A Multi-Algorithm Balancing Convergence and Diversity for Multi-Objective Optimization*

DATONG XIE1,2, LIXIN DING1, YURONG HU1, SHENWEN WANG1,2, CHENGWANG XIE4 AND LEI JIANG1,5
1State Key Lab of Software Engineering
School of Computer
Wuhan University
Wuhan, 430000 P.R. China
2Department of Information Management Engineering
Fujian Commercial College
Fuzhou, 350000 P.R. China
3Department of Information Engineering
Shijiazhuang University of Economics
Shijiazhuang, 050000 P.R. China
4School of Software
East China Jiao Tong University
Nanchang, 330000 P.R. China
5Key Laboratory of Knowledge Processing and Networked Manufacture
Hunan University of Science and Technology
Xiangtan, 411201 P.R. China

As a population-based method, evolutionary algorithms have been extensively used to solve multi-objective optimization problems. However, most of the current multi-objective evolutionary algorithms (MOEAs) cannot strike a good balance between the closeness to the true Pareto front and the uniform distribution of non-dominated solutions. In this paper, we present a multi-algorithm, MABNI, which is based on two popular MOEAs, NSGA-II and IBEA. The proposed algorithm is inspired from the strengths and weaknesses of the two algorithms, e.g., the former can preserve extreme solutions effectively but has a worse diversity while the latter shows a better convergence and makes non-dominated solutions more evenly distributed but easily suffers losses of extreme solutions. In MABNI, modified NSGA-II and IBEA run alternatively and the update principle for the archive population is based on the distances to nearest neighbors. Furthermore, accompanied with preservation of extreme points, an improved differential evolution is employed to speed the search. The performance of MABNI is examined on ZDT-series and DTLZ-series test instances in terms of the selected performance indicators. Compared with NSGA-II and IBEA, the results indicate that MABNI can reach a better balance between convergence and diversity for the approximation of the true Pareto front and obtain more stable results.

Keywords: multi-algorithm, multi-objective optimization, evolutionary algorithm, nearest neighbor, extreme solution

1. INTRODUCTION

Many real-world optimization problems are multi-objective optimization problems...
MOPs which have multiple objectives to be optimized. More often, the objectives contradict one another. Thus, it is impossible to optimize all the objectives simultaneously. In other words, for globally optimal solutions in a MOP, any improvement in certain objectives must cause degradation of at least one other objective. Thereby, it is preferable to obtain a set of incomparable optimal solutions providing for decision makers.

In recent decades, evolutionary algorithms have received increasing concerns owing to the merits of derivative-free, simplicity and flexibility. Moreover, due to a population based mechanism, evolutionary optimization algorithms can find multiple solutions simultaneously unlike conventional methods which can only find a single solution at each iteration. Therefore, a number of multi-objective evolutionary algorithms (MOEAs) have been developed to deal with different engineering design or scheduling problems with conflicting objectives [1].

Most of these algorithms apply Pareto-based ranking method while others not. They have their own advantages. Take the following two algorithms for example. The Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [2] is one of the most prominent MOEAs using Pareto-based ranking method. In single-objective optimization, a total order relation based on the single function value can be easily used to rank the solutions. But such an apparent total order relation does not exist in multi-objective optimization. In Pareto-based ranking method, comparisons among the solutions are conducted under the dominance relation where a partial order is defined to rank the solutions. NSGA-II employs non-dominated sorting to assign a rank to every individual and calculates their crowding distances. Then, the selection mechanism is based on the ranks and the crowding distances. The Pareto-based ranking scheme can identify the non-dominated solutions set effectively. However, it requires an additional technique to maintain the diversity of the non-dominated front. Although the task is handled by the crowding distance method in NSGA-II, the accuracy decreases as the number of objective functions increases [3, 4].

Thus, it seems to have disadvantages in dealing with many-objective optimization problems due to a strict partial ordering relation of Pareto dominance [5]. Many researchers have tried to find a total ordering relation in order to select individuals to survive to the next generation as scalar optimization problems. Zitzler et al. [6] proposed Indicator-Based Evolutionary Algorithm (IBEA) to induce a total order of the approximate set in the objective space. IBEA is an effective non-Pareto evolutionary algorithm for solving MOPs, and does not require any additional diversity preservation mechanisms.

So far, there exist many other improved approaches based on a single algorithm above for solving multi-objective optimization problem. However, a single algorithm usually has both strengths and weaknesses. As we know, there are two goals in multi-objective optimization. The first goal suggests that the approximate Pareto front should approach as close as possible to the true Pareto front. The second goal requires that the front should be uniformly distributed and cover the region of the true Pareto front as far as possible. The goals, in fact, can be briefly represented as convergence and diversity respectively in the objective space. They are not conflicting in nature. However, most MOEAs cannot reach a satisfying balance between them. A typical example is that the approximation capability of NSGA-II is superior to the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [7] while the latter has the advantage over the former in diversity preservation [3]. Also, IBEA has a better convergence performance than NSGA-II and SPEA2 but a worse covering capacity [6].
In recent years, multi-algorithm has received ever-increasing attention from quite a few researchers. Multi-algorithm employs multiple algorithms to solve problems simultaneously. There have been many researches which use multi-algorithm to solve different problems [8-12]. We note that multi-algorithm looks similar to ensemble learning [13]. However, there is an essential difference between them. Ensemble learning employs multiple learners to get multiple models and ultimately the models are combined to form an integrated model. In contrast, multi-algorithm utilizes multiple algorithms to produce the final result directly. Inspired by this idea, we present a multi-algorithm based on NSGA-II and IBEA for solving multi-objective optimization (MABNI). The method makes NSGA-II and IBEA run on the same population alternately so as to keep a balance between convergence and diversity. In addition, we modify the two algorithms. First, for the preservation of non-dominated solutions, we add an archive population which is not used in the two original algorithms explicitly. Then, differential evolution (DE) [14], as a variation operator, is employed in the two algorithms due to its powerful search capability in decision space. Meanwhile, in order to make the solutions evenly distributed along the non-dominated front, the individuals with the shortest distances to its nearest neighbors are removed from the archive population while the number of non-dominated solutions exceeds the archive population size.

