Applied Soft Computing 19 (2014) 343–361

Contents lists available at ScienceDirect

Applied Soft Computing

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

An Improved Decomposition-Based Memetic Algorithm for Multi-Objective Capacitated Arc Routing Problem

Ronghua Shang*, Jia Wang, Licheng Jiao, Yuying Wang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 18 December 2012
Received in revised form 8 July 2013
Accepted 10 March 2014
Available online 19 March 2014

Keywords:
Capacitated Arc Routing Problem
Coevolutionary
Multi-objective optimization
D-MAENS

ABSTRACT

Capacitated Arc Routing Problem (CARP) has attracted the attention of many researchers during the last few years, because it has a wide application in the real world. Recently, a Decomposition-Based Memetic Algorithm for Multi-Objective CARP (D-MAENS) has been demonstrated to be a competitive approach. However, the replacement mechanism and the assignment mechanism of the offspring in D-MAENS remain to be improved. First, the replacement after all the offspring are generated decreases the convergence speed of D-MAENS. Second, the representatives of these sub-problems are reassigned at each generation by only considering one objective function. In response to these issues, this paper presents an improved D-MAENS for Multi-Objective CARP (ID-MAENS). The two improvements of the proposed algorithm are as follows: (1) the replacement of the solutions is immediately done once an offspring is generated, which references to the steady-state evolutionary algorithm. The new offspring will accelerate the convergence speed; (2) elitism is implemented by using an archive to maintain the current best solution in its decomposition direction during the search, and these elite solutions can provide helpful information for solving their neighbor sub-problems by cooperation. Compared with the Multi-Objective CARP algorithm, experimental results on large-scale benchmark instances *egl* show that the proposed algorithm has performed significantly better than D-MAENS on 23 out of the total 24 instances. Moreover, ID-MAENS find all the best nondominated solutions on 13 *egl* instances. In the last section of this paper, the ID-MAENS also proves to be competitive to some state-of-art single-objective CARP algorithms in terms of quality of solutions and computational efficiency.

1. Introduction

Both Arc Routing Problem (ARP) and Vehicle Routing Problem (VRP) are classical combinatorial optimization problems. VRP is to design routes for a fleet of vehicles, which are to serve a set of geographically dispersed points (customers, stores, schools, cities, warehouses, etc.) at the least cost [1]. While ARP is to design routes for a fleet of vehicles, which are to serve a set of task arcs (salting route, mail delivering route, street sweeping route, school bus scheduling route, etc.) at the least cost [2]. Both ARP and VRP have many extensive models [3,4]. In this paper, a special kind of ARP called Capacitated ARP (CARP) is discussed. Considering an important constraint called capacity constraint, this new model has a more practical application [5,6]. CARP is defined on an undirected and connected graph representing the road network, where each edge has a travel cost and a non-negative demand. A number of vehicles are based at the depot with limited capacity [2]. The aim of CARP is to seek a set of minimum cost trips for the vehicles to serve all the positive-demand edges on the conditions: (1) the total demand served by any trip cannot exceed the vehicle's capacity; and (2) each vehicle must start and end at the depot.

For many applications in reality, there is a wide gap between the classical CARP model and the real world situations. Hence, researchers pay more attention to other extended versions of CARP [7–11]. These models are more complex and pose greater challenges for researchers. In 2006, Lacomme firstly proposed a Multi-Objective CARP (MO-CARP) model which not only minimizes the *total-cost* but also balances these trips [11]. That is minimizing the *total-cost* and *makespan* (i.e. the cost of the longest trip) simultaneously. These two objectives considered by Lacomme are conflicted with each other. Thus, no unique global optimal solution exists in this case. Closed to the actual application, this model has attracted the attention of many researchers [12]. Above all, Lacomme proposed a Multi-Objective Genetic Algorithm (LMOGA) to solve MO-CARP [11], which uses the fast nondominated

* Corresponding author. Tel.: +86 02988202279.
E-mail address: rhshang@mail.xidian.edu.cn (R. Shang).

http://dx.doi.org/10.1016/j.asoc.2014.03.005
1568-4946/

sorting and the crowding distance approach of a commonly Multi-Objective Evolutionary Algorithm (MOEA), namely Non-dominated Sorting Genetic Algorithm II (NSGA-II) [13]. Through maintaining a set of solutions which are good "tradeoffs" between the two objectives, this problem is solved as a whole in LMOGA. Furthermore, a comparison is made between LMOGA and LMA (an approach for SO-CARP) [14]. Recently, a new Memetic Algorithm (MA) called Decomposition-Based MA with Extended Neighborhood Search (D-MAENS) was proposed by Yi Mei [2] et al. D-MAENS adopts a decomposition-based framework which is similar to that of MOEA/D [15], and D-MAENS adopts the MAENS approach for SO-CARP [16]. Through following the advanced features of evolution strategy which is NSGA-II [13], D-MAENS shows a superior performance than LMOGA [2]. Experimental studies on three well-known benchmark sets (*gdb*, *val*, *egl*) also demonstrate that D-MAENS is a competitive approach.

However, the replacement mechanism and the assignment mechanism of the offspring in D-MAENS remain to be improved. First of all, the solutions' replacement is done when all the offspring are generated, and it is a one-time replacement. In this way, changing the order of solving the sub-problems does not impact the algorithm, while it will decrease the convergence speed of D-MAENS. Second, in D-MAENS, the representatives are reassigned to sub-problems at each generation by only considering one objective function. In this mechanism, it can make better use of the information of the current population during the search process, while the representative solution of each sub-problem may not be a better solution in its decomposition direction. In response to these issues, this paper presents an Improved Decomposition-Based Memetic Algorithm for Multi-Objective Capacitated Arc Routing Problem (ID-MAENS). ID-MAENS makes two improvements over the existing D-MAENS. One is the use of a steady-state evolutionary algorithm (SSEA) [17–20], and the other is the introduction of an elitism archive [13,21] for the best-so-far solutions. SSEAs are overlapping systems, because parents and offspring compete for survival. Each new offspring will either replace an existing population member, or it will die, depending on the selection pressure [22]. Some representatives of SSEA may be summarized as follows: the reducing genetic drift in SSEA by Branke [22], the median-selection for parallel SSEA by Wakunda [23] and a simple evolutionary algorithm for multi-objective optimization (SEAMO) by Valenzuela [24]. The elitism archiving mechanism is an evolutionary strategy commonly used in some EAs. By retaining the best individual in the current population, the elitism archiving mechanism can accelerate the convergence of the algorithm [13,21]. This shows that these high fitness individuals (elite individuals) play an important role in the evolution of the population. The followings are the specific improvements of the improved algorithm: (1) the replacement is immediately done once an offspring solution is generated, which can be regarded as an online mode, while that of D-MAENS is more similar to a batch mode; (2) elitism is implemented by using an archive to maintain the current best solution in its decomposition direction during the search, and these elite solutions can provide helpful information for solving their neighbor sub-problems by cooperation.

The rest of the paper is organized as follows. The related works are introduced in Section 2, which includes the literature review, the similarities and differences in the problem formulation between MO-CARP and classical CARP and the main frame of D-MAENS. In Section 3, the ID-MAENS is proposed. Afterwards, some experimental studies and a comparative analysis are shown in Section 4. Finally, the conclusions and future work are described in Section 5.

2. Related works

2.1. Literature review

Both CARP and ARP are NP-hard problems, which have been proven by Golden and Wong [25]. Because exact algorithms are only available for very small instances (the branch-and-bound approach [26] can deal with the instances of 20–30 edges), large-scale instances must be solved in practice with heuristics in early stage. To name a few: the Augment-Merge [25] proposed by Golden et al. in 1981, the Path-Scanning [27] proposed by Golden et al. in 1983, the Ulusoy's splitting technique [28] proposed by Ulusoy in 1985, the Construct-and-Strike proposed by Pearn in 1989 [29], the Cycle-Assignment [30] proposed by Benavent et al. in 1900 and the Augment-Insert proposed by Pearn in 1901 [31]. The advantage of heuristics is that it has a low computational cost, and the disadvantage is when applying it to large-scale instances the results may be unsatisfactory. In 21st century, the researchers mainly focus on designing ideal meta-heuristic for CARP. Meta-heuristic can be regarded as a more advanced heuristic algorithm than traditional heuristic algorithms. Among these meta-heuristic methods, Eglese firstly used Tabu Search to solve the CARP problem, which was used to solve the problem of sprinkling salt on a road in 1994 [32]. After that available meta-heuristics include the CARPET proposed by Hertz et al. in 2000 [33], a guided local search (GLS) method proposed by Beullens et al. in 2003 [34], a tabu scatter search meta-heuristic for the Arc Routing Problem by Greistorfer in 2003 [35], an MA for CARP (LMA) proposed by Lacomme et al. in 2004 which combines the Genetic Algorithm (GA) with lots of local search methods [14], an EA for the salting route optimization proposed by Handa et al. in 2006 [36,37], a deterministic Tabu Search algorithm for CARP by Brandão and Eglese in 2008 [38], a global repair operator (GRO) for CARP by Yi Mei et al. in 2009 [39], an MA with Extended Neighborhood Search (MAENS) for CARP proposed by Tang et al. in 2009 [16] and GRASP with evolutionary path-relinking for CARP proposed by Usberti et al. in 2011 [40]. All these meta-heuristic are excellent strategies for solving CARP, and they also have gained a lot of success in practical applications.

Compared with classical CARP, other extended versions of CARP have raised a growing interest during the last two decades because of their more important applications. The extended versions of CARP introduce further complexities to an already complex problem by adding additional constraints. Lacomme firstly proposed a periodic CARP (PCARP) in reference [41]. PCARP has a multi-period horizon and each task has a service frequency over the horizon. Besides, the number of vehicles (investment cost) in PCARP is prior to the *total-cost*. Some representative achievements of the PCARP can be summarized as follows: the first MA for PCARP by Lacomme in 2005 [41], a Scatter Search algorithm by Chu et al. in 2006 [42] and an MA with Route-Merging by Mei et al. in 2011 [7]. CARPs with stochastic demands are studied in reference [43]. Mei et al. conducted a detailed study of the CARP in uncertain environments (UCARP) [11], in which the demand of each task and the cost between two vertices are unknown. The goal of UCARP is to find the solution with the strongest robustness. The CARP with multi-depot (MCARP) can be found in reference [44]. In MCARP, the number and position of depots should be decided in the first step. Then all the tasks are assigned to these depots to construct a feasible solution. In 2010, An Evolutionary Approach to the MCARP is proposed by Xing et al. [9]. This paper proposed a novel EA called HHEA that employs classical heuristics as well as heuristic information to deal with the MCARP and got significantly better results than other heuristics. Lacomme et al. proposed the ECARP in 2004 [14]. The authors considered the following four extensions: (1) mixed multi-graph with different links; (2) two distinct costs per link; (3) turn penalties; and (4) maximum route length. The classic algorithms of ECARP are as follows: an improved evolutionary approach by Xu et al. [45], an improved GA [46] by Mei et al. and a Hybrid Ant Colony Optimization Algorithm [8] by Xing et al.

Although the above various versions of CARP have different constraints, they all be predominantly formulated as a single-objective problem with the only objective of minimizing the *total-cost*. In real life, most of the optimization problems are commonly multi-objective optimization problems (MOPs). In the past few years, many researchers were interested in the field of multi-objective numerical optimization, while how to make the best use of MOEAs in the context of MO-CARP has not been fully investigated. This brief review suggests that MO-CARP is more practical. It is thus necessary to develop novel approaches to tackle MO-CARP and this motivates the work presented in this paper.

2.2. The similarities and differences between MO-CARP and classical CARP

The above sections of this paper concisely described CARP and MO-CARP. In this section, we will describe each problem in detail and discuss the similarities and differences in the problem formulation between CARP and MO-CARP. At last, the mathematical description of MO-CARP is given.

The classical CARP is characterized as follows [8,9]:

(1) Inputs

a) An undirected and connected graph is $G=(V, E)$, with the vertex set V and the edge set E.
b) For each edge u of E, it incurs a positive traveling cost $w(u)$ and a non-negative demand $q(u)$.
c) Only one depot is available.
d) m vehicles with the capacity Q are based at the depot.

(2) Outputs

a) A vehicle arrangement scheme $\boldsymbol{x}=(T_1, \ldots, T_k, \ldots, T_m)$, T_k represents the k-service trip.

(3) Objectives

$$\min \quad f(x)=\sum_{k=1}^{m} cost(T_k)$$

(4) Constraints

a) The total load of any vehicle cannot exceed the vehicle's capacity Q.
b) All required edges in graph G must be served by exactly one vehicle.
c) The vehicle trip starts and ends at the depot.
d) It does not exist the edge whose demand is greater than Q.

The MO-CARP, an extended version of CARP, has the following attributes [2,10]:

(1) Inputs

a) A mixed graph is $G=(V, E, A)$, with the vertex set V, the arc set A and the edge set E.
b) For each arc t of A, it incurs a positive traveling cost $w(t)$ and a non-negative demand $q(t)$. Each arc has a specific direction.
c) For each edge u of E, it incurs a positive traveling cost $w(u)$ and a non-negative demand $q(u)$. Each edge is regarded as a couple of arcs, one for each direction.
d) Only one depot is available.
e) m vehicles with the capacity Q are based at the depot.

(2) Outputs

a) A set of nondominated vehicle arrangement schemes $\boldsymbol{X}=\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}=(T_1, \ldots, T_k, \ldots, T_m)$, T_k represents the k-service trip.

(3) Objectives

$$\min \quad f_1(x)=\sum_{k=1}^{m} cost(T_k)$$
$$\min \quad f_2(x)=\max_{1\le k\le m}(cost(T_k))$$

(4) Constraints

a) The total load of any vehicle cannot exceed the vehicle's capacity Q.
b) All required edges in graph G must be served by exactly one vehicle.
c) The vehicle trip starts and ends at the depot.
d) It does not exist the edge or arc whose demand is greater than Q.

