

---

# Evolutionary Designs for Robust Parameter Design Experimentation

---

**Eduardo Santiago**

Dept. of Industrial Engineering  
Pennsylvania State University  
University Park, PA 16802

## Abstract

The following article presents the application of an evolutionary strategy to produce nearly optimal design matrices, containing the parameter settings to execute an experiment. The properties of such matrices significantly determine the precision of the optimization models used in Robust Parameter Design (RPD). The methodology presented allows the user to produce new experimental design matrices. Some of the innovative evolutionary designs obtained are presented and compared to the corresponding benchmarks.

## 1 INTRODUCTION

Robust Parameter Design (RPD) is the field that considers the optimization of an experiment by taking into account controllable and noise factors. The term RPD was coined by G. Taguchi, who introduced the concept of robustness. RPD deals with the optimization of a process in such a way that it remains insensitive to changes in the noise factors.

Controllable factors are considered to be variables that can be changed in the process to improve its performance. On the other hand, noise factors are those that can only be controlled in a laboratory. Examples of typical noise factors are wind, temperature and humidity.

Specifically, the RPD approach tries to minimize the sensitivity of a process to noise factors, subject to the satisfactory attainment of a target value in the response. The quality of the solution depends on the precision of the model that has been optimized. So, the main purpose in the rest of this article is to show how to get precise models. The emphasis will be on how to design matrices with nearly optimal properties.

In general, several design properties are desired. Typically designs with few runs, easy to use and with a good spatial distribution of the runs across the decision space are preferred. But regardless of this, we need to plan and perform our experiments. To execute the experiment, the settings for every run are arranged in a matrix. These specifications tell the experimenter how to change the factors on each experimental run. We denote this array as the design matrix  $\mathbf{D}$ . In practice, experimental designs have been implemented for two purposes:

1. Perform a multiple regression analysis and do some prediction (e.g. fit the experimental data to a linear, quadratic or cubic regression model).
2. Optimize a process according to some specific assumed model.

As will be shown later, the properties of the matrix  $\mathbf{D}$  determine how accurate our prediction will be for a particular region of the search space. A more accurate prediction will provide better estimates for our model; hence our optimization will be more reliable.

In order to choose the best design matrix  $\mathbf{D}$ , the user needs to provide the following inputs: 1) Number of runs, 2) Number of factors (controllable and noise) and 3) The model (linear, quadratic, RPD).

Designing experiments is ironically, something omitted in experimentation. This field has been intensively studied for more than 50 years and several standard designs have been developed for the most common applications. For a quadratic response surface model, Central Composite designs have been commonly used.

Standard designs for standard situations have saved the experimenter the burden of having to design a matrix. Consequently, these designs have been blindly and indiscriminately implemented in situations where

they should not be used. Such is the case of RPD and problems with constraints in the decision space.

The history of optimal designs goes back to Kiefer [12] who in 1959 introduced two of the most important performance measures for matrix design, D-efficiency ( $D_{\text{eff}}$ ) and G-efficiency ( $G_{\text{eff}}$ ). Currently, these measures are the standard way to compare two different designs.

In 1980, Taguchi [17] introduced the concept of noise factors. He developed the idea of robustness to changes in noise factors and with this RPD was born as a discipline. He defined the first designs for RPD, known as crossed arrays, but the large number of runs required by his designs quickly proved their inconvenience in practice. Since Taguchi’s revolutionary work, only a few authors including Lucas [1, 13], Box and Jones [5], and Myers et al. [16] suggested the use of single factor arrays as alternatives to crossed arrays.

But perhaps some of the best designs made with the RPD model in mind, correspond to the work of Borkowski and Lucas [4] who developed Mixed Resolution designs. More recently, Del Castillo [6] has shown how to generate designs for RPD using a new criterion, noise factor separation. His approach combines this new criterion with the power of Genetic Algorithms.

