Solving Multi Objective Optimization Problems Using Particle Swarm Optimization

L.B. Zhang[1], C.G. Zhou[1], X.H. Liu[1], Z.Q. Ma[1,2], M. Ma[1] and Y.C. Liang[1,3*]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, P. R. China

[2] Department of Computer Science, Northeast Normal University, Changchun 130024, P. R. China

[3] Institute of High Performance Computing, Singapore 117528, Singapore

Abstract: An algorithm for solving multi objective optimization problems is presented based on PSO through the improvement of the selection manner for global and individual extremum. The search for the Pareto Optimal Set of multi objective optimization problems is performed. Numerical simulations show the effectiveness of the proposed algorithm.

Keywords: Particle Swarm Optimization, multi objective optimization, Pareto Optimal

1 Introduction

There are many multi objective optimization problems in production practice, engineering design, social production and economic development, such as production process control, design of complex software and hardware systems, analysis of social and economic benefits and so on. Therefore, it is of great importance to study the multi objective optimization problem.

The multi objective optimization problem can be expressed as follows:

$$\text{Min } y = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$$
$$s.t.\ g_i(x) \leq 0$$

where $x \subset R^m$ is the decision vector, $y \subset R^n$ the objective vector, $f_i(x)$ $(i = 1, 2, \ldots, n)$ the objective functions, and $g_j(x) \leq 0$ the system constraints.

In most cases, the objective functions may conflict with each other. This may cause some multi objective optimization problems not to have the unique best global solution. The solution can make all objective functions be the optimum at the same time. However, there exists a solution that can not be further optimized for one or several objective functions and cannot be further worsened for other objective functions. This solution is called Pareto Optimal.

Definition 1: Let x^1 be a point in the search space, it is a Pareto Optimal iff there doesn't exist *i* (in the search space) which makes $f_i(x) < f_i(x^1)$ hold.

Definition 2: The set composed of all the Pareto Optimal is called Pareto Optimal Set, and is also called Acceptable Set or Effective Set.

The objective vectors corresponding Pareto Optimal are called non-dominator objective vectors. All the non-dominator objective vectors make up Pareto Front of a multi objective problem.

Because there is no unique global best solution in the multi objective problem, to find a solution for the multi objective problem is to find a set of solutions (Pareto Optimal Set) [1]. Traditional multi objective optimization is settled by turning the multi objective problem into single objective problem through weighted sum. However, this method requires a priori knowledge of the problem itself, so it can not solve real multi objective problems. Evolutionary Algorithm is a computer technique based on population, which can search for several solutions in the solution space and can improve the efficiency of working out solutions through the similarity of different solutions. Therefore, Evolutionary Algorithms are very suitable for solving multi objective optimization problems. Schaffer studied multi objective optimization problems using vector evaluated genetic algorithms in 1980's [2]. In recent years many evolutionary algorithms used to solve multi objective optimization problems have been proposed and successfully applied to multi objective optimization problems [3].

Particle Swarm Optimization (PSO) is an optimization algorithm proposed by Kennedy and Eberhart in 1995 [4, 5]. It is easy to be understood and realized and has been applied in many optimization problems [6-9]. The PSO is more effective than traditional algorithms in most cases. The application of PSO in the multi objective optimization problems could

[1] To whom correspondence should be addressed. E-mail: liangyc@ihpc.a-star.edu.sg

be very promising. This paper proposes a PSO method applied to solving multi objective optimization problems. Satisfactory results through numerical experiments are obtained.

2 Particle Swarm Optimization

PSO originated from the research of food hunting behaviors of birds. Researchers found that in the course of flight flocks of birds would always suddenly change direction, scatter and gather. Their behaviors are unpredictable but always consistent as a whole, with individuals keeping the most suitable distance. Through the research of the behaviors of similar biological communities, it is found that there exists a social information sharing mechanism in biological communities. This mechanism provides an advantage for the evolution of biological communities, and provides the basis for the formation of PSO.

