❒   3772

Projectile-target search algorithm: a stochastic metaheuristic optimization technique

Ayong Hiendro
Department of Electrical Engineering, Tanjungpura University, Indonesia

Article Info	**ABSTRACT**

This paper proposes a new stochastic metaheuristic optimization algorithm which is based on kinematics of projectile motion and called projectile-target search (PTS) algorithm. The PTS algorithm employs the envelope of projectile trajectory to find the target in the search space. It has 2 types of control parameters. The first type is set to give the possibility of the algorithm to accelerate convergence process, while the other type is set to enhance the possibility to generate new better projectiles for searching process. However, both are responsible to find better fitness values in the search space. In order to perform its capability to deal with global optimum problems, the PTS algorithm is evaluated on six well-known benchmarks and their shifted functions with 100 dimensions. Optimization results have demonstrated that the PTS algoritm offers very good performances and it is very competitive compared to other metaheuristic algorithms.

Corresponding Author:

Ayong Hiendro,
Departement of Electrical Engineering,
Tanjungpura University,
Jenderal Ahmad Yani Street, Pontianak,
West Kalimantan, Indonesia.
Email: ayong.hiendro@ee.untan.ac.id

1.   INTRODUCTION

Optimization techniques, especially stochastic nature-inspired metaheuristic algorithms have become the important and popular tools to deal with complex high dimensional global optimization problems in many real-life applications. The global optimization problems can be multimodal with a huge number of local optima and non-differentiable which cannot be solved by using traditional numerical optimization methods [1]. Therefore, many metaheuristic optimization algorithms have been proposed to alleviate the problems.

The nature of a stochastic metaheuristic optimization algorithm is employing the random-search mechanism to visit different parts of the search space and then approaches as close as possible to global optimum point [2, 3]. Important problem of the metaheuristic algorithms is how to increase the probability to overcome the local optima and find the better global optimum value.

In this paper, a new metaheurusthic algorithm is introduced. The proposed algorithm is based on kinematics of projectile motion. The projectile is launched from a point at the ground level with a given velocity, moved in various directions under a uniform gravity, and landed on a target at a surface. However, the proposed projectile-target search (PTS) algorithm does not emphasize on the projectile trajectory that launched from an angle. The algorithm tends to utilize the tracking of the envelope of projectile trajectory which encloses all possible points in the search space. The envelope of projectile trajectory could reach points in the search space which are out of reach from any projectiles moving with any initial points and velocities [4]. The information of the trajectory could be usefull for a projectile to hit a target from its starting point. In the proposed PTS algorithm, the beneficial property of the envelope of projectile trajectory is

applied to find a target on a linear shape surface. Futhermore, the PTS algorithm has two main control parameters in order to accelerate convergence and searching processes and hence, it can find the better global optimum value faster than any other optimization algorithms. In order to examine the general performance of the proposed algorithm, it is tested on six benchmarks and their shifted functions. The performance is compared to other algorithms' results which have been reported in [5-10].

2. PROJECTILE MODEL

The projectile is defined to be launched from a ground level (h = 0), with an initial velocity v, and at an angle of inclination θ measured with respect to the horizontal as shown in Figure 1. The path function of the projectile (y) as a function of horizontal distance (x) is specified as follows:

$$y(x) = x. \tan(\theta) - \frac{g.x^2}{2v^2} \sec^2(\theta) \tag{1}$$

where: $g = 9.81$ m/s^2

The size and shape of the projectile trajectories vary according to the launch angles at an initial velocity v above the horizontal, as seen in Figure 2. These trajectories have an envelope of projectile trajectory. The envelope of projectile trajectory (φ) is a path that encloses and intersects all possible projectile paths to find its target onto a hill. The shape of the hill surface is defined as the impact function ψ.
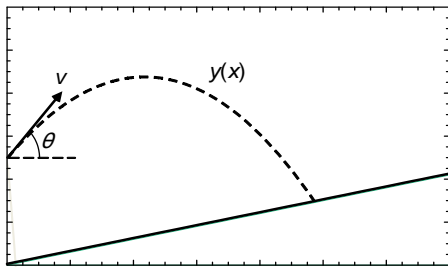

