Global Journal of Pure and Applied Mathematics
ISSN 0973-1768 Volume 5, Number 1 (2009), pp. 9-14

https://dx.doi.org/10.37622/GJPAM/5.1.2009.9-14

Performance of Population Based Metaheuristics on Some Non-Convex Noisy Deceptive Benchmark Test Function

Sanjeev Kumar Singh[1] and Munindra Borah[2]

[1]Sr. Lecturer, Department of Mathematics & Computer Sc.
Union Christian College, Ri-Bhoi-793122, Meghalaya

[2]Prof & Head, Department of Mathematical Sciences
Tezpur University, Assam, India.

Abstract

Several variations of meta-heuristics have been developed recently and each of them claims to outperform others. Through this paper we are going to do the comparative study of three methods, each of them has its origin in Von Neumann's Monte Carlo experiments. We have tested these methods with certain benchmark test problems and some new test functions introduced by us first time.

Keywords: SA(Simulated Annealing,), GA(Genetic Algorithm), RPS(Repulsive Particle Swarm Optimization), Global Optimization

Methods

(I) GA: This method is based on the Darwanian principle of survival of fittest introduced by Holland[3] . A population based method does the random selection of individuals. The selection scheme used here is tournament selection with suffling technique for choosing random pairs for mating. This routine includes jump mutation & creep mutation whichever is suitable and there is an option for single point crossover or uniform crossover. Niching(Sharing) option is also used.

(II) Modified RPSO- PSO(Particle Swarm Optimization) was introduced in 1995 by Kennedy and Eberhart [4] . It was inspired by the swarming behaviour as it is displayed by the flock of bird, a school of fish and even human social behaviour being influenced by other individual.

The repulsive particle swarm optimization is a variant of PSO was being introduced to overcome the pre-mature convergence. The modification of basic PSO scheme is to modify the velocity update formula when the swarm diversity becomes less then the a fixed value (i.e. d_{low}) The velocity is updated by the formula

$$v_{i+1} = \omega v_i + \alpha r_1(\hat{x}_i - x_i) + \omega\beta r_2(\hat{x}_{hi} - x_i) + \omega\gamma r_3 z$$

$$x_{i+1} = x_i + v_{i+1}$$

where,

- x is the position and v is the velocity of the individual particle. The subscripts i and $i+1$ stand for the recent and the next (future) iterations, respectively.
- $r_1,\ r_2,\ r_3$ are random numbers, $\in [0,1]$; $\alpha,\ \beta,\ \gamma$ are constants
- ω is inertia weight, $\in [0.01, 0.7]$; z is a random velocity vector
- $\hat{x}$ is the best position of a particle; x_h is best position of a randomly chosen other particle from within the swarm

Here the algorithm allows each swarm is allowed to search one step left and right, up and down. In the improved RPSO we allow the swarm to search at least fifteen step left and fifteen step right. This improves the performance of RPSO in many of the test function.

(III) Modified-Simulated Annealing: It is a global optimization method that distinguishes between different local minima introduced by Kirkpatrick, Gelatt and Vecchi [5]. Starting from the initial point, the algorithm takes a step and function is evaluated. When minimizing a function, any down hill step is accepted and the process repeats from this new points. The uphill decision is made by the metropolis criteria. Optimization process proceeds, the length of the steps decreases and algorithm closes in the global optima.

Test Functions

Brief Note of Test Functions

The objective of this paper is to present a comparative study of the performance of the Genetic algorithm and Repulsive particle swarm and Simulated Annealing methods on some bench mark numerical test functions [7] and some new test functions introduced first time [6]. These functions are difficult in nature. We present the new test functions in detail. We have given the graphical presentation of new test functions to understand the nature of difficulty.

Experiments

Algorithms used for the comparative study were Genetic Algorithm, Improved-Repulsive Particle swarm Optimization & Simulated Annealing. For all algorithms the dimensions were set to be adjustable, thus based on few preliminary experiments.

(I) Genetic algorithms: We have used and input file to pass the different parameters i.e. npopsiz=5, pcross=.9d0, npsibl1=(2*^{N} N= powers of 2) pmutate=0.02d0 and maxgen=200. Another params.f was included in the main program having three parameters population size=200, nchrommax=60 and nparamax=10. other two parameters are adjustable according to the dimensions of the problems.

