0968979556

Efficient and flexible simulation-based design of complex clinical trials

Abstract

[NOTE - COPY OF SCT / ICTMC ABSTRACT, TO REVISE] **Background**: Finding the optimal sample size for a trial is an important step in its design. In many trials of complex interventions (such as psychotherapies and behavioural interventions) this task is complicated by two factors. Firstly, sample size can be defined by several design parameters rather than a single n. For example, the TIGA-CUB trial compares a psychotherapy intervention with treatment as usual. The design is partially nested, with patients are nested within therapists in the intervention arm but not in the control arm. The design parameters are the number of therapists in the intervention arm, the number of patients seen by each therapist, and the number of patients in the control arm. The second complication is that analytical formulae for calculating power are not always available. As a result power must instead be estimated through Monte Carlo simulation methods, which may be computationally demanding. For example, such a simulation of the TIGA-CUB study would involve the generation of a large number of hypothetical trial data sets and fitting a multilevel model to each one. In combination, these factors make finding an optimal sample size a difficult and time consuming problem. We explore how modern optimisation algorithms can be used to solve these problems in an effective, timely manner.

Methods: We propose using Bayesian optimisation to solve the sample size problem. This method allows optimal or near-optimal choices for sample size to be found with minimal computational effort. The general approach involves the careful choice of the design parameter values where power should be estimated using simulation. Conducting the simulations at these points, a statistical model is then fitted to the output to describe the general relationship between the design parameters and the trial power. This model is then used to find the smallest design parameter values which will give power of at least the nominal level. The method is flexible, can be used for almost any problem for which power can be estimated using Monte Carlo simulation, and can be implemented using existing statistical software packages.

Evaluation: To illustrate the approach we apply it to the TIGA-CUB trial in an illustrative case study. We use the proposed method to identify a set of candidate sample size options, each of which will give power of at least the nominal rate. From this set of options, that which is considered best in terms of its balance between number of therapists and number of

patients can be chosen. We compare this with an alternative approach using simpler heuristics, in terms of both the computation time required and the quality of the resulting solutions.

Conclusions: Bayesian optimisation can be an effective technique for performing sample size calculations when power must be estimated using simulation, particularly when sample size is characterised by several design parameters. By improving the efficiency of these calculations, increasingly complex sample size problems can be solved without the need for unrealistic simplifying assumptions.

1 Introduction

Include:

[25] - Empirical Power and Sample Size Calculations for Cluster-Randomized and Cluster-Randomized Crossover Studies [**?**] - Sample size and power calculations using the noncentral t-distribution [**?**] - Simulated Power Functions

[33] - Evidence-based sample size calculations based upon updated meta-analysis. Simulation approach, "Although for simple situations closed form solutions are often possible,for more complex analyses, including those that include random effects, closed form solutionsoften do not exist and simulation methods become an appealing alternative". Refers to [10] for a general guide. Later argues that even if closed form solutions could be derived, the flexibility of the simulation approach is preferable.

The optimal design of a clinical trial is typically framed as a constrained optimisation problem: choose the design from within a certain family of designs 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 and/or analysis mean that simple analytical expression of the type II error rate (or equivalently, of the power) are not available. In these cases it has been suggested that a Monte Carlo (MC) estimate of power be used instead [20]. 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. This strategy is both conceptually simple and highly flexible, being applicable to almost any trial design and method of analysis. As a result, simulation-based power calculation and sample size determination have been advocated in applied and medical research [1, 20].

The flexibility of the simulation-based design methodology has been illustrated through a variety of application in a broad range of settings. Examples include designs involving hierarchical models [13], proportional hazards models [28], logistic regression models [11], individual patient data meta-analyses [19], and stepped wedge designs [2, 14]. Nevertheless, challenges to its wider uptake remain.

The benefits of the simulation method come at the cost of computation time.

Simulating a sufficiently large [1] number of data sets and carrying out the analysis on each one can often take several minutes. In simple cases, finding the sample size which provides sufficient power can nevertheless be conducted reasonably quickly. In particular, problems which have only one design variable (e.g. the group size in a two-arm balanced RCT), with one quantity to be minimised (e.g. the total sample size), subject to a single constraint (e.g. power under the alternative hypothesis), can be solved with only a handful of iterations [2]. As the complexity of the design problem grows, however, a standard optimisation algorithm will often require thousands of iterations before converging on a optimal (or near-optimal) solution. When analytic power expressions are not available and MC estimates are being used in their place, this process can take days.

Many trial design problems are more complicated. For example, the two-stage phase II trial design for a single binary endpoint proposed in [30] is defined by four variables: the number of patients in stage 1; the number of responses which must be observed to proceed to stage two; the number of patients in stage 2; and the total number of responses which must be observed to proceed to a phase III trial. Their optimal values are those which minimise the total sample size of the study whilst ensuring type I error and power are within nominal bounds. In contrast to a single variable problem, a multi-variable problem involves a large number of candidate designs which must be optimised over.

Often there are several distinct quantites which we wish to minimise when designing a trial. For example, in many cluster randomised trials we wish to minimise both the number of clusters and the number of patients. This problem is typically dealt with by fixing the number of patients per cluster and then minimising the number of clusters [8], or vice-versa [12]. Alternatively, a function which specifies the cost of sampling at the cluster and the patient level could be specified, and the overall cost minimised [32]. The latter approach has been suggested for both two-level and three-level hierarchical trial designs [35, 34]. The *a priori* specification of such a cost function may not always be feasible. An alternative approach to addressing multiple objectives is to search not for a single optimal design, but for a set of designs which offer a range of trade-offs between the objectives. Specifically, we aim to find designs which are *Pareto optimal*, meaning that there is no other design which is better in at least one aspect whilst being at least as good in all others[3]. The search for a set of designs, as opposed to a single optimum, will increase the computational cost of

[1] Typical recommendations for the number of simulations required to estimate power are in the range of 10^3 to 10^5.

[2] A simple bisection search algorithm will require no more than $\lfloor \log_2 n_{max} + 1 \rfloor$ iterations where n_{max} is the maximum feasible value of the variable. For example, given an upper limit on the sample size of $n_{max} = 800$, at most 10 MC evaluations will be required to find the optimal sample size.

[3] An equivalent concept in statistical decision theory is an *admissible* design. In the context of multi-stage designs, where there is an interest in balancing the trial's expected sample size and its largest possible sample size, several authors have proposed that a set of admissible designs should be found and that which offers the most agreeable trade-off between the objectives should then be selected [17, 21]

the design problem.

