Hindawi
Complexity
Volume 2020, Article ID 9291434, 24 pages
https://doi.org/10.1155/2020/9291434

Research Article

Evolutionary Hyperheuristics for Location-Routing Problem with Simultaneous Pickup and Delivery

Yanwei Zhao,[1] **Longlong Leng**,[1] **Jingling Zhang**,[1] **Chunmiao Zhang**,[1] **and Wanliang Wang**[2]

[1]*Key Laboratory of Special Equipment Manufacturing and Advanced Processing Technology, Ministry of Education, Zhejiang University of Technology, Hangzhou 310023, China*
[2]*College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, China*

Correspondence should be addressed to Yanwei Zhao; ywz@zjut.edu.cn

Received 14 June 2019; Revised 11 January 2020; Accepted 3 February 2020; Published 28 February 2020

Guest Editor: Francisco G. Montoya

This paper presents an evolution-based hyperheuristic (EHH) for addressing the capacitated location-routing problem (CLRP) and one of its more practicable variants, namely, CLRP with simultaneous pickup and delivery (CLRPSPD), which are significant and NP-hard model in the complex logistics system. The proposed approaches manage a pool of low-level heuristics (LLH), implementing a set of simple, cheap, and knowledge-poor operators such as "shift" and "swap" to guide the search. Quantum (QS), ant (AS), and particle-inspired (PS) high-level learning strategies (HLH) are developed as evolutionary selection strategies (ESs) to improve the performance of the hyperheuristic framework. Meanwhile, random permutation (RP), tabu search (TS), and fitness rate rank-based multiarmed bandit (FRR-MAB) are also introduced as baselines for comparisons. We evaluated pairings of nine different selection strategies and four acceptance mechanisms and monitored the performance of the first four outstanding pairs in 36 pairs by solving three sets of benchmark instances from the literature. Experimental results show that the proposed approaches outperform most fine-tuned bespoke state-of-the-art approaches in the literature, and PS-AM and AS-AM perform better when compared to the rest of the pairs in terms of obtaining a good trade-off of solution quality and computing time.

1. Introduction

As one of the complex systems, logistics network design brings plenty of benefits in terms of economy, society, and environment. Recent years have witnessed great success in applying various tools for optimizing and managing complex logistics system to drive profit and service quality of freight transport [1]. One of such tools is location-routing problem (LRP) [2], integrating two types of decisions: facility location problem and vehicle routing problem (VRP) [3]. Among various extensions to the basic LRP, CLRP and CLRPSPD have been proposed in order to represent different features of practical problems in the complex logistics systems. Meanwhile, an effective solution method should be developed for providing competitive solutions for benchmark instances within reasonable computing time for both CLRP and CLRPSPD. Hence, this paper presents a novel approach to solve both problems.

Both CLRP and CLRPSPD are NP-hard, and they are more complicated and time-consuming to solve, especially CLRPSPD. Therefore, it is unlikely to achieve the proven optimality for large instances within reasonable computing time [4]. Our previous work [2] applied a novel framework of hyperheuristic to solve CLRPSPD using tabu search, FRR-MAB, and combination of both as selection strategies and five acceptance criteria. However, the main difference between this paper and our previous work can be obtained: (1) the high-level strategies were inspired from evolutionary algorithms instead of the above strategies; (2) four basic acceptance criteria were developed in this paper which were not used by our previous work; (3) this paper aims at solving the benchmark instances of the CLRP and CLRPSPD, but our previous work only tackled the latter; (4) the pool of operators in our previous work was classified under the premise of nature of operators which was given in advance, but in this paper, the nature of operators was unknown.

Moreover, the difference between this paper and Ref. [4] can be drawn that our paper focuses on a different solution method for both CLRP and CLRPSPD, namely, hyperheuristic approach. Therefore, the main contributions are as follows:

(i) Aiming at solving two basic models in complex logistics system, we explore a generality-oriented and emerging heuristic for solving both subjects in complex logistics network, namely, evolutionary hyperheuristic-based iterated local search.

(ii) Inspired from the process of metaheuristics, several ESs are developed for implementing an online learning strategy for adaptively selecting promising sequence of LLHs to attempt to realize the global search, that is, QS, AS, and PS. Other most recent and relevant nonevolutionary selection strategies (RP, TS, and FRR-MAB) [5–7] are also chosen as the baseline for comparison.

(iii) Four basic acceptance criteria are used to pair up with ES and non-ES to examine their performance, that is, All Moves (AM), Naïve Acceptance (NA), Great Deluge (GD), and Simulated Annealing (SA).

The remainder of this study is organized as follows. Section 2 provides a brief review about the approaches for problem domains and hyperheuristic over recent decades. In Section 3, the three-index MIP formulation for CLRPSPD is defined. The proposed evolution-based selection strategies are described in Section 4, and evaluation and discussion are presented in Section 5. Finally, conclusions are given in Section 6.

2. Literature Review

Two types of reviews are concerned in this section: one for problem domains (CLRP/LRPSPD) and one for approaches (i.e., hyperheuristic and HH). The former mainly focuses on effective approaches for solving CLRP and CLRPSPD, while the latter reviews recent technologies in developing HH and applications of HH in real world.

2.1. Location-Routing Problem. In the complex logistics, CLRPSPD is a more practical variant version of CLRP by considering reverse logistics, that is, goods need to be picked up from clients when vehicles deliver goods to clients. Firstly, we review some recent approaches for CLRP, and then recent state of CLRPSPD is investigated.

As the main component in CLRP, VRP has been studied for decades. In the last years, several researchers have devoted to the development of VRP. Ant colony algorithm was applied to solve network intensive vehicle service [8]. Multiple population-based genetic algorithm was utilized to solve train-set circulation plan problem [9]. A hybrid tabu search algorithm was developed for a real-world problem [10], that is, open VRP considering fuel consumption. Meanwhile, several approaches were developed for solving a variant of VRP [11], namely, VRP with time windows. However, the above papers did not analyse the effects of depots on the logistics networks.

Among the variants of LRP, the CLRP has recently emerged as one of the most addressed, initiated in Ref. [12], applied in many practical applications, such as, glass recycling [13], food distribution [14], obnoxious waste [15], disaster relief [16], and so on. Several surveys on CLRP are provided in Refs. [17–21]. The above surveys proposed classification schemes in aspects of either structural characteristics or solution approaches and also summarized variants and extensions of CLRP, but summary on the best approaches for solving CLRP was not provided.

Considering the tailored solution methods of solving the CLRP, several exact methods have been devoted to solving CLRP, such as, branch-and-cut-and-price [22], branch-and-cut [23, 24], and dynamic programming [25], which are the most effective exact methods in tackling CLRP. However, no one is able to achieve proven optimality to large-scale instances within reasonable CPU time. The most effective heuristics have been suggested to tackle large instances and mentioned hereafter.

A two-phase metaheuristic method with TS architecture was proposed by decomposing CLRP into two subproblems [26]. An iterative two-phase approach was applied in Ref. [27]. In their algorithm, clients were clustered into supercustomers, then the Lagrangian relaxation method was used in the location phase, and granular TS (GTS) constraints were suggested to improve the routes in the second phase. A clustering analysis was presented to generate the routes data, and then depot location is solved with the collapsed routes [28]. In the method proposed in Ref. [29], a neural network combining with a self-organizing map and TS was applied to solve single-depot LRP. An iterative heuristic was presented for LRP on the plane by iteratively processing feedback information of two decision problems [30].

More recently, a hybridized GRASP was developed [31], and an evolutionary local search (ELS) procedure was used to search within two solution spaces. SA heuristic based on three random neighbourhoods was studied [32]. An adaptive large neighbourhood search heuristic by combining hierarchical structure and several operators was developed [33]. Two-phase hybrid heuristic was presented by constructing two phases: construction phase and improvement phase [34]. The method was developed using a hierarchical ant colony structure in multiple ant colony optimization algorithm [35]. GTS with a variable neighbourhood search algorithm was for solving CLRP [36]. A unique genetic algorithm was proposed for CLRP [37], using local search procedures in the mutation phase without changing the standard GA framework. Besides, the aforementioned exact and heuristic approaches have been applied successfully to CLRP by many researchers, and most require efficient constructive methods for obtaining the initial population.

Little attention has been received in developing the CLRPSPD. This subject was first addressed in Ref. [38]. In their paper, an exact algorithm named branch-and-cut method combined SA (BC-SA) was studied. Next year, the same authors developed two types of formulas for CLRPSPD: node- and flow-based formulas [39].

Multistart SA was proposed to incorporate multistart hill climbing strategies into SA framework [40]. Two-echelon CLRPSPD has been addressed to tackle the instances with less than 50 clients [41]. However, no instances with more than 100 clients have been solved to prove optimality by the above four approaches. The SA employing three local neighbourhood search mechanisms was developed to solve few instances with more than 100 clients [4]. The aforementioned methods have devoted to the development of CLRPSPD and achieved significant success in obtaining high solution quality at the expense of significantly high computing times. Our previous work [2] could outperform the above papers in terms of computing times and solution quality by developing a general framework of hyperheuristics using iterated local search procedures.

2.2. Hyperheuristic. Conclusion derived from the aforementioned approaches for CLRP and CLRPSPD can be drawn that different search operators are involved in those methods, e.g., hill climbers, mutation heuristics, and crossovers, which may be effective in finding global optima. However, it is hard for them to obtain trade-off between exploration (exploring new solutions) and exploitation (exploiting better solutions) rooting in that no corresponding strategies are suggested to efficiently evaluate and manage those operators at right time, showing poor performance concerning computing times. Moreover, existing search-based approaches are generally domain-dependent, resulting in a hard task for tester without a deep knowledge in domain. The ideal of hyperheuristic was defined as “heuristics to choose heuristics” [5, 42]. Posteriorly, an extensive version was developed as a methodology and classified into two types: heuristic selection and heuristic generation (heuristics to generate heuristics) [43]. This paper focuses on the former based on single-point-search method, and brief description and review is provided hereafter.

In the framework of selection HH, two levels are concerned: HLH and LLH. The HLH manipulates the space consisting of a fixed pool of LLHs which directly modify the space of solutions [44]. Two main categories can be considered in HLH: selection strategies and acceptance criteria [45]. The role of heuristic selection mechanism is to intelligently construct effective sequences of heuristics from the pool of LLHs, while acceptance criterion aims at deciding whether to accept or reject new solution after applying the chosen LLH [46]. By analysing the source of feedback information, three modules can be considered: online, offline, and no-learning. Choice function [5, 47], reinforcement learning [48], TS [2, 6, 49–52], and FRR-MAB [2, 53–57] are examples for online selection strategies, and simple random, random descent, RP, etc., are viewed as no-learning methods. Several metaheuristic-based strategies for designing hyperheuristics have been proposed in the literature: ant colony-, particle-, and quantum-inspired hyperheuristic. Characteristics of most evolutionary hyperheuristics in the literature are presented in Table 1, where for each EHH category, application domain, source of publication, main feature, and publishing year are provided. As far as acceptance criterion is concerned, two types are involved: determinate and nondeterminate methods. The former determinately accepts the resultant result, such as all moves, only improving, improving, and equal while NA, SA, GD, and Monte Carlo are instanced as the nondeterminate methods for accepting the new solutions.

With the popularity of hyperheuristic, it has been widely applied in practice, such as 2D regular and irregular packing problems [73], nurse rostering [74], vehicle routing problem [54], construction levelling problem [52], software project scheduling problem [56], t-ways test suite generation [6, 75], deriving products for variability test of feature models [55], fast machine reassignment [76], and timetabling [77–81]. We refer interested readers to these papers [43, 82, 83] for extensive review on hyperheuristic.

To the best of our knowledge, evolutionary hyperheuristics have not been used thus far to address the basic models of complex logistics network, i.e., CLRP and CLRPSPD.

3. Mathematical Formulation

CLRPSPD is a more practical variant of CLRP considering the reverse logistics in which the pickup and delivery take place at the same time for each client [4]. The CLRPSPD and CLRP are defined on a complete directed graph $G = (V, E)$, where $V = J \cup I$ is a set of nodes in which J and I represent the potential depot and client nodes, respectively, and $E = \{(i, j): i, j \in V, i \neq j\} \backslash \{(i, j): i, j \in J\}$ is the set of arcs. Each arc $(i, j) \in E$ has a nonnegative distance c_{ij}. Each client $i \in I$ has a positive delivery demand q_i and pickup demand p_i (only for the CLRPSPD). A storage capacity w_j and a fixed opening cost f_j are associated with each potential depot $j \in J$. The index set of vehicle types is denoted by K consisting of homogeneous vehicles with capacity Q. The objective is to determine the optimal plan of the set of depots and routes to minimize the total costs consisting of opened depot, vehicle, and travel costs.

Two types of MIP formulations for CLRPSPD are defined in Refs. [38, 39]: node- and flow-based formulations, whose distinction depends on the definition of additional variables in the graph: nodes or arcs, and similarity of the above two is built on two-index concept. In this paper, three-index flow-based MIP formulation is defined for CLRPSPD. Before describing this formulation, some assumptions should be considered:

(i) Each route is served by one vehicle, and each client is served only once.

(ii) Each vehicle must return to the departure depot at the end of the route.

(iii) Goods of each client are delivered and picked up at the same location, and the goods are delivered to each client before being picked up.

(iv) The total vehicle load at any arc must not exceed the vehicle capacity.

Table 1: Ant-, particle-, and quantum-based hyperheuristics.

Evolutionary type	Application domain	Publication	Main feature
Ant-based HH	Scheduling	[60]	Population/paired
	Project presentation scheduling	[61]	Population/paired
	2D bin packing	[62]	Population/paired
	Travelling tournament	[63]	Population/paired
	P-median	[64]	Population/paired
	WDM networks	[65]	Population & single-point/paired
	WDM networks	[66]	Population/paired
	Dynamic environment	[67]	Single-point/paired
	Travelling salesman	[68]	Population/paired
	Flow shop scheduling	[69]	Population/single
Particle-based HH	Academic scheduling	[70]	Population/multi-heuristics
	Examinations scheduling	[71]	Population/multi-heuristics
	Project scheduling	[72]	Population/multi-heuristics
	Grid resource scheduling	[73]	Population/single-heuristic
Quantum-based HH	Energy-aware scheduling	[74]	Population/single-heuristic

(v) The total delivery and pickup demands of clients served by each depot cannot exceed the depot capacity.

Decision variables:

$$x_{ijk} = \begin{cases} 1, & \text{if arc } (i,\ j) \text{ is operated by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{if client } i \text{ is served by depot } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if depot } i \text{ is selected to operate} \\ 0, & \text{otherwise} \end{cases}$$

Additional variables:

U_{ijk}: the dynamic load of each arc $(i,\ j)$ operated by vehicle k.