During the last 20 years, Simulated Annealing and GAs have found their way into the field of design of experiments. In 1986, Bohachevsky et al. [3] formulated a generalized simulated annealing procedure and used the algorithm to find D-optimal designs (matrices with maximum  $D_{\text{eff}}$ ). Inspired by this work, Haines [8], Meyer et al. [14] applied Simulated Annealing in different ways to solve a similar problem.

In 1992, Govaerts and Rubal [7] were the first to implement GAs to find D-optimal designs. Their work was improved by Montepiedra [15] in 1998, and more recently by Heredia-Lagner [10] in 2003. But none of their work was in the field of RPD.

### 1.1 AN ILLUSTRATIVE EXAMPLE

Consider the optimization of a chemical process where there are only two factors in play, temperature and concentration of one of the reactants. The objective is to choose the temperature and concentration levels that maximize the yield of the chemical process.

Assuming there is a quadratic relationship between the product yield and the controllable factors, a design for two factors is needed. The quadratic regression model is displayed next:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

This model can also be written in matrix form as  $Y = \beta_0 + \beta' \mathbf{x} + \mathbf{x}' \mathbf{B} \mathbf{x}$ . A standard design for this situation is the Central Composite Design (CCD) with eight runs which is shown in Fig. 1. The columns correspond to the settings of each one of the two factors  $x_1$  and  $x_2$ .

$$\mathbf{D} = \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \\ \sqrt{2} & 0 \\ -\sqrt{2} & 0 \\ 0 & \sqrt{2} \\ 0 & -\sqrt{2} \end{bmatrix}$$

Figure 1: Design Matrix For Two Factors

Note that the entries of the matrix  $\mathbf{D}$  are in coded units. In order to transform the matrix to the original units of each factor we apply Eq. 1:

$$\xi = d_{ij} \cdot (\xi_{\text{max}} - \xi_{\text{min}}) / 2 + (\xi_{\text{max}} + \xi_{\text{min}}) / 2, \quad (1)$$

where  $d_{ij}$  is the  $(i, j)^{\text{th}}$  entry of  $\mathbf{D}$ , and  $\xi_{\text{max}}$  and  $\xi_{\text{min}}$  are the maximum and minimum values in natural units for each factor.

Notice that when  $d_{ij} = 1$  we get that  $\xi = \xi_{\text{max}}$  and when  $d_{ij} = -1$ ,  $\xi = \xi_{\text{min}}$ . This implies, that the experimenter should carefully choose the values of  $\xi_{\text{max}}$  and  $\xi_{\text{min}}$  in such a way that when the entry is greater than 1 (or smaller than -1) the value in natural units is feasible.

After choosing the design matrix, it is transformed according to the assumed model. This matrix is called the matrix in model-form and is denoted with an  $\mathbf{X}$ . The  $\mathbf{X}$  matrix is used to evaluate the design; another important feature of this matrix is that it contains a column for every parameter in the assumed model.

Fig. 2 shows the transformation of the matrix  $\mathbf{D}$  in Fig. 1 according to the quadratic model presented. Observe that the 1<sup>st</sup> column corresponds to the intercept of the model, the 2<sup>nd</sup> and 3<sup>rd</sup> columns are exactly  $\mathbf{D}$ , the next column corresponds to the interaction term and it is the product of columns 2 and 3. Finally the last two columns correspond to  $x_1^2$  and  $x_2^2$ .

Section 2 reviews the basics of an RPD model, which is a generalization of the quadratic model presented in this section. To motivate this analysis, suppose that in the last example a very important factor was not considered, say humidity. If humidity turns out to be a significant factor, imagine the consequences of optimizing an incomplete model.

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -\sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & 0 & \sqrt{2} & 0 & 0 & 2 \\ 1 & 0 & -\sqrt{2} & 0 & 0 & 2 \end{bmatrix}$$

Figure 2: Matrix In Model Form

## 2 THE RPD MODEL

Early work in Robust Parameter Design includes the research done by Taguchi, Box and Jones, and Myers who proposed the following model.

