

Optimal design generation: an approach based on discovery probability

*Original*

Optimal design generation: an approach based on discovery probability / Fontana, Roberto. - In: COMPUTATIONAL STATISTICS. - ISSN 0943-4062. - STAMPA. - 30:4(2015), pp. 1231-1244. [10.1007/s00180-015-0562-1]

*Availability:*

This version is available at: 11583/2588481 since: 2016-10-25T11:39:44Z

*Publisher:*

Springer Berlin Heidelberg

*Published*

DOI:10.1007/s00180-015-0562-1

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style. Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

#### **Please note**

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: [http://dx.doi.org/\[DOI\]](http://dx.doi.org/[DOI]).

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: <http://www.link.springer.com>.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

# Metadata of the article that will be visualized in OnlineFirst

---

**Please note: Images will appear in color online but will be printed in black and white.**

---

ArticleTitle	Optimal design generation: an approach based on discovery probability	
Article Sub-Title		
Article CopyRight	Springer-Verlag Berlin Heidelberg (This will be the copyright line in the final PDF)	
Journal Name	Computational Statistics	
Corresponding Author	Family Name	<b>Fontana</b>
	Particle	
	Given Name	<b>Roberto</b>
	Suffix	
	Division	Department of Mathematical Sciences
	Organization	Politecnico di Torino
	Address	Corso Duca degli Abruzzi, 24, 10129, Turin, Italy
	Email	roberto.fontana@polito.it
Schedule	Received	9 January 2014
	Revised	
	Accepted	22 January 2015
Abstract	Efficient algorithms for searching for optimal saturated designs for sampling experiments are widely available. They maximize a given efficiency measure (such as D-optimality) and provide an optimum design. Nevertheless, they do not guarantee a <i>global</i> optimal design. Indeed, they start from an initial random design and find a local optimal design. If the initial design is changed the optimum found will, in general, be different. A natural question arises. Should we stop at the design found or should we run the algorithm again in search of a better design? This paper uses very recent methods and software for discovery probability to support the decision to continue or stop the sampling. A software tool written in SAS has been developed.	
Keywords (separated by '-')	Design of experiments - Optimal designs - Unobserved species - Discovery probability	
Footnote Information		

---

# Optimal design generation: an approach based on discovery probability

Roberto Fontana

Received: 9 January 2014 / Accepted: 22 January 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Efficient algorithms for searching for optimal saturated designs for sampling experiments are widely available. They maximize a given efficiency measure (such as D-optimality) and provide an optimum design. Nevertheless, they do not guarantee a *global* optimal design. Indeed, they start from an initial random design and find a local optimal design. If the initial design is changed the optimum found will, in general, be different. A natural question arises. Should we stop at the design found or should we run the algorithm again in search of a better design? This paper uses very recent methods and software for discovery probability to support the decision to continue or stop the sampling. A software tool written in SAS has been developed.

**Keywords** Design of experiments · Optimal designs · Unobserved species · Discovery probability

## 1 Introduction

In the design of experiments, optimal designs, or optimum designs, are a class of experimental designs that are optimal with respect to a given statistical criterion.

In this paper we focus on saturated optimum designs for sampling experiments even if the methodology can also be applied to non-saturated designs without any modification. Saturated designs contain a number of points that is equal to the number of parameters of the model. It follows that saturated optimum designs are often used in place of standard designs, such as orthogonal fractional factorial designs, when

---

R. Fontana (✉)  
Department of Mathematical Sciences, Politecnico di Torino,  
Corso Duca degli Abruzzi, 24, 10129 Turin, Italy  
e-mail: roberto.fontana@polito.it

21 the cost of each experimental run is high. Main references to optimal designs include  
22 [Atkinson et al. \(2007\)](#), [Goos and Jones \(2011\)](#), [Pukelsheim \(2006\)](#), [Rasch et al. \(2011\)](#),  
23 [Shah and Sinha \(1989\)](#) and [Wynn \(1970\)](#).

24 The optimality of a design depends on the statistical model that is assumed and is  
25 assessed with respect to a statistical criterion, which, for information-based criteria,  
26 is related to the variance-matrix of the model parameter estimators. Well-known and  
27 commonly used criteria are A-optimality and D-optimality.

28 Widely used statistical systems like SAS and R have procedures for finding an  
29 optimal design according to the user's specifications. In this paper we will refer to  
30 Proc Optex of SAS/QC ([SAS Institute, Inc. 2010](#)), but the approach can be adopted  
31 for other software.

32 The Optex procedure searches for optimal experimental designs. The user specifies  
33 an efficiency criterion, a set of candidate design points, a model and the size of the  
34 design to be found and the procedure generates a subset of the candidate set so that the  
35 terms in the model can be estimated as efficiently as possible. By default, the standard  
36 output of the procedure is a list of 10 designs that are found as the result of 10 runs of  
37 the exchange search algorithm ([Mitchell and Miller 1970](#)) starting each time from an  
38 initial completely randomly chosen design.

39 The number of times that we decide to run the search algorithm is crucial. Obviously,  
40 if we increase it, in general we will explore different local optima with the possibility  
41 to find better designs. On the other hand, sometimes, the extra time that we use to  
42 explore other possibilities is wasted because new optima do not exist. This work aims  
43 at developing a methodology based on a Bayesian updating methodology that could  
44 support the user in making the decision whether to stop or continue the search.

45 Let us consider an example that will be described in more detail in Sect. 4.1. An  
46 experimenter wants to study the influence on a response  $Y$  (e.g. the fuel consumption  
47 of a given engine) of 7 factors (the type of fuel, the age of the engine, etc.) where  
48 each factor has 2 levels. The full factorial design has  $2^7 = 128$  runs. Let us suppose  
49 that both the size of the full factorial design is too high with respect to the available  
50 budget and the experimenter believes that a model with only the main effects and  
51 2-factor interactions would be sufficiently rich to describe with a good accuracy the  
52 phenomenon under study. A minimum size orthogonal fractional factorial design for  
53 this case requires 64 runs, which is still a high value ([Fontana 2013](#)). The experimenter  
54 decides to use a saturated  $D$ -optimal design (29 runs). The experimenter runs the Proc  
55 Optex procedure of SAS/QC with the default settings and gets a saturated  $D$ -optimal  
56 design with  $D$ -efficiency equal to 82.32. With the methodology described in this paper  
57 the experimenter would have been able to find both a better design ( $D$ -efficiency equal  
58 to 85.63) and a list of 103  $D$ -optimal designs that could be further analysed with respect  
59 to different criteria like space filling.

