

# The influence of Experimental Designs on the performance of surrogate model based costly global optimization solvers

Nils-Hassan Quttineh\* and Kenneth Holmström\*

**Abstract** When dealing with costly objective functions in optimization, one good alternative is to use a surrogate model approach. A common feature for all such methods is the need of an initial set of points, or "experimental design", in order to start the algorithm. Since the behavior of the algorithms often depends heavily on this set, the question is how to choose a good experimental design. We investigate this by solving a number of problems using different designs, and compare the outcome with respect to function evaluations and a root mean square error test of the true function versus the surrogate model produced. Each combination of problem and design is solved by 3 different solvers available in the TOMLAB optimization environment. Results indicate two designs as superior.

**Keywords:** Black-box, Surrogate model, Costly functions, Latin Hypercube Designs, Experimental Design.

## Abbreviations:

CGO      Costly Global Optimization  
ExD      Experimental Design  
MINLP    Mixed-Integer Nonlinear Programming

**Authors:** N-H. Quttineh got a M.Sc. in Optimization at Linköping University 2004 and is a graduate student at Mälardalen University since 2005, with main subject *Algorithms for costly global optimization*.

K. Holmström got a Ph.D. in Numerical Analysis at Umeå University 1988 and did industrial research and development work for ABB 1990-93. Besides his academic career, he has been consultant in more than 100 industrial and scientific projects. Since 2001 he is Full Professor in Optimization at Mälardalen University with main research interests algorithm and software development for industrial and financial optimization problems, in particular costly global mixed-integer constrained nonlinear optimization. Holmström is CEO of Tomlab Optimization Inc. and creator of TOMLAB, an advanced MATLAB optimization environment distributed since 1998.

---

\*Department of Applied mathematics, Mälardalen University, SE-721 23 Västerås, Sweden.

## 1 Introduction

Global optimization of continuous black-box functions that are costly (computationally expensive, CPU-intensive) to evaluate is a challenging problem. Several approaches based on response surface techniques have been developed over the years. A common feature is that, unlike local optimization methods, every computed function value is saved and utilized.

Problems that are costly to evaluate are commonly found in engineering design, as well as industrial and financial applications. A function value could be the result of a complex computer program or an advanced simulation, e.g. computational fluid dynamics (CFD). Hence consuming anything from a few minutes to many hours of CPU time.

From an application perspective there are often restrictions on the variables besides the lower and upper bounds, such as linear, nonlinear or even integer constraints. The most general problem formulation is as follows:

### The Mixed-Integer Costly Global Black-Box Nonconvex Problem

$$\begin{aligned}
 \min_x \quad & f(x) \\
 \text{s/t} \quad & -\infty < x_L \leq x \leq x_U < \infty \\
 & b_L \leq Ax \leq b_U \\
 & c_L \leq c(x) \leq c_U \\
 & x_j \in \mathbb{N} \quad \forall j \in \mathbb{I} \quad ,
 \end{aligned} \tag{1}$$

where  $f(x) \in \mathbb{R}$  and  $x_L, x, x_U \in \mathbb{R}^d$ . Matrix  $A \in \mathbb{R}^{m_1 \times d}$ ,  $b_L, b_U \in \mathbb{R}^{m_1}$ ; defines the  $m_1$  linear constraints and  $c_L, c(x), c_U \in \mathbb{R}^{m_2}$  defines the  $m_2$  nonlinear constraints. The variables  $x_I$  are restricted to be integers, where  $\mathbb{I}$  is an index subset of  $\{1, \dots, d\}$ . Let  $\Omega \in \mathbb{R}^d$  be the feasible set defined only by the simple bounds, the box constraints, and  $\Omega_C \in \mathbb{R}^d$  be the feasible set defined by all the constraints in (1).

Almost every Costly Global Optimization (CGO) solver utilize a surrogate model, or response surface, to approximate the true (costly) function. The RBF algorithm introduced by Powell and Gutmann [2, 9] use radial basis function interpolation to build an approximating surrogate model. The EGO algorithm by Jones et al. [6] utilizes the DACE framework. By optimizing a less costly utility function these algorithms determine a new point, where afterwards the original objective function is evaluated. This is repeated until some convergence criteria is fulfilled.

