International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 04 - Issue 05 || May 2018 || PP. 07-12

An improved simulated annealing algorithm for multiobjective programming problem

Tao Zhang [1]

[1]*School of Information and Mathematics, Yangtze University, Jingzhou 434023, China*

Abstract: In this paper, an improved SA is presented for solving multiobjective programming problems, in which a new method for computing energy difference is proposed and the external file technology is used. And a set of approximate Pareto optimal solutions for multiobjective programming problem is obtained using the elite strategy. This interactive procedure is repeated until the accurate Pareto optimal solutions of the original problem are found. The experimental results show that the proposed algorithm is a feasible and efficient method for solving multiobjective programming problems.

Keywords: Multiobjective programming; Simulated annealing algorithm; Pareto optimal solution; External file

1. Introduction

Simulated annealing (SA) [1] is a stochastic optimization method that is based on an analogy with physical annealing, which has been found to be quite successful in a wide variety of optimization tasks. Initially, SA has been used with combinatorial optimization problems [2]. Afterwards, SA has been extended to the single and multiobjective optimization problems with continuous N-dimensional control spaces [3-5].

The SA algorithm realizes the combination of the local search and global search through the cooling schedule. The cooling schedule is critical to the performance of the algorithm. Usually, the candidate solution is created by the current solution with a random perturbed vector, the probability density function of the random perturbed vector and the accept probability of the corresponding candidate solution are in relation to the temperature. When the temperature is higher, the search range of the candidate solution is wider, and it can also be accepted easily. When temperature is lower, the candidate solutions are constrained in the local area of the current solution, the search become local exploration. In this study, in order to improve the global search ability of the SA algorithm, the random perturbation vector is constructed based on cooling schedule used in literature [6] and the global convergence can be guaranteed. On the other hand, for the multiobjective optimization problem, the method for computing the energy difference between the current solution and the candidate solution used by [5] is employed in this paper.

2. Problem formulation

Let $x \in R^n$, $f: R^n \to R^m$, $g: R^n \to R^q$. The general model of the BLMPP can be written as follows:

$$\min_{y} \quad f(x) = (f_1(x), f_2(x), \cdots, f_m(x))$$

$$s.t. \quad g(x) \leq ($$

$$h(x) = \quad (1)$$

where $f(x,y)$ are the objective function. $g(x,y)$ and $h(x,y)$ denote the equality constraints and inequality constraints. Let $S = \{(x,y) \mid g(x,y) \le 0, h(x,y) = 0\}$.

Definition 2.1. If $x \in S$, then x is a feasible solution of the problem (1).

Definition 2.2. If x^* is a feasible solution to the problem (1), and there are no $x \in S$, such that $f(x) \prec f(x^*)$, then x^* is a Pareto optimal solution to the problem (1), where "$\prec$" denotes Pareto preference.

3. The Algorithm

3.1 The improved SA algorithm

Suppose the current solution is X^k, the candidate solution is Y^k, for the $k-th$ iteration at temperature T_k, the particle is updated by the improved SA as following:

Algorithm 1:

Step1 Create candidate solution according to the current solution.

$$Y^k = X^k + \varepsilon^k \quad (2)$$

Step2 Compute energy difference between the current solution and the candidate solution.

$$\Delta E(X^k, Y^k) = \frac{1}{|\overline{F}|}\left(\left|F_{X^k}\right| - \left|F_{Y^k}\right|\right), \eta = random(0,1). \quad (3)$$

Step3 Compute transition probability.

$$p(Y^k \mid X^k, T_k) = \min\left\{1, \exp(\Delta E(X^k, Y^k))\right\}. \quad (4)$$

Step4 Choose the offspring.

$$\text{If } p(Y^k \mid X^k, T_k) \ge \eta, X^{k+1} = Y^k; \text{ Otherwise } X^{k+1} = X^k. \quad (5)$$

In step1, $\varepsilon^k = (\varepsilon_1^k, \varepsilon_2^k, \cdots, \varepsilon_n^k)$, and the component ε_i^k of the random perturbed vector ε^k is produced as: $\varepsilon_i^k = sign(U_i) T_k \left(\frac{1}{|U_i|^m} - 1 \right)$, $(i = 1, 2, \cdots, n)$, $U_1, U_2, \cdots, U_n \in random(-1,1)$ are mutual independence uniform distribution random variables, $sign(\cdot)$ is the sign function, $m\,(m \ge 1)$ is the predefined constant. In step 2, F is the approximate Pareto front, which is the set of mutually non-dominating solutions found thus far in the annealing. Denote by $\overline{F}$ the union of the F, the current

solution X^k and the proposed solution Y^k, that is $\overline{F} = F \bigcup X^k \bigcup Y^k$. Then, let F_{X^k} be the elements of F that dominate X^k and let F_{Y^k} be the elements of F that dominate Y^k.

