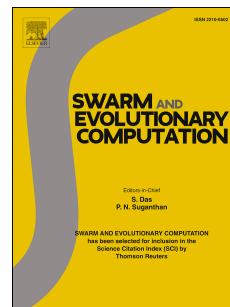


Accepted Manuscript

Multimodal multiobjective optimization with differential evolution

Jing Liang, Weiwei Xu, Caitong Yue, Kunjie Yu, Hui Song, Oscar D. Crisalle, Boyang Qu



PII: S2210-6502(17)30836-2

DOI: <https://doi.org/10.1016/j.swevo.2018.10.016>

Reference: SWEVO 465

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 20 October 2017

Revised Date: 23 August 2018

Accepted Date: 30 October 2018

Please cite this article as: J. Liang, W. Xu, C. Yue, K. Yu, H. Song, O.D. Crisalle, B. Qu, Multimodal multiobjective optimization with differential evolution, *Swarm and Evolutionary Computation BASE DATA* (2018), doi: <https://doi.org/10.1016/j.swevo.2018.10.016>.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Multimodal Multiobjective Optimization with Differential Evolution

Jing Liang^a, Weiwei Xu^a, Caitong Yue^a, Kunjie Yu^a, Hui Song^b, Oscar D. Crisalle^c, Boyang Qu^{d,*}

^a*School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China*

^b*School of Science, Computer Science and Information Technology, RMIT University, Melbourne 3000, Australia*

^c*Department of Chemical Engineering, University of Florida, FL 32611, USA*

^d*School of Electronic and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China*

Abstract

This paper proposes a multimodal multiobjective Differential Evolution optimization algorithm (MMODE). The technique is conceived for deployment on problems with a Pareto multimodality, where the Pareto set comprises multiple disjoint subsets, all of which map to the same Pareto front. A new contribution is the formulation of a decision-variable preselection scheme that promotes diversity of solutions in both the decision and objective space. A new mutation-bound process is also introduced as a supplement to a classical mutation scheme in Differential Evolution methods, where offspring that lie outside the search bounds are given a second opportunity to mutate, hence reducing the density of individuals on the boundaries of the search space. New multimodal multiobjective test functions are designed, along with analytical expressions for their Pareto sets and fronts. Some test functions introduce more complicated Pareto-front shapes and allow for decision-space dimensions greater than two. The performance of the MMODE algorithm is compared with five other state-of-the-art methods. The results show that MMODE realizes superior performance by finding more

*Corresponding author.

Email addresses: liangjing@zzu.edu.cn (Jing Liang), xuweiwei888@163.com (Weiwei Xu), zzuyuecaitong@163.com (Caitong Yue), yukunjie@zzu.edu.cn (Kunjie Yu), hui.song@rmit.edu.au (Hui Song), crisalle@che.ufl.edu (Oscar D. Crisalle), qby1984@hotmail.com (Boyang Qu)

and better distributed Pareto solutions.

Keywords: Evolutionary algorithm, Multimodal, Multiobjective optimization, Differential Evolution, Test functions

1. Introduction

A general multiobjective optimization problem can be described as follows:

$$\text{Minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T, \quad x \in \Omega \quad (1)$$

where Ω is the decision space, $x \in R^k$ is the decision variable, k is the number of decision variables, $F : \Omega \rightarrow R^m$ is a multidimensional objective function with scalar objective components $f_i(x), i = 1, 2, \dots, m \geq 2$, and R^m is the objective space. Hence $F(x)$ consists of m objective functions and the optimization problem involves k decision variables. The decision space $\Omega \subset R^k$ is a closed and connected region, and all the objective functions are continuous over the decision space; therefore Problem (1) is a continuous multivariable optimization problem.

- 10 The multiobjective optimization task defined by Problem (1) is assumed to be nontrivial; that is, there is no solution in the decision space that simultaneously optimizes all the objective functions. Furthermore, the scalar objective functions may have multiple optimal solutions. It is well known that under these conditions there exist a number of solutions in the decision space, with
- 15 conflicting optimal objective functions. A solution to the multiobjective optimization problem is said to be nondominated if none of the scalar objective function values can be improved without degrading the value of other objective functions. The set of all nondominated solutions in the decision space is called a Pareto set and the set associated objective functions is called the Pareto front.
- 20 This work focuses on the problem of Pareto multimodality, where the Pareto set comprises a number of disjoint subsets in the decision space. Fig. 1 shows a simple conceptual diagram illustrating key features of a multimodal optimization function. The two continuous segments in the (x_1, x_2) decision space represent

two Pareto subsets, $PS1$ and $PS2$, which satisfy the multimodal condition $PS1 \cap PS2 = \emptyset$, and that define the Pareto set $PS = PS1 \cup PS2$. The continuous segment on the (f_1, f_2) objective space represents the Pareto front corresponding to PS . The mappings from the two circles in the decision space to the circle in the objective space illustrate that there are solutions in $PS1$ and in $PS2$, which in spite of belonging to disjoint Pareto subsets nevertheless map to the same objective function. Multiobjective functions exhibiting multimodality can be found in a number of references. For example [1] includes functions that feature a Pareto set composed of two Pareto subsets (also called Pareto regions), while [2] documents a function with nine Pareto regions that map to the same Pareto front. More examples can be found in [3].

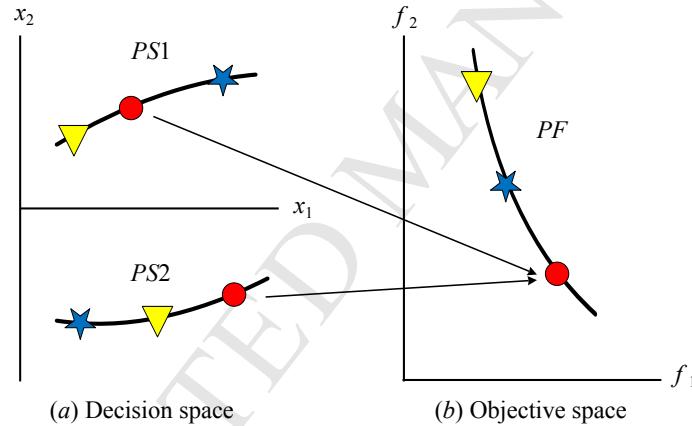


Fig. 1. Conceptual diagram illustrating Pareto multimodality. The Pareto subsets $PS1$ and $PS2$ in the decision space map to the same Pareto front. The Pareto set is the union of the sets $PS1$ and $PS2$.

A number of algorithms for solving general multiobjective optimization problems based on evolutionary optimization ideas have been proposed in the literature over the past two decades. Recent efforts by our research group led to the formulation of the DN-NSGAII [1] and MO-Ring-PSO-SCD [3] algorithms for multimodal problems. Among other ideas, a distinguishing characteristic of our effort is the adoption of the crowding distance concept, inspired on the work of

Deb and Tiwari [2], with highly promising results. However, further improvements are called for to mitigate observed shortcomings. First, a drawback of the approaches on record is that the crowding distance is implemented only in the decision space, ignoring the distribution of solutions. Second, the resulting distribution of solutions in the objective space is often not uniform. Finally, algorithm testing experiments have been carried using test objective functions with relatively simple geometric shapes in the objective space.

We seek to address the drawbacks identified above by proposing a new multimodal multiobjective optimization algorithm with Differential Evolution (MMODE) that seeks to find larger numbers of Pareto-optimal solutions with a more uniform distribution. The work adopts a Differential Evolution framework [4] [5], which is known to be highly efficient for multidimensional numerical searches over continuous spaces, and supplements it with the following innovations: (i) a preselection scheme to find more subsets in the Pareto set, (ii) a mutation-bound processing method to improve the distribution of Pareto sets, and (iii) the introduction of new designs of Pareto-set and Pareto-front shapes to use as test functions for the purpose of evaluating the performance of multimodal optimization algorithms.

Section 2 of the paper presents a brief review of related work. Section 3 introduces the new MMODE algorithm, and Section 4 describes the adopted performance metrics and introduces twelve test functions used for performance assessment. Section 5 presents experimental results and compares MMODE with five state-of-the-art evolutionary algorithms. Finally, Section 6 offers conclusions and suggests future work on the topic.

2. Related Work

2.1. Background

Over the last two decades various multiobjective evolutionary optimization algorithms have emerged in the literature, seeking to find all or most of the solutions in the Pareto set [6–13]. In particular, Knowles and Corne [14] introduce

70 a Pareto-archived evolution strategy to select a population of solutions, improving
 75 to some extent solution diversity and enhancing the convergence speed of
 the numerical algorithm [14]. Yen and Lu [15] propose a dynamic evolutionary
 algorithm to address multiobjective optimization problems by adjusting the
 80 population size [15]. Coello *et al.* [16] use a particle swarm optimization al-
 gorithm for multiobjective problems, demonstrating an approach that proves
 competitive for solving multiobjective optimization. Tan *et al.* [17] propose a
 distributed cooperative coevolutionary algorithm that evolves multiple solutions
 85 in the form of cooperating subpopulations, and exploits inherent algorithmic
 parallelisms by sharing the computational workload among computers collab-
 orating over a digital network. Inspired by techniques from traditional mul-
 tiobjective optimization theory, Zhang and Li [18] introduce a multiobjective
 90 evolutionary algorithm based on decomposition of the problem into a number
 of scalar optimization subproblems which are simultaneously optimized.

Another goal of significance is to obtain numerical solutions that are well
 85 distributed over the Pareto set, thus avoiding the undesirable concentration of
 such solutions in local domains that are not representative of the entire set. A
 typical approach of evolutionary algorithms is to promote population diversity,
 with the expected consequence of improved solution distributions. To this end,
 Tahernezhad *et al.* [19] propose an innovative clustering-based scheme, and
 90 obtain more diverse and better-distributed solutions. Giagkiozis and Fleming
 [20] present a novel methodology to produce additional Pareto-optimal solu-
 tions from a Pareto-optimal set obtained in the evolutionary process, effectively
 dealing with the diversity problems and determining the number of solutions
 available for the decision maker. Motivated by the inefficacy of attempts to bal-
 95 ance convergence and diversity in a high-dimensional objective space, Liu *et al.*
 [21] propose a novel many-objective evolutionary algorithm using a one-by-one
 selection strategy.

Prior approaches include explicit techniques to address the problem of Pareto
 multimodality. Encouraging results have been obtained by adopting the deci-
 100 sion space crowding-distance concept of Deb and Tiwari [2], where used as a

parameter in a nondominated sorting scheme deployed in the Omni-Optimizer algorithm to improve population diversity in the decision space. Preuss *et al.* [22] and Rudolph *et al.* [23] deploy a restart strategy for obtaining an improved approximation to the Pareto set.

¹⁰⁵ Another perspective is found in Zhou *et al.* [24], where a method is introduced address with problems with a Pareto front and Pareto set that are continuous manifolds of high dimensions. Liang *et al.* [1] advance a decision-space-based niching multiobjective evolutionary algorithm for the case of multimodal multiobjective optimization problems. Yue *et al.* [3] propose an algorithm MO-Ring-PSO-SCD. This algorithm can address multimodal multiobjective optimization problems using ring topology and special crowding distance.

2.2. Succinct Overview of Multiobjective Differential Evolution Optimization

¹¹⁵ Differential Evolution (DE) is an efficient evolutionary technique for solving optimization problems [4] [5] [25]. The method requires only a few control parameters, and adequate convergence speeds. An accessible description of the principles involved in DE optimization can be found in [26].

¹²⁰ A classic DE optimization process proceeds as follows. First the user defines a feasible search range $\Omega_s \subseteq \Omega$, which is uniquely characterized by its boundary $\partial\Omega_s$. The boundary is simply expressed as the end-points of a real interval defined for each component of the decision variable. More specifically, the bounds for each component $X^{(q)}$ of the decision variable $x = (X^{(1)}, X^{(2)}, \dots, X^{(k)})^T \in R^k$ are expressed as $X^{(q),L} \leq X^{(q)} \leq X^{(q),U}$, where the superscripts L and U are used to respectively denote the lower and upper bounds of the search range for the component of the decision vector.

The next step is *initialization*, where a starting population P consisting of NP decision vectors (*individuals*) is established by assigning values to each of the components of all individuals. A common approach is to adopt the initialization rule

$$X_{i,0}^{(q)} = X_i^{(q),L} + \text{rand}(0, 1) \left(X_i^{(q),U} - X_i^{(q),L} \right) \quad (2)$$

¹²⁵ where the subindex 0 in $X_{i,0}^{(q)}$ is used to denote the component of a zero-*th* generation decision variable, and where $q = 1, 2, \dots, k$, and $i = 1, 2, \dots, NP$. The function $\text{rand}(0, 1)$ generates uniformly distributed random numbers in the interval $(0, 1)$.

¹³⁰ Next a target individual is selected from the population to produce an offspring that becomes a member of a new generation. Let the subindex G denote the $G - th$ generation, and use the notation $x_{i,G} = (X_{i,G}^{(1)}, X_{i,G}^{(2)}, \dots, X_{i,G}^{(k)})^T$ to identify the target individual of the $G - th$ generation whose components $X_{i,G}^{(q)}$, $q = 1, 2, \dots, k$ are subjected to DE mutation.

A DE mutation process is then implemented to generate a *donor vector* $v_{i,G} = (V_{i,G}^{(1)}, V_{i,G}^{(2)}, \dots, V_{i,G}^{(k)})^T$. One possible approach for finding the components of the donor vector is the DE/rand/2 technique [26], which is defined by the mutation equation

$$V_{i,G}^{(q)} = X_{r_1^i, G}^{(q)} + \Phi_1 (X_{r_2^i, G}^{(q)} - X_{r_3^i, G}^{(q)}) + \Phi_2 (X_{r_4^i, G}^{(q)} - X_{r_5^i, G}^{(q)}) \quad (3)$$

¹³⁵ where $V_{i,G}^{(q)}$ is the $q - th$ component of the donor vector and $X_{r_s^i, G}^{(q)}$ is the $q - th$ component of decision variable $x_{r_s^i, G}$. The indices r_ℓ^i , $\ell = 1, 2, \dots, 5$, are integers randomly generated in the interval $[1, NP]$ under the constraint that the set $\{i, r_1^i, r_2^i, r_3^i, r_4^i, r_5^i\}$ is unique (*i.e.*, the set contains no repeated elements). The *scale factors* $\Phi_1 \in (0, 1)$ and $\Phi_2 \in (0, 1)$ act as amplifiers for the difference terms in Equation (3), and may be chosen as equal to each other; that is, by selecting a single scale factor Φ and setting $\Phi_1 = \Phi_2 = \Phi$.

The left-hand side of Equation (3) may yield a result that lies outside the search-space boundary. In such case the out-of-bounds component of the donor vector is set equal to the closest boundary value, that is $V_{i,G}^{(q)} = X_i^{(1),L}$ or $V_{i,G}^{(q)} = X_i^{(1),U}$.

Next, a crossover process, where the donor vector exchanges components with the target vector, is implemented to potentially increase the diversity of the population. A commonly used technique is the *binomial method*, where the components of a *trial-offspring vector* $u_i = (U_i^{(1)}, U_i^{(2)}, \dots, U_i^{(k)})^T \in R^k$ are

found through the assignment

$$U_i^{(q)} = \begin{cases} V_{i,G}^{(q)} & \text{if } \text{rand}(0, 1) < Cr \\ X_{i,G}^{(q)} & \text{otherwise} \end{cases} \quad (4)$$

where the *crossover rate constant* $Cr \in (0, 1)$ is a user-defined scalar parameter. For the case of multiobjective optimization problems, Cheng *et al.* [27] propose the following *offspring generation process* to complete the crossover operation:

$$x_{i,G+1} = \begin{cases} u_i & \text{if } u_i \text{ dominates } x_{i,G} \\ x_{i,G} & \text{if } x_{i,G} \text{ dominates } u_i \end{cases} \quad (5)$$

- 145 In addition u_i is discarded when the second branch of Equation (5) holds. However, when neither branch holds, vector u_i is added to the $(G+1)-st$ generation, increasing the population size by one.

Finally a truncation process is carried out to keep the population at a constant size NP . In [27] this process is carried out through a nondominated sorting
150 algorithm that includes a grid-density index. A complete algorithm for carrying out multiobjective Differential Evolution optimization is given in [27].

3. The Proposed MMODE Algorithm

This section develops the two most salient components of the MMODE algorithm, namely the novel mutation-bound process and the preselection scheme,
155 followed by a pseudocode listing for the algorithm along with a discussion of relevant details.