The paper proceeds as follows. In section 2, we introduce related background information. Section 3 is devoted to discussing the proposed multi-algorithm. Then, the experimental results and statistical comparison among MABNI, NSGA-II and IBEA on the well-known multi-objective test instances, ZDT-series and DTLZ-series, are shown in section 4. Finally, conclusions are drawn in section 5.

2. RELATED WORKS

2.1 Problem Formulation and Some Concepts

Without loss of generality, we assume that all objectives are to be minimized. Then, MOP can be formalized as Eq. (1).

\[
\text{Minimize } Z(\bar{x}) = \{z_1 = f_1(\bar{x}), z_2 = f_2(\bar{x}), \ldots, z_m = f_m(\bar{x})\}
\]

s.t. \(g_i(\bar{x}) \leq 0, i = 1, 2, \ldots, r\) \hspace{1cm} (1)

Where \(\bar{x} \in \mathbb{R}^n\) is a vector with \(n\) decision variables, \(f_1(\bar{x}), \ldots, f_m(\bar{x})\) are objective functions, \(g_1(\bar{x}), \ldots, g_r(\bar{x})\) are constraint functions. The decision space can be denoted as \(S = \{\bar{x} \in \mathbb{R}^n | g_i(\bar{x}) \leq 0, i = 1, 2, \ldots, r\}\). A solution in \(S\) is called a feasible solution. The objective space can be represented as \(Z = \{z \in \mathbb{R}^m | z_1 = f_1(\bar{x}), z_2 = f_2(\bar{x}), \ldots, z_m = f_m(\bar{x}), \bar{x} \in S\}\).

For the convenience of later discussion, here we give the definitions of some terms frequently used in multi-objective optimization [5].

**Definition 1 (Pareto Dominance, \(\succ\))**

A solution \(\bar{x}_1\) is said to dominate another solution \(\bar{x}_2\), denoted by \(\bar{x}_1 \succ \bar{x}_2\) if and only if

\[
\forall i \in \{1, \ldots, m\}: f_i(\bar{x}_1) \leq f_i(\bar{x}_2) \land \exists j \in \{1, \ldots, m\}: f_j(\bar{x}_1) < f_j(\bar{x}_2).
\]
Otherwise, $\bar{X}_1$ does not dominate $\bar{X}_2$ (denoted by $\bar{X}_1 \not\succeq \bar{X}_2$).

**Definition 2** (Non-dominated, $\sim$)
A solution $\bar{X}_1$ is said to be non-dominated with another solution $\bar{X}_2$, denoted by $\bar{X}_1 \sim \bar{X}_2$, if and only if
\[\bar{X}_1 \not\succeq \bar{X}_2 \land \bar{X}_2 \not\succeq \bar{X}_1.\]

**Definition 3** (Non-dominated set, NDS)
A set $B$ is said to be a non-dominated set if it satisfies the property that:
\[\forall \bar{X}_1, \bar{X}_2 \in B, \bar{X}_1 \sim \bar{X}_2.\]

**Definition 4** (Pareto optimal solution)
A solution $\bar{X}^* \in S$ is called a Pareto optimal solution if it satisfies the property that:
\[\neg\exists \bar{X} \in S: \bar{X} \succ \bar{X}^*.\]

**Definition 5** (Pareto optimal set)
The Pareto optimal set, denoted by $PS$, is the set of all Pareto optimal solutions.

**Definition 6** (Pareto front)
The Pareto front, $PF$, is defined as $PF = \{z(\bar{X}_2) \in Z | \bar{X} \in PS\}$.

In accordance with Definition 5, $PS$ is the ultimate goal of optimization. However, the $PS$ of MOP, especially for continuous optimization problem, cannot be completely represented owing to the enormous amount of solutions. Moreover, too many solutions make no sense to help decision makers. Therefore, an appropriate selection for multi-objective optimization is to find a non-dominated set representing the Pareto optimal set as far as possible.