Algorithm 2:

Step 1. Initializing. Initialize the population P_0 with N particles. The start temperature and the end temperature are noted as $T_{\max}$ and T_f respectively, let the iteration number $k = 0$ and let $T_k = T_{\max}$. Initialize the loop counter $t_l := 0$. The non-domination particles are saved in the elite set A_t

Step 2. Update the $j-th$ ($j = 1, 2, \cdots, N$) particle's position using **Algorithm 1**.

Step 3. $t_l := t_l + 1$.

Step 4. If $t_l \geq T_l$, go to step 5. Otherwise, go to step 2.

Step 5. Update the elite set A_t ..

Step 6. $k := k + 1$, computing $T_k := \dfrac{T_{\max}}{k^m}$.

Step 7. If $T_k < T$, output the elite set A_t. Otherwise, go to step 2.

4. Numerical experiment

In this section, six examples will be considered to illustrate the feasibility of the proposed algorithm for problem (1). The parameter is as following: $P_0 = 200$, , $T_{\max} = 10^6$, $T_f = 10^{-3}$, $m = 3$. All results presented in this paper have been obtained on a personal computer (CPU: AMD 2.80GHz; RAM: 3.25GB) using a C# implementation of the proposed algorithm.

Problem 1[5]

$$\min_x \quad f_1(x) = x^2$$

$$f_2(x) = (x-2)^2$$

$$-10^{-3} \leq x \leq 10^3.$$

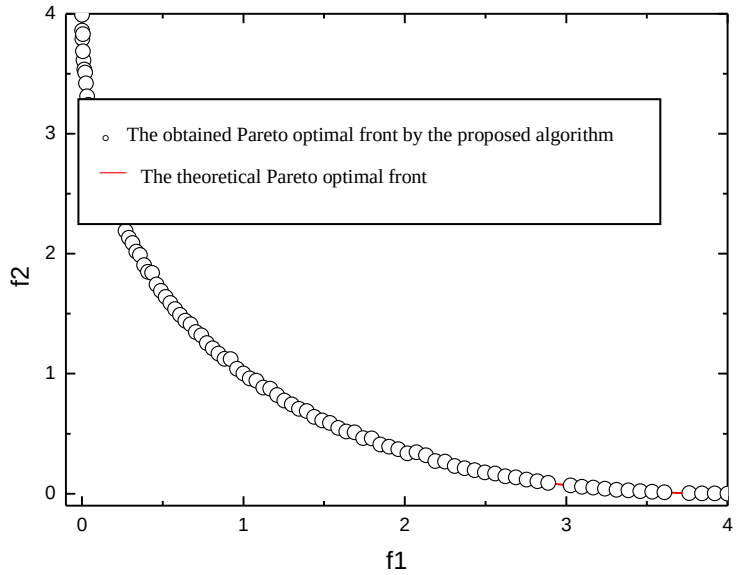

Figure 1 The obtained Pareto front of problem 1

Problem 2[5]