Finally, there may be several hypotheses at which we wish to constrain power to be within some nominal threshold, with each of these estimated using simulation. This will be the case whenever the type I error rate cannot be deduced analytically, as in the multi-stage adaptive design proposed in [37]; or when multiple outcomes lead to multiple alternative hypotheses, as in the two-stage phase II design proposed in [29]. With each rejection rate to be estimated the time needed to evaluate the properties of a single design will increase, and the design problem will become yet more computationally demanding.

While there are a considerable number of software packages for sample size calculation and trial design, few consider problems where evaluations require computing MC estimates. Where these have been developed, they have typically focussed on a specific area of application. MLPowSim [4] 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 [13] is more general, allowing the user to fully specify their own data generation and analysis models. It has been applied to multilevel designs [13], designs for logistic regression [11], and stepped wedge designs [14]. 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. Out-with 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. One family of methods address the problem by substituting the expensive function with an approximation known as a *surrogate model* (or *emulator*). 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. In this paper we will show how complex simulation-based clinical trial design problems can also be addressed using a particular type of surrogate modelling, namely Gaussian process regression, and associated efficient optimisation algorithms [15]. We consider trials where the primary analysis will test a simple hypothesis, and where the alternative hypothesis is also simple. We will not impose any restrictions regarding the nature of the endpoint, 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. In Section 2 we briefly provide the necessary background and notation regarding Monte Carlo estimation and optimisation. Two motivating trial design problems are then described in Section 3. In Section 4 we describe Gaussian process regression, an efficient global optimisation algorithm, and a framework for their application to trial

design problems. We return to the examples in Section 5, illustrating how the framework can be applied in practice. We conclude with a discussion of the implications and limitations of the proposed approach in Section 6.

2 Background and notation

2.1 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, $i = 1, \ldots, N$, the MC estimate is

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

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

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

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.

$$\frac{1}{N} \sum_{i=1}^{N} f(z_i) = \mathbb{E}[f(Z)] + e, \text{ where } e \sim N(0, \omega^2). \tag{3}$$

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 $\mathbb{E}[I(Z \in \Lambda) \mid H]$. Note that this equates to the power of the test if $H = H_1$, the alternative hypothesis, or the type I error rate of the test if $H = H_0$, the null hypothesis. Thus, an MC estimate of the power of a trial design can be obtained given N sampled test statistics $z_1, \ldots, z_N$. The steps required to simulate these statistics are described in [20], 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.

2.2 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 trial designs. As mentioned in Section 1, a simple design problem may have a 1-dimensional solution space, while more complex problems may have several dimensions. Elements of $\mathbf{x}$ will generally describe the sample size and the acceptance region of the trial.

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

A *constraint function* $g(\mathbf{x})$ is a real-valued function $g : \mathcal{X} \to \mathbb{R}$ 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 I error rate is set at β^*, a constraint function would be $g(\mathbf{x}) = \beta(\mathbf{x}) - \beta^*$. We denote C constraint functions as $g_j(\mathbf{x}),\ j = 1, \ldots, C$.

The problem of optimal trial design can now be stated as

$$\min_{\mathbf{x} \in \mathcal{X}} \ f_i(\mathbf{x}),\ i = 1, \ldots, B \tag{4}$$

$$\text{subject to } g_j(\mathbf{x}) \leq 0,\ j = 1, \ldots, C. \tag{5}$$

We denote by $\prec$ the relation of Pareto dominance: $\mathbf{x}_* \prec \mathbf{x}$ if $f_i(\mathbf{x}_*) \leq f_i(\mathbf{x})$ for $i = 1, \ldots, B$, and $f_j(\mathbf{x}_*) < f_j(\mathbf{x})$ for some specific objective j. The Pareto set is the set of non-dominated solutions $\mathcal{X}_p = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{x}_* \in \mathcal{X} s.t. \mathbf{x}_* \prec \mathbf{x}\}$. The Pareto front is the corresponding set of objective values $\mathcal{Y}_p = \{\mathbf{y} \in \mathbb{R}^B \mid \mathbf{x} \in \mathcal{X}_p\}$.

Following [9], we define an approximation set A to be any set $A \in \mathcal{X}$ with $\forall \mathbf{x} \in A : \nexists \mathbf{x}_* \in A : \mathbf{x}_* \prec \mathbf{x}$. A set A is feasible if all constraints are satisfied by every member of A. The hypervolume of set A is the volume of the subspace dominated by solutions in A and bounded by a reference point r:

$$Vol(\{\mathbf{y} \in \mathbb{R}^C \mid \mathbf{y} \text{ is dominated by some } \mathbf{y}_* \in A \text{ and } \mathbf{y} \prec r\}). \tag{6}$$

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

Example Pareto and approximation sets are shown in Figure **??** for a psychotherapy trial where we wish to minimise both the number of patients and the number of therapists, subject to a power constraint. Taking as a reference point $r = (1000, 50)$, $H(\mathcal{X}_p) = 1$, while $H(A_1) = 2, H(A_2) = 3$.

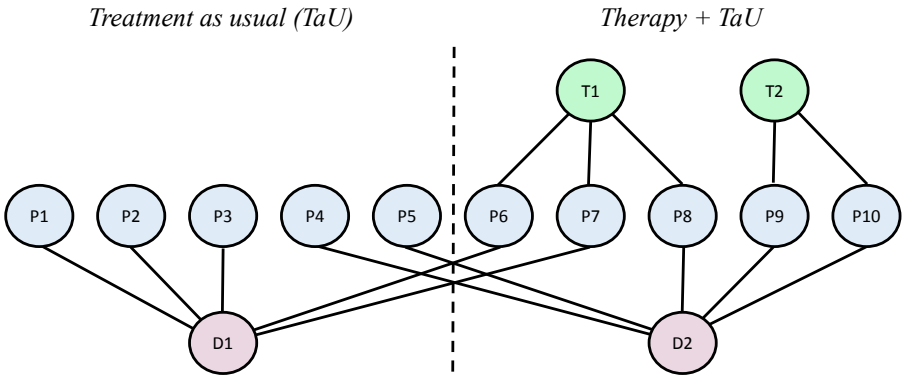

Figure 1: Multilevel structure of a cross-classified psychotherapy trial where patients are nested within doctors in the control arm, patients are cross-classified with therapists and doctors in the intervention arm, and doctors are crossed with treatment.

