

FINDING MULTIPLE SOLUTIONS OF MULTIMODAL OPTIMIZATION USING SPIRAL OPTIMIZATION ALGORITHM WITH CLUSTERING

Kuntjoro Adji Sidarto, Adhe Kania and Novriana Sumarti

Institut Teknologi Bandung
Department of Mathematics
Jl. Ganesha 10, Bandung 40132
Indonesia

sidarto@math.itb.ac.id, adhe.kania@math.itb.ac.id, novriana@math.itb.ac.id

Abstract: A Multimodal optimization is one of the interesting problems in optimization which arises frequently in a wide range of engineering and practical applications. The goal of this problem is to find all of optimum solutions in a single run. Some algorithms fail to find all solutions that have been proven their existence analytically. In our paper [1], a method is proposed to find the roots of a system of non-linear equations using a clustering technique that combine with Spiral Optimization algorithm and Sobol sequence of points. An interesting benefit using this method is that the same inputs will give the same results. Most of the time this does not happen in meta-heuristic algorithms using random factors. Now the method is modified to find solutions of multimodal optimization problems. Generally in an optimization problem, the differential form of the objective function is needed. In this paper, the proposed method is to find optimum points of general multimodal functions that its differential form is not required. Several problems with benchmark functions have been examined using our method and they give good result.

Keywords: Multimodal optimization, multiple optimal solutions, spiral optimization algorithm, clustering, Sobol sequence of points.

1 Introduction

Many problems in the real world can be modelled as optimization problems. Some variation of functions that increase their complexity may be needed as the objective function of multimodal optimization problems. The function can have numerous global and local maximum values. Unfortunately, most of optimization methods only give a single solution at one run. It could give other solutions if the initial points are changed. To guess the initial points is not an easy task either. Recently, metaheuristic methods inspired by nature can get results without dependence with initial points, but it must be run in a number of times to obtain numerous solutions.

There are many approaches in solving the multimodal optimization problems. In [2], the Feature Selection (FSel) problem that reduce the pattern dimensionality in data pre-processing is being solved using Multimodal Optimization (MO) techniques. The FSel problem is a high-dimensional optimization problem in the nature and thus needs a solver with high exploration power. MO's solution conservation capability can find its multiple suitable solutions in a single run.

In [3], the advanced multimodal Particle Swarm Optimization (PSO) algorithms adopted into the species-based PSO (SPSO), the fitness Euclidean-distance ratio based PSO (FER-PSO), the ring topology based PSO and the Euclidean distance-based locally informed particle swarm (LIPS) optimizer. These algorithms are applied to a multimodal buckling maximization problem of composite panels and identify simultaneously not only the first-best-fitness solutions (the global optima) but also the second-best-fitness solutions (the global sub-optima) to this buckling optimization design in a single optimization process.

In [4], Differential Evolution (DE) algorithm was successfully extended with multi-population and multi-resolution approach to locate multiple optima of a multimodal function. In this method, the beneficial of DE, an efficient global optimization method, was combined with multi-population in exploration process and multi-resolution to control initialization step. One and two dimensional benchmark functions is applied to prove the effectivity of this method.

In [5], the authors proposed a framework, called MOMCA (multimodal optimization for model calibration), which is based on niching genetic algorithms. It provides a set of alternative calibration solutions, which is useful to analyse the parameters, model's response, and doing sensitivity analysis. The core component of MOMCA is its niching genetic algorithm, able to reach various optima in multimodal optimization problems by keeping the necessary diversity. The framework is successfully applied to a biological growth model and a managerial model to improve brand equity.

In [6], the stochastic multimodal optimization problems and stochastic resource allocation problem are being solved using a hybrid approach comprising particle swarm optimization (PSO), constrictive factor or PSO (CPSO), and elite group optimal computing budget allocation (EGOCBA). CPSO or PSO is applied to determine the correct direction in the design space, and EGOCBA is adopted to allocate the appropriate number of samples to each alternative and

provide reliable evaluations and identifications to rank particles in the CPSO or PSO procedure. The approach has efficiently and effectively derived multiple optima of sets in a multimodal class and stochastic environment.