## 2 Experimental Designs

Common for all surrogate model CGO solvers is the need of an initial sample of points (experimental design) to be able to generate the initial surrogate model. For all these points the costly function values are calculated. The initial surrogate model is built from these sampled points and used as an approximation of the true function. A new

point to sample is then decided by some algorithmic strategy, and this continues until some convergence criteria is met.

It is not obvious how to choose this initial set of points, but there are some criteria we strive to fulfill. As the problems to solve are considered black-box, we have no idea what the function might look like. Therefore it is most important that the experimental design have some sort of space filling ability, i.e. avoid sampling only a certain part of the design space.

### 2.1 Deterministic Global Solver

It is of course possible to utilize any standard global optimization solver for a limited number of iterations, just in order to get an initial set of sample points for the surrogate model to get going. After all, deterministic global optimization algorithms are designed to find the global optimum as fast as possible, so why not use this fact and let the solver find good initial points.

In this paper we utilize the DIRECT algorithm (DIviding RECTangles) by Jones et al. [5], implemented in the TOMLAB Optimization Environment [4] as solver *glcDirect*<sup>1</sup>. This is a deterministic global optimization solver, but not itself suited for the costly problems considered. The maximal number of sample points  $N$  is possible to set when running *glcDirect*. But, because the algorithm generates more than one new point in each iteration, the costly function value might be computed for a few more sample points than  $N$ .

### 2.2 Corner Point Strategy

RBF solvers tend to sample points on the border, which seldom contribute as much information as interior points to the interpolation surface. This problem is thoroughly discussed by Holmström in [3]. To increase the chances of sampling interior points, a first idea was to sample all corner points of the box constraints  $\Omega$ , and additionally the midpoint of the box.

It turns out that unless the midpoint is the point with lowest function value, the initial interpolation surface will have its minimum somewhere on the boundary, and the CGO solver sometime samples a new border point. To avoid this, we propose to additionally sample corner points of half the bounding box, centered around the original midpoint, until an interior point with lowest function value is found. The idea is demonstrated in Figure 1 on page 4.

For problems in higher dimensions  $d$ , the exponential growth in number of corner points  $N = 2^d$  becomes an issue. A good alternative is then to sample only a subset of corner points. One idea is to sample only the lower left corner point of the bounding box plus all its adjacent corner points. This yields a more moderate number of initial sample points  $N = d + 1$ , which is also the minimum number of initial points needed for the initialization of the RBF algorithm. This is due to the fact that a minimum of  $N \geq d + 1$  points are required to build an interpolation surface.

---

<sup>1</sup><http://www.tomopt.com/tomlab/>

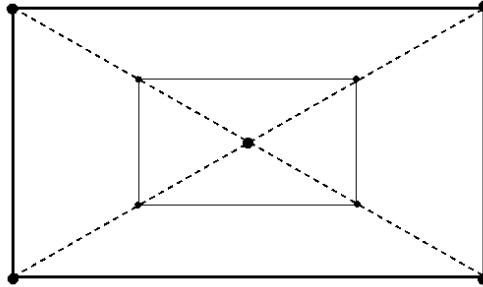


Figure 1: The Corner Point strategy in 2 dimensions.

A generalization of the previous corner idea is to choose both the lower left and the upper right corner points, plus all adjacent corner points. This gives an initial sample of size  $N = 2 \cdot (d + 1)$  if  $d > 2$ . In two and three dimensions, the strategy is equivalent to sampling all corner points.

### 2.3 Maximin LH Designs

Latin Hypercube Designs (LHD) is a popular choice of experimental design. The structure of LHDs ensure that the sampled points cover the sampling space in a good way. They also have a non-collapsing feature, i.e. no points ever share the same value in any dimension. Maximin LHDs give an even better design, as the points not only fulfill the structural properties of LHD designs, but also separate as much as possible in a given norm, e.g. the standard Euclidean norm. It is possible to generate Maximin LHDs for any number of points  $N$ .

A good collection of Maximin LHDs, together with many other space filling designs, can be found at <http://www.spacefillingdesigns.nl> together with state-of-the-art articles in this area.

