

A New Local Search-based Multiobjective Optimization Algorithm

Bili Chen, Wenhua Zeng, Yangbin Lin and Defu Zhang

Abstract—In this paper, a new multiobjective optimization framework based on nondominated sorting and local search (NSLS) is introduced. The NSLS is based on iterations. At each iteration, given a population P , a simple local search method is used to get a better population P' , and then the nondominated sorting is adopted on $P \cup P'$ to obtain a new population for the next iteration. Moreover, the farthest-candidate approach is combined with the nondominated sorting to choose the new population for improving the diversity. Additionally, another version of NSLS (NSLS-C) is used for comparison, which replaces the farthest-candidate method with the crowded comparison mechanism presented in the nondominated sorting genetic algorithm II (NSGA-II). The proposed method (NSLS) is compared with NSLS-C and the other three classic algorithms: NSGA-II, MOEA/D-DE, and MODEA on a set of seventeen bi-objective and three tri-objective test problems. The experimental results indicate that the proposed NSLS is able to find a better spread of solutions and a better convergence to the true Pareto-optimal front compared to the other four algorithms. Moreover, the sensitivity of NSLS is also experimentally investigated in this paper.

Index Terms—Multiobjective optimization, nondominated sorting, local search, diversity, test problems

I. INTRODUCTION

A multiobjective optimization problem (MOP) can be defined as follows:

$$\begin{aligned} & \text{Minimize: } F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ & \text{Subject to: } \mathbf{x} \in S \end{aligned} \quad (1)$$

where $F: S \rightarrow R^m$ consists of m real-valued objective functions, S is the decision variable space, and R^m is the objective space.

Denote a solution \mathbf{x} by $\mathbf{x} = (x_1, x_2, \dots, x_n) \in R^m$. Suppose that there are two solutions \mathbf{x}_1 and \mathbf{x}_2 , \mathbf{x}_1 is said to dominate \mathbf{x}_2 which can be denoted as $\mathbf{x}_1 \succ \mathbf{x}_2$ or $\mathbf{x}_2 \prec \mathbf{x}_1$, if $\forall i \in \{1, 2, \dots, m\}$, $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ and $\exists i \in \{1, 2, \dots, m\}$, $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$. A solution \mathbf{x} is called a Pareto-optimal

Manuscript received April 12, 2013; this work is supported in part by the National Nature Science Foundation of China under Grant No. 61272003, No. 60672018, and No. 40774065.

Bili Chen is with the Department of Computer Science, Xiamen University, Xiamen, Fujian, China (e-mail: bilichen.xmu@qq.com).

Wenhua Zeng is with the Department of Computer Science, Xiamen University, Xiamen, Fujian, China (e-mail: whzeng@xmu.edu.cn).

Yangbin Lin is with the Department of Computer Science, Xiamen University, Xiamen, Fujian, China (e-mail: yblin.xmu@qq.com).

Defu Zhang is with the Department of Computer Science, Xiamen University, Xiamen, Fujian, China (e-mail: dfzhang@xmu.edu.cn).

Copyright (c) 2012 IEEE.

solution when there does not exist a feasible solution $\mathbf{x}' \in S$ such that $\mathbf{x}' \succ \mathbf{x}$. $F(\mathbf{x}')$ is then called a *Pareto optimal vector*. Denote the *Pareto set* (PS) as the set of all the Pareto-optimal solutions. A set which contains all the Pareto-optimal objective vectors is called the *Pareto front*, denoted by PF such that $PF = \{F(\mathbf{x}) \in R^m | \mathbf{x} \in PS\}$. A population that is denoted as P is a set of solutions.

The solving of multiobjective optimization problems (MOPs) has been of great use in many diverse areas including engineering, computer science, industry, economics, and physics, etc., therefore, it has attracted much attention. Over the past decades, a lot of MOP algorithms have been proposed, which can be divided into the following categories: traditional algorithms, evolutionary algorithms (EA), memetic algorithms (MA), particle swarm optimization algorithms (PSO), ant colony algorithms (ACA), simulated annealing algorithms (SA), artificial immune systems (AIS), tabu search algorithms (TS), scatter search algorithms (SS) and so on [1].

In these algorithms, some famous methods are based on EA called MOEA. The vector evaluated genetic algorithm (VEGA) [2], multiobjective genetic algorithm (MOGA) [3], niched Pareto genetic algorithm (NPGA) [4], and nondominated sorting genetic algorithm (NSGA) [5] were the early representative algorithms. In 1999, Zitzler and Thiele [6] presented the strength Pareto EA (SPEA), which employs an external archive to store these nondominated solutions previously found. Since the appearance of the SPEA, most researchers have tried to combine an external archive and the current population with their MOEAs. At each generation, a combined population with the external and the current population is first constructed. And the fitness of an individual is evaluated depending on the number of these external nondominated solutions which dominate it. Furthermore, a clustering technique is used to keep diversity. Another version of SPEA (SPEA2) was proposed in [7], which incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method in contrast to its predecessor. There are another two popular algorithms NSGA [5] and its improved version NSGA-II [8]. The NSGA-II is noticeably more efficient than its previous version. It uses a fast nondominated sorting approach to alleviate the computational complexity in the former method and a mating pool is constructed by a selection operator to select the best solutions. NSGA-II tends to spread quickly and appropriately in most cases, but it has some difficulties to find these nondominated vectors that lie in some certain regions of

the search space [1]. In 2000, a famous MOEA called Pareto-Archived Evolution Strategy (PAES) [9] was presented by Knowles and Corne. PAES consists of a (1+1) evolution strategy in combination with a historical archive that stores some nondominated solutions previously found [1]. And a novel method which consists of a crowding procedure is adopted to maintain diversity.

In 1989, Moscato [10] proposed a memetic algorithm, which combines the local search heuristics with a population-based strategy. Some other researches also combine the local search with evolutionary multiobjective optimization (EMO) to avoid the EMO algorithm converging too fast to the Pareto-optimal front. IM-MOGLS was the first multiobjective genetic local search algorithm which was proposed by Ishibuchi [11] in 1998. This algorithm uses a weighted sum of multiple objectives as a fitness function which is adopted when a pair of parents is selected for generating a new solution by the crossover and mutation operations. And a local search is then employed to each solution generated by genetic operations for maximizing its fitness value. IM-MOGLS was further improved by Jaszkiewicz [12] on the multiple objective 0/1 knapsack problem. The algorithm IM-MOGLS is then modified by Ishibuchi *et al.* [13] by means of selecting only the good solutions as the initial solutions for the local search and specifying an appropriate local search direction to each initial solution. Experimental results proved that the modified algorithm called IM-MOGLS2 is better than IM-MOGLS in the flow shop scheduling problem. Ishibuchi and Narukawa [14] proposed a method called S-MOGLS, which is in fact a hybrid algorithm of NSGA-II and local search. In 2002, Jaszkiewicz [15] designed a new genetic local search algorithm called MOGLS-J-MOGLS. It draws a utility function randomly and a temporary population is constructed by the best solutions from the prior generated solutions. Then it recombines a pair of solutions chosen from the above temporary populations at random and applies the procedure of local search only to the new solutions recombined. Experimental results indicated that the algorithm outperformed other genetic local search methods on the problem of multiobjective travelling salespersons. Another new local search method called C-MOGLS which combines cellular multiobjective genetic algorithm with neighborhood search as a local search was proposed in [16]. It uses a weighted sum of multiple objectives as fitness function in each generation. Knowles and Corne [17] proposed an algorithm called M-PAES, which uses a local search method applied in the PAES [9] and combines it with the use of population and recombination. In 2006, a new local search algorithm called Pareto descent method (PDM) was presented by Harada *et al.* [18]. This method seeks Pareto descent directions and moves solutions to this direction for improving all objective functions. Sindhya *et al.* [19] presented a hybrid algorithm which is a combination of the EMO and the local search, and an augmented achievement scalarizing function (ASF) is solved using the latter. The local search utilizes a reference point which is in fact an offspring solution. The target of the local search is to minimize the augmented ASF to get a locally Pareto-optimal solution closest to the reference point. In

[20], Sindhya *et al.* modified the algorithm in [19] by changing the crowding distance in NSGA-II to a clustering technique. In [21], a local search method using a convergence acceleration operator (CAO) is adapted to the best solutions found in each generation and the best solutions are mapped into the decision variable space. Additionally, two leading MOEAs: NSGA [5] and SPEA [6] are combined with the CAO to deal with some test problems. A biased neighborhood structure is used in the local search in [22]. In 2010, Lara *et al.* [23] proposed a multiobjective memetic algorithm, in which the hill climber with sidestep is adopted to move toward and along the Pareto set. In [24], a combination of the multiobjective evolutionary algorithm and a local search method based on the use of rough set theory to solve the constrained multi-objective optimization problems is designed. Bosman [25] made use of gradient information in MOPs. And a gradient-based optimization algorithm called CORL is used for numerical multiobjective optimization. Some other algorithms based on local search are published in [26] - [32].

In recent years, MOEA/D [33] that is based on decomposition, is becoming a very popular method used to solve MOPs. It decomposes a MOP into a number of scalar optimization sub-problems and optimizes each of them simultaneously by using information mainly from its several neighboring sub-problems. In 2009, a new version of MOEA/D based on differential evolution (MOEA/D-DE) [34] was proposed by Li and Zhang. They compared their algorithm with NSGA-II/DE which is an improved version of NSGA-II by the same reproduction operators and the results have shown that MOEA/D-DE could outperform NSGA-II/DE in some continuous multiobjective optimization test instances. In [35], each subproblem is optimized by the greedy randomized adaptive search procedure (GRASP). In [36], Zhang *et al.* suggested a new algorithm called MOEA/D-EGO to deal with the expensive multiobjective optimizations, which are able to produce reasonably good solutions within a given budget on computational cost. MOEA/D-EGO decomposes a MOP into a number of subproblems and adopts a predictive model based on these points evaluated so far to each subproblem. Other latest versions of MOEA/D are published in [37] - [41].

Besides the above-mentioned approaches, many efforts about the research of evolutionary metaheuristics have been made to deal with MOPs, such as PSO [42] - [47], DE [48] - [58], AIS [59] - [61], SA [62] - [65], SS [66,67] and ACA [68] - [70]. For reason of space, only the studies of DE are given briefly as follows: In [53], Abbass and Sarker proposed a novel Pareto differential evolution (PDE) algorithm to solve MOPs. In [55], Wang and Cai presented an improved CW method [54] and called it as CMODE. CMODE adopts a differential evolution and a novel infeasible solution replacement strategy based on multiobjective optimization to solve the constrained optimization problems. In [56], a new algorithm called DEMOSA is proposed, in which a differential evolution is adapted with self-adaptation. Ali *et al.* [57] extended their modified differential evolution (MDE) [58] used for solving single objective optimization problems and then generalized it to MOPs. Their algorithm is called multiobjective differential

evolution algorithm (MODEA). MODEA adopts opposition based learning (OBL) to produce an initial population. Furthermore, they incorporated the random localization concept in the mutation step. Finally, a new selection mechanism is introduced to produce the Pareto-optimal front.

II. NONDOMINATED SORTING AND LOCAL SEARCH BASED ALGORITHM: NSLS

In this paper, a novel algorithm called nondominated sorting and local search based algorithm (NSLS) is presented. The proposed algorithm maintains a population with size of N . At each iteration, given a population P_t , where t denotes the generation to which the population belongs; a simple local search method is used to get a better population P'_t , and then the nondominated sorting with the farthest-candidate approach is adopted on $P_t \cup P'_t$ to obtain a new population for the next iteration.

A. Local Search Method

1) The proposed local search method

Local search is a heuristic method for solving hard optimization problems. It repeatedly tries to improve the current solution by replacing it with a neighborhood solution. As it is known, the neighborhoods of a solution $\mathbf{x}(x_1, x_2, \dots, x_n)$ (where n is the number of variables) in the search space are unlimited. So an effective strategy to find an appropriate neighborhood is the key process in the local search. The proposed local search schema is given in the following.

Given a population P_t with size of N solutions and a solution $\mathbf{x}_{i,t}(x_{1,i,t}, x_{2,i,t}, \dots, x_{n,i,t})$ in P_t , where n denotes the number of variables, i denotes the i th solution of the population and t denotes the generation to which the population belongs. Define $S_{k,i,t}$ as the set of neighborhoods on the k th variable of solution $\mathbf{x}_{i,t}$, namely:

$$S_{k,i,t} = \{\mathbf{w}_{k,i,t}^+, \mathbf{w}_{k,i,t}^-\} \quad (2)$$

$$\mathbf{w}_{k,i,t}^+ = \mathbf{x}_{k,i,t} + c \times (\mathbf{u}_{k,i,t} - \mathbf{v}_{k,i,t}) \quad (3)$$

$$\mathbf{w}_{k,i,t}^- = \mathbf{x}_{k,i,t} - c \times (\mathbf{u}_{k,i,t} - \mathbf{v}_{k,i,t}) \quad (4)$$

where $\mathbf{w}_{k,i,t}^+$ and $\mathbf{w}_{k,i,t}^-$ are denoted as the two neighborhoods of the solution $\mathbf{x}_{i,t}$, $\mathbf{u}_{i,t}(u_{1,i,t}, \dots, u_{k,i,t}, \dots, u_{n,i,t})$, $k \in \{1, \dots, n\}$ and $\mathbf{v}_{i,t}(v_{1,i,t}, \dots, v_{k,i,t}, \dots, v_{n,i,t})$, $k \in \{1, \dots, n\}$ are two solutions randomly chosen from the population P_t , and c is a perturbation factor following a Gaussian distribution $N(\mu, \sigma^2)$, μ and σ are the mean value and the standard deviation of the Gaussian distribution, respectively. The use of Gaussian perturbation is originally adopted in evolution strategies (ES) in [71, 72]. The main mutation mechanism of ES is changing each variable of the solution by adding random noise drawn from a Gaussian distribution as follows:

$$x'_i = x_i + N(0, \sigma^2) \quad (5)$$

where x_i is denoted as the variable of the solution, x'_i is the new value of x_i by performing a mutation operator and σ is called as mutation strengths. The use of the perturbation factor in the proposed local search schema is a little different from that in ES. The main differences boil down to two points as follows:

- 1) The way of perturbation is different. In NSLS, the value of perturbation factor following $N(\mu, \sigma^2)$ is multiplied by the value of $(\mathbf{u}_{k,i,t} - \mathbf{v}_{k,i,t})$ for giving disturbance to the differential value of two individuals, whereas in ES, the

perturbation value is added to the value of the variable directly for generating a new child.

- 2) The setting of σ is different too. In NSLS, the value of σ is constant, whereas in ES, σ is varied on the fly by some rules such as the famous 1/5th rule in the (1+1)-ES [72] and the self-adaptation method in [73]. The mutation strengths σ is coevolving with the solution and it is also mutated into σ' . The reader may refer to [73] for more information about how to mutate the mutation strength σ .

In fact, the usage of the perturbation factor in the proposed local search schema does not play a critical role but has a little effect of improving the performance of the proposed algorithm. The computational results in Section III-F1 will demonstrate the opinion.

Considering the settings of u and σ in NSLS, for u , it is obvious that $u \geq 0$, because that there are two neighborhoods generated from two opposite directions. Moreover, if u is set too big, it would amplify the value of $(\mathbf{u}_{k,i,t} - \mathbf{v}_{k,i,t})$, which would cause a violent perturbation that affects the convergence of the algorithm. If u is set too small, the value of $c \times (\mathbf{u}_{k,i,t} - \mathbf{v}_{k,i,t})$ would be too small which would slow down the convergence of the algorithm. For σ , it is obvious that $\sigma \geq 0$. Moreover, if σ is set too big, the value of c is instable leading to generating a too big or a too small value of c , which would affect the convergence of the algorithm. On the contrary, if σ is set too small, the effect of perturbation would not be obvious enough. Therefore, in this paper, μ and σ are given a range of $[0,1]$. It is worth mentioning that we have chosen reasonable ranges of μ and σ , and have not made any effort in finding the best range. The settings of the two parameters will be discussed in Section III-C. The effect of the usage of perturbation factor is demonstrated in Section III-F1. Furthermore, the sensitivities of the performances to μ and σ are tested in Section III-I. According to (2), the neighborhood space can be divided into $2n$ sub-neighborhoods.

