Construction of Efficient Fractional Factorial Designs for General Factorials under a Baseline Parametrization

Rahul Mukerjee
Indian Institute of Management Calcutta, Kolkata 700104, India
rmuk0902@gmail.com

Abstract

Fractional factorial designs have wide applicability in diverse fields. While the literature on these designs revolves around the well-known orthogonal parametrization, a baseline parametrization has started gaining popularity in recent years. The latter has found use in cDNA microarray experiments and can arise naturally in many other situations whenever each factor has a control or baseline level. Designing optimal or efficient fractions under baseline parametrization is, however, a challenge; for example, under this parametrization, very often orthogonal arrays do not lead to optimal designs. As a result, the related combinatorics become rather complex for general factorials. We aim at addressing this problem. In order to obtain highly efficient fractions of general factorials with an arbitrary number of levels for each factor, we begin by employing the approximate theory based on directional derivatives. The underlying model is kept quite flexible – it includes the baseline effect, all main effects and selected (user specified) interactions, if any. Somewhat counter intuitively, even after the use of approximate theory, it is seen that a naïve discretization of the resulting optimal design measure often fails to serve our purpose, especially in the practically important case where the run size is relatively small. As a way out, we propose supplementing approximate theory by a step-down procedure which is found to work very well in producing economic and highly efficient fractions of general factorials. Several illustrative examples are given.

Keywords: approximate theory, directional derivative, step-down procedure.

1. Introduction

Design of optimal or efficient factorial fractions has received significant attention in recent years; see Mukerjee and Wu (2006), Xu et al. (2009) and Wu and Hamada (2009) for recent surveys and further references. These designs have wide ranging applications, notably in industry, agriculture and medicine. A vast majority of the existing work on fractional factorial designs centers around the well-known orthogonal parametrization (Gupta and Mukerjee, 1989) and orthogonal arrays play a key role in the construction of optimal fractions.

In recent years, however, another kind of parametrization, namely, the baseline parametrization, has started gaining popularity in factorial experiments. It has found use in cDNA microarray experiments (Yang and Speed, 2002) and can also arise naturally in many other situations whenever each factor has a control or baseline level. Following Kerr (2006), an example is given by a toxicological study with binary factors, each representing the presence or absence of a toxin, where the state of absence is a natural baseline level of each factor. In the context of microarray experiments, Glonek and Solomon (2004), Banerjee and Mukerjee (2008) and Sanchez and Glonek (2009) studied optimal paired comparison designs for full factorials under baseline parametrization. As for designing efficient or optimal factorial fractions under this parametrization, not much work has been reported so far beyond the recent results in Mukerjee and Tang (2012) who studied the case of two-level factors.

Indeed, the lack of orthogonality in baseline parametrization complicates the combinatorics underlying the design problem for general fractional factorials – for exam-

ple, unlike what happens in orthogonal parametrization, designs given by orthogonal arrays cease to remain optimal beyond the situation where each factor has two levels and all interactions are absent; cf. Mukerjee and Tang (2012). The present article aims at addressing this problem. The design objective is formulated in the next section with reference to an underlying model which is kept quite flexible. It includes the baseline effect, all main effects and selected interactions, if any, as specified by the user. Then in Section 3, we show how the approximate theory, in conjunction with a step-down procedure, can yield highly efficient fractions of general factorials with an arbitrary number of levels for each factor and arbitrary run size. Illustrative examples appear in Section 4 and we conclude with some remarks in Section 5.

2. Design objective

Consider an $m_1 \times ... \times m_n$ factorial with n factors $F_1,..., F_n$, the levels of F_i being coded as 0,1, …, $m_i - 1$. Let $\tau(j_1...j_n)$ denote the effect of a typical treatment combination $j_1...j_n$, where $0 \le j_i \le m_i - 1$, $1 \le i \le n$. Suppose each factor has a control or baseline level, say 0. Then under a full factorial model, which includes all main effects and interactions, the baseline parametrization is given by

