# Batch sequential experimental designs for computer experiments

by Aaron Quan and William Notz

DAEW05

Cambridge, UK

September 2011

## Context: Computer experiments.

For this talk, we consider deterministic computer experiments, namely a (slow) computer code (a simulator) produces deterministic output $y(\boldsymbol{x})$ at input

$$\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$$

In what follows, we will assume $\mathcal{X}$ is $[0, 1]^d$.

## Context: Computer experiments.

A well-defined goal exists. This might be to determine the input that yields the optimum output (see Jones et al. 1998), determine the inputs that produce a given value of the output (see Ranjan et al. 2008) , or find a fast emulator (statistical predictor of the output) that is a good global approximation to the simulator (see Ruichen et al. 2003).

## Context: Computer experiments.

To accomplish this goal we will observe the simulator at $n$ points

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$$

and call this the (experimental) design.

## Context: Computer experiments.

A GaSP model is fit to the output from the simulator, and this is used to help accomplish our goal. In the interest of time, I assume everyone is familiar with these models and associated predictors such as the EBLUP.

## Batch Sequential Design.

Design involves the selection of the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ at which to observe the simulator. I assume the following.

- We use sequential designs. By this I mean a few inputs are selected, the simulator is run at these inputs, and based on what we observe, we decide which inputs to observe next for the purpose of accomplishing our goal.

## Batch Sequential Design.

- We have $m > 1$ computers available to run the simulation, so we can observe $m$ inputs simultaneously.

## Batch Sequential Design.

- $n = bm$, so the design will be sequential in $b$ "batches" of $m$ inputs. Observations from previous batches are used to determine the inputs for the next batch.

We refer to such designs as *batch sequential designs.*

## Batch Sequential Design.

- Some criterion for selecting inputs so as to facilitate the goals of the experiment is available.

Examples are improvement criteria, a mean squared error (MSE) criterion, and entropy.

## Batch Sequential Design.

Improvement criteria lend themselves to one-at-a-time sequential design. The next point selected is one that maximizes some function of the data observed so far.

An example of an improvement criteria is that used in Jones et al. (1998) for finding the optimum output. For a possible observation $Y(\boldsymbol{x})$ at input $\boldsymbol{x}$, the improvement is

$$I(\boldsymbol{x}) = max(f_{min} - Y(\boldsymbol{x}), 0)$$

where $f_{min}$ is the current minimum of points observed so far.

## Batch Sequential Design.

The next input to be observed is the one maximizing the expected value of the improvement, namely

$$E[I(\boldsymbol{x})] = (f_{min} - \widehat{y}(\boldsymbol{x}))\Phi(\tfrac{f_{min}-\widehat{y}(\boldsymbol{x})}{s}(\boldsymbol{x})) + s(\boldsymbol{x})\phi(\tfrac{f_{min}-\widehat{y}(\boldsymbol{x})}{s(\boldsymbol{x})})$$

where $\widehat{y}(\boldsymbol{x})$ is the EBLUP, $s(\boldsymbol{x})$ is the estimated standard error of the BLUP, $\Phi$ is the cumulative distribution function of the standard Normal distribution, and $\phi$ is the standard Normal density.

## Batch Sequential Design.

Loeppky, Moore, and Williams (2010) discuss batch sequential design in the context of insuring good global fit of the model. Among other things, they find the following.

## Batch Sequential Design.

- It is not always obvious how to implement some improvement criteria, which work well in a one-at-a-time sequential manner, in batches. In particular, Loeppky, Moore, and Williams (2010) state that “methods relying on the data $\boldsymbol{y}$) are not readily extendable to a batch sequential strategy.” This includes strategies based on improvement criteria.

- The methods that performed best in their study were an entropy based approach and a distance based (space-filling) approach. (Surprising in that these methods use little or no information about the observed response surface nor any formal optimality criteria related to global fit of the model.)

## Batch Sequential Design.

The motivation for our work is the issue of how to extend methods that rely on the data to a batch sequential strategy.

