



Stability analysis based parameter tuning of Social Group Optimization

Junali Jasmine Jena¹ · Samarendra Chandan Bindu Dash² · Suresh Chandra Satapathy¹

Received: 19 April 2021 / Accepted: 30 January 2022 / Published online: 23 February 2022
© The Author(s) 2022

Abstract

Swarm-based optimization algorithms have been popularly used these days for optimization of various real world problems but sometimes it becomes hard to estimate the associated characteristics due to their stochastic nature. To ensure a steady performance of these techniques, it is essential to have knowledge about the range of variables, in which a particular algorithm always provides stable performance and performing stability analysis of an algorithm can help in providing some knowledge regarding the same. Many researchers have performed the stability analysis of several optimization algorithms and analyzed their behavior. Social Group Optimization (SGO) is a newly developed algorithm which has been proven to yield promising results when applied to many real world problems but in literature no work can be found on stability analysis of SGO. In this paper, Von Neumann stability analysis approach has been used for performing stability analysis of Social Group Optimization (SGO) to analyze the behavior of its algorithmic parameters and estimate the range in which they always give stable convergence. The results obtained have been supported by sufficient experimental analysis and simulated using eight benchmark function suite along with their shifted and rotated variations which prove that the algorithm performs better within the stable range and hence convergence is ensured.

Keywords Stability analysis · Social Group Optimization · Evolutionary optimization · Parameter tuning

Introduction

In many real life applications, many complex optimization problems are faced which needs effective solutions. These functions have multiple maxima, minima and saddle points, thus gradient descent/ascent based techniques fail to find the proper solution or most likely lead to a local minima or maxima. For finding the global best value efficient searching of the entire search space becomes necessary. There are several categories of algorithm that fulfill this need. The prominent of them is swarm optimization in which the search space is filled with swarm members with random initial solutions. These members interact with each other (defined by the algorithm), through a series of exploration and

exploitation, finally attaining the global best location. Some examples of such algorithms are Particle Swarm Optimization (PSO) [1] which emulates flocking behavior of birds, Artificial Bee Colony (ABC) [2] which emulates behavior of bees, Social Group Optimization (SGO) [3] which emulates human learning pattern of, Grey Wolf Algorithm (GWA) [5] which emulates hunting patterns of Grey Wolves etc. Social Group Optimization (SGO) is a fairly new addition to the shelf. Nonetheless it is very promising addition as it has better convergence rate as compared to many evolutionary algorithms [4]. It has been successfully applied to segmentation of skin melanoma images [6], segmentation of brain MRI images [7], task scheduling in cloud [8], brain tumor evaluation tool [9], automated detection of COVID-19 infection [10], Transformer fault analysis [16], Antenna array synthesis [17], short-term hydrothermal scheduling [18], structural health monitoring in civil engineering [19], solving Travelling Salesman problem [20], solving multi-objective problems [21] and many more.

One essential requirement to be kept in mind while working with these heuristic algorithms is to ensure that the error values remain bounded in a region and don't explode. To achieve this requirement, stability analysis of an optimiza-

✉ Suresh Chandra Satapathy
suresh.satapathyfcs@kiit.ac.in

Junali Jasmine Jena
junali.jenafcs@kiit.ac.in

Samarendra Chandan Bindu Dash
samarendra109@gmail.com

¹ KIIT (Deemed to be University), Bhubaneswar, India

² Informatica, Bangalore, India

tion algorithm is essential as it provides the stable range of parameters that prevents the errors from growing beyond boundary making the algorithm converge faster to provide desired solution. In many of the works found in literature, it is observed that the efficacy of an algorithm is claimed by experimentation, but that experimental analysis is subject to that specific problem or situation. Stability analysis is such a method which analyzes the robustness of a system and 'Von Neumann' method is a suitable method which have been popularly used for this task for swarm optimization methodologies. Various such works can be found in the literature. Gopal et al. [11] have worked on stability analysis of PSO, Bansal et al. [12] in their work performed stability analysis of Differential Evolution, Nair et al. [13] analyzed the stability of Artificial Bee Colony, Farivar et al. [14] have reported on stability analysis of Gravitational Search Optimization, Biswas et al. [15] have analyzed the stability of Bacteria Foraging Optimization etc. To the best of authors' knowledge the stability analysis of SGO algorithm is not yet attempted. In this paper, authors have addressed the problem of stability of SGO through 'Von Neumann' stability criterion and have found stable ranges of parameters for the algorithm.

The rest of the paper is organized as follows. In the second section introduces the SGO algorithm and effects of its update equations on swarm members, in the third section provides details of the Von Neumann Stability Analysis procedure and explains stability analysis in context of swarm optimization. The mathematical analysis of SGO algorithm is carried out in the fourth section and the final section presents the results of simulation based experiments to demonstrate the simulation based find ups.

Social Group Optimization

Social Group Optimization (SGO) algorithm is a population based distributed optimization algorithm that simulates human learning pattern to iteratively search and reach the global maximum or minimum. It has two distinct phases i.e. improving phase and acquiring phase that execute sequentially in each iteration to exploit and explore the search region respectively.

Initialization

The initial position of the each member in the population is generated by

$$A_{i,j} = A_{j_{\min}} + \mu (A_{j_{\max}} - A_{j_{\min}}), \quad (2.1)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, D,$$

Here A_i denotes position of the i th member in the D dimensional space. $A_{i,j}$ denotes the j th dimension of the

position vector. $A_{j_{\max}}$ $A_{j_{\min}}$ are the upper and the lower bounds of j th parameter. N is the population size and D is the dimensionality of the optimization problem under consideration.

Improving phase

In this phase each swarm member X_i is updated as per Eq. (2.2),

$$A_{\text{new}_{i,j}} = c * A_{\text{old}_{i,j}} + r * (\text{gbest}_j - A_{\text{old}_{i,j}}) \quad (2.2)$$

Here c is the self introspection factor and r is a random number. Then $A_{\text{new}_{i,j}}$ is compared with $A_{i,j}$ and the one who provides better performance is accepted as value of $A_{i,j}$. This equation has a clumping effect on the swarm members as the values of $A_{i,j}$ will be remapped randomly in the connected areas defined by the 4 line segments given in Eqs. (2.3a) to (2.3d) where $r \in [r_n, r_x]$. Eqs. (2.3a) and (2.3b) represent the update equation in improving phase (Eq. (2.2)) when value of r is r_x and r_n respectively. Equations (2.3c) and (2.3d) represent the boundary condition x_{\min} and x_{\max} defined by the optimization problem.

$$y = (c - r_x) * x + r_x * \text{gbest}, \quad (2.3a)$$

$$y = (c - r_n) * x + r_n * \text{gbest}, \quad (2.3b)$$

$$x = x_{\min}, \text{ and } x = x_{\max} \quad (2.3c)$$

$$y = x_{\min}, \text{ and } y = x_{\max} \text{ (due to boundary constraint)} \quad (2.3d)$$

If we observe Eqs. (2.3a) and (2.3b), they intersect at the point $c * \text{gbest}$ as shown in Fig. 1a. Thus, $c * \text{gbest}$ is the cluster centroid. The cluster centroid is formed on the line connecting the origin and gbest . So smaller c value always forms the cluster closer to the origin. From Fig. 1a it is clear that for any value of X_i in the range, there is always a possibility that it gets mapped to $c * \text{gbest}$ (Can be verified by taking a vertical cross section of the grey area shown in Fig. 1a) and the farther a point is from the centroid on y -axis, the range of values of X_i which gets mapped to the value decreases. As the vertical cross-section of the probable area (Grayed portion in Fig. 1a) indicates probability of y falling in the region, most points will clump around the cluster centroid. In Fig. 1, the graphical representation of transformation done by Improving phase is shown. The Fig. 1a shows the area defined by the Eqs. (2.3a)–(2.3d) and how the range of X_i is reduced along each dimension (projection on x -axis shows the initial range, and projection on y -axis shows the range after improving phase). The Fig. 1b shows a 2D visualization of clustering of X_i after improving phase.

The spread of the cluster roughly can be determined by the angle between the lines denoted by ϕ . It is defined by

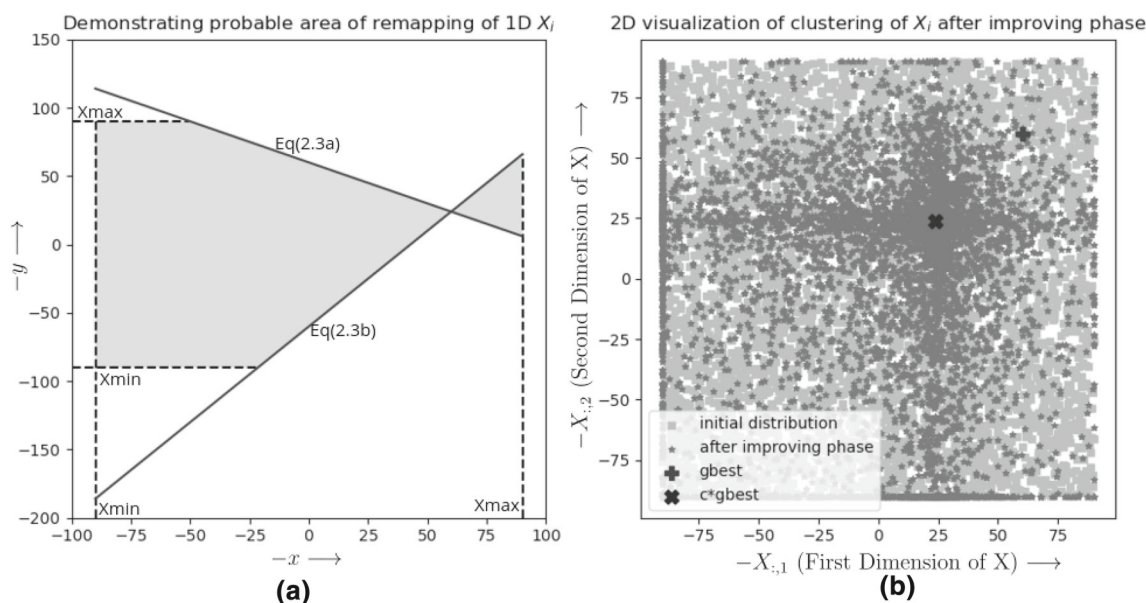


Fig. 1 **a** Mapping of X_i in improving phase, **b** clustering in improving phase after first iteration (created with $c, r_n, r_x, x_{min}, x_{max}, gbest$ values as 0.4, -1, 1, -90, 90, 60 respectively)

Eqs. 2.3a and 2.3b. It is controlled by values of c and r as

$$\tan(\phi) = \frac{m_1 - m_2}{1 + m_1 m_2}, \text{ where } m_1, m_2 \text{ are slopes of lines}$$

defined by Eqs. (12.3a) and (12.3b) respectively

$$\implies \tan(\phi) = \frac{(c - r_n) - (c - r_x)}{1 + (c - r_n) * (c - r_x)}$$

$$\implies \tan(\phi) = \frac{r_x - r_n}{1 + c^2 - (r_x + r_n) * c + r_n r_x}$$

Hence the angle can be found as

$$\phi = \tan^{-1} \left(\frac{r_x - r_n}{1 + c^2 - (r_x + r_n) * c + r_n r_x} \right) \tag{2.4}$$

Acquiring phase

In acquiring phase, each swarm member A_i moves in the search space with respect to $gbest$ and another randomly selected swarm member A_s . The update equation is given by

if $f(A_i) < f(A_s)$ **then**

$$A_{new_{i,:}} = A_{old_{i,:}} + r_1 * (A_{old_{i,:}} - A_{s,:}) + r_2 * (gbest - A_{old_{i,:}})$$

else

$$A_{new_{i,:}} = A_{old_{i,:}} + r_1 * (A_{s,:} - A_{old_{i,:}}) + r_2 * (gbest - A_{old_{i,:}})$$

end if

Here A_s is a member of the swarm s.t. $i \neq s$ and $r_1, r_2 \geq 0$. The update equations in acquiring phase can be written as

$$A_{new_i} = A_i + (r_2 \pm r_1) \left(\frac{r_2 * gbest \pm r_1 * A_s}{r_2 \pm r_1} - A_i \right)$$

$$\implies A_{new_i} = A_i + r_n (A_n - A_i),$$

where $A_n = \frac{r_2 * gbest \pm r_1 * A_s}{r_2 \pm r_1}$ and $r_n = r_2 \pm r_1$

So A_i updates its position with respect to a vector which is weighted mean of A_s and the $gbest$. Figure 2 shows the direction of movement of A_i in acquiring phase.

So every swarm member updates itself by moving towards or away from another swarm member while moving itself slightly closer to $gbest$.

Stability analysis

The stability analysis is an important requirement of numerical estimation schemes involving partial differential equations. In the pre-text context the stability reflects that the total error of a numeric scheme remains bounded. In other words, if some error is introduced it would remain under a bound and would not explode or increase beyond a limit. The Von Neumann stability analysis, which uses Fourier Decomposition, is an important technique, which is chosen to analyse the stability of a scheme.

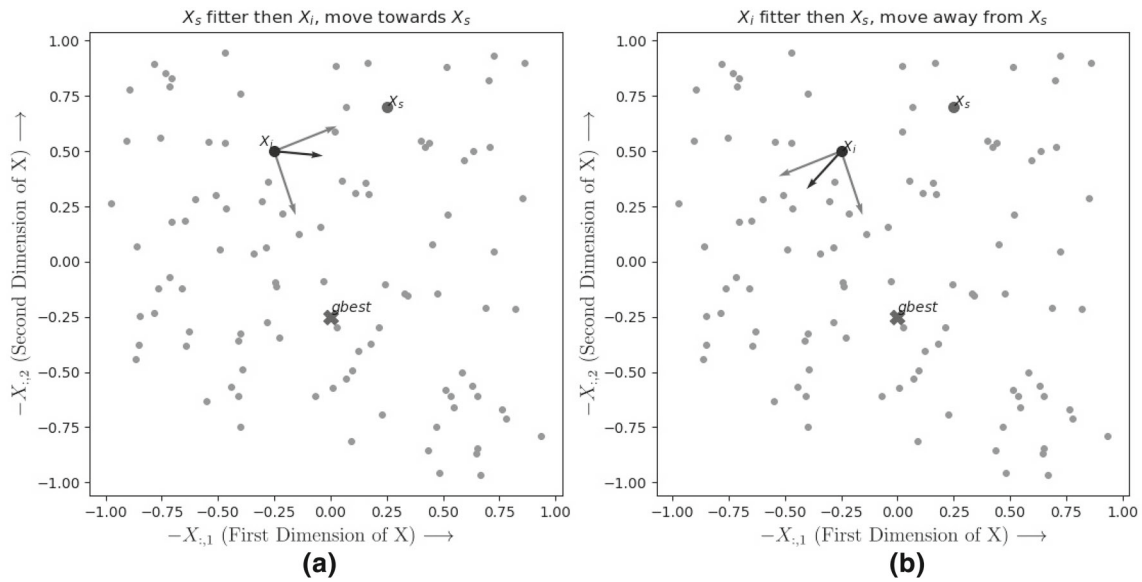


Fig. 2 **a** New Position of X_i is towards X_s . **b** New position of X_i is further from X_s

von Neumann stability analysis

Consider the partial differential equation represents a system X ,

$$\frac{\delta X}{\delta \tau} + a \frac{\delta X}{\delta \chi} = 0 \tag{3.1}$$

a function of χ and τ , which evolves through space (χ) and time(τ). The numeric estimate of the corresponding partial differential equation is given by red

$$\frac{X(\tau + \Delta\tau, \chi) - X(\tau - \Delta\tau, \chi)}{2 * \Delta\tau} + a * \frac{X(\tau, \chi + \Delta\chi) - X(\tau, \chi - \Delta\chi)}{2 * \Delta\chi} = 0 \tag{3.2}$$

Equation (3.2) shows numeric estimate of Eq. (3.1) The Fourier series expansion of $X(\tau, \chi)$ can be written as

$$X(\tau, \chi) = \sum_{\substack{i \in 1, \dots, b_1, \\ t \in 1, \dots, b_2}} A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi} \tag{3.3}$$

where $\chi = i \Delta\chi$, $\tau = t \Delta\tau$. $\Delta\chi$ and $\Delta\tau$ represent the intervals that were used to sample values along the χ and τ axis.

