



A novel evolution strategy for multiobjective optimization problem[☆]

Shu Min Yang^{a,*}, Dong Guo Shao^a, Yang Jie Luo^b

^a State Key Laboratory of Water Resource and Hydropower Engineering Science,
Wuhan University, Wuhan 430072, PR China

^b College of Urban and Environmental Sciences, Northeast Normal University,
Changchun 130024, PR China

Abstract

Recent literatures have suggested that multiobjective evolutionary algorithms (MOEAs) can serve as a more exploratory and effective tool in solving multiobjective optimization problems (MOPs) than traditional optimizers. In order to contain a good approximation of Pareto optimal set with wide diversity associated with the inherent characters and variability of MOPs, this paper proposes a new evolutionary approach— (μ, λ) multiobjective evolution strategy $((\mu, \lambda)$ -MOES). Following the highlight of how to balance proximity and diversity of individuals in exploration and exploitation stages respectively, some cooperative techniques are devised. Firstly, a novel combinatorial exploration operator that develops strong points from Gaussian mutation of proximity exploration and from Cauchy mutation of diversity preservation is elaborately designed. Additionally, we employ a complete nondominance selection so as to ensure maximal pressure for proximity exploitation while a fitness assignment determined by dominance and population diversity information is simultaneously used

[☆] Supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB415206.

* Corresponding author. Also affiliated with College of Water Resource and Hydropower, Wuhan University, Wuhan 430072, PR China.

E-mail address: ouyangshm@hotmail.com (S.M. Yang).

to ensure maximal diversity preservation. Moreover, a dynamic external archive is introduced to store elitist individuals as well as relatively better individuals and exchange information with the current population when performing archive increase scheme and archive decrease scheme. By graphical presentation and examination of selected performance metrics on three prominent benchmark test functions, (μ, λ) -MOES is found to outperform SPEA-II to some extent in terms of finding a near-optimal, well-extended and uniformly diversified Pareto optimal front.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Multiobjective optimization; Pareto optimal; Evolution strategy; Evolutionary algorithm

1. Introduction

When attempting to optimize a decision in industrial and scientific applications, the designer is frequently faced with the problem of achieving several design targets, some of which may be conflicting and noncommensurable and wherein a gain in one objective is at the expense of another. This problem can be generally reduced to multiobjective optimization problems (MOPs) in operational description, which has been in the spotlight of operations research communities over years. Usually, there is no unique optimal solution, but rather a set of alternative solutions and these solutions are optimal in the wider sense that no other solutions in decision space are superior to them when all objectives are considered. They are known as *Pareto optimal* solutions, also termed *nondominated noninferior*, *admissible*, or *efficient* solutions [1].

Consider, for example, the optimal allocation of water resource for sustainable development. A mathematic model of the optimal allocation might be maximizing the overall profits from consumptive uses such as domestic, agriculture and industrial uses while minimizing the environmental deterioration resulted from those consumptive uses [2]. However, these goals are generally conflicting: One allocation plan may gain high economic profits at high environmental cost, an alternative low profit might considerably restrict economy development in despite of beautiful environment—neither of these allocation plans meet the hardcore of sustainable development, or can be said to be optimal if we do not include any preference information (e.g., a ranking of the objectives). Thus if no such information is available, it may be useful to have general knowledge about all these alternative allocation plans. So, the tools exploring decision space for Pareto optimal solutions in reasonable time can essentially aid the decision makers in arriving at a final decision.

During the past decade, various multiobjective evolutionary algorithms (MOEAs) have been proposed and applied in MOPs [1]. A representative collection of these algorithms includes the vector evaluated genetic algorithm

(VEGA) by Schaffer [3], the niched Pareto genetic algorithm (NPGA) [4] and the nondominated sorting genetic algorithm (NSGA) by Srinivas and Deb [5], the nondominated sorting genetic algorithm II (NSGA-II) by Deb et al. [6], the strength Pareto evolutionary algorithm (SPEA) by Zitzler and Thiele [7], the SPEA-II by Zitzler et al. [8], the Pareto archived evolution strategy (PAES) by Knowles and Corne [9], the memetic PAES (M-PAES) by Knowles and Corne [10] and the multiobjective concordance evolutionary algorithm (MOCEA) by Cui et al. [11]. Although these MOEAs differ from each other in both exploitation and exploration, they share the common purpose—searching for a near-optimal, well-extended and uniformly diversified Pareto optimal front for a given MOP. However, this ultimate goal is far from being accomplished by the existing MOEAs as documented in the literature, e.g., [1]. On the one hand, most of MOPs are very complicated and have own inherit characters and variability, computational resources are required to be homogeneously distributed in a high-dimensional decision space. On the other hand, those fitter individuals generally have strong tendencies to restrict searching efforts within local areas because of the genetic drift phenomenon [12], which results in the loss of diversity. This outlines the hot issue of MOEA—how to improve algorithm performance and result quality that indicates the balance between proximity and diversity as well.

Following this issue, the present work mainly deals with the multiobjective version of evolution strategies (ES), which is a crucial branch of EAs. ES can be dated back to Schwefel's work in optimization of alterable parameters and it has shown some great advantages in numerical optimization (see [13] for more details). A special aspect of importance in which ES differs from most other EAs is that it has long self-adaptive mechanisms by usually including strategy parameters, which can automatically guide the search to explore and exploit the local and global topology of decision space. Therefore, we develop the multiobjective version of ES and it is also based on Pareto optimal approach, named (μ, λ) -MOES where the symbol μ refers to the size of the parent population and the symbol λ ($\lambda > \mu$) refers to the number of offspring that is produced in a single generation before selection. It is also noted that in M-PAES [10] only one offspring are produced ($\lambda = 1$), but (μ, λ) -MOES maintains multiple offspring in the intermediate population ($\lambda > 1$) and offspring replace their parents. Furthermore, we introduce some novel techniques in order to achieve higher performance by means of adaptive balance between proximity and diversity.

In (μ, λ) -MOES, a novel combinatorial exploration operator that develops strong points from Gaussian mutation of proximity exploration and from Cauchy mutation of diversity preservation is elaborately devised. Additionally, we employ a complete nondominance selection so as to ensure maximal pressure for proximity exploitation while simultaneously performing a fitness assignment determined by dominance and population diversity to ensure maximal

diversity preservation. Moreover, the most differentia from other MOEAs is the purpose, size and update scheme of the external archive, which is a dynamic external archive storing elitist individuals as well as relatively better individuals and exchanging information with the current population. So the external archive is endowed with the responsibility of saving even dominated individuals that are sometimes useful. In order to keep balance of the external archive size in the probability sense, the archive increase scheme and archive decrease scheme are designed. All these techniques work cooperatively and automatically to provide satisfactory results.

The remainder of this paper is organized as follows. In Section 2, we introduce some basic definition related to our algorithm and analyses of how to balance proximity and diversity in exploration and exploitation stages. Section 3 describes (μ, λ) -MOES arithmetic in detail. The empirical results and comparisons between (μ, λ) -MOES and SPEA-II [8] on three prominent benchmark test functions are presented in Section 4. Eventually, we conclude the paper with some remarks and future research directions in the last section.

