Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 6(3): 2956–2969.
DOI: 10.3934/math.2021179
Received: 25 October 2020
Accepted: 17 December 2020
Published: 08 January 2021

Research article

Multi-stage differential evolution algorithm for constrained D-optimal design

Xinfeng Zhang[1], Zhibin Zhu[2] and Chongqi Zhang[1,*]

[1] School of Economics and Statistics, Guangzhou University, Guangzhou 510006, China
[2] Undergraduate School of Medical Business, Guangdong Pharmaceutical University, Guangzhou 510006, China

* **Correspondence:** Email: cqzhang@gzhu.edu.cn.

Abstract: In practice, objective condition may impose constraints on design region, which make it difficult to find the exact D-optimal design. In this paper, we propose a Multi-stage Differential Evolution (MDE) algorithm to find the global approximated D-optimal design in an experimental region with linear or nonlinear constraints. MDE algorithm is approved from Differential Evolution (DE) algorithm. It has low requirements for both feasible regions and initial values. In iteration, MDE algorithm pursues evolutionary equilibrium rather than convergence speed, so it can stably converge to the global D-optimal design instead of the local ones. The advantages of MDE algorithm in finding D-optimal design will be illustrated by examples.

Keywords: experimental design; D-optimal design; constrained experimental region; multi-stage differential evolution algorithm
Mathematics Subject Classification: 62K05, 62K99

1. Introduction

Optimal design is a common tool in solving industrial problems. In order to determine the production process or evaluate the product quality, experimenters usually take into account the optimal design methods to collect more information. The optimal design is to find a best design with given model and certain optimal criteria in a feasible region. When we deal with practical optimal design problems, the feasible region is usually irregular due to the limitation of some objective conditions, which leads to the constrained experimental regions and the limitations are called constraints. Montgomery et al. [1], Borkowski and Piepel [2] discussed a large number of constrained industrial optimal design examples. It is difficult to find the exact design if the design region is irregular due to constraints.

It is difficult to find the exact design if the design region is constrained. Therefore, it turns out to be a new trend to find the approximate solutions of constrained optimal design problems by computer algorithms.

Scholars have put forward many approaches to solve the constrained optimal design problems. Snee and Marquardt [3] proposed XVERT algorithm to solve constrained mixture design problems. Mitchell [4] applied DETMAX algorithm to construct D-optimal design. Cook and Nachtsheim [5] presented Modified Fedorov Exchange (MFE) algorithm for D-optimal design. According to the iteration processes of the earlier algorithms, the optimality of a design mainly depends on the candidate sets or parameters.

Biomimetic algorithms helped us to find more accurate numerical solutions in the past two decades. For example, Heredia-Langner et al. [6,7] used Genetic Algorithm (GA) to calculate D-optimum, Q-optimum constrained experimental design problems. Yu [8] proposed Cocktail Algorithm to find the D-optimal design of some experimental problems. Liu and Liu [9] constructed constrained uniform mixture design by a special method. Li et al. [10] applied Mixture Design Random Search (MDRS) algorithm to mixture design, they solved D- and A-optimal designs of Scheffé model and Becker model in design region with and without constraints. Duan et al. [11] employed Efficient Computational algorithm for solving optimal continuous experimental designs. In recent years, Particle Swarm Optimization (PSO) algorithm is very common in design problems. Wong et al. and Yang et al. [12,13] applied PSO and Fractional-Order PSO algorithm to obtain optimal designs under D-, A-, I- criteria for linear and Quadratic models.

After an overview of the above biomimetic algorithms, we find that the cocktail algorithm, PSO algorithm and Efficient Computational algorithm have not been applied to the design problem with a constrained region, while the MDRS algorithm and Liu's method have only be used for the mixture constrained design problems. GA can be used for all above situations, but the existing results are not accurate. Thus a stable algorithm with wide application and high precision is imperative.

Differential Evolution (DE) algorithm is well known for its stability and efficiency among the existing biomimetic algorithms. It was proposed by Price [14] first, then Storn et al. and Price et al. [15,16] gave more details and examples of DE algorithm to further shown its advantages. DE algorithm is widely used in many fields, such as Zhao et al. [17] forecasted the per capita annual net income of rural households in China by DE algorithm and Grey model, Rajesha et al. [18] planned least cost generation expansion with DE algorithm, Xu et al. [19] used DE algorithm to find high-dimensional D-optimal designs for logistic models, Deng et al. [20] proposed the EMMSIQDE, a new differential mutation strategy of a difference vector to enhance the search ability and descent ability, etc. DE algorithm gives a steady performance in solving complex problems and above all it has low requirements on feasible region and initial values. DE algorithm is suitable for both optimal design and mixture design problems with irregular feasible region.

