Fertilization optimization algorithm on CEC2015 and large scale problems

Hazim Nasir Ghafik1,2, Shaymaa Alsamia3 and Károly Jármair1

1 Institute of Energy Engineering and Chemical Machinery, Faculty of Mechanical Engineering, University of Miskolc, Miskolc, H-3515, Miskolc-Egyetemváros, Hungary
2 Department of Mechanical Engineering, Faculty of Engineering, University of Kufa, Iraq
3 Department of Structures and Water Resources, Faculty of Engineering, University of Kufa, Iraq

Received: December 29, 2020 • Revised manuscript received: August 3, 2021 • Accepted: August 5, 2021
Published online: November 1, 2021

ABSTRACT

This work, presents a novel optimizer called fertilization optimization algorithm, which is based on levy flight and random search within a search space. It is a biologically inspired algorithm by the fertilization of the egg in reproduction of mammals. The performance of the algorithm was compared with other optimization algorithms on CEC2015 time expensive benchmarks and large scale optimization problems. Remarkably, the fertilization optimization algorithm has overcome other optimizers in many cases and the examination and comparison results are encouraging to use the fertilization optimization algorithm in other possible applications.

KEYWORDS

fertilization optimization algorithm, optimization, biologically inspired algorithms, artificial intelligence, metaheuristics

1. INTRODUCTION

During its history, optimization algorithms have been inspired by natural or human-made phenomena to introduce mathematical formulation that can solve problems in different fields of sciences. Specifically, optimization algorithms used to find the maximum or minimum of a function, and they have a wide range of applications in the industry [1] and engineering problems like as robotic [2] and structures [3]. Developers are more interested in phenomena that could inspire them to develop a new method that can solve new problems or find the best solutions for the existing ones. One of the inspiration engines is flock of animal, birds, and insects that lead to developing swarm intelligence [4, 5] methods; this term can be defined as accumulative and shared knowledge among a group of individuals, and this kind of intelligence cannot be reached by one of them alone. Examples of swarm intelligence Particle Swarm Optimization (PSO) [6], Artificial Bee Colony (ABC) [7], and Grey Wolf Optimization (GWO) [8]. Not all the biologically inspired algorithms are swarm intelligence; bacteria and invasive weeds optimization do not follow the rules of a swarm. In this article, a biologically inspired algorithm from the fertilization process in the reproductive tract of mammal animals during reproduction is presented. The new algorithm is called Fertilization Optimization (FO) algorithm. Computationally expensive benchmarks CEC2015 [9] are employed during experiments. On these mathematical optimization problems, FO was compared with other meta heuristics. Remarkably, FO has shown great performance and overcome many other algorithms in many cases. The variety and difficulty of the mathematical optimization problems that FO could pass through successfully have proved the reliability of the fertilization algorithm for mathematical optimization. In brief, the FO algorithms can be described as follows.
Each solution have a position \( (X) \) and velocity \( (v) \) in the search space. For each iteration, the velocity decreased by some value \( \delta \)

\[
v^{t+1} = \delta v^t, \quad 0 < \delta < 1,
\]

\[
V_{t+1} = V_t e^{-\frac{t}{\tau}},
\]

where \( t \) is the number of iteration in the optimization process. The solutions move in the search space using levy flight \( L \) and the solution is updated by the following equation:

\[
X^t_{i+1} = L(X^t_i - V^t_i), \quad (i = 1, 2, \ldots, n),
\]

where \( i \) the index of solution components, and \( n \) is the total number of variables in the solution (3). The average value of the best \( X^t_{first} \), medium best \( X^t_{middle} \) and worst solutions \( X^t_{end} \) can also have effect on the update solution process:

\[
X^t_{i+1} = \frac{X^t_{first} + X^t_{middle} + X^t_{end}}{3},
\]

The combination of equations (1)–(4) give the search engine of the F algorithm:

\[
X^{t+1}_{ij} = X^{t}_{ij} - V^{t}_{ij} e^{-\frac{t}{\tau}} + L \left(X^{t}_i - V^{t}_j\right) - \frac{X^{t}_{first} + X^{t}_{middle} + X^{t}_{end}}{3}, \quad (j = 1, 2 \ldots, m),
\]

where \( m \) is the number of variables in the proposed solution, and the pseudocode can be seen in Code 1.