Let  $\mathbf{x}' = [x_1, x_2, \dots, x_k]$  be the  $k$  controllable factors and let  $\mathbf{z}' = [z_1, z_2, \dots, z_r]$  be a vector of  $r$  noise variables; the model we assume is:

$$Y(\mathbf{x}, \mathbf{z}) = \beta_0 + \mathbf{x}'\boldsymbol{\beta} + \mathbf{x}'\mathbf{B}\mathbf{x} + \mathbf{z}'\boldsymbol{\gamma} + \mathbf{x}'\boldsymbol{\Delta}\mathbf{z} \quad (2)$$

where  $Y$  is the observed value of the response given a fixed value of the noise variables  $\mathbf{z}$ , and the corresponding values of the controllable factors  $\mathbf{x}$ .

The parameters of the model are  $\boldsymbol{\theta} = \{\beta_0, \boldsymbol{\beta}, \mathbf{B}, \boldsymbol{\gamma}, \text{ and } \boldsymbol{\Delta}\}$ . They are all estimated using the matrix  $\mathbf{X}$ ; hence, the properties of the model depend on the  $\mathbf{X}$  chosen. These properties include variance, the number of runs,  $D_{\text{eff}}$  and  $G_{\text{eff}}$ .

As mentioned before, this model is a generalization of the quadratic model. Without any noise variables taken into consideration, this model reduces to a quadratic model. The complete model is quadratic in the controllable factors and linear in the noise variables, in addition to the corresponding interactions between noise and controllable factors.

## 3 OPTIMAL DESIGN THEORY

Optimal Design Theory was originally developed by Kiefer [12] as a hardcore mathematical tool which practitioners found of no use at all. Years later, the scientific community started to understand Kiefer's theory and its applicability to Design of Experiments.

As we have iterated above, the choice of design has an impact in the quality of the prediction/optimization model we derive from  $\mathbf{X}$ . For instance, the variability of the coefficients in the regression problem depend on  $\mathbf{X}$ . We can show that  $\text{Var}(\boldsymbol{\theta}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$  is the variance-covariance matrix. Based on this, the

D-optimality criterion tries to achieve a spatial distribution of points in the decision space in such a way that the coefficients' variances are minimized. It can also be shown that by maximizing  $|\mathbf{X}'\mathbf{X}|$ , we minimize  $|(\mathbf{X}'\mathbf{X})^{-1}|$ .

Since the variability of the coefficients ( $\boldsymbol{\theta}$ ) of the optimization model depend on  $(\mathbf{X}'\mathbf{X})^{-1}$ , it makes sense to minimize this determinant.

D-efficiency may be defined in several ways, but the most common approach to compare two different designs with different number of runs is the following:

$$D_{\text{eff}} = |\mathbf{M}(\xi)| = \frac{|\mathbf{X}'\mathbf{X}|}{N^p} \quad (3)$$

where  $N$  is the number of runs and  $p$  the number of model parameters. We refer to  $\mathbf{M}(\xi)$  as the moment matrix of the design  $\xi$ .

Another way of defining  $D_{\text{eff}}$  is by using the following relation,  $D_{\text{eff}} = (|\mathbf{M}(\xi)| / \max |\mathbf{M}(\xi^*)|)^{1/p}$ , where the denominator is the maximum determinant over all possible moment matrices.

The second criterion used is G-efficiency. This is a measure of the prediction variance,  $v(\mathbf{x})$ , within the experimental region. Once again, the design chosen will determine the variance in the predictions made with the obtained model. G-efficiency is defined as follows:

$$G_{\text{eff}} = \frac{p}{\max_{\mathbf{x} \in R} v(\mathbf{x})} = \frac{p}{\max_{\mathbf{x} \in R} N\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}} \quad (4)$$

where  $p$  is the number of parameters in the model, and  $\mathbf{x}$  is any of the rows of the  $\mathbf{X}$  matrix.

Unfortunately, optimizing both criteria is not an easy task. The D-optimality criterion tries to maximize the volume spanned by the columns of  $\mathbf{X}$ ; while, G-optimality tries to distribute the points homogeneously across the region of experimentation. Therefore in the neighborhood of fractions of the decision space where no runs are allocated, the predictive variance will be expected to be high.