$$\tau(j_1...j_n) = \Sigma\, \theta(u_1...u_n), \qquad (1)$$

for every $j_1...j_n$, where for each i, the sum keeps u_i fixed at 0 if $j_i = 0$, and allows u_i to take only two distinct values 0 and j_i if $j_i > 0$; cf. Banerjee and Mukerjee (2008). In (1), $\theta(0...0)$ is the baseline effect, while for $u_1...u_n \neq 0...0$, $\theta(u_1...u_n)$ is a main or interaction effect parameter, with the position(s) of the nonzero component(s) of $u_1...u_n$ determining which main or interaction effect such a $\theta(u_1...u_n)$ represents. For example, in a $2^2 \times 3$ factorial, (1) yields

$$\tau(000) = \theta(000), \qquad \tau(101) = \theta(000) + \theta(001) + \theta(100) + \theta(101),$$
$$\tau(112) = \theta(000) + \theta(002) + \theta(010) + \theta(012) + \theta(100) + \theta(102) + \theta(110) + \theta(112), \qquad (2)$$

etc. Here $\theta(001)$ and $\theta(002)$ represent main effect F_3, while $\theta(110)$ represents interaction F_1F_2, and so on.

In order to explore efficient factorial fractions under baseline parametrization, hereafter we consider a model which includes the baseline effect, all main effects and user specified interactions, if any. This leads to a reduced version of (1), namely,

$$\tau(j_1...j_n) = \Sigma^*\, \theta(u_1...u_n), \qquad (3)$$

which discards all $\theta(u_1...u_n)$ representing any interaction not included in the model. It is easily seen that (3) can be expressed in a more compact matrix notation as

$$\tau(j_1...j_n) = x(j_1...j_n)^T \theta, \qquad (4)$$

where $x(j_1...j_n)$ is a known vector which depends on $j_1...j_n$ and the model, θ is the vector of parameters $\theta(u_1...u_n)$ retained in the model, and the superscript T stands for transposition. For illustration, consider again a $2^2 \times 3$ factorial and suppose the model includes the baseline effect, all main effects and interaction F_1F_2. Then the parameters $\theta(101)$, $\theta(102)$, $\theta(011)$, $\theta(012)$, $\theta(111)$ and $\theta(112)$, representing interactions F_1F_3, F_2F_3 and $F_1F_2F_3$ are discarded. So the vector θ, capturing the parameters kept in the model, equals $(\theta(000), \theta(001), \theta(002), \theta(010), \theta(100), \theta(110))^T$. While there is no change in the expression for $\tau(000)$ in (2), those for $\tau(101)$ and $\tau(112)$ reduce to $\tau(101) = \theta(000) + \theta(001) + \theta(100) = x(101)^T \theta$ and $\tau(112) = \theta(000) + \theta(002) + \theta(010) + \theta(100) + \theta(110) = x(112)^T \theta$, respectively, where $x(101) = (1,\ 1,\ 0,\ 0,\ 1,\ 0)^T$ and $x(112) = (1,\ 0,\ 1,\ 1,\ 1,\ 1)^T$.

For notational simplicity, we number the $v = \Pi_{i=1}^{n} m_i$ treatment combinations as 1,…,*v* in any fixed order, and for $1 \le k \le v$, write $x_k = x(j_1 \ldots j_n)$ if the treatment combination $j_1 \ldots j_n$ is numbered *k*. Now suppose *N* experimental runs are allowed and consider a design where treatment combinations numbered 1,…,*v* are replicated f_1, ..., f_v times, respectively, with the nonnegative integers $f_1, \ldots, f_v$ adding up to *N*. Let $\pi_k = f_k / N$, $1 \le k \le v$. Then under the usual assumptions that the observations arising from the design have equal variance and are uncorrelated, the information matrix for θ in (3) or (4) is proportional to $NM(\pi)$, where $M(\pi) = \Sigma_{k=1}^{v} \pi_k x_k x_k^T$. We aim at finding a design that estimates the components of θ with minimum possible total variance. This calls for minimizing $\mathrm{tr}[\{M(\pi)\}^{-1}]$ and amounts to working under the *A*-optimality criterion, which is appropriate in the present context where interest lies specifically in θ itself and not in any linear transformation thereof; cf. Kerr (2012).

