一种基于偏好的多目标调和遗传算法*

崔逊学[1,2+], 林 闯[2]

[1](南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

[2](清华大学 计算机科学与技术系,北京 100084)

A Preference-Based Multi-Objective Concordance Genetic Algorithm

CUI Xun-Xue[1,2+], LIN Chuang[2]

[1](State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

[2](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-551-5769700, E-mail: xxcui@tsinghua.org.cn

Received 2003-11-30; Accepted 2004-05-11

Cui XX , Lin C. A preference-based multi-objective concordance genetic algorithm. *Journal of Software*, 2005,16(5):761−770. DOI: 10.1360/jos0761

Abstract: Recently various evolutionary approaches have been developed for multi-objective optimization. Most of them take Pareto dominance as their selection strategy and do not require any preference information. However these algorithms cannot perform well on problems involving many objectives. By introducing preferences among different criteria, a multi-objective concordance genetic algorithm (MOCGA) is proposed to deal with the problems in the paper. As the number of objectives to be simultaneously optimized increases, the weak dominance is used to compare among the individuals with decision-maker's information. It is proven that the algorithm can guarantee the convergence towards the global optimum. Experimental results of the multi-objective optimization benchmark problems demonstrate the validity of the new algorithm.

Key words: genetic algorithm; multi-objective optimization; preferences information; multi-criterion decision-making

摘 要: 最近涌现了各种进化方法来解决多目标优化问题,多数方法使用 Pareto 优胜关系作为选择策略而没有采用偏好信息.这些算法不能有效处理目标数目许多时的优化问题.通过在不同准则之间引入偏好来解决该问题,提出一种多目标调和遗传算法 MOCGA(multi-objective concordance genetic algorithm).当同时待优化的目标数目增加时,根据决策者提供的信息使用弱优胜关系进行个体优劣的比较.这种算法被证明为能收敛至全局最优.对于目标数目

---

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314804 (国家重点基础研究发展规划(973)); the National Natural Science Foundation of China under Grant Nos.90104002, 60303027 (国家自然科学基金); Opening Foundation of State Key Laboratory for Novel Software Technology in Nanjing University (南京大学计算机软件新技术国家重点实验室开放基金); the Natural Science Foundation of Anhui Province of China (安徽省自然科学基金)

CUI Xun-Xue was born in 1969. He is a post-doctor at the Department of Computer Science and Technology, Tsinghua University. His research areas are evolutionary algorithms, routing algorithm and optimal theory. **LIN Chuang** was born in 1948. He is a professor and doctoral supervisor at the Department of Computer Science and Technology, Tsinghua University. His researches areas are computer networks, stochastic Petri nets and performance analysis of computer systems.

为很多的优化问题,测试实验结果表明了这种新算法的有效性.

关键词: 遗传算法;多目标优化;偏好信息;多准则决策

中图法分类号: TP18 **文献标识码**: A

1 Introduction

Many real-world problems are multi-objective in nature, because they consider several objectives that are to be optimized simultaneously. There is no a single optimal solution, but rather a set of alternative solutions for the kind of problems. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered[1]. Multi-objective optimization problems (MOPs) have received considerable attention in the field of Operation Research. Classical optimization methods such as multi-criterion decision-making methods suggest converting an MOP to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution. When such a method is used for finding multiple solutions, it has to be applied many times and results in obtaining a different and even incompatible solution at each run. Finally a decision-maker is difficult to select optimal alternatives from them using those methods.

Recently MOPs have become a popular area of research within evolutionary computation that is normally called Evolutionary Multi-objective Optimization. Over the past decade, a number of multi-objective evolutionary algorithms (MOEAs) have been proposed[2]. EAs are well suited for MOPs because they process a set of solutions in parallel. Some researchers suggest that MOPs seem to belong to an area where EAs do better than other blind search strategies[3]. Although this statement might be qualified with regard to the "No Free Lunch" (NFL) theorems[4], up to now there are few other alternatives solving MOPs good like EAs. In most cases these EAs are modifications of the genetic algorithms, so genetic algorithm is the most important evolutionary approach among MOEAs.

