Dynamic Archive Evolution Strategy for Multiobjective Optimization

YANG Shu Min[1], SHAO Dong Guo[1], LUO Yang Jie[2]

[1] State Key Laboratory of Water Resource & Hydropower Engineering Science, Wuhan University, Wuhan 430072, Hubei, P.R.China
[2] College of Urban & Environmental Sciences, Northeast Normal University, Changchun 130024, Jilin, P.R.China

Abstract. This paper proposes a new multiobjective evolutionary approach—the dynamic archive evolution strategy (DAES) to investigate the adaptive balance between proximity and diversity. In DAES, a novel dynamic external archive is proposed to store elitist individuals as well as relatively better individuals through archive increase scheme and archive decrease scheme. Additionally, a combinatorial operator that inherits merits from Gaussian mutation of proximity exploration and Cauchy mutation of diversity preservation is elaborately devised. Meanwhile, a complete nondominance selection ensures maximal pressure of proximity exploitation while a corresponding fitness assignment ensures the similar pressure of diversity preservation. By graphical presentation and performance metrics on three prominent benchmark functions, DAES is found to outperform three state-of-the-art multiobjective evolutionary algorithms to some extent in terms of finding a near-optimal, well-extended and uniformly diversified Pareto optimal front.

1. Introduction

A formal notion of multiobjective optimization is given by Fonseca and Fleming in [1]. Without loss of generality, consider the following multiobjective optimization with n decision variables x and m (m>1) objectives y:

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

where $\boldsymbol{x}$ is called decision vector, $\boldsymbol{X}$ decision space, $\boldsymbol{y}$ objective vector and $\boldsymbol{Y}$ objective space; $\boldsymbol{f}$ defines the mapping function. The scenario considered in this paper involves an arbitrary optimization with objectives, which are all to be minimized and all equally important. It means that the objectives cannot be combined into a single scalar objective to be optimized, so the sets of solutions exist such that each solution in this set is equally preferable. The following four concepts are of importance:

1. **Pareto dominance:** A solution $\boldsymbol{x}^0$ is said to *dominate* (*Pareto optimal*) another solution $\boldsymbol{x}^1$ (denoted $\boldsymbol{x}^0 \succ \boldsymbol{x}^1$) if and only if:
   $\forall i \in \{1,\cdots,m\} : f_i(\boldsymbol{x}^0) \le f_i(\boldsymbol{x}^1) \wedge (\exists k \in \{1,\cdots,m\} : f_k(\boldsymbol{x}^0) < f_k(\boldsymbol{x}^1))$.

2. **Pareto optimal:** A solution $\boldsymbol{x}^0$ is said to be *nondominated* (*Pareto optimal*) if and only if: $\neg\exists \boldsymbol{x}^1 \in \boldsymbol{X} : \boldsymbol{x}^1 \succ \boldsymbol{x}^0$.
3. **Pareto optimal set:** The set P_S of all Pareto optimal solutions: $P_S = \{\boldsymbol{x}^0 \mid \neg\exists \boldsymbol{x}^1 \in \boldsymbol{X} : \boldsymbol{x}^1 \succ \boldsymbol{x}^0\}$.
4. **Pareto optimal front:** The set P_F of all objective function values corresponding to the solutions in P_S: $P_F = \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})) \mid \boldsymbol{x} \in P_S\}$.

The optimal result for such multiobjective optimization is no other than the Pareto optimal set P_S. However, the size of this set may be infinite, and 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 multiobjective evolutionary algorithms (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. Because we only consider the subset of all the final nondominated individuals resulted from a MOEA, we call such subset an *approximation set* and denote it by S, 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 the decision space is also a good diversity of the individuals in the objective space correspondingly. However, it is common practice to search for a good diversity of the individuals in the 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.

During the past decade, various MOEAs have been proposed and applied [1]. A representative collection of these influential algorithms includes the Nondominated Sorting Genetic Algorithm (NSGA) and NSGA2 by Srinivas and Deb et al [2] [3], the Strength Pareto Evolutionary Algorithm (SPEA) and SPEA2 by Zitzler et al [4] [5], the Pareto Archived Evolution Strategy (PAES) and the memetic PAES (M-PAES)by Knowles and Corne [6] [7] etc. Although these MOEAs differ from each other, they share the common purpose — searching for a near-optimal, well-extended and uniformly diversified PF_{final} for a given multiobjective optimization. However, this ultimate goal is far from being accomplished by the existing MOEAs as documented in the literature, e.g., [1],[5]. In one respect, most of multiobjective optimizations are very complicated and have their own inherent characters and variabilities, so computational resources are required to be homogenously 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 [8], which results in the loss of diversity. This highlighted the hot issue how to improve the algorithm's robustness and how to balance proximity and diversity during searching process. However, the latter issue we considered in this paper

has not been attended yet. Although in some elitism-based MOEAs, several techniques have been adopted such as crowded comparison [3], amalgamation strategy [4], archive truncation [5] and preselection scheme [9] etc, the manipulation of balance choice remains manual and presetting. Therefore, either the pressure for proximity or for diversity should be autonomously regulated by the current and historical search.

