# A Metaheuristic with Learning Mechanism for Solving the Multi-school Heterogeneous School Bus Routing Problem

Yi Xie, Yunfeng Kong, Hongzhen Xiang, Yan-e Hou, and Daojun Han

***Abstract*—This paper presents a metaheuristic with online learning (MOL) to solve the multi-school heterogeneous-fleet school bus routing problem (MHSBRP). In the iterated local search (ILS) metaheuristic framework, an online learning mechanism is integrated with the neighborhood search heuristic. It evaluates the performance of search operators according to the historical search information, and then adjusts the selection probability of the operators in the following loops of search. The proposed algorithm is tested on a set of benchmark instances. The solution results show that MOL solves both mixed-load and single-load instances of MHSBRP effectively, and outperforms the ILS significantly. Compared with the algorithm without learning mechanism, MOL could improve the convergence of neighborhood search and the quality of solution.**

***Index Terms*—Metaheuristic, online learning, multi-school heterogeneous-fleet, school bus routing problem**

## I. Introduction

OPTIMIZING the school bus route contributes to the cost reduction while ensuring the school bus service quality. It also helps alleviate traffic congestion and reduce carbon emission. The school bus routing problem (SBRP) seeks to find optimal routes for a fleet of school buses that transport students to and from their schools while satisfying various constraints [1]. The SBRP is often classified into sub-problems according to different problem attributes, such as the number of schools, service environment, allowance for mixed-loads, homogeneous or heterogeneous bus fleet, objectives of routing, and constraints on service [2].

The problem is more complicated when several schools share school buses. In some regions in China, primary and secondary schools contract with school bus companies for student delivery. The buses are of various capacities, purchasing and operating costs. The school bus services are usually shared by multiple schools to save costs. Therefore, the research on multi-school heterogeneous fleet SBRP (MHSBRP) has great application potentials in China.

In this paper, both the mixed-load and single-load problems of MHSBRP are addressed. When students from different schools are transported by the same bus simultaneously, the problem is mixed-load. Otherwise, the problem is single-load [3].

The existing approaches to solving the MHSBRP are construction heuristics and metaheuristics. Souza et al [4] propose a construction heuristic method: adaptive location based heuristic (ALBH). This approach considers both the maximum vehicle capacity and the route length. ALBH employs the greedy algorithm to construct feasible solutions. The algorithm calculates the increased cost of inserting each unvisited bus stop into the current route, and then inserts bus stops with the minimum cost increase into the route until a feasible solution is generated. Finally, the bus type is adjusted according to the number of students on each route to further reduce the total costs. Souza Lima et al [5] adopt a hybrid method to solve the heterogeneous-fleet SBRP (HFSBRP) problem and evaluate the algorithm by solving four different data sets. The results show that the heterogeneous motorcade method is suitable for rural areas with few population. Souza Lima et al [6] implement four multi-objective meta-heuristic algorithms to solve the route problem of rural school buses with heterogeneous fleets and mixed-loads with multi-objective capacity constraints. Compared with approaches in the literature, the results show that the overall performance of the algorithm embedded in the route relinking process is the best. Caceres et al [7] address the school bus route problem of special education students and develop a method combining greed heuristic and column generation method to obtain the approximate solution of the benchmark example. This algorithm not only supports the use of heterogeneous fleets, but also adopts the hybrid strategy when the stations and schools are scattered. Sales et al [8] suggest a memetic algorithm to solve the HFSBRP problem. The optimal solution from an experiment of small instances shows the algorithm's better consistency in solution quality, compared with genetic algorithm and greedy algorithm, but the algorithm takes longer time to compute. Hou et al [9] employ a greedy random adaptive algorithm for mixed set partitioning (SP) for HFSBRP with different optimization objectives under different planning scenarios. The algorithm is applicable to HFSBRP with different optimization objectives like vehicle mixing and vehicle number limitation. Dang et al [10] implement a hybrid meta heuristic algorithm combining iterative local search (ILS) with SP. The

Manuscript received March 9, 2021; revised August 1, 2021. This research has been partially supported by the National Natural Science Foundation of China Grant No. 41801310.

Y. Xie is a PhD candidate of Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Henan University; and a Lecturer of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475000, China. (e-mail: xieyi@henu.edu.cn).

Y. F. Kong is a Professor of Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Henan University, Kaifeng 475000, China. (corresponding author e-mail: 34971750@qq.com ).

H. Z. Xiang is a Master degree candidate of College of Computer and Information Engineering, Henan University, Kaifeng 475000, China. (e-mail: 1844033746@qq.com).

Y. E. Hou is an Associate professor of College of Computer and Information Engineering, Henan University, Kaifeng 475000, China. (e-mail: houyane@henu.edu.cn).

D. J. Han is a Professor of College of Computer and Information Engineering, Henan University, Kaifeng 475000, China. (e-mail: hdj@henu.edu.cn).

algorithm is tested on a benchmark and proved effective by the test results.

MHSBRP, due to its computational complexity, is usually divided into sub-problems[3, 11, 12], which are solved sequentially. Based on the bus trips for each school, Kim et al [11] propose a construction algorithm and a mixed integer programming (MIP) model to solve the single-load MHSBRP. Results from benchmark and real-world problems show that the heuristic algorithm can be used for large heterogeneous problems. Chen et al [12] suggest a new MIP model and a simulated annulling algorithm for the same single-load instances. The solution quality is improved by the exact and metaheuristic approaches. Based on a feasible single-load plan, Park et al [3] describe a post-improvement procedure for the mixed-load school bus routing problem. The algorithm is tested on seven real-world instances with heterogeneous fleet. Compared with the number of buses currently in use, the mixed-load algorithm can reduce the utilization of 22.0% buses. However, the solution quality is evaluated by the number of buses rather than the total costs. The information of bus types, capacities and purchasing costs are not described in the algorithm.

Another approach to MHSBRP is presented by modeling it as a pickup and delivery problem with time windows (PDPTW) [13]. This algorithm combines iterative local search (ILS) and variable neighborhood descent (VND) to minimize bus purchasing and operating costs. In the algorithm, three operators for PDPTW (SPI, SBR and WRI) with a strategy of bus type adjustment are used for neighborhood search. The algorithm generates the best solutions so far on a set of benchmark instances.

In the existing local-search based metaheuristics, neighborhood search operators are often executed iteratively in a fixed or random order. Usually, one algorithm could hardly perform well on all instances according to the "no free lunch" theory [14]. Fortunately, some scholars have introduced online learning to solve problems and have attained influential achievements with adaptive metaheuristic and hyper-heuristic algorithms[15, 16, 17, 18]. The basic idea of online learning is to record operator's historical information in iterations, evaluate the adaptivity of each operator with current instance, and then adjust operator's selection probability accordingly in the following iterations. The online learning mechanism is embedded in metaheuristic algorithms to solve the MHSBRP.

