Statistics and Computing (2020) 30:1183–1208
https://doi.org/10.1007/s11222-020-09938-6

A Laplace-based algorithm for Bayesian adaptive design

S. G. J. Senarathne[1] · C. C. Drovandi[1] · J. M. McGree[1]

Received: 25 July 2019 / Accepted: 3 March 2020 / Published online: 5 April 2020

Abstract

This article presents a novel Laplace-based algorithm that can be used to find Bayesian adaptive designs under model and parameter uncertainty. Our algorithm uses Laplace importance sampling to provide a computationally efficient approach to undertake adaptive design and inference when compared to standard approaches such as those based on the sequential Monte Carlo (SMC) algorithm. Like the SMC approach, our new algorithm requires very little problem-specific tuning and provides an efficient estimate of utility functions for parameter estimation and/or model choice. Further, within our algorithm, we adopt methods from Pareto smoothing to improve the robustness of the algorithm in forming particle approximations to posterior distributions. To evaluate our new adaptive design algorithm, three motivating examples from the literature are considered including examples where binary, multiple response and count data are observed under considerable model and parameter uncertainty. We benchmark the performance of our new algorithm against: (1) the standard SMC algorithm and (2) a standard implementation of the Laplace approximation in adaptive design. We assess the performance of each algorithm through comparing computational efficiency and design selection. The results show that our new algorithm is computationally efficient and selects designs that can perform as well as or better than the other two approaches. As such, we propose our Laplace-based algorithm as an efficient approach for designing adaptive experiments.

Keywords Importance sampling · Model discrimination · Parameter estimation · Pareto smoothing · Sequential Monte Carlo · Total entropy

1 Introduction

Bayesian adaptive design is an integral component of scientific investigation in fields such as the physical, chemical and biological sciences (Roy and Notz 2014; Barz et al. 2016; Antognini and Giovagnoli 2015). The approach is to, based on current prior information about the experimental outcomes, find a design to collect the next data point or batch of data. Once this design has been selected, data are collected, and the prior information for the experiment is updated to reflect the new information gained from the new data. This process then iterates a fixed number of times or until a certain stopping criterion has been met. At each iteration, the choice of design is determined by maximising an expected utility function over the space of all possible designs, where the expectation is with respect to the joint distribution of the candidate models, the parameters and the experimental outcomes. The expected utility is defined to encapsulate the experimental aim/s, typically evaluating the expected amount of information to be gained from running a given design. This component of the scientific method can be thought of as adaptive learning and thus fits naturally within a Bayesian framework where the posterior distribution for a given iteration becomes the prior information for the next iteration. Thus, throughout this paper, we will consider design and inference within a Bayesian framework.

Wakefield (1994) proposed an adaptive design algorithm for determining an optimal dosing regimen in a pharmacokinetic study. In their approach, a Gibbs sampler was used to update the posterior distribution at each iteration of the adaptive design process. Palmer and Müller (1998) generalised the approach of Wakefield (1994) to allow more complex models to be considered. This was achieved through a Markov chain Monte Carlo (MCMC) algorithm for sampling from the posterior distribution at each iteration of the adaptive design and for approximating the expected utility. We note that both of the above approaches are rather computationally intensive as a Gibbs or MCMC sampler needs to be run at

✉ S. G. J. Senarathne
jagath.gedara@hdr.qut.edu.au; jagathuop@gmail.com

[1] School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Brisbane, Australia

every iteration of the adaptive design process, and many times within the selection of an optimal design. Thus, recently, a few authors have considered a combination of MCMC samplers and importance sampling for Bayesian adaptive design (Stroud et al. 2001; Weir et al. 2007; McGree et al. 2012). In these algorithms, MCMC was used to update the posterior distribution at each iteration of the design process, and importance sampling was then used to approximate the posterior distribution when estimating the expected utility of the design. As these design algorithms require the use of MCMC at each iteration, they are computationally quite wasteful as all previous data are continually being considered within an MCMC algorithm. A more efficient approach would be to avoid such a step completely or at least avoid having to run it within every iteration of the adaptive design process.

To avoid much of the computation in the above algorithms, Lewi et al. (2009) developed an adaptive design algorithm for neurophysiology experiments based on the Laplace approximation. At each iteration of the adaptive design algorithm, the posterior distribution is approximated by a Laplace approximation, that is, a multivariate normal distribution. For further computational efficiency, the Laplace approximation was based on an approximation to the posterior distribution formed by considering the posterior distribution from the previous iteration. For approximating the expected utility, model predictions were used to approximate the mutual information between the parameters and the data. Thus, the methods of Lewi et al. (2009) could be considered as an approximate version of a standard sequential Laplace algorithm (SLP). In this paper, we will evaluate the performance of such an algorithm in designing various adaptive experiments. As such, further details about SLP will be provided in Sect. 2. Such methods have also been considered for efficiently approximating the posterior distribution when locating Bayesian static designs (Ryan 2003; Long et al. 2013).

The current standard approach to Bayesian adaptive design is based on the sequential Monte Carlo (SMC) algorithm, see Del Moral et al. (2006). In SMC, posterior distributions are approximated by repeatedly applying a reweight, resample and move step. Some of the major benefits of SMC include being more efficient than MCMC in terms of capturing multi-modal distributions, and the algorithm provides an efficient estimate of the model evidence which can be used for model selection (Del Moral et al. 2006). However, one drawback of this algorithm is the move step, where an MCMC sampler is run to diversify the particle set. Such a sampler can be computationally expensive as all previous data need to be considered (further details to come later in Sect. 2.2). Nonetheless, Drovandi et al. (2013) proposed the use of the SMC algorithm in Bayesian adaptive design to approximate the posterior distribution at each iteration of the design process and used importance sampling to approximate expected utility functions. This algorithm was applied to examples where generalised linear and nonlinear models were used to describe discrete data. Further developments of this approach came from Drovandi et al. (2014) who extended the SMC algorithm for Bayesian design to consider model uncertainty. The goal of the experiments was to discriminate between candidate models, so a discrimination utility based on the mutual information between the model and the data was considered. As in the original algorithm, importance sampling was used to estimate the expectation of this utility.

In this paper, we propose a novel adaptive design algorithm based on the Laplace approximation to efficiently derive designs for sequential experiments. This algorithm was developed by adopting the importance sampling (reweight) step of SMC but replacing the resample and move steps with Laplace importance sampling (Kuk 1999). Further, within this importance sampling step, Pareto smoothing (Vehtari et al. 2017) is used to obtain stable importance weights, and hence to improve the efficiency of posterior estimates based on this importance sampling step. As will be seen, this yields considerable computational efficiency with little to no compromise in the selection of efficient designs. Both of these benefits were observed when we compared the performance of our new algorithm (in terms of computational efficiency and design selection) with the standard SMC and SLP algorithms. Of note, the motivating examples used for this comparison were rather complex in that significant model and parameter uncertainty were present a priori. As such, designs were selected for the dual experimental goals of parameter estimation and model discrimination using the total entropy utility (Borth 1975; McGree 2017). To reduce the computational complexity of the sequential design problem, a greedy/myopic design approach was considered (Dror and Steinberg 2008; Cavagnaro et al. 2010). The results from applying our new algorithm to these motivating examples suggest there are considerable benefits in adopting our new algorithm over standard approaches to Bayesian adaptive design.

The paper proceeds as follows. In Sect. 2, an adaptive experimental design framework is formally defined, and background information about the standard SLP and SMC algorithms is provided. Our novel adaptive design algorithm is then introduced in Sect. 3. In Sect. 4, three motivating examples are considered to demonstrate the performance of our new adaptive design algorithm. The paper concludes with Sect. 5 which provides a discussion of key findings and suggestions for future research.

2 Background

Suppose an adaptive experiment is to be run to collect observations y_i across n iterations, for $i = 1, \ldots, n$. For the purposes of this paper, we will assume that only a single data point is observed within each iteration of the adaptive design resulting in the following construction of the likelihood:

$$p(\boldsymbol{y}_{1:n}|\boldsymbol{d}_{1:n}, \boldsymbol{\theta}_m, M = m) = \prod_{i=1}^{n} p(y_i|d_i, \boldsymbol{\theta}_m, M = m),$$

where M is the random variable associated with a set of K candidate models and $\boldsymbol{\theta}_m$ represents the parameters in model m. These candidate models could (but not necessarily) be nested models, and the dimension of $\boldsymbol{\theta}_m$ depends on m, for $m = 1, \ldots, K$. Here, $d_i = (d_{1i}, \ldots, d_{pi}) \in \mathcal{D}$ denotes the design used in the ith iteration in the design space $\mathcal{D} \subset \mathbb{R}^p$, for $i = 1, \ldots, n$.

In adaptive experiments, the Bayesian inference problem is to approximate or sample from a sequence of posterior distributions built up through data annealing. That is, for the ith iteration of an adaptive design, the posterior distribution is defined as:

$$p(\boldsymbol{\theta}_m|M = m, \boldsymbol{d}_{1:i}, \boldsymbol{y}_{1:i}) = \frac{p(\boldsymbol{\theta}_m|M = m)p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, \boldsymbol{\theta}_m, M = m)}{Z_{m,i}},$$

where $Z_{m,i} = p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M = m) = \int_{\boldsymbol{\Theta}_m} p(\boldsymbol{\theta}_m|M = m)p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, \boldsymbol{\theta}_m, M = m)\mathrm{d}\boldsymbol{\theta}_m$ denotes the normalising constant or the model evidence which can be used for model choice via the posterior model probability as follows:

$$p(M{=}m|\boldsymbol{d}_{1:i}, \boldsymbol{y}_{1:i}){=}\frac{p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M = m)p(M = m)}{\sum_{k=1}^{K} p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M = k)p(M = k)}, \tag{1}$$

where there is a preference for the model with the largest posterior model probability $p(M = m|\boldsymbol{d}_{1:i}, \boldsymbol{y}_{1:i})$, for $m = 1, \ldots, K$.

The Bayesian adaptive design problem can then be stated as selecting d_i at each iteration based on a proposed utility function. Such a function is defined to encapsulate the aim/s of the experiment which could include parameter estimation, model discrimination and/or prediction. For the ith iteration of an adaptive design, denote the utility function as $U(d, z, \boldsymbol{\theta_m}, m|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})$, where z is a supposed outcome obtained from running design d and belongs to the same space as the measurement y. As z, $\boldsymbol{\theta}_m$ and m are unknown, the expectation is taken with respect to the joint distribution of these random variables based on the (current) prior information. This yields the following expected utility:

$$\begin{aligned} &U(d|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}) \\ &\quad = E_{z,\boldsymbol{\theta}_m,m|\boldsymbol{d}_{1:i-1},\boldsymbol{y}_{1:i-1}}[U(d, z, \boldsymbol{\theta}_m, m|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})] \\ &\quad = \sum_{m=1}^{K} p(M = m|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}) \\ &\qquad \times \sum_{z\in\mathcal{S}} p(z|d, \boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}, M = m) \\ &\qquad \times \int_{\boldsymbol{\Theta}_m} U(d, z, \boldsymbol{\theta}_m, m|\boldsymbol{d}_{1:i-1}, y_{1:i-1}) \\ &\qquad\quad \times p(\boldsymbol{\theta}_m|M = m, \boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})\mathrm{d}\boldsymbol{\theta}_m, \end{aligned} \tag{2}$$

where $p(z|d, \boldsymbol{d}_{1:i-1,\boldsymbol{y}_{1:i-1}}, M = m) = \int_{\boldsymbol{\Theta}_m} p(z|d, \boldsymbol{\theta}_m, M = m) \times p(\boldsymbol{\theta}_m|M = m, \boldsymbol{d}_{1:i-1,\boldsymbol{y}_{1:i-1}})\mathrm{d}\boldsymbol{\theta}_m$ and S represents the sample space of a discrete response. The above expected utility $U(d|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})$ is defined for discrete data as such data are observed in all of the motivating examples considered in this paper. Extension to other data types, such as continuous, is straightforward and therefore omitted.

Thus, at each iteration of an adaptive design, one seeks to find $d^* = \arg\ \max_{d\in\mathcal{D}} U(d|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})$, which becomes d_i, the design selected at the ith iteration of the algorithm. Unfortunately, the above expression for the expected utility generally cannot be solved analytically and thus needs to be approximated. The most common approach for this is Monte Carlo (MC) integration through the simulation of prior predictive data as follows:

$$\begin{aligned} &U(d|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}) \\ &\quad \approx \sum_{m=1}^{K} p(M = m|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}) \\ &\qquad \times \sum_{z\in\mathcal{S}} p(z|d, \boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}, M = m) \\ &\qquad\quad \times \frac{1}{B}\sum_{b=1}^{B} U(d, z, \boldsymbol{\theta}_{m,b}, m|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1}), \end{aligned} \tag{3}$$

where $\boldsymbol{\theta}_{m,b} \sim p(\boldsymbol{\theta}_m|M = m, \boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})$ and B represents the number of samples considered for MC integration.

The adaptive design process described above is outlined in Algorithm 1 where initially the prior information about the models and parameters is defined along with the design space. The process then, in our case, iterates a fixed number of times where the ith optimal design is found by maximising an expected utility (line 3), the ith data point is collected (line 4) and prior information (about the models and the parameters) is updated based on the information gained from the ith data point (lines 5 and 6). For the motivating examples considered in this desktop study, data cannot actually be collected (line 4). In place of this, we assume data are generated from an underlying model with specified parameter values. Additionally, within our description of the adaptive design process, we include computation of the model evidence (line 6) as we will consider model uncertainty throughout this paper. We also note that such a quantity is needed to evaluate the expected utility in Eq. (3) (via the posterior model probability).

Algorithm 1 Adaptive design algorithm

```
1: Initialise the prior information $p(\boldsymbol{\theta}_m|M = m)$, $p(M = m)$, for $m = 1, 2, \ldots, K$
2: for $i = 1$ to $n$ do
3:     Find the $i$th optimal design point $d_i$, by maximising the utility $U(d|\boldsymbol{d}_{1:i-1}, \boldsymbol{y}_{1:i-1})$
4:     Collect data point $y_i$ at design point $d_i$
5:     Update the posterior distribution $p(\boldsymbol{\theta}_m|M = m, \boldsymbol{d}_{1:i}, \boldsymbol{y}_{1:i})$
6:     Compute model evidence $p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M = m)$ for $m = 1, 2, \ldots, K$
7: end for
```

In considering the adaptive design process as outlined in Algorithm 1, there are two main difficulties (at least computationally). These are: (1) efficiently updating prior information as new data arrive (lines 5 and 6); (2) efficiently approximating the expected utility function (line 3). In this paper, we propose a new algorithm that addresses both of these difficulties. Before describing this approach, we outline two previously proposed algorithms based on standard implementations of: (1) the Laplace approximation and (2) the SMC algorithm. We provide details on both of these algorithms, not only because we benchmark the performance our new algorithm against them, but also so the reader can understand the differences between the three approaches.

2.1 Adaptive design based on the Laplace approximation

In this section, we describe a standard Laplace-based approach to Bayesian adaptive design where the posterior distribution of the parameters at each iteration is approximated by a Laplace approximation, and the expectation of the utility function is approximated via importance sampling. Pseudo-code for this approach is provided in Algorithm 2.

To initialise the algorithm, for each candidate model, a set of particles is drawn from the prior distribution of the parameters, and each particle is given equal weight (line 1). Within each iteration of the design process, importance sampling is used to efficiently approximate expected utilities. That is, assume that we want to evaluate the expectation of $h(X)$ where $h(.)$ is some function and X is a random variable with a probability distribution $p(x)$. The expectation can be defined as follows:

$$E[h(X)] = \int_X h(x)p(x)\mathrm{d}x.$$

In some cases, this integral may not be solvable analytically, but can be approximated using MC integration as follows:

$$E[h(X)] \approx \sum_{j=1}^{N} h(x_j)p(x_j) = \sum_{j=1}^{N} h(x_j)w^j,$$

where $x_j \sim p(x)$ and $w^j = 1/N$.

From the above, it can be seen that sampling from $p(x)$ is required to approximate the expectation. In some cases, direct sampling from this distribution may be difficult. Accordingly, importance sampling can be used through the following formulation:

$$E[h(X)] = \int_X \frac{h(x)p(x)}{q(x)} q(x)\mathrm{d}x, \tag{4}$$

where $q(x)$ is an importance distribution that is straightforward to sample from directly and provides sufficient coverage of $p(x)$.