At each generation of the local search process, the set of neighborhoods on the k th variable of solution $\mathbf{x}_{i,t}$ is generated according to (2) - (4). Next, to decide whether the two neighborhood solutions can replace the current solution, a replacement strategy is presented. Suppose that there are two solutions: the neighborhood solution \mathbf{w} and the current solution \mathbf{x} , while comparing the two solutions, there are three possibilities:

- \mathbf{w} dominates \mathbf{x} ($\mathbf{w} > \mathbf{x}$)
- \mathbf{w} is dominated by \mathbf{x} ($\mathbf{w} < \mathbf{x}$)
- \mathbf{w} and \mathbf{x} are incomparable ($\mathbf{w} \sim \mathbf{x}$)

The replacement strategy is given as follows:

1. If $\mathbf{w}_{k,i,t}^+ > \mathbf{x}_{i,t}$ and $\mathbf{w}_{k,i,t}^- > \mathbf{x}_{i,t}$
Randomly choose one to replace $\mathbf{x}_{i,t}$
2. Else if either $\mathbf{w}_{k,i,t}^+ > \mathbf{x}_{i,t}$ or $\mathbf{w}_{k,i,t}^- > \mathbf{x}_{i,t}$
 $\mathbf{x}_{i,t}$ is replaced by the one that dominates it
3. Else if $\mathbf{w}_{k,i,t}^+ \sim \mathbf{x}_{i,t}$ and $\mathbf{w}_{k,i,t}^- \sim \mathbf{x}_{i,t}$
Randomly choose one of them to replace $\mathbf{x}_{i,t}$
4. Else if $\mathbf{w}_{k,i,t}^+ \sim \mathbf{x}_{i,t}$
Replace $\mathbf{x}_{i,t}$ by $\mathbf{w}_{k,i,t}^+$
5. Else if $\mathbf{w}_{k,i,t}^- \sim \mathbf{x}_{i,t}$

Replace $\mathbf{x}_{i,t}$ by $\mathbf{w}_{k,i,t}^-$

6. Else

Do nothing

7. End if

It is obvious that replacing the current solution $\mathbf{x}_{i,t}$ with any neighborhood solution that dominates $\mathbf{x}_{i,t}$ would help the current solution converge to the Pareto-optimal front. If $\mathbf{w}_{k,i,t}^+ \succ \mathbf{x}_{i,t}$ or $\mathbf{w}_{k,i,t}^- \succ \mathbf{x}_{i,t}$, one neighborhood solution is also chosen to replace $\mathbf{x}_{i,t}$. The reason is that it can enhance the diversity of these nondominated solutions found so far.

As a result, a new population P_t' is created by performing the local search schema on the current population P_t . All solutions in the new population are as good as or better than their counterparts in the current population.

The experimental results in section III-F will show that this simple method can get quite good convergence. The pseudo code of the proposed local search algorithm is given as follows.

Pseudo Code of Local Search on P_t

For $i = 1$ to N **do** // N is the number of population size

For $k = 1$ to n **do** // n is the number of variables

Calculate $c = N(\mu, \sigma^2)$;

Randomly choose two solutions $\mathbf{u}_{i,t}$ and $\mathbf{v}_{i,t}$ from current set;

Generate two neighborhood solutions $\mathbf{w}_{k,i,t}^+$ and $\mathbf{w}_{k,i,t}^-$ by (3) and (4), respectively.

Replace $\mathbf{x}_{i,t}$ according to the proposed replacement strategy.

End For

End For

2) More discussion about the proposed local search schema

To discuss the feasibility of the proposed local search schema, the differential evolution (DE) is used for comparison. We first introduce the basic DE briefly, and then discuss these superiorities of the proposed local search schema over the general DE.

1. The basic concept of DE

DE maintains a population of N solutions: $\mathbf{x}_{i,t}$, $i = 1, \dots, N$ at each generation of the algorithm, where the index i denotes the i th solution of the population and t denotes the generation number of the algorithm. DE contains three main operators: mutation, crossover, and selection. The mutation operation is defined as follows:

$$z_{k,i,t} = x_{k,t}^1 + c * (x_{k,t}^2 - x_{k,t}^3) \quad k \in \{1, \dots, n\} \quad (6)$$

where $\mathbf{x}_t^1(x_{1,t}^1, x_{2,t}^1, \dots, x_{n,t}^1)$, $\mathbf{x}_t^2(x_{1,t}^2, x_{2,t}^2, \dots, x_{n,t}^2)$, and $\mathbf{x}_t^3(x_{1,t}^3, x_{2,t}^3, \dots, x_{n,t}^3)$ are three different solutions chosen randomly from the current population, respectively. $\mathbf{z}_{i,t}(z_{1,i,t}, z_{2,i,t}, \dots, z_{n,i,t})$ is the perturbed solution. c is a control parameter, which is set in $(0, 2]$ in Storn and Price [24]. Once $\mathbf{z}_{i,t}$ is generated by (6), a trial solution $\mathbf{w}_{i,t}(w_{1,i,t}, w_{2,i,t}, \dots, w_{n,i,t})$ is produced by the crossover operation which is defined as follows:

$$w_{j,i,t} = \begin{cases} z_{j,i,t}, & rand_j \leq CR \vee j = CJ \\ x_{j,i,t}, & otherwise \end{cases} \quad (7)$$

where $j = 1, \dots, n$, n is the number of variables, $CJ \in \{1, \dots, n\}$ is a random index, and $CR \in [0, 1]$ is the crossover

rate. To decide whether or not the trial solution $\mathbf{w}_{i,t}$ replaces the target solution entering the next generation, the selection operation is utilized and it is defined according to the following equation:

$$\mathbf{x}_{i,t+1} = \begin{cases} \mathbf{w}_{i,t}, & \mathbf{w}_{i,t} \succ \mathbf{x}_{i,t} \\ \mathbf{x}_{i,t}, & otherwise \end{cases} \quad (8)$$

2. The superiority of the proposed local search schema

Notice that if the second neighborhood solution $\mathbf{w}_{k,i,t}^-$ is removed from (2). The remaining solution $\mathbf{w}_{k,i,t}^+$ in (3) is very similar to the mutation operation of DE in (6). The difference between them is that the first item on the right side of (3) is the variable value of the current solution whereas that of (6) is the variable value of the random solution.

Besides the above difference, there are other three advantages of the proposed local search schema over DE.

- 1) For a solution, selection in DE takes place after all variables of the solution are handled whereas in the proposed local search schema, the replacement strategy is employed after just one variable of the solution is handled. In other words, the proposed local search schema has a higher complexity, which may lead to a faster convergence than DE.
- 2) During the searching process, the proposed local search schema seeks two directions in search of a better solution whereas DE seeks only one direction, which reveals that the proposed local search schema has more candidates than DE. In other words, the proposed local search schema has a higher chance to find a better solution than DE. The computational results in Section III-F2 will support this conclusion.
- 3) The proposed local search schema does not have a crossover operation whereas DE has, which makes the proposed one easy to be implemented and extended.

B. Diversity Preservation

In MOPs, it is desired that an algorithm maintains a good spread of solutions in the nondominated solutions as well as the convergence to the Pareto-optimal set. In NSGA-II [8], a crowded comparison mechanism is adopted to maintain the spread of solutions. Given a solution, an estimation of density is designed to evaluate the density of solutions surrounding it. It calculates the average distance of two solutions on either side of this solution along each of the objectives, and then the sum of above obtained distance values corresponding to each objective is calculated as the overall crowding distance value of the given solution. After all population members are assigned a crowding distance value, a crowded comparison operator is designed to compare them for guiding the selection process towards a uniformly distributed Pareto-optimal front [8]. However, this method is unable to get a good spread result under some situations. Fig. 1 shows a particular case, in which most points are very close to each other whereas the others are not. In this case, we are going to select eight points from sixteen points in line (a). These points in line (c) are the selection result of the crowded comparison mechanism. It is obvious that the point of the high density space has a low chance to be selected so that the spread is not good enough.

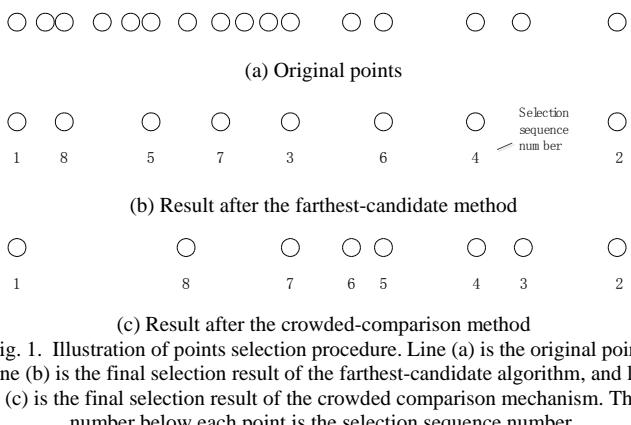


Fig. 1. Illustration of points selection procedure. Line (a) is the original points; line (b) is the final selection result of the farthest-candidate algorithm, and line (c) is the final selection result of the crowded comparison mechanism. The number below each point is the selection sequence number.

The crowded comparison mechanism has been used for diversity maintenance of the obtained solutions in many MOEAs. Additionally, there are several studies, which slightly modified the original crowding distance in [74] - [78]. In [77], Kukkonea proposed an improved pruning of nondominated solutions for bi-objective optimization problems. This method removes the solution that has the smallest crowding distance value one by one and recalculates the crowding distance value after each removal until the number of the remaining solutions is equal to the population size. In [78], a fast and effective method which is based on crowding distance using the nearest neighborhood of solutions in Euclidean sense is proposed for pruning of the nondominated solutions.

In the proposed NSLS, we replace the crowded comparison mechanism in NSGA-II [8] with the farthest-candidate approach that can solve above difficulty to some extent. The suggested method is inspired by the best-candidate sampling algorithm [79] in sampling theory. Suppose that we are going to select K best points from F points, whenever a new point is to be selected, the candidate point in the unselected points which is farthest from the selected points is accepted. Here, the distance is defined by Euclidean Norm. The boundary points (solutions with the smallest and largest function values) are selected first. The algorithm is shown below in Pseudo code, where P_{accept} stores the selected solutions, $D[\mathbf{x}]$ stores the minimum Euclidean distance between \mathbf{x} and the unselected points, and $dis(\mathbf{x}, \mathbf{x}')$ is a function calculating the Euclidean distance between solution \mathbf{x} and \mathbf{x}' . The complexity of this procedure is governed by the selecting procedure. Thus the computational complexity of the proposed algorithm is $O(mF^2)$, where F is the number of the original points and m is the number of the objectives. Because $F \leq 2N$, so the worst-case complexity of the farthest-candidate approach is $O(mN^2)$. The proposed method is good at dealing with above case in Fig. 1 (line (b) is the selection results of the proposed method), from which the farthest-candidate method has a better spread than that of the crowded comparison mechanism. Additionally, the overall crowding distance value in NSGA-II is calculated as the sum of solution distance values corresponding to each objective and the space distance is actually the Euclidean distance. Thus, the computation error

will expand with the increase of the number of objectives. More computational results are shown in Section III.

Pseudo Code of the Farthest-candidate Method

Given a population \mathcal{F} with size of F , and K solutions must be selected from \mathcal{F} ; K is not larger than F .

```

 $P_{accept} = \emptyset$ 
For  $i = 1$  to  $F$  do
     $D[\mathbf{x}_i] = 0$ 
End For
For  $i = 1$  to  $m$  do //  $m$  is the number of objectives.
     $P_{accept} = P_{accept} \cup argmin_{\mathbf{x} \in \mathcal{F}} (f_i(\mathbf{x})) \cup argmax_{\mathbf{x} \in \mathcal{F}} (f_i(\mathbf{x}))$ 
End For
For each  $\mathbf{x}$  in  $\mathcal{F} - P_{accept}$ 
     $D[\mathbf{x}] \leftarrow argmin_{\mathbf{x}' \in P_{accept}} dis(\mathbf{x}, \mathbf{x}')$ 
End For
For  $i = 1$  to  $K - |P_{accept}|$ 
     $\mathbf{x}_1 = argmax_{\mathbf{x} \in (\mathcal{F} - P_{accept})} (D[\mathbf{x}])$ 
    For each  $\mathbf{x}_2$  in  $\mathcal{F} - P_{accept}$ 
         $D[\mathbf{x}_2] \leftarrow min(D[\mathbf{x}_2], dis(\mathbf{x}_1, \mathbf{x}_2))$ 
     $P_{accept} \leftarrow P_{accept} \cup \mathbf{x}_1$ 
End For
End For

```

C. Main Loop

The main framework of NSLS is based on iterations. Each solution has a ranking value that equals to its nondominated level. An initial population P_0 is created randomly. The rank values of all the solutions in P_0 are assigned to zero. Then, the iterations start. We describe the t th iteration of NSLS below, where $t \in [0, T - 1]$.

At the t th iteration, a new population P_t' is created by performing the proposed local search schema on the given population P_t . The fast nondominated sorting approach presented in NSGA-II [8] is adopted on the combined population $P_t \cup P_t'$. The fast nondominated sorting approach tries to rank all the solutions into some nondominated fronts. A naive approach requires $O(mN^3)$ computational complexity whereas NSGA-II [8] improves it to $O(mN^2)$ computation only. Now, there exist some nondominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_g)$. The members of the population P_{t+1} are selected from these nondominated fronts according to their ranking order. Suppose that \mathcal{F}_l , $l \in \{1, 2, \dots, g\}$ is the final nondominated front beyond which no other front can be selected [8]. Usually the number of the solutions in all fronts from \mathcal{F}_1 to \mathcal{F}_l is larger than N , thus, if $l \geq 2$, the members of fronts \mathcal{F}_1 to \mathcal{F}_{l-1} are chosen to the population P_{t+1} firstly. Then the farthest-candidate approach is performed on \mathcal{F}_l to choose the remaining members of the population P_{t+1} . P_{t+1} is used in the next iteration.

Here, we consider the computational complexity of one iteration of NSLS. The worst-case complexities of the basic operations of one iteration are as follows:

- 1) The local search is $O(mNn)$, where n is the number of variables;
- 2) The fast nondominated sorting is $O(mN^2)$;

3) The farthest-candidate procedure is $O(mN^2)$.

In practice, the number of variables (n) is usually smaller than the population size (N), so the worst-case complexity of one iteration of NSLS is dominated by the two parts of the algorithm: the fast nondominated sorting and the farthest-candidate procedure. Therefore, the overall worst-case complexity of NSLS is $O(mN^2)$.

NSLS does not need crossover and mutation operations which are used in NSGA-II. The framework of NSLS only includes a local search schema and a diversity preservation mechanism. Thus, NSLS is simpler than NSGA-II, and can be extended easily with the advanced search methods such as tabu search, simulated annealing, etc. The overall algorithm is shown below in the Pseudo Code of NSLS.

Pseudo Code of NSLS

Initialize the values of P_0 , μ , and σ , where P_0 is the initial population with size of N solutions, and it is created randomly; μ and σ are the mean value and the standard deviation of the Gaussian distribution used in the local search schema, respectively.

For $t = 0$ to $T - 1$ **do**

$P'_t = \text{LocalSearch}(P_t)$

Generate all nondominated fronts of $P'_t \cup P_t$: $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$ by the fast nondominated sorting

Set $P_{t+1} = \emptyset$ and $i = 1$

While $|P_{t+1}| + |\mathcal{F}_i| \leq N$

Take the i th nondominated front into P_{t+1} : $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$

$i = i + 1$

End While

Use the farthest-candidate method to select $N - |P_{t+1}|$ solutions denoted here as W from \mathcal{F}_i

$P_{t+1} = P_{t+1} \cup W$

End For

D. Another version of NSLS: NSLS-C

For comparison, we also introduce another version of NSLS: NSLS-C, which is analogous to the NSLS except that it replaces the farthest-candidate method by the crowded comparison mechanism. The crowded comparison operator \prec_n [8] is used for guiding the selection process at every stage of the algorithm toward a uniformly spread-out Pareto-optimal front. The computational complexity of the crowding distance assignment is $O(m(2N)\log(2N))$. Thus the whole computational complexity of NSLS-C is $O(mN^2)$ too.

chosen but no effort is made in finding the best parameter settings.

TABLE I THE NUMBER OF POINTS IN EACH ZHANG'S PF FILE

Problem	points	Problem	points	Problem	points
ZDT1	1001	LZ07_F3	500	UF4	1000
ZDT2	1001	LZ07_F4	500	UF5	21
ZDT3	269	LZ07_F5	500	UF6	1000
ZDT4	1001	LZ07_F6	990	UF7	1000
ZDT6	1001	LZ07_F7	500	UF9	10000
LZ07_F1	500	LZ07_F8	500	UF10	10000
LZ07_F2	500	LZ07_F9	500		

III. SIMULATION RESULTS AND DISCUSSIONS

In this section, the test problems used in the paper are firstly described and then the performance of NSLS is compared with the performance of NSGA-II, MOEA/D-DE, MODEA, and NSLS-C. For NSGA-II, MOEA/D-DE, and MODEA, the parameter setting is identical with that in their original studies. For NSLS and NSLS-C, a reasonable set of values have been

All the test problems have been executed on Microsoft Window XP Intel core 2 Duo CPU E8400 3.00GHz, with 2GB RAM. NSGA-II, NSLS-C, and NSLS have been implemented in C++. For the MOEA/D-DE, a MOEA/D-DE program is downloaded from <http://dces.essex.ac.uk/staff/qzhang/index.html> to run the problems LZ07s only; a MOEA Framework source is downloaded from <http://www.moeaframework.org/> to run the problems ZDTs and UFs.

