On Sampling Methods for Costly Multi-objective Black-box Optimization

Ingrida Steponavičė Mojdeh Shirazi-Manesh Rob J. Hyndman
Kate Smith-Miles Laura Villanova

September 20, 2015

Abstract

We investigate the impact of different sampling techniques on the performance of multi-objective optimization methods applied to costly black-box optimization problems. Such problems are often solved using an algorithm in which a surrogate model approximates the true objective function and provides predicted objective values at a lower cost. As the surrogate model is based on evaluations of a small number of points, the quality of the initial sample can have a great effect on the overall effectiveness of the optimization. In this study, we demonstrate how various sampling techniques affect the results of applying different optimization algorithms to a set of benchmark problems. Additionally, some recommendations on usage of sampling methods are provided.

1 Introduction

A plethora of practical engineering problems involve multiple conflicting objectives which have to be optimized simultaneously. Solving such problems requires more effort than single-objective optimization as they usually have many (possibly infinite) optimal solutions; such solutions compose the so-called *Pareto optimal set*.

To add further to the challenge, for many real-world optimization problems there is also an absence of algebraic objective or response function definitions. Examples are crash tests, chemical reactions, many laboratory experiments, etc. Therefore an important challenge in optimization practice is how to solve an optimization problem in the absence of an algebraic model of the system to be optimized. Such optimization problems are called *black-box* as the available information is just input-output data without prior knowledge of the characteristics or physics of the relationships involved.

Due to the lack of an analytical description of the objective functions, derivatives are unavailable and derivative-based optimization methods cannot be used. Moreover, in many practical applications, the objective functions (or the associated constraints) are very costly to evaluate and it is desirable to limit the number of evaluations. Consequently, for *costly black-box* multi-objective optimization problems, the main concern is to find the Pareto optimal set with as few function evaluations as possible. Traditional derivative-free methods based on direct search or gradient estimation via numerical differentiation are not usually viable as they require many more function evaluations than can be comfortably afforded.

A popular and successful approach for derivative-free optimization of costly black-box functions is to construct response surface models known as *surrogate models* (or metamodels) that mimic the behavior of the real-word process as closely as possible while being less resource-demanding to evaluate. Surrogate-based optimization methods became popular a few decades ago even though they were proposed much earlier [20]. Among the various potential surrogate

models, polynomial response surface models [3], kriging [35] and radial basis functions (RBF) [6] are widely used in solving costly black-box optimization problems.

In recent years, much attention has been devoted to develop multi-objective optimization methods (e.g., see [30, 40, 50]) to deal with real-world applications characterized as *costly multi-objective black-box* optimization problems using surrogate models to replace the unknown objective functions. Little attention has been focused so far though on the impact of the initial sample on the performance of the developed algorithms.Every black-box optimization algorithm starts the optimization process with an initial sample, usually a very limited one in the case of expensive function evaluations. The initial sample provides some knowledge for the method to further investigate the decision space with the aim of finding the global optimum. When the evaluation of objective functions is costly, these evaluated points are usually fed to a surrogate model to predict the real response function values of unevaluated points. An inexpensive surrogate model is constructed based on an initial sample; the model is then used in a search for the next points to evaluate. This approach decreases the number of resource-consuming function evaluations. Remarkably, this approach suggests that the initial sample selected to build a surrogate model can strongly impact the efficiency of optimization. This consideration motivates our analysis of the sampling effect on the optimization search.

Sampling methods have been used for a wide range of purposes ranging from censuses and surveys [48, 1, 49] to numerical and computational studies [9, 5] to experimental investigations in industry and science [23, 34, 11, 28]. In general, sampling methods can be used in two main types of studies: observational and experimental studies [4]. Observational studies aim to draw inferences about an entire space from a sample [33], whereas experimental studies aim to identify the cause-effect relationship between input and output variables through controlled experiments [39]. In the first case, sampling methods must provide a representative sample of the entire space; in the second case, sampling methods must provide a small informative sample selected from the set of feasible experiments (the decision space). It is the latter experimental scenario that is relevant when using sampling methods in an optimization context.

To our knowledge, there exist only a few studies investigating the impact of sampling methods in the context of single objective surrogate-based optimization [25, 43] and multi-objective optimization [29]. With regards to the multi-objective field, Poles et al. [29] focused on evolutionary optimization algorithms. Evolutionary algorithms require thousands of function evaluations to achieve a good approximation of the Pareto optimal set; therefore, they are not suitable for costly optimization. Instead, we focus on methodologies that require hundreds of function evaluations. Indeed, this study aims to investigate the effect and importance of initial sampling techniques on methods suitable for *costly multi-objective black-box* optimization.

The remainder of this paper is as follows. In Section 2, we recall the basic concepts related to black-box and multi-objective optimization. Widely-used sampling methods and the concepts behind them are outlined in Section 3. In Section 4, we present the experimental set-up used in the study and we illustrate the experimental results. Section 5 summarizes the results obtained, provides insights and suggestions for future research directions, and draws some final conclusions.

2 Problem Description

The *multi-objective optimization problem* comprises multiple objective functions which are to be minimized simultaneously. It can be expressed in the following form:

$$\min \mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\big)^T \qquad \text{subject to } \mathbf{x} \in S, \tag{1}$$

where $S \subset \mathbb{R}^d$ is the feasible set and $f_i :\to \mathbb{R}$, $i = 1, \ldots, m$ ($m \geq 2$), are objective functions to be minimized simultaneously. All objective functions are represented by the vector-valued function $\mathbf{f} : S \to \mathbb{R}^m$. A vector $\mathbf{x} \in S$ is called a *decision vector* and a vector $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ is called an *objective vector*.

We assume that at least one of the functions f_i is "costly"; that is, its evaluation requires a significant amount of resources and no analytic expression is available. Therefore, problem (1) is called a *costly multi-objective black-box optimization problem*.

In multi-objective optimization, the objective functions $f_1, \ldots, f_m$ in (1) are typically conflicting. In that case, there does not exist a decision vector $\bar{\mathbf{x}} \in S$ such that $\bar{\mathbf{x}}$ minimizes f_i in S for all $i = 1, \ldots, m$, but there exists a number (possibly infinite) of Pareto optimal solutions. In mathematical terms, a decision vector $\bar{\mathbf{x}} \in S$ and its image $\bar{\mathbf{z}} = \mathbf{f}(\bar{\mathbf{x}})$ are said to be *Pareto optimal* or *non-dominated* if there does not exist a decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\bar{\mathbf{x}})$ for all $i = 1, \ldots, m$ and $f_j(\mathbf{x}) < f_j(\bar{\mathbf{x}})$ for some $j = 1, \ldots, m$. If such a decision $\mathbf{x} \in S$ does exist, $\bar{\mathbf{x}}$ and $\bar{\mathbf{z}}$ are said to be *dominated* by $\mathbf{x}$ and its image $\mathbf{z} = \mathbf{f}(\mathbf{x})$, respectively. The Pareto optimal set in the objective space is also called the *Pareto optimal front*.