## 3 Handling Constraints

When solving problems with additional constraints, besides the box constraints, it might be better to avoid sampling initial points that are not feasible since the function evaluation is extremely costly. We now describe how the proposed methods in Section 2 are adjusted to handle constraints, whenever possible. The methods presented here can not handle equality constraints at the moment, however nonlinear equality constraints are also difficult in general.

There exist other ideas on how to find a space filling initial sample, taking into account the constraints. Stinstra et al. [10] solve an optimization problem, where the objective is to maximize the minimum (euclidian) distance between  $N$  feasible points.

### 3.1 Constrained Deterministic Global Solver

We need to select a global deterministic solver that is able to handle constraints. The DIRECT algorithm was extended to handle nonlinear inequality constraints by Jones in [7]. In the TOMLAB implementation of the constrained DIRECT, *glcDirect*, the DIRECT algorithm is generalized to separately treat linear equality and inequality constraints, and nonlinear equality and inequality constraints. Since the algorithm always divides a rectangle in three pieces, infeasible points might still be included in the initial iterations, even if *glcDirect* has a feature to delete rectangles that are infeasible with respect to linear inequality constraints, and avoid computing  $f(x)$  for points infeasible with respect to linear and nonlinear constraints.

### 3.2 Corner Point Strategy

The Corner Point Strategy is not able to handle constraints in a straightforward way. It is possible to check which generated points are feasible, but what should be done if only a few of them are feasible? One could develop strategies on how to choose additional points, but then we diverge too much from the original idea of sampling the corner points. Therefore we only consider the basic approach, i.e. not taking constraints into account.

### 3.3 Constrained Maximin LH Designs

We have developed a method to create an initial sample fulfilling both the LHD structure and all constraints given for the problem. The method utilizes large Maximin LHDs, where the number of points in the design is significantly larger than the desired number of initial points, and only picks out the feasible points. The method is described in pseudo-code below, see Algorithm 1.

---

**Algorithm 1** Find  $N$  feasible Maximin LHD points

---

```
1: Initialize  $M := N + \#$  constraints.
2: Apply Maximin LHD with  $M$  points to constrained problem.
3: Calculate number of feasible points  $M_f$ .
4: if  $M_f == N$  then
5:   STOP.
6: else if  $M_f < N$  then
7:   Increase  $M$ , go to 2.
8: else
9:   Decrease  $M$ , go to 2.
10: end if
```

---

If the value of  $M_f$  starts to alternate between two values, one less than  $N$  and the other one greater than  $N$ , stop the algorithm and declare failure to find exactly  $N$  feasible points. The Maximin LHD with too many feasible points is used. The resulting design includes  $N$  feasible points with a Maximin LHD structure. An illustrative example is found in Figure 2 on page 6.

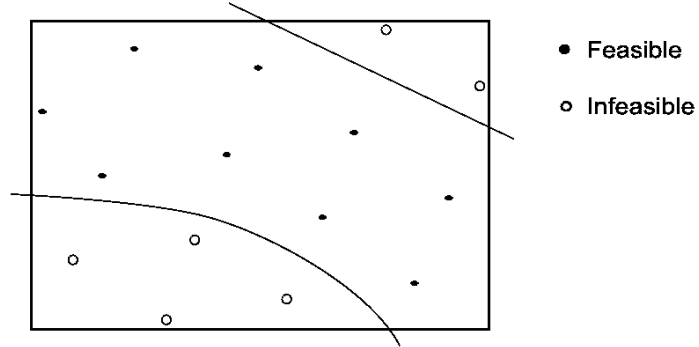


Figure 2: A 15 point Maximin LHD with  $M_f = 9$  feasible points.

## 4 Benchmark and Tests

Our aim is to test the set of experimental designs presented in previous sections. Define the set of experimental designs as  $E$ , and pick a set of test problems  $P$  and a set of solvers  $S$ .

Every combination of problem  $p \in P$  and experimental design  $e \in E$  is solved with each solver  $s \in S$ . Below the different designs, solvers and test problems used in the benchmark is presented. The set of experimental designs  $E$  is summarized in Table 1 on page 7. Information on the test problems are found in Table 2 on page 7.