3.1. The MMODE Mutation-Bound Process

The proposed MMODE algorithm introduces a variation to the DE/rand/2 method for donor-specification in the Differential Evolution algorithm of Section 2.2. In a classical Differential Evolution algorithm, when the mutation resulting from Equation (3) produces a donor component that lies outside the search range $\partial\Omega_s$, the component is reassigned to a boundary value. In MMDOE

such out-of-bound donor-vector components are given a second opportunity to generate an in-bounds donor component, hence decreasing the number of individuals concentrating on the boundary. This is accomplished by implementing the alternative *mutation-bound scheme*

$$V_{i,G}^{(q)} = X_{r_1^i,G}^{(q)} - \Phi \left(X_{r_2^i,G}^{(q)} - X_{r_3^i,G}^{(q)} \right) - \Phi \left(X_{r_4^i,G}^{(q)} - X_{r_5^i,G}^{(q)} \right) \quad (6)$$

Finally, if the alternative donor-mutation step given by Equation (6) yields a value that again lies outside component bounds, the donor-vector component
160 value is then treated as in the case of the classical DE algorithm, and it is assigned the value of the closest boundary point.

3.2. The MMODE Preselection Scheme

Algorithm 1 implements a new nondominated sorting and crowding-distance sorting method that we call *MMODE Preselection Scheme*. Unlike prior Pareto
165 multimodal optimization algorithms, the proposed approach uses both a decision-space crowding distance and an objective-space crowding distance metrics.

Fig. 2 shows a conceptual diagram illustrating the phases of the MMODE Preselection Scheme. The *first phase* of the scheme consists of generating a population P' from a given population P of size NP . This is accomplished by
170 a nondominated sorting of the individuals in P to produce the population P' , which in turn is composed of the subsets shown as $R1, R2, \dots, Rt$ in the figure. This sorting is done based on nondomination rankings involving the crowding distance metric in the objective space. The subset $R1$ of P' shown in the figure comprises the best nondominated solutions. The decision variables in the
175 remaining subsets, are ordered by ranking in order of decreasing objective-space crowding-distance.

In the *second phase* of the scheme a population Q of size $\lfloor NP/2 \rfloor$ is selected from the rank-ordered decision variables in P' . If the size of $R1$ is greater than or equal to $\lfloor NP/2 \rfloor$, then population Q is set equal to the first $\lfloor NP/2 \rfloor$ elements
180 in $R1$. Any remaining elements of $R1$ are ignored, and the first phase of the scheme terminates at this point. On the other hand, if the size of $R1$ is smaller

than $\lfloor NP/2 \rfloor$, the algorithm first selects all members of set R1 as members of population Q . Then the remaining members of Q are chosen from the decision variables in the subsequent nondominated sets R2, R3, etc., in their ranking order, until the size of Q reaches $\lfloor NP/2 \rfloor$. In this fashion, the solutions from set R2 are chosen next, followed by those of R3, and so on.

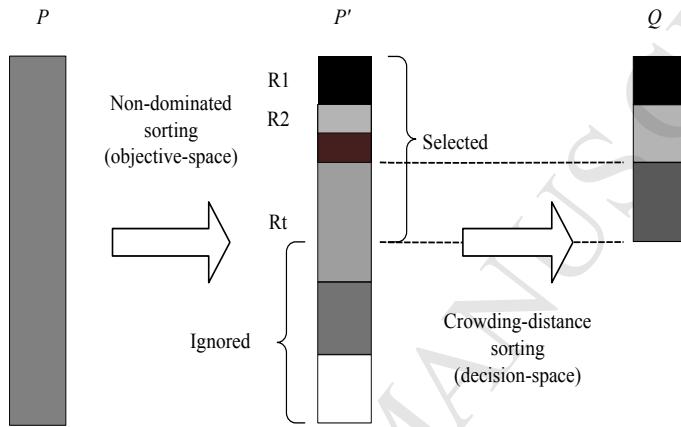


Fig. 2. Conceptual illustration of the phases of the MMODE preselection scheme. The first phase involves a nondominated sorting in the objective space to obtain population P' from P . The second phase implements a crowded-distance sorting in the decision space to obtain population Q from P' .

The *termination* of the second phase is done by carrying out a truncation of the population Q to restrict its size to $\lfloor NP/2 \rfloor$. Let Rt be the first set encountered such that the attempt to incorporate its elements into Q would lead Q to have a size greater than or equal to $\lfloor NP/2 \rfloor$. Also let N_{tot} denote the total number of solutions in sets R1 to Rt. If $N_{tot} = \lfloor NP/2 \rfloor$, then all the solutions in Rt are selected for incorporation into Q , and the second phase is terminated by ignoring all remaining solutions in P' . However, a truncation is implemented in the more general case where $N_{tot} > \lfloor NP/2 \rfloor$. This is done by sorting the solutions in Rt in descending order according to their decision-space crowding distance. The algorithm then chooses as additional elements of Q the top-ranked solutions in the sorted solution-set Rt, until the size of Q reaches

[$NP/2$]. All non-selected decision variables are ignored, and the second phase of the MMODE preselection process is terminated.

²⁰⁰ **4. Pseudocode for the MMODE Algorithm and Discussion**

Algorithm 1 presents pseudocode describing the proposed multimodal multiobjective optimization algorithm with Differential Evolution. The technique involves a preselection scheme used to generate offspring that increase the possibility of finding optimal solutions. In addition, MMODE adjusts the distribution of some mutants to make the decision space more diverse following a Differential Evolution paradigm. The evaluation of each individual is rendered more exact by adopting a maximum number of evaluations as a termination criteria, instead of the more common alternative of using a maximum number of generations. An external archive is used to preserve all superior solutions found. Main algorithm components are devised based on nondominated sorting and on the implementation of a crowding distance selection in both the decision and objective spaces. The algorithm seeks to strike a reasonable balance both in the decision and objective space, for the purpose of enhancing the promise for improved solution distributions in both spaces. The two most salient and novel contributions of the algorithm are the MMODE Preselection Scheme and the Bound Mutation Process, discussed in the ensuing sections.

Algorithm 1. Pseudocode for the Multimodal Multiobjective Optimization with Differential Evolution (MMODE) algorithm.

Input: Population size NP , scaling factor F , crossover rate Cr , maximum number of function evaluations $maxFES$, and search domain boundary $BOUND$

Output: nondominated solutions

1. Generate a random population P of size NP in the decision space
 2. Set $FES = 0$
 3. **While** $FES < maxFES$ **do**
 4. **For** each individual in P calculate its m objective functions
 5. Set $FES = FES + 1$
 6. **End for**
 7. Apply the MMODE Preselection Scheme of Section 3.2
 - (a) Implement the first phase to produce population P'
 - (b) Implement the second phase to produce population Q
 - (c) Terminate the second phase by truncating Q to size $\lfloor NP/2 \rfloor$
 8. **For** each individual in Q
 - (a) Apply the DE mutation process of Equation (3)
 - (b) **If** mutant lies outside $BOUND$
 - (c) Apply the MMODE mutation-bound Process of Equation (6)
 - (d) **End if**
 - (e) Apply the DE crossover process of Equation (4)
 - (f) Apply the DE offspring generation process of Equation (5)
 9. **End for**
 10. Store all offspring in archive R of size $\lfloor NP/2 \rfloor$
 11. **For** each individual in R
 - (a) Calculate its m scalar objective functions
 - (b) Set $FES = FES + 1$
 12. **End for**
 13. Apply a nondominated sorting scheme on $P + R$
 14. Store nondominated solutions in archive E of size NE
 15. **If** $NE < NP$
 16. Add solutions to E until $NE = NP$
 17. **Else**
 18. Remove solutions from E until $NE = NP$
 19. **End if**
 20. $P \leftarrow E$
 21. **End while**
-

5. Experimental Preliminaries

5.1. Performance Metrics

The *inverted generational distance (IGD)* [28, 29] and the *hypervolume difference (IH)* metrics [30, 31] are adopted to evaluate algorithm performance. Usually *IGD* is only utilized in the objective space to measure the distribution of solutions. Here *IGD* is used to evaluate distributions in both the objective and decision spaces, and consequently these metrics are respectively denoted as *IGDf* and *IGDx* [24]. Let P^* be a set of uniformly distributed points in the Pareto front (or Pareto set) and let P be an approximation set to the Pareto front (or Pareto set). The *IGD* metric is defined through the equation

$$IGD(P^*, P) = \sum_{v \in P^*} \frac{d(v, P)}{|P^*|} \quad (7)$$

where v is a point in P^* , $d(v, P)$ is the Euclidean distance between v and P , and $|P^*|$ is the cardinality of P^* . The metric *IDGf* results from Equation (7) when the sets P^* and P are in the objective space (*i.e.*, they represent Pareto fronts), and *IGDx* results when they are in the decision space (they represent Pareto sets).

In this paper the *IH* metric is the hypervolume difference between the set P and a reference point R chosen by user, as defined by the equation

$$I_H(R, P) = \mathcal{I}_H(R) - \mathcal{I}_H(P) \quad (8)$$

where $\mathcal{I}_H(R)$ and $\mathcal{I}_H(P)$ are the hypervolume metrics respectively for the reference point R and for the set P relative to a zero-volume reference [31] in the objective space.

Two important concepts in the evaluation of the performance of algorithms are their *convergence* and *diversity* properties. Both the *IGD* and *IH* metrics have the ability to quantify these two factors. More specifically, better convergence and diversity is obtained when the value of *IDG* is small and the value of *IH* is large.

To realize a small IGD value and a large I_H value, the approximate set P must be close to the actual set P^* and must cover most of the set P^* . This concept applies to the cases where P and P^* are both in the objective space or
²³⁵ both in the decision space.

5.2. Catalog of Test Functions

The geometrical properties of the objective function (such as its convex or concave structure) as well as the dimensionality of the decision space can have a significant effect on the performance of evolutionary optimization algorithms.
²⁴⁰ Okabe *et al.* [32] argue for the necessity of constructing test functions with complicated Pareto-front shapes, though in that work the function shapes are relatively simple and the decision space is only two dimensional. A higher level of complexity is found in Deb and Tiwari [2], which includes a multiobjective multiglobal test problem with nine similar regions in the decision space that correspond to the same Pareto front. Liang and Yue [1, 3] also propose multimodal multiobjective test functions.
²⁴⁵

The authors argue that the types test functions documented in the literature to date are not sufficiently adequate for meaningful performance-testing studies for multimodal multiobjective optimization algorithms. In particular, cases with larger decision-space dimensionality are needed. This paper makes use
²⁵⁰ of twelve objective test functions for performance analysis, labelled F1 - F12, whose mathematical expressions as shown in Table A1 of Appendix A. Table A1 also shows the Pareto set and Pareto front for each function, consistent with the formulation of Theorem 1 in [33]. In particular, functions F1 - F4, F6, and
²⁵⁵ F9 - F12 are introduced in [3]. To address the perceived need, functions F5, F7 and F8 are introduced in this work, with designs inspired from multiobjective optimization test instances first introduced in the *CEC 2009 Special Session and Competition* [34]. The common feature of all twelve test functions is that they have two or more subsets in the Pareto set that map to the same Pareto front.
²⁶⁰

The test functions have different geometric characteristics as described in

Table 1. Relevant characteristics of twelve objective test functions, including the number of Pareto sets, the Pareto front shape, and the dimension of the Pareto front. The last column shows the reference point R used for the hypervolume-difference metric calculation in Equation (8).

Test Function	Number of Pareto Subsets	PF Shape	PF Dimension	Reference Point
F1	2	concave	2	(2,2)
F2	2	concave	2	(2,2)
F3	2	concave	2	(2,2)
F4	9	concave	2	(2,2)
F5	2	convex	2	(2,2)
F6	4	convex	2	(2,2)
F7	2	not convex, not concave	2	(2,2)
F8	2	convex and concave	2	(5,40)
F9	4	concave	2	(2,2)
F10	4	concave	2	(2,2)
F11	2	concave	2	(2,2)
F12	4	convex	2	(2,2)

Table 1. For example, F1 - F4 and F9 - F11 are concave, F5, F6 and F12 are convex, F8 is both concave and convex, and F7 is neither concave nor convex cases. The Pareto set shapes include periodic, symmetric, translational and irregular shapes. Functions F1, F2, F3, F5, F7, F8 and F11 have two Pareto subsets, F6, F9, F10 and F12 have four, and function F4 has nine.

5.3. Alternative Algorithms used for Comparison Studies and Experimental Parameter Settings

Five state-of-the-art evolutionary algorithms are compared with MMODE. They include two existing multimodal multiobjective optimization algorithms: DN-NSGAII [1] and MO-Ring-PSO-SCD [3] (which in the suite is referred to by the acronym MRPS for succinctness of notation). The additional three algorithms considered are NSGAII [35], GAMODE [27] and Omni-Opt [2].

All the numerical optimization experiments are executed on 2.3 GHz Intel core i5 computational engine with 2 GB of random-access memory. The

maximum number of function evaluations $maxFES$ is set to 80,000. Other input-parameter settings are $\Phi = 0.5$ and $Cr = 0.9$. The dimension of the decision space is $n = 2$. All algorithm results are averaged over 31 independent runs. The reference point needed to implement Equation (8) is shown in the
²⁸⁰ last column of Table 1.

6. Experimental Results

6.1. Performance Results under a Constant Population Size

The average values and standard deviations of the $IGDx$, $IGDf$ and I_H metrics are presented in Tables 2-4, where the best value is highlighted in bold
²⁸⁵ font. The population size in all these experiments is kept constant at $NP = 800$. Table 2 shows that more than half of the $IGDx$ results obtained by MMODE are highlighted in bold (indicating the attainment numerous best-value results), while the other algorithms realize a significantly lower number of best values. Table 3 shows that for the case of $IDGf$, most of the results in bold font
²⁹⁰ are realized by the NSGAII algorithm, while the remaining methods have a performance similar to the MMODE method. Table 4 shows that for the case of I_H , most of the reported average values are approximately equal for a given test function (ignoring the standard deviations).

Finally, Tables 5 and 6 show respectively for the $IGDx$ and $IGDf$ cases
²⁹⁵ the rank frequencies for the six compared algorithms. The notation Rank i , $i = 1, 2, \dots, 6$, indicates the number of times (*i.e.*, the frequency) that an algorithm's performance is ranked as 1, 2, 3, and so on, where a rank of 1 designates the best performance, 2 the second-best, and so on. In general, larger frequency values at lower ranks denote better performance. The Total Rank shown on
³⁰⁰ the last column of the tables is obtained by the formula $\sum_{i=1}^6 i \cdot (\text{Rank } i)$, and it represents a weighted rank-frequency. A better performing algorithm is characterized by a lower Total Rank score.

Table 5 shows that MMODE realizes the smallest Total Rank score under the $IDGx$ metric, indicating that it has the best overall performance among the

³⁰⁵ six evolutionary algorithms under consideration, hence outperforming the other methods in the decision-space metric. The MRPS and GAMODE approaches are tied for second place, Omni-Opt and DN-NSGAII stand in third and fourth place, respectively, while NSGAII realizes the lowest performance.

³¹⁰ Table 6 shows that the MMODE algorithm occupies third place in Total Score performance relative to the $IDGf$ function-domain metric. Its Total Score is not significantly different, however, from that of the second-ranked GAMODE or the fourth ranking Omni-Opt algorithms. Hence, the results support the observation that, even though MMODE is not the best performer relative to this metric, its performance is nevertheless similar to that of its ³¹⁵ closest-ranked peers. The NSGAII algorithm is ranked first, hence obtaining the best performance relative to the function-space metric, while algorithm MRPS ranks last in overall performance and DN-NSGAII is ranked in fifth place. The reason why NSGAII has a significantly superior performance is that it only considers the solution distribution in the objective space, without incurring penalties for poor distributions in the decision space. In fact, Table 5 shows ³²⁰ that NSGAII has the worst performance relative to the $IDGx$ metric, hence its superior function-space performance is realized to the significant detriment to its decision-space performance. No rank analysis is done for the I_H metric because all algorithms produce similar values.

³²⁵ In summary, from the analysis of Tables 2-6, it can be concluded that the MMODE approach distinguishes itself as being the method that most successfully finds a large number (and in some cases all) the subsets in the Pareto set, relative to the other five algorithms studied. The method outperforms the other algorithms in terms of the $IDGx$ metric, and shows similar performance to that of the top performers in terms of the $IDGf$ metric. Although outperformed only by the NSGAII algorithm in the $IDGf$ metric tests, the latter is nevertheless the poorest performer in the $IDGx$ metric, making it therefore inferior to the MMODE technique when good distributions in both the decision and objective space are important. Hence, it can be concluded that the MMODE algorithm ³³⁰ outperforms its competitors without incurring a significant degradation of ³³⁵

Table 2. Average $IGDx$ values for the six compared algorithms on test functions F1 - F12.