## 4 IMPLEMENTING CMA-ES: A SINGLE OBJECTIVE PROBLEM

The main goal is to implement an evolutionary strategy to design new matrices for the RPD model displayed in the last section. The heuristic selected is the Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) developed by Hansen and Ostermeier [9]. The objective is to develop competitive matrices that

outperform the designs that have been implemented in Robust Parameter Design for the last two decades.

#### 4.1 DESCRIPTION OF THE ALGORITHM

CMA-ES is a self-adaptive algorithm developed by Hansen et al. for continuous parameter optimization of nonlinear functions. In essence, the algorithm generates the offspring from a “learning” Normal distribution that adjusts its parameters to better fit the best individuals from each generation.

The two main elements of this Normal distribution are its mean  $\mathbf{m}^{(g)}$  and its covariance matrix  $\Sigma$ . The mean is a linear combination (weighted average) of the best solutions from each generation, and represents the centroid of the search; on the other hand,  $\Sigma$  determines how broad the search should be.

During each generation, the algorithm adapts the covariance matrix of the Normal distribution, estimating the inverse Hessian matrix in such a way that it speeds up the convergence of the algorithm towards an improving direction; this adaptation allows the algorithm to align the search for optimum to the contour lines of the objective function.

For more details on the algorithm, the reader may refer to Hansen 2004 [11], as well as the Tutorial for the MATLAB version of his code.

#### 4.2 PROBLEM REPRESENTATION (QUADRATIC MODEL)

The representation of this problem considers three different spaces: i) decision space, ii) model-form space and iii) objective space. This is done because the variables are contained in the matrix  $\mathbf{D}$ , but the model-form  $\mathbf{X}$  is required to compute the fitness of each solution. Another issue is that CMA-ES’ solutions are vectors, and what we need is a matrix (See Fig. 3). In order to use CMA-ES, each matrix is “vectorized”, so an  $n \times p$  matrix is transformed back and forth to a  $np \times 1$  vector.

The decision variables are the entries of the design matrix  $\mathbf{D}$ ; so the design displayed in Fig. 1 exhibits a total of 16 variables ( $8 \text{ runs} \times 2 \text{ factors}$ ). Fig. 3 shows that for every function evaluation matrix  $\mathbf{D}$  is transformed to matrix  $\mathbf{X}$  to map the decision space into the objective space.

The design matrix  $\mathbf{D}$  is the final output of this optimization process.  $\mathbf{D}$  is an  $n \times k$  matrix where  $k$  is the number of factors in the experiment, and  $n$  the number of available runs. Once this is given, we need to specify the model. With these three pieces of in-

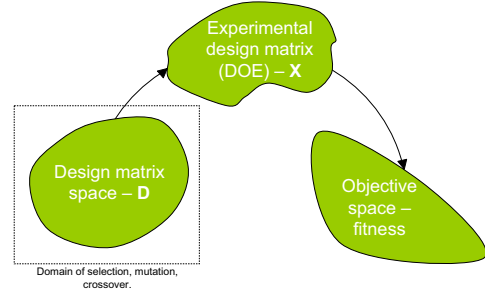


Figure 3: Decision And Objective Spaces

formation we construct the matrix  $\mathbf{X}$  that is used to evaluate the objective function.

Considering a quadratic regression model, the formulation is the following:

$$\text{Maximize } f = |\mathbf{X}'\mathbf{X}|$$

Subject to:

$$G_{\text{eff}} \geq G_{\text{min}} = 0.95$$

$$\sum_{j=1}^p \left( \sum_{i=1}^n d_{ij} \right)^2 = 0.$$

Or equivalently as a single objective minimization problem:

$$f = -|\mathbf{X}'\mathbf{X}| - w_1 (G_{\text{eff}} - 0.95) + w_2 \sum \left( \sum d_{ij} \right)^2 \quad (5)$$

where the two constraints are introduced to the objective function with penalties  $w_1$ ,  $w_2$  and the  $d_{ij}$  are the  $n \times k$  decision variables of the matrix  $\mathbf{D}$ .

