Bioinspired metaheuristic algorithms for global optimization

Marko Mitić*, Najdan Vuković**, Milica Petrović*, Jelena Petronijević*, Ali Diryag*, Zoran Miljković*
* University of Belgrade - Faculty of Mechanical Engineering/Production Engineering Department, Belgrade, Serbia
** University of Belgrade - Faculty of Mechanical Engineering/Innovation Center, Belgrade, Serbia
mmitic@mas.bg.ac.rs, nvukovic@mas.bg.ac.rs, mmpetrovic@mas.bg.ac.rs, jpetronijevic@mas.bg.ac.rs, ali6981@gmail.com, zmiljkovic@mas.bg.ac.rs

Abstract—**This paper presents concise comparison study of newly developed bioinspired algorithms for global optimization problems. Three different metaheuristic techniques, namely Accelerated Particle Swarm Optimization (APSO), Firefly Algorithm (FA), and Grey Wolf Optimizer (GWO) are investigated and implemented in Matlab environment. These methods are compared on four unimodal and multimodal nonlinear functions in order to find global optimum values. Computational results indicate that GWO outperforms other intelligent techniques, and that all aforementioned algorithms can be successfully used for optimization of continuous functions.**

I. Introduction

In recent years, various nonlinear optimization problems are solved using biologically inspired solutions. Main reason lies in a fact that, in these cases, traditional algorithms often fail in producing wanted/expected results. Bioinspired metaheuristic techniques represent a well-known mathematical tool for solving hard optimization tasks that cannot be solved using other approaches. This study represent a concise comparison of three such algorithms for finding global optimum of continuous nonlinear functions.

Bioinspired metaheuristic algorithms mimic the behaviour of animals in nature by turning their swarming, flocking or grouping into mathematical procedures. In these intelligent methods, an algorithm starts with random population of individuals that are next grouped around optimal solution using iterative search. In comparison with single-based algorithms, population-based metaheuristic has significant advantages in finding overall best optimization result [1]:

- Multiple candidate solutions share information about the search space which results in sudden jumps toward the promising part of search space.
- Multiple candidate solutions assist each other to avoid locally optimal solutions.
- Population-based meta-heuristics generally have greater exploration compared to single solution-based algorithms.

These clear advantages influence the development of large number of new bioinspired optimization algorithms over the last decade. Most popular techniques in the field include, firefly algorithm [2], cuckoo search [3], bat algorithm [4], grey wolf optimizer [1], and particle swarm optimization [5], which are successfully applied for solving various engineering problems [6,7,8,9,10]. These studies prove that metaheuristic algorithms are superior in avoiding stagnation in local solution due to their stochastic nature

Despite the aforementioned, the main disadvantage of metaheuristic methods lies in the fact that there is no guarantee that found solution is actually the optimal one. Likewise, in some cases the algorithm dependent parameters is hard to determine. However, in most real world problems, the search space is usually unknown and prone to large number of local optimums, so the metaheuristics with the ability of extensive search in this space represents good option.

In this paper, three popular metaheuristic techniques, namely Accelerated Particle Swarm Optimization (APSO), Firefly Algorithm (FA), and Grey Wolf Optimizer (GWO) are compared in optimization task of different nonlinear functions. These algorithms are tested on four unimodal and multimodal well-known benchmark problems. Results show the efficiency of bioinspired methods since in all cases, the experimentally obtained best algorithm successfully converged.

The paper is organized as follows. After the brief introduction, the main optimization methods are introduced in the second section. In the third part of the paper mathematical description of nonlinear functions is presented. Fourth section show the obtained computational results. Finally, the last section gives the overall conclusion of this study.

II. Bioinspired algorithms

In this section mathematical description of each intelligent method is given in brief.

