Multiobjective Multiple Neighborhood Search Algorithms for Multiobjective Fleet Size and Mix Location-Routing Problem With Time Windows

Jiahai Wang, *Member, IEEE*, Liangsheng Yuan, Zizhen Zhang, Shangce Gao, *Senior Member, IEEE*, Yuyan Sun, and Yalan Zhou

***Abstract*—This paper introduces a multiobjective fleet size and mix location-routing problem with time windows and designs a set of real-world benchmark instances. Then, two versions of multiobjective multiple neighborhood search algorithms based on decomposition and vector angle are developed for solving the problem. In the proposed algorithms, three different kinds of neighborhood search operators, including general local search, objective-specific local search, and large neighborhood search, are carefully designed and combined in a synergistic manner. The experimental results show the effectiveness of the proposed algorithms. Relationships between different objectives in this multiobjective problem are also discussed.**

***Index Terms*—Heterogeneous fleet, location-routing problem (LRP) with time windows, multiobjective optimization, multiple neighborhood search (MNS).**

I. Introduction

RECENTLY, the design of distribution networks has received increasing attention from the government and business organization. Locating depots and determining vehicle routes are two important and inter-related problems in the distribution network design process. They are traditionally solved separately. Location-routing problem (LRP) has been proposed to integrate location and routing together [1]. A solution to LRP includes the locations of depots and their corresponding vehicles routes to serve the customers under a series of constraints [2]. LRP has a wide range of applications in real life, including food and drink distribution, parcel delivery, and telecommunication network design [3]. Considering a heterogeneous fleet and time windows, a more practical variant, fleet size and mix LRP with time windows (FSMLRPTWs), was introduced in [3] recently.

FSMLRPTW is undoubtedly NP-hard since location and routing problems are NP-hard. Therefore, to deal with large-scale FSMLRPTW instances, metaheuristics are often more suitable than exact approaches. In [3], a hybrid evolutionary search algorithm (HESA) was proposed to solve FSMLRPTW. The main objective of HESA is to minimize the sum of depot cost, vehicle fixed cost, and total travel cost of the routes. That is, HESA optimizes multiple objectives in a single-objective manner and can return only one final solution to decision makers. Due to the constraints and problem structure of FSMLRPTW, the improvement of one objective may lead to the deterioration of other objectives. Thus, FSMLRPTW is essentially a multiobjective optimization problem (MOP) [4]–[6]. Since the decision maker's preference is often unknown *a priori*, it is necessary to provide a set of representative Pareto solutions for FSMLRPTW.

This paper defines a multiobjective FSMLRPTW (MO-FSMLRPTW) with six objectives. These objectives include total cost of opening depots, total cost of vehicles, total travel distance, makespan, total waiting time, and total delay time [4]. Then, this paper develops two versions of multiobjective multiple neighborhood search (MNS) for solving the problem. In the proposed multiobjective MNS algorithms, three different kinds of neighborhood search operators, including general local search (GLS), objective-specific local search, and large neighborhood search (LNS), are carefully designed and combined in a synergistic manner. To better study the performance of the proposed algorithms, a set of real-world benchmark instances based on realistic data are generated. The contributions of this paper are as follows: 1) introducing a six-objective version of FSMLRPTW and a set of realistic benchmark instances; 2) proposing two versions of multiobjective MNS for MO-FSMLRPTW; and 3) studying and testing the MNS strategy in the proposed algorithms. As far as we know, this is the first study to consider FSMLRPTW as an MOP.

Manuscript received October 15, 2018; revised January 28, 2019; accepted April 9, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61673403, Grant 71601191, and Grant U1611262, in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313703, and in part by the JSPS KAKENHI under Grant JP17K12751. This paper was recommended by Associate Editor B. Derbel. *(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).

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

S. Gao is with the Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan.

Y. Zhou is with the College of Information, Guangdong University of Finance and Economics, Guangzhou 510320, China.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

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.2019.2912194

The remainder of this paper is organized as follows. Section II introduces the problem formulation and benchmark instances. Section III reviews related work. Section IV proposes an MNS strategy. Section V presents two versions of multiobjective MNS algorithms. Section VI presents the results. Section VII gives a detailed discussion. The conclusions are drawn in Section VIII.

II. Problem Formulation and Real-World Instances

A. MO-FSMLRPTW Formulation

MO-FSMLRPTW consists of opening a set of potential depots, assigning customers to them, and determining a set of routes for heterogeneous vehicles to serve a group of customers. It can be summarized as follows.

1) *Inputs:*
   a) An undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$.
   b) *Customer Node Set* $\mathcal{V}_c \in \mathcal{V}$*:* Each customer $i \in \mathcal{V}_c$ is associated with a positive demand, service time, service time window, and allowed maximum delay time.
   c) *Depot Node Set* $\mathcal{V}_d \in \mathcal{V}$*:* Each depot $p \in \mathcal{V}_d$ has a storage capacity, opening time, and fixed opening cost.
   d) *Edge Set* $\mathcal{A} = \{(i, j) : i \in \mathcal{V}, j \in \mathcal{V}, i \neq j\}$*:* Each edge is associated with a travel distance and travel time.
   e) *A Set K of Different Vehicle Types:* A vehicle of type $k \in K$ has a loading capacity and fixed cost.
2) *Outputs:*
   a) A set of Pareto nondominated solutions.
3) *Objectives:*
   a) *Total Cost of Opening Depots (f_1):* Fixed cost of opening depots from potential depots.
   b) *Total Cost of Vehicles (f_2):* Fixed cost of buying vehicles. Generally, larger capacity of a vehicle leads to larger fixed cost.
   c) *Total Travel Distance (f_3):* Variable cost estimated by using a function of the total distance traveled.
   d) *Makespan (f_4):* The maximum working hours among all drivers. Minimizing the makespan enables a balanced workload.
   e) *Total Waiting Time (f_5):* If a vehicle arrives at a customer before the starting time, it needs to wait. Minimizing the waiting time improves work efficiency and avoids wasting working hours.
   f) *Total Delay Time (f_6):* Service cost related to the satisfaction of customers.
4) *Constraints:*
   a) *Traveling Customer Constraint (C_1):* Each customer must be visited exactly once.
   b) *Demand Constraint (C_2):* The demand of each customer should be satisfied.
   c) *Opening Capacity Constraint (C_3):* The total demands of customers that assign to a depot should not exceed the capacity of the depot.
   d) *Vehicle Capacity Constraint (C_4):* The total serving demands of each route should not exceed the vehicle capacity.
   e) *Maximum Delay Time Constraint (C_5):* Delay time at each customer should not exceed the maximum allowed delay time.
   f) *Return Time Constraint (C_6):* Each vehicle should originate from a depot and terminate at the same depot before the depot closes.

A solution of MO-FSMLRPTW is represented as a set of routes $\mathcal{R} = \{R^1_{p,k}, \ldots, R^{N_{p,k}}_{p,k} : p \in \mathcal{V}_d, k \in K\}$, where $N_{p,k}$ represents the number of routes whose depot is p and vehicle type is k, and $N_p = \sum_{k \in K} N_{p,k}$ represents the number of routes that originates at depot p. $R^j_{p,k}$ denotes the jth route whose vehicle type is k for depot p. For example, there are three opening depots represented as $\mathcal{V}_d = \{1, 2, 3\}$, as shown in Fig. 1. The set of customers is represented as $\mathcal{V}_c = \{4, 5, \ldots, 16\}$. The routes $\mathcal{R} = \{R^1_{1,L}, R^2_{1,L}, R^1_{2,S}, R^1_{2,M}, R^1_{3,M}, R^1_{3,L}\}$ where L, M, S means a large, middle, and small size vehicle, respectively. Considering soft time windows as in [4]–[6], the definitions of six objectives (f_1–f_6) are as follows.

1) *Total Cost of Opening Depots:*

$$f_1 = \sum_{p \in \mathcal{V}_d} h_p C^d_p \tag{1}$$

where $h_p = 1(0)$ represents that depot p is opened (closed), and C^d_p is the opening cost of depot p.

2) *Total Cost of Vehicles:*

$$f_2 = \sum_{k \in K} \sum_{i \in \mathcal{V}_d} \sum_{j \in \mathcal{V}_c} C^v_k z^k_{ij} \tag{2}$$

where z^k_{ij} represents the sequence of customers to be served. $z^k_{ij} = 1$ means that vehicle of type k serves directly from node i to j, 0 otherwise. C^v_k is the cost of vehicle v with type k.

3) *Total Travel Distance:*

$$f_3 = \sum_{p \in \mathcal{V}_d} \sum_{j=1}^{N_{p,k}} D_{R^j_{p,k}}, k \in K \tag{3}$$

where $D_{R^j_{p,k}}$ represents the total travel distance of route $R^j_{p,k}$.

4) *Makespan, i.e., Travel Time of the Longest Route:*

$$f_4 = \max_{p \in \mathcal{V}_d} \max_{j=1}^{N_{p,k}} T_{R^j_{p,k}}, k \in K \tag{4}$$

where $T_{R^j_{p,k}}$ represents the total travel time of $R^j_{p,k}$, which is the sum of travel time of all edges, the waiting time, and the service time at each customer in $R^j_{p,k}$.