The key issue of online learning is to evaluate operators' optimization performance based on its historical information. Fialho et al [19] evaluate operators by extreme value based credit assignment, believing that rare but highly beneficial jumps, in most instances, are more effective than frequent but small improvements. The bigger improvement the operator offers, the higher score it gets, and vice versa. When the operators are selected by roulette method, lower score leads to lower selection probability. Soria-Alcaraz et al [20] solve the education timetabling problem by integrating the above approach with ILS. Mısır et al [21] evaluate operators' performance based on the times and degree of the improvement they make to the solutions. In the model presented by Walker et al [16], every operator's score is based on the average improvement the operator has made to objective function in the whole iteration stage. Then operators are ranked according to their performance, the operator with the best performance at the beginning, and neighborhood search is based on the rank. When evaluating operators, Cowling et al [22] take into account the operator's last call time to avoid long time no-call operators. To optimize the marshalling plan for railroad freight cars, Hirashima et al [23] employs the reinforcement learning system. The possible layouts take into account the evaluation values of the delivery distance and locomotive movement numbers, and the best plan is to move freight cars with the best evaluation. For the same problem, Yoichi et al[24] propose a different reinforcement leanring method to generate marshalling plans for railroad freight cars based on the processing time. Online learning is effective in solving route problem because it has the capability to adjust node allocation and prune nodes, thus effectively reducing hidden notes [25].

A metaheuristic with online learning (MOL) for the MHSBRP is presented in this paper. The effectiveness of the algorithm was tested on a set of benchmark instances. The solution results show that MOL is effective for solving both mixed-load and single-load instances of MHSBRP. Compared with the algorithm without learning mechanism, MOL could improve the convergence of neighborhood search and the quality of solution. In addition, the effects of online learning on benchmark instances are also discussed.

This paper is organized as follows. The next section describes the MHSBRP in detail. Section III presents the algorithm MOL for MHSBRP, including the online learning mechanism, the neighborhood search operators, and the bus type adjustment strategy. Section IV tests the algorithm on benchmark instances and analyzes the algorithm.

## II. Problem definition and mathematical formulations

The MHSBRP in this paper is defined as follows. $a$) Each school bus departs from one of the depots; $b$) every bus stop is visited by one bus and visited only once; $c$) each student is delivered to his/her school; $d$) the number of students on a bus never exceeds the capacity of the bus; $e$) the riding time of each student never exceeds the maximum riding time; $f$) all students are delivered to their schools within the schools' time window; $g$) the objective is to minimize the bus purchasing and operating costs. The parameters and decision variables in MHSBRP model are listed in Table I.

Based on the above description and the mathematical formulations in previous studies [3, 13, 26], the mixed-load MHSBRP is defined as follows.

TABLE I
PARAMETERS AND DECISION VARIABLES IN MODEL

| Parameters | Meaning |
|---|---|
| $D$ | The set of depot nodes |
| $P^+$ | The set of bus stop nodes |
| $P^-$ | The set of school nodes |
| $P$ | $P = P^+ \cup P^-$ |
| $V$ | $V = P \cup D$ |
| $q_i$ | The change of student number at node $i, i \in P$ |

| | |
|---|---|
| $st_i$ | The service time at node $i, i \in P$ |
| $s(i)$ | The school that bus stop $i$ belongs to, $i \in P^+$ |
| $e_i$ | The earliest arrival time at node $i, i \in P$ |
| $l_i$ | The latest arrival time at node $i, i \in P$ |
| $T$ | Maximum riding time |
| $M$ | The collection of bus types |
| $Q_k$ | The bus capacity of bus type $k, k \in M$ |
| $f_k$ | The fixed cost of bus type $k, k \in M$ |
| $v_k$ | The various cost of bus type $k$ |
| $d_{ij}$ | The distance from node $i$ to $j$ ,$i, j \in V$ |
| $t_{ij}$ | The travel time from node $i$ to $j$ ,$i, j \in V$ |
| Decision variables | Meaning |
| $x_{ijk}$ | If bus $k$ visits node $i$ to $j, x_{ijk} = 1$; otherwise$x_{ijk} = 0$ |
| $y_{ik}$ | If bus $k$ visits node $i, y_{ik} = 1$ ; otherwise $y_{ik} = 0$ |
| $T_{ik}$ | The arrival time of bus $k$ at node $i$ |
| $L_{ik}$ | The number of students on bus $k$ after $k$ leaves node $i$ |

$$min f(x) = \sum_{j \in P^+} \sum_{k \in M} f_k x_{0jk} + \sum_{i \in V} \sum_{j \in V} \sum_{k \in M} x_{ijk} d_{ij} v_k \tag{1}$$

$$s.t. \sum_{k \in M} y_{ik} = 1, \forall i \in P^+ \tag{2}$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \forall i \in P, k \in M \tag{3}$$

$$\sum_{j \in V} x_{jik} - \sum_{j \in V} x_{js(i)k} = 0, \forall i \in P^+, s(i) \in P^-, k \in M \tag{4}$$

$$T_{ik} + st_i + t_{is(i)} \leq T_{s(i)k}, \forall i \in P^+, s(i) \in P^-, k \in M \tag{5}$$

$$\sum_{j \in P} x_{j0k} - \sum_{j \in P} x_{0jk} = 1, \forall k \in M \tag{6}$$

$$L_{ik} = 0, \forall i \in D, k \in M \tag{7}$$

$$q_i \leq L_{ik} \leq Q_k, \forall i \in P, k \in M \tag{8}$$

$$0 \leq L_{s(i)k} \leq Q_k - q_i, \forall i \in P^+, s(i) \in P^-, k \in M \tag{9}$$

$$0 \leq L_{jk} \leq Q_k - \sum_{s(i)=j} x_{ijk} q_i, \forall i \in P^+, k \in M \tag{10}$$

$$T_{s(i)k} - T_{ik} \leq T, \forall i \in P^+, k \in M \tag{11}$$

$$e_i \leq T_i \leq l_i, \forall i \in V, k \in M \tag{12}$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in P, k \in M \tag{13}$$

$$y_{ij} \in \{0, 1\}, \forall i, j \in P \tag{14}$$

$$x_{ijk} = 0, \forall i, j \in P^+, s(i) \neq (j), k \in M \tag{15}$$

Equation (1) defines the objective function including fixed bus costs and variable operating costs. Constraint (2) ensures that every bus stop is visited by one school bus. Constraint (3) ensures that each school bus must leave node $i$ after visiting it. Constraint (4) ensures that a school bus must visit the corresponding school $s(i)$ after visiting bus stop $i$. Constraint (5) ensures that a school bus visits bus stop $i$ before visiting the corresponding school. Constraint (6) ensures that the departure and destination of every route is the depot. Constraint (7) ensures that there is no student on bus at the depot. Constraint (8) ensures that the number of students on school bus does not exceed the bus capacity. Constraint (9) shows the change of student number after a school bus visits a school. Constraint (10) shows the change of student number after a school bus visits a set of nodes, and if the node is a school and $q_i$ is negative, it represents the number of alighting students. Constraint (11) shows that students' riding time cannot exceed the maximum riding time. Constraint (12) ensures that the school bus arrives at a school in the given time window. Constraint (13) and (14) define the decision variables $x_{ijk}$ and $y_{ik}$. For the single-load MHSBRP, additional constraints must be added to the models. Mixed-loads are not allowed by (15).