3. Approximate theory and step-down procedure

As it stands, the design problem formulated above is too involved due to the discreteness of $f_1, \ldots, f_v$ which induces the same on $\pi = (\pi_1, \ldots, \pi_v)$. Considerable headway is possible, however, if we pretend that $\pi_1, \ldots, \pi_v$ are continuous, i.e., that they are any nonnegative real numbers which add up to unity. This enables us to use approximate theory via directional derivative considerations (Silvey, 1980) for finding an *A*-optimal design measure $\pi_0 = (\pi_{10}, \ldots, \pi_{v0})$ that minimizes $\mathrm{tr}[\{M(\pi)\}^{-1}]$ over all such continuous π. In the spirit of Torsney and Martin-Martin (2008), a multiplicative algorithm can be conveniently employed to find π_0. Our computational experience in many examples including the ones in Section 4 shows that such an algorithm, implemented for instance on MATLAB, converges very fast in the present context even when a level of accuracy as high as up to ten places of decimals is called for.

At this stage, one may be inclined to believe that use of approximate theory alone can satisfactorily solve the present design problem because after obtaining the *A*-optimal design measure $\pi_0 = (\pi_{10}, \ldots, \pi_{v0})$ it only remains to round off $N\pi_{10}, \ldots, N\pi_{v0}$ to nearest integers to reach an exact *N*-run design which should be highly efficient. However, two difficulties emerge. First, given *N*, there may not exist any constant *c*, including *c* = *N*, such that the quantities $c\pi_{10}, \ldots, c\pi_{v0}$, when rounded off to nearest integers, add up to *N*. Second, even when such a constant *c* exists, it may turn out that the *N*-run exact design so obtained via a naïve discretization has poor *A*-efficiency. Both these difficulties are seen to be quite serious especially for relatively small *N*, a situation which is of practical importance from consideration of experimental economy.

A step-down procedure comes very handy in circumventing these problems. In essence, after the *A*-optimal design measure $\pi_0 = (\pi_{10}, \ldots, \pi_{v0})$ is obtained via approximate theory, it works as follows. Take a large *c* and round off $c\pi_{10}, \ldots, c\pi_{v0}$ to nearest integers to get an exact design d_1, say in N_1 runs, where N_1 is much larger than the target run size *N*. Since *c* and hence N_1 are quite large, the loss of efficiency in d_1, due to discretization, is expected to be negligible. Then the step-down procedure involves $N_1 - N$ steps where in each step one run is deleted from d_1, this run being chosen, from amongst all available candidates for deletion, so as to maximize the *A*-efficiency of the resulting design in this step. As the examples in Section 4 and many others not reported here demonstrate, typically the final *N*-run design obtained in this manner has quite high *A*-efficiency, often larger than 0.90 or even 0.95. Finding the appropriate choice of *c* and hence of N_1 to initiate the step-down procedure

may require a little trial and error. Our computational experience suggests that for $N \leq 50$ it often suffices to start with N_1 in the range 400-500.

4. Examples

Example 1. Consider a $2^5 \times 3$ factorial and a model which includes the baseline effect, all main effects, and interactions F_1F_6 and F_2F_6. Here a saturated design will require 12 runs. We consider N = 14, 17 and 19. Note that each of 17 and 19 is prime to 2 and 3. Table 1 displays the A-optimal design measure $\pi_0 = (\pi_{10},..., \pi_{v0})$ in this example, obtained by the multiplicative algorithm. For instance, it shows that π_0 assigns a mass 0.0241 to each of the treatment combinations 010000 and 100000. One can check that there does not exist any constant c such that rounding off $c\pi_{10}$, ..., $c\pi_{v0}$ to nearest integers yields a nonsingular exact design with run size $N \leq 32$. Thus naïve discretization of π_0 does not work for the run sizes considered here. On the other hand, if we take c = 500 and round off $c\pi_{10},..., c\pi_{v0}$ to nearest integers then we get an exact design d_1 in N_1 = 498 runs and having A-efficiency 0.9999. Step-down from d_1 yields exact designs, say $d^{(14)}$, $d^{(17)}$ and $d^{(19)}$, in N = 14, 17 and 19 runs and having A-efficiencies 0.9058, 0.9392 and 0.9594, respectively. While the high A-efficiency of d_1 is not surprising given its large run size, it is noteworthy that step-down allows retaining much of this efficiency in exact designs having significantly smaller run sizes.

