# THEFT-GANG OPTIMIZATION ALGORITHM: A NOVEL POPULATION BASED META-HEURISTIC FOR UNCONSTRAINED OPTIMIZATION

**M. F. Tabassum[1]*, M. Saeed[1], N. A. Chaudhry[2], J. Ali[1], A. Sana[3]**
[1]Department of Mathematics, University of Management and Technology, Lahore, Pakistan.
[2]Department of Mathematics, UCEST, Lahore Leads University, Lahore, Pakistan.
[3]Department of Mathematics, Lahore Garrison University, Lahore, Pakistan.
* Corresponding Author: Muhammad Farhan Tabassum, farhanuet12@gmail.com , +92-321-4280420

***ABSTRACT:*** *A novel meta-heuristic known as Theft-gang Optimization Algorithm (TOA) is presented in this paper. The proposed method is based on the greedy and criminal behavior of thieves* which *store their expensive objects in hiding places and retrieve it when the objects are needed. The developed method is applied to 26 benchmarking test functions and quality solutions are obtained. The results obtained by TOA are compared with the results of various algorithms like Genetic Algorithm, Differential Evolution, Particle Swarm Optimization, Bees Algorithm and Particle Bee Algorithm. Simulation results reveal that using TOA may lead to finding promising results compared to the other algorithms.*

**Key Words:** *Optimization, Meta-heuristics, Nature inspired Algorithm, Theft-gang Optimization Algorithm,* Unconstrained test functions

## INTRODUCTION

As a fertile source of concepts, principles, and mechanisms, nature can be an inspiration to design artificial computation systems for solving complex computational problems. Evolutionary algorithms inspired by biological evolution, and swarm intelligence algorithms, inspired by collective animal behavior, are two main classes of nature inspired computations which have attracted more and more attentions during recent years. Owing to their simplicity and flexibility, Evolutionary algorithms have been widely applied to solve scientific and engineering problems and have been the most successful artificial computation systems to tackle complex computational problems [1].

An evolutionary algorithm is a generic population based meta-heuristic optimization approach, trying to simulate some mechanisms of biological evolution. There are different variants of evolutionary algorithms, but the common underlying idea behind all these problem-solving techniques is the same [2].

Success of an optimization algorithm depends mostly on its ability to establish good balance between exploration and exploitation [2]. Exploration refers to generation of new solutions in as yet untested regions of search space and exploitation means the concentration of the algorithm's search at the vicinity of current good solutions.

For many years, human have utilized the guidance of nature in finding the most appropriate solution for problems. Hence, during the last decades, there has been a growing attempt in developing algorithms inspired by nature [3–5]. For example, Genetic algorithm was proposed by Holland [6], and simulates Darwnian evolution concepts. Artificial Immune Systems [7], simulate biological immune systems for optimization. Ant Colony Optimization [8] was inspired by behavior of ants foraging for food. Particle Swarm Optimization [9] mimics the social behavior of a flock of migrating birds trying to reach an unknown destination. Marriage in Honey Bee Optimization Algorithm (MBO) was proposed by Abbass [10], and mimics processes of reproduction in the honey bee colony. Bacterial Foraging Algorithm [11] simulates search and optimal for aging of bacteria. The Shuffled Frog Leaping algorithm [12] was inspired by a frog population searching for food. The Cat Swarm algorithm [13] was developed based on the behavior of cats. Invasive weed optimization was proposed by Mehrabian and Lucas [14], and mimics the ecological behavior of colonizing weeds. Monkey Search [15] simulates a monkey in search for food resources. Water flow-like algorithm [16] was inspired by water flowing from higher to lower levels. Biogeography-based optimization algorithm was introduced by Simon [17], and inspired by biogeography which refers to the study of biological organisms in terms of geographical distribution (over time and space). The Fish School Search [18] was proposed based on the gregarious behavior of oceanic fish. Cuckoo Search [19] and Cuckoo optimization algorithm [20] are based on reproduction strategy of cuckoos. Bat-inspired Algorithm [21] was inspired by the echolocation behavior of bats. Firefly algorithm [22] simulates the social behavior of fireflies based on their flashing characteristics. Dolphin Partner Optimization [23] and Dolphin echolocation algorithm [24] were inspired by dolphins' behaviors. Flower pollination algorithm [25] mimics the pollination characteristics of flowering plants and the associated flower consistency of some pollinating insects. Krill herd [26] inspired by the herding behavior of krill individuals. Wolf search [27] and Grey Wolf Optimizer [28] are inspired by behaviors of wolves. Water cycle algorithm [29] was based on the observation of water cycle process and how rivers and streams flow to the sea in the real world. The Social spider optimization, inspired by the social behavior of a kind of spider, has been proposed recently [30]. Forest Optimization Algorithm [31] was inspired by few trees in the forests which can survive for many years, while other trees could live for a short time.

The proposed method is based on the greedy and criminal behavior of thieves which store their expensive objects in hiding places and retrieve it when the objects are needed. An individual with a thief's mentality may steal, but they are just as apt to lie and cheat. The thief's mentality begins as a coping mechanism for dealing with the character flaws that drive them to do what they do, but it progresses from those harmless, white lies to a form of deception that requires that generational foundation.

The rest of this paper is organized as follows: Section 2 provides a basic framework of the proposed TOA. Experimental results based on several benchmarking optimization test functions and comparisons with previously reported results are presented in Section 3. Section 4 presents a discussion and conclusions of the TOA.

## THEFT-GANG OPTIMIZATION ALGORITHM

"To think like a thief, you have to know how a thief's mind works". Generally we associate bad things with a thief but it is not necessary that he is a bad guy. "If you want to think like a thief, act like a thief". The best way to understand how thieves think is to become one of them.