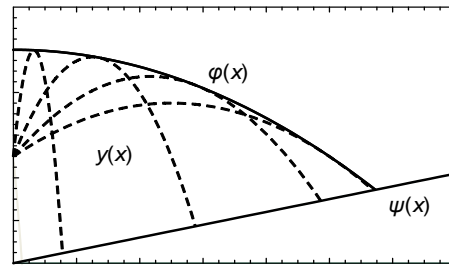Figure 1. The projectile moving path

Figure 2. The enveloping parabola path

The equation for the envelope of projectile trajectory [4, 11] is determined by

$$\varphi(x) = \frac{v^2}{2g} - \frac{g}{2v^2}x^2 \tag{2}$$

For a continuous impact function $\psi(x)$ on $0 \leq x < \infty$ with $\psi(0) = 0$, the surface of ψ and the path of φ have exactly one point of intersection. It is noted that there exists a unique target for which $\psi(x) = \varphi(x)$.

The main goal of the proposed PTS algorithm is to minimize the difference between ψ and φ in order to ensure a projectile reach the target precisely. A function $\gamma(x)$ is then implemented to search the target on the impact surface and it is given by

$$\gamma(x) = \psi(x) - \varphi(x) \tag{3}$$

3. PROJECTILE-TARGET SEARCH ALGORITHM

The initial parameters of the PTS algorithm are the population size N, the number of variables D, the lower bound xmin, the upper bound xmax, and the maximum iteration Imax. The lower and upper bounds of variables are expressed by

$$\boldsymbol{x_{minj}} = [x_{min1} \quad x_{min2} \quad \cdots \quad x_{minD}], \boldsymbol{x_{maxj}} = [x_{max1} \quad x_{max2} \quad \cdots \quad x_{maxD}] \tag{4}$$

The initial projectile population as candidate solutions is randomly generated by assigning random values (rand$_j$) within [0, 1] to each boundary as follows:

$$\boldsymbol{x_{i,j}}^{(I=1)} = \boldsymbol{x_{minj}} + rand_j.\left(\boldsymbol{x_{maxj}} - \boldsymbol{x_{minj}}\right), i = 1, 2, \dots, N; j = 1, 2, \dots, D \tag{5}$$

The target population is randomly created by perturbing a randomly selected projectile with the difference of the two other randomly selected projectiles. The target population is generated by

$$t_{k,j}{}^{(I)} = x_{a1,j}{}^{(I)} + 0.5 \left(x_{a2,j}{}^{(I)} - x_{a3,j}{}^{(I)} \right), k = 1, 2, \dots, N; \ j = 1, 2, \dots, D \tag{6}$$

where the indices a_1, a_2, and $a_3 \in \{1,2,\dots,N\}$ are randomly chosen integers and must be different from each other and all are also different from the base index.

Evaluating the fitness values of the projectiles and the targets are carried out by using (7) and (8) as follows:

$$f_i{}^{(I)} = f(x_{i,j}{}^{(I)}) \tag{7}$$

$$f_k{}^{(I)} = f(t_{k,j}{}^{(I)}) \tag{8}$$

The best projectile $x_{bestj}{}^{(I)}$ and its best value $f_{best}{}^{(I)}$ are then selected by comparing the fitness values of each $x_{i,j}{}^{(I)}$ and $t_{i,j}{}^{(I)}$, as follows:

$$x_{bestj}{}^{(I)} = \begin{cases} x_{i,j}{}^{(I)} \ if \ min(f(x_{i,j}{}^{(I)})) \le min(f(t_{k,j}{}^{(I)})) \\ t_{k,j}{}^{(I)} \quad otherwise \end{cases} \tag{9}$$

$$f_{best}{}^{(I)} = f(x_{bestj}{}^{(I)}) \tag{10}$$

where $I = 1,2,\dots, I_{max}$ which denotes the subsequent generation created for each iteration.

