21

Salp Swarm Algorithm: Modification and Application

Essam H. Houssein
Faculty of Computers and Information
Minia University, Minia, Egypt

Ibrahim E. Mohamed
Faculty of Computers and Information
South Valley University, Luxor, Egypt

Aboul Ella Hassanien
Faculty of Computers and Information
Cairo University, Cairo, Egypt

CONTENTS

21.1 Introduction

Artificial Intelligence (AI) is the smartness displayed by machineries. It can be described as “the study and design of intelligent agents” [1], in which associate intelligent agents represent systems that identify their setting and take actions to exploit their goals. Now general methodologies of AI contain conventional statistical approaches [2], conventional symbolic AI, and Computational Intelligence (CI) [3]. It’s a set of nature-inspired computing paradigms to capture information and make sense of it for which conventional methodologies are inefficient or useless and it includes Evolutionary Computation (EC), Artificial Neural Network and fuzzy logic [4].

Swarm Intelligence (SI) is highly adapted to a group of mobile agents that directly or indirectly communicate with each other, and collectively solve a set of basic problems that cannot be solved if the agents are operating independently [5, 6]. SI is in a pioneer tributary in computer science, bio-signals [7, 8] a complex discipline that expresses a set of nature-inspired mathematical models that are inspired by the collective behavior of natural or artificial decentralized and self-organized systems cum habits of organisms like plants, animals, fish, birds, ants and other elements in our ecosystem that employ the intuitive intelligence of the entire swarm/herd to solve some complex problems for a single agent [9, 10, 11]. The characteristics of swarm-based techniques are:

- They are population-based.
- Their searches are done using multiple agents.
- The agents that frame the population are typically homogenized.
- The collective behaviors of the system arise from each individual interaction with the other and with their environment.
- The agents are always moving randomly in a haphazard way.
- The agents’ actions, principally movements are responsive to the environment.
- There is no centralized control, leaders’ performances are solely the standard for their emergence in each iteration [12].

In 2017, Mirjalili et al. [13] suggested a recent meta-heuristic, the Salp Swarm Algorithm (SSA), deeply influenced by the swarming behaviour of deep-sea salps. SSA aims to develop a new optimizer based on populations by attempting to mimic salps’ swarming conduct in the natural environment.

21.2 Salp Swarm Algorithm (SSA) in brief

SSA is an SI algorithm devised for continuous problem optimization. Compared to some existing algorithmic techniques, this algorithm seems to have a better or equivalent performance [14]. SSA is a stochastic algorithm in which, to start the optimization process, the initial population is formed by creating a set of initial random solutions, and then improving these solutions over the time in two stages, exploring and exploiting. In first stage, the promising regions are discovered by exploring the search space, while in exploitation, we hope to find better solutions than the current ones by searching the neighbourhood of specific solutions.

21.2.1 Inspiration analysis

Salps fit into the gelatinous salpidae family which are barrel-shaped and zoo plankton that form large swarms. They move slowly forwards through the sea as each zooid rhythmically contracts. This flow, hyped by muscle action, concurrently provides chemosensory details, food, exchange of respiratory gas, removal of solid and dissolved waste, sperm dispersal and propulsion by jet. To move forward, the body pumps water as propulsion [15]. SSA is the first technique to imitate the behaviour of salps in nature. The salps are marine organisms living in oceans and seas. They resemble jellyfish in their tissues and movement towards food sources [15]. Salps form groups (swarms) called salp chains; a leader and a set of followers are contained in each salp chain (please see the Figure 21.1). The leader salp attacks directly the target (feeding source), whereas all followers start moving directly or indirectly to ward the rest of the salps (and leader).

