International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 5, October 2024, pp. 5672~5680
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i5.pp5672-5680

An enhanced Giza Pyramids construction for solving optimization problems

Asmaa Hekal Omar, Naglaa M. Mostafa, Abeer S. Desuky, Lamiaa M. El Bakrawy
Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt

Article Info

Article history:

Received Dec 24, 2023
Revised Jul 4, 2024
Accepted Jul 9, 2024

Keywords:

Benchmark test functions
Giza Pyramids construction algorithm
IEEE CEC 2019 benchmarks
Meta-heuristic algorithms optimization
Wilcoxon rank-sum

ABSTRACT

Many real-world optimization problems can be solved by various algorithms that are not fast in convergence or gain enough accuracy. Meta-heuristic algorithms are used to solve optimization problems and have achieved their effectiveness in solving several real-world optimization problems. Meta-heuristic algorithms try to find the best solution out of all available solutions in the possible shortest time. A good meta-heuristic algorithm is characterized by its accuracy, convergence speed, and ability to solve high dimensions' problems. Giza Pyramids construction (GPC) has recently been introduced as a physics-inspired optimization method. This paper suggests an enhanced Giza Pyramids construction (EGPC) by adding a new parameter based on the step length of each individual and iteratively revises the individual' position. The EGPC algorithm is suggested for improving the GPC exploitation and exploration. Experiments were performed on 23 benchmark functions and four IEEE CEC 2019 benchmarks to test the performance of the proposed EGPC algorithm. The experimental results show the high competitiveness of the EGPC algorithm compared to the basic GPC algorithm and another four well known optimizers in terms of improved exploration, exploitation, convergence' rate, and the avoidance of local optima.

Corresponding Author:

Asmaa Hekal Omar
Department of Mathematics, Faculty of Science, Al-Azhar University
Nasr City, Cairo 11754, Egypt
Email: asmaahekl@azhar.edu.eg

1. INTRODUCTION

Many problems in the real world are classified as optimization problems, also many difficult problems in technology and science can be expressed as optimization problems. These problems can help you save money, risk and time while also improving efficiency, quality, and profit. Many optimization problems are difficult to solve so different optimization algorithms are being developed to deal with these difficult problems. Approximation algorithms are presented as novel techniques to solve these problems due to the absence of good accurate optimization methods [1], [2]. There are two types of approximate algorithms: heuristic and Meta-heuristic algorithms. Because Heuristic algorithms are usually designed and used for specific problems, it is less used. Meta-heuristic algorithms have lately been widely employed to process the most complex real-world optimization [2]. Meta-heuristic algorithms are stronger, more flexible, and easier to design and implement. They are faster to solve problems and solve bigger problems.

The meta-heuristics are classified as optimization in computer science and applied mathematics. It can deal with algorithms and complicated computation theory. There are many areas that can be covered by meta-heuristics such as mathematical programming, artificial intelligence, computational intelligence, operations research, and soft computing. In real-world meta-heuristics are very effective in solving complicated issues and have also taken an important role in lowering costs, time and expanding widely in different fields [3].

Meta-heuristics can be classified into three categories: evolutionary-based, trajectory-based, nature-inspired methods, and recently ancient-inspired. Some techniques of algorithms can be classified into various categories. Evolutionary-based is based on the concept of competition and the evolution of the population. It simulates evolution of species. Usually, populations are generated at random, and each population is a solution. The appropriate solution is selected by an objective function. The population selected has better suitability with a higher probability, then is reproduced by different operators such as mutation and crossover to generate a new generation of children. Finally, the parents and children are decided to use. The popular Examples of it are differential evolution (DE) [4], memetic algorithm (MA) [5], genetic algorithm (GA) [6], and harmony search (HS) [7]. Trajectory-based are techniques that work on a solution. Usually, their characteristics are dictated by the nature of the problem. They make one solution better. These are repeated processes through which they go from one solution to another. It has proven to be effective in different problems. popular examples of it are tabu search (TS) [8], simulated annealing (SA) [9], guided local search (GLS) [10], iterated local search (ILS) [11]. Nature-inspired are techniques which follow nature rules which are understandable and simple. Most species behave collectively as a searcher in the issue space, leading them to the target and the solution [12]. It includes bio-inspired such as crow search algorithm (CSA) [13], grey wolf optimizer (GWO), [14] and ant lion optimizer (ALO) [15], swarm-based such as particle swarm optimization (PSO) [16], artificial bee colony (ABC) [17], and ant colony optimization (ACO) [18], Human-based such as imperialist competitive algorithm (ICA) [19], cultural algorithm (CA) [20], and recently published political optimizer (PO) [21], physics/chemistry-based such as chemical reaction optimization (CRO) [22], black hole (BH) [23], multi-verse optimizer (MVO) [24], and plant-based techniques such as invasive weed optimization (IWO) [25], and artificial root foraging algorithm (ARFA) [26]. The technique of Meta-heuristic algorithms to search and generate optimal solutions for an optimization problem depends on two major concepts: exploration and exploitation [27]. The meta-heuristic algorithm works well when it strikes a balance between exploitation and exploration. At the generation stage they use exploration to create new solutions and decrease exploration as the optimization process progresses, the exploitation process progressively rises while the exploration process is lowered [2].