One of the difficulties in solving the constrained experimental design problems is that we don't know the exact number of support points before calculation. The general method is setting more initial points than the number of exact support points, then combine the points with short range to a support point. The computation quantity is enormous if we use DE algorithm directly to this problem. Therefore we propose a Multi-stage Differential Evolution (MDE) algorithm for the constrained optimal design problems. MDE algorithm is composed of two stages: calculations of D-optimal support points and the corresponding measures, both stages base on DE algorithm. MDE algorithm can run on any convex feasible region, and sufficient random initial values selected in the feasible region can effectively ensure the convergence. It maintains the stability and effectiveness of DE algorithm while reduces the computational cost.

The remainder of the paper is organized as follows. In section 2, we introduce some basic concepts of optimal design and D-optimal criterion, then we give the process of finding the D-optimal design by MDE algorithm. Section 3 discussed two examples, i.e. an optimal design problem with linear constraints and a mixture design problem with nonlinear constraint. The conclusions and future research directions are given in the last section.

2. MDE algorithms for D-optimal design

2.1. Optimal design and D-optimal criterion

The optimal design method [21] is widely used and D-optimal is the most popular design criterion. When the model is given, the unknown parameters can be calculated efficiently by the data from optimal design. Before finding an optimal design, we need to select a response model first. Scholars usually choose the general linear models as the response model for convenience. Myers et al. [22] discussed the general linear models in details. Dasgupta [23] studied the compromise designs in heteroscedastic linear model. Chaloner [24] contributed to optimal Bayesian experimental design for linear models. The general linear model is

$$Y=\eta(\mathrm{x})+\varepsilon(\mathrm{x})=f'(\mathrm{x})\beta+\varepsilon(\mathrm{x}), \tag{1}$$

where $\eta(\mathrm{x})$ is the expectation of *Y*, which is the outcome of an experiment with a given point x. β is a $p\times1$vector of unknown parameters. We denote the coordinates of the experiment point $\mathrm{x}\in\Omega$ as a $1\times q$ vector $\mathrm{x}=(x_1,\cdots,x_q)$, $f(\mathrm{x})=\left(f_1(\mathrm{x}),f_2(\mathrm{x}),\cdots,f_p(\mathrm{x})\right)^T$ is a given function vector, here Ω is a set of all possible design points. Generally, $\varepsilon(\mathrm{x})$is a random error with zero mean and constant variance.

In this paper, we focus on the approximate optimal design and assume that Ω is a compact set, therefore the approximate optimal design always exists. That is, there is a probability measure in the design space. Generally, a design ξ can be expressed as a combination of k design points $\mathrm{x}_1,\cdots,\mathrm{x}_k$ and their corresponding measures $r_1,\cdots,r_k$,

$$\xi=\begin{pmatrix}\mathrm{x}_1 & \mathrm{x}_2 & \dots & \mathrm{x}_k\\ r_1 & r_2 & \dots & r_k\end{pmatrix}, \tag{2}$$

where $r_1+\cdots+r_k=1$. Given a sample of n known trials, we can take nr_i observation value at each $\mathrm{x}_i, i=1,\cdots,k$ to implement the design so that $nr_1+\cdots+nr_k=n$, and $nr_i, i=1,2,\cdots,k$ is an integer. This means that an approximate optimal design is a probability measure on Ω. For a given experimental design problem, the set of all possible designs is denoted by Ξ.

The information matrix of approximate design ξ is

$$M(\xi)=E_{\xi}\left(f(\mathrm{x})f'(\mathrm{x})\right)=\int_{\Omega}f(\mathrm{x})f'(\mathrm{x})\xi d(\mathrm{x}). \tag{3}$$

A design is called an optimal design with constraints if the experimental point $\mathrm{x}_i, i=1,\cdots,k$ is limited in the constrained design region χ, see Cook et al [25]. Mixture design focuses on the proportions of mixture components, i.e. all the experimental design points are limited to a simplex S^{q-1},

$$S^{q-1}=\left\{\mathrm{x}=\left(x_1,\cdots,x_q\right) \mid \sum_{j=1}^{q}x_j=1, x_j\geq 0, j=1,\cdots,q\right\}.$$

Further details of mixture design can be found in [26].

