10.1109/HIS.2006.264940

Hybrid Evolutionary Algorithm Based on PSO and GA mutation

A. A. A. Esmin
Dept. of Computer Science
UFLA, MG-Brazil
ahmed@dcc.ufla.br

G. Lambert-Torres
Dept. of Electrical Engineering
UNIFEI, MG-Brazil
germano@unifei.edu.br

G. B. Alvarenga
Dept. of Computer Science
UFLA, MG-Brazil
guilherme@dcc.ufla.br

Abstract

This paper presents a hybrid evolutionary algorithm based on Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs). The main idea is to integrate PSO with GA mutation method. Simulations for a series of benchmark test functions show that the hybrid proposed method possess better ability to find the global optimum than the standard PSO algorithm.

1. Introduction

Many problems in engineering involve the minimization of a suitable cost function. Thus efficient optimization algorithms are of interest. Genetic algorithms (GAs) are a family of computational models who is inspired by evolution. [1,2]. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information.

Particle swarm optimization (PSO) is also an evolutionary introduced in [3], as an alternative to the standard GAs. The PSO was inspired by insect swarms and has, then, showed to be a competitor to the standard GA for function optimization. Compared to GAs, the PSO has much more intelligent background and could be performed more easily. The PSO has been applied widely in the function optimization, artificial neural networks' training, fuzzy control and some other fields. Since then several improved PSO algorithms have been developed.

Some hybrid PSO algorithms were proposed by adding GAs' idea in [4,5]. The algorithm selects a certain number of particles according to the hybrid probabilities at each stage of iteration. The particles are randomly matched into couples. Each couple reproduces two children by crossover. Then the children are used to replace their parents of the previous particles to keep the number of particles unchanged.

This paper proposes a new model called Hybrid Particle Swarm Optimizer with Mutation (HPSOM), by integrate the mutation process often used in GA into PSO. This process allows the search to escape from local optima and search in different zones of the search space. Simulations for a series of benchmark test functions show that the proposed hybrid algorithms possess better ability to find the global optimum than that of the standard PSO algorithm.

2. The Standard GA and Standard PSO

2.1 Standard Genetic Algorithms

The GAs algorithms have been applied successfully to problems in many fields. The GAs are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) as encoded information individuals that evolve in time [2,3].

Generally, GA comprises three different phases of search: phase 1: creating an initial population; phase 2: evaluating a fitness function; phase 3: producing a new population.

A genetic search starts with a randomly generated initial population within which each individual is evaluated by means of a fitness function. Individuals in this and subsequent generations are duplicated or eliminated according to their fitness values. Further generations are created by applying GA operators. This eventually leads to a generation of high performing individuals [1-2].

There are usually three operators in a typical genetic algorithm [12]: the first is the production operator (elitism) which makes one or more copies of any individual that posses a high fitness value; otherwise, the individual is eliminated from the solution pool; the second operator is the recombination (also known as the 'crossover') operator. This operator selects two individuals within the generation and a crossover site and carries out a swapping operation of the string bits to the right hand side of the crossover site of both individuals. Crossover operations synthesize bits of knowledge gained from both parents exhibiting better than average performance. Thus, the probability of a better performing offspring is greatly enhanced; the third operator is the 'mutation' operator. This operator acts as a background operator and is used to explore some of the invested points in the search space by randomly flipping a 'bit' in a population of strings. Since frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its activation.

2.2 Standard PSO Algorithm

The Particle Swarm Optimization (PSO) algorithm is a new optimization algorithm inspired by social behavior in nature. Like Genetic Algorithms, the PSO is a population-based optimization method that searches multiple solutions in parallel [3]. However PSO employs a "cooperative" strategy unlike GA, which utilizes a "competitive" strategy.

