2-2021

MIMOA: A membrane-inspired multi-objective algorithm for green vehicle routing problem with stochastic demands

Yunyun NIU

Yongpeng ZHANG

Zhiguang CAO
Singapore Management University, zgcao@smu.edu.sg

Kaizhou GAO

Jianhua XIAO

See next page for additional authors

Swarm and Evolutionary Computation 60 (2021) 100767

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

MIMOA: A membrane-inspired multi-objective algorithm for green vehicle routing problem with stochastic demands

Yunyun Niu[a], Yongpeng Zhang[a], Zhiguang Cao[b], Kaizhou Gao[c], Jianhua Xiao[d,*], Wen Song[e], Fangwei Zhang[f]

[a] School of Information Engineering, China University of Geosciences in Beijing, Beijing 100083, China
[b] Department of Industrial Systems Engineering and Management, National University of Singapore, 117576 Singapore
[c] Macau Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China
[d] The Research Center of Logistics, Nankai University, Tianjin 300071, China
[e] Institute of Marine Science and Technology, Shandong University, Jinan 250100, China
[f] College of Transport and Communications, Shanghai Maritime University, Shanghai 201306, China

A R T I C L E I N F O

Keywords:
Vehicle routing problem
Stochastic demand
Membrane-inspired algorithm
Clustering strategy
Multi-objective evolutionary algorithm

A B S T R A C T

Nowadays, an increasing number of vehicle routing problem with stochastic demands (VRPSD) models have been studied to meet realistic needs in the field of logistics. In this paper, a bi-objective vehicle routing problem with stochastic demands (BO-VRPSD) was investigated, which aims to minimize total cost and customer dissatisfaction. Different from traditional vehicle routing problem (VRP) models, both the uncertainty in customer demands and the nature of multiple objectives make the problem more challenging. To cope with BO-VRPSD, a membrane-inspired multi-objective algorithm (MIMOA) was proposed, which is characterized by a parallel distributed framework with two operation subsystems and one control subsystem, respectively. In particular, the operation subsystems leverage a multi-objective evolutionary algorithm with clustering strategy to reduce the chance of inferior solutions. Meanwhile, the control subsystem exploits a guiding strategy as the communication rule to adjust the searching directions of the operation subsystems. Experimental results based on the ten 120-node instances with real geographic locations in Beijing show that, MIMOA is more superior in solving BO-VRPSD to other classical multi-objective evolutionary algorithms.

1. Introduction

Vehicle routing problem with stochastic demand (VRPSD) was first introduced by Bertsimas in 1992 [1]. Like in many other vehicle routing problems (VRPs), vehicles depart from a depot and serve geographically dispersed customers, and return to the depot after fulfilling delivery tasks. The major difference of VRPSD from other VRPs is that, the demand of a customer follows a probabilistic distribution and is unknown until the vehicle reaches the location of customer. So far, various models have been investigated to solve VRPSD, such as chance-constraint programming, dynamic programming, and multi-scenario approach [2,3]. Among them, the most widely studied models are those based on two-stage stochastic programming [4]. Specifically, in the first stage, a priori route is planned to offer a guidance for the vehicle, based on the order of which the customers will be visited. In the second stage, the vehicles will travel on the a priori route, and if a vehicle does not have adequate goods to satisfy a customer, it has to return to the depot for replenishment. Afterwards, the vehicle goes back to continue serving the same customer. In this model, the feasibility of a priori route may change due to the uncertain demands and some internal operations of an algorithm, which can be recovered by corrective actions, i.e., *recourse*. The final solution to VRPSD is an optimal complete route begins and ends in the depot and successfully serves all customers with stochastic demands. Due to the uncertainty of demands, VRPSD is considered as more intricate than VRPs with deterministic demands [5,6].

Since the major settings in VRPSD are more in line with the real-world applications, many variants of VRPSD have been studied to model realistic problems, most of which were addressed by heuristics or meta-heuristics algorithms [7]. Among them, a two-echelon VRPSD in city logistics was solved by a genetic algorithm based approach in [8], where stochastic programming with recourse is applied to minimize the travel cost and failure route cost. An adaptive large neighborhood search heuristic was developed to tackle the VRPSD with weight-related cost in [9]. A nature-inspired approach, that integrates glowworm swarm optimization algorithm, variable neighborhood search algorithm and path re-linking algorithm was proposed to solve VRPSD in [10]. A hybrid meta-heuristic to cope with the route duration constraints in VRPSD

* Corresponding author.
E-mail address: jhxiao@nankai.edu.cn (J. Xiao).

https://doi.org/10.1016/j.swevo.2020.100767
Received 26 July 2019; Received in revised form 2 April 2020; Accepted 29 August 2020
Available online 4 September 2020

was presented in [11]. In addition, VRPSD was also extended to consider multi-objectives to meet more realistic needs in [12,13], where the objectives were typically chosen among total cost, total travelling distance, driver remuneration, waiting time, number of vehicles and workload balance and so on.

From the perspective of enterprise development, the total cost and customer satisfaction are deemed as two most crucial factors, which are related to their current and long-term earnings, respectively. On one hand, fuel emission cost is one of the most significant components in the total cost, as energy-saving plays a crucial role in both the economic and environmental concerns. Hence, a lot of studies on green road freight transportation that aim at reducing the carbon dioxide emissions from logistic transportation [14–19] have been conducted. Readers may also refer to [20] for a fundamental survey of green VRP, fuel consumption and vehicle emission models. On the other hand, responsiveness to customers is one of the most important factors of evaluating the service quality, based on which the customer satisfaction can be measured. Prompt services have positive influence on the reputation of an enterprise, and may potentially increase the long-term economic interests. However, in most cases, minimizing fuel emission and improving customer satisfaction are conflicting with each other, and the tradeoff between economic and environmental concerns and customer satisfaction was considered as one of promising directions in the green VRPs [21]. Nevertheless, to the best knowledge, this subject has been rarely studied or addressed [22], especially in the case of VRPSD. Therefore, this paper mainly emphasizes on optimizing the tradeoff between total cost and customer satisfaction in the context of VRPSD.

In general, VRPSD in this paper can be modelled as a bi-objective (total cost and customer satisfaction) vehicle routing problem with stochastic demand (BO-VRPSD), which comes down to a multi-objective optimization problem in nature. Recently, a new bio-inspired optimization model called membrane algorithm (MA) has been introduced to cope with multi-objective optimization problems [23,24]. The MA usually has cell-like or tissue-like structure, since it is inspired by the structure and dynamic activities of living cells, tissues, or organs [25,26]. Moreover, it offers distributed parallel structure to many optimization algorithms (rules) [27–29]. Meanwhile, various multi-objective MA(s) have been proposed to deal with practical problems [30–34], and it is also demonstrated that they can outperform many other types of multi-objective optimization algorithms in a diversity of applications [35–37]. In addition to those classic works, Zhang et al. [38] designed a multi-objective membrane algorithm that combined membrane system and quantum heuristic to solve the multi-objective knapsack problem. Zaharie et al. [39] proposed a new strategy to apply the genetic operators to the conventional and distributed evolutionary algorithms, respectively. Cheng et al. [40] developed a membrane heuristic evolutionary algorithm based on population P system and differential evolution to solve the multi-objective optimization problems. Liu et al. [41] conceived an evolutionary operator to improve the search efficiency, where the non-dominated sorting and crowding distance were introduced into the skin membrane. Although a lot of success has been achieved for the membrane related algorithms, specific design on how to address BO-VRPSD has been rarely studied.

In this paper, a new membrane-inspired multi-objective algorithm (MIMOA) with three membranes (subsystems) is proposed to solve BO-VRPSD, which aims to optimize the total cost and customer dissatisfaction given stochastic customer demands. MIMOA adopts a clustering strategy in operation subsystems and a guiding strategy in control subsystem to govern the direction of population evolution. The main contribution of this paper is summarized as follows.

- To the best knowledge, MIMOA is the first membrane-inspired algorithm applied to solve BO-VRPSD, which aims to seek optimal tradeoff between total cost and customer dissatisfaction with stochastic customer demands.
- The clustering strategy in the operation subsystems divides customers into clusters, and prescribes the scope for the operations of crossover and mutation, which helps to avoid the potentially inferior crossover and mutation of chromosomes between distant zones.
- The guiding strategy in the communication rule is designed based on a skin membrane control strategy, where the multi-population co-evolution in the three subsystems enables the solution to converge faster.

The remainder of this paper is organized as follows. The mathematical model of the BO-VRPSD is proposed in Section 2. An efficient algorithm for solving the BO-VRPSD, i.e., MIMOA, is designed and presented in Section 3. The performance of the MIMOA is analyzed by comparing it with other classic multi-objective algorithms in Section 4. Finally, conclusions and future works are listed in Section 5.

