



A rapid convergent and multi objective genetic algorithm based on new genetic operation

Jinghui Gao, Rui Shan, Lingling Li and Hongfang Cui
College of Science, Yanshan University, Qinhuangdao066004, Hebei, China.

ARTICLE INFO

Article history:

Received: 12 June 2011;
Received in revised form:
18 July 2011;
Accepted: 28 July 2011;

Keywords

Improvement crossover operation,
Uniformity mutation operation,
Niche technology,
Multi-objective genetic,
Convergence.

ABSTRACT

The paper presented a method with an improvement crossover operation, reserve policy and Niche technology to adjust the algorithm, which made algorithm be avoided to converge prematurely, and prevented algorithm falling into best local solutions. Through case emulates, it was found that it had good performance. Finally, the principle of convergence was proved.

© 2011 Elixir All rights reserved.

Introduction

In fact, problems are composed by multi-objective, which is difficult to solve and define because the results from the problems are a group of solutions, called Pareto optimization solutions. Evolutionary algorithm theory forms multi-objective evolutionary algorithm in the field and has successful application in practice at the end of twentieth century^[1].

Recently, the application of genetic algorithm in multi-objective has been hotspot. Schaffer proposed vector appraising multi-objective genetic algorithm (VAGA), which divided the amounts of all sub object into several equal sub population and allocated sub object to each sub population. Each sub object was chose independently, and then we constituted a new sub population and began cross operation and mutation operation, repeated the process above, Pareto optimization solutions of problem will be solved. But the method had a low efficiency and didn't guarantee convergence^[2].

Deb proposed not an inferior select method (NSGA-II)^[3], which sorts individuals in Pareto values from low to high. The method was proved that it had a lower convergence speed. Li raised an improvement multi-objective micro-genetic algorithm (MPMGA)^[4], which improved the distribution of solutions and enhance the precision of solution.

Li presented a multi-objective genetic algorithm based on dynamical e-dominate (DEMOGA)^[5], which improved the distribution problems of boundary solutions through dynamical grid. Zhang ling designed a new crossover^[8] with theory and method in number introduction^[7] to improve the efficiency of GA, the advantage of algorithm was proved in one objective optimization problems.

In this paper, we sort individuals and take keeping policy due to genetic algorithm converge prematurely or not convergence, adjust the distance among individuals with niche radius, guarantee the variety of population with improvement crossover operation and uniformity mutation operation. It

prevents the algorithm converging prematurely and its convergence was proved.

Multi-objective problem definition

Consider minimizing problem^[1], which is composed by n objective components f_k ($k = 1, \dots, n$)

$$\min f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})), \mathbf{x} \subseteq R^n \quad (1)$$

When $\mathbf{x}_m \in U$ is decision-making variable, the conditions of \mathbf{x}_m is the optimal solution of Pareto is that inexistence decision-making variable, which make $\mathbf{x}_v \in U, \mathbf{v} = \mathbf{f}(\mathbf{x}_v) = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ dominate $\mathbf{u} = \mathbf{f}(\mathbf{x}_u) = (\mathbf{u}_1, \dots, \mathbf{u}_n)$. It means inexistence $\mathbf{x}_v \in U$, to make equality (2) true.

$$\forall i \in \{1, \dots, n\}, \mathbf{v}_i \leq \mathbf{u}_i \wedge \exists i \in \{1, \dots, n\} | \mathbf{v}_i < \mathbf{u}_i \quad (2)$$

Multi-objective problem is generally conflicting each other, which causes it impossible that numerous objective functions is attained at the same time. The method to solve it is that concerts and compromises all objects, to make they approximate to optimization only.

New multi-objective genetic algorithm

Improvement crossover operation

The quality of crossover operation is related with convergence speed of genetic algorithm. In traditional algorithm, many researchers take settled crossover probability to fail to succeed the excellent schema of sir generation by sub generation, even has a low convergence speed.

In this paper, we take improvement crossover operation, it means that whether crossover proceeding was decided by individuals' similarity level in^[11]. For the two bunch A and B, we define the similarity level of A and B as follows

Define similarity level:

$$s(i, j) = \frac{l}{n}, 0 < i \leq m, 0 < j \leq m, i \neq j \quad (3)$$

Where l is the longest commonality sub bunch of A and B, n is chromosomal bunch size, m is population scale.

For example, bunch A=10111100, bunch B=01101100, the longest commonality sub bunch is 1100, size is 4, chromosome length n is 8, so the similarity of A and B is 0.5.

We give threshold values $p : 0.5$, two individuals start to cross when the similarity level s between them is much less than p . In this paper, we take one-point crossover.

