New Optimization Algorithm Based on Venus Flytrap Plant

Amany A. Naim and Neveen I. Ghali

***Abstract*—An optimization algorithm can be defined as an attempt to find solutions to a problem under limited conditions. Heuristic algorithms are considered as special type of optimization algorithms. They are suggested by inspiration from nature. For instance, Genetic Algorithm (GA) has been inspired by the mechanics of natural selection and natural genetics. Venus flytrap optimization is a comparatively novel algorithm for the heuristic algorithm family, which is based on the natural behavior of the venus flytrap plant. The proposed algorithm is called Venus Flytrap Optimization (VFO), for solving the numerical optimization problems. Experimental analysis is implemented on some benchmark functions to show the performance of the proposed algorithm.**

***Index Terms*—Venus Flytrap Optimization, Optimization Algorithm, Genetic Algorithm, Venus Flytrap Plant, Heuristic Algorithm, Benchmark Functions.**

I. Introduction

EVOLUTIONARY algorithm (EA) is a comprehensive expression used to describe population-based random search algorithms, which is in some sense imitative natural behavior [1]. Nature-inspired algorithms are a branch of new problem-solving methodologies and have expanded the field for Artificial Intelligence (AL) [2]. Memorable agents of such algorithms are Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4], Whale Optimization Algorithm (WOA) [5], and Shuffled Frog Leaping Algorithm (SFLA) [6].

Recently, Xin-She and Suash in 2009 proposed the Cuckoo Search (CS) as a heuristic method for solving the optimization problems [7]. This method was based on the oblige brood parasitic behavior of the cuckoo in incorporation with the levy flight behavior of birds and fruit flies. This method was implemented in the different applications, authors in [8] solved band selection problem by new method based on a binary version of cuckoo search algorithm and applied on hyperspectral image data. Bilal and David in [9] Hybridized a mutation operator with cuckoo search algorithm and tested them on benchmark functions.

Authors in 2015 approached social spider behavior to solve global optimization problems. It was based on a strategy that depends on the social spiders searching behavior as they used their spider web vibrations to locate prey [10]. This algorithm was applied in the various application, in [11] Emine and Erkan applied binary social spider algorithm on continuous optimization problem. Authors in [12] combined social spider algorithm with the differential evolution algorithm.

Manuscript received December 8, 2020; revised April 24, 2021.

Amany A. Naim is a Lecturer at the Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt (e-mail: amany.naim@azhar.edu.eg).

Neveen I. Ghali is a Professor of Faculty of Computers and Information Technology, Future University in Egypt, Cairo, Egypt (e-mail: neveen.ghali@fue.edu.eg).

In 2015 Seyedali proposed a novel nature-inspired algorithm called Ant Lion Optimizer (ALO) [13]. Ant Lion Optimizer copied the behavior of ant lions in nature. Where, it was based on research factors that represent a group of spiders that move collectively according to the biological behavior of the colony. This optimizer was implemented in various applications, authors in [14] improved the antlion optimization algorithm by modification random walk model and tested by using benchmark function. In [15] Preetha and Ashok implemented (ALO) algorithm in energy management problem.

This paper proposes a new optimization algorithm based on the venus flytrap plant movement. The movement of venus flytrap plant is an important feature of venus behavior, which is divided into three states the fully open state, the semi-closed state, and the fully closed state.

The remainder of this paper is organized as follows: Section II presents venus flytrap plant mechanism. Section III states venus flytrap optimization algorithm. Benchmark Functions are listed in Section IV. Section V presents the experimental results and analysis. Finally, the conclusion is presented in Section VI.

II. Venus Flytrap Plant Mechanism

Venus flytrap plant is a strange plant comprising of 5-7 leaves, each leaf is divided into upper and lower part. The leaf appears as two trapezoid projections collects by a midrib at the base. Every flap in the leaf contains 3 to 5 trigger hairs, which are delicate to any movement as a trap. These trigger hairs on the edges are like protrusions called cilia, that intertwine when the trap is closed to prevent prey from slipping away especially at the edges. Venus flytrap plant movement can be divided into three characteristic states [16]:

1) **The fully opened:**
   It happens in the absence of prey, which is identified by a convex bending of the trap lobes. It is presented in Figure 1(A).
2) **The semi-closed:**
   It happens immediately after the trap activation, which is identified by interlocking cilia that constrain large prey but allow the small prey to escape. It is presented in Figure 1(B, C).
3) **The fully closed:**
   It happens after prolonged stimulation, which is identified by tight oppression and recurved bending of the trap margins. It is presented in Figure 1(D).