A. Accelerated Particle Swarm Optimization - APSO

This algorithm is inspired by fish schooling behavior in nature [5,10]. Each individual (e.g. particle) in swarm flies toward its best and currently best solution of the given problem. Likewise, the algorithm incorporates random component, so the global search is to some point stochastic. Two main equations of the traditional particle swarm optimization algorithm are [10, 11]:

$$v_i^{t+1} = v_i^t + \alpha r_1 \odot \left[g^* - x_i^t \right] + \beta r_2 \odot \left[x_i^* - x_i^t \right] \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

where: symbol $\odot$ is the Hadamard product, v_i^t and x_i^t are current velocity and position of the particle, respectively, v_i^{t+1} and x_i^{t+1} are next velocity and position of an individual respectively, r_1 and r_2 are two random vectors, α and β are algorithm's learning parameters. In recent research work, an accelerated version of this algorithm is introduced [11]. Velocity in APSO is calculated with

$$v_i^{t+1} = v_i^t + \alpha r(t) + \beta\left[g^* - x_i^t\right], \tag{3}$$

where: g^* is global best solution, and r is drawn from standard Gaussian distribution. Equation (2) is also modified, so that location of the particle in APSO is calculated as

$$x_i^{t+1} = (1-\beta)x_i^t + \beta g^* + \alpha r\,. \tag{4}$$

Comparing (2) and (4) one can conclude that APSO does not include velocity parameter. APSO uses only parameters α and β, so it is much easier to initialize and to understand. These are the main reasons for the implementation of APSO over PSO in this research work.

B. Firefly Algorithm - FA

Social behavior of fireflies in nature served as an inspiration for developing FA [2].Two important issues of this method should be noted: the variation of light intensity in real fireflies and the formulation of attractiveness [12]. The light intensity parameter is of crucial importance since it represent fitness function (e.g. optimization goal), and influence the movement of the entire swarm. The attractiveness of one firefly to another individual in swarm corresponds to light intensity and is calculated as:

$$\beta = \beta_0 \cdot e^{-\gamma r^2}, \tag{5}$$

where: r is the Euclidian distance between two individuals, γ is the parameter of light absorption, and β_0 is the attractiveness at $r = 0$. The movement of firefly i towards firefly j is now defined as:

$$x_i = x_i + \beta_o e^{-\gamma r^2}(x_j - x_i) + \alpha\varepsilon_i, \tag{6}$$

where: ε_i is random Gaussian number, and α is the randomization parameter chose by the designer. Using this last two equations, (5) and (6), fireflies can be sorted in accordance with their light intensity (i.e. achieved performance), and then directed towards the better solution.

C. Grey wolf optimizer - GWO

It is known that grey wolves are considered top of the food chain, and are usually grouped in small packs. Of particular interest for the GWO method is that they have a very strict social dominant hierarchy [1]. The solutions in GWO are generated using individuals defined as: hierarchy leader (i.e. alpha), subordinate wolf in the second level (i.e. beta), third level individual (i.e. delta), and low-level wolf (i.e. omega).

Logically, the fittest GWO solution of the optimization problem is described by alpha. Similarly to this, second and third best solutions are beta and delta, respectively. Rest of the solutions are defined using omega wolves. Optimization with GWO is guided by alpha, beta and gamma solutions, obtained through processes of tracking, encircling and attacking prey. Mathematical model for encircling behavior is given with [1]

$$\vec{D} = \left|\vec{C}\cdot\vec{X}_p(t) - \vec{X}(t)\right| \tag{7}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\cdot\vec{D} \tag{8}$$

where: $\vec{X}_p$ and $\vec{X}$ are position vector of a prey and grey wolf, respectively, $\vec{A}$ and $\vec{C}$ indicate coefficient vectors, and t is the current iteration. Coefficients are calculated with

$$\vec{A} = 2\vec{a}\cdot\vec{r}_1 - \vec{a}\,, \tag{9}$$

$$\vec{C} = 2\cdot\vec{r}_2\,, \tag{10}$$

where: $\vec{a}$ is linearly decreased over iterations from 2 to 0, and $\vec{r}_1$, $\vec{r}_2$ are random vectors chosen in the domain $[0,1]$. It is presumed that, for the hunting phase, alpha, beta and gamma have better knowledge about the location of the pray [1]. Therefore, all other individuals (i.e. omega wolves) update their position based on the information of these aforementioned three best solutions. Equations that describe this most important step are as follows

