Hybridization of Social Group Optimization and Past Present Future Algorithm and its Application for Software Cost Estimation

Sagiraju Srinadhraju, Samaresh Mishra, Suresh Chandra Satapathy

School of Computer Engineering, KIIT Deemed to be University, India
Email: 2181073@kiit.ac.in

In this research, the hybrid SGO-PPF (Social Group Optimization based Past Present Future) method is used to optimize the parameters of the Constructive Cost Model (COCOMO) in order to predict the software effort. The COCOMO model's performance is entirely dependent on its parameters, which can be adjusted by the application of meta-heuristic techniques. In order to improve the COCOMO model, this work employs a hybrid SGO-PPF method, which combines the Social Group Optimization (SGO) with the Past Present Future method (PPF). The algorithm SGO demonstrates the social behavior of the human to improve their knowledge level to solve complex problems in the exploration phase. The exploration phase is further enhanced by using PPF as future improvement of the person depends on experience and present work. The algorithm has been analyzed using COCOMO81 and the Turkish industry software projects dataset and assesses the performance using standard evaluation metrics, such as the mean magnitude of relative error (MMRE) and the Manhattan distance (MD). The algorithm has been compared with many other approaches. The results indicate that the SGO-PPF-based optimization significantly enhances the predictive power of software cost estimation models in compassion to other algorithms, making them more adaptable to the intricacies of modern software development.

Keywords: COCOMO81, Turkish industry software projects, PPF, Optimization.

1. Introduction

Optimization issues that are unsolvable, impractical, or that cannot be addressed by conventional approaches are solved using meta-heuristic techniques which are inspired by natuer. These algorithms search the search space for optimal or nearly ideal solutions and these influenced by natural phenomenon like simulated annealing, behavior of swarm, as well as evolution [1]. In the fields of artificial intelligence which incorporates optimization, heuristic optimization algorithm hybridization is an emerging area for research. To enhance state-of-the-art optimization approaches, researchers and practitioners frequently experiment with alternative combinations of algorithms, refine their integration methodologies, and assess their effectiveness on a range of problem domains [2]. The process of merging or integrating two or more distinct algorithms to produce a more effective and efficient optimization strategy is known as the hybridization of heuristic optimization algorithms. It seeks to overcome the shortcomings of each algorithm while utilizing its advantages to get better results and increased performance. Hybridizing two heuristic optimization methods combines their complimentary features and approaches to take advantage of their respective advantages. One algorithm might be better at exploration, for instance, by thoroughly scanning the space which is known as solution space, while another algorithm might be better at the process of exploitation, for example, by locally refining interesting solutions. The hybrid method could benefit from both strategies i.e. exploration and exploitation by combining these two strategies, which will improve performance, solution quality, convergence speed, and robustness while addressing challenging optimization problems [3].

The type of hybridization method that is used will depend on the particular algorithms that are being merged. Enhancing the search process, could entail combining elements or methodologies from two different algorithms or running two algorithms concurrently and sharing information between them at predetermined intervals [4]. The current investigation has taken up consideration of two heuristic optimization algorithms first one is SGO and the other is PPF for hybridization. Naik et al. [5] proposed the PPF in the year 2021, and Satapathy et al. [6] proposed the SGO in the year 2016. With regard to several heuristic optimization techniques, these algorithms performed well in tests using a variety of benchmark functions. Again, they have shown excellent performance while solving many real-life problems like many outstanding algorithms. The NFL theorem [7] encouraged us to improve the PPF and SGO even if their performance is competitive with that of other algorithms. As we can see from the literature, an algorithm functions flawlessly for a particular set of problems, but it always struggles with other sets of problems. Consequently, researchers can propose new algorithms or improve existing techniques by using the NFL theorem. Regarding the enhancement of current algorithms, integrating two or more algorithms is an effective method for achieving algorithm blending by fusing the best features of multiple algorithms. [8] combines “the Cuckoo Search (CS) algorithm along with the Flower Pollination Algorithm (FPA) to improve capabilities like exploitation and exploration. To increase the effectiveness and convergence speed of the search, GWO-BFO combines the Grey Wolf Optimizer (GWO) with Bacterial Foraging Optimisation (BFO) [9]. The Search for Symbiotic Organisms (SOS) and the Firefly Algorithm (FA) (SOS-FA) are combined” by Biswas et al. [10] to enhance the trade-off between exploitation and exploration. ABC-DE [11] combines Differential Evolution (DE) in an Artificial Bee Colony (ABC) to improve global exploration and exploitation capabilities. In [12], HBMO-FPA combines Honey Bee Mating Optimization (HBMO) with Flower Pollination Algorithm (FPA) to enhance the search efficiency. HBMO is inspired by the mating manners of honeybees, and the pollination process of flowers is the foundation of FPA. Deep et al. [13] combine Cultural Algorithms (CAs) with Harmony Search (HS) (CA-HS) to improve solution quality and search efficiency. In [14], GWO-BA combines the Grey Wolf Optimizer (GWO) algorithm with the Bat Algorithm (BA) in order to improve the search efficiency.

The “algorithm for social group optimization solves complicated issues by emulating human social behavior. The acquiring phase and the improving phase are the two phases of the SGO process. In the Improving phase, each person's level of knowledge (solutions) is raised by the influence of the best individual; in the Acquiring phase, interactions with other individuals and the best person who has the highest level of knowledge and the ability to address the issue at hand raise each person's level of knowledge (solutions). To address the short-term hydrothermal scheduling problem and improve the algorithm, Naik et al. proposed the modified SGO (MSGO) algorithm" [15]. Naik suggested a modified form of SGO to solve optimization with multi-objectives of the machining "operations [16]. Naik introduced Chaotic Social Group Optimization (CSGO) to solve structural engineering design problems [17]. Singh et al. [18] used images obtained from chest x-ray to solve the detection problem of COVID-19" Infection by merging SGO and SVC. Kalananda & Komanapalli [19] developed the "Hybrid Social Whale Optimization Algorithm (HS-WOA & HS-WOA+), which combines WOA and SGO to address engineering optimization challenges. The Multiple Objective Social Group Optimization for Time-Cost-Quality-Carbon Dioxide in Generalized Construction Projects has been proposed by Huynh et al. Similarly, there" are several applications in literature where the SGO algorithm performs well in terms of search [21-27].

Using a population-based optimization approach, the PPF algorithm treats the population as a collection of unique members of a community, each representing a potential solution. The quality of their lives indicates their fitness. The best individual globally is called "gbest." The population is split into subgroups depending upon the expertise, with the best in each subgroup termed "SGbest." The best individual across all subgroups, "Leader_gbest," is the final solution. Each individual updates their position based on past experiences, current efforts, and the best individual in their subgroup, sharing information to improve overall performance. Nevertheless, as the study points out, despite the paucity of research on the PPF algorithm, it has proven effective in resolving practical optimization issues. [5].

Since both the SGO and PPF algorithms demonstrate great search performance, this study explores the integration of SGO and PPF with an aim to enhance the SGO algorithm via effective global search PPF. We introduce the Past Preset Future Algorithm (SGO-PPF) based on Social Group Optimization, which enables persons to learn more intelligently through their past experiences and current efforts, in order to solve the complex problems of their lives by acquiring more knowledge. The foundation of this algorithm is the SGO and PPF algorithms' characteristics. This hybrid technology's primary objective is to better balance algorithm exploration and exploitation by combining SGO's robust global search power with PPF's fast convergence capabilities. This capability of the SGO-PPF algorithm has been utilized to optimize the cost estimation of software. Software cost estimates are a crucial procedure since it's necessary to predict accurately the project's cost at the beginning of the project. To calculate the project's budget and resource requirements, cost estimation is necessary [28-29]. The number of times reviews are conducted, implementation efficiency, processing efficiency of pre-development, and other factors all affect how much the project will cost [30]. One technique to estimate software costs is the COCOMO. Based on the project qualities, the model estimates the effort, the cost, and the schedule for software projects using a simple regression technique with a few parameters. In order to maximize the cost estimation of the software, our goal is to optimize those parameters. The simulation section goes into further detail about software cost estimation.

The following summarizes this paper's major contributions:

a. A hybrid form of the meta-heuristic algorithm (SGO-PPF) combining SGO and PPF is proposed.

b. SGO-PPF combines the capability of SGO i.e. global search with PPF's capability i.e. quick convergence.

c. In “order to assess the efficacy of SGO-PPF, the current study measures the search accuracy and statistical performance of the suggested algorithm using 23 benchmark functions.

d. The software's cost estimation is optimized through the use of the SGO-PPF algorithm.

The structure of the remaining section of the document is as follows: A summary of SGO and PPF is given in Section 2. The suggested SGO-PPF method is covered in full in Section 3. The analysis of the results and simulated experiments are done in Section 4. The conclusion, limitations of the research, and suggestions for additional research are included in Section” 5.

2. Related works

2.1 SGO

Inspired by social interaction in humans, the SGO algorithm is made to handle challenging issues. Within a social group, individuals represent potential solutions in this algorithm; each has the knowledge and the capacity to solve the given problem. Each individual's characteristics match the “dimensions of the design variables in the problem. The Improving Phase and the Acquiring Phase are the two stages of the optimization process.

Assume that there are i=1, 2, 3,..., pop_size, members of the social group, P_i. Each of the individual P_i is defined by” a group of traits $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots \dots, p_{iD})$, with D representing the number of traits (dimensions). Each individual is also associated with a fitness value f_i, indicating their “performance.

Phase 1: Improving Phase

During the phase of Improving, 'gbest,' the group's individual who is best, imparts knowledge to the others to help them become better. Each individual updates their position based on the information from 'gbest' as follows:

$$Pnew_i = c * P_i + r * (gbest - P_i) \quad (1)$$

- If the recent solution $Pnew_i$ outperforms the existing solution in terms of fitness, accept it.

- Here, c is the self-introspection parameter, which is set to 0.2, and r is a random number drawn from a uniform” distribution U(0, 1).

Phase 2: Acquiring Phase

In order to learn new things, people in this phase converse with the most knowledgeable members of their group and communicate at random with other participants. The steps are as follows:

For “each of the individual i in the population:

1. Randomly select another individual P_r, where $i \neq r$
2. If f (P_i) < f (P_r):

$$Pnew_i = P_i + r_1 * (P_i - P_r) + r_2 * (\, best_P - P_i)$$

Else :

$$Pnew_i = P_i + r_1 * (P_r - P_i) + r_2 * (best_P - P_i) \qquad (2)$$

- Accept $Pnew_i$ if it” outperforms the existing solution in terms of fitness
- Stochasticity is introduced into the method by using random values from U(0, 1) for r_1, r_2 and other variables 'lb' denote the lower limit and 'ub' denote the upper limit of the design variables.

Please see the cited papers [6] for a more thorough explanation of the SGO algorithm.

2.2 PPF

Like other optimization techniques based on the population, the PPF algorithm is also a population-based optimization method. In this algorithm, the population is considered as individuals in the world, with each individual representing a candidate solution. A person’s life quality indicates their fitness, and the individual with the best life quality is the global best (gbest). Based on areas of skill, the population is divided into subgroups, and the best member of each subgroup is referred to as the subgroup best (SGbest). The overall best individual among all subgroups is called the ‘Leader_gbest’, which is considered the final solution.

Main Framework of the PPF Algorithm

Input:

- N: Population size
- S: Number of subgroups
- M: Subgroup size
- Max_iter : Maximum number of iterations

Output:

- Leader_gbest
- Leader_score at Max_iter

1. Randomly initialize the population ‘P’.
2. Assign ‘presentP’ with ‘P’, ‘pastP’ with ‘P’, and ‘futureP’ with ‘P’.
3. Fitness f(i) for each P(i, :) is evaluated
4. Find [Leader_score, position] =min(f(i), i = 1, 2, 3, ..., N)

5. Set Leader_gbest = P(position, :).

6. Assign ‘presentF’ with ‘f’, ‘pastF’ with ‘f’, and ‘futureF’ with ‘f’

7. Use Algorithm 2 to split population ‘P’ into ‘S’ subgroups and calculate ‘subgroup gbest’, or ‘SGbest’.

8. ‘t’ is initialized with 0

While t <Max_iter:

1. Update pastF with presentF, pastP with presentP.

2. Update presentF with futureF, present with futureP.

3. Use Algorithm 3 to calculate the next changes.

4. Switch subgroups by utilizing Algorithm 4.

5. Incorporate Algorithm 5, and return the futureP population that crosses boundaries.

6. The value of the fitness ‘futureF’ for the population ‘futureP’ is calculated.

7. Utilizing Algorithm 6, update the Leader_score.

8. Split the population ‘futureP’ into ‘S’ subgroups, then use Algorithm 2 to calculate the ‘subgroup gbest’, or ‘SGbest’.

9. Increment the value of t.

Algorithm 2: Division of Population into Subgroups

1. Initialize SGbestInd = zeros(S, 1) and SGbests =zeros(S, D)

2. For i = 1:S :

- SI = (i-1) * M + 1
- EI = SI + M – 1
- Find the index of the minimum fitness within subgroup [~, leadIndex]= min(f(SI:EI).
- Update SGbestInd(i, 1) = (SI - 1) + leadIndex
- Set SGbests(i, :) = P(SGbestInd(i, 1), :) .

Algorithm 3: Future Updating of Population

1. For (a = 1:M):

- For (p = 1:S):
- i = (p-1) * M + a) (index of member)
- For j = 1:D:

\- Set SGD= SGbest(p, j)

\- If pastF (i) >= presentF}(i) (if present work is improved):

Update futureP based on different conditions relative to pastP, presentP, and SGD.

- Else (if present work is deteriorated):

Update futureP based on different conditions relative to pastP, presentP, and SGD.

