An improved salp optimization algorithm inspired by quantum computing

Fanghao Tian1*, Hongkui Wei2, Xu Li2, Meibo Lv1 and Pei Wang1

1School of Astronautics, Northwestern Polytechnical University, Xi’an, Shaanxi, 710072, China
2State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, 100854, China
*Corresponding author’s e-mail: tianfanghao@mail.nwpu.edu.cn

Abstract. Salp Swarm Algorithm (SSA) is a novel optimization algorithm which is widely used in engineering problems. An improved SSA inspired by quantum computing is proposed in this paper. The principles of quantum computing, such as qubits and quantum states, are introduced into the original SSA in order to overcome the defect of trapping into local optimum easily. Instead of updating the salp position directly, the quantum angle related to the quantum state is updated to increase the diversity of states. Two multidimensional benchmark functions are used to verify the proposed improved SSA, the result shows that the introduction of quantum computing can successfully prevent the SSA from falling into the local optimum and increase the accuracy.

1. Introduction
Due to the complexity, non-linearity, constraints, and modelling difficulties of practical engineering problems, it is always essential to find the optimal solution in a complex and enormous search space. Therefore, intelligent optimization algorithms, inspired by the sociality of biological groups, have been widely applied.

The Salp Swarm Algorithm (SSA) is a novel swarm intelligent optimization algorithm, which was firstly proposed by S. Mirjalili[1] in 2017. After that, many researches related to the improvement and applications of SSA have been conducted. Ref.[2] proposed a simplified SSA where a random search radius was introduced, the algorithm efficiency was improved. In [3], the authors proposed a particle-based SSA with global exploration and local exploitation, which improved the convergence speed and accuracy. In [4], gravitational search algorithm was combined with SSA in order to enhance its search ability. In addition, SSA is widely used in many engineering fields, such as the optimization of support vector machine parameters[5], the optimal PID-LQR control[6], power system stabilizer parameter optimization[7], multilevel color image segmentation[8] and optimal reactive power planning[9]. All these applications have proven the availability and effectiveness of SSA.

In [10-13], quantum computing was used to improve the original intelligent swarm optimization algorithm. Inspired by them, the improved SSA with quantum computing is proposed in this paper. The paper is organized in 5 sections. Section 1 is the introduction while section 2 briefly introduces the original SSA. The improved SSA is introduced in section 3 and the simulation is conducted in section 4. The final section is the conclusion.
2. Salp Optimization Algorithm (SSA)
Salp is an almost transparent marine invertebrate that usually lives in cold water and feeds on plankton. The foraging behaviour of the salp group presents a ‘Leader-Follower’ chain pattern, where one leader is responsible for searching food while the other followers follow the leader in turn. The salp chain is as shown in Figure 1.

![Figure 1. The chain of the salp swarm](image)

In order to model the salp foraging chain, all individuals in salp swarm are divided into leaders and followers with two different updating functions. The salp in front of the chain is the leader which update its position based on the optimal food position while the others are the followers which update their positions according to the adjacent individual in front of them. The specific procedures of SSA contains three parts: Initialization, Determination of the leader and Updating.

2.1 Initialization
Suppose the minimization problem with the objective function is as shown in (1):

$$F = f(x_1, x_2, x_3, \ldots, x_N)$$  

where \(N\) is the dimension of the objective function.

Suppose each variable has constant boundaries as shown in (2):

$$l_{b_i} \leq x_i \leq u_{b_i} \quad (i = 1, 2, 3, \ldots, N)$$

where \(l_{b_i}\) refers to the lower boundary of the \(i\)-th variable and \(u_{b_i}\) indicates the upper boundary.

Suppose the population of the salp swarm is \(D\), which is also called the number of searching agents. The initial position of each searching agent is obtained by (3):

$$X_{D \times N} = rand(D, N) \ast (U_b - L_b) + L_b$$

where \(rand(D, N)\) is a \(D \times N\) dimension matrix, of which each element is a random number between 0 and 1.

2.2 Determination of the Leader
After initializing the salp position, the position of each individual is substituted into the objective function (1) and \(D\) fitness values are correspondingly obtained. Then the \(D\) fitness values are sorted, the smallest fitness value is found and the corresponding individual is regarded as the leader, and the remaining individuals are ranked from small to large as followers. It is worth noting that the leader is the individual whose position is nearest to the food position (optimal position).