3 Sampling Methods

Sampling methods for experimental studies have been attracting a great deal of attention since the 1800s and have resulted in a dedicated field of research known as Design of Experiments. Their importance directly relates to the efficient collection of informative data, allowing for the quick delivery of robust results. This translates into considerable savings that minimize costs and time related to both physical (real-world) and computer-based experimentation.

Many sampling methods assume that the unknown objective function can be approximated by a simple model (e.g., linear or quadratic) and recommend samples located on the boundary of the design space. This assumption can be safely made if some knowledge exists of the objective function or if the approximation occurs locally (i.e., in a relatively small sub-area of the decision space) [17]. In black-box optimization, no knowledge exists regarding the objective function and the entire decision space is typically searched. Therefore, sampling methods are required to provide samples that are spread out across the entire decision space. Two such classes of methods are *space-filling methods* and *low-discrepancy sequences*. *Space-filling methods* aim to generate widespread samples using a range of different criteria including equally-spaced intervals and distance measures. On the other hand, *low-discrepancy sequences* use a measure of uniformity (discrepancy) that minimizes the difference between the percentage of points falling in a particular region on a unit cube and the percentage of volume occupied by this region. The main space-filling designs and low-discrepancy sequences are reviewed below and investigated in our computational study.

3.1 Simple random sampling

In simple random sampling (SRS), N decision vectors are randomly sampled from the decision space [38]. Decision vectors have the same probability of being chosen; the constant chance of selection extends to pairs, triplets, and so on (e.g., any given pair of decision vectors has the same chance of selection as any other pair).

SRS is among the most popular sampling methodologies thanks to its simplicity and low computational demand. One drawback of SRS relates to its vulnerability to sampling error; indeed, the randomness of its selection process may result in a sample that is not evenly spread throughout the entire decision space. This is particularly true for small samples in high-dimensional regions that often exhibit apparent clustering and poorly covered regions [36]. Systematic and

stratified sampling techniques have been developed to overcome this issue and choose a "more representative" sample.

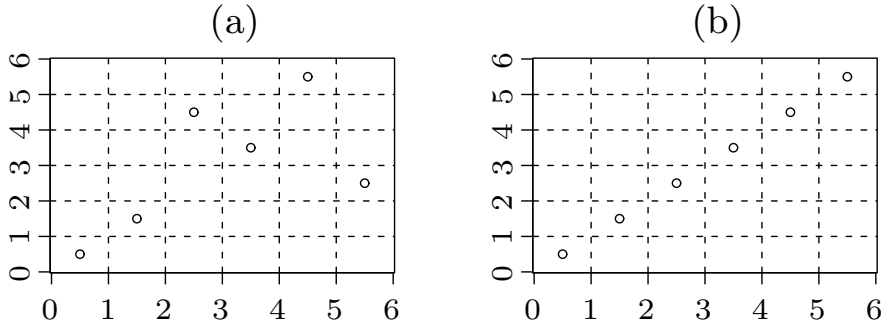
3.2 Latin hypercube sampling

Latin hypercube sampling (LHS) is a stratified sampling technique. LHS controls how random samples are generated from a given probability distribution (usually uniform). To generate a sample of N decision vectors, the domain of each decision variable is divided into N equally-spaced and non-overlapping intervals; then, one value is selected at random from each such interval. Random permutation of the resulting values for all decision variables results in a random latin hypercube sample.

LHS originated in 1979 for computer-based experiments in order to address the need for a better and more efficient coverage of the decision space [21]. The authors showed that LHS reduces the variance in their chosen application of Monte Carlo integration.

The main advantage of LHS over SRS derives from its one-dimensional projection property: a latin hypercube sample projected into one dimension results in a set of evenly distributed points. Due to this property, LHS is the most commonly used stratified sampling technique in many areas of computer-based experiments. Despite this, different studies show that it is not always the best choice [42, 44]. Indeed, LHS does not guarantee an uniform coverage of the decision space as (sometimes large) areas of the decision space might remain unexplored. Two such examples are reported in Figure 1 showing a LHS in two dimensions with six intervals per decision variable. In both cases, there is a large area of the decision space that is not explored; therefore, if we use such a sample to develop a prediction model, then the prediction will be poor in those unexplored areas. In the worst case scenario (b), LHS can generate a sample with two perfectly correlated decision variables; such a sample causes the effects of the two variables to be completely confounded. To overcome these limitations, LHS methods have been improved through the adoption of an additional criterion; such improvements resulted in maximin distance and minimum correlation LHS methods described later on.

Figure 1: Two LHS configurations with two variables in six intervals



3.3 Maximin sampling

Maximin sampling belongs to the class of distance-based sampling methods. Distance-based sampling methods make use of the Euclidean distance to prevent sampled points from clustering too close together so that they over-represent some regions of the design space.

The aim of maximin sampling is to scatter points in the decision space such that the minimal pairwise distance between points is maximized. Let $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jd})$ and $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kd})$ be two different decision vectors in a sample $D(N, S)$, where N is

the sample size and S is the d-dimensional feasible set. The following mathematical problem must be solved:

$$\max \min s^2(\mathbf{x}_j, \mathbf{x}_k) \tag{2}$$

where

$$s^2(\mathbf{x}_j, \mathbf{x}_k) = \sum_{i=1}^{d} \left(\frac{x_{ji} - x_{ki}}{U_i - L_i} \right)^2$$

and $\mathbf{x}_j, \mathbf{x}_k \in D(N, S)$, $j, k = 1, 2, \ldots, N$ $(j \neq k)$, U_i and L_i are the upper and lower limits of the ith variable. Therefore, a maximin sample of size N contains minimum pairwise distances that are maximum compared to any other N-sized sample.

Maximin sampling was first introduced by Johnson *et al.* [14]. It is among the best methods to obtain an even coverage of the decision space. However, it tends to prioritize decision vectors that are located near the boundary of the decision space. Also, despite computational efficiency in low dimensions, the method is very demanding in high dimensions. To overcome this issue, various approximate maximin sampling methods have been developed. Approximate methods use conventional nonlinear programming algorithms to reduce the computational cost of the procedure at the expense of potentially providing solutions that are not globally optimal, only locally optimal.

Here, we propose a simple approximation method we call "Nearly maximin" consisting of the following steps:

Step 1 Randomly generate a decision vector $\mathbf{x}_1$ from the decision space and choose it as the first element of the sample;

Step 2 Randomly generate n decision vectors from the decision space and for each vector, calculate the Euclidean distance to the closest element of the existing sample;

Step 3 Choose the one having the maximum distance out of the n decision vectors as the next element of the sample;

Repeat Step 2 and Step 3 until the sample comprises N decision vectors.

