ISSN:0975-9646

Dolly Verma et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3552-3555

Proposed Methodology of a Hybrid Approach using Particle Swarm Optimization with Linear Crossover to solve Continuous Optimization Problem

Dolly Verma[1], Rashmi Shrivas[2]

[1,2]*MATS School of Engg. & Technology*

Abstract— The focus of this report is on the second topic. Actually, there are already lots of computational techniques inspired by biological systems. For example, artificial neural network is a simplified model of human brain; genetic algorithm is inspired by the human evolution. Here we discuss another type of biological system - social system, more specifically, the collective behaviors of simple individuals interacting with their environment and each other. Someone called it as swarm intelligence.

Particle swarm optimization PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.Keywords— Particle swarm optimization, Evolutionary Algorithms, Genetic Crossover.

1. INTRODUCTION

The term "Artificial Life" (A Life) is used to describe research into human-made systems that possess some of the essential properties of life. A Life includes two-folded research topic:

i. A Life studies how computational techniques can help when studying biological phenomena

ii. A Life studies how biological techniques can help out with computational problems

Genetic algorithm is a search method that employs processes found in natural biological evolution. These algorithms search or operate on a given population of potential solutions to find those that approach some specification or criteria. To do this, the genetic algorithm applies the principle of survival of the fittest to find better and better approximations. At each generation, a new set of approximations is created by the process of selecting individual potential solutions (individuals) according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of population of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

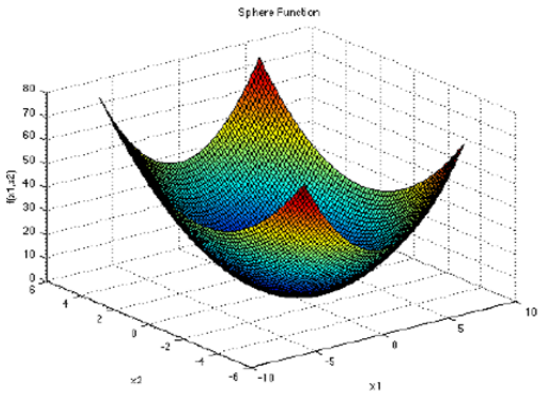
Genetic algorithm (GAs) were invented by John Holland in the 1960s and were developed with his students and colleagues at the University of Michigan in the !(70s. Holland's original goal was to investigate the mechanisms of adaptation in nature to develop methods in which these mechanisms could be imported into computer systems.

GA is a method for deriving from one population of "chromosomes" (e.g., strings of ones and zeroes, or bits) a new population. This is achieved by employing "natural selection" together with the genetics inspired operators of recombination (crossover), mutation, and inversion. Each chromosome consists of genes(e.g. bits), and each gene is an instance of a particular allele(e.g,0 or 1).The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average those chromosomes that have a higher fitness factor(defined below),produce more offspring than the less fit ones. Crossover swaps subparts of two chromosomes, roughly imitating biological recombination between two single chromosome ("haploid") organisms; mutation randomly changes the allele values of some locations (locus) in the chromosome; and inversion reverses the order of a contiguous section of chromosome.

2. PRELIMINARIES

In the field of evolutionary computation, it is common to compare different algorithms using a large test set, especially when the test involves function optimization. However, the effectiveness of an algorithm against another algorithm cannot be measured by the number of problems that it solves better. Here some bench mark functions discussed below for extracting the background information:

2.1 Sphere function:



$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$$

Fig: Sphere Function

2.1.1 Dimensions: d

The Sphere function has d local minima except for the global one. It is continuous, convex and unimodal. The plot shows its two dimensional form

2.1.2 Input Domain:

The function is usually evaluated on the hypercube $x_i x_i \in$ [5.12, 5.12], for all i = 1, …, d.

2.1.3 Formula:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

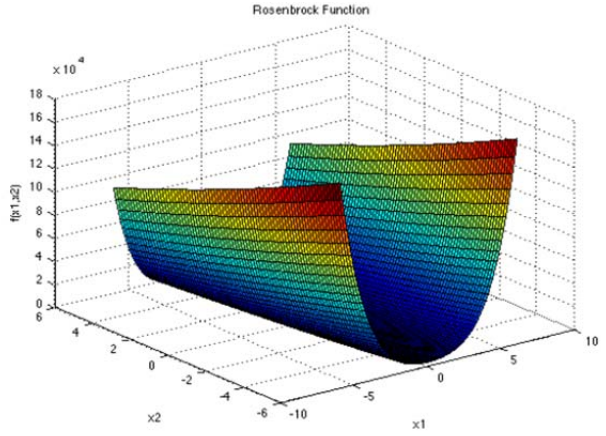
2.1.4 Global Minimum:

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \ldots, 0)$$

2.1.5 Search Domain:

$$-\infty \leq x_i \leq \infty, \quad 1 \leq i \leq n$$

2.1 Rosenbrock Function:



$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

Fig: Rosenbrock Function

2.2.1 Dimensions: d

The Rosenbrock function, also referred to as the Valley or Banana function, is a popular test problem for gradient based optimization algorithms. It is shown in the plot above in its two dimensional form. The function is unimodal, and the global minimum lies in a narrow, parabolic valley. However, even though this valley is easy to find, convergence to the minimum is difficult.