The mathematical model of MO-CARP can be attributed to the following mathematical model:

$$\begin{cases} \min \quad f_1(x)=\sum_{k=1}^{m} cost(T_k) \\ \min \quad f_2(x)=\max_{1\le k\le m}(cost(T_k)) \\ where \quad \sum_{k=1}^{m}(|T_k|)=|S|; \\ R_i=\{T_{ij}|T_{ij}\in(T_{i1},\ldots,T_{i|T_i|}), \quad i=1,2,\ldots,m\}; \\ R_i\cap R_j=\Phi(i\ne j; \quad i,j=1,2,\ldots,m); \\ D(T_k)\le Q, \quad \forall 1\le k\le m; \end{cases} \tag{1}$$

In each of the above equations, T_k represents the k-service trip in graph G. $cost(T_k)$ is the *total-cost* of the k-service trip. $\max(cost(T_k))$ is the cost of the longest trip. $|T_k|$ is the total number of the service tasks contained in the T_k trips. m is the total number of vehicles and S is the set of all the arc tasks and edge tasks.

2.3. The description of D-MAENS

In D-MAENS, the original two objective functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ of MO-CARP are decomposed into N SO-CARPs by a set of uniformly distributed weight vectors $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N$ (each $\boldsymbol{\lambda}_i$ is a two-dimensional vector). The objective function of the ith sub-problem corresponding to the ith weight vector is stated as [2]:

$$g_i(\boldsymbol{x})=\boldsymbol{\lambda}_{i1}\times f_1(\boldsymbol{x})+\boldsymbol{\lambda}_{i2}\times f_2(\boldsymbol{x}) \quad 1\le i\le N \tag{2}$$

where $\boldsymbol{\lambda}_i=(\boldsymbol{\lambda}_{i1}, \boldsymbol{\lambda}_{i2})$. In order to solve the N SO-CARPs respectively, D-MAENS maintains a set $\boldsymbol{X}=\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$throughout the optimization process, which stands for the population. Before each iteration, firstly, a 1-1 mapping is made between the collection of sub-problems and the population $\boldsymbol{X}$ according to certain rules. Specifically, each sub-problem is assigned a unique solution, which is called its representative. For example, $\boldsymbol{x}_i$ is the representative of g_i. Secondly, the sub-population of a sub-problem is composed of the representatives of T sub-problems (including its own), where T is the size of sub-population. For the sub-problem generated by the weight vector $\boldsymbol{\lambda}_i$, its sub-population is constituted by the representatives of T sub-problems whose weight vectors are the T closest in term of Euclidean distance. Next, these problems are solved respectively. For the ith sub-problem, we select two parents from the sub-population associated with the ith sub-problem, and then the operations of crossover and local search of MAENS are applied to the parents to generate an offspring $\boldsymbol{y}$. After generating all the offspring $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ for N sub-problems, the solutions in both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are combined together and then sorted by the fast nondominated sorting procedure and the crowding distance method in NSGA-II. At last, N solutions in the front row are selected to constitute the next generation group.

On the one hand, In D-MAENS, before each iteration, a one-to-one mapping is established between $\boldsymbol{X}$ and N sub-problems $g_1, \ldots, g_N$. In the process of solving each sub-problem separately, this mapping remains unchanged. The new offspring are only reserved and not involved in solving other sub-problems. When all offspring are generated, N new solutions are chosen to constitute the next generation of group according to certain rules. Then a one-to-one mapping is built and the algorithm enters the next generation. In this mode, changing the order of solving the sub-problems will make no difference in the framework of D-MAENS, but it reduces the speed of convergence of the algorithm. If the replacement of solutions is immediately done once an offspring is generated at each generation, the newly generated solutions can be shared by their adjacent solutions.

On the other hand, in D-MAENS, a one-to-one mapping is established between $\boldsymbol{X}$ and N sub-problems $g_1, \ldots, g_N$. Due to the N weight vectors are defined as:

$$\boldsymbol{\lambda}_i = \left(\frac{i-1}{N-1}, 1 - \frac{i-1}{N-1}\right) \quad (3)$$

As i increases, the importance in the aggregated objective function $g_i(\boldsymbol{x})$ decreases on f_1 and increases on f_2. The solution allocation mechanism of D-MAENS is simply described as: it sorts the population $\boldsymbol{X}$ based on the second objective function. Then, the ith solution in the sorted population is assigned to $g_i(\boldsymbol{x})$ [2]. It is very critical to allocate suitable representative solutions for sub-problems. We find it not so reasonable to assign solutions only in accordance with the size of a target function. In this case, it would affect the subsequent operation, which is to exchange information between the representatives of the sub-problems. Thus it will reduce the efficiency of the co-evolution between individuals. So, we need to re-establish a new solution allocation scheme which considers the global information at the same time meets the model of decomposition.

3. The improved D-MAENS

ID-MAENS uses the algorithm framework of D-MAENS and improves the algorithm's two aspects which are the replacement mechanism of the offspring and the sub-populations partition mechanism. First, in ID-MAENS, the solution replacement is immediately done once an offspring is generated at each generation and this will speed up the algorithm's convergence. Second, in ID-MAENS, an elitist strategy is used to retain the best-so-far solutions in each decomposed sub-problem according to the direction vector of the solution during the search process. In solving the ith sub-problem, the current best solutions of the ith, the $(i-1)$th, the $(i+1)$th sub-problems and the original ith sub-population of D-MAENS are combined together to constitute a new sub-population, which immediately participates in the subsequent process of evolution. The improved algorithm has two advantages: (1) it is more in line with the philosophy of using problem decomposition theory for solving MO-CARP; (2) it pays more attention on the co-evolution between sub-populations. These two advantages will speed up the convergence of the algorithm. Next, the improvements will be analyzed specifically.

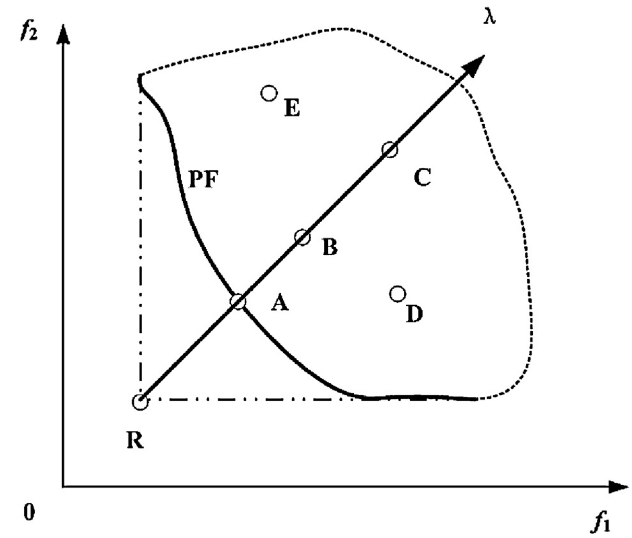

Fig. 1. The understanding about MOP based on problem decomposition.

3.1. The offspring replacement mechanism in ID-MAENS

In the process of using the theory based on problem decomposition to solve MOPs, the original MO-CARP is decomposed into a number of scalar sub-problems using the weighted sum approach with a set of uniformly distributed weight vectors. By assigning different weight coefficients to different objective functions, the algorithm can conduct a local search in different regions of the objective space [15]. However, the direction vector is also a standard to divide the objective space. So, in this point of view, we can establish some mapping relationships between the weight vectors and the direction vectors [47,48].

In the minimization problem shown in Fig. 1, points $\boldsymbol{A}$–$\boldsymbol{E}$ belong to the objective space, and point $\boldsymbol{R} = (\min(f_1(\boldsymbol{x})), \min(f_2(\boldsymbol{x})))$ is the reference point. The direction vectors of all points within the population have $\boldsymbol{R}$ as a reference. Points $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are in the same direction which correspond to weight vector $\boldsymbol{\lambda}$, and among them point $\boldsymbol{A}$ is on the $\boldsymbol{PF}$. For every weight vector through $\boldsymbol{R}$, there is always one point which is the closest to $\boldsymbol{R}$ in the same direction, and these points form a set Ψ. Obviously the nondominated set in Ψ is the Pareto-Optimal Front [47]. Of course, for the maximization problems, the reference point $\boldsymbol{R}$ can be selected to $(\max(f_1(\boldsymbol{x})), \max(f_2(\boldsymbol{x})))$. The examples cited in this paper are all minimization problems. Consequently, the goal of the MOP based on problem decomposition is to find the current best solution for each decomposed sub-problem. That is to find the set of points which are the closest to $\boldsymbol{R}$ along each weight vector $\boldsymbol{\lambda}$ in the objective space.

In D-MAENS, the replacement of the solutions is not to operate in a timely manner, but when all offspring generated to do a one-time replacement [2]. By this way, changing the order of solving the sub-problems do not affect the search process of the whole algorithm. However, as mentioned above, the advantages of the MOP based on problem decomposition lie in the co-evolution and the neighborhood sharing between adjacent sub-populations. Coevolutionary is an evolutionary process among multiple populations, so the changes of one population will lead to the response of others. And co-evolution completes Darwin's theory of evolution. The key is to maintain a relationship which is independent and collaborative between multiple populations [19]. For the shortcomings of D-MAENS, we make the following improvements: in ID-MAENS, the solution replacement is not called after all sub-problems have been solved, while it is immediately done once a superior solution is generated at each generation. In this case, the better solution can immediately participates in the evolutionary process and

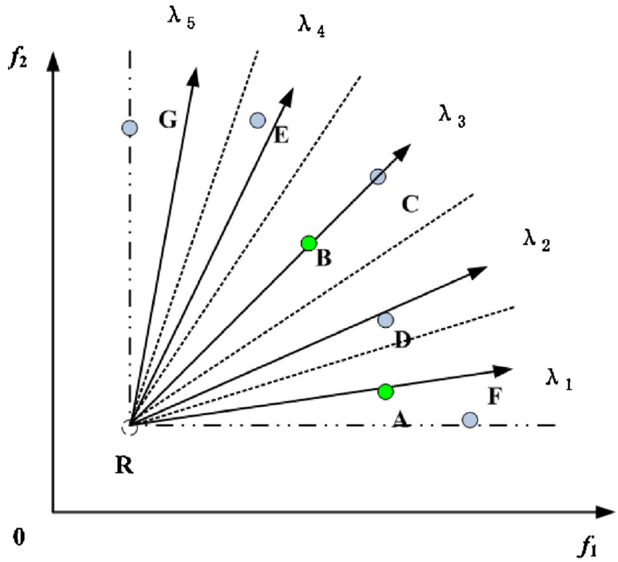

Fig. 2. The timely replacement of solutions in ID-MAENS.

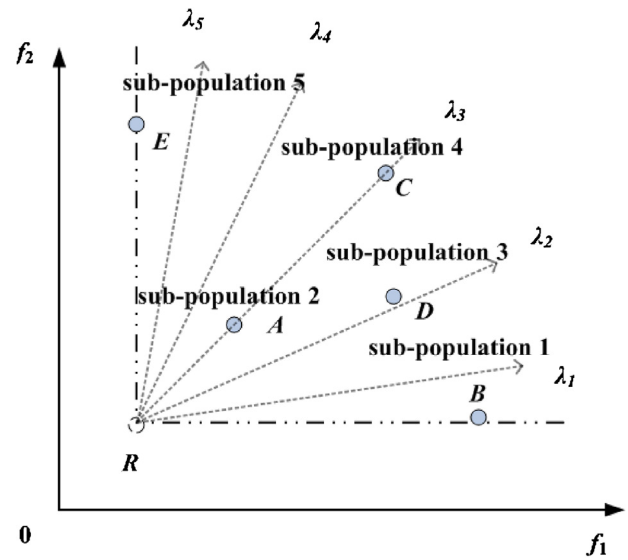

Fig. 3. The division of sub-groups in D-MAENS.

provides helpful information for solving its adjacent sub-problems. The improved algorithm not only speeds up the convergence of the algorithm but also more in line with the mechanism of co-evolution between sub-populations. The specific solution replacement process is shown in Fig. 2.

Suppose there are 5 uniformly distributed weight vectors and the MO-CARP is decomposed into 5 separate sub-problems. After that, these sub-problems are solved respectively. In Fig. 2, ***C*–*G*** are parent individuals and ***A*–*B*** are offspring individuals. Before the start of the evolution, each representative is assigned to each sub-problem respectively: $\boldsymbol{F} \to \boldsymbol{\lambda}_1$, $\boldsymbol{D} \to \boldsymbol{\lambda}_2$, $\boldsymbol{C} \to \boldsymbol{\lambda}_3$, $\boldsymbol{E} \to \boldsymbol{\lambda}_4$, $\boldsymbol{G} \to \boldsymbol{\lambda}_5$. We firstly solve the sub-problem g_1 decomposed by weight vector $\boldsymbol{\lambda}_1$ using the sub-population constituted by ***F*** and ***D***. Then an offspring solution ***A*** which is obviously better than ***F*** in term of g_1 is obtained. Next, we got on to solve the sub-problem g_2 decomposed by weight vector $\boldsymbol{\lambda}_2$.

In D-MAENS, when building a sub-population for g_2, the superior solution ***A*** is not considered and the individuals ***F***, ***D*** and ***C*** are used to constitute a sub-population. In contrast to D-MAENS, ID-MAENS pays more attention to the use of the better solution ***A***. We use the ***A***, ***D*** and ***C*** to solve g_2 and find a new solution ***B***. Obviously ***B*** is superior to ***D*** in the decomposition direction $\boldsymbol{\lambda}_2$. When solving g_3, ***B*** is still used to provide help. It is due to the timely replacement of the solutions that we find ID-MAENS can converge quickly.

3.2. The sub-populations partition mechanism in ID-MAENS

D-MAENS adopts a decomposition-based framework which is with reference to MOEA/D. They both decompose the original MO-CARP into many scalar sub-problems using the weighted sum approach with a set of uniformly distributed weight vectors. While this process can be described as follows in the objective space. The entire population is divided into different sub-populations by using a set of uniformly distributed weight vector. Adjacent sub-populations can share their neighborhoods and evolve together, so as to find the optimal solution in each direction.

