International Journal of Information & Computation Technology.
ISSN 0974-2255 Volume 1, Number 2 (2011), pp. 17-24

http://www.irphouse.com

A comparative study of Repulsive Particle Swarm Optimization and Simulated Annealing on some Numerical Bench Mark Problems

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

[1]Lecturer(Selection Grader), Department of Mathematics & Computer Sc.
Union Christian College, Ri-Bhoi-793122, Meghalaya, India.
E-mail: sanjeev_kr_singh@yahoo.com
[2]Prof & Head, Department of Mathematical Sciences,
Tezpur University, Assam, India.
E-mail: mborah@tezu.ernet.in

Abstract

The objective of this paper is to test the performance of Repulsive Particle Swarm and Simulated Annealing on some test functions. Since Repulsive Particle Swarm Optimizations mimics the nature and SA(Simulated Annealing) follows the physical criteria, it will be interesting to see the performance of these two methods on the certain benchmark test functions. A brief idea of these functions are given in this section are as follows. These functions are also represented by graph to facilitate conceptualization of the nature of these functions by visual means.

Keywords: Particle Swarm Optimization(PSO), SA, Global Optimization, Genetic Algorithm.

Introduction

Optimization is central to any problem involving decision making, whether in Mathematics, Engineering or in Economics. The area of optimization has received enormous attention in recent years, primarily because of the rapid progress in computer technology, including the development and availability of user-friendly software, high speed and parallel processors. The optimization toolbox of MATLAB and many other commercial software like this has given a new dimension to it.

Global optimization

Optimization is essentially the art, science and mathematics of choosing the best among a given set of finite or infinite alternatives. The task of global optimization is to find a solution in the solution set for which the objective function obtaines its smallest value, the global minimum. Global optimization thus aims at determining not just "a local minimum" but "the smallest local minimum" with respect to the solution set[5].

Extending the class of functions to include multimodal functions makes the global optimization problem unsolvable in general. In order to be solvable some smoothness condition on the function in addition to continuity must be known.

Methods available

Many methods were developed in the late 1960s that performs well in optimization are

- **Genetic Programming (GP)** is a related technique popularized by John Koza[8] in which computer programs, rather than function parameters, are optimized. Genetic programming often uses tree-based internal data-structure to represent the computer programs for adaptation instead of the list structures typical of genetic algorithms.
- **Interactive Genetic Algorithm (IGA)** are genetic algorithms that use human evaluation. They are usually applied to domains where it is hard to design a computational fitness function, [1] for example, evolving images, music, artistic designs and forms to fit users' aesthetic preference.
- **Simulated Annealing (SA)** is a related global optimization technique that traverses the search space by testing random mutations on an individual solution[9]. A mutation that increases fitness is always accepted. A mutation that lowers fitness is accepted probabilistically based on the difference in fitness and a decreasing temperature parameter. In SA parlance, one speaks of seeking the lowest energy instead of the maximum fitness. SA can also be used within a standard GA algorithm by starting with a relatively high rate of mutation and decreasing it over time along a given schedule.
- **Tabu Search (TS)** is similar to Simulated Annealing in that both traverse the solution space by testing mutations of an individual solution. While simulated annealing generates only one mutated solution, tabu search generates many mutated solutions and moves to the solution with the lowest energy of those generated. In order to prevent cycling and encourage greater movement through the solution space, a tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the tabu list, which is updated as the solution traverses the solution space.
- **Ant Colony Optimization (ACO)** uses many ants (or agents) to traverse the solution space and find locally productive areas[3]. While usually inferior to genetic algorithms and other forms of local search, it is able to produce results in problems where no global or up-to-date perspective can be obtained, and thus the other methods cannot be applied.

- **Memetic Algorithm (MA)** also called *hybrid genetic algorithm* among others, is a relatively new evolutionary method where local search is applied during the evolutionary cycle. The idea of memetic algorithms comes from memes, which–unlike genes–can adapt themselves. In some problem areas they are shown to be more efficient than traditional evolutionary algorithms.

Through this paper we are going to compare the performance of Simulated Annealing and Particle Swarm Optimization on some bench mark test functions.

