Solving Multitrip Pickup and Delivery Problem With Time Windows and Manpower Planning Using Multiobjective Algorithms

Jiahai Wang, *Senior Member, IEEE*, Yuyan Sun, Zizhen Zhang, and Shangce Gao, *Senior Member, IEEE*

***Abstract*—The multitrip pickup and delivery problem with time windows and manpower planning (MTPDPTW-MP) determines a set of ambulance routes and finds staff assignment for a hospital. It involves different stakeholders with diverse interests and objectives. This study firstly introduces a multiobjective MTPDPTW-MP (MO-MTPDPTWMP) with three objectives to better describe the real-world scenario. A multiobjective iterated local search algorithm with adaptive neighborhood selection (MOILS-ANS) is proposed to solve the problem. MOILS-ANS can generate a diverse set of alternative solutions for decision makers to meet their requirements. To better explore the search space, problem-specific neighborhood structures and an adaptive neighborhood selection strategy are carefully designed in MOILS-ANS. Experimental results show that the proposed MOILS-ANS significantly outperforms the other two multiobjective algorithms. Besides, the nature of objective functions and the properties of the problem are analyzed. Finally, the proposed MOILS-ANS is compared with the previous single-objective algorithm and the benefits of multiobjective optimization are discussed.**

***Index Terms*—Adaptive neighborhood selection, manpower planning, multiobjective optimization, multitrip, pickup and delivery problem with time windows.**

I. Introduction

THE vehicle routing problem (VRP) can be described as the problem of designing an optimal set of routes such that all the customers' requirements and the operational constraints are satisfied. The VRP has direct applications to everyday business routines of distribution or service-providing companies. A broad range of possible extensions to the VRP formulation are covered in [1]–[5]. Most research focuses on a widely used variant, called VRP with time windows (VRPTW). Thereafter, further extensions of the VRPTW, such as the pickup and delivery problem with time windows (PDPTW), the dial-a-ride problem (DARP) and the multitrip VRP with time windows (MTVRPTW), were proposed [6]–[8]. Besides, some research considers integrating the manpower planning into VRP since the driving of vehicles and the provision of services require the participation of manpower [9]. Recently, a more practical variant, called multitrip pickup and delivery problem with time windows and manpower planning (MTPDPTW-MP) was introduced in [9]. Relationship between this problem and other VRP extensions mentioned above is shown in Fig. 1.

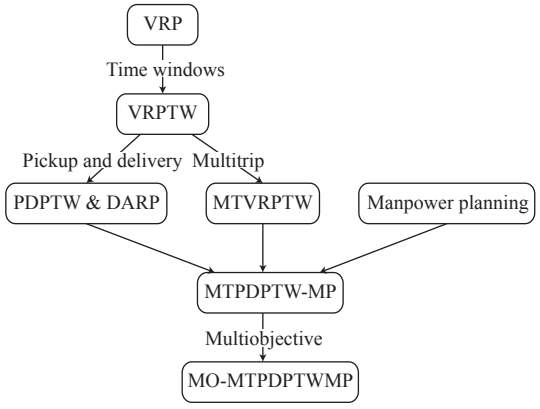


Fig. 1. Relationship between MO-MTPDPTWMP and various VRP extensions. The problem highlighted in bold is studied in this paper.

MTPDPTW-MP is a real-life healthcare problem originated from the application of Hong Kong public hospitals, China. Transportation services are provided to disabled or elderly patients between their residences and clinics. The ambulance routes satisfying a series of constraints should be designed and the staff assignment is also required [10], [11]. MTPDPTW-MP is an NP-hard problem of high complexity, as it is a combination of two well-known NP-hard problems (i.e., PDPTW and the staff scheduling problem). Usually, metaheuristic search techniques [12] are used to solve this kind of problems. In [9], an iterated local search (ILS) metaheuristic using a variable neighborhood descent (VND)

Manuscript received December 10, 2019; revised March 3, 2020, April 13, 2020; accepted April 25, 2020. This work was supported by the National Key R&D Program of China (2018AAA0101203), the National Natural Science Foundation of China (61673403, 71601191), and the JSPS KAKENHI (JP17K12751). Recommended by Associate Editor Shouguang Wang. *(Corresponding author: Zizhen Zhang.)*

Citation: J. H. Wang, Y. Y. Sun, Z. Z. Zhang, and S. C. Gao, "Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1134–1153, Jul. 2020.

J. H. Wang is with the Department of Computer Science, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, 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).

Y. Y. Sun and Z. Z. Zhang are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China (e-mail: sunyy5@mail2.sysu.edu.cn; zhangzzh7@mail.sysu.edu.cn).

S. C. Gao is with the Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan (e-mail: gaosc@eng.u-toyama.ac.jp).

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/JAS.2020.1003204

procedure in the local search phase, called ILS-VND, was proposed to deal with MTPDPTW-MP. [9] considered MTPDPTW-MP as a single-objective problem. It optimized the weighted sum of unserved requests, total traveling cost, and the workload deviation. A fixed weight vector was used, where the number of unserved requests was set as the most important objective. Only one final solution is returned to decision makers as ILS-VND optimizes multiple objectives in a single-objective manner.

In practice, MTPDPTW-MP must consider the conflicting interests of different stakeholders (i.e., the customers, hospital, and staff). According to [5], if we consider only one or two stakeholders in VRP variants, we may arrive at a local optimal solution because only one or two objectives are optimized in this situation. The interests of all stakeholders should be addressed in tandem [13]. Due to the problem structures of MTPDPTW-MP, the improvement of one objective may lead to the deterioration of other objectives. Therefore, MTPDPTW-MP is essentially a multiobjective optimization problem (MOP). Solving it in a single-objective manner requires extensive domain knowledge to determine the relative importance of different objectives. The simple combination of three objectives into a single one, as in [9], fails to provide decision makers with a comprehensive understanding of the relationship between objectives. It is necessary to present decision makers with a set of representative Pareto optimal solutions, instead of a unique optimum for MTPDPTW-MP. The advantage is that it can provide considerable flexibility in terms of a *posteriori* selection of a single preferred solution that best suits the current requirements of decision makers [13], [14].

This study defines a multiobjective MTPDPTW-MP (MO-MTPDPTWMP) with three objectives considering all stakeholders (e.g., the customers, hospital, and staff) to better reflect the real-world situation. The three objectives to be minimized include the number of unserved requests, the total traveling cost, and the workload deviation. The first one is customer-oriented: customers' requests should be served as many as possible. The second one is hospital-oriented, which can help to save money for the hospital. The third one is staff-oriented: the daily workloads for different staff members should not have large variances. Then, an algorithm called multiobjective iterated local search algorithm with adaptive neighborhood selection (MOILS-ANS) is developed to solve the problem. In the proposed MOILS-ANS, seven problem-specific neighborhood structures are adaptively selected in the local search process based on their performances.

The contributions of this study are as follows: 1) A multiobjective MTPDPTW-MP with three objectives considering the interests of all stakeholders is introduced. A multiobjective iterated local search algorithm with adaptive neighborhood selection (MOILS-ANS) is proposed to solve it. The proposed MOILS-ANS significantly outperforms the other two multiobjective algorithms. 2) The nature of objective functions of MO-MTPDPTWMP is analyzed and important properties of MO-MTPDPTWMP are revealed. 3) The proposed MOILS-ANS is compared with the previous single-objective algorithm and the benefits of multiobjective optimization are summarized.

The remainder of this paper is organized as follows. In Section II, problem formulation and related work are introduced. Thereafter, Section III provides a detailed description of the proposed MOILS-ANS. Experimental results are shown and analyzed in Section IV. Conclusions are drawn in Section V.

A. Problem Formulation and Related Work

A. MO-MTPDPTWMP

MTPDPTW-MP is formulated as a multiobjective problem (MO-MTPDPTWMP), which enables us to achieve a set of diverse and competitive solutions by addressing different objectives in a multiobjective manner. MO-MTPDPTWMP is a routing problem derived from the application of Hong Kong public hospitals, China. In this problem, the hospital can be regarded as the depot and ambulances can be regarded as vehicles. Each vehicle starts from the depot to accomplish some requests and returns to the depot within the maximum traveling duration. The goal of the problem is to design and schedule a set of optimal routes satisfying various constraints. Besides, the assignment of staff to vehicles should also be determined. A simple example with 8 requests and 2 vehicles is provided in Fig. 2 (a), in which there are 4 trips in total. The

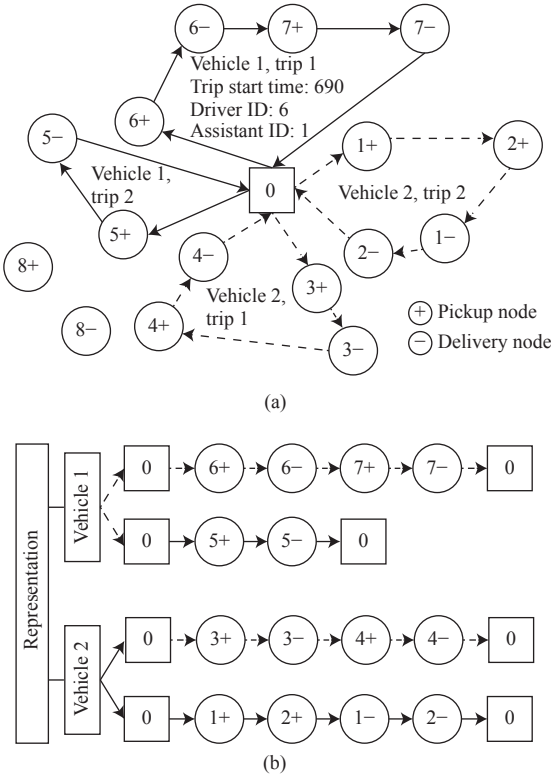


Fig. 2. Solution and its representation. (a) A possible solution; (b) Representation. A simple example with 8 requests and 2 vehicles is provided here. The depot node is 0. The pickup nodes are {1+, 2+, 3+, 4+, 5+, 6+, 7+, 8+} and their corresponding delivery nodes are {1−, 2−, 3−, 4−, 5−, 6−, 7−, 8−}. Request 8 is rejected owing to constraints. Solid lines represent trips for vehicle 1 while dashed lines represent trips for vehicle 2. The trip start time and staff ID are shown for vehicle 1, trip 1.

solution representation for this example is shown in Fig. 2 (b).

The inputs of the problem are as follows:

1) A complete undirected graph $G = (V, E)$, where V is the node set and E is the edge set.

2) The number of requests is N.

3) The node set $V = \{0, 1, \ldots, 2N\}$, where 0 denotes the depot and the remaining $2N$ nodes represent pickup or delivery nodes.

4) The edge set $E = \{(i, j) | i, j \in V, i \neq j\}$.

5) Each node $v \in V$ is associated with a non-negative service time t_v, a staff number requirement s_v, a seat number requirement q_v, and a time window $[e_v, l_v]$.

6) Each edge $(i, j) \in E$ is associated with a non-negative traveling cost c_{ij}. The traveling time is assumed to be the same as c_{ij} in the time-based dimension.

7) K denotes the set of available vehicles with heterogeneous capacity $Q_k (k \in K)$.

8) M denotes the set of available staff. Each staff member $m \in M$ is associated with a working period $[ES_m, LS_m]$, a lunch break $[e_m, l_m]$ and a working type (i.e., driver or assistant). The driver can serve as an assistant when necessary, but not vice versa.

The output is a set of Pareto non-dominated solutions.

A feasible solution must satisfy the following constraints $(C_1 - C_9)$, including pickup before delivery, capacity, time window, maximum duration, and staff demand constraints, which are defined as follows:

1) Pickup Before Delivery Constraint (C_1)*:* The pickup node of a specific request must appear before its delivery node and they should be in the same trip.

2) Vehicle Capacity Constraint (C_2)*:* The total number of staff and cumulated customers at each node cannot exceed the seat capacity of the assigned vehicle. Note that the driver of the vehicle does not occupy the seat.

3) Time Window Constraint (C_3)*:* The service can only start within the given time window $[e_v, l_v]$ of each node v. If the vehicle arrives at a certain node v earlier than e_v, it should wait until e_v to start the service.

4) Maximum Duration Constraint (C_4)*:* The duration of each trip must not exceed $T_{\max}$.