## Batch Sequential Design.

Our goal is to find ways to implement one-at-a-time sequential methods based on an expected improvement (EI) criterion, in $b$ batches consisting of $m > 1$ inputs per batch. Naive implementations (select the $m$ inputs with the largest value of the EI) can lead to "clumping" and this must be avoided.

Here are some methods we have begun studying. These are very preliminary results, but do indicate how one might implement improvement-based methods in batches.

## Batch Sequential Design.

Method 1 - Standardized Sequential Euclidean Distances (SSED)

The first batch point is the site with the best EI. In selecting the subsequent batch points, all sites less than some user specified distance $r$ from any site already selected in the batch are removed. From the remaining sites, the location with the best EI is selected.

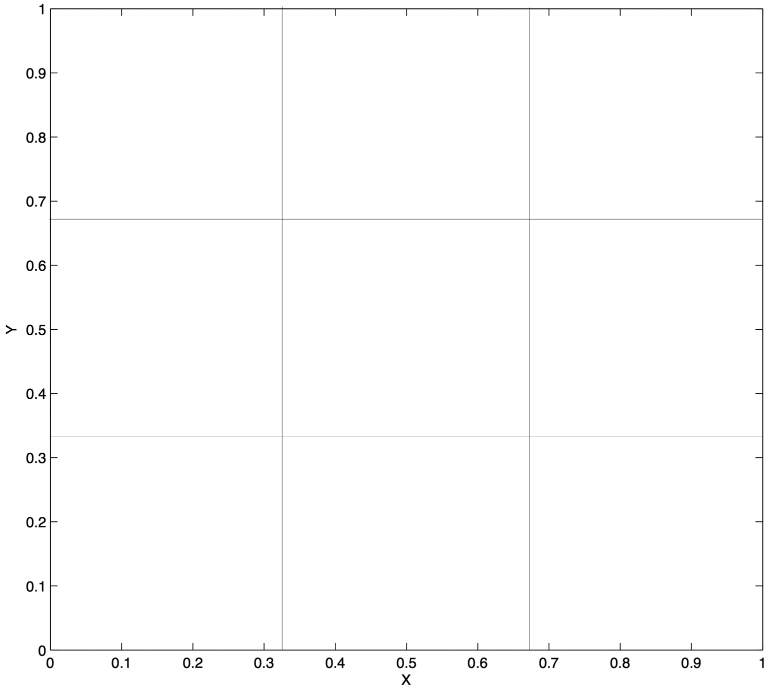
## Batch Sequential Design.

Method 2 - Sequential Batch Latin Hypercube (SBLH)

Divide $\mathcal{X}$ into $m^d$ cells (a grid) with identical dimensions. The first site selected is the one with the best value of EI. All cells in the same row, column, etc. as the cell containing the site selected are eliminated. From the remaining eligible sites, the one with the best EI is selected. Further eliminate all cells in the same row, column, etc. as the cell containing the site selected. The process repeats until $m$ sites are selected.

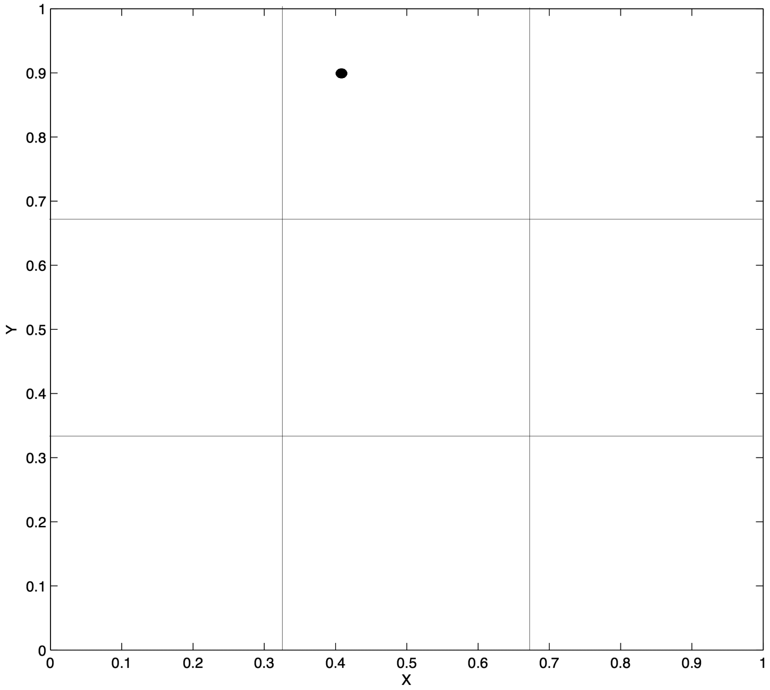
# Batch Sequential Design.