### 2.2 NSGA-II

NSGA-II is an improved version of NSGA [15]. In NSGA-II, for each solution, one has to determine the number of solutions which dominate it and the set of solutions to which it dominates. Finally, every individual is assigned a rank. In addition, NSGA-II estimates the densities of solutions having the same rank. The density for each solution is an accumulation of one-dimensional L1 distance between the nearest neighbors to the left and right of the solution in every sorted objective. This value is also called crowding distance. During selection, NSGA-II employs a crowded-comparison operator which takes into account both rank and crowding distance. Concretely, an individual with a low rank has precedence over the one with a high rank. If the individuals have the same rank, i.e., they are non-dominated with each other, crowding distance will be referred to as a surviving indicator so as to preserve the diversity of the non-dominated front. Instead of an archive population, an elitist mechanism combing the best parents and the best offspring is employed to prevent the excellent individuals being lost.
2.3 IBEA

As described by Zitzler et al. [6], IBEA is based on quality indicators where a function $I$ assigns each Pareto set approximation a real value reflecting its quality, and the optimization goal becomes the identification of a Pareto set approximation that minimizes (or maximizes) $I$. IBEA only compares pairs of individuals instead of entire approximation sets. The main advantage of the indicator concept is that no additional diversity preservation mechanisms are required. Zitzler provided an experimental verification which indicated that IBEA yielded better results than SPEA2 and NSGA-II.

As defined in [6], we assume that the optimization goal is given according to a binary quality indicator. A binary quality indicator is a function that maps multiple Pareto set approximations to a real number, and can be used to compare the quality of two Pareto set approximations. A binary quality indicator can be viewed as a natural extension of the Pareto dominance relation, and therefore can directly be used for fitness calculation similarly to the common Pareto-based fitness assignment schemes.

The indicator also can be used to compare two single solutions, and thus can serve for the selection process of evolutionary algorithms. Here we only introduce the epsilon indicator $I_\varepsilon$ which will be used in MABNI. The indicator quantifies the difference in quality between two solutions. It is defined as Eq. (2).

$$I_\varepsilon(\tilde{X}_1, \tilde{X}_2) = \max_i (f_i(\tilde{X}_1) - f_i(\tilde{X}_2)), \ i \in \{1, \ldots, m\}$$

Fig. 1. Illustration of the $I_\varepsilon$. (Here, $\tilde{X}_1 - \tilde{X}_2, \tilde{X}_2 > \tilde{X}_3$).

The indicator value represents the minimal offset for the objective values of $\tilde{X}_1$ to decrease so that $\tilde{X}_1$ weakly dominates $\tilde{X}_2$. It can be seen from Fig. 1 the indicator function can take negative values as well as positive values.

In order to assign fitness to the individuals, and produce a total ordering relation of the whole population $P$, one can choose one of the follow two schemes:

1. One method is to simply sum up the indicator values for each individual paired with the rest of the population as shown in Eq. (3).
\[ F(\bar{X}_1) = \sum_{\bar{X}_2 \in P(\bar{X}_1)} I(\bar{X}_2, \bar{X}_1) \]  

(3)

(2) Another approach is to amplify the influence of dominating individuals over dominated ones as shown in Eq. (4). Here, the parameter \( k \) is a scaling factor depending on the indicator and the problem.

\[ F(\bar{X}_1) = \sum_{\bar{X}_2 \in P(\bar{X}_1)} e^{-I(\bar{X}_2, \bar{X}_1)}/k \]  

(4)

As proved in [6], \( I \) is dominance preserving, i.e., satisfying transitivity.

**Definition 7 (Dominance preserving) [6]** A binary quality indicator \( I \) is denoted as dominance preserving if:

1. \( \forall \bar{X}_1, \bar{X}_2 \in S, \bar{X}_1 \succ \bar{X}_2 \Rightarrow I(\bar{X}_1, \bar{X}_2) < I(\bar{X}_2, \bar{X}_1) \), and
2. \( \forall \bar{X}_1, \bar{X}_2, \bar{X}_3 \in S, \bar{X}_1 \succ \bar{X}_2 \Rightarrow I(\bar{X}_3, \bar{X}_1) \geq I(\bar{X}_3, \bar{X}_2) \)

\section*{3. THE PROPOSED MULTI-ALGORITHM}

\subsection*{3.1 The Motivation}

It is hard to achieve the two goals of multi-objective algorithm simultaneously, i.e., close approximation and uniform distribution.

NSGA-II is an efficient multi-objective optimization algorithm with good convergence performance. As a result of crowding distance based selection, the extreme solutions are propagated to the next generation, and thus lead to a wide coverage. In addition, NSGA-II can obtain a good diversity of the solutions in the case of two objectives. However, with the number of objectives increasing, the crowding distance fails to obtain uniformly distributed solutions along the approximate Pareto front. This weakness is caused by the fake information provided by the sorting procedure which cannot guarantee the components of distances come from the same individuals.

In IBEA, non-dominated sorting and explicit diversity preservation are not required. It merely employs the indicator values to guide the search. An indicator value reflects the differences between the two individuals while an accumulative indicator value represents the quality rank of each individual in the whole population. It is known that IBEA has a better convergence than NSGA-II and can obtain a solution set which is distributed evenly. However, it is disadvantageous to maintain extreme solutions because they usually have a larger indicator value according to Eq. (5) \((k = 0.02)\). A simple example is shown as follows. As depicted in Fig. 2, \( S_1, S_2 \) and \( S_3 \) are solutions in objective space. According to the objective values of the points, we can calculate their indicator values, \( i.e., F(S_1) = 6.24875E-07, F(S_2) = 3.90777E-13, F(S_3) = 4.93958E-10 \). Provided that one solution would be removed from the population, it is apparent that \( S_1 \) should be selected.