The Giza Pyramids construction (GPC) algorithm, which was presented in [1], is the first meta-heuristic algorithm with an ancient feel. The Giza Pyramids complex consists of three huge pyramids that were constructed in ancient Egypt during the fourth dynasty [28]. The Khufu Pyramid, sometimes referred to as the seven wonders, is the biggest pyramid. Menkaure and Khafre are the names of the other two pyramids [29]. The researchers demonstrated how the building methods for pyramids have evolved over time and are now distinct from one another. This work proposes an enhanced Giza Pyramids construction (EGPC) that adds a new parameter that depends on the step length of each person while rewriting the individual position, hence improving the optimization efficiency and accuracy of GPC. The results of experiments show that the EGPC algorithm compared to the original GPC algorithm and other state-of-the-art algorithms has better exploitation and exploration capability compared to other algorithms.

The rest of this paper structure is as follows: section 2 explores the materials and methods. Section 3 presents experimental results and discussion. Finally, section 4 represents conclusions.

2. MATERIALS AND METHODS

2.1. Giza Pyramids construction

2.1.1. The construction method

The Pyramids of Giza are the largest and most famous pyramid construction in the world. Regarding the building techniques of the pyramid, numerous theories have been put forth, but none of them have gained complete acceptance. There is a common belief that these pyramids' stones had been removed from mines, moved, and then assembled. Ramps have been installed at higher elevations by workers. The Greeks believed the unfair usage of slaves to build the pyramids, but recent studies suggest that the builders were highly talented. Managing the labor force was the main issue encountered throughout the construction of the pyramids. Notwithstanding the scarcity of hardware stores, the quantity of stone blocks needed in construction, and the very short construction period, pyramid construction has been optimized. It took ten to twenty years to build. A maximum of forty thousand persons were involved in the construction of the pyramids, with an average of fourteen thousand. Two million stones were used in the construction of the greatest pyramid. Slaves, masons, coolies, carpenters, metalworkers, and foremen were among the laborers [1].

2.1.2. The motivation

The workmen, who included slaves, masons, metalworkers, carpenters, and coolies, were supervised by a skilled agent known as Pharaoh's special agent, who served as a foreman. With the highest rank in the group, the Pharaoh's agent oversees the laborers while toting a stone block. Every employee at the construction site has a rank or position. Because there is competition for sublime rank, employees perform better in order to be rewarded with sublime rank. Every employee can also acquire new knowledge and abilities as a source of further motivation. Additionally, the laborers become exhausted during the stone block transportation stage and must take a short break, failing which they will be replaced by invigorated and youthful laborers.

Stone blocks are gathered daily from the surrounding area of the construction site and transported to the pyramid by workers like miners. The pyramid was constructed using ramps. They have to go the distance from where the stone block is installed in the pyramid to where it is located. The workers' ability determines the distance traveled. If there are sufficient laborers available, more stone blocks are gathered during the workday for the pyramid's installation location. The stone block moves in response to changes in initial velocity, friction force, and ramp gradient [1].

2.2. The proposed algorithm

The pseudocode of the EGPC algorithm is:

Algorithm 1. EGPC algorithm
```
step 1:
    Create the starting population (population size) as an array of stone blocks or
    individuals;
    Create cost and location of individual or stone block;
    Dictate the best individual;
      step 2: for first iteration to max iteration do
        step3: for i=1: n do (every n stone blocks or individuals)
                   Determine the value of stone block's displacement in (1);
                   Determine the value of individual movement in (2);
                   Examine new location in (3);
                   Investigate the possibility of substituting individual in (4);
                   Update location of each individual
                   Calculate a new parameter, each individual' step length determines this
                    parameter.
                   Calculate the new cost of the new location;
                  if new cost < individual cost then
                      Set new cost as individual cost;
                  end if
        end step 3
              sort solutions to the following iteration;
      end step 2
end step 1
```

