d-QPSO: A Quantum-Behaved Particle Swarm Technique for Finding D-Optimal Designs for Models with Discrete and Continuous Factors and a Binary Response

Supplementary Materials

In the following sections, we further demonstrate the flexibility and utility of the d-QPSO algorithm for finding different types of optimal exact and approximate designs for GLMs with mixed factors and a binary response. We also show how it can be used to address uncertainty in the model assumptions and discuss the general performance of the algorithm.

S1 d-QPSO Computational Timing and Accuracy

We first report the average CPU run time and the average D-efficiency lower bound attained by the d-QPSO algorithm-generated design for some simple models. The model of interest is $logit(\mu) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i$ and the number of factors, k, ranges from two to six, for all combinations of up to three discrete and three continuous factors. For each combination of factor types we apply the d-QPSO algorithm to construct 500 locally D-optimal approximate designs with $\beta_0, \beta_i \sim U(-3, 3)$. The design space is such that the discrete x_i's $\in \{-1, 1\}$ and the continuous x_i's $\in [-1, 1]$. The tuning parameters we used were 30 particles in each swarm, and the number of swarms was chosen to be equal to the number of continuous factors in the model. We initialized our search among designs with up to $\min\{16, 2^k\}$ support points, or in other words, each candidate design has at most 16 support points. The termination rule was either a maximum of $200 \times k$ iterations or when the generated design attained a D-efficiency lower bound of 98%. Grid searches were used to evaluate

Table S1: Average CPU times (seconds) and D-efficiency lower bounds (elb) for 500 simulated experimental designs generated with the d-QPSO algorithm.

Discrete Factors \ Continuous Factors	1		2		3	
	CPU time	elb	CPU time	elb	CPU time	elb
1	0.01	1.00	0.729	0.99	38.07	0.98
2	2.15	1.00	1.427	0.99	64.44	0.96
3	16.36	0.99	65.81	0.96	127.35	0.84

the sensitivity function of each generated design. These searches are included in the CPU time calculation, which is measured using the "user time" reported by R.

Table S1 displays the average CPU time required by the d-QPSO algorithm to obtain the locally D-optimal approximate design and the average D-efficiency lower bounds (elb) when there are different numbers of discrete and continuous factors in the experiment. Our results show that the d-QPSO algorithm is able to quickly identify a very highly D-efficient design or locally D-optimal approximate design.

S2 Minimally Supported Designs

In our second simulation, we delineate cases when and if a minimally supported locally D-optimal approximate design can be found by the d-QPSO algorithm. This is an interesting issue because some methods can only produce optimal designs with a fixed number of points (see for example, Yang et al. (2011)). Minimally supported optimal designs can be desirable because taking observations at a new point can be expensive.

Consider the model $Y \sim Bern(\mu)$, with $logit(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ where $x_1 \in \{-1, 1\}$ and $x_2 \in [-1, 1]$, and the ranges for the nominal values are $\beta_0 \in \{1, 1.5, 2\}$, $\beta_1 \in [-1.5, 1.5]$ and $\beta_2 \in [-3, 3]$. We employ the d-QPSO algorithm to find locally D-optimal approximate designs. The tuning parameters we used were 2 swarms, 25 particles in each swarm, and we initialized our search among designs with up to 4 support points. The termination rule was either a maximum of 1000 iterations or when the generated design attained a D-efficiency lower bound of 99%. For the simulation, we discretize the parameter space for β_1 and β_2 using a grid with resolution 0.01, meaning that each parameter space is divided into a grid with points uniformly spaced 0.01 apart.