Test Fxn.	MMODE mean \pm std dev	DN-NSGAII mean \pm std dev	MRPS mean \pm std dev	NSGAII mean \pm std dev	GAMODE mean \pm std dev	Omni-Opt mean \pm std dev
F1	1.37E-02 \pm 4.70E-04	1.98E-02 \pm 2.84E-04	1.52E-02 \pm 3.34E-04	5.36E-02 \pm 5.89E-03	1.48E-02 \pm 2.10E-04	2.17E-02 \pm 6.54E-03
F2	9.81E-03 \pm 4.14E-03	3.41E-02 \pm 1.14E-02	1.06E-02 \pm 2.69E-03	2.62E-02 \pm 6.14E-03	7.57E-03 \pm 2.04E-04	1.94E-02 \pm 5.03E-03
F3	7.83E-03 \pm 2.88E-03	1.46E-02 \pm 2.50E-03	7.20E-03 \pm 4.87E-04	3.04E-02 \pm 7.94E-03	5.77E-03 \pm 2.99E-03	1.61E-02 \pm 8.43E-03
F4	6.23E-03 \pm 2.66E-04	3.36E-02 \pm 2.23E-02	1.18E-02 \pm 3.91E-04	6.54E-01 \pm 6.32E-01	2.80E-01 \pm 1.29E-01	1.45E-02 \pm 3.06E-02
F5	8.30E-03 \pm 4.31E-04	7.99E-03 \pm 1.20E-03	1.01E-02 \pm 2.16E-04	1.01E-01 \pm 2.57E-01	1.41E-02 \pm 1.46E-02	7.74E-03 \pm 2.67E-04
F6	1.44E-02 \pm 9.36E-04	5.38E-02 \pm 2.03E-02	8.70E-03 \pm 2.19E-04	8.06E-02 \pm 5.15E-02	2.04E-02 \pm 6.58E-03	2.53E-02 \pm 1.20E-02
F7	1.04E-02 \pm 2.23E-04	1.17E-02 \pm 4.70E-04	1.40E-02 \pm 1.64E-04	1.08E-02 \pm 1.56E-04	1.07E-02 \pm 2.07E-04	1.04E-02 \pm 2.94E-04
F8	1.63E-02 \pm 5.24E-04	1.76E-02 \pm 1.20E-03	2.00E-02 \pm 4.50E-04	1.78E-02 \pm 4.69E-02	1.62E-02 \pm 1.35E-04	1.75E-02 \pm 5.61E-04
F9	2.93E-02 \pm 9.57E-04	6.73E-02 \pm 6.50E-03	3.09E-02 \pm 8.27E-04	7.53E-02 \pm 6.65E-03	3.25E-02 \pm 2.37E-04	7.05E-02 \pm 8.05E-03
F10	2.77E-02 \pm 9.74E-04	5.70E-02 \pm 6.12E-03	2.81E-02 \pm 7.53E-04	7.63E-02 \pm 3.57E-03	4.14E-02 \pm 3.64E-04	5.54E-02 \pm 7.77E-03
F11	8.14E-03 \pm 2.92E-04	1.12E-02 \pm 2.64E-03	9.20E-03 \pm 1.23E-04	3.33E-02 \pm 3.60E-03	1.14E-02 \pm 7.44E-04	9.65E-03 \pm 7.43E-04
F12	1.69E-02 \pm 9.44E-04	6.28E-02 \pm 2.75E-02	2.14E-02 \pm 1.12E-03	2.17E+00 \pm 6.89E-01	5.29E-02 \pm 6.47E-02	7.36E-02 \pm 4.73E-03

distribution of solutions in the objective space.

6.2. Exact and Numerically Obtained Pareto Sets and Fronts

Table A1 in Appendix 1 gives analytical expressions for the Pareto set and Pareto front for each of the twelve test functions. Appendix A also includes figures for these analytical expressions for the case of a decision-space dimension $n = 2$. More precisely, Fig. A1 shows graphs of the Pareto sets for all twelve test functions using the analytical expressions given in Table A1 (see the mathematical entry following the row header "PS:" in the table). Likewise, Fig. A2 shows graphs of the Pareto front for all the test functions (obtained using the mathematical expressions following the row header "PF:" in Table A1).

Table 3. Average $IGDf$ values for the six compared algorithms on test functions F1 - F12.

Test Fxn.	MMODE mean \pm std dev	DN-NSGAII mean \pm std dev	MRPS mean \pm std dev	NSGAII mean \pm std dev	GAMODE mean \pm std dev	Omni-Opt mean \pm std dev
F1	6.63E-04 \pm 1.38E-05	8.13E-04 \pm 1.60E-05	1.00E-03 \pm 2.76E-05	6.11E-04 \pm 8.32E-06	6.81E-04 \pm 1.00E-05	7.10E-04 \pm 1.32E-05
F2	2.57E-03 \pm 2.85E-04	8.44E-04 \pm 7.48E-05	4.80E-03 \pm 3.27E-04	6.13E-04 \pm 2.98E-05	4.72E-03 \pm 1.13E-03	6.62E-04 \pm 2.56E-05
F3	2.34E-03 \pm 3.03E-04	6.815E-04 \pm 2.34E-05	3.90E-03 \pm 1.49E-04	6.13E-04 \pm 2.04E-05	3.68E-03 \pm 2.53E-04	7.19E-04 \pm 4.77E-05
F4	1.71E-03 \pm 3.06E-04	1.87E-03 \pm 1.17E-05	2.71E-03 \pm 4.06E-05	1.22E-03 \pm 6.66E-05	1.29E-03 \pm 4.40E-05	1.55E-03 \pm 3.58E-05
F5	6.83E-04 \pm 2.23E-05	7.84E-04 \pm 2.31E-05	9.45E-04 \pm E2.07-05	6.21E-04 \pm 2.35E-05	6.32E-04 \pm 9.43E-06	6.33E-04 \pm 2.39E-05
F6	6.55E-04 \pm 2.08E-05	7.91E-04 \pm 1.77E-05	9.42E-04 \pm 2.98E-05	6.04E-04 \pm 1.09E-05	6.11E-04 \pm 9.42E-06	6.80E-04 \pm 1.06E-05
F7	3.64E-03 \pm 7.33E-05	4.40E-03 \pm 1.01E-04	5.02E-03 \pm 6.79E-05	3.96E-03 \pm 4.45E-05	3.60E-03 \pm 9.27E-05	4.22E-03 \pm 1.16E-04
F8	1.95E-02 \pm 4.89E-04	2.72E-02 \pm 7.63E-04	2.66E-02 \pm 4.50E-05	2.08E-02 \pm 3.55E-03	1.96E-02 \pm 6.71E-04	2.42E-02 \pm 4.86E-04
F9	6.44E-04 \pm 1.18E-05	7.93E-04 \pm 2.43E-05	9.71E-04 \pm 1.13E-05	6.19E-04 \pm 8.80E-06	6.69E-04 \pm 1.74E-05	6.65E-04 \pm 5.02E-06
F10	6.18E-04 \pm 1.39E-05	7.86E-04 \pm 2.57E-05	9.10E-04 \pm 1.37E-05	6.11E-04 \pm 7.61E-06	6.36E-04 \pm 1.96E-05	6.08E-04 \pm 3.58E-05
F11	6.10E-04 \pm 1.13E-05	9.71E-04 \pm 4.53E-05	8.84E-04 \pm 2.00E-05	6.03E-04 \pm 1.07E-05	7.42E-04 \pm 9.09E-06	7.44E-04 \pm 1.10E-05
F12	9.25E-04 \pm 5.22E-05	8.76E-04 \pm 4.75E-05	1.30E-03 \pm 4.89E-05	6.49E-04 \pm 2.66E-05	6.46E-04 \pm 2.91E-05	7.36E-04 \pm 9.12E-06

This work investigates the effect of the population size on the numerical results of each of the six optimization algorithms considered under a decision space dimension set as $n = 2$. The population size NP is set to 50, 100, 200, 400 and 800, in successive numerical experiments. Given the very large number of figures that result from these simulation studies, for brevity of exposition the figures reported in the Appendix include only functions F3, F4, and F12, which are of particular interest because their Pareto sets respectively comprising 2, 9, and 4 Pareto subsets. Figures B1 - B3 show the numerically obtained Pareto set while B4 - B6 show the corresponding numerically obtained Pareto front.
 350

The success of each numerical optimization algorithm on the identification

Table 4. Average I_H values for the six compared algorithms on test functions F1 - F12.

Test	MMODE	DN-NSGAII	MRPS	NSGAII	GAMODE	Omni-Opt
Fxn	mean	mean	mean	mean	mean	mean
	\pm std dev	\pm std dev	\pm std dev	\pm std dev	\pm std dev	\pm std dev
F1	3.67E+00 ± 3.59E-04	3.67E+00 ± 2.40E-05	3.66E+00 ± 4.20E-04	3.67E+00 ± 5.18E-06	3.66E+00 ± 4.38E-04	3.67E+00 ± 1.63E-05
F2	3.66E+00 ± 3.68E-03	3.67E+00 ± 1.33E-04	3.65E+00 ± 2.41E-03	3.67E+00 ± 2.72E-05	3.64E+00 ± 1.29E-02	3.67E+00 ± 3.01E-05
F3	3.66E+00 ± 6.76E-03	3.67E+00 ± 6.59E-05	3.65E+00 ± 7.29E-03	3.67E+00 ± 2.11E-05	3.65E+00 ± 4.30E-03	3.67E+00 ± 5.37E-05
F4	8.14E+00 ± 3.73E-05	8.14E+00 ± 5.00E-05	8.14E+00 ± 4.50E-05	8.14E+00 ± 3.82E-05	8.14E+00 ± 1.10E-04	8.14E+00 ± 7.38E-05
F5	3.21E+00 ± 7.70E-04	3.21E+00 ± 2.81E-03	3.21E+00 ± 4.30E-5	3.21E+00 ± 1.46E-05	3.21E+00 ± 3.61E-04	3.21E+00 ± 3.67E-05
F6	3.21E+00 ± 7.36E-04	3.21E+00 ± 2.43E-03	3.33E+00 ± 3.64E-04	3.21E+00 ± 6.65E-05	3.21E+00 ± 1.52E-04	3.21E+00 ± 8.36E-05
F7	1.53E+00 ± 3.37E-03	1.51E+00 ± 9.92E-04	1.51E+00 ± 1.91E-03	1.52E+00 ± 6.77E-04	1.53E+00 ± 3.28E-03	1.51E+00 ± 6.23E-04
F8	3.75E+02 ± 4.48E-02	3.75E+02 ± 9.31E-03	3.75E+02 ± 1.88E-02	3.75E+02 ± 2.66E-02	3.75E+02 ± 7.43E-02	3.75E+02 ± 7.26E-04
F9	3.67E+00 ± 4.51E-04	3.67E+00 ± 2.34E-04	3.67E+00 ± 4.50E-05	3.67E+00 ± 8.55E-05	3.66E+00 ± 4.12E-04	3.67E+00 ± 4.41E-04
F10	3.67E+00 ± 3.66E-04	3.66E+00 ± 1.12E-03	3.66E+00 ± 2.27E-04	3.67E+00 ± 3.74E-06	3.67E+00 ± 2.37E-04	3.67E+00 ± 1.26E-05
F11	3.67E+00 ± 2.08E-04	3.66E+00 ± 9.35E-04	3.67E+00 ± 2.87E-04	3.67E+00 ± 9.31E-06	3.67E+00 ± 1.43E-04	3.67E+00 ± 1.56E-05
F12	3.21E+00 ± 9.31E-04	3.21E+00 ± 8.78E-04	3.21E+00 ± 5.75E-04	3.21E+00 ± 6.60E-04	3.21E+00 ± 6.57E-04	3.21E+00 ± 1.72E-04

of the actual Pareto set or front of each test function may be qualitatively assessed by the reader by comparing the analytically obtained graphs reported in Figs. A1 and A2 with the corresponding numerically obtained graphs shown in Figs B1 - B6. The result of such qualitative analysis for the three selected test functions shows that, for large population numbers, the MMODE algorithm is able to produce a better distributed set of numerically obtained Pareto-sets and Pareto-front solutions relative to the distribution for the five alternative algorithm considered. The conclusions obtained are also applicable to the test functions not included in Appendix B. The next subsection presents a quantitative study on the performance of the algorithms as a function of population

Table 5. Rank frequencies for the six compared algorithms on F1 - F12 in terms of $IGDx$.

Algorithm	Rank	Rank	Rank	Rank	Rank	Rank	Total
	1	2	3	4	5	6	Rank
MMODE	7	3	2	0	0	0	19
DN-NSGAI	0	1	0	7	3	1	51
MRPS	1	6	2	1	0	2	35
NSGAI	0	0	0	1	2	9	68
GAMODE	3	1	5	0	3	0	35
Omni-Opt	1	1	3	3	4	0	44

Table 6. Rank frequencies for the six algorithms compared on F1 - F12 in terms of $IGDf$.

Algorithm	Rank	Rank	Rank	Rank	Rank	Rank	Total
	1	2	3	4	5	6	Rank
MMODE	1	4	2	4	1	0	36
DN-NSGAI	0	1	1	1	7	2	56
MRPS	0	0	0	0	2	10	70
NSGAI	8	2	2	0	0	0	18
GAMODE	2	4	2	2	2	0	34
Omni-Opt	1	1	5	5	0	0	38

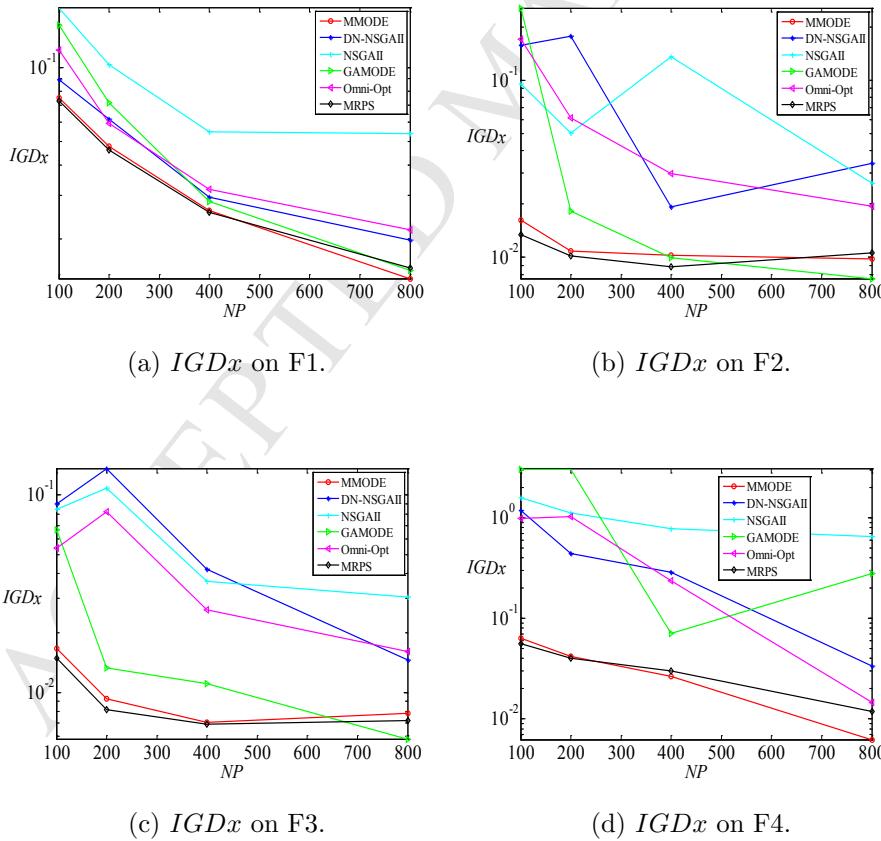
size, for all twelve test functions.