Simulated annealing (SA)

It is a generic probabilistic meta algorithm the global optimisation problem, namely locating a good approximation to the global optimum of a given function in a large search space.

The name and inspiration come from annealing in metallergy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. 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.

By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T (called the *temperature*), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero. The allowance for "uphill" moves saves the method from becoming stuck at local minima which are the bane of greedier methods.

Repulsive Particle Swarm in Global optimization:

Particle swarm optimization' (PSO) is a form of swarm intelligence[4]. This is modelled by particles in multidimensional space that have a position and a velocity. These particles are flying through hyperspace and have two essential reasoning capabilities: their memory of their own best position and knowledge of the swarm's best. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. There are two main ways this communication is done:

- a global best that is known to all
- “neighborhood” bests where each particle only communicates with a subset of the swarm about best positions

The repulsive particle swarm optimization is variant of Particle swarm optimization [5] and belongs to a stochastic evolutionary global optimizers. There are several different realizations of RPSO. Common to all these realization is the repulsion between the particles. This can prevent the swarm trapped in local minima,

which would cause a premature convergence and would lead the optimization algorithm to fail to find the global optimum. The other variants use a dynamic scheme In RPSO the future velocity $\boldsymbol{v_{i+1}}$ of a particle at position $\boldsymbol{x}$ with a recent velocity $\boldsymbol{v_i}$ is calculated by

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

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, $\in[0,1]$; α, β, γ 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

Test Functions

The objective of this paper is to present a comparative study of the performance of the Repulsive Particle Swarm and Simulated Annealing method on the following functions. These functions are difficult in nature. We present these functions in details. A graphical presentation is also given.

I Scaffer Function: In the search domian $x_1, x_2 \in [-100,100]$ this function is defined as follows and has $f_{\min}(0,0) = 0.$

$$f(x) = 0.5 + \frac{\sin^2\sqrt{(x_1^2 + x_2^2)} - 0.5}{\left[1 + 0.001(x_1^2 + x_2^2)\right]^2}.$$

II Perm Function#1: : In the domain $x \in [-4,4]$, the function has f_{min} =0 for x=(1,2,3,4).

It is specified as $f(x) = \left[\sum_{k=1}^{4}\sum_{i=1}^{4}(i^k + \beta)\left\{\left(\frac{x_i}{i}\right)^k - 1\right\}\right]^2$ The value of $\beta(=50)$ introduces difficulty to optimization. Smaller values of β raises difficulty further.

III Power-Sum Function : Defined on four variables in the domain $x \in [0,4]$, this function has $f_{\min} = 0$ for any permutation of x=(1,2,2,3). The function is defined as

$f(x) = \left[\sum_{k=1}^{4} b_k - \sum_{i=k}^{4} x_i^k\right]^2$; $b_k = (8,18,44,114)$ for k = (1,2,3,4) respectively.

IV Weiestrass 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 ($m \geq 1$) variables with search domain $[-0.5 \leq x_i \leq 0.5]$; ($i=1,2,...,m$) and the minimum $f(x^*)=0$ for $x^*=(0,\ 0,...,0);\ a=0.5;\ b=3;\ \mathbb{k}=20$, is given as

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

V Zero-Sum Function (N#7) : Defined in the domain $x \in [-10,10]$ this function (in $m \geq 2$) has $f(x)=0$ if $\sum_{i=1}^{m} x_i = 0$. Otherwise $f(x)=1+\left(10000\left|\sum_{i=1}^{m} x_i\right|\right)^{0.5}$. This function has innumerably many minima but it is extremely difficult to obtain any of them. Large is the value of m (dimension), it becomes more difficult to optimize the function.

I	U(I)	V(I)	Y(I)
1	.286	.645	4.284
2	.973	.585	4.149
3	.348	.310	3.877
4	.276	.058	.533
5	.973	.455	2.211
6	.543	.779	2.389
7	.957	.259	2.145
8	.948	.202	3.231
9	.543	.028	1.998
10	.793	.099	1.379
11	.936	.142	2.106
12	.889	.296	1.428
13	.006	.175	1.011
14	.828	.180	2.179
15	.399	.842	2.858
16	.617	.039	1.388
17	.939	.103	1.651
18	.784	.620	1.593
19	.072	.158	1.046
20	.889	.704	2.152

VI. Judge function: This is a multimodel function defined as

$$f(x)=(x_1+x_2 \sin^2(u_i)+x_2^2 \cos(v_i)-y_i)^2$$

This function has two optima f(0.846,1.23) =16.0817 which is a global minima and f(2.35,-0.319)=20.9805 which is a local minima.