Inspired by the strengths and weaknesses of the two algorithms, a multi-algorithm based on NSGA-II and IBEA (MABNI) is implemented as follows.
3.2 Implementation

We modify the primal NSGA-II and adaptive IBEA [6] by applying DE/1/rand/bin [14] to parent population and adding an archive population which maintains the current Pareto approximation set. Meanwhile, Eq. (4) is converted into Eq. (5) so as to keep the selection of indicator values in conformity with the optimization of the objectives, \( i.e., \) minimization. Additionally, we also employ some measures to promote the convergence and distribution performance of the results. The detailed description is shown as follows.

\[
F(x_i) = \sum_{x_j \in P \setminus \{x_i\}} e^{-F(x_j, x_i) / \left( \max_{x_k \in \{x_i, x_j\}} d(x_k, x^*) \right)}
\]  

(5)

3.2.1 Initialization

In the initial population, parent population, each variable is assigned a random value according to a uniform distribution over its range.

3.2.2 Operators

Operators usually play important roles in evolutionary algorithms. As declared by Tobias Blickle and Lothar Thiele [16], the balance between exploitation and exploration can be adjusted either by the selection pressure of the selection operator or by the variation operators.

In our algorithm, the winner of two individuals selected from the parent population will perform polynomial mutation [2]. If the two individuals are non-dominated with each other, for IBEA, one of them will be randomly selected to perform mutation, and for NSGAII, the individual with minimal rank or same rank but maximal crowding distance will be chosen to perform mutation. Then, the new individual is put into the offspring population. The procedure is repeated until the offspring population is filled.

In addition, the parent population will perform DE/1/rand/bin. To better adapt multi-
objective optimization, a modified differential evolution is employed to speed the searching process and enhance the capability to search objective extreme solutions. Concretely, we choose the best of three randomly selected individuals as the base individual and place the better of the remaining two individuals at the front of the differential item. The layout of three individuals is supposed to speed the convergence. In single objective optimization, the trial individual replaces the target individual using the greedy selection mechanism. For multi-objective optimization, three relations exist between the trial individual and the target individual, i.e., dominative, non-dominated and dominated. If the former dominates the latter, the replacement is performed definitely. In the second case, the counts of non-dominated relation between each of the two individuals and the archive population will be referred to as a selection criterion. More specifically, if the count of the former is larger than the count of the latter, the replacement will take place. Otherwise, the former replaces the latter with a certain probability (we choose 0.5 in the forthcoming experiments). The replacement criterion is expected to enhance the search capacity towards the extreme objective values because extreme solutions usually have higher non-dominated counter than others.

If the above two variation operators were completed, the parent population and the offspring population will be merged and compete to construct the new parent population. In IBEA, the individuals with high indicator values are removed from the merged population until the number of the remainder is equal to the parent population size. In NSGAII, the individuals survive according to their ranks and crowding distances. Meanwhile, the archive population will be updated with the merged population.

3.2.3 Preservation of extreme solutions

Extreme solutions are the points which have best objective value for one objective in the current population. They are not only benefit for the convergence but also necessary to get a solution set with a wide coverage over the objective space.

Except for the measure used in DE to search extreme solutions, we also modify IBEA to preserve extreme solutions. Owing to the extreme solutions being prone to be lost in IBEA while generating the next parent population, one extreme solution of each objective will be kept in the parent population and thus further exploitation to obtain better extreme solutions is possible.

3.2.4 Further promotion of diversity

Although IBEA can find more evenly distributed solutions than NSGA-II, it still cannot obtain more ideal uniformity than some other algorithms such as SPEA2. Therefore, we will introduce a truncation criterion which uses nearest neighbors to update the archive population in both algorithms. Specifically, when the number of non-dominated solutions exceeds the archive population size, the cumulative Euclidean distance to its nearest $k$ neighbors for every non-dominated solution will be calculated. Here, $k$ is set to $2 \times (m - 1)$ and $m$ is the number of objectives. The individuals with smallest densities will be removed one by one accompanying with the information concerning nearest neighbors being updated until the quantity of the rest is equal to the archive population size.
Algorithm 1: A Modified NSGA-II

Input: \( NP \): the parent population size  
\( NO \): the offspring population size  
\( NA \): the archive population size  
\( Max\_Evals \): Maximum function evaluations in one run  
Output: \( PS \): the approximate Pareto set

Step 1: Initialization

Step 1.1: Set \( n = 0 \). /* \( n \) denotes the current generation number*/
Step 1.2: Initialize the Parent Population \( P_n \) /* \( P_n \) denote the parent population at \( n \)-th generation*/
Step 1.3: Perform non-dominated sorting, and calculate crowding distances on \( P_n \)
Step 1.4: Copy the individuals with rank 1 to \( A_n \) /* \( A_n \) denotes the archive population at \( n \)th generation */
Step 1.5: Set \( Evals = NP \) /* \( Evals \) denotes the current function evaluations */

