XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

COMPARATIVE STUDY OF METAHEURISTICS BASED ON SWARM INTELLIGENCE: WHALE OPTIMIZATION ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Kayon V. L. Lopes
Hadriel R. Santos
Lucas N. Mendonça
João P. F. Pereira
Danúbia. S. Pires
kayonvinicius222@hotmail.com
hadrireis@gmail.com
lnm.lucas@hotmail.com
jpfp15@hotmail.com
danubiapires@ifma.edu.br
Instituto Federal do Maranhão
Av. Getúlio Vargas, 04, Monte Castelo, 65030-005, São Luís, MA, Brasil

Abstract. Optimization is the field of studies that seeks to develop techniques to improve processes, leading them to the best operating scenario. Within the field of optimization studies, the idea of constructing new techniques influenced by the adaptation mechanisms of living beings has been developed. In the context of computation, the construction of bio-inspired algorithms gains notoriety because it allows agents that perform computationally simple tasks to contribute to solve complex problems when grouped together. This paper presents a comparative study between the metaheuristic techniques based on swarm intelligence: WOA (Whale Optimization Algorithm) and PSO (Particle Swarm Optimization), in order to identify among the techniques the algorithm that obtains the best performance, investigating the influence of the number of individuals of the population, the influence of parameters on computational cost and execution time, as well as enriching the literature regarding the choice of methods and parameters according to the determined domains. The algorithms were implemented and benchmark functions widely known in the literature were used to evaluate the efficiency of the proposed methods and compare the results obtained. The results demonstrated the optimum performance for the search of minimums and maximums, and the best parameters for each algorithm.

Keywords: Computational Optimization, Metaheuristics, Evolutionary Computation, Whale Optimization Algorithm, Particle Swarm Optimization

1 Introduction

Optimization is the field of studies that seeks to develop techniques to improve processes, leading them to the best operating scenario [1]. Within the field of optimization studies the idea has been developed to construct new techniques influenced by the mechanisms of adaptation of living beings, as observed in nature. In the context of computation, the construction of bio-inspired algorithms gains notoriety because it allows agents that perform computationally simple tasks to contribute to solve complex problems when grouped together. Add to these factors the factor that the techniques of Evolutionary Computing operate on a population in parallel. Thus, they can search in different areas of the solution space, allocating a number of members suitable for searching in various regions. As a result, such techniques have a greater chance of reaching the promising areas of the search space [1]. This article presents a comparative study between the Whale Optimization Algorithm (WOA) and the Particle Swarm Optimization (PSO), in order to enrich the literature on the choice of methods and parameters according to the determined domains.

2 Methods

2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization was developed by Russell Eberhart and James Kennedy in 1965. The inspiration for this method of optimization is the social and individual behavior of some species of birds as they search for food or through their nests. The main idea of this method is that the best solution (the best particle) is found inside a swarm containing n other particles. The best solution among the n particles in an iteration is stored. This value is called $pbest$. The best solution of any swarm obtained so far, called $gbest$ is also stored. At each iteration, the velocity of the particles is changed by taking $pbest$ and $gbest$ into them so that they move toward the minimum or maximum of the function to be minimized or maximized [2].

The initial positions and velocities are randomly defined within the search space. Then the new position of a particle is determined by its previous position and its velocity. The motion of each particle is governed by

$$V_i^{t+1} = wV_i^t + \beta_1\gamma_1(X_{pbest,i}^t - X_i^t) + \beta_2\gamma_2(X_{gbest}^t - X_i^t) \tag{1}$$

$$X_i^{t+1} = X_i^t + V_i^t \tag{2}$$

where $V_i^{(t+1)}$ is the velocity of the i-th particle in the next iteration, V_i^t is the current velocity of the particle, $X_i^{(t+1)}$ is the coordinate of the i-th particle in the next iteration, X_i^t is the current coordinate of this particle. The parameters $X_{pbest,i}^t$ e $X_{gbest,i}^t$ are respectively the best positions of each particle and the best coordinate of the whole swarm; β_1 e β_2 are the cognitive parameter (ponders the local searches) and the social parameter (weights the global searches), respectively; γ_1 e γ_2 are random numbers generated uniformly in the interval [0,1]; $w \in [0,1]$ is the coefficient of inertia, responsible for controlling the ability of particles to explore the search space [2].