From optimization point of view: the thieves are searchers, the environment is search space, each position of the environment is corresponding to a feasible solution, the cost of source object is objective function value (fitness) and the best object source of the environment is the global solution of the problem.

This paper based on the above-mentioned criminal and greedy behaviors of thieves, a population-based meta-heuristic algorithm, TOA, is developed. The principles of TOA are listed as follows:

- Thieves live in the form of groups (gang).
- Thieves memorize the position of their hiding places.
- Thieves follow each other to do thievery.
- Thieves protect their hides from being pilfered by a probability.

It is assumed that there is a *d-dimensional* environment including a number of thieves. The number of thieves (gang size) is '*N*' and the position of thief 'X' at time iteration '*itr*' in the search space is specified by a vector

$$t^{X,itr} \qquad (X = 1,2,3,...,N; \quad itr = 1,2,3,...,itr_{max})$$

where $t^{X,itr} = [t_1^{X,itr}, t_2^{X,itr}, t_3^{X,itr}, \cdots, t_d^{X,itr}]$ and $itr_{max}$ is the maximum number of iterations. Each thief has a memory in which the position of its hiding place is memorized. At iteration '*itr*', the position of hiding place of thief 'X' is shown by $memory^{X,itr}$. This is the best position that thief 'X' has obtained so far. Indeed, in memory of each thief the position of its best experience has been memorized. Thieves move in the environment and search for better sources (hiding places).

Assume that at iteration '*itr*', thief 'Y' wants to visit its hiding place, $memory^{Y,itr}$. At this iteration, thief 'X' decides to follow thief 'Y' to approach to the hiding place of thief 'Y'. In this case, two phases may happen:

**Phase 1:** Thief 'Y' does not know that thief 'X' is following it. As a result, thief 'X' will approach to the hiding place of thief 'Y'. In this case, the new position of thief 'X' is obtained as follows:

$$t^{X,itr+1} = t^{X,itr} + rand(0,1) \times (distance)^{X,itr} \times (memory^{X,itr} - t^{X,itr}) \qquad (1)$$

where *rand(0,1)* is the random number between 0 and 1 and $distance^{X,itr}$ denotes the chasing distance of thief '*X*' at iteration '*itr*'.

Fig. 1 shows the geometry of the effect of chasing distance on the search capability. Small values of '*distance*' leads to local search (at the vicinity of $t^{X,itr}$) and large values results in global search (far from $t^{X,itr}$). As Fig. 1(a) shows, if the value of chasing distance is selected less than 1, the next position of thief X is in between $t^{X,itr}$ and $memory^{X,itr}$. As Fig. 1(b) indicates, if the value of '*chasing distance*' is selected more than 1, the next position of thief '*X*' is any position which may exceed $memory^{X,itr}$.

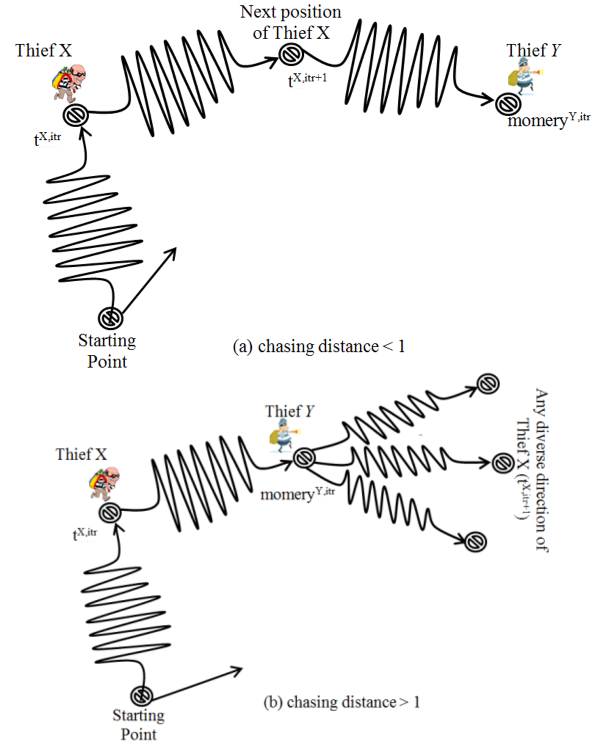


**Figure 1. Geometry of phase 1 in TOA**

**Phase 2:** Thief '*Y*' knows that thief '*X*' is following it. As a result, in order to protect its collection from being stolen, thief '*Y*' will deceive thief '*X*' by going to another position of the search space, which is in any diverse direction. Phase 1 and 2 can be expressed as follows:

$$t^{e,itr+1} = \begin{cases} equation\ (1) & rand(0,1) \geq \mathrm{int.prob}^{Y,itr} \\ a\ random\ position & otherwise \end{cases} \qquad (2)$$

where *rand(0,1)* is the random number between 0 and 1 and $\mathrm{int.prob}^{Y,itr}$ denotes the intelligent probability (int.prob) of thief '*Y*' at iteration '*itr*' in the search space.

In TOA, intensification and diversification are mainly controlled by the parameter of intelligent probability (int.prob). By decrease of the intelligent probability value, TOA tends to conduct the search on a local region where a current good solution is found in this region. As a result, using small values of intelligent probability, increases intensification. On the other hand, by increase of the intelligent probability value, the probability of searching the vicinity of current good solutions decreases and TOA tends to explore the search space on a global scale. As a result, use of large values of IP increases diversification. Figure 2 represents flowchart of Theft-gang Optimization Algorithm.

## EXPERIMENTATION

This study considered several numerical optimization problems from the literature to validate TOA performance.