In this paper, we propose a new clustering method to get all of optimum values, both local and global, of multimodal function in a single run. We use a nature-inspired metaheuristics method based on the spiral optimization, or in short SPO (Spiral Optimization) as in [7, 8]. A stability analysis at the dynamic center of the SPO model has been studied in [8], and it suggests a method for setting the convergence rate parameter. The method has attracted many modeller due to the simplicity to program, short computing time, easy to implement, and the possibility to apply to various applications [9]. One of the improvement proposed in [9], is LAS DA (linear adaptive spiral dynamics algorithm), which is an improved version of SDA (adaptive spiral dynamics algorithm), where the spiral radius and angular displacement are dynamically varied by employing novel mathematical equation based on a linear function. This function establishes a relationship among fitness value, spiral radius and angular displacement. The proposed algorithm is tested with various types of multimodal and unimodal benchmark functions. The results show a better performance in finding an optimal solution for the benchmark functions in comparison with other SDA approaches. The paper [1] and this paper propose an improvement by implementing the clustering technique. There is also a new technique in this paper in dealing with more general multimodal optimization problems, where their differential form of the objective functions is not existing or difficult to obtain. Some benchmark multimodal functions from [10] have been examined, and the proposed method has successfully found numerous possible solutions for those functions.

In the next section a review of Spiral Optimization Algorithm and the proposed clustering technique for locating multiple solutions are given. Section 3 presents numerical experiments on some benchmark functions. Finally some conclusions are given in Section 4.

2 Methodology

2.1 Spiral Optimization Algorithm

Consider an optimization problem of the form

$$\text{maximize } F(\mathbf{x}) \tag{1}$$

with $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in D = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \subset \mathbb{R}^n$ and $F : D \rightarrow \mathbb{R}$. Here F has multiple peaks in D .

SPO algorithm is a metaheuristic method for continuous optimization problems, developed by Tamura and Yasuda [7], based on the analogy of spiral phenomena in nature. Firstly we review 2-dimensional SPO before discuss n -dimensional SPO, as some results are just extension of 2-dimensional SPO.

The multiplication of a vector $\mathbf{x} \in \mathbb{R}^2$ in the plane by matrix $S_2(r, \theta) = \text{diag}_2(r, r) \cdot R^{(2)}$ where $\text{diag}_2(r, r) = \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix}$ and $R^{(2)} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$ is a rotation matrix in the plane, gives the geometric effect by rotating \mathbf{x} anticlockwise through the angle θ ($0 < \theta \leq 2\pi$), and shortening the vector $R^{(2)}\mathbf{x}$ by a factor of r for $0 < r < 1$. Hence the equation

$$\mathbf{x}_{k+1} = S_2(r, \theta) \mathbf{x}_k \quad \text{where } \mathbf{x}_k = \mathbf{x}(k) = (x_1(k), x_2(k))^T \quad k = 0, 1, 2, \dots$$

describes the transformation of an initial point $\mathbf{x}_0 \in \mathbb{R}^2$ repeatedly by $\mathbf{x} \mapsto S_2(r, \theta)\mathbf{x}$. The graph of the sequence of points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ in the plane will produce a trajectory of spiral form. So, it seems that the point \mathbf{x}_0 goes spiral inward and toward the origin for $0 < r < 1$. The spiral model described earlier has a center at the origin. It can be extended to have center at an arbitrary point \mathbf{x}^* by translating the origin toward the point \mathbf{x}^* in Eq.(1):

$$\mathbf{x}_{k+1} - \mathbf{x}^* = S_2(r, \theta)(\mathbf{x}_k - \mathbf{x}^*) \quad \text{such that } \mathbf{x}_{k+1} = S_2(r, \theta)\mathbf{x}_k - (S_2(r, \theta) - I_2)\mathbf{x}^* \quad k = 0, 1, 2, \dots$$

The iterated point $\mathbf{u}_k = \mathbf{u}(k) = \mathbf{x}_k - \mathbf{x}^* \quad k = 0, 1, 2, \dots$ will move spirally inward and toward the origin. Hence, the iterated point \mathbf{x}_0 will move spirally inward and toward the point \mathbf{x}^* in the plane.

Let us consider the optimization problem

$$\text{maximize}_{\mathbf{x} \in \mathbb{R}^2} F(\mathbf{x}) \quad \text{with } \mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$$

Based on the previous spiral model, the SPO algorithm in the plane is developed as follows.

Input : $m (\geq 2)$ the number of search points, θ ($0 < \theta \leq 2\pi$), r ($0 < r < 1$), k_{\max} maximum number of iteration

Process : 1. Randomly generate initial points $\mathbf{x}_i(0) \in \mathbb{R}^2 \quad i = 1, 2, \dots, m$ in the feasible region.

