Soft Comput (2018) 22:2139–2158
https://doi.org/10.1007/s00500-016-2467-5

Adaptive racing ranking-based immune optimization approach solving multi-objective expected value programming

Kai Yang[1] · Zhuhong Zhang[2] · Jiaxuan Lu[2]

Published online: 23 January 2017

Abstract This work investigates a bio-inspired adaptive sampling immune optimization approach to solve a general kind of nonlinear multi-objective expected value programming without any prior noise distribution. A useful lower bound estimate is first developed to restrict the sample sizes of random variables. Second, an adaptive racing ranking scheme is designed to identify those valuable individuals in the current population, by which high-quality individuals in the process of solution search can acquire large sample sizes and high importance levels. Thereafter, an immune-inspired optimization approach is constructed to seek ε-Pareto optimal solutions, depending on a novel polymerization degree model. Comparative experiments have validated that the proposed approach with high efficiency is a competitive optimizer.

Keywords Immune optimization · Multi-objective expected value programming · Sample bound estimate · Adaptive racing ranking · Computational complexity

1 Introduction

Multi-objective expected value programming (MEVP) is a kind of stochastic programming with comprehensive engineering application background, involving multiple conflicting expected value objective functions. Many real-world problems can be formulated by MEVP models, such as transportation logistics, project management, facility location, healthcare management, portfolio investment. Although some achievements on single-objective expected value programming and multi-objective programming were reported (Jin and Branke 2005), less work on MEVP has appeared in the literature, due to computational complexity and challenging difficulty. In particular, it is almost impossible to find a Pareto optimal solution when noisy environments become severe. The main challenge includes three points: (i) designing adaptive sampling schemes with low computational complexity; (ii) discriminating between individuals through individual dominance; and (iii) studying efficient nondominated sorting strategies in noisy environments. When the conventional static sampling method, i.e., the same fixed sample size for each individual, is adopted to handle noises, there are two usual ways to cope with MEVP models (Gutjahr and Pichler 2016; Marler and Arora 2010; Hughes 2001; El-Wahed and Lee 2006; Liu 2009): (i) translating them into single-objective expected value ones solved by some well-known methods such as the weighted sum approach (Marler and Arora 2010; Hughes 2001) and goal programming approaches (El-Wahed and Lee 2006; Liu 2009) and (ii) replacing them by the related sample average approximation models handled by multi-objective evolutionary algorithms (MEAs) (Gutjahr and Pichler 2016). The first type of approach is simple and easy to implement, but usually can only find a single approximate Pareto optimal solution in a run; the second type of approach can produce multiple approximate solutions, but it easily causes high computational complexity.

When solving static multi-objective programming problems, traditional mathematical methods can only give poor

Communicated by V. Loia.

✉ Zhuhong Zhang
sci.zhzhang@gzu.edu.cn

[1] College of Computer Science & Technology, Guizhou University, Guiyang 550025, China

[2] Department of Big Data Science and Engineering, College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China

performances usually, whereas the MEAs are an alternative and competitive tool and become increasingly popular (Deb 2002; Zitzler and Thiele 1999; Corne et al. 2001; Zhang and Li 2007). Some representative MEAs based on Pareto dominance have been reported in the literature, such as nondominated sorting genetic approach (NSGA2) (Deb 2002), strength Pareto evolutionary approach (SPEAII) (Zitzler and Thiele 1999), Pareto envelope-based selection approach (PESAII) (Corne et al. 2001), multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li 2007). Despite the advantages of structural simplicity and strong availability, they also face some difficulties when directly applied to MEVP problems, due to stochastic factors or noises. Even if so, some reported multi-objective approaches in static environments might solve such kind of problem by attaching static sampling strategies, but it is inevitable that they cause high computational complexity in the process of solution search. Consequently, it is still desired to develop novel multi-objective optimization approaches with efficient sampling strategies.

We in this work suggest a multi-objective expected value immune optimization approach (MEIOA) to solve general nonlinear MEVP problems with unknown noise distributions. This approach, inspired by the principles of clonal selection and immune regulation, is an adaptive optimization tool with the merits of simplicity and practicability. It includes mainly three modules of adaptive racing ranking, immune evolution and archive set update. By comparative experiments, we can draw a conclusion that MEIOA presents significant superiority to the compared methods and also prominent potential to complex nonlinear MEVP problems. It is also pointed out that MEIOA is strongly different from any existing immune optimization techniques. On the one hand, a new racing ranking approach in noisy environments is developed to decide the importance of individual; on the other hand, a useful sample bound estimate is derived to control the dynamic sample size of individual, while a novel polymerization degree model is designed to pick up diverse and high-quality individuals which participate in evolution. In particular, our previous work (Zhang and Tu 2007b) also investigated one multi-objective immune optimization approach (PDMIOA) with static sampling. It is emphasized that such approach is completely different from MEIOA, due to different design inspirations. PDMIOA suppresses noisy influence on the optimized qualities of solutions through individual aging with a maximal lifetime and a reproduction scheme. It includes a modified probabilistic dominance model used for discriminating between low- and high-quality individuals. However, since each individual is attached the same sample size, PDMIOA is difficult in avoiding its high computational complexity.

2 Related work survey

2.1 Noise handling approaches

When random variables are with known distributions, analytic formulas are usually adopted to replace MEVP's subobjective functions. Unfortunately, their noise distributions are generally unknown in engineering applications, and hence studies on model approximation are an alternative way to find Pareto optimal solutions. Monte Carlo simulation as a simple and useful sampling method is usually taken into account, but it is difficult in deciding the sample sizes of such random variables. In order to cope with noises, three kinds of sampling approaches are adopted in general, i.e., static sampling (Zhang and Tu 2007b; Robert and Casella 2013), sample bound estimation (Shapiro et al. 2009; Hoeffding 1963) and adaptive sampling (Cantú-Paz 2004; Higle and Zhao 2004; Chen 2003). Among these approaches, the first is simple and available but usually causes high computational complexity, as each individual is attached to the same large sample size. The second can be used to control the sample sizes of the random variables, but needs many mathematical theory foundations. For example, after exhaustively discussing the relation between single-objective expected value programming and its related sample average approximation model, Shapiro et al. 2009 acquired a valuable lower bound estimate on sample size. They claimed that one such estimate could make any δ-optimal solution of the approximation model approach a corresponding ε-optimal solution of the true problem with high probability. The third can greatly reduce computational cost and is helpful to rapidly finding the optimum, and therefore, it becomes increasingly popular in the context of stochastic optimization. About the research on sampling termination conditions of individuals, Cantú-Paz (2004) studied an adaptive sampling technique to decide when to terminate the process of sampling in terms of an one-sided t-test. He suggested that such sampling method could dynamically determine the sample size of individual and thus might reduce computational cost. Thereafter, Higle and Zhao (2004) examined experimentally the difference between adaptive and non-adaptive sampling schemes by using two approaches of stochastic decomposition (SD) and sample average approximation (SAA). Their results show that there exists little difference between SD and SAA when only taking the quality of solution into account. However, such two methods have different efficiencies, namely SAA causes high computational complexity but SD does not. Additionally, in order to identify whether an individual is superior to another one, some researchers (Chen 2003; Lee et al. 2012) try to investigate how to dynamically allocate a total sample size to different individuals by means of special sampling techniques. This can help excellent individuals obtain large sample sizes, and conversely poor individuals only get few samples. For example, Chen (2003) developed an efficient optimal computing budget allocation scheme to determine the most important individuals in a given population. The allocation scheme can allocate a given sample size to different individuals among which better individuals can gain larger sample sizes, but usually it can be only used in finding one or multiple best solutions. In our previous work (Zhang et al. 2013a; Zhang and Tu 2007a), two adaptive sampling schemes were designed to provide different individuals with different sample sizes, based on the hypothesis test and a sample bound estimate. Such schemes can find those valuable individuals in the process of evolution.

2.2 Multi-objective immune optimization approaches

Since the end of the twentieth century, immune optimization as an important branch of artificial immune systems has become popular, as three famous artificial immune models of negative selection, clonal selection and artificial immune network (Aickelin et al. 2014) were originally proposed based on immunology principles. Relying upon such models, researchers developed a large number of immune approaches to handle various kinds of optimization problems. In particular, some researchers have paid great attention to multi-objective immune optimization, for which the main work concentrates on two aspects: (i) probing into original multi-objective immune approaches (MOIAs) (Coello and Cortés 2005; Gong et al. 2008; Qi et al. 2012; Lin and Chen 2013) and (ii) developing hybrid multi-objective immune techniques (Tan et al. 2008; Aydin et al. 2011; Gong 2013; Qi et al. 2015). Several valuable and original MOIAs are drawing researchers' interests in exploring new immune techniques. For example, Coello and Cortés (2005) suggested a multi-objective clonal selection approach by virtue of the concept of dominance and immune evolution inspirations. They claimed that such approach could ensure those elitist individuals to move toward the true Pareto front. Gong et al. (2008) developed a new nondominated neighbor immune approach (NNIA). In the approach, a novel nondominated neighbor-based selection technique is constructed to pick up diverse individuals, and meanwhile, several immune operators and heuristic search methods are chosen to ensure that those survival individuals transform toward the desired region and elitism. Lin and Chen (2013) proposed a valuable multi-objective micro-population immune optimization approach, relying upon a novel adaptive mutation operator and an efficient immune selection strategy; their adaptive mutation operator can strengthen the exploratory capability of boundary and less-crowded individuals. On the other hand, some researchers devoted themselves to hybrid MOIAs by improving some existing techniques. For instance, Tan et al. (2008) presented a multi-objective evolutionary artificial immune approach, which combined the evolutionary approaches' ability of global search with immune learning. Depending on their new selection strategy and a density preservation mechanism, the presented approach can maintain the balance between exploration and exploitation.

2.3 Intelligent optimization on MEVP

MEVP is a kind of extremely difficult uncertain programming in the field of optimization. Some researchers made great efforts to explore the possible potential of intelligent approaches for this kind of problem. Their work mainly focuses on investigating how to design new dominance methods, noise handling techniques and efficient evolutionary methods. In this research, individual dominance as a metric way of individual importance plays a role in identifying whether an individual is superior to another one under uncertain environments (Hughes 2001; Trautmann et al. 2009; Eskandari and Geiger 2009). Hughes (2001) suggested a classical model of probabilistic dominance to formulate the relation between individuals. Such model has been comprehensively adopted by researchers who investigate multi-objective expected value problems. Particularly, Eskandari and Geiger (2009) introduced the concepts of stochastic dominance and significant dominance to identify competitive solutions, and subsequently, a stochastic fast Pareto genetic approach (FPGA) was designed to deal with stochastic multi-objective optimization problems. Their computational results showed that such approach was superior to NSGA2 (Deb 2002).

Since noise is a crucial factor to interfere the quality of individual evaluation during solution search (Bui 2005; Park and Ryu 2011; Phan and Suzuki 2012; Lee 2010), many researchers have paid great attention to how to explore noise handling techniques. Bui (2005) experimentally studied noisy influence on solution quality by comparing the performances of probabilistic and resampling methods related to NSGA2. They asserted that probabilistic approaches could effectively find Pareto optimal solutions. Park and Ryu (2011) proposed a new optimization technique which mainly included three new operators of accumulative sampling, ranking and selection. Their experiments hinted that such approach performed well over several multi-objective approaches, relying upon multiple kinds of performance criteria. Another focus on solving single or multi-objective expected value programming is to find an appropriate way to accelerate the process of solution search. Although many existing static multi-objective approaches can also handle MEVP problems under the fixed sample sizes of individuals, they are difficult in addressing various kinds of complex MEVP problems with high dimensions or unknown noise distributions, since their solution qualities depend greatly on sufficiently large and fixed sample sizes. For instance, Drugan and Nowe (2013) developed a multi-objective approach to solve the problem of multi-objective multi-armed bandits in the field of reinforcement learning, based on a standard upper confidence bound approach and a Pareto dominance order relationship. Unfortunately, despite being capable of solving the multi-objective problem directly, such approach causes low performance efficiency. Additionally, we have also investigated two immune-inspired multi-objective optimization approaches to solve MEVP (Zhang and Tu 2007b) and multi-objective chance-constrained programming (Zhang et al. 2013b). They involve in several design inspirations such as immune suppression, immune selection, aging and probabilistic dominance, in which a sample allocation technique is designed to assign large sample sizes to high-quality individuals.

