Advances in Computer Science and its Applications
Vol. 1, No. 1, March 2012

www.worldsciencepublisher.org

A Robust Hybrid Restarted Simulated Annealing Particle Swarm Optimization Technique

Yudong Zhang, Lenan Wu

School of Information Science and Engineering, Southeast University, Nanjing, China

zhangyudongnuaa@gmail.com, wuln@seu.edu.cn

Abstract: Global optimization is a hot topic of applied mathematics and numerical analysis that deals with the optimization of a function or a set of functions. In this paper we proposed a hybrid restarted simulated annealing particle swarm optimization (RSAPSO) technique to find global minima more efficiently and robustly. The proposed RSAPSO combines the global search ability of PSO and the local search ability of RSA, and offsets the weaknesses of each other. The four benchmark functions demonstrate the superiority of our algorithm.

Keywords: global search, local search, simulated annealing, particle swarm optimization.

1 Introduction

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality [1]. It is commonly known as metaheuristic method as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

PSO does not use the gradient of the problem being optimized, which means PSO does not require for the optimization problem to be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, adaptive, etc [2].

PSO is widely applied in various fields. Lin et al. [3] proposed an immune PSO with functional link based neuro-fuzzy network for image backlight compensation. Fan et al. [4] integrates the PSO and entropy matching estimator to seek the optimal parameter of the generalized Gaussian distribution mixture model. Zahara et al. [5] used PSO to obtain the optimal thresholding of multi-level image segmentation. Zhang et al. [6] proposed an adaptive chaotic PSO for magnetic resonance brain image classification. Samanta et al. [7] integrated PSO to artificial neural networks (ANN) and support vector machine (SVM) for machinery fault detection, and demonstrated the results of PSO is superior to the ones of genetic algorithm. Zhang et al. [8] proposed a neural network by PSO for remote-sensing image classification.

Unfortunately, PSO is easy to be trapped into local minima and its calculation efficiency is low. In the worst case, when the best solution found by the group and the particles are all located at the same local minimum, it is almost impossible for particles to jump out and do further searching due to the velocity update equation [9]. The reason lies in the fact that PSO is powerful of global search but weak on local search. Therefore, our strategy is to introduce in a local search which is applied during each update cycle.

In this study, simulated annealing (SA) was chosen as the local search method. SA comes from annealing in metallurgy [10], a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects [11]. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one [12]. Moreover, we introduced in the restarted simulated annealing (RSA) technique to improve the performance of SA. The hybrid algorithm combines both global search provided by PSO and local search provided by RSA [13].

The structure of the rest of the paper is organized as follows: Next section 2 introduces the basic principles and procedures of PSO; Section 3 gives detailed description of RSA; Section 4 proposes the RSAPSO technique with particular explanation of every step; Experiments in section 5 demonstrate the RSAPSO outperforms GA, SA, and PSO; Final section 0 concludes the paper.

2 Particle Swarm Optimization

PSO is a population based stochastic optimization technique, which simulates the social behavior of a swarm of bird, flocking bees, and fish schooling. By randomly initializing the algorithm with candidate solutions, the PSO successfully leads to a global optimum. This is achieved by an iterative procedure based on the processes of movement and intelligence in an evolutionary system. Fig. 1 shows the flow chart of a PSO algorithm.

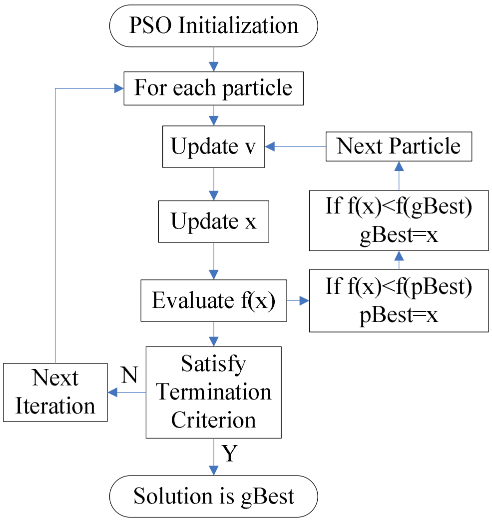


Fig. 1 Flow chart of the PSO algorithm

In PSO, each potential solution is represented as a particle. Two properties (position x and velocity v) are associated with each particle. Suppose x and v of the ith particle are given as