This function has been adapted from Bill Goffe's Simman (Simulated Annealing Problem).

Experiments

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

B RPSO setting: RPSO have several parameters population size 40, In most of the cases 30 works fine. Its value can be increased upto 50 to 100. A randomly chosen neighbours 15. The maximum no of decision variables 100, The Local search Number of iteration was set 1000.

C. Modified SA: The parameter T is very crucial in using the SA. Other parameters N is the dimesion 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

Results

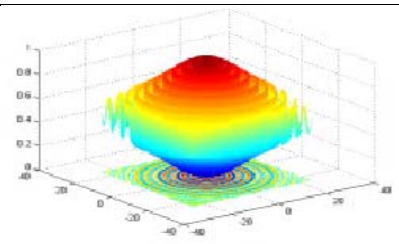
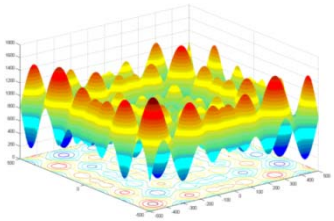
Results Comparing Repulsive Particle Swarm & Simulated Annealing			
Test function where Repulsive Particle Swarm performs better than Simulated Annealing			
Name of the Functions	Repulsive Particle Swarm(RPS)	Simulated Annealing(SA)	True Value
I Scaffer Function	0	9.72E-03	0
II Perm Function#1	0	16	0
III Power-Sum Function 4	0	100	0
Test Functions where Simulated Annealing performs better then Repulsive Particle Swarm			
IV Weiestrass Function	0.0299	7.51E-09	0
V Zero-Sum Function (N#7)	1	0	0
VI Judge Function	20.4050117	16.08173069	16.08173069
Legands : Here two methods have succeeded		Here two methods have failed	True Value

Conclusion

Our program of Repulsive Particle Swarm has given a good result for the functions like Schafer function, Perm function#1, Power-Sum Function 4 where Simulated Annealing have failed in these functions.

Whereas Simulated Annealing has performed better for Weiestrass function, Zero-Sum function and Judge function then Repulsive Particle Swarm.

Appendix

Scaffer Function	**Perm function#1**
	
* Graphical presentations (of most of the functions) are creditable 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	

Reference

[1] Coello-Coello, C. A. (1998) *"An updated survey of GA-Based Multiobjective Optimization Technique"* (techanical Report Lania-RD-09-08). Laboratorio Nacional de Inform ática Avanzada (LANIA), Xalapa, Veracruz, México.

[2] D. E. Goldburg, *Genetic Algorithm in search, Optimization, and Machine Learning*, Reading, MA Addition-Wesley, 1989.

[3] Dorigo M. Ant Coloney optimization web page, http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html

[4] Eberhart and Kennedy, "A New Optimizer using Particle Swarm Theory", *Proceedings sixth symposium on Micro Machine and Human Science, pp. 39-43. IEEE Service center.* Piscataway, NJ, 1995.

[5] Horst, R. and Pardalos, P. M. (Eds.)(1995). "*Handbook of Global Optimization.*" Dordrecht, Netherlands: Kluwer,

[6] Huang, V.L., Suganthan, P.N. and Liang, J.J. (2006) "Comprehensive Learning Particle Swarm Optimizer for Solving Multi-objective Optimization Problems", *International Journal of Intelligent Systems, 21,* pp. 209-226 (Wiley Periodicals, Inc Published online in Wiley Inter Science www.interscience.wiley.com)

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975

[8] Koza, J. R. (1992) "*Genetic Programming: On Programming of Computers by Means of Natural Selection".* The MIT Press

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

[10] V. Cerny, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm". *Journal of Optimization Theory and Applications*, 45:41-51, 1985