A Comparative Study of Sine Cosine Optimizer and Its Variants for Engineering Design Problems

Qusay Shihab Hamad[1,3], Hussein Samma[2], Shahrel Azmin Suandi[1](✉), and Junita Mohamad Saleh[1]

[1] School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia
qusay@student.usm.my, {shahrel, jms}@usm.my
[2] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 UTM, Johor Johor Bahru, Malaysia
hussein.samma@utm.my
[3] University of Information Technology and Communications (UOITC), Baghdad, Iraq

Abstract. Sine Cosine Algorithm (SCA) is one of the simplest optimization algorithms and is used to solve a wide range of problems due to using two simple mathematical equations. However, it faces local optima stagnation because of the constraints in its exploration and exploitation mechanism. To solve this problem, many researchers proposed new versions of sine cosine algorithm (SCA). The main concept of developing SCA performance is to add some methods or layers to original SCA, edit the SCA parameters, or only hybridize it with other optimization algorithms to improve SCA's exploration and exploitation. SCA and three new SCA variants were applied to solve three constrained engineering design problems in this study. The outcomes show that SCA was still able to report a good result more than some of its variants.

Keywords: Optimization algorithms · Metaheuristic algorithm · Swarm intelligence algorithms · Population-based optimization algorithms

1 Introduction

Sine Cosine Algorithm (SCA) [1] is a population-based approach to solve global optimization problems. SCA uses the characteristics of sine and cosine trigonometric functions in the search process. It has been applied for different applications such as feature selection problem [2], train a feedforward neural network [3], handwritten Arabic Manuscript Image Binarization [4], and a wide range of applications, etc.

As per the No Free Lunch (NFL) theorem in optimization [5], there is no optimization algorithm that can solve all world problems. This is also applied for SCA, where many works in the literature try to develop or fix this algorithm to find the best solution for different problems.

Although the literature shows that the SCA has enough ability to explore the search space but like other algorithms, it faces some difficulties like local optima stagnation,

N. M. Mahyuddin et al. (Eds.): *Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications*, LNEE 829, pp. 1083–1089, 2022.
https://doi.org/10.1007/978-981-16-8129-5_166

slow convergence, and skipping of true solutions while solving real-life problems [6]. To alienate these issues from classical SCA, some attempts have been done in the literature. For example, the hybridization of SCA with differential evolution (DE) called SCADE [7], Opposition-Based Sine Cosine Algorithm (OBSCA) [8], and a multi-strategy SCA algorithm named MSCA [9].

The main aim of this study is to evaluate the effectiveness of SCA and its variant in handling engineering design problems. These problems selected because they are difficult and commonly have several local optima [10, 11].

The rest of the paper is organized as follows: Overview of Sine Cosine Algorithm and its variants are introduced in Sect. 2. Experimental analysis of applying these algorithms to solve engineering problems is discussed in Sect. 3. In Sect. 4, we draw the conclusion of this paper.

2 SCA and Its Variants

SCA [1] starts the search process by creating a set of search agents that are positioned randomly in the search space. Then each agent updates its position based on the simple mathematical functions sine and cosine as shown in Eq. (1)

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 * \sin(r_2) * \left|r_3 P_i^t - X_i^t\right|, r_4 < 0.5 \\ X_i^t + r_1 * \cos(r_2) * \left|r_3 P_i^t - X_i^t\right|, r_4 \geq 0.5 \end{cases} \tag{1}$$

Where X_i^t is the position of the current agent in i-th dimension at t-th iteration, | | indicates the absolute value, P_i^t represents the position of the destination solution, in i-th dimension at t-th iteration, r_2 is a random number between [0-2π], r_3 is a random number between [0–2], r_4 is a random number between [0, 1] and used to switch between the sine and cosine equally. The author in [1] used Eq. (2) to update the value of r_1 to achieve a balance between exploration and exploitation.

$$r_1 = a - t\frac{a}{T} \tag{2}$$

Where a is constant, t represents current iterations, T is the total number of iterations.

