Supplementary Material for "Clustering sequence data with mixture Markov chains with covariates using multiple simplex constrained optimization routine (MSiCOR)"

Priyam Das
Department of Biostatistics, Virginia Commonwealth University, USA
Deborshee Sen
Department of Mathematical Sciences, University of Bath, UK
Debsurya De
Department of Applied Mathematics and Statistics, Johns Hopkins University, USA
Jue Hou
Department of Biostatistics, University of Minnesota, USA
Zahra S. H. Abad
Institute of Health Policy, Management and Evaluation, University of Toronto, Canada
Nicole Kim
Department of Biostatistics, Harvard T. H. Chan School of Public Health, USA
Zongqi Xia
Department of Neurology, University of Pittsburgh, USA
Department of Biomedical Informatics, University of Pittsburgh, USA
Tianxi Cai
Department of Biomedical Informatics, Harvard Medical School, USA
Department of Biostatistics, Harvard T. H. Chan School of Public Health, USA

September 13, 2023

Contents

A Brief Discussion on Literature of Blackbox Optimization

Optimization problems based on simplex parameter space arises in many problems in the field of combinatorial geometry, Hidden Markov Models, Probability vector estimation (e.g., mixture models), B-spline modeling, Genetics (e.g., Blei & Lafferty (2009)) etc. Among the existing methods for constrained optimization, most of them are designed for general linear or non-linear constrained space. But there is scarcity of algorithms which are designed specially for simplex parameter space.

Most of the algorithms for constrained optimization method works fine for convex functions. Among them, 'Interior Point (IP)' (see Potra & Wright (2000), Karmakar (1984), Boyd & Vandenberghe (2006), Wright (2005))) and 'Sequential Quadratic Programming (SQP)' (see Wright (2005), Nocedal & Wright (2006), Boggs & Tolle (1996)) algorithms are widely used. Both of these methods use gradient-based approach. But the problem with gradient based methods is that they might be time consuming for optimizing the functions with complex structure. In case, the closed form expression of the derivative of the function is not available (or provided), taking approximation of the derivative might be erroneous if the function has a lot of spikes in a small neighborhood. Lastly, although derivative-based methods works pretty fast for constrained optimazation problems, these methods are prone to get struck at local solutions for non-convex functions.

For optimizing non-convex functions, many deterministic and stochastic methods have been developed over last few decades. Among them 'Genetic Algorithm (GA)' (see Fraser

(1957), Bethke (1980), Goldberg (1989)) and 'Simulated Annealing (SA)' (see Kirkpatrick et al. (1983), Granville et al. (1994)) are widely used nowadays. Although these methods were first developed for unconstrained global optimization, later they were extended for optimizing objective functions with constraints (see Smith & Romeijn (1994), Reid (1996)). During optimizing the objective function, both of these methods jumps within the parameter domain for better solutions. If our interest is to optimize a function on simplex blocks, using general constrained optimization versions of GA and SA, while looking for better solution, a good proportion of points checked by the above-mentioned algorithms are prone to be generated outside the constrained space or in the infeasible region. Thus the efficiency of the algorithm might be affected. GA does not scale well with complexity because in higher dimensional optimization problems there is often an exponential increase in the search space size (Geris (2012), page 21).

B Theoretical Properties

The greatest challenge of solving a non-convex optimization problem is no algorithm can be designed which always reach the global minimum while optimizing it. However, it is a desirable property of any algorithm that it should reach a global minimum when the function is convex. In this section it has been shown that under some basic regularity conditions, taking the values of the parameters ϕ, *tol_fun_1* and *tol_fun_2* significantly small, the stopping criteria of the proposed algorithm ensures that the solution obtained is a global minimum in case the objective function is convex.