Algorithm 4: Subgroup Switching of Population

1. For (p = 1:S):

- For (a = 1:M):
- Set (from = (p-1) * M + a).
- Randomly select another subgroup toS $\neq$ p
- identify the least fit member in the selected subgroup.
- Swap positions and fitness values between the form and the least fit member.

Algorithm 5: Updating the Future Position of the Population

1. For (i = 1:N):

- Adjust futureP(i, :) to ensure it remains within the boundaries of lb and ub

Algorithm 6: Updating Leader Score

1. For i = 1:N :

- If futureF(i, 1) < Leader_score):
- Revise Leader_score = futureF(i, 1)
- Set Leader_gbest = futureP(i, :)

Refer to the cited papers for a more thorough explanation of the SGO algorithm. [5].

3. Hybridization of SGO and PPF (SGO-PPF) algorithm

Through a balanced approach to both exploration and exploitation, the SGO-PPF algorithm combines the best features of both the SGO and PPF algorithms to improve search efficiency. The following provides an explanation of the hybridization concept of the SGO-PPF optimization algorithm.

(1) Initialization

The algorithm SGO-PPF begins by randomly assigning persons (individuals) P to different locations throughout the search space, just like other algorithms. This procedure is represented mathematically “as follows:

$$P = \mathrm{lb} + \mathrm{r}\times(\mathrm{ub} - \mathrm{lb}) \qquad (3)$$

Here, 'r' is a random number chosen at random from a uniform distribution between 0 and 1, and 'lb' and 'ub' stand for the lower and upper limits of the search space”, respectively.

(2) SGO-PPF optimization process

The primary search procedure which is iterative consists of three phases that emulate various types of scenarios involving persons, each of which is designed to implement distinct search strategies.

a) The first phase

This phase is similar to the SGO algorithm's improvement stage and happens when

$Iter < 2\frac{Max\_iter}{4}$. This stage requires more early iterations and places an emphasis on exploration.

b) The second phase

This phase, which happens when, matches the PPF algorithm, $2\frac{Max\_iter}{4} < Iter < 3\frac{Max\_iter}{4}$, in the middle of the iteration. In the second phase, the person promotes both exploration and also exploitation.

c) Third phase

This phase, which comes toward the end of the iteration, is similar to the acquiring stage of algorithm SGO when $3\frac{Max\_iter}{4} < iter$. This transition from exploration to exploitation is symbolized by this phase.

(3) The end phase

This is the completion of the algorithm. After completion of all iterations, we get the best solution as best person and value in terms of score as gbest person score.

Algorithm: Pseudo code of SGO-PPF algorithm

Input: population size, subgroup size, number of subgroups, maximum number of iterations, lower bound, upper bound, dim of the problem as N, M, S, Max_ iter, lb, ub, dim respectively.

Output: best fitness value as leader_score, position of best fitness value as “leader_pos

1. Initialize the population of person (P)
2. Define algorithm parameters: C
3. While iter < Max_iter:

    Perform fitness determinations

    Pick the best solution as the gbest.

    If iter < 2*(Max_iter / 4):

        For each person i in the population (pop_size):

Update the person using Equation 1.

        End for.

Else if (2*Max_iter / 54) < iter < (3 * (Max_iter / 4)):

For each person i in the population (pop_size):

Update the person using the PPF algorithm.

End for.

Else:

For each person i in the population (pop_size):

Update the prey using Equation 2.

End for.

End if.

Evaluate the fitness of the person.

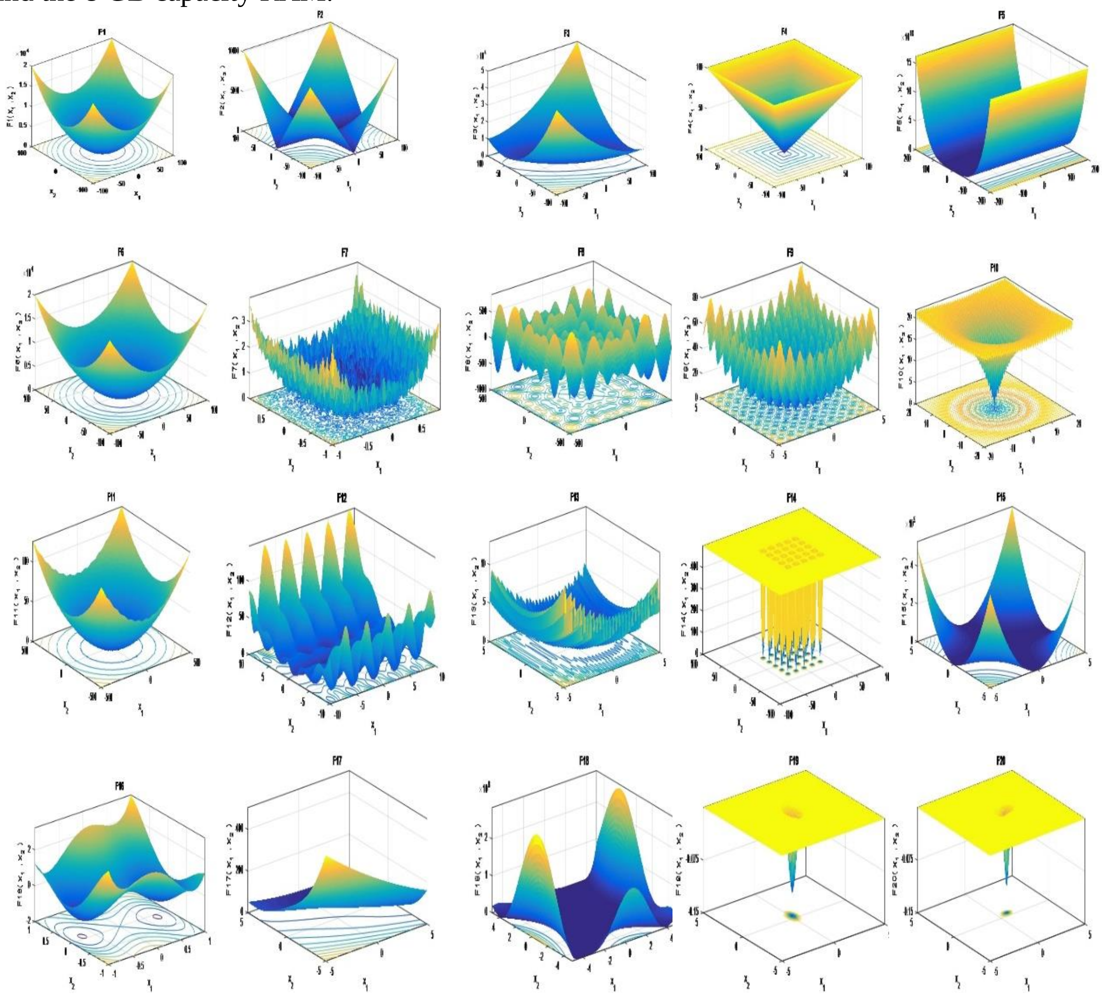
Update the current best" solution.

4. End while.

4. Simulation and experimental results

Current research illustrates efficiency in the SGO-PPF algorithm by conducting two experiments. Experiment 1 involved comparing the performance of the algorithm with thirteen advanced algorithms, including SGO, such as the AVOA (African vultures optimization algorithm) [31], DE [32], EDO (Exponential distribution optimizer)[33], GWO [34], KOA (Kepler optimization algorithm) [35], LSO (Light Spectrum Optimizer) [36], MSA (Mantis Search Algorithm)[37], NOA (Nutcracker optimizer algorithm) [38], RSA (Reptile Search Algorithm) [39], SMA (Slime mold algorithm) [40], SWO (Spider wasp optimizer)[41], and WOA [42] for validation of algorithm. In the 2nd experiment, the algorithm shows its performance in solving software cost estimation problems.

Every novel optimization algorithm must undergo rigorous evaluation using well-defined benchmark functions to assess and validate its performance. Despite the availability of numerous benchmark functions, there is a lack of standardized benchmarks universally accepted for evaluating novel algorithms. To confirm as well as benchmarking efficiency in our newly proposed SGO-PPF algorithm, we conducted simulations using a set of twenty-three benchmark functions. These carefully selected benchmark functions serve as a comprehensive testbed for assessing various aspects of the algorithms, including their ability to achieve rapid convergence, escape local optima, along prevent premature convergence. The selection of these benchmark functions is motivated by their widespread use in existing literature [43-47]. Within the set of twenty-three functions, seven are classified as unimodal benchmark functions (F1–F7), developed to assess an algorithm's potential for exploitation because of its singular global optimum. 6 functions are designated as multimodal benchmark functions, along the remaining ten are categorized as fixed-dimensional multimodal benchmark functions. The multimodal functions (F8–F23) exhibit numerous local optima, making them particularly suitable for assessing the exploration capabilities of algorithms. For a comprehensive understanding of these benchmark functions, detailed descriptions can be found in reference [48], and graphical representations are provided in Fig 1. Experiment 1 uses these benchmark functions to validate the performance comparisons among algorithms. Experiment 2 software cost estimation datasets is used [18] in order to validate the algorithms performance. Implementing all algorithms on the operating system Windows 10, MATLAB 2016a is employed. Simulations are executed on a laptop featuring an i5 Intel Core processor and the 8 GB capacity RAM.



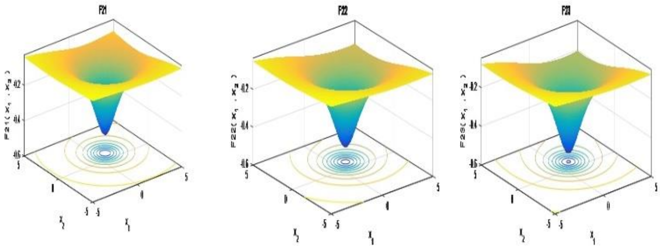


Fig 1. Graphical representation of standard benchmark functions

4.1 Algorithm validation of SGO-PPF

In this experiment, to examine the effectiveness in SGO-PPF, utilize set of 23 benchmark functions and compare the results against thirteen different metaheuristic algorithms, as outlined previously. We keep the population size (pop_size=50) and the maximum number of function fitness assessments (max_FEs)=125,00 constant throughout these experiments. The algorithms' specific parameters adhere to commonly accepted configurations widely employed by researchers, as summarized in Table 1.

Table 1. Parameter setting of algorithms

"Sl. No.	Algorithms	Parameters	Values
1	Social Group Optimization	C	0.2
2	Grey Wolf Optimization	Control parameter	[2, 0]
3	African vultures optimization algorithm (AVOA)	p1	0.6
		p2	0.4
		p3	0.6
		alpha	0.8
		betha	0.2
		gamma	2.5
4	Slime Mould Algorithm(SMA)	Parameter	0.03
5	Exponential distribution optimizer (EDO)	f= 2*rand-1	
		a=f^10	
		b=f^5	
		c=d*f	
6	Differential Evolution[DE]	Scaling Factor (F)	0.5
		Crossover Parameter (Cr)	0.5
7	Kepler optimization algorithm (KOA)	Tc	3
		M0	0.1
		lambda	15
8	Light Spectrum Optimizer (LSO)	Ps	0.05"
		Pe	0.6
		Ph	0.4
		B	0.05
9	Mantis Search Algorithm (MSA)	p	0.5
		A	1.0
		a	0.5
		P	2
		Alp	6
		Pc	0.2
10	Nutcracker optimizer algorithm(NOA)	Alpha	0.05
		Pa2	0.2

11	Reptile Search Algorithm (RSA)	Prb Alpha value	0.2 0.1 0.1
12	Spider wasp optimizer(SWO)	Beta TR Cr	0.1 0.3. 0.2
13	Whale Optimization Algorithm(WOA)	N_min=20 Spiral updating probability Shrinking encircling	20 0.5 0.5
14	SGO-PPF	Random search ability Self-introspection parameter (c) No. of subgroup No of person in each subgroup	0.1 0.2 5 10

Experiment in our study, we evaluate the efficiency of SGO-PPF algorithm by comparing them against other algorithms. To ensure the reliability and statistical significance of our findings, we conducted 30 repetitions of the experiment. The results are presented in Table 2,3,4 and 5, indicating key metrics, including "the best (BEST), average (MEAN), standard deviation (SD) of fitness solutions, the p-values derived from the WRS test [49] at a significance level of 5% for SGO-PPF versus other approaches and h-value. Table 2 provides results for unimodal, Table 3 for multimodal and Table 4, 5 for fixed-dimensional multimodal benchmark functions". The most outstanding results are emphasized in bold in the tables. When p-values fall below 0.05, it indicates the rejection of null hypothesis, whereas "NaN" signifies that the input values are similar. In Table 6, "w" represents that the performance of other approaches is inferior, "b" signifies it is superior, and "*s*" suggests a similar performance when compared to SGO-PPF algorithm. Convergence characteristic of the algorithms is depicted in the Figure 2.

