Studying Solutions of Traveling Salesman Problem with Hybrid Particle Swarm Optimization

HELGA G. MARTINS, MATEUS BARROS, BRUNO TONSIC DE ARAUJO, RENATO Y. BO, LEANDRO FALEIROS, AND GERMANO LAMBERT-TORRES
Federal University of Itajuba - UNIFEI
Av. BPS 1303 – Itajuba – MG – 37500-503
BRAZIL

Abstract: - This paper presents the application of the PSO (Particle Swarm Optimization) as a tool to solve the Traveling Salesman Problem. The proposed application consists of using a technique based on Artificial Intelligent – AI. It is based in the simulation of social interaction between individuals of a population. Here, each element of population moves in the hyper-space and they are attracted by positions that they considers promising.

Key-Words: - Artificial Intelligence, Swarm Intelligence, Traveling Salesman Problem, Optimization

1 Introduction

The desire of building intelligent machines able to relieve human in different tasks stimulates researches in lots of areas. With the advent of computers, machines are becoming able to simulate the brain performance. Like this it was created the science named Artificial Intelligence (AI). AI is a science that cares about creating machine algorithms that have characteristics usually associated with human intelligence, like language understanding, apprenticeship, reasoning based in induction trials, deduction and choice between alternatives of solutions for a determined problem.

Beside this, in the last years, an expressive number of researchers in computation area are utilizing natural mechanisms and behavior as inspiration to solve problems in different areas [1]. This kind o resolution is called Non Algorithmic programming. The use of this programming is broadly applied in problems that present a large number of variables, like problems with high complexity to be developed [2].

This paper presents the application of the Particle Swarm Optimization (PSO). One function optimization method that was created based on biological science. It's one of the most recent topics in the field of Evolutionary Algorithms (EA). The PSO method is very recent in the literature [3].

2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based optimization method first proposed by Kennedy and Eberhart in 1995 [4]. The first researches that lead to the theory about particle swarm where realized by scientists like: ornithologists, biologists, psychologists. In these areas the term Swarm Intelligence is well known and is characterized by the possibility of a large amount of individuals realizing complexes tasks. Some simulations of the swarm were lately abstracted in the mathematical field. The use of the swarm to solve simple natural tasks became in an intriguing idea in optimizing algorithms and functions.

The idea of swarm of particles appears in nature, lots of species behave like an intelligent swarm. In these situations a large amount of individuals solve complexes tasks. Knowing some of these examples, it is able to realize a profound study in the swarm theory [4-8].

The swarm intelligence is an iterative procedure keeps a population structure called individuals, which represent possible solutions for a determinate problem. This technique consists in a recently developed method based in evolutionary computation with stochastic techniques; which model comes from the observation of the collective behavior in populations found in nature.

Considering the biological system as one, it is possible to see that them during their evolution, developed behaviors of adaptation that enabled their survival. One of these interesting techniques consists in the interaction of the individuals in a population that leads to what is best for the whole group, for example, the bird's formation when they have to fly a large distance.

Based in this analogy with the process of biological adaptation of species, the swarm optimization finds the best solution using population of particles, where each particle represents one possible solution to the problem, developing the idea of interaction, comparing information's that each particle assumes in each instant of the analysis. So, we can see that swarm intelligence is used in optimization problems.

Between factors that make swarm intelligence one well succeeded are:

- Simple operation;
- It can be used in situations where the mathematical model isn't known, or the model is inexact and also in linear and non-linear functions.

3 Methodology

In this section the abstraction process is described as well as the most important factors in PSO.

The animals are abstracted into software elements, which occupy a position in n-dimensional search space, having a velocity and fitness at their position considering the problem.

Thus the roosting areas can be seen as the local optima in the problem that should be solver and the global optimum is a local optimum in its environment, too.

The particles need some information about their environment for to do a convergence towards the global optimum. For this the ratio of exploration needs to be adjusted or adjusting of velocity.