Theorem 1 *Suppose* $\mathbf{S} = \Delta^{n_1-1} \times \cdots \times \Delta^{n_B-1}$ *and* f *is convex, continuous and differentiable on* $\mathbf{S}$. *Suppose* $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_B) \in \mathbf{S}$ *and* $\mathbf{u}_j = (u_{j,1}, \ldots, u_{j,n_j}) \in \Delta^{n_j-1}$ *for* $j = 1, \ldots, B$ *and each of it's co-ordinates are non-zero. Consider a sequence* $\delta_{j,k} = \frac{s_j}{\rho^k}$ *for* $k \in \mathbb{N}$, $s_j > 0$, $\rho > 1$ *for all* $j = 1, \ldots, B$. *Define* $\mathbf{u}_{j,k}^{(i+)} = (u_{j,1} - \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,i-1} - \frac{\delta_{j,k}}{n-1}, u_{j,i} + \delta_{j,k}, u_{j,i+1} - \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,n_j} - \frac{\delta_{j,k}}{n-1})$ *and* $\mathbf{u}_{j,k}^{(i-)} = (u_{j,1} + \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,i-1} + \frac{\delta_{j,k}}{n-1}, u_{j,i} - \delta_{j,k}, u_{j,i+1} + \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,n_j} + \frac{\delta_{j,k}}{n-1})$ *for* $j = 1, \ldots, B$ *and* $i = 1, \ldots, n_j$. *If for all* $k \in \mathbb{N}$, $f(\mathbf{U}) \leq f(\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}, \mathbf{u}_{j,k}^{(i+)}, \mathbf{u}_{j+1}, \ldots, \mathbf{u}_B)$ *and* $f(\mathbf{U}) \leq f(\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}, \mathbf{u}_{j,k}^{(i-)}, \mathbf{u}_{j+1}, \ldots, \mathbf{u}_B)$ *(whenever* $\mathbf{u}_{j,k}^{(i+)}, \mathbf{u}_{j,k}^{(i-)} \in \Delta^{n_k-1}$*) for* $j = 1, \ldots, B$ *and* $i = 1, \ldots, n_j$, $\mathbf{U}$ *is a point of global minimum of* f.

Proof: For all $j = 1, \ldots, B$, define,

$$S_j^* = \{(x_{j,1}, \ldots, x_{j,n_j-1}) \in \mathbb{R}^{n_j-1} | \sum_{i=1}^{n_j-1} x_{j,i} \leq 1, x_{j,i} \geq 0, i = 1, \ldots, n_j - 1\}.$$

Define $f^* : S_1^* \times \cdots \times S_B^* \mapsto \mathbb{R}$ such that $f^*(\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*) = f(\mathbf{x}_1, \ldots, \mathbf{x}_B)$ where $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,n_j}) \in \Delta^{n_j-1}$ and $\mathbf{x}_j^* = (x_{j,1}, \ldots, x_{j,n_j-1}) \in S_j^*$ for $j = 1, \ldots, B$. Note that $\mathbf{x}_j^*$ is the first $(n_j - 1)$ co-ordinates of $\mathbf{x}_j$. Consider the map $I_j : \Delta^{n_j-1} \mapsto S_j^*$ such that $I_j(\mathbf{x}_j) = \mathbf{x}_j^*$. It can be seen that I_j is a bijection for any $j = 1, \ldots, B$.

Define $\mathbf{I} : \Delta^{n_1-1} \times \cdots, \times \Delta^{n_B-1} \mapsto S_1^* \times \cdots \times S_B^*$ such that

$$\mathbf{I}(\mathbf{x}_1, \ldots, \mathbf{x}_B) = \big(I_1(\mathbf{x}_1), \ldots, I_B(\mathbf{x}_B)\big) = (\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*).$$

Since $\mathbf{I}$ is the Cartesian product of bijective functions $I_1, \ldots, I_B$, $\mathbf{I}$ is also a bijection. Hence, to prove that $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_B)$ is the global minimum of f on $\Delta^{n_1-1} \times \cdots \times \Delta^{n_B-1}$, it is enough to show that $(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*)$ is the global minimum of $f(\mathbf{I}^{-1}(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*))$ on $S_1^* \times \cdots \times S_B^*$. The definition of f^* reveals that $f^* = f \circ \mathbf{I}^{-1}$. Hence it will be sufficient to show that $(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*)$ is a global minimum of f^* on $S_1^* \times \cdots \times S_B^*$.