Table 2. Comparison of SGO-PPF and other unimodal benchmark algorithms

Algo/Functions		F1	F2	F3	F4	F5	F6	F7
SGO-PPF	"BEST	2.1895e-213	2.6981E-108	1.4483E-211	1.1322E-109	3.5612E-06	0	3.7419E-06
	AVERAGE	4.2176e-207	1.2243E-104	1.5777E-203	1.5328E-104	6.5480E-01	0	1.1780E-04
	SD	0	2.3822E-104	0	2.2140E-104	1.1819E+00	0	8.2538E-05
SGO	BEST	8.3960e-171	4.4243E-86	1.3948E-170	3.2115E-86	2.5847E+01	3.6233E-05	4.8833E-06
	AVERAGE	1.1472e-170	4.9585E-86	2.0990E-170	3.6262E-86	2.6442E+01	5.6241E-04	8.9597E-05
	SD	0	2.7929E-87	0	1.4318E-87	3.3370E-01	4.9330E-04	6.6135E-05
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	2.3400E-01
	h-value	1	1	1	1	1	1	0
AVOA	BEST	0.0047	2.5987E-04	1.2983E+00	1.8600E-02	3.9309E+01	2.1272E+00	1.1800E-02
	AVERAGE	463.2789	1.2138E+01	9.4965E+03	9.7565E+00	2.7509E+05	1.0362E+03	4.5800E-01
	SD	536.5675	9.8733E+00	1.1179E+04	8.5297E+00	6.6346E+05	2.0790E+03	4.7160E-01
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
DE	BEST	4.3846e+04	1.0668E+02	5.3392E+04	7.4794E+01	9.6493E+07	4.4861E+04	5.9719E+01

	AVERAGE	5.3288e+04	4.9326E+08	8.8587E+04	8.3052E+01	1.7524E+08	5.3044E+04	8.2316E+01
	SD	4.2294e+03	8.0856E+08	1.6943E+04	3.7484E+00	3.0257E+07	3.4963E+03	1.2701E+01
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
EDO	BEST	0	1.9693E-97	0	0	2.8640E+01	2.1770E-01	1.1616E-05
	AVERAGE	3.1523e-121	3.9130E-62	9.5924E-119	3.6819E-61	2.8713E+01	5.1960E-01	2.1175E-04
	SD	1.4571e-120	2.0982E-61	5.2538E-118	1.2604E-60	1.6700E-02	1.5150E-01	1.2923E-04
	p-value	8.4808e-09	3.0199E-11	1.0648E-07	5.5727E-10	3.0199E-11	1.2118E-12	4.0000E-03
	h-value	1	1	1	1	1	1	1
GWO	BEST	6.8762e+03	4.6146E+01	1.0910E+04	3.7591E+01	3.6552E+06	6.2973E+03	2.1488E+00
	AVERAGE	1.1533e+04	8.9841E+01	2.4426E+04	4.5446E+01	1.0157E+07	1.0550E+04	4.7317E+00
	SD	2.5341e+03	8.7968E+01	5.9546E+03	5.3819E+00	3.2131E+06	2.2262E+03	1.8771E+00
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
KOA	BEST	1.0724e+04	4.9196E+01	3.5176E+04	4.8829E+01	1.6166E+07	1.4527E+04	9.3986E+00
	AVERAGE	2.5252e+04	1.3582E+05	6.8731E+04	6.2177E+01	4.8737E+07	2.5013E+04	1.9371E+01
	SD	5.0253e+03	3.7831E+05	1.4639E+04	5.9720E+00	1.7865E+07	5.3598E+03	6.9254E+00
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
LSO	BEST	2.0510e+03	1.7968E-22	1.1129E+04	5.9958E+01	7.9675E+03	8.4896E+03	1.5000E-03
	AVERAGE	3.5078e+04	5.0416E+00	7.4673E+04	7.8269E+01	8.1276E+07	4.6434E+04	3.0000E-02
	SD	1.4910e+04	6.8667E+00	2.6274E+04	5.1246E+00	6.0618E+07	9.2284E+03	2.3100E-02
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
MSA	BEST	1.9012e-82	1.2531E-44	3.3738E-88	1.1862E-41	2.4590E+01	6.0417E-07	6.6064E-05
	AVERAGE	7.4356e-58	5.1745E-29	2.2545E-57	7.6975E-28	2.5715E+01	8.9790E-05	5.2263E-04
	SD	2.7961e-57	1.8476E-28	1.1366E-56	3.1983E-27	3.1420E-01	9.7171E-05	3.5693E-04
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	1.5581E-08
	h-value	1	1	1	1	1	1	1
NOA	BEST	1.5971e-54	1.9978E-21	1.0257E-52	3.2320E-30	2.8875E+01	6.2545E+00	3.2863E-04
	AVERAGE	1.0167	1.4070E-01	3.9000E-03	7.7775E-04	2.8980E+01	7.1413E+00	5.4600E-02
	SD	5.4289	4.0090E-01	1.9400E-02	4.0000E-03	3.3800E-02	4.5170E-01	4.3000E-02
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0123E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
RSA	BEST	0	0	0	0	2.8981E+01	7.2836E+00	1.0623E-04
	AVERAGE	1.7040e-15	3.9062E-132	0	4.3624E-08	2.8997E+01	7.4836E+00	4.3000E-03
	SD	7.4987e-15	2.1395E-131	0	1.0248E-07	5.0000E-03	5.2800E-02	3.8000E-03”
	p-value	1.2384e-09	1.7203E-12	1.2118E-12	4.2841E-06	1.4440E-11	6.1400E-14	7.3803E-10
	h-value	1	1	1	1	1	1	1

SMA	BEST	0	0	0	0	2.8890E+01	3.5928E+00	2.4784E-04
	AVERAGE	5.1206e-07	2.7917E-05	2.1990E-01	2.1887E-04	2.8981E+01	6.5110E+00	8.8000E-03
	SD	1.5068e-06	5.8386E-05	6.6940E-01	4.5054E-04	2.4800E-02	6.9770E-01	5.9000E-03
	p-value	0.0267	1	8.4808E-09	1	3.0199E-11	1.2118E-12	3.6897E-11
	h-value	1	0	1	0	1	1	1
SWO	BEST	0.1938	2.5890E-01	7.3770E-01	4.4910E-01	8.6685E+01	7.1143E+00	4.7000E-03
	AVERAGE	673.8667	8.0623E+00	2.5678E+03	1.3848E+01	4.9369E+04	5.9992E+02	2.5760E-01
	SD	907.1299	8.7073E+00	3.1902E+03	1.0773E+01	1.0822E+05	1.0481E+03	1.9450E-01
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1
WOA	BEST	2.2817e+03	8.0088E+00	5.6609E+04	4.1727E+01	5.4234E+04	3.8455E+02	1.6100E-02
	AVERAGE	4.0455e+03	1.8223E+01	1.1858E+05	7.5431E+01	3.4619E+06	3.3078E+03	2.6046E+00
	SD	1.2460e+03	8.1019E+00	3.5214E+04	1.4253E+01	4.1001E+06	2.0584E+03	2.8040E+00
	p-value	3.0199e-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2118E-12	3.0199E-11
	h-value	1	1	1	1	1	1	1

Table 3. Comparing SGO-PPF and other multimodal benchmark functions algorithms

Algo/Functions		F8	F9	F10	F11	F12
SGO-PPF	BEST	-1.2569E+04	0	8.8818E-16	0	1.5705E-32
	AVERAGE	-1.2569E+04	0	8.8818E-16	0	1.5705E-32
	SD	1.4774E+00	0	0	0	5.5674E-48
SGO	BEST	-1.0453E+04	0	8.8818E-16	0	1.8488E-06
	AVERAGE	-7.7292E+03	0	8.8818E-16	0	3.5713E-05
	SD	1.5628E+03	0	0	0	3.6387E-05
	p-value	1.9558E-11	NaN	NaN	NaN	1.2118E-12
	h-value	1	0	0	0	1
AVOA	BEST	-9.5475E+03	2.8943E-05	2.6000E-02	1.1890E-01	8.4300E-02
	AVERAGE	-5.5064E+03	9.8094E+01	4.5403E+00	1.3456E+01	4.9748E+03
	SD	1.7995E+03	7.3003E+01	3.1421E+00	2.1594E+01	1.9278E+04
	p-value	1.9545E-11	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1
DE	BEST	-4.4699E+03	3.2105E+02	1.8966E+01	3.6005E+02	2.2677E+08
	AVERAGE	-3.4294E+03	3.9201E+02	2.0227E+01	4.8888E+02	3.9030E+08
	SD	4.3314E+02	1.8312E+01	3.2140E-01	5.3581E+01	9.1857E+07
	p-value	1.9558E-11	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1
EDO	BEST	-1.2569E+04	0	8.8818E-16	0	1.1500E-02
	AVERAGE	-1.2528E+04	0	8.8818E-16	0	3.6100E-02
	SD	5.1673E+01	0	0	0	1.1700E-02
	p-value	1.0783E-10	NaN	NaN	NaN	1.2118E-12
	h-value	1	0	0	0	1
GWO	BEST	-3.5577E+03	2.2069E+02	1.4390E+01	6.1458E+01	2.7769E+05
	AVERAGE	-2.7946E+03	2.6296E+02	1.5921E+01	9.7940E+01	2.6625E+06
	SD	3.9928E+02	2.1610E+01	8.5270E-01	1.8279E+01	1.9227E+06
	p-value	1.9558E-11	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1
KOA	BEST	-4.7471E+03	2.7689E+02	1.8050E+01	1.6532E+02	5.2396E+06
	AVERAGE	-3.2517E+03	3.2690E+02	1.9405E+01	2.2538E+02	5.7377E+07

	SD	5.1349E+02	2.2092E+01	5.4660E-01	3.5980E+01	3.5302E+07
	p-value	1.9558E-11	1.2118E-12	1.0583E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1
LSO	BEST	-5.5339E+03	9.8908E-12	4.0994E+00	2.7724E+02	2.7821E+00
	AVERAGE	-3.4779E+03	8.2478E+01	1.8762E+01	4.0297E+02	1.6583E+08
	SD	1.0787E+03	8.7195E+01	3.4166E+00	6.2688E+01	1.2484E+08
	p-value	1.9305E-11	1.2118E-12	1.0113E-12	1.2118E-12	1.2118E-12”
	h-value	1	1	1	1	1
MSA	BEST	-1.2569E+04	0	8.8818E-16	0	1.0758E-10
	AVERAGE	-1.0698E+04	0	8.8818E-16	0	3.1227E-09
	SD	1.4551E+03	0	0	0	2.9400E-09
	p-value	2.1384E-10	NaN	NaN	NaN	1.2118E-12
	h-value	1	0	0	0	1
NOA	BEST	-5.4968E+03	0	8.8818E-16	0	6.6970E-01
	AVERAGE	-4.1714E+03	3.5800E-02	9.1000E-03	3.7300E-02	1.3050E+00
	SD	6.3636E+02	1.4450E-01	3.0900E-02	1.8840E-01	3.1650E-01
	p-value	1.9558E-11	3.1349E-04	1.9332E-10	1.2711E-05	1.2118E-12
	h-value	1	1	1	1	1
RSA	BEST	-4.5318E+03	0	8.8818E-16	0	1.5082E+00
	AVERAGE	-3.5065E+03	0	8.8818E-16	0	1.6523E+00
	SD	5.9316E+02	0	0	0	3.6200E-02
	p-value	1.9545E-11	NaN	NaN	NaN	2.0744E-13
	h-value	1	0	0	0	1
SMA	BEST	-1.2569E+04	0	8.8818E-16	0	5.3000E-03
	AVERAGE	-9.8883E+03	5.5970E-01	2.5106E-05	3.4249E-08	9.7720E-01
	SD	2.2424E+03	2.9745E+00	3.5339E-05	7.4177E-08	3.7930E-01
	p-value	5.3799E-11	1.3056E-07	2.2130E-06	2.9343E-05	1.2118E-12
	h-value	1	1	1	1	1
SWO	BEST	-4.4844E+03	1.3300E-02	1.0500E-02	2.6900E-02	4.4750E-01
	AVERAGE	-3.5890E+03	1.0910E+02	5.3697E+00	6.3370E+00	5.0743E+03
	SD	4.5025E+02	8.1340E+01	4.4367E+00	7.5050E+00	2.0978E+04
	p-value	1.9558E-11	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1
WOA	BEST	-9.1488E+03	1.2907E+02	3.4910E-01	2.4566E+00	1.4922E+01
	AVERAGE	-8.2375E+03	2.4755E+02	1.1153E+01	4.1662E+01	1.7700E+06
	SD	6.9719E+02	5.1151E+01	3.5851E+00	2.2677E+01	2.9715E+06
	p-value	1.9558E-11	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
	h-value	1	1	1	1	1

Table 4. Comparing SGO-PPF and other fixed-dimensional multimodal benchmark function algorithms (F13-F18)

“Algo/Functions		F13	F14	F15	F16	F17	F18
SGO-PPF	BEST	3.9282E-06	9.9800E-01	3.0815E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	AVERAGE	8.4000E-03	9.9800E-01	3.3426E-04	-1.0316E+00	3.9790E-01	3.0003E+00
	SD	1.0300E-02	2.6412E-16	1.7120E-05	3.8234E-16	0	3.4392E-10
SGO	BEST	8.9423E-05	9.9800E-01	3.0749E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	AVERAGE	8.2000E-03	9.9800E-01	3.1522E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	SD	1.8300E-02	1.3039E-16	1.7042E-05	6.7752E-16	0	1.5003E-15
	p-value	1.1540E-01	2.7000E-03	3.3242E-06	NaN	NaN	2.4213E-11
	h-value	0	0	1	0	0	1”