Every swarm of PSO is a solution in the solution space. It adjusts its flight according to its own and its companion's flying experience. The best position in the course of flight of each swarm is the best solution that is found by the swarm. The best position of the whole flock is the best solution, which is found by the flock. The former is called *pBest*, and the latter is called *gBest*. Every swarm continuously updates itself through the above mentioned best solution. Thus a new generation of community comes into being. In the practical operation, the fitness function, which is determined by the optimization problem, assesses the extent to which the swarm is *good* or *bad.*

Obviously, each swarm of PSO can be considered as a point in the solution space. If the scale of swarm is *N*, then the position of the *i*-th $(i = 1, 2, \ldots, N)$ particle is expressed as X_i. The "best" position passed by the particle is expressed as $pBest[i]$. The speed is expressed with V_i. The index of the position of the "best" particle of the swarm is expressed with *g*. Therefore, swarm *i* will update its own speed and position according to the following equations [4, 5, 9]:

$$
\begin{aligned}
V_i &= w \times V_i + c_1 \times rand() \times (pBest[i] - X_i) \\
&\quad + c_2 \times Rand() \times (pBest[g] - X_i) \qquad (1) \\
X_i &= X_i + V_i
\end{aligned}
$$

where c_1 and c_2 are two positive constants, *rand()* and *Rand()* are two random numbers within the range [0,1], and w is the *inertia weight*. The equations consist of three parts. The first part is the former speed of the swarm, which shows the present state of the swarm; the second part is the cognition modal, which expresses the thought of the swarm itself; the third part is the social modal. The three parts together determine the space searching ability. The first part has the ability to balance the whole and search a local part. The second part causes the swarm to have a strong ability to search the whole and avoid local minimum. The third part reflects the information sharing among the swarms. Under the influence of the three parts, the swarm can reach an effective and best position.

In addition, the swarm is limited by $V_{\max}$ when it is adjusting its own position according to the speed. The speed V_i is set to be $V_{\max}$ when V_i exceeds $V_{\max}$.

3 Multi Objective PSO

3.1 Introduction of the algorithms

The successful application of PSO in many single objective optimization problems reflects the effectiveness of PSO. However, PSO can not be immediately applied to multi objective optimization problems, because there are essential distinctions between multiple and single objective optimization problems. First, the former is the set of one group or several groups of solutions, while the latter is only single solution or a group of series of solutions. In addition, the successful application of genetic algorithms in multi objective optimization problems and the similarity between PSO and genetic algorithms reflects that PSO is likely a method to deal with multi objective optimization problems. However, there is a great distinction between PSO and genetic algorithms. In genetic algorithms, chromosomes share the information, which causes the whole community moves gradually into a better area, while in PSO the information is sent out by the best particle which is followed by other individuals to quickly converge to a point. Therefore, it may easily cause the swarms to converge to the local area of Pareto Front if the PSO is applied directly to multi objective optimization problems.

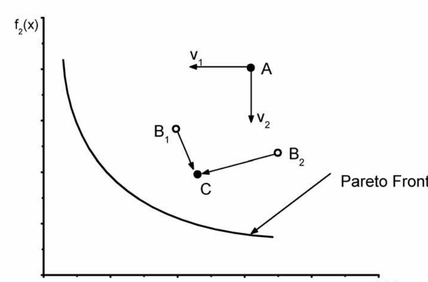

Fig. 1. Objective function space

In view of the above-mentioned reasons, this paper proposes the PSO assessed and chosen by the best solution and applied it to the search of Pareto Optimal Set in multi objective optimization problems. First, the algorithm initializes a particle swarm in the dominant vectors space. Then, the PSO directs the flight of swarm in the dominant vectors space together with each objective function in multi objective optimization problems, which causes the swarm to fall into the Pareto Optimal Set. Reflected in the space of objective function, the swarm will fall into Pareto Front. As illustrated in Figure 1, it is the situation of the space of objective function when minimize $f_1(x)$ and $f_2(x)$. If there is only one objective function $f_1(x)$ or $f_2(x)$, the objective vector A would change in the direction of v_1 or v_2. However, in the algorithm both objective functions $f_1(x)$ and $f_2(x)$ are in the dominant vector space and direct the change of A. Therefore, the possible change of the vector A is neither in the direction of v_1 nor in the direction of v_2. The vector A changes in a certain direction between v_1 and v_2 when $f_1(x)$ and $f_2(x)$ don't increase at the same time, and finally reaches the Pareto Front. The algorithm is performed as follows: First, find out the global best solution gBest[i] $(i = 1, 2, \ldots N)$ and the best individual solution pBest[i,j] $(j = 1, 2, \ldots N)$ in each swarm using each objective function in the multi objective optimization problems. The variables corresponding to each gBest[i] in the dominant vector space make up an area called *quasi-solution area*. When the speed of each swarm is updated, the "average" of each gBest[i] is used as the best global solution gBest. Each particle's pBest[i,j] is determined through judging the dispersed degree of vectors pBest[i,j] and gBest[i] to choose the "average" of the pBest[i,j] or choose randomly in the pBest[i,j]. In addition, when the position of each particle is updated, it should be decided whether the position of each particle is within the *quasi-solution area*. If it is then remain the original value, otherwise update the current value.

