The multiobjective evolutionary algorithm based on determined weight and sub-regional search

Hai-lin Liu, Xueqiang Li

Abstract—By dividing the multiobjective optimization of the decision space into several small regions, this paper proposes multi-objective optimization algorithm based on sub-regional search, which makes individuals in same region operate each other by evolutionary operator and the information between the individuals of different regions exchange through their offsprings re-divided into regions again. Since the proposed algorithm utilizes the sub-regional search, the computational complexity at each generation is lower than the NSGA-II and MSEA. The proposed algorithm makes use of the max-min strategy with determined weight as fitness functions, which make it approach evenly distributed solution in Pareto front. This paper presents a kind of easy technology dealing with the constraint, which makes the proposed algorithm solved unconstrained multiobjective problems can also be used to solve constrained multiobjective problems. The numerical results, with 13 unconstrained multiobjective optimization testing instances and 10 constrained multiobjective optimization testing instances, are shown in this paper.

I. INTRODUCTION

Multiobjective optimization is a usual problem in the fields of science and technology, and management science, etc. Different from singleobjective optimization, multiobjective optimization has more than one objective, and the objectives are often conflicting. Therefore, multiobjective optimization is to find in some sense nondominated solutions about its vector objective functions, which are often many or infinite.

The model of multiobjective optimization problem is as follows:

\[
\begin{align*}
\min & \quad f(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\
\text{s.t} & \quad g_i(x) \geq 0 \quad i = 1, 2, \ldots, k \\
& \quad x \in X \subseteq \mathbb{R}^n
\end{align*}
\]

where \(x\) is decision variable, \(f(x)\) is objective vector, \(X\) is super rectangular region in \(\mathbb{R}^n\) and expresses decision space, \(g_i(x)\) is constraint function \((i = 1, 2, \ldots, k)\).

The merits solving multiobjective optimization problem by evolutionary algorithm attract wide attentions of scholars, and there have been many results of research in the past 20 years [1]. Besides design of evolutionary operators, the performance of multiobjective evolutionary algorithms largely depended on how to assign fitness of individuals. The existing fitness assignment strategies mainly include fitness assignment based on non-dominated sort [9], domination [7], alternative objectives [6], and aggregating objectives [2]-[5],[8],[12]. In most of algorithms based on aggregating objectives (the weighted sum of objective function and the max-min strategy with weight), the weights are generated randomly [4], [5], [8], [12]. In order to lead evolutionary algorithm better to approach evenly distributed solution in Pareto front, the literatures [2], [3] gave a method choosing weights of fitness function based on max-min strategy by spherical coordinate transformation. The complexity of MSEA algorithm in the literature [3] is \(O(mKN) + O(mN\log N)\) (in which \(m\) is the number of objective, \(N\) is population size, and \(K\) is the number of weight vectors). When \(K\) is smaller, the algorithm has a faster convergence velocity. However, when \(K\) is bigger, the complexity of the algorithm will increase. In order to decrease the complexity of algorithm and improve its convergence performance, we propose an improved algorithm about MSEA. By choosing the appropriate \(T\) groups of weight vectors in max-min strategy, the objective space is divided into \(T\) regions, and corresponding decision variable space is also divided into \(T\) regions. The newborn individuals which come from crossover and mutation between two individuals in the same region are divided into \(T\) regions according to some particular rules. The next generation is selected by a group of fitness functions with the uniform designed weight vectors. A new kind of crossover and mutation operators is designed in the improved MSEA algorithm, which largely enhance performance of the algorithm and reduce its complexity. This paper presents a kind of easy technology dealing with the constraint which makes the proposed algorithm solved unconstrained multiobjective problem can also be used to solve constrained multiobjective problem. In this paper, 13 unconstrained multi-objective optimization testing instances and 10 constrained multiobjective optimization testing instances are simulated by computer, and the effectiveness of proposed algorithm is investigated.

The remainder of this paper is organized as follows: In section 2, the way of dividing the search area is introduced. Section 3 gives the design of evolutionary operator, section 4 shows the framework and specialty of proposed algorithm. In section 5, the numerical experiments of unconstrained and constrained multiobjective problem are introduced. The conclusion is in section 6.

