Social Spider Optimization Algorithm: Theory and its Applications

D. Evangeline, T. Abirami

***Abstract:** An extensive variety of optimization problems are solved by swarm intelligence algorithms that are modelled based on the animal or insect behaviour while living in groups. One such recent swarm intelligence algorithm is Social Spider Optimization (SSO). This paper thoroughly reviews and analyses the characteristics of this meta-heuristic algorithm. Since the existing literature of this algorithm is comparatively limited, the paper discusses the research ideas presented in such existing works and classifies the literature on basis of the application areas like image processing, optical flow, electric circuits, neural networks and basic sciences. It also sets a basis for research applications of the algorithm in order to tap the complete potential of the algorithm in other areas to achieve desired results.*

***Index Terms:** Image Processing, Meta-heuristics, Neural Networks, Social Spider Optimization*

I. INTRODUCTION

Artificial Intelligence (AI) is a division in computer science that focuses on imparting human intelligence to computers. Generally, the solution to problems in AI rests with exploring the solution space comprising of many possible solutions [1]. Exhaustive search [2], also known as brute-force approach is a well-known traditional problem solving method wherein all the possible candidate solutions satisfying the specific constraints are applied. Some significant exhaustive search approach includes Local Search, Branch and Bound, Dynamic Programming, etc., Though this method delivers the exact best solution, it miserably fails to solve real world problems since the solution space in this case is huge. Also, exploration of such a large solution space requires exorbitant computation cost. Hence, heuristic algorithms are chosen to answer problems in real world. Such heuristics attempt to determine a fairly good solution (approximate to the exact solution) to the problem within a reasonable time frame. Moreover, these algorithms do not converge to the solution at a quick pace thus avoiding convergence to the local optimum. Generally, these algorithms play a significant role in optimization. Heuristic algorithms [3] are broadly classified as – Evolutionary algorithms [4], Swarm Intelligence [5], Neural algorithms [6], Immune algorithms [7], Stochastic [8], Physics-related [9], Fuzzy algorithms [10], etc., Taxonomy of these optimization algorithms is illustrated in Fig 1. Evolutionary algorithm (EA) adopts the biological processes of reproduction, mutation, recombination and selection to solve problems. Genetic Algorithm (GA) [11] is one of the widely adopted evolutionary algorithms. Bio-geography based optimization (BBO) [12] estimates the strength of biological species in a location at a certain point of time. Genetic programming, Gene Expression Programming and Neuro – Evolution focus on formulating computer programs to solve the problems and the latter employs Artificial Neural Networks (ANN).

Artificial Neural Networks [13] are stimulated by the structure and function of neural cells in the human brain. Such networks are built by arranging artificial neurons that comply with desired input and output patterns. Feed-forward and recurrent networks are two types of ANNs that either adopt Supervised or Unsupervised learning strategies. While Perceptron and Back Propagation are suitable for solving classification and regression problems, Learning Vector Quantization can handle only classification problems. Hop-field Networks solve matching problems and Self-organizing maps focus on feature extraction and visualization. While Immune Algorithms broadly classified as Clonal Selection, Negative Selection, Immune Network and Dendritic Cell can be used to solve bio-informatics problems, Stochastic problems are applicable to deterministic optimization problems.

Communal behaviour of organisms is the basis of Swarm Intelligence. There is vast literature on swarm intelligence algorithms like Ant Colony Optimization (ACO) [16], Artificial Bee Colony (ABC) [15], Cuckoo Search (CS)[17], Particle Swarm Optimization (PSO) [14], Bacterial Foraging Optimization (BFO) [18], Bat Algorithm (BA) [19], Gravitational Search Algorithm (GSA) [20], River Formation Dynamics (RFD) [21], Glow-worm Swarm Optimization (GSO) [22], Lion Optimization (LO) [23], Social Spider Optimization (SSO) [24], Flower Pollination (FP) [25], etc., Many physics-related algorithms based on Newton's gravitational law, Quantum mechanics, Universe theory, electromagnetism, glass demagnetisation and electrostatics have been formulated to address optimization problems. Mathematically, optimization involves maximizing or minimizing an objective function where the input values of the function satisfy specific constraints. The taxonomy of optimization algorithms can be mentioned as below:

1. Continuous and Discrete Optimization: If optimization models employ variables with values from a subset of integers, those models are termed 'Discrete Optimization models'. If there are no restrictions on the values of the variables in the model, then such models are 'Continuous Optimization models'. This explains why solving Continuous Optimization problems requires less effort comparatively.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**

D.Evangeline, Department of Information Science and Engineering, Ramaiah Institute of Technology, Bangalore, Karnataka, India.
Dr.T.Abirami, Department of Information Technology, Kongu Engineering College , Erode, Tamil Nadu, India.

