these stops. If a newfound route is feasible, then the savings will be evaluated for subsequent operations.

In mixed load SBRP, when a student stop is relocated by such operation as insertion, swap, and rearrangement of the stops, the paired school stops must be considered simultaneously. This challenge prevents the common neighborhood search operators, such as one-point move, two-point move, k-opt move, Or-opt move, and cross-exchange, mainly designed for general VRP, to be directly applied in the mixed load SBRP algorithm. Therefore, we select three operations, SPI, SBR, and WRI, for neighborhood search, which were used successfully in a reactive tabu search algorithm to solve three benchmark data sets [22] of PDPTW.

In [22], SPI attempts to move one pair of stops \((P_i, S_j)\) from Route1 to Route2 (Figure 2). Since multiple student stops corresponds to one school in our model, which is not a one-to-one relationship, the following operations for \(S_i\) are required when moving the student stop \(P_i\) onto Route2.

1. Determine whether necessary to insert school stop \(S_i\) onto Route2. If the target route Route2 already contains the school \(S_i\), the insertion of \(S_i\) would not be necessary. In this case, the student stop \(P_i\) can be inserted into Route2 at any point before \(S_i\). If the school stop \(S_i\) is not on target route Route2, the school stop \(S_i\) can be inserted at any point. In this case, the student stop \(P_i\) can be inserted at any point before \(S_i\).

2. Determine whether necessary to remove \(S_j\) from Route1. After moving the student stop \(P_i\) to Route2, if no student from other student stops on Route1 belongs to \(S_j\), we need to remove \(S_j\) from Route1. Otherwise, we keep \(S_j\) on Route1.

The move will be accepted or rejected depending on the feasibility of itself, the costs saved, and the rules of acceptance. Of the three neighborhood operators, SPI has the greatest potential for improving the objective function and is the only operation that can reduce the number of routes and encourage removal of routes. However, it is the most time-consuming search operator used by our algorithm.

The second operator, SBR (Figure 3), swaps two pairs of student and school stops \((P_i, S_i, P_j, S_j)\) on two different routes (Route1 and Route2). Similar as on SPI, the swap operation of \(S_i\) and \(S_j\) on SBR considers not only the insertion of \(S_i\) and \(S_j\) on the new route but also the removal of \(S_i\) and \(S_j\) from the original route.

The SBR operator changes only the total travel distance of the current solution without changing the number of routes and the numbers of student stops in each route. During the neighborhood search process, the SPI search can reach a limit where no SPI moves are allowed. Such a situation is common, especially when time windows are small. SBR moves can overcome this limit and alter the search trajectory to find new areas of the feasible solution space [22].

Within the same route, the WRI operator rearranges student-and-school pairs to the best positions that reduce the total travel distance (Figure 4). WRI not only reduces the travel distance, but also explores the possibility of SPI and SBR moves.

B. NEIGHBORHOOD SOLUTION EVALUATION FUNCTION

The objective of this article is to find ways to minimize the number of school bus routes to ensure that the evaluation function designed for neighborhood search would lead the search process to solutions with fewer buses. Total travel distance is commonly used as a guiding criterion in solving VRP. However, it may lead the search to solutions with a small travel cost and makes it impossible to remove routes. Therefore, we use a lexicographic function defined in (16) [29] to evaluate the new neighborhood solutions.

In (16), \(R\) is a solution, and \(|R|\) is the number of routes in solution \(R\). \(r\) is a route, and \(|r|\) is the number of student stops
For a route solution $R$, the evaluation function consists of three parts: the number of routes $|R|$, the sum of squared number of student stops in every route $\sum_{r \in R} |r|^2$, and the total travel distance $\sum_{r \in R} c(r)$. Here we assume the current solution is $R$ and a new solution is $R_n$. Next, we introduce the procedures to compare $Cost(R)$ and $Cost(R_n)$ as follows.

First, we compare the first part of $Cost(R)$ and $Cost(R_n)$, the number of routes. If $|R_n|$ is smaller than $|R|$, it means that solution $R_n$ contains less routes than solution $R$. In this case, $R_n$ would be better and be accepted as the current solution. In addition, the neighborhood moves prefer to delete some routes if possible.

If $|R_n|$ is not smaller than $|R|$, we need to compare the second part of $Cost(R)$ and $Cost(R_n)$, $-\sum_{r \in R} |r|^2$. Maximizing $-\sum_{r \in R} |r|^2$ indicates that it encourages SPI operator to move stops from shorter routes to longer routes and guides the algorithm to progressively reduce the number of routes [29], [30]. Once the SPI operator removes all the student stops from a short route, the entire route will be deleted.

If the first two parts could not achieve any improvement on the routes, the third part of the evaluation function, $\gamma \sum_{r \in R} c(r)$, will be checked.