II. THE DIVISION OF SEARCHING REGION

In order to reduce the computing cost and improve efficiency of the searching, we divide weight vectors and individuals into \(T\) classes. In selection step, we firstly select a certain number of new individuals for each class and compare these new individuals with the exist individuals in same class.
If the new individual is better than the existing individual, we substitute it. Meanwhile, in order to keep the selection pressure not too high, we define an external set for each class and update the individuals in external sets during the process of selection.

A. Construction of the Fitness function

In this paper, the fitness function utilizes the weighted max-min strategy which is similar to the one in literature [1], [2], the difference is that the weight vectors are calculated as follows:

\[
\begin{align*}
    w_1(x) &= \frac{1}{\cos \theta_1}, \\
    w_2(x) &= \frac{1}{\sin \theta_1 \cos \theta_2}, \\
    &\quad \ldots \\
    w_{m-1}(x) &= \frac{1}{\sin \theta_1 \sin \theta_2 \ldots \sin \theta_{m-2} \cos \theta_{m-1}}, \\
    w_m(x) &= \frac{1}{\sin \theta_1 \sin \theta_2 \ldots \sin \theta_{m-2} \sin \theta_{m-1}}
\end{align*}
\]

where \((\rho, \theta_1, \theta_2, \ldots, \theta_{m-1})\) are the generalized spherical coordinates in objective space. Suppose the intersection point of the unit hypersphere denoted as \(T\). Through the \(t\)th class and put these individuals into population. The remaining \(5l_t\) individuals constitute the external set of \(t\)-th class.

B. Division of weight vectors and individuals

Suppose the size of population is \(N\), we can design \(N\) weight vectors \(w^1, w^2, \ldots, w^N\) from Eq.(2), and denote their corresponding points on the unit hypersphere as \(A_1, A_2, \ldots, A_N\). Similarly, we can design \(T\) weight vectors \((T \leq N)\), corresponding evenly distributed points on the unit hypersphere denoted as \(A_1, A_2, \ldots, A_T\). Through calculating Euclidean distance between each point \(A_i (i \in \{1, 2, \ldots, N\})\) with points \(A_1, A_2, \ldots, A_T\), we may find the nearest point from \(A_i\) in the group of points. Suppose the point is \(A_t (t \in \{1, 2, \ldots, T\})\), the weight vector \(w^i (i \in \{1, 2, \ldots, N\})\) is classified into \(t\)-th class. Obviously, the total number of classes is \(T\).

In the initial step of the algorithm, we uniformly and randomly generate \(6N\) individuals and compute the objective function values of all these individuals. Through Eq.(4), we can calculate the transformed objective vector for each individual and Euclidean distances between its unit vector and points \(A_1, A_2, \ldots, A_T\). Assuming that the number of weight vectors is \(l_t\) in \(t\)-th class, we select \(6l_t\) individuals from \(6N\) individuals, in which the Euclidean distances calculated above to point \(A_t\) is the minimum, and classify the selected \(6l_t\) individuals into \(t\)-th class \((t = 1, 2, \ldots, T)\).

In the case of unconstrained multiobjective optimization problem, since there are \(l_t\) weight vectors in \(t\)-th class as above, we can get \(l_t\) fitness functions from Eq.(3). Furthermore, we can select a best individual from the \(6l_t\) individuals corresponding to \(t\)-th class with each fitness function. Selected total \(l_t\) different individuals are put into population according to \(l_t\) fitness functions. The remaining \(5l_t\) individuals constitute the external set of \(t\)-th class \((t = 1, 2, \ldots, T)\).

In the case of constraint multiobjective optimization problem, in order to facilitate estimating the degree of violating constraints about individuals, we deal with constraints as follows: Let

\[
q(x) = \sum_{i=1}^{k} \min\{g_i(x), 0\}.
\]

When the number of individuals satisfied constraint condition are not less than the number of weight vectors \(l_t\) in \(t\)-th class, similar to the unconstraint case, we select \(l_t\) different individuals from individuals satisfied constraints condition by using fitness function corresponding to each weight vector in \(t\)-th class and put these individuals into population. The remaining \(5l_t\) individuals constitute the external set of \(t\)-th class. If the number of individuals satisfying constraint condition in \(t\)-th class is less than \(l_t\), we select \(l_t\) individuals, of which the values of \(q(x)\) are maximum, from the \(6l_t\) individuals in \(t\)-th class. Then according to each weight vector in the \(t\)-th class, we select \(l_t\) different individuals from above the \(l_t\) individuals and put these individuals into population. Unselected individuals constitute external set of \(t\)-th class \((t = 1, 2, \ldots, T)\).