60 The paper is organized as follows. In Sect. 2 we state the problem of finding new  
61 optimal designs as the problem of finding new species in a population. Then, in Sect. 3  
62 we describe how our methodology, which is based on the estimator of the discovery  
63 probability, could be used for optimal design generation. We also provide a detailed  
64 description of the algorithm. In Sect. 4 we describe the results of a computational  
65 study in which we ran our algorithm in different cases. Concluding remarks are made  
66 in Sect. 5.

67 The software code that has been developed is written in SAS, is available on request  
 68 and can be used for any choice of factors, levels and model. It is worth noting that the  
 69 algorithm, being based on the Proc Optex procedure, can manage not only classical  
 70 linear model but also nonstandard linear or nonlinear models, (SAS Institute, Inc.  
 71 2010).

## 72 2 Optimal designs vs richness of species

73 We consider the following setting that is quite common in optimal design problems.

74 We have  $d$  factors,  $A_1, \dots, A_d$ . The factor  $A_i$  has  $s_i$  levels coded with the integer  
 75  $0, \dots, s_i - 1, i = 1, \dots, d$ . The full factorial design is  $\mathcal{D} = \{0, \dots, s_1 - 1\} \times \dots \times$   
 76  $\{0, \dots, s_d - 1\}$ . For each point  $\zeta = (\zeta_1, \dots, \zeta_d)$  of  $\mathcal{D}$  we consider a real-valued  
 77 random variable  $Y_{\zeta_1, \dots, \zeta_d}$ . We make the hypothesis that the means of the responses,  
 78  $E[Y]$  where  $Y$  is the column vector  $[Y_\zeta; \zeta \in \mathcal{D}]$  can be modeled as

$$79 \quad E[Y] = X_{\mathcal{D}}\beta, \quad (1)$$

80 where  $X_{\mathcal{D}}$  is the non-overparametrized design matrix, as it will be defined in Sect. 2.1,  
 81 and  $\beta$  is the subset of all the effects (constant effect, main effects and interactions)  
 82 that are supposed to affect the response  $Y$ . There is no restriction to the order of the  
 83 interactions; polynomial effects (linear, quadratic, etc) can also be considered.

84 Given an efficiency criterion  $\phi$ , a saturated optimal design is a subset of the full  
 85 factorial design  $\mathcal{D} = \{0, \dots, s_1 - 1\} \times \dots \times \{0, \dots, s_d - 1\}$ , whose size is equal to the  
 86 number of degrees of freedom of the model (1) and that maximizes this criterion  $\phi$ . In  
 87 this paper we focus on information-based criteria and, in particular, on  $D$ -optimality  
 88 but other criteria can be chosen (like  $A$ -optimality and  $G$ -optimality). We denote  
 89 this type of problem with the triple  $(\mathcal{D}, \mathcal{M}, \phi)$  where  $\mathcal{D}$  is the full design,  $\mathcal{M}$  is the  
 90 hypothesized model (see Eq. 1) and  $\phi$  is the optimality criterion.

91 Given a subset  $\mathcal{F}$  of  $\mathcal{D}$ , the information matrix is defined as  $X'_{\mathcal{F}}X_{\mathcal{F}}$  where  $X_{\mathcal{F}}$  is  
 92 the design matrix corresponding to  $\mathcal{F}$  and  $X'$  is the transpose of  $X$ .  $D$ -optimality aims  
 93 at maximizing  $D_{\mathcal{F}}$ , the determinant of the information matrix

$$94 \quad D_{\mathcal{F}} = \det(X'_{\mathcal{F}}X_{\mathcal{F}}). \quad (2)$$

95 There are several algorithms for searching for  $D$ -optimal designs. They have a com-  
 96 mon structure. They start from an initial design, randomly generated or user specified,  
 97 and move, in a finite number of steps, to a better design. In general, if a different initial  
 98 design is chosen, a different optimal design is found.

99 It follows that, given an algorithm  $\alpha$ , a population  $\mathcal{A}_{\alpha}^D$  of  $D$ -optimal designs can  
 100 be defined. This population is made up of all the saturated designs that are the result  
 101 of the execution of the algorithm  $\alpha$  and is a subset of all the subsets of  $\mathcal{D}$  of size equal  
 102 to the number of degrees of freedom of the model.

103 The elements of  $\mathcal{A}_{\alpha}^D$  can be classified into species, according to the criterion for  
 104 which  $\mathcal{F}_1 \in \mathcal{A}_{\alpha}^D$  and  $\mathcal{F}_2 \in \mathcal{A}_{\alpha}^D$  are of the same species if and only if they have the  
 105 same value in terms of the  $D$  criterion,  $D_{\mathcal{F}_1} = D_{\mathcal{F}_2}$ .

106 Studying the species of  $\mathcal{A}_\alpha^D$  or, in general, of  $\mathcal{A}_\alpha^\phi$  where  $\phi$  is an optimal criterion,  
 107 is interesting for optimal design generation. Let us consider the problem  $(\mathcal{D}, \mathcal{M}, \phi)$   
 108 and let us choose an algorithm  $\alpha$  to search for  $\phi$ -optimal saturated designs. If we run  
 109 this algorithm  $n$  times, each time starting from a completely random initial design,  
 110 we will get a sample of  $n$  elements of  $\mathcal{A}_\alpha^\phi$ . Such elements can be classified in  $k_n \leq$   
 111  $n$  different species according to the value of the criterion  $\phi$ . Recent methods for  
 112 discovery probability estimation, Favaro et al. (2012), can be applied to the vector  
 113  $(\ell_1, \ell_2, \dots, \ell_n)$  where  $\ell_r$  is the number of species in the sample with frequency  $r$ ,  
 114  $r = 1, \dots, n$ . In particular, based on a sample of size  $n$ , for any additional unobserved  
 115 sample size  $m \geq 0$  and for any frequency  $k = 0, \dots, n + m$ , these methods provide,  
 116 an explicit estimator for the probability  $U_{n+m}(k)$  that the  $(n + m + 1)$ -th observation  
 117 coincides with a species whose frequency, within the sample size  $n + m$ , is exactly  $k$ .  
 118 The case  $m = k = 0$  corresponds to assessing the probability of finding a new species  
 119 in the subsequent observation, that in the context of optimal designs, is the probability  
 120 of finding a saturated design with a different value of the criterion  $\phi$  in the subsequent  
 121 run of the algorithm. If this probability  $U_{n+0}(0)$  is sufficiently high (let us say greater  
 122 than 0.1 or even 0.05) it would be convenient to run the algorithm again because it  
 123 is quite likely that we could find a new optimal design. If we found a new design, it  
 124 could have a greater value of  $\phi$  and this obviously represents an improvement to our  
 125 optimization process. Even if this new design did not have an higher value of  $\phi$  than  
 126 the existing ones, this would give the possibility to increase the known part of  $\mathcal{A}_\alpha^\phi$ .  
 127 It is quite common, in practical applications, to choose a design where the optimal  
 128 criterion has a slightly smaller value than the maximum obtained but which has other  
 129 better characteristics, such as space filling properties.