### 2. RESULTS AND DISCUSSION

CEC2015 benchmark functions, which are described in Tables 1 and 2, are used in this study to examine the performance of the FO algorithm. The run conditions on CEC2015 experiment are: variable dimensions 10, population size 10, maximum number of iterations 1,000, and 20 independent runs. Firstly, FO algorithm is compared with Hybrid Particle Swarm Optimization algorithm and FireFly algorithm (HPSOFF) [10], and Hybrid Firefly and Particle Optimization (HFPO) algorithm [11]. Tables 3 and 4 show...
Table 2. CEC2015 expensive benchmark problems F10 to F15

| Type              | No. | Description                  | \( f_{\text{min}} \) |
|-------------------|-----|------------------------------|------------------------|
| Hybrid functions  | F10 | Hybrid Function 1 \((N = 3)\) | 1,000                  |
|                   | F11 | Hybrid Function 2 \((N = 4)\) | 1,100                  |
|                   | F12 | Hybrid Function 3 \((N = 5)\) | 1,200                  |
| Composition Functions | F13 | Composition Function 1 \((N = 5)\) | 1,300                  |
|                   | F14 | Composition Function 2 \((N = 3)\) | 1,400                  |
|                   | F15 | Composition Function 3 \((N = 5)\) | 1,500                  |

Table 3. Standard deviation results of the FO algorithm vs. HPSOFF and HFPSO on CEC2015

|                  | HPSOFF | HPSPO | FO  |
|------------------|--------|-------|-----|
| F10              | 3.4292E+07 | 6.6375E+06 | 0E+00 |
| F11              | 1.2383E+04 | 1.5886E+04 | 1.9569E-06 |
| F12              | 1.5636E+00 | 1.4189E+00 | 7.0195E-02 |
| F13              | 3.0718E+02 | 3.9905E+02 | 1.8913E+00 |
| F14              | 8.2275E-01 | 7.4666E-01 | 2.0303E+02 |
| F15              | 1.4977E-01 | 1.4858E-01 | 7.1663E-10 |

Table 4. Average solutions results of the FO algorithm vs. HPSOFF and HFPSO on CEC2015

|                  | HPSOFF | HPSPO | FO  |
|------------------|--------|-------|-----|
| F10              | 4.8387E+07 | 1.3768E+07 | 7.0974E+07 |
| F11              | 3.8331E+04 | 3.8542E+04 | 1.1254E+10 |
| F12              | 3.0474E+02 | 3.6071E+02 | 2.0492E+02 |
| F13              | 1.7084E+03 | 1.3159E+03 | 4.8685E+02 |
| F14              | 5.0273E+02 | 5.0250E+02 | 2.1652E+03 |
| F15              | 6.0063E+02 | 6.0045E+02 | 1.6116E+06 |
| F16              | 7.0087E+02 | 7.0060E+02 | 7.5666E+02 |
| F17              | 8.0740E+02 | 8.0773E+02 | 1.6292E+05 |
| F18              | 9.0388E+02 | 9.0393E+02 | 1.0413E+03 |
| F19              | 3.5402E+05 | 3.3096E+05 | 6.8481E+04 |
| F20              | 1.1067E+03 | 1.1074E+03 | 1.4195E+03 |
| F21              | 1.4157E+03 | 1.3983E+03 | 1.3391E+03 |
| F22              | 1.6333E+03 | 1.6452E+03 | 1.3980E+03 |
| F23              | 1.6053E+03 | 1.6021E+03 | 1.5594E+04 |
| F24              | 1.8365E+03 | 1.9233E+03 | 2.0528E+03 |