Let $\hat{X}(t, i)$ represent an estimate of $X(\tau, \chi)$ with an error $\epsilon_{t,i}$ as shown in Eq. (3.4).

$$X(\tau, \chi) = \hat{X}(t, i) + \epsilon_{t,i} \tag{3.4}$$

Here we want to learn whether the error associated with the estimate grows or shrinks with each iteration. So we assume that the error is associated with some components of the

Fourier expansion of X . We replace X in the numeric estimate is replaced by its fourier component to check whether contribution of a single fourier component increases or diminishes over time. If it diminishes then the error associated with the estimate decreases with time too. If the relation between any Fourier coefficient in t th iteration with the corresponding Fourier coefficient in $(t + 1)$ th iteration can be represented by the Eq. (3.5) and $|g(\alpha_m)| \leq 1$ then contribution of each Fourier component decreases with each iteration. So any error associated with it will also diminish over time and we will conclude that the numeric scheme is stable.

$$A_m e^{-J\beta_m(t+1)\Delta\tau} = A_m e^{-J\beta_m t \Delta\tau} * g(\alpha_m) \tag{3.5}$$

Stability Analysis in context of swarm optimization

In swarm optimization algorithms the update equation is the form

$$A_i^{t+1} = c * A_i^t + f(A_i^t, A_r^t) \tag{3.6}$$

Here A_r is another member of the swarm besides A_i . A_i interacts with another member of the swarm to update it's value over each iteration. To determine if it is stable we would want to know if some error gets introduced in an iteration then it diminishes or increases with each subsequent iteration.

For the stability analysis A is modelled as a continuous variable across space and time, where each iteration and each member of the swarm is taken as samples at $\tau = t \Delta\tau$ and $\chi = i \Delta\chi$. i.e.

$$A_i^t = A(t \Delta\tau, i \Delta\chi) = A(\tau, \chi)$$

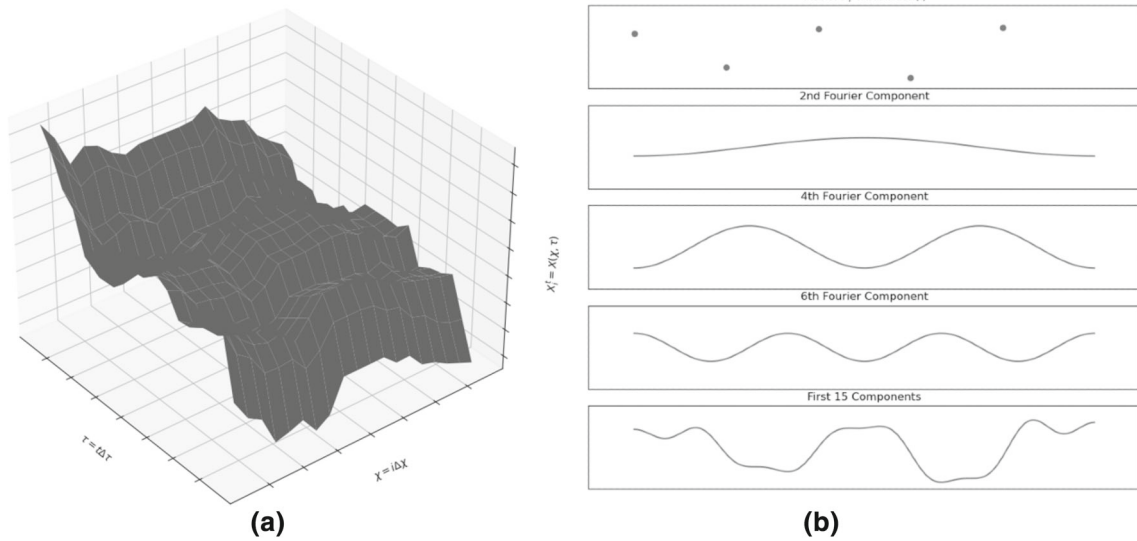


Fig. 3 **a** Representation of A as continuous variable from its grid samples. **b** Fourier decomposition of A_i

So all the distinct members of the swarm and each subsequent iteration is modelled as a single continuous function.

This allows us to find a Fourier series corresponding to A_i^t as Eq. (3.7)

$$A_i^t = A(t\Delta\tau, i\Delta\chi) = A(\tau, \chi) = \sum_{i \in 1, \dots, b_1, t \in 1, \dots, b_2} A_m e^{-j\beta_m t \Delta\tau} e^{j\alpha_m i \Delta\chi} \quad (3.7)$$

Now we can do von Neumann stability analysis on the update equation (Eq. (3.6)) by replacing A_i^t with its Fourier component $A_m e^{-j\beta_m t \Delta\tau} e^{j\alpha_m i \Delta\chi}$. We would get an equation of the form

$$A_m e^{-j\beta_m (t+1)\Delta\tau} = A_m e^{-j\beta_m t \Delta\tau} * g(\alpha_m) \implies e^{-j\beta_m \Delta\tau} = g(\alpha_m)$$

If $e^{-j\beta_m \Delta\tau}$ lies inside the unit circle in the Hilbert Space (complex space) then with each update the contribution of a single Fourier component would diminish over time. So the condition of stability is given as

$$|g(\alpha_m)| \leq 1 \implies |e^{-j\beta_m \Delta\tau}| \leq 1 \quad (3.8)$$

Optimization functions

Table 1 lists eight standard objective functions to be optimized along with their shifted and rotated version. For this the steps mentioned in the CEC 2014 [22] is followed. The functions are mentioned under the column objective function of Table 1 and are expressed as a function of z . But for better control and more variety we define z as a function of x . So

we can shift, scale and rotate x vectors before passing them into the functions.

The mappings from x to z are defined under the column z . Different mappings from x to z are defined to generalize the properties as stated-

Range of A : $x_j \in [-R, R]$

Global Best value location: $gbest = o$

and M is a $D \times D$ rotation matrix. In this paper we will keep $R = 100$ as per CEC 2014 [22] and we will use different values of M, o and D to test the algorithm.

Stability analysis of SGO algorithm

Approach for a solution

In SGO algorithm we have two phases that operate sequentially in a single epoch which gives the updated value of A_{new_i} for the next epoch. The equations governing the improving phase is,

$$A_{imp_{i,j}} = c * A_{i,j} + r * (gbest_j - A_{i,j}) \quad (4.1)$$

and acquiring phase is,

$$A_{acq_{i,j}} = A_{imp_{i,j}} \pm r_1 * (A_{s,j} - A_{imp_{i,j}}) + r_2 * (gbest_j - A_{imp_{i,j}}) \quad (4.2)$$

A_{acq_i} the value of A_{new_i} for the next iteration. The update equation can be written as

$$\implies A_{new_{i,j}} = (c - r) * (1 \mp r_1 - r_2) A_{i,j} \pm r_1 * A_{s,j} + (r_2 + (1 \mp r_1 - r_2) * r) gbest_j \quad (4.3)$$

Table 1 List of optimization functions

Name	z	Objective function
Sphere	$z = M * (x - o)$	$\text{Min } f_1(x) = \sum_{j=1}^D z_j^2$
Rosenbrock	$z = M * \left(\frac{2.048(x-o)}{R} + 1 \right)$	$\text{Min } f_2(x) = \sum_{j=1}^{D-1} (100(z_{j+1} - z_j^2)^2 + (z_j - 1)^2)$
Ackley	$z = M * \left(\frac{32(x-o)}{R} \right)$	$\text{Min } f_3(x) = -20 * \exp \left(-0.2 * \sqrt{\frac{1}{D} \sum_{j=1}^D z_j^2} \right) - \exp \left(\frac{1}{D} \sum_{j=1}^D \cos(2\pi z_j) \right) + 20 + e$
Griewank	$z = M * \left(\frac{600(x-o)}{R} \right)$	$\text{Min } f_4(x) = \sum_{j=1}^D \frac{z_j^2}{4000} - \prod_{j=1}^D \cos \left(\frac{z_j}{\sqrt{j}} \right) + 1$
Rastrigin	$z = M * \left(\frac{5.12(x-o)}{R} \right)$	$\text{Min } f_5(x) = 10D + \sum_{j=1}^D (z_j^2 - 10\cos(2\pi z_j))$
Alpine	$z = M * \left(\frac{10(x-o)}{R} \right)$	$\text{Min } f_6(x) = \sum_{j=1}^D (z_j \sin(z_j) + 0.1 * z_j)$
Sum of powers	$z = M * \left(\frac{(x-o)}{R} \right)$	$\text{Min } f_7(x) = \sum_{j=1}^D z_j^{j+1} $
Zakharov	$z = M * \left(\frac{10(x-o)}{R} \right)$	$\text{Min } f_8(x) = \sum_{j=1}^D z_j^2 + \left(\sum_{j=1}^D 0.5jz_j \right)^2 + \left(\sum_{j=1}^D 0.5jz_j \right)^4$

Equation (4.3) is a four dimensional equation w.r.t to the parameters c, r, r_1, r_2 . This equation is very difficult to solve as von Neumann analysis won't provide enough inequalities to perfectly define ranges of all the parameters. Moreover we will get equations where range of c, r is dependent on r_1, r_2 and vice versa, which is not desirable as both improving and acquiring phases are independent of each other. But we can use a simple trick to simplify these issues. In Eq. (4.1), let's assume $A_{\text{imp},j} = A_{i,j}^{t+1}$ and replace $A_{i,j}^t$ and $A_{i,j}^{t+1}$ with the corresponding Fourier component. We would get an equation of the form,

$$A_m e^{-J\beta_m(t+1)\Delta\tau} = A_m e^{-J\beta_m t \Delta\tau} * g(\alpha_{m1}) \tag{4.4}$$

Doing the same for acquiring phase we would get,

$$A_m e^{-J\beta_m(t+1)\Delta\tau} = A_m e^{-J\beta_m t \Delta\tau} * g(\alpha_{m2}) \tag{4.5}$$

Fourier representation of Eq. (4.3) can be expressed as,

$$A_m e^{-J\beta_m(t+1)\Delta\tau} = A_m e^{-J\beta_m t \Delta\tau} * g(\alpha_{m1}) * g(\alpha_{m2}) \tag{4.6}$$

and the corresponding condition for stability is

$$|g(\alpha_{m1}) * g(\alpha_{m2})| \leq 1$$

But if

$$|g(\alpha_{m1})| \leq 1$$

and

$$|g(\alpha_{m2})| \leq 1$$

are satisfied, then the condition of stability will be automatically satisfied. Therefore the stability analysis of

improving phase and acquiring phase independently can be taken up to obtain the ranges for c, r, r_1, r_2 .

Stability analysis

Improving phase

The Improving Phase has 2 parameters that govern the equation, c and r . c is the self introspection parameter which is constant, i.e. its value stays the same throughout the algorithm. r is a random number. The update equation is given in Eq. (4.7)

$$A_{\text{new},i,j} = c * A_{\text{old},i,j} + r * (g\text{best}_j - A_{\text{old},i,j}) \tag{4.7}$$

By replacing $A_{\text{new},i,j}$ and $A_{\text{old},i,j}$ with $A_{i,j}^{t+1}$ and $A_{i,j}^t$ respectively, Eq. (4.7) can be rewritten as Eq. (4.8).

$$A_{i,j}^{t+1} = c * A_{i,j}^t + r * (g\text{best}_j - A_{i,j}^t) \tag{4.8}$$

In the update equation the dimensions are independent of each other, we can drop j and assume it to be an one dimensional problem without loss of generality. So the Eq. (4.8) reduces to Eq. (4.9).

$$A_i^{t+1} = c * A_i^t + r * (g\text{best} - A_i^t) \tag{4.9}$$

Here the superscript t denotes the value at t th iteration. For simplifying our calculations we can take $g\text{best}$ as a constant. This decision is justified by the fact that after some initial iterations the value of $g\text{best}$ updates only occasionally through the algorithm run.

As $g\text{best}$ is a constant we can remove the term from the calculations for stability analysis as there are no errors associated with a constant value. Mathematically, the m th Fourier

component of a constant term is 0 as the Fourier series of the constant is represented by the constant itself.

Replacing A_i^t with its m th Fourier component $A_m e^{-J\beta_m t \tau} e^{J\alpha_m i \Delta \chi}$ (where J is the imaginary unit) in the Eq. (4.9) an equation similar to the form Eq. (3.5) is obtained.

$$A_m e^{-J\beta_m(t+1)\Delta\tau} e^{J\alpha_m i \Delta\chi} = (c - r) * A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi}$$

$$\implies e^{J\beta_m \Delta\tau} = (c - r)$$

As per von neumann stability criterion(given in Eq. (3.8)) $|e^{J\beta_m \Delta\tau}| \leq 1$. Upon solving the inequality we get,

$$|c - r| \leq 1$$

$$\implies -1 \leq c - r \leq 1$$

$$\implies c - 1 \leq r \leq c + 1$$

We obtain the range of r as defined in Eq. (4.10). The range of r is dependent on c .

$$c - 1 \leq r \leq c + 1 \tag{4.10}$$

Slope: For $r_n = c - 1$ and $r_x = c + 1$ as $r \in [c - 1, c + 1]$, it can be found that

$$r_x - r_n = 2 \tag{4.11a}$$

$$r_x + r_n = 2c \tag{4.11b}$$

$$r_x r_n = c^2 - 1 \tag{4.11c}$$

Putting these values defined in Eqs. (4.11a)–(4.11c) in Eq. (2.4), we get

$$\phi = \tan^{-1} \left(\frac{2}{1 + c^2 - 2c^2 + c^2 - 1} \right) \tag{4.12}$$

$$\phi = \frac{\pi}{2} \tag{4.13}$$

Therefore, if we use the complete stable range for improving phase, the angle of spread has to be $\pi/2$.

Experimental results To verify the findings, the acquiring phase is removed from SGO algorithm and only the improving phase is considered. The stable range of r against the three schemes of unstable ranges is compared. They are

1. $r \in [c + 1, c + 3]$ (r all positive)
2. $r \in [-c - 3, -c - 1]$ (r all negative)
3. $r \in [-c - 2, -c - 1] \cup [c + 1, c + 2]$

Rastrigin, Rosenbrock and Alpine function are used for the testing with $D = 30$, number of swarm members i.e. $N = 50$ with 300 epochs (Or $50 * 300 = 15,000$ function evaluations). Testing is performed in 3 phases.

- M, o as identity matrix and null vector respectively. Average taken over 30 iterations.
- M as identity matrix and o as a random vector whose value will be changed after each 10th iteration. Average taken over 100 iterations.
- M as random rotation matrix and o as a random vector. Their values will be changed after each 12th iteration. Average and Standard deviation taken over 120 iterations.

Figures 4, 5 and 6 show the results of the 3 phases mentioned above. From the figures it can be observed that, as c value increases the stable range performs better than the other 3 unstable schemes.

