Chapter 1

Computer Experiment Designs via Particle Swarm Optimization

Erin Leatherman, Angela Dean, and Thomas Santner

Abstract This paper illustrates the use of the Particle Swarm Optimization (PSO) algorithm in obtaining optimal designs for computer experiments. A small example is given to illustrate the steps of PSO. The quality and prediction ability of "minimax" space filling designs obtained using PSO with different numbers of "swarm particles" and different numbers of iterations are examined .

1.1 Computer Experiments and Emulators

Computer experiments are used widely in diverse research areas such as engineering, biomechanics, and the physical and life sciences. Computer experiments use *computer simulators* as experimental tools to provide outputs $y(x)$ at specified design input points x, where a computer simulator is the computer implementation of a mathematical model that describes the relationships between the input and output variables in the physical system. Computer experiments can be especially attractive when physical experiments are infeasible, unethical, or "costly to run."

For fast running codes, the output response surface can be explored by evaluating (running) the simulator at a set of inputs $x = (x_1, \ldots, x_k)$ that are dense in the space of possible inputs, $\mathscr{X}$. For slow-running codes, an approximator (also called an "emulator" or "metamodel") is often sought for the simulator output $y(x)$; such metamodels allow, for example, the detailed (approximate) exploration of the output surface.

Erin Leatherman
West Virginia University, PO Box 6330, Morgantown, WV 26506, USA, e-mail: erl@stat.wvu.edu

Angela Dean
University of Southampton, Southampton, SO17 1BJ, UK, e-mail: dean.9@osu.edu

Thomas Santner
The Ohio State University, Columbus, OH 43210, USA, e-mail: santner.1@osu.edu

One rapidly-computable class of emulators for deterministic computer simulator output $y(x)$ assumes that $y(x)$ can be modeled as a realization of a Gaussian Stochastic Process $Y(x)$ (GaSP). In this paper, the input space $\mathscr{X}$ for the k inputs is rectangular and, unless otherwise stated, scaled to $[0,1]^k$. The GaSP models are assumed to take the form

$$Y(x) = \sum_{\ell=0}^{p} f_\ell(x)\beta_\ell + Z(x) \;=\; f'(x)\beta \;+\; Z(x)\,, \tag{1.1}$$

where $f'(x) = (f_1(x),\ldots,f_p(x))$ is a vector of known regression functions, $\beta = (\beta_1,\ldots,\beta_p)$ is a vector of p unknown regression coefficients and $Z(x)$ is a zero-mean, stationary Gaussian stochastic process on $\mathscr{X}$ with covariance

$$\mathrm{Cov}\left(Z(x_u),Z(x_v)\right) = \sigma_Z^2 \times R(x_u - x_v \mid \rho) = \prod_{j=1}^{k} \rho_j^{4\left(x_{uj}-x_{vj}\right)^2}\,, \tag{1.2}$$

where x_{uj}, x_{vj} are the j^{th} elements of input points $x_u, x_v \in \mathscr{X}$, $j = 1,\ldots,k$, $\rho = (\rho_1,\rho_2,\ldots,\rho_k)'$, and $\rho_j \in [0,1]$ is the correlation between two outputs whose x_u and x_v differ *only* in the j^{th} dimension by $|x_{uj} - x_{vj}| = 1/2$, which is *half their domain*.

The design for the computer experiment is denoted by an $n \times k$ matrix $X \in \mathscr{D}(n,k)$ whose ith row is defined by the ith design point $x_i' = (x_{i1},\ldots,x_{ik})$; $\mathscr{D}(n,k)$ denotes the class of all designs with n runs, k input variables, and input space $\mathscr{X}$.

Let $y^n = (y(x_1),\ldots,y(x_n))$ denote (training) data to be used for estimating the simulator output $y(x_0)$. When β is *unknown*, but the correlation parameters ρ are *known*, the best linear unbiased predictor (BLUP) of $y(x_0)$, can be shown to be $\widehat{y}(x_0) = f_{x_0}'\widehat{\beta} + r_{x_0}'R^{-1}(y^n - F\widehat{\beta})$, where $\widehat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y^n$ (see for example [18]). Here $\widehat{\beta}$ is generalized least squares estimator of β, F is an $n \times p$ matrix with uth row $f'(x_u)$, R is an $n \times n$ matrix whose $(u,v)^{th}$ element is $R(x_u - x_v \mid \rho)$, and $r_{x_0}' = (R(x_0 - x_1 \mid \rho),\ldots,R(x_0 - x_n \mid \rho))$ is an $1 \times n$ vector.