C. ALGORITHM FOR REDUCING THE NUMBER OF ROUTES

Combining SPI, SBR, and WRI operators with record-to-record travel (RRT) algorithm [31], [32], we design a metaheuristic algorithm for solving the mixed load SBRP problem in C++. The RRT algorithm for the TSP problem, introduced by [31], performs well in solving VRP [30], [32]. This algorithm is a special case of deterministic simulated annealing algorithm [32]. It effectively improves the route solution in iterations. We set Record and deviation for total travel distance to accept worsening solutions to avoid falling into a local optimum during neighborhood searches. The RRT algorithm (Algorithm 1) proposed for solving the mixed load SBRP is as follows.

Step (1) generates a set of feasible school bus routes as an initial solution, for which many methods can be used. For convenience, our algorithm uses a simple method to construct a set of routes with each route containing only one student-school pair. Step (3)~(14) is a top-level loop controlled by a loop count $M$. Step (4)~(13) is a second level loop for every operator in operator set, such as SPI, SBR, WRI. Step (6)~(10) is a third-level loop for each node in stop set $PList$.

The function Permutation() in step (5) changes the sequence of node searching procedures. We provide three searching strategies: sequential, random, and guided (rule1, 0 for sequential, 1 for random and 2 for guided). The node sequence in $PList$ is critical in obtaining a better solution. The fixed-sequence searching of nodes may easily result in node cycling. Therefore, most VRP algorithms employ a strategy of random perturbation of the node list. However, because a real-world mixed load SBRP may actually be a problem of a large area of neighborhood solutions, the random searching strategy may have only a low possibility of finding the best solutions in the iterative neighborhood search. Our algorithm introduces a guided rule to increase the possibility of reducing the objective. When rule1 is set to 2 for guided search, the function Permutation() sorts the pickup node list according to the number of nodes in the route, giving the nodes a higher search priority in shorter routes.

The current solution $R$ is improved iteratively in steps (6)~(10). The function of NeighbourSolution() in Step (7) is to search the neighborhood solutions by the search operators to determine the new $R^*$ by observing the acceptance rule (rule2, 0 for best acceptance, 1 for first acceptance). Best acceptance means that the function in Step (7) returns the best solution among all available neighborhood solutions that have been checked with constraints. In addition, the first acceptance means that the function will return the first available solution and will not continue the search process looking for other solutions. Here, the search operators can be SPI, WRI or SBR. In the next section, we present the algorithm for constructing search space.

Steps (8) and (9) determine whether to update the current solution $R$ to $R^*$ with Record and deviation. If the new solution $R^*$ reduces the objective value, the current solution $R$ will be replaced with $R^*$. Even if $R^*$ does not reduce the object value, the new solution will still be updated as the increased amount of the total travel distance is not greater than deviation. This enables RRT algorithm to accept slightly worse solutions to avoid being trapped by a local optimum. When all pickup nodes in a route are removed, the route will be deleted. Steps (11) and (12) update Record if needed.

The solver supports flexible parameter settings, such as the number of RRT loops ($M$), the length of neighbour list ($L$), the deviation of travel distance ($\text{deviation}$), the rule for permutation of node list, and the search acceptance rule.

D. SPATIOTEMPORAL NEIGHBORHOOD SEARCH

The neighborhood search (step (7) of algorithm 1) is a key process for finding better solutions in the proposed algorithm. Its computation complexity depends on the search space, constraints, and heuristic strategy [33]. Accordingly, reducing the size of neighborhood space is one way to save time spent in the search process. It can improve execution efficiency by defining small but effective neighborhoods [34].

For all three search operators, SPI, SBR, and WRI, the search process for new solutions involves relocating specific paired stops to a new position. All those new possible positions for inserting selected stops would form the neighborhood space of the paired stops. Some of them are identified as infeasible solutions that do not abide by constraints. In this article, we first use the spatiotemporal distance and
Algorithm 1 RouteMinimize_RRT

Input: stop list (PList), the number of loops (M), the length of neighborhood list (L), the deviation value for total travel distance (deviation), node permutation rule (rule1), and search acceptance rule (rule2).

Output: Route solution R.

1) \( R = \text{GetInitialSolution}() \);
2) \( \text{Record} = \text{Cost}(R) \);
3) for \( i = 0; i < M; i + + \) \{ 
4) for (each operator in the operator set) 
5) \( \text{Permutation}(\text{PList}, \text{rule1}); \)
6) for (each \( p \) in PList) 
7) \( R* = \text{NeighbourSolution}(R, \text{PList}, L, \text{operator}, \text{rule2}); \)
8) if \( (\text{Cost}(R*)) < (1 + \text{deviation}) \times \text{Record} \) 
9) \( R = R*; \)
10) \}
11) if \( (\text{Cost}(R* < \text{Record}) \)
12) \( \text{Record} = \text{Cost}(R*); \)
13) \}
14) 
15) \( \text{Return } R \)

spatiotemporal connectivity index to evaluate the possibility for connecting two nodes as part of one route. Next, we sort the list of positions by the connectivity index in descending order. This is because the search process can be accelerated by limiting the positions only located in the front part of the list.

