A Hybrid Multiobjective Memetic Algorithm for Multiobjective Periodic Vehicle Routing Problem With Time Windows

Jiahai Wang, *Member, IEEE*, Wenbin Ren, Zizhen Zhang, Han Huang, *Member, IEEE*, and Yuren Zhou

***Abstract*—Periodic vehicle routing problem with time windows (PVRPTWs) is an important combinatorial optimization problem that can be applied in different fields. It is essentially a multiobjective optimization problem due to the problem nature. In this paper, a typical multiobjective PVRPTW with five objectives is first defined and new nonsymmetric real-world multiobjective PVRPTW instances are generated. Then, a hybrid multiobjective memetic algorithm is proposed for solving multiobjective PVRPTW. In the proposed algorithm, a two-phase strategy is devised to improve the comprehensive performance in terms of the convergence and diversity. In this strategy, several extreme solutions near an approximate Pareto front (PF) are identified at Phase I, and then the approximate PF is extended at Phase II. The proposed algorithm is extensively tested on both real-world instances and traditional instances. Experiment results show that the proposed algorithm outperforms two representative competitor algorithms on most of the instances. The effectiveness of the two-phase strategy is also confirmed.**

***Index Terms*—Extreme solutions (ESs), hybrid local search, multiobjective optimization, periodic vehicle routing problem with time windows (PVRPTWs), two-phase strategy.**

I. Introduction

VEHICLE routing problem (VRP) is one of the most studied combinatorial optimization problems since it can be widely applied in different fields, including transportation, supply chain management, production planning, telecommunications, and others [1], [2]. VRPs include many variants with different constraints [3]–[5]. Periodic VRP (PVRP) can be regarded as a multiday VRP with pattern [6]. In this paper, a PVRP with time windows (PVRPTWs) is studied. PVRPTW also has many practical applications, including courier services, elevator maintenance and repair, vending machine replenishment, the collection of waste and the delivery of interlibrary loan material, and so on [6]. For new-developing applications, PVRPTW can model the last-mile deliveries of unmanned aerial vehicles or drones [7]. The wide applicability and versatility of PVRPTW has led to a vast body of literature addressing both novel applications and solution methods [8].

PVRPTW is NP-hard. Therefore, it is hardly possible to solve large size instances by exact methodologies. Most researchers have focused on metaheuristic. In previous studies, different methods [9]–[19] were proposed for single-objective PVRPTW, where the total travel distance is considered as a sole objective. In real-world situations, more objectives (sources of cost), including the number of vehicles, total travel distance, makespan, total waiting time, and total delay time, should be considered in PVRPTW [20]–[22]. Due to the constraints and the problem structure of PVRPTW, the optimization of one objective may lead to the deterioration of other objectives. Hence, PVRPTW is essentially a multiobjective optimization problem (MOP) [4], [20], [23]. To the best of our knowledge, no previous work utilizes the multiobjective optimization methods for PVRPTW, which motivates this paper.

This paper first defines a multiobjective PVRPTW (MOPVRPTW) with five objectives. These objectives are the number of vehicles, total travel distance, makespan, total waiting time, and total delay time. Then real-world instances are generated based on the data from a distribution company in Tenerife, Spain [23]. Finally, a hybrid multiobjective memetic algorithm (HMOMA) is proposed to solve MOPVRPTW. HMOMA consists of two phases to improve the convergence and the diversity, respectively. In Phase I, several extreme solutions (ESs) are generated by extremized crowded NSGA-II (EC-NSGA-II) [24]. In Phase II, the approximate Pareto front (PF) is extended by SPEA2 with shift-based density estimation (SPEA2+SDE) [25]. The parallel cell coordinate system (PCCS) [26] is adopted to manage the diversity of the approximate PF.

The contributions of this paper are threefold: 1) formulating a five-objective version of MOPVRPTW and introducing a set

Manuscript received November 11, 2017; revised April 25, 2018; accepted July 29, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61673403, Grant 71601191, and Grant U1611262, and in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education under Grant MSC-201606A. This paper was recommended by Associate Editor S. Mostaghim. *(Corresponding author: Jiahai Wang.)*

J. Wang is with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China, also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Sun Yat-sen University, Guangzhou 510275, China, and also with the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510275, China (e-mail: wangjiah@mail.sysu.edu.cn; wjiahai@hotmail.com).

W. Ren, Z. Zhang, and Y. Zhou are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China.

H. Huang is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. This includes tables with results not included in the paper. This material is 0.76 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

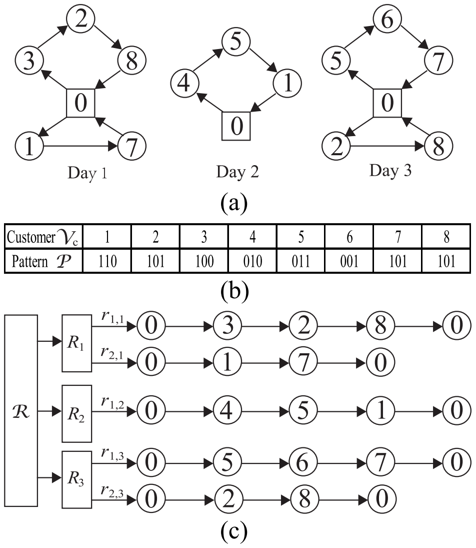
Digital Object Identifier 10.1109/TSMC.2018.2861879

TABLE I
OBJECTIVES

Objectives	Meanings
Number of vehicles (f_1)	the fixed costs of buying (or hiring) and maintaining vehicles.
Total travel distance (f_2)	the variable costs commonly estimated by using a function of the total distance.
Makespan (f_3)	the maximum working hours among all drivers. Minimizing the makespan enables a balanced workload.
Total waiting time (f_4)	the minimization of the waiting time, which is designed to improve work efficiency and avoid wasting working hours.
Total delay time (f_5)	the service cost related to the satisfaction of customers.

of realistic benchmark instances; 2) proposing an HMOMA to solve MOPVRPTW; and 3) designing a hybrid neighborhood structure for MOPVRPTW.

The remaining sections are organized as follows. Section II presents the formulation of MOPVRPTW, benchmark instances, and background. Section III introduces the proposed algorithm, HMOMA. Section IV presents the experimental results and Section V gives the conclusion.



Customer $\mathcal{V}_c$	1	2	3	4	5	6	7	8
Pattern $\mathcal{P}$	110	101	100	010	011	001	101	101

Fig. 1. Solution and its representation. (a) Solution with sequences of customers served on three days. (b) Pattern chromosome $\mathcal{P}$. (c) Route chromosome $\mathcal{R}$.

II. FORMULATION, INSTANCE, AND BACKGROUND

A. MOPVRPTW Formulation

In MOPVRPTW, a group of customers are to be serviced by vehicles within their given time windows during the planning horizon. Each vehicle has a maximum capacity and each customer has a demand of goods, a service time, a fixed number of visiting time, and a given pattern set (containing several allowable combinations of visiting days). Soft time windows are considered here [21], [27]–[29]. Hence, each customer has a maximum delay time. Five objectives (f_1–f_5) are considered in this paper [20]–[22], and shown in Table I.

Formally, MOPVRPTW can be defined as an optimization problem in a complete and directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, given as follows.

1) *Customer Set $\mathcal{V} = \{v_0, v_1, \ldots, v_n\}$:* Vertex v_0 represents the depot, and each vertex in $\mathcal{V}_c = \mathcal{V} - \{v_0\}$ represents a customer.
2) *Edge Set $\mathcal{E} = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$:* Each edge (v_i, v_j) has an associated travel time and travel distance.

To serve all customers in $\mathcal{V}$, the following constraints (C_1–C_4) should be satisfied.

1) *Pattern Constraint (C_1):* The customers' patterns should belong to their given pattern set.
2) *Vehicle Capacity Constraint (C_2):* The total demand of each route should not exceed the vehicle capacity.
3) *Maximum Delay Time Constraint (C_3):* Delay time should not exceed the maximum allowed delay time.
4) *Return Time Constraint (C_4):* Vehicles should return to the depot before the closing time.

The solution of MOPVRPTW is characterized by patterns and routes. It can be encoded into two chromosomes.

1) The pattern chromosome $\mathcal{P} = \{p_1, \ldots, p_n\}$ represents pattern-to-customer assignments, where p_i corresponds to ith customer's pattern for $\mathcal{T}$ days in the planning horizon. $\mathcal{P}$ is a binary vector of $n \times \mathcal{T}$ bits. Staring from the left, the ith sequence of $\mathcal{T}$ bits encodes the pattern assigned to the corresponding customer i.
2) The route chromosome $\mathcal{R} = \{R_1, \ldots, R_d, \ldots, R_{\mathcal{T}}\}$ encodes the routes designed for $\mathcal{T}$ days based on $\mathcal{P}$, where $R_d = \{r_{1,d}, \ldots, r_{j,d}, \ldots, r_{J,d}\}$ contains all routes on the dth day, and $r_{j,d}$ is the jth route on dth day.

An example solution is given in Fig. 1(a) for the problem with eight customers on three days. Fig. 1(b) illustrates the pattern chromosome of the solution, where the first customer is serviced according to pattern $p_1 = 110$, i.e., on day 1 and 2 (days are numbered from left to right). Fig. 1(c) shows the corresponding route chromosome. For each day in the planning horizon, a group of routes services customers on that day. A route is an ordered sequence of customers on one day.

The goal of MOPVRPTW is to find solutions with route set $\mathcal{R}$ and pattern $\mathcal{P}$ to serve all customers in $\mathcal{V}$ for $\mathcal{T}$ days, satisfying the constraints (C_1–C_4) and minimizing the five objectives (f_1–f_5) defined as follows.

