Bayesian Adaptive Estimation of Psychometric Slope and Threshold with Differential Evolution

Hairong Gu, Jay I. Myung, Mark A. Pitt, and Zhong-Lin Lu
{Gu.124, Myung.1, Pitt.2, Lu.535}@Osu.Edu
Department of Psychology, Ohio State University
1835 Neil Avenue, Columbus, OH 43210 USA

Abstract

The adaptive experimentation methodology has been adopted in visual psychophysical modeling in the pursuit of efficiency in experimental time and cost. The standard scheme only optimizes one design in each experimental stage, although simultaneous optimization of multiple designs per stage can be beneficial, but difficult to implement because of a surge in computation. In this study, we incorporated the adaptive experimentation methodology under a Bayesian framework with differential evolution (DE), an algorithm specialized in multi-dimensional optimization problems to explore the multiple-designs-per-stage approach. By taking advantage of parallel computing, DE is computationally fast. The results showed that the multiple-designs-per-stage scheme resulted in a more stable estimation in the early stages of the parameter estimation.

Keywords: Visual psychophysics, Bayesian inference, adaptive estimation, evolutionary computing

Not All Designs are Equally Informative

Experimental design is a critical step in carrying out effective experiments. Traditionally, the practice of experimental design is guided by heuristic norms, using a one-shot design, chosen at the outset, throughout the course of the experiment. Although this approach may be adequate in some scientific quests, its shortcomings are obvious. First, not all experimental designs are equally informative. The traditional approach does not guarantee that the design, including the number of treatments, the values of treatments, and the number of participants in each treatment, is an optimal choice. A non-optimal design may contribute little to the goal of the experiment. Further, the most informative designs may change as the experiment progresses with more responses being observed. Thus, a one-shot design ignores utilizing what can be learned during the course of an experiment.

Second, the traditional experimental design method typically relies on increasing the number of participants or the number of measurements to increase the power of statistical inference. Obviously, this increases the experimental cost, which would matter for experiments that use expensive technology such as fMRI, or research whose target population is difficult to recruit (children, senior citizens, mentally disordered).

Third, the traditional methods of experimental design center on randomization, reduction of variation, blocking etc., with the purpose of revealing the group or treatment effects while ignoring the individual variation. However, more and more recognition has been given to the importance of individual differences. For example, in drug development, it is important to know how different people react differently to the same drug to guide the prescription. Thus, experimental designs should not be identical for every participant.

To illustrate how experimental designs can be unequally informative, suppose that a researcher is interested in studying how the rate of detection changes with the brightness of a stimulus. A psychometric function is used to describe the probability p of detecting a stimulus of certain brightness x. A simplified example assumes a sigmoid function $p = 1/(1 + \exp(-x + t))$, where x is the design variable representing the brightness and t is the parameter, *threshold*, a characteristic associated with a particular individual, reflected in the shift of the model in the design dimension. Suppose that there are only 5 possible values of t. The corresponding predictions are depicted as the five lines in Figure 1. The red line represents a particular subject's true t value and the other four blue lines are from the wrong t values. The researcher conducts an experiment to estimate the threshold value of that subject by presenting two designs with intensity D1 and D2. Visualization of the model suggests that D1 is a good design because the predictions from the five t values are very differentiable so that the observation can be informative of the true t value. On the other hand, D2 would be a bad design because the prediction differences are so small that little information about the exact shift of the true model is given.

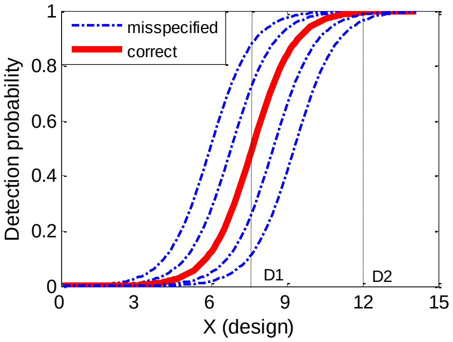


Figure 1: A sample psychometric function with 5 possible parameter values (see text) with the true value indicated by the red line and the wrong values by the blue lines. A good design D1 offers the most discriminability, whereas D2 is a bad design for a lack of differentiability in prediction.