3 Motivating examples

3.1 Cross-classified psychotherapy trial

We 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. We will denote the average cluster size by $\bar{m}_T$, and the coefficient of variation of cluster size by cv_T. 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 population model describes the continuous primary outcome measure of patient i nested within therapist j and doctor k as

$$y_{ijk} = \beta_0 + \beta_1 t_i + t_i u_j + v_k + e_i, \tag{7}$$

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 be a Wald test of the null hypothesis $H_0 : \beta_1 = 0$. Specifically, we fit model (7) using maximum likelihood and obtain an estimate $\hat{\beta}_1$ and its standard error $s.e.(\hat{\beta}_1)$. The test statistic is $\hat{\beta}_1/s.e.(\hat{\beta}_1)$, which is assumed to follow a normal distribution under H_0. Under this assumption, the type I error rate can be controlled at the nominal level of $\alpha^* = 0.025$ (one sided).

The solution space is defined by the trial design parameters r, n_2, k and j. For simplicity, we will assume that the number of doctors is fixed at $j = 50$. The bounds on the remaining design parameters are taken to be $r \in [0.5, 1]$, $n_2 \in [200, 1000]$, and $k \in [5, 40]$. 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 $g_1(\mathbf{x}) = \beta(\mathbf{x}) - 0.2$. To the best of our knowledge there are no analytic formulae which calculate the power of a trial under the model (7), and so we instead use MC estimates.

3.2 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.

The trial is cluster-randomised at the care home level, 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 the arm $i = 1, 2$ is denoted $p_r^{(i)}$. The probability of adherence is denoted p_a. We anticipate variability in the response outcome between clusters, expected to be higher 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

$$a^{(1)} = \frac{50 p_r^{(1)}}{(1 - p_r^{(1)})}, \quad b^{(1)} = 50 \tag{8}$$

in the control arm, and

$$a^{(2)} = \frac{100p_r^{(2)}}{(1-p_r^{(2)})}, \quad b^{(2)} = 100 \tag{9}$$

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

Efficacy will be analysed through a t-test of the null hypothesis that the mean difference in response rate between the two arms, $p_r^{(2)} - p_r^{(1)}$, is equal to 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 and at the cluster means level.

We define the operating characteristics of interest in a manner following the design of [5] for two dichotomous endpoints. If A denotes the event where the result of both t-tests is to reject the null hypothesis, we aim to control

$$\alpha_r = Pr[A \mid H_{0,1}] \leq \alpha_r^* = 0.3 \tag{10}$$
$$\alpha_a = Pr[A \mid H_{1,0}] \leq \alpha_a^* = 0.4 \tag{11}$$
$$\beta = Pr[A \mid H_{1,1}] \geq \beta^* = 0.1, \tag{12}$$

where

$$H_{0,1} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.3, p_a = 0.8 \tag{13}$$
$$H_{1,0} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.5, p_a = 0.6 \tag{14}$$
$$H_{01,1} : p_r^{(1)} = 0.3, p_r^{(2)} = 0.5, p_a = 0.8. \tag{15}$$

Under all hypotheses the odds ratio $OR = 10$, indicating that adherence and response are positively associated. The extent of clustering of response outcomes can be described through the coefficient of variation. In the control arm this is 0.18, while in the intervention arm it is 0.13 under $H_{0,1}$ and 0.07 under $H_{1,0}$ and $H_{1,1}$.

The correlation between outcomes and the between-cluster variability in response rates implies that the assumptions underlying the proposed t-tests may not hold. Moreover, the t statistics will be positively correlated. We therefore compute MC estimates of the power under each of the three hypotheses.

4 Methods

4.1 Fixed experimental design

A simple and flexible approach to obtaining an approximation set as defined in Section 2 is to pre-specify an experimental design $\mathcal{X}_D \subset \mathcal{X}$ and estimate the relevant operating characteristics for each $\mathbf{x} \in \mathcal{X}_D$. Discarding those solutions which do not satisfy the constraints, all remaining non-dominated solutions can easily identified and thus an approximation set found.

Many methods for choosing an appropriate design $\mathcal{X}_D$ are available. A simple 'grid' of evenly-spaced points, as used in the MLPowSim software, will generally perform poorly compared to a space-filling design such as a Latin Hypercube sample **??**. Here we will use a Sobol sequence of size d, denoted $\mathcal{S}_d$. This experimental design has the property that $\mathcal{S}_{d-1} \subset \mathcal{S}_d$ for all $d > 1$, which will be be helpful in the simulation study of Section 5.

4.2 Bayesian optimisation

To solve trial design problems of the type set out in Sections 2 and 3 in a reasonable amount of time, it is important that at each step in an iteritive optimisation process the next point to be evaluated is chosen with great care. One way to encourage this is to use the Efficient Global Optimisation (EGO) algorithm, origninaly proposed in [16]. The central idea is to fit a non-parametric regression model which approximates the true underlying function of interest, f, and to use this model to describe our uncertainty about the value of the function at solutions which have not yet been evaluated. We can then optimise 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 $\mathbf{x}$?'. The general algorithm can be outlined as follows:

Algorithm 1 Efficient Global Optimisation

1: Evaluate the expensive function at an initial set of points $\mathcal{X}_E$, obtaining a vector $\mathbf{y_E}$ of estimates;
2: **while** Computation budget not exhausted **do**
3: Fit non-parametric regression model to $\mathbf{y_E}$ and $\mathcal{X}_E$
4: Find the point $\mathbf{x}$ which will give the largest *expected improvement* when compared with the best solution found so far, according to the regression model
5: Evaluate f at $\mathbf{x}$ to obtain $\mathbf{y}$, and add these to $\mathcal{X}_E$ and $\mathbf{y_E}$ respectively;
6: Update the computational budget
7: **end while**

As previously discussed, steps (1) and (5) of the algorithm involves estimating the function of interest using a computationally demanding MC method. 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 in step (4) will then be defined precisely for the constrained multi-objective problems we are concerned with. Finally, we will cover the remaining aspects of implementation.

4.2.1 Gaussian process regression

Consider a set of E points $\mathcal{X}_E = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(E)}\} \subset \mathcal{X}$ at which an expensive function f 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 $f(\mathbf{x}_*)$. The value of the function f at each solution in $\{\mathcal{X}_E, \mathbf{x}_*\}$ is initially unknown, but can be modelled by a Gaussian process (GP). GP's provide a highly flexible representation of the unknown function, and are suitable whenever the function is smooth in its inputs. In using a GP we assume that our belief regarding the the values of f can be represented as a multivariate normal distribution. Prior to computing any estimates of f, we assume that the mean function of this multivariate normal is equal to zero[4]. 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 estimates $\mathbf{y} = y^{(1)}, \ldots, y^{(E)}$ at each point in $\mathcal{X}_E$. From equation (3), $y^{(i)} = f(\mathbf{x}^{(i)}) + e^{(i)}$ where $e^{(i)} \sim N(0, \omega^{2(i)})$ and $\omega^{(i)}$ is the MC error. We denote by Δ the diagonal matrix where the ith entry is $\omega^{(i)}$. The distribution of $f(\mathbf{x}_*)$ *conditional* on the observed $\mathbf{y}$ can be shown to be normal with mean $\mathbf{k}_*^\top (K + \Delta)^{-1} \mathbf{y}$ and variance $k(x_*, x_*) - \mathbf{k}_*^\top (K + \Delta)^{-1} \mathbf{k}_*$ [24]. Thus, given a prior covariance matrix of the form (16) and some MC estimates $\mathbf{y}$ of f at the points $\mathcal{X}_E$, a conditional predictive distribution of $f(\mathbf{x}_*)$ can be found. It is this conditional distribution which will be used in the optimisation algorithm when deciding which solution should next be evaluated.

