Modified Salp Swarm Algorithm for Global Optimisation With the Use of Chaos

M. Nacer[1], **T. Hamaizia**[2]

[1] Brothers Mentouri Constantine 1, Constantine, Algeria

[2]Brothers Mentouri Constantine 1, Constantine 1, Algeria

Souhilanacer24@gmail.com, h2tayeb@gmail.com

A recently developed meta-heuristic optimization algorithm, Salp Swarm Algorithm (SSA), has manifested its capability in solving various optimization problems and many real-life applications (see, [1]). SSA is based on salps swarming behavior when finding their way and searching for food in the oceans. Nonetheless, like most meta-heuristic algorithms, SSA experiences low convergence and stagnation in local optima and rate. There is a need to enhance SSA to speed its convergence and effectiveness to solve complex problems. In the present study, we will introduce chaos into SSA (CSSA) to increase its global search mobility for robust global optimization. Detailed studies are carried out on real-world nonlinear benchmark functions with chaotic maps (see, [2]). The comparative results show that insert chaos significantly enhances the performance of the SSA algorithm.

Chaotic Salp Swarm Algorithm (CSSA)

1. **Population Initialization Based on Chaotic Mapping.** The core of the swarm intelligence algorithm is the continuous iteration of the population, so the initialization of the population has a direct impact on the final solution and also affects the optimization ability. The more abundant and diverse the initialized population is, the more favorable it will be to find the global optimal solution for the population. Without the help of prior knowledge, most swarm intelligence algorithms are random population initialization, which greatly affects their performance. The chaotic sequence has the characteristics of ergodicity and randomness, and the population initialization by a chaotic sequence can have better diversity. It is usually difficult to evaluate the qualities of the generated sequences, to achieve the task, various SSA algorithms with different chaotic maps were proposed to compare the performance of each initial distribution. In this work, 3 most widely used chaotic maps were used to verify the importance of initial distribution for solving the optimal solutions, which are the Logistic map, Piecewise map, and Tent map. After inserting the chaotic map, the initialization formula of the population becomes:

$$x_j^i = t_j^i * (ub_j - lb_j) + lb_j. \tag{1}$$

   Where x_j^i denote the position of i-th salp in the j-th dimension, The ub_j and lb_j indicates the upper (superior) and lower (inferior) bounds of jth dimension, t_j^i is the value of the chaotic maps.

2. **Chaotic parameters.** The original SSA has mainly three main parameters which affect its performance. These parameters are r_1, r_2 and r_3. It noticed that r_2 and r_1 are the two main parameters influencing the updating position of a salp. In this study, the previous chaotic maps were employed to adjust the parameter r_2 of SSA. Equation (2) shows the updating of the r_2 parameter according to the tent map. Equation (3) shows the updated position of a salp according to every chaotic map.

$$r_2 = t^{l+1}. \tag{2}$$

Where l represents the iteration.

$$x_j^1 = \begin{cases} F_j + r_1((ub_j - lb_j)t^{l+1} + lb_j), & r_3 \geq 0; \\ F_j - r_1((ub_j - lb_j)t^{l+1} + lb_j), & r_3 \leq 0. \end{cases} \tag{3}$$

Where x_j^1 and F_j denote the positions of leaders and feeding sources in the jth dimension, respectively.

3. **Experiments and analysis.** The performance of the newly developed variant of CSSA was tested over a set of two benchmark problems with different characteristics. we apply CSSA for each function. Results of all algorithms are recorded for a population of 30 and a maximum iteration of 500. Linear transformation to compare different test function results with the help of min-max normalization in the range $[0, 1]$. The two functions are:

$$f_1(x) = -\cos(x_1)cos(x_2)e^{(-(x_1-\pi)^2-(x_2-\pi)^2)}, f_2(x) = -\sum_{i=1}^{n} \sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{2m}.$$

4. **Numerical results.** The numerical results are displayed in Table 1.

		SSA	CSSA1	CSSA2	CSSA3
f_1	best value	-1.000	-1	-1	-1
	worst value	-1.000	-1	-1	-1
	(x,y)	$(3.1416, 3.1416)$	$(3.1416, 3.1416)$	$(3.1416, 3.1416)$	$(3.1416, 3.1416)$
	number of iteration	500	400	334	300
	T/s	1.839424	1.657871	1.645761	1.644671
f_2	best value	-1.8013	-1.8013	-1.8013	-1.8013
	worst value	-1.8013	-1.8013	-1.8013	-1.8013
	(x,y)	$(2.2029, 1.5708)$	$(2.2029, 1.5708)$	$(2.2029, 1.5708)$	$(2.2029, 1.5708)$
	number of iteration	500	300	200	150
	T/s	1.939353	1.765432	1.723212	1.703212

Table 1: optimization results over 30 runs for f_1 and f_2

(a) CSSA1: chaotic SSA with the logistic map
(b) CSSA2: chaotic SSA with price-wise map
(c) CSSA4: chaotic SSA with tent map

The fundamental goal of this experiment is to assess the performance of CSSA with different chaotic maps and find the best chaotic map. As shown in Table 1, we notice that the performance of CSSA with different chaotic maps is superior to standard SSA. We can conclude that the Tent map has the best performance, which confirms the ability of Tent map for enhancing the performance of SSA.

1. Mirjalili S., Gandomi A. H., Mirjalili S. Z., Saremi S., Faris H., Mirjalili S. M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 2017, 1, NO. 114, 163 – 191.
2. Ateya A. A., Muthanna A., Vybornova A., Algarni A. D., Abuarqoub A., Koucheryavy A. Chaotic salp swarm algorithm for SDN multicontroller networks. Engineering Science and Technology, 2019, 22, 1001 – 1012.