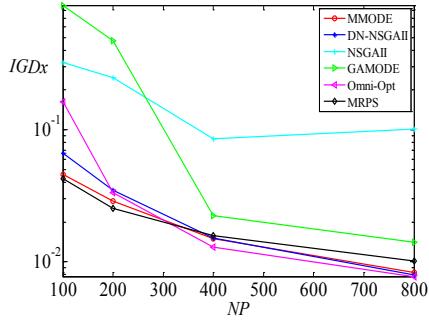
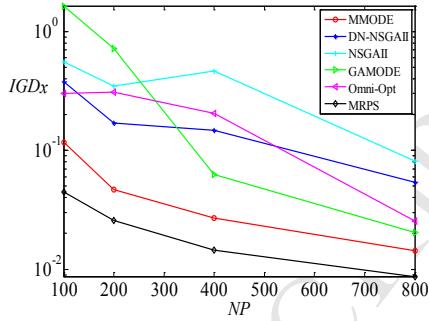
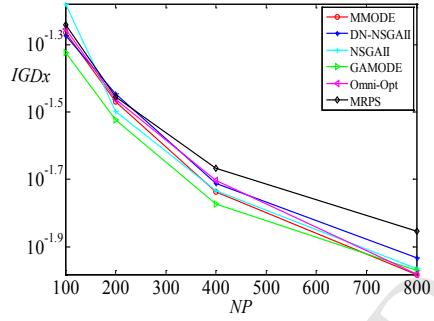
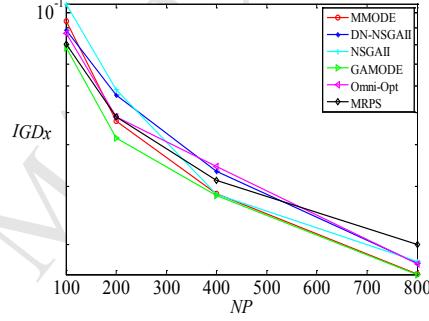
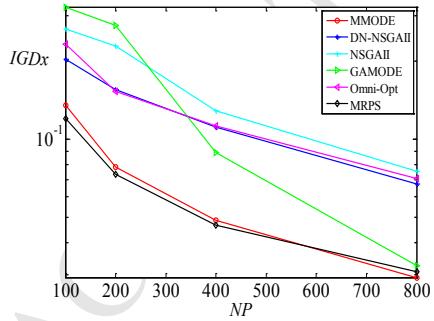
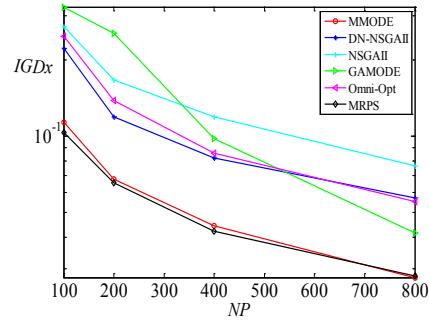
6.3. Effect of the Population Size on Performance

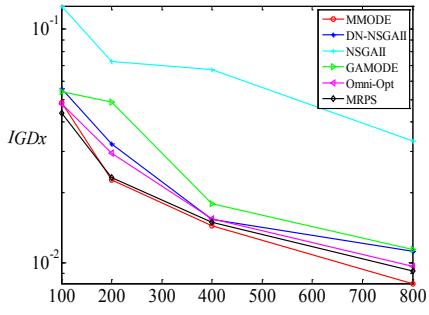
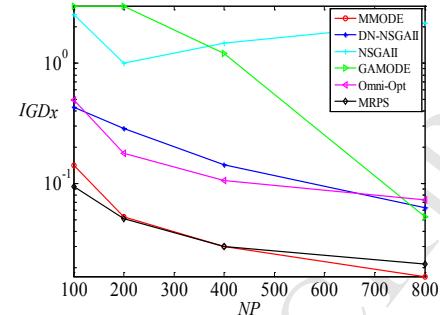
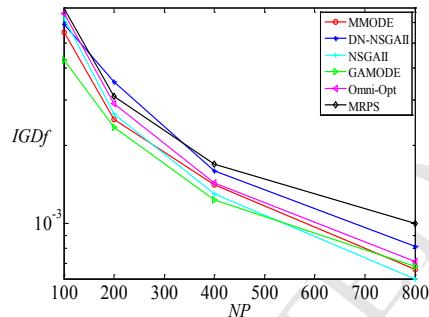
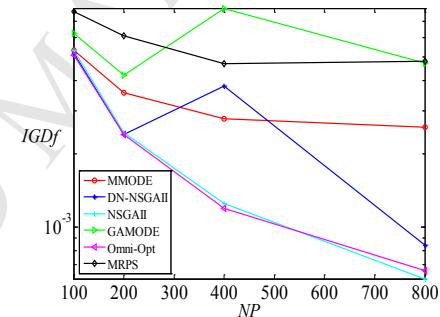
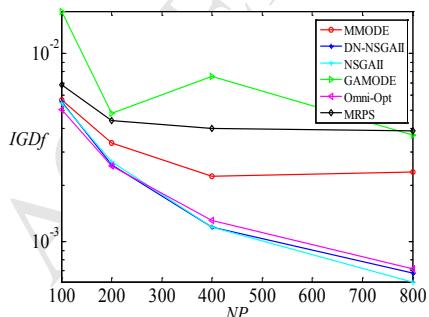
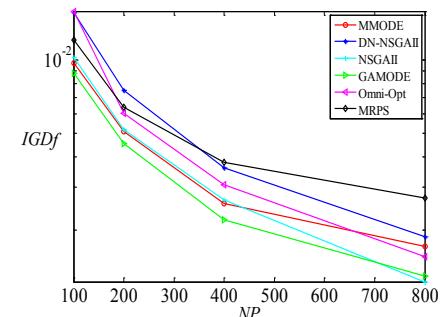
The numerical results obtained by each of the twelve algorithms for test functions F1 - F12 yield quantitative relationships between the population size NP and performance metrics $IGDx$, $IGDf$, and I_H . Figures 3 - 5 show the results in graphical form. Fig. 3 shows that when focusing on the larger population size of $NP = 800$ it becomes evident that the MMODE algorithm realizes the best $IDGx$ values, ranking as the best performer for a majority of the twelve test functions, and never ranking worse than third for any of the functions. This indicates that MMODE obtains excellent decision-space distributions of solutions. Fig. 4 shows that, when $NP = 800$, MMODE typically ranks among the top-3 best performers under the $IDGf$ metric, though not occupying the

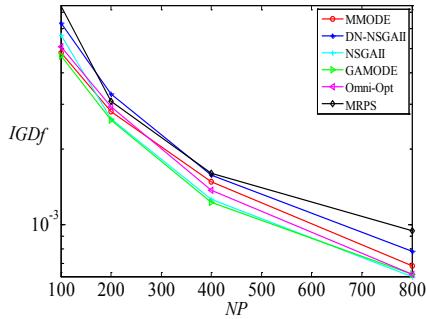
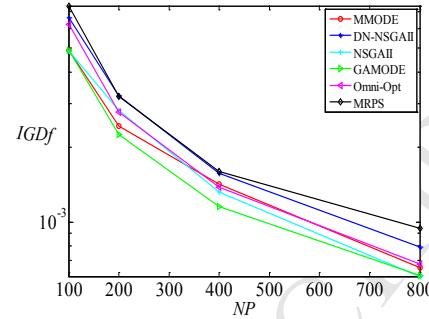
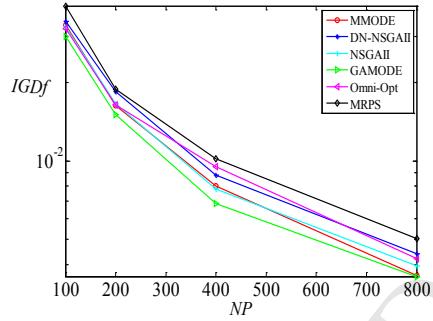
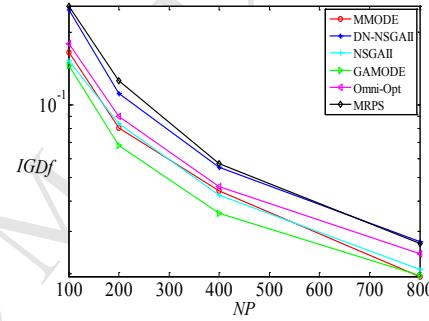
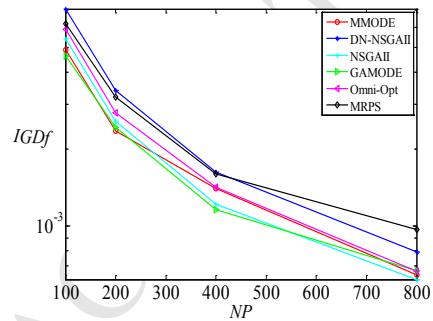
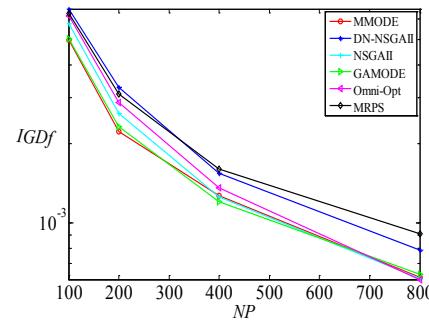
rank-1 place in most cases. This indicates that the function-space distribution of solutions produced by MMODE is not inferior to that of the top performing algorithms considered. Finally, Fig. 5 shows that at $NP = 800$ the performance of MMODE under the I_H metric is always comparable to that of the best performing algorithms when taking in consideration the small magnitudes reported on the y-axis of the graphs.

In summary, it can be concluded that, when the population size is large,
385 MMODE shows overall superior performance relative to the other five algorithms considered as determined using under the $IDGx$, $IDGf$ and I_H metrics for characterizing the diversity of solutions in the decision and objective space. A larger number and a more uniform distribution of Pareto set solutions can be found and maintained.



(e) $IGDx$ on F5.(f) $IGDx$ on F6.(g) $IGDx$ on F7.(h) $IGDx$ on F8.(i) $IGDx$ on F9.(j) $IGDx$ on F10.

(k) $IGDx$ on F11.(l) $IGDx$ on F12.**Fig. 3.** Average $IGDx$ values for the six compared algorithms under different population sizes.(a) $IGDf$ on F1.(b) $IGDf$ on F2.(c) $IGDf$ on F3.(d) $IGDf$ on F4.

(e) $IGDf$ on F5.(f) $IGDf$ on F6.(g) $IGDf$ on F7.(h) $IGDf$ on F8.(i) $IGDf$ on F9.(j) $IGDf$ on F10.

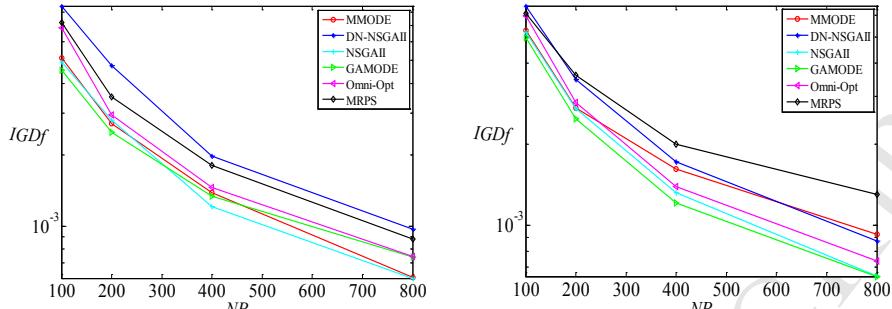
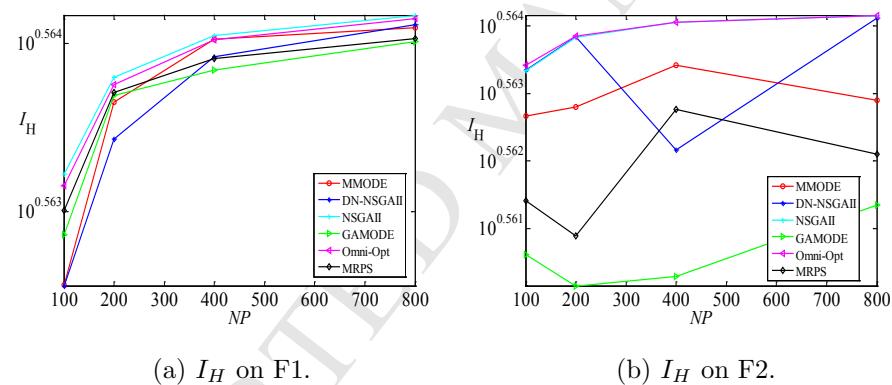
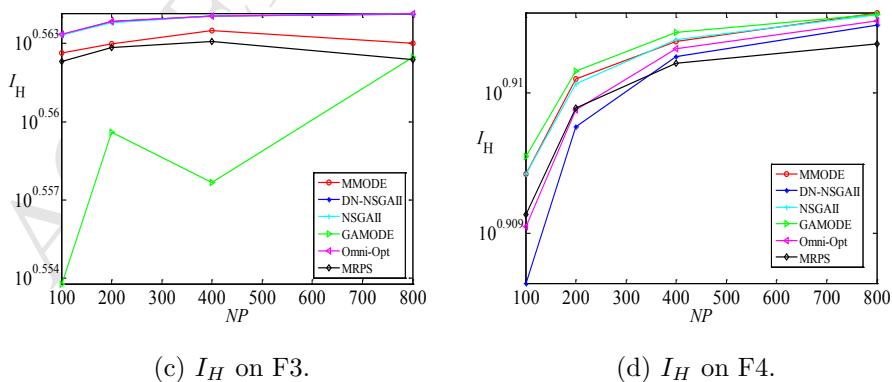
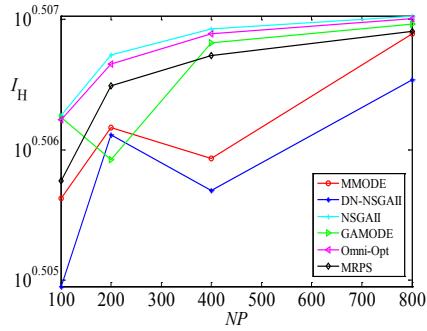
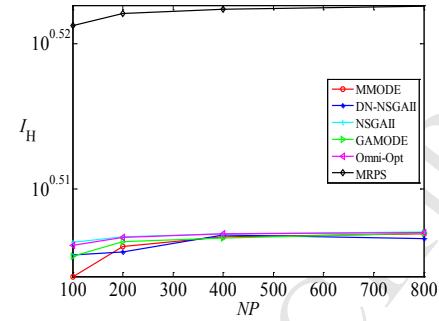
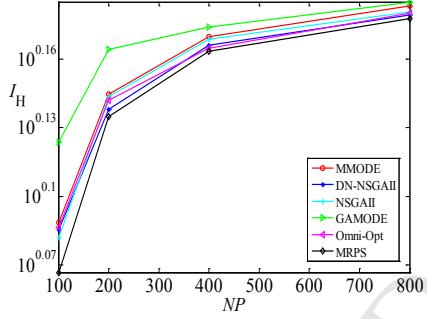
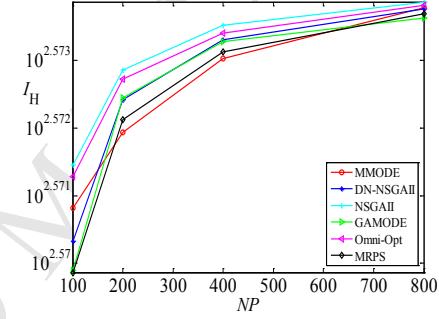
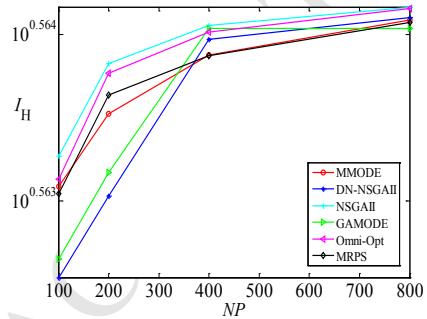
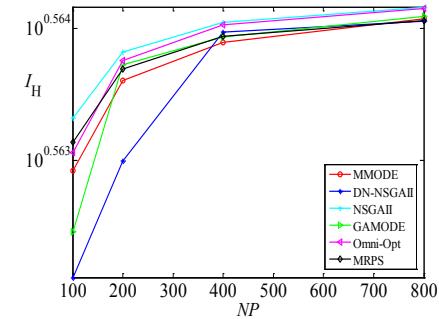
(k) $IGDf$ on F11.(l) $IGDf$ on F12.

Fig. 4. Average $IGDf$ values for the six compared algorithms under different population sizes.