PSO technique finds the optimal solution using a population of particles. Each particle represents a candidate solution to the problem. The particles change their positions by flaying a round the search space until a relatively unchanging has been encountered, or the stop criteria is satisfied. It can be used to solve a wide array of different optimization problems. The PSO definition is presented as follows:

Each individual particle i has the following properties: A current position in search space, $\mathbf{x}_i$, a current velocity, $\mathbf{v}_i$, and a personal best position in search space, $\mathbf{y}_i$.

The personal best position, $\mathbf{y}_i$, corresponds to the position in search space where particle i had the smallest value as determined by the objective function f, assuming a minimization task.

The global best position denoted by $\breve{y}$ represents the position yielding the lowest error amongst all the $\mathbf{y}_i$.

Equations (1) and (2) define how the *personal* and *global* best values are updated at time t, respectively. It is assumed below that the swarm consists of s particles, Thus $i \in 1..s$.

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if} \quad f(y_i(t) \le f(x_i(t+1))) \\ x_i(t+1) & \text{if} \quad f(y_i(t) > f(x_i(t+1))) \end{cases} \quad (1)$$

$$\begin{aligned} \breve{y}(t) \in \{y_0(t), y_1(t), \ldots\ldots, y_s(t)\} \mid \quad & f(\hat{y}(t)) \\ = \min \{f(y_0(t)), f(y_1(t)), \ldots\ldots, & f(y_s(t))\} \end{aligned} \quad (2)$$

Each particle in the swarm is updated during iteration by using the equations (3) and (4). Two pseudo-random sequences, $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are used to effect the stochastic algorithm nature. For all dimensions $j \in 1..n$, let $x_{i;\ j}$, $y_{i;\ j}$ and $v_{i;\ j}$ be the current position, current personal best position and velocity of the j th dimension of the i th particle. The velocity update step is

$$\begin{aligned} v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + \\ c_2 r_{2,j}(t)[\breve{y}_j(t) - x_{i,j}(t)] \end{aligned} \quad (3)$$

The new velocity is then added to the current position of the particle to obtain the new position:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

The PSO algorithm consists of repeated application of the equations (1-4).

3. The Hybrid Algorithm

Different variants of the PSO algorithm were proposed. Some of these variants have been proposed to incorporate the capabilities of other evolutionary algorithms, such as hybrid versions of PSO or the adaptation of PSO parameters, creating the adaptive PSO versions. Many authors have considered incorporating selection, mutation and crossover, as well as differential evolution, into the PSO algorithm. As a result, hybrid versions of PSO have been created and tested, include the Hybrid of Genetic Algorithm and PSO (GA-PSO) and Evolutionary PSO (EPSO). [8-11]

This paper describes the Hybrid Particle Swarm Optimization with Mutation (HPSOM) algorithm to solve the stagnation problem and to prevent the particles from being trapped in local minima. To solve this problem, the HPSMO integrate the mutation process often used in GA into PSO. This process starts with the random choice of a particle in the swarm and moves to different positions inside the search area. In this paper the mutation process is employed by using the following equation:

$$mut\ (p[k]) = p([k]\mathrm{x} - 1) + \omega \quad (5)$$

Where $p[k]$ is the random choice particle from the swarm and ω is randomly obtained within the range $[0, 0.1\mathrm{x}(x_{\max} - x_{\min})]$, representing 0.1 times the length of the search apace.

4. The Numerical Experiment

In a previous work, standard PSO has been used to minimize the loss power problem, producing competitive results [12]. In this paper the performance of the HPSOM algorithm was compared to the standard PSO algorithm by running both algorithms multiple times and computing the average on four benchmark problems, all minimisation problems. These four functions have been commonly used in other studies on particle swarm optimization (e.g. [8-11]). The first two functions were unimodal while the last two were multimodal with many local minima.