$$\begin{aligned}\vec{D}_\alpha &= \left|\vec{C}_1\cdot\vec{X}_\alpha - \vec{X}\right|,\\ \vec{D}_\beta &= \left|\vec{C}_2\cdot\vec{X}_\beta - \vec{X}\right|,\\ \vec{D}_\delta &= \left|\vec{C}_3\cdot\vec{X}_\delta - \vec{X}\right|,\end{aligned} \tag{11}$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1\cdot\left(\vec{D}_\alpha\right),\\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2\cdot\left(\vec{D}_\beta\right),\\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3\cdot\left(\vec{D}_\delta\right),\end{aligned} \tag{12}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}\,. \tag{13}$$

Vector $\vec{A}$ is responsible for attacking the prey, i.e. for exploitation phase. The value $\vec{a}$ is linearly decreasing, and therefore is parameter $\vec{A}$. It is shown that $|A| < 1$

forces the wolves to attack towards the pray [1]. Unlike exploitation vector, parameter $\vec{C}$ favors the exploration. This component provides random weights for prey in order to stochastically emphasize ($C>1$) or deemphasize ($C<1$) the effect of prey in defining the distance [1]. More information on the algorithmic procedure of GWO one can find in [1].

III. BENCHMARK PROBLEMS

We tested these described algorithms on four unimodal and multimodal nonlinear functions. They are chosen in order to reflect different sorts of real world optimization problems. The goal in each case is to determine optimum value of function, which varies depending on the chosen function. The mathematical description of the functions is given in Table I.

TABLE I.
MATHEMATICAL DESCRIPTION OF NONLINEAR FUNCTIONS

Function ID	Mathematical description	Type
F1	$f_1(x)=\sum_{i=1}^{n} x_i^2$	Unimodal
F2	$f_3(x)=-\cos(x)\cdot\cos(y)\cdot e^{\left[-\left((x-\pi)^2+(y-\pi)^2\right)\right]}$	Unimodal
F3	$f_2(x)=\sum_{i=1}^{n-1}\left[(x_i-1)^2+100\left(x_{i+1}-x_i^2\right)^2\right]$	Multimodal
F4	$f_4(x)=-\sum_{i=1}^{n}\sin(x_i)\cdot\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{20}.$	Multimodal

Visual representations of these functions are given on Fig. 1 - Fig. 4. One can note the nonlinear nature of these functions, which makes them a fairly demanding problems for optimization.

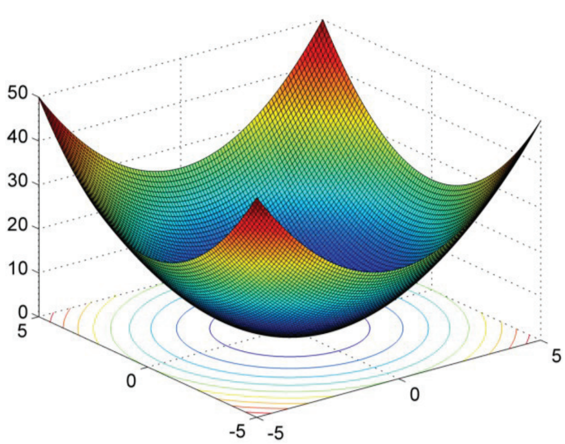

Figure 1. Sphere function (F1) in two dimensions.

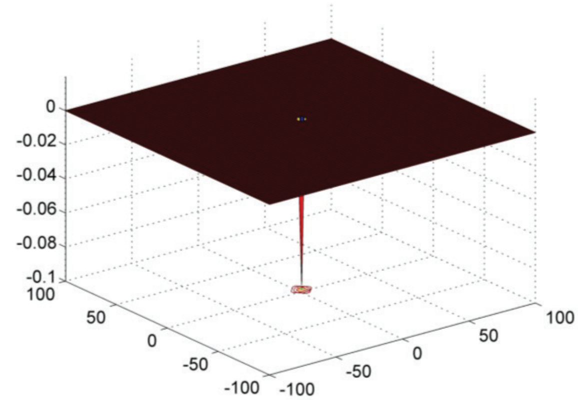

Figure 2. Easom function (F2) in two dimensions.