3.2 Analysis of the algorithm

In the PSO the behavior of each swarm is decided mainly by *gBest* and *pBest***,** so the method proposed in this paper makes the behavior quite different when each swarm moves to the solution. That is to say, each swarm moves to different solutions in the area of solutions. Thus, the whole group would be scattered finally into the Pareto Optimal Set, which avoids the whole group scattering into the local area of the Pareto Optimal Set. It is important that the assessment and selection of *gBest* reflects the restrictions among each objective function while avoiding all individuals falling into the best position of a certain objective function. The selection method of *pBest* especially strengthens this conduct. The renewal method of the selection of the swarms not only assures the swarms gather at the best solution as possible as they could, but also causes each swarm to have enough search ability with freedom, thus avoiding local area.

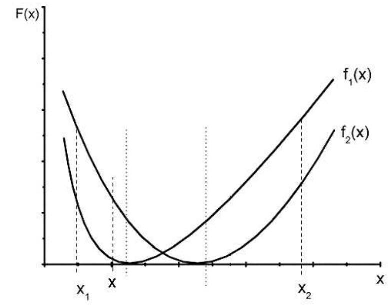

Fig. 2. Dominant vector space

Figure 2 shows the situation of the dominant vector space after a certain cycle when minimizing $f_1(x)$ and $f_2(x)$. X_1 and X_2 are the best solutions (gBest[1] and gBest[2]) to the objective functions $f_1(x)$ and $f_2(x)$, respectively. Corresponding to X_1 and X_2 the objective vectors in the space of objective functions are B_1 and B_2 (Refer to Figure 1). According to the algorithm gBest is obtained through the assessment and the selection of gBest[1] and gBest[2]. Its corresponding solution must be X between X_1 and X_2 (Refer to Figure 2). C is corresponding to the objective vector of X in the space of objective functions. Obviously, C is much nearer to the Pareto Front than B_1 and B_2. This also illustrates that using X(gBest) is better than using gBest[1] or gBest[2]. The best solution gBest can be obtained by using the following equation:

$$gBest = \frac{n-i}{n}(gBest[1]) + \frac{i}{n}(gBest[2]) \qquad (2)$$

where i is the i-th iteration and n is the total iteration times. Eq. (2) causes the value of gBest to change gradually into gBest[2] from gBest[1] during the iteration. In each iteration, Eq. (2) causes gBest[2] and gBest[1] to have influence on gBest with different extent. Therefore, Eq. (2) not only reflects the mutual restriction of the two objective functions, but also results in the solution approaching the Pareto Front.

Compared with the standard PSO, this paper has made some improvement on the best global solution and the best individual solution of particles.

3.3 The execution of the algorithm

The execution of the proposed algorithm is introduced using a two-objective optimization problem. Let's consider the minimization of $f_1(x)$ and $f_2(x)$.

(1) Initialize the particle swarm: Designate the

population size N, generate speed V_i and position X_i of each particle randomly.

(2) Evaluate the fitness of each particle: Obtain Fitness1[i] and Fitness2[i] by using the two objective functions $f_1(x)$ and $f_2(x)$.

(3) Calculate the best individual solutions pBest1[i] and pBest2[i].

(4) Calculate the best global solutions gBest[1] and gBest[2].

(5) Calculate the "average" of the two best global solutions gBest and their distances dgBest'

(a) Evaluate gBest from gBest[1] and gBest[2].

(b) Evaluate the distance dgBest between gBest[1] and gBest[2].

(6) Calculate the distance dpBest[i] between pBest[1,i] and pBest[2,i]

(7) Calculate the best individual solution pBest[i], which is used to update the speed v[i] and position x[i] of each particle:

If (dpBest [i]< dgBest)

Choose pBest[i] randomly between pBest[1,i] and pBest[2,i],

Else

Evaluate pBest[i] using Eq. (2).

(8) Update the speed v[i] of each particle using gBest and pBest[i].

(9) Judge whether the position x[i] is in the *quasi-solution area*. If it is then remain the original value, otherwise perform the update.