The main process of PTS optimization is iterating the projectile in order to reach its best fitness value. The projectile search model for converging towards to the target is as follows:

for $i = 1:N$

    for $j = 1:D$
    If $rand < 0.5$

$$x_{i,j}{}^{(I+1)} = x_{best1,j}{}^{(I)} - \alpha \left(\frac{\gamma(d_{i,j}{}^{(I)})}{\gamma'(d_{i,j}{}^{(I)})} \right) \tag{11}$$

    Else

$$x_{i,j}{}^{(I+1)} = x_{i,1}{}^{(I)} + \alpha \left(\frac{\gamma(d_{i,j}{}^{(I)})}{\gamma'(d_{i,j}{}^{(I)})} \right) \tag{12}$$

    end
    end
end

In this PSA optimization, a linear shaped hill with a slope of m is used as the impact function. The impact function is

$$\psi(x) = m.x \tag{13}$$

As the impact function has been specified, the function $\gamma(x)$ could be defined here by

$$\gamma(x) = m.x - \frac{v^2}{2g} + \frac{g}{2v^2} x^2 \tag{14}$$

and

$$\left(\frac{\gamma(d_{i,j}{}^{(I)})}{\gamma'(d_{i,j}{}^{(I)})} \right) = \left(\frac{m.d_{i,j}{}^{(I)} - \frac{v^2}{2g} + \frac{g}{2v^2}.(d_{i,j}{}^{(I)})^2}{m + \frac{g}{v^2}.d_{i,j}{}^{(I)}} \right) \tag{15}$$

The distance between projectile and target at the current iteration is calculated as follows:

$$d_{i,j}{}^{(I)} = [abs((t_{i,j}{}^{(I)})^r - (x_{i,j}{}^{(I)})^r)]^{1/s} \tag{16}$$

There are 2 parameters for controlling the distance (d), those are r and s, where (r, s) > 0. Other control parameter for (11) and (12) is defined by

$$\alpha = p^q \quad , p = (1 - \frac{I}{I_{max}}) \quad , q > 0 \tag{17}$$

The new target population ($t_{k,j}^{(I+1)}$) is also created in here. Finally, the best projectile for the next iteration is

$$x_{bestj}^{(I+1)} = \begin{cases} x_{i,j}^{(I+1)} & if \; min(f(x_{i,j}^{(I+1)})) \leq min(f(t_{k,j}^{(I+1)})) \\ t_{k,j}^{(I+1)} & otherwise \end{cases} \tag{18}$$

$$f_{best}^{(I+1)} = f(x_{bestj}^{(I+1)}) \tag{19}$$

The processes are repeated until $f_{best}^{(I+1)}$ meets its desired accuracy level (ε) or the iteration has reached I_{max} and results in $x_{bestj}^{(I+1)}$ as the satisfied projectile landing onto the target.

4. RESULTS AND ANALYSIS

There are six benchmarks and their shifted functions used to evaluate performance of the proposed PTS algorithm. The mathematical formulation of the benchmark functions are given in Table 1. In order to verify the performance of the proposed PTS algorithm, it is compared to other algorithms which have been reported in [5-10]. To carry out the comparison of algorithm performance, the approach using is to compare the accuracies for a fixed number of iterations. In the experiment, 100 dimensions ($D = 100$), 30 search agents ($N = 30$), and 1000 iterations ($I_{max} = 1000$) are implemented for each algorithm. The statistical results after 30 independent experiments are evaluated. The mean and standard deviation (SD) values of the best solutions from the last iterations are put as the metrics to assess the performance of algorithms.