3 Problem statement and preliminaries

Consider the following multi-objective expected value programming problem of the form:

$$\text{MEVP} \quad \min_{\boldsymbol{x}\in \boldsymbol{D}} f(\boldsymbol{x}) = E\,[f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}), \ldots, f_q(\boldsymbol{x},\boldsymbol{\xi})],$$

with bounded and closed domain D in R^p, decision vector $\boldsymbol{x}$ in D, where $\boldsymbol{\xi}$ is a r-dimensional random vector with unknown distribution; E $[f_1(\boldsymbol{x},\boldsymbol{\xi}), f_2(\boldsymbol{x},\boldsymbol{\xi}) \ldots, f_q(\boldsymbol{x},\boldsymbol{\xi})]$ denotes the expected value vector function (E $[f_1(\boldsymbol{x},\boldsymbol{\xi})]$, E $[f_2(\boldsymbol{x},\boldsymbol{\xi})], \ldots,$ E $[f_q(\boldsymbol{x},\boldsymbol{\xi})]$); E [.] is the operator of expectation; $f_j(\boldsymbol{x},\boldsymbol{\xi})$ is the jth nonlinear stochastic subobjective function. In order to seek MEVP's solutions, the concept of ε-dominance (Batista et al. 2011) is usually picked up to execute solution comparison. In other words, for two given candidates $\boldsymbol{x}, \boldsymbol{y} \in D$ we say that $\boldsymbol{x}$ ε-dominates $\boldsymbol{y}(\boldsymbol{x} \prec_\varepsilon \boldsymbol{y})$, if

$$E[f_j(\boldsymbol{x},\boldsymbol{\xi})] + \varepsilon \le E[f_j(\boldsymbol{y},\boldsymbol{\xi})], \tag{1}$$

with $1 \le j \le q$, and there exists k satisfying

$$E[f_k(\boldsymbol{x},\boldsymbol{\xi})] + \varepsilon < E[f_k(\boldsymbol{y},\boldsymbol{\xi})]. \tag{2}$$

This way, $\boldsymbol{x}^* \in D$ is called an ε-Pareto optimal solution, if there is no candidate $\boldsymbol{z} \in D$ such that $\boldsymbol{z} \prec_\varepsilon \boldsymbol{x}^*$. In particular, for a given finite population A, $\boldsymbol{x} \in A$ is said to be an ε-nondominated individual, if there is no individual $\boldsymbol{y}$ in A such that $\boldsymbol{y}$ ε-dominates $\boldsymbol{x}$. Similarly, the concept of ε-dominance above may be naturally extended into the version of β-dominance (Trautmann et al. 2009; Eskandari and Geiger 2009) which will be used in identifying competitive individuals. In other words, $\boldsymbol{x}$ β-dominates $\boldsymbol{y}$ with given $\beta = (\beta_1, \beta_2, \ldots, \beta_q)$ and $\beta_i > 0$, if inequalities (1) and (2) are true after replacing ε by β_j and β_k, respectively. Correspondingly, $x \in A$ is said to be a β-nondominated individual, if there is no individual $\boldsymbol{y}$ in A such that $\boldsymbol{y}$ β-dominates $\boldsymbol{x}$.

Generally, when $\boldsymbol{\xi}$ is with known distribution $F_\xi(z)$, each of the above subobjective functions can be replaced by

$$E\,[f_j(\boldsymbol{x},\boldsymbol{\xi})] = \int_{z\in R^r} f_j(\boldsymbol{x},\boldsymbol{z}) dF_\xi(z), \tag{3}$$

and hence the above MEVP can be changed into an analytically deterministic multi-objective programming problem. However, in many practical problems, the noisy information of $\boldsymbol{\xi}$ is unknown and accordingly, the model approximation handling method is an alternative way to solve such kind of problem. Sample average approximation is a simple and popular method used in coping with expected value programming problems with unknown noise distributions. Therefore, we use it to transform the above problem into the following multi-objective sample average approximation model:

$$\text{SAA} \quad \min_{x\in D} \hat{f}(\boldsymbol{x}) = \left(\hat{f}_1(\boldsymbol{x}), \hat{f}_2(\boldsymbol{x}), \ldots, \hat{f}_q(\boldsymbol{x})\right),$$
$$\text{s.t.,} \hat{f}_j(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} f_j(\boldsymbol{x},\boldsymbol{\xi}_i),$$

with $1 \le j \le q; m$ denotes the fixed sampling size; $\boldsymbol{\xi}_i$, $1 \le i \le m$, are the $i.i.d$ samples of $\boldsymbol{\xi}$. $\boldsymbol{x}$ is said to empirically ε-dominate $\boldsymbol{y}$ (simply say $\boldsymbol{x} \prec_{\hat{\varepsilon}} \boldsymbol{y}$) if satisfying $\hat{f}_j(\boldsymbol{x}) + \varepsilon \le \hat{f}_j(\boldsymbol{y})$ with $1 \le j \le q$ and $\hat{f}_k(\boldsymbol{x}) + \varepsilon < \hat{f}_k(\boldsymbol{y})$ for some k. Similar to the version of ε- or β-dominance above, $\boldsymbol{x}$ is called an ε- or β-empirical nondominated individual if there does not exist any individual $\boldsymbol{y}$ in A such that $\boldsymbol{y}$ ε- or β-dominates $\boldsymbol{x}$ empirically.

We easily know that the set of solutions for the above problem SAA can approach that of the MEVP above when m is sufficiently large according to the law of large number. However, in such case any optimization method will cause expensive computational cost. Consequently, we require that different candidates be attached different sample sizes so as to reduce the cost of computation. Hence, the above SAA is transformed into the following multi-objective sample-dependent approximation (SDA) model:

$$\text{SDA} \quad \min_{x\in D} \hat{f}(\boldsymbol{x}) = \left(\hat{f}_1(\boldsymbol{x}), \hat{f}_2(\boldsymbol{x}), \ldots, \hat{f}_q(\boldsymbol{x})\right),$$
$$\text{s.t.,} \hat{f}_j(\boldsymbol{x}) = \frac{1}{m(\boldsymbol{x})}\sum_{k=1}^{m(\boldsymbol{x})} f_j(\boldsymbol{x},\boldsymbol{\xi}_k), 1 \le j \le q,$$

where $m(\boldsymbol{x})$ is the sample size of $\boldsymbol{\xi}$ at the point $\boldsymbol{x}$. We next cite the following conclusions to help us design a novel racing ranking approach to be used in deciding those competitive individuals in a given population.

Theorem 1 (Hoeffding's inequality) (Hoeffding 1963). *Let X be a set, $F(.)$ be a probability distribution function on X; $f_1, \ldots, f_n$ denote the real-valued functions defined on X with $f_j : X \to [a_j, b_j]$ for $j = 1, \ldots, n$, where a_j and b_j are real numbers satisfying $a_j < b_j$. Let $x_1, \ldots, x_n$ be the samples of i.i.d random variables $X_1, X_2, \ldots, X_n$ on X, respectively. Then, the following inequality is true:*

$$\Pr\left(\left|\frac{1}{n}\sum_{j=1}^{n} f_j\left(\boldsymbol{x}_j\right) - \frac{1}{n}\sum_{j=1}^{n}\int_{x\in X} f_j\left(\boldsymbol{x}\right)dF\left(\boldsymbol{x}\right)\right| \geq \varepsilon\right) \leq e^{-\frac{2\varepsilon^2 n^2}{\sum_{j=1}^{n}\left(b_j - a_j\right)^2}}. \tag{4}$$

Corollary 1 (Hoeffding 1963) *If $X_1, X_2, \ldots, X_n$ are i.i.d random variables with $a \leq X_j \leq b$, $1 \leq j \leq n$, and mean μ, then*

$$\left|\bar{X}_n - \mu\right| \leq (b-a)\sqrt{\frac{1}{2n}\ln\left(\frac{2}{\delta}\right)}, \tag{5}$$

with probability at least $1-\delta$, where $\bar{X}_n = \frac{1}{n}\sum_{j=1}^{n}\hat{X}_j$;$\hat{X}_j$ and δ denote the observation of X_j and the significance level, respectively.

4 Lower bound estimate theory

We first investigate two lower bound estimates for the sample sizes of random vector $\boldsymbol{\xi}$, which can reflect the relationship between two individual sets in a finite population A with size N. Then, an adaptive racing ranking approach is designed to find β-empirical nondominated individuals in such population. In order to ensure each individual in A a rational sampling size, we investigate lower bound estimates to control the value of $m(\boldsymbol{x})$ with $\boldsymbol{x} \in A$, based on the sample average approximation model SAA. More precisely, let A_ε and $\hat{A}_\varepsilon$ be two subsets in A composed of ε-nondominated and ε-empirical nondominated individuals, respectively. We next give the following conclusions for which the proofs can be found in "Appendix 1".

Lemma 1 *Let $f_j(\boldsymbol{x}, \boldsymbol{\xi})$ be uniformly bounded with $1 \leq j \leq q$, namely there exists real number a and b such that $\Pr\{a \leq f_j(\boldsymbol{x}, \boldsymbol{\xi}) \leq b\} = 1$ with $\forall \boldsymbol{x} \in D$. Then, for any $\boldsymbol{x} \in A_\varepsilon$ and $\boldsymbol{y} \in \hat{A}_\varepsilon$, the probability, which the event $\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\}$ occurs, satisfies*

$$\Pr\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\} \leq 2e^{-mc\varepsilon^2}, \tag{6}$$

where $c = \frac{1}{2(b-a)^2}$.

Theorem 2 *If $f_j(\boldsymbol{x}, \boldsymbol{\xi})$ be uniformly bounded with $1 \leq j \leq q$, the following formula holds,*

$$\Pr\{A_\varepsilon \subseteq \hat{A}_\varepsilon\} \geq 1 - 2N^2 e^{-mc\varepsilon^2}. \tag{7}$$

Corollary 2 *Under the assumption of Theorem 2, A_ε is a subset of $\hat{A}_\varepsilon$ with probability $1-\delta$, if*

$$m \geq M_1 \equiv \frac{1}{c\varepsilon^2}\ln\frac{2N^2}{\delta}. \tag{8}$$

The above corollary shows that the lower bound estimate M_1 depends on c, δ, ε, and in particular N^2. Once N is large, M_1 is very large, and hence a large sample bound is needed to control the sampling sizes of individuals in population A. If so, such population will cause high computational complexity. Therefore, we improve the above lower bound estimate below.

Theorem 3 *Under the assumption of Lemma 1, the following formula holds,*

$$\Pr\{\hat{A}_\varepsilon \cap A_\varepsilon \neq \emptyset\} \geq 1 - 2Ne^{-mc\varepsilon^2}. \tag{9}$$

Corollary 3 *Under Theorem 3, at least an ε-empirical nondominated individual in A is an ε-nondominated individual with probability $1-\delta$, if*

$$m \geq M_2 \equiv \frac{1}{c\varepsilon^2}\ln\frac{2N}{\delta}. \tag{10}$$

We can easily see that M_2 is smaller than M_1. This helps for reducing the computational cost of deciding empirical objective values for individuals in A. Thereby, M_2 is adopted in the subsequent section to control the sample sizes of individuals.

5 Individual evaluation and similarity measure model

5.1 Individual evaluation

As we mentioned above, a sufficiently large sample size can make the empirical objective value approach the expected objective value at a given individual, but the high computational complexity is inevitable. Fortunately, since those inferior individuals will be gradually eliminated in the process of solution search, it is not necessary to assign large sample sizes to them. Conversely, since those high-quality individuals are required to approach ε-nondominated solutions gradually, their sample sizes should be large. Therefore, in order to spend less computational cost to seek those empirically nondominated individuals in a given population $A = \{x_1, x_2, \ldots, x_N\}$, we develop a new adaptive racing ranking approach (ARRA) to decide the β-empirical nondominated set of one such population, based on the reported Hoeffding's race approach (Even-Dar et al. 2006). The main task of ARRA includes two points: (i) Those superior individuals in A gain large sample sizes and (ii) those ε-empirical nondominated individuals in A will be found gradually by means of β-empirical dominance. More precisely, let s be a sample increment; $r(x)$ denotes the importance level for $\boldsymbol{x} \in A$. ARRA is formulated below:

Step 1: Input the parameters of ε and δ as mentioned above, population A and empty set Q;

Step 2: Define initial sample size m_0 and maximal importance level $R_{\max}$;

Step 3: Set $m \leftarrow m_0, \tau \leftarrow 0, s \leftarrow m_0, \lambda \leftarrow \ln(1+\tau)$, and $Q \leftarrow A$;

Step 4: Evaluate all the individuals in Q with the same sample size m;

Step 5: Calculate the Hoeffding's bound through Corollary 1 given by

$$\beta_j = (b_j - a_j)\sqrt{\frac{1}{2m}\ln\left(\frac{2}{\delta}\right)}, 1 \le j \le q, \tag{11}$$

where b_j and a_j denote the maximum and minimum of $\hat{f}_j(x_i)$ with $x_i \in A$ and $1 \le i \le N$, respectively;

Step 6: Sort Q into P_{nond} and P_d, and set $r(x) = \tau$ with $x \in P_d$, where P_{nond} consists of β-empirical nondominated individuals in Q and P_d is composed of other elements in Q;

Step 7: Update $\tau \leftarrow \tau + 1, \lambda \leftarrow \ln(1+\tau), s \leftarrow \lambda s$, and set $Q \leftarrow \mathrm{P}_{nond;}$

Step 8: Update $m \leftarrow m + s$;

Step 9: Re-evaluate all the individuals in Q with the same sample size m;

Step 10: If $m < \ M_2$ and $\tau < R_{\max}$, go to Step 5; otherwise, set $r(x) = \tau + 1$ with $x \in Q$;

Step 11: Output Q, $\mathrm{A}\backslash Q$ and $r(x)$ with $x \in A$.

In the above approach, Step 6 employs the nondominated sorting (Zheng 2004) to determine the β-nondominated set under sample size m, where the empirical objective values of individuals in Q, instead of their expected objective values, are used to execute comparison between individuals. In particular, each individual is attached an importance level. $R_{\max}$ and M_2 control the sample sizes of individuals. With the increasing sample size of m, β_j will become small, and thus those $\boldsymbol{\beta}$-nondominated individuals will be found gradually. In other words, those valuable individuals in A will be produced. Consequently, ARRA can determine those high-quality individuals in A. Additionally, each individual in such approach is evaluated at most M_2 times. When executing β-nondominated sorting, all individuals in Q are divided into two subpopulations, in which in the worst case the computational complexity is $O(N \log N)$. Therefore, ARRA's complexity is given by $O(N(M_2 + \log N))$.

