Available online at www.isas.org.in/jisas

JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 69(2) 2015 169-182

An Alternative to Extreme Vertices Designs for Constrained Mixture Experiments: QP-Procrustated Designs

Ravindra Khattree
Department of Mathematics and Statistics
Oakland University, Rochester, Michigan, 48309-4485

Received 06 January 2015; Revised 14 May 2015; Accepted 22 June 2015

SUMMARY

We provide an approach alternative to extreme vertices designs when there are constraints on the components of the mixture. The approach is based on the idea that given a suitably chosen base design for the unconstrained simplex, one can find another design within the constrained feasible region, which is closest to the base design in some meaningful sense. The problem is formulated as a quadratic program. These designs are able to avoid certain undesirable features of extreme vertices designs such as exclusive reliance on constraints to identify the design, not knowing a priori the number of vertices and the number of centroids and therefore the overall size of the design and in case, a fixed design size is already specified, issues faced pertaining to which of the centroids should be included in the final design. Several examples illustrating these issues and corresponding solutions are presented.

Keywords: Bounds, Extreme vertices designs, Interior mixture designs, Mixture experiments, Procrustation, Quadratic programming.

1. INTRODUCTION

Mixture experiments are now-a-days common place in chemical, pharmaceutical, food and various other industries. However, much of the literature on mixture experiments essentially deals with designs where most of the design points are located either at the vertices or on the edges of the simplex defined by $x_1 + x_2 + ... + x_q = 1$ where x_i represents the proportion of ith ingredient in the mixture. Clearly at the vertices or the edges of the simplex one or more of x_i is necessarily zero, thereby resulting in an *incomplete* mixture. An exception to the previously mentioned literature is the recent welcome addition of the book by Sinha *et al*. (2014), which largely deals with the mixture designs in the interior of the simplex thereby dealing with the case of *complete* mixtures. When there are linear constraints on the components of a mixture, the design points for any suitable design must be in the interior of the simplex and in this case, for most practical situations the extreme vertices designs suggested by McLean and Anderson (1966) have been the widely accepted choice for mixture experiments. The essential idea, in fact quite an innovative one for its time, was to identify the vertices of the constrained region and take these as the first set of design points; use these as the building block to construct more design points if there is a need by computing the centroids of various faces and edges of the constrained region; identify the over-all centroid of this region and then finally fill in the design with the extra experiments corresponding to these design points. Box and Draper (2007) extensively discuss these designs and point out various advantages and disadvantages of using the

E-mail address: khattree@oakland.edu

extreme vertices designs. The availability of an algorithm to identify all extreme vertices, uniqueness of the set of vertices once the constraints are specified, and their flexibility to a sequential approach to construction are among the stated advantages of using these designs. They do however also point out certain disadvantages. Specifically, as the number of components and the number of constraints increase, so do the number of extreme vertices and the number of various possible centroids. Designs points can cluster together thereby resulting in a poor design. Further, the design is completely determined by various bounds and the locations where these bounds are set, essentially decide the locations of centroids and hence the final design.

It is the last two disadvantages that we deem most serious. From practical point of view, it is well known that the upper and lower bounds are seldom very well known or established. Such bounds are usually chosen by using intuition and one or more of these bounds may not be realistic. Bounds can be viewed as the worst possible scenarios beyond which we have the unacceptable region where experiments should not be performed. By definition, extreme vertices are the points where (q-1) bounds on components hold as equality. Accordingly, the vertices can be interpreted as the combinations of several worst possible scenarios. Not only these but also the centroids using these worst possible scenarios are augmented to the design and thus determine the entire design. Therefore, we argue that a prediction equation arrived at by doing experiments at these worst possible scenarios may not hold in the interior, in which case, it is safe to say, that the corresponding regression function will often fail to predict satisfactorily and consequently will fail to identify the optimum mixture. The problem is further compounded when the vertices and the intermediate centroids are too close to each other resulting in the poor quality of the design especially when we do not have an option to run a large number of design points.

We present Tables 1, 2 and 3 as illustrations of another undesirable feature of the extreme vertices designs. First of all, the number of vertices obtained depends on the particular bounds. Table 1 shows the number of vertices for three-component mixtures (q = 3) obtained under various bounds on the components and considered by various authors in several classical examples illustrated in the literature. It is clear that the number of vertices obtained range from as few as 3 to as many as 6. The number of centroids added to design range from 4 to 7. For q = 4, the number of vertices is as small as 4 and as large as 10. See Table 2. In this case, number of centroid points is from 11 to 23. For q = 5, as Table 3 indicates, two

Table 1. Extreme vertices for certain three-component (q = 3) mixtures

Constraints	Design Point	x_1	x_2	x_3	No. of Centroids
Box and Draper (2007, p 551) Lower Bounds Only $0.2 \le x_1 \le 1$, $0.2 \le x_2 \le 1$, $0.3 \le x_3 \le 1$	1	0.2000	0.2000	0.6000	4
	2	0.2000	0.5000	0.3000	
	3	0.5000	0.2000	0.3000	
Khuri and Cornell (1996, p.382) $0.2 \le x_1 \le 0.5$, $0.05 \le x_2 \le 0.65$, $0.15 \le x_3 \le 0.75$	1	0.5000	0.0500	0.4500	5
	2	0.5000	0.3500	0.1500	
	3	0.2000	0.6500	0.1500	
	4	0.2000	0.0500	0.7500	
Juan *et al.* (2006), Lawson (2010) $0 \le x_1 \le 0.8$, $0.1 \le x_2 \le 0.95$, $0.05 \le x_3 \le 0.5$	1	0.8000	0.1000	0.1000	6
	2	0.8000	0.1500	0.0500	
	3	0.0000	0.9500	0.0500	
	4	0.0000	0.5000	0.5000	
	5	0.4000	0.1000	0.5000	
[1]Anik and Sukumar (1981) with x_4 = 0.0 and x_5 = 0.1 $0.1111 \le x_1^* \le 0.4444$, $0.1111 \le x_2^* \le 0.4444$, $0.3333 \le x_3^* \le 0.7778$	1	0.4444	0.1111	0.4445	6
	2	0.4444	0.2223	0.3333	
	3	0.1111	0.4444	0.4445	
	4	0.1111	0.1111	0.7778	
	5	0.2223	0.4444	0.3333	
Anik and Sukumar (1981) with x_4 = 0.08 and x_5 = 0.1 $0.1220 \le x_1^* \le 0.4878$, $0.1220 \le x_2^* \le 0.4878$, $0.3656 \le x_3^* \le 0.8537$	1	0.4878	0.1220	0.3902	6
	2	0.4878	0.1466	0.3656	
	3	0.1220	0.4878	0.3902	
	4	0.1220	0.1220	0.7560	
	5	0.1466	0.4878	0.3656	
Khuri and Cornell (1996, p.351) $0.2 \le x_1 \le 0.6$, $0.1 \le x_2 \le 0.6$, $0.1 \le x_3 \le 0.5$	1	0.6000	0.1000	0.3000	7
	2	0.6000	0.3000	0.1000	
	3	0.2000	0.6000	0.2000	
	4	0.2000	0.3000	0.5000	
	5	0.3000	0.6000	0.1000	
	6	0.4000	0.1000	0.5000	
Upper Bounds Only $0 \le x_1 \le 0.7$, $0 \le x_2 \le 0.6$, $0 \le x_3 \le 0.5$	1	0.7000	0.0000	0.3000	7
	2	0.7000	0.3000	0.0000	
	3	0.0000	0.6000	0.4000	
	4	0.0000	0.5000	0.5000	
	5	0.4000	0.6000	0.0000	
	6	0.5000	0.0000	0.5000	

