Two-stage Methods for Multimodal Optimization

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r N a t u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik
von

Simon Wessing

Dortmund
2015

Tag der mündlichen Prüfung:
1. 7. 2015

Dekan:
Prof. Dr.-Ing. Gernot A. Fink

Gutachter:
Prof. Dr. Günter Rudolph, Technische Universität Dortmund
Jun.-Prof. Dr. Tobias Glasmachers, Ruhr-Universität Bochum

Contents

1 Introduction

The need of obtaining a set of good solutions in contrast to a single globally optimal solution of a multimodal problem is often mentioned in discussions of practical optimization [14, 1, 95]. The rationale behind this opinion is that a decision maker may have additional criteria to consider, which are not included in the optimization problem [14]. Such criteria may be side constraints or additional objective functions, and there may be various reasons not to incorporate them in the optimization problem. One argument is that different objectives could be incommensurable (e. g., in general, time is *not* money), and thus should not be aggregated into one function. In this case, a multiobjective problem formulation would be possible and probably appropriate [14]. However, sometimes this approach is not feasible because the expert knowledge constituting the additional criteria has not been formalized or the evaluation is more or less subjective. Then, it is often stated only informally that an optimization algorithm should be able to produce several different solutions of good quality for the formalized objective [1].

It is also possible that the objective functions of a multiobjective problem are very heterogeneous [52]. For example, one objective function could be multimodal and the other only linear. Such a situation could, e. g., appear if the first function evaluates the features of a product by running a computationally expensive simulator or doing physical experiments, and the second simply represents the production costs. This also implies that the former objective is a black box, while the latter one is available in analytical form. In summary, one objective may be more difficult than the other one in practically every regard. In this case it seems advisable to focus on the more difficult multimodal function with a single-objective approach. The linear function could perhaps be used for narrowing down the region of interest, but otherwise it would probably only appear in the post-processing of the solution set obtained from the optimization of the first objective. So, this set should possess a high diversity in decision space, to enable different evaluations according to the remaining criteria.

Another more specific example, where the task of identifying several local optima appears as an internal subproblem, are model-based optimization (MBO) algorithms that want to employ parallelization. In model-based optimization, a meta-model $\hat{f}(\boldsymbol{x})$ is built from a finite number of tuples $(\boldsymbol{x}, f(\boldsymbol{x}))$, where $\boldsymbol{x}$ is a point in the domain of f, and $f(\boldsymbol{x})$ is its associated objective value. The model is then used to determine one or more locations for the next exact evaluations of the objective function f. The goodness of a location is assessed with a so-called infill criterion, which is typically a multimodal function. For a batch-sequential version of the MBO algorithm, several distinct optima of the multimodal infill criterion have to be approximated per iteration [138, 19]. Although this work does not directly deal with

meta-modeling, we will encounter several references to this field, because we are borrowing ideas from it. Likewise, some of the methods developed here should also be relevant for meta-modeling.

The area containing single-objective problems with the need to identify a set of solutions is nowadays called multimodal optimization (MMO) in the area of evolutionary computation. The term may have been originally coined by Beasley et al. [14], probably as a short form of "multimodal function optimization", as used by Goldberg and Richardson [50]. In this work we want to formally define MMO, corresponding performance measures, and explicitly assess optimization algorithms under the objective of obtaining a set of good local optima.

1.1 Problem Statement

In the following, we will assume to have a deterministic objective function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} = [\boldsymbol{\ell}, \boldsymbol{u}] \subset \mathbb{R}^n$ is the *search space* or *region of interest* (ROI) and $n \in \mathbb{N}$ is the fixed number of decision variables. The vectors $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)^\top$ and $\boldsymbol{u} = (u_1, \ldots, u_n)^\top$ are called the lower and upper bounds of $\mathcal{X}$, respectively. The function f is assumed to be multimodal and analytically unknown. Naturally, also analytic gradient information is not available in this case. For these reasons we call f a *black-box* function.

Although they did not use the name yet, a multimodal optimization problem is in principle already defined by Törn and Žilinskas [143, pp. 2–3]. The following definition is based on their formulations.

Definition 1 (Multimodal minimization problem)**.** *Let there be ν local minima $f_1^*, \ldots, f_\nu^*$ of f in $\mathcal{X}$. If the ordering of these optima is $f_{(1)}^* < \cdots < f_{(l)}^* < h < \cdots < f_{(\nu)}^*$, a multimodal minimization problem is given as the task to approximate the set $\bigcup_{i=1}^{l} X_{(i)}^*$, where $X_{(i)}^* = \{\boldsymbol{x} \in \mathcal{X} \mid f(\boldsymbol{x}) = f_{(i)}^*\}$.*

The variable h in this definition is simply a threshold to potentially exclude some of the worse optima. For simplicity, we will always assume $h = \infty$ in this work, so we will be interested in all local optima of a problem. This case is described in [114, Sec. 5.1] as the task of recovering "all known" optima of a test problem. Global optimization can be seen as a special case of multimodal optimization, where we are only interested in finding the $X_{(1)}^*$ corresponding to the global optimum $f^* = f_{(1)}^*$. This problem is closely related to the black-box levelset approximation problem [40]. However, usually it is sufficient to find just one $\boldsymbol{x}^* \in X_{(1)}^*$ in practical applications referring to global optimization. Even this problem is mathematically unsolvable [143, p. 6]. Therefore, we will take a pragmatic approach to the problem by trying to obtain the best possible performance (as defined in Section 2.1.3) with restricted resources, i. e., a finite number of objective function evaluations. Further assumptions concern the properties of the objective function: We will restrict our considerations to problems for which a positive constant ε can be specified, so that the distance between any two optimal positions is larger than ε. This implies a finite

number of optimal positions [119], so we could alternatively formulate the problem as the task to find the k best optimal positions of the objective function. It also means that each optimal position is surrounded by its own attraction basin.

Definition 2 (Attraction basin, [141])**.** *For the position $\boldsymbol{x}_i^* \in \mathcal{X}$ of an optimum,* $\text{basin}(\boldsymbol{x}_i^*) \subseteq \mathcal{X}$ *is the largest set of points such that for any starting point $\boldsymbol{x} \in$* $\text{basin}(\boldsymbol{x}_i^*)$ *the infinitely small step steepest descent algorithm will converge to $\boldsymbol{x}_i^*$.*

Realistic functions should possess some smoothness, that is, the probability $p_i = \text{vol}(\text{basin}(\boldsymbol{x}_i^*))/\text{vol}(\mathcal{X})$ to find $\boldsymbol{x}_i^*$ with an ideal descent algorithm started from a random uniform point should be significantly greater than zero [143, pp. 7–10]. Here, $\text{vol}(\cdot)$ denotes the Lebesgue measure of the sets [18, pp. 171–181] and p_i obviously is the relative size of an attraction basin in comparison to the whole search space. In this simplified model, the probability of finding $\boldsymbol{x}_i^*$ can naturally be amplified by carrying out N local searches from random uniform starting points. The corresponding formula for this probability P reads $P = 1 - (1 - p_i)^N$ [143, p. 86]. By solving for N, we obtain the estimate

$$N = \left\lceil \frac{\ln(1 - P)}{\ln(1 - p_i)} \right\rceil \tag{1.1}$$

as the required number of points for finding $\boldsymbol{x}_i^*$ with probability P [140]. The strong influence of p_i on the problem difficulty can be illustrated by inserting some numbers into (1.1): if we require $P = 0.95$ and assume $p_i = 0.01$, the equation yields $N = 299$. Decreasing p_i by a factor increases N by the same factor, e. g., a value of $p_i = 0.001$ already results in $N = 2995$. The problem of obtaining all optima is of course generally more difficult and also the math for estimating the corresponding numbers becomes more complicated. For details, we refer to [114, Sec. 3.3].

The cost of one objective function evaluation is another important property of the problem, as it decides on the number of possible function evaluations. In this work, we assume that this cost is constant, i. e., all function evaluations cost the same. The number of function evaluations in turn influences how much computational effort should be invested to determine the location of the tested points. If only very few function evaluations can be afforded, the amount of additional computations may be very high, because sample locations must be chosen carefully. The assumed budgets in the literature vary widely and have considerable influence on which optimization algorithm obtains a good performance or is even applicable.

Table 1.1 shows how some budgets are associated with research areas and applications. The interesting budgets for this work are highlighted in lime green. This focus has been set by subjectively considering what may be possible in real-world optimization, e. g., for design problems in engineering, and what is manageable in benchmarking. This is also a setting where the common black-box optimization benchmarking (BBOB) practice [44] of measuring consumed resources simply as the number of objective function evaluations seems still admissible, as the assumption of expensive function evaluations in relation to the overhead of an optimization algorithm becomes rather unlikely with larger budgets [38]. If budgets are smaller,

Table 1.1: Different magnitudes for the number of function evaluations N_f.

Magnitude	Application
$n \cdot 10^1$	Initial designs in model-based optimization [70]
$n \cdot 10^2$	Expensive optimization
$n \cdot 10^3$	
$n \cdot 10^4$	Budget of the CEC 2005 competition [135]
$n \cdot 10^5$	
$n \cdot 10^6$	Budget of the black-box optimization benchmark (BBOB) [44]

then probably meta-models should be used [70] and more emphasis should be put on parallelization, if possible.

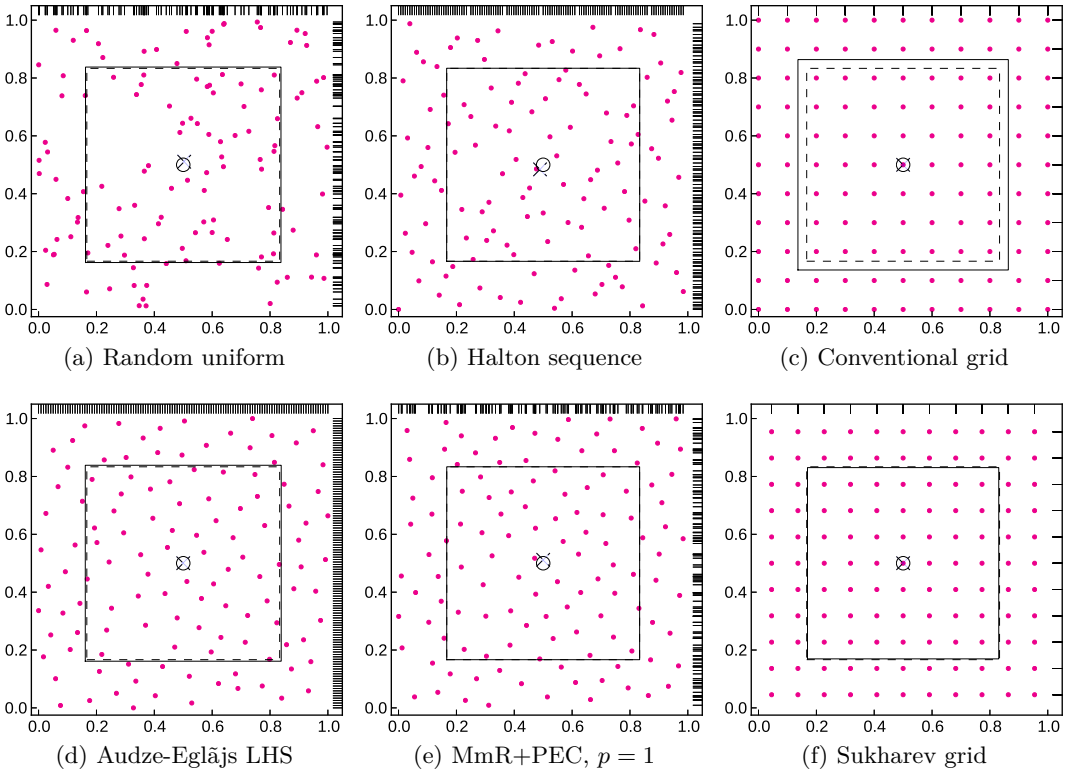
Törn and Žilinskas [143, p. 93] write

> "[...] one can expect that the practitioner would undertake a large number of experiments in order to be convinced that the problem is thoroughly analyzed. The optimization would hardly be stopped until all research resources have been spent."

