Efficient and flexible simulation-based design of complex clinical trials

Abstract

Background: Simulation provides a flexible means with which to estimate clinical trial operating characteristics, allowing complex trial design problems to be addressed without the need for simplifying, yet unrealistic, assumptions. The computational burden of using simulation has, however, restricted its application to only the simplest of trial design problems. These typically involve minimising a single parameter (the sample size) subject to constrained operating characteristics. Many problems of optimal trial design are more complex, requiring optimisation over several parameters to minimise several criteria, subject to several constraints. For the benefits of simulation-based design to be fully realised, a general framework for optimal trial design in problems of this sort is required. In this paper we define such a framework and show how efficient optimisation algorithms can be used to solve these problems in an effective, timely manner.

Methods: We describe a general framework for solving multi-objective trial design problems using Bayesian optimisation. Given estimated operating characteristics of a set of designs generated using simulation, a non-parametric regression model is fitted and used to predict operating characteristics of new designs. These predictions, and their associated uncertainty, are used at each iteration to carefully select the design to be evaluated next. The method is flexible, can be used for almost any problem for which operating characteristics can be estimated using simulation, and can be implemented using existing statistical software packages.

Evaluation: The framework is illustrated using two examples of complex trial design: a cross-classified psychotherapy trial, and a cluster-randomised pilot trial with co-primary endpoints. Simulation studies comparing the proposed approach to a simpler alternative demonstrate the efficiency of the proposed approach.

Conclusions: Bayesian optimisation can be an effective method for complex trial design problems where operating characteristics must be estimated using simulation. By improving the efficiency of these calculations, increasingly complex trial design problems can be addressed without the need for unrealistic assumptions.

1 Introduction

The optimal design of a clinical trial is typically framed as a constrained optimisation problem, where we choose the design such that the sample size is minimised subject to type I and II error rates being within nominal bounds. Solving such a problem can be challenging when complexities in the trial design or analysis mean that a simple analytical expression of the type II error rate (or equivalently, of the power) is not available. In these cases it has been suggested that a Monte Carlo (MC) estimate of power be used instead [1, 27]. An MC estimate can be obtained by simulating trial data under the alternative hypothesis many times, carrying out the prescribed analysis on each data set, and calculating the proportion of times the null hypothesis is rejected. Any other operating characteristic of interest can be estimated in a similar fashion. The method is both conceptually simple and highly flexible, being applicable to almost any trial design and method of analysis. Applications can be found across a variety of settings, including problems involving hierarchical models [15, 19], proportional hazards models [39], logistic regression models [16], individual patient data meta-analyses [42, 26], realistic patient enrolment models [14], stepped wedge designs [2, 20], and cluster randomised crossover designs [35].

The flexibility afforded by the simulation approach comes at the cost of computation time. The optimal design for a clinical trial can nevertheless be found reasonably quickly in simple cases. For example, consider a trial design problem where we must choose the per-arm sample size n from within some feasible range $[0, n_m]$. We aim to find the smallest n such that the trial will be powered at the nominal level, say 80%. A simple bisection search can be used to find this optimal n and will require no more than $\lfloor \log_2 n_m + 1 \rfloor$ iterations. If the upper limit on the sample size was $n_m = 800$, for example, at most 10 simulations will be required. Thus, even if computing an MC estimate of power takes several minutes, the optimal design problem can still be solved in a timely manner. A general framework for simulation-based trial design in problems of this nature is given in [27].

In this paper we are interested in optimal trial design problems which are more complicated, in at least two respects. Firstly, we consider problems with several design parameters, each influential on the trial operating characteristics. As the number of design parameters increases, so too does the set of possible designs which must be searched over, and the restriction regarding the number of designs we can evaluate begins to have stronger implications. Secondly, we consider problems where interest lies in minimising several quantities rather than a single measure of sample size. In particular, we are interested in cases with no *a priori* cost function combing these individual measures into a single quantity to be minimised. A solution to such a problem should can consist not of a single proposed design, but rather a set of designs which offer a range of trade-offs between the different objectives. A simple example exhibiting each of these characteristics is a balanced cluster randomised trial, with two design parameters (the number of clusters in each arm, and the number of participants in each cluster), and two quantities to be minimised (the number of clusters, and the total number of participants).

Methodology and associated software for simulation-based trial design has typically focussed on a specific area of application. MLPowSim [5] makes use of the MLwiN software for multilevel analysis, and can estimate the power of a variety of multilevel trial designs at a set of evenly spaced design points. The ipdpower package in Stata was motivated by problems in individual patient data meta-analysis, although can be applied to any problem with a two-level nested structure. Sample size determination for stepped wedge trials is addressed in the R package SWSamp [2]. The Stata package SimSam [19] is more general, allowing the user to fully specify their own data generation and analysis models. It has been applied to multilevel designs [19], designs for logistic regression [16], and stepped wedge designs [20]. It is, however, restricted to the minimisation of a single objective, over a single variable, with the simulation of a single power constraint.

If we are to extend the ideas of simulation-based trial design to more complex problems, we require a more general framework employing more efficient optimisation algorithms. Outwith the context of clinical trial design a great deal of research has addressed optimisation problems where the evaluation of a solution is a computationally demanding, or *expensive*, operation (see [37] for an early review). One approach addresses the problem by substituting the expensive function with an approximation known as a *surrogate model*. If the surrogate model is a good approximation of the real function and is cheap to evaluate, the time required to solve the optimisation problem can be dramatically reduced [22]. In this paper we will explore how complex simulation-based clinical trial design problems can also be addressed using a particular type of surrogate model, Gaussian process regression [33], and associated efficient optimisation algorithms [22]. Our focus is on flexibility, and so we will not impose any restrictions regarding the nature of the endpoint(s), underlying statistical model, test statistic, or acceptance region, other than that the data generation and analysis can be simulated.

The remainder of the paper is structured as follows. Two motivating trial design problems are described in Section 2. In Section 3 we briefly provide the necessary background and notation regarding Monte Carlo estimation and multi-objective optimisation. In Section 4 we describe Gaussian process regression, an efficient global optimisation algorithm, and a framework for their application to a broad range of trial design problems. We return to the examples in Section 5, illustrating how the framework can be applied in practice. A simulation study comparing the performance of the proposed methodology with a simpler alternative approach is reported in Section 6. We conclude with a discussion of the implications and limitations of the proposed approach in Section 7.

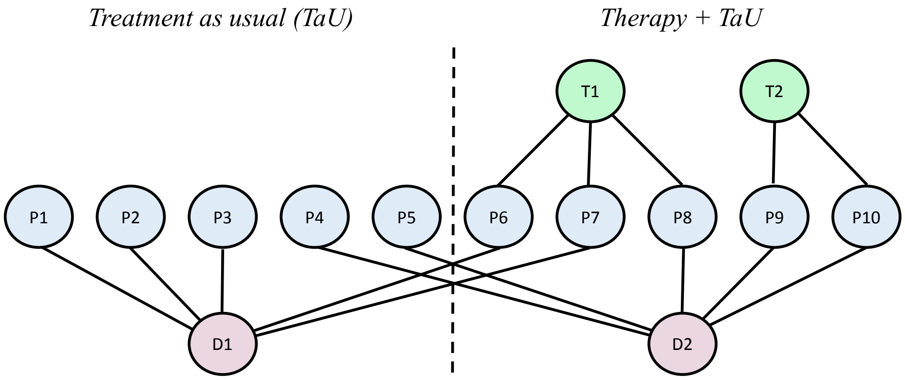

Figure 1: Multilevel structure of a cross-classified psychotherapy trial where patients are nested within doctors in the control arm, patients are cross-classified with therapists and doctors in the intervention arm, and doctors are crossed with treatment.

2 Motivating examples

2.1 Cross-classified psychotherapy trial

Consider a trial where $n_1 + n_2$ individuals are randomised between treatment as usual (TaU) and TaU plus a psychotherapy intervention, under an allocation ratio of $r = n_1/n_2$. In the intervention arm k therapists deliver treatment to n_2 patients, with patients nested within therapists. We assume that patients are allocated a therapist at random, leading to cluster sizes which follow a multinomial distribution. Patients in both arms of the trial receive TaU from j doctors, with patients nested within doctors. Again, patients are assumed to be allocated to doctors at random. This leads to a multilevel data structure where patients are nested within doctors in the control arm, patients are cross-classified with therapists and doctors in the intervention arm, and doctors are crossed with treatment. This structure is illustrated in Figure 1.

The proposed model describing the continuous primary outcome measure of patient i nested within therapist j and doctor k is

$$y_{ijk} = \beta_0 + \beta_1 t_i + t_i u_j + v_k + e_i, \tag{1}$$

where t_i is a binary indicator of allocation to the intervention arm, $u_j \sim N(0, \sigma_T^2)$ and $v_k \sim N(0, \sigma_D^2)$ are therapist and doctor random effects respectively, and $e_i \sim N(0, \sigma_P^2)$ is the patient-level residual. For the purposes of power calculations we assume the variance components are known and equal to $\sigma_T^2 = 0.01, \sigma_D^2 = 0.03, \sigma_P^2 = 0.96$. In the control arm the proportion of the variance attributable to between-doctor variation is $\rho_D^{(1)} = \sigma_D^2/(\sigma_D^2 + \sigma_P^2) = 0.030$. In the intervention arm the proportion of the variance attributable to between-therapist variation is $\rho_T^{(2)} = \sigma_T^2/(\sigma_T^2 + \sigma_D^2 + \sigma_P^2) = 0.01$, and similarly for between-doctor variation, $\rho_D^{(2)} = \sigma_D^{(2)}/(\sigma_T^2 + \sigma_D^2 + \sigma_P^2) = 0.03$.