[1]x_i^* are pseudo components in the sense that with x_4 = 0.0 and x_5 = 0.10, $x_1 + x_2 + x_3$ = 0.90, therefore x_i^*, i = 1, 2, 3, are scaled by dividing each x_i by 0.90, so that $x_1^* + x_2^* + x_3^*$ = 1. Similar adjustments are made in the case that immediately follows after this case and also in the Anik and Sukumar case in Table 2.

Table 2. Extreme vertices for certain four-component (q = 4) mixtures

Constraints	Design Point	x_1	x_2	x_3	x_4	No. of Centroids
Barbuta and Lepadatu (2008) $0.124 \le x_1 \le 0.188$, $0.064 \le x_2 \le 0.128$, $0.374 \le x_3 \le 0.438$, $0.374 \le x_4 \le 0.438$	1	0.1880	0.0640	0.3740	0.3740	11
	2	0.1240	0.1280	0.3740	0.3740	
	3	0.1240	0.0640	0.4380	0.3740	
	4	0.1240	0.0640	0.3740	0.4380	
Gorman (1966), Box and Draper (2007, p. 551) $0 \le x_1 \le 0.24$, $0.25 \le x_2 \le 0.75$, $0.25 \le x_3 \le 0.75$, $0.25 \le x_4 \le 0.75$	1	0.2400	0.2500	0.2500	0.2600	15
	2	0.2400	0.2500	0.2600	0.2500	
	3	0.2400	0.2600	0.2500	0.2500	
	4	0.0000	0.2500	0.2500	0.5000	
	5	0.0000	0.2500	0.5000	0.2500	
	6	0.0000	0.5000	0.2500	0.2500	
McLean and Anderson (1966) $0.4 \le x_1 \le 0.6$, $0.1 \le x_2 \le 0.5$, $0.1 \le x_3 \le 0.5$, $0.03 \le x_4 \le 0.08$	1	0.6000	0.1000	0.2200	0.0800	19
	2	0.6000	0.1000	0.2700	0.0300	
	3	0.6000	0.2200	0.1000	0.0800	
	4	0.6000	0.2700	0.1000	0.0300	
	5	0.4000	0.1000	0.4200	0.0800	
	6	0.4000	0.1000	0.4700	0.0300	
	7	0.4000	0.4200	0.1000	0.0800	
	8	0.4000	0.4700	0.1000	0.0300	
Khuri and Cornell (1996, p.355) $0.89 \le x_1 \le 0.905$, $0.02 \le x_2 \le 0.035$, $0.04 \le x_3 \le 0.065$, $0.01 \le x_4 \le 0.02$	1	0.9050	0.0350	0.0400	0.0200	21
	2	0.9050	0.0350	0.0500	0.0100	
	3	0.9050	0.0200	0.0650	0.0100	
	4	0.9050	0.0200	0.0550	0.0200	
	5	0.8900	0.0350	0.0650	0.0100	
	6	0.8900	0.0350	0.0550	0.0200	
	7	0.8900	0.0250	0.0650	0.0200	
	8	0.8950	0.0200	0.0650	0.0200	
Anik and Sukumar (1981) Design in Pseudo Components with $x_5 = 0.10$ $0.1111 \le x_1^* \le 0.4444$, $0.1111 \le x_2^* \le 0.4444$, $0 \le x_3^* \le 0.0889$, $0.3333 \le x_4^* \le 0.7778$	1	0.4444	0.1111	0.0889	0.3556	23
	2	0.4444	0.1111	0.0000	0.4445	
	3	0.4444	0.1334	0.0889	0.3333	
	4	0.4444	0.2223	0.0000	0.3333	
	5	0.1111	0.4444	0.0889	0.3556	
	6	0.1111	0.4444	0.0000	0.4445	
	7	0.1111	0.1111	0.0889	0.6889	
	8	0.1111	0.1111	0.0000	0.7778	
	9	0.1334	0.4444	0.0889	0.3333	
	10	0.2223	0.4444	0.0000	0.3333	

situations are considered. For the Anik-Sukumar problem (Anik and Sukumar 1981), the number of vertices is 10 while the same for McLean and Anderson (1974, p 347) example is 20! In the first case, we have 69 additional points as centroids while in the second case, number of centroid is 79.The SAS®code given in Appendix A is used to generate these vertices and centroids and hence the design points. With such a large number of design points available, all of which cannot be used for experimentation, we are faced with the problem of how to (and how many of these to) choose our final design points. We hasten to add that the important fact to be further re-emphasized is that, the choice of bounds is somewhat arbitrary and the resulting design overwhelmingly depends on this choice. Thus, the practical alternatives such as, finding an optimal set of design points out of this long list under some optimality criterion is not likely to be very effective as long as prediction is concerned.