Step 2: Evolution

Step 2.1: Set \( n = n + 1 \)
Step 2.2: Mutate the individuals chosen from \( P_{n-1} \) by binary tournament selection, generate an offspring population \( O_n \) and Update \( Evals \)
Step 2.3: Perform DE/1/rand/bin on \( P_{n-1} \) and Update \( Evals \)
Step 2.4: \( U_n = P_{n-1} \cup O_n \) /* \( U_n \) is a union population */
Step 2.5: Perform non-dominated sorting, calculate crowding distance on \( U_n \)
Step 2.6: Generate \( P_n \) based on the union population
Step 2.7: Update \( A_{n-1} \) using the nearest-neighbor distances and Get \( A_n \)
Step 2.8: If \( Evals < Max\_Evals \) then goto Step 2, else goto Step 3

Step 3: Output the results

Algorithm 2: A Modified IBEA

Input: \( NP \): the parent population size  
\( NO \): the offspring population size  
\( NA \): the archive population size  
\( Max\_Evals \): Maximum function evaluations in one run  
Output: \( PS \): the approximate Pareto set

Step 1: Initialization

Step 1.1: Set \( n = 0 \). /* \( n \) denotes the current generation number*/
Step 1.2: Initialize the Parent Population \( P_n \) /* \( P_n \) denote the parent population at \( n \)th generation*/
Step 1.3: Calculate indicator values for \( P_n \) using Eq. (5)
Step 1.4: Identify non-dominated solutions and put them into \( A_n \) /* \( A_n \) denotes the archive population at \( n \)th generation */
Step 1.5: Set \( Evals = NP \) /* \( Evals \) denotes the current function evaluations */

Step 2: Evolution

Step 2.1: Set \( n = n + 1 \)
Step 2.2: Mutate the individuals chosen from \( P_{n-1} \) by binary tournament selection, generate an offspring population \( O_n \) and Update \( Evals \)
Step 2.3: Perform DE/1/rand/bin on \( P_{n-1} \) and Update \( Evals \)
Step 2.4: \( U'_n = P_{n-1} \cup O_n \) /* \( U_n \) is a union population */

Step 2.5: Calculate indicator values for \( U_n \) and Identify non-dominated solutions

Step 2.6: Remove the worst from \( U_n \) according to their indicator values and update the indicator values of the remainder simultaneously, until the size of remainder is equal to the parent population size and generate \( P_n \)

Step 2.7: Update \( A_{n-1} \) using the nearest-neighbor distances and Get \( A_n \)

Step 2.8: If \( Evals < \text{Max}_Evals \) then goto Step 2, else goto Step 3

Step 3: Output the results

3.2.5 Algorithm framework

The pseudo codes of the modified NSGA-II and IBEA are shown in Algorithms 1 and 2 respectively.

In the proposed multi-algorithm, the key steps of the modified NSGA-II and IBEA run on the same populations alternately. Essentially speaking, MABNI mainly utilize the selection/reproduction operators of them. In other words, NSGA-II and IBEA use different strategies to generate the next parent population. In NSGA-II, the reproduction operator gives priority to the non-dominated solutions. That is, it puts emphasis on convergence. In contrast, IBEA makes the solutions of the new parent population uniformly distributed. Note here that the reproduction operators are repeated in turn instead of in a certain probability. The scheme can prevent the parent population from the loss of convergence and uniformity in objective space by using different reproduction operators alternatively. The pseudo code of the multi-algorithm is shown in Algorithm 3.

4. EXPERIMENTS

In this section, we experimentally investigate the performance of MABNI, NSGA-II and IBEA on ZDT-series and DTLZ-series using three performance metrics and compare the results statistically. ZDT-series are two-objective problems. In our experiments, the dimension of decision variable is set to 30 for ZDT1, ZDT2 and ZDT3, 10 for ZDT4 and ZDT6. DTLZ-series are scalable problems. In DTLZ-series, the number of decision variable is equal to \( M + K - 1 \), where \( M \) is the number of objectives and \( K \) can be defined by users. Here, \( M \) is set to 3. For DTLZ1, \( K \) is set to 5. For DTLZ2-6, \( K \) is set to 10. For DTLZ7, \( K \) is set to 20. Thus, the dimension of decision variable is equal to 7 for DTLZ1, 12 for DTLZ2-6, 22 for DTLZ7 respectively. All the experiments are implemented in VC6.0, Matlab 7.1, Sigmaplot 12.0 and run on Intel i5 760 2.8GHz machines with 4GB DDR3 RAM.

Algorithm 3: MABNI

Input: \( NP \): the parent population size
\( NO \): the offspring population size
\( NA \): the archive population size
\( \text{Max}_Evals \): Maximum function evaluations in one run
Output: $PS$: the approximate Pareto set

Step 1: Initialization

Step 1.1: Set $n=0$. /* $n$ denotes the current generation number */
Step 1.2: Initialize the Parent Population $P_n$ /* $P_n$ denote the parent population at $n$-th generation */
Step 1.3: Perform non-dominated sorting, and calculate crowding distances on $P_n$
Step 1.4: Copy the individuals with rank 1 to $A_n$ /* $A_n$ denotes the archive population at $n$-th generation */
Step 1.5: Set $Evals = NP$ /* $Evals$ denotes the current function evaluations */