Selection is the key mechanism in evolutionary computation. In the case of multiple objectives, the selection operator steers the search in direction of the nondominated front, and is controlled by the individual's fitness that reflects its utility of Pareto-optimality. Therefore fitness assignment is the main issue in multi-objective optimization. Many MOEAs use Pareto-based fitness assignment, which directly bases on Pareto dominance and assigns all nondominated solutions equal reproduction probabilities. A scheme was introduced by Fonseca and Fleming, where an individual's rank corresponds to the number of solutions in the population by which it is dominated[5]. Other Pareto-based approaches include the Strength Pareto Evolutionary Algorithm[6] and the Nondominated Sorting Genetic Algorithm[7]. Although they do not require any preference information, the dimensionality of the search space influences their performance. Fonseca and Fleming pointed out that pure Pareto EAs cannot be expected to perform well on problems involving many competing objectives, and may simply fail to produce satisfactory solutions due to the large dimensionality and size of the trade-off surface[3].

Aiming at the problem, preferences have been integrated with a multi-objective environment. By introducing preference information among different objectives, evolutionary population is weakly ranked as the selection strategy according to the theory of multicriterion decision-making. The advantage of the algorithm lies on that it can solve optimization problems with many objectives. Other work in the literatures reported the use of preferences[8], and recently Dragan and Parmee proposed a fuzzy preference method where these preferences were developed with a goal to reduce the cognitive overload[9]. But they were used in the different circumstances and for the different purposes. A survey on the use of preference in a multi-objective context was provided by Coello[10].

The next section introduces some concepts used in the field of evolutionary multi-objective optimization and preference relationship. Section 3 presents MOCGA. Section 4 gives the performances for high dimensionality optimization problems. The convergence property is discussed in Section 5. The paper concludes with Section 6.

2 Main MOP Concepts

2.1 Multi-objective optimization

A general multi-objective problem can be described as a vector function f that maps a tuple of m decision variables to a tuple of n objectives (criteria). Formally:

$$\text{Min/Max } \mathbf{y}=f(\mathbf{x})=(f_1(\mathbf{x}),f_2(\mathbf{x}),\ldots,f_n(\mathbf{x})) \tag{1}$$

Subject to $\mathbf{x}=(x_1,x_2,\ldots,x_m)\in X$, $\mathbf{y}=(y_1,y_2,\ldots,y_m)\in Y$.

where **x** is called *decision vector*, X *parameter space*, **y** *objective vector*, and Y *objective space*.

If the set of solutions consists of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another, these vectors are known as *Pareto optimal*. Several concepts are mathematically defined as follows:

Definition 1. (Pareto Dominance): In a minimum problem, a vector $\mathbf{u}=(u_1,u_2,\ldots,u_n)$ is said to dominate $\mathbf{v}=(v_1, v_2,\ldots,v_n)$ iff **u** is partially less than **v**, i.e., $\forall i\in\{1,\ldots,n\}, u_i\le v_i \wedge \exists i\in\{1,\ldots,n\}: u_i<v_i$.

Definition 2. (Pareto Optimality): A solution $\mathbf{x}\in X$ is said to be Pareto optimal with respect to X iff there is no $\mathbf{x}'\in X$ for which $\mathbf{v}=f(\mathbf{x}')=(f_1(\mathbf{x}'),\ldots,f_n(\mathbf{x}'))$ dominates $\mathbf{u}=f(\mathbf{x})=(f_1(\mathbf{x}),\ldots,f_n(\mathbf{x}))$.

2.2 Semantics of preference

For a multi-objective preference system, the initial set of atomic propositions is given by a fundamental binary relation R, defined on a finite set A of alternatives with the following general semantics:

Definition 3. (Outranking Relation): $\forall a,b\in A:(a\,R\,b)\equiv$ "*a more or less outranks b*"

On the basis of the general outranking relation R, we can define adequate preference, indifference and incomparability relations, denoted respectively as ***P***, ***I*** and ***J***. The semantics of each relation is given as follows:

$\forall i,j\in A$: (i **P** j)≡"i *is more or less preferred to* j"

(i **I** j)≡"i *is more or less indifferent to* j"

(i **J** j)≡"i *is more or less incomparable to* j"

The three relations are different from the ordinary multi-objective optimization and provide a complicated comparison relation between any two alternatives. They elicit two concepts: concordance index and discordance index, which are used in MOCGA.