2. Problem description

The BO-VRPSD is defined on a complete graph $G = (N, E)$, where $N = \{0, 1, 2, \dots, n\}$ is the set of customers and the depot (i.e., 0), and E is the set of arcs between nodes. The random demand of customer i is denoted by d_i, which is unknown until a vehicle arrives at that customer. Each vehicle i has a velocity v_i and a capacity c_i. The vehicle has to return to the depot for replenishment if the goods in the vehicle cannot satisfy the customers' demand. The routes which vehicles have to travel through in order to replenish are defined as *extra routes*. Accordingly, the mathematical model of BO-VRPSD is defined as follows,

$$\text{Minimize } \{f_1(x), f_2(x)\} \text{ subject to } x \in \mathbf{D}, \tag{1}$$

where $f_1(x)$ represents the total cost including carbon emission cost of all vehicles when they are travelling and wages of all drivers, $f_2(x)$ represents the total value of customer dissatisfaction, and $\mathbf{D}$ is the decision space.

2.1. The total cost

In this paper, the total cost includes fuel emissions cost and the driver wages, and is expressed as follows,

$$T = F + W, \tag{2}$$

where F represents fuel emissions cost and W denotes the driver wages.

Assume e represents the complete route of a vehicle, then the nodes sequentially visited by the vehicle is described as follows,

$$\Omega(e) = \langle 0, n_1(e), n_2(e), \dots, n_m(e), 0 \rangle, \tag{3}$$

where 0 represents depot and $n_i(e)$ represents the ith customer node the vehicle served in route e. In the meantime, the fuel emissions cost of route e is calculated as follows,

$$F_t(e) = F_{0,n_1(e)} + \sum_{j=1}^{m-1} F_{n_j(e),n_{j+1}(e)} + F_{n_m(e),0} + E_t(e), \tag{4}$$

where F_{0,n_1} is the fuel emissions cost from the depot to customer 1. Similarly, $F_{n_j(e),n_{j+1}(e)}$ is the fuel emissions cost from customer j to customer $j+1$. $E_t(e)$ is the amount of fuel emissions cost for extra routes in e. Accordingly, the total fuel emissions cost for all vehicles is defined as follows,

$$F = \sum_{j=1}^{k} F_t(e_j), \tag{5}$$

where k is the pre-defined number of vehicles and e_j denotes the route served by the jth vehicle. And homogeneous vehicles will be considered in this model. Moreover, two constraints were also proposed to ensure that each customer cannot be served by different vehicles as follows,

$$\cup_{i=1}^{k} \Omega(e_i) = N, \tag{6}$$

Table 1
Notations in Eq. (7) and their values.

Notation	Description	Value
R	Engine speed (rev/s)	36.67
V	Engine displacement (L)	6.9
l	Engine friction factor (kj/rev/liter)	0.20
A	Frontal surface area (m^2)	8.0
C_d	Coefficient of aerodynamics drag	0.7
C_r	Coefficient of rolling resistance	0.01
ξ	Fuel-to air mass ratio	1
τ	Acceleration (m/s^2)	0
n_{tf}	Vehicle drive train efficiency	0.45
η	Efficiency parameter for diesel engines	0.45
κ	Heating value of a typical diesel fuel (kj/g)	44
ψ	Conversion factor (g/s to L/s)	737
g	Gravitational constant (m/s^2)	9.81
ρ	Air density (kg/m^3)	1.2041

$$P(e_i) \cap P(e_j) = \{0\},\ i \neq j, \tag{7}$$

where $P(e_i)$ represents the set of customers served by the ith vehicle.

Moreover, the fuel consumption on certain route in Eq. (4) can be further calculated as follows [17,18],

$$F_{n_i(e),n_j(e)} = \lambda f_c \left(lRVd_{i,j}/v + M\gamma\alpha d_{i,j} + \beta\gamma d_{i,j} v^2 \right), \tag{8}$$

where $\lambda = \xi/\kappa\psi$, $\gamma = 1/1000 n_{tf}\eta$, $\alpha = \tau + gC_r$, $\beta = 0.5C_d\rho A$; f_c is the fuel emissions cost per liter; $d_{i,j}$ is the distance between $n_i(e)$ and $n_j(e)$; v refers to the speed of vehicles; and M is the total weight of the vehicle. Particularly, Eq. (8) involves three modules, i.e., the engine module ($lRVd_{i,j}/v$), the weight module ($M\gamma\alpha d_{i,j}$), and the speed module ($\beta\gamma d_{i,j}v^2$). Notations and their default values are listed in Table 1.

In addition, the driver wages are defined as follows,

$$W = \sum_{i=1}^{k} W_i = f_d s_i, \tag{9}$$

where W_i is the wage of driver i, f_d represents the wages of the driver per hour, and s_i means the working duration of driver i.

2.2. Customers dissatisfaction

The dissatisfaction model in this paper is developed based on the soft and hard time windows [22], which are characterized by a four-dimensional vector $[ah_i\ as_i\ bs_i\ bh_i]$. More specifically, the vector $[as_i, bs_i]$ represents the soft time window, and the vector $[ah_i, bh_i]$ represents the hard time window. Violation of the soft time window will cause dissatisfaction to customer, and violation of the hard time window will maximize the customer dissatisfaction value.

Accordingly, the dissatisfaction level (DSL) of customer i, which also refers to $f_2(x)$ in Eq. (1), is expressed as follows,

$$DSL_i = \begin{cases} 1 & 0 < y_i \leq ah_i \\ (as_i - y_i)/(as_i - ah_i) & ah_i < y_i \leq as_i \\ 0 & as_i < y_i \leq bs_i, \\ (y_i - bs_i)/(bh_i - bs_i) & bs_i < y_i \leq bh_i \\ 1 & y_i > bh_i, \end{cases} \tag{10}$$

where y_i is the arrival time of the vehicle that supposed to serve customer i.

3. Algorithm

In this section, the MIMOA was proposed to solve the BO-VRPSD model, where only homogeneous vehicles are considered. The MIMOA has a tissue-like structure with three membranes which represent one control subsystem and two operation subsystems, as depicted in Fig. 1. Particularly, control subsystem governs the evolutionary directions of the two operation subsystems by employing a guiding strategy in the communication rule; and operation subsystems leverage evolutionary algorithm and clustering strategy to search better solutions and send them back to the control subsystem for a new round of evolution. In specific, control subsystem not only receives solutions from the operation subsystems, but also distributes back a modified solution according to the status of operation subsystems, so that the whole searching process would be ameliorated. Accordingly, transmission channels between operation subsystems and the control subsystem are bidirectional.

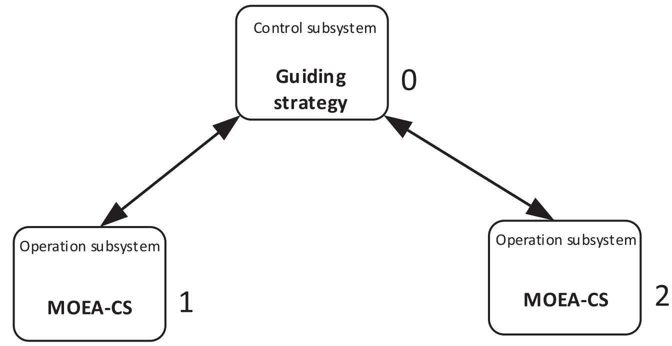


Fig. 1. Three subsystems of MIMOA.

The main procedure of the MIMOA is described in Algorithm 1,

Algorithm 1 Main framework of MIMOA

Input: population size S and customer data D.
Output: P_0.
D' = clustering_control.clustering(D)
$P_0 \leftarrow \emptyset$
for $i = 1 \rightarrow 2$ **do**
  P_i = clustering_control.initialization(D', S)
while maximum iteration number is not reached **do**
  for $i = 1 \rightarrow 2$ **do**
    P_i^c = evaluation(P_i)
    P_i^m = nondominated_sorting(P_i^c)
    P_i^l = clustering_control.crossover(P_i^m)
    P_i^r = clustering_control.mutation(P_i^l)
    P_i^n = local_search(P_i^r)
    P_i = truncation(P_i^n, S)
    $P_0 \leftarrow P_0 \cup P_1 \cup P_2$
  P_0 = truncation(P_0, S)
  P_0, P_1, P_2 = communication_strategy(P_0, P_1, P_2)

where P_0 represents the population in the control subsystem, and P_1 and P_2 represent the populations in the two operation subsystems, respectively. In each operation subsystem, a clustering strategy, a local search algorithm, and a truncation operator are implemented together with the genetic operations. The clustering strategy is designed to divide customers to several zones according to their geographic locations before assigning them to appropriate routes. In doing so, it reduces the probability of wrong evolutionary direction and also endorses the algorithm with better convergence. The local search is used to bootstrap the solution. The truncation operator [43] is performed on populations P_1, P_2 and P_0 as well, to eliminate individuals with lower ranks. In addition, the communication of the whole system is designed based on a guiding strategy [42], to further enhance convergence and diversity of solution sets.