The conditional distributions of interest are dependant on the entries of the prior covariance matrix (16). We now consider how these can be determined. 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. We can ensure this by choosing a function from a parametric family which is known to have these properties. Several such families exist, each having different characteristics. In particular, the choice of function family can influence how smooth the GP model will be. We follow the recommendation of [26] and use functions from the Matern 5/2 family. We actually require a separate covariance function for each of the D dimensions in $\mathcal{X}$. The function for the jth dimension takes the form

$$k(x_j, x_j') = \left(1 + \frac{\sqrt{5}|x_j - x_j'|}{\lambda_j} + \frac{5|x_j - x_j'|^2}{3\lambda_j^2}\right) \exp\left(-\frac{\sqrt{5}|x_j - x_j'|}{\lambda_j}\right). \tag{17}$$

These functions are then combined to give an aggregate covariance function

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{j=1}^{D} k(x_j, x_j'; \lambda_j). \tag{18}$$

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

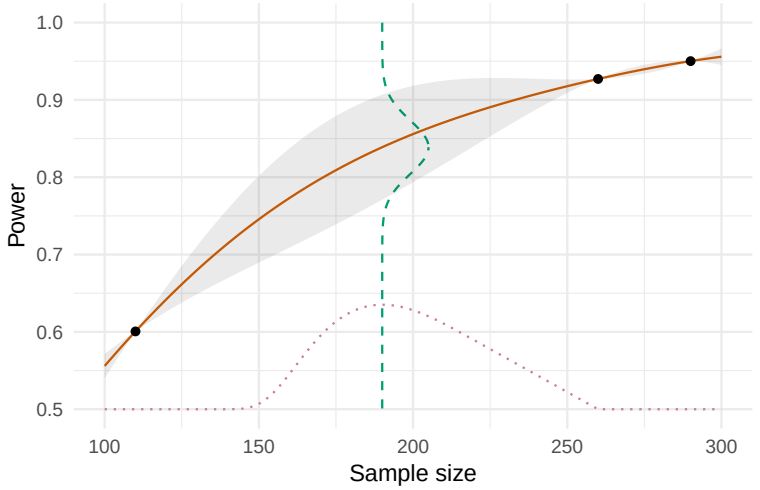

Figure 2: A Gaussian process model of a power function over a one-dimensional sample size (solid line) based on three evaluations. Uncertainty is shown as the shaded area. Expected improvement (dotted line) is maximised at a sample size of 190, where the predicted power is normally distributed around a mean estimate of 0.84 (dashed line).

By using covariance functions of this form we will obtain a Gaussian process which is twice differentiable. This would appear to be a reasonable restriction to place upon the functions of power and expected sample size we are interested in. Having specified the general form of the covariance function, in order to populate the covariance matrix we must choose values of the (hyper-)parameters $\boldsymbol{\theta} = (\sigma, \lambda_1, \ldots, \lambda_D)$. We do this by numerically optimising the log marginal likelihood

$$\log p(\mathbf{y} \mid \mathcal{X}_E, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top [K + \Delta]^{-1}\mathbf{y} - \frac{1}{2}\log|K + \Delta| - \frac{n}{2}\log 2\pi, \qquad (19)$$

considered as a function of $\boldsymbol{\theta}$ [24].

An illustration of a Gaussian process in one dimension is given in Figure 2. Here, 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 dashed line represents the predicted power at a specific point.

4.2.2 Expected improvement

At step (4) in algorithm 1 we consider all solutions $\mathbf{x} \in \mathcal{X}$ and attempt to find that which maximises expected improvement, a measure which we now define.

We first consider the case of expensive constraint functions but cheap and deterministic objective functions, as found in the cross-classified psychotherapy example. We combine multiple objective functions into a single function in the manner proposed in [18]. At each iteration of the algorithm the objective values $f_i(\mathbf{x})$ corresponding to objectives $i = 1, \ldots, B$ are first normalised with respect to their limits, leading to values in $[0, 1]$. We then calculate a weighted composite objective function

$$f_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{B} w_i f_i(\mathbf{x}). \tag{20}$$

The vector of weights $\mathbf{w} = (w_1, \ldots, w_B)$ is sampled from a flat Dirichlet distribution at each iteration and so the relative priorities of the search process fluctuate over time, helping to locate a range of solutions offering different trade-offs between the set of objectives. For notational convenience we will drop the subscript $\mathbf{w}$. Denoting the lowest composite objective value obtained by the solutions $\mathcal{X}_E$ as f_{min}, the improvement which would be obtained by evaluating the solution $\mathbf{x}_*$ is simply $f_{min} - f(\mathbf{x}_*)$.

Any anticipated improvement in the objective f at $\mathbf{x}_*$ must be balanced with the probability that the solution will be feasible. Given some MC estimates of an expensive constraint g, we can construct a GP model of the function as described in Section 4.2.1. The model 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 $p\%$ quantile of this distribution is below 0. We denote this quantile as

$$q(\mathbf{x}_*) = m + \Phi^{-1}(p)s, \tag{21}$$

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 [23] that its predictive distribution is $q_+(\mathbf{x}_*) \sim N(m_+, s_+^2)$ where

$$m_+ = m + \Phi^{-1}(p)\sqrt{\frac{\omega^2 s^2}{\omega^2 + s^2}} \tag{22}$$

$$s_+^2 = \frac{[s^2]^2}{\omega^2 + s^2}, \tag{23}$$

and ω is the MC error of the planned evaluation. This predictive distribution can be used to calculate the probability that the point $\mathbf{x}_*$ will be considered feasible following its evaluation. The improvement in the composite objective function is multiplied by this probability, thus penalising candidate solutions with a low chance of satisfying the constraint [27]. Specifically, we choose to evaluate the solution which maximises

$$EI = [f_{min} - f(\mathbf{x}_*)] \prod_{j=1}^{C} \Phi\left(\frac{-m_{j,+}}{s_{j,+}}\right), \tag{24}$$

where we now include all $j = 1, \ldots, C$ constraint functions.

