

The Improved NSGA - II Based on Reverse Learning Mechanism

Qiong Yuan^{1,2, a*}, Guangming Dai^{1, b}

¹School of Computer Science, China University of Geosciences, Wuhan, 430074, China

²Computer and Science Institute, Wuhan Donghu University, Wuhan, 430212, China

^aemail: yqszb@126.com, ^bemail: gmdai@cug.edu.cn

Keywords: Multi-objective optimization; The NSGA-II algorithm; Reverse learning mechanism

Abstract. In this paper, According to the shortages of the NSGA - II algorithm in terms of the simulated binary crossover (SBX) operator, the speed of convergence and the diversity performance, The reverse learning mechanism (RLM) is applied to the initialization and evolutionary process of the NSGA-II, And introducing an improved arithmetic crossover operator. Through the series of ZDT test functions in two aspects of convergence and diversity evaluation, it show that the improved NSGA - II algorithm on the convergence speed, convergence and diversity is better than the NSGA - II algorithm.

Introduction

In 2002, Deb et al., in view of the disadvantages of NSGA, and put forward the improved algorithm NSGA-II. NSGA-II is a genetic algorithm based on Pareto optimal concept. Compared with NSGA, NSGA-II made the following improvements[1]:

(1) Using the method of rapid non dominated sorting, it reduced the computational complexity from $O(kN^3)$ down to $O(kN^2)$, where k is the number of target, and N is the number of individuals in the population.

(2) Adopting the crowding distance to maintain the diversity of population.

(3) The introduction of the elite strategy to prevent the loss of better solution.

Experimental results proved that the NSGA-II is superior to the other several representative algorithms. But the SBX (Simulated Binary Crossover) crossover operator has relatively weaker performance. To some extent, it limit the search performance of the algorithm. In addition, the NSGA-II in terms of the speed of convergence and the diversity of population remains to be improved.

Reverse Learning Mechanism

Reverse learning model was first proposed by Hamid R. Tizhoosh [2][3][4]. It has been proved to be an effective heuristic learning method, and successfully applied to the differential evolution algorithm[2][3][4], particle swarm optimization algorithm[5][6] to solve the unconstrained optimization problems, and some noise optimization problems.

The basic idea of reverse learning is: For a given problem, we should assess the current solution X , also want to evaluate the current solution of the reverse solution X^* . In the hope to find a better solution that reduces the distance between it and the optimal solution.

When X is a multi-dimensional vector, it can calculate the inverse solution in the following way: Assume that $X = (x_1, x_2, \dots, x_D)$ is a solution vector in a D dimension space, where $x_i \in [L_i, U_i], \forall i \in \{1, 2, \dots, D\}$, U_i and L_i respectively represent lower and upper bounds of the i th dimension variable in the solution space. Then the reverse solution $X^* = (x_1^*, x_2^*, \dots, x_D^*)$ of X is defined as: $x_i^* = L_i + U_i - x_i$.

Lei Peng et al. applied reverse learning mechanism in population initialization process and got the better result[7]. Because it considered not only the population P , but also the reverse population P^* . So, compared to single random initialization, there are more probability to close to the optimal target problem. Reverse learning initialization process is as follows:

(1) Randomly initializing the individuals of population P :

$$x_{ij}(0) = \text{rnd}(0,1)(U_j - L_j) + L_j$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, D$$

Where N is the size of population, D is the dimensionality of the solution space, $x_{ij}(0)$ is the j th dimensional variable value of the i th individual in the initial population P . U_j and L_j respectively represent upper and lower bounds of the i th dimension variable in the solution space.

(2) Calculating the individuals in the reverse population P^* . $x_{ij}^*(0)$ is the reverse variable value of $x_{ij}(0)$, that is the j th dimensional variable value of the i th individual in population P^* . $x_{ij}^*(0)$ is calculated as follows: $x_{ij}^*(0) = L_j + U_j - x_{ij}(0)$.

(3) Choosing N best individuals as the initial population from population P and reverse population P^* .