The present work proposed a novel dynamic archive evolution strategy (DAES) inherited from evolution strategy (ES) [10] to investigate the adaptive balance between the proximity and diversity. A special aspect of importance in which ES differs from most other EAs is that it has a long self-adaptive mechanism by usually including strategy parameters that can adaptively guide the search to explore and exploit the local and global topology of the decision space. DAES hold such mechanism too and have developed some new features. The prominent innovation of DAES is the dynamic external archive with its form, purpose and managing scheme. Other techniques served for exploration and exploitation have also been designed and work cooperatively to guarantee satisfactory results.

The remainder of this paper is organized as follows. In section 2, we introduce some basic rationale used to balance proximity and diversity in exploration stages and exploitation stages respectively by summarizing some prominent state-of-the-art MOEAs. Section 3 describes DAES arithmetic in detail. The empirical results and comparisons between DAES and SPEA2 [5], NSGA2 [3], M-PAES [7] on three prominent benchmark functions are presented in Section 4. Eventually, we conclude the paper with some remarks and future researches in the last section.

2 How to balance proximity and diversity in DAES

2.1 Role of the external archive

In order to obtain new and diverse nondominated individuals, especially when the set of nondominated individuals have approached 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 some MOEAs [4]. Alternatively, an external archive may be more commonly used that contains nondominated individuals, and the current population and the external archive are separated and managed by exchanging individuals between them. It also helps allow preserving the good individuals that are hard to be generated in the exploration stage.

In the 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 still hold some useful information, they should be wisely kept for a certain generation for further exploration and exploitation. However, if under the complete nondominance selection we intend to apply in the exploitation stage, they will definitely lose opportunity to survive into the next population due to their absence of qualification to be stored in the archive. So we need changing the role of the archive by some means. Furthermore, the archive size is highly sensitive to manage selected individuals. If the size is too small, there will not be enough schemas to exploit, resulting in a premature or non-uniformly PF_{final}. Otherwise, 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 autonomously by the online characteristics of the proximity and diversity information of the archive will be more efficient and effective than a constant size. As a result, the archive should be endued with the responsibility of saving elitist individuals as well as relatively better individuals and a dynamic size of it is preferable. The archive increase scheme and the archive decrease scheme are designed concurrently in order to manage the archive in the probability sense.

2.2 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 P_F. Based on an appropriate selection of parents, a competent exploration operator is expected to be able to produce offspring in which good features of the selected parents are inherited, which does not differ much from the rationale in single objective EAs essentially. Since excessive proximity without adequate diversity will inevitably lead to premature convergence or local optimum while excessive diversity without adequate proximity will always confuse the search guide into stagnancy, it is necessary to deal with them equally and adaptively.

An interesting and relatively new field is the combinatorial operator, which attempts to model the regularities of parent structure by means of combination of several operators. In this way different operators are cooperated with to develop each merit and eliminate each weakness, where heuristic method is also applied. An example is taken in the fastEP 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 the next population [11]. Algorithms that use the similar approaches have obtained an increasing amount of attention over the last few years, and certainly have obtained promising results on a large variety of problems [12],[13]. These approaches are also constructive and beneficial for multiobjective optimization as well [14]. By doing that, the exploration has been shown to be more effectively stimulated [15]. Inspired by this idea, DAES appropriately combines the following three operators: discrete recombination, Gaussian mutation and Cauchy mutation. Therein, Gaussian mutation mainly severs for the proximity while Cauchy mutation for the diversity according to status of the parents offered to them and the offspring generated by them.

2.3 Exploitation of proximity and diversity

It is also essential 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 only indicate how diverse we are but cannot indicate how close to 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. So a straightforward way to obtain strong pressure

toward the proximity is a *complete nondominance selection* in which all dominated individuals are neglected and only nondominated individuals have the opportunity to survive and reproduce in each generation [16]. This approach is considered cautious because it easily leads to premature convergence or local optimum unless other strong diversity preservation is accompanied with. Another class such as the concept of *domination count* [14] and *domination rank* [3] are introduced for proximity selection while *crowded comparison* [3] serves for diversity preservation. Although these approaches performed successfully in some situation, the problem caused by inappropriate exploitation scheme was still not completely resolved. Here in DAES, we proposed an exploitation scheme that is also separated into two sides respectively, but these incompatible sides are designed to work in mutual benefit. A complete nondominance selection derived from [16] ensures maximal pressure for the proximity selection while a fitness assignment determined by dominance and population information ensures maximal diversity preservation. Particulars will be discussed in the following Section.

3 Arithmetic of DAES

A few items that are of major importance to implement an adaptive balance between the proximity and diversity can be outlined when constructing our DAES.

- The external archive should be used to save elitist individuals as well as relatively better individuals, and a dynamic size of it is preferable.
- The update scheme of the archive should be devised to accompany dynamic size including the archive increase scheme and the archive decrease scheme.
- The exploration operator should produce better offspring than either of its parents.
- Survive of the better individuals should be automatically done based on both the proximity selection and the diversity preservation.