To illustrate expected improvement we return to the one-dimensional problem illustrated in Figure 2. We aim to find a sample size which has power of at least 80%, with the lowest feasible sample size found so far being

If we aim to find a sample size which has at least 80% power, we can see that the mean function of the GP model crosses this threshold at a sample size of 171. However, the expected improvement

Returning to the one-dimensional example of Figure 2, the expected improvement is plotted as a dashed line. We see that although the GP model of the power function suggests that the expected power of a sample size of 171.

4.3 Implementation

4.3.1 Algorithm parameters

We are required to choose the initial set of points to evaluate before commencing with the EGO algorithm. Whilst a thorough evaluation of all options is beyond the scope of this paper, we can use recommended rules-of-thumb to guide our choice. It has been suggested that 10 points per dimension of the solution space should be included in the set $\mathcal{X}_E$, and that between 25 and 50% of the overall computation budget should be allocated to these initial evaluations []. Given a chosen number of points, a space-filling Latin hypercube sample (specifically, a maximin sample) of this size can be generated using the R package lhs [6]. During the iterative portion of the EGO algorithm, we must choose the number of MC samples N used when computing each estimate. This will be highly dependant on the computational burden of the underlying simulation program. We recommend using 1000 samples where possible. The choice of N should account for the fact that, in practice, fitting GP's to more than around 800 points is infeasible [7].

4.3.2 Regression diagnostics

The performance of the optimisation algorithm is dependant on the GP models providing a good representation of the underlying functions. It is therefore important that these models are regularly examined to try and identify any poor fits. One approach is to regularly plot the predicted mean function of the GP model in one or two dimensions, centred at the last evaluated point. In the psychotherapy example, we could for instance plot the GP power model as a function of k for fixed r and n_2, to ensure it is smooth and monotonic increasing (particularly in the area of interest around the nominal value) as we would expect. We can also contrast the predicted function values with the obtained function values at each iteration.

4.3.3 Programming and interface

We have chosen to use R to implement the proposed framework, principaly becasue of the availability of robust and efficient R packages for fitting Gaussain

process models (DiceKriging [26]) and for global optimisation (rgenoud [22]). 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]. Example R code is provided in the supplementary material. In order to use this code with a new problem, the user must follow a simple interface. Firstly, they must write their own simulation function which takes two arguments. The first argument describes the trial design, i.e. all variables which are to be optimised over. The second argument defines the parameter values for the population model, i.e the hypothesis under which we want to simulate. The function must return a binary value indicating whether or not the null hypothesis was rejected. In addition to this function, the user must define the hypotheses of interest, the nominal error rate constraints, the upper and lower limits of all design variables, and a function which returns a vector of objective function values for any given trial design. Given these components, the example code can be applied without modification to solve the trial design problem.

To summarise, the following objects must be specified:

- design space: data frame with fields 'name', 'low', 'up' specifying the name and limits of each design variable;
- parameters: data frame with fields corresponding to each parameter other than those specified in design space, with each row describing a hypothesis to be simulated under;
- sim trial(design, params): function which takes as arguments a design data frame (with field names corresponding to those of design space) and a params data frame (i.e. one row of parameters), which will simulate data generation and analysis and return a binary variable indicating the resulting stop/go decision;
- objectives(design): function taking a design data frame as an argument and returning a B dimension vector of objective function values.

5 Application to the examples

5.1 Cross-classified psychotherapy trial

5.1.1 Illustration

To illustrate the efficient optimisation method we apply it to the cross-classified psychotherapy trial of Section 3.1. We first translate the details regarding the population model, sampling model and planned analysis into a simulation program in R, which we have provided in the supplementary material. Given the three dimensions of the solution space, we choose a an initial Latin hypercube sample of 30 solutions and estimate the power of each of these using $N = 250$ MC samples. We then apply algorithm 1 using a remaining budget of 22,500 MC samples, using $N = 250$ of these in each iteration.

Figure 3: Objective values of solutions in the initial set $\mathcal{X}_E$ (red), subsequent iterations of the algorithm (blue), and those in the final approximation of the Pareto set (black).

r	n_2	k	$n_1 + n_2$	$\hat{\beta}$ (se)
0.5638016	486	40	760.0076	0.196 (0.02510649)
0.6495110	468	24	771.9712	0.128 (0.02112969)
0.6635676	473	15	786.8675	0.156 (0.02294899)
0.6695354	485	13	809.7247	0.192 (0.02491072)
0.6661190	513	12	854.7190	0.184 (0.02450665)
0.6644832	523	11	870.5247	0.156 (0.02294899)
0.6695527	524	9	874.8456	0.204 (0.02548600)
0.6761564	543	8	910.1529	0.160 (0.02318620)
0.6768232	549	7	920.5759	0.208 (0.02566990)
0.6909196	588	6	994.2607	0.196 (0.02510649)
0.8050653	864	5	1559.5764	0.168 (0.02364538)

Table 1: Solutions found by the EGO algorithm with a total budget of 30,000 MC samples.

The objective space of the problem is illustrated in Figure 3, which plots the number of therapists k against the total number of patients n. Plotted on Figure 3 are three sets of solutions. Firstly, the initial set of 30 points $\mathcal{X}_E$ are seen to be evenly spread across the space. The final set of solutions approximating the Pareto front and plotted in black. Finally, all other solutions evaluated at some point during the iteritive phase of the EGO algorithm are plotted in blue.

The final set of solutions are described further in Table 1 in terms of their design parameters and the MC estimate of their power. The standard error is estimated using the Normal approximation, i.e. $se = \sqrt{\hat{\beta}(1-\hat{\beta})/250}$. Note that in all but one cases the upper 99% confidence interval constructed with this standard error exceeds the nominal type II error rate of 0.2. These solutions are nevertheless considered feasible because the Gaussian process model is used to estimate the type II error rates, allowing information to be borrowed from neighbouring evaluations and greater precision to be obtained. For example, at the point $(r, n_2, k) = (0.6768232, 549, 7)$ the MC evaluation alone estimates type II error as 0.208 ± 0.05971712 (99% confidence interval) whereas the GP model estimates 0.1883946 ± 0.0109728 (99% credible interval). For comparison we computed a further MC estimate of the type II error of this design using $N = 10,000$ samples, which gave 0.1915 ± 0.00003601832 (99% confidence interval).