Reverse Learning-based Evolution

In this paper, reverse learning mechanism was applied to the evolutionary process. In the process of each generation evolution, to population P , calculating the reverse population P^* , and choosing N best individuals from population P and reverse population P^* as the next generation evolutionary population. However, taking into account the later stage of the algorithm, population P has formed certain regularity, closing to the optimal solution area. Therefore, in the later stage of the algorithm, solving reverse population means no sense and slows down the speed of the algorithm. So, this paper designed the following method: In the process of each generation evolution, to population P , using a certain probability r^* to calculate the reverse population P^* . And in the process of evolution, r^* is linear decreasing. That is:

$$r^* = \max_r^* - \frac{g}{G}(\max_r^* - \min_r^*)$$

Where G is the maximum generation number, and g is the current generation number. \max_r^* and \min_r^* respectively are the maximum and the minimum of r^* . So, the application of reverse learning mechanism can accelerate the convergence of the algorithm. At the same time, in order to maintain the diversity of population, in the process of evolution, when the population individual $x_{ij}(g)$ and the reverse population individual $x_{ij}^*(g)$ are mutual non domination, using a certain probability r to accept $x_{ij}^*(g)$. In the same way, probability r is also linear decreasing, that is:

$$r = \max_r - \frac{g}{G}(\max_r - \min_r)$$

Where \max_r and \min_r respectively are the maximum and the minimum of r .

The crossover operator is one of the most important genetic operation. In the process of crossover, the excellent individual's genes pattern is rapidly reproduced and spreaded in a population. It could help other individuals of the population to go forward in the direction of the optimal solution. Compared to the simulated binary crossover operator, the arithmetic crossover operator has better global searching ability, and can maintain the diversity of population better.

The arithmetic crossover operation is as follows: Set X_A^t and X_B^t respectively as the real number encoding of decision variables for the two individuals to crossover in the t th generation. After the crossover, the corresponding decision variable values of two individuals are as follows:

$$X_A^{t+1} = \alpha X_A^t + (1 - \alpha) X_B^t$$

$$X_B^{t+1} = \alpha X_B^t + (1 - \alpha) X_A^t$$

Where α is the parameter. When α is a constant, it is called uniform arithmetic crossover at this time. Otherwise, it is called non uniform arithmetic crossover[8].

Considering that in the process of evolution, it is hoped that, the individual of the population which has better rank(*rank* has low value)and better distribution (*dist* has larger value), occupy a larger proportion of gene in the offspring. So, the paper design the following crossover operator coefficient:

$$\alpha = \begin{cases} \frac{B.rank}{A.rank + B.rank} & A.rank \neq B.rank \\ \frac{A.dist}{A.dist + B.dist} & A.rank = B.rank \end{cases}$$

Where, *A.rank* represents non dominated sorting rank of the current generation of the individual *A*. *A.dist* represents crowding distance of the current generation of the individual *A*. So, in the prophase of the algorithm, better individual genes get better retention. So the algorithm has faster convergence speed. Meanwhile, In the later stages of the algorithm, better distribution individual genes get better retention. So improving the diversity of the algorithm.

Based on the above analysis, the improved NSGA-II algorithm are as follows:

```

/*Opposition-based population initialization*/
Random initialization of population P;
Calculate opposite population P*;
Selecting N fittest individuals from P and P* as initial population;
/*End of Opposition-based population initialization*/
Evaluate initial population;
While(the halting criterion is not satisfied)
{
    Tournamenet Selections routines;
    Arithmetic Crossover routines;
    Polynomial Mutation routines;
    Evaluate population P;
    If(rnd(0,1) < r*)
    {
        Compute opposite population P*;
        Select PopSize fittest Individuals from P and P*;
    }
}

```

Numerical experiments

In order to verify the improved NSGA-II algorithm performance, this article selects the classic series of ZDT (ZDT1,ZDT2,ZDT3,ZDT6) function as test functions.

Table 1 The dimensions in decision space and in object space on the test function

Test function	The dimensions in decision space	The dimensions in object space
ZDT1, ZDT2, ZDT3	30	2
ZDT6	10	2

The evaluation standard of the test results uses the convergence and the diversity of the results.