## Method 2 - Example



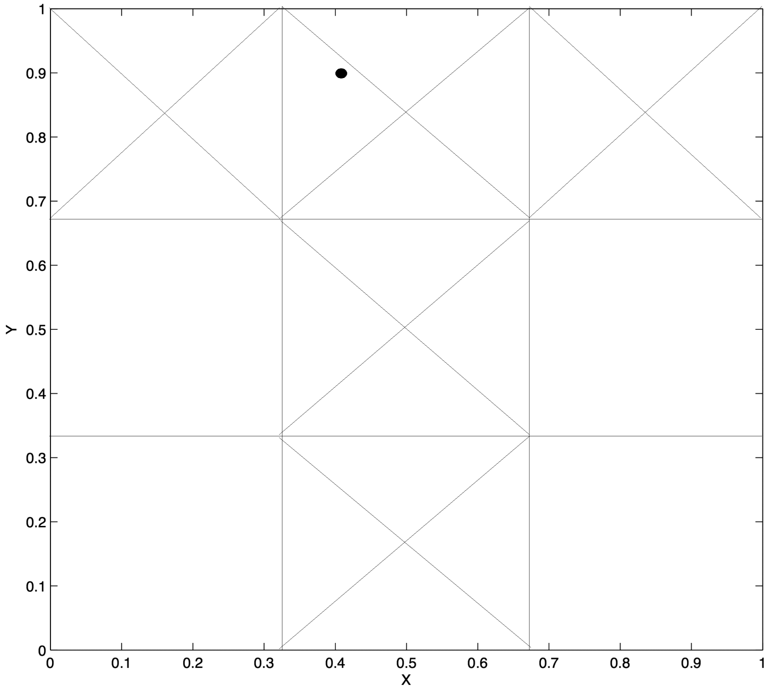
## Batch Sequential Design.

### Method 2 - Example



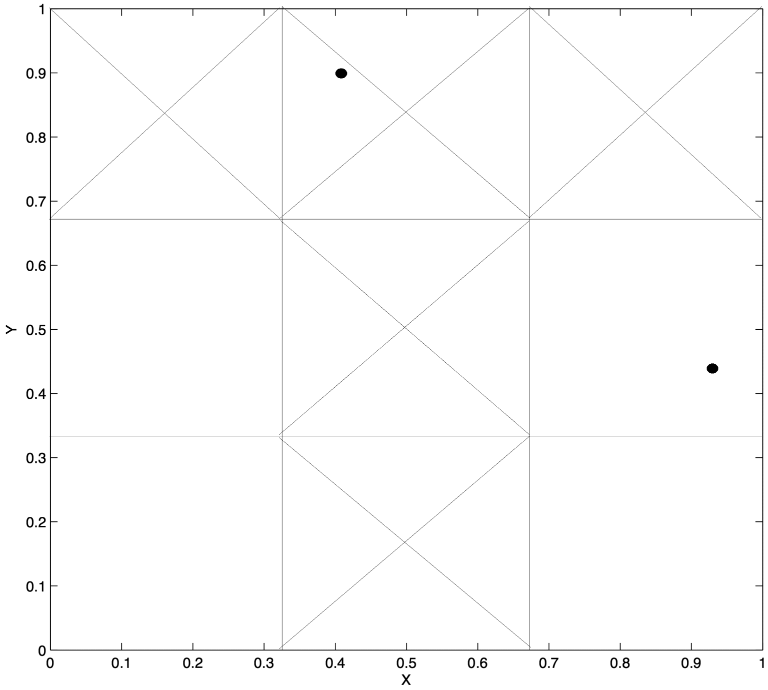
# Batch Sequential Design.