TABLE II

THE GDPS-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE PROBLEMS ZDTs WHERE $N = 100$ AND $T = 250$

GDPS value	Problem	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	best	0.006235	0.00274	0.012225	0.001292	5.371e-12
	worst	0.010046	0.004049	0.016819	0.489856	1.707e-11
	mean	0.009049	0.00332	0.014098	0.051332	8.708e-12
	std	0.001039	0.000447	0.00126	0.146178	3.153e-12
MOEA/D-DE	best	0.026727	0.011943	0.055405	0.706405	0
	worst	0.051772	0.032814	0.091642	2.05987	1.479e-06
	mean	0.035442	0.022609	0.073496	1.333412	7.396e-08
	std	0.00302	0.003531	0.001162	0.628968	3.998e-05
NSLS-C	best	0	0	0	3.266e-08	0
	worst	1.435e-05	0	0.001207	1.11662	0
	mean	7.178e-07	0	7.134e-05	0.371856	0
	std	3.128e-06	0	0.000263	0.333303	0
NSLS	best	0	0	0	3.977e-05	0
	worst	9.338e-06	0	0.000373	0.864728	0
	mean	4.669e-07	0	7.068e-05	0.504795	0
	std	2.035e-06	0	0.000103	0.23849	0

A. Test Problems

The proposed algorithm is tested on the test problems chosen from a number of significant past studies in this area. The first five problems: ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 were proposed by Zitzler *et al.* in [80]. The second nine continuous MOPs with complicated PS shapes: LZ07_F1-LZ07_F9 were presented by Li and Zhang in [34]. The third group of the test

problems contains six unconstrained MOPs: UF4, UF5, UF6, UF7, UF9, and UF10 for the CEC 2009 session and competition in Zhang *et al.* [81]. All these problems have no more than three objective functions. None of these problems have any constraints. All objective functions are to be minimized.

TABLE III
THE IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE PROBLEMS ZDTS WHERE $N = 100$ AND $T = 250$

IGD value	Problem	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	best	0.005296	0.005134	0.005447	0.004623	0.01119
	worst	0.006466	0.005806	0.006105	0.11166	0.01498
	mean	0.005742	0.005355	0.005834	0.016547	0.01286
	std	0.000339	0.000202		0.031741	0.001004
MOEA/D-DE	best	0.013761	0.010497	0.014829	0.255471	0.001832
	worst	0.025867	0.025568	0.01988	2.68964	0.001997
	mean	0.017552	0.017808	0.016913	1.08415	0.001883
	std	0.00302	0.003566	0.001508	0.628968	3.397e-05
NSLS-C	best	0.004459	0.004544	0.00506	0.004643	0.002739
	worst	0.005228	0.005324	0.005875	0.666029	0.006498
	mean	0.004751	0.004829	0.005407	0.130192	0.003586
	std	0.000209	0.000219	0.000216	0.15424	0.000907
NSLS	best	0.004205	0.003878	0.004906	0.004243	0.001932
	worst	0.004383	0.003942	0.005073	0.388916	0.00232
	mean	0.004293	0.003896	0.004991	0.159399	0.002074
	std	5.119e-05	1.912e-05	4.611e-05	0.097893	0.000116

TABLE IV
THE IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE PROBLEMS LZ07S WHERE $N = 100$ AND $T = 250$

IGD value	Problem	LZ07_F1	LZ07_F2	LZ07_F3	LZ07_F4	LZ07_F5	LZ07_F6	LZ07_F7	LZ07_F8	LZ07_F9
NSGA-II	best	0.019085	0.085898	0.052964	0.059865	0.042643	0.119071	0.181327	0.209569	0.097119
	worst	0.045131	0.140218	0.09131	0.124654	0.09617	0.359991	0.395309	0.35478	0.19282
	mean	0.029263	0.108861	0.064963	0.091919	0.054114	0.229712	0.282969	0.258117	0.140065
	std	0.006609	0.011322	0.009004	0.017796	0.012929	0.090372	0.054079	0.046411	0.038214
MOEA/D-DE	best	0.004553	0.052656	0.029201	0.029263	0.030843	0.195182	0.029281	0.113412	0.046921
	worst	0.005094	0.157713	0.129526	0.074993	0.112538	0.340329	0.280379	0.26001	0.205673
	mean	0.00476	0.090646	0.075068	0.042132	0.048535	0.255736	0.109507	0.183999	0.095038
	std	0.000173	0.02933	0.033389	0.009376	0.023521	0.050595	0.069426	0.036095	0.046724
NSLS-C	best	0.005041	0.005517	0.005303	0.004793	0.005844	0.077019	0.039913	0.083839	0.005969
	worst	0.005827	0.006248	0.007437	0.005599	0.008116	0.089887	0.148537	0.148783	0.007219
	mean	0.005388	0.005866	0.006374	0.005191	0.006578	0.083489	0.081343	0.11632	0.006424
	std	0.000211	0.00021	0.000698	0.000191	0.000601	0.003626	0.025964	0.016344	0.000358
NSLS	best	0.004446	0.00422	0.005948	0.004494	0.005463	0.059841	0.042259	0.088495	0.0047
	worst	0.004718	0.005204	0.008484	0.004874	0.007493	0.070057	0.111752	0.164614	0.006443
	mean	0.004574	0.004607	0.007279	0.004687	0.006804	0.06408	0.069427	0.113035	0.005318
	std	8.672e-05	0.000247	0.00068	9.728e-05	0.0005	0.002864	0.021701	0.019458	0.000439

TABLE V
THE IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE PROBLEMS UFs WHERE $N = 100$ AND $T = 250$

IGD value	Problem	UF4	UF5	UF6	UF7	UF9	UF10
NSGA-II	best	0.05858	0.240752	0.138172	0.036976	0.161304	0.23361
	worst	0.069894	0.608796	0.338809	0.415512	0.306872	0.480712
	mean	0.062416	0.36923	0.216269	0.192299	0.249898	0.323995
	std	0.002983	0.092296	0.067579	0.139033	0.033644	0.076587
MOEA/D-DE	best	0.177272	0.691898	0.322676	0.02274	0.192771	0.620568
	worst	0.315909	1.14771	0.979561	0.656495	0.359734	1.15759
	mean	0.234135	0.877999	0.462632	0.330092	0.26986	0.833622
	std	0.051439	0.117228	0.147163	0.201341	0.040822	0.129718
NSLS-C	best	0.024694	0.06207	0.029726	0.005617	0.042406	0.277537
	worst	0.032452	0.080449	0.067504	0.009267	0.056581	0.441055
	mean	0.02812	0.069401	0.043991	0.00717	0.048661	0.382326
	std	0.00198	0.004951	0.008541	0.000791	0.003788	0.038892
NSLS	best	0.027071	0.059124	0.031449	0.005091	0.033875	0.163377
	worst	0.032137	0.077806	0.084391	0.010427	0.041698	0.320334
	mean	0.029527	0.070048	0.044968	0.007211	0.037617	0.224449
	std	0.001505	0.0051	0.012421	0.001383	0.001883	0.041402

B. Performance Measures

The performance measure is used to demonstrate the effect of the algorithm for MOPs. There are two goals in a MOP algorithm: (1) the convergence to the Pareto-optimal set, (2) the distribution and diversity of the nondominated solutions. The performance measures can be classified into three categories depending on whether they can evaluate the convergence, the diversity or both [82]. Three performance measures evaluating each type are introduced as the following. All the three performance measures can be used when the Pareto-optimal front that is denoted as P^* is known. Additionally, we introduce the GDPS metric (generational distance on PS), which can measure the performance of the algorithm on the PS.

1) Inverted generational distance (IGD)

The inverted generational distance [83] is usually adopted in this area to compare the performance of the proposed method with other algorithms. Let P^* denote a set of uniformly distributed solutions in the objective space along the PF. P is an approximation to the PF, and it is obtained by the algorithm. The inverted generational distance called IGD-metric is described as follows:

$$IGD(P^*, P) = \frac{\sum_{x \in P^*} \text{mindis}(x, P)}{|P^*|} \quad (9)$$

where $\text{mindis}(x, P)$ is the minimum Euclidean distance between the solution x and the solutions in P . If $|P^*|$ is large enough, the value of IGD-metric can measure both the convergence and diversity of P to a certain extent [34]. A smaller value of $IGD(P^*, P)$ demonstrates a better convergence and diversity to the Pareto-optimal front.

Zhang *et al.* [34] provided some evenly distributed points in PF and let these points be P^* . In the paper, we use these points as the P^* too. The files of these uniformly distributed solutions used in the experiment are downloaded from website <http://dces.essex.ac.uk/staff/qzhang/index.html>. Table I shows the number of these chosen optimal solutions in each problem.

2) Generational distance on PS (GDPS)

The metric calculates the average sum of the Euclidean distance between the nondominated solutions and the corresponding optimal solution on the PS for ZDTs. GDPS-metric can be used, for cases when the Pareto-optimal set is known. The GDPS-metric can be stated as follows:

$$GDPS = \frac{\sum_{x \in S} \text{mindis}(x, PS)}{|S|} \quad (10)$$

where S is an approximation set to the Pareto set (PS), and $\text{mindis}(x, P)$ means the minimum Euclidean distance between x and the PS. According to different PS shape, it might use a different computation algorithm to obtain the value. All problems of ZDTs have the same simple optimal solutions. Their optimal solutions can be denoted as $x_1 \in [0, 1]$, and $x_i = 0, i = 2, 3, \dots, n$. The shape of the PS is a line in the n dimension space. And the variable bounds of x_1 in ZDTs are all between 0 and 1. Therefore, the first variable can be ignored. Thus the $\text{mindis}(x, PS)$ in ZDTs can be calculated briefly as $\text{mindis}(x, PS) = \sqrt{\sum_{i=2}^n x_i^2}$. A smaller value of the GDPS-metric indicates a better convergence of the obtained nondominated solutions on the PS.

3) Generational distance (GD)

The concept of generational distance (GD) was proposed by Veldhuizen and Lamont [84] and it calculates the distance between the nondominated solutions set P got by the algorithm and the Pareto-optimal front P^* . It can be defined as follows:

$$GD = \frac{\sqrt{\sum_{x \in P} \text{mindis}(x, P^*)^2}}{|P|} \quad (11)$$

where $\text{mindis}(x, P^*)$ is the minimum Euclidean distance between solution x and the solutions in P^* . GD measures how far these nondominated solutions from those in the Pareto-optimal set on PF. A smaller value of GD reveals a better convergence to the Pareto-optimal front.

TABLE VI
THE IGD-METRIC VALUES OF THE NONDominated SOLUTIONS FOUND BY EACH ALGORITHM ON THE PROBLEMS LZ07S WHERE $N = 300$ AND $T = 500$

IGD value	Problem	LZ07_F1	LZ07_F2	LZ07_F3	LZ07_F4	LZ07_F5	LZ07_F6*	LZ07_F7	LZ07_F8	LZ07_F9
NSGA-II	best	0.006478	0.05	0.040242	0.050375	0.025206	0.124037	0.120656	0.122274	0.066232
	worst	0.021931	0.116302	0.079456	0.096124	0.05415	0.310339	0.3057	0.25682	0.127667
	mean	0.014508	0.10009	0.053946	0.072001	0.036467	0.232765	0.199369	0.186826	0.100112
	std	0.003786	0.015868	0.008854	0.015922	0.008013	0.069562	0.054044	0.035011	0.014424
MOEA/D-DE	best	0.001325	0.002457	0.002322	0.002143	0.007062	0.027528	0.001527	0.039516	0.002596
	worst	0.001373	0.006004	0.05408	0.006067	0.018008	1	0.009514	0.139633	0.016566
	mean	0.001343	0.003158	0.009568	0.003778	0.011492	0.076932	0.003823	0.060572	0.005062
	std	1.336e-05	0.000906	0.012323	0.001074	0.003156	0.211768	0.002445	0.022581	0.003244
NSLS-C	best	0.001669	0.001818	0.001698	0.001687	0.001795	0.035725	0.022422	0.066525	0.001948
	worst	0.001789	0.001927	0.002759	0.001808	0.002909	1	0.060904	0.127611	0.002179
	mean	0.001713	0.001878	0.002127	0.001741	0.002339	0.519298	0.036966	0.087625	0.002033
	std	3.003e-05	3.18e-05	0.000277	3.137e-05	0.000277	0.480703	0.011042	0.01558	6.398e-05
NSLS	best	0.001431	0.001292	0.001457	0.00143	0.001714	0.024091	0.021218	0.065472	0.001413
	worst	0.001496	0.001447	0.002552	0.001509	0.002849	0.025485	0.075621	0.109408	0.001688
	mean	0.001464	0.001355	0.001881	0.001457	0.002343	0.024579	0.040538	0.083961	0.001508
	std	1.863e-05	3.774e-05	0.000379	2.325e-05	0.00032	0.000332	0.01385	0.011888	6.298e-05

* The population size of LZ07_F6 here is 595.

TABLE VII
THE IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE TEST PROBLEMS UFS WHERE $N = 300$ AND $T = 500$

IGD value	problem	UF4	UF5	UF6	UF7	UF9*	UF10*
NSGA-II	best	0.044584	0.138234	0.122917	0.024132	0.065178	0.223136
	worst	0.047044	0.37662	0.26026	0.403899	0.200914	0.341425
	mean	0.045229	0.241557	0.169269	0.136405	0.119705	0.281976
	std	0.000518	0.062335	0.032849	0.122188	0.039542	0.028729
MOEA/D-DE	best	0.038349	0.225182	0.161557	0.010995	0.058754	0.179197
	worst	0.046251	0.645213	0.789411	0.40958	0.768593	1
	mean	0.042119	0.379599	0.388096	0.111664	0.139723	0.469093
	std	0.001598	0.123046	0.223219	0.135332	0.149824	0.184069
NSLS-C	best	0.013809	0.056334	0.027811	0.002269	0.019846	0.128121
	worst	0.016578	0.066852	0.065604	0.004606	0.768593	0.403018
	mean	0.014864	0.06103	0.035066	0.003197	0.058482	0.318279
	std	0.000841	0.002924	0.008439	0.000646	0.162913	0.066412
NSLS	best	0.013151	0.055913	0.026787	0.001817	0.014313	0.096121
	worst	0.01541	0.06498	0.038491	0.004312	0.015215	0.156611
	mean	0.014261	0.059321	0.032427	0.00278	0.014616	0.133543
	std	0.000593	0.002402	0.00342	0.000578	0.000236	0.017772

* The population size of UF9 and UF10 here are 595.

4) Spread (Δ)

The spread metric Δ proposed by Wang *et al.* [82] measures the spread effect of the nondominated solutions obtained by the algorithm. Δ is not only used to measure bi-objective problems but also suitable for problems that have more than two objectives. Δ can be defined mathematically as follows:

$$\Delta = \frac{\sum_{i=1}^m d(E_i, \Psi) + \sum_{X \in \Psi} |d(X, \Psi) - \bar{d}|}{\sum_{i=1}^k d(E_i, \Psi) + (|\Psi| - m)\bar{d}} \quad (12)$$

$$d(X, \Psi) = \min_{Y \in \Psi, Y \neq X} \|F(X) - F(Y)\| \quad (13)$$

$$\bar{d} = \frac{1}{|\Psi|} \sum_{X \in \Psi} d(X, \Psi) \quad (14)$$

where Ψ is a set of nondominated solutions, m is the number of objectives and (E_1, \dots, E_m) are m extreme solutions in the set of true Pareto front (PF). A smaller value of Δ demonstrates a better spread and diversity of the nondominated solutions.

5) Wilcoxon signed ranks test

Besides the above performance measures, non parametric Wilcoxon signed ranks test [85] is adopted to analyze the performance of the proposed algorithm with the other algorithm statistically. It is used for finding a significant difference between the behaviors of two algorithms. The description of Wilcoxon signed ranks test is referred to [85]. The test computations are described briefly in the following.

