mathematics

MDPI

Article

Slime Mould Algorithm Based on a Gaussian Mutation for Solving Constrained Optimization Problems

Gauri Thakur [1], Ashok Pal [1], Nitin Mittal [2], Asha Rajiv [3] and Rohit Salgotra [4,5,*]

[1] Department of Mathematics, Chandigarh University, Mohali 140413, Punjab, India; thakurgauri98@gmail.com (G.T.); ashokpmaths.iitr@gmail.com (A.P.)
[2] Department of Industry 4.0, Shri Vishwakarma Skill University, Palwal 121102, Haryana, India; mittal.nitin84@gmail.com
[3] Department of Physics & Electronics, School of Sciences, JAIN (Deemed to Be University), Bangalore 560069, Karnataka, India; asha.rajiv@jainuniversity.ac.in
[4] Faculty of Physics and Applied Computer Science, AGH University of Krakow, 30-059 Krakow, Poland
[5] MEU Research Unit, Middle East University, Amman 11813, Jordan
* Correspondence: r.03dec@gmail.com or rohits@agh.edu.pl

Abstract: The slime mould algorithm may not be enough and tends to trap into local optima, low population diversity, and suffers insufficient exploitation when real-world optimization problems become more complex. To overcome the limitations of SMA, the Gaussian mutation (GM) with a novel strategy is proposed to enhance SMA and it is named as SMA-GM. The GM is used to increase population diversity, which helps SMA come out of local optima and retain a robust local search capability. Additionally, the oscillatory parameter is updated and incorporated with GM to set the balance between exploration and exploitation. By using a greedy selection technique, this study retains an optimal slime mould position while ensuring the algorithm's rapid convergence. The SMA-GM performance was evaluated by using unconstrained, constrained, and CEC2022 benchmark functions. The results show that the proposed SMA-GM has a more robust capacity for global search, improved stability, a faster rate of convergence, and the ability to solve constrained optimization problems. Additionally, the Wilcoxon rank sum test illustrates that there is a significant difference between the optimization outcomes of SMA-GM and each compared algorithm. Furthermore, the engineering problem such as industrial refrigeration system (IRS), optimal operation of the alkylation unit problem, welded beam and tension/compression spring design problem are solved, and results prove that the proposed algorithm has a better optimization efficiency to reach the optimum value.

Keywords: optimization; slime mould algorithm; Gaussian mutation; constrained functions

MSC: 90C27

check for updates

Citation: Thakur, G.; Pal, A.; Mittal, N.; Rajiv, A.; Salgotra, R. Slime Mould Algorithm Based on a Gaussian Mutation for Solving Constrained Optimization Problems. *Mathematics* **2024**, *12*, 1470. https://doi.org/10.3390/math12101470

Academic Editor: Ioannis G. Tsoulos

Received: 8 April 2024
Revised: 5 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

The method of optimization involves finding the best global response for a given problem in the search space. Several real-world problems are facing optimization challenges. The need for novel optimization strategies becomes more apparent than before as problems become more complicated. Several approaches have been significantly advanced in the last few decades to solve optimization problems. For example, prior to the development of heuristic optimization techniques, the only method available for solving issues was mathematical optimization. However, these techniques require knowledge of the optimization problem's properties, such as continuity or differentiability. Metaheuristic techniques have gained increasing momentum in optimization techniques in recent years. Several popular algorithms in this domain are particle swarm optimization (PSO) [1], genetic algorithms (GA) [2], moth flame optimization [3], whale optimization algorithm [4], grey wolf optimization [5], and so on. These algorithms find application in several fields of research and industry as well. Notwithstanding these optimizers' advantages, the fundamental question is whether an optimizer can solve each optimization problem. In optimization, as stated by the No Free-Lunch (NFL) theorem [6], researchers are permitted to create new algorithms that solve optimization problems more successfully. This makes it accessible to researchers to propose algorithm improvements that are more effective in specific situations, which advances the solution of complex problems in optimization.

Constrained optimization problems (COPs) are optimization problems that are currently in existence but are frequently limited by certain restrictions. Constrained evolutionary algorithms handle constrained optimization problems by combining evolutionary algorithms with constraint-handling technology. Evolutionary algorithms are commonly employed to tackle constrained optimization problems because of their strong universality, robustness, reliability, and low information requirements [7]. Meanwhile, over the past 20 years, numerous algorithm-based constraint-handling strategies have been suggested. Currently, the most popular and straightforward approach for handling constraints is the penalty function method [8], which includes or removes a penalty term from the objective function to change a constrained optimization problem into an unconstrained optimization problem. Since most real-world problems include constraints, it is essential to determine the optimum solution for an optimization problem with constraints, such as resources, time, cost, or design constraints. Slime Mould Algorithm (SMA), a unique bio-inspired optimization method [9], was proposed in 2020 by Li et al. The oscillatory behavior of slime mould during foraging serves as its inspiration. The benefits of the SMA are its excellent scalability, few parameters, and straightforward concepts.

While the SMA has been shown to work exceptionally well, it is not without the limitations. It is unable to find stability between the search trends—exploration and exploitation. This limitation could lead to inadequate solution precision, slower convergence, and an inability to come out of local optimum values. To improve the limitations of the SMA algorithm, this study proposes an enhanced algorithm called SMA-GM that presents a Gaussian mutation (GM) technique for improving SMA. The main contributions are as follows:

- The proposed work designs the Gaussian mutation (GM) scheme, which acts on the current positions of slime mould. The proposed scheme can efficiently boost the local search capability of the optimal position of slime mould and avoid falling into the local optima;
- The greedy selection approach is used for selection purposes to make sure that the slime mould with better fitness enters the next generation, ensures the algorithm's convergence speed, and preserves population diversity;
- The proposed strategy is verified by using 13 unconstrained and constrained optimization problems and CEC2022 benchmark functions. Furthermore, the comparison of the SMA-GM with the original SMA and with some well-established optimization algorithms has been shown, and the constrained engineering problem is solved through SMA-GM;
- The experimental results, along with the Wilcoxon rank sum test as a statistical test, demonstrate the advantages and the superior performance of the proposed SMA-GM algorithm. The findings prove that the GM approach effectively improves the classical SMA's search efficiency.

The remaining structure of this paper is organized as follows: Section 2 provides related work. Section 3 presents the preliminaries. In Section 4, the proposed SMA-GM algorithm is explained. Experimental results are given in Section 5. Section 6 includes the mechanical engineering design problem. Lastly, conclusions and directions for further study are presented in Section 7.

2. Related Work

The referenced literature studies show that the researchers have created a wide range of hybrid and metaheuristic versions of SMA to address many kinds of stochastic problems by mimicking the behavior of slime mould in finding out the food source. Yuan L et al. [10] proposed an improved algorithm by embedding elite and chaotic stochastic approaches into the structure of SMA for maintaining better stability among exploration and exploitation stages. Naik et al. [11] enhanced the SMA's exploration capability by adding a balanced optimizer balanced pool, which improves the SMA individuals' randomness. Houssein et al. [12] used an adaptive guided differential evolution (DE) algorithm for improving SMA. Yu et al. [13] introduced a quantum-based enhanced SMA, which gives better results in many optimization problems. A hybrid of arithmetic optimizer with SMA was employed by Chauhan S et al. [14] for improving the lesser internal memory and low convergence at local optimal. Zhao J et al. [15] enhanced the searching ability of SMA by replacing random weights through levy flights. Abualigah L et al. [16] proposed an opposition-based learning algorithm to increase the SMA rate of convergence and further levy flight distribution is incorporated with SMA for improving exploration and exploitation stages. Örnek et al. [17] propose the updating of SMA position by imposing the reciprocal oscillation feature of sine and cosine through each generation, which successfully boosts the exploration phase.

Gaussian mutation (GM) is also an optimization algorithm. Researchers have proposed GM as an improvement mechanism to boost the exploration or exploitation function. Yu et al. [18] proposed the quantum revolving door and GM approach for improving the dragonfly algorithm (DA) performance efficiently. Zhang et al. [19] employed fruit fly optimization (FOA) with a GM strategy and a chaotic local search approach for solving problems of feature. Chen et al. [20] proposed improved bacterial foraging optimization (BFO) with GM and a chaotic design for strengthening its performance. To balance exploration and exploitation, Luo et al. [21] developed an enhanced multi-strategy grasshopper optimization algorithm (GOA) with GM, levy flight, and opposition-based learning strategies. An enhanced Gaussian barebone was proposed by S. Wu et al. [22] as the DE mutation extension to improve the update approach of SMA.

3. Preliminary

3.1. Slime Mould Algorithm

With several novel characteristics and a unique mathematical framework that makes use of adaptive weights to simulate positive and negative feedback through bio-oscillator-based propagation waves as best paths for linking food for better exploration and exploitation, the SMA [9] is a novel stochastic optimizer resulting from the slime mould oscillation model in nature. The SMA mimics the behavioral changes that slime moulds undergo when searching for and encircling food during the foraging phase. They have a fan-like morphology at the front, followed by a web of connecting veins. The diffusive waves are created by the slime mould's bio-oscillator, which modifies the cytoplasmic flow in the veins as they reach the food point and eventually generate the optimal path. The constant z with value 0.03 and transition probabilities p are critical parameters for maintaining balance in the SMA. SMA will randomly explore between the bounds set by lb and ub when randomly created values are smaller than z. On the other hand, SMA will exploit and search in neighboring areas of the current position if the random value, r, is greater than p. SMA will wrap around the current best position when r is smaller than p. The current location's fit will determine the wrapping direction and radius. The following operations highlight the SMA behavior.

Approach food: For simulating the contraction mode and to represent the approaching behavior of slime mould, the given below rule is presented

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \vec{vb}\left(\vec{W} * \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), & r < p \\ \vec{vc} * \overrightarrow{X(t)}, & r \geq p \end{cases} \tag{1}$$

where $\vec{vc}$ decreases linearly from one to zero and $\vec{vb}$ has a range of [−a, a]. $\vec{X}$ indicates the location of the slime mould, $\overrightarrow{X_A}$, and $\overrightarrow{X_B}$ signify two arbitrarily chosen individuals from the population, $\vec{W}$ means the slime mould weight, and t indicates the current iteration. $\overrightarrow{X_b}$ represents the individual position with the maximum odor concentration. The following is the p formula:

$$p = \tanh|\mathrm{S(i)} - \mathrm{DF}| \tag{2}$$

The following is the formula for $\vec{vb}$ and $\vec{W}$:

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{\max\_t}\right) + 1\right) \tag{3}$$

$$\overrightarrow{W(SlI(i))} = \begin{cases} 1 + r * log\left(\frac{bf - S(i)}{bf - wf} + 1\right), & condition \\ 1 - r * log\left(\frac{bf - S(i)}{bf - wf} + 1\right), & others \end{cases} \tag{4}$$

$$SlI = sort(S) \tag{5}$$

where SlI indicates the smell index, i.e., it is the sorted order of fitness values; bf signifies the optimum fitness found in the current iterative manner, wf presents the poorest fitness value found in the iterative course, and $condition$ shows that $S(i)$ ranks the first half of the population and that the random value in the interval [0, 1] is denoted by r.

Wrap food: The mathematical strategy given below is applied for updating the location of slime mould

$$\overrightarrow{X(t+1)} = \begin{cases} rand * (ub - lb) + lb, & rand < z \\ \overrightarrow{x_b(t)} + \vec{vb} * \left(\vec{W} * \overrightarrow{x_A(t)} - \overrightarrow{x_B(t)}\right), & r < p \\ \vec{vc} * \overrightarrow{X(t)} & r \geq p \end{cases} \tag{6}$$

where r and $rand$ indicate the arbitrary value in [0, 1] and lb, ub indicates the lower and upper boundaries of the search range, and z is the parameter taken as 0.03.

Oscillation: With the increasing number of iterations, the $\vec{vb}$ value gradually approaches zero and oscillates arbitrarily between [−a, a]. $\vec{vc}$ oscillates from [−1, 1] and ultimately tends to zero. The pseudocode of SMA is presented in Algorithm 1.

Algorithm 1: Pseudocode of SMA

```
Begin
    1. Initialize the parameters $popsize,\ Max\_iteration$
    2. Initialize the positions of slime mould $X_i (i = 1, 2, \ldots, n)$
    3. While $t \leq Max\_iteration)$
            Calculate the fitness of all slime mould
            Update $bestFitness,\ X_b$
            Calculate the W by Equation (4)
            For each search portion
            Update $p,\ vb,\ vc$
            Update positions by Equation (6)
            End for
            $t = t + 1;$
    4. End While
    5. Return the best fitness and best position
End
```

3.2. Gaussian Mutation

The popular optimization technique that works well at the exploitation stage is the Gaussian mutation (GM). The Gaussian distribution, represented by $N(\mu,\ \sigma^2)$, where μ and σ are the variable's mean and standard deviation, is widely employed in statistics and the natural sciences to represent real-valued random variables. Gaussian distribution includes the renowned $3-\sigma$ rule, which provides a chance for foraging depending upon the requirement of the problem. Gaussian distribution has been widely utilized in the literature to modify control parameter values but it is rarely employed to create new mutation operators to retain a robust local search capability while increasing diversity. The GM operator is used to enhance the SMA to exploit the benefits of the Gaussian distribution fully. The Gaussian probability density formulation is represented in the following way:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{7}$$

The scheme GM is expressed in the following way

$$mutant(x) = x_i * (1 + G(0,\ \sigma)) \tag{8}$$

where x_i is the parameter's current value and $G(0,\ \sigma)$ represents the arbitrary value from Gaussian distribution with mean 0 and standard deviation σ. GM is the mutation operation that introduces population variability and explores the search space. When using GM, the original population is replaced with an arbitrary number that satisfies the variance and mean values. The GM scheme emphasizes a local area close to the original individual based on the normal distribution characteristics and it has demonstrated excellent efficiency in various optimizers.

4. Proposed Methodology

SMA has limitations even though it performs competitively when compared to other algorithms. The reduced population diversity of SMA enables it to enter the local optimum quickly. As the problem grows more complex and converges more slowly in subsequent iterations, SMA finds it challenging to balance exploration and exploitation. To further boost the performance of SMA and considering that the NFL inspires us to enhance the existing algorithms' performance, Gaussian mutation (GM) is proposed to strengthen the SMA's search capability and to achieve local and global optimality and is named the SMA-GM algorithm. Back and Schwefe [23] proposed using the Gaussian mutation (GM) to increase the search efficiency of heuristic algorithms. In general, GM produces new solutions that are near the candidate solutions. When the search is being conducted, it explores the search space in small steps, increasing the diversity of the population. Random perturbations are introduced to the current solution based on a Gaussian (normal) distribution. The proposed SMA-GM holds a better position when compared with the target value of the current optimum individual solution. The algorithm produces well-balanced local and global results because of the proposed strategy. A random variable is created by taking the mean of zero and the standard deviation of one. The Gaussian distribution function can be expressed as follows:

$$G_{gaussian(0,\sigma)}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{9}$$

where the variance value of the candidate solutions is σ^2. To thoroughly study the features of the current population, two locations are arbitrarily selected, and their difference is considered to link with the Gaussian distribution operator to construct the proposed GM mechanism. The following is the presentation for GM strategy:

$$\overrightarrow{G_{new}(t)} = \overrightarrow{X_{best}(t)} * \left(1 + \lambda * \vec{vb} * G(0,1) * (X_C(t) - X_D(t))\right) \tag{10}$$

where $G(0,1)$ is the Gaussian distribution mechanism of Equation (9), and $\overrightarrow{X_{best}(t)}$ is the optimum location for the current iteration of the original SMA. Following the GM operation, the new position is presented by $\overrightarrow{G_{new}(t)}$. The positions of the two slimes selected randomly from the population are represented by $X_C(t)$ and $X_D(t)$, respectively. The value of $\vec{vb}$ oscillates randomly between $[-a,\ a]$ and approaches 0 with increasing iterations. In order to further balance the exploitation and exploration and making the smooth oscillation process, the parameter a is updated as follows:

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{\text{max\_}t}\right) + 1\right) + cos\left(\frac{\pi}{2} * \frac{t}{\text{max\_}t}\right) \tag{11}$$

$$\lambda = 1 - \frac{t^2}{\text{max\_}t^2} \tag{12}$$

where, λ is a dynamic parameter adaptively adjusted with the number of iterations. t indicates the current iteration and max_t indicates maximum number of iterations. Algorithm 2 displays the proposed algorithm's pseudocode after the normal execution of SMA while updating the parameter a. The optimum location is mutated to create a new mutated solution using the GM process through Equation (10).

Algorithm 2: Pseudocode of the SMA-GM algorithm

```
Begin
    1. Initialize the parameters popsize, Max_iteration
    2. Initialize the positions of slime mould $X_i (i = 1, 2, \ldots, n)$
    3. While $t \leq$ Max_iteration)
            Calculate the fitness of all slime mould
            Update $bestFitness,\ X_b$
            Calculate the W by Equation (4) and a by Equation (11)
            For each search portion
            Update $p,\ vb,\ vc;$
            Update positions by Equation (6)
            Update the position of the slime mould and the optimal position as $\overrightarrow{X_{\text{best}}(t.)}$
            Gaussian mutation mechanism
            Update $\lambda$ by Equation (12)
            $\overrightarrow{G_{\text{new}}(t)} = \overrightarrow{X_{\text{best}}(t)} * \left(1 + \lambda * \vec{vb} * G(0,1) * \left(\overrightarrow{X_C(t)} - \overrightarrow{X_D(t)}\right)\right)$
            If F$\left(\overrightarrow{G_{new}(t)}\right) <$ F$\left(\overrightarrow{X_{best}(t)}\right)$
            Update optimal solution to $\overrightarrow{G_{new}(t)}$
            Replace the parent slime with the generated mutant slimes if its fitness is found to be better.
            End If
            End For
            $t = t + 1;$
    4. End While
    5. Return the $best\ fitness$ and $best\ position$
End
```

The new mutated solution is then compared to the optimal position of the original SMA to determine whether the new mutated solution is still the better solution and if not, the optimal position of the original SMA is updated to the optimal solution through a greedy selection mechanism. Individuals who have better fitness are selected to enter the next iteration. The computing process for this is shown below.