5) No Time Conflicting Constraint (C_5)*:* Any two trips of the same vehicle or the same staff member must have no time conflicting.

6) Disinfection Time Constraint (C_6)*:* The vehicles have to be disinfected after each trip. The disinfection time is half an hour.

7) Staff Demand Constraint (C_7)*:* Each trip must be allocated a driver and a sufficient number of assistants.

8) Staff Break Time Constraint (C_8)*:* After each trip, the staff must take a 30-minute break, which means that the staff can only start the next trip half an hour later.

9) Staff Working Time Constraint (C_9)*:* The staff are available during their working period $[ES_m, LS_m]$. Moreover, the staff should not work during their lunch break time $[e_m, l_m]$.

A solution of MO-MTPDPTWMP can be represented as a set of vehicle trips $\{r_1, r_2, ..., r_u\}$, where u is the number of trips. Each trip is a sequence of pickup and delivery nodes $\langle v_0, v_1, ..., v_p, v_{p+1} \rangle$, where v_0 and v_{p+1} represent the depot and p is the number of nodes in this trip. Thereafter, the traveling cost of this trip is defined by

$$C = \sum_{i=0}^{p} c_{v_i v_{i+1}}. \tag{1}$$

Each trip has an assigned vehicle and it is a route that satisfies the following requirements: 1) Each trip starts and terminates at the depot and consists of a sequence of pickup and delivery nodes. 2) The start time is given for each trip. 3) Adequate staff are assigned to the trip. 4) There is no time conflict between any two trips of the same vehicle or assigned to the same staff member.

The following objectives reflect the conflicting interests of all stakeholders: the customers, hospital, and staff.

1) The Number of Unserved Requests (f_1)*:* This objective is to be minimized to serve as many requests as possible. f_1 is a customer-oriented objective.

$$f_1 = U \tag{2}$$

where U is the number of unserved requests.

2) Total Traveling Cost (f_2)*:* The operational cost indicated by the traveling cost of vehicles should be minimized. f_2 is a hospital-oriented objective. According to (1), f_2 can be defined as

$$f_2 = \sum_{k=1}^{|K|} \sum_{j=1}^{u_k} C_{kj} \tag{3}$$

where u_k is the number of trips assigned to the vehicle k and C_{kj} denotes the traveling cost of trip j of vehicle k.

3) The Workload Deviation (f_3)*:* It minimizes the deviation of daily working time from the average time for each staff member. f_3 is a staff-oriented objective. f_3 can be defined as

$$f_3 = \sum_{j=1}^{|M|} |\omega_j - \bar{\omega}| \tag{4}$$

where ω_j is the total daily working time of staff member j and $\bar{w} = \dfrac{\sum_{j=1}^{|M|} \omega_j}{|M|}$ denotes the average working time.

More details about problem description and formulation can be found in [9].

B. Existing Algorithm for MTPDPTW-MP

ILS-VND was proposed in [9] to deal with MTPDPTW-MP, which is a single-objective optimization problem. A fixed weight vector for three objectives is used in the framework, where the number of unserved requests is set as the most important objective. As ILS-VND optimizes three objectives in a single-objective manner, only one final solution is returned to decision makers.

To the best of our knowledge, there is no publication dealing with the proposed MO-MTPDPTWMP. Solving it enables us to achieve a set of diverse and competitive solutions for decision makers by addressing different objectives in a multiobjective manner.

C. Multiobjective Optimization

MOPs are problems with two or more objectives to be optimized simultaneously. An MOP with m objectives can be defined as follows:

$$\text{Minimize } F(x) = (f_1(x), f_2(x), \ldots, f_m(x)), \quad x \in \mathbf{\Omega} \tag{5}$$

where $\mathbf{\Omega}$ is the decision space. Objective function $F : \mathbf{\Omega} \to \mathbb{R}^m$ maps the decision space to the objective space. Given two solutions $x, y \in \mathbf{\Omega}$, x is said to *dominate* y if and only if: $f_i(x) \le f_i(y)$ for all $i \in \{1, 2, ..., m\}$ and $f_j(x) < f_j(y)$ for at least one $j \in \{1, ..., m\}$. A solution x^* is *Pareto optimal* if there is no solution $x \in \mathbf{\Omega}$ such that x dominates x^*. *Pareto optimal* solutions are also called nondominated solutions in the solution space. *Pareto optimal* set (PS) consists of all *Pareto optimal* solutions. The image of solutions in PS in the objective space constitutes Pareto front (PF). It is defined by

$$PF = \{F(x) | x \in PS\}. \tag{6}$$

The goal of MOPs is to find a set of nondominated solutions with good performance in terms of both convergence and diversity. Convergence means that the solutions are close to the PF, while diversity means that the solutions are well distributed along the PF.

Metaheuristics are widely used algorithms for solving large and complex MOPs [15], [16]. Multiobjective metaheuristics algorithms can be roughly divided into two categories: multiobjective evolutionary algorithms (MOEAs) [15], including MOEA/D [17] and NSGA-II [18], and multiobjective local-search-based algorithms [16]. MOEAs simultaneously evolve a population of solutions. They show powerful exploration ability and can achieve high quality solutions for multiobjective continuous benchmark problems. Multiobjective local-search-based algorithms can promote search intensification and speed up convergence for multiobjective combinatorial optimization problems.

III. The Proposed Algorithm for MO-MTPDPTWMP

A. Overview of MOILS-ANS

MOILS-ANS is a multiobjective local search algorithm. At every iteration of MOILS-ANS, a single solution is chosen and the local search is used to explore its neighbors. A novel adaptive neighborhood selection (ANS) strategy is designed and used in the local search process of MOILS-ANS. The ANS strategy adaptively selects neighborhood structures according to their past performances.

Algorithm 1 shows the main framework of MOILS-ANS. Two explicit sets of solutions (i.e., the current solution set C and the archive A) are adopted in MOILS-ANS as suggested in [16]. These two sets enable the use of two different kinds of comparisons in the exploration procedure. Scalar-based comparisons are used in C while Pareto dominance-based comparisons are used in A. Besides, C can help to easily perform explorations outside A (e.g., to explore dominated neighboring solutions). In MOILS-ANS, firstly, $|C|$ initial solutions (Line 3) are constructed to form the current solution set C. The archive A is updated with solutions in C. Then N_λ uniformly distributed weight vectors are generated. At each iteration, a weight vector $\boldsymbol{\lambda}^j$ is randomly selected. Thereafter, the solution with the best value of the weighted sum function $f^{ws}(x|\boldsymbol{\lambda}^j)$ is selected from C. The variable *count* is initialized to 0. The algorithm executes *maxCount* iterations to explore the selected solution x. In each iteration, x is perturbed first to get x'. The perturbation method will be described in detail in Section III-D. And then ANS (Algorithm 2) is applied to x', and a new solution x'' is returned. The archive is updated with solution x''. If x'' is inserted into the archive A, *count* is restarted (Line 14). If one solution is explored without generating a new inserted solution in the archive *maxCount* times, the iteration ends. The current solution set C is updated with solution x (Line 20). If the weighted sum value $f^{ws}(x|\boldsymbol{\lambda}^j)$ is better than the original solution selected in Line 7, solution x will replace it in the current solution set C. When the termination condition is met, the archive A is returned.

The main components of MOILS-ANS, including the initial solution construction, ANS strategy, perturbation, feasibility checking, and archive updating are described in detail as follows.

Algorithm 1 MOILS-ANS

Input: the current solution set's size $|C|$, archive A, the number of weight vectors N_λ, *maxCount*
Output: archive A

```
1  begin
2    A = ∅;
3    initialize the current solution set C with |C| solutions and update the archive A;
4    generate N_λ uniformly distributed weight vectors {λ^1,...,λ^{N_λ}}, where λ^i = (λ^i_1,...,λ^i_m);
5    while termination condition is not met do
6      randomly select a weight vector λ^j from {λ^1,...,λ^{N_λ}};
7      the solution x with the best value of the weighted sum function f^ws(x|λ^j) is selected from C;
8      count = 0;
9      while count < maxCount do
10       perturb solution x to get x';
11       x'' = ANS_{λ^j}(x');
12       inserted = updateArchive(A, x'');
13       if inserted then
14         count = 0;
15         x = x'';
16       else
17         count = count + 1;
18       end if
19     end while
20     update the current solution set C with solution x;
21   end while
22   return archive A;
23 end
```

B. Initial Solution Construction

The basic idea of generating an initial solution is to insert one request into the solution each time. RI heuristic [19] is used when selecting a request to insert. The method is useful for highly constrained problems and can construct high-

quality initial solutions for further improvement. Relevant notations and formulas are as follows.

1) $\Delta C_{h,r}$ denotes the lowest incremental cost of inserting request h into a trip r.

2) If the request cannot be inserted into trip r, set $\Delta C_{h,r}$ as an infinitely large quantity.

3) For each request h, x_i is a variable indicating the trip for which request h has the ith lowest insertion cost.

4) The regret value of request h can be defined as

$$RV = \sum_{i=1}^{k} (\Delta C_{h,x_i} - \Delta C_{h,x_1}) \tag{7}$$

where the parameter k indicates reg-k insertion method. When selecting a request to insert, k is randomly selected from the interval 1 to R, where R is the number of trips in the current solution.

The request with the maximum RV is chosen to insert at each iteration as the maximum RV means that it may cost much if we insert it later. The initialization process ends when no request can be inserted anymore. All the requests that cannot be served are placed in the request pool.

C. The Adaptive Neighborhood Selection Strategy

Local search is one of general approaches with empirical success to combinatorial optimization problems. Its basic idea is that high-quality solutions can be found by iteratively improving a solution using modifications, called moves. A move type is specified by a neighborhood structure, which generates a neighborhood solution of the current solution [20]. The neighborhood solution of the current solution is evaluated after being generated. During the local search procedure, the weighted sum approach is often used to transform a multiobjective problem into a single-objective problem for evaluation as follows [2]:

$$f^{ws}(x|\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i f_i(x) \tag{8}$$

where $\boldsymbol{\lambda}$ is the weight vector.

1) Mutiple Neighborhood Structures: It is important to design multiple neighborhood structures in the local search when solving complex variants of the VRP [2], [6] or other combinatorial optimization problems [21], [22]. Using only one neighborhood structure may easily encounter the local optimum issue. Multiple neighborhood structures can broaden the exploration of the search space and help to escape the local optimum. In this paper, multiple neighborhood structures, $N_1 - N_7$, are designed for the local search of MO-MTPDPTWMP as follows.

a) N_1 [9]: This neighborhood structure deletes one request from the trip and places it into the request pool. Thereafter, the reg-k insertion method is used to try to insert all the requests in the pool to obtain a new solution. We try all deletion possibilities in the trips to generate neighboring solutions until the first improving solution is encountered.

b) N_2 [9]: This neighborhood structure deletes two successive requests in the same trip and places them into the request pool. Successive requests refer to those requests whose nodes (pickup or delivery) are adjacent. The reg-k insertion method is used to deal with all the requests in the request pool. All the deletion possibilities are tried until the first improving solution is encountered.

c) N_3 [9]: It is a type of exchange structure. Two requests h_i and h_j are deleted from two trips r_k and r_l in the current solution. Next, if possible, request h_i and request h_j are inserted into the best positions of trip r_l and trip r_k, respectively. The request is placed into the pool if it cannot be inserted. The reg-k insertion method is finally invoked for the requests in the pool. The number of neighboring solutions for the N_3 structure is $O(N^2)$. In general, the computational complexity of a neighborhood structure depends on the size of its neighborhood [20]. Considering the trade-off between computational complexity and the probability of improvement, we explore 10% of the neighboring solutions for each current solution until the first improving solution is found.

d) N_4 [23]: It is a type of reverse structure. Two nodes are randomly selected in a trip. All the nodes in-between are reversed. If a delivery node appears before its pickup node after reversing, exchange them. Then the reg-k method is used. The N_4 structure is used for all trips until the first improving solution is encountered.

e) N_5: This structure generalizes the N_3 structure. Two trips are selected. A sequence of successive nodes and their corresponding pickup or delivery nodes are deleted from trip r_k and inserted into trip r_l and vice versa. For the requests in the pool, the reg-k method is used. We explore all possible combinations of two trips until an improving solution is found.

f) N_6: This neighborhood structure generalizes request-exchange to trip-exchange. It exchanges two trips under two vehicles. This kind of exchange may obtain solutions with better multitrips scheduling as it changes the trips under vehicles. After the exchange, the reg-k method is invoked. All the exchange possibilities are tried until an improving solution is encountered.

g) N_7: This neighborhood structure deletes the trip with the fewest requests and puts them into the request pool. The reg-k insertion method is used for the remaining requests in the pool. The rationale of the proposed N_7 structure is that reducing the number of trips can reduce total disinfection time to a certain extent, since there are disinfection times between trips. Moreover, good solutions usually have as few routes as possible, since each additional route requires an additional edge and therefore, the total traveling cost is likely to increase [20].