However, in D-MAENS, sub-populations of the sub-problems are re-assigned at each generation [2]. Such a way makes the D-MEANS dynamically allocate the appropriate representatives to the sub-problems according to the current population information in the search process, while the representative of each sub-problem may not be a better solution in its decomposition direction. This is clearly contrary to our idea, which is based on the framework of decomposition for MOP. If the elite solution for each sub-problem is not lost, the convergence rate of the algorithm will speed up by neighbors sharing. Specifically, in D-MAENS, the allocation of solutions can briefly describe as follows. Firstly, it sorts the population ***X*** in accordance with the ascending order of the second objective function. Then, the *i*th solution in the sorted population is assigned to g_i [2]. The detailed description is shown in Fig. 3.

There are 5 evenly distributed weight vector $\boldsymbol{\lambda}_1$–$\boldsymbol{\lambda}_5$ and the population ***X*** includes five individuals ***A*–*E***. The distribution of solutions in D-MAENS is as follows: $\boldsymbol{B} \to \boldsymbol{\lambda}_1$, $\boldsymbol{A} \to \boldsymbol{\lambda}_2$, $\boldsymbol{D} \to \boldsymbol{\lambda}_3$, $\boldsymbol{C} \to \boldsymbol{\lambda}_4$, $\boldsymbol{E} \to \boldsymbol{\lambda}_5$. This distribution is clearly not the most reasonable. So we propose a new approach for the allocation of the sub-populations. The individual assigned operator in ID-MAENS is shown in Table 1.

In ID-MAENS, when a new feasible solution ***x*** is generated, we firstly use the individual assigned operator in Table 1 to find the sub-population *i* which individual ***x*** belongs to. And then we calculate the fitness of individual ***x*** under this sub-problem:

$$fitness(\boldsymbol{x}) = \lambda_{i1} \times f_1(\boldsymbol{x}) + \lambda_{i2} \times f_2(\boldsymbol{x}) \quad (4)$$

where $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are two normalized objective functions. In the above formula, normalization is referenced to D-MAENS [2]. If the fitness of ***x*** is less than the fitness of the current best solution of the *i*th sub-problem, we will replace the representative solution with ***x***. If this sub-problem is without a representative solution, we directly assign individual ***x*** as its current best solution. This method is equivalent to using an elite strategy to maintain the best-so-far solutions in its decomposition direction during the search according to the direction vector. In such a case, when solving the *i*th sub-problem, the current best solutions of the *i*th, the $(i-1)$th,

Table 1
The individual assigned operator in ID-MAENS.

The individual assigned operator
Begin
1: Calculate the slope of the direction vector of ***x*** in the objective space, $K = \frac{f_2(\boldsymbol{x}) - \min(f_2(\boldsymbol{x}))}{f_1(\boldsymbol{x}) - \min(f_1(\boldsymbol{x}))}$;
2: for($i = 1$; $i <= N$; i++)
3: Calculate the slope of the ith weight vector $k_i = \left(\frac{i-1}{N-1}\right) / \left(1 - \frac{i-1}{N-1}\right)$ and the size of $(K - k_i)^2$;
4: end for;
5: The sub-population which individual ***x*** belongs to has the minimum value of $(K - k_i)^2$;
end

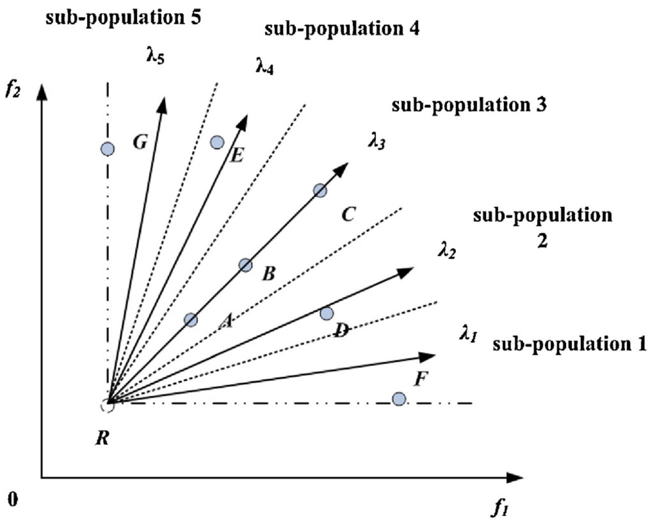

Fig. 4. The division of sub-groups in ID-MAENS.

the $(i+1)$th sub-problems and the original ith sub-population of D-MAENS are combined together to constitute a new sub-population, which immediately participates in the subsequent process of evolution. Two advantages are as follows: (1) when solving the ith sub-problem, the elite solution in the direction of this sub-problem is better used. So, the algorithm converges faster; (2) the advantage that each sub-problem can be assigned a representative according to the information of the current population during the search process in D-MAENS is kept.

In Fig. 4, we can see that 5 uniformly distributed weight vectors $\boldsymbol{\lambda}_1$–$\boldsymbol{\lambda}_5$ which are randomly generated to divide the whole population into five different zones. Each point (individual) is then assigned to different regions which represent different sub-populations, and the individual assigned operator is to determine the ownership of each individual. In Fig. 4, obviously, the slopes of the direction vectors of ***A***, ***B*** and ***C*** in the objective space are close to $\boldsymbol{\lambda}_3$. Therefore, ***A***, ***B*** and ***C*** belong to the sub-population 3. Similarly, ***D*** belongs to the sub-population 2. ***E*** belongs to the sub-population 4. ***F*** belongs to the sub-population 1. ***G*** belongs to the sub-population 5. Because both sub-population 2 and 4 are adjacent sub-populations of the sub-population 3, individuals in them can provide helpful information for solving the sub-problem decomposed by $\boldsymbol{\lambda}_3$.

3.3. The improved D-MAENS

As described above, we analyze the advantages and disadvantages of D-MAENS in the issue of solving MO-CARP and make two improvements: (1) the solution replacement of ID-MAENS is immediately done once a superior solution is generated at each generation. In this case, the better solution can immediately participate in the evolutionary process. (2) using an elite strategy to maintain the elite solutions in each decomposition direction during the search according to the direction vector, we make a new sub-population which includes the current best solutions of the ith, the $(i-1)$th, the $(i+1)$th sub-problems and the original ith sub-population of D-MAENS to participate in the process of evolution. The detailed steps of ID-MAENS are as follows (Table 2).

4. Experimental results and analysis

In order to test the validity of ID-MAENS, three different experiments are carried out. In the first experiment, a key comparison between two MO-CARP algorithms is made, which are ID-MAENS and D-MAENS. After that, three different versions of ID-MAENS are tested in order to evaluate the impact of different improvements on the algorithm's overall performance. In the last experiment, since all test instances in this paper are used for SO-CARP, it is also necessary to compare ID-MAENS with some typical SO-CARP algorithms. In 2009, Mei et al. [22] proposed a global repair operator (GRO) and inserted it to the TSA. Experimental results suggest this algorithm is competitive with a number of state-of-the-art approaches. In 2010, Xing et al. proposed a novel EA (HHEA) for MCARP. HHEA is novel in employing classical heuristics as well as heuristic information. To check whether a nondominated solution can be obtained by the above SO-CARP algorithms, we compare ID-MAENS with GRO and HHEA.

Table 2
Improved Decomposition-Based Memetic Algorithm for MO-CARP.

Improved Decomposition-Based Memetic Algorithm for MO-CARP
Begin
Initialize a population $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$and set the nondominated solutions $\boldsymbol{X}^* = \boldsymbol{\emptyset}$.
Generated N evenly distributed weight vectors $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N$ randomly. Compute the Euclidean distance between each pair of weight vectors and find T weight vectors which are closest to $\boldsymbol{\lambda}_i$, where $1 \le i \le N$.
Decompose the original MO-CARP into N SO-CARPs $g_1, \ldots, g_N$ with $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N$.
Set $it = 0$;
while $(it < G_{max})$ do
Establish a 1-1 mapping between $\boldsymbol{X}$ and $g_1, \ldots, g_N$, then assign each sub-problem a unique representative. Construct a sub-population for the ith sub-problem. The individuals in the ith sub-population are the representative solutions of the sub-problems which decomposed by T weight vectors closest to $\boldsymbol{\lambda}_i$.
for($i = 1$; $i <= N$; i++)
the current best solutions of the ith, the $(i-1)$th, the $(i+1)$th sub-problems and the original ith sub-population of D-MAENS combine together to constitute a new sub-population.
Randomly select two individuals from the new sub-population and apply the crossover and local search operators of MAENS to find the offspring $\boldsymbol{y}_i$.
Call the individual assigned operator on $\boldsymbol{y}_i$, and update the representative solution group. That is to update the representative solution of the ith sub-problem with $\boldsymbol{y}_i$.
if (no individual in $\boldsymbol{X}^*$ dominates $\boldsymbol{y}_i$) then
Insert $\boldsymbol{y}_i$ to $\boldsymbol{X}^*$.
end if.
end for.
$\boldsymbol{Z} = \boldsymbol{X} \cup \boldsymbol{Y}$, Sort the individuals in $\boldsymbol{Z}$ by the fast nondominated sorting procedure and crowding distance approach of NSGA-II, Then, let $\boldsymbol{X}$ be the top of N solutions in the sorted $\boldsymbol{Z}$.
end while.
Export $\boldsymbol{X}^*$.
end

4.1. Experimental setup

The experimental test instances include the *gdb* [49], *val* [50], and *egl* [51]. They are three well-known benchmark sets used to evaluate the performance of the algorithm for SO-CARP. 81 test instances are included in these test sets which are all based on the undirected graphs, and different instances based on each graph are generated by changing the capacity of the vehicles. The original one target of minimizing the total consumption of all trips is extended to two targets which are separately minimizing the *total-cost* and the *makespan*. In the first experiment, in order to carry out a fair comparison, ID-MAENS adopts the same parameters as D-MAENS throughout the experiments: the maximum number of iterations $G_{max} = 200$, the population size $p_{size} = 60$, the local search probability $p_{ls} = 0.1$. The only difference is as follows: the size of sub-population is 9 in D-MAENS, while the size is 12 in ID-MAENS because of the amplification. In the second experiment, all parameter settings and program realizations are identical for the three different versions of ID-MAENS. In the third experiment, the parameter settings of ID-MAENS keep unchanged, while the termination conditions of RTS*(GRO with TSA2) and HHEA are set in such a way that the computational time of the three algorithms are comparable. Because the three algorithms are implemented using Visual C++ language and executed on the same personal computer, the computing resources can be regarded equal. All the experiments are conducted for 25 independent runs.

4.2. Performance measures

An important research content of MOP is the performance comparison between different algorithms. Because the results obtained by different algorithms may be different for the same MOP. In order to evaluate the pros and cons of different algorithms on the same issue, it is natural to point out the advantages and disadvantages of the different methods, and to determine the best algorithm. In the literature, Deb thinks the existing measurement methods can be divided into three categories [52]: (1) the metric of an algorithm's convergence; (2) the metric of the diversity of the solutions in the obtained nondominated set; (3) the metric both of the convergence and the diversity. Convergence is used to evaluate the degree of closeness between the approximate set and the current best solution set, and diversity is used to measure the distribution of solutions in the approximate set. Due to the discreteness of the solution space of the MO-CARP, the true Pareto-Optimal Front of the instance is not uniformly distributed. So, the diversity of an algorithm is not certain to follow the law [2]. In this paper, three metrics are used to evaluate the performance of the two algorithms.

1. Distance From Reference Set (I_D)

This metric was proposed by Czyzzak and Jaszkiewicz [53] and it will indicate the closeness of the set waiting for comparison to the Pareto-Optimal Front. Its definition is listed below.

$$I_D(A) = \frac{\sum_{i=1}^{M}(\min\{d(\boldsymbol{x}_j, \boldsymbol{y}_i)\})}{|R|} \quad 1 \leq j \leq N \tag{5}$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ belong to the set A and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M$ belong to the set R. $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ in objective space. $I_D(A)$ firstly calculates the average distance from a solution in the reference set R to the closest solution in A. A smaller value of $I_D(A)$ indicates that A is closer to R. But, it is difficult to get the integral Pareto-Optimal Front in a MO-CARP instance, and the Pareto-Optimal Front itself may even be not uniformly distributed. Hence in this experiment, for each test instance, we combine the nondominated solutions obtained by the two algorithms in 25 runs, and those solutions remained nondominated in this set are used as the reference set R.

2. Purity

Purity was proposed by Bandyopadhyay, Pal and Aruna [54]. It is mainly used to compare the quality and the convergence of results. It can be stated as follows:

$$Purity(A) = \frac{|A \cap M|}{|A|} \tag{6}$$

where A represents the nondominated solution set of one algorithm. $|A|$ is cardinality of set A, and M is the total nondominated solution set after merging all algorithms' results. $A \cap M$ indicates the same part of A and M. In other words, *Purity* is the ratio of nondominated solutions of one algorithm among the results of all the algorithms. And its size represents the convergence of results. The higher value it has, the greater convergence it has among all the algorithms.

3. Hypervolume (HV)

Hypervolume was suggested by Zizler [55] and it is mainly used to describe the extent of the obtained solution set covering Pareto-Optimal Front. That is suitable for evaluating the broadness of the nondominated solution set. It can be expressed as follows:

$$HV(A) = volume(\cup_{i=1}^{n} v_i) \tag{7}$$

In this paper, the reference point is set by maximum values of the results of all the comparative algorithms. *HV* reflects the closeness between the obtained solution set and Pareto-Optimal Front. To a certain extent, it can represent the diversity of the solutions. The greater the value of the *HV*, the closer the distance between the nondominated solution set and Pareto-Optimal Front is. In addition, *HV* is the only unary assessment standard which is consistent with the relationship of Pareto-Dominance. If a set dominates another one, it always has a better *HV* [56]. For this reason, *HV* is one of the most commonly used measures for evaluating the performance of MOP.