The ***concordance index*** between any two alternatives i and j is a weighted measure of a criterion, for which alternative i is preferred to alternative j (denoted $i\succ j$) or for which i is equal to j (denoted $i\sim j$). It is given as:

$$C(i,j)=\frac{\sum_{k\in A(i,j)}\omega(k)}{\sum_{k=1}^{n}\omega(k)} \tag{2}$$

where $\omega(k)$ is the weight of criterion k, $k=1,\ldots,n$, and $A(i,j)=\{k|i\succeq j\}$, $0\le C(i,j)\le 1$. Concordance is considered as the weighted percentage of criteria for which one alternative is preferred to another.

Determination of the discordance between i and j requires that an interval scale common to each criterion be defined. The scale is used to compare the discomfort caused between the worst and the best criterion values for each pair of alternatives. A range may be chosen where the best rating would be assigned the highest value of the range, and the worst rating would receive the lowest value of the one. Each criterion can have a different range to reflect the leeway available for that criterion. The ***discordance index*** is defined as:

$$D(i,j)=\frac{\max_{k=1,n}(r(j,k)-r(i,k))}{R^*} \tag{3}$$

where $r(j,k)$ is the evaluation of alternative j with respect to criterion k, and R^* is the largest of the n criterion scales by construction existing $0 \le D(i,j) \le 1$.

3 MOCGA

3.1 Multi-objective model

The advantage of outranking relation is that it can handle intransitivity and contradictions in a local manner and allow incomparable relation ***J***. Typical technique is the ELECTRE method proposed by Roy[11], which includes versions I and II. The abbreviation "ELECTRE" is the abbreviated form of *Elimination et Choice Translating Reality* which is derived from French. The method provides a partial and weak ordering of the nondominated alternatives. The ordering is accomplished by the construction of outranking relationships with the preferences. Version I of ELECTRE will be embedded in the selection process of MOCGA. Its character is to choose those alternatives, which are preferred for at least a plurality of the criteria and yet do not cause an unacceptable level of discontent for any one criterion. ELECTRE I operates as follows:

i) A set of m alternatives (R) and a set of n objectives (criteria) are first given. From the decision matrix, a normalized one is constructed to take each criterion value as the same unit vector:

$$r_{ij} := \frac{r_{ij}}{\sqrt{\sum_{i=1}^{n} r_{ij}^2}} \tag{4}$$

If ω_j denotes the weight of criterion j, a weighted normalized decision matrix is then formed whose elements are given by:

ii) Weak outranking relation is constructed with threshold values from the concordance and discordance matrices. Then an ELECTRE's relation graph is designed where each node corresponds to a nondominated alternative and the arrows indicate preference dominances between any two nodes.

$$v_{ij} = \omega_j r_{ij} \tag{5}$$

① Concordance index

If a criterion in an alternative x_k is better than the same criterion in another alternative x_l, it is said to be preferred in terms of that criterion. If x_k is preferred to x_1 ($x_k \boldsymbol{P} x_1$), any normalized criterion value in support of the assumption is said to be concordant with $x_k \boldsymbol{P} x_1$, where $\boldsymbol{P}$ denotes an adequate preference relation defined before. A concordance set and a discordance set are built with the weighted normalized decision matrix:

$$\text{Concordance set: } C_{kl} = \{j \mid r_{kj} \ge r_{1j}\} \tag{6}$$

$$\text{Discordance set: } D_{kl} = \{j \mid r_{kj} < r_{1j}\} = M - C_{kl} \tag{7}$$

where M is the set of $\{1,2,...,n\}$, and the expression significations of C_{kl} and D_{kl} are different from the above $C(i,j)$ and $D(i,j)$. If $C'_{kl} = \{j \mid r_{kj} = r_{1j}\}$, two concordance indexes α_{k1} and $\hat{\alpha}_{kl}$ are defined respectively,

$$\alpha_{kl} = \sum_{j \in C_{kl}} \omega_j \tag{8}$$

$$\hat{\alpha}_{kl} = \left(\sum_{j \in C_{kl}} \omega_j - \sum_{j \in C'_{kl}} \omega_j\right) / \sum_{j \in D_{kl}} \omega_j \tag{9}$$