2. Set $k = 0$.

3. Find $\mathbf{x}^* = \mathbf{x}_{i_g}(0)$ where $i_g = \arg \max_i F(x_i(0)) \quad i = 1, 2, \dots, m$
4. Update $\mathbf{x}_i : \mathbf{x}_i(k+1) = S_2(r, \theta)\mathbf{x}_k - (S_2(r, \theta) - I_2)\mathbf{x}^* \quad k = 0, 1, 2, \dots, m$
5. Update $\mathbf{x}^* : \mathbf{x}^* = \mathbf{x}_{i_g}(k+1), \quad i_g = \arg \max_i F(x_i(k+1)) \quad i = 1, 2, \dots, m$
6. If $k = k_{\max}$ then stop. Otherwise, set $k = k + 1$ and return to step 4.

Output : \mathbf{x}^* as a maximum point of $F(\mathbf{x})$.

The extension to n -dimensional SPO may be done with the help of rotation matrix in n -dimensional which is performed in a similar way as the 2-dimensional rotation taking two-dimensions at a time. Let $R_{ij}^{(n)}$ be $n \times n$ matrix with entries $r_{ii} = r_{jj} = \cos \theta, r_{ij} = -\sin \theta,$ and $r_{ji} = \sin \theta,$ and for other entries, $s \neq i, t \neq j, r_{st} = \delta_{st},$ where $\delta_{st} = 1$ if $s = t$ and $\delta_{st} = 0$ if $s \neq t$. Thus $R_{ij}^{(n)}$ resemble the identity matrix with exception for ii, ij, jj and ji positions. Multiplication of a vector by rotation matrix $R_{ij}^{(n)}$ only alter i^{th} and j^{th} components of the vector with no effect on other components. Hence, $R_{ij}^{(n)}$ acts as the plane rotation. Let $R^{(n)}$ be $n \times n$ matrix defined by

$$R^{(n)} = \prod_{i=1}^{n-1} \left(\prod_{j=1}^i R_{n-i, n+1-j}^{(n)} \right)$$

which is a composition of plane rotations matrix $R_{ij}^{(n)}$. The n -dimensional spiral model with center at the origin is formulated as $\mathbf{x}(k+1) = S_n(r, \theta)\mathbf{x}(k)$ where $S_n(r, \theta) = \text{diag}_n(r, r, \dots, r) \cdot R^{(n)}$ and the n -dimensional spiral model with center at \mathbf{x}^* is formulated as : $\mathbf{x}(k+1) = S_n(r, \theta)\mathbf{x}(k) - (S_n(r, \theta) - I_n)\mathbf{x}^*$.

Let us consider the optimization problem

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{maximize}} F(\mathbf{x}) \quad \text{with} \quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$$

The n -dimensional SPO algorithm is written below.

Input : $m (\geq 2)$ the number of search points, $\theta (0 < \theta \leq 2\pi), r (0 < r < 1), k_{\max}$ maximum number of iteration

Process : 1. Randomly generate initial points $\mathbf{x}_i(0) \in \mathbb{R}^n \quad i = 1, 2, \dots, m$ in the feasible region.

2. Set $k = 0$.
3. Find $\mathbf{x}^* = \mathbf{x}_{i_g}(0)$ where $i_g = \arg \max_i F(x_i(0)) \quad i = 1, 2, \dots, m$
4. Update $\mathbf{x}_i : \mathbf{x}_i(k+1) = S_2(r, \theta)\mathbf{x}_k - (S_2(r, \theta) - I_2)\mathbf{x}^* \quad k = 0, 1, 2, \dots, m$
5. Update $\mathbf{x}^* : \mathbf{x}^* = \mathbf{x}_{i_g}(k+1), \quad i_g = \arg \max_i F(x_i(k+1)) \quad i = 1, 2, \dots, m$
6. If $k = k_{\max}$ then stop. Otherwise, set $k = k + 1$ and return to step 4.

Output : \mathbf{x}^* as a maximum point of $F(\mathbf{x})$.

2.2 A Clustering Technique for Locating Multiple Solution Points

In general, problem (1) may have many solutions corresponding to local minimum and global maximum points of $F(\mathbf{x})$. A single run of SPO algorithm can only obtain a single global maximum point. In an attempt to locate other solutions in a single run, a clustering technique is used to produce a number of clusters of points potentially containing the local and global maximum points. After that, the SPO algorithm is applied simultaneously to each cluster. Note that a cluster with center at \mathbf{x} and radius of ρ is the set of all points \mathbf{y} satisfying $\|\mathbf{x} - \mathbf{y}\| < \rho$.