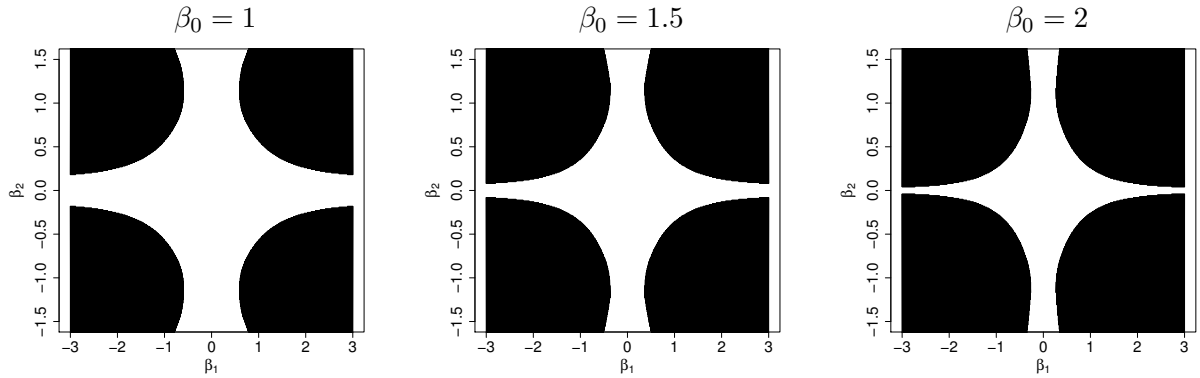

Figure S1: The black areas show the ranges of values for β_1 and β_2 for which a minimally supported locally D-optimal approximate design was found by the d-QPSO algorithm for the two-factor additive model when $\beta_0 = 1$, 1.5, and 2, respectively.

We use d-QPSO and generate designs for all combinations of β_1, β_2, with the intercept β_0 fixed. This results in a total of 180,901 (301 β_1 values $\times$ 601 β_2 values) d-QPSO algorithm-generated locally D-optimal approximate designs for each fixed β_0 setting.

The black curvilinear areas in Figure S1 show parameter values β_1 and β_2 for which the d-QPSO algorithm was able to construct minimally supported designs when $\beta_0 = \{1, 1.5, 2\}$. We observe that as the magnitude of β_0 increases, the region in which a minimally supported design can be constructed also increases. These pictures are similar to the ones obtained theoretically in Figure 2 on page 399 of Yang et al. (2016) and in Figures 1 and 3 of pages 11 and 19 of Yang et al. (2017).

S3 Sensitivity Study

S3.1 Robustness Under Mis-specification of the Link Function

Before a design is implemented, it is important to investigate its robustness properties to model mis-specification. For example, in GLMs with a binary response it is common to choose the logit link, but a prudent researcher should choose a design that reflects the actual goals of the study and has acceptable efficiency if there are violations in the model assumptions. There are several types of such violations. To fix ideas, suppose there is concern whether the link function is correctly specified and we want to know whether the locally D-optimal design found under the assumed link function remains efficient when the true link is another link function. In what follows, we use the

Table S2: Percentiles of the D-efficiencies of the logit link based d-QPSO algorithm-generated locally D-optimal designs relative to the d-QPSO algorithm-generated locally D-optimal approximate designs constructed under the probit, log-log, and complementary log-log link functions.

Percentile \ True Link	Probit	Log-log	C-log-log
0.99	1.0000	1.0000	1.0000
0.95	1.0000	1.0000	0.9900
0.90	1.0000	1.0000	0.9488
0.80	0.9900	0.9737	0.8692
0.70	0.9670	0.9106	0.7925

d-QPSO algorithm to investigate the robustness of locally D-optimal approximate designs found under the logit link when the true link function is probit, log-log, or complementary log-log.

We ran the d-QPSO algorithm using tuning parameters of 2 swarms, 25 particles in each swarm, and we initialized our search among designs with up to 4 support points. The termination rule was either a maximum of 1000 iterations or when the generated design attained a D-efficiency lower bound of 99%. We compare the D-efficiency of the logit link based d-QPSO algorithm-generated design relative to the d-QPSO algorithm-generated designs under the correct link function. In this study we considered the model $Y \sim Bern(\mu)$, with $logit(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ with $x_1 \in \{-1, 1\}$ and $x_2 \in [-1, 1]$, but other models can be used. We take $\beta_0 = 1$ and explored $\beta_1 \in [-1.5, 1.5]$ and $\beta_2 \in [-3, 3]$ over a grid with resolution 0.1; this results in a total of $31 \times 61 = 1891$ individual locally D-optimal approximate designs generated for each link function. We then compare how the d-QPSO algorithm-generated locally D-optimal approximate design from the logit link function performs under various other link functions.