1) *Number of Vehicles (f_1):*
$$f_1 = \sum_{d=1}^{\mathcal{T}} |R_d| \tag{1}$$
where $|R_d|$ is the number of routes in dth day.
2) *Total Travel Distance (f_2):*
$$f_2 = \sum_{d=1}^{\mathcal{T}} \sum_{j=1}^{|R_d|} \mathrm{Dist}_{j,d} \tag{2}$$
where $\mathrm{Dist}_{j,d}$ means the travel distance of $r_{j,d}$.
3) *Makespan (f_3):*
$$f_3 = \max\{T_{j,d} | j = 1 \ldots |R_d|, d = 1 \ldots \mathcal{T}\} \tag{3}$$
where $T_{j,d}$ denotes the travel time of $r_{j,d}$. $T_{j,d}$ is the sum of waiting time, service time, and travel time.

4) *Total waiting time (f_4):*

$$f_4 = \sum_{d=1}^{\mathcal{T}} \sum_{j=1}^{|R_d|} W_{j,d} \tag{4}$$

where $W_{j,d}$ represents the waiting time of $r_{j,d}$. Arriving at a customer earlier than its time windows will cause waiting time. Hence, $W_{j,d}$ can be calculated as the sum of waiting time of all customers in $r_{j,d}$.

5) *Total Delay Time (f_5):*

$$f_5 = \sum_{d=1}^{\mathcal{T}} \sum_{j=1}^{|R_d|} \mathrm{Dt}_{j,d} \tag{5}$$

where $\mathrm{Dt}_{j,d}$ denotes the delay time of $r_{j,d}$. Arriving at a customer later than its time windows will cause delay time. Hence, $\mathrm{Dt}_{j,d}$ can be calculated as the sum of delay time of all customers in $r_{j,d}$.

B. Real-World Instances

Traditional PVRPTW instances are symmetric and show weak correlations between different objectives, and thus are not suitable for conducting a proper multiobjective study [23]. Hence, new nonsymmetric MOPVRPTW instances are needed. Since real-world MOVRPTW instances show strong correlation between objectives [23], we derive new MOPVRPTW instances from the real-world MOVRPTW instances.

Specifically, real-world MOPVRPTW instances are generated from MOVRPTW instances provided in [23] according to the generation method described in [15]–[17]. The generation procedure is described as follows. First, the planning horizon is extended to four, six, and eight days, denoted as $d4$, $d6$, and $d8$, respectively. Then, the number of visiting time of customers is assigned. For $d4$, the customers need to be visited either 1, 2, or 4 times, for $d6$, either 1, 2, 3, or 6 times, and for $d8$, either 1, 2, 3, 4, or 8 times. Finally, the possible visiting patterns are evenly assigned to the customers at random.

C. Background of Multiobjective Optimization

An MOP can be stated as follows:

$$\text{Minimize} \quad F(x) = \{f_1(x), \ldots, f_m(x)\} \tag{6}$$

where $x = \{x_1, x_2, \ldots, x_D\} \in \Omega$ is a decision vector in the D-dimension decision space and $F = \{f_1, f_2, \ldots, f_M\} \in R^M$ is an image in the M-dimension objective space. There are some concepts of MOP as follows.

Definition 1 (Pareto Domination): Let $x, y \in \Omega$, y is said to Pareto dominate x iff $\forall i \in \{1, \ldots, m\}$ $f_i(y) \leq f_i(x)$ and $\exists j \in \{1, \ldots, m\}$ $f_j(y) < f_j(x)$, denoted as $y \prec x$.

Definition 2 (Pareto Optimal): A solution x' is Pareto optimal if there is no solution $x \in \Omega$ such that $x \prec x'$.

Definition 3 (Pareto Set): The set of the whole Pareto optimal solutions is called Pareto set (PS), denoted as $\mathrm{PS} = \{x \in \Omega \wedge x \text{ is Parto optimal}\}$.

Definition 4 (Pareto Front): The image set of PS is called Pareto front (PF), denoted as $\mathrm{PF} = \{F(x) \wedge x \in \mathrm{PS}\}$.

Since the objectives in (6) often conflict with each other, there exists no single solution, which minimizes all objectives simultaneously. Hence, the goal of an algorithm for an MOP is to seek a representative set of Pareto optimal solutions that perform well in terms of convergence and diversity.

Recently, a number of metaheuristics have been proposed to solve MOPs. Multiobjective evolutionary algorithms (MOEAs) strive to obtain a well-distributed approximated of the true PF [30]. Popular MOEAs mainly contain nondominated sorting genetic algorithm (NSGA-II) [31], strength Pareto evolutionary Algorithm 2 (SPEA2) [32], and MOEA based on decomposition (MOEA/D) [33]. Besides, local search-based algorithms are promising alternative approaches to solve multiobjective combinational optimization problems, such as multidirectional local search [34]. The merit of local search-based algorithm is that problem-specific knowledge can be directly used to guide the search toward PF. Local search is usually embedded within MOEAs, composing a new kind of algorithms called multiobjective memetic algorithm [35]. This kind of algorithms achieves good performance on multiobjective combinatorial optimization.

MOPs with more than three objectives are named as many-objective optimization problems. There are a number of challenges that must be addressed for solving many-objective optimization problems, like the dominance resistance phenomenon, the limited solution set size, the visualization of solution set in the objective space, and so on [36]. These challenges highlight the need for new algorithms that can efficiently handle the growing number of objectives.

III. HMOMA FOR MOPVRPTW

A. Motivation and Overview of HMOMA

An MOEA concentrates on achieving two basic and conflicting goals, convergence and diversity [37]. Such conflict has a detrimental impact on the optimization process of algorithm and can be aggravated in many-objective optimization [38]. Thus it is necessary to handle the balance of convergence and diversity during the process of an MOEA [37], [39], [40]. However, the performance in terms of convergence and diversity of a many-objective optimization algorithm is very difficult to reconcile, and it is still far from meeting the requirements of real-world many-objective optimization problems [36]. Most of existing MOEAs focus on convergence and diversity simultaneously at each generation, however, most of the efforts on diversity and uniformity spent at early generations in existing MOEAs are not very meaningful [37], [39].

In order to handle the balance well, a two-phase HMOMA, as shown in Fig. 2, is proposed to solve MOPVRPTW. In the proposed two-phase strategy, the convergence and diversity are separately emphasized at different phases by EC-NSGA-II [24] and a many-objective optimization algorithm, respectively. Specifically, several ESs near an approximate PF of MOPVRPTW are identified at Phase I by using EC-NSGA-II [24]. These ESs are well-converged and widely spreaded solutions of MOPVRPTW (not necessary to be the whole approximate PF), and form a coarse PF. The approximate PF of MOPVRPTW is then extended at Phase II from those ESs obtained at Phase I by using a many-objective

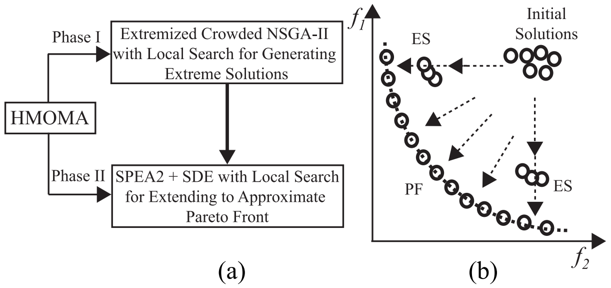

HMOMA
Phase I
Extremized Crowded NSGA-II with Local Search for Generating Extreme Solutions
Phase II
SPEA2 + SDE with Local Search for Extending to Approximate Pareto Front
(a)
f_1
f_2
ES
Initial Solutions
PF
ES
(b)

Fig. 2. (a) HMOMA framework. (b) Solution distributions of two phases.

optimization algorithm, SPEA2+SDE [25]. Then more solutions with a good diversity and uniformity can be extended at Phase II. Thus, the convergence and diversity of an approximate PF can be emphasized in different phases, respectively, in HMOMA. The proposed two-phase strategy is a novel method to improve the comprehensive performance in terms of the convergence and diversity.

Note that EC-NSGA-II is used at Phase I to form a coarse PF with good convergence and spread, while SPEA2+SDE is used at Phase II to refine whole approximate PF with good diversity and uniformity. That is, at the very beginning, search is only conducted along ESs to achieve fast convergence, followed by the many-objective optimization algorithm for approximating a more complete PF. Thus, EC-NSGA-II and SPEA2+SDE cannot be exchanged between two phases since they play different roles at different phases.

B. Phase I: EC-NSGA-II With Local Search for Generating Extreme Solutions

The goal of Phase I is to generate ESs by EC-NSGA-II [24]. Compared with NSGA-II, EC-NSGA-II alters crowding distance to emphasize objective-wise best and worse Pareto solutions, i.e., ESs. The procedure of Phase I is shown in Algorithm 1.

In EC-NSGA-II, the extremized crowding distance is proposed to emphasize the worse and the best solutions on every objective. In the extremized crowding distance, solutions on a particular front are first sorted based on each objective. A solution closer to either extreme objective (minimum or maximum objective values) gets a higher rank compared to that of an intermediate solution. Hence, the two ESs for each objective get a rank equal to N_s (number of solutions in the front), and the solutions next to these ESs get the rank $N_s - 1$ and so on. Furthermore, Fig. 3(a) shows extremized crowding distance calculation. For a solution, the ranks are assigned on each objective and the maximum value of assigned ranks is declared as extremized crowding distance, shown within brackets in Fig. 3(a). Fig. 3(b) shows an example of extremized crowding distance in a three-objective problem. The extremized distance will not only emphasize the ESs A, B, C, and D, but also solutions on edges AB and AD (respectively, having the smallest f_1 and f_2 values) and solutions near them. This approach can effectively find out the ESs and approximately draw the contour of PF in many-objective problem.