2. Preliminaries

2.1. Dominance, proximity and diversity

Without loss of generality, consider the following MOP with n decision variables x and m ($m > 1$) objectives y :

$$\begin{aligned} \min y &= f(x) = \{f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\} \\ \text{s.t. } x &= (x_1, \dots, x_n) \in X \subset R^n, \\ y &= (y_1, \dots, y_m) \in Y \subset R^m, \end{aligned} \quad (1)$$

where x is called decision vector, X decision space, y objective vector and Y objective space; f defines the mapping function between X and Y . It is clear that Y can only be partially ordered. The scenario considered in this paper involves an arbitrary optimization with objectives, which are all to be minimized and all equally important, i.e., no additional knowledge about the problem is available and it means that the objectives cannot be combined into a single scalar objective to be optimized. As a result, the sets of solutions exist such that each solution in this set is equally preferable. To formalize these notions, the following four concepts are of importance:

- *Pareto dominance*: A solution x^0 is said to *dominate* (Pareto) another solution x^1 (denoted $x^0 \succ x^1$) if and only if $\forall i \in \{1, \dots, m\}: f_i(x^0) \leq f_i(x^1) \wedge (\exists k \in \{1, \dots, m\}: f_k(x^0) < f_k(x^1))$.
- *Pareto optimal*: A solution x^0 is said to be (Pareto optimal) *nondominated* if and only if $\neg \exists x^1 \in X: x^1 \succ x^0$.

- *Pareto optimal set*: The set P_S of all Pareto optimal solutions: $P_S = \{\mathbf{x}^0 \mid \neg \exists \mathbf{x}^1 \in X: \mathbf{x}^1 \succ \mathbf{x}^0\}$.
- *Pareto optimal front*: The set P_F of all objective function value corresponding to the solutions in P_S : $P_F = \{f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \mid \mathbf{x} \in P_S\}$.

The optimal result of MOP is no other than the Pareto optimal set P_S . However, the size of this set may be infinite, where it is impossible to find this set by using a finite number of solutions. In this case, a representative subset of P_S is desired. Generally, the characteristic of MOEAs is to search the decision space by maintaining a finite population of individuals (corresponding to the points in the decision space), which work according to the procedures that resemble the principles of natural selection and evolution. In this work, we only consider the subset of all nondominated individuals that is contained in the final population and the external archive resulted from running a MOEA. We call such subset an *approximation set* and denote it by S , correspondingly, and we call the corresponding objective set a *resulting final Pareto optimal front* and denote it by PF_{final} . Ideally, we are interested in finding an S of finite size, which contains a selection of individuals from such that the individuals in PF_{final} are diversified as possible. Unfortunately, we usually have no access to P_F on beforehand. We have to get close to P_F but in such a way that PF_{final} we found is as diversified as possible without compromising as much as possible the proximity of PF_{final} with respect to P_F . Thus the concept of *proximity* and *diversity* should be outlined. Regarding this diversity, it is of importance to note that it depends on the mapping function whether a good diversity of the individuals in decision space is also a good diversity of the individuals in objective space correspondingly. However, it is common practice to search for a good diversity of the individuals in objective space, because decision makers will ultimately have to pick a single individual as final solution according to its objective vector values. Therefore, it is often best to present a wide variety of tradeoff individuals for the specified goals in constructing MOEAs.

2.2. How to balance proximity and diversity

Proximity and diversity are twofold goals along evolutionary process and they have a direct implication on the notions of *exploration* and *exploitation* commonly used in MOEA terminology. To avoid confusion, we will refer to [14] for an exact definition.

- *Exploration* indicates the part of (μ, λ) -MOES that is concerned with the production of new offspring from a given set of parents. Since in this work we are only interested in how a new set of offspring is generated if we supply a set of parents, common definition of exploration includes the way in which recombination and mutation is performed.

- *Exploitation* indicates the parts of (μ, λ) -MOES that are concerned with the selection of a set of parents from the current population and the construction of the next population from the current population, the selected set of parent individuals and the set of offspring individuals.

Note that the update of the external archive does not appear in either of exploration and exploitation stages solely because it is intercrossed amongst both stages. In (μ, λ) -MOES, we increase the external archive in exploration stage and decrease it in exploitation stage respectively.

2.2.1. Exploration of proximity and diversity

It is of importance to have an exploration operator that is capable of producing new nondominated individuals and diversifying them to guide the search towards PF . Based on an appropriate selection of parents, competent exploration operator should be able to produce offspring in which good features of the selected parents are inheritably developed, which does not differ much from the rationale in single-objective EAs essentially. Since excessive proximity without appropriate diversity will lead to premature convergence, while excessive diversity without appropriate proximity will confuse the search guide into stagnancy, it is wise to deal with them equally and adaptively.

An interesting and relatively new field that attempts to model the regularities of parent structure uses combination of several exploration operators, named combinatorial operator. In this way, different operators are cooperated to develop each strong points and eliminate each weakness, where heuristic method is also applied. An example is taken in the fast EP for single optimization, in which Gaussian and Cauchy mutations are combined to produce offspring from the same parent and the better ones are chosen into next population [15]. Algorithms that use the similar approaches have obtained an increasing amount of attention over the last few years, achieving promising results on a large variety of problems [16–18]. These approaches are also constructive and beneficial for multiobjective optimization problem as well [19]. By doing that, the exploration of proximity and diversity has been shown to be effectively stimulated [20]. Inspired by this idea, the following three operators: discrete recombination, Gaussian mutation and Cauchy mutation are appropriately combined in our (μ, λ) -MOES. Herein, Gaussian mutation serves for proximity while Cauchy mutation serves for diversity according to status of the parents offered to them and the offspring generated by them.

2.2.2. Exploitation of proximity and diversity

It is also important to have a competent exploitation scheme that is capable of selecting a diverse set of individuals close to the set of nondominated individuals as possible. Since we can indicate how diverse we are but cannot indicate how close P_F we are to, the best we can do is to find individuals that are

not dominated by any other individuals and are widely diversified in order to ensure pressure on both proximity selection and diversity preservation. Therefore, the exploitation stage is usually split into two steps: proximity selection first and diversity preservation later and the later should never precede the former. A straightforward way to obtain selection pressure toward proximity is *complete nondominance selection* in which all dominated individuals are neglected and only nondominated individuals have the opportunity to survive and reproduce in each generation [21]. This approach is considered cautious because it easily leads to premature convergence unless strong diversity preservation is cooperated. Another class such as the concept of *domination count* [19] and *domination rank* [6] are introduced for proximity selection while *crowded comparison* [6] serves for diversity preservation. Although these approaches performed successfully in some situations, the problem caused by inappropriate exploitation scheme was still not completely resolved. Here, in (μ, λ) -MOES, we emphasize a novel exploitation scheme that is also separated into two sides for proximity selection and diversity preservation respectively, but these incompatible sides are designed to work simultaneously and cooperatively. A complete nondominance selection derived from [21] ensures maximal pressure for proximity selection while a fitness assignment determined by dominance and population information ensures maximal diversity preservation. They work aiming at the weaknesses of each counterpart self-adaptively. Particulars are discussed in Section 3.

