Multi-population Genetic Algorithms with Space Partition for Multi-objective Optimization Problems

Dun-wei Gong and Yong Zhou

School of Information and Electrical Engineering, China University of Mining and Technology,
Xuzhou, 221008 P. R. China

Summary

It is difficult for the existing multi-population genetic algorithms with space partition to be successfully applied to multi-objective optimization problems. Multi-population genetic algorithms with space partition for multi-objective optimization problems are designed in this paper in allusion to the characteristics of multi-objective optimization problems. A complicated optimization problem is converted into several simple optimization problems. Crossover operator for an intra-population evolution has a direction by using information from the super individual archive. The frequency of updating the super individual archive decreases via pre-selecting optimal solutions submitted to the super individual archive. The search scope of a population is expanded via an inter-population evolution. It is shown from analysis that the computational complexity of the algorithm in this paper decreases evidently. The efficiency of the algorithm in this paper is validated through a complicated benchmark multi-objective optimization problem.

Key words:

Genetic algorithms, Multi-population, Space partition, Multi-objective optimization problems.

1. Introduction

Genetic algorithms are stochastic, population-based optimization methods inspired notions of biological evolution and mechanisms of heredity and mutation. But a canonical genetic algorithm can only obtain local optima of the problems for its premature convergence. Hence many scholars improve it from several aspects and a multi-population genetic algorithm is a typical and important improving direction.

The multi-population genetic algorithms proposed in [1] and [2] generate and evolve multiple populations in a search space, and the significance in biology of population partition is not evident for not considering regionality of a population's living. Hence the efficiency of the algorithms should be further improved.

Genetic algorithms with space partition based on the regionality of a population's living have vivid biological characteristics. The multi-population genetic algorithms with region partition were presented in [3] to [5]. In general, these algorithms belong to hierarchical genetic algorithms, which generate different populations in different subspaces of a search space respectively. There are two kinds of evolutions, namely an intra-population evolution and an inter-population evolution. An inter-population evolution refers that genetic operators perform on the individuals that come from the same population, whereas an intra-population evolution refers that genetic operators perform on the individuals that come from different populations. The genetic operation of an intra-population evolution and an inter-population evolution, the method of conserving the best individual of each population and the approach to generating a new population were given in the above literature which applied them to many problems, such as neural network evolutionary design, optimization of TSP and complicated multi-modal functions.

Multi-objective optimization problems exist everywhere, such as multi-package optimization, multi-robot harmony path planning, route selection in communication and so on. Hence it is very important in both theory and practice to study multi-objective optimization problems. It began in the middle of the 1980s to solve multi-objective optimization problems with genetic algorithms, and these methods were well developed in the middle of the 1990s. They are new approaches to studying multi-objective optimization problems. In general, they include two aspects. The one is converting a multi-objective optimization problem into a weighted single objective optimization problem and then adopting genetic algorithms to optimize the single objective optimization problem. But this approach has fated weakness in that the selection of the weight of each objective is subjective and different weighted values correspond to different solutions. The other is studying the Pareto optimization problems, which adopts genetic algorithms to search the Pareto front and the Pareto optimal solutions of being optimized problem [6]. It is difficult for the multi-population genetic algorithms with region partition presented in [3] to [5] to be successfully applied to multi-objective optimization problems, for not considering the characteristics of the Pareto fronts and the

---

Manuscript revised January 2006.

Pareto optimal solutions of multi-objective optimization problems.

Multi-population genetic algorithms with space partition are designed in this paper in allusion to the characteristics of multi-objective optimization problems. Methodology of the algorithm, a method of partitioning a search space and designing genetic operators, a strategy for generating and updating the super individual archive and an approach to generating a new population are presented. The computational complexity of the algorithm is analyzed. The algorithm is applied to a benchmark multi-objective optimization problem and its validity is shown.

2. Multi-population Genetic Algorithms with Space Partition for Multi-objective Optimization Problems

2.1 Methodology of Algorithms

A search space is partitioned into several subspaces, and each subspace is assigned to one population to evolve. Different genetic algorithms are applied to an intra-population evolution and an inter-population evolution respectively to generate a new population or new individuals so as to cover new subspaces. A new method of submitting the Pareto optimal solutions and a new strategy for generating and updating the super individual archive are adopted to guarantee finding the Pareto optimal solutions efficiently that satisfy better distribution, approach and extension.

