Simplified particle swarm optimization algorithm

Ricardo Paupitz Barbosa dos Santos[2], Carlos Humberto Martins[1*] and Fábio Lúcio Santos[3]

[1]Departamento de Engenharia Civil, Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá, Paraná, Brazil. [2]Departamento de Engenharia Agrícola, Universidade Estadual de Maringá, Maringá, Paraná, Brazil. [3]Departamento de Engenharia Agrícola, Universidade Federal de Viçosa, Viçosa, Minas Gerais, Brazil. *Author for correspondence. E-mail: chmartins@uem.br

ABSTRACT. Real ants and bees are considered social insects, which present some remarkable characteristics that can be used, as inspiration, to solve complex optimization problems. This field of study is known as swarm intelligence. Therefore, this paper presents a new algorithm that can be understood as a simplified version of the well known Particle Swarm Optimization (PSO). The proposed algorithm allows saving some computational effort and obtains a considerable performance in the optimization of nonlinear functions. We employed four nonlinear benchmark functions, Sphere, Schwefel, Schaffer and Ackley functions, to test and validate the new proposal. Some simulated results were used in order to clarify the efficiency of the proposed algorithm.

Keywords: optimization, swarm intelligence, global minimum, algorithm.

Algoritmo simplificado de otimização de enxame de partículas

RESUMO. Formigas e abelhas são consideradas insetos sociais, os quais apresentam notáveis habilidades que podem ser empregadas, como inspiração, na solução de complexos problemas de otimização. Portanto, este trabalho apresenta um novo algoritmo o qual pode ser entendido como uma versão simplificada do conhecido Otimização por Enxame de partículas (Particle Swarm Optimization - PSO). O algoritmo proposto permite a redução do esforço computacional e a obtenção de um desempenho considerável na otimização de funções não-lineares. Foram utilizadas quatro funções não-lineares de referência, como Esférica, Schwefel, Schaffer e Ackley, para teste e validação da nova proposta. Alguns resultados simulados são apresentados para ilustrar a eficiência do algoritmo proposto.

Palavras-chave: otimização, inteligência de enxame, mínimo global, algoritmo.

Introduction

In natural systems, we commonly observe mechanisms where natural agents seem to be organized in a rational and ordered way. Social insects, such as ants and bees, are especially interesting examples, and exhibit some remarkable characteristics. Despite the extremely low intellectual capacity of the individuals, the colony can solve surprisingly complex problems while searching for food (DORIGO et al., 1996; KARABOGA; AKAY, 2009; KARABOGA; BASTURK, 2007).

Some of these natural agents are the main inspiration for interesting and powerful algorithms used in the search for optimal solutions of highly non-trivial problems (CHENG et al., 2009; KARABOGA, 2009; KARABOGA; BASTURK, 2007; TOKSARI, 2006; WANG et al., 2007) such as finding the global minimum of nonlinear functions, truss optimization, the classical traveling salesman problem, electric power systems, traffic flow, polymer design, the Schottky-Barrier estimation in diode models and several other applications (ALRASHIDI; EL-HAWARY, 2009; HUANG et al., 2003; MARTINS et al., 2008; TEODOROVIC 2003; WANG; YE, 2009).

This field of study is known as "*swarm intelligence*" and has attracted an increasingly number of researchers since the proposal of *Particle Swarm Optimization* (PSO) algorithm and also of the *Ant Colony Optimization (*ACO) Algorithm (DORIGO et al., 1996).

In the specific case of PSO, the algorithm can be understood as the application of an updating process on a set of particles which are moving throughout the search space. In the traditional version, each particle is defined by three vectors: the position in the D-dimensional search space x_i, the individual best position p_i, and the velocity v_i. Particles are initialized with random vectors for initial positions and initial velocities. At each step, velocities and positions of the entire swarm are updated accordingly with the rules described as follows.

PSO Algorithm – the updating process
for each time step t **do**
  for each particle I in the swarm **do**
    update *position* x_i using $v_i = v_i + ce_1(p_i - x_i) + ce_2(p_g - x_i)$,

$$x_i = x_i + v_i$$

    calculate particle fitness $f(x_i)$
    update p_i and p_g
  end for
end for

where:

    c is a constant with the value of 2.0, e_1 and e_2 are random numbers and p_g is the best position of the neighbors of the particle.