5.1.2 Comparison

We now compare the performance of the efficient optimisation method with the simpler heuristic algorithm described in Section 3.1. Given the stochastic nature of both algorithms, the comparison is repeated 1000 times. In each instance the heuristic algorithm is applied, using N MC samples for each solution evaluation. The total number of MC samples used over the course of the heuristic is recorded, as is the time taken. The EGO algorithm is then applied, terminating when either the number of MC samples or the time taken equal those used by the heuristic. The EGO algorithm always uses a 30-point Latin hypercube sample as its initial set of solutions $\mathcal{X}_E$. The number of MC samples used when evaluating these initial points is set to 25% of the total budget, or 50 samples per point, whichever is larger. This process is carried out for $N = 100, 250$ and 500.

We first consider the ability of each method to find at least some feasible solution for each value of k. Note that such a solution need not be explicit, in the sense that if a feasible solution for a lower value k_* has been found then we can assume that increasing the number of therapists to k whilst keeping all other design variables constant will also be feasible. Across all k and all 1000 runs, the proportions for which no feasible solutions were found are summarised in Table **??**. We see see that for each value of N considered, the EGO algorithm has considerably more success in finding feasible solutions than the heuristic. These results are detailed further in Figures **??**, **??** and **??**, which illustrate number of times no feasible solution was found for each value of k. As we would expect, the lower values of k are the less likely to be obtained.

5.2 Cluster-randomised pilot trial

Define the method parameters.

5.2.1 Illustration

We illustrate the Bayesian optimisation approach as applied to the cluster randomised pilot trial example of Section 3. We used a 60-point Latin hypercube sample generated using the maximinLHS function of the lhs package in R as the initial experimental design. MC estimates of power under each hypothesis were computed using $N = 250$ 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 4.

Only 12 of the initial 60 solutions are displayed in Figure 4, 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, given the upper limit of on the number of participants per cluster of 40. Reduction in 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. Given the scales involved,

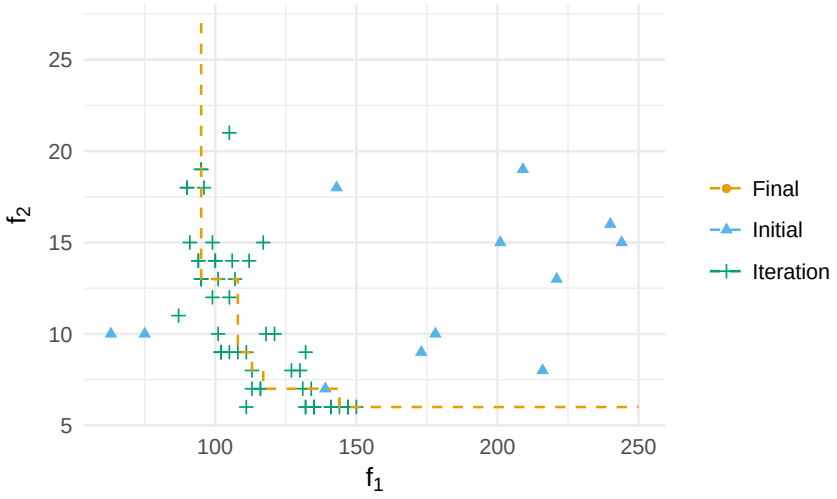

Figure 4: Going up to 60 iterations. We have focussed on the approximation front, omitting 48 points (all from the initial design) which take larger values of f_1 and/or f_2.

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	7	35	10	6	60	0.12	-0.01	0.10 (0.01)	0.29 (0.02)	0.32 (0.02)	95	13
14	3	42	11	6	66	0.12	-0.01	0.10 (0.01)	0.28 (0.02)	0.32 (0.02)	108	9
16	3	48	13	5	65	0.12	-0.02	0.07 (0.01)	0.29 (0.02)	0.35 (0.02)	113	8
19	3	57	15	4	60	0.11	-0.06	0.07 (0.01)	0.28 (0.02)	0.28 (0.02)	117	7
17	3	51	31	3	93	0.12	-0.05	0.09 (0.01)	0.28 (0.02)	0.35 (0.02)	144	6

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

an approximation set of size 5 would appear to provide sufficient flexibility for a trial design with the desired balance between objectives to be found.

As can be seen from Table 2, approximate upper 99% confidence limits for the power quantities formed from the original MC estimates often exceed the corresponding nominal bound. For example, the first solution given would have an approximate 98% two-sided confidence interval of $(0.079, 0.121)$. However, when the GP model is used to predict power instead of the raw MC estimate, the interval is $(0.082, 0.098)$. To verify this prediction we computed a more precise MC estimate using $N = 50,000$ samples, giving the interval $(0.091, 0.096)$.

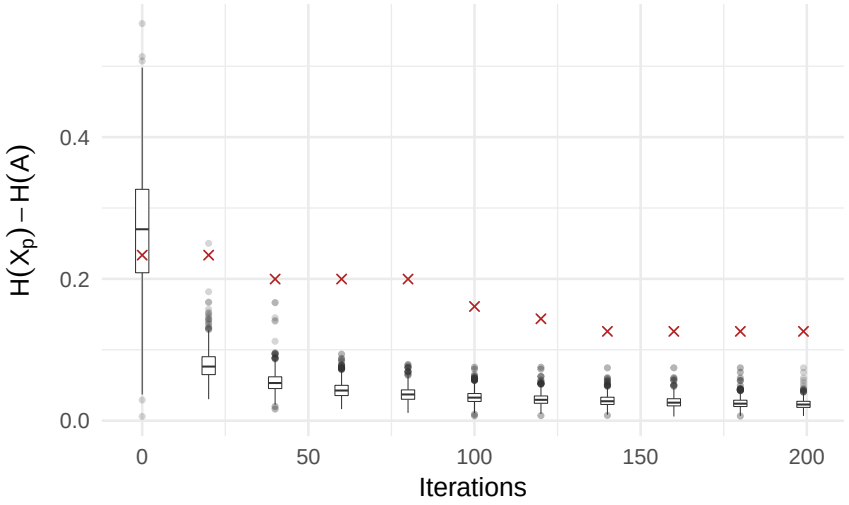

Figure 5: Boxplots illustrating the differences in dominated hypervolumes obtained as the number of iterations in the Bayesian optimisation algorithm increases. Corresponding differences obtained using the fixed experimental design are shown in red.