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{G_{new}(\mathrm{t})} & if\ F\left(\overrightarrow{G_{new}(\mathrm{t})}\right) < F\left(\overrightarrow{X_{best}(\mathrm{t})}\right) \\ \overrightarrow{X_{best}(\mathrm{t})} & else \end{cases} \tag{13}$$

The pseudocode of SMA-GM illustrates how the proposed method reconsiders the position details following the original algorithm's search for a feasible outcome. The above-mentioned technique not only broadens the population's diversity but also boosts the effectiveness of the search process; in the process, the enhanced version keeps the original algorithm's structure relatively straightforward. SMA-GM consists of the subsequent parts for complexity, such as initialization, fitness evaluation, sorting, weight update, position update, Gaussian mutation, and greedy search. Among them, N represents the number of slime mould, dim represents the dimension, and $Maxt$ represents the maximum number of iterations. The total complexity of SMA-GM is $O(dim + Maxt * N * (2 + logN + dim))$. Figure 1 presents the flowchart of the SMA-GM algorithm.

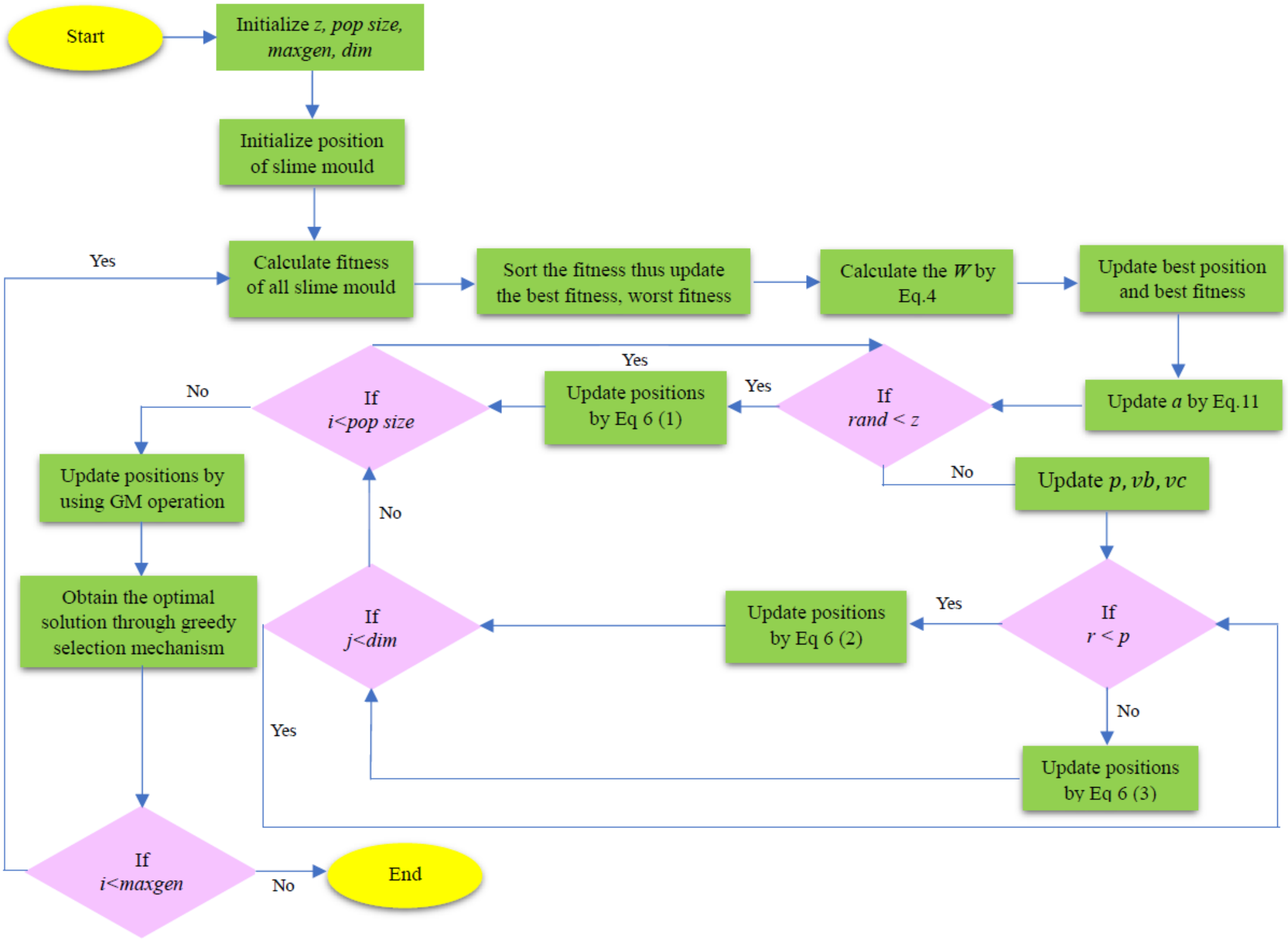
Start
Initialize *z, pop size, maxgen, dim*
Initialize position of slime mould
Calculate fitness of all slime mould
Sort the fitness thus update the best fitness, worst fitness
Calculate the *W* by Eq.4
Update best position and best fitness
Update *a* by Eq.11
If *rand* < *z*
Yes
Update positions by Eq 6 (1)
No
Update *p*, *vb*, *vc*
If *r* < *p*
Yes
Update positions by Eq 6 (2)
No
Update positions by Eq 6 (3)
If *j*<*dim*
Yes
No
If *i*<*pop size*
Yes
No
Update positions by using GM operation
Obtain the optimal solution through greedy selection mechanism
If *i*<*maxgen*
Yes
No
End

Figure 1. Flowchart of the SMA-GM algorithm.

5. Experimental Setup and Results

Optimization problems are generally categorized into two: unconstrained and constrained. The optimization method in the unconstrained problems seeks for an objective function's minimum or maximum possible value. Since the optimal solution for the second kind of problem must meet the constraints specified for the problem, the method is more complex. In simple words, the solution ought to be feasible. To verify the performance of SMA-GM, the results of the proposed algorithm are compared against four state-of-the-art algorithms from the literature, including SMA [9], GWO [5], MFO [3], and WOA [4] and two more improved optimization algorithms are included for fair comparison, which are AGWO [24] and IChoA [25]. The parameter settings of these algorithms are given in Table 1. The environmental conditions are an 11th Gen Intel(R) Core (TM) i3-1125G4 @ 2.00GHz, 8.00 GB of RAM and the Windows 11 operating system and the simulation experiments are set up on the MATLAB R2021a platform. The SMA-GM algorithm was evaluated and compared with other algorithms using 38 functions, including 13 unconstrained, 13 unconstrained (F1–F13), constrained benchmark functions (G1 to G13), and CEC2022 benchmark functions. The "Mean", "St.dev", "Best", and "Worst" values of the objective function are given after each problem was run 30 times and the number of iterations of each algorithm was fixed to 1000 with a population size 30 and the maximum number of function evaluations is set to 30,000.

Table 1. Parameter setting of the involved algorithms.

Algorithm	Parameters
Common	Population size = 30 Maximum iterations = 1000 Number of independent runs = 30
SMA	z = 0.03
GWO	a = [2, 0]
MFO	b = 1, t = [−1, 1], a ∈ [−1, 2]
WOA	a_1 = [2, 0]; a_2 = [−2, −1]; b = 1
AGWO	B = 0.8, a = 2(nonlinear reduction from 2 to 0)
IChoA	m = chaotic vector, C_3 = 1, C_4 = 2, l = 2.5 (nonlinear reduction from 2.5 to 0)

5.1. Unconstrained Benchmark Functions

This section includes the 13 benchmark functions that were utilized for the experiments. These functions are categorized as unimodal (F1–F7) and multimodal functions (F8–F13). The unimodal functions are employed for testing an algorithm's potential for exploitation. Conversely, the ability to explore and the stability of algorithms can be found in the multimodal test functions that have a large number of local minima. For each function, the experiment is carried out for 30, 60, and 200 dimensions and the search range of all the functions is set to [−100, 100]. The main parameter values of each algorithm are listed in Tables 1 and 2 lists the summary of unimodal and multimodal test functions. It should be noted that SMA-GM employed the same parameter values as the original algorithm. As such, it could offer a guarantee of stability in the proposed algorithm's performance. Additionally, for a fair comparison, the test conditions were the same as those listed in Table 1.

Table 2. Summary of unconstrained benchmark functions.

Function	Dim	Range	$\mathbf{f_{min}}$
$F_1 = \sum_{i=1}^{n} u_i^2$	$30, 60, 200$	$[-100, 100]$	0
$F_2 = \sum_{i=1}^{n} \lvert u_i \rvert + \prod_{i=1}^{n} \lvert u_i \rvert$	$30, 60, 200$	$[-10, 10]$	0
$F_3 = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} u_j \right)^2$	$30, 60, 200$	$[-100, 100]$	0
$F_4 = max_i\{\lvert u_i \rvert, 1 \le i \le n\}$	$30, 60, 200$	$[-100, 100]$	0
$F_5 = \sum_{i=1}^{n-1} \left[100\left(u_{i+1} - u_i^2\right)^2 + (u_i - 1)^2 \right]$	$30, 60, 200$	$[-30, 30]$	0
$F_6 = \sum_{i=1}^{n} (\lfloor u_i + 0.5 \rfloor)^2$	$30, 60, 200$	$[-100, 100]$	0
$F_7 = \sum_{i=1}^{n} iu_i^4 + random\ [0, 1)$	$30, 60, 200$	$[-1.28, 1.28]$	0
$F_8 = \sum_{i=1}^{n} -u_i \sin\left(\sqrt{\lvert u_i \rvert}\right)$	$30, 60, 200$	$[-500, 500]$	$-418.9829 \times dim$
$F_9 = \sum_{i=1}^{n} \left[u_i^2 - 10\cos(2\pi u_i) + 10\right]$	$30, 60, 200$	$[-5.12, 5.12]$	0
$F_{10} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} u_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=n}^{n} cos2\pi u_i\right) + 20 + e$	$30, 60, 200$	$[-32, 32]$	0
$F_{11} = \frac{1}{4000}\sum_{i=1}^{n} u_i^2 - \prod_{i=1}^{n} cos\left(\frac{u_i}{\sqrt{i}}\right) + 1$	$30, 60, 200$	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10sin(\pi v_i) + \sum_{i=1}^{n-1} (v_i - 1)^2 \left[\left[1 + 10sin^2(\pi v_{i+1})\right] + (v_{n-1})^2\right]\right\}$ $+\sum_{i=1}^{n} x(u_i, 10, 100, 4),\ v_i = 1 + \frac{u_i+1}{4}$ $x(u_i, a, k, m) = \begin{cases} k(u_i - a)^m & u_i > a \\ 0 & -a < u_i < a \\ k(-\mathrm{u}_i - a)^m & u_i < -a \end{cases}$	$30, 60, 200$	$[-50, 50]$	0
$F_{13} = 0.1\left\{sin^2(3\pi u_1)\right.$ $+\sum_{i=1}^{n} (u_i - 1)^2 \left[1 + sin^2(3\pi u_i + 1)\right]$ $\left. +(u_n - 1)^2\left[1 + sin^2(2\pi u_n)\right]\right\} + \sum_{i=1}^{n} x(u_i, 5, 100, 4)$	$30, 60, 200$	$[-50, 50]$	0

5.1.1. Exploration and Exploitation Analysis

Table 3 displays the results of the unimodal and multimodal test functions. According to the experimental findings, SMA-GM performs better in most of the test functions. Most notably, SMA-GM achieved satisfactory results in both low and high dimensions for the unimodal functions (F1–F7). In all three dimensions, SMA-GM is able to consistently achieve the theoretically optimal solutions for F1 and F3. In comparison, the other comparable algorithms are not very capable of achieving the theoretical optimal value in all the test functions and perform weaker than SMA-GM. Comparing the test results of each dimension, the results show that the SMA-GM's performance has not dropped too much with increasing dimensions, which illustrates that SMA-GM has outstanding local exploitation capability. For the multimodal functions (F8–F13), the SMA-GM steadily achieves the theoretical optimal values at F9–F11 in all three dimensions (30, 60, and 200). SMA-GM has the overall best performance in the multimodal functions, signifying that the improved algorithm extensively boosts the global exploration capability of SMA.

Table 3. Results of the unconstrained benchmark function in different dimensions.

Func.	Dim	Metric	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
F1	30	Mean	0.0000E+00	0.0000E+00	1.1420e-58	3.6667e+03	1.2910E-150	1.3156E-287	1.6126E-21
		Std	0.0000E+00	0.0000E+00	4.6065E-58	5.5605E+03	6.3122E-150	0.0000E+00	4.1464E-21
	60	Mean	0.0000E+00	0.0000E+00	2.7410E-39	9.1879E+03	1.6816E-150	6.1214E-230	1.8770E-10
		Std	0.0000E+00	0.0000E+00	4.2922E-39	1.0502E+04	4.8997E-150	0.0000E+00	6.4058E-10
	200	Mean	0.0000E+00	1.9669E-305	3.9434E-20	1.9153E+05	7.1770E-147	5.3730E-69	8.7000E-02
		Std	0.0000E+00	0.0000E+00	3.6218E-20	1.8886E+04	2.8695E-146	2.9429E-68	1.1720E-01
F2	30	Mean	7.6334E-200	2.6217E-191	8.8027E-35	3.2667E+01	7.6231E-100	2.9271E-163	1.3916E-14
		Std	0.0000E+00	0.0000E+00	7.7522E-35	2.0160E+01	4.1689E-99	0.0000E+00	2.1687E-14
	60	Mean	1.4896E-194	1.4056E-174	1.2233E-23	8.6110E+01	4.2500E-103	6.7393E-128	2.0968E-08
		Std	0.0000E+00	0.0000E+00	6.6131E-24	4.0701E+01	1.8355E-102	2.5883E-127	2.3717E-08
	200	Mean	8.4895E-177	2.7465E-158	1.4618E-12	5.6387E+02	8.7032E-102	2.2380E-47	5.7000E-03
		Std	0.0000E+00	1.5043E-157	4.2765E-13	6.2588E+01	3.0302E-101	1.2258E-46	3.3000E-03
F3	30	Mean	0.0000E+00	0.0000E+00	6.0647E-15	1.7087E+04	2.2457E+04	1.6522E-167	2.7398E+00
		Std	0.0000E+00	0.0000E+00	1.8196E-14	1.0410E+04	1.1010E+04	0.0000E+00	5.1045E+00
	60	Mean	0.0000E+00	0.0000E+00	6.5304E-04	7.5837E+04	2.2594E+05	4.8340E-123	8.2193E+03
		Std	0.0000E+00	0.0000E+00	1.6000E-03	3.1531E+04	4.4994E+04	2.6440E-122	5.9416E+03
	200	Mean	0.0000E+00	6.3250e-320	3.4423E+03	6.9143E+05	4.2199E+06	1.3011E-32	3.4075E+05
		Std	0.0000E+00	0.0000E+00	2.8117E+03	1.5015E+05	1.0616E+06	7.1244E-32	1.1396E+05
F4	30	Mean	4.6331E-210	4.0835E-198	1.3767E-14	6.5007E+01	3.8088E+01	4.6986E-125	2.7000E-03
		Std	0.0000E+00	0.0000E+00	1.3959E-14	1.1037E+01	2.8905E+01	1.9603E-124	5.0000E-03
	60	Mean	6.1647E-202	2.7375E-183	8.5570E-08	8.6026E+01	6.3206E+01	3.3964E-111	3.9345E+00
		Std	0.0000E+00	0.0000E+00	9.8672E-08	3.7970E+00	2.7820E+01	1.4692E-110	3.6291E+00
	200	Mean	9.1750E-189	1.8162E-142	8.8706E+00	9.7009E+01	7.6502E+01	2.7230E-98	5.3073E+00
		Std	0.0000E+00	9.9477E-142	4.6248E+00	9.2640E-01	2.6037E+01	4.1843E-98	1.0233E+01
F5	30	Mean	2.5590E-01	3.7879E+00	2.6922E+01	5.3298E+06	2.7265E+01	2.7589E+01	2.5520E+01
		Std	2.3510E-01	9.3190E+00	6.6630E-01	2.0255E+07	6.0040E-01	7.3900E-01	9.8440E-01
	60	Mean	2.2271E+00	5.8131E+00	5.7298E+01	1.1389E+07	5.7782E+01	5.7853E+01	5.7956E+01
		Std	1.9501E+00	1.4323E+01	9.2360E-01	2.6454E+07	6.0090E-01	6.2410E-01	9.9190E-01
	200	Mean	1.3377E+01	4.1798E+01	1.9763E+02	5.9481E+08	1.9743E+02	1.9852E+02	5.1980E+02
		Std	1.5351E+01	5.3705E+01	6.9930E-01	1.0827E+08	3.2160E-01	2.9500E-01	9.1577E+02
F6	30	Mean	9.7570E-04	9.3449E-04	7.0830E-01	1.6700E+03	5.9300E-02	3.0459E+00	7.5000E-03
		Std	3.7960E-04	5.0173E-04	4.0930E-01	3.7984E+03	6.8300E-02	4.4440E-01	6.5000E-03
	60	Mean	2.0600E-02	2.7500E-02	3.5305E+00	1.0899E+04	6.4480E-01	9.1116E+00	3.3529E+00
		Std	1.6900E-02	1.5600E-02	6.9670E-01	9.7755E+03	3.1760E-01	4.0690E-01	6.9030E-01
	200	Mean	9.0740E-01	2.1431E+00	2.7871E+01	1.8316E+05	6.8375E+00	4.3473E+01	3.7212E+01
		Std	1.0581E+00	2.5638E+00	1.0950E+00	2.4172E+04	1.8806E+00	4.7210E-01	2.3697E+00
F7	30	Mean	8.7485E-05	9.6302E-05	8.7230E-04	2.2649E+00	1.6000E-03	1.1749E-04	1.6000E-03
		Std	7.1310E-05	8.6358E-05	3.6967E-04	4.1453E+00	1.4000E-03	9.1413E-05	9.8050E-04
	60	Mean	8.8723E-05	1.3046E-04	1.7000E-03	4.4999E+01	2.8000E-03	1.7075E-04	5.0000E-03
		Std	8.8346E-05	1.0644E-04	6.1596E-04	4.5085E+01	3.0000E-03	1.3568E-04	3.2000E-03
	200	Mean	1.5401E-04	2.3170E-04	4.4000E-03	1.9388E+03	2.1000E-03	7.4743E-04	7.6600E-02
		Std	1.4408E-04	1.6692E-04	1.5000E-03	4.7091E+02	2.6000E-03	5.3142E-04	7.5000E-02