Table 3. Extreme vertices for certain five-component (q = 5) mixtures

Constraints	Design Point	x_1	x_2	x_3	x_4	x_5	No. of Centroids
Anik and Sukumar (1981) with q = 5 but $x_5 = 0.1$ $0.1 \le x_1 \le 0.4$, $0.1 \le x_2 \le 0.4$, $0 \le x_3 \le 0.08$, $0.3 \le x_4 \le 0.7$ $0.1 \le x_5 \le 0.1$	1	0.4000	0.1000	0.0800	0.3200	0.1000	69
	2	0.4000	0.1000	0.0000	0.4000	0.1000	
	3	0.4000	0.1200	0.0800	0.3000	0.1000	
	4	0.4000	0.2000	0.0000	0.3000	0.1000	
	5	0.1000	0.4000	0.0800	0.3200	0.1000	
	6	0.1000	0.4000	0.0000	0.4000	0.1000	
	7	0.1000	0.1000	0.0800	0.6200	0.1000	
	8	0.1000	0.1000	0.0000	0.7000	0.1000	
	9	0.1200	0.4000	0.0800	0.3000	0.1000	
	10	0.2000	0.4000	0.0000	0.3000	0.1000	
Anderson and McLean (1974, p. 347) $0.0004 \le x_1 \le 0.0010$, $0.08 \le x_2 \le 0.12$, $0.12 \le x_3 \le 0.20$, $0.005 \le x_4 \le 0.02$ $0.65 \le x_5 \le 0.75$	1	0.0010	0.1200	0.2000	0.0200	0.6590	79
	2	0.0010	0.1200	0.2000	0.0050	0.6740	
	3	0.0010	0.1200	0.1200	0.0200	0.7390	
	4	0.0010	0.1200	0.1200	0.0090	0.7500	
	5	0.0010	0.1200	0.1240	0.0050	0.7500	
	6	0.0010	0.0800	0.2000	0.0200	0.6990	
	7	0.0010	0.0800	0.2000	0.0050	0.7140	
	8	0.0010	0.0800	0.1490	0.0200	0.7500	
	9	0.0010	0.0800	0.1640	0.0050	0.7500	
	10	0.0010	0.1090	0.1200	0.0200	0.7500	
	11	0.0004	0.1200	0.2000	0.0200	0.6596	
	12	0.0004	0.1200	0.2000	0.0050	0.6746	
	13	0.0004	0.1200	0.1200	0.0200	0.7396	
	14	0.0004	0.1200	0.1200	0.0096	0.7500	
	15	0.0004	0.1200	0.1246	0.0050	0.7500	
	16	0.0004	0.0800	0.2000	0.0200	0.6996	
	17	0.0004	0.0800	0.2000	0.0050	0.7146	
	18	0.0004	0.0800	0.1496	0.0200	0.7500	
	19	0.0004	0.0800	0.1646	0.0050	0.7500	
	20	0.0004	0.1096	0.1200	0.0200	0.7500	

Consideration of above concerns essentially defines the objective of this work and thus our alternative approach given here is to the contrary. Our objective is to have a design which is dependent on the bounds but not as much as the extreme vertices designs. At the same time, we also believe that the design for the constrained region should be extracted from some suitably predetermined well established and well researched design for the unconstrained region. We also desire that the size of the design in the constrained region is determined apriori and should be comparable to that for the unconstrained region. The purpose of this note is to provide such designs. We will call such designs Quadratic Program Procrustated (QP-Procrustated) mixture designs.

2. DESIGNS THROUGH QP-PROCRUSTATION[2]

We introduce Quadratic Program- or QP-Procrustated designs as an alternative to extreme vertices designs. The basic idea is what we term as *procrustation*. Specifically, let us suppose that a well spread base design $\mathcal{D} = \{z_1, z_2, \ldots, z_t\}$ with design points $z_1, z_2, \ldots, z_t$ is given for the unconstrained region $\mathcal{U}$ which is a simplex. In practical applications, $\mathcal{D}$ should adequately cover the entire simplex. Let $\mathcal{C}$ be the constrained region possibly defined by the upper and lower bounds on various components of the mixture. For a given design point z_i in $\mathcal{U}$, we obtain a point $\boldsymbol{x}_i$ in $\mathcal{C}$ so that $\boldsymbol{x}_i$ is the closest point in $\mathcal{C}$ to z_i. Specifically, (dropping the subscript i) for each design point z in $\mathcal{U}$, we will find a point $\boldsymbol{x} = \boldsymbol{x}^*$ in $\mathcal{C}$, which is a solution to the optimization problem:

Minimize Dist$(\boldsymbol{x}, z)$

Subject to the condition that $\boldsymbol{x} \in \mathcal{C}$,

where *Dist*$(\boldsymbol{x}, z)$ is an appropriately chosen distance function.

Why should one attempt to find a design which is close to the base design? The rationale for doing this lies in the fact (and assumption) that our base design is well spread, thereby, theoretically more suitable for prediction, which is the main goal of mixture modelling. Thus it is anticipated that a design close to it will also be suitable for prediction purposes. It must be added that strictly speaking the base design does not have to be a design with incomplete mixtures. For instance, see Example 2 that follows.

Returning to the above optimization problem to minimize *Dist*$(\boldsymbol{x}, z)$, as the (squared) Euclidean distance and with the region $\mathcal{C}$ defined by lower and upper bounds on mixture components, the optimization problem stated above becomes a quadratic programming problem. Specifically, let $z = (z_1, z_2, \ldots, z_q)'$ be a design point in $\mathcal{U}$ and let $\boldsymbol{x} = (x_1, x_2, \ldots, x_q)'$ be a point to be determined, satisfying the bounds defining the region $\mathcal{C}$. Then the previous problem can be written as,

$$\textit{Minimize } (z-\boldsymbol{x})'(z-\boldsymbol{x}) = \{(z_1-x_1)^2 + (z_2-x_2)^2 + \ldots + (z_q-x_q)^2\}$$

$$\text{Subject to } x_1 + x_2 + \ldots + x_q = 1$$

$$a_j \le x_j \le b_j,\ j = 1, 2, \ldots q.$$

In more standard matrix notations and by using a scaling constant of ½, we can equivalently state the above problem in a more familiar format as,

$$\textit{Minimize } \tfrac{1}{2}\, \boldsymbol{x}'\boldsymbol{x} - z'\boldsymbol{x}$$

$$\textit{Subject to } \mathbf{1}'\boldsymbol{x} = 1,$$

$$a_j \le x_j \le b_j,\ j = 1, 2, \ldots q,$$

where, $\mathbf{1}$ represents a $q \times 1$ vector of unit elements.