Table 5. Standard deviation results of the FO algorithm vs. HPSOFF and HFPSO on CEC2015

| Type              | No. | Description                  | \( f_{\text{min}} \) |
|-------------------|-----|------------------------------|------------------------|
| Hybrid functions  | F10 | Hybrid Function 1 \((N = 3)\) | 1,000                  |
|                   | F11 | Hybrid Function 2 \((N = 4)\) | 1,100                  |
|                   | F12 | Hybrid Function 3 \((N = 5)\) | 1,200                  |
| Composition Functions | F13 | Composition Function 1 \((N = 5)\) | 1,300                  |
|                   | F14 | Composition Function 2 \((N = 3)\) | 1,400                  |
|                   | F15 | Composition Function 3 \((N = 5)\) | 1,500                  |

Table 6. Average solutions results of the FO algorithm vs. HPSOFF and HFPSO on CEC2015

| Type              | No. | Description                  | \( f_{\text{min}} \) |
|-------------------|-----|------------------------------|------------------------|
| Hybrid functions  | F10 | Hybrid Function 1 \((N = 3)\) | 1,000                  |
|                   | F11 | Hybrid Function 2 \((N = 4)\) | 1,100                  |
|                   | F12 | Hybrid Function 3 \((N = 5)\) | 1,200                  |
| Composition Functions | F13 | Composition Function 1 \((N = 5)\) | 1,300                  |
|                   | F14 | Composition Function 2 \((N = 3)\) | 1,400                  |
|                   | F15 | Composition Function 3 \((N = 5)\) | 1,500                  |

The results of comparison on mean solutions and standard deviation among FO, HPSOFF, and HFPSO.

Table 5 reveals the comparison on standard deviation results among FO, PSO, FFPSSO algorithm [12], and FireFly (FF) algorithm while Table 6 reveals the comparison on mean solutions results among the same algorithms in Table 5.

Another experiment has been done to compare the performance of the FO algorithm on large scale optimization problems against Ant Lion Optimizer ALO [13], Butterfly Optimization Algorithm (BOA) [14], GWO [7], PSO, Sine Cosine Algorithm (SCA) optimization [15], Dynamic Differential Annealed Optimization (DDAO) [16], Bat Algorithm (BA) [17], and Tree-Seed Algorithm (TSA) [18]. Tables 7 and 8 illustrate the statistical results for this test in terms of best solution (Best), worst solution (Worst), mean solution (Mean), and Standard Deviation (STD). Four large scale optimization problems are chosen in these experiments, and the run conditions are: variable dimensions 1,000, population size 25, number of iterations 100, and 51 independent runs. The description and formulation of the large scale problems can be written as follows:

- **F16**: Rastrigin: \[ f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i)] \]

  \[ \text{Range} = [-5.12, 5.12], \quad f_{\text{min}} = 0, \]

- **F17**: \[ f(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \]

  \[ \text{Range} = [-2.048, 2.048], \quad f_{\text{min}} = 0, \]

- **F18**: \[ f(x) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right) - \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^{d} \cos (c x_i) } + a + \exp(1), \right) \]

  \[ \text{Range} = [-32.768, 32.768], \quad f_{\text{min}} = 0, \]

- **F19**: \[ f(x) = \sum_{i=1}^{d} \left( \frac{x_i^2}{\sqrt{v_i}} - \Pi_{i=1}^{d} \cos \left( \frac{x_i}{v_i} \right) \right) + 1, \]

  \[ \text{Range} = [-600, 600], \quad f_{\text{min}} = 0. \]

The FO algorithm is less efficient on high-degree multimodal benchmarks, and this behavior can be seen on the statistical results. The experimental results show that the FO algorithm is more effective on large scale optimization than small scale. The behavior on large and small scale problems needs a dedicated study that can be suggested for a future work. In brief, the FO algorithm can be stable and fast convergent on unimodal optimization problems as well as its efficiency on large scale problems.
Table 5. Standard deviation results of the FO algorithm vs. PSO, FF, and FFPSO on CEC2015