2.2.3. Effect of external archive

In order to obtain new and diverse nondominated individuals, especially when the set of nondominated individuals approaches P_F , the concept of elitism that the best individuals of the current population are copied into the next population is accepted to be a very important role for improving the results obtained by any MOEAs [7]. Alternatively, an external archive may be more commonly used to contain nondominated individuals, and the current population and the external archive are separated and updated by exchanging individuals between them. It also helps allow preserving the good individuals that are hard to be generated in exploration stage.

It has been highlighted that if we employ a complete nondominance selection in exploitation stage without strong diversity preservation, the algorithm will easily results in premature convergence or local optimum. In evolutionary process, there are such individuals that they are not ensured nondominated responding to the current population, but are relatively better to their parents. Since these individuals may hold some useful information, they should be kept for a certain generation for further exploration and exploitation. However, they will definitely lose opportunity to survive into the next population under the complete nondominance selection, if they are not stored in the external archive. Furthermore, the external archive size is highly sensitive. In general, the

external archive size adopted in most MOEAs is bounded by a predefined maximum [6–8], so the update scheme is a replacement scheme. If the size is too small, there will not be enough schemas to exploit [20], resulting in a premature or nonuniformly PF_{final} . Otherwise, choosing an excessive size will not be desirable since it may require unnecessarily large computation resources and even confuse search guide into stagnancy. Therefore, a dynamic size adjusted automatically by the online characteristics of proximity and diversity information of the external archive will be more efficient and effective than a constant size. A dynamic size also means that we have opportunity to maintain the individuals that are not the worst ones responding to the external archive under proper circumstance. As a result, the external archive should be endowed with the responsibility of saving elitist individuals as well as relatively better individuals and a dynamic size is preferable. In order to keep balance of the external archive size in the probability sense, the archive increase scheme and archive decrease scheme should also be designed in (μ, λ) -MOES.

3. Descriptions of (μ, λ) -MOES

A few items that are of major importance to implement an adaptive balance between the proximity and diversity can be outlined when constructing our (μ, λ) -MOES.

- The exploration operator should automatically correct direction for both proximity and diversity after absorbing information of parents that are offered to it.
- Survive of better individuals should be automatically done based both on proximity selection as well as on diversity preservation.
- The external archive should be used to save elitist individuals as well as relatively better individuals, and a dynamic size is preferable.
- The external archive update scheme including archive increase scheme and archive decrease scheme should be devised to operate on the dynamic size.

Similar considerations have led to the definition of the unified model for MOEAs (UMMOEA) by Laumanns et al. [22]. Although the UMMOEA framework is important, the underlying algorithm is different in many aspects. In Fig. 1, the global pseudocode is shown based on these considerations. It should be noted that the heart of (μ, λ) -MOES involves two populations: the current population P and the external archive G . In detail, P works similar to (μ, λ) -ES for single optimization; but G serves a binary purpose—storage of elitist individuals as well as relatively better individuals found during the run and cooperative one-side parent in producing offspring. At the beginning of each generation, parents are randomly selected from P and G (the later

```

function result = MOES (  $\mu$ ,  $\lambda$ , age_min, size_min )


---


begin
  generate initial population  $P$  of  $\mu$  random individuals and evaluate
  place each nondominated member of  $P$  in an external archive  $G$ 
  set generation counter zero:  $t = 0$ 
  do
    reset the current intermediate population empty:  $H = \Phi$ 
    set offspring counter zero:  $r = 0$ 
    %%% exploration stage:
    %%% repeat until  $\lambda$  offspring is generated while archive increase scheme is performed synchronously
    do
      randomly select one individuals  $a$  and another individual  $b$  that is not dominated by  $a$  from  $P \cup G$ 
       $c = \text{ProduceOffspring}(a, b, P, G)$  %%% generate offspring via our exploration operator
       $\text{IncreaseArchive}(c, a, b, G, \text{size\_min})$  %%% decide whether to add  $c$  to  $G$  as necessary
      add offspring  $c$  into the current intermediate population:  $H \leftarrow H + c$ 
      increase offspring counter by one:  $r \leftarrow r + 1$ 
    while ( $r < \lambda$ )
     $\text{DecreaseArchive}(G, \text{age\_min}, \lambda)$  %%% decide whether to remove member from  $G$  as necessary
    %%% exploitation stage:
    %%% transfer from the intermediate population to the next population
     $\text{AssignFitness}(H)$  %%% assign each member of  $H$  fitness with our fitness assignment scheme
    sort  $H$  in descending order of fitness
    select the first  $\mu$  individuals into the next population:  $P(t+1) \leftarrow H(\mu)$ 
    increase generation counter by one:  $t \leftarrow t + 1$ 
  while (terminal criterion is not satisfied)
  set result the unique nondominated subset of  $P(t) \cup G$ 
  return result
end function

```

Fig. 1. Global pseudocode of (μ, λ) -MOES.

selected parent should not be dominated by the former in order to accelerate prepotency), and λ offspring will be produced within a loop in exploration stage while the archive increase scheme is performed simultaneously. After the current intermediate population H has been filled with the λ offspring, the archive decrease scheme starts, and subsequently, the individuals of H are sorted in descending order of fitness assigned so as to select the first μ individuals with higher fitness into the next population. These sequences of instructions are repeated until terminal criterion is satisfied. In the following subsection, we shall present respective particular description.

3.1. Code representation

In (μ, λ) -MOES, real number representation is used instead of binary string implementation. Since many problems in the real-world are expressed in real variables, faster computation can be obtained without conversions between different representations. It is characterized that (μ, λ) -MOES borrows a ternary group representation from [23], where an individual was denoted by $(\mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\theta})$ group. That is, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the original decision vector, corresponding to a point in decision space, and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n)$ is the standard deviation

used to instruct mutation, and $\theta = (\theta_{1,2}, \theta_{1,3}, \dots, \theta_{1,n}, \theta_{2,3}, \dots, \theta_{(n-1),n})$ is the rotation angle used to change the orientation of the mutation associated with all possible pairs of decision vectors. Both σ and θ are called strategy parameters. It is very similar to a hillclimber algorithm with self-adaptive step σ and angle θ . In other words, all components are submitted to evolutionary process by applying exploration operator and exploitation scheme on them. Thus, an appropriate adjustment and diversity of parameters can be automatically modified in demand, and this modification just corresponds to the local regulation.