Since all the constraints are linear, the above problem can be solved without much difficulty using the standard software (*e.g.* SAS®) available. This is especially so since the dimension of the problem is same as the number of mixture components q, which is usually small (< 7). However, the quadratic programming problem must be repeatedly solved for

[2]In Greek mythology, Procrustes was a Greek innkeeper with a very unique idea and approach about the standardization and customer service. In his inn, he had the fixed size cots. If a customer was too short for the cot, his or her legs were stretched to fit the cot. On the other hand, if he or she was too tall, the legs were chopped off to confine to the cot. Our designs do the same. If an original design is too enlarged to fit in within the feasible region with design points falling off the feasible region, the design needs to be appropriately resized. Thus, we term such a process of resizing as Procrustation. In keeping with the Procrustes' approach that only the smallest lengths of the legs must be cut (otherwise, they would need to be stretched again), we also rely on shortest distance. In multidimensional analysis Procrustes Analysis uses the same idea of resizing albeit in a very different context and using the different techniques.

each design point[3] $\boldsymbol{x}$ in $\mathcal{C}$. An illustrative SAS® code is given in Appendix B for the example considered by Khuri and Cornell (1996) and for $q = 3$. For this example, the base as well as QP-procrustated designs are both illustrated in Fig. 1.

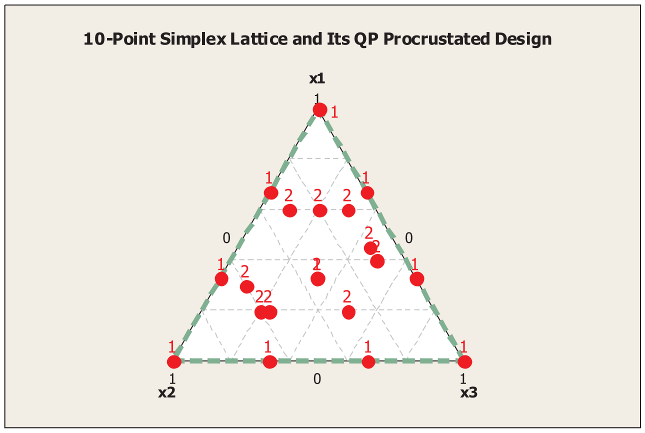

Fig. 1. 10-point Simplex Lattice Design and its QP-Procrustated Version (1 = Simplex Lattice Design Point, 2 = Procrustated Design Point)

Naturally, the quality of the new design obtained depends on the quality of the base design. Thus, the base design plays a crucial role and must be chosen with some thought and care. If the base design covers the entire simplex comprehensively and if we have chosen the closest design points in the constrained feasible region, then it is reasonable to hope that this new set of design points will also cover the constrained region comprehensively. Let us consider three examples including the one mentioned earlier which we consider first to illustrate above mentioned points and to make comparisons with some of the existing designs used in the literature.

Example 1. Khuri and Cornell (1996) considered the problem of finding the extreme vertices for a three-component mixture design, under the constraints,

$$0.20 \leq x_1 \leq 0.60,\ 0.10 \leq x_2 \leq 0.60,\ 0.10 \leq x_3 \leq 0.50.$$

The six extreme vertices obtained are listed in Columns 11-13 of Table 4, which are also available in Table 1. Also listed are seven centroid points. Fig. 2a shows the simplex representation of these vertices and design points. Instead of obtaining these 13 design points, our approach is to work with a given and suitably chosen standard design. Let us separately

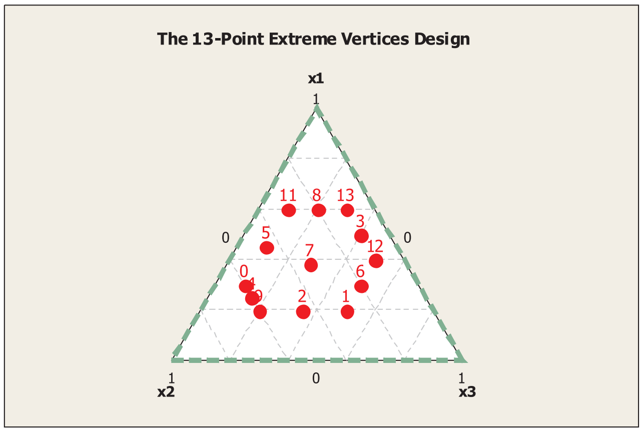

Fig. 2a. Design points for 13-point extreme vertices design

consider the two designs advocated by Cornell (1986) viz., the ten-point simplex lattice and the ten-point simplex centroid designs. As Cornell clearly states, these designs cover the simplex very well. Also see Myers *et al.* (2009). The simplex lattice design consists of one interior point and nine points on vertices or edges. The simplex centroid design has four interior points and six points on the vertices and edges.

However, in both cases, only one point, namely the overall centroid of the simplex, is within the feasible region defined by the constraints. We try to find a ten-point design in the feasible regions defined by above constraints on mixture components so that in each case, for every point in the base design, a point closest to it and in the feasible region is obtained. Clearly, in terms of economy of resources, these new ten (or less) point designs are more efficient than the 13 point extreme vertices design, especially when enough degrees of freedom are still available to fit a second order Scheffe's model and perform a lack of fit test. By taking the ten (z_1, z_2, z_3) values corresponding to each design point in the base design, and by solving the corresponding quadratic programs, we obtain, the ten points (x_1, x_2, x_3) as listed in Table 4.

In case of procrustation of the 10-point simplex lattice design, five of the six extreme vertices are part of the QP-procrustated design. Further, one of the centroid of the extreme vertices design is also included. Of the remaining four design points, one is the centroid of the simplex which, by default of falling within the feasible region, is automatically included. The remaining three points are projected at the three points

[3]Since all these problems involve different design points, the corresponding objective functions are all separable. Therefore, one can, in fact, lump all of these smaller-dimensional quadratic programs into one big quadratic program with an objective function equal to sum of all objective functions of smaller dimensional problems. However, the number of constraints will increase. But that poses no problem in terms of formulation. For the sake of simplicity and understanding, we prefer the separate optimization for each individual design point.