The global pseudocode of DAES is shown in Fig.1 based on these considerations.

```
function result = DAES ( μ, λ, age_min, size_min )
   Generate initial population P of μ random individuals and evaluate;
   Copy nondominated member of P in the external archive G;
   Initialize generation counter t zero;
   do  //exploration stage
      Reset the current intermediate population H empty;
      Set offspring counter r zero;
      do  //generate λ offspring
         Pick individuals a and b ( not dominated by a ) from P È G randomly;
         c = ProduceOffspring ( a, b, P, G );  //via our exploration operator
         IncreaseArchive ( c, a, b, G, size_min );  Add c into H;   r = r + 1;
      while ( r < λ )
      DecreaseArchive ( G, age_min, G, λ );   //below is exploitation stage
      AssignFitness ( H );  //via our fitness assignment scheme
      Sort H in descending order of fitness;
      Pick the first μ members into the next population P( t+1);  t = t + 1;
   while ( terminal criterion is not satisfied )
   Set result the unique nondominated subset of P( t ) È G;  return result;
end function.
```

Fig. 1. Global pseudocode of DAES

It should be noted that the heart of DAES involves two populations: the current population P and the external archive G. In detail, P works similar to the single objective ES; but G serves a binary purpose — storage of elitist individuals as well as relatively better individuals found during the run, and the cooperative role of one-side parent in producing offspring. In the beginning of each generation, parents are randomly selected from P and G (the later selected parent side should not be dominated by the former in order to accelerate prepotency), and λ offspring will be produced within a loop in the 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 the assigned fitness to select the first μ individuals with higher fitness into the next population. These sequences of instructions are repeated until terminal criterion is satisfied.

3.1 Code representation

In DAES, the real number representation is used instead of the binary string implementation. Since many problems in the real-world are expressed in real variables, faster computation can be obtain without conversions between different representations. It is characterized that DAES borrow a ternary group representation from [17], where an individual was denoted by $(\boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\theta})$ group. That is, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is the original decision vector, corresponding to a point in the decision space, and $\boldsymbol{\sigma} =(\sigma_1, \sigma_2,\ldots, \sigma_n)$ is the standard deviation used to instruct mutation, and $\boldsymbol{\theta} =(\theta_{1,2}, \theta_{1,3},\ldots,\theta_{1,n}, \theta_{2,3},\ldots, \theta_{(n-1),\, n})$ is the rotation angle used to change orientation of the mutation associated with all possible pairs of the decision vectors. Both $\boldsymbol{\sigma}$ and $\boldsymbol{\theta}$ are called strategy parameters. It is very similar to a hillclimber algorithm with a self-adaptive step $\boldsymbol{\sigma}$ and angle $\boldsymbol{\theta}$. In other words, all components are submitted to the evolutionary process by applying the exploration operator and the 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 Archive increase scheme

Generally, if an MOEA has an external archive with fixed size, a replacement scheme is always applied. In this scheme, in order to keep 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 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 [4] and preselection scheme in [9] have been introduced, the problem caused by the replacement scheme is still not completely resolved. Therefore, DAES adopts two independent schemes—the increase scheme and the 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 we used the archive to store elitist individuals as well as relatively better individuals that obtain progress responding to their parents, the dominated individuals also have the opportunity to survive in the archive. This eliminates the disadvantage in [16] 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) [18] 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. In DAES, if a newborn offspring is better than either of its parents in the proximity or the diversity (indicated by dominance or location of less crowded region [6]), it has the priority to be added into the 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 to be one when it is added for the first time. As a result, this scheme will guarantee that a new-added individual in the archive will have higher proximity or diversity than at least one of its parents, which helps DAES provide both elitist individuals and relatively better individuals participating in the exploration and exploitation stages so as to cover all the unexplored regions in the objective space. Fig. 2 describes the pseudocode of the archive increase scheme.

```
subprocedure IncreaseArchive ( offspring, par1, par2, ex_ar, size_min )
  if ( size of ex_ar is less than size_min ) {
    Add offspring into ex_ar;   Set age of offspring one; //initialize age }
  else
    if ( offspring dominates par1 or par2 ); {  //apply dominance
      Add offspring into ex_ar;  Set age of offspring one; }
    else if ( offspring in a less crowded region of ex_ar than par1 or par2 )  {
      Add offspring into ex_ar;  Set age of offspring one; }
end subprocedure.
```

Fig. 2. Pseudocode of the archive increase scheme

3.3 Archive decrease scheme

An archive decrease scheme is necessary to prevent the archive from growing without bound. In DAES, whether an individual will be removed from the archive or not depends on its *age* and *fitness*. The initial age is one, and it will grow generation by generation as long as it survives in the archive. To ensure that an removed individual has a lowest fitness value and has been adequately explored and exploited, the scheme removes individuals in each generation according to the following principles:

- In order to keep balance of the archive in the probability sense, there are λ trials to remove relatively worse individuals in each decrease scheme because λ offspring have been produced and correspondingly there are λ trials to add relatively better individuals in each increase scheme. This means that there are at most λ individuals can be removed if we perform the decrease scheme once. Fig.3 shows an example.
- We only remove the individuals with lowest fitness (either positive value or zero value) and whose ages are larger than the prespecified age threshold *age_min*. From the example in Fig.3, the archive is different after decrease due to their different age, fitness and age threshold when λ is 4. Fig. 4 describes this pseudocode.