AVOA	BEST	1.2423E+00	9.9800E-01	8.2109E-04	-1.0316E+00	3.9790E-01	3.0003E+00
	AVERAGE	4.1791E+04	7.2145E+00	5.2000E-03	-1.0300E+00	4.0120E-01	3.1442E+00
	SD	1.7231E+05	5.5392E+00	3.5000E-03	2.4000E-03	4.8000E-03	3.5720E-01
	“p-value	3.0199E-11	2.3982E-11	3.0199E-11	6.6955E-11	3.0199E-11	1.0937E-10
	h-value	1	1	1	1	1	1
DE	BEST	3.4029E+08	9.9930E-01	1.7000E-03	-1.0298E+00	4.0040E-01	3.0151E+00
	AVERAGE	7.5580E+08	5.2963E+00	1.8100E-02	-9.6540E-01	5.2290E-01	6.4280E+00
	SD	1.4358E+08	2.7974E+00	1.0100E-02	4.4200E-02	9.9000E-02	3.4697E+00
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1	1
EDO	BEST	1.2860E-01	9.9800E-01	4.9553E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	AVERAGE	2.5690E-01	9.9800E-01	7.1461E-04	-1.0316E+00	3.9790E-01	3.0000E+00 ”
	SD	6.6800E-02	1.8439E-06	1.0340E-04	2.7447E-08	3.8982E-06	1.2787E-07
	p-value	3.0199E-11	1.0840E-04	3.0199E-11	7.7725E-09	3.2000E-03	3.0199E-11
	“h-value	1	1	1	1	1	1
GWO	BEST	1.4275E+06	9.9800E-01	5.8121E-04	-1.0316E+00	3.9800E-01	3.0005E+00
	AVERAGE	2.2866E+07	9.4988E+00	8.1000E-03	-1.0289E+00	4.2840E-01	3.0497E+00
	SD	1.0852E+07	4.9816E+00	6.2000E-03	3.9000E-03	3.9600E-02	5.1900E-02
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	7.3891E-11
	h-value	1	1	1	1	1	1
KOA	BEST	3.9656E+07	1.0021E+00	6.6115E-04	-1.0287E+00	4.0000E-01	3.1628E+00
	AVERAGE	1.6270E+08	8.0481E+00	2.1900E-02	-8.6250E-01	5.2260E-01	6.6078E+00
	SD	5.4658E+07	4.5173E+00	1.2300E-02	1.4340E-01	1.1870E-01	2.9210E+00
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1	1
LSO	BEST	3.3173E+00	1.0289E+00	3.2000E-03	-1.0301E+00	4.1900E-01	3.0007E+00
	AVERAGE	2.9691E+08	7.2959E+00	1.7400E-02	-9.9120E-01	5.8620E-01	6.2798E+00
	SD	2.2515E+08	4.3961E+00	1.0300E-02	4.0200E-02	1.3630E-01	3.8154E+00
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	5.4941E-11
	h-value	1	1	1	1	1	1
MSA	BEST	5.8343E-09	9.9800E-01	3.0749E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	AVERAGE	3.4476E-06	9.9800E-01	3.0759E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	SD	8.3901E-06	1.4867E-16	1.4762E-07	5.5319E-16	0	1.6369E-15
	p-value	5.4941E-11	9.2000E-03	3.0199E-11	1.3369E-11	NaN	2.6706E-11
	h-value	1	1	1	1	1	1”
NOA	BEST	2.1661E+00	1.1495E+00	1.1000E-03	-1.0302E+00	4.0480E-01	3.0014E+00
	AVERAGE	3.0025E+00	6.6538E+00	1.1300E-02	-9.8610E-01	5.9560E-01	5.7636E+00
	SD	2.0510E-01	3.6573E+00	8.9000E-03	4.0300E-02	1.8220E-01	2.6606E+00
	p-value	3.0180E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1	1
RSA	BEST	2.9651E+00	2.5182E+00	4.6376E-04	-1.0314E+00	4.0080E-01	3.0220E+00
	AVERAGE	2.9987E+00	1.0062E+01	3.8000E-03	-1.0106E+00	5.9260E-01	1.6475E+01
	SD	6.4000E-03	3.3389E+00	2.5000E-03	1.4800E-02	2.3470E-01	1.2411E+01
	p-value	7.8787E-12	2.3829E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1	1
SMA	BEST	4.2100E-02	9.9800E-01	4.8067E-04	-1.0316E+00	3.9790E-01	3.0000E+00
	AVERAGE	2.4699E+00	5.1410E+00	4.3000E-03	-1.0309E+00	4.0040E-01	5.7267E+00
	SD	9.8600E-01	4.4081E+00	3.4000E-03	1.5000E-03	3.3000E-03	8.2312E+00
	p-value	3.3384E-11	4.5684E-09	3.0199E-11	8.9934E-11	3.0199E-11	4.4592E-04
	h-value	1	1	1	1	1	1
SWO	BEST	3.0495E+00	1.8431E+00	2.1000E-03	-1.0307E+00	3.9870E-01	3.0106E+00
	AVERAGE	3.4211E+04	6.3557E+00	1.2400E-02	-9.5520E-01	4.9400E-01	5.0448E+00

	SD	9.2960E+04	3.4291E+00	6.8000E-03	7.4400E-02	9.2200E-02	3.2043E+00
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1	1
WOA	BEST	3.7968E+04	9.9800E-01	4.0586E-04	-1.0316E+00	3.9820E-01	3.0000E+00
	AVERAGE	1.6924E+07	9.2478E+00	9.1000E-03	-1.0277E+00	5.6000E-01	1.0330E+01
	SD	1.8635E+07	5.6008E+00	8.8000E-03	5.7000E-03	2.3220E-01	1.2078E+01
	p-value	3.0199E-11	2.3982E-11	3.0199E-11	3.4742E-10	3.0199E-11	2.8314E-08
	h-value	1	1	1	1	1	1

Table 5. Comparing SGO-PPF and other fixed-dimensional multimodal benchmark function algorithms (F19-F23)

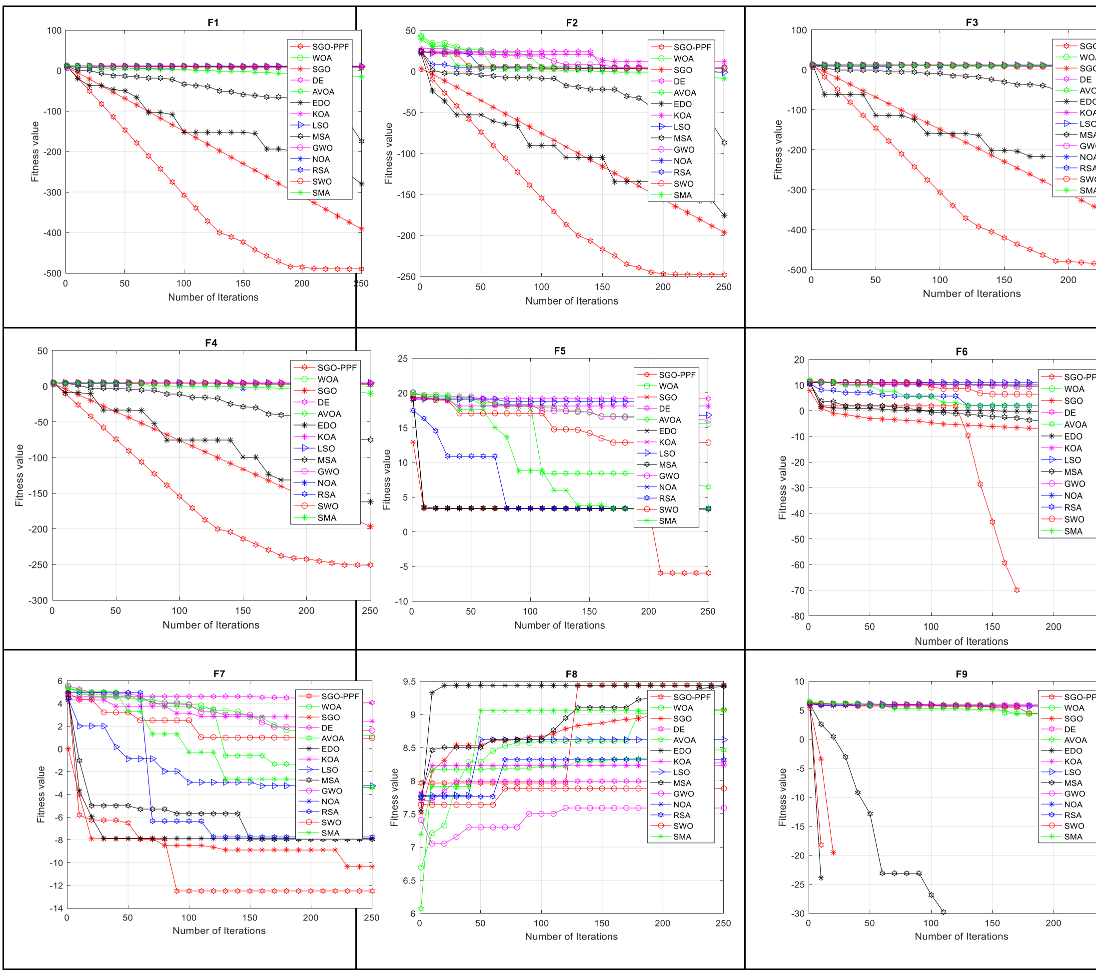
“Algo/Functions		F19	F20	F21	F22	F23
SGO-PPF	BEST	-3.8628E+00	-3.3220E+00	-1.0153E+01	-1.0403E+01	-1.0536E+01
	AVERAGE	-3.8572E+00	-3.3220E+00	-1.0153E+01	-1.0403E+01	-1.0530E+01
	SD	1.1400E-16	6.3500E-12	5.2000E-13	1.5300E-12	1.2100E-12
SGO	BEST	-3.8628E+00	-3.3220E+00	-5.0552E+00	-1.0403E+01	-5.1285E+00”
	AVERAGE	-3.8628E+00	-3.3022E+00	-5.0552E+00	-5.2648E+00	-5.1285E+00
	SD	2.7101E-15	4.5100E-02	9.0336E-16	9.7040E-01	4.5826E-15
	p-value	NaN	1.0351E-07	1.2118E-12	1.4345E-10	2.3638E-12
	h-value	1	1	1	1	1
AVOA	BEST	-3.8624E+00	-3.2614E+00	-9.1239E+00	-8.5204E+00	-9.3773E+00
	AVERAGE	-3.8453E+00	-3.0334E+00	-4.1794E+00	-4.3441E+00	-4.4665E+00
	SD	1.4500E-02	1.3470E-01	2.0207E+00	1.8547E+00	2.3203E+00
	p-value	8.8829E-06	2.1947E-08	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
DE	BEST	-3.8594E+00	-3.0960E+00	-2.3871E+00	-4.3313E+00	-3.1548E+00
	AVERAGE	-3.8125E+00	-2.6362E+00	-1.3365E+00	-1.7645E+00	-1.6642E+00
	SD	3.0000E-02	2.0470E-01	4.2810E-01	8.1170E-01	4.2700E-01
	p-value	3.4971E-09	3.6897E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
EDO	BEST	-3.8628E+00	-3.2985E+00	-1.0140E+01	-1.0369E+01	-1.0402E+01
	AVERAGE	-3.8628E+00	-3.2122E+00	-1.0018E+01	-1.0119E+01	-1.0000E+01
	SD	2.8885E-06	2.7100E-02	9.3400E-02	1.5380E-01	2.8840E-01
	p-value	3.0199E-11	3.9530E-01	4.5043E-11	4.5043E-11	3.0199E-11
	h-value	1	0	1	1	1
GWO	BEST	-3.8598E+00	-3.3085E+00	-8.9322E+00	-9.6624E+00	-8.5770E+00
	AVERAGE	-3.8380E+00	-3.1698E+00	-4.0831E+00	-4.1487E+00	-4.0754E+00
	SD	1.6100E-02	8.9500E-02	2.4706E+00	2.4131E+00	2.1228E+00
	p-value	1.0666E-07	5.5500E-02	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	0	1	1	1
KOA	BEST	-3.8491E+00	-3.1131E+00	-2.2411E+00	-2.3281E+00	-3.4595E+00
	AVERAGE	-3.7628E+00	-2.7200E+00	-1.1707E+00	-1.4722E+00	-1.6716E+00
	SD	6.0000E-02	2.2110E-01	3.9080E-01	3.7480E-01	4.4920E-01
	p-value	2.6099E-10	4.5043E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
LSO	BEST	-3.8571E+00	-2.8757E+00	-3.9940E+00	-2.3357E+00	-3.2322E+00
	AVERAGE	-3.6926E+00	-2.5227E+00	-1.1215E+00	-1.3005E+00	-1.4150E+00
	SD	1.0430E-01	2.0250E-01	7.3710E-01	4.3190E-01	4.7400E-01
	p-value	6.0658E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
MSA	BEST	-3.8628E+00	-3.3220E+00	-1.0153E+01	-1.0403E+01	-1.0536E+01
	AVERAGE	-3.8628E+00	-3.3220E+00	-1.0153E+01	-1.0403E+01	-1.0536E+01
	SD	2.5538E-15	7.3758E-12	2.7114E-10	3.3878E-12	2.5187E-11