The primary analysis will fit model (1) to the observed data using maximum likelihood, providing an estimate $\hat{\beta}_1$ of the treatment effect. A Wald test of the null hypothesis $H_0 : \beta_1 = 0$ will then be conducted. The test statistic $\hat{\beta}_1/s.e.(\hat{\beta}_1)$ is assumed to follow a normal distribution under H_0, allowing the type I error rate to be controlled at the nominal level of $\alpha^* = 0.025$ (one sided). We require that the power at the alternative hypothesis $H_1 : \beta_1 = 0.3$ is no less than 80%. Subject to these constraints, we aim to find trial configurations which minimise both the number of participants and the number of therapists, recognising that these two objectives will conflict with one another.

To the best of our knowledge there are no analytic formulae which calculate the power of a trial under the model (1). Using MC estimates of power will be computationally demanding since each sample will require the fitting of a multilevel model, placing a practical limit on the number of designs which we can evaluate in a timely manner.

2.2 Cluster randomised pilot trial

For our second example we consider a pilot trial for a complex intervention delivered in care homes. The aim of the pilot is to inform whether or not a large confirmatory trial of the intervention should be carried out on grounds of efficacy and feasibility. The primary outcome measures are binary indicators of response and adherence to the intervention, both measured at the patient level. Randomisation is at the care home level to prevent contamination, with k_1 clusters in the control arm and k_2 in the intervention arm. In each of the k_i care homes, m_i residents will be recruited $(i = 1, 2)$.

The overall probability of response in arm $i = 1, 2$ is denoted $p_r^{(i)}$. The probability of adherence is denoted p_a. We anticipate between-cluster variability in the response outcome, and for this to be greater in the control arm than in the more standardised intervention arm. This variability is modelled by assuming that the true response rate in a given cluster follows a Beta distribution, with parameters

$$a^{(1)} = \frac{50p_r^{(1)}}{(1 - p_r^{(1)})}, \quad b^{(1)} = 50 \tag{2}$$

in the control arm, and

$$a^{(2)} = \frac{100p_r^{(2)}}{(1 - p_r^{(2)})}, \quad b^{(2)} = 100 \tag{3}$$

in the intervention arm. Response and adherence outcomes in the intervention arm are expected to be highly correlated at the patient level, encapsulated through an odds ratio of 10 (constant across clusters).

Efficacy will be assessed through a two-sample t-test of the null hypothesis of equal response rates in the two arms, $p_r^{(2)} - p_r^{(1)} = 0$. Adherence will be assessed using a one-sample t-test of adherence rates in the intervention arm alone, with null hypothesis $p_a = 0.6$. Both tests will be one-sided tests of cluster means, and the test of response will use the Welch approximation to the degrees of freedom. If both tests result in the rejection of their null hypotheses, then a phase III trial will be undertaken. We denote this event as A.

Operating characteristics of interest are as in the phase II design for two binary endpoints proposed in [6]. Specifically, we require that

$$\alpha_r = Pr[A \mid H_{0,1}] \leq \alpha_r^* = 0.3 \tag{4}$$

$$\alpha_a = Pr[A \mid H_{1,0}] \leq \alpha_a^* = 0.4 \tag{5}$$

$$\beta = Pr[A \mid H_{1,1}] \geq \beta^* = 0.1, \tag{6}$$

where

$$H_{0,1} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.3, p_a = 0.8 \tag{7}$$

$$H_{1,0} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.5, p_a = 0.6 \tag{8}$$

$$H_{1,1} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.5, p_a = 0.8. \tag{9}$$

We aim to find a design which minimises both the total number of clusters and total number of participants, subject to the above constraints. Note that relatively high nominal type I error rates α_r^* and α_a^* have been set to allow for sufficiently high power to be obtained within the sample size constraints of a pilot trial [28].

Given the correlation between outcomes and the between-cluster variability in response rates, an analytic expression for $Pr[A \mid H]$ is not available. Instead, each operating characteristic must be estimated through simulation. Note that because the assumptions underlying each t-test may not be met[1] , we cannot assume that the individual type I error will be controlled at the level of the test. We therefore allow for an adjustment to the critical region of each test, now rejecting the null hypothesis if $t_r + c_r > T^{-1}(\alpha_r^*)$. MC estimation of a given operating characteristic will be less computationally demanding than in the case of the previous example, as only two t-tests are required in the data analysis step. However, in this case we must compute MC estimates for three quantities rather one when evaluating a design, and the overall computational burden will be significant.

[1] in particular, normally distributed cluster means when m_1, m_2 are low.

3 Preliminaries

3.1 Monte Carlo estimation

Monte Carlo methods can be used to numerically approximate expectations $\mathbb{E}[f(Z)]$ of real valued functions $f(Z)$ with respect to the probability distribution of Z. Given N samples of Z, denoted z_i, $i = 1, \dots, N$, the MC estimate is

$$\mathbb{E}[f(Z)] \approx \frac{1}{N} \sum_{i=1}^{N} f(z_i). \tag{10}$$

The estimate is unbiased for all N and has variance equal to

$$\omega^2 = Var[\frac{1}{N} \sum_{i=1}^{N} f(z_i)] = \frac{1}{N} Var[f(z_i)]. \tag{11}$$

The standard error of the MC estimate will therefore reduce at a rate of $1/\sqrt{N}$ as we increase N. When N is large we can consider an MC estimate to be the true expectation plus a normally distributed error term with 0 mean and variance ω^2, i.e.

$$\frac{1}{N} \sum_{i=1}^{N} f(z_i) = \mathbb{E}[f(Z)] + e, \text{ where } e \sim N(0, \omega^2). \tag{12}$$

In the context of simulation-based trial design, if Z is the test statistic to be compared with an acceptance region Λ then the probability of acceptance under hypothesis H is $\mathbb{E}[I(Z \in \Lambda) \mid H]$. Thus, an MC estimate of the power of a trial design under H can be obtained given N test statistics $z_1, \dots, z_N$ sampled under H. The steps required to simulate these statistics are described in [27], and we briefly summarise them here:

1. Define the population model. This describes the underlying target population and should specify all population parameters and distributions under the hypothesis of interest.

2. Define the sampling strategy. This should specify how the sample of patients in the trial will be drawn from the population, and will include the sample size of the trial.

3. Define the method of analysis. For hypothesis testing, this will include defining the form of the test statistic Z and the acceptance region Λ.

Given each of the above elements, pseudo-random number generators can be used to simulate the recruitment, randomisation and primary outcome measure of patients under the hypothesis of interest, from which a test statistic z_i can be calculated.

3.2 Multi-objective optimisation of trial designs

A solution to the trial design problem consists of a vector of design parameters $\mathbf{x}$, and the *solution space* $\mathcal{X}$ is the set of all solutions. A simple design problem may have a 1-dimensional solution space, while more complex problems may have several dimensions. Elements of $\mathbf{x}$ may include parameters defining the sample size of the trial, the acceptance region to be used in the analysis, or any other design aspect which we have control of and which may influence the trial operating characteristics.

An *objective function* $f(\mathbf{x})$ is a function $f : \mathcal{X} \rightarrow \mathbb{R}$ which we wish to minimise. In a multi-objective problem with B objectives, we denote the vector of objective values as $\mathbf{y} = (f_1(\mathbf{x}), \dots, f_B(\mathbf{x})) \in \mathbb{R}^B$. We will describe $\mathbb{R}^B$ as the *objective space.*

A *constraint function* $g(\mathbf{x})$ is a function $g : \mathcal{X} \rightarrow \mathbb{R}$ which must be less than or equal to 0 for the solution $\mathbf{x}$ to be considered feasible. For example, if type II error rate is denoted by $\beta(\mathbf{x})$ and the nominal type II error rate is set at β^*, a constraint function would be $g(\mathbf{x}) = \beta(\mathbf{x}) - \beta^*$. We denote C constraint functions as $g_j(\mathbf{x}),\ j = 1, \dots, C$. The problem of optimal trial design can now be stated as

$$\min_{\mathbf{x} \in \mathcal{X}} \ f_i(\mathbf{x}),\ i = 1, \dots, B \tag{13}$$

$$\text{subject to } g_j(\mathbf{x}) \leq 0,\ j = 1, \dots, C. \tag{14}$$

We denote by $\prec$ the relation of Pareto dominance. Specifically, $\mathbf{x}_* \prec \mathbf{x}$ if $f_i(\mathbf{x}_*) \leq f_i(\mathbf{x})$ for $i = 1, \dots, B$, and $f_j(\mathbf{x}_*) < f_j(\mathbf{x})$ for some j. The *Pareto set* is the set of non-dominated solutions $\mathcal{X}_p = \{\mathbf{x} \in \mathcal{X} \mid \nexists\ \mathbf{x}_* \in \mathcal{X} \text{ s.t. } \mathbf{x}_* \prec \mathbf{x}\}$.

Following [13] we define an approximation set $\mathcal{A}$ to be any set $\mathcal{A} \in \mathcal{X}$ with $\forall\ \mathbf{x} \in \mathcal{A} :\ \nexists\ \mathbf{x}_* \in \mathcal{A} : \mathbf{x}_* \prec \mathbf{x}$. A set $\mathcal{A}$ is feasible if all constraints are satisfied by every member of $\mathcal{A}$. The dominated hypervolume of set $\mathcal{A}$ is the volume of the subspace dominated by solutions in $\mathcal{A}$ and bounded by a reference point r:

$$H(\mathcal{A}) = \text{Vol}(\{\mathbf{y} \in \mathbb{R}^B \mid \mathbf{y} \text{ is dominated by some } \mathbf{y}_* \in \mathcal{A} \text{ and } \mathbf{y} \prec r\}). \tag{15}$$

The largest possible hypervolume of any feasible approximation set $\mathcal{A}$ is achieved when $\mathcal{A} = \mathcal{X}_p$. We can therefore frame the multi-objective optimisation problem as finding the feasible set $\mathcal{A}$ with largest hypervolume.