The hybridization of SCA with differential evolution (DE) was proposed by Nenavath et al. [7]. Their hybrid scheme was applied for solving object tracking problem. Nevertheless, integrating SCA with DE will increase algorithm complexity and fitness evaluations required for each population, i.e., SCA and DE. An improved version of SCA named Opposition-Based Sine Cosine Algorithm (OBSCA) was proposed by Abd Elaziz et al. [8]. OBSCA considers opposition-based learning as a mechanism to enhance search exploration ability. Their results indicated that a better outcome was achieved as compared with the standard SCA. A multi-strategy SCA algorithm termed MSCA was proposed by Chen et al. [9]. MSCA combines multiple control mechanisms, including the Cauchy mutation operator, chaotic local search mechanism, opposition-based learning strategy, and other differential evolution operators. These four strategies were sequentially executed to generate a new search solution. However, MSCA requires fitness evaluation computation after each strategy which increases evaluation cost.

3 Experimental Analysis

A total of three engineering design problems have been used in this analysis. The design problems are cantilever beam [10], three-bar truss [11], and multiple disk clutch brake [12] as shown in Fig. 1.

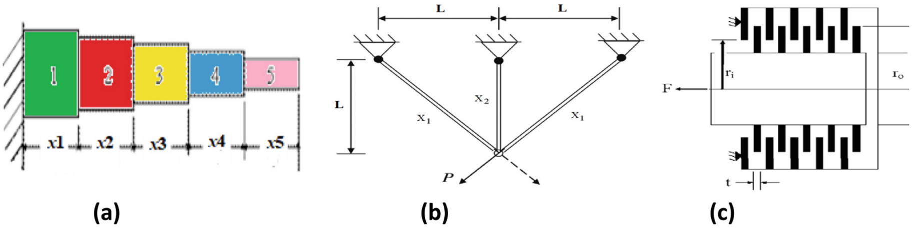


Fig. 1. Engineering design problems (a) Cantilever beam, (b) Three-bar truss, and (c) Multiple disk clutch brake

We compare the performances of SCA against recently developed SCA variants (SCADE [7], OBSCA [8], and MSCA [9]) according to the settings shown in Table 1. Each experiment was executed 30 times, and the average cost and standard deviation are shown in Table 2. As can be seen, MSCA achieved a better result in a cantilever beam with an average equal to 18.581. OBSCA achieved a better result in three-bar truss with an average equal to 268.38 while original SCA achieved a better result in multiple disk clutch brake with an average equal to 0.35. A reasonable answer for these result where SCA got the best result than all-new its variants in multiple disk clutch brake problem we can be found in NFL theorem [5], which clarify some algorithms perform well in dealing with a specific set of problems which cannot ensure to solve all optimization problems. In other words, no one can establish an omnipotent algorithm for solving all world problem.

Table 1. Algorithms parameters settings

Algorithm	Population size	Maximum no. of iteration	Other parameters
SCA	30	10^3	a = 2
SCADE	30	10^3	a = 2; F = random in [0.2, 0.8]; P_c = 0.8
OBSCA	30	10^3	a = 2
MSCA	30	10^3	a = 2; μ = 4; F = random in [0.2, 0.8]; P_c = 0.8

Table 2. Cost of engineering design

Algorithms	SCA		SCADE		OBSCA		MSCA	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Cantilever beam	32.56	9.3648	59.616	25.433	63.104	16.593	**18.581**	**2.5661**
Three-bar truss	269.1	5.5592	277.25	6.7231	**268.38**	**3.6767**	268.67	2.79672
Multiple disk clutch brake	**0.35**	**0.0349**	0.6643	0.1742	0.3889	0.0362	0.3979	0.04051

The convergence curve of SCA versus SCADE, OBSCA, and MSCA for cantilever beam, three-bar truss, and multiple disk clutch brake design problems are presented in Figs. 2, 3, and 4, respectively.

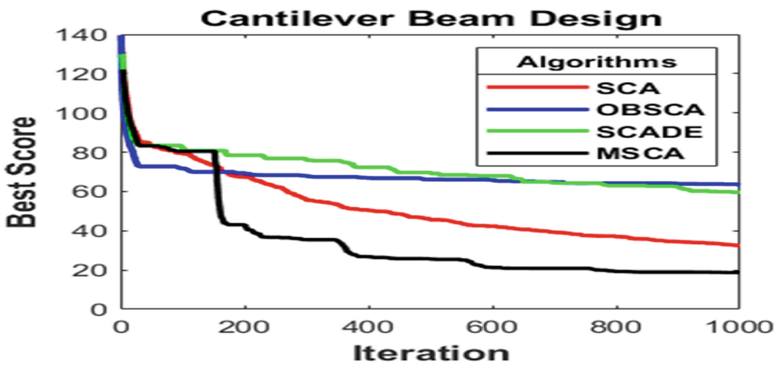


Fig. 2. The convergence curve of SCA, SCADE, OBSCA, and MSCA in Cantilever Beam.