Based on the aforementioned assumptions and notations, the proposed flow-based formulation [2] is as follows:

$$\min \text{Fitness} = \sum_{j\in J} f_j y_j + \sum_{i\in J}\sum_{j\in I}\sum_{k\in K} f_k x_{ijk} + \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij} x_{ijk}, \tag{1}$$

subject to

$$\sum_{i\in V}\sum_{k\in K} x_{ijk} = 1,\ \forall j \in I, \tag{2}$$

$$\sum_{i\in J}\sum_{j\in J} x_{ijk} = 0,\ \forall k \in K, \tag{3}$$

$$\sum_{i\in V} x_{ijk} = \sum_{i\in V} x_{jik},\ \forall j \in V, \forall k \in K, \tag{4}$$

$$\sum_{i\in J}\sum_{j\in I} x_{ijk} \le 1,\ \forall k \in K, \tag{5}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \le |S| - 1,\ \forall k \in K, \tag{6}$$

$$\sum_{j\in J} z_{ij} = 1,\ \forall i \in I, \tag{7}$$

$$\sum_{j\in I} x_{ijk} + \sum_{j\in I} x_{jgk} \le 1,\ \forall i,\ g \in J,\ \forall k \in K,\ i \ne g, \tag{8}$$

$$\sum_{i\in I} z_{ij} \ge y_j,\ \forall j \in J, \tag{9}$$

$$z_{ij} \le y_j,\ \forall i \in I,\ \forall j \in J, \tag{10}$$

$$\sum_{j\in I}\sum_{k\in K} x_{ijk} \ge y_i,\ \forall i \in J, \tag{11}$$

$$\sum_{j\in I} x_{ijk} \le y_i,\ \forall i \in J,\ \forall k \in K, \tag{12}$$

$$x_{ijk} + z_{ig} + \sum_{m\in J, g\ne m} z_{jm} \le 2,\ \forall i,\ j \in I,\ \forall k \in K,\ \forall g \in J, \tag{13}$$

$$\max\left\{\sum_{i\in I} z_{ij} \times q_i, \sum_{i\in I} z_{ij} \times p_i\right\} \le w_j \times y_j,\ \forall j \in J, \tag{14}$$

$$\sum_{i\in J}\sum_{j\in I} U_{ijk} = \sum_{i\in V}\sum_{j\in I} x_{ijk} \times q_j,\ \forall k \in K, \tag{15}$$

$$\sum_{i\in I}\sum_{j\in J} U_{ijk} = \sum_{i\in V}\sum_{j\in J} x_{ijk} \times p_j,\ \forall k \in K, \tag{16}$$

$$\sum_{k\in K}\sum_{i\in V} U_{ijk} - q_j = \sum_{k\in K}\sum_{i\in V} U_{jik} - p_j,\ \forall j \in I, \tag{17}$$

$$0 \le U_{ijk} \le Q \times x_{ijk},\ \forall i,\ j \in V,\ \forall k \in K. \tag{18}$$

In this three-index flow-based MIP formulation, objective function (1) represents the fitness value *Fitness* consisting of the total costs composed of fixed lease cost of depots and vehicles and travelling cost of edges; constraint (2) guarantees that each client is served only once; constraint (3) eliminates routes between depots; constraint (4) ensures that entering and leaving edge to each node is equal; constraint (5) requires that only one route is assigned to one vehicle and the vehicle only departs from one depot; subcircuits in each route are forbidden by constraint (6); constraint (7) eliminates the situation that one client is served by plural different depots; constraint (8) specifies that plural different depots must not exist in the same route; constraints (9) and (10) enforce that clients are assigned to the selected depots and each depot to open must serve at least one client; each vehicle is assigned to the chosen depots and each selected depot must have at least one vehicle, which is ensured by constraints (11) and (12); constraint (13) ensures two consecutive clients of each route are served by the same depot; constraint (14) forbids that total delivery and pickup demands of clients must not exceed the depot's capacity; constraint (15) guarantees that the total delivery demands of all clients served by each depot should equal to depot's total delivery load; constraint (16) requires that the total pickup demands of all clients served by each depot should equal to depot's total pickup load; equation (17) expresses the proper movement of load delivery and/or pickup; constraint (18) makes sure that the total load of each arc of vehicles must not exceed vehicle's capacity.

4. Proposed Approach

In the following sections, we introduce HH by describing the adopted solution representation, effective constructive heuristic, low-level heuristics for domains, nine selection strategies, and four acceptance criteria for high-level heuristic.

4.1. Solution Representation. The solution representation plays an important role in improving the performance of an algorithm for the CLRPSPD, which should include all routes and the chosen depot of each route. Therefore, a string of cells are used to represent a complete solution which determines the assigned clients to each vehicle, the depots to be selected, and the sequence of clients to be visited by a specific vehicle starting and ending at the same depot. The solution representation in this paper applies a simple and efficient encoding [2], which is a complete set of vehicle routes, that is, $R = \{r_1, r_2, \ldots, r_K\}$ with inserting the chosen depot at two ends of each route. Meanwhile, we also store the attributes of each route in the second subcell of each route. It is worth noting that our chromosome representation can meet the constraints (2)–(18) for avoiding restoring the feasibility of solutions and allowing fast evaluation of its fitness value without the need for decoding.

4.2. Initial Solution. The initialization method is an improved greedy heuristic (inspired from the approach in Ref. [4]), named here as regret-k greedy heuristic (RKGH), proposed by our previous work [2]. After implementing greedy heuristic [4], configuration of depots ($m \geq 2$) has been determined. Let $x_{ik} \in \{1, 2, \ldots, m\}$ be a variable that indicates the depot for client i that has the k^{th} lowest cost, that is,$c(i, x_{ik}) \leq c(i, x_{ik'})$. Using this notation, we can define a regret-k value $\Delta c^k(i)$ as $\Delta c^k(i) = c(i, x_{ik}) - c(i, x_{i1})$. In other words, the regret-k value is the difference in the cost for client placed in the first best depot and its k^{th} best depot. The RKGH chooses to assign the client i that maximizes

$$\max \left\{ \alpha \times \sum_{j \in k} \Delta c^j (i) \right\}. \tag{19}$$

The multiplier α depends the remaining capacity of each depot, $\alpha = 1$ representing that client i is served by this depot without violating the capacitated constraints; otherwise, α is set to 0. In this paper, the value of k is also set to 2, namely, R2GH is applied.

4.3. Hyperlevel Heuristics. In this paper, we use different pairings of operators' vector and acceptance criterion. There are 36 possible pairings of nine operators' vectors and four acceptance criteria. Several strategies are involved: RP, TS, FRR-MAB, QS, AS, and PS, and three variants are also developed. The strategies implementing QS, PS, AS, and their variants can be categorised as ES. Four acceptance criteria are provided in this paper: AM, NA, GD, and SA. The brief descriptions are presented in the following sections.

4.3.1. Operator Selection Vector Design. Operator selection strategies in the literature generally assign a probability to each operator and use a roulette wheel or tournament-like process to select the LLHs according to them [54], named as operator selection vector. The selection vector consists of an array of operators, each with a probability of selection. In this paper, selection vector is designed by assigning a probability to each LLH, and the initial selection vector of LLHs has an equal probability of selection.

(1) RP: the initial vector is not changed during the run, and all LLHs have an equal selection probability regardless of performance, also called as fixed selector [84], so the random one is chosen as a baseline for comparison with others.

(2) TS: tabu list is used to prohibit LLHs with recently poor performance from being applied too soon. The LLHs belonging to be tabu (i.e., the selection probability is 0) are released (i.e., the selection probability is initial value) whenever other LLHs make a change in objective function, which is slightly different from the aspiration criterion [51].

(3) FRR-MAB [7]: FRR-MAB was proposed to adaptively select appropriate LLH based on its recent performance storing in slide windows implementing FIFO mechanism. Two successive stages are discussed: credit assignment and selection mechanism. The former offers the method to measure the impact on the quality in search process caused by the application of recent LLHs, while the latter selects one LLH with maximizing received credit values for generating new solution. For applying selection vector, the reward value of each LLH is normalized as selection probability. The other version based on

FRR-MAB is combined with TS (viewed as FMT) proposed by our previous work [2], absorbing the rapid response capability of TS to exclude LLHs with poor performance.

(4) QS: a heuristics search space with ξ LLHs in QS is defined as

$$H = \left(h_1, h_2, \mathrm{L}, h_\xi\right) = \begin{bmatrix} \beta_1 & \beta_2 & \mathrm{L} & \beta_\xi \\ \alpha_1 & \alpha_2 & \mathrm{L} & \alpha_\xi \end{bmatrix}, \tag{20}$$

where $h_i = [\,\beta_i \;\; \alpha_i\,]$ represents $h_i Q$-bit and $|\beta_i|^2$ and $|\alpha_i|^2$ give the probability that h_i selects and disuses state, respectively, and guarantee $|\alpha_i|^2 + |\beta_i|^2 = 1$. A learning strategy based on Q-gate is utilized to update selection probability $|\beta_i|^2$ of h_i based on real-time performance.

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\left(\theta_i^t\right) & -\sin\left(\theta_i^t\right) \\ \sin\left(\theta_i^t\right) & \cos\left(\theta_i^t\right) \end{bmatrix} \times \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix}, \tag{21}$$

$$\Delta\theta^t = \Delta\theta_0 \times \left(1 - k\frac{t}{t_{\max}}\right), \tag{22}$$

$$\theta_i^t = \frac{\left(f\left(\pi^t\right) - f\left(\pi_{best}^t\right)\right) \times \alpha_i^t \times \beta_i^t \times \left(x_i^t - bx_i^t\right)}{\left|f\left(\pi^t\right) - f\left(\pi_{best}^t\right)\right| \times \left|\alpha_i^t \times \beta_i^t\right|} \times \Delta\theta^t, \tag{23}$$

where $[\,\alpha_i^t \;\; \beta_i^t\,]$ is the Q-bit of h_i during t^{th} iteration and θ_i^t is the rotation angle of each Q-bit; $\Delta\theta^t$ is rotation angle in t^{th} iteration; k is uniformly distributed within $[\,0 \;\; 1\,]$; $t_{\max}$ is the maximum iteration; π^t and π_{best}^t are, respectively, sequence of current selected LLHs and selected LLHs in obtaining the best solution; x_i^t and bx_i^t are the state of h_i of current and global sequence of selected LLHs; $\pi^t = [\,4 \;\; 6 \;\; 7 \;\; 1\,]$ represents the current sequence of LLHs; and x^t can be illustrated as$x^t = [\,1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1\,]$ if $\xi = 7$. The other version based on QS is combined with TS (named as QS2), absorbing the rapid response capability of TS to exclude LLHs with poor performance.

(5) AS: inspired from application of ant colony optimization (ACO) in travelling salesman problem (TSP) and 0/1 knapsack problem, two versions of AS are developed into adaptively making decisions on selecting favourable sequence of LLHs. The first variant (paired-AS, AS2) highlights the joint performance of pairs of LLHs by determining next LLH (one at a time) based on probabilities proportional to the pheromone levels of each heuristic pair shown in equation (24); in this version, pheromone laying criterion [59] is modified for laying the pheromone for pairs which reach an improvement to the previous function, that is, if an ant performs heuristics h_x, h_y, and h_z and h_y leads to a nonimproving solution and h_z provides a better solution, pheromone will be laid on edge *x-z* and neither *x-y* nor *y-z*, while the latter (single-AS, AS) emphasizes individual performance of each LLH by viewing the performance information of individual LLH as its pheromone trail [67]. Possible sequence of LLHs is selected by normalizing pheromone trail as selection probabilities.

$$p_{ij}^{t+1} = \frac{\tau_{ij}^{t+1}}{\sum\limits_{j \in \xi, j \neq i} \tau_{ij}^{t+1}}, \tag{24}$$

$$\tau_{ij}^{t+1} = \begin{cases} \rho \cdot \tau_{ij}^t + \dfrac{FIR_j}{\eta_{ij}}, & \text{if ant } k \text{ selects } (i, j), \\ \rho \cdot \tau_{ij}^t, & \text{otherwise,} \end{cases} \tag{25}$$

where τ_{ij}^t $(\xi \times \xi)$ and p_{ij}^t $(\xi \times \xi)$ define, respectively, the pheromone trail and selection probability of *i-j* edge in t^{th} iteration; $0 < \rho \leq 1$ represents the evaporation rate for pheromone; FIR_j is fitness improvement rate compared to previous solutions [7, 55]; and η_{ij} $(\xi \times \xi)$ is the visibility information illustrating heuristic information. The above equation (25) is used to update pheromone trail with $\eta_{ij} = T_i + T_j$ (T_i indicates the running time of h_i), while equation (25) can be modified for AS with $\eta_j = T_j$ by ignoring the previous h_i.

(6) PS: PS has been nominated as HLH strategies to measure effectiveness of their order placement and selection mechanism. Each particle is a vector of ξ numbers representing the selection probabilities of LLHs obtained by normalizing the position (using the original formulations) of this particle. The particles impose the order that the LLHs are applied to solution domain. Each particle is evaluated using performance indicator FIR.

$$p_i^t = \frac{X_i(t)}{\sum\limits_{i \in \xi} X_i(t)}. \tag{26}$$

Among the last seven selection strategies, probabilities of selecting LLHs are utilized to order LLHs, with the maximum utility score/weight being placed in the front of the list. The chosen LLHs are then applied in sequence following this order. However, the sequence of selected LLHs are randomly permutated to modify the current solution for RP and TS.

4.3.2. Acceptance Criteria Design. After recent application of the LLHs, the obtained solution is considered for accepting as incumbent solution into next iteration. If the new solution is at least as good as the previous solution it will replace, and then it is automatically accepted as current solution regardless of the nature of an acceptance mechanism; otherwise, acceptance criterion is utilized to whether discard the new solution or not [65, 84]. The following four acceptance criteria are proposed to pair up with the above selection strategies, aiming at pointing out which pairs perform better than others.

(1) AM [5]: the current result is replaced by all new results with probability value at 1.

(2) NA [84]: the current result is replaced by no-improving results with probability value at 0.5.

(3) GD [50, 84]: a no-improving result is accepted if it is better than a dynamically changing threshold value which depends on the current and historical fitness information. For determining the range of accepting a child solution, a threshold value is applied, which decreases with the increase in iteration.

$$S^{t}_{threshold} = \left(1 + 0.5 \times \left(1 - \frac{t}{t_{\max}}\right)\right) \times G_{best}(t), \tag{27}$$

where t and $t_{\max}$ indicate the current steps and the maximum iteration and $G_{best}(t)$ represents the global best solution objective value found so far.

(4) SA [65]: new solutions are accepted as the current solutions if the probability criterion is met, that is, improving results are accepted with probability value at 1 while no-improving results are accepted if a uniform value within [0, 1] is less the critical value. The probability of accepting a no-improving solution in SA is calculated for making a decision on whether accepting it or not [65]:

$$p^{t} = e^{-\left(\Delta f/\Delta F^{t}\left(1-(t)/\left(t_{\max}\right)\right)\right)}, \tag{28}$$

where Δf is the difference in the quality between the new solution and current solution and ΔF^{t} represents the expected range for maximum solution quality change, $\Delta F^{t} = G_{\text{best}}(t) - G_{\text{best}}(0)$.

4.4. Low-Level Heuristics. The pool of LLHs can be viewed as a "black box" which are used to perturb the incumbent solution by either intensifying or diversifying the search in the search region, which is used in Ref. [2]. They normally are a set of simple, cheap, and knowledge-poor LLHs [5]. The module of this paper is composed of 13 LLHs $h_1, h_2, \ldots, h_{13}$ across two pools of heuristics: mutational heuristic (MH) and local search/hill climber (HC), which are identified by the role in improving or worsening the solution. The first six LLHs are implemented to explore new region: 2-opt, or-opt, shift, interchange, add, and Shaw. Others are utilized to exploit better solutions: relocation, 2-opt, 3-opt, relocate, and 1-interchange algorithm, which apply intra and inter-route moves.

The detailed information of h_1~h_4 can be obtained from Ref. [54], which are simple and basic mutational heuristics. The h_5 heuristic diversifies the open depots by opening a new one and randomly assigning between 1 and 2/3 of the routes to it [37]. As to the last diversified LLH h_6 [85], it is a method named as destroy-and-repair operator proposed to remove related clients, i.e., clients that are geographically close to each other and reinserted into best positions. In our paper, basic greedy heuristic [86] is suggested to reinsert removal clients back into routes. The disturbed solutions are accepted as long as the above acceptance criteria and the vehicle capacity at each arc and depots capacity are obeyed.

The relocation h_7 is developed to reselect the appropriate depots for determined routes. Each route is collapsed into a cluster and the smallest insertion costs can be calculated as the distance of the depots in the original routes. The depots with the larger number of clusters take priority to open. And then other unassigned clusters (from large to small order) sharing same closest depot will be arranged to its closest depot, unless the assigned depots' capacity and cost are not satisfied. The 2-opt [87] inside the routes is equivalent to the well-known 2-opt move (h_8) [88], whereas the other version of 2-opt implemented between different routes is identified (h_9). Due to reduced CPU time, the intral-3-opt (h_{10}) [89] is only considered. The relocate heuristic [90] reinserts a single client in another position inside the incumbent route (h_{11}) or in another route (h_{12}). The 1-interchange [91] heuristic swaps each position of client from one route with each client in another route if not sharing the same depot (h_{13}). If improvements are found and depot capacity and vehicle load at each arc are met, the above moves are implemented, allowing them to improve incumbent solutions.

The MHs often perform simple random moves to perturb the incumbent solution without guaranteeing competitive results. Although insufficient for achieving competitive results, they are useful for providing randomization and helpful for navigating out of local optima. However, HCs play a key role in quickly obtaining much better solutions at each step. Moreover, it might be desirable to apply HCs after application of any MH and to apply MHs if it is hard for HCs to provide promising solutions. Considering the above features, the proposed nine selection strategies are utilized to adaptively manage the HCs, inspired from the mechanism on management of local search heuristics [5, 54] and the method for automatically classifying according to the performance proposed in Ref. [57]. The probability for selecting any one MH as guider for getting rid of local optimality is set to 1 if and only if no HC can provide an improvement to previous objective value; otherwise, the probability value is set to 0. Hence, Figure 1 is the framework of hyperheuristics applied in this paper.