This section is divided into two sub-sections. Frist section provides a large set of complex 12 uni-model mathematical benchmark problems to be tested, with results compared against other metaheuristic algorithms. Second section examines 14 multi-model mathematical benchmark problems to be tested, with results compared against other meta-heuristic algorithms.

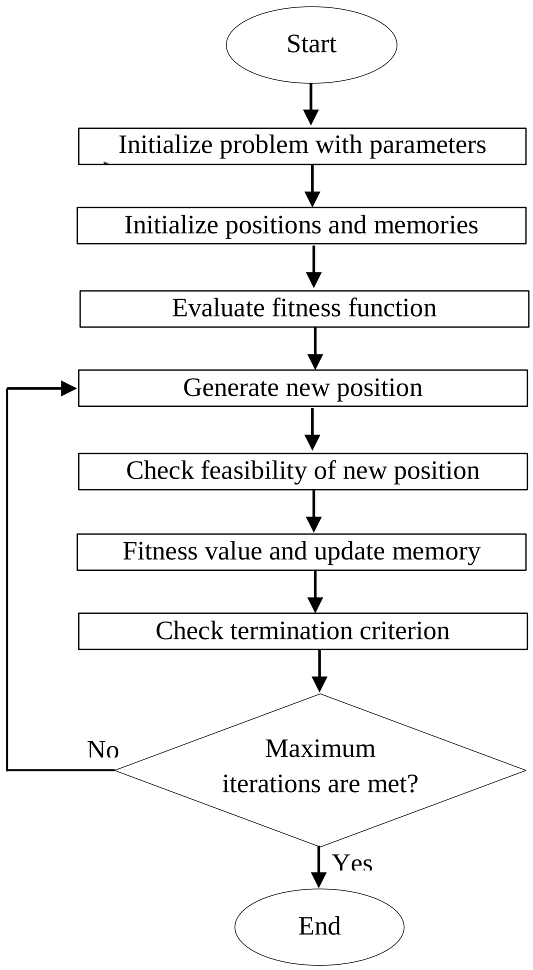


**Figure 2. Flowchart of Theft-gang Optimization Algorithm.**

## Benchmark Test Functions

### Uni-model Test Functions

Function Number: 1
Function Name: Beale
Range: [-4.5, 4.5]
Dimension: 2
Type: Non-separable
Test Function:

$$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$$

Global Optimum: 0

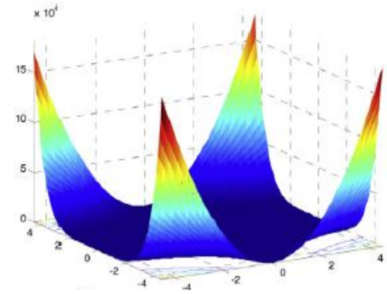

**Figure 3: Geometrical representation of Beale Function**

Function Number: 2
Function Name: Colville
Range: [-10, 10]
Dimension: 4
Type: Non-separable
Test Function:

$$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$$

Global Optimum: 0

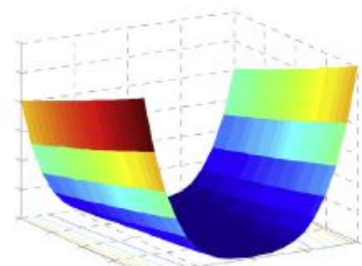

**Figure 4: Geometrical representation of Colville Function**

Function Number: 3
Function Name: Dixon-Price
Range: [-10, 10]
Dimension: 30
Type: Non-separable
Test Function: $f(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_i - 1)^2$
Global Optimum: 0

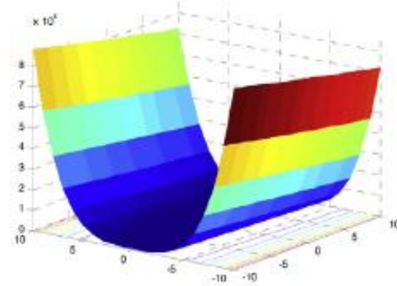

**Figure 5: Geometrical representation of Dixon-Price Function**

Function Number: 4
Function Name: Easom
Range: [-100, 100]
Dimension: 2
Type: Non-separable
Test Function: $f(x) = -\cos(x_1)\cos(x_1)e^{(-(x_1-\pi)^2 - (x_2-\pi)^2)}$
Global Optimum: -1

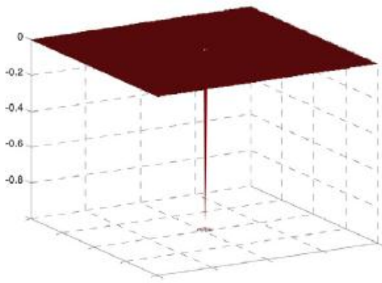

**Figure 6: Geometrical representation of Easom Function**

Function Number: 5
Function Name: Matyas
Range: [-10, 10]
Dimension: 2
Type: Non-separable
Test Function: $f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
Global Optimum: 0

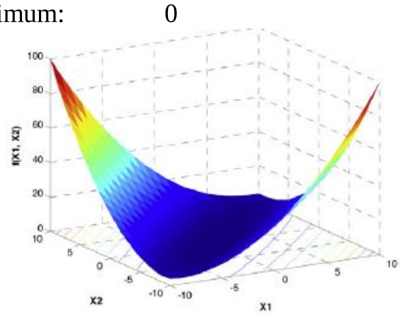

**Figure 7: Geometrical representation of Matyas Function**

Function Number: 6
Function Name: Quartic
Range: [-1.28, 1.28]
Dimension: 30
Type: Separable
Test Function: $f(x) = \sum_{i=1}^{D} ix_i^4 + Rand$
Global Optimum: 0