    The main simplification was done by suppressing the velocity updating present in the original PSO. In this way, we may save some computational effort and obtain a reasonable performance (*i.e.* in comparison to that from the PSO) in the optimization of nonlinear functions. Furthermore, the implementation of the algorithm described in the next section is straightforward, making it easy to work with.

Material and methods

Simplified Particle Swarm Optimization algorithm

    The proposed algorithm, which is presented in Figure 1, can be understood as a simplified version of the PSO algorithm and can be used to find the global minimum of nonlinear functions. Initially, a *swarm* of particles is defined, then they are randomly distributed over the search space, and an updating strategy is applied at each step in order to define the new positions of the particles. This updating process is made accordingly to the relation given by the Equation 1:

$$x_{i+1} = x_i + (R + S_i)w_i \qquad (1)$$

where:

    x_i represents the position vector of particle *i*, *R* is a Random number between -1 and +1, w_i is a vector pointing the direction of the line joining the point "i" and x_{Global} and with size $|w|$ = (domain size)/(number of particles). S_i is determined accordingly with the relative position of the particle and the best result obtained until that moment in the simulation, which is called x_{Global} and calculated by equations (2a), (2b) and (2c), as follow:

$$x_i - x_{Global} < 0 \Rightarrow S_i = -1|x_i - x_{Global}| \qquad (2a)$$

$$x_i - x_{Global} > 0 \Rightarrow S_i = |x_i - x_{Global}| \qquad (2b)$$

$$x_i - x_{Global} = 0 \Rightarrow S_i = 0 \qquad (2c)$$

    The calculation of S_i has fundamental importance on the proposed algorithm, since it is the feature that makes the particles move in a non-symmetric way. The updated position tends to be located closer to the global solution than its predecessor. This behavior allows a fast concentration of particles around the "instantaneous" best solution until the maximum number of iterations is reached, or until a new best solution (x_{best}) is found. Thus, this "directed random walk" near x_{best} can rapidly improve the obtained solution when it is not the global minimum yet.
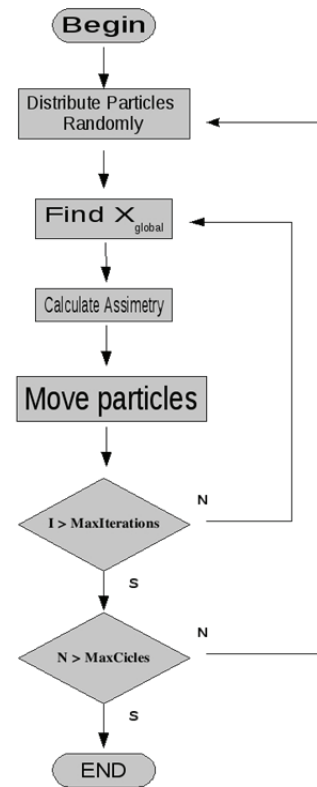
Figure 1. Diagram of the proposed Simplified PSO algorithm.

Algorithm test and validation

    The aim of the test and the validation step is to discuss the effect of the size of the swarm (or colony) proposed algorithm on the performance in the search for a global minimum of a nonlinear function. In this way, since the main interest is the optimization of functions, it is necessary to define some nonlinear benchmark functions to compare the obtained results with previous studies (KARABOGA; BASTURK, 2007; TOKSARI, 2006). The chosen functions are presented in Table 1.

Table 1. Benchmark nonlinear functions employed to test the proposed swarm optimization algorithm.

| Function	Analytical Expression	Ranges	Minimum
Sphere	$F_1 = x_1^2 + x_2^2$	[-10,10]	0
Schwefel	$F_2 = 837.9658 - x_1 \sin\left(\sqrt{\|x_1\|}\right) - x_2 \sin\left(\sqrt{\|x_2\|}\right)$	[-500,500]	0
Schaffer	$F_4 = 0.5 + \dfrac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$	[-3,3]	0
Ackley	$F_3 = 20 + e - 20e^{\left(-0.2\sqrt{\frac{1}{2}\sum_{i=1}^{2} x_{i^2}}\right)} - e^{\frac{1}{2}\sum_{i=1}^{2}\cos(2\pi x_i)}$	[-20,20]	0

The benchmark nonlinear functions constructed by analytical expressions (presented in Table 1) are shown in Figure 2. These functions, considered objective functions, were generated using the proposed ranges, also presented in Table 1.