|       | FO       | FF         | FFPSO     | PSO       |
|-------|----------|------------|-----------|-----------|
| F1    | 1.3549E+08| 2.8495E+08 | 4.9786E+09| 0E+00     |
| F2    | 1.5114E+04| 9.7404E+03 | 4.6261E+08| 1.959E-06 |
| F3    | 1.3529E+00| 1.2487E+00 | 1.6124E+00| 7.0195E-02|
| F4    | 3.5521E+02| 3.2112E+02 | 2.6203E+02| 1.8913E+00|
| F5    | 6.3611E-01| 5.9796E-01 | 9.3430E-01| 2.0303E+02|
| F6    | 2.8490E-01| 5.8361E-01 | 1.3580E+00| 7.1663E-10 |
| F7    | 1.8947E+00| 5.8077E+00 | 3.3292E+01| 6.1138E+00|
| F8    | 2.7690E+01| 1.7256E+02 | 2.7423E+05| 4.7333E+04 |
| F9    | 3.2749E-01| 1.7838E-01 | 4.6658E-13|           |
| F10   | 1.9786E+05| 6.5054E+05 | 7.7896E+07| 8.8424E+04|
| F11   | 2.9153E+00| 2.4020E+00 | 6.7179E+01| 0E+00     |
| F12   | 1.1574E+02| 9.1615E+01 | 4.3894E+02| 2.9857E-01|
| F13   | 1.9141E+01| 2.9519E+01 | 8.9197E+02| 3.3013E+01|
| F14   | 4.5254E+00| 3.5980E+00 | 4.2292E+01| 2.8957E+02|
| F15   | 1.4570E+02| 7.4514E+01 | 1.0567E+02| 6.8567E+00|

Table 6. Average solutions results of the FO algorithm vs. PSO, FF, and FFPSO on CEC2015

|       | FO       | FF         | FFPSO     | PSO       |
|-------|----------|------------|-----------|-----------|
| F1    | 2.4533E+08| 4.3059E+08 | 1.6287E+10| 7.0974E+07|
| F2    | 3.8112E+04| 3.3304E+04 | 1.4957E+08| 1.1254E+10|
| F3    | 3.0779E+02| 3.0773E+02 | 3.1455E+02| 3.2049E+02|
| F4    | 2.2534E+03| 1.5473E+03 | 3.1120E+03| 4.6855E+02|
| F5    | 5.0277E+02| 5.0293E+02 | 5.0350E+02| 2.1652E+03|
| F6    | 6.0098E+02| 6.0092E+02 | 6.0673E+02| 1.6116E+06|
| F7    | 7.0193E+02| 7.0586E+02 | 8.0586E+02| 7.5666E+02|
| F8    | 8.1583E+02| 8.6344E+02 | 2.7632E+05| 1.6292E+05|
| F9    | 9.0391E+02| 9.0395E+02 | 9.0451E+02| 1.0413E+03|
| F10   | 2.9540E+05| 5.3162E+05 | 5.1186E+07| 6.8481E+04|
| F11   | 1.1088E+03| 1.1080E+03 | 1.2198E+03| 1.4195E+03|
| F12   | 1.4620E+03| 1.3995E+03 | 2.1953E+03| 1.3391E+03|
| F13   | 1.6415E+03| 1.6437E+03 | 3.0005E+03| 1.3908E+03|
| F14   | 1.6076E+03| 1.6111E+03 | 1.6770E+03| 1.5594E+04|
| F15   | 1.9149E+03| 1.9269E+03 | 2.1840E+03| 2.0528E+03|

Table 7. Results for large scale optimization on F16 and F17

| Function | F16    | F17    |
|----------|--------|--------|
| ALO      | 2.4634E+04 | 9.6145E+04 |
| BOA      | 1.5088E+04 | 2.6251E+04 |
| GWO      | 3.3526E+04 | 9.8982E+02 |
| PSO      | 6.3429E+03 | 3.3616E+03 |
| SCA      | 7.4800E+03 | 7.2573E+03 |
| DDAO     | Mean   | 1.7431E+03 | 2.3237E+05 |
| BA       | Mean   | 1.5735E+00 | 1.4957E+04 |
| TSA      | Mean   | 1.4455E+09 | 1.6689E+05 |
| FO       | Mean   | 1.4142E+03 | 7.7892E-04 |