III. THE DESIGN OF EVOLUTIONARY OPERATORS

A. Crossover and mutation

Usually, the Euclidean distance between two individuals in same class is very close. Thus, the crossover operation between them can largely play a role exploring the good individuals in this area. However, the crossover operation only performing between the individuals of the same area is not very useful for exploring a wider area. The algorithm only by mutating to exploit the new area will not be very efficient. Therefore, we perform the crossover between an individual and another individual which are randomly selected from its corresponding external set. Suppose individual \(x^i\) of population belongs to the \(t\)-th class \(t \in \{1, 2, \ldots, T\}\), choosing the \(x^j\) at random in \(t\)-th class external set, after crossover, the new individual is generated as follows:

\[
\tilde{x}(x) = x^i + re \cdot (x^j - x^i),
\]

where \(re = rand \cdot (1 - rand^{-\frac{\text{max_gen}}{\text{gen}}})^{0.7}\), \(rand\) is a random number in \([-1, 1]\), \(\text{gen}\) is the current generation and \(\text{Max_gen}\) is the maximum generation. The defined \(re\) can gradually approach to zero with increasing evolutionary generation similar to simulated annealing. If \(k\)-th component
the best, into

\( x^c \) is out of the boundary of domain \( X \), namely, if \( x^c \) smaller than the low boundary \( lb(k) \), \( x^c = lb(k) + 0.5 \cdot r \cdot \text{rand} \cdot (x^c - lb(k)) \). If \( x^c \) bigger than the upper boundary \( ub(k) \), \( x^c = ub(k) - 0.5 \cdot r \cdot \text{rand} \cdot \) \( (lb(k) - x^c) \), where \( \text{rand} \) is a random number in \([0, 1] \).

After crossover, every component in \( \tilde{x}^c \) is mutated with the probability of \( P_m \) and \( \tilde{x}^c \) is mutated once at least. If \( h \)-th component of \( \tilde{x}^c \) turn into \( \tilde{x}^c_h \) after mutation, then

\[
x^c_h = \tilde{x}^c_h + r \cdot \text{rand} \cdot (\text{rand} - 1) \cdot (\text{rand} \cdot (\tilde{x}^c_h - lb(h)))
\]

where \( \text{rand} \) is a random number in \([-1, 1] \), \( x^c_h \) is out of \([lb(h), ub(h)] \), when \( x^c_h < \) \( lb(h) \), \( x^c_h = lb(h) + 0.5 \cdot r \cdot \text{rand} \cdot (\tilde{x}^c_h - lb(h)) \); when \( x^c_h > ub(h) \), \( x^c_h = ub(h) - 0.5 \cdot r \cdot \text{rand} \cdot (ub(h) - \tilde{x}^c_h) \); \( \text{rand} \) is a random number in \([0, 1] \), \( x^c \) is the new individual after crossover and mutation by \( x^c \).

B. Selection operator

In the algorithm generation \( \text{gen} \), \( N \) new individuals are generated after crossover and mutation. Calculate unit vector function values of all these individuals in the objective space and the Euclidean distances from the unit vectors to points \( \tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_T \). Suppose the number of weight vectors is \( l_t \) in \( t \)-th class, select \( 3l_t \) individuals, of which distances to the \( \tilde{A}_t \) are shortest, from the \( N \) new individuals. Suppose these individuals are \( x^{t1}, x^{t2}, \ldots, x^{t3l_t} \), then classify them into \( t \)-th classes \((t = 1, 2, \ldots, T) \). Suppose there exist \( l_t \) weight vectors \( W^{t1}, W^{t2}, \ldots, W^{t3l_t} \) in the \( t \)-th class and \( W^{tj} \) correspond to the individual \( x^{tj} \) of which fitness is the best, \( W^{tj} = (w^{tj1}, w^{tj2}, \ldots, w^{tjN})^T \) \((j = 1, 2, \ldots, l_t) \).