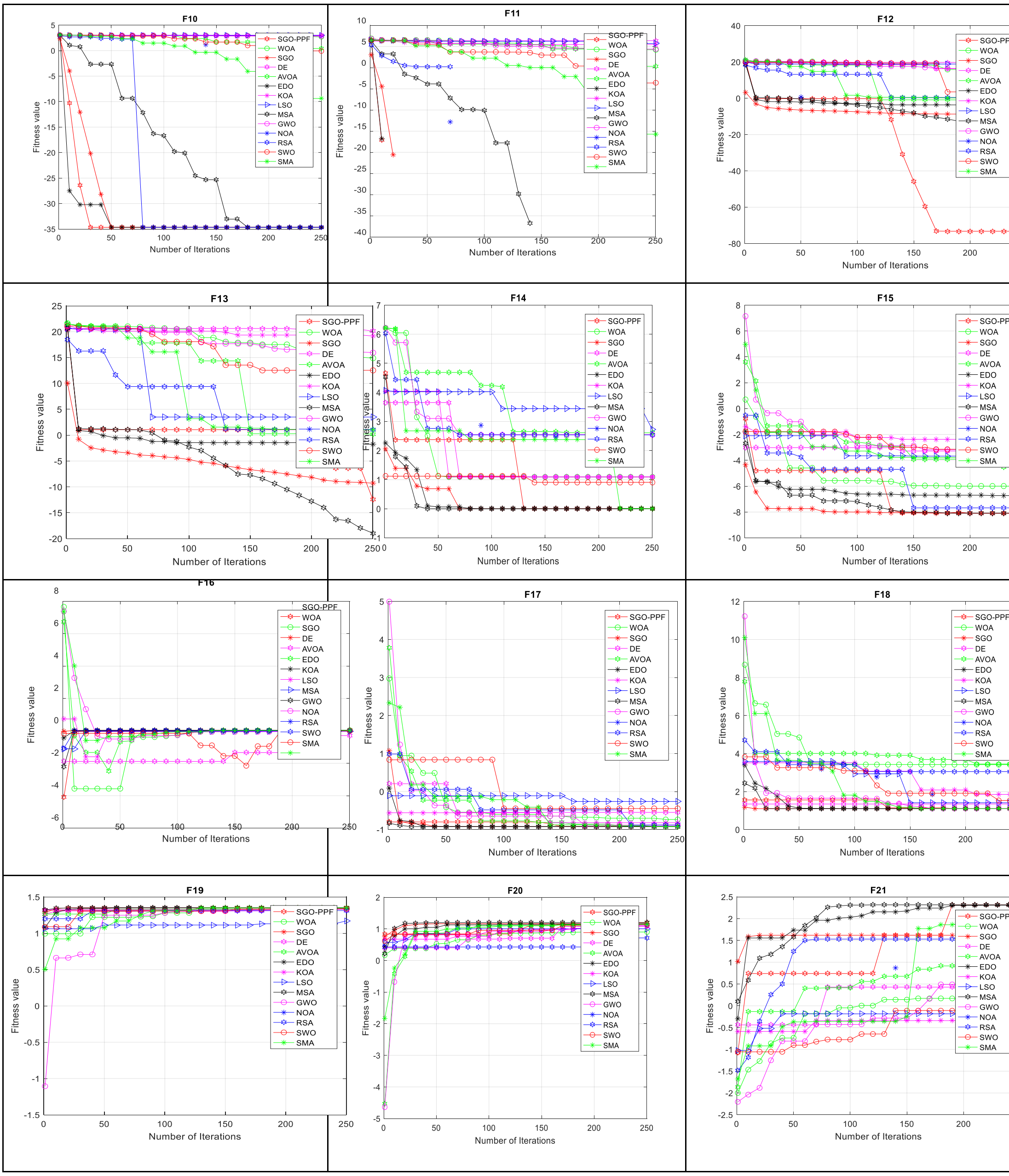
	p-value	NaN	3.0047E-01	NaN	NaN	NaN
	h-value	0	0	0	0	0
NOA	BEST	-3.8620E+00	-3.0196E+00	-8.1733E+00	-5.0500E+00	-7.5226E+00
	AVERAGE	-3.8116E+00	-2.6702E+00	-3.0556E+00	-3.0608E+00	-3.2386E+00
	SD	3.6900E-02	2.0330E-01	1.4702E+00	1.1151E+00	1.4691E+00
	p-value	1.2023E-08	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
RSA	BEST	-3.8252E+00	-2.8785E+00	-5.0551E+00	-5.0876E+00	-5.1280E+00
	AVERAGE	-3.7080E+00	-1.9740E+00	-4.5162E+00	-4.7838E+00	-4.1630E+00
	SD	6.4200E-02	4.3700E-01	1.1252E+00	8.6990E-01	1.4602E+00
	p-value	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
SMA	BEST	-3.8627E+00	-3.3059E+00	-1.0148E+01	-1.0383E+01	-1.0408E+01
	AVERAGE	-3.8577E+00	-3.1059E+00	-5.8686E+00	-7.3132E+00	-6.3334E+00
	SD	4.5000E-03	1.2850E-01	2.5878E+00	2.8570E+00	2.4130E+00
	p-value	2.6000E-03	3.7704E-04	1.2057E-10	4.9752E-11	3.0199E-11
	h-value	1	1	1	1	1
SWO	BEST	-3.8627E+00	-3.1760E+00	-6.8672E+00	-4.8012E+00	-8.2518E+00
	AVERAGE	-3.8179E+00	-2.8055E+00	-2.4620E+00	-2.3507E+00	-2.5917E+00
	SD	3.4000E-02	2.2940E-01	1.4864E+00	7.7140E-01	1.5717E+00
	p-value	7.0881E-08	1.0937E-10	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1
WOA	BEST	-3.8596E+00	-3.2211E+00	-9.6700E+00	-8.1224E+00	-1.0088E+01
	AVERAGE	-3.7785E+00	-2.7035E+00	-4.3206E+00	-3.6399E+00	-3.8439E+00
	SD	7.1900E-02	3.1190E-01	1.9452E+00	1.6404E+00	2.0211E+00
	p-value	2.4386E-09	9.7555E-10	3.0199E-11	3.0199E-11	3.0199E-11
	h-value	1	1	1	1	1

Table 6. Tables 2-5 Wilcoxon rank-sum test p-values

Algorithms	F1	F2	F3	F4	F5	F6	F7	F8
SGO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
AVOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘s’	‘w’
DE	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
EDO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
GWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
KOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
LSO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
MSA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
NOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
RSA	‘w’	‘b’	‘b’	‘w’	‘w’	‘w’	‘w’	‘w’
SMA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
SWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
WOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
Algorithms	F9	F10	F11	F12	F13	F14	F15	F16
SGO	‘s’	‘s’	‘s’	‘w’	‘w’	‘s’	‘b’	‘s’
AVOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
DE	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
EDO	‘s’	‘s’	‘s’	‘w’	‘w’	‘w’	‘w’	‘w’
GWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
KOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
LSO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
MSA	‘w’	‘s’	‘s’	‘w’	‘b’	‘s’	‘w’	‘w’
NOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’

RSA	‘w’	‘s’	‘s’	‘w’	‘w’	‘w’	‘w’	‘w’
SMA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
SWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
WOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’
Algorithms	F17	F18	F19	F20	F21	F22	F23	
SGO	‘s’	‘b’	‘s’	‘w’	‘w’	‘w’	‘w’	
AVOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
DE	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
EDO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
GWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
KOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
LSO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
MSA	‘s’	‘b’	‘s’	‘s’	‘s’	‘s’	‘s’	
NOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
RSA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
SMA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
SWO	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	
WOA	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	‘w’	




F10
F11
F12
F13
F14
F15
F16
F17
F18
F19
F20
F21
Fitness value
Number of Iterations
SGO-PPF
WOA
SGO
DE
AVOA
EDO
KOA
LSO
MSA
GWO
NOA
RSA
SWO
SMA

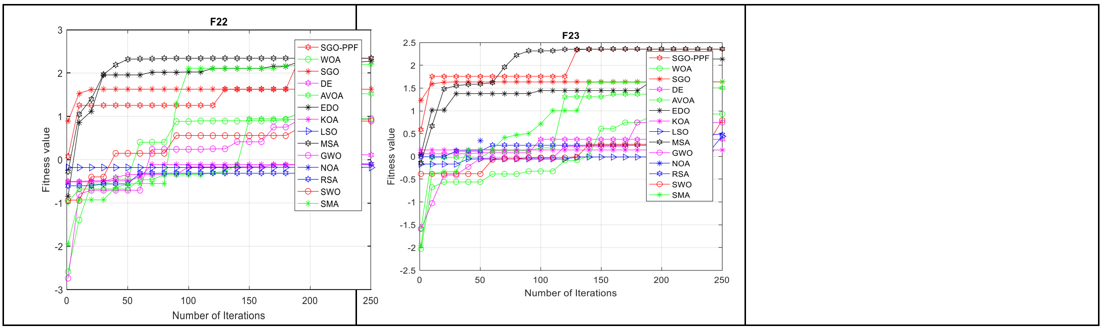


Fig 2. Graphical representation of convergence characteristics of algorithms

Discussion

The “objective of this study was to examine SGO-PPF's ability to explore, exploit, and escape local minima while navigating a variety of benchmark functions, both unimodal and multimodal.

Unimodal functions can assess an algorithm's potential for exploitation and have a single global optimum. The outcome of SGO-PPF & other algorithms on unimodal test functions (F1–F7) are displayed in Table” 2, and they reveal that SGO-PPF performs better than most approaches across all test functions. These findings demonstrate SGO-PPF 's exploitation capability, which enables it to quickly and precisely converge to the optimal state. This skill originates from the defined acquiring phase of SGO and by PPF algorithm.

The count of local optima in multimodal test functions grows exponentially with dimensionality. Assessing an algorithm's exploration capability is advantageous with multiple optimum states. The multimodal benchmarks F8–F13 are high-dimensional, whereas the multimodal benchmarks F14–F23 are fixed-dimensional. Tables 3,4 and 5 demonstrate that SGO-PPF has an exceptional exploratory capacity in comparison to other techniques. SGO-PPF outperforms all algorithms on multimodal functions and finds an optimal solution in thirteen out of sixteen cases; its results are comparable to those of high-performance optimizers. The acquiring phase of the SGO may serve as the basis for the exploitation of SGO-PPF.

The SGO-PPF algorithm is based on the WRS test results shown in Table 6:

- It performs worse than SGO for F15, F18, MSA for F13, F18, RSA for F2, F3.
- SGO for F9, F10, F11, F14, F16, F17, and F19, AVOA for F7, EDO for F9, F10, and F11, and MSA for F10, F11, F14, F17, F19, F20, F21, F22, and F23 all perform similarly to it.
- SGO-PPF outperforms other algorithms in all other functions.
- It consistently surpasses DE, GWO, KOA, LSO, NOA, SMA, SWO, WOA across all twenty-three functions.

As shown in Table 6, the SGO-PPF algorithm outperforms SGO, AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in 14, 22, 23, 20, 23, 23, 23, 12, 23, 21, 23, 23, 23 cases out of 23 cases, respectively. Conversely, the CoSGO algorithm performs worse than SGO, AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0 cases out of 23 cases, respectively. Additionally, SGO-PPF exhibits equivalent results with SGO in 7 cases, with AVOA in 1 case, with EDO in 3 cases, with MSA in 9 cases. In summary, out of 299 examples, SGO-PPF demonstrates equivalent results in 20, worse solutions in 6, and superior outcomes relative to other scenarios in 273 cases.

Overall, SGO-PPF exhibits outstanding performance in addressing both unimodal and multimodal functions.

4.2 Software cost estimation using SGO-PPF

4.2.1 COCOMO

For estimating project costs or effort, this algorithmic method is used. COCOMO calculates the development effort by multiplying the software cost variables by the software size (measured in KDSI). Boehm [50,51] first suggested the model in 1981, and it was calibrated using data from about 63 NASA-developed experiments. The project's development mode—organic, semi-detached, or embedded—was taken into account while estimating the development effort. The effort and project size have a non-linear relationship, according to the COCOMO Model.

Estimated effort (MM) $= a * (KDSI)^b$ ……………………………………..(4)

where, “the project's development mode determines the values of the constants {a, b}, and the estimated effort, which was given in man months (MM). KDSI is the project size, measured in numerous delivered source instructions. Table 7 lists the values of a and b for embedded, semi-detached, along with organic” projects.

Table 7. COCOMO Models

Model	a	b
Organic	3.2	1.05
Semi-Detached	3	1.12
Embedded	2.8	1.2

The COCOMO Model has identified fifteen cost drivers that could affect any software project's anticipated “effort. Each of these cost drivers was assigned a weight that could be multiplied by the expected effort, based on its grade (very low, low, nominal, high, very high, extremely high).

4.2.2 COCOMO II

Barry Boehm presented COCOMO II in 2000 as a model that provided more precise details regarding improvements in some cost factors. Several software attributes are included in the COCOMO II, which are employed in COCOMO II Architecture Post Model. These attributes include 17 EM, 5 Scale Factors (SF), Software Size (in KLOC), and the estimated” effort. There are 5 SFs and 4 categories for multiplier attempts.

Estimated effort (PM)= $a * (SIZE)^E * \prod_{I=1}^{17} EM_i$ (5)

Equation (5) displays the formulas employed in the COCOMO II model to compute software development efforts. The multiplication constant "a," which assesses effort based on a certain project condition, has a value of 2.94. The term "size" refers to the anticipated magnitude of software, typically measured “in Kilo Source Lines of Code (KSLOC). The scale expansion for effort is denoted as E. The Efforts Multiplier is represented by 〖EM〗_i, where i ranges from 1 to 17. The exponential factor accounts for the proportional magnitude of economies and diseconomies in offsetting” the increasing size of software projects. Equation (6) is used to get the scale factor and determine the coefficient of E.

$$E = b + 0.01 * \sum_{j=1}^{5} SF_j \quad (6)$$

where, SF_j is a Scale Factor as well as b is an exponential constant with a value of 0.91 for j = 1, 2, 3, 4, or 5.

4.2.3 Experiment 1

In this experiment, the SGO-PPF “algorithm is utilized to estimate the parameters of the COCOMO model, relying on software cost estimates. The estimated parameters will greatly simplify the computation of developed effort for all types of projects (organic, semidetached, and embedded). We used the SGO-PPF algorithm to estimate the COCOMO model's parameters, as stated in Eq. (4). The parameters listed in Table 8 have been utilized to control the SGO-PPF algorithm's evolutionary process. The COCOMO81 software project dataset, which comprises 63 projects, was used to assess the performance of the created model.