Suppose that there are h test problems, let d_i denote the difference between the performance scores of the two algorithms on the i th test problem. These ranks of $d_i = 0$ are neglected. Then the rest of these differences are ranked in

ascending order. The sum of positive ranks and negative ranks are denoted as R^+ and R^- , respectively. If there exist equal ranks, these ranks are split evenly. Let h' be the number of test problems for which the difference is not equal to 0. The null hypothesis will be rejected if the test statistic T ($T = \min(R^+, R^-)$) is not greater than the value of the distribution of Wilcoxon for h' degrees of freedom; the normal approximation for the Wilcoxon T statistics is used for getting p -value [85]. We use the SPSS software packages to compute the p -value in the study. The level of significance α is assigned as 0.05, which indicates that if the p -value is smaller than α , there is a significant difference between the two algorithms.

C. Parameter Settings

All the four algorithms have two common parameters: the population size N and the maximum number of generations T . Here, $N = 100$ and $T = 250$. For the other parameters in NSGA-II and MOEA/D-DE, they have remained the same with their original studies [8, 34]. For NSLS-C and NSLS, there are only two special parameters: μ and σ used in the local search that need to be set. Here we set $\mu = 0.5$ and $\sigma = 0.1$ for the experiment. The sensitivities of μ and σ are discussed in Section III-I. In addition, each algorithm is run 20 times independently for each problem. The algorithms stop after a given iteration T .

TABLE VIII
THE IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM IN 20 RUNS WHERE $N = 100$ AND $T = 250$

IGD value	Problem	LZ07_F1	LZ07_F2	LZ07_F6	UF4	UF9
NSLS-WN	mean±std	0.004753±8.492e-05	0.004835±0.000505	0.075878±0.013393	0.040522±0.000371	0.03949±0.002656
NSLS	mean±std	0.004574±8.672e-05	0.00469±0.000497	0.06408±0.002864	0.029527±0.001505	0.037617±0.001883

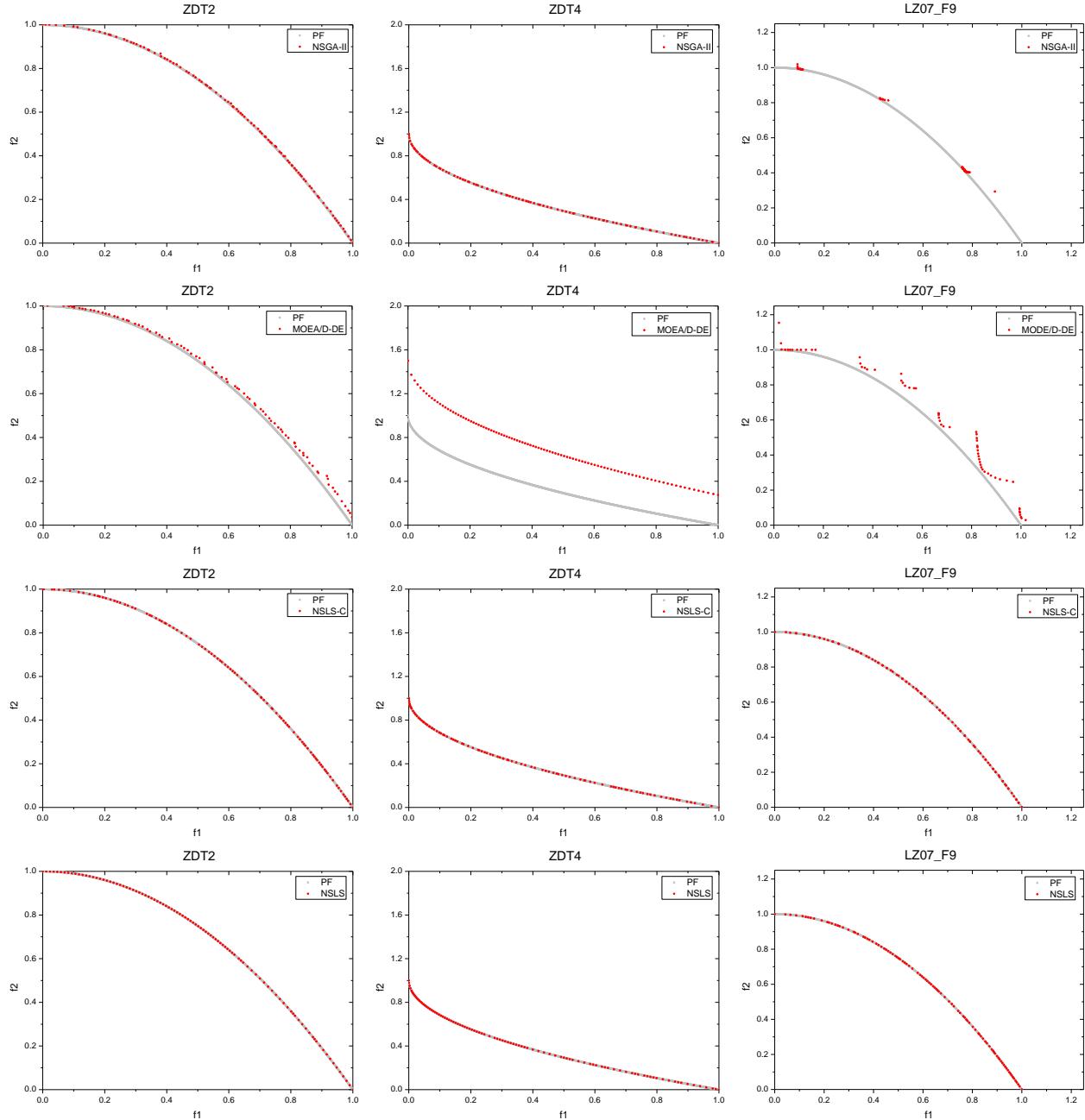


Fig. 2. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on ZDT2, ZDT4, and LZ07_F9 where $N = 100$ and $T = 250$.

D. Discussion of the Results

I) Results of the bi-objective problems of ZDTs

Tables II and III show the best, worst, mean, and standard deviation of the GDPS-metric and IGD-metric values of the 20 populations on ZDTs where $N = 100$ and $T = 250$. The best values among these four algorithms: NSGA-II, MODE/D-DE, NSLS-C, and NSLS are noted with bold font. For illustration, the plots of the nondominated solutions of the problems ZDT2 and ZDT4 with the best IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space are shown in Fig. 2, respectively.

It is clear from Tables II and III that NSLS and NSLS-C perform better than NSGA-II and MOEA/D-DE in all test

problems except in ZDT4 and ZDT6. The GDPS-metric values of NSLS-C and NSLS in Table II are all equal to zero on ZDT2 and ZDT6, which means that NSLS and NSLS-C keep pace with the optimal solutions in the PS whereas other algorithms could not have this ability. ZDT1 is an easy problem in ZDTs. For ZDT1, all the algorithms have a good IGD values except MODE/D-DE. ZDT2 has a nonconvex Pareto-optimal front. It can be seen from Fig. 2 that NSLS and NSLS-C have better convergence than NSGA-II and MOEA/D-DE on ZDT2. In the problem ZDT3, the Pareto-optimal front consists of five disjoint curves that make it difficult. However, it is clear from Tables II and III that the best IGD value is found by NSLS. As mentioned in [8], ZDT4 has many different local Pareto-optimal fronts in the search space, and NSGA-II gets stuck at different local Pareto-optimal sets. NSLS and NSLS-C

also have this problem. However, NSLS performs better than MOEA/D-DE on ZDT4. The performances of the nondominated solutions of ZDT4 got by NSGA-II, NSLS-C, and NSLS are almost similar from Fig. 2, which also shows that the plots of the nondominated solutions of MOEA/D-DE are far away from the PF on ZDT4 whereas that of NSGA-II, NSLS-C, and NSLS converge close to the PF. The last problem ZDT6 in ZDTs, is another difficult problem. NSLS obtains a worse

result than MODE/D-DE on ZDT6, the probable reason for this is the fact that ZDT6 has a thin density of solutions towards the PF and the solutions of the front is nonuniform spread [57], which NSLS gets stuck at. Moreover, the IGD-metric and GDPS-metric values of NSLS are better than NSLS-C on most problems in ZDTs, which reveals that the farthest-candidate method in NSLS is better than the crowded comparison mechanism for enhancing diversity.

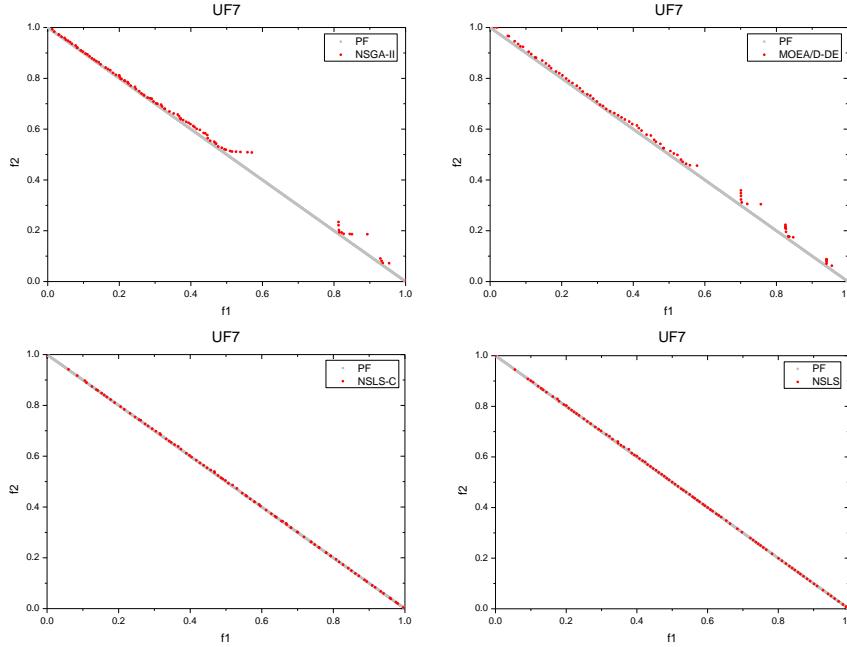


Fig. 3. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on UF7 where $N = 100$ and $T = 250$.

2) Results of the bi-objective problems in LZ07s

These test problems LZ07_F1–LZ07_F9 (except LZ07_F6) are bi-objective problems. LZ07_F1–LZ07_F5 have the same convex PF shape. However, they have various PS shapes in the decision space. The search space of LZ07_F2–LZ07_F5 is $[0,1] * [-1,1]^{n-1}$ but that of LZ07_F1 is $[0,1]^n$. LZ07_F7 and LZ07_F8 are different from LZ07_F1–LZ07_F5 because that there are many local Pareto solutions in LZ07_F7 and LZ07_F8 whereas there are no local PFs in LZ07_F1–LZ07_F5. LZ07_F7 and LZ07_F8 not only have the same PF shape as LZ07_F1, but also have similar PS shapes as that of LZ07_F1. The reader may refer to [34] for more information about these problems.

Table IV shows the best, worst, mean, and standard deviation of the IGD-metric values obtained by the four algorithms on LZ07_F1–LZ07_F9 where $N = 100$ and $T = 250$. It is clear from Table IV that NSLS-C and NSLS both perform better than NSGA-II on all problems. NSLS-C performs better than MOEA/D-DE on most problems except LZ07_F1 and LZ07_F7 in terms of the best IGD-metric value. NSLS outperforms MOEA/D-DE on all problems except that MOEA/D-DE finds a better value in terms of the best IGD-metric value on LZ07_F7. For NSLS-C and NSLS, the latter performs better than the former on most problems. The probable reason for this is the fact that the farthest-candidate approach in NSLS is better at maintaining diversity than the crowded comparison approach in NSLS-C. To give a graphical

overview of the behaviors of these four algorithms, the plots of the nondominated solutions obtained by each algorithm with the lowest IGD values for LZ07_F9 are shown in Fig. 2, from which the nondominated solutions obtained by NSLS achieve the best performance.

3) Results of the bi-objective problems in UFs

These test problems UF4–UF7 are all bi-objective problems. UF5–UF7 have the same search space: $[0,1] * [-1,1]^{n-1}$, but the search space of UF4 is $[0,1] * [-2,2]^{n-1}$. UF4–UF7 have different PF shapes and PS shapes [81]. For more information about these test problems, the reader is referred to [81].

Table V presents the best, worst, mean, and standard deviation of the IGD-metric values obtained by the four algorithms on the problems UF4–UF10 where $N = 100$ and $T = 250$. From Table V, NSLS-C and NSLS perform better than NSGA-II and MOEA/D-DE on all problems. The problem UF7 is chosen to display the nondominated solutions in Fig. 3. The PF shape of UF7 is a continuous line. It is obvious from Fig. 3 that NSLS and NSLS-C are able to find a better uniformly spread of solutions than the other two algorithms. Additionally, though NSLS outperforms NSLS-C on ZDTs and LZ07s above, NSLS performs worse than NSLS-C on UF4–UF7. The probable reason is that some solutions in the population obtained at each iteration in NSLS converge too close to each other so that the effect of diversity selection method is not obvious enough when the population size is set to be 100 only.

The results will be opposite when the population size is increased to be 300 in Section III-E2.

4) Results of the tri-objective problems: LZ07_F6, UF9, and UF10

The three test problems LZ07_F6, UF9, and UF10 are tri-objective problems. It is apparent from Tables IV and V that NSLS outperforms the other three algorithms on all the three

problems. According to [34], when the population size is set to be 100, it is not large enough to illustrate the effect of MOEA/D-DE on these tri-objective problems. Therefore, the nondominated solutions with the best IGD-metric values are not presented in this section. However, they will be shown in Section III-E3 with different parameter settings.

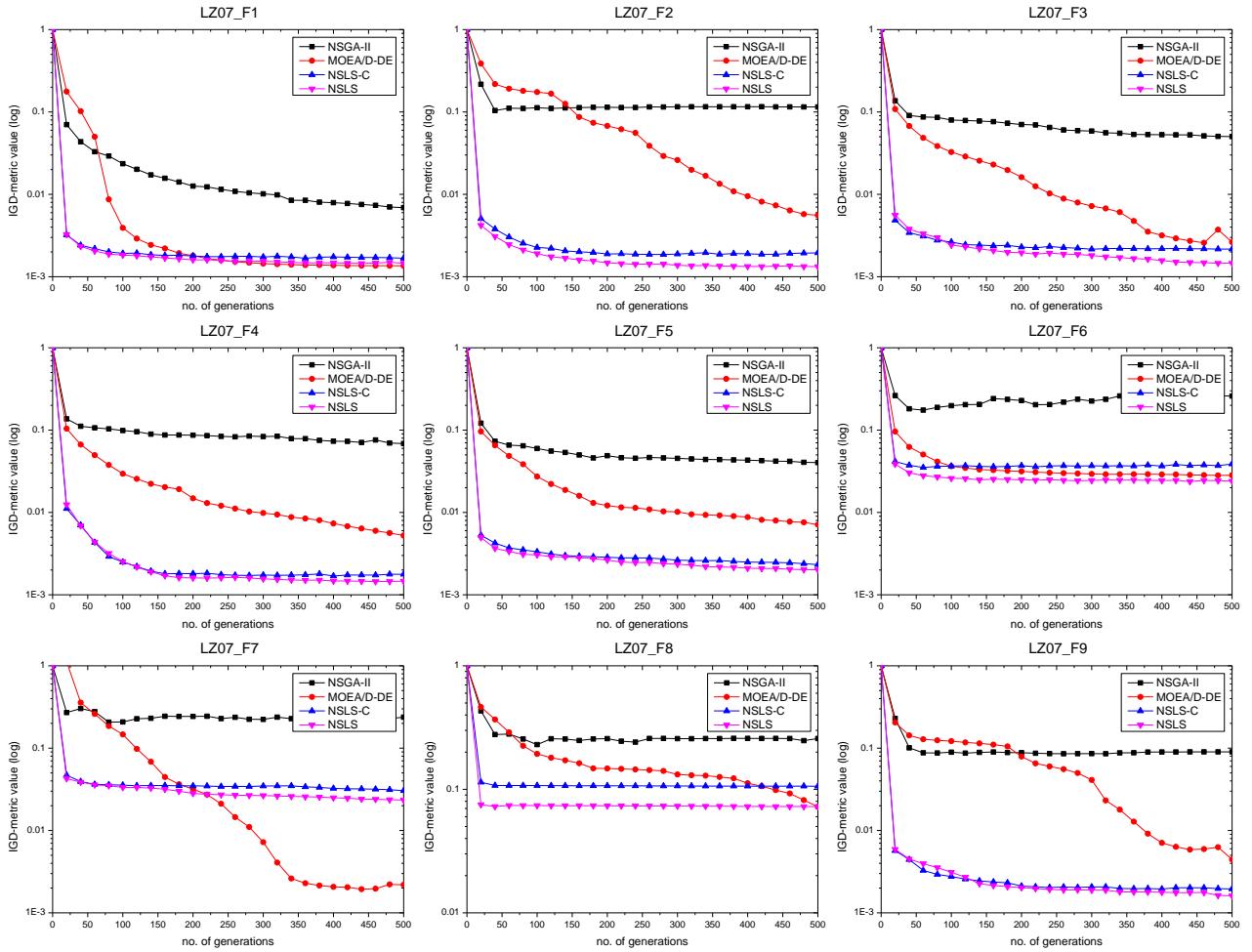


Fig. 4. Evolution of IGD-metric values versus the number of generations.