Objective values of example Pareto and approximation sets are shown in Figure 2 for a hypothetical cluster randomised trial where we wish to minimise both the number of participants and the number of clusters, subject to a power constraint.

4 Methods

We propose to apply the efficient global optimisation algorithm of [23], also known as Bayesian optimisation, to solve trial design problems of the type described in Section 2. This algorithm recognises that the evaluation of a solution

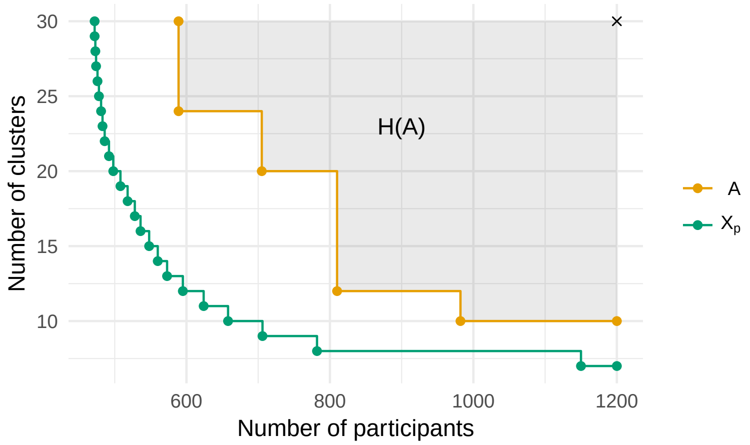

Figure 2: Example Pareto front $\mathcal{X}_p$ (green) and approximation set $\mathcal{A}$ (yellow) for a cluster randomised trial design problem. The dominated hypervolume of the approximation set with respect to a reference point (cross) is the shaded area.

is an expensive operation, and invests resources in carefully selecting each solution to be evaluated next. To do so, a non-parametric regression model is approximating the true underlying function of interest is constructed and used to describe our knowledge and uncertainty about the value of the function at solutions which have not yet been evaluated. Optimisation then proceeds in a probabilistic framework by considering a point $\mathbf{x}_*$ for evaluation and asking questions such as 'what is the probability that the solution $\mathbf{x}$ is better than all the solutions I have evaluated so far?', and 'how much of an improvement can I expect to see if I evaluate $\mathbf{x}$?'. For clarity, we will focus here on the case where there is a single constraint function g, e.g. the power function, which must be evaluated using the Monte Carlo method. The algorithm can then be described as in Algorithm 1.

Algorithm 1 Efficient Global Optimisation

```
1: Compute MC estimates $\mathbf{y}_E = (g(\mathbf{x}^{(1)}), \dots, g(\mathbf{x}^{(E)}))$
2: while Computation budget not exhausted do
3:     Regress $\mathbf{y_E}$ on $\mathcal{X}_E = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(E)})$
4:     Find $\mathbf{x}_* = \arg\max EI(\mathbf{x})$
5:     Compute MC estimate $y_* = g(\mathbf{x}_*)$ and add to $\mathbf{y_E}$, $\mathcal{X}_E$
6:     Update the computational budget
7: end while
```

In what follows we will first consider step (3), describing Gaussian process regression models and outlining how they can be fitted and used to make predictions. The notion of expected improvement (EI) in step (4) will then be defined for the constrained multi-objective problems we are concerned with. We then cover the remaining aspects of implementation.

4.1 Gaussian process regression

Consider a set of points $\mathcal{X}_E = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(E)}\} \subset \mathcal{X}$ at which an expensive function g will be estimated using the Monte Carlo method. Consider also some other point $\mathbf{x}_* \notin \mathcal{X}_E$ where we are interested in making a prediction of $g(\mathbf{x}_*)$. The value of g at each point in $\{\mathcal{X}_E, \mathbf{x}_*\}$ is initially unknown, but can be modelled by a Gaussian process (GP).

In using a GP we assume that our belief regarding the the values of g can be represented as a multivariate normal distribution. Prior to computing any estimates of g, we assume that the mean function of this multivariate normal is equal to zero[2]. We write the (symmetric, positive definite) covariance matrix of the distribution as

$$\begin{pmatrix} K(\mathcal{X}_E, \mathcal{X}_E) & K(\mathcal{X}_E, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathcal{X}_E) & K(\mathbf{x}_*, \mathbf{x}_*) \end{pmatrix}. \tag{16}$$

$K(\mathcal{X}_E, \mathcal{X}_E)$ is the $E \times E$ covariance matrix for the points $\mathcal{X}_E$, $\mathbf{k}_* = K(\mathcal{X}_E, \mathbf{x}_*)$ is the E-length vector of covariances between $\mathcal{X}_E$ and $\mathbf{x}_*$, and $K(\mathbf{x}_*, \mathbf{x}_*)$ is the variance at $\mathbf{x}_*$.

Given this prior distribution, we compute the MC estimates $y^{(1)}, \dots, y^{(E)}$ at each point in $\mathcal{X}_E$. From equation (12), $y^{(i)} = g(\mathbf{x}^{(i)}) + e^{(i)}$ where $e^{(i)}$ is a zero-mean normally distributed error term with standard deviation $\omega^{(i)}$. We denote by Δ the $E \times E$ diagonal matrix where the ith entry is $[\omega^{(i)}]^2$. The distribution of $g(\mathbf{x}_*)$ conditional on the observed $\mathbf{y}$ can be shown to be normal with mean $\mathbf{k}_*^\top (K + \Delta)^{-1}\mathbf{y}$ and variance $k(x_*, x_*) - \mathbf{k}_*^\top (K + \Delta)^{-1}\mathbf{k}_*$ [33]. Thus, given a prior covariance matrix of the form (16) and some MC estimates of g at the points $\mathcal{X}_E$, a conditional predictive distribution of $g(\mathbf{x}_*)$ can be found. It is this distribution which will be used in the optimisation algorithm when deciding which solution should next be evaluated.

The distributions of interest are dependant on the entries of the prior covariance matrix (16). The matrix is populated using a covariance function (or *kernel*), $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. This function must be symmetric and positive definite for the covariance matrix to have the same properties. One such covariance function is the squared exponential, which has the form

$$k(\mathbf{x}, \mathbf{x}') = \sigma \exp\left(- \sum_{j=1}^{D} \frac{(x_j - x'_j)^2}{\lambda_j^2} \right). \tag{17}$$

By using covariance functions of this form we will obtain a Gaussian process which is infinitely differentiable over $\mathcal{X}$ and thus very smooth. This would appear to be a reasonable restriction to place upon the power functions we are

[2] This is not a restrictive assumption. After observing estimates of the function g and updating the GP model to account for these, the mean function can take on non-zero values.

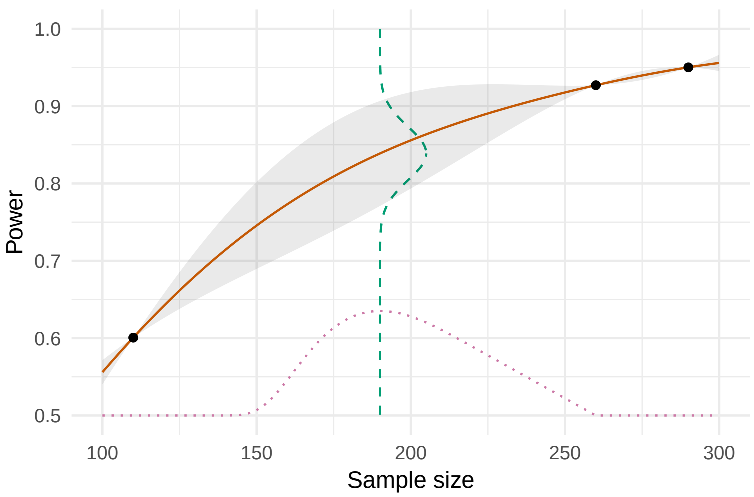

Figure 3: A Gaussian process model of a power function over a one-dimensional sample size (solid line) based on three evaluations. Uncertainty is shown as the shaded area. Expected improvement (dotted line) is maximised at a sample size of 190, where the predicted power is normally distributed around a mean estimate of 0.84 (dashed line).

interested in. In order to populate the covariance matrix we must choose values of the parameters $\boldsymbol{\theta} = (\sigma, \lambda_1, \dots, \lambda_D)$. We do this by numerically optimising the log marginal likelihood

$$\log p(\mathbf{y} \mid \mathcal{X}_E, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top [K + \Delta]^{-1}\mathbf{y} - \frac{1}{2}\log|K + \Delta| - \frac{n}{2}\log 2\pi, \qquad (18)$$

considered as a function of $\boldsymbol{\theta}$ [33]. Fitting a GP model by maximum likelihood in this manner can be done using the function `km` in the R package DiceKriging, as shown in the appendix.

An illustration of a Gaussian process regression model of a power function in one dimension is given in Figure 3. The power of three different choices of sample size have been calculated and a GP model fitted to the results. The figure illustrates how the uncertainty in the model predictions (shaded area) increases the further we are from a point which has been evaluated. The GP prediction of power at a sample size of $n = 190$, shown as a dashed line, is normally distributed with mean 0.84 and standard deviation 0.035.