Notice that the first constraint tries to achieve G-optimality, while the second is a balance constraint. The 0.95 was chosen to pressure the algorithm to provide almost G-optimal designs.

Having a constrained single objective problem is justified because the  $|\mathbf{X}'\mathbf{X}|$  criterion should be the main driver of the optimization and the constraints should only refine the solutions provided by the  $|\mathbf{X}'\mathbf{X}|$  term. Recall that by minimizing  $-|\mathbf{X}'\mathbf{X}|$ , we maximize D-efficiency.

In addition, constraints were incorporated to derive designs with better properties, high  $G_{\text{eff}}$  and small number of levels. Because of the scale of the terms in the objective function, the optimization is driven primarily by the  $|\mathbf{X}'\mathbf{X}|$  term.

### 4.2.1 Parameter Setup

The default parameters of CMA-ES seemed to work well across several problems solved. Since we are not trying to solve a problem with unique features, i.e. same number of runs ( $n$ ) and factors ( $k$ ), optimizing the parameters makes no sense.

Instead, focus was centered at computing the penalty weights of the objective function. Notice the values displayed on Table 1 are all clearly on a different scale. In addition, the maximum value of the determinant will be unknown in most of the cases analyzed.

Table 1: Minimum And Maximum Values For The Objective Function Terms.

| Objective terms                   | Minimum | Maximum       |
|-----------------------------------|---------|---------------|
| $ \mathbf{X}'\mathbf{X} $         | 0       | Max. problem  |
| $G_{\text{eff}} - G_{\text{min}}$ | -0.95   | 0.05          |
| $\sum (\sum d_{ij})^2$            | 0       | $p \cdot n^2$ |

To scale the terms of the objective function, perform the following steps:

1. Define the objective function as  $f = -|\mathbf{X}'\mathbf{X}|$  and optimize this without any constraints.
2. Once we have the maximum value ( $\det_{\text{max}}$ ), scale the penalty weights as  $w_1 = \det_{\text{max}}/0.05$  and  $w_2 = \det_{\text{max}}/(p \cdot n^2)$ .

The convergence criterion used was to stop the algorithm when the changes in the best solution ( $\mathbf{x}$ ) from generation to generation were smaller than  $\epsilon$ .

### 4.2.2 Results

For two factors, the design displayed in Fig. 4 was found; notice that the eight external points are lying in a circle with the center runs in the origin. This makes sense, because in this situation the model is symmetric. Therefore, the algorithm finds a symmetric design to fit the corresponding quadratic model.

Figure 5 displays the variance of the regression model obtained with the EVO1 design. The  $y$  axis displays the variance. On average the variance of the design is below  $p$  and only the maximum variance is beyond  $p$  after moving 1.2 units away from the origin.

The comparison of this design with some of the benchmarks previously established is summarized in Table 2. The second column corresponds to the total number of runs and the number between parenthesis is the center runs included in the total. The determinant of the

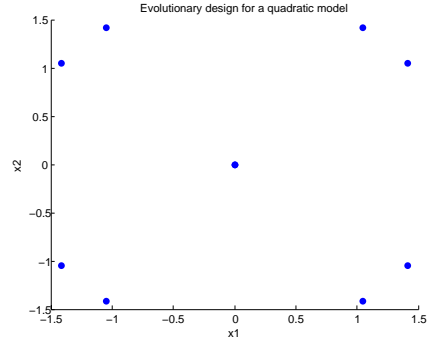


Figure 4: Evolutionary Design For  $k=2$  Factors

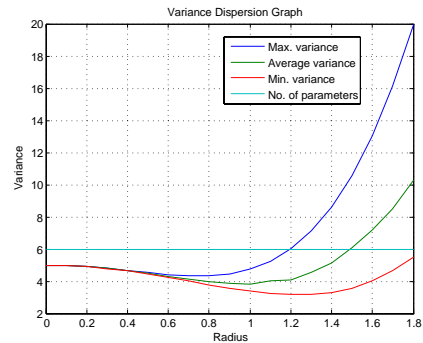


Figure 5: Variance Dispersion Graph For EVO1

moment matrix is displayed in the third column, so anyone can replicate our results. From the third column, D-efficiencies were computed.