The algorithm describes the EGPC algorithm which is used (1) to determine a stone block's displacement on the ramp:

$$d = \frac{v_0}{2g(sin\theta + \mu_k cos\theta)} \tag{1}$$

where θ is the angle the ramp makes with the horizon. d is the amount of stone block's displacement. g is the earth's gravity which is 9.8. v_0 is the stone block's starting velocity and is calculated by a uniformly distributed random number between 0 and 1 in every iteration. μ_k is the kinetic friction coefficient equal to $rand[min\mu_k, max\mu_k]$ the min μ_k and the max μ_k are pre-decided, and every iteration assumes between these two values random number. The new location of the worker pushing the stone block is found from (2):

$$x = \frac{v_0^2}{2g\ sin\theta} \tag{2}$$

where x is the worker movement value. A new location can be calculated from the result of the two (1) and (2). So, from the algorithm in (3), to get a new location (a new solution).

$$\vec{\mathrm{P}} = \left(\vec{\mathrm{P_i}} + \mathrm{d}\right) \times \mathrm{x}\overrightarrow{\epsilon_i} \tag{3}$$

where $\overrightarrow{P_i}$ is the current location, and $\overrightarrow{\epsilon_i}$ is a random vector which uses the Levy, Normal, or Uniform distribution. The primary solutions of the problem are $\emptyset = (\emptyset_1, \emptyset_2, \ldots, \emptyset_n)$, and the generated solutions using (3) are $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$, with a fifty percent probability. We will have new solutions since some of the created solutions will replace some of the original ones, $Z = (\xi_1, \xi_2, \ldots, \xi_n)$, equation (4) is used for substitution.

$$\xi_k = \begin{cases} \omega_k \,, if\; rand[0,1] \leq 0.5 \\ \emptyset_k \,,\; otherwise \end{cases} \tag{4}$$

To improve the basic GPC performance, the EGPC algorithm is suggested. A new parameter is added to accomplish the enhancement. Each individual's step length determines this parameter. During the exploitation phase, the search space's upper, lower, and parameter Mu are used to adjust each individual for iterations and improve the ratio of exploration to exploitation of the search' space.

3. EXPERIMENTAL RESULTS AND DISCUSSION

Every experiment was carried out on a 64 bit Windows 10 system that had an Intel (R) Core (TM) i7 processor, 2.4 GHz CPU, 16 GB RAM, and MATLAB R2018a. For every method, the maximum iterations' number (Max-Iteration) was selected to be 1000. On each benchmark function, each algorithm was executed 30 times independently, and the average of the most optimal solution was given. Table 1 displays the remaining variables.

3.1. Benchmark functions description and parameter settings

To test the effectiveness of the suggested EGPC algorithm, 23 benchmark functions were used in experiments, these functions were classified into three groups:

- (F_1 - F_7) Unimodal functions, which have a single global optimum and are capable of assessing an algorithm's exploitation potential and convergence.
- (F_8 - F_{13}) Multimodal functions, which can assess an algorithm's capacity for exploration and avoidance of local optima and have a large number of local optima.
- (F_{14} - F_{23}) Fixed dimensional multimodal functions, which are useful for evaluating how well an algorithm finds a global optimum. These functions' specifics are provided in [30].

Table 1. Setting of parameters for algorithms

Algorithm	Setting of parameters
PSO	Max-inertia weight of 0.9, and min-weight of 0.2
	cognitive coeff. (c_1) equals cognitive coeff. (c_2)=2
WOA	$\vec{a}$ decreases in linear from 2 to 0
	b=1, $\vec{a}_2$ decreases in linear from -1 to -2
GWO	$\vec{a}$ decreases in linear from 2 to 0
ALO	a constant called W has ranges between 2 and 6 depending on the iteration.
GPC	the earth's gravity =9.8, ramp's angle =14, initial velocity =*rand*(0,1)
	Substitution probability=0.9. max- friction= min-friction=-100
EGPC	the earth's gravity =9.8, ramp's angle =14, initial velocity =*rand*(0,1)
	Substitution probability=0.9. max-friction= min-friction=-100

3.2. Results and discussion

The original GPC algorithm and four recent popular algorithms: ALO [15], GWO [14], WOA [31], and PSO [16] are used in a comparison with the proposed EGPC algorithm. Table 2 compares the performance of these four algorithms with the proposed algorithm on 23 benchmark functions and four IEEE CEC 2019 benchmarks [32]. It demonstrates the high competitiveness of the EGPC algorithm in comparison with ALO, GWO, WOA, PSO, and the original GPC algorithm. It achieves exact optimum results for functions F9 and F11 and the best optimal solutions for functions F5, F7, F8, F10, F14, F15, and F18 in terms of average. In the case of function F16, F17, F21, F22, CEC1, and CEC3, EGPC algorithm achieves the 2-nd best average optimal' value. The following results demonstrate that the EGPC algorithm has the best exploitation and exploration capabilities. comparing with other algorithms. Figure 1 demonstrates a 2-Dview of some benchmark functions. The proposed EGPC achieves a higher convergence' rate from the first iterations' steps for the functions F5, F7, F10, and F18 and CEC3 as shown in Figure 2.

