Comparative Study of Particle Swarm Optimization and Genetic Algorithms for Complex Mathematical functions

Fevrier Valdez, Patricia Melin

Abstract:

The Particle Swarm Optimization (PSO) and the Genetic Algorithms (GA) have been used successfully in solving problems of optimization with continuous and combinatorial search spaces. In this paper the results of the application of PSO and GAs for the optimization of mathematical functions are presented. These two methodologies have been implemented with the goal of making a comparison of their performance in solving complex optimization problems. This paper describes a comparison between a GA and PSO for the optimization of complex mathematical functions.

Keywords: *genetic algorithms, particle swarm optimization, hybrid systems, optimization.*

1. Introduction

We describe in this paper the application of a Genetic Algorithm (GA) [1] and Particle Swarm Optimization (PSO) [2] for the optimization of mathematical functions. In this case, we are using the Rastrigin's function, Rosenbrock's function, Ackley's function, Shubert's function, Sphere's function and Griewank's function [4] to compare the optimization results between a Genetic Algorithm and Particle Swarm Optimization.

2. Genetic Algorithm for Optimization

John Holland, from the University of Michigan began his work on genetic algorithms at the beginning of the 1960s. His first achievement was the publication of *Adaptation in Natural and Artificial System* [7] in 1975.

Holland had two goals in mind: to improve the understanding of natural adaptation process, and to design artificial systems having similar properties to natural systems [8].

The basic idea is as follows: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects. Only the association of different genomes can lead to the solution.

Holland's method is especially effective because it not only considers the role of mutation, but it also uses genetic recombination, (crossover) [9]. The crossover of partial solutions greatly improves the capability of the algorithm to approach, and eventually find, the optimal solution.

The essence of the GA in both theoretical and practical domains has been well demonstrated [1]. The concept of applying a GA to solve engineering problems is feasible and sound. However, despite the distinct advantages of a GA for solving complicated, constrained and multiobjective functions where other techniques may have failed, the full power of the GA in application is yet to be exploited [12].

To bring out the best use of the GA, we should explore further the study of genetic characteristics so that we can fully understand that the GA is not merely a unique technique for solving engineering problems, but that it also fulfils its potential for tackling scientific deadlocks that, in the past, were considered impossible to solve. In figure 1 we show the reproduction cycle of the Genetic Algorithm.

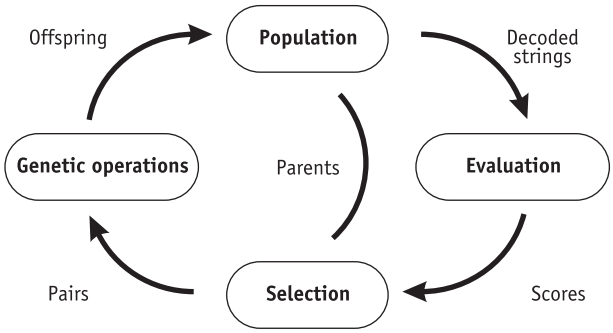

Fig. 1. The Reproduction cycle of a GA.

The Simple Genetic Algorithm can be expressed in pseudo code with the following cycle:

1. Generate the initial population of individuals aleatorily P(0).
2. While (number _ generations <= maximum _ numbers _ generations)
   Do:
   {
   Evaluation;
   Selection;
   Reproduction;
   Generation ++;
   }
3. Show results
4. End of the generation

2.1. Genetic operators

Once we have the genetic representation and the fitness function defined, the GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, and selection operators.

2.1.1. Initialization

Initially many individual solutions are randomly

generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated ran' domly, covering the entire range of possible solutions (the *search space*). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found [26].

2.1.2. Selection

During each successive epoch, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a *fitness-based* process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time consuming [26].

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

2.1.3. Reproduction

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

2.1.4. Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

A solution is found that satisfies minimum criteria.

Fixed number of generations reached.

Allocated budget (computation time/money) reached.

3. Biological Background

All living organisms consist of cells. In each *cell* there is the same set of *chromosomes*. Chromosomes are strings of DNA and serves as a model for the whole organism. A chromosome's characteristic is determined by the genes. Each gene has several forms or alternatives which are called *alleles*, producing differences in the set of characteristics associated with that gene. The set of chromosome is called the genotype, which defines a *phenotype* (the individual) with a certain fitness [26].

During reproduction, first occurs *recombination* (or *crossover*). Genes from parents form in some way the whole new chromosome. The new created *offspring* can then be mutated.

Mutation means, that the elements of DNA are a bit changed. This changes are mainly caused by errors in copying genes from parents. The *fitness* of an organism is measured by success of the organism in its life. According to Darwinian theory the highly fit individuals are given opportunities to “reproduce” whereas the least fit members of the population are less likely to get selected for reproduction, and so “die out”.