Table 2: Design Comparison On A Cuboidal Region.

| Design       | Runs  | $ M(\xi) $            | $D_{\text{eff}}$ | $G_{\text{eff}}$ |
|--------------|-------|-----------------------|------------------|------------------|
| $3^2$ design | 9(1)  | $9.8 \times 10^{-3}$  | 100              | 82.76            |
| Composite    | 9(1)  | $2.41 \times 10^{-4}$ | 53.92            | 66.67            |
|              | 10(2) | $2.56 \times 10^{-4}$ | 54.47            | 96               |
| EVO1         | 9(1)  | $2.5 \times 10^{-3}$  | 79.64            | 66.67            |
|              | 10(2) | $2.6 \times 10^{-3}$  | 80.16            | 96               |

The best solutions are given by the  $3^2$  and the EVO1 designs. Clearly, they outperform the Central Composite design for two factors. We have beaten the benchmark and created a new design, which has good optimality properties, a small number of levels and a desired symmetry.

A reliability analysis using 100 random seeds,  $10k+100$  for  $k = 1, 2, \dots, 100$ , shows convergence to the exact same solution in 80% of the cases.

### 4.3 PROBLEM FORMULATION(RPD MODEL)

Once again consider the model displayed in Eq. 2. The noise parameters ( $\gamma$ ,  $\Delta$ ) in the model are crucial in the optimization of the RPD model.

It can be shown that in order to make a process insensitive to noise, levels in the controllable factors are chosen to minimize the variance of the process. This variance can be computed from:

$$\text{Var}_{\mathbf{z},\epsilon} [Y(\mathbf{x}, \mathbf{z})] = (\gamma + \Delta' \mathbf{x})' \Sigma_{\mathbf{z}} (\gamma + \Delta' \mathbf{x}) + \sigma^2 \quad (6)$$

From Eq. 6 observe that the variance is a function of the controllable variables  $\mathbf{x}$ , and not of the noise variables.

To achieve the minimum there must exist some correlation between the Noise Effects (N), or  $\gamma$ , and the Noise-Controllable interactions (N×C), or  $\Delta$ . In other words N and N×C should not be orthogonal; otherwise we can not select levels of  $\mathbf{x}$  that minimize the term  $(\gamma + \Delta' \mathbf{x})$ . Similarly, there has to be non-orthogonality between the N×C interaction terms.

These constraints will be handled through a penalty function, just like before. The formulation of the problem is:

$$\text{Min.} \quad -|X'X| - w_1 (G_{\text{eff}} - 0.95) + w_2 \sum \left( \sum d_{ij} \right)^2$$

Subject to:

$$\begin{aligned} \Psi &> 0, \Phi > 0 \\ -1 &\leq d_{ij} \leq 1 \end{aligned}$$

where  $\Psi$  is the sum of squares of the elements in the matrix  $(\mathbf{X}'\mathbf{X})^{-1}$  corresponding to the orthogonality of the N and N×C terms. A sum of squares close to zero indicates that the matrix is nearly orthogonal. Similarly,  $\Phi$  is equal to the sum of squares of the N×C terms. To create a design in coded units, the decision variables  $d_{ij}$  are constrained as shown in the formulation.

To include the additional constraints in the objective function we consider the weights  $w_3$  and  $w_4$ . With this scheme, scaling issues will also arise.

#### 4.3.1 Parameter Setup

Similarly to what was done in Section 4.2.1, the default parameters of CMA-ES are utilized. For the same reason as before, only the weights of the objective function were considered. Only this time some exact D-optimum values are available in Borkowski [2].