## Method 2 - Example



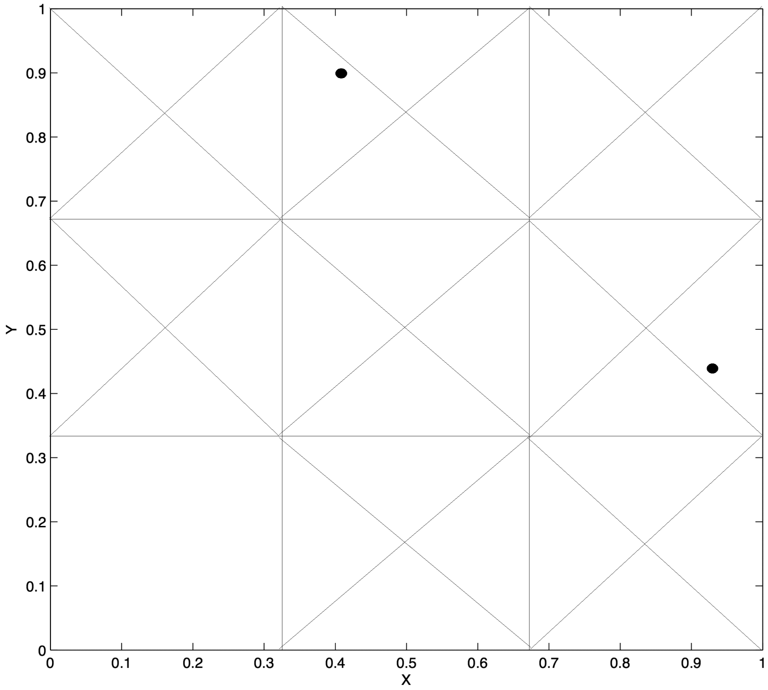
# Batch Sequential Design.

## Method 2 - Example



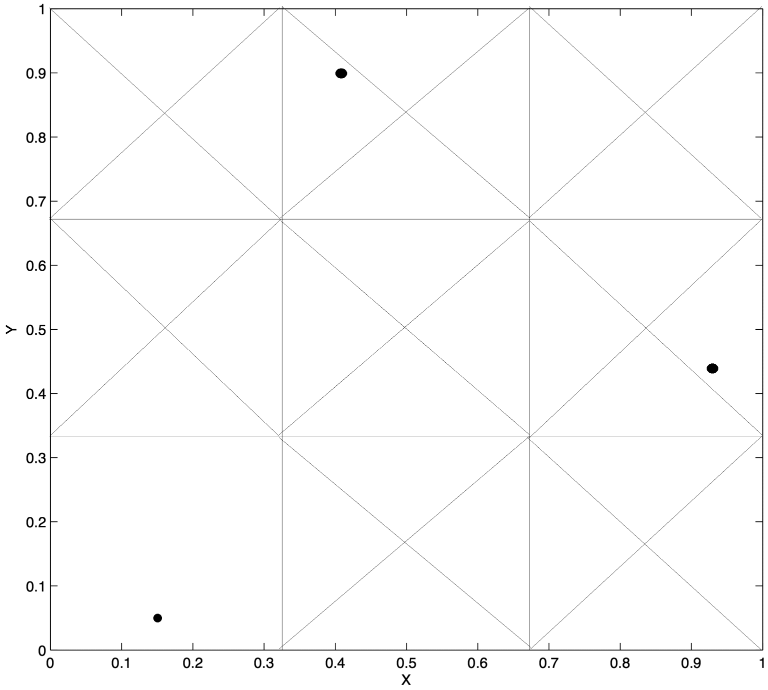
## Batch Sequential Design.