The approximation to the expectation is then given as follows:

$$E[h(X)] \approx \sum_{j=1}^{N} \frac{h(x_j)p(x_j)}{q(x_j)} = \sum_{j=1}^{N} h(x_j)w^j,$$

where $x_j \sim q(x)$ and $w^j = \frac{p(x_j)}{q(x_j)}$ are the importance weights. Then, the weighted sample $\{x_j, w^j\}_{j=1}^N$ is a particle approximation to the target distribution $p(x)$.

For the purpose of evaluating the expectation of a utility function, $p(x)$ will be the posterior distribution of $\boldsymbol{\theta}_m$. Accordingly, evaluating $p(x)$ will yield density values that are only proportional to the posterior distribution. Again, importance sampling can still be used in such cases (i.e. when $p(x)$ is an unnormalised density). Here, the approximation becomes:

$$E[h(X)] \approx \sum_{j=1}^{N} h(x_j)\left(\frac{w^j}{\sum_{k=1}^N w^k}\right) = \sum_{j=1}^{N} h(x_j)W^j, \tag{5}$$

where $w^j = \frac{p(x_j)}{q(x_j)}$ and $W^j = \frac{w^j}{\sum_{k=1}^N w^k}$ are the normalised importance weights. Thus, we have obtained a weighted sample from the target distribution which can be denoted as $\{x_j, W^j\}_{j=1}^N$.

For approximating the expectation of utilities, the utility function $U(d, z, \boldsymbol{\theta}_m, m|\boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1})$ is evaluated based on the posterior distribution of $\boldsymbol{\theta_m}$. This requires updating the posterior distribution based on a supposed outcome z, at a design point d. For this purpose, importance sampling is used throughout the SLP algorithm with importance distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1}, M = m)$ (i.e. the posterior distribution at the $(i-1)$th iteration). Thus, the evaluation of the utility function will be based on a weighted particle set, see Drovandi et al. (2013) for further details.

Once the ith design has been selected and the data collected, a Laplace approximation is formed for each candidate model. This is achieved by forming a multivariate normal approximation to the posterior distribution (of each model) by first finding the posterior mode as follows:

$$\boldsymbol{\theta}^*_m = \arg\max_{\boldsymbol{\theta}_m} \Big\{ \log p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, \boldsymbol{\theta}_m, M=m) + \log p(\boldsymbol{\theta}_m|M=m)\Big\}. \tag{6}$$

This mode becomes the mean of the multivariate normal approximation. The variance–covariance matrix is formed via the inverse of the negative Hessian matrix at this mode. This Hessian matrix can be defined as follows:

$$\boldsymbol{H}(\boldsymbol{\theta}^*_m) = \frac{\partial^2\{f(\boldsymbol{\theta}_m)\}}{\partial\boldsymbol{\theta}_m \partial\boldsymbol{\theta}_m{}'}\Big|_{\boldsymbol{\theta}_m=\boldsymbol{\theta}^*_m}, \tag{7}$$

where $f(\boldsymbol{\theta}_m) = \log p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, \boldsymbol{\theta}_m, M=m) + \log p(\boldsymbol{\theta}_m|M = m)$.

The approximation to the posterior distribution can then be defined as follows:

$$p(\boldsymbol{\theta}_m|M=m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}) \approx MVN(\boldsymbol{\theta}^*_m, \boldsymbol{\Sigma}(\boldsymbol{\theta}^*_m)), \tag{8}$$

where $\boldsymbol{\Sigma}(\boldsymbol{\theta}^*_m) = [-\boldsymbol{H}(\boldsymbol{\theta}^*_m)]^{-1}$. The Laplace approximation to the posterior distribution is computationally efficient when compared to alternatives such as MCMC and SMC (Lewi et al. 2009). When the posterior concentrates around a single mode or sub-manifold, the Laplace approximation can be highly efficient for estimating utility functions in Bayesian design (Long et al. 2015). However, this approximation may introduce error/bias when the posterior distribution is skewed or multi-modal. To reduce such error/bias, other authors have proposed including high-order derivatives in the approximation (Shun and McCullagh 1995; Clark and Dixon 2017; Ogden 2018). However, this comes with additional computational costs.

As shown in Bernardo and Smith (2000), the model evidence for model m can be approximated based on a given Laplace approximation and has the following form:

$$\begin{aligned} &p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M=m) \\ &\quad= (2\pi)^{(\frac{q_m}{2})}|\boldsymbol{\Sigma}(\boldsymbol{\theta}^*_m)|p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, \boldsymbol{\theta}^*_m, M=m) \\ &\quad\times p(\boldsymbol{\theta}^*_m|M=m), \end{aligned} \tag{9}$$

where q_m is the number of parameters in model m. As shown in Eq. (1), these model evidences can be normalised to approximate posterior model probabilities. Such approximations have been considered previously in Bayesian static design, see, for example, Overstall et al. (2018a).

Algorithm 2 Sequential Laplace algorithm

```
1: Draw $\boldsymbol{\theta}^j_{m,0}$ from prior $p(\boldsymbol{\theta}_m|M = m)$ and set $W^j_{m,0} = 1/N$ for $j = 1, 2, \ldots, N$ and for $m = 1, 2, \ldots, K$
2: for $i = 1$ to $n$ do
3:     Find the $i$th optimal design point $d_i$, by maximising the utility $U(d|\boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1})$
4:     Collect data $y_i$ at design point $d_i$
5:     for $m = 1$ to $K$ do
6:         Find the posterior mode $\boldsymbol{\theta}^*_m$ and Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}^*_m)$ at this mode using Eqs. (6) and (7)
7:         Approximate $p(\boldsymbol{\theta}_m|M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$ via the Laplace approximation (Eq. 8)
8:         Approximate the model evidence, $p(\boldsymbol{y}_{1:i}|\boldsymbol{d}_{1:i}, M = m)$ (Eq. 9)
9:         Sample $\boldsymbol{\theta}^j_{m,i} \sim MVN(\boldsymbol{\theta}^*_m, \boldsymbol{\Sigma}(\boldsymbol{\theta}^*_m))$ and set $W^j_{m,i} = 1/N$, for $j = 1, 2, \ldots, N$
10:    end for
11: end for
```

Algorithm 2 provides pseudo-code for the SLP approach to adaptive design. Initially, a sample of N particles is drawn from the prior for each model m (line 1). Then, for each iteration, the next optimal design is found by maximising a utility function based on the current information about each model and the parameter values (line 3). Given this design, data are simulated from a supposed model with assumed parameter values (line 4). For each model, the Laplace approximation is used to approximate the posterior distribution of the parameters and to update the model evidence (lines 6–7). A sample of N particles is then drawn from each of these posterior distributions such that they can be used to approximate the utility for finding the next design point (line 8). This process continues until a fixed number of data points have been observed.

2.2 Sequential Monte Carlo

SMC methods are a collection of techniques that approximate a sequence of distributions known up to a normalising constant. The approach combines importance sampling (reweighting), resampling and MCMC techniques (move step) to approximate a sequence of target distributions. When there exists uncertainty about the model, SMC also provides an estimate of the marginal likelihood which can be used for model choice.

In SMC, particles are propagated through a sequence of target distributions rather than re-generating a whole new set of particles within each iteration of the design process as in the SLP algorithm. Consequently, as more data are observed, the particle weights become more variable and skewed; hence, the effective sample size (ESS$_m$) will decrease and is therefore monitored throughout the algorithm. In SMC, ESS$_m$ can be estimated as follows (Doucet et al. 2000):

$$\text{ESS}_m \approx \frac{1}{\sum_{j=1}^{N}(W^j_{m,i})^2},$$

where $W^j_{m,i}$ for $j = 1, 2, \ldots, N$ are the normalised importance weights of the particle set for model m at the ith iteration.

When the ESS_m drops below a defined threshold, particles are resampled and moved. The resampling step is used to increase the ESS_m back up to (approximately) the initial number of samples N. This is achieved by sampling the particles with replacement with probability equal to the particle weights. Such resampling will most likely select particles with high weight and remove particles with low weight thus yielding duplicate particles. To diversify each particle set, the move step is used where an MCMC kernel with invariant distribution is used to propose new particles. Here, a random walk Metropolis–Hastings algorithm is used, where the variance–covariance of the current particle set is used to form efficient proposals. That is, given the particle set $\{\boldsymbol{\theta}^j_{m,i}, W^j_{m,i}\}^N_{j=1}$, the proposal distribution $\eta_{m,i}(\cdot|\cdot)$ is multivariate normal with the mean being the value of a given particle and variance–covariance based on the current particle set. Upon completion of this step, each particle set should contain (approximately) independent draws from the required posterior distribution which are then used for design selection in the next iteration of the adaptive design.

To approximate the model evidence within this SMC framework, we note that the ratio of normalising constants $Z_{m,i}/Z_{m,i-1}$ is equivalent to the predictive distribution of y_i given the current data y_{i-1} (Del Moral et al. 2006). Thus, in the SMC framework, this ratio can be approximated as follows:

$$\begin{aligned} Z_{m,i}/Z_{m,i-1} &= \int\limits_{\boldsymbol{\Theta}_m} p(y_i|d_i, \boldsymbol{\theta}_m, M = m) \\ &\quad \times p(\boldsymbol{\theta}_m|M = m, \boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1})\mathrm{d}\boldsymbol{\theta}_m \qquad (10) \\ &\approx \sum_{j=1}^{N} w^j_{m,i}, \end{aligned}$$

where $w^j_{m,i}$ for $j = 1, 2, \ldots, N$ are the unnormalised importance weights of the particle set for model m at the ith iteration.

Given that $\log Z_{m,i} = \sum_{t=0}^{i-1} \log\left(Z_{m,i-t}/Z_{m,i-1-t}\right)$ and $Z_{m,0} = 1$, the log model evidence of model m at the ith iteration can be approximated as follows:

$$\log Z_{m,i} = \log Z_{m,i-1} + \log\left(Z_{m,i}/Z_{m,i-1}\right). \qquad (11)$$

After approximating the model evidence of each model, they can be normalised to estimate the posterior model probability of the mth model for $m = 1, 2, \ldots, K$ as given in Eq. (1).

A major disadvantage of the SMC algorithm is the move step where it is required to traverse the particle set through an MCMC kernel many times (R_m). As proposed by Drovandi and Pettitt (2011), R_m should satisfy the condition:

$$R_m = \frac{\log c}{\log(1-p)},$$

where p is the acceptance probability of the MCMC kernel and $(1 - c)$ is a pre-specified probability that the particle is moved by the MCMC kernel at least once. This acceptance probability p can be estimated by traversing the particle set through the MCMC kernel one time and determining the proportion of particles which move. According to the above condition, the number of times that the particle set should move through the MCMC kernel will increase as the acceptance probability decreases. Further, as the number of data points increases, the MCMC move step becomes more computationally expensive as the likelihood for all observations needs to be evaluated a large number of times. As such, these issues will increase the computational time of this algorithm, potentially limiting the general applicability of this approach in adaptive design.

Algorithm 3 SMC algorithm

```
1: Draw $\boldsymbol{\theta}^j_{m,0}$ from prior $p(\boldsymbol{\theta}_m|M = m)$ and set $W^j_{m,0} = 1/N$ for $j = 1, 2, \ldots, N$ and for $m = 1, 2, \ldots, K$
2: Set $\log \hat{Z}_{m,0} = 0$ for $m = 1, 2, \ldots, K$
3: for $i = 1$ to $n$ do
4:    Find the $i$th optimal design point, $d_i$, by maximising the utility $U(d|\boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1})$
5:    Collect data $y_i$ at design point $d_i$
6:    for $m = 1$ to $K$ do
7:       Re-weight $w^j_{m,i} = W^j_{m,i-1} p(y_i|d_i, \boldsymbol{\theta}^j_{m,i}, M = m)$ for $j = 1, 2, \ldots, N$
8:       Update log evidence $\log \hat{Z}_{m,i} = \log \hat{Z}_{m,i-1} + \log \sum_{j=1}^N w^j_{m,i}$
9:       Normalise the weights $W^j_{m,i} = w^j_{m,i} / \sum_{k=1}^N w^k_{m,i}$ for $i = 1, 2, \ldots, N$
10:      Calculate $\text{ESS}_m = 1/\sum_{j=1}^N (W^j_{m,i})^2$
11:      if $\text{ESS}_m < E$ then
12:         Resample particle set $m$ yielding $\{\boldsymbol{\theta}^j_{m,i}, W^j_{m,i}\}^N_{j=1}$
13:         Compute the parameters of the MCMC proposal $\eta_{m,i}(\cdot|\cdot)$ using the particles $\{\boldsymbol{\theta}^j_{m,i}, W^j_{m,i}\}^N_{j=1}$
14:         for $i = 1$ to $N$ do
15:            Move particle $\boldsymbol{\theta}^j_{m,i}$ with an MCMC kernel of invariant distribution $\pi_i(\boldsymbol{\theta}^j_{m,i}|M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$ iterated $R_m$ times
16:         end for
17:         Reset $W^j_{m,i} = 1/N$ for $j = 1, 2, \ldots, N$
18:      else
19:         Set $\boldsymbol{\theta}^j_{m,i} = \boldsymbol{\theta}^j_{m,i-1}$ for $j = 1, 2, \ldots, N$
20:      end if
21:   end for
22: end for
```

Pseudo-code for the SMC algorithm is given in Algorithm 3. To initialise the algorithm, a sample of N particles $\{\boldsymbol{\theta}^i_{m,0}, W^i_{m,0}\}^N_{i=1}$ is drawn from the prior distribution for each model m. Then, at each iteration, a design point is selected by maximising a utility function based on the available information about each model and the parameter (line 4). As in the SLP algorithm, importance sampling is used here to efficiently approximate the expectation of the utility function.

Next, in our simulation study, a new data point is generated from an assumed model using the design obtained in the previous step (line 5). Once a data point is collected (or generated), the weights of the particle set and model evidence for each model are updated (lines 7 to 9), and the ESS_m is approximated for each model (line 10). If the ESS_m is less than a predefined threshold E (for a given model), then the resample and move steps are undertaken (lines 12 to 15). This process continues until a fixed number of data points have been observed.

3 New algorithm for Bayesian adaptive design

In this section, a novel adaptive design algorithm is introduced. Further, we explain how the model evidence and the posterior distribution are approximated within this new design algorithm.

3.1 Adaptive design using Laplace-based SMC algorithm

In this section, we describe our new algorithm for Bayesian adaptive design. The algorithm is constructed through adopting the re-weight step as given in the SMC algorithm but replacing the resample and move steps of this algorithm with Laplace importance sampling. We thus term this new algorithm the Laplace-based SMC (LP-SMC) algorithm. Given this, the algorithm initially proceeds as described above for the SMC algorithm. However, when the ESS_m for a given model becomes undesirably small, instead of running the resample and move steps, Laplace importance sampling (Kuk 1999; Beck et al. 2018) is used to approximate the posterior distribution. Such an importance sampling method has been previously used to reduce the computational complexity of double-loop MC for estimating expected utility functions in Bayesian design (Beck et al. 2018). For a similar purpose, Feng and Marzouk (2019) propose a layered multiple importance sampling scheme. In contrast, we propose Laplace importance sampling within SMC to improve the computational efficiency of the SMC algorithm and thus sequential Bayesian design. In our proposed algorithm, for each candidate model, this will yield a weighted particle set which approximates the appropriate posterior distribution. Given these particles will not generally have equal weight, the ESS_m of each particle set will not be N but should be close to N (and much larger than E) if the importance distribution is effective. Thus, within this algorithm, it is important that the importance distribution is effectively constructed and can be assessed accordingly. If we are able to do so, then this replacement avoids the computational cost associated with the move step in the SMC algorithm. Of course, the likelihood of all observations still needs to be evaluated within each iteration of our adaptive design algorithm, but this evaluation will be performed much fewer times than in the move step; thus, there will be significant computational savings in general, but particularly so when the likelihood is expensive to evaluate. Further, adopting Laplace importance sampling allows us to more precisely capture some departures from normality such as heavy-tailed posterior distributions and nonlinear posterior dependence between parameters when compared to the standard SLP algorithm.