4.3. The comparison between ID-MAENS and D-MAENS

In the following tables, V, T and E respectively denote the number of vertices, the number of tasks and the total number of edges in the instance. By the number of tasks we can estimate the size of the test problems. For the I_D and the *Purity* obtained by the two algorithms (ID-MAENS and D-MAENS), we correct the results to 4 decimal places. The performance of a MOEA is usually evaluated from the convergence and the diversity, which can hardly be reflected by a single metric. Hence, the winner of the two algorithms is denoted as that at least two of the three indicators of the algorithm are better than those of the other algorithm. Otherwise, the two algorithms are neck and neck and the winner is denoted as "both". The best results are indicated in bold in each test instance.

The *gdb* set was generated by DeArmon in [49]. This test set contains 23 small or medium scale test instances. Table 3 is the comparison results on the *gdb* benchmark test set between ID-MAENS and D-MAENS. Due to the small scale of the test set, the number of the nondominated solutions in each instance is small. On most of the *gdb* test instances, the comparison results between ID-MAENS and D-MAENS have little difference. Table 3 shows that D-MAENS obtains a better solution on 3 out of 23 *gdb* instances than ID-MAENS, while ID-MAENS is superior to D-MAENS on 6 test instances.

The nondominated solutions obtained by the two algorithms on *gdb* set are also plotted in the objective space after 25 runs. The results are shown in Fig. 5, where "*" represents the results obtained by ID-MAENS and "o" represents the results obtained by D-MAENS.

As can be seen from Table 3, on the test instances of *gdb*2, *gdb*9, *gdb*11, *gdb*18, *gdb*22 and *gdb*23, the results of ID-MAENS are better than D-MAENS. Moreover, Fig. 5 also demonstrates this conclusion. On the *gdb*2, ID-MAENS finds a new nondominated solution and improves the diversity of solutions. On the *gdb*9, in the intermediate area ($300 < total\text{-}cost < 315$, $38 < makespan < 43$), the solutions found by ID-MAENS dominates the solutions obtained by D-MAENS. For *gdb*22, ID-MAENS still has a stronger capability of reaching the area with lower *makespan* than D-MAENS. Also on the *gdb*23, ID-MAENS again finds the solution set which is more close to the true Pareto-Optimal Front. Moreover, on the *gdb*18 and *gdb*11, ID-MAENS covers the objective space more completely than D-MAENS. Summing up the above, the high convergence speed of ID-MAENS is not apparent because of the small size of *gdb* test set.

The *val* test set was generated by Benavent et al. in [50]. 34 *val* instances are a total of 10 groups composed by 10 different graphs. In each group, all instances are based on the same graph and the vehicles in different instances have a different capacity. *val* is the medium-sized test set of CARP and the number of nondominated solutions in each instance is relatively large. Table 4 is the comparison results on the *val* benchmark test set between ID-MAENS and D-MAENS.

Table 3
Results on the *gdb* benchmark test set between ID-MAENS and D-MAENS.

Instance	*V*	*T*	*E*	D-MAENS			ID-MAENS			Winner
				I_D	*Purity*	*HV*	I_D	*Purity*	*HV*	
1	12	22	22	0.0000	1.0000	112	0.0000	1.0000	112	**Both**
2	12	26	26	0.3333	0.6667	668	0.0000	**1.0000**	**674**	**ID-MAENS**
3	12	22	22	0.0000	1.0000	304	0.0000	1.0000	304	**Both**
4	11	19	19	0.0000	1.0000	390	0.0000	1.0000	390	**Both**
5	13	26	26	0.0000	1.0000	1494	0.0000	1.0000	1494	**Both**
6	12	22	22	0.0000	1.0000	543	0.0000	1.0000	543	**both**
7	12	22	22	0.0000	1.0000	420	0.0000	1.0000	420	**Both**
8	27	46	46	**0.3333**	**0.8333**	**665**	2.2208	0.6000	658	**D-MAENS**
9	27	51	51	0.8536	0.3333	455	**0.0000**	**1.0000**	**474**	**ID-MAENS**
10	12	25	25	**0.0000**	**1.0000**	**3561**	0.4472	0.9000	3541	**D-MAENS**
11	22	45	45	1.1497	0.4375	7993	**0.2276**	**0.8125**	**8037**	**ID-MAENS**
12	13	23	23	0.0000	1.0000	83	0.0000	1.0000	83	**Both**
13	10	28	28	0.0000	1.0000	0	0.0000	1.0000	0	**Both**
14	7	21	21	0.0000	1.0000	180	0.0000	1.0000	180	**Both**
15	7	21	21	0.0000	1.0000	50	0.0000	1.0000	50	**Both**
16	8	28	28	0.1429	1.0000	584	0.1429	1.0000	586	**Both**
17	8	28	28	0.8246	**1.0000**	**28**	**0.2000**	0.8000	26	**D-MAENS**
18	9	36	36	0.2857	0.6250	562	**0.0000**	**1.0000**	**574**	**ID-MAENS**
19	8	11	11	0.0000	1.0000	0	0.0000	1.0000	0	**Both**
20	11	22	22	0.0000	1.0000	188	0.0000	1.0000	188	**Both**
21	11	33	33	0.0000	1.0000	642	0.0000	1.0000	642	**Both**
22	11	44	44	1.5603	**0.8333**	292	**0.2500**	0.7500	**308**	**ID-MAENS**
23	11	55	55	0.9211	0.6000	118	**0.0000**	**1.0000**	**132**	**ID-MAENS**

In Table 4, ID-MAENS performs significantly better than D-MAENS on 24 out of the total 34 *val* instances. Through the further analysis of the metric I_D, we find that ID-MAENS obtains a better solution on 21 out of 34 *val* instances than D-MAENS. Note that on *val1A*, *val2A*, *val5B*, *val6C*, *val7C*, *val9C*, *val9D*, *val10B* and *val10D*, ID-MAENS reaches the theoretical optimal value 0 of I_D. And it shows that ID-MAENS finds all the best nondominated solutions after 25 independent runs on these instances. However, only on 6 test instances, ID-MAENS gets a worse result than D-MAENS, and the number of instances on which the results of ID-MAENS are completely dominated by D-MAENS is 1. Then, we focus on the metric *purity*. It is shown from the table that ID-MAENS finds the

Table 4
Results on the *val* benchmark test set between ID-MAENS and D-MAENS.

Instance	*V*	*T*	*E*	D-MAENS			ID-MAENS			Winner
				I_D	*Purity*	*HV*	I_D	*Purity*	*HV*	
1A	24	39	39	0.2857	0.8571	691	**0.0000**	**1.0000**	**693**	**ID-MAENS**
1B	24	39	39	0.0000	1.0000	947	0.0000	1.0000	947	**Both**
1C	24	39	39	0.0000	1.0000	6	0.0000	1.0000	6	**Both**
2A	24	34	34	1.3652	1.0000	4924	**0.0000**	1.0000	**4943**	**ID-MAENS**
2B	24	34	34	0.0000	1.0000	3886	0.0000	1.0000	3886	**Both**
2C	24	34	34	0.0000	1.0000	24	0.0000	1.0000	24	**Both**
3A	24	35	35	0.0000	1.0000	244	0.0000	1.0000	244	**Both**
3B	24	35	35	0.0000	1.0000	37	0.0000	1.0000	37	**Both**
3C	24	35	35	0.0000	1.0000	0	0.0000	1.0000	0	**Both**
4A	41	69	69	1.9659	0.1875	9751	**0.8374**	**0.9000**	**9869**	**ID-MAENS**
4B	41	69	69	1.6739	0.2941	5851	**0.1597**	**1.0000**	**5937**	**ID-MAENS**
4C	41	69	69	**1.0858**	0.4000	1997	1.1409	**1.0000**	**2034**	**ID-MAENS**
4D	41	69	69	**1.1180**	0.3333	**252**	4.3012	**1.0000**	250	**D-MAENS**
5A	34	65	65	4.7692	0.3000	22542	**0.6200**	**0.8947**	**22756**	**ID-MAENS**
5B	34	65	65	5.4250	0.1857	12029	**0.0000**	**1.0000**	**12253**	**ID-MAENS**
5C	34	65	65	**1.2043**	0.3529	6229	3.6000	**0.8000**	**6275**	**ID-MAENS**
5D	34	65	65	**1.3452**	0.3333	1365	2.6681	**0.7500**	**1378**	**ID-MAENS**
6A	31	50	50	0.2727	0.7500	2247	**0.1286**	**1.0000**	**2254**	**ID-MAENS**
6B	31	50	50	0.2500	0.8333	1471	**0.2012**	0.8333	**1474**	**ID-MAENS**
6C	31	50	50	1.2761	0.3333	88	**0.0000**	**1.0000**	**106**	**ID-MAENS**
7A	40	66	66	**0.0000**	**1.0000**	**8904**	0.4706	0.8182	8894	**D-MAENS**
7B	40	66	66	**0.1667**	**0.9167**	**2635**	0.4647	0.7500	2628	**D-MAENS**
7C	40	66	66	0.4828	0.6000	823	**0.0000**	**1.0000**	**834**	**ID-MAENS**
8A	30	63	63	3.6359	0.0000	20747	**0.1500**	**1.0000**	**21009**	**ID-MAENS**
8B	30	63	63	3.1270	0.1905	9903	**0.2648**	**0.8235**	**10132**	**ID-MAENS**
8C	30	63	63	3.7067	0.6000	1982	**1.1818**	**0.7500**	**2093**	**ID-MAENS**
9A	50	92	92	3.9822	0.0667	9111	**1.0118**	**0.9231**	**9378**	**ID-MAENS**
9B	50	92	92	5.1254	0.1176	5041	**0.6429**	**0.8571**	**5257**	**ID-MAENS**
9C	50	92	92	2.9086	0.0000	3293	**0.0000**	**1.0000**	**3492**	**ID-MAENS**
9D	50	92	92	1.7625	0.1250	825	**0.0000**	**1.0000**	**858**	**ID-MAENS**
10A	50	97	97	5.7739	0.1000	25207	**0.0000**	**1.0000**	**26165**	**ID-MAENS**
10B	50	97	97	5.2809	0.0000	11424	**0.0000**	**1.0000**	**12055**	**ID-MAENS**
10C	50	97	97	5.4528	0.0588	7356	**0.0588**	**0.9412**	**7888**	**ID-MAENS**
10D	50	97	97	5.2715	0.1333	3621	**1.4000**	**0.8000**	**3737**	**ID-MAENS**

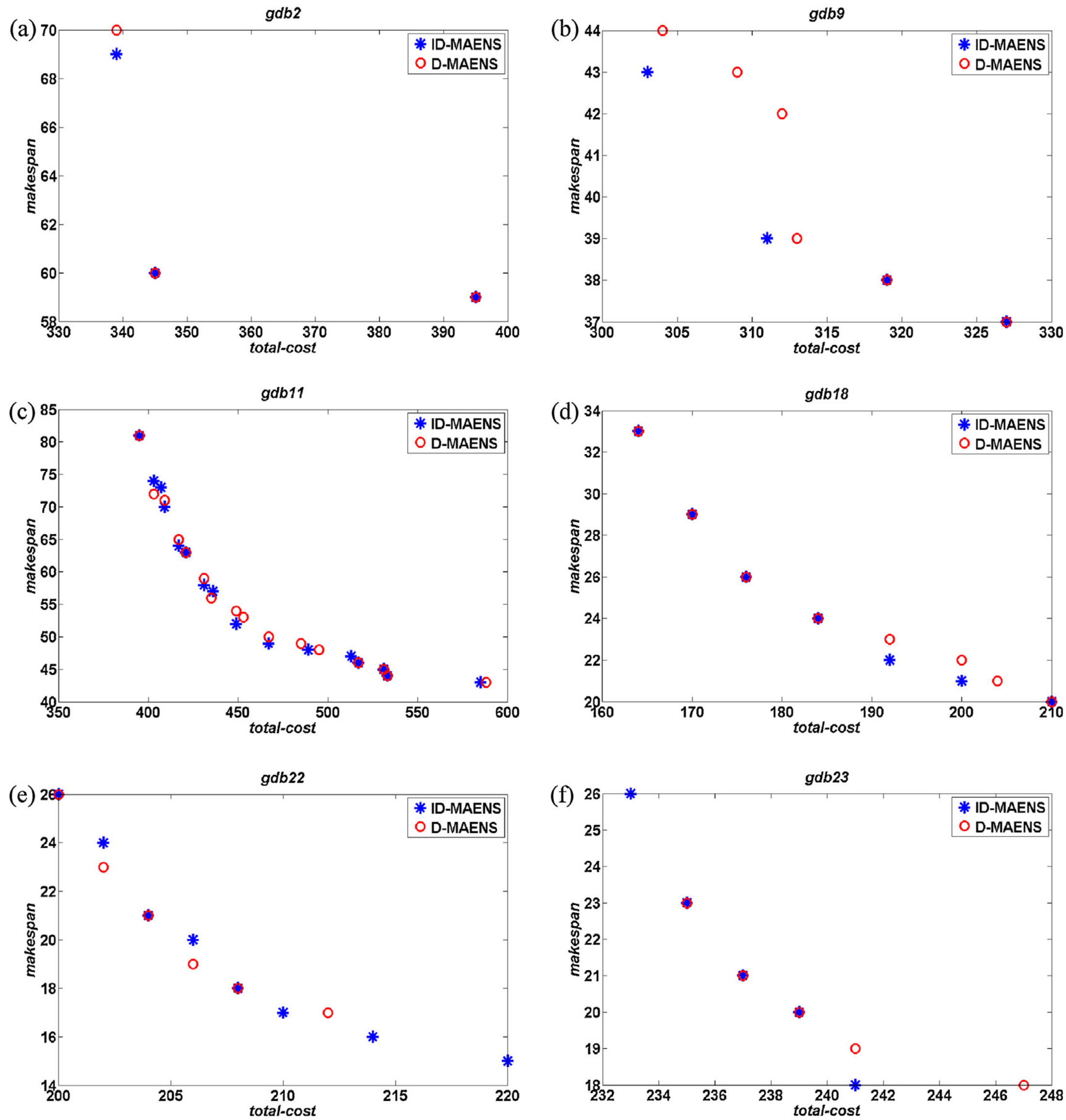

Fig. 5. Nondominated solutions obtained by all 25 runs of two algorithms for the *gdb* benchmark test set.

theoretical optimal value 1 on 21 out of the total 34 instances of *val* set. The number of instances on which D-MAENS achieves the same indicators is 9. Moreover, on most of the *val* set, the *purity* of the improved algorithm is far superior to the old algorithm. The comparison results of *HV* are similar to I_D.