(a) I_H on F1.(b) I_H on F2.(c) I_H on F3.(d) I_H on F4.

(e) I_H on F5.(f) I_H on F6.(g) I_H on F7.(h) I_H on F8.(i) I_H on F9.(j) I_H on F10.

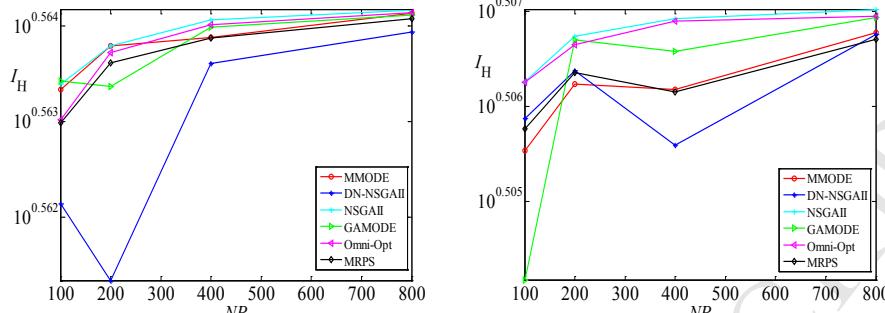
(k) I_H on F11.(l) I_H on F12.

Fig. 5. Average I_H values for the six compared algorithms under different population sizes.

390 6.4. Analysis of the Effectiveness of the Preselection and Mutation-Bound Schemes

The MMODE algorithm described in Section 3 includes two novel evolutionary steps, namely a preselection and a mutation-bound scheme. The proposed MMODE algorithm incorporates both schemes. To assess the relative effectiveness of new schemes three additional versions of MMODE, denoted as MMODE1, MMODE2, and MMODE3, are coded. The MMODE1 version includes only the mutation-bound scheme, MMODE2 incorporates only the preselection scheme, and MMODE3 uses neither the preselection and mutation-bound schemes.

Fig. 6 shows bar graphs quantifying the resulting performance of the four versions of MMODE under consideration when they are deployed on the twelve test functions using the $IDGx$, $IDGf$, and I_H metrics. To make the study more meaningful in terms of comparable scales, the values of I_H for test function F8 are reported in Fig. 6(c) after division by a factor of 100.

Fig. 6(a) shows that MMODE attains the best $IGDx$ metric relative to its three variations. This shows that the inclusion of both the preselection scheme and the mutation-bound scheme featured in MMODE have an advantageous effect on the resulting performance in terms of this metric. The figure can also be used to conclude that the incorporation of the mutation-bound scheme also

leads to better performance because MMODE1 is an overall better performer
 410 than MMODE3. Likewise, the incorporation of the preselection scheme also has a favorable impact because the performance of MMODE2 is also better than that of MMODE3. The same conclusion is arrived when considering the $IGDf$ metric as reported in Fig. 6(b), where it is also clear that the addition of preselection and bound-mutation steps improve the metric. Fig. 6(b) also
 415 shows that in general the adoption of the preselection scheme only is more impactful than the addition of the mutation-bound step only, as the MMODE2 metrics are typically lower than the MMODE1 metrics. In all cases, however, the addition of both evolutionary steps leads to improved performance over the other alternatives considered. Finally, Fig. 6(c) also shows that the two-scheme
 420 MMODE algorithm has superior performance relative to the variants considered, though the differences are of a minor nature. In summary, it is concluded that the preselection and the mutation-bound schemes have a favorable impact on the performance of the MMODE algorithm.

6.5. Influence of the Distance Between Pareto Subsets on MMODE Performance

When the distance between Pareto subsets is small, it is well known through experience that traditional evolutionary algorithms, such as NSGAII, tend to remove some optimal solutions in both the decision and the objective space.
 430 The preselection scheme defined in Section 3.2 is introduced in MMODE for the explicit purpose of mitigating the loss of solutions from closely-spaced but otherwise distinct Pareto subsets.

The MMODE algorithm has proven to be able to yield good results for test functions such as F2, which features two Pareto subsets. Nevertheless it is expected that performance quality also may decrease as the inter Pareto-subset distance become smaller. This conjecture is experimentally confirmed by adjusting the parameter d shown in Table A1 for function F2, which is the distance between the Pareto subsets for this function. The results are shown in Fig. 7, where the $IDGx$ and $IDGf$ performance metrics for MMODE are plotted as a

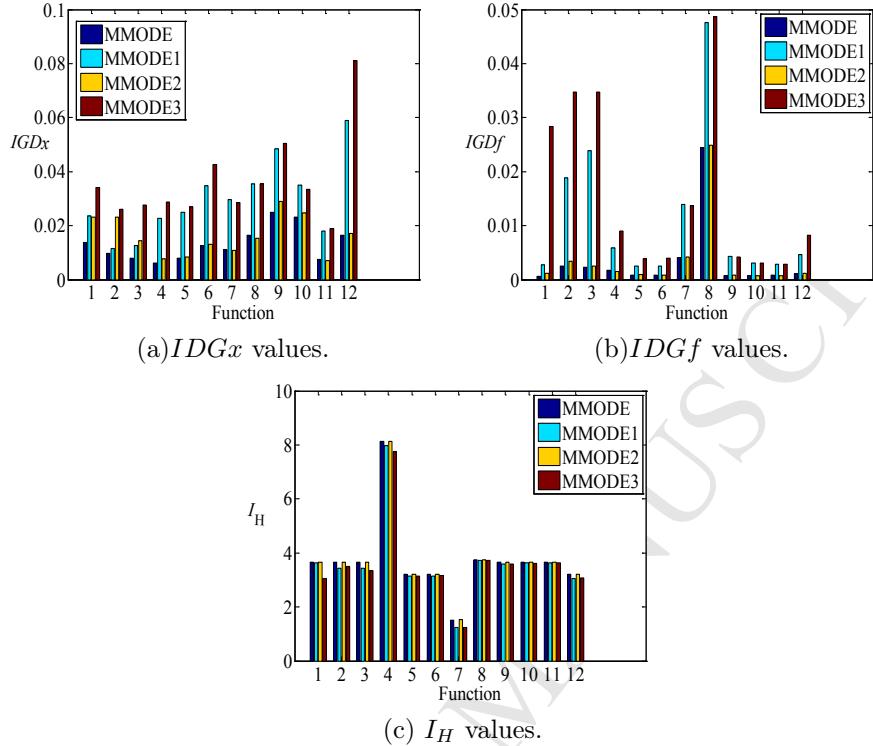


Fig. 6. Average $IGDx$, $IGDf$ and I_H values on test functions F1 - F12 obtained by the MMODE algorithm and its variations MMODE1 (mutation-bound scheme only), MMODE2 (preselection scheme only), and MMODE3 (no preselection nor mutation bound schemes).

function the inter subset distance, where d adopts the values 0, 1/6, 1/3, 1/2,
440 2/3 and 5/6. The figure shows that for both metrics the performance increases with distance, as anticipated. It is concluded that the distance between multiple Pareto subsets in the decision space indeed has an effect on the performance of MMODE, and that in general, the closer the multiple subsets are, the harder it is to solve the underlying multimodal multiobjective optimization problem.

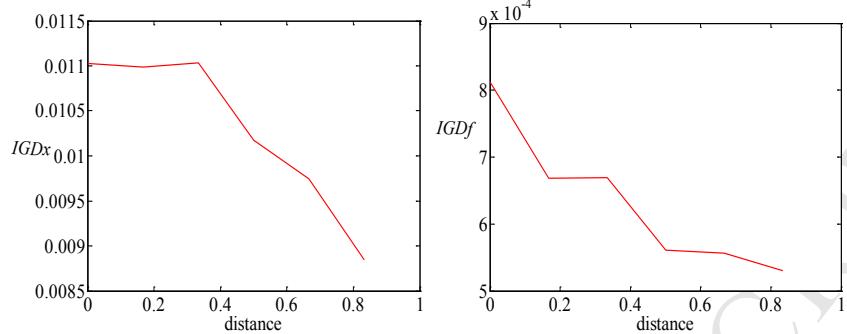


Fig. 7. Effect of the distance between the two Pareto subsets of test function F2 on the IDG_x and IDG_f metrics from MMODE.

445 7. Conclusions and Future Work

A multimodal multiobjective Differential Evolution optimization algorithm (MMODE) is proposed. In the proposed algorithm, a preselection scheme is designed to find more Pareto optimal solutions. Moreover, a mutation-bound precessing method is used to improve the distribution of the population. Novel multimodal multiobjective test functions are also designed in this paper. The experimental results demonstrate that the proposed MMODE algorithm outperforms all the five compared algorithms. Although the performance in the decision space is advantageous, the performance in the objective space still has room for improvement. In our future work, we will seek to further improve the distribution in objective space and try to solve multimodal many objective problems.

Acknowledgments

This work was supported by the *National Natural Science Foundation of China* (61473266, 61673404, 61876169, 61806179), the *Research Award Fund for Outstanding Young Teachers in Henan Provincial Institutions of Higher Education of China* (2014GGJS-004, 2016GGJS-094), the *Program for Science*

and Technology Innovation Talents in Universities of Henan Province in China (16HASTIT041), and the *China Postdoctoral Science Foundation* (2017M622373).

References

- 465 [1] J. Liang, C. Yue, B. Qu, Multimodal multi-objective optimization: A preliminary study, in: Proceedings of the 2016 IEEE Evolution Computation Congress, 2016, pp. 2454–2461.
- 470 [2] K. Deb, S. Tiwari, Omni-optimizer: A procedure for single and multi-objective optimization, in: Proceedings of the third International Evolutionary Multi-Criterion Optimization Conference, Guanajuato, Mexico, March, 2005, pp. 47–61.
- [3] C. Yue, B. Qu, J. Liang, A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems, *IEEE Transactions on Evolutionary Computation* PP (99) (2017) 1–1.
- 475 [4] R. Storn, On the usage of differential evolution for function optimization, in: Proceedings of North American Fuzzy Information Processing, 1996, pp. 519–523.
- 480 [5] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (4) (1997) 341–359.
- [6] I. F. Sbalzarini, S. Muller, P. Koumoutsakos, Multiobjective optimization using evolutionary algorithms, in: Summer Program, 2001, pp. 63–74.
- 485 [7] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, *Evolut. Comput.* 10 (3) (2002) 263.
- [8] E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective search, *Lecture Notes in Computer Science* 3242 (2004) 832–842.

- [9] C. Gil, A. Márquez, R. Baños, M. G. Montoya, J. Gómez, A hybrid method for solving multi-objective global optimization problems, *Journal of Global Optimization* 38 (2) (2007) 265–281.
- [10] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, *Evolut. Comput.* 15 (1) (2007) 1–28.
- [11] R. C. Purshouse, P. J. Fleming, On the evolutionary optimization of many conflicting objectives, *IEEE Trans. Evolut. Comput.* 11 (6) (2007) 770–784.
- [12] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, *IEEE Trans. Evolut. Comput.* 12 (3) (2008) 269–283.
- [13] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, R. Misra, Dominance-based multiobjective simulated annealing, *IEEE Trans. Evolut. Comput.* 12 (3) (2008) 323–342.
- [14] J. Knowles, D. Corne, The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation, in: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Vol. 1, IEEE, 1999, pp. 98–105.
- [15] G. G. Yen, H. Lu, Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation, *IEEE Trans. Evolut. Comput.* 7 (3) (2003) 253–274.
- [16] C. A. C. Coello, G. T. Pulido, M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evolut. Comput.* 8 (3) (2004) 256–279.
- [17] K. C. Tan, Y. J. Yang, C. K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, *IEEE Trans. Evolut. Comput.* 10 (5) (2006) 527–549.

- [18] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evolut. Comput.* 11 (6) (2007) 712–731.
- [19] K. Tahernezhad, K. B. Lari, A. Hamzeh, S. Hashemi, A multi-objective evolutionary algorithm based on complete-linkage clustering to enhance the solution space diversity, in: *CSI International Symposium on Artificial Intelligence and Signal Processing*, 2012, pp. 128–133.
- [20] I. Giagkiozis, P. J. Fleming, Pareto front estimation for decision making, *Evolut. Comput.* 22 (4) (2014) 651–678.
- [21] Y. Liu, D. Gong, S. Jing, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, *IEEE Trans. on Cybern.* pp. (99) (2017) 1–14.
- [22] M. Preuss, B. Naujoks, G. Rudolph, Pareto set and EMOA behavior for simple multimodal multiobjective functions, *Lecture Notes in Computer Science* 4193 (2006) 513–522.
- [23] G. Rudolph, B. Naujoks, M. Preuss, *Capabilities of EMOA to detect and preserve equivalent Pareto subsets*, Springer Berlin Heidelberg, 2007.
- [24] A. Zhou, Q. Zhang, Y. Jin, Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm, *IEEE Trans. Evolut. Comput.* 13 (5) (2009) 1167–1189.
- [25] K. Price, R. M. Storn, J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media, 2006.
- [26] S. Das, A. Abraham, A. Konar, Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives, in: *Advances of computational intelligence in industrial systems*, Springer, 2008, pp. 1–38.
- [27] J. Cheng, G. G. Yen, G. Zhang, A grid-based adaptive multi-objective differential evolution algorithm, *Information Sciences* 367 (2016) 890–908.

- [28] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, *IEEE Trans. Evolut. Comput.* 12 (1) (2008) 41–63.
- [29] M. Reyes Sierra, C. A. C. Coello, A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization, in: Proceedings of the 2005 IEEE Evolution Computation Congress, Vol. 1, 2005, pp. 65–72.
- [30] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, *Tik Report* 214 (2006) 327–332.
- [31] T. Friedrich, F. Neumann, C. Thyssen, Multiplicative approximations, optimal hypervolume distributions, and the choice of the reference point, *Evolut. Comput.* 23 (1) (2013) 131–159.
- [32] T. Okabe, Y. Jin, M. Olhofer, B. Sendhoff, On test functions for evolutionary multi-objective optimization, in: Proceedings of the 2004 International Parallel Problem Solving From Nature Conference, 2004, pp. 792–802.
- [33] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evolut. Comput.* 13 (2) (2009) 284–302.
- [34] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari, Multi-objective optimization test instances for the CEC 2009 special session and competition, University of Essex.
- [35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evolut. Comput.* 6 (2) (2002) 182–197.

565 **Appendix A: Test Objective Function Details**

Table A1. Definition of test functions F1 - F12, along with their Pareto set (PS) and Pareto front (PF).

