IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

Enhanced Competitive Swarm Optimization (ECSO) algorithm for solving optimization problems

Ghani Ali Mohammed *, Nazar K. Hussein
Department of Mathematics, College of Computer Sciences and Mathematics, Tikrit University, Tikrit, Iraq.

https://doi.org/10.54153/sjpas.2026.v8i2.1348

Article Information
Received: 22/11/2025
Revised: 03/8/2025
Accepted: 10/1/2026
Published:30/6/2026

Keywords:
Competitive, Swarm, Optimization, exploitation, and exploration.

Corresponding Author
E-mail: *ga230025pcm@st.tu.edu.iq,*
Mobile:

Abstract

This article introduces the Enhanced Competitive Swarm Optimization (ECSO) algorithm, which was developed to improve the balance of exploration and exploitation in difficult optimization situations. The proposed ECSO uses adaptive control parameters and an improved leader-follower mechanism to efficiently direct the search agents to the global optimum while avoiding premature convergence. To assess its performance, ECSO was run through a series of standard benchmark functions from the CEC2021 test suite and compared to various well-known metaheuristic algorithms. The comparative research revealed that ECSO consistently produced the lowest average fitness values with the smallest standard deviation, indicating higher precision, stability, and robustness. These findings demonstrate that ECSO strikes a good balance between global exploration and local exploitation, making it a dependable and powerful optimization strategy for solving high-dimensional and nonlinear optimization problems.

Introduction

Numerous scientific disciplines and engineering physical applications place a strong emphasis on optimization problems, which call for efficient techniques to select the best answers from a wide range of potential solutions. Methods such as gradient descent method [1], Newton's method [2], and conjugate gradient methods [3],and Quasi-Newton methods [4], and other traditional methods, which rely mostly on derivatives. Even though these approaches produce good results in many applications, their efficacy gradually declines when applied to problems with many variables. This makes the methods' implementation somewhat complex in terms of calculations, which ultimately results in a relatively long implementation time [5].

Furthermore, most applications particularly those in artificial intelligence, deep learning, and engineering have unclear or nonexistent feature or data spaces, which makes it difficult to create accurate mathematical models to find the best solution for all of this. To overcome this difficulty, researchers have tended to create meta-meteorological algorithms based on the random optimization principle, which is distinguished by its reliance on input and output analysis rather than prior knowledge of the problem's nature [6].

These algorithms' foundation is random guided search, which works well for handling challenging optimization issues that are hard to resolve with conventional techniques. conventional. Despite the fact that it does not ensure the best solution thorough and less Its features, such as the absence of mathematical derivatives and the simplicity of application, account for the computational complexity of other approaches and the high demand for its use [6].

One of the hardest problems for metaheuristic algorithms is striking a balance between exploration and exploitation. To avoid getting stuck in the solution area, the algorithm can explore new or at least uncharted territory. regional, enhancing current solutions through regional research is referred to as exploitation. promising. As a result, maintaining equilibrium between these two procedures is essential to guaranteeing the caliber of the results. on them. The algorithm may take a very long time to produce high-quality results if the exploration process is too taxing. Elevated. On the other hand, the algorithm might be restricted to a small portion of the research space, which could prevent it from finding answers. The exploitation process is professional. finest In places Other [7].

Therefore, adaptive mechanisms to adjust this balance during the research stages are necessary to design an algorithm with high efficiency and a strong and focused research capacity. Other. While some algorithms use intelligent triggers like random mutations and local escape strategies to enable the exploratory capabilities of the algorithm without lowering the efficiency of searching for solutions, other algorithms rely on time-based techniques where exploration is gradually reduced to calculate exploitation. Excellent. As a result, this problem is among the most crucial areas on which researchers concentrate in order to create and design meta-hormone algorithms that are more reliable and effective, particularly for large-scale applications. and intricate [8].

In addition to the previous division of algorithms within their dependence on the derivative or not, metaheuristic algorithms are also divided into two types, the first type is algorithms that depend on improving solutions based on one solution and the second type is algorithms that are based on improving solutions based on a community of Solutions. Type I algorithms often rely On the standard of greed [9], Specifically the criterion of profit or expected benefit from solutions, and one of the most famous algorithms within this field is (Simulated Amealing SA) [10] and algorithm (Tabu Search TS) [11]. The second type of algorithm, which depends on the community, is based on the creation of a set or a large number of possible solutions to the problem to be solved through the process of iterative creation. and optimization. As these algorithms depend on what is called the transfer of knowledge, dealing and interaction between possible solutions, which is considered a matter of cancel importance in the process of improving Solutions.

The mechanisms of metaheuristic algorithms are derived from various natural and behavioral phenomena, and they have demonstrated the capacity to both adapt to various complex problems and explore areas of promising solutions. However, the majority of these algorithms may lose many beneficial features, like increased stability and direction toward promising solutions, because they do not update their solutions based on information from prior solutions or (memory).