This quotation shall serve as justification to consider a fixed-budget scenario, i. e., measuring optimization performance after a certain amount of resources has been spent. (Although this does not mean that only a single fixed budget size is investigated.)

1.2 Average Case versus Worst Case

A large part of this work deals with (uniform) sampling algorithms and the properties of the point sets produced by them. Some examples of such point sets are presented in Figure 1.1. With exception of Figure 1.1d with $N = 120$, each set contains $N = 121$ points. (The corresponding algorithms will be discussed in Chapter 4.) One permissible quality feature is the distribution of one-dimensional projections of the points, which should be as uniform as possible. The set in Figure 1.1d (taken from [12]) is perfect in this regard. Another important feature is the uniformity in the original, high-dimensional space. In Section 4.2 we will see that deterministic quasirandom sequences (as in Figure 1.1b) with low discrepancy (i. e., low deviation from uniformity) provide a worst-case error bound for the integration error in numerical integration. Corresponding randomized variants provide a variance reduction compared to random uniform sampling (Figure 1.1a) [88]. These results are relevant for global and multimodal optimization, because estimating the size of an attraction basin is nothing more but numerical integration. In practice, we are of course not interested in the actual basin size, but simply in discovering all existing basins, and thus all existing optima. This requirement is also served, because reducing the variance of basin size estimates encompasses reducing the risk of missing any



(a) Random uniform (b) Halton sequence (c) Conventional grid

(d) Audze-Eglājs LHS (e) MmR+PEC, $p = 1$ (f) Sukharev grid

Figure 1.1: Examples of different sampling methods in two dimensions. One-dimensional projections of the points are indicated at the top and right axis. The cubes, cross, and circle indicate certain uniformity-related properties of the point sets, which will be explained in Chapter 3.

basin completely. Consequently, our multimodal optimization performance can be improved by employing variance reduction techniques, e. g., some kind of stratified sampling. Also Törn [140] argues that if we are using q strata, the required number of points N_{strat} to find optimum $\boldsymbol{x}_i^*$ with probability P can be bounded by

$$\left\lceil \frac{q \ln(1-P)}{\ln(1-qp_i)} \right\rceil \leq N_{\text{strat}} \leq \left\lceil \frac{\ln(1-P)}{\ln(1-p_i)} \right\rceil .$$

This result implies an improvement over random uniform sampling, because the upper bound is identical to (1.1).

However, theory predicts and experiments confirmed that variance reduction is only possible for sufficiently smooth functions [100]. In case of high-frequency functions, the performance of any uniform sampling method should be identical to that of random uniform sampling [100]. In summary, this means that in a black-box scenario, no deterioration of performance in comparison to random uniform sampling is possible by using stratified sampling, as long as uniformity is ensured. Unfortunately, creating and measuring uniformity in high dimensions is not as trivial as it sounds (see Section 2.1.1 and Chapter 4).

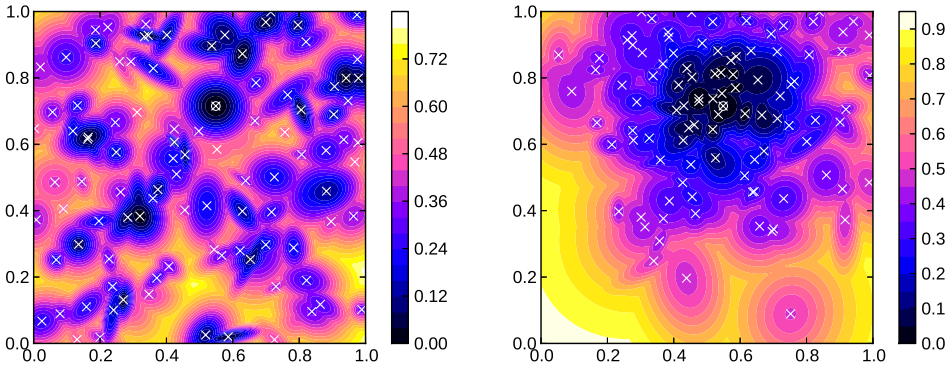


Figure 1.2: Examples of multimodal optimization problems with $\nu = 100$ optima. The left problem exhibits no global structure, the right one contains one large funnel. Every optimum is marked with a cross, the global one is encircled.

As soon as we begin sampling, the black-box assumption is weakened, because we are gaining more and more information about the problem. Special features of the global topology as the distribution of optima locations and the correlation of their objective values could be exploited to improve performance. The importance of the former aspect for problem difficulty was already recognized by Törn and Žilinskas [143, p. 11]. If we know, for example, that optima are embedded in a so called funnel structure [86] (see Figure 1.2), it would be advisable to deviate from uniformity and to sample with a higher density in better areas. By doing this, we are sacrificing the worst-case bound in favor of better average-case performance, and this is what is typically done in optimization [109], either by using local search or an adaptive sampling. The next section will elaborate on the kind of interplay between global and local approaches that is considered in this work.

1.3 The Two-stage Optimization Paradigm

Törn and Žilinskas [143, pp. 16–21] summarize several classification schemes of the early literature for global optimization algorithms. However, the criteria for these classifications seem often arbitrary and therefore no single ideal classification can be found. So, the approach here is to define a simple high-level concept that covers a large class of optimization algorithms. The proposed concept is based on the observation that many successful meta-heuristics are composed of several modules, which are quite sophisticated themselves. The complexity of these meta-heuristics often hinders the analysis of the individual components, i. e., it is difficult to assess how much influence a component has on the overall performance. Possibly the most simple representative of these compositions is that of two alternating stages [85]. Naturally, this idea is not restricted to multimodal optimization, but could, e. g., be also applied to multiobjective optimization or completely unrelated topics. And

indeed, it has surfaced multiple times in different contexts. For example, Jones [69] gives a survey of model-based two-stage methods for optimization. He distinguishes the two stages as follows:

1. Fit a response surface to training data (including estimation of the model's parameters).

2. Use the surface to compute new search points under the assumption that the parameters are correct.

This approach has evolved from response surface methodology, which was originally applied to physical experiments (with a human in the loop) [34, pp. 7–11]. Another example is the restart variant of the covariance matrix adaptation evolution strategy (CMA-ES) [9]. The CMA-ES belongs to the class of evolutionary algorithms (EA), which are nature-inspired optimization methods [16]. They maintain a set of solutions during the optimization and create new candidate solutions by randomly combining and modifying the current ones. This process is typically described in the language of biology, by saying that a population of individuals is maintained and offspring are created from parent solutions by recombination and mutation. In each generation, the best individuals are selected to form the parent population of the next generation. The later development of the CMA-ES was driven by the question what to do in situations when the local search has converged, but part of the budget is still available. If the performance measurement does not reward savings in budget, a natural idea is to just restart the algorithm with a random starting point. In this case, the sampling of the random point trivially represents the other stage.

A "stage" can be defined abstractly as an algorithm that takes three inputs: the maximal number of function evaluations that may be consumed by this very call, a (possibly empty) set of individuals, and the optimization problem. We are using the terms *individual* and *problem* here instead of point and function, respectively, to indicate that we are dealing with objects that may contain additional meta-information. The object-oriented view emphasizes that the abstract concept can be instantiated in multiple ways. The outputs are another set of individuals and the consumed budget (the number of function evaluations actually used). The stage must assure that the maximal budget is never exceeded.

Törn and Žilinskas [143, p. 14] mention that most global optimization algorithms consist of a *global stage* and a *local stage*. Although the distinction between these two is rather fuzzy, it shall be a guiding theme of this work. Algorithm 1 sketches a whole abstract two-stage method. It basically consists of a loop to ensure the consumption of the whole budget and the two interacting stages in the loop. While the code indicates that we enter the global stage before the local stage, this can be the other way around as well. At the end of each iteration, an archive is updated with the new individuals. This function is called "combine" to emphasize that it does not necessarily have to be a set union. Also note that we define the communication between the stages as being based on the exchanged individuals, in contrast to Jones' definition [69]. The algorithms he considered do in principle also match our

Algorithm 1 General two-stage optimization framework

Input: budget B, archive $\mathcal{A}$, problem F
Output: solution/approximation set

```
while B > 0 do
    P, c_g ← globalStage(B, A, F)
    B ← B − c_g                    // reduce budget by costs c_g of global stage
    Q, c_ℓ ← localStage(B, P, F)
    B ← B − c_ℓ                    // reduce budget by costs c_ℓ of local stage
    A ← combine(A, Q)
end while
return filter(A)                   // potential subset selection
```

definition, only the distinction to what he calls "one-stage" methods disappears. Schoen [127] defines "two-phase" optimization methods in a very similar way as we do in Algorithm 1, but applies additional requirements to the local stage. First of all, an explicit local search algorithm has to be present. Secondly, he emphasizes that the local stage does not employ all the information available in $\mathcal{A}$, but only certain selected starting points. This is also reflected in Algorithm 1 by giving the local stage the input $\mathcal{P}$. Thirdly, he demands that no other information than the final result of the local search may be fed back to the global stage. While this last behavior often seems to be a convenient choice, because it focuses on the "high-quality information", we do not necessarily want this property to be enforced.

As already said, our definition encompasses a very broad class of algorithms, perhaps even to a point beyond recognition. For example, an evolutionary algorithm would fit into this paradigm if it uses both recombination and mutation. We could view the offspring creation by parent selection and recombination as the global stage, mutation and the evaluation of the offspring population as the local stage, and the survivor selection step as combination of $\mathcal{A}$ and $\mathcal{Q}$. On the other hand, simulated annealing and an evolutionary algorithm without recombination would not fit into the two-stage paradigm [127]. However, the concept should be more interesting on a higher level anyway. Figure 1.3 makes a further categorization of stages into four groups. As in EAs, the global stage often does not use any evaluations of the objective function. Another such example would be a uniform sampling algorithm, which is also in the category of model-free, cost-free stages. The corresponding modules in model-based optimization are methods using infill criteria to determine candidate points. Stages with function evaluations are often whole optimization algorithms, although the consumed budget may vary greatly.

If one wants to emphasize the local-search aspect, the methods described by Jones can only be classified as a module of a two-stage method, because they lack an explicit local search and need relatively few function evaluations. However, also the Restart-CMA-ES is a borderline case, because drawing a random uniform starting point is just too trivial. As a compromise we can conclude that we are interested in methods somewhere in between these two extremes.