In step 1 of SPO algorithm in Section 2.1, an initial population of points is generated randomly. One difficulty with random numbers is that the obtaining numbers may fail to distribute ‘uniformly’ in the sense of equidistributional (see [11]). As the consequence, the generated initial population of points may not uniformly distributed in the feasible region of the problem. It is much more desirable to have the uniform distribution of initial population of points in order to obtain as much as possible candidate solutions in the clustering phase. It will be helpful if the generated population of points in the search region has minimum deviation from uniformity (also called discrepancy). Various low discrepancy sequence of points have been constructed and Sobol sequence is one of them. The usefulness of Sobol sequence of points in finding multiple solutions of system of nonlinear equations in a single run have been shown in [1]. Considering the advantage of having low discrepancy sequence of points rather than pseudo-random sequence of points, we propose the usage of Sobol sequence of points as initial points $\mathbf{x}_i(0) \in \mathbb{R}^n$ both in step 1 of SPO algorithm and of

Clustering phase. In this paper, we use the algorithm developed by Joe and Kuo [12]. This no-random technique will make this algorithm always give the same results from same input. Therefore we can concentrate to revise the parameter input without the need to repeatedly run the program in a large number of iterations.

Below we give more detail on the clustering technique.

Input : $m_{cl}, r_{cl}, \theta_{cl}, k_{cl}$ parameters for SPO algorithm at diversification phase

ε ($0 < \varepsilon < 1$) parameter for optimum points acceptance

δ ($0 < \delta < 1$) parameter to distinguish between one candidate optimum and another one in case they are very close each other

m, r, θ, k_{max} parameters for SPO algorithm at intensification phase

Process :

Diversification phase

1. Generate m_{cl} Sobol sequence of points as initial points $\mathbf{x}_i(0) \in \mathbb{R}^n$, $i = 1, 2, \dots, m_{cl}$ in the feasible region $D = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \subset \mathbb{R}^n$, and set $k = 0$.
 2. Set $\mathbf{x}^* = \mathbf{x}_{i_g}(0)$, $i_g = \arg \max_i F(\mathbf{x}_i(0))$, $i = 1, 2, \dots, m_{cl}$ as center of the first cluster with radius equal to $\frac{1}{2} \left(\min_l |b_l - a_l| \right)$, $l = 1, 2, \dots, n$.
 3. For $i = 1, 2, \dots, m_{cl}$ do :
If \mathbf{x}_i is not the center of existing cluster, then do the function cluster below with \mathbf{x}_i as input.
Function Cluster (input \mathbf{y})
 - a. Find a cluster with its center nearest to \mathbf{y} . Let C be that cluster, with centre at \mathbf{x}_C .
 - b. Set \mathbf{x}_m as mid-point between \mathbf{y} and \mathbf{x}_C .
 - c. Compare $F(\mathbf{y})$, $F(\mathbf{x}_C)$, and $F(\mathbf{x}_m)$:
 - If $F(\mathbf{x}_m) < F(\mathbf{y})$ and $F(\mathbf{x}_m) < F(\mathbf{x}_C)$: Set a new cluster with center at \mathbf{y} and radius equal to the distance between \mathbf{y} and \mathbf{x}_m .
 - Else, if $F(\mathbf{x}_m) > F(\mathbf{y})$ and $F(\mathbf{x}_m) > F(\mathbf{x}_C)$: Set a new cluster with \mathbf{y} as its center and radius equal to the distance between \mathbf{y} and \mathbf{x}_m . Redo *Function Cluster* with \mathbf{x}_m as its input.
 - Else, if $F(\mathbf{y}) > F(\mathbf{x}_C)$, set \mathbf{y} as the centre of C .
 - d. Change the radius of C equal to the distance between \mathbf{y} and \mathbf{x}_m .
4. Set $\mathbf{x}^* = \mathbf{x}_{i_g}$ where $i_g = \arg \max_i F(\mathbf{x}_i(k))$, $i = 1, 2, \dots, m_{cl}$.
5. Update \mathbf{x}_i : $\mathbf{x}_i(k+1) = S_n(r_{cl}, \theta_{cl})\mathbf{x}_i(k) - (S_n(r_{cl}, \theta_{cl}) - I_n)\mathbf{x}^*$, $i = 1, 2, \dots, m_{cl}$, and set $k = k + 1$.
6. Do steps 3 to 5 k_{cl} times