3.2. Exploration operator

In (μ, λ) -MOES, exploration operator includes recombination and mutation. Basically, recombination works choosing parents with uniform probability, and, characteristics of parents are mixed to create one offspring. The discrete recombination that is commonly used in ES [24] and has also produced good results with real-coded MOEAs [25] is employed in (μ, λ) -MOES. Its implementation is that each component of offspring inherits from one of the two pre-selected parents randomly. If parent-1 is $(x^{(1)}, \sigma^{(1)}, \theta^{(1)})$ and parent-2 $(x^{(2)}, \sigma^{(2)}, \theta^{(2)})$, we produce one offspring as

$$(x', \sigma', \theta') = \left((x_1^{q_1}, \dots, x_n^{q_n}), (\sigma_1^{q_1}, \dots, \sigma_n^{q_n}), (\theta_{1,2}^{q_1}, \theta_{1,3}^{q_1}, \dots, \theta_{1,n}^{q_1}, \theta_{2,3}^{q_1}, \dots, \theta_{(n-1),n}^{q_1}) \right), \quad (2)$$

where, (x', σ', θ') is the offspring and q_i is chosen between 1 and 2 with probability 0.5. Notice that recombination is performed independently on decision vectors as well as on strategy parameters.

Generally, mutation is more emphasized than recombination in ES. It is typically implemented as Gaussian distribution around the generated individual being mutated. A new individual is produced via Gaussian mutation as

$$\begin{aligned} \sigma_i'' &= \sigma_i' \exp(\tau' N(0, 1) + \tau N_i(0, 1)); \\ \theta_{i,j}'' &= \theta_{i,j}' + \gamma N_{i,j}(0, 1), \quad \forall i, j \in \{1, \dots, n\}, \quad j > i; \\ x'' &= x' + N(0, \text{cov}(\sigma'', \theta'')), \end{aligned} \quad (3)$$

where, $N(0, 1)$ denotes to generate a random scalar that is characterized by Gaussian distribution with zero mean and deviation one, and additional subscripts of $N(0, 1)$ denote independent regeneration for each element respectively. $N(0, \text{cov})$ denotes a vector function which returns a random vector distribution that is Gaussian distributed with zero mean and covariance matrix cov^{-1} . The rotations along with the variances are used to fill the covariance matrix and the variances form the diagonal of the covariance matrix as

$$c_{i,i} = \sigma_i', \quad c_{i,j} = \theta_{i,j}'; \quad \forall i, j \in \{1, \dots, n\}, \quad j > i. \quad (4)$$

In order to limit the rotation angles within the interval $[-\pi, \pi]$, they are implemented using sine and cosine functions. If mutation moves rotation outside of this range, it is circularly remapped back into the range $[-\pi, \pi]$. τ' , τ and γ are parameters, which are suggested as the following values in [26]:

$$\tau' \propto (\sqrt{2n})^{-1}, \quad \tau \propto (\sqrt{2\sqrt{n}})^{-1}, \quad \gamma \approx 0.0873. \quad (5)$$

Although the significant behavior of Gaussian mutation on proximity exploration, a new type of mutation using Cauchy distributed random was inspired in [15,16,18], which all claimed that Cauchy mutation outperformed its Gaussian counterpart on diversity preservation and search efficiency. Cauchy probability density function is

$$f(x) = \frac{\beta}{\pi(\beta^2 + (x - \alpha)^2)}, \quad \alpha > 0, \quad \beta > 0, \quad -\infty < x < \infty, \quad (6)$$

represented as $C(\alpha, \beta)$ where α and β are two parameters. It is plotted in Fig. 2 in company with Gaussian distribution by applying the same parameters. We can investigate that Cauchy distribution is symmetrical and long-tailed, in the other word, it has a lower extremum in middle and a slower horizontal decline than Gaussian one and the horizontal decline is getting smaller as it departs from the middle. Therefore, probabilistically speaking, Cauchy distribution is more expanded, and it allows larger mutations and in this way producing more diversified individuals and covering more major space. On the contrary, Gaussian mutation is accomplished in accurate search of its nearest space for

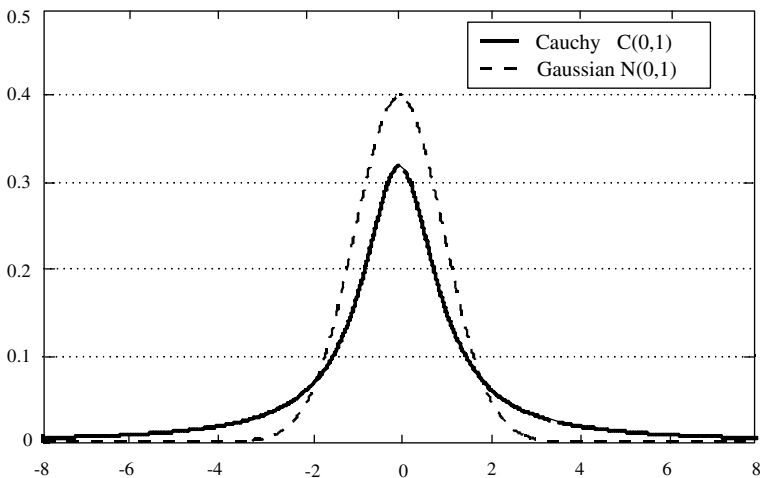


Fig. 2. Cauchy and Gaussian distributions (with the same parameters).

proximity exploration. Now we may produce a new individual via Cauchy mutation as

$$\begin{aligned}\sigma''_i &= \sigma'_i \exp(\tau' C(0, 1) + \tau C_i(0, 1)); \\ \theta''_{i,j} &= \theta'_{i,j} + \gamma C_{i,j}(0, 1), \quad \forall i, j \in \{1, \dots, n\}, \quad j > i; \\ x'' &= x' + C(0, \text{cov}(\sigma'', \theta'')), \end{aligned} \quad (7)$$

where $C(0, \cdot)$ denotes to yield random scalar or vector submitted to Cauchy distribution with respective parameters. The similar denotations are omitted here.

It sounds possible to provide higher performance on balancing proximity and diversity, if we take measures to develop the respective strong point from Gaussian mutation and Cauchy mutation. Fig. 3 shows the measures that will be taken to realize our idea. Therefore, coupling the recombination and mutation, (μ, λ) -MOES produces an offspring completely as pseudocode in Fig. 4 describes.

3.3. Fitness assignment scheme

Since fitness assignment scheme is a crucial company to the complete non-dominance selection, we can use fitness value as an indicator to distinguish dominated and nondominated individuals. It should be devised meeting the following requirements:

- The dominated individuals must share lowest fitness than nondominated ones due to their absence of qualification for survival into the next generation.
- The fitness of nondominated individuals must indicate their properties; so much better individuals of them can be filtered into the next step after competition.