130 In particular, for  $D$ -optimal designs, we know that designs with different values of  
 131  $D_{\mathcal{F}}$  are non-isomorphic designs. Indeed we observe that, as proved in Proposition 1,  
 132 see Angelopoulos et al. (2007), isomorphic designs belong to the same species. In  
 133 general, the opposite is not true because there are designs with the same value of the  
 134  $D$  criterion but that are not isomorphic. As is known two designs are isomorphic if  
 135 one can be obtained from the other by relabeling the factors, reordering the runs, and  
 136 switching the levels of factors, e.g. Clark and Dean (2001).

137 **Proposition 1** *Let us consider  $\mathcal{F}_1 \subseteq \mathcal{D}$  and  $\mathcal{F}_2 \subseteq \mathcal{D}$ . If  $\mathcal{F}_1$  and  $\mathcal{F}_2$  are isomorphic*  
 138 *then  $D_{\mathcal{F}_1} = D_{\mathcal{F}_2}$ .*

139 *Proof* We separately analyse row/column permutations and the switching of the levels  
 140 of some factors. If  $\mathcal{F}_2$  is obtained permuting the rows and/or the columns of  $\mathcal{F}_1$  it  
 141 follows that

$$142 \quad X_{\mathcal{F}_2} = R X_{\mathcal{F}_1} C$$

143 where  $R$  and  $C$  are permutation matrices. Then

$$144 \quad \begin{aligned} D_{\mathcal{F}_2} &= \det((X'_{\mathcal{F}_2} X_{\mathcal{F}_2})) \\ 145 \quad &= (\det(R))^2 \det((X'_{\mathcal{F}_1} X_{\mathcal{F}_1})) (\det(C))^2 \\ 146 \quad &= D_{\mathcal{F}_1} \end{aligned}$$

147 being  $\det(R) = \det(C) = 1$ . A similar argument holds for switching the levels of  
148 some factors.  $\square$

149 The knowledge of a set of non-isomorphic designs can also be used for non parametric  
150 testing procedures, [Giancristofaro et al. \(2012\)](#) and [Basso et al. \(2004\)](#).

151 In Sect. 2.1 and 2.2 we provide some details on how the design matrix is built and  
152 on how to compute the estimates of the discovery probabilities.

## 153 2.1 The design matrix

154 The design matrix  $X_{\mathcal{D}}$  in Eq. 1 is built as follows.

155 The first column is equal to 1 and corresponds to the constant effect, denoted by  $\mu$ .  
156 The constant effect is always considered as a term of the model.

157 If the main effect of the factor  $A_i$  is to be considered in the model, the corresponding  
158  $s_i - 1$  columns are computed as follows. For a design point with  $A_i$  at its  $k$ -th level if  
159  $1 \leq k \leq s_i - 1$  the columns are all 0 except for the  $k$ -th column that is 1; if  $k = s_i$  the  
160 columns are all  $-1$ .

161 If an interaction  $A_{i_1} \star \dots \star A_{i_k}$  is to be considered in the model, the corresponding  
162  $(s_{i_1} - 1) \cdot \dots \cdot (s_{i_k} - 1)$  columns are computed by taking the horizontal direct product  
163 of the columns corresponding to the main effects of  $A_{i_1}, \dots, A_{i_k}$ .

164 This coding corresponds to modeling without over parametrization and  $X_{\mathcal{D}}$  is full  
165 rank.

166 For a subset  $\mathcal{F}$  of  $\mathcal{D}$ , the design matrix  $X_{\mathcal{F}}$  is simply built deleting from  $X_{\mathcal{D}}$  the  
167 rows that correspond to the points of  $\mathcal{D}$  that are not in  $\mathcal{F}$ .

## 168 2.2 Discovery probability

169 We briefly summarize the main results that are used in this work, as in [Favaro et al.](#)  
170 [\(2012\)](#). The interested reader should refer to the original paper for a detailed descrip-  
171 tion of the methodology. We observe that the results in [Favaro et al. \(2012\)](#) are an  
172 improvement of those in [Lijoi et al. \(2007\)](#) concerning the evaluation of the probabil-  
173 ity that further sampling reveals new species.

174 Given a sample of size  $n$ , the vector  $(\ell_1, \dots, \ell_n)$  is built, where  $\ell_r$  is the frequency  
175 of species that have been observed  $r$ -times in the sample,  $r = 1, \dots, n$ . We have  
176  $\sum_{i=1}^n i \ell_i = n$ . We denote the number of different species that have been observed in  
177 the sample by  $j$ . We get  $\sum_{i=1}^n \ell_i = j$ .

178 Based on a sample of size  $n$ , for an additional unobserved sample size  $m \geq 0$  and for  
179 any frequency  $k = 0, \dots, n + m$ , using a non parametric Bayesian approach, [Favaro](#)  
180 [et al. \(2012\)](#) provide an estimator for the probability  $U_{n+m}(k)$  that the  $(n + m + 1)$ -th  
181 observation coincides with a species whose frequency, within the sample of size  $n + m$ ,  
182 is exactly  $k$ .

183 We are interested in discovering new species, that correspond to the case  $k = 0$ .

184 From Sect. 2, p.1,190 of [Favaro et al. \(2012\)](#) we obtain

$$185 \quad U_{n+0}(0) = \frac{V_{n+1,j+1}}{V_{n,j}}$$

186 where, for the two-parameter Poisson-Dirichlet process, we have  $V_{n,j} = \prod_{i=1}^{j-1} (\theta +$   
 187  $i\sigma)/(\theta + 1)_{n-1}$ ,  $\sigma \in (0, 1)$ ,  $\theta > -\sigma$ . The symbol  $(a)_n$  denotes the  $n$ -th ascending  
 188 factorial of  $a$ ,  $(a)_n = a(a + 1) \dots (a + n - 1)$ ,  $(a)_0 \equiv 1$ . It follows that

$$189 \quad U_{n+0}(0) = \frac{\theta + j\sigma}{\theta + n}$$

190 and, for  $m > 0$ , we obtain

$$191 \quad U_{n+m}(0) = \frac{\theta + j\sigma}{\theta + n} \frac{(\theta + n + \sigma)_m}{(\theta + n + 1)_m}.$$