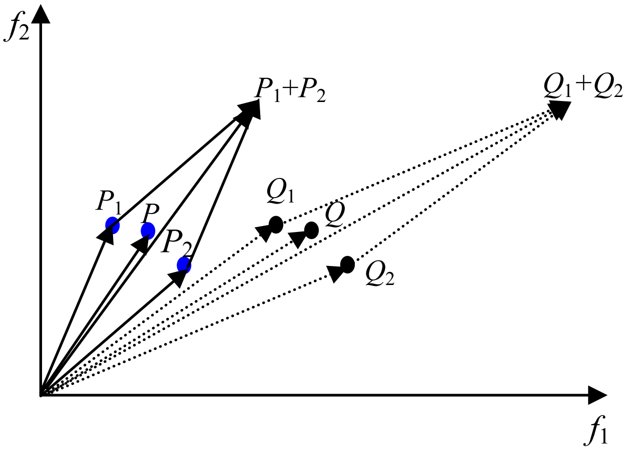

Fig. 1 Schematic illustration on individual's similarity and importance

5.2 Similarity measure model

The well-known crowding distance method (Deb 2002) is usually used to measure the similarity degree of a single individual to other individuals in the objective space. Once an individual has a large crowding distance, its neighborhood includes few individuals. Whereas such method can reflect some similarity features of all the individuals in population A, it cannot present any information on individual quality. We here design a metric to emphasize the similarity and importance for an individual in A. Precisely, let the objective vectors of individuals in A appear in Fig. 1; P_1 and P_2 are the two closest points of P; so are Q_1 and Q_2 for Q. We easily see that P and Q have the same crowding distance but different qualities. Since P_1, P_2 and P are close to the origin, we obtain that $|S(P, P_1 + P_2)| < |S(Q, Q_1 + Q_2)|$, and hence P is better than Q (take the problem of minimization for example), where $S(.,.)$ and $|.|$ represent the inner product of two vectors and the absolute value of real number, respectively. Thus, in order to measure whether an individual has similar individuals in its neighborhood and whether it is of high quality, we give the following polymerization degree model (PDM),

$$\text{PDM}(\boldsymbol{x}) = \frac{d(\boldsymbol{x})}{S^2(\boldsymbol{x}, \boldsymbol{x}_1) + S^2(\boldsymbol{x}, \boldsymbol{x}_2)}, \tag{12}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the two closest points of $\boldsymbol{x}$ in A based on the objective space and $d(\boldsymbol{x})$ presents the crowding distance of $\boldsymbol{x}$. Notice that $S(\boldsymbol{x}, \boldsymbol{y})$ is the inner product of empirical objective vectors of $\boldsymbol{x}$ and $\boldsymbol{y}$. If $\boldsymbol{x}$ approaches the origin and $d(\boldsymbol{x})$ is large, we think preferably that it is a competitive individual. Thus, Eq. (12) can be used to pick up those both high-quality and diverse individuals in A, while prohibiting more extreme individuals from being selected, e.g., those individuals presented on the top left corner or on the lower right corner.

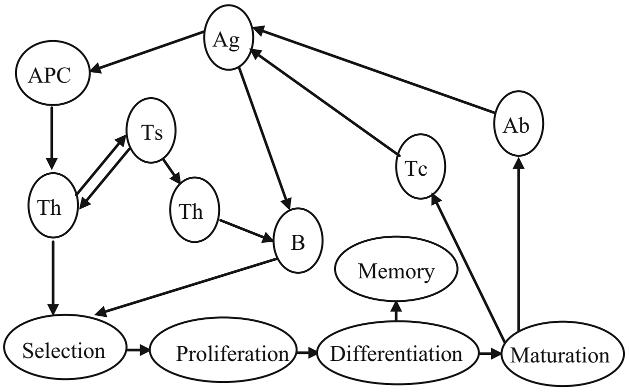

Fig. 2 Artificial immune model

6 Artificial immune model and algorithm formulation

6.1 Artificial immune model

Adaptive immune response is usually triggered by invading pathogens viewed as antigens (Ag), including the humoral immunity and cell-mediated immunity mediated by B cells and T cells (Owen et al. 2013), respectively. T cells can be divided into several types, e.g., Th1, Th2, Ts and Tc, according to their immune functions. Th1 cells can induce the immune system to kill the infected target cells, whereas Th2 cells can assist B cells to produce antibodies in order to eliminate antigens in body fluid. Additionally, Ts and Th2 can regulate humoral immunity and cell-mediated immunity so as to keep the balance of cell populations. When the antigens invade the living body, B cells and T cells will conduct an immune response process of activation, proliferation and differentiation. In such process, antigen's counter-specific B cells and T cells will be selected to multiply for effectors. These cells include cytotoxic T cells (Tc) and antibody (Ab). Thereafter, some antigen-specific cells as memory cells stay at the immune system for a long time, while others take place a change through somatic maturation. Figure 2 presents the relationship between different cells, while emphasizing a simple learning and evolutionary process of immune cells, which gives us sufficient bio-inspirations to study how to design immune optimization approaches solving the MEVP problem in Sect. 3.

6.2 Algorithm formulation

Inspired by Fig. 2 and immune metaphors, this section formulates our new approach (MEIOA) in detail. As associated with the above problem SDA in Sect. 3 and the racing ranking approach ARRA in Sect. 5.1, a real-coded candidate solution is regarded as an antigen-specific B cell, and meanwhile the problem itself is viewed as the antigen; those best solutions found until now correspond to memory cells. Based on the above ARRA and bio-immune inspirations, MEIOA is shown in Fig. 3 above. It divides the current population into superior and inferior subpopulations by ARRA. On the one hand, the superior subpopulation consists of β-empirical nondominated B cells which also update the memory pool so that those elitist B cells can be reserved in the process of evolution. One such subpopulation also transforms its elements toward the desired regions through proliferation and mutation. On the other hand, the inferior subpopulation is composed of β-empirical dominated B cells, in which each B cell makes change only through uniform mutation. In the whole process of solution search, the operator of immune regulation is used to regulate the sizes of the two subpopulations. Once the number of β-empirical nondominated B cells is beyond a limited bound, some of them are regarded as inferior B cells to enter the inferior subpopulation. MEIOA's detailed descriptions are as follows:

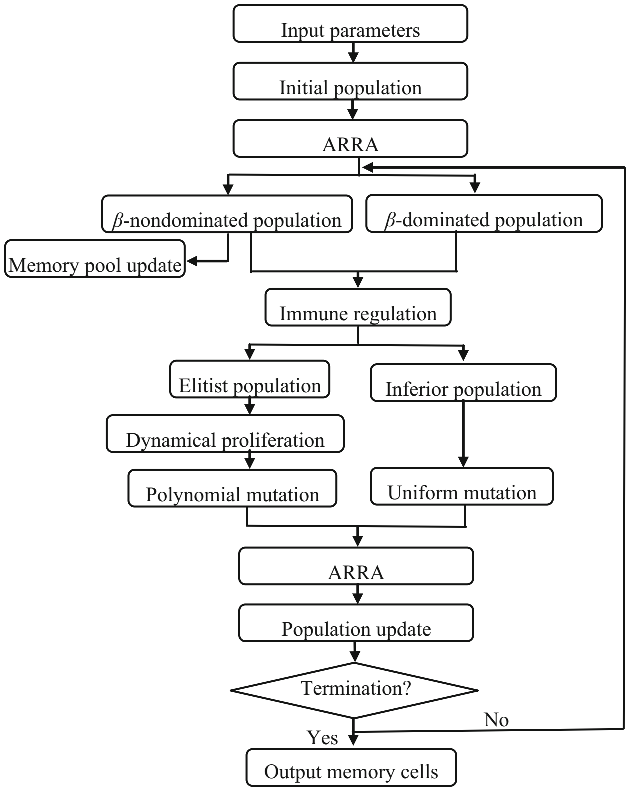

Fig. 3 Flowchart of MEIOA

Step 1: Input parameters: population size N, memory pool size M, initial sample size m_0, tolerance factor ε, significance level δ, high-quality solution ratio $\eta\%$, maximal clonal size $C_{\max}$, mutation weight μ and maximal importance level $R_{\max}$;

Step 2: Set $n \leftarrow 1$. Generate an initial population A_n of N random B cells, and initialize memory pool M_{set};

Step 3: Evaluate all the elements in A_n by ARRA, and acquire the importance levels of B cells, and $r(\boldsymbol{x})$

with $\boldsymbol{x} \in$ A$_n$; divide A$_n$ into two subpopulations B$_n$ and C$_n$, where B$_n$ consists of β-empirical nondominated B cells (in other words, their importance levels are the highest) and C$_n$ is composed of elements in A_n but not in B$_n$;

Step 4: Copy elements in B$_n$ into M$_{set}$, and eliminate those ε-empirical dominated B cells; if $|\mathrm{M}_{set}| > M$, the conventional crowding distance method is used to remove those redundant cells;

Step 5: Calculate PDM values of B cells in A$_n$ through Eq. (12);

Step 6: If B$_n$ includes at least η% of elements in A$_n$, those elements in B$_n$ with small *PDM* values move from B$_n$ to C$_n$; subsequently, those B cells in C$_n$ with large PDM values are kept so that C$_n$ includes only r B cells with $r = N - |\mathrm{B}_n|$;

Step 7: B cell $\boldsymbol{x}$ in B$_n$ proliferates clonal subpopulation $Cl(\boldsymbol{x})$ with size $cl(\boldsymbol{x}) = \min(r(\boldsymbol{x}), C_{\max})$; the clones of all the B cells in B$_n$ constitute clonal population D$_n$;

Step 8: B cell $\boldsymbol{x}$ in D$_n$ changes its genes through polynomial mutation with mutation rate $p_m(\boldsymbol{x}) = \mu + (1-\mu)/(r(\boldsymbol{x}) + r_{\max})$; B cell $\boldsymbol{y}$ in C$_n$ mutates its genes through uniform mutation with mutation rate $p_m(\boldsymbol{x}) = \mu + (1-\mu)/(r(\boldsymbol{x}) + 1)$; all those mutated B cells form E$_n$;

Step 9: Evaluate all elements in E_n by ARRA;

Step 10: Execute comparison between the mutated cells and their parents. For each $\boldsymbol{x}$ in A$_n$, if $\boldsymbol{x}$ is inferior to $\boldsymbol{y}$ by virtue of the version of β-dominance, $\boldsymbol{x}$ is replaced by $\boldsymbol{y}$, by which an updated population A$_{n+1}$ is created, where $\boldsymbol{y}$ denotes the mutated cell of $\boldsymbol{x}$ or the best of the mutated clones of it; if the termination criterion is not satisfied, then set $n \leftarrow n+1$ and go to Step 3; otherwise, output the cells in M$_{set}$.

In the above approach, Step 3 estimates the objective values of B cells and divides A$_n$ into B$_n$ and C$_n$. In Step 6, once the size of B$_n$ is beyond a limited bound, some elements in B$_n$ with small *PDM* values enter C$_n$, due to the requirement of the diversity of population. Such two subpopulations evolve toward different directions by Step 8. All the mutated B cells are evaluated in Step 9 through ARRA.

7 Computational complexity and performance criteria

7.1 Computational complexity

Based on the above approach description, MEIOA's computational complexity is decided by Steps 4, 5, 8 and 9. We acquire the following conclusion.

Theorem 3 *MEIOA's computational complexity in the worst case is*

$$O_c = O((N+M)\log(N+M) + NC_{\max}(M_2 + p + NC_{\max})). \tag{13}$$

Through Eq. (13), MEIOA's complexity depends mainly on N, M, M_2, $C_{\max}$ and p. We know that M_2 helps the ARRA suppress the noisy influence on high-quality individuals; M is used to ensure that the solutions acquired by MEIOA have satisfactory distributions. Consequently, in order to consider MEIOA's efficiency, N and $C_{\max}$ take values as small as possible, e.g., $N = 30$ and $C_{\max} = 2$.

7.2 Performance criteria

We here cite four reported criteria (Gong et al. 2008; Hu 2010) to execute algorithm comparison. Assume that two Algorithms A and B execute only once on some optimization problem, respectively. Correspondingly, they acquire ε-empirical nondominated sets P and Q in sequence.

(A1) Coverage rate (*CR*). This can measure the difference of the optimized qualities for A and B, defined as

$$\mathrm{CR(A,B)}= \frac{|\{\boldsymbol{x}\in Q|\exists \boldsymbol{y}\in P, s.t. \boldsymbol{y}\prec_{\hat{\varepsilon}}\boldsymbol{x}\}|}{|Q|}. \tag{14}$$

Obviously, $0 \le CR(A,B) \le 1$. If $CR(A,B) > CR(B,A)$, A can achieve better solution search than B. If $CR(A,B) = 1$, the solution quality gained by A is absolutely superior to that obtained by B.

(A2) Coverage density (*CD*). This can formulate the distribution performance of solutions in P, given by

$$CD = \frac{1}{|P|-1}\sum_{j=1}^{|P|}(d_j - \bar{d})^2, \tag{15}$$

$$d_j = \min_{j\neq i, 1\le k\le |P|}\left\{\|\boldsymbol{x}_j - \boldsymbol{x}_k\|, \boldsymbol{x}_{\mathbf{j}}, \boldsymbol{x}_k \in P\right\},$$

$$\bar{d} = \frac{1}{|P|}\sum_{j=1}^{|P|} d_j.$$

Equation (15) indicates that *CD* takes values within 0 and 1. If $CD = 0$, the solution set P is totally with uniform distribution, and thus if *CD* is smaller, the distribution of the solutions in P is better.