Table 3. *Cont.*

Func.	Dim	Metric	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
F8	30	Mean	−1.2569E+04	−1.2569E+04	−6.0789E+03	−8.6141E+03	−1.1297E+04	−3.3244E+03	−7.7426E+03
		Std	6.3400E−02	1.0400E-01	8.0426E+02	1.1705E+03	1.5114E+03	3.8411E+02	9.0124E+02
	60	Mean	−2.5138E+04	−2.5138E+04	−1.0625E+04	−1.5298E+04	−2.2263E+04	−4.4142E+03	−1.0631E+04
		Std	1.1267E+00	1.0520E+00	1.0626E+03	1.4798E+03	3.2644E+03	7.2947E+02	1.7171E+03
	200	Mean	−8.3789E+04	−8.3786E+04	−2.9174E+04	−4.0031E+04	−7.6030E+04	−8.1330E+03	−2.5357E+04
		Std	1.0139E+01	1.8694E+01	4.6872E+03	3.7644E+03	9.5686E+03	9.2865E+02	2.6464E+03
F9	30	Mean	0.0000E+00	0.0000E+00	2.4930E-01	1.6683E+02	0.0000E+00	0.0000E+00	1.9671E+01
		Std	0.0000E+00	0.0000E+00	7.7990E-01	4.2543E+01	0.0000E+00	0.0000E+00	1.7795E+01
	60	Mean	0.0000E+00	0.0000E+00	1.1348E+00	3.7428E+02	0.0000E+00	0.0000E+00	2.3614E+01
		Std	0.0000E+00	0.0000E+00	2.5719E+00	7.2331E+01	0.0000E+00	0.0000E+00	2.8944E+01
	200	Mean	0.0000E+00	0.0000E+00	1.5482E+00	1.9325E+03	1.5158E-14	0.0000E+00	4.1362E+01
		Std	0.0000E+00	0.0000E+00	3.4541E+00	1.0024E+02	8.3025E-14	0.0000E+00	3.1214E+01
F10	30	Mean	8.8818E-16	8.8818E-16	1.5573E-14	1.4131E+01	3.7303E-15	4.6777E-15	2.4759E-12
		Std	0.0000E+00	0.0000E+00	2.5945E-15	8.0236E+00	2.8605E-15	9.0135E-16	7.7754E-12
	60	Mean	8.8818E-16	8.8818E-16	4.2218E-14	1.9587E+01	3.7303E-15	6.3357E-15	2.2566E-06
		Std	0.0000E+00	0.0000E+00	3.0208E-15	4.8200E-01	2.1681E-15	1.8027E-15	3.2071E-06
	200	Mean	8.8818E-16	8.8818E-16	1.2428E-11	1.9942E+01	3.7303E-15	7.6383E-15	1.9300E-02
		Std	0.0000E+00	0.0000E+00	5.1053E-12	1.6200E-02	2.5380E-15	1.0840E-15	1.3000E-02
F11	30	Mean	0.0000E+00	0.0000E+00	1.8000E-03	1.8086E+01	5.7000E-03	0.0000E+00	3.2000E-03
		Std	0.0000E+00	0.0000E+00	6.4000E-03	4.3768E+01	3.1100E-02	0.0000E+00	5.6000E-03
	60	Mean	0.0000E+00	0.0000E+00	2.6000E-03	1.0863E+02	6.2000E-03	0.0000E+00	2.7000E-03
		Std	0.0000E+00	0.0000E+00	6.1000E-03	1.0392E+02	2.3600E-02	0.0000E+00	7.2000E-03
	200	Mean	0.0000E+00	0.0000E+00	2.9000E-03	1.6379E+03	0.0000E+00	0.0000E+00	4.2600E-02
		Std	0.0000E+00	0.0000E+00	9.3000E-03	2.3127E+02	0.0000E+00	0.0000E+00	4.7800E-02
F12	30	Mean	8.3411E-04	1.3000E-03	3.5500E-02	9.0740E-01	4.3200E-02	2.2990E-01	8.9249E-04
		Std	8.5665E-04	1.6000E-03	1.9600E-02	1.0168E+00	2.0430E-01	6.2200E-02	1.4000E-03
	60	Mean	2.1000E-03	2.6000E-03	1.2490E-01	2.5636E+07	1.1500E-02	5.1330E-01	1.9970E-01
		Std	3.0000E-03	5.0000E-03	6.8700E-02	7.8110E+07	6.2000E-03	6.2300E-02	6.4100E-02
	200	Mean	1.6000E-03	3.1000E-03	4.9390E-01	1.1459E+09	2.3900E-02	9.1770E-01	1.0377E+00
		Std	3.2000E-03	6.8000E-03	3.8100E-02	3.5311E+08	9.7000E-03	2.3400E-02	1.5627E+00
F13	30	Mean	6.2781E-04	1.4000E-03	6.1850E-01	4.8580E-01	2.0330E-01	2.1164E+00	1.2368E+00
		Std	4.5182E-04	2.8000E-03	2.5900E-01	1.1769E+00	1.3870E-01	1.5080E-01	4.2830E-01
	60	Mean	6.5000E-03	9.5000E-03	2.8045E+00	7.7544E+07	8.3050E-01	5.1814E+00	4.7267E+00
		Std	4.7000E-03	1.5400E-02	4.1280E-01	1.9330E+08	3.8300E-01	1.3920E-01	3.0040E-01
	200	Mean	7.9700E-02	1.7200E-01	1.5998E+01	2.4956E+09	3.9585E+00	1.9365E+01	3.0459E+01
		Std	1.1340E-01	2.5920E-01	4.6650E-01	5.9820E+08	1.2726E+00	1.1070E-01	2.2853E+01

5.1.2. Convergence and Scalability Analysis

In order to study the convergence performance of SMA-GM, convergence curves are plotted according to the results with 60 dimensions, as given in Figure 2. In unimodal functions, the curve is smooth and continuously decreasing, demonstrating the algorithm's ability to find the optimal solution. For multimodal functions, the convergence curve descends in steps, signifying the algorithm's capability to steadily escape local optima and reach the global optimum. At the beginning of the iteration, the decline speed for F1, F2, F4, and F10 of SMA and AGWO in the convergence curve is faster than that of SMA-GM. But in the later stage of the iteration, SMA-GM exceeds SMA and AGWO, indicating the robust development ability of the proposed algorithm. Therefore, the improvement strategy proposed in this paper can successfully enhance the convergence speed of SMA and achieve better optimization results.

The scalability test reveals the performance fluctuation of optimization algorithms. In this study, the performance of SMA-GM in various dimensions (dim = 30, 60, 200) was also tested. It is evident that the algorithm will have greater difficulty in locating the global optimal solution for the higher dimension. Notably, all other experimental parameters were constant in accordance with the aforementioned descriptions, with the exception of the dimension setting. The experiments' mean value (mean) and standard deviation (Std) are used to measure the results and Table 3 displays the experimental results. A particular algorithm's quality and accuracy are indicated by its mean value, while its stability is indicated by its standard deviation. The results of the unimodal and multimodal functions both showed that SMA-GM performed exceptionally well in higher dimensions. SMA-GM outperformed SMA and other well-known algorithms in all of the functions except F6. It should be mentioned that these comparison algorithms (GWO, MFO, WOA, AGWO, and IChoA) usually showed weak optimization capabilities, particularly in higher dimensions where the algorithms' performance fluctuates with increasing dimensionality. Hence, the proposed SMA-GM performed more consistently with better optimization behavior when handling high-dimensional problems.

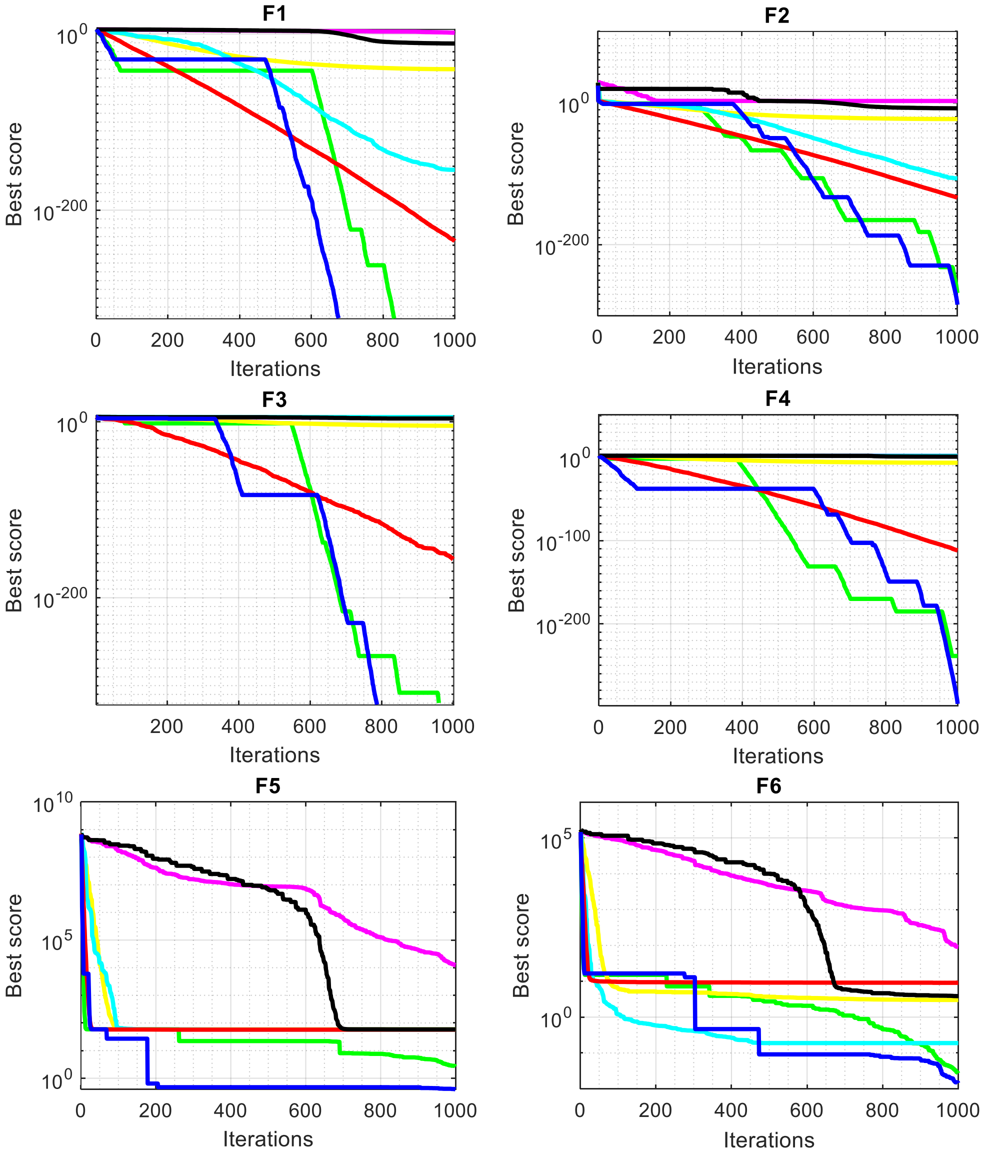

Figure 2. *Cont.*

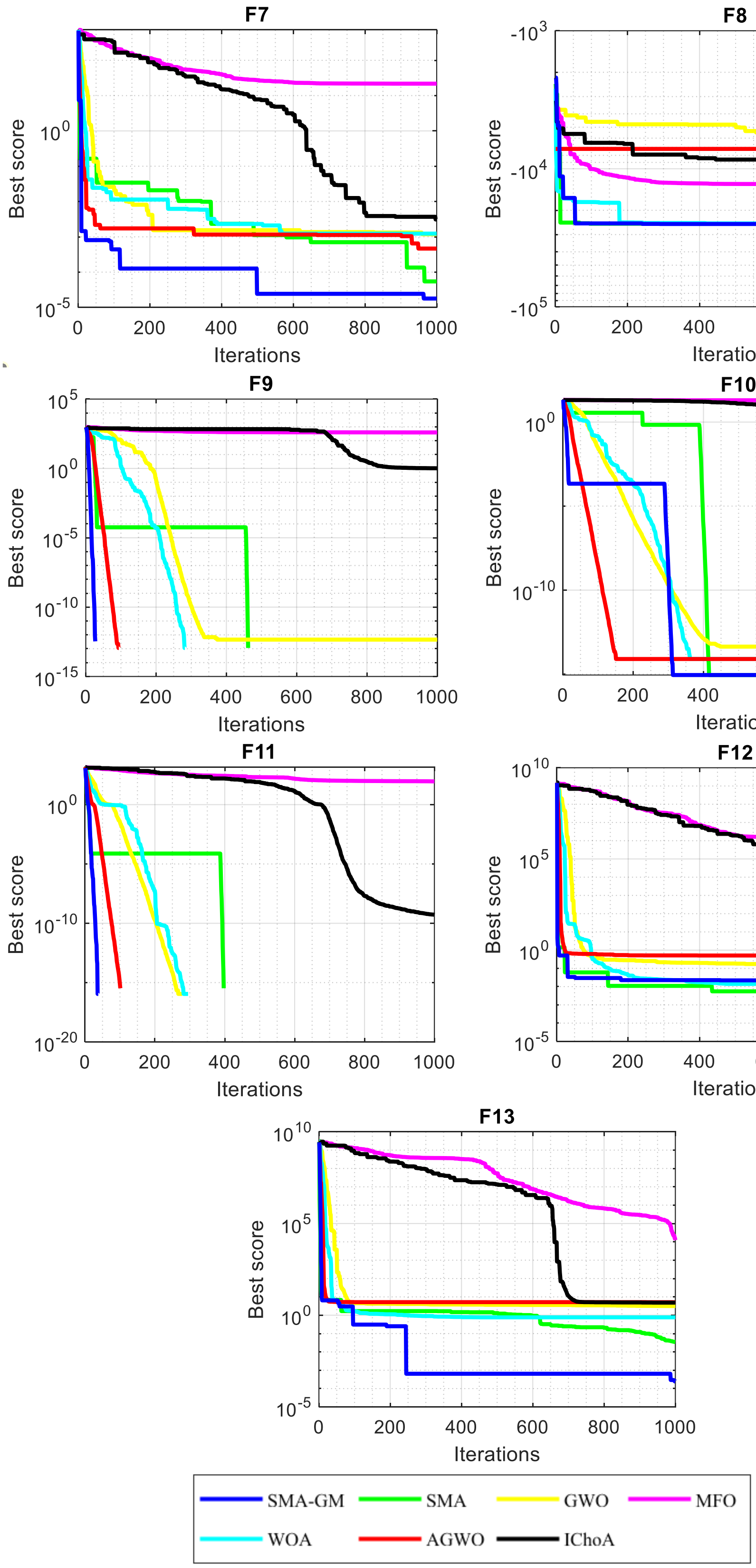

Figure 2. Convergence curves of unconstrained benchmark functions with 60 dim.

5.1.3. Diversity Analysis

Generally, the algorithm that gives priority to the exploitation phase will have a high convergence rate but there would be a risk of trapping into the local optima. However, the algorithm that focusses more on the exploration phase finds the more search space and the chance of finding the global solution increases though the convergence rate may be weak. Thus, a rationally working algorithm attempts to maintain the balance between these two phases. The diversity test is one of the methods for balance analysis between these two phases. The diversity curves, as seen in Figure 3 for unconstrained functions, indicate the average distance between the search agents during the iterations. The diversity is derived using the following equations, which were inspired by certain studies [26,27]:

$$DV_j = \frac{1}{N}\sum_{i=1}^{N}\Big|median\Big(x^j - x_i^j\Big|\Big) \tag{14}$$

$$DV = \frac{1}{m}\sum_{j=1}^{m} DV_j \tag{15}$$

where N is the number of search agents, m indicates the dimension of the problem, x_i^j presents the dimension j of search agent i, and $median\left(x^j\right)$ presents the median of dimension j of the total population.

