Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 2462891, 14 pages
https://doi.org/10.1155/2017/2462891

*Research Article*

# An Improved Hybrid Algorithm Based on Biogeography/Complex and Metropolis for Many-Objective Optimization

**Chen Wang,[1,2] Yi Wang,[3] Kesheng Wang,[1,4] Yao Dong,[1] and Yang Yang[1]**

[1] *Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai University, Shanghai 200072, China*
[2] *Department of Mechanical Engineering, Hubei University of Automotive Technology, Shiyan 442002, China*
[3] *School of Materials, Manchester University, Sackville St Blg, Manchester M13 9PL, UK*
[4] *Department of Production and Quality Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

Correspondence should be addressed to Chen Wang; 20090011@huat.edu.cn

Received 8 December 2016; Revised 17 January 2017; Accepted 29 January 2017; Published 30 March 2017

Academic Editor: Thomas Hanne

It is extremely important to maintain balance between convergence and diversity for many-objective evolutionary algorithms. Usually, original BBO algorithm can guarantee convergence to the optimal solution given enough generations, and the Biogeography/Complex (BBO/Complex) algorithm uses within-subsystem migration and cross-subsystem migration to preserve the convergence and diversity of the population. However, as the number of objectives increases, the performance of the algorithm decreases significantly. In this paper, a novel method to solve the many-objective optimization is called Hmp/BBO (Hybrid Metropolis Biogeography/Complex Based Optimization). The new decomposition method is adopted and the PBI function is put in place to improve the performance of the solution. On the within-subsystem migration the inferior migrated islands will not be chosen unless they pass the Metropolis criterion. With this restriction, a uniform distribution Pareto set can be obtained. In addition, through the above-mentioned method, algorithm running time is kept effectively. Experimental results on benchmark functions demonstrate the superiority of the proposed algorithm in comparison with five state-of-the-art designs in terms of both solutions to convergence and diversity.

## 1. Introduction

In the scientific research and engineering practice, multiple objectives are usually needed to be optimized simultaneously. Because of the conflict between multiple targets in multiobjective optimization, the performance improvement of one subobjective may cause the performance of another subobjective to decrease. Only through the compromise method can all objectives go as far as possible to attain optimal. The set of all the optimal Pareto optimal solutions is known as the Pareto set (PS) and their corresponding objectives vector in the objective space is the Pareto Front (PF) [1]. The purpose of many multiobjective optimization evolutionary algorithms (MOEAs) is to determine a better approximation of the PF and PS. Although Many MOEAs have effectively solved multiobjective optimization problems (MOPs) with only two or three objectives [2], the MOPs with more than three objectives are too difficult to be solved by most already existing MOEAs [3]. In the recent literature report, the MOPs that have more than three objectives are often described as many-objective optimizations (MaOPs) [4, 5].

Due to the fact that minimization and maximization problem can be mutual transformation, without loss of generality, this article mainly describes minimizing many-objective problem and its related concepts MaOPs which can be defined as follows:

$$\begin{aligned} \text{Minimize} \quad & F(x) = \left(f_1(x), f_2(x) \cdots f_n(x)\right)^T \\ \text{Subject to} \quad & x \in \Omega, \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)^T$ is the $n$-dimensional decision variable vector, $\Omega$ is the feasible search region, and

$F : \Omega \rightarrow R^n$ consists of $n$ objective functions $f_i(x)$ $(i = 1, 2, \ldots, n)$, $n > 3$.

It is generally agreed that, evolutionary algorithms (EAs) are well suited for MOPs, due to their population based strategy for achieving an approximation to the PF. EAs achieve a Pareto approximation set in MOPs via pursing the entire PF and maximizing the diversity of solutions. The fundamentally balanced convergence and diversity between the EAs depend on the selection operator [6]. As far as we know, the popular MOEAs, including Hypervolume Estimation Algorithm for multiobjective optimization (HYPE) [7], nondominated sorting genetic algorithm II (NSGAII) [8], and grid-based evolutionary algorithm (GREA) [9], could effectively deal with two or three objectives' optimization problems. However, there MOEAs are all confronted with great difficulties in many-objective optimization. With the increase of the objective space in size, the first problem is almost all solutions to the original population becoming nondominated with one another, which is mainly caused by the phenomenon called *dominance resistance* [10]. This will result in the selection pressure toward PF severely deteriorating and considerably slowing down the evolutionary process. This is primarily because most of EMO algorithms used Pareto dominance as selection criteria. The other problem is that the conflict between diversity and convergence becomes deteriorated. This is mainly because most of the current diversity operators (like crowding distance) prefer selecting the dominance resistant solutions. The third problem is the computational complexity and efficiency. With the increase of the number of objectives, the complexity of the EMO algorithm is significantly increased, and the efficiency of the algorithm is substantially reduced. To tackle the above problems, a lot of methods have been proposed [11, 12], which can be roughly divided into mainly four categories:

(1) The dimensional reduction–based methods: by analyzing the relationship between objectives or using feature selection techniques, this method reduces the amount of objectives. In order to decrease the difficulty of the original problem, this method attempts to reduce those unimportant objectives. However, the method assumes that the MaOPs have redundant objectives. This assumption may restrict the application of the dimensional reduction–based methods [13].

(2) The relaxed Pareto-dominance-based methods: the relaxed Pareto-dominance-based methods enhance the selection pressure on the Pareto Front, such as the methods of $\alpha$-dominate [14] and $\varepsilon$-dominate [15]. But the main drawback of this method is that it involves one or more parameters and these parameters are difficult to choose.

(3) The indicator-based approaches: the indicator-based method is not subject to the selection pressure problem, since it is not depending on the Pareto-dominance to push the population toward the PF. However, it has also suffered from the *curse of dimensionality* [16]. The current metrics available to Hypervolume is probably the most popular one that is employed for multiobjective search. However, computation cost of the Hypervolume grows exponentially with the number of objectives [17].

(4) The decomposition-based method: this method has developed the decomposition strategy and the neighborhood concept. As one of the popular algorithms, many-objective evolutionary algorithm based on decomposition (MOEA/D) was proposed by Zhang and Li (2007) [18]. The aggregation function is used to compare the solutions and the uniform distribution weight vectors preserving solution convergence and diversity.

MOEA/D has a good convergence and diversity, with low computational complexity, so as to get an effective method. With the BBO/complex algorithm decomposition option, each subsystem has multiobjectives and multiple constraints [19]. It has more flexible decomposition options compared to traditional methods founded on decomposition. BBO/complex detailed explanations are introduced in Section 2. Despite the advance in adapting MOEAs for leading with MaOPs, very few have been recorded in the sense of improving BBO/complex for solving MaOPs so as to balance both convergence and diversity simultaneously. We present a new algorithm, the hybrid Metropolis BBO/Complex (Hmp/BBO) for many-objective optimization; Hmp/BBO, under the basic framework of BBO/Complex algorithm, improves the convergence and diversity in many-objective optimization by introducing the decomposition strategy and PBI aggregation function in MOEA/D.

Furthermore, selection in BBO/complex plays a major role in the information exchange among subsystems. We hope the useful information can be forwarded to the appropriate subsystems to improve them and will not mislead other unsuitable subsystems. In the original version of BBO/complex, the roulette wheel selection is based on the emigration rates to select the emigration islands. On the stage of within-subsystem migration, each SIV in an immigrating island will have a chance to be replaced by SIV from an emigrating island. However, it is not clear whether the new emigration islands are suitable for these subsystems. During the within-subsystem migration phase, some solutions with high quality are easily found at initial search stages, and they will be replaced easily by most current solutions. Consequently various subsystems will be trapped at their local convergence. The simulated annealing (SA) algorithm was proposed by Kirkpatrick et al. [20] and Černý [21]. SA is an intelligent algorithm based on probabilistic stochastic search optimization. It has the capability of potential jumping, and it can accept noninferior solution and inferior solution. Therefore, it effectively avoids falling into the local minimum solution and keeps solutions diversity. We are inspired here by the Metropolis criterion of SA algorithm to solve the problem posed above. Details about SA will be introduced in next section.

On the other hand, when sharing information in the subsystem, because there are numerous objectives and constraints, we need a new method to reduce the computation time of the central processor.

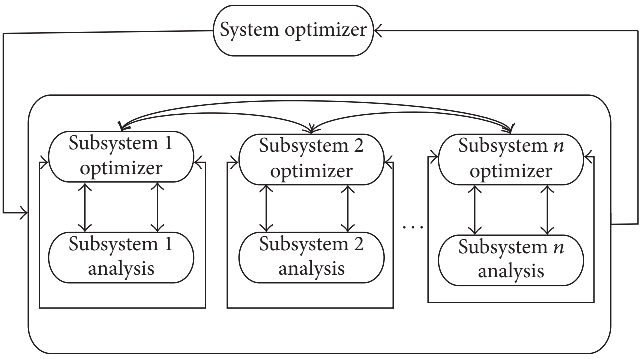


Figure 1: BBO/Complex formulation.