D-optimality is one of the most widely used criteria in experimental design problems. We can get D-optimum by choosing the appropriate design ξ to maximize the determinant value of the Fisher information matrix $M(\xi)$, or minimum $\det\left(M^{-1}(\xi)\right)$.

In the problem of finding D-optimal design, the variance function is

$$d(\mathrm{x},\xi)=f'(\mathrm{x})M^{-1}(\xi)f(\mathrm{x}), \mathrm{x}\in\Omega. \tag{4}$$

ξ_D is a D-optimal design of the linear model (1) if and only if it satisfies the following general equivalence theorems, see [27] .

(1) $\Psi(\xi_D)=\det\left(M(\xi)\right)\to\sup_{\xi\in\Xi}$; (5)

(2) $\max_{\mathrm{x}\in\Omega}d(\mathrm{x},\xi_D)=p$.

D-efficiency is usually used to compare two k runs D-optimal designs, the D-efficiency in constrained optimal design problem can be expressed as follows

$$D_{eff}=\left(\frac{det\left(M_1^{-1}\right)}{det\left(M_2^{-1}\right)}\right)^{\frac{1}{k}},$$

where $M_i = \int_{\chi} f\left(\mathrm{x}_i\right) f'\left(\mathrm{x}_i\right) d\left(\mathrm{x}\right), i = 1, 2$ come from different D-optimal designs of same constrained optimal design problem.

2.2. MDE algorithms for constrained optimal design problems

When we try to find an optimal design with MDE algorithm, we are solving an optimization problem

$$\max_{\xi \in \Xi} \Psi(\xi), \tag{6}$$

where objective function is determined by the optimal criterion.

In this paper, we consider that a D-optimal design consists of two parts: D-optimal experimental support points and the correlation measures, then we calculate them respectively by the following two stages.

2.2.1. Stage 1- finding D-optimal design points

At this stage, we assume that all experimental design points have the same measure, then we only focus on finding the experimental design points.

2.2.1.1. Encoding

The calculation process of MDE algorithm begins with the coding of design points. For model (1), in order to convert the design points in design ξ into the form in which we can solving by MDE algorithm, we code it as a row vector *X*,

$$X = \left(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_s\right), \tag{7}$$

where *X* is called design points chromosome vector, in case of optimal design problems with constraints, $s \geq k$, $\mathrm{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iq}) \in \chi$, $i = 1, \cdots, s$.

It is obvious that the design points in a design ξ correspond to a design point chromosome vector *X* one by one. Each element in the design point chromosome vector is called a gene, a design points chromosome *X* contains a total of $s \times q$ genes.

2.2.1.2. Selection of initial value

To ensure the implementation of MDE algorithm, we need to establish a design points population of N_1 design points chromosome vectors $A_{N_1}^{(t)}$, where

$$A_{N_1}^{(t)} = \begin{pmatrix} X_1^{(t)} \\ \vdots \\ X_{N_1}^{(t)} \end{pmatrix}_{N_1 \times (sq)} ,$$

N_1 is called design points population size, t is the generation of population, $t \in Z,\ t \geq 0$, $A_{N_1}^{(t)}$ is the parent population of $A_{N_1}^{(t+1)}$, and $A_{N_1}^{(t+1)}$ is child population $A_{N_1}^{(t)}$, $X_i^{(t)}$ is the t^{th} generation of i^{th} design points chromosome vector, the design points population size $N_1 \geq 3sq$ by experience. We call $A_{N_1}^{(0)}$ the initial design points population, which is given randomly on the feasible region rather than chosen especially.

2.2.1.3. Mutation

For t^{th} generation population, we choose three different $X_k^{(t)}, X_l^{(t)}, X_m^{(t)},\ k, l, m \in [1, N_1]$, $k \neq l \neq m \neq i$ for each $X_i^{(t)}, i \in [1, N_1]$. Mutation operation is performed to generate mutated chromosomes V_i^1 for each $X_i^{(t)}, i = 1, \cdots, N_1$ with formula

$$V_i^1 = X_m^{(t)} + F \times \left(X_k^{(t)} - X_l^{(t)} \right),$$

where scale factor $F \in [0, 2]$. The mutation chromosome V_i^1 is retained for the next cross transformation.