Accordingly, the rows of matrix \(X\) is rearranged based on the sorted fitness values. In this case, among all \(D\) individuals, the leader is represented by the \(N\)-dimensional row vector \(X^1\). The \(i\)-th component of the leader is represented by \(X^1_i\), where \(i = 1, 2, 3, \ldots, N\). Similarly, the \(i\)-th dimension of the \(j\)-th follower is represented by \(X^j_i\), where \(j = 2, 3, \ldots, D\).

2.3 Updating
The positions of the leader and followers are updated in turn. The leader position is updated as (4):
where $X_i$ is the updated i-th variable of the leader position, $F_i$ represents the i-th variable of the individual with the best fitness value in the previous frame, $ub, lb$ are the upper and lower bounds of the corresponding dimensional variables, respectively, $c_1, c_2, c_3$ are three control parameters and $c_2, c_3$ are both random number between 0 and 1.

It is worth noting that the position updating of the leader is only related to the individual position with the optimal fitness value and has nothing to do with the historical position of other individuals or itself. $c_3$ determines the updating direction, which can be added or subtracted from the original basis. $c_3$ is the key parameter that determines the updating step. It is defined as:

$$c_3 = 2e^{-\gamma(l/L)^2}$$

where $l, L$ represent the current and maximum iterations, respectively, $e$ is the natural constant.

It is clear that the smaller the number of iterations, the larger the value $c_3$. In the first few iterations, the large update step can ensure the leader approaching the global optimal area faster. When the number of iterations approaches the maximum number of iterations, the value gradually approaches 0.

Next, the follower locations are updated. In the process of foraging, the followers are connected end to end, showing a chain-like combination shape. This chain structure makes the followers’ strong motions strongly influenced by individuals before and after. The position update depends on itself and the individual position in front of it. The update formula is as follows:

$$X_j^i = \frac{1}{2}(X_j^i + X_j^{i+1})$$

where $X_j^i$ Represents the i-th variable of the j-th individual and $X_j^{i+1}$ represents the j-th position of the adjacent individual in front of the j-th individual.

After the leader and follower positions are updated separately, the fitness values of all individuals are calculated in turn, and then the leader is updated again, and the corresponding followers are updated in order to make the followers approach the leader in turn. Iterations are repeated several times to complete the optimization.

3. Improved Salp Optimization Algorithm inspired by quantum computing

The original SSA has the advantages of simple updating functions, yet has the disadvantages of easily falling into the local optimum, especially facing the multidimensional functions. Inspired by the idea of applying quantum computing methods into intelligent optimization algorithms to improve their global optimization performance, the Quantum-inspired Salp Swarm Algorithm (QSSA) is proposed. In this section, the principle of quantum computing and the specific QSSA procedures are introduced.

3.1 The principle of quantum computing

In classical computing, information is represented by binary numbers of 0 and 1, usually called bits. In quantum computing, $|0\rangle$ and $|1\rangle$ are used to represent the two basic states of microscopic particles, which are called quantum bits (qubits). $|\rangle$ is called Dirac symbol, representing states in quantum mechanics. A qubit can be stored continuously and randomly on any superposition state of two basic states, as is shown in (7).

$$|\phi\rangle = \alpha |0\rangle + \beta |1\rangle$$

where $\alpha$ and $\beta$ represent the probability amplitude of the quantum state, that is, the quantum state $|\phi\rangle$ is collapsed to $|0\rangle$ and $|1\rangle$ with the probability of $\alpha^2$ and $\beta^2$ respectively.

$$\alpha^2 + \beta^2 = 1$$
Since each single qubit has 2 possible states, a qubit string of length $n$ can represent $2^n$ different states. As a result, the $n$-dimensional salp individuals can be expressed by the qubit string of length $n$ with the probability amplitudes:

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_n \end{bmatrix}$$ (9)

Based on (8), the quantum angle can be defined as (10):

$$\theta_i = \arctan \frac{\beta_i}{\alpha_i} = \arctan \frac{\sin \theta_i}{\cos \theta_i}$$ (10)

Thus, the state of a salp individual is depicted by qubit and quantum angle. Equation (9) can be rewritten into

$$q = \begin{bmatrix} \sin \theta_1 & \sin \theta_2 & \sin \theta_3 & \cdots & \sin \theta_n \\ \cos \theta_1 & \cos \theta_2 & \cos \theta_3 & \cdots & \cos \theta_n \end{bmatrix}$$ (11)

3.2 Quantum-inspired SSA (QSSA)

With the introduction of basic quantum computing principle, the initialization and updating process of original SSA are changed.