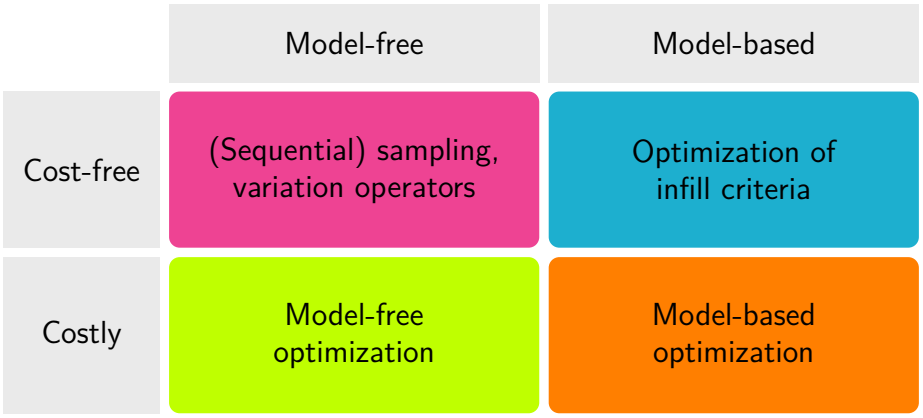


Figure 1.3: Categorization of possible modules for global and local stages.

1.4 Distances and Neighbors

As already mentioned, sampling algorithms play an important role in this work. We will be dealing a lot with methods to generate point sets and methods to analyze their spatial properties (e. g., diversity). Thus, a core notion for us is the distance between two points. This distance is also a building block of many MMO performance measures and optimization algorithms.

Definition 3 (Minkowski distance, L_p distance)**.** *The distance between two points* $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ *and* $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ *is defined as*

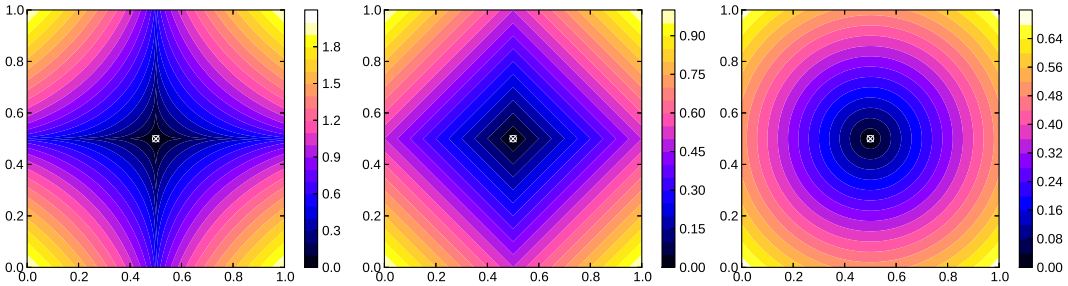
$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} .$$

For $1 \leq p < \infty$, $d(\boldsymbol{x}, \boldsymbol{y})$ is a metric [18, p. 242]. However, it is seldom relevant for us that the Minkowski inequality (triangle inequality) $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ is violated for $0 < p < 1$, so also these values are in principle possible choices for us. Other notable properties of distance metrics are non-negativity ($d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$), identity of indiscernibles ($d(\boldsymbol{x}, \boldsymbol{y}) = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$), and symmetry ($d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$). We will usually write $d(\boldsymbol{x}, \boldsymbol{y})$ to emphasize the independence of p, and $\|\boldsymbol{x} - \boldsymbol{y}\|_p$ if a certain p is assumed. If nothing else is said, the Euclidean distance with $p = 2$ is meant. Figure 1.4 shows some examples of distances for different values of p. In practice, we are often interested in short distances, that is, distances to neighbors of some point.

Definition 4. *The distance to the nearest neighbor of* $\boldsymbol{x} \in \mathcal{X}$ *in* $\mathcal{P} \subset \mathcal{X}$, $|\mathcal{P}| < \infty$, *is defined as*

$$d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P}) = \min\{d(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{P} \setminus \{\boldsymbol{x}\}\} . \tag{1.2}$$

The nearest neighbor itself shall be denoted $\mathrm{nn}(\boldsymbol{x}, \mathcal{P})$ and is obtained by using arg min instead of min in (1.2). Throughout this work, $\mathcal{P}$ will always denote a discrete



(a) $p = 0.5$ (b) $p = 1$ (c) $p = 2$

Figure 1.4: Distances from $(0.5, 0.5)^\top$ for different values of p.

set of points. Then, also the average distance to the k nearest neighbors may be a sensible characteristic.

Definition 5. *Let $\boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(N)}$ be the ordered elements of $\mathcal{P}$, so that $d(\boldsymbol{x}, \boldsymbol{y}_{(1)}) \leq \cdots \leq d(\boldsymbol{x}, \boldsymbol{y}_{(N)})$ for some $\boldsymbol{x} \notin \mathcal{P}$. Then, the average distance to the k nearest neighbors is denoted*

$$d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P}, k) = \frac{1}{k} \sum_{i=1}^{k} d(\boldsymbol{x}, \boldsymbol{y}_{(i)}) \ .$$

Finding the nearest neighbors of points in $\mathbb{R}^n$ is a common problem and sophisticated data structures exist to accelerate this task (cf. [151] for an overview). However, these data structures are affected by the *curse of dimensionality*, which means that in high dimensions they yield no improvement over the naïve approach of enumerating all neighbor candidates. Weber et al. [151] show experimentally, that the break-even point may already appear at only $n = 10$ dimensions. Related results of Beyer et al. [17] additionally show that under broad conditions, the difference between the Euclidean distances of a random uniform query point to its farthest and nearest neighbor in a point set vanishes with increasing dimension. This result puts a lot of doubt on approximate nearest neighbor data structures. Subsequently, Aggarwal et al. [5] found out that values of $p \leq 1$ can prevent this effect. Their result serves as incentive to also investigate such distances in this work. However, doing so further hinders the use of acceleration techniques, as they typically require metric distances. Thus, we will only use the linear search through all points to find nearest neighbors.

1.5 Objectives of this Work

Two-stage optimization algorithms are an attractive concept for global and multimodal optimization due to their simplicity and demonstrated performance [114, 2, 1]. Especially the Restart-CMA-ES has proved its competitiveness in global optimization tasks [53]. The main focus of this work will be on improving the global stages, as local search is already a thoroughly researched topic.

Although our investigations should also largely apply to global optimization, the primary interest of this work is in multimodal optimization, as it seems more challenging and interesting from a practical perspective. While meta-modeling and model-based optimization are not directly a part of the work, we will often discuss concepts with a view to them, because many requirements are shared between the areas. Model-based approaches would be especially promising on low-dimensional problems (e. g., $n \leq 10$) with small budgets (e. g., $N_f \leq 10^3 n$). For larger budgets, the approach is rather unattractive, because fitting the model becomes a run time bottleneck. However, there also is a big discrepancy between theory and practice in the area of small budgets: On the one hand, a lot of emphasis is put on using infill criteria that guarantee *asymptotic* convergence to the global optimum, on the other hand the considered budgets of function evaluations are extremely small.

As already indicated in Section 1.1, we will deal with larger budgets and consequently with model-free optimization algorithms. In summary, the following three research questions shall receive the most attention:

- How can performance be measured in multimodal optimization?
- Which is the best sampling algorithm according to the chosen performance measures, especially in high dimensions?
- How can the two two-stage approaches *restarted local search* and *clustering method* be improved and how do they perform in a comparison?

As most results in this work are experimental, we will establish the fundamentals for benchmarking in multimodal optimization in Chapter 2. This encompasses the definition of quality indicators, performance measures, and test problems. As the name implies, quality indicators are used for measuring the quality of the solutions to the optimization problem. A performance measure is usually regarded as something more general, especially incorporating the required time to obtain the solution. Test problems are required because real-world problems are too difficult to work with in large, controlled experiments.

In Chapter 3 we talk about useful characteristics of point sets designated for incorporation in the global stages of MMO algorithms. These characteristics as, e. g., diversity, irregularity, or the mean distance to the boundary are also highly related to space-filling experimental designs [126, pp. 121–161], which are an important building block for meta-modeling.

In Chapter 4, an overview of different sampling algorithms, collected from various research disciplines, is given. Every algorithm is evaluated with appropriate summary characteristics of the previous chapters and two of them, namely improved latin hypercube sampling and the part-and-select algorithm, are enhanced in terms of run time. Finally, a new algorithm called *maximin reconstruction* (MmR) is proposed, which has the advantage that it can be configured very flexibly. To conclude the chapter, the most interesting algorithms are compared regarding their MMO performance in a first basic experiment.

In Chapter 5, we finally arrive at the part of the work that deals with complete two-stage optimization algorithms. After a short discussion of relevant approaches in this area, two algorithm classes are investigated experimentally regarding their improvability by using MmR as global stage, before the chapter is closed with a comparison between the best variants of these two classes. Chapter 6 closes the work with conclusions and an outlook.

2 Benchmarking

This chapter first surveys general diversity indicators and then specific quality indicators for multimodal optmization. Afterwards, it is discussed how these indicators can be incorporated into a performance assessment workflow for multimodal optimization. As we will see, there are some hidden pitfalls in the process that can cause misleading results. Then, an overview of artificial test problems is given, followed by a description of the test problem generator used for our experiments. The chapter concludes with some general remarks on experimentation, trying to explain the mindset that governed this work.

2.1 Quality Indicators for Multimodal Optimization

This section deals with measuring the quality of a multiset of points $\mathcal{P} \subset \mathcal{X}$ with $|\mathcal{P}| = N < \infty$. This set may for example be the result or the initialization of some optimization algorithm. While the diversity indicators in Section 2.1.1 are versatile in application, Section 2.1.3 deals especially with the assessment of the outcomes of multimodal optimization. The survey is an extended version of earlier ones in [117] and [155].

With the denomination *quality indicator*, we exclusively mean unary quality indicators in this work, which are simply mappings from populations $\mathcal{P}$ to real numbers. In other words, we could say that we are interested in numerical summary characteristics to describe $\mathcal{P}$. The term *indicator* is used to show that the mathematical properties associated to metrics or measures are not necessarily given. The name is adopted from multiobjective optimization [160], because of its close similarity to MMO, especially concerning the virtues of *a-posteriori* methods. It should also be stressed that in the following, we will require that no gradient information and no additional function evaluations may be used for the assessment.

2.1.1 Diversity Indicators

Measuring the diversity of a multiset $\mathcal{P}$ is a common problem in many research areas. We will encounter approaches from biology, physics, operations research, computer experiments, and numerical integration. Although "diversity" is an abstract concept and can mean different things, there are surprisingly many similarities between approaches that emerged from different applications. We will try to relate them to each other by using a few axiomatic properties originating from Weitzman [152]. He formulates several desirable properties for diversity measures, of which Solow and Polasky [131] selected three that seemed especially natural to them. This subset has

also been adopted by Ulrich et al. [144]. As these requirements have been developed in the context of biological diversity, the term *species* is used for the elements of $\mathcal{P}$ in the following definitions. In our application, however, a species is simply some point $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$. Weitzman required neither the triangle inequality nor identity of indiscernibles for his axiomatic treatment. Instead of the latter one, the relaxed condition $d(\boldsymbol{x}, \boldsymbol{x}) = 0$ was assumed [152]. However, some of the diversity indicators in the following survey do assume the triangle inequality.