192 The estimates  $\hat{\sigma}$ ,  $\hat{\theta}$  of  $\sigma$ ,  $\theta$  are obtained as

$$193 \quad \arg \max_{(\sigma, \theta)} \frac{\prod_{i=1}^{j-1} (\theta + i\sigma)}{(\theta + 1)_{n-1}} n! \prod_{i=1}^n \left\{ \frac{(1 - \sigma)_{i-1}}{i!} \right\}^{\ell_i} \frac{1}{\ell_i!}. \quad (3)$$

194 Using  $(\hat{\theta}, \hat{\sigma})$  we finally obtain the estimates of the discovery probability at the  
 195  $(n + 1)$ -th observation

$$196 \quad \hat{U}_{n+0}(0) = \frac{\hat{\theta} + j\hat{\sigma}}{\hat{\theta} + n} \quad (4)$$

197 and at the  $(n + m + 1)$ -th observation,  $m > 0$ ,

$$198 \quad \hat{U}_{n+m}(0) = \frac{\hat{\theta} + j\hat{\sigma}}{\hat{\theta} + n} \frac{(\hat{\theta} + n + \hat{\sigma})_m}{(\hat{\theta} + n + 1)_m} \quad (5)$$

### 199 3 Methodology

200 We repeat the search for optimal designs to analyse the population  $\mathcal{A}_\alpha^D$  of  $D$ -optimal  
 201 designs that can be found for a given problem using a predefined algorithm  $\alpha$ . Each  
 202 time the algorithm starts from a randomly chosen initial design. We set a maximum  
 203 (minimum) number of iterations equal to  $M_\star$  ( $m_\star$ ). We continue the process until the  
 204 minimum number  $m_\star$  of iterations is performed and the estimate of the discovery  
 205 probability at the subsequent observation goes under a given threshold  $p_\star$ , or until the  
 206 maximum number  $M_\star$  of iterations is reached.

207 The procedure can be described as follows. A problem  $(\mathcal{D}, \mathcal{M}, \phi)$ , with  $\phi = D$   
 208 in our examples, is defined and an algorithm  $\alpha$  for  $\phi$ -optimal design generation is  
 209 chosen. For each iteration  $s$ ,  $s = 1, \dots, M_\star$ ,

- 210 1. using the algorithm  $\alpha$ , a  $\phi$ -optimal saturated design  $\mathcal{F}_s$  is obtained;
- 211 2. the value of the  $\phi$ -criterion of  $\mathcal{F}_s$  is computed;
- 212 3. the vector  $(\ell_1, \dots, \ell_s)$  is built, where  $\ell_r$  is the number of species with frequency  
 213  $r$ ,  $r = 1, \dots, s$ ;
- 214 4. an estimate  $(\hat{\sigma}_s, \hat{\theta}_s)$  is obtained, see Eq. 3;
- 215 5. an estimate of  $\hat{U}_{s+0}(0)$  is computed using Eq. 4;

216 6. if  $\hat{U}_{s+0}(0) < p_*$  and  $s \geq m_*$  the algorithm stops, otherwise the next iteration  $s + 1$   
 217 is performed (if  $s + 1 > M_*$  the algorithm stops).

218 The main output of the algorithm is a set of designs, where each design belongs to  
 219 a different species, i.e. has a different value of the  $\phi$ -criterion.

220 We now provide a detailed description of each step of the algorithm.

221 3.1 Steps 1 and 2

222 At iteration  $s$ , with the chosen algorithm  $\alpha$ , the Proc Optex procedure is used to  
 223 generate a  $D$ -optimal design,  $\mathcal{F}_s$ . The species of  $\mathcal{F}_s$  is the value of its  $D$ -efficiency,  
 224  $E_{\mathcal{F}_s}^D$ . The  $D$ -efficiency of a  $\mathcal{F}$ , is defined as

225 
$$E_{\mathcal{F}}^D = 100 \times \left( \frac{1}{\#\mathcal{F}} D_{\mathcal{F}}^{\frac{1}{\#\mathcal{F}}} \right)$$

226 where  $\#\mathcal{F}$  is the number of runs of  $\mathcal{F}$  that coincides with the degrees of freedom of  
 227 the model for saturated designs and  $D_{\mathcal{F}}$  is the determinant of the information matrix.

228 The value of the efficiency is rounded to four decimal digits to avoid creating  
 229 different species from numerical effects.

230 3.2 Step 3

231 Using all the designs  $\mathcal{F}_1, \dots, \mathcal{F}_s$  with their corresponding  $D$ -efficiencies,  $E_{\mathcal{F}_1}^D, \dots,$   
 232  $E_{\mathcal{F}_s}^D$  the vector  $(\ell_1, \dots, \ell_s)$  is built, where  $\ell_r$  is the number of species with frequency  
 233  $r, r = 1, \dots, s$ .

234 3.3 Step 4

235 An estimate  $(\hat{\sigma}_s, \hat{\theta}_s)$  must be obtained searching for  $(\sigma, \theta), \sigma \in (0, 1), \theta > -\sigma$  that  
 236 maximizes  $f(\sigma, \theta)$ , (see Eq. 3),

237 
$$f(\sigma, \theta) = \frac{\prod_{i=1}^{j-1} (\theta + i\sigma)}{(\theta + 1)_{n-1}} n! \prod_{i=1}^n \left\{ \frac{(1 - \sigma)_{i-1}}{i!} \right\}^{\ell_i} \frac{1}{\ell_i!}$$

238 The Genetic Algorithm module of SAS/IML has been used. In order to manage  
 239 the constraints  $\sigma \in (0, 1), \theta > -\sigma$  the search has been performed in the region  
 240  $\mathcal{R} = [\delta, 1 - \delta] \times [-(1 - \delta), T_M]$  with  $\delta = 0.01$  and  $T_M = 1,000$ . This region contains  
 241 the non-feasible region made by the points inside the simplex  $\mathcal{S} = \mathcal{R} \cap \{(\sigma, \theta) : \theta \leq$   
 242  $-\sigma\}$  whose vertices are  $(\delta, -(1 - \delta)), (\delta, -\delta)$  and  $(1 - \delta, -(1 - \delta))$ . We observe that  
 243 the edges of  $\mathcal{S}$  contain non-feasible points.

244 We decided to manage this constraint with the penalty method, because this method  
 245 usually works well when most of the points in the solution space do not violate the  
 246 constraints, as in our problem. The way in which the penalty in the objective function  
 247 for unsatisfied constraints has been imposed is described here.