The fast movement of water releases a highly elastic energy and causes a rapid change in the curvature of the lobes. Hence the impressive closing speed is essentially because of the rapid water transportation. By focusing on the mechanism for the venus flytrap plant, the movement

TABLE I
BENCHMARK FUNCTIONS

Name	Expression	Range	Min
$Sphere$	$f(z) = \sum_{i=1}^{n}(z_i^2)$	-100≤z≤100	$f(z^*) = 0$
$Schwefel's$ $Problem2.22$	$f(z) = \sum_{i=1}^{n}(\|z_i\|) + \prod_{i=1}^{n}(\|z_i\|)$	-10≤z≤10	$f(z^*) = 0$
$Schwefel's$ $Problem1.2$	$f(z) = \sum_{i=1}^{n}(\sum_{j=1}^{i}(z_i))^2$	-100≤z≤100	$f(z^*) = 0$
$Generalized$ $Rosenbrock's$ $Function$	$f(z) = \sum_{i=1}^{n-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$	-10≤z≤10	$f(z^*) = 0$
$Generalized$ $Schwefel's$ $Problem2.26$	$f(z) = -\sum_{i=1}^{n}(z_i \sin(\sqrt{\|z_i\|}))$	-500≤z≤500	$f(z^*) = -12569.5$
$Generalized$ $Rastrigin's$ $Function$	$f(z) = 10 + \sum_{i=1}^{n}[z_i^2 - 10\cos(2\Pi z_i)]$	-5.12≤z≤5.12	$f(z^*) = 0$
$Ackley's$ $Function$	$f(z) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2})$ $-exp\sqrt{(\frac{1}{n}\sum_{i=1}^{n}\cos(2\Pi z_i))} + 20 + exp$	-32.76≤z≤32.76	$f(z^*) = 0$
$Generalized$ $Griewank$ $Function$	$f(z) = 1 + \frac{1}{4000}\sum_{i=1}^{n}(z_i)^2 - \prod_{i=1}^{n}\cos(\frac{z_i}{\sqrt{i}})$	-600≤z≤600	$f(z^*) = 0$

TABLE II
THE PROPERTIES OF THE BENCHMARK FUNCTIONS

Name	Function	Property
$Sphere$	F1	(Continuous, Differentiable, Separable, Scalable, Multimodal)
$Schwefel's$ $Problem2.22$	F2	(Continuous, Differentiable, Non-Separable, Scalable, Unimodal)
$Schwefel's$ $Problem1.2$	F3	(Continuous, Differentiable, Non-Separable, Scalable, Unimodal)
$Generalized$ $Rosenbrock's$ $Function$	F4	(Continuous, Differentiable, Non-Separable, Scalable, Unimodal)
$Generalized$ $Schwefel's$ $Problem2.26$	F5	(Continuous, Differentiable, Separable, Scalable, Multimodal)
$Generalized$ $Rastrigin's$ $Function$	F6	(Continuous, Differentiable, Non-Separable, Scalable, Multimodal)
$Ackley's$ $Function$	F7	(Continuous, Differentiable, Non-separable, Scalable, Multimodal)
$Generalized$ $Griewank$ $Function$	F8	(Continuous, Differentiable, Non-Separable, Scalable, Multimodal)

process can be along the lines of a macroscopic level as follows [16]:

- The fast water movement, which can be modeled as follows:

$$WC = WS - WCO + WT \tag{1}$$

  Where WC is the water change rate, WS is the water supply rate, WCO is the water consumption rate, and WT is the water transport rate.
- Many plants have the ability to control the rates of transpiration by controlling the opening of the stomatal pores. This ability to keep aqueous tissue concentrations relatively constant. So, the total water volume of the lobe tissue is constant and can be normalized to 1 as follows:

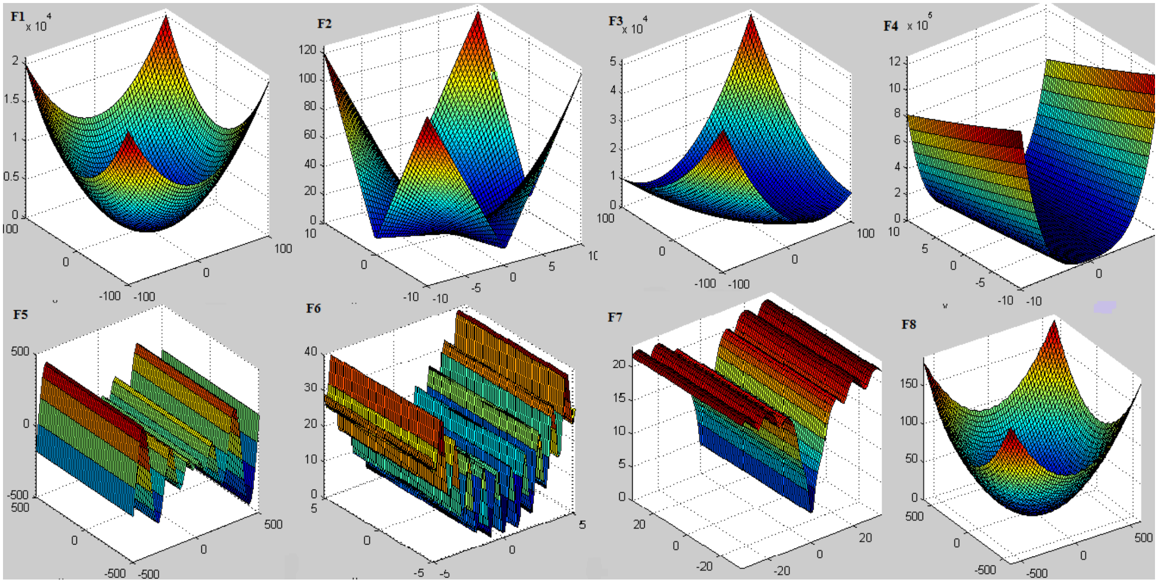
$$Z_O + Z_I = 1 \tag{2}$$

  Where Z_O and Z_I are the volume of water in the outer and the inner layer of the lobes respectively.