Intensification phase

7. Having done diversification phase, we obtain a number of clusters : C_1, C_2, \dots, C_{n_c} . Each cluster has center at \mathbf{x}_{C_i} and radius of ρ_i ($i = 1, 2, \dots, n_c$). To each cluster, perform SPO algorithm to obtain a candidate of maximum point in each cluster. Use m, r, θ, k_{max} as SPO input in this phase, with $D_i = [x_{1,i} - \rho_i, x_{1,i} + \rho_i] \times [x_{2,i} - \rho_i, x_{2,i} + \rho_i] \times \dots \times [x_{n,i} - \rho_i, x_{n,i} + \rho_i] \subset \mathbb{R}^n$ as domain of the search region where $\mathbf{x}_{C_i} = (x_{1,i}, x_{2,i}, \dots, x_{n,i})^T$, $i = 1, 2, \dots, n_c$.

Final selection

8. Keep only candidate maximum points which satisfy condition $F(\mathbf{x} - \varepsilon) < F(\mathbf{x})$ and $F(\mathbf{x} + \varepsilon) < F(\mathbf{x})$.
9. Suppose from step 8 there result n_g candidate maximum points. From these n_g candidates, select only those which satisfy $\|\mathbf{x}_i - \mathbf{y}_j\| > \delta$, $i, j = 1, 2, \dots, n_g$ where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the distance between the candidates \mathbf{x}_i and \mathbf{x}_j . In case where $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta$, select only \mathbf{x}_i as a maximum point if $F(\mathbf{x}_i) \geq F(\mathbf{x}_j)$, otherwise select \mathbf{x}_j as a maximum point.

Output : all candidates from step 9 that meet final selection become the maximum points.

Note : to solve minimization problems, we run the same algorithm above with $G(\mathbf{x}) = -F(\mathbf{x})$ as the objective function.

In some cases, it is intended to obtain all of global optimum points only, without local optimum points. For this case, we do some modifications in the algorithm in section 2.2 to get global optimum only. See problems 4 and 5 in Section 3. The modifications are written below.

- Add γ ($0 < \gamma < 1$) ‘cut-off’ parameter as additional input
- In step 3, add criteria $F(\mathbf{x}_i) > \gamma F(\mathbf{x}^*)$ before execute *Function Cluster*
- In step 8, add condition $F(\mathbf{x}) > (1 - \gamma)F(\mathbf{x}^*)$

3 Numerical Experiments

Several test cases from various problems, so called benchmark problems, have been examined to validate our proposed method. All numerical experiments were performed on a desktop computer equipped with processor Intel Core™ i5 with 4 GB ram and 3.2 GHz CPU running Linux Mint 17.2. The code was written in C++ and compiled using g++.

3.1 Problem 1

Consider the 2-dimensional second minima function as follows

$$g(x, y) = \frac{1}{2}(x^4 - 16x^2 + 5x) + \frac{1}{2}(y^4 - 16y^2 + 5y) \text{ where } D = \{(x, y)^T \in \mathbb{R}^2 \mid -4 \leq x \leq 4, -4 \leq y \leq 4\} \quad (2)$$

Using parameters $m_{cl} = 300$, $r_{cl} = 0.95$, $\theta_{cl} = \pi/4$, $k_{cl} = 10$, $\varepsilon = 10^{-7}$, $\delta = 0.1$, $m = 200$, $k_{max} = 200$, $r = 0.95$, and $\theta = \pi/4$, the results are obtained in 0.65s and shown in Table 2.

Table 2 : The optimum points for function (2)

No	x	y	g(x,y)	No	x	y	g(x,y)
Minimum Max				imum			
1	-2.90353	-2.90353	-78.3323	1	0.156731	0.156731	0.391225
2	-2.90353	2.7468	-64.1956				
3	2.7468	-2.90353	-64.1956				
4	2.7468	2.7468	-50.0589				

These optimum points can be confirmed by observing the graph in Figure 1 .