E. Different Parameter Settings

For more effective persuasive messages of the comparison, additional experiments are performed to show the effect of different values of a couple of parameter settings on the performance of the proposed algorithm. All other parameters are kept as before, but increase the values of N and T . In this experiment, $N = 300$ for the bi-objective problems, $N = 595$ for the tri-objective problems, and $T = 500$ (instead of 250 used before) for all problems.

Increasing the values of N and T will decrease the speed of computation. However, it could enhance the diversity at the same time.

In this experiment, the problems LZ07s and UFs are chosen as the comparison problems. The problems ZDTs are not used in this section. The reason of this is that the problems ZDTs are more simple than LZ07s and UFs. In most papers such as [8] and [57], when comparing the performances of ZDTs, the

values of N and T are usually set to be 100 and 250, respectively. Tables VI and VII present the best, worst, mean, and standard deviation of the IGD-metric values on the problems LZ07s and UFs.

Fig. 4 shows the evolution of IGD-metric values of one random run versus the number of generations in these four algorithms on the test problems LZ07_F1-LZ07_F9. It is clear from Fig. 4 that NSLS-C and NSLS converge faster than NSGA-II and MOEA/D-DE on all problems. However, the fast convergence speed indicates that it is easy to trap into local optima easily. It is also the main reason of NSLS could not find better IGD-metric values than MOEA/D-DE on the test problems LZ07_F1, LZ07_F7, and LZ07_F8 in Section III-E1.

Figs. 5-9 show plots of the nondominated solutions with the best IGD-metric values found by these four algorithms in 20 runs in the objective space on all problems.

It is clear from Tables VI and VII that the IGD-metric values of NSLS-C and NSLS get noticeable improvements when given higher values of N and T . NSLS and NSLS-C converge

closer to the Pareto-optimal fronts and have a much better diversity when $N = 300$ and $T = 500$.

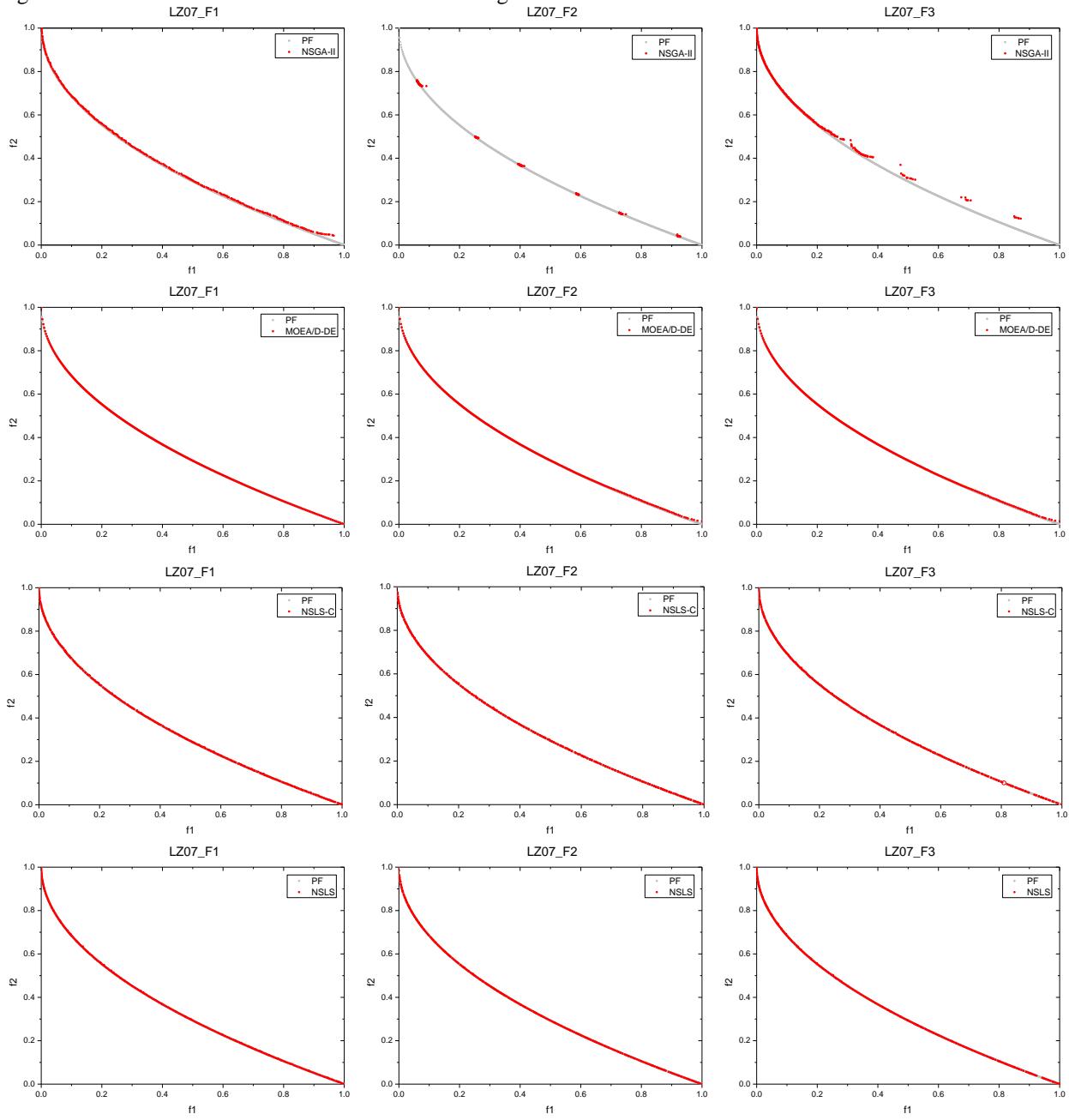


Fig. 5. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on LZ07_F1 - LZ07_F3 where $N = 300$ and $T = 500$.

TABLE IX

THE GD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE BI-OBJECTIVE PROBLEMS OF LZ07S WHERE $N = 300$ AND $T = 500$

GD value	Problem	LZ07_F1	LZ07_F2	LZ07_F3	LZ07_F4	LZ07_F5	LZ07_F7	LZ07_F8	LZ07_F9
NSGA-II	mean \pm	0.000318 \pm	0.000691 \pm	0.000806 \pm	0.001771 \pm	0.000757 \pm	0.001441 \pm	0.003936 \pm	0.000566 \pm
	std	3.213e-05	0.001501	0.000301	0.001383	0.000162	0.001305	0.003335	0.000903
MOEA/D-DE	mean \pm	6.742e-05\pm	0.00037 \pm	0.000476 \pm	0.002128 \pm	0.000816 \pm	0.008018 \pm	0.006189 \pm	0.000541 \pm
	std	6.396e-06	0.000325	0.000416	0.002067	0.000373	0.007956	0.002449	0.000597
MODEA	mean \pm	0.000404 \pm	0.000681 \pm	0.001185 \pm	0.00538 \pm	0.001174 \pm	0.00694 \pm	0.010251 \pm	8.4e-05 \pm
	std	4.365e-05	0.000995	0.00029	0.002595	0.000366	0.009321	0.011449	2.43e-05 \pm
NSLS	mean \pm	0.000152 \pm	0.000102\pm	0.000148\pm	0.000151\pm	0.000153\pm	0.0002\pm	0.003729\pm	5.66e-05\pm
	std	5.054e-06	1.527e-05	8.695e-06	7.784e-06	7.5e-06	0.000138	0.000907	1.589e-06

1) Results of the bi-objective problems in LZ07s

Table VI shows that the IGD-metric values of NSLS-C and NSLS are better than NSGA-II on LZ07s and are better than MOEA/D-DE on the test problems LZ07_F2, LZ07_F3, LZ07_F4, LZ07_F5, and LZ07_F9. Figs. 5-7 display that NSGA-II is not suitable for dealing with the group of test problems LZ07s, the reason of this is that it converges too fast and does not have a uniform distribution on most of these problems. NSLS-C, NSLS, and MOEA/D-DE have a nearly equally good performances on LZ07_F1, LZ07_F2, LZ07_F3, LZ07_F4, and LZ07_F9. When observing carefully from Figs. 5-7, the solutions obtained by NSLS have a better uniform distribution than MOEA/D-DE and NSLS-C on these problems

except LZ07_F1. On the problem LZ07_F5, the nondominated solutions of NSLS and NSLS-C with the lowest IGD-metric values are significantly better than those of MOEA/D-DE in approximating on the PF. From Figs. 6 and 7, the nondominated solutions obtained by NSLS does not spread as uniformly as MOEA/D-DE does on LZ07_F7 and LZ07_F8, the reason is that NSLS and NSLS-C converge too fast so that many solutions gather to one optimal point. The sameness of these three test problems LZ07_F1, LZ07_F7, and LZ07_F8 is that they have the same PF shapes and PS shapes, which NSLS is unable to handle well. In the further study, it is a task to control convergence speed for NSLS on this kind of problems.

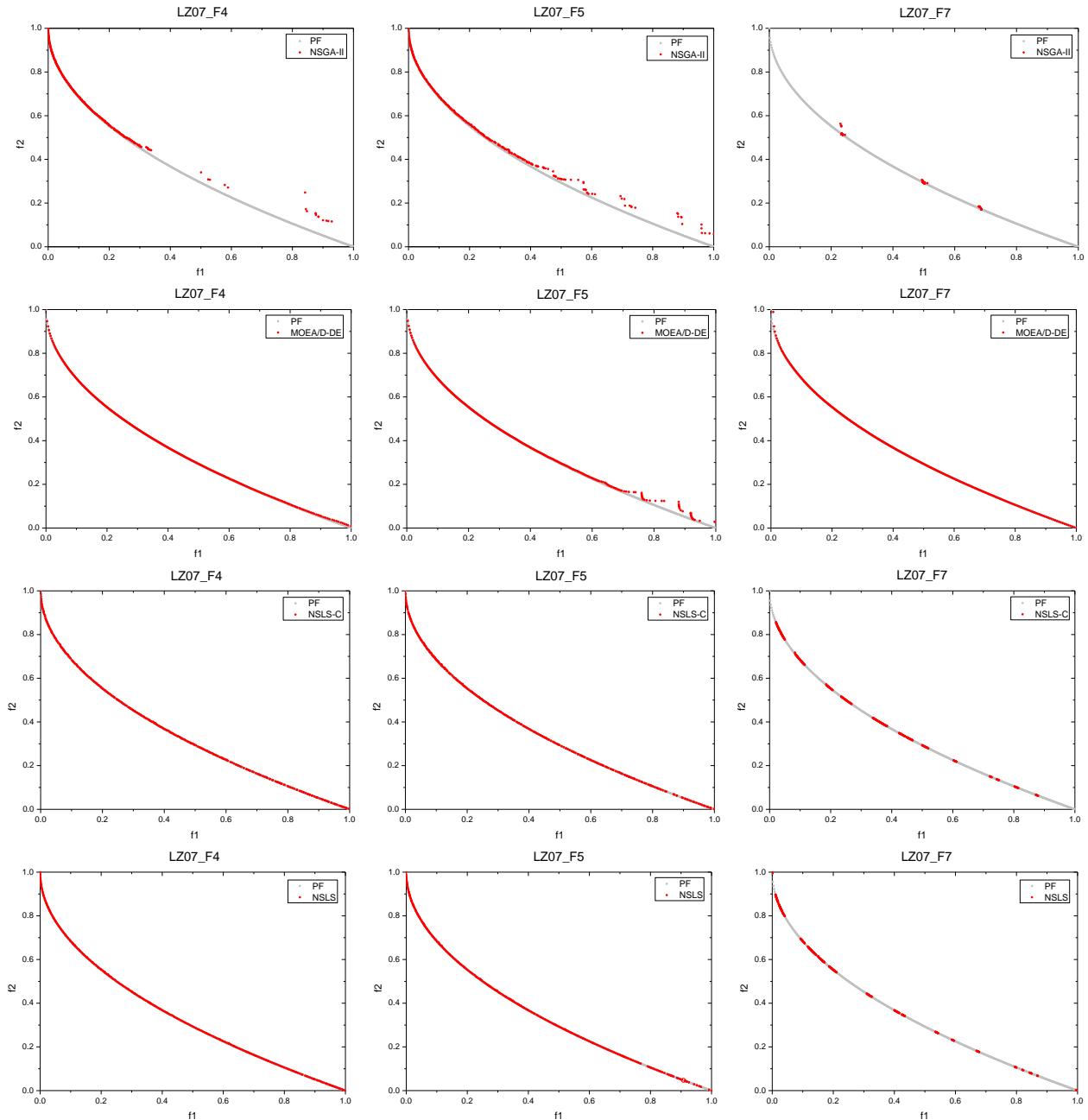


Fig. 6. Plots of the non-dominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on LZ07_F4, LZ07_F5, and LZ07_F7 where $N = 300$ and $T = 500$.

2) Results of the bi-objective problems in UFs

Table VII reveals that NSLS-C and NSLS get much better IGD-metric values than NSGA-II and MOEA/D-DE on all problems of UFs. Figs. 7 and 8 indicate that neither NSGA-II nor MOEA/D-DE could handle UF4, UF5, UF6, and UF7 as well as NSLS and NSLS-C does. It can be observed from Figs. 7 and 8 that the nondominated solutions obtained by NSGA-II and MOEA/D-DE deviate from the PF on the problems UF4, UF5, and UF6 whereas NSLS-C and NSLS find a good approximation to the PF on these three problems. The PF shape of UF5 consists of only 20 Pareto-optimal solutions. The PF shape of UF6 contains one isolated solution and two

disconnected parts. For the problems UF5 and UF6 in Fig. 8, the nondominated solutions obtained by NSLS seem to converge to the PF but cannot achieve a good distribution. The probable reason for this is the fact that the discontinuity of the problems UF5 and UF6. The performance of UF7 handled by NSLS in Fig. 8 may support the point. The PF shape of UF7 is a continuous line. For the problem UF7 in Fig. 8, NSLS and NSLS-C could approximate the PF on UF7 perfectly. Conversely, some solutions obtained by NSGA-II and MOEA/D-DE do not have a good convergence to the PF on UF7.