4.2 Expected improvement

At any given point in the algorithm, an approximation set $\mathcal{A}$ will have been obtained. Upon evaluating the point $\mathbf{x}_*$, and assuming that $g_j(\mathbf{x}_*) < 0$ for all $j = 1, \dots, C$, a new approximation set $\mathcal{A}_*$ will be identified. The improvement resulting from the evaluation of $\mathbf{x}_*$ is the difference in the dominated hypervolumes:

$$I = H(\mathcal{A}_*) - H(\mathcal{A}). \tag{19}$$

Improvement I is conditional on $\mathbf{x}_*$ satisfying all constraints. Before the necessary MC estimates are computed, however, we do not that this will be the case. We therefore modify I to account for the probability that $\mathbf{x}_*$ will be considered feasible after the MC estimates have been obtained. This probability can be estimated using the GP regression methodology described in Section 4.1. A GP model of unknown constraint function g will give a prediction $g(\mathbf{x}_*) \sim \mathcal{N}(m, s^2)$, and we will consider the point $\mathbf{x}_*$ feasible if the upper $100 \times p\%$ quantile of this distribution is below 0. We denote this quantile as

$$q(\mathbf{x}_*) = m + \Phi^{-1}(p)s, \tag{20}$$

where Φ is the standard normal cummulative distribution. Following an evaluation of $\mathbf{x}_*$ the GP model will be updated and the quantile revised to $q_+(\mathbf{x}_*)$. Before the evaluation the value of $q_+(\mathbf{x}_*)$ is unknown, but it is shown in [32] that its predictive distribution is $q_+(\mathbf{x}_*) \sim N(m_+, s_+^2)$ where

$$m_+ = m + \Phi^{-1}(p)\sqrt{\frac{\omega^2 s^2}{\omega^2 + s^2}} \tag{21}$$

$$s_+^2 = \frac{[s^2]^2}{\omega^2 + s^2}, \tag{22}$$

and ω is the MC error of the planned evaluation, estimated as $m(1-m)/N$ where N is the number of MC samples to be used. The predictive distribution can then be used to calculate the probability that the point $\mathbf{x}_*$ will be considered feasible following its evaluation. Following [38], we multiply the theoretical improvement I by this probability, thus penalising candidate solutions with a low chance of satisfying the constraint. We then choose to evaluate the solution which maximises

$$EI = [H(\mathcal{A}_*) - H(\mathcal{A})] \prod_{j=1}^{C} \Phi\left(\frac{-m_{j,+}}{s_{j,+}}\right), \tag{23}$$

where we now include all $j = 1, \ldots, C$ constraint functions. This maximisation problem is in itself complex, with a potentially large number of local maxima. We therefore use the particle swarm optimisation algorithm as implemented in the R package pso [4], designed to avoid becoming trapped in local maxima, to solve this sub-problem.

An illustration of expected improvement for a single-objective problem is given in Figure 3. When choosing which sample size to evaluate next and aiming to find the lowest sample size with at least 80% power, we balance the potential improvement over the best current solution (a sample size of 260) with the probability of constraint satisfaction. In this case, we would choose to evaluate the sample size of 190, estimated by the GP model to have a power of 84%.

4.3 Implementation

To apply Algorithm 1 in practice we must first choose the initial set of points to be evaluated, $\mathcal{X}_E$. One recommendation is to include 10 points for each dimension of the solution space, and to allocate between 25 and 50% of the total computation budget to their evaluation []. To select the location of the points in $\mathcal{X}_E$ we use a random Latin hypercube sample generated using the R package lhs [8], with the sample optimised to maximise the minimum distance between the points. The number of iterations and the number of MC samples N used at each iteration must also be chosen. Given a total computational budget in terms of MC samples, the choice of these values should account for the fact that fitting GP regression models in R to more than around 800 points is infeasible [9].

As the algorithm depends on GP regression models, it is important that the fit of these models is assessed. One approach is to regularly plot the predicted mean and standard deviation in one dimension, centred at the last evaluated point. Poor model fit could be identified if the mean function is not, for example, strictly increasing as expected. We can also contrast the predicted function values with the obtained function values at each iteration.

We have chosen to use R to implement the proposed framework, partly due to the availability of robust and efficient R packages for fitting Gaussain process models (DiceKriging [36]) and for global optimisation (pso [4]). Using R also provides flexibility in terms of the user-writen simulation routines by facilitating various complicated analysis procedures, e.g. multilevel modelling through lme4 [3]. Our implementation works to a simple interface. The user must provide instances of three data frames. The first, `sol_space`, contains a row for each design parameter describing its name and its lower and upper bounds. The second, `hypotheses`, contains a row for each hypothesis which power is to be simulated under. Each row should include a label and a value for each parameter needed (together with design parameters) to define the simulation. Finally, `constraints` contains a row for each constraint function g_j. Each row should include a label for the constraint, the hypothesis it pertains to, a nominal power which should not be exceeded, and the confidence we require in the constraint being satisfied (i.e. the p in Equation 20). Further, two functions are required. The first, `objectives`, takes as argument a vector of design parameter values $\mathbf{x}$ and returns a vector of objective values $(f_1(\mathbf{x}), \dots, f_B(\mathbf{x}))$. The second, `sim_trial`, takes as arguments a vector of design parameter values $\mathbf{x}$ and a hypothesis (i.e. a row from the hypotheses data frame). The function should simulate the necessary data generation and analysis and return a boolean indicator of the rejection of the null hypothesis, or, more generally, of declaring 'success'. Given these components, the example code can be applied to solve the trial design problem at hand.

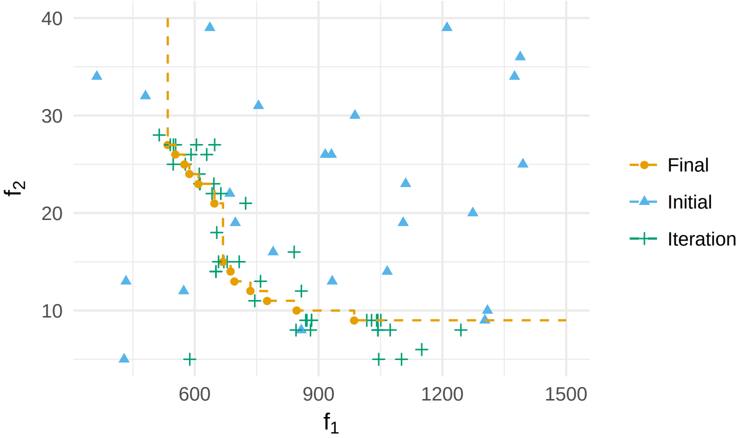

Figure 4: Objective values of solutions in the initial set $\mathcal{X}_E$ (blue), subsequent iterations of the algorithm (green), and those in the final approximation of the Pareto set (yellow).

5 Application to the examples

5.1 Cross-classified psychotherapy trial

Recall our first motivating example of a cross-classified psychotherapy trial. In terms of the framework described in Section 3.2, the design parameters are the number of participants in the intervention arm n_2, the between-arm allocation ration r, the number of therapists k and the number of doctors j. For simplicity we will fix $j = 50$. The bounds on the remaining design parameters are taken to be $r \in [0.5, 1]$, $n_2 \in [200, 1000]$, and $k \in [5, 40]$. This defines the solution space $\mathcal{X} = \{(r, n_2, k) \mid r \in [0.5, 1] \subset \mathbb{R}, n_2 \in [200, 1000] \subset \mathbb{Z}, k \in [5, 40] \subset \mathbb{Z}\}$.

The objective space has two dimensions, as we wish to minimise both the total number of patients $f_1(\mathbf{x}) = (r+1)n_2$ and the number of therapists $f_2(\mathbf{x}) = k$. The only constraint we must satisfy is that the type II error rate $\beta(\mathbf{x})$ under the alternative hypothesis $H_1 : \beta_1 = 0.3$ is no more than $\beta^* = 0.2$. This gives the constraint function $g_1(\mathbf{x}) = \beta(\mathbf{x}) - 0.2$. A program to simulate the data generation and analysis under H_1 for a given $\mathbf{x}$ is given in the appendix.

We initialised the optimisation algorithm by generating a random Latin hypercube sample of size 30 and computed MC estimates of power for each point using $N = 250$ samples. Following this, 60 iterations of Algorithm 1 were applied, with $N = 500$ samples used at each iteration. In Figure 4 we plot the 90 evaluated solutions in the objective space, distinguishing between those in the initial design $\mathcal{X}_E$, those which where subsequently evaluated during the iterative phase of the algorithm, and those which together form the final approximation set.

At the ith iteration of the algorithm we calculated the dominated volume

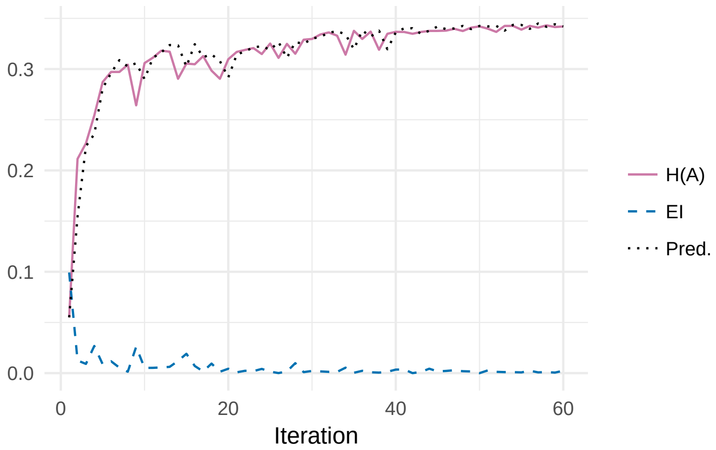

Figure 5: Quality of the approximation set obtained as the algorithm proceeds (solid line). The maximised expected improvement at each iteration is also plotted (dashed line), together with the corresponding predicted approximation set quality (dotted line).

$H(\mathcal{A}_i)$, plotted in Figure 5. Also plotted is the expected improvement, calculated at iteration i with respect to iteration $i+1$, and the subsequent prediction of the dominated volume $H(\mathcal{A}_{i+1})$. In this instance the algorithm appears to plateau after around 40 iterations.

The solutions which form the final approximation set are detailed in Table 1. As few as 9 therapists can lead to a sufficiently powered trial, although nearly 1000 participants are required in this configuration. On the other hand, if minimising the total sample size is deemed more important than minimising the number of therapists, we see that as few as 535 participants are necessary providing 27 therapists are included. The approximation set contains 13 solutions in total, providing a reasonable set of options from which the solution best representing the priorities of the decision maker(s) can be selected.