In a preliminary study that will be published elsewhere, the Nearly maximin method showed extremely promising results. It will be used in our computational study to allow for the investigation of high-dimensional test problems.

3.4 Maximin LHS

Both LHS and maximin sampling produce samples with attractive properties. LHS guarantees that the one-dimensional projection of the sample presents an even spread in the variables' domains; maximin guarantees that no two elements (decision vectors) in the sample are close together. However, both methods suffer from limitations. In particular, LHS might occasionally generate samples with points that are close to each other as in the examples of Figure 1, whereas maximin tends to select samples that are located near the boundary of the decision space.

To overcome these limitations, Morris and Mitchel [24] suggested that LHS be combined with the maximin criterion. The resulting method is known as Maximin LHS (M-LHS). It consists of (a) generating the maximum number of possible LHS samples, (b) measuring their maximin distances and (c) selecting the most evenly spread sample (optimal sample).

M-LHS preserves the one-dimensional projection property of LHS while ensuring that no two points in the LHS design are very close to each other. Therefore, a good spread of decision vectors is achieved not just in each single variable domain but also in the entire decision space. Also, the decision vectors in the sample are preferentially located in the interior of the decision

space thus providing a compromise between maximin property and good projective properties in each dimension (as guaranteed by Latin hypercubes) [24]. Unfortunately, constructing samples by M-LHS can be quite time consuming when the number of dimensions and design points increase. Indeed, there exists $(N!)^{d-1}$ LHS samples for N divisions and d dimensions; for each such sample, the maximin distances need to be calculated in order to identify the optimal one.

3.5 Correlation LHS

To find optimal LHS Iman and Conover [12], Owen [27], and Tang [41] proposed to use a criterion minimizing correlation between the factors. This is useful in applications requiring a sample to be composed of decision vectors without (or with small) correlation. Owen proposed to measure the goodness of LHS with respect to a criterion of minimum pairwise correlations which is defined as follows:

$$\rho^2 = \frac{\sum_{i=2}^{d} \sum_{j}^{i-1} \rho_{ij}^2}{d(d-1)/2}, \tag{3}$$

where ρ_{ij} is the pairwise correlation between columns i and j of the design, and $\rho_{ij} \in [0, 1]$. The smaller ρ^2 is, the weaker the pairwise correlation is.

In Correlation LHS (C-LHS) method suggested by Owen [27], the sum of between-column squared correlation is decreased by alternating forward and backward Gram-Schmidt orthogonalization. In our computational study we used the Matlab implementation of Owen's method.

One might think that minimizing correlation should spread out the points and maximizing the distance between the points should reduce the correlation. However in practice, there is no one-to-one relationship between the two, and designs obtained by these two criteria can be quite different [15]. In other words, C-LHS not necessarily provides a well spread sample.

3.6 Halton sequence sampling

The Halton sequence sampling method generates quasi-random numbers of high-dimensionality with a high level of uniformity across the space. Halton sequence is constructed according to a deterministic method that uses different prime bases for different dimensions to create a d-dimensional low-discrepancy sequence [7, 19]. The method is based on the fact that each non-negative integer can be expanded using a prime base. Construction of Halton sequence in d-dimensional space is as follows:

1. Choose d prime integers $p_1, p_2, \ldots, p_d$ (usually the first primes $p_1 = 2$, $p_2 = 3, \ldots$, are chosen).

2. To generate the i-th sample, consider the base p representation for i which takes the form:

   $$i = a_0 + a_1 p + a_2 p^2 + a_3 p^3 + \ldots$$

   where each a_j is an integer in $[0, p-1]$.

3. The following point in $[0, 1]$ is obtained by reversing the order of the bits and moving the decimal point:

   $$r(i, p) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \frac{a_3}{p^4} + \ldots$$

4. Starting from $i = 0$, the i-th sample in the Halton sequence is

   $$(r(i, p_1), r(i, p_2), \ldots, r(i, p_d)) \tag{4}$$

Halton is an extension of the Van der Corput sequence, which was originally introduced for one dimension and a base of 2. The Van der Corput sequence is obtained by using $p = 2$. However, Halton sequences based on large primes ($d > 10$) can be highly correlated, and their coverage can be worse than that of the pseudo-random uniform sequences.

3.7 Hammersley sequence sampling

The Hammersley sequence [8] belongs to the class of low-discrepancy sequences, and is closely related to the Fibonacci series. The Hammersley sequence is an adaptation of the Halton sequence (4) when the required sample size N is known. In such a case, a better uniformly distributed sample can be obtained by using only $d - 1$ distinct primes. In a Hammersley sequence with N elements and starting from $i = 0$, the i-th d-dimensional vector will be

$$\left(\frac{i}{N}, r(i, p_1), r(i, p_2), \ldots, r(i, p_{d-1})\right) \quad \text{for } i = 0, 1, 2, \ldots, N-1. \tag{5}$$

Hammersley sequence sampling provides better uniformity properties over LHS [22]; in particular, the chance of samples with clustered decision vectors is lower. Also, compared to other conventional techniques, Hammersley sampling requires far smaller samples to approximate the mean and variance of distributions based on empirical studies [16].

3.8 Sobol sequence sampling

Sobol sequence sampling is an improved version of the Halton and Hammersley methods. Indeed, despite the Halton and Hammersley methods being relatively simple and efficient, they suffer from a common pitfall — the performance of these two sampling methods degrades substantially in higher dimensions. Sobol sequences have been proposed to approximate the integral over the d-dimensional unit cube:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} f(x_i) = \int_{[0,1]^d} f(x)\, \mathrm{d}x$$

where f is a real integrable function over a d-dimensional unit hypercube and $x_0, \ldots, x_{n-1}$ are n points in $[0, 1]^d$ comprising a "Sobol sequence". The Sobol sequence, as originally defined by Sobol [37], is generated from a set of special binary vectors of length w bits, v_i^j, $i = 1, 2, \ldots, w$, $j = 1, 2, \ldots, d$. These numbers, v_i^j, are called direction numbers. To generate them for dimension j, one should begin with a primitive polynomial over the finite field $\mathcal{F}_2$ with elements $\{0, 1\}$. Let us assume the primitive polynomial is

$$p_j(x) = x^q + a_1 x^{q-1} + \cdots + a_{q-1}x + 1.$$

Then we use its coefficients to define a recurrence relation for calculating v_i^j, the direction number in dimension j. It is generated using the following q-term recurrence relation:

$$v_i^j(x) = a_1 v_{i-1}^j \oplus a_2 v_{i-2}^j \oplus \cdots \oplus a_{q-1} v_{i-q+1}^j \oplus v_{i-q}^j \oplus (v_{i-q}^j / 2^q),$$

where $i > q$, $\oplus$ denotes the bitwise XOR operation, and the last term is v_{i-q} shifted right q places. The initial numbers $v_1^j \cdot 2^w, v_2^j \cdot 2^w, \ldots, v_q^j \cdot 2^w$ can be arbitrary odd integers smaller than $2, 2^2, \ldots, 2^q$, respectively. The Sobol sequence x_n^j ($n = \sum_{i=0}^{w} b_i 2^i$, $b_i \in \{0, 1\}$) in dimension j is generated by

$$x_n^j = b_1 v_1^j \oplus b_2 v_2^j \oplus \cdots \oplus b_w v_w^j.$$

Different primitive polynomials should be used to generate Sobol sequence in each dimension. Currently there are more efficient ways of generating Sobol sequences proposed in the literature (see, e.g. [26]).