2.2 Partition of Search Space

A search space is partitioned into a number of subspaces according to the dimension and range of the problem to be optimized. To a search space, such partition should be complete and each subspace should be separate [4], namely, if $P_1, P_2, \cdots, P_n$ denotes a kind of partition of a search space *P*, then $\bigcup_{i=1}^{n} P_i = P; \forall 1 \le i, j \le n, P_i \cap P_j = \phi$ is satisfied.

2.3 Design of Intra-population Evolution Genetic Algorithms

In order to reduce the length of coding and the workload resulting from encoding and decoding, real coding is adopted to encode decision variables in this paper. Without loss of generality, a problem to be considered here assumes minimization of a function with *m* objectives in a *d*-dimensional decision space.

A number of initial populations are generated in some subspaces of a search space and each subspace is assigned to one population to evolve. Because real coding is adopted, a subspace that an individual belongs to can be directly determined according to the coding form of the individual and the range of the subspace, and then a population that the individual belongs to is thus determined.

Pareto optimal individuals are conserved to the next generation at every evolutionary generation. Pareto optimal individuals are submitted to the Pareto optimal solution set of its own population according to a certain criterion in every population evolutionary period, and it is also submitted to the super individual archive that decides whether accepts it or not according to the criterion given in 2.4.

For an individual being real coding, arithmetic crossover operator is adopted. The closed crossover avoidance strategy with real coding is adopted in crossover operator, and information of super individual is sufficiently used so that offspring have better distribution and approach, hence increasing the efficiency of crossover operator.

Let's discuss according to the following 2 cases:

(i) If there is no individual in the super individual archive $S(t)$ belonging to P_k, then choose two parents x_k^1, x_k^2 in P_k, which satisfy:

$$\left\| f(x_k^1) - f(x_k^2) \right\| > \alpha_k^1 \max_{x_k^i, x_k^j \in P_k} \left\| f(x_k^i) - f(x_k^j) \right\| \tag{1}$$

and generate two offspring $x_k^{1'}, x_k^{2'}$ via:

$$\begin{aligned} x_k^{1'} &= \alpha_k^2 x_k^1 + (1-\alpha_k^2) x_k^2 \\ x_k^{2'} &= \alpha_k^2 x_k^2 + (1-\alpha_k^2) x_k^1 \end{aligned} \tag{2}$$

where $\|\cdot\|$ denotes a kind of norm and $\alpha_k^1, \alpha_k^2 \in (0,1)$.

(ii) If there is an individual $x_k^s, s \in I$ in $S(t)$ belonging to P_k, where I is the set of the individuals' index in $S(t)$ belonging to P_k, then generate offspring $x_k^{1''}$ from parent x_k^1 via:

$$x_k^{1''} = \alpha_k^3 x_k^1 + (1-\alpha_k^3) x_k^s \tag{3}$$

where $\alpha_k^3 \in (0,1)$. Further, if $|I| > 1$, then generate several offspring $x_k^{1'''}$s via:

$$x_k^{1'''} = \alpha_k^4 x_k^1 + (1-\alpha_k^4)\beta_k \left(x_k^{s_1} - x_k^{s_2} \right) \qquad s_1, s_2 \in I \tag{4}$$

where $\alpha_k^4, \beta_k \in (0,1)$.

Individuals of an intra-population evolution adopt Gauss mutation that was expounded in detail in [7].

An intra-population evolution in P_k is terminated when it is mature and unsatisfied with deleting condition.

2.4 Generation and Update of Super Individual Archive

Let $S_k(t)$ be the Pareto optimal solutions set generated via an intra-population evolution in P_k at the *t*-th generation. $\forall x_k^1, x_k^2 \in S_k(t)$, if:

$$\left\|x_k^1 - x_k^2\right\| < \gamma_k \max_{x_k^i, x_k^j \in P_k} \left\|x_k^i - x_k^j\right\| \tag{5}$$

then delete x_k^1 or x_k^2 from $S_k(t)$, which makes the Pareto optimal solutions in $S_k(t)$ have better distribution and not overcrowded so as to reduce the frequency of updating the super individual archive, where $\gamma_k \in (0,1)$ is a crowded coefficient.