## III. Metaheuristic with Online Learning

In this section, we briefly outline the metaheuristic with online learning (MOL) to solve the MHSBRP.

### *A. Algorithm Framework*

A learning mechanism is embedded in the procedures of iterative local search (ILS). In ILS, three PDPTW operators, SPI (Single Pair Insertion), SBR (Swap Pairs between Routes) and WRI (Within Route Insertion) are used [27]. The operators have been successfully used to solve the multi-school SBRP in [28] and [13]. In addition, a strategy for bus type adjustment is employed, which could reduce the route cost because of the selection of low-cost bus.

The online learning mechanism is used to guide the process of local search. In classical ILS, the neighborhood operators are implemented with a fixed or random sequence in each search iteration. This means all the operators are equally executed. In MOL, all operators are evaluated, and the operator with a higher score will have a higher selection possibility in the following iterations. The reinforcement learning could improve the performance of ILS by adaptively using neighborhood operators.

The framework of MOL is as follows. After input of the problem instance and the number of loops, a feasible solution is generated (line 1): each bus stop and its corresponding school are constructed as a route; the initial solution is constructed by greedily merging some routes. Line 4 selects the to-be-called operator. For each bus stop, line 6 uses the selected operator to search neighborhood solutions $S_{temp}$. The current solution ($S_{cur}$) and the current best solution ($S_{best}$) are updated in line 7 and line 8 respectively. Line 10 updates the score of each operator. Line 11 perturbs the best solution by a ruin and recreation method [29] in order to escape from local optimum.

In the framework of MOL, $N$ represents a neighborhood operator; $NL$, the collection of operators; $NSL$, the scoring information of operators; $NCL$, information of successively no-call operators; and $R$, the degree of perturbation.

| Framework: MOL |
|---|
| Input:*Instance, Loop* |
| 1. $S_{cur}$=Initialize(*Instance*); |
| 2. $S_{best}$=$S_{cur}$; |
| 3. While(Loop>0){ |
| 4. $N$=SelectOperator($NSL$, $NCL$); |
| 5. Foreach(stop $i$ in stops){ |
| 6. $S_{temp}$ = Neighbor ($S_{cur}$, $N$, $i$); |

```
        Scur = BestAcceptance(S_temp, S_cur);
        S_best = IsBetter(S_best, S_cur);
    }//end foreach
    EvaluateOperators(N, NSL, NCL);
    S_cur = Perturbation(S_best, R);
    Loop − −; }//end while
Output:the best solution: S_best.
```

### *B. Online Learning Mechanism*

The online learning mechanism in MOL consists of two components: the evaluation and the selection. We evaluate the operator's performance by reinforcement learning, which is an important method of machine learning [30]. To make it more environment-adaptive, an action is rewarded when its effect on the state is positive. In other words, if an operator improves the current best solution, the operator is rewarded scores. In each iteration, the operator's performance is evaluated by two parameters: the degree of improving the current best solution and the computing time. The performance of operator $N_i$ in iteration $t$ is generated by (16).

$$P_t(N_i) = \begin{cases} (\frac{I_t(N_i)}{T_t(N_i)} + 1)^2, & I_t(N_i)>0 \\ 0, otw \end{cases} \tag{16}$$

In (16), $I_t(N_i)$presents the improvement of the current best solution by operator $N_i$ in iteration $t$ ; and $T_t(N_i)$ is the computing time of operator $N_i$ in iteration $t$ . The operator's performance is evaluated only when the operator improves the current best solution.

The score of each operator is the accumulation of its historical performance. Formula (17) generates the score of operator $N_i$ in iteration $t$.

$$S_t(N_i) = \sum_{k=0}^{k<t} a^k P_{t-k}(N_i) \tag{17}$$

In (17), $a$ (ranging from 0 to 1) is the attenuation coefficient. It shows the weight of the operator's evaluation history in its score. When the value of $a$ is 0, it shows only the evaluation information of the latest iteration is taken into consideration, and value 1 shows the operator's evaluation information in each iteration has the same weight in its final score.

The selection of an operator takes into consideration the operator's score and its successive no-call times. The operator with a higher score has a greater probability of being selected. We adopt the roulette algorithm to choose an operator in current iteration. Formula (18) shows the selection probability of operator $N_i$ in iteration $t$.

$$Probability(N_i)_t = \frac{S_t(N_i)}{\sum_{j=1}^{n} S_t(N_i)} \tag{18}$$

According to (18), some operators might not be called because of their low scores. Although they cannot greatly improve the current best solution, the call of them could maintain the search diversity for the solution and benefit the other operator's performance. Therefore, we use $NCL$ to record the operator's no-call times in MOL. When the no-call times reach a threshold value, the operator will be called in the next iteration. The initial value of $NCL$ of each operator is set to the threshold value, which aims to call every operator in the beginning iterations to get the operator's initial score.

### *C. Neighborhood Search Operators*

In MOL, the route solution is gradually improved by neighborhood search. In multi-school SBRP, when a stop is moved from the route, neighborhood operators should consider its corresponding school. As a result, the widely-used neighborhood operators such as one-point move, two-point exchange and 2-opt for vehicle routing are inapplicable. We use three operators (SPI, SBR, and WRI) for PDPTW in local search to move the pairs of stop and its corresponding school [27].

The SPI on the route is shown in Fig 1. After the SPI operation, the pair (P, S) to be removed on route $r1$ is moved to the corresponding position on route $r2$.

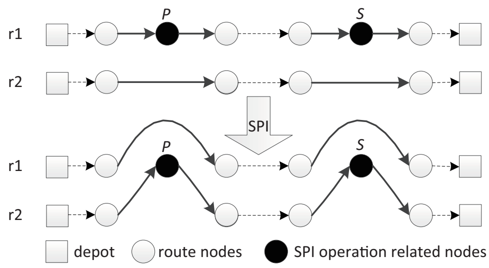


Fig. 1. An example of SPI operation

The operation of SBR is shown in Fig 2, in which the pair to be removed on route $r1$ is swapped with the pair to be removed on route $r2$.

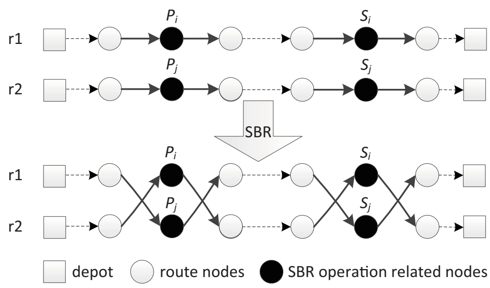


Fig. 2. An example of SBR operation

Fig 3 shows the operator's WRI. The aim of changing the position of the pairs on the same route is to reduce the total travel distance and the total cost.