2.2.1.4. Crossover

In this step, we obtain a crossover chromosome U_i^1 by compare each gene of $X_i^{(t)} (1 \leq i \leq N_1)$ with its mutated chromosome V_i^1. For a given cross factor $CR \in [0,1]$, the crossover approach is to take a random number $b \in [0,1]$ for the j^{th} gene in $X_i^{(t)}, j = 1, \cdots, sq$, let

$$U_i^1(j) = \begin{cases} X_i^{(t)}(j) & \text{if } b \leq CR, \\ V_i^1(j) & \text{otherwise.} \end{cases}$$

After processing all the genes in $X_i^{(t)} (1 \leq i \leq N_1)$, crossover chromosomes $U_i^1, i = 1, \cdots, N_1$ will be generated. If there is no special value tendency, CR will be taken as 0.5.

2.2.1.5. Selection and decoding

We decode the crossover chromosomes $U_i^1, i = 1, \cdots, N_1$ and $X_i^{(t)}, i = 1, \cdots, N_1$ into uniform measure designs with $\xi_{U_i^1}$ and $\xi_{X_i^{(t)}}$ respectively, the child chromosome vector $X_i^{(t+1)}, i = 1, \cdots, N_1$ is decided by formula

$$X_i^{(t+1)} = \begin{cases} U_i^1 & \Psi(\xi_{U_i^1}) > \Psi(\xi_{X_i^{(t)}}), \\ X_i^{(t)} & \text{otherwise.} \end{cases}$$

The selection process ensures that the chromosome with larger objective function value will be chosen into the child population, i.e. the iteration with MDE algorithm holds a correct evolution direction and converges fast.

2.2.1.6. Conditions of termination

The convergence of MDE algorithm ensures there must exist T_1, so that after the iteration times $t\left(t \geq T_1\right)$, $X_i^{(t)} \rightarrow X$, $i = 1, \cdots, N_1$. Therefore, X is the design points of D-optimal design under the uniform measure assumption.

The child population is obtained after mutation, crossover and selection of the parent population. For each child population, it needs to be evaluated whether the termination condition has been satisfied. If the termination condition is satisfied, the calculation ends and the optimal solution is obtained. Otherwise, go back to the mutation process and the iteration continues.

After merging nearby points in X , we obtain $X_D = \left(\mathrm{x}_{D1}, \mathrm{x}_{D2}, \cdots, \mathrm{x}_{Dk}\right), 1 \leq k \leq s$, here $\mathrm{x}_{D1}, \mathrm{x}_{D2}, \cdots, \mathrm{x}_{Dk}$ are D-optimal support points, k is the number of D-optimal support points.

2.2.2. Stage 2-solving the corresponding measure

In this section, we calculate the corresponding measure $r_1, r_2, \cdots, r_k$ to get the D-optimal design.

2.2.2.1. Encoding

Let

$$R = \left(r_1, r_2, \cdots, r_k\right),$$

where R is called measure chromosome vector, r_i is the corresponding measure of x_{Di}, $i=1,\cdots,k$, $\sum_{i=1}^{k} r_i = 1$. Each element in the measure chromosome vector is called a gene, a measure chromosome R contains a total of k genes.

2.2.2.2. Selection of initial value

Establish a measure population $B_{N_2}^{(t)}$, where

$$B_{N_2}^{(t)} = \begin{pmatrix} R_1^{(t)} \\ \vdots \\ R_{N_2}^{(t)} \end{pmatrix}_{N_2 \times k},$$

N_2 is called measure population size, t is the generation of population, $t \in Z,\ t \ge 0$, $B_{N_2}^{(t)}$ is the parent population, and $B_{N_2}^{(t+1)}$ is child population, $R_i^{(t)}$ is the t^{th} generation of i^{th} measure chromosome vector, here $N_2 \ge 3k$ by experience. $B_{N_2}^{(0)}$ is the initial measure population, and $R_i^{(0)},\ i=1,\cdots,N_2$ come from S^{k-1} randomly.