As can be seen from Table 1, approximate upper 99% confidence limits based on the MC estimates of power often exceed the corresponding nominal bound of 0.2. For example, the solution described in the last row would have an approximate 98% two-sided confidence interval of $(0.172, 0.256)$. However, when the GP model is used to predict power instead of the raw MC estimate, the interval is $(0.173, 0.197)$. To verify this prediction we computed a more precise MC estimate using $N = 10,000$ samples, which gave an interval of $(0.175, 0.193)$.

5.2 Cluster-randomised pilot trial

Recall our second motivating example, the cluster randomised pilot trial in care homes. In this case we have 6 design parameters to optimise over: the numbers of clusters in each arm, k_1 and k_2; the cluster size in each arm, m_1 and m_2; and

r	n_2	k	$\hat{\beta}$ $(s.e.)$	f_1	f_2
0.87	286	27	0.178 (0.017)	535	27
0.85	299	26	0.200 (0.018)	553	26
0.82	316	25	0.178 (0.017)	575	25
0.83	321	24	0.166 (0.017)	587	24
0.82	334	23	0.188 (0.017)	609	23
0.82	356	21	0.208 (0.018)	648	21
0.85	362	15	0.168 (0.017)	669	15
0.83	375	14	0.172 (0.017)	687	14
0.85	377	13	0.214 (0.018)	696	13
0.87	394	12	0.166 (0.017)	735	12
0.88	412	11	0.180 (0.017)	775	11
0.92	442	10	0.168 (0.017)	847	10
0.92	513	9	0.176 (0.017)	986	9

Table 1: Solutions in the approximation set after 60 iterations of the Bayesian optimisation algorithm.

the critical region adjustments for the tests of response and adherence, c_r and c_a. Together with their bounds they specify the solutions space

$$\mathcal{X} = \{(m_1, k_1, m_2, k_2, c_a, c_r) \mid m_1, m_2 \in [5, 25] \subset \mathbb{Z}, k_1, k_2 \in [3, 15] \subset \mathbb{Z}, c_r, c_a \in [-2, 2] \subset \mathbb{R}\}. \tag{24}$$

As in the preceding example, the objective space has two dimensions. We aim to minimise the total number of participants $f_1(\mathbf{x}) = m_1k_1 + m_2k_2$ and the total number of clusters $f_2(\mathbf{x}) = k_1 + k_2$. In this example we have three constraint functions: $g_1(\mathbf{x}) = \alpha_r(\mathbf{x}) - 0.3$; $g_2(\mathbf{x}) = \alpha_a(\mathbf{x}) - 0.4$; and $g_3(\mathbf{x}) = \beta(\mathbf{x}) - 0.1$. Again, a program for the simulation of the trial data generation and analysis is given in the appendix.

A 60-point Latin hypercube sample $\mathcal{X}_E$ of initial solutions was generated, and these were evaluated using $N = 250$ MC samples. Following this, 60 iterations of Bayesian optimisation were then applied, using $N = 500$ samples for each MC estimate. The resulting approximation set is illustrated in Figure 6, together with some of the initial set $\mathcal{X}_E$ and the solutions evaluated during the iterative phase of the algorithm.

Only 12 of the initial 60 solutions are displayed in Figure 6, with the remainder being located outwith the plotted range. The final solutions which form the approximation set are described in further detail in Table 2. The results suggests that at least 6 clusters are required in total. Reducing the total number of participants required from 144 to 95 is possible, providing an increase in the number of clusters from 6 to 13 is deemed acceptable. The approximation set contains 5 solutions in total, which, given the scales involved, would appear to provide sufficient flexibility to enable a trial design with the desired balance between objectives to be found.

As can be seen from Table 2, approximate upper 98% confidence limits for

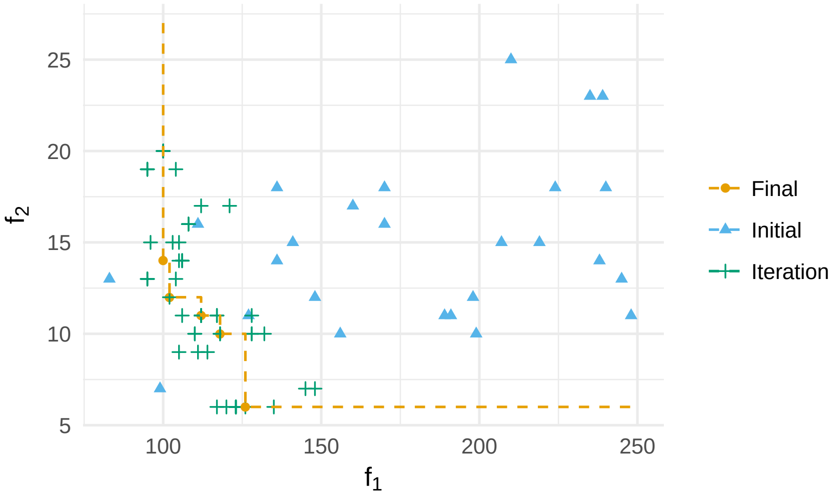

Figure 6: Going up to 60 iterations. We have focussed on the approximation front, omitting 34 points (all from the initial design) which take larger values of f_1 and/or f_2.

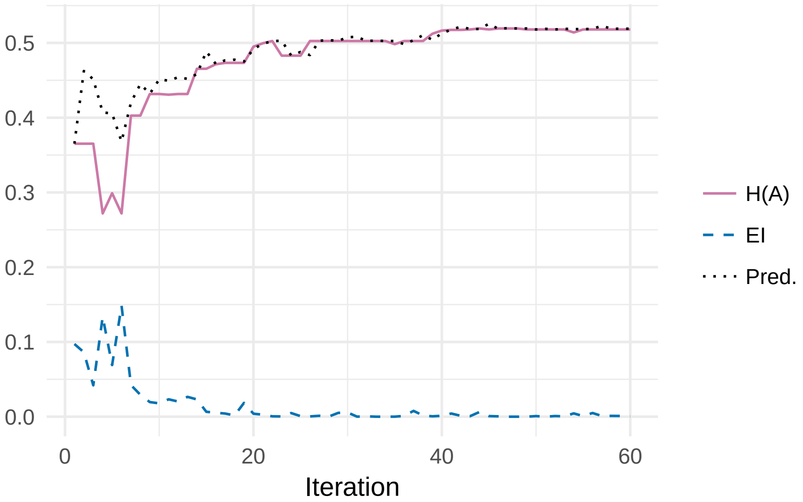

Figure 7: Quality of the approximation set obtained as the algorithm proceeds (solid line). The maximised expected improvement at each iteration is also plotted (dashed line), together with the corresponding predicted approximation set quality (dotted line).

m_1	k_1	n_1	m_2	k_2	n_2	c_r	c_a	$\hat{\beta}$ $(s.e.)$	$\hat{\alpha}_r$ $(s.e.)$	$\hat{\alpha}_a$ $(s.e.)$	f_1	f_2
5	8	40	10	6	60	0.094	-0.013	0.092 (0.013)	0.270 (0.020)	0.328 (0.021)	100	14
6	6	36	11	6	66	0.086	-0.036	0.084 (0.012)	0.234 (0.019)	0.310 (0.021)	102	12
8	5	40	12	6	72	0.104	-0.050	0.080 (0.012)	0.288 (0.020)	0.286 (0.020)	112	11
10	4	40	13	6	78	0.068	-0.099	0.092 (0.013)	0.266 (0.020)	0.294 (0.020)	118	10
19	3	57	23	3	69	0.121	0.084	0.110 (0.014)	0.262 (0.020)	0.344 (0.021)	126	6

Table 2: Solutions in the approximation set after 60 iterations of the Bayesian optimisation algorithm.

the MC power estimates often exceed the corresponding nominal bound. For example, the last solution given would have an approximate 98% two-sided confidence interval of $(0.078, 0.143)$. However, when the GP model is used to predict power instead of the raw MC estimate, the interval is $(0.077, 0.099)$. To verify this prediction we computed a more precise MC estimate using $N = 10,000$ samples, which gave a interval of $(0.088, 0.102)$.

6 Simulation study

Algorithm 1 is stochastic and will return a different final approximation set each time it is applied to the same problem. To assess the variability in the quality of these sets, we conducted two simulation studies using the two illustrative examples of Section 2. For each problem the algorithm was applied 1000 times. Each application was as described in Section 5, although now using 200 iterations as opposed to 60.

We recorded the approximation sets $\mathcal{A}_i^{(j)}$ found at iteration $i \in [1, 200]$ for each run $j \in [1, 1000]$. Recall that the quality of an approximation set $\mathcal{A}_i^{(j)}$ is measured by the volume of the dominated subspace $H(\mathcal{A}_i^{(j)})$ as compared with that of the Pareto set $\mathcal{X}_p$:

$$H(\mathcal{X}_p) - H(\mathcal{A}_i^{(j)}), \tag{25}$$

As $\mathcal{X}_p$ is unknown, we instead use the set of non-dominated solutions in the union of the final approximation sets produced by the 1000 applications of the algorithm, $\mathcal{X}_p \approx \cup_{j=1}^{1000} \mathcal{A}_{200}^{(j)}$.

For comparison, we implemented an alternative method based on a fixed experimental design. To compare against the algorithm after i iterations, the average time taken to reach iteration i over the 1000 runs was calculated. The number of solution evaluations (using $N = 500$ MC samples) which could be completed in this time was then found. A fixed space-filling experimental design[3] of this size, $\mathcal{X}_d$, was then generated. For each problem, the set $\mathcal{X}_d$ was chosen from a restricted region of the full solution space. Specifically, in the

[3] We used a Sobol sequence, which has the convenient property that the sequence of size $i + 1$ contains the sequence of size i for all $i > 0$.

cross-classified example we restricted solutions to those with balance between the arms, i.e. $r = 1$, and in the pilot trial example to those with balance between arms and clusters, i.e. $m_1 = m_2$ and $k_1 = k_2$. These simple heuristics were designed to allow the set $\mathcal{X}_d$ to more fully explore the fundamental parameters of total sample size and total number of clusters[4].