Step 2: Evolution

Step 2.1: Set $n=n+1$
Step 2.2: Mutate the individuals chosen from $P_{n-1}$ by binary tournament selection, generate an offspring population $O_n$ and Update $Evals$
Step 2.3: Perform DE/1/rand/bin on $P_{n-1}$ and Update $Evals$
Step 2.4: $U_n = P_{n-1} \cup O_n$ /* $U_n$ is a union population */
Step 2.5: If $n \mod 2 = 0$, then run step 2.5-2.6 of the modified IBEA; otherwise run step 2.5-2.6 of the modified NSGA-II
Step 2.6: Update $A_{n-1}$ using the nearest-neighbor distances and Get $A_n$
Step 2.7: If $Evals < Max_Evals$ then goto Step 2, else goto Step 3

Step 3: Output the results

4.1 Evaluation Criteria

In general, the quality of the non-dominated sets should be assessed in terms of convergence and diversity. Convergence depicts the closeness of the final non-dominated solutions to the true PF, whereas diversity aims at the distribution of the final solutions along the true PF. A number of performance metrics such as hypervolume (HV) [17], generational distance (GD) [18], inverted generational distance (IGD) [19], spacing metric (Spacing) [2, 20] and set coverage (SC) have been proposed. However, none of the performance metrics can reliably evaluate both the performance goals [21]. Thus, we will employ HV, IGD, and Spacing to compare the non-dominated solutions obtained by three algorithms. A brief description of the metrics is given as follows.

(1) HV metric

HV computes the volume in the objective space covered by the non-dominated solutions and a reference point. It was originally proposed by Zitzler and Thiele [22], who called it the size of dominated space. Coello Coello, Van Veldhuizen and Lamont [5] described it as the Lebesgue measure $\Lambda$ of the union of hypercubes enwrapped by a non-dominated set $B$ and a bounding reference point $z_r$ which is dominated by all points.

$$HV(B, z_r) = \Lambda(\bigcup_{z \in B} \{z' | z \prec z' \prec z_r\}), B \subseteq R^n$$

(6)

A major drawback of the metric is the time-consuming process of recursively calculating HV. We will use a fast algorithm named HSO (Hypervolume by Slicing Objectives) [23] for calculating hypervolume. A non-dominated set with a higher hypervolume
value has a higher quality. As elaborated by Knowles [21], the hypervolume indicator is the only unary indicator that is capable of detecting that solution set $A$ with low indicator value is not better than $B$ with higher value.

In the later experiments, we will take the hypervolume difference to a reference set $RS$ as a performance indicator which will be referred to as $HV^*$. The indicator value is defined as follows. Hence, a smaller value of $HV^*$ corresponds to a higher quality.

$$HV^* = HV(RS, Z_r) - HV(A, Z_r)$$

(2) IGD

Suppose $P^*$ and $P$ represent the set of uniformly distributed sampled solutions along the true PF and the set of non-dominated solutions obtained by an algorithm respectively. The mean distance from $P^*$ to $P$ is calculated as follows:

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

(8)

where $d(v, P)$ indicates the minimum Euclidean distance between $v$ and the points in $P$. If $P^*$ can represent the PF very well, the metric can measure both the convergence and the diversity while GD metric can only measure the convergence.

(3) Spacing

The Spacing metric measures how evenly the points in the approximation set are distributed in the objective space. We use an improved version of the metric, where the Euclidean distances between the two extreme ends of non-dominated front and the corresponding ends of the Pareto front are considered. A smaller spacing value indicates the solution set has a better spread. The metric is given by Eq. (9).

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{\lfloor |NDS|/2 \rfloor} [d_i - \bar{d}]}{d_f + d_l + \lfloor |NDS|/2 \rfloor \bar{d}}$$

(9)

where $d_f$ and $d_l$ are the Euclidian distances between the extreme solutions in PF and NDS, $d_i$ denotes the distance between the adjacent solutions via sorting one objective, and $\bar{d}$ denotes the average of $d_i$, ($i \in [1, |NDS| - 1]$). Although the metric is utilized to measure the diversity of non-dominated set extensively, it only works well for two objective problems.

4.2 Parameters Settings

According to our preliminary experiments, the common parameters are set as follows: $NP = 50$, $NO = 50$, $NA = 100$ (150 for DTLZ-series), $F = 0.3$, $CR = 0.8$, $PC = 0.8$, $PM = 0.1$, $eta_cross = 20$, $eta_mut = 20$, RUNs = 30, Max_Evals = 50000 (100000 for DTLZ-series). Here, $NP$ represents the parent population size, $NO$ indicates the offspring population size, $NA$ denotes archive population size, $F$ and $CR$ are the scaling factor and
crossover rate of DE respectively, \( PC \) and \( PM \) represent the probabilities of simulated binary crossover (SBX) and polynomial mutation respectively, \( \eta_{cross} \) and \( \eta_{mut} \) are the distribution indices of SBX and polynomial mutation respectively, \( RUNs \) and \( Max_{Evals} \) indicate the number of runs and evaluations in one run respectively. In addition, a factor \( k \) is set to 0.02 in IBEA and MABNI. In the calculation of performance indicators, we use the objective values normalized by \( \arctan \) function and finally mapped onto \([0, 1]\), and each objective value of a reference point is set to 1.0. Note that \( PC \), \( \eta_{cross} \) and \( PM \) are only used in the primal NSGAII and IBEA instead of the modified versions and MABNI.