III. VENUS FLYTRAP OPTIMIZATION ALGORITHM

Venus Flytrap Optimization (VFO) is a stochasticoptimization algorithm and simulating the venus flytrap plant movement. In VFO, every solution is considered as state,

TABLE III
THE GRAPHS FOR BENCHMARK FUNCTIONS



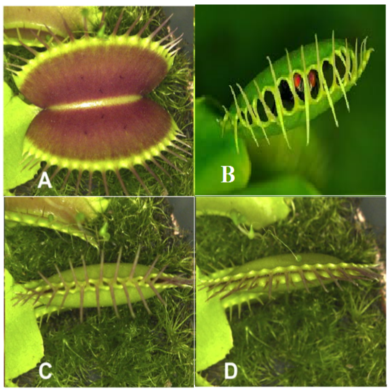


Figure 1: The Venus Flytrap Plant Movement.

each state constitution according to the reaction for the plant, and the combination of states constitutes the final decision for the plant. VFO uses the fast water movement, which can be described by water kinetics and can use a two-dimension system of the ordinary differential. The optimal state can be found using equations (3) and (4) as follows:

$$Z_O^* = \frac{\alpha Z_O^W}{Z_O^W - Z_I^W} - \psi Z_O \tag{3}$$

$$Z_I^* = \frac{\alpha Z_I^W}{Z_I^W - Z_O^W} - \psi Z_I \tag{4}$$

Where $\frac{\alpha Z_O^W}{Z_O^W - Z_I^W}$ and $\frac{\alpha Z_I^W}{Z_I^W - Z_O^W}$ are the water supply rate in the outer and the inner layer of the lobes respectively. α and ψ are representing the water consumption rate and the water supply rate respectively. $f(z_O)$ where $z_O = [z_{1O}, z_{2O}, ..., z_{dO}]$ and $f(z_I)$ where $z_I = [z_{1I}, z_{2I}, ..., z_{dI}]$ are the objective minimization functions. The cooperative coefficient is W. If $W = 1$, the dynamics of water can be balanced at any state in the line $Z_O + Z_I = 1$. Then, find $f(z_O)$ and $f(z_I)$ at each state. If $f(z_I) < f(z_O)$ then move to next state, otherwise back to the previous state.

The states of VFO are, respectively:

- Open state
- Semi-close state
- Close state

Venus Flytrap Optimization Algorithm is summarized as follows in **Algorithm 1**.

Algorithm 1 :Venus Flytrap Optimization Algorithm

Objective functions minimization $f(z_O), z_O = [z_{1O}, z_{2O}, ..., z_{nO}]$ and $f(z_I), z_I = [z_{1I}, z_{2I}, ..., z_{nI}]$, n is the number of population, α define the water consumption rate, ψ is the water supply rate, W is the cooperative coefficient, t is the iteration, MaxIter is maximum number of iteration, and d is the number of dimension.

```
while (t ≤ MaxIter) do
  Calculate fitness value for each state.
  for (each population n) do
    Find f(z_O) and f(z_I).
    if ( f(z_I)) < f(z_O) then
      Accept the next state.
    end if
    for (each dimension d) do
      Calculate new solution by equation [(3, 4)]
    end for
  end for
end while
```

IV. Benchmark Functions

Benchmark functions are used to validate the general performance of the optimization algorithm. In benchmark functions, there are a wide range of test functions that designed to emphasize various parts of the global optimization algorithm [17]. This section describes some classical benchmark functions, which are Sphere, $Schwefel's$ Problem 2.22, $Schwefel's$ Problem 1.2, Generalized $Rosenbrock's$

Function, Generalized $Schwefel's$ Problem 2.26, Generalized Rastrigrin Function, $Ackley's$ Function, and Generalized Griewank Function. These functions properties are either Unimodal, Multimodal, Differentiable, Non-Differentiable, Separable, Non-Separable, Scalable, or Non-Scalable [17], [18]. The purpose of Table I is to provide the basic information for each function. Where, Name is the name function, Expression is the mathematical equation for function, Range is the limits of variable z, and Min is the minimum value of function. Table II shows the properties of each function, where Function is the symbol for function and Property is the properties of function. Table III includes the graphical representation of each function.