Each solution in $\mathcal{X}_d$ was evaluated using $N = 500$ samples. Approximate 98% two-sided confidence intervals for each constraint function value were calculated. As with Algorithm 1, a constraint was considered to be satisfied if the upper confidence interval was within the nominal value. The approximation set of non-dominated feasible solutions was then identified.

6.1 Cross-classified psychotherapy trial

Figure 8 illustrates the distribution in the quality of approximation sets found at different stages of the optimisation algorithm. Both variability and median dominated hypervolume reduce steadily as the number of iterations increases, with the rate of change being most extreme in the initial stages of the search. Before the iterative portion of the optimisation algorithm starts, the majority of runs return approximation sets of worse quality than the matched fixed design method. However, as the number of iterations increase the optimisation algorithm consistently out-performs the fixed experimental design method.

A random selection of 10 approximation fronts with 0 iterations completed is illustrated in Figure 9. The approximation sets from the same 10 runs are shown again in Figure 10, now after 60 iterations of the optimisation algorithm. Average quality has clearly increased, although at the cost of computation time - on average, 4 hours and 38 minutes were required to reach 60 iterations, whereas 1 hour and 2 minutes were needed to evaluate the initial experimental design. However, note that the benefits of the Bayesian optimisation algorithm over the fixed experimental design become clear by this point. Comparing the approximation set of the first run (bold red in Figure 10) with that of the fixed design in terms of the numbers of participants required (i.e. objective f_1), we see that the algorithm requires between 24 and 396 fewer participants for the same number of therapists. Moreover, it provides solutions requiring as few as 6 therapists, whereas the fixed design does not provide an option for less than 7. As the number of iterations increases to 200, illustrated in Figure 11, the average computation time increases to over 13 hours but the improvement in solution quality is limited.

6.2 Cluster-randomised pilot trial

We now consider the second motivating example. The quality of approximation sets, for increasing number of iterations, is plotted in Figure 12. The results of the fixed experimental design are shown for comparison.

[4] We replicated the studies without these restrictions and found inferior performance - see appendix.

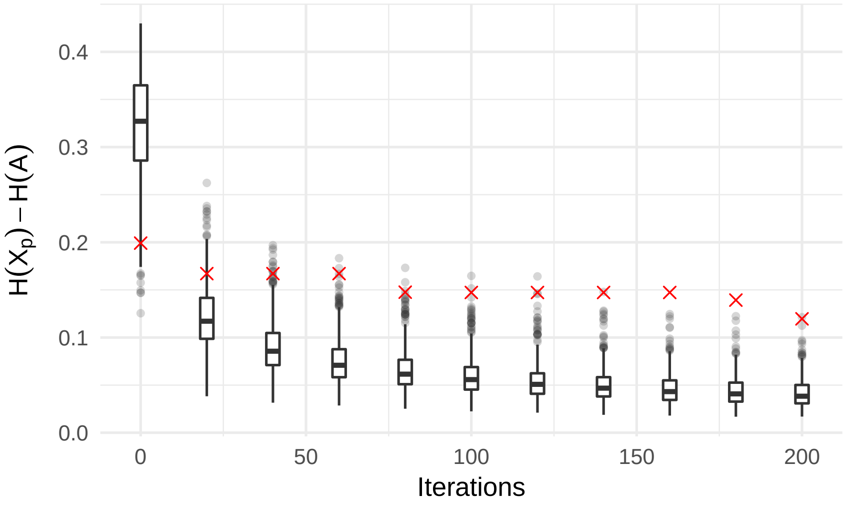

Figure 8: Box plots illustrating the differences in dominated hypervolumes obtained as the number of iterations in the Bayesian optimisation algorithm increases. Corresponding differences obtained using the fixed experimental design are shown in red.

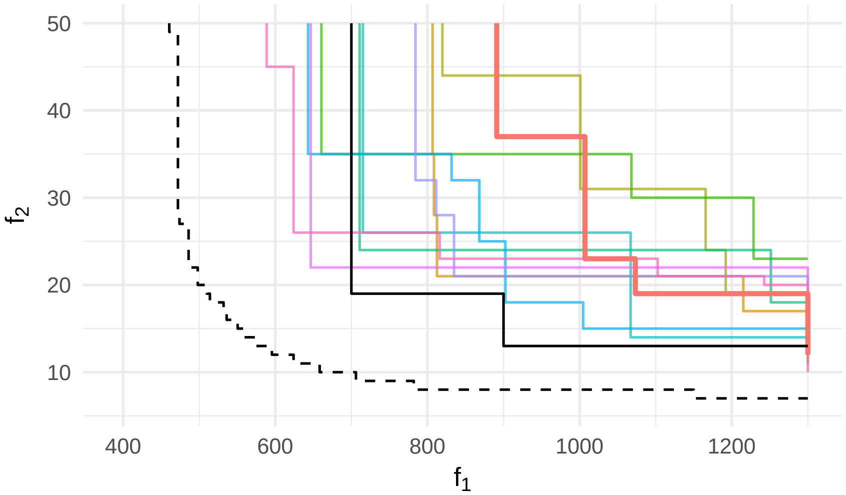

Figure 9: Random selection of approximation fronts obtained after 0 iterations of the Bayesian optimisation algorithm (coloured lines) for the cross-classified problem, together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line). Time taken - 1 hr 2 mins.

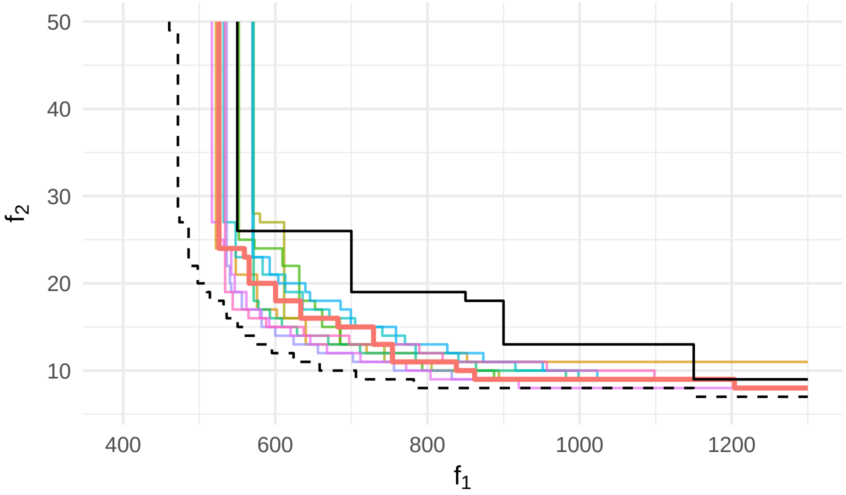

Figure 10: Random selection of approximation fronts obtained after 60 iterations of the Bayesian optimisation algorithm (coloured lines) for the cross-classified problem, together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

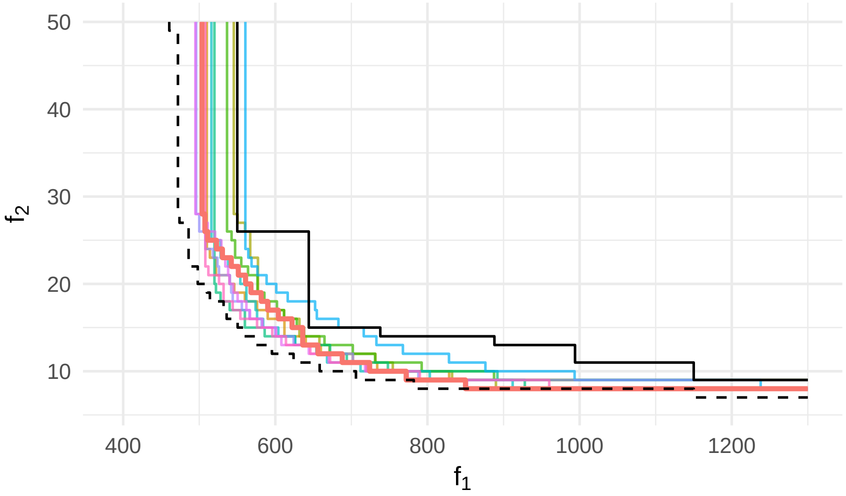

Figure 11: Random selection of approximation fronts obtained after 200 iterations of the Bayesian optimisation algorithm (coloured lines) for the cross-classified problem, together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

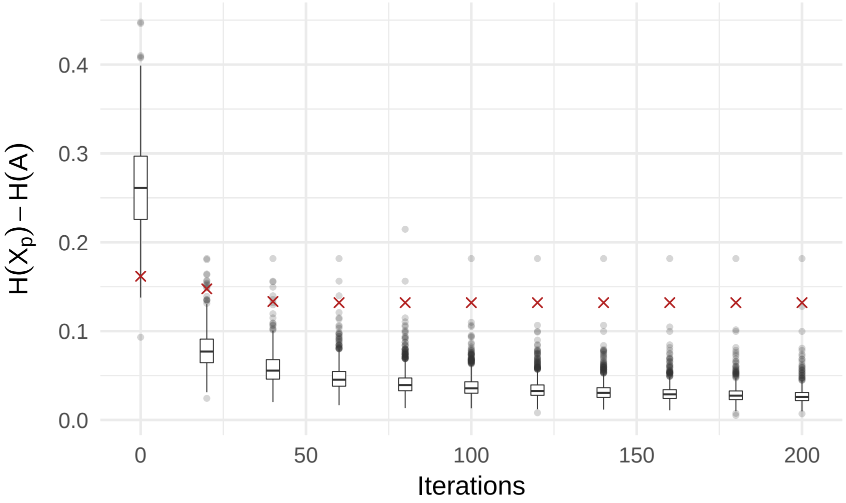

Figure 12: Box plots illustrating the differences in dominated hypervolumes obtained as the number of iterations in the Bayesian optimisation algorithm increases. Corresponding differences obtained using the fixed experimental design are shown in red.