If $\alpha_{k1} \ge \alpha_0$ and $\hat{\alpha}_{kl} \ge 1$, the concordance check is passed. α_0 denotes the lowest concordance threshold value determined by a decision maker. For example, α_0 can be taken as the average of concordance index α_{k1}:

$$\alpha_0 = \sum_{\substack{k=1 \\ k \ne l}}^{m} \sum_{\substack{l=1 \\ l \ne k}}^{m} \alpha_{kl} / m(m-1) \tag{10}$$

② Discordance index

The discordance index is defined as follows:

$$\beta_{kl} = \frac{\max\limits_{j\in D_{kl}} |v_{kj} - v_{lj}|}{\max\limits_{j\in M} |v_{kj} - v_{lj}|} \tag{11}$$

If $\beta_{kl} \le \beta_0$, the discordance check is passed. β_0 denotes the highest discordance threshold value determined by a decision maker. The weights, α_0 and β_0 reflect the preference of the decision maker. For example, β_0 can be taken as the average of discordance index β_{kl}:

$$\beta_0 = \sum_{\substack{k=1\\k\ne l}}^{m} \sum_{\substack{l=1\\l\ne k}}^{m} \beta_{kl} / m(m-1) \tag{12}$$

iii) A minimal dominance subset R_D is obtained from R.

iv) If the number of designs in R_D is small enough to be chosen as preference alternatives from them, the iteration process ends; Otherwise, repeat (i) to (iv) after the concordance level is adjusted.

3.2 Detailed description of MOCGA

The subsection describes the details of MOCGA, whose emphasis is laid on integrating preference model with population evolution. In a finite population, each individual is weakly sorted (ranked) by the ELECTRE method, and avoids being difficult to be sorted for several objectives with usually strict Pareto-based comparison. The main procedure of MOCGA is described as follows.

(1) Initialize control parameters: population size N, the size of external nondominated set N', number of evolutionary iteration *gen*, crossover probability P_c, mutation probability P_m, the weight of every objective ω_i, the lowest concordance threshold value α_0, and the highest discordance threshold value β_0;

(2) Generate an initial population P and an empty external nondominated set P' when $gen=0$;

(3) Sort individuals in P with ELECTRE and copy the produced members in the minimal dominance set to P';

(4) Remove solutions in P' which are covered by any other member of P' (the step is ignored for the first copy, as now there is no dominated solutions in P');

(5) If the number of externally stored nondominated solutions exceeds a given maximum value, prune P' by means of averagely correlative clustering;

(6) Calculate the fitness of each individual in P and P';

(7) Select individuals from $P+P'$ with binary tournament rule whose constraints satisfy constraint values, until the mating pool is filled, then $gen:=gen+1$;

(8) Apply crossover and mutation operators related with a solving problem;

(9) If the maximum number of generations is reached, stop algorithm, else go to (3);

(10) Users adjust α_0 and β_0 values according to their needs, then go to step (1).

The fitness assignment in (6) includes two stages, which uses strength concept proposed by Zitzler and Thiele[6]:

Step 1: Each solution $i\in P'$ is assigned a real value $s_i\in[0,1]$ as its strength; s_i is proportional to the number of population members $j\in P$ for which $i\succeq j$. Let k denote the number of individuals in P that are covered by i, and assume N is the size of P. Then s_i is defined as $s_i=k/(N+1)$. The fitness f_i of i is equal to its strength: $f_i=s_i$.