3.1 Algorithm and Definitions

In the algorithm each individual particle i has the properties: a current position, x_i, a current speed, vi, and a best local position y_i in the search space [9].

The best local position, y_i, correspond the particle i position in the search space, which presents the smallest error. This error is determined by objective function f and has minimization tasks.

The best global position denoted by g represents the position that produces the smallest error between all y_i.

The strength of a collective structure like a swarm lies in its social qualities, meaning that an individual can utilize knowledge that has been acquired by the swarm to improve its own position (L. Schoeman, 2005).

The local and global values are showed in the equations (1) and (2). The equations define how the best values are updated at time t, respectively [9].

Considering that the swarm consists of s particles, thus *i* varies *(1, . . ., s)*.

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

$$g(t) \in \{y_0(t), y_1(t), \ldots, y_s(t)\} \quad | \quad f(g(t)) = \min\{f(y_0(t)),\ f(y_1(t)), \ldots,\ f(y_s(t))\} \quad (2)$$

At each interaction all particles are updated utilizing the equations (3) and (4). For all dimensions *j* varies *(1,. . . , n)*. Thus $x_{i,\,j}$, $y_{i,\,j}$ and $v_{i,\,j}$ are the current position, current local best position and velocity of the *j*th dimension of the *i*th particle, respectively.

The adjusting the velocity is presented as follow:

$$v_{i,j}(t+1) = W.v_{i,j}(t) + C_1.r_1.[y_{i,j}(t) - x(t)] + + C_2.r_2.[g_j(t) - x(t)] \quad (3)$$

The *Inertia Weight*, *W*, is used in this equation and it is to control the convergence behavior of the PSO. When the *Inertia Weight* V*alues* are small then the convergence of the result on a sub optimal position is usually more rapid, while the Inertia Weight Values are large it may prevent divergence.

The *Acceleration Coefficients* presented in the equation (3), C_1 and C_2, control how far a particle will move in a single iteration. The coefficients are typically both set to a value of 2.0, although it has been shown that setting $C_1 \neq C_2$ can lead to a good performance [9].

After this step, in order to obtain the new position of the particle, the new velocity is added to the current position of the particle:

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

Notices that for each dimension these equations must be initiated again, using it as random effect of natural.

3.2 Hybrid Swarm Intelligence with Mutation

In order to correct the problem of premature convergence at the software caused by particles stagnation, this paper suggests the incorporation of the mutation used in Genetics Algorithms - GA.

This process consists in search the solution in different regions when the solution converges to an excellent place.

The adopted process for this is based on the draw of a random number in a known interval. In this interval is chosen the value of the mutation tax and if the raffled number is inside this predefined tax a new random route will be formed and the routine starts again [9]. Figure 1 shows the basic algorithm.

```
IF (tax = Mutation tax)
        *Particle Random Choice;
        *Update Search Space;
        *Search solution according
            to Objective Function (f).
ELSE
      Premature Convergence.
END
```

Figure 1. Basic Mutation Algorithm.

3.3 Visual Basic.NET

The Visual Basic is a Microsoft's programming software and is integrant part of the package Microsoft Visual Studio. The last version of the Visual Basic is part of the Visual package Studio .NET, directed toward .Net applications and the previous version was part of the Microsoft Visual Studio 6.0, still used.

The improving of the Visual Basic is the language directed by events (event driven) and has an integrated development environment (IDE - Integrated Development Environment) totally graphical. This helps in the construction of the interface of the applications (GUI - Graphical User Interface) named "Visual".

The Microsoft Visual Studio .NET, is a development tool appeared in the end of 2001. Its sprouting revolutionized the way of programming, mainly for the Web and mobile devices, for being a tool that facilitated what was more complex to make before.

3.4 Performing Traveling Salesman Problem (TSP)

The Traveling Salesman Problem can be shaped as a minimization function of the each presented (candidate) route distances.

This paper presents an optional pattern for creation of a graphical interface, contend sixteen cities and four candidates to solution (four particles). The routes presented by each solution are measured by Test Function and they are kept for to be able to previously start the patterns of the swarm intelligence through the presented equations. From the each calculated route distances is defines current position, current speed, best personal position and best global position.