5) *Total Waiting Time:*

$$f_5 = \sum_{p \in \mathcal{V}_d} \sum_{j=1}^{N_{p,k}} \mathrm{WT}_{R^j_{p,k}}, k \in K \tag{5}$$

where $\mathrm{WT}_{R^j_{p,k}}$ represents the total waiting time in $R^j_{p,k}$. If a vehicle arrives early at a customer, it must wait until

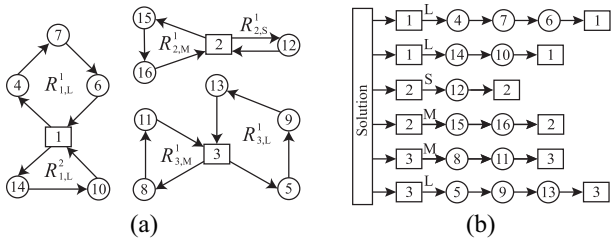


Fig. 1. Solution and its representation.

the earliest service time, otherwise, the waiting time at this customer is 0.

6) *Total Delay Time:*

$$f_6 = \sum_{p \in \mathcal{V}_d} \sum_{j=1}^{N_{p,k}} \mathrm{DL}_{R_{p,k}^j}, k \in K \tag{6}$$

where $\mathrm{DL}_{R_{p,k}^j}$ represents the total delay time in $R_{p,k}^j$. If a vehicle arrives later than the time window of a customer, delay time is caused.

B. Real-World Instances

The traditional benchmark instances for FSMLRPTW in [3] are extended from Soloman's symmetric instances [7] for VRPTW. Thus, the distance and time matrices are the same and symmetric. This is hardly a realistic scenario. Further, Solomon data set is not suitable to conduct a proper multiobjective study because weak dependency relationships are presented among objectives. In order to construct a more real and challenging scenario for MO-FSMLRPTW, we generate MO-FSMLRPTW benchmark instances based on real-world MO-VRPTW instances from [4]. These MO-FSMLRPTW benchmark instances are also called real-world MO-FSMLRPTW instances in this paper since they are based on real-world MO-VRPTW instances. Thus, we can distinguish between the generated real-world MO-FSMLRPTW instances and the traditional benchmark instances for FSMLRPTW in [3]. The MO-VRPTW instances are originally proposed by Castro-Gutierrez *et al.* [4] based on a distribution company in Tenerife Spain [8]. These real-world MO-VRPTW instances show some superiority than symmetric Solomon instances. First, the distance and time matrices are distinct and asymmetric, which represent a realistic tradeoff between travel distance and travel time. Second, strong dependency relationships are presented among objectives. Thus, the real-world instances exhibit more realistic multiobjective nature and are suitable to assess the performance of multiobjective optimization algorithms [8].

A real-world MO-VRPTW instance only contains one depot and one type of vehicle. However, a real-world MO-FSMLRPTW instance has $|\mathcal{V}_d|$ depots and $|K|$ classes of vehicles with different capacities and fixed cost. To generate our real-world MO-FSMLRPTW instances, the cost and capacity of the heterogeneous vehicles, shown in Table S1 in the supplementary material, are obtained by following a similar procedure in [9]. Then, k-means clustering is adopted to transform $|\mathcal{V}_d| - 1$ customers into depots. Specifically, the customers nearest to the cluster centers are transformed into depots. The cost and capacity of the depots are shown in Table S2 in the supplementary material.

III. Related Work

A. Existing Algorithms for FSMLRPTW

As mentioned in Section I, an HESA was proposed in [3] to solve FSMLRPTW. HESA optimizes multiple objectives in a single-objective manner; therefore, only one final solution is returned to decision makers. To the best of our knowledge, no previous work utilizes the multiobjective optimization methods for FSMLRPTW, which motivates this paper. The proposed algorithms can serve as a baseline for MO-FSMLRPTW.

B. Multiobjective Optimization and Multiobjective Evolutionary Algorithms

An MOP with m objectives to be optimized is defined as

$$\text{Minimize} \quad F(x) = (f_1(x), f_2(x), \ldots, f_m(x)), \quad x \in \mathbf{\Omega} \tag{7}$$

where $\mathbf{\Omega}$ is the decision space. The objective space is the set $\mathbf{\Phi} = \{f_1, f_2, \ldots, f_m\}$. For two feasible solutions x and y, it is said x dominates y, if and only if: $\forall f_i \in \mathbf{\Phi}, f_i(x) \leq f_i(y) \wedge \exists f_i \in \mathbf{\Phi}, f_i(x) < f_i(y)$. A solution x is a Pareto optimal if it is not dominated by any other solutions in $\mathbf{\Omega}$. Pareto optimal set (PS) consists of all Pareto optimal solutions and Pareto front (PF) is defined as

$$\mathrm{PF} = \{F(x) | x \in \mathrm{PS}\}. \tag{8}$$

The goal of a multiobjective optimization algorithm is to seek a set of nondominated solutions that is not only close to the PF but also well distributed along it. Multiobjective evolutionary algorithms (MOEAs) simultaneously deal with a set of possible solutions (the so-called population), thus they find a set of the PS in a single run. Additionally, MOEAs are less susceptible to the shape or continuity of the PF (e.g., they can easily deal with discontinuous and concave PFs). Over the past few decades, a number of MOEAs have been proposed to deal with multiobjective continuous optimization problems. Generally, these MOEAs can be divided into three categories: 1) Pareto dominance-based methods (e.g., NSGA-II [10], SPEA2 [11], and VaEA [12]); 2) decomposition-based methods (e.g., MOEA/D [13] and its variants [14]–[17]); and 3) indicator-based methods (e.g., SMS-EMOA [18] and HypE [19]).

C. General Local Search

A classical or general local search, the current solution can generate a set of neighboring solutions by using neighborhood operators. GLS is based on neighborhood operators with only small modification is introduced to the current solution, and thus supports intensification. Many neighborhood operators of GLS have been developed for VRP variants [20]. A set of commonly used neighborhood operators, GLS $= \{G_1, G_2, \ldots, G_7\}$ are introduced as follows.

1) *G_1:* This operator removes the worst customer from a randomly selected route, and then reinserts this customer into the best position of the same route.
2) *G_2:* Two routes are randomly selected and the worst customer in these two routes is selected, and then this customer is reinserted into the best position of the other route.
3) *G_3:* It considers all vehicle routes. First, it randomly removes a customer from a randomly selected route, and then reinserts this customer into the best position in the current solution. This operator may open a closing depot.
4) *G_4:* This operator is similar to G_3 except for removing a random number of customers from a randomly selected route. This operator also may open a closing depot.
5) *G_5:* This operator exchanges two customers from a randomly selected route.
6) *G_6:* This operator is similar to G_5 except that two customers are chosen from two different randomly selected routes, respectively.
7) *G_7:* This operator exchanges a sequence of customers between two randomly selected routes without changing the orientation of the sequences. More details of this operator can be found in [5].

These neighborhood operators can be divided into two categories: G_2, G_3, G_4, G_6, and G_7 are inter-route operators, while G_1 and G_5 are intra-route operators.

D. Multiobjective Local Search and Multiobjective Memetic Algorithms

For a single-objective combinatorial optimization, it is commonly believed that local search can promote search intensification and speed up convergence. In fact, it also holds for the multiobjective combinatorial optimization. Recently, many multiobjective local search algorithms [21]–[28] were proposed for multiobjective combinatorial optimization problems. Multiobjective local search algorithms also can be divided into three categories [29]: 1) Pareto dominance-based local search; 2) decomposition-based local search; and 3) indicator-based local search algorithms. The merit of local search-based algorithms is that problem-specific knowledge can be directly used to guide the search toward PF. More details about local search for multiobjective combinatorial optimization can be found in the recent review [30].

Most of the existing multiobjective local search algorithms [21]–[28] are based on GLS. During the neighborhood search procedure, weighted sum approach is often used to transform a multiobjective problem, for example, MO-FSMLRPTW, into a single-objective problem for evaluation. The weighted sum function is defined as follows:

$$f^{ws}(x|\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i f_i(x) \tag{9}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a weight vector.

Local search is also usually embedded into MOEAs, composing a new kind of algorithms called multiobjective memetic algorithm (MOMA) [31]–[33]. This kind of algorithms achieves good performance on multiobjective combinatorial optimization. More details about MOMA can be found in review [31]. Typical MOMAs related with this paper are [32] and [33]. In [32], GLS with weighted sum function is embedded into MOEA/D and NSGA-II for multiobjective capacitated arc routing problem. In [33], GLS with weighted sum function is embedded into NSGA-II for multiobjective flexible job shop scheduling.

IV. Multiple Neighborhood Search

A. Motivation

The motivation comes from two aspects as follows.

1) In MO-FSMLRPTW, different objectives have different characteristics in terms of their physical meaning, scale/granularity, continuous/discrete value, and thus have different difficulties [34]. All the above neighborhood operators in GLS have different search abilities for the six objectives in MO-FSMLRPTW. For example, it is hard to close an opening depot to optimize f_1 using $G_1, G_2, \ldots, G_7$; it is also hard to reduce the vehicle cost by the above operators for f_2; additionally, f_4 only concerns the longest travel time route. However, the above operators randomly select a route from the current solution, thus it is difficult to optimize f_4. In summary, f_1, f_2, and f_4 are the hardest objectives to be optimized by GLS among the six concerned objectives. They should be improved directly by properly designing objective-specific local search.
2) All the above GLS apply small changes to the current solution, and thus can only search a neighborhood that is quite similar to the current solution. However, MO-FSMLRPTW may have a large solution space. The constraints $(C_1 - C_6)$ in MO-FSMLRPTW could be very tight, which may isolate the infeasible regions and feasible regions in the solution space. Then, GLS might be difficult to find the global optimum and easily trapped in local optima. In this case, LNS with a larger searching step may be a promising choice to help GLS escape from local optimal solutions.