In the neighborhood structures, all deletion/exchange operations are tried in a random order. Additionally, the reg-k is biased to the objective f_2 (travel cost) for calculating the regret value. In fact, a reg-k variant must select a request with the largest regret value to insert each time. At the time for insertion of request to a partial solution, only objective f_2 (travel cost) can be calculated. The other two objectives, the number of unserved requests (objective f_1) and the workload deviation (objective f_3), cannot be obtained until the whole solution has been constructed. However, new solutions

produced by the neighborhood structures are all evaluated by the weighted sum function (8), in which all objectives are considered simultaneously.

$N_1 - N_4$ are simple neighborhood structures commonly used in the literatures. N_6 and N_7 are designed in this paper. These neighborhood structures can be divided into three categories: N_1, N_2, and N_4 are intra-route structures corresponding to small modifications of solutions; N_3 and N_5 are inter-route structures, corresponding to moderate modifications of solutions; N_6 and N_7 are inter-vehicle structures, corresponding of large modifications to solutions. The combination of these three types of neighborhood structures in a synergistic manner is expected to provide a better balance of exploitation and exploration.

2) Adaptive Neighborhood Selection: How to flexibly coordinate multiple neighborhood structures to adapt to different instances of a problem or different stages of the search is a key challenge. In order to better combine multiple neighborhood structures, the concept of hyper-heuristics [24], [25] or adaptive operator selection [26] are adopted. Hence, this study proposes an adaptive neighborhood selection (ANS) strategy to tackle MO-MTPDPTWMP. Different from previous research [24]–[26], the proposed ANS is carefully designed for a multiobjective real-world combinatorial optimization problem rather than continuous benchmark problems or single-objective problems.

In each iteration of the local search, one neighborhood structure is selected to optimize the solution. The proposed ANS strategy adaptively selects neighborhood structures according to their previous performances, which is measured by the weighted-objective improvement z. The detailed calculation process is shown as follows:

$$z = \frac{f^{ws}(x|\boldsymbol{\lambda}) - f^{ws}(x'|\boldsymbol{\lambda})}{f^{ws}(x|\boldsymbol{\lambda})} \tag{9}$$

where x is the current solution and x' is the returned solution in local search.

By considering the weighted-objective improvement, neighborhood structures can be selected more appropriately during the local search process in ANS. There are $L = 7$ neighborhood structures for the local search, relevant notations about the ANS strategy are given as follows.

a) Neighborhood structures: $\{N_1, N_2, ..., N_L\}$.

b) The performance of neighborhood structures on improving the weighted-objective value of solutions: $\{z_1, z_2, ..., z_L\}$.

c) Selection probabilities of neighborhood structures: $\{p_1, p_2, ..., p_L\}$.

Given the above notations, the performance z_i and the selection probability p_i of neighborhood structure N_i in the ANS strategy are computed as follows:

a) Suppose that N_i improves the weighted-objective value of a solution, z, at a certain iteration, then z_i can be updated as follows:

$$z_i = (1-\alpha) \times z_i + \alpha \times z \tag{10}$$

where the adaptation rate $\alpha \in [0, 1]$ controls the importance of the recently performance compared to the cumulative performance in the past.

b) The better the performance of N_i on improving the solution quality in the past, the more probability it will be selected next. The selection probability of a neighborhood structure N_i is computed as follows:

$$p_i = p_{\min} + (1 - L \times p_{\min}) \times \frac{z_i}{\sum_{j=1}^{L} z_j} \tag{11}$$

where the minimum selection probability $p_{\min}$ is defined to foster the exploration of poorly performing neighborhood structures as in [26]. The factor $(1 - L \times p_{\min})$ makes the selection probability of all neighborhood structures summing up to 1. Moreover, (11) also ensures that the minimum selection probability of neighborhood structures is $p_{\min}$ [26].

The ANS strategy is shown in Algorithm 2. The input of ANS is a solution x and a weight vector $\boldsymbol{\lambda}$. The parameter I is the search depth of ANS. At each iteration of ANS, a neighborhood structure N_i is selected from $\{N_1, N_2, ..., N_L\}$ according to their selection probabilities with Roulette Wheel Rule. Then, the neighborhood structure N_i is applied to the current solution x. The returned solution of $N_i(x)$ is x'. Next, the weighted sum function (8) is used to compare x' and x. The performance and selection probabilities of neighborhood structures are updated as shown in Lines 7 and 9 of Algorithm 2, respectively. A total of I iterations are performed.

Algorithm 2 $\mathrm{ANS}_{\boldsymbol{\lambda}}(x)$

Input: a solution x, a weight vector $\boldsymbol{\lambda}$
Output: x
1 **begin**
2 **for** $depth = 1, \ldots, I$ **do**
3 select a neighborhood operator N_i from $\{N_1, N_2, \ldots, N_L\}$ according to their selection probabilities $\{p_1, p_2, \ldots, p_L\}$;
4 $x' = N_i(x)$;
5 **if** $f^{ws}(x'|\lambda) < f^{ws}(x|\lambda)$ **then**
6 calculate the weighted-objective improvement $\{z\}$ according to (9);
7 update the performance $\{z_i\}$ according to (10);
8 **for** $i = 1 : L$ **do**
9 update the selection probability p_i according to (11);
10 **end for**
11 $x = x'$;
12 **end if**
13 **end for**
14 return x;
15 **end**

D. Perturbation

Perturbation is generally used to escape local optimum in metaheuristics. Its idea is to change the solution x, but this change does not necessarily have to result in an improving solution [20]. Perturbation may drive the solution to a new region that basic local search cannot achieve. In general, a very large perturbation may behave like a random restart, while a small perturbation will limit the search space to a small region and cannot escape from local minimum.

Therefore, the neighborhood structure N_3 corresponding to moderate modifications to solutions is used as the perturbation mechanism in this study. We explore 10% of the neighboring solutions for the current solution until the first feasible solution is found, no matter it is improved or not.

E. Feasibility Checking

The feasibility of a temporal solution should be checked when a request is inserted. The feasibility checking procedure consists of three steps: single trip checking, multitrip scheduling, and staff assignment. We should first check whether the inserted request will cause a single trip violating the first four constraints ($C_1 - C_4$). If all the trips are legal, we should consider multitrips scheduling and staff assignment. If any constraints are violated during the three-step checking procedure, the temporal solution is considered illegal.

1) Single Trip Checking: The method is adopted from the segments concatenation developed by [7]–[9]. A segment is a sequence of consecutive nodes. For each trip, relevant information for all segments should be computed and saved in a preprocessing phase. When inserting or removing a request, different segments can be directly concatenated. The feasibility of a trip can be evaluated in constant time by using the relevant information of each segment.

For two segments $\sigma = \left(\sigma_i, ..., \sigma_j\right)$ and $\sigma' = \left(\sigma'_i, ..., \sigma'_j\right)$, where $\sigma_i, ..., \sigma_j$ and $\sigma'_i, ..., \sigma'_j$ are non-overlapping nodes in the node set V, the following information from the concatenated segment $\sigma \oplus \sigma'$ is computed ($\oplus$ represents the concatenation operator):

$$D(\sigma \oplus \sigma') = D(\sigma) + D(\sigma') + c_{\sigma_j \sigma'_i} + \Delta_{WT} \tag{12}$$

$$C(\sigma \oplus \sigma') = C(\sigma) + C(\sigma') + d_{\sigma_j \sigma'_i} \tag{13}$$

$$E(\sigma \oplus \sigma') = \max\{E(\sigma') - \Delta, E(\sigma)\} - \Delta_{WT} \tag{14}$$

$$L(\sigma \oplus \sigma') = \min\{L(\sigma') - \Delta, L(\sigma)\} \tag{15}$$

$$Q(\sigma \oplus \sigma') = Q(\sigma) + Q(\sigma') \tag{16}$$

$$PQ(\sigma \oplus \sigma') = \max\{PQ(\sigma), Q(\sigma) + PQ(\sigma')\} \tag{17}$$

$$SF(\sigma \oplus \sigma') = \max\{SF(\sigma), SF(\sigma')\} \tag{18}$$

where $\Delta = D(\sigma) + c_{\sigma_j \sigma'_i}$ and the minimal waiting time $\Delta_{WT} = \max\{E(\sigma') - \Delta - L(\sigma), 0\}$. $D(\sigma)$ and $C(\sigma)$ denote the corresponding minimum duration and cumulated cost, respectively. $E(\sigma)$ and $L(\sigma)$ denote the earliest start time and the latest start time, respectively, which form the start-time window of the trip. If the vehicle starts the service at any time in this time window, the trip will involve the least waiting time. More details about the start-time window can be referred to [9]. $Q(\sigma)$ represents the load after serving the last node of the segment. $PQ(\sigma)$ and $SF(\sigma)$ denote the maximum load and the maximum number of required staff at certain nodes of the segment, respectively.

The new segment $\sigma \oplus \sigma'$ is feasible if and only if the following four conditions are satisfied. i) Segments σ and σ' are feasible; ii) The time windows are not violated (i.e., $E(\sigma \oplus \sigma') \le L(\sigma \oplus \sigma')$); iii) The maximum duration constraint is satisfied (i.e., $D(\sigma \oplus \sigma') \le T_{\max}$); iv) $PQ(\sigma \oplus \sigma') + SF(\sigma \oplus \sigma') - 1 \le Q_{\max}$, where $Q_{\max}$ is the maximum capacity of the vehicles. It implies that there is at least one vehicle with adequate capacity to perform the trip. For a segment $\sigma_0 = (v)$ containing a single node v, $D(\sigma_0) = t_v$, $C(\sigma_0) = 0$, $E(\sigma_0) = e_v$, $L(\sigma_0) = l_v$, $Q(\sigma_0) = q_v$, $PQ(\sigma_0) = q_v$, and $SF(\sigma_0) = s_v$. More details about the above notations can be referred to [9].

2) Multitrip Scheduling: After the first step, all the trips are legal. Each vehicle may contain several trips. Multitrip scheduling should be performed to obtain a legal scheduling order for each vehicle.

After inserting or removing requests, a new trip is generated. First, we should check whether its original vehicle can still perform multitrip scheduling. If not, dispatch the trip to other vehicles until a legal scheduling order is found.

For a specific vehicle k, when a new trip is assigned, the capacity constraint should be checked first. If the capacity constraint is satisfied, multitrip scheduling is performed to determine an order for the assigned trips. The assigned trips are sorted by their earliest start times to generate a greedy order. The greedy order is usually feasible. Otherwise, all permutations of the trips are enumerated to obtain a legal order. The feasibility of a scheduling order $\langle r_1, r_2, ..., r_u \rangle$ can be examined by a forward sweep as follows:

$$E(r_i) = \max\{E(r_{i-1}) + D(r_{i-1}) + \delta, E(r_i)\}, \quad i = 2, ..., u \tag{19}$$

where δ is the disinfection time. The multitrip scheduling is feasible if $E(r_i) \le L(r_i)$ is satisfied for all trips. If the schedule is feasible, the start-time window for each trip is further narrowed with the following backward sweep:

$$L(r_i) = \min\{L(r_{i+1}) - D(r_i) - \delta, L(r_i)\}, \quad i = u - 1, ..., 1. \tag{20}$$

3) Staff Assignment: After the second step, the staff assignment procedure should be invoked for all the vehicles. A greedy heuristic is used to perform staff assignment.