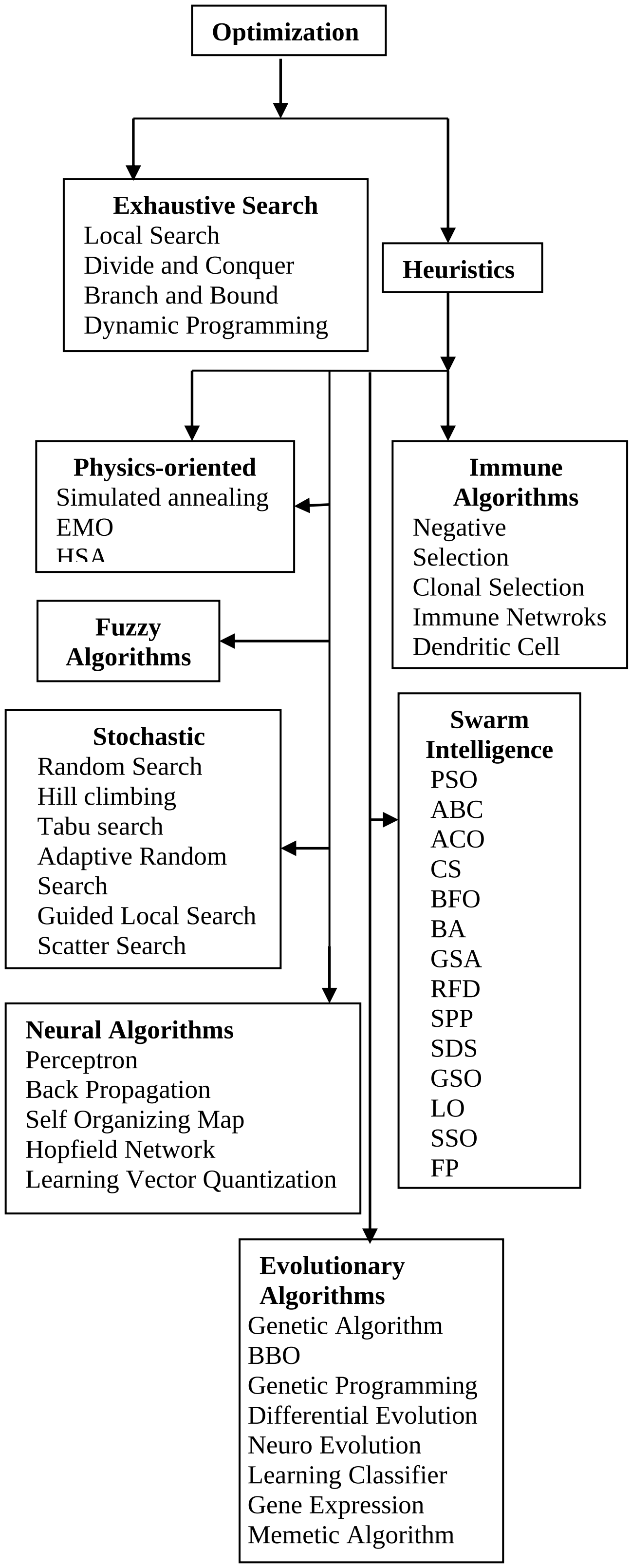

Fig. 1 Taxonomy of Optimization Algorithms

2. Unconstrained and Constrained Optimization: Optimization problems involving constraints on variables are constrained. Those without any constraints are called unconstrained problems.
3. Single or multi-objective Optimization: Those optimization problems with single objective function are called single objective optimization whereas those with multiple objective functions are multi-objective optimization.
4. Deterministic and Stochastic Optimization: If the details of the optimization function are precise, they are deterministic in nature. If the optimization function involves uncertainty, it is called stochastic optimization.