B. Objective-Specific Local Search

Three specific neighborhood operators are carefully designed for optimizing f_1, f_2, and f_4, respectively. They are denoted as $\text{OLS} = \{O_1, O_2, O_3\}$.

1) O_1 for Depot Cost f_1: It is shown in Algorithm 1. First, a temporary solution x' is generated, which is a copy of the solution x. Then, delete the depot with the fewest customers from x and randomly enumerate those customers from the deleted depot to insert them into other depots. Here, the customer is inserted into the first feasible position satisfying constraints $C_1 - C_6$. If all the customers can be inserted successfully, then one depot can be directly reduced and the operator is terminated. Otherwise, the solution x is recovered with the temporary solution x'.

2) O_2 for Vehicle Cost f_2: The total customer demands of a route should be close to, but no more than, the capacity of the used vehicle type in order to improve the utilization efficiency of each vehicle type. Thus, O_2 tries to adopt a smaller

Algorithm 1: Specific Neighborhood Operator O_1 for f_1

Input: a solution x
Output: an updated solution x
begin
  $x' = x$;
  close the depot $p, p \in \mathcal{V}_d$ with the fewest customers from x';
  enumerate all customers in p and try to insert them into other depots of x';
  if *all customers can be inserted into other depots successfully* **then**
    $x = x'$;
  end if
  return x;
end

size vehicle for a route to obtain saving and reduce the cost of vehicles. As shown in Algorithm 2, first, the operator checks whether a route can be served with a smaller vehicle by removing several customers with total demands no more than C_T (capacity threshold value). If yes, O_2 randomly removes a customer in this selected route until the route can be served by a smaller size vehicle finally. Note that if the vehicle type of this route is the smallest one and the total demands of all the customers are smaller than C_T, all the customers from this route are removed. Finally, all removed customers are reinserted into other routes properly without changing the vehicle type of the inserted route.

C_T is better set to a random number from the interval $[2 \cdot d_m, 5 \cdot d_m]$, where d_m is the average demand of customers in the given instance. It is expected that a route can be served with a smaller vehicle by removing about 2∼5 customers. In this paper, $C_T = 40$ is adopted according to the demand of customers in the benchmark instances.

3) O_3 for Makespan f_4: The specific neighborhood operator O_3 is designed for f_4, which is shown in Algorithm 3. All the above general neighborhood operators in GLS involve a basic operator, namely, route selection. Since f_4 only concerns the longest travel time route, O_3 chooses the longest route to implement the selected neighborhood operator. Further, in O_3, when a neighborhood operator is selected from GLS, we also require that the makespan of a generated solution is not worse than that of the original solution.

Our designed objective-specific local search operators, O_1, O_2, and O_3, ensure that the specific objectives, f_1, f_2, and f_4, in the local search path is nonincreasing. However, the increasing of other objectives is suitably controlled or restrained by using weighted sum function (9) to compare two solutions, which make a good comprise between the six objectives.

Algorithm 2: Specific Neighborhood Operator O_2 for f_2

Input: a solution x
Output: an updated solution x
begin
  for each route R in x **do**
    $x' = x$;
    if R is selected to remove some customers to fit smaller vehicle type **then**
      removal list $L_r = \emptyset$;
      if R is served with the smallest vehicle **then**
        remove all customers in R and put them in removal list L_r;
      else
        while R can not be serviced with a smaller vehicle **do**
          randomly select a customer c from R and remove this customer in R;
          $L_r = L_r \bigcup c$;
        end while
      end if
      enumerate all customers in L_r and try to insert them into other routes without changing the vehicle type of the inserted route;
      if a customer cannot be inserted into other routes successfully **then**
        $x = x'$;
      end if
    end if
  end for
  return x;
end

Algorithm 3: Specific Neighborhood Operator O_3 for f_4

Input: a solution x, a weight vectors λ
Output: an updated solution x
begin
  randomly select a neighborhood operator from GLS;
  directly select the longest route of x to implement the selected a neighborhood operator without generating a longer makespan;
  return the updated solution x;
end

Algorithm 4: LNS

Input: a solution x
Output: a new solution x'
begin
  /* destroy operator */
  select n customers set $\mathcal{C}'$ in x ;
  delete all customers successively in $\mathcal{C}'$ from s to produce an intermediate solution x';
  /* repair operator */
  reinsert the removed customers into x'. If the inserted customer belongs to a closing depot, then open the depot;
  return x';
end

C. Large Neighborhood Search

LNS has obtained good performance on some single-objective VRP variants [35]–[37]. It can hugely change the structure of the solution to escape from a possible local optimum. The main procedure of LNS in MO-FSMLRPTW is shown in Algorithm 4. There are two operators in LNS, namely, destroy operator and repair operator. The destroy operator removes n customers from the solution, where n is in the interval $[r_l \cdot |\mathcal{V}_c|, r_u \cdot |\mathcal{V}_c|]$. The interval is defined by a lower and an upper bound calculated as a percentage of total number of customers in a given instance. In this paper, $[r_l, r_u] = [0.1, 0.3]$ is adopted. In the repair operator, the removed n customers successively are reinserted into the solution.

In this paper, a simple LNS, including two removal operators and one insertion operator, is designed as follows.

1) *Random Removal (RR):* Randomly select n customers from $\mathcal{V}_c$ and remove them from the solution.
2) *Weighted Sum Removal (WSR):* This operator iteratively removes n customers based on the weighted sum function (9). In each iteration, it removes a customer

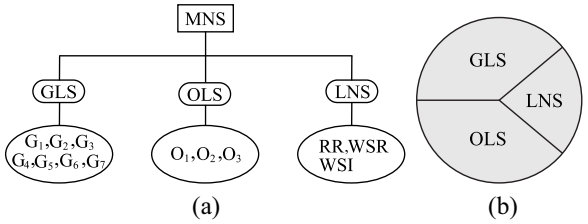


Fig. 2. MNS. (a) Framework of MNS. (b) Selection probabilities of different kinds of neighborhood search.

such that the resultant partial solution has the minimal weighted sum of objectives.

3) *Weighted Sum Insertion (WSI):* During the insert procedure, the removed n customers are sequentially inserted into the best positions in the solution with the minimal weighted sum of objectives.

D. Multiple Neighborhood Search

Three different kinds of neighborhood search operators, GLS, OLS, and LNS, are combined in a synergistic manner for MO-FSMLRPTW. This combined neighborhood search is called MNS strategy in this paper, as shown in Fig. 2(a). GLS and OLS are two kinds of local search to promote intensification, while LNS is to promote diversification.

One key problem of MNS is how to coordinate these operators. In this paper, a stochastic selection procedure, in the form of a simple roulette wheel selection as shown in Algorithm 5, is adopted to select one kind of neighborhood search to potentially improve a given solution. The selection probabilities of GLS, OLS, and LNS are set to P_G, P_O, and P_L, respectively, where $P_G + P_O + P_L = 1$. In general, GLS and OLS are set to relatively large selection probabilities with the same value to optimize all objectives for convergence, while LNS is set to a relatively small selection probability to help GLS and OLS escape from possible local minima, as shown in Fig. 2(b) in Algorithm 5. The parameter MaxDepth means the search depth of MNS.

Within one kind of neighborhood search, neighborhood operators are selected randomly for simplicity. Specifically, when GLS has been selected, a neighborhood operator is randomly selected from $\{G_1, G_2, \ldots, G_7\}$. When OLS has been selected, a specific neighborhood operator is also randomly selected from $\{O_1, O_2, O_3\}$. When LNS has been selected, the removal operator is randomly selected from {RR,WSR}, then WSI is used to reinsert the deleted customers.

V. Two Versions of Multiobjective Multiple Neighborhood Search Algorithms

In this section, two versions of multiobjective MNS algorithms, i.e., MOMNS/D and MOMNS/V, are proposed to solve MO-FSMLRPTW. Specifically, MOMNS/D is multiobjective MNS based on decomposition [13], while MOMNS/V is multiobjective MNS based on Pareto dominance and vector angles for convergence and diversity (niche preservation), respectively, [12]. The general frameworks of MOMNS/D and MOMNS/V are presented in Algorithms 6 and 7, respectively.

Algorithm 5: $\mathrm{MNS}_{\boldsymbol{\lambda}}(x, A)$

```
   Input: a solution x, an archive A, a weight vector λ
   Output: x
1  begin
2      for depth = 1, ..., MaxDepth do
3          Rand = rand(0, 1);
4          if Rand < P_G then
               /* select a neighborhood operator from
                  GLS to improve solution x          */
5              randomly select a neighborhood operator G from
               {G_1, G_2, ..., G_7};
6              x' = G(x);
7          else
8              if Rand < P_G + P_O then
                   /* select a neighborhood operator
                      from OLS to improve solution x  */
9                  randomly select a specific neighborhood O from
                   {O_1, O_2, O_3};
10                 x' = O(x);
11             else
                   /* select LNS to update x          */
12                 x' = LNS(x);
13             end if
14         end if
15         update archive A with x';
16         if f^ws(x'|λ) ≤ f^ws(x|λ) then
17             x = x';
18         end if
19     end for
20     return x;
21 end
```

A. MOMNS/D

In MOMNS/D, an MOP is decomposed into a set of scalar optimization subproblems to be solved simultaneously by evolving a population of solutions [13]. At the beginning, N initial solutions and N uniformly distributed weight vectors $\boldsymbol{\Lambda}^1, \ldots, \boldsymbol{\Lambda}^N$ are generated (lines 3 and 4 in Algorithm 6). The jth optimization subproblem is defined as follows:

$$\text{minimize} \quad g^{ws}\left(x|\boldsymbol{\Lambda}^j\right) = \sum_{i=1}^{m} \Lambda_i^j f_i(x), j \in \{1, 2, \ldots, N\}. \quad (10)$$

The neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors (line 5). At each iteration, for each subproblem, a solution x^I is randomly selected from neighborhood of the subproblem. Then, MNS is used to improve x^I (line 9).