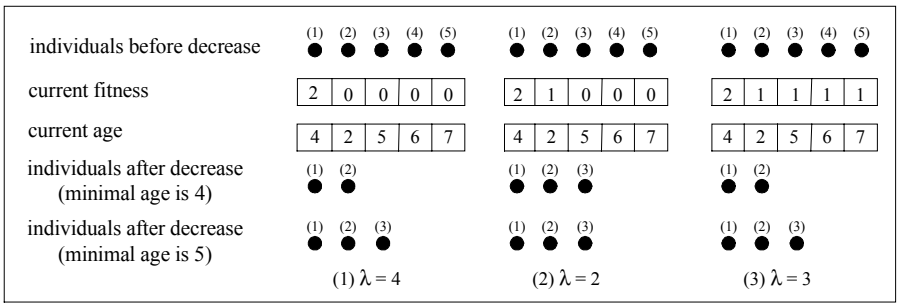

Fig. 3. Illustration of the decrease scheme: (1) There are 4 trials to remove the worse individuals, so the last three individuals (all zero fitness) are removed where minimal age is 4; (2) Only two worst individuals (all zero fitness) are removed and an individual whose age is larger than age threshold exist. (3) The worst individuals (positive fitness) are removed.

```
subprocedure DecreaseArchive ( ex_ar, size_min, age_min, offspring_num )
  if ( size of ex_ar is larger than size_min ) {
    for ( each member m of ex_ar )  Increase age of m by one;
    Initialize remover counter r zero;
    AssignFitness ( ex_ar );  //assign fitness for direct remove
    do  //remove dominated members from ex_ar within offspring_num trials
      if ( find a member m1 with fitness 0 and age > age_min ) {
        Remove m1 from ex_ar;  Decrease size of ex_ar by one; }
      else { if ( find a member m2 whose age > age_min ) {
          Remove m2 from ex_ar;  Decrease size of ex_ar by one; }} r = r + 1;
    while ( r < offspring_num Ã size of ex_ar is larger than size_min ); }
end subprocedure.
```

Fig. 4. Pseudocode of the archive decrease scheme

3.4 Exploration operator

In DAES, exploration includes recombination and mutation. Basically, recombination works choosing parents with uniform probability, and characteristics of parents are mixed to create one offspring. DAES employs the discrete recombination [19] that is commonly used in ES, and has also produced good results with real-coded MOEAs. Each component of offspring inherits from one of the parents randomly. If parent-1 is ($\boldsymbol{x}^{(1)}$, $\boldsymbol{\sigma}^{(1)}$, $\boldsymbol{\theta}^{(1)}$) and parent-2 ($\boldsymbol{x}^{(2)}$, $\boldsymbol{\sigma}^{(2)}$, $\boldsymbol{\theta}^{(2)}$), then we produce one offspring as

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

where, q_i is chosen between 1 and 2 with probability 0.5. Notice that recombination is performed independently on the decision vectors as well as on the 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

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

where, τ', τ and γ are parameters. $N(0, 1)$ denotes to generate a random scalar characterized by Gaussian distribution with zero mean and deviation one, and additional subscripts denote independent regeneration for each element respectively. $\boldsymbol{N}(0, cov)$ 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 diagonal of the covariance matrix as

$$c_{i,i} = \sigma_i'; \quad c_{i,j} = \theta_{i,j}', \forall i, j \in \{1, \cdots, n\}, j > i \tag{4}$$

Recently, a new type of mutation using Cauchy distribution was inspired in [11] and [12], which all claimed that Cauchy mutation outperformed Gaussian one on diversity preservation and search efficiency. Cauchy probability density function is

$$f(x) = \beta / \pi(\beta^2 + (x - \alpha)^2) \quad \alpha > 0, \beta > 0, -\infty < x < \infty \tag{5}$$

represented as $C(\alpha, \beta)$, where α and β are two parameters. We may produce a new individual via Cauchy mutation as

$$\begin{aligned} &\sigma_i'' = \sigma_i' \exp(\tau' C(0,1) + \tau C_i(0,1)); \\ &\theta_{i,j}'' = \theta_{i,j}' + \gamma C_{i,j}(0,1); \quad \forall i, j \in \{1, \cdots, n\}, j > i; \\ &\boldsymbol{x}'' = \boldsymbol{x}' + \boldsymbol{C}(0, cov(\boldsymbol{\sigma}'', \boldsymbol{\theta}'')) \end{aligned} \tag{6}$$

where $\boldsymbol{C}(0,\cdot)$ denotes to yield random scalar or vector submitted to Cauchy distribution with respective parameters. The similar denotations are omitted here. We plot two distributions in Fig.5 by applying the same parameters. We can investigate that Cauchy distribution is symmetrical and long-tailed, and 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.