Table 1. A-optimal design measure in Example 1

Mass	Treatment combination
0.0479	000000
0.0241	each of 010000, 100000
0.0228	each of 000010, 000100, 001000
0.0221	each of 000001, 000002
0.0150	110000
0.0139	each of 010010, 010100, 011000, 100010, 100100, 101000
0.0130	each of 000110, 001010, 001100
0.0127	each of 000011, 000012, 000101, 000102, 001001, 001002
0.0119	111110
0.0110	each of 110010, 110100, 111000
0.0107	each of 010001, 010002, 100001, 100002
0.0102	each of 110110, 111010, 111100
0.0099	each of 010110, 011010, 011100, 100110, 101010, 101100
0.0093	each of 111111, 111112
0.0092	001110
0.0090	each of 000111, 000112, 001011, 001012, 001101, 001102
0.0088	each of 011110, 101110
0.0081	each of 001111, 001112
0.0077	each of 011111, 011112, 101111, 101112
0.0074	each of 010011, 010012, 010101, 010102, 011001, 011002, 100011, 100012, 100101, 100102, 101001, 101002
0.0066	each of 010111, 010112, 011011, 011012, 011101, 011102, 100111, 100112, 101011, 101012, 101101, 101102
0.0065	each of 110001, 110002
0.0064	each of 110111, 110112, 111011, 111012, 111101, 111102
0.0057	each of 110011, 110012, 110101, 110102, 111001, 111002

For completeness, we explicitly show the designs $d^{(14)}$, $d^{(17)}$ and $d^{(19)}$. The last of these consists of the 17 treatment combinations

000000, 000002, 000110, 000111, 000112, 001010, 001101, 010000, 010100, 011001, 011102, 100000, 100100, 101001, 101102, 110012, 111111,

each replicated once and the treatment combination 111010, replicated twice. The design $d^{(17)}$ is obtained by deleting the treatment combination 001101 and one replication of 111010 from $d^{(19)}$. Finally, $d^{(14)}$ is obtained by deleting the treatment combinations 000110, 010000, and 100000 from $d^{(17)}$.

Example 2. Consider a 2^6 factorial and a model which includes the baseline effect, all main effects, and the nine two-factor interactions $F_i F_j$, where $i \in \{1,2,3\}$ and $j \in \{4,5,6\}$. Here a saturated design will require 16 runs. We consider N = 16, 17 and 23. Note that each of 17 and 23 is prime to 2. As in Example 1, here too a naïve discretization of the A-optimal design measure π_0 fails to yield nonsingular exact designs for the target run sizes. We employ step-down procedure starting from an exact design d_1 having N_1 = 492 runs and obtained from π_0 by rounding off. This yields exact designs, say $d^{(16)}$, $d^{(17)}$ and $d^{(23)}$, in N = 16, 17 and 23 runs and having A-efficiencies 0.9185, 0.9559 and 0.9707, respectively. The last of these exact designs consists of the five treatment combinations 000000, 000011, 000101, 000110 and 110000, each replicated twice, and the thirteen treatment combinations

011000, 011001, 011010, 011100, 011111, 101000, 101001, 101010, 101100, 101111, 110011, 110101, 110110,

each replicated once. The design $d^{(17)}$ is obtained by deleting the treatment combinations 011000, 101000, and one replication of each of 000011, 000101, 000110 and 110000 from $d^{(23)}$. Finally, $d^{(16)}$ is obtained by deleting one replication of 000000 from $d^{(17)}$.

