

Constructing model robust mixture designs via weighted G-optimality criterion

Wanida Limmun^{a*}, Boonorm Chomtee^b and John Borkowski^c

^aDepartment of Mathematics and Statistics, Walailak University, Thasala, Nakhon Si Thammarat, 80160, Thailand

^bDepartment of Statistics, Kasetsart University, Chatuchak, Bangkok, 10903, Thailand

^cDepartment of Mathematical Sciences, Montana State University, Bozeman, MT, 59717, United States

CHRONICLE

Article history:

Received August 26 2018
Received in Revised Format
April 21 2019
Accepted April 29 2019
Available online
April 29 2019

Keywords:

Genetic algorithm
Model-robust design
Mixture experiment
G-optimality

ABSTRACT

We propose and develop a new G-optimality criterion using the concept of weighted optimality criteria and certain additional generalizations. The goal of the weighted G-optimality is to minimize a weighted average of the maximum scaled prediction variance in the design region over a set of reduced models. A genetic algorithm (GA) is used for generating the weighted G-optimal exact designs in an experimental region for mixtures. The performance of the proposed GA designs is evaluated and compared to the performance of the designs produced by our genetic algorithm and the PROC OPTEX exchange algorithm of SAS/QC. The evaluation demonstrates the advantages of GA designs over the designs generated using exchange algorithm, showing that the proposed GA designs have better model-robust properties and perform better than the designs generated by the PROC OPTEX exchange algorithm.

© 2019 by the authors; licensee Growing Science, Canada

1. Introduction

The increasing use of experimental design in the industry has resulted in the widespread utilization of mixture experiments. Mixture experiment is a special case of the response surface design that aims to optimize the end product by combining several components in a mixture. It has been applied for purposes of developing, improving and optimizing processes such as chemical formulation, pharmaceutical drug formulation, textile fiber production and food processing, etc. For example, if an engineer would like to optimize the burning rate of propellant in aircrew escape systems, the factors or the components of interest could be the proportions of fuel, oxidizer, and binder. Frequently, the components are constrained and the experimental region is an irregularly shaped polyhedron within the simplex, the optimal designs are commonly used. Optimal designs are usually constructed by specifying the model, choosing an optimality criterion, and using an algorithm to select n design points from a finite set of candidate points.

The true model is unnecessarily known in many applications, but can be assumed to be an unknown element of a known set of models. Frequently, after the data are collected, the model may be subsequently

* Corresponding author Tel.: +66 -7567-3000 ext. 2035. Fax number: +66 -7567-2004
E-mail: lwanda@mail.wu.ac.th (W. Limmun)

changed because the experimenter finds that some model effects may be small and insignificant leading to a decision of choosing a model containing fewer terms (i.e., a “reduced model”) in the analysis phase. If the initial model does not turn out to be the final model in the analysis phase, then the initial design will generally not be optimal. Therefore, the experimenter would opt for a design that allows model modification for a class of specific models. Consequently, the robustness of a design to model change, known as model robustness, plays an imperative role for addressing this problem.

In this study, we propose a genetic algorithm (GA) that can be used to create a mixture design with a robustness property. Genetic algorithm is a type of heuristic search motivated by the biological evolution theory of Charles Darwin. This algorithm is far more powerful and efficient than random search and exhaustive search algorithms, yet requires no extra information about the given problem (Kinnear, 1994). The effectiveness of the genetic algorithm depends on the choice of design region, encoding scheme, evaluation function, and genetic operators. Although the global optimal solution is not guaranteed, the chance of getting trapped in a local optimum region is smaller than some gradient-based search methods in a case that there are multiple local optima or the objective function is not smooth, noisy, or stochastic. Novel applications of genetic algorithms provide an alternative approach to classic exchange-point algorithms to generate designs. In some situation, the genetic algorithm gives an answer that is as good as the exchange algorithms. Examples of applying genetic algorithms to generate designs can be found in Borkowski (2003), Heredia-Langner et al. (2003, 2004), Drain et al. (2004), Park et al. (2006), Rodriguez et al. (2009) and Limmun et al. (2013).

Although there have been many studies on the problem focused on how to find the best or optimal design, relatively little attention has been paid to the problem of the robustness of the design to model misspecification, known as model robustness. A researcher is generally interested in generating designs that have good efficiency across a set of candidate models to assist him/her in choosing the best design regardless of the final model selected. In this work, we employ a genetic algorithm to extend the work of Limmun et al. (2013). The difference is that in Limmun et al. (2013), a single model was employed to generate model-robust mixture design, while in the current work, all possible reduced models are employed to generate model robust-mixture design. However, the experimental region remains to be an irregularly-shaped polyhedral region within the simplex. This proposed procedure offers a degree of flexibility in constructing designs that allows it to overcome restrictions that may limit the applicability of other more commonly used algorithms.

The paper is structured as follows. Section 2 gives the theoretical background of this research, while Section 3 gives the review of literature. The development of the weighted G-optimality criterion is presented in Section 4. The genetic algorithm implementation is delineated in Section 5, followed by numerical study with examples in Section 6. Finally, the main conclusions of the paper are presented in Section 7.

2. Theoretical background

2.1 Notation and mixture models

Mixture experiment is an experiment in which the measured response depends only on the relative proportions of the different components of a mixture, and the design factors are the proportions of the q components of a mixture. In a standard mixture experiment, the proportion of component (x_i) needs to satisfy the following simplex properties:

$$\sum_{i=1}^q x_i = 1 \text{ and } 0 \leq x_i \leq 1. \quad (1)$$

Because of these constrained proportions, an alteration of the proportion of one component must cause a change in the proportion of at least one of the other components. The shape of the design space for

mixture experiments given by Eq. (1) is a regularly $(q - 1)$ -dimensional simplex. In many mixture experiments, the single-component proportions must also satisfy single-component constraints (SCCs) expressed as:

$$0 \leq L_i \leq x_i \leq U_i \leq 1 \quad ; i = 1, 2, \dots, q, \quad (2)$$

and possibly multiple-component constraints (MCCs) expressed as:

$$C_j \leq A_{1j}x_1 + A_{2j}x_2 + \dots + A_{qj}x_j \leq D_j \quad ; j = 1, 2, \dots, h, \quad (3)$$

where L_i and U_i are lower and upper bounds of the SCCs, respectively, and A_{ij} are scalars defining multivariate linear with C_j and D_j being the lower and upper bounds of the MCCs, respectively. If the SCCs and/or MCCs are placed on the component proportions, then the experimental region will alter from a simplex to an irregularly shaped polyhedron within the simplex. For additional information of constrained mixture experiments, see Cornell (2002) and Smith (2005).