Notice that the four first candidates to solution are gotten in random way, that is, the position that each particle will assume and its route are consequently gotten through a random draw of sixteen values that vary between zero and one and after that it is made its increasing in sequence. This order defines which are the departure city and the order of the established route consequently.

This paper presents to stop criterion, the number of iterations that software will make, therefore this value is chosen through the user need.

That the values established for each constant are gotten in empirical way, therefore they are variable for each application.

This paper used the following functions of test:

- Minor route.
- Major route.
- Major number of crossings with axle x.
- Minor number of crossings with axle x.
- Major number of crossings with axle y.
- Minor number of crossings with axle y.

4 Experimental Results

When the swarm algorithm is applied into the Traveling Salesman Problem, the component "current position" (xi) is defined as the route that the particle is assumes in that moment. It means that in the each iteration of the particle takes different routes. The positions are measured by the test functions (f), following two criterias (depending on the software operation mode):

- Route distance;
- Number of crossing trough the axis x=0 and y=0.

The first criterion acts when the software needs to find the route which has the *minimum* and the *maximum length*. The second criterion acts when the software searches for the *maximum* and the *minimum number of crossings* trough the axis x=0 (horizontal line) and *the minimum number of crossings* trough the axis y=0 (vertical line).

In each iteration of the software one of these functions is applied to each particle of the system, and when convenient, the route in the current iteration becomes the component "local best" ($\mathbf{y}_i$).

In example, if the Test Function searches for the route with shorter length and a particle (route $\mathbf{x}_i$) assumes a shorter value after iteration that the anterior local best, the particle in this iteration become the new local best (**yi**).

The component "global best" (**g**) is defined as the value that is more adequate to the Test Function between the local bests of all the particles. The term velocity refers to the iterative process, so it's responsible to balance the variables in the problem and lead them to a common sense.

After a reasonable number of iterations, the software resulted in considerate as satisfactory. Some results are shown bellow.

Figure 2 is presented the developed graphical interface.

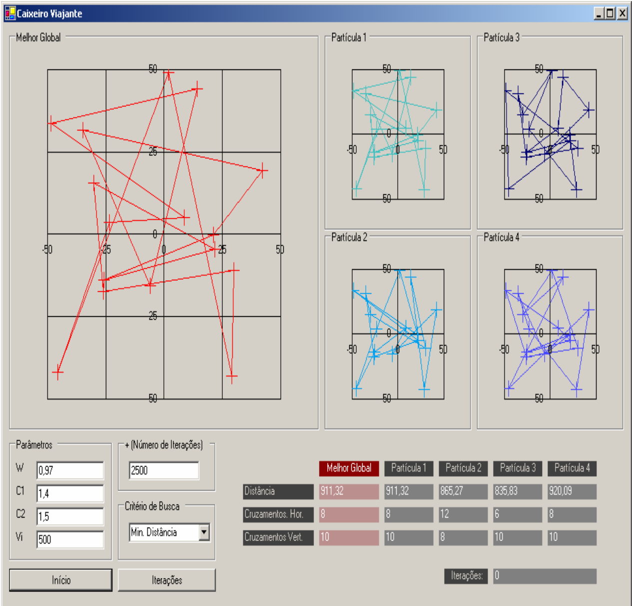

Figure 2. Developed graphical interface.

The software used to solve the Traveling Salesman Problem (TSP) presented answers with satisfactory values after a reasonable number of iterations. This occurred because the use of iterations techniques owed to consider like "stop criterion" an error tax.

The result referring a shorter length is showed in Figure 3.