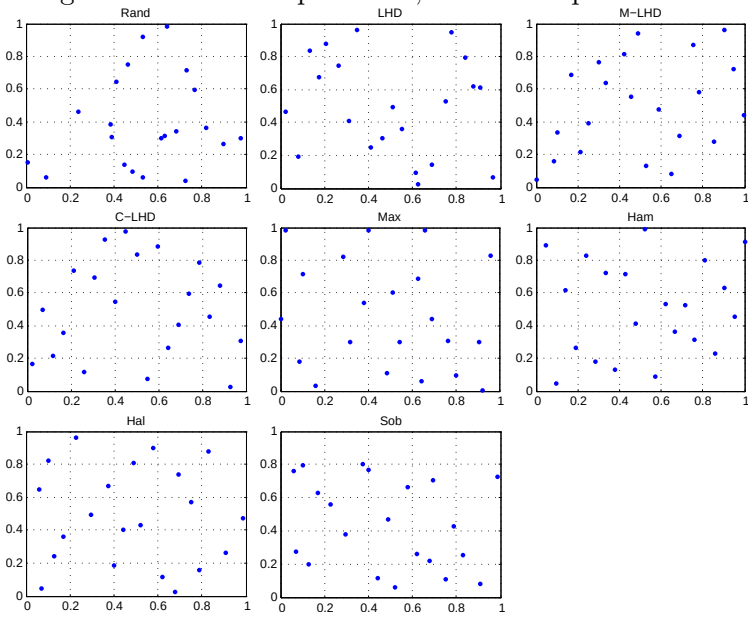
3.9 Summary of sampling methods

This section outlines the main characteristics of sampling methods discussed above. Table 1 summarizes the sampling methods in terms of their main features. Samples consisting of 32 points in two dimensional space, and generated by different sampling methods, are presented in Figure 2 for a visual comparison.

Table 1: Characteristics of sampling methods

Sampling method	Simple	Low computational cost	One-dimensional projection	Uniform coverage	Suitable for high-dimensions	Stochastic	Additional features
Random	+	+	–	–	–	+	minimum bias
LHS	+	+	+	–	–	+	stratified
Maximin	–	–	–	+	–	+	maximum minimum distance
M-LHS	–	–	+	+	–	+	stratified, maximum minimum distance
C-LHS	–	–	+	–	–	+	stratified, low correlation
Halton	+	+	–	+	–	–	minimum discrepancy
Hammersley	+	+	–	+	–	–	minimum discrepancy
Sobol	–	+	–	+	+	–	minimum discrepancy

Figure 2: Initial sample in 2D, number of points = 32



An important consideration relates to the methods' computational cost in the context of costly optimization. For those methods that demand moderate to intensive computational efforts, it is important to investigate the compromise between (a) the time required to generate the sample and (b) the sample quality. The sample quality is its ability to decrease the number of further function evaluations without affecting the results of the optimization procedure. Obviously, if function evaluations are highly costly and involve resources other than time, even a small decrease in the number of function evaluations justifies the higher computational time required to generate the optimal sample.

4 Experiments

The impact of sampling method on multi-objective optimization algorithm efficiency is evaluated by means of a comprehensive benchmark problem set. The design of the experimental study is described in Section 4.4. Section 4.2 gives an overview of the benchmark problems. The major part of this section is devoted to discussion of the obtained results and the appropriate observations. This is covered in Section 4.5.

4.1 Optimization algorithms considered

In our study, we considered three algorithms designed for costly multi-objective optimization problems, namely ParEGO, SMS-EGO and ϵ-EGO. These algorithms were selected due to their available implementation in the R package **mlrMBO** [2].

ParEGO is a state-of-art algorithm developed by Knowles [18]. It uses the augmented Tchebycheff norm to convert a multi-objective problem into a scalarized one:

$$f_\lambda(\mathbf{x}) = \max_{j=1,\ldots,m} \left(\lambda_j f_j(\mathbf{x})\right) \pm \rho \sum_{j=1}^{m} \lambda_j f_j(\mathbf{x}), \tag{6}$$

where $\rho > 0$ is a small positive number and λ is a weight vector. ParEGO randomly selects w from a uniformly distributed set in each iteration. Then a surrogate model is fitted to the respective scalarized problem. At each iteration of the algorithm, a different weight vector is drawn uniformly at random from the set of evenly distributed vectors allowing the model to gradually build up an approximation to the true Pareto set. Before scalarization, the objective functions are normalized with respect to the known (or estimated) limits of the objective space to the range $[0, 1]$. At each iteration, the method uses a genetic algorithm to search for the solutions that maximizes an infill criterion, called expected improvement, with respect to a surrogate model. Only the best solution is evaluated on the actual problem. After evaluation of the selected solution on the real expensive function, ParEGO updates the GP surrogate model of the landscape and repeats the same steps.

The other two algorithms do not convert a multi-objective optimization problem to a single one but use a multi-objective optimization of infill criteria on each objective in order to obtain a candidate set for evaluation. SMSEGO [30] optimizes the hypervolume and ϵ-EGO [45] looks at search solutions with respect to the additive ϵ-indicator which has been introduced by Zitzler *et al.* [53]. An additive ϵ-indicator of approximation set A gives the minimum value ϵ by which each point in the real front R can be added such that resulting transformed approximation is dominated by A.

4.2 Test problems

The test set consists of different benchmark problems with a variety of characteristics in both the decision and objective spaces. The objectives of test problems can be either unimodal (U) or multimodal (M). Multimodal problems are more difficult than unimodal problems, and more representative of real-world problems. The Pareto optimal front can be convex, linear, concave, disconnected, or some combination of the former. It is well known that the type of Pareto front can directly affect the performance of the optimization algorithms. For example, disconnected Pareto fronts can increase the likelihood that an algorithm will fail to find all regions of the Pareto optimal front. The fitness landscape may be one-to-one or many-to-one and the later property impacts some algorithms' ability to find multiple, otherwise equivalent optima. For a more detailed discussion on test problems properties we refer readers to [10].