248 From the point of view of the search of the point  $(\sigma_*, \theta_*)$  that maximizes  $f(\sigma, \theta)$ ,  
 249 it is equivalent to consider  $\log f(\sigma, \theta)$  instead of  $f(\sigma, \theta)$

$$250 \quad \log f(\sigma, \theta) = \log \left( \prod_{i=1}^{j-1} (\theta + i\sigma) \right) + \log(n!) \\
 251 \quad - \log((\theta + 1)_{n-1}) + \log \left( \prod_{i=1}^n \left\{ \frac{(1-\sigma)_{i-1}}{i!} \right\}^{\ell_i} \right) - \log(\ell_i!).$$

252 Omitting the terms that do not depend on  $\sigma$  and  $\theta$  and as  $(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)}$  where  $\Gamma$  is  
 253 the gamma function, the previous equation becomes the function  $f_*(\sigma, \theta)$  here

$$254 \quad f_*(\sigma, \theta) = f_*^{(1)}(\sigma, \theta) + f_*^{(2)}(\sigma, \theta),$$

255 where

$$256 \quad f_*^{(1)}(\sigma, \theta) = \sum_{i=1}^{j-1} f_*^{(1,i)}(\sigma, \theta)$$

257 with  $f_*^{(1,i)}(\sigma, \theta) = \log(\theta + i\sigma)$  and

$$258 \quad f_*^{(2)}(\sigma, \theta) = -\log \Gamma(\theta + n) + \log \Gamma(\theta + 1) \\
 259 \quad + \sum_{i=1}^n \ell_i \log \Gamma(i - \sigma) - j \log \Gamma(1 - \sigma).$$

260 We observe that, when the point  $(\sigma, \theta) \in \mathcal{R}$  does not satisfy the constraint  $\theta > -\sigma$   
 261 only  $f_*^{(1)}(\sigma, \theta)$  becomes not defined. We apply a penalty value to  $f_*^{(1)}(\sigma, \theta)$  and to  
 262  $f_*^{(2)}(\sigma, \theta)$  as described below.

263 Given a point  $P_1$  in the non-feasible region,  $P_1 = (\sigma, \theta) \in \mathcal{S}$ ,  $\tilde{P}_1$ , the closest  
 264 point to  $P_1$  with respect to the Euclidean distance that lies in the feasible region, is  
 265 determined

$$266 \quad \tilde{P}_1 = (\tilde{\sigma}, \tilde{\theta}) = \left( \frac{1}{2}(\sigma - \theta + \epsilon), \frac{1}{2}(\theta - \sigma + \epsilon) \right)$$

267 where  $\epsilon$  is a very small number to ensure that  $\tilde{P}_1$  is feasible, i.e.  $\tilde{P}_1 \in \mathcal{R} \cap \bar{\mathcal{S}}$ . We  
 268 used  $\epsilon = 0.001$ . The value of the function  $f_*^{(1,1)}$  is computed in  $\tilde{P}_1$  getting  $\tilde{Y}_1 =$   
 269  $f_*^{(1,1)}(\tilde{\sigma}, \tilde{\theta}) = \log \epsilon$ . Then the value  $Y_1$  of  $f_*^{(1,1)}$  in  $P_1$  is defined as  $f_*^{(1,1)}(\sigma, \theta) = (1 +$   
 270  $b_1)\tilde{Y}_1$  where  $b_1$  is the Euclidean distance between  $P_1$  and  $\tilde{P}_1$ ,  $b_1 = \sqrt{\frac{1}{2}(\sigma + \theta - \epsilon)^2}$ .  
 271 In an analogous way, we apply this penalty method to all  $P_i = (i\sigma, \theta)$  that even-  
 272 tually fall in the non-feasible region  $\mathcal{S}$  getting  $f_{*,p}^{(1)}(\sigma, \theta)$ , the penalized version of  
 273  $f_*^{(1)}(\sigma, \theta)$ ,

$$f_{\star, P}^{(1)}(\sigma, \theta) = \sum_{i=1}^{j-1} f_{\star, P}^{(1,i)}(\sigma, \theta)$$

where

$$f_{\star, P}^{(1,i)} = \begin{cases} \log(\theta + i\sigma) & \text{if } \theta + i\sigma > 0 \\ (1 + b_i) \log(\epsilon) & \text{if } \theta + i\sigma \leq 0 \end{cases}, i = 1, \dots, j - 1,$$

and  $b_i$  is the Euclidean distance between  $P_i = (i\sigma, \theta)$  and  $\tilde{P}_i = (\frac{1}{2}(i\sigma - \theta + \epsilon), \frac{1}{2}(\theta - i\sigma + \epsilon))$  determined as described above. The penalized version  $f_{\star, P}^{(2)}(\sigma, \theta)$  of  $f_{\star}^{(2)}(\sigma, \theta)$  is simply defined as

$$f_{\star, P}^{(2)}(\sigma, \theta) = \begin{cases} f_{\star}^{(2)}(\sigma, \theta) & \text{if } \theta + \sigma > 0 \\ (1 + b_1)f_{\star}^{(2)}(\sigma, \theta) & \text{if } \theta + \sigma \leq 0 \\ & \text{and } f_{\star}^{(2)}(\sigma, \theta) \leq 0 \\ (1 - b_1)f_{\star}^{(2)}(\sigma, \theta) & \text{if } \theta + \sigma \leq 0 \\ & \text{and } f_{\star}^{(2)}(\sigma, \theta) > 0 \end{cases}$$

We observe that

$$\begin{cases} p < q \Rightarrow b_p > b_q, p, q = 1, \dots, j - 1; \\ b_1 \leq \frac{\sqrt{2}}{2}(1 + \epsilon - 2\delta). \end{cases}$$

For  $\delta = 0.01$  and  $\epsilon = .001$  we get  $b_1 < 0.694$ .

Using the penalty method, an estimate  $(\hat{\sigma}_s, \hat{\theta}_s)$  is obtained finding the maximum of  $f_{\star, P}(\sigma, \theta) = f_{\star, P}^{(1)}(\sigma, \theta) + f_{\star, P}^{(2)}(\sigma, \theta)$ .

Finally we point out that if  $n = 1$  then  $f_{\star}(\sigma, \theta) = 0 \forall \sigma, \theta$ . It follows that  $n$  must be greater than 1 to obtain the estimates of  $\sigma$  and  $\theta$ . If  $j = 1$  then  $\ell_1 = \dots = \ell_{n-1} = 0$ ,  $\ell_n = 1$ ,  $f_{\star}(\sigma, \theta) \equiv f_{\star}^{(2)}(\sigma, \theta)$  and

$$f_{\star}^{(2)}(\sigma, \theta) = -\log \Gamma(\theta + n) + \log \Gamma(\theta + 1) + \log \Gamma(n - \sigma) - \log \Gamma(1 - \sigma).$$

In this case we get  $\hat{\sigma} = \delta$  and  $\hat{\theta} = -\delta + \epsilon$ .