### Method 2 - Example



# Batch Sequential Design.

## Method 2 - Example



## Batch Sequential Design.

Method 3 - Sobol Sequential Batch Design (SSBD)

Generate a “large” $d$-dimensional Sobol sequence. To begin the selection of the $m$ points in a batch, the site in the Sobol sequence with the best EI is selected. The next $m-1$ sites of the Sobol sequence are added to complete the batch. Eliminate all points in the Sobol sequence that precede those selected. For the next batch, the site with the best EI in the remaining points in the sequence is selected and again the next $m-1$ sites of the Sobol sequence are added to complete the batch. This process is repeated until $b$ batches have been selected.

## Batch Sequential Design.

Method 4 - Improvement with Distance Penalty (IDP)

Define

$$F(\boldsymbol{x}|\boldsymbol{X}_b) = max_{\boldsymbol{x}^* \in \boldsymbol{X}_b} p(\boldsymbol{x}, \boldsymbol{x}^*) EI(\boldsymbol{x})$$

where $EI(\boldsymbol{x})$ is the EI at the site $\boldsymbol{x}$ and $p(\boldsymbol{x}, \boldsymbol{x}^*)$ is a penalty function for $\boldsymbol{x}$. This penalty function satisfies $p(\boldsymbol{x}, \boldsymbol{x}^*) = 0$ when $\boldsymbol{x} = \boldsymbol{x}^*$ for any point $\boldsymbol{x}^*$ in the previously selected batch sites $\boldsymbol{X}_b$. Possible penalty functions are Euclidean distance, squared Euclidean distance, and absolute distance.

## Batch Sequential Design.

A generalization of $F(\boldsymbol{x}|\boldsymbol{X}_b)$ is

$$max_{\boldsymbol{x}^* \in \boldsymbol{X}_b} C(p(\boldsymbol{x}, \boldsymbol{x}^*), EI(\boldsymbol{x}))$$

where $C(\cdot, \cdot)$ is a strictly monotone increasing function in each argument.

While this method is intended to spread out the sites in a batch, all that it does is prevent extreme clustering of points. The reason is that EI is close to 0 at most sites and the penalty function may have negligible effect at these sites.

## Batch Sequential Design.

Method 5 - Maximum Improvement Space-Filling Selection (MISFS)

The region where sites in a batch can be sampled is restricted to sites whose EI is higher then a preset minimum value, $h$. Denote this region by $\mathcal{X}_{EI>h}$ We allow $h$ to change for each batch, and ideally it will be some percentage of the maximum improvement value since this maximum EI tends to decrease as points are added. In $\mathcal{X}_{EI>h}$, a space-filling design is used select the batch sites in addition to the point with the highest EI.

The challenge with this method is that the region where sites can be selected may be nonrectangular and disconnected. Selecting a space-filling design from such a region may be difficult. This leads to two implementations.

## Batch Sequential Design.

Method 5a - Maximum Improvement Uniform Random Sampling (MIURS)

Select the first point in the batch as the point with the maximum value of EI. Use uniform random sampling to select points in $\mathcal{X}$, retaining the first $m - 1$ that fall within $\mathcal{X}_{EI>h}$.

## Batch Sequential Design.

Method 5b - Maximum Improvement Sobol Sequence Batch Design (MISSB)

Select the first point in the batch as the point with the maximum value of EI. The next site is the first one in the Sobol sequence that has an EI value of at least $h$. This process continues until a total of $m - 1$ sites are selected for the batch. For each subsequent selection of points in batches, the sequence continues onward from the point where the last batch site was sampled in the previous batch.

## Batch Sequential Design.

Method 6 - Refitted Best Linear Unbiased Predictors (RBLUP)

The first point in a batch, $\boldsymbol{x}_{b1}$, is the point with maximum EI. $\widehat{Y}(\boldsymbol{x}_{b1})$ is calculated using the EBLUP, treated as if it is the value $y(\boldsymbol{x}_{b1})$ that would be produced by the simulator, and is temporarily "added" to the data. The EBLUP is refitted to this expanded data set, and the next point $\boldsymbol{x}_{b2}$is selected by EI, subject perhaps subject to proximity constraints. Again, $\widehat{Y}(\boldsymbol{x}_{b2})$ is calculated using the BLUP, treated as if it is the value $y(\boldsymbol{x}_{b2})$ that would be produced by the simulator, and is temporarily "added" to the data. This is repeated until all $m$ sites for the batch are selected.