(A3) Coverage scope (*CS*). This denotes the coverage width of solutions in P, given by

$$CS = \max_{1\le j,k\le |P|}\{\|\boldsymbol{x}_j - \boldsymbol{x}_k\|, \boldsymbol{x}_j, \boldsymbol{x}_k \in P\}. \tag{16}$$

(A4) Convergence metric (*CM*). This is utilized to measure how close the solution set P approaches P^*, where

P^* is the theoretical Pareto optimal solution set. CM is defined by

$$CM\left(P^*, P\right) = \frac{1}{|P^*|} \sum_{x \in P^*} dist\,(x, P), \tag{17}$$

where $dist(x, P)$ represents the Hausdorff distance from x to P. It is clear that A is convergent if $CM(P^*, P) = 0$. Therefore, the smaller the $CM(P^*, P)$, the better the convergence.

8 Experimental study

All experiments are executed on a Windows XP system with CPU/3.50 GHz and 2.98 GB RAM by means of Visual C++ platform. In order to analyze MEIOA's intrinsic characteristics, two evolutionary approaches [HEA (Zhang et al. 2015) and FPGA (Eskandari and Geiger 2009)] and three immune optimization approaches [PDMIOA (Zhang and Tu 2007b), NNIA (Gong et al. 2008) and MAMMOIA (Hu 2010)] are selected to participate in comparison by means of two benchmark test suites and one engineering example. It is pointed out that, although NNIA and MAMMOIA are not specially designed to solve stochastic programming problems, they are still efficient and competitive in stochastic environments. On the other hand, FPGA, HEA and PDMIOA with static sampling strategies are developed to deal with MEVP problems, and their individuals in this experiment take the same reported sampling size 300. After manual experimental tuning, MEIOA takes population size 30 but 100 for the compared approaches. In order to be fair, each approach is required to stop its solution search procedure when the total of evaluations in a run is beyond T, while executing 100 times on each test problem. It is emphasized that T is defined according to different kinds of problems below. Other parameter settings of the compared approaches are the same as those in their corresponding literature. In particular, in order to avoid noisy influence on solution comparison between the above approaches, all the best solutions found by them are required to re-evaluate 10^5 times, since their empirical objective values are used to execute statistical analysis.

8.1 MEIOA's parameter settings

In MEIOA, ε and δ do not depend on any prior information on problems solving. The parameter ε as a factor of tolerance takes values within 0.01 and 0.1 usually. δ used in deciding M_2 and the Hoeffding's bound is the significance level presented in Corollary 2 above, changing within 0.05 and 0.2. In addition, M, $C_{\max}$ and $R_{\max}$ influence the optimized quality and efficiency. In order to acquire a set of solutions with uniform distribution and wide coverage scope, M takes values within 60 and 100. $C_{\max}$ is an important parameter to strengthen the ability of population exploitation. We think that it is rational to define $C_{\max}$ within 1 and 3. η helps for improving the quality of solution search, taking values within 60 and 80 generally. $R_{\max}$ is used to decide the maximal sorting level of individuals in the process of evolution. If $R_{\max}$ is large, ARRA spends much more runtime to decide competitive individuals, and conversely those excellent individuals are identified difficultly. We designate it as 5. Further, μ as a minimal mutation rate influences the solution quality to be acquired; it takes values within 0 and 0.3. m_0 is an initial sample size of individual included in ARRA, taking values within 10 and 60. After experimental trials, we define $M = 100$, $m_0 = 30$, $\eta = 80$, $C_{\max} = 2$, $\varepsilon = 0.05$, $\delta = 0.1$ and $\mu = 0.1$.

8.2 Theoretical test example design

In this experiment, nine static multi-objective benchmark problems, proposed originally by Van Veldhuizen and Deb et al. (1999, 2000), i.e., five test problems [MOP1, MOP2, MOP3, MOP4 and MOP6 (Van Veldhuizen 1999)] and four test problems [ZDT1, ZDT2, ZDT3 and ZDT4 (Zitzler et al. 2000)], are picked up to transform into MEVP problems (see "Appendices 2 and 3") by, respectively, adding a Gaussian noise with the standard normal distribution to their respective subobjective functions. Such MEVP problems are still named by the same versions as their original ones and divided into two sections. The first section, namely the first test suite, includes the above former five test problems, while the other test problems, i.e., the latter four test problems, constitute the second test suite.

It is worth pointing out that in the first test suite, besides MOP1 and MOP3 involving in at most two decision variables, other three examples are all three-dimensional problems. Whereas MOP1 is easily solved in that its subobjective functions are quadratic, other test problems are handled difficultly because of multimodality. These examples are chosen to check if the above approaches can all effectively solve low-dimensional uni- and multi-modal expected value problems. Additionally, the test problems in the second test suite are still difficult, since their Pareto fronts are composed of disjoint curve segments. In order that the approaches can present clear differences for one such test suite, our experiment is executed on these problems according to different dimensional settings, i.e., $n = 30$ and 80.

8.3 Experimental analysis on the first test suite

All the above approaches take the same maximal evaluation number 2×10^5, among which each of the compared approaches directly executes 100 single executions on the sample average approximation model (SAA) of each test example, and so does MEIOA on the sample-dependent

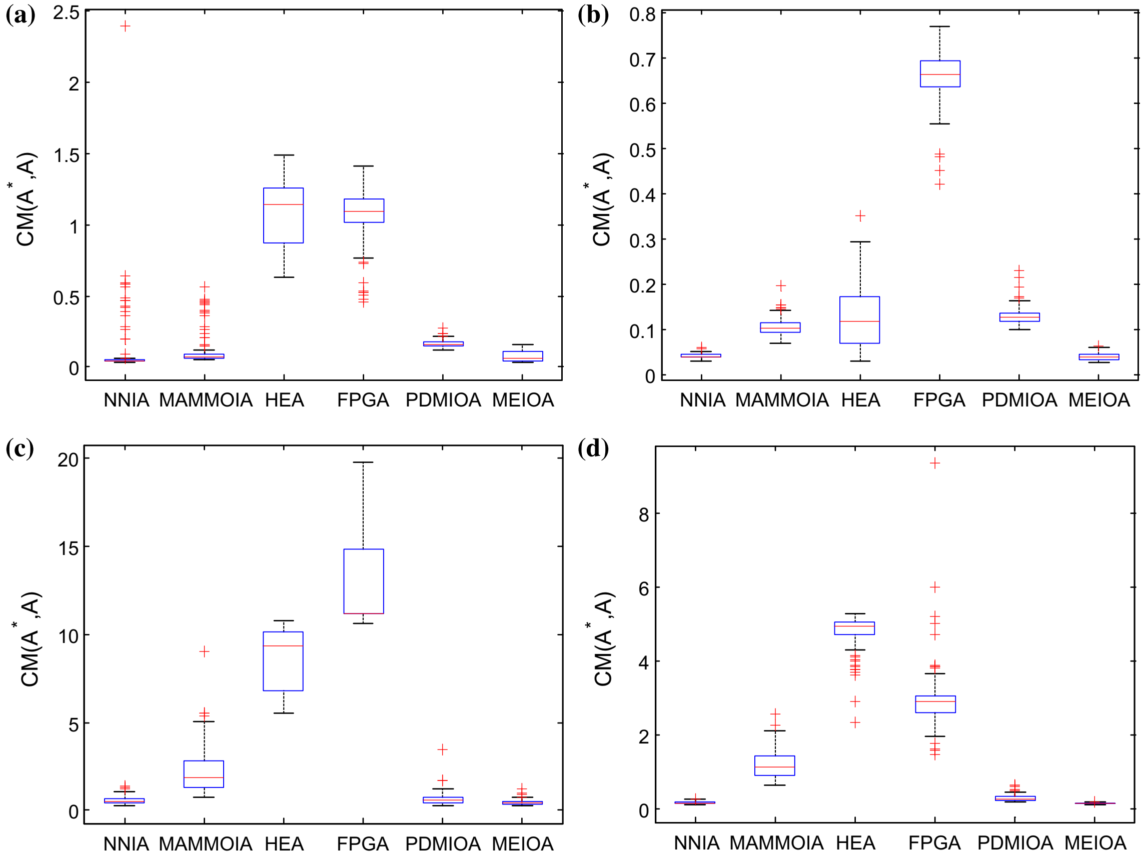

Fig. 4 Comparison of results on *CM* based on box plots for problems MOP1 to MOP4. **a** MOP1, **b** MOP2, **c** MOP3, **d** MOP4

approximation model (SDA) of it. Based on the performance criteria in Sect. 7.2, their statistical results are given in Table 1. Additionally, we only take four examples, MOP1 to MOP4, for example, to display the corresponding box plots (see Fig. 4) on *CM*, because of the limit of space.

Table 1 presents the performance characteristics of the above four approaches and also their main differences with respect to evolution, noise handling, diversity, search stability, solution distribution, execution efficiency and so forth. By the statistical results on *ACR* in columns 3–8 and Fig. 4, MEIOA can clearly acquire the best solution qualities for the above examples, since there are always the following coverage relationships for any given test example:

ACR (MEIOA, PDMIOA) $\geq$ ACR (PDMIOA, MEIOA),
ACR (MEIOA, FPGA) $\geq$ ACR (FPGA, MEIOA),
ACR (MEIOA, HEA) $\geq$ ACR (HEA, MEIOA),
ACR (MEIOA, MAMMOIA) $\geq$ ACR (MAMMOIA, MEIOA),
and ACR (MEIOA, NNIA) $\geq$ ACR (NNIA, MEIOA).

These inequalities illustrate that MEIOA, relying upon the proposed ARRA, can work well over the compared methods and effectively overcome noise interference. For instance, for MOP2 the nondominated sets found by MEIOA cover averagely in order 41, 50, 43, 6 and 46% of those obtained by NNIA, MAMMOIA, HEA, FPGA and PDMIOA; conversely, the latter five approaches can only gain smaller *ACR* values in comparison with MEIOA. On the other hand, the values on *CR* and Fig. 4 can ensure that PDMIOA behaves well over the other compared approaches, since it only performs worse for MOP2 and MOP6. We also notice that the compared approaches except PDMIOA have similar solution qualities. As associated with the values on *CS*, we see that the two evolutionary approaches FPGA and in particular HEA indeed get easily into local search, whereas the immune optimization approaches are opposite. In particular, although NNIA presents the phenomenon of search instability, it is still competitive for the above examples. Totally, with respect to solution quality we claim that MEIOA is best, PDMIOA is secondary, and NNIA and MAMMOIA are better than FPGA and HEA.

Through the results on *CD* given in columns 9 to 12 in Table 1, we note that the best solutions gained by the above six approaches present different distribution characteristics. For each test example, the solutions obtained by MEIOA can almost uniformly and stably distribute over the theoretical Pareto front, whereas all the compared approaches except NNIA cause poor solution distributions for at least one test

Table 1 Statistical comparison of ε-nondominated sets acquired for the first test suite

Prob.	Algor.	*ACR* (%)						*CD*		*CS*		*AT* (s)
		NNIA	MAMMOIA	HEA	FPGA	PDMIOA	MEIOA	Mean	SD	Mean	SD	
MOP1	NNIA	0	3	76	0	3	3	0.07	0.02	9.32	3.97	14.4
	MAMMOIA	14	0	77	0	2	0	0.18	0.08	9.23	5.95	12.6
	HEA	1	0	0	0	0	0	0.05	$<10^{-2}$	0.10	0.16	15.1
	FPGA	2	1	75	0	0	0	0.27	0.51	2.66	2.07	16.6
	PDMIOA	12	2	77	0	0	1	0.04	$<10^{-2}$	7.62	0.65	15.5
	MEIOA	**17**	**3**	**78**	**0**	**3**	**0**	**0.03**	**0.01**	**9.71**	**0.18**	**3.8**
MOP2	NNIA	0	49	7	11	56	39	$<10^{-2}$	$<10^{-2}$	1.92	$\mathbf{<10^{-2}}$	14.3
	MAMMOIA	16	0	9	7	31	18	$<10^{-2}$	$<10^{-2}$	1.92	$<10^{-2}$	14.4
	HEA	46	55	0	5	44	37	$<10^{-2}$	$<10^{-2}$	1.39	0.09	15.2
	FPGA	3	2	0	0	7	4	$<10^{-2}$	$<10^{-2}$	0.02	$<10^{-2}$	16.4
	PDMIOA	21	56	5	1	0	16	0.02	$<10^{-2}$	1.80	0.01	13.9
	MEIOA	**41**	**50**	**43**	**6**	**46**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.96**	$\mathbf{<10^{-2}}$	**5.2**
MOP3	NNIA	0	56	4	6	6	6	**0.20**	0.04	44.94	11.4	14.2
	MAMMOIA	48	0	4	1	2	3	3.75	28.53	43.47	13.6	14.0
	HEA	36	3	0	0	0	3	8.84	163.43	10.54	79.7	14.8
	FPGA	23	2	3	0	0	1	0.30	0.06	1.28	22.1	16.6
	PDMIOA	74	14	8	8	0	7	2.45	100.04	40.96	5.67	15.9
	MEIOA	**78**	**26**	**7**	**7**	**7**	**0**	0.23	**0.04**	**48.34**	**4.00**	**8.2**
MOP4	NNIA	0	47	2	16	23	20	0.32	0.06	16.60	3.53	15.2
	MAMMOIA	25	0	4	2	2	4	0.93	0.48	15.59	5.41	15.0
	HEA	20	50	0	0	7	6	0.31	0.63	1.33	0.60	16.2
	FPGA	15	2	0	0	2	2	1.93	16.4	1.89	3.14	17.6
	PDMIOA	69	28	7	20	0	16	0.37	0.21	17.01	**0.06**	15.1
	MEIOA	**64**	**29**	**8**	**18**	**17**	**0**	**0.25**	**0.06**	**17.50**	2.01	**7.9**
MOP6	NNIA	0	12	14	6	34	7	$<10^{-2}$	$<10^{-2}$	1.65	0.54	14.6
	MAMMOIA	26	0	9	5	8	8	0.45	0.31	1.90	3.22	14.9
	HEA	75	5	0	13	36	9	0.58	0.87	1.50	2.06	15.9
	FPGA	38	10	9	0	29	6	0.10	$<10^{-2}$	1.20	0.10	16.9
	PDMIOA	14	57	8	5	0	12	0.13	0.06	1.06	1.14	14.7
	MEIOA	**31**	**55**	**31**	**25**	**49**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**2.30**	**0.04**	**5.3**