The consideres axioms are as follows:

Axiom 1 (Monotonicity in species)**.** *If a species $t \notin \mathcal{P}$ has a positive distance to all species in $\mathcal{P}$, then adding it to $\mathcal{P}$ may not decrease the diversity:*

$$\forall s \in \mathcal{P} : d(s,t) > 0 \Rightarrow D(\mathcal{P}) \leq D(\mathcal{P} \cup \{t\}) \; .$$

Axiom 2 (Twin property TP1)**.** *Diversity should neither be increased nor decreased by the addition of a species $t \notin \mathcal{P}$ that is identical to a species already in the set:*

$$D(\mathcal{P} \cup \{t\}) = D(\mathcal{P}) \Leftrightarrow \exists s \in \mathcal{P} : d(s,t) = 0 \; .$$

Axiom 3 (Monotonicity in distance)**.** *Let $|\mathcal{P}| = |\mathcal{P}'| \geq 2$. For a one-to-one mapping of $\mathcal{P}$ onto $\mathcal{P}'$ so that no distance is decreasing and at least one increasing, the diversity of $\mathcal{P}'$ may not be smaller:*

$$\begin{aligned} &\forall i, j \in \{1, \ldots, |\mathcal{P}|\} : d(s_i, s_j) \leq d(s'_i, s'_j) \\ \wedge\; &\exists i, j \in \{1, \ldots, |\mathcal{P}|\} : d(s_i, s_j) < d(s'_i, s'_j) \Rightarrow D(\mathcal{P}) \leq D(\mathcal{P}') \; . \end{aligned}$$

Monotonicity in species and TP1 are also discussed informally by Bursztyn and Steinberg [21] in the context of computer experiments. Note that the three properties together are not sufficient to rule out less sensible indicators. For example, let $\mathrm{supp}(\mathcal{P})$ be the support of $\mathcal{P}$, i. e., the set where the duplicates of $\mathcal{P}$ have been removed. Then, it is easy to see that $|\mathrm{supp}(\mathcal{P})|$, which is an ecological diversity index called *species richness* [64], fulfills all three properties, although it does not take any distances into account. Another observation is that the original definitions in [152] are not always reproduced faithfully. Axiom 1 is a slightly generalized variant by Solow and Polasky [131]. Ulrich et al. [144] require strict monotonicity in species instead of simple monotonicity. Finally, we observe that strict monotonicity in distance would be the requirement where $|\mathrm{supp}(\mathcal{P})|$ fails.

On the other hand, Weitzman argues that diversity indicators, which do not conform to all properties, simply "do not work" [152, p. 376]. The following survey, however, demonstrates that there are many applications where a subset of the properties is sufficient, and such indicators are indeed frequently used. Finally, also run time constraints may definitely bias our choice towards less sophisticated ones.

In the following, twelve diversity indicators are listed in an order that seemed to support easy transitions in reading. Each indicator is discussed mostly based on practical considerations and sometimes a new name is assigned if the old one seems unintuitive. Afterwards, the axiomatic view of Weitzman is used to obtain a general overview and identify relations between the indicators.

Sum of Distances (SD)

Probably the first indicator that comes to mind of everyone that thinks about diversity is the sum of all distances in the point set, which is also known as the N-dispersion-sum measure [95]. However, this figure is criticized by [131, 144, 95] as being inappropriate for measuring diversity, because it only rewards the spread, but not the diversity of a population. Therefore, it should not be used. However, if it is used, we suggest to take the square root of the sum,

$$\mathrm{SD}(\mathcal{P}) := \sqrt{\sum_{i=1}^{N} \sum_{j=i+1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)}\,,$$

to obtain indicator values of reasonable magnitude.

Average Distance (AD)

Izsák and Papp [64] show that the average distance

$$\mathrm{AD}(\mathcal{P}) = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N} \sum_{j=i+1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

does not provide monotonicity in species, while SD does. This is an interesting result, as the division by N instead of $\binom{N}{2}$ does not seem to change the indicator's properties in comparison to SD.

Sum of Distances to Center of Mass (SDCM)

Ulrich et al. [145] propose an indicator offering the same properties as SD (given that d satisfies the triangle inequality), but with linear run time. First, the population's center of mass $\bar{\boldsymbol{c}}_{\mathcal{P}} = 1/N \sum_{i=1}^{N} \boldsymbol{x}_i$ has to be calculated. To obtain the indicator value, the distances of all points to the center of mass are summed:

$$\mathrm{SDCM}(\mathcal{P}) = \begin{cases} 0 & \text{if } \mathcal{P} = \emptyset\,, \\ 1 + \sum_{i=1}^{N} d(\boldsymbol{x}_i, \bar{\boldsymbol{c}}_{\mathcal{P}}) & \text{else.} \end{cases}$$

Analogously, an average distance to the center of mass (ADCM) could be defined. The distinction of cases and adding of 1 are used to handle cases where $N < 2$. This workaround could be applied to several other indicators, too, but we omit it elsewhere because we are interested in much larger sets anyway.

Minimal Distance (MD)

The indicator $\mathrm{MD}(\mathcal{P}) = \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{P}, i \neq j\}$ is another basic indicator and thus goes by many names. Emmerich et al. [40] call it minimal gap, Damelin et al. [29] separation distance, Illian et al. [63, p. 58] hard-core distance, Meinl et

al. [95] the N-dispersion measure. Algorithms attempting to maximize this criterion are usually called *maximin* approaches for short. As all other distances except the minimal one are disregarded, often regularizations are sought, which are easier to optimize but asymptotically yield the same result [118].

Sum of Distances to Nearest Neighbor (SDNN)

Meinl et al. [95] propose the "N-dispersion-min-sum measure" as a way to combine the advantages of MD and SD. This indicator calculates the sum of distances to the nearest neighbor:

$$\mathrm{SDNN}(\mathcal{P}) := \sum_{i=1}^{N} d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{P})\,.$$

In contrast to SD, SDNN penalizes the clustering of solutions, because only the nearest neighbor of every point is considered. Emmerich et al. [40] mention $\frac{1}{N}\,\mathrm{SDNN}(\mathcal{P})$ as the "arithmetic mean gap". We omit the averaging here to avoid potential penalizing of larger sets. Suppose, for example, we have $\mathcal{P} = \{\boldsymbol{x}, \boldsymbol{y}\}$ with $d(\boldsymbol{x}, \boldsymbol{y}) = 1$. Now we add a point $\boldsymbol{z}$ to the set $\mathcal{P}$ so that $d(\boldsymbol{y}, \boldsymbol{z}) = 0.6$ and $d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P})$ is unchanged. Then the indicator value of $\mathrm{SDNN}(\mathcal{P})$ increases from 2 to 2.2, while $\frac{1}{N}\,\mathrm{SDNN}(\mathcal{P})$ decreases from 1 to $\frac{2.2}{3}$. However, it is also possible to construct situations where adding a new point to the set decreases the indicator value for both variants.

Product of Distances to Nearest Neighbor (PDNN)

Analogous to SDNN, PDNN is defined as

$$\mathrm{PDNN}(\mathcal{P}) := \prod_{i=1}^{N} d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{P})\,.$$

Also this indicator is inspired by the geometric mean gap $\mathrm{PDNN}(\mathcal{P})^{1/N}$, which is defined by Emmerich et al. [40]. It puts a higher focus on the regularity of the point set than SDNN, but simultaneously has the disadvantage that the indicator value is always zero if there is a duplicate point in the set.

Weitzman Diversity (WD)

Weitzman [152] developed a diversity indicator recursively defined as

$$\begin{aligned}\mathrm{WD}(\mathcal{P}) &= \max_{\boldsymbol{x}\in\mathcal{P}} \{\mathrm{WD}(\mathcal{P}\setminus\{\boldsymbol{x}\}) + d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P}\setminus\{\boldsymbol{x}\})\} \\ &= d(\boldsymbol{g}, \boldsymbol{h}) + \max\{\mathrm{WD}(\mathcal{P}\setminus\{\boldsymbol{g}\}), \mathrm{WD}(\mathcal{P}\setminus\{\boldsymbol{h}\})\}\,,\end{aligned}$$

where $(\boldsymbol{g}, \boldsymbol{h})$ is the closest pair in the respective set. The base case is

$$\forall \boldsymbol{x} \in \mathcal{P} : \mathrm{WD}(\{\boldsymbol{x}\}) = 1\,.$$

While this indicator has interesting theoretical properties, it unfortunately has a runtime of $O(2^N)$ and is thus of little practical use for our application.

Solow-Polasky Diversity (SPD)

Solow and Polasky [131] developed an indicator to measure a population's biological diversity and showed that it has superior theoretical properties compared to the sum of distances and other indicators. Monotonicity in distance is fulfilled as long as the triangle inequality holds. Ulrich et al. [144] discovered the indicator's applicability to multiobjective optimization. They also verified the inferiority of the sum of distances experimentally by directly optimizing the indicator values. To compute this indicator for $\mathcal{P}$, it is necessary to build an $N \times N$ correlation matrix $\mathbf{R}$ with entries $r_{ij} = \exp(-\theta d(\boldsymbol{x}_i, \boldsymbol{x}_j))$. The indicator value is then the scalar resulting from

$$\mathrm{SPD}(\mathcal{P}) := \mathbf{1}^\top \mathbf{R}^{-1} \mathbf{1}\,,$$

where $\mathbf{1} = (1, \ldots, 1)^\top$. This measure also appears in [21] as a part of an entropy criterion. It is advisable to use the pseudo-inverse in practice, to alleviate numerical problems. As the (numerically stable) matrix inversion requires time $O(N^3)$, the indicator is only applicable to relatively small sets, although Ulrich et al. [146] show how update operations can be carried out more efficiently. The indicator also requires a user-defined parameter θ, which depends on the size of the search space. These properties make this theoretically appealing indicator rather unattractive in practice.

Average of Inverse Distances (AID)

Santner et al. [126, p. 139] define the so-called average distance criterion function

$$m_q(\mathcal{P}) = \left(\frac{1}{\binom{N}{2}} \sum_{h=1}^{N} \sum_{i=h+1}^{N} \left[\frac{d_{\max}}{d(\boldsymbol{x}_h, \boldsymbol{x}_i)} \right]^q \right)^{1/q} \tag{2.1}$$

to measure how diverse the points in an experimental design are. This indicator is obviously related to the harmonic mean of all distances. Putting $d(\boldsymbol{x}_h, \boldsymbol{x}_i)$ into the denominator causes it to be undefined for point sets that contain duplicates. For optimization, it is convenient to assume $m_q(\mathcal{P}) = \infty$ in this case. The normalization factor $d_{\max} = d(\boldsymbol{u}, \boldsymbol{\ell})$ denotes the largest possible distance between points in the search space. If this value is unavailable, it is possible to insert another constant (e. g., 1), albeit sacrificing comparability of indicator values across different numbers of variables. For $q \to \infty$, minimizing (2.1) becomes equivalent to maximizing the minimal distance between all pairs of design points [126, p. 139]. Interestingly, a simpler but otherwise identical formula is used in potential theory to describe the energy level of a point set. This is the Riesz energy, defined as

$$E_q(\mathcal{P}) = \sum_{h=1}^{N} \sum_{i \neq h} \frac{1}{d(\boldsymbol{x}_h, \boldsymbol{x}_i)^q}\,. \tag{2.2}$$

Hardin and Saff [57] show that for n-dimensional manifolds, asymptotically uniformly distributed point sets minimize the Riesz energy if $q \geq n$. If q is chosen

smaller, the optimal point density increases towards the outer regions of the manifold. Thus, we can hope to obtain an indicator that reflects our intuitive concept of diversity in $\mathcal{X} \subset \mathbb{R}^n$ by setting $q = n$. We will call the indicator *average of inverse distances* and define $\mathrm{AID}(\mathcal{P}) := m_n(\mathcal{P})$.