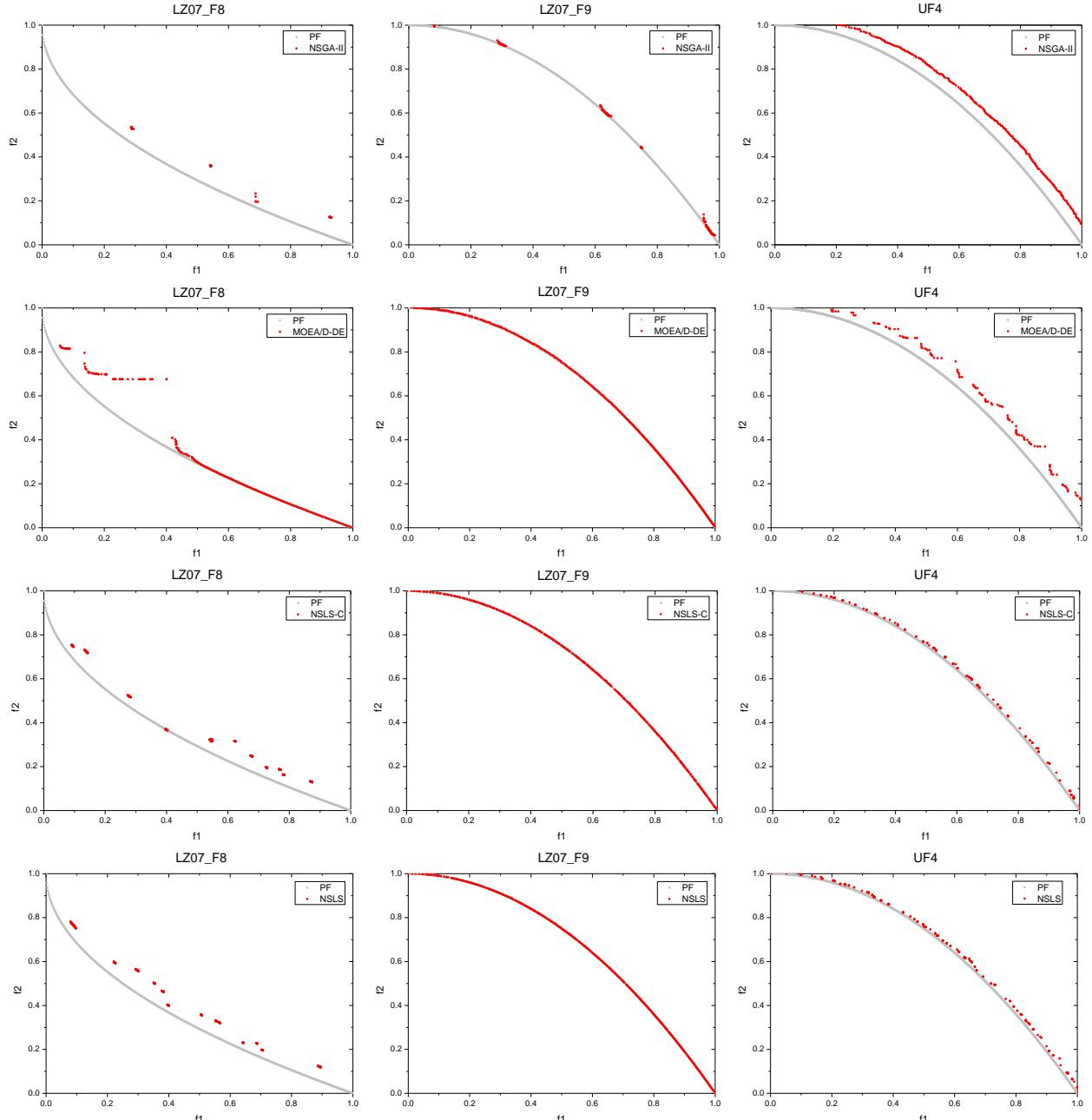


Fig. 7. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on LZ07_F8, LZ07_F9, and UF4 where $N = 300$ and $T = 500$.

3) Results of the tri-objective problems: LZ07_F6, UF9, and UF10

The above discussed problems are all bi-objective problems. In this section, NSLS will be compared with the other three algorithms on the tri-objective problems: LZ07_F6, UF9, and UF10. These three problems have the same search space: $[0,1]^2 * [-2,2]^{n-2}$. The problems LZ07_F6 and UF10 have the

same PS shape and the same PF shape. The PS of UF9 has two disconnected parts and the PF of UF9 has two parts too. It can be observed from Tables VI and VII that the IGD-metric values of NSLS and NSLS-C are better than the other two algorithms on these three problems. It is apparent from Fig. 9 that NSLS and NSLS-C can approximate the PF better than the other two algorithms on these three problems.

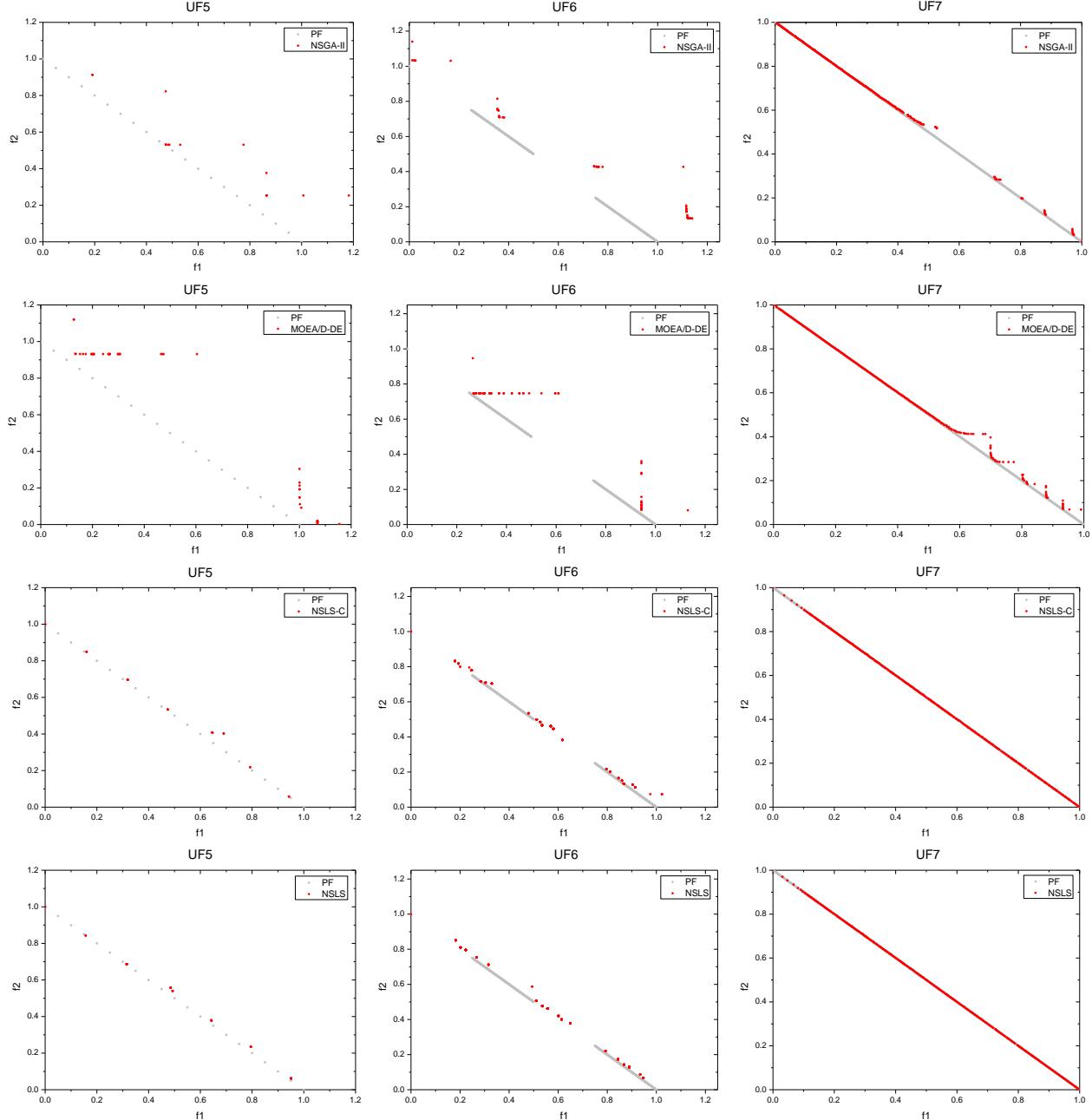


Fig. 8. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on UF5, UF6, and UF7 where $N = 300$ and $T = 500$.

TABLE X
THE GD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE TRI-OBJECTIVE PROBLEMS LZ07_F6, UF9, AND UF10 WHERE $N = 595$ AND $T = 500$

GD value	Problem	LZ07_F6	UF9	UF10
NSGA-II	mean \pm std	0.183068 \pm 0.060713	0.003173 \pm 0.001376	0.23978 \pm 0.236217
MOEA/D-DE	mean \pm std	0.014958 \pm 0.048007	0.013421 \pm 0.0289	0.018737 \pm 0.049341
MODEA	mean \pm std	0.111012 \pm 0.054924	0.0023 \pm 0.001062	0.34001 \pm 0.109281
NSLS	mean \pm std	0.000861\pm1.875e-05	0.000467\pm1.615e-05	0.000457\pm0.000198

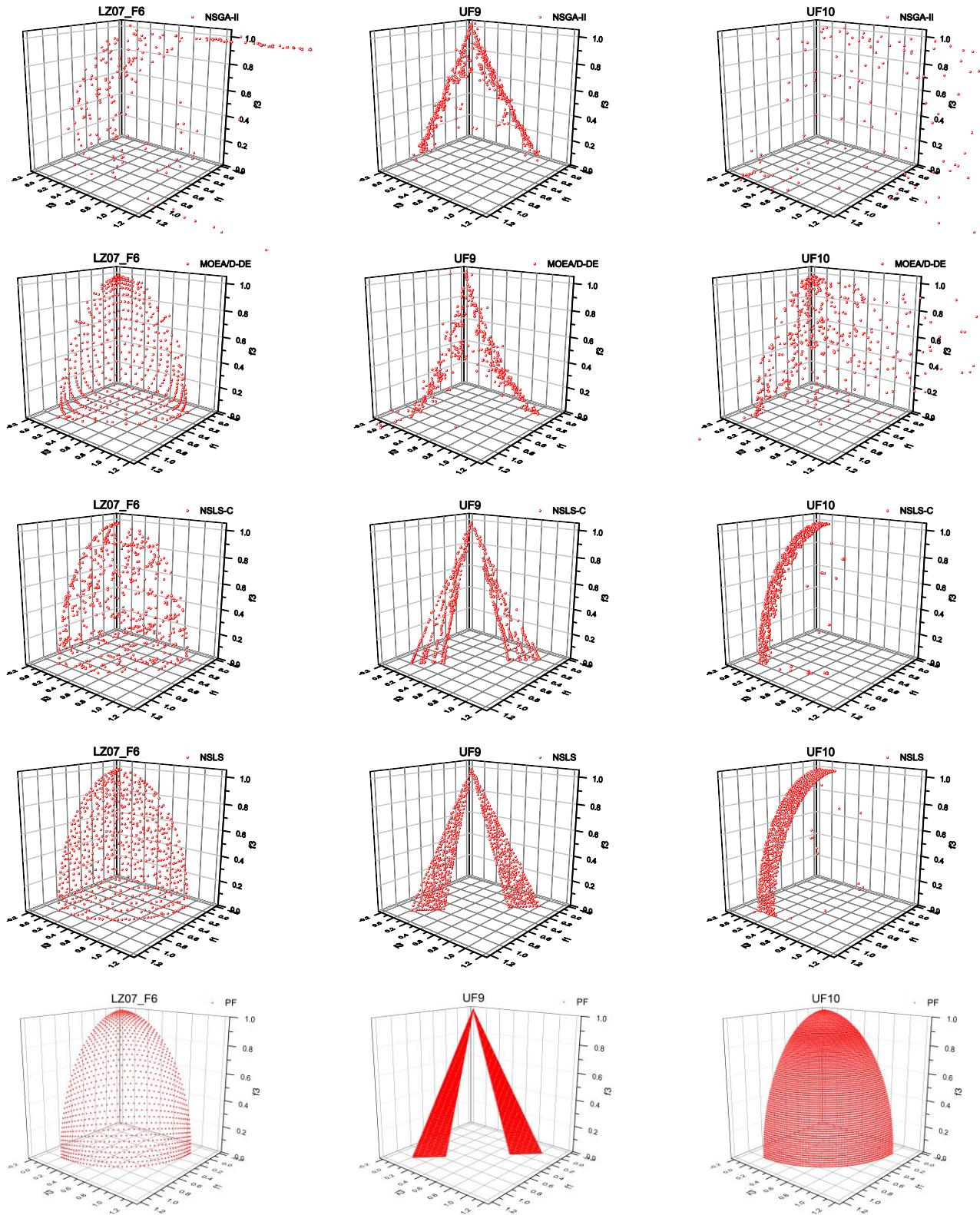


Fig. 9. Plots of the nondominated solutions with the lowest IGD-metric values found by NSGA-II, MOEA/D-DE, NSLS-C, and NSLS in 20 runs in the objective space on LZ07_F6, UF9, and UF10 where $N = 595$ and $T = 500$. And the three images in the last line are the corresponding Pareto-optimal front of LZ07_F6, UF9, and UF10.

TABLE XI
THE Δ -METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE BI-OBJECTIVE PROBLEMS OF LZ07S
WHERE $N = 300$ AND $T = 500$

Δ value	Problem	LZ07_F1	LZ07_F2	LZ07_F3	LZ07_F4	LZ07_F5	LZ07_F7	LZ07_F8	LZ07_F9
NSGA-II	mean \pm std	0.449628 \pm 0.031751	1.04029 \pm 0.152766	0.590548 \pm 0.049037	0.723243 \pm 0.153356	0.523907 \pm 0.080935	1.24461 \pm 0.227475	1.18784 \pm 0.18225	1.0089 \pm 0.279361
	MOEA/D-DE	0.296322 \pm 0.006123	0.409152 \pm 0.091307	0.454606 \pm 0.153432	0.612257 \pm 0.26867	0.460844 \pm 0.088488	1.08671 \pm 0.424313	1.07675 \pm 0.236991	0.304033 \pm 0.137462
NSLS-C	mean \pm std	0.291826 \pm 0.026757	1.02708 \pm 0.062567	0.469953 \pm 0.038744	0.389794 \pm 0.031395	0.422184 \pm 0.031308	1.1111 \pm 0.192615	1.04935 \pm 0.027844	1.10443 \pm 0.06271
	NSLS	0.256672 \pm 0.007605	0.258294 \pm 0.01665	0.272995 \pm 0.0138764	0.249314 \pm 0.007426	0.304147 \pm 0.016828	1.02092 \pm 0.578565	1.03755 \pm 0.285274	0.252957 \pm 0.007939

4) More discussion about NSLS-C and NSLS

In this paper, the proposed NSLS-C and NSLS outperform the other two algorithms on most problems. In addition, NSLS is superior to NSLS-C. The difference of them is the diversification method. The crowded comparison mechanism in NSLS-C calculates the sum of the average distance of each objective rather than the Euclidean distance, thus when the number of objectives increases, the computation error increases too. Moreover, from Tables II-V, when the population size is set to be 100, the performances of NSLS-C and NSLS are almost similar. However, when the population size is increased to be 300 or 595, NSLS finds better results than NSLS-C. It is apparent from Figs. 2-9 that NSLS has a better uniformly spread than NSLS-C, from which we can reach a general conclusion that the effect of the farthest-candidate method is superior to the crowded comparison mechanism with the number of objectives or the number of population size increases.

F. Effect of the Proposed Local search Schema

1) Effect of the use of the perturbation factor c

Here, to prove the effect of the perturbation factor c following the Gaussian distribution $N(\mu, \sigma^2)$, a version of NSLS without using the Gaussian distribution is proposed and we call it as NSLS-WN for convenience. In NSLS-WN, we set $c = 1$ constantly. Some test problems are randomly chosen to show the effect of the Gaussian distribution and the computational results are shown in Table VIII. It is clear from Table VIII that all the IGD-metric values of NSLS are a little better than that of NSLS-WN on all test problems, which reveals that the using of the perturbation factor c is valid.

2) NSLS vs. NSLS-S

As mentioned in Section II-A2, (3) is very similar to the mutation operation of DE in (6). In this section, to verify the superiority of NSLS, a modified version of NSLS: NSLS-S (Nondominated sorting and local search with single neighborhood) is used for comparison. NSLS-S is analogous to NSLS except that it generates only one neighborhood: $\mathbf{w}_{k,i,t}^+$ in the local search schema. Fig. 10 shows the IGD-metric values versus the number of generations for the problem LZ07_F2. It is clear from Fig. 10 that NSLS has a faster convergence than NSLS-S and can achieve a better IGD-metric value than NSLS-S in every value of generations.

It is worth mentioning that there are two candidates generated each time in NSLS, whereas there is only one candidate in NSLS-S, therefore, another comparison test based on the same number of function evaluations is performed. Take

LZ07_F5 as an example, the IGD-metric values versus the number of function evaluations are also shown in Fig. 11, from which it is obvious that NSLS achieves a better result than NSLS-S in every value of function evaluations. The results of the other test problems are the same with that of LZ07_F5. What shows evidence that the proposed local search schema is feasible and efficient.

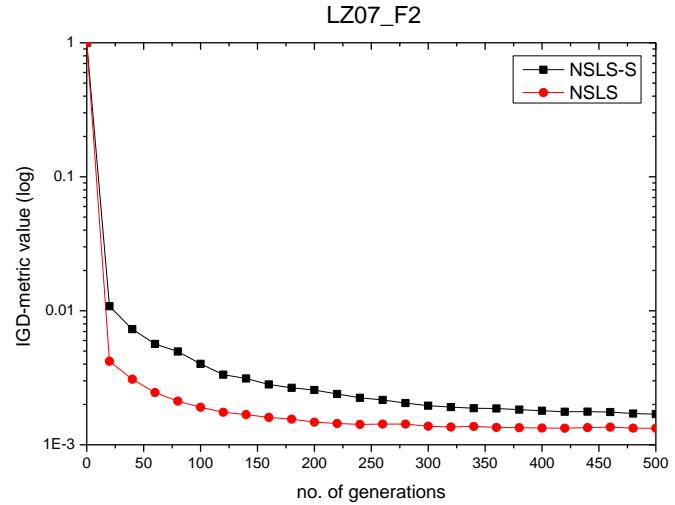


Fig. 10. The IGD-metric values versus the number of generations for LZ07_F2.