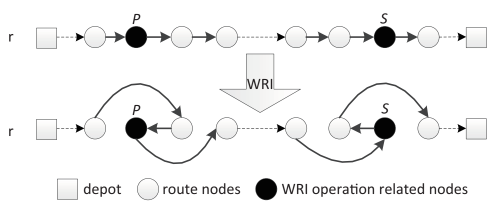


Fig. 3. An example of WRI operation

However, it requires attention when we move pair $(P, S)$ from route $r1$ to route $r2$ . If school $S$ exists on $r2$ , only bus stop $P$ is moved. However, if school $S$ serves other bus stops on $r1$ , then $S$ must be remained on $r1$ .

### *D. Bus Type Adjustment Strategy*

Appropriate buses could reduce the total costs of route solution. In MOL, we use a strategy of bus type adjustment to reduce the total costs. For neighborhood search operations, there are several chances to reduce the bus cost. $a$) If two routes can be merged, one bus will be saved. Even if a larger bus is needed for the new route, the bus cost still could be reduced significantly. $b$) When two routes are changed by SPI or SBR operator, there is a chance to use one or two new buses to reduce the total costs. However, we shall consider the route load, bus capacity, and bus operating cost when adjusting the bus type for a route. Route load represents the maximum number of students on the school bus on a route. In case that the route load is changed, the search operator will adjust the bus type according to the bus capacity.

## IV. Computational Results

The algorithm is programmed by C#, and the test uses Core i7-4790, 8GB memory, Win7 X64. We tested the multi-school SBRP instance set in [3], shown in Table II. The characteristic of the distribution of school and stop is that CSCB instances are centralized while RSRB instances are scattering. The vehicle speed, the bus stops, and the service time of school stop are the same as those in the set of instances. The distance between two nodes is the Manhattan distance. The bus type is the same as [13], including three types : A , B , and C. The bus capacity is 50, 60, and 70 respectively, and the fixed cost is 2500, 2800, and 3000 respectively. The cost is the total travel distance in miles.

TABLE II
TEST INSTANCE INFORMATION

| Instance | stops | schools | students |
|---|---|---|---|
| CSCB01 | 250 | 6 | 3907 |
| CSCB02 | 250 | 12 | 3204 |
| CSCB03 | 500 | 12 | 6813 |
| CSCB04 | 500 | 25 | 7541 |
| RSRB01 | 250 | 6 | 3409 |
| RSRB02 | 250 | 12 | 3670 |
| RSRB03 | 500 | 12 | 6794 |
| RSRB04 | 500 | 25 | 6805 |

### A. Solution Results

Here are the settings of the algorithm parameters. The max iteration is 150; the attenuation coefficient is 0.2; the threshold value of operator's no-call times is 6; ruin and recreation of 25 stop nodes each perturbation, the acceptance rules of neighborhood solution are bestmove (the best neighborhood solution will be accepted). The student's maximum riding time (MRT): 2700 seconds and 5400 seconds. The algorithm was tested in both mixed-load and single-load operational modes. Every instance was run 10 times, and the best solution (Sbest), the average solution (Savg), the standard deviations (Dev) and the average computing time (Tavg) were recorded. The results are shown in Table III.

There are several findings from Table III. First, MOL has relatively stable performance as the average standard deviation for mixed-load and single-load operational mode is 1.55% and 1.51% respectively. Second, all the instances can be solved efficiently. The average computing time for each instance ranged between 47.10 and 258.85 seconds. Third, lowering school bus service quality (changing MRT from 2700s to 5400s) could reduce the total operational cost, so the education authority could weigh between service quality and total cost in operation. Fourth, mixed-load could further reduce the total costs. Compared with single-load, the average solution with all instances in mixed-load is improved by 2.04%. Finally, compared with the single-load, the mixed-load saves 3.3% cost on average in CSCB instance and saves 0.4% cost on average in RSRB instance. The mixed-load tends to be more effective when bus stops are located relatively close together.

### B. Comparison of algorithm effectiveness

The proposed MOL is compared with ILS in [13]. Although the optimization is implemented by iteration search

TABLE III
TEST INSTANCE INFORMATION

| Instance | MRT | Mixed-load | | | | Single-load | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $S_{best}$ | $S_{avg}$ | Dev/% | $T_{avg}$/sec | $S_{best}$ | $S_{avg}$ | Dev/% | $T_{avg}$/sec |
| CSCB01 | 2700 | 75440.55 | 76786.47 | 1.27% | 49.69 | 76567.29 | 79929.91 | 2.36% | 47.10 |
| CSCB02 | 2700 | 68928.44 | 72827.90 | 2.72% | 56.00 | 73655.37 | 74395.47 | 1.37% | 52.92 |
| CSCB03 | 2700 | 138928.19 | 141921.23 | 1.24% | 183.87 | 150221.13 | 153236.48 | 1.19% | 170.81 |
| CSCB04 | 2700 | 156101.54 | 158798.80 | 1.56% | 191.48 | 157498.04 | 160922.62 | 0.98% | 191.42 |
| RSRB01 | 2700 | 74710.74 | 75837.91 | 1.43% | 40.49 | 74437.35 | 76860.43 | 2.07% | 39.08 |
| RSRB02 | 2700 | 73619.83 | 74624.97 | 1.24% | 47.10 | 73444.65 | 74328.18 | 1.09% | 44.94 |
| RSRB03 | 2700 | 141471.12 | 143815.65 | 1.29% | 128.69 | 141713.45 | 144380.15 | 1.75% | 132.41 |
| RSRB04 | 2700 | 148552.13 | 151235.20 | 1.31% | 164.60 | 152715.06 | 154659.97 | 0.98% | 155.91 |
| CSCB01 | 5400 | 65168.85 | 67862.37 | 2.19% | 81.43 | 66734.43 | 68451.80 | 1.35% | 69.06 |
| CSCB02 | 5400 | 54789.18 | 56695.23 | 1.99% | 80.23 | 57854.78 | 59060.61 | 1.02% | 69.00 |
| CSCB03 | 5400 | 109927.58 | 113109.36 | 1.62% | 258.85 | 111635.90 | 117617.29 | 1.91% | 241.10 |
| CSCB04 | 5400 | 117651.15 | 118942.13 | 0.87% | 232.45 | 118666.26 | 121085.43 | 1.71% | 225.37 |
| RSRB01 | 5400 | 67203.81 | 69496.95 | 1.93% | 51.83 | 67907.61 | 69168.56 | 1.79% | 46.07 |
| RSRB02 | 5400 | 62930.14 | 64545.78 | 2.00% | 54.63 | 62809.91 | 63645.64 | 1.64% | 48.86 |
| RSRB03 | 5400 | 131594.94 | 132524.03 | 0.64% | 170.65 | 129333.39 | 133075.53 | 1.23% | 156.17 |
| RSRB04 | 5400 | 117437.17 | 120940.33 | 1.51% | 200.64 | 119675.01 | 122680.64 | 1.78% | 182.19 |
| Average | | 100278.46 | 102497.77 | 1.55% | 124.54 | 102179.35 | 104593.67 | 1.51% | 117.02 |

with the same neighborhood operators, the operators' call methods in each iteration are different. MOL employs online learning to select operators, whereas ILS uses VND in operator selection. There are also differences in initial solution generation, bus type adjustment strategy, perturbation approaches, neighborhood size, route structure and the acceptance rules of neighborhood solutions. The two algorithms were tested with identical software and hardware environment, benchmark instances, and school bus type and cost. Table IV and Table V show the solution results from MOL and ILS with mixed-load and single-load respectively. Column $S_{best}$ and $S_{avg}$ represent the best solution and the average solution respectively; and $T_{avg}$ indicates the average computing time. Compared with the solutions from ILS, Gap1/% and Gap2/% show the cost reduction of $S_{best}$ and Savg respectively.