Table 8. Parameters of Software cost estimation based SGO-PPF Algorithm

Sl. No	Parameter Name	Description	Value
1	Pop	Size of the population	50
2	‘a’	Domain for ‘a’	-5 to 5
3	‘b’	Domain for ‘b’	-5 to 5
4	Max_iter	Maximum number of iteration	100
5	Dim	Dimension	2
6	‘c’	Self-introspection parameter	0.2
7	S	Search space for given application	a, b

Evaluation criteria

The actual costs incurred during project design in real-world environmental conditions are reflected in the evaluation criteria used” to identify the accuracy of cost estimation for the software in the proposed model. Appendix contains the actual effort and KDSI for each project (organic, semidetached, and embedded). The evaluation is carried out using the dataset that was made available for the software that was originally built. The MMRE evaluation criteria, which is listed below, has been the subject of this discussion.

1. MRE(Magnitude of Relative Error)

Magnitude of relative error defined as

MRE= $\frac{|Actual_{effort} - Estimate_{effort}|}{Actual_{effort}}$ -------- (7)

2. MMRE(Mean Magnitude of Relative Error)

The relative error mean % of an entire data set is known as the mean absolute value of relative errors, or MMRE. It can be computed using the equation that follows, which is explained below:

$$\text{MMRE}=\frac{1}{N}\sum_{i=1}^{N}\frac{|Actual_{effort}-Estimate_{effort}|}{Actual_{effort}} \quad \text{-------------------- (8)}$$

where $Actual_{effort}$ denotes Actual Effort and $Estimate_{effort}$ denotes Estimate Effort.

Experimental results and analysis

On the basis of the constructed SGO-PPF based model, tests have been carried out on the COCOMO81 dataset to investigate its strengthening. In this case, Table 9 provides the parametric value for a and b. This suggested model is contrasted with other COCOMO algorithms that are currently in use, such as Homeostasis Adaption Based Differential Evolution (HABDE)[56], Differential Evolution-Based Model (DEBM)[54], GA [52], PSO [53], Hybrid Algorithm (Hybrid Algo) [53], and DE [55]. The outcome discussion for the embedded, semidetached, and organic COCOMO81 software projects is as follows:

A. Comparison Effort of algorithms based on COCOMO: First, a comparison is made between the computed estimated effort from Eq. (4) and the COCOMO81 real dataset. It is evident from the comparison of efforts in Tables 10–12 that the suggested model outperforms the other soft computing methods. The following are the final thoughts:

- Table 10 displays the effort outcomes for seven algorithms used in organic projects. In the majority of organic projects, SGO-PPF has produced better effort values than algorithms such as GA, PSO, DE, DEBM, Hybrid Algo, SGO, and HABDE. In the majority of the semi-detached projects, it has also received higher effort values, as seen in Table 11. Table 12 demonstrates that it has attained higher effort values in most embedded projects. This suggests that, following the proposed model, the effort values (measured in person/months) obtained from SGO-PPF exhibit greater diversity and convergence rate.

Table 9. SGO-PPF Models

Model	a	b
Organic	2.075533609323931	1.070495414522126
Semi-Detached	3.086017314268252	1.152959324897693
Embedded	2.054594568041332	0.940275386580845

Table 10. COCOMO-based algorithm comparison on organic projects

Sl.No	GA	PSO	DE	DEBM	Hybrid Algo	HABDE	SGO	SGO-PPF
1	0.419895045	0.372687909	0.298150327	0.303098376	0.292245865	0.229036056	0.23756654	0.237524381
2	0.173631032	0.154110384	0.123288307	0.129350203	0.126701085	0.100082937	0.10213750	0.10212834
3	0.039510482	0.035068475	0.02805478	0.031032256	0.031210965	0.024982	0.02480981	0.02481124
4	0.008363829	0.007423517	0.005938813	0.006943662	0.007179986	0.005827258	0.00562411	0.00562530
5	0.065131451	0.057808981	0.046247184	0.050250336	0.050090621	0.039915193	0.04000631	0.04000662
6	0.021777155	0.019328836	0.015463069	0.017471969	0.050090621	0.039915193	0.01403853	0.01404017
7	0.036755316	0.032623061	0.026098449	0.028942919	0.029147193	0.023345171	0.02315345	0.02315495
8	0.173631032	0.154110384	0.123288307	0.129350203	0.126701085	0.100082937	0.10213750	0.10212834
9	0.036755316	0.032623061	0.026098449	0.028942919	0.029147193	0.023345171	0.02315345	0.02315495
10	0.013663962	0.012127777	0.009702221	0.011146712	0.0114255	0.009232355	0.00899134	0.00899281

11	0.006060016	0.005378712	0.00430297	0.005089261	0.005292837	0.004308028	0.00413327	0.00413428
12	0.0114627	0.010173994	0.010173994	0.009409828	0.009675478	0.007830525	0.00760150	0.00760287
13	0.127371045	0.113051223	0.090440978	0.095943614	0.094500061	0.074853687	0.07595611	0.07595163
14	0.075723016	0.067209778	0.053767822	0.058108422	0.057768089	0.045971171	0.04620403	0.04620369
15	0.08167816	0.072495408	0.057996326	0.062509051	0.062058993	0.049352451	0.04967157	0.04967084
16	0.094941077	0.084267228	0.067413783	0.072269866	0.071557002	0.056829348	0.05735538	0.05735368
17	0.216281588	0.191965907	0.153572725	0.159864654	0.155977602	0.122967468	0.12600000	0.12598595
18	0.062220649	0.055225428	0.044180343	0.048083039	0.04796936	0.038240453	0.03829552	0.03829600
19	0.023339844	0.020715838	0.01657267	0.018679436	0.018964439	0.015251102	0.01500000	0.01500165
20	0.0114627	0.010173994	0.008139195	0.009409828	0.009675478	0.007830525	0.00760150	0.00760287
21	0.009306041	0.008259799	0.006607839	0.007696489	0.00794328	0.006440603	0.00622830	0.00622955
22	0.065131451	0.057808981	0.046247184	0.050250336	0.050090621	0.039915193	0.04000631	0.0400066
23	0.059324368	0.052654765	0.042123812	0.045922955	0.045853397	0.0365692	0.03658976	0.03659038
24	0.014903962	0.013228368	0.010582695	0.012120612	0.012404501	0.010015665	0.00976980	0.00977131
25	0.023339844	0.020715838	0.01657267	0.018679436	0.018964439	0.015251102	0.01500000	0.01500165

Table 11 Semi-detached COCOMO algorithm comparison.

Sl.No	GA	PSO	DE	DEBM	Hybrid Algo	HABDE	SGO	SGO-PPF
1	1.28539338	1.137900396	1.251690436	1.065845111	0.986180343	0.758600264	0.74878398	0.74940815
2	5.784279909	5.262524067	5.788776473	5.499084096	4.560854191	3.508349378	3.62114232	3.62560511
3	0.295543542	0.254731191	0.28020431	0.214408413	0.220767032	0.169820794	0.16046293	0.16053416
4	0.041765661	0.03473994	0.038213934	0.025362086	0.030107948	0.02315996	0.02064829	0.02064675
5	0.00574033	0.004605491	0.00506604	0.002910379	0.003991426	0.003070327	0.00258068	0.00257913
6	0.232869459	0.199844042	0.219828447	0.16531908	0.17319817	0.133229362	0.12499993	0.12504752
7	0.041765661	0.03473994	0.038213934	0.025362086	0.030107948	0.02315996	0.02064829	0.02064675
8	0.078231615	0.065818456	0.072400302	0.050295179	0.057042662	0.043878971	0.03985633	0.03986000
9	2.131333716	1.9042004	2.09462044	1.850437962	1.650307013	1.269466933	1.27200000	1.27323085
10	0.025158046	0.020734045	0.022807449	0.014589136	0.017969505	0.013822696	0.01213951	0.0121369
11	0.097130475	0.082040729	0.090244802	0.063685797	0.071101965	0.054693819	0.05000000	0.05000745

Table 12. COCOMO-based embedded project algorithm comparison

Sl.No	GA	PSO	DE	DEBM	Hybrid Algo	HABDE	SGO	SGO-PPF
1	0.78907428	0.762771804	0.73383908	0.32878095	0.39453714	0.289327236	0.26413381	0.26439519
2	0.027544993	0.026626826	0.025616843	0.01147708	0.013772496	0.010099831	0.01907601	0.01908182
3	0.110746745	0.107055187	0.102994473	0.046144477	0.055373373	0.04060714	0.05673231	0.05676571
4	0.160682752	0.15532666	0.149434959	0.066951147	0.080341376	0.058917009	0.07593514	0.07598561
5	0.15427706	0.149134491	0.143477665	0.064282108	0.07713853	0.056568255	0.07355351	0.07360178
6	0.17362159	0.167834204	0.161468079	0.072342329	0.086810795	0.075236022	0.08068427	0.08073917
7	0.206664473	0.199775657	0.19219796	0.086110197	0.103332236	0.089554605	0.09248163	0.09254785
8	0.010138396	0.00980045	0.009428709	0.004224332	0.005069198	0.004393305	0.00871917	0.00872004
9	0.013889969	0.01342697	0.012917671	0.005787487	0.006944985	0.006018987	0.01115779	0.01115961
10	0.02375853	0.022966579	0.022095432	0.009899387	0.011879265	0.010295363	0.01698952	0.01699418

11	0.012617891	0.012197294	0.011734638	0.005257454	0.006308945	0.005467753	0.01034910	0.01035061
12	2.751716763	2.659992871	2.559096589	1.146548651	1.375858381	1.192410597	0.70267854	0.70355316
13	1.841731577	1.780340525	1.712810367	0.767388157	0.920865789	0.798083683	0.51305715	0.51365367
14	2.065876733	1.997014175	1.921265362	0.860781972	1.032938367	0.826350693	0.56135237	0.56201821
15	0.831155273	0.803450097	0.772974404	0.346314697	0.415577636	0.332462109	0.27510517	0.27538032
16	0.600502427	0.580485679	0.558467257	0.250209344	0.300251213	0.240200971	0.21326589	0.21346504
17	0.213385091	0.206272254	0.198448134	0.088910454	0.106692545	0.085354036	0.09482920	0.09489772
18	0.28243157	0.273017184	0.26266136	0.117679821	0.141215785	0.112972628	0.11811607	0.11820818
19	0.006157749	0.005952491	0.005726707	0.002565729	0.003078875	0.0024631	0.00590000	0.00589999
20	3.489002526	3.372702442	3.24477235	1.453751053	1.744501263	1.395601011	0.84627922	0.84737362
21	0.240615114	0.23259461	0.223772056	0.100256298	0.120307557	0.096246046	0.10418338	0.10426121
22	0.116814609	0.112920788	0.108637586	0.048672754	0.058407304	0.046725843	0.05915299	0.05918845
23	0.608517996	0.588234063	0.565921736	0.253549165	0.304258998	0.263691132	0.21549252	0.21569433
24	0.02469633	0.023873119	0.022967587	0.010290137	0.012348165	0.010701743	0.01751260	0.01751755
25	0.141599028	0.13687906	0.131687096	0.058999595	0.070799514	0.061359579	0.06877526	0.06881919
26	0.081276026	0.078566825	0.075586704	0.033865011	0.040638013	0.035219611	0.04452242	0.04454581
27	0.03839489	0.03711506	0.035707248	0.015997871	0.019197445	0.016637786	0.02474369	0.02475290

B. MMRE comparison for algorithms based on COCOMO: Regarding the project being discussed, as indicated in Equation (8), a different popular error computation approach termed MMRE has been utilized; the results are shown in Tables 13. The MMRE value for other compared algorithms is imported from [56]. From the table it is cleared with decreased MMRE for SGO algorithm as compared with other COCOMO based algorithms and it is clearly visible through Fig 3- Fig 5.

Table 13. Cocomo81 dataset MMRE comparison for proposed algorithm.

Dataset	SGO	GA	PSO	DE	DEBM	HYBRID	HABDE	SGO-PPF
MMRE (organic)	0.5371	0.6721	0.6280	0.5704	0.5817	0.5777	0.5388	0.5370
MMRE (Semi-detached)	0.5057	0.5462	0.5347	0.5428	0.5228	0.5241	0.5084	0.5057
MMRE(Embedded)	0.6639	0.7630	0.7484	0.7484	0.7306	0.7211	0.7253	0.6638

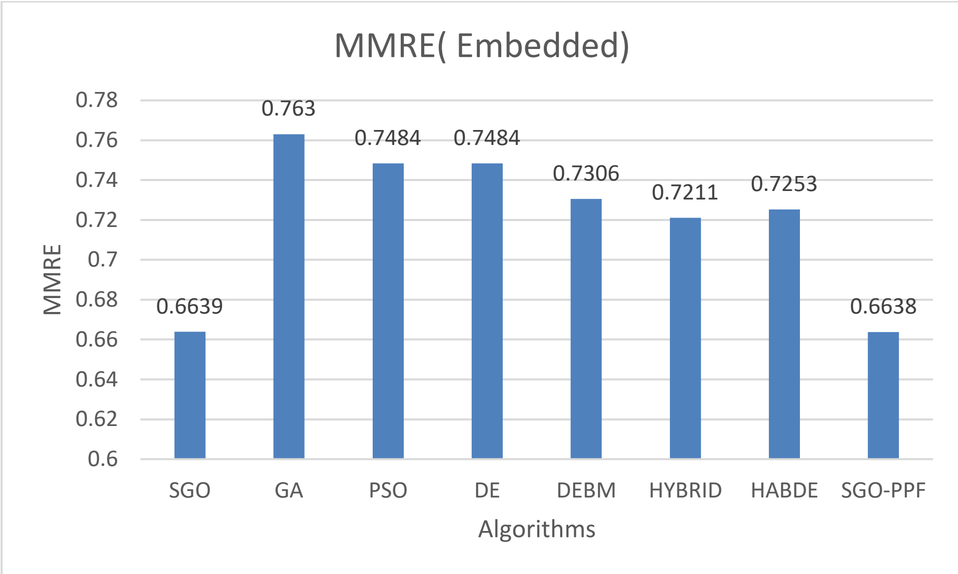


Fig 3. MMRE of proposed SGO-PPF, SGO, GA, PSO, DE, DEBM, HYBRID, HABDE using Organic datasets.