Table 7. Continued

| Function | F16    | F17    |
|----------|--------|--------|
| Mean     | 4.3059E+08 | 1.6287E+10 |
| STD      | 9.1819E+02 | 3.8524E+04 |
| Worst    | 2.2573E-01 | 9.8897E+02 |
| Mean     | 5.7423E+02 | 1.0382E+03 |
| STD      | 1.0834E+03 | 8.9853E+01 |
| Worst    | 1.2267E+00 | 4.5067E+04 |
| Mean     | 1.4585E+04 | 1.8689E+05 |
| STD      | 1.3142E+03 | 7.7892E+04 |
| Worst    | 6.3653E+03 | 1.2015E+04 |
| Mean     | 1.4767E+00 | 5.2552E+04 |
| STD      | 1.0217E+04 | 2.7948E+04 |
| Worst    | 2.1618E+03 | 9.2913E+03 |
| Mean     | 0.0000E+00 | 9.9889E+00 |
| STD      | 0.0000E+00 | 9.9896E+02 |
Table 8. Results for large scale optimization on F18 and F19

| Function | F18 | F19 |
|----------|-----|-----|
| ALO      |     |     |
| Best     | 1.9469E+01 | 9.8149E+03 |
| Worst    | 2.0307E+01 | 1.4948E+04 |
| Mean     | 1.9750E+01 | 1.1345E+04 |
| STD      | 2.4107E-01 | 1.4212E+03 |
| BOA      |     |     |
| Best     | 1.2957E-08 | 2.0416E-06 |
| Worst    | 6.1917E-07 | 5.9821E-04 |
| Mean     | 5.9824E-08 | 5.5767E-05 |
| STD      | 9.7536E-08 | 1.2535E-04 |
| GWO      |     |     |
| Best     | 9.1437E+00 | 6.8521E+02 |
| Worst    | 1.2941E+01 | 1.4847E+03 |
| Mean     | 1.0128E+01 | 9.7278E+02 |
| STD      | 9.8562E-01 | 1.4800E+02 |
| PSO      |     |     |
| Best     | 1.7001E+01 | 8.1418E+02 |
| Worst    | 1.7812E+01 | 1.0013E+03 |
| Mean     | 1.7356E+01 | 9.2068E+02 |
| STD      | 1.6680E-01 | 4.4017E+01 |
| SCA      |     |     |
| Best     | 7.0434E+00 | 2.0794E+03 |
| Worst    | 1.9671E+01 | 1.2949E+04 |
| Mean     | 1.6601E+01 | 8.5366E+03 |
| STD      | 3.2924E+00 | 2.5264E+03 |
| DDAO     |     |     |
| Best     | 2.4380E-02 | 1.3322E+00 |
| Worst    | 9.5690E+00 | 6.0214E+02 |
| Mean     | 3.3104E+00 | 8.4837E+01 |
| STD      | 2.1546E+00 | 5.1576E+02 |
| BA       |     |     |
| Best     | 1.9362E+01 | 1.0246E+04 |
| Worst    | 2.1148E+01 | 8.2739E+04 |
| Mean     | 2.0312E+01 | 1.7242E+04 |
| STD      | 4.7706E-01 | 4.9752E+03 |
| TSA      |     |     |
| Best     | 6.4128E+00 | 3.1366E+02 |
| Worst    | 1.2601E+01 | 3.3672E+03 |
| Mean     | 9.0431E+00 | 1.0870E+03 |
| STD      | 1.5954E+00 | 5.1949E+02 |
| FO       |     |     |
| Best     | 8.9617E-13 | 0.0000E+00 |
| Worst    | 2.4896E-07 | 7.2283E-07 |
| Mean     | 5.0366E-09 | 1.4257E-08 |
| STD      | 3.4971E-08 | 1.0120E-07 |

3. CONCLUSION

The fertilization optimization algorithm is a powerful biologically inspired algorithm developed for mathematical optimization problems. It mimics the interaction between sperms and uterus in the process of fertilization the egg. The statistical results on 19 test functions; CEC2015 time expensive benchmarks, unimodal, multimodal, small scale, and large scale problems have shown the efficiency of the proposed algorithm compared with many optimization algorithms. During examinations of the FO algorithm, it has been noticed that the performance of the FO algorithm on large scale problems is better than its performance on small scale problems. The statistical results illustrate that FO algorithm is stable with less STD and best solutions than other eight competitive. The FO algorithm has proven its powerful on unimodal functions and it has promising applications on continuous differentiable objective functions and large scale optimization. The FO algorithm is fast and simple and can efficiently skip local points in the search space and go-ahead to the global point.