The code of the longer commonality sub order length as follows

```
Void crossover (char *p1, char *p2, float p)
```

```
{
int m=0, n=0, i =0;
float s;
char *s1= p1,*s2= p2;
while (*s1!='\0')
{
    if (*s1==*s2)
    {
        s1++;
        s2++;
    }
    else
    {
        s1=&(p1 [i++]);
        s2= p2;
    }
    n =strlen(s1);
    i=1;
    s1= p1;
    s2= p2;
while (*s1!='\0')
{
    if (*s1==*s2)
    {
        s1++;
        s2++;
    }
    else
    {
        s1= p1;
        s1[strlen (s1)- i] ='\0';
        s2= p2;
        i++;
    }
}
m =strlen (s1);
if (m > n)
{
    n = m;
}
s = n/strlen (p1);
if (s < p)
{
    start crossover operation;
}
}
```

Individuals would be restoration when theirs' fitness values are extraordinary close to each other to cause algorithm lingering. The crossover operation above keeps populations various, prevents repeated individuals turning up and stasis

status, which is a simple method without increasing whole algorithm's complexity.

Chromosomal Encode

Use real coding, that is to say the chromosome coding of

$$X = \{x_1, x_2, \dots, x_n\}^T \text{ is } x_1 x_2 \dots x_n.$$

Niche Operation

Encoding region and solution region are related with mapping, but the former can't guarantee the distances among solutions well-distributed. In this paper, we use niche radius to distribute the distances among solutions (individuals).

First, allocate niche radius parameter L , then compare distance in each two individuals, if the distance is smaller than L , force a doughty penalty function on individuals with secondary small fitness to reduce vastly its fitness. So they are easily weeded out and even exists an excellent individual within distance L , forming population's diversity and retaining the distance among solutions.

Adjust the distance of optional two individuals x_i, x_j as follows, which is inner of one population.

Consider

$$d(x_i, x_j) = \|X_i - X_j\| = \sqrt{\sum_{k=1}^p (x_{ki} - x_{jk})^2}, (k \leq p) \quad (4)$$

Where $i = 1, 2, \dots, N-1; j = i+1, j+2, \dots, N$. N is

initialized population amount ; $d(x_i, x_j)$ shows the *Hamming* distance between individual x_i and x_j here.

Consider

$$S[d(x_i, x_j)] = \begin{cases} 1 & d(x_i, x_j) = 0 \\ 0 & d(x_i, x_j) > \sigma_{share} \\ 1 - [d(x_i, x_j)]^\alpha / \sigma_{share} & 0 < d(x_i, x_j) \leq \sigma_{share} \end{cases} \quad (5)$$

σ_{share} is niche radius, α is used to adjust sharing function. The smaller sharing function value, the more similar among individuals.

Select Operation

Previous select operation methods (such rotate bet, expectation values model, etc.) all exist two problems as follows: (1) when few individuals have large fitness values in current population, select operation above will be genetic rapidly to cause converging prematurely. (2)

When inner population individuals' fitness values tend to the same size, it makes the former algorithm tend to randomly search algorithm and process stagnate.

In this paper, we take non-dominated select method, it means that sort individuals in theirs' fitness value and number them, which tells us that set individuals with large fitness first and ones with small fitness last. So we can select the sorting number according to some probability, the individuals with large fitness could be chose.

Fitness Function

Due to evaluate individuals' quality in solution region properly according to theirs fitness values, we design appraisal function as follows

$$F(f(x)) = \frac{f_{\max}(x) - f(x)^{r-1}}{f_{\max}(x) - f_{\min}(x)} Q \quad (6)$$

Where r is chromosomal order number, parameter $a \in (0,1)$, $Q = 200$ used to adjust the fitness value.

Crossover Operation

The detailed process is given in step 3.1.

Uniformity mutation operation

We design the effective algorithm code to show uniformity mutation operation:

```

While generation <= maxgen
function [g]=mutation(g, pm)
[m, n]=size (g);
ran =rand(1,m);
r=rand (1, 3);
rr=floor(n*rand(1,3)+1);
[x,mu]=find (ran<pm);
for k=1:length(mu)
    for i=1:length(r)
        umax (i)=n+1-rr(i);
        umin (i)=1;
        g(mu(k),rr(i))=umin(i)+floor((umax(i)-umin(i))*r(i));
    end
end
    
```

Keeping Policy

If maximum value of objective function of subgeneration is less than it of sirgeneration, replace the former for the latter, which outputs as optimization solutions. It makes algorithm converge rapidly.

