

A decomposition-based chemical reaction optimization for multi-objective vehicle routing problem for simultaneous delivery and pickup with time windows

Hongye Li¹ · Lei Wang¹ · Xinghong Hei¹ · Wei Li¹ · Qiaoyong Jiang¹

Received: 5 May 2016 / Accepted: 30 December 2016 / Published online: 3 February 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract A practical variant of the vehicle routing problem (VRP), with simultaneous delivery and pickup and time windows (VRPSDPTW) is a challenging combinatorial optimization problem that has five optimization objectives in transportation and distribution logistics. Chemical reaction optimization has been used to solve mono and multi-objective problems. However, almost all attempts to solve multi-objective problems have included continuous problems less than four objectives. This paper studies discrete multi-objective VRPSDPTW using decomposition-based multi-objective optimization chemical reaction optimization. In the proposed algorithm, each decomposed sub-problem is represented by a chemical molecule. All of the molecules are divided into a few groups, with each molecule having several neighboring molecules. To balance the diversity and convergence, we designed operators of on-wall ineffective collision and inter-molecular ineffective collision for a local search, as well as operators of decomposition and synthesis to enhance global convergence. The proposed approach is compared with two different algorithms on hypervolume performance measures. Experimental results show that the proposed algorithm outperform the other algorithms in most benchmarks.

Keywords Multi-objective optimization · Chemical reaction optimization · Decomposition · Vehicle routing problem with simultaneous delivery and pickup and time windows

✉ Lei Wang
leiwang@xaut.edu.cn

Hongye Li
lihongye8@163.com

¹ School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

1 Introduction

Vehicle Routing Problem (VRP) is a classical combinatorial optimization problem, which occurs in many real-world problems, i.e., transportation, logistics, and supply chain management, et al. [1,2]. With customers having more requirements for logistics, VRP with different constraints are constantly appearing, which include time window, simultaneous pickup and delivery and so on. Up to now, variants of the VRP can be basically categorized into two classes. One is single objective VRP, and the other is multi-objective VRPs.

The single objective VRP aims to determine a set of vehicle routes originating and terminating at a central depot and visiting all clients such that the total travel distance is minimized [3]. According to the different constraints, variants of the single objective VRP has been developed different categories, that one of the most typical is VRP with time windows. Recently, governments and business organization have vigorously promoted green manufacturing and logistics. Therefore, there is an urgent need solution to energy-saving VRPs. In order to develop an energy-saving VRP, many researchers have proposed reverse logistics [4], which entail collecting end-of-life products from customers for either reuse or proper disposal. Reverse logistics have two characteristics in distribution systems: one is delivery, and the other is pickup. The VRP with pickup and delivery (PDP) can be divided into six categories [5]:

- Delivery first, pickup-second, or the vehicle routing problem with backhauls (VRPB);
- Simultaneous pickups and deliveries, or the vehicle routing problem with simultaneous pickup and delivery (VRPSPD): clients demanding both delivery and pickup operations have to be visited once, as first proposed by Min;

- Mixed deliveries and pickups, or the vehicle routing problem with mixed pickup and delivery (VRPMPD);
- Inter-related pickups and deliveries, or the dial-a-ride problem;
- The heterogeneous vehicle routing problem with simultaneous pickups and deliveries;
- The vehicle routing problem with simultaneous pickups and deliveries and time windows (VRPSPDTW): The customers sometimes request that their goods be delivered and picked up within pre-defined time windows.

Due to the constraints and problem structure of the variant of the VRP, the optimization of one objective may lead to the deterioration of other objectives; thus, many researchers have proposed multi-objective VRPs [6]. Because the decision maker's preference is not known a priori, multi-objective formulation is necessary for providing a set of solutions that represent the tradeoffs among the objectives with the VRPSDPTW, rather than a single solution [7]. The feature of multi-objective formulation is to consider all objectives with the same importance and obtain a set of Pareto optimal solutions. So far, algorithms to solve the multi-objective vehicle routing problem have mainly concerned the multi-objective vehicle routing problem with time windows and rarely more than three objectives. Many classical heuristic and meta-heuristic algorithms have recently been proposed to solve the multi-objective vehicle routing problem with time windows. These methods are mainly divided into two categories. One is based on the Pareto ranking technique, and the other is based on the decomposition method.

A Pareto ranking technique to solve this problem has been developed by many researchers. For example, genetic algorithm using Pareto ranking was developed that simultaneously minimizes the number of routes, the total travel distance, and the delivery time [6,8]. Iqbal et al. [9] proposed an artificial bee colony (ABC) algorithm combined with a two-step constrained local search for neighborhood selection in a multi-objective vehicle routing problem with soft time windows.

The other method is based on decomposition [11], which decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. Each subproblem is optimized by only using information from its several neighboring subproblems, which makes MOEA/D have lower computational complexity at each generation. It is a popular multiobjective optimization method, which is applied to solve discrete [12] and continuous multi-objective optimization problems [11]. Qi et al. [13] proposed a decomposition based memetic algorithm for the multi-objective vehicle routing problem with time windows. They proposed a novel selection operator and three local search methods in their algorithm. However, no researcher before 2016 studied the MO-VRPSDPTW with

five objectives problem. In 2016, the MO-VRPSDPTW with five objectives was proposed by Wang et al. [7]. They considered the constraints and problem structure of VRPSDPTW, which the optimization of one objective may lead to the deterioration of other objectives, thus considering the VRPSDPTW as a multi-objective optimization problem (MOP) is a necessary research direction [7]. All objectives of MO-VRPSDPTW were considered with the same importance. This could facilitate decision making without a priori knowledge.

More recently, a new metaheuristic algorithm chemical reaction optimization (CRO) was proposed by Lam and Li [14], imitating the interactions of molecules in a container by balancing local exploitation and global exploration to reach internal stability. CRO has been successfully employed to solve many mono-objective discrete optimization problems [15]. Recently, many researchers have studied continuous CRO to solve multiple objectives, such as an orthogonal multi-objective chemical reaction optimization approach for brushless DC motor design proposed by Duan and Gan [16]. Bechikh et al. [17] are the first to use the indicator-based chemical reaction method to tackle MOP. Li et al. [18] developed a decomposition-based multi-objective chemical reaction optimization to solve continuous multi-objective optimization problems. Chemical reaction optimization has not been used to solve the discrete combinatorial optimization problem of more than three objectives. Therefore, developing the character of chemical reaction optimization to solve the discrete multi-objective optimization problem is of great significance and motivates this paper.

In this paper, we developed a decomposition-based multi-objective chemical reaction optimization for the multi-objective vehicle routing problem with simultaneous delivery and pickup and time windows (MO-VRPSDPTW). In the process of algorithm design, we decompose the multi-objective optimization problems (MOPs) into a group of single objective optimization problems and use ϵ -dominance to update the archive. In this work, a chemical reaction optimization (CRO) algorithm based on decomposition is used to solve discrete MOPs (D-MOCRO). In our algorithm, the framework of decomposition is used; the approach transforms the original multi-objective problem into smaller sub-problems and solves each problem separately. Each sub-problem is optimized by combining the information of several neighboring sub-problems. Each molecule represents a sub-problem that achieves a state of energy balance by a chemical reaction (the optimal solution set). To balance diversity and convergence, we design the operators of on-wall ineffective collision and inter-molecular ineffective collision to carry on the local search, and the operators of decomposition and synthesis enhance global convergence. The proposed approach is compared with two different algorithms on hypervolume performance measures.

The rest of this paper is organized as follows. Section 2 presents the problem formulation of the MO-VRPSDPTW. Section 3 presents the relate work. Section 4 proposes the D-MOCRO for the MO-VRPSDPTW. Section 5 presents the experimental results, the parameter settings for the investigation, and the performance of the proposed algorithm. Finally, our conclusions and further work are discussed in Sect. 6.

2 Problem

2.1 MO-VRPSDPTW

The VRPSDPTW is a complex combination optimization problem that belongs to the class of NP-hard problems [2]. The VRPSDPTW plays an important role in computing theory and in many real-life applications. It entails finding a minimum-cost set of routes to deliver demand for customers with specific delivery time windows using a fleet of identical vehicles with limited capacity.

The objectives and constraints of the VRPTW can be mathematically defined using notations for the route, depot, customer, and vehicles as a theoretical graphical problem [7]. There is a non-directed complete graph $G = (V, E)$, where the vertices $V = \{0, \dots, N\}$ correspond to the depot and the customers, and the edges $e \in E\{(i, j) : i, j \in V\}$ to the links between them. The vertex 0 is called the depot, and the others are called customers. The customers are served using a fleet of vehicles with capacity C . Every customer has a demand of goods $g_i > 0$ and a time window $[b_i, e_i]$. The service constraint of the time window indicates that a vehicle should arrive at the location of customer i before the latest service time e_i . A vehicle is allowed to arrive before the earliest service time b_i , in which case it must wait until b_i and then begins to serve the customer. Each customer i has a service time s_i , which is the actual time that the delivery takes once a vehicle arrives at the customer's location. In addition to the customer, the depot also has a time window $[0, e_0]$. The demand of the depot is $g_0 = 0$. The travel distance and travel time between two locations i and j are denoted by d_{ij} and t_{ij} . The goal of VRPSDPTW is to design a set of M routes $R = \{r_1, \dots, r_M\}$ with the lowest cost. A feasible solution to the VRPSDPTW must satisfy that each vehicle departs and returns to the same depot and each customer is served by exactly one vehicle. Among R , $r_1 = \langle c(1, j), \dots, c(N_j, j) \rangle$ represents the sequence of N_j customers served in the j th route, where $c(i, j)$ the i th customers to be visited in this route indicates. $c(0, j) = c(N_j + 1, j) = 0$ represent the depot. As shown in Fig. 1, there are $M = 3$ routes $R = \{r_1, \dots, r_M\}$. $r_1 = \langle 0, c(2, 1), c(3, 1), c(7, 1), c(8, 1), 0 \rangle$ represents the sequence of four customers (customers 2, 3, 7 and 8) served in route1. This process includes three routes that have four, three and two customers served, respectively.