Example 3. Consider a $2^2 \times 3^2 \times 4$ factorial and a model which includes the baseline effect and all main effects. Here a saturated design requires 10 runs. We consider N = 16, 19 and 21. Once again, a naïve discretization of the A-optimal design measure π_0 fails to yield nonsingular exact designs for these run sizes. We employ step-down procedure starting from an exact design d_1 having N_1 = 410 runs and obtained from π_0 by rounding off. This yields exact designs, say $d^{(16)}$, $d^{(19)}$ and $d^{(21)}$, in N = 16, 19 and 21 runs and having A-efficiencies 0.9071, 0.9528 and 0.9537, respectively. The last of these exact designs consists of the treatment combination 10002 replicated twice and the nineteen treatment combinations

00000, 00003, 00020, 00100, 00122, 00201, 00220, 01003, 01010, 01111, 01212, 10011, 10110, 10213, 11020, 11021, 11100, 11123, 11200,

each replicated once. The design $d^{(19)}$ is obtained by deleting the treatment combination 11020 and one replication of 10002 from $d^{(21)}$. Finally, $d^{(16)}$ is obtained by deleting the treatment combinations 00100, 01003 and 10011 from $d^{(19)}$.

In all these examples, the reported A-efficiency figures are relative to the A-optimal design measure π_0 which is not attainable in finite run size. Hence the true A-efficiency of any exact design obtained in Examples 1-3 is even higher and it is quite likely that some of the exact designs reported in these examples are actually A-optimal among all exact designs having the same N.

5. Concluding remarks

While concluding, we indicate merits of the step-down procedure advocated here. First, as our examples show, it yields highly efficient exact designs keeping the run size N relatively small and quite arbitrary. Second, it is computationally far less intensive than deleting $N_1 - N$ runs from the large exact design d_1 in one shot in the most

efficient manner. Finally, our examples reveal yet another advantage of the step-down procedure. It allows an efficient exact design with a smaller number of runs to be nested in another with a larger number of runs. For instance, in Example 1, $d^{(14)}$ is nested in $d^{(17)}$ which in turn is nested in $d^{(19)}$. This feature is very useful in efficient sequential planning of experiments when the available resources are available in several stages.

References

Banerjee, T. and Mukerjee, R. (2008) "Optimal factorial designs for cDNA microarray experiments," *The Annals of Applied Statistics*, 2, 366-385.

Glonek, G.F.V. and Solomon, P.J. (2004) "Factorial and time course designs for cDNA microarray experiments," *Biostatistics*, 5, 89-111.

Gupta, S. and Mukerjee, R. (1989) *A Calculus for Factorial Arrangements*, Springer, Berlin.

Kerr, K.F. (2006) "Efficient 2^k factorial designs for blocks of size 2 with microarray applications," *Journal of Quality Technology*, 38, 309-318.

Kerr, K.F. (2012) "Optimality criteria for the design of 2-color microarray studies," *Statistical Applications in Genetics and Molecular Biology*, 11: Issue 1, Article 10.

Mukerjee, R. and Tang, B. (2012) "Optimal fractions of two-level factorials under a baseline parametrization," *Biometrika*, 99, 71-84.

Mukerjee, R. and Wu, C.F.J. (2006) *A Modern Theory of Factorial Designs*, Springer, New York.

Sanchez, P.S. and Glonek, G.F.V. (2009) "Optimal designs for 2-color microarray experiments," *Biostatistics*, 10, 561-574.

Silvey, S.D. (1980) *Optimal Design*, Chapman and Hall, London.

Torsney, B. and Martin-Martin, R. (2008) "Multiplicative algorithms for computing optimum designs," www.newton.ac.uk/programmes/DOE/seminars/081516001.pdf .

Wu, C.F.J. and Hamada, M.S. (2009) *Experiments: Planning, Analysis and Optimization* (2nd ed.), Wiley, Hoboken, New Jersey.

Xu, H., Phoa, F.K.H. and Wong, W.K. (2009) "Recent developments in nonregular fractional factorial designs," *Statistics Surveys*, 3, 18-46.

Yang, Y.H. and Speed, T. (2002) "Design issues for cDNA microarray experiments," *Nature Genetics* (*Supplement*), 3, 579-588.