Table 4. Design points for QP-Procrustated simplex designs and extreme vertices design of Khuri and Cornell (1996, p. 351)

Design Point[4*]	Original Design			Procrustation of						Extreme Vertices Designs ('V' indicates an extreme vertex)			
				Simplex Lattice Design			Simplex Centroid Design						
	z_1	z_2	z_3	x_1	x_2	x_3	x_1	x_2	x_3	x_1	x_2	x_3	
1										0.20000	0.30000	0.50000	V
2										0.20000	0.45000	0.35000	
3										0.50000	0.10000	0.40000	
4										0.25000	0.60000	0.15000	
5										0.45000	0.45000	0.10000	
6										0.30000	0.20000	0.50000	
7										0.38333	0.33334	0.28333	
8	1.00	0.000	0.00	0.60	0.20	0.20	0.60	0.20	0.20	0.60000	0.20000	0.20000	
9	0.00	1.000	0.00	0.20	0.60	0.20	0.20	0.60	0.20	0.20000	0.60000	0.20000	V
10	0.00	0.000	1.00	0.45	0.10	0.45	0.45	0.10	0.45				
11	0.67	0.330	0.00	0.30	0.60	0.10				0.30000	0.60000	0.10000	V
12	0.33	0.670	0.00	0.60	0.30	0.10				0.60000	0.30000	0.10000	V
13	0.33	0.000	0.67	0.40	0.10	0.50				0.40000	0.10000	0.50000	V
14	0.67	0.000	0.33	0.60	0.10	0.30				0.60000	0.10000	0.30000	V
15	0.00	0.330	0.67	0.20	0.30	0.50							
16	0.00	0.670	0.33	0.20	0.57	0.23							
17	0.33	0.340	0.33	0.33	0.34	0.33	0.33	0.34	0.33				
18	0.50	0.500	0.00				0.45	0.45	0.10				
19	0.50	0.000	0.50				0.45	0.10	0.45				
20	0.00	0.500	0.50				0.20	0.40	0.40				
21	0.66	0.170	0.17				0.60	0.20	0.20				
22	0.17	0.660	0.17				0.20	0.60	0.20				
23	0.17	0.170	0.66				0.45	0.45	0.10				

[4]The design points 8-17 corresponds to the procrustation of the 10-point simplex lattice design for the entire simplex. Design points 8-10 and 17-23 corresponds to the procrustation of 10-point simplex centroid design for the entire simplex.

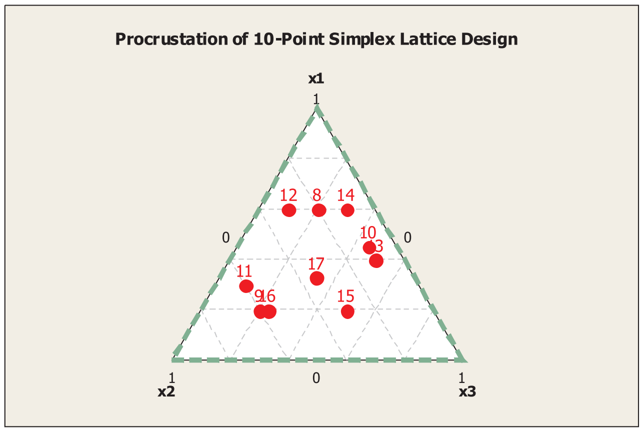

Fig. 2b. Design points for procrustated 10-point simplex lattice design

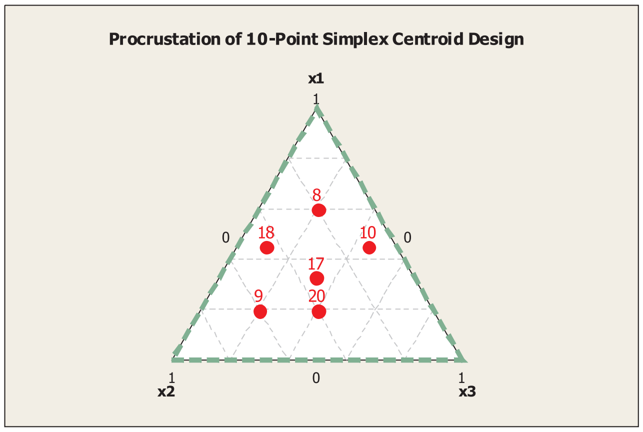

Fig. 2c. Design points for procrustated 10-point simplex centroid design

Table 5. QP-Procrustation of Parshvanath Design

Design Point	12-Point Parshvanath Design				QP-Procrustated Design of Parshvanath Design Constraints: $0.10 \le x_i \le 0.45$, for $i = 1, \ldots, 4$			
	z_1	z_2	z_3	z_4	x_1	x_2	x_3	x_4
1	0.20588	0.35294	0.02941	0.41176	0.18235	0.32942	0.10000	0.38823
2	0.05882	0.38235	0.23529	0.32353	0.10000	0.36863	0.22157	0.30980
3	0.47059	0.08824	0.29412	0.14706	0.45000	0.10000	0.29853	0.15147
4	0.26471	0.17647	0.44118	0.11765	0.26471	0.17647	0.44117	0.11765
5	0.20588	0.05882	0.47059	0.26471	0.19559	0.10000	0.44999	0.25442
6	0.35294	0.38235	0.08824	0.17647	0.34902	0.37843	0.10000	0.17255
7	0.02941	0.23529	0.29412	0.44118	0.10000	0.21176	0.27059	0.41765
8	0.41176	0.32353	0.14706	0.11765	0.41176	0.32353	0.14706	0.11765
9	0.20588	0.38235	0.29412	0.11765	0.20588	0.38235	0.29412	0.11765
10	0.26471	0.08824	0.23529	0.41176	0.26079	0.10000	0.23137	0.40784
11	0.11765	0.29412	0.38235	0.20588	0.11765	0.29412	0.38235	0.20588
12	0.41176	0.23529	0.08824	0.26471	0.40784	0.23137	0.10000	0.26079

in the feasible region located on the edges where only one constraint is active. This design with fewer runs appears to be as comprehensive as the corresponding 13-point extreme vertices design. See Fig. 2b.

Procrustation of simplex centroid design is however much different. The design has only two points which are common with the extreme vertices design and only one of these corresponds to a vertex. However, it so happens that certain points of the simplex centroid design map to the same point as a result of their respective procrustations (See the design points (8, 21),(9, 22), (10, 19) and (18, 23) in Table 4). The overall centroid is in the feasible region. The remaining three points are on the edges where only one constraint is active. Clearly, with four redundancies indicated above, this is a 6-point design. See Fig. 2c. In this case, for a second order model, no degree of freedom is available for error.

Example 2. To further illustrate, we consider the case of $q = 4$ and this time we consider the feasible region defined by the constraints, $0.10 \le x_i \le 0.45$, for $i = 1, \ldots, 4$. Further, we will consider a base design which does not emphasize vertices or edges and which resides in the interior of the simplex. Specifically, we will take the *Parshvanath* design recently introduced by Khattree (2014a, 2014b) with 12 design points, which are shown as Columns 2-5 in Table 5[5]. Note that four design points, namely 4, 8, 9 and 11 are already in the feasible region and most of the other points are barely outside it. In fact, all these points violate exactly one of the four constraints. We find the design points in the feasible region by Quadratic Program-procrustation and all these are listed in Columns 6-9 of Table 5.