V. Experimental Results and Analysis

The proposed algorithm is applied over eight predefined benchmark functions, which are defined in TABLE I, II, and III with three different population sizes (30, 40, and 50). The experimental results are shown in TABLE IV. Where, the best optimal value for each benchmark function is inversely proportion with the number of the population. For example, the Generalized $Rosenbrock's$ function gives best optimal value 0.793 at population number 50, 0.800 at population number 40 and 0.892 at population number 30.
The experimental results graphical representation of each benchmark function at different population size is illustrated in Figures 2-25. Where, $Sphere$ function (F1) is visualized at population size (30, 40, and 50) in Figures (2-4) respectively. $Schwefel's$ Problem 2.22 function (F2) is visualized at population size (30, 40, and 50) in Figures (5-7) respectively. While, $Schwefel's$ Problem 1.2 function (F3) is visualized at population size (30, 40, and 50) in Figures (8-10) respectively. The Generalized $Rosenbrock's$ Function (F4) is visualized at population size (30, 40, and 50) in Figures (11-13) respectively. The Generalized $Schwefel's$ Problem 2.26 function (F5) is visualized at population size (30, 40, and 50) in Figures (14-16) respectively. Generalized $Rastrigrin's$ Function (F6) is visualized at the same earlier population sizes in Figures (17-19) respectively. $Ackley's$ Function (F7) is visualized at population size (30, 40, and 50) in Figures (20-22) respectively. Finally, Generalized Griewank Function (F8) is visualized at population size (30, 40, and 50) in Figures (23-25) respectively.
TABLE V presents the experimental results for all above functions at population size 50 including the best solution, mean solution, and worst solution for each the function. The worst, the mean, and the best optimal values for $Sphere$ function are 2.711, 2.574, and 2.356 respectively. The worst, the mean, and the best optimal values for $Schwefel's$ Problem 2.22 are -1.112, -1.376, and -1.753 respectively. The worst value, the mean value, and the best optimal value for $Schwefel's$ Problem 1.2 are -5.987, -7.117, and -7.838 respectively. The worst, the mean, and the best optimal values for Generalized $Rosenbrock's$ Function are 0.886, 0.787, and 0.776 respectively. The worst, the mean, and the best optimal values for $Schwefel's$ Problem 1.2 are -5.987, -7.117, and -7.838 respectively. The worst, the mean, and the best optimal values for Generalized $Schwefel's$ Problem 2.26 are -452.453, -453.254, and -454.632 respectively. The worst, the mean, and the best optimal values for Generalized $Rastrigrin's$ Function are 305.943, 305.653, and 304.446 respectively. The worst, the mean, and the best optimal values for $Ackley's$ Function are 0.081, 0.078, and 0.062 respectively. The worst, the mean, and the best optimal values for Generalized Griewank Function are 0.819, 0.809, and 0.793 respectively.

TABLE IV
Experimental results for all functions at population size 30, 40, and 50

Benchmark Function	Number of Population	Best Optimal Value
Sphere	30	2.870
	40	2.364
	50	2.356
$Schwefel's$ *Problem 2.22*	30	-1.2268
	40	-1.5510
	50	-1.753
$Schwefel's$ *Problem 1.2*	30	-4.111
	40	-6.647
	50	-7.838
Generalized $Rosenbrock's$ *Function*	30	1.141
	40	0.799
	50	0.776
Generalized $Schwefel's$ *Problem 2.26*	30	-451.666
	40	-452.841
	50	-454.632
Generalized $Rastrigin's$ *Function*	30	402.921
	40	389.357
	50	304.446
$Ackley's$ *Function*	30	0.567
	40	0.113
	50	0.062
Generalized Griewank Function	30	0.892
	40	0.800
	50	0.793

Figures (26-33) show the number of failures at population size 30, 40, and 50. In Figure 26, the number of failures for $Sphere$ function are 11, 18, and 22 respectively. In Figure 27, the number of failures for $Schwefel's$ Problem 2.22 are 8, 11, and 11 respectively. In Figure 28, the number of failures for $Schwefel's$ Problem 1.2 are 5, 7, and 11 respectively. In Figure 29, the number of failures for Generalized $Rosenbrock's$ Function are 20, 29, and 29 respectively. In Figure 30, the number of failures for Generalized $Schwefel's$ Problem 2.26 are 4, 7, and 8 respectively. In Figure 31, the number of failures for Generalized Generalized $Rastrigrin's$ Function are 6, 9, and 15 respectively. In Figure 32, the number of failures for $Ackley's$ Function0 are 11, 16, and 19 respectively. In Figure 33, the number of failures for Generalized Griewank Function are 5, 9, and 10 respectively. By the analyzing these experimental results, the number of failures is directly proportion with the population size. From all the experimental results, the properties of VFO are continuous, differentiable, non-separable, scalable, and unimodal.

TABLE V
EXPERIMENTAL RESULTS FOR ALL FUNCTIONS AT POPULATION SIZE 50

Benchmark Function	Solution	Solution Value
Sphere	Worst	2.711
	Mean	2.574
	Best	2.356
Schwefel's Problem 2.22	Worst	-1.112
	Mean	-1.376
	Best	-1.753
Schwefel's Problem 1.2	Worst	-5.987
	Mean	-7.117
	Best	-7.838
Generalized Rosenbrock's Function	Worst	0.886
	Mean	0.787
	Best	0.776
Generalized Schwefel's Problem 2.26	Worst	-452.453
	Mean	-453.254
	Best	-454.632
Generalized Rastrigin's Function	Worst	305.943
	Mean	305.653
	Best	304.446
Ackley's Function	Worst	0.081
	Mean	0.078
	Best	0.062
Generalized Griewank Function	Worst	0.819
	Mean	0.809
	Best	0.793

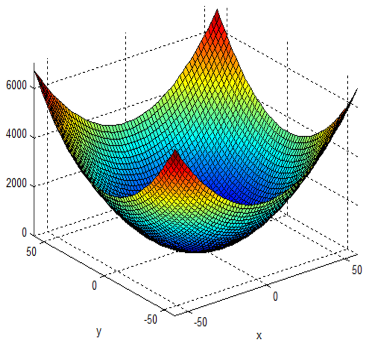


Figure 2: F1 with Population 30.

