Statistics and Applications {**ISSN 2452-7395 (online)**}
Volume 17, No. 1, 2019 (New Series), pp 33-40

D-Efficient Composite-type Second Order Designs via Computer Search

Aloke Dey and Deepayan Sarkar
Indian Statistical Institute, New Delhi 110 016, India

Received: April 17, 2018; Revised: September 11, 2018; Accepted: September 11, 2018

Abstract

Using computer search algorithms, second order designs of composite type over a k-cube $[-1, 1]^k$ are obtained, where k is the number of factors. The advantages of the proposed approach are that (i) it is possible to obtain designs with higher D-efficiencies than a comparable orthogonal array composite design (OACD), and (ii) designs with fewer points than those required by an OACD and having comparable D-efficiencies can be obtained.

Key words: Response surface designs; Random search; D-efficiency.

1 Introduction and Preliminaries

Response surface methodology is used for exploring the relationship between one (or more) response variable(s) and several explanatory or input variables. We consider only a single response variable. In practice, often a second order model is used to explore such a relationship. With k quantitative factors $x_1, x_2, \ldots, x_k$, a full second order model is given by

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \epsilon, \tag{1}$$

where $\beta_0, \beta_i, \beta_{ii}$, and β_{ij} are respectively the intercept, linear, quadratic, and interaction terms, and ϵ is the random error term. A response surface design is called a *second order design* if it allows the estimability of all the parameters in (1).

Several second order designs are available in the literature. Excellent accounts of response surface methodology, including designs, are available in Box and Draper (2007), Khuri and Cornell (1996), and Myers et al. (2016). A popular choice of second order designs has been the central composite designs (CCD) introduced by Box and Wilson (1951). A central composite design has the following sets of points: (i) n_1 'cube' points, $(\pm 1, \pm 1, \ldots, \pm 1)$, (ii) $2k$ axial points, $(\pm\alpha, 0, \ldots, 0), (0, \pm\alpha, 0. \ldots, 0), \ldots, (0, 0, \ldots, 0. \pm \alpha)$, (iii) n_0 center points, $(0, 0, \ldots, 0)$. Thus, a central composite design has 5 levels for each factor if $\alpha \neq 1$ and 3 levels, if $\alpha = 1$. We restrict

Corresponding Author: Aloke Dey
E-mail: aloke.dey@gmail.com

attention to $\alpha = 1$ in this paper. The cube points of a central composite design are taken to be either those of a complete 2^k factorial with levels $-1, 1$ or a suitable fraction of a 2^k factorial. More generally, a second order design may be called a composite-type design if it has the following sets of points: (a) points with two levels, ± 1, for each factor, (b) points with three levels, $0, \pm 1$, for each factor and, (c) center points.

Xu et al. (2014) introduced a class of second order designs, called *orthogonal array composite designs* (OACD), as an alternative to the usual central composite designs. In OACD, the axial points of a central composite design are replaced by the runs of a 3-symbol orthogonal array. Like the CCD, OACD can be used for a single experiment or for sequential experimentation. The advantage of using the runs of a 3-symbol orthogonal array is that these provide information on the interaction terms in (1), apart from providing information on the linear and quadratic terms, whereas the axial points in a central composite design do not provide information on the interaction terms. In view of this, an OACD is likely to perform better than a corresponding CCD when the interaction terms are important.

In a recent paper, Zhou and Xu (2017) further investigated the properties of OACD by providing results on the D-efficiency of these designs over a k-cube, $[-1, 1]^k$, relative to the D-optimal (approximate) design measure given by Farrell et al. (1967). The investigations of Zhou and Xu (2017) demonstrate the superiority of OACD over CCD in terms of D-efficiency.