B. MOMNS/V

In MOMNS/V, first, N initial solutions and N_λ uniformly distributed weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^{N_\lambda}$ are generated (lines 3 and 4 in Algorithm 7). Different from MOMNS/D, MOMNS/V is a Pareto dominance-based local search algorithm. Thus, weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^{N_\lambda}$ can be seen as a weight pool for MNS, which is only used for guiding the search direction in MNS. For each generation, we generate N solutions and put them in a set $\mathcal{Q}$ (lines 6–12). Then, Selection (line 13) is used to select N solutions as the new population.

In the Selection procedure, the solution set $\mathcal{P} \bigcup \mathcal{Q}$ is normalized (line 2 in Algorithm 8). Then, nondominated sorting is used to divide $\mathcal{P} \bigcup \mathcal{Q}$ into different layers and find the critical layer L_i (lines 4–7). If the population $\mathcal{P}$ is not full, we

Algorithm 6: MOMNS/D

Input: population size N, neighborhood size T, maximum generation G_{max}, archive A
Output: archive A

begin
  archive $A = \emptyset$;
  initialize population $\mathcal{P}$ with N solutions $x^1, \ldots, x^N$ and update the archive A;
  generate N uniformly distributed weight vectors $\mathbf{\Lambda}^1, \ldots, \mathbf{\Lambda}^N$, where $\mathbf{\Lambda}^i = (\Lambda_1^i, \Lambda_2^i, \ldots, \Lambda_m^i)$;
  compute a neighborhood set $B(i) = \{i_1, \ldots, i_T\}$ for each weight vector $\mathbf{\Lambda}^i$, where $\mathbf{\Lambda}^{i_1}, \ldots, \mathbf{\Lambda}^{i_T}$ are T closest weight vectors to $\mathbf{\Lambda}^i$ based on the Euclidean distance;
  for $g = 1 : G_{max}$ **do**
    for $i = 1 : N$ **do**
      randomly select an index I from $B(i)$;
      $x' = \mathbf{MNS}_{\mathbf{\Lambda}^i}(x^I, A)$;
      for each $j \in B(i)$ **do**
        if $g^{ws}(x'|\mathbf{\Lambda}^j) \leq g^{ws}(x^j|\mathbf{\Lambda}^j)$ **then**
          $x^j = x'$;
        end if
      end for
    end for
  end for
  return archive A;
end

Algorithm 7: MOMNS/V

Input: population size N, maximum generation G_{max}, archive A, the number of weight vectors N_λ
Output: archive A

begin
  Archive $A = \emptyset$;
  initiate population $\mathcal{P}$ with N solutions and update the archive A;
  generate N_λ uniformly distributed weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^{N_\lambda}$, where $\boldsymbol{\lambda}^j = (\lambda_1^j, \ldots, \lambda_m^j)$;
  for $g = 1 : G_{max}$ **do**
    $\mathcal{Q} = \emptyset$;
    for $i = 1 : N$ **do**
      randomly select a solution x from $\mathcal{P}$;
      randomly select a weight vector $\boldsymbol{\lambda}^j$ from $\{\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^{N_\lambda}\}$;
      $x' = \mathbf{MNS}_{\boldsymbol{\lambda}^j}(x, A)$;
      $\mathcal{Q} = \mathcal{Q} \bigcup \{x'\}$;
    end for
    `/* Algorithm 8 */`
    $\mathcal{P}$ = **Selection**$(\mathcal{P} \bigcup \mathcal{Q})$;
  end for
  return archive A;
end

Algorithm 8: Selection($\mathcal{S}$)

Input: the population $\mathcal{S}$ that are the union of $\mathcal{P}$ and $\mathcal{Q}$, population size N
Output: the new population $\mathcal{P}$

begin
  normalize the set $\mathcal{S}$ and calculate the norm of each solution in the normalized objective space;
  $\mathcal{P} = \emptyset$, $i = 1$;
  $\{L_1, L_2, \ldots, L_i, \ldots, \}$ = Nondominated-sorting($\mathcal{S}$);
  while $|\mathcal{P}| + |L_i| \leq N$ **do**
    $\mathcal{P} = \mathcal{P} \bigcup L_i$;
    $i = i + 1$;
  end while
  if $|\mathcal{P}| < N$ **then**
    associate each member of L_i with a solution in $\mathcal{P}$ that based on vector angles;
    select $N - |\mathcal{P}|$ solutions one by one from L_i to construct final population $\mathcal{P}$ using niche preservation based on vector angles;
  end if
  return population $\mathcal{P}$;
end

associate each member of L_i with a solution in $\mathcal{P}$ with the minimum vector angle, and select $N - |\mathcal{P}|$ solutions with the maximum associated vector angle one by one (lines 8–11).

The selection procedure is adopted from a vector angle-based evolutionary algorithm for continuous many-objective optimization proposed recently [12]. More details of the operator in Algorithm 8 can be found in [12]. It is expected that MOMNS/V using this advanced selection mechanism should produce good results for MO-FSMLRPTW, a many-objective combinatorial optimization problem.

C. Algorithm Components

1) Initialization: In this phase, N initial solutions are generated. An initial solution is generated as follows. First, we randomly select a customer c ($c \in \mathcal{V}_c$), and assign this customer to a random depot p ($p \in \mathcal{V}_d$). If there is a feasible position that c can be inserted, then it is inserted. Otherwise, this customer is assigned to another depot. Repeat this procedure until all the customers are inserted successfully.

2) Feasibility Checking: In our algorithm, only feasible solutions are considered. Therefore, when neighborhood operators are applied to a solution, it is necessary to check time, capacity of vehicle, and depot constraints. Efficient feasibility checking is beneficial to reduce the computational complexity. It is easy to check the capacity constraint of vehicle and depot with time complexity $O(1)$. For time constraint, a slack time $\mathrm{Sla}_{c_r}(i)$ can be used to define the maximum time shift allowed of the ith customer in route r [5], [6]. Using the slack time method, the computational complexity of feasibility checking can be reduced to $O(1)$. Details can be found in [5] and [6].

3) Archive: To obtain a set of representative Pareto solutions with good convergence and diversity, an external archive A with maximal size E is adopted. All nondominated solutions are stored if the current size of A is less than E. Otherwise, a diversity evaluation mechanism is used to assess the density of each solution in A, and then the solution with the maximum density is discarded from A. In this paper, an advanced density estimation, named parallel cell coordinate system (PCCS) [38], is adopted. PCCS has been shown to be more effective than adaptive grid and crowding distance [38]. MO-FSMLRPTW is a many-objective optimization problem, thus it is difficult for simple density estimator such as crowding distance to maintain a set of representative Pareto solutions. Details of calculating the density of the solution in A by PCCS can be found in [38].

D. Complexity Analysis

The complexity of MOMNS/D mainly depends on neighborhood search and archive updating. For neighborhood search, the main complexity comes from LNS. The complexity of LNS is $O(|\mathcal{V}_c|^2)$, where $|\mathcal{V}_c|$ is the number of customers. For archive updating, the complexity is $O(m \cdot E^2)$, where m is the number of objectives, and E is the maximum size of the archive. The maximum depth of local search is MaxDepth. Hence,

the worst-case complexity of local search in MOMNS/D is $\max(\text{MaxDepth} \cdot O(|\mathcal{V}_c|^2),\ \text{MaxDepth} \cdot O(m \cdot E^2))$.

The complexity of MOMNS/V should consider the Selection procedure. In Selection, the nondominated sorting has a time complexity of $O(N \cdot \log^{m-2} N)$, and the association and niching operator has a time complexity of $O(m \cdot N^2)$. Thus, the complexity of MOMNS/V is $\max(\text{MaxDepth} \cdot O(|\mathcal{V}_c|^2), \text{MaxDepth} \cdot O(m \cdot E^2), O(N \cdot \log^{m-2} N), O(m \cdot N^2))$.

VI. Experimental Studies

A. Parameter Settings

1) In MOMNS/D, the number of weight vectors is calculated as C_{H+m-1}^{m-1}, where m is 6 in MO-FSMLRPTW and H is set to 5. Thus, $C_{10}^{5} = 252$. Since each subproblem corresponds to a weight vector, the population size is equal to the number of weight vectors. The neighborhood size T is set to 5.
2) In MOMNS/V, the weight vectors are only used in MNS for guiding the search direction of neighborhood solutions. The method of generating of weight vectors in MOMNS/V is the same as that in MOMNS/D. H is also set to 5.
3) In MNS, $P_G = 0.4$, $P_O = 0.4$, and $P_L = 0.2$. The depth MaxDepth is set to 10 based on the balance of solution quality and computational time.
4) In both MOMNS/D and MOMNS/V, the population size N is set to 252, the maximum generation $G_{\max}$ is set to 700, and the archive size E is set to 200 for a fair comparison.

B. Benchmark Instances

As pointed out in [4], real-world instances are more suitable for conducting multiobjective studies. The traditional instances are also considered to test the robustness and stability of the proposed algorithm frameworks. The details of real-world instances and traditional instances are as follows.