Actually, the idea of some kind of asymmetric sorting for the direction of the next step is the central feature of the algorithm. In this way, the asymmetry can be described as an "attraction", or a scent comparing with ant's and bee's algorithms. The exclusive source of this scent will always be the best solution found until that moment in the simulation, and this source changes readily when the better solution is improved.

Results and discussion

Several tests were carried out in order to compare the performance of the proposed algorithm with the well known ACO (Ant Colony Optimization) and ABC (Artificial Bee Colony) algorithms. For this, a computational code was developed using the Fortran 90 Programming Language. This software makes it possible to test the proposed algorithm on the task of finding the global minimum from some benchmark functions, as presented in Table 1.
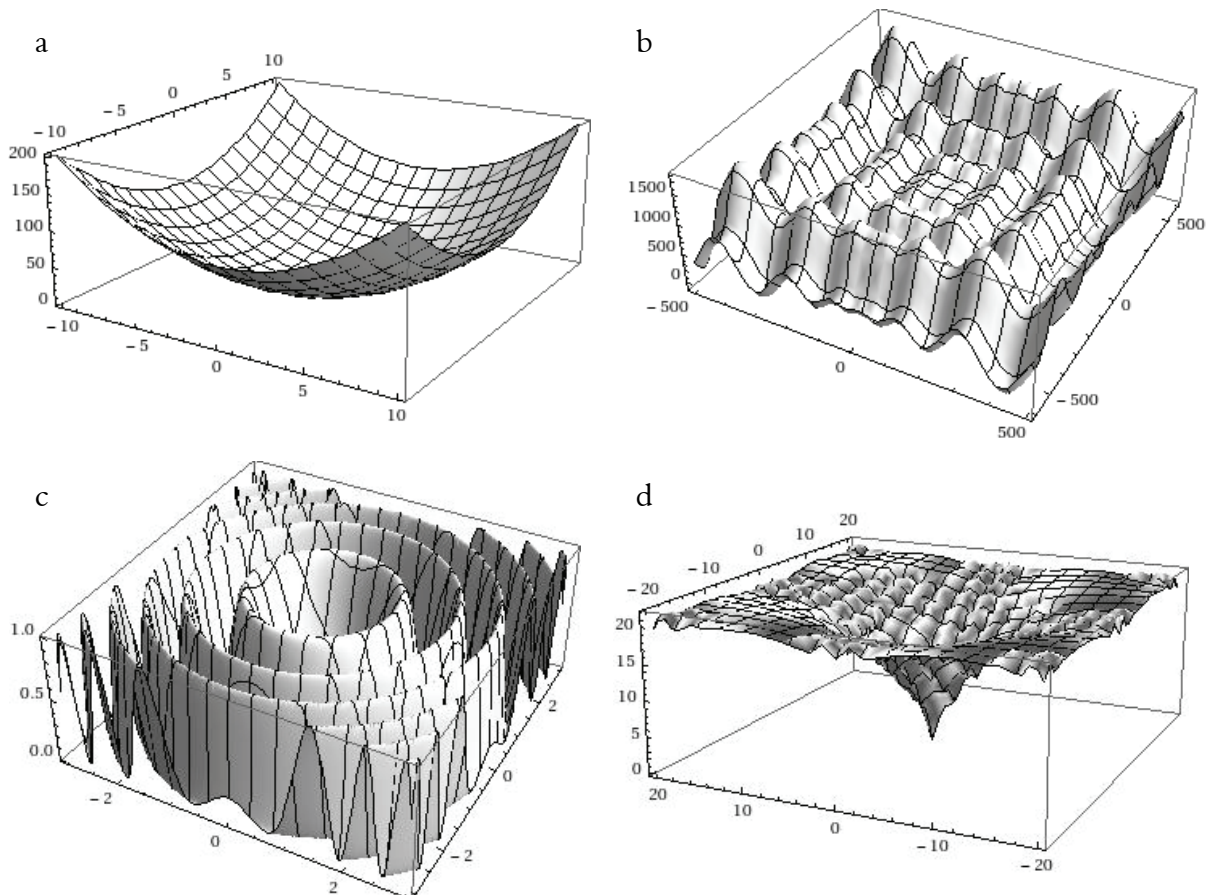
Figure 2. Benchmark functions: a) Sphere, b) Schwefel, c) Schaffer and d) Ackley.