(II) Modified-RPSO setting: RPSO have several parameters population size=40, In most of the cases n=30 works fine. Its value can be increased up to 50 to 100. A randomly chosen neighbors NN=31. The maximum no of decision variables MX=100, The Local search for this Improved RPSO has been increased up to 21, NSTEP=21, Number of iteration was set 1000.
Here the algorithm allows each swarm is allowed to search one step left and right, up and down. In the improved RPSO we allow the swarm to search at least ten step left and ten step right. This improves the performance of RPSO in many of the test function.

(III) Modified SA : The parameter T is very crucial in using the SA. Other parameters N is the dimension of the function can be changed from the parameter statement N=?. VM step length. T is imposed upon the system with the RT variable by T(I+1) = RT*T(i). The RT value was set 1.5
In a traditional SA for different random seed, result were different. So, we modified the program to save the optimum value in a particular iteration by setting the extra variable ffopt, and indexopt to get the particular iteration which gave the value of ffopt. We got these value printed. This we called it as Modified SA.

(IV) Numerical bench mark test function: For evaluating the three algorithms, we used 40 bench mark test functions and some of them given in the result table.

Ackley function: An $m-$variable ($m \geq 1$) function with search domain $[-15 \leq x_i \leq 30]$ for ($i = 1, 2, ..., m$) given as

$$f(x) = 20 + \exp(1) - 20\exp\left[-0.2\left((\sum_{i=1}^{m} x_i^2)/m\right)^{0.5}\right] - \exp\left[\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi x_i)\right]$$

is called the Ackley function. It is a multi-modal function. The global minimum of this function is $f(x^*) = 0$ for $x^* = (0, 0, ..., 0)$.

Easom function: This function is in 2 variables ($m = 2$) with search domain $[-100 \leq x_i \leq 100]$; ($i = 1, 2$) and $f(x^*) = -1$ at $x^* = (\pi,\ \pi)$. It is given as

$$f(x) = -\cos(x_1)\cos(x_2)\exp[-(x_1-\pi)^2 - (x_2-\pi)^2].$$

Griewank function: It is a typical multi-modal function with a large number of local minima in the search domain $[-600 \leq x_i \leq 600]$, $i = 1, 2, ..., m$ and global minimum $f(x^*) = 0$ at $x^* = (0, 0, ..., 0)$. It is given as

$$f(x)=\sum_{i=1}^{m}(x_i^2/4000)-\prod_{i=1}^{m}\cos(x_i/\sqrt{i})+1$$

Booth Function: A $2-$variable ($m=2$) function with search domain $[-10\le x_i\le 10]$; ($i=1,2$) given as.

$$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$$

This function is multimodal with the global minimum $f(x^*)=0$ at $x^*=(1,\ 3)$.

Matyas function: It is a 2-variable ($m=2$) function with search domain $[-10\le x_i\le 10]$; ($i=1,2$) and minimum $f(x^*)=0$ at $x^*=(0,\ 0)$. It is given as

$$f(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$$

Weierstrass function: The Weierstrass function [in its original form, $f(x)=\sum_{k=0}^{\infty}a^k\cos(b^k x)$ while b is an odd integer, $0<a<1$; $ab>(1+3\pi/2)$] is one of the most notorious functions (with almost fractal surface) that changed the course of history of mathematics. Weierstrass proved that this function is *throughout continuous but nowhere differentiable*. In its altered form this function in m ($m\ge 1$) variables with search domain $[-0.5\le x_i\le 0.5]$; ($i=1,2,...,m$) and the minimum $f(x^*)=0$ for $x^*=(0,\ 0,...,0)$; $a=0.5$; $b=3$; $\text{k}=20$, is given as.

$$f(x)=\sum_{i=1}^{m}\sum_{k=0}^{\text{k}}[a^k\cos(2\pi b^k(x_i+0.5))]-m\sum_{k=0}^{\text{k}}[a^k\cos(2\pi b^k 0.5)];\ \ x_i\in[-0.5,\ 0.5];\ \ i=1,2,...,m$$