1) Real-World Instances: Thirty real-world MO-FSMLRPTW instances are generated by combining three kinds of customer size, five kinds of time windows, and two kinds of depot number. Each instance is named as $c - t - d$, where $c \in \{50, 150, 250\}$ represents the size of customer, $t \in \{0, 1, 2, 3, 4\}$ indicates the type of time windows and $d \in \{4, 6\}$ represents the depot number. The cost and capacity of vehicles and depots in each instance can be found in Tables S1 and S2 in the supplementary material. The maximum allowed delay time of each customer is set to 30 min as in [39].

2) Traditional Instances: Fifty six traditional instances can be found in [3]. These instances include three classes: C, R, and RC, which means clustered, random, and semi-clustered data set, respectively. In these data sets, C1XX, R1XX, and RC1XX correspond to a small vehicle capacities and short scheduling horizon, while C2XX, R2XX, and RC2XX correspond to a large vehicle capacities and long scheduling horizon. In [9], three different vehicle types A, B, and C, are considered. In this paper, we only consider the largest scale instances with 100 customers and vehicle type of A. Similar to [39], the maximum delay time of each customer is set as $0.3(e_i - b_i)$, where $[b_i, e_i]$ denotes the time window of customer i.

C. Performance Metric

Inverted generational distance (IGD) [40] and hypervolume (HV) [41] are two widely used metrics for performance evaluation. It is argued in [42] that it is not easy to evaluate the difference between the obtained solution set and the PF using distance-based performance measures such as IGD for many-objective optimization. This is because: 1) Pareto optimal solutions of many-objective problems are usually unknown and 2) a large number of Pareto optimal solutions are needed to calculate the measure in a reliable manner. Thus, as in [42], only HV is used for performance evaluation. HV calculates the volume of the space enclosed by a solution set and a reference point. It assesses the quality of a nondominated set in terms of convergence and diversity. Higher value of HV indicates better performance. The HV value is calculated based on the objective values normalized into [0, 1], and the point (1.2, 1.2, 1.2, 1.2, 1.2, 1.2) is used as the reference point.

To have statistically sound conclusions, single-problem Wilcoxon rank-sum test [43] at a 0.05 significance level is adopted. The best values among algorithms are highlighted in boldface in tables. Statistical results on all instances are summarized as $w/t/l$, which means that the performance of benchmark algorithm is better than, similar to and worse than the corresponding competitor in w, t, and l instances, respectively. Multiproblem Wilcoxon signed-rank test is also conducted. A final ranking of all algorithms on sets of instances is given by Friedman test [44], [45].

Due to limited space, details of numerical values of HV over 30 independent runs are presented in Tables S3–S8 in the supplementary material.

D. Effect of Multiple Neighborhood Search

To analyze the effect of MNS for MOMNS/D, we design three variants of MOMNS/D as follows.

1) *MOMNS/D-G:* This variant only uses one kind of neighborhood search, GLS, as in most of existing multiobjective local search algorithms [21]–[28] and MOMAs [32], [33]. Thus, it can be seen as a common baseline algorithm. Comparing this variant with MOMNS/D can show the contribution of the introduced OLS and LNS. Moreover, MOMNS/D-G is not only the variant of MOMNS/D, but also can be seen as the state-of-the-art MOMA, as existing MOMAs [32], [33], for MO-FSMLRPTW. Thus, comparing this variant with MOMNS/D can show how much improvement can be achieved with respect to the state-of-the-art algorithm.
2) *MOMNS/D-GL:* This variant removes OLS from MNS. Comparing this variant with MOMNS/D can show the contribution of the proposed OLS. In this variant, $P_L = 0.2$ and $P_G = 0.8$.
3) *MOMNS/D-GO:* This variant removes LNS from MNS. Comparing this variant with MOMNS/D can show the

TABLE I
STATISTICS OF PERFORMANCE COMPARISONS OF MOMNS/D AND ITS THREE VARIANTS ON REAL-WORLD INSTANCES

HV	$w/t/l$	$R+$	$R-$	p-value	α = 0.05	α = 0.10
MOMNS/D vs MOMNS/D-G	30/0/0	465.0	0.0	1.7344e-006	YES	YES
MOMNS/D vs MOMNS/D-GL	27/3/0	465.0	0.0	1.7344e-006	YES	YES
MOMNS/D vs MOMNS/D-GO	27/3/0	465.0	0.0	1.7344e-006	YES	YES

TABLE II
AVERAGE RANKING OF MOMNS/D AND ITS THREE VARIANTS BY FRIEDMAN TEST FOR REAL-WORLD INSTANCES

HV	Average ranking value	Final rank
MOMNS/D	1	1
MOMNS/D-GO	2.2333	2
MOMNS/D-GL	2.7667	3
MOMNS/D-G	4	4

contribution of the proposed LNS. In this variant, $P_G = 0.5$ and $P_L = 0.5$.

1) Comparison Results on Real-World Instances: Table S3, in the supplementary material, provides comparative results (mean and STD) of HV obtained by MOMNS/D and its three variants on real-world instances. MOMNS/D obtains best mean values on all instances.

Table I provides the statistics of performance comparisons of MOMNS/D and its three variants on real-world instances. In column $w/t/l$ of Table I, it shows that MOMNS/D significantly outperforms its three variants on 30, 27, and 27 instances, respectively. In terms of the multiproblem Wilcoxon signed-rank test, the obtained p values are less than 0.05 in all instances, which means that MOMNS/D is significantly better than its three variants.

The average ranking of MOMNS/D and its three variants by Friedman test on all instances is shown in Table II. MOMNS/D gets the first rank, followed by MOMNS/D-GO, MOMNS/D-GL, and MOMNS/D-G.

To visually demonstrate the performance of MOMNS/D and its three variants, the projections of nondominated solutions obtained in a single run of MOMNS/D on the instance 250-2-6 at f_2–f_4 and f_3–f_5 planes are compared with its three variants, respectively, as shown in Fig. 3. The single run is selected according to the way in [42] as follows. First, each algorithm is conducted 30 independent runs and thus 30 archives of solutions are obtained. Next, HV is calculated for each archive, and the average HV value is calculated over 30 runs. Finally, the single run with the closest HV to the average value is selected and reported in Fig. 3. It seems that $f_3 - f_5$ is a conflicting objective pair, as analyzed in [4]. It is clear that the solution set obtained by MOMNS/D is generally wider and further from the reference point, which leads to better HV value.

2) Comparison Results on Traditional Instances: Table S4, in the supplementary material, provides comparative results (mean and STD) of HV obtained by MOMNS/D and its three variants on traditional instances. It shows that MOMNS/D, MOMNS/D-GO, and MOMNS/D-GL obtain best mean values on 35, 12, and 9 instances, respectively.

Table III provides the statistics of performance comparisons of MOMNS/D and its three variants on traditional instances. In column $w/t/l$ of Table III, it shows that MOMNS/D significantly outperforms MOMNS/D-G, MOMNS/D-GL, and MOMNS/D-GO on 42, 27, and 27 instances, respectively,

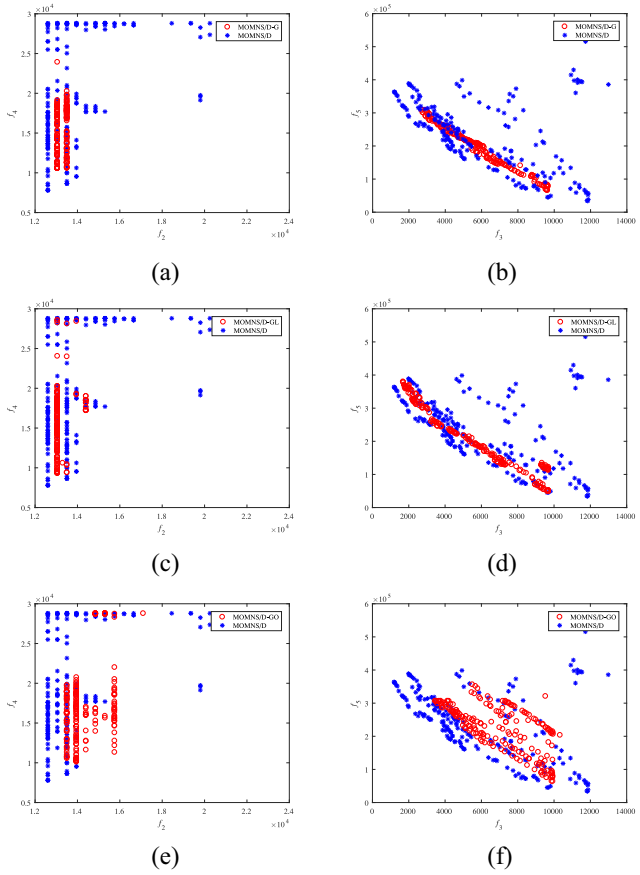


Fig. 3. Nondominated solutions obtained by MOMNS/D and its three variants on instance 250-2-6 in a single run. (a) MOMNS/D and MOMNS/D-G at $f_2 - f_4$. (b) MOMNS/D and MOMNS/D-G at $f_3 - f_5$. (c) MOMNS/D and MOMNS/D-GL at $f_2 - f_4$. (d) MOMNS/D and MOMNS/D-GL at $f_3 - f_5$. (e) MOMNS/D and MOMNS/D-GO at $f_2 - f_4$. (f) MOMNS/D and MOMNS/D-GO at $f_3 - f_5$.

while it is outperformed by MOMNS/D-G, MOMNS/D-GL, and MOMNS/D-GO on only 1, 3, and 9 instances, respectively. In terms of the multiproblem Wilcoxon signed-rank test, the obtained p values are less than 0.05 on all instances, which means that MOMNS/D is significantly better than its three variants.