4. Particle Swarm Optimization

Team formation has been observed in many animal species [3]. For some animal species, teams, or groups, are controlled by a leader, for example a pride of lions, a troop of baboon, or a troop of wild buck, to name a few. One of the first studies of such animal societies is the work of Eugéne N. Marais in his studies of the wild chacma baboon in the early 1900s [5]. In these societies the behavior of individuals is strongly dictated by social hierarchy. More interesting is the self-organizing behavior of species living in groups where no leader can be identified, for example, a flock of birds, a school of fish, or a herd of sheep. Within these social groups individuals have no knowledge of the global behavior of the entire group, nor do they have any global information about the environment. Despite this, they have the ability to gather and move together, based on local interactions between individuals. From the simple, local interaction between individuals, more complex collective behavior emerges, such as flocking behavior, homing behavior, exploration and herding [6, 10, 11].

A large number of studies of the collective behavior of social animals have been done, for example,

- bird flocking behavior [11];
- fish schooling behavior [13, 14, 15];
- the hunting behavior of humpback whales [16];
- the foraging behavior of wild monkeys [17, 18]; and
- the courtship-like and foraging behavior of the basking shark [19, 20].

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [21], is based on a social-psychological model of social influence and social learning [22]. Individuals in a particle swarm (PS) follow a very simple behavior: emulate the success of neighboring individuals. The collective behavior that emerges is that of discovering optimal regions of high dimensional search space. This behavior is in accordance with the hypotheses of Wilson [23]. PSO has its origins in the work of the Reynolds [11]. A simplified social model has been developed, where the simple behaviors of determining nearest neighbors and velocity matching have been implemented. The initial intent of the simulation model

was to simulate the graceful, but unpredictable choreography of birds in a flock. The simulation model randomly initializes positions of birds on a torus pixel grid [24]. At each iteration, each individual determines its nearest neighbor and replaces its velocity with that of its neighbor. This simple behavior resulted in synchronous movement of the flock. However, the flock settled too quickly on an unanimous, unchanging flying direction. To solve this, a craziness component was introduced in the form of random adjustments to velocities.

To further expand the model, the rooster concept of Heppner and Grenander was added, in the form of a memory of previous best and neighborhood best positions, referred to as the cornfield. The personal best position of each individual is the best position found by that individual since the start of the simulation. The neighborhood best position is the best position found by the neighborhood of flocks mates. These two best positions proportionally to the distance from best positions, the flock clustered around the goal within a few iterations. This resulted even without velocity matching and craziness.

4.1. Basic Particle Swarm Optimization

Since its introduction in 1995 [24, 25] particle swarm optimization (PSO) has seen many improvements and applications. Most modifications to the basic PSO are directed towards improving convergence of the PSO and increasing the diversity of the swarm.

A PSO algorithm maintains a swarm of particles, where each particle represents a potential solution. In analogy with evolutionary computation paradigms, a swarm is a similar to a population, while a particle is similar to an individual. In simple terms, the particles are "flown" through a multidimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors. Let $x_i(t)$ denote the position of particle i in the search space at time step t; unless otherwise stated, t denotes discrete time steps. The position of the particle is changed by adding a velocity, $v_i(t)$, to the current position, i.e. see equation 1.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{1}$$

with $x_i(0) \sim \mathrm{U}(\mathrm{Xmin}, \mathrm{Xmax})$

It is a velocity vector that drives the optimization process, and reflects both the experiential knowledge of the particle and socially exchanged information from the particle's neighborhood. The experiential knowledge of a particle is generally referred to as the cognitive component, which is proportional to the distance of the particle from its own best position (referred to as the particle's personal best position) found since the first time step. The socially exchanged information is referred to as the social component of the velocity equation. See equation 1.

Originally, two PSO algorithms have been developed which differ in the size of their neighborhoods. These two algorithms, namely the "gbest" and "lbest", are considered in this paper.

4.2. Global Best PSO

For the global best PSO, or gbest PSO, the neighborhood for each particle is the entire swarm. The social network employed by the gbest PSO reflects the star topology. For the star topology, the social component of the particle velocity update reflects information obtained from all the particles in the swarm. In this case, the social information is the best position found by the swarm, referred to as $\hat{y}(t)$.

For gbest PSO, the velocity of particle i is calculated as (see equation 2)

(2)

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t)\ x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}(t)\ x_{ij}(t)]$$

where $v_{ij}(t)$ is the velocity of particle i in dimension $j = 1, \ldots, nx$ at time step t, $x_{ij}(t)$ is the position of particle i in dimension j at time step t, c_1 and c_2 are positive acceleration constants used to scale the contribution of the cognitive and social components respectively, and $r_{1j}(t)$, $r_{2j}(t) \sim U(0,1)$ are random values in the range $[0,1]$, sampled from a uniform distribution. This random values introduce a stochastic element to the algorithm.