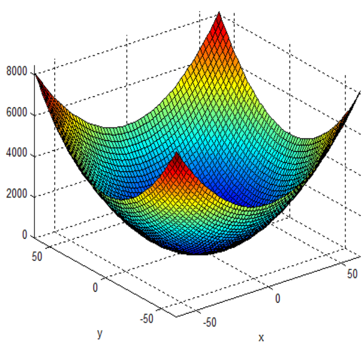


Figure 3: F1 with Population 40.

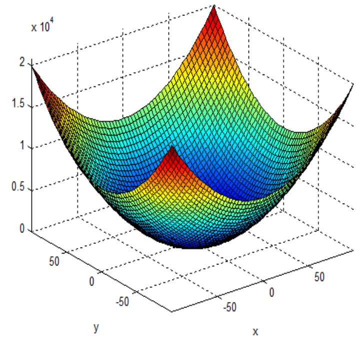


Figure 4: F1 with Population 50.

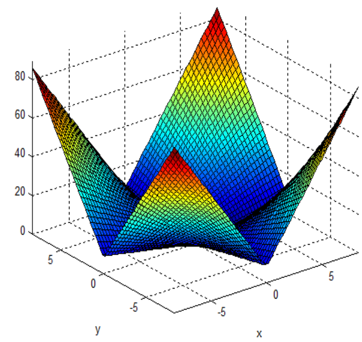


Figure 5: F2 with Population 30.

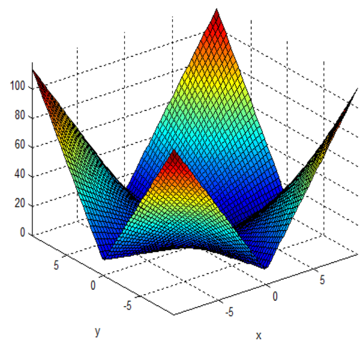


Figure 6: F2 with Population 40.

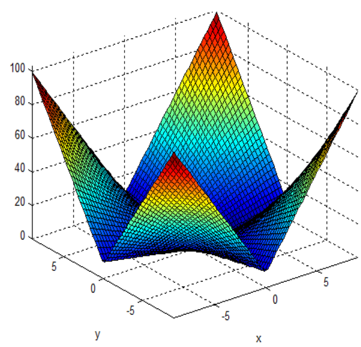


Figure 7: F2 with Population 50.

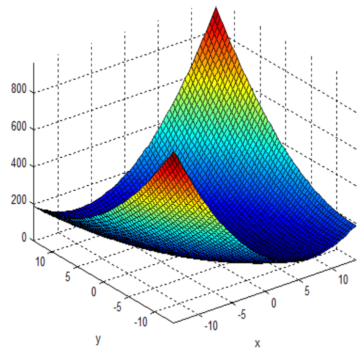


Figure 8: F3 with Population 30.

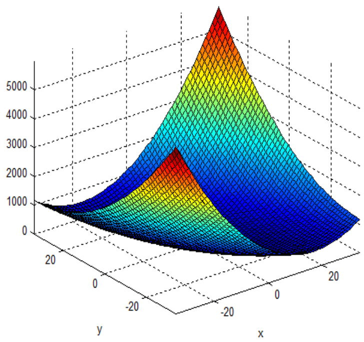


Figure 9: F3 with Population 40.

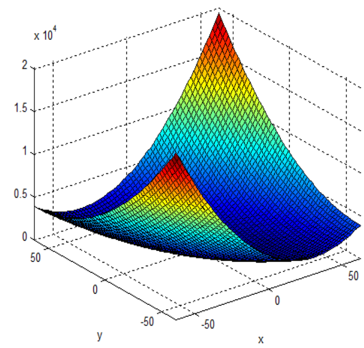


Figure 10: F3 with Population 50.

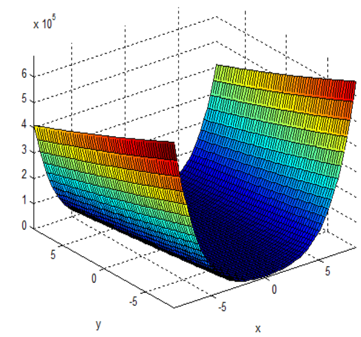


Figure 11: F4 with Population 30.

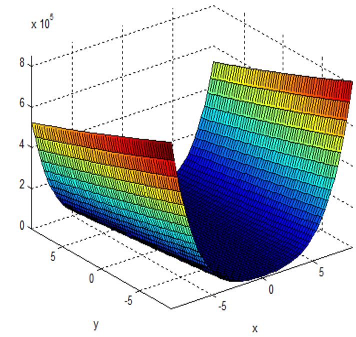


Figure 12: F4 with Population 40.