Fig. 6 shows the nondominated solutions of the two algorithms after 25 independent runs. The locations of these nondominated solutions are shown in objective space, where "*" represents the results obtained by ID-MAENS and "o" represents the results obtained by D-MAENS.

In Fig. 6, ID-MAENS on medium-sized *val* test set performs significantly better than on small-scale *gdb* test set. On *val1A* and *val2A*, both algorithms can converge to Pareto-Optimal Front substantially, and the advantage of ID-MAENS is only reflected in a few individual solutions. On *val4A*, *val5C*, *val6C* and *val9A*, the solutions obtained by ID-MAENS lie completely below the solutions obtained by D-MAENS. It indicates that the solution set of the old algorithm is completely dominated by the solution set of the improved algorithm on the above instances. In a word, the improved algorithm speeds up the convergence, and most of the results of ID-MAENS converge to the true Pareto-Optimal Front on *val* test set. Moreover, the diversities of ID-MAENS are maintained well. By counting the number of the nondominated solutions on *val*4A, *val*4B, *val*5C, *val*9A, *val*9B, *val*9D and *val*10A after 25 independent runs, the population size of ID-MAENS is slightly less than that of D-MAENS. But the uniformity of the nondominated solutions and the distribution of the front are not changed. This shows the diversity of the two algorithms is considerable.

Compared with the small and medium instances, the large-scale test data is more and more popular [57]. The *egl* test set was generated by Eglese in [51] which is a large-scale test set for CARP. Based on the data from a winter gritting application in Lancashire, it includes 24 instances based on two graphs. Each graph corresponds to 12 instances. Table 5 is the comparison results on the *egl* benchmark test set between ID-MAENS and D-MAENS.

As can be seen from Table 6, for the large-scale *egl* set, ID-MAENS has more obvious advantages. Taking a closer look at the *HV*, it can be found that ID-MAENS obtains a better solution on 23 out of 24 *egl* instances than D-MAENS. Only on 1 test instances, ID-MAENS gets a worse result than D-MAENS. Next, we analyze the *purity* of the two algorithms. ID-MAENS find the theoretical optimal value 1 on 13 out of the total 24 instances of *egl* set,

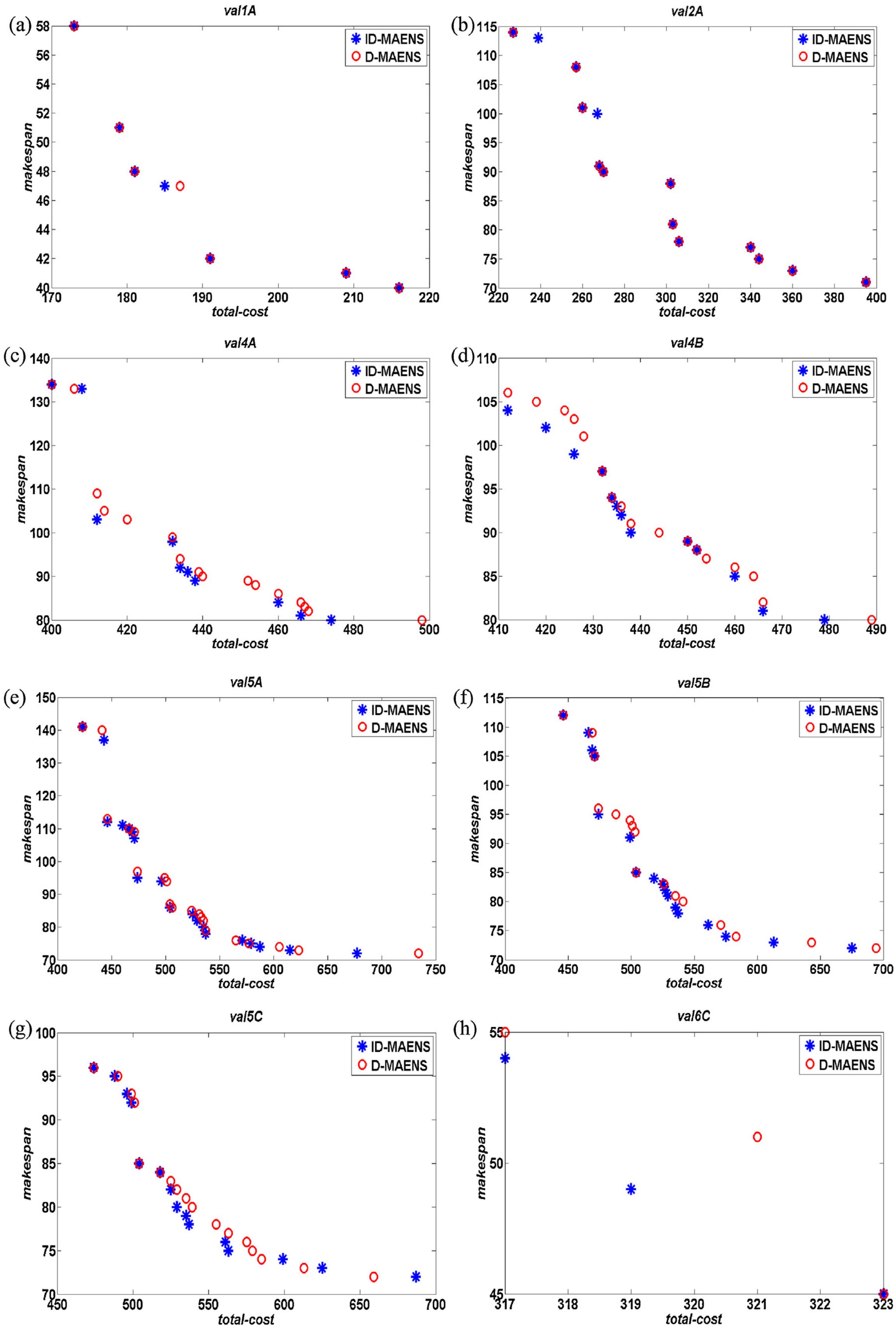

Fig. 6. Nondominated solutions obtained by all 25 runs of two algorithms for the *val* benchmark test set.

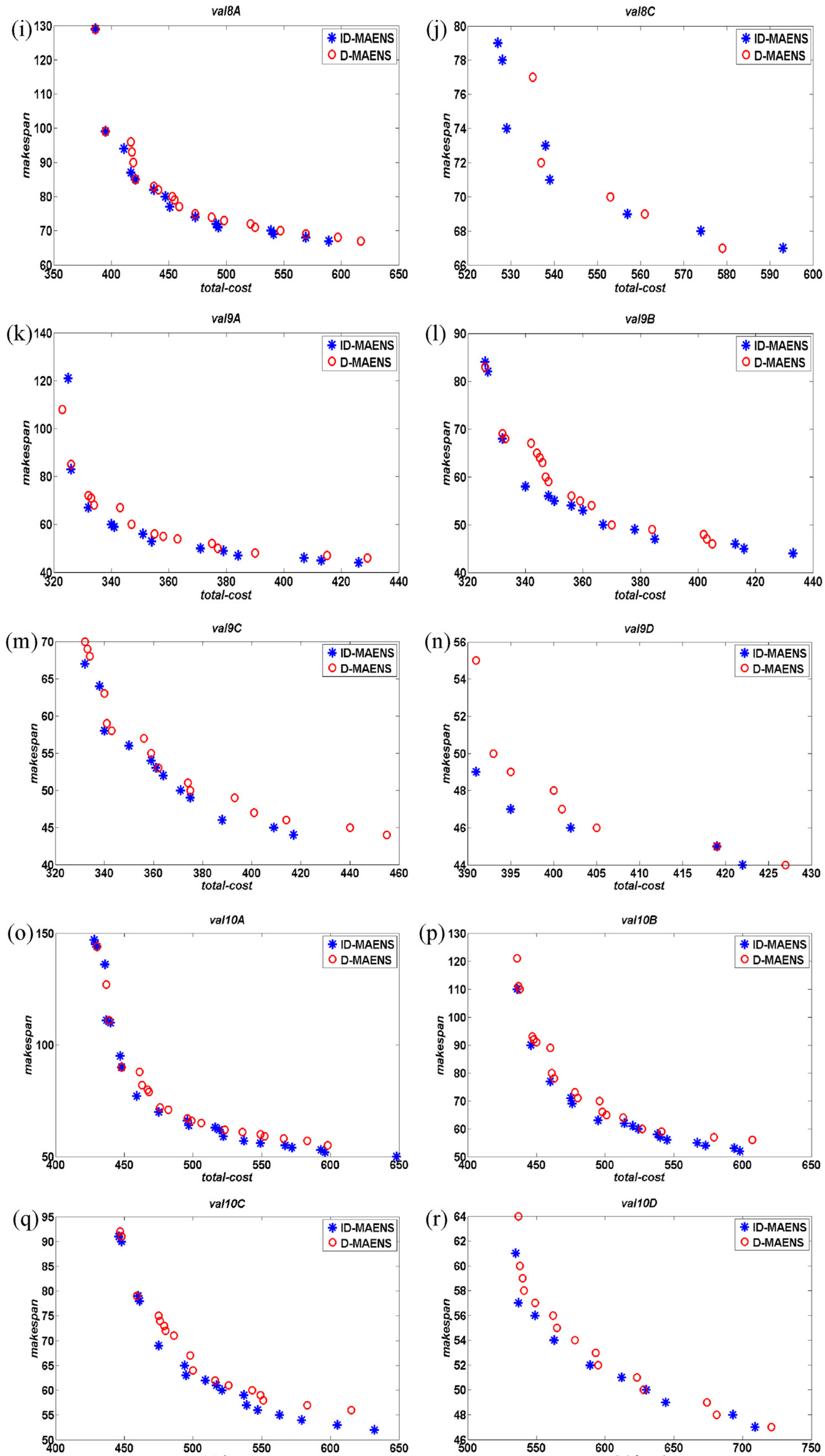

Fig. 6. *Continued*

Table 5
Results on the *egl* benchmark test set between ID-MAENS and D-MAENS.

Instance	V	T	E	D-MAENS			ID-MAENS			Winner
				I_D	*Purity*	*HV*	I_D	*Purity*	*HV*	
E1-A	77	51	98	4.8541	0.6000	26853	**0.0000**	**1.0000**	**27350**	**ID-MAENS**
E1-B	77	51	98	13.4442	1.0000	2499	**0.0000**	1.0000	**2502**	**ID-MAENS**
E1-C	77	51	98	8.0000	0.5000	13005	**0.0000**	**1.0000**	**13261**	**ID-MAENS**
E2-A	77	72	98	4.0355	0.6667	172278	**0.0000**	**1.0000**	**172387**	**ID-MAENS**
E2-B	77	72	98	**1.1374**	0.6667	**44249**	4.2379	**0.8000**	44203	**D-MAENS**
E2-C	77	72	98	7.0000	0.6000	10364	**1.6422**	0.6000	**10585**	**ID-MAENS**
E3-A	77	87	98	33.1337	0.1333	248889	**3.9455**	**1.0000**	**257846**	**ID-MAENS**
E3-B	77	87	98	25.1117	0.3750	48603	**8.4394**	**0.8571**	**51352**	**ID-MAENS**
E3-C	77	87	98	23.3618	0.0000	19638	**0.0000**	**1.0000**	**20627**	**ID-MAENS**
E4-A	77	98	98	37.5344	0.1538	217408	**2.2927**	**0.8182**	**225164**	**ID-MAENS**
E4-B	77	98	98	13.4127	0.1111	62203	**0.0000**	**1.0000**	**64953**	**ID-MAENS**
E4-C	77	98	98	120.9730	0.0000	6942	**0.0000**	**1.0000**	**13622**	**ID-MAENS**
S1-A	140	75	190	13.5591	0.5714	332507	**4.1798**	0.5714	**343319**	**ID-MAENS**
S1-B	140	75	190	16.3741	**0.6000**	188008	**11.4042**	0.3846	**188986**	**ID-MAENS**
S1-C	140	75	190	19.9475	0.1667	63299	**0.0000**	**1.0000**	**66118**	**ID-MAENS**
S2-A	140	147	190	29.6590	0.1905	383272	**15.3230**	**0.8667**	**398821**	**ID-MAENS**
S2-B	140	147	190	46.0224	0.1250	85060	**6.8658**	**0.6250**	**97619**	**ID-MAENS**
S2-C	140	147	190	148.0337	0.0000	27137	**0.0000**	**1.0000**	**38885**	**ID-MAENS**
S3-A	140	159	190	36.9561	0.3333	363850	**16.4767**	**0.8235**	**379111**	**ID-MAENS**
S3-B	140	159	190	83.1551	0.1111	172978	**1.3000**	**0.9000**	**201716**	**ID-MAENS**
S3-C	140	159	190	275.1747	0.0000	41580	**0.0000**	**1.0000**	**72777**	**ID-MAENS**
S4-A	140	190	190	68.0156	0.0000	51986	**0.0000**	**1.0000**	**63441**	**ID-MAENS**
S4-B	140	190	190	325.6264	0.0000	14310	**0.0000**	**1.0000**	**31791**	**ID-MAENS**
S4-C	140	190	190	345.5446	0.0000	4496	**0.0000**	**1.0000**	**7152**	**ID-MAENS**

while D-MAENS fails to be the best on most of the instances. D-MAENS only find the theoretical optimal value 1 on 1 out of the total 24 instances. Moreover, the promotion on the metric of I_D is more significant in ID-MAENS. On the 21 test instances of *egl*, the I_D of ID-MAENS is over two times smaller than that of D-MAENS. This describes the convergence of ID-MAENS on the large-scale problems is significantly better than that of D-MAENS.

Fig. 7 shows the nondominated solutions of the two algorithms after 25 independent runs. The locations of these nondominated solutions are shown in objective space, where "*" represents the results obtained by ID-MAENS and "o" represents the results obtained by D-MAENS.