(10) If the termination condition is achieved then stop, otherwise go to step (2).

4 Numerical Simulations

Four test functions are used here to perform the numerical experiments. Test function 1 was proposed by Schaffer in reference [2], which is used to test the effectiveness of algorithms in most multi objective optimization problems. Test functions 2- 4 were used in references [10] and [11]. The parameters of the PSO are that: learning rate $c_1 = c_2$ =0.5, inertia weight is taken from 0.8 to 0.4 with a linear decreasing rate. The maximum velocity $V_{\max}$ is taken as the dynamic range of the particle in each iteration.

The test functions and the simulation results are listed as follows.

(1) Test function 1

$$Min \quad f_1 = x^2$$

$$Min \quad f_2 = (x-2)^2$$

$$x \in [-5, 7]$$

The Pareto curve is shown in Fig. 3. In the simulation, 100 particles and 100 iterations are used and 100 non-dominated solutions are obtained.

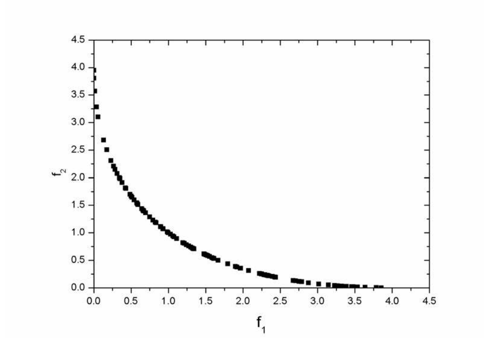

Fig. 3. Pareto curve of test function 1.

(2) Test function 2

$$Min \quad f_1(x) = \begin{cases} -x & if \ x \le 1 \\ -2+x & if \ 1 < x \le 3 \\ 4-x & if \ 3 < x \le 4 \\ -4+x & if \ x > 4 \end{cases}$$

$$Min \quad f_2(x) = (x-5)^2$$

$$x \in [-5, 10]$$

The Pareto curve is shown in Fig. 4. In the simulation, 100 particles and 100 iterations are used and 100 non-dominated solutions are obtained.

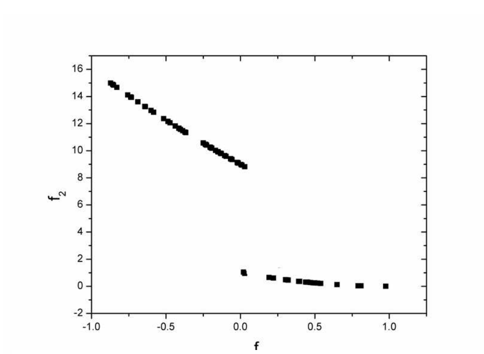

Fig. 4. Pareto curve of test function 2.

(3) Test function 3

$$Min \quad f_1(x, y) = (x^2 + y^2)^{\frac{1}{8}}$$

$$Min \quad f_2(x, y) = \left((x-0.5)^2 + (y-0.5)^2\right)^{\frac{1}{4}}$$

$$x \in [-5, 10]$$

The Pareto curve is shown in Fig. 5. In the simulation, 100 particles and 100 iterations are used and 90 non-dominated solutions are obtained.

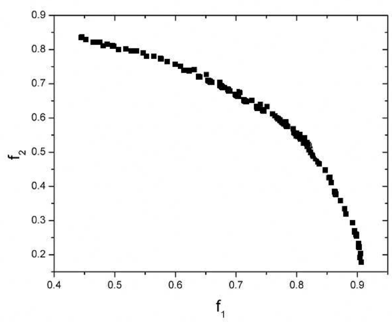

Fig. 5. Pareto curve of test function 3.

(4) Test function 4

$$Min \quad f_1 = x_1$$

$$Min \quad f_2 = 1 - \left(\frac{f_1}{g}\right)^2$$

$$g = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$

$$x_i \in [0,1] \quad n = 30$$

The Pareto curve is shown in Fig. 6. In the simulation, 200 particles and 200 iterations are used and 130 non-dominated solutions are obtained.

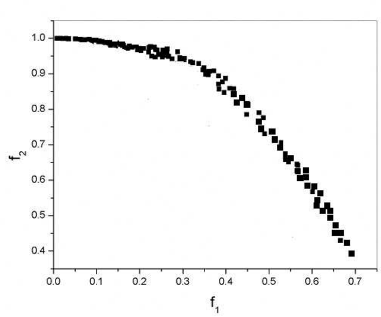

Fig. 6. Pareto curve of test function 4.