From discussion above, this paper mainly focuses on the Hmp/BBO algorithm that promotes the performance of convergence and diversity in many-objective optimization. Our contributions to this topic are summarized as follows:

(1) We have designed a new framework of Hmp/BBO algorithm for many-objective optimization.

(2) We have introduced PBI aggregation function to improve the convergence and diversity for Hmp/BBO.

(3) We have to adopt the Metropolis criterion to improve the performance of exploration and exploitation.

(4) We have introduced a checking unit; this unit ensured that the new islands are only generated within the Hmp/BBO, so it has saved an additional amount of CPU time.

The remainder of this paper is organized as follows: the principles of BBO/Complex and SA are, respectively, covered in Section 2. The hybrid Metropolis BBO/Complex algorithm (Hmp/BBO) is presented in Section 3. In Section 4, the experimental studies are given to demonstrate the efficiency of the proposed method as well as some discussions on this paper. Finally, Section 5 concludes the paper and future research directions are proposed.

## 2. BBO/Complex and Simulated Annealing

*2.1. BBO/Complex.* The biogeography-based optimization algorithm is an inventive algorithm, introduced for the first time in 2013, and according to [19] it provides competitive optimization performance with NSGAII [8], differential evolution (DE) [22], ant colony optimization (ACO) [23], and a lot of other algorithms. BBO/Complex extends BBO algorithm system of multiple subsystems; each subsystem contains multiobjectives and multiconstraints. The BBO/complex framework comprises $N$ archipelagos, where $N$ equals the number of subsystems. Each archipelago appears to have a lot of islands. These islands represent candidate solutions to the problem. BBO/complex framework is distinct from other MOEA algorithms. It includes framework and optimization algorithm, as showed in Figure 1. It provides an efficient model for communication between subsystems and provides a new way of migrating to share information between within-subsystems and cross-subsystems.

The classical BBO/Complex algorithm proposed can be described with the following algorithm:

(1) Initialize the control parameters: population size, stop condition, and mutation probability. Initialization of the population is initialized by randomly generated individuals.

(2) Get the objective and constraints value similarity levels between all pairs of subsystems.

(3) Obtain the rank of islands in each subsystem.

(4) Perform within-subsystem migration.

(5) Perform cross-subsystem migration.

(6) Mutation on each island.

(7) Replace the worst island in the population with the generation's good islands.

(8) If the termination condition is not met, go to step (3); otherwise, terminate.

*2.2. Simulated Annealing Algorithm.* The algorithm is built on the metaheuristic technique of thermodynamics of material annealing [24–26]. At the beginning of the process the temperature rises and it is gradually cooled down to a minimum. The objective is to minimize the cost function and is expected to reach the lowest cost function for the freezing temperature. When the process is there, the temperature decreases and a new state is created. A simulated annealing algorithm based on statistical mechanics is established. In 1953, [27] adopted the concept of Boltzmann's probability distribution. This means that if a system maintains its thermal equilibrium at temperature $T$, the probability distribution $P$ of its energy $E$ can be calculated by the following [27]:

$$P(E) = \frac{-\Delta E}{e^{K_B T}}, \tag{2}$$

where $K_B$ is Boltzmann's constant. The difference in energy $\Delta E$ means the difference in cost function between the past and current iterations, which can be determined as follows:

$$\Delta E = f(x_n) - f(x_o). \tag{3}$$

For minimization problems, $\Delta E \leqslant 0$ means $f(x_n) \leqslant f(x_o)$, so the new point is directly accepted. Otherwise, the Metropolis criterion will be enabled to decide whether to accept or reject $x_o$. For this case where $\Delta E > 0$, the acceptance is treated probabilistically according to the relation $P = 1/1 + e^{(\Delta E/\max(T))}$. It can be seen that it is affected by the temperature of the receiving process. For the maximum amplitude of $T$, the acceptance probability is also selected to be a much worse state too. This process will avoid falling into local optima. As the temperature decreases, the algorithm accepts only states which minimize the $F_0$ cost. Thus, in the iterative process, the temperature reduction way is one of the key parameters, which is named as the cooling schedule. On the other hand, once the iteration $n$ is complete, the next cycle will begin

```
Output: population $p$
Initialization all the parameters
  generating parent population, weighting vectors and neighborhood index set
  decomposition strategy
while the termination condition is not met do
  calculate the nondominated ranking system in subsystem
    perform within-subsystem migration
    perform cross-subsystem migration
    perform mutation
    clear the duplicate solution
    replace the worst solutions with good islands
end while
return $p$
```

Algorithm 1: Hmp/BBO pseudocode.

with a smaller value. Therefore, the number of cycles in $L$ and the number of iterations $N$ per each cycle are crucial settings. Large $N$ and $L$ lead to better performance but with a longer process time, and vice versa, for small $N$ and $L$, based on the need to choose a compromise between the quality of the solution and the processing speed. The Metropolis criterion can help the algorithm balance the performance of exploration and exploitation.

## 3. The Hybrid Metropolis BBO/Complex Algorithm

*3.1. Framework of Proposed Algorithm.* Hmp/BBO uses the original BBO/complex framework but extends it to multiple subsystems environments to accommodate many-objective optimization problems. First, many-objective problems are decomposed into multiple subsystems. Generally speaking, there are two decomposition methods: one is built on the system requirements and the other is based on the physical system; also the number of subsystems is set by the user according to the decomposition strategy. The original BBO/complex decomposes the problem based on system requirements. Because the objective space dimension is higher, we need to use the new decomposition method. Afterwards, like Figure 2, a PBI aggregate function decomposition method can enhance the convergence and diversity of the algorithm when solving many-objective problems. With the above step, Hmp/BBO migration is divided into two categories: within-subsystem and cross-subsystem. During the within-subsystem migration phase, we enhance the probability of the solution and employ the Metropolis criterion to make the algorithm jump out of local optimal. During the cross-subsystem migration stage, we adopt the roulette wheel method to get best solution. Then we perform mutation and clear the duplicate algorithm.

Our algorithm does not take into account constraints problem. Constraints processing research questions will be in the rest of the work. Algorithm 1 presents the general framework of Hmp/BBO.

*3.2. Generating Weighting Vectors.* We set weight vectors such that the optimal solutions of their subsystems are uniformly distributed along the PF. Most of the approaches for generating weight vectors for different decomposition strategies in MODE/D have been suggested [28]; we use Das's method in [29], with a predefined integer $H$ which controls the divisions along each axis. The total number of such vectors is $N = C_{H+m-1}^{m-1}$. Distributed reference points are $\lambda_k = x_k/H,\ k = 1, 2, \ldots, m$. Taking the three-dimension problem as an example, there are $C_5^2 = 10$ reference points.

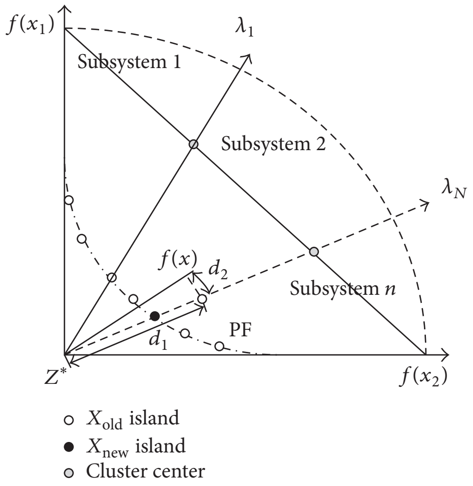


Figure 2: Illustration of decomposition strategy and PBI distance.

*3.3. Decomposition Strategy.* The Hmp/BBO initialization generates $N_w$ weight vectors; PBI aggregate function [30] is employed to divide the weight vectors into a set of $L_s$ $(CO_1, CO_2, \ldots, CO_L)$. Because the weight vectors uniformly distribute in the objective space, the number of weight vectors in one subsystem is approximate to Nw/L. Therefore, the whole objective space is divided into $L_s$ subsystems.