For completeness and later use, we briefly review the concept of D-efficiency; see, e.g., Kiefer (1961) for more details. Let d be a second order design involving N runs. Let $X_d = [\mathbf{1}, L, Q, I]$ denote the model matrix of d, where $\mathbf{1}$ is a column of ones, and L, Q, I denote the columns corresponding to the linear, quadratic, and interaction terms, respectively, in (1). The information matrix corresponding to d is then $M_d = X_d' X_d / N$. The D-optimality criterion seeks to maximize $\det(M_d)$, where $\det(\cdot)$ denotes the determinant of a square matrix. Let ξ^* denote the approximate D-optimal design over a cube $[-1, 1]^k$. It was shown by Kiefer (1961) and Farrell et al. (1967) that the determinant of the information matrix under ξ^* is given by

$$\det(M_{\xi^*}) = u^k v^{k(k-1)/2} (u - v)^{k-1} \{u + (k-1)v - ku^2\}, \tag{2}$$

where

$$\begin{aligned} u &= \frac{k+3}{4(k+1)(k+2)^2} \{2k^2 + 3k + 7 + (k-1)w\} \\ v &= \frac{k+3}{8(k+2)^3(k+1)} \{4k^3 + 8k^2 + 11k - 5 + (2k^2 + k + 3)w\} \\ w &= (4k^2 + 12k + 17)^{1/2}. \end{aligned}$$

The D-efficiency of d is then given by

$$D\text{-efficiency} = \left(\frac{\det(M_d)}{\det(M_{\xi^*})} \right)^{1/p}, \tag{3}$$

where $p = (k+1)(k+2)/2$ is the number of parameters in the model (1). The D-efficiency expression (3) actually provides a conservative lower bound to the D-efficiency of the design d

as it is computed relative to a hypothetical optimal design which will be unattainable with a finite number of observations.

The purpose of this paper is to propose new second order designs over a k-cube with high D-efficiencies, obtained via random computer search algorithms similar in spirit to previously proposed exchange algorithms (see St John and Draper, 1975, for an overview). The advantages of this approach are twofold, viz., (i) it is possible to obtain designs with higher D-efficiencies than a comparable OACD, and (ii) designs with fewer points than those required by an OACD and having high D-efficiencies can be obtained. In Section 2, we describe the proposed algorithms. Section 3 presents the results and a discussion.

2 The Algorithms

We propose two approaches to obtain second order designs with high D-efficiencies. These are based on three simple basic operations, described fully in the Appendix for completeness: Algorithm 1 (RANDOMDESIGN) generates a random design on a cube; Algorithm 2 (IMPROVEDESIGNGREEDY) updates a given design in a deterministic way by changing a single element by choosing the change that provides the best improvement in D-efficiency; and Algorithm 3 (IMPROVEDESIGNRANDOM) improves a given design by changing a single element randomly, provided that the change improves D-efficiency.

Our first proposed search procedure, which we call *Random-Greedy*, involves two steps for given values of the number of runs N and the number of factors k:

procedure RANDOM-GREEDY(N, k)
  Obtain an initial design $X \leftarrow$ RANDOMDESIGN(N, k)
  repeat
    $X \leftarrow$ IMPROVEDESIGNGREEDY(X)
  until no further improvement is possible
end procedure

Alternatively, the *Random-Random* search procedure updates the design using IMPROVEDESIGN-RANDOM() rather than IMPROVEDESIGNGREEDY().

procedure RANDOM-RANDOM(N, k)
  Obtain an initial design $X \leftarrow$ RANDOMDESIGN(N, k)
  repeat
    $X \leftarrow$ IMPROVEDESIGNRANDOM(X)
  until no further improvement is possible
end procedure

Of course, one may expect the final design to have higher D-efficiency if the initial starting point already has high D-efficiency. This suggests the following natural variants of the above algorithms, where the initial random design is replaced by a suitable OACD following Zhou and Xu (2017). This works only when an OA of the desired size is available. The *OA-Greedy* search procedure runs as follows.

procedure OA-GREEDY(N, k)
  $X \leftarrow$ OACD(N, k)
  repeat
    $X \leftarrow$ IMPROVEDESIGNGREEDY(X)
  until no further improvement is possible
end procedure