### 3.4 Steps 5 and 6

The estimate of the discovery probability at the next iteration,  $\hat{U}_{s+0}(0)$ , is computed as described in Sect. 2, Eq 4. If its value is lower than  $p_{\star}$  and  $s \geq m_{\star}$  the algorithm stops, otherwise the next iteration  $s + 1$  is performed (if  $s + 1 > M_{\star}$  the algorithm stops). The algorithm takes a decision only if  $s \geq m_{\star}$  because we want to avoid that the estimates  $(\hat{\sigma}_s, \hat{\theta}_s)$  and consequently  $\hat{U}_{s+0}(0)$  be based on too small sample sizes. We suggest using  $m_{\star}$  at least equal to 50.

**Table 1** Test cases description; ID is the test case number,  $d$  is the number of factors,  $p$  is the number of levels of each factor,  $\mathcal{M}$  is the model and Method is the algorithm that is used for  $D$ -optimal design generation. The notation  $x_1 | \dots | x_d @ h$  means that all the  $k$ -factor interactions,  $k \leq h$ , are included in the model  $\mathcal{M}$ ;  $k = 1$  refers to main effects

ID	$d$	$p$	$\mathcal{M}$	Method
1	7	2	$x_1   \dots   x_7 @ 2$	Exchange
2	7	2	$x_1   \dots   x_7 @ 2$	Fedorov
3	6	2	$x_1   \dots   x_6 @ 1$	Exchange
4	6	2	$x_1   \dots   x_6 @ 1$	Fedorov
5	5	3	$x_1   \dots   x_5 @ 2$	Exchange

**Table 2** Number  $\ell_r$  of  $D$  optimal designs that have found  $r$  times,  $r = 1, \dots, 487$ ; only  $\ell_r \neq 0$  are shown

$r$	1	2	3	4	5	6	9	10	11	12	14	15	16	17	20	35	39	40	45	$T$
$\ell_r$	48	17	8	10	1	4	1	1	1	1	2	2	1	1	1	1	1	1	1	103

## 4 Computational study

We show how the methodology works using the test cases summarized in Table 1. We point out that all the test cases consider the problem of finding saturated  $D$ -optimal designs.

### 4.1 Test cases 1 and 2

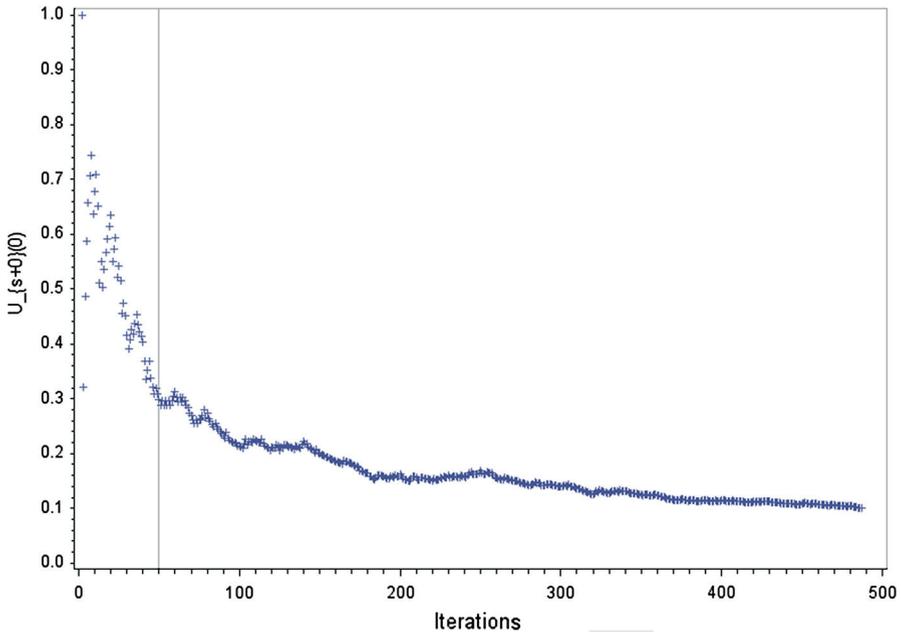
Let us consider 7 factors, each with 2 levels, and the model that contains the overall mean, the main effects and all the 2-factor interactions for a total of  $1 + 7 + 21 = 29$  degrees of freedom. We search for *saturated*  $D$ -optimal designs, that is,  $D$ -optimal designs that contain 29 points.

In test case 1, we use Proc Optex [SAS Institute, Inc. \(2010\)](#) with the exchange method, which is its default search method. With the default setting, the algorithm starts from 10 initial randomly chosen designs providing 10  $D$ -optimal designs. We consider the design with the highest value of the  $D$ -efficiency of the 10 optimal designs as the optimal design found by the algorithm.

Setting the seed that is used for the random generation of the initial designs at 6789, the best of the 10 optimal designs, that we denote by  $\mathcal{F}_1$ , has  $D_{\mathcal{F}_1} = 9.0911E39$  and  $E_{\mathcal{F}_1}^D = 82.3162$ .

Now we run the procedure above with  $M_\star = 1,000$ ,  $m_\star = 50$  and  $p_\star = 0.10$ . After 487 runs, the estimate of the discovery probability at the next observation becomes lower than  $p_\star = 0.10$  and the algorithm stops ( $\hat{U}_{487+0}(0) \approx 0.099$ ). We find 103 different species of local  $D$ -optimal designs. All these designs are not isomorphic (Proposition 1). The maximum (minimum) value of  $D$ -efficiency is 85.6265 (78.9605) and it has been found 9 times (1 time). The statistics  $\ell_r$ ,  $r = 1, \dots, 487$  are shown in Table 2.

The estimates of the discovery probability at the next iteration  $\hat{U}_{s+0}(0)$  as a function of the iteration  $s$  are plotted in Figure 1. The increase of the sample size seems clearly to stabilize the discovery probability estimates.



**Fig. 1** Estimate of the discovery probability at the next iteration  $\hat{U}_{s+0}(0)$  as a function of the iteration  $s$ . Test case 1,  $M_\star = 1,000$ ,  $m_\star = 50$  and  $p_\star = 0.10$

325 We decide to continue the search for new species choosing  $p_\star = 0.05$  and  
 326  $M_\star = 2,000$ . The latter value is chosen taking into account that using Eq. 5 we  
 327 get  $\hat{U}_{487+1000}(0) = 0.048$  and  $\hat{U}_{487+2000}(0) = 0.034$ . These supplementary runs are  
 328 added to the previous ones.