*3.4. The Nondominated Ranking System (NDRS).* NDRS was introduced in [31] as the ranking system in the multiobjective genetic algorithms (MOGA). It uses inconsecutive integers as ranks to reflect the relative performance of each individual in a population. Assume that we have a subsystem: $R_i$ is the rank

```
(1) Probabilistically choose the immigrating islands based on the NDRS.
(2)  Use the Metropolis criterion based on emigration rates to select the emigrating island.
(3) each SIV in an immigration island will have a chance to be replaced by an SIV from an emigrating island.
```

Algorithm 2: Within-migration pseudocode.

```
(1) Randomly select $k$ indices from $E(i)$ (choose similar subsystem) and select $k$ immigrating islands based on immigration rages from population $p$.
(2) Calculate Euclidean distance between the islands of neighboring subsystem
(3) Use the roulette wheel selection based on emigration rates to select the emigrating islands.
(4) each SIV in an immigration island will have a chance to be replaced by an SIV from an emigrating island.
```

Algorithm 3: Cross-migration pseudocode.

of the $i$th island, where a lower rank is better. If objective $o_1$ of island $i_1$ is better than $o_1$ of $i_2$, $R_{i2} = R_{i2} + 1$; if objective $o_1$ of island $i_2$ is better than $o_1$ of $i_1$, $R_{i1} = R_{i1} + 1$.

*3.5. PBI Aggregate Function.* In MOEA/D, several methods have been proposed for decomposing MOPs into a single-objective optimization subproblem, such as weighted sum method, Tchebycheff method, and Penalty-based Boundary Intersection (PBI) method, but it was shown that the MOEA/D-PBI is only best for many-objective optimization problem in [32]. This paper uses the PBI approach. A scalar optimization function is defined as

$$\begin{aligned} \text{Min} \quad & g^{\text{pbi}} x\lambda, \ Z^* = d_1 + \theta d_2 \\ & d_1 = \frac{z^* - Fx^T\lambda}{|\lambda|} \\ & d_2 = \left|Fx - (z^* - d_1\lambda)\right|. \end{aligned} \tag{4}$$

In Figure 2, $Z^*$ is the ideal point, $d_1$ is the Euclidean distance between origin point and the foot point drawn from the solution to the reference direction, and $d_2$ is the perpendicular distance of the solution from the reference direction. The $\theta$ value on the performance of Hmp/BBO is present in Section 4.3.

*3.6. Within-Subsystem Migration.* Within-subsystem migration is mainly for information sharing; we need a new rapid method for selection of immigration islands and emigration islands. First, immigrating islands are selected based on the NDRS. Afterward, emigrating islands are selected by emigration rates, in order to further improve the algorithm's convergence speed, while avoiding falling into local optima. Features ($n$ SIVs) of the islands will not be directly substituted with the new values that come from the probabilistically selected original islands mentioned above. Instead, $n$ SIVs of the islands are maintained in two temporary matrices with size $t_1 \times t_2$. Each row of the matrices represents one individual. The old independent variables are used again if and only if the modified individual exhibits lower solution quality and does not comply with the Metropolis criterion. The inferior migrated island will be selected only if they pass the Metropolis criterion of SA. With this restriction on the with-subsystem migration, the Hmp/BBO algorithm can avoid falling into the local optimum. The algorithm is described in Algorithm 2.

*3.7. Cross-Subsystem Migration.* During cross-system migration of the stage, it is desirable that the migration should be chosen from a neighborhood subsystem as much as possible. In Hmp/BBO, each island is uniquely specified by a weight vector; each weight vector has been assigned based on the Euclidean distance from a neighborhood. So we can select islands from its neighboring subsystems by neighboring index. First, we select islands between subsystems based neighboring index set $E(i)$. Second, emigrating islands are selected based on the PBI distance. Furthermore, we need to eliminate those poor candidate islands to improve the algorithm diversity between subsystems. The inferior migrated island will not be selected unless it passes the roulette wheel method. With this restriction on the cross-subsystem migration, the Hmp/BBO algorithm's diversity performance can be enhanced. The algorithm is described in Algorithm 3.

*3.8. Mutation Algorithm.* In the Hmp/BBO, these events are modeled as SIV mutation. The mutation rate $m_i$ can be determined by the following number of species involved in the probability $p_i$ to the following equation:

$$m_i = m_{\max}\left(1 - \frac{p_i}{p_{\max}}\right), \tag{5}$$

where the $m_{\max}$ and $p_{\max} = \max(p_i)$ are a user-predefined maximum mutation rate that $m_i$ can reach. The ISI is the variable and function of SIV. The mutation processes are described in Algorithm 4.

*3.9. Clear the Duplicate Algorithm.* Clear duplication will increase the diversity of solutions and avoid duplication of features on other islands. The algorithm can be described in Algorithm 5.

```
for $i \leftarrow 1$ to $k$ do
    {where L = number of islands}
  Calculate probability $P_i$ based on $\lambda_i$ and $\mu_i$
    {by iterative method}
  Calculate mutation rate $m_i$
  if rand < $m_i$ and $i \geq R_m$ then {$R_m$ is a predefined mutation range}
    Replace $n$ SIVs vector of ISI$_i$ with a randomly generated $n$ SIVs vector
  end if
end for
```

Algorithm 4: Mutation pseudocode.

```
Require: Check all $n$ SIVs on all ISI$_i$
while there is a duplicated SIV do
  for $i \leftarrow 1$ to $l$ do {where $l$ = number of islands}
  if any duplicated SIVs is detected then
    Replace the duplicated SIVs in ISI$_i$ with a randomly generated SIVs
  end if
```

Algorithm 5: Clear duplication pseudocode.

In conclusion, The Hmp/BBO would prove the performance of convenience and diversity by introducing a new framework for many-objective optimization. In addition, the SA algorithm comes with only one individual, while the Hmp/BBO is a population based algorithm. Instead of running SA with only one island, it can be executed with all islands in subsystems. Thus, internal searching loops $L$ of SA with each cycle of $n$ can be invalid in Hmp/BBO without affecting the solution quality. This method will save one significant amount of CPU time. By these methods, the exploration and exploitation of the Hmp/BBO algorithm are much improved. The Hmp/BBO framework is described in Algorithm 6.

*3.10. Computational Complexity of One Generation of Hmp/BBO.* In this section, we discuss the computational complexity of the Hmp/BBO. For a population size $N$ and a problem of $M$ objectives, the major computational costs are in steps (5), (6), and (7) of Algorithm 1. Step (5) is the computational complexity of NDRS; it requires $O(NM)$ computations. Computing of within-subsystem (step (6)) needs $O(LN_1{}^2)$ computations, where $L$ is cycles of Metropolis and $N_1$ is generated by parents' populations. Cross-subsystem migration (step (7)) needs $O(LN_1{}^2R)$ computations and $R$ is equal to the number of subsystems. So for $M$ objectives, reviewing all the above computations, the overall complexity of one generation of Hmp/BBO is $O(NM{+}LN_1{}^2{+}LN_1{}^2R)$.

## 4. Simulation Results

In this section, to prove the validity of Hmp/BBO, we compare it with five state-of-the-art algorithms for comparisons, including BBO/Complex, NSGAIII, MOEA/D-PBI, HYPE, and GREA. NSGAIII is based on the Pareto dominance relationship, but the maintenance of population diversity is supported by the use of a group of uniformly distributed reference points. MOEA/D-PBI is a representative of the decomposition-based approach and keeps the diversity of solutions using a series of predefined weight vectors. HYPE is a Hypervolume-based evolutionary algorithm for many-objective optimizations. GREA uses the NSGA-II framework and introduces two concepts (grid dominance and grid difference) and three grid-based standards, and a finesse adjustment strategy. We describe the test questions used in Section 4.1 and the quality indicators in Section 4.2. Then we introduce five state-of-the-art algorithms that used comparison and the corresponding parameter settings in Section 4.3. Finally, the discussion results are in Section 4.4.

*4.1. Test Problems.* In checking to verify the proposed algorithm, the well-known test functions definitions of DTZ1–DTZ4 [33] functions and all of WFG test suite [34] are listed in Table 1. We only consider DTLZ 1–4 problems for DTLZ test suite, because the nature of DTLZ5 and DTLZ6's PFs is unclear beyond three objectives [35]. DTLZ1 is a liner and multimodal function; DTLZ2 is a concave function; DTLZ3 is concave and multimodal function; DTLZ4 is concave and biased function, as summarized in Table 1.

*4.2. Quality Indicators.* In our empirical study, the following three extensively used quality indicators are examined. The first one can reflect the convergence of an algorithm. In the second one and last one, the convergence and diversity of the solutions can be recorded simultaneously.