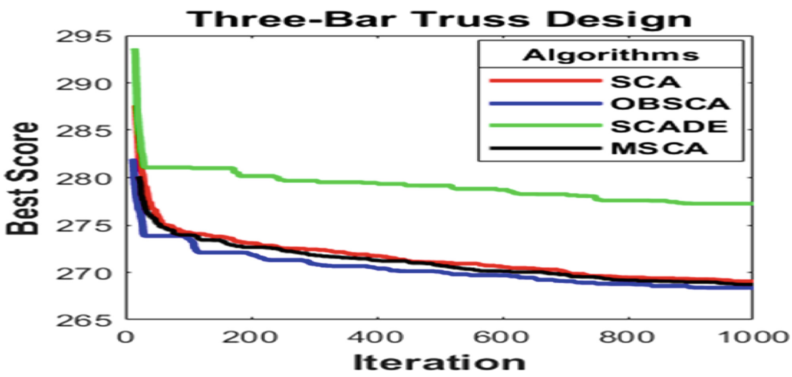


Fig. 3. The convergence curve of SCA, SCADE, OBSCA, and MSCA in Three-bar truss.

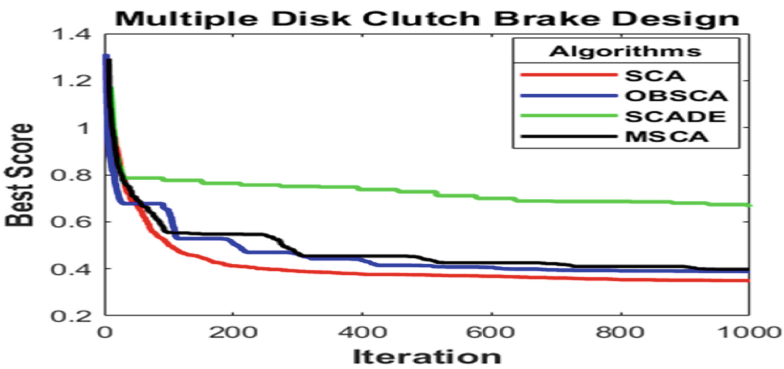
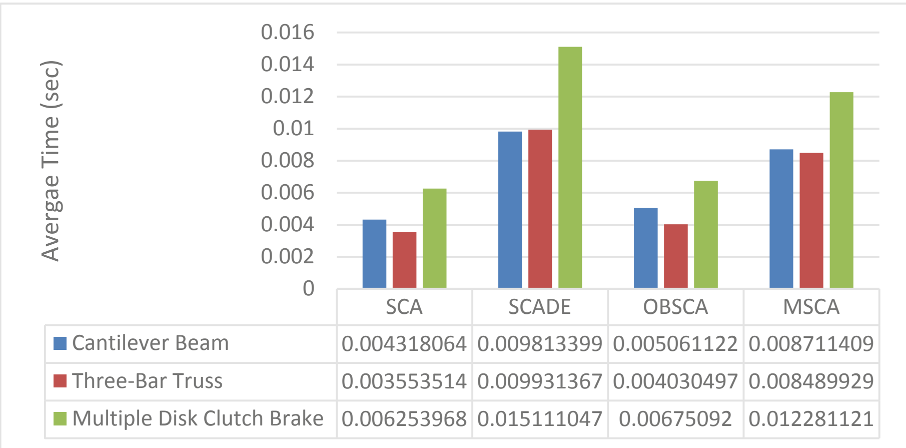


Fig. 4. The convergence curve of SCA, SCADE, OBSCA, and MSCA in Multiple disk clutch brake.

The computational time required by each algorithm is given in Fig. 5. As can be seen from Fig. 5, SCADE requires extra execution time compared to others due to its hybridization SCA with differential evolution (DE) [7] so that two algorithms work to solve the problem. On the other hand, SCA required the minimum execution time with 0.004 s in cantilever design, 0.003 s in three-bar, and 0.006 s in multiple desks, which obviously other algorithms add extra layers to the original SCA, therefore these modifications consume more time.