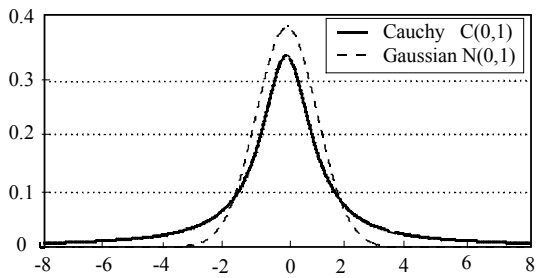

Fig. 5. Cauchy and Gaussian Distributions (with the same parameters)

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. The opposite, Gaussian mutation is accomplished in accurate search of its nearest space for the proximity exploration. If we take measures to develop the respective merits from two mutations, it sounds possible to provide higher performance on both proximity and diversity, Fig.6 shows the measures that will be taken to realize our idea. Coupling the recombination and mutation, DAES produces an offspring completely as pseudocode in Fig.7 describes.

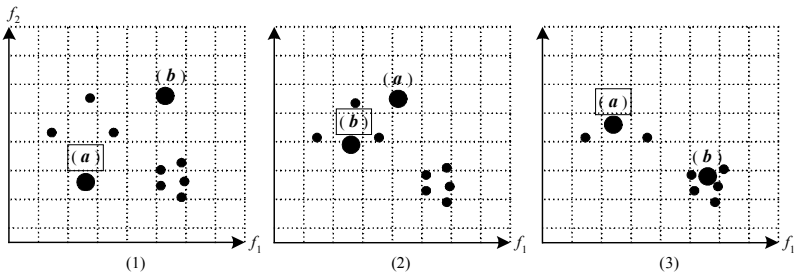

Fig. 6. Illustration of measures for exploration operator: (1) ***a*** dominates ***b***, so ***a*** is chosen; (2) ***b*** dominates ***a***, so ***b*** is chosen; (3) incomparable, ***b*** is located in a more crowded region, ***a*** is chosen

```
function offspring = ProduceOffspring ( par1, par2, cur_pop, ex_ar )
   Produce G_offspring with Gaussian mutation; // via Eq.(2) & Eq.(3)
   Produce C_offspring with Cauchy mutation;  //via Eq.(2) & Eq.(6)
   // compare two offspring and decide which one is accepted
   if ( G_offspring dominates C_offspring )  offspring = G_offspring;
   else if ( C_offspring dominates G_offspring ) offspring = C_offspring;
   else  //incomparable, so the crowded location is applied
      if ( G_offspring in a less crowded region of cur_pop È ex_ar than C_offspring )
         offspring = G_offspring;
      else  offspring = C_offspring;
   return offspring;
end function.
```

Fig. 7. Pseudocode of exploration operator

3.5 Fitness assignment scheme

Since fitness assignment scheme is crucially accompanied with the complete nondominance selection, we use fitness value as an indicator to distinguish dominated and nondominated individuals. It should be devised to meet the following requirements.

- The dominated individuals must share the lowest fitness than nondominated ones due to their absence of qualification for survival into 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.
- The individuals far away from the center of the current nondominated set must be assigned a higher fitness so as to share a higher probability of survival, which contributes to the diversity preservation and reducing the risk of premature convergence.

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

$$F(j) = \min \sum_{l=1}^{m} \{\|\boldsymbol{x} - \boldsymbol{x}_l\|_{obj}\} \Big/ m, \boldsymbol{x}, \boldsymbol{x}_l \in H \tag{7}$$

where, j denotes the nondominated individual, $\boldsymbol{x}$ is the decision vector of j and function $\|\cdot\|_{obj}$ calculates the Euclid distance in the objective space between two individuals.

4 Comparison study

In order to validate the proposed DAES and quantitatively compare its performance with other advanced MOEAs, three recently designed benchmark functions [20] described in Table.1 are tested by three existed algorithms — SPEA2 [5], NSGA2 [3], M-PAES [7] and the proposed DAES in this comparison study, and each algorithm runs 50 times independently for each function to obtain statistical results. Since the comparison focuses 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 each algorithm, and the nondominated set in the final population and the archive was taken as the output of an optimization run. Table.2 lists the general parameters setting of four algorithms for all runs, referred to literatures, e.g., [4], [5]. Then we use two methods: (1) graphical presentation for visual inspection and (2) the performance metric to show quantitative inspection.