Acquiring phase

The acquiring phase is governed by two parameters i.e. r_1 and r_2 . In acquiring phase, if A_i is fitter than A_s , then A_i is updated as

$$A_{\text{new}i,:} = A_{\text{old}i,:} + r_1 * (A_{\text{old}i,:} - A_{s,:}) + r_2 * (\text{gbest} - A_{\text{old}i,:}) \tag{4.14}$$

else it is updated as

$$A_{\text{new}i,:} = A_{\text{old}i,:} + r_1 * (A_{s,:} - A_{\text{old}i,:}) + r_2 * (\text{gbest} - A_{\text{old}i,:}) \tag{4.15}$$

Here A_s is a candidate solution such that $i \neq s$ and $r_1, r_2 \geq 0$. We start with Eq. (4.15). The equation can be rewritten as

$$A_{i,:}^{t+1} = A_{i,:}^t + r_1 * (A_{s,:}^t - A_{i,:}^t) + r_2 * (\text{gbest} - A_{i,:}^t) \tag{4.16}$$

Here the superscript t denotes the value at t th iteration. Replacing

$$A_s = A_{i \pm a}, \text{ where } a \in \{\pm 1, \pm 2, \dots\} \text{ and } 1 \leq i \pm b \leq N$$

as A_s is a member of the swarm and $s \neq i$, we get

$$A_{i,:}^{t+1} = A_{i,:}^t + r_1 * (A_{i \pm a,:}^t - A_{i,:}^t) + r_2 * (\text{gbest} - A_{i,:}^t) \tag{4.17}$$

In the update equation the dimensions are independent of each other. So it can be assumed as an one dimensional problem without loss of generality. Then the Eq. (4.17) can be rewritten as

$$A_i^{t+1} = A_i^t + r_1 * (A_{i \pm a}^t - X_i^t) + r_2 * (\text{gbest} - A_i^t) \tag{4.18}$$

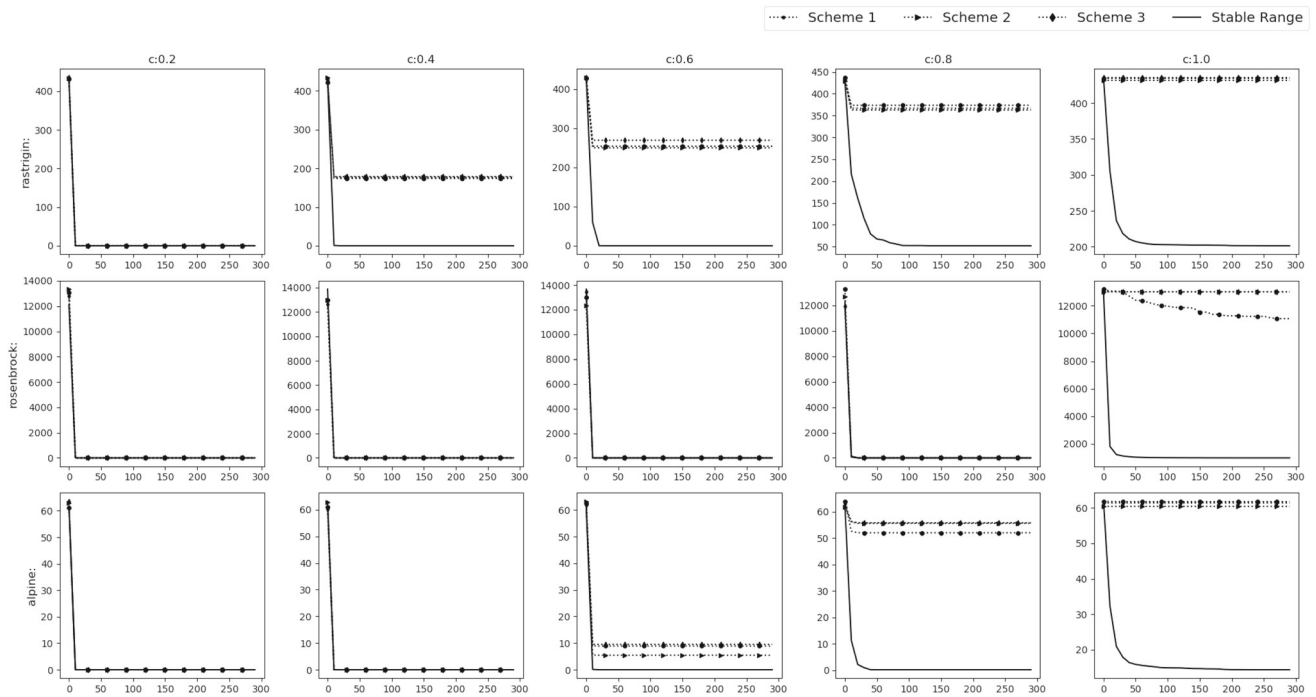


Fig. 4 Experimental results of testing of the improving phase with normal functions

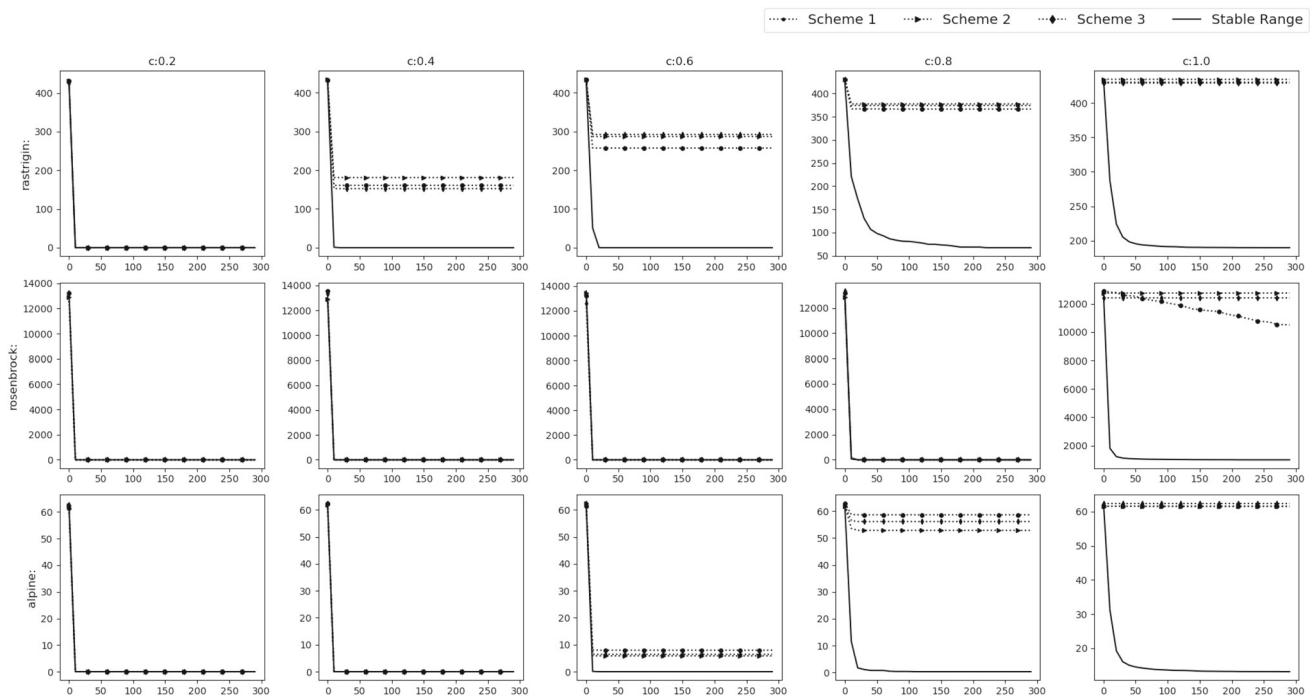


Fig. 5 Experimental results of testing of the improving phase with shifted functions

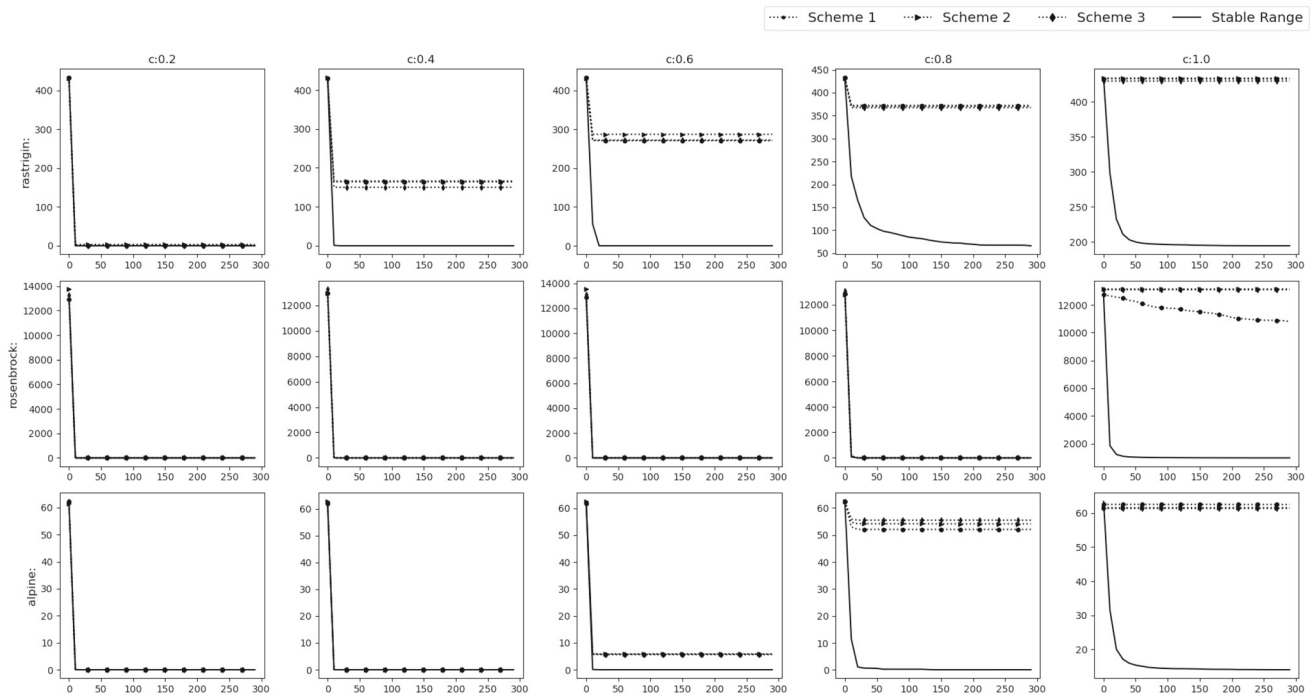


Fig. 6 Experimental results of testing of the improving phase with shifted+rotated functions

Similar to improving phase, g_{best} can be removed from the stability analysis calculations of acquiring phase, as it being a constant won't contribute to the errors.

Replacing A'_i with its m th Fourier component $A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi}$ we get

$$\begin{aligned}
 A_m e^{-J\beta_m(t+1)\Delta\tau} e^{J\alpha_m i \Delta\chi} &= (1 - r_2) * \\
 A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi} &+ r_1 * (A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m(i\pm a)\Delta\chi} \\
 - A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi}) &\quad (4.19)
 \end{aligned}$$

Upon simplifying we get,

$$\begin{aligned}
 A_m e^{-J\beta_m(t+1)\Delta\tau} e^{J\alpha_m i \Delta\chi} &= (1 - r_2) * A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi} + r_1 \\
 &* (A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m(i\pm a)\Delta\chi} - A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi}) \\
 \implies A_m e^{-J\beta_m(t+1)\Delta\tau} e^{J\alpha_m i \Delta\chi} &= (1 - r_1 - r_2) * A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m i \Delta\chi} + r_1 \\
 &* A_m e^{-J\beta_m t \Delta\tau} e^{J\alpha_m(i\pm a)\Delta\chi} \\
 \implies e^{-J\beta_m \Delta\tau} &= (1 - r_1 - r_2) + r_1 * e^{J\theta}, \\
 \text{where } \theta &= \alpha_m(\pm a)\Delta\chi
 \end{aligned}$$

As per Von Neumann stability criterion, $|e^{J\beta_m \Delta\tau}| \leq 1$. So,

$$\begin{aligned}
 \implies |(1 - r_1 - r_2) + r_1 * e^{J\theta}| &\leq 1 \\
 \implies \sqrt{(1 - r_1 - r_2 + r_1 \cos\theta)^2 + (r_1 \sin\theta)^2} &\leq 1 \\
 \implies \sqrt{(1 - r_2)^2 - 4r_1(1 - r_1 - r_2)\sin^2\left(\frac{\theta}{2}\right)} &\leq 1 \\
 \implies (1 - r_2)^2 &\leq 1 + 4r_1(1 - r_1 - r_2)\sin^2\left(\frac{\theta}{2}\right)
 \end{aligned}$$

Thus we obtain the Eq. (4.20). The inequality applies for all values of $\sin^2\left(\frac{\theta}{2}\right)$.

$$(1 - r_2)^2 \leq 1 + 4r_1(1 - r_1 - r_2)\sin^2\left(\frac{\theta}{2}\right) \quad (4.20)$$

Substituting $\sin^2\left(\frac{\theta}{2}\right) = 0$ in Eq. (4.20) we get,

$$\begin{aligned}
 (1 - r_2)^2 &\leq 1 \\
 \implies -1 &\leq 1 - r_2 \leq 1 \\
 \implies 0 &\leq r_2 \leq 2
 \end{aligned}$$

Hence the range of r_2 is obtained as

$$0 \leq r_2 \leq 2 \quad (4.21)$$

Putting $\sin^2\left(\frac{\theta}{2}\right) = 1$ in Eq. (4.20) we get,

$$\begin{aligned} (1 - r_2)^2 &\leq 1 + 4r_1(1 - r_1 - r_2) \\ \implies ((1 - r_2 - r_1) + r_1)^2 - 4r_1(1 - r_1 - r_2) &\leq 1 \\ \implies (1 - r_2 - 2r_1)^2 &\leq 1 \\ \implies -1 \leq 1 - r_2 - 2r_1 &\leq 1 \\ \implies -r_2 \leq 2r_1 \leq 2 - r_2 \end{aligned}$$

Thus the Eq. (4.22) defining relation between r_1 and r_2 is obtained as

$$-r_2 \leq 2r_1 \leq 2 - r_2 \tag{4.22}$$

But there is an additional condition that $r_1 \geq 0$. By considering this condition, we get the triangular region with boundaries defined by Eqs. (4.23a)–(4.23c).

$$r_1 = 0 \tag{4.23a}$$

$$r_2 = 0 \tag{4.23b}$$

$$r_2 + 2r_1 = 2 \tag{4.23c}$$

But as $X_{s,j} - X_{i,j}$ can be negative, r_1 can be replaced with $-r_1$. The region obtained by replacing with $-r_1$ has boundaries defined by

$$r_1 = 0 \tag{4.24a}$$

$$r_2 = 0 \tag{4.24b}$$

$$r_2 + 2r_1 = 2 \tag{4.24c}$$

So the complete region for r_1 and r_2 , as shown in Fig. 7a, is equal to the union of the regions defined by Eqs. (4.23a)–(4.23c) and Eqs. (4.24a)–(4.24c). For ease of calculation, this region has been approximated by the rectangular region bounded by the constraint $0 \leq r_1 \leq 1$ and $0 \leq r_2 \leq 1$ as shown in Fig. 7b.

The calculations for Eq. (4.14) is similar and by following the same steps which was done for Eq. (4.15), we obtain the equation as

$$\begin{aligned} e^{-J\beta_m \Delta\tau} &= (1 + r_1 - r_2) - r_1 * e^{J\theta}, \text{ where} \\ \theta &= \alpha_m(\pm a)\Delta\chi \end{aligned} \tag{4.25}$$

As per von neumann stability criterion $|e^{J\beta_m \Delta\tau}| \leq 1$. So,

$$\begin{aligned} \implies |(1 + r_1 - r_2) - r_1 * e^{J\theta}| &\leq 1 \\ \implies (1 - r_2)^2 &\leq 1 - 4r_1(1 + r_1 - r_2)\sin^2\left(\frac{\theta}{2}\right) \end{aligned}$$

Thus we obtain the inequality

$$(1 - r_2)^2 \leq 1 + 4r_1(1 - r_1 - r_2)\sin^2\left(\frac{\theta}{2}\right) \tag{4.26}$$

The inequality applies for all values of $\sin^2\left(\frac{\theta}{2}\right)$. Putting $\sin^2\left(\frac{\theta}{2}\right) = 0$ in Eq. (4.26) gives us the same condition as in Eq. (4.22). Putting $\sin^2\left(\frac{\theta}{2}\right) = 1$ in Eq. (4.20) we get,

$$\begin{aligned} (1 - r_2)^2 &\leq 1 - 4r_1(1 + r_1 - r_2) \\ \implies ((1 - r_2 + r_1) - r_1)^2 + 4r_1(1 + r_1 - r_2) &\leq 1 \\ \implies (1 - r_2 + 2r_1)^2 &\leq 1 \\ \implies -1 \leq 1 - r_2 + 2r_1 &\leq 1 \\ \implies r_2 - 2 \leq 2r_1 \leq r_2 \end{aligned}$$

Thus the resulting inequality is

$$-r_2 \leq 2r_1 \leq 2 - r_2 \tag{4.27}$$

Using the condition that $r_1 \geq 0$ with Eq. (4.27), the triangular region with boundaries defined by the lines Eqs. (4.24a)–(4.24c) is obtained. Again, as done before, replacing r_1 with $-r_1$, the region bounded by the lines is defined by Eqs. (4.23a)–(4.23c). So for Eq. (4.14) the same rectangular region as Eq. (4.14), which is bounded by the constraint $0 \leq r_1 \leq 1$ and $0 \leq r_2 \leq 1$ is obtained.

Thus for acquiring phase the stable range for r_1 and r_2 is obtained as $r_1 \in [0, 1]$ and $r_2 \in [0, 2]$.

Experimental results

To verify our findings, we remove the acquiring phase from SGO algorithm and only keep the improving phase. The stable range of r is compared against 3 schemes of unstable ranges of r , i.e.

- (i) $r_1 \in [1, 2], r_2 \in [0, 2]$ (r_1 outside stable range)
- (ii) $r_1 \in [0, 1], r_2 \in [2, 4]$ (r_2 outside stable range)
- (iii) $r_1 \in [1, 1.5], r_2 \in [2, 3]$ (Both outside stable range)

We have used Rastrigin, Rosenbrock and Alpine function for the testing with $D = 30$, number of swarm members i.e. $N = 50$ with 300 epochs (or $50 * 300 = 15,000$ function evaluations). The testing is done in 3 phases.

- M, o as identity matrix and null vector respectively. Average and Standard deviation taken over 30 iterations.
- M as identity matrix and o as a random vector whose value will be changed after each 10th iteration. Average and Standard deviation taken over 100 iterations.