$ACR(A, B)$ denotes the average of 100 coverage rates (*CR*) by Eq. (14) for Algorithm A versus Algorithm B; the symbols of *CD* and *CS* are mentioned in Sect. 7.2; AT stands for the average runtime spent by a given algorithm

Each bold value denotes the best of results acquired by the approaches for a given performance index

problem. We also observe that those best solutions acquired by NNIA are with relatively uniform and stable distributions by comparison against the other compared approaches. Additionally, with respect to solution coverage scope, the values on *CS* hint that HEA and in particular FPGA can only find extremely crowding solutions with narrow coverage scopes and thus get easily into local search. The four immune optimization approaches, however, can make their individuals evolve along multiple directions and thus present their strong population diversity. With respect to performance efficiency, the values on AT illustrate that, although the above six approaches are with the same termination criterion when solving the same test example, their performance efficiencies are different. The compared approaches spend twice runtime to solve each test problem in comparison with MEIOA. We also know that these compared approaches have similar performance efficiencies, and relatively PDMIOA is more efficient.

As a whole, all the above comments can draw the conclusions: (i) When solving the above first test suite, MEIOA can averagely perform well with the aspects of solution quality, solution distribution and coverage scope as well as performance efficiency, (ii) PDMIOA behaves well over the other compared approaches, while NNIA is also a still competitive

Table 2 Statistical comparison of ε-nondominated sets found for the second test suite with dimension 30

Prob.	Algor.	*ACR* (%)						*CD*		*CS*		*AT* (s)
		NNIA	MAMMOIA	HEA	FPGA	PDMIOA	MEIOA	Mean	SD	Mean	SD	
ZDT1	NNIA	0	13	0	0	22	0	0.51	0.40	1.12	1.17	15.8
	MAMMOIA	9	0	0	0	14	0	0.30	0.27	1.43	22.92	28.5
	HEA	13	10	0	0	25	0	$<10^{-2}$	$<10^{-2}$	0.02	0.01	15.8
	FPGA	70	38	0	0	24	9	$<10^{-2}$	$<10^{-2}$	1.23	0.28	17.2
	PDMIOA	72	70	0	20	0	17	0.03	$<10^{-2}$	1.62	1.07	14.6
	MEIOA	**96**	**79**	**0**	**31**	**30**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.73**	**0.02**	**6.3**
ZDT2	NNIA	0	10	0	0	20	0	0.56	0.53	1.34	0.63	17.4
	MAMMOIA	10	0	0	2	16	3	0.33	0.36	1.65	6.88	32.3
	HEA	22	17	0	79	30	6	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	17.4
	FPGA	18	10	0	0	28	2	0.34	0.61	1.54	1.16	18.8
	PDMIOA	65	38	3	9	0	22	0.01	$<10^{-2}$	1.59	0.87	16.1
	MEIOA	**89**	**20**	**8**	**4**	**30**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.81**	**0.02**	**7.0**
ZDT3	NNIA	0	18	0	1	27	4	0.75	0.38	1.83	0.70	15.2
	MAMMOIA	32	0	0	2	27	3	0.61	0.62	1.68	2.45	27.6
	HEA	16	11	0	16	30	2	$<10^{-2}$	$<10^{-2}$	0.03	0.03	15.3
	FPGA	23	18	0	0	29	10	0.22	0.47	1.46	1.60	16.4
	PDMIOA	56	51	1	18	0	29	0.02	$<10^{-2}$	1.90	1.64	14.2
	MEIOA	**40**	**38**	**5**	**11**	**42**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**2.08**	**0.31**	**5.1**
ZDT4	NNIA	0	0	0	0	0	0	1.38	9.60	9.56	**4.36**	163.4
	MAMMOIA	22	0	6	1	13	2	55.38	706.67	**50.96**	36.02	290.7
	HEA	89	17	0	2	22	3	29.24	242.37	20.00	62.81	162.9
	FPGA	86	83	40	0	82	17	13.49	51.40	9.58	15.40	174.7
	PDMIOA	54	50	17	2	0	4	19.09	241.78	14.06	32.76	206.4
	MEIOA	**87**	**79**	**25**	**25**	**82**	**0**	**0.75**	**8.60**	8.45	4.73	**92.5**

Each bold value denotes the best of results acquired by the approaches for a given performance index

approach, and (iii) compared by MEIOA, each of the compared approaches spends much more than runtime to solve each test problem above.

8.4 Experimental analysis on the second test suite

Let the test problems in such test suite share the same dimension 30 or 80. This is done to examine sufficiently whether the above experimental conclusions are true when some hard problems with higher dimensions are taken into account. After manual trials, in the case of $n = 30$ we define the total of evaluations for each approach as 2×10^5 for problems ZDT1 to ZDT3 but 2×10^6 for ZDT4 because of difficulty. When $n = 80$, however, we set $T = 2 \times 10^6$ for the former three problems, but $T = 2 \times 10^7$ for the latter one problem. Like the above experiment, each compared approach directly runs 100 times on the sample average approximation model (SAA) of each of those MEVP problems, and MEIOA executes 100 times on the sample-dependent approximation model (SDA) of it.

Case 1: 30-dimensional experimental analysis

The results acquired by the above approaches are listed in Table 2, and correspondingly, their box plots on convergence metric *CM* are displayed in Fig. 5.

The values on *ACR* in Table 2 illustrate that, besides the inequalities in Sect. 8.3 still being true, the solutions obtained by MEIOA for each test problem cover greatly those obtained by other approaches. For example, when solving the hard ZDT4, we can clearly see that the solutions found by MEIOA dominate averagely in order 87, 79, 25, 25 and 82% of those acquired by NNIA, MAMMOIA, HEA, FPGA and PDMIOA; conversely, these latter five approaches can only gain smaller *ACR* values by comparison against MEIOA. This indicates that ARRA presented in MEIOA can effectively distinguish between individuals in noisy environments. Additionally, FPGA behaves well over NNIA, MAMMOIA and HEA, while PDMIOA and HEA are superior to NNIA and MAMMOIA. Whereas the compared approaches can acquire some nondominated solutions with relatively stable distributions for problems ZDT1 to ZDT3, they fail to solve ZDT4. In particular, it seems to be true that HEA is better

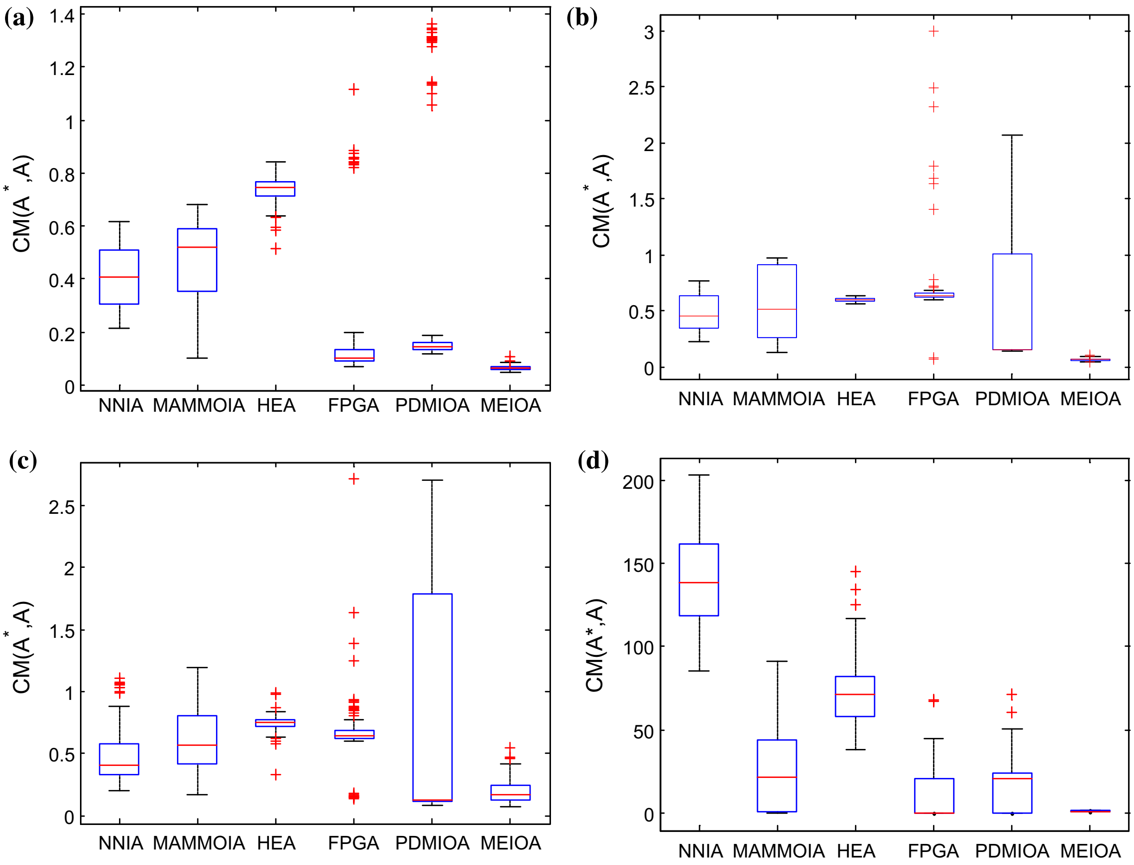

Fig. 5 Comparison of convergence based on box plots for the second test suite with dimension 30. **a** ZDT1, **b** ZDT2, **c** ZDT3, **d** ZDT4

than either NNIA or MAMMOIA. In fact, the values on *CS* hint that it is of poor population diversity and gets easily into local search. In addition, although NNIA is not suitable for ZDT4, we think that it is a still competitive optimizer.

The statistical results on Mean in *CD* and*CS* listed in columns 9 and 12 in Table 2 follow that the solutions acquired by MEIOA are with almost uniform distributions and wide coverage scopes; the values on SD demonstrate sufficiently that such approach can perform stable search for the above problems. Other approaches, however, expose some weak search performances when solving ZDT4. Namely, whereas they can acquire some nondominated solutions with wide coverage scopes in comparison with MEIOA, their values on Mean and SD clearly show that their search performances are extremely instable, which can also be found in Fig. 5. In addition, the values on AT, listed in Table 2 derive that, when solving each test problem, the compared approaches spend twice runtime in comparison with MEIOA; MAMMOIA needs the most average runtime; the other compared approaches have similar performance efficiencies.

Graphically, Fig. 5 displays some important and different properties for the approaches. When solving each problem above, those solutions obtained by MEIOA are very close to the Pareto optimal front, as its box plot is very narrow and near the horizontal line. Thus, MEIOA can present the best effects for all the above test problems, whereas other approaches are opposite. Most notably, by the experimental results in Sect. 8.3 we know that PDMIOA is globally superior to the other compared approaches. This, however, is not true in this experiment.

Case 2: 80-dimensional experimental analysis

The statistical results are listed in Table 3, and meanwhile, the corresponding box plots are shown in Fig. 6.