In mixed load SBRP, whether two nodes can be connected to form an arc depends not only on the distance between them, but also on the temporal distance between them. To facilitate the connection between two nodes, we used the concept of spatiotemporal connectivity index, which is defined in (17).

\[
\text{Related}_{ij} = 1/(ST_{ij} + h) \tag{17}
\]

In formula (17), \( h \) is an additional factor. If all the combinations do not satisfy the constraints, then we let \( h = +\infty \). For \( ST_{ij} \) defined in (18), \( c'_{ij} \) is the normalized value of space distance \( c_{ij} \), \( d'_{ij} \) is the normalized value of temporal distance \( d_{ij} \). \( \alpha \) and \( \beta \) are weights that reflect the proportion between space and temporal distance.

\[
ST_{ij} = \alpha c'_{ij} + \beta d'_{ij} \tag{18}
\]

The temporal distance between two pickup stops in general PDPTW was defined as the minimum gap of their time windows [35]. For two student stops \( (P_i \) and \( P_j \) and their schools \( (S_i \) and \( S_j \)) in the mixed load SBRP, the pickup times at \( P_i \) and \( P_j \) are constrained by their school time windows \( e_{Si} \) \( e_{Sj} \) \( l_{Si} \) and \( l_{Sj} \). Therefore, the temporal distance between \( P_i \) and \( P_j \) relates to the time windows of \( P_i \) \( P_j \) \( S_i \) and \( S_j \). Six possible combinations of the two pairs exist, as illustrated in Figure 5. The pickup time windows of \( P_i \) and \( P_j \), can be estimated from the six scenarios. The temporal distance between \( P_i \) and \( P_j \) is the smallest gap of estimated time windows of \( P_i \) and \( P_j \).

\[
d_{ij} \text{ is simplified in formula (19). Here } e_{Si} \text{ and } e_{Sj} \text{ represent the earliest time to arrive at school } i \text{ and } j, \text{ while } l_{Si} \text{ and } l_{Sj} \text{ represent the latest time to arrive at school } i \text{ and } j.
\]

\[
d_{ij} = \begin{cases} 
    e_{Si} - l_{Si}, & e_{Si} > l_{Sj} \\
    e_{Si} - l_{Sj}, & e_{Si} > l_{Sj} \\
    0, & e_{Si} \leq l_{Si} \text{ and } l_{Sj} \geq l_{Sj} \\
    0, & e_{Si} \leq l_{Si} \text{ and } l_{Sj} \geq l_{Sj} 
\end{cases} \tag{19}
\]

Neighbours of each node can be generated by sorting other nodes in descending order based on the connectivity index. In each step of moving nodes, the most relevant nodes will be examined preferentially. Thus, the effective nodes will be accepted as early as possible, and the size of the neighborhood lists can be reduced substantially. The neighborhood solution search area can be constructed based on the spatiotemporal connectivity index.

Let the triplets \( (u, d, r) \) be a node pair being considered to be moved in the current solution, where \( u \) is a student stop, \( d \) is the school which is the destination for the student at stop \( u \), and \( r \) is the number of route in which node \( u \) and \( d \) appear. Triplet \( (i, j, k) \) denotes the new position, which \( (u, d, r) \) can be moved to. That is to say, we let node \( u \) and node \( d \) from route \( r \) to be moved to route \( k \), then inserted them after node \( i \) and node \( j \), respectively. In a case study using SPI operator, search space of \( (u, d, r) \) is \( M = \{ (i, j, k) | i \in k, j \in k, k! = r \} \). The algorithm for constructing search space of \( (u, d, r) \) is described as follows.

Algorithm 2 Construct_Search_Space

Input: \( u, d, r, \text{neighborhood_list} \)
Output: Searchspace array

1) for (int \( i = 0; i < \text{neighborhood_list.len}; i + + \) \{ 
2) \( m = \text{get Routenumber(neighborhood_list[i]);} \)
3) if (\( m == r \) ) continue;
4) \( \text{pos} = \text{route}[m].\text{firststop}; \)
5) while (\( \text{pos} != \text{NULL} \) )
6) \( \text{searchspace[index].i} = i; \)
7) \( \text{searchspace[index].j} = \text{pos}; \)
8) \( \text{searchspace[index++].k} = m; \)
9) \( \text{pos} = \text{next(pos)}; \)
10) \}
11) 