Table 2. The results of five algorithms and proposed algorithm on benchmark functions and IEEE CEC 2019 benchmark results

Fn	GPC	EGPC	ALO	GWO	WOA	PSO
F1	1.25E-26	5.58E-09	2.40E-98	3.04E-142	3.04E-142	7.30E-35
F2	4.38E-17	6.27E-15	0.4641	3.08E-56	1.08E-101	6.73E-20
F3	6.95E-33	6.30E-31	1.08E-04	6.12E-46	56.8676	4.59E-11
F4	3.63E-18	2.26E-15	1.21E-04	8.58E-31	1.9959	8.41E-08
F5	7.6978	0.5445	35.5666	7.1541	6.7318	3.8044
F6	1.2004	0.0109	3.47E-09	1.22E-06	0.001	1.36E-32
F7	7.69E-05	3.32E-05	0.0203	5.59E-04	9.56E-04	0.0029
F8	-2.22E+03	-9.86E+03	-2.58E+03	-2.71E+03	-2.96E+03	-2.28E+03
F9	0	0	25.4709	0.1992	0	6.5667
F10	8.88E-16	8.88E-16	3.29E-01	4.44E-15	3.73E-15	7.28E-15
F11	0	0	0.1758	0.0128	0.0823	0.2228
F12	2.89E-01	4.60E-03	2.10E+00	6.97E-07	1.70E-02	3.52E-30
F13	7.74E-01	2.36E-02	4.20E-03	2.11E-06	2.47E-02	6.40E-32
F14	12.6705	1.0109	1.7924	2.5845	4.5375	3.9597
F15	0.0021	5.59E-04	0.0048	0.0043	7.06E-04	7.96E-04
F16	-0.9274	-1.023	-1.0316	-1.0316	-1.0316	-1.0316
F17	1.2154	0.4043	0.3979	0.3979	0.3979	0.3979
F18	23.09	3	3	3	3.0001	3
F19	-3.1681	-3.4066	-3.8628	-3.8599	-3.8592	-3.8628
F20	-1.2924	-2.5921	-3.2506	-3.2382	-3.2406	-3.2269
F21	-1.2131	-9.4601	-5.65	-10.1526	-8.1124	-7.1546
F22	-1.3064	-9.9296	-8.6899	-10.4024	-8.0544	-9.3481
F23	-2.6332	-9.9163	-6.7839	-10.5354	-6.7977	-10.5364
CEC1	6.1627e+04	7.8246e+04	1.3120e+10	3.1639e+08	6.2856e+10	1.8340e+12
CEC2	17.8701	17.8326	17.3456	17.3437	17.3534	1.4204e+04
CEC3	12.7026	12.7025	12.7024	12.7024	12.7024	12.7024
CEC4	7.1647e+03	1.4316e+04	30.4459	276.6082	478.6463	18.9045