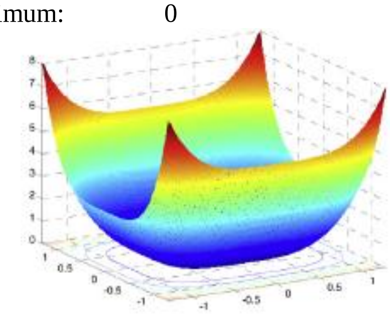

**Figure 8: Geometrical representation of Quartic Function**

Function Number: 7
Function Name: Schwefel 1.2
Range: [-100, 100]
Dimension: 30
Type: Non-separable
Test Function: $f(x) = \sum_{i=1}^{D} \left( \sum_{j=1}^{i} x_j \right)^2$
Global Optimum: 0

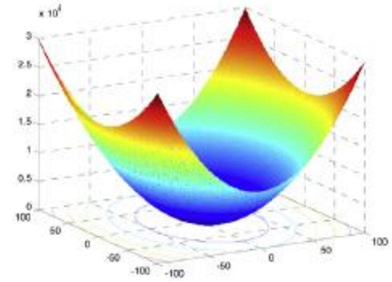

**Figure 9: Geometrical representation of Schwefel 1.2 Function**

Function Number: 8
Function Name: Schwefel 2.22
Range: [-10, 10]
Dimension: 30
Type: Non-separable
Test Function: $f(x) = \sum_{i=1}^{D} |x_i| + \Pi_{i=1}^{D} |x_i|$
Global Optimum: 0

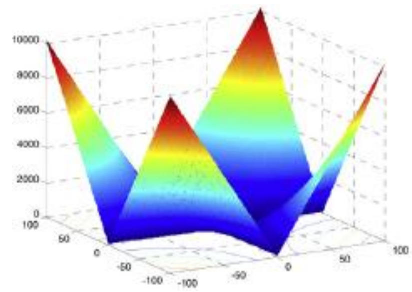

**Figure 10: Geometrical representation of Schwefel2.22 Function**

Function Number: 9
Function Name: Sphere
Range: [-100, 100]
Dimension: 30
Type: Separable
Test Function: $f(x) = \sum_{i=1}^{D} x_i^2$
Global Optimum: 0

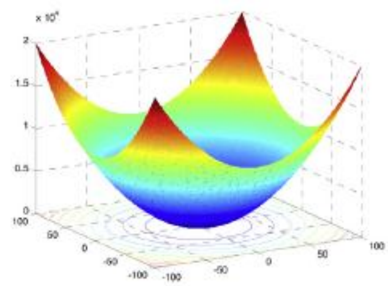

**Figure 11: Geometrical representation of Sphere Function**

Function Number: 10
Function Name: Step
Range: [-5.12, 5.12]
Dimension: 30
Type: Separable
Test Function: $f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$
Global Optimum: 0

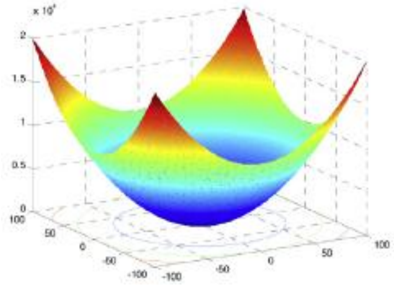

**Figure 12: Geometrical representation of Step Function**

Function Number: 11
Function Name: Sum Sphere
Range: [-10, 10]
Dimension: 30
Type: Separable
Test Function: $f(x)=\sum_{i=1}^{D} i x_i^2$
Global Optimum: 0

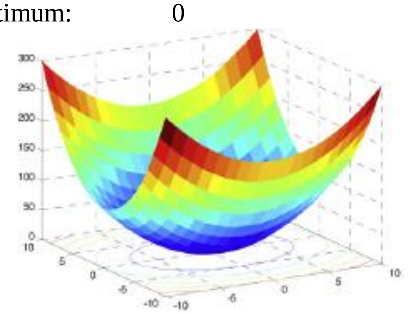

**Figure 13: Geometrical representation of Sum Sphere Function**

Function Number: 12
Function Name: Zakhrov
Range: [-5, 10]
Dimension: 10
Type: Non-separable
Test Function: $f(x)=\sum_{i=1}^{D} x_i^2+\left(\sum_{i=1}^{D} 0.5 i x_i\right)^2+\left(\sum_{i=1}^{D} 0.5 i x_i\right)^4$
Global Optimum: 0

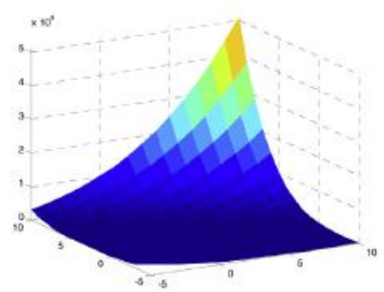

**Figure 14: Geometrical representation of Zakhrov Function**

**Multi-model Test Functions**

Function Number: 1
Function Name: Ackley
Range: [-32, 32]
Dimension: 30
Type: Non-separable
Test Function: $f(x)=-20e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right)}-e^{\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right)}+20+e$
Global Optimum: 0

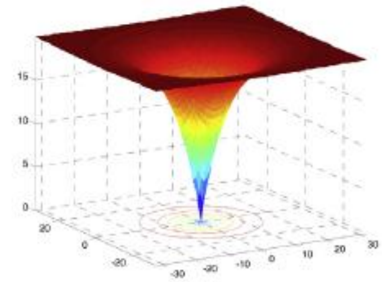

**Figure 15: Geometrical representation of Ackley Function**

Function Number: 2
Function Name: Boachevsky2
Range: [-100, 100]
Dimension: 2
Type: Non-separable
Test Function: $f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)(4\pi x_2)+0.3$
Global Optimum: 0