Adaptive Experimentation

In practice, we do not possess full knowledge of the approximate values of good designs because the model can be quite complex and the range of parameters can be much larger. In addition, an experiment usually contains multiple trials, so the best designs at the beginning of an experiment may be different from those at the later trials of the experiment. Therefore, an efficient experimentation should adaptively identify the best design for the current trial based on the responses already collected from the participant. Facing these challenges for a better experimental design regime, a statistical methodology, dubbed *adaptive design optimization* (ADO, Cavagnaro et al., 2010; Myung et al., 2012) under a Bayesian framework has been developed to meet these needs.

The general framework of ADO is illustrated in Figure 2. The traditional experimentation starts from a particular experimental design, with which data are collected, and it stops at cognitive modeling where data are fit to a proposed model to make statistical inferences. In contrast, in ADO, the inference from cognitive modeling continues to influence the choice the designs for the next experimental stage. To put it in another way, the whole experiment is divided into multiple *stages*, and in each stage, the design is based on what is learned from the data collected in the previous stages. By doing that, every selected design is the most imminently useful one for the immediate trial. As such, ADO is efficient in a way that it reduces the time, cost of experiments and the number of participants without sacrificing the quality of the statistical inferences.

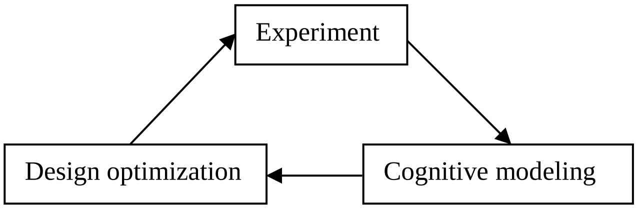


Figure 2: Schematic illustration of ADO paradigm.

There are other desirable features of ADO that make it more attractive to the traditional experimental methods. It is found that bad designs not only increase the cost of experiments, but also deteriorate the quality of data so as to hurt the final inference. ADO adopts an information theoretic computational algorithm to ensure the quality of the selected designs so that the risk of having bad designs is minimized. Additionally, ADO is able to reveal individual differences in response strategy or characteristics because the designs are tailored based on the subject's responses in each experiment. Classification of participants can also be done after individuals' properties are estimated.

Because of its efficiency and versatility, ADO has found its usage in various disciplines. It has been used for designing electrophysiology experiments in neuroscience (Lewi et al., 2008), drug dosage assignment in clinical drug development (Miller, et al. 2007), etc. In psychology, it has been implemented in the discrimination of retention models in simulations (Cavagnaro et al., 2010) and human experiments (Cavagnaro et al., 2011).

A promising application of ADO is in psychophysical experiments with potential clinical applications that put high stake on the reliability of the results and usually have tight time restraint on the experiments. In this area, the previous studies have only optimized one design in each experimental stage. The difficulty of exploring a different scheme, multiple designs per stage, lies in a lack of a smarter algorithm and the increase in computation.

In this paper, we explore ways to improve upon the current efficiency of ADO by implementing the multiple-designs-per-stage scheme that is solved with an evolutionary computation algorithm known as differential evolution (DE). In what follows, we begin with a brief introduction of the ADO methodology. We will then review past studies in adaptive experimentation of visual psychophysics, followed by a discussion of the motivation and application of the multiple-designs-per-stage scheme and DE. Finally, we present and discuss results from ADO simulations.

How ADO Works

In this section, we provide some technical details of ADO. Readers who prefer to skip technicalities may bypass this section. Figure 3 is a schematic illustration of the steps involved in ADO. First, the application of ADO requires that the model should be formulated as a statistical model defined as a parametric family of probability distributions, *p(y|θ, d)*'s, which specifies the probability of observing an experimental outcome *y* given a parameter value *θ* and a design *d.* As mentioned before, ADO is a circulating process going through design optimization (DO), experiments and cognitive modeling. In each round, the process starts with the assumed or learnt probability distribution of the parameters, the prior distribution *p(θ).* Next, in the step of DO, the optimal design *d** is selected from a design set *D* by the principle of maximum utility.