### 4.3 Experimental Results and Discussions

In this section, we present the experimental results obtained by MABNI, NSGA-II and IBEA on ZDT-series except ZDT5 and DTLZ-series test instances.

Figs. 3-5 show the best approximation of the true Pareto front found by MABNI for ZDT-series and DTLZ-series in terms of \( HV^* \) metric.

From Fig. 3, it can be seen clearly that the non-dominated fronts converge evenly along the true Pareto fronts of ZDT-series. Seen from Figs. 4 and 5, the non-dominated fronts also approximate to the true Pareto front of DTLZ-series precisely and evenly to a certain extent.

Figs. 6 and 7 show convergence curves of mean \( HV^* \) value based on 30 independent runs of three algorithms aiming at ZDT-series and DTLZ-series respectively.

![Fig. 3. Selected non-dominated fronts obtained by MABNI in contrast with the true Pareto front for ZDT1-4, 6.](image-url)
Fig. 4. Selected non-dominated fronts obtained by MABNI in contrast with the true Pareto front for DTLZ 1-4.

Fig. 5. Selected non-dominated fronts obtained by MABNI in contrast with the true Pareto front for DTLZ 5-7.
From the experimental results in Fig. 6, it is clear that the average $HV^*$ value of MABNI decreases much faster than the other two. It is not surprising that NSGA-II obtains the better results than IBEA because IBEA cannot keep the extreme solutions effectively and thus has a worse coverage of the non-dominated front.

Fig. 6. Convergence curves of MABNI, NSGAII and IBEA for ZDT1-4, 6. X axis represents fitness evaluations and Y axis represents average $HV^*$ values over 30 runs.

Fig. 7. Convergence curves of MABNI, NSGAII and IBEA for DTLZ-series. X axis represents fitness evaluations and Y axis represents average $HV^*$ values over 30 runs.
From the experimental results in Fig. 7, we can see clearly that the indicator value of MABNI also decreases faster than the other two on the whole although IBEA is almost as good as MABNI for DTLZ2 and NSGA-II outperforms MABNI slightly for DTLZ5. It is hard to converge to the true Pareto front for DTLZ1 because the instance is multimodal, e.g., there exist $11^3 - 1$ local optima in DTLZ1 with 3 objectives and 7 decision variables. DTLZ3 with 3 objectives and 12 decision variables is also a complicated problem having $3^{10} - 1$ local optima and one global Pareto front. Although DTLZ6 has a same Pareto front as DTLZ5, it is more difficult to converge to the true Pareto front. For DTLZ7, the Pareto front is divided into 4 discrete regions and thus the difficulty rises. For these more difficult instances, MABNI is markedly superior to the other two algorithms. In contrast with the results of ZDT-series, IBEA has an advantage over NSGA-II for DTLZ-series except DTLZ4 and DTLZ5. It could be speculated that IBEA is more competent for problems having more objectives. Overall, MABNI converges faster than the other two algorithms in terms of HV* indicator.

The box-plots in Figs. 8-10 statistically reflect the differences of the three algorithms by three performance indicators. For the two test sets, the overall predominance is achieved by MABNI. Seen from Fig. 8, all three indicators on ZDT1 and ZDT6 indicate that the results of MABNI are preferable to those of NSGA-II while the results of NSGA-II are superior to those of IBEA. Moreover, it is clear that the indicator values of IBEA are much worse and less stable than the other two. For ZDT2, MABNI outperforms NSGA-II slightly while IBEA has the worst indicator values. For ZDT3 and ZDT4, IBEA also show the poorest performance while MABNI has almost the same median values with NSGA-II on IGD and HV* but some abnormal values far away from the median values exist in the results obtained by NSGA-II. Seen from Figs. 9 and 10, three indicators on DTLZ1 and DTLZ2 are consistent in showing that MABNI outperforms IBEA except for the spacing values of DTLZ2 and IBEA is superior to NSGA-II. For DTLZ7, MABNI is the best while NSGA-II has similar performance to IBEA except for worse abnormal values. For DTLZ4, MABNI merely gets ahead of NSGA-II a little bit according to the median values. However, some abnormal values far away from the median values exist in the results of NSGA-II. Meanwhile, IBEA is inferior to MABNI and NSGA-II. For DTLZ5, according to the values of IGD and Spacing, MABNI predominates over IBEA but has worse HV* values than NSGA-II. For DTLZ6, MABNI is superior to the other two. Simultaneously, IBEA overmatches NSGA-II except for the Spacing values. For DTLZ7, MABNI precedes the other two algorithms. It is surprising that the comparison results of HV* values between NSGA-II and IBEA disagree with the comparison results of other two indicators.

Next, we perform statistical tests to examine the significance of differences in the distribution of HV* values and present some statistical inferences by comparing the test results with a certain level of confidence. Three statistical test methods, Wilcoxon signed-rank test, Mann-Whitney U test and Kruskal-Wallis test, recommended by Knowles [21] for nonparametric inference test are employed here. We state the null hypothesis $H_0$: samples A and B are drawn from the same distribution and the alternate hypothesis $H_1$: sample A comes from a better distribution than sample B. If $H_0$ is rejected and $H_1$ is accepted, it means the algorithm corresponding to sample A is significantly better than the algorithm related to sample B considering a certain instance. The significance level defines the largest acceptable $p$-value. In the following statistical tests, we always con-
sider a confidence level of 95%, i.e., a significance level of 5%.