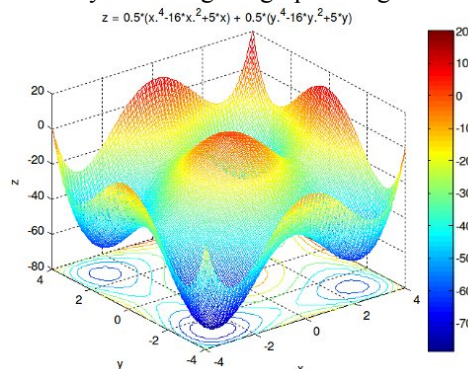


Figure 1: 2-dimensional second minima function

3.2 Problem 2

Consider the Six Hump Camel Back function as follows:

$$g(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (4y^2 - 4)y^2, \text{ where } D = \{(x, y) \mid -1.9 \leq x \leq 1.9, -1.1 \leq y \leq 1.1\} \quad (3)$$

Based on [10], it has two global optima and two local optima. Using parameters $m_{cl} = 1000$, $r_{cl} = 0.99$, $\theta_{cl} = \pi/2$, $k_{cl} = 20$, $\varepsilon = 10^{-5}$, $\delta = 0.1$, $m = 200$, $k_{max} = 200$, $r = 0.95$, and $\theta = \pi/4$, the results are obtained in 5.14s. They consist of 2 couples of global optima, which are 2 global maximum points and 2 global minimum points, and 2 couples of local optima, which are 4 local minimum points with 2 different values of $g(x, y)$.

Table 3: The optimum points for Six Hump Camel Back function (3)

No	x	y	g(x,y)	No	x	y	g(x,y)
Minimum Max				imum			
1	-1.70361	0.796084	-0.215464	-	1.23024	-0.162328	2.4963

2 -	1.6071	-0.568651	2.10425	2	1.23023	0.162334	2.4963
3 -	0.089842	0.712656	-1.03163				
4 0	.089842	-0.712656	-1.03163				
5 1	.6071	0.568651	2.10425				
6 1	.70361	-0.796084	-0.215464				

3.3 Problem 3

Consider the n -dimensional Rastrigin function with our chosen domain as follows:

$$g(\mathbf{x}) = \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) + 10, \text{ where } D = \{ \mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n \mid -1 \leq x_i \leq 1, i = 1, \dots, n \} \quad (4)$$

For $n = 2$, the chosen parameters are $m_{cl} = 500$, $r_{cl} = 0.95$, $\theta_{cl} = \pi / 4$, $k_{cl} = 10$, $\varepsilon = 10^{-6}$, $\delta = 0.1$, $m = 200$, $k_{max} = 200$, $r = 0.95$, and $\theta = \pi / 4$, and the results shown in Table 4 are obtained only in 0.53 s. It contains 4 global maximum points for $g(\mathbf{x}) = 40.5025$, a global minimum point, and 2 sets containing four points of local minimum points where $g(\mathbf{x}) = 0.994959$ and $g(\mathbf{x}) = 1.98992$ respectively.

Table 4: The optimum points for 2-dimensional Rastrigin function (4)

No	x_1	x_2	$g(\mathbf{x})$	No	x_1	x_2	$g(\mathbf{x})$
Minimum Max				imum			
1 -	0.994959	-0.994959	1.98992	1	-0.502545	-0.502546	40.5025
2 -	0.994959	1.6366e-09	0.994959	2	-0.502537	0.502538	40.5025
3 -	0.994959	0.994959	1.98992	3	0.502546	-0.502545	40.5025
4 1	.06667e-09	-0.994959	0.994959	4	0.502548	0.502544	40.5025
5 0		0	0				
6 -	5.10989e-09	0.994959	0.994959				
7 0	.994959	-0.994959	1.98992				
8 0	.994959	1.20281e-09	0.994959				
9 0	.994959	0.994959	1.98992				

For $n = 3$, the chosen parameters are $m_{cl} = 20000$, $r_{cl} = 0.95$, $\theta_{cl} = \pi / 4$, $k_{cl} = 20$, $\varepsilon = 10^{-5}$, $\delta = 0.1$, $m = 200$, $k_{max} = 200$, $r = 0.95$, and $\theta = \pi / 4$, and the results shown in Table 5 are obtained in 40.39 s. It consists of 8 global maximum points, a global minimum point, and 3 sets of local minimum points containing 1, 5 and 5 points respectively.