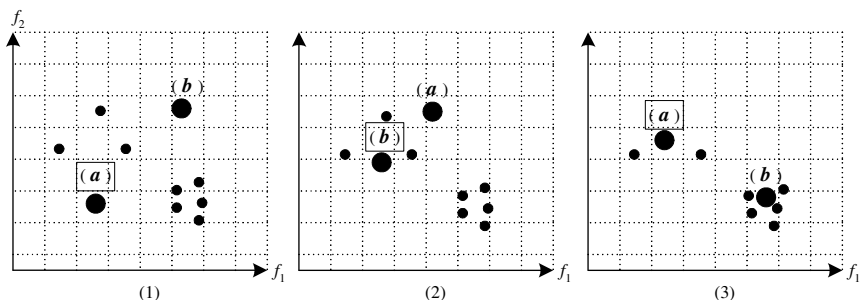


Fig. 3. Illustration of measures for exploration operator: (1) a dominates b , so a is chosen as the current offspring; (2) b dominates a , so b is chosen; (3) a and b are incomparable but b is located in a more crowded region, so a is chosen.

```

function offspring = ProduceOffspring( parent1, parent2, current_population, external_archive )


---


begin
  %% calculated via Eq.(2) and Eq.(3)
  produce one offspring off_Gauss with discrete recombination and Gaussian mutation
  %% calculated via Eq.(2) Eq.(7)
  produce one offspring off_Cauchy with discrete recombination and Cauchy mutation
  %% compare two offspring and decide which is returned
  if ( off_Gauss dominates off_Cauchy )
    offspring = off_Gauss
  else if ( off_Cauchy dominates off_Gauss )
    offspring = off_Cauchy
  else %% two offspring are incomparable, so the crowded location is applied
    if ( off_Gauss is in a less crowded region of current_population  $\cup$ 
      external_archive than off_Cauchy ) %% calculated via Eq.(8)
      offspring = off_Gauss
    else
      offspring = off_Cauchy
    end if
  end if
  return offspring
end function

```

Fig. 4. Pseudocode of exploration operator.

- The individuals far away from the center of the current nondominated set must be assigned a higher fitness so that they will share a higher probability of survival, which contributes to the diversity of population and reducing the risk of premature convergence.

Therefore, for all dominated individual of the intermediate population or the external archive, we assigned their fitness the same zero; but for each non-dominated individual, the average Euclid distance in objective space between it and other m individuals that are its nearest neighbors is equal to evaluate its contribution to diversity preservation, defined as

$$F(j) = \frac{\min \sum_{l=1}^m \{\|\mathbf{x} - \mathbf{x}_l\|_{\text{obj}}\}}{m}, \quad \mathbf{x}, \mathbf{x}_l \in H, \quad (8)$$

where, j denotes nondominated individual, \mathbf{x} is decision vector of j and function $\|\cdot\|_{\text{obj}}$ calculates the Euclid distance in objective space between two individuals. Fig. 5 describes the pseudocode of our fitness assignment scheme.

3.4. Archive increase scheme

In general, if an MOEA has an external archive with fixed size, a replacement scheme is always applied. In this scheme, in order to keep the archive size unchanged, a new-added individual will replace one members of archive if it is considered to be better than the other individual. However, this scheme brings

```

function AssignFitness( population )


---


begin
  for ( each individual m of population )
    if ( m is nondominated ) %% nondominated individuals
      assign fitness of m via Eq.(8)
    else %% dominated individuals
      assign fitness of m zero
    end if
  end for
end function

```

Fig. 5. Pseudocode of fitness assignment scheme.

up a problem that some of the replaced individuals may still be very valuable and have not been well explored or exploited yet before they are replaced. Although some approaches such as *amalgamation strategy* in [7] and *preselection scheme* in [14] have been introduced, the problem caused by the replacement scheme is still not completely resolved. Therefore, (μ, λ) -MOES applies two independent schemes—archive increase scheme and archive decrease scheme. The first scheme only focuses on pure population increment and ensures that each individual survives enough generations so that it can contribute its valuable schemas. Meanwhile an archive decrease scheme is also enforced to prevent the population size from growing excessively. The second scheme will be discussed in the next subsection.

Because the purpose of the external archive we used is to store elitist individuals as well as relatively better individuals that contain advance responding to their parents, the dominated individuals that cannot survive in the next population have opportunity to survive in external archive. This eliminates the disadvantage in [21] where dominated individuals are completely discarded and none of their useful information is under consideration. Therefore, we adopt the *diffusion scheme* in particle swarm optimization (PSO) [27] to guide archive increase scheme—an individual shares its information with the leading individuals in order to locate its moving direction. This idea is inspired by its significant performance. So, in (μ, λ) -MOES, if a newborn offspring is better than one of its parents in proximity or diversity (indicated by dominance or location of less crowded region [28]), it has priority to be added into the external archive; otherwise, it will be drastically discarded. Since the property of *age* is a crucial factor for the archive decrease scheme, we initialize age of new individual as one when it is added for the first time. As a result, this scheme will guarantee that a new-added individual in the external archive will have higher proximity or diversity than at least one of its parents, which helps (μ, λ) -MOES provide both elitist individuals as well as relatively better individuals participating in exploration and exploitation so as to cover all the unexplored and potential regions in objective space. Fig. 6 shows the pseudocode of archive increase scheme.

```

function IncreaseArchive( offspring, parent1, parent2, external_archive, size_minimal)


---


begin
  if (size of external_archive is less than size_minimal )
    add offspring into external_archive
    set age of offspring one  %% age of offspring is initially one
  else
    if ( offspring dominates one of parent1 and parent2 ) %% dominance is applied first
      add offspring into external_archive
      set age of offspring one
    else  %% otherwise, crowded location is applied later
      if( offspring is in a less crowded region of external_archive than parent1 or parent2 )
        add offspring into external_archive
        set age of offspring one
      end if
    end if
  end if
end function

```

Fig. 6. Pseudocode of archive increase scheme.

3.5. Archive decrease scheme

As discussed in the previous subsection, an archive decrease scheme is necessary to prevent the archive size from growing without bound. In (μ, λ) -MOES, whether an individual will be eliminated from the external archive or not depends on its *age* and *fitness* defined in Section 3.3. As far as age is concerned, the age of new-added individual is initialized to be one, and its age will grow generation by generation as long as it survives in the external archive.

To ensure that an eliminated individual has a lowest fitness value and has been adequately explored and exploited for a certain generation, (μ, λ) -MOES eliminates individuals from the external archive in each generation according to the following principles:

- In order to keep balance of archive size in probability sense, there are λ times (opportunities) to eliminate relatively worse individuals in each decrease scheme because λ offspring have been produced in exploration stage and correspondingly there are λ times (opportunities) to add relatively better individuals in each increase scheme. This means that there are at most λ individuals can be eliminated if we perform the decrease scheme once. Fig. 7 shows the different results from decrease scheme with different λ value.
- We only eliminate the individuals with lowest fitness (*zero* value) and age larger than the prespecified age threshold *age_min* from the external archive. For the example in Fig. 7, the external archive is different after decrease due to different age threshold under situation when λ is 4. Fig. 8 shows the pseudocode of archive decrease scheme.