21.2.2 Mathematical model for salp chains

The swarming behaviours [16] and population of salp [17] are seldom in the literature to be mathematically modelled. Furthermore, to solve optimization problems, swarms of various animals (such as bees and ants) are commonly designed and utilized as a mathematical model, while it is rare to find mathematical pattern physical processes (like salp swarms) to solve various optimization issues. Through the next sub section, the standard model of salp chains in the review is proposed [13] to solve different problems with the optimizing process. Mathematically, the salp chains are divided into two groups by random division of the population (salps): leader and followers. The first salp in the series of salps is called the leader, whereas the remaining salps are regarded as followers. Through the given name of both types of these salps, the leader directs swarms and the remaining of these series follow each other (and the leader either explicitly or implicitly).

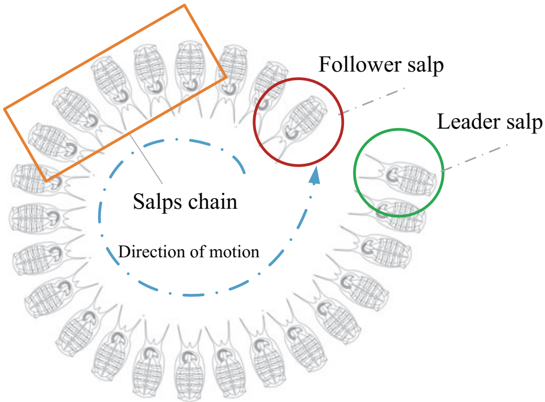

FIGURE 21.1
Demonstration of salp's series.

Given M is a counter for variables in a particular problem, like the other SI-based methodologies, the salps' location is denoted in a M-dimensional search space. Therefore, the population of salps X is composed of N swarms with M dimension. It could therefore be identified by a $N \times M$ matrix, as outlined in the equation below:

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_M^1 \\ x_1^2 & x_2^2 & \dots & x_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \dots & x_M^N \end{bmatrix} \tag{21.1}$$

A feeding source termed F is also thought to be the target of the swarm in the search space. The position of the leader is updated by the following equation:

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0 \end{cases} \tag{21.2}$$

Where X_j^1 and F_j denote the positions of leaders and feeding source in the j^{th} dimension, respectively. The ub_j and lb_j indicate the upper (superior) and lower (inferior) bounds of the j^{th} dimension. c_2 and c_3 are two random floats from the closed interval $[0, 1]$. In actuality, they guide the next location in the j^{th} dimension toward the $+\infty$ or $-\infty$ besides determining the step size. Equation 21.2 indicates that the leader only updates its location with respect to the feeding source. The coefficient c_1, the most effective parameter in SSA, gradually decreases over the course of iterations to balance exploration and exploitation, and is defined as follows:

$$c_1 = 2e^{-(\frac{4l}{L})^2} \tag{21.3}$$

where l and L represent the current iteration and maximum number of iterations, respectively. To update the position of the followers, the next equation is used (Newton's motion law):

$$X_j^i = \frac{X_j^i + X_j^{i-1}}{2} \tag{21.4}$$

where $i \geq 2$ and X_j^i is the location of the i^{th} follower at the j^{th} dimension. In SSA, the followers move toward the leader, whereas the leader moves toward the feeding source. During the process, the feeding source location can be changed and consequently the leader will move towards the new feeding source location. The flowchart of SSA is demonstrated in Fig. 21.2.

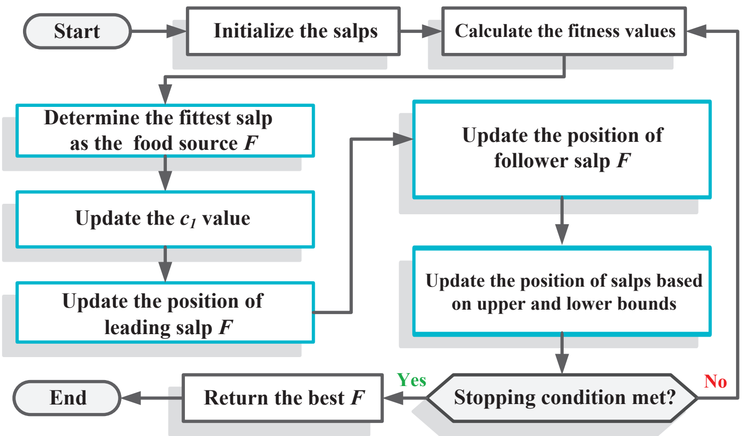

FIGURE 21.2
The flowchart of SSA.