Table 1. Benchmark functions

Function	Formula	Test range (x)	Shift position (o)	Global optimum x*	F(x*)
Sphere	$F_1(x) = \sum_{i=1}^{D}[x_i^2]$	$[-100, 100]^D$	-	[0, 0, …, 0]	0
Rosenbrock	$F_2(x) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	-	[1, 1, …, 1]	0
Schwefel	$F_3(x) = -\sum_{i=1}^{D}[x_i \sin(\sqrt{|x_i|})]$	$[-500, 500]^D$	-	[s, s, …, s] S = 420.968746	-41898.2887
Rastrigin	$F_4(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	-	[0, 0, …, 0]	0
Ackley	$F_5(x) = e^1 + 20 - 20.\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)$	$[-32, 32]^D$	-	[0, 0, …, 0]	0
Griewank	$F_6(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D}x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600. 600]^D$	-	[0, 0, …, 0]	0
Shifted Sphere	$F_7(x) = \sum_{i=1}^{D}[z_i^2], z = x - o$	$[-100, 100]^D$	[-30,-30, …, -30]	[-30, -30, …, -30]	0
Shifted Rosenbrock	$F_8(x) = \sum_{i=1}^{D-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$, z = x - o	$[-30, 30]^D$	[-15,-15, …, -15]	[-14, -14, …, -14]	0
Shifted Schwefel	$F_9(x) = -\sum_{i=1}^{D}[z_i \sin(\sqrt{|z_i|})]$, z = x - o	$[-500, 500]^D$	[-300, -300, …, -300]	[s, s, …, s] S = 120.968746	-41898.2887
Shifted Rastrigin	$F_{10}(x) = \sum_{i=1}^{D}[z_i^2 - 10\cos(2\pi z_i) + 10]$, z = x - o	$[-5.12, 5.12]^D$	[-2, -2, …, -2]	[-2, -2, …, -2]	0
Shifted Ackley	$F_{11}(x) = e^1 + 20 - 20.\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D}z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right)$, z = x - o	$[-32, 32]^D$	[-5, -5, …, -5]	[-5, -5, …, -5]	0
Shifted Griewank	$F_{12}(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D}z_i^2 - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right)$, z = x - o	$[-600. 600]^D$	[-400, -400, …, -400]	(-400, -400, …, -400)	0

The parameters needed to drive the PTS algorithm are: gravitational acceleration (g), initial velocity of projectile (v), slope of target's surface (m), q, r, and s. Gravitational acceleration is a constant of $g = 9.81$. The initial velocity of projectile, the slope of the target's surface, and q give important contributions to the convergence processes, while r and s are controlled the searching process of the PTS algorithm.

To facilitate the analysis, selected functions are used as experiment objects. Figure 3 describes the influence of v on convergence process of the Ackley function under fixed m condition, while Figure 4 is its performance with variations of m at v fixed condition. As seen in Figure 3, v has a strong effect to convergence speed of the PTS algorithm. The convergence process runs relative slowly when v has large values. On the other hand, m has slightly effect to the convergence speed. However, the final fitness values of the searching process vary significantly with both v and m as seen in Table 2.

Other control parameter of the PTS algorithm is q, as shown in Figure 5. Increasing the value of q will speed up the convergence process of the PTS algorithm. This parameter also gives very high contribution to the best final fitness value which could be reached by the algorithm. Table 3 presents the best final fitness value of Sphere function according to its q values. However, not all problems behave the same way as Ackley and Sphere functions. For example, Schwefel function will suffer from stagnation condition and lose its convergence to the global optimum if q is set to a high value, as shown in Figure 6. The parameter q speeds up convergence process for all test functions, but Schwefel function needs relative small speed in order to go to the global optimum path.
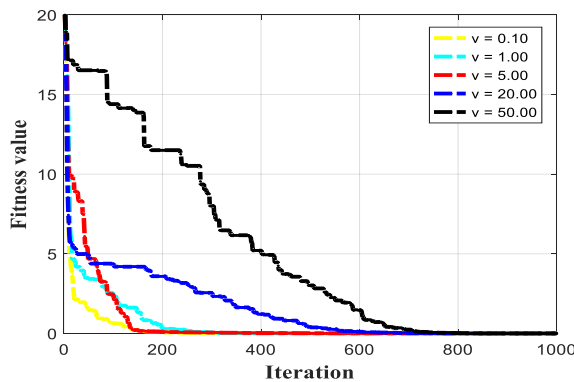

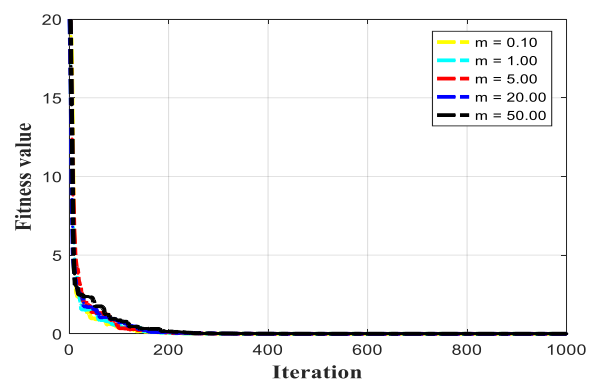Figure 3. Fitness value curves with variations of v