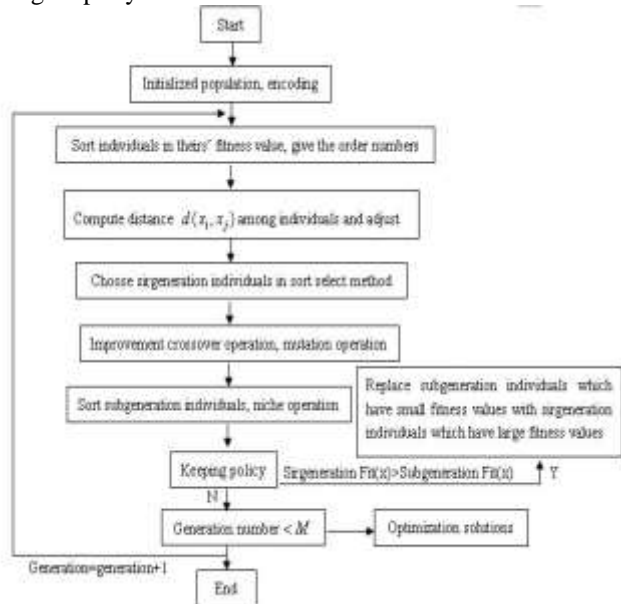


Fig.1 Algorithm flow chart

Numerical emulation

We take Shaffer function in [6] for comparing the numerical results different algorithms.

$$\begin{aligned}
 \min. F &= (f_1(x), f_2(x)) [\quad], \\
 f_1(x) &= x^2, f_2(x) = (x-2)^2, \\
 -10^3 &\leq x \leq 10^3.
 \end{aligned}
 \tag{7}$$

Parameters of genetic algorithm are given as follows:
 Population number: 80,
 Max generation (M): 40,
 Select probability : 0.8,
 Cross probability : 0.85,
 Mutation probability : 0.1, σ_{share} : 0.2, $Q = 200$.

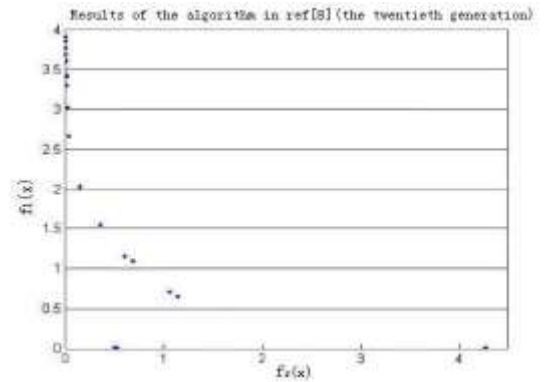


Fig.2 results of the arithmetic in [8].

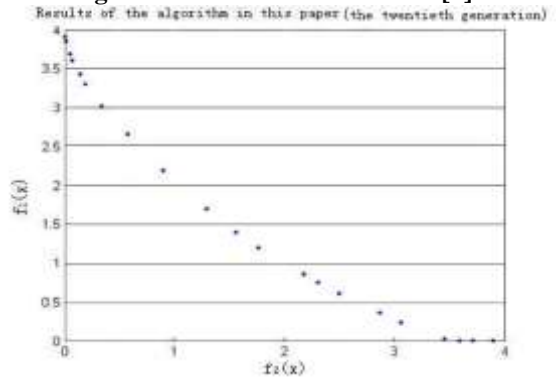


Fig.3 results of the arithmetic in this paper.

We can know from Fig.2 and Fig.3, the solutions(Fig.2) from the method in [8] distribute asymmetrically, leading to premature convergence. However, the solutions in Pareto collections from the algorithm in [6] disperse in feasible region, there is good convergence nature, but obvious no uniformity.

The optimization solutions [Fig.3] from the algorithm in this paper distribute uniformly in feasible region and prevent the algorithm falling into local search as well, which is better than the algorithms in [6] and [8]. The algorithm in this paper has a rapid convergence speed according to comparing evolutionary generations.

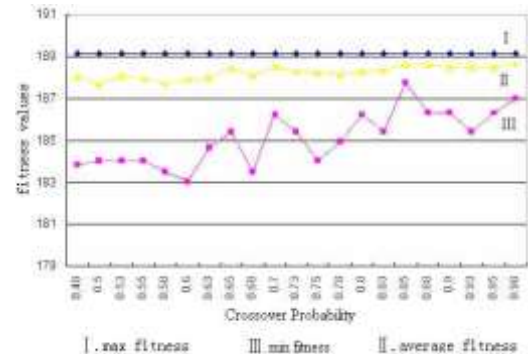


Fig. 4 F(x) change chart

Convergences

Define 1 Suppose $p(t)$ is the t th population created by multi-objective, $\| \cdot \|$ is the norm in feasible region space R^n , $p(t) = \{x_1(t), x_2(t), \dots, x_M(t)\}$, M is the maximum population scale, $F(t) = F(p(t))$ is the t th solutions in objective space corresponding to those in encode space ,if they satisfy

$$P\{\lim_{t \rightarrow \infty} |F(t) - F_{pareto}| \leq \varepsilon\} = 1 \tag{8}$$

Where F_{pareto} stands for front surface of *Pareto*, all solutions caused by algorithm should converge to F_{pareto} . So we define that algorithm converges to *Pareto* front surface F_{pareto} according to probability.