Step 2: The fitness of an individual $j\in P$ is calculated by summing the s_i value of all external nondominated solution $i\in P'$ that cover j. Fitness is to be minimized (where $f_j\in[1,N]$):

$$f_j = 1 + \sum_{i, i \succeq j} s_i \tag{13}$$

The method of fitness assignment and clustering in step (5) avoids the phenomenon of genetic drift, which would influence the population diversity. Note that as the fitness is calculated during the algorithm execution, the dominated or nondominated relation between any two individuals means an absolutely Pareto dominated or nondominated one. Step (7) practically eliminates the unqualified individuals whose constraints exceed the bounds.

4 Performance Evaluation and Analysis

4.1 Multi-Objective optimization problem

4.1.1 4-D objective problem

In order to verify the effectiveness of MOCGA, a 4-D problem is tested in which one dimension is more than the usual benchmark problems. The paper designs a four-objective functional optimization problem of a single variable without constraints:

$$\text{Min } F(x)=(f_1(x),f_2(x),f_3(x),f_4(x)) \tag{14}$$
$$f_1(x)=x^2, f_2(x)=(x-2)^2, f_3(x)=(x-1)^2, f_4(x)=0.5*(x-1)^2$$

The decision variable x is taken as a real value between [−10, 10]. When the weight of every sub-objective is equal, i.e. 0.25, the optimal solution is obtained as x=1.0, which is solved by linear weighted method of multi-objective programming in advance. In the paper the famous Strength Pareto Evolutionary Algorithm (SPEA)[6] is compared with MOCGA for the problem. The main difference between the two algorithms lies on that MOCGA uses a weakly outranking to sort an individual, while SPEA uses Pareto-based purely superior relation to do it.

Here N=100, N'=30, P_c=0.75, P_m=0.01. The values of α_0 and β_0 in MOCGA are assigned by the related averages, i.e. α_0 taken as the average of concordance index, β_0 taken as the average of discordance index. Comparative results cannot be visually shown in graphical form for four-objective problem, indicating who performs better. The evaluation for comparing their performances is adopted by the statistical number of optimal solutions produced. The maximal generation is taken as 50, 100, 200, 500, respectively. Each time ten experiments were done while the two algorithms use the same initial population. The produced results are provided in Table 1. In Table the numbers before and after commas represent the number of optimal solutions in an external set created by MOCGA and SPGA, respectively.

Table 1 Results of a 4-D objective problem by MOCGA and SPEA

Gen. \ No.	1	2	3	4	5	6	7	8	9	10
50	11, 3	6, 2	5, 1	5, 2	6, 3	12, 5	4, 2	8, 2	12, 3	9, 4
100	13, 4	11, 5	11, 6	6, 3	14, 2	15, 4	7, 2	19, 4	11, 2	15, 3
200	5, 2	8, 4	5, 2	7, 3	17, 4	10, 2	14, 3	6, 2	9, 3	5, 2
500	8, 3	7, 2	8, 2	14, 3	11, 4	20, 2	12, 3	16, 4	12, 2	7, 4

In addition during each experiment the optimal solutions were mixed, which were done at the end of an evolution process of MOCGA and SPEA. The dominance relation among them was assessed to determinate whether there exist some dominated solutions. It was found out that most of these optimal solutions were nondominated by others with a few exceptions. The fact shows that MOCGA creates an evolutionary optimal set along the Pareto front as SPEA, although it adopts rather weak ranking than strictly Pareto-based dominance relation. Due to the reason, the two algorithms could be compared with the number of the last optimal solutions generated by them.

It is found from Table 1 that the number of optimal solutions from SPEA is much smaller than that from

MOGA. The trend behaves more and more evidently when the evolutionary generation is higher. As the dimensionality of optimization objectives is high, SPEA is difficult to sort individuals in a finite genetic population by Pareto-based rank, even existing a situation where all individuals are Pareto-optimal to each other in a population. Thus the normal iteration process might be interrupted using pure Pareto selection strategy; whereas MOCGA has not the disadvantage.