Typically, mixture experiment models are of the Scheffé type which are no-intercept models with matrix form given by the following:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (4)$$

where $\mathbf{y} = (y_1, y_2, \dots, y_n)'$ is the $n \times 1$ vector of measured responses, \mathbf{X} is the $n \times p$ matrix with columns associated with the p model effects (such as component proportions and cross-products terms), $\boldsymbol{\beta}$ is the $p \times 1$ vector of model parameters, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)'$ is the vector of random error associated with natural variation of \mathbf{y} around the underlying surface assumed to be independent and identically normally distributed with zero mean and variance $\sigma^2 \mathbf{I}_n$. Consider the q -component (Scheffé) quadratic mixture model below as an example:

$$y = \sum_{i=1}^q \beta_i x_i + \sum_{i < j}^q \beta_{ij} x_i x_j + \varepsilon. \quad (5)$$

In the full matrix, the information matrix, $\mathbf{X}'\mathbf{X}$, is non-singular matrix and the least squares estimate of $\boldsymbol{\beta}$ is $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ which is equivalent to the maximum likelihood estimator under normal error. Also, the variance-covariance of the regression coefficients is $Var(\mathbf{b}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. The design matrix for the Scheffé quadratic model is defined as $\mathbf{X} = (x_1, x_2, \dots, x_q, x_1x_2, x_1x_3, \dots, x_qx_{q-1})'$. In this paper, we concentrate on the Scheffé quadratic mixture model. For further information on the mixture model, see Cornell (2002).

2.2 G-optimality criterion

The design optimality criteria are single value criteria that represent different variance properties of a design. The theory of optimality criteria was first developed by Kiefer and Wolfowitz (1959). The alphabetic optimality criteria including the D, A, G and IV criteria are the commonly used design optimality criteria. These optimality criteria involve selecting the row of \mathbf{X} so as to optimize some function of Fisher's information matrix ($\mathbf{X}'\mathbf{X}$). The Fisher's information matrix ($\mathbf{X}'\mathbf{X}$) is a scaled version of the inverse of the variance-covariance matrix of the least square estimators of the model parameters for optimizing the desired properties. D- and A-optimality are focused on parameter estimation, while G- and IV-optimality are focused on prediction variance. The G- and IV-efficiency are calculated over a continuous design region. For a discussion of the composite design optimality criteria, see Atkinson et al. (2007) and Zaharn et al. (2003).

The best known and most often used design optimality criterion is D-optimality. The G-efficiency is particularly appealing when the experimenter is interested in the prediction variance property of the response. The G-optimally criterion is not differentiable; G-optimal designs are harder to construct and

study than D-optimal designs. The equivalence theorem says that these two design criteria are identical when the design is expressed as a probability measure (Kiefer and Wolfowitz, 1960). The equivalence of the two criteria does not hold for the exact designs. All designs used by experimenters in practice are the exact designs by definition because the number of trials for any design point is a specified integer n .

The design optimality criterion that we focus on is the G-criterion because there is no commercially available software that currently constructs designs using it. The G-optimality criterion seeks to find a set of points that will minimize the maximum scaled prediction variance (SPV) over the entire design region. Formally, if the prediction equation is expressed as $\hat{y}(\mathbf{x})$, we define the SPV as:

$$\text{SPV} = \frac{n\text{Var}(\hat{y}(\mathbf{x}))}{\sigma^2} = n\mathbf{x}'^{(m)}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}^{(m)}, \quad (6)$$

where $\text{Var}(\hat{y}(\mathbf{x}))$ is the variance of the prediction response at a given point \mathbf{x} , n is the number of design points, σ^2 is the process variance, and $\mathbf{x}^{(m)}$ is a vector of a point in the design region expanded to model form. For example, if there are three components and the model is the Scheffé quadratic model, then $\mathbf{x}^{(m)} = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3)$ and \mathbf{X} has the dimension of $n \times 6$. The G-efficiency of a design is defined as follows:

$$G_{\text{eff}} = \frac{100p}{\max_{\mathbf{x}^{(m)} \in R} \text{SPV}}, \quad (7)$$

where p is the number of parameters in the model and R is the constrained design region. The G-efficiency is based on the performance of SPV. The G-criterion is a minimax criterion, that is, the design should minimize the maximum prediction variance in the design region. A G-optimal design is the one that maximizes G-efficiency over the region of interest.

In this paper, the G-efficiency is approximated over a set of points from an extreme vertices design because the maximum scaled prediction variance typically occurs at a boundary point such as a vertex, face centroid or edge centroid. As a practical rule of thumb, Wheeler (1972) suggested that a mixture design with G-efficiency $\geq 50\%$ is a good design for practical purposes and showed that pursuit of higher efficiencies is not generally justified in practice. Our algorithm guarantees this much efficiency as shown in Section 6.

2.3 Genetic algorithm

A genetic algorithm is basically a search heuristic or optimization technique that is inspired by Charles Darwin's theory of natural process of evolution, which states that an organism that can adapt best to their environment compared with other members of their species will survive. This algorithm was developed by John Holland in 1975 and was popularized through the work of Goldberg in 1989. The development and success of genetic algorithms have greatly contributed to a wider interest in computational approaches based on natural phenomena and it is now a major strand of the wider field of Computational Intelligence (Engelbrecht, 2002).

Genetic algorithm serves as an initial population of potential solutions (parent chromosomes) to a problem, and with the best one being the fittest member of the population. Then through evolutionary reproduction operators, the current parent population passes some of its properties (genes) to offspring chromosomes. A subset of the best parent and offspring will be passed onto future generation, and the reproduction process is repeated for many generations until a termination criterion is satisfied. The genetic algorithm is commonly used to generate high quality solutions to optimization and search problems by relying on genetic operators such as selection, crossover, and mutation. The objective of the selection operator is to emphasize the fitter chromosomes in a population in hopes that their offspring will in turn have even higher fitness level while keeping the population size constant. The *selection*

operator is used to choose the best and the fittest chromosomes for reproduction based on a probability distribution defined by the user. The *crossover* operator is the main process of the genetic algorithm to vary the chromosomes from one generation to the next. This operator chooses one or more crossover points that separate the encoded chromosomes, and then swaps a subsequence of two of the chosen chromosomes to create two offspring. The *mutation* operators act on an individual chromosome to flip one or more allele values. This process helps overcome trapping at local optimum. Typically, the mutation happens with a very low probability. For an introduction and review of genetic algorithms, see Goldberg (1989), Michalewicz (1992), Kinnear (1994), and Haupt and Haupt (2004).

3. Related works

Model robustness was addressed by several researchers. DuMouchel and Jones (1994) presented a Bayesian approach for choosing an optimal design robust to the modeling assumptions. This approach considered model terms as either primary or potential terms. Li and Nachtsheim (2000) constructed a model robust design by considering a class of factorial designs that can be used for efficient estimation of main effects and any combination of g interactions where g is specified by the experimenter. Two weighted criteria that are based on a design's estimation capacity and the D-criterion are combined. Heredia-Langner et al. (2004) implemented a genetic algorithm approach to optimize a weighted D-optimality criterion from a very large search space for finding the robust design. This approach creates a design that minimizes the volume of the joint confidence region for the vector of model coefficients for a given set of possible models. Chung et al. (2007) employed a genetic algorithm to generate designs that simultaneously optimize the scaled prediction variances (SPVs) for both the mean-SPV model and the slope-SPV model for mixture-process experiments with control and noise variables. They show how designs that are robust to levels of control-by-noise interaction can be obtained. Anderson-Cook et al. (2009) developed graphical approach for assessing the model robustness of the designs to model misspecification using the expected mean square error (EMSE) over cuboidal regions of interest. Smucker et al. (2011) developed modified Fedorov exchange algorithm for creating the exact robust design when the true model form is unknown. This approach optimizes the determinants of the information matrices for a given set of possible models.