Our test set includes the following benchmark problems:

- *OKA2* $m = 2$, $d = 3$. The true Pareto optimal set for this problem is a spiral shaped curve in the decision space, and the density of the Pareto optimal solutions in the objective space is low.
- *Kursawe* This problem has a scalable number of decision variables. In our experiment we used $d = 3$, $m = 2$. Its Pareto optimal set is disconnected and symmetric in the decision space, and disconnected and concave in the objective space.
- *Viennet* $m = 3$, $d = 2$. The true Pareto optimal set is convex in the objective space.
- *ZDT family*: ZDT problems share such characteristics as multimodality, discontinuity and possession of multiple Pareto fronts; for all problems $m = 2$ and d is scalable, however we used d values suggested by the authors.
  - ZDT1: $d = 30$; Pareto optimal set in the objective space is convex.
  - ZDT2: $d = 30$; Pareto optimal set in the objective space is nonconvex.
  - ZDT3: $d = 30$; Pareto optimal set is disconnected in both objective and decision spaces. Pareto optimal set consists of of one mixed convex/concave component and several convex components in the objective space.
  - ZDT4: $d = 10$; it has 21 local Pareto optimal fronts and therefore is highly multimodal. Pareto optimal set in the objective space is convex.
  - ZDT6: $d = 10$; it has a nonuniform search space, i.e., the Pareto optimal solutions in the decision space are non-uniformly distributed along the global Pareto set, and also the density of the solutions is lowest near the Pareto optimal set and highest away from it. Pareto optimal set in the objective space is concave.
- DTLZ1: is a scalable problem in both objective and decision space and has multiple global optima. Thus, the only difficulty provided by this problem is convergence to the Pareto optimal hyperplane. We solved three sizes of this problem: 1) $m = 4$ and $d = 13$; 2) $m = 6$ and $d = 15$; 3) $m = 8$ and $d = 17$.

The major characteristics of the selected benchmark problems are summarized in Table 2.

4.3 Performance assessment

In multi-objective optimization, the definition of solution quality is substantially more complex than for single-objective problems as the optimization goal itself consists of several objectives

Table 2: Summary of test problems characteristics

Problem	No of objectives	No of variables	Modality	Convex	Concave	Disconnected	Linear	Pareto many to one
OKA2	Bi-objective	3	U	+	–	–	–	–
Kursawe		3	U	–	+	+	–	–
ZDT1		30	U	+	–	–	–	–
ZDT2		30	U	–	+	–	–	–
ZDT3		30	M	+	–	+	–	–
ZDT4		10	M	+	–	–	–	–
ZDT6		10	M	–	+	–	–	+
Viennet	3	2	U	+	–	–	–	–
DTLZ1	4	13	M	–	–	–	+	+
DTLZ1	6	15	M	–	–	–	+	+
DTLZ1	8	17	M	–	–	–	+	+

such as convergence to the true Pareto frontier, uniform distribution of obtained nondominated solutions and maximum extent of obtained nondominated set with respect to each objective. Therefore, a number of quality metrics usually taking into account one solution quality characteristic have been proposed (see, e.g, [13, 47]). The most widely used performance metric is a hypervolume (HV) indicator (also known as an S-metric) [51] which defines the size of the region dominated by the relevant Pareto set approximation. As such it provides information about closeness and diversity at the same time. In addition, it possesses a desirable property: whenever one approximation completely dominates another approximation, the HV of the former will be greater than the HV of the latter [52]. The HV metric corresponds to the size of the region of the objective space bounded by a reference point. In our study, we calculated the HV metric using normalized values of the objective functions.

4.4 Experimental setup

In this study we control: (a) the size of the initial sample, (b) the optimization budget, (c) the dimension of decision space, and (d) the dimension of objective space.

The initial design size was set to $n_{\text{init}} = 11d - 1$ based on the recommendations in [18]. An example of the initial samples generated by different sampling methods for two decision variables is given in Figure 2. The number of optimization iterations was restricted to 200 resulting in a total budget of $n_{\text{total}} = 200 + 11d - 1$. Taking into account the different number of dimensions, the algorithms were evaluated on the 11 test problems discussed in Section 4.2.

The Pareto front approximations of the algorithms were compared not only at the last iteration ($n = 200$) but as well at intermediate iterations ($n = 50$, 100, and 150) with respect to the HV metric. For each test function the reference point was estimated based on the nondominated set of initial samples.

With regards to the sampling methods, it is important to point out the following aspects. It can be computationally very expensive to achieve optimal maximin and low correlation LHS samples; therefore, we have chosen the best sample out of 1000 randomly generated LHS samples with regard to the corresponding criterion (maximin or minimum correlation). The low-discrepancy sampling methods (i.e. Hammersley, Halton and Sobol sequence) are deterministic as there is no run-to-run difference between generated samples; therefore, we have used the "random-start sequence" trick [46]. By defining random starts for generation of these samples, the sequences differ in each run resulting in stochasticity of the samples.

4.5 Results

This section elaborates on a detailed analysis of the results from the experiments performed. In each numerical experiment an initial sample was generated by one sampling method running it 100 times. Then an optimization search was performed by each algorithm over these runs. Their performance has been estimated with respect to the HV metric.

Corresponding results of each sampling method were compared to those of the other seven methods, to determine if its results had a statistically significant advantage. Comparisons were performed using the unpaired t-test [31] and differences were deemed statistically significant at the 0.01 significance level. The significance was tested multiple times, i.e. after 50, 100, 1500, and 200 function evaluations.

The average performance of optimization algorithms against sampling methods on different test problems is represented in Figures 3–7. However, in real-world applications, to know the average performance is not enough and usually the worst case scenarios are taken into account as well. The worst case scenario provides some additional information but sometimes it is considered as too conservative. Instead of the worst case scenario, we may want to know the mean of the realizations above a specified quantile; i.e., the conditional value-at-risk (CVaR) introduced in [32]. We calculated CVaR with a selected confidence level of 0.05, giving the average value over a distribution tail consisting of the 5% worst realizations. Due to space limitations, we could not provide the graphs of all the optimization algorithms and test problems considered. Therefore, we have selected the ones providing most of the information and supporting the main observations.

We have noticed that the largest variability over 100 runs is produced by Hammersley, Sobol and Halton Sequences which by nature are deterministic methods as sequences are finite. However, in order to generate 100 runs we used Matlab functions with various *leap* and *skip* parameters values which produce different subsets of these sets, sometimes not fully covering the whole decision space. To our knowledge, there is no recommendation on how to select these parameters. Therefore, the average performance of the deterministic sampling methods is influenced by some initial samples not spread throughout the entire space.