21.3 Modifications of SSA

There are many techniques of SSA in the literature [18, 19, 20]. To categorize them, a certain classification scheme is required. This part is about enhancing SSA by using chaotic, robust, simplex, weight factor, adaptive mutation and other methods.

21.3.1 Fuzzy logic

In [21], Majhi et al. proposed an Automobile Insurance Fraud Detection System (AIFDS) that uses a hybrid fuzzy clustering technique using SSA (SSA-FCM) for outliers detection and removal. The statistical tests revealed that the use of clustering provides better accuracy. The proposed fuzzy clustering was applied for under-sampling of majority class samples of the automobile insurance data set for enhancing the effectiveness of the classifiers. The SSA helps in obtaining the optimal cluster center in the SSA-FCM. The SSA-FCM calculates the distance of the data-points from the cluster centers based on which suspicious classes are detected.

21.3.2 Robust

Suppose we have an application to an engineering optimization problem; multiple adjustment parameters are not desirable to solve a problem. In a limited calculation time, without trial and error for adjustment parameters, it is necessary to get as good a solution as possible. We therefore believe that metaheuristics should have the robustness capability to ensure the performance of searching for pre-adjusted parameters against predetermined structural variation of problems to be solved.

Tolba et al. [22] developed a novel methodology that relies on SSA to find and optimize the size of Renewable Distribution Generators (RDGs) and Shunt Capacitor Banks (SCBs) on Radial Distributed Networks (RDNs). The approach offers various objectives, functions and different constraint conditions to enhance the voltage level, properly reduce energy loss and annual operating costs.

21.3.3 Simplex

The simplex methods are strategies of stochastic variants, which maximize population diversity and improve the algorithm's local search capability. This approach helps to accomplish a better trade off between the swarm algorithm's exploration and exploitation capabilities and makes the swarm algorithm more robust and faster. The simplex method [23] has the strong ability to avoid the local optimum and enhance the ability of searching the global optimum.

21.3.4 Weight factor and adaptive mutation

A weighting factor is usually used for calculating a weighted mean, to give less (or more) importance to group members. For balancing between global exploration and local exploitation, dynamic weight factor is included in updating the formulation of the place of population.

21.3.5 Levy flight

Levy flight is a specific random walk category that distributes the step size according to the tails of the heavy power law. Sometimes, the large step helps an approach perform a global search. It would be helpful to use the Levy flight trajectory [24, 25] to gain a better trade-off between exploring and exploiting the algorithm and based on the local optimum avoidance, it has positive points.

21.3.6 Binary

Different types of optimization problems cannot be solved by meta-heuristics. A binary optimization problem has diverse decision factors that are the components of the interval $[0, 1]$. In the term, 0 or 1 represents a digital meaning of each decision variable of the binary problems, such as Features Selection (FS) problem [26], unit commitment problem [27, 28], and Knapsack Problems (KP) [29, 30].

Using a modified Arctan transformation, Rizk-Allah et al. [31] suggested a novel binary approach of the SSA called BSSA with the objective of transforming the continuous space to binary space. To enhance the exploration and exploitation capabilities, multiplicity and mobility are two advantages with regard to the modification transfer function.