Discrepancy (DISC)

In the area of quasi-Monte Carlo methods, a lot of theory has been developed regarding error bounds of estimated integrals, depending on the point sequences used for the numerical integration. To achieve low error bounds, point sets must possess a low *discrepancy*. Niederreiter [108, p. 13] states that "discrepancy can be viewed as a quantitative measure for the deviation from uniform distribution". Several types of discrepancy can be defined by changing the aggregation of individual deviations or by considering differently shaped subsets of the region of interest. Without loss of generality, we will assume that $\mathcal{X} = [0,1]^n$ in this section. The first theoretical results have been obtained using an L_∞ norm for aggregation and considering the family $\mathcal{J}^*$ of all subsets $J = [0, u_1) \times \ldots \times [0, u_n)$ of the unit hypercube [108, p. 14]. This way, the discrepancy

$$D_N^* = \sup_{J \in \mathcal{J}^*} \left| \frac{N_J}{N} - \mathrm{vol}(J) \right| \tag{2.3}$$

was defined, where $\mathrm{vol}(J) = \prod_{i=1}^n (u_i - \ell_i)$ is the volume of the respective subset and $N_J = |\{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{P} \wedge \boldsymbol{x} \in J\}|$ is the number of points of the set that fall into it. Using D_N^* and an appropriate definition for the variation $V(f)$ of f, the integration error can be bounded by the Koksma-Hlawka inequality [108, p. 20]

$$\left| \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i) - \int_{\mathcal{X}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right| \leq V(f) D_N^*(\mathcal{P}) \, . \tag{2.4}$$

This result suggests that it is generally advisable to minimize the discrepancy to obtain low integration errors, if f shall be treated as a black box. Unfortunately, several problems are associated with D_N^*. First of all, calculating L_∞ discrepancies is an NP-hard problem [36], which makes it infeasible in most practical situations. Secondly, Santner et al. [126, pp. 146–148] give an example where D_N^* favors points along the diagonal of the region of interest, and thus does not reflect the human intuition of unifomity. They further argue that discrepancy's relation to integration error is not necessarily relevant in the context of computer experiments [126, p. 144]. To avoid the run time problem, the L_2 norm of the deviations from uniformity is usually taken, leading to the discrepancy

$$T_N^* = \left(\int_{\mathcal{X}} \left(\frac{N_J}{N} - \mathrm{vol}(J) \right)^2 \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} \right)^{1/2} .$$

It should be noted that integration error can in principle also be bounded by L_2 discrepancy [103, 60, 150]. However, also T_N^* is disputed. Morokoff and Caflisch [103]

write: "While useful in theoretical discussions due to its relationship with D_N^*, T_N^* suffers as a means of comparing sequences and predicting performance because of the strong emphasis it puts on points near $\mathbf{0}$." In the example of Santner et al., this means that D_N^* regards the diagonal as better than the antidiagonal, which also seems dubious. Even worse, Matoušek [91] shows that T_N^* generally gives unreliable results for $N < 2^n$. As a workaround, the more general family $\mathcal{J}$ of subsets $J = [\ell_1, u_1) \times \ldots [\ell_n, u_n)$ can be considered, yielding

$$T_N = \left(\int_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}\times\mathcal{X},\, x_i < y_i} \left(\frac{N_J}{N} - \mathrm{vol}(J) \right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \right)^{1/2} .$$

This *unanchored* discrepancy formula at least gets rid of the origin's special role, but no information could be found on its effect on the problem addressed by Matoušek. The analogous formula for D_N is obtained by simply replacing $\mathcal{J}^*$ with $\mathcal{J}$ in (2.3).

While a lot of the literature deals with theoretical bounds on the discrepancy of quasirandom sequences, we are interested in computing the discrepancy of arbitrary point sets. Conveniently, Morokoff and Caflisch [103] derive the following explicit formula for T_N, which can be computed in $O(N^2 n)$:

$$\begin{aligned}(T_N)^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \prod_{k=1}^{n} \left(1 - \max\{x_{i,k}, x_{j,k}\}\right) \cdot \min\{x_{i,k}, x_{j,k}\} \\ - \frac{2^{1-n}}{N} \sum_{i=1}^{N} \prod_{k=1}^{n} x_{i,k}(1 - x_{i,k}) + 12^{-n}\end{aligned}$$

Their experiments, however, indicate that neither T_N^* nor T_N are monotone in species. Regarding the monotonicity in distance, neither a proof nor a counterexample could be found so far. A useful property of discrepancy is that it is possible to compute its expected value for a random uniform point set. For $(T_N)^2$ the formula is [103]

$$\mathrm{E}((T_N)^2) = \frac{6^{-n}(1 - 2^{-n})}{N} .$$

Overall, setting $\mathrm{DISC}(\mathcal{P}) = T_N$ seems to be a good choice. Hickernell [60] proposes several other variants of discrepancy that possess certain additional invariance properties. We will keep using T_N instead, because he does not give the expected values of these discrepancies.

Covering Radius (CR)

Definition 6 (Covering radius)**.** *If $(\mathcal{X}, d)$ is a bounded metric space and the point set $\mathcal{P}$ consists of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{X}$, then the* covering radius *of $\mathcal{P}$ in $\mathcal{X}$ is defined by*

$$\mathrm{CR}(\mathcal{P}) := d_N(\mathcal{P}, \mathcal{X}) = \sup_{\boldsymbol{x}\in\mathcal{X}} \left\{ \min_{1\le i\le N} \{d(\boldsymbol{x}, \boldsymbol{x}_i)\} \right\} = \sup_{\boldsymbol{x}\in\mathcal{X}} \{d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P})\} .$$

This definition is due to Niederreiter [108, p. 148], who coined the term *dispersion* for d_N, which "may be viewed as a measure for the deviation from denseness" [108, p. 149]. However, the name did not become widely accepted, because it does not reflect the intuitive meaning of d_N and is also used differently in other diversity-related research (see, e. g., [41, 86]). Meinl et al. [95] call this indicator the N-center measure. We will use the name *covering radius*, which is used for example by Damelin et al. [29], because d_N is the smallest radius for which closed balls around the points of $\mathcal{P}$ completely cover $\mathcal{X}$.

Definition 6 is actually also identical to that of the minimax distance design criterion as defined by Johnson et al. [66]. Based on this definition, Niederreiter [108, p. 149] proves an error bound on the estimate $\hat{f}^* = f(\hat{\boldsymbol{x}}^*)$ of the global minimum $f(\boldsymbol{x}^*)$. In this estimate, $\hat{\boldsymbol{x}}^* = \arg\min\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{P}\}$ denotes the point in the finite approximation set $\mathcal{P}$ for which the best objective value could be observed.

Theorem 1 (Niederreiter [108, p. 149])**.** *If $(\mathcal{X}, d)$ is a bounded metric space then, for any point set $\mathcal{P}$ of N points in $\mathcal{X}$ with covering radius $d_N = d_N(\mathcal{P}, \mathcal{X})$, we have*

$$\hat{f}^* - f(\boldsymbol{x}^*) \leq \omega(f, d_N) \,,$$

where

$$\omega(f, t) = \sup_{\substack{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X} \\ d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq t}} \{|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)|\}$$

is, for $t \geq 0$, the modulus of continuity *of f.*

This means that the prediction error for the global optimum is bounded by a function only depending on f and the covering radius of the set $\mathcal{P}$. The error bound actually holds not only for $\hat{f}^*$ as defined above, but also anywhere in $\mathcal{X}$ for a nearest neighbor estimator of f. Formally, it holds that

$$\forall \boldsymbol{x} \in \mathcal{X} : |f(\boldsymbol{x}) - f(\mathrm{nn}(\boldsymbol{x}, \mathcal{P}))| \leq \omega(f, d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P})) \leq \omega(f, d_N(\mathcal{P}, \mathcal{X})) \,. \tag{2.5}$$

This is a trivial observation following directly from the definitions of d_{nn}, ω, and d_N. Santner et al. [126, p. 149] explain this result intuitively in the context of metamodeling:

> "Suppose we plan to predict the response at an unobserved input site using our fitted stochastic model. Suppose we believe the absolute difference (absolute error) between this predicted value and the actual response generated by the computer code is proportional to the distance of the untried input site from the nearest input site at which we have observed the code. Then a minimax distance design would intuitively seem to minimize the maximum absolute error because it minimizes the maximum distance between observed design sites and untried sites. Unfortunately, minimax distance designs are difficult to generate and so are not widely used."

The difficulty in calculating and thus optimizing $d_N(\mathcal{P}, \mathcal{X})$ is due to the involvement of the uncountable $\mathcal{X}$. Although no explicit formula is known for arbitrary $\mathcal{X}$, Pronzato and Müller [118] give an algorithm for calculating $d_N(\mathcal{P}, [0,1]^n)$ regarding Euclidean distance with run time $O((nN)^{\lfloor n/2 \rfloor})$, based on Delaunay tessellation. The other resort would be using a Monte Carlo approach, because if $\mathcal{X}$ is finite with $|\mathcal{X}| = M$, calculation of the indicator becomes straightforward with run time $O(MNn)$.

Union of Balls (UB)

Ulrich et al. [145] propose an indicator related to CR. The basic idea is to consider balls of a predefined radius around the points of $\mathcal{P}$ and then to measure the union of these balls. Ulrich et al. call this indicator coverage diversity. The only tractable instantiation of this principle seems to be using L_∞ distance, in which case it is equivalent to Klee's measure problem [74]. For a diameter b, the formula is

$$\mathrm{UB}(\mathcal{P}) = \frac{1}{b^n} \int_{\mathcal{Z}} c^b_{\mathcal{P}}(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \,,$$

where

$$c^b_{\mathcal{P}}(\boldsymbol{z}) = \begin{cases} 1 & \text{if } \exists \boldsymbol{x} \in \mathcal{P} : \forall 1 \leq i \leq n : |z_i - x_i| \leq b/2 \,, \\ 0 & \text{else} \end{cases}$$

is the indicator function for membership in the union of balls. When integrating, one has to decide whether $\mathcal{Z} = \mathbb{R}^n$ or $\mathcal{Z} = \mathcal{X}$. In the former case, UB provides monotonicity in distance, but is more similar to a maximin criterion. The latter case gives an incentive to avoid the boundaries, because parts of the balls outside $\mathcal{X}$ would be wasted. UB can be calculated for uncountable $\mathcal{X}$ with a run time of $O(N^{n/2})$ as shown by Chan [24]. The parameter b, however, seems to be very critical. If it is too large, only the spread is relevant; if it is too small, the discriminative power is low.