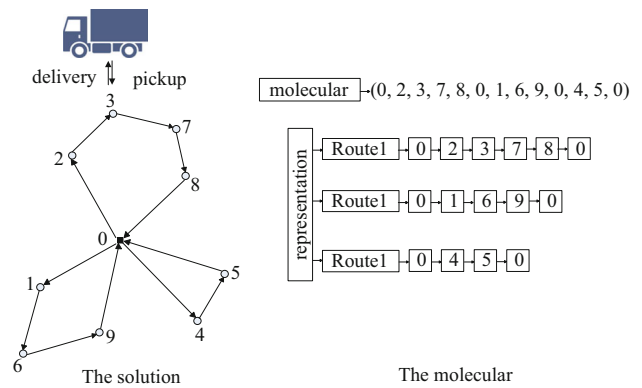


Fig. 1 Illustration of the molecular representation

With customers' demand for service quality growing higher and higher, a multi-objective vehicle routing problem must be designed to meet the expected benefits of customer demand. Before describing the multi-objective vehicle routing problem, several terms are described.

The total travel distance of the j th route is defined as

$$\text{Dist}_j = \sum_{i=0}^{N_j} d_{c(i,j)c(i+1,j)} \tag{1}$$

The arrival time of vehicle j at i th vertex is defined as follows. Among of them, the arrival time and the leaving time are represented by symbol $a_c(i, j)$ and $l_c(i, j)$ respectively.

$$a_c(i, j) = l_c(i - 1, j) + t_{c(i-1,j)c(i,j)} \tag{2}$$

where $l_c(0, j)$, which represents that vehicle j departs from the depot at time 0. If a vehicle arrived at the customer earlier than the earliest service time of customer requirements, it must wait until the earliest service time. As shown in Fig. 1 TW1, it must wait 20 min to arrive. Thus, a waiting time is caused by arriving early. The waiting time of vehicle j at i th vertex can be described as

$$w_{c(i,j)} = \begin{cases} 0, & \text{if } a_{c(i,j)} \geq b_{c(i,j)} \\ b_{c(i,j)} - a_{c(i,j)}, & \text{otherwise} \end{cases} \tag{3}$$

And the leaving time of i th customer of vehicle j is

$$l_c(i, j) = a_c(i, j) + w_{c(i,j)} + s_{c(i,j)} \tag{4}$$

Thus, the total travel time of route r_j is

$$T_j = \sum_{i=0}^{N_j} (t_{c(i,j)c(i+1,j)} + w_{c(i+1,j)} + s_{c(i+1,j)}) \tag{5}$$

where $s_{c(i+1,j)}$ denotes the service time of $(i+1)$ th customer in j th route, and the total waiting time of this route is

$$W_j = \sum_{i=1}^{N_j} w_{c(i,j)} \quad (6)$$

The delay time of vehicle j at i th vertex is

$$\text{delay}_{c(i,j)} = \begin{cases} 0, & \text{if } a_{c(i,j)} \leq e_{c(i,j)} \\ ac(i,j) - ec(i,j), & \text{otherwise} \end{cases} \quad (7)$$

and the total delay time of this route is

$$\text{Delay}_j = \sum_{i=1}^{N_j} \text{delay}_{c(i,j)} \quad (8)$$

Based on the above description, MO-VRPTW can be defined as follows.

$$\begin{cases} \min F(x) = (f_1, f_2, f_3, f_4, f_5) \\ f_1 = |R| = M \\ f_2 = \sum_{j=1}^M \text{Dist}_j \\ f_3 = \max\{T_j | j = 1 \dots M\} \\ f_4 = \sum_{j=1}^M W_j \\ f_5 = \sum_{j=1}^M \text{Delay}_j \end{cases} \quad (9)$$

f_1 and f_2 are transportation costs, where f_1 indicates the number of vehicle, the objective of f_1 is reducing the fixed cost of buying (or hiring) and repairing vehicles. f_2 is the total travel distance, which indicates the variable costs in routing. f_3 denotes the makespan, i.e., travel time of the longest route (from/to depot). The objective of f_4 improves work efficiency and avoids wasting time. The last one is f_5 , it can be considered as a service cost related to the satisfaction of customers.

The constraints of MO-VRPTW can be defined as follows.

$$\sum_{i=1}^{N_j} g_{c(i,j)} \leq C \quad \forall j = 1, \dots, M. \quad (10)$$

$$\text{delay}_{c(i,j)} \leq md \quad \forall i = 1, \dots, N_j, \quad \forall j = 1, \dots, M. \quad (11)$$

$$a_{c(N_j+1,j)} \leq e_{c(N_j+1,j)} \quad \forall j = 1, \dots, M. \quad (12)$$

Vehicle capacity constraint is represented by formula 10, which represents that the total demand of each route r_j should not exceed the vehicle capacity. The travel time constraint is represented by formula 11, which represents that the delay

time should not exceed the maximum allowed delay time. The last constraint is the return time constraint that represents vehicles should return to the depot before the closing time. MO-VRPSDPTW test benchmarks that we can visit the homepage of J. Wang.

3 Relate work

3.1 Multiobjective optimization

Multiobjective optimization is different from the single objective optimization which is to find a set of Pareto solutions rather than a single solution. In multiobjective optimization, the goal is to find the best possible trade off in different objectives, in which one objective can be improved only at the expense of worsening another. To describe the concept of optimality for problem (13) the following definitions are provided [11]. A general multiobjective optimization problem (MOP) is defined as simultaneously optimizing (minimizing or maximizing)

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{subject to : } x \in \Omega \end{cases} \quad (13)$$

where $\Omega \subset \mathbb{R}^n$ is the decision space and $x = (x_1, x_2, \dots, x_n) \in \Omega$ is a decision variable which represents a solution to the target MOP. $F(x) : \Omega \in \mathbb{R}^m$ denotes the m -dimensional objective vector of the solution x . The attainable objective set is defined as the set $\{F(x) | x \in \Omega\}$.

Definition 1 (Dominance Relation) In the minimization problem, a solution vector $u = (u_1, u_2, \dots, u_k)$ is said to dominate another vector $v = (v_1, v_2, \dots, v_k)$ (denoted by $u < v$) if u is partially less than v , i.e., $\forall i \in \{1, \dots, k, u_i \leq v_i\}$ and $\exists i \in \{1, \dots, k : u_i < v_i\}$.

Definition 2 (ε Dominated Solution) There are two solution vectors $u, v \in \mathbb{R}^k$ and $\varepsilon > 0$, if u ε dominates v , it is said to be ε dominating, i.e., $\forall i \in \{1, \dots, k, (1 - \varepsilon) \cdot u_i \leq v_i\}$.

Definition 3 (Pareto Optimum) If there does not exist any other vector $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \in \Omega$ such that $F(\hat{x}) < F(x)$, a decision variable vector $x = (x_1, x_2, \dots, x_n)$ from some universe Ω that is said to be *Pareto Optimum*.

Definition 4 (Pareto Set) The *Pareto Optimal* set \dot{P} is defined as $\dot{P} = \{x \in \Omega | x \text{ is Pareto Optimum}\}$.

Definition 5 (Pareto Optimal Set) The *Pareto Optimal front* PF is defined as $PF = \{F(x) \in \mathbb{R}^k | x \in \dot{P}\}$

With the above definitions, an MOP requires finding or approximating the set of Pareto optimal solutions and their corresponding objective vectors. The evolutionary algorithm

for solving the MOP has three issues that need to be emphasized, i.e., fitness assignment, diversity preservation and elitism strategy. A single-objective optimization problem searches only for an optimal solution; however, the MOP needs to assign the fitness of a solution to multiple criteria. Diversity preservation is an important issue for MOEAs to obtain solutions that are uniformly distributed on the Pareto front for decision makers to change. An elitism strategy is implemented in MOEAs to keep the non-dominated solutions in the population during the search.

Recently, a chemical reaction optimization was proposed to solve many mono-objective optimization problems efficiently, which inspired by chemical reactions launched during collisions, inherits several features from other meta-heuristics such as simulated annealing and particle swarm optimization. It has not solved the discrete multi-objective optimization problems so far. Then, we further develop its performance to optimize the discrete MO-VRPSDPTW. We develop a D-MOCRO to optimize the MO-VRPSDPTW, which divides the complex MO-VRPSDPTW into a number of sub-problems and solves them in a chemical reaction optimization collaborative manner. The energy conservation in the process of chemical reaction balances the local (exploitation) and the global (exploration). The chemical reaction optimization algorithm is described as follows.

3.2 Chemical reaction optimization

The chemical reaction optimization algorithm is used to solve many optimization problems involving single-objectives and multiple objectives in continuous and discrete combinatorial optimization problem.

The chemical reaction optimization algorithm includes a set of molecules put in a container with certain energy. Every molecular has a unique structure that includes kinetic energy, potential energy and the number of collision. Another important feature of this algorithm is the law of conservation. In the chemical reaction process, the PE orientates towards the minimum at the balanced state. In the reaction process, the potential energy goes towards the minimal state, similar to the objective function in optimization problems. PE is usually used as the fitness of the objective function.