To use Laplace importance sampling effectively in our LP-SMC algorithm, Pareto smoothing is adopted (Vehtari et al. 2017). Such smoothing has the effect of stabilising the importance weights and thus provides a more efficient approximation to the target distribution and estimates of quantities based on this distribution. More specifically, we fit the generalised Pareto distribution to the upper tail of the distribution of the importance weights derived from Laplace importance sampling and re-evaluate these weights based on this fitted distribution. This re-evaluation will smooth these more extreme weights, resulting in a more stable approximation to the target distribution. Additionally, in fitting the generalised Pareto distribution to these weights, we are provided with a diagnostic measure to determine whether the importance distribution appropriately captures the target distribution. This diagnostic measure is obtained through inspecting the estimated value of the shape parameter of the generalised Pareto distribution (denote as ξ) and thus can be inspected automatically throughout our algorithm. If this value is undesirably large, then it suggests that the Pareto-smoothed importance weights still have a heavy right tail. This indicates that the proposed importance distribution does not suitably capture the target distribution. In such cases, we can revert to a different importance distribution, for example, a proposal distribution based on the multivariate t distribution (as opposed to the multivariate normal distribution). The same Pareto smoothing approach can then be adopted to assess the appropriateness of this importance distribution. If this is again found to not be appropriate, then we can even revert to the move step from the SMC algorithm. This is indeed the sequence of steps that we adopt within our algorithm. Collectively, this ensures we obtain efficient particle approximations to the posterior distribution (of each model) within each iteration of our adaptive design algorithm.

To show how the LP-SMC algorithm works, we need to show how posterior distributions are approximated by Pareto-smoothed Laplace importance sampling within our adaptive design algorithm. To do this, we start by defining Laplace importance sampling. In such an approach, a Laplace approximation (i.e. a normal distribution) is used as the importance distribution in importance sampling. As such, the expectation in Eq. (4) is estimated as follows:

$$E[h(X)] = \int_X \frac{h(x)p(x)}{q_{LP}(x)} q_{LP}(x)\mathrm{d}x \approx \sum_{j=1}^{N} \frac{h(x_j)p(x_j)}{q_{LP}(x_j)}, \tag{12}$$

where $q_{LP}(x)$ is a normal density based on a Laplace approximation to the target distribution.

However, in general, the above normal distribution may not be an appropriate importance distribution in itself because the tails of this distribution may not capture the tails of the target distribution. To mitigate against this, the variance of the normal distribution is inflated. To do so in our algorithm, we multiple the variances by a factor of two (Brinch 2008). Adopting this approach leads to the following importance distribution in the LP-SMC algorithm:

$$\boldsymbol{\theta}_m | M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i} \sim MVN(\boldsymbol{\theta}_m^*, \boldsymbol{\Sigma}(\boldsymbol{\theta}_m^*)), \tag{13}$$

where $\boldsymbol{\Sigma}(\boldsymbol{\theta}_m^*) = [-\boldsymbol{H}(\boldsymbol{\theta}_m^*)]^{-1} + \mathrm{diag}([-\boldsymbol{H}(\boldsymbol{\theta}_m^*)]^{-1})$, and $\boldsymbol{\theta}_m^*$ and $\boldsymbol{H}(\boldsymbol{\theta}_m^*)$ are obtained as shown in Eqs. (6) and (7), respectively.

Here, the target distribution is the posterior distribution $p(\boldsymbol{\theta}_m | M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$ at the ith iteration. Thus, it can be approximated by drawing a sample of N particles $\{\boldsymbol{\theta}_{m,i}^j\}_{j=1}^N$, from the importance distribution given in Eq. (13) and giving them equal weight. Then, these particle weights are updated as follows:

$$w_{m,i}^j = \frac{p(\boldsymbol{\theta}_{m,i}^j | M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})}{p_{\mathrm{LIS}}(\boldsymbol{\theta}_{m,i}^j | M = m, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})}, \tag{14}$$

where $p_{\mathrm{LIS}}(\cdot|\cdot)$ is the density function of the importance distribution given in Eq. (13).

In some instances, the above importance weights can be highly variable and thus lead to an inefficient approximation to the target distribution. In such cases, these weights can be smoothed using Pareto smoothing as detailed in Vehtari et al. (2017). In our algorithm, this is achieved by replacing the $L = \min(N/5, 3\sqrt{N})$ largest weights above the threshold u with the expected value of the order statistics of the fitted generalised Pareto distribution as follows:

$$w_{m,i}^{s'} = F_{u,\sigma,\xi}^{-1}\left(\frac{s - 0.5}{L}\right), \quad s = 1, \ldots, L,$$

where $F_{u,\sigma,\xi}^{-1}$ is the inverse cumulative distribution function of the generalised Pareto distribution with the lower bound parameter u, scale parameter σ and the shape parameter ξ. Here, the estimated values of the parameters of the generalised Pareto distribution were obtained as described in Vehtari et al. (2019). We then replace the corresponding particle weights with the Pareto-smoothed weights.

In practice, we only evaluate densities that are proportional to the above posterior distribution, so the resulting weights should be normalised to obtain $W_{m,i}^j$, as shown in Eq. (5). Then, the weighted particle set, $\{W_{m,i}^j, \boldsymbol{\theta}_{m,i}^j\}_{j=1}^N$, represents the current posterior distribution for model m at the ith iteration.

In adopting our LP-SMC algorithm, we can also obtain computationally efficient estimates of the model evidences for each model. Here, we follow the same procedure as described for the SMC algorithm, see Eq. (11).

Algorithm 4 LP-SMC algorithm

```
1: Draw $\boldsymbol{\theta}_{m,0}^j$ from prior $p(\boldsymbol{\theta}_m|M = m)$ and set $W_{m,0}^j = 1/N$ for $j = 1, 2, \ldots, N$ and for $m = 1, 2, \ldots, K$
2: Set $\log \hat{Z}_{m,0} = 0$ for $m = 1, 2, \ldots, K$
3: for $i = 1$ to $n$ do
4:   Find the $i$th optimal design point, $d_i$, by maximising the utility $U(d|\boldsymbol{y}_{1:i-1}, \boldsymbol{d}_{1:i-1})$
5:   Collect data $y_i$ at design point $d_i$
6:   for $m = 1$ to $K$ do
7:     Re-weight $w_{m,i}^j = W_{m,i-1}^j p(y_i|d_i, \theta_{m,i-1}^j, M = m)$ for $j = 1, 2, \ldots, N$
8:     Update log evidence $\log \hat{Z}_{m,i} = \log \hat{Z}_{m,i-1} + \log \sum_{j=1}^N w_{m,i}^j$
9:     Normalise the weights $W_{m,i}^j = w_{m,i}^j / \sum_{k=1}^N w_{m,i}^k$ for $j = 1, 2, \ldots, N$
10:    Calculate $\mathrm{ESS}_m = 1/\sum_{j=1}^N (W_{m,i}^j)^2$
11:    if $\mathrm{ESS}_m < E$ then
12:      Find the importance distribution $MVN(\boldsymbol{\theta}_m^*, \boldsymbol{\Sigma}(\boldsymbol{\theta}_m^*))$ using Eqs. (8) and (13)
13:      Draw $\boldsymbol{\theta}_{m,i}^j$ from the importance distribution for $j = 1, 2, \ldots, N$
14:      Calculate the importance weights $w_{m,i}^j$ using Eq. (14)
15:      Estimate the Pareto-smoothed importance weights $w_{m,i}^{j'}$ and the shape parameter $\xi_{m,i}$
16:      if $\xi_{m,i} < 0.7$ then
17:        Normalise the weights $W_{m,i}^j = w_{m,i}^{j'} / \sum_{k=1}^N w_{m,i}^{k'}$
18:      else
19:        Repeat the steps 12–15 by using the multivariate t distribution $t_p(\boldsymbol{\theta}_m^*, \boldsymbol{\Sigma}(\boldsymbol{\theta}_m^*))$ with $p$ degrees of freedom as the importance distribution.
20:        if $\xi_{m,i} < 0.7$ then
21:          Repeat step 17
22:        else
23:          Use the MCMC move step in SMC algorithm (Algorithm 3, lines 13–15)
24:        end if
25:      end if
26:    else
27:      Set $\boldsymbol{\theta}_{m,i}^j = \boldsymbol{\theta}_{m,i-1}^j$ for $j = 1, 2, \ldots, N$
28:    end if
29:  end for
30: end for
```

Pseudo-code for the LP-SMC algorithm is provided in Algorithm 4. Steps 1–10 of the LP-SMC algorithm are the same as the corresponding steps of the SMC algorithm. Otherwise, when the ESS_m drops below E (for a given model), LP-SMC algorithm uses a Laplace approximation to find an efficient importance distribution for importance sampling (line 12). Next, Pareto-smoothed Laplace importance sampling is carried out to obtain the unnormalised importance weights (lines 13–15). Then, as suggested by Vehtari et al. (2017), we consider a threshold value of 0.7 for the shape parameter $\xi_{m,i}$ to check the stability of the importance weights (line 16) and thus determine whether the importance distribution is efficient with respect to the target distribution. If so (i.e. $\xi_{m,i} < 0.7$), then the weights are normalised (line 17). If not, then Pareto-smoothed Laplace importance sampling is carried out using the t distribution as the importance distribution (line 19). Again, stability of these weights is checked and normalised if they are stable (line 21). If not, then the move step from the SMC algorithm is implemented (line 22).

4 Simulation study

In this simulation study, three examples are considered to investigate the performance of the adaptive design algorithm discussed in Sect. 3. In each of the three examples considered, the three design algorithms (SMC, SLP and LP-SMC) using the total entropy utility (Borth 1975; McGree 2017) were used to sequentially select designs for the dual experimental goals of parameter estimation and model discrimination.

The total entropy utility proposed by Borth (1975) combines the parameter estimation utility, U_{P}, and the model discrimination utility, U_{M}, via the additivity property of entropy as follows:

$$U_{\mathrm{T}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$$
$$= U_{\mathrm{P}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}) + U_{\mathrm{M}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}),$$

where $U_{\mathrm{M}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}) = \log p(M = m|\boldsymbol{y}_{1:i}, z, \boldsymbol{d}_{1:i}, d)$ and $U_{\mathrm{P}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}) = \int_{\boldsymbol{\Theta}_m} p(\boldsymbol{\theta}_m|m, z, \boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i}, d) \log \left(p(z|d, \boldsymbol{\theta}_m, M = m)\right) \mathrm{d}\boldsymbol{\theta}_m - \log\left(\frac{Z_{m,i}(d,z)}{Z_{m,i}}\right)$. For the outcome z at the design point d, the normalising constant based on additionally observing z from design d ($Z_{m,i}(d, z)$) can be estimated using Eq. (11) as shown in Sect. 2.2.

Here, the parameter estimation utility is defined based on the Kullback–Leibler divergence between the prior and the posterior distribution of the parameters (Kullback and Leibler 1951). Therefore, this utility evaluates how much has been learned about $\boldsymbol{\theta}_m$. The model discrimination utility is defined based on the mutual information between the model indicator m and the predicted outcome z. Using the approximation in Eq. (10) and a particle approximation to the integral in the parameter estimation utility, we can approximate the utility $U_{\mathrm{P}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$ within each of the design algorithms as follows:

$$\hat{U}_{\mathrm{P}}(d, z, m|\boldsymbol{y}_{1:i}, \boldsymbol{d}_{1:i})$$
$$= \sum_{j=1}^{N} W_{m,i}^{j} \log\left(p(z|d, \boldsymbol{\theta}_{m,i}^{j}, M = m)\right) - \log \sum_{j=1}^{N} w_{m,i}^{j}. \tag{15}$$

It should be noted that when using the SLP algorithm for design selection, a posterior sample with equal weights ($W_{m,i}^{j} = 1/N$ for $j = 1, 2, \ldots, N$) is used, whereas a weighted sample (i.e. unequal weights) can be used in the other two algorithms. Further, when approximating the expected utility function within the three design algorithms, we used B MC samples, where $B < N$. These B samples are randomly drawn with replacement from the weighted sample $\{\theta_{m,i}^{j}, W_{m,i}^{j}\}_{j=1}^{N}$ at each iteration of the sequential design.

In Sects. 2 and 3, we have shown how to approximate the posterior model probability within each of the design algorithm (see Eqs. (9) and (11)). As such, with the approximation given in Eq. (15) for the parameter estimation utility, we can approximate the total entropy utility within each of the design algorithms presented in this paper.

In the first motivating example, we consider a logistic regression example from McGree (2017) where binary outcomes were observed. Then, we consider an example from Senarathne et al. (2019) where bivariate binary data were considered for estimating and discriminating between different Copula models. Finally, a biological application (Moffat et al. 2019) is considered where optimal designs were selected for a predator–prey functional response experiment with the same experimental goals of parameter estimation and model discrimination. For all three examples, binary and count data are considered. However, it is worth noting that our methodology is not limited to such data types. Further, in each example, the $\mathcal{M}$-closed perspective of Bernardo and Smith (2000) is considered for each set of candidate models. Within each simulation study, data were generated from a particular candidate model with specific parameter values. These data were then used to update the posterior model probabilities and the posterior distributions of all candidate models. All models were considered equally likely a priori.

For Examples 1 and 2, the approximate coordinate exchange algorithm (Overstall and Woods 2017) was used for optimising the total entropy utility over a continuous design space, with the default settings as detailed in Overstall et al. (2018b). As Example 3 considers a one-dimensional discrete design space, an exhaustive search was used to determine the next optimal design point. For the three examples considered in this manuscript, the optimisation algorithm proposed in Byrd et al. (1995) was used to find the mode of the posterior distribution (for use within the Laplace approximation).

When using the SMC algorithm for design selection, the tuning parameters N and E need to be specified. As such, we set $N = 5000$ and $E = 75\%$ for all examples considered in this paper. The same tuning parameter values were considered for the other two algorithms, as appropriate. The values for these tuning parameters were chosen based on trading off computational efficiency and accuracy. That is, in SMC, the parameter N can be chosen based on maintaining a reasonably high effective sample size and number of unique particle values throughout the algorithm (Liu 2008). This generally means that the value of N will increase with the number of estimable parameters and covariates. For design, there are similar considerations but we also need to consider the computational cost in finding Bayesian designs, and we note that computation will generally increase with N. As such, $N = 5000$ was found to be a reasonable trade-off between computational efficiency and maintaining reasonable effective sample sizes and unique particles. Further, as the results are subject to variability through the simulated data, all simulated studies were repeated a large number of times to explore the range of outcomes that could be observed. All simulations were run using R 3.4.2, and code to reproduce our results can be found in the following GitHub repository, https://github.com/SenarathneSGJ/Laplace_based_sequential_design_algorithms.

After the designs and the corresponding observed data were obtained from the SLP and LP-SMC algorithms for each example, they were re-evaluated within the SMC framework. This re-evaluation step was carried out so that the designs selected by each algorithm could be assessed (with respect to the utility) under the same procedure and thus not subject to different approximations. In doing so, we were then able to directly compare parameter estimation and model discrimination performance of designs found under each algorithm. Further, we also evaluated the computational time required to run each algorithm. This was simply the time required to run each algorithm for selecting a design with a fixed number of design points.

4.1 Example 1

Following the work of McGree (2017), consider an example with a binary response modelled by a 4-factor main effect logistic model (Model 1) as follows:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4, \tag{16}$$

where p is the probability of success, $\beta_0, \beta_1, \beta_2, \beta_3$ and β_4 are the parameters of the model, and the covariates $X_1, X_2, X_3, X_4 \in [-1, 1]$.

In this experiment, the goal is both parameter estimation and determining whether X_3 is needed in the model. Therefore, the discrimination problem is to determine model preference between the above model and the following model (Model 2):

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_4 X_4. \tag{17}$$

For data generation, parameter values were taken as $\boldsymbol{\beta} = [0, -3, 3, -3, 3]$, see McGree (2017). The prior distributions of the model parameters were assumed to be independent normal distributions with mean 0 and variance 100.

Results Figure 1 compares the distribution of designs selected from each of the design algorithms when Model 1 and 2 were responsible for data generation. This figure shows that for both models, the designs obtained from SMC and LP-SMC algorithms have similar distributions. However, the distribution of the designs obtained from the SLP algorithm is significantly different than those of the other two algorithms. For Model 1, the designs obtained from SMC and LP-SMC preferred points on the boundary, while the designs obtained from SLP preferred points near the boundary. When data were generated from Model 2, the majority of the design points selected from SMC and LP-SMC algorithms were again on the boundary. In contrast, the points selected by the SLP algorithm were rather varied and often away from the boundaries. In Fig. 2, the posterior model probabilities of the true model over each iteration of the experiment are shown so that the model discrimination performance of the designs could be assessed. Here, for both models, the optimal designs obtained from SMC and LP-SMC algorithms perform well for discrimination compared to the optimal designs obtained from the SLP algorithm. As such, fewer design points were required to discriminate between these two models when designs were obtained from the SMC or LP-SMC algorithms when compared to the SLP algorithm.