Moreover, a stopping mechanism is defined by evaluation limits with maximum number of fitness evaluation ($\alpha_{\max}$), aiming at providing fair comparison.

5. Computational Evaluation

In this paper, we explore the performance of a set of hyperheuristics in solving CLRP and CLRPSPD. The experiments consist of four parts. In the first part, an overall picture related to the selection strategies and acceptance alternatives is provided to rank the performance of 36 pairs and determine first four outstanding pairs for the following experiments. Then, the efficiency and features are analysed for the chosen pairs by implementing on the three sets of

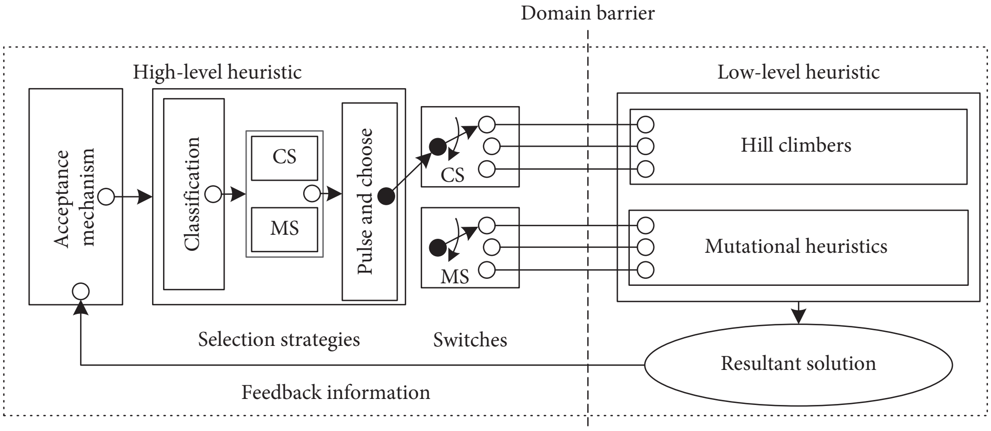


Figure 1: Framework of hyperheuristic approach.

Table 2: Configuration for each selection strategy.

Strategies	Parameters	Values	Common parameters
FRR-MAB	Scaling factor C	0.5 [7]	Initial probability $p = 0.5$
	Sliding windows size W	25 [56]	Population size $S_{LLH} = 1$ (single-point search)
QS/QS2	Initial rotation angle, $\Delta\theta_0$	0.02π	Population size $S_{HLH} = 1$ (only one individual)
	Initial probability amplitudes α, β	0.7071 [74]	Maximum fitness evaluation $\alpha_{\max}$ (Eq. 29)
AS/AS2	Initial pheromone, τ_{ij}/τ_i	0	Maximum iteration $t_{\max} = 10^6$
	Initial visibility, η_{ij}/η_i	0	
	Evaporation rate, ρ	0.2 [69]	
PS	Learning factors, c_1, c_2	2 [72]	
	Inertia weight, w	1 [72]	

CLRP benchmark instances. In the third part, we explore the performance of selected pairs in solving CLRPSPD benchmark sets. Finally, the performance of selected pairs is tested against recent and effective tailored approaches in the literature.

5.1. Experimental Design. The presented HH approach with 36 pairs is coded in Matlab 8.6 and runs on a desktop computer with Intel Core (TM) i7-6700K (4.00 GHz) and 8 GB RAM, under Windows 10; it is embedded in the CLOR tool, available by emailing to us.

As far as calibration is considered, two types of parameters are concerned: specific selection strategy parameters and common hyperheuristic parameters. The former relates to the setting of selection strategies, while the latter accounts for the common parameters between LLHs and HLH, and they are summarized in Table 2.

Some of the specified parameters follow defaults suggested in the literature, and others were determined by conducting an initial experiment with various settings aiming at obtaining relative better results within reasonable CPU time. As to common hyperheuristic parameters, initial selection probability p is set to 0.5 for each LLH in line with the initial probability amplitudes of QS; tabu list size l_t is set to 7 which is the number of HCs; as single-point-based search hyperheuristic framework is investigated in this paper, the sizes of population used in the LLH and HLH are set at 1. For fairly comparing the running time of each pair, the maximum number of fitness evaluation for each instance is provided, which was different from the method in Ref. [2] (i.e., the maximum iterations without improving solution), depending directly on the number of clients, depots, and vehicles:

$$\alpha_{\max} = a \times (m + n + K)^2. \tag{29}$$

Aiming at obtaining a good trade-off between solution quality and computing time, the multiplier a is taking on the value 5 for all instances in this paper.

5.2. Benchmark Instances. Three sets of CLRP benchmark instances were adopted to evaluate the efficiency and features of each pair for hyperheuristic. These sets are provided in Refs. [26–28]. One separation approach is used to generate the datasets of CLRPSPD from the above CLRP benchmarks, instanced as W type in Ref. [92] with setting 0.8 for β.

The first benchmark instances [28] contain 19 cases. The number of clients ranges from 12 to 318, and the number of depots m varies between 2 and 15, and the fixed vehicle cost f_v is not given. In the randomly generated set [27], the number of clients ranges from 20 to 200, the number of depots m is 5 and 10, and vehicle capacity Q is {70, 150} and $f_v = 1000$. The benchmark instances [26] are also randomly generated which contain 36 cases with $n \in \{100, 150, 200\}$, $m \in \{10, 20\}$, $Q = 150$, and $f_v = 10$. And the clients' demand of this benchmark ranges from 1 to 20. The above three benchmark sets are available at http://sweet.ua.pt/sbarreto/or_http://prodhonc.free.fr/homepage.

Euclidean distances are applied in all datasets. In Prins et al. benchmark set for CLRPSPD, the obtained distances are multiplied by 100 (rounded up to the next integer in CLRP) and the total costs are rounded up to the next integer, which is different from the sets obtained in Refs. [4, 38]. The working cost of vehicle for each instance with less than 100 clients in the first set increases to 20, namely, $f_v = 20$.

5.3. Results

5.3.1. Experiment on 36 Pairs. The first set of experiment is performed to compare the performance of 36 pairs of hyper-heuristics aiming at determining the first four outstanding pairs for the following experiments. Nine CLRP instances in Barreto set with the number of clients ranging from 50 to 318 were used to test performance of each pair, implementing fifty runs on each instance. The maximum number of fitness evaluation is set to 10000. Aiming at comparing and scoring for all pairs, formula (1) scoring system was adopted, which was used to rank the approaches by CHESC (Cross-domain Heuristic Search Challenge) before 2010 (http://www.asap.cs.nott.ac.uk/external/chesc2011/). The top eight approaches are given a score of 10, 8, 6, 5, 4, 3, 2, and 1 point for each problem from the best to the worst, successively [65]. The rest of the methods receive a score of 0. In this paper, the score of 10 points is assigned to the pairs which can exploit the best known solution (BKS), and the scores for the rest of pairs are similar to formula (1) scoring system. Therefore, 90 is the maximum overall score a pair can get.

Experiment results are shown in Table 3, where the overall scores of the best four pairs, nine selection strategies, and four acceptance criteria are provided. As can be seen from the results, HH performing AS-AM is the best clear winner, and PS-AM, QS2-SA, and TS-SA also outperform other pairs. The hyperheuristic using AS, FRR-MAB-TS, TS, and RP as selection components performs better than others regardless of the acceptance component, and several interesting conclusions can be drawn from the overall scores that (1) TS could promote better performance for selection strategies by excluding the LLHs with poor performance, when compared with the performance of selection strategies without TS, reaching an average improvement about score 25 point; (2) the performance of AS using 0/1 knapsack problem outperforms AS2 applying TSP, in other words, compared to the one laying ant pheromone trails at routes, the mechanism laying ant pheromone trails on vertices (i.e., LLHs) have a greater chance of success in guiding ants to select the promising sequence of LLHs; (3) PS-AM and QS2-SA ranks the second/third place in overall scores of pairings, even if the performance of PS/QS2 is barely satisfactory. From the last four scores of acceptance component regardless of the selection component, AM and SA rank, respectively, the first and second with a score 529 and 518, while the other two receive a score of 473 and 429, respectively.

Table 3: The overall formula (1) scores for the top four pairs and strategies.

Pairs	Overall score
AS-AM	73
PS-AM	67
QS2-SA	66
TS-SA	65
RP	222
TS	229
FRR-MAB	204
FMT	233
QS	176
QS2	215
AS	242
AS2	212
PS	216
AM	529
NA	473
GD	429
SA	518

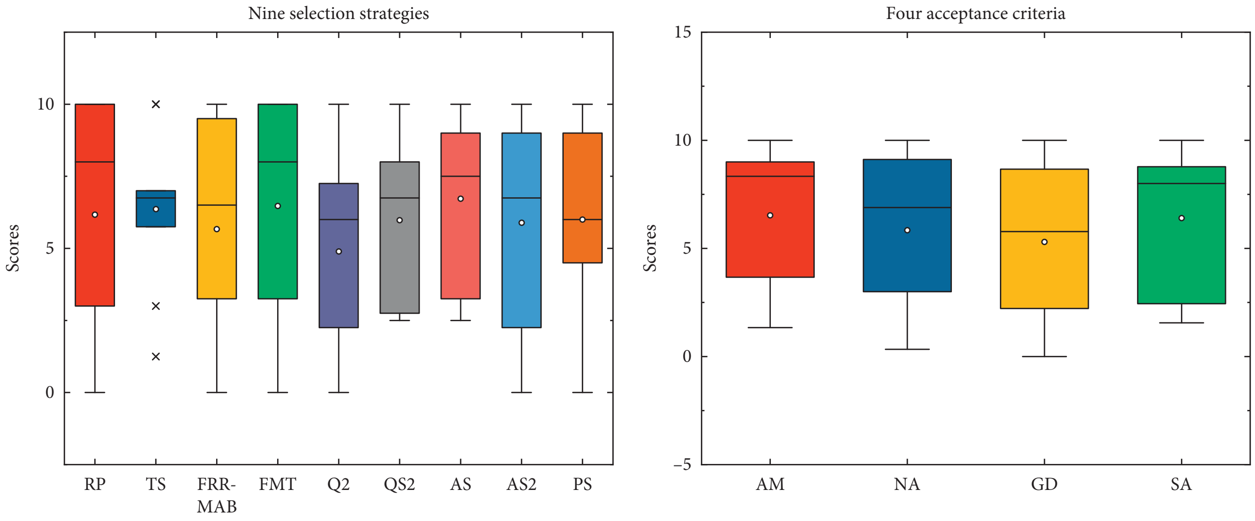
Figure 2 shows the box plot for the scores of nine selection strategies and four acceptance criteria. In the box plot, the minimum and maximum values obtained (excluding the outliers), the lower and upper quartiles, and median are shown.

From such analysis and answering the first experiments, we identify AS-AM as the best pair considering most cases and scores, and additional PS-AM, QS2-SA, and TS-SA are also selected to conduct the following experiments.

5.3.2. Experiment on CLRP. The first four outstanding pairs were tested performing fifty runs on each instance, from which best found solution, gaps to BKS, and computing time are shown, providing the insufficient page space. Results for three CLRP benchmark instances are shown in Tables 4–6. The first column of each table is the name of each instances, followed by BKS, and results concerning four pairs which contain best found results (Best), gap (in percentage) to the BKS, and average computing time in seconds (CPU). Average and median values for gaps and CPU are displayed in the last two rows.

For the first three pairs (Table 4), the BKSs of instances with less than 150 clients are exploited with average gap at −0.04% for TS-SA, −0.07% for QS2-SA, and −0.05% for AS-AM. Only one BKS cannot be found by the last pair with highest gap at 0.06% and average value at −0.04%. Concerning the median values of gaps to BKS, they take on the value 0. Computing times are on average less than 29, 25, 27, and 17 s, and median values are lower than 6 s even if the largest instances exist. Two new BKS are found by four pairs, obtaining an improvement of over 0.5% for Perl83-318 × 4 by PS-AM and over 0.7% for Perl83-318 × 4-2 by QS2-SA.

In the second benchmark (Table 5), at least 16 BKS can be found by the four pairs, with difference in finding the Best of 50-5-2b. Four new BKSs in this set are obtained, respectively, reaching an improvement of over 0.3% for 200-10-1b by TS-SA and PS-AM and 0.1% for 200-10-2b and



(a)

(b)

Figure 2: Box plots of scores for a statistical comparison of (a) selection strategies and (b) acceptance criteria.

Table 4: Results for the CLRP instances in Barreto set.

Set	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
P12-2	204.0	**204.0**	0.00	0.1	**204.0**	0.00	0.1	**204.0**	0.00	0.1	**204.0**	0.00	0.1
G21-5	424.9	**424.9**	0.00	0.4	**424.9**	0.00	0.4	**424.9**	0.00	0.4	**424.9**	0.00	0.3
G22-5	585.1	**585.1**	0.00	0.3	**585.1**	0.00	0.4	**585.1**	0.00	0.3	**585.1**	0.00	0.3
M27-5	3062.0	**3062.0**	0.00	0.6	**3062.0**	0.00	0.6	**3062.0**	0.00	0.5	**3062.0**	0.00	0.5
G29-5	512.1	**512.1**	0.00	0.6	**512.1**	0.00	0.6	**512.1**	0.00	0.5	**512.1**	0.00	0.5
G32-5	562.2	**562.2**	0.00	0.7	**562.2**	0.00	0.7	**562.2**	0.00	0.6	**562.2**	0.00	0.6
G32-5-2	504.3	**504.3**	0.00	0.6	**504.3**	0.00	0.7	**504.3**	0.00	0.6	**504.3**	0.00	0.5
G36-5	460.4	**460.4**	0.00	0.9	**460.4**	0.00	0.9	**460.4**	0.00	0.8	**460.4**	0.00	0.7
C50-5	565.6	**565.6**	0.00	2.0	**565.6**	0.00	2.2	**565.6**	0.00	1.8	**565.6**	0.00	1.6
P55-15	1112.1	**1112.1**	0.00	4.2	**1112.1**	0.00	5.2	**1112.1**	0.00	4.6	1112.8	0.06	4.2
C75-10	844.4	**844.4**	0.00	7.9	**844.4**	0.00	7.9	**844.4**	0.00	7.0	**844.4**	0.00	6.3
P85-7	1622.5	**1622.5**	0.00	9.6	**1622.5**	0.00	9.7	**1622.5**	0.00	8.6	**1622.5**	0.00	7.7
D88-8	355.8	**355.8**	0.00	7.0	**355.8**	0.00	6.3	**355.8**	0.00	6.5	**355.8**	0.00	5.9
C100-10	833.4	**833.4**	0.00	11.4	**833.4**	0.00	12.3	**833.4**	0.00	10.6	**833.4**	0.00	8.3
O117-14	12290.3	**12290.3**	0.00	18.3	**12290.3**	0.00	18.4	**12290.3**	0.00	16.4	**12290.3**	0.00	14.6
M134-8	5709.0	**5709.0**	0.00	32.7	**5709.0**	0.00	33.0	**5709.0**	0.00	29.9	**5709.0**	0.00	24.1
D150-10	43919.9	**43919.9**	0.00	42.7	**43919.9**	0.00	45.0	**43919.9**	0.00	38.3	**43919.9**	0.00	33.1
P318-4	7249.3[a]	4484.2	−0.50	156.4	4145.9	−0.56	110.0	*3966.0*	−0.59	126.8	3991.1	−0.58	78.8
P318-4-2	13070.0[b]	10866.5	−0.33	243.4	*8293.0*	−0.72	213.0	10269.4	−0.42	260.6	10905.1	−0.33	125.3
AV	68099.3	67837.8	−0.04	28.4	67684.6	−0.07	24.6	67779.1	−0.05	27.1	67813.9	−0.04	16.5
MD			0.00	4.2		0.00	5.2		0.00	4.6		0.00	4.2

Bold numbers are the best known solutions. Italic numbers are the minimum value among four obtained solutions. Underscore numbers are the obtained values less than the best known solutions. [a]Results minus 550000. [b]Results minus 650000.