Based on the mentioned challenges above, the motivation behind proposing a new algorithm that takes into account several aspects that will not be fully integrated into the algorithms Previous. Despite the effectiveness and power of algorithms based on cooperation and interaction between individuals, such as PSO [12], GA [13] and algorithm [14] However, tracking historical information or learning from the past limits the ability of these algorithms to better stabilize updated solutions and avoid a return to weak solutions, and many algorithms suffer from stagnation in local solutions because there is no effective mechanism to escape from local solutions.

The proposed distance-based smart alignment strategy involves a two-phase mechanism where individuals update their location based on global or local guidelines (based on Euclidean distance). This system seeks to find a proper balance between exploration and exploitation.

In global alignment, the update is based on three types of differences: the difference with the world's best-in-class solution, the difference between two random individuals, and the mutual difference between the present individual and another individual. They are coupled with adaptive random weights to increase the diversity of search pathways.

During the swing phase, individuals learn from their nearest neighbor and the world's best solution, enhancing local exploitation and accelerating convergence to optimal solutions. To maintain diversity, Gaussian noise proportional to has been injected during the alignment phase. This reduces the likelihood of slipping into local minimum values.

Randomly variable weighting coefficients are used to achieve a flexible balance between global and local guidance components, enhancing the flexibility and efficiency of research.

Distance Matrix-Based Local Interaction: The Distance Matrix identifies persons who are closest to each aspect in the community, allowing for effective local interaction and exploitation.

The Greedy Selection Mechanism replaces current solutions with superior alternatives, resulting in a declining trend in the community's best fit value over time.

Metaheuristic algorithms were developed based on different and varied sources of inspiration from nature and some physical and mathematical phenomena, for example, for algorithms inspired by evolution, there are several algorithms within this class, the genetic algorithm (GA) (Genetic Algorithm) [13]Which simulates the process of biological evolution through the process of selection, mating and mutation in order to improve successive generations of solutions, the algorithm of differential evolution (DE) [15]Rely on the differences between individuals in the solution community, in order to generate new solutions more effective than the original, evolutionary algorithm (EAS) It is a general framework that includes an algorithm D.A.A. In other words it comprises natural evolution algorithms, an algorithm to improve biodiversity (BBO) [16]It is an algorithm that relies on biogeography in the distribution of species on islands, where new features are exchanged between solutions and organisms are moved between islands.

Also physics-inspired algorithms such as simulated annealing algorithm SA (Simulated Annealing) [10]It is an algorithm inspired by the annealing process of metals where the temperature is adopted for local acceptance of the worst solutions for practical solutions, the black hole algorithm BHA(Black Hole Algorithm) [10]It is an algorithm that simulates the behavior of black holes that attract surrounding objects where solutions are represented by surrounding objects and the best solution is by a black hole, a nuclear reaction algorithm (Nuclear Reaction Optimization) NRO [17]It is an algorithm that simulates the interactions of fusion and nuclear fission as the interaction between solutions generates a dynamic diversity in the space of solutions, the equilibrium algorithm (Equilibrium Optimization) EO [17] it is an algorithm inspired by the idea of mechanically physical systems called dynamic equilibrium where Solutions suggest Towards the (equilibrium state), which represents the best solutions within the research space, as well as the algorithm [18]GBO(Gradient Based Optimization) It is an algorithm inspired by Newton's numerical method and is considered algorithms that mimic the behavior of derivation-based algorithms.

2- Proposed Algorithm

In metaheuristic algorithms, it starts with exploration at a very large percentage, or it can even be completely and then gradually decreases with increasing frequencies, or it can be in the form of a probability based on a random value, as in this algorithm, the algorithm starts by calculating the matrix of Euclidean distances between the solutions.

$$dist - matrix(i,j) = \sum_{k=1}^{d}(X_{ik} - X_{jk})^2 \text{ .... (1)}$$

The exploration process begins with a random mutation with a probability of 0.03 and follows

$$X_{new} = X(i) + r_n \text{ .... (2)}$$

where a random value follows the natural distribution with an average of 0 and a standard deviation of $\mathrm{r_n} 0.02 * (\mathrm{Ub} - \mathrm{Ub})$

Then a random value is tested if it is greater than half the update process is as follows

$$delta\_g = g\ best - x(i) \text{ .... (3)}$$

$$delta\_b = X(r_1) - X(r_2) \text{ .... (4)}$$

$b = 1 - (\frac{iter}{maxiter})$, where $iter$ is the current iteration and $maxiter$ is the maximum iteration.

$$delta\_m = X(r_1) - X(i) \text{ .... (5)}$$

$$Noise = rand(0, 0.1) * (1 - b) * levy(dim) \text{ .... (6)}$$

$$Alignment = W_1 * delta\_g + W_2 * delta\_b + levy * delta\_m + noise \text{ .... (7)}$$

where it is a random value that follows the standard normal distribution and that W_1

$$W_2 = 1 - W_1 \text{ .... (8)}$$

Therefore, the value of the updated solution is as follows:

$$X_{new} = \mathrm{gbest} + \mathrm{alignment} \text{ .... (9)}$$

If the random value is less than half, the update process is done as follows:

We consider the solution to be the solution that we want to update through the closest solution to it, where we take the row from the distance matrix and choose the lowest value in this row while neglecting the distance between the solution and itself because it is equal to zero, as followsiidist_matrix

$$dist\_i = dist\_matrix\ (i) \text{ .... (10)}$$

$$Disti(i) = inf \text{ .... (11)}$$

$$closest\_i = min(dist\_i) \text{ .... (12)}$$

Then the account is done

$$delta\_closest = X(closest\_i) - X(i) \text{ .... (13)}$$

$$dela\_best = gbest - X(i) \text{ .... (14)}$$

$$Movement = R_1 * deltabest + R_2 * deltawine \text{ .... (15)}$$

$$X_{new} = X(i) + vc * M \text{ .... (16)}$$

where it is a random value that follows the standard normal distribution and that R_1

$$R_2 = 1 - R_1 \text{ .... (17)}$$

In flowchart of ECSO shown in Figure1 and the pseudocode of ECSO also shown in Algorithm1

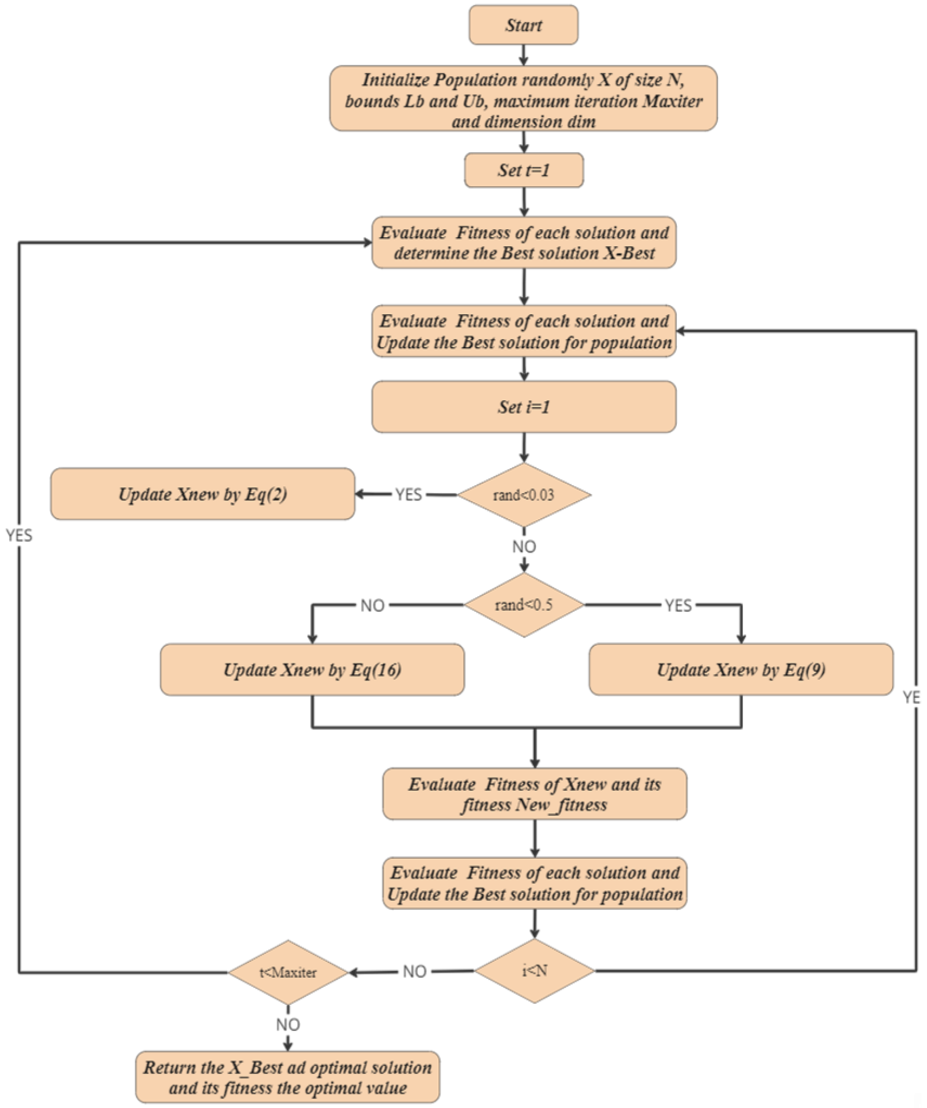


Fig. 1: *The flowchart of the RECO optimizer.*

Algorithm 1: The pseudo-code of the proposed (ECSO).