The personal best position, y_i, associated with particle i is the best position the particle has visited since the first time step.

The gbest PSO is summarized in Algorithm 1. In this algorithm, the notation $S.xi$ is used to denote the position of particle i in swarm S.

Algorithm 2 lbest PSO

```
Create and initialize an nx-dimensional swarm, S;
    Repeat
        For each particle i = 1,...,S.ns do
            //set the personal best position
            If f(S.x_i) < f(S.y_i) then
                S.y_i = S.x_i;
            End
            //set the global best position if
            if f(S.y_i) < f(S.ŷ_i) then
                S.ŷ = S.y_i;
            End
        End
        For each particle i = 1,...,S.ns do
            Update the velocity using equation (1);
            Update the position using equation (2);
        End
Until stopping condition is true;
```

4.3. Local Best PSO

The local best PSO, or lbest PSO, using a ring social network topology where smaller neighborhoods are defined for each particle. The social component reflects information exchanged within the neighborhood of the particle, reflecting local knowledge of the environment. With reference to the velocity equation, the social contribution to particle velocity is proportional to the distance between a particle and the best position found by the neighborhood of particles. The velocity is calculated as (see equation 3)

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] \quad (3)$$

where $\hat{y}_{ij}$ is the best position, found by the neighbor hood of particle i in dimension j.

Algorithm 2 lbest PSO

Create and initialize an nx-dimensional swarm, S;
  Repeat
    For each particle $i = 1,\ldots,S.ns$ do
      //set the personal best position
      If $f(S.x_i) < f(S.y_i)$ then
        $S.y_i = S.x_i$;
      End
      //set the neighborhood best position if
      if $f(S.y_i) < f(S.\hat{y}_i)$ then
        $S.\hat{y} = S.y_i$;
      End
    End
    For each particle $i = 1,\ldots,S.ns$ do
      Update the velocity using equation (3);
      Update the position using equation (1);
    End
Until stopping condition is true.

5. Mathematical Functions

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. If we compare two searching algorithms with all possible functions, the performance of any two algorithms will be, on average, the same. As a result, attempting to design a perfect test set where all the functions are present in order to determine whether an algorithm is better than another for every function. The reason why, when an algorithm is evaluated, we must look for the kind of problems where its performance is good, in order to characterize the type of problems for which the algorithm is suitable. In this way, we have made a previous study of the functions to be optimized for constructing a test set with six benchmark functions and a better selection. This allows us to obtain conclusions of the performance of the algorithm depending on the type of function. The mathematical functions analyzed in this paper are the Rastrigin's function Rosenbrock's function, Ackley's function, Shubert's function, Sphere's function and Griewank's function. All the functions were evaluated considering 2 variables.

5.1. Rastrigin's Function

- Number of variables: 2 variables.
- Definition: See (equation 4)

$$f(x) = 10n + \sum_{i=1}^{n} (x^2 - 10\cos(2\pi x_i)) \quad (4)$$

- Number of local minima: several local minima.
- The global minima: $x^* = (0,\ldots,0), f(x^*) = 0.$
- Function graph: for $n = 2$. See (Fig. 2)

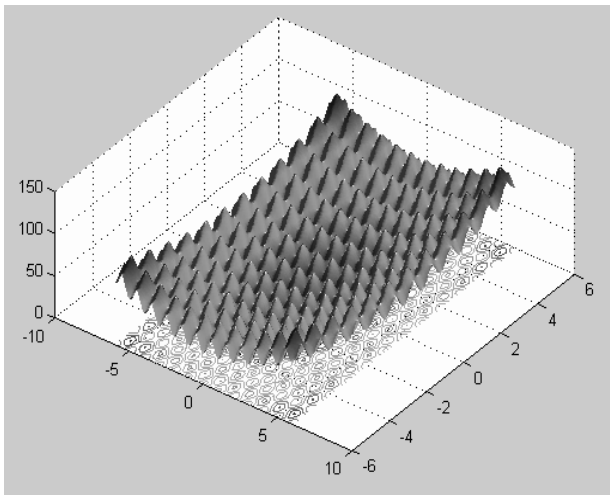

Fig. 2. Plot of Rastrigin's function.