200-10-3b by TS-SA. Concerning gaps of Best, average values are, respectively, 0.06% for TS-SA and 0.07% for other pairs, and the highest gaps are less than 0.8% for four pairs. Median values equal to 0 for all pairs. Average and median values for CPU are, respectively, 50.8 and 14.9 s for TS-SA, 47.7 and 15.3 s for QS2-SA, 41 and 13.1 s for AS-AM, and 33.6 and 12.6 s for PA-AM.

For the Tuzun and Burke benchmark (Table 6), TS-SA and PS-AM tie for first place in the number of new BKS found with 13 new BKSs, and QS2-SA tails the winner to take the second place by exploiting 12 new BKSs and AS-AM is in the last place. Concerning the number of solution less than or equal to BKS, TS-SA, PS-AM, and QS2-SA, respectively, rank the first, second, and third with 29, 28, and 27 solutions less than or equal to BKS, and AS-AM ranks the last place with 26 solutions less than or equal to BKS. What surprised us is that the number of new BKS accounts for over 30%, except for AS-AM with 27.8% of 36 instances in this benchmark. Concerning gaps of Best, average values are, respectively, −0.01% for TS-SA, 0.01% for QS2-SA and PS-

Table 5: Results for the CLRP instances in Prins set.

Set	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
20-5-1	54793	**54793**	0.00	0.5	**54793**	0.00	0.5	**54793**	0.00	0.5	**54793**	0.00	0.5
20-5-1b	39104	**39104**	0.00	0.3	**39104**	0.00	0.3	**39104**	0.00	0.2	**39104**	0.00	0.2
20-5-2	48908	**48908**	0.00	0.5	**48908**	0.00	0.6	**48908**	0.00	0.4	**48908**	0.00	0.4
20-5-2b	37542	**37542**	0.00	0.3	**37542**	0.00	0.3	**37542**	0.00	0.3	**37542**	0.00	0.2
50-5-1	90111	**90111**	0.00	3.9	**90111**	0.00	4.1	**90111**	0.00	2.9	**90111**	0.00	2.6
50-5-1b	63242	**63242**	0.00	1.8	**63242**	0.00	2.0	**63242**	0.00	1.9	**63242**	0.00	1.7
50-5-2	88298	**88298**	0.00	4.2	**88298**	0.00	4.3	**88298**	0.00	3.6	**88298**	0.00	3.4
50-5-2b	67308	67373	0.10	2.1	**67308**	0.00	2.1	67449	0.21	1.7	**67308**	0.00	1.5
50-5-2bis	84055	**84055**	0.00	3.5	**84055**	0.00	3.7	**84055**	0.00	3.1	**84055**	0.00	2.9
50-5-2bbis	51822	**51822**	0.00	2.4	**51822**	0.00	2.5	**51822**	0.00	2.1	**51822**	0.00	2.0
50-5-3	86203	**86203**	0.00	3.9	**86203**	0.00	4.3	**86203**	0.00	3.4	**86203**	0.00	3.1
50-5-3b	61830	**61830**	0.00	1.9	**61830**	0.00	1.9	**61830**	0.00	1.6	**61830**	0.00	1.6
100-5-1	274814	276096	0.47	28.9	276144	0.48	30.9	275601	0.29	25.8	*275598*	0.29	22.6
100-5-1b	213615	*213654*	0.02	14.1	*213654*	0.02	13.5	213671	0.03	12.2	*213654*	0.02	10.9
100-5-2	193671	**193671**	0.00	35.1	**193671**	0.00	34.4	**193671**	0.00	26.6	**193671**	0.00	26.0
100-5-2b	157095	157129	0.02	14.1	157144	0.03	15.2	157110	0.01	13.8	157129	0.02	12.3
100-5-3	200079	**200079**	0.00	34.3	**200079**	0.00	34.5	**200079**	0.00	25.7	**200079**	0.00	26.8
100-5-3b	152441	**152441**	0.00	13.8	**152441**	0.00	13.0	**152441**	0.00	12.3	**152441**	0.00	11.3
100-10-1	287695	*287688*	0.00	31.7	*287688*	0.00	32.8	*287688*	0.00	27.4	*287688*	0.00	25.4
100-10-1b	230989	*231608*	0.27	17.6	231833	0.37	16.7	231833	0.37	11.1	232222	0.53	13.8
100-10-2	243590	**243590**	0.00	28.6	**243590**	0.00	29.2	**243590**	0.00	24.4	**243590**	0.00	22.0
100-10-2b	203988	**203988**	0.00	15.7	**203988**	0.00	17.4	**203988**	0.00	15.5	**203988**	0.00	14.1
100-10-3	250882	252890	0.80	33.8	252890	0.80	33.2	*252751*	0.74	27.9	252890	0.80	27.2
100-10-3b	204317	*204567*	0.12	15.8	*204567*	0.12	15.4	*204567*	0.12	13.8	*204567*	0.12	12.9
200-10-1	475294	*475770*	0.10	268.1	476166	0.18	263.8	476002	0.15	219.6	476722	0.30	190.0
200-10-1b	377043	*375820*	−0.32	135.0	376313	−0.19	121.4	376662	−0.10	96.7	375836	−0.32	91.0
200-10-2	449006	*449037*	0.01	250.9	449142	0.03	236.7	449332	0.07	210.4	449110	0.02	132.4
200-10-2b	374280	373882	−0.11	153.0	373973	−0.08	151.1	*374039*	−0.06	125.8	374093	−0.05	110.6
200-10-3	469433	*470780*	0.29	247.8	471043	0.34	223.1	471043	0.34	208.2	471073	0.35	124.8
200-10-3b	362653	*362276*	−0.10	161.2	362349	−0.08	122.0	362596	−0.02	110.0	362640	0.00	113.1
AV	196470	196608	0.06	50.8	196663	0.07	47.7	196667	0.07	41.0	196674	0.07	33.6
MD		196875	0.00	14.9		0.00	15.3		0.00	13.1		0.00	12.6

Bold numbers are the best known solutions. Italic numbers are the minimum value among four obtained solutions. Underscore numbers are the obtained values less than the best known solutions.

AM, and 0.03% for SA-AM, and median values are 0 for all pairs. Computing times are on average 77.6, 74.5, 59.6, and 52.5 s, and median values are 62.6, 64.4, 53.8, and 50.3 s.

We conduct statistical analysis for obtained results of three CLRP benchmark sets in Tables 4–6 through Table 7 based on multiple pairwise comparisons with 95% confidence level (i.e., $\alpha = 0.05$), namely, Friedman tests. In the Friedman test, the null hypothesis (H_0) is only rejected if the Friedman statistic (χ^2) is greater than the critical value, looking up in χ^2 table through sample size and confidence level, and post hoc test based on Wilcoxon Rank-Sum should be carried out to detect the significant difference among samples, when needed; otherwise, the null hypothesis (H_0) is accepted indicating that no significant difference is concerned among four pairs. In Table 7, Friedman statistics (χ^2) take the values 3.923 for Barreto et al., 6.750 for Prins et al., and 6.201 for Tuzun and Burke, which are, respectively, lower than 10.117, 18.493, and 23.269, indicating that there is no significant difference among performance of four pairs.

5.3.3. Experiment on CLRPSPD. The CLRPSPD benchmark instances, which were first converted in Refs. [2, 38], were also used to evaluate the features of the best four pairs. However, this paper only considered the *W* type of the above sets for the CLRPSPD. The main reason is that the sets of *X*, *Y*, and *Z* types show the same characteristics with the original sets of CLRP, and it is most difficult to obtain BKSs for the instances using *W* type separation approach. All BKSs were obtained from Ref. [1]. Results for the three CLRPSPD benchmark instances are shown in Tables 8–10. The first column of each table is the name of each instances, followed by the best known solution (BKS), data concerning the first four best pairs which contain best results (Best), gap (in percentage) to the BKS, and average computing time in seconds (CPU). Average (AV) and median values (MD)for gaps or gap and CPU are displayed in the last two rows.

For the Barreto benchmark set, computing times are on average lower than 35, 30, 29, and 18 s and median values are 3.9, 4, 3.4, and 3.1 s, respectively. Concerning gaps of Best,

Table 6: Results for the CLRP instances in Tuzun and Burke.

Set	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
P1	1467.68	**1467.68**	0.00	15.2	**1467.68**	0.00	15.4	**1467.68**	0.00	13.0	**1467.68**	0.00	11.6
P2	1449.20	*1448.37*	−0.06	20.4	**1449.20**	0.00	20.1	**1449.20**	0.00	17.6	*1448.37*	−0.06	14.7
P3	1394.80	**1394.80**	0.00	15.1	**1394.80**	0.00	16.4	1394.93	0.01	14.2	**1394.80**	0.00	12.3
P4	1432.29	**1432.29**	0.00	20.7	**1432.29**	0.00	20.9	**1432.29**	0.00	17.6	**1432.29**	0.00	15.4
P5	1167.16	**1167.16**	0.00	14.5	**1167.16**	0.00	15.7	**1167.16**	0.00	13.2	**1167.16**	0.00	12.0
P6	1102.24	**1102.24**	0.00	19.0	**1102.24**	0.00	18.8	**1102.24**	0.00	16.2	**1102.24**	0.00	14.5
P7	791.66	**791.66**	0.00	15.6	**791.66**	0.00	15.8	**791.66**	0.00	13.4	**791.66**	0.00	12.6
P8	728.30	**728.30**	0.00	17.8	**728.30**	0.00	17.6	**728.30**	0.00	15.4	**728.30**	0.00	12.9
P9	1238.24	*1238.49*	*0.02*	16.6	*1238.49*	0.02	15.9	*1238.49*	0.02	14.6	*1238.49*	0.02	13.6
P10	1245.31	**1245.31**	0.00	23.9	1245.42	0.01	27.6	**1245.31**	0.00	18.1	**1245.31**	0.00	15.5
P11	902.26	**902.26**	0.00	15.3	**902.26**	0.00	15.4	**902.26**	0.00	12.9	**902.26**	0.00	11.9
P12	1018.29	**1018.29**	0.00	17.9	**1018.29**	0.00	18.2	**1018.29**	0.00	15.6	**1018.29**	0.00	12.7
P13	1866.75	1895.83	*1.56*	61.8	1895.83	1.56	63.1	*1892.17*	1.36	53.6	1895.83	1.56	52.9
P14	1823.20	*1820.12*	−0.17	62.1	1822.69	−0.03	61.3	1822.15	−0.06	53.8	1822.69	−0.03	42.1
P15	1964.30	*1965.12*	0.04	57.0	*1965.12*	0.04	*57.9*	*1965.12*	0.04	48.9	*1965.12*	0.04	42.6
P16	1792.80	*1792.77*	−0.00	70.9	*1792.77*	−0.00	69.9	*1792.77*	−0.00	66.9	*1792.77*	−0.00	55.3
P17	1443.33	*1443.32*	−0.00	58.6	*1443.32*	−0.00	64.5	*1443.32*	−0.00	53.7	*1443.32*	−0.00	51.3
P18	1434.60	1434.82	0.02	63.0	1433.16	−0.10	64.1	1433.49	−0.08	53.2	*1431.24*	−0.23	46.6
P19	1204.42	**1204.42**	0.00	63.5	**1204.42**	0.00	64.6	**1204.42**	0.00	57.1	**1204.42**	0.00	61.7
P20	930.99	*927.63*	−0.36	64.3	931.28	0.03	65.6	931.28	0.03	54.3	929.26	−0.19	43.8
P21	1694.18	**1694.18**	0.00	71.3	**1694.18**	0.00	64.2	**1694.18**	0.00	55.3	**1694.18**	0.00	52.1
P22	1392.01	**1392.01**	0.00	61.6	1392.18	0.01	64.6	**1392.01**	0.00	53.1	**1392.01**	0.00	49.3
P23	1198.20	*1197.95*	−0.02	57.5	*1197.95*	−0.02	*57.2*	*1197.95*	−0.02	48.3	*1197.95*	−0.02	40.8
P24	1151.80	**1151.80**	0.00	72.1	**1151.80**	0.00	73.1	**1151.80**	0.00	61.2	**1151.80**	0.00	52.9
P25	2243.40	*2237.73*	−0.25	136.6	2240.37	−0.14	131.1	2244.64	0.06	91.2	2244.79	0.06	84.9
P26	2138.40	*2139.67*	0.06	137.9	*2139.67*	0.06	130.1	2144.49	0.28	102.5	2141.09	0.13	73.2
P27	2209.30	*2204.93*	−0.20	178.8	2208.48	−0.04	151.4	2209.36	0.00	110.0	2207.69	−0.07	96.1
P28	2222.90	2226.50	*0.16*	160.4	2223.61	0.03	150.3	2228.27	0.24	113.2	*2221.59*	−0.06	110.4
P29	2073.70	*2074.86*	*0.06*	137.7	2079.96	0.30	130.6	2076.16	0.12	100.4	2077.01	0.16	88.9
P30	1692.17	*1685.65*	−0.39	135.8	*1685.65*	−0.39	129.5	1687.38	−0.28	102.6	*1685.78*	−0.38	80.6
P31	1453.18	*1449.96*	−0.22	175.1	*1449.46*	−0.26	153.4	1450.90	−0.16	114.8	1453.89	0.05	100.4
P32	1082.46	**1082.46**	0.00	167.3	**1082.46**	0.00	152.6	**1082.46**	0.00	108.5	**1082.46**	0.00	111.2
P33	1954.70	1949.29	−0.28	126.3	1950.60	−0.21	127.8	*1947.84*	−0.35	112.6	1949.38	−0.27	78.6
P34	1918.93	1916.18	−0.14	137.4	*1911.73*	−0.38	129.6	1917.98	−0.05	91.7	1912.61	−0.33	82.5
P35	1762.00	1760.60	−0.08	155.6	1761.22	−0.04	161.1	*1760.04*	−0.11	117.3	1760.63	−0.08	113.3
P36	1390.87	**1390.87**	0.00	167.3	**1390.87**	0.00	148.2	**1390.87**	0.00	111.1	1390.94	0.01	117.5
AV	1499.33	1499.32	−0.01	77.6	1499.63	0.01	74.5	1499.97	0.03	59.6	1499.59	0.01	52.5
MD		1439.07	0.00	62.6		0.00	64.4		0.00	53.8		0.00	50.3

Bold numbers are the best known solutions. Italic numbers are the minimum value among four obtained solutions. Underscore numbers are the obtained values less than the best known solutions.

Table 7: Friedman's test results for CLRP benchmark instances ($\alpha = 0.05$).

Benchmark	Mean ranks for four pairs				Test statistics				Conclusion
	TS-SA	QS2-SA	AS-AM	PS-AM	n	χ^2	df	p	
Barreto set	2.58	2.42	2.37	2.63	19	3.923	3	0.270	$\chi^2 <$ critical value, accept H_0
Prins set	2.18	2.58	2.60	2.63	30	6.750	3	0.080	
Tuzun and Burke	2.19	2.61	2.65	2.54	36	6.201	3	0.102	

average values are, respectively, 0, 0.03%, 0.01%, and 0.03%, and median values equal to 0. New BKSs for instance with 318 clients are found with an improvement of 0.11% to previous BKS by AS-AM and PS-AM. Gaps to BKS are very low for all instances with highest gaps lower than 0.08% for the first pair, 0.25% for the second and third pairs, and 0.42% for the last one.

Looking at Table 9, computing times are on average less than 65, 60, 50, and 40 s, respectively, and median values are, respectively, 17.6, 18.3, 16.2, and 13.8 s. Concerning gaps to

Table 8: Results for the CLRPSPD instances in Barreto set.