Test Function	Objectives, PS and PF
F1	$f_1 = \sum_{i=1}^{n-1} x_i - 2 , 1 \leq x_i \leq 3, i = 1, 2, \dots, n-1, -1 \leq x_n \leq 1$ $f_2 = 1 - \sqrt{\sum_{i=1}^{n-1} x_i - 2 } + 2(x_n - \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi))^2$ PS: $x_n = \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi)$ PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
F2	$f_1 = x_1, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$ $(1) f_2 = 1 - \sqrt{x_1} + 2(4(x_2 - \sqrt{x_1})^2 - 2 \cos(\frac{20(x_2 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2)$ $(2) f_2 = 1 - \sqrt{x_1} + 2(4(x_2 - d - \sqrt{x_1})^2 - 2 \cos(\frac{20(x_2 - 1 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2), d = 1$ PS: $x_1 = x_2^2, 0 \leq x_2 \leq 1, x_1 = (x_2 - 1)^2, 1 < x_2 \leq 2$ PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
F3	$f_1 = x_1, (1) 0.5 < x_2 \leq 1 \text{ if } x_1 \in (0.25, 1]; \text{ otherwise } 0 \leq x_2 \leq 0.5$ $(2) 0.5 < x_2 \leq 1 \text{ if } x_1 \in [0, 0.25]; \text{ otherwise } 1 \leq x_2 \leq 1.5$ $(1) f_2 = 1 - \sqrt{x_1} + 2(4(x_2 - \sqrt{x_1})^2 - 2 \cos(\frac{20(x_2 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2)$ $(2) f_2 = 1 - \sqrt{x_1} + 2(4(x_2 - 0.5 - \sqrt{x_1})^2 - 2 \cos(\frac{20(x_2 - 0.5 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2)$ PS: $x_1 = x_2^2, 0 \leq x_2 \leq 1, x_1 = (x_2 - 0.5)^2, 0.5 < x_2 \leq 1.5$ PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
F4	$f_1 = \sum_{i=1}^n \sin(\pi x_i), 0 \leq x_i \leq 6, i = 1, 2, \dots, n$ $f_2 = \sum_{i=1}^n \cos(\pi x_i)$ PS: $2m + 1 \leq x_i \leq 2m + 3/2, m \text{ is an integer}$ PF: $f_2 = -\sqrt{n^2 - f_1^2}, -n \leq f_1 \leq 0$
F5	$f_1 = \sin(x_1), -\pi < x_1 < \pi, 0 \leq x_2 \leq 1$ $f_2 = \sqrt{1 - \sin(x_1)^2} + 2(x_2 - \sin(x_1))^2$ PS: $x_2 = \sin(x_1)$ PF: $f_2 = \sqrt{1 - f_1}, 0 \leq f_1 \leq 1$
F6	$f_1 = \sin(x_1), -\pi < x_1 < \pi, (1) 0 \leq x_2 \leq 1, (2) 1 < x_2 \leq 2$ $(1) f_2 = \sqrt{1 - \sin(x_1)^2} + 2(x_2 - \sin(x_1))^2$ $(2) f_2 = \sqrt{1 - \sin(x_1)^2} + 2(x_2 - 1 - \sin(x_1))^2$ PS: $x_2 = \sin(x_1), x_2 = \sin(x_1) + 1$ PF: $f_2 = \sqrt{1 - f_1}, 0 \leq f_1 \leq 1$

	$f_1 = x_1, -\pi < x_1 < \pi, 0 \leq x_2 \leq 1$
	$f_2 = 1 - x_1 + 2(x_2 - \sin(x_1))^2$
F7	PS: $x_2 = \sin(x_1)$
	PF: $f_2 = 1 - f_1, -\pi \leq f_1 \leq \pi$
	$f_1 = x_1, -\pi < x_1 < \pi, 0 \leq x_2 \leq 1$
	$f_2 = 1 - x_1^3 + 2(x_2 - \sin(x_1))^2$
F8	PS: $x_2 = \sin(x_1)$
	PF: $f_2 = 1 - f_1^3, -\pi \leq f_1 \leq \pi$
	$f_1 = \sum_{i=1}^{n-1} x_i - 2 , 1 \leq x_i \leq 3, i = 1, 2, \dots, n-1, (1) -1 \leq x_n \leq 1, (2) 1 < x_n \leq 3$
	(1) $f_2 = 1 - \sqrt{\sum_{i=1}^{n-1} x_i - 2 } + 2(x_n - \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi))^2$
F9	(2) $f_2 = 1 - \sqrt{\sum_{i=1}^{n-1} x_i - 2 } + 2(x_n - 2 - \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi))^2$
	PS: $x_n = \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi), x_n = \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi) + 2$
	PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
	$f_1 = \sum_{i=1}^{n-1} x_i - 2 , i = 1, 2, \dots, n-1, (1) -1 \leq x_n \leq 0 \text{ if } x_i \in (1 + k/6, 7/6 + k/6],$
	$k = 1, 3, 5, 6, 8, 10, \text{ or } 0 < x_n \leq 1 \text{ if } x_i \in [1 + k/6, 7/6 + k/6], k = 0, 2, 4, 7, 9, 11,$
	(2) otherwise
	(1) $f_2 = 1 - \sqrt{\sum_{i=1}^{n-1} x_i - 2 } + 2(x_n - \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi))^2$
F10	(2) $f_2 = 1 - \sqrt{\sum_{i=1}^{n-1} x_i - 2 } + 2(x_n - 1 - \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi))^2$
	PS: $x_n = \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi), x_n = \sum_{i=1}^{n-1} \sin(6\pi x_i - 2 + \pi) + 1$
	PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
	$f_1 = x_1 - 2 , 1 \leq x_1 \leq 3, -1 \leq x_2 \leq 1$
	$f_2 = 1 - \sqrt{x_1} + (x_2 - (0.3x_1^2 \cos(24\pi x_1) + 0.6x_1) \sin(6\pi x_1 + \pi))^2$
F11	PS: $x_2 = (0.3x_1^2 \cos(24\pi x_1) + 0.6x_1) \sin(6\pi x_1 + \pi)$
	PF: $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$
	$f_1 = \sin(x_1), -\pi < x_1 < \pi, (1) 0 \leq x_2 \leq 4, (2) 4 < x_2 \leq 9$
	(1) $f_2 = \sqrt{1 - \sin(x_1)^2} + 2(x_2 - \sin(x_1) - x_1)^2$
F12	(2) $f_2 = \sqrt{1 - \sin(x_1)^2} + 2(x_2 - 4 - \sin(x_1) - x_1)^2$
	PS: $x_2 = \sin(x_1) + x_1 , x_2 = \sin(x_1) + x_1 + 4$
	PF: $f_2 = \sqrt{1 - f_1^2}, 0 \leq f_1 \leq 1$

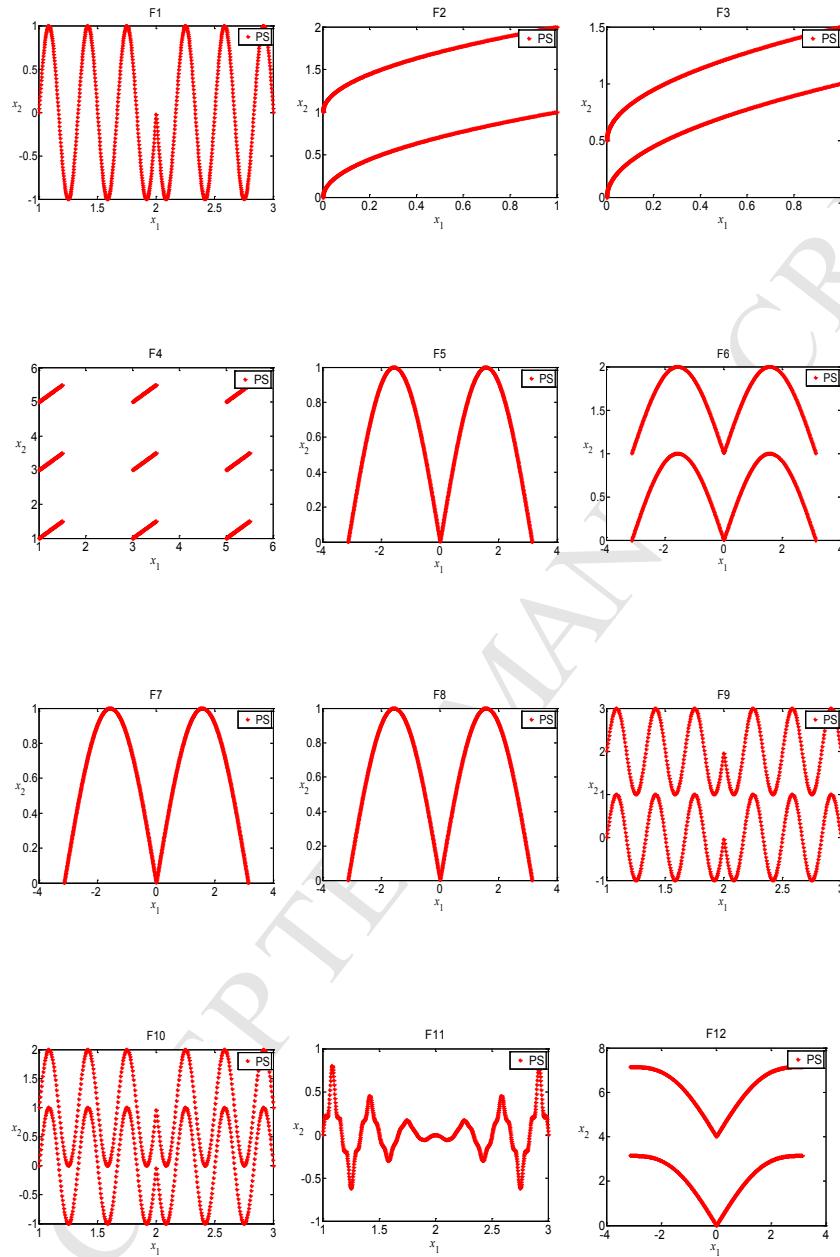


Fig. A1. Images of the decision-space Pareto sets for functions F1 - F12 using the analytical formulas of Table A1.

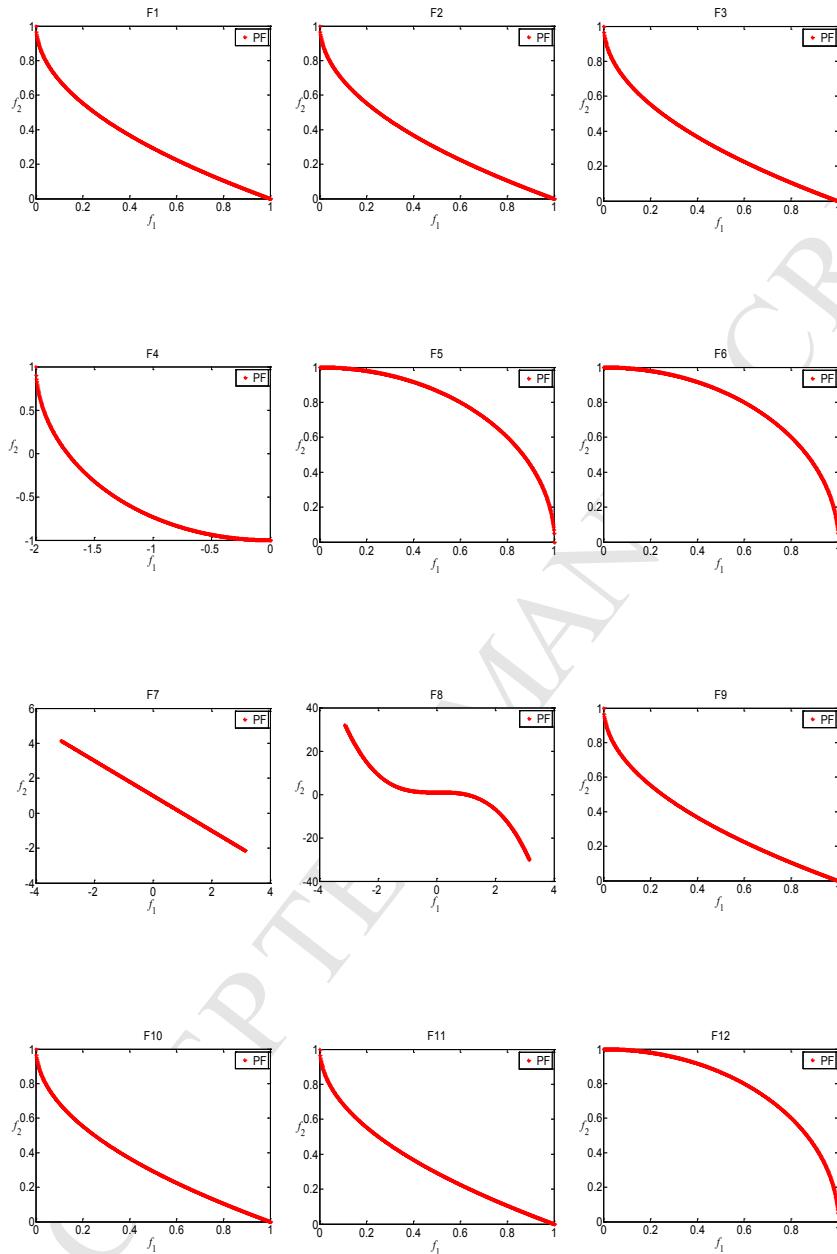
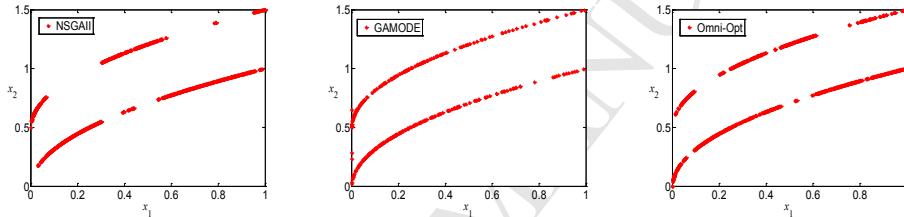
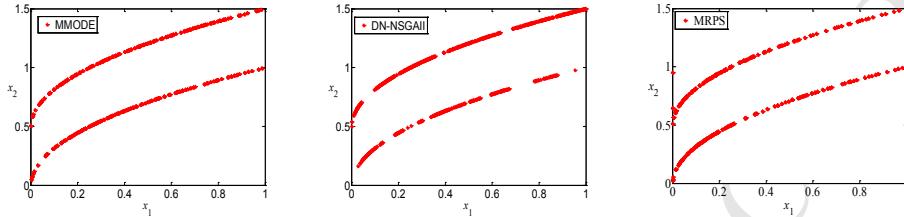
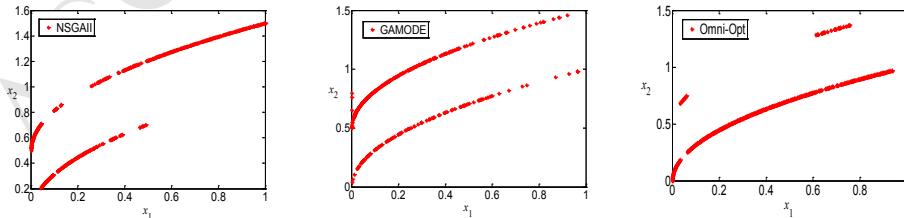
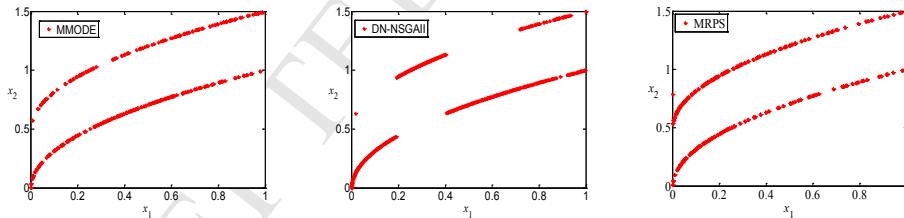


Fig. A2. Images of the objective-space Pareto fronts for functions F1 - F12 using the analytical formulas of Table A1.