*(1) Generational Distance (GD).* Let $P$ be the last series of nondominated points in the objective space and $P^*$ be a set

```
Initialization with all the parameters. Initialization $N_w$ weight vectors, $C_L$ are the center of the weight vectors cluster. $L_s$ are the
    number of subsystems and neighborhood set of weight vectors $E(i)$.
Decomposition strategy
for $g \leftarrow 1$ to $G$ do {$G$ = number of generations}
Calculate the rank of islands in each subsystem (NDRS)
Probabilistically select the immigrating islands based on the islands rank, probabilistically select the emigrating islands based
    on the emigration rates.
 if $g = 1$ then
      find the initial temperature $T_1$
     else
         Update the temperature for the $g$th generation
 end if
Save the vectors of all the populations (before migration) in matrix $\mathrm{mm}_1$ with size $t_1 * t_2$ and their cost functions vector vv1
    with length $k$.
Do within-subsystem migration.
Save the vectors of all the populations (after migration) in matrix $\mathrm{mm}_2$ with size $t_1 * t_2$ and their cost functions in vector vv2
    with length $k$.
for $i \leftarrow 1$ to $k$ do ($k$ is number of islands)
Calculate $\Delta E = V_2(i) - V_1(i)$
if $\Delta E > 0$ then
   $P(E) = \frac{-\Delta E}{e^{T(g)K_B}}$
  if $P(E) >$ rand then
     Accept $\mathrm{mm}_2(I{,}1{\rightarrow}n)$ vector of matrix $\mathrm{mm}_2$ as an updated population for ISI
  else
     Re-select the past $\mathrm{mm}_1(I{,}1{\rightarrow}n)$ vector of matrix $\mathrm{mm}_1$ as an updated population for $\mathrm{ISI}_i$
  end if
end if
Randomly select $k$ indices from $E(i)$ and select $k$ immigrating islands based immigration rates.
if$g^{\mathrm{pbi}}(x_{\mathrm{new}} \mid w, z^*) \leqslant g^{\mathrm{pbi}}(x \mid w, z^*)$
save the best islands $x_{\mathrm{new}}$ for emigrating
else
Save the vectors of all the populations (before mutation) in matrix $\mathrm{mm}_3$ with size $t_3 * t_4$ and their cost functions in vector $\mathrm{vv}_3$
    with length $k$
Do cross-subsystem migration
use the roulette wheel method to select population
    with length $k$.
Do mutation
  Do clear duplicated SIV
  if $g > 1$ then
     Repalce the worst ISI with the good ISI saved in the elitism stage
   end if
   end for
   display the best islands
  end if
 end if
end if
Do mutation
Do clear duplicated SIV
 if $g > 1$ then
Replace the worst ISI with the good ISI saved in the elitism stage
 end if
end for
Display the best islands.
```

Algorithm 6: Hybrid Metropolis Biogeography/Complex Based Optimization.

Table 1: Test functions utilized in this experiment.

| Name of function | Characteristics |
|---|---|
| DTLZ1 | Liner, multimodal |
| DTLZ2 | Concave |
| DTLZ3 | Concave, multimodal |
| DTLZ4 | Concave, biased |
| WFG1 | Mixed, Biased, scaled |
| WFG2 | Convex, disconnected |
| WFG3 | Linear, degenerate |
| WFG4 | Concave, multimodal |
| WFG5 | Concave, deceptive |
| WFG6 | Concave, scaled |
| WFG7 | Biased, scaled |
| WFG8 | Biased, nonseparable |
| WFG9 | Concave, scaled |

of points evenly distributed over the actual PF. GD can only reflect the performance of convergence for an algorithm; a smaller value means better quality. Then GD is described as follows:

$$\mathrm{GD}\left(P, P^{*}\right) = \frac{\sqrt{\sum_{u\in P} d\left(u, P^{*}\right)}}{|P|}. \quad (6)$$

*(2) Inverted Generational Distance (IGD).* Let $P^{*}$ be a set of points uniformly sampled over the actual efficient front (EF), and let $P$ be the set of solutions obtained by an algorithm. IGD may be used to measure convergence and diversity in a sense, and a smaller value means better quality. Then IGD is described as follows:

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

*(3) Hypervolume (HV) Indicator.* HV indicator can measure both convergence and diversity of a solution set. The HV is used for WFG test suite, and the larger the HV value is, the better the solution's quality for the PF is. Then HV is described as follows:

$$\mathrm{HV}\left(A, r\right) = \mathrm{Volume}\left(\bigcup_{f\in A}\left[f_1, r_1\right]\times\cdots\times\left[f_m, r_m\right]\right). \quad (8)$$

$M$ is the number of objectives. $A$ is the set of nondominated points obtained in the objective space by an algorithm. $r = (r_1, r_2, \ldots, r_m)T$ is a reference point in the objective space which is dominated by all Pareto points.

*4.3. Parameters Setting.* The parameters for the six MOEAs considered in this study are listed below.

(1) Population size: the population size $N$ used in this study for Hmp/BBO is 91, 206, and 210 for three-, four-, and five-objective problems, respectively. Furthermore, the size of the NSGA-III population was mildly adjusted as in the original NSGAIII study, that is, 92, 212, and 276 for three-, four-, and five-objective problems.

Table 2: Maximum number of fitness evaluation for different function.

| Function | $M = 3$ | $M = 4$ | $M = 5$ |
|---|---|---|---|
| DTLZ1 | 36500 | 26400 | 12600 |
| DTLZ2 | 22840 | 42500 | 73600 |
| DTLZ3 | 92000 | 14600 | 220000 |
| DTLZ4 | 54700 | 13200 | 220000 |
| WFG1–9 | 36500 | 74300 | 157500 |

Table 3: Settings of grid division div in GREA.

| Function | $M = 3$ | $M = 4$ | $M = 5$ |
|---|---|---|---|
| DTLZ1 | 10 | 10 | 10 |
| DTLZ2 | 10 | 10 | 9 |
| DTLZ3 | 11 | 10 | 11 |
| DTLZ4 | 10 | 10 | 7 |
| WFG1 | 7 | 9 | 14 |
| WFG2 | 18 | 17 | 16 |
| WFG3 | 14 | 14 | 14 |
| WFG4–9 | 10 | 10 | 11 |

(2) Stop condition: each algorithm runs 20 times independently on each test question. All algorithms are implemented on a 2.6 GHz CPU desktop computer, 8 GB RAM, and Windows 10. The stop condition of the algorithm is the maximum number of fitness evaluations, as outlined in Table 2.

(3) Parameter setting in Hmp/BBO: set immigration rate is $\lambda_{\max} = 1$, emigration rate is $\mu_{\max} = 1$, max $= 0.01$, $t_0 = 1000$, $K = 0.9$, and $T_{\mathrm{end}} = 0.001$, and penalty parameter in PBI: $\theta_{\mathrm{Hmp}} = 5$, is also suggested in [18]. Neighborhood sizes $M_{\mathrm{Hmp}} = 20$.

(4) Parameter setting in BBO/Complex: set immigration rate is $\lambda_{\max} = 1$, emigration rate is $\mu_{\max} = 1$, mutation probability $p = 0.05$, and the generation count $g = 30$.

(5) Reproduction operator setting: crossover probability $p_c = 1, \eta_c = 30$ in NSGAIII. The mutation probability is $p_m = 1/V$ and its distribution index $\eta_m = 30$.

(6) Parameter setting in MOEA/D-PBI: neighborhood size $M_{\mathrm{MOEA/D}} = 20$; probability used for selection in neighborhood $\delta = 0.9$. Penalty parameter $\theta_{\mathrm{MOEA/D}} = 5$.

(7) The grid division (div) in GREA: Div is set according to [9] as shown in Table 3.

(8) Number of points in Monte Carlo sampling: it is placed at 1,000,000 to ensure accuracy.