Genetic algorithm has been used in design of experiments to solve a variety of problems. Borkowski (2003) presented a genetic algorithm technique for generating near-optimal D, A, G, and IV exact n -point response surface designs for the second-order model in the hypercube. The local grid search algorithm is adopted to these near-optimal designs for finding the optimal exact designs. Heredia-Langner et al. (2003) developed genetic algorithm for generating D-optimal designs. This approach is able to implicitly search vast spaces and it eliminates the need to explicitly consider a candidate set of experimental points. They found that the performance of the genetic algorithm was at least as good as the other procedures such as the coordinate exchange, k-exchange, and modified-Fedorov algorithms. Goldfarb et al. (2005) used a genetic algorithm to generate statistical designs for experiments involving mixture variables and process variables using either the mean or slope prediction variance and applied the G-optimality criterion for the objective function. They show how designs that can improve the prediction variance property of either the mean or the slope of the designs can be constructed. Park et al. (2006) employed a genetic algorithm for finding the cost-constrained G-efficient designs over cuboidal region when the experimental budget or cost are limited. The genetic algorithm can create the G-efficient design with minimum cost. Rodríguez et al. (2010) introduced a new coordinate exchange algorithm to create highly G-efficient experimental design on cuboidal region. Limmun et al. (2013) presented a genetic algorithm approach for generating D-optimal designs for mixture experiments where the experimental region is an irregular shaped region that is a subspace of a simplex. They found that the performance of the genetic algorithm was at least as good as the other procedures such as extreme vertices design or conventional exchange algorithms. In the optimal design theory, such objectives can be expressed using the weighted optimality criteria. Morgan and Wang (2010) proposed a broad class of weighted variances based on quadratic forms of a diagonal matrix \mathbf{W} that allows a researcher to specify

relative importance of information for any effects in treatment–control experiments. The weighted optimality criteria are constructed by applying standard optimality criteria to weighted information matrices. Borkowski et al. (2010) developed the weighted D, G, A, and IV-efficiencies for response surface designs by assigning the prior probability to a suite of potential models based on the concepts of the model heredity principle. Emmett et al. (2011) proposed a weighted V-optimality criterion that seeks to minimize the average variance of prediction for a continuous response. The weights are derived from a logistic regression model. Stallings and Morgan (2015) proposed a weighted information matrix, allowing precise design selection according to expressed relative interest in different functions in the estimation space.

4. Weighted design optimality criteria

The weighted optimality criteria provide a meaningful approach of expressing the interest in a given set of linear estimable functions using the theory of optimality criterion for estimating the given functions of interest. The weighted G-optimality (WG-optimality) criterion is developed in this paper. The WG-optimality aims to minimize the weighted average of the maximum SPV in the design region over a set of reduced models. The strong heredity (SH) and weak heredity (WH) principles developed by Chipman (1996) are adopted for constructing reduced models. SH requires that if a model contains the quadratic term $(\beta_{ij}x_ix_j)$, then it must contain both corresponding linear effects $(\beta_ix_i$ and $\beta_jx_j)$; WH requires that if a model contains the quadratic term $(\beta_{ij}x_ix_j)$, then it must contain one or both of the corresponding linear effects $(\beta_ix_i$ or $\beta_jx_j)$. Then, we additionally assume that all linear component effects $(\beta_ix_i$ for $i = 1, \dots, q)$ must occur in each possible reduced model. Thus, the set of SH reduced models of the full Scheffé quadratic model are identical with the set of WH reduced models because of the additional condition that all linear component effects must be included in each reduced model.

Suppose the full model is the Scheffé quadratic mixture model with three mixture components. The sets of SH and WH reduced models of the full Scheffé quadratic mixture model are equal, and there are eight reduced models for both. There are 1, 3, 3 and 1 reduced models having 6-, 5-, 4- and 3-parameter, respectively, as shown in Table 1. In Table 1, 1's and 0's in the 'Terms in model' columns represent the presence and absence of the terms in the model parameters, respectively. Column p is the number of model parameters.

A complete enumeration of the set of subset models (possible reduced models) is necessary for calculating the WG-efficiency. After complete enumeration of the set of reduced models, the G-optimality criterion is calculated for each reduced model. Recall that the G-efficiencies are approximated over the set of points from an extreme vertices design. Finally, the WG-optimality criterion is estimated using the sum of the products between the weight and the G-efficiency of each model; that is, the WG-efficiency can be expressed as:

$$WG = \sum_{i=1}^r w_i G_i \quad (8)$$

where $G_i = \frac{100p_i}{\max_{x(m) \in R} (x'(m)(X^i X^i)^{-1} x(m))}$ is the G-efficiency for model i

r is the number of reduced models for a given full model

w_i is the weight for model i

p_i is the number of parameters for model i

X^i is a mixture model matrix for model i .

Table 1

The set of reduced models for the full Scheffé quadratic mixture model with three components

Model	Terms in model						<i>p</i>
	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	
1 st	1	1	1	1	1	1	6
2 nd	1	1	1	0	1	1	5
3 rd	1	1	1	1	0	1	5
4 th	1	1	1	1	1	0	5
5 th	1	1	1	0	0	1	4
6 th	1	1	1	0	1	0	4
7 th	1	1	1	1	0	0	4
8 th	1	1	1	0	0	0	3

Even though the concept of weighted optimality criteria is not up-to-date, further generalization that only models having an equal number of model parameters receive equal weight adopted in this paper is up-to-date. To illustrate this concept, we treat each parameter as equally important and we wish to assign a weight to the set of reduced models in order to reflect its importance relative to the number of model parameters. However, if the researcher has prior information regarding model weighting, one can assign those weights to the models of interest. Given the above criteria, we now define the weight associated with each reduced model. Suppose model *i* has *j* parameters, then, the weight for model *i* is defined as

$$w_i = \frac{\psi_j}{m_j}, \tag{9}$$

- where ψ_j is the *j*th weighting factor, $j = q, q + 1, \dots, s - 1, s$
- q is the number of parameters of the linear mixture model
- s is the number of parameter of the full mixture model
- m_j is the number of reduced models equivalent to a *j*-parameter model.

The weighting factors $\psi_q, \psi_{q+1}, \dots, \psi_{s-1}, \psi_s$ are positive and subject to the restriction $\sum_{j=q}^s \psi_j = 1$. To compute the weighting factors, suppose the relative weighting factor (supplied by the experimenter) is defined as:

$$R = \frac{\psi_s}{\psi_q}. \tag{10}$$

The value of *R* is an arbitrary constant such that $R \geq 1$, and is the ratio of the weighting factor of the full model to the weighting factor of the linear model. Suppose also that the experimenter sets $R = 100$. That is, the experimenter weights the full mixture model 100 times the weight to be assigned to the linear mixture model. The number of levels for a weighting factor is $k = s - q + 1$. Thus, the range of the weighting factor is:

$$\psi_k = \psi_s - \psi_q = R\psi_q - \psi_q = (R - 1)\psi_q. \tag{11}$$

The increment value of the weighting factor is given by:

$$INCR = \frac{\psi_k}{k-1} = \frac{(R-1)\psi_q}{k-1}. \tag{12}$$

Given that there are *k* equispaced levels of the weighting factor, the values of the weighting factor can be represented as follows:

$$\begin{aligned} \psi_q &= \psi_q, \\ \psi_{q+1} &= \psi_q + INCR, \end{aligned}$$

$$\psi_{q+2} = \psi_q + 2INCR,$$

$$\vdots$$

$$\psi_{q+(k-1)} = \psi_q + (k-1)INCR,$$

when $s = q + (k - 1)$. Then the weighting factor can be rewritten in the form:

$$\sum_{j=q}^s \psi_j = \psi_q + (\psi_q + INCR) + \cdots + (\psi_q + (k-1)INCR) = 1 \quad (13)$$

which may be reduced to:

$$k\psi_q + \binom{k}{2} INCR = 1. \quad (14)$$

Therefore, the minimum weighting factor can be written as:

$$\psi_q = \frac{(k-1)INCR}{R-1}. \quad (15)$$

After simplifying, substitution of Eq. (15) into Eq. (12) yields the following equation:

$$INCR = \frac{2(R-1)}{k(k-1)(R+1)}. \quad (16)$$

The other weighting factors follow directly via substitution.