21.3.7 Chaotic

SSA can approximate an optimum solution with high convergence, but SSA is not yet beneficial in searching the optimum solution which affects the algorithm performance. Therefore, to decrease this impact and to enhance its potential and effectiveness, Chaotic SSA (CSSA) was proposed by Sayed et al. [32] by merging between SSA and the chaos algorithm. Chaos, a novel numerical approach, has recently been used to improve the execution of meta-heuristic approaches. Chaos is described as simulation for self-motivated conduct of a non-linear system [33]. The population of meta-heuristic methods has the same advantages including scalable approach, simplicity, and reduced computation time. However, these approaches have two intrinsic weaknesses; low convergence rate and recession in local optima [34].

To solve the graph colouring problem, Meraihi et al. [35] proposed a new Chaotic Binary SSA (CBSSA). First, the Binary SSA (BSSA) was gained from the standard SSA where the S-Shaped transfer function is used. Second, a common chaotic map, called a logistic map was used. Using the well-known DIMACS benchmark instances, the performance of the proposed approach was stronger in comparison with various relevant colouring methods.The experimental results verified the performance and strength of the proposed CBSSA approach compared to the previously mentioned algorithms.

In [36], five chaotic maps were utilized for diagnosing and designing different feature selection methods based on SSA for data classification. Ateya et al. [37] introduced the latency and cost aware Software-Defined Networking (SDN) controller placement problem. To minimize the deployment cost and the latency, to achieve the optimum amount of controllers and also the ideal allocation of switches to controllers, a CSSA method was constructed. By introducing chaotic maps, optimizer efficiency was enhanced and local optima were prevented. The method has been evaluated for different real functionalities from the topology of the zoo. The effect of variation of different network parameters on the performance was checked. In terms of reliability and execution time, simulation outcomes proved that the introduced algorithm outperforms a GT-based approach in addition to meta-heuristic methodologies.

To get over the potential shortcomings of native SSA, Zhang et al. [38] improved an SSA-based optimizer. The designed variant was defined as a Chaos-induced and Mutation-driven SSA (CMSSA) that simultaneously combines two approaches. First, to boost the exploitation of the algorithm, the basic SSA was used to introduce a chaotic exploitative mechanism with ”shrinking” mode. Then, to get full benefit from the reliable diversification of Cauchy mutation and the strong intensification of Gaussian mutation, a combined mutation scheme was adapted. The statistical tests on a representative benchmark showed the effectiveness of the proposed method in solving optimization and engineering design problems by alleviating the precocious convergence of SSA.

21.3.8 Multi-Objective Problems (MOPS)

This part of our publication presents the basics of MOPS [39, 40]. The target of MOPS is assumed to minimize or maximize incompatible objectives functions [41, 42]. On the contrary, to improve individual target problems, including multiple objective and MOPS objectives contradict each other. A difficult task of MOPS is to find an optimal solution to optimize objectives of each function concurrently. Therefore, balancing should be done between all objectives of each function to achieve an optimum solution collection.

To find out an appropriate solution for a virtual machine locating problem, Alresheedi et al. [43] combined the SSA and sine cosine algorithm (SCA) with the means of improving MOPS techniques (MOSSASCA). The main purposes of the proposed MOSSASCA are to minimize quality of services infringements, to reduce power consumption, and maximize average time before an agent shutdown in addition to minimize conflict between the three objectives. In SCA, to increase convergence speed and to avoid getting trapped on a local optimum solution, a local search technique is followed to increase the performance of SSA. Various virtual and physical machines were in a set of experiments to evaluate the performance of the combined algorithm. Well-known MOP methods were compared with the results of MOSSASCA. Results indicate a balance between achieving the three objectives.

21.4 Application of SSA for welded beam design problem

21.4.1 Problem description

The real issue of welded beam structure gives rise to an optimization problem with four parameters. The beam is made from steel and has a shaped cross-section. The beam is focused on supporting a force at the beam’s tip of F=6000

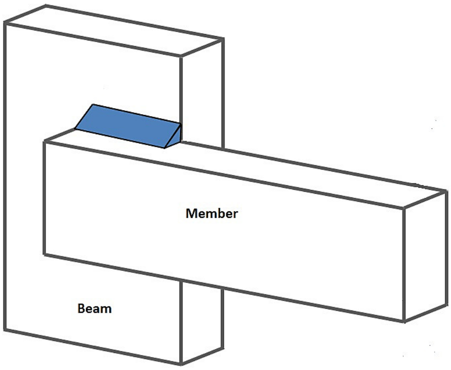

FIGURE 21.3
Welded beam design problem.