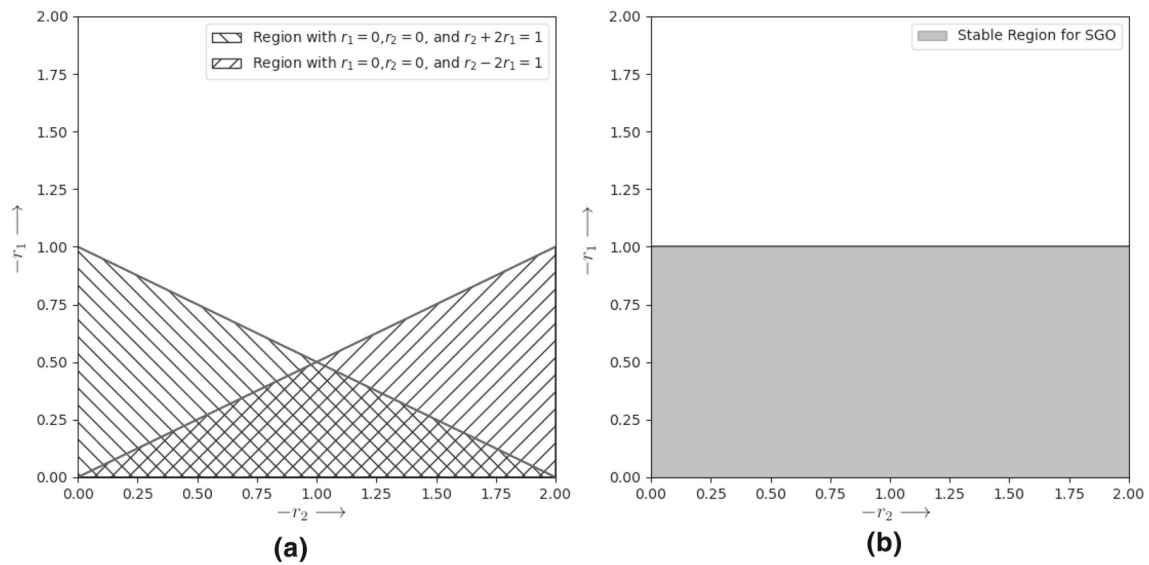


Fig. 7 Graph demonstrating stable region for SGO w.r.t r_1 and r_2

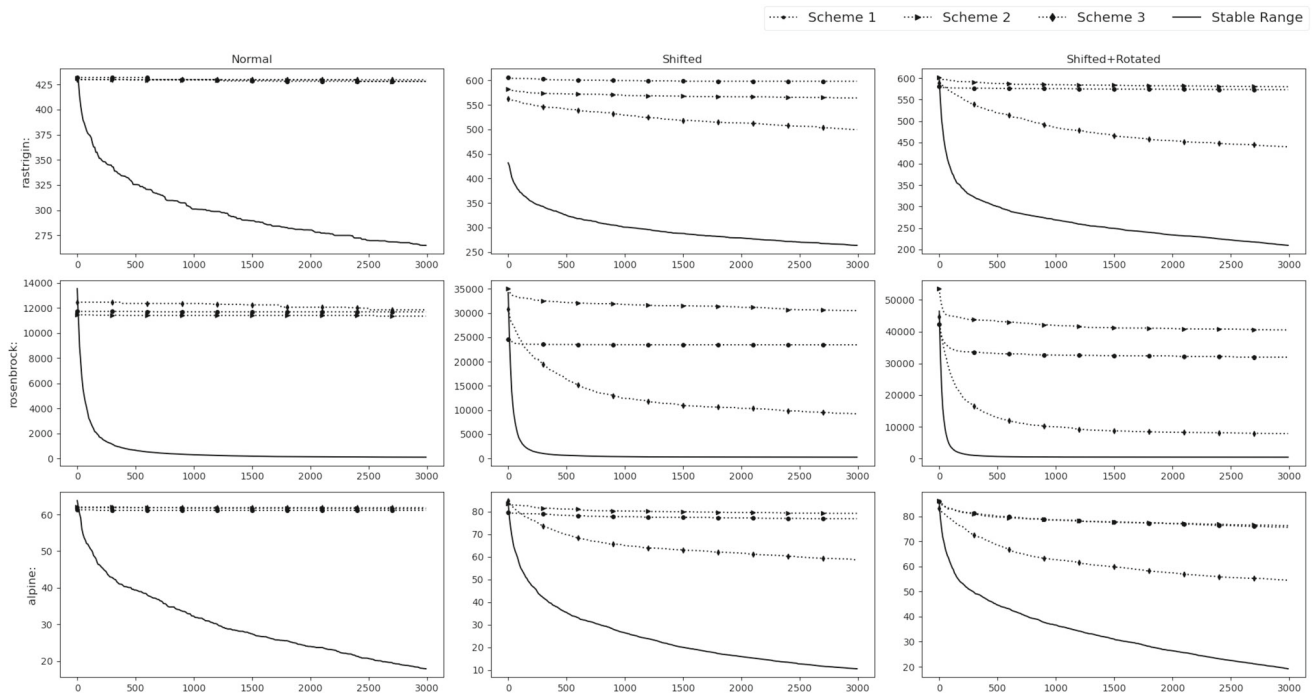


Fig. 8 Experimental results of testing of the acquiring phase

- M as random rotation matrix and o as a random vector. Their values will be changed after each 12th iteration. Average and Standard deviation taken over 120 iterations.

Figure 8 shows the results of the 3 phases mentioned above. We can clearly see that the stable range performs better than the 3 unstable schemes.

Numerical experiments

All the simulations were carried out in Matlab R2016a on the system having Intel Core i7 2.67 GHz processor and 8 GB RAM.

To verify the findings, the performance of SGO algorithm (with both acquiring and improving phase) is tested with the

Table 2 Result of optimization (10D default functions)

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$	Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$	Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Sphere	C:0.2	4.042e - 138 ± 4.3e - 138	C:0.2	2.933e - 136 ± 8.4e - 137
	C:0.4	1.735e - 77 ± 2.4e - 77	C:0.4	5.762e - 78 ± 8.6e - 78
	C:0.6	1.833e - 41 ± 9.9e - 42	C:0.6	8.766e - 48 ± 2.6e - 47
	C:0.8	1.439e - 16 ± 9.5e - 17	C:0.8	7.427e - 19 ± 1.8e - 18
	C:1.0	2015 ± 1.2e + 03	C:1.0	259.9 ± 4.5e + 02
	C:0.2	0 ± 0	C:0.2	0 ± 0
	C:0.4	0 ± 0	C:0.4	0 ± 0
	C:0.6	1.115e - 30 ± 2.6e - 30	C:0.6	0.2218 ± 1.2
	C:0.8	3.777e - 17 ± 2e - 17	C:0.8	0.1563 ± 0.84
	C:1.0	209.1 ± 2e + 02	C:1.0	110 ± 78
Ackley	C:0.2	0 ± 0	C:0.2	0 ± 0
	C:0.4	2.605e - 15 ± 1.6e - 15	C:0.4	4.737e - 16 ± 1.2e - 15
	C:0.6	4.145e - 15 ± 1.3e - 15	C:0.6	2.842e - 15 ± 1.4e - 15
	C:0.8	5.519e - 09 ± 1.5e - 09	C:0.8	0.1891 ± 0.57
	C:1.0	12.75 ± 2	C:1.0	11.69 ± 4.5
	C:0.2	0.003128 ± 0.017	C:0.2	0 ± 0
	C:0.4	0.02456 ± 0.11	C:0.4	0.002927 ± 0.0072
	C:0.6	0.3743 ± 0.41	C:0.6	0.08866 ± 0.18
	C:0.8	0.8694 ± 0.32	C:0.8	0.09183 ± 0.1
	C:1.0	29.56 ± 18	C:1.0	4.479 ± 4
Rastrigin	C:0.2	0 ± 0	C:0.2	0 ± 0
	C:0.4	5.64 ± 7.9	C:0.4	1.07 ± 4.9
	C:0.6	13.48 ± 6.3	C:0.6	10.13 ± 11
	C:0.8	20.32 ± 8.3	C:0.8	17.7 ± 14
	C:1.0	39.23 ± 14	C:1.0	41.89 ± 18
	C:0.2	0.862 ± 0.4	C:0.2	0.862 ± 0.4
	C:0.4	173.9 ± 49	C:0.4	173.9 ± 49
	C:0.6	3.665 ± 20	C:0.6	3.665 ± 20
	C:0.8	57.45 ± 50	C:0.8	57.45 ± 50
	C:1.0	80.54 ± 38	C:1.0	80.54 ± 38
Griewank	C:0.2	1.996 ± 6	C:0.2	1.996 ± 6
	C:0.4	19.83 ± 0.36	C:0.4	19.83 ± 0.36
	C:0.6	0 ± 0	C:0.6	0 ± 0
	C:0.8	0.2788 ± 0.46	C:0.8	0.2788 ± 0.46
	C:1.0	0.5341 ± 0.5	C:1.0	0.5341 ± 0.5
	C:0.2	2.013e - 15 ± 1.8e - 15	C:0.2	2.013e - 15 ± 1.8e - 15
	C:0.4	0.6656 ± 3.6	C:0.4	0.6656 ± 3.6
	C:0.6	1.996 ± 6	C:0.6	1.996 ± 6
	C:0.8	19.83 ± 0.36	C:0.8	19.83 ± 0.36
	C:1.0	0 ± 0	C:1.0	0 ± 0

Table 2 continued

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$		Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$		Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Alpine	C:0.2	4.05e - 71 ± 5.2e - 71	C:0.2	5.531e - 70 ± 1.1e - 70	C:0.2	5.034e - 70 ± 1e - 70
	C:0.4	1.124e - 40 ± 7.2e - 41	C:0.4	7.371e - 41 ± 9.1e - 41	C:0.4	6.4e - 40 ± 1.2e - 40
	C:0.6	0.02448 ± 0.089	C:0.6	3.491e - 07 ± 1.9e - 06	C:0.6	5.313 ± 6.8
	C:0.8	0.4768 ± 0.53	C:0.8	0.2318 ± 0.63	C:0.8	18 ± 2.8
	C:1.0	3.075 ± 1.2	C:1.0	1.396 ± 2	C:1.0	18.05 ± 2.9
	C:0.2	1.321e - 195 ± 0	C:0.2	3.462e - 143 ± 8.1e - 143	C:0.2	2.134e - 141 ± 2.5e - 141
Sum of powers	C:0.4	5.869e - 138 ± 3.2e - 137	C:0.4	1.63e - 123 ± 5.4e - 123	C:0.4	2.893e - 81 ± 4.4e - 81
	C:0.6	1.776e - 50 ± 8.8e - 50	C:0.6	4.994e - 105 ± 1.9e - 104	C:0.6	6.261e - 46 ± 6.3e - 46
	C:0.8	1.27e - 23 ± 6e - 23	C:0.8	7.06e - 54 ± 3.8e - 53	C:0.8	4.975e - 21 ± 7.8e - 21
	C:1.0	0.01267 ± 0.024	C:1.0	0.0003156 ± 0.0009	C:1.0	0.5998 ± 0.28
	C:0.2	6.194e - 138 ± 6.3e - 138	C:0.2	1.385e - 137 ± 2.5e - 137	C:0.2	7.322e - 138 ± 2.8e - 138
	C:0.4	2.742e - 78 ± 1.6e - 78	C:0.4	1.218e - 77 ± 6.8e - 78	C:0.4	1.677e - 77 ± 2.9e - 77
Zakharov	C:0.6	4.847e - 43 ± 4e - 43	C:0.6	9.801e - 45 ± 1.8e - 44	C:0.6	3.194e - 42 ± 3.1e - 42
	C:0.8	4.191e - 18 ± 2.9e - 18	C:0.8	2.046e - 19 ± 4.4e - 19	C:0.8	3.608e - 17 ± 2.3e - 17
	C:1.0	77.2 ± 39	C:1.0	72.33 ± 78	C:1.0	469.5 ± 1.8e + 02

Table 3 Result of optimization (30D default functions)

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$	Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$	Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$
Sphere	C:0.2	0 ± 0	0 ± 0
	C:0.4	8.148e - 237 ± 0	1.14e - 234 ± 0
	C:0.6	6.179e - 130 ± 2.1e - 130	1.786e - 141 ± 3.2e - 141
	C:0.8	5.035e - 55 ± 1.5e - 55	3.313e - 58 ± 1.8e - 57
	C:1.0	4405 ± 9.9e + 02	307.4 ± 2.7e + 02
	C:0.2	0 ± 0	0 ± 0
	C:0.4	0 ± 0	0 ± 0
	C:0.6	5.777e - 31 ± 2.3e - 30	0 ± 0
Rosenbrock	C:0.8	2.12e - 29 ± 2.3e - 29	2.336e - 29 ± 2.2e - 29
	C:1.0	516.5 ± 2.1e + 02	362.8 ± 2.1e + 02
	C:0.2	0 ± 0	0 ± 0
	C:0.4	0 ± 0	0 ± 0
	C:0.6	0 ± 0	0 ± 0
	C:0.8	1.681e - 29 ± 2.8e - 29	1.681e - 29 ± 2.8e - 29
	C:1.0	9842 ± 2.7e + 03	9842 ± 2.7e + 03
	C:0.2	0 ± 0	0 ± 0
Ackley	C:0.4	2.605e - 15 ± 1.6e - 15	7.105e - 16 ± 1.4e - 15
	C:0.6	3.671e - 15 ± 6.4e - 16	3.553e - 15 ± 0
	C:0.8	9.237e - 15 ± 3.3e - 15	8.053e - 15 ± 2.9e - 15
	C:1.0	12.29 ± 1	12.05 ± 2.9
	C:0.2	0 ± 0	0 ± 0
	C:0.4	0.0003399 ± 0.0018	0 ± 0
	C:0.6	0.005177 ± 0.018	0.001802 ± 0.0077
	C:0.8	0.02604 ± 0.076	0.01272 ± 0.015
Griewank	C:1.0	46.76 ± 13	6.1 ± 6.7
	C:0.2	0 ± 0	0 ± 0
	C:0.4	0.03509 ± 0.19	0.03509 ± 0.19
	C:0.6	0.138 ± 0.35	0.138 ± 0.35
	C:0.8	0.2734 ± 0.45	0.2734 ± 0.45
	C:1.0	574.2 ± 59	574.2 ± 59
	C:0.2	0 ± 0	0 ± 0
	C:0.4	0 ± 0	0 ± 0

Table 3 continued

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$	Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$	Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$
Rastrigin	C:0.2	0 ± 0	C:0.2
	C:0.4	2.457 ± 7.6	C:0.4
	C:0.6	17.29 ± 13	C:0.6
	C:0.8	28.29 ± 10	C:0.8
	C:1.0	106.5 ± 19	C:1.0
	C:0.2	2.883e - 209 ± 0	C:0.2
	C:0.4	4.944e - 120 ± 2.8e - 120	C:0.4
	C:0.6	1.354e - 66 ± 2.8e - 67	C:0.6
	C:0.8	0.02008 ± 0.06	C:0.8
	C:1.0	9.763 ± 2.5	C:1.0
sum of powers	C:0.2	0 ± 0	C:0.2
	C:0.4	0 ± 0	C:0.4
	C:0.6	1.314e - 185 ± 0	C:0.6
	C:0.8	8.053e - 71 ± 1.9e - 70	C:0.8
	C:1.0	4.438e - 05 ± 5.8e - 05	C:1.0
	C:0.2	0 ± 0	C:0.2
	C:0.4	7.831e - 238 ± 0	C:0.4
	C:0.6	9.289e - 132 ± 1.1e - 131	C:0.6
	C:0.8	2.87e - 58 ± 1e - 57	C:0.8
	C:1.0	37.47 ± 19	C:1.0
zakharov	C:0.2	0 ± 0	C:0.2
	C:0.4	6.687 ± 28	C:0.4
	C:0.6	45.57 ± 38	C:0.6
	C:0.8	156.3 ± 31	C:0.8
	C:1.0	2.68e - 209 ± 0	C:1.0
	C:0.2	5.572e - 119 ± 1.7e - 119	C:0.2
	C:0.4	6.703e - 73 ± 7e - 73	C:0.4
	C:0.6	0.1081 ± 0.54	C:0.6
	C:0.8	4.77 ± 4.4	C:0.8
	C:1.0	0 ± 0	C:1.0
Rastrigin	C:0.2	0 ± 0	C:0.2
	C:0.4	63.39 ± 5.2	C:0.4
	C:0.6	0 ± 0	C:0.6
	C:0.8	1.628e - 240 ± 0	C:0.8
	C:1.0	9.934e - 135 ± 9.9e - 135	C:1.0
	C:0.2	9.611e - 60 ± 8.6e - 60	C:0.2
	C:0.4	0.796 ± 0.22	C:0.4
	C:0.6	0 ± 0	C:0.6
	C:0.8	2.188e - 236 ± 0	C:0.8
	C:1.0	2.081e - 130 ± 2.6e - 130	C:1.0
zakharov	C:0.2	1.236e - 55 ± 1e - 55	C:0.2
	C:0.4	1506 ± 4.3e + 02	C:0.4
	C:0.6	2.509e - 236 ± 0	C:0.6
	C:0.8	9.312e - 131 ± 5.3e - 131	C:0.8
	C:1.0	8.57e - 58 ± 1.6e - 57	C:1.0
	C:0.2	0 ± 0	C:0.2
	C:0.4	6.563e - 08 ± 2.1e - 07	C:0.4
	C:0.6	2.358e - 287 ± 0	C:0.6
	C:0.8	2.509e - 236 ± 0	C:0.8
	C:1.0	166.4 ± 1.3e + 02	C:1.0