2.1.2 Discussion of Diversity Indicators

Table 2.1 gives an overview of the diversity indicators' properties. The information about the properties was either taken from the original papers or obtained by finding counter examples. Question marks indicate the cases where no answer could be found. The given run time bounds mostly refer to the naïve implementations, because in the high dimensional spaces we are interested in, more sophisticated approaches usually yield little benefit. A similar survey has also been given by Ulrich et al. [144]. One could argue that one important application of the diversity indicators would be directly optimizing the indicator value for a fixed-size point set. In this scenario, monotonicity in species is not required. The twin property seems more interesting, since we can further differentiate the indicators that do not possess the property by defining two relaxed twin properties. First of all, it seems undesirable that indicator values can be improved by adding more duplicates. This is the case for SD and SDCM. Of the remaining indicators without TP1, many assign the worst possible

Table 2.1: Properties of diversity indicators.

	(Strict) monotonicity in species	Twin property TP1/TP2/TP3	(Strict) monotonicity in distance	Run time
SD	✔ / ✔	✘ / ✘ / ✘	✔ / ✔	$O(N^2n)$
AD	✘ / ✘	✘ / ? / ?	✔ / ✔	$O(N^2n)$
SDCM	? / ✔	✘ / ✘ / ✘	? / ✔	$O(Nn)$
MD	✘ / ✘	✘ / ✘ / ✔	✘ / ✔	$O(N^2n)$
SDNN	✘ / ✘	✘ / ✔ / ✔	✘ / ✔	$O(N^2n)$
PDNN	✘ / ✘	✘ / ✘ / ✔	✘ / ✔	$O(N^2n)$
WD	? / ✔	✔ / ✔ / ✔	✘ / ✘	$O(2^N)$
SPD	? / ✔	✔ / ✔ / ✔	? / ✔	$O(N^3)$
AID	✘ / ✘	✘ / ✘ / ✔	✔ / ✔	$O(N^2n)$
DISC	✘ / ✘	✘ / ✔ / ✔	? / ?	$O(N^2n)$
CR	✘ / ✔	✔ / ✔ / ✔	✘ / ✘	$O((nN)^{\lfloor n/2 \rfloor})$
UB	✘ / ✔	✔ / ✔ / ✔	✘ / (✔)	$O(N^{n/2})$

indicator value to point sets with duplicates. This is disadvantageous, too, because we want to at least be able to discriminate between different remaining parts of the sets [21].

Axiom 4 (Relaxed twin properties)**.** *Without loss of generality, let D be a diversity indicator that is to be maximized. For a set of species $\mathcal{P}$ with $|\mathcal{P}| \geq 2$, and species $t \notin \mathcal{P}$ with $d(t, s) = 0$ for some $s \in \mathcal{P}$, we make the following two requirements:*

- *TP2: The diversity of $\mathcal{P} \cup t$ should be greater than the worst possible value and not exceed the diversity of $\mathcal{P}$: $D_{\min} < D(\mathcal{P} \cup t) \leq D(\mathcal{P})$*
- *TP3: The diversity should at least not improve by adding a redundant species: $D(\mathcal{P} \cup t) \leq D(\mathcal{P})$*

Obviously, TP1 ⇒ TP2 ⇒ TP3. While TP1 could always be enforced by removing all the duplicates from $\mathcal{P}$ in a preprocessing step, one can easily see that this would break the monotonicity in distance, at least for some indicators.

AID is one of the indicators without TP2. At least this deficit could be removed easily, as indicated by Damelin et al. [29]. They show that an energy definition closely related to (2.2) is equivalent to the L_2 star discrepancy. The required modification of (2.2) is actually an introduction of a "nugget factor", which prevents the singularities when distances of zero occur. The resulting energy definition satisfies TP2 and also the integration error can be bounded by it. Although this modification obviously also applies to AID, we do not follow this path because further investigations would be required to set the additional parameter.

Izsák and Papp [64] observed that an appropriate averaging can eliminate monotonicity in species for SD. The same also holds for SDCM. Simultaneously, this averaging also seems to influence the twin properties in the following way:

Conjecture 1. *The penalty imposed on the indicator value by averaging over all distances seems to be sufficient to enable TP2 for the two indicators* average distance *and* average distance to the center of mass*.*

Note that monotonicity in distance is obviously not affected by averaging, as the number of species is constant in the definition of this axiom. However, although better twin properties would be desirable, TP3 and monotonicity in distance seem to be often sufficient to sensibly optimize the diversity of a fixed-size point set [57, 40, 95].

To conclude the discussion of diversity, we will try to summarize *what* the diversity indicators actually measure. After all, there seem to be only three or four main diversity concepts, depending on the level of detail. The first one is *spread*, which is measured by SD and SDCM. Although this approach seems to be used often, it is not recommended, because it does not reflect the intuitive understanding of diversity. As a reader, one often wonders if the respective authors used it consciously, or if they chose the simplest/first approach they could think of. The other category is *uniformity*, which is subdivided into uniformity with and without consideration of $\mathcal{X}$. The former one could be described as *coverage* or *representativeness*, and can be measured by CR and UB. Also WD seems to have some relations to this concept, based on its properties. DISC is a bit difficult to categorize, because it does not employ a distance function and is not easily applicable to arbitrary $\mathcal{X}$. Additionally, there are several different instantiations of the discrepancy concept, with slightly different properties [103, 60]. Experiments in Chapter 4 seem to confirm the theoretical results of Matoušek [91], in the sense that DISC seems to represent true uniformity for $N \gg 2^n$ and some kind of *maximin diversity* otherwise. The latter term characterizes indicators whose optimal point sets exhibit a higher point density at the boundary of $\mathcal{X}$ than random uniform sampling, if no countermeasures are taken. This property is shared by all indicators that asymptotically reward a maximization of minimal distance(s) between points in $\mathcal{P}$, hence the name. This approach clearly disregards $\mathcal{X}$ and ultimately leads to very *regular* point sets, which is shown by its close connection to sphere packing problems [4].

2.1.3 Other Quality Indicators

Diversity indicators are so important to us, because they work independently of the actual optimization problem. However, as we are dealing with optimization, also the objective function has to be considered somehow. In the extreme case of global optimization, only a single solution is considered, and thus only the best objective value is relevant. The best objective value in turn can be interpreted as belonging to a class of quality indicators which encompasses all kinds of statistics of the objective value distribution. These indicators obviously are problem-independent, too.

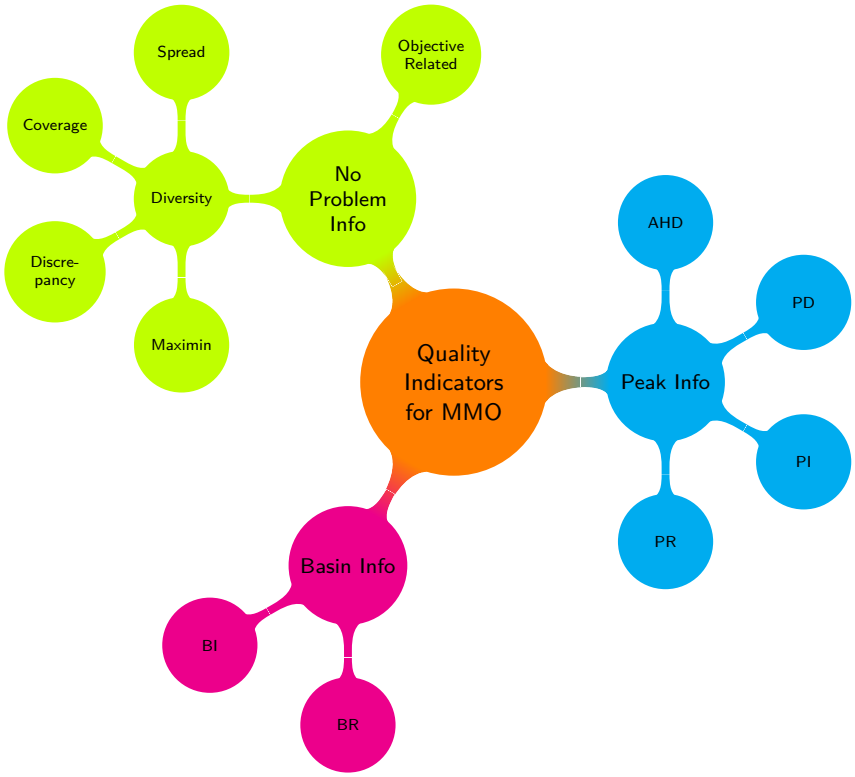


Figure 2.1: One possible classification of quality indicators for multimodal optimization.

However, we are still missing problem-independent indicators that are really useful for multimodal optimization by combining diversity and objective values. To date, such indicators always employ information regarding the location and optionally the attraction basins of the problems' optima. This is reflected in Figure 2.1, which classifies indicators according to the amount of additional information that is necessary for the assessment. Some important instances of problem-dependent indicators are already displayed there. These will be explained in the following.

Some indicators in this section require a given set of locally optimal positions $\mathcal{O} = \{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\nu^*\}$, $\nu < \infty$, to assess $\mathcal{P}$. This means they can only be employed in a benchmarking scenario on test problems that were specifically designed so that $\mathcal{O}$ is known. Note, however, that $\mathcal{O}$ does not necessarily have to contain all existing optima, but can also represent a subset (e. g., only the global ones). Even more challenging to implement are indicators that require information about which basin each point of the search space belongs to. This information can either be provided by a careful construction of the test problem, or by running a descent algorithm for each $\boldsymbol{x} \in \mathcal{P}$ as a starting point during the assessment and then matching the obtained local optima with the points of the known $\mathcal{O}$. Regardless of how it is achieved, we

will assume the existence of a function

$$b(\boldsymbol{x}, \boldsymbol{x}^*) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \text{basin}(\boldsymbol{x}^*), \\ 0 & \text{else.} \end{cases}$$

The rationale of indicators for covered basins instead of distances to local optima is that the former also enable measuring in early phases of an optimization, when the local optima have not been approximated well yet. If the basin shapes are not very regular, the latter indicator type may be misleading in this phase.

Statistics of the Distribution of Objective Values

Regarding the assessment of the population's raw performance, few true alternatives seem to exist. The only things that come to mind are conventional statistics of the objective value distribution, with the mean or median as the most obvious measures. Values from the tail of the distribution, as the best or worst objective value, do not seem robust enough to outliers. Thus, as an example for this group of indicators, the average objective value (AOV) shall be explicitly named:

$$\text{AOV}(\mathcal{P}) := \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i)\,.$$

Peak Ratio (PR)

Ursem [147] introduced the average number of found peaks as a performance measure for MMO. Following this concept, we define the number of found optima

$$o = |\{\boldsymbol{x}^* \in \mathcal{O} \mid d_{\text{nn}}(\boldsymbol{x}^*, \mathcal{P}) \leq r\}| \tag{2.6}$$

divided by the total number of optima as peak ratio $\text{PR}(\mathcal{P}) := o/\nu$. This formulation was probably first expressly mentioned by Thomsen [139]. The indicator requires some constant r to be defined by the user, to decide if an optimum has been approximated appropriately. This approach corresponds to the assumption of binary relevance in the area of information retrieval [90, p. 152] and the peak ratio is equivalent to a measure called sensitivity, recall, or true positive rate there [90, p. 155]. Note that the term peak ratio is also used in the CEC 2013 competition [80], but only an approximation, which does not employ the actual positions of the optima, is calculated. Also the maximum peak ratio by Miller and Shaw [99] is not directly related to our measure.