Three principles describe the exploration process that the PSO uses to find the best solutions [2]:

- Evaluation: the particles have the ability to measure the environment and estimate their own behavior before it;
- Comparison: the particles share information among themselves to establish comparative references within the swarm;
- Imitation: from the social behavior, the particles assimilate relevant information from the best agents, guaranteeing the acquisition and maintenance of cognitive abilities.

With this, it is possible to find optimal solutions from information that each particle finds individually and shares with the whole swarm. Random individual initialization and search processes in the solution space ensure a global optimization character [2].

2.2 Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm was developed by Seyedali Mirjalili and Andrew Lewis in 2016. It mimics the social behavior of humpback whales and it is inspired by the strategy of hunting for bubble nets.

Whales are the largest mammals on Earth. In addition, they are very intelligent predators, capable of feeling emotions, develop social bonds, judge, learn and some may even develop their own dialect [3]. With regard to their social behavior, whales may live alone or, as is most commonly observed, in groups. Humpback whales have a special method of hunting. It has been observed that they dive and create bubbles around the tusks as they travel a spiral path toward the surface. WOA is developed from this behavior [3] [4]. WOA has three operators: surrounding fangs, bubble neutralization and search for prey. These operators are applied to generate new populations [3] [4].

In the Operator of Surrounding Fangs, the position of a humpback whale in space is defined by the ”target prey” solution. The best candidate is defined and the other candidates try to update their positions in relation to this prey (considered the best solution of the generation). Equating it has:

$$\vec{D} = |\vec{C}\vec{X}^*(t) - \vec{X}(t)| \tag{3}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}\vec{D} \tag{4}$$

where t indicates iteration number, $\vec{D}$ is the distance to the best solution $\vec{X}^*$. The coefficients $\vec{A}$ and $\vec{C}$ are given by $\vec{A} = \vec{a}\vec{r} - \vec{a}$ and $\vec{C} = 2\vec{r}$, respectively, where $\vec{a}$ is $\vec{a} = 2 - \frac{2t}{t_{max}}$ being that $\vec{a}$ linearly decreases in the interval [2,0] along the iterations and $\vec{r}$ is a vector of random numbers in the interval [0,1] [3].

The Bubble Neutralization Operator has two basic mechanisms, the curved shrinkage and the spiral positioning, which are responsible for updating the best candidates of the population. In the first, $\vec{A}$ is a random number located in the interval [-a, a], which is reduced over the generations. Defining,$\vec{A}$, the new position of a search candidate is defined from anywhere between the original position and the position of the best candidate of the current generation [3] [4]. In the second, based on the spiral movement of humpback whales, a relation is defined to generate a new position. The Bubble Neutralization Operator has two basic mechanisms, the curved shrinkage and the spiral positioning, which are responsible for updating the best candidates of the population. This operator is given by:

$$\vec{X}(t+1) = \vec{D}' e^{bl} cos(2\pi l) + \vec{X}^*(t) \tag{5}$$

where $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ represents the distance that the whale is from the prey (best solution obtained so far) in the tth generation, b is a constant that defines the shape of the logarithmic, l is a random number between [-1,1] [3] [4].

To describe the movement of the whales around the prey (along a more and more shrinking spiral path), it is assumed that the possibilities of shrinking the surrounding mechanism and updating the position of the whales are equal, that is

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A}\vec{D}, & \text{se } p < 0.5 \\ \vec{D}' e^{bl} cos(2\pi l) + \vec{X}^*(t), & \text{se } p \geq 0.5. \end{cases} \tag{6}$$

In the Hold Search Operator (scan phase), the parameters $\vec{A}$ and $\vec{C}$ of each generation are updated. When $|\vec{A}| < 1$, the exploit is used to update the position of the candidates from the best search agent of the generation, when $|\vec{A}| \geq 1$ find the overall optimum. Mathematically

$$\vec{D}'' = |\vec{C}\vec{X}_{rand}(t) - \vec{X}(t)| \tag{7}$$

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{A}\vec{D}'' \tag{8}$$

where $\vec{X}_{rand}$ is a random position vector chosen from the current population [3] [4].