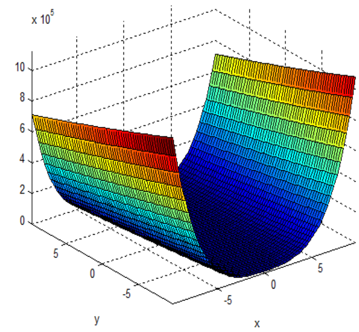


Figure 13: F4 with Population 50.

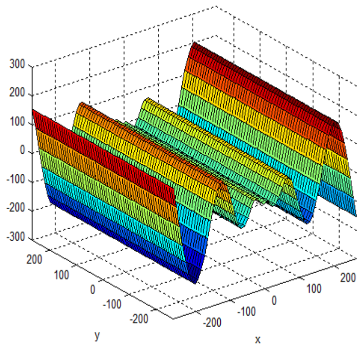


Figure 14: F5 with Population 30.

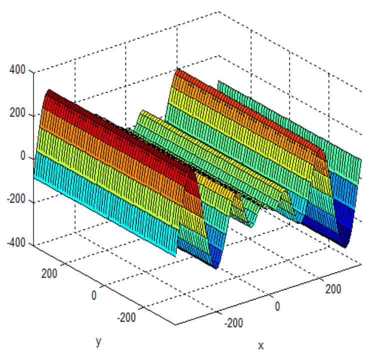


Figure 15: F5 with Population 40.

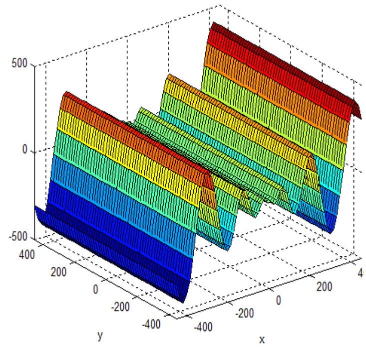


Figure 16: F5 with Population 50.

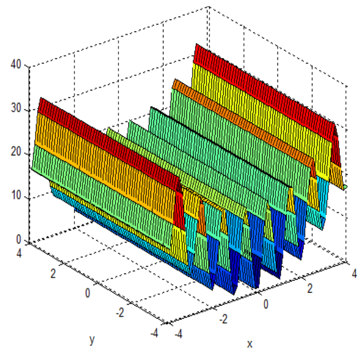


Figure 17: F6 with Population 30.

VI. Conclusion

Venus Flytrap Optimization is a new optimization algorithm, which proposed based on the behavior of venus flytrap plant. In this paper, the algorithm, the performance, and the hardiness of VFO are shown. Benchmark functions play importance role in the evaluation of algorithms and they are represented serious difficulties in obtaining a global minimization. The experimental results for VFO with benchmark problems are quite competitive and show the relation between the number of failures and the population size. From experimental results, it was found that the objective function is reached at a population size 50. Continuous, differentiable, non-separable, scalable, and unimodal are been the properties of VFO. In future work, will be made some improvements and extensions to convergence, preserving, and improving diversity. Also, deficiencies will be compensated by hybridization with the Evolutionary Computing (EC) models.

References

[1] B. Thomas, B. Juergen, M. Jorn, M. Olaf, "Evolutionary algorithms", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 4, no. 3, pp. 1-28, 2014.

[2] A. Ajay, B. Gururaj, S. Vasudha, T. Chandrasegar, "Nature Inspired Algorithm", 2017 International Conference on Trends in Electronics and Informatics (ICEI), IEEE, pp. 1131-1134, 2017.

[3] S. Santosa, R. A. Pramunendar, D. P. Prabowo, and Y. P. Santosa, "Wood Types Classification using Back-Propagation Neural Network based on Genetic Algorithm with Gray Level Co-occurrence Matrix for Features Extraction," IAENG International Journal of Computer Science, vol. 46, no.2, pp. 149-155, 2019.

[4] M. A. H. Akhand, Shahina Akter, M. A. Rashid, and S. B. Yaakob, "Velocity Tentative PSO: An Optimal Velocity Implementation based Particle Swarm Optimization to Solve Traveling Salesman Problem," IAENG International Journal of Computer Science, vol. 42, no.3, pp. 221-232, 2015.

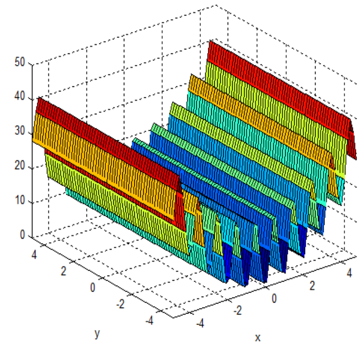


Figure 18: F6 with Population 40.

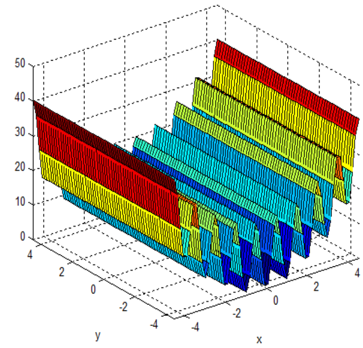


Figure 19: F6 with Population 50.