3.1. Chromosome representation

In order to better differentiate the partial and whole route, from here on, *route* and *complete route* are used to denote them, respectively. Generally, a *complete route* consists of multiple *routes*. In the proposed MI-

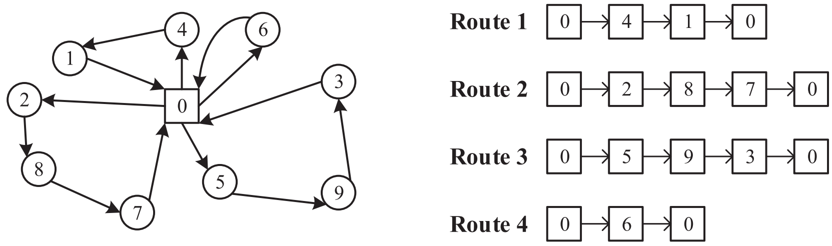


Fig. 2. A complete route and its chromosome representation.

MOA, variable-length chromosomes [12,44,45] are adopted to represent routes, which are illustrated in Fig. 2. Particularly, a chromosome, consisting of several customer nodes, represents a route out of the complete route. Each vehicle starts its route from the depot at beginning, and returns to the depot after completeness.

3.2. Algorithms in operation subsystems

In each operation subsystem, a multi-objective evolutionary algorithm with clustering strategy (MOEA-CS) was proposed to find candidate solutions for the BO-VRPSD.

3.2.1. Clustering strategy

At the beginning of the algorithm, the information for customers and vehicles is taken as input. A clustering strategy based on K-Means algorithm is designed to classify customers according to geographic location, in which Euclidean distance is adopted as the similarity index. The specific steps are described in Algorithm 2. Taking the illustration in Fig. 3 as an example, if the number of clusters is set as four, the seventeen customers will be divided into four zones by applying the clustering strategy. After clustering, each customer has an initial label regarding the zone it belongs to, and it can be changed when applying the adjacent swap operations.

Algorithm 2 Clustering strategy

Input: Number of clusters K, customer data D and Maximum number of iterations M.
Output: Cluster Set $C = \{C_1, C_2, \ldots, C_K\}$.

$U \leftarrow \{\}$
$U_{tmp} \leftarrow \{\}$
for $k = 1 \rightarrow K$ **do**
  u_k = random_sampling_without_replacement(D)
  $U \leftarrow U \bigcup u_k$
for $m = 1 \rightarrow M$ **do**
  for $k = 1 \rightarrow K$ **do**
    $C_k \leftarrow \{\}$
  for $i = 1 \rightarrow len(D)$ **do**
    for $k = 1 \rightarrow K$ **do**
      $d_{i,k}$ = Euclidean_distance($D[i], U[k]$)
    $d_{i,t} = \min\{d_{i,1}, d_{i,2}, \ldots, d_{i,K}\}$
    $C_t \leftarrow C_t \bigcup D[i]$
  for $k = 1 \rightarrow K$ **do**
    $u_k \leftarrow \frac{1}{|C_k|} \sum_{c \in C_k} c$
    $U_{tmp} \leftarrow U_{tmp} \bigcup u_k$
  if $U == U_{tmp}$ **then**
    break
  else
    $U \leftarrow U_{tmp}$
    $U_{tmp} \leftarrow \{\}$
return $C \leftarrow \{C_1, C_2, \ldots, C_K\}$

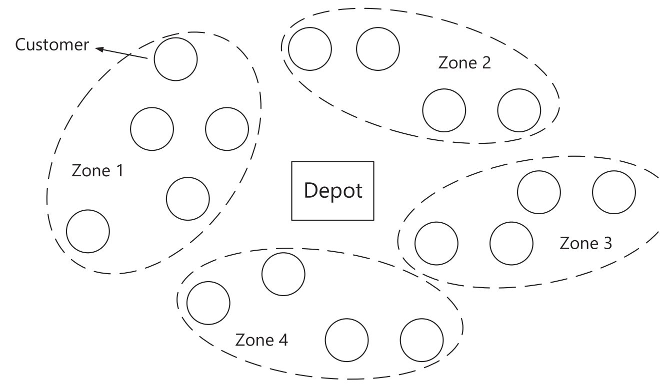


Fig. 3. A example of clustering strategy.

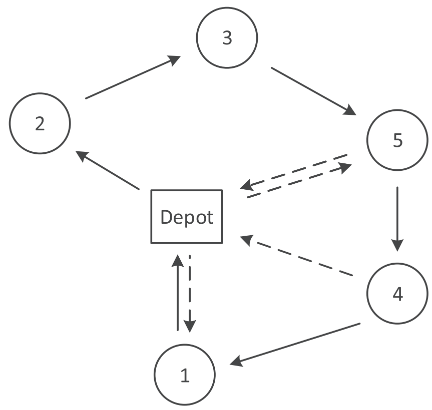
3.2.2. Initialization

At the initialization stage, each vehicle only serves customers belonging to the same zone, and customers in different zones can not be assigned to the same route. In each zone, a customer sequence is generated randomly to identify the probable service order, and customers are sorted according to the sequence one by one. A customer with higher priority is always preferred in comparison to the ones with lower priorities, unless the mean of its demand exceeds the remaining capacity of the vehicle at that moment. For example, suppose that the vehicle capacity is 50, the customer sequence in a zone is [1, 2, 3, 4, 5, 6, 7], and their corresponding demands are 22, 15, 15, 20, 12, 20, 10, respectively. The first route begins at the depot and adds customer 1 and 2 in turn. After serving the first two customers, the remaining capacity of the vehicle is 13. At that time, it cannot satisfy the demand of customer 3, but can satisfy customer 5, so it will assign customer 5 to its route. Then, the remaining capacity becomes 1, and no more customer can be served by this vehicle. Afterwards, another route will start. The same procedures will repeat until all customers are served. Consequently, three routes, i.e., 0–1–2–5–0, 0–3–4–7–0, and 0–6–0, would be generated. And each vehicle is in charge of one route.

3.2.3. Evaluation and selection

Chromosomes are evaluated and ranked based on the Pareto dominance, where a binary tournament is adopted to choose parent chromosomes for genetic operators. In particular, a pair of chromosomes are chosen randomly, and the one with lower rank is picked out for reproduction. This process will repeat until sufficient parent chromosomes are acquired.

The route simulation method (RSM) [45] is adopted to estimate the expected cost of the solution. An example of a route sequence is illustrated in Fig. 4, in which solid lines represent planned routes before departure, and dashed lines represent extra route incurred in actual service. Suppose that the capacity of each vehicle is 50, and arrows point to the heading directions of the vehicle. In this example, the vehicle leaves the depot and first arrives at customer 2, after which the remaining capacity of the vehicle is 30. Then the vehicle visits customer 3 and 5 subsequently. When it reaches customer 5, the remaining capacity



Customer	Real demand
1	10
2	20
3	25
4	35
5	15

Fig. 4. An example of RSM.

can not meet the demand. The vehicle unloads all the remaining goods and return to the depot to replenish goods. After that, it goes back to customer 5 to continue serving. Then the vehicle continues to visit customer 4, after which the remaining capacity of the vehicle becomes 0. It has to return to the depot again, and continues to visit customer 1 and other remaining customers (if any) after replenishment. Upon the completeness of serving all customers, the vehicle returns to the depot for termination. Given the randomness of the customer demands, the above operations are repeated several times for each complete route, and the average is considered as the travel cost of this complete route.

3.2.4. Route-exchange crossover

MIMOA adapts an existing route-exchange crossover [45], which is ameliorated by incorporating geographic location constraints. Only routes belonging to the same zone (their customers with the same label) can be exchanged with each other, as shown in Figure 5. Sequences of routes in one chromosome are reproduced and shared with other chromosomes. When a route is inserted to another chromosome as a new route, duplicated customers are deleted from the original route to ensure feasibility of the chromosome.

3.2.5. Mutation operators

In the proposed MIMOA, four mutation operators are designed to explore a larger search space.

(1) *Single node swap operator*: This operator randomly chooses two customer nodes with the same label from two different routes, and exchanges them, as depicted in Fig. 6(a).
(2) *Route merging operator*: This operator combines two shorter routes with the same label, as depicted in Fig. 6(b). It is executed only when there is a route with length smaller than a threshold, which will be merged with the remaining shortest route. The motivation is that, too rare customer requests for a single route may cause a waste of the vehicle capacity, thus increase the cost for the complete route.
(3) *Route splitting operator*: This operator breaks a long route into two routes at a random node, as depicted in Fig. 6(c). This operation is performed only when there is a route with length exceeding a threshold. The motivation is that, too many customer requests for a single route may cause more potential failures.
(4) *Adjacent swap operator*: This operator randomly selects two nodes from two adjacent zones, and exchanges them, as depicted in Figure 6(d). If a customer is reassigned to another zone, its label will be changed accordingly.

3.2.6. Local search

After generic operations, several local search algorithms are designed to improve the quality of current populations. They also help to identify gaps in the Pareto front while enhancing the convergence. Three local search algorithms are adopted in MIMOA, the first two of which are modified based on the ones in [45,46].

(1) Shortest Path Search (SPS): It rearranges customers in a particular route. The customer farthest from the depot will be placed to the beginning of the route, and the customer closest to the first customer will be chosen as the second customer. This step will be repeated until all customers are reordered.
(2) Which Directional Search (WDS): It constructs a new route by reversing the direction of a given route. If the new route is better, it will replace the original one.
(3) Time Priority Search (TPS): It reorders the customers in each route according to the time window constraints of each customer. The priority of a customer is determined based on the end of time windows. In general, the smaller the end time, the higher the priority.