The results in Table 3 and Fig. 6 present some distinct differences for the above approaches and expose some prominent drawbacks for the compared approaches. MEIOA can still perform well over other approaches with the aspects of solution quality, distribution, efficiency and so on. Although the compared approaches can exhibit their stable search performances for problems ZDT1 to ZDT3, they cause poor solution search. Particularly, HEA still gets easily into local search. Relatively, FPGA is more suitable for the above high-dimensional problems in comparison with the other compared approaches. NNIA and MAMMOIA are superior to PDMIOA and HEA. On the other hand, the values on AT presented in Table 3 illustrate that the same conclu-

Table 3 Statistical comparison of ε-nondominated sets found for the second test suite with dimension 80

Prob.	Algor.	*ACR* (%)						*CD*		*CS*		*AT* (s)
		NNIA	MAMMOIA	HEA	FPGA	PDMIOA	MEIOA	Mean	SD	Mean	SD	
ZDT1	NNIA	0	14	0	0	16	0	0.18	0.26	1.53	3.59	187.8
	MAMMOIA	10	0	0	0	15	0	0.26	0.29	1.21	23.32	378.2
	HEA	6	12	0	2	95	0	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	170.6
	FPGA	67	29	0	0	83	10	0.01	$<10^{-2}$	1.51	63.07	182.8
	PDMIOA	0	0	0	0	0	8	0.46	0.85	0.72	2.05	221.7
	MEIOA	**86**	**44**	**0**	**28**	**60**	**0**	**$<10^{-2}$**	**$<10^{-2}$**	**1.68**	**1.02**	**60.0**
ZDT2	NNIA	0	8	0	0	6	0	0.09	0.10	1.29	5.22	186.6
	MAMMOIA	7	0	0	0	6	1	0.20	0.21	1.36	12.46	398.0
	HEA	8	8	0	51	96	0	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	167.8
	FPGA	11	7	0	0	89	1	0.46	9.27	0.26	1.04	177.9
	PDMIOA	0	0	0	0	0	2	0.80	0.70	0.96	1.28	209.0
	MEIOA	**83**	**20**	**10**	**2**	**42**	**0**	**$<10^{-2}$**	**$<10^{-2}$**	**1.50**	**0.73**	**57.7**
ZDT3	NNIA	0	17	0	0	20	0	0.26	0.46	1.28	1.36	188.7
	MAMMOIA	42	0	0	0	26	3	0.49	0.35	0.88	1.07	380.3
	HEA	5	13	0	7	56	1	$<10^{-2}$	$<10^{-2}$	0.02	0.02	170.9
	FPGA	14	21	0	0	56	6	1.09	3.15	0.50	3.05	181.1
	PDMIOA	4	4	0	0	0	0	0.15	0.15	0.37	**0.61**	221.9
	MEIOA	**28**	**35**	**7**	**9**	**63**	**0**	**$<10^{-2}$**	**$<10^{-2}$**	**1.38**	1.52	**59.5**
ZDT4	NNIA	0	72	0	7	29	0	32.81	21.04	73.72	5.82	1800.7
	MAMMOIA	16	0	26	2	12	0	92.56	322.2	10.15	17.70	3857.4
	HEA	38	1	0	0	18	0	71.21	154.3	24.94	27.15	1755.3
	FPGA	74	53	8	0	68	0	1.62	9.06	4.06	3.60	1903.1
	PDMIOA	47	47	14	1	0	0	100.6	198.5	**122.4**	198.7	2284.5
	MEIOA	**98**	**84**	**72**	**91**	**92**	**0**	**0.27**	**0.82**	5.95	**2.87**	**1080.9**

Each bold value denotes the best of results acquired by the approaches for a given performance index

sion on performance efficiency as that in Sect. 8.3 is also true. Consequently, in comparison with the above experimental results in Table 2, we can assert that MEIOA is effective and efficient for the second test suite and FPGA is secondary. The other compared approaches cannot present stable search performances, when the dimensions make a change.

Summarily, when solving the above two test suites, the six approaches exhibit their respective performance characteristics. For the simple uni-modal MEVP problem (MOP1), whereas they can all find some approximate Pareto optimal solutions in a single run, their nondominated sets present different distribution characteristics. MEIOA can effectively solve all the above problems with high efficiency, but other approaches are difficult. Relatively, PDMIOA can perform well over the other compared approaches for the first test suite but fails to solve the second test suite. FPGA can work well over the other compared approaches for the second test suite but cannot do so for the first test suite; HEA can only find local Pareto optimal solutions.

8.5 Experimental analysis on significant difference

We next detect whether MEIOA shows a significant improvement over the above five compared approaches for the above two test suites. By means of the average of convergence metric values acquired by each approach for the thirteen test problems, i.e., MOP1 to MOP4, MOP6, and ZDT1 to ZDT4 with dimensions 30 and 80, we obtain the statistical values of the Friedman rank and Friedman aligned rank tests given in Table 4. As the table formulates, with respect to solution quality MEIOA shows a significant improvement over the compared approaches with significance level $\alpha = 5\%$; therefore, we can reject the null hypothesis of error convergence metric for MEIOA versus the other approaches. Table 4 also highlights that MEIOA is the most effective algorithm with a rank of 1.00 and 20.77 for the Friedman and Friedman aligned tests, respectively. In addition, since the statistical values 35.42 and 30.63 are larger than 5.99 ($X^2_{1-\alpha/2} = 5.99$), the above six approaches have significant difference with respect to solution quality.

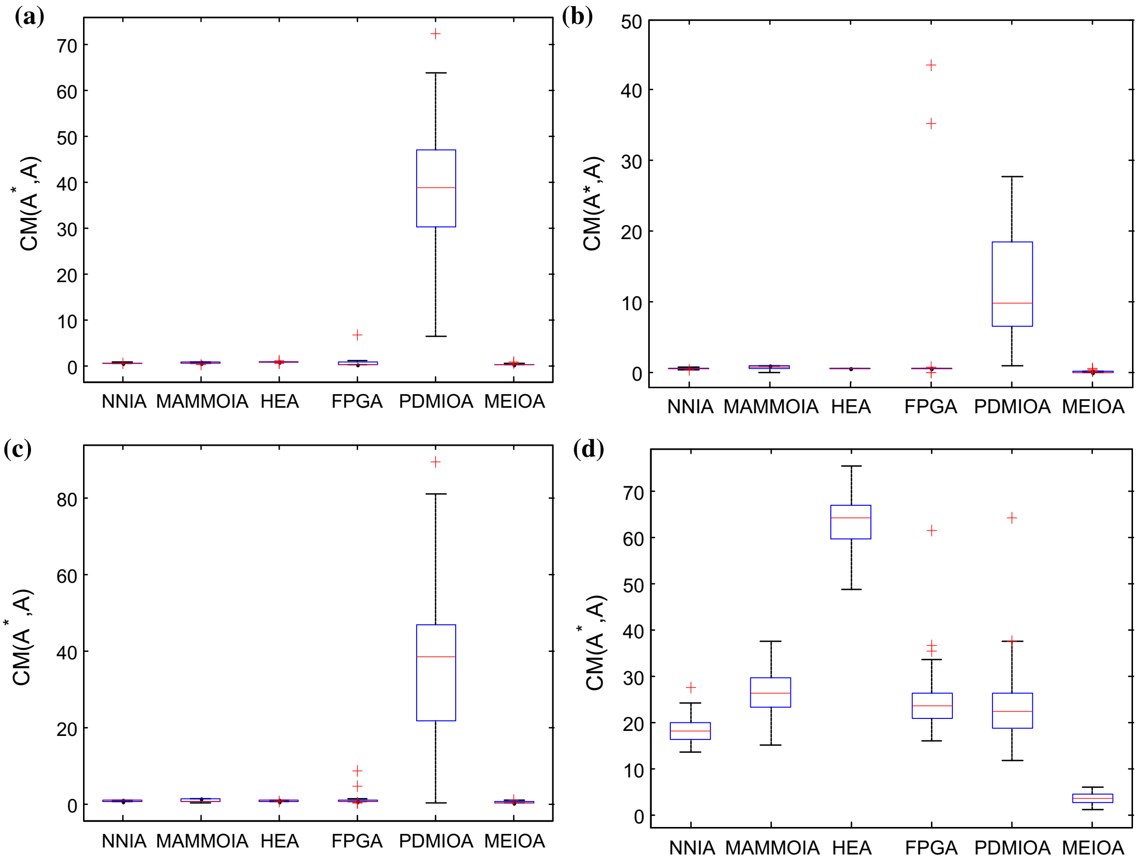

Fig. 6 Comparison of convergence based on box plots for the second test suite with dimension 80. **a** ZDT1, **b** ZDT2, **c** ZDT3, **d** ZDT4

Table 4 Statistical comparison of *CM* based on the Friedman rank and Friedman aligned rank tests

	NNIA	MAMMOIA	HEA	FPGA	PDMIOA	MEIOA	Statistic
Friedman rank	2.92	3.92	4.85	4.38	3.92	1.00	35.42
Friedman aligned rank	33.54	42.08	49.00	48.23	43.38	20.77	30.63

8.6 Sensitivity analysis

We conduct the current experiment to check if MEIOA's solution quality is sensitive to the settings of parameters. Here, only two parameters of m_0 and μ are used to analyze MEIOA's sensitivity. m_0 as an initial sample size of individual included in ARRA influences the efficiency of solution search, while μ as a crucial parameter can strengthen the diversity of population and accelerate population evolution. We define (m_0, μ) as $(10k, 0.1(k-1))$ or $(10(7-k), 0.1(k-1))$ with $1 \leq k \leq 6$. Other settings of parameters are the same as those in Sect. 8.1. After executing 100 runs on MOP1 and MOP3 as well as ZDT1 and ZDT3 with dimension 30, MEIOA acquires the following statistical results on *CM* in Table 5.

Table 5 illustrates that the settings of the two parameters influence MEIOA's solution quality. When m_0 is directly proportional to μ with $m_0 > 10$ and $\mu > 0$, MEIOA are not totally sensitive to the parameter settings in that its solution quality is not influenced seriously. This phenomenon also takes place in the case where m_0 is conversely proportional to μ with $m_0 < 60$ and $\mu > 0$. We also notice that, when μ is set as 0, MEIOA's solution quality is poor, due to large values on *ACM* for MOP1 and MOP3. Summarily, MEIOA is not sensitive to the settings of m_0 and μ, provided that $m_0 \geq 20$ and $0.1 \leq \mu \leq 0.4$. However, in order to take a trade-off between effect and efficiency, we suggest that the two parameters take values with $30 \leq m_0 \leq 40$ and $0.1 \leq \mu \leq 0.3$.

8.7 Engineering application example

We examine further whether the above approaches can effectively solve the machine tool spindle design optimization problem with noise. The original problem includes four decision variables, described by a static bi-objective model (Tan et al. 2001). In order to examine the above approaches, we transform such static model into the following MEVP model:

Table 5 Comparison of *ACM* values with different settings of (m_0, μ); *ACM* denotes the average of 100 results on *CM*

(m_0, μ)	*ACM*				(m_0, μ)	*ACM*			
	MOP1	MOP3	ZDT1	ZDT3		MOP1	MOP3	ZDT1	ZDT3
(10, 0)	0.21	1.58	0.05	0.07	(60, 0)	0.10	1.34	0.02	0.05
(20, 0.1)	0.14	0.47	0.03	0.06	(50, 0.1)	0.09	0.46	0.02	0.05
(30, 0.2)	0.11	0.40	0.03	0.05	(40, 0.2)	0.10	0.35	0.02	0.05
(40, 0.3)	0.11	0.31	0.03	0.06	(30, 0.3)	0.12	0.34	0.03	0.05
(50, 0.4)	0.11	0.27	0.05	0.06	(20, 0.4)	0.14	0.33	0.06	0.07
(60, 0.5)	0.12	0.22	0.28	0.08	(10, 0.5)	0.20	0.31	0.08	0.12

$$
\begin{aligned}
&\min E\left[f_1(x,\xi), f_2(x,\xi)\right] \\
&f_1(\boldsymbol{x},\xi) = \frac{\pi}{4}\left[a\left(d_a^2 - d_0^2\right) + l\left(d_b^2 - d_0^2\right)\right] + \xi_1, \\
&f_2(\boldsymbol{x},\xi) = \frac{Fa^3}{3EI_a}\left(1 + \frac{lI_a}{aI_b}\right) + \frac{F}{c_a}\left[\left(1 + \frac{a}{l}\right)^2 \right.\\
&\qquad\qquad \left. + \frac{c_a a^2}{c_b l^2}\right] + \xi_2, \\
&x = \{d_a, d_b, d_0, l\}, 180\,\text{mm} \le l \le 200\,\text{mm}, \\
&d_0 \le \frac{d_b}{1.25}, d_b \le \frac{d_a}{1.05}, d_a \in \{80, 85, 90, 95\}, \\
&d_b \in \{75, 80, 85, 90\}, c_a = 35400\,|\delta_{ra}|^{\frac{1}{9}}\, d_a^{\frac{10}{9}}, \\
&c_b = 35400\,|\delta_{rb}|^{\frac{1}{9}}\, d_b^{\frac{10}{9}}, I_a = 0.049\left(d_a^4 - d_0^4\right), \\
&I_b = 0.049\left(d_b^4 - d_0^4\right), \\
&E = 210000\,\text{N/mm}^2, F = 10000\,\text{N}, \\
&\delta_{ra} = 0.001, \delta_{rb} = -0.001, \\
&\xi_1, \xi_2 \sim N(0, \sigma^2).
\end{aligned}
$$

This is a difficult optimization problem, as both continuous and discrete decision variables as well as some constraints are included. In this experiment, the six approaches take the same maximal evaluation number 10^5; we examine their ability of noise suppression by considering different settings of variance σ. Here, take $\sigma = 0, 0.5, 1$ and 1.5. Like the above experiments, for a given value on σ each approach executes 100 runs. Thereafter, the statistical values are shown in Table 6 below. Take $\sigma = 0.5$ and 1.5 for example. We only display the acquired nondominated fronts in Fig. 7, due to the unknown Pareto front of such problem.