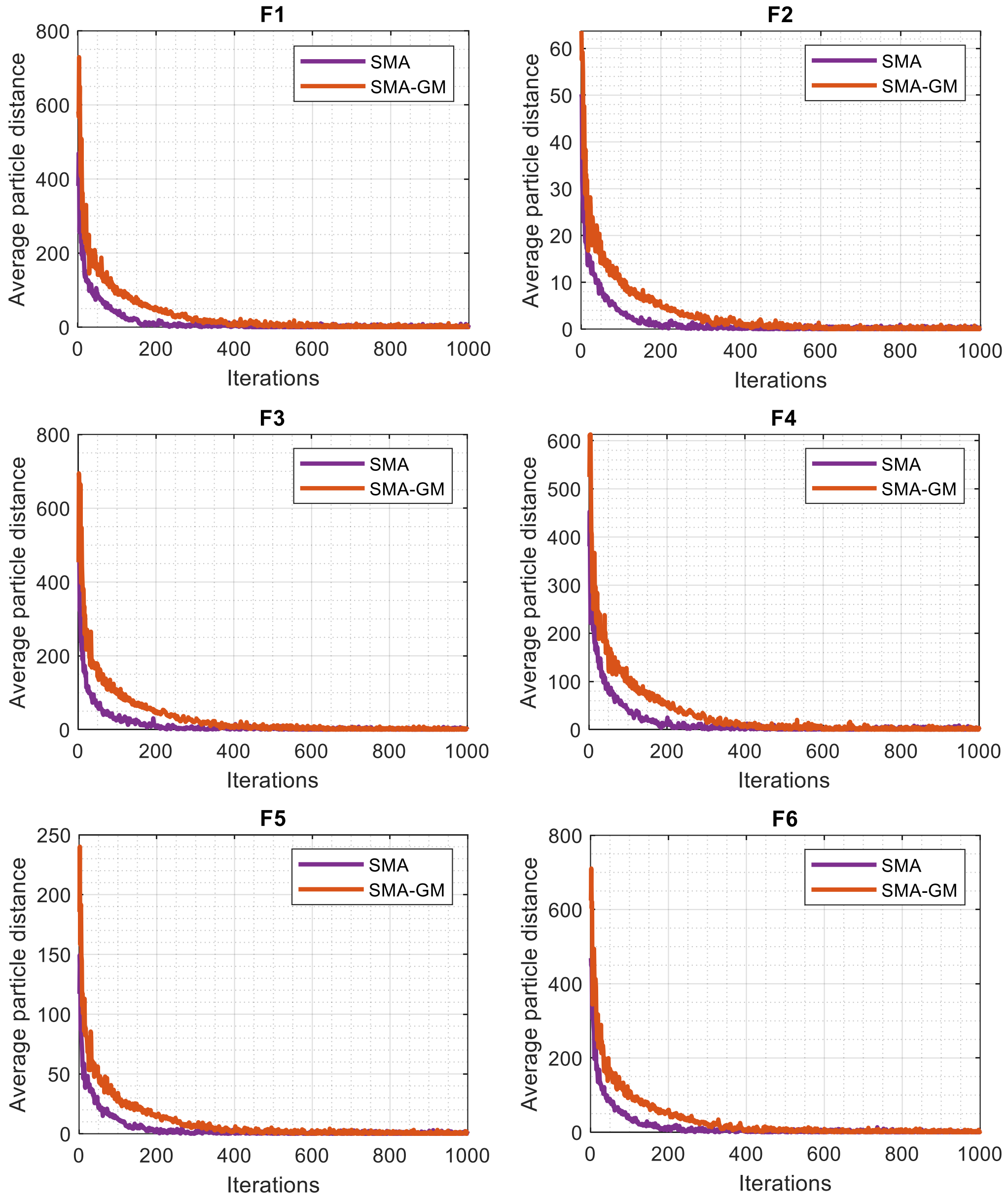

Figure 3. *Cont.*

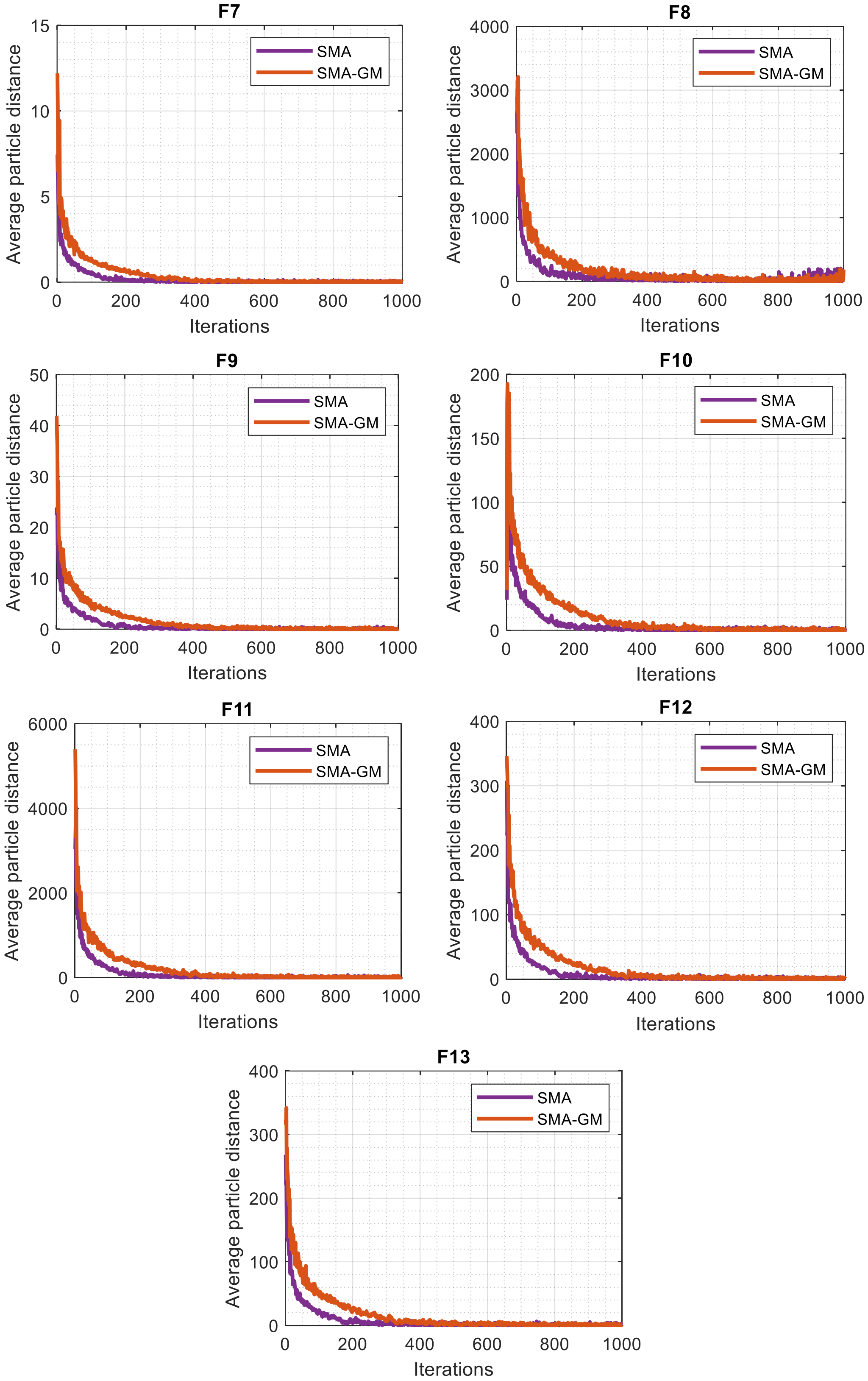

Figure 3. Diversity analysis of SMA and SMA-GM.

The diversity in each dimension is defined as the distance between the dimension *j* of each search agent and the average median of the dimension. The variation in diversity between the original SMA and SMA-GM is shown in Figure 3. The average distance between the search agents is displayed on the vertical axis, while the number of iterations is displayed on the horizontal axis. Because of the random initialization process, it is evident that both algorithms reflect high diversity in the initial phase of the execution. Then, as a result of various algorithmic operations, the diversity varies. As the iterations increase, the search agents traverse the whole search space and the average distance gradually becomes smaller, making the diversity decrease, and the algorithms gradually go into the exploitation phase, which is consistent with the above analysis. However, in most of the test functions, the average distance in each iteration is high for the SMA-GM algorithm as compared to SMA, which shows the better ability of search in the SMA-GM in terms of exploring new search regions of the search space. This shows the effect of improving the search mechanism of slime moulds through the GM strategy in the SMA-GM algorithm.

5.2. Constrained Handling Technique

The intelligent constrained handling strategy must use the constraints (restrictions) conditions during optimization to optimize the constrained problems [28,29]. The method of including a penalty to the objective function's value is employed in this article. Equations (16)–(19) were applied for this purpose. The constraint and objective function values are denoted by the parameters $c(u)$ and $f(u)$, respectively, in these equations. Furthermore, the constraint values are $g(u)$ for the inequality and $h(u)$ for the equality constraints. Here, u is the solution vector. Additionally, the number of inequality and equality constraints is q and m, respectively, and the whole constraint is j. In this study, 1E-04 is the small tolerance for the δ parameter.

$$fitness = f(u) + 10^{20}\sum_{1}^{j} C_j^2 \tag{16}$$

$$C_j(u) = \begin{cases} max(0,\ g_j(u)) & if\ 1 \le j \le q \\ max(0,\ (|h_j(u)|) - \delta) & if\ q+1 \le j \le m \end{cases} \tag{17}$$

$$inequality\ constraints : g_j(u) \le 0, (j = 1, \ldots, q) \tag{18}$$

$$equality\ constraints : h_j(u) = 0, (j = q+1, \ldots, m) \tag{19}$$

5.3. Results and Discussion on Constrained Functions

Considering 13 commonly used constrained benchmark functions, the proposed SMA-GM's capability to handle constrained problems was evaluated. Table 4 shows the different linear, non-linear, and quadratic equations for each problem in the set that take the form of equality and inequality constraints. Table 5 illustrates the results for each constrained benchmark function employed with SMA-GM and other algorithms. The results clearly illustrate that SMA-GM outperforms all other algorithms for 10 of the 13 constrained benchmark functions with excellent efficiency, showing the superiority of the proposed algorithm. IChoA outperformed SMA-GM in the G2 problem; in the G8 problem, SMA-GM performed better than SMA but could not outperform the other algorithms for the G8 problem and for the G5 problem, IChoA performed better than SMA-GM. For problem G11, compared with SMA, GWO, and IChoA, SMA-GM obtained similar results and for the G12 problem, SMA-GM, MFO, and IChoA acquired identical results. The algorithm has produced satisfactory results, either optimal or nearly optimal. According to the results, the proposed Gaussian mutation (GM)-based SMA-GM outperforms the competitive meta-heuristic techniques in terms of solution accuracy except for the G2, G5, and G8 problems. Boxplots in Figure 4 strengthen the conclusion mentioned above. The SMA-GM algorithm demonstrates excellent robustness for handling constrained problems, as evidenced by the lowermost position and least height of the box, which is plotted by the obtained outcomes.

Table 4. Summary of constrained benchmark functions.

Func.	Objective Function	Constraints	No. of Variables	Global Best
G1	$f(u) = 5\sum_{i=1}^{4} u_i - 5\sum_{i=1}^{4} u_i^2 - 5\sum_{i=5}^{13} u_i$	$g1 = 2u_1 + 2u_2 + u_{10} + u_{11} - 10 \leq 0$ $g2 = 2u_1 + 2u_3 + u_{10} + u_{12} - 10 \leq 0$ $g3 = 2u_2 + 2u_3 + u_{11} + u_{12} - 10 \leq 0$ $g4 = -8u_1 + u_{10} \leq 0$ $g5 = -8u_2 + u_{11} \leq 0$ $g6 = -8u_3 + u_{12} \leq 0$ $g7 = -2u_4 - 2u_5 + u_{10} \leq 0$ $g8 = -2u_6 - u_7 + u_{11} \leq 0$ $g9 = -2u_8 - u_9 + u_{12} \leq 0$	13	−15
G2	$f(u) = \frac{\sum_{i=1}^{n} cos^4(u_i) - 2\prod_{n=1}^{n} cos^2(u_{\ i})}{\sqrt{\sum_{i=1}^{4} iu_i^2}}$	$g1 = -\prod_{i=1}^{n} u_i + 0.75 \leq 0$ $g2 = \sum_{i=1}^{n} u_i - 7.5n \leq 0$	20	−0.803619
G3	$f(u) = (\sqrt{n}^n)\prod_{i=1}^{n} u_i$	$g1 = \sum_{i=1}^{n} u_i^2 - 1 = 0$	20	−1
G4	$f(u) = -5.3578547u_3^2 +$ $0.8356891u_1u_5 +$ $37.293239\ u_1 - 40792.141$	$g1 = p(u) - 92\ \leq 0$ $g2 = -p(u) \leq 0$ $g3 = r(u) - 110 \leq 0$ $g4 = -r(u) + 90 \leq 0$ $g5 = s(u) - 25 \leq 0$ $g6 = -s(u) + 20 \leq 0$ *where* $p(u) = 85.334407 + 0.0056858u_2u_5 + 0.0006262u_1u_4 - 0.0022053u_3u_5$ $r(u) = 80.51249 + 0.0071317u_2u_5 + 0.0029955u_1u_2 + 0.0021813u_3^2$ $s(u) = 9.300961 + 0.0047026u_3u_5 + 0.0012547u_1u_3 + 0.0019085u_3u_4$	5	−30 665.539
G5	$f(u) = 3u_1 + 10^{-6}u_3^1 + 2u_2 +$ $\frac{2}{3} * 10^{-6}u_3^2$	$g1 = u_3 - u_4 - 0.55 \leq 0$ $g2 = u_4 - u_3 - 0.55\ \leq 0$ $g3 = 1000[\sin(-u_3 - 0.25)] + \sin(-u_4 - 0.25) + 894.8 - u_1 = 0$ $g4 = 1000[\sin(u_3 - 0.25)] + \sin(u_3 - u_4 - 0.25) + 894.8 - u_2 = 0$ $g5 = 1000[\sin(u_4 - 0.25)] + \sin(u_4 - u_3 - 0.25) + 1294.8 = 0$	4	5126.4981
G6	$f(u) = (u_1 - 10)^3 + (u_2 - 20)^3$	$g1 = -(u_1 - 5)^2 - (u_2 - 5)^2 + 100 \leq 0$ $g2 = (u_1 - 6)^2 + (u_2 - 5)^2 - 82.81 \leq 0$	2	−6961.81388
G7	$f(u) = u_1^2 + u_2^2 + u_1u_2 -$ $14u_1 - 16u_2\ + (u_3 - 10)^2 +$ $4(u_4 - 5)^2 + (u_5 - 3)^2 +$ $2(u_6 - 1)^2 + 5u_7^2 + 7(u_8 - 11)^2 +$ $2(u_9 - 10)^2 + (u_{10} - 7)^2 + 45$	$g1 = 4u_1 + 5u_2 - 3u_7 + 9u_8 - 105 \leq 0$ $g2 = 10u_1 - 8u_2 - 17u_7 + 2u_8 \leq 0$ $g3 = -8u_1 + 2u_2 + 5u_9 - 2u_{10} - 12 \leq 0$ $g4 = 3(u_1 - 2)^2 + 4(u_2 - 3)^2 + 2u_3^2 - 7u_4 - 120 \leq 0$ $g5 = 5u_1^2 + 8u_2 + (u_3 - 6)^2 + 2u_4 - 40 \leq 0$ $g6 = u_1^2 + 2(u_2 - 2)^2 - 2u_1u_2 + 14u_5 - 6u_6 \leq 0$ $g7 = 0.5(u_1 - 8)^2 + 2(u_2 - 4)^2 + 3u_5^2 - u_6 - 30 \leq 0$ $g8 = -3u_1 + 6u_2 + 12(u_9 - 8)^2 + 7u_{10} \leq 0$	10	24.3062091
G8	$f(u) = \frac{sin^3(2\pi u_1)\sin(2\pi u_2)}{u_1^3(u_1 + u_2)}$	$g1 = u_1^2 - u_2 + 1\ \ \leq 0$ $g2 = 1 - u_1 + (u_2 - 4)^2 \leq 0$	2	0.095825
G9	$f(u) = (u_1 - 10)^2 + 5(u_2 - 12)^2 +$ $u_3^4 + 3(u_4 - 11)^2 +$ $10u_5^6 + 7u_6^2 + u_7^4 - 4u_6u_7 -$ $10u_6 - 8u_7$	$g1 = 2u_1^2 + 3u_2^4 + u_3 + 4u_4^2 + 5u_5 - 127\ \ \leq 0$ $g2 = 7u_1 + 3u_2 + 10u_3^2 + u_4 - u_5 - 282 \leq 0$ $g3 = 23u_1 + u_2^2 + 6u_6^2 - 8u_7 - 196 \leq 0$ $g4 = 4u_1^2 + u_2^2 - 3u_1u_2 + 2u_3^2 + 5u_6 - 11u_7 \leq 0$	7	680.6300573
G10	$f(u) =\ u_1 + u_2 + u_3$	$g1 = -1 + 0.0025(u_4 + u_6) \leq 0$ $g2 = -1 + 0.0025(-u_4 + u_5 + u_7) \leq 0$ $g3 = -1 + 0.01(-u_5 + u_8) \leq 0$ $g4 = 100u_1 - u_1u_6 + 833.33252u_4 - 83333.333\ \ \leq 0$ $g5 = u_2u_4 - u_2u_7 - 1250u_4 + 1250u_5 \leq 0$ $g6 = u_3u_5 - u_3u_8 - 2500u_5 + 1250000\ \ \leq 0$	8	7049.3307

Table 4. *Cont.*

Func.	Objective Function	Constraints	No. of Variables	Global Best
G11	$f(u) = u_1^2 + (u_2 - 1)^2$	$g1 = u_2 - u_1^2 = 0$	2	0.75
G12	$f(u) = 1 - 0.01\Big[(u_1 - 5)^2 + (u_2 - 5)^2 + (u_3 - 5)^2\Big]$	$g_{\{(p,r,s)\}} = (u_1 - p)^2 + (u_2 - r)^2 + (u_3 - s)^2 - 0.0625 \leq 0$ $p, r, s = 1, \ldots, 9$	3	−1
G13	$f(u) = e^{u_1 u_2 u_3 u_4 u_5}$	$g1 = u_1^2 + u_2^2 + u_3^2 + u_4^2 + u_5^2 - 10 = 0$ $g2 = u_2 u_3 - 5u_4 u_5 = 0$ $g3 = u_1^3 + u_2^3 + 1 = 0$	5	0.0539498

Table 5. Comparison results of constrained benchmark functions.