In Figures 3 to 6, we present the results for the global minimum search, using the benchmark functions shown in Table 1. In these graphs we have the mappings of the evolution of the global minimum results (y axis) as a function of the number of iteration cycles (marked on the x axis).
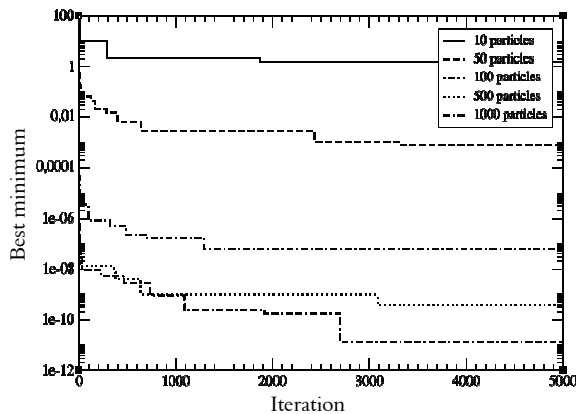

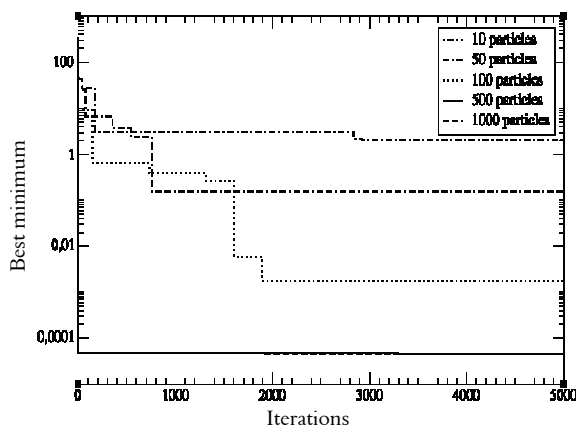**Figure 3.** Evolution of best value Sphere function for various colony sizes.

Figure 4. Evolution of best value Schweffel function for various colony sizes.

Besides that, we may observe that as colony size increases, the convergence of the global minimum result becomes faster. Actually, for colonies with a number of particles higher than a given number, which depends on the function considered, the global minimum found can be orders of magnitude smaller than before. On the other hand, when the colony size is larger than the critical size, increasing it will not improve significantly the performance. This last characteristic evidenced that the analysis of these "size-effects" can be an important task while evaluating the performance of a new algorithm, and that tests could be done using colony sizes that optimize the performance for the specific problem.
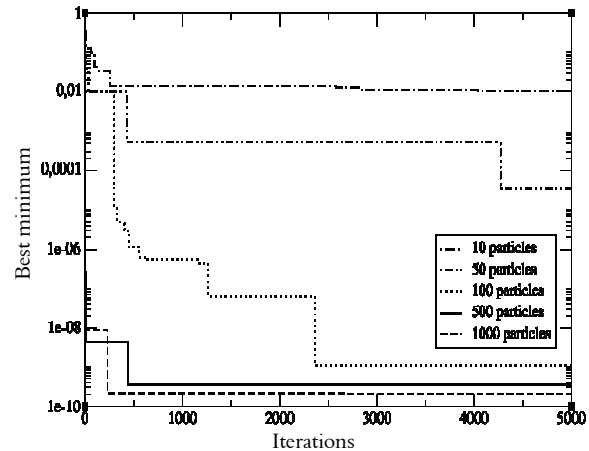