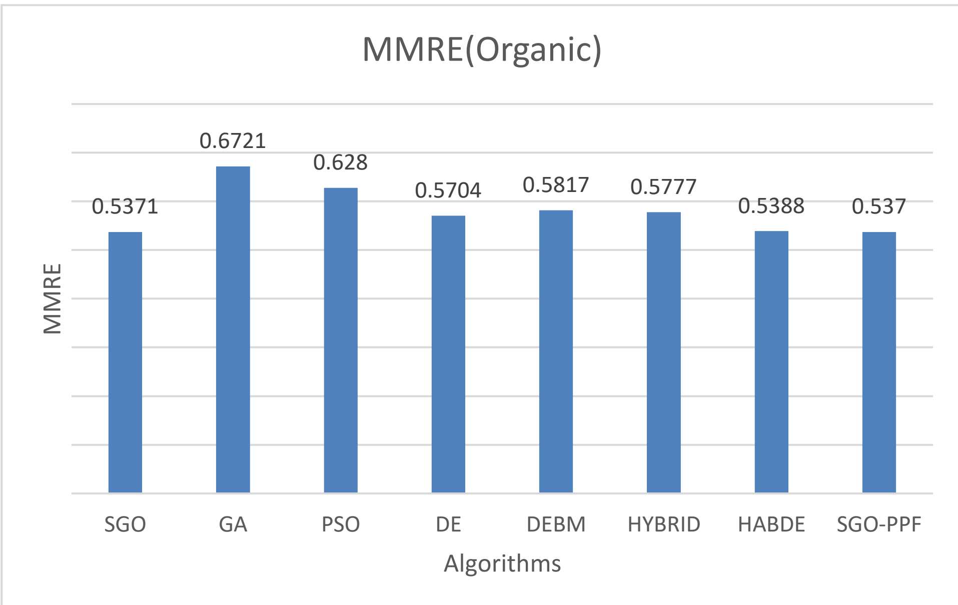


Fig 4. MMRE of proposed SGO-PPF, SGO, GA, PSO, DE, DEBM, HYBRID, HABDE using Semi-detached datasets.

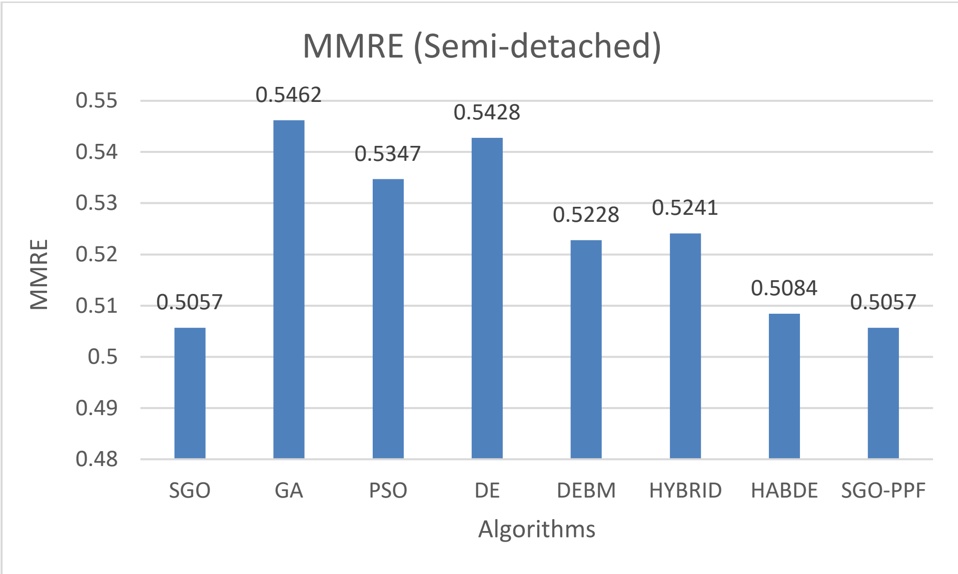


Fig 5. MMRE of proposed SGO-PPF, SGO, GA, PSO, DE, DEBM, HYBRID, HABDE using Embedded datasets.

C. Parameter of COCOMO based SGO algorithm

The rate at which an estimation model approaches the target value, as determined by Eq. (4), is indicated by the rate of convergence parameter (a and b). The findings, displayed in Fig. 6, demonstrate that the suggested COCOMO-based SGO model has a higher rate of convergence because the SGO method can find the optimal values for parameters "a" and "b" after just 18 iterations.

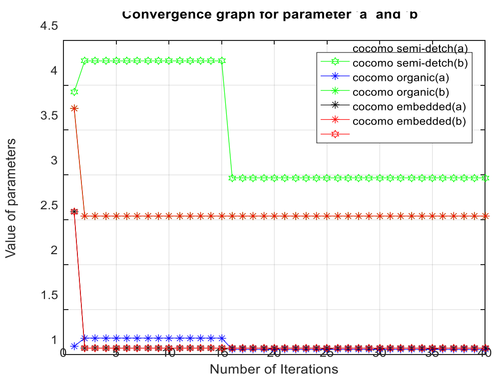


Fig 6. Convergence graph for parameter ‘a’ and ‘b’

Discussion

Comparing the SGO-PPF algorithm to GA, PSO, DE, Hybrid Algorithm, DEBM, and HABDE, we find that it has a greater performance enhancement. Once more, we can observe that the SGO-PPF algorithm requires less iterations to determine the parametric value. In general, other approaches' complexity rises because they generate less diversity, but the suggested algorithm will cut down on both the number of iterations and mistake rate.

4.2.4 Experiment 2

The SGO-PPF algorithm, which is based on software cost estimation, is employed in this experiment to enhance the parameters of the COCOMO II model. We employed SGO-PPF algorithm to determine the parameters of the COCOMO II model, as stated in Eq. (5). We have controlled the SGO-PPF algorithm's evolutionary process using the same set of parameters as in experiment 1. The most recent dataset from the Turkish Software Industry was used to assess the generated model's performance. Five distinct software businesses across multiple domains provided the dataset. The data set comprises twelve projects, each with 25 attributes: Project ID, 5 scale factors, 17 effort multipliers ranging from very low to very high, measured effort as actual effort, as well as project size expressed in KLOC. The appendix's Table IV displays the dataset specifics. For the calibration, all project data points will be employed. Future initiatives in the same category can make use of the calibration results.

Evaluation criteria

The actual costs incurred during project design in real-world environmental conditions are reflected in the evaluation criteria used to calculate the proposed model's software cost estimation accuracy. The evaluation criteria for the experiment are the Mean Magnitude of Relative Error (MMRE) and the Mean Difference (MD) between the actual effort and the estimated effort.

This estimating method's primary goal confirms whether the assumption is accurate; the difference among the realistic actual effort, $Actual\_effort_i$ as well as the expected effort, $Estimate\_effort_i$, ought to measure as closely as feasible. Variance in the large values among $Actual\_effort_i$ and $Estimate\_effort_i$ will cause the prediction to be less accurate and may have a detrimental effect on the effort required to construct the software system. Magnitude of Relative Error (MRE) [57] employed in this work as a standard criterion for software cost estimating to assess the precision of the estimated labor. Each projected point is computed using the MRE as specified in (9):

$$MRE_i=\frac{|Actual\_effort_i-Estimate\_effort_i|}{Actual\_effort_i}\times 100 \qquad \text{--------} \qquad (9)$$

The usage of MMRE [58] "to count the average value of the results from each distinct accuracy prediction value that was measured in the MRE criterion is demonstrated by Equation (10):

$$\text{MMRE}=\frac{1}{N}\sum_{i=1}^{N} MRE_i \qquad \text{-------------------} \ (10)$$

(11) computes the Manhattan distance, which determines the absolute difference in distance between the estimated effort and the actual effort.

$$\text{MD}=\sum_{i=1}^{N}|Actual\_effort_i - Estimate\_effort_i| \qquad \text{---------------} \ (11)$$

Experimental results and analysis

The experiment and the outcomes of applying the suggested method to the dataset are shown in this section. Reducing unknown parameters ranging "a" and "b" coefficients of a COCOMO

II model employing the SGO-PPF methodology is the primary goal of the optimization, which is then compared with BBO-COCOMO-II, PSO, GA, IVR, SEL, Bailey-Basil, Doty, and" Halstead [58–62].

Table 14 provides the values for parameters "a" and "b" for our suggested SGO-PPF algorithm as well as a few other techniques. Table 15 presents the relative error magnitudes for predictions employing models such as SGO-PPF, BBO, COCOMO-II, and others using software projects from the Turkish Industry. Table 16 reports a comparison of mean MMRE and manhattan distance for proposed models such as PSO, SGO-PPF, GA, BBO, and others employing datasets from the Turkish Industry.

Table 14. Value of parameters 'a' and 'b' for different model using Turkish Industry software projects)

Models	'a'	'b'
Proposed SGO-PPF	4.3916	-0.1832
SGO	4.3950	-0.1834
PPF	4.4625	-0.1885
BBO-COCOMO-II	4.2826	-0.1757
GA	4.719	-0.858
PSO	4.2366	-0.1682

Table 15. Magnitude of relative errors for estimations using proposed SGO-PPF, SGO, PPF, BBO, COCOMO-II and others models using Turkish Industry datasets

Project id	1	2	3	4	5	6	7	8	9	10	11	12
	0.3066	0	0.365	0.3651	0.2217	0.9671	0.6295	0.0004	0.861	0.228	0.021	0.1374
SGO	0.3073	0.0006	0.3647	0.3654	0.2219	0.9671	0.6294	0.0005	0.8611	0.228	0.0203	0.1369
PPF	0.32	0.0124	0.3597	0.3702	0.2236	0.9672	0.6263	0.0029	0.8623	0.2286	0.0069	0.1258
BBO	0.2817	0.0198	0.3739	0.3543	0.2157	0.967	0.6349	0.0003	0.8596	0.2293	0.043	0.1559
PSO	0.2819	0.0253	0.3739	0.363	0.2273	0.9664	0.635	0.0155	0.8561	0.2194	0.0511	0.1619
COCOMO-II	1.907	0.4282	1.0676	10.325	14.7888	0.26069	0.14142	28.4174	15.5334	14.999	0.0761	0.0275
GA	0.331	0.3269	0.7429	0.6899	0.7889	0.9971	0.8451	0.896	0.9939	0.9003	0.1489	0.3282
FPA	0.1918	0.0314	0.3951	0.489	0.2187	0.881	0.634	0.0133	0.887	0.2095	0.0522	0.1667
SEL	2.2409	0.3337	0.1949	2.9719	3.3434	0.9904	1.5705	5.9982	5.3792	5.4913	0.849	0.0387
Halstead	2.061	0.01	0.3629	6.3786	9.1666	7.2099	1.8527	24.158	46.509	18.4368	0.1213	0.3184
IVR	9.0228	2.7725	2.9895	15.009	18.1392	9.9144	7.4923	35.392	42.9412	30.456	3.8791	1.8017
Doty	12.9206	4.4631	4.3456	18.6413	21.5216	10.593	10.4385	38.6284	40.9482	34.404	6.3469	3.1486
B-Basil	15.3924	5.1451	5.5475	25.4997	30.8103	17.3215	12.9309	59.9426	73.5306	51.5077	6.9175	3.5538

Table 16. "Mean MMRE and Manhattan distance comparison for Proposed SGO-PPF, SGO, PPF, PSO, GA, and other models using Turkish Industry datasets.

Model	MMRE	MD
Proposed SGO-PPF	34.1910	43.2495
SGO	34.1922	43.2508
PPF	34.2148	43.2846
COCOMO-II	733.13	585.9424
PSO	34.800	43.3571
GA	66.57	60.0558
SEL	245.23	201.1912

Halstead	971.30	1254.72
IVR	1498.41	1459.898
Doty	1719.98	1520.708
Baily Basil	256.749	2504.08
FPA	34.54	43.3417”
BBO-COCOMO-II	34.47	43.2952

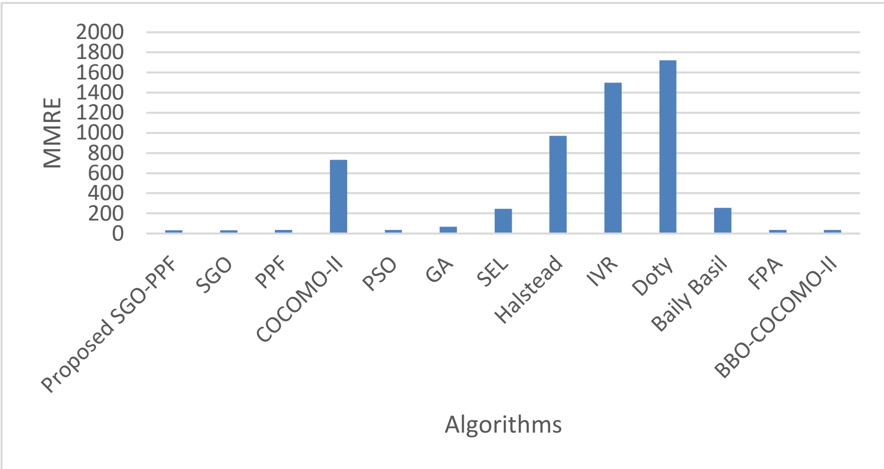


Fig. 7. MMRE for proposed SGO-PPF, SGO, PPF, BBO, PSO, GA, COCOMO-II and others models using Turkish Industry datasets.