2.2.2.3. Mutation

For generation t , we choose three different $R_k^{(t)}, R_l^{(t)}, R_m^{(t)}, k, l, m \in [1, N_2], k \ne l \ne m \ne i$ for each $R_i^{(t)}, i \in [1, N_2]$. We generate mutated chromosomes $V_i^2, i=1,\cdots,N_2$ for each $R_i^{(t)}, i=1,\cdots,N_2$ with formula

$$V_i^2 = R_m^{(t)} + F \times \left(R_k^{(t)} - R_l^{(t)} \right).$$

2.2.2.4. Crossover

We obtain a crossover chromosome U_i^2 by following formula

$$U_i^2(j) = \begin{cases} R_i^{(t)}(j) & \text{if } s \le CR, \\ V_i^2(j) & \text{otherwise,} \end{cases}$$

where $i = 1, \cdots, N_2, \ j = 1, \cdots, k$.

2.2.2.5. Selection and decoding

$$R_i^{(t+1)} = \begin{cases} U_i^2 & \Psi\left(\xi_{U_i^2}\right) > \Psi\left(\xi_{R_i^{(t)}}\right), \\ R_i^{(t)} & \text{otherwise.} \end{cases}$$

For design ξ_R, the design points are D-optimal support points X_D and the corresponding measures are measure chromosome vector *R*.

2.2.2.6. Conditions of termination

There must be T_2, when $t \geq T_2$,

$$\Psi\left(\xi_R\right) \to \sup_{\xi \in \Xi},$$

where $R_D = \left(r_{D1}, r_{D2}, \cdots, r_{Dk}\right)$ is the limit of $R_i^{(t)}$, T_2 is the generation of measure population. If last formula is satisfied, we obtain the D-optimal design

$$\xi_D = \begin{pmatrix} \mathrm{x}_{D1} & \mathrm{x}_{D2} & \cdots & \mathrm{x}_{Dk} \\ r_{D1} & r_{D2} & \cdots & r_{Dk} \end{pmatrix},$$

of constrained optimal design problem, otherwise go back to 2.2.2.3.

3. Results

We programmed the following examples in Matlab with $F = 0.9, \ CR = 0.5$.

3.1. Experiment of adhesive bonding strength

An industrial experiment of adhesive bonding strength is discussed in detail by Montgomery et al. [1] and Heredia-Langner et al [7]. The experimenters used an adhesive to bond two parts together and then cured the adhesive at an elevated temperature. It is known that the adhesive strength is related to the amount of adhesive and bonding temperature. If the amount of adhesive is too little and the curing temperature is too low, the bonding strength will be poor. If the amount of adhesive is too much and the curing temperature is too high, the parts may be damaged. The amount of adhesive and the curing temperature should be constrained by objective conditions. We should give an approximate D-optimal design to arrange the experiment more effectively. Here, let x_1 is the amount of adhesive, x_2 is the bonding temperature. After standardization, the experimental area is the classical optimal design area $\left[-1,1\right]\times[-1,1]$, and x_1, x_2 should be within a constrained region $x_1 + x_2 \le 1$, $x_1 + x_2 \ge -0.5$. It is assumed that the adhesive strength y is a quadratic function of x_1, x_2,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^{\ 2} + \beta_{22} x_2^{\ 2} + \varepsilon.$$

Actually it is an optimal design problem with linear constraints. In order to be consistent with the previous works, we set 12 design points, each with 2 components, i.e. $\mathrm{x}_i = \left(x_1, x_2\right)$, $i = 1, \cdots, 12$.

According to stage 1, Table 1 is the result of the D-optimal design points with equal measures.

Table 1. The 12 runs D-optimal support points of experiment of adhesive bonding strength.

Design point	Amount of adhesive	Bonding temperature
1	1.0000	0.0000
2	−1.0000	1.0000
3	−1.0000	0.5000
4	0.1223	0.1037
5	−0.3151	−0.1849
6	0.5000	−1.0000
7	0.1224	0.1035
8	1.0000	−1.0000
9	1.0000	0.0000
10	0.0000	1.0000
11	0.0000	1.0000
12	0.0000	1.0000

We calculate 12 runs D-optimal designs of adhesive bonding strength by several typical algorithms and the results are shown in Table 2. It is easy to see that the first stage of MDE algorithm gives the minimum $\det\left(M^{-1}\right)$ and the maximum D-efficiency in following 4 D-optimal designs, which means MDE algorithm is most accurate in following algorithms.

Table 2. The results comparison of 12 runs D-optimal design.