Selection operator is divided into following two cases.

1) The unconstrained multi-objective optimization problem: Calculate the fitness functions \( F_{t1}(x^{t1}), F_{t2}(x^{t2}), \ldots, F_{t3l_t}(x^{t3l_t}) \) of the individuals \( x^{t1}, x^{t2}, \ldots, x^{t3l_t} \) corresponding to weight vector \( W^{t1} \), and select the best individual \( x^* \). If \( F_{t1}(x^*) \leq F_{t1}(x^{t1}) \), let \( x^* \) exchange position with \( x^{t1} \), and \( x^* \) belongs to \( \{x^{t1}, x^{t2}, \ldots, x^{t3l_t}\} \). Else, \( x^* \) is unchanged. Similarly, we utilize \( F_{tj}(x) \) corresponding to weight vector \( W^{tj} \) to select best individual and to update \( x^{tj} \) \((j = 2, 3, \ldots, l_t) \). The \( 3l_t \) individuals randomly selected in the \( t \)-th external set are replaced by the individuals \( x^{t1}, x^{t2}, \ldots, x^{t3l_t} \).

2) The constrained multi-objective optimization problem: We combine the individuals \( \{x^{t1}, x^{t2}, \ldots, x^{t3l_t}\} \) corresponding to \( W^{t1}, W^{t2}, \ldots, W^{t3l_t} \) respectively in \( t \)-th class and newly generating individuals \( x^{t1}, x^{t2}, \ldots, x^{t3l_t} \) together. Similar to method selecting initial individuals of constrained multi-objective problem, \( l_t \) different best individuals are selected from individuals satisfied constraint condition by fitness functions \( F_{t1}(x), F_{t2}(x), \ldots, F_{t3l_t}(x) \) when the number of individuals satisfied constraint condition is not less than \( l_t \). Otherwise, these individuals are selected from \( l_t \) individuals obtaining largest values of \( q(x) \). When ending the selection of \( l_t \) best individuals in \( t \)-th class, unselected individuals randomly replace same number of individuals in the \( t \)-th class external set.

IV. THE FRAMEWORK AND CHARACTERISTICS OF MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

A. The main framework of algorithm

We bring forward the framework of algorithm based on above chapters.

Step 1. Initialization: Set the population size \( N \), the number of class \( T \), the number of evolution generation \( \text{gen} \) and the maximum number of evolution generation \( \text{Max_gen} \), give the probability \( P_m \) in mutation;

Step 2. Design the weighted vectors \( w^1, w^2, \ldots, w^N \) according to Eq. (2), classify all the weighted vectors into \( T \) classes, and set the representative point \( \bar{A}_t \) in each category;

Step 3. Initialize the population, external set and the minimum value of each subobjective \( f^*_i (i = 1, 2, \ldots, m) \);

Step 4. Classify \( N \) new individuals which are generated from crossover and mutation into each category in accordance with the above-mentioned manner;

Step 5. Update the population, external set and the minimum value of each subobjective \( f^*_i (i = 1, 2, \ldots, m) \) by the above-mentioned selection operator in current generation;

Step 6. If \( \text{gen} \leq \text{Max_gen} \), go to step 4; otherwise, stop.

B. The characteristics of algorithm

a. In the proposed algorithm, individuals in population are divided into \( T \) classes in each generation and each class corresponds to a external set. Each individual in population crosses with randomly selected individual from the external set of its corresponding to the class. Since individuals of external set are around any individual in population corresponding to the class, the crossover between two adjacent individuals serves as local search and the crossover between two individuals whose distance is far serves as exploiting new region.

b. The proposed algorithm makes use of the max-min strategy with determined weight as fitness functions, which can maintain the diversity of population and approach the evenly distributed Pareto optimal solutions (For more information, please see literature [3]). The complexity of the algorithm is reduced by dividing the weight vectors and individuals into some regions, and the complexity of the algorithm is \( O((T + m + 2(n + 1))N) \) in each generation, where \( N \) is the size of population, \( m \) is the dimension of objective space, \( T \) is the number of class, \( n \) is the dimension of search space.