Several issues were observed when the new constraints were weighted accordingly to the maximum determinant (D-optimum values). It turned out that when the same scale was used for each term, the optimization was too noisy as each term carried the optimization towards a *different* path. Therefore, a scaled objective function only on the first three terms, that considers in addition the satisfaction of the constraints was implemented.

#### 4.3.2 Results

Results were obtained for one of the simplest cases, two controllable factors and one noise variable. Unfortunately, in the literature no one has ever published results for this case. Our designs had 20 runs (plus two center points) and high optimality features, such as  $D_{\text{eff}} = 99.3\%$  and  $G_{\text{eff}} = 91.34\%$ . There are currently no benchmarks for this case, but still the properties of the design are notoriously very good.

Table 3 includes one of our best designs and compares it with the corresponding benchmarks. The case analyzed takes into consideration two controllable and two noise factors, i.e.  $k=2$ ,  $r=2$ .

Table 3: RPD Design Comparison On A Cuboidal Region ( $k=2$ ,  $r=2$ ).

| Design       | Runs  | $ M(\xi) $            | $D_{\text{eff}}$ | $G_{\text{eff}}$ |
|--------------|-------|-----------------------|------------------|------------------|
| CMR 4A*      | 23(3) | 0.0046                | 100              | 88.81            |
| Hoke D6      | 22(3) | $1.06 \times 10^{-4}$ | 73.11            | 61.02            |
| Box-Behnken  | 27(3) | $1.63 \times 10^{-7}$ | 42.63            | 78.05            |
| Hybrid 416A* | 19(3) | $2.5 \times 10^{-3}$  | 93.32            | 67.07            |
| Notz         | 18(3) | $7.1 \times 10^{-5}$  | 70.70            | 70.67            |
| k2r2n18      | 18(-) | 0.0022                | 94.16            | 76.19            |
| EVO2-2       | 18(2) | 0.00202               | 93.48            | 91.16            |

The designs marked with a star are not directly comparable to the other designs since they contained runs beyond the experimental region (beyond  $\pm 1$ ). This obviously increases the determinant; so their apparent good properties are not a consequence of an optimal spatial distribution of their runs.

To illustrate this, the CMR 4A design was scaled into a hypercube. As a result, its  $D_{\text{eff}}$  was reduced to 35.47%!

The methodology presented consistently provides good designs that have both high D and G efficiencies. In addition, notice that this has been accomplished with only 18 runs. In industries where experimentation is expensive, this might be a crucial decision factor.

Further results are presented in Table 4. Once again,

the CMR 5A design has points outside the hypercube, and therefore its superiority is only apparent. With 28 runs, the design proposed EVO3-2 achieves better properties.

Table 4: RPD Design Comparison On A Cuboidal Region ( $k=3, r=2$ ).

| Design  | Runs  | $ M(\xi) $            | $D_{\text{eff}}$ | $G_{\text{eff}}$ |
|---------|-------|-----------------------|------------------|------------------|
| CMR 5A* | 25(3) | $5.2 \times 10^{-5}$  | 93.17            | 82.82            |
| Notz    | 24(3) | $1.31 \times 10^{-7}$ | 66.80            | 80.65            |
| Hoke    | 29(3) | $1.69 \times 10^{-9}$ | 52.47            | 67.56            |
| k3r2n24 | 24(-) | $7.05 \times 10^{-7}$ | 73.36            | 75.0             |
| EVO3-2  | 28(2) | $3.82 \times 10^{-5}$ | 91.58            | 76.42            |
|         | 30(2) | $6.66 \times 10^{-5}$ | 94.45            | 78.56            |

## 5 CONCLUSIONS

Traditionally the problem of designing an experiment is avoided by the practitioner due to the existence of a vast variety of designs for several kinds of models. Catalogues of design matrices may be easily found.

However, there is still a lot of work to be done. Nice theoretical results have been derived, but the work done previously has been limited to designs with a finite support (specific points in the decision space). Also, most of the designs currently used are modifications/adaptations of designs that were initially conceived with a different model in mind.