2.2.2 Input Domain:

The function is usually evaluated on the hypercube $x_i x_i \in$ [5,10], for all i = 1, …, d, although it may be restricted to the hypercube $x_i x_i \in$ [2.048,2.048], for all i = 1, …, d.

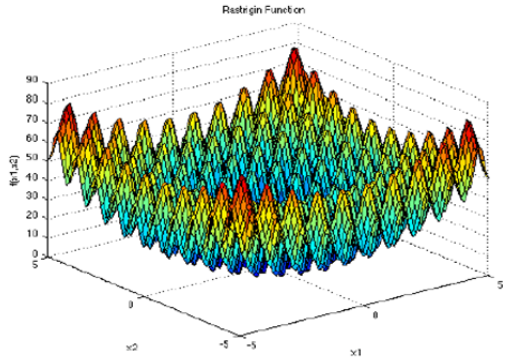
2.2.3 Global Minimum:

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (1, \ldots, 1)$$

2.2.4 Formula:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$$

2.2 Rastrigin Function:



$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$$

Fig: Rastrigin Function

2.3.1 Description: ***Dimensions: d***

The Rastrigin function has several local minima. It is highly multimodal, but locations of the minima are regularly distributed. It is shown in the plot above in its two dimensional form.

2.3.2 Input Domain:

The function is usually evaluated on the hypercube $x_i x_i \in$ [5.12, 5.12], for all i = 1, …, d.

2.3.3 Global Minimum:

$$f(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, \ldots, 0)$$

2.3.4 Formula:

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$$

2.3 Griewank Function:

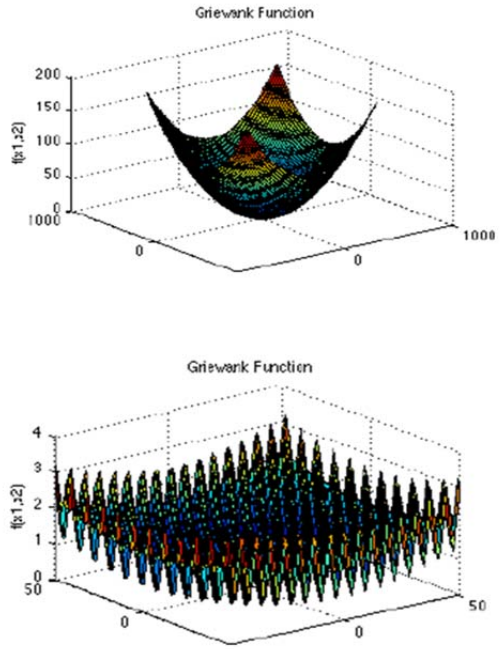


Fig: Griewank Fun ction

$$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

2.4.1 Dimensions: d

The Griewank function has many widespread local minima, which are regularly distributed. The complexity is shown in the zoomed in plots.

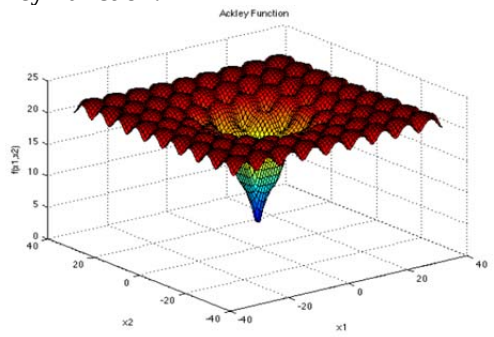
2.4.2 Input Domain:

The function is usually evaluated on the hypercube $x_i x_i \in$ [600,600], for all i = 1, …, d.

2.4.2 Formula:

$$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

2.5 Ackley Function:



$$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)\right) + a + \exp(1)$$

Fig: Ackley Function

2.5.1 Dimensions: d

The Ackley function is widely used for testing optimization algorithms. In its two dimensional form, as shown in the plot above, it is characterized by a nearly flat outer region, and a large hole at the centre. The function poses a risk for optimization algorithms, particularly hill climbing algorithms, to be trapped in one of its many local minima. Recommended variable values are: a = 20, b = 0.2 and c = 2π.