Next, we assessed the estimation performance of designs obtained from each of the algorithms. For this purpose, the log-determinant of the variance–covariance matrix of the each intermediate posterior distribution was considered. Figure 3 shows the inter-quartile range of the distribution of these log-determinant values across all simulations. As can be seen, for both models, the posterior distributions obtained from the SMC and LP-SMC algorithms had lower log-determinant values, and hence higher precision, when compared to the SLP algorithm.

Finally, we evaluated the computational time of each of the design algorithms. For the comparison, the inter-quartile ranges of the distributions of cumulative time required to run each algorithm were plotted (Fig. 4). These results show that the LP-SMC algorithm was the fastest, and SMC was the slowest algorithm of the three. For a more direct comparison of run times, the reader is referred to Fig. 15. For the LP-SMC algorithm, it is worth noting that the MCMC step (line 23) was not needed throughout any of the simulations. This suggests that our approach to forming an efficient proposal

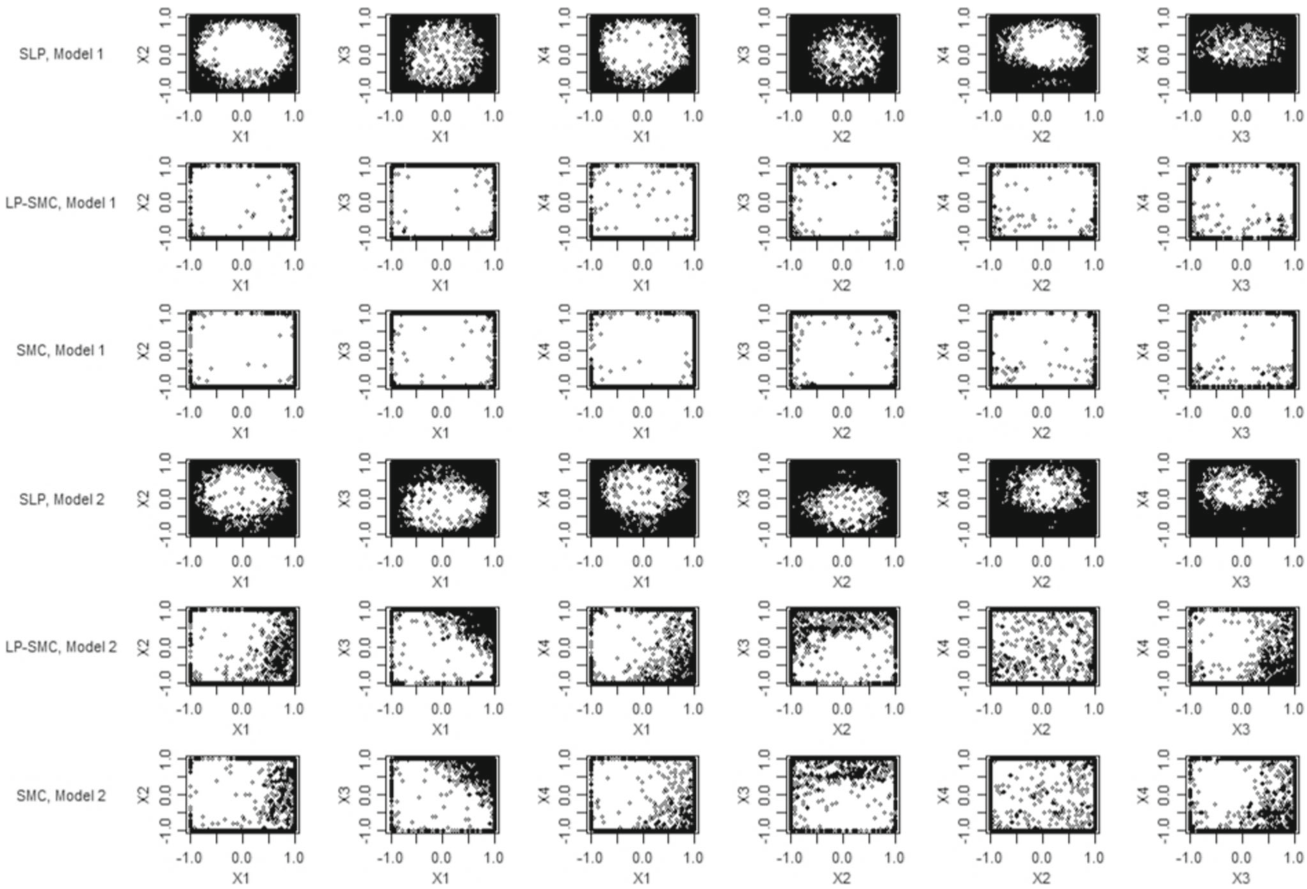


Fig. 1 The selected optimal design points over 100 simulations when data were generated from the first model (rows 1–3) and the second model (rows 4–6) from Example 1

distribution for the Laplace importance sampling step was effective for this example. That is, for this example, the tails of the proposal distribution obtained at each iteration of the adaptive design had sufficient coverage to capture the target distribution appropriately.

4.2 Example 2

Motivated by the work of Denman et al. (2011) and Senarathne et al. (2019), consider the following two binary responses (Y_1 and Y_2) where each response was modelled by a 3-factor main effect logistic model as follows:

$$\log\left(\frac{p_1}{1-p_1}\right) = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3, \qquad (18)$$

$$\log\left(\frac{p_2}{1-p_2}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3. \qquad (19)$$

In this example, Copula models are considered to describe the dependence between the two binary responses. To define the Copula model, we consider the joint probability distribution of the bivariate binary response $p_{y_1,y_2} = \text{prob}(Y_1 = y_1, Y_2 = y_2)$, $y_1, y_2 = 0, 1$. This has four possible outcomes {(0,0),(0,1),(1,0),(1,1)} where '1' represents a success and '0' a failure. The Copula representation (Denman et al. 2011) of the bivariate distribution can be expressed as:

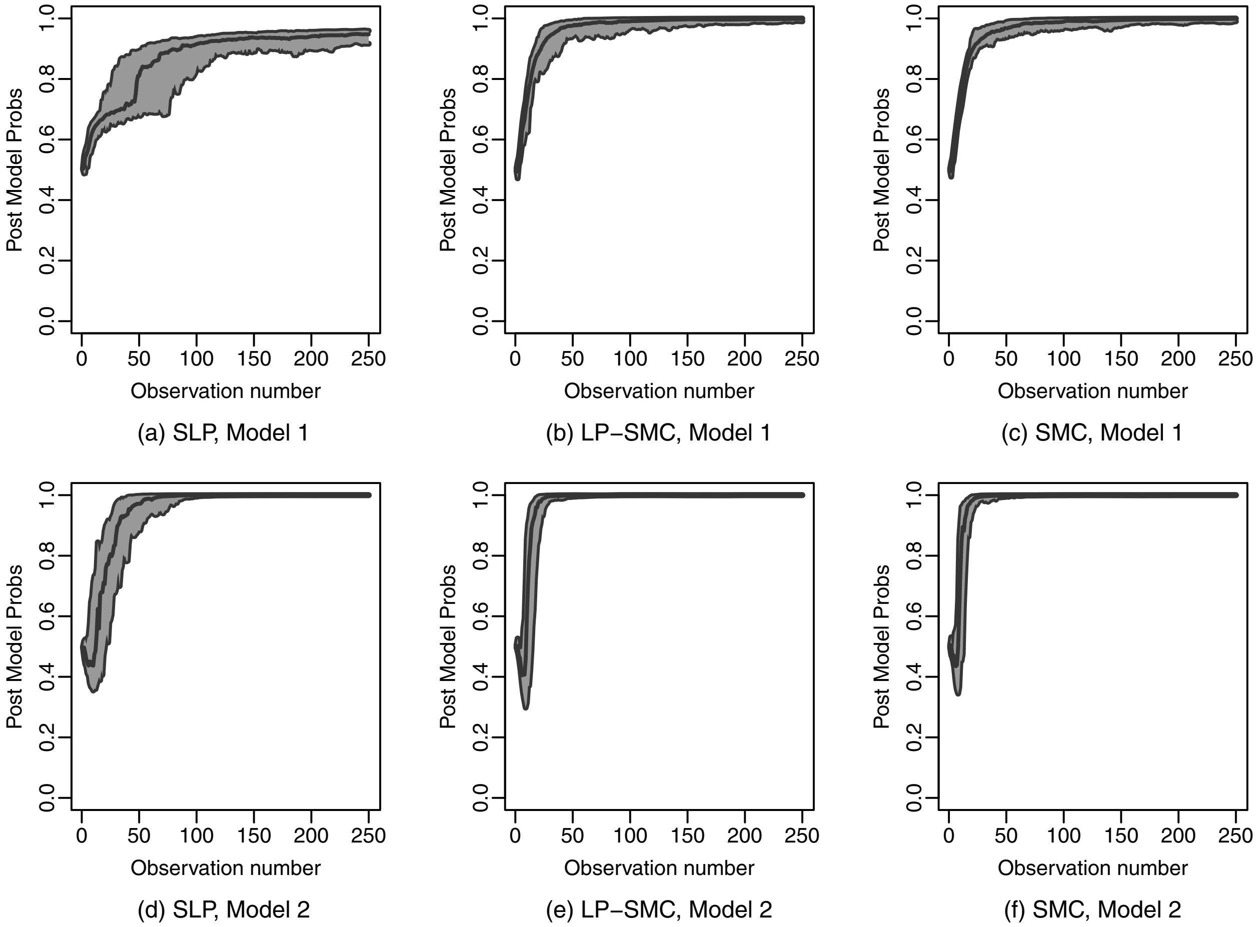
$$\begin{aligned} p_{11} &= C(\pi_1, \pi_2; \alpha), & p_{10} &= \pi_1 - p_{11}, \\ p_{01} &= \pi_2 - p_{11}, & p_{00} &= 1 - \pi_1 - \pi_2 + p_{11}, \end{aligned}$$

where π_1 and π_2 are the marginal probabilities of success of the responses Y_1 and Y_2, respectively, and α is the Copula parameter.

Motivated by the work of Senarathne et al. (2019), we selected the following two Copula models for this example. In the first Copula model, it is assumed that there is a positive association between the two responses, and hence, the Frank Copula is considered to construct the joint distribution as follows:

$$C(\pi_1, \pi_2; \alpha) \\ = -\alpha^{-1} \log\left(1 + \frac{(e^{-\alpha\pi_1} - 1)(e^{-\alpha\pi_2} - 1)}{e^{-\alpha} - 1}\right), \quad \alpha \neq 0. \qquad (20)$$

In the second Copula model, it is assumed that the two responses are independent, and hence, the Product Copula is considered to construct the joint distribution as follows:



(a) SLP, Model 1

(b) LP–SMC, Model 1

(c) SMC, Model 1

(d) SLP, Model 2

(e) LP–SMC, Model 2

(f) SMC, Model 2

Fig. 2 The distribution of the posterior model probabilities of the true model over 100 simulations for 250 observations from Example 1 under each design algorithm

$$C(\pi_1, \pi_2) = \pi_1 \pi_2. \tag{21}$$

For data generation, parameter values were taken as $\boldsymbol{\beta} = [1, 4, 1, -1]$ and $\boldsymbol{\gamma} = [1, -0.5, 1, -1]$, see Denman et al. (2011). It should be noted that there is a direct relationship between the Frank Copula parameter α and the Kendall's tau (τ) value (Nelsen 2006). As such, it is convenient to use τ for comparing dependence between bivariate responses as this parameter has the same interpretation under different Copula models. In Senarathne et al. (2019), different τ values were considered to generate data, and the corresponding dual-purpose designs were compared to evaluate the impact of the strength of the dependence on design selection. However, here we select designs to benchmark the new adaptive design algorithm and the SLP algorithm against the SMC algorithm. As such, a single τ value (0.75) was considered to generate data from the Frank Copula model. The prior distributions of the model parameters were assumed to be independent normal distributions with mean 0 and variance 16 as given in Senarathne et al. (2019). Since a positive dependence between the responses was assumed, a normal prior on $\log\left(\frac{\tau}{1-\tau}\right)$ with mean 0 and variance 2.25 was placed for τ.

Results Figures 5 and 6 show the distribution of the designs selected from each of the design algorithms when Frank Copula model and Product Copula model were responsible for data generation, respectively. Similar to the first example, optimal designs selected from SMC and LP-SMC algorithms had similar distributions when each Copula model was generating the data. The distribution of designs selected from the SLP algorithm is slightly different, with this being most noticeable in the pairwise plot of x_2 versus x_3.

When data were generated from the Frank Copula, covariates x_2 and x_3 preferred points on the boundaries, while the covariate x_1 preferred points around 0. When the Product Copula was responsible for data generation, covariate x_1 of the designs obtained from SMC and LP-SMC algorithms preferred the range of values from -0.5 to 0.5, while x_2 and x_3 preferred boundary points similar to the Frank Copula. As such, when using these two algorithms for design selec-

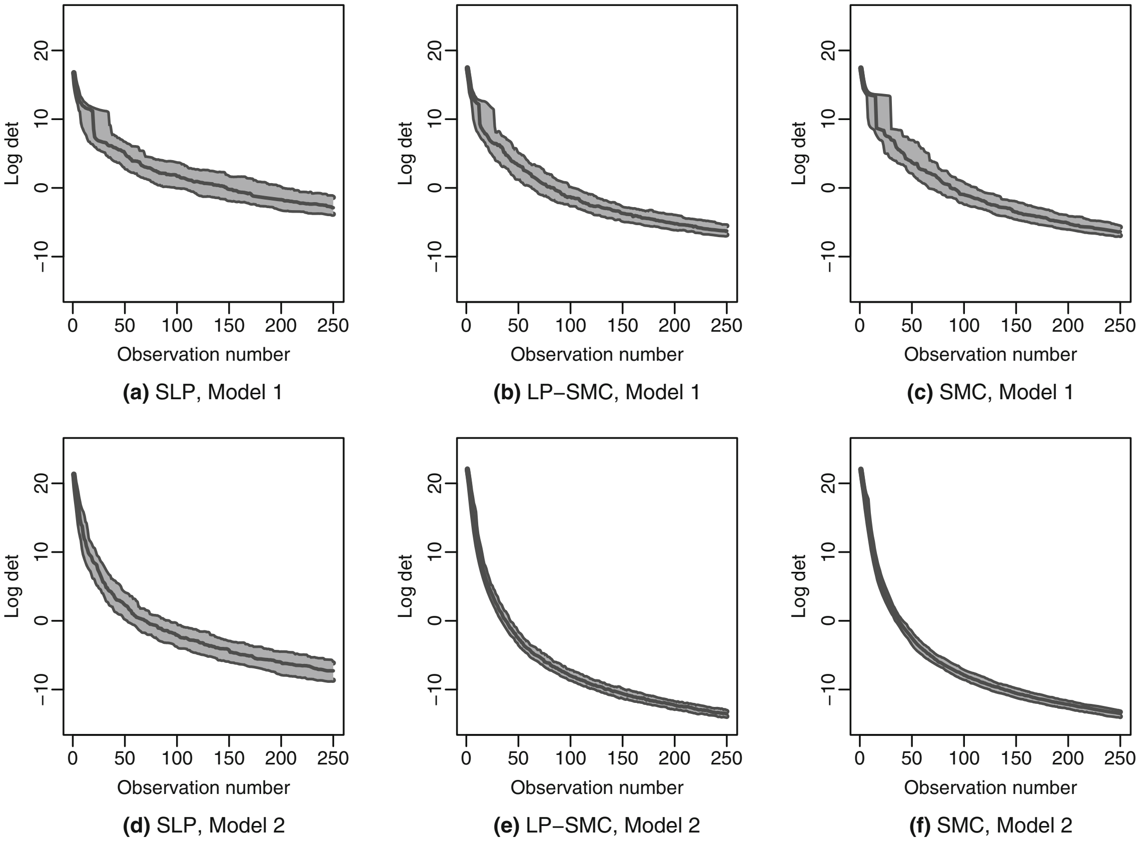


Fig. 3 The inter-quartile range of the distribution of the log-determinant of the posterior variance–covariance matrix for each design point over 100 simulations for 250 observations from Example 1

tion, there was a significant difference between the optimal designs when Product Copula was generating data compared to the Frank Copula. However, there was not a significant difference between the designs selected from the SLP algorithm when each Copula model was responsible for data generation.