According to the obtained results regarding the average performance of the optimization algorithms based on a normalized HV metric, we can observe some trends. Generally, it can be noticed that sampling methods do not affect optimization algorithm performance significantly (the difference of HV metric values over 100 runs is not statistically significant at the 1% level) on the problems with objective and decision space dimensions both lower or equal to three. Also, we discovered that algorithm performance is very similar when using samples generated by stochastic sampling methods (namely, SRS, LHS, M-LHS, and C-LHS) on the bi-objective problems with high-dimensional decision space, and there is no statistically significantly difference among them as illustrated in Figures 5 and 6. The ParEGO method with initial samples generated by LHS has not demonstrated the best performance on any single problem, while its improved versions (i.e., M-LHS and C-LHS) have shown very good performance on

Figure 3: ParEGO performance on OKA2 problem

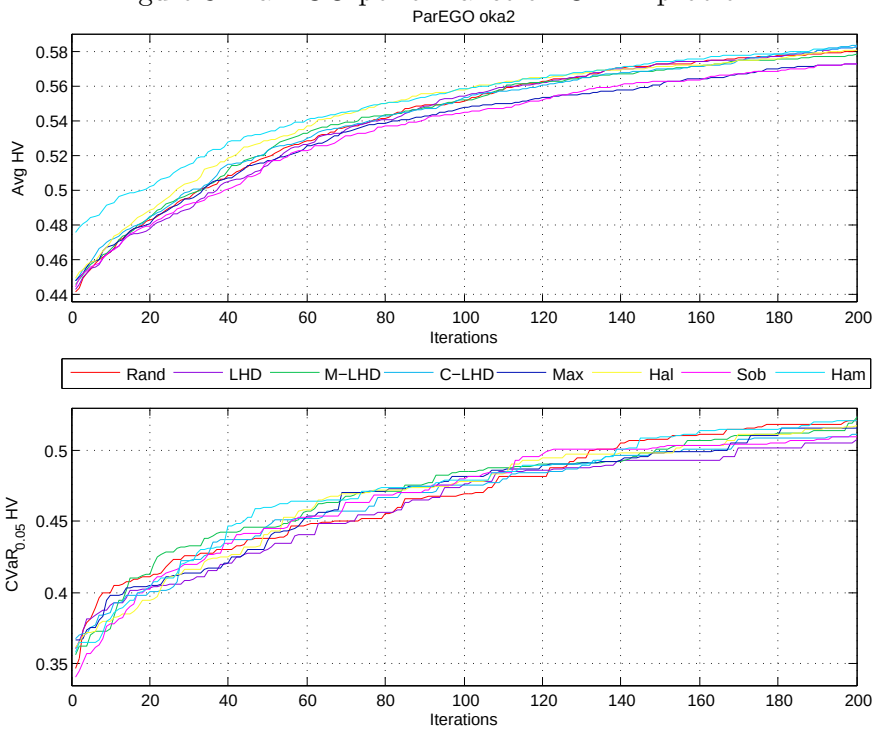


Figure 4: SMS-EGO performance on Kursawe problem

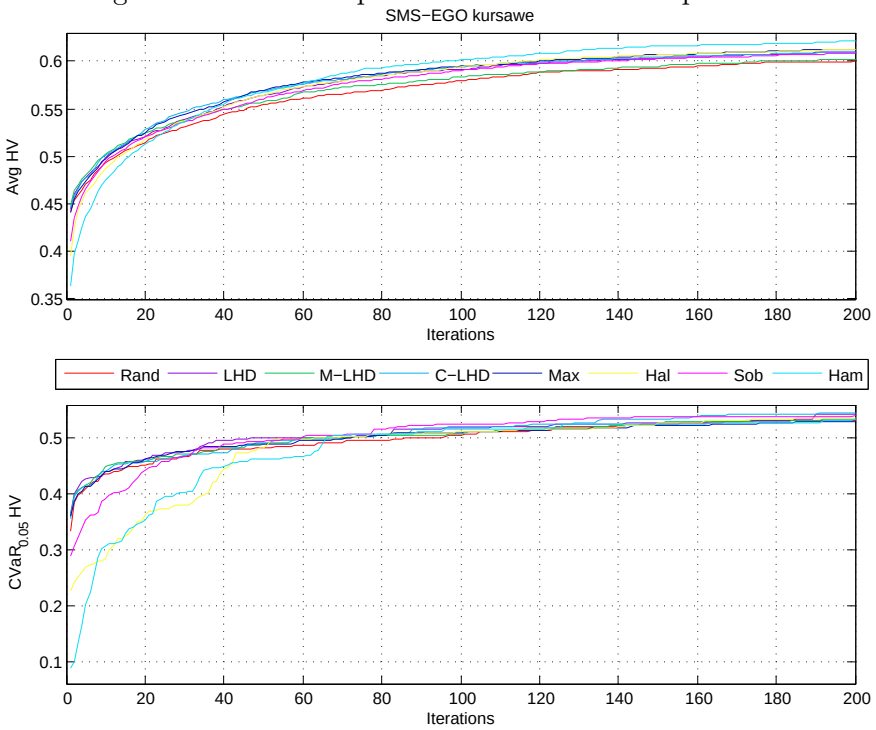


Figure 5: ϵ-EGO performance on ZDT1 problem

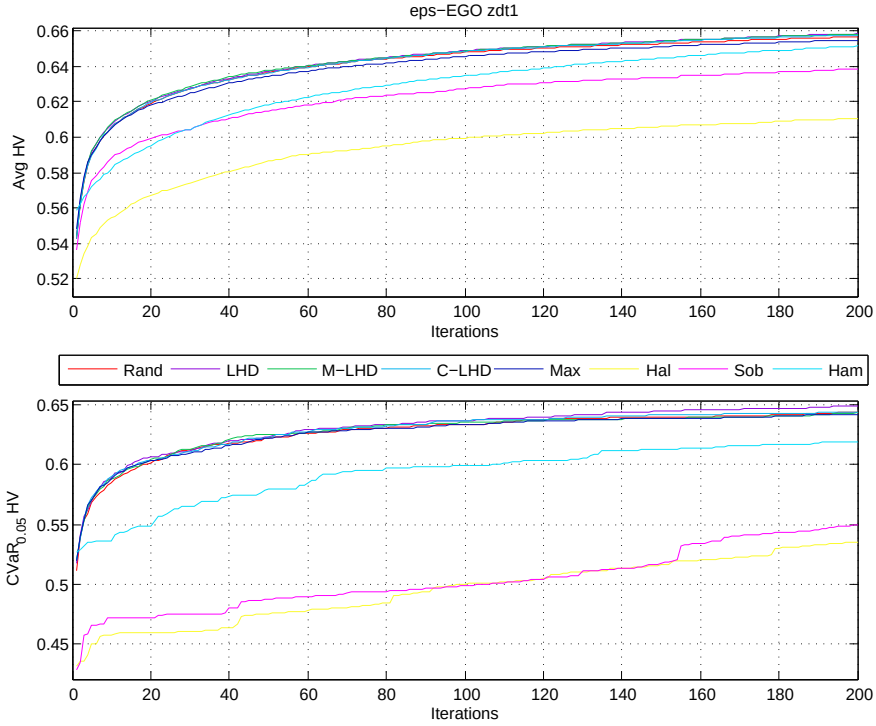


Figure 6: SMS-EGO performance on ZDT2 problem

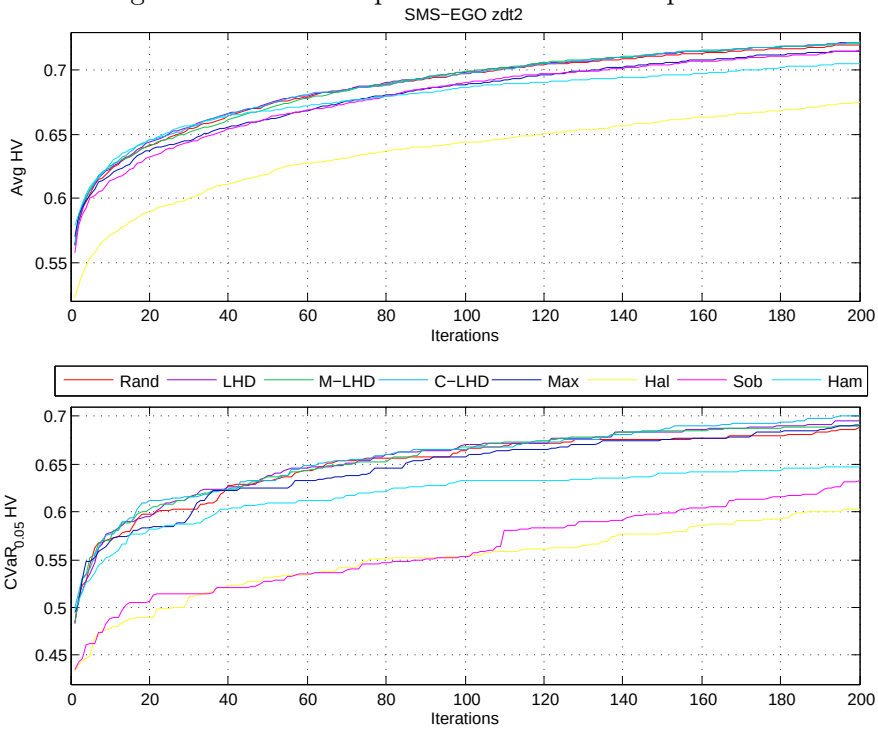
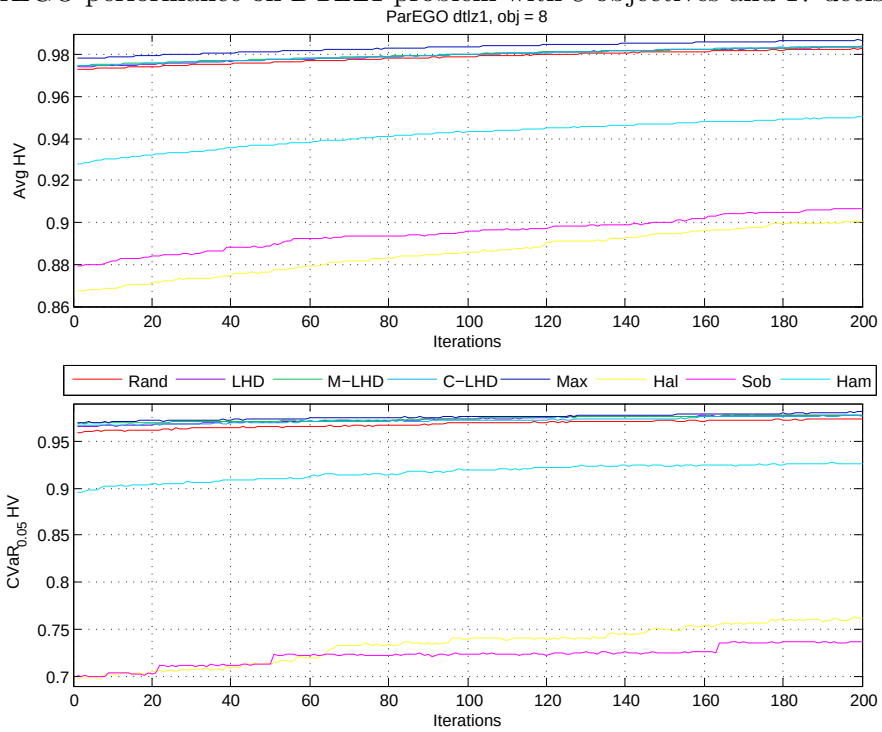


Figure 7: ParEGO performance on DTLZ1 problem with 8 objectives and 17 decision variables



bi-objective problems with a larger number of decision variables. Hammersley sequence sampling and ParEGO showed best performance or close to it on unimodal bi-objective problems of low-dimensionality with continuous Pareto front. Halton sequence sampling in conjunction with any of the considered optimization algorithms in most of the cases performed poorly, especially with a larger number of decision variables, except for low dimensional problems with a convex Pareto front. Also, it has been outperformed by the other two deterministic techniques quite a number of times. M-LHS sampling technique paired with SMS-EGO proved to behave well on the problems with a larger amount of decision variables and is outperformed by other sampling techniques on smaller problems. All optimization