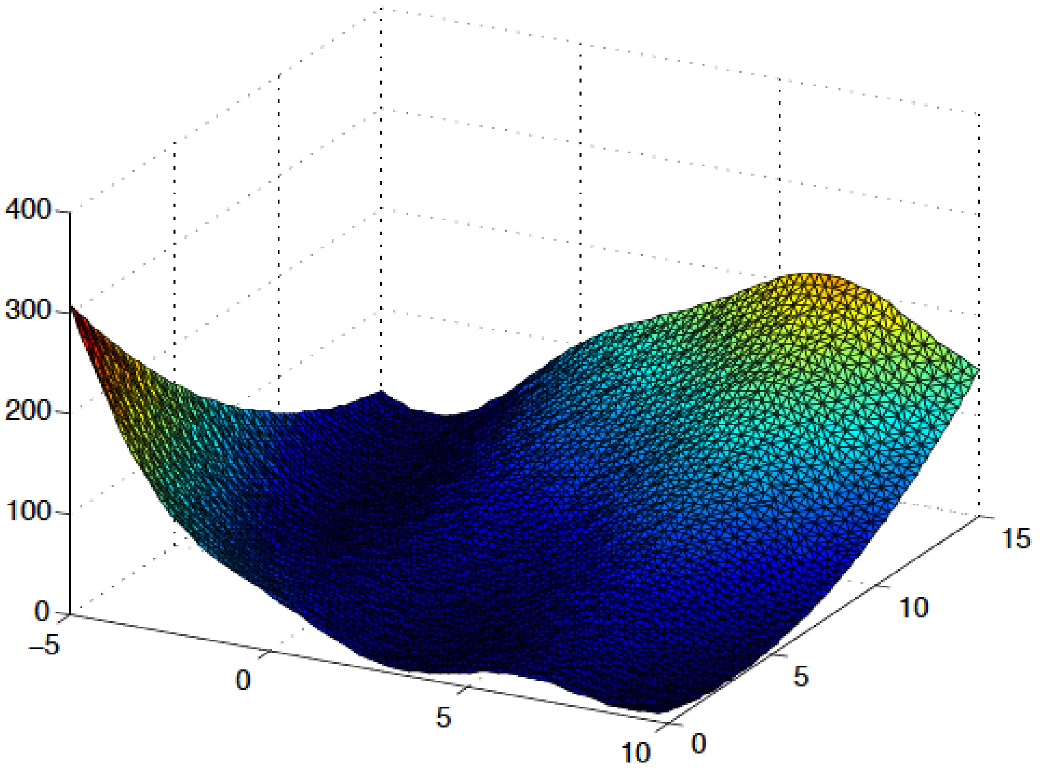
## Example.

To explore the performance of these methods, we use the Branin function

$$B(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(\boldsymbol{x}_2 - \frac{5}{4\pi^2}\boldsymbol{x}_1^2 + \frac{5}{pi}\boldsymbol{x}_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) cos(\boldsymbol{x}_1) + 10$$

as our emulator. We assume our goal is to find the minimum of the Branin function. Here $-5 \leq \boldsymbol{x}_1 \leq 10$, $0 \leq \boldsymbol{x}_2 \leq 15$. There are three global minima at $(\boldsymbol{x}_1, \boldsymbol{x}_2) = (-\pi, 12.275)$, $(\pi, 2.275)$, and $(9.42478, 2.475)$, with $B(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0.397887$ at these minima.