To calculate the WG-efficiency, the Scheffé linear models receive the smallest weight, while the full Scheffé quadratic model receive the largest weight, with other models having equal number of terms sharing a common weight. Unlike the case of equally weighted models, we treat each parameter as equally important and the weights are assigned to the set of reduced models in order to reflect its importance relative to the number of model parameters.

For three components, suppose that the experimenter weights the full mixture model 100 times the weight to be assigned to the linear mixture model. Therefore, $R = 100$, $q = 3$, $s = 6$, $k = 4$, and $INCR = \frac{33}{202}$.

We now consider the 7th reduced model,

$$E(y) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2,$$

that has 4 parameters. The weighting factor value of all models having 4-parameter (ψ_4) is defined as

$$\psi_4 = \psi_3 + INCR = \frac{1}{202} + \frac{33}{202} = \frac{34}{202}.$$

There are three models having 4 parameters, $m_4 = 3$, so the weight of the 7th model is

$$w_7 = \frac{\psi_4}{m_4} = \frac{34}{606}.$$

That is, the weight for models 5, 6, and 7 having 4 parameters are equal to $\frac{34}{606}$. For three components, there are 8 reduced models which have 1, 3, 3 and 1 reduced models having 6, 5, 4 and 3 parameters with weighting factor values of $\frac{100}{202}$, $\frac{67}{202}$, $\frac{34}{202}$ and $\frac{1}{202}$, respectively.

We now consider the weight with varying the R values. Table 2 summarizes the corresponding weight for the set of reduced models in Table 1 with $R=1$, 10, 100 and 1,000 for three components. We found that (i) when the values of R is 10, 100 and 1,000, the weight of the models perform nearly equally for each of the number of the parameters except the model having the q -parameter (the Scheffé linear model); and (ii) the pattern of the weights for each of the number of the parameters with $R=1$ differs from that of other R values. The experimenter has the freedom to choose the ratio of the weighting factor of the full

model to the weighting factor of the linear model (R) where $R \geq 1$. However, we do not recommend using $R = 1$.

Table 2

Comparison of the weight of each model with $R=1, 10, 100$ and $1,000$ for three components.

R	Model							
	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th
1	$\frac{1}{4}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{4}$
10	$\frac{10}{22}$	$\frac{7}{66}$	$\frac{7}{66}$	$\frac{7}{66}$	$\frac{4}{66}$	$\frac{4}{66}$	$\frac{4}{66}$	$\frac{1}{22}$
100	$\frac{100}{202}$	$\frac{67}{606}$	$\frac{67}{606}$	$\frac{67}{606}$	$\frac{34}{606}$	$\frac{34}{606}$	$\frac{34}{606}$	$\frac{1}{202}$
1000	$\frac{1000}{2002}$	$\frac{222}{1999}$	$\frac{222}{1999}$	$\frac{222}{1999}$	$\frac{167}{3003}$	$\frac{167}{3003}$	$\frac{167}{3003}$	$\frac{1}{2002}$

5. Genetic algorithm for constructing weighted G-optimal design

The genetic algorithm is very appropriate for construction of a design when the optimality criterion is crucial to work and/or where the experimental region is constrained such as in mixture experiments. We use the genetic algorithm developed by Limmun et al. (2013) to construct the model-robust designs. Throughout this paper, the chromosomes are encoded using real-value encoding instead of binary or another encoding. Real value encoding is flexible enough to allow a unique representation for every variable, compatible with other optimization algorithms (such as simulated annealing), easy to interpret, and can be modified to adjust for many applications. The reproduction process including *blending*, *between-parent crossover*, *within-parent crossover*, and *mutation* operators is a process that usually operates on the genes to produce offspring chromosomes. The success probabilities for these operators are α_b , α_{bc} , α_{wc} and α_m , respectively. A gene or set of genes is altered when a probability test is passed (PTIP). A probability test is a Bernoulli distribution with probability of success α_i . If u is a random deviate from a continuous uniform distribution on the interval $[0,1]$ and $0 \leq u \leq \alpha_i$, then a probability test is passed and the operator is applied. Otherwise, the chromosome remains untouched. Furthermore, if a probability test is passed, a random variate ε from a normal distribution with a mean of 0 and a standard deviation of V is added to a gene to form a new gene.

In this algorithm, the potential design (solution) is represented as a chromosome C having an $n \times q$ matrix, where n is the number of the design points, and q is the number of mixture components. The components are represented as genes on a chromosome. Thus, each row in C is a gene $\mathbf{x}_i = [x_{i1} \ x_{i2} \ \dots \ x_{iq}]$ and represents one experimental mixture. Suppose there are five design points, a chromosome $C = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5)'$. We wish to optimize the objective function F that measures a chromosome's fitness as a solution to the function. The objective function F reflects information about both the quality and feasibility of solutions and takes a chromosome as input and outputs of an objective function value. We propose the WG-efficiency as the objective function for assessing the fitness of each chromosome (design) as shown in Section 4. Recall that the goal of the WG-efficiency is to obtain a design that minimizes the weighted average of the maximum scaled prediction variance throughout the entire experimental region for a set of reduced models representing possible model misspecification. Specifically, the principal purpose is to find a design that maximizes objective function F where

$$F = WG \quad (17)$$

Our genetic algorithm process is as follows:

- Step 1: Determine the number of chromosomes (M), number of generation (I), selection method, blending rate (α_b), crossover rates (α_{bc} and α_{wc}) and mutation rate (α_m). We use 21 as the number of chromosomes. We investigated the number of generations to convergence when running the GA. The choice of each α_i and V as the parameter of genetic algorithm is based on the judgment of the author through the examination of convergence rate. However, we also investigated the values of parameters before running the GA. Larger α_i and V values are used for the early iterations and smaller values are used for the later iterations to allow for large changes early and convergence to a more accurate solution later.
- Step 2: Generate an initial population of M chromosomes (mixture designs) with a random value. Assume the number of chromosomes (M) is odd. We use the real value encoding with four decimal places to encode each chromosome. To generate the initial population, a random sample is first taken in a hypercube. Then each sampled point in a hypercube is mapped to the constrained mixture space by applying the function used by Borkowski and Piepel (2009).
- Step 3: Evaluation of fitness value of chromosomes in the population by calculating the objective function F . We use the WG-efficiency as the objective function.
- Step 4: Chromosomes selection. We use a selection process with elitism for which the chromosome with the largest fitness at each generation is unchanged and is kept to be in the next generation. The remaining $M - 1$ chromosomes are randomly paired for the reproduction process.
- Step 5: Produce offspring of the next generation by using genetic operators: blending, between-parent crossover, within-parent crossover, and mutation. The blending process, which is a reproduction, is an applied operation between two parents. Afterwards, two crossover operators, which are the Between-Parent Crossover and the Within-Parent Crossover, are also performed. For the Between-Parent Crossover, offspring are obtained by swapping the tail fractions of two genes of two paired parents. For the Within-Parent Crossover, an offspring is obtained by swapping the tail fractions within a gene. Finally, mutation is applied in which one of the mixture components is randomly chosen and its proportion is mutated using random normal deviates to form the new mixture component, and after that the remaining mixture components are adjusted to ensure that sum of all mixture proportions equals one.
- Step 6: Evaluation of fitness value of chromosomes in the offspring by calculating the objective function F . According to the principle of survival of the fittest, the more adaptive chromosomes are kept, and the less adaptive ones are discarded in the course of generating a new population. That is, the fitter chromosome in the parent/offspring pair is retained for the next generation.
- Step 7: Repeat Steps 5 and 6 for I generations.
- Step 8: Employ a local grid search to the best design to further improve objective function F yielding the WG-optimal design. A local grid search looks for designs in a small neighbourhood of best design. This is accomplished by perturbing the component proportions by small increments to search for further improvements in the objective function F . This continues until no further improvement is found.

6. Numerical study and results

In this section, we present two examples illustrating WG-optimal exact designs produced by our GA and compare them to the designs generated by the PROC OPTEX exchange algorithm of SAS/QC (SAS Institute 2012). Fedorov's algorithm via PROC OPTEX is used to find the D-optimal designs and the A-optimal designs that have maximal G-efficiency for the model individually. All designs generated by the exchange algorithm in SAS will be referred simply as EA designs. We used a candidate list where the points are in a resolution 0.01 grid placed over the design space. The GA designs are generated using author-written program with MATLAB.

The fraction of design space (FDS) plots show the distributions of the SPV over the region of interest for different models on the same graph. Thus, in this work we use the FDS plots for assessing the robustness of a design to model change. For FDS plots, we will restrict the study to the quadratic and linear models because the curves for these models give a manageable summary of that design's prediction variance performance. Therefore, only the Scheffé quadratic model (or full model) and linear models (the most reduced model) will be considered for the assessment of model robustness for a design. For additional information of the FDS curves for examining model robustness, see Goldfarb et al. (2004) and Ozol-Godfrey et al. (2005).

In this paper, we also investigate the values of genetic parameters before running the genetic algorithm. We begin by sequentially creating many sets of values for the α_i and V that correspond to a certain range of values supplied by the experimenter. The best set of the α_i and V values is the set that generates the highest objective function values and then becomes very stable throughout the later iterations. For GA-generated designs, we use $0.02 \leq \alpha_b, \alpha_{bc}, \alpha_{wc} \leq 0.1$; $0.05 \leq \alpha_m \leq 0.2$ as the range of α_i and $0.1 \leq V \leq 0.3$ as the range of V values. The α_i and V values are reset to a smaller value after 1,500 generations and this algorithm was run for 9,000 generations for the WG-optimal exact designs.

6.1 Three components with only SCCs example

A poultry feed blend experiment with three mixture components taken from Snee and Rayner (1982) serves as an initial example in illustrating our method. The objective of the experiment is to evaluate the effects of the component proportions on the weight gain of chicks consuming the feed. There are three ingredients in the mixture process; maize (x_1), fish (x_2), and soybean (x_3). These ingredients have the following lower and upper proportion constraints:

$$0.3 \leq x_1 \leq 0.8; \quad 0.0 \leq x_2 \leq 0.3; \quad 0.0 \leq x_3 \leq 0.5.$$

The region of the poultry experiment has six vertices. The full model under consideration is the Scheffé quadratic mixture model. The complete 6-term model is:

$$E(y) = \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \beta_{12}x_1x_2 + \beta_{13}x_1x_3 + \beta_{23}x_2x_3.$$

For the EA designs, the candidate list for this example consisted of 1,316 points constituting a grid of resolution 0.01 placed over the design space. We consider six competing EA and GA designs with $n = 10$ runs. The competing designs are the GA designs based on the WG-criterion with $R = 1, 10, 100$ and 1,000 (WG10-R1, WG10-R10, WG10-R100 and WG10-R1000), the EA designs based on the D-criterion that has maximal G-efficiency using STATIC and NONE options as the type of coding (DS10, DN10), and the EA designs based on the A-criterion that has maximal G-efficiency using STATIC and NONE options as the type of coding (AS10, AN10). We found that the DS10 design is identical with the DN10 design, so only the DS10 design will be referenced. As shown in Table 3, the EA designs based on A-criterion that has maximal G-efficiency were also studied but had poorer SPV properties than the other designs, and therefore excluded in this study. Therefore, we compare the five designs. The WG10 designs are referred to as the WG-optimal exact design using genetic algorithm with all the R values. The WG10-R1, WG10-R10, WG10-R100, WG10-R1000 designs are shown in Table A1 in the Appendix.

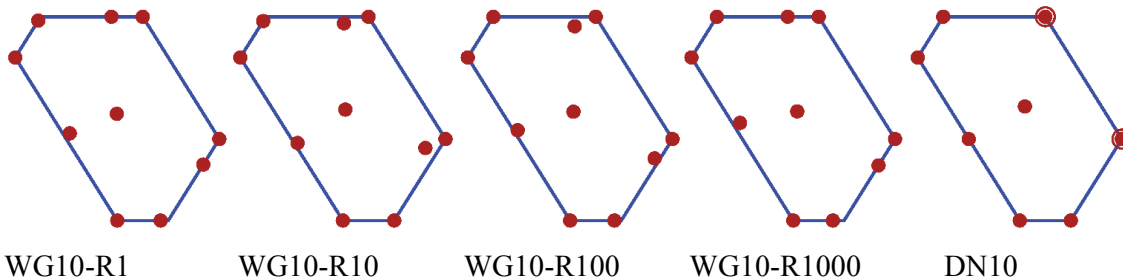


Fig. 1. The competing designs for the poultry experiment. ● and ⊙ represent 1 and 2 replicates of that design point, respectively.

The differences in the distributional patterns of all competitive designs are shown in Fig. 1. We notice that the WG10 designs have similar patterns of points since they contain four or five vertices, near the vertices, near an edge centroid and near the overall centroid, while the patterns of points of the DS10 design lie on all vertices with varying replication and contain edge centroid and near overall centroids. Note that the WG10 designs have no replicated points.

Table 3

The approximate WG-efficiencies, approximate G-efficiencies and Max SPV values for eight competing designs for the poultry experiment.