Table 1. Three benchmark functions. All P_F is formed with $g = 1$. n is the number of decision vector

No	Function	n	Boundary	Characters
T_1	$f_1(\boldsymbol{x}) = x_1; f_2(\boldsymbol{x}) = g\cdot(1-\sqrt{f_1/g})$ $g = 1+9\cdot(\sum_{i=2}^{n} x_i)/(n-1)$	30	$[0,1]^n$	continuous convex
T_2	$f_1(\boldsymbol{x}) = x_1; f_2(\boldsymbol{x}) = g\cdot(1-f_1/g)^2$ $g = 1+9\cdot(\sum_{i=2}^{n} x_i)/(n-1)$	30	$[0,1]^n$	continuous concave
T_3	$f_1(\boldsymbol{x}) = x_1; f_2(\boldsymbol{x}) = g\cdot(1-\sqrt{f_1/g}-(f_1/g)\sin(10\pi f_1))$ $g = 1+9\cdot(\sum_{i=2}^{n} x_i)/(n-1)$	30	$[0,1]^n$	discrete

Table 2. General parameters setting of four algorithms

Common Parameters	DAES	SPEA2	NSGA2	M-PAES
Chromosome length	-	15×*dec_num*	15×*dec_num*(3)	15×*dec_num*
Population size	20	80	100	1
Archive size	60(1)	20	0(2)	100
Offspring per generation	60	10	10	10
Crossover rate	-	0.7	0.7	-
Individuals perform mutation	1	1	1	10
Number of binary bit flipped	-	1	1	1
Maximum generation	10,000	10,000	10,000	10,000

(1) It is the minimal size of the archive.
(2) Instead of using archive, NSGA2 combines the populations from two consecutive generations.
(3) *dec_num* represents the number of decision variables.

4.1 Graphical presentation

The first method is the graphical presentation. We unify the outcomes of each benchmark function from each algorithm, and all the nondominated individuals are plotted as shown in Fig.8 and Fig.9 in company with its *true* P_F (called PF_{true} in this section).

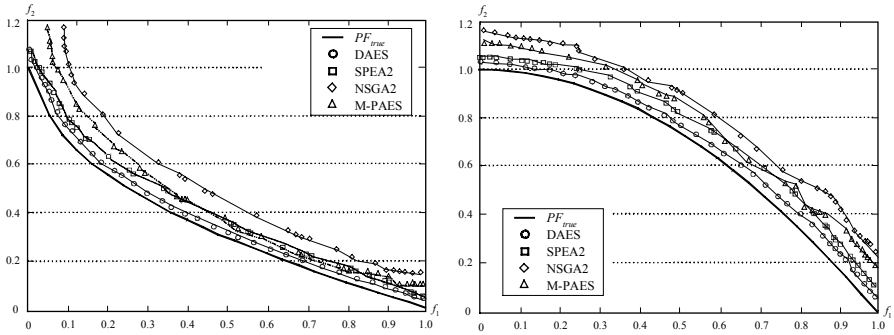

Fig. 8. Graphical presentation based on T_1 (left) and T_2 (right)

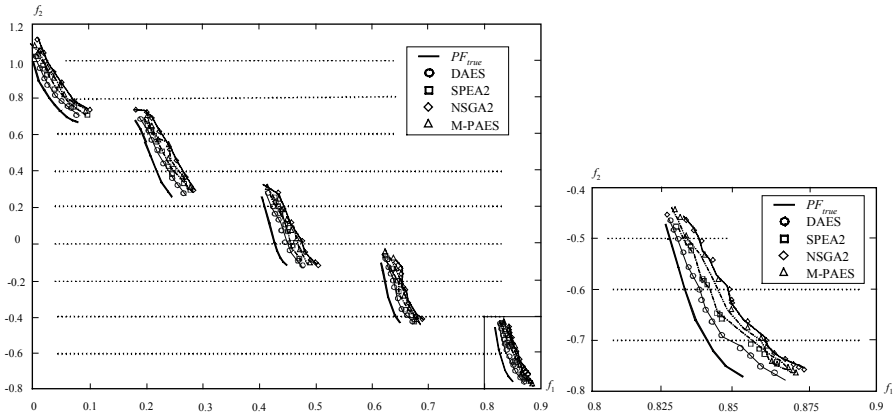

Fig. 9. Graphical presentation based on T_3 (left) and enlarged rectangle of T_3 from left (right)

For continuous function T_1 and T_2, the lines plotted by the solutions of DAES are much closer to the front than all its competitors. Besides, the solution distribution also varied. DAES distributes its solutions symmetrically in both middle and tail without obvious difference, which more than seventy percent of solutions of the latter three algorithms are crowded within thirty percent of region in the tail. This asymmetric distribution makes it difficult for decision making on middle compromise solutions. The results of function T_3 from either of the algorithms are a bit close. Both show less perfect than what has achieved in continuous functions. To show the discrepancy, we enlarged the region of the closest proximity between 0.8 and 0.9 of f_1 and displayed it on the right. It also demonstrates that DAES distributes their solutions closer to PF_{true} and more diversely than the others do. As a result, DAES is superior and the pre-dominance decreases from continuous functions to discrete function to some extent.