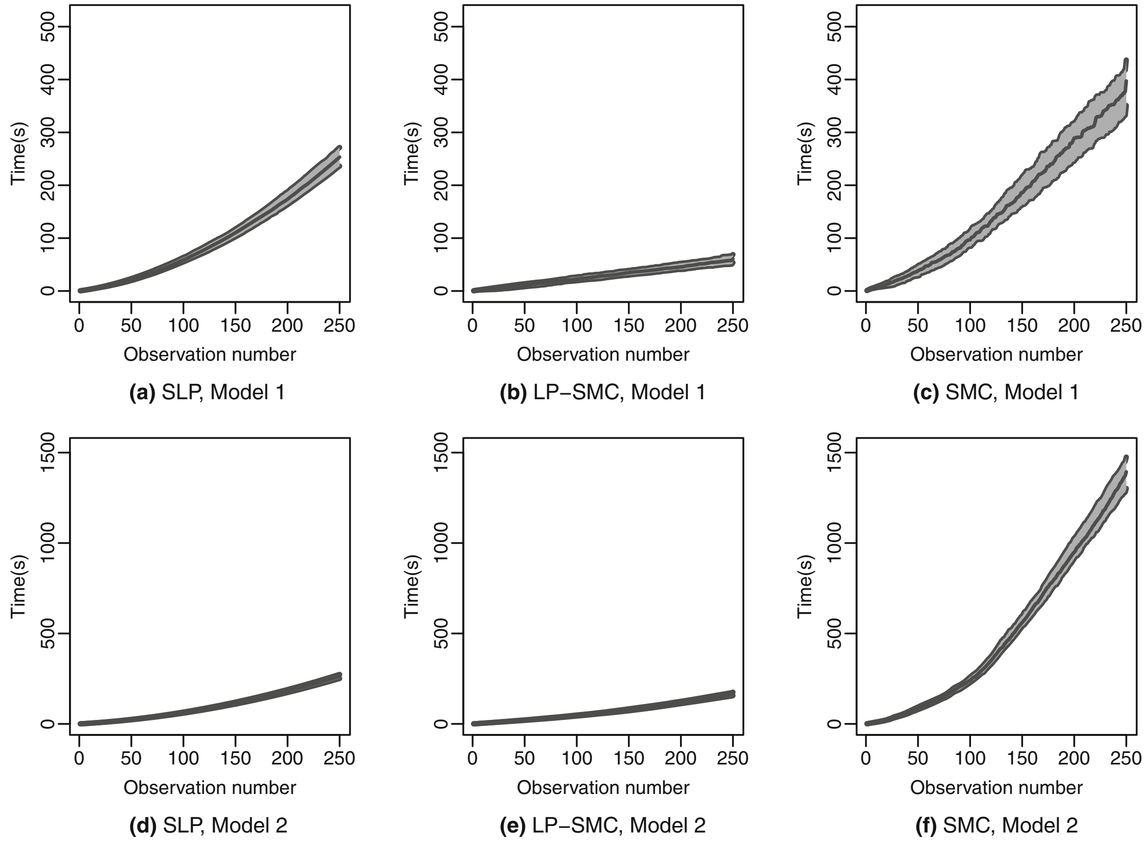
The distribution of the posterior model probabilities for the true Copula model over each iteration of the adaptive design algorithms is displayed in Fig. 7. For this example, the designs obtained from each of the three design algorithms perform equally well for discriminating between the Copula models. However, when data were generated from the Frank Copula model, a fewer number of design points were required to discriminate between the two Copula models.

Figure 8 displays the parameter estimation results of each Copula model when using the adaptive design algorithms for design selection. According to Fig. 8, for both Copula models, the optimal designs obtained from each algorithm performed similarly well for parameter estimation.

Figure 9 shows the inter-quartile range of the distributions of cumulative time required to run each algorithm. As can be seen, LP-SMC was the fastest algorithm when data were generated based on the Frank Copula, while it was the second fastest algorithm when data were generated based on the Product Copula. Again, the SMC algorithm was the least computationally efficient algorithm when either Copula model was responsible for data generation. For further comparison of run times, the reader is referred to Fig. 15. Of note, for this example, the MCMC step (line 23) of the LP-SMC algorithm was required in 443 out of 25,000 iterations (i.e. 100 simulated studies each with 250 observations) when the Frank Copula was responsible for generating data. When the Product Copula was responsible for data generation, this MCMC step was required in 1494 out of 25,000. Again, this suggests that efficient importance distributions are being formulated for the Laplace importance sampling step.

4.3 Example 3

Following the work of Moffat et al. (2019), consider a functional response experiment where predator–prey interaction is modelled by two mechanistic models developed in Holling (1959). The first model is the Holling's type II functional response model which is also referred as the disc equation. The Holling's type II functional response model is given by the ordinary differential equation:



(a) SLP, Model 1

(b) LP−SMC, Model 1

(c) SMC, Model 1

(d) SLP, Model 2

(e) LP−SMC, Model 2

(f) SMC, Model 2

Fig. 4 The inter-quartile ranges of the distributions of the cumulative time required to run each algorithm over 100 simulated studies in Example 1

Fig. 5 The selected optimal design points over 100 simulations when data were generated from the Frank Copula model from Example 2

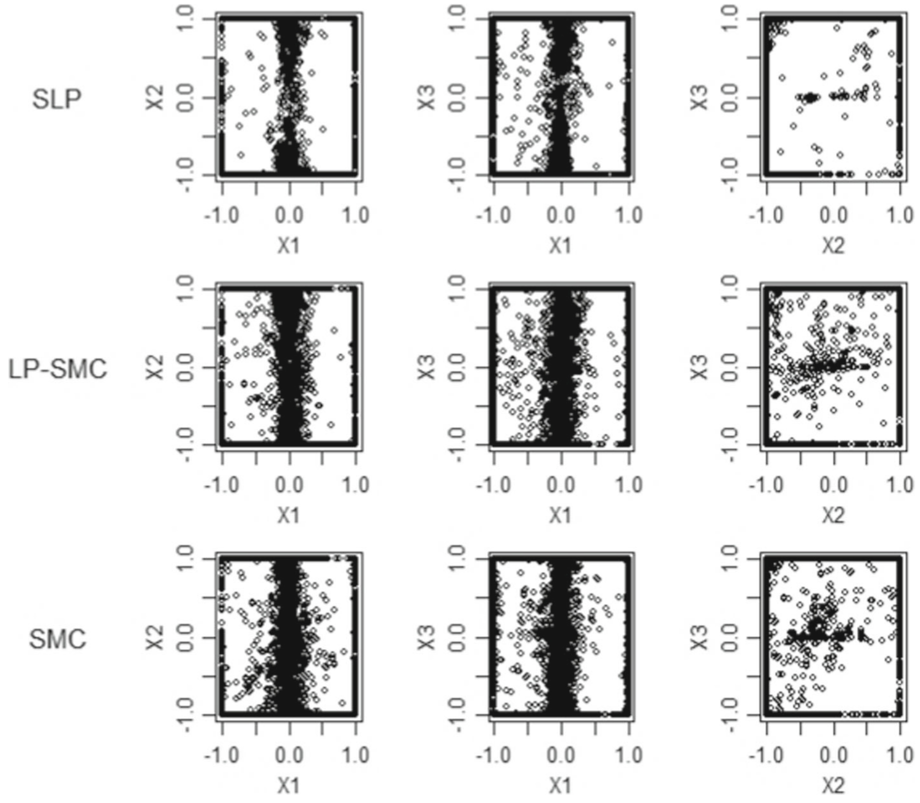
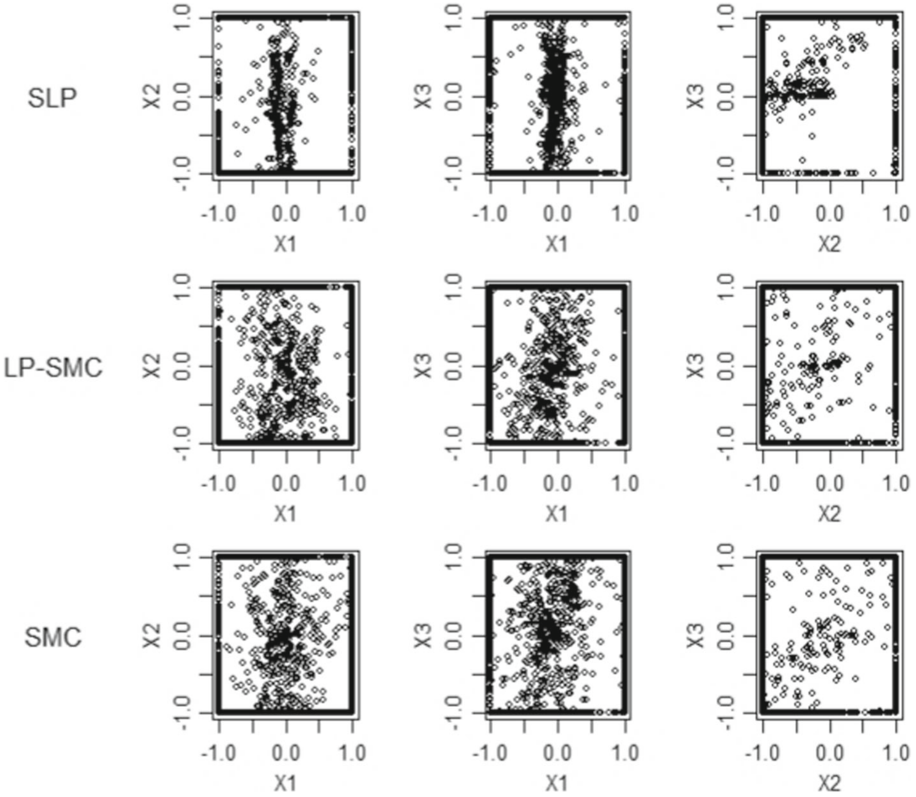
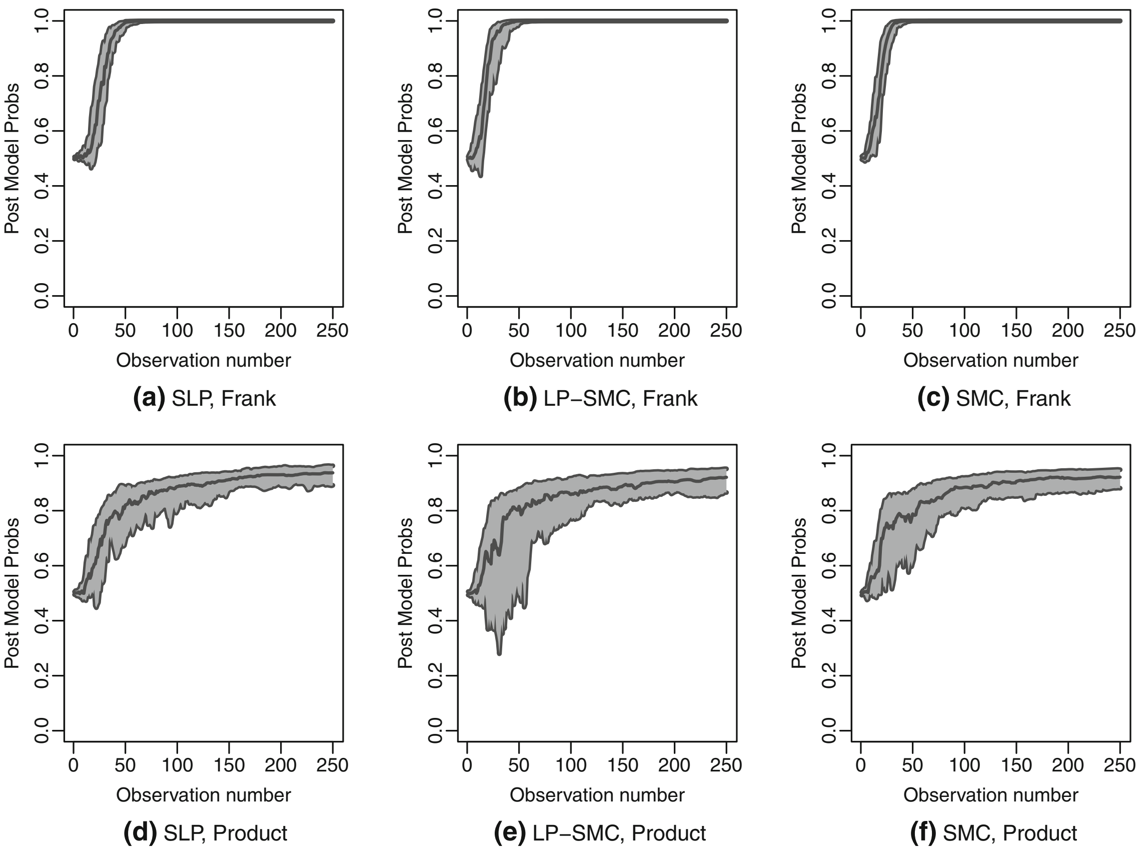


Fig. 6 The selected optimal design points over 100 simulations when data were generated from the Product Copula model from Example 2





(a) SLP, Frank

(b) LP−SMC, Frank

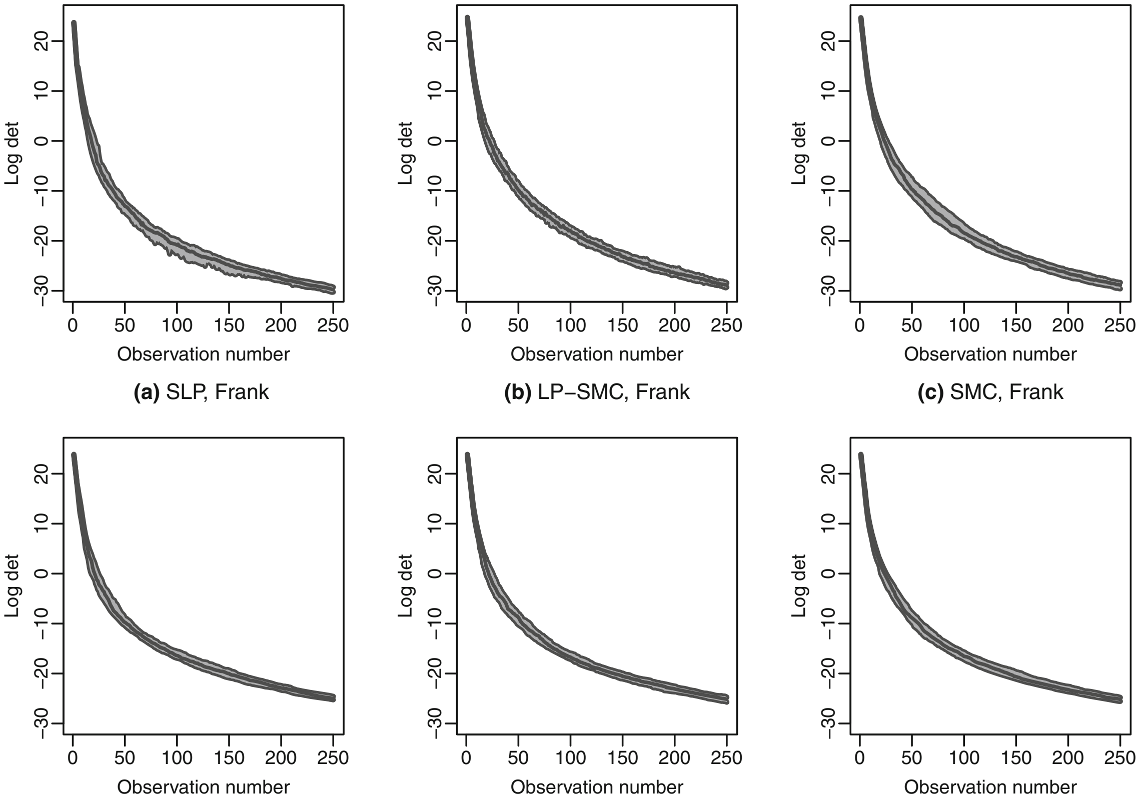
(c) SMC, Frank

(d) SLP, Product

(e) LP−SMC, Product

(f) SMC, Product

Fig. 7 The distribution of the posterior model probabilities of the true model over 100 simulations for 250 observations from Example 2 under each design algorithm



(a) SLP, Frank **(b)** LP–SMC, Frank **(c)** SMC, Frank

(d) SLP, Product **(e)** LP–SMC, Product **(f)** SMC, Product

Fig. 8 The inter-quartile range of the distribution of the log-determinant of the posterior variance–covariance matrix for each design point over 100 simulations for 250 observations from Example 2

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{aN}{1 + aT_\mathrm{h}N}, \tag{22}$$

where N denotes the prey density in a given area, a represents the per capita prey consumption in low prey densities, and T_h is the handling time per prey attacked. Here, we assumed that initially the system contained N_0 prey in the given area.