Algorithm	$\det\left(M^{-1}\right)$	Relative D-efficiency
MFE	3.696E-3	0.9711
GA	3.545E-3	0.9778
DE	3.101E-3	0.9999
MDE	3.099E-3	1.0000

We combine nearby design points in Table 1, run stage 2 of MDE algorithm, the result is shown in Table 3. There are 8 support points in this D-optimal design. The amount of adhesive and bonding temperature are shown in the column 2 and 3 respectively. Besides extreme vertices, there are two particular support points (0.1223, 0.1037) and (−0.3151, −0.1849) included in the optimal design. The data in column 4 is the related measures of the support points, and the results show that the measures of support points are not equal due to the existence of constraints. The values of variance function at the support points are displayed in the last column.

Table 3. The D-optimal design of experiment of adhesive bonding strength.

Support point	Amount of adhesive	Bonding temperature	r	$d\left(\mathrm{x},\xi\right)$
1	1.0000	0.0000	0.1530	5.9983
2	−1.0000	1.0000	0.1249	5.9996
3	−1.0000	0.5000	0.1166	5.9999
4	0.1223	0.1037	0.1549	6.0000
5	−0.3151	−0.1849	0.0537	6.0000
6	0.5000	−1.0000	0.1213	6.0000
7	1.0000	−1.0000	0.1227	6.0000
8	0.0000	1.0000	0.1529	6.0000

Figure 1 shows that we obtain an extraordinary accurate D-optimal design with constraints. Figure 1(a) is the contour map of variance function. If the measures at the support points are not considered, the variance function can show as Figure 1(b). Figure 1(c) is the variance function of D-optimal design. We find that the variance function is approximately equal to 6 at the D-optimal design support points and less than 6 at other points in the feasible region. The result is consistent with the general equivalence theorem of D-optimal design.

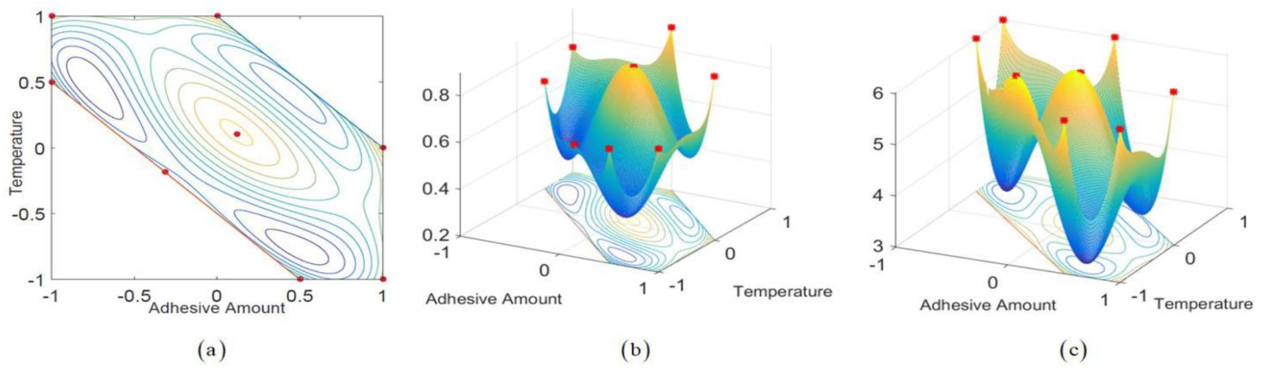


Figure 1. D-optimal design results of adhesive bonding strength experiment.

3.2. *Mixture Becker model with nonlinear constraint*

Liu et al. and Li et al. [9,10] discussed an example of mixture Becker model with nonlinear constraint. The Becker model is

$$y=\sum_{i=1}^{3}\beta_i x_i+\sum_{i<j}^{3}\beta_{ij}min\{x_i,x_j\}+\beta_{123}min\{x_1,x_2,x_3\},$$