The convexity of f^* follows from the convexity of f. Consider any two points $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_B)$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_B)$ in $\Delta^{n_1-1} \times \cdots, \times \Delta^{n_B-1}$. Define $\mathbf{X}^* = (\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*)$ and $\mathbf{Y}^* = (\mathbf{y}_1^*, \ldots, \mathbf{y}_B^*)$ where $\mathbf{x}_j^*$ and $\mathbf{y}_j^*$ denotes the first $(n_j - 1)$ co-ordinates of $\mathbf{x}_j$ and $\mathbf{y}_j$ respectively for $j = 1, \ldots, B$. Take any constant $\gamma \in (0, 1)$. Now

$$\begin{aligned}
\gamma f^*(\mathbf{X}^*) + (1 - \gamma) f^*(\mathbf{Y}^*) =& \gamma f(\mathbf{I}^{-1}(\mathbf{X}^*)) + (1 - \gamma) f(\mathbf{I}^{-1}(\mathbf{Y}^*)) \\
=& \gamma f(\mathbf{X}) + (1 - \gamma) f(\mathbf{Y}) \\
\geq& f(\gamma \mathbf{X} + (1 - \gamma)\mathbf{Y})
\end{aligned} \tag{1}$$

Now we show

$$\gamma \mathbf{X} + (1 - \gamma)\mathbf{Y} = \mathbf{I}^{-1}(\gamma \mathbf{X}^* + (1 - \gamma)\mathbf{Y}^*). \tag{2}$$

Suppose $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_B) = \gamma \mathbf{X} + (1 - \gamma)\mathbf{Y}$. Hence $\mathbf{z}_j = \gamma \mathbf{x}_j + (1 - \gamma)\mathbf{y}_j$ for $j = 1, \ldots, B$. Now,

$$\begin{aligned}
\mathbf{z}_j &= \gamma \mathbf{x}_j + (1 - \gamma)\mathbf{y}_j \\
&= \big(\gamma x_{j,1} + (1 - \gamma) y_{j,1}, \ldots, \gamma x_{j,n_j} + (1 - \gamma) y_{j,n_j}\big)
\end{aligned}$$

Since $\gamma x_{j,k} + (1-\gamma)y_{j,k} \geq 0$ for $k = 1, \ldots, n_j$ and

$$\sum_{k=1}^{n_j}[\gamma x_{j,k} + (1-\gamma)y_{j,k}] = \gamma \sum_{k=1}^{n_j} x_{j,k} + (1-\gamma)\sum_{k=1}^{n_j} y_{j,k} = \gamma + (1-\gamma) = 1,$$

hence $\mathbf{z}_j \in \Delta^{n_j-1}$ for $j = 1, \ldots, B$. Therefore $\mathbf{Z} \in \Delta^{n_1-1} \times \cdots, \times \Delta^{n_B-1}$. So,

$$\begin{aligned}\mathbf{I}(\mathbf{Z}) &= \big(I_1(\mathbf{z}_1), \ldots, I_B(\mathbf{z}_B)\big)\\ &= (\mathbf{z}_1^*, \ldots, \mathbf{z}_B^*)\end{aligned}$$

where

$$\begin{aligned}\mathbf{z}_k^* &= (z_{j,1}, \ldots, z_{j,n_j-1})\\ &= \big(\gamma x_{j,1} + (1-\gamma)y_{j,1}, \ldots, \gamma x_{j,n_j-1} + (1-\gamma)y_{j,n_j-1}\big)\\ &= \gamma \mathbf{x}_j^* + (1-\gamma)\mathbf{y}_j^*\end{aligned}$$

for $j = 1, \ldots, B$. Therefore,

$$\begin{aligned}\mathbf{I}(\mathbf{Z}) &= (\gamma \mathbf{x}_1^* + (1-\gamma)\mathbf{y}_1^*, \ldots, \gamma \mathbf{x}_B^* + (1-\gamma)\mathbf{y}_B^*)\\ &= \gamma \mathbf{X}^* + (1-\gamma)\mathbf{Y}^*.\end{aligned}$$