By extending the disc equation, Holling (1959) developed another functional response model known as Holling's type III functional response model. The Holling's type III functional response model is given by:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{aN^2}{1 + aT_\mathrm{h}N^2}. \tag{23}$$

The primary objective of this experiment is to efficiently estimate the parameters a and T_h in Eqs. (22) and (23). As such, this experiment should be conducted with different initial prey densities. We denote the initial prey density for each observation, t, as $N_{0,t}$ for $t = 1, 2, \ldots, T$. Here, T represents the total number of observations of the predator–prey system. The number of prey consumed, $n_{e,t}(\tau)$, in a fixed time period, τ, is the response variable of this experiment. To account for the uncertainty in the data, it is convenient to link the Holling's type II and type III models to probabilistic models. Therefore, the binomial and beta-binomial distributions are considered to model the response, $n_{e,t}(\tau)$, as detailed in Moffat et al. (2019).

Let us consider a single experiment where the number of prey consumed in a fixed time period τ can be defined as $n_e(\tau)$. For the case where $n_e(\tau)$ follows a binomial distribution, the probability of a success, $p(\tau)$, can be defined as follows:

$$n_e(\tau) \sim \text{Binom}(N_0, p(\tau)) \text{ and}$$
$$p(\tau|a, T_\mathrm{h}) = \frac{N_0 - N(\tau|a, T_\mathrm{h})}{N_0},$$

where $p(\tau)$ is the probability that single prey has been consumed in a fixed time period, τ. The number of prey population, $N(\tau|a, T_\mathrm{h})$, after the time period τ can be obtained by solving the differential equation in Holling's type II or type III model.

For the case where $n_e(\tau)$ follows a beta-binomial distribution, we consider a re-parameterised version of the beta-binomial distribution as detailed in Fenlon and Faddy (2006). Thus, the expected proportion, $p(\tau)$, and the over-dispersion parameter, λ, of the beta-binomial distribution are given by:

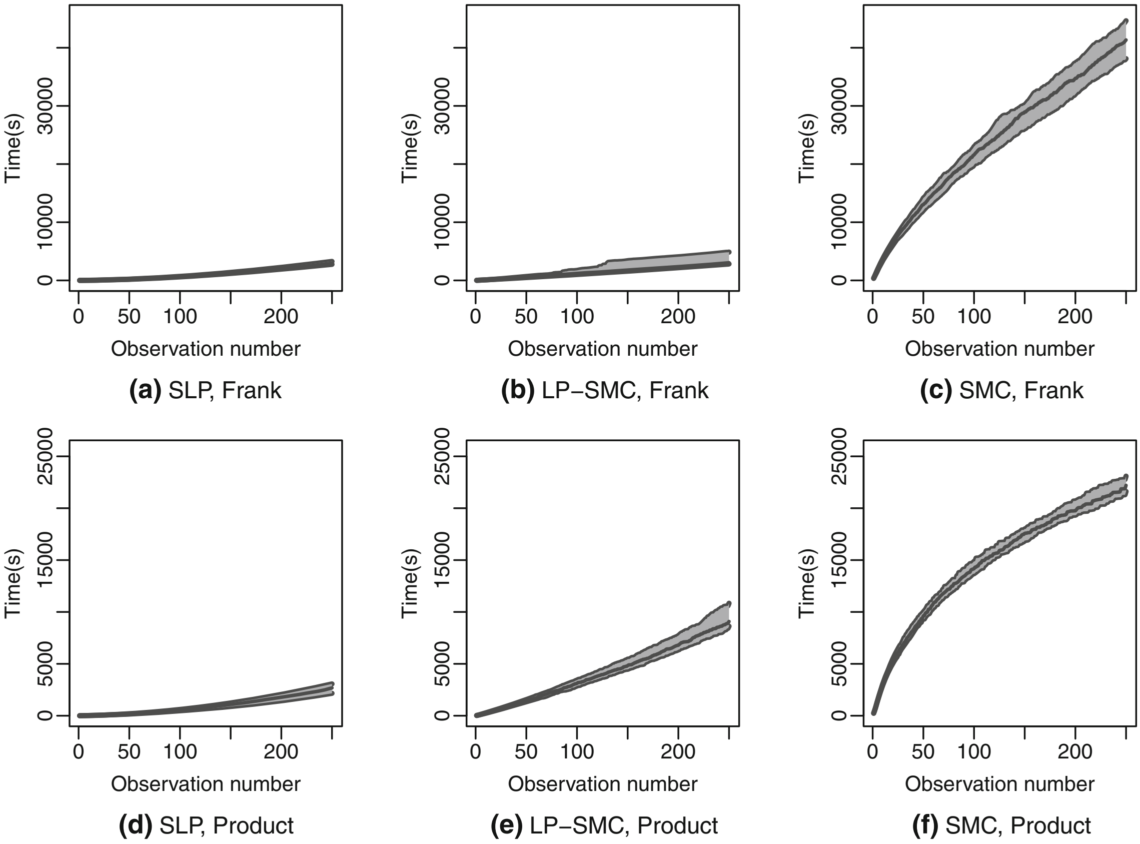


Fig. 9 The inter-quartile ranges of the distributions of the cumulative time required to run each algorithm over 100 simulated studies in Example 2

$$n_e(\tau) \sim \text{BetaBinom}(N_0, p(\tau), \lambda),$$

$$p(\tau|a, T_\text{h}) = \frac{\alpha}{\alpha+\beta} = \frac{N_0 - N(\tau|a, T_\text{h})}{N_0} \text{ and } \lambda = \frac{1}{\alpha+\beta},$$

where α and β are the parameters of the beta function.

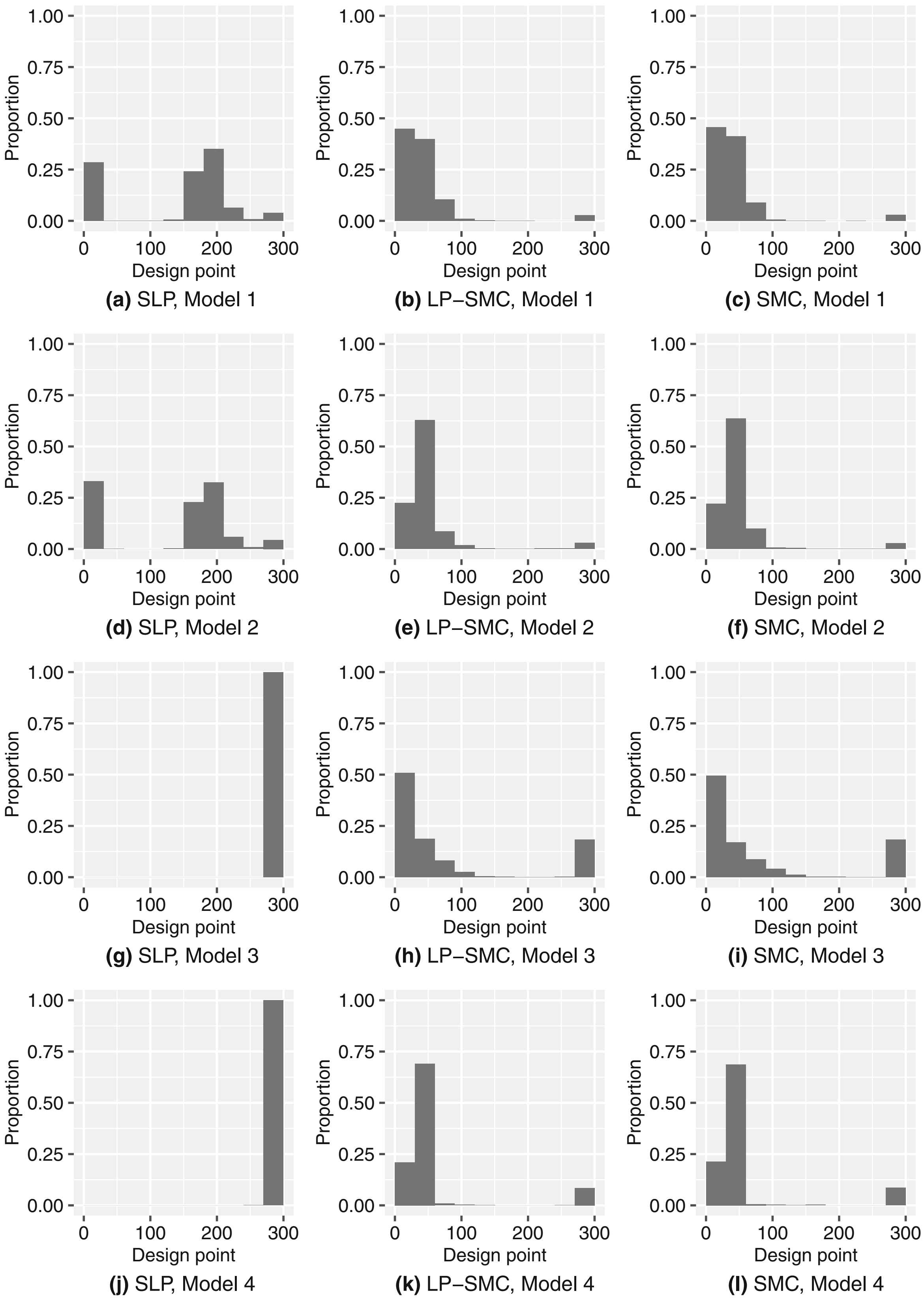
As each probabilistic model links to a particular mechanistic model, four different models can be obtained to model the response $n_e(\tau)$ as follows.

For our simulation study, we consider the four different models in Table 1 with the following true parameter values: $a = 0.5$, $T_\text{h} = 0.7$ and $\lambda = 0.5$, where relevant. The total number of observations collected in the experiment is $T = 40$, and the total exposure time is $\tau = 24$ h. This study was undertaken in a restricted design space with only a fixed number of design points being available. That is, $N_{0,t} \in \{1, 2, \ldots, 300\}$ for $t = 1, 2, \ldots, T$. The response $n_{e,t}(\tau)$ can take any value from the set $\{0, 1, \ldots, N_{0,t}\}$ for $t = 1, 2, \ldots, T$. The prior distributions of the model parameters are given by $\log(a) \sim N(-1.4, 1.35^2)$, $\log(T_\text{h}) \sim N(-1.4, 1.35^2)$ and $\log(\lambda) \sim N(-1.4, 1.35^2)$. Moffat et al. (2019) considered adaptive design for these predator–prey experiments based on the SMC algorithm and the total entropy utility as given in McGree (2017). We consider this approach as well as that based on employing the SLP and LP-SMC algorithms.

Table 1 Models used for the experiment in Example 3

Distribution of $n_e(\tau)$	Mechanistic model	
	Holling's type II	Holling's type III
Beta-binomial	Model 1	Model 2
Binomial	Model 3	Model 4

Results Figure 10 compares the distribution of designs selected from each of the three design algorithms when each of the four models was responsible for data generation. In this example, the designs obtained from SMC and LP-SMC algorithms had similar distributions for all four models. However, the distribution of designs obtained from the SLP algorithm was substantially different. That is, when Model 1 or Model 2 was responsible for data generation, the designs obtained from SLP preferred values close to 0 or 200, while the designs obtained from SMC and LP-SMC preferred the range of values from 0 to 100. When Model 3 or Model 4 was responsible for data generation, the designs obtained from SLP preferred the value of 300, while the designs obtained from the other



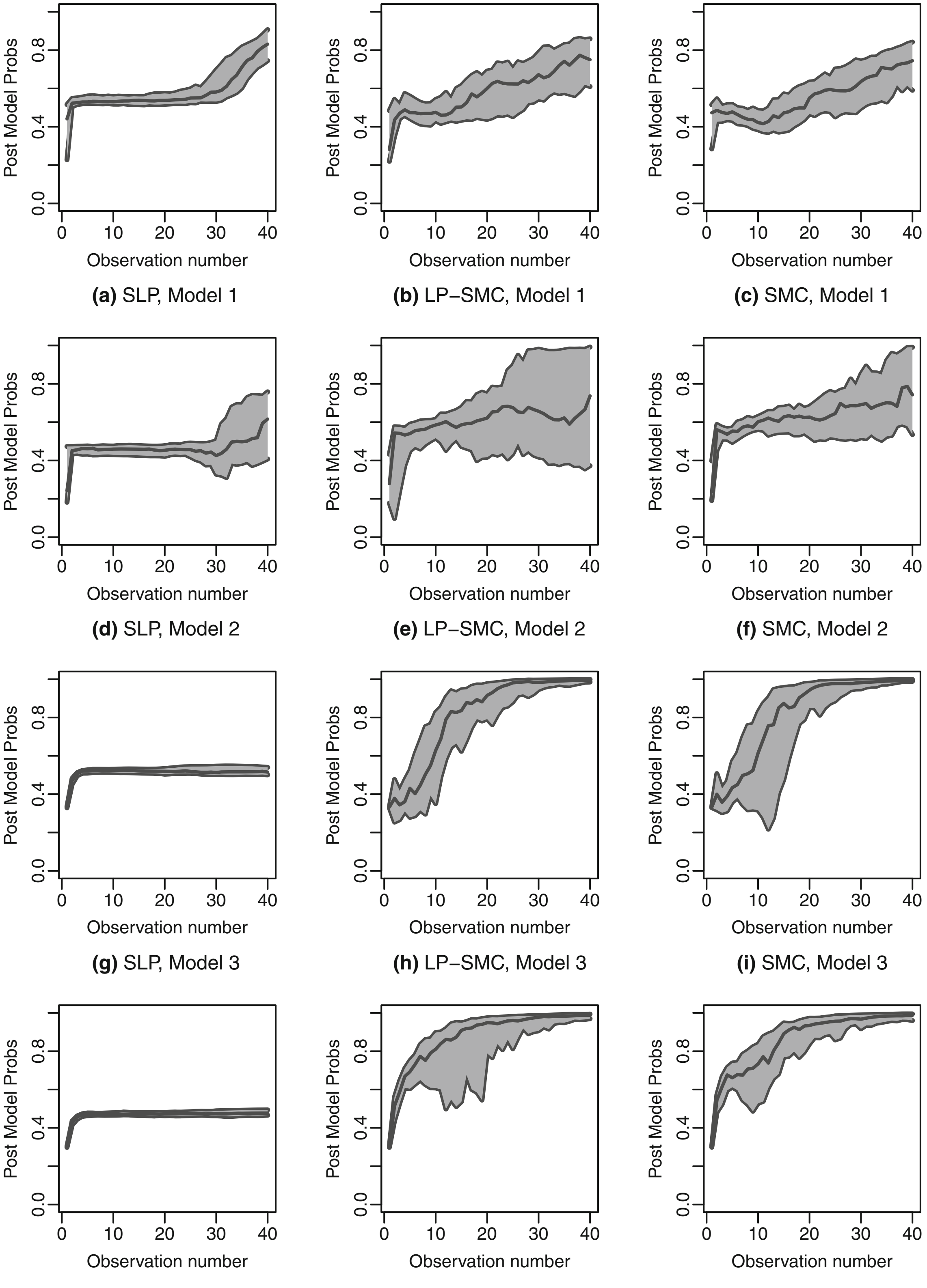
(a) SLP, Model 1 **(b)** LP–SMC, Model 1 **(c)** SMC, Model 1

(d) SLP, Model 2 **(e)** LP–SMC, Model 2 **(f)** SMC, Model 2

(g) SLP, Model 3 **(h)** LP–SMC, Model 3 **(i)** SMC, Model 3

(j) SLP, Model 4 **(k)** LP–SMC, Model 4 **(l)** SMC, Model 4

Fig. 10 The proportion of the selected optimal design points over 50 simulations when data were generated from Model 1 (first row), Model 2 (second row), Model 3 (third row) and Model 4 (fourth row) from Example 3