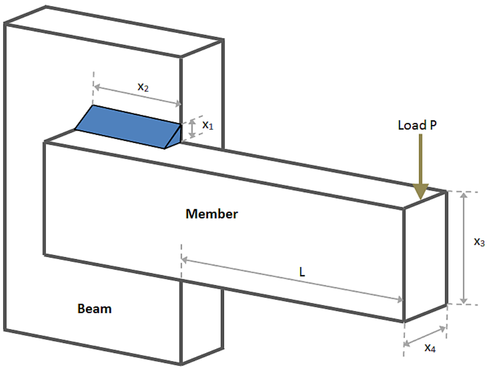

FIGURE 21.4
Parameters of the welded beam design problem.

lbs. The concern is to have four continuous factors of design: weld thickness h (x1), weld length l (x2), beam thickness t (x3) and beam width b (x4). The goal is to obtain minimum assembly costs. The τ, σ, δ and P-c care amounts are measured by treating the beam as a width "L" girder beam. The bending stress, buckling load, and weld stress are defined with notations as σ(x), $P_c(x)$, and $\tau(x)$. Both Fig. 21.3 and Fig. 21.4 illustrate the problem shape and its parameters.

21.4.2 How can SSA be used to optimize this problem?

To optimize the problem parameters of the welded beam design, we proposed a SSA-based method to achieve an optimal solution to the optimization problem. The details of our method are described follows:

Step 1, initialize the main problem factors like P = 6000 lb, L = 14 in, $\delta_{\max} = 0.25$ in, E = 30 $\times 10^6$ psi, G = 12 $\times$ 10^6 psi, $\tau_{\max} = 13600$ psi, $\sigma_{\max} = 30000$ psi.

Step 2, initialize the salps population randomly and the termination conditions shall be set. Step 3, calculate the fitness values on the basis of the objective function and the constraints (G1, G2, G3, G4, G5, G6, G7) are shown as follows:

$$\begin{aligned}
&\textbf{Cost Function:}\\
&\min_x f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4\,(14 + x_2)\\
&\textbf{Constraints}\\
&G_1(x) = \tau_{\max} - \tau(x) \geq 0\\
&G_2(x) = \sigma_{\max} - \sigma(x) \geq 0\\
&G_3(x) = x_4 - x_1 \geq 0\\
&G_4(x) = 5 - 1.10471x_1^2x_2 + 0.04811x_3x_4\,(14.0 + x_2) \geq 0\\
&G_5(x) = x_1 - 0.125 \geq 0\\
&G_6(x) = P_c(x) - P \geq 0\\
&G_7(x) = \delta\max = \delta(x) \geq 0\\
&\textbf{Variables Boundaries:}\\
&0.125 \leq x_1 \leq 5,\\
&0.1 \leq x_2 \leq 10,\\
&0.1 \leq x_3 \leq 10\\
&0.125 \leq x_4 \leq 5
\end{aligned} \tag{21.5}$$

The weld stress $\tau(x)$ has two components which are τ' (1st derivative of shear stress) and τ''(2nd derivative of shear stress). $\tau(x)$ is computed using the following equation:

$$\tau(x) = \sqrt{(\tau')^2 + \tau'' \cdot \tau' \frac{x_2}{2R} + (\tau'')^2}$$

where

$\tau' = P/(\sqrt(2) * x_1 * x_2)$,

$\tau'' = (M * R)/J$,

$R = \sqrt{x_2^2/4 + ((x_1 + x_3)/2)^2}$,

$J = 2 * (\sqrt{2} * x_1 * x_2 * ((x_2^2)/12 + ((x_1 + x_3)/2)^2))$,

$M = P * (L + x_2/2)$.