Func.	Metric	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
G1	Mean	−1.4834E+01	−1.3171E+01	−1.0193E+01	−1.1600E+01	−5.9204E+00	−7.8403E+00	−1.2915E+01
	Std	5.9100E−02	1.7645E+00	2.3500E+00	2.0443E+00	3.2965E+00	1.7856E+00	1.5116E+00
	Best	−1.5000E+01	−1.5000E+01	−1.4965E+01	−1.5000E+01	−1.4927E+01	−1.1854E+01	−1.4954E+01
	Worst	−1.4692E+01	−9.0061E+00	−5.9998E+00	−9.0000E+00	−2.0000E+00	−5.0000E+00	1.5116E+00
G2	Mean	−5.3780E-01	−4.4930E-01	−7.0890E-01	−4.5880E-01	−4.2790E-01	−5.6220E-01	−7.7500E-01
	Std	1.1250E-01	2.1100E-02	5.7100E-02	1.2080E-01	1.1570E-01	5.3500E-02	1.4300E-02
	Best	−7.7790E-01	−5.0900E-01	−7.9150E-01	−6.4820E-01	−6.1450E-01	−7.1270E-01	−7.9160E-01
	Worst	−3.0110E-01	−4.3220E-01	−5.8000E-01	−2.2600E-01	−2.5680E-01	5.3500E−02	−7.3300E−01
G3	Mean	−1.0000E+00	−9.8190E-01	−8.9930E-01	−9.8070E-01	−9.8400E-02	−9.6580E-01	−9.9360E-01
	Std	2.5096E-08	9.9000E-03	4.0650E-02	1.0600E-02	1.3470E-01	1.2300E-02	1.9000E-03
	Best	−1.0000E+00	−9.9100E-01	−9.9700E-01	−9.9610E-01	−5.7960E-01	−9.8630E-01	−9.9630E-01
	Worst	−1.0000E+00	−9.5580E-01	0.0000E+00	−9.5730E-01	0.0000E+00	−9.3630E-01	−9.8970E-01
G4	Mean	−3.0666E+04	−3.0666E+04	−3.0660E+04	−3.0662E+04	−2.9825E+04	−3.0652E+04	−3.0664E+04
	Std	1.7000E-03	4.9000E-03	3.1921E+00	2.0642E+01	2.5829E+02	7.8038E+00	1.1844E+00
	Best	−3.0666E+04	−3.0666E+04	−3.0665E+04	−3.0666E+04	−3.0153E+04	−3.0663E+04	−3.0665E+04
	Worst	−3.0665E+04	4.9000E-03	−3.0654E+04	−3.0552E+04	−2.8958E+04	−3.0628E+04	1.1844E+00
G5	Mean	5.2398E+03	5.3587E+03	5.2782E+03	5.4571E+03	5.7436E+03	5.2632E+03	5.1614E+03
	Std	9.6915E+01	2.3378E+02	1.0116E+02	2.8011E+02	4.1954E+02	4.8620E+01	1.1303E+01
	Best	5.1265E+03	5.1269E+03	5.1289E+03	5.1280E+03	5.1701E+03	5.1565E+03	5.1401E+03
	Worst	5.4664E+03	5.9295E+03	5.5960E+03	6.0631E+03	6.6413E+03	5.3114E+03	5.1869E+03
G6	Mean	−6.9618E+03	−6.9616E+03	−6.9179E+03	1.5639E+19	2.5558E+19	1.4141E+18	−6.9588E+03
	Std	2.3600E-02	1.9630E-01	2.3803E+01	4.5209E+19	3.9435E+19	7.7453E+18	2.2065E+00
	Best	−6.9618E+03	−6.9618E+03	−6.9734E+03	−6.9562E+03	−6.9496E+03	−6.9504E+03	−6.9607E+03
	Worst	−6.9617E+03	−6.9610E+03	−6.8502E+03	2.3980E+20	2.0342E+20	4.2423E+19	−6.9498E+03
G7	Mean	2.5207E+01	2.6543E+01	3.7482E+01	1.5192E+02	1.3137E+02	3.2642E+01	2.6184E+01
	Std	4.0710E-01	1.8443E+00	2.4819E+01	1.8031E+02	1.6180E+02	3.9615E+02	4.6660E-01
	Best	2.4380E+01	2.4390E+01	2.7835E+01	2.5275E+01	3.4038E+01	3.2642E+01	2.5374E+01
	Worst	2.7376E+01	3.1575E+01	1.3469E+02	6.0889E+02	7.7964E+02	9.6900E+02	2.7474E+01
G8	Mean	−8.4600E-02	1.2524E+20	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02
	Std	2.5600E-02	3.3375E+20	3.7948E-07	1.8937E-17	2.6006E-07	2.2121E-06	1.6094E-17
	Best	−9.5400E-02	−2.1700E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02
	Worst	−2.5500E-02	1.6954E+21	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02	−9.5800E-02
G9	Mean	6.8080E+02	6.8186E+02	6.8640E+02	6.8133E+02	7.2166E+02	7.1236E+02	6.8088E+02
	Std	1.0620E-01	1.2547E+00	5.8186E+00	7.1000E-01	2.9694E+01	4.8318E+01	7.5500E-02
	Best	6.8065E+02	6.8077E+02	6.8100E+02	6.8068E+02	6.8838E+02	6.8436E+02	6.8076E+02
	Worst	6.8112E+02	6.8669E+02	7.0999E+02	6.8335E+02	8.2469E+02	9.0186E+02	6.8113E+02

Table 5. *Cont.*

Func.	Metric	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
G10	Mean	7.8444E+03	8.2845E+03	8.0568E+03	9.0121E+17	1.0923E+19	8.4891E+03	8.1973E+03
	Std	3.4986E+02	4.5156E+02	4.0786E+02	2.1431E+18	2.6706E+19	3.5825E+02	3.9594E+02
	Best	7.0652E+03	7.2193E+03	7.2777E+03	7.0684E+03	9.6935E+03	7.7745E+03	7.6514E+03
	Worst	8.5464E+03	9.2693E+03	8.9547E+03	8.3416E+18	1.2393E+20	9.0926E+03	8.6863E+03
G11	Mean	7.5000E-01	7.5000E-01	7.5000E-01	7.5070E-01	7.5050E-01	7.5010E-01	7.5000E-01
	Std	1.7737E-05	5.9117E-05	1.5477E-05	6.7194E-04	1.0000E-03	8.0545E-05	1.0865E-05
	Best	7.5000E-01	7.5000E-01	7.5000E-01	7.5000E-01	7.5000E-01	7.5000E-01	7.5000E-01
	Worst	7.5010E-01	7.5020E-01	7.5010E-01	7.5270E-01	7.5460E-01	7.5030E-01	7.5010E-01
G12	Mean	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00
	Std	0.0000E+00	1.7076E-09	2.0459E-08	0.0000E+00	4.9676E-08	1.6854E-07	0.0000E+00
	Best	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00
	Worst	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00
G13	Mean	9.0910E-01	1.1229E+00	1.5451E+00	9.7310E-01	7.5399E+15	3.4936E+15	2.9115E+16
	Std	3.5220E-01	6.0810E-01	2.4258E+00	4.2510E-01	1.7540E+16	5.2593E+16	4.6794E+16
	Best	7.7400E-02	5.9700E-01	2.5760E-01	3.0590E-01	9.3930E-01	7.2668E+13	2.9424E+13
	Worst	1.9110E+00	3.5430E+00	1.0792E+01	2.9538E+00	6.4159E+16	5.2593E+16	2.1279E+17

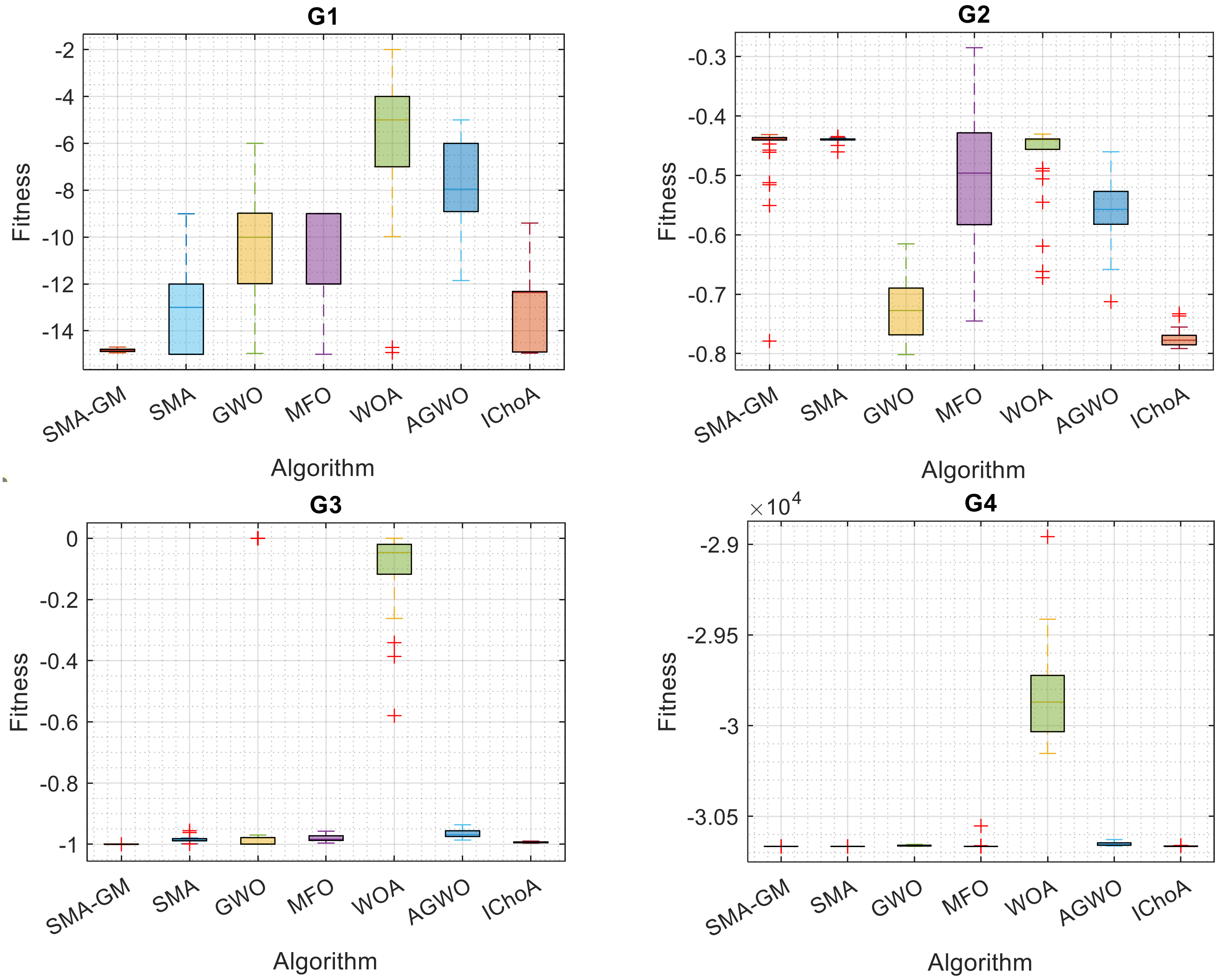

Figure 4. *Cont.*

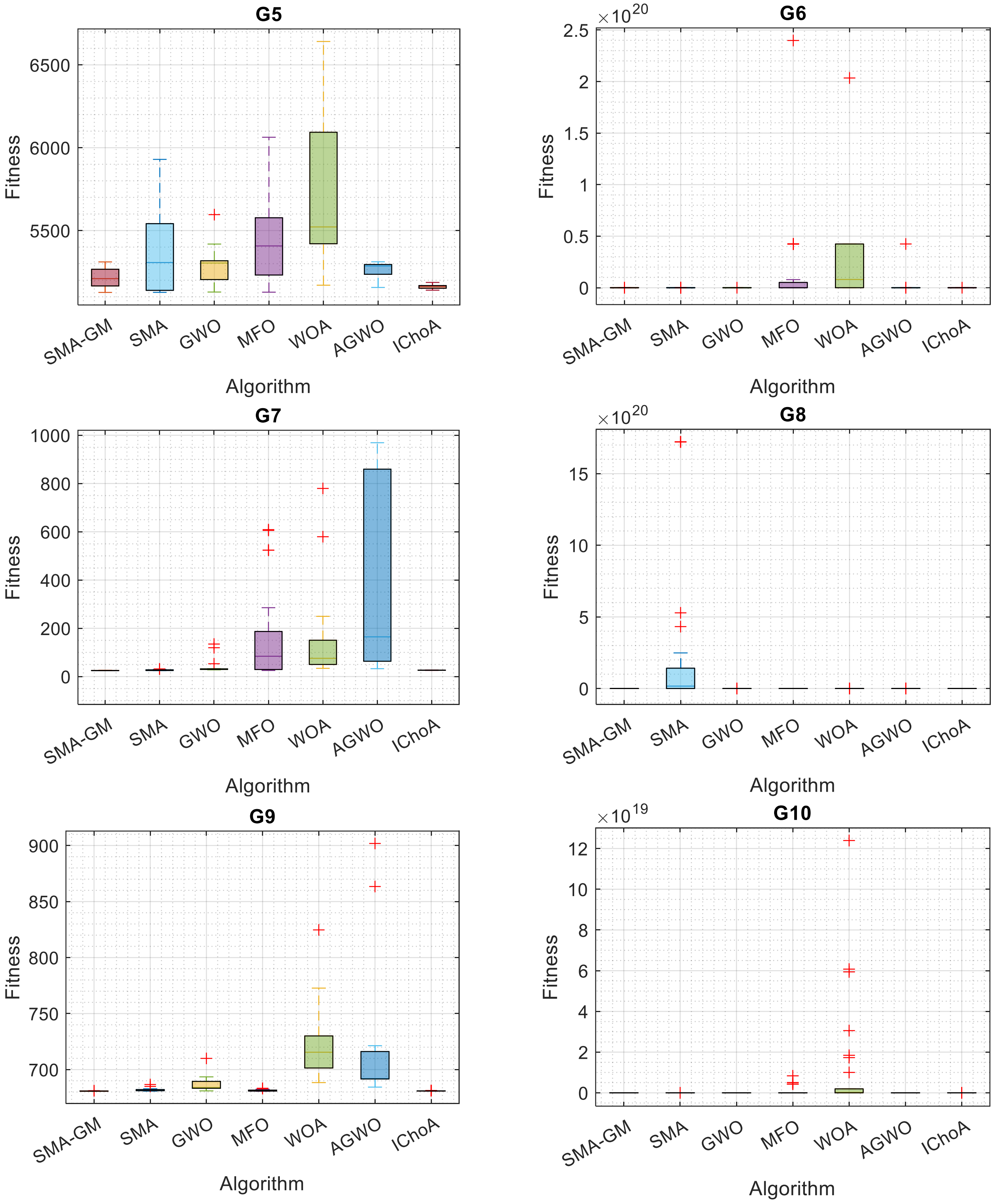

Figure 4. *Cont.*

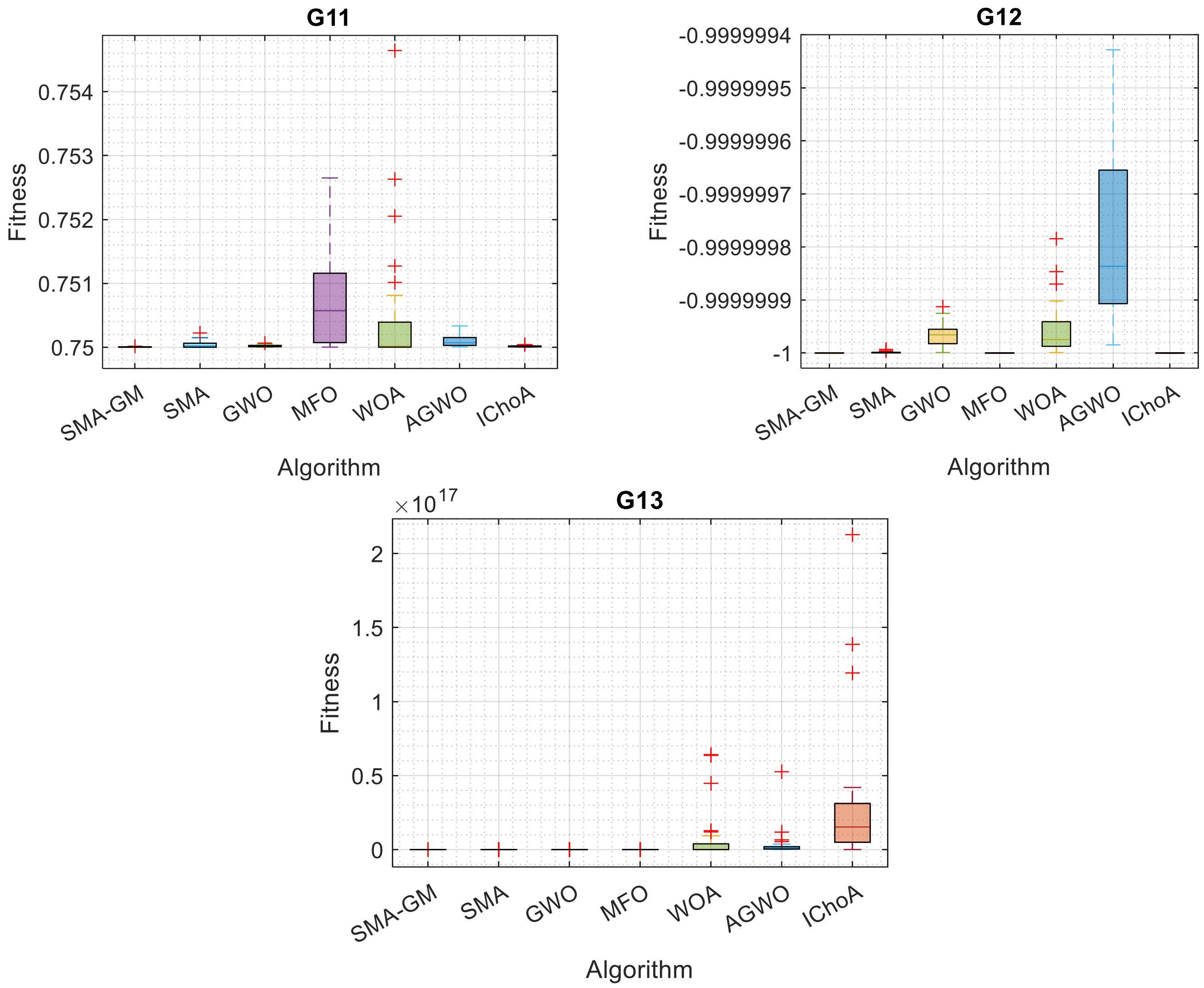

Figure 4. Boxplot of the SMA-GM algorithm and other algorithms on constrained functions.