The design principles of Social Spider Optimization (SSO) algorithm are focused in this paper. Emphasis is laid on the recent applications and advancements of the algorithm in varied domains. The organisation of the paper is as follows: We will first present the fundamental characteristics of the algorithm in II. Then the applications are classified in III and the study is concluded in IV.

II. SOCIAL SPIDER OPTIMIZATION (SSO)

A. Biological Aspects

Spiders are broadly categorized into two groups based on their behavioural traits as solitary and social spiders. While solitary spiders prefer living in their own web with no or minimal contact to other spiders, social spiders live in their colonies called communal web so that they are spatially close to other neighbours. The algorithm focuses on social spiders. Both male and female spiders co-exist in the communal web with females outnumbering males by around nearly 70%. It is observed that dominant males mate with their female neighbours positioned within a particular range of distance and non-dominant males remain in a position close to other males of web and depend upon the latter for nutrition. Apart from mating, spiders interact with each other by means of vibrations and the intensity of the vibrations is dependent upon two significant factors – heaviness of the spider and the distance between the communicating spiders.

B. Features

SSO, proposed by Cuevas [26] is a meta-heuristic algorithm. It is based on the cooperative behaviour of spiders. Search agents include the male and female spiders whose population can be calculated as

$$N_f = \lfloor (0.9 - rand * 0.25) * N \rfloor \qquad (1)$$

$$N_m = N - N_f \qquad (2)$$

where N, N_f and N_m represent the total population of spiders symbolizing the count of female spiders and male spiders in the communal web respectively. The heaviness or weight of every spider i, w_i is found by the following formula:

$$w_i = \frac{fitness_i - worst}{best - worst} \qquad 0 \leq i \leq N \qquad (3)$$

where $fitness_i$, $best$ and $worst$ are the objective function, objective function's best value and worst values respectively.

In the communal web, the spiders communicate using vibrations which is computed as:

$$V_{i,j} = w_j * e^{-d_{i,j}^2} \tag{4}$$

where w_j is the heaviness of the spider transmitting vibration and $d_{i,j}$ is the Euclidean distance measured between the communicating spiders.

Such vibrations may be classified based on the transmitting spider as mentioned below:

1. Transmitting spider, s_c is heavier than the perceiving spider, i . (i.e, $w_c > w_i$) resulting in vibration $V_{c,i}$.
2. Transmitting spider, s_b is the heaviest in the communal web resulting in vibration$V_{b,i}$.

(i.e., $w_b = \max(fitness_i)$)

3. Transmitting spider, s_f is a female neighbour resulting in vibration $V_{f,i}$.

At each and every iteration of the optimization algorithm, the positions of the female and male spiders are updated using the formula as given below:

$$f_i^{k+1} = \begin{cases} f_i^k + \alpha V_{c,i}(s_c - f_i) + \beta V_{b,i}(s_b - f_i) + \gamma\left(rand - \frac{1}{2}\right) \\ \qquad if\ r_m < PF \\ f_i^k - \alpha V_{c,i}(s_c - f_i) - \beta V_{b,i}(s_b - f_i) + \gamma\left(rand - \frac{1}{2}\right) \\ \qquad if\ r_m \geq PF \end{cases} \tag{5}$$

where PF, α, β, γ and r_m are random numbers between [0,1]. PF is a threshold which is compared with these randomly generated numbers when the spiders move.

$$m_i^{k+1} = \begin{cases} m_i^k + \alpha V_{f,i}(s_f - m_i) + \gamma\left(rand - \frac{1}{2}\right) \ if\ w_{N_f+i} > w_{N_f+m} \\ m_i^k + \alpha\left(\frac{\sum_{h=1}^{N_m} m_h^k w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - m_i^k\right) if\ w_{N_f+i} \leq w_{N_f+m} \end{cases} \tag{6}$$

where $\left(\frac{\sum_{h=1}^{N_m} m_h^k w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - m_i^k\right)$ refers the weighted mean of the male spider population.

As mentioned previously, the dominant males mate with females within mating radius given by

$$r = \frac{\sum_{j=1}^{n}\left(p_j^{high} - p_j^{low}\right)}{2n} \tag{7}$$

where p_j^{high} and p_j^{low} are the upper and lower initial bounds. The operations of this SSO algorithm is shown in Fig 2.