(a) SLP, Model 1

(b) LP−SMC, Model 1

(c) SMC, Model 1

(d) SLP, Model 2

(e) LP−SMC, Model 2

(f) SMC, Model 2

(g) SLP, Model 3

(h) LP−SMC, Model 3

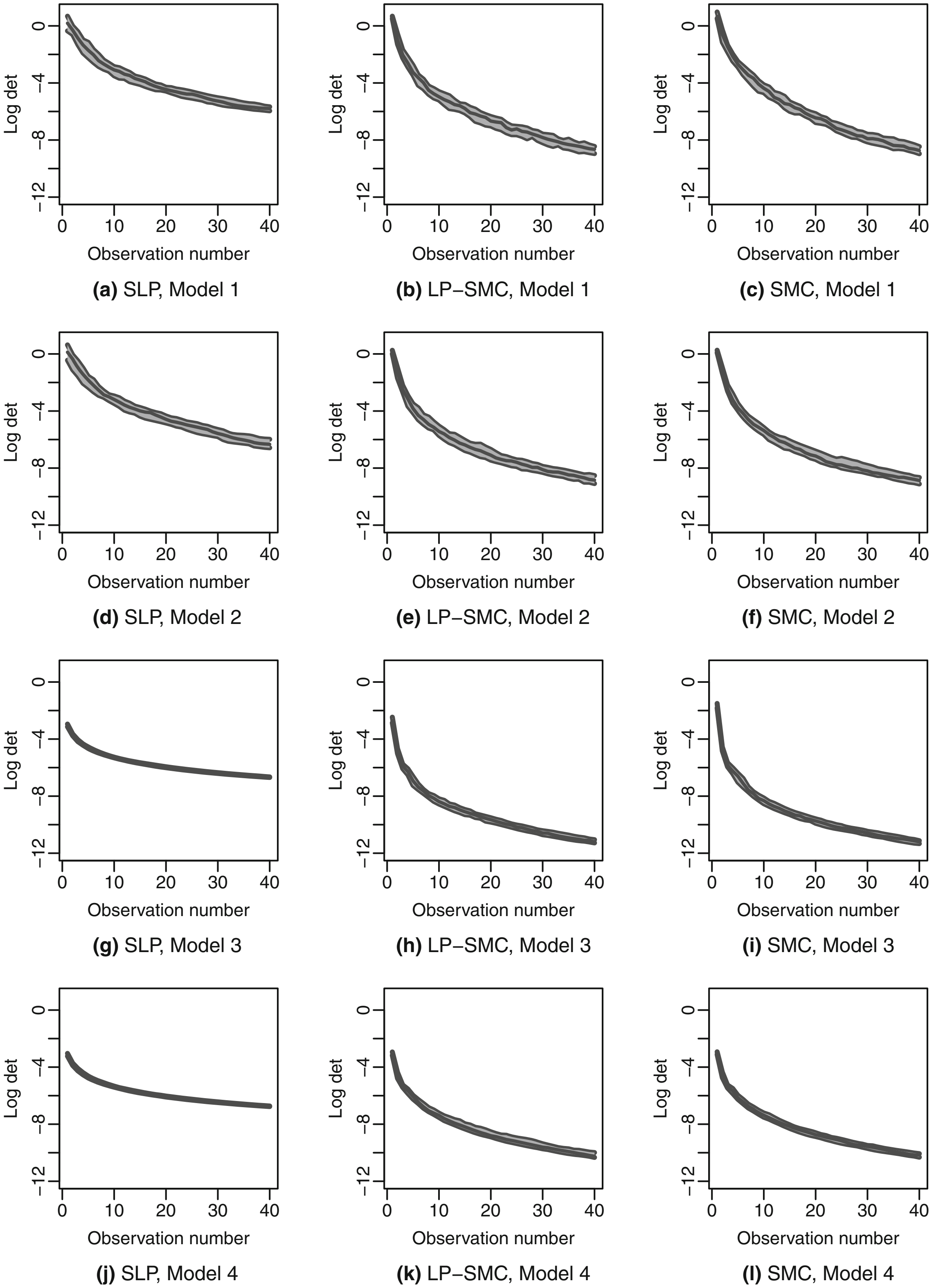
(i) SMC, Model 3

(j) SLP, Model 4

(k) LP−SMC, Model 4

(l) SMC, Model 4

Fig. 11 The distribution of the posterior model probabilities of the true model over 50 simulations for 40 observations from Example 3 under each design algorithm



(a) SLP, Model 1 **(b)** LP−SMC, Model 1 **(c)** SMC, Model 1

(d) SLP, Model 2 **(e)** LP−SMC, Model 2 **(f)** SMC, Model 2

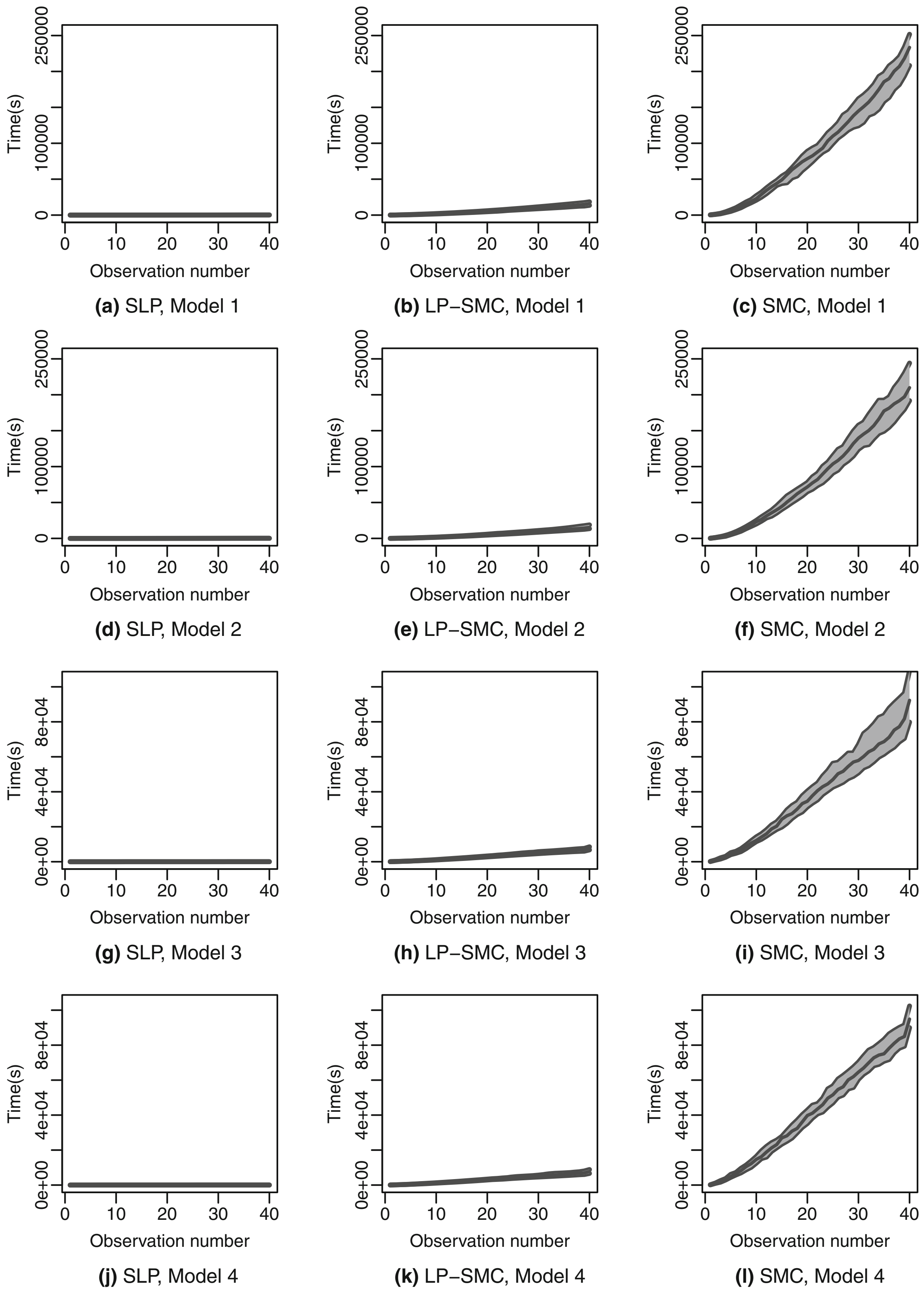
(g) SLP, Model 3 **(h)** LP−SMC, Model 3 **(i)** SMC, Model 3

(j) SLP, Model 4 **(k)** LP−SMC, Model 4 **(l)** SMC, Model 4

Fig. 12 The inter-quartile range of the distribution of the log-determinant of the posterior variance–covariance matrix for each design point over 50 simulations for 40 observations from Example 3

two algorithms preferred the range of values from 0 to 100 and close to 300.

In Fig. 11, the distribution of posterior model probabilities of the true model over each iteration of the selected algorithm



(a) SLP, Model 1 **(b)** LP−SMC, Model 1 **(c)** SMC, Model 1

(d) SLP, Model 2 **(e)** LP−SMC, Model 2 **(f)** SMC, Model 2

(g) SLP, Model 3 **(h)** LP−SMC, Model 3 **(i)** SMC, Model 3

(j) SLP, Model 4 **(k)** LP−SMC, Model 4 **(l)** SMC, Model 4

Fig. 13 The inter-quartile ranges of the distributions of the cumulative time required to run each algorithm over 50 simulated studies in Example 3

is shown. As can be seen, for all four models, the designs obtained from SMC and LP-SMC algorithms perform well for model discrimination compared to the designs obtained from the SLP algorithm. A possible reason for this is the poor approximation to the posterior distribution (of which utility evaluation and therefore design choice are based on) given within the SLP algorithm, see Fig. 16 for an example. Of note, it appears to be difficult to discriminate between the models when data were generated based on the designs obtained from SLP algorithm. Overall, it appears to be relatively difficult to determine a preferred model when data were generated from either Model 1 or Model 2. This is highlighted by no algorithm yielding a median posterior model probability of 1.

Figure 12 compares the parameter estimation results of each model when designs were obtained from each of the three design algorithms. For all four models, the posterior distributions obtained from the designs of either SMC or LP-SMC algorithm had lower log-determinant values compared to those obtained from the designs of the SLP algorithm.

Figure 13 shows the inter-quartile range of the distributions of cumulative time required to run each algorithm (again, also see Fig. 15). As can be seen, SLP was the fastest algorithm, with LP-SMC being the second fastest and SMC being the slowest algorithm among the three. For this example, the MCMC step of the LP-SMC algorithm was required in a small number of iterations when each model was responsible for data generation. More specifically, out of the total 2000 iterations (i.e. 50 simulations each with 40 observations), the MCMC step of the LP-SMC algorithm was required in 213, 261, 319 and 332 iterations when Model 1, 2, 3 and 4 were responsible for data generation, respectively.

5 Discussion

In this article, we have presented a novel adaptive design algorithm to efficiently design experiments in the presence of parameter and model uncertainty. This design algorithm was derived by replacing the resample and move steps of the standard SMC algorithm with a Pareto-smoothed Laplace importance sampling step. This step was used to significantly reduce the computational expense of the standard SMC algorithm. Notably, as in the SMC algorithm, our adaptive design algorithm provides a computationally efficient approximation to the posterior distribution and the model evidence at each iteration of the experiment.

Three adaptive design examples were considered to benchmark the proposed adaptive design algorithm against the standard SMC and SLP algorithms, in terms of computational time and design efficiency. In each example, designs were selected for the dual experimental goals of parameter estimation and model discrimination using the total entropy utility. For all three examples, both LP-SMC and SMC algorithms performed equally well in terms of design efficiency. However, the designs selected under the SLP algorithm were less efficient than the designs found by the other two algorithms. In terms of computational time, there was significant benefit in using either the SLP algorithm or LP-SMC algorithm compared to the SMC algorithm. Therefore, when comparing the algorithms in terms of both computational time and design efficiency, the proposed LP-SMC algorithm appears as the preferred choice among the three. Further, when LP-SMC algorithm was considered, the MCMC step was only required within a small number of iterations of the adaptive design process. This provides evidence that the Pareto-smoothed Laplace importance sampling method provides an efficient proposal distribution within our adaptive design algorithm.

The generic approach to adaptive design outlined in Algorithm 1 is often referred to as a greedy/myopic approach to adaptive design. This is because optimal designs are found by only looking one step ahead. Ideally, such a choice would be made by enumerating all possible decisions into the future, then selecting the series of decisions that led to the best outcome. Such an approach is known as backwards induction and has been considered by Müller et al. (2006) in Bayesian adaptive design. Unfortunately, this approach is highly computational and thus is generally limited to a small number of decisions, a small number of outcomes and a small number of potential designs. Indeed, the work of Müller et al. (2006) was limited to a discrete design space with only three possible decisions. Due to such limitations, the pragmatic myopic approach to adaptive design is generally undertaken. However, a recent paper of Huan and Marzouk (Huan and Marzouk 2016) proposed an approximate backwards induction approach for sequential design. By taking account into future decisions when selecting designs sequentially (rather than looking one step ahead), there is potential to obtain more efficient designs. Adopting such an approach within our algorithm is an avenue of research with which we plan to explore into the future.

In our proposed algorithm, Pareto-smoothed Laplace importance sampling is used to approximate the posterior distribution at each iteration of the sequential design. In the first several iterations of the sequential design, the posterior distributions could be multi-modal (see, for example, Fig. 14). In such situations, it is likely that Pareto smoothing will not work as intended. Thus, it may be sensible to run an SMC algorithm in the first few iterations of the adaptive design.

Future development of our algorithm could include extensions to adaptive experiments run in batches (McGree et al. 2016; Prakash and Datta 2013). For such experiments, the potential for variability within and between batches should be taken into account. This can be achieved by fitting a hierarchical random effects model to the data. Such an approach was adopted in McGree et al. (2016) who proposed extensions to the SMC algorithm such that it could be used for estimating hierarchical models. However, the significant computation involved in doing so meant high-performance computing facilities were used to consider motivating examples. Alternatively, a Laplace approximation for hierarchical models could be considered (see Skaug and Fournier 2006; Raudenbush et al. 2000), and this should result in significant computational savings. We plan to explore such an approach in future research.

Another possible extension to our work would be to develop an adaptive design algorithm based on variational Bayesian (VB) methods. Such an approach allows more flexible parametric distributions to be used within importance sampling. In addition, VB methods provide computationally efficient approximations to the posterior distributions and a lower bound on the model evidence for use in Bayesian design (Jaakkola and Jordan 2000; Ormerod and Wand 2010). The error of the VB approximations is generally unknown, and this is potentially a reason that such an approach is rarely considered within a Bayesian experimental design context (Foster et al. 2019). However, importance sampling could be used to correct such an error, and thus, the combination of these two methods could be used to potentially develop a more efficient adaptive design algorithm.

Acknowledgements SGJS was supported by QUTPRA scholarship from the Queensland University of Technology. CCD was supported by an Australian Research Council Discovery Project (DP200102101). JMM was supported by an Australian Research Council Discovery Project (DP200101263). Computational resources and services used in this work were provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

Appendix

See Figs. 14, 15 and 16.