Initialize population X with size N , dim,Lb,Ub.and maximum iteration M.
Compute the fitness of each initial solution in the population
Determine best solution gbest according to its Fitness
Set $l = 0$
Main loop:
while $(l < M)$
    while (i<N)
        if rand < 0.03:
            Calculate X_{new} *by Eq.(2)*
            if rand<0.5
                Calculate by Eq.(7)
                Calculate X_{new} *by Eq.(9)*
            else
                Calculate Movement by Eq.(15)
                Calculate X_{new} *by Eq.(16)*
            end if
        end if
        Update gbest
    end while
end while

3- Experimental Results and Analysis

To evaluate the performance efficiency of the ECSO algorithm, a series of experiments were conducted using the standard IEEE CEC2021 benchmark functions. These experiments were designed to assess the algorithm's ability to achieve optimal performance across various testing environments and were all carried out under identical operating conditions, with results compared against several advanced and state-of-the-art algorithms.

All experiments were performed on a computer running the Windows operating system, equipped with an Intel(R) Core i5-7300U processor (2.50 GHz) and 8 GB of RAM. Both the ECSO algorithm and the comparative algorithms were implemented using the Python programming language.

The CEC2021 benchmark suite includes a set of 20-dimensional functions specifically designed to evaluate algorithmic performance on shifted, rotated, and biased optimization problems**.** Further details about these benchmark functions can be found in reference[19]**.** Each experiment was independently executed 30 times, with each run consisting of 2500 iterations, the population size for each optimization technique was set to 30 individuals to ensure the reliability of the statistical analysis. The algorithm was terminated once the maximum number of iterations was reached.

To compare the performance of different algorithms, standard statistical measures were employed, including the average value (Avg), median value (Med), and standard deviation (Std).

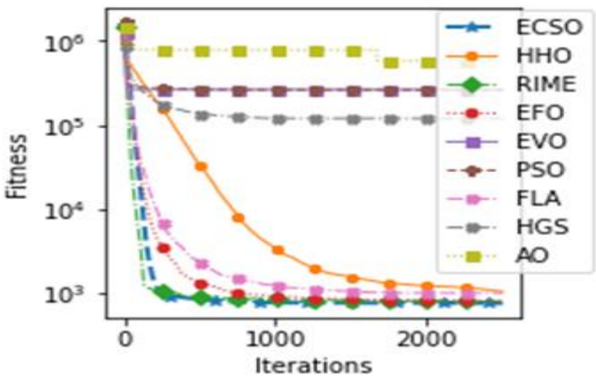
.In this research project, the effectiveness of ECSO is compared to eight variants optimization techniques including: HHO[20], RIME[21], EFO[22], EVO[23], PSO[12], FLA[24], HGS[25] and AO[26]. The authors of these publications have provided guidelines for important aspects of competition, which are outlined in Table 1. In this section, we follow these suggestions.

Table 1: Hyperparameters settings.

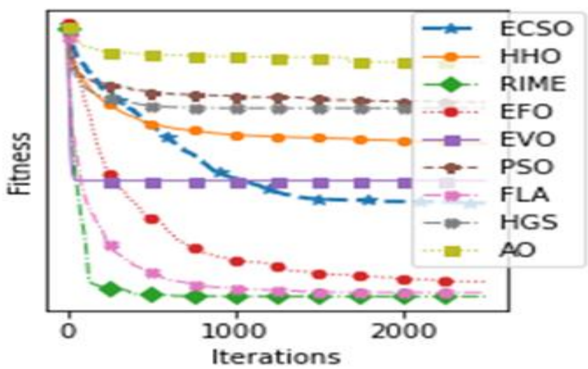
Algorithm	Parameter setting
HHO	*popSize=30;*
RIME	*Soft-rime(sr)=5.0; popSize=30;*
EFO	*r_rate=0.3; ps_rate=0.85; p_field=0.1; n_field=0.45; popSize=30;*
EVO	*popSize=30;*
PSO	*Inertia weight (wmin=0.04, wmax=0.09) Cognitive coefficient=2*
FLA	*C1=0.5; C2=2.0; C3=0.1; C4=0.2; C5=2.0; DD=0.01; popSize=30;*
HGS	*The probability of updating position (PUP)=0.08; Largest hunger / threshold (LH)= 10000*
AO	*popSize=30;*
ECSO	*c2=c3= random, μ=0.5, beta=1.5*

Table.2 Comparison of the "ECSO with some popular algorithms during 2500 iterations (cec2021)"