The experiment parameters are set as follows:

The population size 100, Maximum generation number 200, Crossover probability 0.9, Mutation probability 0.1.

The two upper and lower bounds of the probability on the improved NSGA- II:

$\max_r^* = 0.8, \min_r^* = 0.5; \max_r = 1.0, \min_r = 0.0;$

The experimental results are the mean of 10 times.

Table 2 convergence and diversity of the two algorithms on ZDT1

Algorithm	ZDT1	
	Convergence	Diversity
NSGA-II	0.00277211 +- 0.00140097	0.372788 +- 0.108156
My NSGA-II	0.00111181 +- 0.000172208	0.370298 +- 0.0815029

Table 3 convergence and diversity of the two algorithms on ZDT2

Algorithm	ZDT2	
	Convergence	Diversity
NSGA-II	0.00297525 +- 0.00223656	0.386197 +- 0.0671831
My NSGA-II	0.000852262 +- 0.000107841	0.417186 +- 0.10672

Table 4 convergence and diversity of the two algorithms on ZDT3

Algorithm	ZDT3	
	Convergence	Diversity
NSGA-II	0.00184749 +- 0.000186385	0.685294 +- 0.0449552
My NSGA-II	0.00107189 +- 3.01321e-05	0.58475 +- 0.0886773

Table 5 convergence and diversity of the two algorithms on ZDT6

Algorithm	ZDT6	
	Convergence	Diversity
NSGA-II	0.0192366 +- 0.00406833	0.374652 +- 0.0639926
My NSGA-II	0.00486568 +- 0.00232125	0.317342 +- 0.0793183

In order to facilitate a better comparison, Figure 1 to figure 4 are comparison charts of NSGA-II and improved NSGA-II program running after 150 generations of the results with the standard front.

From table 2 to table 5, the improved NSGA - II algorithm on the convergence performance is excellent. It's convergence on four test function is better than the NSGA-II algorithm. Meanwhile, The improved NSGA - II algorithm also has large improvement in terms of diversity. In addition to the diversity in the ZDT2 function lower than the NSGA - II, Other test functions are superior to the diversity of the NSGA - II. From chart 1 to chart 4, the convergence rate of the improved NSGA - II is better. So, it proved the validity of the improved algorithm.

Conclusion

Aiming at the deficiency of the NSGA-II algorithm, in this paper, it introduced reverse learning mechanism to the initialization and evolutionary process of the algorithm, And introduced the improved arithmetic crossover operator instead of the original simulated binary crossover operator. Finally, Getting an improved the NSGA-II algorithm. The experimental results show that: The improved NSGA-II algorithm has a different degree of improvement on the convergence speed, convergence and the diversity.

Acknowledgement

In this paper, the research was sponsored by the National Natural Science Foundation (Project No. 61472375) and Youth Fund Project of Wuhan Donghu University (Project No. 2014) .

