An improved sine cosine algorithm for assignment problem

Dan Ye, Yu Liu, Shan Zhang*, Fei Yu, Hongyu Meng, Bo Li and Erzhen Shang
Grid Anshan Electric Power Supply Company, Anshan, Liaoning, 114000, China
*Corresponding author’s e-mail: asgdgs_zhangshan@163.com

Abstract. The assignment problem is a NP-hard combinatorial optimization problem, where assignees are being assigned to perform tasks. This paper presents an improved sine and cosine algorithm (ISCA) to solve this problem. Nonlinear decreasing inertia weight, chaotic map and greedy strategy are added to the original sine cosine algorithm (SCA) to enhance the ability of focusing on optimal and avoiding local optima. Simulation results show that the proposed algorithm can get more competitive solutions when compared with differential evolution (DE), particle swarm optimization (PSO) and original SCA.

1. Introduction
The State Grid Corporation of China (SGCC) hires many employees every year and each employee has different skills to perform different kinds of tasks. Reasonable human resource employment strategy may promote the motivation and creativity of employees and reduce human resource costs [1]. A big problem is how to assign employees to do the right work. This problem is also known as assignment problem [2].

The assignment problem is a special type of linear programming problem where assignees are being assigned to perform different tasks. A well-known algorithm designed for this problem is called the Hungarian algorithm which is developed by Hungarian mathematicians [3]. However, the Hungarian algorithm is difficult to understand and apply. It is often used to solve small scale problems. In recent years, swarm intelligent optimization algorithm has gained great concern in the theory of intelligent systems research. DE [4], PSO [5] and other algorithms have been widely used in each engineering fields.

In this paper, we develop the ISCA to solve the assignment problem. The main contribution is as follows:

- A nonlinear decreasing inertia weight is added to the SCA.
- Two chaotic random number generating techniques based on Logistic map and Chebyshev map respectively are used to generate new random numbers instead of uniform random number.
- We propose a greedy strategy to enhance the convergence performance.

The rest paper are organized as follows: the basic assignment problem model is described in Section 2. Section 3 provides the improvement to the original SCA. Section 4 describes the implementation of the Proposed ISCA. The simulation results and discussion are illustrated in Section 5. Finally, conclusions and future extensions are addressed in Section 6.

2. The basic assignment problem
The basic assignment problem should satisfy the following assumptions:
- The number of assignees and the number of tasks are the same.
Each assignee is to be assigned to exactly one task.
Each task is to be performed by exactly one assignee.
There is a cost associated with each assignee when performing each task.
The objective is to determine how all assignments should be made to minimize the total cost.

The assignment problem model uses the following decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if assignee performs } i \text{ task } j \\ 0 & \text{if not} \end{cases}$$

for $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, n$. Each $x_{ij}$ is a binary variable.

The assignment problem model is as follows:

Minimize $Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, 2, \cdots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1, 2, \cdots, n$$

where $c_{ij}$ represents the cost that assignee $i$ performing task $j$. The first constraint specifies that each assignee is to perform exactly one task. The second set requires each task to be performed by exactly one assignee.

3. The improved sine and cosine algorithm

3.1 Mathematical formulation of SCA

The SCA is a population-based optimization technique that founds the optimization process with a set of random solutions [6-8]. The SCA creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution using a mathematical model based on sine and cosine functions. Some random and adaptive variables are integrated to this algorithm to emphasize exploration and exploitation of the search space.

The following position updating equations are proposed for both phases:

$$X_{i}^{t+1} = X_{i}^{t} + r_{1} \times \sin(r_{2}) \times |r_{3} \times p_{i}^{t} - x_{i}^{t}|$$

$$X_{i}^{t+1} = X_{i}^{t} + r_{1} \times \cos(r_{2}) \times |r_{3} \times p_{i}^{t} - x_{i}^{t}|$$

where $X_{i}^{t}$ is the position of the current solution in $i$-th dimension at $t$-th iteration, $r_{1}/r_{2}/r_{3}$ are uniform random numbers in [0,1], $P_{i}^{t}$ is position of the destination point in $i$-th dimension, and $| \ |$ indicates the absolute value.

These two equations are combined to be used as follows:

$$X_{i}^{t+1} = \begin{cases} X_{i}^{t} + r_{1} \times \sin(r_{2}) \times |r_{3} \times p_{i}^{t} - x_{i}^{t}| & r_{4} < 0.5 \\ X_{i}^{t} + r_{1} \times \cos(r_{2}) \times |r_{3} \times p_{i}^{t} - x_{i}^{t}| & r_{4} \geq 0.5 \end{cases}$$

where $r_{4}$ is a uniform random number in [0,1].