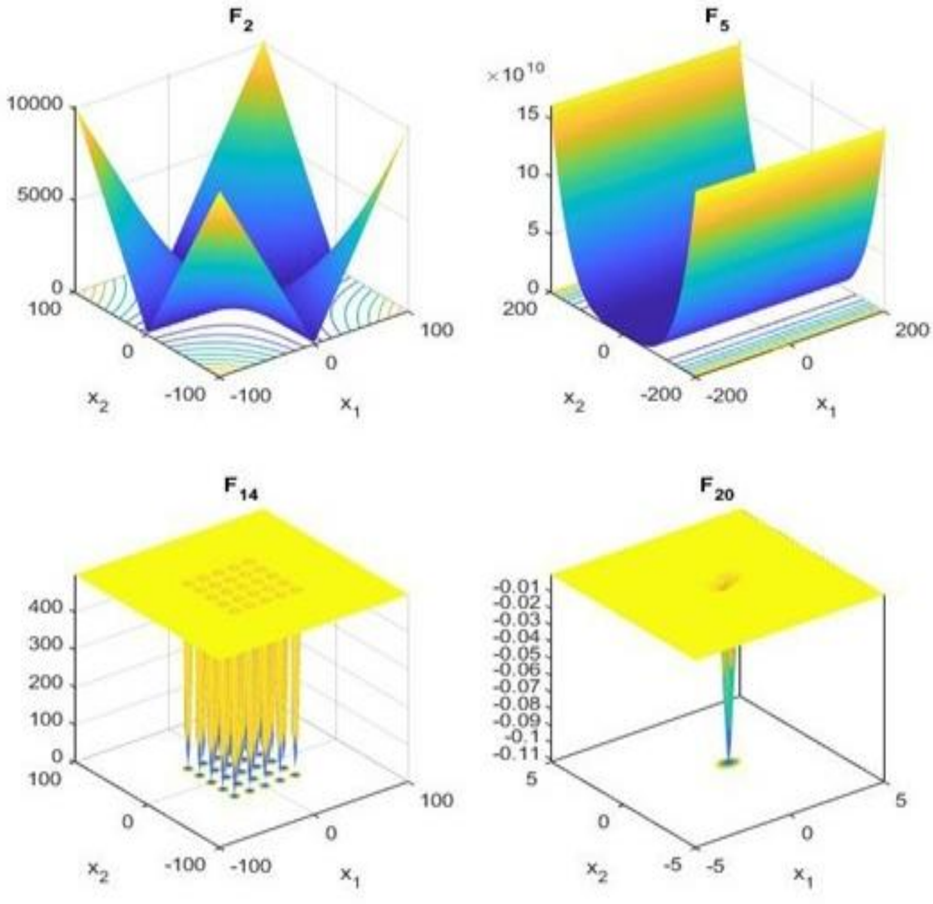
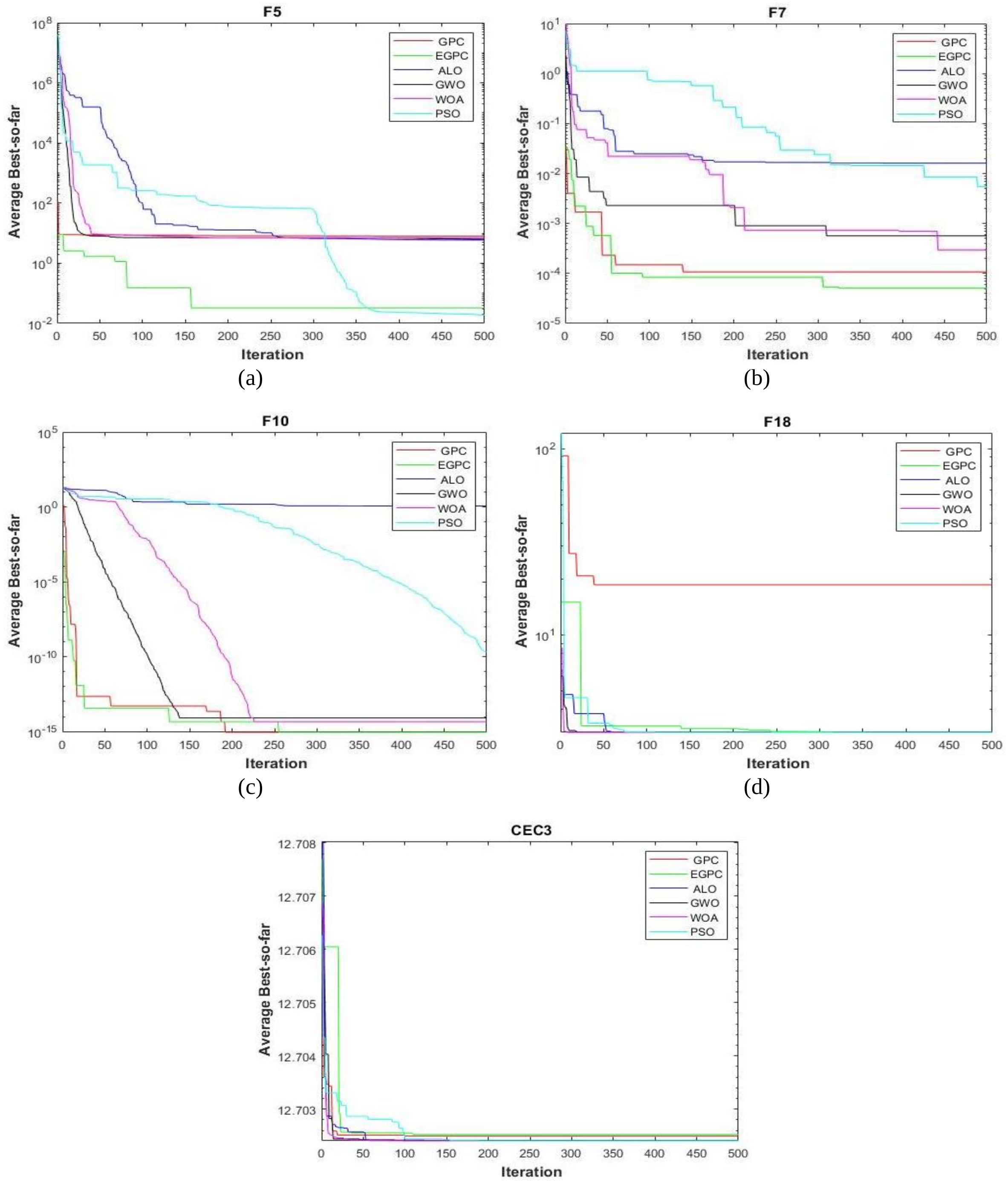