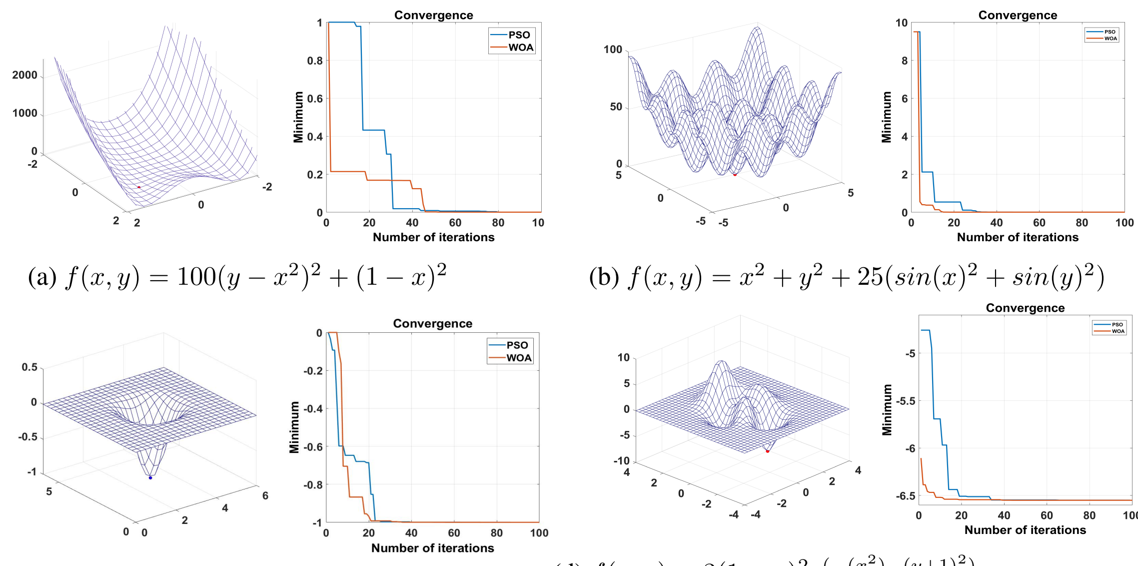
3 Results and discussions

In this article a comparative study was carried out between Whale Optimization Algorithm, proposed in 2016 by [3], and *Particle Swarm Optimization*, proposed in 1995 by [2], a method of optimization widely used and recognized as efficient.

To demonstrate the efficiency of algorithms and evaluating each of the methods, both were employed in minimization eight functions *benchmark* widely used in the literature [3] [5] and [6]. In this work we used the benchmark functions shown on Fig. 1, known as the Rosenbrock function, with a minimum of 0 in (1,1); Eggcrate, with a minimum of 0 in (0,0); Easom, with a minimum of -1 in (π,π); and Peaks function, with a minimum of -6.5511 in (0.228279,-1.62553), to be addressed in detail. The tests were performed in the range of [-10,10] for x and y.

In the implementation of the PSO, the cognitive parameter $\beta_1 = 2$ and the social parameter $\beta_2 = 2$ was used. The coefficients control the influence of the individual position of each particle and an increase of the positions of the individuals in a collective way. For simplicity, in this work, we adopted equal values for both parameters. However, the weights can be changed. For the inertia calculation, values between 0.4 and 0.9 are used, in this work we used $w = 0.8$. In case of WOA it was necessary to choose the parameter b responsible for defining the scheme of the logarithmic spiral defined by Eq. (5). In [3] satisfactory results are obtained for $b = 1$, therefore, this value was adopted during implementation.

Figure 1 show the curves of the benchmark functions with the minimum obtained through the PSO and WOA for a fixed number of 100 iterations.