Table 4 Result of optimization (10D shifted functions)

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$		Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$		Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Sphere	C:0.2	$3027 \pm 2.9e + 03$	C:0.2	$366.9 \pm 6e + 02$	C:0.2	$1.629e + 04 \pm 5.1e + 03$
	C:0.4	$4219 \pm 3.5e + 03$	C:0.4	$90.55 \pm 2.8e + 02$	C:0.4	$1.401e + 04 \pm 5.4e + 03$
	C:0.6	$4674 \pm 3.2e + 03$	C:0.6	$83.06 \pm 2.6e + 02$	C:0.6	$1.339e + 04 \pm 5.6e + 03$
	C:0.8	$5085 \pm 3.6e + 03$	C:0.8	4.261 ± 13	C:0.8	$1.327e + 04 \pm 4.9e + 03$
	C:1.0	$8703 \pm 5e + 03$	C:1.0	$1232 \pm 1.6e + 03$	C:1.0	$1.953e + 04 \pm 5.6e + 03$
Rosenbrock	C:0.2	$321 \pm 4.6e + 02$	C:0.2	$69.21 \pm 1.3e + 02$	C:0.2	$1565 \pm 9.3e + 02$
	C:0.4	$367.1 \pm 4.5e + 02$	C:0.4	39.98 ± 49	C:0.4	$1332 \pm 7.4e + 02$
	C:0.6	$482.6 \pm 5.8e + 02$	C:0.6	37.38 ± 45	C:0.6	$885.1 \pm 4.9e + 02$
	C:0.8	$618.2 \pm 9.2e + 02$	C:0.8	37.73 ± 40	C:0.8	$884.5 \pm 5.6e + 02$
	C:1.0	$1419 \pm 1.5e + 03$	C:1.0	$187.9 \pm 2.8e + 02$	C:1.0	$1695 \pm 1e + 03$
Ackley	C:0.2	15.6 ± 3.3	C:0.2	7.754 ± 5.5	C:0.2	19.88 ± 0.79
	C:0.4	16.34 ± 3.2	C:0.4	7.513 ± 5.5	C:0.4	19.54 ± 0.8
	C:0.6	16.52 ± 3.3	C:0.6	7.78 ± 6.4	C:0.6	19.13 ± 1
	C:0.8	17.21 ± 2.9	C:0.8	7.489 ± 6.8	C:0.8	18.92 ± 1.6
	C:1.0	17.34 ± 2.4	C:1.0	14.36 ± 4.5	C:1.0	19.83 ± 1
Griewank	C:0.2	34.41 ± 29	C:0.2	3.27 ± 3.8	C:0.2	132 ± 40
	C:0.4	39.38 ± 32	C:0.4	2.564 ± 4	C:0.4	121.9 ± 39
	C:0.6	48.84 ± 33	C:0.6	1.257 ± 1.5	C:0.6	118.4 ± 49
	C:0.8	43.49 ± 35	C:0.8	1.299 ± 2.6	C:0.8	108.9 ± 57
	C:1.0	82.74 ± 44	C:1.0	15.55 ± 22	C:1.0	167.5 ± 43
Rastrigin	C:0.2	57.92 ± 23	C:0.2	47.78 ± 18	C:0.2	124.6 ± 21
	C:0.4	58.92 ± 19	C:0.4	50.39 ± 18	C:0.4	120.3 ± 19
	C:0.6	61.83 ± 25	C:0.6	45.84 ± 17	C:0.6	114.8 ± 20
	C:0.8	56.62 ± 24	C:0.8	41.75 ± 15	C:0.8	109 ± 24
	C:1.0	68.83 ± 24	C:1.0	52.04 ± 21	C:1.0	116.8 ± 21
Alpine	C:0.2	2.039 ± 1.6	C:0.2	2.692 ± 2.5	C:0.2	15.16 ± 2.8
	C:0.4	1.762 ± 1.3	C:0.4	2.825 ± 2.8	C:0.4	13.56 ± 3.2
	C:0.6	2.279 ± 1.7	C:0.6	2.378 ± 2.9	C:0.6	15.44 ± 3.5
	C:0.8	1.98 ± 1.5	C:0.8	1.468 ± 2.3	C:0.8	14.65 ± 4
	C:1.0	4.478 ± 1.9	C:1.0	2.593 ± 2.1	C:1.0	15.43 ± 3.3
Sum of powers	C:0.2	0.07342 ± 0.1	C:0.2	0.0006466 ± 0.0019	C:0.2	0.4429 ± 0.32
	C:0.4	0.09127 ± 0.16	C:0.4	0.001545 ± 0.0082	C:0.4	0.4041 ± 0.29
	C:0.6	0.1412 ± 0.18	C:0.6	0.0006761 ± 0.0051	C:0.6	0.438 ± 0.4
	C:0.8	0.1539 ± 0.23	C:0.8	$4.373e - 06 \pm 2.1e - 05$	C:0.8	0.2851 ± 0.33
	C:1.0	0.2864 ± 0.41	C:1.0	0.03304 ± 0.14	C:1.0	0.8882 ± 1.2
zakharov	C:0.2	70.72 ± 75	C:0.2	32.9 ± 44	C:0.2	$325.1 \pm 1.2e + 02$
	C:0.4	96.77 ± 93	C:0.4	22.92 ± 21	C:0.4	$260.9 \pm 1.5e + 02$
	C:0.6	74.78 ± 62	C:0.6	20.71 ± 29	C:0.6	$247.3 \pm 1.2e + 02$
	C:0.8	112.2 ± 83	C:0.8	6.52 ± 5.8	C:0.8	$227.4 \pm 1.3e + 02$
	C:1.0	$257.3 \pm 1.5e + 02$	C:1.0	$132.8 \pm 1.7e + 02$	C:1.0	$560.5 \pm 3.1e + 02$

Table 5 Result of optimization (30D shifted functions)

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$		Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$		Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Sphere	C:0.2	77.6 ± 2.1e + 02	C:0.2	390.2 ± 6.3e + 02	C:0.2	8.633e + 04 ± 1.4e + 04
	C:0.4	41.89 ± 1.1e + 02	C:0.4	207.9 ± 5.2e + 02	C:0.4	8.061e + 04 ± 1.4e + 04
	C:0.6	95.06 ± 2.5e + 02	C:0.6	123.3 ± 2.5e + 02	C:0.6	7.681e + 04 ± 1.5e + 04
	C:0.8	135.5 ± 3.5e + 02	C:0.8	28.29 ± 69	C:0.8	8.327e + 04 ± 1.9e + 04
	C:1.0	2.657e + 04 ± 8.6e + 03	C:1.0	3391 ± 3.5e + 03	C:1.0	7.047e + 04 ± 1.1e + 04
Rosenbrock	C:0.2	154.9 ± 65	C:0.2	131.5 ± 83	C:0.2	1.83e + 04 ± 5.7e + 03
	C:0.4	148.3 ± 66	C:0.4	132.4 ± 94	C:0.4	1.781e + 04 ± 7.2e + 03
	C:0.6	149 ± 1.2e + 02	C:0.6	98.49 ± 50	C:0.6	1.632e + 04 ± 6.5e + 03
	C:0.8	162.8 ± 73	C:0.8	88.5 ± 45	C:0.8	1.378e + 04 ± 7.4e + 03
	C:1.0	3719 ± 2e + 03	C:1.0	526.9 ± 4.7e + 02	C:1.0	1.07e + 04 ± 4e + 03
Ackley	C:0.2	14.99 ± 3.4	C:0.2	5.736 ± 4	C:0.2	20.88 ± 0.23
	C:0.4	16.05 ± 3	C:0.4	5.402 ± 3.9	C:0.4	20.85 ± 0.25
	C:0.6	16.1 ± 2.9	C:0.6	4.728 ± 4.1	C:0.6	20.71 ± 0.34
	C:0.8	17 ± 2.6	C:0.8	3.782 ± 6.3	C:0.8	20.64 ± 0.54
	C:1.0	18.42 ± 0.98	C:1.0	15.7 ± 3.5	C:1.0	20.93 ± 0.25
Griewank	C:0.2	1.029 ± 0.65	C:0.2	4.266 ± 6.7	C:0.2	757.1 ± 1.2e + 02
	C:0.4	2.702 ± 9	C:0.4	3.707 ± 5.7	C:0.4	703.1 ± 1.3e + 02
	C:0.6	1.852 ± 2.5	C:0.6	2.472 ± 3.2	C:0.6	714.7 ± 1.2e + 02
	C:0.8	1.712 ± 1.7	C:0.8	0.5808 ± 0.59	C:0.8	712.2 ± 1.5e + 02
	C:1.0	216.5 ± 65	C:1.0	34.43 ± 44	C:1.0	625.5 ± 1.2e + 02
Rastrigin	C:0.2	163.7 ± 37	C:0.2	210.5 ± 48	C:0.2	489.9 ± 31
	C:0.4	187.2 ± 43	C:0.4	215.6 ± 47	C:0.4	481 ± 48
	C:0.6	179 ± 35	C:0.6	213.2 ± 49	C:0.6	469.1 ± 46
	C:0.8	163.9 ± 37	C:0.8	186.4 ± 46	C:0.8	455.1 ± 60
	C:1.0	199.2 ± 36	C:1.0	168.9 ± 42	C:1.0	451 ± 43
Alpine	C:0.2	0.906 ± 1.1	C:0.2	11.17 ± 8.6	C:0.2	66.76 ± 8.1
	C:0.4	0.6791 ± 0.89	C:0.4	12.48 ± 10	C:0.4	63.11 ± 8.5
	C:0.6	0.9522 ± 1	C:0.6	10.28 ± 8.4	C:0.6	61.82 ± 7.8
	C:0.8	0.6743 ± 0.89	C:0.8	6.801 ± 6.6	C:0.8	62.73 ± 8.2
	C:1.0	18.27 ± 4.1	C:1.0	8.297 ± 4.6	C:1.0	60.34 ± 10
Sum of powers	C:0.2	1.08e - 05 ± 8.5e - 05	C:0.2	0.0002463 ± 0.0014	C:0.2	1.878 ± 0.94
	C:0.4	8.633e - 07 ± 3.8e - 06	C:0.4	6.075e - 06 ± 2.6e - 05	C:0.4	2.552 ± 0.91
	C:0.6	1.116e - 05 ± 3.9e - 05	C:0.6	1.256e - 05 ± 9.6e - 05	C:0.6	2.401 ± 1
	C:0.8	3.137e - 06 ± 1e - 05	C:0.8	1.498e - 06 ± 1.3e - 05	C:0.8	2.081 ± 1
	C:1.0	0.2864 ± 0.43	C:1.0	0.04058 ± 0.21	C:1.0	6.963 ± 9.3
Zakharov	C:0.2	17.89 ± 14	C:0.2	235.9 ± 71	C:0.2	1308 ± 2.4e + 02
	C:0.4	15.9 ± 14	C:0.4	191.2 ± 63	C:0.4	1281 ± 2.8e + 02
	C:0.6	24.19 ± 33	C:0.6	170 ± 61	C:0.6	1148 ± 2.7e + 02
	C:0.8	17.14 ± 16	C:0.8	99.43 ± 36	C:0.8	1134 ± 3.1e + 02
	C:1.0	135.9 ± 76	C:1.0	124.3 ± 1.4e + 02	C:1.0	1646 ± 3.7e + 02

ranges of r, r_1, r_2 found in the “Improving phase” section and “Acquiring phase” section. These ranges are referred to as stable range. We use $r \in [-c - 2, -c - 1] \cup [c + 1, c + 2]$ for the improving phase (defined in “Improving phase”

section (b)) and $r_1 \in [1, 1.5], r_2 \in [2, 3]$ for the acquiring phase (defined in “Acquiring phase” section (a)) to test the performance of

Table 6 Result of optimization (10D shifted and rotated functions)

Name	Default range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$		Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$		Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Sphere	C:0.2	$4150 \pm 3.7e + 03$	C:0.2	$152.7 \pm 3.3e + 02$	C:0.2	$1.583e + 04 \pm 4.4e + 03$
	C:0.4	$4741 \pm 3.6e + 03$	C:0.4	$196.6 \pm 4.3e + 02$	C:0.4	$1.316e + 04 \pm 4.3e + 03$
	C:0.6	$4688 \pm 4e + 03$	C:0.6	$73.86 \pm 2e + 02$	C:0.6	$1.413e + 04 \pm 6.2e + 03$
	C:0.8	$5886 \pm 4.5e + 03$	C:0.8	9.128 ± 31	C:0.8	$1.202e + 04 \pm 5.5e + 03$
	C:1.0	$9112 \pm 5.2e + 03$	C:1.0	$1170 \pm 1.6e + 03$	C:1.0	$1.519e + 04 \pm 6.3e + 03$
Rosenbrock	C:0.2	$441 \pm 5.2e + 02$	C:0.2	$129.1 \pm 2.2e + 02$	C:0.2	$2301 \pm 1.5e + 03$
	C:0.4	$680.6 \pm 7.2e + 02$	C:0.4	$150.5 \pm 2.1e + 02$	C:0.4	$1688 \pm 8.5e + 02$
	C:0.6	$742.5 \pm 1e + 03$	C:0.6	$90.62 \pm 1.2e + 02$	C:0.6	$1223 \pm 7.4e + 02$
	C:0.8	$1167 \pm 1e + 03$	C:0.8	$105.7 \pm 1.1e + 02$	C:0.8	$865.1 \pm 6.6e + 02$
	C:1.0	$2651 \pm 2.7e + 03$	C:1.0	$269.4 \pm 3.7e + 02$	C:1.0	$2126 \pm 2.3e + 03$
Ackley	C:0.2	16.32 ± 3.4	C:0.2	7.619 ± 5.1	C:0.2	19.79 ± 0.64
	C:0.4	16.42 ± 3.4	C:0.4	7.395 ± 5.5	C:0.4	19.4 ± 1
	C:0.6	17.14 ± 2.8	C:0.6	6.921 ± 6.1	C:0.6	19.05 ± 1.3
	C:0.8	17.17 ± 3.4	C:0.8	9.28 ± 6.8	C:0.8	18.78 ± 1.4
	C:1.0	17.63 ± 2.3	C:1.0	15.17 ± 4.7	C:1.0	19.84 ± 0.73
Griewank	C:0.2	28 ± 22	C:0.2	3.259 ± 5.9	C:0.2	156.5 ± 40
	C:0.4	40.82 ± 36	C:0.4	2.472 ± 3.2	C:0.4	119.5 ± 40
	C:0.6	51.83 ± 40	C:0.6	1.399 ± 1.4	C:0.6	99.43 ± 49
	C:0.8	62.38 ± 48	C:0.8	0.865 ± 0.56	C:0.8	89.79 ± 51
	C:1.0	80.87 ± 49	C:1.0	16.93 ± 29	C:1.0	148.6 ± 43
Rastrigin	C:0.2	60.08 ± 20	C:0.2	60.45 ± 23	C:0.2	118.7 ± 19
	C:0.4	62.86 ± 23	C:0.4	61.63 ± 19	C:0.4	114.3 ± 20
	C:0.6	64.35 ± 22	C:0.6	56.46 ± 18	C:0.6	112.2 ± 19
	C:0.8	58.07 ± 20	C:0.8	49.31 ± 19	C:0.8	103 ± 21
	C:1.0	68.27 ± 24	C:1.0	56.37 ± 25	C:1.0	113 ± 19
Alpine	C:0.2	3.978 ± 1.8	C:0.2	4.701 ± 3	C:0.2	14.16 ± 2.9
	C:0.4	3.314 ± 1.9	C:0.4	5.029 ± 3.2	C:0.4	12.99 ± 2.3
	C:0.6	3.742 ± 1.9	C:0.6	4.314 ± 3.3	C:0.6	12.58 ± 3.1
	C:0.8	4.057 ± 2.5	C:0.8	3.464 ± 3.1	C:0.8	11.63 ± 2.9
	C:1.0	5.353 ± 2.3	C:1.0	4.025 ± 2.4	C:1.0	13.01 ± 3.5
Sum of powers	C:0.2	0.0389 ± 0.071	C:0.2	0.001253 ± 0.0051	C:0.2	0.433 ± 0.26
	C:0.4	0.05334 ± 0.21	C:0.4	0.000174 ± 0.00089	C:0.4	0.3124 ± 0.24
	C:0.6	0.05502 ± 0.11	C:0.6	$6.444e - 05 \pm 0.00046$	C:0.6	0.2639 ± 0.26
	C:0.8	0.1219 ± 0.27	C:0.8	$1.661e - 05 \pm 7.9e - 05$	C:0.8	0.2353 ± 0.3
	C:1.0	0.1476 ± 0.25	C:1.0	0.02807 ± 0.18	C:1.0	0.6131 ± 0.63
Zakharov	C:0.2	112.8 ± 96	C:0.2	35.11 ± 61	C:0.2	$361.1 \pm 3.3e + 02$
	C:0.4	$107 \pm 1e + 02$	C:0.4	30.75 ± 62	C:0.4	$299.9 \pm 1.7e + 02$
	C:0.6	$114.8 \pm 1e + 02$	C:0.6	18.94 ± 38	C:0.6	$252.1 \pm 1.2e + 02$
	C:0.8	100.5 ± 78	C:0.8	5.325 ± 7.4	C:0.8	$5767 \pm 5.5e + 04$
	C:1.0	$253.1 \pm 1.4e + 02$	C:1.0	$112.1 \pm 1.5e + 02$	C:1.0	$545 \pm 3e + 02$