Let the capacity of the super individual archive satisfy $|S(t)| \le S_{size}$. $\forall x_k \in S_k(t)$, the relationship between x_k and $S(t)$ has 3 kinds, namely, x_k dominates an individual *x* in $S(t)$, an individual *x* in $S(t)$ dominates x_k, and x_k non-dominates any individual *x* in $S(t)$ and vice versa.

Let's discuss them respectively as follows [8]:

(i) If x_k dominates an individual *x* in $S(t)$, that is to say $\exists x \in S(t), \ni x < x_k$, then delete *x* from $S(t)$ and accept x_k.

(ii) If an individual *x* in $S(t)$ dominates x_k, namely, $\exists x \in S(t), \ni x_k < x$, then not submit x_k to $S(t)$.

(iii) If x_k non-dominates any individual *x* in $S(t)$ and vice versa and $|S(t)| < S_{size}$, then accept x_k. Otherwise, choose an individual $x_0 \in S(t)$ satisfying:

$$\left\|x_k - x_0\right\| = \min_{x \in S(t)} \left\|x_k - x\right\| \tag{6}$$

then delete x_0 from $S(t)$ and accept x_k.

2.5 Design of Inter-population Evolution Genetic Algorithms

If there exists a subspace $P_k \subset P, k \in \{1,2,\cdots,n\}$ without any evolution population in it or the population size in it is less than M_0, then perform an inter-population evolution one generation to generate a population in P_k after performing an intra-population evolution a number of generations. Here two subspaces $P_i, P_j, i, j \in \{1,2,\cdots,n\}$ closest to P_k are selected to perform an inter-population evolution. Non-mutation genetic algorithms are applied to an inter-population evolution.

In order to guarantee offspring generated via an inter-population evolution distributing as much as in P_k and improve the efficiency of crossover operator, the probability of an individual being selected is in inverse ratio with the distance between it and P_k when selecting individuals in P_i, P_j, namely, $\forall x_i^l \in P_i$, the probability of x_i^l being selected is:

$$p\left(x_i^l\right) = \frac{\left\|x_i^l - P_k\right\|^{-1}}{\sum_l \left\|x_i^l - P_k\right\|^{-1}} \tag{7}$$

where $\left\|x_i^l - P_k\right\|$ denotes the distance between x_i^l and P_k.

For individuals being selected $x_i \in P_i, x_j \in P_j$, generate new individuals via:

$$\begin{aligned} x_k^1 &= \lambda x_i + (1-\lambda) x_j \\ x_k^2 &= \lambda x_j + (1-\lambda) x_i \end{aligned} \tag{8}$$

where λ is a parameter whose value is different in different cases. Individuals in P_k are generated via adjusting λ dynamically till the population size in P_k is larger than a given value.

2.6 Generation of New Population

There are 2 approaches to generate a new population. One is via an inter-population evolution that is a main approach to generate new population individuals, the other is via mutation operator of other intra-population evolution and individuals generated via this approach are very limited. The new generated population is called young population [3]. No matter what approach is adopted, an inter-population evolution performs only when its population size is not less than M_0.

2.7 Steps of Algorithms

(i) Partition a search space into *n* subspaces according to the dimension and range of the problem to be optimized.

(ii) Let *t*=0, select $n_g(t)$ subspaces and generate initial populations in them, let the population size be M, hence the whole population size in the search space is $n_g(t) \cdot M \cdot$

(iii) Assign values to the parameters of genetic operators, including probabilities of intra-population crossover, intra-population mutation and inter-population crossover and population evolution period T. Set the super individual archive empty.

(iv) Perform an intra-population evolution T generations on the population in each subspace of the $n_g(t)$ subspaces.

(v) Update the super individual archive according to the criterion in 2.4.

(vi) Judge whether the termination condition is satisfied or not, if yes, then go to (viii).

(vii) Select a number of subspaces in $n-n_g(t)$ not searched spaces and some populations satisfying given condition in 2.5, and generate new populations via inter-population evolutions until their size is M, go to (iv).

(viii) Submit the Pareto optimal solutions of the super individual archive and end the algorithms.