Similarly, the *OA-Random search* algorithm runs as follows.

procedure OA-RANDOM(N, k)
  $X \leftarrow$ OACD(N, k)
  repeat
    $X \leftarrow$ IMPROVEDESIGNRANDOM(X)
  until no further improvement is possible
end procedure

In practice, it may be useful to terminate after a fixed number of updates even if further improvements are possible, especially for large values of N or k. Also, for the *Random-Greedy* and *Random-Random* search procedures, the final result depends on the initial random choice, so it is useful to choose the initial random design as the best obtained from several tries instead of just one. One can also repeat the entire procedure multiple times and choose the best design thus obtained. These details, along with properties of the resulting designs, are described in Section 3.

3 Results and Discussion

In this section, we compare the D-efficiencies of the OACD with those obtained via the random search algorithms proposed in the previous section.

For the results summarized here, we performed four sets of experiments. The first two employ a given OACD(N, k) design as a starting point, and tries to improve it using either procedure *OA-Random* or procedure *OA-Greedy*. In both cases, improvement is attempted for 30000 iterations, or until no further improvements are possible, whichever is earlier. Additionally, for the first case employing procedure *OA-Random*, the whole search procedure is repeated 10 times and the design with the highest D-efficiency retained.

The last two sets of experiments are based on the *Random-Random* and *Random-Greedy* procedures, which are used to obtain designs with the number of runs N varying from p, the number of parameters in (1), to the number of runs in the respective OACD. In both cases, the initial design is chosen to be the one with highest D-efficiency among 5000 random designs, and improvement over this initial choice is attempted for 20000 iterations, or until no further improvements are possible. Additionally, the search procedure is repeated 20 times and the design with the highest D-efficiency retained.

In Table 1 and Figure 1, we present this comparison for $k = 4, 5, \ldots, 10$ factors. In particular, the D-efficiencies of the OACD designs are compared with four designs obtained by computer search: *OACD+random* starts from the respective OACD and employs procedure *OA-random*. Similarly, *OACD+greedy* starts from the OACD and employs procedure *OA-greedy*. Finally,

Table 1: Comparison of Percentage D-efficiency of OACD with the new designs of same size obtained through computer search. The best value for each row is highlighted in bold.

k	N	OACD	OACD+ random	OACD+ greedy	random+ random	random+ greedy
4	25	93.1	96.7	96.5	**97.7**	97.3
5	34	95.3	95.3	95.3	**96.4**	96.3
6	50	96.6	**97.5**	97.3	96.7	97.3
7	82	93.9	97.9	97.7	97.8	**97.9**
8	91	95.9	**97.4**	97.3	96.8	96.8
9	155	94.9	**98.3**	98.2	98.0	98.2
10	155	95.3	**97.6**	97.5	96.7	97.0