Instance	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
P12-2	243.98	**243.98**	0.00	0.1	**243.98**	0.00	0.1	**243.98**	0.00	0.1	**243.98**	0.00	0.1
G21-5	528.42	**528.42**	0.00	0.4	**528.42**	0.00	0.5	**528.42**	0.00	0.3	**528.42**	0.00	0.3
G22-5	653.8	**653.80**	0.00	0.3	**653.80**	0.00	0.4	**653.80**	0.00	0.3	**653.80**	0.00	0.2
M27-5	3142.02	**3142.02**	0.00	0.6	**3142.02**	0.00	0.7	**3142.02**	0.00	0.5	**3142.02**	0.00	0.5
G29-5	592.1	**592.10**	0.00	0.6	**592.10**	0.00	0.6	**592.10**	0.00	0.4	**592.10**	0.00	0.4
G32-5	696.38	**696.38**	0.00	0.9	**696.38**	0.00	1.0	**696.38**	0.00	0.8	**696.38**	0.00	0.7
G32-5-2	595.27	**595.27**	0.00	0.7	**595.27**	0.00	0.8	**595.27**	0.00	0.6	**595.27**	0.00	0.6
G36-5	540.37	**540.37**	0.00	0.9	**540.37**	0.00	1.0	**540.37**	0.00	0.7	**540.37**	0.00	0.7
C50-5	708.37	**708.37**	0.00	1.9	**708.37**	0.00	2.0	**708.37**	0.00	1.5	**708.37**	0.00	1.4
P55-15	1327.06	**1327.06**	0.00	3.9	**1327.06**	0.00	4.0	**1327.06**	0.00	3.4	**1327.06**	0.00	3.1
C75-10	1132.8	**1132.80**	0.00	8.4	1135.61	0.25	8.6	1135.61	0.25	7.2	1137.59	0.42	6.1
P85-7	1855.55	**1855.55**	0.00	12.2	**1855.55**	0.00	12.4	**1855.55**	0.00	10.3	**1855.55**	0.00	6.3
D88-8	497.6	497.98	0.08	7.1	497.98	0.08	7.4	**497.60**	0.00	6.4	497.98	0.08	5.8
C100-10	1011.53	**1011.53**	0.00	13.7	1013.12	0.16	14.0	**1011.53**	0.00	11.4	**1011.53**	0.00	10.6
O117-14	12350.2	**12350.20**	0.00	17.4	**12350.20**	0.00	17.2	**12350.20**	0.00	15.3	**12350.20**	0.00	13.3
M134-8	5913.51	**5913.51**	0.00	28.5	5922.38	0.15	29.8	**5913.51**	0.00	24.8	5922.38	0.15	21.2
D150-10	44955.31	**44955.31**	0.00	46.6	**44955.31**	0.00	42.2	44960.92	0.01	37.2	**44955.31**	0.00	35.7
P318-4	4650.09[a]	4650.85[a]	0.00	199.7	4650.85[a]	0.00	144.9	4650.85[a]	0.00	151.3	4650.85[a]	0.00	105.0
P318-4-2	6435.73[b]	**6435.73**[b]	0.00	310.7	**6435.73**[b]	0.00	268.2	*5633.50*[b]	−0.11	259.9	*5633.50*[b]	−0.11	114.6
AV	72362.55	70938.49	0.00	34.5	70939.18	0.03	29.3	70896.69	0.01	28.0	70896.98	0.03	17.2
MD			0.00	3.9		0.00	4.00		0.00	3.4		0.00	3.1

Bold numbers are the best known solutions. Underscore and italic numbers are the obtained values less than the best known solutions. [a]Results minus 560000. [b]Results minus 700000.

Table 9: Results for the CLRPSPD instances in Prins set.

Instance	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
20-5-1	56350	**56350**	0.00	0.5	**56350**	0.00	0.5	**56350**	0.00	0.3	**56350**	0.00	0.3
20-5-1b	40678	**40678**	0.00	0.3	**40678**	0.00	0.3	**40678**	0.00	0.2	**40678**	0.00	0.2
20-5-2	49712	**49712**	0.00	0.5	**49712**	0.00	0.6	**49712**	0.00	0.3	**49712**	0.00	0.3
20-5-2b	38357	**38357**	0.00	0.3	**38357**	0.00	0.3	**38357**	0.00	0.2	**38357**	0.00	0.2
50-5-1	100240	**100240**	0.00	4.0	**100240**	0.00	4.1	**100240**	0.00	3.4	**100240**	0.00	3.1
50-5-1b	68993	**68993**	0.00	1.8	**68993**	0.00	2.3	**68993**	0.00	2.1	**68993**	0.00	1.8
50-5-2	94280	**94280**	0.00	5.1	**94280**	0.00	5.2	**94280**	0.00	4.4	**94280**	0.00	3.1
50-5-2b	69123	**69123**	0.00	2.7	**69123**	0.00	2.7	**69123**	0.00	2.2	**69123**	0.00	2.0
50-5-2bis	86349	**86349**	0.00	4.0	**86349**	0.00	4.0	**86349**	0.00	3.5	**86349**	0.00	3.3
50-5-2bbis	52356	**52356**	0.00	2.8	**52356**	0.00	2.6	**52356**	0.00	2.3	**52356**	0.00	2.1
50-5-3	90419	**90419**	0.00	4.9	*89737*	−0.75	4.7	**90419**	0.00	3.4	**90419**	0.00	3.8
50-5-3b	62178	**62178**	0.00	2.0	**62178**	0.00	2.1	**62178**	0.00	1.7	**62178**	0.00	1.6
100-5-1	290168	290280	0.04	43.5	290280	0.04	46.6	290401	0.08	38.7	290505	0.12	34.1
100-5-1b	219199	223968	2.18	18.9	223966	2.17	18.6	223968	2.18	16.9	223966	2.17	18.0
100-5-2	206049	206170	0.06	32.9	206078	0.01	36.7	206170	0.06	30.2	206308	0.13	29.8
100-5-2b	162077	**162083**	0.00	15.1	**162077**	0.00	15.7	**162077**	0.00	13.4	**162077**	0.00	12.1
100-5-3	209142	**209142**	0.00	43.2	**209142**	0.00	41.5	**209142**	0.00	34.5	**209142**	0.00	30.6
100-5-3b	156971	157027	0.04	16.2	157027	0.04	15.9	157027	0.04	14.4	157027	0.04	12.2
100-10-1	326093	**326093**	0.00	40.5	**326093**	0.00	40.8	**326093**	0.00	35.1	**326093**	0.00	30.8
100-10-1b	272889	**272889**	0.00	21.8	**272889**	0.00	21.6	**272889**	0.00	18.5	**272889**	0.00	14.1
100-10-2	253074	253609	0.21	39.4	253096	0.01	40.5	253640	0.22	34.1	**253074**	0.00	29.6
100-10-2b	208941	**208941**	0.00	20.9	**208944**	0.00	21.8	**208941**	0.00	18.9	**208941**	0.00	16.5
100-10-3	268137	268451	0.12	41.3	268416	0.10	39.0	268531	0.15	35.7	268349	0.08	32.6
100-10-3b	214828	**214828**	0.00	16.0	**214828**	0.00	17.9	**214828**	0.00	15.5	**214828**	0.00	13.5
200-10-1	529512	*498257*	−5.90	334.6	*498451*	−5.87	306.0	*499084*	−5.75	233.5	*498066*	−5.94	187.8
200-10-1b	393699	*384277*	−2.39	195.2	393924	0.06	170.2	*385287*	−2.14	155.7	*385737*	−2.02	135.7
200–10-2	463637	463968	0.07	316.4	463913	0.06	282.1	464111	0.10	249.2	464031	0.08	178.3
200-10-2b	380670	380924	0.07	206.4	380833	0.04	173.4	381031	0.09	138.5	*380611*	−0.02	117.0
200-10-3	485379	485882	0.10	307.9	485946	0.12	304.5	486009	0.13	240.8	486134	0.16	162.3
200-10-3b	369201	369938	0.20	183.3	370303	0.30	171.8	370213	0.27	131.5	370412	0.33	101.6
AV		206192	−0.17	64.1	206485	−0.12	59.8	206283	−0.15	49.3	206241	−0.16	39.3
MD			0.00	17.6		0.00	18.3		0.00	16.2		0.00	13.8

Bold numbers are the minimum value among four obtained solutions. Underscore and italic numbers are the obtained values less than the best known solutions.

Table 10: Results for the CLRPSPD instances in Tuzun and Burke.

Case	BKS	TS-SA			QS2-SA			AS-AM			PS-AM		
		Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU	Best	Gap	CPU
P1	1504.45	**1504.45**	0.00	18.8	**1504.45**	0.00	19.5	**1504.45**	0.00	16.5	1504.86	0.03	14.2
P2	1474.33	**1474.33**	0.00	19.5	**1474.33**	0.00	18.8	**1474.33**	0.00	17.2	**1474.33**	0.00	14.4
P3	1414.35	1414.46	0.01	17.6	1415.13	0.06	17.5	**1414.35**	0.00	15.4	**1414.35**	0.00	13.2
P4	1477.49	1481.50	0.27	20.3	1484.29	0.46	19.5	**1477.49**	0.00	18.2	**1477.49**	0.00	15.1
P5	1230	1233.69	0.30	18.9	1233.24	0.26	18.6	*1227.89*	−0.17	16.5	1233.54	0.29	14.7
P6	1123.72	**1123.72**	0.00	17.9	**1123.72**	0.00	19.2	**1123.72**	0.00	16.6	**1123.72**	0.00	12.2
P7	812.42	**812.42**	0.00	18.8	**812.42**	0.00	22.0	**812.42**	0.00	16.3	**812.42**	0.00	15.0
P8	752.66	*751.89*	−0.10	24.0	754.15	0.20	23.3	*752.41*	−0.03	19.0	759.21	0.87	11.6
P9	1273.3	**1273.30**	0.00	18.8	**1273.30**	0.00	19.2	**1273.30**	0.00	15.8	**1273.30**	0.00	14.5
P10	1265.1	**1265.10**	0.00	22.5	*1265.02*	−0.01	22.8	1266.20	0.09	20.0	**1265.10**	0.00	17.4
P11	911.52	**911.52**	0.00	20.3	**911.52**	0.00	18.4	**911.52**	0.00	16.8	**911.52**	0.00	15.4
P12	1033.38	*1031.94*	−0.14	23.5	**1033.38**	0.00	22.6	*1033.27*	−0.01	20.2	**1033.38**	0.00	18.1
P13	1943.41	1947.92	0.23	59.1	1946.58	0.16	58.5	1949.76	0.33	49.9	1947.98	0.24	41.9
P14	1848.69	**1848.69**	0.00	66.1	1856.10	0.40	67.4	1859.17	0.57	58.4	1862.25	0.73	46.4
P15	2011.29	2011.32	0.00	67.7	2011.39	0.00	71.3	2016.29	0.25	58.5	2012.29	0.05	49.1
P16	1856.03	1861.73	0.31	63.5	1859.97	0.21	69.8	1862.65	0.36	59.4	1859.32	0.18	47.5
P17	1479.77	1484.97	0.35	69.3	1490.83	0.75	69.3	1483.73	0.27	58.7	1491.36	0.78	48.9
P18	1452.49	1453.05	0.04	67.1	1453.58	0.08	68.4	*1452.28*	−0.01	58.6	1452.73	0.02	47.8
P19	1228.49	*1228.11*	−0.03	60.6	1228.52	0.00	61.3	**1228.49**	0.00	53.9	1228.69	0.02	47.5
P20	949.82	949.82	0.00	68.3	951.62	0.19	68.7	950.06	0.03	58.9	**949.82**	0.00	45.7
P21	1732.81	1736.52	0.21	60.9	1736.45	0.21	61.9	1737.80	0.29	53.2	1735.63	0.16	42.9
P22	1426.41	**1426.45**	0.00	76.6	1426.66	0.02	77.4	1427.57	0.08	64.6	*1425.37*	−0.07	57.6
P23	1238.78	*1229.69*	−0.73	70.0	*1226.08*	−1.03	69.6	*1229.69*	−0.73	59.0	*1229.69*	−0.73	49.4
P24	1166.8	**1166.80**	0.00	64.0	**1166.80**	0.00	66.3	**1166.80**	0.00	57.0	**1166.80**	0.00	51.4
P25	2338.89	2343.12	0.18	153.2	2339.54	0.03	146.9	2343.64	0.20	102.4	2351.63	0.54	76.7
P26	2224.86	2226.41	0.07	177.6	2240.00	0.68	173.0	2238.67	0.62	122.9	2228.80	0.18	115.1
P27	2293.85	2298.34	0.20	175.9	2298.04	0.18	178.3	2297.87	0.18	115.3	2302.65	0.38	112.1
P28	2292.91	2303.17	0.45	154.4	2303.63	0.47	147.1	2301.04	0.35	107.2	2307.62	0.64	95.8
P29	2159.62	2163.17	0.16	160.2	*2158.57*	−0.05	159.8	2159.83	0.01	110.2	2169.93	0.48	89.4
P30	1727.67	1731.95	0.25	175.1	1733.80	0.35	172.7	1729.77	0.12	132.2	1729.04	0.08	103.6
P31	1483.34	1486.36	0.20	172.0	1484.89	0.10	171.2	1484.87	0.10	116.8	1486.35	0.20	114.3
P32	1101.65	1102.31	0.06	149.0	1102.29	0.06	138.4	1103.93	0.21	96.7	1102.44	0.07	100.5
P33	2001.36	2002.82	0.07	154.9	2003.39	0.10	147.4	2002.79	0.07	129.5	2003.16	0.09	75.5
P34	1973.43	1975.63	0.11	164.1	1977.83	0.22	165.5	1981.09	0.39	115.5	1982.10	0.44	107.9
P35	1801.82	1810.44	0.48	177.9	1825.68	1.32	164.5	1826.61	1.38	130.4	1822.02	1.12	148.1
P36	1418.8	1425.63	0.48	151.5	1423.96	0.36	140.0	1422.24	0.24	123.9	1432.10	0.94	101.2
Av.		1541.47	0.10	83.3	1542.53	0.16	82.1	1542.44	0.14	63.9	1543.42	0.21	55.3
MD.			0.05	66.6		0.09	68.6		0.08	58.6		0.08	47.7

Bold numbers are the minimum value among four obtained solutions. Underscore and italic numbers are the obtained values less than the best known solutions. The instances are represented by "PN," where *N* represents the serial number.

BKSs, average and median values are, respectively, −0.17% and 0 for TS-SA, −0.12% and 0 for QS2-SA, −0.15 and 0 for third pair, and −0.16 and 0 for the last pair. From the perspective of gaps to BKSs, we verified the solutions achieved in Ref. [2]. Meanwhile, we obtained four improvements for four instances, i.e., 50-5-3 (QS2-SA), 200-10-1 (four pairs), 200-10-1b (four pairs except for QS2-SA), and 200-10-2b (PS-AM), one of them reaching an improvement of over 5.9% to previous BKSs.

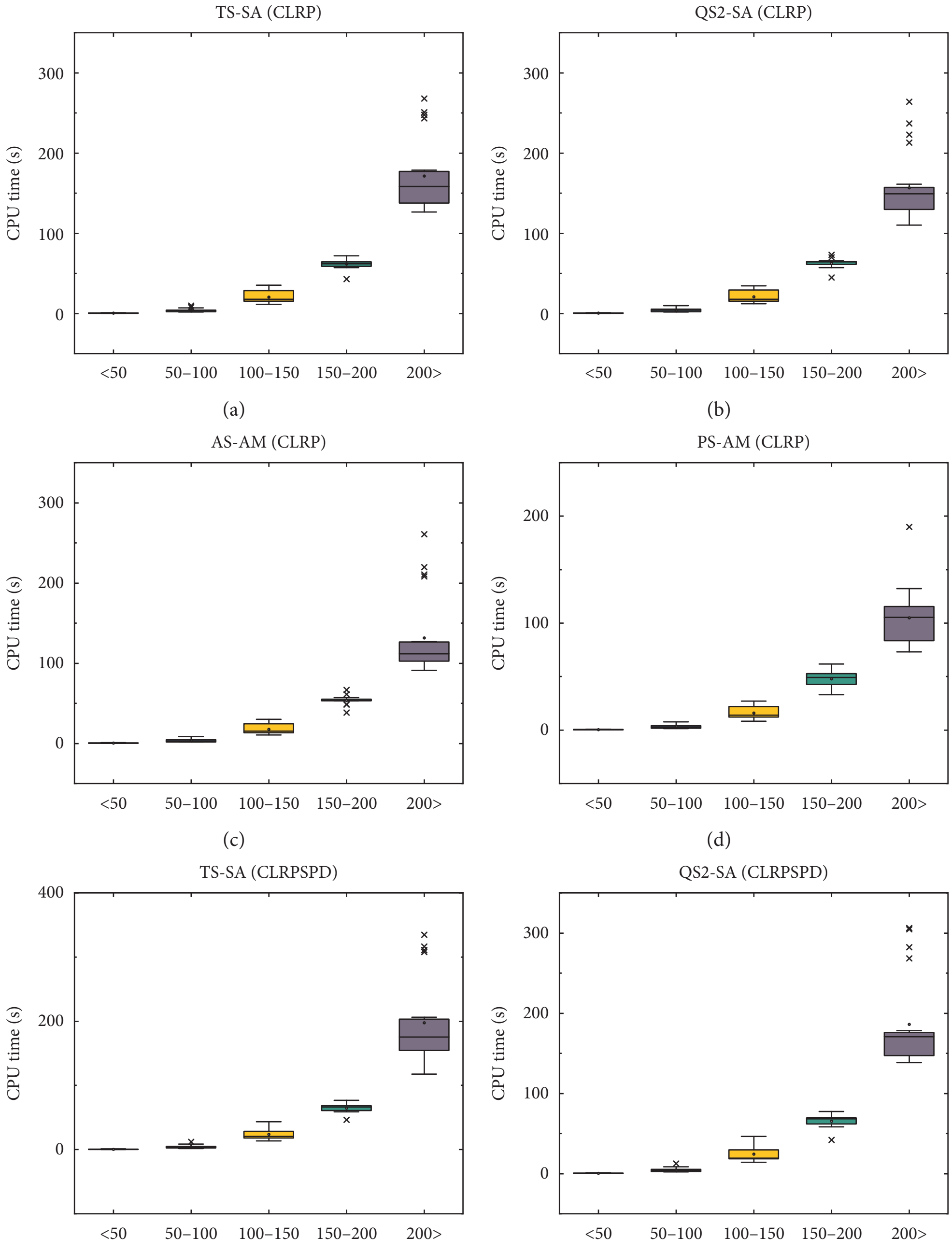
As shown in Table 10, the computing times are on average less than 84, 83, 64, and 56 s, and median values are 66.6, 68.6, 58.6, and 47.7 s, respectively. As far as gap to BKS is concerned, average/median values are 0.1/0.05%, 0.16/0.09%, 0.14/0.08%, and 0.21/0.08%, respectively. Meanwhile, new BKSs of nine instances were obtained by our proposed algorithms, and the largest improvement was over 1% for P23 obtained by QS2-SA. The AS-AM outperformed others obtaining five new BKSs, followed by TS-SA and QS2-SA, and PS-AM performed the worst. However, the best-performing pairs was TS-SA in terms of average gap to BKSs.