The input parameter \( \text{neighborhood_list} \) is a list of nodes related to node \( u \) that is ordered by the spatiotemporal connectivity index in descending order. Nodes with a higher spatiotemporal connectivity index value must be moved to the beginning of the list. Therefore, we can define a shorter length.
TABLE 2. Computational results from RSRB benchmark instances.

| Instance | MRT(s) | V₁₀₀ | V₅₀ | V₂₀ | V₁₀ | V₀ | V₀₀ | V₀₁₀₀ | Gap1 (%) | Gap2 (%) |
|----------|--------|------|-----|-----|-----|----|-----|------|---------|---------|
| RSRB01   | 2700   | 28   | 27  | 27  | 27  | 26 | 26  | 26   | 30       | 10.00   | 13.33   |
| RSRB02   | 2700   | 27   | 27  | 27  | 27  | 27 | 27  | 27   | 29       | 6.90    | 6.90    |
| RSRB03   | 2700   | 54   | 53  | 53  | 53  | 53 | 53  | 53   | 56       | 5.36    | 5.36    |
| RSRB04   | 2700   | 56   | 54  | 53  | 52  | 52 | 52  | 52   | 59       | 10.17   | 11.86   |
| RSRB05   | 2700   | 95   | 93  | 92  | 92  | 91 | 91  | 91   | 98       | 6.12    | 7.14    |
| RSRB06   | 2700   | 80   | 78  | 78  | 76  | 76 | 76  | 76   | 89       | 12.36   | 14.61   |
| RSRB07   | 2700   | 147  | 143 | 141 | 140 | 139| 138 | 138  | 154      | 8.44    | 10.39   |
| RSRB08   | 2700   | 154  | 148 | 144 | 142 | 141| 139 | 139  | 138      | 138   | 157    |
| RSRB01   | 5400   | 25   | 25  | 25  | 25  | 24 | 24  | 24   | 24       | 11.71   | 11.01   |
| RSRB02   | 5400   | 24   | 23  | 23  | 23  | 23 | 23  | 23   | 23       | 0.00    | 0.00    |
| RSRB03   | 5400   | 47   | 47  | 47  | 47  | 47 | 47  | 47   | 47       | 0.00    | 2.13    |
| RSRB04   | 5400   | 43   | 42  | 41  | 40  | 40 | 40  | 40   | 40       | 10.87   | 13.04   |
| RSRB05   | 5400   | 78   | 76  | 75  | 75  | 75 | 75  | 75   | 75       | 5.06    | 6.33    |
| RSRB06   | 5400   | 66   | 65  | 63  | 62  | 61 | 60  | 59   | 58       | 12.50   | 19.44   |
| RSRB07   | 5400   | 131  | 125 | 124 | 123 | 122| 122 | 122  | 121      | 7.46    | 10.45   |
| RSRB08   | 5400   | 131  | 124 | 120 | 120 | 118| 117 | 117  | 117      | 15.49   | 17.61   |
| Average  | -      | 74.1 | 71.9| 70.8| 70.3| 69.8| 69.4| 69.3  | 69.1      | 68.9    | 77.6    |

V. COMPUTATIONAL RESULTS AND PERFORMANCE ANALYSIS

A. BENCHMARK INSTANCES

To test our metaheuristic algorithm, we considered the benchmark data sets provided by [7]. The instances are classified into two groups: a random spatial distribution of schools and bus stops (RSRB) and a clustered distribution (CSCB). For a more detailed description of benchmark instances, see [7]. We choose 16 data sets to test our algorithm. The instance set has a school set and a stop set with the number of schools ranging from 6 to 100 and the number of stops ranging from 250 to 2,000.

All problem parameters, including the bus capacity, the traveling speed of school bus, and the maximum riding time, are set to be the same as those in [7]. Specifically, the bus capacity is set at 66 and the average speed is set at 20 miles per hour. The service times at pickup and delivery stops are calculated by the regression models developed by [6].

B. COMPUTATIONAL RESULTS

The values of parameters used in the algorithm are \( M = 100 \), \( L = 20\% \) of the number of student stops (for small-scale instances, \( L \) is set to 100 if 20\% of the number of student stops is smaller than 100), \( Rule1=2 \) (guided permutation of node list), \( deviation=10^{-5} \), and \( rule2=0 \) (best acceptance). The CPU times for solving the instances were recorded on a desktop computer with an Intel Core i7-3400 CPU, 2 GB RAM, and 32-bit Windows 7.