Three solvers from the TOMLAB /CGO environment are used. The `rbfSolve` and `arbfmip` solvers utilize radial basis functions, and the `EGO` solver utilizes the DACE framework. The algorithmic structures are coded in MATLAB but all heavy calculations are in TOMLAB implemented in Fortran and C code, and interfaced using so called mex file interfaces.

### 4.1 Set of Experimental Designs

There are two main parameters to consider: first, the number of initial sample points  $N$ , and second, for constrained problems, whether or not to take the constraints into account. The tested combinations are described and motivated below.

#### Size $N$

The Corner Point Strategy generates a fixed number of initial sample points, one for each corner point of the bounding box. The other two strategies can generate any number of initial sample points. We use  $N_1 = (d + 1)(d + 2)/2$  and  $N_2 = 10 \cdot d + 1$ , where  $d$  is the dimension of the problem to be solved.

### Constraints

The Maximin LHD strategy can handle constraints by applying Algorithm 1. To test whether it is more efficient to force all the initial sample points to be feasible, all problems with constraints in combination with the Maximin LHD design are solved twice. First using the standard strategy and then applying a Maximin LHD with only feasible points.

### Combinations

Inspired by some preliminary results we also tried to combine the Corner Point Strategy with the other two designs. All corner points (no interior points) were added to the result of either the global optimization solver or the Maximin LHD.

**Table 1:** The Set of Experimental Designs ( $E$ ).

Experimental Design	Size of $N$	Constraints
Corner Points	Fixed	No
GO Solver	$N_1$ and $N_2$	Yes
Maximin LHD	$N_1$ and $N_2$	No
Maximin LHD	$N_1$ and $N_2$	Yes
Corners + GO	$N_1$ and $N_2$	Yes
Corners + LHD	$N_1$ and $N_2$	No
Corners + LHD	$N_1$ and $N_2$	Yes

## 4.2 Set of Test Problems

In total, a set of 15 box-bounded unconstrained problems  $P_U$  and a set of 6 constrained problems  $P_C$  are solved. Most of them are 2-dimensional problems, except a few problems in 3 and 4 dimensions. All problems in  $P_C$ , in combination with the Maximin LHD experimental design, are solved twice (with and without taking constraints into account).

**Table 2:** The Set of Test Problems ( $P$ ).

Problem set	$P_U$			$P_C$	
Dimension $d$	2	3	4	2	3
No. of problems	13	1	1	4	2

None of the test problems above have a global minimum in a corner point or midpoint, as this obviously would benefit the Corner Point Strategy.

Problems in only 2 or 3 dimensions might seem very simple, but even problems of this size are non-trivial and might be hard to solve when the problems are costly to compute. It is quite common that costly problems are of small size, with less than 10 unknowns.

## 5 Numerical Results

To present the benchmark results in an easy way, we utilize profiling. A performance profile [1] shows the relative performance of the solvers in  $S$  on the given set of problems  $P$ . However, performance profiles do not provide the number of function evaluations required to solve any of the problems.

Since function evaluations are expensive we are interested in the percentage of problems solved (for a given tolerance) within a given number of function evaluations. Data profiles [8] are specifically designed to handle this. These profiles are both probability density functions, but with an important difference. A data profile is independent of the set of solvers  $S$ , while the performance profiles are computed relative the other solvers in  $S$ .

### 5.1 Metrics

The solvers are set to break after 200 function evaluations or earlier if convergence to the known global optimum is obtained. The relative error is defined as

$$E_r = \frac{f_{min} - f_{opt}}{|f_{opt}|},$$

where  $f_{min}$  is the current best function value and  $f_{opt}$  is the known global optimum. Convergence is assumed if the relative error  $E_r$  is less than  $10^{-4}$ . In the case  $f_{opt} = 0$ , stop when  $f_{min}$  is less than  $10^{-4}$ . To compare the outcome of each experimental design, a number of metrics are used:

- f%** Number of function evaluations needed to reach 1,2,3 and 4 digits of accuracy ( $E_r \leq 10^{-k}$   $k = 1, 2, 3, 4$ ). This is the primary goal for most optimization problems.
- x%** Number of function evaluations needed to sample a point within 10% and 1% of the design space, centered around the global optimum. It is very important to sample points close to the global optimum. When this basin is found, the CGO solvers tend to converge quickly.
- RMS** When the algorithm stops, the final surrogate model  $s(x)$  is compared to the true function  $f(x)$ . A grid of points is used to calculate the Root Mean Square error.