1.2 Design Criteria

Space-filling designs are popular choices for computer experiments when fitting GaSP models, (see, for example, [9] and [3]). Space-filling criteria ensure that the entire input space is sampled by preventing design points from being “close” together.

Two important space-filling criteria are the maximin (Mm) and the Average Reciprocal Distance (ARD) criteria. The *Mm* criterion specifies that a design $X_{Mm} \in \mathscr{D}(n,k)$ that maximizes the minimum interpoint distance

$$\min_{x_u,x_v \in X} q_z(x_u,x_v) \tag{1.3}$$

is optimal where $q(x_u, x_v)$ is the distance between x_u and x_v. Here and below, we use Euclidean distance, but other metrics could equally well be used.

The *ARD* criterion is specified by a given set $J \subset \{1, \ldots, k\}$ of sub-dimensions over which the distances are to be computed (e.g. [1], [14]). A design X_{ARD} is ARD-optimal with respect to J if it minimizes

$$av_z(X) = \frac{1}{\binom{n}{2} \sum_{j \in J} \binom{k}{j}} \sum_{j \in J} \sum_{\ell=1}^{\binom{k}{j}} \sum_{x_u^\star, x_v^\star \in X_{\ell j}} \left[\frac{j^{1/z}}{q(x_u^\star, x_v^\star)} \right] \tag{1.4}$$

where $X_{\ell j}$ is the ℓth subspace of X having dimension j, $x_u^\star$ and $x_v^\star$ are the projections of x_u, x_v onto $X_{\ell j}$, and $q(x_u^\star, x_v^\star)$ is the distance between $x_u^\star$ and $x_v^\star$.

For prediction, [12] showed that process-based design criteria produce better designs than do space-filling criteria (see also [16]). Process-based criteria involve the chosen emulator rather than geometric properties. Such criteria include the *minimum integrated mean squared prediction error (IMSPE)* ([17]), maximum entropy ([20]), and maximum expected improvement [4]. For example, for a given ρ, σ_Z^2 and predictor $\widehat{y}(\cdot)$, the IMSPE-optimal design $X_I \in \mathscr{D}(n,k)$ minimizes

$$\text{IMSPE}^*(X \mid \rho) \ = \ \frac{1}{\sigma_Z^2} \int_{\mathscr{X}=[0,1]^d} E\left[\left(\widehat{y}(w) - Y(w)\right)^2 \mid \rho, \sigma_Z^2 \right] \mathrm{d}w \tag{1.5}$$

where the expectation is over the joint distribution of $(Y(w), Y^n)$. If the values of the correlation parameters ρ cannot be specified in advance of the experiment but a distribution $\pi(\rho)$ of possible values is approximately known, an alternative criterion is to minimize the IMSPE weighted by $\pi(\rho)$, as in [12]. The examples in [12] use $\pi(\rho) = \prod_{j=1}^{k} \pi(\rho_j)$ and independent Beta distributions for $\pi(\rho_1), \ldots, \pi(\rho_k)$. For given $\pi(\rho)$, a design X_A that minimizes *weighted* (averaged) *integrated mean squared prediction error*:

$$\text{AIMSPE}^*(X) = \int_{[0,1]^k} \text{IMSPE}^*(X \mid \rho)\, \pi(\rho)\, \mathrm{d}\rho \tag{1.6}$$

is said to be AIMSPE*-optimal.

For each of the four criteria (1.3)–(1.6), Figure 1.1 shows approximate optimal-designs with $k = 2$ inputs and $n = 20$ runs constructed using PSO followed by a quasi-Newton optimizer. The PSO used is described in Section 1.3; it took $N_{\text{des}} = 4nk = 160$ "particles" and $N_{\text{its}} = 8nk = 320$ "iterations". Maximin designs tend to have design points on the boundary of the input region; as seen in the top left of Figure 1.1, this is true in this example where 12 of the 20 points are on, or close to, the boundary. The minimum distance between the points in this design is 0.2729, which is close to the maximum achievable minimum interpoint distance of 0.2866 (`http://www.packomania.com/`).