329 After 1,252 supplementary runs the estimate of the discovery probability at the  
 330 next observation becomes lower than 0.05,  $\hat{U}_{1739+0}(0) \approx 0.0499$ . After 1,252 +  
 331 487 = 1,739 simulations we observe 191 different species of  $D$ -optimal designs. The  
 332 maximum value of  $D$ -efficiency is still 85.6265, while the minimum is 78.1134.

333 In test case 2 we use the Fedorov algorithm, Fedorov (1972), that is considered more  
 334 reliable, even if slower, than the exchange algorithm. We keep the standard setting for  
 335 which, at each iteration, 10 local  $D$ -optimal designs are generated and the one among  
 336 them that has the highest  $D$ -efficiency value is taken as the optimal design.

337 We choose 3456 as the initial seed. The first iteration provides an optimal design  
 338  $\mathcal{F}_1$  with  $E_{\mathcal{F}_1}^D = 82.7079$ . Now we repeat the procedure with  $M_\star = 1,000$  and  $p_\star =$   
 339 0.10. After only 18 iterations the estimate of the discovery probability at the next  
 340 observation becomes less than 0.10,  $\hat{U}_{18+0}(0) \approx 0.097$ . But being  $m_\star = 50$  the  
 341 algorithm continues to iterate. It stops after 50 iterations, with the discovery probability  
 342 approximately equal to 2% and with 4 designs. The maximum (minimum) value of  
 343  $D$ -efficiency is 83.9844 (82.4212). Thus we have empirical evidence that the Fedorov  
 344 algorithm is more stable than the exchange algorithm. We observe that the best design  
 345 found with the exchange algorithm, that has  $D$ -efficiency equal to 85.6265, is not  
 346 found in this first sample. We were able to find it running the algorithm again with  
 347  $M_\star = 1,000$  and  $p_\star = 0.01$ . In this case after 47 supplementary runs, 97 in total, we

348 find 5 designs and  $\hat{U}_{97+0}(0) = 0.0099$ . The maximum value of  $D$ -efficiency becomes  
 349 85.6265. If we run the algorithm again with  $M_\star = 2,000$  and  $p_\star = 0.001$  after 1,009  
 350 supplementary runs, 1,106 in total we find 8 designs and  $\hat{U}_{1106+0}(0) = 0.00099$ . The  
 351 maximum (minimum) value of  $D$ -efficiency is 85.6265 (82.3622).

#### 352 4.2 Test cases 3 and 4

353 Let us consider 6 factors, each with 2 levels, and the model that contains the overall  
 354 mean and the main effects for a total of  $1 + 6 = 7$  degrees of freedom. We search for  
 355  $D$ -optimal designs that contains 7 points.

356 In test case 3 we use the same  $D$ -optimal design generation method of test case 1  
 357 (i.e. at each iteration the optimal design is the best design among ten optimal designs  
 358 found using the exchange method starting from ten initial randomly chosen designs).  
 359 We run our algorithm setting the initial seed to 6116 and with  $M_\star = 1,000$ ,  $m_\star = 50$   
 360 and  $p_\star = 0.10$ . After 50 iterations the algorithm stops with  $\hat{U}_{50+0}(0) \approx 0.005$  and  
 361 has found two classes of designs with  $D$ -efficiency equal to 84.91 and 87.92.

362 After 50 iterations the estimate of the discovery probability at the next iteration is  
 363 already quite small, around 0.5% but we decide to run a maximum of  $M_\star = 1,000$   
 364 supplementary runs setting  $p_\star = 0.0001$ . After a total of 1,050 runs the algorithm  
 365 stops. No designs with different  $D$ -optimal efficiency are found. The estimate of the  
 366 discovery probability at the next iteration is  $\hat{U}_{1050+0}(0) \approx 0.00014$ .

367 In test case 4 we replace the exchange algorithm with the Fedorov method. After 50  
 368 iterations we find only one class of optimal designs with  $D$ -efficiency equal to 87.92.  
 369 The estimate of the discovery probability at the next iteration is  $\hat{U}_{51+0}(0) \approx 0.00002$ .

#### 370 4.3 Test case 5

371 Let us now consider 5 factors, each with 3 levels, and the model that contains the overall  
 372 mean, the main effects and all the 2-factor interactions for a total of  $1 + 5 \cdot 2 + 10 \cdot 4 = 51$   
 373 degrees of freedom. We search for  $D$ -optimal designs that contain 51 points.

374 If we run the algorithm with the setting of test case 1 or 3 ( $M_\star = 1,000$ ,  $m_\star = 50$ ,  
 375  $p_\star = 0.10$ , method=exchange) after 1,000 iterations we find  $\hat{U}_{1000+0}(0) \approx 95\%$   
 376 and 978 different  $D$ -optimal designs with efficiencies ranging between 25.4333 and  
 377 28.6677.

378 In this case it can be appropriate to round the efficiency values not to four decimal  
 379 digits (as in all the previous test cases) but to one decimal digit. If we adopt this  
 380 rounding, after 3,600 iterations we get 33 different designs with  $D$ -efficiencies ranging  
 381 between a minimum of 25.2 to a maximum of 28.7. The estimate of the discovery  
 382 probability at the next iteration is  $\hat{U}_{3600+0}(0) \approx 0.0014$  ( $\hat{U}_{3600+1000}(0) \approx 0.0011$ ).

#### 383 4.4 Practical considerations and guidelines

384 After  $s$  iterations the algorithm provides  $\hat{U}_{s+0}(0)$ , an estimate of the probability of  
 385 discovery of a new value of efficiency at the next iteration. This value is useful to assess  
 386 how far the set of optimal designs that has been collected up to the iteration  $s$  is repre-

387 sentative of *all* the optimal designs  $\mathcal{A}_\alpha^\phi$ .  $1 - \hat{U}_{s+0}(0)$  estimates the *sample coverage*, that  
 388 is the proportion of distinct species present in the sample observed with respect to the  
 389 total population. If the probability of discovery is judged to be still too high or, equiv-  
 390 alently, the sample coverage too low, the user can decide to run the algorithm again.

391 From a practical point of view, the ideal situation is when the computational budget  
 392 of the user,  $M_\star$ , is large enough to reach a very high coverage, let us say of around  
 393 99.9% (i.e.  $\hat{U}_{s+0}(0) \approx 0.1\%$ ). In practice, in this ideal case, the algorithm can be run  
 394 for the first time with  $p_\star = 0.001$ ,  $M_\star = 1,000$  and  $m_\star = 50$ . If after the first 1,000  
 395 iterations the estimate of the discovery probability at the next iteration has not gone  
 396 under  $p_\star$  the algorithm can be run again to obtain supplementary observations. The  
 397 value of  $M_\star$  can be chosen computing the estimates of the discovery probability at  
 398 the  $(1000 + m + 1)$ -th observation with different values  $m$  of the size of an additional  
 399 unobserved sample size.