Algorithm 1: Phase I: EC-NSGA-II + Local Search

```
Input: population size ($N_I$) and max generation ($G_I$)
Output: population $P_I$ and archive $A$
$A = \emptyset$ ;
randomly generate an initial population $P_I$ of $N_I$ individuals;
generate two types of weight vectors $\Lambda^1, \ldots, \Lambda^{2M}$;
for $gen = 1$ to $G_I$ do
    $Q = \emptyset$;
    while $|Q| < |P_I|$ do
        choose $p, q$ from $P_I$ using binary tournament;
        $x_1 = PatternCr(p, q)$;        /* Pattern crossover */
        $x_2 = RouteCr(p, q)$;          /* Route crossover */
        $Q = Q \bigcup \{x_1, x_2\}$;
    end while
    $Q' = \emptyset$;
    while $|Q| > 0$ do
        for $i = 1$ to $2M$ do
            $x = \mathrm{argmin}_{x \in Q}\, g(x|\Lambda^i)$;
            $Q = Q - \{x\}$;
            $x' = LS_{\Lambda^i}(x)$;             /* Local search */
            update $A$ with $x'$;
            $Q' = Q' \bigcup \{x'\}$;
        end for
    end while
    $P_I$ = environmental selection from $P_I \bigcup Q'$ by fast nondominated
    sorting and extremized crowding distance;
end for
return $P_I$ and $A$;
```

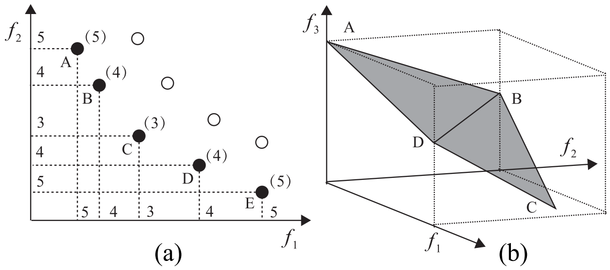


Fig. 3. Extremized crowding distance. (a) Extremized crowding distance calculation. (b) Extremized crowding distance in a three-objective problem.

However, it is hard to converge to ESs by genetic operator of EC-NSGA-II since MOPVRPTW is a complicated combinational optimization problem. Hence, local search is adopted to speed up convergence. Here, the weighted sum approach [35] is used to transfer MOPVRPTW into a single-objective optimization problem during local search procedure. The weighted sum function is defined as follows:

$$\min\, g\big(x|\Lambda^i\big) = \sum_{k=1}^{M} \lambda_k^i f_k(x) \tag{7}$$

where $\Lambda^i = (\lambda_1^i, \ldots, \lambda_M^i)$ is the weight vector. Since five objectives are of different scales, normalization is required.

Two types of weight vectors are designed to cooperate with EC-NSGA-II for converging to ESs. One is to optimize one objective while the other is to deteriorate one objective. The weight vectors are listed as follows:

$$\begin{bmatrix} \Lambda^1 \\ \vdots \\ \Lambda^M \\ \Lambda^{M+1} \\ \vdots \\ \Lambda^{2M} \end{bmatrix} = \begin{bmatrix} 1 & \epsilon & \dots & \epsilon \\ \epsilon & 1 & \dots & \epsilon \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon & \epsilon & \dots & 1 \\ \epsilon & \frac{1-\epsilon}{M-1} & \dots & \frac{1-\epsilon}{M-1} \\ \frac{1-\epsilon}{M-1} & \epsilon & \dots & \frac{1-\epsilon}{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-\epsilon}{M-1} & \frac{1-\epsilon}{M-1} & \dots & \epsilon \end{bmatrix} \tag{8}$$

where $\epsilon = 1e - 6$, approaching to zero. The details of local search would be introduced later.

C. Phase II: SPEA2 + SDE With Local Search for Extending to Approximate Pareto Front

The goal of Phase II is to obtain an approximate PF as accurately and uniformly as possible by SPEA2+SDE [25]. The procedure of Phase II is presented in Algorithm 2.

Experiments in [25] show that SPEA2+SDE is very competitive against other state-of-the-art MOEAs for many-objective optimization problems. In SPEA2, the fitness of solution is the sum of its strength raw fitness plus a density estimation [35]. Since MOPVRPTW is a many-objective problem, Pareto dominance relation-based primary selection criterion cannot effectively distinguish solutions. SDE [25] can maintain the distribution characteristic of solutions in the population and reflect the convergence of different solutions. SDE assigns high density to the poorly converged solutions by shifting the location of solutions. Therefore, these poorly converged solutions can be filtered by the density-based second selection criterion [41].

Local search is also embedded at Phase II. Since the goal of Phase II is to obtain an approximate PF, the weight vectors should be uniformly distributed, which is different from Phase I.

D. Solution Initialization

In Phase I, a solution is generated as follows. Randomly select a pattern from a given pattern set for each customer $c \in \mathcal{V}_c$ to construct $\mathcal{P}$ and then construct R based on assigned customers on each day. For constructing routes on each day, customers are first sorted in increasing order of the angle they make with the depot, attaining a sequence of customers $1, \dots, d_n$, where d_n is the number of customers in dth day. Then, a customer j is chosen randomly and the sequence $1, \dots, d_n$ is adjusted to the sequence $j, j+1, \dots, d_n, 1, \dots, j-1$. Finally, we make an empty route and customers are inserted into the route with the first feasible position sequentially. If a customer cannot be inserted into any existing route, a new route is created. This step is repeated until all customers are inserted.

The initial population of Phase II is the population in the last generation of Phase I.

Algorithm 2: Phase II: SPEA2 + SDE + Local Search

```
Input: population size ($N_{II}$) and max generation ($G_{II}$)
Output: archive $A$
$P_{II} = P_I$;
generate uniformly distributed weight vectors
$\Lambda^1, \dots, \Lambda^W$;
for $gen = 1$ to $G_{II}$ do
    evaluate the fitness of individuals in the population
    $P_{II}$;
    $Q = \emptyset$;
    while $|Q| < |P_{II}|$ do
        choose $p, q$ from $P_{II}$ using binary tournament;
        $x_1 = PatternCr(p, q)$;          /* Pattern crossover */
        $x_2 = RouteCr(p, q)$;    /* Route crossover */
        randomly choose $\Lambda^i, \Lambda^j$ from weight vectors;
        $x_1{}' = LS_{\Lambda^i}(x_1)$;          /* Local search */
        $x_2{}' = LS_{\Lambda^j}(x_2)$;
        update $A$ with $x_1{}', x_2{}'$;
        $Q = Q \bigcup \{x_1{}', x_2{}'\}$;
    end while
    $P_{II}$ = environmental selection from $P_{II} \bigcup Q$ by
    SPEA2+SDE;
end for
return $A$;
```

E. Crossover Operators

Based on the solution representation, two crossover operators are designed, which are pattern crossover operator (PatternCr) and route crossover operator (RouteCr). PatternCr aims at exploring different visit-day assignment while RouteCr combines existing route from different days.

PatternCr follows pattern-first-route-second scheme. Patterns for each customer are randomly selected from parents and then routes are constructed as in initialization for exploration.

RouteCr follows route-first-pattern-second scheme. Routes of children are randomly selected from their parents for each day and then the customers with infeasible pattern are repaired. For each infeasible customer, it is removed from routes on each day. Then, each pattern in the given pattern set is tried and customer is inserted into the best position based on weighted sum function. Finally, the best solution is saved.

PatternCr aims at exploring different visit-day assignments, which is the crossover operator for exploration. While RouteCr aims at combining existing routes from different days, which is the crossover operator for exploitation. Applied to the same pair of parents and generating two offspring, the former crossover creates new visit-day assignments, sometimes destroying useful information contained in the parents, while the latter crossover helps intensify the search by preserving routes or route components in the parents. These two crossover operators cooperate to balance exploration and exploitation, and thus to improve the search efficiency of the algorithm [19].

F. Archive Updating Scheme

An archive is used to collect nondominated solutions. As more and more nondominated solutions will be found during the search process, it is necessary to control the size of the archive. In this paper, PCCS [26] is used to estimate the density of nondominated solutions. The solution with the largest density will be removed when the size of archive exceeds its upper bound.

In PCCS, each solution x_i in the archive is assigned an identification array $\mathrm{B}_i = \{B_{i,1}, \ldots, B_{i,M}\}$, that is,

$$B_{i,m} = \left\lceil |A| \cdot \frac{f_m(x_i) - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right\rceil \tag{9}$$

where $f_m^{\max}$ is the maximum value on mth objective in the archive and $f_m^{\min}$ is the minimum value. Specially, $B_{i,m}$ is set to 1 if $f_m^{\max} = f_m^{\min}$.

The distance between two identification arrays is the sum of differences of cell coordinate over all objectives, called parallel cell distance (PCD). The PCD between two nondominated solutions x_i and x_j, $\mathrm{PCD}(x_i, x_j)$, can be calculated as

$$\mathrm{PCD}\big(x_i, x_j\big) = \begin{cases} 0.5, & \text{if } \forall m \; B_{i,m} = B_{j,m} \\ \sum\limits_{m=1}^{M} \left|B_{i,m} - B_{j,m}\right|, & \text{otherwise.} \end{cases} \tag{10}$$

With the PCD between each pair of solutions, the density of solution x_i can be defined as

$$\mathrm{Density}(x_i) = \sum_{j=1, j\neq i}^{|A|} \frac{1}{\mathrm{PCD}\big(x_i, x_j\big)^2}. \tag{11}$$

G. Local Search

Local search can effectively speed up the convergence of algorithm. Single neighborhood operator is easy to trap in local optima, while multiple neighborhood operators have more possibilities to escape from local optima. Therefore, multiple neighborhood operators are adopted and stored in the neighborhood pool, that is, $\mathrm{NPool} = \{N_1, N_2, \ldots, N_n\}$, where $\boldsymbol{N_i}$ represents the ith neighborhood operator. Specially, objective f_1 is the number of vehicles and it is difficult to reduce the number of vehicles by using regular neighborhood operators. Therefore, two well-designed neighborhood operators, LS_{R1} and LS_{P1}, are applied to optimize objective f_1. Algorithm 3 shows local search procedure, with I depths on weight vector $\Lambda^i = (\lambda_1^i, \ldots, \lambda_M^i)$.