3.3. Communication rule in control subsystem

Control subsystem implements the communication rule based on a guiding strategy, in order to facilitate the search directions of operation subsystems. Transmission channels between operation subsystems and control subsystem are bidirectional, which are governed by the guiding strategy. Firstly, operation subsystems send solutions in the archive population to the control subsystem, and control subsystem performs non-dominated sorting and truncation operations [43] to eliminate the inferior solutions. Then the updated solutions in control subsystem will be distributed to operation subsystem according to convergence and distribution of populations in operation subsystem [42]. For any solution s^* in control subsystem, the number of solutions in operation subsystems which is dominated by s^* is calculated. Then, s^* will be allocated to the operation subsystem which has the more dominated solutions. If two operation subsystems have the same number of dominated solutions, s^* will be allocated to the operation subsystem with smaller crowding distance. It should be notedthat if the operating subsystem already contains a solution assigned by the control subsystem, the duplicate solution will not be received any more. Finally, those inferior solutions are eliminated, and the population size of the operation subsystem is reduced to a specified value by the truncation operator. Consequently,

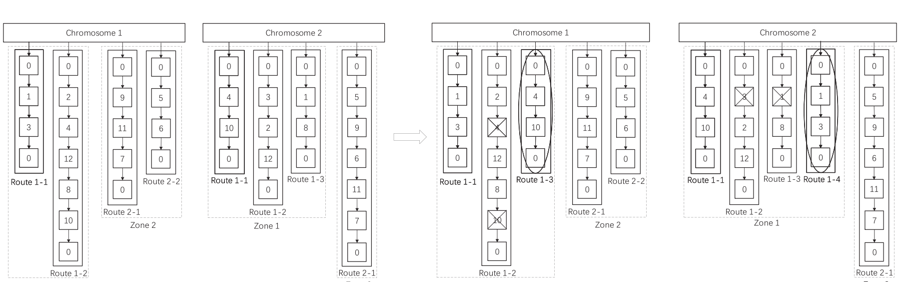
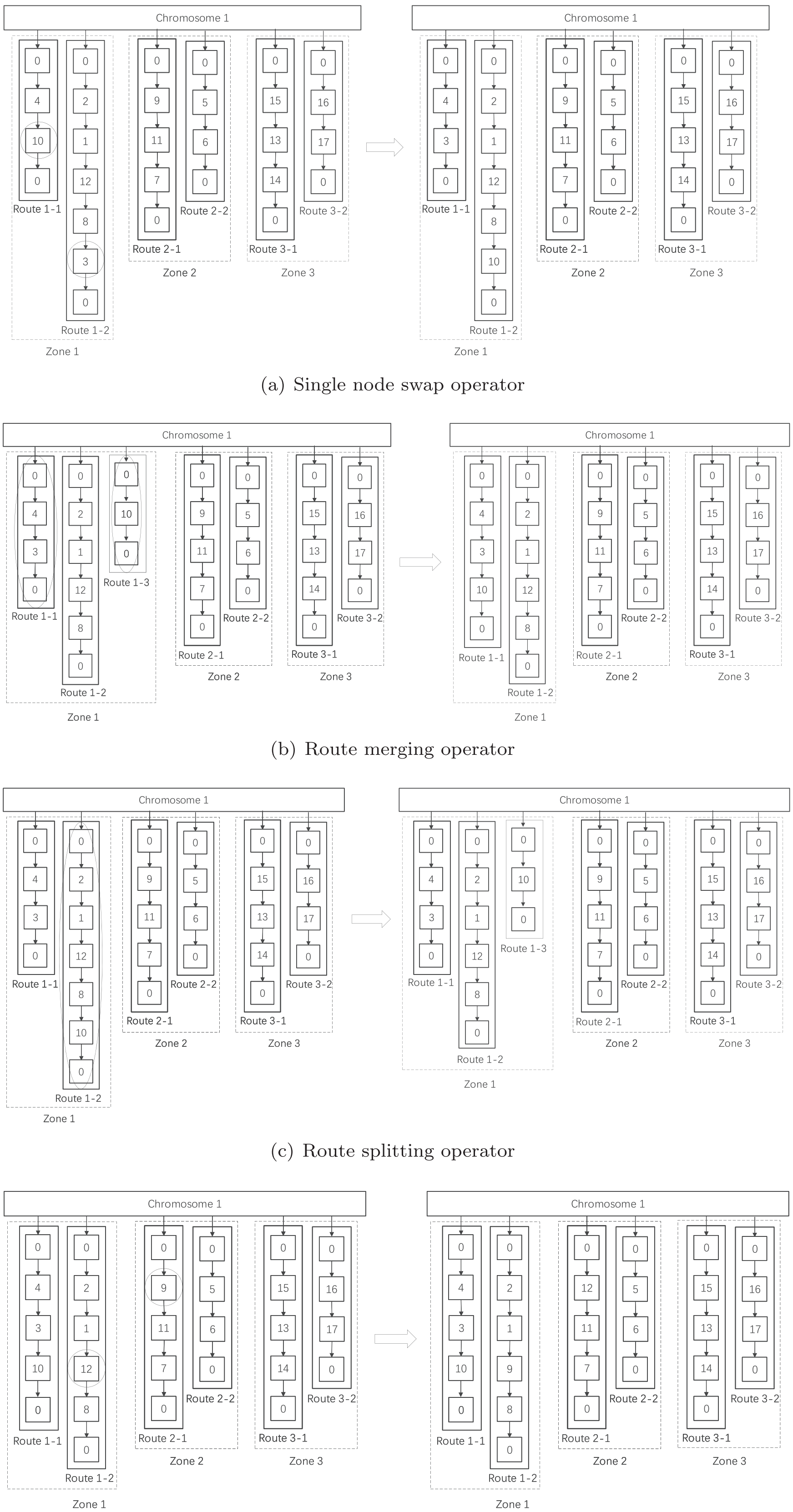


Fig. 5. Route-exchange crossover.



(a) Single node swap operator

(b) Route merging operator

(c) Route splitting operator

(d) Adjacent swap operator

Fig. 6. Mutation operators.

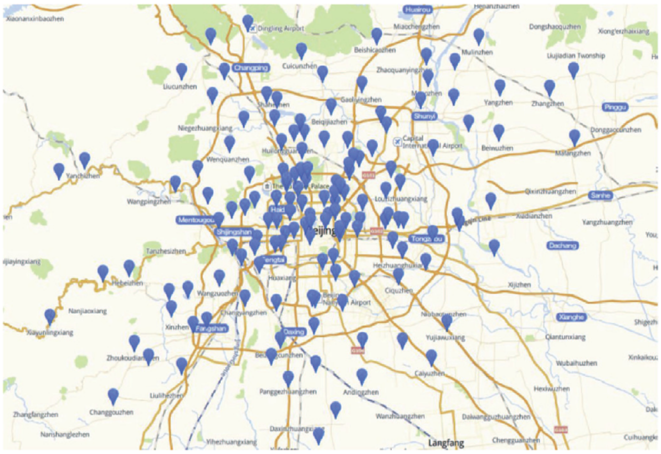


Fig. 7. Customer nodes.

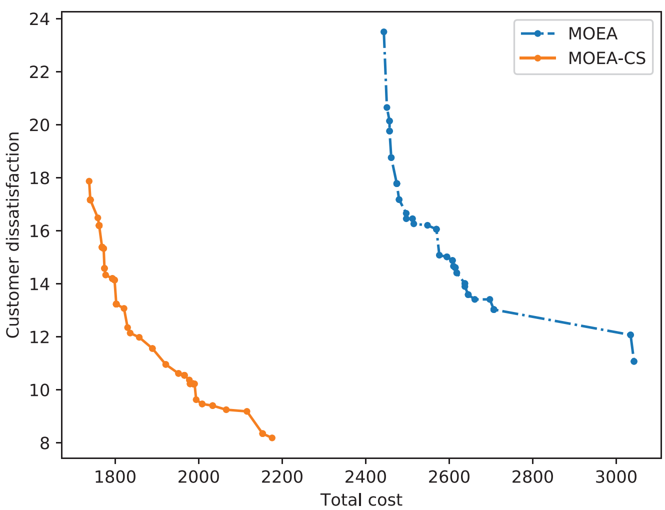


Fig. 8. The non-dominated solution sets of MOEA-CS and MOEA.

Table 2
Parameters.

Parameter	Value
Population size	300
Maximum iteration number of MIMOA	500
Number of clusters	4
Crossover rate	0.7
Mutation rate	0.4
Merge threshold	7
Split threshold	15
RSM sampling time	10
Capacity of vehicle	12,500

new populations are generated, and all operation subsystems are updated.

4. Experimental studies

The MIMOA was implemented in Python and experiments were conducted on Windows 10 with the AMD Rayzen7 1700x processor. The key experimental parameters are listed in Table 2.