SGO when the parameters are outside of the stable range. These ranges are referred as Unstable Range. Finally the ranges of SGO mentioned in the original paper [3] is taken to test how well the stable range performs compared to the

default ranges. These ranges are referred to as Default Range. The test is performed for 10D and 30D versions of the functions mentioned in Table 1 using $N = 10$, epoch = 100 and $N = 50$, epoch = 300 respectively. Values of M and o

Table 7 Result of optimization (30D shifted and rotated functions)

Name	Normal range $r \in [0, 1]$ $r_1, r_2 \in [0, 1]$		Stable range $r \in [c - 1, c + 1]$ $r_1 \in [0, 1], r_2 \in [0, 2]$		Unstable range $r \in [c - 2, c - 1] \cup [c + 1, c + 2]$ $r_1 \in [1, 1.5], r_2 \in [2, 3]$	
Sphere	C:0.2	44.55 ± 1.5e + 02	C:0.2	499.7 ± 8.7e + 02	C:0.2	8.174e + 04 ± 1.5e + 04
	C:0.4	50.19 ± 1.4e + 02	C:0.4	318.4 ± 5.5e + 02	C:0.4	8.125e + 04 ± 1.1e + 04
	C:0.6	60.07 ± 1.5e + 02	C:0.6	80.06 ± 1.8e + 02	C:0.6	7.913e + 04 ± 1.5e + 04
	C:0.8	76.53 ± 1.8e + 02	C:0.8	28.51 ± 68	C:0.8	8.128e + 04 ± 1.9e + 04
	C:1.0	2.672e + 04 ± 8.5e + 03	C:1.0	3372 ± 3.3e + 03	C:1.0	7.654e + 04 ± 1.4e + 04
Rosenbrock	C:0.2	300.2 ± 1.6e + 02	C:0.2	218.1 ± 97	C:0.2	2.025e + 04 ± 7.1e + 03
	C:0.4	301.2 ± 1.3e + 02	C:0.4	319.3 ± 1.7e + 02	C:0.4	1.825e + 04 ± 6.3e + 03
	C:0.6	232.4 ± 75	C:0.6	247.1 ± 1.2e + 02	C:0.6	1.543e + 04 ± 6.2e + 03
	C:0.8	255.5 ± 1.1e + 02	C:0.8	276.6 ± 1.5e + 02	C:0.8	1.157e + 04 ± 6.2e + 03
	C:1.0	5432 ± 3.2e + 03	C:1.0	703.6 ± 4.9e + 02	C:1.0	1.255e + 04 ± 4.9e + 03
Ackley	C:0.2	13.54 ± 3.6	C:0.2	5.819 ± 3.7	C:0.2	20.83 ± 0.17
	C:0.4	14.91 ± 3.6	C:0.4	5.07 ± 3.5	C:0.4	20.76 ± 0.22
	C:0.6	15.53 ± 3.4	C:0.6	3.752 ± 3.2	C:0.6	20.69 ± 0.34
	C:0.8	15.27 ± 3.2	C:0.8	3.137 ± 4.9	C:0.8	20.47 ± 0.64
	C:1.0	18.34 ± 1	C:1.0	15.87 ± 3.5	C:1.0	20.83 ± 0.21
Griewank	C:0.2	1.252 ± 0.87	C:0.2	3.504 ± 4.8	C:0.2	781.7 ± 1.3e + 02
	C:0.4	1.443 ± 2	C:0.4	4.617 ± 7.5	C:0.4	737.8 ± 1.4e + 02
	C:0.6	1.905 ± 5	C:0.6	1.698 ± 2.3	C:0.6	706 ± 1.3e + 02
	C:0.8	1.413 ± 1.3	C:0.8	0.7584 ± 0.68	C:0.8	719.5 ± 1.4e + 02
	C:1.0	222.3 ± 75	C:1.0	27.92 ± 28	C:1.0	665.1 ± 1.1e + 02
Rastrigin	C:0.2	164.1 ± 36	C:0.2	251.8 ± 54	C:0.2	514.2 ± 42
	C:0.4	187.6 ± 40	C:0.4	266.4 ± 47	C:0.4	483.4 ± 47
	C:0.6	202.5 ± 35	C:0.6	266.6 ± 40	C:0.6	469.1 ± 53
	C:0.8	157.4 ± 32	C:0.8	208.1 ± 55	C:0.8	448.3 ± 53
	C:1.0	214.3 ± 43	C:1.0	182.3 ± 50	C:1.0	431.8 ± 42
Alpine	C:0.2	8.209 ± 3.3	C:0.2	26.75 ± 13	C:0.2	66.29 ± 7.5
	C:0.4	8.573 ± 3.2	C:0.4	24.11 ± 11	C:0.4	63.25 ± 6.4
	C:0.6	8.227 ± 3.3	C:0.6	23.71 ± 12	C:0.6	60.11 ± 7.6
	C:0.8	8.554 ± 3.6	C:0.8	14.64 ± 11	C:0.8	56.65 ± 7.1
	C:1.0	21.24 ± 4.8	C:1.0	11.81 ± 4.8	C:1.0	56.12 ± 8.4
Sum of powers	C:0.2	1.089e - 06 ± 2.6e - 06	C:0.2	8.12e - 06 ± 4.1e - 05	C:0.2	7.027 ± 9.1
	C:0.4	2.836e - 06 ± 1.3e - 05	C:0.4	8.852e - 06 ± 4.3e - 05	C:0.4	4.929 ± 5.1
	C:0.6	8.458e - 06 ± 7.4e - 05	C:0.6	2.259e - 06 ± 4.7e - 06	C:0.6	5.645 ± 6.7
	C:0.8	1.965e - 06 ± 3.2e - 06	C:0.8	8.893e - 07 ± 1.1e - 06	C:0.8	4.328 ± 5.5
	C:1.0	0.1477 ± 0.26	C:1.0	0.03755 ± 0.27	C:1.0	7.039 ± 15
Zakharov	C:0.2	15.93 ± 15	C:0.2	230.9 ± 72	C:0.2	1358 ± 2.6e + 02
	C:0.4	19.02 ± 21	C:0.4	233.8 ± 1.4e + 02	C:0.4	1231 ± 2.7e + 02
	C:0.6	17.32 ± 28	C:0.6	155.7 ± 56	C:0.6	1166 ± 2.7e + 02
	C:0.8	12.97 ± 15	C:0.8	107.7 ± 52	C:0.8	1072 ± 2.8e + 02
	C:1.0	147.7 ± 98	C:1.0	146.2 ± 1.5e + 02	C:1.0	1619 ± 4.6e + 02

(“Optimization functions” section) are changed to test the optimization functions in different scenarios. In Tables 2, 3, 4, 5, 6, 7, the bold value represents the optimal value obtained out of three types of ranges mentioned above.

Default numerical benchmark functions

This is the simplest scenario where the gbest or the o vector lies at the origin and no rotation is done i.e. M is the identity

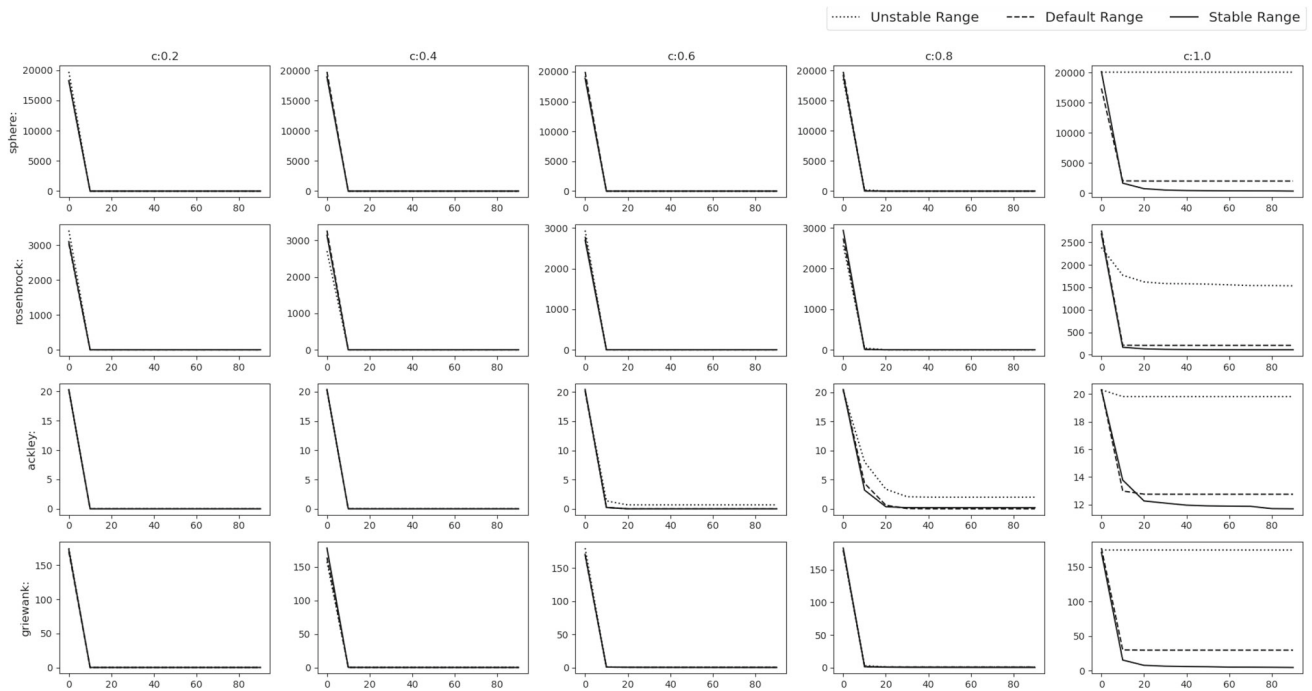


Fig. 9 Experimental results of optimizing 10D Default functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

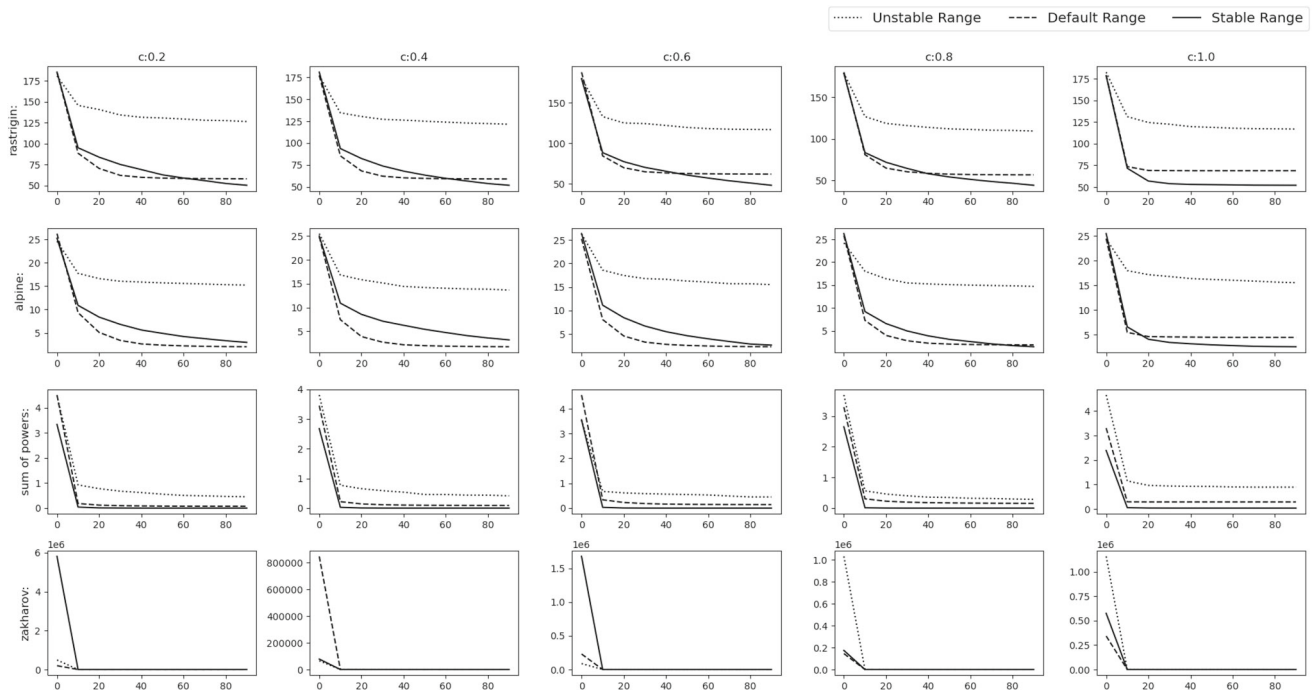


Fig. 10 Experimental results of optimizing 10D Default functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

matrix. The results can be seen in Figs. 9, 10 and Table 2 for the 10D experiment and Figs. 11, 12 and Table 3 for the 30D experiment.

The results show that the default range and stable range perform nearly similar while the unstable range gives the worst results. These results are more apparent as the c value increases.

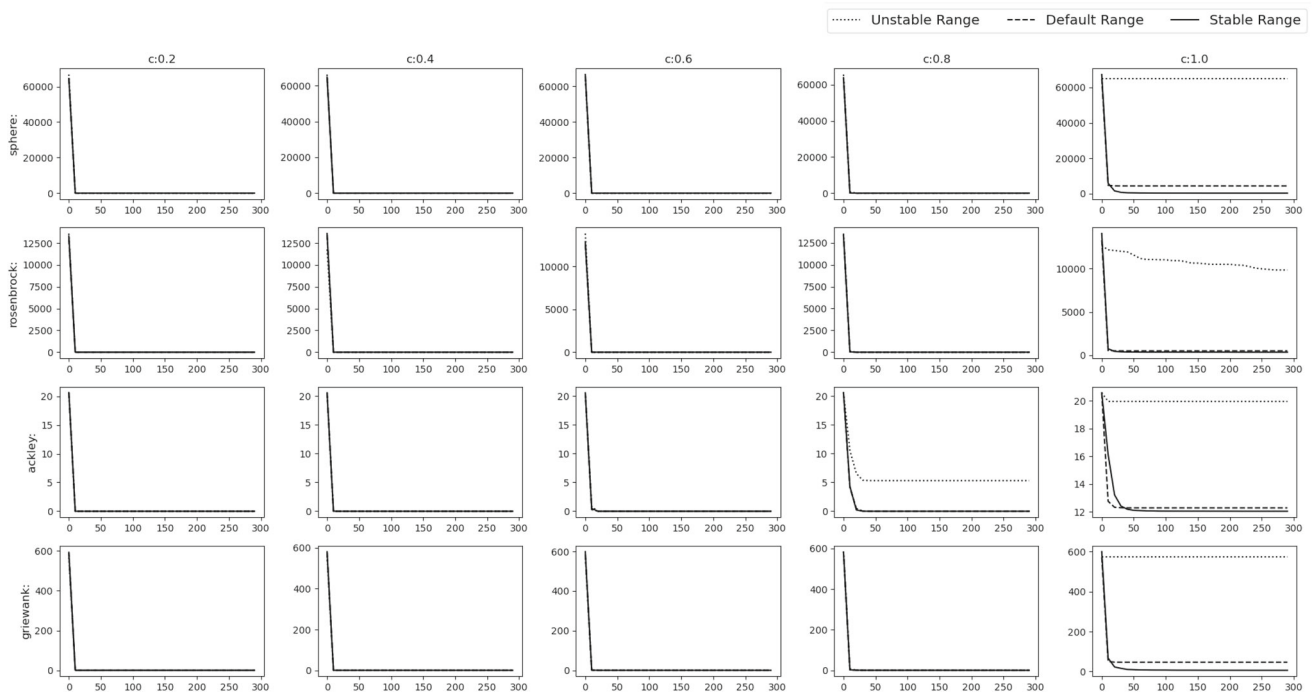


Fig. 11 Experimental results of optimizing 30D Default functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