Table IV shows the ranking of MOMNS/D and its three variants on all traditional instances by Friedman test. It shows that MOMNS/D ranks first, followed by MOMNS/D-GL, MOMNS/D-GO, and MOMNS/D-G.

Fig. 4 shows the nondominated solutions obtained by MOMNS/D and its three variants on a selected instance RC202 in a single run. The solution set obtained by MOMNS/D is generally wider and further from the reference point, which leads to better HV values.

3) Summary: To analyze the effect of MNS for MOMNS/V, three variants of MOMNS/V are also designed and tested for comparison. Tables S5 and S6, in the supplementary material, provide the comparative results obtained by MOMNS/V and its three variants. Tables S9–S12, in the supplementary material, summarize the statistics of performance comparisons. Results show that MOMNS/V is better than its three variants. In summary, the proposed MOMNS/D (MOMNS/V) performs better than its variants on both real-world and

TABLE III
STATISTICS OF PERFORMANCE COMPARISONS OF MOMNS/D AND ITS THREE VARIANTS ON TRADITIONAL INSTANCES

HV	$w/t/l$	$R+$	$R-$	p-value	α = 0.05	α = 0.10
MOMNS/D vs MOMNS/D-G	42/13/1	1515.0	81.0	4.9564e-009	YES	YES
MOMNS/D vs MOMNS/D-GL	27/26/3	1193.0	403.0	0.0013	YES	YES
MOMNS/D vs MOMNS/D-GO	27/20/9	1346.0	250.0	7.8190e-009	YES	YES

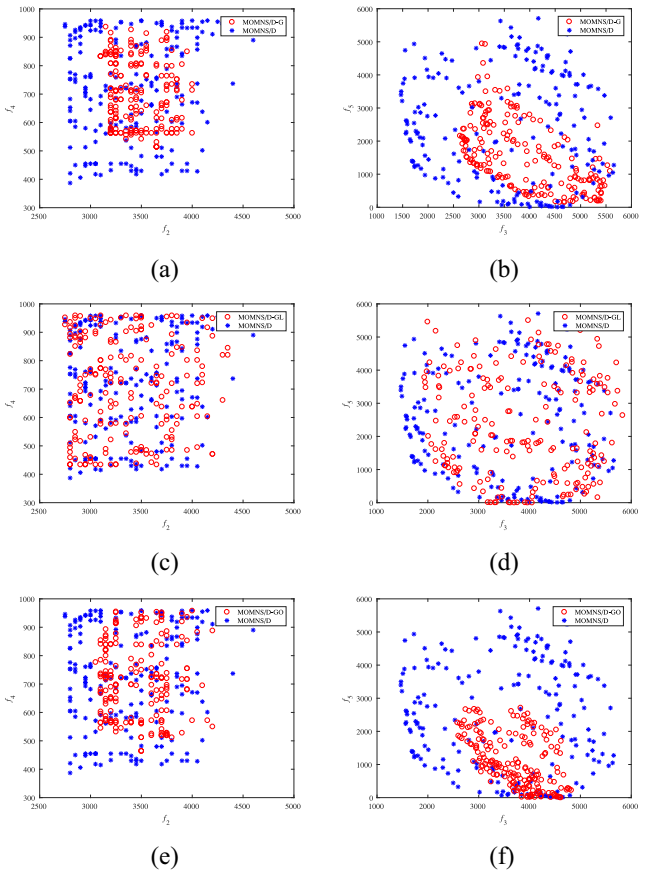


Fig. 4. Nondominated solutions obtained by MOMNS/D and its three variants on instance RC202 in a single run. (a) MOMNS/D and MOMNS/D-G at f_2–f_4. (b) MOMNS/D and MOMNS/D-G at f_3–f_5. (c) MOMNS/D and MOMNS/D-GL at f_2–f_4. (d) MOMNS/D and MOMNS/D-GL at f_3–f_5. (e) MOMNS/D and MOMNS/D-GO at f_2–f_4. (f) MOMNS/D and MOMNS/D-GO at f_3–f_5.

TABLE IV
AVERAGE RANKING OF MOMNS/D AND ITS THREE VARIANTS BY FRIEDMAN TEST FOR TRADITIONAL INSTANCES

HV	Average ranking value	Final rank
MOMNS/D	1.6964	1
MOMNS/D-GL	2.1786	2
MOMNS/D-GO	2.3929	3
MOMNS/D-G	3.7321	4

traditional instances. Thus, the effectiveness of the proposed MNS strategy is confirmed.

The superiority of MOMNS/D (MOMNS/V) to its three variants on real-world instances is a little more significant than on traditional instances. The reason may be that it is not suitable for conducting a proper multiobjective study on traditional instances, as pointed out in [4]. Real-world instances show strong conflict relationship between f_3 and f_5, as shown in Fig. 3(b), (d), and (f), while traditional instances show weak conflict relationship between f_3 and f_5, as shown in Fig. 4(b), (d), and (f). As shown in Fig. 4(b), (d), and (f), the solutions in PF of traditional instance are clustered, while the solutions in PF of real-world instance spread along the diagonal line of figure as shown in Fig. 3(b), (d), and (f). Thus, future researches are suggested to test their algorithms focusing on the proposed real-world instances.

TABLE V
STATISTICS OF PERFORMANCE COMPARISONS OF MOMNS/D AND MOMNS/V ON REAL-WORLD INSTANCES

HV	$w/t/l$	$R+$	$R-$	p-value	α = 0.05	α = 0.10
MOMNS/D vs MOMNS/V	7/8/15	145.0	320.0	0.0719	NO	YES

TABLE VI
STATISTICS OF PERFORMANCE COMPARISONS OF MOMNS/D AND MOMNS/V ON TRADITIONAL INSTANCES

HV	$w/t/l$	$R+$	$R-$	p-value	α = 0.05	α = 0.10
MOMNS/D vs MOMNS/V	27/7/22	791	805	0.9545	NO	NO

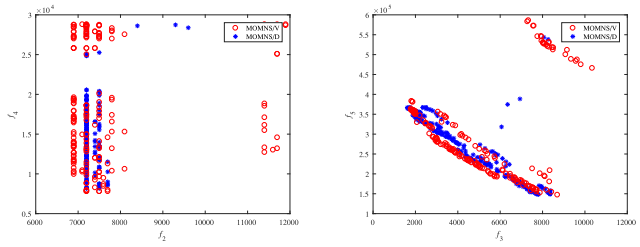
E. Comparisons Between MOMNS/D and MOMNS/V

Table S7, in the supplementary material, provides the HV values of MOMNS/D and MOMNS/V on real-world instances. It shows that MOMNS/D obtains best values on 10 instances, while MOMNS/V obtains best values on 20 instances. Table V shows the statistics of performance comparison of MOMNS/D and MOMNS/V on real-world instances. We can see that MOMNS/D significantly outperforms MOMNS/V on 7 instances, while it is significantly outperformed by MOMNS/V on 15 instances.

Table S8, in the supplementary material, provides the HV values of MOMNS/D and MOMNS/V on traditional instances. MOMNS/D obtains best values on 32 instances, while MOMNS/V obtains best values on 24 instances. In Table VI, it provides the statistics of performance comparison of MOMNS/D and MOMNS/V on traditional instances. It shows that MOMNS/D significantly outperforms MOMNS/V on 27 instances, while it is significantly outperformed by MOMNS/V on 22 instances.

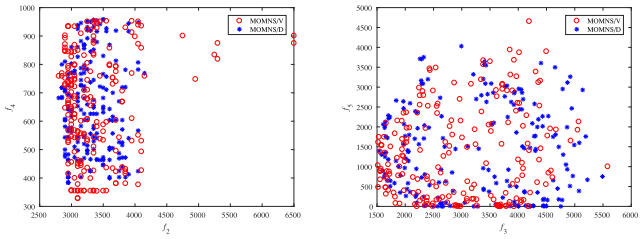
To visually demonstrate the performance of MOMNS/D and MOMNS/V, Figs. 5 and 6 show the nondominated solutions obtained by MOMNS/D and MOMNS/V on a selected real-world instance 150-3-4 and a traditional instance RC207, respectively, in a single run. It is also shown that the solution set obtained by MOMNS/V is generally wider and further from the reference point, which leads to better HV value.

On real-world instances, it shows that the performance of MOMNS/V is better than MOMNS/D in column $w/t/l$ of Table V. On traditional instances, it shows that the performances of MOMNS/V is outperformed by MOMNS/D in column $w/t/l$ of Table VI. Additionally, among traditional instances, it shows from Table S8, in the supplementary material, that MOMNS/D is better on instances C1XX, R1XX, and RC1XX, while MOMNS/V is better on instances C2XX, R2XX, and RC2XX.



(a) (b)

Fig. 5. Nondominated solutions obtained by MOMNS/D and MOMNS/V on instance 150-3-4 in a single run. (a) MOMNS/V and MOMNS/D at f_2–f_4. (b) MOMNS/V and MOMNS/D at f_3–f_5.



(a) (b)

Fig. 6. Nondominated solutions obtained by MOMNS/D and MOMNS/V on instance RC207 in a single run. (a) MOMNS/D and MOMNS/V at f_2–f_4. (b) MOMNS/D and MOMNS/V at f_3–f_5.