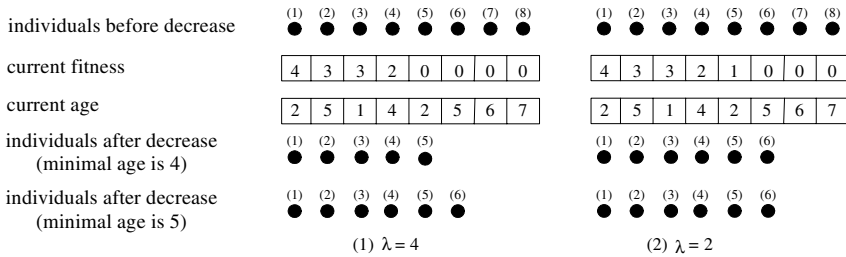


Fig. 7. Illustration of archive decrease scheme: (1) $\lambda = 4$ means there are four times to remove worse individuals at most, so the last three individuals are removed where minimal age is 4; (2) $\lambda = 2$ means only two worse individuals are removed and there even exists an individual whose age is larger than the age threshold 4.

```

function DecreaseArchive ( external_archive, size_minimal, age_minimal, offspring_number )


---


begin
  if (size of external_archive is more than size_minimal )
    for ( each individual m of external_archive )
      increase age of m by one
    end for
    set elimination counter zero:  $n = 0$ 
    AssignFitness ( external_archive ) %% assign fitness to direct remove
    do %% offspring_number trials are performed to eliminate dominated individuals from external_archive
      if ( these exist an individual m1 with zero fitness and its age is larger than age_minimal )
        eliminate m1 from external_archive
        decrease size of external_archive by one
      end if
      increase elimination counter by one:  $n \leftarrow n + 1$ 
    while (  $n < \text{offspring\_number} \wedge$  size of external_archive is more than size_minimal )
  end if
end function

```

Fig. 8. Pseudocode of archive decrease scheme.

4. Comparison study

In order to validate the proposed (μ, λ) -MOES and quantitatively compare its performance with other advanced MOEAs, graphical presentation, statistical analysis of the experimental results and associated observations are presented in this section. In our comparison study, three prominent benchmark test functions with distinct Pareto optimal front are selected from [30], which brought forward various benchmark functions for testing MOEAs. Table 1 describes them in characteristic details.

Table 1
Three test functions

No.	Function	n	Boundaries	Characteristics of PF_{true}
T_1	$f_1(\mathbf{x}) = x_1, f_2(\mathbf{x}) = g \cdot (1 - \sqrt{f_1/g})$ $g = 1 + 9 \cdot (\sum_{i=2}^n x_i)/(n-1)$	30	$[0, 1]^n$	Continuous Convex
T_2	$f_1(\mathbf{x}) = x_1, f_2(\mathbf{x}) = g \cdot (1 - f_1/g)^2$ $g = 1 + 9 \cdot (\sum_{i=2}^n x_i)/(n-1)$	30	$[0, 1]^n$	Continuous Concave
T_3	$f_1(\mathbf{x}) = x_1, f_2(\mathbf{x}) = g \cdot (1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1))$ $g = 1 + 9 \cdot (\sum_{i=2}^n x_i)/(n-1)$	30	$[0, 1]^n$	Discrete

For all functions PF_{true} is formed with $g = 1$. n is the number of decision vector.

SPEA-II (strength Pareto evolutionary algorithm II) [8] is an enhanced version of SPEA [7]. In SPEA-II, instead of calculating standard Pareto rank, each individual in both the main population and elitist archive is assigned a strength value, which incorporates both dominance and density information. On the basis of the strength value, the final rank value is determined by the summation of the strengths of the points that dominate the current individual. Meanwhile, a density estimation method is applied to obtain the density value of each individual. The final fitness is the sum of rank and density values. Additionally, a truncation method is used in the elitist archive to maintain a constant number of individuals in the archive. SPEA-II shows better performances than SPEA over all test functions in the experimental results therein [8]. Nevertheless, SPEA-II also used an estimated population size and a fixed elitist archive. Because SPEA-II was claimed to outperform other well-known MOEAs, it is representatively selected into this comparison study with our algorithm although there are also other Pareto-based MOEAs that may be under consideration for the comparison. Table 2 lists the General parameters set-

Table 2
General parameters setting of SPEA-II and (μ, λ) -MOES for all test functions

SPEA-II		(μ, λ) -MOES	
Population size (N)	80	Population size (μ)	20
External archive size (N')	30	Offspring per generation (λ)	60
Crossover probability	0.7	Minimum age threshold (age_min)	10
Mutation probability	0.01	Minimum size of external archive ($size_min$)	20
Maximum generation	10,000	Maximum generation	10,000
Offspring per generation	20	Parameter (τ')	0.1581
Chromosome length	$15 \times dec_num^a$	Parameter (τ)	0.3344
		Parameter (γ)	0.0873

^a dec_num represents the number of decision variables.

ting of SPEA-II and (μ, λ) -MOES for all runs, referred to literatures, e.g., [7,8,10,26], etc.

Since the comparison focus is on minimizing the proximity of PF_{final} as well as on maximizing the diversity of PF_{final} , we have to consider both the online performance and offline performance of two MOEAs. Thus, the final population and external archive containing the nondominated set regarding all individuals generated over all generation was taken as the output of an optimization run. Altogether 50 independent runs were performed per MOEA and test function in order to restrict the influence of random effects. Another randomly created initial population was taken each time, and for each test function both MOEAs are operated on the same initial populations. We use two methods in the following: (1) graphical presentation for visual inspection and (2) the performance metric to show quantitative inspection.

4.1. Graphical presentation

The first method is the graphical presentation. We unify the outcomes of each test function from each algorithm, and all the nondominated individuals are plotted as shown in Figs. 9 and 10 in company with its *true* Pareto optimal front (named PF_{true} in this section).

It is clear that (μ, λ) -MOES outperforms SPEA-II on all three test functions. For function T_1 , whose PF_{true} is convex drawn in the left of Fig. 9, the solutions of (μ, λ) -MOES are much closer to the front than SPEA-II in the middle, but almost cover each other in the tail. Moreover, the solution distribution also varied. Seventy percent solutions of SPEA-II are crowded within 30% region of the tail, which goes against the diversity preservation for decision making on middle compromise solutions. In contrast with SPEA-II, (μ, λ) -MOES distributes its solutions symmetrically in both middle and tail without obvious difference. Therefore, the solutions' proximity and diversity from (μ, λ) -MOES are

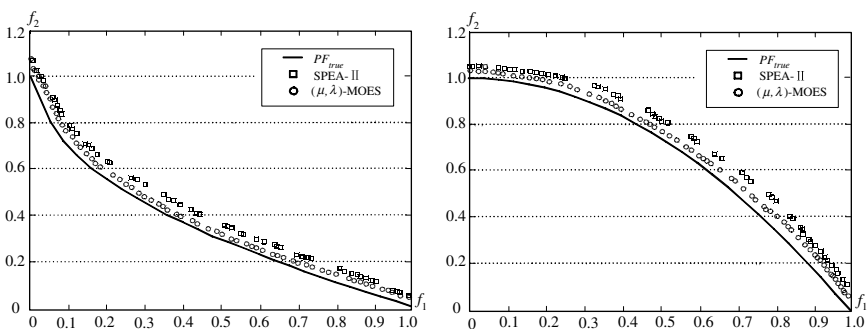


Fig. 9. Graphical presentation based on test function T_1 (left) and T_2 (right).