The minimum ARD design, shown in the top right of Figure 1.1, used $J = \{1, 2\}$ so that the ARD was calculated as an average over the 2-dimensional input space and its two 1-dimensional projections. The resulting design has more uniformly spread

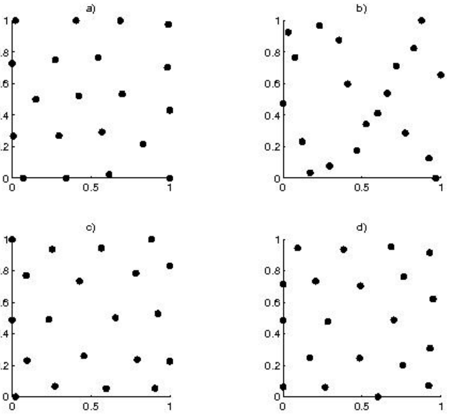

Fig. 1.1 Approximate optimal designs for $k = 2$ inputs, $n = 20$ runs, using criteria (1.3)–(1.6) (criterion value in parentheses): Panel (a) Mm design (0.2729); Panel (b) min ARD design (2.2096); Panel (c) min IMSPE* design (2.2827×10^{-6}); Panel (d) min AIMSPE* design (4.1192×10^{-4})

points in the 1-dimensional subspaces than the maximin design, but at the cost of less uniformity in the 2-dimensional space. To obtain A more uniform distribution of 2-dimensional points would arise if $J = \{2\}$ rather than $J = \{1,2\}$.

For the minimum IMSPE* design, shown in the bottom left of Figure 1.1, the correlation parameters, ρ_1 and ρ_2 were set to 0.75 (see [17, 12]). For the minimum AIMSPE* design, $\pi(\rho)$ took each of ρ_1 and ρ_2 to be Beta(37.96,37.96) (found by [12] to perform well for prediction). Although, visually, both of these designs seem to have more uniform 2-dimensional spread than the maximin design, their minimum interpoint distances (MIPDs) are, respectively, 0.1954 and 0.2043, about 75% of the MIPD 0.2729 for the Mm design. For more information on the prediction performances of space-filling, IMSPE*-optimal, and AIMSPE*-optimal designs for different parameter values, see [12].

1.3 Particle Swarm Optimization

Many optimization methods have been suggested in the literature; see, for example, [7] and [21] for surveys. Some methods are most effective in local searches of the input space; for example, gradient-based methods such as the Newton and Quasi-Newton algorithms (see, for example, [7]). Other optimization methods emphasize a global search over the entire input space; for example, genetic algorithms [8], simulated annealing [11], and particle swarm optimization ([10]). Some methods, such as simulated annealing ([11]) and mesh adaptive direct search ([2]), have iteration-dependent parameters that enable them to search both globally and locally.

PSO algorithms were introduced in [10]. PSO algorithms have had many applications including the computation of optimal designs for physical experiments using classical criteria ([5, 6]) and by [12] to find optimal designs for computer experiments. [12] used the output of PSO to identify starting points for a gradient-based, constrained non-linear optimizer (`fmincon.m` from the MATLAB Optimization toolbox).

In more detail, to find an $n \times k$ optimal design, PSO starts with a number (N_{des}) of $n \times k$ initial designs $X_1, \ldots X_{N_{\text{des}}}$. Each X_i is reshaped (column-wise) into an $nk \times 1$ vector $z_i^1 = vec(X_i)$, called the i^{th} *particle*, $i = 1, 2, \ldots, N_{\text{des}}$. To ensure wide exploration of the nk-dimensional input space, the initial set of N_{des} particles can be selected as an $N_{\text{des}} \times nk$ approximate Mm Latin Hypercube Design.

At iteration t, $t = 1, 2, \ldots, N_{its}$, every particle z_i^t is "updated," using (1.7) to z_i^{t+1}, and then evaluated under the criterion of interest. The update requires the following notation. At iteration t, let g^t denote that particle $z_i^t \in \left\{ z_i^{t^\star} \mid i = 1, \ldots, N_{\text{des}}; t^\star \leq t \right\}$ that produces the *global best value of the criterion of interest*. Similarly, for each particle i, let p_i^t denote that $z_i^t \in \left\{ z_i^{t^\star} \mid t^\star \leq t \right\}$ having *particle best value of the criterion*. Then

$$z_i^{t+1} = z_i^t + v_i^{t+1}, \tag{1.7}$$

where $v_i^{t+1} = \theta v_i^t + \alpha {\varepsilon_1}_i^t \circ (g^t - z_i^t) + \beta {\varepsilon_2}_i^t \circ (p_i^t - z_i^t)$, $\circ$ is elementwise product of vectors, ${\varepsilon_1}_i^t$ and ${\varepsilon_2}_i^t$ are independent random vectors whose elements are independent Uniform[0,1], α and β are weights put on the step toward the global- and personal- best positions respectively, $\theta \in [0, 1]$ is the 'inertia' parameter, and $v_i^t \in [-0.25, 0.25]$.