$$RMS = \frac{1}{K} \cdot \left( \sum_{k=1}^K (f(x_k) - s(x_k))^2 \right)^{1/2}$$

For  $d = 2$ ,  $41^2 = 1681$  points are used. For  $d = 3$ ,  $21^3 = 9261$  points are used. For  $d = 4$ ,  $11^4 = 14641$  points are used. It is preferable if the final surrogate model capture the main features of the costly function, however the main goal is to find the global minimum with few function evaluations rather than having an overall good approximation of the objective function.

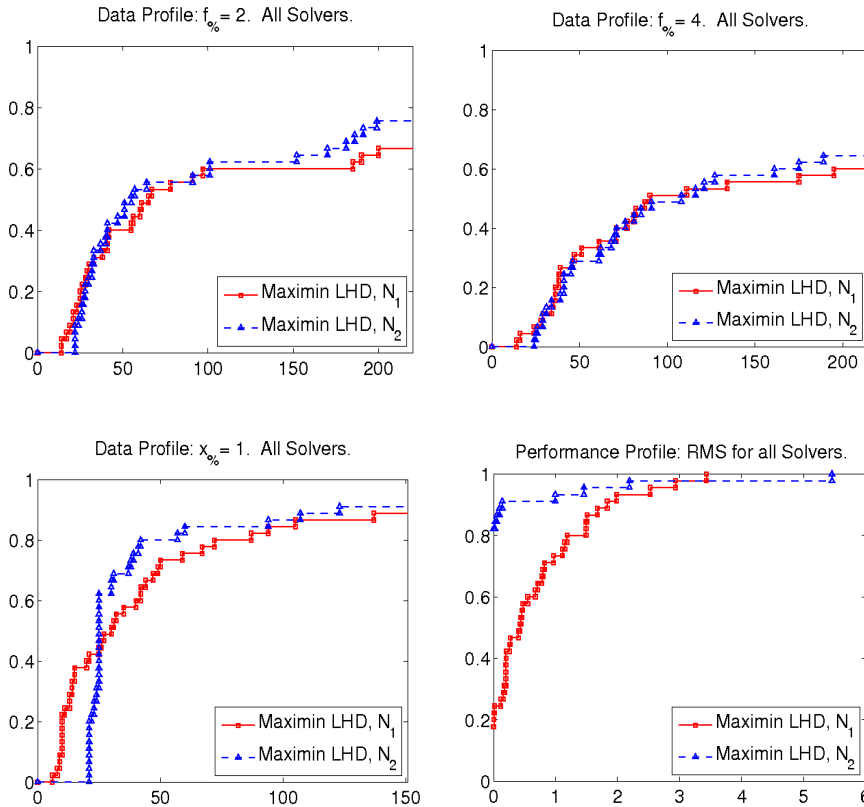
Smaller values are better for all metrics. To compare the experimental designs, data profiles for the costly  $f\%$  and  $x\%$  metrics are used. The RMS measure is not costly and presented using performance profiles.



## 5.2 Results

Since our focus of interest is to compare the performance of experimental designs, not specific solvers, accumulated results for each design are presented and discussed. Analysis for each solver has been done as well, and if any result differs significantly for a specific solver, a note is given.

We present the analysis for the set of unconstrained problems  $P_U$ , but results are valid for  $P_C$  as well if not specified otherwise. First compare the experimental designs where  $N$ , the number of initial points, was set to either  $N_1 = (d + 1)(d + 2)/2$  or  $N_2 = 10 \cdot d + 1$ .



**Figure 3:** Comparison of setting  $N_1$  and  $N_2$  for Maximin LHDs. Data profiles for the metrics  $f_{\%}$  and  $x_{\%}$  are used, and a performance profile for RMS.

Figure 3 shows that the Maximin LHD with  $N_2$  performs slightly better for all metrics. The results are similar for the deterministic global solver, and hence consider only the  $N_2$  setting in forthcoming analysis.