The steps of the SCA is employed as follows:

**Initialize** a set of search agents (solutions) ($X$)

**Do**
**Evaluate** each of the search agents by the objective function

**Update** the best solution obtained so far ($P = X^*$)

**Update** $r_1$, $r_2$, $r_3$, and $r_4$

**Update** the position of search agents using Eq. (5)

**While** ($t <$ maximum number of iterations)

**Return** the best solution obtained so far as the global optimum

3.2 The proposed ISCA

In this section, the onlinear decreasing inertia weight, chaotic map and greedy strategy are listed.

(1) Nonlinear decreasing inertia weight

In Eq. (5), $r_1$ is replace with the following expression:

$$\alpha(t) = a \times e^{\frac{t}{maxIter}}$$  \hspace{1cm} (6)

where $maxIter$ represents the maximum number of iterations, $t$ is the current iteration, and $a$ and $b$ are two positive constants less than one with $a > b$. Fig. 1 shows the value of $\alpha$ with iterations.

![Fig. 1. Value of $\alpha(t)$ from 1.9 to 0.01, where $a=2$ and $b=0.01.$](image)

(2) Chaotic map

In Eq. (5), $r_2$ is replace with the Chebyshev map [9]:

$$\beta(t+1) = \cos \left( k \times \arccos \left( \beta(t) \right) \right), \quad \beta(t) \in (-1,1)$$  \hspace{1cm} (7)

where $\beta(1) = 0.9$ and $k = 2$.

In Eq. (5), $r_3$ is replace with the Logistic map [10]:

$$\gamma(t+1) = a \times \gamma(t) \times (1-\gamma(t)), \quad \gamma(t) \in (0,1)$$  \hspace{1cm} (8)

where $\gamma(1) = 4$.

(3) Greedy strategy

When a candidate solution $U_i^{t+1}$ is generated by Eq. (5), the following expression is used.

$$X_i^{t+1} = \begin{cases} U_i^{t+1} & f(U_i^{t+1}) < f(X_i^t) \\ X_i^t & \text{other} \end{cases}$$  \hspace{1cm} (9)
4. ISCA for the assignment problems

The implementation steps for solving the OPF problem are done as follows:

1. Initialization
   Generate \( N \) solutions by the following equation:
   \[
   X = \{X_1, X_2, \ldots, X_n\}
   \]  
   where \( X_i \) is a uniformly distributed random number between 0 and 1.

2. Calculate the objective function
   First, sort the vector components ordered by ascends for each \( X \). Then, find the column with the smallest number in cost matrix. At last, find the rower with the smallest number in that column and assign task. Here, 3 assignees and 3 tasks for instance. The cost matrix and a solution are described as follows:
   \[
   C = \begin{bmatrix}
   2 & 5 & 1 \\
   4 & 3 & 2 \\
   7 & 1 & 6 \\
   \end{bmatrix}, \quad X = (0.6, 0.8, 0.1)
   \]  
   The smallest number in \( X \) is 0.1 and it is in the third column. Select the third vector components in \( C \). Then find the rower with the smallest number in this vector and assign assignee W1 perform task T3. Next, 0.6 is the smallest number in \( X \) except 0.1 and it is in the first column. Select the first vector components in \( C \). Then find the rower with the smallest number in this vector. Because assignee W1 have assigned task T1, we find the rower with the smallest number except 2 in this vector and assign assignee W2 perform task T1. At last, assign assignee W3 perform task T2. The total cost is 6.

3. Update the best solution obtained so far (\( P = X^* \))

4. Update \( r_1, \, r_2, \, r_3 \) with Eqs. (6~8) respectively, update \( r_4 \)

5. Update the new candidate position of search agents using Eq. (5)

6. Calculate the candidate position’s objective function value and use Eq. (9)

Do the loop until the maximum number of iterations is reached.

5. Simulation results

In order to demonstrate the efficiency and robustness of ISCA for solving the assignment problems. A 10 assignees and 10 tasks case is selected, and the results are compared with DE and PSO algorithms in the literature. Simulation Experiment is implemented in MATLAB R2016a, on a personal computer with Intel i5 2.3GHz processor, 4GB of RAM and Windows 10 Professional. Due to the stochastic nature of an evolutionary algorithm in each case, 30 independent trials are conducted to calculate the best, mean, and the worst costs, and its standard deviation for each test system.