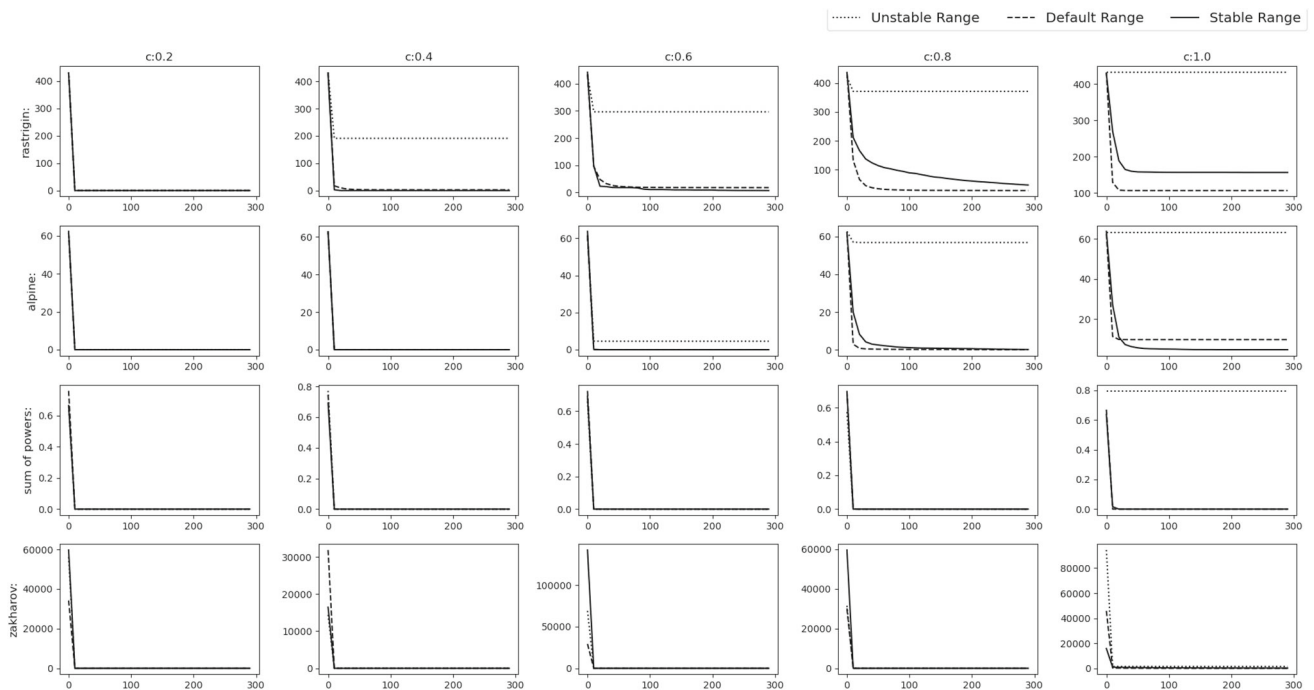


Fig. 12 Experimental results of optimizing 30D Default functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

Shifted Gbest numerical benchmark functions

In this scenario the gbest or the o vector is shifted to a different position randomly chosen inside the D-dimensional space

defined by the range of A_i . No rotation is done i.e. M is the identity matrix. The results can be observed in Figs. 13, 14 and Table 4 for the 10D experiment and Figs. 15, 16 and Table 5 for the 30D experiment. The unstable range clearly gives

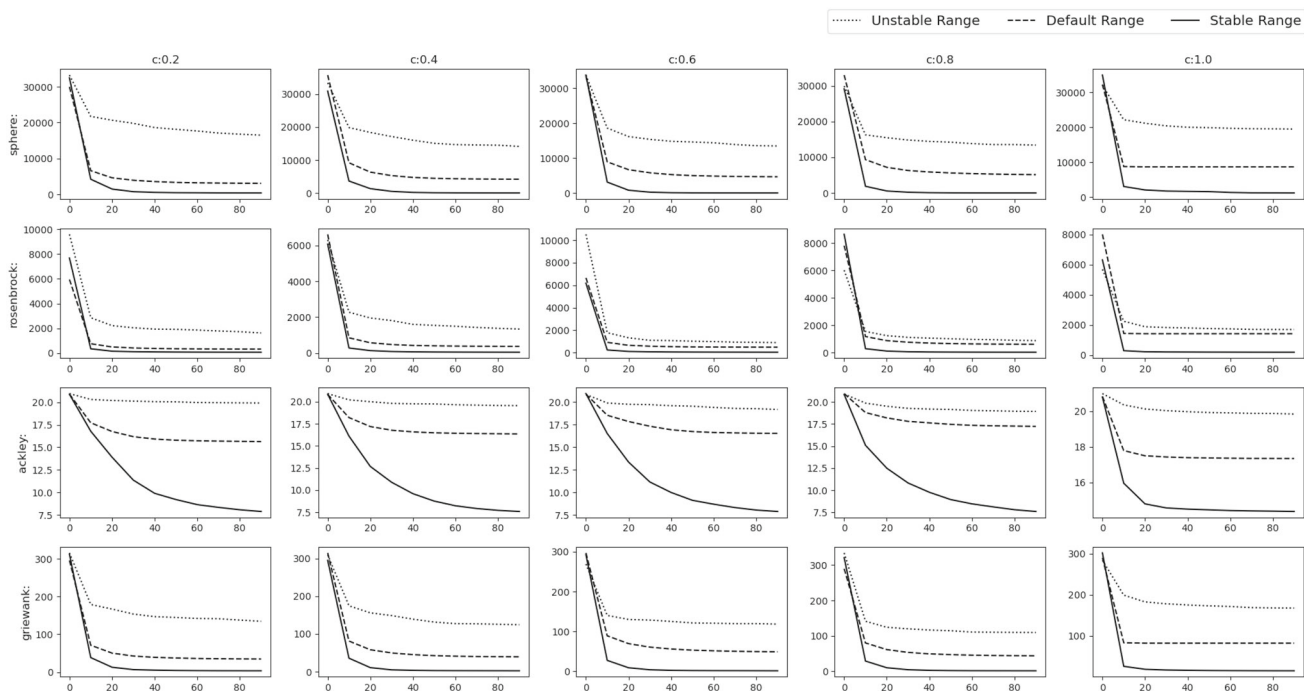


Fig. 13 Experimental results of optimizing 10D Shifted functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

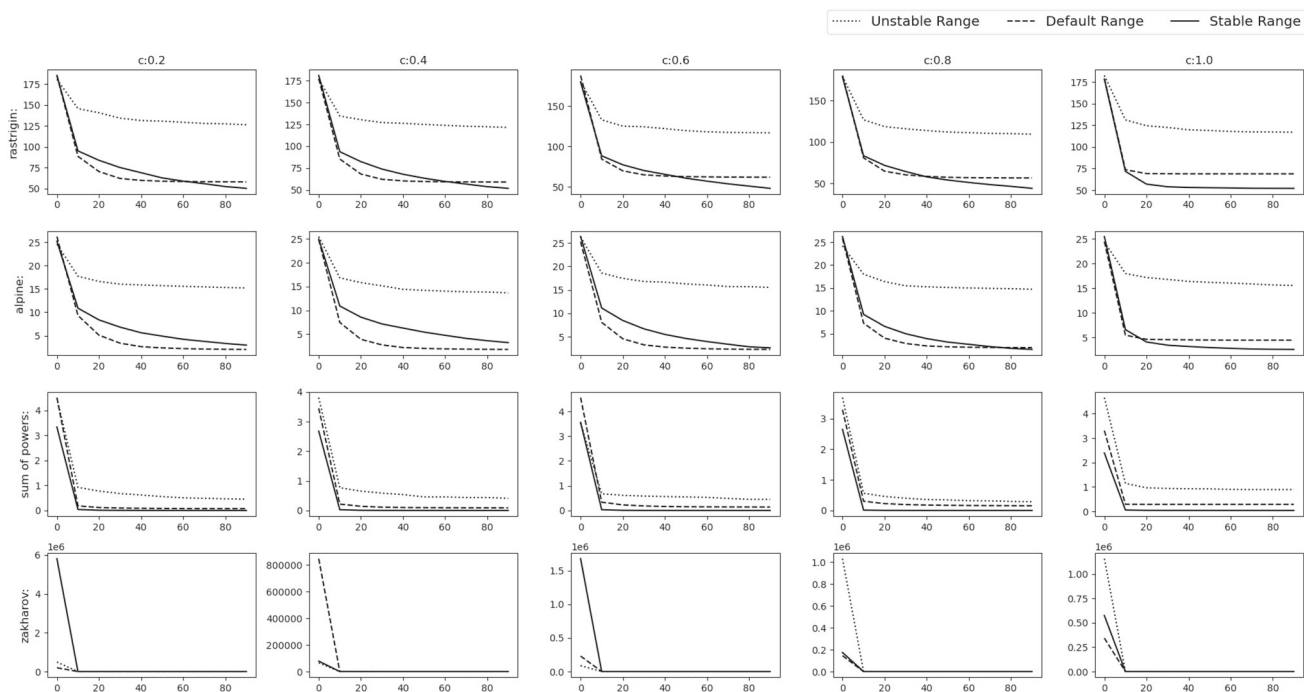


Fig. 14 Experimental results of optimizing 10D Shifted functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

the worst result. Between default range and stable range, the stable range converges a bit slower than default range in some cases, but it always goes on to find a better value than the default range in all the functions. The controlled randomness

may be the reason. In most of the algorithms, it is assumed that the range of random parameters should be (0,1), but the relationship and dependencies among the parameter is ignored, which should not be the case.

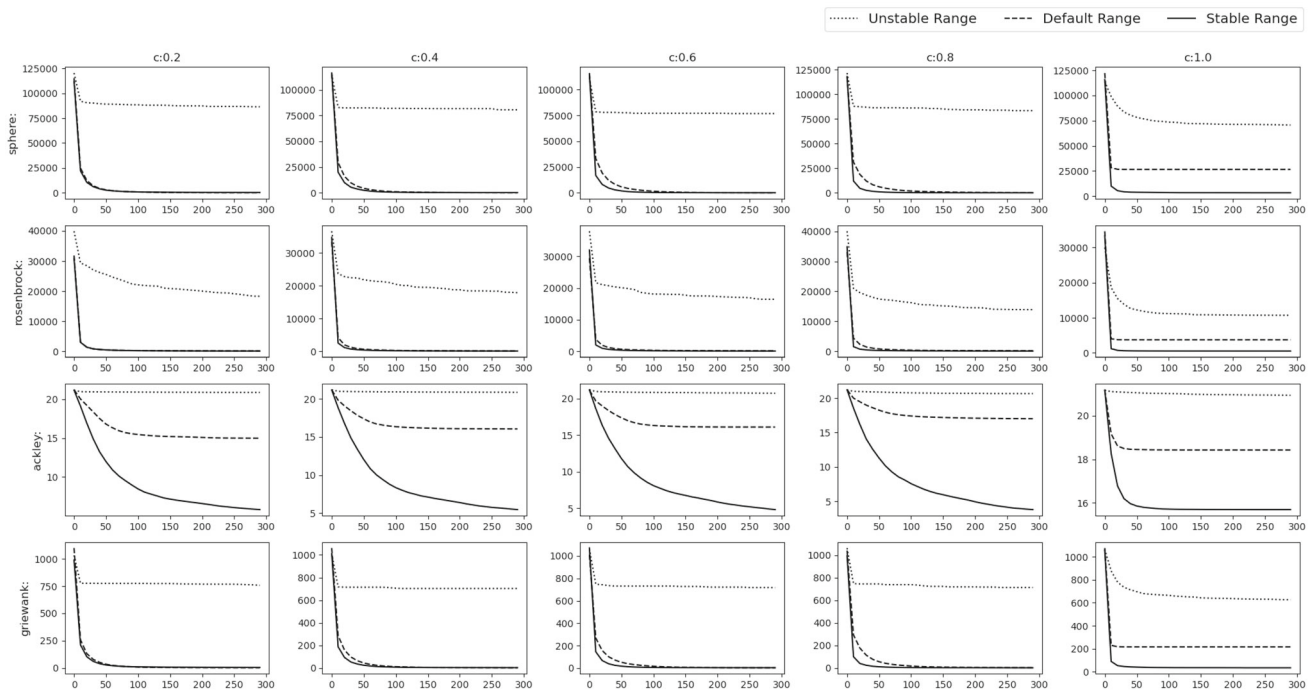


Fig. 15 Experimental results of optimizing 30D Shifted functions. The solid,dashed and dotted lines denote stable range, default range and unstable range respectively

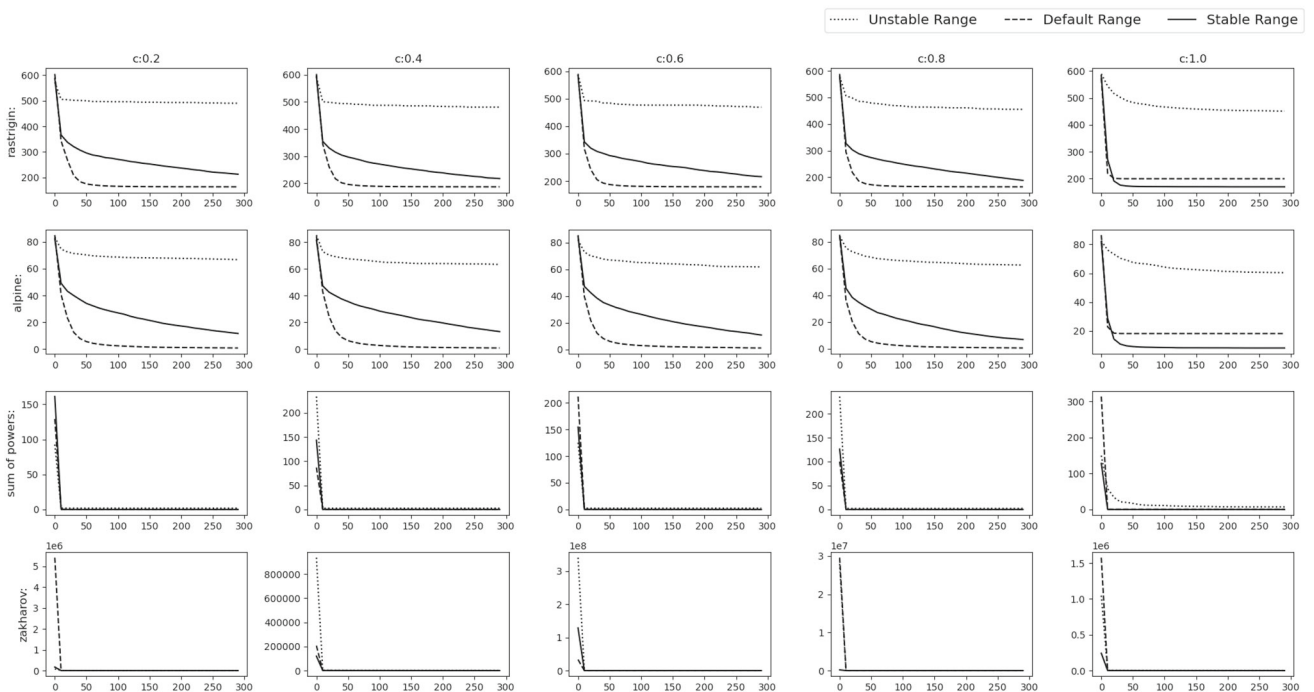


Fig. 16 Experimental results of optimizing 30D Shifted functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

Shifted and rotated numerical benchmark functions

In this scenario the *gbest* or the *o* vector is shifted to a different position randomly chosen inside the D-dimensional

space defined by the range of A_j . Random rotation is done i.e. M is set as a random D-dimensional rotation matrix. From Figs. 17, 18 and Table 6 could be observed that stable range performed better than the other unstable ranges for the 10D