Few observations must be made. In case of all design points, and for all components which do not violate the corresponding constraint, the changes in the mixture components are very moderate and the component violating the corresponding constraint is set at the closest bound. Thus, the essential features of the original design have largely remained intact. This is desirable especially in situations when for economic, efficiency or practicality reasons the levels of various mixture components and hence a design has been chosen and must be modified to accommodate the constraints. Extreme vertices approach does not allow any such possibility. Quadratic program-procrustation can readily provide a suitable design to serve the purpose. In contrast, the extreme vertices design for these constraints consists of 12 vertices which are all at the considerable distances from the design points specified by the base Parshvanath design. Further, these vertices result in 27 additional centroid points.

Example 3. What if some of the constraints are very tight? This will naturally results in a very small feasible region. When q is large, this can especially be a problem if one opts for an extreme vertices design. Many of these vertices may be too close to each other and centroids defined by them may be even closer and practically indistinguishable for experimentation. As Box and Draper (2007) point out, one may need to average some of these vertices to obtain an 'average vertex for the region'. Anik and Sukumar (1981) attempt to deal with this issue in the context of an application and as the first step, attempt to circumvent the problem by averaging and then carefully choosing certain centroids defined by these "average vertices". Their 14 point design is reproduced in Table 6 (Columns 2-5). Also see Fig. 3a for the distribution of points in certain projections. The corresponding constraints are,

$$0.1 \le x_1 \le 0.4,\ 0.1 \le x_2 \le 0.4,$$
$$0 \le x_3 \le 0.08,\ 0.3 \le x_4 \le 0.7.$$

As our alternative design, we will consider a 15 point simplex centroid design for the entire simplex as the base design and obtain its QP-procrustation under the same constraints. The design is listed in Table 6 (Columns 6-9). For the full second order Scheffe's model,

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{14} x_1 x_4 + \beta_{23} x_2 x_3 + \beta_{24} x_2 x_4 + \beta_{34} x_3 x_4 + \varepsilon.$$

the $(\boldsymbol{X'X})$ matrix corresponding to extreme vertices design turns out to be singular and hence the full model cannot be implemented. Consequently, in their paper, Anik and Sukumar (1981) chose to drop the x_2x_4 term to avoid this singularity and hence fit the smaller model,

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{14} x_1 x_4 + \beta_{23} x_2 x_3 + \beta_{34} x_3 x_4 + \varepsilon.$$

This singularity occurs despite the authors' judicious selection of points, some of the design points are still too close to each other. However, for our 15 points procrustated design, the $(\boldsymbol{X'X})$ matrix remains nonsingular and the full second order Scheffe's model can be fitted without any difficulty. This observation clearly indicates the shortcomings of extreme vertices designs and makes a point for the need of an alternative approach and an alternative class of designs for the constrained feasible region.

The distribution of 15 design points for our design and for certain projections is shown in Fig. 3b. A

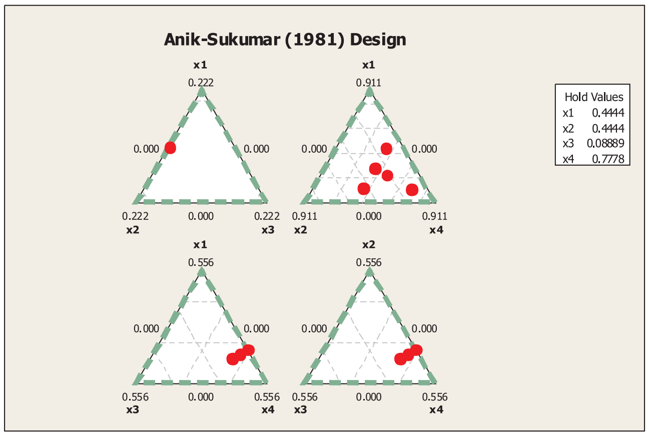

Fig. 3a. Design points for Anik-Sukumar (1981) design*

*Values listed in the 'Hold Values' box are the scaled version of the original levels, namely 0.4, 0.4, 0.08 and 0.7, by dividing each by 0.9.

[5]These designs are so named since these are derived from a Yantram inscribed on a wall of Parshvanath Jain Temple in Khajuraho, Madhya Pradesh, India.

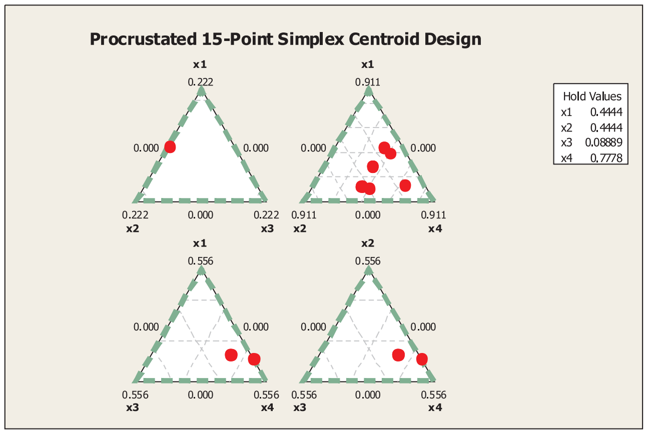

Fig. 3b. Design points for Procrustated 15-point Simplex Centroid design for Anik-Sukumar (1981) constraints*

*Values listed in the 'Hold Values' box are the scaled version of the original levels, namely 0.4, 0.4, 0.08 and 0.7, by dividing each by 0.9.

comparison with Fig. 3a, clearly shows that the distribution in Fig. 3b is considerably superior. While there may be more hidden points, the clustering of multiple design points in Frames 3 and 4 of Fig. 3a is clearly evident for the Anik-Sukumar design. Fig. 3b in that respect shows considerably better distribution of design points within the feasible region.

3. CONCLUDING REMARKS

The QP-procrustated designs provide an alternative to extreme vertices designs for mixture experiments. Given a base design such as simplex lattice, simplex centroid or any other suitably chosen non-interior or interior design (See, Sinha *et al.* 2014), QP-procrustation can be achieved to satisfy the constraints on the components of the mixture. Extreme vertices designs require one to append extra design points chosen as centroids of various faces and edges. As evident from Tables 2 and 3, the number of centroids can be excessive and then there is considerable subjectivity and ambiguity as to which and how many of these centroids should be included in the design. All these issues are avoided by using the QP-procrustated designs. The SAS® code given in Appendix B can be readily adopted for any number of constraints and for any $q > 1$. It may also be remarked that in general, these constraints only need to be linear in mixture components and do not have to be individually placed on each component.