Through Table 6, it is obvious that the similar conclusions to those in Sects. 8.3 and 8.4 can be drawn. More precisely, whatever σ takes within 0 and 1.5, MEIOA can stably seek some approximate solutions for each run, while Fig. 7 hints that the nondominated fronts acquired by it are with relatively satisfactory distributions and wide coverage scopes. Thus, it can effectively suppress noisy influence on solution search quality. However, other approaches expose their weak ability of noise handling and the poor diversity of population, as the solutions obtained by them depend greatly on the amplitude of σ. It seems that their solutions found are with uniform distributions, due to their small values on Mean in *CD*. In fact, when σ takes a large value, the values on Mean in *CS* show that the compared approaches, in particular HEA, can only find some solutions which cover narrow regions, because their static sampling schemes result that those high-quality individuals in a given evolving population cannot be effectively discriminated. In addition, we easily see that MEIOA spends the least runtime to solve the above problem and PDMIOA is secondary. We can clearly emphasize that MEIOA is the best for the above problem.

9 Conclusions and further work

This work aims to investigate one multi-objective immune optimization approach (MEIOA) solving a challenging topic in the context of optimization-nonlinear multi-objective expected value programming with unknown noise distribution. We first transform one such kind of multi-objective programming into a sample average approximation model with the same sample size for each candidate solution. Second, this approximation model is extended into a sample-dependent approximation model which the sample size of each candidate depends on its quality. One such model is used to seek the desired approximate solutions. In order to reduce MEIOA's computational complexity, we study the relationship between the theoretically best individual set and the empirically best individual set included in a given population. This derives out a useful sample bound estimate for each individual. By means of one such estimate, a reported Hoeffding's inequality and an adaptive sampling rule, an adaptive racing ranking approach (ARRA) is designed to help MEIOA determine those empirically best individuals presented in the current population. Additionally, a novel metric criterion model, based on the well-known crowding distance approach and the version of vector inner product is constructed to pick up some excellent and diverse individuals. Thereafter, as related to the principle of the immune response, some bio-immune inspirations are adopted to develop an immune optimization mechanism which consists of ARRA, proliferation, adaptive mutation and immune selection. The theoretical analysis demonstrates that MEIOA's computa-

Table 6 Statistical comparison of ε-nondominated sets for the above engineering problem

σ	Algor.	*ACR* (%)						*CD*		*ACS*		*AT* (s)
		NNIA	MAMMOIA	HEA	FPGA	PDMIOA	MEIOA	Mean	SD	Mean	SD	
0	NNIA	0	31	2	5	28	1	$<10^{-2}$	$<10^{-2}$	1.22	0.01	1.9
	MAMMOIA	5	0	0	1	4	0	0.41	0.11	1.19	0.06	2.2
	HEA	14	12	0	0	14	1	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	$<10^{-2}$	1.8
	FPGA	8	13	0	0	9	1	0.14	0.05	0.60	0.25	3.2
	PDMIOA	22	18	1	5	0	2	0.02	$<10^{-2}$	0.91	0.10	1.0
	MEIOA	**62**	**42**	**5**	**16**	**70**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.22**	**0.01**	**0.4**
0.5	NNIA	0	20	2	9	15	32	0.02	$<10^{-2}$	0.67	0.10	8.2
	MAMMOIA	52	0	12	27	24	45	$<10^{-2}$	$<10^{-2}$	1.23	$<10^{-2}$	8.5
	HEA	8	4	0	23	10	4	0.01	$<10^{-2}$	0.05	0.02	8.9
	FPGA	14	10	4	0	10	15	0.04	0.01	0.21	0.09	9.7
	PDMIOA	22	10	2	7	0	18	0.01	$<10^{-2}$	0.18	0.02	7.4
	MEIOA	**60**	**65**	**6**	**16**	**34**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.36**	$\mathbf{<10^{-2}}$	**5.4**
1.0	NNIA	0	20	3	12	11	25	0.03	$<10^{-2}$	0.63	0.08	7.6
	MAMMOIA	54	0	9	28	32	42	$<10^{-2}$	$<10^{-2}$	1.11	$<10^{-2}$	7.8
	HEA	6	3	0	22	8	3	0.01	$<10^{-2}$	0.04	0.01	8.2
	FPGA	9	7	6	0	10	8	0.01	$<10^{-2}$	0.09	0.02	8.8
	PDMIOA	12	7	0	4	0	11	0.01	$<10^{-2}$	0.06	0.02	6.8
	MEIOA	**64**	**65**	**7**	**26**	**52**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.23**	**0.01**	**3.8**
1.5	NNIA	0	21	2	5	15	17	0.07	0.01	0.62	0.09	7.5
	MAMMOIA	52	0	9	25	22	37	$<10^{-2}$	$<10^{-2}$	1.20	$<10^{-2}$	7.8
	HEA	9	3	0	23	7	7	0.01	$<10^{-2}$	0.04	0.02	8.2
	FPGA	11	4	5	0	9	8	0.01	$<10^{-2}$	0.07	0.02	8.8
	PDMIOA	11	9	3	2	0	14	0.01	$<10^{-2}$	0.08	0.02	6.8
	MEIOA	**43**	**57**	**12**	**11**	**32**	**0**	$\mathbf{<10^{-2}}$	$\mathbf{<10^{-2}}$	**1.30**	**0.02**	**4.8**

Each bold value denotes the best of results acquired by the approaches for a given performance index

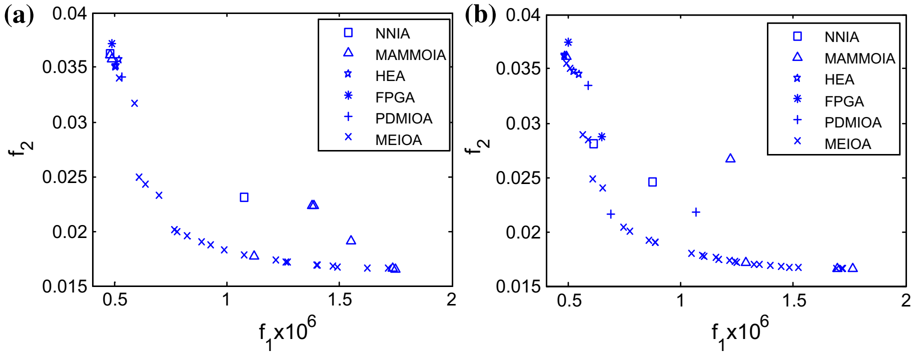

Fig. 7 Comparison of nondominated fronts found by the six approaches for the above engineering problem. **a** $\sigma = 0.5$, **b** $\sigma = 1.5$

tional cost is decided mainly by $M$and M_2. By two test suites of benchmark problems and an engineering example, comparative experiments have illustrated that MEIOA is a competitive optimizer and also performs well over the compared approaches. The sensitivity analysis has indicated that MEIOA is robust for nonlinear MEVP problems. However, whereas we make some studies on how to explore immune optimization to solve nonlinear MEVP, some issues will be further studied. For example, its structures needs to be optimized in the precondition of improving the solution quality, while its engineering applications are to be further discussed.

Acknowledgements This work is supported by National Natural Science Foundation NSFC (61563009).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1

Proof Lemma 1. In the case of $\boldsymbol{x} \in A_\varepsilon$, there exists at least a subscription j_0 satisfying that

$$E[f_{j_0}(\boldsymbol{x}, \boldsymbol{\xi})] + \varepsilon < E[f_{j_0}(\boldsymbol{y}, \boldsymbol{\xi})]. \tag{18}$$

Hence, once $\hat{f}_{j_0}(\boldsymbol{y}) + \varepsilon \le \hat{f}_{j_0}(\boldsymbol{x})$, one can imply that

$$\hat{f}_{j_0}(\boldsymbol{y}) + \varepsilon < E[f_{j_0}(\boldsymbol{y}, \boldsymbol{\xi})] - \frac{\varepsilon}{2}$$

or

$$\hat{f}_{j_0}(\boldsymbol{x}) > E[f_{j_0}(\boldsymbol{x}, \xi)] + \frac{\varepsilon}{2}.$$

Otherwise, by Eq. (18) we have that

$$\hat{f}_{j_0}(\boldsymbol{y}) + \varepsilon \ge E[f_{j_0}(\boldsymbol{y}, \boldsymbol{\xi})] - \frac{\varepsilon}{2} > E[f_{j_0}(\boldsymbol{x}, \boldsymbol{\xi})] + \frac{\varepsilon}{2} \ge \hat{f}_{j_0}(\boldsymbol{x}). \tag{19}$$

This yields contradiction. Consequently,

$$\begin{aligned}\Pr\left\{\hat{f}_{j_0}(y) + \varepsilon \le \hat{f}_{j_0}(x)\right\} &\le \Pr\{\hat{f}_{j_0}(y) < E[f_{j_0}(y, \xi)] \\ &-\frac{3\varepsilon}{2}\} + \Pr\{\hat{f}_{j_0}(x) > E[f_{j_0}(x, \boldsymbol{\xi})] + \frac{\varepsilon}{2}\}.\end{aligned} \tag{20}$$

Further, it follows from Eq. (20) and Theorem 1 that

$$\Pr\left\{\hat{f}_{j_0}(\boldsymbol{y}) + \varepsilon \le \hat{f}_{j_0}(\boldsymbol{x})\right\} \le 2e^{-mc\varepsilon^2}. \tag{21}$$

Thus,

$$\begin{aligned}\Pr\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\} &\le \prod_{j=1}^{q} \Pr\left\{\hat{f}_j(\boldsymbol{y}) + \varepsilon \le \hat{f}_j(\boldsymbol{x})\right\} \\ &\le \Pr\left\{\hat{f}_{j_0}(\boldsymbol{y}) + \varepsilon \le \hat{f}_{j_0}(\boldsymbol{x})\right\} \le 2e^{-mc\varepsilon^2}.\end{aligned} \tag{22}$$

The proof is completed. □

Proof of Theorem 2. For a given $\boldsymbol{x} \in A_\varepsilon$, if $\boldsymbol{x} \notin \hat{A}_\varepsilon$, there exists $\boldsymbol{y} \in \hat{A}_\varepsilon$ such that $\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}$. Hence,

$$\begin{aligned}\Pr\{\boldsymbol{x} \notin \hat{A}_\varepsilon\} &\le \Pr\{\exists \boldsymbol{y} \in \hat{A}_\varepsilon, s.t. \boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\} \\ &\le \sum_{\boldsymbol{y} \in \hat{A}_\varepsilon} \Pr\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\},\end{aligned} \tag{23}$$

and accordingly,

$$\begin{aligned}\Pr\{A_\varepsilon \not\subset \hat{A}_\varepsilon\} &= \Pr\{\exists \boldsymbol{x} \in A_\varepsilon s.t. \boldsymbol{x} \notin \hat{A}_\varepsilon\} \\ &\le \Pr\{\exists \boldsymbol{x} \in A_\varepsilon, \boldsymbol{y} \in \hat{A}_\varepsilon, s.t. \boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\} \\ &\le \sum_{\boldsymbol{x} \in A_\varepsilon} \sum_{\boldsymbol{y} \in \hat{A}_\varepsilon} \Pr\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\}.\end{aligned} \tag{24}$$

Therefore, as related to Lemma 1 and Eq. (24), we can obtain that

$$\Pr\{A_\varepsilon \subseteq \hat{A}_\varepsilon\} = 1 - \Pr\{A_\varepsilon \not\subset \hat{A}_\varepsilon\} \ge 1 - 2N^2 e^{-mc\varepsilon^2}. \tag{25}$$

This finishes the proof. □

The proof of Theorem 3. In the case of $x \in A_\varepsilon$, we know that

$$\Pr\{\boldsymbol{x} \notin \hat{A}_\varepsilon\} = \Pr\{\exists \boldsymbol{y} \in \hat{A}_\varepsilon, s.t. \boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\} \le \sum_{\boldsymbol{y} \in \hat{A}_\varepsilon} \Pr\{\boldsymbol{y} \prec_{\hat{\varepsilon}} \boldsymbol{x}\}. \tag{26}$$

By Lemma 1, Eq. (21) implies that

$$\Pr\{\boldsymbol{x} \notin \hat{A}_\varepsilon\} \le 2|\hat{A}_\varepsilon| e^{-mc\varepsilon^2}. \tag{27}$$

Subsequently, we can easily obtain that

$$\Pr\{\hat{A}_\varepsilon \cap A_\varepsilon = \emptyset\} \le \prod_{x \in A_\varepsilon} \Pr\{x \notin \hat{A}_\varepsilon\} \le 2Ne^{-mc\varepsilon^2}. \tag{28}$$

Thereby, the conclusion is true. □

The proof of Theorem 3. In Step 4, those β-dominated and redundant B cells in M_{set} will be eliminated after copying B_n into M_{set}, for which the computational complexity is $\text{O}(|\text{M}_{set}| \log |\text{M}_{set}|)$. In particular, it is possible that the number of B cells in M_{set} is beyond $(N + M)$. Thus, in the worst case we can assert that the complexity of Step 4 is $\text{O}((N + M)\log(N + M))$. Step 5 needs to calculate PDM values of elements in A_n; more precisely, we need to calculate their crowding distances with at most $N\text{C}_{\max} \log N\text{C}_{\max}$ times and their values on S with $(N\text{C}_{\max}+1)N\text{C}_{\max}/2$ times. Step 8 executes mutation with at most $Np\text{C}_{\max}$ times. In Step 9, the size of $\text{B}_n \cup \text{E}_n$ is at most $\text{NC}_{\max}$. Thus, when ARRA is enforced on $\text{B}_n \cup \text{E}_n$, the complexity in the worst case is $O(NC_{\max}(M_2 + \log NC_{\max}))$ by means of the computational complexity in Sect. 5.1. Summarily, MEIOA's computational complexity in the worst case is decided by

$$\begin{aligned}O_c &= O\left((N + M)\log(N + M)\right) + O\left(NpC_{\max}\right) \\ &\quad + O\left(NC_{\max}(M_2 + \log NC_{\max})\right) + O((NC_{\max})^2) \\ &= O\left((N + M)\log(N + M) + NC_{\max}(M_2 + p\right. \\ &\quad \left. + NC_{\max})\right).\end{aligned}$$

Therefore, the conclusion is right. □

Appendix 2

See Table 7.