Hence, $\mathbf{I}$ being bijection,

$$\mathbf{Z} = \mathbf{I}^{-1}(\gamma \mathbf{X}^* + (1-\gamma)\mathbf{Y}^*).$$

Hence (2) holds true. From (1) and (2) we get

$$\begin{aligned}\gamma f^*(\mathbf{X}^*) + (1-\gamma) f^*(\mathbf{Y}^*) \geq & f(\mathbf{I}^{-1}(\gamma \mathbf{X}^* + (1-\gamma)\mathbf{Y}^*))\\ =& f^*(\gamma \mathbf{X}^* + (1-\gamma)\mathbf{Y}^*).\end{aligned}$$

Hence f^* is convex.

Fix any $j \in \{1, \ldots, B\}$. Define $f_j : \Delta^{n_j-1} \mapsto \mathbb{R}$ such that

$$f_j(\mathbf{v}_j) = f(\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}, \mathbf{v}_j, \mathbf{u}_{j+1}, \ldots, \mathbf{u}_B)$$

where $\mathbf{v}_j = (v_{j,1}, \ldots, v_{j,n_j}) \in \Delta^{n_j-1}$ for $j = 1, \ldots, B$. Note that $(x_{j,1}, \ldots, x_{j,n_j}) \in \Delta^{n_j-1}$

implies $(x_{j,1}, \ldots, x_{j,n_j-1}) \in S_j^*$. Define $\mathbf{u}_j^* = (u_{j,1}, \ldots, u_{j,n_j-1})$ and

$$\mathbf{u}_{j,k}^{*(i+)} = (u_{j,1} - \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,i-1} - \frac{\delta_{j,k}}{n-1}, u_{j,i} + \delta_{j,k}, u_{j,i+1} - \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,n_j-1} - \frac{\delta_{j,k}}{n-1}),$$
$$\mathbf{u}_{j,k}^{*(i-)} = (u_{j,1} + \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,i-1} + \frac{\delta_{j,k}}{n-1}, u_{j,i} - \delta_{j,k}, u_{j,i+1} + \frac{\delta_{j,k}}{n-1}, \ldots, u_{j,n_j-1} + \frac{\delta_{j,k}}{n-1}).$$

Note that $\mathbf{u}_j^*, \mathbf{u}_{j,k}^{*(i+)}$ and $\mathbf{u}_{j,k}^{*(i-)}$ are the first $n_j - 1$ co-ordinates of $\mathbf{u}_j, \mathbf{u}_{j,k}^{(i+)}$ and $\mathbf{u}_{j,k}^{(i-)}$ respectively.

Following the argument in the proof of Theorem 1 in Das (2021), under the given conditions, there exists a positive integer N_j such that for all $k \geq N_j$, $\mathbf{u}_{j,k}^{(i+)}, \mathbf{u}_{j,k}^{(i-)} \in \Delta^{n_j-1}$. Hence for $k \geq N_j$, $\mathbf{u}_{j,k}^{*(i+)}, \mathbf{u}_{j,k}^{*(i-)} \in S_j^*$. Define $f_j^* : S_j^* \mapsto \mathbb{R}$ such that

$$f_j^*(x_{j,1}, \ldots, x_{j,n_j-1}) = f_j(x_{j,1}, \ldots, x_{j,n_j-1}, 1 - \sum_{i=1}^{n_j-1} x_{j,i}).$$