$$x = (x_{i1}, x_{i2}, \cdots, x_{iN}) \tag{1}$$

$$v = (v_{i1}, v_{i2}, \cdots, v_{iN}) \tag{2}$$

where N stands for the dimensions of the problem. In each iteration, a fitness function is evaluated for all the particles in the swarm. The velocity of each particle is updated by keeping track of the two best positions. One is the best position a particle has traversed so far and called "*pBest*". The other is the best position that any neighbor of a particle has traversed so far. It is a neighborhood best called "*nBest*". When a particle takes the whole population as its neighborhood, the neighborhood best becomes the global best and is accordingly called "*gBest*". Hence, a particle's velocity and position are updated as follows

$$v = \omega \cdot v + c_1 r_1 (pBest - x) + c_2 r_2 (nBest - x) \tag{3}$$

$$x = x + v\Delta t \tag{4}$$

where ω is called the "*inertia weight*" that controls the impact of the previous velocity of the particle on its current one. The parameters c_1 and c_2 are positive constants, called "*acceleration coefficients*". The parameters r_1 and r_2 are random numbers uniformly distributed in the interval [0,1]. These random numbers are updated every time when they occur. The parameter Δt stands for the given time-step.

The population of particles is then moved according to (3) and (4), and tends to cluster together from different directions. However, a maximum velocity v_{max}, should not be exceeded by any particle to keep the search within a meaningful solution space. The PSO algorithm runs through these processes iteratively until the termination criterion is satisfied [14].

3 Restarted Simulated Annealing

The SA algorithm is a probabilistic hill-climbing technique that is based on the annealing/cooling process of metals [15]. This annealing process occurs after the heat source is removed from a molten metal and its temperature starts to decrease. At each temperature level the energy of the metal molecules reduces, and the metal becomes more rigid. The procedure continues until the metal temperature has reached the surrounding ambient temperature, at which stage the energy has reached its lowest value and the metal is perfectly solid [16].

The SA procedure begins by generating an initial solution at random. At initial stages, a small random change is made in the current solution X_c. The new solution is called X_n. The perturbation depends on a temperate parameter T, and a scaling constant k.

$$pert(T) = k \times T \times r_3 \tag{5}$$

Here r_3 is a random value between 0 and 1 with uniform distribution. The temperature T decreases with each iteration of the algorithm, thus reducing the size of the perturbations as the search progresses. This mechanism produces large perturbation in the initial stages of the search and ensures that the resulting parameters are fine tuned towards the end of the optimization [17].

A move is made to the new solution X_n if it has better energy F or if the probability function has a higher value than a randomly generated number. Otherwise a new solution is generated, evaluated and compared again. The probability p of accepting a new solution X_n which called "Metropolis law" is given as follows:

$$p = \begin{cases} 1 & \text{if } F(X_n) < F(X_c) \\ \exp\left(\dfrac{F(X_c) - F(X_n)}{T}\right) & \text{otherwise} \end{cases} \tag{6}$$

In order to avoid getting trapped at local extrema points, the reduction rate of T should be slow enough. In this study the following method to reduce the temperature has been used:

$$T_n = T_0 \times \beta^n \tag{7}$$

Here T_0 is the initial temperature, β is the reduction constant, and n is the number of iterations. In total, most worsening moves may be accepted at initial stages, but at the final stage only improving ones are likely to be allowed. This can help the procedure jump out of a local minimum.

However, sometimes it is better to move back to a former solution that was significantly better rather than always moving from the current state. This process is called "restarting" of SA [18]. To do this we set the temperature to a former value and restart the annealing schedule. The decision to restart can be based on several criteria, including whether a fixed number of steps had passed, whether the current energy being too high from the best obtained so far, or randomly restart. In this paper, we restart the SA when the current energy is too high from the best energy because it performs best among all criterions [19]. The flowchart of RSA is shown in Fig. 2.

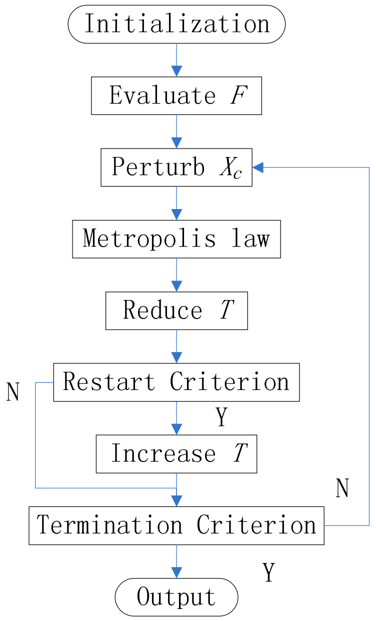


Fig. 2 Flowchart of RSA algorithm

4 Hybrid RSAPSO Technique

Traditional PSO algorithm suffers from getting trapped at the early stage. On the other side, RSA accepts a worse solution so it can escape from local point, resist earliness convergence, and increase the diversity of PSO [2]. Therefore, a new hybrid strategy was proposed which is referred to as RSAPSO. The proposed algorithm makes full use of the exploration ability both of PSO and of RSA and offsets the weaknesses of each other. The flowchart of RSAPSO is depicted in Fig. 3, and its detailed steps are given as follows:

Step 1 Initialize the population randomly;
Step 2 Evaluate each particle's fitness function *f*;
Step 3 Halve the population randomly: one half was updated by PSO as the formula (3)(4), the other half was updated by SA as the formula (5)(6)(7);
Step 4 Repeat Step 2 and Step 3 until the termination criteria was satisfied;
Step 5 Output the final results.