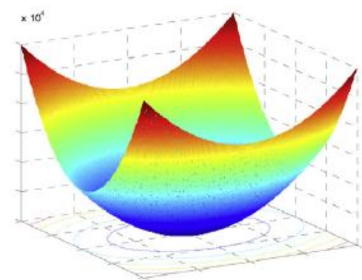

**Figure 16: Geometrical representation of Boachevsky2Function**

Function Number: 3
Function Name: Boachevsky3
Range: [-100, 100]
Dimension: 2
Type: Non-separable
Test Function: $f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$
Global Optimum: 0

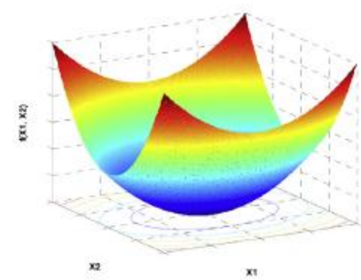

**Figure 17: Geometrical representation of Boachevsky3 Function**

Function Number: 4
Function Name: Bohachevsky1
Range: [-100, 100]
Dimension: 2
Type: Separable
Test Function: $f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$
Global Optimum: 0

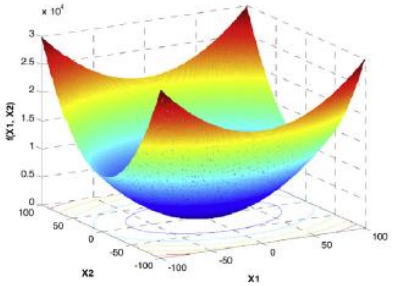


**Figure 18: Geometrical representation of Bohachevsky1Function**

Function Number: 5
Function Name: Booth
Range: [-10, 10]
Dimension: 2
Type: Separable
Test Function: $f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
Global Optimum: 0

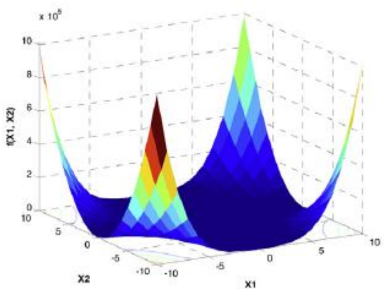


**Figure 19: Geometrical representation of Booth Function**

Function Number: 6
Function Name: Griewank
Range: [-600, 600]
Dimension: 30
Type: Non-separable
Test Function: $f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D}(x_i - 100)^2\right) - \left(\Pi_{i=1}^{D}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right)\right) + 1$
Global Optimum: 0

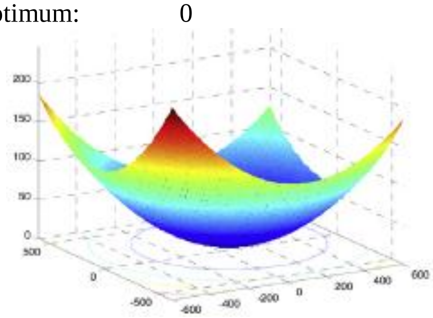

**Figure 20: Geometrical representation of Griewank Function**

Function Number: 7
Function Name: Michalewicz2
Range: [0, π]
Dimension: 2
Type: Separable
Test Function: $f(x) = -\sum_{i=1}^{D}\sin(x_i)(\sin(\frac{ix_i^2}{\pi}))^{20}$
Global Optimum: -1.8013

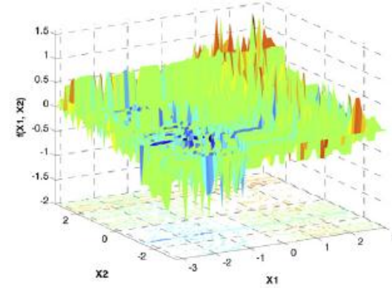


**Figure 21: Geometrical representation of Michalewicz2 Function**

Function Number: 8
Function Name: Michalewicz5
Range: [0, π]
Dimension: 5
Type: Separable
Test Function: $f(x) = -\sum_{i=1}^{D}\sin(x_i)(\sin(\frac{ix_i^2}{\pi}))^{20}$
Global Optimum: -4.6877

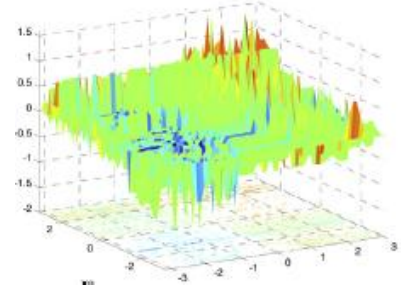

**Figure 22: Geometrical representation of Michalewicz5 Function**

Function Number: 9
Function Name: Michalewicz10
Range: [0, π]
Dimension: 10
Type: Separable
Test Function: $f(x) = -\sum_{i=1}^{D}\sin(x_i)(\sin(\frac{ix_i^2}{\pi}))^{20}$
Global Optimum: -9.6602

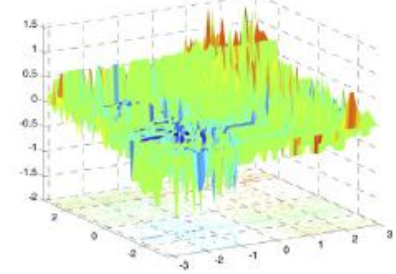

**Figure 23: Geometrical representation of Michalewicz10 Function**

Function Number: 10
Function Name: Rastrigin
Range: [-5.12, 5.12]
Dimension: 30
Type: Separable
Test Function: $f(x) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$
Global Optimum: 0