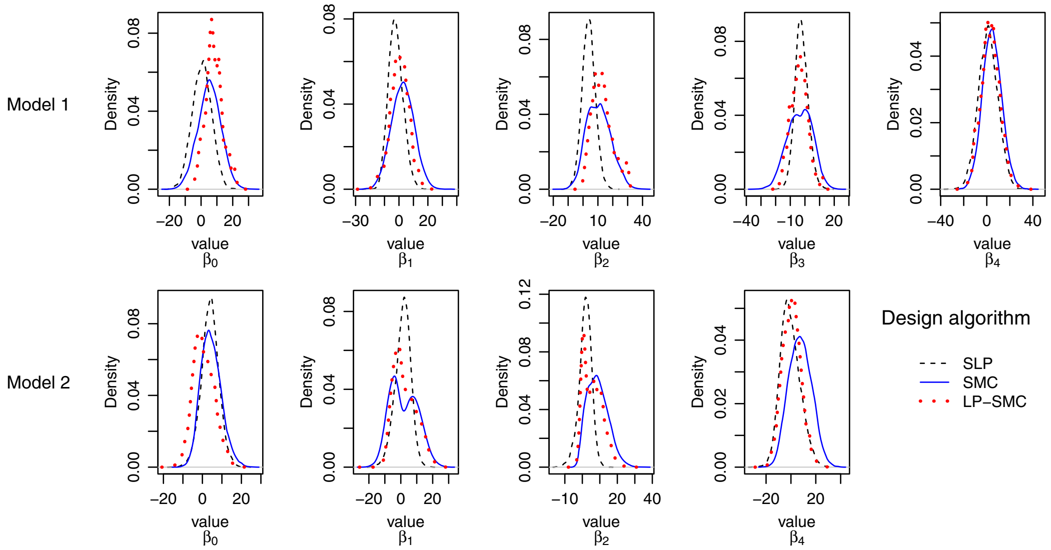
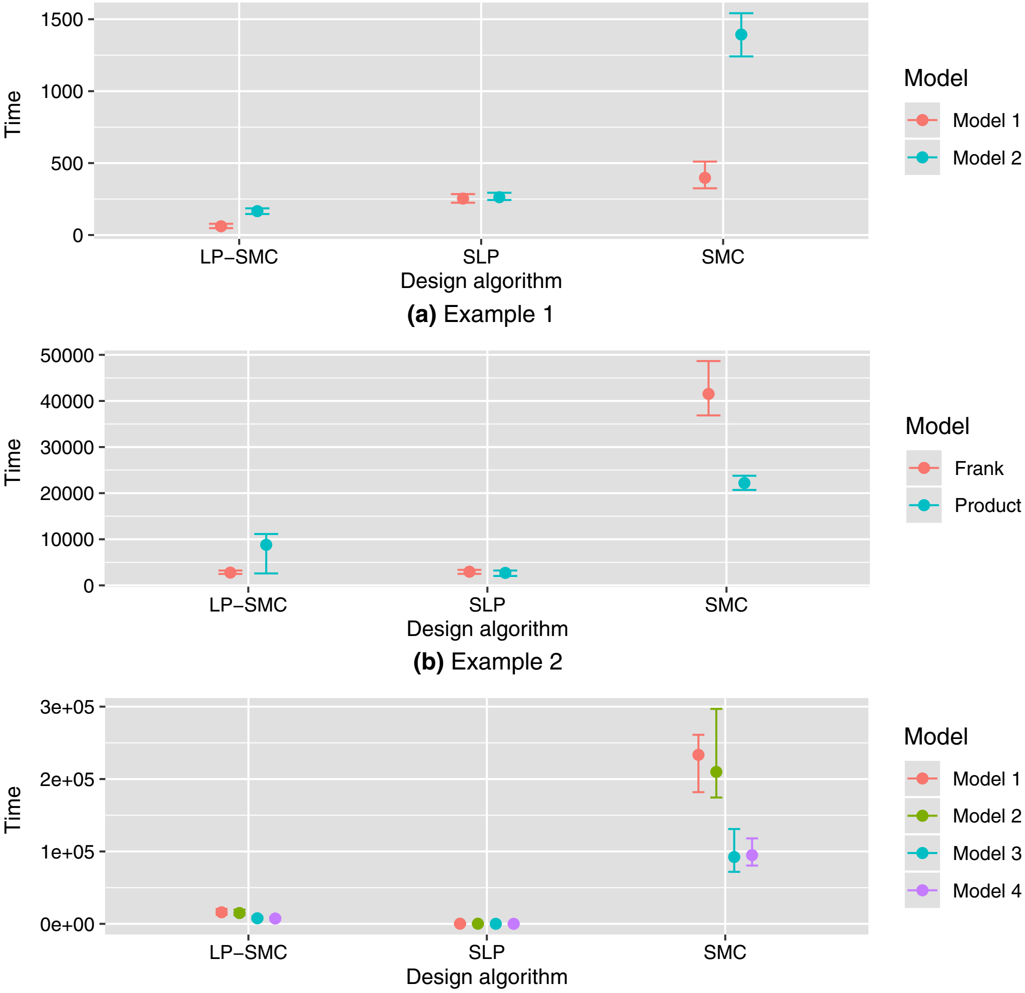


Fig. 14 The posterior distributions obtained at the sixth iteration of the sequential design from each design algorithm based on the same design and data from Example 1



(a) Example 1

(b) Example 2

(c) Example 3

Fig. 15 The 95% credible intervals of the distribution of the cumulative time required to run each algorithm in the simulation studies for each of the three examples

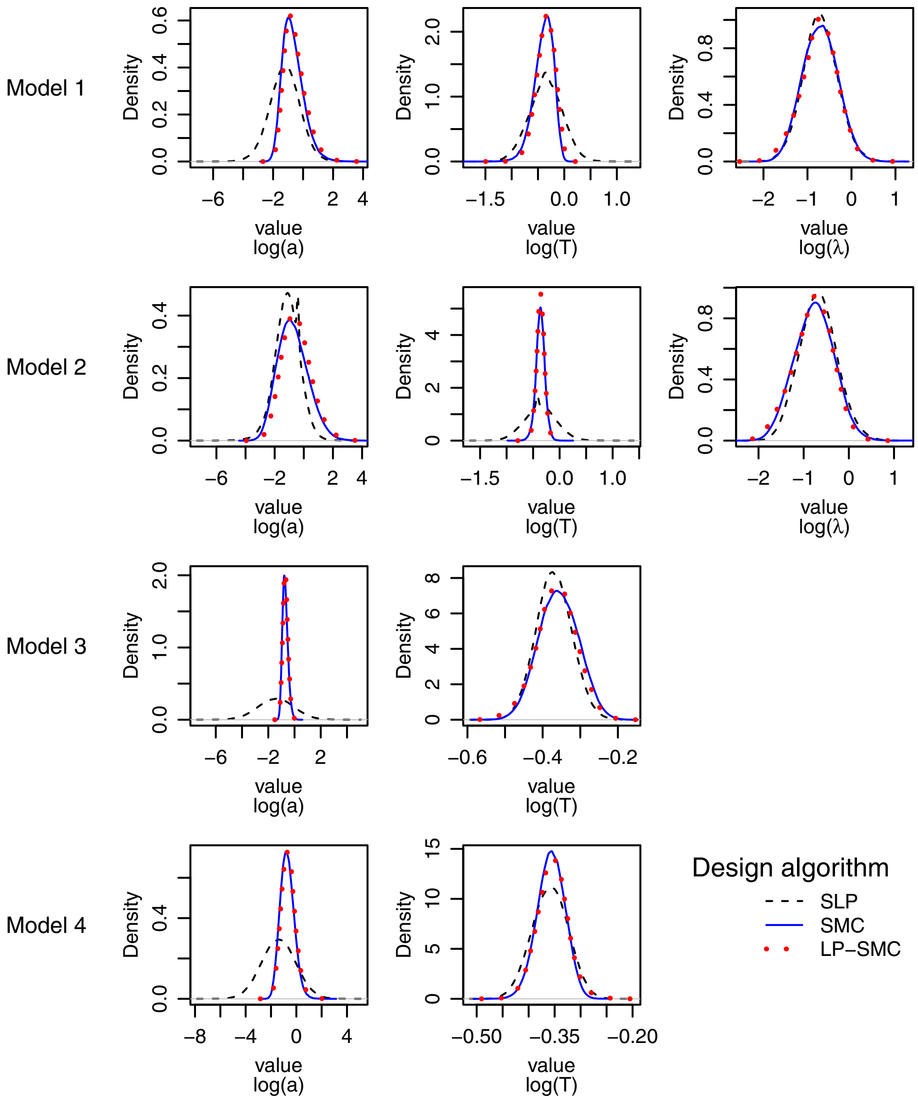


Fig. 16 The posterior distributions of the parameters of the data generating model obtained from each design algorithm using the design and data gathered from a single simulation of the SMC algorithm from Example 3

References

Antognini, A.B., Giovagnoli, A.: Adaptive Designs for Sequential Treatment Allocation. Chapman and Hall/CRC, New York (2015)

Barz, T., López, D.C., Bournazou, M.N.C., Körkel, S., Walter, S.F.: Real-time adaptive input design for the determination of competitive adsorption isotherms in liquid chromatography. Comput. Chem. Eng. **94**, 104–116 (2016)

Beck, J., Dia, B.M., Espath, L.F., Long, Q., Tempone, R.: Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain. Comput. Methods Appl. Mech. Eng. **334**, 523–553 (2018)

Bernardo, J.M., Smith, A.F.M.: Bayesian Theory, 2nd edn. Wiley, Hoboken (2000)

Borth, D.M.: A total entropy criterion for the dual problem of model discrimination and parameter estimation. J. R. Stat. Soc.: Ser. B (Methodol.) **37**(1), 77–87 (1975)

Brinch, C.: Simulated maximum likelihood using tilted importance sampling. Discussion papers, Statistics Norway, Research Department (2008). https://www.ssb.no/a/publikasjoner/pdf/DP/dp540.pdf

Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. **16**(5), 1190–1208 (1995)

Cavagnaro, D.R., Myung, J.I., Pitt, M.A., Kujala, J.V.: Adaptive design optimization: a mutual information-based approach to model discrimination in cognitive science. Neural Comput. **22**(4), 887–905 (2010)

Clark, N.J., Dixon, P.M.: An extended Laplace approximation method for Bayesian inference of self-exciting spatial-temporal models of count data (2017). arXiv:1709.09952 [stat.CO]

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc.: Ser. B (Stat. Methodol.) **68**(3), 411–436 (2006)

Denman, N.G., McGree, J.M., Eccleston, J.A., Duffull, S.B.: Design of experiments for bivariate binary responses modelled by Copula functions. Comput. Stat. Data Anal. **55**(4), 1509–1520 (2011)

Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Stat. Comput. **10**(3), 197–208 (2000)

Dror, H.A., Steinberg, D.M.: Sequential experimental designs for generalized linear models. J. Am. Stat. Assoc. **103**(481), 288–298 (2008)

Drovandi, C.C., Pettitt, A.N.: Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. Biometrics **67**(1), 225–233 (2011)

Drovandi, C.C., McGree, J.M., Pettitt, A.N.: Sequential Monte Carlo for Bayesian sequentially designed experiments for discrete data. Comput. Stat. Data Anal. **57**(1), 320–335 (2013)

Drovandi, C.C., McGree, J.M., Pettitt, A.N.: A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design. J. Comput. Graph. Stat. **23**(1), 3–24 (2014)

Feng, C., Marzouk, Y.M.: A layered multiple importance sampling scheme for focused optimal Bayesian experimental design (2019). arXiv:1903.11187 [stat.CO]

Fenlon, J.S., Faddy, M.J.: Modelling predation in functional response. Ecol. Model. **198**, 154–162 (2006)

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y.W., Rainforth, T., Goodman, N.: Variational optimal experiment design: efficient automation of adaptive experiments (2019). arXiv:1903.05480 [stat.ML]

Holling, C.S.: Some characteristics of simple types of predation and parasitism. Can. Entomol. **91**(7), 385–398 (1959)

Huan, X., Marzouk, Y.M.: Sequential Bayesian optimal experimental design via approximate dynamic programming (2016). arXiv:1604.08320 [stat.ME]

Jaakkola, T.S., Jordan, M.I.: Bayesian parameter estimation via variational methods. Stat. Comput. **10**(1), 25–37 (2000). https://doi.org/10.1023/A:1008932416310

Kuk, A.Y.C.: Laplace importance sampling for generalized linear mixed models. J. Stat. Comput. Simul. **63**(2), 143–158 (1999)

Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)

Lewi, J., Butera, R., Paninski, L.: Sequential optimal design of neurophysiology experiments. Neural Comput. **21**(3), 619–687 (2009)

Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2008)

Long, Q., Scavino, M., Tempone, R., Wang, S.: Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations. Comput. Methods Appl. Mech. Eng. **259**, 24–39 (2013)

Long, Q., Scavino, M., Tempone, R., Wang, S.: A Laplace method for under-determined Bayesian optimal experimental designs. Comput. Methods Appl. Mech. Eng. **285**, 849–876 (2015)

McGree, J.M.: Developments of the total entropy utility function for the dual purpose of model discrimination and parameter estimation in Bayesian design. Comput. Stat. Data Anal. **113**, 207–225 (2017)

McGree, J.M., Drovandi, C.C., Thompson, M., Eccleston, J., Duffull, S., Mengersen, K., Pettitt, A.N., Goggin, T.: Adaptive Bayesian compound designs for dose finding studies. J. Stat. Plan. Inference **142**(6), 1480–1492 (2012)

McGree, J.M., Drovandi, C.C., White, G., Pettitt, A.N.: A pseudo-marginal sequential Monte Carlo algorithm for random effects models in Bayesian sequential design. Stat. Comput. **26**(5), 1121–1136 (2016)

Moffat, H., Hainy, M., Papanikolaou, N.E., Drovandi, C.C.: Sequential experimental design for functional response experiments (2019). arXiv:1907.02179 [stat.AP]

Müller, P., Berry, D.A., Grieve, A.P., Krams, M.: A Bayesian decision-theoretic dose-finding trial. Decis. Anal. **3**(4), 197–207 (2006)

Nelsen, R.B.: An Introduction to Copulas. Springer Series in statistics, 2nd edn. Springer, New York (2006)

Ogden, H.: On the error in Laplace approximations of high-dimensional integrals (2018). arXiv:1808.06341 [math.ST]

Ormerod, J.T., Wand, M.P.: Explaining variational approximations. Am. Stat. **64**(2), 140–153 (2010). https://doi.org/10.1198/tast.2010.09058

Overstall, A.M., Woods, D.C.: Bayesian design of experiments using approximate coordinate exchange. Technometrics **59**(4), 458–470 (2017)

Overstall, A.M., McGree, J.M., Drovandi, C.C.: An approach for finding fully Bayesian optimal designs using normal-based approximations to loss functions. Stat. Comput. **28**(2), 343–358 (2018a)

Overstall, A.M., Woods, D.C., Adamou, M.: acebayes: an R package for Bayesian optimal design of experiments via approximate coordinate exchange (2018b). arXiv:170508096

Palmer, J.L., Müller, P.: Bayesian optimal design in population models for haematologic data. Stat. Med. **17**(14), 1613–1622 (1998)

Prakash, O., Datta, B.: Sequential optimal monitoring network design and iterative spatial estimation of pollutant concentration for identification of unknown groundwater pollution source locations. Environ. Monit. Assess. **185**(7), 5611–5626 (2013)

Raudenbush, S.W., Yang, M.L., Yosef, M.: Maximum likelihood for generalized linear models with nested random effects via high-order, multivariate Laplace approximation. J. Comput. Graph. Stat. **9**(1), 141–157 (2000)

Roy, S., Notz, W.I.: Estimating percentiles in computer experiments: a comparison of sequential-adaptive designs and fixed designs. J. Stat. Theory Pract. **8**(1), 12–29 (2014)

Ryan, K.: Estimating expected information gains for experimental designs with application to the random fatigue-limit model. J. Comput. Graph. Stat. **12**, 585–603 (2003)

Senarathne, S.G.J., Drovandi, C.C., McGree, J.M.: Bayesian sequential design for Copula models. TEST (2019). https://doi.org/10.1007/s11749-019-00661-7

Shun, Z., McCullagh, P.: Laplace approximation of high dimensional integrals. J. R. Stat. Soc.: Ser. B (Methodol.) **57**(4), 749–760 (1995)

Skaug, H.J., Fournier, D.A.: Automatic approximation of the marginal likelihood in non-Gaussian hierarchical models. Comput. Stat. Data Anal. **51**(2), 699–709 (2006)

Stroud, J.R., Muller, P., Rosner, G.L.: Optimal sampling times in population pharmacokinetic studies. J. R. Stat. Soc.: Ser. C (Appl. Stat.) **50**(3), 345–359 (2001)

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., Gabry, J.: Pareto smoothed importance sampling (2017). arXiv:1507.02646 [stat.CO]

Vehtari, A., Gelman, A., Gabry, J., Yao, Y., Bürkner, P.C., Goodrich, B., Piironen, J., Magnusson, M.: loo: efficient leave-one-out cross-validation and WAIC for Bayesian models (2019). https://CRAN.R-project.org//package=loo, R package version 2.1.0

Wakefield, J.: An expected loss approach to the design of dosage regimens via sampling-based methods. J. R. Stat. Soc. Ser. D (Stat.) **43**(1), 13–29 (1994)

Weir, C.J., Spiegelhalter, D.J., Grieve, A.P.: Flexible design and efficient implementation of adaptive dose-finding studies. J. Biopharm. Stat. **17**(6), 1033–1050 (2007)