Each time we select an unsettled trip with the earliest start time and try to assign a driver and sufficient assistants to the trip. $[ES_m, LS_m]$, $[e_m, l_m]$ and the working type are checked to see if staff m can be assigned to the current trip. When sufficient staff have been assigned, set the trip start time to the larger value between the maximum available time of the assigned staff and the earliest start time of the trip. The available time of all assigned staff should be updated. Moreover, the forward sweep method should be used to update the earliest start time of the corresponding trips. The process is iterated until all trips have sufficient staff or there is no feasible assignment for a trip.

The feasibility checking for MO-MTPDPTWMP is sophisticated. More details about the feasibility checking can be referred to [7]–[9].

F. Archive Updating

To obtain a set of Pareto nondominated solutions with good performance in terms of convergence and diversity, an external archive A with maximum size S is adopted. If no solutions in A dominate the newly generated solution x'' (Line 12) of Algorithm 1, x'' will be stored into A. All solutions

dominated by x'' will be removed from A. The archive A stores all nondominated solutions until $|A| > S$, where $|A|$ is the current size of A. If $|A| > S$, the parallel cell coordinate system (PCCS) proposed in [27] is used to decide which solution is discarded. PCCS is an advanced density evaluation mechanism. Details about PCCS density evaluation procedure can be found in [27]. The solution with the maximum density will be discarded when $|A| > S$. Experimental results about density estimation in [27] illustrates that the performance of PCCS is superior to that of adaptive grid and crowding distance in NSGA-II [18] in terms of convergence and diversity.

G. Complexity Analysis

The complexity of the proposed MOILS-ANS mainly depends on the local search in ANS strategy as shown in Algorithm 2, which has the highest complexity. The local search consists of different neighborhood structures. Among them, the most complex one is the N_3 neighborhood structure, with a complexity of $O(N^2)$. The complexity of the local search is $O(I \cdot N^2)$ since the search depth of ANS is I. Therefore, considering the iterated local search (the outer loop of Algorithm 1) process, the total complexity of MOILS-ANS is $O(maxCount \cdot I \cdot N^2)$.

IV. Experimental Studies

MOILS-ANS was coded in C++. All the experiments were conducted on a machine with Intel(R) Xeon(R) Gold 5118 2.30GHz CPU, 64 GB RAM, and the Ubuntu 16.04.5 operating system.

A. Parameter Settings

In MOILS-ANS, the size of the current solution set C is set to 50 and the archive size S is set to 100 [27]. N_λ uniformly distributed weighted vectors are generated. The number of weight vectors is calculated as C_{H+m-1}^{m-1} using a parameter H [28]. In this problem, $m = 3$ and H is set to 7, and thus $N_\lambda = C_9^2 = 36$. In addition, $maxCount = 15$.

In ANS, the search depth $I = 20$, $\alpha = 0.8$, and $p_{\min} = 0.1$. These parameters are selected empirically. In our preliminary experiment, alternative parameter settings are studied. For example, I is set to 10, 20, 50, and α is set to 0.2, 0.5, 0.8, respectively. Finally, $I = 20$ and $\alpha = 0.8$ can lead to the best result. $p_{\min} = 0.1$ is often used in the references (e.g., [26]), so it is directly adopted.

B. Benchmark Instances and Termination Criterion

Benchmark instances collected from public hospitals in Hong Kong, China are provided by [9]. An instance is constructed using the requests in a day. A total of 59 instances from January (31 instances) to February (28 instances) of 2009 are used to evaluate MOILS-ANS for MO-MTPDPTWMP. The format of an instance is 2009MMDD, where MM and DD denote the month and day, respectively. For example, 20090101 indicates daily requests on January 1st. There are 38 non-holiday instances and 21 holiday instances. In general, non-holiday instances have narrow time windows and extensive requests, while holiday instances have wide time windows and a few requests. The number of requests in all instances varies from 28 to 205. Each instance input consists of the following information:

1) Requests Data: available seats, staff demand, time windows, and service time.

2) Vehicle Data: vehicle number and vehicle capacity.

3) Staff Data: work period and lunch break for each staff.

4) Cost Matrix: traveling cost between any two locations.

The termination criterion is set to the maximum runtime [9]. Larger instances with more requests require more computation time, and thus the maximum runtime for each instance is set according to the number of requests as follows: 1800 seconds for $N \le 50$; 3600 seconds for $50 < N \le 100$; 7200 seconds for $N > 100$.

C. Performance Indicators and Statistics

1) Performance Indicators: The performance of a multiobjective algorithm is evaluated in terms of convergence and diversity of the nondominated solutions obtained. Hypervolume (HV) and inverted generational distance (IGD) are two widely used indicators for performance evaluation. Therefore, we also adopt them in this study.

a) HV: The volume of the space enclosed by the solution set and a reference point is calculated as the HV value. For a minimization task, the larger the HV value is, the closer the solution set is to the Pareto front. In this work, the HV value is calculated based on the objective values normalized into [0, 1]. The reference point is set to (1.2, 1.2, 1.2).

b) IGD: The distance of the elements in the approximate Pareto front towards those in the true Pareto front is estimated as the IGD value. A smaller IGD indicates a better approximate Pareto set.

The true Pareto front of each instance is unknown. For each instance, all nondominated solutions produced by all algorithms over 20 runs are collected to form an approximation of its true Pareto front.

Additionally, two indicators proposed by [13] are used to compare nondominated solutions obtained by multiobjective algorithms with the best solution reported by the previous single-objective algorithm for MTPDPTW-MP [9]. Suppose that A is the set of all nondominated solutions in the objective space generated by a multiobjective algorithm and s is the best solution in the objective space generated by a previous single-objective algorithm on the same problem instance, the two indicators can be defined as follows:

a) Generated dominating solutions (GDS): GDS is the number of solutions in A that can dominate s.

$$\mathrm{GDS}(A, s) = |\{a \in A | a \text{ dominates } s\}|. \tag{21}$$

b) Generated alternative solutions (GAS): GAS is the number of solutions in A that can be used as alternatives to s. If the sum of results, obtained by subtracting normalized objective values of s from a, is negative, then a can be considered as an alternative to s.

$$\mathrm{GAS}(A, s) = |\{a \in A | \sum_{i=1}^{m} \frac{a_i - s_i - z_i^*}{z_i^{\mathrm{nad}} - z_i^*} < 0\}| \tag{22}$$

where m is the number of objectives. Ideal point (z^*) and Nadir

point (z^{nad}) are constructed using the best and worst objective values of the *PF* respectively.

According to the above definitions, larger GDS and GAS indicate better nondominated solution sets.

2) Statistics by Wilcoxon and Friedman Tests: To test the significant differences between the results obtained by MOILS-ANS and other competitors on a single instance over 20 runs, single-problem Wilcoxon rank-sum test [29]–[31] at 5% significance level was carried out for each instance. Statistical results on sets of instances are summarized as $w/t/l$, which means that MOILS-ANS significantly performs better than, similar to and worse than the corresponding competitor on w, t, and l instances, respectively. Moreover, to show the differences between the results obtained by MOILS-ANS and the competitor on sets of instances, multiproblem Wilcoxon signed-rank test is conducted. Finally, to get the ranking of all algorithms on sets of instances, the Friedman test is applied.

D. Multiobjective Competitors

MO-MTPDPTWMP is a newly introduced problem in the literature and there is no existing algorithm applied to it yet. No existing results can be directly used for comparisons. Two competitor algorithms (MOILS-R and MOEA/D-ANS) are adopted for MO-MTPDPTWMP. Their characteristics are as follows.

1) MOILS-R: Unlike MOILS-ANS, MOILS-R randomly selects a neighborhood structure. The selection probabilities of different neighborhood structures are set to the same value. That is, seven neighborhood structures are selected uniformly. It is natural to adopt uniform selection probabilities for different neighborhood structures without prior knowledge about the performance of them. Comparing MOILS-ANS with MOILS-R, we can analyze the effect of the ANS strategy.

2) MOEA/D-ANS: MOEA/D [17] is adapted to form a state-of-the-art competitor for MO-MTPDPTWMP. MOEA/D is a representative MOEA [15]. A multiobjective optimization problem is decomposed into a number of scalar optimization subproblems in MOEA/D. These subproblems are optimized simultaneously. The algorithm optimizes each subproblem only using information from its several neighboring subproblems. According to [15], [17], [32], MOEA/D seems to be more suitable for tackling multiobjective combinatorial optimization problems than NSGA-II [18], because problem-specific (local search) techniques can be directly used to intensify the exploration of promising regions in the solution space. It provides a natural framework for using single-objective local search techniques [4]. ANS is embedded into MOEA/D to form a competitor called MOEA/D-ANS (shown in Algorithm 3).

For fair comparisons, the same algorithm components (initial solution construction, the ANS strategy, feasibility checking, and archive) described in Section III are also used in MOEA/D-ANS. The termination criterion of MOEA/D-ANS is also set to the maximum runtime as that in MOILS-ANS (Section IV-B). The number of weight vectors is set to $N_\lambda = 36$ and the archive size is set to $S = 100$. Such settings are the same as those in MOILS-ANS. The neighborhood size T is set to 6. Given the limited computation resource, MOEA/D-ANS with a larger population size, for example, $N_\lambda = 100$, does not lead to better results. MO-MTPDPTWMP only has three objectives (does not count as many objectives), and thus $N_\lambda = 36$ can generate satisfactory results by our experiment. $T = 6$ can lead to better balance between convergence and diversity. If T is too small (e.g., $T = 2$), the performance of convergence is deteriorated by our preliminary experiment.

Algorithm 3 MOEA/D-ANS

Input: the number of weight vectors N_λ, neighborhood size T, archive A
Output: archive A
1 **begin**
2 $A = \emptyset$;
3 initialize population $\mathcal{P}$ with N_λ solutions $x^1, \ldots, x^{N_\lambda}$ and update the archive A;
4 generate N_λ uniformly distributed weight vectors $\{\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^{N_\lambda}\}$, where $\boldsymbol{\lambda}^i = (\lambda_1^i, \ldots, \lambda_m^i)$;
5 compute the neighborhood set $B(i) = \{i_1, \ldots, i_T\}$ for each weight vector $\boldsymbol{\lambda}^i$, where $\boldsymbol{\lambda}^{i_1}, \ldots, \boldsymbol{\lambda}^{i_T}$ are the T closest weight vectors to $\boldsymbol{\lambda}^i$ based on the Euclidean distance;
6 **while** termination condition is not met **do**
7 **for** $i = 1: N_\lambda$ **do**
8 randomly select an index I from $B(i)$;
9 $x' = \text{ANS}_{\lambda i}(x^I)$;
10 **for** each $j \in B(i)$ **do**
11 **if** $f^{ws}(x'|\boldsymbol{\lambda}^j) \le f^{ws}(x^j|\boldsymbol{\lambda}^j)$ **then**
12 $x^j = x'$
13 **end if**
14 **end for**
15 updateArchive (A; x');
16 **end for**
17 **end while**
18 return archive A;
19 **end**

E. Comparison Results Among Multiobjective Algorithms

Numerical values of performance indicators (HV and IGD) over 20 independent runs are shown in Table S1 in Appendix due to the limited space. Statistics summarizing those numerical values are shown in Tables I–IV. Table S1 in Appendix shows the HV and IGD values (mean and standard deviation (STD)) on all 59 instances. In terms of HV and IGD, the proposed MOILS-ANS obtains best mean values on 49 and 54 instances, respectively. In comparison, MOILS-R obtains best mean values on 4 and 3 instances. MOEA/D-ANS obtains best mean values on 6 and 2 instances, respectively. The results suggest that MOILS-ANS significantly outperforms MOILS-R and MOEA/D-ANS.

1) Comparisons Between MOILS-ANS and MOILS-R: Table I provides the statistics of performance comparisons of MOILS-ANS and MOILS-R on 59 instances. Results in column $w/t/l$ show that MOILS-ANS outperforms MOILS-R on 47, 55 instances in terms of HV and IGD, respectively.

