Optimal design generation: an approach based on discovery probability

Springer

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].
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	**Fontana**
	Particle	
	Given Name	**Roberto**
	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 *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 (separated by '-') Design of experiments - Optimal designs - Unobserved species - Discovery probability

Footnote Information

Comput Stat
DOI 10.1007/s00180-015-0562-1

ORIGINAL PAPER

Optimal design generation: an approach based on discovery probability

Roberto Fontana

Received: 9 January 2014 / 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 *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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

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

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

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

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

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

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

2 Optimal designs vs richness of species

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

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

$$E[Y] = X_{\mathcal{D}}\beta\,, \tag{1}$$

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

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

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 the design matrix corresponding to $\mathcal{F}$ and X' is the transpose of X. D-optimality aims at maximizing $D_{\mathcal{F}}$, the determinant of the information matrix

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

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

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

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

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

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 then $D_{\mathcal{F}_1} = D_{\mathcal{F}_2}$.*

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

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

where R and C are permutation matrices. Then

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

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

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

2.1 The design matrix

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

The first column is equal to 1 and corresponds to the constant effect, denoted by μ. The constant effect is always considered as a term of the model.

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

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

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

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

2.2 Discovery probability

We briefly summarize the main results that are used in this work, as in Favaro et al. (2012). The interested reader should refer to the original paper for a detailed description of the methodology. We observe that the results in Favaro et al. (2012) are an improvement of those in Lijoi et al. (2007) concerning the evaluation of the probability that further sampling reveals new species.

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

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

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

From Sect. 2, p.1,190 of Favaro et al. (2012) we obtain

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

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

and, for $m > 0$, we obtain

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

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

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

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

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

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

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

3 Methodology

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

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

1. using the algorithm α, a ϕ-optimal saturated design $\mathcal{F}_s$ is obtained;
2. the value of the ϕ-criterion of $\mathcal{F}_s$ is computed;
3. the vector $(\ell_1, \ldots, \ell_s)$ is built, where ℓ_r is the number of species with frequency r, $r = 1, \ldots, s$;
4. an estimate $(\hat{\sigma}_s, \hat{\theta}_s)$ is obtained, see Eq. 3;
5. an estimate of $\hat{U}_{s+0}(0)$ is computed using Eq. 4;

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

6. if $\hat{U}_{s+0}(0) < p_\star$ 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 main output of the algorithm is a set of designs, where each design belongs to a different species, i.e. has a different value of the ϕ-criterion.

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

3.1 Steps 1 and 2

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

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

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

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

3.2 Step 3

Using all the designs $\mathcal{F}_1, \ldots, \mathcal{F}_s$ with their corresponding D-efficiencies, $E^D_{\mathcal{F}_1}, \ldots, E^D_{\mathcal{F}_s}$ the vector $(\ell_1, \ldots, \ell_s)$ is built, where ℓ_r is the number of species with frequency r, $r = 1, \ldots, s$.

3.3 Step 4

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

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

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

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

From the point of view of the search of the point $(\sigma_\star, \theta_\star)$ that maximizes $f(\sigma, \theta)$, it is equivalent to consider $\log \mathrm{f}(\sigma, \theta)$ instead of $\mathrm{f}(\sigma, \theta)$

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

Omitting the terms that do not depend on σ and θ and as $(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)}$ where Γ is the gamma function, the previous equation becomes the function $\mathrm{f}_\star(\sigma, \theta)$ here

$$\mathrm{f}_\star(\sigma, \theta) = \mathrm{f}_\star^{(1)}(\sigma, \theta) + \mathrm{f}_\star^{(2)}(\sigma, \theta),$$

where

$$\mathrm{f}_\star^{(1)}(\sigma, \theta) = \sum_{i=1}^{j-1} \mathrm{f}_\star^{(1,i)}(\sigma, \theta)$$

with $\mathrm{f}_\star^{(1,i)}(\sigma, \theta) = \log(\theta + i\sigma)$ and

$$\mathrm{f}_\star^{(2)}(\sigma, \theta) = -\log\Gamma(\theta + n) + \log\Gamma(\theta + 1) + \sum_{i=1}^{n} \ell_i \log\Gamma(i - \sigma) - j\log\Gamma(1-\sigma)\,.$$

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

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

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

where ϵ is a very small number to ensure that $\tilde{P}_1$ is feasible, i.e. $\tilde{P}_1 \in \mathcal{R} \cap \overline{\mathcal{S}}$. We used $\epsilon = 0.001$. The value of the function $\mathrm{f}_\star^{(1,1)}$ is computed in $\tilde{P}_1$ getting $\tilde{Y}_1 = \mathrm{f}_\star^{(1,1)}(\tilde{\sigma}, \tilde{\theta}) = \log\epsilon$. Then the value Y_1 of $\mathrm{f}_\star^{(1,1)}$ in P_1 is defined as $\mathrm{f}_\star^{(1,1)}(\sigma, \theta) = (1 + b_1)\tilde{Y}_1$ where b_1 is the Euclidean distance between P_1 and $\tilde{P}_1$, $b_1 = \sqrt{\frac{1}{2}(\sigma + \theta - \epsilon)^2}$. In an analogous way, we apply this penalty method to all $P_i = (i\sigma, \theta)$ that eventually fall in the non-feasible region $\mathcal{S}$ getting $\mathrm{f}_{\star,P}^{(1)}(\sigma, \theta)$, the penalized version of $\mathrm{f}_\star^{(1)}(\sigma, \theta)$,