Results

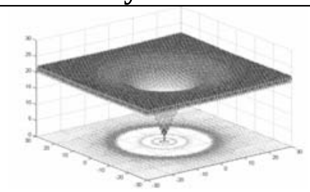
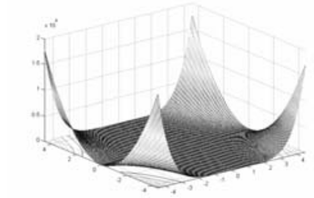
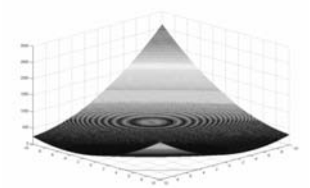
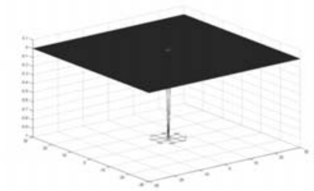
Results of some benchmark test problems						
SN	Functions	Dim	GA	I-RPS	SA	T. Value
1	Ackley Fun.	5	**0.00000**	**0.000000**	**0.189945E-07**	0
2	Easom Fun.	2	**-1.00001**	**-1.00000**	**-0.953971**	-1
3	Griewank Fun	5	**0.00000**	**0.000000**	**0.0172410**	0
4	Beale Fun	5	5.45315	**0.00000**	**0.1080137E-09**	0
5	Weierstrass Fun.	5	**0.00000**	0.02990	**0.7513280E-08**	0
6	Booth fun	2	-20.999	**0.00000**	**0.4368455E-09**	0.000000
7	Michalewich Fun	2	*******	**-1.80130**	**-1.80130**	-1.8013
8	Simple Quad Fun	2	-3846.15	**-3872.7**	**-3873.7**	3873
9	Hump Fun	2	**-1.00000**	-1.03162	-1.03162	-1
10	Matya fun	2	0.00000	0.00000	0.4148318E-09	0.00000

Discussion

The results clearly show that no methods are able to outperform for all the functions. In functions 1-3,10 three methods give the same results. Whereas for function 4 GA fails, 5-I-RPS fails, 6-GA fails,7-GA overflows, 8,10-Modified RPS & Modified SA outperforms GA, 9,11-GA outperforms Modified-RPS & Modified SA, 12-GA & Modified-RPS outperforms Modified SA & 13-all the three methods fails.

Acknowledgement

Graphical presentations (of most of the functions) are credited to Dr. AR Hedar, Dept. of Computer Science, Faculty of Computer & Information Sciences, Assiut University, Egypt. A few of the functions and their properties mentioned in different pages at the site (below) may, however, be taken with caution.
http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/go.htm.
For other new functions we have used MATLAB to draw the diagram.

<table>
<tr><th colspan="2">Bench mark test functions</th></tr>
<tr><th>Ackley Function</th><th>Beale Function</th></tr>
<tr><td></td><td></td></tr>
<tr><th>Booth Function</th><th>Easom Function</th></tr>
<tr><td></td><td></td></tr>
</table>

References

[1] Deb K. (1991), "*Genetic algorithms in multimodal function optimization*", Master's Thesis, (TCGA Report No. 89002). Tuscaloosa: University of Alabama.

[2] Goldberg, D.E., 1989, "*Genetic Algorithms in Search, Optimization and Machine Learning*", Kluwer Academic Publishers, Boston, MA.

[3] Holland, J. H. (1975), " *Adaptation in Natural and Artificial Systems*", University of Michigan Press, Ann Arbor

[4] Kennedy J. , and Eberhart R., (1995), "*Particle swarm optimization*", in Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942–1948,

[5] Kirkpatrick S. and C. D. Gelatt and M. P. Vecchi, (1983) "*Optimization by Simulated Annealing*", *Science, Vol 220, Number 4598*, pages 671-680,. http://citeseer.ist.psu.edu/kirkpatrick83optimization.html.

[6] Mishra, S.K. (2006(a)) "*Some Experiments on fitting og Gielis Curves by Simulated Annealing and Particle Swarm methods of Global Optimization*", Social Science Research Networks (SSRN): http://ssrn.com/abstract=913667, Working Papers series,

[7] Nagendra, S. (1997), "*Catalogue of Test Problems for Optimization Algorithm Verification*", Technical Report 97-CDR-110, General Electric Company.

[8] Srinivas, N and Deb, K. (1994). "*Multiobjective function optimization using non-dominated sorting genetic algorithms*". Evolutionary Computation Journal 2(3), 221-248.