400 In any case, even if  $M_\star$  is not large enough to make the estimate of the discovery  
 401 probability at the next iteration as small as desired, the user has valuable information  
 402 about the sample coverage that has been reached so far. Again, an estimate of the dis-  
 403 covery probability at the next  $(n + m + 1)$ -th observation can be computed. It is useful  
 404 to consider this estimate for some values of  $m$ , let us say  $m = 1,000$  and  $m = 2,000$ ,  
 405 to make the decision to continue or stop the sampling.

406 It is important that the number of simulations is not too small to make the estimates  
 407 of the discovery probabilities sufficiently stable. We chose to work with  $m_\star = 50$ . To  
 408 support the choice of  $m_\star$  the plot of the estimate of the discovery probability  $\hat{U}_{s+0}(0)$   
 409 as function of the iteration  $s$  is useful (see Fig. 1).

410 The values of the efficiency must be rounded to avoid creating different species from  
 411 numerical effects. Different values of rounding allow us to reduce or to enlarge the  
 412 number of species of the population under study. This reflects the user's opinion regard-  
 413 ing the difference between efficiency values that must be considered significant from a  
 414 practical point of view. For example, at the initial stage of our exploration we can round  
 415 the values of the efficiency to one decimal digit just to know their approximate range.  
 416 Then in the next stages we can decide to run the algorithm with more decimal digits.

417 We ran the simulation study on a standard laptop (CPU Intel Core i7-2620M CPU  
 418 2.70 GHz 2.70 GHz, RAM 8 Gb). To give an idea of the computational times required  
 419 we report that the first stage of test case 1 (487 iterations, exchange method) needed  
 420 around 387 seconds (1.26 second per iteration) while the the first stage of test case 2  
 421 (50 iterations, Fedorov method) needed around 27 seconds (1.85 second per iteration).

## 422 5 Conclusion

423 Given an optimality criterion  $\phi$ , the problem of  $\phi$ -optimal design generation has been  
 424 addressed. A methodology to support the decision whether to continue or stop the  
 425 search for optimal designs has been developed. It combines recent advances on dis-  
 426 covery probability estimation, based on a Bayesian non parametric approach, Favaro  
 427 et al. (2012), with well known methods for optimal design generation.

428 In principle, this methodology could be applied to any discrete optimisation prob-  
 429 lem. This topic will be part of future research.

A software code, written in SAS, that makes use of the Proc Optex procedure, has been developed.

It should also be pointed out that the innovative side of this work lies in using sampling stopping rules to improve the generation process of optimal designs. In this paper we used the Bayesian updating of the discovery probability as proposed by Favaro et al. (2012) but other approaches could be adopted. For example Christen and Nakamura (2003) developed an algorithm based on backward induction. It makes use of a utility function based on the number of new species to be observed and the effort saved from the maximum horizon for accumulation. It could be part of future research using this algorithm in the context of optimal designs generation.

**Acknowledgments** I would like to thank both Mauro Gasparini (Politecnico di Torino), Gasparini (2012), and Giovanni Pistone (Collegio Carlo Alberto, Moncalieri, Torino) for the helpful discussions I had with them. A preliminary version of the work has been presented at the workshop on Model-Oriented Data Analysis and Optimum Design (MODA10, June 2013, Poland). The author wishes to thank the Editor and Referee for the accurate revision that helped produce a clearer version of the work.

## References

- Angelopoulos P, Evangelaras H, Koukouvinos C, Lappas E (2007) An effective step-down algorithm for the construction and the identification of nonisomorphic orthogonal arrays. *Metrika* 66(2):139–149
- Atkinson AC, Donev AN, Tobias RD (2007) Optimum experimental designs, with SAS. Oxford University Press, New York
- Basso D, Salmaso L, Evangelaras H, Koukouvinos C (2004) Nonparametric testing for main effects on inequivalent designs. In: Bucchianico A, Luter H, Wynn H (eds) mODa 7 advances in model-oriented design and analysis contributions to statistics. Physica-Verlag, HD, pp 33–40. doi:10.1007/978-3-7908-2693-7\_4
- Christen JA, Nakamura M (2003) Sequential stopping rules for species accumulation. *J Agric Biol Environ Stat* 8(2):184–195
- Clark JB, Dean A (2001) Equivalence of fractional factorial designs. *Stat Sin* 11(2):537–548
- Favaro S, Lijoi A, Prunster I (2012) A new estimator of the discovery probability. *Biometrics* 68(4):1188–1196. doi:10.1111/j.1541-0420.2012.01793.x
- Fedorov VV (1972) Theory of optimal experiments. Access Online via Elsevier
- Fontana R (2013) Algebraic generation of minimum size orthogonal fractional factorial designs: an approach based on integer linear programming. *Comput Stat* 28(1):241–253
- Gasparini M (2012) Mixtures and limits of symmetric random integer partitions. *Metron LXX(2–3):1–11*
- Giancristofaro RA, Fontana R, Ragazzi S (2012) Construction and nonparametric testing of orthogonal arrays through algebraic strata and inequivalent permutation matrices. *Commun Stat Theory Methods* 41(16–17):3162–3178. doi:10.1080/03610926.2011.579380
- Goos P, Jones B (2011) Optimal design of experiments: a case study approach. Wiley.com
- Lijoi A, Mena RH, Prünster I (2007) Bayesian nonparametric estimation of the probability of discovering new species. *Biometrika* 94(4):769–786
- Mitchell TJ, Miller Jr F (1970) Use of design repair to construct designs for special linear models. *Math Div Ann Progr Rept(ORNL-4661)* pp 130–131
- Pukelsheim F (2006) Optimal design of experiments, vol 50. Society for Industrial Mathematics, New York
- Rasch D, Pilz J, Verdooren L, Gebhardt A (2011) Optimal experimental design with R. Taylor & Francis, US
- SAS Institute, Inc (2010) SAS/QC 9.2 User's Guide, 2nd edn. SAS Institute Inc, Cary
- Shah KR, Sinha BK (1989) Theory of optimal designs, vol 582. Springer-Verlag, New York
- Wynn HP (1970) The sequential generation of  $d$ -optimum experimental designs. *Ann Math Stat* 41(5):1655–1664

Journal: 180  
Article: 562

## Author Query Form

**Please ensure you fill out your response to the queries raised below  
and return this form along with your corrections**

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Please check and confirm the inserted publisher location.	

uncorrected proof