5.4. Results and Discussions on CEC2022 Benchmark Functions

In this section, the most recent CEC2022 test functions are selected to further evaluate the performance of the proposed SMA-GM and compare it with other above-mentioned algorithms. The highly complex global optimization problems are simulated by the CEC 2022 test function [30]. All functions were tested in 20-dimensional space to guarantee the validity and fairness of the experimental results. In total, 12 test functions, 4 of which are classified as compositional, unimodal, essential, and hybrid, are included in Table 6 as the summary of CEC2022. The algorithm parameter values used were the same as those listed in Table 1. Table 7 presents the mean and standard deviation (Std) for 30 independent runs, illustrating the higher performance of SMA-GM for 10 out of the 12 test functions in the CEC-2022 test suite. By analyzing the results, it is clear that SMA-GM has satisfactory exploration and exploitation abilities to solve optimization problems. Compared with other algorithms, its optimization accuracy, solution stability, and adaptability to different functions have noticeable advantages and SMA-GM has gained more competitive results.

As seen in Figure 5, the convergent graphs are extracted from the CEC2022 test functions to thoroughly validate the SMA-GM algorithm's performance. The convergence curve shows that whereas other algorithms enter local optima too early, SMA-GM has a faster rate of convergence in the functions C1–C3 and C6–C12. In order to prevent stagnation into local minima, SMA-GM's performance first explored the majority of the search space. Later, it gradually switched to the exploitation operator to quickly reduce diversity during the second half of the optimization process, speeding convergence toward the most promising region so far found. Due to the search process variation, the SMA-GM performs well, which allows the algorithm to have a strong exploration and exploitation operator during the optimization process. The GM operator helps to preserve population diversity, prevents the system from becoming stuck in local optima, and speeds up the convergence towards the best-so-far solution.

Table 6. Description of CEC2022 functions.

No.	Type	Functions	Global Optima
C1	Unimodal function	Shifted and Full Rotated Zakharov Function	300
C2	Basic functions	Shifted and Full Rotated Rosenbrock's Function	400
C3		Shifted and Full Rotated Expanded Schaffer's f6 Function	600
C4		Shifted and Full Rotated Non-continuous Rastrigin's Function	800
C5		Shifted and Rotated Levy Function	900
C6	Hybrid functions	Hybrid Function 1 (N = 3)	1800
C7		Hybrid Function 2 (N = 6)	2000
C8		Hybrid function 3 (N = 5)	2200
C9	Composition Functions	Composition Function 1 (N = 5)	2300
C10		Composition Function 2 (N = 4)	2400
C11		Composition Function 3 (N = 5)	2600
C12		Composition Function 4 (N = 6)	2700
Search range: $[-100, 100]^{D}$			

Table 7. Comparison results of SMA-GM with other algorithms for CEC2022 functions.

Function	Measure	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
C1	Mean	3.0075E+02	3.0606E+02	1.5556E+04	3.7841E+04	2.6617E+04	1.8679E+04	6.3721E+03
	Std	1.0474E+00	1.1659E+01	5.3757E+03	2.2427E+04	8.5331E+03	4.6890E+03	2.1821E+03
C2	Mean	4.5264E+02	4.6056E+02	5.0443E+02	5.5281E+02	5.5932E+02	6.7119E+02	4.5511E+02
	Std	2.6936E+01	3.5103E+01	4.6753E+01	1.3298E+02	6.3667E+01	1.0402E+02	1.1528E+01
C3	Mean	6.0185E+02	6.0336E+02	6.2079E+02	6.0498E+02	6.6742E+02	6.4909E+02	6.0198E+02
	Std	1.2369E+00	3.2547E+00	1.0238E+01	2.8857E+00	1.2927E+01	8.6198E+00	3.1905E+00
C4	Mean	8.6863E+02	8.7890E+02	8.5632E+02	8.9231E+02	9.2433E+02	8.9681E+02	9.1443E+02
	Std	1.7006E+01	2.6378E+01	2.0991E+01	2.6189E+01	2.9938E+01	1.6206E+01	2.5496E+01
C5	Mean	1.4174E+03	1.6059E+03	1.1853E+03	2.8859E+03	3.7702E+03	2.2381E+03	2.1262E+03
	Std	4.2994E+02	5.2857E+02	2.8897E+02	1.0928E+03	1.2968E+03	5.3361E+02	6.6054E+02
C6	Mean	1.6826E+04	1.6887E+04	1.0582E+06	5.3890E+06	1.9337E+06	9.3540E+06	2.2690E+06
	Std	8.1372E+03	7.7127E+03	2.7053E+06	1.1969E+07	3.8576E+06	1.1970E+07	5.6906E+06
C7	Mean	2.0754E+03	2.0852E+03	2.0937E+03	2.1307E+03	2.2072E+03	2.1544E+03	2.0962E+03
	Std	3.0956E+01	4.5286E+01	4.1485E+01	5.5167E+01	7.1031E+01	4.5667E+01	2.8800E+01
C8	Mean	2.2442E+03	2.2840E+03	2.2783E+03	2.2724E+03	2.3070E+03	2.2744E+03	2.2603E+03
	Std	4.2107E+01	7.4723E+01	5.9580E+01	6.1209E+01	8.4745E+01	6.8001E+01	3.8452E+01
C9	Mean	2.4810E+03	2.4810E+03	2.5107E+03	2.5126E+03	2.5661E+03	2.5804E+03	2.4814E+03
	Std	1.9860E-01	2.1650E-01	2.5276E+01	3.3465E+01	4.9679E+01	3.0063E+01	2.2150E-01
C10	Mean	2.9457E+03	2.9854E+03	3.5612E+03	3.9601E+03	4.7196E+03	4.8970E+03	3.6656E+03
	Std	3.1960E+02	3.4560E+02	8.7356E+02	9.2908E+02	1.3148E+03	1.2388E+03	1.6121E+03
C11	Mean	2.9271E+03	2.9743E+03	3.5838E+03	4.1465E+03	3.6249E+03	4.5947E+03	3.0641E+03
	Std	1.1749E+02	1.1053E+02	3.6136E+02	8.5659E+02	6.5494E+02	6.5468E+02	1.9326E+02
C12	Mean	2.9458E+03	2.9495E+03	2.9769E+03	2.9565E+03	3.0552E+03	3.0823E+03	3.0141E+03
	Std	6.5767E+00	8.8285E+00	2.4279E+01	1.2255E+01	7.2826E+01	6.5608E+01	6.6373E+01

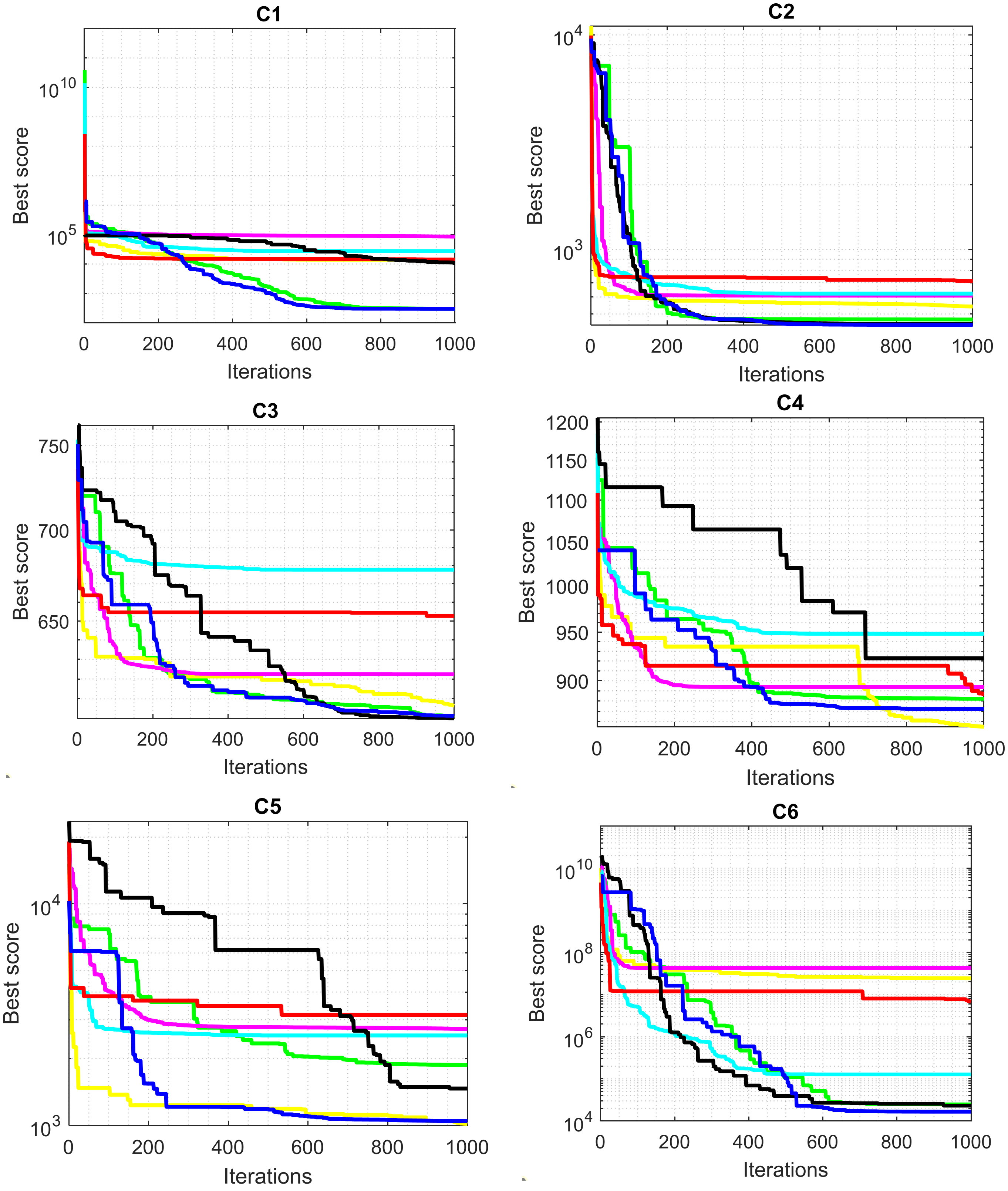

Figure 5. *Cont.*

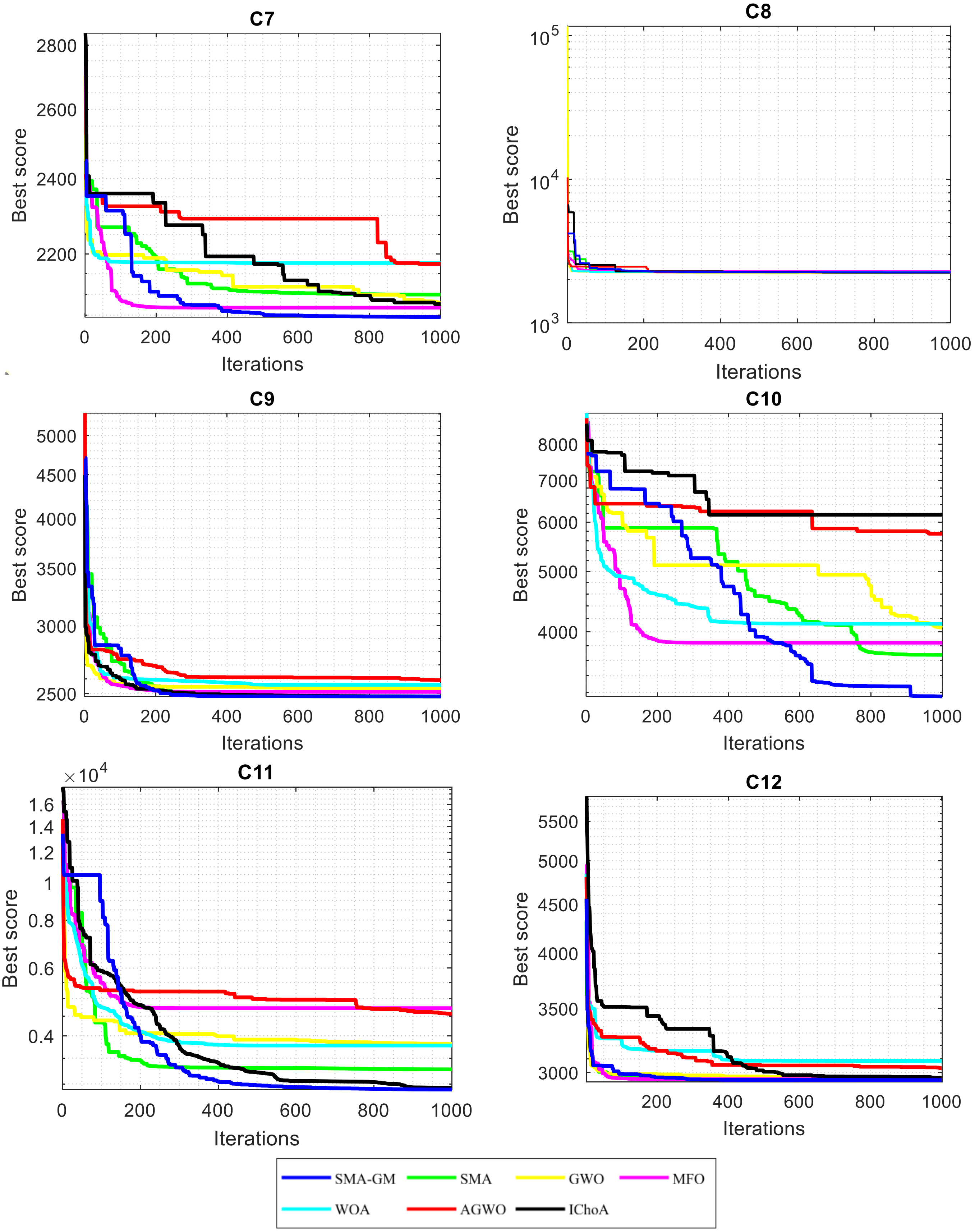

Figure 5. Convergence curves of CEC2022 benchmark functions.

5.5. Wilcoxon Rank Sum Test

The statistical analysis of the significant difference between the two methods can be conducted using the non-parametric Wilcoxon rank sum test [31]. The Wilcoxon rank sum test method is employed in this study to evaluate if the experimental findings of the SMA-GM algorithm and the counterpart algorithms differ significantly from one another. The results of each algorithm running independently 30 times on the 13 constrained functions (G1–G13) and CEC2022 test functions listed above provide the data utilized in the rank sum test. The rank sum test p-value between SMA-GM and each comparison algorithm is displayed in Table 8. The null hypothesis is rejected when the difference between the two algorithms is significant ($p < 0.05$). In Table 8, the NaN denotes the independent results obtained from SMA-GM and its corresponding optimizer are identical. The majority of p-values shown in Table 8 is less than 5%, which indicates that the alternative hypothesis is accepted. Hence, the outcome of SMA-GM is different from those of the other compared algorithms. Thus, SMA-GM is a robust optimizer, as verified by its capability to overcome SMA, GWO, MFO, WOA, AGWO, and IChoA, which are the most competent and highly cited metaheuristic optimizers.

Table 8. Wilcoxon rank sum test results (p-values) of SMA-GM with each algorithm for the constrained function and CEC2022 test function.