In the objective functions space, the Pareto Front is the boundary of the fitness area, that is, the availability interface. As for the minimum two-dimensional cases in the experiment, the availability interface should be the left boundary on the bottom of fitness. From Figs. 3-6 it can be seen that all the test functions provide the exact availability interface. As for the test functions 1 and 2 (Refer to Figs. 3 and 4), the complete Pareto curve is obtained and the objective vectors distribute correctly and uniformly in the availability interface. For test functions 3 and 4, especially, for function 4, the optimization problem is difficult. The Pareto curve obtained using the proposed algorithm is comparatively correct, and the objective vector distributes evenly. From Figs. 5 and 6 it can be seen that although some individuals diverge a little from the availability interface, the results provide valuable references to the practical decision-maker in multi objective optimization problems. Numerical simulations show that the proposed algorithm is very effective to deal with the multi objective optimization problems.

The comparison between the test results obtained in this paper and those from references [10] and [11] shows the effectiveness of the proposed algorithm. Lis and Eiben proposed an algorithm for solving multi objective optimization problems using multi-sexual genetic algorithm and presented test results [10]. Deb and Thiele analysed and compared some multi objective evolutionary algorithms, and presented the test results [11]. The results of reference [11] show that the SPEA (Strength Pareto Evolutionary Algorithm) is the best method among the examined algorithms. Our simulations show that the test results obtained in this paper are as good as those by using the SPEA. Note that the choice of the population size in [10] and [11] influences the effectiveness of the algorithms strongly, i.e., it needs enough population size to enable the iterations to converge towards the Pareto front in references [10] and [11]. Whereas, it is not the case in the proposed algorithm. Only a small population size is needed to obtain the desired results. In addition, the proposed algorithm can be understood and performed easily because there is no operations such as "crossover" and "mutation" used in other evolutionary algorithms solving multi-objection problems.

5 Conclusions

Compared with the standard PSO, the proposed algorithm to handle multi objective optimization problems has different selection manner for the best global solution and best individual solution. Numerical experiments indicate the effectiveness of the algorithm. The comparison with the results of references [10] and [11] shows that the proposed algorithm possesses good performance. The correct form of Pareto curve for the comparatively difficult problem can also be obtained and most individuals can fall into the Pareto Optimal Set by using the proposed algorithm.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 60175024), the Key Science-Technology Project of the National Education Ministry of China (Grant No. 02090), and the Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China.

References

[1] Coello C.A., A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, Knowledge and Information Systems, vol. 1,No.3, pp.269-308, Aug.1999.

[2] Schaffer J.D., Multiple objective optimization with vector evaluated genetic algorithms. Genetic Algorithms and their Applications: Proceeding of the First International Conference on Genetic Algorithms, Lawrence Erlbaum, pp93-100, 1985.

[3] Veldhuizen D.A.V. and Lamont G.B., Multiobjective evolutionary algorithm research: a history and analysis, TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, OH, USA, 1998.

[4] Eberhart R. and Kennedy J., A new optimizer using particle swarm theory, Proc. 6 Int. Symposium on Micro Machine and Human Science, pp. 39-43, 1995.

[5] Kennedy J. and Eberhart R., Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995.

[6] Parsopoulos K.E. and Vrahatis M.N., Particle Swarm Optimizer in Noisy and Continuously Changing Environments. Hamza M.H. (Ed.), Artificial Intelligence and Soft Computing, IASTED/ACTA Press, pp. 289-294, 2001.

[7] Parsopoulos K.E. and Vrahatis M.N., Particle swarm optimization method for constrained optimization problems, Proceedings of the Euro-International Symposium on Computational Intelligence 2002, 2002.

[8] Eberhart R.C. and Hu X., Human tremor analysis using particle swarm optimization, Proceedings of the IEEE Congress on evolutionary computation (CEC 1999), Washington D.C., pp.1927-1930, 1999.

[9] Shi Y. and Eberhart R., A Modified Particle Swarm Optimizer, IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[10] Lis J. and Eiben A.E., A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In Toshio Fukuda and Takeshi Furuhashi, editors, Proceedings of the 1996 International Conference on Evolutionary Computation, Nagoya, Japan, pp. 59-64,1996.

[11] Deb E.Z.K. and Thiele L., Comparison of multi objective evolutionary algorithms: Empirical results, Tech. Rep. 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, February 1999.