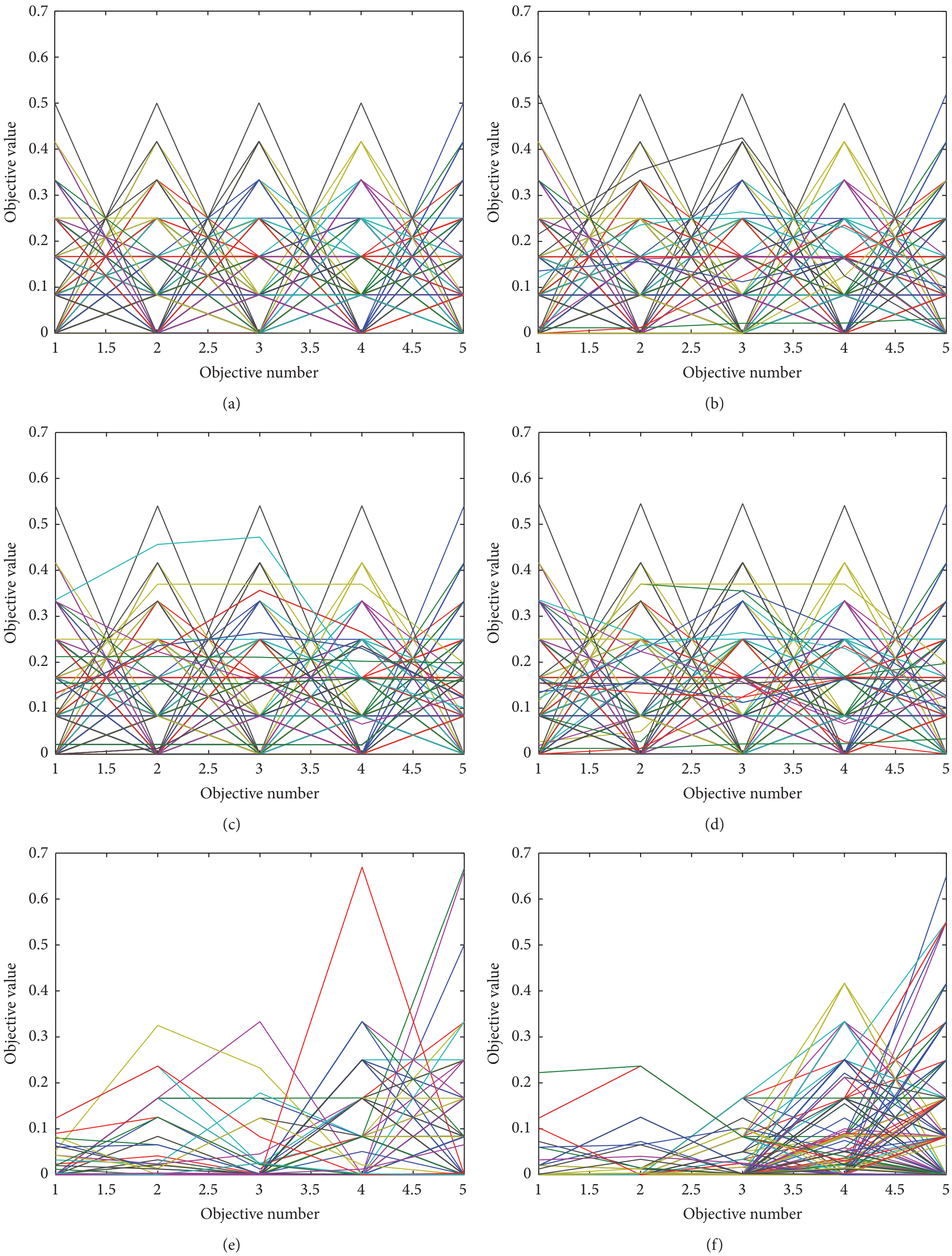
*4.4. Result and Discussion.* In this section, the GD metrics are used to compare the convergence capacity between the proposed Hmp/BBO and other five algorithms. Table 4 shows the best, median, and worst GD values obtained by six algorithms on DTLZ1 to DTLZ4 with different number of

Table 4: Best, median, and worst GD values obtained by DTLZ1 to DTLZ4 with six algorithms by different number of objectives.

| Functions | $m$ | Values | Hmp/BBO | BBO/Complex | NSGAIII | MOEA/D-PBI | HYPE | GREA |
|---|---|---|---|---|---|---|---|---|
| DTLZ1 | 3 | Best | **6.078e − 05** | 1.380e − 04 | 3.324e − 04 | 1.478e − 04 | 2.073e − 04 | 2.544e − 04 |
| | | Median | **1.032e − 04** | 1.650e − 04 | 6.749e − 04 | 1.652e − 04 | 5.546e − 04 | 3.132e − 04 |
| | | Worst | **2.218e − 04** | 5.833e − 03 | 6.129e − 03 | 7.634e − 03 | 7.333e − 03 | 8.326e − 03 |
| | 4 | Best | **1.733e − 05** | 2.478e − 05 | 2.189e − 05 | 3.452e − 05 | 3.321e − 05 | 2.235e − 05 |
| | | Median | **2.348e − 05** | 2.742e − 05 | 2.689e − 05 | 3.874e − 05 | 3.981e − 05 | 4.035e − 05 |
| | | Worst | **2.426e − 05** | 2.859e − 05 | 3.056e − 05 | 7.723e − 05 | 5.885e − 05 | 5.126e − 05 |
| | 5 | Best | 1.740e − 04 | 1.857e − 04 | **1.032e − 04** | 2.023e − 04 | 2.253e − 04 | 3.224e − 04 |
| | | Median | 2.304e − 04 | 2.006e − 04 | **1.112e − 04** | 2.635e − 04 | 3.867e − 04 | 4.325e − 04 |
| | | Worst | 2.941e − 04 | 2.506e − 04 | **1.688e − 04** | 3.227e − 04 | 4.954e − 04 | 5.532e − 04 |
| DTLZ2 | 3 | Best | **7.130e − 05** | 1.231e − 04 | 1.326e − 04 | 1.154e − 04 | 1.632e − 04 | 3.184e − 04 |
| | | Median | **7.458e − 05** | 1.335e − 04 | 1.631e − 04 | 1.536e − 04 | 2.547e − 04 | 4.368e − 04 |
| | | Worst | **1.102e − 04** | 1.421e − 04 | 1.967e − 04 | 1.859e − 04 | 5.456e − 04 | 1.559e − 03 |
| | 4 | Best | **2.682e − 05** | 4.819e − 05 | 7.765e − 05 | 8.132e − 05 | 8.569e − 05 | 9.142e − 05 |
| | | Median | **5.791e − 05** | 5.321e − 04 | 7.969e − 04 | 8.486e − 04 | 5.723e − 04 | 8.836e − 04 |
| | | Worst | **1.008e − 04** | 7.021e − 04 | 9.366e − 04 | 1.455e − 03 | 8.454e − 04 | 1.357e − 03 |
| | 5 | Best | **8.613e − 05** | 8.621e − 05 | 8.752e − 05 | 9.123e − 05 | 9.439e − 05 | 9.783e − 05 |
| | | Median | **1.193e − 04** | 1.323e − 04 | 1.456e − 04 | 1.523e − 04 | 1.717e − 04 | 1.986e − 04 |
| | | Worst | **2.101e − 04** | 2.458e − 04 | 2.569e − 04 | 2.834e − 04 | 5.956e − 04 | 3.549e − 04 |
| DTLZ3 | 3 | Best | **1.128e − 04** | 1.227e − 04 | 1.360e − 04 | 1.257e − 04 | 1.453e − 04 | 1.257e − 04 |
| | | Median | **1.402e − 04** | 1.411e − 04 | 1.587e − 04 | 1.605e − 04 | 1.742e − 04 | 1.926e − 04 |
| | | Worst | **2.597e − 04** | 2.625e − 04 | 2.767e − 04 | 2.858e − 04 | 2.903e − 04 | 2.759e − 04 |
| | 4 | Best | **1.325e − 04** | 1.423e − 04 | 1.560e − 04 | 1.568e − 04 | 1.954e − 04 | 2.856e − 04 |
| | | Median | **1.589e − 04** | 1.621e − 04 | 1.897e − 04 | 2.225e − 04 | 2.273e − 04 | 3.863e − 04 |
| | | Worst | **1.759e − 04** | 1.791e − 04 | 1.969e − 04 | 3.851e − 04 | 5.956e − 04 | 6.989e − 04 |
| | 5 | Best | 7.138e − 05 | **7.026e − 05** | 1.161e − 04 | 1.229e − 04 | 1.156e − 04 | 1.559e − 04 |
| | | Median | 1.090e − 04 | **1.014e − 04** | 1.363e − 04 | 1.454e − 04 | 1.557e − 04 | 1.852e − 04 |
| | | Worst | 2.599e − 04 | **2.423e − 04** | 2.567e − 04 | 2.859e − 04 | 2.456e − 04 | 2.558e − 04 |
| DTLZ4 | 3 | Best | **1.434e − 05** | 1.450e − 05 | 1.354e − 04 | 1.625e − 04 | 1.457e − 04 | 1.590e − 04 |
| | | Median | **1.658e − 05** | 1.752e − 05 | 1.811e − 04 | 1.757e − 04 | 1.956e − 04 | 1.869e − 04 |
| | | Worst | **1.766e − 05** | 1.781e − 05 | 1.867e − 04 | 1.853e − 04 | 2.156e − 04 | 1.954e − 04 |
| | 4 | Best | **1.524e − 05** | 1.556e − 05 | 1.667e − 05 | 1.659e − 05 | 1.853e − 05 | 1.771e − 05 |
| | | Median | **1.626e − 05** | 1.629e − 05 | 1.711e − 05 | 1.731e − 05 | 1.950e − 05 | 1.859e − 05 |
| | | Worst | **1.735e − 05** | 1.778e − 05 | 1.797e − 05 | 2.741e − 05 | 2.757e − 05 | 1.959e − 05 |
| | 5 | Best | **1.786e − 05** | 1.801e − 05 | 1.862e − 05 | 1.859e − 05 | 1.856e − 05 | 1.853e − 05 |
| | | Median | **1.938e − 05** | 2.024e − 05 | 3.767e − 05 | 2.850e − 05 | 5.456e − 05 | 3.559e − 05 |
| | | Worst | **7.809e − 05** | 8.421e − 05 | 8.760e − 05 | 8.857e − 05 | 8.454e − 05 | 8.554e − 05 |