c. Since the proposed algorithm carry out selection operation in every generation after it assigns all the new individuals which are generated by crossover and mutation operation into \( T \) classes, the new individuals can be fully utilized. Furthermore, the selection operator can conveniently deal with the constraint.
V. COMPUTER SIMULATION

We simulate 13 unconstrained multiobjective test functions and 10 constrained multiobjective test functions from the literature [14] by computer. The parameters of algorithm are set as follows:

Set $P_m = \frac{1}{n}$, where $n$ is the dimension of search space. The population size $N$ is 100 for two objective problems, 150 for three objective problems, and 762 for five objective problems. The number of class $T$ is 15 for two objective problems, 33 for three objective problems, and 60 for five objective problems. Set $Max_{gen} = [\left(\frac{300000}{6N}\right)]$, where $[x]$ stands for the largest integer of not greater than $x$.

In order to measure the quality of solutions, the IGD metric [14] has been defined. Suppose $P$ be a set of uniformly distributed points in Pareto front, and $S$ be the approximate solutions got by different algorithm, so IGD metric is defined as follows:

$$IGD(S) = \frac{1}{n}. \sum_{i=1}^{n} d(v_i, S)$$  \hspace{1cm} (7)

where $v_i \in P$, $n$ is the number of points in $P$, and $d(v_i, S)$ is the minimum Euclidean distance between $v_i$ and the solutions in $A$.

The Configuration of computer in simulation are System: Microsoft Windows XP Professional sp3; RAM: 1G; CPU: T2050; CPU 1.60 GHz; Computer Language: Matlab 7.0.

Table I and Table II gives the IGD value and mean CPU time used by the algorithm in this paper for each test function in 30 independently runs. The Fig.1 and Fig.2 show the nondominated front with the lowest IGD value for each unconstrained and constrained test problems.

<table>
<thead>
<tr>
<th>Test Functions</th>
<th>IGD value</th>
<th>CPU times(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.004891</td>
<td>0.014017</td>
</tr>
<tr>
<td>UF2</td>
<td>0.008276</td>
<td>0.021584</td>
</tr>
<tr>
<td>UF3</td>
<td>0.004669</td>
<td>0.132567</td>
</tr>
<tr>
<td>UF4</td>
<td>0.041930</td>
<td>0.045230</td>
</tr>
<tr>
<td>UF5</td>
<td>0.100246</td>
<td>0.228266</td>
</tr>
<tr>
<td>UF6</td>
<td>0.060198</td>
<td>0.322172</td>
</tr>
<tr>
<td>UF7</td>
<td>0.005801</td>
<td>0.009778</td>
</tr>
<tr>
<td>UF8</td>
<td>0.066502</td>
<td>0.092058</td>
</tr>
<tr>
<td>UF9</td>
<td>0.047366</td>
<td>0.171044</td>
</tr>
<tr>
<td>UF10</td>
<td>0.240305</td>
<td>0.839926</td>
</tr>
<tr>
<td>UF11</td>
<td>0.123791</td>
<td>0.138966</td>
</tr>
<tr>
<td>UF12</td>
<td>2.91471297</td>
<td>636.066294</td>
</tr>
<tr>
<td>UF13</td>
<td>2.028863</td>
<td>2.423187</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

We define the fitness function by using a set of determined weight vectors and max-min strategy, divide the decision-making space and objective space into several small classes, and let individuals in the same class cross with each other. The information between individuals in different regions is exchanged by their offsprings which will be assigned to the region. In this paper, we propose the multiobjective evolutionary algorithm based on determined weight and sub-regional search. Because of making use of the partition searching, the complexity of algorithm is reduced and the convergence of algorithm is improved. The dealing with constraint problem becomes easier by selection operator and designing external set. The weight vector in fitness function is specially designed, so the approach lead by the fitness function is effective to obtain the evenly distributed Pareto optimal solutions on the Pareto front.

VII. ACKNOWLEDGMENT

This work was jointly supported by the Natural Science Foundation of Guangdong Province (No.07011797, No.815100901000044), the Science and Technology Project of Guangzhou (No.2007J1-C0501).

REFERENCES

Fig. 1. Plots of the nondominated front with the lowest IGD value for each unconstrained test problems.
Fig. 2. Plots of the nondominated front with the lowest IGD value for each constrained test problems.