Table 7 The first test suite (Van Veldhuizen 1999)

Problem	Definition	Constraints
MOP1	$\min \mathrm{E}[f_1(x, \boldsymbol{\xi}), f_2(x, \boldsymbol{\xi})]$ $f_1(x, \boldsymbol{\xi}) = x^2 + \xi_{1,} f_2(x, \boldsymbol{\xi}) = (x-2)^2 + \xi_2$	$x \in= [-1000, 1000], \xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$
MOP2	$\min \mathrm{E}[f_1(\boldsymbol{x}, \boldsymbol{\xi}), f_2(\boldsymbol{x}, \boldsymbol{\xi})]$ $f_1(\boldsymbol{x}, \boldsymbol{\xi}) = 1 - \exp(-\Sigma_{i=1}^{m}(x_i - m^{-1/2})^2) + \xi_{1,}$ $f_2(\boldsymbol{x}, \boldsymbol{\xi}) = 1 - \exp(-\Sigma_{i=1}^{m}(x_i + m^{-1/2})^2) + \xi_2$	$m = 3, x_1, \ldots, x_m \in [-4, 4], \xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$
MOP3	$\max \mathrm{E}[f_1(x, y, \boldsymbol{\xi}), f_2(x, y, \boldsymbol{\xi})]$ $f_1(x, y, \boldsymbol{\xi}) = -1 - (\mathrm{A}_1 - \mathrm{B}_1)^2 - (\mathrm{A}_2 - \mathrm{B}_2)^2 + \xi_{1,}$ $f_2(x, y, \boldsymbol{\xi}) = -(x+3)^2 - (y+1)^2 + \xi_2$	$x, y \in [-3.1416, 3.1416], \xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$ $\mathrm{A}_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $\mathrm{A}_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2,$ $\mathrm{B}_1 = 0.5 \sin x - 2 \cos x + \sin y - 1.5 \cos y$ $\mathrm{B}_2 = 1.5 \sin x - \cos x + 2 \sin y - 0.5 \cos y$
MOP4	$\min \mathrm{E}[f_1(\boldsymbol{x}, \boldsymbol{\xi}), f_2(\boldsymbol{x}, \boldsymbol{\xi})]$ $f_1(\boldsymbol{x}, \xi) = \Sigma_{i=1}^{m-1}(-10 \exp(-0.2 \times (x_i^2 + x_{i+1}^2)^{1/2})) + \xi_{1,}$ $f_2(\boldsymbol{x}, \xi) = \Sigma_{i=1}^{m}(\lvert x_i \rvert^{0.8}) + 5 \sin(x_i)^3 + \xi_2$	$m = 3, x_1, \ldots, x_m \in [-5, 5],$ $\xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$
MOP6	$\max \mathrm{E}[f_1(x, y, \boldsymbol{\xi}), f_2(x, y, \boldsymbol{\xi})]$ $f_1(x, y, \boldsymbol{\xi}) = x$ $+ \xi_1, f_2(x, y, \xi) = (1 + 10y)$ $\times (1 - (x/(1+10y)))^{\alpha}$ $- x \sin(2\pi q x)/(1+10y)) + \xi_2$	$x, y \in [0, 1], q = 4, \alpha = 2, \xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$

Appendix 3

See Table 8.

Table 8 The second test suite (Zitzler et al. 2000)

Problem	Definition	Constraints
ZDT1	$\min \mathrm{E}[f_1(\boldsymbol{x}, \xi), f_2(\boldsymbol{x}, \xi)]$ $f_1(\boldsymbol{x}, \boldsymbol{\xi}) = x_1 + \xi_1, f_2(\boldsymbol{x}, \boldsymbol{\xi})$ $= g(x_2, \ldots, x_m) h(x_1, g(x_2, \ldots, x_m)) + \xi_2, \boldsymbol{x} = (x_1, \ldots, x_m),$ $g(x_2, \ldots, x_m) = 1 + 9\Sigma_{i=2}^{m} \quad x_i, h(f_1, g) = 1 - (x_1/g)^{1/2}$	$m = 30, x_1, \ldots, x_m \in [0, 1],$ $\xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$
ZDT2	$\min \mathrm{E}[f_1(\boldsymbol{x}, \xi), f_2(\boldsymbol{x}, \xi)]$ $f_1(\boldsymbol{x}, \boldsymbol{\xi}) = x_1 + \xi_1, f_2(\boldsymbol{x}, \xi) = g(x_2, \ldots, x_m) h(x_1, g(x_2, \ldots, x_m)) + \xi_2$ $\boldsymbol{x} = (x_1, \ldots, x_m), g(x_2, \ldots, x_m) = 1 + 9\Sigma_{i=2}^{m} \quad x_i, h(f_1, g)$ $= 1 - (x_1/g)^2$	$m = 30, x_1, \ldots, x_m \in [0, 1],$ $\xi_1 \sim N(0,1), \xi_2 \sim N(0,1)$
ZDT3	$\min \mathrm{E}[f_1(\boldsymbol{x}, \xi), f_2(\boldsymbol{x}, \xi)]$ $f_1(\boldsymbol{x}, \boldsymbol{\xi}) = x_1 + \xi_1, f_2(\boldsymbol{x}, \boldsymbol{\xi})$ $= g(x_2, \ldots, x_m) h(x_1, g(x_2, \ldots, x_m)) + \xi_2, \boldsymbol{x} = (x_1, \ldots, x_m),$ $g(x_2, \ldots, x_m) = 1 + 9\Sigma_{i=2}^{m} \quad x_i,$ $h(f_1, g) = 1 - (x_1/g)^{1/2} - (x_1/g) \sin(10\pi x_1)$	$m = 30, x_1, \ldots, x_m \in [0, 1],$ $\xi_1 \sim N(0, 1), \xi_2 \sim N(0, 1)$
ZDT4	$\min \mathrm{E}[f_1(\boldsymbol{x}, \xi), f_2(\boldsymbol{x}, \xi)]$ $f_1(\boldsymbol{x}, \boldsymbol{\xi}) = x_1 + \xi_1, f_2(\boldsymbol{x}, \boldsymbol{\xi})$ $= g(x_2, \ldots, x_m) h(x_1, g(x_2, \ldots, x_m)) + \xi_{2,} \boldsymbol{x} = (x_1, \ldots, x_m),$ $g(x_2, \ldots, x_m)$ $= 1 + 10(m-1) + \Sigma_{i=2}^{m}(x_i^2 - 10 \cos(4\pi x_i)), h(f_1, g) = 1 - (x_1/g)^{1/2}$	$m = 30, x_1 \in [0, 1], x_2, \ldots, x_m \in [-5, 5], \xi_1 \sim N(0, 1),$ $\xi_2 \sim N(0,1)$

References

Aickelin U, Dasgupta D, Gu F (2014) Artificial immune systems. Search Methodologies. Springer US, pp 187–211

Aydin I, Karakose M, Akin E (2011) A multi-objective artificial immune approach for parameter optimization in support vector machine. Appl Soft Comput 11:120–129

Batista LS, Campelo F, Guimarães FG et al (2011) Pareto cone ε-dominance: improving convergence and diversity in multiobjective evolutionary approaches. In: Evolutionary multi-criterion optimization, Springer, Berlin, pp 76–90

Bui LT et al (2005) Fitness inheritance for noisy evolutionary multi-objective optimization. In: The 7th annual conference on genetic and evolutionary computation, ACM, pp 779–785

Cantú-Paz E (2004) Adaptive sampling for noisy problems. In: Genetic and evolutionary computation conference, GECCO2004, pp 947–958

Chen CH (2003) Efficient sampling for simulation-based optimization under uncertainty. In: Fourth International symposium on uncertainty modeling and analysis, ISUMA'03, pp 386–391

Coello CAC, Cortés NC (2005) Solving multi-objective optimization problems using an artificial immune system. Genet Program Evol Mach 6:163–190

Corne DW, Jerram NR, Knowles JD et al (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Genetic and evolutionary computation conference, GECCO'2001, pp 283–290

Deb K et al (2002) A fast and elitist multi-objective genetic approach: NSGA-II. IEEE Trans Evol Comput 6:182–197

Drugan MM, Nowe A (2013) Designing multi-objective multi-armed bandits approaches: a study. In: International joint conference on neural networks, IJCNN, pp 1–8

El-Wahed WFA, Lee SM (2006) Interactive fuzzy goal programming for multi-objective transportation problems. Omega 34(2):158–166

Eskandari H, Geiger CD (2009) Evolutionary multi-objective optimization in noisy problem environments. J Heuristics 15:559–595

Even-Dar E, Mannor S, Mansour Y (2006) Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. J Mach Learn Res 7:1079–1105

Gong MG, Jiao LC, Du HF, Bo LF (2008) Multi-objective immune approach with nondominated neighbor-based selection. Evol Comput 16:225–255

Gong MG et al (2013) Identification of multi-resolution network structures with multi-objective immune approach. Appl Soft Comput 13:1705–1717

Gutjahr WJ, Pichler A (2016) Stochastic multi-objective optimization: a survey on non-scalarizing methods. Ann Oper Res 236:475–499

Higle JL, Zhao L (2004) Adaptive and nonadaptive samples in solving stochastic linear programs: a computational investigation. The University of Arizona, Tucson

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58(301):13–30

Hu ZH (2010) A multiobjective immune approach based on a multiple-affinity model. Eur J Oper Res 202:60–72

Hughes EJ (2001) Constraint handling with uncertain and noisy multi-objective evolution. In: Congress on evolutionary computation 2001, CEC'2001, pp 963–970

Jin Y, Branke J (2005) Evolutionary optimization in uncertain environments: a survey. IEEE Trans Evol Comput 9:303–317

Lee LH et al (2010) Finding the nondominated Pareto set for multi-objective simulation models. IIE Trans 42:656–674

Lee LH, Pujowidianto NA, Li LW et al (2012) Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints. IEEE Trans Autom Control 57:2940–2945

Lin Q, Chen J (2013) A novel micro-population immune multi-objective optimization approach. Comput Oper Res 40:1590–1601

Liu B (2009) Theory and practice of uncertain programming. Physica, Heidelberg

Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. Struct Multidiscip Optim 41(6):853–862

Owen J, Punt J, Stranford S (2013) Kuby immunology, 7th edn. Freeman, New York

Park T, Ryu KR (2011) Accumulative sampling for noisy evolutionary multi-objective optimization. In: the 13th annual conference on Genetic and evolutionary computation, ACM, pp 793–800

Phan DH, Suzuki J (2012) A non-parametric statistical dominance operator for noisy multi-objective optimization. In: Simulated evolution and learning, SEAL'12, pp 42–51

Qi Y, Liu F, Liu M et al (2012) Multi-objective immune approach with Baldwinian learning. Appl Soft Comput 12:2654–2674

Qi Y, Hou Z, Yin M et al (2015) An immune multi-objective optimization approach with differential evolution inspired recombination. Appl Soft Comput 29:395–410

Robert C, Casella G (2013) Monte Carlo statistical methods. Springer, Berlin

Shapiro A, Dentcheva D, Ruszczyński A (2009) Lectures on stochastic programming: modeling and theory. SIAM-MPS Philadelphia

Tan KC, Lee TH, Khor EF (2001) Evolutionary approaches with dynamic population size and local exploration for multiobjective optimization. IEEE Trans Evol Comput 5:565–588

Tan KC, Goh CK, Mamun AA et al (2008) An evolutionary artificial immune system for multi-objective optimization. Eur J Oper Res 187:371–392

Trautmann H, Mehnen J, Naujoks B (2009) Pareto-dominance in noisy environments. In IEEE congress on evolutionary computation(CEC'09), pp 3119–3126

Van Veldhuizen DA (1999) Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Ph. D. Thesis, OH: Air force Institute of Technology, Technical Report No. AFIT/DS/ENG/99-01, Dayton

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11:712–731

Zhang ZH, Tu X (2007a) Immune approach with adaptive sampling in noisy environments and its application to stochastic optimization problems. IEEE Comput Intell Mag 2:29–40

Zhang ZH, Tu X (2007b) Probabilistic dominance-based multi-objective immune optimization approach in noisy environments. J Comput Theor Nanosci 4:1380–1387

Zhang ZH, Wang L, Liao M (2013a) Adaptive sampling immune approach solving joint chance-constrained programming. J Control Theory Appl 11:237–246

Zhang ZH, Wang L, Long F (2013b) Immune optimization approach solving multi-objective chance constrained programming. Evol Syst 6:41–53

Zhang W, Xu W, Liu G, et al (2015) An effective hybrid evolutionary approach for stochastic multiobjective assembly line balancing problem. J Intell Manuf 1–8. doi:10.1007/s10845-015-1037-5

Zheng JH et al (2004) A multi-objective genetic approach based on quick sort. Advances in Artificial Intelligence. Springer, Berlin

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary approaches: empirical results. Evol Comput 8:173–195

Zitzler E, Thiele L (1999) Multi-objective evolutionary approaches: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3:257–271