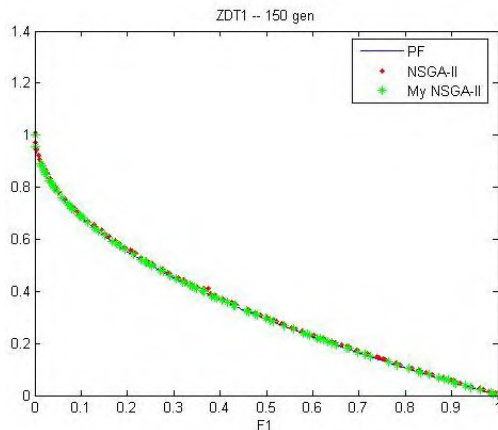


Chart 1 two algorithms on ZDT1 with frontier

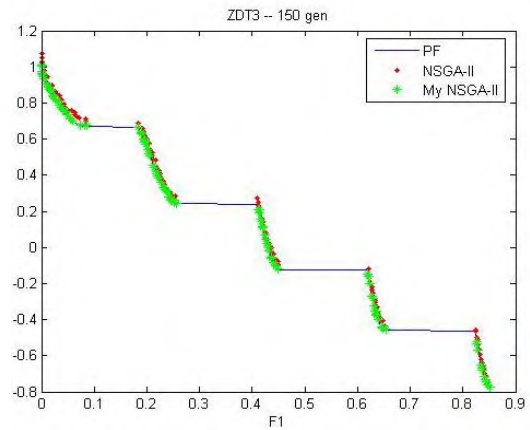


Chart 3 two algorithms on ZDT3 with frontier

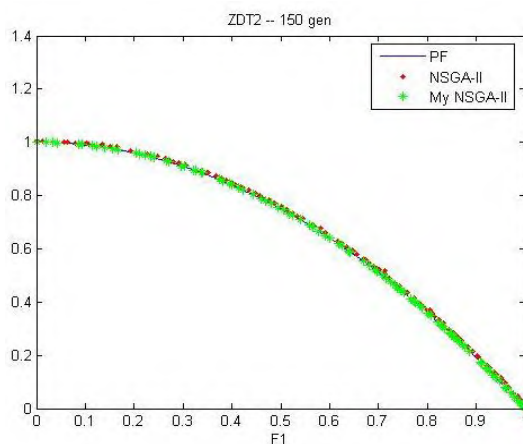


Chart 2 two algorithms on ZDT2 with frontier

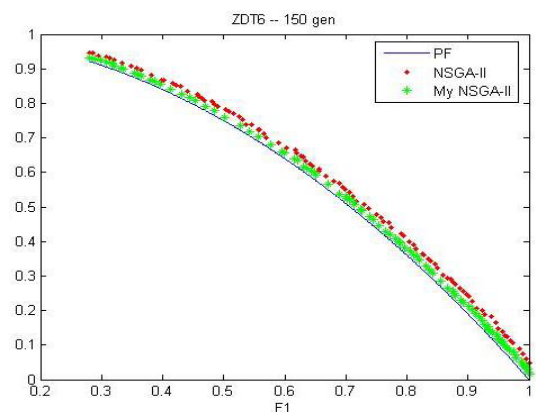


Chart 4 two algorithms on ZDT4 with frontier

References

- [1] Q.Zheng. Elitist Nondominated Sorting Genetic Algorithm and Its Application[D].hangzhou : School of information science and engineering, zhejiang university, 2006.
- [2]S.Rahnamayan,H.R.Tizhoosh,M.M.A.Salama.Opposition-based Differential Evolutionary Algorithms[C]. In Proceeding of the IEEE Congress on Evolutionary Computation,2006, 2010-2017.
- [3]S.Rahnamayan,H.R.Tizhoosh,M.M.A.Salama. Opposition-based Differential Evolution for Optimization of Noisy Problems[C]. In Proceeding of the IEEE Congresson Evolutionary Computation, 2007, 1865-1872.
- [4]S. Rahnamayan, H. R.Tizhoosh,M. M. A. Salama. Opposition-Based Differential Evolution[J], IEEE Trans.Evolutionary Computation, VOL. 12, NO. 1, 2008, 64-79.
- [5] H.Wang, Y. Liu, S. Zeng, H. Li,C. Li. Opposition-based Particle Swarm Algorithm with Cauchy Mutation[C]. In Proceeding of the IEEE Congresson Evolutionary Computation, 2007, 4750-4756.
- [6]H.Wang. Research on Hybrid Particle Swarm Algorithms with Cauchy Mutation[D].Wuhan: Computer college, China university of geosciences, 2008.
- [7]Lei Peng, Yuanzhen Wang, Guangming Dai. A Novel Opposition-Based Multi-objective Differential Evolution Algorithm for Multi-objective Optimization[C]. ADVANCES IN COMPUTATION AND INTELLIGENCE Lecture Notes in Computer Science, 2008, Volume 5370/2008, 162-170.
- [8]X.H.Liu,Y.S.Liu,G.Y.Zhang etc..Improvement of Multi-objective Optimization Algorithm NSGA-II[J].Computer engineering and application : 2005,15,73.