TABLE 2 and 3 show the results for 16 benchmark instances with a maximum riding time of 2,700 s and 5,400 s by applying the proposed metaheuristic algorithm.
TABLE 3. Computational results from CSCB benchmark instances.

| Instance | MRT(s) | \( V_{10} \) | \( V_{20} \) | \( V_{30} \) | \( V_{40} \) | \( V_{50} \) | \( V_{60} \) | \( V_{70} \) | \( V_{80} \) | \( V_{90} \) | \( V_{100} \) | \( V_{park} \) | Gap1 (%) | Gap2 (%) |
|----------|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|----------------|
| CSCB01   | 2700   | 28          | 27          | 27          | 27          | 26          | 26          | 26          | 26          | 26          | 30          | 10.00       | 13.33       |
| CSCB02   | 2700   | 27          | 27          | 26          | 26          | 26          | 26          | 26          | 26          | 26          | 30          | 13.33       | 13.33       |
| CSCB03   | 2700   | 52          | 49          | 49          | 49          | 49          | 49          | 49          | 49          | 49          | 55          | 10.91       | 10.91       |
| CSCB04   | 2700   | 60          | 59          | 59          | 58          | 57          | 57          | 57          | 57          | 57          | 62          | 4.84        | 8.06        |
| CSCB05   | 2700   | 106         | 102         | 101         | 101         | 100         | 100         | 100         | 100         | 100         | 116         | 12.93       | 13.79       |
| CSCB06   | 2700   | 114         | 109         | 106         | 105         | 104         | 104         | 104         | 104         | 104         | 120         | 11.67       | 13.33       |
| CSCB07   | 2700   | 167         | 163         | 162         | 159         | 159         | 158         | 157         | 157         | 156         | 156         | 7.43        | 10.86       |
| CSCB08   | 2700   | 165         | 161         | 156         | 153         | 151         | 150         | 150         | 148         | 147         | 147         | 175         | 10.86       | 16.00       |
| CSCB01   | 5400   | 23          | 23          | 23          | 23          | 23          | 23          | 23          | 23          | 23          | 23          | 4.17        | 4.17        |
| CSCB02   | 5400   | 21          | 20          | 20          | 19          | 19          | 19          | 18          | 18          | 18          | 22          | 9.09        | 18.18       |
| CSCB03   | 5400   | 41          | 40          | 40          | 39          | 39          | 39          | 39          | 39          | 39          | 41          | 2.44        | 4.88        |
| CSCB04   | 5400   | 42          | 41          | 38          | 38          | 37          | 37          | 37          | 37          | 37          | 43          | 11.63       | 13.95       |
| CSCB05   | 5400   | 92          | 88          | 88          | 88          | 86          | 86          | 86          | 86          | 86          | 85          | 9.78        | 12.37       |
| CSCB06   | 5400   | 96          | 93          | 91          | 90          | 89          | 88          | 88          | 88          | 88          | 88          | 102         | 10.78       | 13.73       |
| CSCB07   | 5400   | 133         | 130         | 125         | 123         | 122         | 120         | 119         | 119         | 119         | 118         | 133         | 6.02        | 11.28       |
| CSCB08   | 5400   | 137         | 131         | 127         | 124         | 121         | 119         | 118         | 118         | 117         | 117         | 141         | 9.93        | 17.02       |
| Average  | -       | 81.5        | 78.9        | 77.4        | 76.4        | 75.6        | 75.1        | 74.9        | 74.6        | 74.4        | 74.4        | 85.4        | 9.37        | 12.88       |

In TABLE 2 and TABLE 3, the column MRT is the students’ maximum riding time. Columns \( V_{10} \), \( V_{20} \), \ldots, \( V_{100} \) show the number of buses in the solution after 10, 20, \ldots, 100 loops of the algorithm, respectively. Column \( V_{park} \) cited the results of [7]. Column Gap1 indicates the different between \( V_{park} \) and \( V_{30} \). Column Gap2 denotes the difference between \( V_{park} \) and \( V_{100} \).

The results show that our algorithm significantly outperforms the post-improvement algorithm. After 30 loops of RRT, the average numbers of buses for RSRB and CSCB instances are 70.8 and 77.4. Compared to Park’s solutions, our algorithm reduces the total number of buses by 8.76% and 9.37% in the two sets of instances, respectively. Furthermore, by increasing the number of solver loops, better solutions could be achieved with the results improved by 11.21% and 12.88% respectively.

Unlike the post-improvement algorithm [7] where every change must have a route removed from the current solution, our metaheuristic algorithm tries to look for positions that are more possible to relocate stops among routes. This is also the main advantage of our algorithm.

C. TEST OF ITERATION NUMBERS

Our proposed algorithm executed iteratively SPI, WRI and SBR three neighborhood operators to find the solutions in the local search. The number of iterations, that is parameter \( M \), controls the execution time of the algorithm. Taking iterative \( M \) times as the termination of the algorithm, the best solution obtained was outputted. As a result, different iteration numbers can affect the quality of the algorithm. We calculate the average buses numbers of all instances, when the parameter \( M \) is assigned to a certain value in the set \{1, 5, 10, 15, \ldots, 90, 95, 100\}. The results are shown in Figure 6. The average buses numbers is decreased quickly at first, especially when \( M \) is smaller than 30. It is means that some instances may have the chance of obtaining better solution with the increasing of iterations. When \( M \) is bigger than 30, the average buses numbers is dropped slowly. From the trends of these broken lines, the number of buses is reduced extremely difficult. If increasing the number of loops, it will lead to a significant increase in execution time of the algorithm. Without considering CPU time, you can set \( M \) to a bigger value. To balance the quality and efficiency of the algorithm, we generally advise that the number of loops is set to 30.