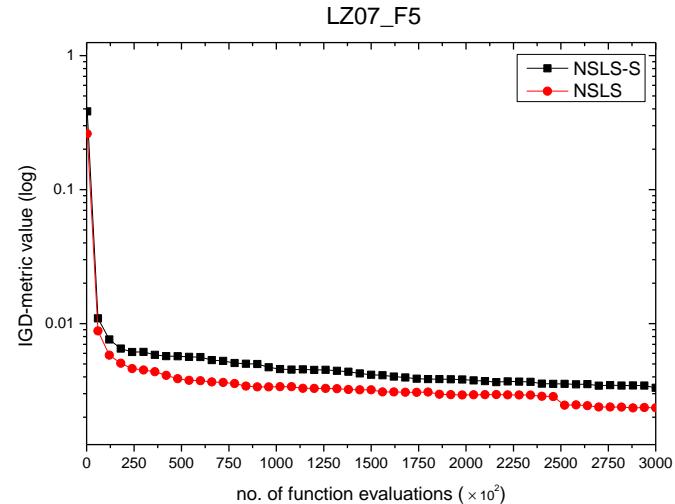


Fig. 11. The IGD-metric values versus the number of function evaluations for LZ07_F5.

3) Results of the proposed local search method on GD

In order to demonstrate the performance of the proposed local search schema applied in NSLS, the performance measure:

generational distance (GD) which is introduced in Section III-B3 is adopted. GD measures how far the nondominated solutions from those in the Pareto-optimal front. In other words, GD is an effective performance measure to demonstrate the convergence ability of the algorithm. For reason of space, only the results of eight bi-objective problems of LZs and three tri-objective problems: LZ07_F6, UF9, and UF10 with $N = 300$ (595 in the tri-objective problems instead) and $T = 500$ in Section III-E are shown to demonstrate the performances of the proposed algorithm (NSLS) and other algorithms in terms of GD. In addition, to show the effect of the proposed local search schema clearly, the performance of MODEA proposed by Ali *et al.* [57] is used for comparison too. For MODEA, we have identical parameter settings as suggested in the original study except N and T . They are taken for 300 (595 in the tri-objective problems instead) and 500, respectively. MODEA has been implemented in C++. Tables IX and X show the comparison results of the proposed algorithm (NSLS) and the other algorithms based on GD. It is clear from Tables IX and X that NSLS has a better convergence performance than the other algorithms on almost all problems except on LZ07_F1, which reveals that NSLS is an effective method to solve the MOPs. To demonstrate the effect of the proposed local search schema, the GD-metric values versus the number of generations for the problem LZ07_F9 is shown in Fig. 12, from which it is obvious that NSLS has a faster convergence speed than the other three algorithms and can achieve the best GD value. What gives evidence to support the analysis of the proposed local search schema in Section II-A2.

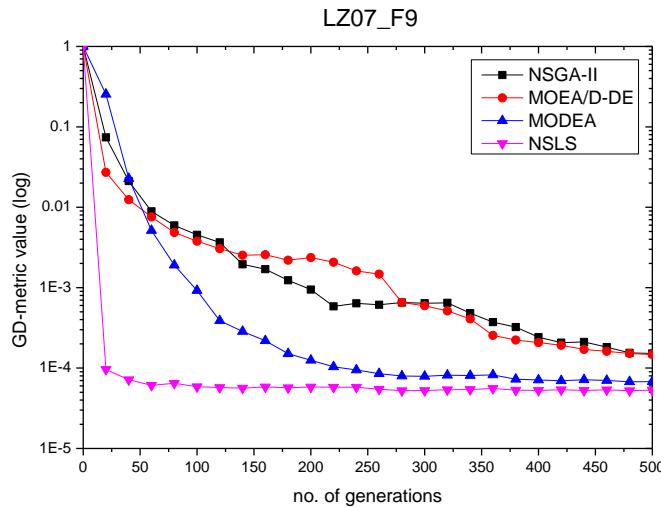


Fig. 12. The GD-metric values versus the number of generations for LZ07_F9.

G. Effect of the Farthest-candidate Approach

1) Comparison of NSLS and the above algorithms

In this section, we demonstrate an additional performance metric: the spread metric Δ introduced in Section III-B4 to

show the effect of the farthest-candidate approach. Δ is designed to measure the extent of spread archived among the nondominated solutions. The spread metric Δ could determine whether the farthest-candidate approach really plays an important role in the proposed algorithm (NSLS). For reason of space, only the results of eight bi-objective problems of LZs and three tri-objective problems: LZ07_F6, UF9, and UF10 in Section III-E ($N = 300$ for the bi-objective problems, $N = 595$ for the tri-objective problems and $T = 500$) are shown to demonstrate the performances of the proposed algorithm (NSLS) and the other algorithms on the spread metric Δ . Tables XI and XII show the comparison results of the proposed algorithm (NSLS) and the other algorithms based on the spread metric Δ . From the spread results in Tables XI and XII, we can find that the proposed algorithm clearly obtains a better spread than the other algorithms on all problems. The comparison results reveal that the farthest-candidate approach is better than other algorithms in maintaining the diversity of the nondominated solutions.

2) Comparison of NSLS and NSLS-V

In this section, to further demonstrate the superiority of the farthest-candidate approach, the performance of a pruning method proposed by Kukkonea and Deb [78] is used for comparison too. The pruning method in [78] is based on a crowding estimation technique using nearest neighbors of solutions rather than the crowding distance proposed in NSGA-II. The crowding estimation is based on a vicinity distance, which is the product of the distances to the m (the number of objectives) nearest neighbors. The solution with the smallest value of vicinity distance is supposed to be the most crowded one. For a good maintenance of crowding information, the values of the vicinity distances are recalculated after removal of the most crowded one. The overall computational complexity of the pruning method is $O(m^2 N \log N)$.

For a fair comparison, the farthest-candidate approach in NSLS is replaced with the pruning method in [78] and the new version of NSLS is called as NSLS-V. The algorithm is tested on seven test problems out of the above seventeen test problems. The experimental results in 20 runs are shown in Table XIII, from which the effect of the farthest-candidate approach is almost the same with the pruning method in [78]. Considering the spread metric Δ , the performance of the farthest-candidate approach is not better than the pruning method in [78], however, the values of IGD-metric, which measures the convergence and the diversity of these obtained nondominated solutions are reasonable to solve MOPs. Moreover, the farthest-candidate approach is an easy and new technique for diversity preservation. In a word, the farthest-candidate approach is a promising technique for dealing with MOPs.

TABLE XII

THE Δ -METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY EACH ALGORITHM ON THE TRI-OBJECTIVE PROBLEMS: LZ07_F6, UF9, AND UF10 WHERE $N = 595$ AND $T = 500$

Δ value	Problem	LZ07_F6	UF9	UF10
NSGA-II	mean \pm std	0.532075 \pm 0.068359	0.626447 \pm 0.062319	0.837408 \pm 0.420489
MOEA/D-DE	mean \pm std	0.580103 \pm 0.098679	1.04274 \pm 0.136498	1.35852 \pm 0.308079
NSLS-C	mean \pm std	0.724513 \pm 0.275907	0.556887 \pm 0.102623	0.521169 \pm 0.086526
NSLS	mean \pm std	0.135941\pm0.007563	0.119704\pm0.005636	0.208127\pm0.045232

TABLE XIII
RESULTS OF NSLS AND NSLS-V WHERE $N = 300$ AND $T = 500$

Metric	Problem	ZDT4	LZ07_F7	LZ07_F8	UF5	UF6	LZ07_F6	UF10
IGD-value (mean \pm std)	NSLS-V	0.198996 \pm 0.132894	0.04108 \pm 0.011059	0.086738 \pm 0.00921	0.059575 \pm 0.0036	0.033027 \pm 0.004591	0.031878 \pm 0.000318	0.178985 \pm 0.040136
		0.178723 \pm 0.092986	0.040538 \pm 0.01385	0.083961 \pm 0.011888	0.059321 \pm 0.002402	0.032427 \pm 0.00342	0.034787 \pm 0.000651	0.190905 \pm 0.033656
	NSLS	0.282511 \pm 0.0878	1.21056 \pm 0.502481	1.15514 \pm 0.293878	1.28444 \pm 0.316038	1.67209 \pm 0.117166	0.114924 \pm 0.009323	0.172057 \pm 0.029047
Δ value (mean \pm std)	NSLS-V	0.378406 \pm 0.052768	1.02092 \pm 0.578565	1.03755 \pm 0.285274	1.23642 \pm 0.267617	1.61499 \pm 0.194479	0.143333 \pm 0.0082	0.192213 \pm 0.053941

TABLE XIV

STATISTICAL RESULTS BY WILCOXON TEST FOR IGD-METRIC CONSIDERING ZDT5, LZ07S, AND UFS WHERE $N = 100$ AND $T = 250$

NSLS vs.	R^+	R^-	p -value
NSGA-II	197	13	0.001
MOEA/D-DE	208	2	0
NSLS-C	84	21	0.048

TABLE XV

STATISTICAL RESULTS BY WILCOXON TEST FOR IGD-METRIC CONSIDERING LZ07S AND UFS WHERE $N = 300$ AND $T = 500$

NSLS vs.	R^+	R^-	p -value
NSGA-II	120	0	0.001
MOEA/D-DE	103	17	0.015
NSLS-C	108	12	0.006

TABLE XVI

STATISTICAL RESULTS BY WILCOXON TEST FOR GD AND Δ METRICS CONSIDERING UF9, UF10, AND LZ07S WHERE $N = 300$ AND $T = 500$

Metric name	GD-metric			Δ -metric		
	R^+	R^-	p -value	R^+	R^-	p -value
NSLS vs.	R^+	R^-	p -value	R^+	R^-	p -value
NSGA-II	66	0	0.003	66	0	0.003
MOEA/D-DE	65	1	0.004	66	0	0.003
MODEA	66	0	0.003	—	—	—
NSLS-C	—	—	—	66	0	0.003

H. Results of Wilcoxon signed ranks test analysis

The above results are also compared statistically and the corresponding results are given in Tables XIV-XV. From Tables XIV and XV, it is clear that in case of IGD-metric, the p -values are all smaller than $\alpha = 0.05$, which reveals that NSLS outperforms all the other algorithms on both of the two experiments with different parameter settings of the population size and the number of iterations. Likewise, from Table XVI, which shows the results on the basis of the GD and the spread Δ metrics, it is obvious that NSLS is better than the other algorithms too.

I. Sensitivity in NSLS

1) Sensitivity of μ

μ is a control parameter adopted in the proposed local search schema in NSLS. To study the sensitivity of the performance to μ in NSLS, we have tested different settings of μ in the implementation of NSLS on ZDT1. All parameter settings are similar to that in Section III-D, except the setting of μ . Fig. 13 shows the mean IGD-metric values in 20 runs versus the value of μ in NSLS on ZDT1. It is clear from Fig. 13 that NSLS performs well with μ from 0 to 1.0 on ZDT1. And it works well for all values of μ with a small difference. Therefore, it can be concluded that NSLS is not very sensitive to the setting of μ , at

least for the MOPs that are more or less similar to these problems adopted in this paper.

ZDT1

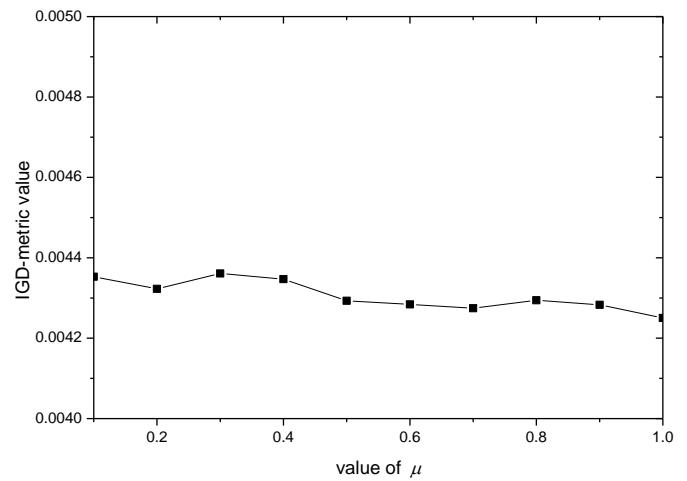


Fig. 13. The mean IGD-metric values versus the value of μ in NSLS for ZDT1.

ZDT1

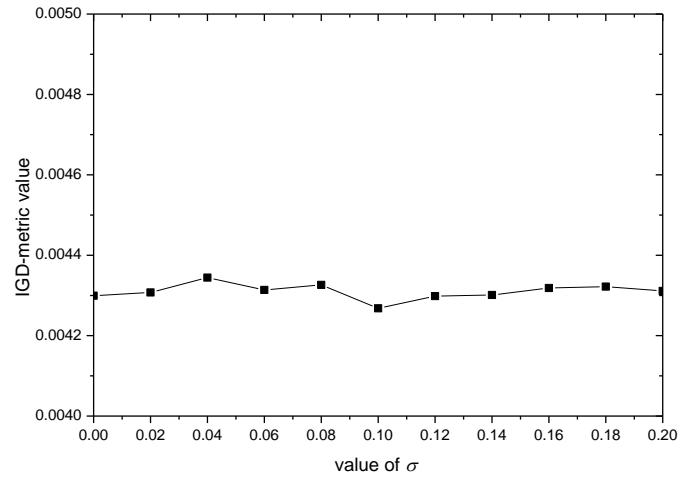


Fig. 14. The mean IGD-metric values versus the value of σ in NSLS for ZDT1.

2) Sensitivity of σ

σ is another control parameter utilized in the proposed local search schema in NSLS. To discuss the sensitivity of the performance to σ in NSLS, we have tested different settings of σ in the implementation of NSLS on ZDT1. All parameter settings are similar to that in Section III-D, except the setting of σ . It is clear from Fig. 14 that NSLS performs well with σ from 0 to 0.2 on ZDT1. And it works well for all values of σ with a small difference and achieves the best value with $\sigma = 0.1$.

Therefore, we can claim that NSLS is not very sensitive to the setting of σ too.

IV. CONCLUSION

This paper has proposed a new framework algorithm called NSLS for solving MOPs. NSLS maintains a population with a constant size. At each iteration, for each solution in the current population, NSLS first uses a proposed local search schema to generate two neighborhoods for each variable in the current solution. Then, a replacement strategy is employed to decide whether the two neighborhoods replace the current solution or not and which one to be chosen to replace the current solution. This process ends until all the variables of the current solution are handled. When all solutions in the population are processed, the farthest-candidate approach combined with the fast nondominated sorting is adopted to choose the new population for the next iteration.

We have compared NSLS with another version of NSLS (NSLS-C) and three famous classical algorithms: NSGA-II, MOEA/D-DE, and MODEA on some typical problems. To give a more comprehensive display of the comparison, two experiments have been performed with different values of population size and generations. In addition, we have shown the effect of the local search method and the farthest-candidate approach with corresponding performance metric, separately. The experimental results have shown that NSLS could significantly outperform NSGA-II, MOEA/D-DE, NSLS-C, and MODEA on most of these problems. Moreover, the results have been compared statistically using Wilcoxon signed ranks test analysis, which reveals that NSLS is superior to all the other algorithms.

In general, though the proposed NSLS-C is also better at solving the MOP than the other algorithms, NSLS is a typical novel method which is the most efficient one. It implies that the combination of the local search strategy and the farthest-candidate approach with nondominated sorting technology can be a very promising algorithm to deal with MOPs.

Additionally, we have experimentally investigated the feasibility of the proposed local search schema and the effectiveness of the farthest-candidate approach, separately. To study the effect of the proposed local search schema, a version of NSLS without using the perturbation factor called NSLS-WN and a new version of NSLS-S have been proposed for comparison. Experimental results have shown that the proposed local search schema is efficient. To further study the effect of the farthest-candidate approach, another version of NSLS: NSLS-V has been presented for comparison too. Experimental results have shown that the employment of the farthest-candidate approach is reasonable.

We have experimentally investigated the sensitivities to the two parameters: μ and σ . We have found that NSLS is not very sensitive to the settings of μ and σ .

The computational results have shown that the proposed NSLS is superior to the other algorithms on most of test problems rather than all of test problems. The main weakness of NSLS is that it is easy to trap into the local optimal on some test

problems. Additionally, we do not affirm that NSLS is always better than the other algorithms. The strengths and weaknesses of the algorithm should be studied based on the characteristics of the test problems.