TABLE I
STATISTICS OF PERFORMANCE COMPARISONS BETWEEN MOILS-ANS AND MOILS-R

HV	$w/t/l$	R^+	R^-	p-value	$\alpha = 0.05$	$\alpha = 0.10$
MOILS-ANS vs MOILS-R	47/11/1	1745.0	25.0	3.1364E–015	YES	YES
IGD	$w/t/l$	R^+	R^-	p-value	$\alpha = 0.05$	$\alpha = 0.10$
MOILS-ANS vs MOILS-R	55/1/3	1744.0	26.0	3.7100E–015	YES	YES

TABLE II
STATISTICS OF PERFORMANCE COMPARISONS BETWEEN MOILS-ANS AND MOEA/D-ANS

HV	$w/t/l$	R^+	R^-	p-value	$\alpha = 0.05$	$\alpha = 0.10$
MOILS-ANS vs MOEA/D-ANS	48/8/3	1659.0	111.0	4.5952E–011	YES	YES
IGD	$w/t/l$	R^+	R^-	p-value	$\alpha = 0.05$	$\alpha = 0.10$
MOILS-ANS vs MOEA/D-ANS	46/11/2	1689.0	81.0	3.1537E–012	YES	YES

TABLE IV
THE z-VALUES, UNADJUSTED p-VALUES, AND ADJUSTED p-VALUES FOR THE FRIEDMAN TEST ALONG WITH HOLM'S POST-HOC PROCEDURE ACCORDING TO HV AND IGD AT 5% SIGNIFICANCE LEVEL

HV	z-values	Unadjusted p-values	Adjusted p-values
MOILS-R vs MOILS-ANS	6.996367	0.000000	0.016667
MOILS-ANS vs MOEA/D-ANS	6.259907	0.000000	0.025000
MOILS-R vs MOEA/D-ANS	0.736460	0.461451	0.050000
IGD	z-values	Unadjusted p-values	Adjusted p-values
MOILS-R vs MOILS-ANS	8.101057	0.000000	0.016667
MOILS-ANS vs MOEA/D-ANS	6.536080	0.000000	0.025000
MOILS-R vs MOEA/D-ANS	1.564977	0.117588	0.050000

TABLE III
AVERAGE RANKING OF MOILS-ANS, MOILS-R, AND MOEA/D-ANS BY FRIEDMAN TEST ACCORDING TO HV AND IGD

HV	Average ranking value	Final rank
MOILS-ANS	1.1864	1
MOEA/D-ANS	2.3390	2
MOILS-R	2.4746	3
IGD	Average ranking value	Final rank
MOILS-ANS	1.1017	1
MOEA/D-ANS	2.3051	2
MOILS-R	2.5932	3

Moreover, the obtained p values are less than 0.05 and MOILS-ANS obtains much higher R^+ values than R^- values on the multiproblem Wilcoxon signed-rank test. It means that MOILS-ANS is significantly better than MOILS-R.

To visually demonstrate the performance of MOILS-ANS and its two competitor algorithms, the projections of nondominated solutions of MOILS-ANS, MOILS-R, and MOEA/D-ANS (in red, blue, and purple, respectively) on a selected instance 20090125 at f_1–f_2 and f_1–f_3 over 20 runs are compared with the Pareto front (in green dots), as shown in Fig. 3. In the selected 2D plane (subspace), regions that are not fully covered by some algorithms are highlighted and marked with orange circles.

Comparing MOILS-ANS with MOILS-R from Fig. 3, we can easily find that some regions of the Pareto front are not fully covered or approximated by MOILS-R. As shown in Fig. 3(a), the final solution set obtained by MOILS-ANS spreads along the whole Pareto front, and is wider than those obtained by MOILS-R, as shown in Fig. 3(c). Comparing Figs. 3(b) and 3(d), we can find that MOILS-ANS covers all the regions of the Pareto front well, while MOILS-R misses some regions. Therefore, MOILS-ANS obtains better HV and IGD values than MOILS-R.

To sum up, MOILS-ANS outperforms MOILS-R. The effectiveness of the ANS strategy is revealed from comparisons between MOILS-ANS and MOILS-R.

2) Comparisons Between MOILS-ANS and MOEA/D-ANS: As summarized in Table II, MOILS-ANS significantly outperforms MOEA/D-ANS on 48 and 46 instances in terms of HV and IGD, respectively. Results of the multiproblem Wilcoxon signed-rank test show that MOILS-ANS is significantly better than MOEA/D-ANS in terms of HV and IGD.

Comparing MOILS-ANS with MOEA/D-ANS from Fig. 3, we can easily find that some regions of the Pareto front are not fully covered or approximated by MOEA/D-ANS. As shown in Fig. 3(e), although the final solution set obtained by MOEA/D-ANS spreads along the whole Pareto front, the solution set is not denser and closer to the true Pareto front as that of MOILS-ANS, as shown in Fig. 3(a). Moreover, comparisons between Figs. 3(b) and 3(f) show that MOEA/D-ANS misses some regions of the Pareto front, marked with the orange circle in Fig. 3(f).

Both MOILS-ANS and MOEA/D-ANS maintain and update a solution set, and they all can be seen as global search

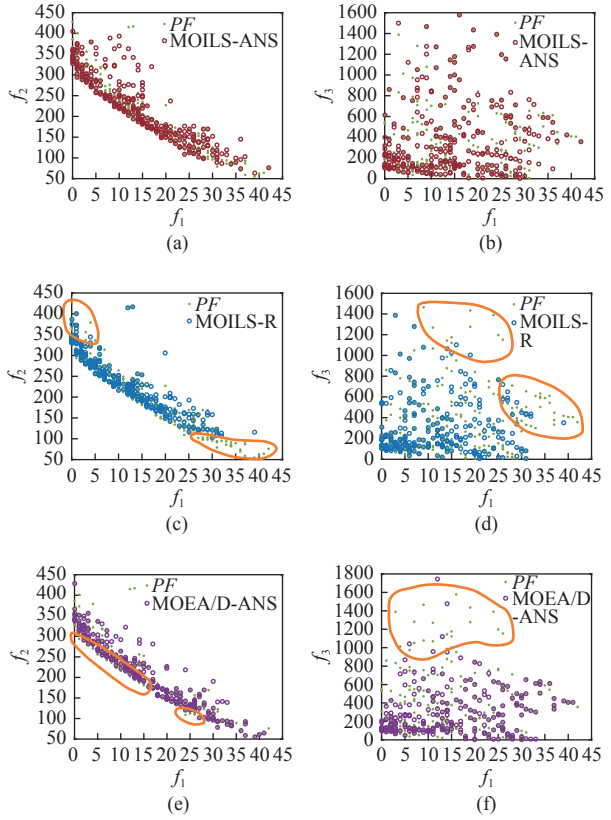


Fig. 3. Nondominated solutions obtained by all algorithms on instance 20090125 over 20 runs. (a) MOILS-ANS at f_1–f_2; (b) MOILS-ANS at f_1–f_3; (c) MOILS-R at f_1–f_2; (d) MOILS-R at f_1–f_3; (e) MOEA/D-ANS at f_1–f_2; (f) MOEA/D-ANS at f_1–f_3.

algorithms [16]. In MOILS-ANS, a solution in the current solution set is randomly selected for exploration and updating in each iteration, while, in MOEA/D-ANS, the whole population is explored and updated from generation to generation. The main characteristic of MOILS-ANS is that it extends the single-objective ILS framework [9] with perturbation scheme for escaping local minima [16]. Experimental results above show that MOILS-ANS performs better than MOEA/D-ANS. This might be due to the following reasons. In MOEA/D-ANS, it is likely that a high-quality solution will gradually fill the large portion of the population with its variants/copies because the neighborhood is defined based on uniformly distributed weight vectors. This may lead to a loss of diversity of the population in MOEA/D-ANS. In contrast, the perturbation mechanism of MOILS-ANS can help MOILS-ANS to escape local minima, which leads to better convergence of the obtained solution set.

3) Summary: Table III shows the average ranking of all algorithms by Friedman test on all instances. Table IV shows the test statistics and adjusted p-values for the Friedman test along with Holm's post-hoc procedure. In terms of both HV and IGD, MOILS-ANS gets the first rank, followed by MOEA/D-ANS and MOILS-R. In conclusion, the proposed MOILS-ANS significantly outperforms the two competitor algorithms, MOILS-R and MOEA/D-ANS.

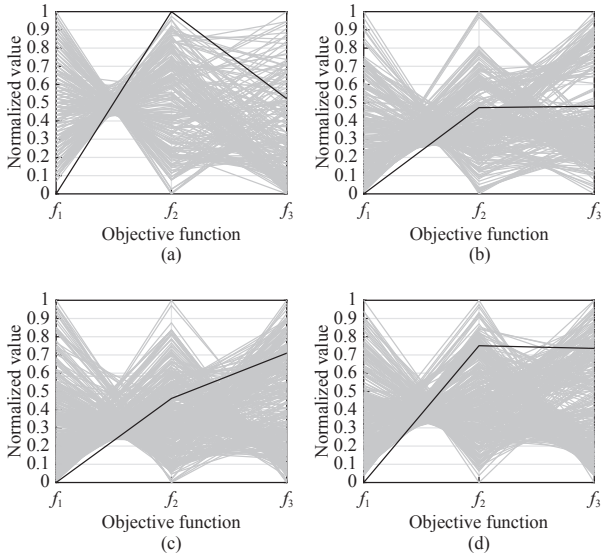


Fig. 4. Parallel coordinates visualizing the alternative solutions generated by MOILS-ANS and the best solutions reported by the previous single-objective algorithm. In each figure, our three objective functions are represented as three parallel axes. Each solution (a 3-D vector) is drawn as a polyline with vertices on the parallel axes; the position of the vertex on the ith axis corresponds to the ith objective value of the solution. The black line is the best solution reported by the previous single-objective algorithm and the gray lines are its alternatives generated by MOILS-ANS. (a) Instance 20090118; (b) Instance 20090122; (c) Instance 20090211; (d) Instance 20090218.

F. Nature of Objectives and Problems

Understanding the nature of the relationships between objectives can help develop efficient and tailored problem-solving techniques in a multiobjective optimization problem. Usually, important properties of objectives and relationships between them can be revealed from the *PF*. Therefore, this study uses the method proposed recently in [33] to visualize and analyze the nature of objectives through the *PF*. For each instance, the approximate *PF* mentioned in Section IV-C is used as *PF* since the true *PF* is unknown. Four randomly selected instances (i.e., 20090118, 20090122, 20090211, and 20090218) from all instances of January and February are chosen as representatives for analysis. We follow the four analysis steps used in [33]. The corresponding analysis results are shown in Figs. S1–S5 in the Appendix. Observations can be concluded as follows.

1) The pairwise correlation values in Fig. S1 and the scatter plots in Fig. S5 show that conflicting relationship exists between $f_1 - f_2$ for the four selected instances. Fig. S2 shows that all objectives have large ranges. These large ranges also indicate 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. To summarize, all the instances provide interesting multiobjective challenges. The improvement of one objective may lead to the deterioration of the other objective. Therefore, it is not wise to combine them into a single one using a fixed weight vector.

2) Fig. S4(a) shows that almost all regions have solutions,

representing a wide variety of options for decision makers. Besides, Fig. S4(b) shows that the frequency of instances is high in most regions, meaning that the fitness landscapes of instances are alike. That is, recurring features exist in the fitness landscapes of different instances. Using this characteristic, it may be wise to solve 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 [34].

G. Comparisons With Previous Single-Objective Algorithm

1) Comparison Results: A single-objective algorithm called ILS-VND was proposed in [9] to deal with single-objective MTPDPTW-MP. The best solutions generated by ILS-VND are collected for comparisons since the same instances are solved by both multiobjective algorithms and ILS-VND. For each instance, all nondominated solutions generated by a multiobjective algorithm over 20 runs are combined to form its nondominated solution set, which is used to compare with the best solution generated by ILS-VND.

Table S2 in Appendix summarizes GDS and GAS values of MOILS-ANS, MOILS-R, and MOEA/D-ANS with respect to best solutions of ILS-VND for each instance. Table S3 in Appendix summarizes the number of nondominated solutions generated by MOILS-ANS, MOILS-R, and MOEA/D-ANS for each instance. Key observations can be found as follows.

a) All GDS values are zero. The single-objective algorithm invests all computing resources to optimize three objectives with a fixed weight vector, and returns just one solution finally. In contrast, the proposed multiobjective algorithm must make a balance among three objectives and return a set of tradeoff solutions finally. Therefore, it is quite difficult for the proposed algorithm to dominate the solution generated by the single-objective algorithm for MO-MTPDPTWMP. In this regard, all GDS values are 0. It suggests that the proposed algorithm may still have room for further improvement in future.

b) For each instance, GAS values show that each algorithm can generate numerous alternative solutions with respect to the best solution. These alternative solutions can provide decision makers with a wide variety of options to best suit their specific requirements [13], [14].

c) The proposed MOILS-ANS obtains the best GAS values on most instances (32 instances, about 54.24% of all 59 instances). Besides, the average GAS value of MOILS-ANS is the best among all multiobjective algorithms.