The cost data for the experiment is shown in Table 1.

|   | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|----|----|----|----|----|----|----|----|----|-----|
| W1| 3  | 3  | 10 | 9  | 5  | 2  | 11 | 11 | 11 | 5   |
| W2| 6  | 2  | 7  | 11 | 4  | 10 | 4  | 4  | 4  | 4   |
| W3| 9  | 7  | 9  | 10 | 4  | 4  | 5  | 5  | 4  | 5   |
| W4| 8  | 6  | 7  | 8  | 8  | 10 | 6  | 3  | 9   |
| W5| 7  | 2  | 8  | 6  | 10 | 9  | 6  | 6  | 11  | 10  |
| W6| 5  | 11 | 3  | 6  | 10 | 3  | 6  | 7  | 2   | 10  |
| W7| 4  | 11 | 11 | 5  | 9  | 11 | 7  | 9  | 10  | 11  |
| W8| 11 | 10 | 5  | 4  | 11 | 4  | 7  | 8  | 7   | 3   |
| W9| 11 | 5  | 5  | 3  | 2  | 5  | 7  | 10 | 7   | 3   |
| W10| 10 | 4  | 5  | 2  | 11 | 11 | 7  | 8  | 2   |
In DE, PSO, SCA and ISCA, the population size is 10 and the maximum number of iterations 100. Other parameters are set the recommended value in corresponding paper. Table 2 shows the best assignment solution and the best cost is 29.

| Table 2. The best assignment solution |
|--------------------------------------|
| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|----|----|----|----|----|----|----|----|----|-----|
| W1 | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0   |
| W2 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0   |
| W3 | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   |
| W4 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0   |
| W5 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0   |
| W6 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0   |
| W7 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| W8 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| W9 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| W10 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0   |

Table 3 shows the obtained best cost, mean cost, worst cost and standard deviation after 30 trial runs. As observed from it, all algorithms could get the best assignment solution. The PSO and SCA have the same results. ISCA produces the optimal or the same in all experiment when compared with other algorithms.

| Table 3. Comparison between ISCA and other algorithms |
|------------------------------------------------------|
| No. | Algorithm | Best | Mean | Worst | Std.dev |
|-----|-----------|------|------|-------|---------|
| 1   | DE        | 29   | 29.3333 | 30    | 0.4302 |
| 2   | PSO       | 29   | 29.0667 | 30    | 0.2537 |
| 3   | SCA       | 29   | 29.0667 | 30    | 0.2537 |
| 4   | ISCA      | 29   | 29     | 29    | 0      |

Figure 2 shows the graphs of the convergence of mean costs with iterations for DE, PSO, SCA and ISCA for a typical run. It can be seen ISCA enjoy smooth convergence.
6. Conclusion
In this paper, an improved sine and sine algorithm is proposed. It is implemented to solve the classic assignment problems. Three techniques including nonlinear decreasing inertia weight, chaotic map and greedy strategy are used in the proposed algorithm. Statistical results are compared with the reported results in literature. It is found that ISCA can yield a suitable balance between exploitation and exploration and has a better performance in terms of efficiency and robustness. The ISCA shows a good application prospect in enterprise. In the future, we will try to use the ISCA to solve other engineering optimization problems.

References
[1] Yang Z. Strategic Consideration on the Development and Employment of Human Resource in Chinas Enterprises[J]. Journal of Changsha Aeronautical Vocational and Technical College, 2005.
[2] Salman A, Ahmad I, Al-Madani S. Particle swarm optimization for task assignment problem[J]. Microprocessors and Microsystems, 2002, 26(8):363-371.
[3] Mills-Tettey A, Stent A, Dias M B. The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs[J]. carnegie mellon university, 2007.
[4] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4):341-359.
[5] Valle Y D, Venayagamoorthy G K, Mohagheghi S, et al. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(2):171-195.
[6] Mirjalili S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems[J]. Knowledge-Based Systems, 2016, 96.
[7] Alqushaibi A, Abdulkadir S J, Rais H M, and et al. Enhanced Weight-Optimized Recurrent Neural Networks Based on Sine Cosine Algorithm for Wave Height Prediction[J]. Journal of Marine Science and Engineering, 2021, 9(5):524.
[8] Wan Y, Ma A, Zhang L, et al. Multi objective Sine Cosine Algorithm for Remote Sensing Image Spatial-Spectral Clustering[J]. IEEE Transactions on Cybernetics, 2021, (99):1-15.
[9] Yoon E J, Jeon I S. An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map[J]. Communications in Nonlinear Science & Numerical Simulation, 2011, 16(6):2383-2389.
[10] Pareek N K, Patidar V, Sud K K. Image encryption using chaotic logistic map[J]. Image and Vision Computing, 2006, 24(9):926-934.