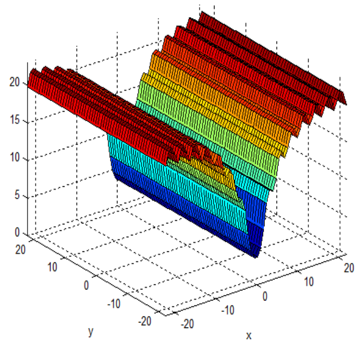


Figure 20: F7 with Population 30.

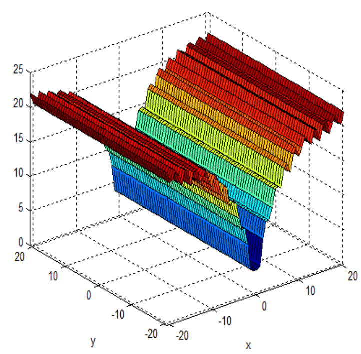


Figure 21: F7 with Population 40.

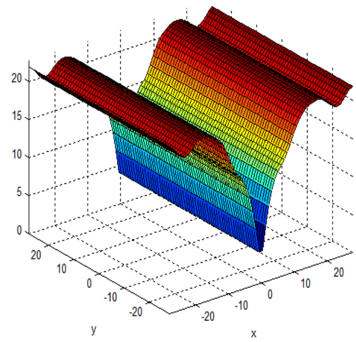


Figure 22: F7 with Population 50.

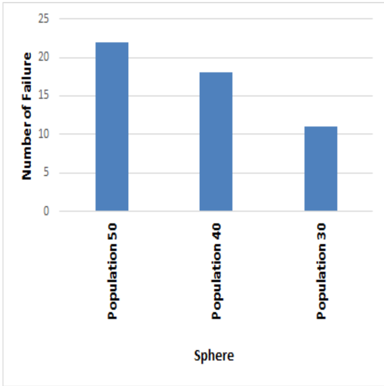


Figure 26: F1 with Number of Failure.

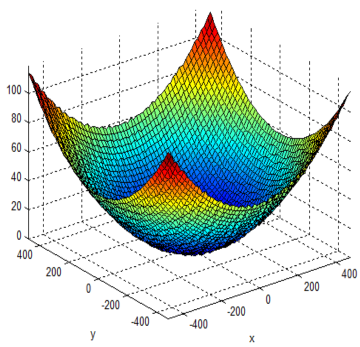


Figure 23: F8 with Population 30.

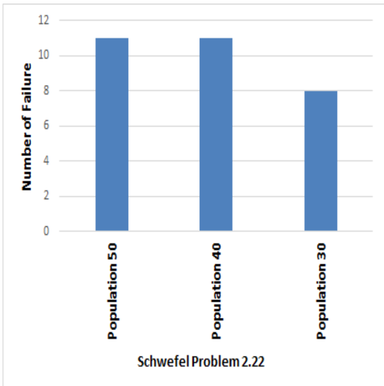


Figure 27: F2 with Number of Failure.

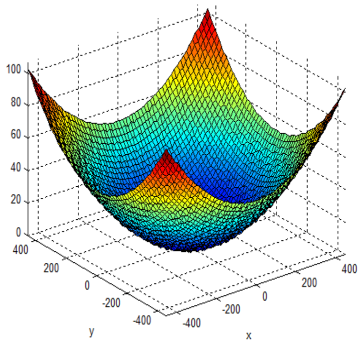


Figure 24: F8 with Population 40.

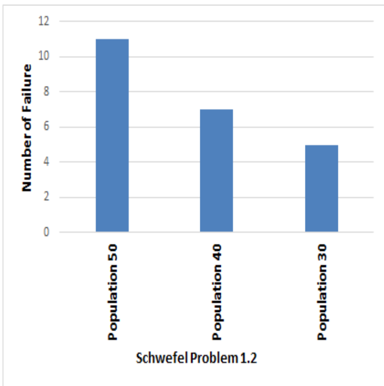


Figure 28: F3 with Number of Failure.

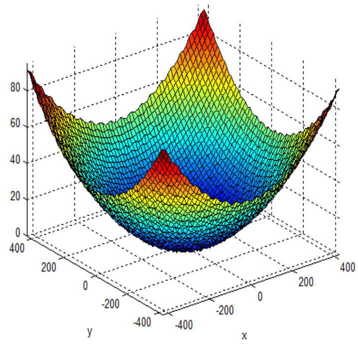


Figure 25: F8 with Population 50.

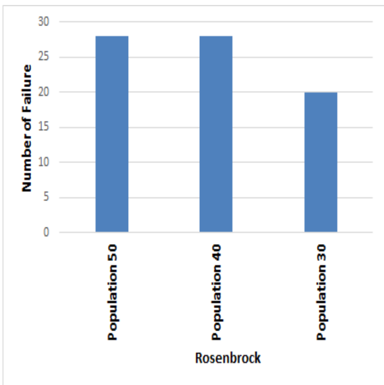


Figure 29: F4 with Number of Failure.