At the start of the iterative portion of the Bayesian optimisation algorithm, Figure 12 shows substantial variability in the quality of the approximation set obtained. A random selection of 10 approximation fronts with 0 iterations completed is illustrated in Figure 13.

The quality of solutions increases substantially as the number of iterations is increased, with the rate of improvement levelling off quickly. The approximation sets from the same 10 runs of Figure 13 are shown again in Figure 14, now after 60 iterations of the optimisation algorithm. Average quality has clearly increased, although at the cost of computation time - on average, 2 hours and 45 minutes were required to reach 60 iterations, whereas only 19 minutes were needed to evaluate the initial experimental design. However, note that the benefits of the Bayesian optimisation algorithm over the fixed experimental design are clear. Comparing the approximation set of the first run (bold red in Figure 14) with that of the fixed experimental design in terms of the numbers of participants required, we see that the optimisation algorithm leads to solutions which require between 14 and 48 fewer participants for the same number of clusters. Moreover, it provides solutions requiring as few as 6 clusters, whereas the fixed design does not provide options for fewer than 8. As the number of iterations increases to 200, illustrated in Figure 15, the average computation time increases to over 9 hours but the improvement in solution quality is limited.

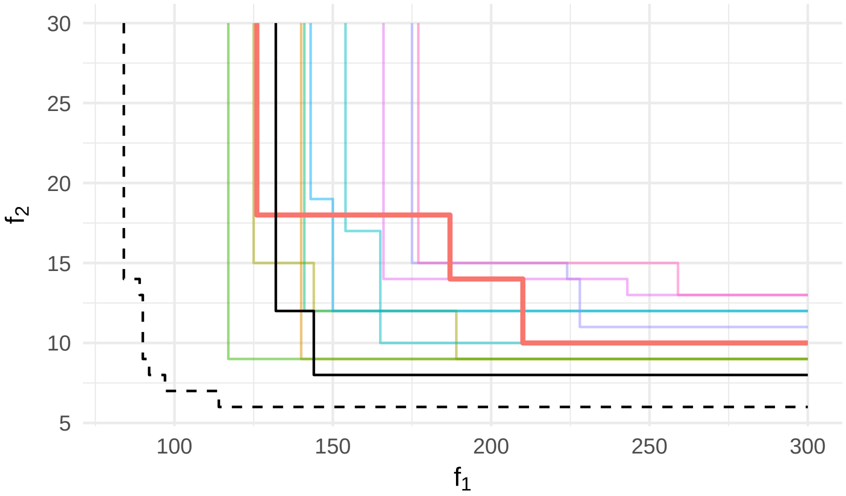

Figure 13: Random selection of approximation fronts obtained after 0 iterations of the Bayesian optimisation algorithm (coloured lines), together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

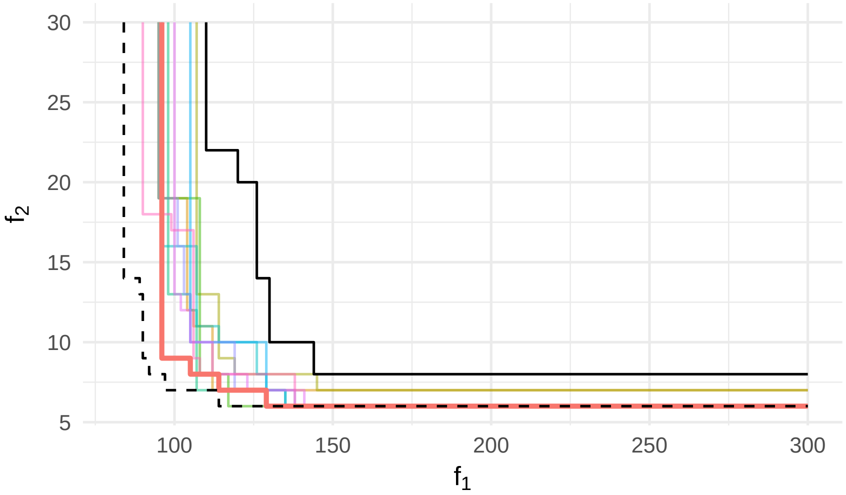

Figure 14: Random selection of approximation fronts obtained after 60 iterations of the Bayesian optimisation algorithm (coloured lines), together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

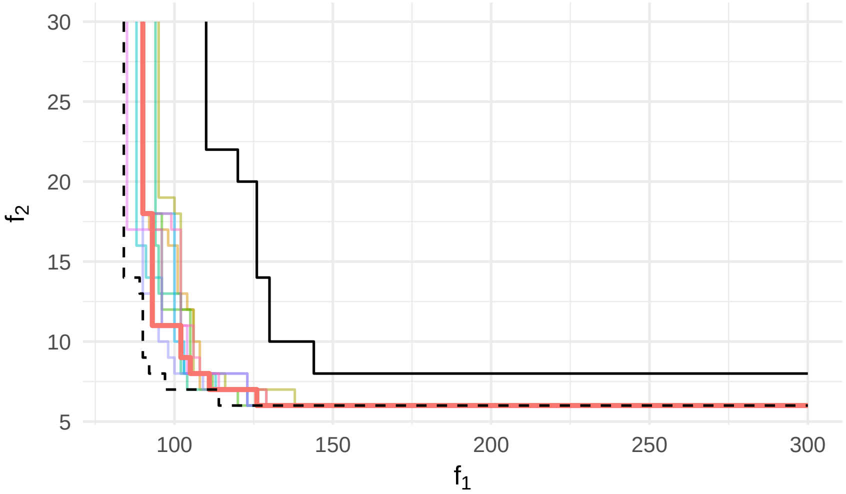

Figure 15: Random selection of approximation fronts obtained after 200 iterations of the Bayesian optimisation algorithm (coloured lines), together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

7 Discussion

Simulation-based design is often required for complex trials, yet the associated computational burden has led to the associated optimal design problems being simplified to enable a timely solution. These simplifications include fixing a number of design parameter values so that optimisation can take place over a single dimension and with respect to a single objective. Methods and associated software for simulation-based design are available for such problems [27, 19], but the more general case has yet to be addressed. In this paper we have explored how surrogate models of operating characteristics and associated algorithms for efficient global optimisation can be used to facilitate simulation-based design of clinical trials where there are several parameters to be chosen, several objectives to minimise, and several constraints to satisfy.

The general optimisation framework we have suggested recognises that in many complex trials we are interested in minimising more than one quantity subject to constraints on operating characteristics. Problems of this sort are common in multilevel trial design [17], but are typically approached by first reducing the multiple objectives down to a single objective. For example, the design of a cluster randomised trial is often approached by fixing the number of patients per cluster and then minimising the number of clusters [11], or vice-versa [18, 12]. Alternatively, a function which specifies the cost of sampling at the cluster and the patient level could be specified [21, p. 175], and the overall cost minimised [41]. The latter approach has been suggested for both two-level [34] and three-level hierarchical trial designs [44, 43]. However, the *a priori* specification of such a cost function may not always be feasible, particularly when several stakeholders are involved. The Pareto optimisation framework we have described leads to a more computationally challenging optimisation problem, but produces solutions which will be more helpful to decision makers, enabling the available trade-offs between objectives to be seen and selected from. Multiple objectives have also arisen in adaptive designs, which can aim to minimise the expected sample size under several different hypotheses. Although several authors have described methods for finding the Pareto set (or equivalently, the set of admissible designs) in this setting [24, 25], the methods used thus far have been inefficient, solving a large number of single-objective optimisation problems using different weighted combinations of objectives [29]. Using multi-criteria optimisation algorithms such as the benchmark NSGA-II [10], as implemented in the R package mco [30], would improve the efficiency of these methods.

As noted in Section 1, related work in simulation-based design methodology has often focussed on a specific area of application. In this paper we have taken the view of [19] and proposed a general framework which can be applied to almost any problem where the trial data generation and analysis can be simulated. It is argued in [26] that requiring the user to specify their own simulation program can be prohibitive in practice. One way to address this limitation is to share example programs for a range of problems, providing a starting point for the development of a new program. We have provided two such examples in the appendix.

When designing a clinical trial using simulation, replicating the analysis of the proposed design operating characteristics is more challenging than in the conventional situation. The simulation program may be supplied to the reviewer for inspection, and given the program, any MC estimates of operating characteristics can be re-generated. In addition to this verification of the program, validation should then be completed by assessing the simulation code to ensure it is an accurate representation of the model in question. This does, however, assume a level of familiarity with the programming language used to write the simulation program. This has partly motivated our use of R. Although significantly slower than a compiled language such as C++, it has been argued that software written in R is more transparent [40]. Validation will be further facilitated ff a simulation protocol of the sort described in [40] is provided alongside the code. Nevertheless, to an applied statistician more familiar with Stata or SAS, understanding and using an R program may require significant time and effort. Future work could focus on developing an interface with these statistical languages, allowing a simulation program to be written in them and connect with the R implementation of the optimisation algorithm through a suitable interface.

The examples have demonstrated that the time required to converge to a high quality approximation set can be significant, of the order of hours. Given that the majority of computational effort is expended generating MC samples when evaluating solutions, it is important that these simulation programs are as efficient as possible. Code profiling, used to identify areas which are consuming the most resources and making them more efficient, should be conducted. Future work could focus on using more sophisticated methods for surrogate modelling and efficient optimisation, such as imposing monotonicity conditions as described in [13], to further improve computation time.