Design	R	WG-efficiency	Model type	G-efficiency	Max SPV values
WG10-R1	1	77.7845	Quadratic model	87.25%	6.8772
			Linear model	79.41%	3.7778
WG10-R10	10	80.4258	Quadratic model	88.76%	6.7600
			Linear model	77.45%	3.8733
WG10-R100	100	81.0606	Quadratic model	88.67%	6.7667
			Linear model	77.73%	3.8596
WG10-R1000	1000	80.6900	Quadratic model	87.55%	6.8530
			Linear model	78.15%	3.8389
DN10, DS10	1	74.1677	Quadratic model	72.45%	8.2820
	10	74.0465			
	100	74.0225			
	1000	74.0198			
AN10	1	60.7252	Quadratic model	69.41%	8.6440
	10	62.3689			
	100	62.6944			
	1000	62.7302			
AS10	1	36.0593	Quadratic model	35.52%	16.8922
	10	34.0072			
	100	33.6008			
	1000	33.5562			
	1	60.7252	Linear model	64.66%	4.6393
	10	62.3689			
	100	62.6944			
	1000	62.7302			
	1	36.0593	Linear model	46.21%	6.4918
	10	34.0072			
	100	33.6008			
	1000	33.5562			

Table 3 gives the approximate WG-efficiencies, approximate G-efficiencies and Max SPV values for five competing designs. The WG-efficiencies of the WG10 designs are comparable except the WG10-R1 design. The WG-efficiencies values of the DS10 design is quite comparable for all the R values. The G-efficiencies and maximum SPV values of the WG10 designs are quite comparable for both the Scheffé quadratic and Scheffé linear models. The EA designs perform poorly when examining the G-efficiencies and maximum SPV values for both the Scheffé quadratic and Scheffé linear models. The WG10-R1 design is inferior to other WG10 designs. The EA designs perform poorer than the WG10 designs in terms of the higher WG-efficiencies, the higher G-efficiencies and lower maximum SPV values for both the Scheffé quadratic and Scheffé linear models.

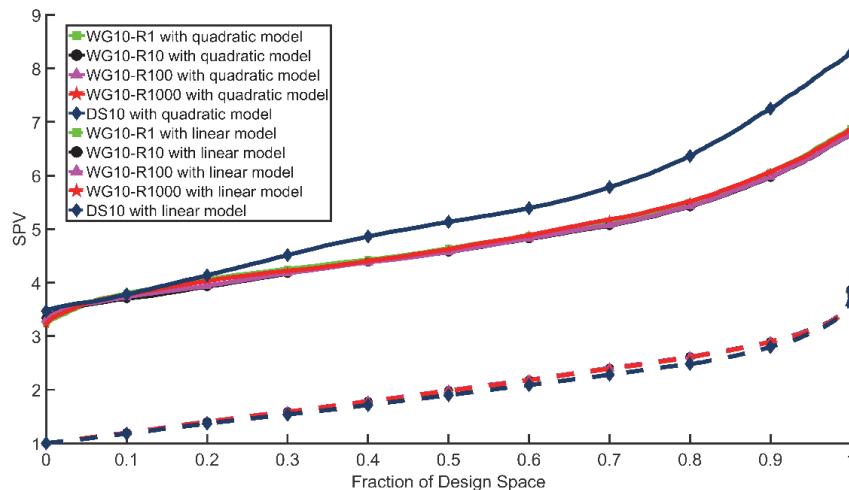


Fig. 2. FDS plot for comparing the WG10 designs and the EA designs for the Scheffé quadratic and linear models for the poultry experiment

Fig. 2 shows the comparison between five competing designs. The FDS curve for both the Scheffé quadratic and linear models of the DS10 design performs poorly for most of the design space compared to the WG10 designs. For the Scheffé quadratic and linear models, the WG10 designs have similar SPV distribution for most of the design space. For the majority of the design space, the WG10 designs appear to have good robustness properties because it has much flatter and lower FDS curves for both the Scheffé quadratic and linear models, indicating more desirable SPV distributions. All designs are nearly identical for the Scheffé linear model except for the largest 20% of SPV values for which the WG10 designs are superior.

6.2 Three components with SCCs and MCCs example

The second case study is a three-component mixture experiment with both SCCs and MCCs, taken from Snee (1979). The three SCCs are the following:

$$0.1 \leq x_1 \leq 0.5; \quad 0.1 \leq x_2 \leq 0.7 \text{ and } 0.0 \leq x_3 \leq 0.7.$$

Then, the space is further constrained by the following MCCs:

$$0.9 \leq 0.85x_1 + 0.9x_2 + x_3 \leq 0.95$$

$$0.4 \leq 0.7x_1 + x_3.$$

The boundary formed has six vertices. The full model under consideration is the Scheffé quadratic mixture model:

$$E(y) = \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \beta_{12}x_1x_2 + \beta_{13}x_1x_3 + \beta_{23}x_2x_3.$$

The EA designs are also used to solve this problem for performance comparison purposes. The competing designs with $n = 10$ runs are considered in this example: the GA design based on the WG-criterion with $R = 1, 10, 100$ and $1,000$ (WGM10-R1, WGM10-R10, WGM10-R100, WGM10-R1000), and the EA designs based on D-criterion that has maximal G-efficiency using STATIC and NONE coding (DSM10, DNM10). We found that the DSM10 design is identical with the DNM10 design, so only the DSM10 design will be referenced. Therefore, we compare five designs. The WGM10 designs are referred to as the WG-optimal exact design using GA with all the R values. The set of points of the WGM10 designs is given in Table A2 in the Appendix.

Fig. 3 shows the differences in the distributional patterns of all competitive designs. We notice that the WGM10 designs have similar patterns of points since they contain three vertices, near the vertices, near an edge centroid and near the overall centroid, while the patterns of points of the DSM10 design lie on all vertices with varying replication and contain near overall centroids with replication. Note that the WGM10 designs have no replicated points.

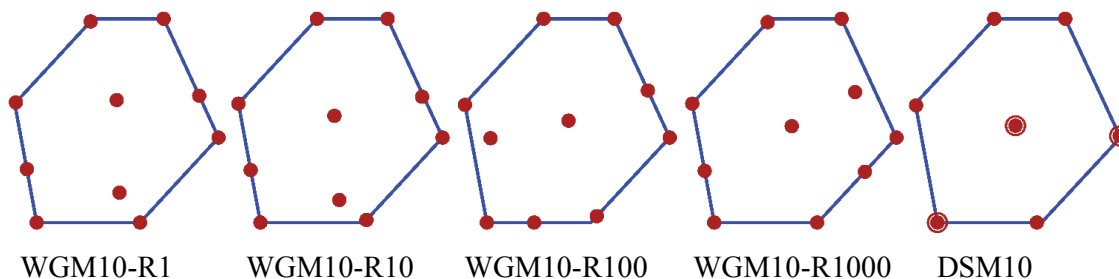


Fig. 3. The competing designs for Snee's experiment. ● and ● represent 1 and 2 replicates of that design point, respectively.

The approximate WG-efficiencies, approximate G-efficiencies and maximum SPV values for five competing designs are given in Table 4. The WG-efficiencies of the WGM10 designs are comparable except for the WG10-R1 design. For the DSM10 design, the WG-efficiencies values with $R = 10, 100$ and $1,000$ are comparable. The WGM10 designs are quite comparable based on the G-efficiencies and maximum SPV values for both the Scheffé quadratic and Scheffé linear models. The DSM10 designs perform poorly when examining the G-efficiencies and maximum SPV values for both the Scheffé quadratic and Scheffé linear models.