4.1.2 10-D objective problem

How does MOCGA deal with many competing objectives more efficiently? It would greatly convince people if it could be revealed that MOCGA can deal with more objectives. This paper gives a test with ten objectives of DTLZ2 problem proposed by Deb[12]. It can be scalable to more objectives. This is a really challenging problem and naturally enough to judge the performance of MOCGA on the high dimensional objective space. The optimization task considers a real-parameter function and investigates the EA′s ability to scale up its performance in a large number of objectives, which is defined as follows.

$$\text{Minimize } f_1(x)=(1+g(X_n))\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{n-2}\pi/2)\cos(x_{n-1}\pi/2)$$
$$\text{Minimize } f_2(x)=(1+g(X_n))\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{n-2}\pi/2)\sin(x_{n-1}\pi/2)$$
$$\cdots\cdots$$
$$\text{Minimize } f_n(x)=(1+g(X_n))\sin(x_1\pi/2) \tag{15}$$

where $g(X_n)=\sum_{x_i\in X_n}(x_i-0.5)^2$, with $X_n=[x_n,\ldots,x_m]$, and $0\le x_i\le 1$, for $i=1,2,\ldots,m$, with $m=n+k-1$.

As the above definitions, here n is the number of objectives, m is the number of variable, and k is a difficulty parameter. In this study n=10, m=11, and k=2. The function code design is followed in *C* language:

```
Pi=4.0*atan(1.0);
for(j=n, xkk=0.0; j<m; j++)    xkk+ =(x[j]–0.5) * (x[j]–0.5);
   for(j=n–1; j>=0; j--)
      {    temp=1.0;
         for(i=0; i<n–j–1; i++)    temp* = (i>=0)? cos(x[i] *Pi/2.0): 1.0;
        f[j]=(1.0+xkk)*temp* ((j>0) ?sin(x[n–j–1]*Pi/2.0): 1.0);
      }
```

The Pareto-optimal solutions correspond to $x_i^*=0.5$ ($x_i^*\in x_n$) and all objective function values must satisfy the $\sum_{i=1}^{n}(f_i^*)^2=1$. MOCGA was compared with the important NSGA-II proposed by Deb[7]. According to the viewpoint of Deb, the distribution of solutions (n=10) obtained with NSGA-II is poor in the problem[12]. His viewpoint was validated, whereas MOCGA got a good approximation set. If N_{ds} denotes the size of nondominated set (N') and the theoretical value of $\sum_{i=1}^{n}(f_i^*)^2$ is taken as 1.0, an evaluation measure is constructed based on the average deviation:

$$\Delta=\left(\sum_{j\in N'}\left|\sum_{i=1}^{n}f_i^2-1.0\right|\right)/N_{ds} \tag{16}$$

The benchmark problem was tested using the same genetic parameters as those in the 4-D experiment. Table 2 shows the 10 comparison results (Δ) after 100 generations. From the results of Table 2, the Pareto-optimal solutions of MOCGA have a little deviation. However they have a big deviation in NSGA-II which did not work well on the problem as Deb said[12]. It is well known that NSGA-II is a famous algorithm for solving search and optimization problems involving multiple conflicting objectives used in many studies. Its selection mechanism is not well fit for high dimensional objective optimization problems according to the paper. This result also proves the points of Fonseca and Fleming, i.e. pure Pareto EAs cannot perform well on problems involving many objectives[3]. MOCGA uses a weak dominance to sort individuals in which preference information is integrated. Some individuals are selected as local Pareto-optimal solutions in current population after all individuals are compared with weak dominance. Thus after a finite generations many Pareto-based solution points are successfully obtained to approximate the front of optimal set.