Algorithm 3: Local Search

```
Input: The weight vector $\Lambda^i$, solution $x$ and archive $A$
Output: Solution $x$
if $\lambda_1^i == 1$ then
    /* Local search for objective 1    */
    $x' = LS_{R1}(x)$;
    update $A$ with $x'$;
    $x'' = LS_{P1}(x')$;
    update $A$ with $x''$;
    $x = x''$;
else
    /* Local search for other situations
       */
    for $depth = 1$ to $I$ do
        randomly choose $\boldsymbol{N_j}$ from NPool;
        $x' = \boldsymbol{N_j}(x, \Lambda^i)$;
        update $A$ with $x'$;
        if $g(x'|\Lambda^i) < g(x|\Lambda^i)$ then
            $x = x'$
        end if
    end for
end if
return $x$;
```

H. Neighborhood Operators

Neighborhood operators can be divided into inter-route operators and intraroute operators, as shown in Fig. 4. All neighborhood operators involve two basic functions: *selectRoute* and *bestPosition*. Function *selectRoute* randomly selects a route. Note that, the route with the longest trave time is selected when the weight vector is only to optimize objective f_3. Function *bestPosition* defines the best position when inserting a customer based on weighted sum function. The acceptance criterion is the best improvement strategy. Since PVRPTW involves $\mathcal{T}$ days, the neighborhood operator is used for each day. If there are no improvement on one day, the operator will be undone in that day.

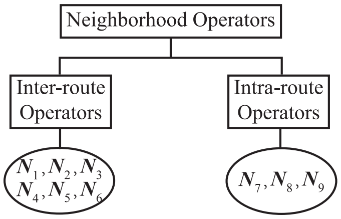


Fig. 4. Neighborhood operators.

$\boldsymbol{N_1}$ randomly removes a customer from a selected route, and then reinserts it into the best position among all possible positions of all possible routes.

$\boldsymbol{N_2}$ removes a random number of customers from a selected route. Then, these customers are reinserted into their best positions, respectively. As is shown in Fig. 5. Two customers, customer 2 and customer 3, are selected and then reinserted in Route 2 and Route 3 (a new route), respectively.

$\boldsymbol{N_3}$ exchanges a sequence of customers between two routes. A sequence of customers in one route, including all customers after a selected customer, is exchanged with all possible sequences on the other route. More details about $\boldsymbol{N_3}$ can be referred in [21] and [29].

$\boldsymbol{N_4}$ selects a random number of customers from the selected route, and then changes their patterns to new patterns randomly if there is any. The customers are removed and reinserted into their best positions based on new pattern.

$\boldsymbol{N_5}$ moves a customer segment from one selected route to another one. In Fig. 6, segment $\langle 2, 8\rangle$ of Route 1 is moved to Route 2.

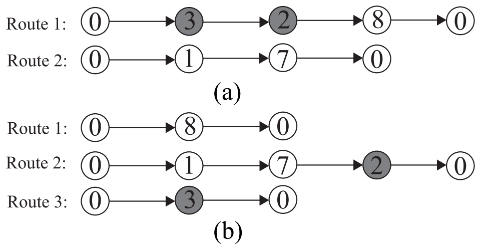


Fig. 5. N_2 operator. (a) Initial route. (b) Route after N_2.

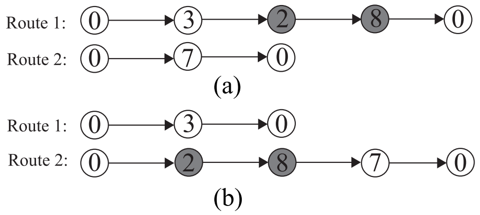


Fig. 6. N_5 operator. (a) Initial route. (b) Route after N_5.

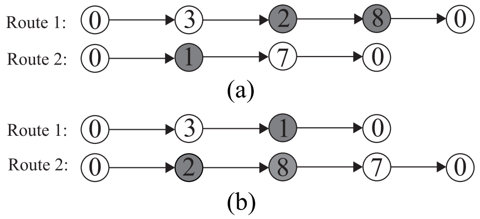


Fig. 7. N_6 operator. (a) Initial route. (b) Route after N_6.

N_6 exchanges a customer segment from the selected two routes. In Fig. 7, segment $\langle 2, 8\rangle$ and segment $\langle 1\rangle$ are exchanged between Route 1 and Route 2.

N_7 removes a customer segment and inserts into its best position in the same route.

N_8 swaps two customers in the selected route.

N_9 selects a segment of customers and reverses it.

The neighborhood operators above cannot directly reduce the number of vehicles. Hence, two specific neighborhood operators, LS_{R1} and LS_{P1}, are designed to optimize objective f_1, based on route and pattern, respectively. LS_{R1} changes the customers' locations to reduce the number of vehicles in one day while LS_{P1} changes the patterns of customers.

LS_{R1} selects the route which has the fewest customers first. All customers in the selected route are tried to insert into other routes with the first feasible location. If one vehicle is successfully reduced, LS_{R1} proceeds to reduce one more vehicle. LS_{R1} ends when a customer cannot be properly inserted into other routes.

LS_{P1} also selects the route which has the fewest customers first. The patterns of all customers in the selected route are changed. The customers are deleted and inserted into the first feasible position.

I. Feasibility Checking

During search process, only feasible solutions are considered. Pattern constraint and capacity constraint are easy to check in time complexity of $O(\mathcal{T})$ and $O(1)$, respectively. For time constraint, the slack time is proposed in [42] and [43]. The slack time $S_{c(i,j,d)}$ indicates the maximum time allowed to be late for the ith customer in jth route on dth day, which can be denoted as follows:

$$S_{c(i,j,d)} = \begin{cases} l_{c(i,j,d)} - a_{c(i,j,d)} & i = N_{k+1} \\ & d = 1, \ldots, \mathcal{T} \\ \min(l_{c(i,j,d)} + md - a_{c(i,j,d)}, & i = N_k, \ldots, 1 \\ \quad w_{c(i,j,d)} + S_{c(i+1,j,d)}) & d = 1, \ldots, \mathcal{T}. \end{cases} \tag{12}$$

When a customer is inserted between $(i-1)$th and ith vertices in jth route on dth day, the insertion is feasible if increment in $a_{c(i,j,d)}$ does not exceed $S_{c(i,j,d)}$. Hence, the time complexity can be reduced from $O(N)$ to $O(1)$. Furthermore, the route does not need to be reevaluated until it is changed.

J. Complexity Analysis

The running time of proposed algorithm mainly depends on local search. Local search consists of optimizing procedure and updating archive both with I depths for $\mathcal{T}$ days. Among all neighborhood operators, the maximum complexity one is ***N_6***, with a complexity of $O(\mathcal{T} \cdot I \cdot N^4)$. The complexity of updating archive is $O(\mathcal{T} \cdot I \cdot M \cdot K^2)$, where K is the upper bound of archive. Hence, the complexity of local search is $\max(O(\mathcal{T} \cdot I \cdot N^4), O(\mathcal{T} \cdot I \cdot M \cdot K^2))$.

IV. Experimental Results

A. Parameter Settings

HMOMA is implemented in C++. The weight vectors of Phase II are uniformly distributed. Hence, the number of weight vectors $W = C_{H+M-1}^{M-1}$ is controlled by a parameter H [33]. Since M is 5 and H is set to 8, $W = 495$. The capacity of the archive is set to 200. In local search, the depth I of local search is set to 10.

Because the instances are of difference scale, the stopping criterion should be different [19]. All instances can be classified into three cases: 1) small instances; 2) median instances; and 3) large instances based on the number of customers, whose customers' number is in (0,100), [100, 200), and [200, 288], respectively. For three cases, the population size is set to 70, 100, and 100, respectively, and the maximum number of generations is set to 500, 1500, and 2000, respectively. The population sizes of two phases are the same, that is $N_I = N_{II}$. Besides, there is a parameter, G_I/G_{II}, to determine the ratio of computing resources of two phases in HMOMA. $G_I/G_{II} = 1/3$ is set in this paper.

B. Benchmark Instances

The main aim of this paper is to propose a new algorithm framework, HMOMA, for the real-world five-objective MOPVRPTW instances generated in this paper. As pointed out in [23], traditional instances are not suitable for conducting a proper multiobjective study. Real-world instances simulate the conflicting nature of real life scenario and are more suitable for testing multiobjective optimization algorithms. In order to test the stability and robustness of HMOMA across data sets, it is also tested on the traditional instances.

1) Real-World Instances: There are 45 real-world MOVRPTW instances [23] with three sizes of customers, three types of the vehicle capacities and five time window profiles. By combining each instance with three kinds of the planning horizon (four, six, and eight days), we can generate 135 real-world MOPVRPTW instances. Each instance is named as *c*-*q*-*t*-*d*, where $c \in \{50, 150, 250\}$ indicates customer size, $q \in \{0, 1, 2\}$ represents three kinds (large, median, and small) of vehicle capacities, $t \in \{0, 1, 2, 3, 4\}$ represents five types of time windows, and $d \in \{4, 6, 8\}$ indicates the visit days. In this paper, a maximum delay of 30 min is allowed for each customer [23].

2) Traditional Instances: The dataset provided by Cordeau *et al.* [9] includes 20 instances, named as pr01–pr20. The maximum delay allowed for each customer i is set to 30% of the length of its time windows [44].