Extreme vertices designs are exclusively and completely determined by the constraints. Changing any

Table 6. Anik-Sukumar design vs. QP-procrustation of Simplex Centroid design

Design Point	Anik- Sukumar (1981) Design				QP- Procrustation of 15-Point Simplex Centroid Design			
	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4
1	0.100	0.100	0.000	0.700	0.400	0.120	0.080	0.300
2	0.100	0.100	0.080	0.620	0.120	0.400	0.080	0.300
3	0.150	0.400	0.000	0.350	0.260	0.260	0.080	0.300
4	0.110	0.400	0.080	0.310	0.100	0.100	0.000	0.700
5	0.400	0.150	0.000	0.350	0.300	0.300	0.000	0.300
6	0.400	0.110	0.080	0.310	0.400	0.120	0.080	0.300
7	0.100	0.100	0.040	0.660	0.400	0.100	0.000	0.400
8	0.400	0.130	0.040	0.330	0.120	0.400	0.080	0.300
9	0.130	0.400	0.040	0.330	0.100	0.400	0.000	0.400
10	0.216	0.216	0.000	0.468	0.123	0.123	0.080	0.574
11	0.203	0.203	0.080	0.414	0.260	0.260	0.080	0.300
12	0.255	0.255	0.080	0.310	0.300	0.300	0.000	0.300
13	0.275	0.275	0.000	0.350	0.360	0.100	0.080	0.360
14	0.210	0.210	0.040	0.440	0.100	0.360	0.080	0.360
15					0.260	0.260	0.080	0.300
The fifth mixture component x_5 has been set to constant at $x_5 = 0.10$								

of the upper or lower bound in any of the constraints may substantially change not only the locations of the vertices but the number of vertices as well. As a consequence of this change in the shape of the feasible region, locations of various centroids and their counts may also drastically change. Clearly, this overemphasis on constraints and vertices is undesirable and completely avoided in our approach. Designs such as simplex lattice and simplex centroid are essentially the gold standards for the unconstrained mixture problem as long as experiments on vertices, faces and edges of the complete simplex are permissible. Many good interior designs can also be found for the complete mixture and for the unconstrained problem. Thus it is natural to start with one such suitably chosen base design and modify it by obtaining a design under the constraints, which is closest to the base design in some meaningful sense. QP-procrustated designs do just that.

We must however note that one can always come up with a situation where the feasible region is very small and hence any approach to obtain a design may yield points some of which are too close to each other to cause severe multicollinearity. QP-procrustation may fail to fix such problem as the inherent difficulty in such a case is one or more constraints allowing only a very narrow slice in that component. We feel that still in such situations, procrustation will result in a superior design than the extreme vertices approach. However, it is certainly of interest to further research the comparative merits of the QP-procrustated designs and extreme vertices designs under various other criteria.

ACKNOWLEDGEMENTS

Author wishes to thank the referee for his/her valuable suggestions.

REFERENCES

Anderson, V.L. and McLean, R.A. (1974). *Design of Experiments, A Realistic Approach*. Dekker.

Anik, S.T. and Sukumar, L. (1981). Extreme vertexes design in formulation development: Solubility of Butoconazole Nitrate in a multicomponent system. *J. Pharma. Scis.*, **70,** 897-900.

Barbuta, M. and Lepadatu, D. (2008). Mechanical characteristics investigation of polymer concrete using mixture design of experiments and response surface method. *J. Appl. Sci.*, **8,** 2242-2249.

Box, G.E.P. and Draper, N.R. (2007). *Response Surfaces, Mixtures and Ridge Analyses*. Wiley.

Cornell, J.A. (1986). A Comparison between two ten-point designs for studying three-component mixture systems. *J. Qual. Tech.*, **18**, 1-15.

Cornell, J.A. (2002). *Experiments with Mixtures, Design, Models and Analysis of Mixture Data*. Wiley.

Gorman, J.W. (1966). Discussion of extreme vertices design of mixture experiments by McLean, R.A. and Anderson, V.L., *Technometrics*, **8,** 455-456.

Juan, E.M.S., Edra, E.V., Sales, J.M., Lustre, A.O. and Resurreccion, A.V.A. (2006). Utilization of peanut fines in the optimization of peanut polvoron using mixture response surface methodology. *Inter. J. Food Sci. Tech.*, **41**, 768-774.

Khattree, R. (2014a). Mixture experiments in the interior: Yantram designs. To appear in the *J. Stat. Theo. Prac.*

Khattree, R. (2014b). A Class of Designs for Mixture Experiments Implied by the Numerical Configurations in Hindu Yantram, Preprint.

Khuri, A.I. and Cornell, J.A. (1996). *Response Surfaces, Designs and Analyses*. Dekker.

Lawson, J. (2010). *Design and Analysis of Experiments with SAS*®. CRC Press.

McLean, R.A. and Anderson, V.L. (1966). Extreme vertices design of mixture experiments. *Technometrics,* **8,** 447-457.

Myers, R.H., Montgomery, D.C. and Anderson-Cook, C.M. (2009). *Response Surface Methodology and Product Optimization using Designed Experiments*. Wiley.

Sinha, B.K., Mandal, N.K., Pal, M. and Das, P. (2014). *Optimal Mixture Experiments*. Springer.

Appendix A

The SAS® code to generate extreme vertices and all centroids for all problems reported in Table 1 (q = 3):

```
***Macro for the Computation of Extreme Vertices and all Centroids***;
****************** q = 3***************;
***The l1, l2 l3 are the lower bounds and u1, u2, u3 are the upper bounds on respective mixture components. The
corresponding data set is specified as dataset = . ***;

%macro extreme(dataset = , l1 = , l2 = , l3 = , u1 = , u2 = , u3 = );
%adxgen
%adxmix
%adxinit
%adxxvert(exvert&dataset,x1 &l1 -&u1 /x2 &l2-&u2 /x3 &l3 -&u3)

%adxmamd(mamd&dataset,x1 &l1 -&u1 /x2 &l2-&u2 /x3 &l3 -&u3)
title "Extreme Vertices for &dataset";
proc print data = exvert&dataset;run;
proc print data = mamd&dataset;run;
title;
%mend;

%extreme(dataset = boxdraperlow, l1 = .2, l2 = .2, l3 = .3, u1 = 1, u2 = 1, u3 =1 );

%extreme(dataset = khurip382, l1 = .20, l2 = .05, l3 = .15, u1 = .5, u2 = .65, u3 =.75 );

%extreme(dataset = Juanlawson, l1 = .0, l2 = .1, l3 = .05, u1 = .8, u2 = .95, u3 =.5 );

%extreme(dataset = aniksukumar3a, l1 = .1111, l2 = .1111, l3 = .3333, u1 = .4444, u2 = .4444, u3 =.7778 );

%extreme(dataset = aniksukumar3b, l1 = .1220, l2 = .1220, l3 = .3656, u1 = .4878, u2 = .4878, u3 =.8537 );

%extreme(dataset = khurip351, l1 = .2, l2 = .1, l3 = .1, u1 = .6, u2 = .6, u3 =.5 );

%extreme(dataset = upperbound, l1 = .0, l2 = .0, l3 = .0, u1 = .7, u2 = .6, u3 =.5 );
```