Table S2 provides results of the above simulation, and Figure S2 displays the D-efficiencies. We observe that many of the d-QPSO algorithm-generated designs are fairly robust against model misspecification in the link function. When the true link is the probit or log-log link, the logit-based designs tend to perform very well and less so when the true link function is the complementary log-log. Figure S2 suggests the problematic areas occur when β_1 is near 0 for the log-log link and when β_1 and β_2 are both near their extremes for the complementary log-log link. We note that these results assume $\beta_0 = 1$; for different values of β_0, the d-QPSO algorithm-obtained locally D-optimal approximate designs under an incorrect link function may behave differently.

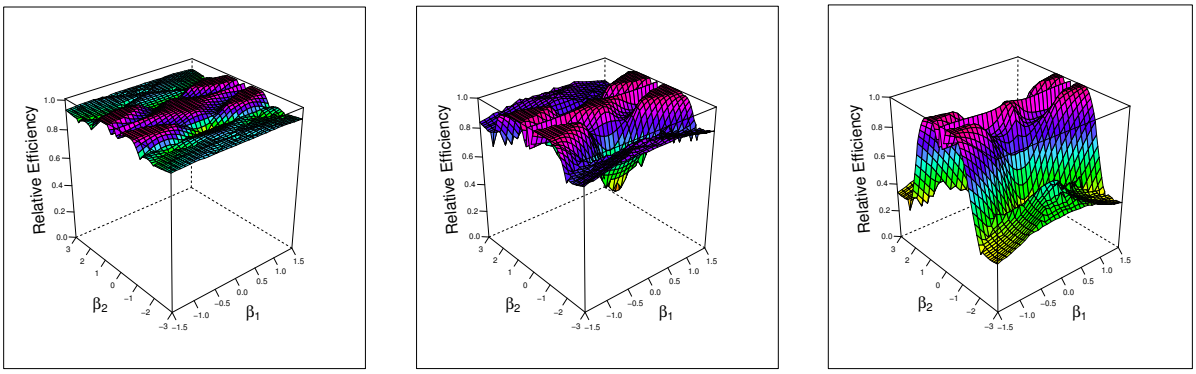

Figure S2: D-efficiencies of the logit link based d-QPSO algorithm-generated locally D-optimal approximate designs relative to the d-QPSO algorithm-generated locally D-optimal approximate designs constructed under the probit (left), log-log (middle), and complementary log-log (right) link functions.