3.2.1 Initialization

Instead of initializing the positions of salp swarm, the quantum angles are initialized in QSSA, by generalizing a $n$-dimensional random row from 0 to $2\pi$ for each individual. Then, an initialized qubits string is generalized as (11). It is worth noting that one quantum angle chain produces two rows of qubits chains, which will generalize two equivalent position chains. The scales of each component in $q$ are all from -1 to 1, so the scale transfer function is needed. Take $\sin \theta_i$ as an example, the scale transfer function is as shown in (12).

$$p_{1i} = 0.5ub_1 (1 + \sin \theta_i) - 0.5lb_1 (1 - \sin \theta_i)$$ (12)

where $p_{1i}$ is the first component of the first equivalent position chain, $ub_1$ and $lb_1$ are the upper and lower boundaries.

The population of salp swarm is $D$. Since each individual has two $n$-dimensional position chains, the 2 sets of positions of each individual are substituted into the objective function (1) and 2D fitness values are correspondingly obtained. Then the $D$ fitness values are sorted according to the first chain, the leader and followers are determined.

3.2.2 Updating

In QSSA, the quantum angle is updated. The update formula for the leader is:

$$\theta^i = \begin{cases} F_{\theta} + c_1 \ast 2\pi \ast c_2 & c_1 \geq 0.5 \\ F_{\theta} - c_1 \ast 2\pi \ast c_2 & c_1 < 0.5 \end{cases}$$ (13)

where $\theta^i$ is the updated $i$-th variable of the leader quantum angle, $F_{\theta}$ represents the $i$-th variable of the quantum angle with the best fitness value in the previous frame. Compared with (4), the upper and lower boundaries are $2\pi$ and 0 respectively. $c_1,c_2,c_3$ stay the same as that in (4).

The update formula for the quantum angle of the followers is as:

$$\theta^j = \frac{1}{2} (\theta^j + \theta^{j+1})$$ (14)

where $\theta^j$ represents the $i$-th variable of the $j$-th individual quantum angle and $\theta^{j+1}$ represents the $j$-th quantum angle of the adjacent individual in front of the $j$-th individual.
After quantum angles of the leader and followers are updated separately, the new qubits and the scaled equivalent positions of all individuals are calculated in turn, and then updating the quantum angles for leader and followers. Iterations are repeated several times to complete the optimization.

In order to compare the original SSA with QSSA, the pseudo codes of both algorithms are as shown in table 1:

| Salp Swarm Algorithm | Quantum-Inspired Salp Swarm Algorithm |
|----------------------|--------------------------------------|
| %Initialization as (3) | %Initialization |
| InitPos=random('unif', lb,ub,D,N); | Ang=random('unif', 0,2pi,D,N) |
| Calculate the fitness values for each InitPos; | Popu(:,:,1)=cos(InitAng);Popu(:,:,2)=sin(InitAng); |
| Sort(Fitness);%sort positions by fitness | Pos(:,:,2)=scaled(Popu); |
| For i=1:iter | Calculate fitness values for both cos and sin chain |
| If i==1 | Sort(Fitness(:,1));%sort by fitness of the cos chain |
| Update Pos(1,:) as (4); | For i=1:iter |
| Else | Update Ang(1,:) as (13); |
| Update Pos(i,:) as (5); | Else |
| End | Update Ang(i,:) as (14); |
| End | End |
| Recalculate fitness; | Recalculate Popu and Pos based on Ang; |
| Update the global best fitness and positions; | Recalculate fitness for both cos and sin chains |
| %Leader will update based on the best position | Update the global best fitness, best angle, and best chain; %Leader will update based on best angle |

4. Simulation

In order to experimentally evaluate the results of the proposed QSSA, two multidimensional benchmark functions are used. The first function is as shown in (15)

\[
F_1 = -20\exp\left(\frac{-0.2}{n}\sum_{i=1}^{n}x_i^2\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)^2 + 20 + e
\]  

where \(x_i \in [-32,32]\), the minimized optimum is 0 when \(x_i = 0\).

The second function is s shown in (16)

\[
F_2 = \frac{\pi}{n}\left[10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_i + 1)\right] + (y_n - 1)^2\right] + \sum_{i=1}^{n}u(x_i,0,100,4)
\]

where \(y_i = 1 + 0.25(x_i + 1)\), definition of \(u\) is as follows:

\[
u(x_i,a,k,m) = \begin{cases} 
k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i > a \end{cases}
\]

where \(x_i \in [-5.12,5.12]\), the minimized optimum is also 0 when \(x_i = -1\). When the dimension of these functions is 2, they can be visualized as shown in figure 2.