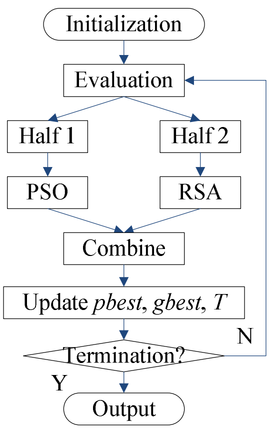


Fig. 3 Flowchart of RSAPSO

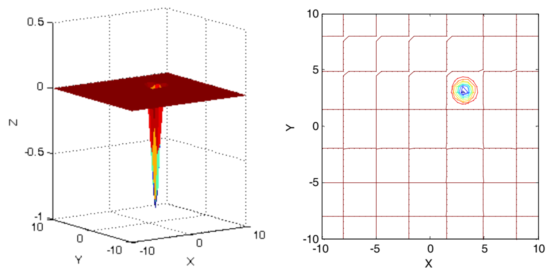
5 Experiments

The experiments are performed with an IBM P4 with 2GHz processor and 1GB memory and by the software platform Matlab 2010b. The parameters are determined by trial-and-error method. The number of population *NP* is set 20, the acceleration coefficient of *pbest* and *gbest* are both set 1.5, the initial temperature is set 100, and the reduction constant is set 0.995.

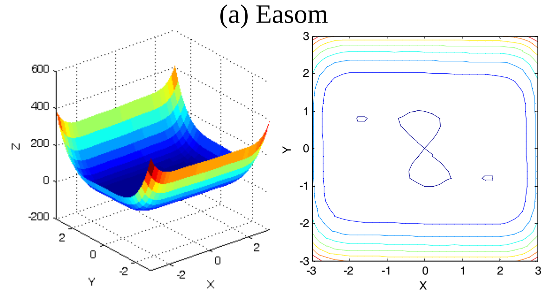
Tab. 1 Four Test functions

Function	Formula	Search Space	Global Minima
Easom	$-\cos(x_1)\cos(x_2)*\exp(-[(x_1-\pi)^2+(x_2-\pi)^2])$	[-10, 10]×[-10, 10]	$f(\pi, \pi) = -1$
Hump	$(4-2.1x_1^2+x_1^{4/3})x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	[-3, 3] × [-3 3]	$f(-0.0898,0.7126) = f(0.0898,-0.7126) = -1.0316$
Rosen-brock	$100(x_1^2-x_2)^2+(1-x_1)^2$	[-2 2]×[-2 2]	$f(1,1)=0$
Rastrigin	$20+x_1^2-10*\cos(2\pi x_1)+x_2^2-10*\cos(2\pi x_2)$	[-2 2]×[-2 2]	$f(0,0)=0$

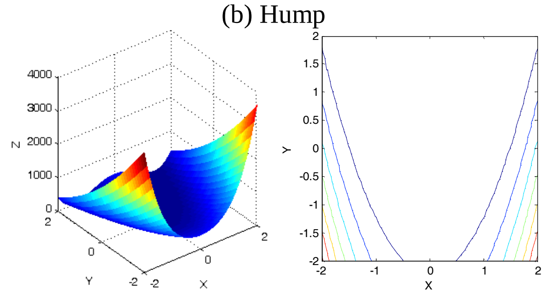
Four continuous optimization problems are chosen with their formula, search spaces, and global minima shown in Tab. 1. Their surface and contour images are shown in Fig. 4. In order to investigate our algorithm, we compared the proposed RSAPSO algorithm with GA, SA, and PSO. Each algorithm was run 100 times to remove the randomness, and the corresponding success rates were listed in Tab. 2. For the four test functions, the success rates of RSAPSO are 73%, 95%, 92%, and 90%, respectively, higher than those by GA, TS, and PSO. Therefore, the proposed RSAPSO is most robust.