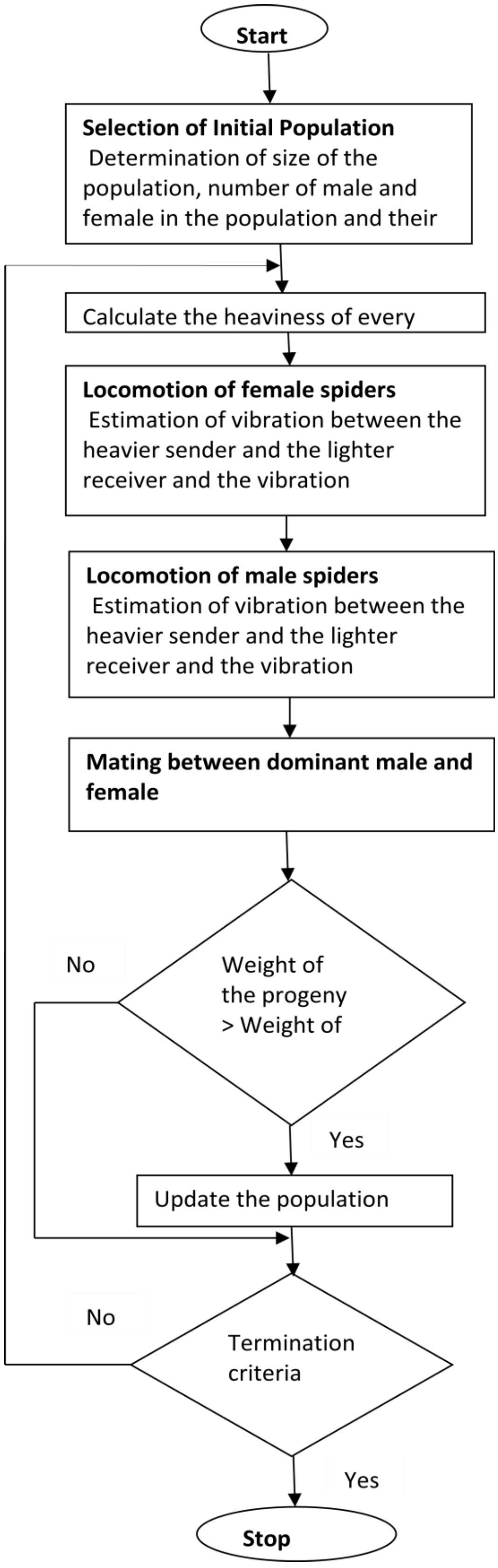
Start
Selection of Initial Population
Determination of size of the population, number of male and female in the population and their
Calculate the heaviness of every
Locomotion of female spiders
Estimation of vibration between the heavier sender and the lighter receiver and the vibration
Locomotion of male spiders
Estimation of vibration between the heavier sender and the lighter receiver and the vibration
Mating between dominant male and female
No
Weight of the progeny > Weight of
Yes
Update the population
No
Termination criteria
Yes
Stop

Fig 2. Flowchart of Social Spider Algorithm

Retrieval Number: I8261078919/19©BEIESP
DOI: 10.35940/ijitee.I8261.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The SSO algorithm can be formally presented as follows:

```
Input:
N - Total count of spider members
$N_f$ – Count of female spiders
$N_m$ - Count of male spiders
Initial positions of male and female spiders
$N_{iter}$ – Count of iterations
Output:
Optimal position of social spiders and the fitness value of objective function.
Process:
while $i<=N_{iter}$ do
Calculation of radius of mating for female and male spiders
Calculate the weight of the spiders
Compute the movement of female and male spiders based on female and male cooperative operators
Perform mating operation between dominant male and female
Update solutions if the spider progenies are heavier
End
```

C. Discussion on SSO

SSO algorithm is different from other algorithms like PSO, GA, EA etc., Unlike all swarm intelligence algorithms, SSO is self-organized because solutions are processed based on their gender. Since the female and male spiders accomplish wide examination and extensive exploitation, the algorithm avoids premature convergence and incorrect exploration [30]. All particles pursue their own best position and a common global best position in Particle swarm Optimization. On the other hand social spiders in SSO algorithm chase positions through the knowledge of others' current positions and those positions of which it is not sure to be already visited by the population before [24]. This feature of SSO improves its ability to solve optimization problems with many local optimums but makes its convergence slower. While all particles in PSO are conscious about the system information without any loss, the spiders in SSO communicating through vibrations form all-purpose information scheme with information loss [24].