The examples in Section 1.4 took $\alpha = \beta = 2$, $\theta = 0.5$, and initial velocity $v_i^1 = 0_{nk}$, as recommended by [10] and [21]. There we describe the results of PSO in searching for a Mm design with $(n, k) = (60, 6)$ for different numbers of "particles" and different numbers of iterations, with and without final local optimization. The use of PSO for obtaining IMSPE*-optimal and AIMSPE*-optimal designs is described in [12].

We now illustrate the working of PSO in a "toy" example with $(n, k) = (1, 2)$ so that each z_i^t is a $(nk =)$ $2-$dimensional vector. Figure 1.2 shows $N_{\text{des}} = 8$ z_i^t positions after $N_{\text{its}} = 1, 2, 3, 5, 10, 24$ iterations, together with the (unknown) contours of the design criterion, which is to be *minimized*. The optimal value is 1.0116, located at [0.1215, 0.8240].

Panel (a) of Figure1.2, (labelled "Iteration $t = 1$") shows the initial particle starting locations, chosen as a maximin LHD. The particle located in the top left corner of the scatterplot corresponds to the design that has the minimum criterion value $(= 3.6087)$ when $t = 1$, so this location is g^1. At Iteration $t = 2$, the particles have taken one step towards g^1 plus a random perturbation, using (1.7). The stars denote the current particle positions z_i^2, and the open circles denote the starting positions which form the current particle-best p_i^2. An evaluation of the criterion values of the designs corresponding to the new particle positions, z_i^2, $i = 1, \ldots, 8$, finds that the global best design is remains unchanged, i.e., $g^2 = g^1$. At $t = 3$, each particle i $(= 1, \ldots 8)$ moves from z_i^2 towards a weighted combination of the global best par-

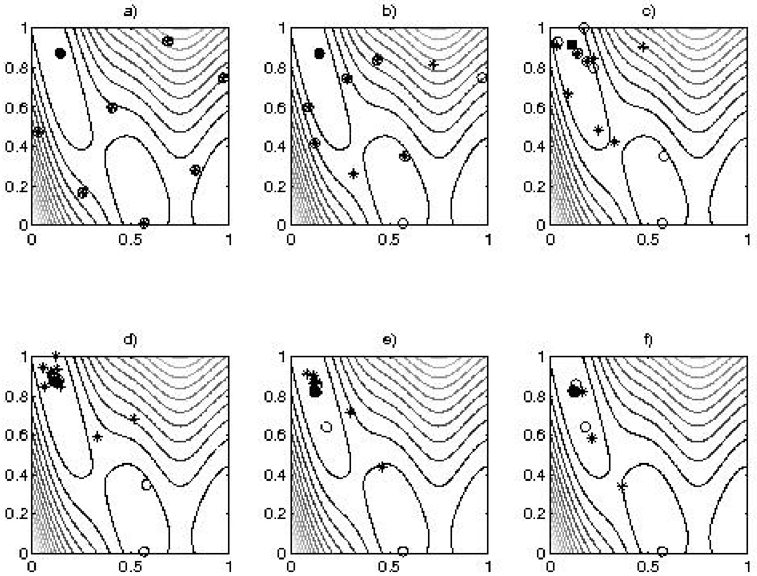

Fig. 1.2 Panel (a) Iteration 1, min fnc value = 3.6087; Panel (b) Iteration 2, min fnc value = 3.6087; Panel (c) Iteration 3, min fnc value = 3.6087; Panel (d) Iteration 5, min fnc value = 2.0557; Panel (e) Iteration 10, min fnc value = 1.1776; Panel (f) Iteration 24, min fnc value = 1.0118

ticle position, g^2, and its personal best position p_i^2 resulting in z_i^3. Again the global best position is unchanged so that $g^3 = g^2 = g^1$. Some of the particle-best positions (open circles) have changed, such as that closest to the bottom left of the picture, while others remain the same, such as that one on the bottom border. By iteration $t = 5$ (Panel (d)), most of the particles are closing in on the optimum, and one particle has found a better location than g^3 with a smaller criterion value of 2.0557. This implies that the previous best particle, which had not moved in previous iterations, will now start to move towards the new best position.