Figure 1. 2-D view of some benchmark functions



(e)

Figure 2. Convergence curves of algorithms for some benchmark functions and IEEE CEC 2019 benchmarks

For more validation to our results, we used the non-parametric Wilcoxon rank-sum test (WRS) [33] to compare the EGPC algorithm with other optimization algorithms. Table 3 displays the pairwise p-values of this test, with "*NaN*" indicating "Not a Number". Symbols "∨", "∧", and "≈" denote significant inferiority, superiority, or similarity of EGPC to other algorithms. Results in Table 3 reveal EGPC's superiority over GPC in 16 functions, inferiority in 3 functions, and similarity in 5 functions. EGPC also outperforms PSO in 15 functions, is inferior in 11 functions, and similar in 1 function. Additionally, EGPC surpasses WOA in 13 functions, lags behind in 9 functions, and matches in 5 functions. Compared to ALO and GWO, EGPC excels in 12 functions but falls short in 10 and 12 functions respectively while being similar in others. These p-values demonstrate EGPC's significant performance over other algorithms.

Table 3. Pairwise WRS p-values of EGPC vs the used algorithms

Fn	GPC	ALO	GWO	WOA	PSO
F1	1.83E-04	1.83E-04	1.83E-04	4.27E-01	1.83E-04
	∧	∧	∧	≈	∧
F2	1.83E-04	1.83E-04	1.83E-04	8.90E-02	1.83E-04
	∧	∧	∧	≈	∧
F3	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	∧	∧	∧	∧	∧
F4	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	∧	∧	∧	∧	∧
F5	1.83E-04	2.83E-03	1.83E-04	1.83E-04	2.20E-03
	∧	∧	∧	∧	∧
F6	1.83E-04	1.83E-04	2.83E-03	3.76E-02	1.83E-04
	∧	∨	∨	∧	∨
F7	5.21E-01	1.83E-04	4.40E-04	1.01E-03	1.83E-04
	≈	∧	∧	∧	∧
F8	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	∧	∧	∧	∧	∧
F9	NaN	6.39E-05	5.97E-03	1.49E-02	6.39E-05
		∧	∧	∧	∧
F10	NaN	6.39E-05	3.29E-05	4.04E-05	6.39E-05
		∧	∧	∧	∧
F11	NaN	6.39E-05	7.51E-04	5.97E-03	6.39E-05
		∧	∧	∧	∧
F12	1.83E-04	1.40E-01	3.61E-03	1.01E-03	1.83E-04
	∧	≈	∨	∧	∨
F13	1.83E-04	2.57E-02	6.40E-02	1.83E-04	1.83E-04
	∧	∨	≈	∧	∨
F14	1.01E-03	1.40E-01	1.83E-04	7.91E-01	2.56E-02
	∧	≈	∧	≈	∧
F15	2.12E-01	1.71E-03	4.73E-01	2.11E-02	5.80E-03
	≈	∧	≈	∨	∧
F16	1.40E-02	1.83E-04	1.83E-04	1.83E-04	8.74E-05
	∨	∨	∨	∨	∨
F17	2.83E-03	1.83E-04	1.83E-04	1.83E-04	6.39E-05
	∨	∨	∨	∨	∨
F18	2.20E-03	1.83E-04	3.76E-02	3.76E-02	1.74E-04
	∧	∨	∨	∨	∨
F19	6.23E-01	1.83E-04	1.83E-04	1.83E-04	1.29E-04
	≈	∨	∨	∨	∨
F20	1.01E-03	1.83E-04	1.83E-04	1.83E-04	1.41E-04
	∧	∨	∨	∨	∨
F21	1.83E-04	1.40E-01	3.12E-02	7.28E-03	4.71E-01
	∧	≈	∨	∧	≈
F22	1.83E-04	1.40E-01	1.62E-01	3.45E-01	2.27E-03
	∧	≈	≈	≈	∨
F23	1.83E-04	1.40E-01	1.73E-02	3.07E-01	2.44E-02
	∧	≈	∨	≈	∨
CEC1	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	∨	∧	∧	∧	∧
CEC2	3.85E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	≈	∨	∨	∨	∧
CEC3	3.85E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	≈	∨	∨	∨	∧
CEC4	1.01E-03	1.46E-04	1.83E-04	1.83E-04	6.39E-05
	∧	∨	∨	∨	∨