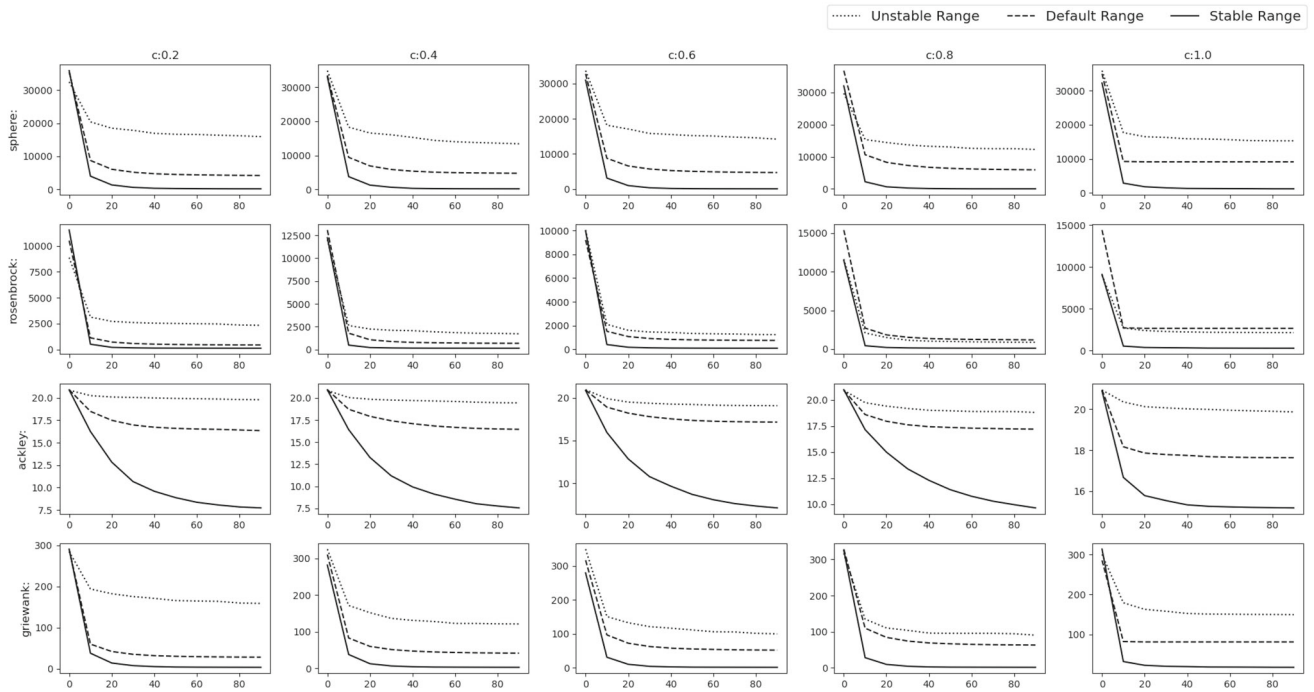


Fig. 17 Experimental results of optimizing 10D Shifted and Rotated functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

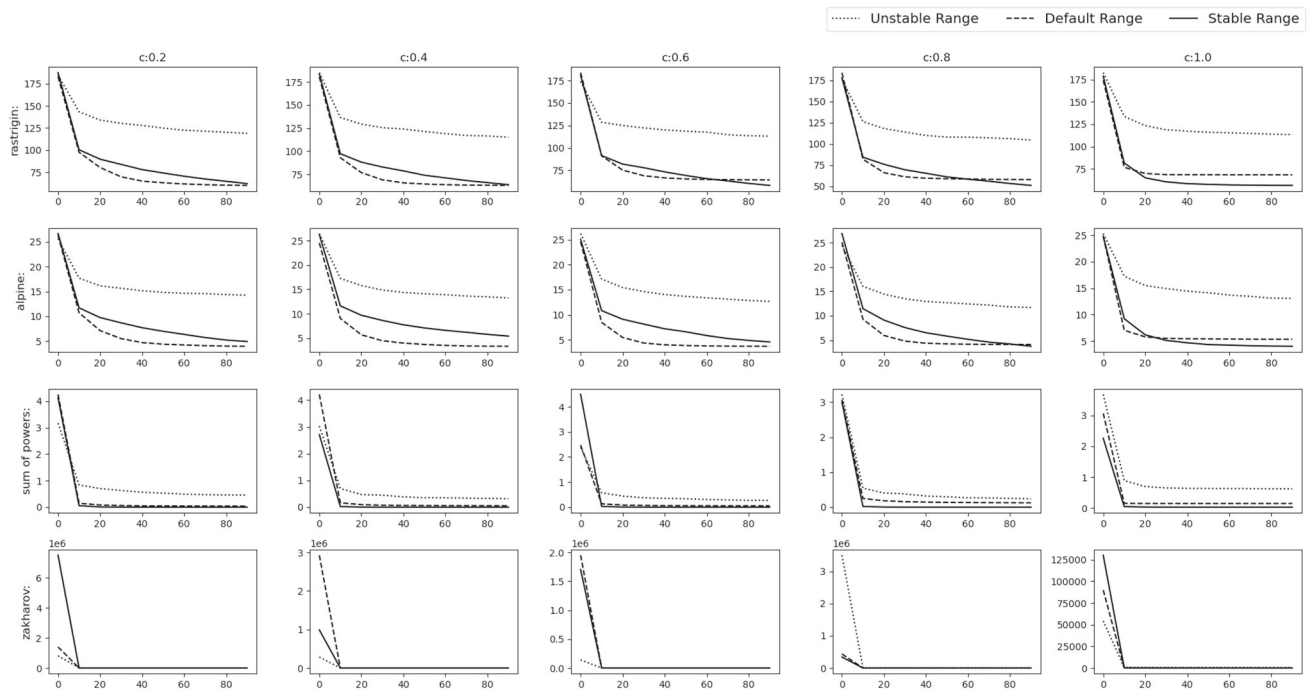


Fig. 18 Experimental results of optimizing 10D Shifted and Rotated functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

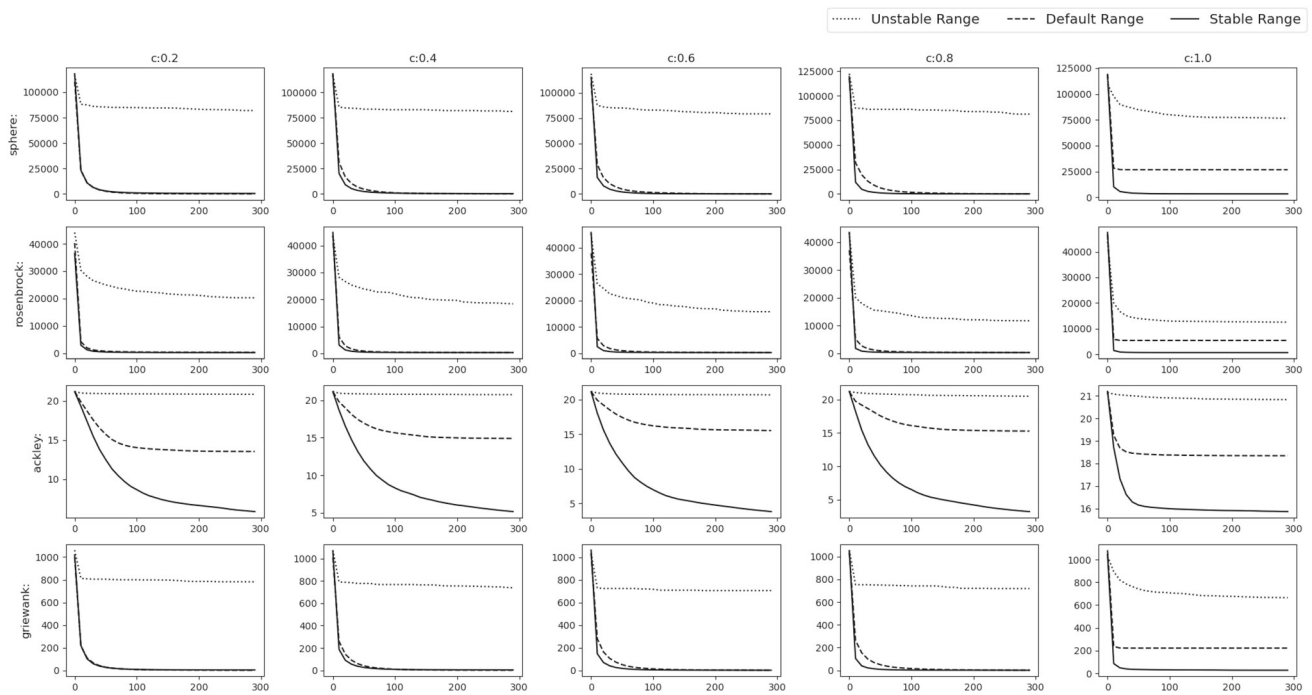


Fig. 19 Experimental results of optimizing 30D Shifted and Rotated functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

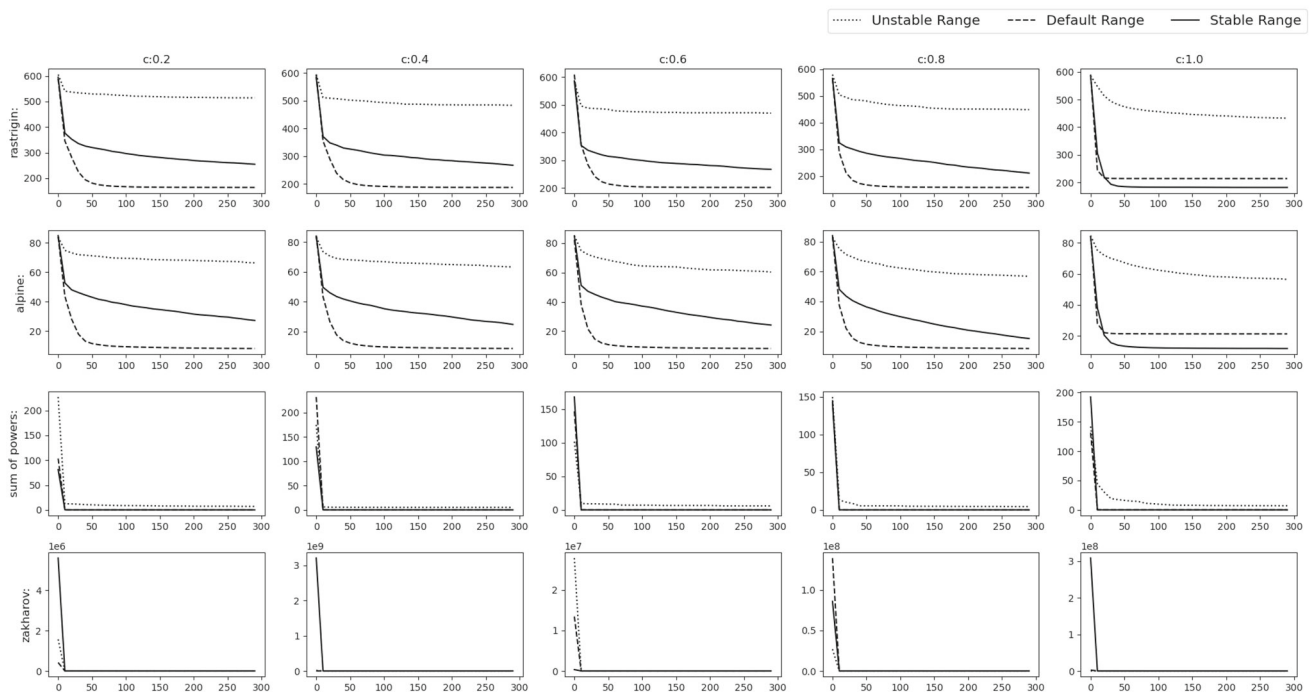


Fig. 20 Experimental results of optimizing 30D Shifted and Rotated functions. The solid, dashed and dotted lines denote stable range, default range and unstable range respectively

experiment and in Figs. 19, 20 and Table 7 for the 30D experiment same observations were obtained. Here also unstable range performed worst and stable range always converged.

Conclusion

Stability analysis helps in determining the reliability factor of an algorithm. In this paper, Von Neumann stability analy-

sis procedure was used to determine the appropriate range of parameters for which SGO algorithm always converges. The results were supported experimentally with suitable figures and tables. From the analysis, it is deduced that unstable range of parameters of an algorithm may degrade its performance and should be avoided. Moreover, dependencies and relationship between the parameters of an algorithm should be determined because they contribute a lot to the performance and stability analysis provides a way to solve the purpose. This theoretical analysis procedure could be combined with other experimental methodologies in determining the robustness of any algorithm and thus minimizing the chance of failure. Stability analysis of many recent algorithms such as monarch butterfly optimization (MBO) [23], earthworm optimization algorithm (EWA) [24], elephant herding optimization (EHO) [25], moth search (MS) algorithm [26], Slime mould algorithm (SMA) [27], Harris hawks optimization (HHO) [28] and Past Present Future optimization (PPF) [29] which claims to provide promising results experimentally, yet cannot be found in the literature. So, this paves a way for the researchers to work upon more such algorithms.

Declarations

Conflict of interest The authors have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks (vol 4, pp 1942-1948), IEEE
- Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697
- Satapathy S, Naik A (2016) Social group optimization (SGO): a new population evolutionary optimization technique. *Compl Intell Syst* 2(3):173–203
- Naik A, Satapathy SC, Ashour AS, Dey N (2018) Social group optimization for global optimization of multimodal functions and data clustering problems. *Neural Comput Appl* 30(1):271–287
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Dey N, Rajinikanth V, Ashour AS, Tavares JMR (2018) Social group optimization supported segmentation and evaluation of skin melanoma images. *Symmetry* 10(2):51
- Rajinikanth V, Satapathy SC (2018) Segmentation of ischemic stroke lesion in brain MRI based on social group optimization and Fuzzy-Tsallis entropy. *Arab J Sci Eng* 43(8):4365–4378
- Praveen SP, Rao KT, Janakiramaiah B (2018) Effective allocation of resources and task scheduling in cloud environment using social group optimization. *Arab J Sci Eng* 43(8):4265–4272
- Dey N, Rajinikanth V, Shi F, Tavares JMR, Moraru L, Karthik KA, Emmanuel C (2019) Social-Group-Optimization based tumor evaluation tool for clinical brain MRI of Flair/diffusion-weighted modality. *Biocybern Biomed Eng* 39(3):843–856
- Dey N, Rajinikanth V, Fong SJ, Kaiser MS, Mahmud M (2020) Social group optimization-assisted Kapur's entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images. *Cogn Comput* 12(5):1011–1023
- Gopal A, Sultani MM, Bansal JC (2020) On stability analysis of particle swarm optimization algorithm. *Arab J Sci Eng* 45(4):2385–2394
- Gopal A, Bansal JC (2016) Stability analysis of differential evolution. In: 2016 international workshop on computational intelligence (IWCI) (pp 221-223), IEEE
- Bansal JC, Gopal A, Nagar AK (2018) Stability analysis of artificial bee colony optimization algorithm. *Swarm Evol Comput* 41:9–19
- Farivar F, Shoorehdeli MA (2016) Stability analysis of particle dynamics in gravitational search optimization algorithm. *Inf Sci* 337:25–43
- Biswas A, Das S, Abraham A, Dasgupta S (2010) Stability analysis of the reproduction operator in bacterial foraging optimization. *Theoret Comput Sci* 411(21):2127–2139
- Fang J, Zheng H, Liu J, Zhao J, Zhang Y, Wang K (2018) A transformer fault diagnosis model using an optimal hybrid dissolved gas analysis features subset with improved social group optimization-support vector machine classifier. *Energies* 11(8):1922
- Chakravarthy VS, Chowdary PSR, Satapathy SC, Terlapu SK, Anguera J (2018) Antenna array synthesis using social group optimization. In: *Microelectronics, electromagnetics and telecommunications* (pp 895–905). Springer, Singapore
- Naik A, Satapathy SC, Abraham A (2020) Modified Social Group Optimization-a meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Appl Soft Comput* 95:106524
- Das S, Saha P, Satapathy SC, Jena JJ (2020) Social group optimization algorithm for civil engineering structural health monitoring. *Eng Optim* 2020:1–20
- Verma S, Jena JJ, Satapathy SC, Rout M (2020) Solving travelling salesman problem using discreet social group optimization. *J Sci Ind Res* 79(10):928–930
- Satapathy SC, Naik A, Jena JJ (2021) Non-dominated Sorting Social Group Optimization algorithm for multiobjective optimization. *J Sci Ind Res* 80(02):129–136
- Liang J, Qu B, Suganthan P (2013) Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization
- Wang GG, Deb S, Cui Z (2019) Monarch butterfly optimization. *Neural Comput Appl* 31(7):1995–2014
- Wang GG, Deb S, Coelho LDS (2018) Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. *Int Jo Bio-inspired Comput* 12(1):1–22
- Wang GG, Deb S, Coelho LD (2015) Elephant herding optimization. In: 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI) (pp 1–5), IEEE
- Wang GG (2018) Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput* 10(2):151–164

27. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. *Futur Gener Comput Syst* 111:300–323
28. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 1(97):849–72
29. Naik A, Satapathy SC (2021) Past present future: a new human-based algorithm for stochastic optimization. *Soft Comput* 25(20):12915–12976

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.