3. Computational Complexity Analysis of Algorithms

It has been proven by Deb that the computational complexity of NSGA-II is $O(m\times N^2)$ [9]. The computational complexity of the algorithm in this paper is analyzed as follows.

For $P_i, i=1,2,\ldots,n$, the computational complexity of the algorithm is $O(m\times N^2)$. In the worst case, the whole computational complexity of the algorithm is: $O(n\times m\times N^2)$.

For P_i, $i=1,2,\ldots,n$, in order to increase the distribution of the solutions, the distance between each pair of optimal solutions need compute. The computational complexity resulting from it is:

$$O(m\times(N-1)\times(N-2)\times\cdots\times 1)=O(\frac{m\times N\times(N-1)}{2})=O(\frac{m\times N^2}{2})\cdot$$

Then the computational complexity of n subspaces resulting from the computation of the distances is: $O(\frac{n\times m\times N^2}{2})\cdot$

For the process of updating the super individual archive, the Pareto optimal solutions in each subspace need compare with the solutions in the super individual archive. In the worst case, the computational complexity resulting from updating the super individual archive is $O(n\times m\times N\times S_{size})$. Hence the computational complexity of the algorithm in this paper is:

$$O(n\times m\times N^2)+O(\frac{n\times m\times N^2}{2})+O(n\times m\times N\times S_{size})\cdot$$

For S_{size} is small and updating the super individual set does not perform at every generation, the computational complexity resulting from it can be neglected compared with that of each subspace. Hence the computational complexity of the algorithm in this paper is approximate to: $O(n\times m\times N^2)+O(\frac{n\times m\times N^2}{2})\cdot$

Compared with NSGA-II, in the case of the two algorithms having same population size, the ratio of the computational complexity of the algorithm in this paper to that of NSGA-II is:

$$\frac{O(n\times m\times N^2)+O(\frac{n\times m\times N^2}{2})}{O(m\times(nN)^2)}=\frac{O(\frac{3}{2}\times n\times m\times N^2)}{O(m\times n\times n\times N)^2)}=\frac{3}{2n} \qquad (9)$$

It can be shown from formula (9) that when the number of subspaces is 3, the efficiency of the algorithm in this paper improves 50% compared with that of NSGA-II. Hence the algorithm in this paper can improve computational efficiency notably.

4. Application in Optimization of Numerical Functions

The following multi-objective optimization problem is considered.

Minimization:

$$\begin{aligned} f_1(x) &= 1-e^{-4x_1}\sin^6(6\pi x_1) \\ f_2(x) &= g(x)\left(1-\frac{f_1(x)}{g(x)}\right)^2 \end{aligned} \qquad (10)$$

where $g(x)=1+4\left(\sum_{i=2}^{6}x_i/4\right)^{0.25}$, $0\le x_i\le 1, i=1,2,\cdots,6\cdot$

In order to analyze computational results quantitatively, C-measure and SP-measure are used [11]. C-measure compares domination of the two different Pareto optimal solution sets. $C(x_1,x_2)$ denotes the rate of the number of the Pareto optimal solutions of x_2 dominated by x_1 to the number of the Pareto optimal solutions of x_2. The larger $C(x_1,x_2)$, the better the domination of x_1 to x_2 is. SP-measure denotes the distribution of the Pareto optimal solution set in an objective space. The smaller SP, the better the distribution is.

4.1 Parameters Setting

The parameters setting is listed in Table 1. Let $\alpha_k^i=0.3$, where $i=1,2,3,4$, $k=1,2,\cdots,n$, $\beta_k=0.3\cdot$

Table 1: Parameters setting

Parameter	Value
No. of space partition	12
sub-population size	50
p_c	1
p_m	0.01
γ_k	0.005
T	10
S_{size}	50

4.2 Influence of Number of Space Partition on Pareto Front

When the whole population size is 600 and the other parameters are same, Pareto fronts in 3 kinds of different space partition numbers are shown as Fig. 1. Table 2 are C-measures in those cases, where x_p denotes the Pareto optimal solution set when a decision space is partitioned into p subspaces and x_q denotes the Pareto optimal solution set when a decision space is partitioned into q subspaces. It is shown from Table 2 that the more the number of space partition numbers, the better the approach property of the Pareto optimal solution set is.

