

An effective solution to finding global best guides in particle swarm for typical MOPs

Zheng Li

School of Control and Computer Engineering, North China Electric Power University, Baoding, Hebei province, China

yeziperfect@163.com

Keywords: MOPSO, Pareto archive, non-dominated solutions, crowding distance, diversity, distribution

Abstract. It is of critical importance for convergence and diversity of final solutions that finding out a feasible global best guide for each particle of the current swarm in multi-objective particle swarm optimization (MOPSO). An improved approach for determining the best local guide in MOPSO is proposed, where the Pareto archive with size limit is used to store the non-dominated solutions. While selecting the local best particle, a random number is used to judge whether the crowding distance is taken into account or not. A new solution is referred to overcome the problem that it is much harder to generate a new particle dominating the original one in MOPs than in single-objective optimal problems. In addition, to improve the efficiency of search and avoid precocity, the inertial weight changes in the iteration process. The proposed approach is applied to some typical testing functions, and the experimental results of Pareto fronts for these functions are satisfied.

Introduction

One of the major problems existing today in the area of science research and industrial practice is multi-objective optimization problem (MOP)[1, 2]. The most important characteristic of MOP, which is different with single objective problem, is the final solutions. A problem with conflicting objective functions gives rise to a set of optimal solutions, instead of one optimal solution. Among these solutions, none is better than any other with respect to all of the conflicting objectives. The concept of Pareto optimal is usually used in the solving of MOPs, whose final goal can be boiled down to a Pareto non-dominated set meeting all constraints of the problem.

Due to the reasons mentioned above, multi-objective evolutionary algorithms(MOEAs) have been widely used, such as Non-dominated Sorting Genetic Algorithm (NSGA-II)[3, 4], Artificial Bee Colony Algorithm(ABC)[5], Particle Swarm Optimization(PSO)[6] and so on. During the past decades, it has been confirmed that PSO is effective while solving optimal problem including only one objective. The particularities of MOPs make PSO not applicable directly before necessary modification. Therefore, more and more extended PSO has been proposed. In [7], fuzzy mechanism was used to select the particle possessing global best position in the proposed optimal algorithm. New data structures called dominated and non-dominated trees were introduced to facilitate the use of an unconstrained elite archive[8].

Mathematical Model of MOP

Without loss of generality, it is assumed all of the objectives are to be minimized, so a MOP can be of the form:

$$\min f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad (1)$$

Obviously, there are m conflicting objective functions need to be minimized simultaneously, and each of them depends on the decision vector $x = (x_1, x_2, \dots, x_n)$. Each $x_i (i = 1, 2, \dots, n)$ represent a dimension of the vector and the value must be feasible with some limits of itself or among each other. As mentioned before, for MOPs, where the objectives are typically conflictive, performance

improving of one objective may sacrifice the performance of at least one another, so the concept of Pareto dominating will be given.

Decision vector x is said to strictly dominate another y iff

$$f_i(x) \leq f_i(y) \quad \forall i=1,2,\mathbf{L},m. \quad (2)$$

$$f_i(x) < f_i(y) \quad \text{for some } i. \quad (3)$$

If there is no such decision vector x^* exist which dominate x , x is called Pareto optimal solution. The set of all Pareto optimal solutions is said to be a non-dominated set.

The Improved MOPSO

In this paper, some concepts are defined as followings.

Assume that nds represent Pareto solution space, the size limit of which is S . Firstly, initialize the population including N particles randomly, taking consideration of the variables' feasible region. Choose the first particle and input it into nds , and examine the second one according to Algorithm 1. The rest particles can be done in the same manner until the last one. As $x_i (i=1,2,\mathbf{L},m)$ is n -dimensional vector, nds can be expressed as formulation (4):

$$nds = (x_1, x_2, \dots, x_m)^T = \begin{pmatrix} x_{11} & x_{12} & \mathbf{L} & x_{1n} \\ x_{21} & x_{22} & \mathbf{L} & x_{2n} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ x_{m1} & x_{m2} & \mathbf{L} & x_{mn} \end{pmatrix}. \quad (4)$$