In DO, a utility function *U(d)* is pre-defined to quantify the usefulness of a design *d* ∈ *D* for the purpose of the experiment. For parameter estimation, the utility *U(d)* of each design *d* is the expectation of the local utility *u(d, θ , y)* taken over the parameter space and the outcome's sample space, formally written as

$$\begin{aligned} d^* &= \underset{d\in D}{\operatorname{argmax}}(U(d)) \\ &= \underset{d\in D}{\operatorname{argmax}} \iint_{y,\theta} u(d,\theta,y)\,p(y\mid\theta,d)\,p(\theta)\,dy\,d\theta\,, \end{aligned} \tag{1}$$

where *u(d, θ , y)* is defined on a set of particular design *d*, parameter value *θ* and observation *y.* The goal of parameter estimation is to obtain accurate estimation of the true parameter values with the smallest number of experimental trials. Functionally, an appropriate utility quantifies the usefulness of designs in reducing the variation of the parameter estimates. Or in the language of information theory, a utility amounts to the information gain or the uncertainty reduction of the unknown parameters after observations are collected. One formulation of utility that directly quantifies the information gain of the parameter Θ with the observation Y is Mutual Information $I(Y_d;\Theta)$. According to the property of mutual information, the utility $U(d)$ can be written as

$$U(d) = I(\mathrm{Y}_d;\Theta) \\ = \iint (\log \frac{P(\mathrm{y}_d \mid \theta)}{P(y_d)}) p(y_d \mid \theta) p(\theta) d\theta dy_d,$$

In which $\log\frac{P(y_d|\theta)}{P(y_d)}$ corresponds to the local utility $u(d, \theta, y)$ in Equation (1).

Two general methods have been used in ADO to solve the multiple integral problem of Equation (1), grid search and sequential Monte Carlo (SMC). In grid search, the design space is discretized and grids are the fixed designs on the space. To calculate $U(d)$, one way is to discretize the parameter space also and just replace the integral with summation. Or we can draw a large sample of (θ, y) from the model's prior and sampling distribution, and then calculate Equation (1) by Monte Carlo approximation. On the other hand, in SMC, solving ADO is recasted as a probability density simulation problem. The utility function is extended to a joint distribution with parameters, observations and designs. By adopting Metropolis-Hasting algorithm and simulated annealing procedure, the marginal distribution of d can be obtained. In this paper, we will present a third method, differential evolution (DE) (Storn & Price, 1997) as an alternative that is specialized in multi-dimensional optimization problems.

After DO, the optimal design d_s for the current stage will be presented to the participant. The responses until the current stage will be used to update the knowledge of the parameters. Mathematically, we calculate the posterior probability distribution of the parameters by Bayes' rule, $P_{g+1}(\theta|y,d) = \frac{p(y|d,\theta)p_g(\theta)}{p(y|d)}$. Then the posterior distribution of the parameters of stage g is treated as the prior distribution of the next stage $g+1$. And the ADO process continues.

Adaptive Estimation of Psychometric Function

In visual psychophysics, a major interest is to study the relationship between the intensity of visual stimuli and their perception. This relationship is usually modeled by a psychometric function with two parameters, threshold and slope. Accurate estimation of the parameter values on individual level not only provides knowledge of the underlying psychophysical process, but also assists in the diagnosis and classification (Lesmes et al., 2010). A major, practical challenge is that a large number of experimental trials is often needed to accurately estimate the parameters with the finding that different design schemes of fixed patterns produce varying accuracy, precision of parameter estimation and model fit (Wichmann & Hill, 2001).

Addressing this issue, a variety of adaptive experimental methods have been proposed for efficient parameter estimation while the design dimension was restricted to be one. ADO, as a more general optimization algorithm, is able to handle large scale, non-linear models with multiple design variables. Next, within the framework of ADO, the Ψ method (Kontsevish & Tyler, 1999) was developed that can easily be generalized to incorporate more than one stimulus. It has been applied to such research as diagnosis of visual deficit (Lesmes et al., 2010).