5.2. Rosenbrock's Function

- Number of variables: 2 variables.
- Definition: See (equation 5)

$$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_i^2 + 1)^2 + (x_i - 1)^2] \quad (5)$$

- Number of local minima: several local minima.
- The global minima: $x^* = (1,\ldots,1), f(x^*) = 0.$
- Function graph: for $n = 2$. See (Fig. 3)

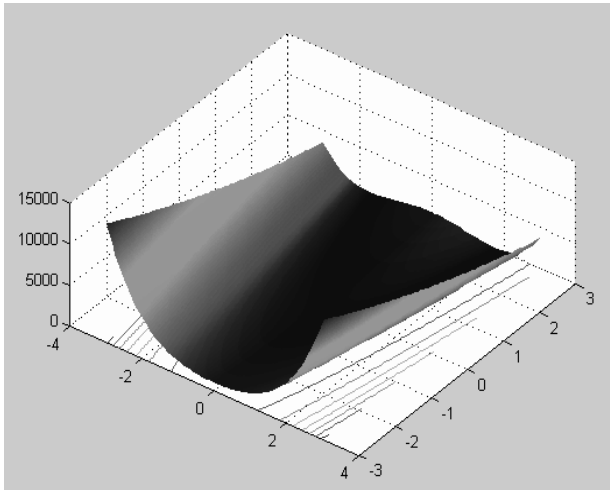

Fig. 3. Plot of Rosenbrock's function.

5.3. Ackley's Function

- Number of variables: 2 variables.
- Definition: See (equation 6)

$$f(x) = 20 + e - 20e^{-1/5}\sqrt{1/n\sum_{i=1}^{n} x_i^2} - \quad (6)$$
$$- e^{1/n}\sum_{i=1}^{n} \cos(2\pi x_1)$$

- Number of local minima: several local minima.
- The global minima: $x^* = (1,\ldots,1), f(x^*) = 0.$
- Function graph: for $n = 2$. See (Fig. 4)

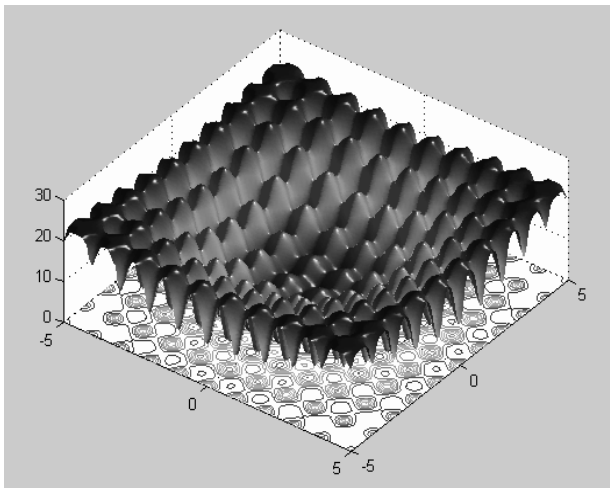

Fig.4. Plot of Ackley's function.

5.4. Shubert's Function

- Number of variables: $n = 2$.
- Definition: See (equation 7)

$$f(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right) \tag{7}$$

- Number of local minima: several local minima.
- The global minima: 18 global minima, $f(x^*) = -186.7309$.
- Function graph: for $n = 2$. See (Fig. 5)

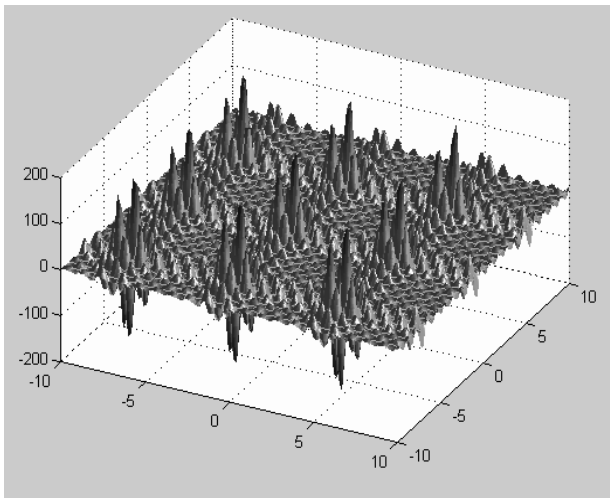

Fig.5. Plot of Shubert's function.