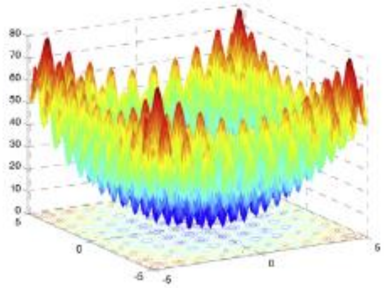

**Figure 24: Geometrical representation of Rastrigin Function**

Function Number: 11
Function Name: Rosenbrock
Range: [-30, 30]
Dimension: 30
Type: Non-separable
Test Function: $f(x)=\sum_{i=1}^{D-1}100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2$
Global Optimum: 0

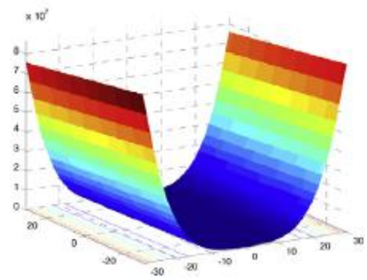

**Figure 25: Geometrical representation of Rosenbrock Function**

Function Number: 12
Function Name: Schaffer
Range: [-100, 100]
Dimension: 2
Type: Non-separable
Test Function: $f(x)=0.5+\frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2}$
Global Optimum: 0

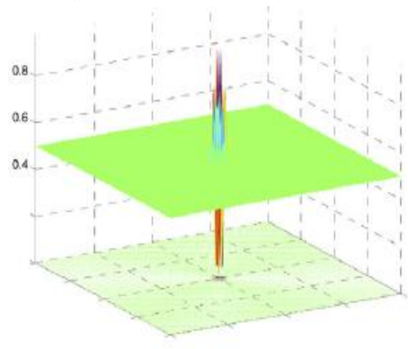

**Figure 26: Geometrical representation of Schaffer Function**

Function Number: 13
Function Name: Shubert
Range: [-10, 10]
Dimension: 2
Type: Non-separable
Test Function: $f(x)=\left(\sum_{i=1}^{5}i\cos\left(i+1\right)x_1+i\right)\left(\sum_{i=1}^{5}i\cos\left(\left(i+1\right)x_2+i\right)\right)$
Global Optimum: -186.73

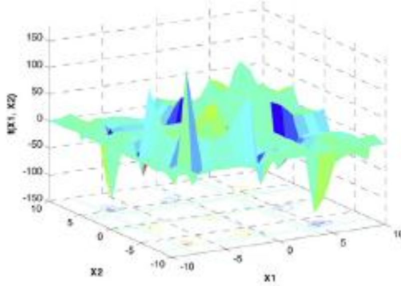

**Figure 27: Geometrical representation of Shubert Function**

Function Number: 14
Function Name: Six Hamp Camel Back
Range: [-5, 5]
Dimension: 2
Type: Non-separable
Test Function: $f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$
Global Optimum: -1.03163

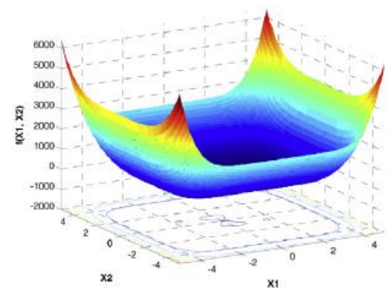

**Figure 28: Geometrical representation of Six Hamp Camel Back Function**

This section compares the performance of TOA to the performance of other meta-heuristic algorithms including Genetic Algorithm, Differential Evolution, Particle Swarm Optimization, Bees Algorithm and Particle Bee Algorithm [32] using 26 meta-heuristic algorithm benchmark functions described by Cheng and Lien [32]. Some functions are two-dimensional; some of them are four- and five-dimensional; and remaining functions are 30-dimensional. All functions may be separated into the type categories of multi-modal/uni-modal and separable/non-separable.

Geometrical representation of these benchmarks is shown in Figs. 3–28. Cheng and Lien [32] previously conducted experiments on all functions with a 500,000 maximum number of function evaluations. They reported any value less than $e^{-12}$ as 0. To maintain comparison consistency, TOA was also tested using these same conditions. Table 1 lists control and specific parameter settings for each algorithm.

**Table 1. Parameter settings of the algorithms**

| GA | DE | PSO | BA | PBA | TOA |
|---|---|---|---|---|---|
| n = 50 | n = 50 | n = 50 | n = 50 | n = 50 | N = 50 |
| m = 0.01 | c = 0.9 | w = 0.9–0.7 | e = NP/2 | e = NP/2 | rd=2 |
| c = 0.8 | F = 0.5 | v = $X_{min}$/10 ~ $X_{max}$/10 | b = NP/4 | b = NP/4 | IP=0.1 |
| g = 0.9 | | | r = NP/4 | r = NP/4 | |
| | | | $n_1$ = 2 | w = 0.9–0.7 | |
| | | | $n_2$ = 1 | v = $X_{min}$/10 ~ $X_{max}$/10 | |
| | | | | Pelite = 15 | |
| | | | | Pbest = 9 | |

Table 2 delineates the respective performance of TOA and other algorithms in solving benchmark functions. Performance values for all algorithms except for TOA reference Cheng and Lien [32]. The mean value and standard deviation for TOA were obtained after 30 independent runs, in line with standards followed in the previous work. In Table 2, bolded numbers represent the comparatively best values. TOA found the global optimum value for 24 of the 26 functions and outperformed all other algorithms tested. Further, TOA was the only algorithm able to solve Dixon-Price (function 8) and produced the best result of all on the exceptionally difficult Rosenbrock (function 17).

.