As there is no commonly used benchmark dataset for the BO-VRPSD in the literature, the problem instances for testing are generated based on the actual geographic distance for customers in Beijing. As shown in Fig. 7, it contains 10 different instances with 120 customers. Time windows and service time for each customer are generated randomly. Mean demand of each customer is obtained according to a discrete uniform distribution within the interval [300, 1800], and standard deviation is randomly generated which falls between 0 and 1/3 of the mean demand of the customer. As shown in Table 2, the capacity of vehicle is set of 12500.

4.1. Performance of clustering strategy

In this section, to verify the effectiveness of clustering strategy, the multi-objective evolutionary algorithm with clustering strategy (MOEA-CS) is compared with MOEA, which is implemented in operation subsystems.

In general, MOEA-CS is supposed to guide the evolution direction of solutions proactively and alleviate the disadvantages caused by incorrect evolution, thus increase the chance of finding better solutions.

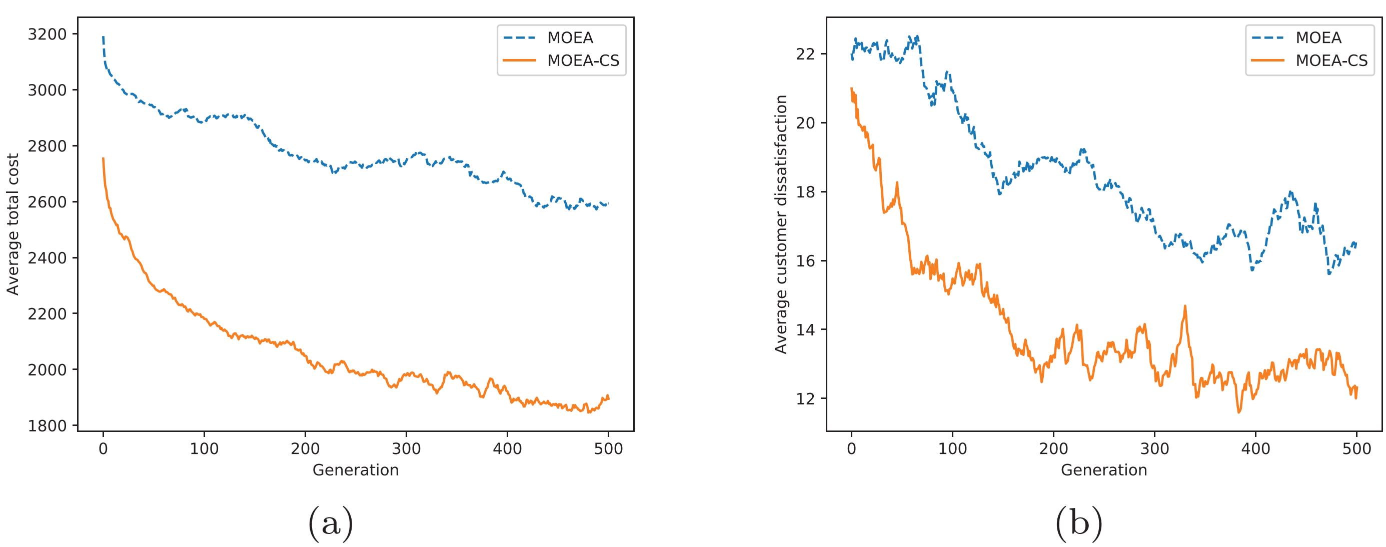
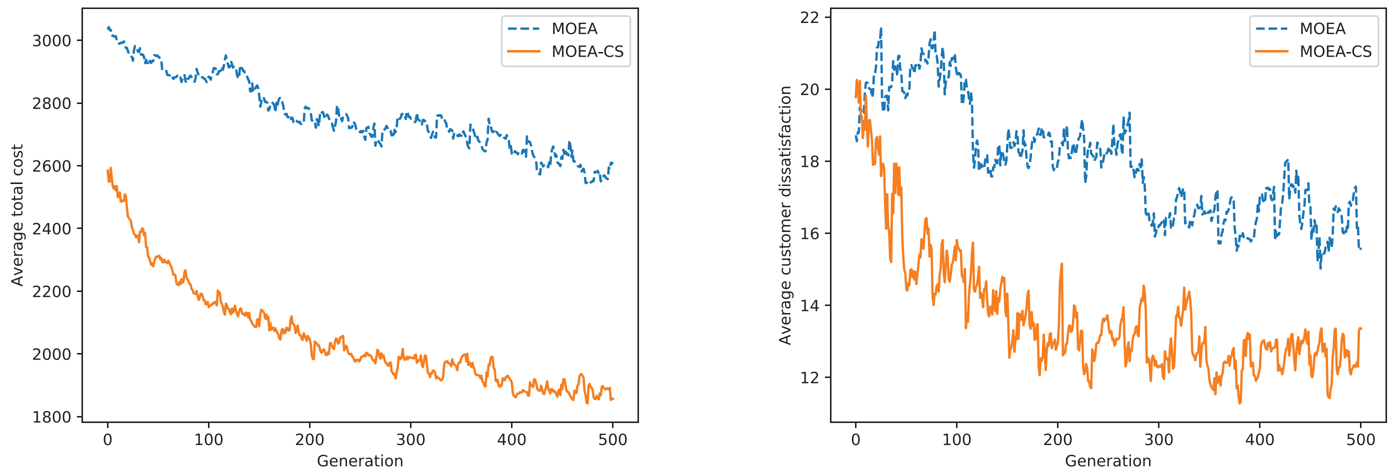


Fig. 9. (a) Average total cost and (b) average customer dissatisfaction of archive populations for MOEA-CS and MOEA.



(a) (b)

Fig. 10. (a) Average total cost and (b) average customer dissatisfaction of non-dominated solutions for MOEA-CS and MOEA.

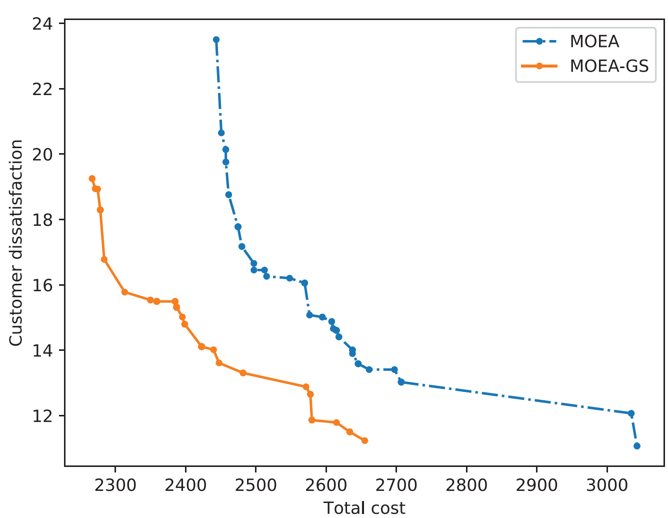


Fig. 11. The non-dominated solution set of MOEA-GS and MOEA.

Fig. 8 shows that the lowest cost value and the lowest customer dissatisfaction achieved by MOEA-CS are 1737.64 and 8.18, respectively, both of which are significantly superior to that of MOEA. Fig. 9(a) and (b) shows the convergence of the mean of the total cost and customer dissatisfaction in solution sets, which cover all solutions in the archived population and generations. Fig. 10(a) and (b) shows the convergence of the mean of the total cost and customer dissatisfaction in non-dominated solutions for MOEA-CS and MOEA, where the total cost and customer dissatisfaction are averaged over the 10 runs. Combining Figs. 9 and 10, it is obvious to see that MOEA-CS can converge faster than MOEA.

The reason for the better performance yield by the clustering strategy can be explained as follows. (1) During initialization, the customer nodes are clustered according to the geographical location, based on which, an initial solution is generated. This type of initial solution is generally more reasonable than those generated directly based on all customer nodes, which also facilitates speeding up the convergence. (2) The clustering strategy helps to limit the scope of crossover and mutation. It states that another chromosome involved in crossover and mutation needs to be in the same region or in an adjacent region, so that the route for each vehicle might not include nodes far away from each other. Moreover, together with the solution quality improving by local search, the chance of yielding poor solutions after cross mutation would be significantly reduced.

Table 3
Performance comparison for different optimization algorithm.