III. OVERVIEW OF SSO BY APPLICATION AREA

The applications on which this heuristic algorithm was focussed includes load balancing in cloud, congestion management in power system, theft detection in power distribution systems, selection of optimal feature for prediction of dissolution profile of PLGA (poly-lactide-co-glycolide), image segmentation, solving integer programming problems, minimization of molecular potential energy function, construction of feed-forward neural networks, optimization in optical flow methods, web service selection, etc., Some of the existing literature is classified on basis of the application area and tabulated in Table 1.

A. Image Processing

Image Processing is a wide domain where segmentation is a significant operation that delves into the image to determine regions of interest (ROI) or separation of pixels under different labels. Segmentation can be based on thresholding, region splitting and merging, edge based approach, etc., Thresholding [27] is a widely-used method that determines threshold intensity value for pixel classification. If two classes are required as a result of segmentation, a single threshold value is found. This process is called bi-level thresholding. But, in multi-level thresholding, more than two classes must be found and at least two threshold values are required.

Threshold can be estimated by making the best use of between-class variance between the regions in Otsu's method or maximizing entropy to find the similarity of classes in Kapur's method. SSO algorithm is adopted for maximizing the objective function of Otsu's method as mentioned below:

$$f(t_1, t_2, t_3, \dots t_K) = \arg\max\left(\textstyle\sum_{i=0}^{K} \sigma_i\right) \qquad (8)$$

where,

$$\sigma_i = w_i(\mu_i - \mu_T)^2$$

$$\mu_i = \sum_{j=t_i}^{t_{i+1}-1} \frac{iP_j}{w_j}$$

$$w_j = \sum_{j=t_i}^{t_{i+1}-1} P_j$$

where μ_T and K are the mean intensity and number of gray levels of the image respectively.

t_i $(0 \leq i \leq K)$ gives the threshold intensity and P_j $(0 \leq j \leq K)$ represents each gray level's probability of occurrence. Likewise, the objective function of Kapur's method is maximisation of sum of entropies of heterogenous classes.

$$f(t_1, t_2, t_3, \dots t_K) = \arg\max\left(\textstyle\sum_{i=0}^{K} H_i\right) \qquad (9)$$

where,

$$H_i = -\sum_{i=t_i}^{t_{i+1}-1} \frac{P_i}{\omega_i} \ln \frac{P_i}{\omega_i}$$

$$\omega_j = \sum_{j=t_i}^{t_{i+1}-1} P_j$$

As mentioned, the parameter $fitness_i$ in Eq. (3) may be replaced with the objective functions given in Eq. (8) - (9). Similarly, maximum and minimum gray levels of the image may replace the parameters p_j^{high} and p_j^{low} in Eq. (7). This approach results to be effective in identifying thresholds better than exhaustive search. In general, when the number of levels of thresholding is increased, performance in terms of stability and accuracy deteriorate. But, SSO distinguishes itself by stability and accuracy.

B. Optical Flow

The action of objects and surfaces in a visual scene is estimated in optical flow. It is widely used in Computer Vision, Image Segmentation, cell tracking, video stabilization, motion detection, etc., In general, there are brightness patterns in an image that keep moving.

The distribution of that apparent velocity is given by a vector field in optical flow. [28] applies SSO algorithm and compares the same against Large Displacement Optical Flow (LDOF). It is inferred that SSO outperforms the conventional algorithm.

Table 1: Classification of existing literature on basis of application area

Application area	Literature
Image Processing	Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study
Optical Flow	Evolutionary Optimization Applied for Fine-Tuning Parameter Estimation in Optical Flow-Based Environments
Neural Networks	Designing evolutionary feedforward neural networks using social spider optimization algorithm
Design of Electric Circuits	Modified Social-Spider Optimization applied to Electromagnetics Social-Spider Optimization based Support Vector Machines applied for energy theft detection
Basic physical and mathematical sciences	A simplex social spider algorithm for solving integer programming and minimax problems A hybrid social spider optimization and genetic algorithm for minimizing molecular potential energy function.