To sum up, MOILS-ANS performs best, which is in line with our previous analysis based on HV and IGD.

2) Benefits of Multiobjective Optimization: To illustrate the benefits of multiobjective MTPDPTW-MP, alternative solutions generated by MOILS-ANS on four selected instances (i.e., 20090118, 20090122, 20090211, and 20090218) are visualized using parallel coordinates in Fig. 4. Our three objective functions are represented as three vertical axes in parallel coordinates. Each solution (a 3-D vector) is drawn as a polyline with vertices on the parallel axes, and the position of the vertex on the ith axis corresponds to the ith objective value of the solution. The black line is the best solution reported by the previous single-objective algorithm ILS-VND and the gray lines are its alternatives generated by MOILS-ANS. Observations can be found as follows.

a) For each instance, MOILS-ANS generates numerous alternatives with respect to the best solution. These alternatives can help decision makers to better understand the situation of the problem and provide them with more flexibility to select a solution that best matches their requirements. If all objectives are simply combined into a single one like ILS-VND, many alternatives will be lost during the search process.

b) Best solutions reported by ILS-VND always have good performances on f_1. They are obviously customer-oriented solutions. The observation is reasonable because f_1 is set as the most important objective in the single-objective algorithm ILS-VND.

c) Because of the inter-dependency among different objectives, a solution with the best performance on one objective has to compromise some other objectives. Therefore, "extreme" solutions with the best value for one objective may not be acceptable to decision makers. For example, customer-oriented solutions (black lines in Fig. 4) provided by ILS-VND are the best in terms of f_1 but poor in terms of f_2, and f_3. The interests of the hospital and the staff are ignored in these customer-oriented solutions. In such a situation, "middle ground" solutions that optimize all the objectives in the best possible way may be better choices [13]. Formulating MTPDPTW-MP as a multiobjective problem can satisfy this purpose. It is evident from Fig. 4 that we can choose a solution among its alternatives that performs a little worse on f_1, but performs much better on both f_2, and f_3 than the best solutions.

V. Conclusions

This study formulates MTPDPTW-MP as a multiobjective optimization problem to better meet the requirements of different stakeholders in real-world scenarios. To solve MO-MTPDPTWMP, a multiobjective algorithm MOILS-ANS is proposed. Problem-specific neighborhood structures and an adaptive neighborhood selection strategy are designed to better explore the search space. Experimental results show that MOILS-ANS is significantly better than the other two multiobjective algorithms. The nondominated solutions obtained by all multiobjective algorithms for MO-MTPDPTWMP are mined. The nature of objective functions and important properties of the problem are revealed. Moreover, by comparing solutions generated by MOILS-ANS with the best solutions generated by the previous single-objective algorithm, the benefits of multiobjective optimization are summarized. The mining and analysis make a step toward *explainable* multiobjective optimization.

In the future, this study can be extended from several aspects. Firstly, the proposed algorithm requires a great deal of time when dealing with large scale instances. Therefore, the proposed algorithm can be speeded up through modern computing architectures, such as computer cluster and GPU,

as in [35]. It also can be extended to solve other multiobjective routing and scheduling problems [1]. Secondly, the crossover operator is not adopted in the proposed algorithm and MOEA/D-ANS is in consistent with the existing multiobjective algorithms [2], [36]–[41]. Previous experience [38] showed that the crossover of solutions in a highly constrained problem always produces infeasible solutions. Feasibility checking is too sophisticated for MO-MTPDPTWMP. This means that a good crossover operator or repairing heuristics need to be designed, and thus the application of crossover-based algorithms to MO-MTPDPTWMP is a possible research direction. Finally, the properties of MO-MTPDPTWMP can be further studied from nondominated solutions generated by the proposed algorithms. Making full use of problem-specific knowledge can improve the search ability of multiobjective algorithms. Selection of proper hyperparameters of the algorithms using the method in [42], [43] is also our future work.

APPENDIX

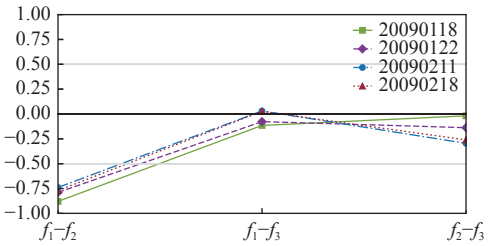


Fig. S1. Pairwise correlation values (y-axis) for each pair of objectives (x-axis) for four selected instances. The results for each instance are shown in different colors and linestyles. It shows the global pairwise relationships using the Kendall correlation method. Conflicting relationship (value < –0.5) exists between $f_1 - f_2$, which indicates that the problem instances provide interesting multiobjective challenges.

TABLE SI
COMPARATIVE RESULTS (MEAN AND STD) OF HV AND IGD ON 59 INSTANCES

Instances	HV			IGD		
	MOILS-ANS	MOILS-R	MOEA/D-ANS	MOILS-ANS	MOILS-R	MOEA/D-ANS
20090101	**0.7890 (0.0047)**	0.7850 (0.0042)–	0.7669 (0.0192)–	**0.0552 (0.0042)**	0.0613 (0.0035)–	0.0781 (0.0170)–
20090102	**0.6810 (0.0185)**	0.6417 (0.0224)–	0.6178 (0.0346)–	**0.1068 (0.0102)**	0.1512 (0.0131)–	0.1538 (0.0220)–
20090103	**0.7937 (0.0101)**	0.7842 (0.0133)–	0.7739 (0.0237)–	**0.0689 (0.0072)**	0.0768 (0.0084)–	0.0898 (0.0125)–
20090104	**0.7398 (0.0042)**	0.7366 (0.0076)=	0.7145 (0.0118)–	**0.0535 (0.0025)**	0.0585 (0.0057)–	0.0705 (0.0097)–
20090105	**0.6747 (0.0176)**	0.6025 (0.0643)–	0.6232 (0.0254)–	**0.1123 (0.0194)**	0.1930 (0.0515)–	0.1465 (0.0156)–
20090106	**0.6488 (0.0322)**	0.5754 (0.0684)–	0.6021 (0.0254)–	**0.1483 (0.0233)**	0.2087 (0.0459)–	0.1590 (0.0220)=
20090107	**0.6171 (0.0388)**	0.5794 (0.0559)–	0.5937 (0.0337)–	**0.1451 (0.0251)**	0.2018 (0.0394)–	0.1604 (0.0215)–
20090108	0.5478 (0.0231)	0.5015 (0.0530)–	**0.5734 (0.0259)+**	**0.1369 (0.0155)**	0.2032 (0.0430)–	0.1419 (0.0074)=
20090109	**0.6278 (0.0248)**	0.5848 (0.0641)–	0.6026 (0.0349)–	**0.1311 (0.0200)**	0.1774 (0.0399)–	0.1521 (0.0144)–
20090110	**0.6354 (0.0115)**	0.6139 (0.0114)–	0.6159 (0.0186)–	**0.0726 (0.0048)**	0.0890 (0.0079)–	0.0948 (0.0108)–
20090111	**0.6989 (0.0063)**	0.6919 (0.0057)–	0.6769 (0.0146)–	**0.0554 (0.0025)**	0.0602 (0.0038)–	0.0742 (0.0094)–
20090112	**0.6622 (0.0214)**	0.6260 (0.0381)–	0.6385 (0.0275)–	**0.1136 (0.0133)**	0.1630 (0.0335)–	0.1300 (0.0174)–
20090113	0.4561 (0.0895)	0.3998 (0.0890)–	**0.4962 (0.0362)=**	0.2175 (0.0720)	0.2706 (0.0594)–	**0.1583 (0.0173)+**
20090114	0.6223 (0.0337)	0.6023 (0.0638)=	**0.6413 (0.0344)=**	**0.1329 (0.0199)**	0.1617 (0.0404)–	0.1383 (0.0164)=
20090115	**0.6257 (0.0200)**	0.5948 (0.0532)–	0.6074 (0.0307)–	**0.1333 (0.0215)**	0.1752 (0.0360)–	0.1503 (0.0144)–
20090116	**0.6248 (0.0227)**	0.5664 (0.0685)–	0.5844 (0.0187)–	**0.1358 (0.0155)**	0.1956 (0.0465)–	0.1470 (0.0118)–
20090117	**0.6126 (0.0109)**	0.6044 (0.0109)=	0.5990 (0.0187)–	**0.0650 (0.0030)**	0.0767 (0.0089)–	0.0825 (0.0084)–
20090118	0.6787 (0.0091)	**0.6816 (0.0051)=**	0.6552 (0.0129)–	0.0639 (0.0102)	**0.0581 (0.0053)+**	0.0810 (0.0100)–
20090119	0.6093 (0.0182)	0.5637 (0.0519)–	**0.6291 (0.0400)+**	**0.1178 (0.0148)**	0.1715 (0.0317)–	0.1230 (0.0141)=
20090120	**0.6157 (0.0245)**	0.5779 (0.0684)=	0.6024 (0.0243)=	**0.1428 (0.0209)**	0.1822 (0.0459)–	0.1511 (0.0126)=
20090121	**0.6223 (0.0261)**	0.5561 (0.0805)–	0.5748 (0.0258)–	**0.1428 (0.0211)**	0.1986 (0.0530)–	0.1681 (0.0185)–
20090122	**0.5844 (0.0228)**	0.5349 (0.0486)–	0.5618 (0.0353)–	**0.1405 (0.0176)**	0.1961 (0.0304)–	0.1512 (0.0139)–
20090123	**0.6055 (0.0339)**	0.5494 (0.0471)–	0.5685 (0.0259)–	**0.1578 (0.0270)**	0.2184 (0.0323)–	0.1694 (0.0201)=
20090124	**0.5626 (0.0299)**	0.5502 (0.0223)=	0.5361 (0.0271)–	**0.0682 (0.0076)**	0.0821 (0.0089)–	0.0888 (0.0069)–
20090125	**0.7298 (0.0076)**	0.6998 (0.0100)–	0.7158 (0.0173)–	**0.0605 (0.0025)**	0.0732 (0.0053)–	0.0704 (0.0047)–
20090126	**0.7474 (0.0038)**	0.7335 (0.0033)–	0.7353 (0.0160)–	**0.0480 (0.0023)**	0.0505 (0.0021)–	0.0543 (0.0091)–
20090127	0.7211 (0.0036)	**0.7216 (0.0030)=**	0.7137 (0.0080)–	0.0559 (0.0022)	**0.0545 (0.0018)+**	0.0627 (0.0039)–
20090128	0.7298 (0.0088)	0.7443 (0.0078)+	**0.7461 (0.0041)+**	0.0661 (0.0084)	**0.0493 (0.0058)+**	0.0519 (0.0036)+
20090129	**0.6394 (0.0179)**	0.6195 (0.0283)–	0.6046 (0.0251)–	**0.1052 (0.0125)**	0.1336 (0.0188)–	0.1324 (0.0132)–