Multiple-designs-per-stage Scheme

All the methods mentioned above assume that there is just one design to be optimized and one response to be collected in each adaptive estimation stage. It is worthwhile to explore if there is any benefit when more than one design is optimized simultaneously and executed in each stage, by which d in Equation (1) becomes a vector. Intuitively, a multiple-designs-per-stage approach can be beneficial because multiple responses are collected jointly in one stage, and according to the information theory, the joint entropy or information from a set of random variables is more than or equal to the sum of entropy from individual variables. Therefore, we hypothesize that if multiple responses are collected in one stage, the relationship or synergy of the responses can benefit the modeling process more than the case when the responses are collected one by one.

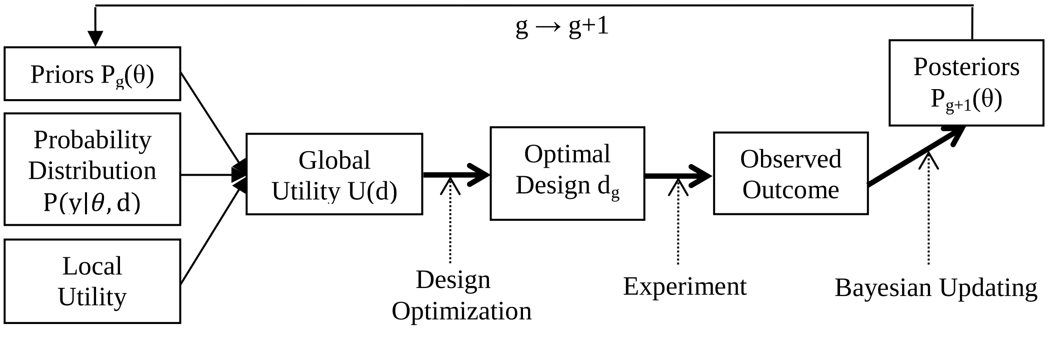


Figure 3: Schematic illustration of the steps involved in adaptive design optimization (ADO).

One computational challenge in the application and implementation of multiple-designs-per-stage scheme is the *curse of dimensionality*. Most published studies on parameter estimation with psychometric functions used brute-force grid search, which is to fix a certain number of design points on the design space. Because the dimension of the design space increases with the number of designs per stage, the quantity of grids need to enlarge exponentially to keep a certain resolution, which causes a waste of computing resource because most of the grids are far from the best design and not worth being computed in each stage. As such, it begs for a different algorithm that suits multi-dimensional optimization problems in an accurate and efficient way.

Differential Evolution Search

DE is an evolutionary computation algorithm to optimize nonlinear and non-differentiable continuous functions by keeping track of, iteratively evolving and updating multiple particles. A brief explanation of the algorithm is as follows. To search the global maximum of a D-dimensional space, it keeps track of *NP* D-dimensional vectors $x_{i,G}$ $(i = 1,2,\dots,NP)$, where *NP* is the number of particles and *G* the generation index. At the beginning, the vectors can be randomly selected. Then for each target vector x_i, a mutant vector $v_{i,G+1}$ for the next stage is generated by $v_{i,G+1} = x_{r1,G} + F \times \left(x_{r2,G} - x_{r3,G}\right)$ where *r1*, *r2* and *r3* are randomly chosen integers from 1 to NP except *i*, and F is a constant factor controlling the contribution of the difference of the two randomly chosen vectors. The next step, crossover, creates a trial vector for each target vector with each element either from the mutant vector v_i or the target vector x_i. Then the cost function values of both the target vector x_i and the mutant vector v_i are computed. If the mutant vector v_i yields a smaller cost, the target vector is set to v_i. Otherwise, the target vector is retained from the last generation. DE is illustrated in Figure 4 with a simple toy example in which DE was used to search the global maximum of a bimodal distribution.

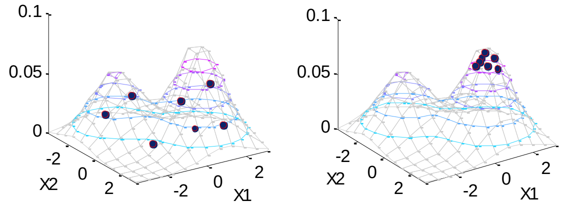


Figure 4: Illustration of DE algorithm searching for the global maximum of a 2-dimensional bimodal distribution. Initially (left), the particles are randomly selected. At 30th generation (right), they converged to the larger mode.