algorithms demonstrated better average performance on problems with more than three objectives when using maximin for initial sampling (see, e.g., Figure 7). SRS method can be considered as a decent choice because the performance of selected optimization algorithms in most cases was not significantly worse than other stochastic sampling methods. Although, there is one exception for Kursawe problem optimized with SMS-EGO algorithm (see Figure 4), where its initial samples lead to the statistically significant worst performance with respect to Hammersley and Halton sequences as well as maximin sampling method.

5 Conclusions

This section draws some conclusions and provides some recommendations based on the experiments performed and the results obtained. In addition, we shed some light on the impact of the selected sampling techniques on costly black-box multi-objective optimization, and discuss some future research questions which we leave for further investigation.

To summarize, sampling methods have no statistically significant impact (with a significance level 0.01) on the algorithm performance measured by the HV metric for low-dimensional problems, i.e., $m, d \leq 3$. Therefore, SRS method can be considered as a decent choice especially for

low-dimensional problems.

Also, when using deterministic sampling methods, one has to check that the initial sample is a good representative sample in the sense of covering the entire decision space. Otherwise a 'bad' initial sample can cause the optimization outcome to deteriorate significantly; i.e., the variance of the deterministic methods is larger than the stochastic sampling methods. Although, LHS method is often used as a default sampling method in multiobjective optimization, the obtained results did not confirm it to outperform other sampling methods; one could use M-LHS or C-LHS instead as these sampling methods in many cases obtain better results. For high-dimensional problems in both objective and decision spaces deterministic methods led to large variability which resulted in significantly lower average algorithm performance with respect to stochastic sampling methods, where the maximin sampling method outperformed other stochastic methods though this advantage was not statistically significant.