The chemical reaction attributes of the manipulated molecules (M) in CRO include the following:

- w is the structure of the molecule which is used to represent a potential solution.
- PE (potential energy) is defined as the objective function value of the corresponding solution that is represented by w .
- KE (kinetic energy) is a non-negative number quantifying the tolerance of the system to accept a worse solution than the existing one.

- $NumHit$ is the number of hits which is a record of the total number of collisions that a molecule has taken.
- $MinPE$ is the minimum potential energy which represents the minimum value of the function of the current state.
- $MinHit$ is the minimum hit number which represents the minimum number of hits when the molecule attains the value of the function of the current state.
- $Buffer$ is the total energy of the container.

The basic chemical reaction algorithm has four types of reaction operators, including on-wall ineffective collision, decomposition, inter-molecular ineffective collision and synthesis. These four operators are the same as the paper [14].

(1) On-wall ineffective collision operator of CRO [14]

An on-wall ineffective collision reaction occurs when a molecule hits the wall of the container and then bounces back. After the on-wall collision, the structure of the molecules will change. If the current molecular structure is w , it will turn into another state w' . The reaction process is defined by Eq. (2).

$$w(i) \rightarrow w(i)' \quad (14)$$

where, $w(i)' = Neighbor(w(i))$.

As a rule of thumb, when a molecule hits a wall, a portion of its KE will be lost, the lost energy is stored in the central energy buffer, when the reaction has completed. Its KE is updated as follows:

$$KE_{w'} = (PE_w - PE_{w'} + KE_{w'}) \times \alpha \quad (15)$$

where α is a random number that lies in between $[KELossRate, 1]$, where $KELossRate$ is a parameter of CRO.

If the KE processed by the molecule is high, then there is a possibility that the PE could increase, depicting a worse solution. This change is desirable to make the algorithm escape from its local minimum. The KE drops as an effect of collision. Its level of tolerance of obtaining a worse solution is lowered.

(2) Decomposition operator of CRO [14]

A decomposition operator means that a molecule w hits the wall of the container and then breaks into two or more molecules. Compared with the ineffective collision, the decomposition is more vigorous and the resultant molecule structures have greater differences from that of the original one. This operator can be considered as the situation when we finish the local search from w to w'_1 and w'_2 . Due to the conservation of energy, w may sometimes not have enough

energy to sustain its transformation into w'_1 and w'_2 . A certain portion of energy in the buffer accumulated from on-wall ineffective collisions can be utilized to support the change. The specific implementation process of the decomposition operator is described in Sect. 4.2.

(3) Inter-molecular ineffective collision operator of CRO [14]

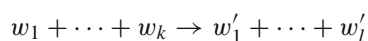
An intermolecular ineffective collision occurs when two or more molecules collide with each other and then separate. The number of molecules involved in this collision remains unchanged after the collision. If more molecules are involved in the reaction, more energy is needed, and the structure of the molecule is also more flexible. In the original implementation in the simulation, we assume only two molecules, e.g., with molecular structures w_1 and w_2 , involved. Similar to the on-wall ineffective collision, this collision is also not vigorous and the new molecular structures w'_1 and w'_2 are produced from their own neighborhoods separately.

(4) Synthesis operator of CRO [14]

Synthesis does the opposite of decomposition. A synthesis refers to when two or more molecules collide and then combine to form one new single molecule. This process implies that the search regions are expanded, i.e., the diversification of solutions. The operator of synthesis uses crossover of a genetic algorithm. The specific implementation process of the synthesis operator is described in Sect. 4.2.

3.2.1 Energy handling [14]

Energy can be transformed from one type to another but all energy manipulations must follow the first law of thermodynamics, which states the energy can neither be created nor destroyed. A generalized form of the elementary reaction can be written as follows:



where k and l are the numbers of molecules involved before and after the reaction, respectively. For $k=1$ and $l=2$, the reaction can be said to be a decomposition reaction.

The corresponding energy equation can be written as follows:

$$\begin{aligned} & (PE_{w_1} + \dots + PE_{w_k}) + (KE_{w_1} + \dots + KE_{w_k}) \\ & \quad + \text{buffer} \\ & = (PE_{w'_1} + \dots + PE_{w'_l}) + (KE_{w'_1} + \dots + KE_{w'_l}) \\ & \quad + \text{buffer}' \end{aligned}$$

In the equation above, the change in the total energy of the molecules before and after the reaction is represented by the left and the right hand sides of the equality respectively.

The general acceptance rule for the new solution is given as follows:

$$\sum_{i=1}^k PE_{w_i} + \sum_{i=1}^k KE_{w_i} - \sum_{i=1}^l PE_{w_i} \geq 0$$

3.2.2 The sketch of CRO algorithm

As explained above, the step-wise procedure for the implementation of CRO can be summarized as follows.

Algorithm 1: CRO()

Begin

Step 1: Define the optimization problem and initialize the optimization parameters.

Step 2: Initialize the population of molecules.

Step 3: **while** ($FES < FESLimit$)

If ($rand > MoleColl$) || (the number of population is equal to one)

 go to **Step 4**;

else

 go to **Step 5**;

endif

endwhile

Step 4: Uni-molecular operator.

If $w_i \text{ numHit} - w_i \text{ minHit} > DecThres$

 Decomposition operator of CRO ();

else

 On-wall Ineffective Collision operator of CRO ();

endif

Update energy management rules ();

Step 5: Inter-molecular operator.

If ($w_i KE < SynThres$) && ($w_j KE < SynThres$)

 Synthesis operator of CRO ();

else

 Inter-molecular Ineffective Collision operator of CRO ();

endif

Step 6: Termination criterion. Stop if the maximum generation number is achieved; otherwise repeat from **Step 3**.

end

CRO has good searching ability and shows excellent operation in two important features of optimization metaheuristics, i.e., controlling the convergence and diversity. It can control the convergence by benefiting from the crossover of the GA. The diversity of chemical reaction uses the operators of on-wall ineffective collision. The conservation of energy of this algorithm can balance the diversity and convergence of solutions.

4 Proposed algorithms for MO-VRPSDPTW

4.1 Solution representation for MO-VRPSDPTW

In the proposed algorithm, the solution representation for the MO-VRPSDPTW is based on variable-length solution representation [7]. The solution of the proposed algorithm is denoted by the chemical molecular, which is shown in Fig. 1 (The molecular). The solution includes several routes, and each route has a sequence of customers to be served. It is worth noting that each route was designed with a vertex 0 as the beginning and ending of the route.

4.2 D-MOCRO for MO-VRPSDPTW

The decomposition method chosen to solve the MO-VRPSDPTW is defined by (9). MOEA/D has achieved good results in solving multi-objective optimization problems. In our algorithm, we use the MOEA/D framework. In [19], Ishibuchi et al. successfully used the decomposition method to solve the combinatorial optimization problem. He analyzed the performance of various methods of decomposition, such as weighted sum, weighted Tchebycheff and penalty-based boundary intersection function. It showed that the weighed sum has high diversification ability. Therefore, the weighted sum method is used to solve the MO-VRPSDPTW, in this paper. The decomposition method is does not directly solving the multiple objectives; rather, it uses decomposition to create a series of quantitative sub-problems, with a parallel chemical reaction optimization algorithm to solve these sub-problems. In this method, each sub-problem is a neighborhood. Each sub problem is optimized by the information of its neighborhood sub problem. In each generation of reaction, the group is made up of the best solution found by each sub-problem. D-MOCRO uses the framework of MOEA/D. It decomposes the target MO-VRPSDPTW into a number of scalar sub-problems using the weighted sum approach.

D-MOCRO uses a specially designed local search methods to improve the quality of the solution to each scalar sub-problems periodically. It works as follows:

(1) Initialization Procedure

In the initialization of the MO-VRPSDPTW, a solution is randomly generated as follows. First, a customer should be selected and a route should be created. Another customer is randomly selected to insert after the first customer in the route. Other customers are inserted until another cannot to be inserted, at which point a new route should be created. This procedure is repeated until all customers are inserted, and then one molecular with MO-VRPSDPTW characteristics is built. The five objectives of the above molecular are

computed according to Formula (9). This is followed by the initialization weight vector, the calculation of the Euclidean distance and the setting of the neighborhood. In addition, the ε -dominance archive is adopted in the proposed algorithm.

D-MOCRO decomposes the MO-VRPSDPTW into a scalar number of single objective sub-problems. The procedure of D-MOCRO, whose pseudo-code is given in Algorithm 6, includes three main points: the initialization of population (molecules) P , the generation of weight vectors and the assignment of neighborhood. To be specific, the initial population P is randomly sampled from feasible space via a uniform distribution. A set of uniformly distributed weight vectors $W = \{w^1, \dots, w^N\}$ are generated using Das and Dennis' method [20] where weight vectors are sampled from a unit simplex. Given H and M , a total number of $N = \binom{H+M-1}{M-1}$ uniformly distributed points, with a uniform spacing $\delta = 1/H$, where $H > 0$ is the number of divisions considered along each objective coordinate, which sampled on the simplex for any number of objectives, and mapping the points on the search space. An example is shown in Fig. 2 to illustrate Das and Dennis' weight vector generation method.