Statistical analysis is carried out for obtained results of three sets in Tables 8–10 through Table 11 based on Friedman test. In Table 11, Friedman statistics (χ^2) take the values 3.279 for Barreto et al., 4.621 for Prins et al., and 4.087 for Tuzun and Burke, which are, respectively, lower than 10.117, 18.493, and 23.269, indicating that there is no significant difference among the performance of four pairs. Therefore, we can strongly conclude that the proposed pairs could achieve high-quality solutions for the CLRPSPD benchmark instances.

Overall, among the four pairs, two types of discussion and conclusions are conducted: one for problem domains and one for hyperheuristic pairs. From perspective of both subjects, the amount of CPU time depends directly on the

Table 11: Friedman's test results for CLRPSPD benchmark instances ($\alpha = 0.05$).

Benchmark	Mean ranks for four pairs				Test statistics				Conclusion
	TS-SA	QS2-SA	AS-AM	PS-AM	n	χ^2	df	p	
Barreto set	2.39	2.68	2.37	2.55	19	3.279	3	0.351	$\chi^2 <$ critical value, accept H_0
Prins set	2.42	2.33	2.77	2.48	30	4.621	3	0.202	
Tuzun and burke	2.25	2.51	2.44	2.79	36	4.087	3	0.252	



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: Continued.

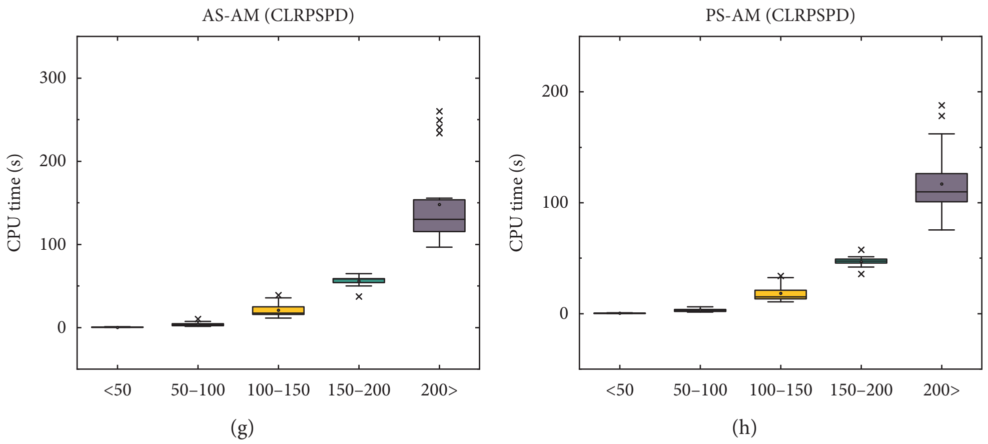


FIGURE 3: Box plots for relationship between computing time and number of clients.

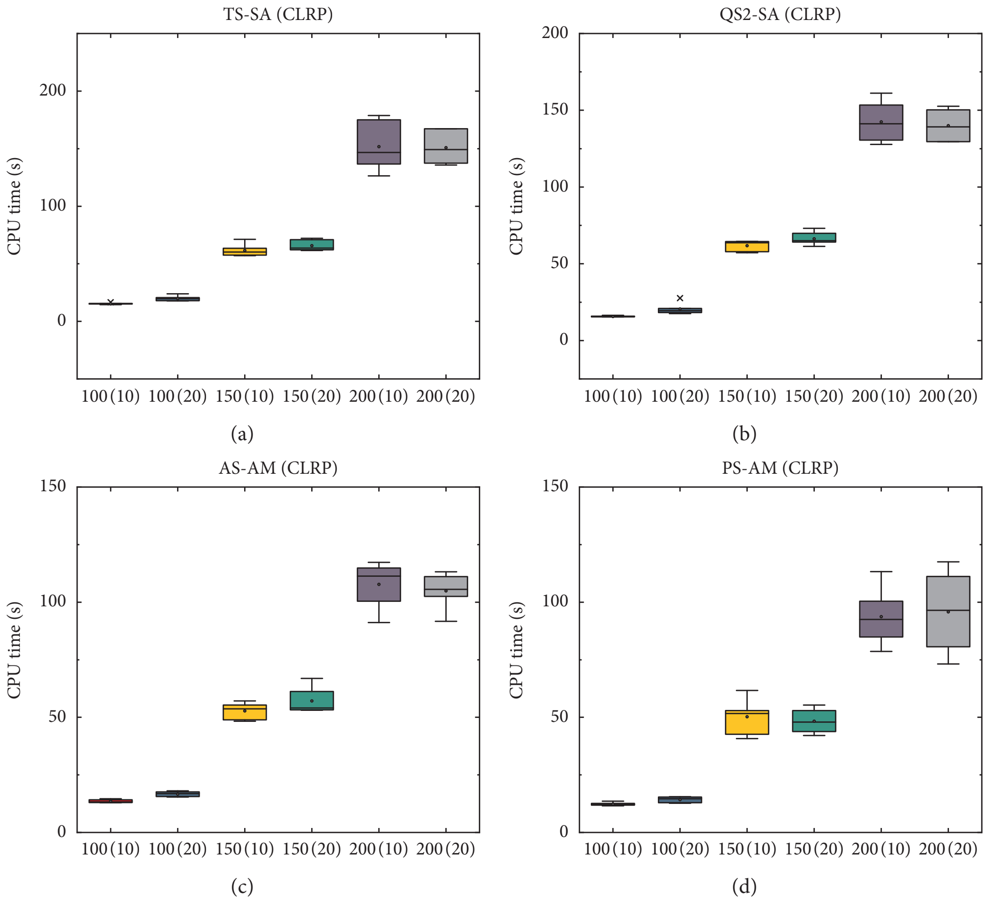
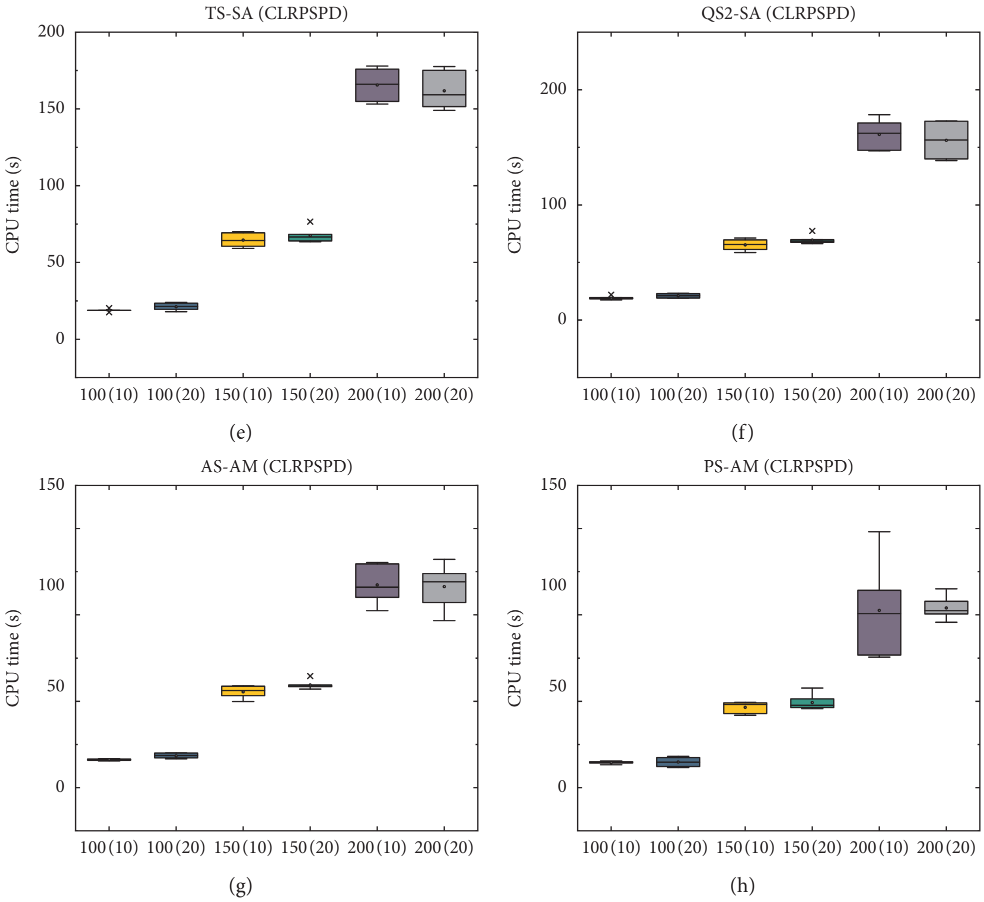


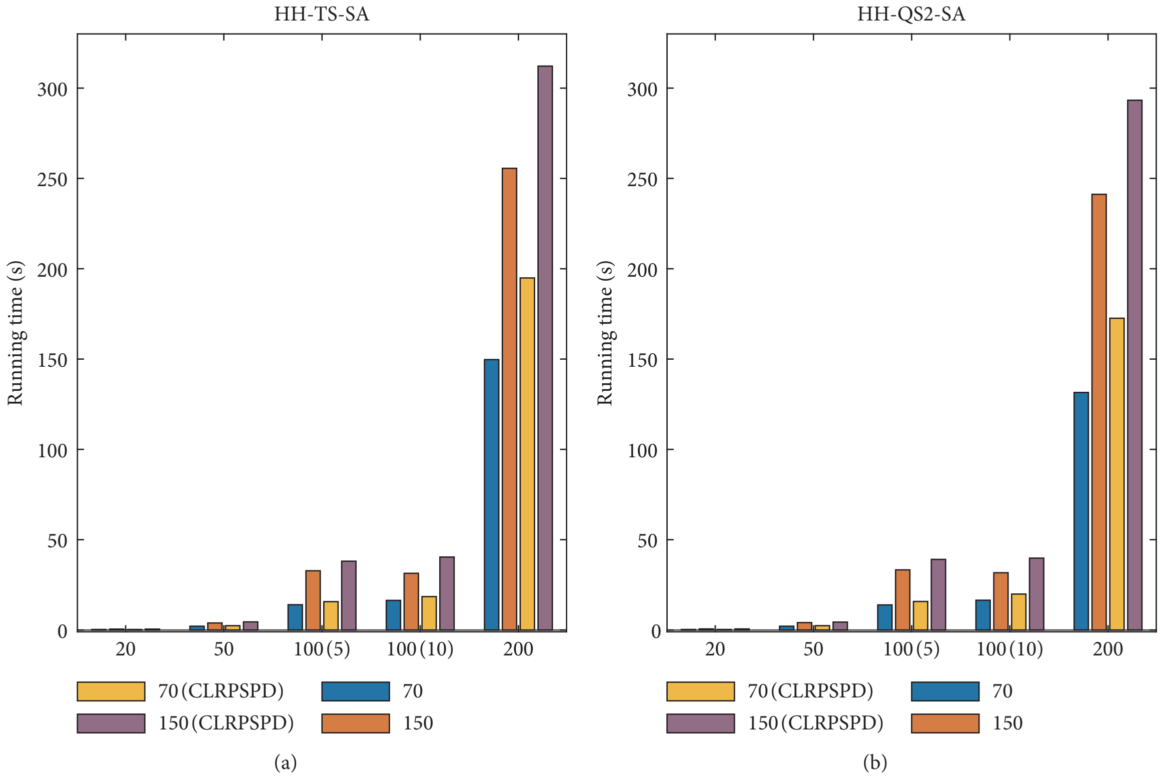
FIGURE 4: Continued.



(e) (f)

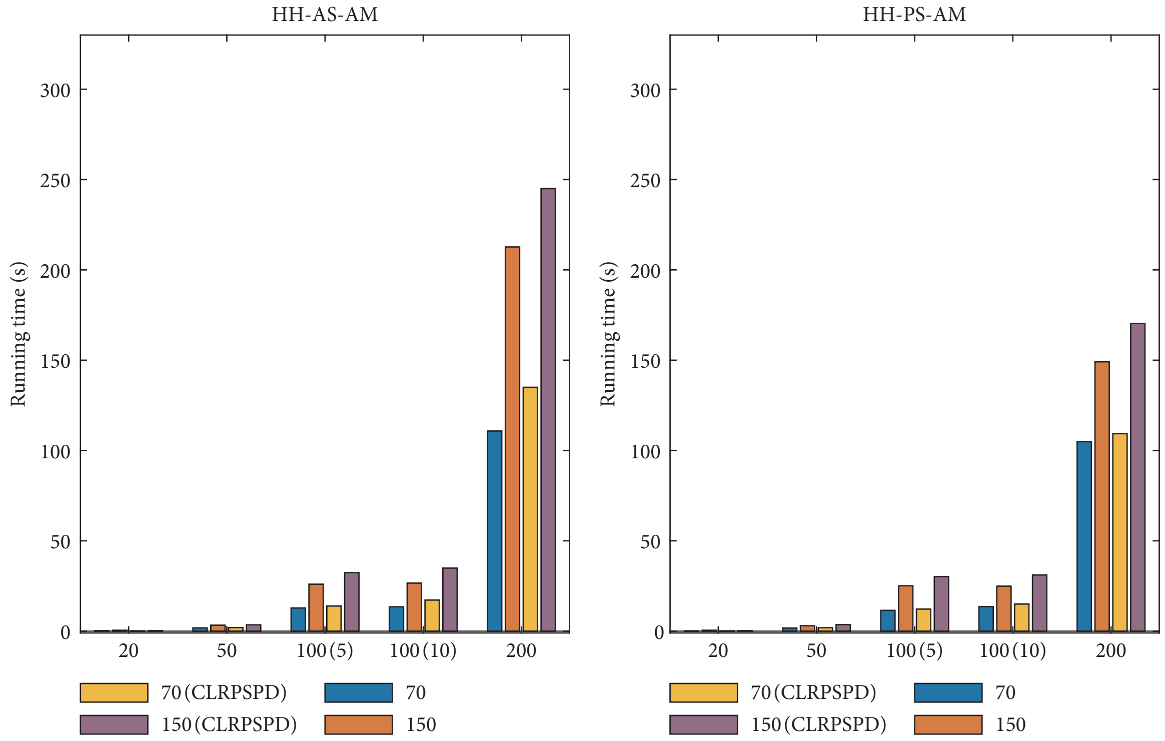
(g) (h)

Figure 4: Box plots for relationship between computing time and number of depots.