Hence we have $f_j^*(\mathbf{u}_j^*) = f_j(\mathbf{u}_j), f_j^*(\mathbf{u}_{j,k}^{*(i+)}) = f_j(\mathbf{u}_{j,k}^{(i+)})$ and $f_j^*(\mathbf{u}_{j,k}^{*(i-)}) = f_j(\mathbf{u}_{j,k}^{(i-)})$ for $i = 1, \ldots, n_j - 1$. Define $\nabla_{j,i} = \frac{\partial}{\partial x_{j,i}} f_j^*(\mathbf{u}_j^*)$ for $i = 1, \ldots, n_j - 1$. Again following the arguments made in the proof of Theorem 1 in Das (2021), under the given conditions, $\nabla_{j,i} = 0$ for $i = 1, \ldots, n_j - 1$. Now,

$$f_j^*(\mathbf{u}_j^*) = f_j(\mathbf{u}_j) = f(\mathbf{u}_1, \ldots, \mathbf{u}_B) = f(\mathbf{I}^{-1}(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*)) = f^*(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*).$$

Hence for each $j = 1, \ldots, B$,

$$\frac{\partial}{\partial x_{j,i}} f^*(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*) = \frac{\partial}{\partial x_{j,i}} f_j^*(\mathbf{u}_j^*) = \nabla_{j,i} = 0$$

for $i = 1, \ldots, n_j - 1$. That implies the partial derivative of f^* with respect to each co-ordinate is zero at $(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*)$. Since f^* has been shown to be convex, $(\mathbf{u}_1^*, \ldots, \mathbf{u}_B^*)$ is a global minimum of f^*. Hence, $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_B)$ is a global minimum of f. □

C Performance Evaluation on Benchmark Functions

For simulation purpose, some multidimensional benchmark functions are considered on transformed unit-simplex blocks parameter space. The values of the tuning parameters are considered as follows: $s_{initial} = 1$, $\rho = 1.01, \phi = 10^{-4}$ and $\lambda = 10^{-6}$. The values of *max_iter* and *max_runs* are taken to be 5000 and 200 respectively. Suppose a function f is to be minimized on a d-dimensional hypercube D^d where $D = [l, u]$ for some constants l, u in $\mathbb{R}$. Consider the map $g : D \mapsto [0, \frac{1}{d}]$ such that $g(x_i) = y_i = \frac{x_i - l}{d(u-l)}$ for $i = 1, \ldots, d$. Clearly g is a bijection. After replacing the original parameters of the problem with the transformed parameters we get

$$f(x_1, \ldots, x_d) = f(g^{-1}(y_1), \ldots, g^{-1}(y_d)),$$

where $g^{-1}(y_i) = (u-l)dy_i + l$ for $i = 1, \ldots, d$. Define $h : [0, \frac{1}{d}]^d \mapsto \mathbb{R}$ such that

$$h(\mathbf{y}) = h(y_1, \ldots, y_d) = f(g^{-1}(y_1), \ldots, g^{-1}(y_d)).$$

Consider the set $S = \{(z_1, \ldots, z_d) \mid z_i \geq 0, \ \sum_{i=1}^{d} z_i \leq 1\}$. Note that $[0, \frac{1}{d}]^d \subset S$. Define $h' : S \mapsto \mathbb{R}$ which is equal to function h considered on the extended domain S. Since $y_i \in [0, \frac{1}{d}]$ for $i = 1, \ldots, d$ and $0 \leq \sum_{i=1}^{d} y_i \leq 1$, hence $0 \leq 1 - \sum_{i=1}^{d} y_i \leq 1$. Define $y_{d+1} = 1 - \sum_{i=1}^{d} y_i$. Hence we can conclude that $\bar{\mathbf{y}} = [\mathbf{y}, \ y_{d+1}] \in \Delta^d$ where $\mathbf{y} = (y_1, \ldots, y_d)$ and

$$\Delta^d = \{(y_1, \ldots, y_{d+1}) \in \mathbb{R}^{d+1} \mid y_i \geq 0, \ i = 1, \ldots, d+1, \ \sum_{i=1}^{d+1} y_i = 1\}.$$