In summary, MOMNS/V performs better than MOMNS/D on real-world instances and traditional instances C2XX, R2XX, and RC2XX. On the one hand, as explained in [4], traditional instances with narrow time windows (C1XX, R1XX, and RC1XX) show weaker correlation between different objectives, thus are not suitable to be addressed as real multiobjective instances. Traditional instances with wider time windows (C2XX, R2XX, and RC2XX) present a sound but still not ideal benchmark scenario for multiobjective optimization, while real-world instances show stronger correlation between different objectives, which indicates a more truly multiobjective nature.

On the other hand, MOMNS/D and MOMNS/V use different selection mechanisms to maintain the population. Since MOMNS/D is a decomposition-based algorithm, it needs to specify a set of weight vectors, which significantly affects the performance in terms of diversity [46]. The configuration of weight vectors is not always an easy task, especially when the true PF of MO-FSMLRPTW is unknown. In the selection operator of MOMNS/V, no weight vectors need to be specified beforehand. The evolutionary process is dynamically guided by the current population [12], which shares a similar idea as in some adaptive decomposition algorithms. In the adaptive algorithms, the weight vectors are adaptively adjusted according to the distribution of the current population, and the search directions are also adaptively changed. MOEA based on decomposition with fixed weights promotes convergence, while MOEA based on decomposition with adaptive weights promotes diversity at the cost of convergence [47]. This may be the reason that MOMNS/V obtains a desirable distribution of solutions on real-world instances with strong conflict relationships among objectives.

VII. Discussions

A. Effect of the Selection Probabilities of GLS, OLS, and LNS

In MNS, $P_G = 0.4$, $P_O = 0.4$, and $P_L = 0.2$ is adopted. GLS and OLS are set to relatively large selection probabilities with the same values to optimize all objectives for convergence, while LNS is set to a relatively small selection probability to help GLS and OLS escape from local minima sometimes. In order to study the sensitivity of such settings, P_G, P_O, and P_L are set to the same value for comparisons. That is, three kinds of operators, GLS, OLS, and LNS, are selected uniformly. MOMNS/D with uniform selection probabilities is named MOMNS/D-U here as a baseline. Without priori knowledge or bias about performances of different kinds of neighborhood search operators, it is simple and natural to adopt uniform selection probabilities for different kinds of operators. By comparing MOMNS/D with MOMNS/D-U, the effective of the parameter setting rule used in our MNS can also be checked.

Tables S13 and S14, in the supplementary material, provide the comparison results of MOMNS/D and MOMNS/D-U on nine selected real-world instances and nine selected traditional instances, respectively. Table S13, in the supplementary material, shows that MOMNS/D significantly outperforms MOMNS/D-U on large size instances, while it is significantly outperformed by MOMNS/D-U on small and medium size instances. Table S14, in the supplementary material, shows that there is no significant difference between the performance of MOMNS/D and MOMNS/D-U on traditional instances. However, the average HV value of MOMNS/D is better than that of MOMNS/D-U on 5 out of 9 instances. The selection probabilities used in MOMNS/D are more suitable to traditional instances and large size real-world instances due to promoting convergence. However, the performance difference between MOMNS/D and MOMNS/D-U is not so significant, which indicates the algorithm is not sensitive to the parameter settings of GLS, OLS, and LNS.

In this paper, the selection probabilities are manually set since the main aim is to show the efficiency of the combination of three different kinds of neighborhood search operators. However, this simple static parameter setting strategy may not the best option to deal with different kinds of problems or instances. In order to further improve the performance, adaptive parameter control (operator selection) strategy can be designed, such as adaptive operator selection with bandits [48] and intelligent neighborhood selection [49], in the future. Comprehensive review and comparison of existing multiobjective adaptive operator selection can be found in [50]. One possible design principle of adaptive strategy for MNS is as follows. Different instances, or different stages during the search progress on the same instance, need different parameter settings of the selection probabilities. Thus, the real-time evolutionary states of individuals or population in each

generation can be estimated and monitored. Then, the search behavior of each individual, determined by the parameters and operators, in MOMNS/D or MOMNS/V can be adaptively controlled according to the real-time evolutionary states estimated. Finally, different individuals can play different roles during the search progress, and even the same individual can play different roles during different search progress.

B. Effect of Scalarizing Functions

Similar to the previous MOMAs proposed in [32] and [33] for multiobjective combinatorial optimization problems, weighted sum function is adopted in the proposed algorithms. If the properties of the problem are unknown (particularly the shape of the PF), the weighted sum function may not be the best option to deal with the problem. Thus, MOMNS/D using Techbycheff function, named MOMNS/D-T, is designed to study the effect of scalarizing functions.

Tables S13 and S14, in the supplementary material, provide the comparison results of MOMNS/D and MOMNS/D-T on 9 selected real-world instances and 9 selected traditional instances, respectively, with different configurations. Table S13, in the supplementary material, shows that MOMNS/D significantly outperforms MOMNS/D-T on 7 instances, while it is significantly outperformed by MOMNS/D-T on 1 instance. Table S14, in the supplementary material, shows that MOMNS/D significantly outperforms MOMNS/D-T on 8 instances, while it is significantly outperformed by MOMNS/D-T on 1 instance. Thus, the performance of MOMNS/D for the selected instances deteriorates when Techbycheff function is used.

In the proposed algorithms, any existing scalarizing function, in principle, can be freely used in the decomposition or local search for transforming MO-FSMLRPTW into a number of single-objective optimization problems. In fact, existing scalarizing functions have their own strengths and drawbacks [51], hence improved scalarizing functions [52], [53], ensemble of different scalarizing functions [54], and new and more complex scalarizing functions [51], [55] are proposed recently. However, the study with respect to scalarizing functions is beyond the scope of this paper. In the future, new scalarizing functions [55] can be easily adopted in the proposed algorithms to test possible improvement.

C. Comparisons With the State-of-the-Art Algorithms

MOEA/D [13] is a representative decomposition-based MOEA, and VaEA [12] is a representative Pareto-based MOEA proposed recently for many-objective optimization. In MOMNS/D-G (a variant of MOMNS/D), GLS is introduced in MOEA/D framework, thus MOMNS/D-G can be seen as a strong MOMA competitor based on MOEA/D for comparison. Similarly, MOMNS/V-G (a variant of MOMNS/V) can also be seen as a strong MOMA competitor based on VaEA for comparison. Thus, both MOMNS/D-G and MOMNS/V-G are not only the variants of the proposed algorithms, but also can be seen as the state-of-the-art MOMAs, as existing MOMAs [32], [33], for MO-FSMLRPTW since no existing algorithm directly utilizes the multiobjective optimization methods for FSMLRPTW.

MNS, as classical or general local search, is a single-trajectory neighborhood search algorithm. It also can be embedded into any population-based MOEA. In this paper, the proposed MNS is embedded into two representative MOEA frameworks, MOEA/D and VaEA, respectively. The resulting algorithms are MOMNS/D and MOMNS/V, respectively. The performance difference between MOMNS/D and MOMNS/V is also provided and analyzed in Section VI-E. On the one hand, these frameworks are very simple, which can be easily understood and adopted for solving real-world problems, such as MO-FSMLRPTW, by algorithm engineer even outside the field of MOEA. On the other hand, these frameworks can be seen as the current state-of-the-art MOEAs. For example, VaEA [12] is proposed recently and obtained better results than most of existing MOEAs. Hence, both MOMNS/D and MOMNS/V can be seen as new state-of-the-art MOMAs for MO-FSMLRPTW, which can be compared by later works. In the future, the proposed operators can also be embedded into other MOEAs, for example, MOEA/D variants proposed by Li *et al.* [14], [15], [17] and Wu *et al.* [16].

Crossover is not adopted in our algorithms in consistency with the existing multiobjective algorithms [21]–[28]. The previous experience [23] showed that the crossover of solutions in a highly constrained problem almost always produces infeasible solutions. Certainly, this means that good crossover operators or repairing heuristics need to be designed, and thus the application of crossover-based MOEAs to MO-FSMLRPTW can be considered in our future work.

D. Correlation Between Objectives

Understanding the relationships between objectives in a many-objective combinatorial optimization problem is important for developing tailored and efficient problem-solving techniques. This paper adopts the method proposed recently in [56] to visualize and analyze relationships between objectives. Two real-world instances (150-4-6 and 250-4-6) and two traditional instances (C105 and R101) are selected as representatives for correlation analysis. Four steps are used as in [56], and the corresponding analysis results are provided in Figs. S1–S6 in the supplementary material. From these figures, observations can be found as follows.

1) Fig. S1, in the supplementary material, shows that three pairs of objectives (f_3-f_4, f_3-f_5, and f_4-f_5) have either high harmonious or conflicting relationships. Fig. S2, in the supplementary material, shows that the objective ranges for most objectives are very large. It indicates that the selected four instances have conflicting objectives. Although there are solutions with good values for a given objective, at least one other objective has a poor value. These strong relationships between objectives indicate that both real-world instances and traditional instances provide interesting multiobjective challenges. The pairwise correlation values in Fig. S1, in the supplementary material, and the scatter plots in Fig. S6, in the supplementary material, show that $f_3 - f_4$ has

higher harmonious relationships and $f_3 - f_5$ has higher conflicting relationships in real-world instances than traditional instances. In summary, all the instances provide interesting multiobjective challenges. Moreover, real-world instances show stronger dependency relationships between objectives than traditional instances.