4. CONCLUSION

The GPC algorithm has been improved in this paper using the EGPC algorithm, which adds a new parameter based on the step length of each individual and iteratively revises the individual' position to improve the fundamental GPC performance. To better balance the processes of exploration and exploitation, this new parameter, upper, and lower bounds of search' space were used. The EGPC algorithm's performance was compared to four recent popular optimizers and tested using 23 benchmark functions and four IEEE CEC 2019 benchmarks. The results demonstrated that the EGPC algorithm has higher exploitation capabilities for the unimodal functions, and high exploration capabilities for the multimodal, and the fixed dimensional multimodal functions. In addition, the proposed EGPC algorithm has ability to escape from the local optima in addition to achieving a higher convergence' rate than the original GPC and the other recent optimizers. Moreover, the WRT was used, showing that the proposed EGPC outperforms other optimization algorithms significantly in terms of performance.

REFERENCES

[1] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Giza Pyramids construction: An ancient-inspired metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 14, no. 4, pp. 1743–1761, Jul. 2020, doi: 10.1007/s12065-020-00451-3.

[2] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, "Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Advances in Engineering Software*, vol. 174, p. 103282, Dec. 2022, doi: 10.1016/j.advengsoft.2022.103282.

[3] S. Darvishpoor, A. Darvishpour, M. Escarcega, and M. Hassanalian, "Nature-inspired algorithms from oceans to space: A comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones," *Drones*, vol. 7, no. 7, p. 427, Jun. 2023, doi: 10.3390/drones7070427.

[4] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Engineering Journal*, vol. 61, no. 5, pp. 3831–3872, May 2022, doi: 10.1016/j.aej.2021.09.013.

[5] L. Moalic and A. Gondran, "Variations on memetic algorithms for graph coloring problems," *Journal of Heuristics*, vol. 24, no. 1, pp. 1–24, Aug. 2017, doi: 10.1007/s10732-017-9354-9.

[6] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm- a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, IEEE, Feb. 2019, doi: 10.1109/comitcon.2019.8862255.

[7] J. Dias Santos, F. Marques, L. P. Garcés Negrete, G. A. Andrêa Brigatto, J. M. López-Lezama, and N. Muñoz-Galeano, "A novel solution method for the distribution network reconfiguration problem based on a search mechanism enhancement of the improved harmony search algorithm," *Energies*, vol. 15, no. 6, p. 2083, Mar. 2022, doi: 10.3390/en15062083.

[8] V. K. Prajapati, M. Jain, and L. Chouhan, "Tabu search algorithm (TSA): A comprehensive survey," in *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, IEEE, Feb. 2020, doi: 10.1109/icetce48199.2020.9091743.

[9] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of Metaheuristics*, Springer International Publishing, 2018, pp. 1–35, doi: 10.1007/978-3-319-91086-4_1.

[10] F. Arnold and K. Sörensen, "Knowledge-guided local search for the vehicle routing problem," *Computers & Operations Research*, vol. 105, pp. 32–46, May 2019, doi: 10.1016/j.cor.2019.01.002.

[11] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of Metaheuristics*, Springer International Publishing, 2018, pp. 129–168, doi: 10.1007/978-3-319-91086-4_5.

[12] K. G. Dhal, S. Ray, A. Das, and S. Das, "A survey on nature-inspired optimization algorithms and their application in image enhancement domain," *Archives of Computational Methods in Engineering*, vol. 26, no. 5, pp. 1607–1638, Sep. 2018, doi: 10.1007/s11831-018-9289-9.

[13] S. N. Kumar *et al.*, "Improved crow search algorithm based on arithmetic crossover—a novel metaheuristic technique for solving engineering optimization problems," in *Multi-Objective Combinatorial Optimization Problems and Solution Methods*, Elsevier, 2022, pp. 71–91, doi: 10.1016/b978-0-12-823799-1.00011-5.

[14] S. M. Ahmad, H. B. Marqas, and R. B. Asaad, "Grey wolf optimizer: Overview, modifications and applications," *International Research Journal of Science*, vol. 1, no. 1, pp. 44–56, 2021, doi: 10.5281/zenodo.5195644.

[15] L. Abualigah, M. Shehab, M. Alshinwan, S. Mirjalili, and M. A. Elaziz, "Ant lion optimizer: A comprehensive survey of its variants and applications," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1397–1416, Apr. 2020, doi: 10.1007/s11831-020-09420-6.

[16] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, Jan. 2017, doi: 10.1007/s00500-016-2474-6.

[17] M. I. Jarrah *et al.*, "A novel explanatory hybrid artificial bee colony algorithm for numerical function optimization," *The Journal of Supercomputing*, vol. 76, no. 12, pp. 9330–9354, Feb. 2020, doi: 10.1007/s11227-019-03083-2.