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

where

$$\mathrm{f}^{(1,i)}_{\star,P}=\begin{cases}\log(\theta+i\sigma) & \text{if } \theta+i\sigma>0\\ (1+b_i)\log(\epsilon) & \text{if } \theta+i\sigma\le 0\end{cases},\ i=1,\ldots,j-1,$$

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 $\mathrm{f}^{(2)}_{\star,P}(\sigma,\theta)$ of $\mathrm{f}^{(2)}_{\star}(\sigma,\theta)$ is simply defined as

$$\mathrm{f}^{(2)}_{\star,P}(\sigma,\theta)=\begin{cases}\mathrm{f}^{(2)}_{\star}(\sigma,\theta) & \text{if } \theta+\sigma>0\\ (1+b_1)\mathrm{f}^{(2)}_{\star}(\sigma,\theta) & \text{if } \theta+\sigma\le 0\\ & \text{and } \mathrm{f}^{(2)}_{\star}(\sigma,\theta)\le 0\\ (1-b_1)\mathrm{f}^{(2)}_{\star}(\sigma,\theta) & \text{if } \theta+\sigma\le 0\\ & \text{and } \mathrm{f}^{(2)}_{\star}(\sigma,\theta)>0\end{cases}.$$

We observe that

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

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 $\mathrm{f}_{\star,P}(\sigma,\theta)=\mathrm{f}^{(1)}_{\star,P}(\sigma,\theta)+\mathrm{f}^{(2)}_{\star,P}(\sigma,\theta)$.

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

$$\begin{aligned}\mathrm{f}^{(2)}_{\star}(\sigma,\theta)=&-\log\Gamma(\theta+n)+\log\Gamma(\theta+1)\\ &+\log\Gamma(n-\sigma)-\log\Gamma(1-\sigma).\end{aligned}$$

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 $s\ge m_\star$ 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 $s\ge m_\star$ 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.

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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 $x_1|\ldots|x_d@h$ 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	$x_1\|\ldots\|x_7@2$	Exchange
2	7	2	$x_1\|\ldots\|x_7@2$	Fedorov
3	6	2	$x_1\|\ldots\|x_6@1$	Exchange
4	6	2	$x_1\|\ldots\|x_6@1$	Fedorov
5	5	3	$x_1\|\ldots\|x_5@2$	Exchange

Table 2 Number ℓ_r of D optimal designs that have found r times, $r = 1, \ldots, 487$; 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
ℓ_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 $E^D_{\mathcal{F}_1} = 82.3162$.

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 ℓ_r, $r = 1, \ldots, 487$ 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.

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

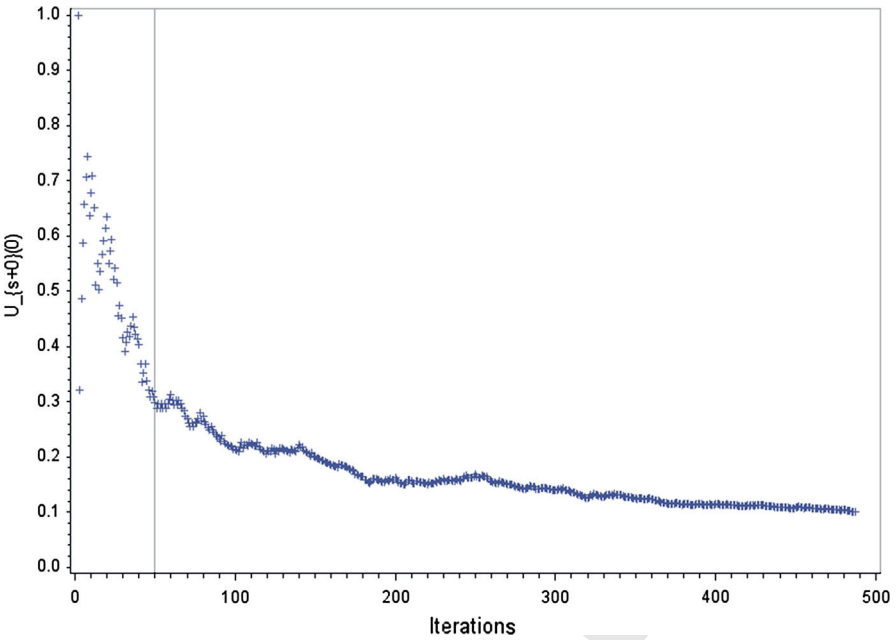

Fig. 1 Estimate of the discovery probability at the next iteration $\hat{U}_{s+0}(0)$ as a function of the iteration s. Test case 1, $M_\star = 1{,}000$, $m_\star = 50$ and $p_\star = 0.10$

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

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

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

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

4.2 Test cases 3 and 4

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

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

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

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

4.3 Test case 5

Let us now consider 5 factors, each with 3 levels, and the model that contains the overall mean, the main effects and all the 2-factor interactions for a total of $1+5\cdot2+10\cdot4 = 51$ degrees of freedom. We search for D-optimal designs that contain 51 points.

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

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

4.4 Practical considerations and guidelines

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

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

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

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

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

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

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

5 Conclusion

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

In principle, this methodology could be applied to any discrete optimisation problem. This topic will be part of future research.

Journal: 180 Article No.: 0562 TYPESET DISK LE CP Disp.:2015/1/31 Pages: 14 Layout: Small-X

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

Springer
the language of science

Author Proof

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.	