Figure 4. Fitness value curves with variations of m

Table 2. Influence of v and m upon the best final fitness value

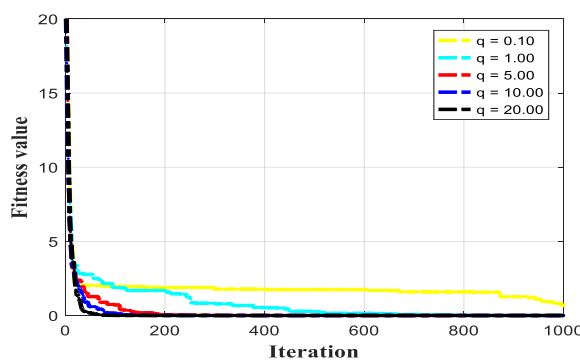| Performance of the Ackley function with variations of m at v				
$m = 10$				
v 0.1	1	5	20	50
2.66E-15	5.82E-13	3.76E-12	4.95E-11	4.13E-09
$v = 0.1$				
m 0.1	1	5	20	50
1.48E-13	3.82E-14	2.66E-15	2.66E-15	2.66E-15

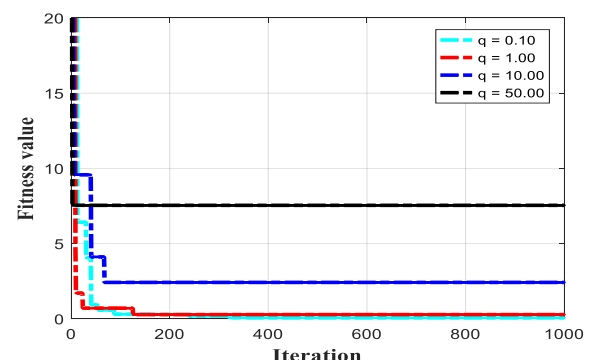Figure 5. Fitness value curves with variations of q

Figure 6. Fitness value curves with variations of q for Schwefel function

Table 3. Influence of q upon the best final fitness value

q	0.1	1	5	10	20	30	40	50
	0.832749	1.37E-08	1.94E-28	2.85E-47	3.76E-69	8.18E-90	1.17E-102	1.14E-128

It has been mentioned that parameters v, m, and q influence to the convergence speed and the closest optimum value that could be reached by the PTS searching processes. However, the convergence towards global optimum will be failed if the projectiles are trapped into local optima in the search space. Furthermore, the projectiles generated by iteration processes are also dependent to distance control parameters: r and s. These parameters are very important in order to enhance probability of the new projectiles to avoid from local optima.

Comparisons of algorithms' performances are summarized in Table 4 and Table 5. The obtain results for benchmarks F_1-F_6 are shown in Table 4. As summarized in Table 4, PTS outperforms hybrid firefly algorithm (HFA), velocity-based artificial bee colony algorithm (VABC), alternative differential evolution algorithm (ADE) and opposition-based magnetic optimization algorithm (FMOA), excluding modified monkey algorithm (MMA), on F_1. Once again, PTS has the best performance on F_2 and F_3. Investigation on F_4 shows that ADE is the best, but PTS and VABC are almost as good. On function F_5, PTS is the best, but MMA, FA, and ADE perform almost as well as PTS. The comparative results also demonstrate that PTS, firefly algorithm (FA), and ADE have the best performances on F_6. Experiments on F_7 and F_{12} show that PTS could find their best global optima points as seen in Table 5. For shifted functions F_7-F_{12}, PTS performs much better than moth-flame optimization algorithm (MFO) [6].