and the feasible region is

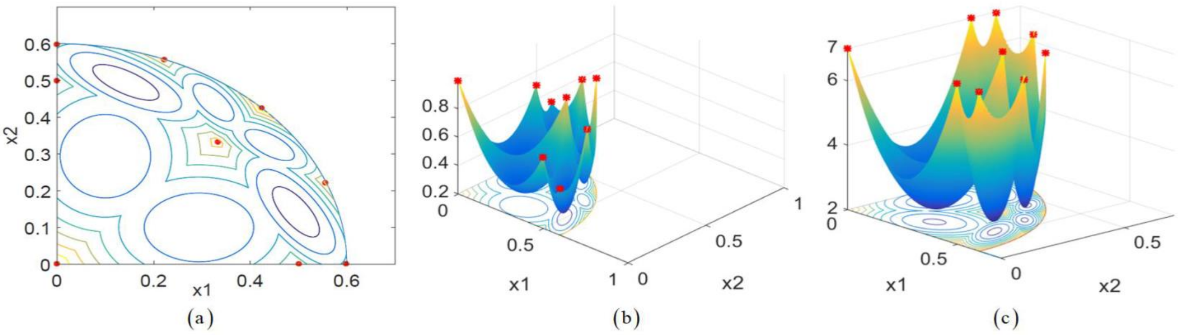
$$\chi=\left\{\left(x_1,x_2,x_3\right)|\sum\nolimits_{i=1}^{3}x_i=1,x_i\geq 0,x_1^{\ 2}+x_2^{\ 2}\leq 0.36\right\}.$$

We calculate this D-optimal design example with MDE algorithm, the calculations are shown as Table 4. We solve this problem by the first stage of MDE algorithm, then combine nearby points to get 9 support points, which are listed in the second column. The third column is the measures corresponding to the support points, which are obtained by the second stage of MDE algorithm. The variance function values at each support point are given in the last column, and all the values are less than 7.

Table 4. The D-optimal design of mixture Becker model with nonlinear constraint.

Support point	Coordinate	r	$d\left(\mathrm{x},\xi\right)$
1	(0.0000, 0.4999, 0.5001)	0.1247	6.9752
2	(0.3332, 0.3333, 0.3334)	0.1344	6.9876
3	(0.2211, 0.5577, 0.2212)	0.1330	6.8547
4	(0.0000, 0.5999, 0.4001)	0.0287	6.9650
5	(0.5000, 0.0000, 0.5000)	0.1249	6.9727
6	(0.4242, 0.4243, 0.1515)	0.1418	6.9673
7	(0.5578, 0.2207, 0.2214)	0.1418	6.8132
8	(0.0000, 0.0000, 1.0000)	0.1419	6.9999
9	(0.5999, 0.0000, 0.4001)	0.0288	6.9656

Figure 2 shows that the MDE algorithm is valid for nonlinear constrained mixture design problems. Figure 2(a) is the contour map of the variance function. Figure 2(b) is the variance function without considering measures. Figure 2(c) shows the variance function of Mixture D-optimal design of Becker model with nonlinear constraints. The value of variance function is close to 7 at each support points and less than 7 at the rest points in the feasible region, which means the general equivalence theorem holds.



(a) (b) (c)

Figure 2. Mixture D-optimal design results of Becker model with nonlinear constraint.

4. Conclusions

Reliable results can usually be obtained when MDE algorithm is used to solve constrained D-optimal design problems. For a given constrained design problem, the significant advantage of MDE algorithm is that the global optimal design can be obtained in many situations. Another advantage of MDE algorithm is that any set with sufficient initial values can lead to a global optimal design. The accuracy of the solution can be determined by the parameter in the algorithm, and more iterations are needed for high-precision solution.

MDE algorithm has low requirements to models, initial values and feasible regions, so it is suitable in solving constrained optimal design problems. We will try to apply MDE algorithm to the following optimal design problems: A-optimal design problems, E-optimal design problems, optimal design problems with heteroscedasticity, multi-response optimal design problems or the combination of the above problems.

Acknowledgements

The authors would like to appreciate all the reviewers and the editors for their careful reviews and constructive suggestions to help us improve the quality of this paper. This research was funded by National Nature Sciences Foundation of China under Grant (No.12071096), Scientific research project of Guangdong Bureau of Traditional Chinese Medicine (No.20201198) and Quality engineering construction project of Department of Education of Guangdong Province (No.220030301-36).

Conflicts of interest

The authors declare that they have no conflicts of interest.