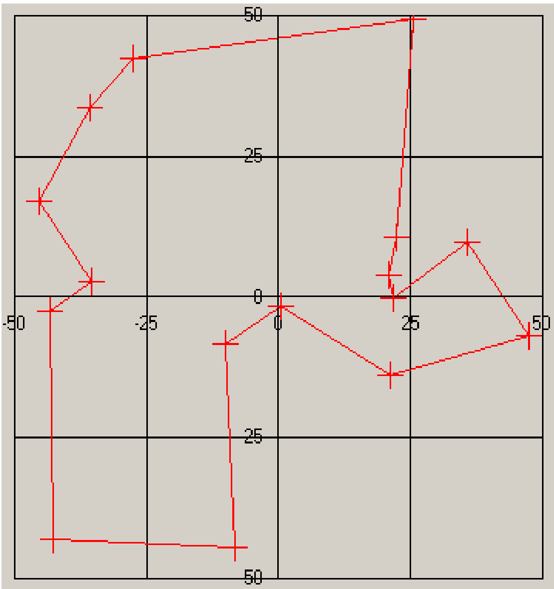

Figure 3. Results referring a shorter length.

The result referring a larger length is showed in Figure 4.

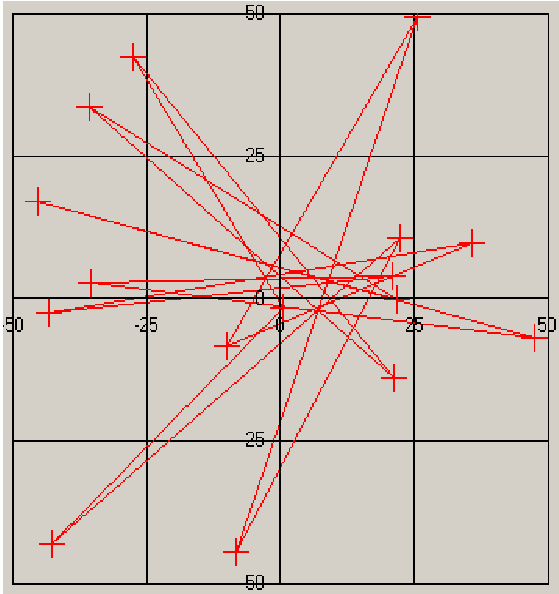

Figure 4. Results referring a larger length.

The result referring a larger number of crossings with the axe x is showed in Figure 5.

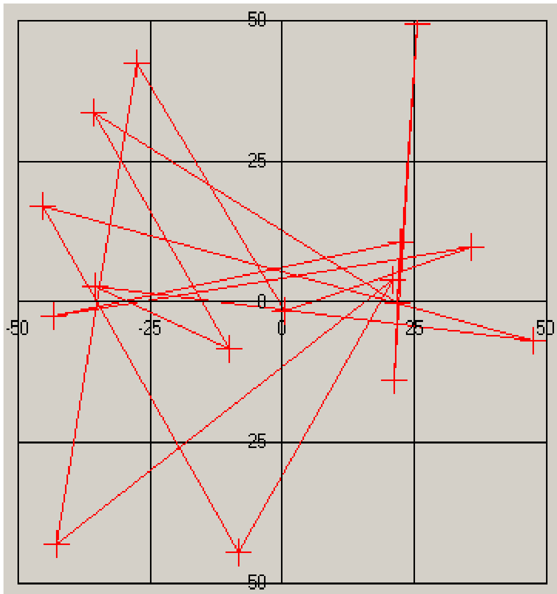

Figure 5. Results referring a larger number of crossings with axe x.

The result referring a larger number of crossings with the axe y is showed in Figure 6.

The results obtained by Hybrid System Intelligence with Mutation are better than ones by using traditional PSO.

The incorporation of mutation processes often used in Genetic Algorithm was extremely important to the solution development. This hindered the premature convergence.