objectives, and the one that is significantly better than the other is marked in bold face. Table 4 shows that all the test functions of Hmp/BBO performed well, especially in the DTLZ1 and DTLZ4 problems. Results of DTLZ test suite in comparison to results of Hmp/BBO with other five MOEAs in terms of IGD values on DTLZ test suite are presented in Table 5. It shows both the mean and IGD values of over 20 independent runs for the six compared MOEAs, where the best mean and standard deviation values are marked in bold. Furthermore, Table 6 presents the average and standard deviation of the Hypervolume over 20 independent runs on WFG problems, where the best performance is highlighted in bold. The quality of the solution sets obtained by the six algorithms on all WFG test problems in terms of Hypervolume was compared. From the experimental results of DTLZ and WFG test problem, we find that Hmp/BBO shows better performance than the other five algorithms.

Figure 3 shows the parallel coordinates of PF obtained by Hmp/BBO and the other five MOEAs with the five-objective of DTLZ1. This run is involved in the result closest to the mean IGD value. A parallel coordinates system is a visual representation of the overall performance of the algorithm for high-dimensional space, where each row represents a point in a high-dimensional space. If the lines on each object on the graph are distributed in $[0, 1]$, it is shown that the convergence of the better algorithm, if the lines can be evenly distributed in $[0, 1]$, indicates that the distribution of the algorithm is better. From this figure, it is found that the



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: Parallel coordinates figure by six algorithms on the five-objective DTLZ1. (a) Hmp/BBO; (b) BBO/Complex; (c) NSGAIII; (d) MOEA/D-PBI; (e) HYPE; (f) GREA.

Table 5: Best, median, and worst IGD values obtained by DTLZ1 to DTLZ4 with six algorithms by different number of objectives.

| Functions | $m$ | | Hmp/BBO | BBO/Complex | NSGAIII | MOEA/D-PBI | HYPE | GREA |
|---|---|---|---|---|---|---|---|---|
| DTLZ1 | 3 | Best | **3.114e − 04** | 3.380e − 04 | 3.424e − 04 | 4.274e − 04 | 4.071e − 04 | 4.644e − 04 |
| | | Median | **4.276e − 04** | 3.650e − 03 | 3.742e − 03 | 4.332e − 04 | 4.542e − 04 | 5.422e − 04 |
| | | Worst | **2.032e − 03** | 3.833e − 03 | 3.929e − 03 | 5.731e − 03 | 6.241e − 03 | 7.321e − 03 |
| | 4 | Best | 1.873e − 04 | **1.978e − 05** | 1.789e − 04 | 1.884e − 04 | 1.981e − 04 | 2.014e − 04 |
| | | Median | 1.956e − 04 | **1.916e − 04** | 2.033e − 04 | 2.075e − 04 | 2.837e − 04 | 4.9233e − 04 |
| | | Worst | 2.126e − 04 | **2.054e − 04** | 3.126e − 04 | 3.071e − 04 | 3.985e − 04 | 5.126e − 04 |
| | 5 | Best | **2.047e − 04** | 2.234e − 04 | 2.532e − 04 | 2.643e − 04 | 2.753e − 04 | 3.224e − 04 |
| | | Median | **2.273e − 04** | 2.301e − 03 | 2.611e − 03 | 2.831e − 04 | 3.817e − 04 | 4.125e − 04 |
| | | Worst | **3.162e − 04** | 2.706e − 03 | 2.754e − 03 | 2.927e − 03 | 3.914e − 04 | 4.512e − 04 |
| DTLZ2 | 3 | Best | **3.209e − 04** | 5.201e − 04 | 3.371e − 04 | 3.694e − 04 | 3.732e − 04 | 3.984e − 04 |
| | | Median | **4.561e − 04** | 5.335e − 04 | 6.631e − 04 | 7.536e − 04 | 8.547e − 04 | 9.368e − 04 |
| | | Worst | **6.255e − 04** | 6.421e − 04 | 7.767e − 04 | 7.859e − 04 | 7.956e − 04 | 9.559e − 04 |
| | 4 | Best | **4.682e − 05** | 5.717e − 05 | 6.765e − 05 | 7.732e − 05 | 8.579e − 05 | 9.742e − 05 |
| | | Median | **5.392e − 05** | 2.821e − 04 | 2.922e − 04 | 8.486e − 04 | 8.783e − 04 | 8.886e − 04 |
| | | Worst | **1.104e − 04** | 1.211e − 03 | 1.366e − 03 | 1.455e − 03 | 1.454e − 03 | 1.857e − 03 |
| | 5 | Best | **8.613e − 05** | 1.621e − 04 | 1.752e − 04 | 2.123e − 04 | 2.432e − 04 | 5.783e − 04 |
| | | Median | **7.687e − 04** | 1.922e − 03 | 1.956e − 03 | 2.513e − 03 | 2.727e − 03 | 2.926e − 03 |
| | | Worst | **1.038e − 03** | 2.458e − 03 | 2.569e − 03 | 2.834e − 03 | 2.956e − 03 | 3.549e − 03 |
| DTLZ3 | 3 | Best | **6.123e − 04** | 6.222e − 04 | 6.360e − 04 | 7.251e − 04 | 7.423e − 04 | 7.757e − 04 |
| | | Median | **1.587e − 03** | 1.616e − 03 | 1.784e − 03 | 1.905e − 03 | 1.942e − 03 | 1.926e − 02 |
| | | Worst | **3.929e − 03** | 4.695e − 03 | 7.555e − 03 | 2.858e − 02 | 2.903e − 02 | 3.359e − 02 |
| | 4 | Best | **1.516e − 04** | 1.826e − 04 | 1.961e − 04 | 1.968e − 04 | 2.114e − 04 | 2.552e − 04 |
| | | Median | **1.782e − 04** | 1.922e − 04 | 2.117e − 04 | 2.221e − 04 | 2.371e − 04 | 2.661e − 04 |
| | | Worst | **1.951e − 04** | 1.993e − 04 | 2.762e − 04 | 2.852e − 04 | 2.952e − 04 | 2.989e − 04 |
| | 5 | Best | 5.342e − 04 | 7.932e − 04 | **4.011e − 04** | 5.829e − 04 | 5.956e − 04 | 6.559e − 04 |
| | | Median | 1.137e − 03 | 1.514e − 03 | **1.113e − 03** | 1.154e − 03 | 1.157e − 03 | 1.252e − 03 |
| | | Worst | 1.574e − 03 | 2.423e − 03 | **1.537e − 03** | 1.859e − 03 | 1.956e − 03 | 2.151e − 03 |
| DTLZ4 | 3 | Best | **9.243e − 05** | 1.420e − 04 | 1.554e − 04 | 1.835e − 04 | 1.957e − 04 | 1.992e − 04 |
| | | Median | **1.092e − 04** | 1.752e − 04 | 2.211e − 04 | 2.777e − 04 | 2.856e − 04 | 2.969e − 04 |
| | | Worst | **1.892e − 04** | 1.971e − 04 | 2.867e − 04 | 2.893e − 04 | 2.916e − 04 | 2.984e − 04 |
| | 4 | Best | **1.231e − 05** | 1.526e − 05 | 1.667e − 05 | 1.669e − 05 | 1.543e − 05 | 1.541e − 05 |
| | | Median | **1.526e − 05** | 1.727e − 05 | 1.731e − 05 | 1.732e − 05 | 1.650e − 05 | 1.759e − 05 |
| | | Worst | **1.635e − 05** | 1.793e − 05 | 1.797e − 05 | 1.841e − 05 | 3.857e − 05 | 2.879e − 05 |
| | 5 | Best | 1.336e − 04 | 1.701e − 04 | 1.852e − 04 | **1.179e − 05** | 2.346e − 05 | 2.353e − 05 |
| | | Median | 1.392e − 04 | 2.024e − 04 | 2.267e − 04 | **1.451e − 05** | 2.556e − 05 | 2.759e − 05 |
| | | Worst | 2.052e − 04 | 8.421e − 04 | 9.760e − 04 | **1.817e − 05** | 9.854e − 04 | 9.954e − 04 |

Hmp/BBO is slightly better in convergence and diversity than BBO/Complex, NSGAIII, and MOEA/D-PBI. HYPE and GREA are far away from the PF because the maximum value of some objectives is much larger than 0.5. Obviously, the solutions achieved by HYPE and GREA are worse than the above four algorithms.