Now define $\bar{h} : \Delta^d \mapsto \mathbb{R}$ such that $\bar{h}(\bar{\mathbf{y}}) = \bar{h}(y_1, \ldots, y_{d+1}) = h'(y_1, \ldots, y_d)$ for $\bar{\mathbf{y}} \in \Delta^d$. It can be seen that $\bar{\mathbf{y}} \in \Delta^d$ implies $(y_1, \ldots, y_d) \in S$. Suppose the global minimum of the function f occurs at $(m_1, \ldots, m_d)$ in D^d. Hence, the function $\bar{h}$ will have the global minimum at $\bar{\mathbf{y}} = \left(g^{-1}(m_1), \ldots, g^{-1}(m_d), 1 - \sum_{i=1}^{d} g^{-1}(m_i)\right)$ in Δ^d.
Now consider there are n blocks of simplexes $\bar{\mathbf{y}}_r \in \Delta^d$ for $r = 1, \ldots, n$. Define,

$$H(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n) = \bar{h}(\bar{\mathbf{y}}_1) + \cdots + \bar{h}(\bar{\mathbf{y}}_n)$$

The comparative study of performances of MSiCOR along with a few other existing methods is evaluated in Table 1 and 2 of the main paper.

C.1 Modified Rastrigin Function

d-dimensional Rastrigin function is given by

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - \cos(2\pi x_i)], \ \mathbf{x} = (x_1, \ldots, x_d) \in D \tag{3}$$

where D, the domain of the parameters are typically taken to be $[-5.12, 5.12]^d$. Hence we have $l = -5.12, u = 5.12$. After performing the above-mentioned transformations, we obtain

$$\bar{h}(\bar{\mathbf{y}}) = 10d + \sum_{i=1}^{d} [(10.24dy_i - 5.12)^2 - \cos(2\pi(10.24dy_i - 5.12))], \ \bar{\mathbf{y}} = (y_1, \ldots, y_{d+1}) \in \Delta^d. \tag{4}$$

We consider the case where we need to minimize $H(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n) = \sum_{v=1}^{n} \bar{h}(\bar{\mathbf{y}}_v)$ for $\bar{\mathbf{y}}_v \in \Delta^d$ for $v = 1, \ldots, n$. In Table 1 of the main paper, is noted that MSiCOR outperformed other algorithms and the average computation time of MSiCOR is 4-5 folds smaller than that of GA.

C.2 Modified Ackley's Function

d-dimensional Ackley's function is given by

$$f(\mathbf{x}) = -20 \exp(-0.2\sqrt{0.5 \sum_{i=1}^{d} x_i^2} - \exp(0.5(\sum_{i=1}^{d} \cos(2\pi x_i))) + e + 20, \ \mathbf{x} = (x_1, \ldots, x_d) \in D \tag{5}$$

where D, the domain of the parameters are typically taken to be $[-5, 5]^d$. After necessary transformations, we get

$$\bar{h}(\bar{\mathbf{y}}) = -20 \exp(-0.2\sqrt{0.5 \sum_{i=1}^{d} (10dy_i - 5)^2} - \exp(0.5(\sum_{i=1}^{d} \cos(2\pi(10dy_i - 5)))) + e + 20, \tag{6}$$

where $\bar{\mathbf{y}} = (y_1, \ldots, y_{d+1}) \in \Delta^d$. Our objective is to minimize $H(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n) = \sum_{v=1}^{n} \bar{h}(\bar{\mathbf{y}}_v)$ over $\bar{\mathbf{y}}_v \in \Delta^d$ for $v = 1, \ldots, n$. In this case also, MSiCOR outperforms all other algorithms with 2-3 fold time improvement over GA (see Table 1 of the main paper).