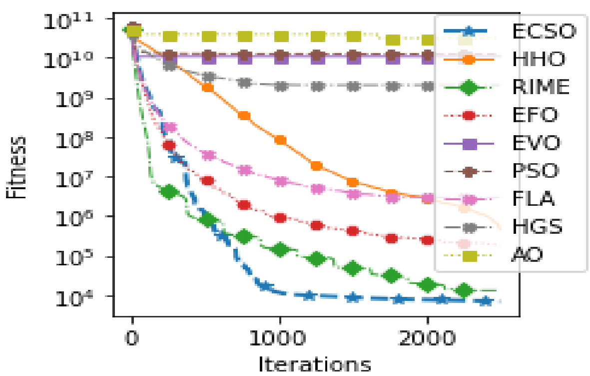
F	Criteria	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
ECSO	Avg	**6.78E+03**	2.76E+03	**7.58E+02**	1.91E+03	**2.92E+04**	4.19E+03	1.43E+05	2.34E+03	**2.63E+03**	**3.05E+03**
	Std	1.02E+04	6.27E+02	1.78E+01	4.13E+00	2.15E+04	8.92E+03	5.89E+05	2.11E+01	7.16E+01	5.53E+01
	Med	4.18E+03	2.71E+03	7.52E+02	1.90E+03	2.27E+04	2.40E+03	1.22E+04	2.33E+03	2.60E+03	3.04E+03
HHO	Avg	4.64E+05	3.73E+03	1.04E+03	1.93E+03	7.63E+05	2.21E+04	2.59E+06	2.60E+03	2.71E+03	3.33E+03
	Std	1.86E+05	5.49E+02	9.17E+01	7.81E+00	8.78E+05	1.54E+04	1.80E+06	2.90E+02	1.40E+02	1.29E+02
	Med	4.43E+05	3.77E+03	1.03E+03	1.93E+03	3.21E+05	1.75E+04	2.62E+06	2.52E+03	2.66E+03	3.32E+03
RIME	Avg	1.30E+04	**1.70E+03**	8.00E+02	**1.90E+03**	8.72E+04	**2.84E+03**	**6.15E+04**	**2.31E+03**	**2.63E+03**	3.19E+03
	Std	6.98E+03	2.80E+02	2.05E+01	1.05E+00	4.34E+04	3.13E+03	4.34E+04	9.58E+00	6.35E+01	6.30E+01
	Med	1.18E+04	1.68E+03	7.98E+02	1.90E+03	7.98E+04	1.97E+03	5.03E+04	2.30E+03	2.61E+03	3.17E+03
EFO	Avg	1.24E+10	4.63E+03	2.59E+05	5.08E+04	4.40E+06	7.21E+04	1.45E+07	2.54E+03	1.23E+04	4.71E+03
	Std	4.55E+09	4.61E+02	9.92E+04	8.42E+04	7.15E+06	3.35E+04	4.14E+07	1.10E+02	1.89E+03	8.24E+02
	Med	1.16E+10	4.65E+03	2.48E+05	2.46E+04	1.88E+06	5.83E+04	4.92E+06	2.53E+03	1.25E+04	4.68E+03
EVO	Avg	1.81E+05	1.84E+03	8.13E+02	1.91E+03	2.68E+05	9.08E+03	1.37E+05	**2.31E+03**	2.65E+03	3.22E+03
	Std	2.17E+05	3.05E+02	2.94E+01	7.18E+00	2.52E+05	8.52E+03	2.10E+05	8.39E+00	9.93E+01	8.51E+01
	Med	1.24E+05	1.82E+03	8.10E+02	1.91E+03	1.71E+05	4.28E+03	5.98E+04	2.31E+03	2.62E+03	3.18E+03
PSO	Avg	1.08E+10	3.09E+03	2.62E+05	1.71E+05	1.99E+07	1.85E+07	5.81E+07	2.43E+03	1.20E+04	5.55E+03
	Std	3.94E+09	5.20E+02	8.67E+04	2.04E+05	2.05E+07	3.37E+07	1.32E+08	5.88E+01	3.31E+03	8.28E+02
	Med	1.18E+10	3.04E+03	2.62E+05	9.98E+04	1.19E+07	2.70E+06	3.23E+07	2.41E+03	1.14E+04	5.56E+03
FLA	Avg	2.95E+06	1.74E+03	1.00E+03	1.90E+03	1.03E+05	6.42E+03	1.38E+05	**2.31E+03**	2.77E+03	3.17E+03
	Std	1.20E+06	3.24E+02	7.69E+01	1.61E+00	5.82E+04	5.25E+03	1.94E+05	3.22E+00	6.88E+01	4.48E+01
	Med	2.69E+06	1.75E+03	9.91E+02	1.90E+03	8.58E+04	4.40E+03	7.32E+04	2.31E+03	2.75E+03	3.16E+03
HGS	Avg	2.04E+09	4.49E+03	1.19E+05	5.06E+03	1.02E+06	1.85E+04	5.88E+06	2.64E+03	9.92E+03	3.79E+03
	Std	2.24E+09	8.45E+02	5.11E+04	3.36E+03	1.44E+06	1.38E+04	1.98E+07	1.58E+02	3.87E+03	4.81E+02
	Med	1.43E+09	4.32E+03	1.19E+05	4.06E+03	3.70E+05	1.42E+04	1.84E+06	2.60E+03	9.52E+03	3.64E+03
AO	Avg	3.12E+10	5.69E+03	5.75E+05	6.64E+05	1.54E+07	4.22E+07	3.27E+08	2.78E+03	2.03E+04	1.09E+04
	Std	5.86E+09	3.52E+02	8.88E+04	4.48E+05	1.08E+07	5.87E+07	3.25E+08	2.33E+02	3.71E+03	2.80E+03
	Med	3.22E+10	5.67E+03	5.83E+05	6.53E+05	1.62E+07	2.37E+07	2.17E+08	2.73E+03	2.11E+04	1.08E+04