Peak Distance (PD)

This indicator simply calculates the average distance of a member of the reference set $\mathcal{O}$ to the nearest individual in $\mathcal{P}$:

$$\text{PD}(\mathcal{P}) := \frac{1}{\nu} \sum_{i=1}^{\nu} d_{\text{nn}}(\boldsymbol{x}_i^*, \mathcal{P})\,. \tag{2.7}$$

A first version of this indicator (without the averaging) was presented by Stoean et al. [134] as "distance accuracy". With the $1/\nu$ part, peak distance is analogous to the indicator inverted generational distance [25], which is computed in the objective space of multiobjective problems.

Proposition 1. *If $(\mathcal{X}, d)$ is a bounded metric space, then the peak distance of a set $\mathcal{P} \subset \mathcal{X}$, $|\mathcal{P}| = N$, can be bounded as follows:*

$$\mathrm{PD}(\mathcal{P}) \leq d_N(\mathcal{P}, \mathcal{X})\,. \tag{2.8}$$

Proof. Definition 6 tells us that $\forall \boldsymbol{x}_i^* \in \mathcal{O} : d_{\mathrm{nn}}(\boldsymbol{x}_i^*, \mathcal{P}) \leq d_N(\mathcal{P}, \mathcal{X})$. Thus, the result follows directly from inserting d_N into (2.7):

$$\mathrm{PD}(\mathcal{P}) = \frac{1}{\nu}\sum_{i=1}^{\nu} d_{\mathrm{nn}}(\boldsymbol{x}_i^*, \mathcal{P}) \leq \frac{1}{\nu}\sum_{i=1}^{\nu} d_N(\mathcal{P}, \mathcal{X}) = d_N(\mathcal{P}, \mathcal{X})\,. \qquad \square$$

Peak Inaccuracy (PI)

Thomsen [139] proposed the basic variant of the indicator

$$\mathrm{PI}(\mathcal{P}) := \frac{1}{\nu}\sum_{i=1}^{\nu} |f(\boldsymbol{x}_i^*) - f(\mathrm{nn}(\boldsymbol{x}_i^*, \mathcal{P}))| \tag{2.9}$$

under the name "peak accuracy". To be consistent with PR and PD, we also add the $1/\nu$ term here. We allow ourselves to relabel it to peak inaccuracy, because speaking of accuracy is a bit misleading as the indicator must be minimized. PI has the disadvantage that $\mathcal{P}$ does not necessarily have to cover $\mathcal{O}$ well, because it is possible for one solution to satisfy several optima at once. On the other hand, comparing the indicator value to a baseline performance, e. g., calculated as the peak inaccuracy for the global optimum alone, might relativize seemingly good performances.

Proposition 2. *If $(\mathcal{X}, d)$ is a bounded metric space, then the peak inaccuracy of a set $\mathcal{P} \subset \mathcal{X}$, $|\mathcal{P}| = N$, can be bounded as follows:*

$$\mathrm{PI}(\mathcal{P}) \leq \omega(f, d_N(\mathcal{P}, \mathcal{X}))\,.$$

Proof. This time the result follows from inequality (2.5), which allows us to insert $\omega(f, d_N(\mathcal{P}, \mathcal{X}))$ into (2.9). $\square$

Averaged Hausdorff Distance (AHD)

This indicator can be seen as an extension of peak distance due to its relation to the inverted generational distance. It was defined by Schütze et al. [129] as

$$\begin{aligned}\mathrm{AHD}(\mathcal{P}) &:= \Delta_p(\mathcal{P}, \mathcal{O}) \\ &= \max\left\{\left(\frac{1}{\nu}\sum_{i=1}^{\nu} d_{\mathrm{nn}}(\boldsymbol{x}_i^*, \mathcal{P})^p\right)^{1/p}, \left(\frac{1}{N}\sum_{i=1}^{N} d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{O})^p\right)^{1/p}\right\}.\end{aligned}$$

The definition contains a parameter p that controls the influence of outliers on the indicator value (the more influence the higher p is). For $1 \leq p < \infty$, AHD has the property of being a semi-metric and for $p = \infty$ it coincides with the conventional Hausdorff distance d_{H}, which is a metric [129]. Note that the conventional Hausdorff distance relies heavily on the covering radius, i. e., for finite sets $\mathcal{P}$ and $\mathcal{O}$ it can be written as $d_{\mathrm{H}} = \max\{d_N(\mathcal{P}, \mathcal{O}), d_\nu(\mathcal{O}, \mathcal{P})\}$. We constantly choose $p = 1$, analogously to Emmerich et al. [40]. The practical effect of the indicator is that it rewards the approximation of the optima (as PD does), but as well penalizes any unnecessary points in remote locations. This makes it an adequate indicator for the comparison of approximation sets of different sizes.

Proposition 3. *If $(\mathcal{X}, d)$ is a bounded metric space, then the averaged Hausdorff distance of a set $\mathcal{P} \subset \mathcal{X}$, $|\mathcal{P}| = N$, can be bounded as follows:*

$$\mathrm{AHD}(\mathcal{P}) \leq \max\{d_N(\mathcal{P}, \mathcal{X}), d_\nu(\mathcal{O}, \mathcal{X})\} \ .$$

Proof. Again, the result follows from inserting the covering radii of $\mathcal{P}$ and $\mathcal{O}$, respectively, and simplifying the expression:

$$\begin{aligned}
\mathrm{AHD}(\mathcal{P}) &\leq \max\left\{\left(\frac{1}{\nu}\nu d_N(\mathcal{P}, \mathcal{X})^p\right)^{1/p}, \left(\frac{1}{N}N d_\nu(\mathcal{O}, \mathcal{X})^p\right)^{1/p}\right\} \\
&= \max\left\{(d_N(\mathcal{P}, \mathcal{X})^p)^{1/p}, (d_\nu(\mathcal{O}, \mathcal{X})^p)^{1/p}\right\} \\
&= \max\{d_N(\mathcal{P}, \mathcal{X}), d_\nu(\mathcal{O}, \mathcal{X})\} \ .
\end{aligned}$$

□

Note that Proposition 3 provides further evidence that there is a fundamental difference between AHD on the one hand, and PD and PI on the other hand. The bound indicates that minimizing $d_N(\mathcal{P}, \mathcal{X})$ is not sufficient to optimize AHD, due to its dependency on the set of optima $\mathcal{O}$.

Basin Ratio (BR)

The number of covered basins is calculated as

$$o = \sum_{i=1}^{\nu} \min\left\{1, \sum_{j=1}^{N} b(\boldsymbol{x}_j, \boldsymbol{x}_i^*)\right\} .$$

The basin ratio is then $\mathrm{BR}(\mathcal{P}) := o/\nu$, analogous to PR. This indicator can only assume $\nu + 1$ distinct values. If the basin sizes do not vary too much, it should be quite easy to obtain a perfect score in low dimensions by a moderately sized simple random uniform sampling (SRS) of the search space. The indicator makes sense especially when not all of the existing optima are relevant. Then, its use can be justified by the common assumption in global optimization that the actual optima can be found relatively easily with a hill climber, once there is a start point in each respective basin [141].

Table 2.2: Overview of some MMO indicators.

Indicator	Best	Worst	Regards $f(\boldsymbol{x})$	Use with variable N	Without optima	Without basins	Without params
AOV	$f(\boldsymbol{x}^*)$	$f_{\max}$	✔	✔	✔	✔	✔
PR	1	0	✘	✘	✘	✔	✘
PD	0	$d_{\max}$	✘	✘	✘	✔	✔
PI	0	$f_{\max}$	✔	✘	✘	✔	✔
AHD	0	$d_{\max}$	✘	✔	✘	✔	✘
BR	1	0	✘	✘	✘	✘	✔
BI	0	$f_{\max}$	✔	✘	✘	✘	✔

Basin Inaccuracy (BI)

This combination of basin ratio and peak inaccuracy was proposed by Preuss and Wessing [117]. It is defined as

$$\mathrm{BI}(\mathcal{P}) := \frac{1}{\nu}\sum_{i=1}^{\nu}\begin{cases}\min\left\{|f(\boldsymbol{x}_i^*) - f(\boldsymbol{x})| \mid \boldsymbol{x} \in \mathcal{P} \wedge b(\boldsymbol{x}, \boldsymbol{x}_i^*)\right\} & \exists \boldsymbol{x} \in \mathrm{basin}(\boldsymbol{x}_i^*)\,, \\ f_{\max} & \text{else,}\end{cases}$$

where $f_{\max}$ denotes a penalty value, e. g., the difference between the global optimum and the worst possible objective value. For each optimum, the indicator calculates the minimal difference in objective values between the optimum and all solutions that are located in it's basin. If no solution is present in the basin, a penalty value is assumed for it. Finally, all the values are averaged. The rationale behind this indicator is to enforce a good basin coverage, while simultaneously measuring the deviation of objective values.

2.1.4 Discussion of Other Quality Indicators

The ideal indicator for multimodal optimization would probably regard both diversity and objective values, enable fair comparisons of sets with different sizes, and require no problem information or additional parameters. Table 2.2 shows a classification of the indicators in this section regarding these properties. With the notable exception of AOV and AHD, all of them are monotone in species, meaning that for two sets $\mathcal{P}, \mathcal{Q}$ with $\mathcal{P} \subset \mathcal{Q}$, the superset $\mathcal{Q}$ will never be regarded as the worse one. In some situations, this is inappropriate, as we will see in the next section. On the other hand, this behavior is not provided by all diversity indicators, although it would be desirable in their case.

Unfortunately, only diversity indicators and AOV can be applied in real-world applications, as they are the only ones not needing any problem knowledge. The others are only available in benchmarking scenarios. AHD is more challenging than PD, because the former not only rewards the approximation of $\mathcal{O}$, but also penalizes superfluous points in remote locations. AHD's parameter should be well-tempered.

While PR has a straightforward interpretation, it can be easily misconfigured [157], by making the approximation task to easy or too difficult. Therefore, BR is a good parameterless alternative, although with a slightly different focus. It may be especially useful if only a subset of all optima is requested to be found, as in [80]. BR and BI also put a higher emphasis on diversity than the "peak-oriented" indicators, but are more difficult to implement, because they rely on more problem knowledge. PI can be easily deceived when an optimum is not covered by any solution, but another similarly good solution exists in another basin nearby [117].

There should be some room for further development of indicators. In [117] for example, it was suggested to incorporate objective values by simply augmenting the search points before evaluating with PD or AHD. Instead of $\mathcal{P}$, then the points $\mathcal{P}' = \{(\boldsymbol{x}_1^\top, f(\boldsymbol{x}_1))^\top, \ldots, (\boldsymbol{x}_N^\top, f(\boldsymbol{x}_N))^\top\}$ would be assessed. Likewise, it would be straightforward to define an "averaged Hausdorff inaccuracy" by replacing the distances in AHD by deviations in objective space, as in PI. (Also a "basin distance" is imaginable, by using search space distances in an indicator analogous to BI.) Another suggestion that was made in [117] is to incorporate quantity-adjustment into PR and BR by introducing a penalization factor. A better way seems to be the approach usually taken in information retrieval, where besides recall $= o/\nu$ the conflicting measure precision $= o/|\mathcal{P}|$ is recorded. (Note that recall is conceptually equivalent to PR and BR.) These two can be aggregated using the measure [90, p. 156]

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \,. \tag{2.10}$$

However, none of these ideas has received much experimental analysis, yet.