Appendix B

The SAS® code to generate a "Quadratic Program Procrustated Design" from the 10-point Simplex Lattice Design for three-component mixture ($q = 3$) and for Khuri and Cornell (1996, p. 351):

```
%let title =  "For q = 3, Closest QP-procrustated Design for the 10-Point
Simplex Lattice Design";
%let title2  = "Khuri and Cornell (1996 p. 351)";
%let title3 = " Constraints: x1 in [.2,.6],   x2 in [.1, .6],   x3 in [.1,
.5]" ;


data mixture;
input field1 $ field2 $ field3$ field4 field5 $ field6 @;

***Specify upperbounds, lowerbounds and sum of all components in RHS block***;
datalines;
NAME     .        EXAMPLE    .                .           .
ROWS     .        .           .                .           .
N        OBJ      .           .                .           .
L        R1       .           .                .           .
L        R2       .           .                .           .
L        R3       .           .                .           .

G        R4       .           .                .           .
G        R5       .           .                .           .
G        R6       .           .                .           .

E        R7        .          .                 .          .
COLUMNS  .       .           .                .           .
.        X1       R1          1.0               .          .
.        X1       R4          1.0              .           .
.        X1       R7          1.0               .           .
.        X1       OBJ         -1000               .           .

.        X2       R2          1.0               .          .
.        X2       R5          1.0              .           .
.        X2       R7          1.0               .           .
.        X2       OBJ         -2000            .           .

.        X3       R3          1.0               .          .
.        X3       R6          1.0              .           .
.        X3       R7          1.0               .           .
.        X3       OBJ         -3000                .           .

RHS      .        .           .                .           .
```

```
.        RHS     R1          .6                .          .
.        RHS     R2          .6            .          .
.        RHS     R3          .5              .          .
.        RHS     R4          .2              .          .
.        RHS     R5          .1             .          .
.        RHS     R6          .1             .          .
.        RHS     R7          1               .          .

RANGES  .        .           .                 .          .
BOUNDS  .        .           .                 .          .
QUADOBJ  .       .           .                 .          .
.        X1      X1          2.0               .          .
.        X2      X2          2.0               .          .
.        X3      X3          2.0                .          .
ENDATA  .        .           .                 .          .
;

%macro datasets(point= , field4_1 = , field4_2 = , field4_3 = );
data point&point; set mixture;
if field4 = -1000 then field4 = -2*&field4_1;
if field4 = -2000 then field4 = -2*&field4_2;
if field4 = -3000 then field4 = -2*&field4_3;
run;
title &title ;
title2 &title2 ;
title3 &title3 ;
proc optqp data=point&point
    primalout = mixpout&point
    dualout    = mixdout&point;
run;

%mend;


%datasets(point= 1, field4_1 = 1, field4_2 =0 , field4_3 = 0);
%datasets(point= 2, field4_1 = 0, field4_2 = 1, field4_3 = 0);
%datasets(point= 3, field4_1 = 1, field4_2 = 0, field4_3 = 1);
%datasets(point= 4, field4_1 = .3333, field4_2 =.6667 , field4_3 =0 );
%datasets(point= 5, field4_1 =.6667 , field4_2 =.3333 , field4_3 =0 );
%datasets(point= 6, field4_1 = .3333, field4_2 = 0, field4_3 = .6667);
%datasets(point= 7, field4_1 = .6667, field4_2 = 0, field4_3 = .3333);
%datasets(point= 8, field4_1 = 0, field4_2 =.3333 , field4_3 =.6667 );
%datasets(point= 9, field4_1 = 0, field4_2 = .6667 , field4_3 =.3333 );
%datasets(point= 10, field4_1 = .3333 , field4_2 = .3334, field4_3 = .3333);
data mixpout1; set mixpout1; value = _VALUE_; keep value;run;
data mixpout2; set mixpout2; value = _VALUE_; keep value;run;
data mixpout3; set mixpout3; value = _VALUE_; keep value;run;
data mixpout4; set mixpout4; value = _VALUE_; keep value;run;
data mixpout5; set mixpout5; value = _VALUE_; keep value;run;
data mixpout6; set mixpout6; value = _VALUE_; keep value;run;
data mixpout7; set mixpout7; value = _VALUE_; keep value;run;
data mixpout8; set mixpout8; value = _VALUE_; keep value;run;
data mixpout9; set mixpout9; value = _VALUE_; keep value;run;
data mixpout10; set mixpout10; value = _VALUE_; keep value;run;

proc transpose data = mixpout1 out = value1;run;
proc transpose data = mixpout2 out = value2;run;
proc transpose data = mixpout3 out = value3;run;
proc transpose data = mixpout4 out = value4;run;
proc transpose data = mixpout5 out = value5;run;
proc transpose data = mixpout6 out = value6;run;
proc transpose data = mixpout7 out = value7;run;
proc transpose data = mixpout8 out = value8;run;
proc transpose data = mixpout9 out = value9;run;
proc transpose data = mixpout10 out = value10;run;


data design;set value1-value10; run;


data QPclosest; set design;
x1 = col1; x2 = col2; x3 = col3;
designpt = _n_;
run;

proc print data= QPclosest; var x1 x2 x3;run;


data orig_design; input z1 z2 z3 ;
designpt = _n_;
datalines;
1 0 0
0 1 0
0 0 1
.6667 .3333 0
.3333 .6667 0
.3333 0 .6667
.6667 0 .3333
0 .3333 .6667
0 .6667 .3333
.3333 .3334 .3333
;
data base_and_procrustated; merge orig_design QPclosest; by designpt; run;
proc print data= base_and_procrustated ; var designpt z1 z2 z3 x1 x2 x3;run;
```