C. Competitor Algorithms

Since there is no existing algorithm for solving MOPVRPTW directly, we adapt two other algorithms as the competitor algorithms to investigate the performance of the proposed HMOMA, which are described as follows.

1) MOLS: MOLS is shown to be much competitive in solving variants of multiobjective VRP [21], [29]. When MOLS is adapted for MOPVRPTW, local search proposed in this paper are also applied to optimize different objectives. The same archive strategy, PCCS, is adopted in MOLS. Hence, MOLS can be seen as an existing state-of-the-art algorithm for comparison. The stop criterion of MOLS is the number of local search, where MOLS and HMOMA have the same number of local search for each instance.

2) SPEA2+SDE: When SPEA2+SDE is adapted for MOPVRPTW, the whole procedure is almost the same as the Phase II of HMOMA, except that the initial population of SPEA2+SDE is randomly generated. Comparing SPEA2+SDE with HMOMA, the effectiveness of two-phase strategy can be studied. For SPEA2+SDE, the population size and the maximum number of generations are set to the same values as HMOMA.

As a matter of fact, MOLS obtains better results than MOEA/D and NSGA-II for multiobjective VRP in [21] and [29]. SPEA2+SDE obtains better results than other state-of-the-art MOEAs including MOEA/D for many-objective optimization problems in [25]. Thus, MOLS and SPEA2+SDE can be seen as representatives of state-of-the-art algorithms for comparison.

D. Performance Indicators and Statistics

1) Performance Indicators: The convergence and diversity are the important standards for MOP algorithms. Two popular unary equality indicators, inverted generational distance (IGD) and hypervolume (HV), are used as performance indicators to evaluate them. A smaller value of IGD and a larger value of HV can be considered as a better set of solutions approximating the true PF from the convergence and diversity viewpoints. From the analysis of multiple runs, the unary quality indicators of each individual run are computed, and then the mean and the standard deviation of IGD and HV are reported. Since the true PF of MOPVRPTW is unknown, the final nondominated solutions obtained by all the algorithms in 30 runs are regarded as the true PF.

In order to further infer whether one nondominated set is actually better than another, a binary epsilon indicator, $I_{\epsilon+}$ [45], is used. Assume that $I_{\epsilon+}$ for the nondominated sets of A and B are denoted as $I_A = I_{\epsilon+}(A, B)$ and $I_B = I_{\epsilon+}(B, A)$, respectively. A pair of numbers ($I_A < 0$ and $I_B > 0$) indicates that A is strictly better than B. A pair of numbers ($I_A > 0$ and $I_B > 0$) indicates that A and B are incomparable. However, if $I_A < I_B$, A could be interpreted to be better than B *in a weaker sense* [46]. To summarize the results of the multiple runs, $I_{\epsilon+}$ for each pair of runs is computed in turn, and then the median and interquartile range (IQR) values of I_A and I_B are reported. The median and IQR, instead of the mean and standard deviation, are adopted [46].

2) Statistics by Wilcoxon and Friedman Test [47]–[49]: For IGD and HV, the Wilcoxon rank-sum test at 5% significance level is conducted to show the significant differences between the two algorithms on a single problem. The result of the test is also summarized as $w/t/l$, which means that HMOMA is significantly better than, equal to, and worse than the corresponding competitor on w, t, and l problems, respectively. The best mean values among HMOMA and its competitors are highlighted in boldface. To identify differences between a pair of algorithms on all problems, the multiproblem Wilcoxon signed-rank test is carried out. The Friedman test is used to obtain the rankings of multiple algorithms on all problems.

For $I_{\epsilon+}$, the Wilcoxon rank-sum test at 5% significance level is also conducted to decide whether the distribution of I_A values is significantly different from the distribution of I_B values. The result of the test is also summarized as $w/t/l$. To identify differences between I_A and I_B on all problems, the multiproblem Wilcoxon signed-rank test is also carried out.

Due to limited space, numerical values of performance indicators (HV, IGD, and $I_{\epsilon+}$) over 30 independent runs are presented in Tables S1–S4 of the supplementary material. Statistics summarizing those numerical values, including $w/t/l$ and ranking values, are shown in Tables II–V in this paper.

E. Performance on Real-World Instances

Table II provides the statistics summarizing all performance comparisons. In terms of IGD, HMOMA significantly outperforms MOLS in 73 instances and is outperformed by MOLS in 41 instances. HMOMA significantly outperforms SPEA2+SDE in 85 instances and is outperformed by SPEA2+SDE in 1 instances. In terms of HV, HMOMA significantly outperforms MOLS in 71 instances and is outperformed by MOLS in 32 instances. HMOMA significantly outperforms SPEA2+SDE in 74 instances and is outperformed by SPEA2+SDE in 19 instances. In terms of $I_{\epsilon+}$, HMOMA significantly outperforms MOLS in 66 instances and is outperformed by MOLS in 40 instances. HMOMA significantly

TABLE II
Statistics of Performance Comparisons of HMOMA With MOLS and SPEA2+SDE on Real-World Instances

IGD	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	73/21/41	5135.0	4045.0	$\geq$ 0.2	NO	NO
HMOMA vs SPEA2+SDE	85/49/1	8992.0	188.0	4.1253E-22	YES	YES

HV	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	71/31/32	5810.0	3370.0	0.0074	YES	YES
HMOMA vs SPEA2+SDE	74/42/19	7975.0	1205.0	1.0508E-13	YES	YES

$I_{\epsilon+}$	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	66/29/40	5630.5	3414.5	0.0147	YES	YES
HMOMA vs SPEA2+SDE	94/40/1	8753.5	426.5	6.1567E-20	YES	YES

TABLE III
Average Ranking of HMOMA, MOLS, and SPEA2+SDE By Friedman Test for Real-World Instances According to IGD and HV

IGD	Average ranking value	Final rank	HV	Average ranking value	Final rank
HMOMA	1.5037	1	HMOMA	1.7185	1
MOLS	2.0593	2	MOLS	2.0815	2
SPEA2+SDE	2.4370	3	SPEA2+SDE	2.2000	3

outperforms SPEA2+SDE in 94 instances and is outperformed by SPEA2+SDE in 1 instances.

From the multiproblem Wilcoxon signed-rank test in Table II, it is clear that HMOMA obtains higher $R+$ values than $R-$ values in all cases. It means that HMOMA is better than the competitors for all problems.

The final rankings of all algorithms by the Friedman test for all real-world instances are shown in Table III. Overall, HMOMA gets the first rank, followed by MOLS and SPEA2+SDE in terms of IGD and HV.

To visually demonstrate the performance of HMOMA, the projection of the final nondominated solutions obtained by three algorithms on a selected instance 150-2-1-6 in 30 runs at the f_1–f_3 and f_2–f_4 (approximated PF is in blue while true PF is in red) are shown in Fig. 8. It is very clear that HMOMA is the best in terms of both convergence and diversity on the selected problem, since the distribution of solutions in HMOMA spreads much wider and is nearer to the final PF than those in SPEA2+SDE and MOLS.

To further demonstrate the effectiveness of the two-phase strategy, the online search behavior of the algorithm on the selected instance 150-2-1-6 at f_1–f_3, and f_2–f_4 planes during a run is analyzed. The population distributions of HMOMA after initialization, Phase I, and Phase II are shown in Fig. 9. For comparison, the population distributions of SPEA2+SDE after corresponding generation are also shown in Fig. 9. In Fig. 9(a), populations of HMOMA and SPEA2+SDE have the same distributions at f_1–f_3 since the same initial populations are used. After Phase I, the distribution of solutions in HMOMA spreads much wider than that in SPEA2+SDE, as shown in Fig. 9(c). After Phase II, the distribution of solutions in HMOMA spreads much wider and is nearer to the final PF than that in SPEA2+SDE, as shown in Fig. 9(e). A similar observation can also be made at f_2–f_4 plane, as shown in Fig. 9(b), (d), and (f).

From the numeric and visual comparisons, we can conclude that HMOMA performs better than the other two algorithms in terms of both convergence and diversity. Comparing SPEA2+SDE with HMOMA, the effectiveness of two-phase strategy is also confirmed.