ACKNOWLEDGMENTS

The research was supported by the Hungarian National Research Development and Innovation Office-NKFIH under the project number K 134358.

REFERENCES

[1] H. N. Ghafl and K. Jármai, “Research and application of industrial robot manipulators in vehicle and automotive engineering, a survey,” in Vehicle and Automotive Engineering 2, Lecture Notes in Mechanical Engineering, K. Jármai and B. Bolló, Eds, Springer, 2018, pp. 611–623.

[2] H. N. Ghafl and A. H. Mohammed, “A virtual reality environment for 5-DOF robot manipulator based on XNA framework,” Int. J. Cotputer Appl., vol. 113, no. 3, pp. 33–37, 2015.

[3] H. N. Ghafl and K. Jármai, “Kinematic-based structural optimization of robots,” Pollack Period., vol. 14, no. 3, pp. 213–222, 2019.

[4] F. Figueiredo, M. Macedo, H. V. Siqueira, C. J. Santana, Jr, A. Gokhale, and C. J. A. Bastos-Filho, “Swarm intelligence for clustering-A systematic review with new perspectives on data mining,” Eng. Appl. Artif. Intell., vol. 82, pp. 313–329, 2019.

[5] P. Šulek and T. Kinczer, “Expert control system of shipping operation on the Gabčíkovo project,” Pollack Period., vol. 14, no. 1, pp. 139–150, 2019.

[6] S. Alsamia, D. S. Ibrahim, and H. N. Ghafl, “Optimization of drilling performance using various metaheuristics,” Pollack Period., vol. 16, no. 2, pp. 80–85, 2021.

[7] H. Ghafl and K. Jármai, “Comparative study of particles warm optimization and artificial bee colony algorithms,” in Multiscience XXXII. MicroCAD International Multidisciplinary Scientific Conference, Miskolc-Egyetemváros, Hungary, Sep. 5–8, 2018, pp. 1–6.

[8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Adv. Eng. Softw., vol. 69, pp. 46–61, 2014.

[9] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, “Problem definitions and evaluation criteria for the CEC2015 competition on learning-based real-parameters in gie objective optimization,” Tech. Rep. 201411A, Comput. Intell. Lab. Zhengzhou Univ., Zhengzhou China Nanyang Technol. Univ. Singapore, vol. 29, pp. 625–640, 2014.

[10] S. Arunachalam, T. A. Bhomila, and M. R. Babu, “Hybrid particles warm optimization algorithm and firefly algorithm based combined economic and emission dispatch including valve point effect,” in International Conference on Swarm, Evolutionary, and Memetic Computing. Swarm, vol. 8947, B. Panigrahi, P. Suganthan, and S. Das, Eds, Springer, 2014, pp. 647–660.

[11] P. Kora and K. S. R. Krishna, “Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block,” Int. J. Cardiovasc. Acad., vol. 2, no. 1, pp. 44–48, 2016.

[12] I. B. Aydilek, “A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems,” Appl. Soft Comput., vol. 66, pp. 232–249, 2018.
[13] S. Mirjalili, “The ant lion optimizer,” Adv. Eng. Softw., vol. 83, pp. 80–98, 2015.

[14] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, “BOA: The Bayesian optimization algorithm,” in Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, Orlando Florida, Jul. 13, 1999, pp. 525–532.

[15] S. Mirjalili, “SCA: a sine cosine algorithm for solving optimization problems,” Knowledge-Based Syst., vol. 96, pp. 120–133, 2016.

[16] H. N. Ghafil and K. Jármai, “Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications,” Appl. Soft Comput., 2020, Paper no. 106392.

[17] X. S. Yang, “A new metaheuristic bat-inspired algorithm,” in Nature Inspired Cooperative Strategies for Optimization Studies in Computational Intelligence, vol. 284, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds, Berlin, Heidelberg: Springer, 2010, pp. 65–74.

[18] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,” Eng. Appl. Artif. Intell., vol. 90, 2020, Paper no. 103541.