In the near future, we will continue research on a larger set of test problems with a larger number of both objectives and decision variables possesing a variety of properties. Hopefully, this will provide a better insight and enable us to determine more concrete recommendations. Our conclusion for now though, is that choice of initial sample matters in higher dimensions. In this work, we have studied the algorithm performance with respect to the most widely used performance metric HV. However, it would be interesting to investigate the impact of sampling methods with respect to other metrics. Moreover, the question of what initial sample size one should use and how it affects the optimization results is also open. Clearly, there is a trade-off involved between the size of an initial sample and the number of evaluations used to run an optimization algorithm when dealing with costly real-world optimization problems. Thus, this research direction will be considered in near future as well.

Acknowledgments

This research was partly financially supported by the Linkage project "Optimizing experimental design for robust product development: a case study for high-efficiency energy generation" funded by Australian Research Council.

References

[1] Jeanne Altmann. Observational study of behavior: sampling methods. *Behaviour*, 49(3):227–266, 1974.

[2] Bernd Bischl, Jakob Bossek, Daniel Horn, and Michel Lang. *mlrMBO: Model-Based Optimization for mlr*, 2015. R package v1.0. `https://github.com/berndbischl/mlrMBO`.

[3] George EP Box and Norman R Draper. *Empirical model-building and response surfaces.* John Wiley & Sons, 1987.

[4] David Roxbee Cox and Nancy Reid. *The theory of the design of experiments.* CRC Press, 2000.

[5] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[6] Hongbing Fang and Mark F Horstemeyer. Global response approximation with radial basis functions. *Engineering Optimization*, 38(04):407–424, 2006.

[7] John H Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, 1960.

[8] John M Hammersley. Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.

[9] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[10] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477–506, 2006.

[11] Laura Ilzarbe, María Jesús Álvarez, Elisabeth Viles, and Martin Tanco. Practical applications of design of experiments in the field of engineering: a bibliographical review. *Quality and Reliability Engineering International*, 24(4):417–428, 2008.

[12] Ronald L Iman and WJ Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics-Simulation and Computation*, 11(3):311–334, 1982.

[13] Siwei Jiang, Yew-Soon Ong, Jie Zhang, and Liang Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *Cybernetics, IEEE Transactions on*, 44(12):2391–2404, 2014.

[14] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

[15] V Roshan Joseph and Ying Hung. Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, 18(1):171, 2008.

[16] Jayant R Kalagnanam and Urmila M Diwekar. An efficient sampling technique for off-line quality control. *Technometrics*, 39(3):308–319, 1997.

[17] André I Khuri and Siuli Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[18] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[19] Ladislav Kocis and William J Whiten. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)*, 23(2):266–294, 1997.

[20] Harold J Kushner. A versatile stochastic model of a function of unknown and time varying form. *Journal of Mathematical Analysis and Applications*, 5(1):150–167, 1962.

[21] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[22] Martin Meckesheimer, Andrew J Booker, Russell R Barton, and Timothy W Simpson. Computationally inexpensive metamodel assessment strategies. *AIAA journal*, 40(10):2053–2060, 2002.

[23] Douglas C Montgomery. *Design and analysis of experiments*. John Wiley & Sons, 2008.

[24] Max D Morris and Toby J Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402, 1995.

[25] Juliane Müller and Christine A Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144, 2014.

[26] Harald Niederreiter and NSF-CBMS Regional Conference on Random Number Generation. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM, 1992.

[27] Art B Owen. Controlling correlations in latin hypercube samples. *Journal of the American Statistical Association*, 89(428):1517–1522, 1994.

[28] Vinayak Govind Panse, Pandurang Vasudeo Sukhatme, et al. Statistical methods for agricultural workers. *Statistical methods for agricultural workers.*, 1954.

[29] Silvia Poles, Yan Fu, and Enrico Rigoni. The effect of initial population sampling on the convergence of multi-objective genetic algorithms. In *Multiobjective Programming and Goal Programming*, pages 123–133. Springer, 2009.

[30] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted\ mathcal {S}-metric selection. In *Parallel Problem Solving from Nature–PPSN X*, pages 784–794. Springer, 2008.

[31] John Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[32] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[33] Paul R Rosenbaum. *Observational studies*. Springer, 2002.

[34] Ranjit K Roy. *Design of experiments using the Taguchi approach: 16 steps to product and process improvement*. John Wiley & Sons, 2001.

[35] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

[36] ThomasJ. Santner, BrianJ. Williams, and WilliamI. Notz. Space-filling designs for computer experiments. In *The Design and Analysis of Computer Experiments*, Springer Series in Statistics, pages 121–161. Springer New York, 2003.

[37] Ilya M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational mathematics and mathematical physics*, (7):86–112, 1967.

[38] Daren S Starnes, Dan Yates, and David S Moore. *The practice of statistics*. Macmillan, 2010.

[39] David M Steinberg and William G Hunter. Experimental design: review and comment. *Technometrics*, 26(2):71–97, 1984.

[40] Ingrida Steponavičė, Rob J Hyndman, Kate Smith-Miles, and Laura Villanova. Efficient identification of the pareto optimal set. In *Learning and Intelligent Optimization*, pages 341–352. Springer, 2014.

[41] Boxin Tang. Selecting latin hypercubes using correlation criteria. *Statistica Sinica*, 8(3):965–977, 1998.

[42] Yoel Tenne. An analysis of the impact of the initial sample on evolutionary metamodel-assisted optimization. *Applied Artificial Intelligence*, 27(8):669–699, 2013.

[43] Yoel Tenne. Initial sampling methods in metamodel-assisted optimization. *Engineering with Computers*, pages 1–20, 2014.

[44] Yoel Tenne. Initial sampling methods in metamodel-assisted optimization. *Engineering with Computers*, pages 1–20, 2014.

[45] Tobias Wagner. *Planning and Multi-objective optimization of manufacturing processes by means of empirical surrogate models*. Vulkan, 2013.

[46] Xiaoqun Wang and Fred J Hickernell. Randomized halton sequences. *Mathematical and Computer Modelling*, 32(7):887–899, 2000.

[47] Jin Wu and Shapour Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2001.

[48] Frank Yates et al. Sampling methods for censuses and surveys. *Sampling methods for censuses and surveys*, 1949.

[49] Frank Yates et al. Sampling methods for censuses and surveys. *Sampling methods for censuses and surveys*, 54, 1960.

[50] Antanas Žilinskas. A statistical model-based algorithm for black-boxmulti-objective optimisation. *International Journal of Systems Science*, 45(1):82–93, 2014.

[51] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms – a comparative case study. In *Parallel Problem Solving from Nature - PPSN-V*, pages 292–301. Springer, 1998.

[52] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876. Springer, 2007.

[53] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.