Spherical: $f_1(x) = \sum_{i=1}^{n} x_i^2$

Rosenbrock: $f_2(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$

Griewank: $f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$

Rastrigin: $f_4(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$ ψ

The search space for the experiments is listed in Table 1. All experiments consisted of 100 runs. The PSO and HPSOM parameters were set to the values c1=c2 = 2.0 and a linearly decreasing inertia weight starting at 0.7 and ending at 0.4 was used. The maximum velocity (V_{max}) of each particle was set to be half the length of the search space in one dimension

The population size in the experiments was fixed to 20 particles in order to keep the computational requirements low [10]. The HPSOM has the additional parameter related to the mutation rate were set to 30%.

To extend the target of the comparison, Table 1 contains the results obtained using the standard Genetic Algorithm as it was presented in [5]. Crossover and mutation probabilities for each of the four test functions are listed in table 2. As shown at Table 1, the experiments of the four functions were done using dimension (30) and for iteration (2000).

5. Experimental Results

Table 1 list the test function, the search space, and the average best fitness with the standard error for the best particle found for the 100 runs.

Figures (1,2) presents the graphs that show the average best fitness for both the standard PSO and the HPSOM model. The graphs illustrate experiments with both a unimodal (Rosenbrock) and a multimodal test function (Griewank) both of 30 dimensions.

In these graphs in Figure 3, the PSO enters in stagnation while the HPSOM continues in the search of better position.

In experiments with the unimodal functions (f_1, f_2), the HPSOM achieved better results and had much faster convergence than the standard PSO. In the experiments with the multimodal functions (f_3, f_4), the Griewank function and the Rastrigin function, the HPSOM model had also a faster convergence than the standard PSO, and found the minimum value (zero).

The performance results shown at Table 1, shows that the HPSOM model is better than the standard PSO model. This performance is achieved by more exploring the search space caused by the numerical mutation operation [12].

Table 1: Results of average best fitness of 100 runs for dim. = 30 and Iter. = 2000

f	Search space	Std. PSO	Std. GA	HPSOM
f_1	$-100 \le x_i \le 100$	2.07E-014± 2.90E-014	0.00442± 1.78E-04	6.5764E-147± 5.6809E-146
f_2	$-100 \le x_i \le 100$	151.7675± 9.5624	199.730± 16.285	27.3682± 0.7146
f_3	$-600 \le x_i \le 600$	0.0189± 0.0586	889.537± 3.939	0.00± 0.00
$f4$	$-10 \le x_i \le 10$	49.4664± 0.6299	49.3212± 1.1204	0.00± 0.00

Table 2: Crossover and mutation probabilities used in std. GA.

Function	Crossover prob.	Mutation prob.
f_1	0.60	0.30
f_2	0.50	0.30
f_3	0.50	0.40
f_4	0.20	0.02

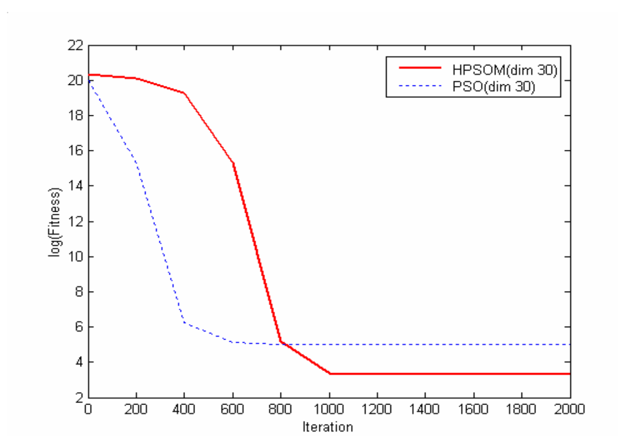

Figure1. PSO versus HPSOM model for f_2

Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06)
0-7695-2662-4/06 $20.00 © 2006 IEEE