Overall best Experimental Design

Comparing the results of the three originally proposed experimental designs, the Corner Point Strategy and the global solver approach have a very similar success rate for all metrics, as seen in Figure 4. The Maximin LHD falls behind when it comes to finding many digits of accuracy, but is superior when looking at the RMS error. But as noted, a good RMS error is not the main goal in global optimization.

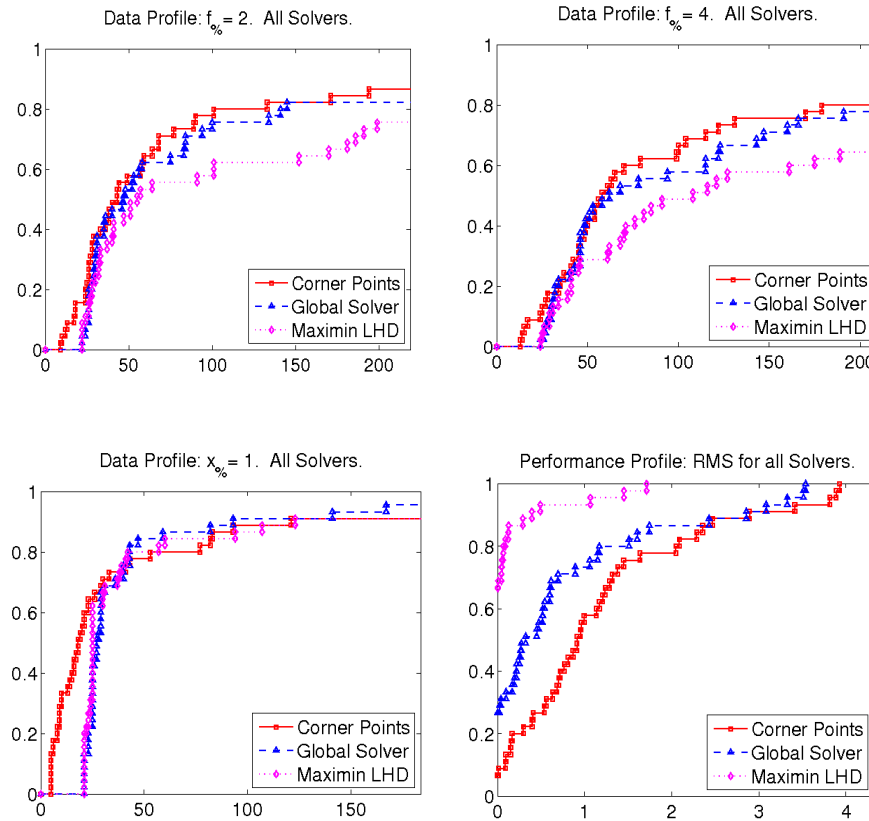


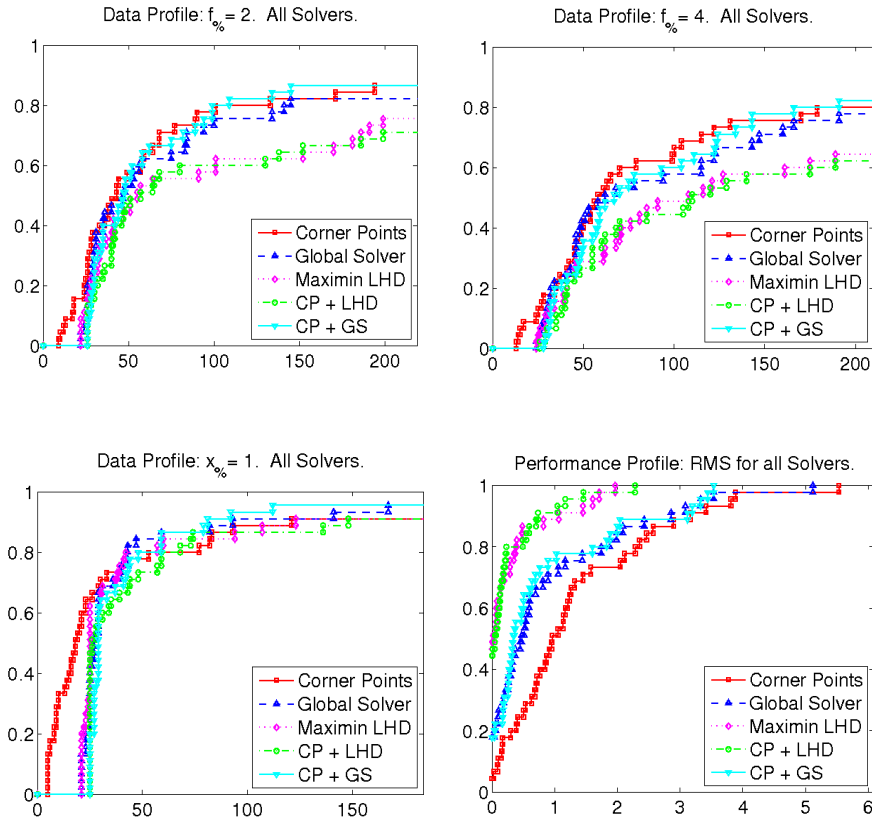
Figure 4: Comparison of the 3 proposed Experimental Designs. Data profiles for the metrics  $f_{\%}$  and  $x_{\%}$  are used, and a performance profile for RMS.