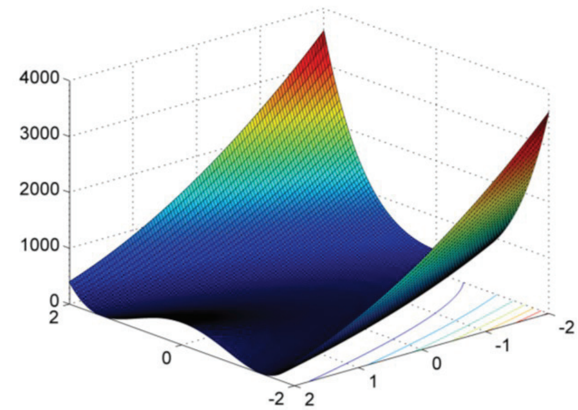

Figure 3. Rosenbrock function (F3) in two dimensions.

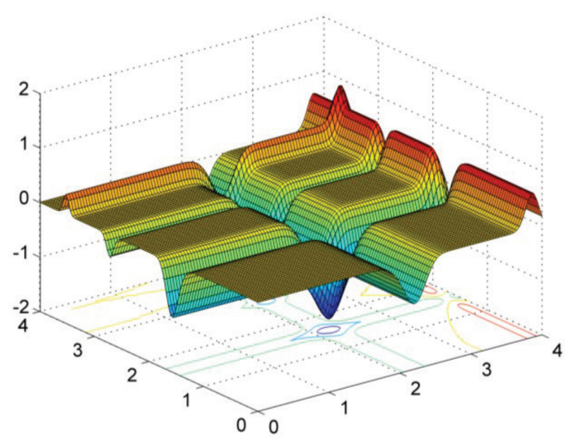

Figure 4. Michalewicz function (F4) in two dimensions.

IV. COMPUTATIONAL RESULTS

Experimental results for each described algorithm are given in this section. Reported results is derived using Matlab software using laptop PC with 4GB of RAM that runs on 64-bit Windows 7. Maximum number of iterations and algorithms specific parameters are chosen to be recommended values. Each algorithm is tested 30 times for each optimization problem in order to obtain a statistical evaluation of method's performance. The initial population is scattered across the search space at the beginning of each population. The algorithm dependent parameters are given in Table II. Fig. 5 - Fig. 8 show top view of these functions with swarm individuals dispersed across the entire surface. Finally, Table III - Table V present the computational results, in which best result refers to the determined optimum value of a particular metaheuristic algorithm for a given function.

TABLE II.
ALGORITHM DEPENDENT PARAMETERS

Algorithm	Specific parameter values
APSO	$\beta = 0.5$; $\alpha = 0.7^i$, i=*num_generation*
FA	$\beta_0 = 1$; $\gamma = 1$; $\alpha = 0.2$
GWO	$a = 2 - iter \cdot ((2)/Max\_iter)$ r_1, r_2 - *random numbers in* $[0,1]$

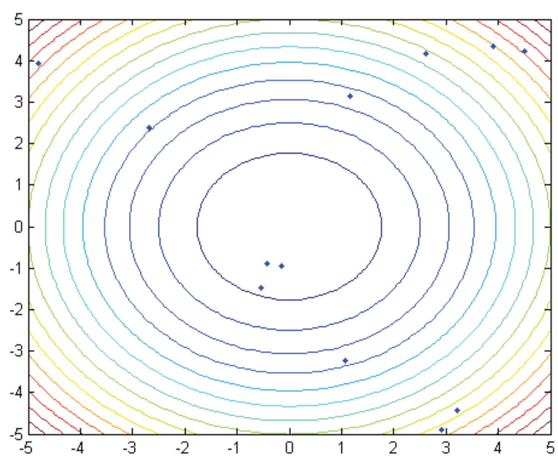

Figure 5. Initial population over the surface of Sphere function (top view).