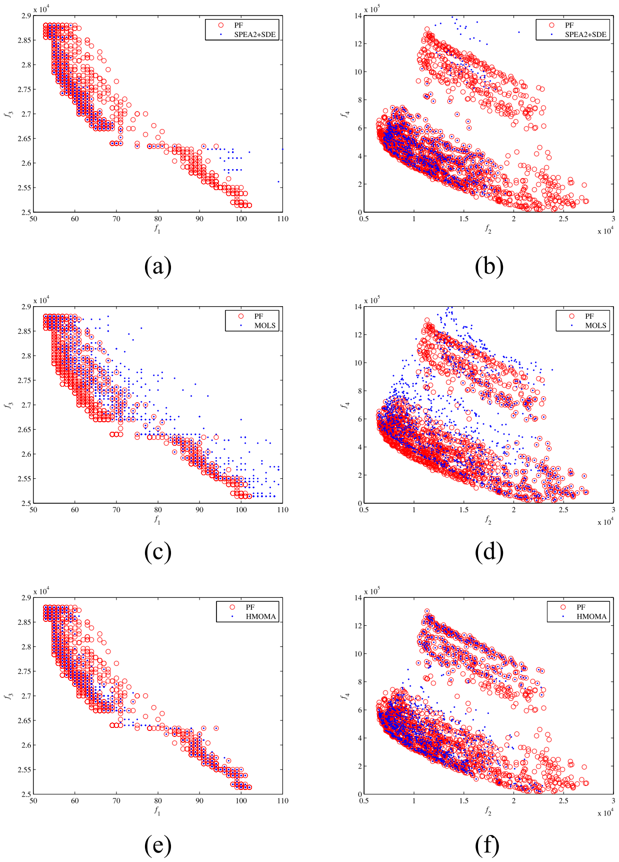


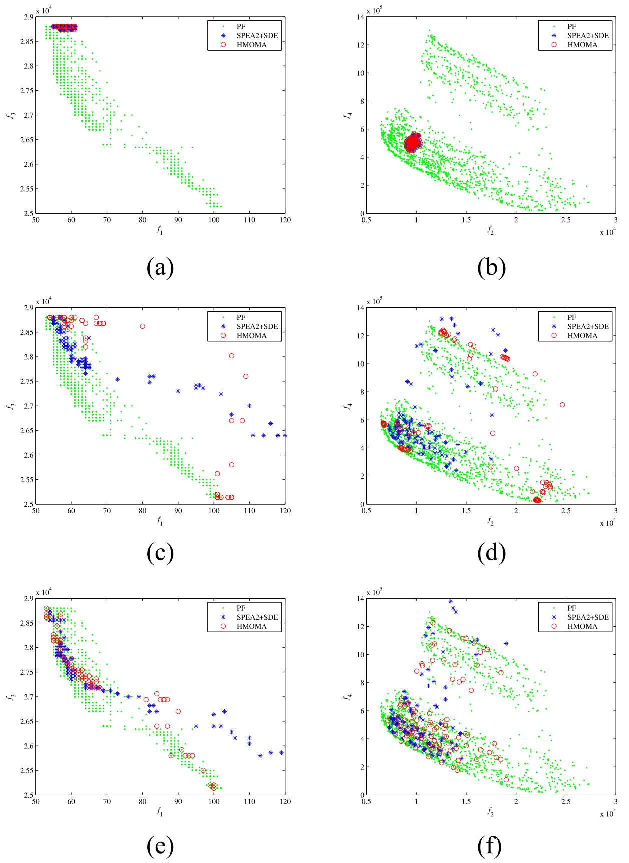
Fig. 8. Plots of nondominated solutions of the instance 150-2-1-6 by all the algorithms in the 30 runs. (a) SPEA2+SDE at f_1–f_3. (b) SPEA2+SDE at f_2–f_4. (c) MOLS at f_1–f_3. (d) MOLS at f_2–f_4. (e) HMOMA at f_1–f_3. (f) HMOMA at f_2–f_4.

F. Performance on Traditional Instances

Table IV provides the statistics summarizing all performance comparisons. In terms of IGD, HMOMA significantly outperforms MOLS in 20 instances (the whole instances). HMOMA significantly outperforms SPEA2+SDE in 11 instances and is outperformed by SPEA2+SDE in 2 instances. In terms of HV, HMOMA significantly outperforms MOLS in 18 instances and is not outperformed by MOLS. HMOMA significantly outperforms SPEA2+SDE in 5 instances and is outperformed by SPEA2+SDE in 7 instances. In terms of $I_{\epsilon+}$, HMOMA significantly outperforms MOLS in 16 instances and is outperformed by MOLS in 1 instances. HMOMA significantly outperforms SPEA2+SDE in 6 instances and is outperformed by SPEA2+SDE in 8 instances.

From the multiproblem Wilcoxon signed-rank test in Table IV, HMOMA obtains higher $R+$ values than $R-$ values compared to MOLS. Compared to SPEA2+SDE, HMOMA obtains higher $R+$ values than $R-$ values in terms of IGD, but obtains lower $R+$ values than $R-$ values in terms of HV and $I_{\epsilon+}$. Note that since p value is more than 0.2 in terms of HV and $I_{\epsilon+}$, there is no significant difference between HMOMA and SPEA2+SDE.

The final rankings of all algorithms by the Friedman test for all traditional instances are shown in Table V. Overall,



(a) (b)

(c) (d)

(e) (f)

Fig. 9. Plots of populations at different phases on instance 150-2-1-6 during a run with the same initial populations. (a) Initial populations at f_1–f_3. (b) Initial populations at f_2–f_4. (c) Populations after Phase I at f_1–f_3. (d) Populations after Phase I at f_2–f_4. (e) Populations after Phase II at f_1–f_3. (f) Populations after Phase II at f_2–f_4.

TABLE IV
Statistics of Performance Comparisons of HMOMA With MOLS and SPEA2+SDE on Traditional Instances

IGD	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	20/0/0	210	0	8.8575E-5	YES	YES
HMOMA vs SPEA2+SDE	11/7/2	152	58	0.0793	NO	YES

HV	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	18/2/0	209.0	1.0	1.0335E-4	YES	YES
HMOMA vs SPEA2+SDE	5/8/7	101.0	109.0	$\geq$ 0.2	NO	NO

$I_{\epsilon+}$	$w/t/l$	R+	R-	p-value	α = 0.05	α = 0.15
HMOMA vs MOLS	16/3/1	203.0	7.0	2.5360E-4	YES	YES
HMOMA vs SPEA2+SDE	6/6/8	94.0	96.0	$\geq$ 0.2	NO	NO

TABLE V
Average Ranking of HMOMA, MOLS, and SPEA2+SDE By Friedman Test for Traditional Instances According to IGD and HV

IGD	Average ranking value	Final rank	HV	Average ranking value	Final rank
HMOMA	1.40	1	SPEA2+SDE	1.55	1
SPEA2+SDE	1.60	2	HMOMA	1.60	2
MOLS	3.00	3	MOLS	2.85	3

HMOMA gets the first rank, followed by SPEA2+SDE and MOLS in terms of IGD. In terms of HV, SPEA2+SDE gets the first rank, followed by HMOMA and MOLS.

To visually demonstrate the performance of HMOMA, the projection of the final nondominated solutions obtained by three algorithms on a selected instance pr09 in 30 runs at f_1–f_3 and f_2–f_4 (approximated PF is in blue while true PF is in red) are shown in Fig. 10. HMOMA is better than MOLS in terms of both convergence and diversity on the selected problem. HMOMA is a little better than SPEA2+SDE in terms of diversity and convergence.

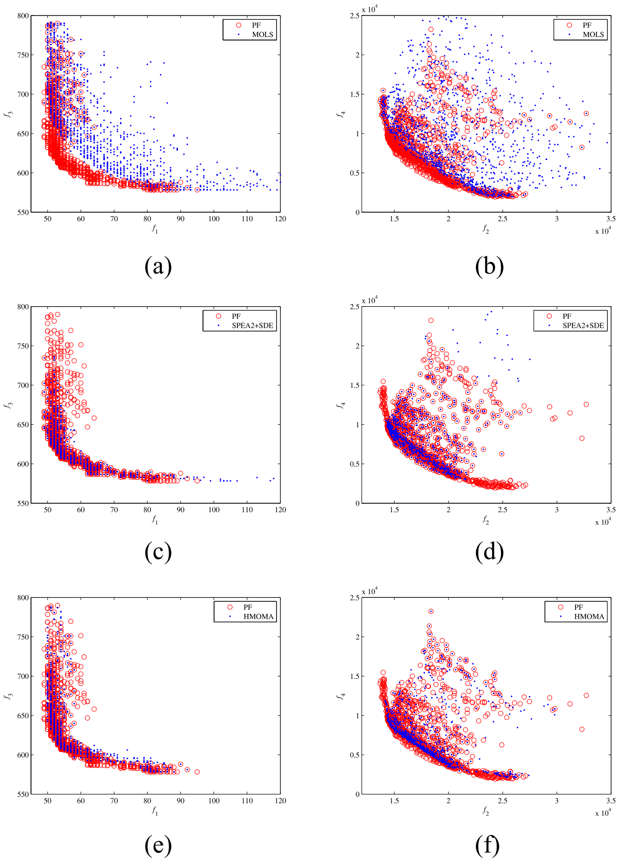
To further demonstrate the effectiveness of the two-phase strategy, populations of HMOMA and SPEA2+SDE on the selected instance pr09 at three different phases are also shown and compared. In Fig. 11(a), solutions of two populations have the same distributions at the f_1–f_3 since the same initial populations are adopted. After Phase I, the distribution of solutions in HMOMA spreads a little wider than those in SPEA2+SDE, as shown in Fig. 11(c). After Phase II, the distribution of solutions in HMOMA spreads much wider, but is further away from final PF than those in SPEA2+SDE as shown in Fig. 11(e). A similar observation can also be made at f_2–f_4 plane, as shown in Fig. 11(b), (d), and (f). Since traditional instances show weak correlation between different objectives, the distributions of the PF in traditional instances are not so wide as those in real-world instances. HMOMA takes a certain amount of computing resource to generating ESs in Phase I to spread along PF, which gets little benefit to search narrow PF of traditional instances. In the future, we will develop adaptive resource allocation scheme for Phase I in HMOMA according to the difficulty or characteristic of MOPs to improve the stability and robustness of two-phase strategy.

Note that HMOMA is obviously better than SPEA2+SDE in terms of IGD, but is slightly worse in terms of HV. A possible reason may be that the distribution of the reference front (which was obtained by all nondominated solutions of all algorithms over 30 runs) distorts the results of IGD since IGD totally depends on the given reference points [50]. IGD is calculated as the average distance from each reference point to the nearest solution in the solution set, which can be viewed as an approximate distance from the PF to the solution set in the objective space. Thus, IGD totally depends on the given reference points [50]. As shown in Fig. 10(c) and (d), it is obvious that the distribution of solution set obtained by SPEA2+SDE is different from the distribution of given reference points (PF). Specifically, at bottom-right region of Fig. 10(c), SPEA2+SDE obtains some solutions far from PF, and it misses the top-left region of PF. Similarly, at top-right region of Fig. 10(d), SPEA2+SDE obtains some solutions far from PF, and it misses the bottom-right region of PF.

G. Summary and Discussion

1) Summary: Experiments show that HMOMA outperforms the competitor algorithms on real-world instances. The effectiveness of the two-phase strategy is also confirmed.