(a) (b)

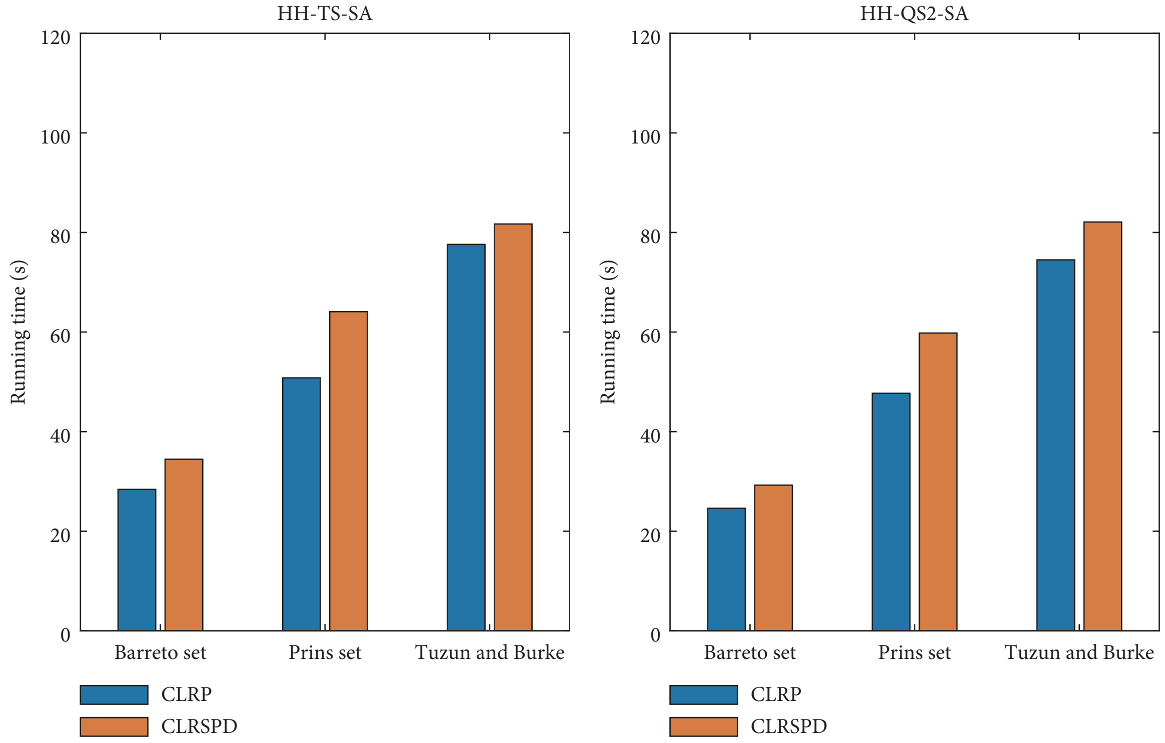
Figure 5: Continued.



(c)

(d)

Figure 5: Histogram of computing times of different vehicle capacity.



(a)

(b)

Figure 6: Continued.

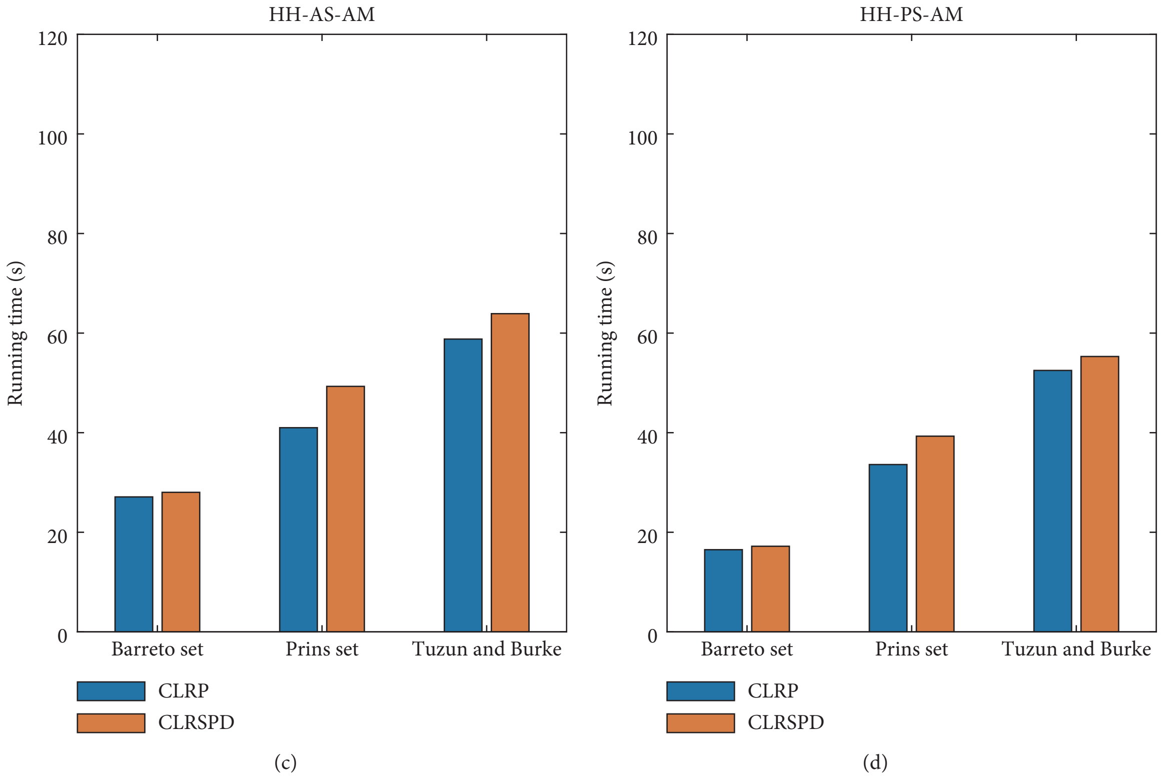


Figure 6: Histogram of computing times of both subjects.

number of clients (n), depots (m), and vehicle capacity (Q), and the relationship between running time and each of above factors is drawn in Figures 3 and 4 (excluding the outliers) and Figure 5. Several conclusions on running time are reached: (1) running time is affected by the number of clients in exponential way; (2) number of depots has slight impact on the running time, which may be resulted by few depots; (3) computing time increases with the decrease of the vehicle capacity in the form of polynomial; (4) the order of impact of the above factors effecting the running time is n, Q, and m. The first and third conclusions were also shared in Ref. [2]. Moreover, the time complexity and solution quality may depend in part on the complexity of the mathematical model. Figure 6 illustrates the impact of complexity of mathematical model on the running time. More specifically, the average computing time of tackling CLRPSPD instances increases by 4.2–21.5% for Barreto set, 30.6–46.8% for Prins set, and 5.3–10.2% for Tuzun and Burke, respectively, compared with the average running times of tackling CLRP cases. Concerning solution quality and average standard deviation, the complexity of instances is the direct acting factor consisting of the number of clients, depots and vehicles, and the distribution of clients and depots. Furthermore, the complexity of approaches or pairs of selection strategies and acceptance criteria also significantly influence the running time, standard deviation, and solution quality. PS-AM seems to be among the fastest with declining 36, 33, and 20% average running time compared to others, and AS-AM and QS2-SA rank the second and third in aspects of average running time. In terms of solutions quality, there is no significant difference among these four pairs concluded by the above Friedman tests, indicating that PS-AM and AS-AM outperform the others in obtaining a better trade-off solution quality and running time, but all pairs can obtain high solution quality within reasonable running time.

5.3.4. Comparison and Evaluation. To illustrate advantages of our proposed approaches for CLRP and CLRPSPD, comparisons with the most recent and effective methods in the literature are carried out in Table 12, where average gaps (AG) of best found solutions to BKS and average running time of the proposed approaches for the CLRP are provided, and they are, respectively, GRASP + ELS [31], SALRP/SALRP+ [32], ALNS/ALNS+ [33], MACO [35], HGTS [34], GRASP + ILP [25], GVTNS [36], and HGA/HGA+ [37], and for the CLRPSPD, the approaches were BC-SA [38], SA/SA* [4], and RP/TS/FRR-MAB/FMT [2].

Even though the time execution comparison can be unfair, depending on many factors (computer configuration, programming languages, the number of fitness evaluation, etc.), it is a significant performance indicator for judging the advantage of approaches. As shown in Table 12, our four pairs of hyperheuristic seem to be the fastest among the above methods. Furthermore, our approaches are the among best in aspects of solution quality on all CLRP and CLRPSPD benchmark sets with very small gaps. In conclusion, we can strongly conclude that our hyperheuristics are able to outperform the most recent and effective methods in terms of either solution quality or computing time.

Table 12: Average solutions for CLRP and CLRPSPD benchmark sets.

Algorithm	Barreto set		Prins set		Tuzun and Burke	
	AG	CPU	AG	CPU	AG	CPU
CLRP benchmark						
GRASP + ELS	0.08[a]	187.6[a]	1.11	258.2	1.30	606.6
SALRP	0.48	464.1	0.46	422.4	1.49	826.4
SALRP+	0.08	NA	0.08	NA	0.53	NA
ALNS	0.16[a]	177.2[a]	0.44	451.1	0.43	829.6
ALNS+	0.06[a]	1772.0[a]	0.27	4221.0	0.18	8103.0
HGTS	0.78[a]	105.2[a]	0.57	176.4	1.14	392.3
MACO	0.17	191.7	0.40	191.3	1.23	201.9
GRASP + ILP	0.14[a]	264.3[a]	0.12	1163.0	0.17	2589.5
GVTNS	0.67[a]	53.0[a]	0.37	91.2	0.76	201.2
HGA	0.63	42.4	0.37	73.1	0.86	86.0
HGA+	0.00	429.6	0.32	199.1	0.70	363.6
TS-SA	−0.04	28.4	0.06	50.8	−0.01	77.6
QS2-SA	−0.07	24.6	0.07	47.7	0.01	74.5
AS-AM	−0.05	27.1	0.07	41.0	0.03	58.8
PS-AM	−0.04	16.5	0.07	33.6	0.01	52.5
CLRPSPD benchmark						
BC-SA	1.47[a]	6604.1[a]	NA	NA	NA	NA
SA	0.48	1460.1	NA	NA	NA	NA
SA*	0.21	NA	NA	NA	NA	NA
RP	0.08	32.2	0.07	60.2	0.21	83.4
TS	0.02	30.54	0.06	55.2	0.10	77.6
FRR-MAB	0.01	32.46	0.02	62.7	0.08	77.3
FMT	0.00	32.73	0.01	59.8	0.02	78.6
TS-SA	0.00	34.5	−0.17	64.1	0.10	83.3
QS2-SA	0.03	29.3	−0.12	59.8	0.16	82.1
AS-AM	0.01	28.0	−0.15	49.3	0.14	63.9
PS-AM	0.03	17.2	−0.16	39.3	0.21	55.3

[a]Considering only (the same) 13 out of the 19 instances.

6. Conclusion and Future Work

In this study, evolution-based hyperheuristics are proposed to tackle the CLRP and one of its variants, namely, CLRPSPD. Five evolutionary and four nonevolutionary selection strategies are developed as high-level selection strategies to pair up with four acceptance criteria to make a right decision on constructing effective sequences of LLHs at each step, guiding the proposed approaches to achieve the optimal values.

The first experiment was carried out to determine the top four pairs which can rightly reflect to performance information of each LLH and quickly respond to select the promising LLHs. Results show that (1) AM and SA as acceptance criteria performed better when compared to the rest two; (2) AS, FRR-MAB-TS, and TS as selection strategies performed better when compared to the rest of the selection strategies; (3) average performance of AS-AM, PS-AM, QS2-SA, and TS-SA as HLHs perform significantly better than the rest of the pairs. The first four outstanding pairs were picked out to implement the remaining two experiments. PS-AM and AS-AM outperform others in terms of obtaining a good trade-off between solution quality and computing time, and all four pairs can achieve solution of high quality within reasonable running time, providing several new BKSs accounted for over 30% in Tuzun and Burke CLRP set. A comparative analysis is also conducted with tailored approaches in the published literature, and the results illustrate that the performance of our proposed approaches outperform the above methods in terms of both solution quality and computing time. As advocated in Ref. [93] for hyperheuristic, the proposed approaches have capability in exploiting good enough, soon enough, and cheap enough solutions.

The proposed hyperheuristic (36 pairs) has been already been implemented in a decision-support tool, and we refer the interested readers to consult us about the package. The next research content will focus on the adaptive selection of MHs and other selection strategies of hill climbers, e.g., choice function [5, 47], adaptive pursuit [94], and fair-share method [95]. For bringing CLRP/CLRPSPD closer to the reality, more practical constraints will also be taken into account in future works, such as time windows, dynamic demands, period delivery, and pickup.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (nos. 51875524 and 61873240) and National Natural Science Foundation of Zhejiang (no. Y19F030017).

References

[1] N. Bhusiri, A. G. Qureshi, and E. Taniguchi, "The trade-off between fixed vehicle costs and time-dependent arrival penalties in a routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 62, pp. 1–22, 2014.

[2] Y. Zhao, L. Leng, and C. Zhang, "A novel framework of hyper-heuristic approach and its application in location-routing problem with simultaneous pickup and delivery," *Operational Research*, 2019.

[3] L. Bouhafs, A. Hajjam, and A. Koukam, "A combination of simulated annealing and ant colony system for the capacitated location-routing problem," in *Proceedings of the 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 4251, pp. 409–416, Bournemouth, UK, October 2006.

[4] V. F. Yu and S.-Y. Lin, "Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing," *International Journal of Production Research*, vol. 54, no. 2, pp. 526–549, 2016.

[5] P. Cowling, G. Kendall, and E. Soubeiga, "A hyperheuristic approach to scheduling a sales summit," in *Proceedings of the Practice and Theory of Automated Timetabling III: Third International Conference*, vol. 2079, pp. 176–190, Constance, Germany, August 2001.

[6] K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A tabu search hyperheuristic strategy for t-way test suite generation," *Applied Soft Computing*, vol. 44, pp. 57–74, 2016.

[7] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.

[8] S. P. Chen, J. Yang, Y. Li, and J. F. Yang, "Multiconstrained network intensive vehicle routing adaptive ant colony algorithm in the context of neural network analysis," *Complexity*, vol. 2017, Article ID 8594792, 9 pages, 2017.

[9] Y. Zhou, L. S. Zhou, Y. Wang, Z. Yang, and J. Wu, "Application of multiple-population genetic algorithm in optimizing the train-set circulation plan problem," *Complexity*, vol. 2017, Article ID 3717654, 14 pages, 2017.

[10] Y. Y. Niu, Z. H. Yang, P. Chen, and J. H. Xiao, "A hybrid tabu search algorithm for a real-world open vehicle routing problem involving fuel consumption constraints," *Complexity*, vol. 2018, Article ID 5754908, 12 pages, 2018.

[11] R. Grosso, J. Muñuzuri, A. Escudero-Santana, and E. Barbadilla-Martín, "Mathematical formulation and comparison of solution approaches for the vehicle routing problem with access time windows," *Complexity*, vol. 2018, Article ID 4621694, 10 pages, 2018.

[12] C. Watson-Gandy and P. Dohrn, "Depot location with van salesmen—a practical approach," *Omega*, vol. 1, no. 3, pp. 321–329, 1973.

[13] F. Rahim and C. Sepil, "A location-routing problem in glass recycling," *Annals of Operations Research*, vol. 223, no. 1, pp. 329–353, 2014.

[14] S. Solak, C. Scherrer, and A. Ghoniem, "The stop-and-drop problem in nonprofit food distribution networks," *Annals of Operations Research*, vol. 221, no. 1, pp. 407–426, 2014.

[15] N. Asgari, M. Rajabi, M. Jamshidi, M. Khatami, and R. Z. Farahani, "A memetic algorithm for a multi-objective obnoxious waste location-routing problem: a case study," *Annals of Operations Research*, vol. 250, no. 2, pp. 279–308, 2017.

[16] C. Burkart, P. C. Nolz, and W. J. Gutjahr, "Modelling beneficiaries' choice in disaster relief logistics," *Annals of Operations Research*, vol. 256, no. 1, pp. 41–61, 2017.