As can be seen from Fig. 7, the front of ID-MAENS is significantly better than that of D-MAENS for large-scale *egl* set and the advantages of convergence can be fully reflected in ID-MAENS. On *e3-B*, *e3-C*, *e4-A*, *e4-C*, *s2-C*, *s3-C*, *s4-B*, the solutions obtained by D-MAENS are completely dominated by the solutions obtained by ID-MAENS. This advantage is particularly significant on *e4-C*, *s2-C* and *s3-C*. On the other test instances, such as *e2-C*, *s1-A*, *s2-A*, *s2-B* and *s3-A*, ID-MAENS still has a stronger capability of reaching the area with lower *total-cost* and the intermediate area than D-MAENS, although it is outperformed by D-MAENS in the lower *makespan* area. Observing the number of the nondominated solutions obtained by the two algorithms, the disadvantage of having few nondominated solutions in ID-MAENS does not appear on

Table 6
Results on the *egl* benchmark test set among different versions of ID-MAENS.

Instance	I_D				*HV*			
	1	2	3	4	1	2	3	4
E1-A	4.8541	4.8541	4.8541	**0.0000**	49116	49244	49116	**49613**
E1-B	13.4442	9.8647	19.2170	**0.0000**	2499	**2502**	1636	**2502**
E1-C	8.5000	13.0000	**0.5000**	**0.5000**	13005	11483	**13261**	**13261**
E2-A	4.0355	3.7143	0.4286	**0.0000**	172455	172367	172520	**172564**
E2-B	17.4605	19.3834	**3.3333**	15.9204	44249	44308	**46282**	44203
E2-C	15.0237	9.2500	7.8661	**7.2177**	14264	**14747**	14619	14542
E3-A	34.2785	**23.4416**	44.2291	25.4026	256049	262568	258134	**265006**
E3-B	36.9403	**17.5828**	27.1399	18.6459	48883	**53139**	50500	51632
E3-C	26.7583	**1.5000**	13.5491	22.174	20726	**23202**	21825	21715
E4-A	36.2112	37.3142	30.5336	**5.5961**	217408	221401	222461	**225164**
E4-B	23.3829	15.7971	15.3219	**0.0000**	67705	72407	70455	**74645**
E4-C	125.9365	17.8632	139.1883	**1.3463**	9152	17537	8067	**18093**
S1-A	19.9104	18.3849	23.0768	8.7954	335585	344464	345890	**346397**
S1-B	29.7079	17.9706	**11.5528**	22.2786	192838	195659	**198364**	193912
S1-C	29.2697	**5.8952**	12.0763	8.4770	63229	**66646**	65060	66118
S2-A	38.2378	41.7964	**29.7972**	52.4231	447862	454163	**465359**	464311
S2-B	65.9953	45.5460	45.5018	**40.7417**	87648	93378	99343	**100541**
S2-C	167.7924	50.5136	**13.2810**	15.7118	27137	36520	**41216**	38885
S3-A	43.3237	32.9170	**8.3445**	23.8345	414050	418719	430877	**431511**
S3-B	85.2208	88.9589	49.1581	**9.7564**	172978	178208	188179	**201716**
S3-C	284.812	118.2469	10.4216	**7.9354**	67095	89325	116201	**116967**
S4-A	82.3636	57.1233	27.8093	**15.5268**	59062	64774	70109	**71417**
S4-B	168.8623	180.0084	**51.3562**	254.2707	37912	45472	**77727**	56395
S4-C	516.0277	190.0635	157.1681	**0.0000**	10465	14703	14916	**16987**

large-scale test instances. So the diversity of ID-MAENS is maintained well.

By combining Figs. 5–7 and Tables 3–5 together, through three different sizes of test sets, we get the following conclusions: with the expansion of the scale of the instance, the advantages of fast convergence and wide distribution of nondominated solutions in ID-MAENS are more and more obvious. It shows that the improved algorithm has stronger ability in keeping diversity and converging to the true Pareto-Optimal Front. This is due to ID-MAENS reserves the elite individual in each decomposed direction to participate in

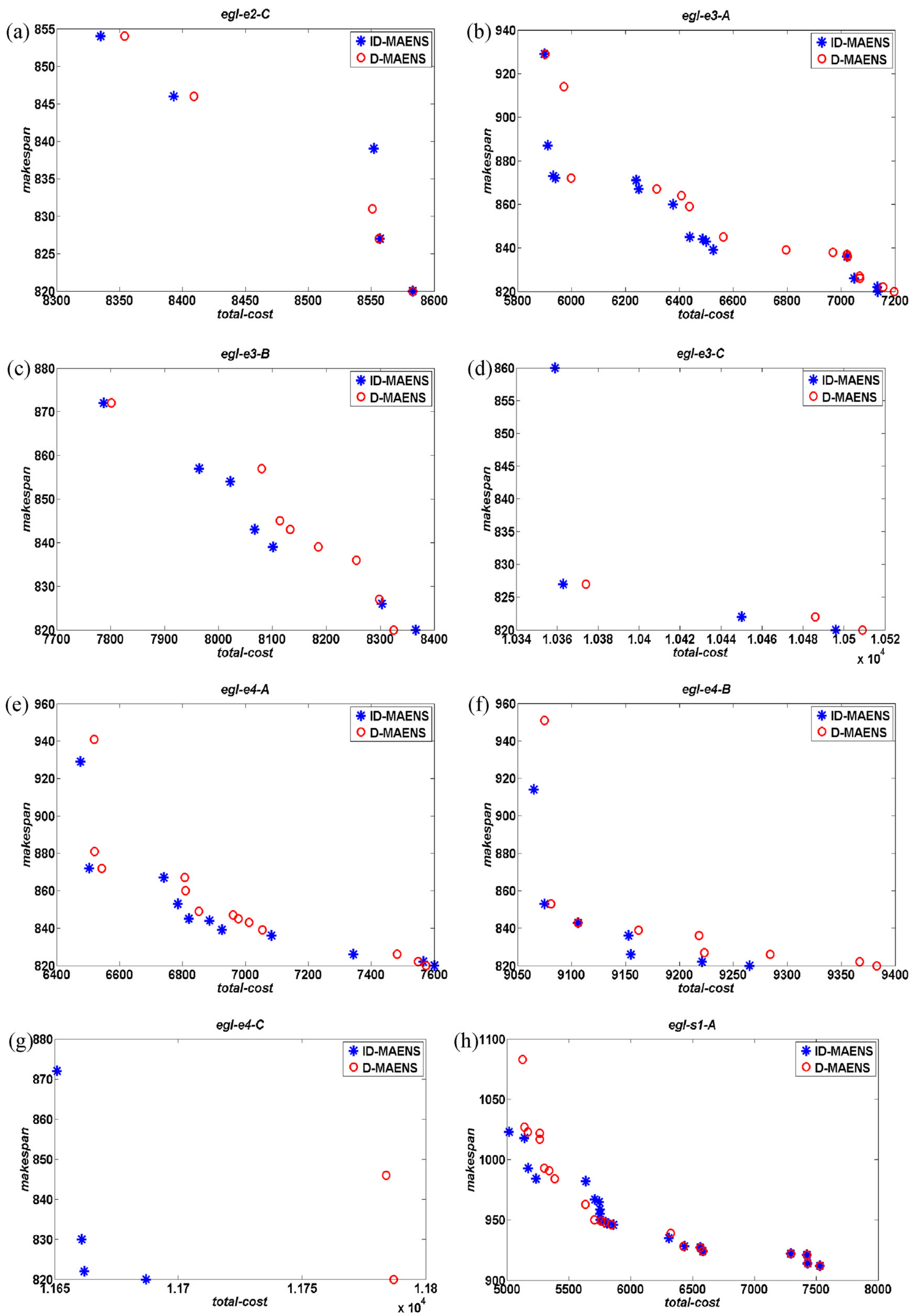

Fig. 7. Nondominated solutions obtained by all 25 runs of two algorithms for the *egl* benchmark test set.

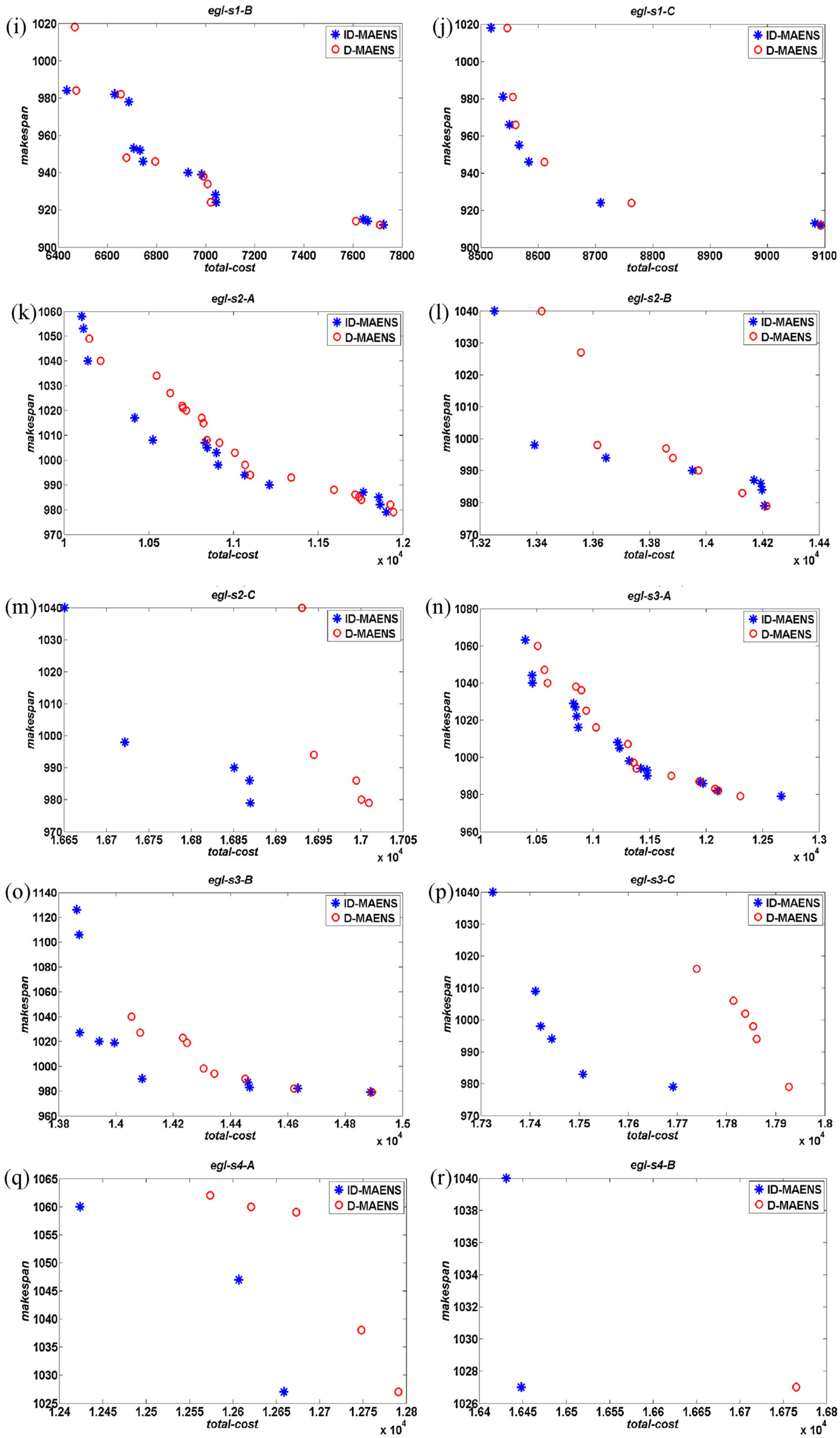

Fig. 7. *Continued.*

Table 7
The leftmost solution by ID-MAENS versus the best solutions by HHEA and RTS* on *gdb* set.

gdb	1	2	3	4	5	6	7	8	9	10	11	12
HHEA												
f_1	316	339	275	287	377	298	325	350	309	275	395	458
f_2	77	73	65	77	84	82	73	50	48	73	91	97
RTS*												
f_1	316	339	275	287	377	298	325	348	303	275	395	458
f_2	84	73	65	78	83	80	74	51	44	76	96	97
ID-MAENS												
f_1	**316**	**339**	275	**287**	**377**	**298**	**325**	**350**	**303**	**275**	**395**	458
f_2	**74**	**69**	65	**74**	**78**	**75**	**68**	**44**	**43**	**70**	**81**	97

gdb	13	14	15	16	17	18	19	20	21	22	23
HHEA											
f_1	544	100	58	127	91	164	55	121	156	200	235
f_2	128	26	17	28	22	36	21	36	33	30	30
RTS*											
f_1	536	100	58	127	91	164	55	121	156	200	233
f_2	140	24	15	29	21	39	21	36	33	26	33
ID-MAENS											
f_1	536	**100**	58	**127**	**91**	**164**	55	**121**	**156**	200	**233**
f_2	151	**21**	15	**26**	**13**	**33**	21	**34**	**27**	26	**25**

the evolution process and uses the timely replacement mechanism to accelerate the convergence of the algorithm. For the large-scale *egl* set which requires more computing resources, ID-MAENS can achieve the optimal performance in the same evolutionary algebra.

4.4. The comparison among different versions of ID-MAENS

ID-MAENS incorporates two different improvements to deal with the MO-CARP. In this section, three different versions of ID-MAENS are tested in order to evaluate the impact of these components on the algorithm's overall performance. It would be useful to know which one plays which role in improving the final results. These variants are summarized as follows: "1" denotes the D-MAENS; "2" denotes that only the first improvement is applied in the optimization process; "3" denotes that only the second improvement is applied in the optimization process; while "4" denotes both steady-state evolutionary algorithm and elite archive are used. The *egl* benchmark test is applied to validate the performance of different versions, and the experimental results are listed in Table 6. The best results are indicated in bold in each test instance.