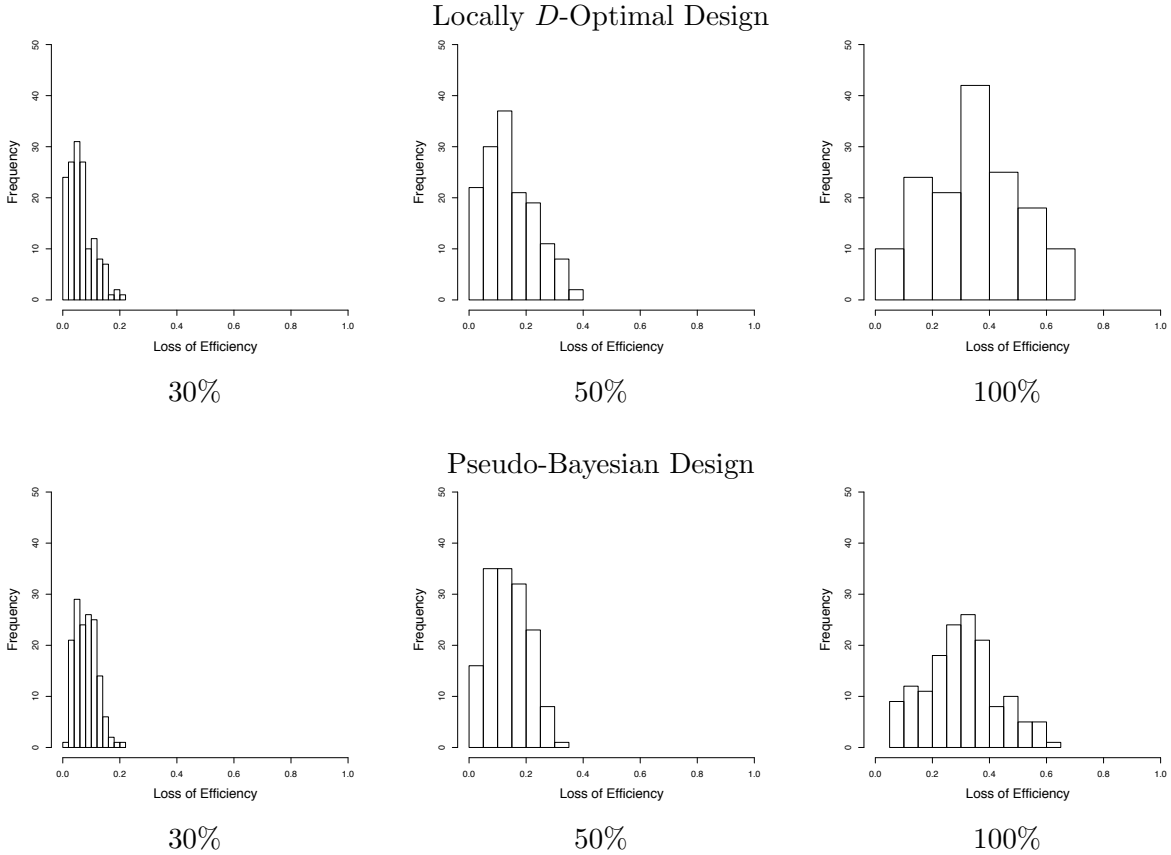

Figure S3: Loss of D-efficiency under 3 different levels of mis-specification: up to 30% of parameter magnitude, 50% of parameter magnitude, and 100% of parameter magnitude, corresponding to the columns in Table S5. Loss of D-efficiency was calculated by taking 1− (D-efficiency of each design to the d-QPSO algorithm-generated locally D-optimal approximate design).

S3.2 Sensitivities of Locally Optimal Designs to Mis-specified Nominal Values

Before a locally D-optimal design is implemented, it is important to investigate if it is robust to mis-specification of the nominal values. When there are multiple parameters the model, the problem becomes complicated since it may not be clear how to vary the nominal values systematically and draw meaningful conclusions. To fix ideas, let us return to the odor removal experiment, and conduct three robustness studies of the d-QPSO algorithm-generated locally D-optimal approximate design in Section 3.1. For the purpose of comparison, we also examine the d-QPSO algorithm-generated pseudo-Bayesian approximate design in Section 4.1. First, we investigate the drop in D-efficiency of the locally D-optimal approximate design when one of the nominal parameters is mis-specified by 10%, 20%, and 30% of its true value. Second, we examine the performance of the design when two parameters are mis-specified using the same setup as before, where all $\binom{6}{2} = 15$ combinations of parameters were considered for mis-specification.

As a third and probably more effective way to assess the effects of mis-specification of the nominal values on the optimal design, we consider cases where the entire nominal parameter vector is mis-specified to some extent. To this end, we perform similar robustness experiments to those carried out by Woods et al. (2006) and Gotwalt et al. (2009) and generate 150 random parameter vectors $\boldsymbol{\theta}_i$, $i = 1, \ldots, 150$ from the prior specification (uniform $\pm$ 100% the magnitude of the nominal values) and compute the locally D-optimal approximate designs $\boldsymbol{\psi}_{\boldsymbol{\theta}_1}, \boldsymbol{\psi}_{\boldsymbol{\theta}_2}, \ldots, \boldsymbol{\psi}_{\boldsymbol{\theta}_{150}}$ using d-QPSO. We evaluate the D-efficiencies of the d-QPSO algorithm-generated locally D-optimal approximate design $\boldsymbol{\psi}_{\boldsymbol{\theta}_0}$ in Section 3.1, and the pseudo-Bayesian design $\boldsymbol{\psi}_{\boldsymbol{B}}$ in Section 4.1, relative to the d-QPSO algorithm-generated locally D-optimal designs $\boldsymbol{\psi}_{\boldsymbol{\theta}_1}, \boldsymbol{\psi}_{\boldsymbol{\theta}_2}, \ldots, \boldsymbol{\psi}_{\boldsymbol{\theta}_{150}}$ as