DE is a natural approach to our problem of optimizing multiple designs per stage simultaneously. Because different particles can be processed independently in one stage, DE can benefit from parallel computing.

GPU-based Parallel Computing

Although ADO retains the quality of the data with fewer trials, the heavy computation of ADO is still an issue to reckon with, especially in real-time experiments. One solution to speed up the computation lies in parallel computing. Traditionally, computer instructions are stored and processed by a central processing unit (CPU), and executed in a serial manner. On the other hand, parallel computing employs multiple cores on a single chip to perform many independent numerical operations simultaneously. Graphic processing units (GPUs) were originally dedicated to processing graphics. However, in recent years, GPUs are being increasingly popular as a general-purpose parallel computing tool in image processing, data mining, and machine learning.

In our previous work, we have implemented GPU computing to accelerate ADO computing. Compared with CPU-based ADO, GPU-based ADO is around 100 times faster, which substantiates the feasibility of using GPU computing to accelerate the computational speed of ADO computing (Gu, 2012). Given that the DE algorithm is intrinsically parallelizable, GPU computing can be beneficial for accelerating the computation.

In the present work, we implemented DE on graphic processing units (GPUs) to speed up the ADO computation.

Simulations

ADO-based parameter estimation of the psychophysical model in Kontsevich and Tyler (1999) was simulated with artificial data under the assumption that the data are from a stationary process with no variation of lapses or learning. The data-generating model was defined in the following equations

$$Y \sim Binomial(1, \Psi(x));$$

$$\Psi(x) = \Phi(r(x)/\sqrt{2}; \mu = 0, \sigma = 1);$$

$$r(x) = 10^{\wedge}(10^{s}(x-t)),$$

in which $\Phi(m;\ \mu,\sigma) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{m}\exp(-\frac{(c-\mu)^2}{2\sigma^2})dc$; Y represents the experimental observation; *x*, *t* and *s* are the design variable and the parameters, threshold and slope, transformed in log decimal scale. The range of *x*, *t* and *s* are set to be (0, 3), (0, 3) and ($\log_{10}0.7$, $\log_{10}7$), respectively. The prior distributions of *t* and *s* are both uniform. In the simulation, the true values for *t* and *s* are set to be 1.5 and $\log_{10}3.5$ or approximately 0.544.

Multiple designs are optimized at the same time in one stage by DE algorithm. Computationally, DE is used to search for the global maximum of the defined utility function. For a two-alternative forced choice (2AFC) problem, the response *y* is either 0 or 1. So the utility function of an *n*-dimensional space can be written as

$$U(\tilde{d}) = \sum_{\theta}\sum_{y_i=0,1} u(d_1 \ldots d_n, y_1 \ldots y_n, \theta) P(y_1 \ldots y_n \mid \theta) P(\theta),$$

in which the parameter space θ is also discretized so that the integral in Equation (1) becomes a summation. The local utility $u(d_1...d_n,\ y_1...y_n,\ \theta)$ is in the form of mutual information $\log\frac{P(y_1...y_n|\theta)}{P(y_1...y_n)}$.

First the two-designs-per-stage scheme was implemented. Five two-dimensional particles were generated and shown to be enough for the convergence, which was evaluated by the closeness of the particles at the last generation. Until the 50th generation, the 5 particles are identical up to the second decimal number, indicating that 50 generations are enough for DE to locate the maximum of the utility space. The algorithm was coded in parallel computing with a single GPU card, Tesla C2050 by Nvidia, which contains 448 CUDA cores. A third party library in C++, Arrayfire, is called to access the GPU computing function.

One experiment contains a total of 150 stages or 300 trials. To visualize the effect of parameter estimation, the model predictions based on the prior distribution and the posterior distribution at the last stage is shown in Figure 5. On the left, the model prediction is based on the initial uniform distribution of the two parameters. On the right, the prediction is based on the posterior distribution of the 150th stage of the two parameters. Compared to the initial stage, the range of the likely outcome of the model is much narrowed and concentrated, indicating the convergence of the estimation.