4.2 Analysis

It should be emphasized that the total performance of MOCGA is not very good for all MOPs, although MOCGA has its success in the numeric problems. Any algorithmic approach is bound to have some advantages and shortfalls when applied to certain problems, as proved by the NFL theorems[4]. Because many concordance and discordance checks need to be done in determining outranking relations, the computational complexity of MOCGA is increased. Thus MOCGA could be well suited for high dimensionality optimization problems and might not be good at MOPs of low dimensionality. The kind of multi-objective problems has an increased complexity introduced by large dimensionalities as more objectives are added. The excess computational capacity of MOCGA is counteracted by the difficulty of a multi-objective problem, and therefore the significance of MOCGA is obvious. Moreover, while the computation time is not important without considering real-time requirement, the computation complexity of MOCGA could be accepted by users.

5 Convergence Property

5.1 Regular MOEA theory

During a genetic process the local set of Pareto optimal solutions is determined at the tth generation and termed $P_{current}(t)$, which is with respect to the current population. The solutions are added to a secondary population termed $P_{known}(t)$, and the process continued until the process termination. $P_{known}(0)$ is always an empty set. Some MOEAs including MOCGA use an extern set to store known optimal solutions. All corresponding vectors of $P_{known}(t)$ are tested at each generation, and the dominated solutions are removed. The mixture of $P_{current}(t)\cup P_{known}(t)$ creates optimal solutions $P_{known}(t+1)$ of the next generation. The result is a final set of Pareto optimal solution found by an algorithm that is denoted as P_{known} without t symbol. The actual Pareto optimal solution set (termed P_{true}) is not usually known for problems. P_{true} is fixed and does not change with generation change.

$P_{current}(t)$ and $P_{known}(t)$ are sets of EA genotypes. The fitness is judged via phenotypes that is a Pareto front. The associated Pareto front for each of the above sets are denoted as $PF_{current}(t)$, $PF_{known}(t)$, and PF_{true}, respectively. The global optimum for an MOP is a set of vectors. Thus the Pareto front PF_{true} determined by evaluating the Pareto optimal set P_{true} is the global optimum of an MOP.

5.2 MOCGA convergence

Assumption 1: When using an EA to solve MOPs, the implicit qualification is that one of the following holds: $P_{known}=P_{true}$, $P_{known}\subset P_{true}$, or $PF_{known}\in[PF_{true}-\varepsilon, PF_{true}]$ over some norm.

Lemma 1. Under the condition of Assumption 1, given any non-empty solution set, at least one Pareto optimal solution exists within that set.

The proof of the lemma was presented by Van Veldhuizen and Lamont in Ref.[13].

Theorem 1 (**MOCGA convergence**). MOCGA with an infinite population converges to the global optimal of an MOP with probability one. Namely, for a Pareto front F^*, which is composed of at most an infinite number of vectors $V_1^*, V_2^*, \ldots, V_n^*$ such that:

$$\mathrm{P}\{\lim_{t\to\infty} F^* \in P_{\mathrm{curren}}\mathrm{t}(t)\} = 1 \quad (17)$$

Proof. It was proved that an EA would converge with probability one if it could fulfill the following two conditions[13]: (1) $\forall F, F' \in I, F'$ is reachable from F by means of mutation and recombination; (2) the population sequence $P(0), P(1), \ldots$ is monotone, i.e., $\forall t$:

$$\mathrm{Min}\{\Phi(F(t+1))|F(t+1)\in P(t+1)\} \leqslant \mathrm{Min}\{\Phi(F(t))|F(t)\in P(t)\} \quad (18)$$

MOCGA is assumed with infinite precision only provided that it operates on real value, a minimization MOP, an infinite population size, and appropriate mutation and crossover operators allowing every point in a search space to be visited. Thus $\forall F, F' \in I, F'$ is reachable from F. MOCGA satisfies Bäck's condition 1.

In order to prove the monotone character of the population sequence in MOCGA, both the fitness function and selection in it should be guaranteed monotone. When using Pareto-based fitness assignment, any given pair of Pareto optimal solutions receives identical fitness values; they also receive better fitness than dominated solutions. Therefore any fitness function assigning fitness in this way like MOCGA is monotonic.