C.3 Modified Sphere Function

All of the above-mentioned functions being non-convex, we consider the Sphere function which is convex. d-dimensional Sphere function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2,\ \mathbf{x} = (x_1, \ldots, x_d) \in D \tag{7}$$

Here the domain of the parameters D is taken to be $[-5.12, 5.12]^d$. The modified Sphere function on simplex is given by,

$$\bar{h}(\bar{\mathbf{y}}) = \sum_{i=1}^{d} (10.24 d y_i - 5.12)^2,\ \bar{\mathbf{y}} = (y_1, \ldots, y_{d+1}) \in \Delta^d. \tag{8}$$

We minimize $H(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n) = \sum_{v=1}^{n} \bar{h}(\bar{\mathbf{y}}_v)$ over $\bar{\mathbf{y}}_v \in \Delta^d$ for $v = 1, \ldots, n$. It should be noted that, the Sphere function being convex, IP and SQP functions are expected to work better than MSiCOR and GA while minimizing it. Note that, MSiCOR yields significantly better solution than GA with 3-4 folds improvement in computation time (see Table 1 of the main paper).

C.4 Modified Griewank Function

d-dimensional Griewank function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}),\ \mathbf{x} = (x_1, \ldots, x_d) \in D \tag{9}$$

Here the domain of the parameters D is taken to be $[-500, 500]^d$. After transformation, the Griewank function on simplex is given by,

$$\bar{h}(\bar{\mathbf{y}}) = \sum_{i=1}^{d} \frac{(1000 d y_i - 500)^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{1000 d y_i - 500}{\sqrt{i}}\right),\ \bar{\mathbf{y}} = (y_1, \ldots, y_{d+1}) \in \Delta^d. \tag{10}$$

We minimize $H(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n) = \sum_{v=1}^{n} \bar{h}(\bar{\mathbf{y}}_v)$ over $\bar{\mathbf{y}}_v \in \Delta^d$ for $v = 1, \ldots, n$. For the case $n = 5, d = 5$, MSiCOR performs significantly better than others (see Table 1 of the main paper). In the other case, IP and SQP performs better than GA and MSiCOR. In this case also, MSiCOR outperforms GA with 4-7 folds improvement in computation time.

D Recursive Modified Pattern Search (RMPS) for unconstrained parameter space

Following the RMPS algorithm for global optimization of parameter belonging to hyper-rectangular space (Das 2023), here we modify it for unconstrained parameter space, provided in Algorithm 1. Suppose $\mathbf{l} \in \mathrm{R}^n$ denotes the unconstrained parameter and we need to minimize $g : \mathrm{R}^n \mapsto \mathrm{R}$. Here $\hat{\mathbf{l}}^{(R)}$ denotes value of $\mathbf{l}$ at the end of R-th *run* and $\mathbf{l}^{(h)}$ denotes value of $\mathbf{l}$ at the end of h-th iteration within a *run*.