5.5. Sphere's Function

- Number of variables: 2 variables.
- Definition: See (equation 8)

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{8}$$

- Search domain: $-5.12 \le x_i \le 5.12, i = 1, 2, \ldots, n$.
- Number of local minima: no local minimum except the global one.
- The global minima: $x^* = (0, \ldots, 0), f(x^*) = 0$
- Function graph: for $n = 2$. See (Fig. 6)

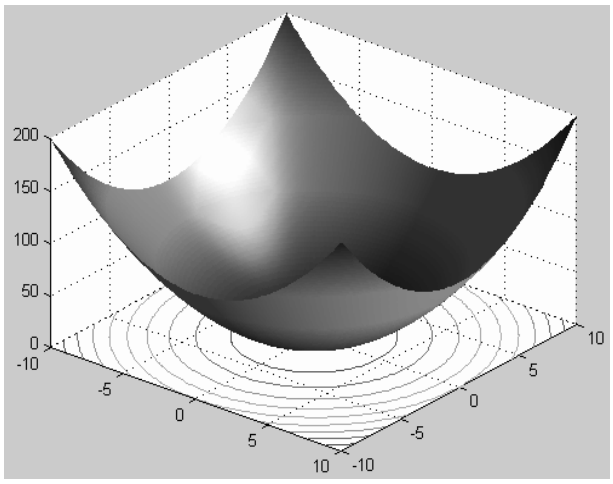

Fig.6. Plot of Sphere's function.

5.6. Griewank's Function

- Number of variables: 2 variables.
- Definition: See (equation 9)

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(x_i / \sqrt{i}) + 1 \tag{9}$$

- Search domain: $-600 \le x_i \le 600, i = 1, 2, \ldots, n$.
- Number of local minima: several local minima.
- The global minima: $x^* = (0, \ldots, 0), f(x^*) = 0$
- Function graph: for $n = 2$. See (Fig. 7)

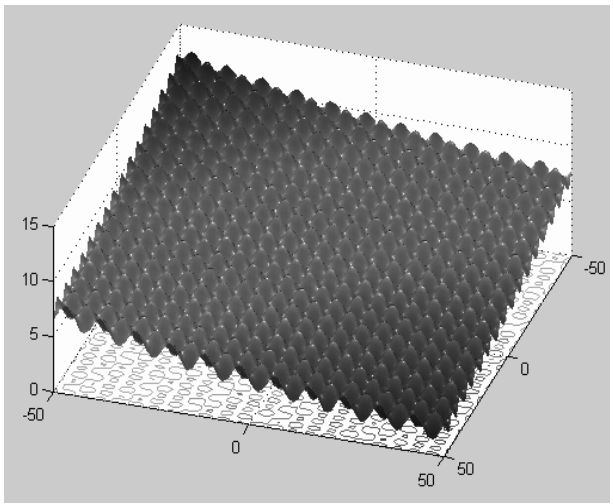

Fig. 7. Plot of Griewank's function.

6. Simulation Results

Several tests of the PSO and GA algorithms were made in the Matlab programming language. All the implementations were developed using a computer with processor AMD turion X2 of 64 bits that works to a frequency of clock of 1800 MHz, 2 GB of RAM Memory and Windows Vista Ultimate operating system.

6.1. Experimental Results with the Genetic Algorithm (GA)

The results obtained after applying the genetic algo-

rithm to the mathematical functions are shown on tables 1, 2, 3, 4, 5 and 6:

Parameters of Tables 1, 2, 3, 4, 5 and 6:
No. = Number of experiment
TEST = Number of times that the Genetic Algorithm was executed with the same parameters
GEN = Generations number
POP = Population size
CROSS =Crossover type and % crossover
MUT = Mutation type and % mutation
BEST = Best Fitness Value
MEAN = Mean of 50 tests

6.1.1. Results obtained after applying the genetic algorithm to the Rastrigin's function

From Table 1 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 7.36E-07, which is the shaded value in Table 1 (Experiment 1). The best average objective value was 2.15 E-03 obtained in Experiment 1.

Table 1. Results obtained after applying the genetic algorithm to the Rastrigin's function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
100	100	scat	80	gauss	2	Rou	7.36E-07	2.15E-03
100	70	scat	50	gauss	10	Rou	8.71E-04	3.41E-02
150	150	scat	90	gauss	9	Rou	4.44E-05	7.00E-03
80	40	two point	90	gauss	5	Rou	2.05E-04	8.53E-02
80	40	scat	25	gauss	10	Rou	2.49E-04	3.10E-01

6.1.2. Results obtained after applying the genetic algorithm to the Rosenbrock's function.

From Table 2 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 2.35E-07, which is the shaded value in Table 2 (Experiment 1). The best average objective value was 1.02 E-05 obtained in Experiment 1.

Table 2. Results obtained after applying the genetic algorithm to the Rosenbrock's function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
200	150	scat	50	gauss	1	Rou	2.35E-07	1.02E-05
80	40	scat	25	gauss	10	Rou	6.03E-04	2.96E-02
150	90	scat	90	gauss	6	Rou	6.35E-04	1.61E-02
150	100	scat	65	gauss	8	Rou	3.38E-06	2.98E-02

6.1.3. Results obtained after applying the genetic algorithm to the Ackley function.

From Table 3 it can be appreciated that after executing the GA 50 times, in 5 cases the value closest to the global minimum was 2.98 and the genetic algorithm were not able to find the global minimum for the Ackley function.