The results in Table IV and Table V show that the proposed MOL outperforms ILS. For the 16 mixed-load instances, MOL improves 14 best solutions and 15 average solutions. For the 16 single-load instances, MOL improves 15 best solutions and all average solutions. The costs are reduced significantly for most instances. In addition, MOL is more efficient than ILS in terms of computing time for the mixed-load instances.

### C. The effectiveness of learning mechanism

Four experiments were designed to analyze the effectiveness of learning mechanism from different perspectives.

TABLE IV
COMPARISON OF MIXED-LOAD RESULTS FROM MOL AND ILS

| Instance | MRT | MOL | | | ILS | | | Gap1/% | Gap2/% |
|---|---|---|---|---|---|---|---|---|---|
| | | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | | |
| CSCB01 | 2700 | 75440.55 | 76786.47 | 49.69 | 78206.83 | 81963.76 | 78.96 | 3.54% | 6.32% |
| CSCB02 | 2700 | 68928.44 | 72827.90 | 56.00 | 70096.54 | 74876.08 | 90.45 | 1.67% | 2.74% |
| CSCB03 | 2700 | 138928.19 | 141921.23 | 183.87 | 141227.16 | 148782.98 | 316.42 | 1.63% | 4.61% |
| CSCB04 | 2700 | 156101.54 | 158798.80 | 191.48 | 157246.92 | 163395.87 | 252.84 | 0.73% | 2.81% |
| RSRB01 | 2700 | 74710.74 | 75837.91 | 40.49 | 80059.15 | 85024.96 | 59.61 | 6.68% | 10.81% |
| RSRB02 | 2700 | 73619.83 | 74624.97 | 47.10 | 75811.45 | 79991.72 | 62.29 | 2.89% | 6.71% |
| RSRB03 | 2700 | 141471.12 | 143815.65 | 128.69 | 153119.55 | 174196.38 | 197.6 | 7.61% | 17.44% |
| RSRB04 | 2700 | 148552.13 | 151235.20 | 164.60 | 153601.61 | 161501.74 | 225.94 | 3.29% | 6.36% |
| CSCB01 | 5400 | 65168.85 | 67862.37 | 81.43 | 66911.91 | 72086.74 | 97.36 | 2.61% | 5.86% |
| CSCB02 | 5400 | 54789.18 | 56695.23 | 80.23 | 56834.79 | 60285.96 | 94.82 | 3.60% | 5.96% |
| CSCB03 | 5400 | 109927.58 | 113109.36 | 258.85 | 112928.49 | 114522.38 | 311.61 | 2.66% | 1.23% |
| CSCB04 | 5400 | 117651.15 | 118942.13 | 232.45 | 113756.89 | 121735.01 | 273.37 | -3.42% | 2.29% |
| RSRB01 | 5400 | 67203.81 | 69496.95 | 51.83 | 72290.94 | 81602.47 | 51.68 | 7.04% | 14.83% |
| RSRB02 | 5400 | 62930.14 | 64545.78 | 54.63 | 61349.26 | 63792.19 | 73.35 | -2.58% | -1.18% |
| RSRB03 | 5400 | 131594.94 | 132524.03 | 170.65 | 137510.73 | 145548.21 | 179.46 | 4.30% | 8.95% |
| RSRB04 | 5400 | 117437.17 | 120940.33 | 200.64 | 121234.66 | 128363.54 | 244.34 | 3.13% | 5.78% |
| Average | | 100278.46 | 102497.77 | 124.54 | 103261.68 | 109854.37 | 163.13 | 2.89% | 6.70% |

TABLE V
COMPARISON OF SINGLE-LOAD RESULTS FROM MOL AND ILS

| Instance | MRT | MOL | | | ILS | | | Gap1/% | Gap2/% |
|---|---|---|---|---|---|---|---|---|---|
| | | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | | |
| CSCB01 | 2700 | 76567.29 | 79929.91 | 47.10 | 80856.87 | 85673.8 | 55.19 | 5.31% | 6.70% |
| CSCB02 | 2700 | 73655.37 | 74395.47 | 52.92 | 74371.82 | 78141.87 | 61.91 | 0.96% | 4.79% |
| CSCB03 | 2700 | 150221.13 | 153236.48 | 170.81 | 152354.28 | 163223.23 | 178.63 | 1.40% | 6.12% |
| CSCB04 | 2700 | 157498.04 | 160922.62 | 191.42 | 162594.13 | 172144.24 | 173.07 | 3.13% | 6.52% |
| RSRB01 | 2700 | 74437.35 | 76860.43 | 39.08 | 81670.15 | 88332.56 | 49.53 | 8.86% | 12.99% |
| RSRB02 | 2700 | 73444.65 | 74328.18 | 44.94 | 78032.18 | 81694.04 | 51.93 | 5.88% | 9.02% |
| RSRB03 | 2700 | 141713.45 | 144380.15 | 132.41 | 151705.6 | 169092.69 | 155.82 | 6.59% | 14.61% |
| RSRB04 | 2700 | 152715.06 | 154659.97 | 155.91 | 160816.51 | 165525.1 | 166.82 | 5.04% | 6.56% |
| CSCB01 | 5400 | 66734.43 | 68451.80 | 69.06 | 68698.92 | 72590.23 | 55.61 | 2.86% | 5.70% |
| CSCB02 | 5400 | 57854.78 | 59060.61 | 69.00 | 57707.51 | 62262.367 | 58.13 | -0.26% | 5.14% |
| CSCB03 | 5400 | 111635.90 | 117617.29 | 241.10 | 119835.28 | 125401.31 | 164.34 | 6.84% | 6.21% |
| CSCB04 | 5400 | 118666.26 | 121085.43 | 225.37 | 122147.47 | 128493.44 | 163.43 | 2.85% | 5.77% |
| RSRB01 | 5400 | 67907.61 | 69168.56 | 46.07 | 72231.21 | 86489.16 | 37.92 | 5.99% | 20.03% |
| RSRB02 | 5400 | 62809.91 | 63645.64 | 48.86 | 63919.06 | 67283.8 | 47.25 | 1.74% | 5.41% |
| RSRB03 | 5400 | 129333.39 | 133075.53 | 156.17 | 134331.26 | 154683.54 | 138.93 | 3.72% | 13.97% |
| RSRB04 | 5400 | 119675.01 | 122680.64 | 182.19 | 127013.53 | 131986.66 | 147.48 | 5.78% | 7.05% |
| Average | | 102179.35 | 104593.67 | 117.02 | 106767.86 | 114563.63 | 106.62 | 4.30% | 8.70% |

*1) Comparison with RSH:* To evaluate the effectiveness of the learning mechanism, this paper compared MOL with RSH. In the iterative process, RSH adopted random selection operator, and the other parameters were set the same as MOL. Each instance was run 10 times, and the best solution $S_{best}$, average solution $S_{avg}$ and average operation time $ST_{avg}/sec$ were counted. The test results of RSH and MOL on benchmark instances were compared. Gap1 /% and gap2 /% represent the improvement of the best solution and average solution of MOL compared with RSH in each instance. The test results of mixed-load and single-load operation modes are shown in Table VI and Table VII respectively.