On traditional instances, HMOMA performs better than MOLS in terms of both convergence and diversity. HMOMA is a little better than SPEA2+SDE in terms of diversity and convergence. The superiority of HMOMA on traditional instances is not so obvious as on real-world instances. Traditional instances show weaker correlation between different objectives than real-world instances do, and thus the distributions of PFs in traditional instances are not so wide as those in real-world

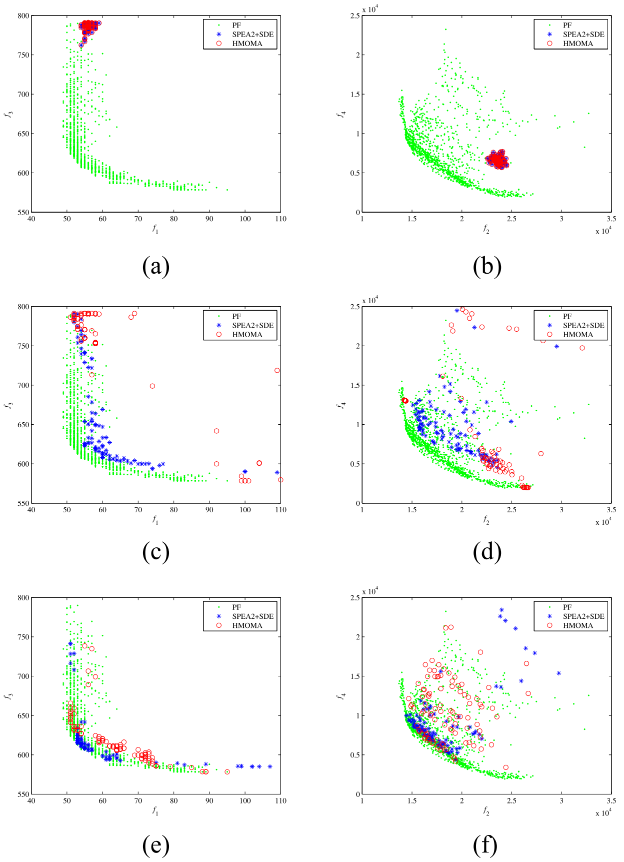


(a) (b)

(c) (d)

(e) (f)

Fig. 10. Plots of nondominated solutions of instance pr09 by all the algorithms in the 30 runs. (a) MOLS at f_1–f_3. (b) MOLS at f_2–f_4. (c) SPEA2+SDE at f_1–f_3. (d) SPEA2+SDE at f_2–f_4. (e) HMOMA at f_1–f_3. (f) HMOMA at f_2–f_4.



(a) (b)

(c) (d)

(e) (f)

Fig. 11. Plots of populations at different phases on instance pr09 with the same initial population (a) Initial populations at f_1–f_3. (b) Initial populations on f_2–f_4. (c) Populations after Phase I on f_1–f_3. (d) Populations after Phase I at f_2–f_4. (e) Populations after Phase II at f_1–f_3. (f) Populations after Phase II at f_2–f_4.

instances. Although ESs can be identified at Phase I, it has little benefit and wastes a certain amount of computing resource comparing to SPEA2+SDE. Traditional instances are not suitable for conducting a proper multiobjective study, as pointed out in [23]. Future researchers should test their algorithms on the proposed real-world instances to show usefulness in real life environment.

2) Effectiveness of PCCS: An advanced density estimation, PCCS [26], is adopted for archive maintenance in this paper. Unlike ϵ-dominance archive [29], PCCS has no additional parameter to be set and it is simpler. Crowding distance may provide an incorrect estimation of an individual's density when the number of objectives is larger than two due to the separate consideration of the neighbors on each objective [25]. The experimental results in [26] indicate that density estimation by PCCS is more effective on uniformity and convergence than that by adaptive grid and crowding distance.

Will the performance of HMOMA deteriorate if crowding distance is used to remove extra archive members? HMOMA with PCCS is compared with HMOMA with crowding distance, named HMOMA-CD, on 11 selected instances with different configurations. Tables S5 and S6 in the supplementary material provide the comparison results. In terms of IGD, HMOMA significantly outperforms HMOMA-CD in all 11 instances. In terms of HV, HMOMA significantly outperforms HMOMA-CD in 10 instances and is not outperformed by HMOMA-CD. In terms of $I_{\epsilon+}$, HMOMA significantly outperforms HMOMA-CD in 9 instances and is not outperformed by HMOMA-CD. Thus, the performance of HMOMA deteriorates if crowding distance is used instead of PCCS.

3) Effectiveness of Multiple Neighborhoods: The benefits of using multiple neighborhoods for complex (rich) VRPs [13], [15], [19], [51]–[53] and other complex combinatorial optimization problems [54]–[56] are widely studied and well established. Different structural neighborhood methods are used in multiple neighborhood scheme to broaden the exploration of the search space. Thus, multiple neighborhood scheme can overcome the myopic behavior and local minima problem of an approach that uses only one neighborhood. Further, with an arsenal of different neighborhoods implementing different types of perturbations, multiple neighborhood scheme is able to adapt to different problem types or different instances of a problem by flexible combination [51].

Will the performance of HMOMA deteriorate if only one neighborhood is used? HMOMA with multiple neighborhoods is compared with HMOMA with only one neighborhood N_8, named HMOMA-N_8, on 11 selected instances with different configurations. Neighborhood N_8 is selected since it is a mostly used neighborhood operator for VRPs [57]. Tables S5 and S6 in the supplementary material provide the comparison results. For all performance indicators (IGD, HV, and $I_{\epsilon+}$), HMOMA significantly outperforms HMOMA-N_8 in all 11 instances. Obviously, the performance of HMOMA deteriorates greatly if only one neighborhood is used.

H. Further Discussion About General Two-Phase Framework

In our previous study [58], a similar two-phase (stage) framework, called TS-MOEA, is proposed for five-objective multidepot VRP with time windows. In TS-MOEA, EC-NSGA-II with local search is adopted at Phase I for generating ESs, which is similar to HMOMA. At Phase II, MOEA/D and SPEA2+SDE are used for extending to approximate PF in TS-MOEA and HMOMA, respectively. MOEA/D and SPEA2+SDE are representatives of decomposition and Pareto-based algorithms for many-objective optimization, respectively. Simulation results of TS-MOEA and HMOMA show that two-phase strategy can improve the performance of both decomposition and Pareto-based algorithms, MOEA/D and SPEA2+SDE. Thus, the proposed two-phase framework can be seen as a more general framework, where any existing many-objective optimization algorithm, in principle, can be freely and easily introduced into Phase II. In the future, more existing many-objective optimization algorithms [36], [41], [59]–[62], for example, NSGA-III, can be incorporated into the two-phase framework for different MOPs [63]–[66].

Note that in TS-MOEA, weighted Tchebycheff is used as a scalarizing function, while in HMOMA, weighted sum is used. In fact, the weighted sum, weighted Tchebycheff, and penalty-based boundary intersection are the top three commonly used scalarizing functions [35], [67], [68]. In the two-phase framework, any existing scalarizing function, in principle, can be freely used in decomposition or local search for transforming an MOP into a number of single-objective optimization problems. Thus, weighted Tchebycheff and weighted sum are directly adopted in TS-MOEA and HMOMA, respectively. In fact, these scalarizing functions, respectively, have their own strengths and drawbacks [67], [68]. In view of the advantages and disadvantages of each scalarizing function, improved scalarizing functions [69], [70], ensemble of different scalarizing functions [71], [72], and new and more complex scalarizing functions [68] are proposed recently. However, the study with respect to scalarizing functions is beyond the scope of this paper. In the future, new scalarizing functions [68] proposed recently can be easily adopted in the proposed framework for possible improvement.

In the future, how to design more efficient algorithms considering characteristics of the practical problems and algorithms should be further investigated. Recent work [68] pointed out that decomposition-based methods are less robust than dominance-based ones using scalarizing functions, in particular for those problems with complex (nonuniform, discrete, and degenerated) PFs. The performance of different scalarizing functions is also compared [68]. However, most of previous studies about characteristics of algorithms are based on multiobjective continuous benchmark problems. In general, the PF characteristics of the continuous benchmark problems are known, while the PF characteristics of the practical discrete problems are unknown in advance. Therefore, characteristics of algorithms on practical many-objective combinatorial optimization problems, for example, MOPVRPTW, should be further studied [73].

V. Conclusion

This paper has defined a multiobjective variant of PVRPTW and generated a set of real-world benchmark instances. HMOMA has been proposed for MOPVRPTW. In HMOMA, a two-phase strategy is proposed to handle the balance of convergence and diversity. In this strategy, several ESs near an approximate PF are identified at Phase I and the approximate PF is extended at Phase II. Experimental results have shown that HMOMA outperforms the competitor algorithms on most of the instances. The effectiveness of the two-phase strategy is also confirmed.

In the future, this paper can be extended from multiple directions. First, the algorithm can be further improved by using advanced local search strategies, such as tabu search and large neighborhood search. Second, the proposed HMOMA, as a general framework, can be instantiated for solving other variants of multiobjective VRP [2], [22], scheduling problems [62], [63], and so on. The proposed real-world instances and the source code of HMOMA can be downloaded at https://www.researchgate.net/profile/Jiahai_Wang3/contributions.

References

[1] A. Hoff *et al.*, "Industrial aspects and literature survey: Fleet composition and routing," *Comput. Oper. Res.*, vol. 37, no. 12, pp. 2041–2061, 2010.

[2] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, "Survey of green vehicle routing problem: Past and future trends," *Exp. Syst. Appl.*, vol. 41, no. 4, pp. 1118–1138, 2014.

[3] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, 2009.

[4] N. Labadie and C. Prodhon, "A survey on multi-criteria analysis in logistics: Focus on vehicle routing problems," in *Applications of Multi-Criteria and Game Theory Approaches*. London, U.K.: Springer, 2014, pp. 3–29.