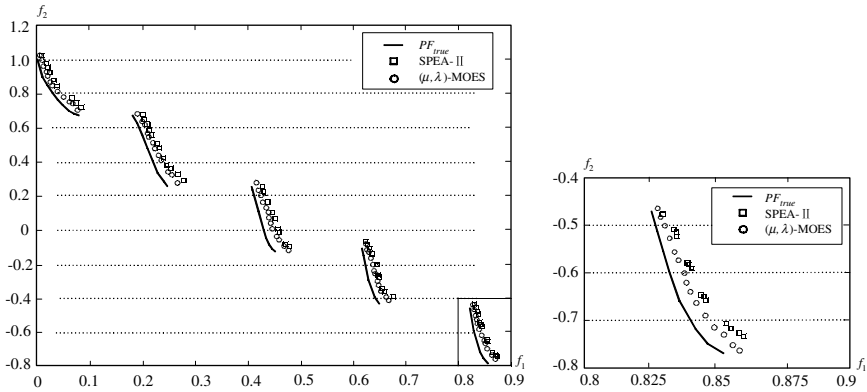


Fig. 10. Graphical presentation based on test function T_3 (left) and enlarged rectangle from a specified region (right).

more satisfactory than SPEA-II. From the right of Fig. 9, the similar phenomena also occur in T_2 , a concave function. The results of function T_3 from either (μ, λ) -MOES or SPEA-II are a bit close. Both show less perfect than what has been achieved in continuous functions T_1 and T_2 . To show the discrepancy, we enlarged the region of closest proximity between 0.8 and 0.9 of f_1 and displayed it on the right of Fig. 10. It is clear that (μ, λ) -MOES distributes their solutions closer to PF_{true} and more diversified than SPEA-II does.

From visual inspection, our result is that (μ, λ) -MOES remarkably outperforms SPEA-II for continuous functions on proximity and diversity. As far as discrete function is concerned, the predominance still exists although it decreases to some extent.

4.2. Performance metrics

Since no single metric can represent total performance of an MOEA, a series of appropriate metrics is chosen to measure the performance. For this reason, we use the first metric of general distance (GD) to show how far PF_{final} are away from PF_{true} , where is known in *a priori* for the given test functions. This metric was originally introduced in [29], where the final individual distance is calculated by

$$GD \triangleq \frac{1}{N} \left(\sum_{i=1}^N d_i^p \right)^{1/p}, \quad (9)$$

where N is the number of nondominated individuals in PF_{final} , $p = 2$, and d_i is the Euclidean distance in objective space between each of these individuals and a point on PF_{true} that is the closest to it. A value of zero for GD indicates the ideal convergence that $PF_{final} = PF_{true}$, so a smaller value of GD is preferable.

Corresponding to the diversity aspect, another desire to measure the spread of nondominated individuals throughout PF_{final} appears. We adopt such a metric of spacing measuring (SP) the range (distance) variance of neighboring individuals in PF_{final} , which is defined as

$$SP \triangleq \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - d_i)^2}, \quad (10)$$

where $d_i = \min_j (|f_1^i(\mathbf{x}) - f_1^j(\mathbf{x})| + |f_2^i(\mathbf{x}) - f_2^j(\mathbf{x})|)$, \bar{d} is the mean of all d_i and N is the number of nondominated individuals in PF_{final} . A value of zero for this metric indicates the ideal diversity that all member of PF_{final} are equidistantly and uniformly spaced. A smaller value of SP is preferable.

Moreover, in order to compare the dominance relationship between two PF_{final} from two different MOEAs, the coverage of two sets (C value) [9] is also included to show how PF_{final} of one algorithm dominates that of another. The function C maps the ordered pair (X_i, X_j) to interval $[0, 1]$, defined as:

$$C(X_i, X_j) \triangleq \frac{|\{y \in X_j \wedge (\exists x \in X_i : x \succ y \vee x = y)\}|}{|X_j|}, \quad (11)$$

where X_i and X_j denote PF_{final} from algorithm i and algorithm j , respectively. The value $C(X_i, X_j) = 1$ means that all individuals in X_j are dominated by, or equal to, individuals in X_i . The opposite, $C(X_i, X_j) = 0$, represents the situation when none of the points in X_j are covered by the set X_i . Note that both $C(X_i, X_j)$ and $C(X_j, X_i)$ have to be considered independently since they express distinct meanings.

Therefore, three metrics represent quantitative measures that describe the quality of PF_{final} of selected MOEAs. The first two metrics have the advantage that each MOEA can be evaluated independently and the last one obtain the advantage that it can be used to show that the outcomes of one algorithm dominate the outcomes of another algorithm, although it does not tell how much better it is. Here, *box plots* [31] are used to visualize the distribution of these samples of three metrics from 50 independent runs. Each box plot represents the distribution of a sample set where a horizontal line within the box encodes the median, while the upper and lower ends of the box are the upper and lower quartiles. The appendages illustrate the spread and shape of distribution, and dots represent the outside values.

The box plots based on the results concerning GD and SP are shown in Fig. 11 respectively, and the dominance relationship based on C values of two algorithms is depicted in Fig. 12. It is revealed that (μ, λ) -MOES is completely superior to SPEA-II on metrics of GD and SP from Fig. 11, and the predominance descends from continuous functions to discrete functions as the graphical presentation has discovered. The box plots in Fig. 12 demonstrate that (μ, λ) -

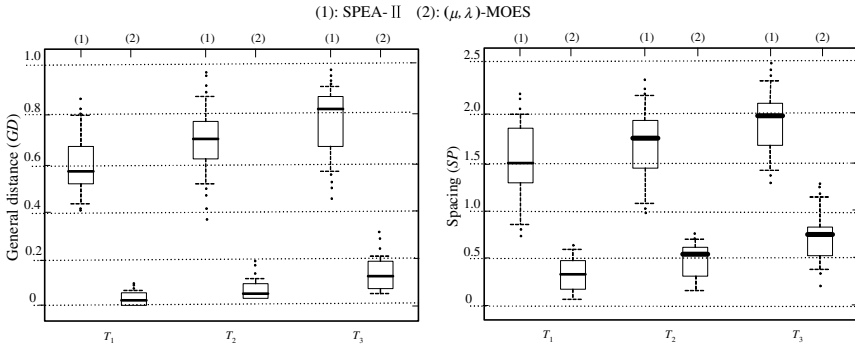


Fig. 11. Box plots based on metrics of GD (left) and SP (right) of all test functions from two MOEAs.