Algorithm 1 RMPS for unconstrained parameter space

$R \leftarrow 1$
top:
$h \leftarrow 1$
$gs^{(0)}, gs^{(1)} \leftarrow s_{initial}$
if $R = 1$ **then**
    $\mathbf{l}^{(0)} \leftarrow$ Initial guess (unconstrained parameter)
else
    $\mathbf{l}^{(0)} \leftarrow \hat{\mathbf{l}}^{(R-1)}$
while $\left(h \leq max\_iter\right.$ and $\left.gs^{(h)} > \phi\right)$ **do**
    $F \leftarrow f(\mathbf{l}^{(h-1)})$
    $gs \leftarrow gs^{(h-1)}$
    for $k = 1 : 2n$ **do**
        $i \leftarrow [\frac{(k+1)}{2}]$ ($[\cdot]$ denotes largest smaller integer function)
        $\mathbf{l}_k \leftarrow \mathbf{l}^{(h-1)}$
        $\mathbf{l}^{temp} \leftarrow \mathbf{l}^{(h-1)}$
        $s_k \leftarrow (-1)^k gs^{(h)}$
        $\mathbf{l}_k(i) \leftarrow \mathbf{l}^{temp}(i) + s_k$
        $g_k \leftarrow g(\mathbf{l}_k)$
    $k_{best} \leftarrow \arg\min_k g_k$, over $k = 1, \ldots, 2n$
    $FF \leftarrow g_{k_{best}}$
    if $\left(FF < F\right)$ **then**
        $\mathbf{l}^{(h)} \leftarrow \mathbf{l}_{k_{best}}$
    if $\left(h > 1\right)$ **then**
        if $\left(|F - min(F, FF)| < tol\_fun\right.$ and $\left.gs > \phi\right)$ **then**
            $gs \leftarrow \frac{gs}{\rho}$
    $gs^{(h)} \leftarrow gs$
    $h \leftarrow h + 1$
$\hat{\mathbf{l}}^{(R)} \leftarrow \mathbf{l}^{(h)}$,
if $|f(\hat{\mathbf{l}}^{(R)}) - f(\hat{\mathbf{l}}^{(R-1)})| < tol\_fun\_2$ **then**
    return $\hat{\mathbf{l}} = \hat{\mathbf{l}}^{(R)}$ as **final solution**
    exit
else
    $R \leftarrow R + 1$
    goto *top*.

References

Bethke, A. (1980), 'Genetic algorithms as function optimizers'.

Blei, D. M. & Lafferty, J. D. (2009), 'Topic models'. http://www.cs.columbia.edu/ blei/papers/BleiLafferty2009.pdf.

Boggs, P. T. & Tolle, J. W. (1996), 'Sequential quadratic programming', *Acta Numerica* pp. 1–52.

Boyd, S. & Vandenberghe, L. (2006), *Convex Optimization*, Cambridge University Press, Cambridge.

Das, P. (2021), 'Recursive modified pattern search on high-dimensional simplex : A black-box optimization technique', *The Indian Journal of Statistics - Sankhya B* **83**, 440–483.

Das, P. (2023), 'Black-box optimization on hyper-rectangle using recursive modified pattern search and application to roc-based classification problem', *The Indian Journal of Statistics - Sankhya B (to appear)* `https://arxiv.org/abs/1604.08616v3`.

Fraser, A. S. (1957), 'Simulation of genetic systems by automatic digital computers i. introduction', *Australian Journal of Biological Sciences* **10**, 484–491.

Geris, L. (2012), *Computational Modeling in Tissue Engineering*, Springer.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Operations Research Series, Addison-Wesley Publishing Company.

Granville, V., Krivanek, M. & Rasson, J. P. (1994), 'Simulated annealing: A proof of convergence', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 652–656.

Karmakar, N. (1984), 'New polynomial-time algorithm for linear programming', *Combinatorica* **4**, 373–395.

Kirkpatrick, S., Gelatt, C. & Vecchi, M. (1983), 'Optimization by simulated annealing', *Australian Journal of Biological Sciences* **220**(4598), 671–680.

Nocedal, J. & Wright, S. J. (2006), *Numerical Optimization*, Operations Research Series, 2nd edn, Springer.

Potra, F. A. & Wright, S. J. (2000), 'Interior-point methods', *Journal of Computational and Applied Mathematics* **4**, 281–302.

Reid, D. (1996), ‘Genetic algorithms in constrained optimization’, *Mathematical and Computer Modelling* **23(5)**, 87–111.

Smith, R. L. & Romeijn, H. E. (1994), ‘Simulated annealing for constrained global optimization.” journal of global optimization’, *Journal of Global Optimization* **5(2)**, 101–126.

Wright, M. H. (2005), ‘The interior-point revolution in optimization: History, recent developments, and lasting consequences’, *Bulletin of American Mathematical Society* **42**, 39–56.