$$RE_{L,i} = \left(\frac{\left| I_{\boldsymbol{\psi}_{\boldsymbol{\theta}_0}}(\boldsymbol{\theta}_i) \right|}{\left| I_{\boldsymbol{\psi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\theta}_i) \right|} \right)^{1/6} \text{ and } RE_{\boldsymbol{B},i} = \left(\frac{\left| I_{\boldsymbol{\psi}_{\boldsymbol{B}}}(\boldsymbol{\theta}_i) \right|}{\left| I_{\boldsymbol{\psi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\theta}_i) \right|} \right)^{1/6},$$

respectively, for $i = 1, \ldots, 150$. Here, $RE_{L,i}$ evaluates the objective function value of the design constructed under the nominal values, $\boldsymbol{\theta}_0$, at the true parameter vector, $\boldsymbol{\theta}_i$, and compares that value to the objective function value of the d-QPSO algorithm-generated locally D-optimal approximate design for $\boldsymbol{\theta}_i$. Similarly, for the pseudo-Bayesian design, $RE_{\boldsymbol{B},i}$ compares the value of the objective function of the design constructed under the prior vector at the true parameter vector $\boldsymbol{\theta}_i$ with that

Table S3: Mean and median D-efficiencies (RE) of the d-QPSO algorithm-generated locally D-optimal and pseudo-Bayesian approximate designs relative to the d-QPSO algorithm-generated locally D-optimal design when one parameter is mis-specified at a time by $\pm 10\%$, $\pm 20\%$, and $\pm 30\%$ of the magnitude of its nominal value.

Design	$\pm 10\%$		$\pm 20\%$		$\pm 30\%$	
	Mean RE	Median RE	Mean RE	Median RE	Mean RE	Median RE
Locally Optimal	0.9964	0.9989	0.9852	0.9945	0.9690	0.9872
Pseudo Bayesian	0.9618	0.9645	0.9538	0.9620	0.9424	0.9575

Table S4: Mean and median D-efficiencies (RE) of the d-QPSO algorithm-generated locally D-optimal and pseudo-Bayesian approximate designs relative to the the d-QPSO algorithm-generated locally D-optimal approximate designs when two parameters are mis-specified at a time by $\pm 10\%$, $\pm 20\%$, and $\pm 30\%$ of the magnitude of their nominal values (i.e. we randomly generate parameter vectors from uniform priors).

Design	$\pm 10\%$		$\pm 20\%$		$\pm 30\%$	
	Mean RE	Median RE	Mean RE	Median RE	Mean RE	Median RE
Locally Optimal	0.9934	0.9933	0.9745	0.9726	0.9475	0.9427
Pseudo Bayesian	0.9595	0.9617	0.9457	0.9544	0.9264	0.9427

of the d-QPSO algorithm-generated locally D-optimal approximate design for $\boldsymbol{\theta}_i$.

We also sample parameter vectors from two narrower priors ($\pm 30\%$ and $\pm 50\%$) and compare the loss in D-efficiency following the same procedure. We note that the uniform prior of nominal values $\pm 100\%$ was used to construct the pseudo-Bayesian design under consideration and the robustness was evaluated by constructing $\boldsymbol{\psi}_{\boldsymbol{\theta}_1}, \ldots, \boldsymbol{\psi}_{\boldsymbol{\theta}_{150}}$ for each of $\pm 30\%$, $\pm 50\%$, and $\pm 100\%$ mis-specification. Thus the two narrower mis-specifications correspond to situations where the experimenter took a very conservative approach even when the true parameter values were actually fairly close to the supposed nominal values.