Example. The Branin function



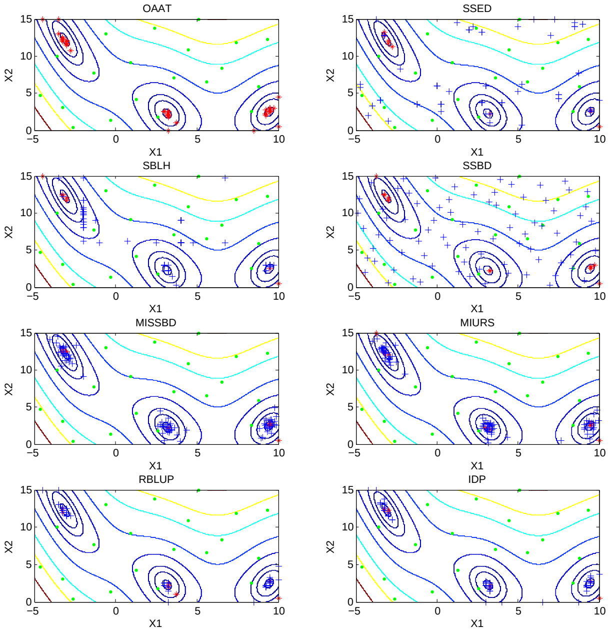
## Example.

All the methods described are used, with the MISSBD and MI-URS variants of the MISFS, and tested to determine which minima the methods arrive at and to examine their general behavior. An initial maximin LHD consisting of 20 sites is used (the same for all methods). The batch size is $m = 5$ and 20 batches are added. We also include the traditional OAAT approach described in Schonlau (1998).

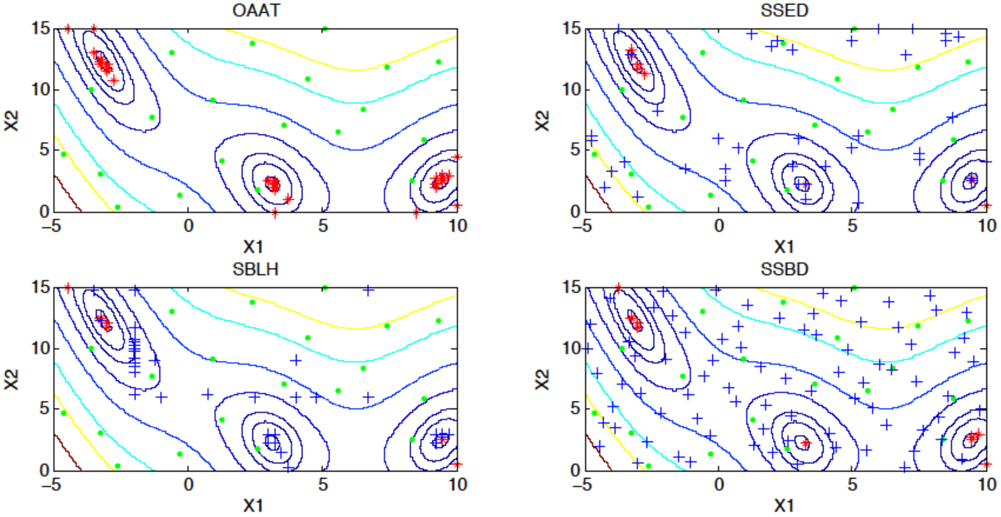
## Example.

In the following, selected site locations of the different batch methods are overlaid on a contour plot of the Branin function. Green dots $(\cdot)$ represent the location of the initial set selected by the LHD, with the same set used for all methods. Red asterisks $(*)$ represent the location of the sites sampled for having he highest EI. Blue pluses $(+)$ represent the location of the batch sites selected by the methods.