It is clear that both functions have many local optimum points but only one global optimum point. Before starting the simulation, the dimension is set 10, the population size is 30, and the number of iterations is 1000. SSA and the improved SSA proposed in this paper are used to optimize the 2 functions
Figure 2. Visualized image of function 1(left) and function 2(right) and 50 experiments are conducted respectively. The average convergence curves of both SSA and QSSA are shown in figure 3.

Figure 3. Average Convergence Curves for function 1(left) and function 2(right)

Statistic the data of 50 sets of optimization results, and obtain the best, worst and average results as shown in table 2:

| Function | SSA      | QSSA     |
|----------|----------|----------|
| Function1| 1.8677   | 8.31E-6  |
| Average  |          |          |
| Function2| 1.5367   | 1.01E-5  |
| Best Result | 0.0045   | 1.28E-6  |
| Worst Result | 3.4046   | 9.81E-6  |
|          | 1.5367   | 1.01E-5  |
|          | 0.1237   | 3.24E-12 |
|          | 2.1608   | 2.21E-3  |

It can be seen from the figure and table that the improved SSA has better optimization results for the two functions. The QSSA can approach the optimal solution more accurate while the original SSA has a tendency of falling into the local optimum. The introduction of the principle of quantum computing enhance the global search ability of the SSA algorithm and make it more applicable in practical engineering problems.

5. Conclusion
The SSA was inspired by the ‘leader-follower’ foraging process of salp swarm. Based on the original SSA, the principles of quantum computing, such as qubits and quantum states, are introduced and used to improve the original SSA. The updating of salp positions is replaced by updating the quantum state through updating the quantum angle. Two multidimensional benchmark functions are used to verify the proposed improved SSA, the result shows that the proposed improved SSA has better performance on the result accuracy and successfully prevents the SSA from falling into the local optimum.
Acknowledgment:
The work in this paper was supported by the State Key Laboratory of Intelligent Manufacturing System Technology of China.

References

[1] Mirjalili, S., Gandomi, A., Mirjalili, S.Z., et al. (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software., 7: 1-29.
[2] Xiao, B., Wang, R., Xu, Y., et al. (2019) Simplified Salp Swarm Algorithm. In: IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). Dalian. pp. 226-230.
[3] Xiao, B., Wang, R., Xu, Y., et al. (2019) Salp Swarm Algorithm based on Particle-best. In: IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chengdu. pp. 1383-1387.
[4] Li, S., Yu, Y., Sugiyama, D., et al. (2018) A Hybrid Salp Swarm Algorithm With Gravitational Search Mechanism. In: 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). Nanjing. pp. 257-261.
[5] Rajalaxmi, R.R., Vidyathara, E. (2019) A Mutated Salp Swarm Algorithm for Optimization of Support Vector Machine Parameters. In: 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore. pp. 979-983.
[6] Baygi, S.M.H., Karsaz, A. (2018) A hybrid optimal PID-LQR control of structural system: A case study of salp swarm optimization. In 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC). Bam. pp. 1-6.
[7] Ekinci, S., Hepkimoglu, B. (2018) Parameter optimization of power system stabilizer via Salp Swarm algorithm. In: 5th International Conference on Electrical and Electronic Engineering (ICEEE). Istanbul. pp. 143-147.
[8] Xing, Z., Jia, H. (2019) Multilevel Color Image Segmentation Based on GLCM and Improved Salp Swarm Algorithm. IEEE Access., 7: 37672-37690.
[9] Mahdad, B., Kamel, S., New strategy based modified Salp swarm algorithm for optimal reactive power planning: a case study of the Algerian electrical system (114 bus). IET Generation, Transmission & Distribution., 13:4523-4540.
[10] Chen, Y., Tsai, C., Chiang, M., et al. (2018) An Improved Quantum-Inspired Evolutionary Algorithm for Data Clustering. In IEEE International Conference on Systems, Man, and Cybernetics (SMC). Miyazaki. pp. 3411-3416.
[11] Abraham, J.B., Wahyunggoro, O., Setiawan, N.A. (2019) An Effective Quantum Inspired Genetic Algorithm for Continuous Multiobjective Optimization. In 5th International Conference on Science in Information Technology (ICSITech). Yogyakarta. pp. 161-166.
[12] Ding, Y., Li J. (2017) The application of Quantum-inspired ant colony algorithm in automatic segmentation of tomato image. In 2nd International Conference on Image, Vision and Computing (ICIVC). Chengdu. pp. 341-345.