Table 4

The approximate WG-efficiency, approximate G-efficiency and Max SPV values for competing designs for Snee’s experiment.

Design	R	WG-efficiency	Model type	G-efficiency	Max SPV values
WGM10-R1	1	73.9436	Quadratic model	84.4189	7.1074
			Linear model	78.1509	3.8387
WGM10-R10	10	75.9750	Quadratic model	84.2884	7.1184
			Linear model	76.6696	3.9129
WGM10-R100	100	75.9929	Quadratic model	84.5098	7.0998
			Linear model	77.0661	3.8928
WGM10-R1000	1000	76.2369	Quadratic model	84.4424	7.1054
			Linear model	75.9242	3.9513
DNM10, DSM10	1	62.4809	Linear model	66.3833	9.0384
	10	61.4238			
	100	61.2145			
	1000	61.1915			

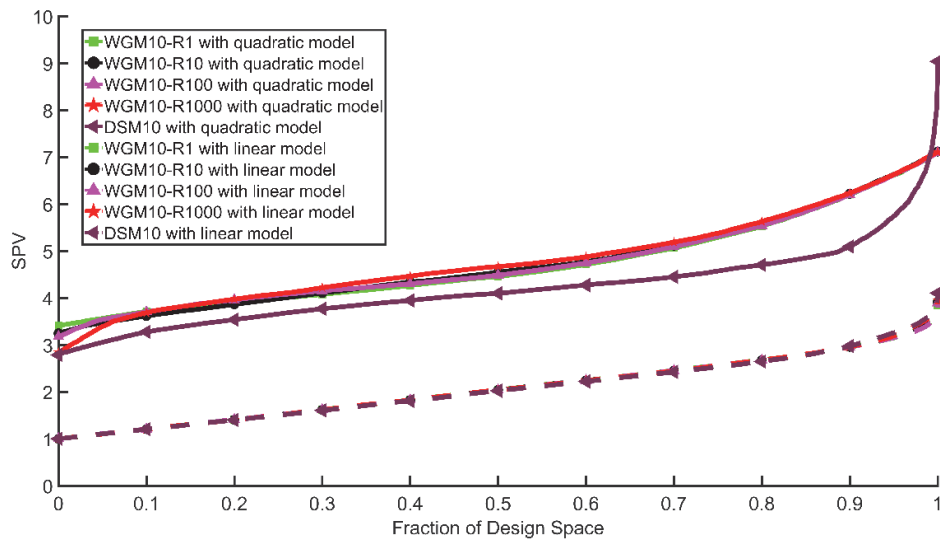


Fig. 4. FDS plot for comparing the WGM10 designs and the EA design for the Scheffé quadratic and linear models for Snee’s experiment.

Fig. 4 shows the comparison between five competing designs. For the Scheffé quadratic model, the DSM10 design performs poorly for a small fraction ($\approx 2\%$) of the design space. Therefore, if the experimenter decides to use the DSM10 design, there is a risk of having poor prediction variance properties in a potentially important region of the design space. For the Scheffé quadratic and Scheffé linear models, the WGM10 designs have similar SPV distribution for most of the design space. For the majority of the design space, the WGM10 designs appear to have good robustness properties because it has much flatter and lower FDS curves for both the Scheffé quadratic and linear models, indicating more desirable SPV distributions.

7. Conclusions

The goal of a genetic algorithm is to optimize an objective function, and its success is based on the survival of the fittest evolutionary principal. In our genetic algorithm, we develop the WG-efficiency as the objective function for evaluating the fitness of each chromosome (design). We focus on the best design based on minimax criteria with the goal of controlling the worst prediction variance. The goal of weighted G-optimality is to minimize the weighted average of the maximum SPV in the design region over a set of reduced models. Thus, our algorithm constructs model-robust mixture designs. The WG-optimality criterion requires all those involved in planning the experiment to consider a set of reduced models and assignment of weights based on the additional generalizations as well as characteristics of scientific knowledge.

Before performing an experiment, the experimenter must decide which type of design for fitting the approximating model. Frequency, the initial model may not turn out to be the final model, the experimenter should exercise caution when choosing a design. Therefore, the model robustness properties of a design are an important consideration. When the experimenter decides which mixture design is the best based on design optimality criteria, it is important that the optimality criteria should be determined over a set of reduced models.

The genetic algorithm presented in this paper generates designs by recruiting points from a continuous design space. The search in a continuous design space produces an upper bound for efficiencies that is larger than the upper bound when the search is in a discrete design space. Our algorithms present an interesting alternative for generating optimal designs. In this paper, we only considered mixture designs on an irregularly-shape polyhedron region within the simplex.

Based on the results, the WG-optimal exact designs appear to have model robustness because our designs start with small values of the SPV and remain relatively flat throughout the design space across a set of possible reduced models. The WG-optimal exact designs perform better than the EA designs in terms of the approximate WG-efficiencies, approximate G-efficiencies, maximum SPV values and the robustness properties. That is, the WG-efficiency criterion as the objective function is smooth and well behaved.

Genetic algorithms are not available in the most popular software packages for generating the optimal designs. The genetic algorithm presented in this paper can prove to be useful in situation where the experimenter finds that some model effects may be insignificant. That is, the experimenter will likely decide to use a reduced model in the analysis phases.

References

- Atkinson, A. C., Donev A. N., & Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS*. Oxford University Press Inc.
- Anderson-Cook, C. M., Borror, C. M., & Jones, B. (2009). Graphical tool for assessing the sensitivity of response designs to model misspecification. *Technometrics*, 51(1), 75-87.
- Borkowski, J. J. (2003). Using genetic algorithm to generate small exact response surface designs. *Journal of Probability and Statistical Science*, 1(1), 65-88.
- Borkowski, J. J., Chomtee, B., & Turk, P. (2010). Using Weak and Strong Heredity to Generate Weighted Design Optimality Criteria for Response Surface Designs. *Journal of Statistical Theory and Applications*, 10(2), 163-192.
- Borkowski, J. J., & Piepel, G. F. (2009). Uniform designs for highly constrained mixture experiments. *Journal of Quality Technology*, 41(1), .13-1
- Chipman, H. A. (1996) Bayesian variable selection with related predictors. *The Canadian Journal of Statistics*, 24(1), 17-36.