Tables S3 and S4 list, respectively, results of one- and two-parameter mis-specification in the nominal values, and Table S5 shows the mean and median D-efficiencies based on the 150 simulated values for the full vector of model parameters. Figure S3 provides histograms of the corresponding loss in D-efficiency. For the one- and two-parameter mis-specification simulations we observe that the d-QPSO algorithm-generated locally D-optimal and pseudo-Bayesian designs are both very efficient, even with 30% mis-specification of the magnitude of the nominal values. For the full parameter vector mis-specification simulations, Table S5 and Figure S3 show that the d-QPSO algorithm-generated designs also appear to perform quite well.

Table S5: Mean and median D-efficiencies (RE) of the d-QPSO algorithm-generated locally D-optimal and pseudo-Bayesian approximate designs relative to the the d-QPSO algorithm-generated locally D-optimal approximate designs when all parameters are mis-specified and sampled from an independent uniform prior over ±30%, ±50%, ±100% of the parameter magnitudes, $\boldsymbol{\mu}$.

Design	$\boldsymbol{\beta} \sim U(0.7\boldsymbol{\mu}, 1.3\boldsymbol{\mu})$		$\boldsymbol{\beta} \sim U(0.5\boldsymbol{\mu}, 1.5\boldsymbol{\mu})$		$\boldsymbol{\beta} \sim U(0\boldsymbol{\mu}, 2\boldsymbol{\mu})$	
	Mean RE	Median RE	Mean RE	Median RE	Mean RE	Median RE
Locally Optimal	0.9359	0.9446	0.8559	0.8690	0.6538	0.6443
Pseudo Bayesian	0.9182	0.9204	0.8612	0.8624	0.6980	0.6995

S4 Locally D-optimal Exact Designs

In Section 4, we showed that the d-QPSO algorithm could be used to find a pseudo-Bayesian exact design. Here we further demonstrate that the algorithm can also find locally D-optimal exact designs. We apply the d-QPSO algorithm to generate locally D-optimal exact designs for the odor removal experiment with nominal values as $\boldsymbol{\beta} = (-1, 2, 0.5, -1, -0.25, 0.13)^T$. We find locally D-optimal exact designs when the total number of observations, N, is specified. We generate locally D-optimal exact designs for $N = 6, 10, 25, 50$ and 100 and note that (i) the case $N = 6$ corresponds to finding a minimally supported locally D-optimal exact design and (ii) when $N = 100$ (which is large), the d-QPSO algorithm-generated locally D-optimal exact design should be similar to the d-QPSO algorithm-generated locally D-optimal approximate design in Section 3.1.

The first three locally D-optimal exact designs were found by the d-QPSO algorithm using 10 swarms, 20 particles, and a termination rule of 5000 iterations. The last two designs were found using 15 swarms instead of 10. For each problem, we ran the d-QPSO algorithm four times to ensure the objective function value was about the same. Table S6 reports the number of support points in the d-QPSO algorithm-generated exact designs, along with their objective function values and the CPU times required to find them. Clearly, when $N > 25$ the objective function value of the exact designs becomes close to that of the d-QPSO algorithm-generated locally D-optimal approximate design, which has a value of 0.3519.

Table S7 displays the d-QPSO algorithm-generated exact designs for $N = 6, 10$, and 15 and shows how their support points are distributed. Table S8 compares the d-QPSO algorithm-generated locally D-optimal exact design for $N = 100$ (right) with the d-QPSO algorithm-generated locally D-optimal approximate design (left). The two designs are aligned by support points, such that each support point on the left is very similar to the one on the right. The three points listed at

Table S6: Properties of the d-QPSO algorithm-generated exact designs for the odor removal experiment with nominal values $\boldsymbol{\beta} = (-1, 2, 0.5, -1, -0.25, 0.13)^T$.

N	Number of Support Points	CPU Time	Objective Function Value
6	6	14.368	0.3368
10	10	15.616	0.3438
25	17	30.223	0.3504
50	18	89.696	0.3510
100	18	337.073	0.3513

Table S7: The d-QPSO algorithm-generated locally D-optimal exact designs for the odor removal experiment with nominal values $\boldsymbol{\beta} = (-1, 2, 0.5, -1, -0.25, 0.13)^T$ for $N = 6, 10$, and 25.