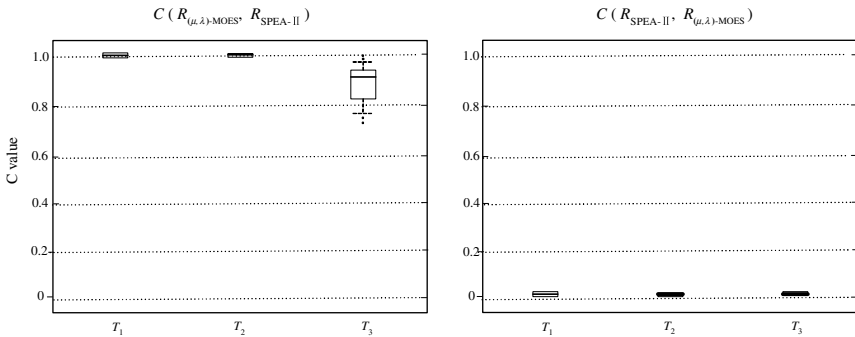


Fig. 12. Box plots based on the metric of C value of all test functions from two MOEAs.

MOES covers SPEA-II except for a few occasions. On the other hand, the results from SPEA-II never covers any of (μ, λ) -MOES' solutions. Consequently, our proposed algorithm can be claimed to outperform SPEA-II to some extent in our comparison study.

5. Conclusions

In this paper, we have proposed a novel (μ, λ) -MOES. It can be characterized as: (1) effectively improving the competence of exploration operator to provide new nondominated individuals and diversify them well by using a combinatorial operator from discrete recombination, Gaussian mutation and Cauchy mutation; (2) powerfully ensuring maximal pressure on proximity selection

by using a complete nondominance selection and ensuring maximal diversity preservation by using a fitness assignment determined by dominance and population diversity information; (3) appropriately handling the relationship between nondominated and dominated individuals by using a dynamic external archive to store elitist individuals as well as relatively better individuals; (4) adaptively increasing or decreasing the external archive based on the current fitness and age status and (5) converging to a near-optimal, well-extended and uniformly diversified Pareto optimal front. From comparison study including graphical presentation and performance metric inspection, (μ, λ) -MOES has shown its potential in producing statistically superior results to SPEA on three prominent benchmark test functions to some extent. So we suggest that (μ, λ) -MOES be a potential candidate in solving more complicated problems.

However, as the benchmark test functions used in this paper are still far from covering all the challenging characteristics of MOPs, a more profound study in dealing with real-world MOPs is absolutely necessary in the future work. Additionally, there is no special design built in (μ, λ) -MOES to handle strong constraints. In near future, it will be revised to deal with those MOPs with strong and complicated constraints.

Acknowledgments

The authors gratefully acknowledge the support of the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB415206. The authors wish to express their most sincere appreciation to Prof. Lei Sheng-long, who has read the manuscript carefully and given some valuable suggestions. Tremendous thanks also are owed to Dr. Liu Bin-jun for helping computational programs.

References

- [1] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation* 3 (1) (1995) 1–16.
- [2] J.L. Cohon, D.H. Marks, A review and evaluation of multiobjective programming techniques, *Water Resource Research* 11 (2) (1975) 208–220.
- [3] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: J.J. Grefenstette, et al. (Eds.), *Genetic Algorithms and Their Applications*, Proceedings of the 1st International Conference on Genetic Algorithms, Lawrence Erlbaum, Mahwah, NJ, 1985, pp. 93–100.
- [4] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: J.J. Grefenstette et al. (Eds.), *IEEE World Congress on Computational Intelligence*, Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1994, pp. 82–87.

- [5] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation* 3 (2) (1994) 221–248.
- [6] K. Deb, S. Agrawal, A. Pratab et al., A fast elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II, *IEEE Trans. on Evolutionary Computation* 6 (4) (2002) 182–197.
- [7] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *Evolutionary Computation* 3 (1) (1999) 257–271.
- [8] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: *Evolutionary methods for design, optimization and control with applications to industrial problems*, EUROGEN 2001, Athens, Greece, 2001.
- [9] J.D. Knowles, D.W. Corne, The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization, in: *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 1999, pp. 98–105.
- [10] J.D. Knowles, D.W. Corne, M-PAES: a memetic algorithm for multiobjective optimization, in: *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 2000, pp. 325–332.
- [11] X.X. Cui, M. Li, T.J. Fang, A multiobjective concordance evolutionary algorithm, *Chinese Journal Computers* 24 (9) (2001) 979–984 (in Chinese).
- [12] S.W. Mahfoud, Genetic drift in sharing methods, in: J.J. Grefenstette, et al. (Eds.), *IEEE World Congress on Computational Intelligence, Proceedings of the 1st IEEE Conference on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 1994, pp. 67–72.
- [13] H.P. Schwefel, T. Back, A survey of evolutionary strategies, in: R. Belew (Ed.), *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, CA, 1991, pp. 92–99.
- [14] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, *Evolutionary Computation* 7 (2003) 174–188.
- [15] X. Yao, Y. Liu, Fast evolutionary programming, in: *Proceedings of the 5th Annual Conference on Evolutionary Programming*, The MIT Press, Cambridge, MA, 1996, pp. 451–460.
- [16] J.R. Gomes, O.R. Saavedra, A Cauchy-based evolution strategy for solving the reactive power dispatch problem, *Electrical Power and Energy Systems* 24 (2002) 277–283.
- [17] J.H. Zhang, X.H. Xu, An efficient evolutionary programming algorithm, *Computers and Operations Research* 26 (1999) 645–663.
- [18] L.B. Shi, G.Y. Xu, Self-adaptive evolutionary programming and its application to multiobjective optimal operation of power systems, *Electric Power Systems Research* 57 (2001) 181–187.
- [19] P.A.N. Bosman, D. Thierens, Multiobjective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms, *International Journal of Approximate Reasoning* 31 (2002) 259–289.
- [20] K.C. Tan, T. Lee, E. Khor, Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization, *Evolutionary Computation* 5 (12) (2001) 565–588.
- [21] R. Sarker, K.-H. Liang, C. Newton, A new multiobjective evolutionary algorithm, *European Journal of Operational Research* 140 (2002) 12–23.
- [22] M. Laumanns, A unified model for multi-objective evolutionary algorithms with elitism, in: *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 2000, pp. 46–53.
- [23] H.P. Schwefel, *Numerical Optimization for Computer Models*, John Wiley, Chichester, UK, 1981.
- [24] T. Bäck, H.P. Schwefel, An overview of evolutionary algorithm for parameter optimization, *Evolutionary Computation* 1 (2) (1993) 1–23.

- [25] H. Muhlenbein, D. Schlierkamp Voosen, Predictive models for the breeder genetic algorithm: continuous parameter optimization, *Evolutionary Computation* 1 (2) (1993) 25–49.
- [26] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
- [27] X. Hu, R.C. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: *Proceedings of the 2002 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 2002, pp. 1677–1681.
- [28] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, *Evolutionary Computation* 8 (2) (2000) 149–172.
- [29] A. Van David Veldhuizen, B. Lamont Gary, On measuring multiobjective evolutionary algorithm performance, in: *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 2000, pp. 204–211.
- [30] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, *Evolutionary Computation* 8 (2) (2000) 173–195.
- [31] J.M. Chambers, W.S. Cleveland, B. Kleiner et al., *Graphical Methods for Data Analysis*, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1983.