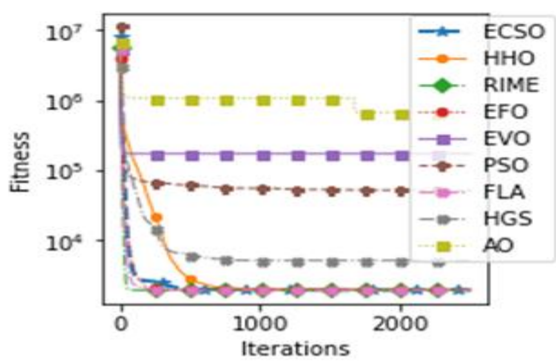
F_1



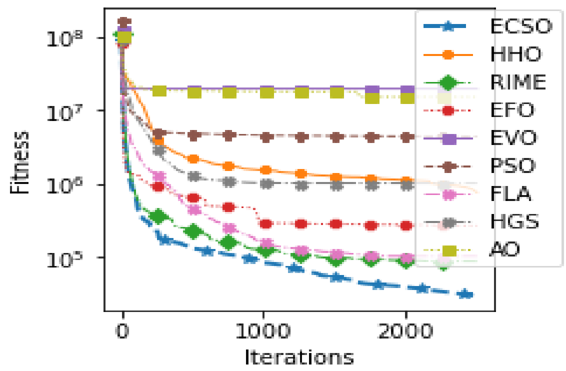
F_2

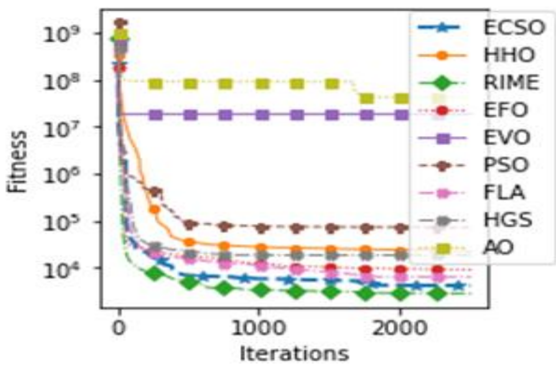


F_3

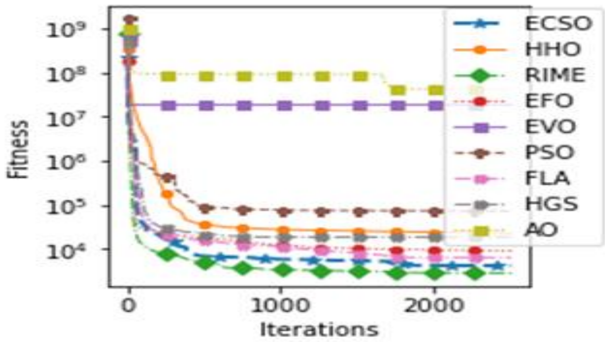


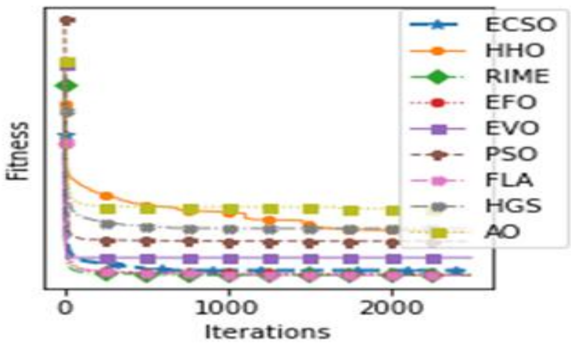
F_4



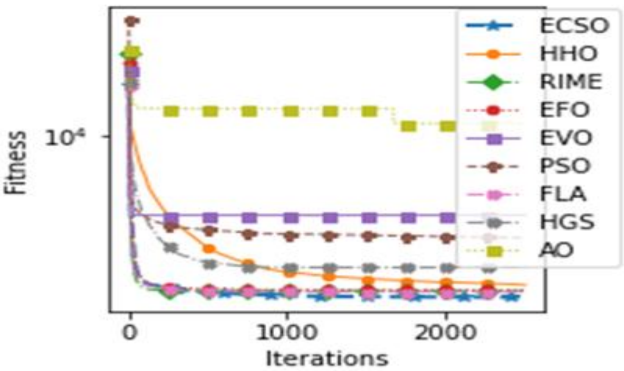


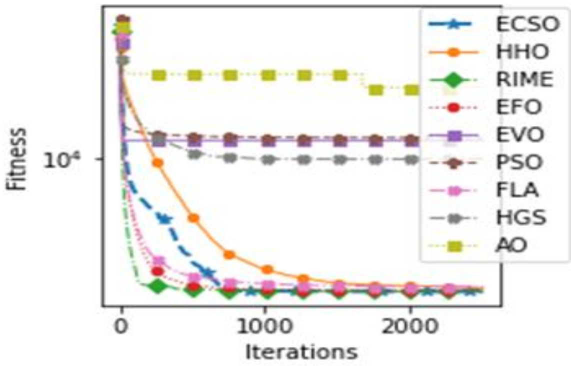
F_5 F_6





F_7 F_8





F_9 F_{10}

Fig. 2: *Performance of ECSO for CEC2021 test function* $F_1 - F_{10}$.