TABLE SI (continued)
COMPARATIVE RESULTS (MEAN AND STD) OF HV AND IGD ON 59 INSTANCES

Instances	HV			IGD		
	MOILS-ANS	MOILS-R	MOEA/D-ANS	MOILS-ANS	MOILS-R	MOEA/D-ANS
20090130	**0.6351 (0.0189)**	0.6034 (0.0439)–	0.5744 (0.0305)–	**0.1048 (0.0149)**	0.1466 (0.0268)–	0.1394 (0.0183)–
20090131	**0.7152 (0.0109)**	0.6880 (0.0212)–	0.6868 (0.0138)–	**0.0834 (0.0098)**	0.1169 (0.0159)–	0.1148 (0.0111)–
20090201	**0.7145 (0.0046)**	0.6959 (0.0125)–	0.6734 (0.0223)–	**0.0595 (0.0029)**	0.0751 (0.0080)–	0.0928 (0.0188)–
20090202	**0.6760 (0.0239)**	0.6118 (0.0647)–	0.6262 (0.0242)–	**0.1197 (0.0182)**	0.1727 (0.0412)–	0.1454 (0.0162)–
20090203	**0.6128 (0.0295)**	0.5195 (0.0787)–	0.5618 (0.0265)–	**0.1664 (0.0327)**	0.2422 (0.0428)–	0.1789 (0.0197)=
20090204	**0.6260 (0.0453)**	0.5848 (0.0820)–	0.5947 (0.0305)–	**0.1622 (0.0323)**	0.2139 (0.0504)–	0.1757 (0.0196)–
20090205	**0.5787 (0.0269)**	0.5248 (0.0787)–	0.5598 (0.0381)=	0.1554 (0.0259)	0.2181 (0.0491)–	**0.1531 (0.0190)=**
20090206	**0.6349 (0.0164)**	0.6120 (0.0508)=	0.6041 (0.0260)–	**0.1065 (0.0134)**	0.1373 (0.0333)–	0.1312 (0.0127)–
20090207	**0.7130 (0.0067)**	0.6903 (0.0129)–	0.6654 (0.0215)–	**0.0622 (0.0046)**	0.0789 (0.0075)–	0.1003 (0.0090)–
20090208	**0.6639 (0.0098)**	0.6462 (0.0114)–	0.6354 (0.0166)–	**0.0548 (0.0066)**	0.0752 (0.0099)–	0.0752 (0.0135)–
20090209	0.6300 (0.0316)	0.5790 (0.0688)–	**0.6394 (0.0397)=**	**0.1434 (0.0240)**	0.1974 (0.0446)–	0.1500 (0.0163)=
20090210	**0.6264 (0.0219)**	0.5724 (0.0624)–	0.5907 (0.0261)–	**0.1472 (0.0189)**	0.2035 (0.0386)–	0.1660 (0.0172)–
20090211	**0.6671 (0.0171)**	0.6109 (0.0539)–	0.6329 (0.0389)–	**0.1180 (0.0142)**	0.1790 (0.0303)–	0.1496 (0.0164)–
20090212	**0.6113 (0.0246)**	0.5721 (0.0530)–	0.5746 (0.0526)–	**0.1353 (0.0213)**	0.1834 (0.0332)–	0.1667 (0.0206)–
20090213	**0.6699 (0.0303)**	0.6176 (0.0686)–	0.5868 (0.0295)–	**0.1274 (0.0207)**	0.1820 (0.0408)–	0.1827 (0.0199)–
20090214	**0.7489 (0.0058)**	0.7394 (0.0090)–	0.7062 (0.0161)–	**0.0633 (0.0045)**	0.0703 (0.0080)–	0.0944 (0.0078)–
20090215	**0.6966 (0.0066)**	0.6794 (0.0055)–	0.6768 (0.0150)–	**0.0550 (0.0053)**	0.0599 (0.0054)–	0.0647 (0.0066)–
20090216	**0.6511 (0.0135)**	0.6089 (0.0293)–	0.6192 (0.0306)–	**0.1066 (0.0150)**	0.1490 (0.0195)–	0.1255 (0.0147)–
20090217	**0.5600 (0.0328)**	0.5358 (0.0770)=	0.5278 (0.0517)–	**0.1452 (0.0301)**	0.1891 (0.0475)–	0.1642 (0.0290)–
20090218	**0.6281 (0.0192)**	0.6000 (0.0252)–	0.5856 (0.0275)–	**0.1215 (0.0200)**	0.1649 (0.0227)–	0.1367 (0.0148)–
20090219	**0.6444 (0.0394)**	0.5960 (0.0772)–	0.6232 (0.0287)–	**0.1490 (0.0297)**	0.1974 (0.0457)–	0.1556 (0.0136)=
20090220	**0.5946 (0.0345)**	0.5569 (0.0423)–	0.5759 (0.0374)=	**0.1569 (0.0235)**	0.2103 (0.0408)–	0.1628 (0.0145)=
20090221	0.7512 (0.0056)	**0.7519 (0.0111)=**	0.7220 (0.0131)–	**0.0729 (0.0064)**	0.0807 (0.0141)=	0.1065 (0.0152)–
20090222	**0.6143 (0.0063)**	0.6016 (0.0100)–	0.5783 (0.0323)–	**0.0571 (0.0044)**	0.0682 (0.0060)–	0.0855 (0.0227)–
20090223	0.7076 (0.0133)	**0.7097 (0.0189)=**	0.6735 (0.0246)–	**0.0879 (0.0060)**	0.1080 (0.0133)–	0.1197 (0.0091)–
20090224	**0.5466 (0.0307)**	0.5090 (0.0528)–	0.5443 (0.0378)=	**0.1344 (0.0204)**	0.1926 (0.0455)–	0.1467 (0.0118)–
20090225	**0.6963 (0.0254)**	0.6638 (0.0565)–	0.6459 (0.0188)–	**0.0948 (0.0093)**	0.1414 (0.0394)–	0.1342 (0.0136)–
20090226	**0.5967 (0.0267)**	0.5152 (0.0685)–	0.5397 (0.0245)–	**0.1592 (0.0184)**	0.2348 (0.0554)–	0.1934 (0.0176)–
20090227	**0.6213 (0.0303)**	0.5736 (0.0695)–	0.6092 (0.0278)=	**0.1396 (0.0336)**	0.1809 (0.0422)–	0.1509 (0.0148)–
20090228	**0.7646 (0.0050)**	0.7523 (0.0105)–	0.7296 (0.0127)–	**0.0654 (0.0041)**	0.0806 (0.0094)–	0.0961 (0.0124)–
w/t/l		47/11/1	48/8/3		55/1/3	46/11/2

"+", "=" and "–" indicate that the result obtained by the corresponding competitor is significantly better than, similar to, and worse than the result obtained by MOILS-ANS, respectively. The best mean values of each instance are highlighted in bold.

TABLE SII
COMPARISON OF MULTIOBJECTIVE ALGORITHMS WITH PREVIOUS SINGLE-OBJECTIVE ALGORITHM

Instances	MOILS-ANS		MOILS-R		MOEA/D-ANS	
	GDS	GAS	GDS	GAS	GDS	GAS
20090101	0	196	0	**228**	0	200
20090102	0	**307**	0	301	0	248
20090103	0	267	0	**290**	0	242
20090104	0	192	0	162	0	**193**
20090105	0	**370**	0	278	0	278
20090106	0	**324**	0	206	0	189
20090107	0	**356**	0	251	0	306
20090108	0	**363**	0	305	0	219

TABLE SII (continued)
COMPARISON OF MULTIOBJECTIVE ALGORITHMS WITH PREVIOUS SINGLE-OBJECTIVE ALGORITHM

Instances	MOILS-ANS		MOILS-R		MOEA/D-ANS	
	GDS	GAS	GDS	GAS	GDS	GAS
20090109	0	**347**	0	298	0	250
20090110	0	209	0	**233**	0	215
20090111	0	233	0	**255**	0	233
20090112	0	313	0	**334**	0	278
20090113	0	359	0	282	0	**373**
20090114	0	248	0	**276**	0	229
20090115	0	385	0	**396**	0	284
20090116	0	**357**	0	293	0	346
20090117	0	**277**	0	270	0	242
20090118	0	**213**	0	213	0	202
20090119	0	323	0	**324**	0	307
20090120	0	**216**	0	215	0	202
20090121	0	**321**	0	290	0	235
20090122	0	**276**	0	231	0	176
20090123	0	**408**	0	332	0	276
20090124	0	**266**	0	264	0	257
20090125	0	164	0	**193**	0	116
20090126	0	173	0	**205**	0	155
20090127	0	199	0	**261**	0	184
20090128	0	121	0	**170**	0	104
20090129	0	295	0	**350**	0	251
20090130	0	**334**	0	319	0	271
20090131	0	252	0	**330**	0	169
20090201	0	199	0	170	0	**201**
20090202	0	**220**	0	198	0	120
20090203	0	**308**	0	176	0	222
20090204	0	**363**	0	279	0	274
20090205	0	**379**	0	260	0	267
20090206	0	**327**	0	323	0	257
20090207	0	307	0	**333**	0	206
20090208	0	**227**	0	217	0	219
20090209	0	**311**	0	246	0	247
20090210	0	323	0	**351**	0	267
20090211	0	**372**	0	336	0	289
20090212	0	**374**	0	364	0	272
20090213	0	**314**	0	312	0	224
20090214	0	159	0	**185**	0	142
20090215	0	125	0	**214**	0	135
20090216	0	385	0	**435**	0	305
20090217	0	**428**	0	369	0	326
20090218	0	**358**	0	312	0	266
20090219	0	**374**	0	307	0	282
20090220	0	**436**	0	312	0	288
20090221	0	247	0	**255**	0	214
20090222	0	147	0	154	0	**169**

TABLE SII (continued)
COMPARISON OF MULTIOBJECTIVE ALGORITHMS WITH PREVIOUS SINGLE-OBJECTIVE ALGORITHM

Instances	MOILS-ANS		MOILS-R		MOEA/D-ANS	
	GDS	GAS	GDS	GAS	GDS	GAS
20090223	0	295	0	**325**	0	272
20090224	0	**452**	0	378	0	356
20090225	0	334	0	**365**	0	249
20090226	0	**409**	0	352	0	272
20090227	0	**365**	0	281	0	295
20090228	0	266	0	**271**	0	235
Average	0	**296.0678**	0	278.5593	0	239.0000

The best GAS values of each instance are highlighted in bold.

TABLE SIII
NUMBER OF NONDOMINATED SOLUTIONS GENERATED BY MOILS-ANS, MOILS-R, AND MOEA/D-ANS FOR EACH INSTANCE

Instances	No. of generated nondominated solutions		
	MOILS-ANS	MOILS-R	MOEA/D-ANS
20090101	226	**243**	218
20090102	**341**	306	267
20090103	**316**	305	262
20090104	**254**	250	250
20090105	**373**	313	299
20090106	**366**	234	299
20090107	**381**	261	379
20090108	**426**	325	301
20090109	**384**	363	298
20090110	221	**238**	217
20090111	256	**288**	286
20090112	**349**	346	285
20090113	359	282	**373**
20090114	**370**	346	310
20090115	387	**396**	285
20090116	**392**	324	359
20090117	**295**	287	268
20090118	**262**	245	233
20090119	323	**324**	307
20090120	**404**	312	285
20090121	**333**	321	260
20090122	**362**	312	332
20090123	**418**	332	277
20090124	**316**	273	286
20090125	**276**	235	220
20090126	239	**247**	230
20090127	296	**313**	249
20090128	186	**240**	223
20090129	330	**350**	256
20090130	**363**	328	277

TABLE SIII (continued)
NUMBER OF NONDOMINATED SOLUTIONS GENERATED BY MOILS-ANS, MOILS-R, AND MOEA/D-ANS FOR EACH INSTANCE

Instances	No. of generated nondominated solutions		
	MOILS-ANS	MOILS-R	MOEA/D-ANS
20090131	363	**381**	241
20090201	**274**	260	238
20090202	**372**	347	308
20090203	**346**	217	294
20090204	**372**	303	290
20090205	**381**	277	273
20090206	**396**	371	300
20090207	331	**356**	228
20090208	**252**	225	239
20090209	**404**	355	310
20090210	364	**369**	277
20090211	**387**	341	295
20090212	**377**	371	277
20090213	**378**	355	270
20090214	**330**	324	254
20090215	**304**	274	267
20090216	385	**435**	306
20090217	**428**	369	328
20090218	**358**	313	267
20090219	**395**	319	299
20090220	**436**	312	290
20090221	**306**	297	270
20090222	147	154	**170**
20090223	295	**328**	272
20090224	**453**	378	356
20090225	345	**386**	265
20090226	**409**	366	283
20090227	**380**	332	337
20090228	314	**318**	300
Average	**340.4407**	311.3898	279.5763

The largest number of nondominated solutions of each instance are highlighted in bold.