The bar bending stress $\sigma(x)$ is calculated from the following equation:

$$\sigma(x) = \frac{6PL}{x_4x_3^2}.$$

The bar buckling load is found from the following equation:

$$Pc(x) = \frac{4.013E\sqrt{x_3^2 x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right).$$

The bar displacement is computed using the following equation:

$$\delta(x) = \frac{4PL^3}{Ex_4X_3^3}.$$

Step 4, determine the minimum fitness value as the source food F. Step 5, update the leader's position by equation 21.2 and followers by equation 21.4. Step 6, update the new salp positions to the upper and lower limits. Step 7, the conditions of termination are checked, the method is stopped when conditions are reached or jump to step 3 if the conditions of termination are not met. Finally, the best fitness value and positions are displayed where the minimum cost is achieved.

21.4.3 Result obtained

The algorithm was tested with 10000 iterations, 60 search agents and 4 design variables. The result is compared with several techniques like Gravitational Search Algorithm (GSA), modified Particle Swarm Optimization algorithm (CPSO), Genetic algorithm and Simplex (GA) algorithm as shown in Table 21.1.

21.5 Conclusion

The SSA algorithm has been presented in detail in this paper. Some SSA algorithm modifications have been illustrated and explained. Finally, the SSA algorithm's exemplary application to a real-world issue has been shown. The SSA

TABLE 21.1
A summary of comparison results for beam problem.

Algorithm	x_1	x_2	x_3	x_4	**Fitness value**
SSA	0.2057	3.4714	9.0366	0.2057	1.72491
GSA	0.18213	3.8569790	10.0	0.2023760	1.879950
CPSO	0.20237	3.5442140	9.04821	0.2057230	1.731480
GA	0.18290	4.04830	9.36660	0.20590	1.8242
Simplex	0.27920	5.62560	7.75120	0.27960	2.530730

algorithm was used for the design of welded beams with special attributes of design variables. As shown in Section 21.4, the SSA algorithm can be used to optimize welded beam design with characteristic design variables. Using the SSA algorithm we can achieve higher automation of the welded beam design process because there is no need for expert knowledge of welded beam design. It should also be noted that the swarm-based welded beam design differs essentially from the traditional welded beam design process based on individual intervention. The application of the SSA algorithm to the welded beams design problem enables the avoidance of errors with high efficiency and obtains minimum assembly making costs.

References

1. S. Russell, P. Norvig, Peter. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
2. B.L. Agarwal. *Basic Statistics*. New Age International, 2006.
3. K.E. Voges, N. Pope. “Computational intelligence applications in business: A cross-section of the field” in *Business Applications and Computational Intelligence*, pp. 1-18, IGI Global, 2006.
4. Y. Zhang, S. Wang, G. Ji. “A comprehensive survey on particle swarm optimization algorithm and its applications”. *Mathematical Problems in Engineering*, vol. 2015, Article ID 931256, 38 pages, 2015, doi: https://doi.org/10.1155/2015/931256.
5. A. Hamad, E.H. Houssein, A.E. Hassanien, A.A. Fahmy. “Hybrid grasshopper optimization algorithm and support vector machines for automatic seizure detection in eeg signals” in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 82-91, 2018.
6. M.M. Ahmed, E.H. Houssein, A.E. Hassanien, A. Taha, E. Hassanien. “Maximizing lifetime of wireless sensor networks based on whale optimization algorithm” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 724-733, 2017.
7. E.H. Houssein, A. Hamad, A.E. Hassanien, A.A. Fahmy. “Epileptic detection based on whale optimization enhanced support vector machine”. *Journal of Information and Optimization Sciences*, vol. 40(3), pp. 699-723, 2019.
8. A. Hamad, E.H. Houssein, A.E. Hassanien, A.A. Fahmy. “A hybrid eeg signals classification approach based on grey wolf optimizer enhanced svms for epileptic detection” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 108-117, 2017.