Instances		NSGAII	SMG-MOMA	MIMOA
BJ120_01	Min *TC*	2443.49	2266.78	**1781.44**
	Min *CD*	11.06	11.23	**8.33**
	Min *TC* × *CD*	33677.80	29828.69	**16159.71**
BJ120_02	Min *TC*	2196.63	2102.81	**1675.21**
	Min *CD*	11.25	9.67	**7.05**
	Min *TC* × *CD*	33678.77	27022.15	**13620.02**
BJ120_03	Min *TC*	2404.58	2279.57	**1784.54**
	Min *CD*	10.68	9.48	**7.14**
	Min *TC* × *CD*	29299.79	24475.85	**13725.37**
BJ120_04	Min *TC*	2183.62	2077.51	**1651.78**
	Min *CD*	13.75	12.81	**8.23**
	Min *TC* × *CD*	37276.04	34055.46	**15176.42**
BJ120_05	Min *TC*	2338.58	2295.97	**1915.95**
	Min *CD*	9.47	10.71	**7.07**
	Min *TC* × *CD*	24761.09	31292.90	**15116.97**
BJ120_06	Min *TC*	2488.53	2234.51	**1804.48**
	Min *CD*	8.95	9.72	**6.09**
	Min *TC* × *CD*	26869.85	24072.22	**12217.33**
BJ120_07	Min *TC*	2391.07	2395.87	**1984.14**
	Min *CD*	8.27	8.29	**7.01**
	Min *TC* × *CD*	21373.30	20737.66	**15196.90**
BJ120_08	Min *TC*	2425.62	2336.72	**1821.99**
	Min *CD*	7.89	8.03	**7.0**
	Min *TC* × *CD*	22154.81	22195.16	**13488.27**
BJ120_09	min *TC*	2321.87	2237.77	**1758.13**
	Min *CD*	11.12	10.03	**6.49**
	Min *TC* × *CD*	30585.92	26920.64	**12561.85**
BJ120_10	Min *TC*	2481.82	2397.21	**2041.19**
	Min *CD*	7.99	7.71	**7.07**
	Min *TC* × *CD*	23436.88	21854.76	**16243.08**

4.2. Performance of guiding strategy

One of the major effectiveness for the membrane framework in MIMOA lies in the communication between control subsystem and two operation systems, where control subsystem leverages a guiding strategy to speed up convergence and improve the quality of solutions. To ver-

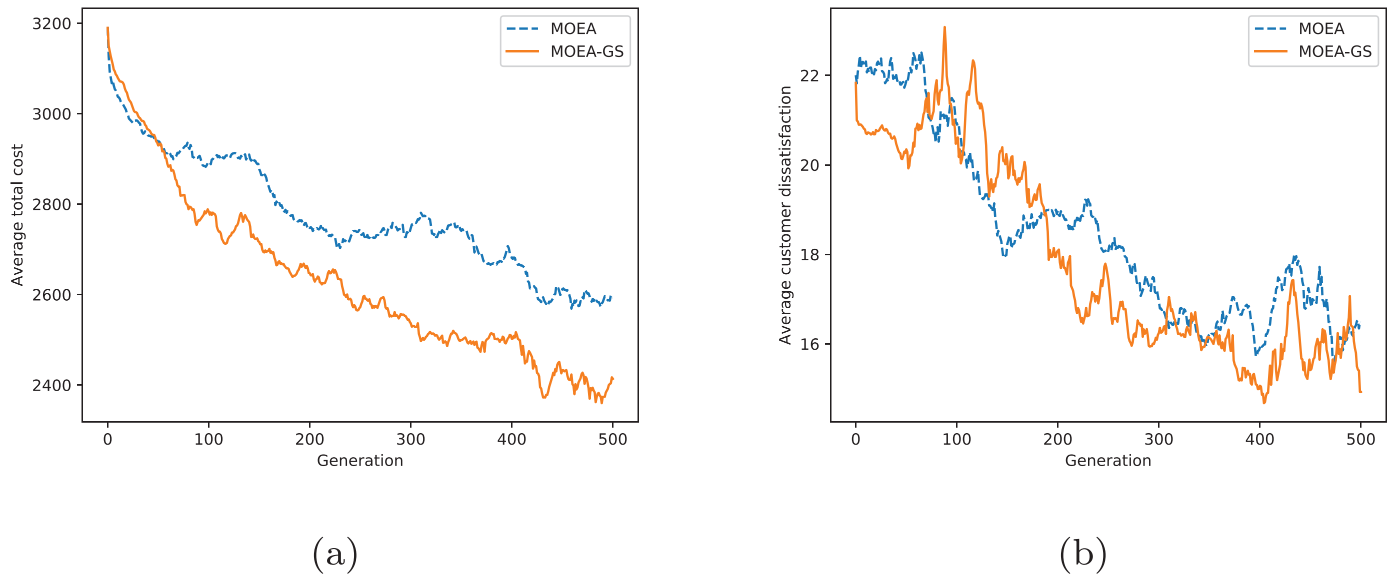


Fig. 12. (a) Average total cost and (b) average customer dissatisfaction of archive populations for MOEA-GS and MOEA.

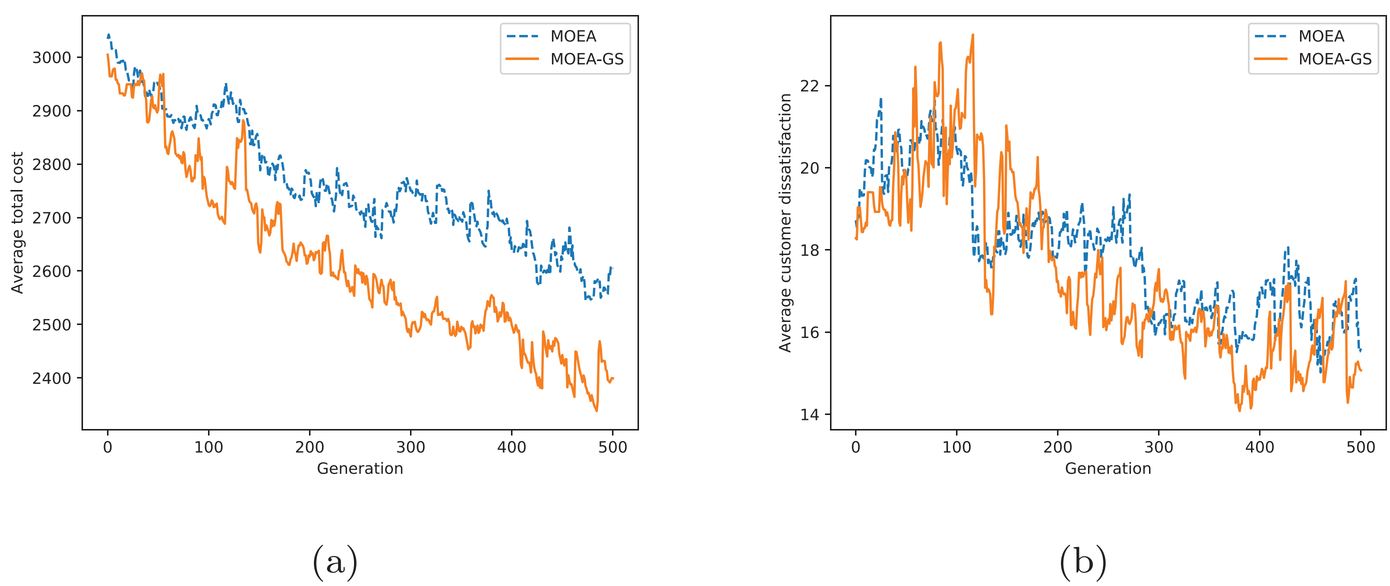


Fig. 13. (a) Average total cost and (b) average customer dissatisfaction of non-dominated solutions for MOEA-GS and MOEA.

ify this effectiveness, MOEA is also compared with the multi-objective evolutionary algorithm with guiding strategy, which is referred to as MOEA-GS. And the sets of Pareto optimal solutions achieved by MOEA and MOEA-GS are plotted in Fig. 11, respectively. It is easy to observe that MOEA-GS is more efficient in satisfying the two conflict objectives than that of MOEA.

Fig. 12 shows the convergence of total cost and customer dissatisfaction of average solutions for the archive population. Fig. 13 shows the convergence of total cost and customer dissatisfaction of average non-dominated solutions. It can be observed that MOEA-GS has a significant superiority in finding solutions with smaller total cost and customer dissatisfaction compared with MOEA, and this advantage is particularly obvious in optimizing the objective of total cost.

4.3. Performance of MIMOA

In this section, the overall performance of MIMOA is evaluated based on three objectives, i.e., Min *TC* (total cost), Min *CD* (customer dissatisfaction) and Min *MP* (multi-objective performance), respectively. Multiplicative aggregation [47] for total cost and customer dissatisfaction in non-dominated solutions is adopted to evaluate the multi-objective performance. Meanwhile, two more multi-objective optimization algorithms, i.e., NSGAII [50] and SMG-MOMA [42] are implemented as baselines to compare with MIMOA, the results of which are recorded in Table 3.

From Table 3, it can be obviously found that MIMOA performs better than NSGAII and SMG-MOMA for all ten instances in terms of optimizing the Min *TC*, Min *CD* and Min *MP*. Moreover, Fig. 14 shows the detailed results of the three algorithms for the ten instances, which demonstrates the significant advantage of MIMOA in optimizing the two objectives over NSGAII and SMG-MOMA.