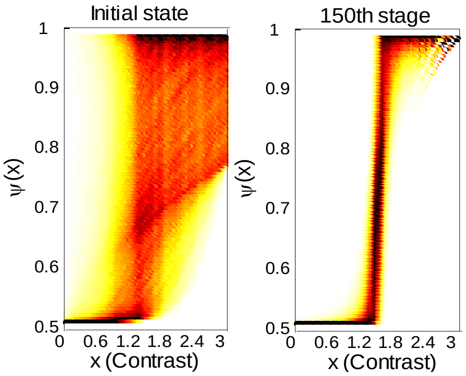


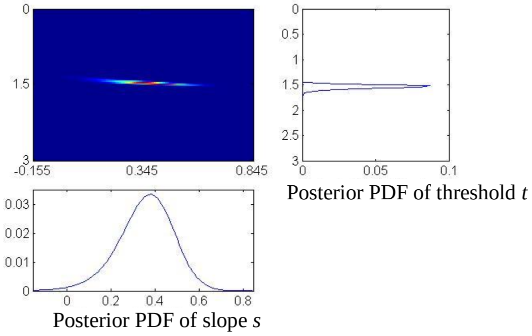
Figure 5: The model predictions based on the prior distribution (left) and the posterior distribution at the 150th stage (right). Darker colors indicate high probabilities.

The joint and marginal posterior distributions of threshold and slope at the end of the experiment are shown in Figure 6. Both the posterior distributions tend to converge to the true values of the parameters. Conforming to the previous studies, the estimation of the threshold is more accurate and has less variation in its posterior distribution while the estimation of slope is less stable.

In each stage, one point estimate is computed for each parameter by calculating the mean of the distribution. 100 experiments of 150 stages were run. Let θ_i be the point estimate in each stage, and θ_{true} be the true parameter value, each in log decimal scale. Then we can compute the average bias and standard deviation of the estimation in each stage across the 100 experiments by

$$bias(\theta)=\frac{\sum_{i=1}^{i=I}(\theta_i-\theta_{true})}{I}\cdot 20dB,$$

$$SD(\theta)=\sqrt{\frac{\sum_{i=1}^{i=I}(\theta_i-\theta_{true})^2}{I-1}}\cdot 20dB.$$



Posterior PDF of threshold *t*

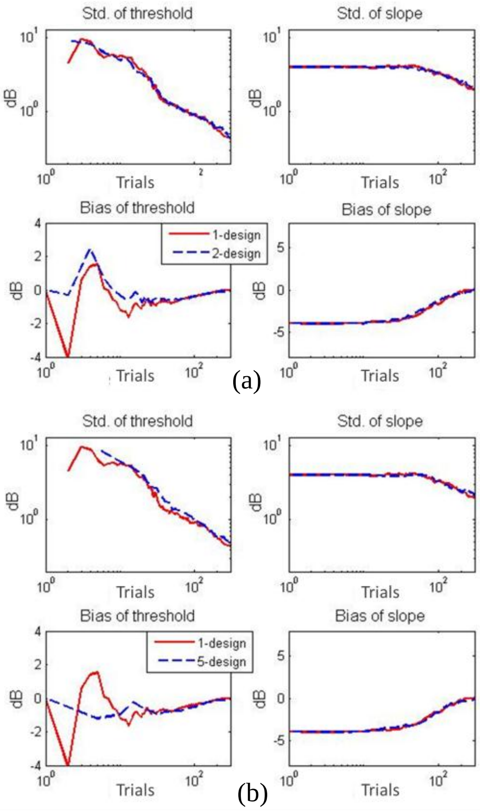
Posterior PDF of slope *s*

Figure 6: The joint and marginal posterior distributions of threshold and slope at the 300th trial.

To compare the two-designs-per-stage scheme with the traditional one-design-per-stage scheme, we ran 100 experiments of 300 stages with one design in each stage and computed the bias and standard deviation of the estimates in each stage. Figure 7(a) shows the comparison between the two different schemes. In the later trials, the two different schemes do not seem to have significant differences. There is no significant bias at the 300th trial for both threshold and slope. The standard deviation of threshold is about 0.2dB and that of slope is about 1.1dB. Although the two-designs-per-stage scheme has less fluctuation in the early stages in the bias of threshold, the difference may result from the random effect.