5.2.2 Comparison

To compare the two methods described in Section 4 we compare the dominated hypervolumes of the resulting approximation sets. In particular, we wish to calculate

$$\Delta = \mathcal{H}(\mathcal{X}_p) - \mathcal{H}(A), \tag{25}$$

the difference in dominated hypervolumes between the true Pareto set $\mathcal{X}_p$ and the attained approximation set A. As $\mathcal{X}_p$ is unknown, we instead use an approximation generated by repeatedly running the Bayesian optimisation algorithm to obtain M approximation sets $A_i, i = 1, \ldots, M$ and then finding the set of non-dominated solutions from within the union $\cup_{i=1}^{M} A_i$.

Each approximation set A_i is obtained through an initial experimental design of 60 points, with $N = 250$ samples used for each MC estimate. This is followed by 200 iterations of Bayesian optimisation using $N = 500$ for each MC estimate. After every 20 iterations the approximation set was found and the difference Δ calculated. The distributions of these differences as the number of iterations increases is plotted in Figure 5. For comparison are the results obtained using a fixed experimental design using a Sobol sequence, where the number of solutions included in the design is chosen to lead to a similar computational running time as required by the optimisation algorithm.

At the start of the iterative portion of the Bayesian optimisation algorithm, Figure 5 shows substantial variability in the quality of the approximation set obtained, with some high quality sets close to the the Pareto front occasionally

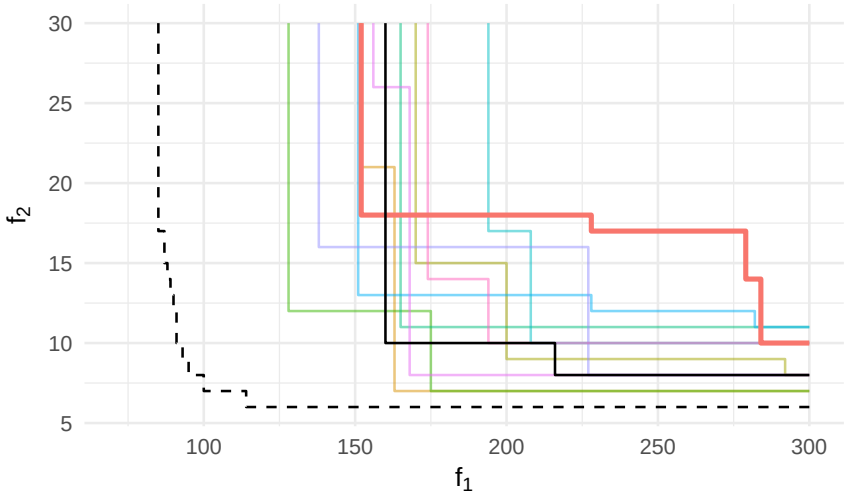

Figure 6: Random selection of approximation fronts obtained after 0 iterations of the Bayesian optimisation algorithm (coloured lines), together with the Pareto front (dashed black line) and the approximation front obtained using the fixed experimental design (solid black line).

found already at this point. A random selection of 10 approximation fronts with 0 iterations completed are illustrated in Figure 6.

The quality of solutions increases substantially as the number of iterations is increased, with the rate of improvement levelling off quite quickly. The updated approximation sets from Figure 6 are shown again in Figure 7, 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. As the number of iterations increases to 200, illustrated in Figure 8, the average computation time increases to over 9 hours but the improvement in solution quality is limited.

5.2.3 Results

- Evaluate the operating characteristics of the approximate solution using the simulation.
- ...

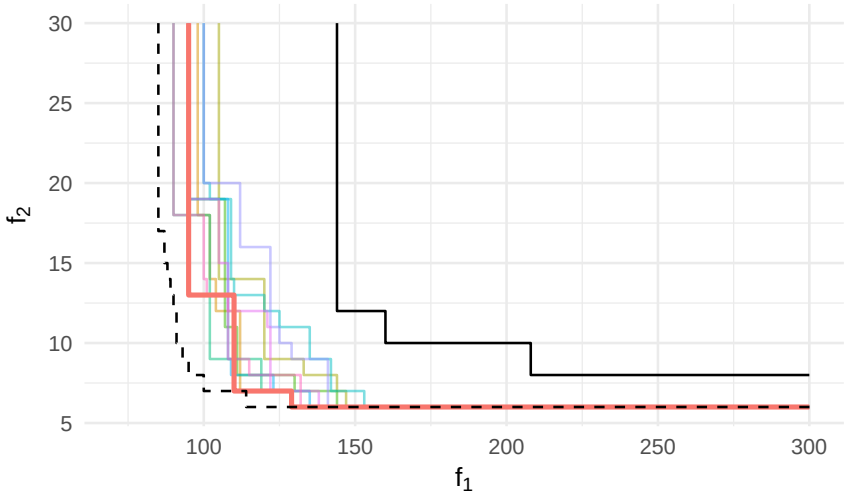

Figure 7: Time taken - 2 hrs 45 mins.

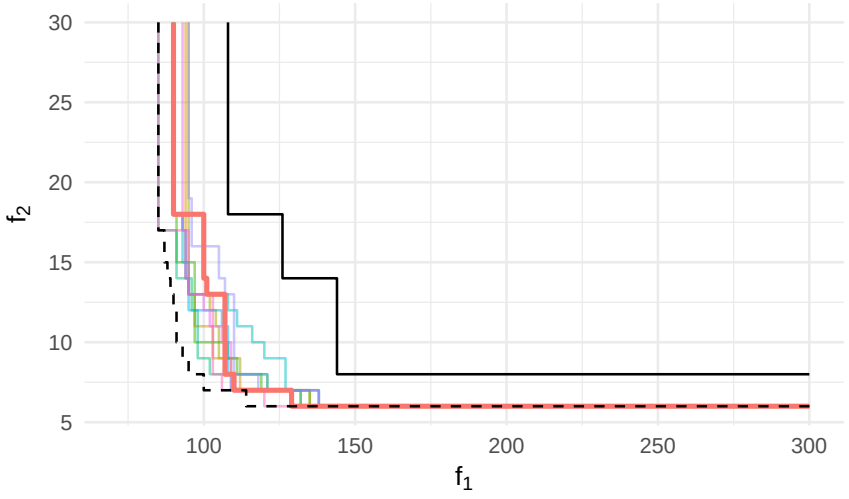

Figure 8: Time taken - 9 hrs 23 mins.