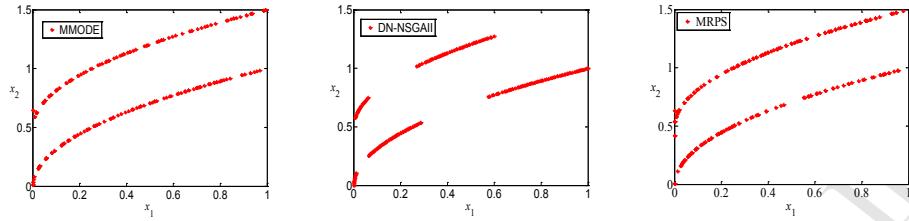
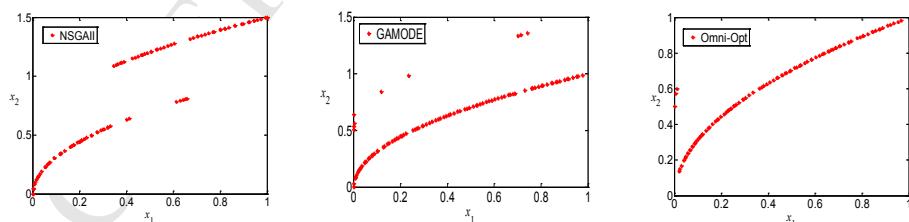
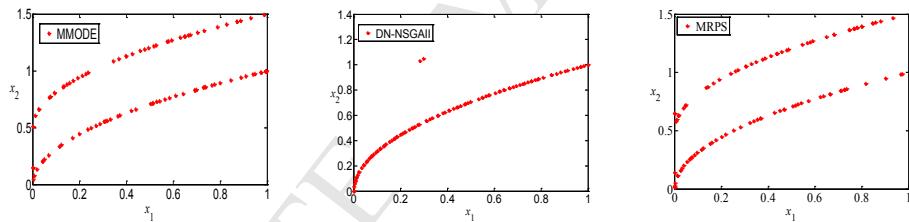
Appendix B: Results for the six algorithms considered under different population sizes



(a) $NP = 800$



(b) $NP = 400$

(c) $NP = 200$ (d) $NP = 100$

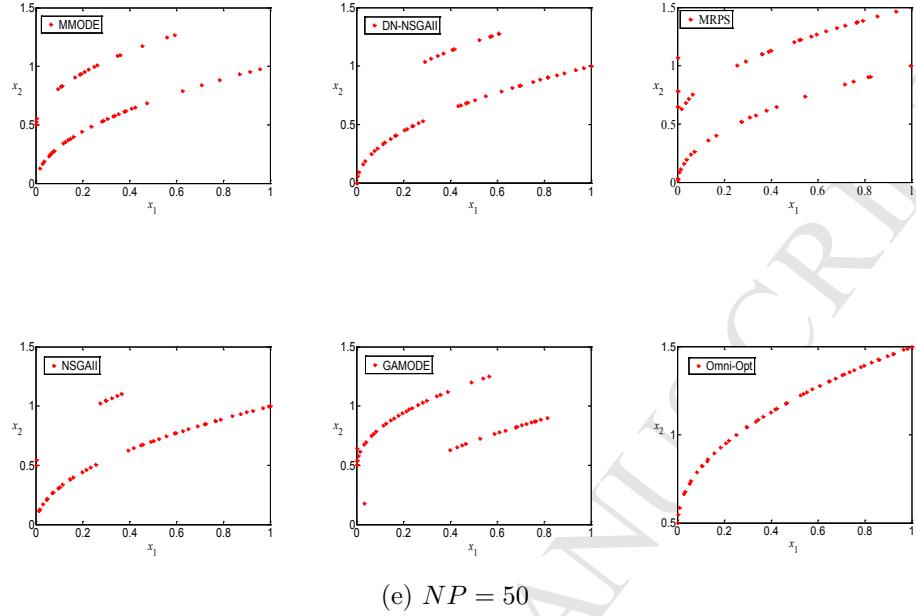
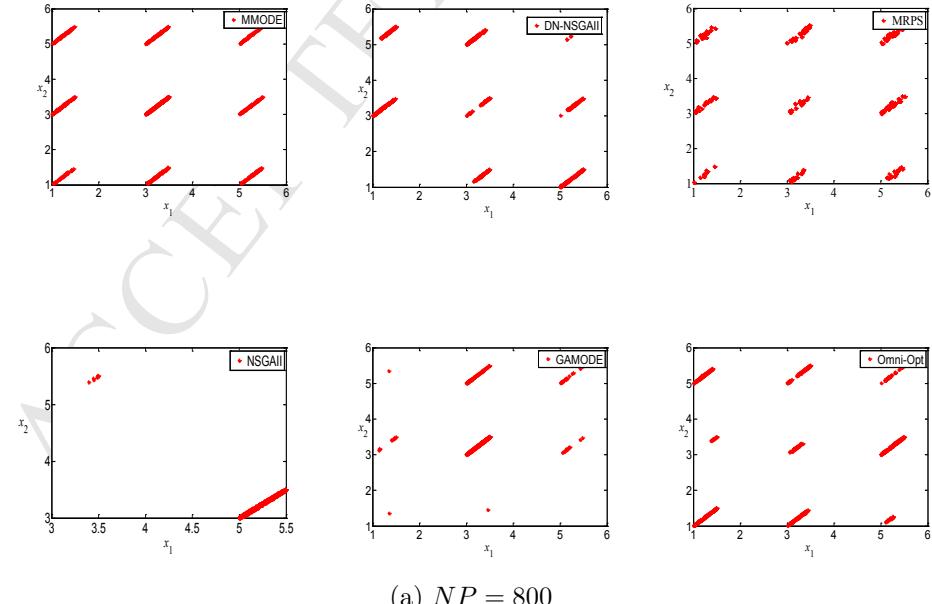
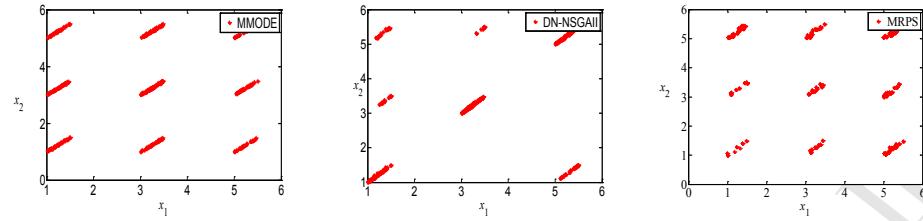
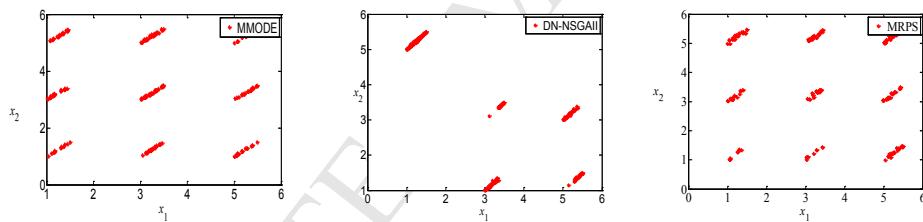
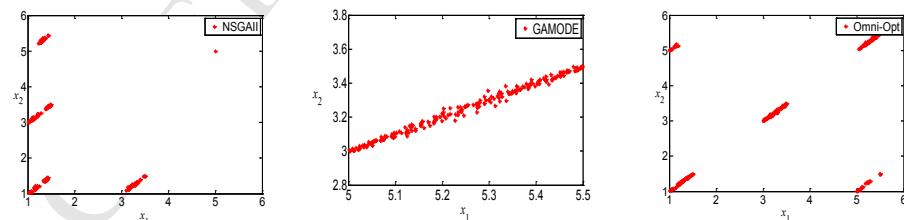


Fig. B1. Search results in the decision space for objective-function F3 for population sizes NP equal to 50, 100, 200, 400 and 800.



(b) $NP = 400$ (c) $NP = 200$ 

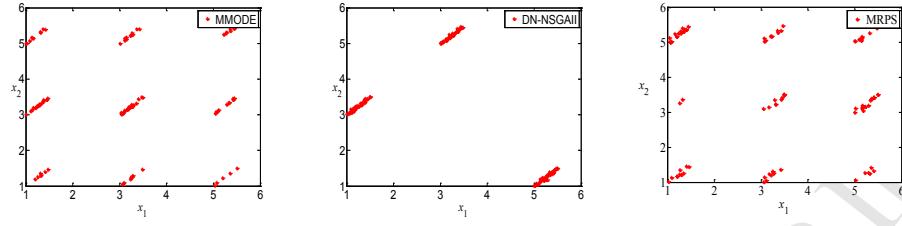
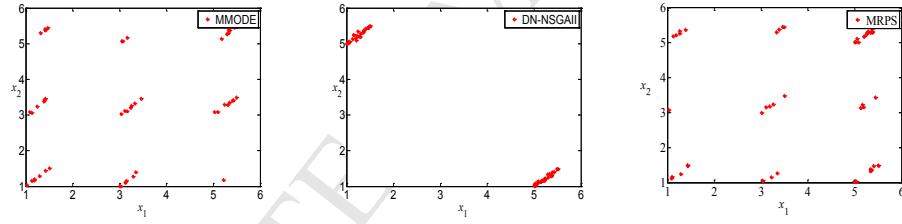
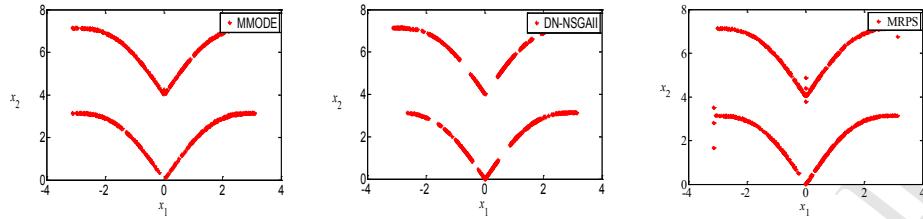
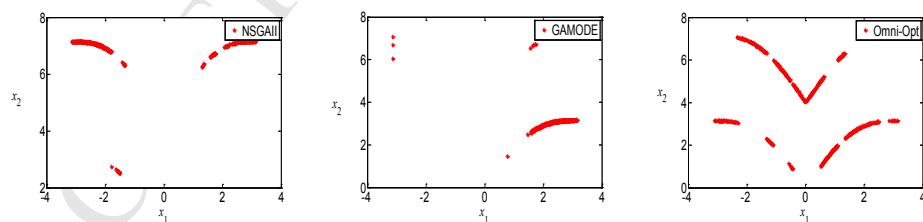
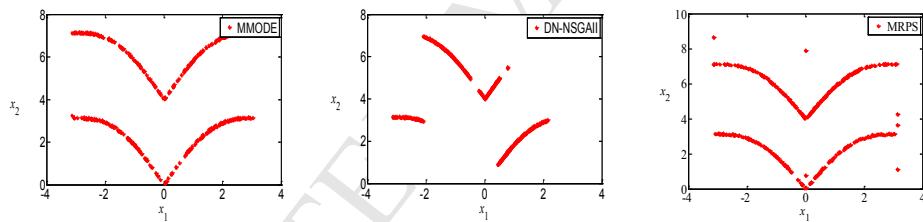
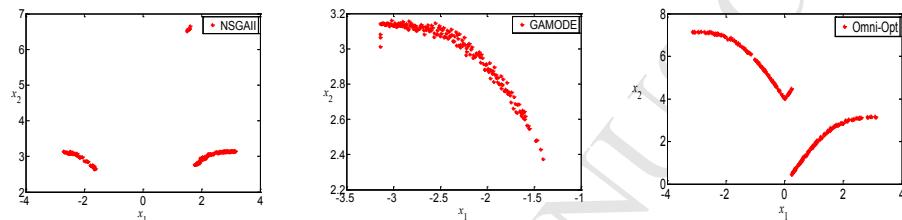
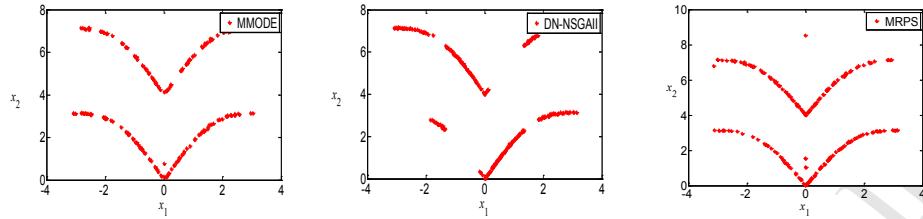
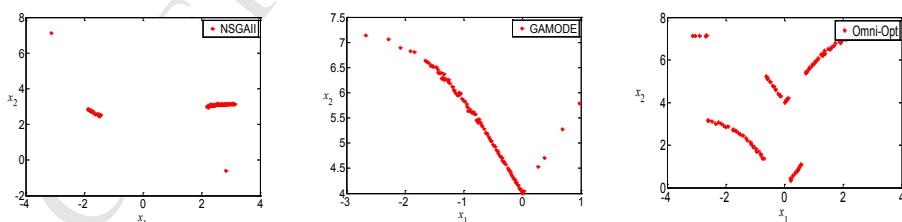
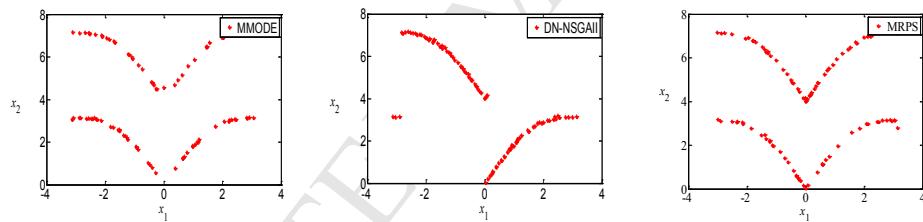
(d) $NP = 100$ (e) $NP = 50$

Fig. B2. Search results in the decision space for objective-function F4 for population sizes NP equal to 50, 100, 200, 400 and 800.

(a) $NP = 800$ (b) $NP = 400$

(c) $NP = 200$ (d) $NP = 100$

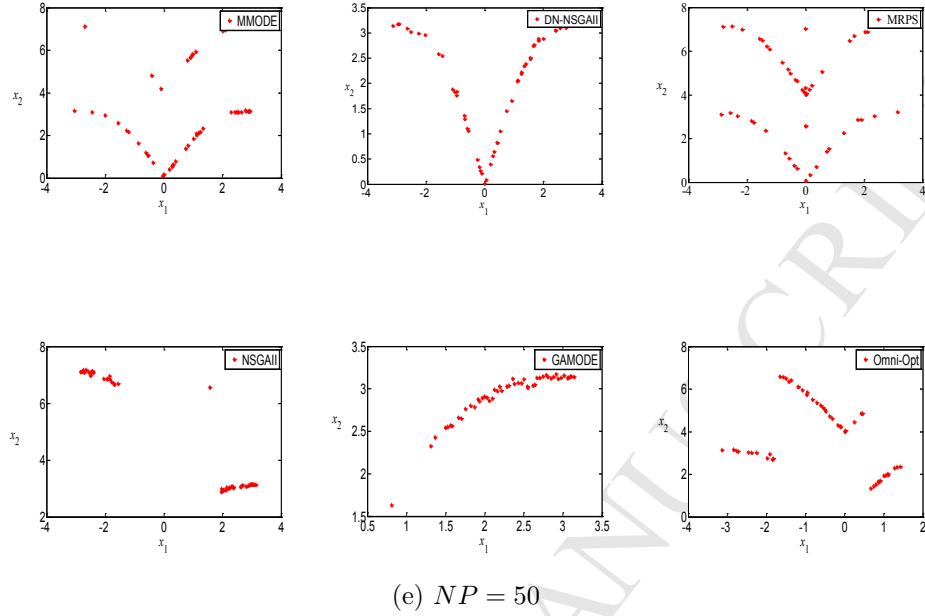
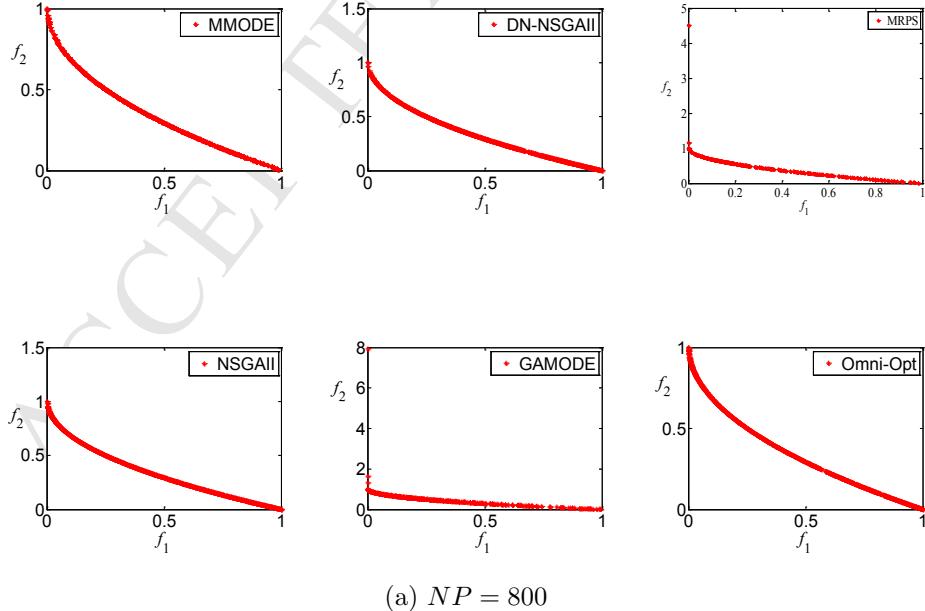
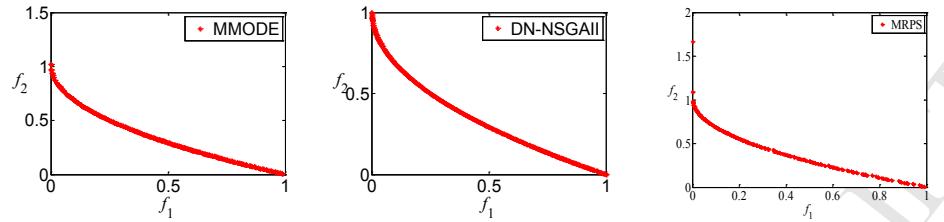
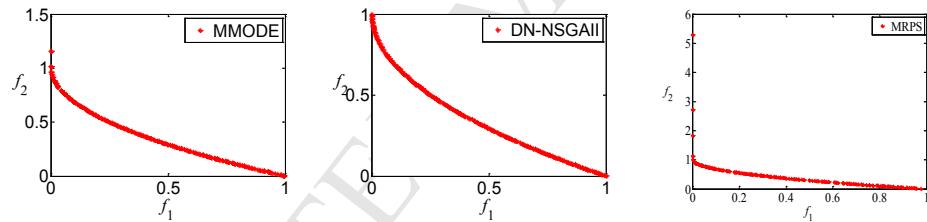
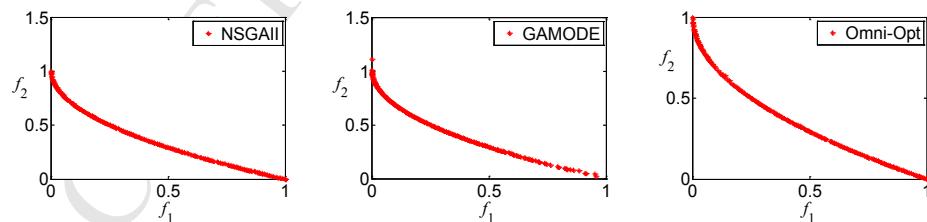


Fig. B3. Search results in the decision space for objective-function F12 for population sizes NP equal to 50, 100, 200, 400 and 800.