D. TEST OF NEIGHBORHOOD SIZE

The use of SPI, SBR and WRI expands the neighborhood search space, enabling the algorithm to find better solutions. However, because of the increase of search space, the running...
time of algorithm increases. Neighborhood search consists of three fundamental steps: feasibility check, best move selection, and solution updates. For a single neighborhood search, more CPU time is required if there is a large number of feasible neighbours. Therefore, we adopt a spatiotemporal connectivity index to evaluate the probability of connecting any two nodes, by which we filter out some unnecessary moves of stops and reduce the number of feasibility tests to save the running time of the algorithm. For each node, we construct a neighborhood list in descending order based on the connectivity index values. Then, we can just try to insert selected node before or after the node at the top of its neighborhood list.

We test the influence of different neighborhood size on our proposed algorithm. As described in above, the parameter $L$ represents the length of neighborhood size. Figure 7 shows the extent to which the length of neighborhood size affects CPU time. When the length is set bigger than 20%, the average CPU time of all instances is increased quickly. However, when the length is set from 15% to 20%, the average CPU time of all instances is added smoothly. These results demonstrate the benefit of our neighborhood search strategy on reducing CPU time spent by neighborhood operators.

Figure 8 indicates that the average numbers of buses required do not increase remarkably along as the size of neighborhood decreases from 100% to 15%. Even in some cases, a solution with fewer school buses required was found with a smaller length of neighborhood list. This is because in limited iteration times, there is a chance to find better solutions if the solver always tries to connect the nodes at the top of neighborhood list. However, this does not mean that the length of neighborhood list can be reduced without limit. When the search space shrinks along with the reduction of the list, the algorithm is likely to fall into a local optimum and lose the chance of finding the global optimal solution. Therefore, we set the value of parameter $L$ to 20% in our algorithm to balance the CPU time and the quality of solutions.

Although the computation time can be significantly reduced by using the spatiotemporal neighborhood search techniques without diminishing the quality of the solution, there is still a big gap in the execution time comparing to [7]. As [17] suggested, such search time is reasonable because SBP is a static PDPTW problem and bus routes do not need frequent updates.

E. Test of Node Permutation Strategies

In addition to the size of neighborhood size, we tested three node permutation strategies discussed in Section IV, including guided, sequential, and random searches. The random searching has a chance to produce different solutions for each test, so we ran the solver 10 times for each instance. The best numbers of buses required were obtained from the sets of solutions.

Figure 9 shows the results from using different strategies on RSRB and CSCB instances with different maximum riding time constraints respectively. Shown from these figures, the guided permutation of node list can get the best solution quality for all RSRB and CSCB instances. The algorithm with random node permutation can obtain the better average number of school buses than that with sequential node permutation. It may be explained that random selection stops in local search can increase the diversity of the searching space. But for some instances, it is difficult for random approach to find equal or better solutions than guided approach with limit iterations. Our approach is a guided permutation of node list, in which the stops on the shorter route are sorted before those on the longer route. Compared to random approach, the guided search approach has the advantage of improved solution quality, because it could guide the search along potential direction.

F. Influence of Different Deviation Value

To avoid trapping local optima, our algorithm uses the parameter deviation to determine whether accept a worse neighborhood solution. To explore the setting rule of parameter deviation, we set it to 0, $10^{-1}$, $10^{-3}$, $10^{-5}$ and $10^{-7}$ five different deviation values respectively without changing other parameters of the algorithm. The test results with different deviation coefficient are shown in Figure 10. The algorithm can obtain the least average school buses number of all instances when deviation is $10^{-5}$. In this study, we set deviation to $10^{-5}$ rather than $10^{-2}$ recommended in [27], because of different benchmark dataset. The product of deviation and objective value of best solution determine whether accept a worse solution. Therefore, the parameter deviation is set to suitable value reference to the average distance of stops in benchmark dataset.
VI. CONCLUDING REMARKS

Allowing students from different schools to be picked up and transported in a school bus at the same time creates the chance to reduce the number of school bus required to serve all students in all schools. This also adds to the pressure on planning effective routes for mixed load SBRP. In this
article, we developed a record-to-record travel metaheuristic algorithm for the mixed load SBRP problem. Guided by a lexicographic neighborhood solution evaluation function, three neighborhood operators SPI, SBR and WRI originally designed for PDPTW were used for neighborhood search iteratively in deleting redundant routes and improving the current solution. Results for 16 benchmark instances show that our algorithm can significantly reduce the average number of buses required. Main findings of this research are described as follows.