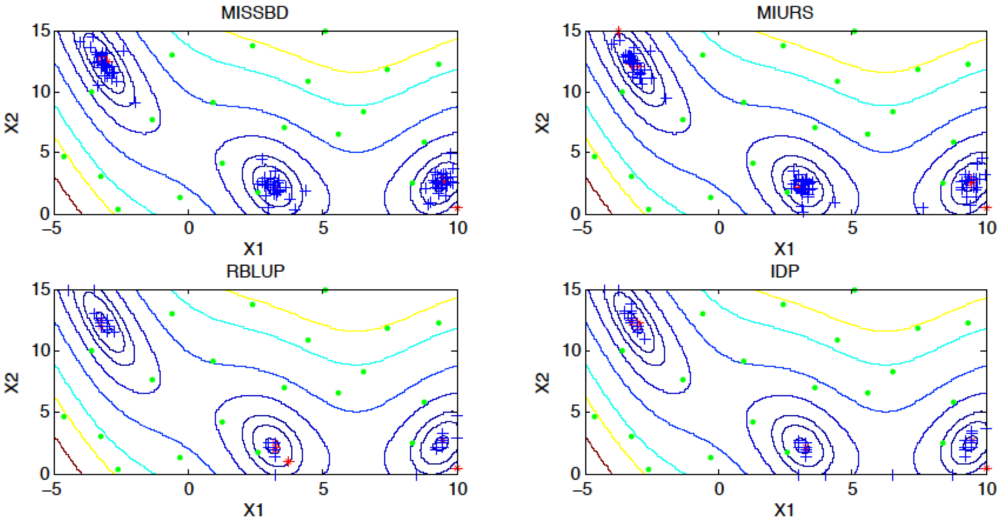
# Example.



# Example.



# Example.



## Example.

| Method | Solution 1 | Solution 2 | Solution 3 |
|---|---|---|---|
| OAAT | (-3.25, 12.5) | (3.25, 2.25) | (9.5, 2.5) |
| SSED | (-3, 12) | (3.25, 2.25) | (9.5, 2.5) |
| SBLH | (-3.25, 12.5) | (3.25, 2.25) | (9.5, 2.5) |
| SSBD | (-3.25, 12.5) | (2.617, 2.461) | (9.5, 2.5) |
| MISSBD | (-3.196, 12.585) | (3.150, 2.195) | (9.387, 2.521) |
| MIURS | (-3.171, 12.356) | (3.142, 2.460) | (9.5, 2.5) |
| RBLUP | (-3.25, 12.5) | (3.25, 2) | (9.5, 2.5) |
| IDP | (-3.25, 12.5) | (3.25, 2.25) | (9.5, 2.5) |
| Actual | $(-\pi, 12.275)$ | $(\pi, 2.275)$ | $(9.42478, 2.475)$ |

## Conclusions.

This is just one example, but here are some comments.

## Conclusions.

- OAAT: Converges quickly (in terms of value of EI) to the solutions.
- SSED: Converges more slowly, tends not to jump between solutions.
- SBLH: Converges more slowly. Seems to lock onto solution (3) for most of the runs.
- SSBD: Converges more slowly, sites spread out over the domain, alternates around all three solutions.

- MISSBD: Converges quickly, sites somewhat clustered, alternates around all three solutions.

- MIURS: Converges quickly, sites somewhat clustered, alternates around all three solutions.

- RBLUP: Converges quickly, sites heavily clustered, seems to lock onto solution (2)

- IDP: Converges quickly, sites heavily clustered, does not revisit solution (1) after a few iterations, but finds all three solution

## Conclusions.

All these results are very preliminary and more a proof of concept than anything else. Improvements in some of the computational aspects are needed and a variety of examples (surfaces, improvement functions) need to be investigated.

Questions?

A copy of this talk is available at

http://www.stat.ohio-state.edu/~comp_exp/jour.club/papers.html

## References

1. Jones, D. R., Schonlau. M. and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions.

2. Williams, B. J., Santner, T. J., and Notz, W. I. (2000). Sequential design of computer experiments to minimize integrated response functions, *Statistica Sinica*, **10**, 1133-1152.

3. Ruichen, J., Chen, W., and Sudjianto, A. (2003). On sequential sampling for global metamodeling in engineering design. In *Proceedings of DETC 2002: 2002 ASME Design Engineering Technical Conference*.

4. Ranjan, P., Bingham, D., and Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics,* **50**, 527-541.

5. Lam, C.Q. and Notz, W.I. (2008). Sequential adaptive designs in computer experiments for response surface model fit. *Statistics and Applications,* **6**, 207-233. *Journal of Global Optimization,* **13**, 455-492.

6. Loeppky, J., Moore, L., and Williams, B.J. (2010). Batch sequential designs for computer experiments. *Journal of Statistical Planning and Inference,* **140**, 1452-1464.