Func.	SMA	GWO	MFO	WOA	AGWO	IChoA
G1	2.7100E-02	5.5727E-10	8.0899E-06	5.3893E-10	3.0199E-11	3.8307E-05
G2	1.0300E-02	3.4742E-10	6.1000E-03	3.3000E-03	1.2541E-07	1.3289E-10
G3	3.0199E-11	2.9543E-11	3.0199E-11	3.0180E-11	3.0199E-11	3.0199E-11
G4	1.1000E-03	3.0199E-11	5.3195E-05	3.0199E-11	3.0199E-11	3.0199E-11
G5	2.8100E-02	8.0000E-03	1.6813E-04	2.0338E-09	8.7000E-03	8.5641E-04
G6	3.4971E-09	5.5727E-10	2.9822E-11	3.0199E-11	3.0199E-11	3.0199E-11
G7	8.1465E-05	3.0199E-11	2.3715E-10	3.0199E-11	3.0199E-11	1.8500E-08
G8	1.7769E-10	9.2340E-01	1.2118E-12	7.0100E-02	7.7310E-01	1.2118E-12
G9	8.1014E-10	5.4941E-11	2.2539E-04	3.0199E-11	3.0199E-11	3.8307E-05
G10	3.0939E-06	4.0000E-03	9.0595E-08	3.0199E-11	5.5329E-08	3.5923E-05
G11	2.8100E-02	1.9900E-02	2.1947E-08	1.8580E-01	4.8011E-07	3.7900E-01
G12	1.2118E-12	1.2118E-12	NaN	1.2118E-12	1.2118e-12	NaN
G13	3.1500E-02	2.4200E-02	7.9580E-01	1.4643E-10	3.0199E-11	3.0199E-11
C1	5.5999E-07	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
C2	3.5100E-02	3.3520E-08	2.1327E-05	3.4742E-10	3.3384E-11	2.9730E-01
C3	2.6100E-02	3.0939E-06	3.6897E-11	3.0199E-11	3.0199E-11	4.5100E-02
C4	4.5100E-02	8.2400E-02	2.8389E-04	1.5465E-09	3.8053E-07	1.6980E-08
C5	4.2100E-02	5.3700E-02	2.3897E-08	8.9934E-11	4.8011E-07	2.5974E-05
C6	7.2400E-02	1.6000E-03	9.9000E-03	3.0199E-11	3.0199E-11	3.0059E-04
C7	4.5100E-02	2.3200E-02	7.1988E-05	1.3289E-10	6.0720E-11	2.6384E-06
C8	8.0000E-03	7.6973E-04	1.4932E-04	2.1959E-07	3.4285E-04	5.9000E-03
C9	2.9200E-02	3.3384E-11	5.5282E-08	3.0199E-11	7.5527E-11	5.0757E-13
C10	3.5100E-02	5.8000E-03	8.1465E-05	4.4205E-06	3.1770E-01	2.1232E-06
C11	4.6800E-02	1.7769E-10	4.4440E-07	4.9752E-11	1.9460E-09	5.6073E-05
C12	3.6400E-02	2.6695E-09	1.9963E-05	3.6897E-11	9.5867E-18	4.3116E-12

6. Constrained Engineering Design Problem

The goal of developing and improving any stochastic search algorithm is to find solutions to real-world problems. In the preceding section, the SMA-GM efficacy was studied through numerical constrained experiments. This section evaluates the exploration and exploitation abilities in real-life situations using the SMA-GM algorithm. In this section, the SMA-GM algorithm is used to solve mechanical engineering design problems, such as the industrial refrigeration system (IRS) design problem, optimization of the alkylation unit procedure, welded beam, and tension/compression spring design problem.

6.1. Optimal Design of an Industrial Refrigeration System

Solving engineering design problems aims to provide the best possible design scheme while taking into account a variety of constraints. SMA-GM is compared with the metaheuristic algorithms mentioned above and the same parameter settings are listed in Table 1. Nowadays, energy saving and emission reduction work has become the focus of several fields. IRS accounts for a large proportion of energy consumption, so it is essential to optimize and control the IRS [32]. The optimal design of IRS is a highly complex mechanical engineering design problem, which has 14 design variables and 15 constraints. Below is an illustration of its mathematical formulation.

Minimize

$$\begin{aligned} f(u) = {} & 63098.88u_2u_4u_{12} + 5441.5u_2^2u_{12} + 115055.5u_2^{1.664}u_6 + 6172.27u_2^2u_6 \\ & + 63098.88u_1u_3u_{11} + 5441.5u_1^2u_{11} + 115055.5u_1^{1.664}u_5 + 6172.27u_1^2u_5 \\ & + 140.53u_1u_{11} + 281.29u_3u_{11} + 70.26u_1^2 + 281.29u_1u_3 + 281.29u_3^2 \\ & + 14437u_8^{1.8812}u_{12}^{0.3424}u_{10}u_{14}^{-1}u_1^2u_7u_9^{-1} + 20470.2u_7^{2.893}u_{11}^{0.316}u_1^2 \end{aligned}$$

subject to

$$\begin{gathered} g_1(u) = 1.524u_7^{-1} \le 1, \\ g_2(u) = 1.524u_8^{-1} \le 1, \\ g_3(u) = 0.07789u_1 - 2u_7^{-1}u_9 - 1 \le 0, \\ g_4(u) = 7.05305u_9^{-1}u_1^2u_{10}u_8^{-1}u_2^{-1}u_{14}^{-1} - 1 \le 0, \\ g_5(u) = 0.0833u_{13}^{-1}u_{14} - 1 \le 0, \\ g_6(u) = 47.136u_2^{0.333}u_{10}^{-1}u_{12} - 1.333u_8u_{13}^{2.1195} + 62.08u_{13}^{2.1195}u_{12}^{-1}u_8^{0.2}u_{10}^{-1} - 1 \le 0, \\ g_7(u) = 0.04771u_{10}u_8^{1.8812}u_{12}^{0.3424} - 1 \le 0, \\ g_8(u) = 0.0488u_9u_7^{1.893}u_{11}^{0.316} - 1 \le 0, \\ g_9(u) = 0.0099u_1u_3^{-1} - 1 \le 0, \\ g_{10}(u) = 0.0193u_2u_4^{-1} - 1 \le 0, \\ g_{11}(u) = 0.0298u_1u_5^{-1} - 1 \le 0, \\ g_{12}(u) = 0.056u_2u_6^{-1} - 1 \le 0, \\ g_{13}(u) = 2u_9^{-1} - 1 \le 0, \\ g_{14}(u) = 2u_{10}^{-1} - 1 \le 0, \\ g_{15}(u) = u_{12}u_{11}^{-1} - 1 \le 0, \end{gathered}$$

With bounds

$$0.001 \le u_i \le 5, i = 1, \ldots, 14.$$

Table 9 compares the results of SMA-GM with the other four optimizers to show which one is best at solving the IRS design problem. Table 10 presents the optimal constraint values. These findings show that SMA-GM outperforms the other optimizers and converges faster than the other optimizers, as shown in Figure 6.

Table 9. Optimal design of IRS.

Variable	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
u_1	0.001	0.001	0.001	0.001	0.0010253	0.001	0.0010023
u_2	0.001	0.001	0.0010429	0.001	0.001026	0.001	0.0011995
u_3	0.0010002	0.0010004	0.0010131	0.001	0.00573	0.001	0.0016858
u_4	0.0010007	0.001	0.0010678	0.001	0.0010258	0.0010188	0.009546
u_5	0.001	0.001	0.001041	0.001	0.0010253	0.0010128	0.0015728
u_6	0.001	0.001	0.0023688	0.001	0.0010252	0.001009	0.0013415
u_7	1.524	1.524	1.5244	1.524	1.524	1.5388	1.559
u_8	1.524	1.5241	1.5249	1.524	1.5319	1.5249	1.5677
u_9	5	5	4.9991	5	4.9996	4.9699	4.9931
u_{10}	2.0002	2.7834	2.0433	2.1899	2.3719	2.0121	2.4023
u_{11}	0.0010001	0.0019607	0.0035254	0.026399	0.012986	0.001	0.0030193
u_{12}	0.001	0.0019593	0.0030939	0.026399	0.0010252	0.001	0.0029526
u_{13}	0.0072936	0.011703	0.012496	0.034757	0.0010256	0.0065311	0.012581
u_{14}	0.087553	0.14049	0.14981	0.41726	0.011957	0.077074	0.13863
Optimal value	0.032216	0.036524	0.036843	0.054407	0.2536	0.035847	0.046458

Table 10. Constraint values of the IRS design problem.

Constraint	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
g1	−0.0000	−0.0000	0.0000	0	−0.2368	−0.0000	−0.0287
g2	−0.0000	0.0006	−0.0003	0	−0.0168	0.2793	−0.0420
g3	−7.5614	−7.5616	−7.5617	−7.5616	−6.0062	−0.0010	−7.3166
g4	−0.9788	−0.9774	−0.9899	−0.9935	−0.8470	0.0867	−0.9864
g5	−0.0002	−0.0001	−0.0056	0	−0.4332	−0.0000	−0.0513
g6	−0.9802	−0.9777	−0.9665	−0.8329	−0.9796	−0.0010	−0.9687
g7	−0.9389	−0.9389	−0.8976	−0.8123	−0.8982	−0.0010	−0.9021
g8	−0.9901	−0.9901	−0.9903	−0.9901	−0.9975	−0.0010	−0.9950
g9	−0.9807	−0.9807	−0.9917	−0.9807	−1.0000	−0.0010	−0.9967
g10	−0.9702	−0.9702	−0.9730	−0.9702	−0.9702	−0.0010	−0.9860
g11	−0.0004	0.0000	−0.0053	0	−0.9491	0.0974	−0.2135
g12	−0.9440	−0.9440	−0.9742	−0.9440	−0.9963	−0.0009	−0.9501
g13	−0.6000	−0.6000	−0.6000	−0.6000	−0.5999	1.9990	−0.5964
g14	−0.0010	−0.1145	−0.0049	−0.6000	0.0003	0.3814	−0.0725
g15	−0.0000	0	−0.1081	0	0	−0.0001	−0.3702

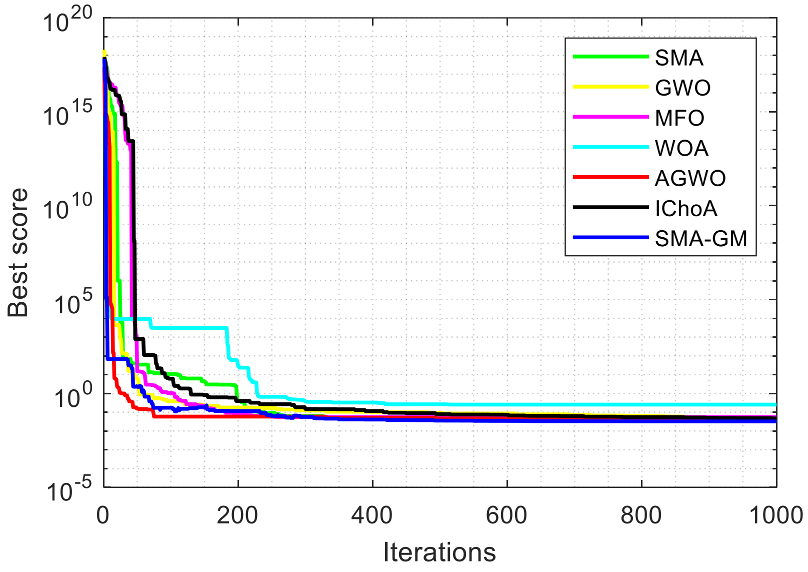

Figure 6. Convergence curve for IRS.

6.2. Optimal Operation of an Alkylation Unit

The optimal operation of an alkylation unit is very common in the petroleum industry [33]. In the alkylation process, the reactor is filled with the olefin feedstock (100% butane), pure isobutane recycling, and 100% isobutene supplement, along with the acid catalyst. The reactor product is then sent through a fractionator to separate the isobutene and alkane. Additionally, spent acid, the base product, is taken out of the reactor. The alkylation product is the objective function and the primary goal of this problem is to raise the octane number of the olefin feedstock in acidic conditions. This is a complex optimization problem with 14 constraints and 7 variables. Below is an illustration of its mathematical formulation.

Maximize

$$f(u) = 0.035u_1u_6 + 1.715u_1 + 10.0u_2 + 4.0565u_3 - 0.063u_3u_5$$

subject to

$$\begin{gathered}
g_1(u) = 0.0059553571u_6^2u_1 + 0.88392857u_3 - 0.1175625u_6u_1 - u_1 \leq 0, \\
g_2(u) = 1.1088u_1 + 0.1303533u_1u_6 - 0.0066033u_1u_6^2 - u_3 \leq 0, \\
g_3(u) = 6.66173269u_6^2 - 56.596669u_4 + 172.39878u_5 - 10000 - 191.20592u_6 \leq 0, \\
g_4(u) = 1.08702u_6 - 0.03762u_6^2 + 0.32175u_4 + 56.85075 - u_5 \leq 0, \\
g_5(u) = 0.006198u_7u_4u_3 + 2462.3121u_2 - 25.125634u_2u_4 - u_3u_4 \leq 0, \\
g_6(u) = 161.18996u_3u_4 + 5000.0u_2u_4 - 489510.0u_2 - u_3u_4u_7 \leq 0, \\
g_7(u) = 0.33u_7 + 44.333333 - u_5 \leq 0, \\
g_8(u) = 0.022556u_5 - 1.0 - 0.007595u_7 \leq 0, \\
g_9(u) = 0.00061u_3 - 1.0 - 0.0005u_1 \leq 0, \\
g_{10}(u) = 0.819672u_1 - u_3 + 0.819672 \leq 0, \\
g_{11}(u) = 24500.0u_2 - 250.0u_2u_4 - u_3u_4 \leq 0, \\
g_{12}(u) = 1020.4082u_4u_2 + 1.2244898u_3u_4 - 100000u_2 \leq 0, \\
g_{13}(u) = 6.25u_1u_6 + 6.25u_1 - 7.625u_3 - 100000 \leq 0, \\
g_{14}(u) = 1.22u_3 - u_6u_1 - u_1 + 1.0 \leq 0,
\end{gathered}$$

With bounds

$$\begin{gathered}
1000 \leq u_1 \leq 2000, 0 \leq u_2 \leq 100, 2000 \leq u_3 \leq 4000, 0 \leq u_4 \leq 100, \\
0 \leq u_5 \leq 100, 0 \leq u_6 \leq 20, 0 \leq u_7 \leq 200.
\end{gathered}$$

Table 11 shows that the SMA-GM algorithm performs better than other algorithms, signifying that it can maximize the alkylation product value and has a better result for the optimization of the alkylation process. The convergence curve for the optimal operation of the alkylation unit is shown in Figure 7 and Table 12 shows the constraint values. It can be seen from the Figure that the convergence performance of the proposed algorithm is not optimal at the initial stage. But at the later stage of the iteration, the SMA-GM algorithm can get rid of the local optima and continue exploring to improve the overall optimization performance.

Table 11. Optimal operation of an alkylation unit.

Variable	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
u_1	2000	2000	2000	2000	1999.9606	1706.8626	1991.0172
u_2	0	4.8302E-18	0	0	0	0	1.4796E-06
u_3	2571.4216	2473.1669	2457.5344	2820.406	3094.9607	2682.5542	2904.3814
u_4	0	0	0	0	0	0	6.1598E-07
u_5	58.139421	57.723041	57.663054	59.229379	61.102948	61.070298	61.106755
u_6	1.2386014	0.8260777	0.76253731	2.3850848	4.2767875	4.5800474	3.1824157
u_7	41.381011	40.379902	40.033248	44.584016	50.817011	50.391441	26.440249
Optimal value	−4529.1132	−4526.428	−4524.7028	−4513.7265	−4370.0026	−3761.7272	3.2702E+13

Table 12. Constraint values of the optimal operation of an alkylation unit.

Constraint	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
g1	−0.0000	0.0000	−0.0054	−0.0050	−0.0056	−0.0015	−0.0035
g2	−0.0053	−0.0054	−0.0003	−0.0009	−0.0001	−0.0025	−0.0009
g3	−0.0205	−0.0205	−0.0221	−0.0215	−0.0234	−0.0188	−0.0102
g4	−0.0000	−0.0000	0.0000	0.0000	0.0000	−0.0000	−0.0001
g5	0.0000	0.0000	0	0	0	0	0.0000
g6	−0.0000	−0.0000	0	0	0	0	−0.0000
g7	−0.0000	−0.0000	−0.0000	−0.0000	−0.0000	−0.0013	−0.0009
g8	−0.0000	−0.0000	−0.0000	−0.0000	−0.0000	0.0000	0.0000
g9	−0.0000	−0.0000	−0.0000	0.0000	−0.0000	−0.0001	−0.0000
g10	−0.1030	−0.1068	−0.1629	−0.1639	−0.1631	−0.0639	−0.0865
g11	0.0000	0.0000	0	0	0	0	0.0000
g12	−0.0000	−0.0000	0	0	0	0	−0.0000
g13	−8.6920	−8.5088	−3.2088	−4.1828	−3.5747	−9.9144	−9.0460
g14	−0.2092	−0.2385	−1.0865	−0.9306	−1.0279	−0.0136	−0.1525

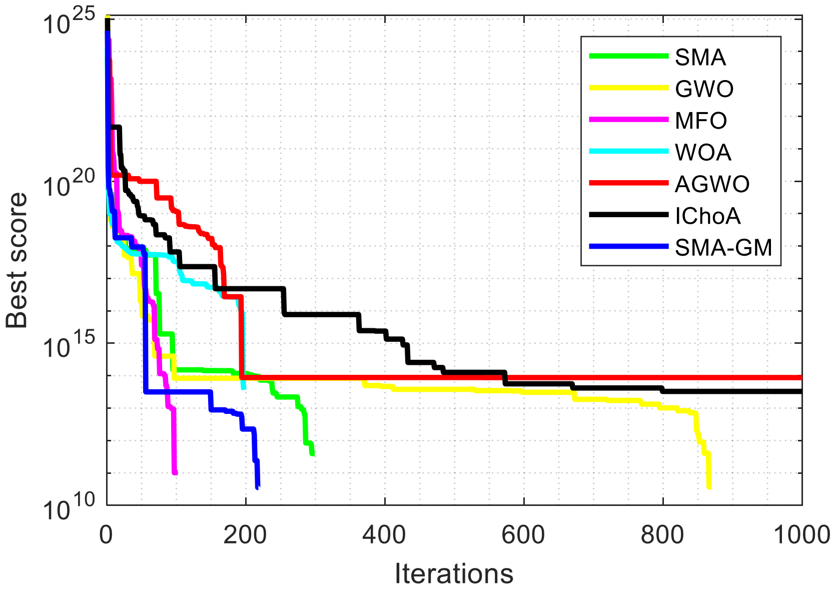

Figure 7. Convergence curve for the optimal operation of an alkylation unit.