9. V. Pandiri, A. Singh. “Swarm intelligence approaches for multidepot salesmen problems with load balancing”. *Applied Intelligence*, vol. 44(4), pp. 849-861, 2016.
10. A.A. Ewees, M.A. Elaziz, E.H. Houssein. “Improved grasshopper optimization algorithm using opposition-based learning”. *Expert Systems with Applications*, vol. 112, pp. 156-172, 2018.
11. A.G. Hussien, E.H. Houssein, A.E. Hassanien. “A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection” in *Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 166-172, 2017.
12. R.S. Parpinelli, H.S. Lopes. “New inspirations in swarm intelligence: a survey” in *International Journal of Bio-Inspired Computation*, vol. 3(1), pp. 1-16, 2011.
13. S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili. “Salp swarm algorithm: A bio-inspired optimizer for engineering design problems”. *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017.
14. H. Faris, S. Mirjalili, I. Aljarah, M. Mafarja, A.A. Heidari. “Salp swarm algorithm: Theory, literature review, and application in extreme learning machines” in *Nature-Inspired Optimizers*, pp. 185-199, Springer, 2020.
15. L.P. Madin. “Aspects of jet propulsion in salps”. *Canadian Journal of Zoology*, vol. 68(4), pp. 765-777, 1990.
16. V. Andersen, P. Nival. “A model of the population dynamics of salps in coastal waters of the Ligurian Sea”. *Journal of Plankton Research*, vol. 8(6), pp. 1091-1110, 1986.
17. N. Henschke, J.A. Smith, J.D. Everett, I.M. Suthers. “Population drivers of a thalia democratica swarm: insights from population modelling”. *Journal of Plankton Research*, vol. 37(5), 1074-1087, 2015.
18. A.G. Hussien, A.E. Hassanien, E.H. Houssein. “Swarming behaviour of salps algorithm for predicting chemical compound activities” in *Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 315-320, 2017.
19. H.M. Kanoosh, E.H. Houssein, M.M. Selim. “Salp swarm algorithm for node localization in wireless sensor networks”. *Journal of Computer Networks and Communications*, 2019.
20. A.G. Hussien, A.E. Hassanien, E.H. Houssein, S. Bhattacharyya, M. Amin. “S-shaped binary whale optimization algorithm for feature selection” in *Recent Trends in Signal and Image Processing*, pp. 79-87, Springer, 2019.