First, we demonstrate that mixed load SBRP can be addressed by an algorithm that has been successfully applied to PDPTW problem. To this end, the neighborhood search is the core function of our algorithm, and the record-to-record travel approach is a framework for conducting neighborhood search. Therefore, any other heuristic algorithm based neighborhood search can be employed to replace the top-level part of our algorithm for developing new approach, such as simulated annealing, tabu search algorithm, variable neighborhood search algorithm and so on.

Second, we illustrate that spatiotemporal connectivity index can be used for reducing search space to improve the efficiency of search process. Because searching for neighborhood solutions is time consuming, we defined a spatiotemporal connectivity index for neighborhood search. We sorted neighborhood stops for every stop according to the index value, such that the algorithm will not attempt to connect the stops located at the bottom of the neighborhood list. The results show that it was useful in decreasing the CPU time without losing overall solution quality.

Third, for a large scale mixed SBRP, we enhanced the performance of our algorithm using a guided permutation strategy for neighborhood operators. Neighborhood search process needs to select a student stop for operator to relocate position repeatedly. Thus, how to select the node is an important consideration for further effective searching solutions. In theory, the random strategy may find more options to obtain better solutions. However, due to the scale of the problem, the random searching strategy provided only a low possibility of finding the best solutions. Our algorithm adopts a strategy that stops in short routes have a higher priority to be selected, and the results for benchmark data sets show that it is effective.

Four, we tried to explore the setting rule of parameters that are used in our proposed algorithm. With regard to large-scale mixed SBRP, it is necessary to set appropriate parameters to obtain better quality and performance of the algorithm. We tested the number of loops, neighborhood size and deviation constant on the algorithm to find the optimal parameters of our proposed algorithm. Through different experimental parameters, we obtain the specific impact to the quality and effective of the algorithm. It will have a certain reference for the study of similar mixed SBRP problems.

There are still some promising research areas in mixed load SBRP waiting to be explored in future works. Most of existing algorithms mainly focused on minimizing the number of school buses required, although few researchers have considered minimizing the total travel distance. Balancing different objectives to conduct multi-objective optimization is a challenge for solving this problem. In addition, several variants of the mixed load SBRP, such as the consideration of heterogeneous buses, adjustment of school bell time and transfer stations, are practical and significant research directions in the future.