A more extensive comparison of NSLS with other multiobjective methods, the task of avoiding the local minima in the proposed local search schema, and applying NSLS for constrained MOPs remain as future studies.

REFERENCES

- [1] C. A. Coello Coello, G. B. Lamont, and D. A. VanVeldhuizen, "Evolutionary Algorithms for Solving Multi-Objective Problems," 2007, ch.1, sec.1, pp. 1- 40.
- [2] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, Lawrence Erlbaum, 1985, pp. 93-100.
- [3] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization," in *Proc. 5st Int. Conf. Genetic Algorithms*, CA: Morgan Kauffman, 1993, pp. 416-423.
- [4] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st Int. Conf. Evol. Comput.*, Piscataway, NJ, 1994, pp. 82-87.
- [5] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no.3, pp. 221-248, 1994.
- [6] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach". *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257-271, Feb. 1999.
- [7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Swiss Federal Institute of Technology, Lausanne, Switzerl, Tech. Rep, pp. 103, 2003.
- [8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, Apr. 2002.
- [9] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archive evolutionary strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149-172, 2000.
- [10] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," Pasadena, California, USA: Tech. Rep. Caltech Concurrent Computation Program, Report 826, California Institute of Technology, 1989.
- [11] H. Ishibuchi, "A multi-objective genetic local search and its application to flow shop scheduling," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, no. 3, pp. 392-403, Aug. 1998.
- [12] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem – A comparative experiment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 402-412, Aug. 2002.
- [13] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multi-objective permutation flow shop," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204-223, Apr. 2003.
- [14] H. Ishibuchi and K. Narukawa, "Performance evaluation of simple multi-objective genetic local search algorithms on multi-objective 0/1 knapsack problems," in *Proc. Conf. Evol. Comput.*, 2004, pp. 441- 448.
- [15] A. Jaszkiewicz, "Genetic local search for multiple objective combinatorial optimization," *Eur. J. Oper. Res.*, vol. 371, no. 1, pp. 50-71, Feb. 2002.
- [16] T. Murata , H. Nozawa., Y. Tsujimura, M. Gen, and H. Ishinuchi, "Effect of local search on the performance of cellular multi-objective genetic algorithms for designing fuzzy rule based classification systems," In *Proc. Conf on Evol. Comput.*, 2002, pp. 663-668.
- [17] J. D. Knowles and D. Corne, "M-PAES: A memetic algorithm for multiobjective optimization," In *Proc. Conf. Evol. Comput.*, 2000, pp. 325-332.
- [18] K. Harada, J. Sakuma, and S. Kobayashi, "Local search for multiobjective function optimization: Pareto descent method," In *Proc. Genetic Evol. Comput.*, 2006, pp. 659-666.
- [19] K. Sindhya, K. Deb, and K. Miettinen, "A local search based evolutionary multiobjective approach for fast and accurate convergence," in *Parallel Problem Solving from Nature-PPSN X*, ser. Lecture Notes in Computer Science, vol. 5199/2008. Springer Berlin/Heidelberg, 2008, pp. 815-824.

- [20] K. Sindhya, A. Sinha, K. Deb and K. Miettinen, "Local search based evolutionary Multi-objective optimization algorithm for constrained and unconstrained problems," in *Proc. Evol. Comput.*, 2009, pp. 18-21.
- [21] S. F. Adra, T. J. Dodd, I. A. Griffin, and P. J. Fleming, "Convergence Acceleration Operator for Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol.13, no. 4, pp. 825-847, Aug. 2009.
- [22] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of biased neighborhood structures in multiobjective memetic algorithms," *Soft Comput.*, vol. 13, no.8-9, pp. 795-810, Jul. 2009.
- [23] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schutze, "HCS: a new local search strategy for memetic multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 112-132, Feb. 2010.
- [24] L. V. Santana-Quintero, A. G. Hernández-Díaz, J. Molina, C. A. Coello Coello, and R. Caballero, "DEMORS: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 470 - 480, Mar. 2010.
- [25] P. A. N. Bosman, "On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multi-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 51-69, Feb. 2012.
- [26] L. Y. Tseng and C. Chen, "Multiple Trajectory Search for Multiobjective Optimization," in *Proc. CEC'07*, Singapore, 2007, pp. 3609-3616.
- [27] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multi-objective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems," *IEEE Trans. Syst. Man Cybern. B*, vol. 37, no. 3, pp. 560-575, Jun. 2007.
- [28] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective Immune Algorithm with Nondominated Neighbor-Based Selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225-255, 2008.
- [29] S. Z. Martínez and C. A. Coello Coello, "A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques," in *Proc. Parallel Problem Solving from Nature --PPSN X*, Dortmund, Germany, 2008, pp. 837-846.
- [30] S. Zapotecas Martínez, and C. A. Coello Coello, "Hybridizing an Evolutionary Algorithm with Mathematical Programming Techniques for Multi-Objective Optimization," in *Proc. GECCO '08*, New York, NY, USA, 2008, pp. 769-770.
- [31] S. Tiwari, G. Fadel, P. Koch, and K. Deb, "Performance assessment of the hybrid archive-based micro genetic algorithm (AMGA) on the CEC09 test problems," in *Proc. the 11th conf. Evol. Comput.*, Piscataway, NJ, USA, 2009, pp. 1935-1942.
- [32] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an Estimation of Nadir Objective Vector Using a Hybrid of Evolutionary and Local Search Approaches," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 821-841, Dec. 2010.
- [33] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, Dec. 2007.
- [34] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol.13, no. 2, pp. 284-302, Apr. 2009.
- [35] H. Li and D. L. Silva, "An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem," presented at the 5th Int. Conf. EMO 2009, pp. 481-494.
- [36] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456-474, Jun. 2010.
- [37] Q. Zhang, H. Li, D. Maringer, and E. P. K. Tsang, "MOEA/D with NBI-style Tchebycheff Approach for Portfolio Management," in *Proc. CEC 2010*, Barcelona, Spain, pp. 1-8, July 2010.
- [38] C. M. Chen, Y. P. Chen, and Q. Zhang, "Enhancing MOEA/D with Guided Mutation and Priority Update for Multi-Objective Optimization," in *Proc. CEC 2009*, Trondheim, Norway, 2009, pp. 209-216.
- [39] R. Battiti and A. Passerini, "Brain-Computer Evolutionary Multiobjective Optimization: A Genetic Algorithm Adapting to the Decision Maker," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 671-687, Oct. 2010.
- [40] L. F. Santos, S. L. Martins, and A. Plastino, "Applications of the DM-GRASP heuristic: A Survey," *Int. Trans. Oper. Res.*, vol. 15, no. 4, pp. 387-416, Jun. 2008.
- [41] Y. Mei, K. Tang, and X. Yao, "Decomposition-Based Memetic Algorithm for Multiobjective Capacitated Arc Routing Problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151- 165, Apr. 2011.
- [42] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multiobjective particle swarm optimization algorithm design," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 42-54, Apr. 2010.
- [43] A. Elhossini, S. Areibi, and R. Dony, "Strength Pareto particle swarm optimization and hybrid EA-PSO for multiobjective optimization," *Evol. Comput.*, vol. 18, no. 1, pp. 127-156, 2010.
- [44] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," In *Proc. 6th Symposium MHS95*, 4-6 pp. 39-43, Oct. 1995.
- [45] D. S. Liu, K. C. Tan, S. Y. Huang, C. K. Goh, and W. K. Ho, "On solving multiobjective bin packing problems using evolutionary particle swarm optimization," *Eur. J. Oper. Res.*, vol.190, no. 2, pp. 357-382, Oct. 2008.
- [46] S. Agrawal, B. K. Panigrahi, and M. K. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 529-541, Oct. 2008.
- [47] L. Tang and X. Wang, "A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems," *IEEE Trans, Evol. Comput.*, vol.17, no. 1, pp. 20-45, Feb. 2013.
- [48] R. Storn and K. Price, "Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep. TR-95-012, ICSI, 1995.
- [49] K. V. Price, "An introduction to differential evolution, in: *New Ideas in Optimization*," McGraw-Hill, 1999, pp. 79-108.
- [50] W. Gong and Z. Cai, "An improved multiobjective differential evolution based on Pareto-adaptive epsilon-dominance and orthogonal design," *Eur. J. Oper. Res.*, vol. 198, no. 2, pp. 576-601, Oct. 2009.
- [51] B. Alatas, E. Akin, and A. Karci, "MODENAR: multi-objective differential evolution algorithm for mining numeric association rules," *Appl. Soft Comput.*, vol. 8, no.1, pp. 646-656, Jan. 2008.
- [52] B. Y. Qu and P. N. Suganthan, "Multi-objective differential evolution with diversity enhancement," *J. Zhejiang Univ. Sci.*, vol. 11, no. 7, pp. 538-543, Jul. 2010.
- [53] H. A. Abbass and R. Sarker, "The Pareto differential evolution algorithm," *Int. J. Artif. Intell. T.*, vol.11, no.4, pp. 531-552, Mar. 2002.
- [54] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658-675, Dec. 2006.
- [55] Y. Wang and Z. Cai, "Combining Multiobjective Optimization with Differential Evolution to Solve Constrained Optimization Problems," *IEEE Trans, Evol. Comput.*, vol.16, no. 1, pp. 117-134, Aug. 2012.
- [56] A. Cichon, E. Szlachcic, and J. F. Kotowski, "Differential evolution for multiobjective optimization with self adaptation," in *Proc. 14th Int. Conf. Intell. Eng. Syst.* , 2010, pp. 165-169.
- [57] M. Ali, P. Siarry, and M. Pant, "An efficient differential evolution based algorithm for solving multi-objective optimization problems," *Eur. J. Oper. Res.*, vol. 217, no. 2, pp. 404-416, Mar. 2012.
- [58] M. Ali, M. Pant, and A. Abraham, "A modified differential evolution algorithm and its application to engineering problems," In *Proc. SoCPaR-2009*, 2009, pp.196-201.
- [59] T. Fukuda, K. Mori, and M. Tsukiyama, "Immune networks using genetic algorithm for adaptive production scheduling," in *Proc. 15th IFAC World Congress*, vol. 3, 1993, pp. 57-60.
- [60] Z. H. Hu, "A multiobjective immune algorithm based on a multiple-affinity model," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 60-72, Apr. 2010.
- [61] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft Comput.*, vol.8, no.2, pp.959-971, Mar. 2008.
- [62] S. Kirkpatrick, C.D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May. 1983.
- [63] L. Sánchez and J. R. Villar, "Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms," *Inform. Sciences*, vol. 178, no. 4, pp. 952-970, Feb. 2008.
- [64] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra, "Dominancebased multiobjective simulated annealing," *IEEE Trans, Evol. Comput.*, vol. 12, no. 3, pp. 323-342, Jun. 2008.
- [65] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealingbased multiobjective optimization algorithm: AMOSA," *IEEE Trans, Evol. Comput.*, vol. 12, no. 3, pp. 269-283, Jun. 2008.
- [66] R. P. Beausoleil, "MOSS multiobjective scatter search applied to nonlinear multiple criteria optimization," *Eur. J. Oper. Res.*, vol. 169, no. 2, pp. 426-449, Mar. 2006.
- [67] A. Krishna and K. Rao, "Multiobjective optimization of surface grinding operations using scatter search approach," *Int. J. Adv. Manuf. Technol.*, vol. 29, no. 5, pp. 475-480, Jun. 2006.
- [68] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, 2004.

- [69] D. Angus, "Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem," presented at IEEE Symposium Comput. Intell. *MCDM* 2007, 2007, pp. 333-340.
- [70] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection," *Eur. J. Oper. Res.*, vol. 171, no. 3, pp. 830-841, Jun. 2006.
- [71] I. Rechenberg, "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Ph.D. Thesis, Technical University of Berlin, Department of Process Engineering, 1971.
- [72] I. Rechenberg, "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Frommann-Holzboog Verlag, Stuttgart, 1973.
- [73] H. G. Beyer and H. P. Schwefel, "Evolution strategies—A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3-52, May 2002.
- [74] H. Sato, H. E. Aguirre, and K. Tanaka, "On the locality of dominance and recombination in multiobjective evolutionary algorithms," in *Proc. CEC '05*, 2005, pp. 451-458.
- [75] B. de la Iglesia, A. P. Reynolds, and V. J. Rayward-Smith, "Developments on a multi-objective metaheuristic (MOMH) algorithm for finding interesting sets of classification rules," in *Proc. EMO'05*, Guanajuato, Mexico, 2005, pp. 826-840.
- [76] N. Hallam, P. Blanchfield, and G. Kendall, "Handling diversity in evolutionary multiobjective optimization," in *Proc. CEC'05*, 2005, pp. 2233-2240.
- [77] S. Kukkonen and K. Deb, "Improved pruning of nondominated solutions based on crowding distance for bi-objective optimization problems," in *Proc. CEC'06*, Vancouver, Canada, 2006, pp. 1179-1186.
- [78] S. Kukkonen and K. Deb, "A fast and effective method for pruning of nondominated solutions in many-objective problems," in *Proc. PPSN IX*, Reykjavik, Iceland, 2006, pp. 553-562.
- [79] D. P. Mitchell, "Spectrally optimal sampling for distribution ray tracing," in *Proc. ACM SIGGRAPH*, New York, NY, USA, 1991, pp. 157-164.
- [80] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, 2000.
- [81] Q. Zhang, A. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multi-objective optimization test instances for the CEC09 special session and competition," University of Essex and Nanyang Technological University, Tech. Rep. CES-487, 2008.
- [82] Y. Wang, L. Wu, and X. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Comput.*, vol. 14, no. 3, pp. 193-209, Feb. 2010.
- [83] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. daFonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117-132, Apr. 2003.
- [84] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: a history and analysis," Department of Electrical and Computer Engineering. Graduate School of Engineering, Air Force Inst Technol, Wright Patterson, Tech. Rep. TR-98-03, 1998.
- [85] Handbook of Parametric and Nonparametric Statistical Procedures. 3rd ed., FL: Chapman & Hall/CRC, London, U.K./Boca Raton, 2003, pp. 609-632.



Bili Chen received the B.Sc. degree in Software Engineering from Fuzhou University, Fuzhou, China, in 2008, and the M.Sc. degree in computer science from Xiamen University, Xiamen, China, in 2011. She is working toward the Ph.D. degree in computer science from the Xiamen University, Xiamen, China.

She is a Member of the Intelligent Software Development Laboratory of Xiamen University. Her current research interests include multiobjective optimization, evolutionary computation, metaheuristics and portfolio optimization problem.



Wenhua Zeng received the B.Sc. degree in automatic control from the Nanjing Institute of Chemical Engineering, Nanjing, China, in 1983, and the M.S. and Ph.D. degrees in automatic control from Zhejiang University, Zhejiang, China, in 1986 and 1989, respectively.

He is currently a Professor and Vice Dean in the college of software, University of Xiamen, Xiamen, China. He is the author of one book, more than 80 articles. His main research areas are multiobjective optimization, evolutionary computation, cloud computing, grid computing, embedded software, intelligent control, artificial intelligence and their applications.



Yangbin Lin received the B.Sc. and M.Sc. degrees in computer science from Xiamen University, Xiamen, China, in 2008 and 2011, respectively. He is working toward the Ph.D. degree in computer science from the Xiamen University, Xiamen, China.

He is currently a Member of the Center of Excellence for Remote Sensing and Spatial Informatics, Xiamen University, Xiamen, China.

He worked as a Research Assistant of Hong Kong City University between July 1 and September 27 in 2010. He worked as a Software Engineer in Google Company, Shanghai, China between 2011 and 2012. His current research interests include multiobjective optimization, irregular packing, computational geometry, point cloud, remote sensing and spatial informatics.



Defu Zhang received the B.Sc. and M.Sc. degrees in computational mathematics from Department of Mathematics, Xiangtan University, Hunan, China, in 1996 and 1999, respectively, and the Ph.D. degree in School of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan, China, in 2002.

He worked as a PostDoc at the Longtop for financial data mining group. He worked as a Visiting Scholar of Hong Kong City University in summer (08,09,11,12,13). He worked at University of Wisconsin-Madison as a Visiting Scholar in 2010. He worked as a Visiting Fellow of Macau University between August 28 and November 27 in 2013. He is currently an Associate head in the department of Computer Science, Xiamen University, Xiamen, China and a Chief Scientist of Xiamen Intellecision Info-Tech Co., LTD. His main research areas are computational intelligence, data mining, big data, large scale optimization, online decision optimization, operational research, multiobjective optimization and their applications.