During the iterative process, distance matrix for nds is identified by dm , which is a diagonal matrix.

$$dm = \begin{pmatrix} d(x_1, x_1) & d(x_1, x_2) & \mathbf{L} & d(x_1, x_m) \\ d(x_2, x_1) & d(x_2, x_2) & \mathbf{L} & d(x_2, x_m) \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ d(x_m, x_1) & d(x_m, x_2) & \mathbf{L} & d(x_m, x_m) \end{pmatrix}. \quad (5)$$

where the definition of $d(x_i, x_j)$ can be described as follows:

If i is equal to j , obviously, the distance between them is zero, that is, $d(x_i, x_j)=0$; Otherwise, $d(x_i, x_j)$ would be the geometric distance in the n -dimensional space, i.e. $d(x_i, x_j) = \sqrt{\sum_{t=1}^n (x_{it} - x_{jt})^2}$. As in the MOPs, great attention will be paid to the distribution of functional value rather than independent variable, thus it would be more suitable regarding distance between functional values as the value of distance matrix.

The dense distance, d_dis , is a m -dimensional column vector, which can be expressed as follows:

$$d_dis = (d_1^{\min}, d_2^{\min}, \dots, d_m^{\min}). \quad (6)$$

where each d_i^{\min} represents the minimal distance between x_i and all other Pareto solutions, i.e., the minimal value of every row in dm .

The sparse distance, s_dis , a m -dimensional column vector, can be expressed as follows:

$$s_dis = (d_1^{\min}, d_2^{\min}, \dots, d_m^{\min}). \quad (7)$$

where each d_i^{\min} represents the average value of two minimal distances between x_i and all other Pareto solutions, i.e., the average of two minimal value of every row in dm .

Algorithm1: Judge whether or not a feasible solution is Pareto-optimal, if it is, input it into nds . Here Algorithm 2 would be invoked. For each Pareto solution in nds , invoke Algorithm 2, and determine whether or not deleting the current Pareto solution according to the value of $domiRel$. If the size limit of Pareto solution space is exceeded, delete one of the solutions whose dense distance is minimal.

Algorithm 2: Calculate the functional value of two particles, and judge the dominance relationship between them.

Algorithm 3: Finding out the local best guide for the current particle. Generate a random number between zero and 3. If it is zero, regard any one of the vectors whose sparse distances are biggest as the local best particle. If the random number falls between one and three, choose the local best randomly.

Here the key algorithms are described.

(1) Initialize the population according to the given size, where the position and speed of each particle can be set randomly, of course, the value limits must be satisfied. The non-dominated solutions nds and distance matrix dm are both initialized as null. The current iteration count t is set to 0.

(2) Each particle of the population will be taken into account, and update nds and dm according to the function outlined in Algorithm 1.

(3) The best memory of each particle, $pbest$, which serves as a guide for particle to travel through the search space, is initialized as the particle itself.

(4) $t = t + 1$.

(5) Update W if necessary.

(6) As explained in Algorithm 3, the function is used to determine the local best particle for the current iteration, which is represented as $gbest$. In this algorithm, the concept of distance matrix for Pareto solutions is of great significance.

(7) Calculate positions and velocities of particles in new generation, where the limit values, for example, the assigned value of the position interval for each component of particles must be noted.

(8) According to the objective functions, calculate the objective vectors for all particles, and store the results into a particular three-dimensional matrix, named $goals$, whose values are composed of the objective function value of a certain particle in a given iteration generation.

(9) Adjust the best memory of each particle, $pbest$, through judging the dominance relationship between the new particle and the original $pbest$ using Algorithm 2.

(10) Examine the dominance relationship between each particle in new generation and nds , and update the values of nds and dm .

(11) If the terminate criterion is satisfied, the iterative process ends; Otherwise, jump to (4).