2.5.2 Input Domain:

The function is usually evaluated on the hypercube $x_i x_i \in$ [32.768, 32.768], for all i = 1, …, d, although it may also be restricted to a smaller domain.

2.5.3 Formula:

$$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)\right)$$

3. Proposed Methodology

3.1 Particle Swarm Optimization (PSO)

- Particle Swarm Optimization is a technique used to explore the search space of a given problem to find the settings or parameters required to maximize a particular solution.
- PSO have been proposed two equations:

$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})$

And the position is updated using:

$x_{id} = x_{id} + v_{id}$

shows a general flow chart of PSO.

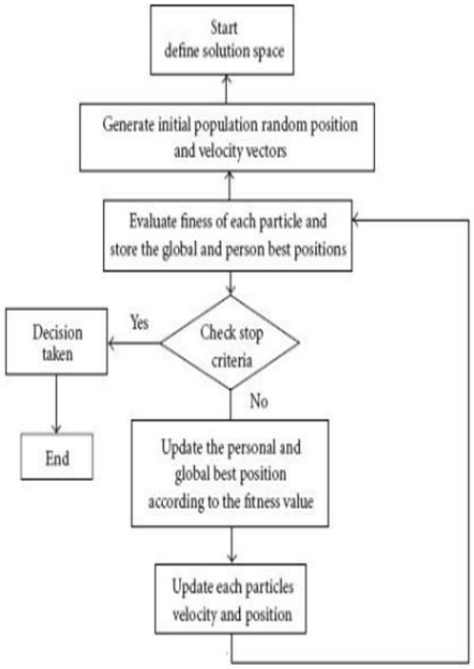


Fig: Flowchart of PSO

3.2 Genetic Crossover

A Genetic Algorithm (SGA) is a computational abstraction of biological evolution that can be used to solve optimization problems. In a Genetic Algorithm a population of candidate solution to an optimization problem is evolved toward better solutions. Crossover is of two types, Binary and Real Coded crossover. In this project linear crossover is used.

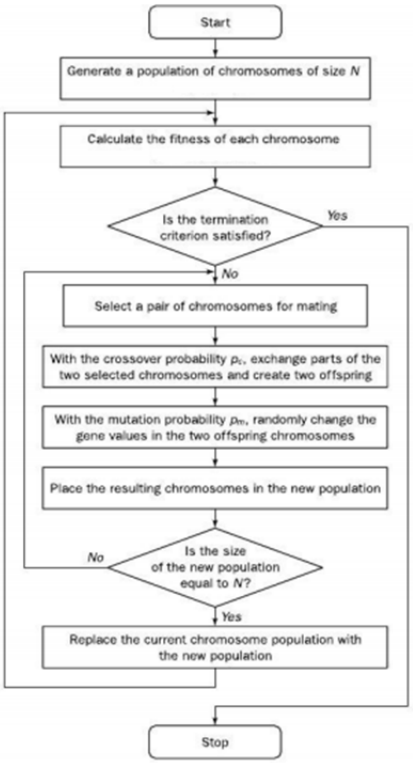


Fig: Flowchart of GA

3.3 Linear Crossover

Let $(x_1^{(1,t)}\ x_2^{(1,t)}\ x_3^{(1,t)}\ x_4^{(1,t)}\ \ldots x_n^{(1,t)})$ and $(x_1^{(2,t)}\ x_2^{(2,t)}\ x_3^{(2,t)}\ x_4^{(2,t)}\ \ldots x_n^{(2,t)})$ are two parent solutions of dimension n at generation t.

It generates three offsprings:

$0.5(x_i^{(1,t)} + x_i^{(2,t)})$

$1.5x_i^{(1,t)} - x_i^{(2,t)})$

$-0.5x_i^{(1,t)} + x_i^{(2,t)})$

i = 1,2,…….,n

3.4 PSO with Genetic Linear Crossover Operator

PSO shares many similarities with evolutionary computing techniques in general and GAs in particular. PSO and GA techniques begin with a group of a randomly 9 Hybrid PSO and GA models generated population; both utilize a fitness value to evaluate the population. They update the population and search for the optimum with random techniques.

- PSO algorithms have been successful in solving a wide variety of problems, their performance is criticized one certain aspects.
- For example the problem of the loss of diversity after subsequent iterations which lead to premature convergence leading to suboptimal solution.
- One of the simplest methods to overcome the problem of diversity loss is to capitalize the strengths of Evolutionary Algorithms (EA) and PSO together in an algorithm. Therefore crossover operator is being hybridized to the standard PSO algorithm.