Table2 shows the performance comparison between the proposed ECSO algorithm and a number of common meta-innovation algorithms, namely: HO, RIME, EFO, EVO, PSO, FLA, HGS, and AO, when applied to ten standard functions of the CEC2021 packet after 2500 iterations. Three statistical values were calculated for each function and each algorithm: the arithmetic mean (AVG) to measure overall performance, the standard deviation (STD) to assess stability, and the median (Med) to measure the similarity between the results.

By analyzing the results, the following can be observed:

- F1 and F2 functions: ECSO clearly achieved the lowest average values compared to all other algorithms, demonstrating its superior ability to handle unimodal functions and quick access to the global optimal solution. The low standard deviation in these functions also demonstrates that the performance is stable and the results are not affected by the initial randomness.

- F3 and F4 functions: ECSO also outperformed all competing algorithms, with the exception of very close to AO and RIME performance in some runs, where the values were close but remained slightly higher than ECSO.
  This suggests that the ECSO algorithm is able to maintain an excellent balance between exploration and exploitation in multimodal functions.
- F5: ECSO showed the lowest average value compared to all algorithms, while EFO came in second. It is noted here that the standard deviation of EFO was relatively high, indicating its instability, while the ECSO maintained a clear stability.
- F6: ECSO performed well and balanced with RIME and AO, with very minor differences, suggesting that all three algorithms handled this function with similar efficiency, but ECSO had less standard deviation, which gives it a stability advantage.
- F7: ECSO took first place with a clear difference from the rest of the algorithms, as the corresponding values of HHO and PSO were very high, reflecting their poor ability to avoid local solutions.
- F8: ECSO results were close to HGS and AO in the mean values, but maintained the best mean and the lowest standard deviation, confirming their higher stability even in high-dimensional spaces.
- F9 and F10: In these two functions, ECSO decisively outperformed all other algorithms, especially when compared to PSO and FLA, which showed a very high standard deviation, indicating fluctuation of results. The AO algorithm was the closest performer to the ECSO but it did not outperform it in any case.

Based on the above, it can be argued that the proposed ECSO algorithm achieved superiority in eight of the ten functions (F1–F5, F7, F9, F10), and was almost equal to the best algorithms in two functions (F6 and F8) without any case being outperformed by any other algorithm. In addition, low standard deviation values across all functions show that the ECSO enjoys a high degree of stability and reliability, reflecting the efficiency of its design in achieving a dynamic balance between the exploration and exploitation phases during the research process. Compared to traditional algorithms such as PSO and FLA, ECSO is clear to excel in speed and proximity accuracy, while hybrid algorithms such as RIME and AO have shown acceptable performance but have remained below ECSO's overall performance. This confirms that the ECSO’s integrated adaptive tuning and reverse structure mechanisms have clearly improved its ability to avoid local solutions and achieve a more stable and faster convergence towards optimal solutions.

Conclusion

Through detailed numerical and statistical analysis, it is clear that the proposed ECSO algorithm has proven to be superior in performance to most of the standard functions of the CEC2021 package, achieving the best results in eight out of ten test functions, namely F1, F2, F3, F4, F5, F7, F9, and F10, while showing very close results with the other best algorithms in F6 and F8 without any performance degradation.

These results indicate that ECSO is highly adaptable to the nature of different functions, whether they are single-bottomed, multi-peaked, or high-dimensional, reflecting the strength of the adaptive search mechanism built into it.

The low standard deviation (STD) values across all functions indicate high consistency and reliability in the results, which confirms the efficiency of the algorithm in avoiding early convergence and maintaining the diversity of solutions within the research space.

When compared to algorithms such as PSO and FLA, ECSO was found to be more stable and faster in reaching optimal solutions, while newer algorithms such as AO and RIME showed acceptable performance but did not outperform ECSO in any of the experimental cases.

Therefore, it can be concluded that the proposed ECSO algorithm is one of the most efficient in achieving a dynamic balance between global exploration and local exploitation, with a strong ability to adaptively control the behavior of molecules or entities within the research space, making it a promising candidate for complex and multidimensional optimization applications.

References

[1] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*, Ieee, 2011, pp. 1–7.

[2] T. J. Ypma, "Historical development of the Newton–Raphson method," *SIAM Rev.*, vol. 37, no. 4, pp. 531–551, 1995.