References

1. D. C. Montgomery, E. N. Loredo, D. Jearkpaporn, M. C. Testik, Experimental designs for constrained regions, *Qual. Eng.*, **14** (2002), 587–601.
2. J. J. Borkowski, G. F. Piepel, Uniform designs for highly constrained mixture experiments, *J. Qual. Technol.*, **41** (2009), 35–47.
3. R. D. Snee, D. W. Marquardt, Extreme vertices designs for linear mixture models, *Technometrics*, **16** (1974), 399–408.
4. T. J. Mitchell, An Algorithm for the construction of D-optimal experimental designs, *Technometrics*, **16** (1974), 203–210.
5. R. D. Cook, C. J. Nachtsheim, A comparison of algorithms for constructing exact D-optimal designs, *Technometrics*, **22** (1980), 315–324.
6. A. Heredia-Langner, D. C. Montgomery, W. M. Carlyle, C. M. Borror, Model-robust optimal designs: A genetic algorithm approach, *J. Qual. Technol.*, **36** (2004), 263–279.
7. A. Heredia-Langner, W. M. Carlyle, D. C. Montgomery, C. M. Borror, G. C. Runger, Genetic algorithms for the construction of D-optimal designs, *J. Qual. Technol.*, **35** (2003), 28–46.
8. Y. Yu, D-optimal designs via a cocktail algorithm, *Stat. Comput.*, **21** (2010), 475–481.

9. Y. Liu, M. Q. Liu, Construction of uniform designs for mixture experiments with complex constraints, *Commun. Stat.-Theor. M.*, **45** (2015), 2172–2180.
10. G. Li, C. Zhang, Random search algorithm for optimal mixture experimental design, *Commun. Stat. Theor. M.*, **47** (2017), 1413–1422.
11. J. Duan, W. Gao, H. K. T. Ng, Efficient computational algorithm for optimal continuous experimental designs, *J. Comput. Appl. Math.*, **350** (2019), 98–113.
12. W. K. Wong, R. B. Chen, C. C. Huang, W. Wang, A modified particle swarm optimization technique for finding optimal designs for mixture models, *PLoS One*, **10** (2015), e0124720.
13. P. Y. Yang, F. I. Chou, J. T. Tsai, J. H. Chou, Adaptive uniform experimental design based fractional order particle swarm optimizer with non-linear time varying evolution, *Appl. Sci.*, **9** (2019), 5537.
14. K. Price, Differential evolution: A fast and simple numerical optimizer, In: *Proceedings of North American Fuzzy Information Processing*, 1996, 524–527.
15. R. Storn, K. Price, Differential Evolution − A Simple and Efficient Heuristic for global optimization over continuous spaces, *J. Global. Optim.*, **11** (1997), 341–359.
16. K. V. Price, R. M. Storn, J. A. Lampinen, *Differential evolution-A practical approach to global optimization*, Springer, Berlin, 2005.
17. Z. Zhao, J. Z. Wang, J. Zhao, Z. Y. Su, Using a Grey model optimized by Differential Evolution algorithm to forecast the per capita annual net income of rural households in China, *Omega*, **40** (2012), 525–532.
18. K. Rajesh, A. Bhuvanesh, S. Kannan, C. Thangaraj, Least cost generation expansion planning with solar power plant using Differential Evolution algorithm, *Renew. Energ.*, **85** (2016), 677–686.
19. W. Xu, W. K. Wong, K. C. Tan, J. Xu, Finding high-dimensional D-optimal designs for logistic models via differential evolution, *IEEE Access* **7** (2019), 7133–7146.
20. W. Deng, J. J. Xu, X. Z. Gao, H. M. Zhao, An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems, *IEEE T. Syst. Man Cy-S.*, (2020), 1–10.
21. J. Kiefer, Optimum experimental designs, *J. R. Stat. Soc. B.*, **21** (1959), 272–319.
22. R. H. Myers, D. C. Montgomery, G. G. Vining, T. J. Robinson, *Generalized Linear Models: With Applications in Engineering and the Sciences*, *2 Eds.*, John Wiely & Sons, New York, 2002.
23. A. Dasgupta, S. Mukhopadhyay, W. J. Studden, Compromise designs in heteroscedastic linear models, *J. Stat. Plan. Infer.*, **32** (1992), 363–384.
24. K. Chaloner, Optimal bayesian experimental design for linear models, *Ann. Stat.*, **12** (1984), 283–300.
25. R. D. Cook, V. V. Fedorov, Constrained optimization of experimental design, *Statistics*, **26** (1995), 129–178.
26. J. A. Cornell, *Experiments with mixtures: designs, models and the alysis of mixture data*, *3 Eds.*, John Wiely & Sons, New York, 2002.
27. J. Kiefer, J. Wolfowitz, Optimum extrapolation and interpolation designs II, *Ann. I. Stat. Math.* **16** (1964), 295–303.