Figure 5. Evolution of best value for Schaffer function for various colony sizes.
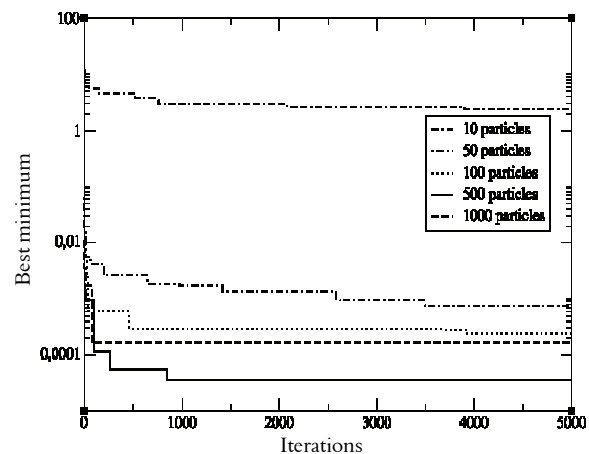

Figure 6. Evolution of best value for Ackley function for various colony sizes.

In the Table 2, we may compare numerical results obtained through the simulations carried out with the benchmark functions. The simulations of figures 3 to 6 were chosen to represent typical results for specific parameters. Several tests were accomplished using different seeds for the random number generator in order to ensure that the results shown are typical, instead of pathological ones.

Conclusion

A simplified version of the PSO algorithm was presented. The version proposed here is very simple to implement and the performance is comparable to other "colony" algorithms.

Four nonlinear "benchmark" functions were selected to test the algorithm in the search for the global minimum. This kind of test is of great importance in the case of functions, such as Schwefel, Ackley and Schaffer functions, which

present serious difficulties to have their global minimum "revealed" by traditional methods.

Moreover, there is an optimal size (*i.e.* number of particles) for the colony that optimizes the convergence and provides great precision for the global minimum localization.

The results shown here, despite their simplicity, indicate that even simple algorithms can obtain good performance while solving non-trivial problems. Other aspect is the simple implementation, which makes it good as a "first trying" algorithm, at least when proving the power of swarm algorithms.

References

ALRASHIDI, M. R.; EL-HAWARY, M. E. A survey of Particle Swarm Optimization applications in electric power systems. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 4, p. 913-918, 2009.

CHENG, F. Q.; YE, F. F.; YANG, J. G. An ant colony optimization algorithm for partner selection in virtual enterprises. **International Journal of Materials and Product Technology**, v. 34, n. 3, p. 227-240, 2009.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics**, v. 26, n. 1, p. 29-41, 1996.

HUANG, L.; ZHOU, C. G.; WANG, K. P. Hybrid ant colony algorithm for traveling salesman problem. **Progress in Natural Science**, v. 13, n. 4, p. 295-299, 2003.

KARABOGA, D.; AKAY, B. A comparative study of artificial bee colony algorithm. **Applied Mathematics and Computation**, v. 214, n. 1, p. 108-132, 2009.

KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. **Journal of Global Optimization**, v. 39, n. 3, p. 459-471, 2007.

KARABOGA, N. A new design method based on artificial bee colony algorithm for digital IIR filters. **Journal of the Franklin Institute-Engineering and Applied Mathematics**, v. 346, n. 4, p. 328-348, 2009.

MARTINS, B. V. C.; BRUNETTO, G.; SATO, F. Designing conducting polymers using bioinspired ant algorithms. **Chemical Physics Letters**, v. 453, n. 4-6, p. 290-295, 2008.

TEODOROVIC, D. Transport modeling by multi-agent systems: a swarm intelligence approach. **Transportation Planning and Technology**, v. 26, n. 4, p. 289-312, 2003.

TOKSARI, M. D. Ant colony optimization for finding the global minimum. **Applied Mathematics and Computation**, v. 176, n. 1, p. 308-316, 2006.

WANG, J.; LIU, Y. H.; TIAN, D. X. An ant colony algorithm with global adaptive optimization. **Journal of Computational and Theoretical Nanoscience**, v. 4, n. 7-8, p. 1283-1289, 2007.

WANG, K.; YE, M. Parameter estimation of Schottky-barrier diode model by Particle Swarm Optimization. **International Journal of Modern Physics C**, v. 20, n. 5, p. 687-699, 2009.