REFERENCES

[1] J. Park and B.-I. Kim, “The school bus routing problem: A review,” Eur. J. Oper. Res., vol. 202, no. 2, pp. 311–319, Apr. 2010.
[2] W. A. Ellegood, S. Solomon, J. North, and J. F. Campbell, “School bus routing problem: Contemporary trends and research directions,” Omega, vol. 95, Sep. 2020, Art. no. 102056, doi: 10.1016/j.omega.2019.03.014.
[3] A. J. Swersey and W. Ballard, “Scheduling school buses,” Manage. Sci., vol. 30, no. 7, pp. 844–853, Jul. 1984.
[4] M. Spada, M. Bierlaire, and T. M. Liebling, “Decision-aiding methodology for the school bus routing and scheduling problem,” Transp. Sci., vol. 39, no. 4, pp. 477–490, Nov. 2005.
[5] B.-I. Kim, S. Kim, and J. Park, “A school bus scheduling problem,” Eur. J. Oper. Res., vol. 218, no. 2, pp. 577–585, Apr. 2012.
[6] J. Braca, J. Bramel, B. Posner, and D. Simchi-Levi, “A computerized approach to the New York city school bus routing problem,” IIE Trans., vol. 29, no. 8, pp. 693–702, Aug. 1997.
[7] J. Park, H. Tae, and B.-I. Kim, “A post-improvement procedure for the mixed load school bus routing problem,” Eur. J. Oper. Res., vol. 217, no. 1, pp. 204–213, Feb. 2012.
[8] R. M. Newton and W. H. Thomas, “Design of school bus routes by computer,” Socio-Econ. Planning Sci., vol. 3, no. 1, pp. 75–85, Jun. 1969.
[9] J. S. Arias-Rojas, J. F. Jiménez, and J. R. Montoya-Torres, “Solving of school bus routing problem by ant colony optimization,” Revista EIA, vol. 17, pp. 193–208, Jun. 2012.
[10] D. Ripplinger, “Rural school vehicle routing problem,” Transp. Res. Rec., J. Transp. Res. Board, vol. 1922, no. 1, pp. 105–110, Jan. 2005.
[11] L. D. Bodin and L. Berman, “Routing and scheduling of school buses by computer,” Transp. Sci., vol. 13, no. 2, pp. 113–129, May 1979.
[12] D.-S. Chen, H. A. Kallsen, and R. C. Snider, “School bus routing and scheduling: An expert system approach,” Comput. Ind. Eng., vol. 15, nos. 1–4, pp. 179–183, Jan. 1988.
[13] J. Bramel and D. Simchi-Levi, “A location based heuristic for general routing problems,” Location Sci., vol. 5, no. 1, p. 60, 1997.
[14] S. R. Thangiah, A. Fergany, B. Wilson, P. Anthony, and M. William, “School bus routing by ant colony optimization,” Transp. Sci., vol. 56, no. 2, pp. 320–334, Sep. 2016.
[15] F. M. S. Lima, D. S. Pereira, S. V. Conceição, and N. T. R. Nunes, “A mixed load capacitated rural school bus routing problem with heterogeneous fleet and mixed loads,” 4OR, vol. 15, no. 4, pp. 1–28, 2017.
[16] M. Kang, S.-K. Kim, J. T. Felan, H. R. Choi, and M. Cho, “Development of a genetic algorithm for the school bus routing problem,” Int. J. Softw. Eng. Appl., vol. 9, no. 5, pp. 107–126, May 2015.
[17] W. A. Ellegood, J. F. Campbell, and J. North, “Continuous approximation models for mixed load school bus routing,” Transp. Res. B, Methodol., vol. 77, pp. 182–198, Jul. 2015.
[18] F. M. S. Lima, D. S. Pereira, S. V. Conceição, and N. T. R. Nunes, “A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context,” Expert Syst. Appl., vol. 56, no. 2, pp. 320–334, Sep. 2016.
[19] F. M. S. Lima, D. S. Pereira, S. V. Conceição, and N. T. R. Nunes, “A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads,” 4OR, vol. 15, no. 4, pp. 1–28, 2017.
[20] J. Ren, W. Jin, and W. Wu, “A two-stage algorithm for school bus stop location and routing problem with walking accessibility and mixed load,” IEEE Access, vol. 7, no. 8, pp. 119519–119540, 2019.
[21] M. D. Miranda, R. S. Camargo, S. V. Conceição, M. F. Porto, and N. T. R. Nunes, “A multi-loading school bus routing problem,” Expert Syst. Appl., vol. 101, pp. 228–242, Jul. 2018.
YAN-E HOU received the B.S. degree in computer science from Henan University, Kaifeng, China, in 2003, the M.S. degree in computer science from the Shenyang Institute of Computing Technology, Chinese Academy of Science, Shenyang, China, in 2006, and the Ph.D. degree in geographical information science from Henan University, in 2016.

From 2006 to 2009, she was a Research Lecturer with the Department of Computer and Information Engineering, Henan University. From 2009 to 2015, she was a Lecturer. Since 2016, she has been an Assistant Professor. At the same time, she is also a Cooperative Researcher with the Key Laboratory of Big Data Analysis and Processing of Henan Province. She is the author of five books and more than 20 articles. Her research interests include intelligent algorithms, spatial analysis, and software engineering.

LANXUE DANG was born in Nanyang, Henan, China, in 1980. He received the M.S. degree in computer science and the Ph.D. degree in geographical information science from Henan University, China, in 2006 and 2014, respectively.

From 2009 to 2014, he was a Lecturer with the School of Computer Science and Information Engineering. Since 2015, he has been an Assistant Professor. He has also been engaged in scientific research and project development with the Henan Key Laboratory of Big Data Analysis and Processing. He has served as a Project Manager and a Technical Director for several projects. He is the author or coauthor of three books and more than 20 articles. His research interests include software engineering, operational research, and artificial intelligence.

WEICHUAN DONG received the Bachelor of Applied Mathematics degree from the Heilongjiang University of Science and Technology, Harbin, China, in 2011, and the Master of Geography degree from The State University of New York at Buffalo, Buffalo, NY, USA, in 2015. He is currently pursuing the Ph.D. degree in geography with the Department of Geography, Kent State University. His research interests include spatial analysis and social media networks.

YUNFENG KONG was born in Luoyang, Henan, China, in 1967. He received the B.Eng. degree in surveying engineering from the Guilin University of Technology, China, in 1989, the M.Eng. degree in surveying engineering from Tongji University, China, in 1992, and the Ph.D. degree in geography from The Chinese University of Hong Kong, in 2001.

Since 2004, he has been a Professor with the College of Environment and Planning, Henan University, China. He is the author of one book and more than 110 articles. His research interests include spatial analysis, geographic hypermedia, and GIS applications.

Dr. Kong was a recipient of the Creative Talent Award (Education Department of Henan Province, China), in 2004, and the Science and Technology Progress Award (Henan Province, China), in 2010.