**Table 2. Comparative results of TOA with GA, DE, PSO, BA, and PBA**

| No. | Functions | | Min | **GA** [33] | **DE** [33] | **PSO** [33] | **BA** [33] | **PBA** [33] | **TOA** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Ackley | Mean | 0 | 14.67178 | **0** | 0.16462 | **0** | 3.12e-8 | **0** |
| | | Std Dev | | 0.17814 | 0 | 0.49387 | 0 | 3.98e-8 | 0 |
| 2 | Beale | Mean | 0 | **0** | **0** | **0** | 1.88e-5 | **0** | **0** |
| | | Std Dev | | 0 | 0 | 0 | 1.94e-5 | 0 | 0 |
| 3 | Boachevsky2 | Mean | 0 | 0.06829 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0.07822 | 0 | 0 | 0 | 0 | 0 |
| 4 | Boachevsky3 | Mean | 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | Bohachevsky1 | Mean | 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | Booth | Mean | 0 | **0** | **0** | **0** | 0.00053 | **0** | **0** |
| | | Std Dev | | 0 | 0 | 0 | 0.00074 | 0 | 0 |
| 7 | Colville | Mean | 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0.01949 | 0.01949 | 0 | 1.11760 | 0 | 0 |
| 8 | Dixon-Price | Mean | 0 | 1.22e+3 | 0.66667 | 0.66667 | 0.66667 | 0.66667 | **0** |
| | | Std Dev | | 2.66e+2 | 1e-9 | 1e-8 | 1.16e-9 | 5.65e-10 | 0 |
| 9 | Easom | Mean | -1 | **-1** | **-1** | **-1** | -0.99994 | **-1** | **-1** |
| | | Std Dev | | 0 | 0 | 0 | 4.5e-5 | 0 | 0 |
| 10 | Griewank | Mean | 0 | 10.63346 | 0.00148 | 0.01739 | **0** | 0.00468 | **0** |
| | | Std Dev | | 1.16146 | 0.00296 | 0.02081 | 0 | 0.00672 | 0 |
| 11 | Matyas | Mean | 0 | **0** | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | Michalewicz2 | Mean | -1.8013 | **-1.8013** | **-1.8013** | **-1.8013** | **-1.8013** | **-1.8013** | **-1.8013** |
| | | Std Dev | | 0 | 0 | 0.11986 | 0 | 0 | 0 |
| 13 | Michalewicz5 | Mean | -4.6877 | **-4.6877** | **-4.6877** | **-4.6877** | **-4.6877** | **-4.6877** | **-4.6877** |
| | | Std Dev | | 0.09785 | 0.01253 | 0.25695 | 0 | 0 | 0 |
| 14 | Michalewicz10 | Mean | -9.6602 | -9.49683 | -9.59115 | -4.00718 | **-9.6602** | **-9.6602** | **-9.6602** |
| | | Std Dev | | 0.14112 | 0.06421 | 0.50263 | 0 | 0 | 0 |
| 15 | Quartic | Mean | 0 | 0.18070 | 0.00136 | 0.00116 | 1.72e-6 | 0.00678 | **7.12e-9** |
| | | Std Dev | | 0.02712 | 0.00042 | 0.00042 | 1.85e-6 | 0.00133 | 1.35e-7 |
| 16 | Rastrigin | Mean | 0 | 52.92259 | 11.71673 | 43.97714 | **0** | **0** | **0** |
| | | Std Dev | | 4.56486 | 2.53817 | 11.72868 | 0 | 0 | 0 |
| 17 | Rosenbrock | Mean | 0 | 1.96e+5 | 18.20394 | 15.08862 | 28.834 | 4.2831 | **0.84e-9** |
| | | Std Dev | | 3.85e+4 | 5.03619 | 24.17019 | 0.10597 | 5.7877 | 1.86e-8 |
| 18 | Schaffer | Mean | 0 | 0.00424 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0.00476 | 0 | 0 | 0 | 0 | 0 |
| 19 | Schwefel 1.2 | Mean | 0 | 7.40e+3 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 1.14e+3 | 0 | 0 | 0 | 0 | 0 |
| 20 | Schwefel 2.22 | Mean | 0 | 11.0214 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 1.38686 | 0 | 0 | 0 | 0 | 0 |
| 21 | Shubert | Mean | -186.73 | **-186.73** | **-186.73** | **-186.73** | **-186.73** | **-186.73** | **-186.73** |
| | | Std Dev | | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | Six Hamp Camel Back | Mean | -1.03163 | **-1.03163** | **-1.03163** | **-1.03163** | **-1.03163** | **-1.03163** | **-1.03163** |
| | | Std Dev | | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | Sphere | Mean | 0 | 1.11e+3 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 74.21447 | 0 | 0 | 0 | 0 | 0 |
| 24 | Step | Mean | 0 | 1.73e+3 | **0** | **0** | 5.12370 | **0** | **0** |
| | | Std Dev | | 76.56145 | 0 | 0 | 0.39209 | 0 | 0 |
| 25 | Sum Sphere | Mean | 0 | 1.48e+3 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 12.40929 | 0 | 0 | 0 | 0 | 0 |
| 26 | Zakhrov | Mean | 0 | 0.01336 | **0** | **0** | **0** | **0** | **0** |
| | | Std Dev | | 0.00453 | 0 | 0 | 0 | 0 | 0 |
| Number of algorithm found global minimum | | | | 11 | 20 | 19 | 19 | 21 | 24 |

## CONCLUSION

This paper presents a new meta-heuristic algorithm called Theft-gang Optimization Algorithm (TOA) inspired by the greedy and criminal behavior of thieves which store their expensive objects in hiding places and retrieve it when the objects are needed. TOA simulated this natural pattern using the two strategies of thievery. Its application to sample problems demonstrated the ability of TOA to generate solutions at a quality significantly better than other metaheuristic algorithms. Based on mathematical benchmark function results, TOA precisely identified 24 of 26 benchmark function solutions, surpassing the performance of GA, DE, BA, PSO, and PBA. The two phases of the TOA algorithm are simple to operate, with only simple mathematical operations to code. We thus conclude that the novel TOA algorithm, while robust and easy to implement, is able to solve various numerical optimization problems despite using fewer control parameters than competing algorithms