The MO-VRPSDPTW can be simulated based on a vector that is represented by the molecular of CRO. The neighborhood search operator is applied in this algorithm, which is based on the vector used in the evolutionary algorithm. An old solution can be found by neighbor search. Each sub-problem (molecular) has its own population and sees computational effort at each generation. D-MOCRO follows the framework of MOEA/D. It decomposes the MO-VRPSDPTW defined by formula (9) into Q single objective optimization sub-problems. Each sub-problem has a niche, which maintains could keep diversity management in the objective space.

In the classical multi-objective decomposition method, the most popular approached are weighted sum, weighted Tchebycheff and boundary intersection. In this paper, we use the weighted sum approach [11]. The weighted sum method is described for m -objective problems using a weight vector $\Lambda^i = (\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im})$

$$\min g^{ws}(x^i | \Lambda^i) = \lambda_{i1} \cdot f_1(x^i) + \lambda_{i2} f_2(x^i) + \dots + \lambda_{im} f_m(x^i) \quad (16)$$

where m equals five in this paper $\Lambda^i = (\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im})$ corresponds to a sub-problem i ($i = 1, 2, \dots, Q$), $\lambda_1 + \lambda_2 + \dots + \lambda_m = 1$, and x^i is a solution to be optimized.

Many practical optimization problems have a Pareto front with an irregular shape. To approximate the Pareto optimal solutions of a multi-objective optimization problem, Zhang and Li [11] recently developed a novel multi-objective evolutionary algorithm based on decomposition (MOEA/D). It can work well if the curve shape of the Pareto-optimal front

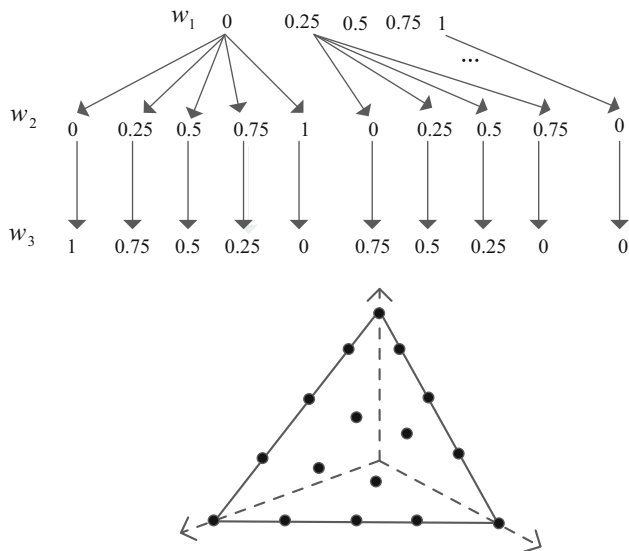


Fig. 2 Structured weight vector $w = (w_1, w_2, w_3)^T$ generation process with $\delta = 0.25$, i.e., $H = 4$ in 3-D space. $\binom{4+3-1}{3-1} = 15$ weight vectors are sampled from a unit simplex

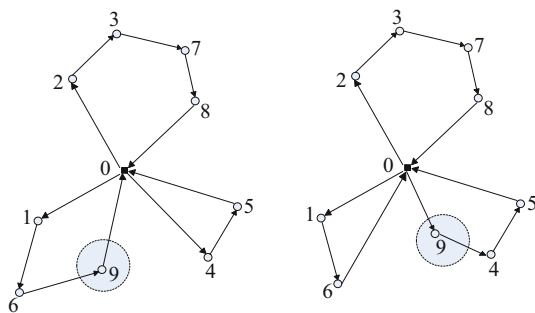


Fig. 3 On-wall ineffective collision operator of CRO

is friendly. However, it shows poor performance to the irregular shape of the Pareto-optimal front. For this problem, Liu et al. [21] proposed an improved MOEA/D algorithm (denoted as TMOEA/D) that utilizes a monotonic increasing function to transform each individual objective function into one so that the curve shape of the non-dominant solutions of the transformed multi-objective problem is close to the hyper-plane whose intercept of coordinate axes is equal to one in the original objective function space. We consider the MO-VRPSDPTW as a discrete optimization problem, so T-MOEA/D [21] was applied in this work.

(2) Reproduction Procedure

The on-wall ineffective collision operator randomly selects a customer from a selected route and then re-inserts it into a good position. This operator is shown in Fig. 3, which the pseudo code described in Algorithm 2. This process includes two points, one is selecting a route, and the other is defining

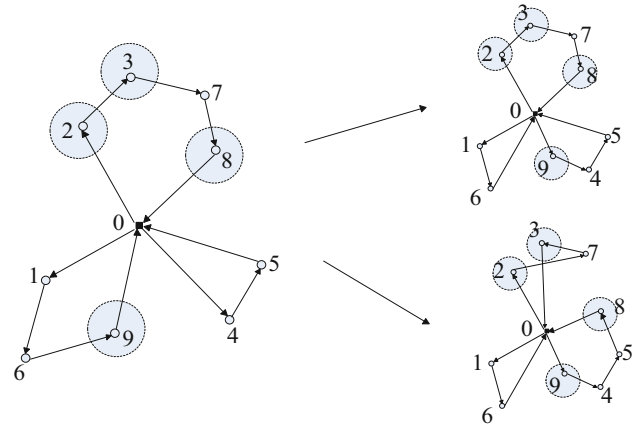


Fig. 4 Decomposition operator of CRO

the good position to insert the customer. Selecting a route needs to rely on the longest travel time for f_3 from formula 9. A good position includes different objectives: the position that has the lowest total distance, the lowest travel time among all routes, the lowest total waiting time and the lowest total delay time from f_2, f_3, f_4 and f_5 separately. In short, the best position is the position that achieves the lowest cost.

Algorithm 2: On-wall ineffective collision operator of CRO ()

```

1 Begin
2 Input: a solution  $w$ 
3 Duplicate  $w$  to produce  $w'$ 
4 Select a route according to the above description
5 Randomly select a number customer from the selected route and remove from this route
6 The removed customer reinsert into the good position
7 Repair( $w'$ )
8 Output:  $w'$ 
9 end
    
```

The decomposition operator of CRO has a high KE, which is active and similar to a global search. The specific operation is that it selects two routes and then randomly selects a number of customers from the selected routes. This operator is shown in Fig. 4, with the pseudo code described in Algorithm 3.

Algorithm 3: Decomposition operator of CRO ()

```

1 Begin
2 Input: a solution  $w$ 
3 Duplicate  $w$  to produce  $w_1'$  and  $w_2'$ 
4 Select two routes from  $w$ 
5 Select a number of customers from the selected routes randomly
6 Reinsert those customer into good points and delete customers of selected routes
7 Repair( $w_1'$ ); Repair( $w_2'$ );
8 Output:  $w_1'$  and  $w_2'$ 
9 end
    
```

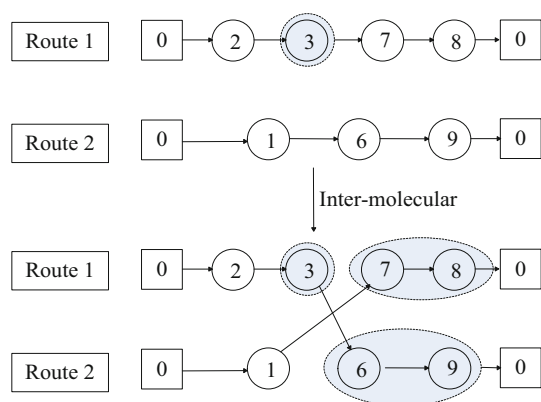



Fig. 5 Inter-molecular ineffective collision operator of CRO

The inter-molecular ineffective collision operator of CRO will lead to changing the structure of the molecule. The inter-molecular ineffective collision operator of D-MOCRO exchanges a sequence of customers between routes while preserving the orientation of the sequences or routes. This operator is shown in Fig. 5, with the pseudo code described in Algorithm 4.

Algorithm 4: Inter-molecular ineffective collision operator of CRO ()

```

1 Begin
2   Input: solution  $w_1$  and  $w_2$ 
3     Select two different routes from  $w_1$ , select sequence customers
       from one route and
4     exchange selection customers between two routes
5     Select two different routes from  $w_2$ , select sequence customers
       from one route and
6     exchange selection customers between two routes
7     Repair( $w_1'$ ); Repair( $w_2'$ );
8   Output:  $w_1'$  and  $w_2'$ 
9 end

```

The synthesis operator of D-MOCRO generates new solutions described as follows. First, a random number of routes are selected from the first molecule, and copied into the offspring. Then, all the routes from the second molecule, that are not in conflict with customers already copied from the first parent are copied into the offspring.

Algorithm 5: Synthesis operator of CRO ()

```

1 Begin
2   Input: solution  $w_1$  and  $w_2$ 
3     Randomly select two different customers  $i$  and  $j$  from the set of  $\{1, \dots, n\}$ 
4     Synthesis ( $w_1'(i)$  and  $w_2'(j)$ );
5     Repair( $w_1'$ ); Repair( $w_2'$ );
6   Output:  $w_1'$  and  $w_2'$ 
7 end

```

Thus, the crossover operator makes offspring inherit routes from parents. Once inherited routes are chosen, they can be regarded as seed routes. The existing routes should include all customers. If one customer cannot be inserted properly, the customer will establish a new route. This oper-

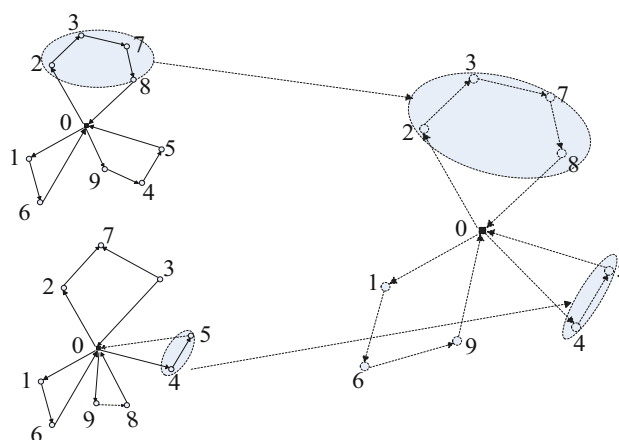


Fig. 6 Synthesis operator of CRO

ator shown in Fig. 6, with the pseudo code described in Algorithm 5.