The average computation time (in seconds) for the three algorithms is recorded in Fig. 15. In specific, the population size is 300; each instance contains 120 customers; and the maximum number of iterations is 500. Note that all previous results are obtained with 500 as the maximum number of iterations. From Fig. 15, it is easy to see that, the computation time for MIMOA is only slightly longer than the other two for several instances. Given the significant superiority of MIMOA in reduc-

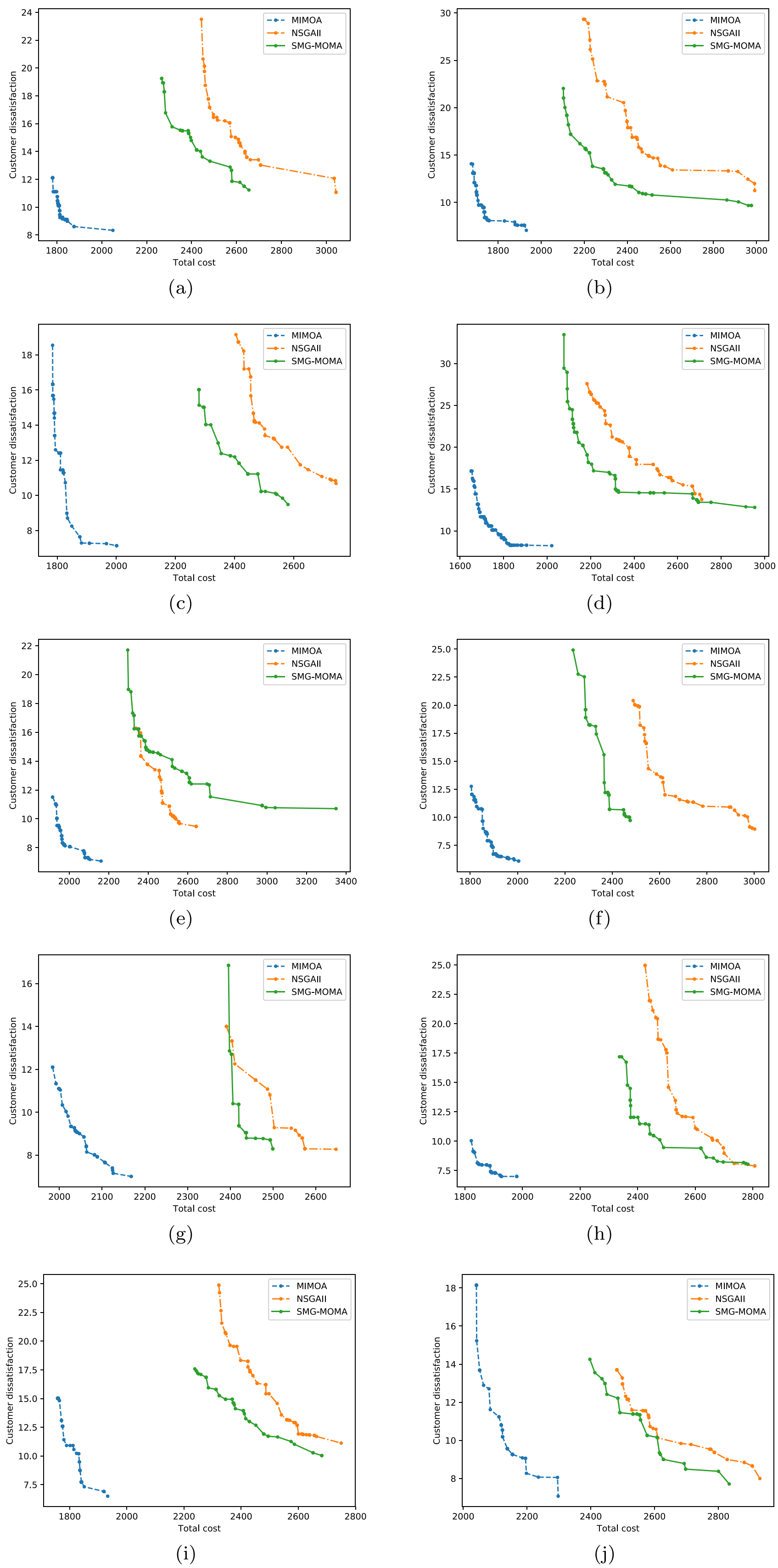


Fig. 14. The non-dominated solution set of MIMOA, NSGAII and SMG-MOMA for ten instances.

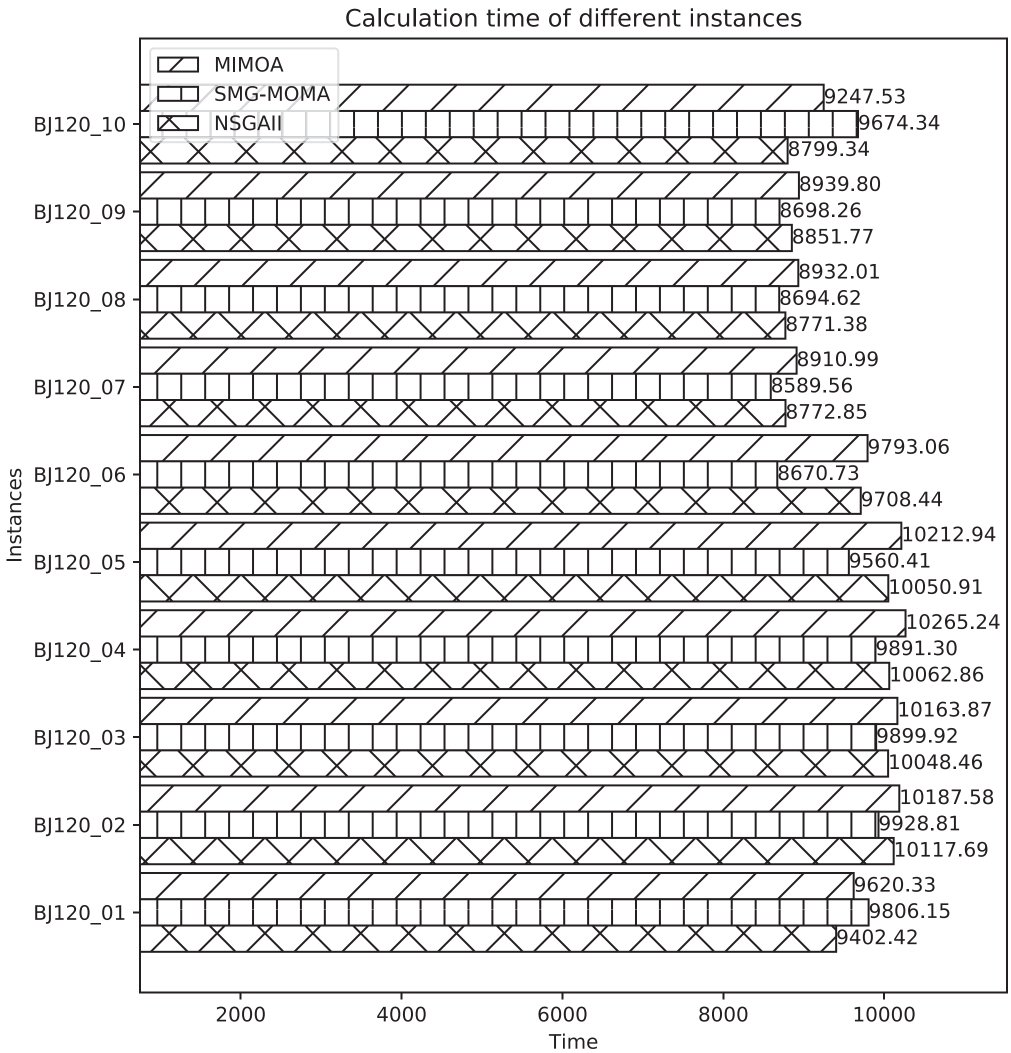


Fig. 15. Calculation time of different instances.

ing objective values, the slightly longer computation time is acceptable. On the other hand, it also shows that the clustering strategy and the guidance strategy did not consume much additional time.

5. Conclusions

This paper studies a mathematical model of bi-objective VRPSD (BO-VRPSD), which considers two conflicting objectives, i.e., total cost and customer satisfaction. Generally, BO-VRPSD aims to achieve desirable tradeoff between business interests and customer satisfaction for route planning.

To address BO-VRPSD, a membrane-inspired multi-objective algorithm(MIMOA) was proposed, which has a tissue-like framework including three subsystems, i.e., one control subsystem and two operation subsystems. In MIMOA, an improved MOEA with clustering strategy was designed for the operation subsystems; communications between control subsystem and operation subsystems were implemented based on a guiding strategy. Experimental results based on ten 120-node instances with real geographic geographic locations in Beijing show that, membrane-inspired framework together with the clustering strategy and guiding strategy can significantly improve the performance over other classical multi-objective evolutionary algorithms. In future, (1) measures on further improving the efficiency or execution speed of MIMOA will be investigated, by leveraging the models and algorithms in [48,49], (2) impacts of different clustering strategies and cluster numbers to MIMOA will be studied.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yunyun Niu: Conceptualization, Methodology. **Yongpeng Zhang:** Software, Writing - original draft. **Zhiguang Cao:** Conceptualization, Writing - review & editing. **Kaizhou Gao:** Conceptualization. **Jianhua Xiao:** Conceptualization, Methodology, Supervision. **Wen Song:** Software, Visualization, Writing - review & editing. **Fangwei Zhang:** Methodology.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61872325, 61772290, 61803104); the Fundamental Research Funds for the Central Universities (2652019028, 63192616); the Science and Technology Development Strategy Research Program of Tianjin (18ZLZXZF00320); the China Scholarship Council [grant number 201806405004]; the Singapore National Research Foundation (NRF-RSS2016004); and the Collaborative Innovation Center for China Economy.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.swevo.2020.100767.