[17] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[18] R. B. Lopes, C. Ferreora, B. S. Santo, and S. Barreto, "A taxonomical analysis, current methods and objectives on location-routing problems," *International Transactions in Operational Research*, vol. 20, no. 6, pp. 795–822, 2013.

[19] C. Prodhon and C. Prins, "A survey of recent research on location-routing problems," *European Journal of Operational Research*, vol. 238, no. 1, pp. 1–17, 2014.

[20] M. Drexl and M. Schneider, "A survey of variants and extensions of the location-routing problem," *European Journal of Operational Research*, vol. 241, no. 2, pp. 283–308, 2015.

[21] J. W. Escobar, R. Linfati, and W. A. Jaimes, "The capacitated location routing problem: review of literature," *Facultad De Ingeniería*, vol. 24, no. 39, pp. 85–98, 2015.

[22] R. Baldacci, A. Mingozzi, and R. Wolfler Calvo, "An exact method for the capacitated location-routing problem," *Operations Research*, vol. 59, no. 5, pp. 1284–1296, 2011.

[23] J.-M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler Calvo, "A branch-and-cut method for the capacitated location-routing problem," *Computers & Operations Research*, vol. 38, no. 6, pp. 931–941, 2011.

[24] C. Contardo, J.-F. Cordeau, and B. Gendron, "A computational comparison of flow formulations for the capacitated location-routing problem," *Discrete Optimization*, vol. 10, no. 4, pp. 263–295, 2013.

[25] C. Contardo, J.-F. Cordeau, and B. Gendron, "An exact algorithm based on cut-and-column generation for the capacitated location-routing problem," *Informs Journal on Computing*, vol. 26, no. 1, pp. 88–102, 2014.

[26] D. Tuzun and L. I. Burke, "A two-phase tabu search approach to the location routing problem," *European Journal of Operational Research*, vol. 116, no. 1, pp. 87–99, 1999.

[27] C. Prins, C. Prodhon, A. Ruiz, P. Soriano, and R. Wolfler Calvo, "Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic," *Transportation Science*, vol. 41, no. 4, pp. 470–483, 2007.

[28] S. Barreto, C. Ferreira, J. Paixão, and B. S. Santos, "Using clustering analysis in a capacitated location-routing problem," *European Journal of Operational Research*, vol. 179, no. 3, pp. 968–977, 2007.

[29] M. Schwardt and K. Fischer, "Combined location-routing problems—a neural network approach," *Annals of Operations Research*, vol. 167, no. 1, pp. 253–269, 2009.

[30] S. Salhi and G. Nagy, "Local improvement in planar facility location using vehicle routing," *Annals of Operations Research*, vol. 167, no. 1, pp. 287–296, 2009.

[31] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon, "A GRASP × ELS approach for the capacitated location-routing problem," *Computers & Operations Research*, vol. 37, no. 11, pp. 1912–1923, 2010.

[32] V. F. Yu, S.-W. Lin, W. Lee, and C.-J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010.

[33] V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," *Computers & Operations Research*, vol. 39, no. 12, pp. 3215–3228, 2012.

[34] J. W. Escobar, R. Linfati, and P. Toth, "A two-phase hybrid heuristic algorithm for the capacitated location-routing problem," *Computers & Operations Research*, vol. 40, no. 1, pp. 70–79, 2013.

[35] C.-J. Ting and C.-H. Chen, "A multiple ant colony optimization algorithm for the capacitated location routing problem," *International Journal of Production Economics*, vol. 141, no. 1, pp. 34–44, 2013.

[36] J. W. Escobar, R. Linfati, M. G. Baldoquin, and P. Toth, "A granular variable tabu neighborhood search for the capacitated location-routing problem," *Transportation Research Part B: Methodological*, vol. 67, no. 9, pp. 344–356, 2014.

[37] R. B. Lopes, C. Ferreira, and B. S. Santos, "A simple and effective evolutionary algorithm for the capacitated location-routing problem," *Computers & Operations Research*, vol. 70, pp. 155–162, 2016.

[38] I. Karaoglan, F. Altiparmak, I. Kara, and B. Dengiz, "A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery," *European Journal of Operational Research*, vol. 211, no. 2, pp. 318–332, 2011.

[39] I. Karaoglan, F. Altiparmak, I. Kara, and B. Dengiz, "The location-routing problem with simultaneous pickup and delivery: formulations and a heuristic approach," *Omega*, vol. 40, no. 4, pp. 465–477, 2012.

[40] V. F. Yu and S.-W. Lin, "Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery," *Applied Soft Computing*, vol. 24, pp. 284–290, 2014.

[41] E. A. Demircan-Yildiz, I. Karaoglan, and F. Altiparmak, "Two echelon location routing problem with simultaneous pickup and delivery: mixed integer programming formulations and comparative analysis," in *Proceedings of the International Conference on Computational Logistics*, pp. 275–289, Lisbon, Portugal, September 2016.

[42] J. Denzinger, M. Fuchs, M. Fuchs, and M. Fuchs, "High performance ATP systems by combining several AI methods," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 102–107, Nagoya, Japan, August 1997.

[43] E. K. Burke, M. Gendreau, M. Hyde et al., "Hyper-heuristics: a survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[44] M. Kalender, A. Kheiri, E. Ö.eir, and E. K. Burke, "A greedy gradient-simulated annealing selection hyper-heuristic," *Soft Computing*, vol. 17, no. 12, pp. 2279–2292, 2013.

[45] E. Ö.bMe, B. Bilgin, and E. E. Korkmaz, "A comprehensive analysis of hyper-heuristics," *Intelligent Data Analysis*, vol. 12, no. 1, pp. 3–23, 2008.

[46] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 309–325, 2015.

[47] J. H. Drake, E. Ö.ake, and E. K. Burke, "An improved choice function heuristic selection for cross domain heuristic search," in *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*, pp. 307–316, Taormina, Italy, September 2012.

[48] A. Nareyek, "Choosing search heuristics by non-stationary reinforcement learning," in *Proceedings of the 4th Metaheuristics International Conference*, vol. 86, pp. 523–544, Kyoto, Japan, August 2003.

[49] L. Han and G. Kendall, "An investigation of a tabu assisted hyperheuristic genetic algorithm," in *Proceedings of the Congress on Evolutionary Computation (CEC)*, pp. 2230–2237, Canberra, Australia, December 2003.

[50] G. Kendall and N. M. Hussin, "A tabu search hyperheuristic approach to the examination timetabling problem at the MARA University of Technology," in *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, vol. 3616, pp. 270–293, Pittsburgh, PA, USA, August 2004.

[51] F. Glover and M. Rafael, "Tabu search," in *Metaheuristic Procedures for Training Neutral Networks*, E. Alba and R. Marti, Eds., Springer, Boston, MA, USA, pp. 53–69, 2006.

[52] G. K. Koulinas and K. P. Anagnostopoulos, "A new tabu search-based hyper-heuristic algorithm for solving construction leveling problems with limited resource availabilities," *Automation in Construction*, vol. 31, no. 3, pp. 169–175, 2013.

[53] A. Fiahlo, L. D. Costa, M. Schoenauer, and M. Sebag, "Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms," in *Proceedings of the 3rd Conference on Learning and Intelligent Optimization*, vol. 5851, pp. 176–190, Trento, Italy, January 2009.

[54] J. D. Walker, G. Ochoa, M. Gendreau, and E. K. Burke, "Vehicle routing and adaptive iterated local search within the HyFlex hyper-heuristic framework," in *Proceedings of the International Conference on Learning & Intelligent Optimization*, pp. 265–276, Catania, Italy, January 2012.

[55] A. Strickler, J. A. Prado Lima, S. R. Vergilio, and A. T. R. Pozo, "Deriving products for variability test of feature models with a hyper-heuristic approach," *Applied Soft Computing*, vol. 49, pp. 1232–1242, 2016.

[56] X. Wu, P. Consoli, L. Minku, G. Ochoa, and X. Yao, "An evolutionary hyper-heuristic for the software project scheduling problem," in *Proceedings of the 14th International Conference on Parallel Problem Solving from Nature (PPSN)*, vol. 9921, pp. 37–47, Edinburgh, UK, September 2016.

[57] L. L. Leng, Y. W. Zhao, Z. Wang, H. W. Wang, and J. L. Zhang, "Shared mechanism-based self-adaptive hyperheuristic for regional low-carbon location-routing problem with time windows," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8987402, 21 pages, 2018.

[58] E. K. Burke, G. Kendall, R. F. J. O'Brien, D. Redrup, and E. Soubeiga, "An ant algorithm hyperheuristic," in *Proceedings of the Fifth Meta-Heuristics International Conference (MIC 2003)*, Kyoto, Japan, 2003.

[59] E. K. Burke, G. Kendall, D. L. Silva, R. F. J. O'Brien, and E. Soubeiga, "An ant algorithm hyper heuristic for the project presentation scheduling problem," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2263–2270, Edinburgh, Scotland, September 2005.

[60] A. Cuesta-Cañada, L. Garrido, and H. Terashima-Marín, "Building hyper-heuristics through ant colony optimization for the 2D bin packing problem," in *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pp. 654–660, Melbourne, Australia, September 2005.

[61] P. C. Chen, G. Kendall, and G. V. Berghe, "An ant based hyperheuristic for the travelling tournament problem," in *Proceedings of the IEEE Symposium on Computational Intelligence in Scheduling*, pp. 19–26, Honolulu, Hawaii, April 2007.

[62] Z. Ren, H. Jiang, J. Xuan, and Z. Luo, "Ant based hyper heuristics with space reduction: a case study of the p-median problem," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature*, pp. 546–555, Krakov, Poland, September 2010.

[63] F. C. Ergin, A. Ş. Uyar, and A. Yayimli, "Investigation of hyper-heuristics for designing survivable virtual topologies in optical WDM networks," in *Proceedings of the Conference on EvoApplications*, pp. 1–10, Torino, Italy, April 2011.

[64] F. C. Ergin, A. Yayimli, and A. S. Uyar, "Survivable cross-layer virtual topology design using a hyperheuristic approach," in *Proceedings of the 13th International Conference on Transparent Optical Networks*, pp. 1–4, Stockholm, Sweden, June 2011.

[65] B. Kiraz, A. S. Etaner-Uyar, and E. Özcan, "An ant-based selection hyperheuristic for dynamic environments," in *Proceedings of the Applications of Evolutionary Computation 16th European Conference*, pp. 626–635, Málaga, Spain, April 2012.

[66] Z. A. Aziz, "Ant colony hyper-heuristics for travelling salesman problem," in *Proceedings of the IEEE International Symposium on Robotics and Intelligent Sensors*, vol. 76, pp. 534–538, Langkawi, Malaysia, October 2015.

[67] L. Chen, H. Zheng, D. Zhang, and D. Li, "An ant colony optimization-based hyperheuristic with genetic programming approach for a hybrid flow shop scheduling problem," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 814–821, Sendai, Japan, May 2015.

[68] A. Ahmed, A. H. S. Bukhari, and I. A. Ismaili, "A comparative study on three hyperheuristic approaches for solving benchmark scheduling problems," *Sindh University Research Journal*, vol. 43, pp. 261–267, 2011.

[69] A. Ahmed, M. Ali, and A. Sajid, "Particle swarm based hyperheuristic for tackling real world examinations scheduling problem," *Australian Journal of Basic & Applied Sciences*, vol. 5, no. 10, pp. 1406–1413, 2011.

[70] G. Koulinas, L. Kotsikas, and K. Anagnostopoulos, "A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem," *Information Sciences*, vol. 277, no. 2, pp. 680–693, 2014.

[71] R. Aron, I. Chana, and A. Abraham, "A hyper-heuristic approach for resource provisioning-based scheduling in grid environment," *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1427–1450, 2015.

[72] S. Chen, Z. Li, B. Yang, and G. Rudolph, "Quantum-inspired hyper-heuristics for energy-aware scheduling on heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1796–1810, 2016.

[73] H. Terashima-Marín, P. Ross, C. J. Farías-Zárate, E. López-Camacho, and M. Valenzuela-Rendón, "Generalized hyper-heuristics for solving 2D regular and irregular packing problems," *Annals of Operations Research*, vol. 179, no. 1, pp. 369–392, 2010.

[74] J. Blazewicz, E. K. Burke, G. Kendall, W. Mruczkiewicz, C. Oguz, and A. Swiercz, "A hyper-heuristic approach to sequencing by hybridization of DNA sequences," *Annals of Operations Research*, vol. 207, no. 1, pp. 27–41, 2013.

[75] K. Z. Zamli, F. Din, G. Kendall, and B. S. Ahmed, "An experimental study of hyper-heuristic selection and acceptance mechanism for combinatorial t-way test suite generation," *Information Sciences*, vol. 399, pp. 121–153, 2017.

[76] F. Buttelle, L. Alfandari, C. Coti et al., "Fast machine reassignment," *Annals of Operations Research*, vol. 242, no. 1, pp. 133–160, 2016.

[77] E. K. Burke, G. Kendall, M. Mısır, and E. Özcan, "Monte Carlo hyper-heuristics for examination timetabling," *Annals of Operations Research*, vol. 196, no. 1, pp. 73–90, 2012.

[78] S. Abdul-Rahman, E. K. Burke, A. Bargiela, B. McCollum, and E. Özcan, "A constructive approach to examination timetabling based on adaptive decomposition and ordering," *Annals of Operations Research*, vol. 218, no. 1, pp. 3–21, 2014.

[79] E. K. Burke, R. Qu, and A. Soghier, "Adaptive selection of heuristics for improving exam timetables," *Annals of Operations Research*, vol. 218, no. 1, pp. 129–145, 2014.

[80] N. Pillay, "A review of hyper-heuristics for educational timetabling," *Annals of Operations Research*, vol. 239, no. 1, pp. 3–38, 2016.

[81] A. Kheiri, E. Özcan, and A. J. Parkes, "A stochastic local search algorithm with adaptive acceptance for high-school timetabling," *Annals of Operations Research*, vol. 239, no. 1, pp. 135–151, 2016.

[82] K. Chakhlevitch and P. Cowling, "Hyperheuristics: recent developments," *Studies in Computational Intelligence*, vol. 136, pp. 3–29, 2008.

[83] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, and E. Özcan, "A classification of hyperheuristic approaches," in *Handbook of Metaheuristics*, M. Gendreau and J. Y. Potvin, Eds., pp. 449–468, Springer, Boston, MA, USA, 2010.

[84] R. J. Marshall, M. Johnston, and M. Zhang, "Hyper-heuristic operator selection and acceptance criteria," in *Proceedings of the 15th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP)*, vol. 9026, pp. 99–113, 2015.

[85] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, pp. 417–431, Pisa, Italy, October 1998.

[86] S. Ropke and D. Pisinger, "An adaptive large neighbourhood search heuristic for pickup and delivery with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, 2006.

[87] C. Prins, C. Prodhon, and R. W. Calvo, "Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking," *4OR*, vol. 4, no. 3, pp. 221–238, 2006.

[88] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[89] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, 1965.

[90] M. W. P. Savelsbergh, "The vehicle routing problem with time windows: minimizing route duration," *ORSA Journal on Computing*, vol. 4, no. 2, pp. 146–154, 1992.

[91] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research*, vol. 41, no. 4, 1993.

[92] E. Angelelli and R. Mansini, "The vehicle routing problem with time windows and simultaneous pick-up and delivery," in *Quantitative Approaches to Distribution Logistics and Supply Chain Management*, A. Klose, M. G. Speranza, and L. N. Van-Wassenhove, Eds., Springer, Berlin,Heidelberg, Germany, pp. 249–267, 2002.

[93] E. K. Burke, G. Kendall, and E. Soubeiga, “A tabu-search hyperheuristic for timetabling and rostering,” *Journal of Heuristics*, vol. 9, no. 6, pp. 451–470, 2003.

[94] D. Thierens, “An adaptive pursuit strategy for allocating operator probabilities,” in *Proceedings of the Genetic & Evolutionary Computation Conference*, pp. 1539–1546, Washington, DC, USA, June 2005.

[95] S. Adriaensen, T. Brys, and A. Nowe, “Fair-share ILS: a simple state-of-the-art iterated local search hyperheuristic,” in *Proceedings of the 16th Genetic & Evolutionary Computation Conference*, pp. 1303–1310, Vancouver, BC, Canada, July 2014.