For the optimization algorithm, a good balance between exploration and exploitation is necessary. Exploration refers to ability of the algorithm to “explore” or search different regions of feasible search space, whereas “exploitation” refers to the ability or all the individuals to converge to the near optimal solutions as quickly as possible. Too much stress on exploration will result in a purely random search, whereas too much stress on exploitation will result in a purely local search. In this paper, the chemical reaction optimization operator of inter-molecular ineffective collision and synthesis serve as vehicles of exploration, where the chemical reaction optimization operators of decomposition and on-wall ineffective collision serve as vehicles of exploitation (Fig. 7).

In this section, D-MOCRO uses the framework of MOEA/D to optimize the MO-VRPSDPTW. To maintain the diversity of solutions, D-MOCRO uses an external population to store non-dominated solutions, with multiple single objective sub-problems optimized in parallel. For each sub-problem, a chemical reaction optimization search is applied to improve the current solution. To maintain diversity in searching different parts of the POF, D-MOCRO applies an adaptive weight [10]. The main features of D-MOCRO include (1) using chemical reaction optimization to optimize the MO-VRPSDPTW, (2) using the adaptive weight vectors, (3) using ε -dominance to update the feasible solution and (4) using elite population to contain nondominant Pareto feasible solution sets by using an efficient non-dominated sort on sequential search strategy [22]. The procedure of D-MOCRO is given in Algorithm 6.

5 Experimental results and discussions

In this section, 45 real-world benchmark instances are used to test the performance of the proposed D-MOCRO. The

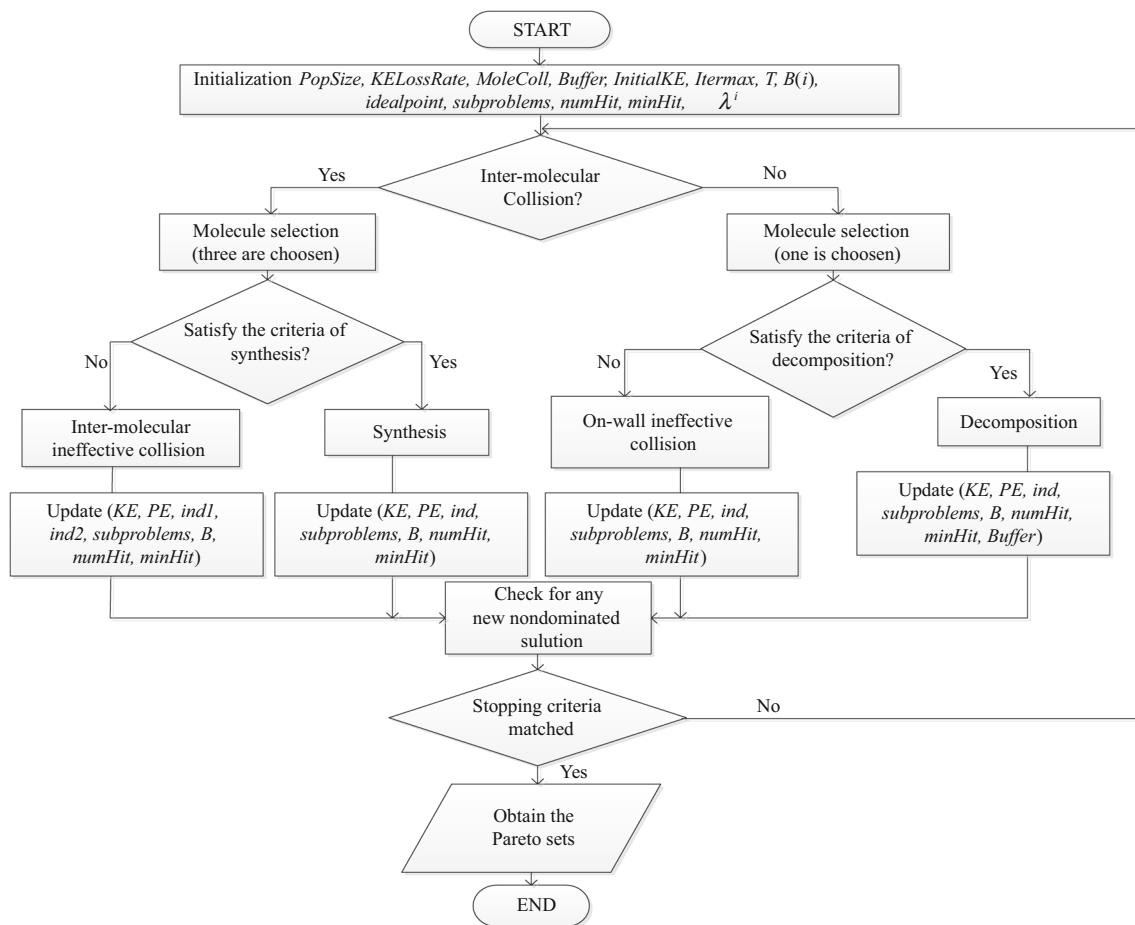


Fig. 7 Flow chart showing the working of D-MOCRO algorithm

MOMA [7] and MOLS [7] algorithms are compared with our proposed algorithm.

5.1 Benchmark functions

See Table 1.

5.2 Performance metrics

The performance of a MOEA is evaluated from two aspects. First, the obtained non-dominated set should be as close to the true Pareto front as possible to achieve good convergence. Second, the solutions in the obtained non-dominated set should be distributed as diversely and uniformly as possible. In this paper, the Hypervolume metric is used.

Hypervolume (I_H) is used to indicate the area in the objective space that is dominated by at least one solution of the nondominated set. This metric was suggested by Zitzler et

al. [23] to indicate the area in the objective space that is dominated by at least one solution of the nondominated set. In practice, I_H is calculated as follows:

$$I_H(A) = \int_{z \in \cup_{x \in A} HV(f(x), f^*)} \dots \int 1 \cdot dz \quad (17)$$

where $HV(f(x), f^*) = [f_1(x), f_1^*] \times \dots \times [f_m(x), f_m^*]$ is the Cartesian product of the closed intervals $[f_i(x), f_i^*]$, $i = 1, \dots, m$, the $f^* = (f_1^*, \dots, f_m^*)$ is reference point and the set A is nondominated set. An example with two objectives is shown in Fig. 8, where the solution a , b , c and d are objective vectors. The area of shadow represents the hypervolume of nondominated set. From Fig. 8 shows that the large I_H represents the nondominated close towards the Pareto front. Therefore, the I_H metric can also measure the convergence and diversity of an algorithm.

Algorithm 6: D-MOCRO

Input: An instance of MO-VRPSDPTW, the number of subpopulation N , a set of uniformly distributed direction vectors $\lambda_1, \dots, \lambda_N$, the maximal number of generations FES_{max} ,

Output: A set of nondominated solutions X^* .

Begin

Step 1 Initialization

Step 2: Initialize the population of molecules.

Create the initial population P_0 with N random molecules, set the external elitism archive $X^* = \emptyset$;

Using the Direction Vectors Generation Mechanism to generate N uniformly distributed direction vectors

$\lambda^1, \lambda^2, \dots, \lambda^N$. Computing the Euclidean distances between any two weight vectors to each weight

vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$ where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the T closest weight vectors to λ^i .

Decomposition the original MO-VRPSDPTW into N SO-VRPSDPTW g_1, \dots, g_N with $\lambda^1, \lambda^2, \dots, \lambda^N$.

Step 3: while ($FES \leq MaxFunEvals$) do

Step 4: Reproduction:

Decompose the original MO-VRPTW into N scalar subproblems with weight set $\lambda^1, \lambda^2, \dots, \lambda^N$;

Selection of Reaction/Update Range: Uniformly randomly generate a number $rand$ from $[0, 1]$. Then set

$$P = \begin{cases} B(i) & \text{if } rand < \delta, \\ Archive(i) & \text{otherwise.} \end{cases}$$

Selection of the number of molecular: Uniformly randomly generate a number of $rand$ from $[0, 1]$.

If ($(rand > collRate) \parallel (pop.size () = 1)$)

If ($(subproblems(i).curHitIndex - subproblems(i).minHitIndex) > threshold$)

Decomposition ($subproblems(i)$);

$FES = FES + 2$;

else

On-wall ($subproblems(i)$);

$FES = FES + 1$;

endif

else

Selection subproblems (j) from the neighborhood of the subproblems (i).

If ($subproblems(i).KE < SynThres \ \&\& \ subproblems(j).KE < SynThres$)

Synthesis($subproblems(i), subproblems(j)$);

$FES = FES + 2$;

else

InterMolecular($subproblems(i), subproblems(j)$);

$FES = FES + 2$;

endif

endif

Repair: the neighborhoods of the subproblems(i) and the subproblems(j).

Update of z : For each $j, i = 1, \dots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$,

If $z_i > f_i(y)$, then set $z_i = f_i(y)$.