For the  $P_C$  problems, the Maximin LHD design performed much better and outperformed the other designs for all metrics. But since  $P_C$  contains only 6 problems this might just be a coincidence.

The high success rate of the Corner Point Strategy encouraged us to explore two combined versions. The global solver approach and the Maximin LHD is used as before, but the corner points of the bounding box are then added to the initial design.

## The influence of ExD on the Performance of CGO Solvers

This increases the number of initially sampled points somewhat, but should contribute to a more robust design. The outcome of this experiment is found in Figure 5.



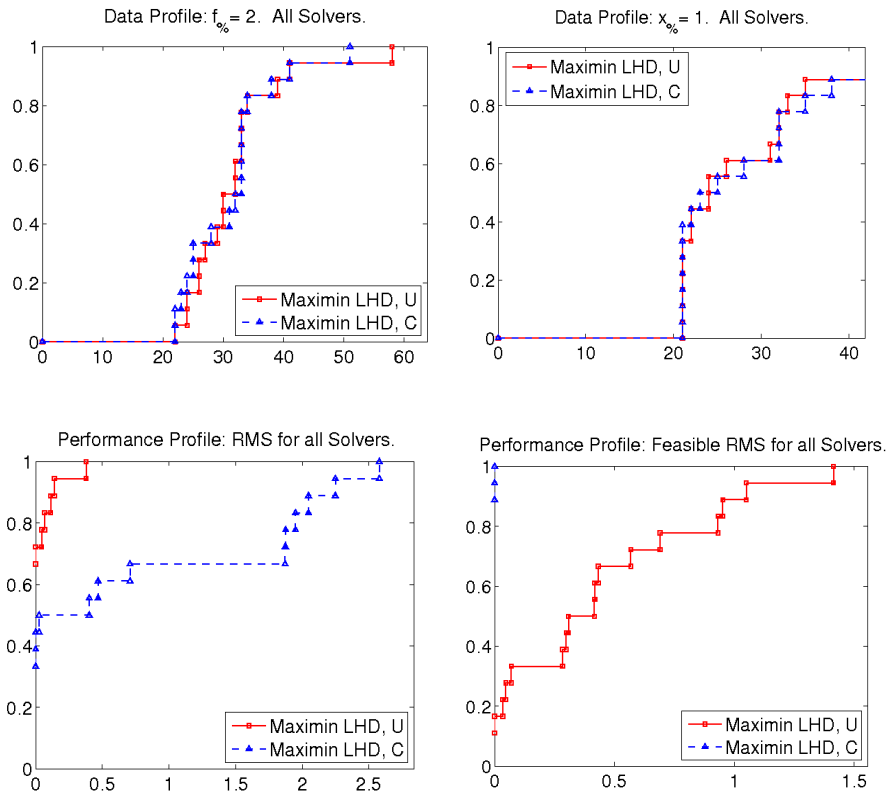
**Figure 5:** Comparison of Experimental Designs. Data profiles for the metrics  $f_{\%}$  and  $x_{\%}$  are used, and a performance profile for RMS.

A slight improvement can be seen for the Corner Points - deterministic global solver combination (CP+DGS). The second combination, Corner Points - Maximin LHD (CP+LHD), has no obvious effect on the  $f_{\%}$  and  $x_{\%}$  metrics.