By iteration $t = 10$, all but two of the z_i^{10} are in the top left corner of the figure, and one of these six has found a better location with criterion value 1.1776. The two remaining z_i^{10} are still drawn towards their previous particle-best positions further "south". One of these z_i^t particle has not found a position better than the location where it started. Because PSO requires only that one particle find the optimum, increasing the number of particles simply increases the chance that the optimum is located quickly. Here, with only 8 particles in 2-dimensional space, by iteration 24, the global best z_i^t is $g^{24} = [0.1211, 0.8249]$ corresponding to a criterion value of 1.0118, very close to the true optimum of 1.0116. The PSO search could be followed by a gradient-based, constrained non-linear optimizer to hone in on the exact optimum.

Table 1.1 MIPDs and computation times to find a 60×6 design using PSO with $N_{\text{des}} = p \times n \times k$ particles ($p \in \{.1, 1, 4, 10\}$) and $N_{\text{its}} = q \times n \times k$ PSO iterations ($q \in \{.2, 2, 8, 20\}$) optionally followed by quasiNewton algorithm (`fmincon`). The column labeled "1" is the MIPD for the design obtained by applying `fmincon` to the design g^1. The empMSPE for predicting one output based on this design are also listed.

	PSO only				PSO + fmincon				
$N_{\text{des}} = 36$; N_{its}	$.2nk$	$2nk$	$8nk$	$20nk$	1	$.2nk$	$2nk$	$8nk$	$20nk$
Criterion value	0.5181	0.6425	0.6881	0.7222	0.7571	0.6750	0.8412	0.8065	0.8086
Time (secs)	46.4	78.5	172.5	355.7	267.9	113.0	476.4	344.7	442.8
empMSPE	5.2116	4.9567	4.7519	4.7866	5.0415	5.3442	5.0821	4.8269	4.8469
$N_{\text{des}} = 360$; N_{its}	$.2nk$	$2nk$	$8nk$	$20nk$	1	$.2nk$	$2nk$	$8nk$	$20nk$
Criterion value	0.5676	0.6947	0.7210	0.7230	0.7369	0.7652	0.7730	0.7235	0.7230
Time (secs)	257.4	312.8	485.9	830.3	431.7	375.7	434.2	489.5	830.6
empMSPE	4.9607	5.2233	5.2379	5.3221	5.5145	5.1181	5.5151	5.2354	5.3222
$N_{\text{des}} = 1440$; N_{its}	$.2nk$	$2nk$	$8nk$	$20nk$	1	$.2nk$	$2nk$	$8nk$	$20nk$
Criterion value	0.6004	0.6944	0.7125	0.7168	0.8216	0.7784	0.6948	0.7279	0.7168
Time (secs)	2926.5	3043.5	3424.3	4182.8	3330.2	3138.7	3046.0	3458.0	4184.4
empMSPE	5.1696	5.1995	5.0018	4.9909	4.8221	4.9231	5.0689	5.2398	4.7784
$N_{\text{des}} = 3600$; N_{its}	$.2nk$	$2nk$	$8nk$	$20nk$	1	$.2nk$	$2nk$	$8nk$	$20nk$
Criterion value	0.6361	0.7281	0.7444	0.7637	0.7784	0.8047	0.7281	0.7444	0.7637
Time (secs)	34415.4	34641.0	35387.7	36895.6	34678.9	34600.2	34641.3	35388.0	36896.0
empMSPE	5.1640	4.6342	4.8644	5.4624	4.9741	5.1198	4.6342	4.8644	5.4624

1.4 Quality of Designs Produced

Table 1.4 investigates the effect of varying N_{des} and N_{its} in a PSO search for a Mm design having $k = 6$ inputs and $n = 60$ runs. The running times on a Linux compute machine, having a Dual Quad Core Xeon 2.66 processor with 32GB RAM are shown, together with the achieved MIPD (to be *maximized*). The effect of following PSO by the local optimizer, `fmincon.m` starting at at $g^{N_{\text{its}}}$ is also shown.