Update of Solutions: Set $count = 0$ and then do the following:

If $count = PopNum$ or the number of subproblems is empty

go to **Step 3**.

else randomly pick an index j from the set of subproblems.

$Z = X \cup Y$, Sort the molecules in Z by Efficient non-dominated sort on sequential search strategy[23]

endif

endwhile

end

Table 1 Details of 45 real-world instances and stopping time (in seconds) for each instance

Instance	Size of customer	Vehicle capacity	Time window profile	Running time
50-0-0	50	690	<i>Profile 1</i>	117
50-0-1	50	690	<i>Profile 2</i>	245
50-0-2	50	690	<i>Profile 3</i>	282
50-0-3	50	690	<i>Profile 4</i>	253
50-0-4	50	690	<i>Profile 5</i>	142
50-1-0	50	250	<i>Profile 1</i>	114
50-1-1	50	250	<i>Profile 2</i>	222
50-1-2	50	250	<i>Profile 3</i>	270
50-1-3	50	250	<i>Profile 4</i>	221
50-1-4	50	250	<i>Profile 5</i>	139
50-2-0	50	85	<i>Profile 1</i>	130
50-2-1	50	85	<i>Profile 2</i>	199
50-2-2	50	85	<i>Profile 3</i>	200
50-2-3	50	85	<i>Profile 4</i>	184
50-2-4	50	85	<i>Profile 5</i>	146
150-0-0	150	1854	<i>Profile 1</i>	183
150-0-1	150	1854	<i>Profile 2</i>	267
150-0-2	150	1854	<i>Profile 3</i>	301
150-0-3	150	1854	<i>Profile 4</i>	309
150-0-4	150	1854	<i>Profile 5</i>	239
150-1-0	150	638	<i>Profile 1</i>	163
150-1-1	150	638	<i>Profile 2</i>	245
150-1-2	150	638	<i>Profile 3</i>	300
150-1-3	150	638	<i>Profile 4</i>	308
150-1-4	150	638	<i>Profile 5</i>	239
150-2-0	150	182	<i>Profile 1</i>	163
150-2-1	150	182	<i>Profile 2</i>	258
150-2-2	150	182	<i>Profile 3</i>	295
150-2-3	150	182	<i>Profile 4</i>	341
150-2-4	150	182	<i>Profile 5</i>	220
250-0-0	250	3087	<i>Profile 1</i>	258
250-0-1	250	3087	<i>Profile 2</i>	345
250-0-2	250	3087	<i>Profile 3</i>	354
250-0-3	250	3087	<i>Profile 4</i>	407
250-0-4	250	3087	<i>Profile 5</i>	292
250-1-0	250	1046	<i>Profile 1</i>	225
250-1-1	250	1046	<i>Profile 2</i>	303
250-1-2	250	1046	<i>Profile 3</i>	338
250-1-3	250	1046	<i>Profile 4</i>	400
250-1-4	250	1046	<i>Profile 5</i>	261
250-2-0	250	284	<i>Profile 1</i>	237
250-2-1	250	284	<i>Profile 2</i>	292
250-2-2	250	284	<i>Profile 3</i>	329
250-2-3	250	284	<i>Profile 4</i>	395
250-2-4	250	284	<i>Profile 5</i>	282

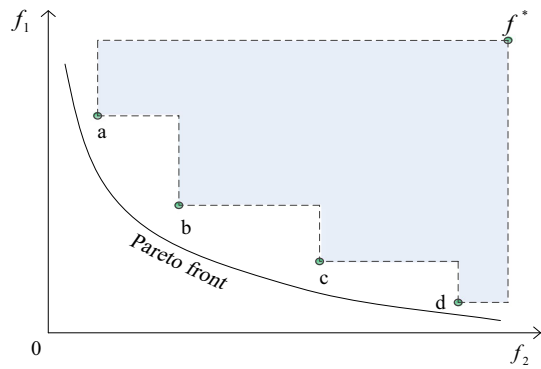


Fig. 8 Hypervolume of nondominated solutions

5.3 Parameter settings

In our experimental study, the $MAX_FES = 50,000$, the number of the subproblems $N = 495$, the neighborhood list size $T = 5$. Parameters of D-MOCRO are set as follows. The $initKE$ equals to 10,000, the $initBuffer$ equals to 100, the $decThres$ equals to 800, the $synThres$ equals to 15, the $lossRate$ equals to 0.1, the $collRate$ equals to 0.2. The $initKE$ controls the molecular kinetic energy. The $initBuffer$ controls the container total energy which plays a keeping energy conservation role. The $decThres$ is a threshold value of carrying on decomposition operator. The $synThres$ is a threshold value of carrying on synthetic operator. The $collRate$ is a threshold value of carrying on whether the molecules collide with the container wall or collide with each other. In this paper, CRO with $collRate$ is also used in balancing exploration and exploitation processes of the proposed CRO algorithm. Population size is set to $N = 500$ for all the benchmarks. All the compared algorithms stop when the number of function evaluation reaches the maximum number. The maximum function evaluation number is set to 30,000 for the benchmarks. The MOLS [7] and MOMA follow the implementation of J. Wang [7].

For all experiments, 20 independent runs are carried out on the same machine with a Celoron 3.40 GHz CPU, 4 GB memory, and windows 7 operating system, and conducted with the maximum number of function evaluations (MAX_FES) as the termination criterion. The proposed D-MOCRO and comparison algorithms were implemented in Microsoft Visual Studio 2013 (C++).

5.4 Experimental results and discussions

5.4.1 Comparison of the experimental results on real world instances

Recently, a many-objective VRPSDPTW was proposed. Most previous algorithms consider only the VRPSDPTW/

Table 2 The HV metric of comparative algorithms

Instance	I_H		
	D-MOCRO	MOLS	MOMA
50-0-0	0.8570	0.8690 §	0.5540†
50-0-1	0.7190	0.7500 §	0.4870†
50-0-2	0.6110	0.6820 §	0.5660†
50-0-3	0.5570	0.5340†	0.5240†
50-0-4	0.7020	0.6460†	0.6630†
50-1-0	0.8450	0.8540 §	0.6270†
50-1-1	0.7290	0.7150†	0.5460†
50-1-2	0.6950	0.6720†	0.5810†
50-1-3	0.5780	0.5560†	0.5290†
50-1-4	0.6900	0.6880†	0.6460†
50-2-0	0.7120	0.7020†	0.6600†
50-2-1	0.6070	0.5890†	0.5420†
50-2-2	0.5740	0.6080 §	0.5440†
50-2-3	0.5600	0.5800≈	0.5990 §
50-2-4	0.5850	0.6080 §	0.5330†
150-0-0	0.8960	0.8770†	0.6150†
150-0-1	0.7470	0.7570 §	0.5820†
150-0-2	0.7170	0.7450 §	0.5410†
150-0-3	0.6903	0.6150 §	0.5010†
150-0-4	0.7060	0.7410 §	0.5540†
150-1-0	0.8570	0.8830 §	0.6270†
150-1-1	0.7670	0.7530†	0.5660†
150-1-2	0.7170	0.7370 §	0.5380†
150-1-3	0.6220	0.5930†	0.4940†
150-1-4	0.7140	0.7140 §	0.5580†
150-2-0	0.8450	0.8290†	0.5990†
150-2-1	0.7390	0.7380†	0.5540†
150-2-2	0.6640	0.7050 §	0.4870†
150-2-3	0.6060	0.5890†	0.5060†
150-2-4	0.7300	0.7160†	0.5400†
250-0-0	0.8470	0.8410†	0.5790†
250-0-1	0.7650	0.7620†	0.5500†
250-0-2	0.7140	0.7260 §	0.5210†
250-0-3	0.5850	0.5970 §	0.4840†
250-0-4	0.6990	0.7050 §	0.5640†
250-1-0	0.8450	0.8420†	0.6020†
250-1-1	0.7280	0.7690 §	0.5660†
250-1-2	0.6990	0.7170	0.5320†
250-1-3	0.5960	0.6300 §	0.5010†
250-1-4	0.6900	0.6900≈	0.5070†
250-2-0	0.7610	0.8210 §	0.6120†
250-2-1	0.7662	0.7690 §	0.5040†
250-2-2	0.8012	0.7180†	0.4820†
250-2-3	0.6318	0.5970†	0.4840†
250-2-4	0.7001	0.6970†	0.5230†
†/§/≈		23/20/2	44/1/0

The bold data represents the HV metric of corresponding algorithm is better than other's