Lemma 1 If random order created by algorithm is neat and finite-state MARKOV, status transferring matrix is brief.

Theorem 1 If $p(t), t > 0$ is the stochastic population order caused by algorithm, not neat and finite-state MARKOV in limited status space,

$$P\{\lim_{t \rightarrow \infty} |F(t) - F_{pareto}| \leq \varepsilon\} = 1.$$

It is convergent according to probability.

Prove: we take keeping policy, which retains the individuals with front order number. If individuals $F(t)$ above enter into front surface of *Pareto*, they will keep in $F(t)$ for ever with evolution generation increasing, it means individuals with last order number will be replaced by *Pareto* optimal individuals.

$$P\{\lim_{t \rightarrow \infty} |F(t) - F_{pareto}| \leq \varepsilon\} = 1$$

It is true in probability in limited evolutionary generations.

Theorem 2 Algorithm converges to *Pareto* front surface F_{pareto} of minimum problem (1) in probability.

Prove: the feasible region of algorithm is limited according to fact. We know from Lemma 1 that the transferring matrix of MARKOV chain is neat. $p(t)$ is the population caused in some time. So we can obtain matrix $p^c(t)$ and $p^m(t)$ through crossover operation and second-evolution matrix $p^{c1}(t)$ and $p^{m1}(t)$. The total transferring matrix is

$$H = p^c(t) \cdot p^m(t) \cdot p^{c1}(t) \cdot p^{m1}(t) \quad (9)$$

$p^c(t), p^{c1}(t), p^m(t)$ and $p^{m1}(t)$ are positive matrix because of improved crossover operation and uniformity mutation, so transferring matrix H is positive matrix, however, a positive matrix must be a neat matrix. So the order $p(t)$ satisfies Lemma 1 and even the theorem 2 is true.

Conclusions

The paper keeps population divers with improvement crossover operation and holds the algorithm on with keeping policy to avoid algorithm falling into local search. In other way, restricts population increase limitless with niche radius to make fitness enhance increasingly and adjust the distribution of solutions in feasible region. Even we can spread algorithm into the optimization problems about the counts of object function are more than two.

References

- [1] Xiaoping Wang, Cao. Genetic Algorithm-Theorem, Application and Soft Achievement [M]. Xi'an: Xi'an Jiaotong University Press, 2006, 1.
- [2] Qinxia Yong, Guangqiu Huang, Zhanquan Wang. genetic algorithm and genetic programming—a search optimization technology[M]. Metallurgy Industry Press, 1997,4.
- [3] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Trans. on Evolutionary Computation, 2002, 6(2): 182-197.
- [4] Na Li, Daoerji R.Q. An Improvement Multi—objective Micro—genetic Algorithm [J]. Inner Mongol Industry University journal in NeiMengGu, 2009,28(1):19-22.
- [5] Ke Li, Zhen, Cong Zhou. Multi—objective Genetic Algorithm Based on Dynamical e- dominates [J]. Computer Engineering and Application, 2009, 45(1):69-72.
- [6] Yujia Rao, Cheng, Xia. etal. Multi—objective Genetic Algorithm Based on Fine—points [J]. Computer technology and development, 2008, 18(12):67-69.
- [7] Luogeng Hua, Yuan Wang. Number Introduction Application in Modern Times Analysis [M]. Beijing: Science Press, 1978.
- [8] Ling Zhang, Ba Zhang. A Fine—points Genetic Algorithm [J]. Computer journal, 2001, 24(9): 917-922.
- [9] Deb K. Multi-objective Genetic Algorithms : problem difficulties and construction of test Problems [J]. Evolutionary Computation, 1999, 7 (3) : 205-230.
- [10] Deb K, Thiele L, Laumanns M, etal. Scalable multi-objective optimization test problem [C]//Congress on Evolutionary Computation (CEC'2002), 2002,1: 825-830.
- [11] Jones P R M, West G, Harris D. The Loughborough Anthropometric Shadow Scanner (LASS) [J]. Endeavour, 1989, 13(4):164-168.