Table 5: The optimum points for 3-dimensional Rastrigin function (4)

No	x_1	x_2	x_3	$g(\mathbf{x})$	No	x_1	x_2	x_3	$g(\mathbf{x})$
Minimum Max					imum				
1	-0.994957	-0.99496	-0.994959	2.98488	1	-0.502544	-0.502554	-0.502541	60.7538
2	-0.994959	3.25509e-06	-1.10394e-06	0.994959	2	0.502514	-0.502551	0.502528	60.7538
3 -	4.14247e-07	-0.994957	-0.99496	1.98992	3	-0.502576	0.502551	-0.50251	60.7538
4	-1.65449e-08	-0.994959	-2.49482e-08	0.994959	4	0.502537	0.502541	0.502545	60.7538
5	6.19158e-07	-0.994959	0.994959	1.98992	5	0.502525	-0.502547	-0.502529	60.7538
6	-2.2665e-08	3.28811e-08	-0.994959	0.994959	6	0.502574	-0.502550	0.502525	60.7538
7	0 0 0			0	7	0.502526	0.502573	-0.502552	60.7538
8 2	.12297e-06	0.994959	-0.994958	1.98992	8 0	.502563	0.502542	0.502535	60.7538
9	4.95836e-08	0.994959	1.7007e-08	0.994959					
10	-4.1896e-08	0.994956	0.994957	1.98992					
11	0.994959	-4.88344e-08	2.27944e-08	0.994959					
12	0.994959	0.994959	1.72968e-06	1.98992					

3.4 Problem 4

Consider the 2-dimensional Vincent function as follows:

$$g(x, y) = \frac{1}{2}(\sin(10 \log(x)) + \sin(10 \log(y))), \text{ where } D = \{ (x, y)^T \in \mathbb{R}^2 \mid 0.25 \leq x \leq 10, 0.25 \leq y \leq 10 \} \quad (5)$$

The function with the general dimension d in $[10]$ has 6^d global optima and no local optima. Using parameter $m_{cl} = 1000$, $r_{cl} = 0.95$, $\theta_{cl} = \pi / 4$, $k_{cl} = 10$, $\varepsilon = 10^{-5}$, $\delta = 0.01$, $\gamma = 0.2$, $m = 150$, $k_{max} = 150$, $r = 0.95$, and $\theta = \pi / 4$, we obtain $6^2 = 36$ maximum points in only 4.34s, which are shown in Table 6. The maximum value $g(x, y)$ is 1 (one) for all points.

Table 6: The optimum points for Vincent function (5)

No	x	y	No	x	y	No	x	y	No	x	y
1	0.333018	0.333018	10	0.624228	2.19328	19	2.19328	0.333018	28	4.11121	2.19328
2	0.333018	0.624228	11	0.624228	4.11121	20	2.19328	0.624228	29	4.11121	4.11121
3	0.333018	1.17009	12	0.624228	7.70628	21	2.19328	1.17009	30	4.11121	7.70628
4	0.333018	4.11121	13	1.17009	0.333018	22	2.19328	2.19328	31	7.70628	0.333018
5	0.333018	4.11121	14	1.17009	0.624228	23	2.19328	4.11121	32	7.70628	0.624228
6	0.333018	7.70628	15	1.17009	1.17009	24	2.19328	7.70628	33	7.70628	1.17009
7	0.624228	0.333018	16	1.17009	2.19328	25	4.11121	0.333018	34	7.70628	2.19328
8	0.624228	0.624228	17	1.17009	4.11121	26	4.11121	0.624228	35	7.70628	4.11121
9	0.624228	1.17009	18	1.17009	7.70628	27	4.11121	1.17009	36	7.70628	7.70628

3.5 Problem 5

Consider the n -dimensional Shubert function as follows:

$$g(\mathbf{x}) = -\prod_{i=1}^n \sum_{j=1}^5 j \cos[(j+1)x_i + j], \text{ where } D = \{\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n \mid -10 \leq x_i \leq 10, i = 1, \dots, n\} \quad (6)$$

In [10], the function has $n3^n$ global optima and many local optima. For $n = 2$, we choose parameters $m_{cl} = 10000$, $r_{cl} = 0.95$, $\theta_{cl} = \pi/4$, $k_{cl} = 20$, $\varepsilon = 10^{-6}$, $\delta = 0.1$, $\gamma = 0.5$, $m = 200$, $k_{max} = 200$, $r = 0.95$, and $\theta = \pi/4$. There are 18 global maximum points with value 186.731 obtained only in 111.66 s. The results are shown in Table 7.