(b) $NP = 400$ (c) $NP = 200$ 

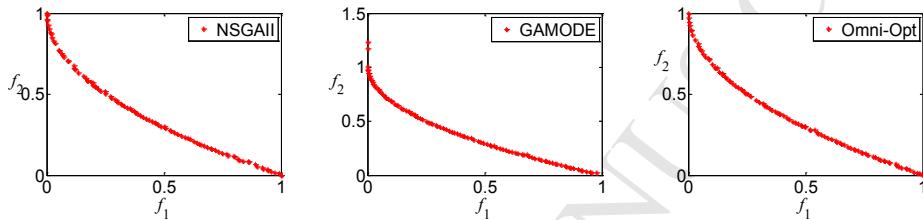
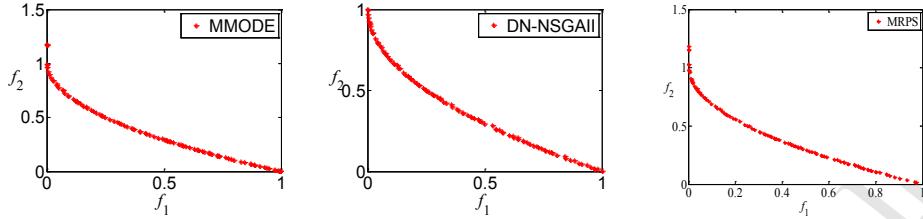
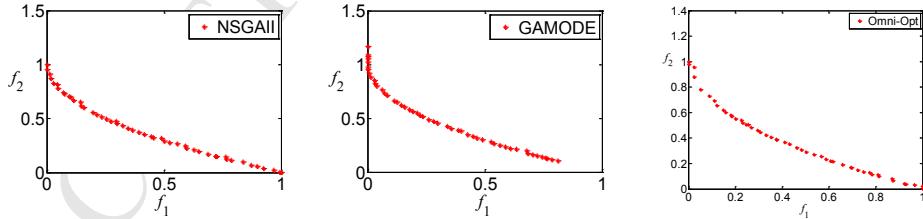
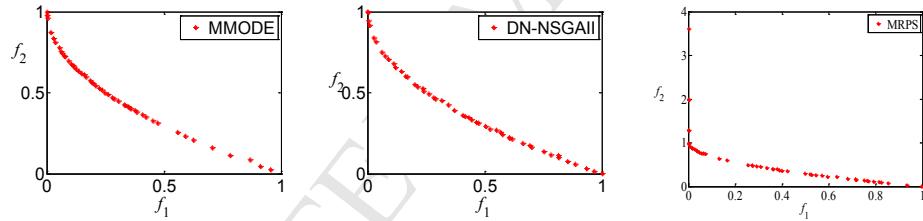
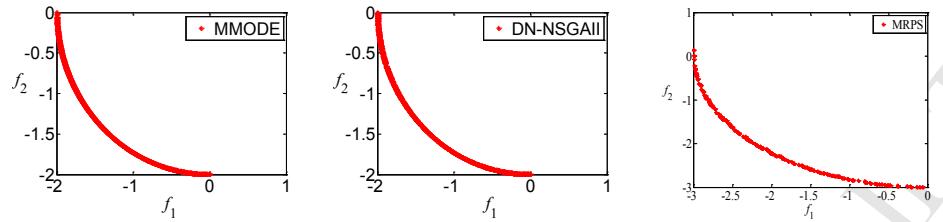
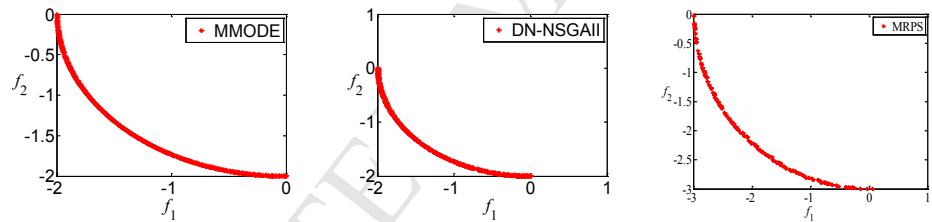
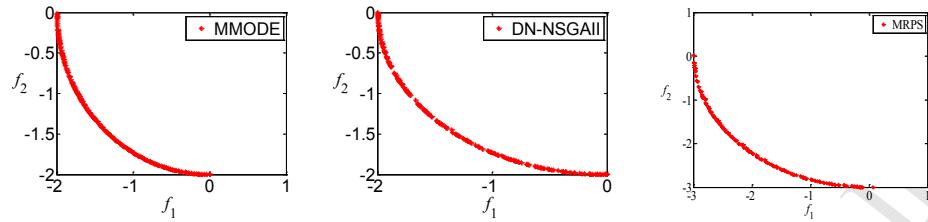
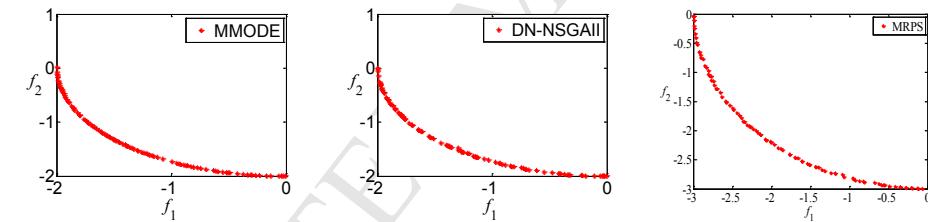
(d) $NP = 100$ (e) $NP = 50$

Fig. B4. Search results in the objective space for objective-function F3 for population sizes NP equal to 50, 100, 200, 400 and 800.

(a) $NP = 800$ (b) $NP = 400$

(c) $NP = 200$ (d) $NP = 100$

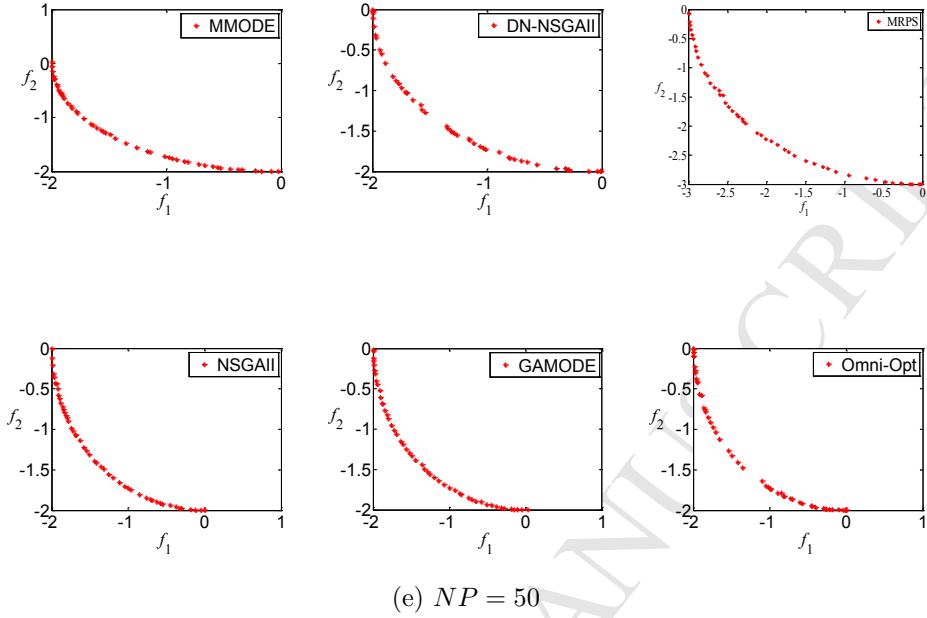
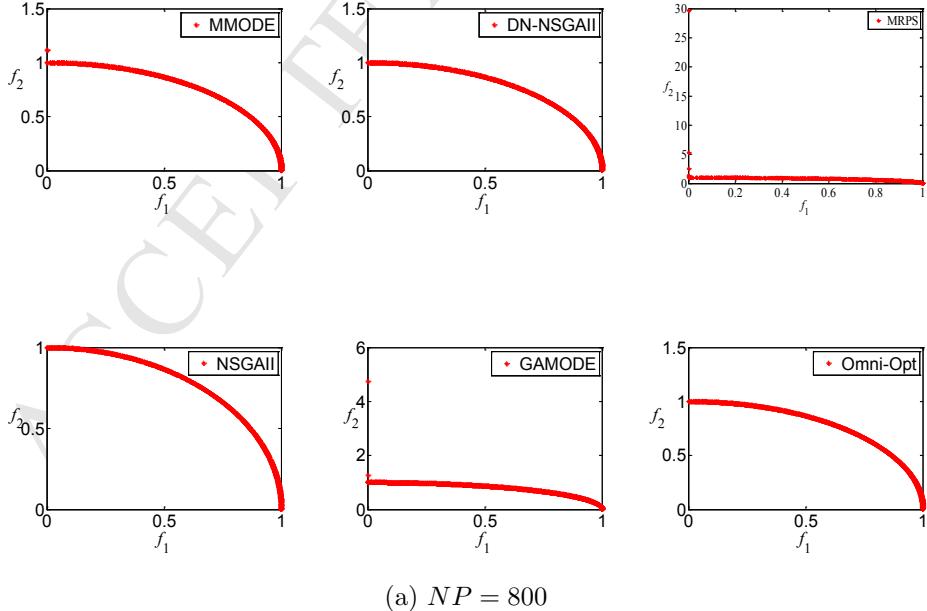
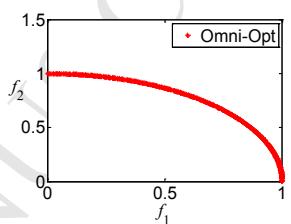
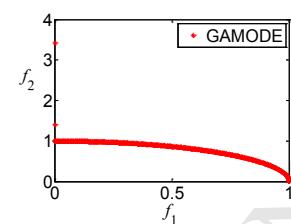
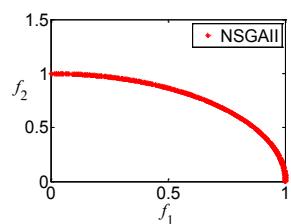
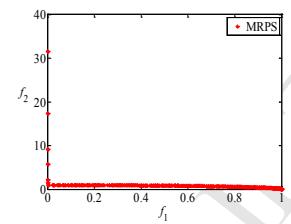
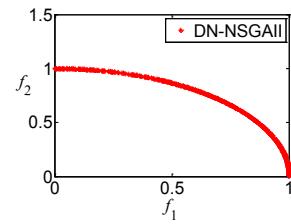
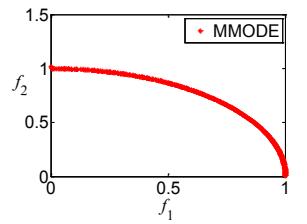
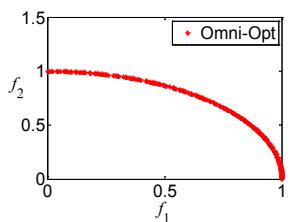
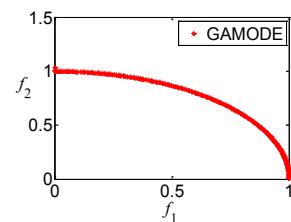
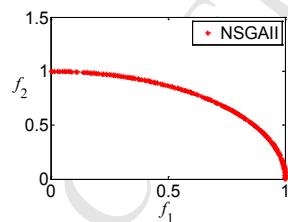
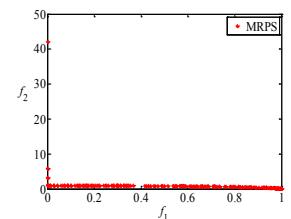
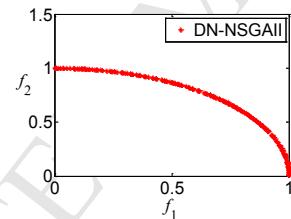
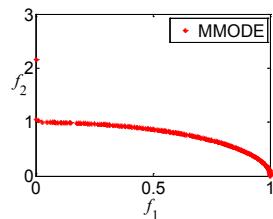


Fig. B5. Search results in the objective space for objective-function F4 for population sizes NP equal to 50, 100, 200, 400 and 800.



(b) $NP = 400$ (c) $NP = 200$

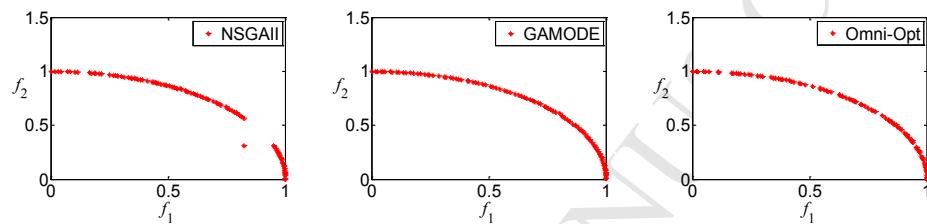
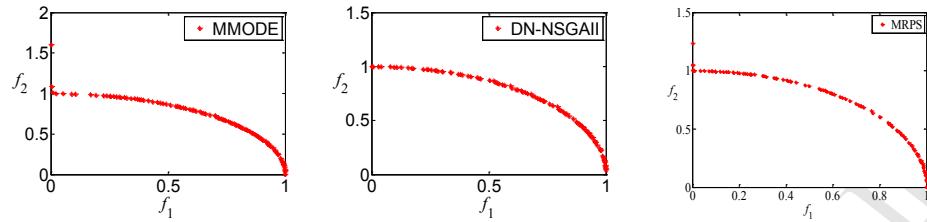
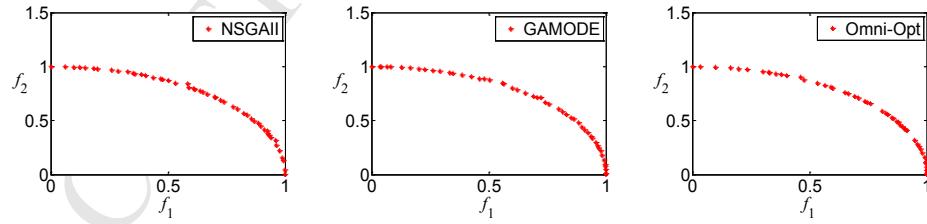
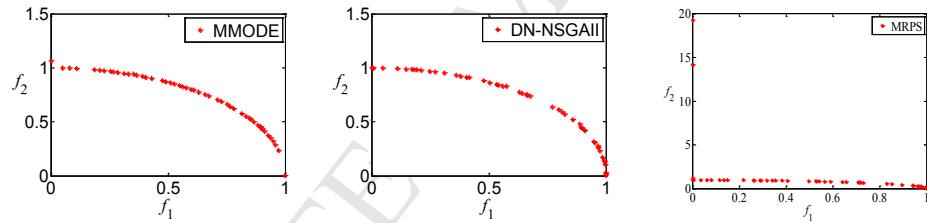
(d) $NP = 100$ (e) $NP = 50$

Fig. B6. Search results in the objective space for objective-function F12 for population sizes NP equal to 50, 100, 200, 400 and 800.

Highlights:

- A multimodal multi-objective optimization with differential evolution method is proposed to solve multimodal multi-objective optimization problems.
- A preselection scheme and a mutation bound processing method are designed to obtain multiple subsets in the Pareto set and improve their distribution.
- Novel test functions with different Pareto set shapes are designed to verify the validity of the proposed algorithm.
- Compared with two existing multimodal multiobjective optimization algorithms and three state-of-the-art multi-objective optimization algorithms, the proposed algorithm shows statistical superiority.