## REFERENCES


[1] He S, Wu QH, Saunders JR. "Group search optimizer: an optimization algorithm inspired by animal searching behavior", *IEEE Trans Evol Comput*. **13**: 973–90 (2009)

[2] Eiben AE, Smith JE. "Introduction to evolutionary computing", Springer Verlag, (2003)

[3] Oftadeh R, Mahjoob MJ, Shariatpanahi M. "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search", *Comput. Math. Appl.* **60**(7): 2087–98 (2010)

[4] Bhargava V, Fateen SEK, Bonilla-Petriciolet A. "Cuckoo search: a new nature inspired optimization method for phase equilibrium calculations", *Fluid Phase Equilib*. **337:** 191–200 (2013)

[5] Zheng Y-J. "Water wave optimization: a new nature inspired meta-heuristic", *Comput. Oper. Res*. 55: 1–11 (2015)

[6] Holland JH. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", U Michigan Press, (1975)

[7] Farmer JD, Packard NH, Perelson AS. "The immune system, adaptation, and machine learning", *Physica D: Nonlinear Phenom.* **22**(1): 187–204 (1986)

[8] Dorigo M. "Optimization, learning and natural algorithms" Ph.D. thesis. Italy: Politecnico di Milano, (1992)

[9] R.C., Eberhart and J. Kennedy. "A new optimizer using particle swarm theory", Proceedings of the sixth International Symposium on *Micro Machine and Human Science*, New York, (1995)

[10] H.A., Abbass. "MBO: marriage in honey bees optimization – a haplometrosis polygynous swarming approach", *Proceedings of the IEEE Congress on Evolutionary Computation*, (2001)

[11] Passino KM. "Biomimicry of bacterial foraging for distributed optimization and control", *Control Syst. IEEE*. **22**(3): 52–67 (2002)

[12] Eusuff MM, Lansey KE. "Optimization of water distribution network design using the shuffled frog leaping algorithm", *J. Water Res. Plan. Manag.* 129(3):210–25 (2003)

[13] Chu S-C, Tsai P-W, Pan J-S. "Cat swarm optimization", *Trends in Artificial Intelligence*. Springer, 854–8 (2006)

[14] Mehrabian AR, Lucas C. "A novel numerical optimization algorithm inspired from weed colonization", *Ecol. Inform.* **1**(4) 355–66 (2006)

[15] Mucherino A, Seref O. "Monkey search: a novel meta-heuristic search for global optimization", *Data Mining, Systems Analysis and Optimization in Biomedicine*. AIP Publishing, (2007)

[16] Yang F-C, Wang Y-P. "Water flow-like algorithm for object grouping problems", *J. Chin. Inst. Ind. Eng*. **24**(6): 475–88 (2007)

[17] Simon D. "Biogeography-based optimization", *Evolut. Comput. IEEE Trans*. **12**(6): 702–13 (2008)

[18] F., de Lima Neto, et al. "A novel search algorithm based on fish school behavior", in: *Proceedings of the IEEE International Conference on Systems*, Manand Cybernetics, SMC, (2008)

[19] X.-S., Yang and S. Deb. "Cuckoo Search via Lévy flights", in: *Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing*, NaBIC, (2009)

[20] Rajabioun R. "Cuckoo optimization algorithm", *Appl. Soft Comput*. **11**(8): 5508–18 (2011)

[21] Yang X-S. "A new meta-heuristic bat-inspired algorithm", *Nature Inspired Cooperative Strategies for Optimization* Springer, 65–74 (2010)

[22] Yang X-S. "Firefly algorithms for multimodal optimization", Stochastic Algorithms: Foundations and Applications. Springer, 169–78 (2010)

[23] Y., Shiqin, J. Jianjun, and Y. Guangxing. "A dolphin partner optimization", in: *Proceedings of the IEEE WRI Global Congresson Intelligent Systems*, GCIS, (2009)

[24] Kaveh A, Farhoudi N. "A new optimization method: dolphin cholocation", *Adv. Eng. Softw*. **59**: 53–70 (2013)

[25] Yang X-S. "Flower pollination algorithm for global optimization", *Unconventional Computation and Natural Computation*. Springer, 240–9 (2012)

[26] Gandomi AH, Alavi AH. "Krill herd: a new bio-inspired optimization algorithm", *Commun. Nonlinear Sci. Numer. Simul*. **17**(12): 4831–45 (2012)

[27] R., Tang, et al., "Wolf search algorithm with ephemeral memory", in: *Proceedings of the Seventh International Conference on IEEE Digital Information Management* (ICDIM), (2012)

[28] Mirjalili S, Mirjalili SM, Lewis A. "Grey wolf optimizer", *Adv. Eng. Softw*. **69**: 46–61 (2014)

[29] Eskandar, H, et al. "Water cycle algorithm – a novel meta-heuristic optimization method for solving constrained engineering optimization problems", *Comput. Struct*. **110–111**:151–66 (2012)

[30] Cuevas, E,etal. "A swarm optimization algorithm inspired in the behavior of the social-spider", *Expert Syst. Appl*. **40**(16): 6374–84 (2013)

[31] Ghaemi M, Feizi-Derakhshi M-R. "Forest optimization algorithm*", Expert Syst. Appl*. **41**(15): 6676–87 (2014)

[32] Cheng M-Y, Lien L-C. "Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization", *J Comput Civil Eng*. **26**: 612–24 (2012)

[33] Cheng M-Y, Doddy P. "Symbiotic Organisms Search: A new meta-heuristic optimization algorithm" *Computers and Structures* **139**: 98–112 (2014)