Performance scores are introduced to the global performance of compared algorithms. For a problem test question, suppose that there are $S$ algorithms $\mathrm{al}_1, \mathrm{al}_2, \ldots, \mathrm{al}_S$; let $\gamma_{ij}$ be 1, if $\mathrm{al}_i$ outperforms $\mathrm{al}_j$ in terms of IGD metric, and 0 otherwise. This value indicates how many other algorithms are better than $\mathrm{al}_i$ on the problem test question. Therefore, the smaller the value is, the better the algorithm is. Then, performance score $S(\mathrm{al}_i)$ is determined as

$$S\left(\mathrm{al}_i\right) = \sum_{\substack{j=1 \\ j\neq i}}^{s} \gamma_{ij}. \tag{9}$$

In Figure 4, the average performance scores for different number of objectives and different test problems are briefly summarized. Score of Hmp/BBO is smaller than other algorithms in the corresponding number of objectives.

Table 6: Average and standard deviation of the HV values on WFG1 to WFG9 with six algorithms by different number of objectives.

| Functions | $m$ | Hmp/BBO | BBO/Complex | NSGAIII | MOEA/D-PBI | HYPE | GREA |
|---|---|---|---|---|---|---|---|
| WFG1 | 3 | 4.232*e* − 01 | 4.806*e* − 01 | 5.138*e* − 01 | 4.821*e* − 01 | 5.385*e* − 01 | **7.235e − 01** |
| | 4 | 4.611*e* − 01 | 3.252*e* − 01 | 4.054*e* − 01 | 4.458*e* − 01 | **6.141e − 01** | 5.326*e* − 01 |
| | 5 | 5.340*e* − 01 | 4.587*e* − 01 | 5.032*e* − 01 | 5.023*e* − 01 | **5.953e − 01** | 5.224*e* − 01 |
| WFG2 | 3 | 8.658*e* − 01 | 8.538*e* − 01 | 8.616*e* − 01 | 7.184*e* − 01 | 8.485*e* − 01 | **8.901e − 01** |
| | 4 | 9.181*e* − 01 | 8.705*e* − 01 | 8.061*e* − 01 | 7.934*e* − 01 | 8.801*e* − 01 | **9.342e − 01** |
| | 5 | **9.523e − 01** | 9.401*e* − 01 | 9.232*e* − 01 | 9.101*e* − 01 | 9.239*e* − 01 | 9.324*e* − 01 |
| WFG3 | 3 | 8.032*e* − 01 | 8.127*e* − 01 | 8.311*e* − 01 | 9.102*e* − 01 | **9.333e − 01** | 9.202*e* − 01 |
| | 4 | **8.926e − 01** | 8.422*e* − 01 | 7.552*e* − 01 | 7.456*e* − 01 | 8.014*e* − 01 | 8.716*e* − 01 |
| | 5 | **9.337e − 01** | 8.323*e* − 01 | 8.001*e* − 01 | 7.123*e* − 01 | 8.326*e* − 01 | 7.007*e* − 01 |
| WFG4 | 3 | 7.212*e* − 01 | 7.362*e* − 01 | 7.852*e* − 01 | 6.952*e* − 01 | 8.654*e* − 01 | **8.852e − 01** |
| | 4 | **9.012e − 01** | 9.002*e* − 01 | 8.567*e* − 01 | 8.688*e* − 01 | 8.893*e* − 01 | 8.324*e* − 01 |
| | 5 | **9.386e − 01** | 9.201*e* − 01 | 8.862*e* − 01 | 8.559*e* − 01 | 8.756*e* − 01 | 8.453*e* − 01 |
| WFG5 | 3 | 6.331*e* − 01 | 6.433*e* − 01 | 6.321*e* − 01 | 6.512*e* − 01 | 6.213*e* − 01 | **6.790e − 01** |
| | 4 | **6.823e − 01** | 6.522*e* − 01 | 6.247*e* − 01 | 5.955*e* − 01 | 5.896*e* − 01 | 6.001*e* − 01 |
| | 5 | 6.743*e* − 01 | **6.801e − 01** | 6.363*e* − 01 | 6.442*e* − 01 | 6.782*e* − 01 | 6.553*e* − 01 |
| WFG6 | 3 | 6.832*e* − 01 | 6.655*e* − 01 | 6.324*e* − 01 | 6.725*e* − 01 | **6.859e − 01** | 6.531*e* − 01 |
| | 4 | **6.521e − 01** | 6.454*e* − 01 | 6.447*e* − 01 | 6.353*e* − 01 | 6.553*e* − 01 | 6.321*e* − 01 |
| | 5 | **6.772e − 01** | 6.521*e* − 01 | 6.232*e* − 01 | 6.325*e* − 01 | 6.526*e* − 01 | 6.326*e* − 01 |
| WFG7 | 3 | 7.137*e* − 01 | 7.150*e* − 01 | **7.254e − 01** | 7.025*e* − 01 | 6.957*e* − 01 | 7.012*e* − 01 |
| | 4 | 7.024*e* − 01 | 7.536*e* − 01 | 7.237*e* − 01 | **7.558e − 01** | 7.001*e* − 01 | 7.121*e* − 01 |
| | 5 | **7.874e − 01** | 7.321*e* − 01 | 6.969*e* − 01 | 7.051*e* − 01 | 7.326*e* − 01 | 7.353*e* − 01 |
| WFG8 | 3 | **6.231e − 01** | 6.020*e* − 01 | 6.151*e* − 01 | 5.925*e* − 01 | 6.050*e* − 01 | 6.012*e* − 01 |
| | 4 | 6.020*e* − 01 | **6.552e − 01** | 6.327*e* − 01 | 6.239*e* − 01 | 6.153*e* − 01 | 6.271*e* − 01 |
| | 5 | **6.286e − 01** | 6.001*e* − 01 | 6.112*e* − 01 | 6.139*e* − 01 | 6.210*e* − 01 | 6.213*e* − 01 |
| WFG9 | 3 | 6.144*e* − 01 | 6.235*e* − 01 | 6.312*e* − 01 | 6.121*e* − 01 | **6.352e − 01** | 6.124*e* − 01 |
| | 4 | **6.441e − 01** | 6.348*e* − 01 | 6.320*e* − 01 | 6.259*e* − 01 | 6.404*e* − 01 | 6.472*e* − 01 |
| | 5 | **6.281e − 01** | 6.101*e* − 01 | 6.161*e* − 01 | 6.254*e* − 01 | 6.056*e* − 01 | 6.113*e* − 01 |

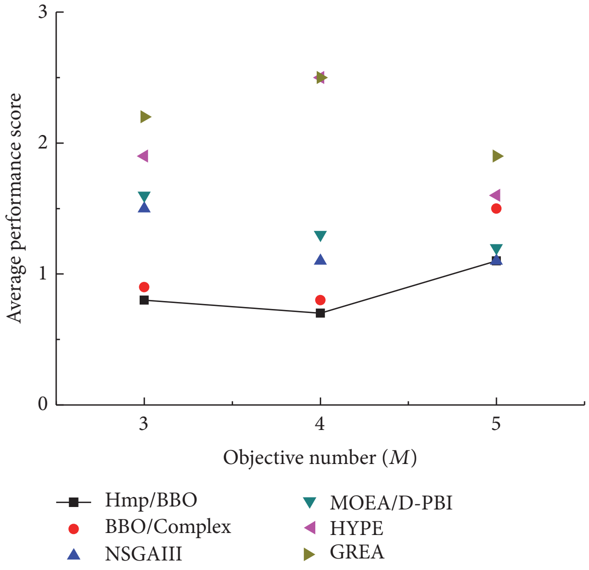


Figure 4: Average performance score over all test problems for different number of objectives. The smaller the score, the better the performance in the corresponding number of objectives.