From Table 1, it can be inferred that MABNI is significantly better than NSGA-II on ZDT-series except ZDT2 and ZDT3 with a significance level of 0.05 in terms of HV* indicator. From Table 2, we can conclude MABNI is significantly better than NSGA-II on DTLZ-series except DTLZ5. In addition, MABNI is significantly better than IBEA on ZDT-series and DTLZ-series except DTLZ4 with respect to the Mann-Whitney U test and DTLZ7.

Fig. 8. The box plots of three indicator values obtained by MABNI, NSGAII and IBEA for ZDT1-4, 6.
Fig. 9. The box plots of three indicator values obtained by MABNI, NSGAII and IBEA for DTLZ1-4.

Fig. 10. The box plots of three indicator values obtained by MABNI, NSGAII and IBEA for DTLZ5-7.
Table 1. Wilcoxon signed-rank (W), Mann-Whitney U (M), and Kruskal-Wallis (K) test results with respect to the HV* values of 30 runs on ZDT-series. The first algorithm of each paired test is from the left column and the second algorithm is from the top row.

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 2. Wilcoxon signed-rank (W), Mann-Whitney U (M), and Kruskal-Wallis (K) test results with respect to the HV* values of 30 runs on DTLZ-series. The first algorithm of each paired test is from the left column and the second algorithm is from the top row.

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MABNI</th>
<th>NSGA-II</th>
<th>IBEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W</td>
<td>M</td>
<td>K</td>
</tr>
<tr>
<td>MABNI</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IBEA</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>
To investigate the measurable impacts on the performance of the algorithm for a single component, we test four reduced algorithms of MABNI. Here we only use ZDT-series instances. The comparative result can be seen from the box plots as shown in Fig. 11. From the graph, we can conclude that the performance of MABNI can be promoted by any one of these measures. However, the measures have different effects on MABNI. Position adjustment of three randomly selected individuals has a minimal impact on overall performance of the solution set. K-nearest neighbors updating rule has a greatest impact on the diversity of the solution set and yet a slight impact on the HV* values of the solution set. Extreme solution preservation strategy in IBEA only shows apparent improvements on the results of ZDT1 and ZDT4. Nevertheless, the strategy is important especially for ZDT4 because ZDT4 is the most difficult problem in ZDT-series due to the presence of 99 local non-dominated fronts [24].

### 5. CONCLUSION

We have presented a multi-algorithm based on NSGA-II and IBEA. It is important to find out the proposed algorithm can achieve a preferable balance between the convergence and diversity of the obtained non-dominated front utilizing the advantages of the
foregoing two algorithms and employing a nearest neighbor truncation operator. In addition, a well-directed differential evolution is utilized to speed the search. Furthermore, the preservation strategies for the extreme objective solutions are employed to generate a wide coverage. The simulation results show the multi-algorithm is superior to NSGA-II and IBEA in approximating the Pareto front and maintaining the diversity of the ap-
proximate solution set.

It is worthy of further studies for self-adaptation of the primary parameters and the self-adaptive operators aiming at decision space. Meanwhile, self-adaptive selection to multiple MOEAs is promising for future research.

REFERENCES


**Datong Xie** (谢大同) received his M.S. degree in Computer Science from China University of Geosciences in 2007. He has been a Lecturer at Department of Information Management Engineering, Fujian Commercial College, Fuzhou, China, since 2008. Currently he is also a Ph.D. Candidate in the State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China. His research interests include evolutionary computation, computer algorithm and software engineering.

**Lixin Ding** (丁立新) received his Ph.D. degree in 1998. He is currently a Professor in the State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China. His main research interests include evolutionary computation, intelligent information processing, and quantum computation.
Yurong Hu (胡玉荣) received her M.S. degree in Computer Science from Yunnan University in 2006. She has been an Assistant Professor of the School of Computer of the Jingchu University of technology since 2008. At present, she is a Ph.D. student in the State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China. Her research interests include rough set theory, data mining and intelligent computing.

Shenwen Wang (汪慎文) received his M.S. degree in Computer Science from the Guizhou University in 2005. He has been a Lecturer of the Department of Computer Science, Shijia-zhuang University of Economic, Shijiazhuang, China, since 2008. He is also a Ph.D. student in the State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China. His research interests include intelligence computation and artificial intelligence.

ChengWang Xie (谢承旺) received his M.S. and Ph.D. degrees in Computer Science from Wuhan University of Technology in 2005 and Wuhan University in 2010, respectively. He has been a Lecturer of School of Software, East China Jiaotong University, Nanchang, China, since 2010. His research interests include evolutionary computation and artificial intelligence.

Lei Jiang (姜磊) received his M.S. and Ph.D. degrees in Computer Science from Southwest Petroleum University in 2005 and Wuhan University in 2012, respectively. He has been a Lecturer of Hunan University of Science and Technology, Xiangtan, China, since 2005. His research interests include pattern recognition, evolutionary computation, machine learning.