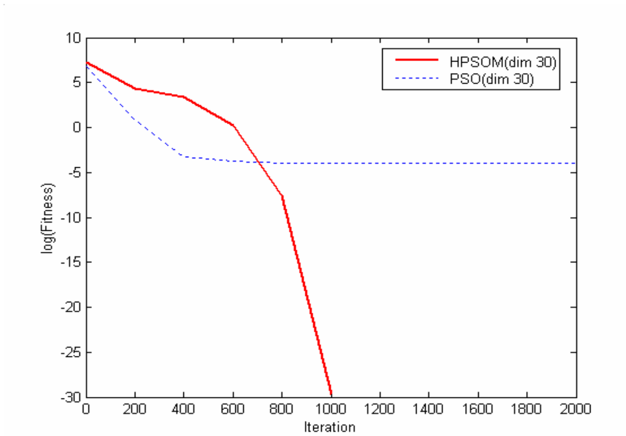

Figure 2. PSO versus HPSOM model for f_3

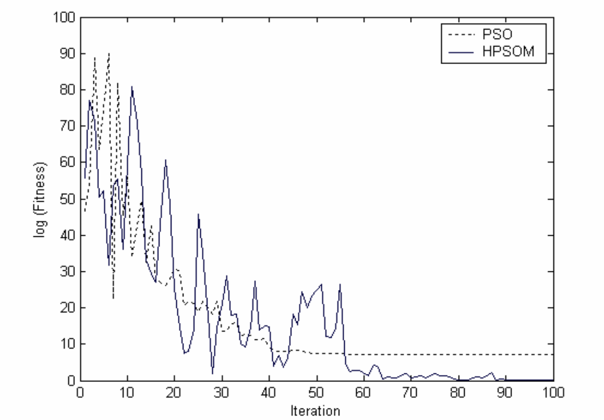

Figure 3. The behavior of the *gbest* particles in PSO/HPSOM for f_2

6. Conclusions

A new hybrid particle swarm algorithm (HPSOM) was introduced which exhibits properties of both the PSO algorithm and the AG algorithm. The main idea is to integrate PSO and GA mutation. Simulation was used to show that on the average the HPSOM algorithm performs better than the standard PSO algorithm in minimizing functions.

The hybrid algorithms make use of advantages of both GA and PSO methods therefore it is benefit in solving optimal problems. The optima found by the hybrid were better and present faster convergence.

Acknowledgment

The authors thank FAPEMIG, in Brazil, for the support of this work.

7. References

[I] J.H. Holland, Adaptation in Natural and Artificial System, The University of Michigan Press, Ann Arbor, 1975.

[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning. Reading, MA: Addison-Wesley, 1989.

[3] J. Kennedy and R.C. Eberhart - "Particle swarm optimization", proceeding of the 1995 IEEE International Conference on Neural Networks, vol. 4, 1942-1948. IEEE Press.

[4] P. J. Angeline, ”Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences”, Evolutionary Programming VII (1998), Lecture Notes in Computer.

[5] M. Løvbjerg, T. K. Rasmussen, and T. Krink, “Hybrid Particle Swarm Optimiser with Breeding and Subpopulations”, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), San Francisco, USA, July 2001.

[6] B. Awadh, N. Sepehri, O. Hawaleshka, A computer-aided process planning model based on genetic algorithms, Comput. Oper. Res. 22 (8) (1995) 841–856.

[7] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand, New York, 1991.

[8] J. Kennedy, ”Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance”, Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1931–1938. IEEE Press.

[9] P. N. Suganthan, ”Particle Swarm Optimiser with Neighbourhood Operator”, Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1958–1962. IEEE Press.

[10] Y. Shi and R. C. Eberhart, ”Parameter Selection in Particle Swarm Optimization”, Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 591–600. Springer.

[11] R. C. Eberhart and Y. Shi, ”Comparison between Genetic Algorithms and Particle Swarm Optimization”, Evolutionary Programming VII (1998), Lecture Notes in Computer Science 1447, 611–616. Springer.

[12] A. A. A. Esmin, G. Lambert-Torres, and A. C. Zambroni de Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Transactions on Power Systems,* vol. 20, no. 2, 859-866, 2005.