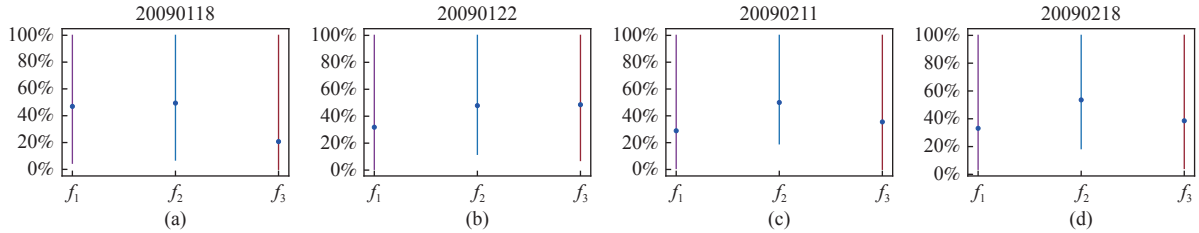


Fig. S2. Results for the objective ranges analysis for four selected instances. (a) Instance 20090118; (b) Instance 20090122; (c) Instance 20090211; (d) Instance 20090218. The *y*-axis presents the minimum, maximum, and average value of each objective as a percentage of the overall maximum value found for the respective objective. Longer lines indicate larger ranges. All objectives have large ranges (over 80%). 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. More details can be found in [33].

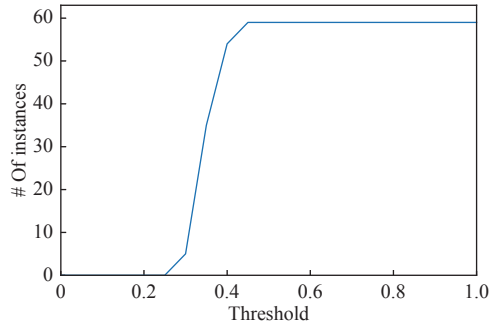


Fig. S3. Threshold analysis for all 59 instances, which shows the number of instances with solutions in r_0 when the threshold increases (normalized values). Region r_0 represents solutions with good values in all objectives. There are no solutions with good values for all objectives. More details can be found in [33].

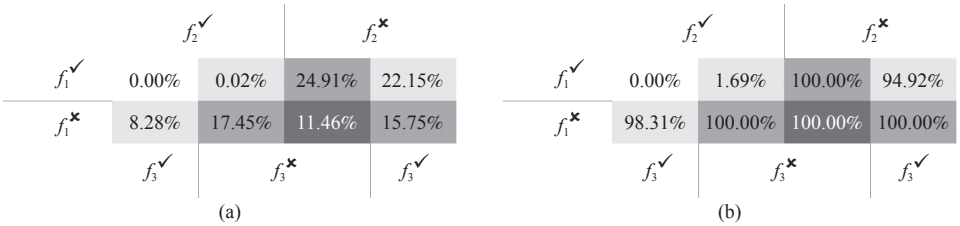


Fig. S4. Region maps for all 59 instances. (a) Overall distribution of solutions; (b) Frequency of instances. The solid lines between labels divide the map into different regions. $f_i^{\surd}$ represents good value in f_i while $f_i^{\times}$ represents bad value in f_i. The cells of the map follow the Gray code (replacing 0's and 1's by $\surd$ and $\times$). Each cell in the map represents a region r_k using a binary encoding such that the least significant digit represents objective f_1 and the most significant digit is objective f_m. For example, the solution with good values in all objectives, i.e., ($f_1^{\surd}$, $f_2^{\surd}$, $f_3^{\surd}$), falls into region r_0 in the upper left corner of the map because $0_{10} = 000_2$. More details can be found in [33]. The percentage of solutions in each region was computed for each instance. The distribution map shows the average percentage of the solutions in each region. The frequency map shows the percentage of instances that contain at least one solution in a region. The range threshold was set to the minimum value (i.e., 0.25) such that there are no solutions in r_0. There are no solutions with good values in all objectives. In the distribution map, almost all regions have solutions, representing a wide variety of options for decision makers. Besides, the frequency of instances is high in most regions, meaning that the fitness landscapes of instances are alike.

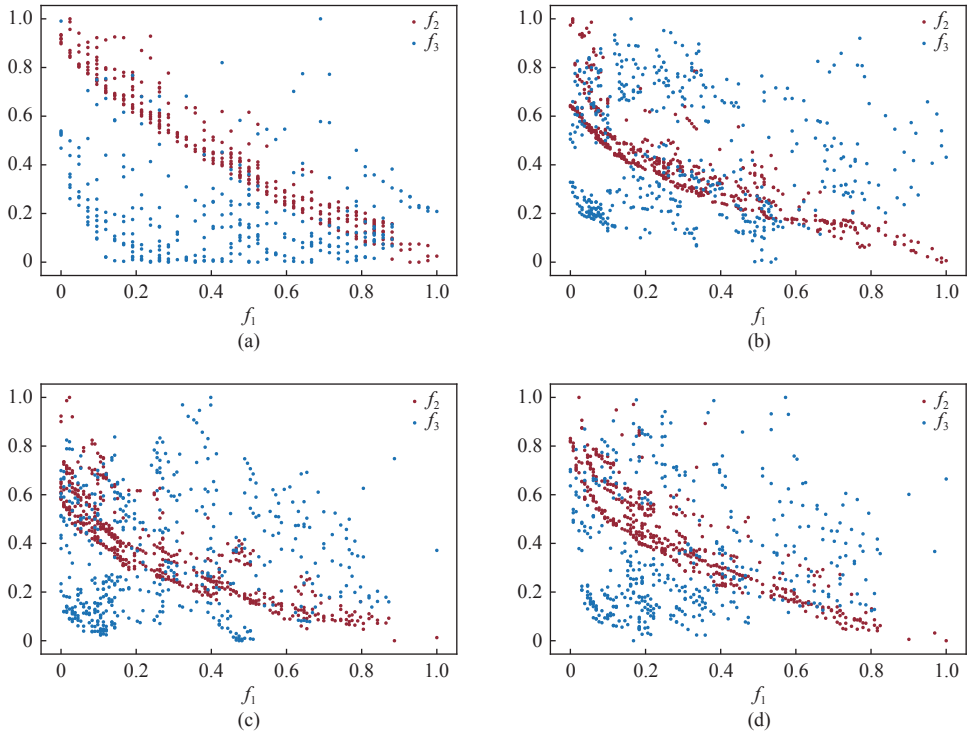


Fig. S5. Scatter plots for four selected instances. (a) Instance 20090118; (b) Instance 20090122; (c) Instance 20090211; (d) Instance 20090218. These scatter plots show the relationship between objective f_1 and each of the other two objectives. The horizontal axis is supposed to be the x-axis and the vertical axis is supposed to be the y-axis. The x-axis shows the value of objective f_1 and the y-axis shows the values of each of the other objectives in different colors. It shows that $f_1 - f_2$ has high conflicting relationships.

REFERENCES

[1] L. Wang and J. Lu, "A memetic algorithm with competition for the capacitated green vehicle routing problem," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 516–526, 2019.

[2] J. Wang, L. Yuan, Z. Zhang, S. Gao, Y. Sun, and Y. Zhou, "Multiobjective multiple neighborhood search algorithms for multiobjective fleet size and mix location-routing problem with time windows," *IEEE Trans. Systems*, *Man*, *and Cybernetics*: *Systems*, 2019.

[3] 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. Cybernetics*, vol. 49, no. 7, pp. 2467–2478, 2019.

[4] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances and algorithms," *IEEE Trans. Cybernetics*, vol. 46, no. 3, pp. 582–594, 2016.

[5] G. Kim, Y.-S. Ong, C. K. Heng, P. S. Tan, and N. A. Zhang, "City vehicle routing problem (city VRP): A review," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1654–1666, 2015.

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

[7] 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," *Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.

[8] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems," *European J. Operational Research*, vol. 234, no. 3, pp. 658–673, 2014.

[9] A. Lim, Z. Zhang, and H. Qin, "Pickup and delivery service with manpower planning in Hong Kong public hospitals," *Transportation Science*, vol. 51, no. 2, pp. 688–705, 2016.

[10] Z. Zhang, H. Qin, K. Wang, H. He, and T. Liu, "Manpower allocation and vehicle routing problem in non-emergency ambulance transfer service," *Transportation Research Part E*, vol. 106, pp. 45–59, 2017.

[11] Y. Cai, Z. Zhang, S. Guo, H. Qin, and A. Lim, "A tree-based tabu search for the manpower allocation problem with time windows and job-teaming constraints," in *Proc. 23rd Int. Joint Conf. Artificial Intelligence*, 2013, pp. 496–502.

[12] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 379–394, 2019.

[13] M. A. Nayeem, M. M. Islam, and X. Yao, "Solving transit network design problem using many-objective evolutionary approach," *IEEE Trans. Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3952–3963, 2019.

[14] A. Gupta, C. K. Heng, Y.-S. Ong, P. S. Tan, and A. N. Zhang, "A generic framework for multi-criteria decision support in eco-friendly urban logistics systems," *Expert Systems With Applications*, vol. 71, pp. 288–300, 2017.

[15] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[16] 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.

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

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

[19] M. Diana and M. M. Dessouky, "A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows," *Transportation Research Part B*: *Methodological*, vol. 38, no. 6, pp. 539–557, 2004.

[20] F. Arnold and K. Sörensen, "Knowledge-guided local search for the vehicle routing problem," *Computers & Operations Research*, vol. 105, pp. 32–46, 2019.

[21] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems," *Information Sciences*, vol. 191, pp. 146–168, 2012.

[22] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of heuristic local search for single-objective optimization in multiobjective memetic algorithms," in *Proc. Int. Conf. Parallel Problem Solving from Nature*. Springer, 2008, pp. 743–752.

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

[24] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *J. Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[25] N. Pillay and R. Qu, *Hyper-Heuristics: Theory and Applications*, Natural Computing Series. Springer, 2018. DOI: 10.1007/978-3-319-96514-7_11.

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

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

[28] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

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

[30] 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 and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[31] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, and F. Haerrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009.

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

[33] 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-Criteria Decision Analysis*, vol. 24, no. 1-2, pp. 37–56, 2017.

[34] 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. 7th Int. Conf. Operations Research and Enterprise Systems,* 2018. DOI: 10.1007/978-3-030-16035-7_8.

[35] Z. Zhang, Y. Sun, H. Xie, Y. Teng, and J. Wang, "GMMA: GPU based multiobjective memetic algorithms for vehicle routing problem with route balancing," *Applied Intelligence*, vol. 49, no. 1, pp. 63–78, 2019.

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

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

[38] E. Burke and J. Landa Silva, "The influence of the fitness evaluation method on the performance of multiobjective search algorithms,"

European J. Operational Research, vol. 169, no. 3, pp. 875–897, 2006.

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

[40] B. Derbel, A. Liefooghe, Q. Zhang, H. Aguirre, and K. Tanaka, "Multi-objective local search based on decomposition," *Lecture Notes in Computer Science*, vol. 9921 LNCS, pp. 431–441, 2016.

[41] J. Shi, Q. Zhang, and J. Sun, "PPLS/D: Parallel Pareto local search based on decomposition," *IEEE Trans. Cybernetics*, vol. 50, no. 3, pp. 1068–1071, 2020.

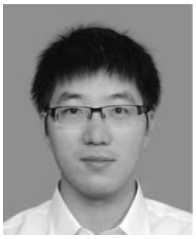
[42] J. Wang and T. Kumbasar, "Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 247–257, 2019.

[43] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neural model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–604, 2019.

Jiahai Wang (M'07–SM'19) received the Ph.D. degree in computer science from University of Toyama, Japan, in 2005. In 2005, he joined Sun Yat-sen University, where he is currently a Professor with the Department of Computer Science. His main research interests include computational intelligence and its applications.

Yuyan Sun received the B.S. degree in the School of Data and Computer Science, Sun Yat-sen University, in 2017. She is currently pursuing the M.S. degree. Her research interests include multiobjective optimization and its application to vehicle routing problems.

Zizhen Zhang received the B.S. and M.S. degree in the Department of Computer Science from Sun Yat-sen University, in 2007 and 2009, respectively. He received the Ph.D degree from City University of Hong Kong in 2014. He is currently an Associate Professor at Sun Yat-sen University. His 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 University of Toyama, Japan in 2011. He is currently an Associate Professor at the University of Toyama, Japan. His research interests include nature-inspired technologies, mobile computing, and neural networks.