4. ALGORITHM

1. **Initialize variable**
 i:=0
size:=0
 dim:=0
2 .do until i=max(**number of function evaluation**)
3**.** do until size:=swarm size
4. do until dim:= problem dimension
modify velocity
 modify position
 end
5.compute fitness (**updated position**)
6.[? If needed]
 update historical information for Pi & Pg
 end for size
7.if [**crossover**]
 i) select two random particles as parent particles from the current swarm for
 crossover operation.
 ii) Apply crossover
 New offsprings
8.if (new offspring is better than worst parent particle)
 replace it.
9.if Pg meets problem required then terminate
 end if
 end if
 end for i

REFERENCES

[1] Dongyong Yang, Jinyin Chen, Matsumoto Naofumi, "Self-adaptive Crossover Particle Swarm Optimizer for Multi-dimension Functions Optimization", Third International Conference on Natural Computation (ICNC 2007) 0-7695-2875-9/07, IEEE Computer Society, 2007.

[2] Zhi-Feng Hao**,** Zhi-Gang Wang, Han Huang**,** "A Particle Swarm Optimization Algorithm with Crossover Operator", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.

[3] Jiahua Xie, Jie Yang, "A Novel Crossover Operator for Particle Swarm Algorithm", International Conference on Machine Vision and Human-machine Interface, 978-0-7695-4009-2/10, IEEE, 2010, pp. 161-164.

[4] James Kennedy, Russell Eberhart, "Particle Swarm Optimization", 0-7803-2768-3/95/$4.00 0 1995 IEEE, pp. 1942-1948.

[5] Yuhui Shi, Russell Eberhart, "A Modified Particle Swarm Optimizer", 0-7803-4869-9198, IEEE, pp. 69-73.

[6] Ashuri, B. and Tavakolan, M., "Fuzzy Enabled Hybrid Genetic Algorithm–Particle Swarm Optimization Approach to Solve TCRO Problems in Construction Project Planning" *J. Constr. Eng. Manage.*, 138(9), 2012, 1065–1074.

[7] Millie Pant, Radha Thangaraj and V. P. Singh, "Particle Swarm Optimization with Crossover Operator and its Engineering Applications", IAENG International Journal of Computer Science, 36:2, IJCS_36_2_02, 2009.

[8] Jong-Bae Park, Yun-Won Jeong, "A Hybrid Particle Swarm Optimization Employing Crossover Operation for Economic Dispatch Problems with Valve-point Effects", The 14th International Conference on Intelligent System Applications to Power Systems, ISAP 2007, pp. 281-286.

[9] W. Y. Qian, G. L. Liu, "Particle Swarm Optimization with Crossover Operator for Global Optimization Problems", Applied Mechanics and Materials, Vols 373-375, pp. 1131-1134, Aug. 2013.

[10] Kwang Y. Lee and Jong-Bae Park, "Application of Particle Swarm Optimization to Economic Dispatch Problem: Advantages and Disadvantages", *IEEE*, 2010.

[11] Bineet Mishra, Rakesh Kr. Patnaik, "Genetic Algorithm & is Variants: Theory & Applications", Project report, NIT Rourkela, 2009.

[12] Millie Pant, Radha Thangaraj and Ajith Abraham, "A New Particle Swarm Optimization Algorithm Incorporating Reproduction Operator for Solving Global Optimization Problems", Seventh International Conference on Hybrid Intelligent Systems, IEEE, 2007, 144-149.

[13] Sapna Katiyar, "A Comparative Study of Genetic Algorithm and the Particle Swarm Optimization", AKGEC International Journal of Technology, Vol. 2, No. 2, pp 21-24.

[14] J.J.Liang. Qin A.K., Suganthan P.N., Baskar S., "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", Evolutionary Computation, IEEE Transactions, **ISSN :** 1089-778X, Volume:10, Issue: 3 , pp. 281-295, June 2006.

[15] Jong-Bae Park, Ki-Song Lee , Joong-Rin Shin, Lee, K.Y., "A particle swarm optimization for economic dispatch with nonsmooth cost functions", Power Systems, IEEE Transactions, **ISSN :** 0885-8950, Volume:20 , Issue: 1, pp. 34-42.

[16] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with the valve point loading," IEEE Trans. on Power Systems, Vol. 8, No. 3, pp. 1325-1332, Aug. 1993.

[17] Picheny, V., Wagner, T., & Ginsborger, D. , "A benchmark of kriging based infill criteria for noisy optimization", 2012.