The results in Table VI and Table VII indicate the following: $a$) The overall performance of MOL is better than RSH. The mean value of the best solution and the mean value of the average solution of MOL are better than those of RSH, and the average improvement degree is increased by 0.93% and 0.79% respectively in the mixed-load mode, and 0.72% and 0.65% respectively in the single-load mode. Also, among the 32 test results, MOL outperforms RSH in 28 average solutions and 26 best solutions, improving the average solution and best solution by 87.50% and 81.25% respectively. $b$) The average operation time of MOL is slightly higher than that of RSH, 17.26% higher in mixed-load and 17.64% higher in single-load mode. This attributes to the increase in the calls of operators with high time complexity and the extra time cost brought about by online learning mechanism.

TABLE VI
COMPARISON OF TEST RESULTS BETWEEN MOL AND RSH IN THE MIXED-LOAD

| Instance | MRT | MOL | | | RSH | | | Gap1/% | Gap2/% |
|---|---|---|---|---|---|---|---|---|---|
| | | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | | |
| CSCB01 | 2700 | 75440.55 | 76786.47 | 49.69 | 76359.17 | 77782.93 | 39.67 | 1.22% | 1.30% |
| CSCB02 | 2700 | 68928.44 | 72827.90 | 56.00 | 71452.67 | 73577.36 | 43.92 | 3.66% | 1.03% |
| CSCB03 | 2700 | 138928.19 | 141921.23 | 183.87 | 138957.70 | 142236.05 | 152.23 | 0.02% | 0.22% |
| CSCB04 | 2700 | 156101.54 | 158798.80 | 191.48 | 157003.25 | 160718.34 | 150.25 | 0.58% | 1.21% |
| RSRB01 | 2700 | 74710.74 | 75837.91 | 40.49 | 75724.64 | 77198.95 | 30.69 | 1.36% | 1.79% |
| RSRB02 | 2700 | 73619.83 | 74624.97 | 47.10 | 73294.88 | 74670.59 | 36.32 | -0.44% | 0.06% |
| RSRB03 | 2700 | 141471.12 | 143815.65 | 128.69 | 143797.62 | 146152.61 | 102.16 | 1.64% | 1.62% |
| RSRB04 | 2700 | 148552.13 | 151235.20 | 164.60 | 150348.07 | 155164.25 | 132.40 | 1.21% | 2.60% |
| CSCB01 | 5400 | 65168.85 | 67862.37 | 81.43 | 66503.39 | 67756.74 | 71.23 | 2.05% | -0.16% |
| CSCB02 | 5400 | 54789.18 | 56695.23 | 80.23 | 54613.29 | 56272.13 | 69.66 | -0.32% | -0.75% |
| CSCB03 | 5400 | 109927.58 | 113109.36 | 258.85 | 110447.85 | 113163.33 | 231.75 | 0.47% | 0.05% |
| CSCB04 | 5400 | 117651.15 | 118942.13 | 232.45 | 118400.23 | 119119.57 | 195.93 | 0.64% | 0.15% |
| RSRB01 | 5400 | 67203.81 | 69496.95 | 51.83 | 68396.04 | 69611.07 | 41.80 | 1.77% | 0.16% |
| RSRB02 | 5400 | 62930.14 | 64545.78 | 54.63 | 63075.75 | 64100.55 | 43.03 | 0.23% | -0.69% |
| RSRB03 | 5400 | 131594.94 | 132524.03 | 170.65 | 132211.67 | 133751.08 | 138.04 | 0.47% | 0.93% |
| RSRB04 | 5400 | 117437.17 | 120940.33 | 200.64 | 118849.63 | 121663.98 | 169.78 | 1.20% | 0.60% |
| Average | | 100278.46 | 102497.77 | 124.54 | 101214.74 | 103308.72 | 103.05 | 0.93% | 0.79% |

TABLE VII
COMPARISON OF TEST RESULTS BETWEEN MOL AND RSH IN THE SINGLE-LOAD

| Instance | MRT | MOL | | | RSH | | | Gap1/% | Gap2/% |
|---|---|---|---|---|---|---|---|---|---|
| | | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | $S_{best}$ | $S_{avg}$ | $T_{avg}$/sec | | |
| CSCB01 | 2700 | 76567.29 | 79929.91 | 47.10 | 77406.31 | 80121.28 | 36.78 | 1.10% | 0.24% |
| CSCB02 | 2700 | 73655.37 | 74395.47 | 52.92 | 73626.92 | 75058.27 | 41.73 | -0.04% | 0.89% |
| CSCB03 | 2700 | 150221.13 | 153236.48 | 170.81 | 151189.01 | 153959.32 | 135.20 | 0.64% | 0.47% |
| CSCB04 | 2700 | 157498.04 | 160922.62 | 191.42 | 158721.17 | 162224.09 | 148.25 | 0.78% | 0.81% |
| RSRB01 | 2700 | 74437.35 | 76860.43 | 39.08 | 75028.37 | 76932.19 | 30.23 | 0.79% | 0.09% |
| RSRB02 | 2700 | 73444.65 | 74328.18 | 44.94 | 73276.55 | 74694.90 | 34.57 | -0.23% | 0.49% |
| RSRB03 | 2700 | 141713.45 | 144380.15 | 132.41 | 142960.50 | 145320.75 | 100.02 | 0.88% | 0.65% |
| RSRB04 | 2700 | 152715.06 | 154659.97 | 155.91 | 153216.52 | 154640.33 | 130.35 | 0.33% | -0.01% |
| CSCB01 | 5400 | 66734.43 | 68451.80 | 69.06 | 66847.65 | 68618.02 | 67.34 | 0.17% | 0.24% |
| CSCB02 | 5400 | 57854.78 | 59060.61 | 69.00 | 57033.67 | 59110.62 | 62.81 | -1.42% | 0.08% |
| CSCB03 | 5400 | 111635.90 | 117617.29 | 241.10 | 116246.00 | 119355.45 | 198.53 | 4.13% | 1.48% |
| CSCB04 | 5400 | 118666.26 | 121085.43 | 225.37 | 119184.79 | 123293.68 | 181.16 | 0.44% | 1.82% |
| RSRB01 | 5400 | 67907.61 | 69168.56 | 46.07 | 67982.06 | 69246.06 | 41.38 | 0.11% | 0.11% |
| RSRB02 | 5400 | 62809.91 | 63645.64 | 48.86 | 61640.91 | 64480.43 | 41.95 | -1.86% | 1.31% |
| RSRB03 | 5400 | 129333.39 | 133075.53 | 156.17 | 132061.75 | 133634.40 | 134.04 | 2.11% | 0.42% |
| RSRB04 | 5400 | 119675.01 | 122680.64 | 182.19 | 120214.13 | 123625.77 | 157.81 | 0.45% | 0.77% |
| Average | | 102179.35 | 104593.67 | 117.02 | 102914.77 | 105269.72 | 96.38 | 0.72% | 0.65% |