Table 4. Algorithm performance metrics for basic functions, $D = 100$

F	PTS Mean±SD	MMA [6] Mean±SD	HFA/FA [7] Mean±SD	VABC [8] Mean±SD	ADE [9] Mean±SD	FMOA [10] Mean±SD
F_1	9.85E-177 ±0.00E+00	0.00E+0 ±0.00E+00	2.64E-171 ±0.00E+00	1.05E-25 ±2.34E-25	6.37E-45 ±1.12E-44	1.67E-01 ±8.00E-02
F_2	2.62E-19 ±4.77E-19	-	0.077152 ±0.16183	1.60E-11 ±2.62E-11	8.90E+01 ±3.46E+01	9.80E+01 ±3.21E-03
F_3	-39212.93 ±4044.13	-	-12439 ±133.24	-	-	-6.17E+03 ±2.42E+03
F_4	2.31E-15 ±2.39E-15	-	3.39E-08 ±7.29E-09	3.34E-14 ±7.47E-14	0.00E+00 ±0.00E+00	8.74E-02 ±4.93E-02
F_5	4.03E-15 ±2.07E-15	4.4E-15 ±0.00E+00	1.25E-14 ±3.36E-15	1.50E-05 ±3.33E-05	6.21E-15 ±0.00E+00	6.58E-02 ±1.34E-02
F_6	0.00E+00 ±0.00E+00	-	0.00E+0 ±0.00E+00	-100 ±0.00E+00	0.00E+00 ±0.00E+00	1.15E+00 ±7.98E-02

Table 5. Algorithm performance metrics for shifted functions, $D = 100$

Function	PTS		MFO [11]	
	Mean	SD	Mean	SD
F_7	0.00E+00	0.00E+00	0.000117	0.00015
F_8	3.214048E-22	6.740477E-22	139.1487	120.2607
F_9	-42600.949577	1473.517398	-8496.78	725.8737
F_{10}	1.928150E-06	1.222059E-06	84.60009	16.16658
F_{11}	1.170795E-07	9.668957E-08	1.260383	0.72956
F_{12}	0.00E+00	0.00E+00	0.01908	0.021732

5. CONCLUSION

From experiments on selected well-known benchmarks and their shifted functions, it has been demonstrated that PTS algorithm is an effective optimization algorithm to deal with high dimensions global optimization problems. It is also proven to be a very competitive algorithm compared to other well-known metaheuristic algorithms

REFERENCES

[1] Y.D. Sergeyev, D.E. Kvasov, M.S. Mukhametzhanov, "On The Efficiency of Nature-Inspired Metaheuristics in Expensive Global Optimization with Limited Budget," *Scientific Reports*, vol.8, pp.1-9, 2018.

[2] C. Blum, A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Computing Surveys*, vol. 35, pp. 268–308, 2003.

[3] H. Makas, N. Yumusak, "Balancing Exploration and Exploitation by Using Sequential Execution Cooperation Between Artificial Bee Colony and Migrating Birds Optimization Algorithms," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, pp. 4935 – 4956, 2016.

[4] M. Baće, S. Ilijić, Z. Narancić, L. Bistricić, "The Envelope of Projectile Trajectories," *European Journal of Physics*, vol. 23, pp.637-642, 2002.

[5] S. Mirjalili, "Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm," *Knowledge-Based Systems*, vol. 89, pp.228-249, 2015.

[6] R.H. Abiyev, M. Tunay, "Experimental Study of Specific Benchmarking Functions for Modified Monkey Algorithm," *Procedia Computer Science*, vol. 102, pp.595-602, 2016.

[7] L. Zhang, L. Liu, X.S. Yang, Y. Dai, "A Novel Hybrid Firefly Algorithm for Global Optimization," *PLoS One*, vol. 11, e0163230, 2016.

[8] N. Imanian, M.E. Shiri, P. Moradi, "Velocity Based Artificial Bee Colony Algorithm for High Dimensional Continuous Optimization Problems," *Engineering Applications of Artificial Intelligence*, vol. 36, pp.148-163, 2014.

[9] A.W. Mohamed, H.Z. Sabry, M. Khorshid, "An alternative Differential Evolution Algorithm for Global Optimization," *Journal of Advanced Research*, vol. 3, pp.149-165, 2012.

[10] M. Aziz, M.H. Tayarani-N, "Opposition-Based Magnetic Optimization Algorithm with Parameter Adaptation Strategy," *Swarm and Evolutionary Computation*, vol. 26, pp.97-119, 2016.

[11] E.I. Butikov, "The Envelope of Ballistic Trajectories and Elliptic Orbits," *American Journal of Physics*, vol. 83, pp.952-958, 2015.