21. S.K. Majhi, S. Bhatachharya, R. Pradhan, S. Biswal. "Fuzzy clustering using salp swarm algorithm for automobile insurance fraud detection". *Journal of Intelligent & Fuzzy Systems*, vol. 36(3), pp. 2333-2344, 2019.
22. M. Tolba, H. Rezk, A. Diab, M. Al-Dhaifallah. "A novel robust methodology based salp swarm algorithm for allocation and capacity of renewable distributed generators on distribution grids". *Energies*, vol. 11(10), pp. 25-56, 2018.
23. X.-S. Yang. *Engineering Optimization: An Introduction with Metaheuristic Applications*. John Wiley & Sons, 2010.
24. X.-S. Yang, S. Deb. "Cuckoo search via Lévy flights" in *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210-214, 2009.
25. A.F. Kamaruzaman, A.M. Zain, S.M. Yusuf, A. Udin. "Levy flight algorithm for optimization problems-a literature review". *Applied Mechanics and Materials*, vol. 421, pp. 496-501, 2013.
26. H. Faris, M.M. Mafarja, A.A. Heidari, I. Aljarah, A. Ala'M, S. Mirjalili, H. Fujita. "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems". *Knowledge-Based Systems*, vol. 154, pp. 43-67, 2018.
27. L.K. Panwar, S. Reddy, A. Verma, B.K. Panigrahi, R. Kumar. "Binary grey wolf optimizer for large scale unit commitment problem". *Swarm and Evolutionary Computation*, vol. 38, pp. 251-266, 2018.
28. Y.-K. Wu, H.-Y. Chang, S.M. Chang. "Analysis and comparison for the unit commitment problem in a large-scale power system by using three meta-heuristic algorithms". *Energy Procedia*, vol. 141, pp. 423-427, 2017.
29. Y. He, X. Wang. "Group theory-based optimization algorithm for solving knapsack problems". *Knowledge-Based Systems*, 2018.
30. E. Ulker, V. Tongur. "Migrating birds optimization (mbo) algorithm to solve knapsack problem". *Procedia Computer Science*, vol. 111, pp. 71-76, 2017.
31. R.M. Rizk-Allah, A.E. Hassanien, M. Elhoseny, M. Gunasekaran. "A new binary salp swarm algorithm: development and application for optimization tasks". *Neural Computing and Applications*, pp. 1-23, 2018.
32. G.I. Sayed, G. Khoriba, M.H. Haggag. "A novel chaotic salp swarm algorithm for global optimization and feature selection". *Applied Intelligence*, vol. 48(10), pp. 3462-3481, 2018.

33. L. dos Santos Coelho, V.C. Mariani. "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization". *Expert Systems with Applications*, vol. 34(3), pp. 1905-1913, 2008.

34. K.-L. Du, M.N.S. Swamy. "Particle swarm optimization" in *Search and Optimization by Metaheuristics*, pp. 153-173, Springer, 2016.

35. Y. Meraihi, A. Ramdane-Cherif, M. Mahseur, D. Achelia. "A chaotic binary salp swarm algorithm for solving the graph coloring problem" in *International Symposium on Modelling and Implementation of Complex Systems*, pp. 106-118, Springer, 2018.

36. A.E. Hegazy, M.A. Makhlouf, G.S. El-Tawel. "Feature selection using chaotic salp swarm algorithm for data classification". *Arabian Journal for Science and Engineering*, vol. 44(4), pp. 3801-3816, 2019.

37. A.A. Ateya, A. Muthanna, A. Vybornova, A.D. Algarni, A. Abuarqoub, Y. Koucheryavy, A. Koucheryavy. "Chaotic salp swarm algorithm for sdn multi-controller networks". *Engineering Science and Technology*, pp. 31243-31261, 2019.

38. Q. Zhang, H. Chen, A.A. Heidari, X. Zhao, Y. Xu, P. Wang, Y. Li, C. Li. "Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers". *IEEE Access*, vol. 7, pp. 31243-31261, 2019.

39. S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, I. Aljarah. "Grasshopper optimization algorithm for multi-objective optimization problems". *Applied Intelligence*, vol. 48(4), pp. 805-820, 2018.

40. A. Tharwat, E.H. Houssein, M.M. Ahmed, A.E. Hassanien, T. Gabel. "Mogoa algorithm for constrained and unconstrained multi-objective optimization problems". *Applied Intelligence*, pp. 48(8), pp. 2268-2283, 2018.

41. A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang. "Multiobjective evolutionary algorithms: A survey of the state of the art". *Swarm and Evolutionary Computation*, vol. 1(1), pp. 32-49, 2011.

42. B.Y. Qu, Y.S. Zhu, Y.C. Jiao, M.Y.Wu, P.N. Suganthan, J.J. Liang. "A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems". *Swarm and Evolutionary Computation*, vol. 38, pp. 1-11, 2018.

43. S.S. Alresheedi, S. Lu, M.A. Elaziz, A.A. Ewees. "Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing". *Human-centric Computing and Information Sciences*, vol. 9(1), pp. 15-38, 2019.