Table 3. Results obtained after applying the genetic algorithm to the Ackley's function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
100	100	scat	80	gauss	2	Rou	2.981082489	2.980868171
150	100	scat	65	gauss	8	Rou	2.981169	2.994222
100	120	scat	70	gauss	8	Rou	2.980857	2.986018
100	90	scat	80	gauss	5	Rou	2.985482	2.986464
100	90	scat	30	gauss	9	Rou	2.989068	3.028187

6.1.4. Results obtained after applying the genetic algorithm to the Shubert's function.

From Table 4 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at -186.7303, which is the shaded value in Table 4 (Experiment 2). The best average objective value was 186.668 obtained in Experiment 3.

Table 4. Results obtained after applying the genetic algorithm to the Shubert function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
100	90	scat	30	gauss	9	Rou	-186.725467	-186.026239
120	120	scat	75	gauss	9	Rou	-186.7303482	-186.560965
120	200	scat	75	gauss	9	Rou	-186.27423	-186.668163
120	100	scat	95	gauss	6	Rou	-186.726104	-184.202502
80	60	scat	95	gauss	8	Rou	-186.663188	-178.940276

6.1.5. Results obtained after applying the genetic algorithm to the Sphere's function.

From Table 5 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 3.49E-07, which is the shaded value in Table 5 (Experiment 4). The best average objective value was 1.62E-04 obtained in Experiment 1.

Table 5. Results obtained after applying the genetic algorithm to the Sphere's function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
100	20	Scat	80	gauss	1	Rou	1.02E-05	1.62E-04
50	40	Scat	50	gauss	9	Rou	0.05831066	6.97E-03
80	50	two point	72	gauss	20	Rou	8.80E-07	1.26E-03
80	70	two point	84	gauss	9	Rou	3.49E-07	6.44E-04
100	30	Scat	74	gauss	10	Rou	6.44E-04	4.03E-03

6.1.6. Results obtained after applying the genetic algorithm to the Griewank function.

From Table 6 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 1.84E-07, which is the shaded value in Table 6 (Experiment 5). The best average objective value was 2.552E-05 obtained in Experiment 5.

Table 6. Results obtained after applying the genetic algorithm to the Griewank's function.

GEN	POP	CROSS	%CROSS	MUT	%MUT	SEL	BEST	MEAN
80	30	Scat	74	gauss	10	Rou	5.35E-04	0.06094737
100	80	Scat	90	gauss	6	Rou	0.09573318	0.47031062
100	100	two point	30	gauss	7	Rou	3.89E-05	0.00307528
80	70	two point	84	gauss	7	Rou	4.78E-04	0.15430235
80	80	Scat	90	gauss	10	Rou	1.84E-07	2.552E-05

6.2. Experimental Results with the Particle Swarm Optimization (PSO)

The results obtained after applying the particle swarm optimization to the mathematical functions are shown on tables 7, 8, 9, 10, 11 and 12:

Parameters of Tables 7, 8, 9, 10, 11 and 12:
BEST = Best fitness value
DIM = Dimensions
POP = Population size
MEAN = Mean of 50 tests

6.2.1. Results obtained after applying the particle swarm optimization to the Rastrigin's function

From Table 7 it can be appreciated that after executing the PSO 50 times, only in 3 cases the global minimum was achieved with the best objective value at 3.48E-05, which is the shaded value in Table 7 (Experiment 2). The best average objective value was 2.87 obtained in Experiment 1.

Table 7. Results obtained after applying the PSO to the Rastrigin's function.

POP	DIM	BEST	MEAN
80	10	2.27E-03	2.87E+00
20	10	3.48E-05	5.47E+00
80	10	9.95E-01	1.29E+01
20	20	9.00E+00	1.29E+01
40	20	8.96E+00	5.07E+01

6.2.2. Results obtained after applying the particle swarm optimization to the Rosenbrock's function

From Table 8 it can be appreciated that after executing the PSO 50 times, only in 5 cases the global minimum was achieved with the best objective value at 2.46E-03, which is the shaded value in Table 8 (Experiment 3). The best average objective value was 1.97E+01 obtained in Experiment 3.

Table 8. Results obtained after applying the PSO to the Rosenbrock's function.