For a given number of particles, N_{des}, the left portion of Table 1.4 shows a steady increase in the maximized MIPD of the computed design as the number of iterations, N_{its}, increases. The right portion of the table shows that an increase in MIPD could usually be achieved by following PSO with `fmincon` starting at particle $g^{N_{\text{its}}}$. The extra run time needed for additional iterations and/or use of a local optimizer is worthwhile.

Interestingly, for all 4 N_{des} values, running `fmincon` with starting particle g^1 produced a better design than was obtained by running $20nk = 7200$ iterations of PSO alone. This suggests that a considerably larger value of N_{des} would be needed to find the optimum using only PSO. Results of a modified PSO are given by [6] for searching for maximin LHDs using approximately $N_{\text{des}} = 8000nk$ and $N_{\text{its}} = 100nk$.

Finally, Table 1.4 shows the empirical mean squared prediction error (empMSPE) for using the design to fit the empirical best linear unbiased predictor obtained from (1.1) to outputs from one particular $k = 6$ output function. The values are generally, but not always, lower for designs with MIPD close to 0.8. However, maximin is not the best criterion for prediction ([12], [16]). A study is currently being carried out on PSO in constructing AIMSPE*-optimal designs for calibration ([13]).

Acknowledgements This research was sponsored by the National Science Foundation under Agreements DMS-0806134 and DMS-1310294 (The Ohio State University).

References

[1] Audze, P. and Eglais, V. (1977). “New approach for Planning out of Experiments,” *Problems of Dynamics and Strengths*, 35, 104–107.

[2] Audet, C. and Dennis, Jr., J. E. (2006), “Mesh Adaptive Direct Search Algorithms for Constrained Optimization,” *SIAM J. on Optimization*, 17, 188–217.

[3] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P. (1996), “Experimental design and observation for large systems,” *Journal of the Royal Statistical Society B*, 58, 77–94.

[4] Bernardo, M. C., Buck, R. J., Liu, L., Nazaret, W. A., Sacks, J., and Welch, W. J. (1992), “Integrated circuit design optimization using a sequential strategy,” *IEEE Transactions on Computer-Aided Design*, 11, 361–372.

[5] Chen, R.-B., Chang, S.-P., Wang, W., Tung, H.-C., and Wong, W. K. (2013a), “Optimal Minimax Designs via Particle Swarm Optimization Methods,” `http://www.newton.ac.uk/preprints/NI13039.pdf`.

[6] Chen, R.-B., Hsieh, D.-N., Hung, Y., and Wang, W. (2013b), “Optimizing Latin hypercube designs by particle swarm,” *Statistics and Computing*, 23, 663–676.

[7] Givens, G. and Hoeting, J. (2012), *Computational Statistics*, Wiley.

[8] Goldberg, D. E. (1989), *Genetic algorithms in search, optimization, and machine learning*, Reading, Mass.: Addison-Wesley Pub. Co.

[9] Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), “Minimax and maximin distance designs,” *Journal of Statistical Planning and Inference*, 26, 131–148.

[10] Kennedy, J. and Eberhart, R. (1995), “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948.

[11] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), “Optimization by Simulated Annealing,” *Science*, 220, pp. 671–680.

[12] Leatherman, E., Santner, T., and Dean, A. (2013), “Optimal Bayesian designs for computer simulator experiments,” *In Preparation*.

[13] — (2014), “Optimal Bayesian designs for combined physical and deterministic simulator experiments in the prediction setting,” *In Preparation*.

[14] M. Liefvendahl, Stocki, R. (2006), “A study on algorithms for optimization of Latin hypercubes”, *Journal of Statistical Planning and Inference*, 136 (2006) 3231 3247

[15] Nelder, J. A. and Mead, R. (1965), “A simplex method for function minimization,” *Computer Journal*, 7, 308–313.

[16] Pronzato, L. and Muller, W. G. (2012, “Design of computer experiments: space filling and beyond”, *Statistics and Computing*, 22, 681701.

[17] Sacks, J., Schiller, S., and Welch, W. (1989a), “Design for computer experiments,” *Technometrics*, 31, 41–47.

[18] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989b), “Design and analysis of computer experiments,” *Statistical Science*, 4, 409–423.

[19] Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer Verlag.

[20] Shewry, M. C. and Wynn, H. P. (1987), “Maximum entropy sampling,” *Journal of Applied Statistics*, 14, 165–170.

[21] Yang, X.-S. (2010), *Engineering Optimization: An Introduction with Metaheuristic Applications*, Wiley Publishing, 1st ed.