The continuous nature of the problem and the infinite number of designs have limited the analysis of statisticians and scientists to finite search spaces, constraining the potential of the decision space. The lack of computational power was another important obstacle, but now all of this can be easily overcome.

Take for instance, the quadratic model in two dimensions which is a well-studied problem. Using CMA-ES (a continuous search heuristic) we derived a new design with good properties that may be easily implemented in practice.

So far, no one has included the G-optimality criterion in their optimization procedure. This is a strength in particular of our procedure, which includes center runs a priori and optimizes with this taken into account.

Some concerns about using CMA-ES arose because of the multimodality nature of our objective function, especially in high dimensions (200 or more variables). This was noticed while optimizing the penalty weights of the objective function; small changes in the weights, drastically provided different results.

Another concern is the implementation of constraints in the optimization, since every solution derived from the Normal distribution needs to be feasible. Especially in high dimensional problems, solutions had to be brute-forced until feasibility was reached. This slows down the optimization as the number of variables involved increases.

There is also a limit in the number of decision variables considered, as this quantity grows geometrically as the number of factors (controllable and noise) and runs increase. The number of decision variables for a particular problem is computed from the following equation:

$$\text{Decision variables} = \frac{n[(k+1)(k+2) + 2r(k+1)]}{2}$$

Hence, for problems with a total of more than eight factors, CMA-ES will not provide satisfying solutions. Nevertheless, in practice, most experimentation is done for a small number of factors.

## References

- [1] *Achieving a Robust Process Using Response Surface Methodology*. Proceeding of the American Statistical Association, Sesquicentennial Invited Paper Sessions, 1989.
- [2] J.J. Bhorkowski. Optimal linear-quadratic (lq) model-based designs.
- [3] Ihor O. Bohachevsky, Mark E. Johnson, and Myron L. Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28(3):209–217, 1986.
- [4] J.J. Borkowski and J.M. Lucas. Design of mixed resolution for process robustness studies. *Technometrics*, 39:63–70, 1997.
- [5] G. E. P. Box and S. Jones. Designing products that are robust to the environment. Technical Report 56, 1990.
- [6] E. Del Castillo, M.J. Alvarez, L. Ilzarbe, and E. Viles. A new design criterion for robust parameter experiments. December 2005.
- [7] B. Govaerts and R. P. Sanchez. Construction of exact d-optimal designs for linear regression models using genetic algorithms. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32:153–174, 1992.
- [8] L.M. Haines. The application of the annealing algorithm to the construction of exact optimal designs for linear regression models. *Technometrics*, 29:439–447, 1987.

- [9] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [10] A. Heredia-Lagner, W.M. Carlyle, D.C. Montgomery, C.M. Borrer, and G.C. Runger. Genetic algorithms for the construction of d-optimal designs. *Journal of Quality Technology*, 35:28–46, 2003.
- [11] S. Kern, N. Hansen, S. D. Muller, D. Buche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms – a comparative review. 2004.
- [12] J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 21(2):272–319, 1959.
- [13] J.M. Lucas. How to achieve a robust process using response surface methodology. *Journal of Quality Technology*, 26:248–260, 1994.
- [14] R.K. Meyer and C.J. Nachtstein. Simulated annealing in the construction of exact optimal design of experiments. *American Journal of Mathematical and Management Science.*, 8:329–359, 1988.
- [15] G. Montepiedra, BD. Myers, and A. B. Yeh. Application of genetic algorithms to the construction of exact d-optimal designs. *Journal of Applied Statistics*, 25(6):817–826, 1998.
- [16] Raymond H. Myers, Andre I. Khuri, and Geoffrey Vining. Response surface alternatives to the taguchi robust parameter design approach. *The American Statistician*, 46(2):131–139, 1992.
- [17] G. Taguchi and Y. Wu. Introduction to off-line quality control. Technical report, Central Japan Quality Control Association, Nagoya, Japan, 1980.