	SCA	SCADE	OBSCA	MSCA
Cantilever Beam	0.004318064	0.009813399	0.005061122	0.008711409
Three-Bar Truss	0.003553514	0.009931367	0.004030497	0.008489929
Multiple Disk Clutch Brake	0.006253968	0.015111047	0.00675092	0.012281121

Fig. 5. Computational time

4 Conclusion

This paper presents the Sine Cosine Algorithm (SCA) and three algorithms that were recently proposed to improve standard SCA searchability. Three constrained engineering design problems were employed to compare the effectiveness of SCA and its variants. The outcomes of the conducted analysis prove that SCA and its variants still have difficulties of local optima stagnation when dealing with complex real-world design problems. Results show that none of these algorithms was able to reach the optimal solution. From this observation, it demonstrates that improvement for SCA is desired. The improvements should focus on the balancing between exploration and exploitation to find an excellent way to escape from local optima stagnation.

Acknowledgements. This research is supported by the Malaysia Ministry of Higher Education (MOHE)Fundamental Research Grant Scheme (FRGS), no. FRGS/1/2019/ICT02/USM/03/3.

References

1. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016). https://doi.org/10.1016/j.knosys.2015.12.022
2. Hafez, A.I., Zawbaa, H.M., Emary, E., Hassanien, A.E.: Sine cosine optimization algorithm for feature selection. In: 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), August 2016, pp. 1–5. https://doi.org/10.1109/INISTA.2016.7571853
3. Sahlol, A.T., Ewees, A.A., Hemdan, A.M., Hassanien, A.E.: Training feedforward neural networks using Sine-Cosine algorithm to improve the prediction of liver enzymes on fish farmed on nano-selenite. In: 2016 12th International Computer Engineering Conference (ICENCO), December 2016, pp. 35–40. https://doi.org/10.1109/ICENCO.2016.7856442
4. Abd Elfattah, M., Abuelenin, S., Hassanien, A.E., Pan, J.S.: Handwritten Arabic manuscript image binarization using sine cosine optimization algorithm. In: Advances in Intelligent Systems and Computing, vol. 536, Springer Verlag, pp. 273–280 (2017). https://doi.org/10.1007/978-3-319-48490-7_32
5. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
6. Gupta, S., Deep, K.: Improved sine cosine algorithm with crossover scheme for global optimization. Knowl.-Based Syst. **165**, 374–406 (2019). https://doi.org/10.1016/j.knosys.2018.12.008
7. Nenavath, H., Jatoth, R.K.: Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl. Soft Comput. J. **62**, 1019–1043 (2018). https://doi.org/10.1016/j.asoc.2017.09.039
8. Abd Elaziz, M., Oliva, D., Xiong, S.: An improved opposition-based Sine Cosine algorithm for global optimization. Expert Syst. Appl. **90**, 484–500 (2017). https://doi.org/10.1016/j.eswa.2017.07.043
9. Chen, H., Wang, M., Zhao, X.: A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Appl. Math. Comput. **369**, 124872 (2020). https://doi.org/10.1016/j.amc.2019.124872

10. Thanedar, P.B., Vanderplaats, G.N.: Survey of discrete variable optimization for structural design. J. Struct. Eng. **121**(2), 301–306 (1995). https://doi.org/10.1061/(ASCE)0733-9445(1995)121:2(301)
11. Ray, T., Saini, P.J., Saini, P.: Engineering design optimization using a swarm with an intelligent information sharing among individuals, engineering optimization. En#. Opt.. 2W1, vol. 33, no. 6, pp. 735–148 (2001). https://doi.org/10.1080/03052150108940941
12. Steven, G.: Evolutionary algorithms for single and multicriteria design optimization by A. Osyczka. Struct. Multidiscip. Optim. **24**(1), 88 (2002). https://doi.org/10.1007/s00158-002-0218-y

Author Index

N. M. Mahyuddin et al. (Eds.): *Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications*, LNEE 829, pp. 1091–1096, 2022.
https://doi.org/10.1007/978-981-16-8129-5

I

J

K

L

M

Z