Table 3 Box plots of the three algorithms on HV Metric of D-MOCRO, MOLS and MOMA

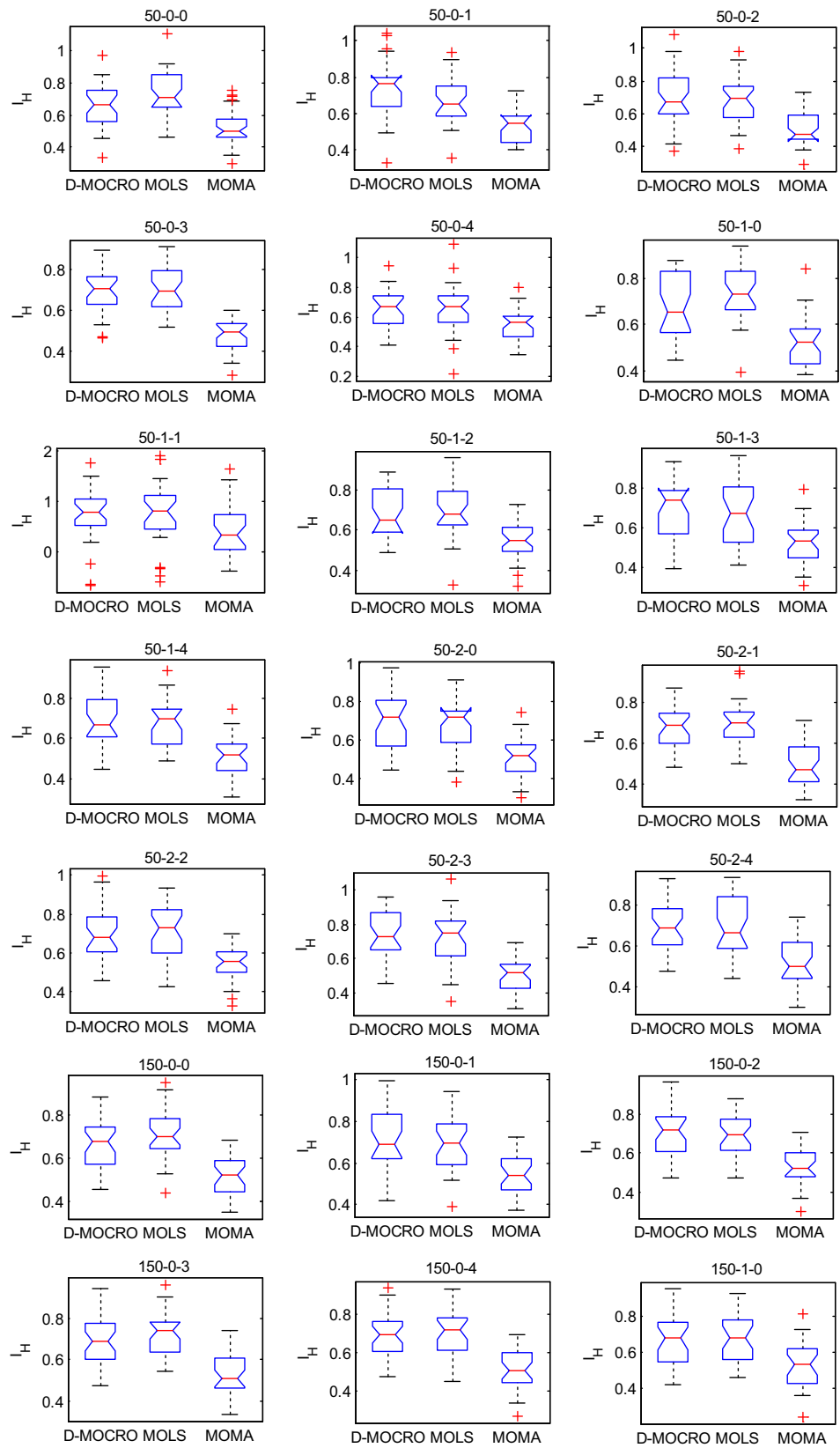
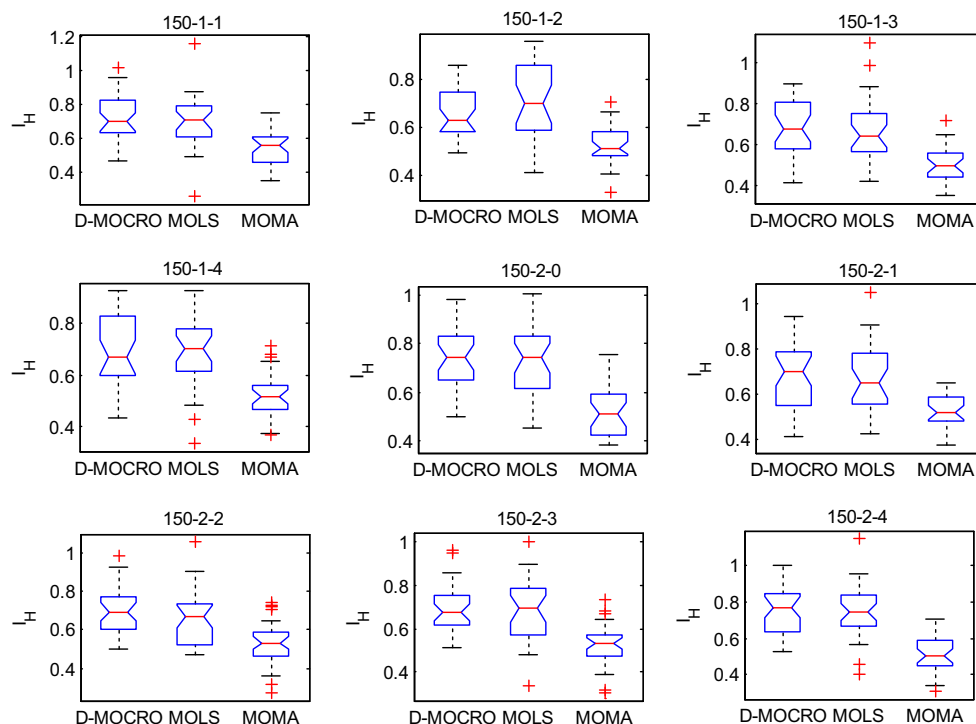


Table 3 continued

VRPTW with two objectives. In this paper, we develop a multi-objective chemical reaction optimization to solve this problem. To balance diversity and convergence, we design local search operators and operators of global convergence. The results of the experiment are shown below.

The average values of the metrics over 30 independent runs of both algorithms on the test sets were calculated and are shown in Tables 2 and 3. Due to space limitations, the standard deviations of the metrics are not presented in the tables.

To test the performance of the D-MOCRO algorithm with other algorithms on different MO-VRPSDPTW benchmarks, we use the Wilcoxon signed-rank test [24] at a significance level of 0.05. The results of the benchmarks are summarized as †/§/≈, which indicate whether the performance of D-MOCRO is better than, worse than or similar to others. From Tables 2, 3 and Fig. 9, several observations can be made.

- (1) For 45 real-world instances, D-MOCRO significantly outperforms MOLS and MOMA in most instances. Specifically, D-MOCRO outperforms MOLS in 23 instances and MOLS is outperformed by MOMA in 22 instances in terms of HV. According to the attribute of HV, the results in Table 2 show that D-MOCRO can obtain better convergence and diversity than MOLS and MOMA.
- (2) To demonstrate the convergence and diversity of the comparative algorithms, we plot the non-dominated

solutions obtained by D-MOCRO (denoted as in blue), MOLS and MOMA on the instance 50-0-1, on $f_1 - f_2$, $f_1 - f_3$ and $f_2 - f_3$ planes in Fig. 9. In Fig. 9, we can see that most objective vectors generated by D-MOCRO are better than those generated by MOMA and MOLS.

- (3) To further show the convergence and diversity properties of the three algorithms, the box-plots of three algorithms are shown in Table 3. The HV metric is used in this paper. From Table 3, we found instances of 50-0-3, 50-0-4, 50-1-1, 50-1-2, 50-1-3, 50-1-4, 50-2-0, 50-2-1, 150-0-0, 150-1-1, 150-1-3, 150-2-0, 150-2-1, 150-2-3, 150-2-4, 250-0-0, 250-0-1, 250-1-0, 250-1-4, 250-2-2, 250-2-3 and 250-2-4, where D-MOCRO was better than other two algorithms. D-MOCRO can search for more solutions using an efficient local search, especially in the instance 50-2-0.

5.4.2 Discussions

In this paper, a chemical reaction optimization algorithm that solves the MO-VRPSDPTW with five objectives has been proposed. The main idea is to use a decomposition-based chemical reaction optimization to solve the MO-VRPSDPTW. When the number of objectives is five, we use an archive to save solutions. We use an efficient non-dominated sort on sequential searches [22] to update the archive of D-MOCRO.