[5] R. Lahyani, M. Khemakhem, and F. Semet, "Rich vehicle routing problems: From a taxonomy to a definition," *Eur. J. Oper. Res.*, vol. 241, no. 1, pp. 1–14, 2015.

[6] P. M. Francis, K. R. Smilowitz, and M. Tzur, "The period vehicle routing problem and its extensions," in *The Vehicle Routing Problem: Latest Advances and New Challenges*. Boston, MA, USA: Springer, 2008, pp. 73–102.

[7] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[8] A. M. Campbell and J. H. Wilson, "Forty years of periodic vehicle routing," *Networks*, vol. 63, no. 1, pp. 2–15, 2014.

[9] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *J. Oper. Res. Soc.*, vol. 52, no. 8, pp. 928–936, 2001.

[10] J.-F. Cordeau, G. Laporte, and A. Mercier, "Improved Tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows," *J. Oper. Res. Soc.*, vol. 55, no. 5, pp. 542–546, 2004.

[11] J.-F. Cordeau and M. Maischberger, "A parallel iterated tabu search heuristic for vehicle routing problems," *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2033–2050, 2012.

[12] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Comput. Oper. Res.*, vol. 40, no. 1, pp. 475–489, 2013.

[13] S. Pirkwieser and G. R. Raidl, "A variable neighborhood search for the periodic vehicle routing problem with time windows," in *Proc. 9th EU Meeting Metaheuristics Logistics Veh. Routing*, Troyes, France, 2008, pp. 23–24.

[14] S. Pirkwieser and G. R. Raidl, "A column generation approach for the periodic vehicle routing problem with time windows," in *Proc. Int. Netw. Optim. Conf.*, vol. 2009, 2009, pp. 1–6.

[15] S. Pirkwieser and G. R. Raidl, "Boosting a variable neighborhood search for the periodic vehicle routing problem with time windows by ILP techniques," in *Proc. 8th Metaheuristic Int. Conf.*, Hamburg, Germany, 2009, pp. 33–35.

[16] S. Pirkwieser and G. R. Raidl, "Multiple variable neighborhood search enriched with ILP techniques for the periodic vehicle routing problem with time windows," in *Hybrid Metaheuristics*. Heidelberg, Germany: Springer, 2009, pp. 45–59.

[17] S. Pirkwieser and G. R. Raidl, "Matheuristics for the periodic vehicle routing problem with time windows," in *Proc. Matheuristics*, 2010, pp. 28–30.

[18] M. Maischberger and J.-F. Cordeau, "Solving variants of the vehicle routing problem with a simple parallel iterated tabu search," in *Network Optimization*. Heidelberg, Germany: Springer, 2011, pp. 395–400.

[19] P. K. Nguyen, T. G. Crainic, and M. Toulouse, "A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows," *J. Heuristics*, vol. 20, no. 4, pp. 383–416, 2014.

[20] N. Jozefowiez, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *Eur. J. Oper. Res.*, vol. 189, no. 2, pp. 293–309, 2008.

[21] Y. Zhou and J. Wang, "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1100–1113, Sep. 2015.

[22] G. Kim *et al.*, "City vehicle routing problem (city VRP): A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1654–1666, Aug. 2015.

[23] J. Castro-Gutierrez, D. Landa-Silva, and J. M. Pérez, "Nature of real-world multi-objective vehicle routing with evolutionary algorithms," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2011, pp. 257–264.

[24] K. Deb, S. Chaudhuri, and K. Miettinen, "Towards estimating nadir objective vector using evolutionary approaches," in *Proc. ACM 8th Annu. Conf. Genet. Evol. Comput.*, 2006, pp. 643–650.

[25] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[26] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, Feb. 2015.

[27] D. Taş, N. Dellaert, T. Van Woensel, and T. De Kok, "Vehicle routing problem with stochastic travel times including soft time windows and service costs," *Comput. Oper. Res.*, vol. 40, no. 1, pp. 214–224, 2013.

[28] H. Hashimoto, T. Ibaraki, S. Imahori, and M. Yagiura, "The vehicle routing problem with flexible time windows and traveling times," *Discr. Appl. Math.*, vol. 154, no. 16, pp. 2271–2290, 2006.

[29] J. Wang *et al.*, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.

[30] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[32] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Eurogen*, vol. 3242, 2001, pp. 95–100.

[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[34] F. Tricoire, "Multi-directional local search," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3089–3101, 2012.

[35] A. Jaszkiewicz, H. Ishibuchi, and Q. Zhang, "Multiobjective memetic algorithms," in *Handbook of Memetic Algorithms*. Heidelberg, Germany: Springer, 2012, pp. 201–217.

[36] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, p. 13, 2015.

[37] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45–65, Nov. 2015.

[38] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[39] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[40] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.

[41] J. Wang, W. Zhang, and J. Zhang, "Cooperative differential evolution with multiple populations for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2848–2861, Dec. 2016.

[42] G. A. Kindervater and M. W. Savelsbergh, "Vehicle routing: Handling edge exchanges," in *Local Search in Combinatorial Optimization*. Princeton, NJ, USA: Princeton Univ. Press, 1997, pp. 337–360.

[43] N. Labadie and C. Prins, "Vehicle routing nowadays: Compact review and emerging problems," in *Production Systems and Supply Chain Management in Emerging Countries: Best Practices*. Heidelberg, Germany: Springer, 2012, pp. 141–166.

[44] Z. Fu, R. Eglese, and L. Y. Li, "A unified tabu search algorithm for vehicle routing problems with soft time windows," *J. Oper. Res. Soc.*, vol. 59, no. 5, pp. 663–673, 2008.

[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[46] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[47] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[48] J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.

[49] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[50] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," in *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis. Making*, 2014, pp. 170–177.

[51] U. Derigs and M. Pullmann, "A computational study comparing different multiple neighbourhood strategies for solving rich vehicle routing problems," *IMA J. Manag. Math.*, vol. 27, no. 1, pp. 3–23, Jan. 2016.

[52] A. S. Azad, M. M. Islam, and S. Chakraborty, "A heuristic initialized stochastic memetic algorithm for MDPVRP with interdependent depot operations," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4302–4315, Dec. 2017.

[53] J. Li, P. Pardalos, H. Sun, J. Pei, and Y. Zhang, "Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups," *Exp. Syst. Appl.*, vol. 42, no. 7, pp. 3551–3561, 2015.

[54] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems," *Inf. Sci.*, vol. 191, pp. 146–168, May 2012.

[55] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, *Use of Heuristic Local Search for Single-Objective Optimization in Multiobjective Memetic Algorithms* (Lecture Notes in Computer Science), vol. 5199. Heidelberg, Germany: Springer, 2008, pp. 743–752.

[56] S.-Y. Wang and L. Wang, "An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 1, pp. 139–149, Jan. 2016.

[57] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, part I: Route construction and local search algorithms," *Transp. Sci.*, vol. 39, no. 1, pp. 104–118, 2005.

[58] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multi-depot vehicle routing problem with time windows," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2821180.

[59] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y. S. Ong, "A new decomposition-based NSGA-II for many-objective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 7, pp. 1191–1210, Jul. 2018.

[60] Y. R. Naidu and A. K. Ojha, "Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 820–832, Jun. 2018.

[61] W. Yuan, Y. Liu, H. Wang, and Y. Cao, "A geometric structure-based particle swarm optimization algorithm for multiobjective problems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 9, pp. 2516–2537, Sep. 2017.

[62] X.-L. Zheng and L. Wang, "A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 790–800, May 2018.

[63] X. Yu *et al.*, "Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2139–2153, Jul. 2018.

[64] C.-H. Chen and J.-H. Chou, "Multiobjective optimization of airline crew roster recovery problems under disruption conditions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 133–144, Jan. 2017.

[65] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[66] L. Ma *et al.*, "Two-level master–slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2723483.

[67] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, and T. Zhang, "On the effect of reference point in MOEA/D for multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 25–34, Sep. 2017.

[68] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, Apr. 2018.

[69] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3–18, Feb. 2018.

[70] X. Ma, Q. Zhang, J. Yang, and Z. Zhu, "On tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, Apr. 2018.

[71] H. Ishibuchi, N. Akedo, and Y. Nojima, *A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems* (Lecture Notes in Computer Science), vol. 7997. Heidelberg, Germany: Springer, 2013, pp. 231–246.

[72] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, Dec. 2016.

[73] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

Jiahai Wang (M'07) received the Ph.D. degree in computer science from the University of Toyama, Toyama, Japan, in 2005.

In 2005, he joined Sun Yat-sen University, Guangzhou, China, where he is currently a Professor with the Department of Computer Science. His current research interest includes computational intelligence and its applications.

Wenbin Ren received the M.S. degree in computer science from Sun Yet-sen University, Guangzhou, China, in 2017.

His current research interest includes multiobjective optimization for vehicle routing problems.

Zizhen Zhang received the B.S. and M.S. degrees in computer science from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2007 and 2009, respectively, and the Ph.D. degree in management science from the City University of Hong Kong, Hong Kong, in 2014.

He is currently an Associate Professor with Sun Yat-sen University. His current research interest includes computational intelligence and its applications in production, transportation, and logistics.

Han Huang (M'15) received the B.Man. degree in applied mathematics and the Ph.D. degree in computer science from the South China University of Technology (SCUT), Guangzhou, China, in 2002 and 2008, respectively.

He is currently a Professor with the School of Software Engineering, SCUT. His current research interests include evolutionary computation, swarm intelligence, and their application.

Dr. Huang is a Senior Member of CCF.

Yuren Zhou received the B.Sc. degree in mathematics from Peking University, Beijing, China, in 1988 and the M.Sc. degree in mathematics and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 1991 and 2003, respectively.

He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include design and analysis of algorithms, evolutionary computation, and social networks.