Algae	Scav.	Resin	Comp.	Temp.	N	Algae	Scav.	Resin	Comp.	Temp.	N	Algae	Scav.	Resin	Comp.	Temp.	N
−1	−1	−1	1	29.74	1	−1	−1	−1	1	13.25	1	−1	−1	−1	−1	7.60	1
−1	−1	1	1	27.95	1	−1	−1	−1	1	30.50	1	−1	−1	−1	−1	26.92	2
−1	1	−1	−1	5.00	1	−1	−1	1	−1	34.39	1	−1	−1	−1	1	28.48	2
−1	1	1	−1	33.59	1	−1	1	−1	−1	17.06	1	−1	−1	1	−1	24.95	1
1	−1	1	−1	5.00	1	−1	1	−1	−1	5.00	1	−1	−1	1	−1	35.00	1
1	1	1	1	5.00	1	−1	1	1	1	18.49	1	−1	−1	1	1	29.66	1
						−1	1	1	1	35.00	1	−1	−1	1	1	35.00	1
						1	−1	1	−1	5.00	1	−1	1	−1	−1	5.00	2
						1	−1	1	1	5.00	1	−1	1	−1	1	5.13	2
						1	1	1	1	5.00	1	−1	1	1	−1	19.24	1
												−1	1	1	−1	33.74	1
												−1	1	1	1	35.00	2
												1	−1	−1	1	5.00	1
												1	−1	1	−1	5.00	2
												1	−1	1	1	5.00	2
												1	1	1	−1	5.00	1
												1	1	1	1	5.00	2

Table S8: The d-QPSO algorithm-generated D-optimal exact design (right) for the odor removal experiment with nominal values $\boldsymbol{\beta} = (-1, 2, 0.5, -1, -0.25, 0.13)^T$ with $N = 100$ experimental units and the corresponding d-QPSO algrorithm-generated locally D-optimal approximate design (left).

Algae	Scav.	Resin	Comp.	Temp.	p_i (%)	Algae	Scav.	Resin	Comp.	Temp.	N
−1	−1	−1	−1	9.04	3.70	−1	−1	−1	−1	8.56	4
−1	−1	−1	−1	25.79	4.30	−1	−1	−1	−1	26.25	6
−1	−1	−1	1	29.71	10.17	−1	−1	−1	1	29.95	9
−1	−1	1	−1	35.00	4.73	−1	−1	1	−1	35.00	5
−1	−1	1	1	29.58	11.59	−1	−1	1	1	30.55	8
−1	1	−1	−1	5.00	9.75	−1	1	−1	−1	5.00	9
−1	1	−1	1	5.21	7.86	−1	1	−1	1	5.36	8
−1	1	1	−1	16.89	2.20	−1	1	1	−1	16.73	2
−1	1	1	−1	33.37	8.80	−1	1	1	−1	32.45	6
−1	1	1	1	35.00	6.10	−1	1	1	1	35.00	8
1	−1	−1	1	5.00	5.11	1	−1	−1	1	5.00	5
1	−1	1	−1	5.00	10.75	1	−1	1	−1	5.00	8
1	−1	1	1	5.00	5.23	1	−1	1	1	5.00	8
1	1	1	1	5.00	9.71	1	1	1	1	5.00	7
						−1	−1	−1	1	12.89	1
						−1	−1	1	−1	25.43	2
						1	1	1	−1	5.00	4

the bottom of the exact design have no similar points in the approximate design. The D-efficiency of the exact design relative to the approximate design is 99.8%. Clearly the exact design found by the d-QPSO algorithm is both highly efficient and very similar to the approximate design found in Section 3.1, which is what we expect when N is large.

S5 Optimal Designs on an Irregular Design Space

The bulk of the D-optimal designs reported in the literature are on prototype design spaces. For example, when factors are continuous the default design space is usually the unit cubiod, or, for mixture experiments, the design space is the regular simplex. In practice, some studies have irregularly shaped design spaces, and this is likely to pose additional difficulties for finding an analytical description of the D-optimal design. Such design problems seem to have not been well studied in the literature even though they appear in real problems. In this subsection we show that the d-QPSO algorithm is flexible and can be directly modified to find a locally D-optimal approximate design on an irregularly-shaped design space.