Next, the five-designs-per-stage scheme was implemented. Because the dimension increases, 200 generations are needed for DE to converge. One hundred experiments of 60 stages (300 trials in total still) were run and the point estimates were computed for each stage. Figure 7(b) shows the comparison between the five-designs-per-stage and the one-designs-per-stage schemes. We can see that there is much less fluctuation in the bias of threshold for five-designs-per-stage than that of one-design-per-stage at the early trials, which is consistent with the improvement in the two-designs-per-stage scheme. Other than that, there is no obvious difference between the two schemes.

As expected, simply increasing the number of designs in one stage while still keeping the total number of trials constant resulted in improvement in the accuracy of parameter estimation, at least at the early stages. As we hypothesized, the relationship or synergy provided by multiple responses is greater or at least different than the sum of the information from single responses. We expect that such improvement can be more obvious when it is applied to more complex models because in those cases, more trials are needed for simply exploring the model in the early stages of an experiment. However, we should not expect that the performance continues to improve as the number of designs per stage increases. By the principle of ADO, a good design should be based on solid information conveyed by the participants' responses. A large number of designs per stage may probe into unfruitful regions of the design space. A balance must be sought in deciding how many designs per stage are good for different models.



(a)

(b)

Figure 7: The comparison of one design per stage with two designs per stage (a) and five designs per stage (b) in the bias and standard deviation of the estimates of threshold and slope.

Conclusion

In psychophysical studies, many endeavors have been made to bring further efficiency to the process in parameter estimation. One clear direction is in *global optimization* or multiple steps ahead to improve the current greedy method that only evaluates the design utilities at the next stage. If global optimization provides the ultimate solution, the approach we studied in this paper, multiple designs per stage, is an initial step in this direction. Thus, in this paper, we sought one eclectic choice between the traditional one-shot experimental design at the very beginning of an experiment and the advanced adaptive experimentation with only one design per stage. The results showed that multiple designs per stage can benefit the estimation in the early stages of an experiment. The reason for the benefit is reminiscent of holistics in Gestalt psychology and the principle in information theory, with the multiple responses offering extra information than the sum of the individual responses.

To realize the optimization of multiple designs in one stage, we integrated the adaptive design optimization framework with an evolutionary computation algorithm, differential evolution, which is specialized in searching a multi-dimensional space for the purpose of optimization. DE can also be naturally applied to models that contain multiple design variables, for which brute-force grid search is usually applied. DE is less computationally demanding than grid search when the design space is large. Other than that, DE can also benefit from parallel computing to accelerate the computation within each experimental stage.

As such, DE-based adaptive design optimization has large potential of applications in the future experiments for parameter estimation.

References

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

Cavagnaro, D.R., Pitt, M.A., & Myung, J.I. (2011). Model discrimination through adaptive experimentation. *Psychonomic Bulletin & Review*, 18(1), 204-210.

Gu, H. (2012). Graphic-Processing-Units Based Adaptive Parameter Estimation of a Visual Psychophysical Model. Master thesis submitted to the Department of Psychology of the Ohio State University.

Kontsevich, L. & Tyler, C. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, *39,* 2729-2737.

Lesmes, L.A., Lu, Z., Baek, J., & Albright, T.D. (2010). Bayesian adaptive estimation of the contrast sensitivity function: the quick CSF method. *Journal of Vision*, *10(3)*, *17*, 1-21.

Lewi, J., Butera, R., & Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, *21*, 619-687.

Miller, F., Dette, H., & Guilbaud, O. (2007). Optimal designs for estimating the interesting part of a dose-effect curve. *Journal of Biopharmaceutical Statistics*, *17*, 6.

Myung, J. I., Cavagnaro, D. R. & Pitt, M. A. (2012). A tutorial on adaptive design optimization. Manuscript submitted for publication.

Storn, R. & Price, K. (1997). Differential evolution – a simple and efficient heuristic for glabal optimization over continuous spaces. *Journal of Global Optimization 11:* 341-359.

Wichmann, F. A. & Hill, N. J (2001). The psychometric function: I. fitting, sampling and goodness of fit. *Perception & Psychophysics*, 63(8),1293-1313.