2) Figs. S4 and S5, in the supplementary material, show that the fitness landscapes of problems in traditional instances are more alike than those in real-world instances. That is, there are recurring features in the fitness landscapes when searching for solutions to different problem instances. This characteristic can be exploited by solving one instance of a given problem scenario using computationally expensive multiobjective algorithms to obtain a good approximation set and then using goal programming with efficient single-objective algorithms to solve other instances of the same problem scenario. More details can be found in [57] and [58].

Other properties of MO-FSMLRPTW, including the nature of the PS, the shape of the PF, and the correlation (in the decision space) of approximated Pareto optimal solutions should be further studied in the future. One possible research thought is as follows. The solutions generated by the proposed algorithms may hold valuable knowledge that can give a better understanding of MO-FSMLRPTW. Data mining of these solutions by machine learning methods can reveal interesting properties about the Pareto optimal solutions and further can help discover domain-specific knowledge [59], [60].

VIII. Conclusion

This paper considers FSMLRPTW with six objectives and proposes two versions of multiobjective MNS algorithms, MOMNS/D and MOMNS/V. In MNS strategy, three different kinds of neighborhood search operators are carefully designed and combined in a synergistic manner. The experimental results show the effectiveness of MNS. Two versions of multiobjective MNS algorithms are compared with each other.

In the future, this paper can be extended from several aspects. First, the proposed algorithms can be extended to solve other variants of multiobjective VRP [5], [6], [61], [62]. Further, the proposed MNS strategy can be introduced into other advanced MOEA frameworks (e.g., [63]–[65]) to solve multiobjective combinatorial optimization problems (e.g., [66]–[72]). Second, adaptive coordination strategy of GLS, OLS, and LNS in MNS can be developed by measuring their contributions during the evolutionary process. Finally, the properties of MO-FSMLRPTW problem can be further studied by mining the nondominated solutions obtained by the proposed algorithms.

References

[1] C. Prodhon and C. Prins, "A survey of recent research on location-routing problems," *Eur. J. Oper. Res.*, vol. 238, no. 1, pp. 1–17, 2014.

[2] M. Schneider and M. Drexl, "A survey of the standard location-routing problem," *Ann. Oper. Res.*, vol. 259, nos. 1–2, pp. 389–414, 2017.

[3] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm," *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 33–51, 2016.

[4] 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.

[5] 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.

[6] 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.

[7] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.

[8] J. Castro-Gutierrez, "Multi-objective tools for the vehicle routing problem with time windows," Ph.D. dissertation, School Comput. Sci., Univ. Nottingham, Nottingham, U.K., 2012.

[9] F.-H. Liu and S.-Y. Shen, "The fleet size and mix vehicle routing problem with time windows," *J. Oper. Res. Soc.*, vol. 50, no. 7, pp. 721–732, 1999.

[10] 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.

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

[12] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, Feb. 2017.

[13] 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.

[14] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[15] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2014.

[16] M. Wu, K. Li, S. Kwong, Y. Zhou, and Q. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 554–568, Aug. 2017.

[17] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[18] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[19] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

[20] 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.

[21] Y. Xu and R. Qu, "Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighbourhoods," *J. Oper. Res. Soc.*, vol. 62, no. 2, pp. 313–325, 2011.

[22] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, Dec. 2011.

[23] E. K. Burke and J. D. L. Silva, "The influence of the fitness evaluation method on the performance of multiobjective search algorithms," *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 875–897, 2006.

[24] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[25] B. Derbel, A. Liefooghe, Q. Zhang, H. Aguirre, and K. Tanaka, *Multi-Objective Local Search Based on Decomposition* (LNCS 9921). Cham, Switzerland: Springer, 2016, pp. 431–441.

[26] J. Shi, Q. Zhang, and J. Sun, "PPLS/D: Parallel Pareto local search based on decomposition," *IEEE Trans. Cybern.*, to be published.

[27] A. Liefooghe, J. Humeau, S. Mesmoudi, L. Jourdan, and E.-G. Talbi, "On dominance-based multiobjective local search: Design, implementation and experimental analysis on scheduling and traveling salesman problems," *J. Heuristics*, vol. 18, no. 2, pp. 317–352, 2012.

[28] M. Basseur, A. Liefooghe, K. Le, and E. K. Burke, "The efficiency of indicator-based local search for multi-objective combinatorial optimisation problems," *J. Heuristics*, vol. 18, no. 2, pp. 263–296, 2012.

[29] E.-G. Talbi, M. Basseur, A. J. Nebro, and E. Alba, "Multi-objective optimization using metaheuristics: Non-standard algorithms," *Int. Trans. Oper. Res.*, vol. 19, nos. 1–2, pp. 283–305, 2012.

[30] A. Blot, M.-É. Kessaci, and L. Jourdan, "Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation," *J. Heuristics*, vol. 24, no. 6, pp. 853–877, 2018.

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

[32] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151–165, Apr. 2011.

[33] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 336–353, Jan. 2015.

[34] H. Ishibuchi, M. Yamane, and Y. Nojima, *Difficulty in Evolutionary Multiobjective Optimization of Discrete Objective Functions With Different Granularities* (LNCS 7811). Heidelberg, Germany: Springer, 2013, pp. 230–245.

[35] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows," *Comput. Oper. Res.*, vol. 64, pp. 11–27, Dec. 2015.

[36] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 346–359, 2012.

[37] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, 2006.

[38] 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.

[39] Z. Fu, R. Eglese, and L. Y. O. 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.

[40] P. A. N. 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.

[41] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms: A comparative case study," in *Proc. Int. Conf. Parallel Prob. Solving Nat.*, 1998, pp. 292–301.

[42] 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.

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

[44] 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.

[45] 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.

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

[47] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Towards understanding the cost of adaptation in decomposition-based optimization algorithms," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2013, pp. 615–620.

[48] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.

[49] J. Molina, A. López-Sánchez, A. G. Hernández-Díaz, and I. Martínez-Salazar, "A multi-start algorithm with intelligent neighborhood selection for solving multi-objective humanitarian vehicle routing problems," *J. Heuristics*, vol. 24, no. 2, pp. 111–133, 2018.

[50] N. Hitomi and D. Selva, "A classification and comparison of credit assignment strategies in multiobjective adaptive operator selection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 294–314, Apr. 2017.

[51] 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.

[52] 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.

[53] X. Ma, Q. Zhang, G. Tian, 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.

[54] 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.

[55] M. M. Drugan, "Scaling-up many-objective combinatorial optimization with cartesian products of scalarization functions," *J. Heuristics*, vol. 24, no. 2, pp. 135–172, 2018.

[56] R. L. Pinheiro, D. Landa-Silva, and J. Atkin, "A technique based on trade-off maps to visualise and analyse relationships between objectives in optimisation problems," *J. Multi Crit. Decis. Anal.*, vol. 24, nos. 1–2, pp. 37–56, 2017.

[57] R. L. Pinheiro, D. Landa-Silva, W. Laesanklang, and A. A. Constantino, "Using goal programming on estimated Pareto fronts to solve multiobjective problems," in *Proc. 7th Int. Conf. Oper. Res. Enterprise Syst. (ICORES)*, Funchal, Portugal, Jan. 2018, pp. 132–143.

[58] R. L. Pinheiro, D. Landa-Silva, W. Laesanklang, and A. A. Constantino, "An efficient application of goal programming to tackle multiobjective problems with recurring fitness landscapes," in *Proc. Commun. Comput. Inf. Sci.*, 2018, pp. 1–20.

[59] S. Bandaru, A. H. C. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part A—Survey," *Expert Syst. Appl.*, vol. 70, pp. 139–159, Mar. 2017.

[60] S. Bandaru, A. H. C. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part B—New developments and applications," *Expert Syst. Appl.*, vol. 70, pp. 119–138, Mar. 2017.

[61] 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.

[62] Y.-H. Jia *et al.*, "A dynamic logistic dispatching system with set-based particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1607–1621, Sep. 2018.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] J.-J. Wang and L. Wang, "A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[68] 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.

[69] L. Ma *et al.*, "Two-level master–slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 861–880, May 2019.

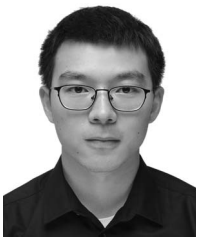
[70] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[71] 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.

[72] P. Wu, F. Chu, A. Che, and Y. Zhao, "Dual-objective optimization for lane reservation with residual capacity and budget constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

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.

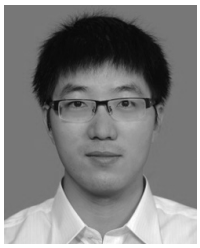


Liangsheng Yuan received the M.S. degree in computer science from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China, in 2018.

His current research interests include multiobjective optimization and its application to 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 interests include computational intelligence and its applications in production, transportation, and logistics.

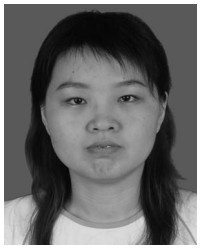


Shangce Gao (M'11–SM'16) received the Ph.D. degree in innovative life science from the University of Toyama, Toyama, Japan, in 2011.

He is currently an Associate Professor with the University of Toyama. His current research interests include nature-inspired technologies, mobile computing, and neural networks.

Yuyan Sun received the B.S. degree in computer science from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China, in 2017, where she is currently pursuing the M.S. degree in computer science.

Her current research interests include multiobjective optimization and its application to vehicle routing problems.



Yalan Zhou received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2008.

In 2008, she joined the Guangdong University of Finance and Economics, Guangzhou, where she is currently an Associate Professor. Her current research interests include artificial intelligence and its applications.