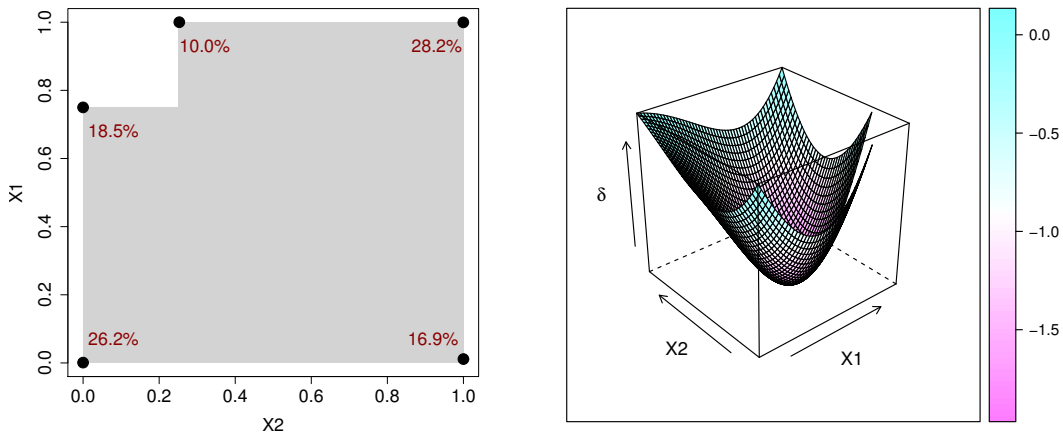

Figure S4: The d-QPSO algorithm-generated locally D-optimal approximate design on an irregular design space with a corner cut off from a box (left) and its sensitivity function (right) with nominal values $\boldsymbol{\beta} = (1.0, -1.7, 1.3)^T$.

To fix ideas, consider a design space which is box-shaped with a corner removed. We consider the model $Y \sim Bern(\mu)$ with $logit(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$, where Y takes values 1 or 0, and $x_1, x_2 \in [-1, 1]$. We remove the upper left hand corner of this design space by adding the constraint that we cannot have both X_1 above 0.75 and X_2 below 0.25. The nominal parameter vector is $\boldsymbol{\beta} = (1.0, -1.7, 1.3)^T$. We run the d-QPSO algorithm using tuning parameters of 2 swarms, 25 particles in each swarm, and we initialized our search among designs with up to 6 support points. The termination rule was either a maximum of 1000 iterations or when the generated design attained a D-efficiency lower bound of 99%. Figure S4 shows the d-QPSO algorithm-generated locally D-optimal approximate design and the sensitivity plot of the design in the same figure confirms its local D-optimality.

S6 The d-QPSO Algorithm for Finding the Optimal Designs for the Odor Removal Experiment

The C++ code that we provide is the d-QPSO algorithm for finding the D-optimal designs for the odor removal experiment. The code can generate locally D-optimal exact and approximate designs, and also pseudo-Bayesian designs.

References

Gotwalt, C. M., Jones, B. A., and Steinberg, D. M. (2009). Fast computation of designs robust to parameter uncertainty for nonlinear settings. *Technometrics*, 51(1):88–95.

Woods, D. C., Lewis, S. M., Eccleston, J. A., and Russell, K. G. (2006). Designs for generalized linear models with several variables and model uncertainty. *Technometrics*, 48(2):284–292.

Yang, J., Mandal, A., and Majumdar, D. (2016). Optimal designs for 2^k factorial experiments with binary response. *Statistica Sinica*, 26(1):385–411.

Yang, J., Tong, L., and Mandal, A. (2017). D-optimal designs with ordered categorical data. *Statistica Sinica*, 27(4):1879–1902.

Yang, M., Zhang, B., and Huang, S. (2011). Optimal designs for generalized linear models with multiple design variables. *Statistica Sinica*, 21(3):1415–1430.