Table 7: The maximum points of Shubert function (6) with $n = 2$

No	x_1	x_2	No	x_1	x_2	No	x_1	x_2
1	-7.70831	-7.08351	7	-1.42513	-7.08351	13	4.85806	-7.08351
2	-7.70831	-0.80032	8	-1.42513	-0.80032	14	4.85806	-0.80032
3	-7.70831	5.48286	9	-1.42513	5.48287	5	4.85806	5.48286
4	-7.08351	-7.70831	10	0.80032	-7.70831	16	4.8286	-7.70831
5	-7.08351	-1.42513	11	0.80032	-1.42513	17	4.8286	-1.42513
6	-7.08351	4.85806	12	0.80032	4.85806	18	4.8286	4.85806

For $n = 3$, we choose parameters $m_{cl} = 50000$, $r_{cl} = 0.99$, $\theta_{cl} = \pi/4$, $k_{cl} = 100$, $\varepsilon = 10^{-2}$, $\delta = 0.3$, $\gamma = 0.5$, $m = 300$, $k_{max} = 300$, $r = 0.95$, and $\theta = \pi/4$. They are 81 global maximum points with value 2709.09 only in 1468.63 s.

4 Conclusion

Combination of the proposed clustering technique with SPO algorithm and Sobol sequence of points have been shown able to obtain the maximum and minimum points, both local and global for the benchmark functions in a single run for each problem. Optimization of Vincent $n = 2$ and Shubert $n = 2, 3$ functions gives the same numbers of global optima as in [10]. However, the result from Six Hump Camel Back function gives more points than in [10].

Acknowledgement: This work was partially supported by the P 3MI Industrial and Financial Mathematics Research Group, Institut Teknologi Bandung 2017.

References

- [1] Sidarto, K.A., Kania, A.: Finding all solutions of systems of nonlinear equations using spiral dynamics inspired optimization with clustering. *J. of Advanced Computational Intelligence and Intelligent Informatics (JACIII)* 19(5), 697-707 (2015). DOI 10.20965/jaciii.2015.p0697
- [2] Kamyab, S., Eftekhari, M.: Feature selection using multimodal optimization techniques. *Neurocomputing*, 171, 586-597 (2016). DOI 10.1016/j.neucom.2015.06.068
- [3] Huang, L., Ng, C., Sheikh, A.H., Griffith, M.C.: Niching particle swarm optimization techniques for multimodal buckling maximization of composite laminates. *Applied Soft Computing*, 57, 495-503 (2017). DOI 10.1016/j.asoc.2017.04.006
- [4] Zaharie, D.: A multipopulation differential evolution algorithm for multimodal optimization. In: R. Matousek (ed.) *Proceedings of 10th International Conference on Soft Computing – MENDEL 2004*, no. 10 in MENDEL, pp. 17-24. Brno University of Technology, VUT Press, Brno (2004)
- [5] Chica, M., Barranquero, J., Kajdanowicz, T., Damas, S., Cordon, O.: Multimodal optimization: An effective framework for model calibration. *Information Sciences*, 375, 79-97 (2017). DOI 10.1016/j.ins.2016.09.048

- [6] Chiu, C., Lin, J.T.: Novel hybrid approach with elite group optimal computing budget allocation for the stochastic multimodal problem, *Neurocomputing* (2017). DOI 10.1016/j.neucom.2017.04.041
- [7] Tamura, K., Yasuda, K.: Spiral Dynamics Inspired Optimization. *J. of Advanced Computational Intelligence and Intelligent Informatics (JACIII)* 15(8), 1116-1122 (2011). DOI 10.20965/jaciii.2011.p1116.
- [8] Tamura, K., Yasuda, K.: A Parameter Setting Method for Spiral Optimization from Stability Analysis of Dynamic Equilibrium Point. *SICE Journal of Control, Measurement, and System Integration* 7(3), 173–182 (2014). DOI 10.9746/jcmsi.7.173
- [9] Nasir, A.N.K, Ismail, R.M.T.R., Tokhi, M.O.: A adaptive spiral dynamics metaheuristic algorithm for global optimisation with application to modelling of a flexible system. *Applied Mathematical Modelling* 40(9-10), 5442–5461 (2016). DOI 10.1016/j.apm.2016.01.002
- [10] Li, X., Engelbrecht, A., Epitropakis, M.G.: Benchmarks Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. (Last updated 21 September 2016). Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia.
- [11] Seydel, R.: *Tools for Computational Finance*. Springer-Verlag (2002)
- [12] Joe, S., Kuo, F.Y.: Constructing Sobol sequences with better two dimensional projections. *SIAM J. Sci. Comput.* 30(5), 2635–2654 (2008). DOI 10.1137/070709359