Simulation results and analysis

Here several typical test problems with two objectives are chosen to test the proposed algorithms, including Schaffer's study(SCH) and parts of developed test problems for multi-objective optimization by Zitzler. In the case of setting the same parameters and testing many times, the Pareto front curves for all tested problems are given in Fig.1.

In the implementation of this study, the key parameters are set as follows. The inertia weight W gradually drops to 0.4 from the initial value 1.0, which is to control the impact of history velocity on the current velocity. f_{r1} and f_{r2} are both random values in the range [0,1], and the population size is set to 50 and 100, while the program is to run 200 generations. The Pareto archive is limited to 100. Simulation results are acquired with two different population sizes.

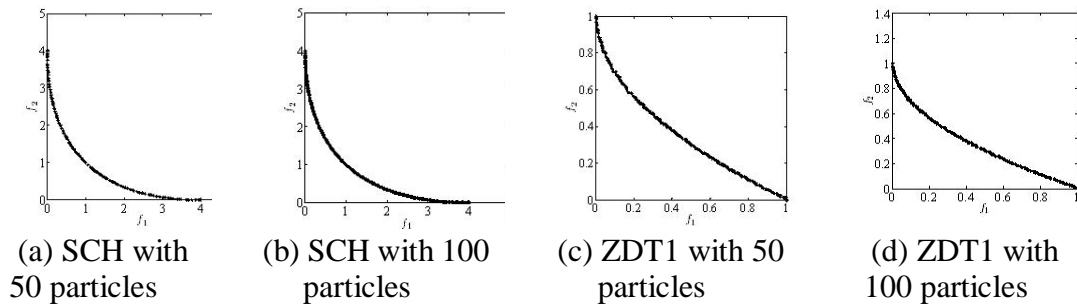


Fig.1 Pareto front for all test problems with different population sizes

Conclusions

In this study, a new improved MOPSO is proposed to implement the typical MOPs. In this algorithm, the main work is to find the best local guides for each particle in order to obtain satisfied Pareto fronts with high diversity. From the simulation results, it can be seen that the Pareto fronts of the improved approach are satisfied. In future work, it will be applied to more complicated MOPs with more constraints or objectives.

Acknowledgements

This work was financially supported by the Hebei Province Natural Science Foundation(F2014502081) and the Fundamental Research Funds for the Central Universities(2015MS128).

References

- [1] Florentino, H.O., D.R. Cantane, F.L.P. Santos, et al. Multiobjective Genetic Algorithm applied to dengue control[J]. *Mathematical Biosciences*, 2014, 258: 77-84.
- [2] Wang, X.L., W.J. Cai, J.G. Lu, et al. Optimization of Liquid Desiccant Regenerator with Multiobject Particle Swarm Optimization Algorithm[J]. *Industrial & Engineering Chemistry Research*, 2014, 53(49): 19293-19303.
- [3] Deb, K., A. Pratap, S. Agarwal, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-97.
- [4] S.Dhanalakshmi, S.Kannan, K.Mahadevan, et al. Application of modified NSGA-II algorithm to Combined Economic and Emission Dispatch problem[J]. *Electric Power and energy systems*, 2011, 33(4): 992-1002.
- [5] Zou, W., Y. Zhu, H. Chen, et al. Solving Multiobjective Optimization Problems Using Artificial Bee Colony Algorithm[J]. *Discrete Dynamics in Nature and Society*, 2011: 1-37.
- [6] Abido, M.A. Multiobjective particle swarm optimization for environmental/economic dispatch problem[J]. *Electric Power Systems research*, 2009, 79(7): 1105-1113.
- [7] Cai, J., X. Ma, and Q. Li. A multi-objective chaotic particle swarm optimization for environmental/economic dispatch[J]. *Energy Conversion and Management*, 2009, 50(5): 1318-1325.
- [8] Fieldsend, J.E., R.M. Everson, and S. Singh. Using unconstrained elite archives for multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(3): 305-323.