TABLE VIII
THE RUNNING INFORMATION OF THREE OPERATION IN MOL AND RSH

| Operator | MOL | | | | | RSH | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $N_c$ | $N_t$ | $N_i$ | $N_m$ | $N_{bm}$ | $N_c$ | $N_t$ | $N_i$ | $N_m$ | $N_{bm}$ |
| SPI | 87.86 | 69.56 | 23450.46 | 16702.13 | 163.90 | 49.78 | 39.76 | 19977.87 | 9875.83 | 123.47 |
| SBR | 39.61 | 39.07 | 2802.18 | 10798.39 | 75.16 | 50.04 | 47.96 | 3907.27 | 13363.81 | 100.97 |
| WRI | 22.53 | 1.07 | 2.82 | 406.59 | 0.47 | 50.18 | 2.34 | 7.11 | 899.91 | 1.23 |
| Total | 150 | 109.7 | 26255.46 | 27907.11 | 239.53 | 150 | 90.06 | 23892.25 | 24139.55 | 225.67 |

*2) Convergence analysis:* To verify the effectiveness of our algorithm, MOL was compared with ILS with random selection of operators (RSH). In each iteration of RSH, the operator was randomly selected. The solution results show that MOL not only increases the solution quality, but also increases the speed of solution convergence. Fig 4 exhibits the convergent curves of the averaged current best solution.

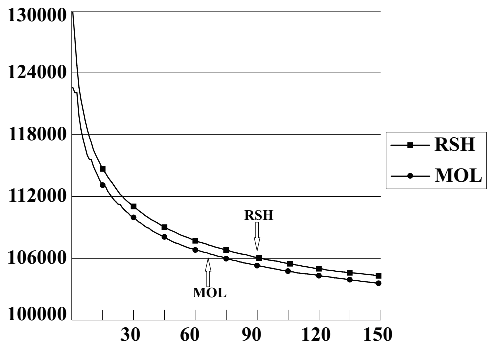


Fig. 4. Convergence curves of two algorithms

*3) Adaptability analysis of the learning mechanism:* Table VIII shows the running information of three operators in MOL and RSH: the average call times of operators ($N_c$), the average computing time ($N_t$) in seconds, the cumulative improvement of current best solution ($N_i$), the average move times ($N_m$), and the average update times of the current best solution ($N_{bm}$). Table VIII indicates that online learning significantly changes the operator's call probability. Furthermore, the accumulative improvement for current best solution in MOL is 9% higher than that in RSH. Besides, the operation times of move and bestmove in MOL increase 13.5% and 5.79% respectively than that in RSH.

Under the mixed-load and single-load operation modes, the average call times of the operator when solving different instances by MOL are shown in Fig 5 and Fig 6 respectively.

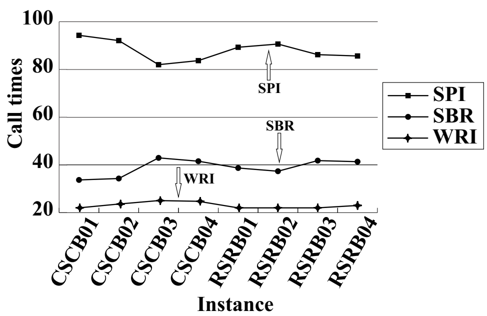


Fig. 5. Average call times of operators in mixed-load

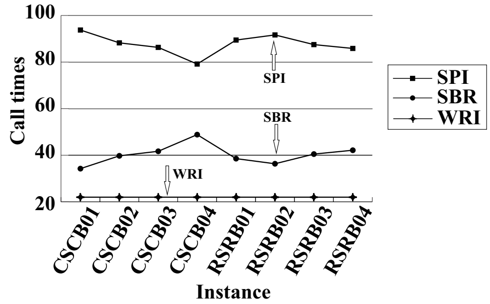


Fig. 6. Average call times of operators in single-load

As can be seen from Fig 5 and Fig 6, the number of operator calls varies in different instances. $a)$ In all instances, the average call of SPI operator is the most, followed by SBR operator, and WRI operator is the least. This is because SPI reduces vehicle numbers, and the fixed cost of vehicles accounts for a large proportion of the total costs. The SBR operator can reduce the cost by reducing the vehicle size through the exchange of pairs between routes. WRI operator adjusts the order of stations in the route to reduce some variable costs. $b)$ There are differences in average call times of the three operators in different instances, which shows the adaptability of the MOL and accounts for the stable performance of the MOL. $c)$ There is a positive correlation between the call times of SBR operator and the number of stops in the instance.

*4) Influence analysis of attenuation coefficient:* The effect of attenuation coefficient was also tested. Table IX shows the average results on all instances. Column SPI, SBR and WRI indicate the average call times of the three operators respectively. The different value of attenuation coefficient, in general, has no significant effect on the solution quality. However, the solution is better when $a$ is set to 1 than the other values.

TABLE IX
EFFECTS OF ATTENUATION COEFFICIENT ON MOL

| $a$ | $S_{best}$ | $S_{avg}$ | SPI | SBR | WRI |
|---|---|---|---|---|---|
| 0 | 101506.83 | 103578.26 | 98.58 | 29.31 | 22.11 |
| 0.2 | 101228.91 | 103545.72 | 87.86 | 39.61 | 22.53 |
| 0.4 | 101465.01 | 103650.37 | 85.51 | 41.95 | 22.54 |
| 0.6 | 101271.43 | 103673.98 | 86.17 | 41.20 | 22.64 |
| 0.8 | 101266.91 | 103611.35 | 86.82 | 40.69 | 22.49 |
| 1 | 101082.41 | 103601.07 | 86.16 | 41.27 | 22.57 |

## V. Conclusion

In this paper, an ILS metaheuristic with online learning (MOL) was proposed to solve the MHSBRP. Three operators SPI, SBR, and WRI are employed to search neighborhood solutions for PDPTW. The performance of operators was evaluated by the online mechanism in the metaheuristic framework. In the proposed algorithm, the search operators were selected according to their historical performance, which was different from the sequential or random selection of search operator in standard ILS.

The performance of the algorithm was tested on a set of benchmark instances. The solution results show that the MOL is effective and robust for solving the MHSBRP instances. The MOL also significantly outperforms the ILS in terms of solution quality and computation time. The comparison between the ILS metaheuristics with and without online learning shows that metaheuristics with online learning could not only increase the solution quality, bus also increase the speed of solution convergence.