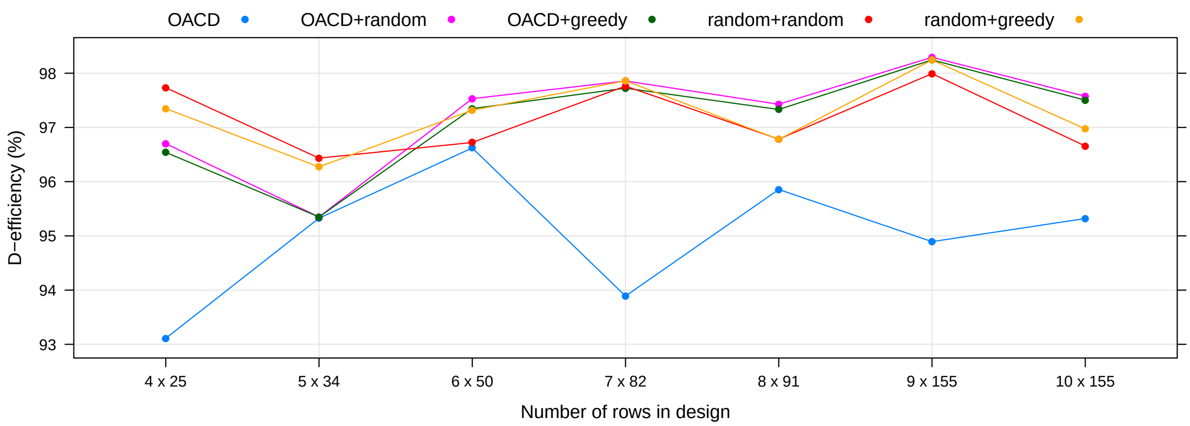

Figure 1: Comparison of Percentage D-efficiency under OACD with the new designs.

random+random employs procedure *Random-Random* and *random+greedy* employs procedure *Random-Greedy* to obtain designs without using the respective OACD.

As may be seen from the results, the designs obtained by computer search provide substantial improvement in almost all cases, although no single method outperforms the others. One interesting observation is that for the methods starting from the respective OACD, random search usually performs better, presumably because the greedy approach being deterministic gets stuck in a local optimum. On the other hand, among the other two methods which use multiple random starting points, the greedy method often gives better results.

Figure 2 shows the D-efficiency of designs obtained by *Random-Greedy* procedure for $k = 4, 5, \ldots, 10$ factors, with the number of runs N varying from the lowest permitted value to the number of runs in the respective OACD. It is clear from the figure that a D-efficiency comparable to OACD can be obtained in a design with substantially fewer runs. The maximum improvement obtained in our experiments is summarized in Table 2.

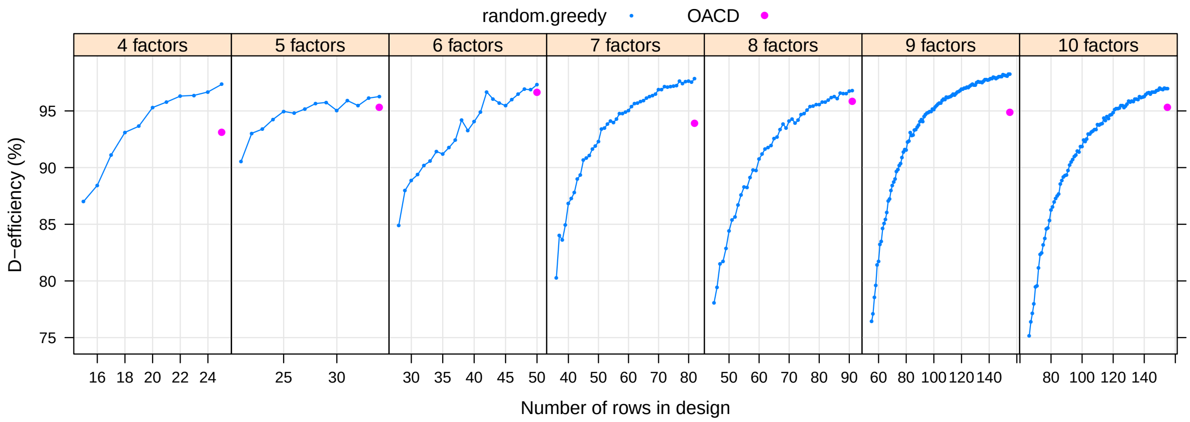

Figure 2: Percentage D-efficiency of designs obtained by *Random-Greedy* search for various values of k and N. The D-efficiency of the respective OACD is shown for comparison. Corresponding results for *Random-Random* search are not shown, but the resulting D-efficiencies are largely similar.

4 Supplementary material

The following are included as online supplementary material.

The file `random_greedy_summary.csv` gives a table of the largest D-efficiencies obtained using the *Random-Greedy* procedure for all values of k and N, and the file `random_random_summary.csv` similarly gives a table of the largest D-efficiencies obtained using the *Random-Greedy* procedure.

For readers interested in the actual designs, the largest D-efficiency obtained (using all applicable methods) for all values of k and N are given in file `best_design_summary.csv`, and the corresponding designs are given as individual files in the `best_designs` folder.