Table 2:Influence of number of space partition on C-measure

p	q	$C(x_p, x_q)$	$C(x_q, x_p)$
12	1	0.9565	0
6	1	0.9565	0.0769
12	6	0.4872	0

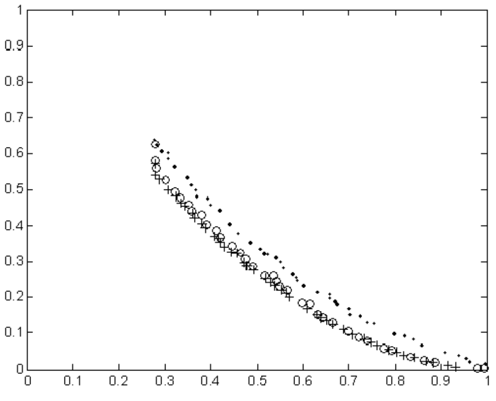
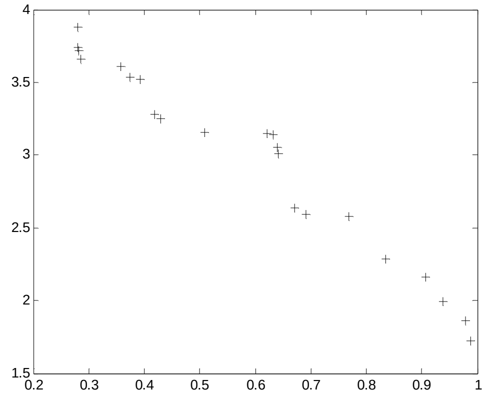


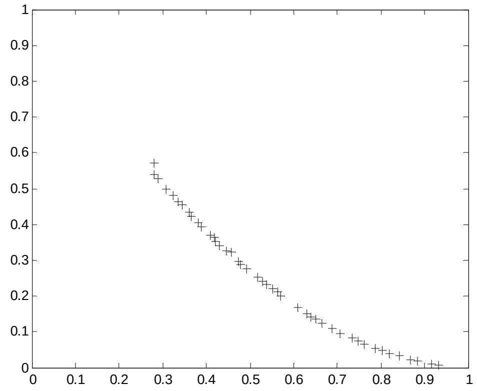
Fig.1. Influence of number of space partition on Pareto front, where •, o and + denote no search space partition, search space is partitioned into 6 and 12 subspaces respectively.

4.3 Influence of Intra-population Evolutions on Pareto Front

For crossover operator of the algorithm in this paper takes information of the Pareto optimal solutions into consideration so as to guarantee the direction of crossover, and hence the offspring resulting from crossover approach to the true Pareto optimal solution set is good. It is shown from Fig. 2 that the Pareto front obtained by not applying strategy in subsection 2.3 is not good and the Pareto front obtained by applying such strategy is better evidently.



(a) Pareto front obtained by not applying strategy in subsection 2.3.



(b) Pareto front obtained by applying strategy in subsection 2.3.

Fig. 2. Influence of intra-population evolutions on Pareto front.

4.4 Influence of Generating and Updating Super Individual Archive on Pareto Front

The influence of updating the super individual archive on the Pareto front is shown as Fig. 3. The influence of updating the super individual archive on C-measures and SP-measures at different generations are listed in Table 3, where SP1 is the SP-measure updating the super individual archive, SP2 is the SP-measure not updating the super individual archive, *ps* denotes the Pareto optimal solution set generated by updating the super individual archive, and *p* denotes the Pareto optimal solution set generated by not updating the super individual archive. It is shown from Table 3 that updating the super individual archive improves not only the approach property but also the distribution of the Pareto front.

Table 3: Influence of updating super individual archive on C-measure and SP-measure

Generation	SP1	SP2	C(ps,p)	C(p,ps)
200	0.0004	0.0044	0.7021	0.1087
300	0.0006	0.0063	0.9400	0.0400
400	0.0047	0.0172	0.5800	0.0400

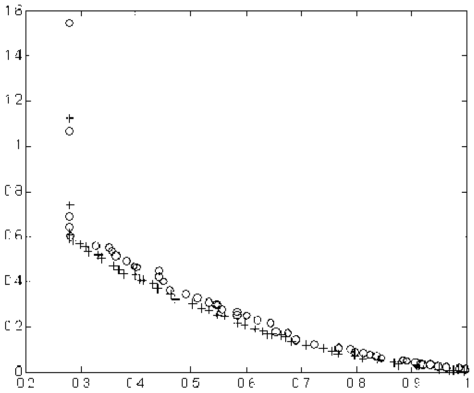


Fig. 3. Influence of updating super individual archive on Pareto front, where + and o denote Pareto front generated by updating super individual archive and by not updating super individual archive respectively.