[18] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/mci.2006.329691.

[19] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *2007 IEEE Congress on Evolutionary Computation*, IEEE, Sep. 2007, doi: 10.1109/cec.2007.4425083.

[20] M. Airoldi and J. Rokka, "Algorithmic consumer culture," *Consumption Markets & Culture*, vol. 25, no. 5, pp. 411–428, Jun. 2022, doi: 10.1080/10253866.2022.2084726.

[21] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: A novel socio-inspired meta-heuristic for global optimization," *Knowledge-Based Systems*, vol. 195, p. 105709, May 2020, doi: 10.1016/j.knosys.2020.105709.

[22] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, Jun. 2010, doi: 10.1109/tevc.2009.2033580.

[23] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, Feb. 2013, doi: 10.1016/j.ins.2012.08.023.

[24] L. Abualigah, "Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12381–12401, Mar. 2020, doi: 10.1007/s00521-020-04839-1.

[25] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, Dec. 2006, doi: 10.1016/j.ecoinf.2006.07.003.

[26] L. Ma, Y. Zhu, Y. Liu, L. Tian, and H. Chen, "A novel bionic algorithm inspired by plant root foraging behaviors," *Applied Soft Computing*, vol. 37, pp. 95–113, Dec. 2015, doi: 10.1016/j.asoc.2015.08.014.

[27] H. Shayanfar and F. S. Gharehchopogh, "Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 71, pp. 728–746, Oct. 2018, doi: 10.1016/j.asoc.2018.07.033.

[28] M. Verner, *The pyramids: the mystery, culture, and science of Egypt's great monuments*. Open Road+Grove/Atlantic, 2007.

[29] G. Magli, "Akhet Khufu: Archaeo-astronomical hints at a common project of the two main Pyramids of Giza, Egypt," *Nexus Network Journal*, vol. 11, no. 1, pp. 35–50, Apr. 2009, doi: 10.1007/s00004-008-0080-3.

[30] A. S. Desuky, S. Hussain, S. Kausar, M. A. Islam, and L. M. El Bakrawy, "EAOA: An enhanced Archimedes optimization algorithm for feature selection in classification," *IEEE Access*, vol. 9, pp. 120795–120814, 2021, doi: 10.1109/access.2021.3108533.

[31] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[32] J. M. Abdullah and T. Ahmed, "Fitness dependent optimizer: Inspired by the bee swarming reproductive process," *IEEE Access*, vol. 7, pp. 43473–43486, 2019, doi: 10.1109/access.2019.2907012.

[33] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, p. 80, 1945, doi: 10.2307/3001968.

BIOGRAPHIES OF AUTHORS

Asmaa Hekal Omar received the BSc, MSc, and PhD degrees from the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt, in 2002, 2009, and 2013, respectively, where she is currently an associate professor in computer science. She has coauthored publications in different journals. Her research interests include machine learning and optimization. She can be contacted at email: asmaahekl@azhar.edu.eg.

Naglaa M. Mostafa is a lecture of computer science at Al-Azhar University (girls) in Cairo, Egypt. She was awarded a BSc degree in computer science and pure mathematics by Faculty of Science, Menoufia University in 1996. She received her MSc and PhD in computer science from Faculty of Science, Al-Azhar University, Cairo, Egypt in 2009, 2012, respectively. She can be contacted at email: NaglaaMohammed.2159@azhar.edu.eg.

Abeer S. Desuky received the BSc degree in science, in 2003, and the MSc and PhD degrees in computer science, in 2008 and 2012, respectively. She is currently professor in computer science with the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published several research papers in the field of AI, machine learning, metaheuristic optimization, and data mining and analysis. She is also a supervisor of some master's and PhD theses. She is a reviewer in many Scopus-indexed journals, such as IEEE Access, *Egyptian Informatics Journal*, and *Advances in Systems Science and Applications* (ASSA). She can be contacted at email: abeerdesuky@azhar.edu.eg.

Lamiaa M. El Bakrawy received the MSc and PhD degrees in computer science from the Faculty of Science, Al-Azhar University, Egypt, in 2009 and 2012, respectively. Her master's topic was optimization using swarm intelligence. Her PhD degree topic was machine learning in image authentication. She is currently an associate professor with the Mathematics and Computer Department, Faculty of Science, Al-Azhar University. She has published many articles in well-known international journals. Her research interests include image processing, social network analysis, computational intelligence, machine learning, meta-heuristic optimization, and information security. She has reviewed research papers for many Scopus-indexed journals. She is also a program committee member in various international conferences. She can be contacted at email: dr_lamiaa_el_bakrawy@azhar.edu.eg.