4.2 Performance metrics

We use three indicators to benchmark the comparison. The first indicator is the general distance (GD) [21] that can show how far PF_{final} are away from PF_{true}, calculated by

$$GD = (\sum_{i=1}^{N} d_i^p)^{1/p} \Big/ N, d_i = \min_j(\left|f_1^i(\boldsymbol{x}) - f_1^j(\boldsymbol{x})\right| + \left|f_2^i(\boldsymbol{x}) - f_2^j(\boldsymbol{x})\right|) \tag{8}$$

where N is the number of nondominated individuals in PF_{final} , p = 2, and d_i is the Euclidean distance in the objective space between each of these individuals and a closest point on PF_{true}. A smaller value of GD is preferable.

Because if another desire to measure diversity of solutions. We adopt the spacing (SP) to measure the range variance of neighboring individuals, defined as

$$SP = \sqrt{\sum_{i=1}^{N} (\overline{d} - d_i)^2 \Big/ (N-1)}, \overline{d} = (\sum_{i=1}^{N} d_i) / N \tag{9}$$

A zero value of SP indicates the ideal diversity that all nondominated individuals are equidistantly and uniformly spaced, and a smaller value of SP is preferable.

Moreover, the C value [20] is also included to compare the dominance relationship between two algorithms. It maps the ordered pair (X_i, X_j) to interval [0, 1], defined as

$$C(X_i, X_j) = \left|\{\boldsymbol{y} \in X_j \wedge (\exists \boldsymbol{x} \in X_i : \boldsymbol{x} \succ \boldsymbol{y} \vee \boldsymbol{x} = \boldsymbol{y})\}\right| \Big/ \left|X_j\right| \tag{10}$$

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. Both should be considered independently.

Here, *box plots* [22] are used to visualize the distribution of these indicator samples. The box plots concerning the three indicators are shown in Fig.10-12 respectively.

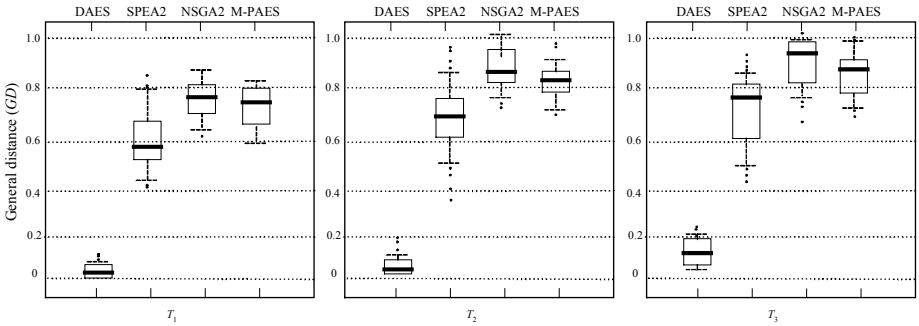

Fig. 10. Box plots based on metrics of the general distance (GD) of three functions

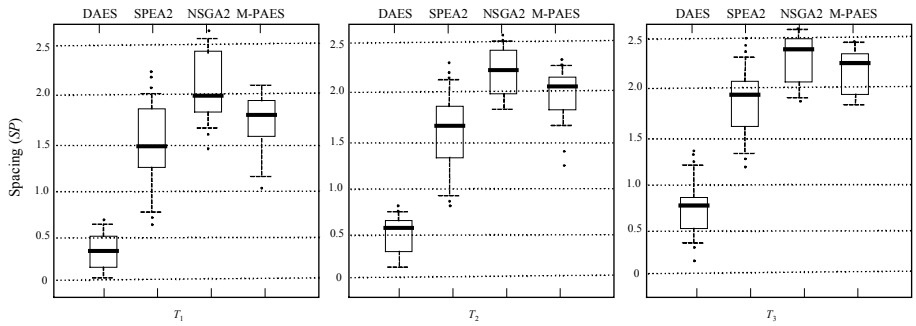

Fig.11. Box plots based on metrics of the spacing (SP) of three functions

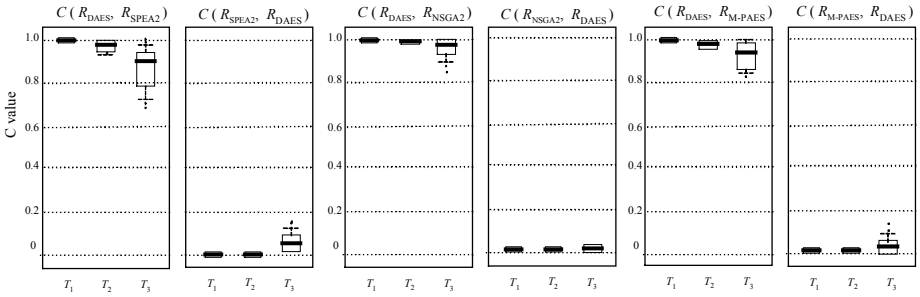

Fig. 12. Box plots based on the metric of *C* value of three functions

It is revealed that DAES is completely superior to all other algorithms on metrics of GD and SP. As far as *C* values is concerned, DAES covers all its competitors except for a few occasions, however, none of the other algorithms covers any of DAES' solutions. Consequently, the inspection of performance metrics shows the similar trend as the graphical presentation has discovered.