POP	DIM	BEST	MEAN
20	10	5.19E-02	7.41E+01
80	10	6.12E-02	2.40E+01
40	10	2.46E-03	1.97E+01
80	10	6.02E-03	2.59E+01
80	30	2.97E+00	1.26E+02

6.2.3. Results obtained after applying the particle swarm optimization to the Ackley's function

From Table 9 it can be appreciated that after executing the PSO 50 times, in 5 cases the value closest to the global minimum was 2.98 and the particle swarm optimization were not able to find the global minimum for the Ackley function.

Table 9. Results obtained after applying the PSO to the Ackley's function.

POP	DIM	BEST	MEAN
10	2	2.980500	2.980500
6	1	2.980499	4.56
50	1	2.980500	2.980500
30	1	2.980500	2.980500
40	2	2.980500	2.980500

6.2.4. Results obtained after applying the particle swarm optimization to the Shubert's function.

From Table 10 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at -186.7309. The best average objective value was -186.7309.

Table 10. Results obtained after applying the PSO to the Shubert's function.

POP	DIM	BEST	MEAN
20	10	-186.7308	-186.7308
40	10	-186.7309	-186.7309
80	10	-186.7309	-186.7309
80	20	-186.7309	-186.7309
80	30	-186.7309	-186.7309

6.2.5. Results obtained after applying the particle swarm optimization to the Sphere's function

From Table 11 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 3.89E-11, which is the shaded value in Table 9 (Experiment 4). The best average objective value was 8.26E-11.

Table 11. Results obtained after applying the PSO to the Sphere's function.

POP	DIM	BEST	MEAN
20	10	4.88E-11	8.26E-11
40	10	4.12E-11	8.51E-11
80	10	4.46E-11	8.29E-11
20	20	3.89E-11	6.86E-10
40	20	6.31E-11	9.11E-11

6.2.6. Results obtained after applying the particle swarm optimization to the Griewank's function

From Table 12 it can be appreciated that after executing the GA 50 times, only in 5 cases the global minimum was achieved with the best objective value at 9.77E-11, which is the shaded value in Table 12 (Experiment 5). The best average objective value was 2.56E-02.

Table 12. Results obtained after applying the PSO to the Griewank's function.

POP	DIM	BEST	MEAN
40	10	3.20E-02	1.06E-01
20	10	9.86E-03	8.82E-02
80	10	1.23E-02	7.58E-02
20	20	3.69E-08	2.77E-02
40	20	9.77E-11	2.56E-02

7. Conclusions

The analysis of the simulation results of the two evolutionary methods considered in this paper, in this case the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO), lead us to the conclusion that for the problems of optimization of these 6 mathematical functions, in all cases one can say that the two proposed methods work correctly and they can be applied for this type of problems.

After studying the two methods of evolutionary computing (GA and PSO), we reach the conclusion that for the optimization of these 6 mathematical functions, GA and PSO evolved in a similar form, achieving both methods the optimization of 5 of the 6 proposed functions, with values very similar and near the objectives. Also it is possible to observe that even if the GA as the PSO did not achieve the optimization of the Ackley's function, this may have happened because they were trapped in local minima.

Figure 8 shows the comparison of the results obtained for these 6 test functions and it can be appreciated that the values that were taken from the tables of results above mentioned, the GA and the PSO obtained very good results and was very little the difference between of them. Table 13 shows the values corresponding to figure 8.

The advantage to use PSO is that there are few parameters used for the implementation. The genetic algorithm uses more parameters for its implementation.

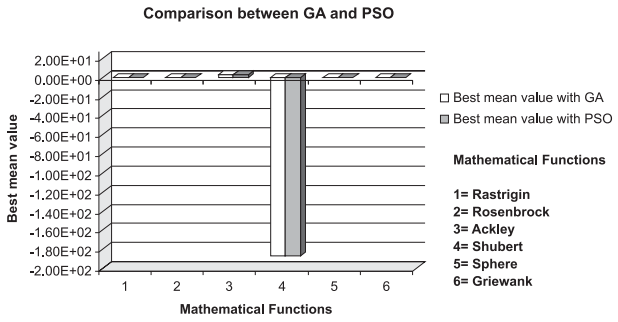

Fig. 8. Comparison between GA and PSO.

From Table 13 it can be appreciated that after execu' ting the GA and PSO, the comparison of the results obtained between GA and PSO for the optimization of the 6 proposed mathematical functions of this paper. The table shows the results for the figure 8, it can be appreciated in some cases, the GA was better than the PSO, for example, for the Rastrigin's function, Rosenbrock's function and Griewank's function. In other cases, the PSO was better than the GA, for example, the Sphere's function.

Table 13. Comparison between GA and PSO.