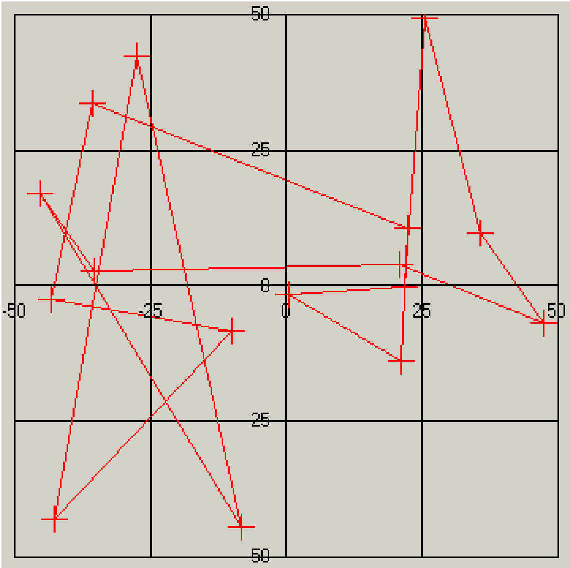

Figure 6. Results referring a shorter number of crossings with axe y.

5 Conclusion and Future Work

In this article an answer for the Traveling Salesman Problem - TSP through the technical Particle Swarm Optimization - PSO is presented, which technique belongs to the field of Evolutionary Algorithms - EA.

The Traveling Salesman Problem is a problem of optimization admits many answers however the end products induced by the developed software are altogether coherent and corrects. This technique is better comparing than others techniques of Evolutionary Algorithms (EA) because the updating of information will occur only if the current date is better than the storage date memory then this technique work with memory.

The development of a hybrid system, in other words, the incorporation of processes often used in Genetic Algorithm, was extremely important to the solution development, hindering the early convergence, inducing optimized answer.

Despite being a classic example of optimization problem, the concept of Traveling Salesman Problem could be applied in actual and practice situations as the optimization of the energy provision to trajectories determination in electric network, permitting an ample field of applications.

This software is available for educational purposes without cost. For that, please, contact the last author.

Acknowledgment

The authors gratefully acknowledge the CNPq, a Brazilian research funding agency, CAPES and SESu/PET, in the form of research scholarships, and FAPEMIG, a Minas Gerais State research funding agency, which supported this work.

References:

[1] A.R. Aoki, A.A.A. Esmin & G. Lambert-Torres, “An Architecture of a Multi-Agent System for Power System Operation”, WSEAS Transactions on Computers, World Scientific and Engineering Society Press, ISSN 1109-2750, No. 2, Vol. 3, pp. 408-412, April 2004.

[2] H.G. Arango and G. Lambert-Torres, “Spatial Electric Load Distribution Forecasting using Simulated Annealing”, WSEAS Transactions on Systems, World Scientific and Engineering Society Press, ISSN 1109-2777, No. 1, Vol. 3, pp. 14-19, January 2004.

[3] J. Kennedy, Russel Eberhart, “A Discrete Binary Version fo the Particle Swarm Algorithm”, IEEE Conference on Systems, Man and Cybernetics, Vol. 5, pp; 4104 – 4108, Oct., 1997.

[4] J. Kennedy and R. C. Eberhart, Particle swarm optimization, in Proc. IEEE Int. Conf. on Neural Networks, pp. 1942-1948,1995(4).

[5] J. Kennedy, Russell Eberhart, “Swarm Intelligence”, San Mateo, CA: Morgan Kaufmann, 2001.

[6] J. Kennedy, “Small Worlds and Mega-Minds: Effects of Neighbourhood Topology on Particle Swarm Performance”, Proc. of IEEE Conference on Evolutionary Computation, Washington D.C, USA, July, 1999.

[7] J. Kennedy, Rui Mendes, “Population Structure and Particle Swarm Performance”, Proc. of Congress on Evolutionary Computation, Vol. 2, pp. 1671 – 1676, May, 2002.

[8] L. Schoeman and A. P. Engelbrecht, “Containing Particles inside Niches when Optimizing Multimodal Functions”, Proceedings of SAICSIT, pp. 78 – 85, 2005.

[9] A.A. A. Esmin, G. Lambert-Torres and A.C.Z. Souza, "A hybrid particle swarm optimization applied to loss power minimization," IEEE Trans. Power Systems, vol. 20(2), pp. 859-866, 2005.