6 Discussion

- MLPowSim and SimSam both do simulation based SSD, the former using a very simple grid (although this does lead to a multi-criteria solution of sorts), the latter is limited to problems in one dimension. Note that SimSam could have been used as part of the heuristic algorithm in Example 1 to improve its efficiency.
- Many simulation SSD papers already do informal model-assisted design, by getting power at a set of n's and drawing a line through the results.
- Our approach is very general, can be applied to pretty much any problem as long as we can simulate it and the number of design variables / criteria aren't too large (say, ¡ 10).
- Minimisation of multiple criteria has been discussed for multilevel designs, but always through the *a priori* specification of a weighting function to combine the criteria. We would argue that it will generally be difficult to define such a weighting, that different parties may have different priorities, and that being able to view the exact trade-offs available would often lead to an adjustment of priorities.
- Multiple criteria has also been picked up to some extent in the multi-stage adaptive design field, where expected sample size and maximal sample size are often considered. General solution proposed has been to look for 'admissible' designs, essentially by solving the problem many times using different weightings in a weighting functions, so very inefficient. Advocated because they can often get a huge reduction in one ciretia at only a small increase in another, so reinforces the point above.
- The methods here can be applied fairly easily in R for any proposed simulation, and we can provide the necessary guide to defining the structure and interface of the simulation, and the necessary functions for doing the computations, in an appendix.
- Just because no analytic solution is available, and simulation must be used, we do not necessarily need the methods here. The MAMS design uses simulation, but the original model is sufficiently simple as to allow very fast simulations. Similarly, in some cases exact solutions are available in the sense that a numerical solver could do the integrations necessary, but the time needed to do so would necessitate the use of the proposed algorithm (although in this case we would have deterministic output).
- Reproducibility. Don't need to be able to repeat someone's SSD, but do want to be able to confirm the design they propose has the operating characteristics they claim. Will this mean people will have to make their simulation code freely available? If so, would want our software to be easy to use for this purpose. May connect with writing a simulation protocol as described in [31].

- Could have modelled the penalised objective functions directly, but suspect this may have made the problem much harder to solve as we would be modelling a strange objective function which is very smooth but has a huge, almost discontinuous jump, once a constraint is violated.

- Modelling all the functions separately assumes independence, when in fact we know there will be correlation. This could potentially be addressed by modelling everything together ('co-krigging'), but this is apparently difficult to do well [RFE], with limited benefits to be gained [REF].

- Code profiling and efficiency - how much time are we spending getting Monte Carlo estimates, how much time fitting GP models, and how much time optimising the expected improvement criteria? Can any of these be improved? Look at both implementation issues (i.e. writing fast code implementing the proposed algorithm, contrast using R with C / C++, issues around flexibility and transparency [31]) and methodological issues (i.e. using better MC algorithms e.g. importance sampling).

- SSD really involves trading off error rates with sample size, and should be an iterative process. The proposed methods can be used in this manner by simply changing the nominal rates as the search progresses and we get a better picture of the trade-offs involved. We can still make use of all the evaluations made, as opposed to alternative methods which would just start everything again.

- The proposed optimisation algorithm has a number of parameters which have to be chosen. We have made some suggestions, but for every new problem there will be different optimal choices. Further work and experience in applying the method to a wider range of problems may help in identifying good robust choices for these parameters.

- We have used GPs, but could consider other non-parametric regression models, e.g. splines, GAMs, etc. Key advantage of GPs is that they provide uncertainty in predictions, so good for sequential optimisation. Key limitation is the scaling with number of points, which in R makes models of more than 200 points quite slow.

- Could also consider parametric regression models, arguing that the power functions and expected sample size functions may have a form which we could guess at in advance. Although, can always add a model component to the GP and still get the flexibility when modelling the errors.

- We have focussed, through simulation and non-parametric modelling, on a flexible 'black-box' type approach. There may be designs which are very common, yet complex, in which case we may be able to find better solutions tailored specifically to them.

- When optimising the improvement criteria, we repeatedly get predictions of the GP models for just one point. We can get predicted values for

many points at once with very little extra effort, which suggests a different algorithm which evaluates several points at once may be more efficient.

- Extension to composite hypothesis testing, e.g. when we have nuisance parameters e.g. feasibility parameters around recruitment or correlation parameters for multiple endpoints, which we need to maximise over.
- Extension to Bayesian SSD, where the simulation will (?) be even more computationally demanding than here.
- We have worked in R. To help Stata and SAS users make use of this approach, we could try to link the software so that they can write the simulation and analysis in their preferred language and then use an R based routine (GUI?) which calls those simulations.
- We have only considered frequentist power calculations, but the approach can be applied equally for some Bayesian power equivalent. We have already cited [36], also look at "SAMPLE SIZE FOR BINOMIAL REGRESSION WITH MISCLASSIFICATION".
- We have considered a problem with three error rate constraints, but the method will extend naturally to designs with more e.g. multiple outcome designs with two types of power.
- For common design problems, rather than everyone optimising to their own local nuisance parameter specification, we could consider building a larger model and holding it centrally for others to use. Analogous to printing sample size tables - we have access to resources here, and could use these to pre-compute and help others without those resources. We would then be interested in a model that gives good predictions, potentially form a very large set of noisy data. So some of the non-parametric machine learning methods mentioned above might be more appropriate.
- We consider the best solution to be one which has been evaluated, which is a common recommendation. But in the MC setting, it may be that the GP says another point is much better, and we could have it with only a few MC samples. This suggests we could improve through better dynamic allocation of the resources.
- We have considered deterministic and cheap objective functions only. To tackle trial design problems with stochastic objectives which require simulation to evaluate, e.g. adaptive designs, we need to allow expensive stochastic objectives also.

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), jun 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] 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.

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

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

[7] 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.

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

[9] 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.

[10] A. H. Feiveson. Power by simulation. 2:107–124, 2002.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

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

[18] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, feb 2006.

[19] 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.

[20] 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.

[21] 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.

[22] Walter R. Mebane, Jr. and Jasjeet S. Sekhon. Genetic optimization using derivatives: The rgenoud package for R. *Journal of Statistical Software*, 42(11):1–26, 6 2011.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Michael W Sill, Larry Rubinstein, Samuel Litwin, and Greg Yothers. A method for utilizing co-primary efficacy outcome measures to screen regimens for activity in two-stage phase II clinical trials. *Clinical Trials*, 9(4):385–395, 2012.

[30] Richard Simon. Optimal two-stage designs for phase II clinical trials. *Controlled Clinical Trials*, 10(1):1 – 10, 1989.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Fei Wang and Alan E. Gelfand. A simulation-based approach to bayesian sample size determination for performance under a given model and for separating models. *Statistical Science*, 17(2):pp. 193–208, 2002.

[37] James M.S. Wason, Adrian P Mander, and Simon G. Thompson. Optimal multistage designs for randomised clinical trials with continuous outcomes. *Statistics in Medicine*, 31(4):301–312, dec 2011.