In Table 6, the experimental results demonstrate that "4" consistently outperforms "2" and "3", suggesting that using two improvements is more powerful than using just one. Moreover, the experimental results of "2" and "3" are significantly better

Table 8
The leftmost solution by ID-MAENS versus the best solutions by HHEA and RTS* on *val* set.

val	1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C
HHEA												
f_1	173	173	245	227	259	457	81	87	138	400	412	436
f_2	83	63	41	115	108	71	41	33	27	147	127	101
RTS*												
f_1	173	173	245	227	259	457	81	87	138	400	412	428
f_2	80	65	41	115	108	71	41	33	27	141	115	109
ID-MAENS												
f_1	**173**	**173**	245	**227**	259	457	81	**87**	138	**400**	**412**	434
f_2	**58**	**59**	41	**114**	108	71	41	**32**	27	**138**	**106**	96

val	4D	5A	5B	5C	5D	6A	6B	6C	7A	7B	7C	8A
HHEA												
f_1	538	423	446	474	595	223	233	317	279	283	334	386
f_2	84	147	120	101	80	77	68	55	90	76	50	134
RTS*												
f_1	530	423	446	474	583	233	233	317	279	283	334	386
f_2	84	147	117	105	99	80	68	54	94	68	50	133
ID-MAENS												
f_1	536	**423**	**446**	**474**	586	**223**	245	317	**279**	**283**	334	**386**
f_2	82	**141**	**112**	**96**	83	**75**	54	54	**85**	**58**	50	**129**

val	8B	8C	9A	9B	9C	9D	10A	10B	10C	10D
HHEA										
f_1	395	530	325	326	332	398	430	436	447	538
f_2	105	94	123	88	83	54	153	147	106	76
RTS*										
f_1	395	524	324	326	332	391	**428**	**436**	446	534
f_2	106	85	111	86	71	51	**145**	**116**	94	70
ID-MAENS										
f_1	**395**	**523**	**323**	**326**	**332**	**391**	429	437	**446**	**533**
f_2	**99**	**82**	**108**	**84**	**67**	**49**	146	116	**90**	**66**

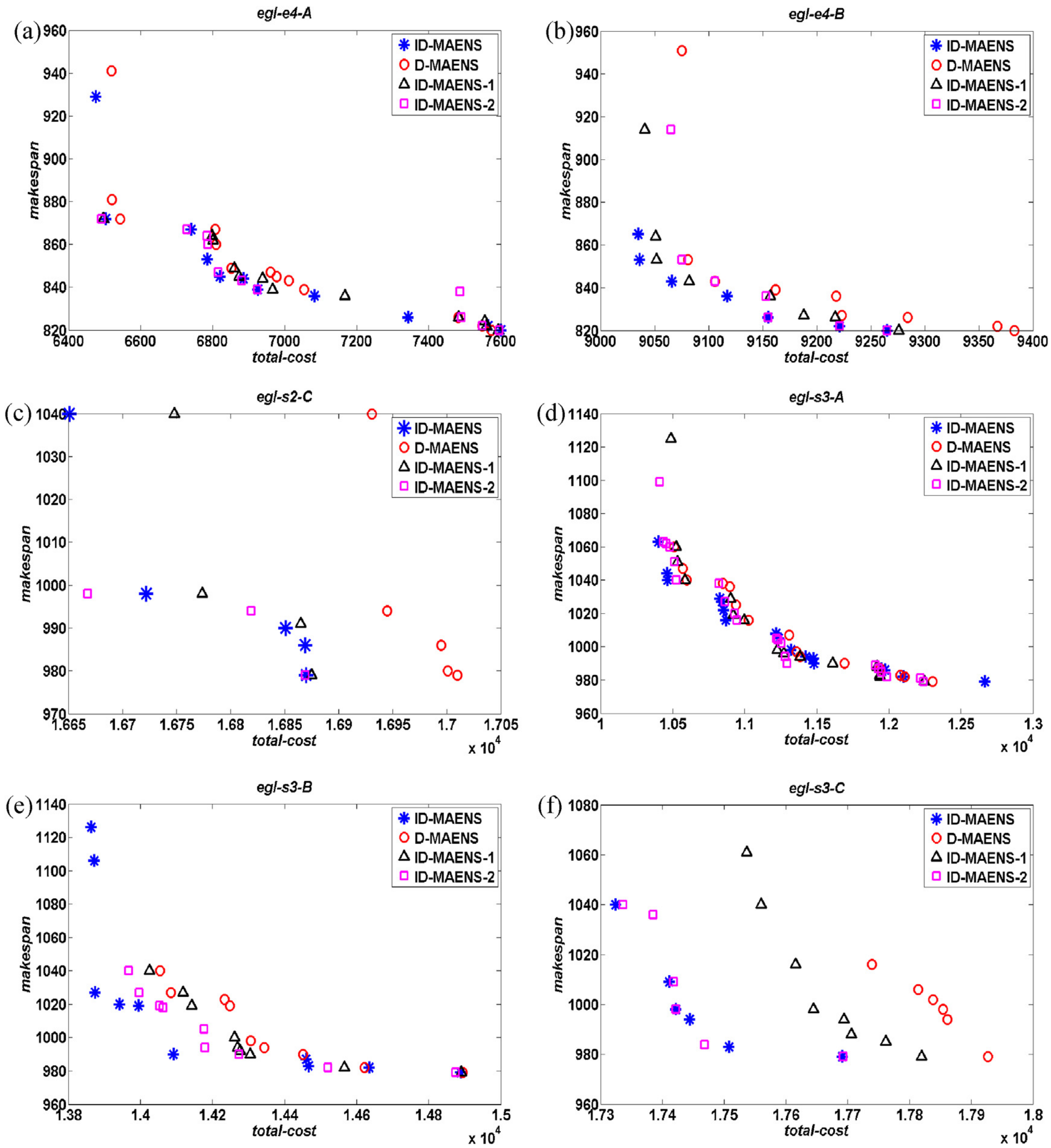

Fig. 8. Nondominated solutions obtained by different versions of ID-MAENS after total 25 runs.

than "1", which concludes that both the two improvements are playing an important role in improving the final results. In a word, the results above highlight the following: (1) The timely replacement of individuals and the elite archive can accelerate the convergence speed and (2) By combining together, the two improvements give an overall improvement in the performance of the algorithm. Fig. 8 shows the nondominated solutions of the different versions of ID-MAENS after 25 independent runs. ID-MAENS-1 denotes that only the first improvement is applied; ID-MAENS-2 denotes that only the second improvement is applied.

The nondominated solutions are given in Fig. 8. The two versions with different improvements (ID-MAENS-1 and ID-MAENS-2) are much better than D-MAENS. They both found the solution sets which are more close to the Pareto-Optimal Front than D-MAENS. As expected, ID-MAENS with both two improvements performed the best. ID-MAENS not only had a stronger convergence but also kept a better diversity than other versions.

4.5. The comparison among ID-MAENS, HHEA [9] and RTS* [39]

The MO-CARP algorithm ID-MAENS is compared with SO-CARP algorithms in terms of quality of solutions and computational efficiency in this section. In this section, the target of SO-CARP algorithms is still the *total-cost* of *N* vehicle trips. After each independent run, the best solution with the least *total-cost* (which is the f_1) is obtained in SO-CARP algorithms. Then we calculate the *makespan* of the above best solution and regard it as f_2. The comparison between the leftmost solution (the nondominated solution with the least f_1) obtained by ID-MAENS and the best solution (if the SO-CARP optimal solutions are repeatedly achieved in 25 runs, the one with the least f_2 is selected as the best solution) obtained by some classical SO-CARP algorithms is made on all the benchmark sets in this section. The RTS* (GRO with TSA2) can be download at the homepage of Yi Mei (http://nical.ustc.edu.cn/ymei/publications.htm). Both RTS* and HHEA run 25 times and all their results are preserved. To make a fair comparison, the termination conditions of RTS* and HHEA are set in such a way that the running time of the three algorithms

Table 9
The leftmost solution by ID-MAENS versus the best solutions by HHEA and RTS* on *egl* set.

egl	E1-A	E1-B	E1-C	E2-A	E2-B	E2-C	E3-A	E3-B	E3-C	E4-A	E4-B	E4-C
HHEA												
f_1	3548	4514	5605	5018	6321	8343	5899	7797	10317	6476	9061	11712
f_2	943	883	844	953	878	854	929	875	875	945	914	914
RTS*												
f_1	3548	4498	5595	5018	6317	**8339**	5898	7789	**10305**	6476	9026	11626
f_2	943	899	836	953	878	**854**	958	872	**875**	930	914	872
ID-MAENS												
f_1	3548	4525	5595	5018	6340	8414	**5898**	7789	10307	**6472**	9004	**11618**
f_2	943	839	836	953	864	854	**929**	872	875	**930**	918	**820**

egl	S1-A	S1-B	S1-C	S2-A	S2-B	S2-C	S3-A	S3-B	S3-C	S4-A	S4-B	S4-C
HHEA												
f_1	5104	6436	8518	10079	13387	16651	10434	14008	17444	12537	16665	21077
f_2	1032	984	1018	1063	1040	1051	1126	1060	1061	1100	1067	1027
RTS*												
f_1	5018	6394	8518	9984	13310	**16648**	**10295**	13860	**17326**	12417	16503	20989
f_2	1023	1050	1018	1076	1060	**1040**	**1070**	1126	**1040**	1093	1047	1035
ID-MAENS												
f_1	5018	6422	8518	10122	13345	16682	10347	13918	17363	12442	**16443**	21195
f_2	1023	984	1018	1063	1040	1040	1070	1060	1086	1058	**1027**	1027

are comparable. The solution dominating the others is indicated in bold in each test instance.

In Table 7, it can be found that ID-MAENS obtain the better solutions which dominate the solutions by RTS* and HHEA on 17 *gdb* instances. On *gdb*3, *gdb*12, *gdb*15, *gdb*19 and *gdb*22, the three algorithms get the same results. Through the above analysis, compared with SO-CARP algorithms, ID-MAENS is able to obtain better solutions in terms of both the *total-cost* and the *makespan*.

Table 8 gives leftmost solution by ID-MAENS versus the best solutions by HHEA and RTS* on val set. In Table 8, from the view of dominating, ID-MAENS finds the solutions dominating those obtained by the other two SO-CARP algorithms on 21 *val* instances. At the same time, RTS* only get better solutions on 2 *val* instances and HHEA fails to be the best on any instance. Next, we focus on the f_1 (the total-cost which is valued more by SO-CARP algorithms) of these solutions. It is shown from the table that these three algorithms get the same f_1 on most *val* instances (22 instances). ID-MAENS obtains the smallest f_1 on 2 instances and RTS* obtains the smallest f_1 on 4 instances, while HHEA does not win on all *val* instances again. The experiments on *val* set have proved that

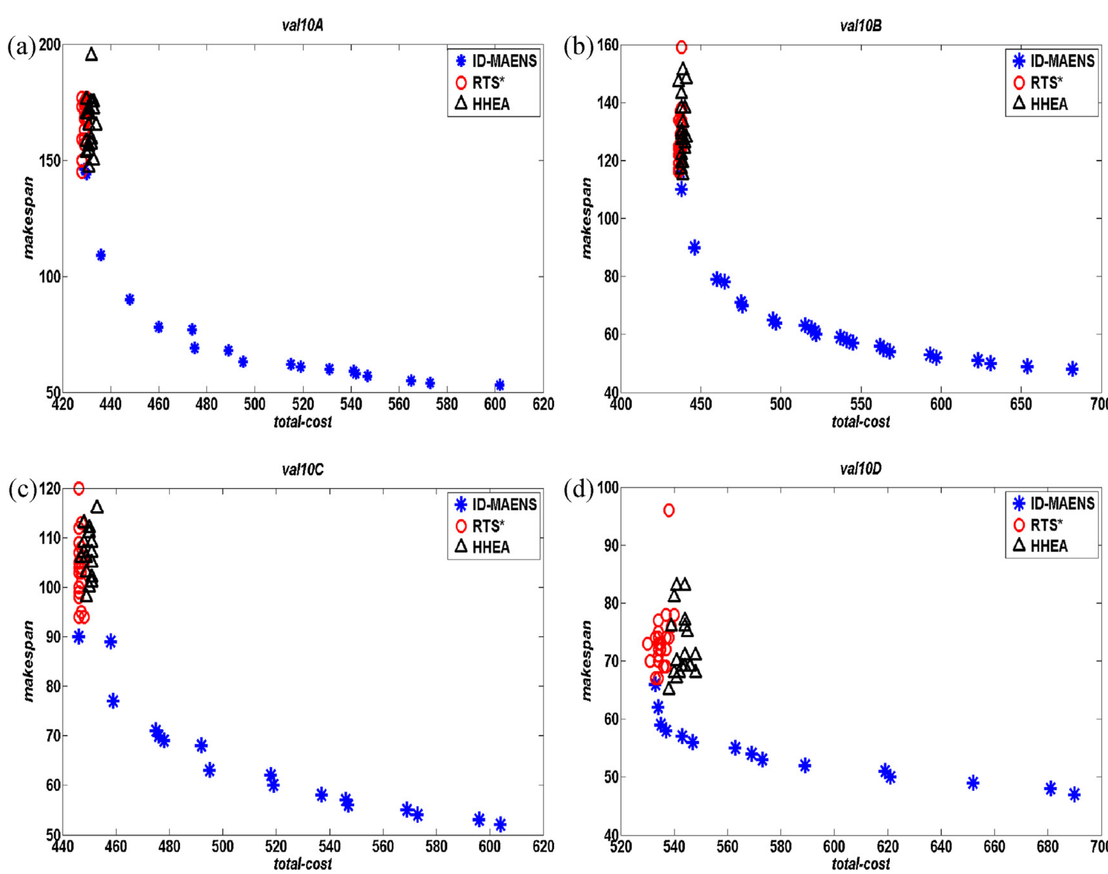

Fig. 9. Nondominated solutions by ID-MAENS and all the solutions obtained by RTS* and HHEA.

ID-MAENS is competitive in terms of quality of solutions compared with SO-CARP algorithms.

Table 9 gives the leftmost solution by ID-MAENS versus the best solutions by HHEA and RTS* on *egl* set.