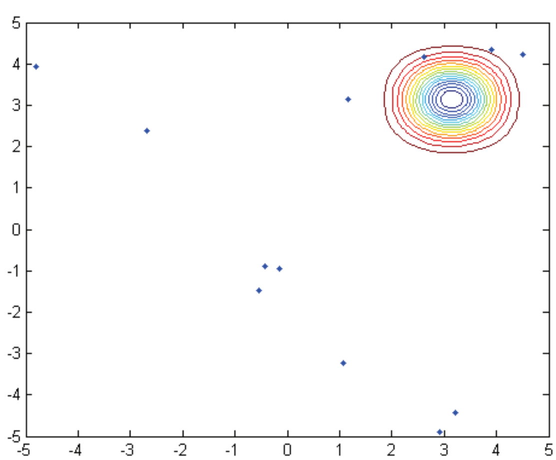

Figure 6. Initial population over the surface of Easom function (top view).

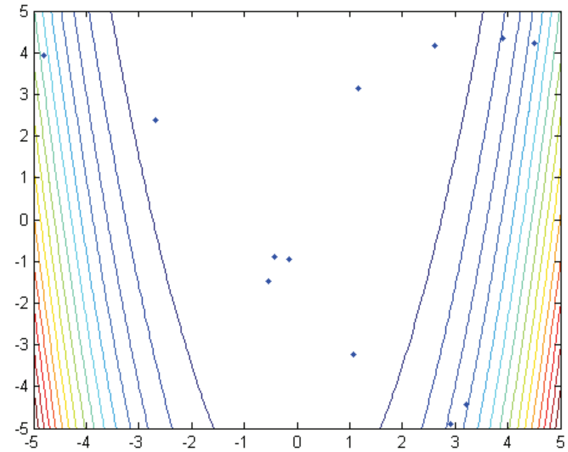

Figure 7. Initial population over the surface of Rosenbrock function (top view).

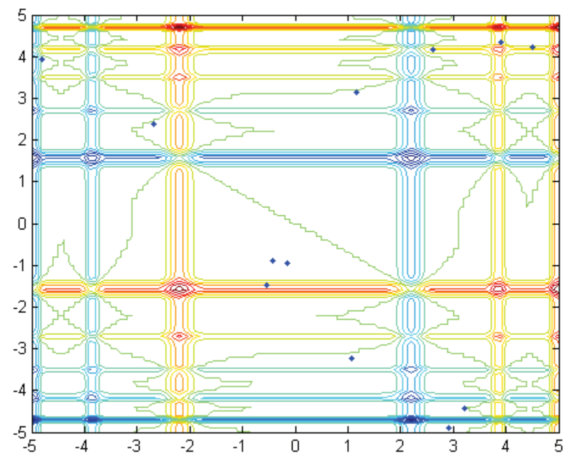

Figure 8. Initial population over the surface of Michalewicz function (top view).

TABLE III.
COMPUTATIONAL RESULTS FOR ACCELERATED PARTICLE SWARM OPTIMIZATION

Function ID	Best result	Mean value	Convergence success rate
F1	0	$6.8777*10^{-6}$	100%
F2	-1	-1	100%
F3	$-1.7909*10^{-5}$	0.1339	63.33%
F4	-1.8011	-1.7547	90%

TABLE IV.
COMPUTATIONAL RESULTS FOR FIREFLY ALGORITHM

Function ID	Best result	Mean value	Convergence success rate
F1	0	0.0419	100%
F2	-0.9748	-0.9748	100%
F3	0.5913	0.5913	0%
F4	-1	-1	0%

TABLE V.
COMPUTATIONAL RESULTS FOR GREY WOLF OPTIMIZER

Function ID	Best result	Mean value	Convergence success rate
F1	0	10^{-10}	100%
F2	-1	-0.9999	100%
F3	0	0	100%
F4	-1.8013	-1.8012	100%

Comparing Table III, Table IV and Table V one can conclude that GWO algorithm obtained best results. Convergence rate also indicate the superiority of GWO in comparison with other methods. The convergence is determined experimentally using ""convergence to an optimum criteria" [11], which indicate that the algorithm successfully converges when its best solution reaches ε likelihood of theoretical optimum for a given problem. Therefore, it can be concluded that GWO is the most reliable of all tested metaheuristic methods.

GWO algorithm also show optimal performance in terms of best obtained results and also mean value over 30 independent runs. This corresponds to some other optimization studies that indicate the same conclusion [1]. Second best result show APSO, while FA gives overall worst result. However, it should be noted that these problems are fairly hard to solve, and that both APSO and FA show better performance in optimizing real engineering problems [10,11,12,13,14].

Future research will include more bioinspired algorithms which will be tested on larger number of functions. Likewise, comparison of these intelligence methods should be studied on real world engineering problems; for example, metaheuristic methods can be employed for intelligent transportation in indoor environment using nonholonomic differential drive mobile robots [6].

V. CONCLUSIONS

In this paper, a concise experimental comparison study on bioinspired algorithms on function optimization is given. The study includes three newly developed metaheuristic techniques, namely Accelerated Particle Swarm Optimization (APSO), Firefly Algorithm (FA), and Grey Wolf Optimizer (GWO). These intelligent techniques are tested in optimization task of four different unimodal and multimodal functions. Each algorithm is tested 30 times on each optimization problem in order to obtain statistical evaluation of the experiments. Computational results show that GWO algorithm is the best one, and that it successfully converge in each simulation run with rate of 100% for all optimization tasks.

ACKNOWLEDGMENT

This work is supported by the Serbian Government - the Ministry of Education, Science and Technological Development - Project title: An innovative, ecologically based approach to the implementation of intelligent manufacturing systems for the production of sheet metal parts (2011-2015) under grant TR35004.

REFERENCES

[1] S. Mirjalili, S.M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Soft.*, vol. 69, pp. 46-61, 2014.

[2] X. S. Yang, "Firefly algorithms for multimodal optimization," in: *Stochastic algorithms: foundations and applications*, Springer Berlin Heidelberg, pp. 169-178, 2009.

[3] X. S. Yang, and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. Math. Model. Num. Opt.*, vol. 1, pp. 330-343, 2010.

[4] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in: *Nature inspired cooperative strategies for optimization* (NICSO 2010), pp. 65-74, 2010.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in: *IEEE/RSJ International Conference on Neural Networks*, pp. 1942-1948, 1995.

[6] M. Mitić and Z. Miljković, "Bio-inspired approach to learning robot motion trajectories and visual control commands," *Expert Syst. Appl.*, vol. 42, pp. 2624 - 2637, 2015.

[7] G. Wang, L. Guo, H. Duan, L. Liu and H. Wang, "A bat algorithm with mutation for UCAV path planning," *Sci. World J.*, pp. 418946, 2014.

[8] S. Karthikeyan, P. Asokan and S. Nickolas, "A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints," *Int. J. Adv. Manuf. Tech.,* vol. 72, pp. 1567-1579, 2014.

[9] M. R. Soltanpour and M. H. Khooban, "A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator," *Nonlinear Dynam.*, vol. 74, pp. 467-478, 2013.

[10] A. H. Gandomi, G. J. Yun, X. S. Yang and S. Talatahari, "Chaos-enhanced accelerated particle swarm algorithm," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 18, pp. 327–340, 2013.

[11] X. S. Yang, "Engineering optimization: an introduction with metaheuristic applications," *John Wiley & Sons*, 2010.

[12] I. Fister, I. Jr. Fister, X. S. Yang and J. Brest, "A comprehensive review of firefly algorithms," *Swarm Evol. Comput.*, vol. 13. pp. 34-46, 2013.

[13] A. H. Gandomi, X. S. Yang, S. Talatahari and A. H. Alavi, "Firefly algorithm with chaos," *Commun. Nonlinear Sci. Numer. Simulat.* vol., 18, pp. 89–98, 2013.

[14] X. S. Yang, "Nature-inspired metaheuristic algorithms," *Luniver Press*, 2008.