(a) $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$

(b) $f(x, y) = x^2 + y^2 + 25(sin(x)^2 + sin(y)^2)$

(c) $f(x, y) = -cos(x)cos(y)e^{(-(x-pi)^2-(y-pi)^2)}$

(d) $f(x, y) = 3(1 - x)^2 e^{(-(x^2)-(y+1)^2)} - 10(\frac{x}{5} - x^3 - y^5)e^{(-x^2-y^2)} - \frac{1}{3}e^{(-(x+1)^2-y^2)}$

Figure 1. Benchmark functions used in the tests: (a) Rosenbrock, (b) Eggcrate, (c) Easom and (d) Peaks.

From the algorithm convergence graphs for each function, it was observed that, although both algorithms found the global minimum with few iterations, WOA presented a slightly faster convergence for the same number of individuals and number of iterations.

To verify the influence of the number of individuals in the execution time, tests were performed varying the size of the population between 10 and 50 individuals. For each variation the algorithms were executed 30 times, and the execution time was obtained from the average between the times obtained for the PSO and the average between the times obtained for WOA. As the stopping criterion a minimum error of 10^{-6} was used between the known minimum and the value obtained. The average times obtained as a function of population size are shown in Fig. 2.

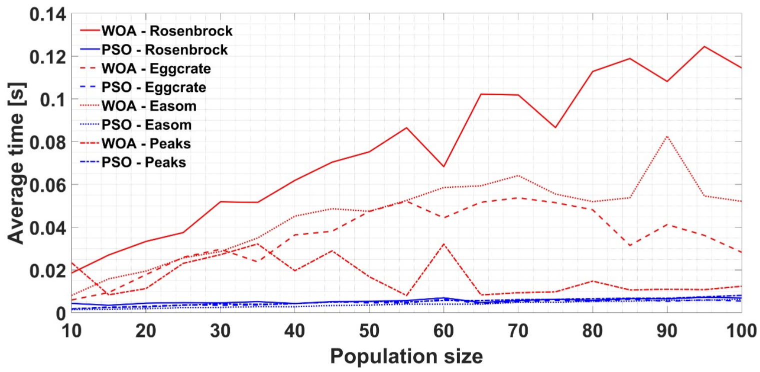


Figure 2. Effect of population size on average time.

4 Conclusion

From the graphs and execution times obtained, it was observed that the WOA obtained a convergence slightly faster than the PSO, although both met the criterion of stopping with a small number of iterations even for functions with many local minimums and maximums. With regard to the execution time, it was observed that the PSO showed little variation of the average time with the same gradual increase in the quantity of particles. WOA showed a longer mean time varying with the number of agents almost linear as can be seen from Fig. 2. Therefore, the PSO has proved to be more efficient in dealing with problems in which the number of particles required is relatively large. However, the difference between times is in the order of hundredths of seconds.

Also, in WOA, a more efficient exploration of the search space was observed. In the tests performed, functions such as Eggcrate, where there is a presence of many minimums and maximums, the PSO was more susceptible to stagnation in local minimums and maxima, differently WOA, which was higher in the exploration, and consequently, converges to the global minimum with a smaller number of iterations.

Acknowledgements

This work was supported by FAPEMA and Federal Institute of Maranhão.

References

[1] Gaspar-Cunha, A., Takahashi, R., & Antunes, C. H., 2012. *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra/Coimbra University Press.

[2] Eberhart, R. & Kennedy, J., 1995. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43. Ieee.

[3] Mirjalili, S. & Lewis, A., 2016. The whale optimization algorithm. *Advances in engineering software*, vol. 95, pp. 51–67.

[4] Lobato, F. S., Mundim-Filho, O. G., & Souza, D. L., 2018. Projeto de um reator de amônia usando o whale optimization algorithm. *REVISTA CEREUS*, vol. 10, n. 2.

[5] Yao, X., Liu, Y., & Lin, G., 1999. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, vol. 3, n. 2, pp. 82–102.

[6] Yang, X.-S., 2010. Firefly algorithm, stochastic test functions and design optimisation. *arXiv preprint arXiv:1003.1409*.