C. Neural Networks

The Central Nervous System of vertebrates inspire Artificial Neural Networks (ANNs), which can be employed to estimate functions that are dependent on multiple inputs. Such networks are broadly classified as – Feed forward neural networks and feed backward neural networks. One of the biggest challenges is to train the feed forward neural networks. [29] proposes to utilize SSO algorithm to train the networks based on two variables - the weights of the connections between neurons and biases of the neurons represented in vector form. An objective function is suitably framed for the problem. While measuring the performance of this algorithm against PSO, ACO, GA, ES and PBIL in terms of Mean Squared Error (MSE), it showed high exploration, very reasonable classification accuracy and high local optima avoidance.

D. Design of Electric Circuits

A benchmark problem called Loney's solenoid design focuses on finding two parameters – position and dimension of two coils by which uniform magnetic field is generated. Since solution to the problem in terms of traditional derivative based numerical methods incurs huge computational costs, the problem is formulated as a minimization problem [32]. Another such optimization problem in electric circuit includes design of a DC motor with appropriate values for five parameters – bore stator diameter, magnetic induction found in the air gap, current density present in the conductor, magnetic induction in the teeth and back iron. This benchmark problem is also formulated as an optimization problem to maximise motor efficiency or minimize motor losses [32].

Support Vector Machines (SVMs) are widely used in pattern recognition systems but the computation costs are too high for training datasets. When SSO was employed for feature selection in datasets concerning theft detection in power distribution systems, the results obtained suggest that SSO outperforms PSO and a variant of HS [33].

E. Basic physical and mathematical sciences

In [30], a variant of SSO was employed in combination with Nelder-Mead to crack not only integer programming but also minimax problems that overcame the slow convergence of SSO. While comparing this algorithm against Branch and Bound, GA, DE, PSO and CS, it outperformed with minimal error rate. A combination of GA and SSO can be employed in minimization of molecular potential energy in order to guess the three dimensional protein structure [31].

IV. CONCLUSION

This paper has brought an overview of the applications of Social Spider Optimization algorithm that has proved to be successful in unravelling real-time decision making problems. The overview is provided on basis of the application areas which will be of great help to practitioners and beginners to understand the recent developments and applications of the algorithm. Other swarm algorithms have different parameters which when tuned can yield desired results. However, in SSO, tuning of parameter PF while determining the female cooperative operator is essential to employ the algorithm in many new applications and tap its complete potential which is crucial for the future success of this algorithm.

In future, parameters may be tuned and analysis can be done on the effect of parameter tuning. A detailed study can be carried out to determine whether parameter tuning enables practitioners to arrive at results in a quicker and easier way.

REFERENCES

1. Russell Stuart J and Norvig Peter. *Artificial Intelligence: A Modern Approach. 2nd ed.. (*Book style).Upper Saddle River, New Jersey, USA, Prentice Hall. 2003, pp. 68-70.
2. Hlavac V., Jeffery K. G., and Wiedermann J."Exhaustive Search, Combinatorial Optimization and Enumeration: Exploring the Potential of Raw Computing Power." *Lecture Notes in Computer Science* .1963. pp.18–35. Available: https://link.springer.com/chapter/10.1007/3-540-44411-4_2
3. Xin Yuan. "Heuristic Algorithms for Multiconstrained Quality-of-Service Routing". *IEEE ACM Transactions on Networking*. Vol.10, 2002.pp.244-256. Available: https://ieeexplore.ieee.org/document/993305
4. Felix-Antoine Fortin, Francois-Michel De Rainville, Marc-Andre Gardner, Marc Parizeau and Christian Gagne. "DEAP: Evolutionary Algorithms Made Easy". *Journal of Machine Learning Research*. Vol 13. 2012.pp.2171-2175. Available: http://www.jmlr.org/papers/v13/fortin12a.html
5. Yan-fei Zhu and Xiong-min Tang. "Overview of swarm intelligence". In proceedings of *IEEE 2010 International Conference on Computer Application and System Modeling*. 2010. pp.400-403. Available: https://ieeexplore.ieee.org/document/5623005