5 Conclusions

In this paper, we have proposed a novel DAES. It can be characterized as: (1) 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; (2) adaptively increasing or decreasing the external archive; (3) effectively improving the competence of exploration operator to provide new nondominated individuals and diversify them well by means of a combinatorial operator; (4) powerfully ensuring maximal pressure on proximity selection via a complete nondominance selection and ensuring maximal diversity preservation via a fitness assignment determined by dominance and population diversity information; and (5) converging to a near-optimal, well-extended and uniformly diversified Pareto optimal front. From comparison study, DAES has shown its potential in producing statistically superior results to SPEA2, NSGA2 and M-PAES on three prominent benchmark functions. So we suggest that DAES be a potential candidate in solving complicated problems.

However, as the benchmark functions used in this paper are still far from covering all the challenging characteristics of multiobjective optimization, a more profound study by applying DAES in dealing with other real-world problems is absolutely necessary in the future work. Additionally, there is no special design built in DAES to handle strong constraints. In near future, DAES will be revised to deal with those multiobjective optimization with strong and complicated constraints.

Acknowledgements

The authors gratefully acknowledge the support of the National Grand Fundamental Research 973 Program of China under Grant No.2003CB415206.

References

1. Fonseca, C.M., Fleming, P.J.: An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation. 1 (1995) 1–16
2. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation. 2 (1994) 221–248
3. Deb, K., Agrawal, S., Pratap, A., et al.: A Fast Elitist Nondominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II. Evolutionary Computation. 2 (2002) 182–197
4. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. Evolutionary Computation. 1 (1999) 257–271
5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: EUROGEN 2001, Athens, Greece (2001)
6. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation. 2 (2000) 149-172
7. Knowles, J.D., Corne, D.W.: M-PAES: A Memetic Algorithm for Multiobjective Optimization. In: Proceedings of the 2000 congress on evolutionary computation. Piscataway, NJ: IEEE Press (2000) 325–332
8. Mahfoud, S.W.: Genetic Drift in Sharing Methods. In: Grefenstette, J.J. (eds.): Proceedings of the 1st IEEE conference on evolutionary computation. Piscataway, NJ: IEEE Press (1994) 67–72
9. Schwefel, H.P, Back, T.: A Survey of Evolutionary Strategies, In: Belew, R. (eds.): Proceedings of the 4th international conference on genetic algorithms. Morgan Kaufmann publishers, San Mateo, CA (1991) 92-99
10. Bosman, P.A.N., Thierens, D.: The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms. Evolutionary Computation. 7 (2003) 174-188
11. Yao, X., Liu, Y.: Fast Evolutionary Programming. In: Proceedings of the 5th annual conference on evolutionary programming, The MIT Press, Cambridge, MA (1996) 451-460
12. Gomes, J.R., Saavedra, O.R.: A Cauchy-based Evolution Strategy for Solving the Reactive Power Dispatch Problem. Electrical Power and Energy Systems. 24 (2002) 277-283
13. Shi, L.B., Xu, G.Y.: Self-adaptive Evolutionary Programming and its Application to Multiobjective Optimal Operation of Power Systems. Electric Power Systems Research. 57 (2001) 181-187
14. Bosman, P.A.N., Thierens, D.: Multiobjective Optimization with Diversity Preserving Mixture-based Iterated Density Estimation Evolutionary Algorithms. Int. J. Approx. Reasoning. 31 (2002) 259–289
15. Tan, K.C., Lee, T., Khor, E.: Evolutionary Algorithms with Dynamic Population Size and Local Exploration for Multiobjective Optimization. Evolutionary Computation. 12 (2001) 565–588
16. Sarker, R., Liang, K.H., Newtom, C.: A New Multiobjective Evolutionary Algorithm. European Journal of Operational Research. 140 (2002) 12-23
17. Schwefel, H.P.: Numerical Optimization for Computer Models. John Wiley. Chichester, UK (1981) 129-132
18. Hu, X., Eberhart, R.C.: Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In: Proceedings of the 2002 congress on evolutionary computation. Piscataway, NJ: IEEE Press (2002) 1677–1681
19. Back, T., Schwefel, H.P.: An Overview of Evolutionary Algorithm for Parameter Optimization. Evolutionary Computation. 2 (1993) 1-23
20. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms : Empirical Results. Evolutionary Computation. 2 (2000) 173-195
21. van Veldhuizen, D.A., Lamont, G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance. In: Proceedings of the 2000 congress on evolutionary computation. Piscataway, NJ: IEEE Press (2000) 204-211
22. Chambers, J.M., Cleveland, W.S., Kleiner, B., et al.: Graphical Methods for Data Analysis. Pacific Grove, CA: Wadsworth & Brooks/Cole (1983)