- Chung, P. J., Goldfarb, H., & Montgomery, D. C. (2007). Optimal designs for mixture-process experiments with control and noise variables. *Journal of Quality Technology*, 39(3), 179-190.
- Cornell, J. A. (2002) *Experiments with Mixtures : Designs, Models, and the Analysis of Mixture Data*. (3rd edition). John Wiley & Sons, Inc.
- Drain, D., Carlyle, W. M., Montgomery, D. C., Borror, C., & Anderson-Cook, C. (2004). A genetic algorithm hybrid for constructing optimal response surface designs. *Quality and Reliability Engineering International*, 20(7), 637-650.
- DuMouchel, W., & Jones, B. (1994). A simple Bayesian modification of D-optimal design to reduce dependence on an assumed model. *Technometrics*, 36(1), 37-47.
- Emmett, M., Goos, P., & Stillman, E. C. (2011). A weighted prediction-based selection criterion for response surface designs. *Quality and Reliability Engineering International*, 27(5), 719-729.
- Engelbrecht, A.P. (2002). *Computational Intelligence: An Introduction*. John Wiley & Sons, Inc.
- Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Goldfarb, H. B., Anderson-Cook, C. M., Borror, C. M., & Montgomery, D. C. (2004). Fraction of design space plots for assessing mixture and mixture-process designs. *Journal of Quality Technology*, 36(2), 169-179.
- Goldfarb, H. B., Borro, C. M., Montgomery, D. C., & Anderson-Cook, C. M. (2005). Using genetic algorithms to generate mixture-process experimental designs involving control and noise variables. *Journal of Quality Technology*, 37(1), 60-74.
- Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms*. John Wiley & Sons, Inc.
- Heredia-Langner, A., Carlyle, W. M., Montgomery, D. C., Borror, C. M., & Runger, G. C. (2003). Genetic algorithms for the construction of D-optimal designs. *Journal of Quality Technology*, 35(1), 28-46.
- Heredia-Langner, A., Montgomery, D. C., Carlyle, W. M., & Borror, C. M. (2004). Model-robust optimal designs: a genetic algorithm approach. *Journal of Quality Technology*, 36(3), 263-279.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. MIT Press.
- Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society, Series B*, 21(2), 272-304.
- Kiefer, J., & Wolfowitz, J. (1959). Optimum design in regression problems. *Annals of Mathematical Statistics*, 30(2), 271-294.
- Kiefer, J., & Wolfowitz, J. (1960). The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12(3), 363-366.
- Kinney, K. E. (1994). *Advances in Genetic Programming*. MIT Press.
- Li, W., & Nachtsheim, C. J. (2000). Model-robust factorial designs. *Technometrics*, 42(4), 345-352.
- Limmun, W., Borkowski, J. J., & Chomtee, B. (2013). Using a Genetic Algorithm to Generate D-optimal Designs for Mixture Experiments. *Quality and Reliability Engineering International*, 29(7), 1055-1068.
- The MathWorks. (2016). *Matlab high performance numeric computation software*. The MathWorks, Inc.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- Morgan, J. P., & Wang, X. (2010). Weighted Optimality in Designed Experimentation. *Journal of the American Statistical Association*, 105(492), 1566-1580.
- Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2009). *Response Surface Methodology : Process AND Product Optimization Using Designed Experiments*. (3rd edition). John Wiley & Sons, Inc.
- Ozol-Godfrey, A., Anderson-Cook, C.M., & Montgomery, D.C. (2005). Fraction of design space plots for examining model robustness. *Journal of Quality Technology*, 37(3), 223-235.
- Park, Y., Montgomery, D. C., Fowler, J. W., & Borror, C. M. (2006). Cost-constrained G-efficient response surface designs for cuboidal regions. *Quality and Reliability Engineering International*, 22(2), 121-139.

- Rodriguez, M., Montgomery, D.C., & Borror, C. M. (2009). Generating Experimental Designs Involving Control and Noise Variables using Genetic Algorithms. *Quality and Reliability Engineering International*, 25(8), 1045-1065.
- SAS Institute. (2012). *SAS QC 14.1 user's guide, Version 9*. SAS Institute Inc.
- Smith, W. F. (2005). *Experimental Design for Formulation*. The American Statistical Association and the Society for Industrial and Applied Mathematics.
- Smucker, B. J., Del Castillo, E., & Rosenberger, J. L. (2011). Exchange Algorithms for Constructing Model-Robust Experimental Designs. *Journal of Quality Technology*, 43(1), 1-15.
- Snee, R. D. (1979). Experimental design for mixture systems with multicomponent constraints. *Communications in Statistics-Theory and Methods*, 8(4), 303-326.
- Snee, R. D., & Rayner, A. A. (1982). Assessing the accuracy of mixture model regression calculations. *Journal of Quality Technology*, 14(2), 67-79.
- Stallings, J. W. & Morgan, J. P. (2015). General weighted optimality of designed experiments. *Biometrika*, 102(4), 925-935.
- Wheeler, R. E. (1972). Efficient experimental design. *Annual Meeting of the American Statistical Association*, Canada, August, 1972.
- Zaham, A., Anderson-Cook, C. M., & Myers, R. M. (2003). Fraction of design space to assess prediction capability of response surface designs. *Journal of Quality Technology*, 35(4), 377-386.

Appendix

The WG-optimal exact designs produced by our genetic algorithm are shown in this appendix.

Table A1

The WG10 designs for the poultry experiment.

x_1	x_2	x_3
Common runs for all designs		
0.3000	0.3000	0.4000
0.5000	0.0000	0.5000
0.7000	0.3000	0.0000
0.8000	0.0000	0.2000
Addition runs for the WG10-R1 design		
0.3000	0.2151	0.4849
0.4374	0.0626	0.5000
0.5139	0.2861	0.2000
0.5617	0.1700	0.2683
0.7906	0.2094	0.0000
0.8000	0.0614	0.1386
Addition runs for the WG10-R10 design		
0.3000	0.2000	0.5000
0.4779	0.0503	0.4718
0.4902	0.2933	0.2165
0.5722	0.1591	0.2687
0.7839	0.0560	0.1601
0.7894	0.2106	0.0000
Addition runs for the WG10-R100 design		
0.3000	0.2133	0.4867
0.4524	0.0587	0.4889
0.5215	0.2918	0.1867
0.5670	0.1600	0.2730
0.7765	0.0528	0.1707
0.8000	0.2000	0.0000
Addition runs for the WG10-R1000 design		
0.3000	0.2220	0.4780
0.4352	0.0648	0.5000
0.5395	0.2848	0.1757
0.5676	0.1584	0.2740
0.8000	0.0608	0.1392
0.8000	0.2000	0.0000

Table A2

The WGM10 designs for Snee's experiment.

x_1	x_2	x_3
Common runs for all designs		
0.1000	0.57	0.33
0.2667	0.1	0.6333
0.5000	0.1000	0.4000
Addition runs for the WGM10-R1 design		
0.1000	0.3500	0.5500
0.1585	0.3646	0.4769
0.2047	0.5384	0.2569
0.3358	0.4963	0.1679
0.3403	0.2797	0.3800
0.3488	0.1000	0.5512
0.4947	0.2579	0.2474
Addition runs for the WGM10-R10 design		
0.1049	0.3427	0.5524
0.1443	0.3803	0.4754
0.2030	0.5391	0.2579
0.3095	0.3082	0.3823
0.3333	0.5000	0.1667
0.3469	0.1034	0.5497
0.5000	0.2500	0.2500
Addition runs for the WGM10-R100 design		
0.1000	0.4700	0.4300
0.1125	0.3313	0.5562
0.2655	0.4795	0.2550
0.3000	0.2972	0.4028
0.3309	0.5007	0.1684
0.3591	0.1000	0.5409
0.5000	0.2500	0.2500
Addition runs for the WGM10-R1000 design		
0.1000	0.3516	0.5484
0.1996	0.2006	0.5998
0.2014	0.5395	0.2591
0.2892	0.3110	0.3998
0.3333	0.5000	0.1667
0.3559	0.1431	0.5010
0.4933	0.2600	0.2467



© 2019 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).