The RMS error is improved for both combinations, and since more points are sampled initially this seems reasonable. Once again the Maximin LHD design, and the combination (CP+LHD), performed better on the  $P_C$  problems.

**Constrained versions or not**

The results of the ordinary Maximin LHD and the constrained version are compared on the set of constrained problems  $P_C$ . Like before, only the  $N_2$  setting is used, since it outperforms the  $N_1$  setting. The extra effort of finding feasible points initially seem not to pay off as one might expect. Figure 6 does not show any significant improvement for the  $f\%$  and  $x\%$  metrics.



**Figure 6:** Constrained Maximin LHD versus standard Maximin LHD.

A possible reason for this is that although some points are infeasible, they still give information about the shape of the function. Since only feasible points are sampled by the CGO solvers, these initial infeasible points give extra stability to the surrogate model, compared to sampling only feasible points initially.

When considering the RMS metric for constrained problems, there are two ways to measure the error. One can look at the whole design space, like before, but it is also interesting to measure only the feasible space. As seen in the plots, these two results are in conflict. Using a fully feasible initial design naturally gives better RMS error when only considering the feasible design space, but not as good when measuring the whole design space.

## 6 Conclusions

The  $N_2$  setting performed better for all experimental designs, so this is definitely good. One could of course try to start with even more points, but since the CGO solvers are constructed in a way where each new point is chosen carefully by utilizing information from all the sampled points, this is probably not a good idea.

Finding a feasible experimental design with space filling capacity is not easy. The algorithm proposed in this paper generates an initial design with feasible points having the structure of a Maximin LHD. To see any real effect of a fully feasible experimental design, one must probably have test problems where a large area of the design space is infeasible. Most of the problems in  $P_C$  have large feasible areas and thus the effect is not as noticeable. Also, as the number of initial points  $N$  is typically a small part of the total number of sampled points, the effect is limited.

Sampling the corner points of the bounding box add a tremendous stability to the solvers, one could think of it as pinpointing the corners of the surface and therefore getting a more stable description of the boundary. This feature is important as it tends to help the solvers sample more interior points, which often helps the convergence.

The Maximin LHD approach is superior when looking at the RMS error. Combining this with the success of the Corner Point Strategy seemed like a promising idea, but unfortunately did not improve the  $f\%$  and  $x\%$  metrics as we had hoped.

There is no obvious winner since all the experimental designs work satisfactory. But since we consider costly functions, even small differences do matter. The combination of Corner Points and global solver performs very well compared to the other experimental designs, with robust results for all metrics.

## References

- [1] E. D. Dolan, J. J. Moré, and T. S. Munson: Optimality Measures for Performance Profiles. *Preprint ANL/MCS-P1155-0504* (2004).
- [2] H. M. Gutmann: A radial basis function method for global optimization. *Journal of Global Optimization* **19** (3), 201–227 (2001).
- [3] K. Holmström: An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization. *Journal of Global Optimization* **41**, 447–464 (2008).
- [4] K. Holmström and M. M. Edvall: January 2004, ‘CHAPTER 19: THE TOMLAB OPTIMIZATION ENVIRONMENT’. In: L. G. Josef Kallrath, BASF AB (ed.): *Modeling Languages in Mathematical Optimization*. Boston/Dordrecht/London.
- [5] D. R. Jones, C. D. Perttunen, and B. E. Stuckman: Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* **79**, 157–181 (1993).
- [6] D. R. Jones, M. Schonlau, and W. J. Welch: Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* **13**, 455–492 (1998).

- [7] D. R. Jones: DIRECT. *Encyclopedia of Optimization* (2001).
- [8] J. J. Moré and S. M. Wild: Benchmarking Derivative-Free Optimization Algorithms. *Preprint ANL/MCS-P1471-1207* (2007).
- [9] M. J. D. Powell: Recent Research at Cambridge on Radial Basis Functions. *New Developments in Approximation Theory*, 215–232 (2001).
- [10] E. Stinstra, D. den Hertog, P. Stehouwer, and A. Vestjens: Constrained Maximin Designs for Computer Experiments. *Technometrics*, **45**, 340–346 (2003).