## 5. Conclusions

In this paper, we propose a novel many-objective optimization algorithm called Hmp/BBO. It addresses the problem of a many-objective optimization which uses the mechanism of adaptive hierarchical. The problem is decomposed into many subsystems through PBI aggregate function. Through combining the advantages of MOEA/D, Hmp/BBO is expected to enhance the convergence ability and the valuable diversity performance. As it can be seen in the results obtained in Section 4, Hmp/BBO performance can be enhanced by PBI aggregation function and the control the migration flow of $n$ SIV between rich and poor islands. Within this process, the old features will not always be overwritten by the newly emigrated features from other islands. It has more flexible decomposition optimization options compared to other five state-of-the-art algorithms.

The future work will discuss the Hmp/BBO in terms of objective number greater than five and in terms of solving constrained many-objective problems. The constrained many-objective problems seem to be a new challenge for MaOPs, and we try to extend our algorithm to solve them by combining constraint handling techniques. And also we would want to apply Hmp/BBO to solve more real-world problems.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National 863 Project, China (no. 2015AA043003) and the Hubei Provincial Department of Education Instructive Foundation China (no. B2016084).

## References

[1] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: a survey," *ACM Computing Surveys*, vol. 48, no. 1, article 13, 2015.

[2] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: a short review," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 2419–2426, Hong Kong, June 2008.

[3] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[4] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proceedings of the 2002 Annual Meeting of the North American on Fuzzy Information Processing Society (NAFIPS '02)*, vol. 71B, pp. 233–238, New Orleans, La, USA, June 2002.

[5] T. Murata and A. Taki, "Many-objective optimization for knapsack problems using correlation-based weighted sum approach," in *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO '09)*, pp. 468–480, ACM, Nantes, France, April 2009.

[6] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 183–195, 2011.

[7] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[9] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[10] J. Fleming Peter, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: an engineering design perspective," in *Evolutionary Multi-Criterion Optimization. EMO 2005*, vol. 3410 of *Lecture Notes in Computer Science*, pp. 14–32, Springer, 2005.

[11] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[12] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '08)*, pp. 673–680, Atlanta, Ga, Usa, July 2008.

[13] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[14] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Proceedings of the Congress on Evolutionary Computation*, vol. 2, pp. 957–962, IEEE, Seoul, Republic of Korea, May 2001.

[15] K. Deb, M. Mohan, and S. Mishra, "Evaluating the $\varepsilon$-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.

[16] D. K. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: employing correntropy and a novel maximum variance unfolding," in *Evolutionary Multi-Criterion Optimization*, vol. 4403 of *Lecture Notes in Computer Science*, pp. 772–787, Springer, Berlin, Germany, 2007.

[17] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 832–842, 2015.

[18] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[19] D. Du and D. Simon, "Complex system optimization using biogeography-based optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 456232, 17 pages, 2013.

[20] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *American Association for the Advancement of Science. Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[21] V. Černý, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[22] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[23] Z. Zhang, C. Gao, Y. Lu, Y. Liu, and M. Liang, "Multi-objective ant colony optimization based on the *Physarum*-inspired mathematical model for bi-objective traveling salesman problems," *PLoS ONE*, vol. 11, no. 1, Article ID e0146709, 2016.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[25] R. W. Eglese, "Simulated annealing: a tool for operational research," *European Journal of Operational Research*, vol. 46, no. 3, pp. 271–281, 1990.

[26] J. Bard, "Engineering optimization: theory and practice, Third Edition," *History of Economic Thought*, vol. 52, no. 5, pp. 27–45, 1996.

[27] G. V. Gonzales, E. S. D. Estrada, L. R. Emmendorfer et al., "A comparison of simulated annealing schedules for constructal design of complex cavities intruded into conductive walls with internal heat generation," *Energy*, vol. 93, pp. 372–382, 2015.

[28] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Information Sciences*, vol. 282, pp. 363–387, 2014.

[29] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[30] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

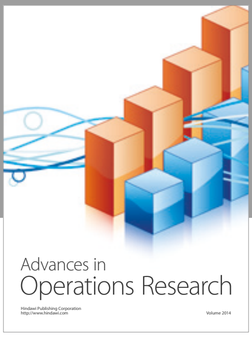
[31] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization," in *Proceedings of the International Conference on Genetic Algorithms*, pp. 416–423, 1999.

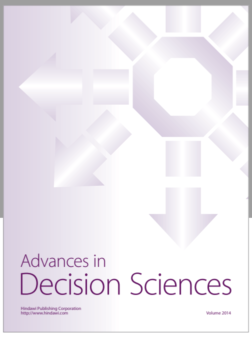
[32] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

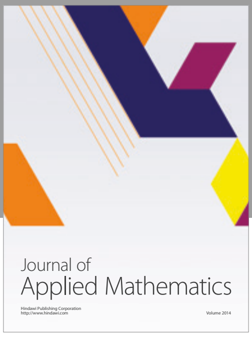
[33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pp. 105–145, Springer, London, UK, 2005.

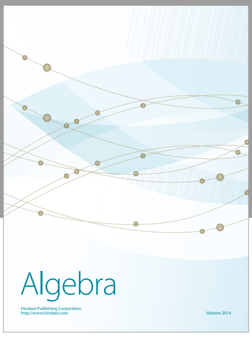
[34] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

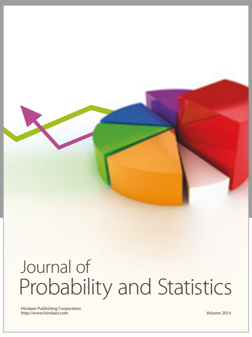
[35] K. Narukawa, Y. Tanigaki, and H. Ishibuchi, "Evolutionary many-objective optimization using preference on hyperplane," in *16th Genetic and Evolutionary Computation Conference, GECCO 2014*, pp. 91–92, can, July 2014.


Advances in
Operations Research

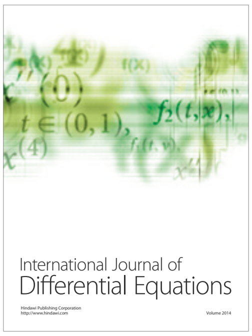

Advances in
Decision Sciences

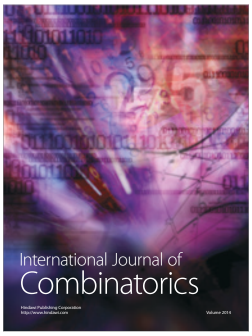

Journal of
Applied Mathematics

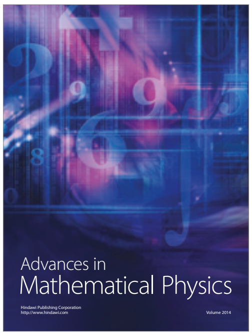

Algebra

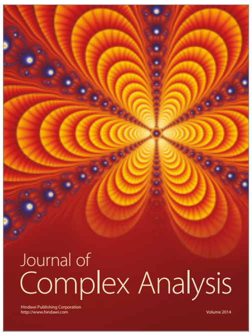

Journal of
Probability and Statistics

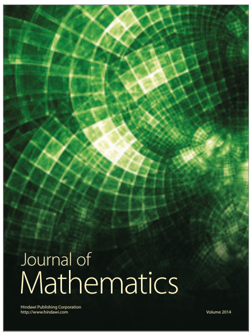
The Scientific
World Journal

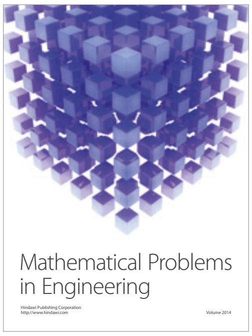

International Journal of
Differential Equations

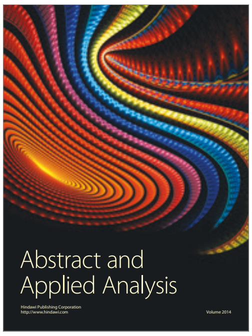

International Journal of
Combinatorics

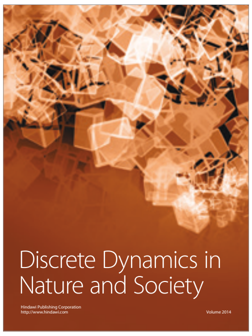

Advances in
Mathematical Physics

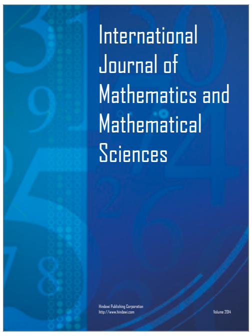

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

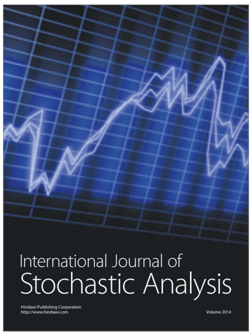

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization