SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 1, ID. SAUCIS-5-2018, APRIL 2018

Nature Inspired Optimization Algorithms and Their Performance on the Solution of Nonlinear Equation Systems

***Pakize ERDOĞMUŞ*[1]**

1; Corresponding Author;
E-Mail: pakizeerdogmus@duzce.edu.tr;
Phone Number:+90 5335429728
Duzce University, Engineering Faculty, Computer Engineering, DUZCE

Received 25 March 2018; Accepted 31 March 2018; Published online 2 April 2018

Abstract

The aim of this article is both to introduce recent published nature-inspired optimization algorithms and to compare the performances of them. Four benchmark test problems(two unimodal, two multimodal) and four nonlinear equations systems were used for the comparison. The results were submitted. It was seen with these test results, we can not say that one of the algorithms outperforms. But all af them can be an alternative for solving the nonlinear equation systems.

Keywords: *Nature-Inspired Optimization Algorithms; Nonlinear Equation Systems;Grey Wolf Optimization; Ant-Lion Algorithm; Whale Optimization Algorithm; Salp Swar Algorithm*

1. Introduction

Optimization is a discipline looking for the best solution to a given problem under some constraints. Since nearly all engineering aims to find best design and production, optimization is an indispensible part of engineering. But sometimes it is difficult to find best solution to a given problem, in a limited time. So this creates a dilemma to what will be optimized.Solution time or solution? Heuristic algorithms find near optimal solutions in a reasonable time. Nature-inspired optimization algorithms are also accepted as heuristic, since they use the natural problem solution techniques. Recent studies show that nature-inspired optimization algoritms are successful to find good solutions to engineering problems. The first nature inspired algorithm is Genetic Algorithm (GA) developed by Holland. After GA, several heuristic algorithms have been developed simulating natural events, physical laws and process or foraging behavior of animals. Particle Swarm Algorithm (PSO) is one of the nature inspired algorithm simulating the bird flocking and fish schooling foraging behaviours. Since then, optimization algorithms inspired from nature gained popularity and were applied for the solution of different real-life engineering problems.

One of the recent studies on optimization is the power allocation and trajectory optimization problem for unmanned aerial vehicles. Two problems are nested; Transmit power optimization with given trajectory and trajectory optimization with given transmit power (Wang et al. 2018). Another study is on a wireless powered sensor network (WPSN), where sensors harvest energy from a hybrid access point (H-AP) and transmit information to the H-AP. For the energy efficiency (EE) maximization problem, particle-swarmoptimization-based solution algorithm is proposed (Song et al. 2018). Given a communication network, minimizing the end-to-end communication delay is an optimization task with applications in multiple domains. A greedy heuristic algorithm is proposed to solve the general problem (Medya et al. 2018). High-dimensional data are ubiquitous in many areas of machine learning, signal and image processing. Sparce subspace clustering for these data is another optimization task. With clustering, recovering low-dimensional structures in the data reduce the computational cost and memory requirements of algorithms (Elhamifar et al. 2013). For salient object detection, a principled optimization framework is proposed and taken efficient results on several benchmark datasets (Zhu et al. 2014). Another recent study is on memory optimization. Memory is now a very significant role in data processing and with the rapid development of the Internet-related technologies, such as cloud computing, big data, and 5G networks. Phase-change memory (PCM) is one of the most promising alternative techniques to the dynamic random-access memory (DRAM) that faces the scalability wall. A genetic-based optimization algorithm for chip multiprocessor (CMP) equipped with PCM memory in green clouds is proposed (Qiu et al. 2015)

Since the 70's, heuristic optimization algorithms have gained popularity, due to the computer technology. Another reason is that some real-life optimization problems are quite difficult to solve with classical methods in a reasonable solution time. Near optimal solutions have been started to be accepted. Starting with Genetic Algorithm, the inspiration from nature has successfully solved the difficult optimization problems. According to the No Free Lunch Teorem(Wolpert et. al 1997), there is not only one algorithm solving each type of optimization problems successfully. So today it has been developed a lot of different optimization algorithms. Each of them solves specific types of optimization problems.

Crow Search Algorithm(CSA) (Askarzadeh 2016), Runner Root Algorithm(RRA)(Bayat 2015), Ant Lion Optimizer(ALO)(Mirjalili 2015),Water Cycle Algorithm(WCA)(Eskendar et al. 2012), The Whale Optimization Algorithm(WOA)(Mirjalili et al. 2016), Grey Wolf Optimizer(GWO)(Mirjalili et al. 2014),Monarch Butterfly Optimization(MBO)(Wang et al. 2015), Moth Flame Optimization(Mirjalili 2015), Wind Driven Optimization(WDO)(Bayraktar et al, 2010), Biogeography Based Optimization(BBO)(Simon 2008), Selfish Herd Optimization(SHO)(Fausto et al. 2017), Salp Swarm Algorithm(SSA)(Mirjalili et al. 2017), Ideology Algorithm(IO)(Huan et al. 2017) and Cohort Intelligence(CI)(Kulkarni et al. 2017) are recent developed nature-inspired optimization algorithms. In this study, ALO, WOA, GWO and SSA have been selected for the comparison.

Nonlinear Equation Systems (NES) are quite common in Electrical, Chemical and Mechanical Engineering problems. Mostly the solution of NES is difficult, because the classic solution of NES requires matrix operations and a good initial solution. NES involving 50 or more equations are difficult to solve unless a good estimate is available before iteration. However, this is only true for systems in which the Jacobian matrix is filled, or nearly so, with nonzero elements. As the matrix sizes increase, the number of nonzero elements decreases (Dennis et al. 1983). In this study NES is accepted as optimization problem and solved with some recent nature-inspired optimization algorithms.

The rest of the paper is organized as follows. Section 2 presents the selected Nature-Inspired algorithms. Section 3 presents the solution of optimization problems with Nature-Inspired Algorithms. Finally, Section 4 concludes the study.

2. Nature-Inspired Optimization Algorithms

In this study, it has been studied with four different optimization algorithms inspired from nature. So, all of them have common properties. They have been designed for the solution of contionus optimization problems, they are population-based and they mimic the hunting behavior of animals. The citation statistic of the algorithms is given in Table 1.

Algorithms start with random initial positions. In this study, the objective function dependent to d independent variables is called f as given in Equation 1.

$$Fobj=f(x_1, x_2, \ldots x_d) \tag{1}$$

Since, nature-inspired algorithms in the article simulate the foraging behaviors of animals, they have n cooperative agents in the population. So, these algorithms are called population-based optimization algorithms. Each agent is called particle or individual. It is assumed that there are n particles. It means that the algorithm starts with n initial solutions as given in Equation 2 and 3.

Table 1. The citation statistic of the Latest Nature Inspired Algorithms (02/2018)

Algorithm	Publication Year	Scholar Citation	Web of Science Citation
Grey Wolf Optimizer	2014	902	420

Ant Lion Optimization	2015	276	142
Whale Optimization Algorithm	2016	208	69
Salp Swarm Optimization	2017	6	0

$$x_{initial} = \begin{bmatrix} x_{1_{init},1} & \cdots & x_{n_{init},1} \\ \vdots & \ddots & \vdots \\ x_{1_{init},d} & \cdots & x_{n_{init},d} \end{bmatrix} \quad (2)$$

$$F_{initial}=\{F_{1\_init},F_{2\_init},\ldots\ldots.F_{n\_init}\} \quad (3)$$

So, nxd values are created as given in Equation 2 for the initial solution. After initial solution is created, initial objective function values are evaluated for the fitness. The $x_1, x_2, \ldots x_d$ values are accepted the positions of the solution i. The next positions of the particles are calculated with a formula, related to the fitness of the particles as given in Equation 1. So, n initial solution fitness values are found as given in Equation 3. The local and global best positions are evaluated for the next position. From one iteration, to another, particles converge the position of optimum value of the objective function. Since the algorithms simulate the hunting behaviors of the animals, preys or foods are aceepted as the optimum point. Initial solutions for Rastrigin test function is given in Figure 1. In general, initial solutions are created randomly. But in the Figure 1, it is given linearly for visually spread.

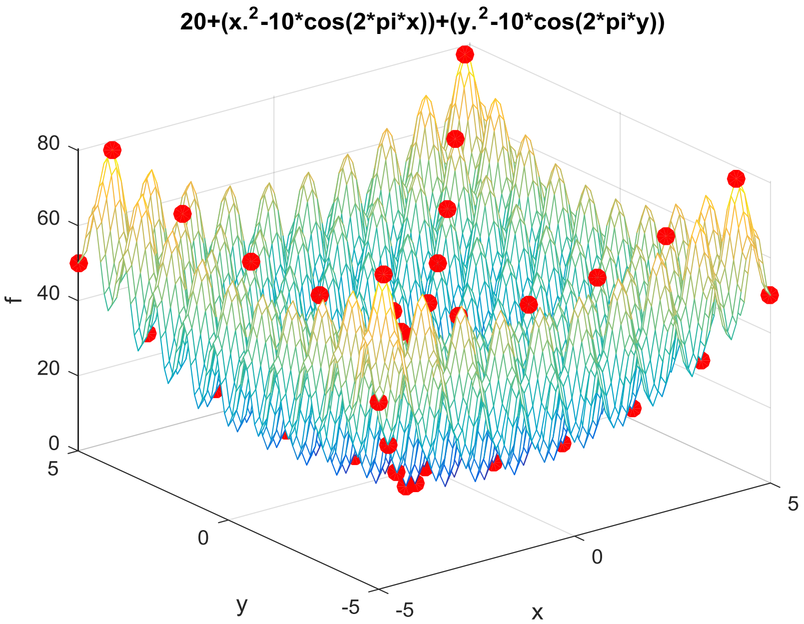

Figure 1. Rastrigin test function and initial solutions for Nature-Inspired algorithms

2.1. Grey Wolf Optimizer

GWO, is developed in 2014 by Mirjalili (Mirjalili et al. 2014). GWO simulates the hunting behavior of Grey Wolves. GWO is applied for the solution of lot of real-life optimization problem. Optimizing hybrid renewable energy systems (Yahiaoui et. Al 2017), economic dispatch (Naderi et. Al 2017) and sensor node localization (Kaur et. al 2017) have been solved with GWO.

A Grey wolf is seen in Figure 2. Grey wolves live as social groups and there is a hierarchy among them. They are classified as alpha, beta, delta and omega. Their hunting strategy can be summarized with three steps as tracking, enrcircling and attacking. After tracking the prey, they encircle the prey until the prey stops moving and at last attack towards the prey.

Figure 2. A Grey Wolf (Canis Lupus)(Cranshaw et al. 2018)

Prey represents the optimal solution, while each wolf represents a solution. Wolves change their position in such a way that they converge the optimal solution. For the encircling behavior of wolves, Equation 4 and 5 are proposed.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \quad (4)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}\vec{D} \quad (5)$$

X represents the position of grey wolf; Xp represents the position of prey, $\vec{A}$ and $\vec{C}$ are the coefficients, t represents current iteration and t+1 represents the next iteration. $\vec{A}$ and $\vec{C}$ are given in Equation 6 and 7.

$$\vec{A} = 2\vec{a}.\vec{r_1} - a \quad (6)$$

$$\vec{C} = 2\vec{r}_2 \quad (7)$$

a is decreased linearly from 2 to 0, r_1, r_2 are random values between 0-1. The real wolves see their prey but in GWO it is not possible. Since there is not a priory knowledge about the optimum point(prey) in the solution space, the hunting behavior is simulated with the three best solution found among the wolves. So, Equations between 8-14 are proposed for hunting behavior.

$$\overrightarrow{D_\alpha} = \left|\vec{C}_1.\overrightarrow{X_\alpha} - \vec{X}\right| \quad (8)$$

$$\overrightarrow{D_\beta} = \left|\vec{C}_2.\overrightarrow{X_\beta} - \vec{X}\right| \quad (9)$$

$$\overrightarrow{D_\delta} = \left|\vec{C}_3.\overrightarrow{X_\delta} - \vec{X}\right| \quad (10)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}(\overrightarrow{D_\alpha}) \quad (11)$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}(\overrightarrow{D_\beta}) \quad (12)$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}(\overrightarrow{D_\delta}) \quad (13)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \quad (14)$$

Attacking prey is simulated decreasing the value of a. The pseudocode of the GWO is given in Figure 3.

```
Initialize the Grey Wolf Population
Initialize parameter A,a,C
Calculate each wolf fitness value
Specify first,second and third best solutions
while (t< max_iteration)
     for each wolf
              Update the position
     EndFor
     Update a,A,C
     Update fitness of each wolf
    Update first, second and third best soluitons
     t=t+1;
EndWhile
```

Figure 3. The Pseudocode of GWO

2.2 Ant Lion Optimizer

Ant Lion Optimizer (ALO) is one of the recent nature-inspired optimization algorithm developed by Mirjalili especially for the solution of continuous optimization problems in 2015 (Mirjalili 2015). Since then, different types of engineering problems have been solved with ALO. Automatic generation control of multi-area system using ALO based PID control (Raju et al. 2016), Feature selection for machine learning (Emary et al. 2016), optimal power flow with enhancement of voltage stability and reduction of power loss, solution of non-convex economic load dispatch problem for small-scale power systems, route planning for unmanned aerial vehicle, community detection in complex network are some of the optimization problems solved successfully with ALO.

Antlions are accepted as insects. Their names come from their preys. Their life span are nearly three years and they pass most of their lifes as larvae. ALO is inspired from the hunting behaviour of antlions. It has seen antlion larvaes in Figure 4(a) and dig cone-shaped pits in sand by moving along a circular path and throwing out sands in Figure 4(b). (Cranshaw et al. 2018)

a)AntLion Larvae b)Cone-shaped pits

Figure 4. The antlion larvae and its cone-shaped traps for ants

After they build cone-shaped pits, they waits ants to fall this trap. After ant got trapped this pit, the ant lion catchs and hunts it.

The ALO mimics interaction between antlions and ants. So artificial ants move over the search space, and antlions are allowed to hunt them. Since ants move stochastically in nature when searching for food, a random walk is chosen for modelling ants' movement. In this algorithm, two initial matrix with nxd dimension are created both for ants and antlions as given in Equation 2. The nxd matrix belong to the ants save the positions of n ants. The fitness of each ants are saved in a nx1 vector.

The antlions are also hiding somewhere in the search space. In order save their positions and fitness values, two matrix nxd and nx1 are used. The nxd matrix belong to the antlions save the positions of n antlions. The fitness of each antlions are saved in a nx1 vector. In ALO, each antlions match an ant. While ants mobile in the search space, antlions are immobile. Ants random walks are updated with Equation 15.

$$X_i^t = \frac{(X_i^t - a_i) * (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{15}$$

a_i and b_i are minimum and maximum of random walk of i*th* variable respectively, c_i^t and d_i^t are minimum and maximum of random walk of i*th* variable for the iteration t.

$$c_i^t = \mathrm{Antlion}_t^j + c^t \tag{16}$$

$$d_i^t = Antlion_t^j + d^t \tag{17}$$

c and d are updated with the Equation 18 and 19 respectively.

$$c^t = \frac{c^t}{I} \tag{18}$$

$$d^t = \frac{d^t}{I} \tag{19}$$

Random walks are affected by the traps of antlions. Antlions build pits according to their fitness values. Random walks are decreased from iteration to iteation. If an ant' fitness better than an antlion, then ant is caught by the antlion. After eating this ant, antlion updates its position in order to increase its chance of catching new prey. Elitism is used like GA. In this study the best antlion obtained so far in each iteration is saved and considered as an elite. The pseudocode of antlion is given in Figure 5.

```
Initialize the n ants' and Antlions' positions
Calculate the fitness values of them
Find elite antlion
While stopping conditions not satisfied
    for each ant
        Select an antlion using roulette wheel
            Update the position
    EndFor
    Calculate the fitness of all ants
    Replace an antlion with its corresponding ant it if becomes fitter
    Update elite if an antlion becomes fitter than the elite
EndWhile
Return Elite
```

Figure 5. The pseudocode of ALO

2.3. Whale Optimization Algorithm

WOA simulates the hunting behaviour of humpback whales developped by Mirjalili in 2016 (Mirjalili 2016). Even if formulations of WOA is quite similar with GWO, use of a spiral to simulate bubble-net attacking mechanism of humpback whales and using only one global best are the main differences of WOA.

Even if, we, human being accept ourselves as the most intelligent creatures in the World, most of the animals show intelligent behaviours. Whales are one of them. They are mammals. Since they breath, they never sleep. They have some emotional behaviours. They can improve a dialect between them. Foraging behaviour of humpback whales are quite interesting. They hunt their preys near to surface of the water, creating bubbles. A humpback whale is seen in the Figure 6(Iliya, 2018)

Figure 6. Humpback Whale

Humpback whales' preys are small fish herds and krills. Their hunting behaviour consist of three strategy; encircling prey, spiral bubble-net feeding maneuver, and search for prey. They recognize their prey, create bubbles as sprially and encircle them like wolves. Like GWO, the prey is accepted best optimal solution, since the global optimum is not known a priori. The positions of population are updated towards the best optimal solution as given in Equation 20 and 21.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_*}(t) - \vec{X}(t)\right| \tag{20}$$

$$\vec{X}(t+1) = \overrightarrow{X_*}(t) - \vec{A}\vec{D} \tag{21}$$

D is the absolute distance, between the prey and whale. A and C are coefficent vectors calculated with Equation 22 and 23, t and t+1 are current iteration and the next iteration respectively. X is the position vector of whales, X_* is the position of best solution attained up to this iteration and updated for each iteration. a is a value, starting from 2, decreased to 0, which simulates a shrinking behaviour of encircling. r is a random vector.

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - a \tag{22}$$

$$\vec{C} = 2\vec{r}_2 \tag{23}$$

The bubble-net strategy of whales is also simulated in the algorithm as given in Equation 24,25. In the algorithm both strategy is used with %50 percent of possibility, given in Equation 26.

$$D' = |X_*(t) - X(t)| \tag{24}$$

$$X(t+1) = D'^{e^{bl}} \cos(2\pi l) + X_*(t) \tag{25}$$

$$X(t+1) = \begin{cases} \overrightarrow{X_*}(t) - \vec{A}\vec{D} & \text{if p} < 0.5 \\ D'^{e^{bl}} \cos(2\pi l) + X_*(t) & p > 0.5 \end{cases} \tag{26}$$

Both shrinking enricling and spiral bubble-feeding is for exploitation. And some random movements are also accepted in the algorithm for exploration as given in Equation 27, 28.

$$\vec{D} = \left|\vec{C}.\vec{X}_{rand}(t) - \vec{X}(t)\right| \tag{27}$$

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{A}\vec{D} \tag{28}$$

The pseudocode of the algorithm is given in Figure 7.

```
Create positions for  n initial solutions
Calculate the fitness
Find best optimal value as X*
While termination criteria not satisfied
   For each whales do
       Update a,A,C,l,p  for each whale
        If p<0.5
              If |A|<1
                    Update the current position with Equation 20
             else(|A|>1)
                  Update the current position with Equation 27
             end
       else(p>0.5)
               Update the current position with Equation 25
        end
     end
  Update positions and  best solution X*
End
```

Figure 7. The pseudocode of WOA

2.4. Salp Swarm Algorithm

SSA is one of the recent nature inspired optimization algorithm, developed by Mirjalili (Mirjalili et. al 2017) for the solution of continuous optimization problems. Since SSA has only one parameter to adapt it is quite easy to implement. Salps are quite interesting creatures looking like jellyfish. But they are evolutionarily very different from jellyfish. They have complex nervous and digestive systems with a brain, heart, and intestines. SSA simulates the salps foraging behaviours. Since the algorithms is quite new, there a few study using this algorithm. A salp chain can be seen in Figure 8(Richard Herman, 2018)

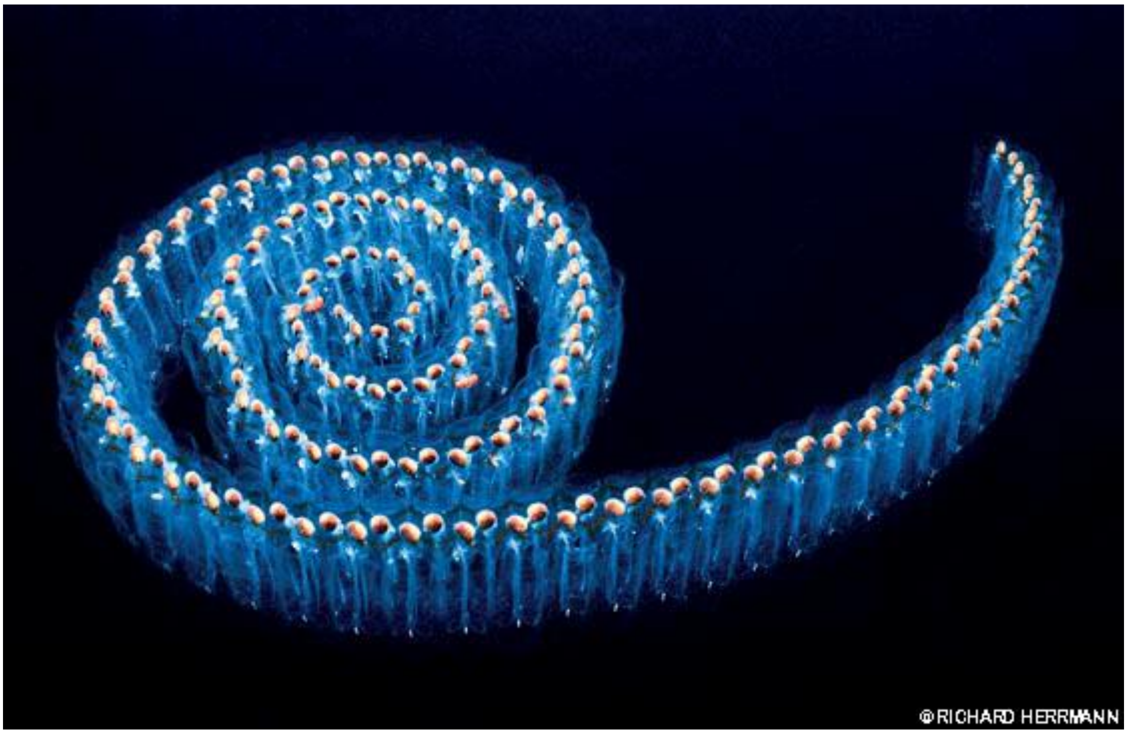

Figure 8. Salp Chain

Salps generally create chains and move cooperatively. In SSA, the salps are classified as two grourp: leader salp and followers. The followers follow the leader salp' position. The position of leader salp is updated with the equation 29.

$$x_j^1 = \begin{cases} F_j + c_1\left(ub_j - lb_j\right)c_2 + lb_j;\ if\ c_3 \geq 0 \\ F_j - c_1\left(ub_j - lb_j\right)c_2 + lb_j;\ if\ c_3 < 0 \end{cases} \tag{29}$$

x_j^1 shows the position of the leader salp in the j th dimension, F_j is the position of the food source in the j. dimension, ub_j and lb_j represents the upper bound and the lower bound of j. dimension, c_1 is calculated with equation 30 c_2, and c_3 are random uniform numbers between 0 and 1.

$$c_1 = 2e^{-(\frac{4l}{L})^2} \tag{30}$$

l is the current iteration, L is the maximum iteration. The follower salps update their positions with equation 31.

$$x_j^i = \frac{1}{2}at^2 + v_0 t \tag{31}$$

t shows time, but in the simulation t shows iteration, v_0 shows the initial velocity, a; shows the acceleration, which is given with equation 32. Since the difference between the iteration Δt is 1.

$$a = \frac{\Delta v}{\Delta t} = \frac{v_{final} - v_0}{1} \tag{32}$$

V_{final} is calculated with equation 33.

$$v_{final} = \frac{x - x_0}{1} \tag{33}$$

If a is replaced in the equation, the positions of the follower salps can be found with the equation 34.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{34}$$

The basic pseudocode of SSA is given in the figure 9.

```
Create initial positions for n salp considering lb,ub
While termination criteria not satisfied
      Calculate the fitness of each salp
       Find best optimal value as F
       Update c1
      For each salps do
          if salp is leader
               Update leader salp position with Equation 29
         else
              Update the current position of follower salp with Equation 34
           end
       end
  check all the salps for lb,ub
end
```

Figure 9. The pseudocode of SSA

3. Solution of the optimization problems with Nature-Inspired Optimization Algorithms

A mathematical formulation of an optimization problem has three main elements: Variables, constraints and objective function. Optimization problems are classified according to these three elements. According to the variables, it can be classified as single-variable or multi-variable optimization problems. It can also be classified discrete or continuous optimization problems according to the variable type. Unconstrained optimization problems are much easier then constrained optimization problem. And another classification is based on the objective function. Optimization problems are classified as multi-objective or single-objective optimization problems.

In this study, one of the objectives is to compare the performances of the four recent nature-inspired optimization algorithms on NES. General representations of NES are given in equation 35.

$f_1(x_1, x_2, \dots, x_n) = 0,$

$f_2(x_1, x_2, \dots, x_n) = 0\ ,$

………………………..

$f_n(x_1, x_2, \dots, x_n) = 0$ (35)

f_1, f_2 and f_n are nonlinear functions dependent to the variables x_1, x_2 and x_n. Some classical and heuristic solution methods have been proposed for the solution of the nonlinear equations system in the literature. Classical methods are generally use the initial estimates and try to solve the system. Since most of the classical methods require derivative and matrix operations, the complexity of the solution for a nonlinear equation system increases with the number of independent variable and the number of equation. So, some heuristic methods are used for the solution of nonlinear equation systems (Jaberipour et. al 2011 and Joshi et. al 2014)

NES can be modelled as single objective unconstrained, multi-objective unconstrained and single-objective constrained optimization problem. In this study, NES are modeled as single objective unconstrained optimization problem, summing the squares of all equation as given in Equation 36.

Objective function : $fmin = \sum_{i=1}^{n} f_i\ (x_1, x_2, \dots, x_n)^2$ (36)

In this study, simulations were made with a computer using Intel® Core™ i5 3230M CPU 2.6GHz processor and Microsoft Windows 7 operating system. Firstly two unimodal and two multimodal optimization test problems were solved with four algorithms. Veryfying the results all simulations were run 30 times. The maximum number of iterations for each run is 1000. The number of agents is accepted as 10 for unimodal and multimodal problems and 30 for NES.

Unimodal and multimodal test problems are given in Table 2. The optimum values of the test problems found with four algorithms were given in Table 3.

Then four NES are given in Table 4 and the solutions found with four algorithms are given in Table 5.

Table 2. Unimodal and multimodal test problems

Test Function	**Uniodal /Multimodal**	**dim**	**range**	$\mathbf{f_{min}}$
$f_3(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	Unimodal	10	$[-100{,}100]$	0
$f_6(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	Unimodal	10	$[-100{,}100]$	0
$f_8(x) = \sum_{i=1}^{n} -x_i\, sin\left(\sqrt{\lvert x_i \rvert}\right)$	Multimodal	10	$[-500{,}500]$	-418.9829×5
$f_9(x) = \sum_{i=1}^{n}\left[x_i^2 - 10cos(2\pi x_i) + 10\right]$	Multimodal	10	$[-5.12, 5.12]$	0

Table 3. Unimodal and multimodal test problems optimum values found four nature-inspired optimization algorithms

	Average Optimum values and Standart Deviation of the Optimum values							
Test function	**GWO**		**ALO**		**WOA**		**SSA**	
	Average Optimum	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**

F3	**2,2768E-31**	**4,42587E-31**	0,871253	1,52797	663,3658	726,1636	2.431e-09	1.8138e-09
F6	0,091457	0,12689	1,86275E-08	2,18355E-08	0,10565	0,113055	**1,187E-09**	**4,409E-10**
F8	-2505,64	244,7711	**-2147,6**	**387,2083**	-3173,39	532,4177	-2635,65	371,0163
F9	**1,1842E-15**	**4,10232E-15**	24,45498	10,70219	1,58E-15	6,6990E-15	28,07988	7,239577

Table 4. The Systems of Nonlinear Equations used in this study

Test problem Name and Reference	**The system of Nonlinear Equation**	**The number of independent variable**	**Decision Space**
P1	$x_1^2+2x_2^2+\cos(x_3)-x_4^2=0$ $3x_1^2+x_2^2+\sin^2(x_3)-x_4^2=0$ $-2x_1^2-x_2^2-\cos(x_3)+x_4^2=0$ $-x_1^2-x_2^2-\cos^2(x_3)+x_4^2=0$	4	$[-2,2]^4$
P2 Aritmethic Benchmark	$x_1-0.25428722-0.18324757x_4x_3x_9=0$ $x_2-037842197-0.16275449x_1x_{10}x_6=0$ $x_3-0.27162577-0.16955071x_1x_2x_{10}=0$ $x_4-0.19807914-0.15585316x_7x_1x_6=0$ $x_5-0.44166728-0.19950920x_7x_6x_3=0$ $x_6-0.14654113-0.18922793x_8x_5x_{10}=0$ $x_7-0.42937161-0.21180486x_2x_5x_8=0$ $x_8-0.07056438-0.17081208x_1x_7x_6=0$ $x_9-0.34504906-0.19612740x_{10}x_6x_8=0$ $x_{10}-0.42651102-0.21466544x_4x_8x_1=0$	10	$[-2,2]^{10}$
P3 Neuro-physology Application	$x_1^2+x_3^2-1=0$ $x_2^2+x_4^2-1=0$ $x_5\, x_3^3+x_6\, x_4^3=c_1$ $x_5\, x_1^3+x_6\, x_2^3=c_2$ $x_5\, x_1\, x_3^2+x_6\, x_4^2\, x_2=c_3$ $x_5\, x_1^2\, x_3+x_6\, x_2^2\, x_4=c_4$ $c_i=0$, $i=1,..4$	6	$[-1,1]^6$
P4 Chemical Equilibrium	$x_1\, x_2 +x_1-3x_5 =0$ $2x_1x_2+x_1+x_2\, x_3^2 +R_8x_2 - Rx_5+$ $2R_{10}x_2^2+R_7\, x_2\, x_3+ R_9\, x_2\, x_4=0$ $2x_2\, x_3^2 + 2R_5\, x_3^2 -8\, x_5 +R_6\, x_3$ $+R_7\, x_2\, X_3=0$ $R_9\, x_2x_4+2x_4^2 – 4Rx_5=0$ $x_1(x_2 +1)+ R_{10}\, x_2^2$ $+ x_2\, x_3^2+ R_8\, x_2 +R_5\, x_3^2$ $+x_4^2-1 +R_6\, x_3 + R_7\, x_2x_3+ R_9x_2x_4=0$ R values can be found(Grosan et. al. 2008)	5	$[-40,40]^5$

Table 5. NES solutions found with four nature-inspired optimization algorithms

	Average Optimum values and Standart Deviation of the Optimum values							
Test function	**GWO**		**ALO**		**WOA**		**SSA**	
	Average Optimum	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**	**Average Optimum**	**Standart Deviation of Opt.**
P1	0,704194	2,13E-05	0,704164	4,72414E-14	0,70422	8,6478E-05	**0,704164**	**5,0927E-14**
P2	0,017619	0,062108	9,61E-13	3,457E-13	2,31188E-05	1,7095E-05	**2,8982E-13**	**7,2464E-14**
P3	6,37E-06	1,03E-05	**8,91471E-13**	**3,79967E-12**	7,24871E-05	0,000118	2,1671E-10	9,4456E-10
P4	0,197969	0,174349	0,001603	0,004095	**0,000987**	**0,002104**	0,001344	0,003641

4. Conclusion

In this study four recently nature-inspired optimization algorithms developed by Mirjalili have been introduced and they have been tested both some unconstrained test problems and NES. Nonlinear Equation Systems can be accepted as unconstrained optimization problems, taking the sum of squares of each function. Since the main focus is to compare the algorithms performance in view of optimality, each equation value in the NES is not considered.

As a result, since these algorithms use similar formulations and ideas, there is no big difference among the optimum values found with the algorithms. But as it can be seen from the tables, since the sum of squares are approaching zero, these optimization algorithms can be an alternative to the solution of NES.

Acknowledgments

Pakize Erdogmus was born in Erzurum in 1972. She received the B.S. in Electronic and Communication Engineering from Yildiz Technical University, Kocaeli Engineering Faculty in 1993, and M.S. and Ph.D degree Computer Sciences and Numerical Methods from Ataturk University in 1997, 2003 respectively.

From 2003 to 2010, she was Assistant Professor in Duzce University, Technical Education Faculty. From 2010 she has been working in Computer Engineering Department of Duzce University, Engineering Faculty. She is the author of 10 articles. Her research interests numerical analysis, image processing, nature-inspired optimization algorithms and their performance analysis.

References

H. Wang, G. Ren, J. Chen, G. Ding and Y. Yang, "Unmanned Aerial Vehicle-Aided Communications: Joint Transmit Power and Trajectory Optimization," in *IEEE Wireless Communications Letters*, vol. PP, no. 99, pp. 1-1. doi: 10.1109/LWC.2018.2792435

M. Song and M. Zheng, "Energy Efficiency Optimization for Wireless Powered Sensor Networks with Non-orthogonal Multiple Access," in *IEEE Sensors Letters*, vol. PP, no. 99, pp. 1-1. doi: 10.1109/LSENS.2018.2792454

S. Medya, P. Bogdanov and A. Singh, "Making a Small World Smaller: Path Optimization in Networks," in *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1-1. doi: 10.1109/TKDE.2018.2792470

E. Elhamifar and R. Vidal, "Sparse Subspace Clustering: Algorithm, Theory, and Applications," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765-2781, Nov. 2013. doi: 10.1109/TPAMI.2013.57

W. Zhu, S. Liang, Y. Wei and J. Sun, "Saliency Optimization from Robust Background Detection," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 2814-2821. doi: 10.1109/CVPR.2014.360

M. Qiu, Z. Ming, J. Li, K. Gai and Z. Zong, "Phase-Change Memory Optimization for Green Cloud with Genetic Algorithm," in *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3528-3540, Dec. 1 2015. doi: 10.1109/TC.2015.2409857

Alireza Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Computers & Structures, Volume 169, 2016, Pages 1-12, ISSN 0045-7949, https://doi.org/10.1016/j.compstruc.2016.03.001.

F. Merrikh-Bayat, The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature, Applied Soft Computing, Volume 33, 2015, Pages 292-303, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2015.04.048.

Seyedali Mirjalili, The Ant Lion Optimizer, Advances in Engineering Software, Volume 83, 2015, Pages 80-98, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2015.01.010.

Hadi Eskandar, Ali Sadollah, Ardeshir Bahreininejad, Mohd Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Computers & Structures, Volumes 110–111, 2012, Pages 151-166, ISSN 0045-7949, https://doi.org/10.1016/j.compstruc.2012.07.010.

Seyedali Mirjalili, Andrew Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, Volume 95, 2016, Pages 51-67, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2016.01.008.

Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey Wolf Optimizer, Advances in Engineering Software, Volume 69, 2014, Pages 46-61, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2013.12.007.

Wang, GG., Deb, S. & Cui, Z. Neural Comput & Applic (2015). https://doi.org/10.1007/s00521-015-1923-7

Seyedali Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-Based Systems, Volume 89, 2015, Pages 228-249, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2015.07.006.

Z. Bayraktar, M. Komurcu and D. H. Werner, "Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics," *2010 IEEE Antennas and Propagation Society International Symposium*, Toronto, ON, 2010, pp. 1-4. doi: 10.1109/APS.2010.5562213

D. Simon, "Biogeography-Based Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702-713, Dec. 2008. doi: 10.1109/TEVC.2008.919004

Fernando Fausto, Erik Cuevas, Arturo Valdivia, Adrián González, A global optimization algorithm inspired in the behavior of selfish herds, Biosystems, Volume 160, 2017, Pages 39-55, ISSN 0303-2647, https://doi.org/10.1016/j.biosystems.2017.07.010.

Seyedali Mirjalili, Amir H. Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, Seyed Mohammad Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Advances in Engineering Software, Volume 114, 2017, Pages 163-191, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2017.07.002.

Huan, T.T., Kulkarni, A.J., Kanesan, J. et al. Neural Comput & Applic (2017) 28(Suppl 1): 845. https://doi.org/10.1007/s00521-016-2379-4

Kulkarni, Anand & Krishnasamy, Ganesh & Abraham, Ajith. (2017). Socio-Inspired Optimization Using Cohort Intelligence. 114. 9-24. 10.1007/978-3-319-44254-9_2.

Dennis, J.E., Jr. and Schnabel, R.B., 1983, Numerical methods for unconstrained optimization and nonlinear equations: Englewood Cliffs, N.J., Prentice-Hall, 378 p.

More Raju, Lalit Chandra Saikia, Nidul Sinha, Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller, International

Journal of Electrical Power & Energy Systems, Volume 80, 2016, Pages 52-63, ISSN 0142-0615, https://doi.org/10.1016/j.ijepes.2016.01.037.

E. Emary, Hossam M. Zawbaa, Aboul Ella Hassanien,Binary ant lion approaches for feature selection, Neurocomputing,Volume 213, 2016, Pages 54-65, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2016.03.101.

http://www.insectsimages.org, Whitney Cranshaw, Colorado State University, Bugwood.org [Accesed: 20.02.2018]

Karim Illiya, Karim Illıya Studio,https://karimphotography.com, [Accesed: 20.02.2018]

Richard Herman, http://www.alertdiver.com/Blue_Water [Accesed: 20.02.2018]

A. Yahiaoui, F. Fodhil, K. Benmansour, M. Tadjine, N. Cheggaga, Grey wolf optimizer for optimal design of hybrid renewable energy system PV-Diesel Generator-Battery: Application to the case of Djanet city of Algeria, Solar Energy, Volume 158, 2017, Pages 941-951, ISSN 0038-092X, https://doi.org/10.1016/j.solener.2017.10.040.

Ehsan Naderi, Ali Azizivahed, Hossein Narimani, Mehdi Fathi, Mohammad Rasoul Narimani, A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm,Applied Soft Computing,Volume 61,2017,Pages 1186-1206,ISSN 1568-4946,https://doi.org/10.1016/j.asoc.2017.06.041.

Kaur R, Arora S, Nature Inspired Range Based Wireless Sensor Node Localization Algorithms, International Journal of Interactive Multimedia and Artificial Intelligence, Volume 4, Number 6, Pages 7-17, ISSN 1989-1660.

M. Jaberipour, E. Khorram, and B. Karimi, “Particle swarm algorithm for solving systems of nonlinear equations,” Comput. Math. Appl., vol. 62, no. 2, pp. 566–576, 2011.

G. Joshi and M. B. Krishna, "Solving system of non-linear equations using Genetic Algorithm," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, New Delhi, 2014, pp. 1302-1308.

C. Grosan and A. Abraham, "A New Approach for Solving Nonlinear Equations Systems," in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 3, pp. 698-714, May 2008.

D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, Apr 1997.