Through experimental analysis, it is found that the MOL with online learning mechanism positively affects the solution process in three aspects. $a)$ It increases the convergence speed of MOL as the evaluation and selection of operators enables the competitive operators to be more frequently chosen. $b)$ It equips MOL with adaptability. In the experiment, the number of calls of each operator varies in different instances, and the average calls of operators are affected by the bus stop density. This shows the distribution characteristics of stops impacts the optimization ability of operators, and the learning mechanism can dynamically adjust the number of operator calls, so the MOL has self-adaptability. $c)$ It promotes the stability of MOL solution. The experimental results show that both the average solution quality and the number of optimal solutions are improved.

As for future work, we intent to test the algorithm on more MHSBRP instances and other SBRP instances. The evaluation model and selection rules are also critical issues for developing more effective algorithms.

## References

[1] R. M. Newton, and W. H. Thomas, "Design of school bus routes by computer,"Socio-Economic Planning Sciences, vol.3, no.1, pp75-85, 1969

[2] J. Park, and B.I. Kim, "The school bus routing problem: A review," European Journal of Operational Research, vol.202, no.2, pp311-319, 2010

[3] J. Park, H. Tae, and B. I. Kim, "A post-improvement procedure for the mixed load school bus routing problem," European Journal of Operational Research, vol.217, no.1, pp204-213, 2012

[4] L. V. D. Souza, and P. H. Siqueira, "Heuristic Methods Applied to the Optimization School Bus Transportation Routes: A Real Case," Lecture Notes in Computer Science, vol.6097, pp247-256, 2010

[5] F. M. Souza Lima, D. S. Pereira, S. V. Conceição, and N. T. Ramos Nunes, "A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context," Expert Systems with Applications, vol.56, pp320-334, 2016

[6] F. M. Souza Lima, D. S. D. Pereira, S. V. da Conceição, and R. S. de Camargo, "A multi-objective capacitated rural school bus routing problem with heterogeneous fleet and mixed loads," 4OR, vol.15, no.4, pp359-386, 2017

[7] H. Caceres, R. Batta, and Q. He, "Special need students school bus routing: Consideration for mixed load and heterogeneous fleet," Socio-Economic Planning Sciences, vol.65, pp10-19, 2019

[8] A. Sales, L. D. Padua, Melo, C. Sousa, D. O. E. Bonates, Tiberius, et al. "Memetic Algorithm for the Heterogeneous Fleet School Bus Routing Problem," Journal of urban planning and development, 2018

[9] Y. E. Hou, Y. F. Kong, and L. X. Dang, "Greedy Randomized Adaptive Search Procedure Algorithm Combining Set Partitioning for Heterogeneous School Bus Routing Problem," Computer Science, vol.45, no.4, pp240-246, 2018

[10] L. X. Dang, Y. E. Hou, Q. S. Liu, et al. "A Hybrid Metaheuristic Algorithm for the Bi-objective School Bus Routing Problem," IAENG Internaitonal Journal of Computer Science, vol.46, no.3, pp409-416, 2019

[11] B. I. Kim, S. Kim, and J. Park, "A school bus scheduling problem," European Journal of Operational Research, vol.218, no.2, pp577-585, 2012

[12] X. Chen, Y. Kong, L. Dang, Y. Hou, and X. Ye, "Exact and Metaheuristic Approaches for a Bi-Objective School Bus Scheduling Problem," Plos One, vol.10, no.4, 2016

[13] Y. E. Hou, Y. F. Kong, L. X. Dang, and Y. J. Wang, "Metaheuristic Algorithm for Solving Multi-school Heterogeneous School Bus Routing Problem," Computer Science, vol.44, no.8, pp216-224, 2017

[14] D. H. Wolpert, and W. G. Macready, "No Free Lunch Theorems for Optimization," IEEE Transactions on Evolutionary Computation, vol.1, pp67-82, 1997

[15] Y. Xie, Y. E. Hou, X. P. Chen, and Y. F. Kong, "Review of research progress of hyper-heuristic algorithms," Computer Engineering and Applications, vol.53, no.14, pp1-8, 2017

[16] J. D. Walker, G. Ochoa, M. Gendreau, and E. K. Burke, "Vehicle Routing and Adaptive Iterated Local Search within the HyFlex Hyper-heuristic Framework," Lecture Notes in Computer Science, vol.7219, pp265-276, 2012

[17] M. I. Gualtieri, P. Pietramala, and F. Rossi, "Heuristic Algorithms For Scheduling Jobs On Identical Parallel Machines Via Measures Of Spread," IAENG International Journal of Applied Mathematics, vol.39, no.2, pp100-107, 2009

[18] G. B. Hima Bindu, K. Ramani, and C. Shoba Bindu, "Optimized Resource Scheduling using the Meta Heuristic Algorithm in Cloud Computing," IAENG International Journal of Computer Science, vol.47, no.3, pp360-366, 2020

[19] Á. Fialho, L. D. Costa, and M. Schoenauer, "Extreme Value Based Adaptive Operator Selection," International Conference on Parallel Problem Solving From Nature: PPSN X. pp175-184, 2008

[20] J. A. Soria-Alcaraz, G. Ochoa, J. Swan, M. Carpio, H. Puga, and E. K. Burke, "Effective learning hyper-heuristics for the course timetabling problem," European Journal of Operational Research, vol.238, no.1, pp77–86, 2014

[21] M. Mısır, K. Verbeeck, P. D. Causmaecker, and G. V. Berghe, "An Intelligent Hyper-Heuristic Framework for CHeSC 2011," Springer Berlin Heidelberg, 2012

[22] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," International Conference on the Practice and Theory of Automated Timetabling, Springer, Berlin, Heidelberg, 2000

[23] Y. Hiroshima, "A Reinforcement Learning Method for Train Marshaling Based on Movements of Locomotive," IAENG International Journal of Computer Science, vol.38, no.3, pp242-248, 2011

[24] H. Yoichi, "On Reinforcement Learning Methods for Generating Train Marshaling Plan Considering Group Layout of Freight Cars," IAENG International Journal of Computer Science, vol.39, no.3, pp239-245, 2012

[25] M. J. Er, L. Y. Zhai, X. Li, et al, "A Hybrid Online Sequential Extreme Learning Machine with Simplified Hidden Network," IAENG International Journal of Computer Science, vol.39, no.1, pp1-9, 2012

[26] L. X. Dang, "Optimization algorithms for large scale mixed load school bus routing problem," Ph.D. dissertation, Henan University, 2014

[27] W. P. Nanry, and J. W. Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search," Transportation Research Part B Methodological, vol.34, no.2, pp107-121, 2000

[28] L. X. Dang, "Heuristic Algorithm for Solving Mixed Load School Bus Routing Problem," Computer Science, vol.40, no.7, pp248-253, 2013

[29] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, "Record Breaking Optimization Results Using the Ruin and Recreate Principle," Journal of Computational Physics, vol.159, no.2, pp139–171, 2000

[30] Sutton, S. Richard, Barto, and G. Andrew, "Introduction to Reinforcement Learning," Machine Learning, vol.16, no.1, pp285-286, 2005