$$\min_x \; f_1(x) = 1 - \exp\left(\sum_{i=1}^{3}(x_i - 1/\sqrt{3})^2\right)$$

$$f_2(x) = 1 - \exp\left(\sum_{i=1}^{3}(x_i + 1/\sqrt{3})^2\right)$$

$$-4 \le x \le 4$$

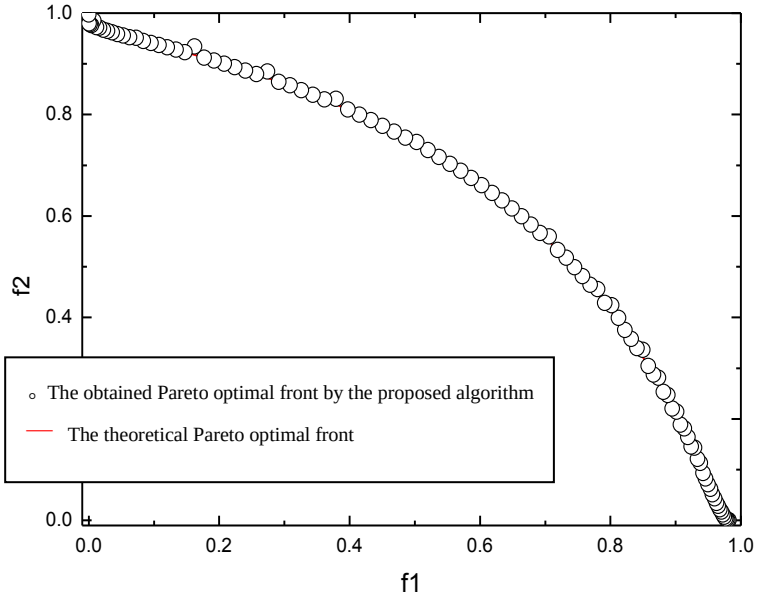

Figure 2 The obtained Pareto front of problem 2

Problem 3[5]

$$\min_{x} \quad f_1(x) = x_1$$

$$f_2(x) = g(x)\left[1 - \sqrt{x_1 / g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$$

其中，$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) / (n-1)$

$$0 \le x_i \le 1 \quad i = 2,3,\cdots,30.$$

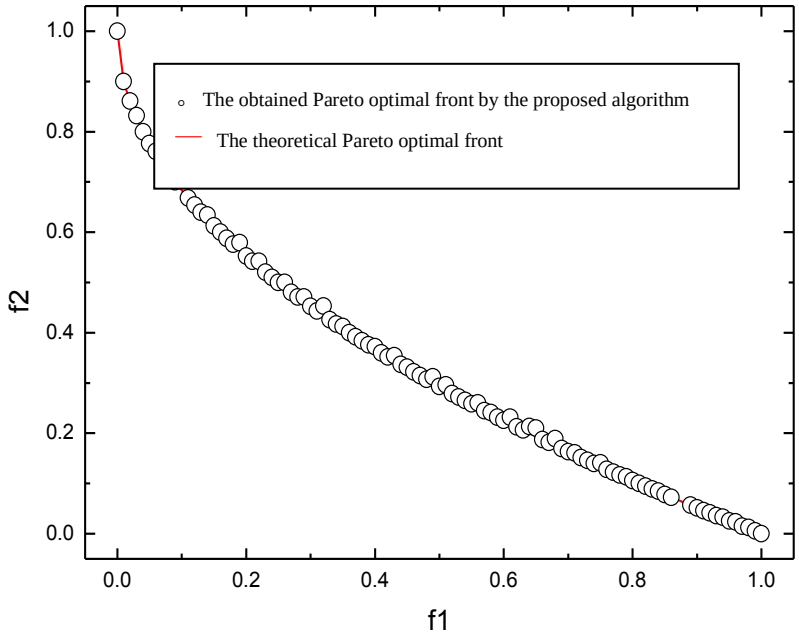

Figure 3 The obtained Pareto front of problem 3

5. Conclusion

In this paper, an improved SA algorithm is presented, in which a heuristic criterion for determining the temperature updating function of SA algorithm is applied in this paper, enabling the particle to escape the local optima. The improved SA algorithm is employed for solving multiobjective programming problem for the first time. In this study, some numerical examples are used to explore the feasibility and efficiency of the proposed algorithm. The experimental results indicate that the obtained Pareto front by the proposed algorithm is very close to the theoretical Pareto optimal front, and the solutions are also distributed uniformly on entire range of the theoretical Pareto optimal front. The proposed algorithm is simple and easy to implement, which provides another appealing method for further study on multiobjective programming problems.

References

[1]. S. Kirkpatrick, C.Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671-680.

[2]. F. Maffioli, Randomized heuristic for NP-hard problem. Advanced School on Stochastic in Combinatorial Optimization, World Scientific, 1987, 760–793.

[3]. B. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, Journal of the Operational Research Society 57(2006) 1143–1160

[4]. S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A Simulated Annealing Based Multi-objective Optimization Algorithm: AMOSA, IEEE Transaction on evolutionary computation 12(3) (2008) 269-283.

[5]. K. Smith, R. Everson, J. Fieldsend, Dominance measures for multi-objective simulated annealing, in Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC'04) 2004 23-30.

[6]. R. L. Yang, J. F. Gu, An efficient simulated annealing algorithm for global optimization, System Engineering-Theory & Practice (in Chinese) 5 (1997) 29-35.