We have followed the conventional approach to clinical trial design whereby constraints on operating characteristics are set and then a constrained optimisation problem is solved. In practice, the constraints are not fixed in advance, but rather adjusted iteratively in response to the design requirements they produce. Such an iterative procedure will clearly further increase an already substantial computational burden for simulation-based design. However, note that a change to a constraint does not mean starting the design process again. Any simulation data generated can still be used when fitting the GP model. An interactive routine whereby the user adjusts the constraints in response to the observed design requirements could be developed.

We have chosen to model each constraint function using a separate GP regression model. An alternative approach is to specify a single GP model which jointly models all constraint functions simultaneously, known as 'co-krigging'. However, it has been noted that this is prone to error and may not improve efficiency significantly []. A further alternative is to incorporate constraints into objective functions as penalisation terms, after which GP models of the penalised objectives could be modelled and a unconstrained optimisation problem solved. Disadvantages of this approach include greater difficulty in interpreting the GP models and thus diagnosing a poor fit, and the fact that the resulting function being modelled will be less smooth, with a large step change over the region of the constraint boundary, which may be more challenging to model well. We could consider other regression methods other than GPs, including parametric models. While parametric regression models facilitate interpretation and, they are less flexible and more suited to global fitting, as opposed to the local fitting required in the optimisation problem. Moreover, other modelling approaches do not provide the formal quantification of uncertainty in prediction, and therefore are less amenable for use in global optimisation.

Numerous extensions to the proposed approach can be considered. One argument for simulation-based design is the ease with which sensitvity to model assumptions, such as the value of nuisance parameters, can be assessed [27]. We have not considered such investigations in this paper. An extension to the methodology could consider how a systematic assessment of sensitivity to nuisance parameters could be conducted, given a proposed trial design. Such investigations fall under the heading of *uncertainty quantification* and can be carried out using GP regression and associated techniques [**?**]. A further extension could consider Bayesian approaches to trial design, including hybrid Bayesian-frequentist assurances [31] and fully Bayesian measures such as average coverage criterion [7]. Aside from very simple cases involving only conjugate analyses, evaluating these Bayesian criteria will generally require simulation [31] and so optimal design may benefit from the efficient methods discussed here.

8 Conclusion

Efficient optimisation algorithms based on surrogate models of expensive operating characteristic functions can be used to solve complex problems of clinical trial design. By using these methods we can avoid making unrealistic simplifying assumptions at the trial design stage, both in terms of the statistical model underlying the trial and of the optimal trial design problem.

References

[1] Benjamin F Arnold, Daniel R Hogan, John M Colford, and Alan E Hubbard. Simulation methods to estimate design power: an overview for applied research. *BMC Med Res Methodol*, 11(1), 6 2011.

[2] Gianluca Baio, Andrew Copas, Gareth Ambler, James Hargreaves, Emma Beard, and Rumana Z Omar. Sample size calculation for a stepped wedge trial. *Trials*, 16(1), aug 2015.

[3] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[4] Claus Bendtsen. *pso: Particle Swarm Optimization*, 2012. R package version 1.0.3.

[5] William J Browne, Mousa Golalizadeh Lahi, and Richard MA Parker. *A Guide to Sample Size Calculations for Random Effect Models via Simulation and the MLPowSim Software Package*, 2009.

[6] John Bryant and Roger Day. Incorporating toxicity considerations into the design of two-stage phase II clinical trials. *Biometrics*, 51(4):1372–1383, 1995.

[7] Jing Cao, J. Jack Lee, and Susan Alber. Comparison of bayesian sample size criteria: ACC, ALC, and WOC. *Journal of Statistical Planning and Inference*, 139(12):4111 – 4122, 2009.

[8] Rob Carnell. *lhs: Latin Hypercube Samples*, 2016. R package version 0.14.

[9] Clément Chevalier, Victor Picheny, and David Ginsbourger. KrigInv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 71:1021–1034, mar 2014.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, apr 2002.

[11] Allan Donner and Neil Klar. *Design and Analysis of Cluster Randomization Trials in Health Research*. London Arnold Publishers, 2000.

[12] Sandra M Eldridge, Ceire E Costelloe, Brennan C Kahan, Gillian A Lancaster, and Sally M Kerry. How big should the pilot study for my cluster randomised trial be? *Statistical Methods in Medical Research*, 2015.

[13] Michael TM Emmerich, André H Deutz, and Jan Willem Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE, 2011.

[14] Valerii Fedorov and Byron Jones. The design of multicentre trials. *Statistical Methods in Medical Research*, 14(3):205–248, 2005. PMID: 15969302.

[15] Ziding Feng and James E. Grizzle. Correlated binomial variates: Properties of estimator of intraclass correlation and its effect on sample size calculation. *Statistics in Medicine*, 11(12):1607–1614, 1992.

[16] Andrew P. Grieve and Shah-Jalal Sarker. Simulation-based sample-sizing and power calculations in logistic regression with partial prior information. *Pharmaceutical Statistics*, 15(6):507–516, sep 2016.

[17] K Hemming, S Eldridge, G Forbes, C Weijer, and M Taljaard. How to design efficient cluster randomised trials. *BMJ*, 358, 2017.

[18] Karla Hemming, Alan Girling, Alice Sitch, Jennifer Marsh, and Richard Lilford. Sample size calculations for cluster randomised controlled trials with a fixed number of clusters. *BMC Medical Research Methodology*, 11(1):102, 2011.

[19] Richard Hooper. Versatile sample-size calculation using simulation. *The STATA Journal*, 13(1):21–38, 2013.

[20] Richard Hooper, Steven Teerenstra, Esther de Hoop, and Sandra Eldridge. Sample size calculation for stepped wedge and other longitudinal cluster randomised trials. *Statistics in Medicine*, 35(26):4718–4728, jun 2016.

[21] Joop Hox. *Multilevel Analysis: Techniques and Applications.* Lawrence Erlbaum Associates, Inc., 2002.

[22] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[23] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[24] Sin-Ho Jung, Mark Carey, and Kyung Mann Kim. Graphical search for two-stage designs for phase II clinical trials. *Controlled Clinical Trials*, 22(4):367 – 372, 2001.

[25] Sin-Ho Jung, Taiyeong Lee, Kyung Mann Kim, and Stephen L. George. Admissible two-stage designs for phase II cancer clinical trials. *Statistics in Medicine*, 23(4):561–569, 2004.

[26] Evangelos Kontopantelis, David A Springate, Rosa Parisi, and David Reeves. Simulation-based power calculations for mixed effects modeling: ipdpower in stata. *Journal of Statistical Software*, 74(12), 2016.

[27] Sabine Landau and Daniel Stahl. Sample size and power calculations for medical studies by simulation when closed form expressions are not available. *Statistical Methods in Medical Research*, 22(3):324–345, 2013.

[28] Ellen Lee, Amy Whitehead, Richard Jacques, and Steven Julious. The statistical interpretation of pilot trials: should significance thresholds be reconsidered? *BMC Medical Research Methodology*, 14(1):41, 2014.

[29] Adrian P. Mander, James M.S. Wason, Michael J. Sweeting, and Simon G. Thompson. Admissible two-stage designs for phase II cancer clinical trials that incorporate the expected sample size under the alternative hypothesis. *Pharmaceutical Statistics*, 11(2):91–96, 2012.

[30] Olaf Mersmann. *mco: Multiple Criteria Optimization Algorithms and Related Functions*, 2014. R package version 1.0-15.1.

[31] Anthony O'Hagan, John W. Stevens, and Michael J. Campbell. Assurance in clinical trial design. *Pharmaceutical Statistics*, 4(3):187–201, 2005.

[32] Victor Picheny and David Ginsbourger. Noisy kriging-based optimization methods: A unified implementation within the DiceOptim package. *Computational Statistics & Data Analysis*, 71:1035–1053, mar 2014.

[33] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[34] Stephen W. Raudenbush and Xiaofeng Liu. Statistical power and optimal design for multisite randomized trials. *Psychological Methods*, 5(2):199–213, 2000.

[35] Nicholas G. Reich, Jessica A. Myers, Daniel Obeng, Aaron M. Milstone, and Trish M. Perl. Empirical power and sample size calculations for cluster-randomized and cluster-randomized crossover studies. *PLOS ONE*, 7(4):1–7, 04 2012.

[36] Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[37] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[38] Michael J. Sasena, Panos Papalambros, and Pierre Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 34(3):263–278, jan 2002.

[39] David A. Schoenfeld and Michael Borenstein. Calculating the power or sample size for the logistic and proportional hazards models. *Journal of Statistical Computation and Simulation*, 75(10):771–785, oct 2005.

[40] Mike K. Smith and Andrea Marshall. Importance of protocols for simulation studies in clinical drug development. *Statistical Methods in Medical Research*, 2010.

[41] Tom A. B. Snijders and Roel J. Bosker. Standard errors and sample sizes for two-level research. *Journal of Educational and Behavioral Statistics*, 18(3):237–259, 1993.

[42] Alexander J. Sutton, Nicola J. Cooper, David R. Jones, Paul C. Lambert, John R. Thompson, and Keith R. Abrams. Evidence-based sample size calculations based upon updated meta-analysis. *Statistics in Medicine*, 26(12):2479–2500, 2007.

[43] S. Teerenstra, M. Moerbeek, T. van Achterberg, B. J. Pelzer, and G. F. Borm. Sample size calculations for 3-level cluster randomized trials. *Clinical Trials*, 5(5):486–495, sep 2008.

[44] Gerard J.P. van Breukelen and Math J.J.M. Candel. Calculating sample sizes for cluster randomized trials: We can keep it simple and efficient! *Journal of Clinical Epidemiology*, 65(11):1212 – 1218, 2012.