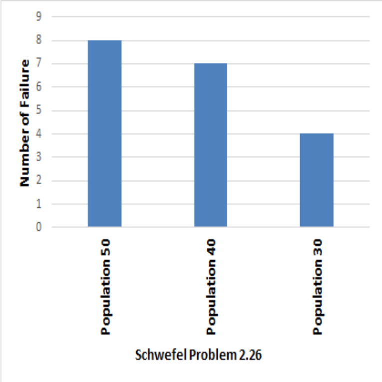


Figure 30: F5 with Number of Failure.

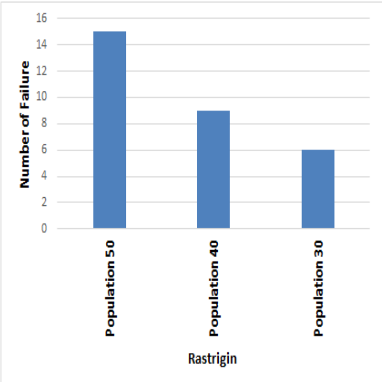


Figure 31: F6 with Number of Failure.

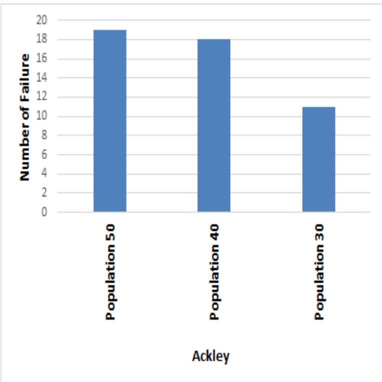


Figure 32: F7 with Number of Failure.

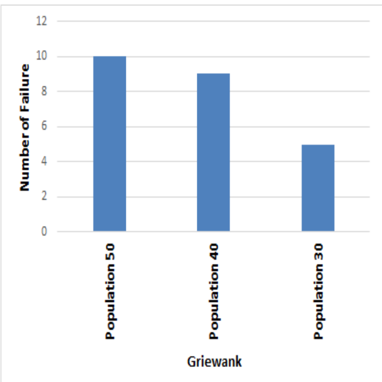


Figure 33: F8 with Number of Failure.

[5] Q. Qu, Y. Huang, X. Wang, and X. Chen, ”Complementary Differential Evolution-based Whale Optimization Algorithm for Function Optimization,” IAENG International Journal of Computer Science, vol. 47, no.4, pp. 805-815, 2020.
[6] Y. Sun, J. Wang, and X. Ma, ”An Improved Krill Herd Algorithm Based on Meme Grouping Strategy of Shuffled Frog Leaping Algorithm,” IAENG International Journal of Computer Science, vol. 46, no.2, pp. 156-162, 2019.
[7] Y. Xin-She and D. Suash, ”Cuckoo search via lvy flights”, 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, pp. 210-214, 2009.
[8] P. Jothibasu, and P. Rangarajan, ”Spatio and Efficient l1-l1 minimization based Impulse Noise Removal in Gray Images Using Dictionary Learning,” IAENG International Journal of Computer Science, vol. 42, no.3, pp. 160-173, 2015.
[9] H. Bilal and J. David, ”Hybridizing the Cuckoo Search Algorithm with Different Mutation Operators for Numerical Optimization Problems”, Journal of Intelligent Systems, vol. 29, no. 1, pp. 1043-1062, 2018.
[10] J. James and O. Victor, ”A social spider algorithm for global optimization”, Applied Soft Computing, Elsevier, vol. 30, pp. 614-627, 2015.
[11] B. Emine and Ü. Erkan, ”A binary social spider algorithm for continuous optimization task”, Soft Computing, pp. 1-10, 2020.
[12] A. Fatih , G. Fatih, Y. Tuncay, ”Improved Social Spider Algorithm via Differential Evolution”, Artificial Intelligence and Applied Mathematics in Engineering Problems, The International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2019), pp.437-445, 2020.
[13] M. Seyedali, ”The ant lion optimizer”, Advances in Engineering Software, Science Direct, vol. 83, pp. 80-98, 2015.
[14] K. Haydar and Y. Ugur, ”Improved antlion optimization algorithm,” 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, pp. 84-88, 2017.
[15] P. Preetha and K. Ashok, ”Implementation of Ant-Lion Optimization Algorithm in Energy Management Problem and Comparison”, Advances in Decision Sciences, Image Processing, Security and Computer Vision, pp.462-469, 2020.
[16] Y. Ruoting, L. Scott, Z. Mingjun, X. Lijin, ”A mathematical model on the closing and opening mechanism for Venus flytrap”, Plant Signaling & Behavior, vol. 5, no. 8, pp. 968-978, 2010.
[17] K. Hasan and C. Rahime,”A PSO based approach: Scout particle swarm algorithm for continuous global optimization problems”, Journal of Computational Design and Engineering, vol. 6, no. 2, pp. 129-142, 2019.
[18] J. Momin and Y. Xin-She, ”A Literature Survey of Benchmark Functions For Global Optimization Problems”, Int. Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, pp. 150-194, 2013.