6. Hakan Bakırcıoğlu, Taşkın Koçak. “Survey of random neural network applications.” *European Journal of Operational Research*.Vol.126. 2000.pp.319-330. Available: https://www.sciencedirect.com/science/article/abs/pii/S0377221799004816
7. Chien-Wei Chu , Min-Der Lin, , Gee-Fon Liu and Yung-Hsing Sung. “Application of immune algorithms on solving minimum-cost problem of water distribution network.” *Mathematical and Computer Modelling*.Vol 48. 2008..pp. 1888-1990. Available: https://www.sciencedirect.com/science/article/pii/S0895717708001131
8. C.D. Yang. “Efficient Stochastic Algorithms on Locally Bounded Image Space.” *CVGIP-Graphical Models and Image Processing*. Vol 55.1993.pp.494-506. Available: https://www.sciencedirect.com/science/article/pii/S1049965283710370
9. Colin Paul Gloster. “Optimization Algorithms in Physics.” *International Journal of Nonlinear Sciences and Numerical Simulation*. Vol 11. 2010.pp.387-388. Available: https://www.degruyter.com/view/j/ijnsns.2010.11.6/ijnsns.2010.11.6.387/ijnsns.2010.11.6.387.xml
10. Ricardo Martinez-Soto, Oscar Castillo, Luis T. Aguilar and Patricia Melin. “Fuzzy Logic Controllers Optimization Using Genetic Algorithms and Particle Swarm Optimization.” *Advances in Soft Computing*. Vol 6438. 2010.pp.475-486. Available: https://link.springer.com/chapter/10.1007/978-3-642-16773-7_41
11. John McCall. “Genetic algorithms for modelling and optimisation.” *Journal of Computational and Applied Mathematics*. Vol 184. 2005. pp.205-222. Available: https://www.sciencedirect.com/science/article/pii/S0377042705000774
12. Dan Simon. “Biogeography-Based Optimization.” *IEEE Transactions on Evolutionary Computation*.Vol 12. 2008.pp.702-713. Available: https://ieeexplore.ieee.org/document/4475427/
13. R. E. Uhrig. “Introduction to artificial neural networks.” In proceedings of *IEEE 21st International Conference on Industrial Electronics, Control, and Instrumentation*. 1995.pp.33-37. Available: https://ieeexplore.ieee.org/document/483329
14. AlRashidi M. R and El-Hawary M. E. “A survey of particle swarm optimization applications in electric power systems.” *IEEE Transactions on Evolutionary Computation*. Vol 13. 2009.pp.913-918. Available: https://ieeexplore.ieee.org/document/4358752
15. Weifeng Gao , Sanyang Liu and Lingling Huang. “A global best artificial bee colony algorithm for global optimization.” *Journal of Computational and Applied Mathematics.* Vol 236. 2012.pp.2741-2753. Available: https://www.sciencedirect.com/science/article/pii/S0377042712000246
16. Marco Dorigo, Mauro Birattari and Thomas Stutzle. “Ant colony optimization.” *IEEE Computational Intelligence Magazine*.Vol 1. 2007.pp.28-39. Available: https://ieeexplore.ieee.org/document/4129846
17. Xin-She Yang and Suash Deb. “Cuckoo search: recent advances and applications.” *Neural Computing and Applications*. Vol 24. 2014.pp.169-174. Available: https://link.springer.com/article/10.1007/s00521-013-1367-1
18. Vivek Agrawal, Harish Sharma and Jagdish Chand Bansal. “Bacterial Foraging Optimization: A Survey.” In proceedings of *Springer 2011 International Conference on Soft Computing for Problem Solving*. 2011.pp.227-242. Available: https://link.springer.com/chapter/10.1007/978-81-322-0487-9_23
19. Xin-She Yang and Amir Hossein Gandomi. “Bat algorithm: a novel approach for global engineering optimization.” *Engineering Computations*. Vol 29. 2012.pp.464 – 483.
20. Esmat Rashedi, Hossein Nezamabadi-pour and Saeid Saryazdi. “GSA: A Gravitational Search Algorithm.” *Information Sciences*. Vol 179. 2009. pp.2232–2248. Available: https://www.sciencedirect.com/science/article/pii/S0020025509001200
21. Pablo Rabanal, Ismael Rodríguez and Fernando Rubio. “Solving Dynamic TSP by Using River Formation Dynamics.” In proceedings of *IEEE Fourth International Conference on Natural Computation*. 2008. pp.246-250. Available: https://ieeexplore.ieee.org/document/4666848
22. Mingyu Du, Xiujuan Lei and Zhenqiang Wu. “A simplified glowworm swarm optimization algorithm.” In proceedings of *IEEE Congress on Evolutionary Computation*. 2014. pp.2861 - 2868. Available: https://ieeexplore.ieee.org/document/6900409