(a) Easom



(b) Hump



(c) Rosenbrock

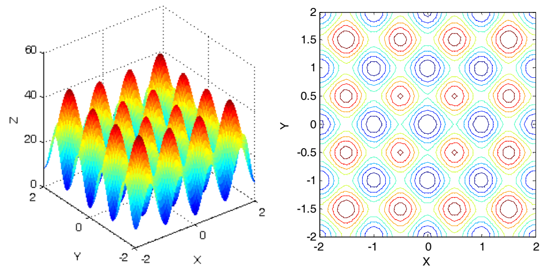

(d) Rastrigin

Fig. 4 Surface and Contour Images

Tab. 2 Success rates of different algorithms

Function	GA	SA	PSO	RSAPSO
Easom	25%	14%	62%	**73%**
Hump	89%	13%	95%	**95%**
Rosenbrock	23%	17%	87%	**92%**
Rastrigin	26%	19%	76%	**90%**

6 Discussions

In this paper, a robust hybrid RSAPSO method was proposed. It combines the exploration ability of PSO and of RSA. Experiments on four test functions all demonstrate the success rates of RSAPSO are more robust than GA, SA, and PSO. The future work mainly focuses on the influence of parameters of the proposed algorithm, and the application of the RSAPSO to other industrial fields such as stock prediction, weights optimization, and pattern recognition.

References

[1] M.S. Arumugam, M.V.C. Rao, A.W.C. Tan, A novel and effective particle swarm optimization like algorithm with extrapolation technique, Applied Soft Computing, 9 (2009) 308-320.

[2] Y. Zhang, J. Yan, G. Wei, L. Wu, Find multi-objective paths in stochastic networks via chaotic immune PSO, Expert Systems with Applications, 37 (2010) 1911-1919.

[3] C.-J. Lin, Y.-C. Liu, Image backlight compensation using neuro-fuzzy networks with immune particle swarm optimization, Expert Systems with Applications, 36 (2009) 5212-5220.

[4] S.-K.S. Fan, Y. Lin, C.-C. Wu, Image thresholding using a novel estimation method in generalized Gaussian distribution mixture modeling, Neurocomputing, 72 (2008) 500-512.

[5] E. Zahara, S.-K.S. Fan, D.-M. Tsai, Optimal multi-thresholding using a hybrid optimization approach, Pattern Recognition Letters, 26 (2005) 1082-1095.

[6] Y. Zhang, S. Wang, L. Wu, A Novel Method for Magnetic Resonance Brain Image Classification based on Adaptive Chaotic PSO, Progress in Electromagnetics Research, 109 (2010) 325-343.

[7] B. Samanta, C. Nataraj, Use of particle swarm optimization for machinery fault detection, Engineering Applications of Artificial Intelligence, 22 (2009) 308-316.

[8] Y. Zhang, S. Wang, L. Wu, PSONN used for Remote-Sensing Image Classification, Journal of Computational Information Systems, 6 (2010) 4417- 4425.

[9] V. Kumar, P. Balasubramanian, Kinetic parameter estimation in hydrocracking using hybrid particle swarm optimization, Fuel, 88 (2009) 2171-2180.

[10] A. Nakib, H. Oulhadj, P. Siarry, Image histogram thresholding based on multiobjective optimization, Signal Processing, 87 (2007) 2516-2534.

[11] H. Mohamadi, J. Habibi, M.S. Abadeh, H. Saadi, Data mining with a simulated annealing based fuzzy classification system, Pattern Recognition, 41 (2008) 1824-1833.

[12] S. Rubenthaler, T. Rydén, M. Wiktorsson, Fast simulated annealing in with an application to maximum likelihood estimation in state-space models, Stochastic Processes and their Applications, 119 (2009) 1912-1931.

[13] Y. Zhang, S. Wang, L. Wu, Y. Huo, Artificial Immune System for Protein folding model, Journal of Convergence Information Technology, 6 (2011) 55-61.

[14] A. Mahmood, Replicating web contents using a hybrid particle swarm optimization, Information Processing & Management, 46 (2010) 170-179.

[15] E. Cosola, K. Genovese, L. Lamberti, C. Pappalettere, A general framework for identification of hyper-elastic membranes with moiré techniques and multi-point simulated annealing, International Journal of Solids and Structures, 45 (2008) 6074-6099.

[16] E.W. McGookin, D.J. Murray-Smith, Submarine manoeuvring controllers' optimisation using simulated annealing and genetic algorithms, Control Engineering Practice, 14 (2006) 1-15.

[17] Y. Zhang, L. Wu, N. Naggaz, S. Wang, G. Wei, Remote-sensing Image Classification Based on an Improved Probabilistic Neural Network, Sensors, 9 (2009) 7516-7539.

[18] N. Cretu, M.-I. Pop, Acoustic behavior design with simulated annealing, Computational Materials Science, 44 (2009) 1312-1318.

[19] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognition, 43 (2010) 5-13.