[3] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand. (1934).*, vol. 49, no. 6, pp. 409–436, 1952.

[4] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John wiley & sons, 2006.

[5] P. M. Vasant, *Meta-heuristics optimization algorithms in engineering, business, economics, and finance*. IGI Global, 2012.

[6] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[7] M. Qaraad, S. Amjad, N. K. Hussein, and M. A. Elhosseini, "An innovative quadratic interpolation salp swarm-based local escape operator for large-scale global optimization problems and feature selection," *Neural Comput. Appl.*, vol. 34, no. 20, pp. 17663–17721, 2022.

[8] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," *Int. j. adv. soft Comput. appl*, vol. 5, no. 1, pp. 1–35, 2013.

[9] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[10] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[11] R. A. Gallego, R. Romero, and A. J. Monticelli, "Tabu search algorithm for network synthesis," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 490–495, 2000, doi: 10.1109/59.867130.

[12] S. Kiranyaz, T. Ince, M. Gabbouj, S. Kiranyaz, T. Ince, and M. Gabbouj, "Particle swarm optimization," *Multidimens. Part. swarm Optim. Mach. Learn. pattern Recognit.*, pp. 45–82, 2014.

[13] R. Leardi, "Genetic algorithms," *Compr. Chemom.*, vol. 1, pp. 631–653, 2009.

[14] S. Mirjalili, S. Mirjalili, A. L.-A. in engineering software, and undefined 2014, "Grey wolf optimizer," *Elsevier*, doi: 10.1016/j.advengsoft.2013.12.007.

[15] K. V Price, "Differential evolution," in *Handbook of optimization: From classical to modern approach*, Springer, 2013, pp. 187–214.

[16] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, 2008.

[17] Z. Wei, C. Huang, X. Wang, T. Han, and Y. Li, "Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization," *IEEE Access*, vol. 7, pp. 66084–66109, 2019.

[18] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci. (Ny).*, vol. 540, pp. 131–159, 2020.

[19] A. W. Mohamed, K. M. Sallam, P. Agrawal, A. A. Hadi, and A. K. Mohamed, "Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems," *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1493–1517, 2023.

[20] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/J.FUTURE.2019.02.028.

[21] H. Su *et al.*, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, pp. 183–214, 2023.

[22] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 8–22, 2016.

[23] M. Azizi, U. Aickelin, H. A. Khorshidi, and M. Baghalzadeh Shishehgarkhaneh, "Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization," *Sci. Rep.*, vol. 13, no. 1, p. 226, 2023.

[24] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, and K. M. Sallam, "Fick's Law Algorithm: A physical law-based algorithm for numerical optimization," *Knowledge-Based Syst.*, vol. 260, p. 110146, 2023.

[25] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Syst. Appl.*, vol. 177, p. 114864, 2021.

[26] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, vol. 157, p. 107250, 2021.

IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS

مجلة سامراء للعلوم الصرفة والتطبيقية

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

خوارزمية تحسين سرب المنافسة المحسّنة (ECSO) لحل مسائل التحسين

غني علي محمد *، نزار حلف حسين

قسم الرياضيات، كلية علوم الحاسوب والرياضيات، جامعة تكريت، تكريت، العراق.

معلومات البحث:

تأريخ الاستلام:

تاريخ التعديل:

تأريخ القبـــول:

تاريخ الــنشر:

الكلمات المفتاحية:

التنافس، السرب، التحسين، الاستغلال، والاستكشاف.

معلومات المؤلف

الايميل:

ga230025pcm@st.tu.edu.iq,

الموبايل:

الخلاصة:

تُقدّم هذه المقالة خوارزمية التحسين التنافسي المحسَّن(ECSO) ، والتي تم تطويرها بهدف تحسين التوازن بين الاستكشاف والاستغلال في مسائل الأمثلية الصعبة. تعتمد الخوارزمية المقترحة ECSO على معاملات تحكم تكيفية وآلية محسّنة لقائد–تابع، مما يساعد على توجيه عوامل البحث بكفاءة نحو الحل الأمثل الشامل مع تجنّب الوصول المبكر إلى حل غير مثالي، ولأجل تقييم أدائها، تم اختبار خوارزمية ECSO باستخدام مجموعة من دوال الاختبار القياسية من حزمة CEC2021، ومقارنتها بعدد من الخوارزميات فوق الحدسية المعروفة. وقد أظهرت النتائج المقارنة أن ECSO حققت باستمرار أقل قيم لمتوسط الدالة الهدف مع أصغر انحراف معياري، مما يشير إلى دقة أعلى واستقرار أكبر ومتانة أفضل .تُظهر هذه النتائج أن خوارزمية ECSO تحقق توازناً جيداً بين الاستكشاف الشامل والاستغلال المحلي، مما يجعلها استراتيجية أمثلية قوية وموثوقة لحل مسائل الأمثلية عالية الأبعاد وغير الخطية.