23. B R Rajakumar. “Lion algorithm for standard and large scale bilinear system identification: A global optimization based on Lion's social behavior.” In proceedings of *IEEE Congress on Evolutionary Computation*. 2014. pp.2116 – 2123. Available: https://ieeexplore.ieee.org/document/6900561
24. James J.Q. Yu and Victor O.K. Li. “A social spider algorithm for global optimization.” *Applied Soft Computing*. Vol 30. 2015..pp.614-627. Available: https://www.sciencedirect.com/science/article/pii/S1568494615001052
25. Emad Nabil. “A Modified Flower Pollination Algorithm for Global Optimization.” *Expert Systems with Applications*. Vol 57. 2016..pp.192-203. Available: https://www.sciencedirect.com/science/article/pii/S0957417416301415
26. Erik Cuevas, Miguel Cienfuegos, Daniel Zaldívar and Marco Pérez-Cisneros. “A swarm optimization algorithm inspired in the behavior of the social-spider.” *Expert Systems with Applications*. Vol 40. 2013.pp.6374-6384. Available: https://www.sciencedirect.com/science/article/pii/S0957417413003394
27. Salima Ouadfel and Abdelmalik Taleb-Ahmed. “Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study.” *Expert Systems with Applications*. Vol 55. 2016. pp.566-584. Available: https://www.sciencedirect.com/science/article/pii/S0957417416300550
28. Danillo Roberto Pereira, José Delpiano and João Paulo Papa. “Evolutionary Optimization Applied for Fine-Tuning Parameter Estimation in Optical Flow-Based Environments.” In proceedings of *27th IEEE SIBGRAPI Conference on Graphics, Patterns and Images.* 2014.pp. 125-132. Available: https://ieeexplore.ieee.org/document/6915299/
29. Seyedeh Zahra Mirjalili, Shahrzad Saremi, Seyed Mohammad Mirjalili. “Designing evolutionary feedforward neural networks using social spider optimization algorithm.” *Neural Computing and Applications.* Vol 26. 2015.pp.1919-1928. Available: https://link.springer.com/article/10.1007/s00521-015-1847-6
30. Mohamed A. Tawhid, Ahmed F. Ali. “A simplex social spider algorithm for solving integer programming and minimax problems.” *Memetic Computing*. Vol 01. 2016..pp.01-20. Available: https://link.springer.com/article/10.1007/s12293-016-0180-7
31. Mohamed A. Tawhid, Ahmed F. Ali. “A hybrid social spider optimization and genetic algorithm for minimizing molecular potential energy function.” *Soft Computing* . Vol 01. 2016.pp.01-16. Available: https://link.springer.com/article/10.1007/s00500-016-2208-9
32. Carlos E Klein, Emerson H V Segundo, Viviana C Mariani, Leondro dos S. Coelho. “Modified Social-Spider Optimization applied to Electromagnetics.” *IEEE Transactions on Magnetics*. Vol 52. 2016. Available: https://ieeexplore.ieee.org/document/7279148
33. Danillo R. Pereira, MarioA. Pazotia, Luís A.M.Pereira, Douglas Rodrigues, Caio O.Ramos, André N.Souzad, João P.Papa. “Social-Spider Optimization based Support Vector Machines applied for energy theft detection.” *Computers and Electrical Engineering*. Vol 49. 2016.pp.25–38. Available: https://www.sciencedirect.com/science/article/pii/S0045790615003572

AUTHORS PROFILE

Ms. D. Evangeline is working as an Assistant Professor in the Department of Information Science and engineering at Ramaiah Institute of Technology, Bengaluru. Her research interests include Computer Graphics and Image Processing. She is the author of “Computer Graphics and Multimedia: Insights, Mathematical Models and Programming Paradigms”. She has published 12 papers in various journals and conferences. She is a member of Computer Society of India.

Dr. T.Abirami is an Associate Professor in Department of Information Technology at Kongu Engineering College, Perundurai, Erode. She has over 15 years of Teaching Experience. She is awarded doctorate from Anna University, Chennai in Feb 2014. She is life member of CSI and ISTE. She has published 12 International Journal and 25 National conference papers. She is author of “An Insight to Android” book published by Research India Publications, New Delhi. Her areas of specialization include Mobile Computing, Wireless Sensor Networks, Data mining and Mobile Application Development.