Mathematical Functions	GA	PSO	Objective Value	Number of variables
Rastrigin	7.36E-07	3.48E-05	0	2
Rosenbrock	2.35E-07	2.46E-03	0	2
Ackley	2.98108249	2.9805	0	2
Shubert	-186.730348	-186.7309	-186.73	2
Sphere	1.62E-04	8.26E-11	0	2
Griewank	2.552E-05	2.56E-02	0	2

AUTHORS
Fevrier Valdez* - PhD Student of Computer Science in the Universidad Autónoma de Baja California, Tijuana, B.C., México. E-mail: fevrier@tectijuana.mx.
Patricia Melin - Computer Science in the Graduate Division, Tijuana Institute of Technology Tijuana, B.C. E-mail: pmelin@tectijuana.mx.
* Corresponding author.

References

[1] K.F. Man, K.S. Tang, and S. Kwong, "Genetic Algorithms: Concepts and Designs", *Springer Verlag*, 1999.
[2] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory". In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan. 1995, pp. 39-43.
[3] C. Anderson, N.R. Franks, Teams in Animal Societies, *Behavioral Ecology*, vol. 12, no. 5, 2001, pp.534-540.
[4] *Matlab Toolbox*. Available at: www.mathworks .com
[5] E.N. Marais, *The Soul of the Ape*, Human & Rousseau Publishers, 1969.
[6] O.B. Bayazit, J-M. Lien, and N.M. Amato, "Roadmap-Based Flocking for Complex Environments". In: *Proceedings of the Tenth Pacific Conference on Computer Sciencie and Applications*, 2002, pp. 104-113.
[7] J.H. Holland, *Adaptation in natural and artificial system*, Ann Arbor, The University of Michigan Press, 1975.
[8] D. Goldberg, *Genetic Algorithms*, Addison Wesley, 1988.
[9] C. Emmeche, *Garden in the Machine. The Emerging Science of Artificial Life*, Princeton University Press, 1994, p. 114.
[10] M.J. Mataric, *Interaction and Intelligent Behavior*. PhD thesis, Department of Electrical, Electronic and Computer Engineering, MIT, 1994.
[11] C.W. Reynolds, Flocks, Herds, and Schools, "A Distributed Behavioral Model", *Computer Graphics*, vol. 21, no. 4, 1987, pp. 25-34.
[12] T. Back, D. B. Fogel, and Z. Michalewicz, (Eds), *Handbook of Evolutionary Computation*, Oxford University Press, 1997.
[13] B.L. Partridge, "The Structure and Function of Fish Schools", *Scientific American*, no. 246, 1982, pp. 114-123.
[14] T.J. Pitcher, B.L. Partridge, and C.S. Wardle. "Blind Fish Can School", *Science*, vol. 194, 1976, pp. 963-965.

[15] E. Shaw, "Schooling in Fishes: Critique and Review". In: *Development and Evolution of Behavior*, W.H. Freeman, 1970, pages 452-480.

[16] F.A. Sharpe, *Social Foraging of Southeast Alaskan Humback Whales*, PhD thesis, Simon Fraser University, Burnaby, British Columbia, 2000.

[17] C.H. Janson, "Experimental Evidence for Spatial Memory in Foraging Wild Capuchin Monkeys, Cebus Apella", *Animal Behaviour*, vol. 55, 1998, pp. 1229-12243.

[18] C.R. Menzel, "Cognitive Aspects of Foraging in Japanese Monkeys", *Animal Behaviour*, vol. 41, 1991, pp. 397-402.

[19] D.W. Sims and V.A. Quayle, "Selective Foraging Behaviour of Basking Sharks on Zooplankton in a Small-Scale Front", *Nature*, no. 393, 1998, pp. 460-464.

[20] D.W. Sims, E.J. Southall, V.A. Quayle, and A.M. Fox, "Annual Social Behaviour of Basking Sharks Associated with Coastal Front Areas". In: *Proceedings of the Royal Society of London*, vol. 267, 2000, pages 1897-1904.

[21] J. Kennedy and R.C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm". In: *Proceedings of the World Multiconference on Systematics, Cybernetics and Informatics*, 1997, pages 4104-4109.

[22] J. Kennedy and R. Mendes, "Neighborhood Topologies in Fully-Informed and Best-of-Neighborhood Particles Swarms". In: *Proceedings of the IEEE International Workshop on Soft Computing in Industrial Applications*, June 2003, pages 45-50.

[23] E.O. Wilson, Sociobiology. The New Synthesis, Belknap Press, 1975.

[24] J. Kennedy and R.C. Heberhart, "Particle Swarm Optimization". In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, IEEE Press, 1995, pages 1942-1948.

[25] R.C. Eberhart and J. Kennedy, "A new Optimizer using Particle Swarm Theory". In: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, 1995, pages 39-43.

[26] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press. 1998.