Table 3 continued

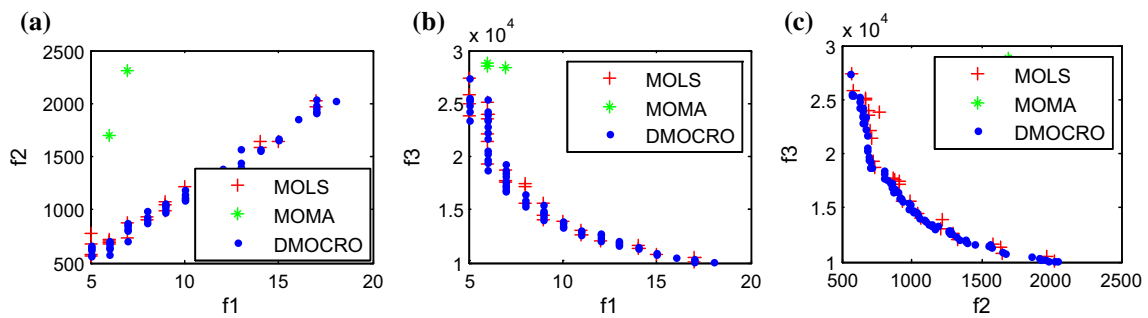
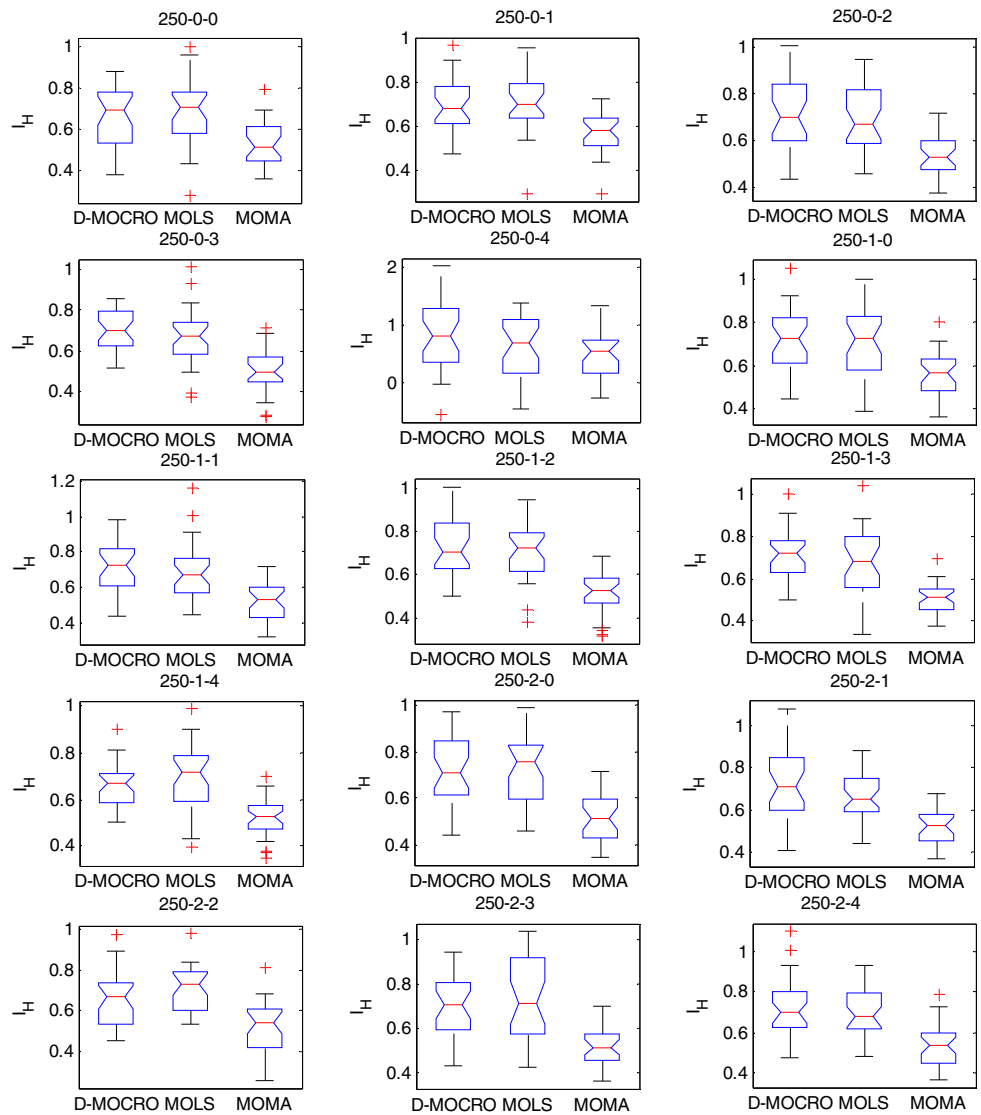


Fig. 9 Plots of nondominated solutions obtained by D-MOCRO (denoted as *dot*), MOLS (denoted as *plus*) and MOMA (denoted as *asterisk*) on the instance 50-0-1, respectively

In D-MOCRO, a chemical reaction optimization algorithm is successfully developed to solve discrete many-objective combinational optimization problem, for the first time. D-MOCRO has good convergence, benefitting from the operator of synthesis. The other operators are used to search locally for neighbors of molecules and can obtain a good diversity.

Finally, it can be concluded that D-MOCRO is an effective mechanism to solve real-world MO-VRPSDPTW. Extensive experiments have shown the effectiveness of the proposed algorithm. In order to further study the performance of D-MOCRO, the proposed algorithm can also be extended to solve other multiobjective VRPs in reverse logistics. Although D-MOCRO was shown to be very promising for real-world MO-VRPSDPTW, however, it has its limitation. The source of the major limitation is the sensitivity to the parameter. In order to enhance its performance, adaptive control of parameter will be developed to make its performance more efficient.

6 Conclusion and future work

In this paper, a discrete multiobjective CRO has been developed to solve multiobjective vehicle routing problem with five objectives, for the first time. In our work, a D-MOCRO is proposed which is based on the method of decomposition. It converts the complex MO-VRPSDPTW into a number of single-objective subproblems and optimizes them in CRO simultaneously. CRO has four operators to search, which includes one molecular operation and two molecular operations. The one molecular operation includes an on-wall ineffective collision operator and decomposition operator, and the two molecular operations include a synthesis operator and inter-molecular ineffective collision operator. The on-wall ineffective collision operator and inter-molecular ineffective collision operator execute a local search. The decomposition operator and synthesis operator execute global search. The energy conservation in the process of chemical reaction balances the local (exploitation) and the global (exploration). In addition, D-MOCRO uses an archive to save non-dominant solutions. The archive updating uses the efficient non-dominated sort on sequential searches [22].

From the analysis and experiments, we observe that the decomposition-based multi-objective chemical reaction optimization enables D-MOCRO to utilize energy conservation balancing the local (exploitation) and global (exploration) to generate better quality solutions. Compared with the other algorithms on the 45 benchmark problems, we can conclude that D-MOCRO significantly outperforms MOLS and MOMA in most instances. Specifically, D-MOCRO outperforms MOLS in 23 instances and MOLS is outperformed by MOLS in 22 instances in terms of HV. According to the

attribute of HV, the results of Table 1 show that D-MOCRO can obtain better convergence and diversity than MOLS and MOMA.

Further work includes research on the adaptive control of parameter to make the algorithm more efficient. Moreover, the algorithm can be applied to constrained, dynamic and noisy many-objectives and multi-objective optimization domain. It is expected that D-MOCRO will be used for many real-world optimization problems.

Acknowledgements The authors would also like to thank Prof. J. Wang for providing the 45 MOVRPTW Instances, Prof. H. L. Liu for providing the source codes of TMOEA/D. This work is partially supported by National Natural Science Foundation of China (Nos. U1534208, 61272283, U1334211), Shaanxi Science and Technology Innovation Project (2015KTZDGY01-04).

References

- Goldberg MC, Asconavieta PH, Goldberg EFG (2012) Memetic algorithm for the traveling car renter problem: an experimental investigation. *Memet Comput* 4(2):89–108
- Xu Y, Lim MH, Ong YS (2008) Automatic configuration of metaheuristic algorithms for complex combinatorial optimization problems. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, pp 2380–2387
- Földesi P, Botzheim J (2010) Modeling of loss aversion in solving fuzzy road transport traveling salesman problem using eugenic bacterial memetic algorithm. *Memet Comput* 2(4):259–271
- Wang C, Mu D, Zhao F et al (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Comput Ind Eng* 83:111–122
- Cherkesly M, Desaulniers G, Laporte G (2015) A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. *Comput Oper Res* 62:23–35
- Tan KC, Chew YH, Lee LH (2006) A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Comput Optim Appl* 34(1):115–151
- Wang J, Zhou Y, Wang Y et al (2016) Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms *IEEE Trans Cybern* 46(3):582–594. doi:10.1109/TCYB.2015.2409837
- Hsu WH, Chiang TCA (2012) Multiobjective evolutionary algorithm with enhanced reproduction operators for the vehicle routing problem with time windows. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 1–8
- Iqbal S, Kaykobad M, Rahman MS (2015) Solving the multi-objective vehicle routing problem with soft time windows with the help of bees. *Swarm Evol Comput* 24:50–64
- Li H, Landa-Silva D (2011) An adaptive evolutionary multi-objective approach based on simulated annealing. *Evol Comput* 19(4):561–595
- Zhang Q, Li H (2007) MOEA/D: a multi-objective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Gong M, Cai Q, Chen X et al (2014) Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Trans Evol Comput* 18(1):82–97

13. Qi Y, Hou Z, Li H et al (2015) A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows. *Comput Oper Res* 62:61–77
14. Lam AYS, Li VOK (2012) Chemical reaction optimization: a tutorial. *Memet Comput* 4(1):3–17
15. Xu J, Lam A, Li VOK (2011) Chemical reaction optimization for task scheduling in grid computing. *IEEE Trans Parallel Distrib Syst* 22(10):1624–1631
16. Duan H, Gan L (2015) Orthogonal multiobjective chemical reaction optimization approach for the brushless DC motor design. *IEEE Trans Magn* 51(1):1–7
17. Bechikh S, Chaabani A, Said L (2015) An efficient chemical reaction optimization algorithm for multiobjective optimization. *IEEE Trans Cybern* 45(10):2051–2064
18. Li H, Wang L, Hei X (2015) Decomposition-based chemical reaction optimization (CRO) and an extended CRO algorithms for multiobjective optimization. *J Comput Sci*. doi:[10.1016/j.jocs.2015.09.003](https://doi.org/10.1016/j.jocs.2015.09.003)
19. Ishibuchi H, Akedo N, Nojima Y (2015) Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Trans Evol Comput* 19(2):264–283
20. Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optim* 8(3):631–657
21. Liu HL, Gu F, Cheung Y (2010) T-MOEA/D: MOEA/D with objective transform in multi-objective problems. In: 2010 international conference of information science and management engineering (ISME), vol 2. IEEE, pp 282–285
22. Zhang X, Tian Y, Jin Y (2015) A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(6):761–776
23. Zitzler E, Multiobjective Thiele L (1999) Evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
24. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 1(6):80–83