Finally, R code to run the algorithms is provided in the file `cubic_designs.R`.

Acknowledgment

The work of A. Dey was supported by the National Academy of Sciences, India, under the Senior Scientist program of the Academy. The support is gratefully acknowledged.

Appendix: Main algorithms

The random search procedures discussed in this article are based on the following basic algorithms. The procedure Random(S) used in these algorithms, when called with a finite set S as argument, selects an element of S randomly with equal probability.

Algorithm 1 Generate random design

```
procedure RANDOMDESIGN(m, n)
    X ← new n × m matrix
    for i ← 1, 2, ..., n do
        for j ← 1, 2, ..., m do
            X_ij ← Random({−1, 0, 1})          ▷ Select one with equal probability
        end for
    end for
    return X
end procedure
```

Algorithm 2 Improve design by greedy search

```
procedure IMPROVEDESIGNGREEDY(X)
    m ← number of rows of X
    n ← number of columns of X
    for i ← 1, 2, ..., n do
        for j ← 1, 2, ..., m do
            for x ← −1, 0, 1 do
                E_ijx ← D-efficiency when X_ij is replaced by x
            end for
        end for
    end for
    if max E_ijx > D-efficiency of X then
        return updated X corresponding to max E_ijx
    else
        return X                                ▷ Unchanged
    end if
end procedure
```

Algorithm 3 Improve design by random search

```
procedure IMPROVEDESIGNRANDOM(X)
    Y ← X
    m ← number of rows of X
    n ← number of columns of X
    i ← Random({1, 2, ..., n})
    j ← Random({1, 2, ..., m})
    Y_ij ← Random({−1, 0, 1} \ {X_ij})          ▷ Replace Y_ij randomly
    if D-efficiency of Y > D-efficiency of X then
        return Y                                ▷ Updated
    else
        return X                                ▷ Unchanged
    end if
end procedure
```

Table 2: Number of runs in smallest design found by *Random-Greedy* or *Random-Random* search to have D-efficiency at least as high as that of the OACD with same number of factors. For k factors, $N_{min} = (k+1)(k+2)/2$, the number of parameters in (1), N_{OACD} is the number of runs in the OACD, and N_{search} is the number of runs in the best design found by random search. The corresponding D-efficiency values are given in parentheses. The percent reduction in number of runs is relative to the maximum possible reduction.

k	N_{min}	N_{OACD}	N_{search}	Reduction (%)
4	15	25 (93.1)	19 (93.6)	60.0
5	21	34 (95.3)	28 (95.7)	46.2
6	28	50 (96.6)	42 (96.7)	36.4
7	36	82 (93.9)	54 (93.9)	60.9
8	45	91 (95.9)	82 (96.2)	19.6
9	55	155 (94.9)	97 (94.9)	58.0
10	66	155 (95.3)	125 (95.5)	33.7

References

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

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society, Series B*, **13**, 1–45.

Farrell, R. H., Kiefer, J., and Walbran, A. (1967). Optimum multivariate designs. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 113–138. University of California Press Berkeley, CA.

Khuri, A. I. and Cornell, J. A. (1996). *Response Surfaces: Designs and Analyses*. CRC press, 2nd edition.

Kiefer, J. (1961). Optimum designs in regression problems II. *The Annals of Mathematical Statistics*, **32**, 298–325.

Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2016). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, Inc, 4th edition.

St John, R. C. and Draper, N. R. (1975). D-optimality for regression designs: a review. *Technometrics*, **17(1)**, 15–23.

Xu, H., Jaynes, J., and Ding, X. (2014). Combining two-level and three-level orthogonal arrays for factor screening and response surface exploration. *Statistica Sinica*, **24**, 269–289.

Zhou, Y.-D. and Xu, H. (2017). Composite designs based on orthogonal arrays and definitive screening designs. *Journal of the American Statistical Association*, **112(520)**, 1675–1683.