6.3. Welded Beam Design

The objective of the problem is to construct a welded beam [33] with minimal cost under the bounds of buckling load (P_c), end deflection of the beam (δ), bending stress (θ), and shear stress (τ). It considers the weld thickness h, the joint length l, the height t of the beam, and the thickness b as variables and the mathematical formulation of this problem is shown as follows:

Consider

$$Variable\ u = [u_{1,}u_2, u_3, u_4] = [h,\ l,\ t,\ b]$$

Minimize

$$f(u) = 0.04811u_3u_4(u_2 + 14) + 1.10471u_1^2u_2$$

subject to

$$g_1(u) = u_1 - u_4 \leq 0,$$
$$g_2(u) = \delta(u) - \delta_{max} \leq 0,$$
$$g_3(u) = P \leq P_c(u),$$
$$g_4(u) = \tau_{max} \geq \tau(u),$$
$$g_5(u) = \sigma(u) - \sigma_{max} \leq 0,$$

where,

$$\tau = \sqrt{\tau'^2 + \tau''^2 + 2\tau'\tau''\tfrac{u^2}{2R}}, \tau'' = \tfrac{RM}{J},\ \tau' = \tfrac{P}{\sqrt{2}u_2u_1},\ M = p\left(\tfrac{u_2}{2} + L\right),$$
$$R = \sqrt{\tfrac{u_2^2}{4} + \left(\tfrac{u_1+u_3}{2}\right)^2}, J = 2\left(\left(\tfrac{u_2^2}{4} + \left(\tfrac{u_1+u_3}{2}\right)^2\right)\sqrt{2}u_1u_2\right),\ \sigma(u) = \tfrac{6PL}{u_4u_3^2},$$
$$\delta(u) = \tfrac{6PL^3}{Eu_4u_3^2}, P_c(u) = \tfrac{4.013Eu_3u_4^3}{6L^2}\left(1 - \tfrac{u_3}{2L}\sqrt{\tfrac{E}{4G}}\right), L = 14in, P = 6000lb,$$
$$E = 30.10^6psi, \sigma_{max} = 30,000psi, T_{max} = 13,600psi, G = 12.10^6psi, \delta_{max} = 0.25in.$$

With bounds

$$0.1 \leq u_3, u_2 \leq 10,\ 0.1 \leq u_4 \leq 2,\ 0.125 \leq u_1 \leq 2.$$

The optimal solution of SMA-GM and other counterparts on the welded beam problem are presented in Tables 13 and 14, which gives the corresponding constraint values of these algorithms for this problem. According to the comparative results, SMA-SM achieved optimum parameter values resulting in optimum cost function value 1.7249 with ideal parameter [0.20573, 3.4704, 9.037, 0.20573] for welded beam design problem. Figure 8 shows the convergence behavior of the SMA-GM and other algorithms for this problem.

Table 13. Optimal result of the welded beam design problem.

Variable	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
u_1	0.20573	0.20557	0.20555	0.20612	0.17825	0.2004	0.2057
u_2	3.4704	3.474	3.4724	3.4654	4.5602	3.5899	3.4719
u_3	9.037	9.0366	9.0506	9.028	8.9823	9.0493	9.0357
u_4	0.20573	0.20573	0.20572	0.20612	0.20823	0.20587	0.20577
Optimum cost	1.7249	1.7251	1.7271	1.7263	1.8302	1.7358	1.7252

Table 14. Constraint values of the welded beam design problem.

Constraint	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
g1	−0.0100	0.3094	−34.3397	0	−890.2884	−25.7247	9.9250
g2	−2.4395	5.0397	−41.6530	0	−194.3126	176.6754	10.3670
g3	−0.0000	0.0000	−0.0010	0. 0000	−0.0300	−0.0023	−0.0001
g4	−3.4329	−0.0000	−3.4279	−3.4272	−3.3184	−3.4287	−3.4325
g5	−0.0807	0. 0000	−0.0802	−0.0829	−0.0536	−0.0795	−0.0810
g6	−0.2355	0.0002	−0.2356	−0.2355	−0.2356	−0.2354	−0.2355
g7	−0.1228	0.0001	−39.7059	−169.2802	−245.6852	−70.7822	−24.0572

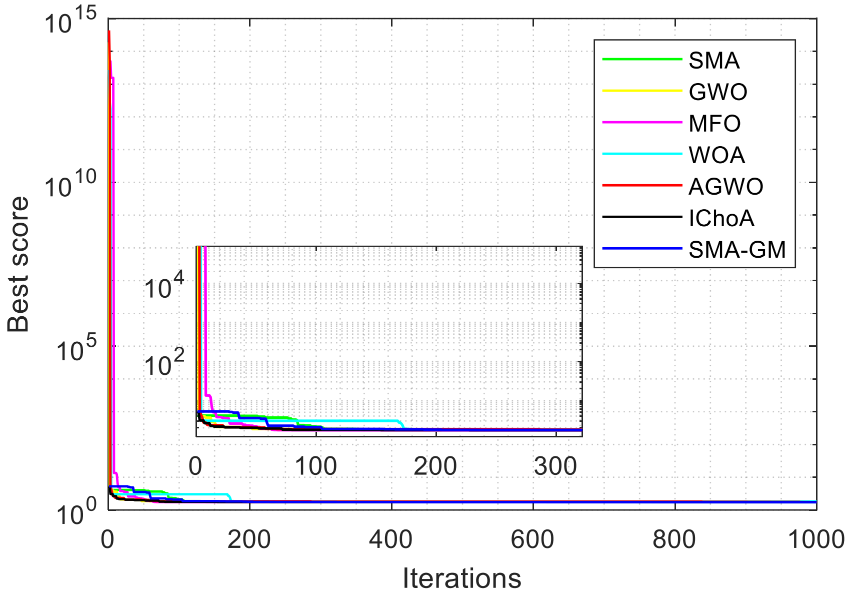

Figure 8. Convergence curve for the welded beam design problem.

6.4. Tension/Compression Spring Design Problem

The tension/compression spring problem is a classic structural engineering design problem [33], whose purpose is to minimize the weight of tension or compression spring under certain constraints such as shear stress, surge frequency, and deflection. To solve the problem, three core variables are needed: the diameter of the wire (d), mean value of the coil diameter (D), and number of active coils (N). The mathematical model of this problem is as follows:

Consider

$$\text{Variable } u = [u_1, u_2, u_3] = [d, D, N]$$

Minimize

$$f(u) = u_1^2 u_2 (2 + u_3)$$

subject to

$$g_1(u) = 1 - \frac{u_2^3 u_3}{71785 u_1^4} \le 0,$$

$$g_2(u) = \frac{4u_2^2 - u_1 u_2}{12566\left(u_2 u_1^3 - u_1^4\right)} + \frac{1}{5108 u_1^2} - 1 \le 0,$$

$$g_3(u) = 1 - \frac{140.45 u_1}{u_2^2 u_3} \le 0,$$

$$g_4(u) = \frac{u_1 + u_2}{1.5} - 1 \le 0,$$

with bounds

$$0.05 \le u_1 \le 2.00,\ 0.25 \le u_2 \le 1.30,\ 2.00 \le u_3 \le 15.0.$$

The results of SMA-GM and competitor algorithms in terms of achieving the optimal solution for the tension/compression spring design optimization problem are reported in Table 15. In addition, Table 16 provides the corresponding constraint values of these algorithms for this problem. As can be seen from Table 15, the SMA-GM algorithm not only performed the best out of all comparison algorithms but also obtained the solution closest to the optimal value of the tension/compression spring design problem. For qualitative analysis, the convergence plots were recorded and illustrated in Figure 9 and the figure simply proves the efficient convergence potential of the SMA-GM. Therefore, we can say that the SMA-GM algorithm not only has an excellent ability to solve complex engineering problems but also shows high efficiency and accuracy and obtains the optimal solution with the best parameter and constraint values.

Table 15. Optimal result of the tension/compression spring design problem.

Variable	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
u_1	0.0514206	0.0503801	0.0517674	0.0523125	0.0526395	0.05	0.0513004
u_2	0.350294	0.326035	0.358598	0.371903	0.380017	0.317405	0.347299
u_3	11.6757	13.3437	11.1885	10.4512	10.0432	14.0373	11.8796
Optimum weight	0.012667	0.012697	0.012674	0.012672	0.012681	0.012726	0.012686

Table 16. Constraint values of the tension/compression spring design problem.

Variable	SMA-GM	SMA	GWO	MFO	WOA	AGWO	IChoA
g_1	0	0.9303	0.0007	0	−0.0000	0.5120	−0.0873
g_2	0	−0.1657	−0.0012	−0.0000	0.0040	−0.1662	0.0022
g_3	−4.0409	−55.1800	−4.0544	−4.0828	−4.1181	−7.0128	−3.6005
g_4	−0.7322	−0.8000	−0.7265	−0.7172	−0.7121	−0.8000	−0.7486

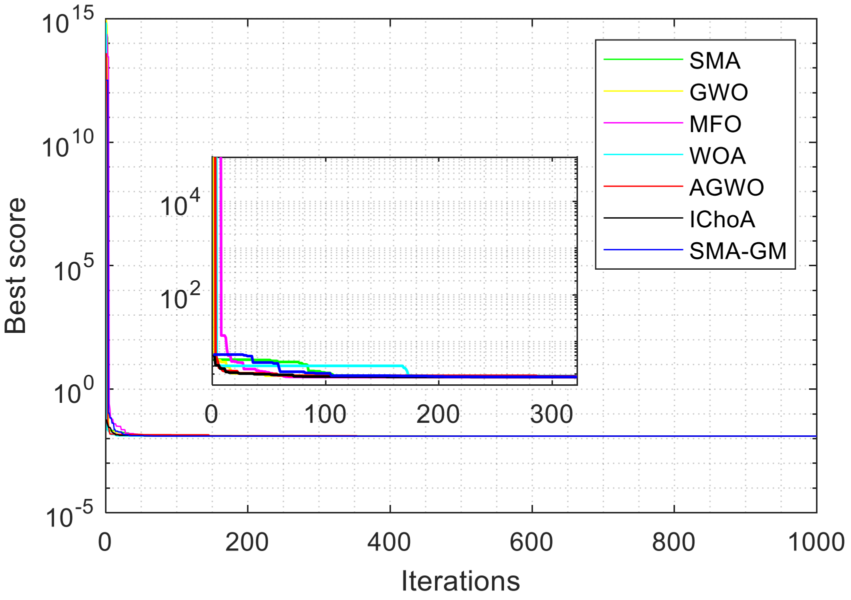

Figure 9. Convergence curve for the tension/compression spring design problem.

7. Conclusions

A novel algorithm named SMA-GM has been proposed in this paper. SMA-GM has been proposed to address the shortcomings of the original SMA algorithm, such as its tendency to trap into local optima and insufficient exploitation. In the proposed method for avoiding SMA's limitations, the exploitation phase of the original SMA has improved through the Gaussian mutation (GM) mechanism. The proposed SMA-GM performs exceptionally well in terms of convergence rate, stability, and optimization reliability compared to other metaheuristic algorithms. According to experiments conducted on the unconstrained, constrained, and CEC2022 functions and to further check the potential of SMA-GM, the complex constrained engineering problems show the superiority of the proposed algorithm as compared to other algorithms. To avoid the local optima, the GM strategy introduces a slight arbitrary variation into the group of search agents, improving convergence and enhancing exploitation potential while expanding the individual search space. Strengthening the global search and boosting the updated population is the primary goal of the proposed GM strategy. To guarantee that slime moulds with better fitness enter the next generation, the greedy selection mechanism was included. Convergence curves, diversity curves, and boxplot results showed that the proposed SMA-GM algorithm has efficiently enhanced the convergence accuracy, fitness performance, and the algorithm's stability.

The proposed SMA-GM has some limitations in practical application. If the scale of the problem is too large, difficulty could arise in selecting the optimal solution. However, like other stochastic algorithms, SMA-GM might not always reach the global optima for all benchmark functions and is susceptible to being trapped at local minima. The limitations of SMA-GM must be addressed to improve its ability to provide global solutions. By overcoming these challenges, SMA-GM can advance toward becoming a widely accepted and widely used state-of-the-art algorithm.

In future studies, more effective constraint processing methods must be developed and incorporated into other metaheuristic algorithms. Multi-objective optimization is also a highly challenging area. Future work can focus on combining the proposed algorithm with different techniques to handle multi-objective optimization problems, focusing more on the operators to analyze the diversity and striking a balance between exploration and exploitation. Consequently, the proposed algorithm can be applied to complex optimization problems involving many design variables. It is worthwhile to investigate whether the SMA-GM may be used for a more extensive variety of real-life problems in future research. This will contribute to further validating the algorithm's applicability to generate optimal solutions for a wide variety of optimization problems.

Author Contributions: G.T.: Conceptualization and Writing—original Draft, A.P.: Writing—review and editing and Supervision, N.M.: Methodology and Investigation, A.R.: Writing—review and editing, R.S.: Methodology, Supervision, Visualization, Writing—review and editing, Project Administration, and Funding Acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest regarding the publication of this research.

References

1. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In *MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995*; IEEE: Piscataway, NJ, USA, 1995; pp. 39–43.
2. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
3. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
4. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
5. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
6. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
7. Mirrashid, M.; Naderpour, H. Transit search: An optimization algorithm based on exoplanet exploration. *Results Control Optim.* **2022**, *7*, 100127. [CrossRef]
8. Schlüter, M.; Gerdts, M. The oracle penalty method. *J. Glob. Optim.* **2010**, *47*, 293–325. [CrossRef]
9. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime Mould Algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
10. Yuan, L.; Ji, J.; Liu, X.; Liu, T.; Chen, H.; Chen, D. An Improved Elite Slime Mould Algorithm for Engineering Design. *CMES-Comput. Model. Eng. Sci.* **2023**, *137*, 415–454. [CrossRef]
11. Naik, M.K.; Panda, R.; Abraham, A. An entropy minimization based multilevel colour thresholding technique for analysis of breast thermograms using equilibrium slime mould algorithm. *Appl. Soft Comput.* **2021**, *113*, 107955. [CrossRef]
12. Houssein, E.H.; Mahdy, M.A.; Blondin, M.J.; Shebl, D.; Mohamed, W.M. Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst. Appl.* **2021**, *174*, 114689. [CrossRef]
13. Yu, C.; Heidari, A.A.; Xue, X.; Zhang, L.; Chen, H.; Chen, W. Boosting quantum rotation gate embedded slime mould algorithm. *Expert Syst. Appl.* **2021**, *181*, 115082. [CrossRef]

14. Chauhan, S.; Vashishtha, G.; Kumar, A. A symbiosis of arithmetic optimizer with slime mould algorithm for improving global optimization and conventional design problem. *J. Supercomput.* **2022**, *78*, 6234–6274. [CrossRef]
15. Juan, Z.; Zheng-Ming, G.; Wu, S. The improved slime mould algorithm with Levy flight. *J. Phys. Conf. Ser.* **2020**, *1617*, 012033.
16. Abualigah, L.; Diabat, A.; Elaziz, M.A. Improved slime mould algorithm by opposition-based learning and Levy flight distribution for global optimization and advances in real-world engineering problems. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 1163–1202. [CrossRef]
17. Örnek, B.N.; Aydemir, S.B.; Düzenli, T.; Özak, B. A novel version of slime mould algorithm for global optimization and realworld engineering problems: Enhanced slime mould algorithm. *Math. Comput. Simul.* **2022**, *198*, 253–288. [CrossRef]
18. Yu, C.; Cai, Z.; Ye, X.; Wang, M.; Zhao, X.; Liang, G.; Li, C. Quantum-like mutation-induced dragonfly-inspired optimization approach. *Math. Comput. Simul.* **2020**, *178*, 259–289. [CrossRef]
19. Zhang, X.; Xu, Y.; Yu, C.; Heidari, A.A.; Li, S.; Chen, H.; Li, C. Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst. Appl.* **2020**, *141*, 112976. [CrossRef]
20. Chen, H.; Zhang, Q.; Luo, J.; Xu, Y.; Zhang, X. An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [CrossRef]
21. Luo, J.; Chen, H.; Xu, Y.; Huang, H.; Zhao, X. An improved grasshopper optimization algorithm with application to financial stress prediction. *Appl. Math. Model.* **2018**, *64*, 654–668. [CrossRef]
22. Wu, S.; Heidari, A.A.; Zhang, S.; Kuang, F.; Chen, H. Gaussian bare-bone slime mould algorithm: Performance optimization and case studies on truss structures. *Artif. Intell. Rev.* **2023**, *56*, 9051–9087. [CrossRef]
23. Bäck, T.; Schwefel, H.P. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1993**, *1*, 1–23. [CrossRef]
24. Ma, C.; Huang, H.; Fan, Q.; Wei, J.; Du, Y.; Gao, W. Grey wolf optimizer based on Aquila exploration method. *Expert Syst. Appl.* **2022**, *205*, 117629. [CrossRef]
25. Preeti; Kaur, R.; Singh, D. Dimension learning based chimp optimizer for energy efficient wireless sensor networks. *Sci. Rep.* **2022**, *12*, 14968. [CrossRef] [PubMed]
26. Qtaish, A.; Albashish, D.; Braik, M.; Alshammari, M.T.; Alreshidi, A.; Alreshidi, E.J. Memory-based Sand Cat Swarm Optimization for Feature Selection in Medical Diagnosis. *Electronics* **2023**, *12*, 2042. [CrossRef]
27. Duan, Y.; Liu, C.; Li, S.; Guo, X.; Yang, C. Manta ray foraging and Gaussian mutation-based elephant herding optimization for global optimization. *Eng. Comput.* **2023**, *39*, 1085–1125. [CrossRef]
28. Deb, K. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 311–338. [CrossRef]
29. Runarsson, T.P.; Yao, X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **2000**, *4*, 284–294. [CrossRef]
30. Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Chakrabortty, R.K. Light spectrum optimizer: A novel physics-inspired metaheuristic optimization algorithm. *Mathematics* **2022**, *10*, 3466. [CrossRef]
31. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]
32. Azizi, M.; Talatahari, S.; Giaralis, A. Optimization of engineering design problems using atomic orbital search algorithm. *IEEE Access* **2021**, *9*, 102497–102519. [CrossRef]
33. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.