References

[1] D.J. Bertsimas, A vehicle routing problem with stochastic demand, Oper. Res. 40 (3) (1992) 574–585.
[2] M. Savelsbergh, M. Goetschalckx, A comparison of the efficiency of fixed versus variable vehicle routes, J. Bus. Logist. 16 (1995) 163–187.
[3] V. Pillac, M. Gendreau, C. Guéret, A.L. Medaglia, A review of dynamic vehicle routing problems, Eur. J. Oper. Res. 225 (1) (2013) 1–11.
[4] J. Cordeau, G. Laporte, M. Savelsbergh, D. Vigo, Vehicle routing, in: C. Barnhart, G. Laporte (Eds.), Handbooks in Operations Research and Management Science: Transportation, 14, 2006, pp. 367–428.
[5] Z. Cao, H. Guo, J. Zhang, D. Niyato, U. Fastenrath, Finding the shortest path in stochastic vehicle routing: a cardinality minimization approach, IEEE Trans. Intell. Transp. Syst. 17 (6) (2016) 1688–1702.
[6] Z. Cao, H. Guo, J. Zhang, D. Niyato, U. Fastenrath, Improving the efficiency of stochastic vehicle routing: a partial lagrange multiplier method, IEEE Trans. Veh. Technol. 65 (6) (2016) 3993–4005.
[7] Y. Marinakis, M. Marinaki, A. Migdalas, A hybrid clonal selection algorithm for the vehicle routing problem with stochastic demands, in: Learning and Intelligent Optimization, Springer International Publishing, 2014, pp. 258–273.
[8] K. Wang, S. Lan, Y. Zhao, A genetic-algorithm-based approach to the two-echelon capacitated vehicle routing problem with stochastic demands in logistics service, J. Oper. Res. Soc. 68 (11) (2017) 1409–1421.
[9] Z. Luo, H. Qin, D. Zhang, A. Lim, Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost, Transp. Res. Pt. e-Logist. Transp. Rev. 85 (2016) 69–89.
[10] M. Marinaki, Y. Marinakis, A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands, Expert Syst. Appl. 46 (2016) 145–163.
[11] J.E. Mendoza, L.M. Rousseau, J.G. Villegas, A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints, J. Heurist. 22 (2016) 539–566.
[12] S.B. Gee, W.A. Arokiasami, J. Jiang, Decomposition-based multi-objective evolutionary algorithm for vehicle routing problem with stochastic demands, Soft Comput. 20 (9) (2016) 3443–3453.
[13] S. Iqbal, M. Kaykobad, M.S. Rahman, Solving the multi-objective vehicle routing problem with soft time windows with the help of bees, Swarm Evol. Comput. 24 (2015) 50–64.
[14] Z. He, P. Chen, H. Liu, Z. Guo, Performance measurement system and strategies for developing low-carbon logistics: a case study in china, J. Clean Prod. 156 (10) (2017) 395–405.
[15] Y. Niu, Z. Yang, P. Chen, J. Xiao, Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost, J. Clean Prod. 171 (2018) 962–971.
[16] Y. Xiao, A. Konak, A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem, J. Clean Prod. 167 (2017) 1450–1463.
[17] M. Barth, K. Boriboonsomsin, Real-world CO_2 impacts of traffic congestion, Transp. Res. Rect. 2058 (1) (2008) 163–171.
[18] T. Bektaş, G. Laporte, The pollution-routing problem, Transp. Res. Pt. B-Methodol. 45 (8) (2011) 1232–1250.
[19] Z. Cao, S. Jiang, J. Zhang, H. Guo, A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion, IEEE Trans. Intell. Transp. Syst. 18 (7) (2017) 1958–1973.
[20] E. Demir, T. Bektaş, G. Laporte, A review of recent research on green road freight transportation, Eur. J. Oper. Res. 237 (2014) 775–793.
[21] E.M. Toro, A.H. Escobar, M. Granda, Literature review on the vehicle routing problem in the green transportation context, Luna Azul. 42 (2016) 362–387.
[22] M. Afshar-Bakeshloo, A. Mehrabi, H. Safari, M. Maleki, F. Jolai, A green vehicle routing problem with customer satisfaction criteria, J. Ind. Eng. Int. 12 (2016) 529–544.
[23] T.Y. Nishida, Membrane algorithm: an approximate algorithm for NP-complete optimization problems exploiting p systems, in: Proceedings of the Sixth International Workshop on Membrane Computing, 2005, pp. 26–43.
[24] G. Zhang, M. Gheorghe, L. Pan, M.J. Pérez-Jiménez, Evolutionary membrane computing: a comprehensive survey and new results, Inf. Sci. 279 (2014) 528–551.
[25] Y. Niu, J. Xiao, Y. Jiang, Solving 3-coloring problem with time-free tissue p systems, Chin. J. Electron. 25 (3) (2016) 407–412.
[26] C. Liu, L. Fan, A hybrid evolutionary algorithm based on tissue membrane systems and CMA-ES for solving numerical optimization problems, Knowledge-Based Syst. 105 (2016) 38–47.
[27] G. Păun, G. Rozenberg, A. Salomaa (Eds.), Handbook of Membrane Computing, Oxford University Press, 2009.
[28] G. Zhang, J. Chen, M. Gheorghe, Q. Meng, A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, Appl. Soft. Comput. 13 (3) (2013) 1528–1542.
[29] G. Zhang, M.J. Pérez-Jiménez, M. Gheorghe, Real-life Applications with Membrane Computing, Springer, Berlin, 2017.
[30] J. Xiao, Y. Huang, C. Zhen, J. He, Y. Niu, A hybrid membrane evolutionary algorithm for solving constrained optimization problems, Optik 125 (2014) 897–902.
[31] Y. Niu, S. Wang, J. He, J. Xiao, A novel membrane algorithm for capacitated vehicle routing problem, Soft. Comput. 19 (2015) 471–482.
[32] L. Huang, X. He, N. Wang, Y. Xie, P systems based multiobjective optimization algorithm, Prog. Nat. Sci. 17 (4) (2007) 458–465.
[33] H. Peng, P. Shi, J. Wang, A. Riscos-Núñez, M.J. Pérez-Jiménez, Multiobjective fuzzy clustering approach based on tissue-like membrane systems, Knowledge-Based Syst. 125 (2017) 74–82.
[34] Y. Ju, S. Zhang, N. Ding, X. Zeng, X. Zhang, Complex network clustering by a multi--objective evolutionary algorithm based on decomposition and membrane structure, Sci. Rep. 6 (33870) (2016) 1–13.
[35] Q.F. Zhang, H. Li, MOEA/d: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evolut. Comput. 11 (6) (2007) 712–731.
[36] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems, Eur. J. Oper. Res. 172 (3) (2006) 855–885.
[37] A. Konstantinidis, K. Yang, Q. Zhang, A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks, Comput. Netw. 54 (6) (2010) 960–976.
[38] G. Zhang, Y. Li, M. Gheorghe, A multi-objective membrane algorithm for knapsack problems, in: Proceedings of the Forth International Conference on Bio-inspired Computing: Theories & Applications, IEEE, 2010, pp. 604–609.
[39] D. Zaharie, G. Ciobanu, Distributed evolutionary algorithms inspired by membranes in solving continuous optimization problems, in: Proceedings of the Seventh International Workshop on Membrane Computing, 2006, pp. 536–553.
[40] J. Cheng, G. Zhang, T. Wang, A membrane-inspired evolutionary algorithm based on population p systems and differential evolution for multi-objective optimization, J. Comput. Theor. Nanos. 12 (7) (2015) 1150–1160.
[41] C. Liu, M. Han, X. Wang, A multi-objective evolutionary algorithm based on membrane systems, in: Proceedings of the Fourth International Workshop on Advanced Computational Intelligence, IEEE, 2011, pp. 103–109.
[42] X. Zhang, J. Li, L. Zhang, A multi-objective membrane algorithm guided by skin membrane, Nat. Comput. 15 (4) (2016) 597–610.
[43] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm, Technical report, Computer Engineering and Networks Laboratory (TIK), 2001.
[44] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems, Eur. J. Oper. Res. 172 (3) (2006) 855–885.
[45] K.C. Tan, C.K. Cheong, C.K. Goh, Solving multi-objective vehicle routing problem with stochastic demand via evolutionary computation, Eur. J. Oper. Res. 177 (2) (2007) 813–839.
[46] M. Dror, P. Trudeau, Stochastic vehicle routing with modified savings algorithm, Eur. J. Oper. Res. 23 (2) (1986) 228–235.
[47] D.A.V. Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithm research: a history and analysis, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio, 1998 Technical report tr-98-03.
[48] J. Xu, Probe machine, IEEE Trans. Neural Netw. Learn. Syst. 27 (7) (2016) 1405–1416.
[49] X. Zhang, Y. Tian, Y. Jin, A knee point driven evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (6) (2015) 761–776.
[50] K. Deb, S. Agrawal, A. Pratap, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, Evol. Comput. 1917 (2000) 849–858.