5. Conclusion

Multi-population genetic algorithms with space partition are designed in allusion to the characteristics of multi-objective optimization problems. By using the algorithm proposed in this paper, a complicated problem can be converted into several simple problems. The efficiency of crossover operator for an intra-population evolution proposed in this paper improves. The frequency of updating the super individual archive decreases. Besides, the search scope of a population is expanded. It is shown from analysis that the computational complexity of the algorithm proposed in this paper decreases evidently. The efficiency of the algorithm proposed in this paper is validated through a complicated benchmark multi-objective optimization problem. The problem to be further studied is to apply adaptive genetic algorithms to intra-population evolutions so as to improve the efficiency of the algorithm.

Acknowledgment

The authors would like to acknowledge National Natural Science Foundation of China for its financial support under Grant No. 60304016.

References

[1] S. Joachim, Parallel Genetic Algorithms: Theory and Applications, ISO Press, 1993.
[2] Z. Ye and Y. Shan, “A new transmission network expansion planning based on multiple-population algorithm,” Automation of Electric Power Systems, vol. 24, pp. 24-27, 2000.(in Chinese)
[3] Y. Lin, J. Hao and Z. Ji, “A study of genetic algorithm based on isolation niche technique,” Journal of Systems Engineering, vol.15, pp. 86-91, 2000. (in Chinese)
[4] Z. Li and T. Tong, “Research on ANN evolutionary design method based on populations evolution niche genetic algorithm,” Control and Decision, vol.18, pp. 607-610, 2003. (in Chinese)
[5] M. Li, X. Yang and L. Zhou, “Genetic algorithm with dynamic regional Multi-species,” Acta Automatica Sinica, vol.29, pp. 214-218, 2003.
[6] G. G. Yen and H. Lu, “Dynamic multi-objective evolutionary algorithm: adaptive cell-based rank and density estimation,” IEEE Trans. on Evolutionary Computation, vol.7, pp. 253-274, 2003.
[7] Z. Michalewicz, Genetic algorithms + data structure= evolutionary programs. Springer-Verlag, 1994.
[8] J. Knowles and D. Corne, “Properties of an adaptive archiving algorithm for storing nondominated vectors,” IEEE Trans. on Evolutionary Computation, vol.7, pp. 100-116, 2003.
[9] K. Deb, S. Agarwal, A. Pratap and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA II,” in Proc. of Parallel Problem Solving from Nature, vol.VI, 2000, pp. 849-858.
[10] K. Deb, “Multi-objective genetic algorithms: problem difficulties and construction of test problems,” Evolutionary Computation, vol.7, pp. 205-230, 1999.
[11] E. Zitller, K. Deb and L. Thiele, “Comparison of multi-objective evolutionary algorithms: empirical results,” Evolutionary Computation, vol.8, pp. 173-195, 2000.

Dun-wei Gong received the B.S. in applied mathematics from China University of Mining and Technology, Xuzhou, P.R.China, in 1992, M.S. in control theory and its applications from Beijing University of Aeronautics and Astronautics, Beijing, P.R.China, in 1995, and Ph.D. in control theory and control engineering from China University of Mining and Technology, Xuzhou, P.R.China, in 1999.From 1995 to 2002, he has been a research associate in School of Information and Electrical Engineering, China University of Mining and Technology. Since 2003, he takes up the director of the Center for Intelligent System and Intelligent Control, China University of Mining and Technology. His main research interests are evolutionary computation and intelligent control.

Yong Zhou received the B.S. in industrial automatization from Hehai University, Nanjing, P.R.China, in 1997, M.S. in control theory and control engineering from China University of Mining and Technology, Xuzhou, P.R.China, in 2003, and now he is a PH.D. student in control theory and control engineering of China University of Mining and Technology.