On *egl* set, most of the solutions found by the three algorithms are non-dominated. It can be observed that ID-MAENS only find the solutions dominating those obtained by RTS* and HHEA on 4 instances. RTS* only performs the best on 5 instance. By only analyzing the f_1, we find that RTS* provides the smallest f_1 on 13 instances. The superiority of RTS* is more and more evident with the increase in size of data (RTS* gets the smallest f_1 on 4 *egl-E* series instances, while it gets the smallest f_1 on 9 *egl-S* series instances). On the one hand, it is due to the effectiveness of global repair operator (GRO) and the excellent performance of TSA algorithm. On the other hand, the computational resources assigned to each sub-problem might not be sufficient for ID-MAENS. SO-CARP algorithms solely focus on optimizing one objective (*total-cost*) and their searching areas focus on the lower-f_1 region in solution space. However, by using the decomposition method for MOP, ID-MAENS uniformly allocates its computational resources to N decomposition directions to search for the whole Pareto-Front. On these simple test instances such as the *gdb* and *val* series, ID-MAENS can find good solutions in terms of both the *total-cost* and the *makespan* because of the sufficient computational resources. Although the result of ID-MAENS is inferior to RTS*, ID-MAENS performs significantly better than another SO-CARP algorithm HHEA (ID-MAENS can obtain smaller f_1 than HHEA on total 14 *egl* instances). In HHEA, the heuristic information of top individuals is constantly learning and then it feeds back to guide the next evolution. The advantage of the above strategy is that it has a high convergence speed under a limited function evaluation when applying to small size test instance. But on some large scale instances, HHEA is easy to fall into the local optimal solution and do not evolve. Moreover, the 2-opt operator used in HHEA has a small search step. While in ID-MAENS, a large step local search method is used and it can search within a large neighborhood of the current solution. Hence, it can easily jump out of the local optimum and find a better solution than HHEA. At the end of this experiment, the nondominated solutions found by ID-MAENS and all the solutions obtained by the two SO-CARP algorithms after 25 independent runs on val10 set are shown in the objective space. This aims to observe the differences of solutions between MO-CARP algorithms and SO-CARP algorithms. Fig. 9 gives the nondominated solutions by ID-MAENS and the all the solutions obtained by RTS* and HHEA.

In Fig. 9, we find that MO-CARP algorithm not only obtains these low total-cost solutions but also returns a series of “tradeoff” solutions between the two goals (*total-cost* and *makespan*), which is the attractive advantage of MOPs. It demonstrates that MO-CARP has greater research value because decision makers can choose the right vehicle scheme according to their different preferences (emphasis) on the two objectives.

5. Conclusion and future work

In this paper, the advantages and disadvantages of D-MAENS are analyzed and an improved algorithm called ID-MAENS is proposed. ID-MAENS is more in line with the theory which is based on the decomposition framework to solve MOPs. By retaining the current best solution for each sub-problem and adding it to the original sub-population of D-MAENS, the improved algorithm not only keeps the original population characteristics of D-MAENS but also speeds up the convergence with the neighbors sharing mechanism between sub-populations. At the same time, the timely replacement mechanism between the new solutions and the old solutions is adopted to ensure that the current best solutions can be the first time to participate in the evolution process. The experimental results demonstrate that ID-MAENS is able to find much better diversity of the solutions and better convergence to the true Pareto-optimal. Moreover, The ID-MAENS also proves to be competitive to SO-CARP algorithms in terms of quality of solutions and computational efficiency.

Compared with the SO-CARP, the mathematical model of MO-CARP is more effective. But we still need to consider other factors in most practical application, such as: the time window constraints, multi-depot, multi-vehicles. Therefore, our future work will focus on solving the model of CARP which is more close to the practical application.

Acknowledgements

We would like to express our sincere appreciation to the anonymous reviewers for their valuable comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Basic Research Program (973 Program) of China under Grant 2013CB329402, the National Natural Science Foundation of China, under Grants 61001202, 61371201, and 61272279, the National Research Foundation for the Doctoral Program of Higher Education of China, under Grants 20100203120008, the Fundamental Research Funds for the Central Universities, under Grant K5051302028, the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grant B07048, and the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

References

[1] M.R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, E. Talbi, A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests, Appl. Soft Comput. 12 (2012) 1426–1439.
[2] Y. Mei, K. Tang, X. Yao, Decomposition-Based Memetic Algorithm for Multi-objective Capacitated Arc Routing Problems, IEEE Trans. Evol. Comput. 15 (2) (2011) 151–165.
[3] K. Ghoseiria, S.F. Ghannadpoura, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, Appl. Soft Comput. 10 (2010) 1096–1107.
[4] A.A. Juan, J. Faulin, R. Ruiz, B. Barrios, S. Caballé, The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem, Appl. Soft Comput. 10 (2010) 215–224.
[5] M. Yaghini, M. Momeni, M. Sarmadi, A Simplex-based simulated annealing algorithm for node-arc capacitated multicommodity network design, Appl. Soft Comput. 12 (2012) 2997–3003.
[6] J. Kratica, M. Milanović, Z. Stanimirović, D. Tošić, An evolutionary-based approach for solving a capacitated hub location problem, Appl. Soft Comput. 11 (2011) 1858–1866.
[7] Y. Mei, K. Tang, X. Yao, A Memetic Algorithm for Periodic Capacitated Arc Routing Problem, IEEE Trans. Syst. Man Cybern. B 41 (6) (2011) 1654–1667.
[8] L.N. Xing, P. Rohlfshagen, Y.W. Chen, X. Yao, A Hybrid Ant Colony Optimization Algorithm for the Extended Capacitated Arc Routing Problem, IEEE Trans. Syst. Man Cybern. B 41 (4) (2011) 1110–1123.
[9] L.N. Xing, P. Rohlfshagen, Y.W. Chen, X. Yao, An Evolutionary Approach to the Multidepot Capacitated Arc Routing Problem, IEEE Trans. Evol. Comput. 14 (3) (2010) 356–374.
[10] P. Lacomme, C. Prins, M. Sevaux, A Genetic Algorithm for a bi-objective capacitated arc routing problem, Comput. Oper. Res. 33 (12) (2006) 3473–3493.
[11] Y. Mei, K. Tang, X. Yao, Capacitated Arc Routing Problem in Uncertain Environments, in: Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC2010), Barcelona, Spain, 18–23 July, 2010, pp. 1400–1407.
[12] K. Ghoseiria, B. Nadjari, An ant colony optimization algorithm for the bi-objective shortest path problem, Appl. Soft Comput. 10 (2010) 1237–1246.
[13] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[14] P. Lacomme, C. Prins, W. Ramdane-Cherif, Competitive memetic algorithms for arc Routing Problems, Ann. Oper. Res. 131 (1) (2004) 159–185.
[15] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[16] K. Tang, Y. Mei, X. Yao, Memetic Algorithm with Extended Neighborhood Search for Capacitated Arc Routing Problems, IEEE Trans. Evol. Comput. 13 (5) (2009) 1151–1166.

[17] G. Syswerda, Uniform crossover in genetic algorithms, in: J. David Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1989, pp. 2–9.
[18] D. Whitley, The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best, in: J. David Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1989, pp. 116–121.
[19] A. Ghosh, S. Tsutsui, H. Tanaka, Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals, in: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, New York, 1998, pp. 666–671.
[20] M.S. Alam, M.M. Islam, X. Yao, K. Murase, Diversity Guided Evolutionary Programming: a novel approach for continuous optimization, Appl. Soft Comput. 12 (2012) 1693–1707.
[21] K.C. Tan, Y.J. Yang, C.K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, IEEE Trans. Evol. Comput. 10 (5) (2006) 527–549.
[22] J. Branke, M. Cutaia, H. Dold, Reducing genetic drift in steady state evolutionary algorithms, in: W. Banzhaf, et al. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 68–74.
[23] J. Wakunda, A. Zell, Median-selection for parallel steady-state evolution strategies, in: Proceedings of the Parallel Problem Solving from Nature—PPSNVI, Lecture Notes in Computer Science, vol. 1917, Springer, 2000, pp. 405–414.
[24] C.L. Valenzuela, A simple evolutionary algorithm for multi-objective optimization (SEAMO), in: Congress on Evolutionary Computation (CEC), Honolulu, Hawaii, 2002, pp. 717–722.
[25] B.L. Golden, R.T. Wong, Capacitated arc routing problems, Networks 11 (3) (1981) 305–316.
[26] R. Hirabayashi, Y. Saruwatari, N. Nishida, Tour construction algorithm for the capacitated arc routing problem, Asia-Pacific J. Oper. Res. 9 (2) (1992) 155–175.
[27] B.L. Golden, J.S. DeArmon, E.K. Baker, Computational experiments with algorithms for a class of routing problems, Comput. Oper. Res. 10 (1) (1983) 47–59.
[28] G. Ulusoy, The fleet size and mix problem for capacitated arc routing, Eur. J. Oper. Res. 22 (3) (1985) 329–337.
[29] W.L. Pearn, Approximate solutions for the capacitated arc routing problem, Comput. Oper. Res. 16 (6) (1989) 589–600.
[30] E. Benavent, V. Campos, A. Corberan, E. Mota, The capacitated arc routing problem, a heuristic algorithm, Qiiestiio 14 (1–3) (1990) 107–122.
[31] W.L. Pearn, Augment-insert algorithms for the capacitated arc routing problem, Comput. Oper. Res. 18 (2) (1991) 189–198.
[32] R.W. Eglese, Routing winter gritting vehicles, Discrete Appl. Math. 48 (3) (1994) 231–244.
[33] A. Hertz, G. Laporte, M. Mittaz, A tabu search heuristic for the capacitated arc routing problem, Oper. Res. 48 (1) (2000) 129–135.
[34] P. Beullens, L. Muyldermans, D. Cattrysse, D. Van Oudheusden, A guided local search heuristic for the capacitated arc routing problem, Eur. J. Oper. Res. 147 (3) (2003) 629–643.
[35] P. Greistorfer, A tabu scatter search metaheuristic for the arc routing problem, Comput. Ind. Eng. 44 (2) (2003) 249–266.
[36] H. Handa, L. Chapman, X. Yao, Robust route optimization for gritting/salting trucks: a CERCIA experience, IEEE Comput. Intell. Mag. 1 (1) (2006) 6–9.
[37] H. Handa, D. Lin, L. Chapman, X. Yao, Robust solution of salting route optimization using evolutionary algorithms, in: Proc. IEEE Congr. Evol. Comput., Vancouver, BC, Canada, 2006, pp. 3098–3105.
[38] J. Brandão, R. Eglese, A deterministic tabu search algorithm for the capacitated arc routing problem, Comput. Oper. Res. 35 (4) (2008) 1112–1126.
[39] Y. Mei, K. Tang, X. Yao, A global repair operator for capacitated arc routing problem, IEEE Trans. Syst. Man Cybern. B 39 (3) (2009) 723–734.
[40] F.L. Usberti, P.M. França, A.L.M. França, GRASP with evolutionary path-relinking for the capacitated arc routing problem, Comput. Oper. Res. 40 (2) (2013) 3206–3217.
[41] P. Lacomme, C. Prins, W. Ramdane-Chérif, Evolutionary algorithms for multiperiod arc routing problems, in: Proc. 9th Int. Conf. Inf. Process. Manage. Uncertainty Knowl.-Based Syst., 2002, pp. 845–852.
[42] F. Chu, N. Labadi, C. Prins, A scatter search for the periodic capacitated arc routing problem, Eur. J. Oper. Res. 169 (2) (2006) 586–605.
[43] G. Laporte, R. Musmanno, F. Vocaturo, An adaptive large neighbourhood search heuristic for the capacitated arc routing problem with stochastic demands, Transp. Sci. 44 (1) (2010) 125–135.
[44] A. Amberg, W. Domschke, S. Vo, Multiple center capacitated arc routing problems: a tabu search algorithm using capacitated trees, Eur. J. Oper. Res. 124 (2) (2000) 360–376.
[45] H. Xu, C.H. Zhang, Y.A. Tan, J. Lu, An improved evolutionary approach to the Extended Capacitated Arc Routing Problem, Expert Syst. Appl. – ESWA 38 (4) (2011) 4637–4641.
[46] Y. Mei, K. Tang, X. Yao, An improved genetic algorithm for the extended capacitated arc routing problem, in: Proceedings of the IEEE Congress on Evolutionary Computation, United States, 18–21 May, 2009, pp. 1699–1706.
[47] L.C. Jiao, H.D. Wang, R.H. Shang, F. Liu, A Coevolutionary Multi-Objective Optimization Algorithm Based on Direction Vectors, Inform. Sci. 00 (2011) 1–35.
[48] R.H. Shang, L.C. Jiao, F. Liu, A Novel Immune Clonal Algorithm for MO Problems, IEEE Trans. Evol. Comput. 16 (1) (2012) 35–49.
[49] J.S. DeArmon, A comparison of heuristics for the capacitated Chinese postman problem (Master's thesis), University of Maryland, 1981.
[50] E. Benavent, V. Campos, A. Corberan, E. Mota, The capacitated arc routing problem: lower bounds, Networks 22 (1992) 669.
[51] R.W. Eglese, Routeing winter gritting vehicles, Discrete Appl. Math. 48 (3) (1994) 231–244.
[52] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, Wiley, Chichester, UK, 2001.
[53] P. Czyzak, A. Jaszkiewicz, Pareto simulated annealing-a metaheuristic technique for multiple-objective combinatorial optimization, J. Multicriteria Decis. Anal. 7 (1998) 34–47.
[54] S. Bandyopadhyay, S. Pal, B. Aruna, Multiobjective GAs, quantitative indices, and pattern classification, IEEE Trans. Syst. Man Cybern. B 4 (5) (2004) 2088–2099.
[55] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, in: EUROGEN, 2001, pp. 95–100.
[56] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. (2009).
[57] K.Q. Cai, J. Zhang, C. Zhou, X.B. Cao, K. Tang, Using computational intelligence for large scale air route networks design, Appl. Soft Comput. 12 (2012) 2790–2800.