Idealistically MOCGA sorts individuals in the current population of an infinite size. The produced minimal dominance set by ELECTRE method forms $P_{current}(t)$, which is at least composed by local Pareto-optimal solutions and always a not-empty set. After $P_{current}(t) \cup P_{known}(t)$ is mixed and dominated solutions in the compound are removed, the new $P_{known}(t+1)$ can be considered as a part of P_{true}, i.e. $P_{known}(t+1) \subset P_{true}$. Thus MOCGA satisfies Assumption 1.

The selection of MOCGA is naturally classified as $(\mu + \lambda)$ selection strategy. In the plus strategy an elitist selection strategy is applied, which chooses μ best individuals from λ children and μ parents together as parents for the next generation. According to Lemma 1, at least one Pareto optimal solution exists within $P_{known}(t)$. The solution is held to $P_{known}(t+1)$ so that the best fitness at the $(t+1)$th generation is at least equal to the one at the tth generation. Moreover the better solutions would be appeared in $P_{current}(t)$ because of crossover and mutation, which obtain better fitness in $P_{known}(t+1)$ after $P_{current}(t) \cup P_{known}(t)$. The evolutionary results at $(t+1)$th generation are not at least inferior to the ones at the tth generation. Consequently, selection strategy in MOCGA satisfies monotone.

MOCGA are qualified for the above two conditions; therefore, it converges to the global optimum of an MOP with probability one. Theorem 1 holds.

6 Conclusions

The paper proposes a feasible multi-objective evolutionary algorithm based on preferences for multi-objective optimization problems. The algorithm differs from the existing MOEAs as it uses a different selection strategy by the weak dominance comparison between any two individuals. From the results of test problems in the literature, the proposed MOCGA has the ability to scale up its performance in a large number of objectives. Property of MOCGA about stochastic convergence is analyzed. Experimental results of the multi-objective optimization benchmark problems demonstrate the validity of the new algorithm.

References:

[1] Van Veldhuizen DA, Lamont GB. Multi-Objective evolutionary algorithms: Analyzing the State-of-the-Art. IEEE Trans. on Evolutionary Computation, 2000,8(2):125−147.

[2] Coello CAC. List of Reference on Evolutionary Multi-objective Optimization. http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

[3] Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multi-objective optimization. Evolutionary Computation, 1995,3(1):11−16.

[4] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation, 1997,1(1): 67−82.

[5] Fonseca CM, Fleming PJ. Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In: Stephanie Forrest, ed. Proc. of the 5th Int'l Conf. on Genetic Algorithms. University of Illinois at Urbana-Champaign: Morgan Kauffman Publishers, 1993. 416−423.

[6] Zitzler E, Thiele L. Multi-Objective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. on Evolutionary Computation, 1999,3(4):257−271.

[7] Deb K, Agarwal S, Pratap A, Heyarivon T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation, 2002,6(2):182−197.

[8] Dragan C, Ian CP. Designer's preferences and multi-objective preliminary design processes, In: Ian CP, ed. Proc. of the 4th Int'l Conf. on Adaptive Computing in Design and Manufacture. London: Springer, 2000. 249−260.

[9] Dragan C, Ian CP. Preferences and their application in evolutionary multi-objective optimization. IEEE Trans. on Evolutionary Computation, 2002,6(1):42−57.

[10] Coello CAC. Handling preferences in evolutionary multi-objective optimization: A survey. In: Coello CAC, ed. Proc. Congress of Evolutionary Computation. New Jersey: IEEE Service Center, 2000. 30−37.

[11] Triantaphyllou E, Shu B, Nieto SS. Multi-Criteria decision making: An operations research approach. In: Webster JG, ed. Encyclopedia of Electrical and Electronics Engineering. New York: John Wiley & Sons, 1998. 175−186.

[12] Deb K, Thiele L, Laumanns M, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Yao X, ed. Proc. of the 2002 Congress on Evolutionary Computation. New Jersey: IEEE Service Center, 2002. 825−830.

[13] Van Veldhuizen DA, Lamont GB. Evolutionary computation and convergence to a pareto front. In: Koza JR, ed. Late Breaking Papers at the Genetic Programming 1998 Conf. California: Stanford University Bookstore, 1998. 221−228.