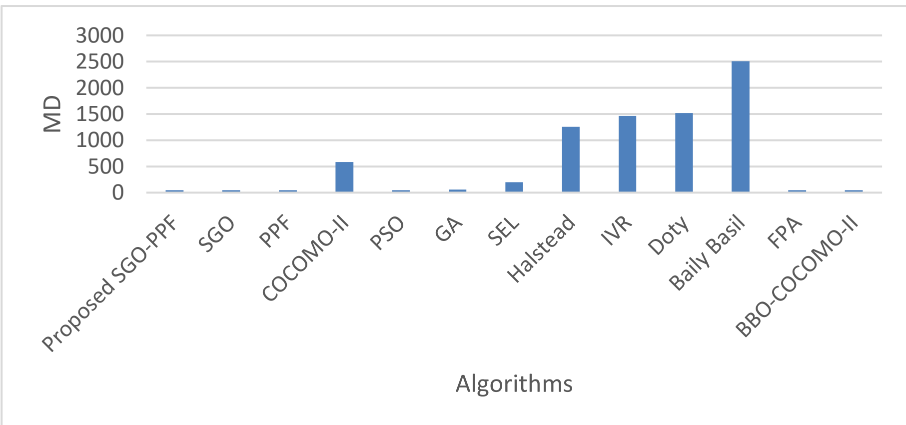


Fig. 8. MD for proposed SGO-PPF, SGO, PPF, BBO, PSO, GA, COCOMO-II and others models using Turkish Industry datasets.

Figs. 7 and 8 compare the MMRE and MD of SGO-PPF, BBO, and other cost estimation methods using Turkish Industry software projects.

Discussion

In this experiment, SGO-PPF based algorithm optimized present parameters of COCOMO-II. The algorithm SGO-PPF which is proposed has undergone testing by using the Turkish Industry software projects. Simulations indicate the SGO-PPF potential-based approach performs COCOMO-II, SGO, PPF, BBO, PSO, GA, FPA, and other cost estimation methods.

5. Conclusion

This research introdues a novel hybrid approach SGO-PPF (social group optimization based past present future) algorithm, as a nature-inspired optimization technique for enhancing the current parameters in COCOMO and COCOMO-II models. Through rigorous testing and experimentation on COCOMO81 & Turkish industry software projects, our proposed SGO-PPF method has demonstrated remarkable superiority over conventional approaches, including COCOMO-II itself, as well as widely-used optimization algorithms like PSO, GA, BBO, SGO, PPF along with other cost estimation techniques. The simulation results prove that SGO-PPF improves software cost estimation accuracy.

To sum up, this study's exceptional outcomes demonstrate the SGO-PPF algorithm's promise as a useful tool for software cost estimation. Its superior performance compared to conventional methods and popular optimization algorithms underscores its capability to enhance estimation accuracy significantly. As we look to the future, our research will concentrate on a number of crucial areas for additional development. We plan to apply various evolutionary algorithms to optimize not only the coefficients of the Constructive Cost Model (COCOMO) but also those of the Constructive Quality Estimation Model (CQEM). This approach aims to create a comprehensive software estimation framework that considers both cost and quality, leading to more accurate project planning and resource allocation.

Furthermore, we intend to examine the adaptability of our SGO-PPF based method across different industry sectors and software development methodologies. This involves customizing and fine-tuning the algorithm to meet the unique characteristics and requirements of various software projects, thereby enhancing its applicability and versatility.

By continuing to develop and refine these techniques, we aim to contribute significantly to the evolution of software project management, fostering more successful and cost-effective software development practices.

References

1. Talbi, E. G. (2009). Metaheuristics: From design to implementation. Wiley.
2. SS, V. C., & HS, A. (2022). Nature inspired meta heuristic algorithms for optimization problems. Computing, 104(2), 251-269..https://doi.org/10.1007/s00607-021-00955-5
3. Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2020). Efficient hybrid nature-inspired binary optimizers for feature selection. Cognitive Computation, 12, 150-175..https://doi.org/10.1007/s12559-019-09668-6.
4. Fister, I., Mernik, M., & Brest, J. (2013). Hybridization of evolutionary algorithms. arXiv preprint arXiv:1301.0929.
5. Naik, A., Satapathy, S.C. Past present future: a new human-based algorithm for stochastic optimization. Soft Comput 25, 12915–12976 (2021). https://doi.org/10.1007/s00500-021-06229-8
6. Satapathy, S., & Naik, A. (2016). Social group optimization (SGO): A new population evolutionary optimization technique. Complex & Intelligent Systems, 2(3), 173–203.

7. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67–82.
8. Salgotra, R., Mittal, N., & Mittal, V. (2023). A New Parallel Cuckoo Flower Search Algorithm for Training Multi-Layer Perceptron. Mathematics, 11(14), 3080.
9. Mohammed, H., Rashid, T. A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. Neural Comput & Applic 32, 14701–14718 (2020). https://doi.org/10.1007/s00521-020-04823-9
10. Goldanloo, M. J., & Gharehchopogh, F. S. (2022). A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems. The Journal of Supercomputing, 78(3), 3998-4031.
11. Jadon, S. S., Tiwari, R., Sharma, H., & Bansal, J. C. (2017). Hybrid artificial bee colony algorithm with differential evolution. Applied Soft Computing, 58, 11-24..
12. El-Shorbagy, M. A., & El-Refaey, A. M. (2020). Hybridization of grasshopper optimization algorithm with genetic algorithm for solving system of non-linear equations. IEEE Access, 8, 220944-220961.
13. Song, H., Bei, J., Zhang, H., Wang, J., & Zhang, P. (2023). Hybrid Algorithm of Differential Evolution and Flower Pollination for Global Optimization Problems. Expert Systems with Applications, 121402.
14. Wang, J. S., & Li, S. X. (2019). An improved grey wolf optimizer based on differential evolution and elimination mechanism. Scientific reports, 9(1), 7181.
15. Naik, A., Satapathy, S. C., & Abraham, A. (2020). Modified Social Group Optimization—A meta-heuristic algorithm to solve short-term hydrothermal scheduling. Applied Soft Computing, 95, 106524.
16. Naik, A. Multi-objective social group optimization for machining process. Evol. Intel. (2023). https://doi.org/10.1007/s12065-023-00856-w
17. 17. Naik, A & Chokkalingam, P. (2011). Binary social group optimization algorithm for solving 0-1 knapsack problem.Decision Science Letters , 11(1), 55-72.
18. Naik, A., Satapathy, S.C., Ashour, A.S. et al. Social group optimization for global optimization of multimodal functions and data clustering problems. Neural Comput & Applic 30, 271–287 (2018). https://doi.org/10.1007/s00521-016-2686-9
19. 19. Singh, A.K., Kumar, A., Mahmud, M. et al. COVID-19 Infection Detection from Chest X-Ray Images Using Hybrid Social Group Optimization and Support Vector Classifier. Cogn Comput (2021). https://doi.org/10.1007/s12559-021-09848-3
20. Kalananda, V. K. R. A., & Komanapalli, V. L. N. (2021). A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems. Applied Soft Computing, 99, 106903.,
21. Satapathy, S. C., Naik, A., & Jena, J. J. (2021). Non-dominated Sorting Social Group Optimization algorithm for multiobjective optimization. Journal of Scientific & Industrial Research, 80(02), 129-136.
22. Rajinikanth, V., Satapathy, S.C. Segmentation of Ischemic Stroke Lesion in Brain MRI Based on Social Group Optimization and Fuzzy-Tsallis Entropy. Arab J Sci Eng 43, 4365–4378 (2018). https://doi.org/10.1007/s13369-017-3053-6
23. Dey, N., Rajinikanth, V., Shi, F., Tavares, J. M. R., Moraru, L., Karthik, K. A., ... & Emmanuel, C. (2019). Social-Group-Optimization based tumor evaluation tool for clinical brain MRI of Flair/diffusion-weighted modality. Biocybernetics and Biomedical Engineering, 39(3), 843-856.
24. Jena, J. J., & Satapathy, S. C. (2021). A new adaptive tuned Social Group Optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems. Multimedia Tools and Applications, 1-35.
25. Tran, D. H. (2020). Optimizing time–cost in generalized construction projects using multiple-objective social group optimization and multi-criteria decision-making methods. Engineering,

Construction and Architectural Management, 27(9), 2287-2313.
26. Swathi, A. S., Chakravarthy, V. V. S. S. S., & Krishna, M. V. (2021). Circular antenna array optimization using modified social group optimization algorithm. Soft Computing, 25(15), 10467-10475.
27. Kraiem, H., Touti, E., Alanazi, A., Agwa, A. M., Alanazi, T. I., Jamli, M., & Sbita, L. (2023). Parameters Identification of Photovoltaic Cell and Module Models Using Modified Social Group Optimization Algorithm. Sustainability, 15(13), 10510.
28. B. W. Boehm, "Software cost estimation meets software diversity", In: Proc. of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017, pp. 495–496, 2017.
29. S. Aljahdali and A. F. Sheta, "Software effort estimation by tuning COOCMO model parameters using differential evolution", In: Proc. of the ACS/IEEE Int. Conf. Comput. Syst. Appl. - AICCSA 2010, pp. 1–6, 2010.
30. I. M. Keshta, "Software Cost Estimation Approaches: A Survey", J. Softw. Eng. Appl., Vol. 10, No. 10, pp. 824–842, 2017.
31. Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering, 158, 107408.https://doi.org/10.1016%2Fj.cie.2021.107408
32. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11, 341-359..
33. Abdel-Basset, M., El-Shahat, D., Jameel, M., & Abouhawwash, M. (2023). Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems. Artificial Intelligence Review, 1-72.
34. Mirjalili, S. M. S. M., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer Adv Eng Softw 69: 46–61.
35. Abdel-Basset, M., Mohamed, R., Azeem, S. A. A., Jameel, M., & Abouhawwash, M. (2023). Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowledge-Based Systems, 268, 110454..https://doi.org/10.1016/j.knosys.2023.110454.
36. Abdel-Basset, M., Mohamed, R., Sallam, K. M., & Chakrabortty, R. K. (2022). Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics, 10(19), 3466.
37. Abdel-Basset, M., Mohamed, R., Zidan, M., Jameel, M., & Abouhawwash, M. (2023). Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems. Computer Methods in Applied Mechanics and Engineering, 415, 116200.
38. Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowledge-Based Systems, 262, 110248.
39. Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. Expert Systems with Applications, 191, 116158..https://doi.org/10.1016/j.eswa.2021.116158.
40. Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111, 300-323.111:300-23.doi: https://doi.org/10.1016/j.future.2020.03.055
41. Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Spider wasp optimizer: A novel meta-heuristic optimization algorithm. Artificial Intelligence Review, 1-64.
42. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.
43. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks

optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.
44. Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. Applied Soft Computing, 36, 315-333.
45. Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Applied Soft Computing59:596-621.
46. Nematollahi AF, Rahiminejad A, Vahidi B (2020) A novel meta-heuristic optimization method based on golden ratio in nature. Soft Computing. 1117-51.
47. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. Applied Soft Computing. 64:161-85
48. Naik, A. Chaotic Social Group Optimization for Structural Engineering Design Problems. J Bionic Eng 20, 1852–1877 (2023). https://doi.org/10.1007/s42235-023-00340-2
49. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation. 1(1):3-18
50. B. W. Boehm, Software engineering economics vol. 197: Prentice-hall Englewood Cliffs (NJ), 1981.
51. B. W. Boehm, "Software engineering economics," IEEE transactions on Software Engineering, pp. 4-21, 1984
52. Gharehchopogh, Farhad Soleimanian, Pourali, Amir, 2015. A new approach based on continuous genetic algorithm in software cost estimation. J. Sci. Res. Dev. 2 (4), 87–94.
53. Gharehchopogh, Farhad S., Maleki, I., Khaze, Seyyed R., A novel particle swarm optimization approach for software effort estimation, Int. J. Acad. Res., Part A: 6 (2), pp. 69–76, 2014
54. Bardsiri, Amid Khatibi, Hashemi, Seyyed Mohsen, 2016. A differential evolution based model to estimate the software services development effort. J. Software Evol. Proceeding 28, 57–7
55. Thamarai, I., Murugavalli, S., An evolutionary computation approach for project selection in analogy based software effort estimation, Indian J. Sci. Technol., 9 (21) 2016
56. Shailendra Pratap Singh, Vibhav Prakash Singh, Ashok Kumar Mehta, Differential evolution using homeostasis adaption based mutation operator and its application for software cost estimation Journal of King Saud University – Computer and Information Sciences 33 (2021) 740–752
57. K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, “Data mining techniques for software effort estimation: A comparative study,” IEEE Transactions on Software Engineering, vol. 38, no. 2, pp. 375–397, 2012.
58. Aman Ullah, Bin Wang, Jinfang Sheng, Jun Long, Muhammad Asim & Zejun Sun,” Optimization of software cost estimation model based on biogeography-based optimization algorithm” Intelligent Decision Technologies 14 (2020),1–448 441DOI 10.3233/IDT-200103.
59. Srivastava DK, Chauhan D, Singh R. VRS model: a model for estimation of efforts and time duration in development of IVR software system. Int J of Software Engineering, IJSE. 2012; 5(1).
60. Chulani S. Bayesian analysis of software cost and quality models. In: Proceedings IEEE International Conference on Software Maintenance. ICSM 2001. IEEE; 2001. pp. 565–568.
61. Boehm BW, Papaccio PN. Understanding and controlling software costs. IEEE Transactions on Software Engineering. 1988; 14(10): 1462–1477.
62. Rarick R, Simon D, Villaseca FE, Vyakaranam B. Biogeography-based optimization and the solution of the power fl w problem. In: 2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE; 2009. pp. 1003– 1008