2.2 Performance Measurement

Unfortunately, comparing optimization algorithms is an inherently multiobjective problem [141]. One objective is the effort spent for optimization, which is to be minimized, and the other objective is some measure of solution quality. As already indicated in Section 1.1, the conflict is often avoided by fixing either the target quality or the budget. Both approaches have advantages and disadvantages. A fixed target enables easy speed comparisons(e. g., "algorithm A is on average twice as fast as algorithm B") [54]. On the downside, it is prone to *floor effects*, which means that the posed task may be too difficult to measure any progress. For example, in comparisons based on the BBOB procedure, expected running times are often reported as ∞, because the algorithms never reach the target [112]. The fixed-budget scenario has the advantage that we are always comparing algorithms for exactly the budgets that we deem as realistic for the application (see Table 1.1). It does not need much calibration to avoid floor effects and it is easy to measure several different characteristics in one run. However, the measured differences between algorithms are problem-dependent and thus not intuitively accessible [54]. In any case, only less powerful statistical tests may be available, because requirements of tests may be not

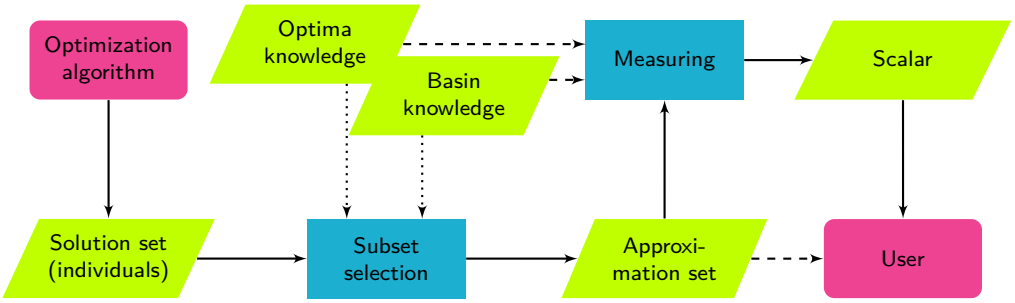


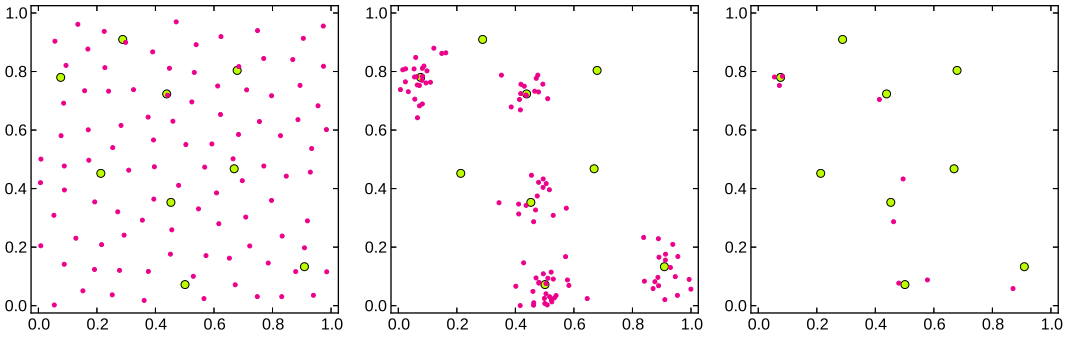
Figure 2.2: Data flow from the solution set obtained by an optimization algorithm up to the final scalar measure value. Problem knowledge can be an optional input for the measuring and (rather unlikely) to the subset selection part.

fulfilled. For example, distributions of indicator values or run times aggregated over different problems may be multimodal and heteroscedastic.

The direct multiobjective assessment should be seen as the "holy grail" of algorithm comparison, as it provides both views at once. However, it requires more sophisticated analysis software, higher skilled experimenters and audience, more memory to store the data, and is less convenient for advertising algorithms with simple messages. So, the multiobjective assessment is still not in use to date. Instead, we choose the fixed-budget scenario throughout this work, for the reasons stated above.

Regarding multimodal optimization, performance assessment in real-world scenarios is difficult, because necessary information on the number and position of the problems' local optima is of course lacking under the black-box model. When benchmarking on artificially constructed test instances, this information is naturally available. Nevertheless, meaningful performance measurement in benchmarking is still challenging. Here, the difficulty lies in assessing algorithms in a way that is similar to how they will be used in practice. As a general guideline, those outcomes which leave the least effort to the human decision maker should be evaluated best. Or, stated by Eiben and Jelasity [38] in the context of evolutionary computation, "the choice of how we evaluate and compare EAs should be a consequence of what we want the EAs to do".

To clarify the situation, regard the sketch of a performance assessment workflow in Figure 2.2, as it was introduced in [117]. This figure emphasizes that the measurement must be carried out on an *approximation* set, which is a subset of the "raw" set of all solutions generated by the optimization algorithm. The corresponding subset selection may be viewed as a part of the assessment or the optimization algorithm (line 8 in Algorithm 1), but the important characteristic is that it must be automated. If this step is missing (i. e., the approximation set size is effectively unlimited), the optimization algorithm is released from any requirement to identify the actual solutions that approximate the optima well. But as we assumed budgets of several thousand objective function evaluations, the solution set is much too large



(a) Maximal exploration (b) Multistart local search (c) Approximation set

Figure 2.3: Hypothetical results of optimization algorithms that require different ways of performance assessment. Lime green dots mean local optima, magenta dots the solution/approximation set.

for human inspection. In Section 2.1.3 we already mentioned that a similar problem exists in information retrieval, where conflicting measures as precision and recall have to be balanced.

Figure 2.3 shows three imaginable results of MMO algorithms. (Such outcomes are entirely possible in practice, as was demonstrated in [158].) Figure 2.3a is the most difficult scenario, where a large number of candidates without any noticeable structure is provided. In this case, it is very difficult to identify the interesting solutions by post-processing. Thus, a custom-built subset selection method for MMO is our only hope for a sensible assessment (see, e. g., [157, 117, 114]). In case of Figure 2.3b, the structure of the data corresponds to the local optima. Here, an ordinary clustering approach would be probably sufficient to identify the clusters [143, pp. 95–116]. Then, extracting a representative of each cluster could be easily done by applying an appropriate criterion (e. g., objective value). Finally, Figure 2.3c represents the simplest case, where the subset selection has already been carried out and the approximation set can be directly passed to any quality indicator and/or decision maker.

From these observations it follows that if a complete MMO algorithm is to be evaluated, its aim should be to *deliver the optima, and only the optima.* (Note that this requirement is more or less equivalent to asking the algorithms to carry out a subset selection by themselves.) This behavior can be encouraged by different means. Firstly, indicators that penalize large approximation set sizes could be used [117]. AHD does not do this, but seems to be useful in scenarios where the approximation set size is unbounded as well, because it penalizes solutions far away from optima, as in Figure 2.3a. As sets that are good according to AHD must have a structure as in Figure 2.3b, this approach should be admissible, because they are easy to post-process. Another option would be F_1 (see Equation 2.10), the widely used performance measure in information retrieval [90, p. 156], or analyzing its two individual parts in a multiobjective fashion. Otherwise, a hard limit can be imposed

on the approximation set size. This limit could be, e. g., the actual number of optima ν, but the approach is only reasonable if this number is quite low, e. g., in the double digits. Then, we have a situation as in Figure 2.3c, and also all other quality indicators can be applied.

However, these considerations only hold if a complete MMO algorithm is assessed. For comparing global stages, it is of course admissible to simply measure their exploratory capability, disregarding the above recommendations.

2.3 Test Problems

Theoretical results on the performance of optimization algorithms often only hold under certain conditions, which makes them dissatisfying for many practical situations. For example, Auger and Teytaud [10] show that for continuous optimization problems, a generally optimal optimization algorithm does exist (there is a "free lunch"). However, this optimal algorithm is not computationally feasible for reasonable budgets. So, we may still try to find the best feasible optimization algorithm for more restricted classes of problems. This is usually done by experimental comparisons.

Besides the actual algorithm implementation, appropriate test problems are required for carrying out this benchmarking. The problems should be quick to evaluate and pose "realistic" challenges for the optimization algorithms. However, following this guideline is hindered by the fact that it is usually unknown what "important" or "typical" real-world problems are, and how close "artificial" problems are related to them. Obtaining the mathematical definition of a real-world problem is often impossible because it is a trade secret or contains the invocation of (proprietary) third-party software. If not for legal reasons, dealing with such software is often a hassle, because of non-portability or high run time. Therefore real-world problems themselves do not seem to be a good choice for benchmarking [143, p. 175], maybe with the exception of the CEC 2011 [31] collection of problems with low run time requirements.

One workaround would be to try to identify features that characterize real-world problems properly. If the features generalize sufficiently, it would be possible to predict the performance of optimization algorithms and thus find artificial test problems that have similar difficulty for a given algorithm. The main disadvantage of this relatively new approach called *exploratory landscape analysis* (ELA) [97, 96] is that the computation of the features requires (potentially expensive) function evaluations and that the predictor may be inaccurate because of weak interpretability of the features. Similar criticism holds for test problems that are created by meta-modeling a test sample of a real-world problem, as we can never be sure that the meta-model captures the original problem's features appropriately. Instead, we want to follow an approach proposed by Eiben and Jelasity [38]. The idea is to use a generator that can produce test problem instances randomly from a given distribution. This way, we can directly control important problem features as the number of optima and

thus the hardness of the instance.

Before turning to the actual problem generator used in our experiments, let us review the literature for artificial test problems. We make a subdivision of the historic development of such problems into four phases:

1. The simplest (and probably oldest) test problems are carefully designed formulas which offer no options for configuration at all. Advantages of these problems are the usually known locations of all local optima and often non-separability. A disadvantage are the fixed and usually low numbers of dimensions ($n \leq 10$) and optima ($\nu \leq 10$). Törn et al. [141] give an overview of such problems, which Rönkkönen et al. [122] call the common family.

2. The second stage contains problems that are defined for arbitrary dimensions. However, due to their simplicity, these problems are often separable and have their global optimum at $\mathbf{0}$. The collections CEC 2005 [135] and BBOB [44] contain many of these problems, although the problem definitions have been extended by random translations or rotations of the landscapes to eliminate the two mentioned drawbacks. However, the regular structure of, e. g., Rastrigin's or Schwefel's problem persists. Furthermore, there is a "difficulty gap", because these problems are usually either unimodal or contain a number of optima that is exponential in n. Rönkkönen et al. [122] call such constructions the cosine family and present an approach to reduce their regularity by nonlinear transformations of the search space.

3. In the third category we find problems that are randomized or configurable in some additional properties that are decisive for their difficulty. Apparently ahead of their time, Fletcher and Powell [45] published already in 1963 a multimodal and non-separable test problem that was inspired by a real-world application. By random initialization, different instances with 2^n local optima can be generated. Lunacek et al. [87] define a problem with two funnels, i. e., a global structure consisting of two large attraction basins. The depth and size of the worse funnel can be manipulated to adjust the difficulty of the problem. Addis and Locatelli [3] present a parameterized, multimodal test problem that consists of several superposed waveforms. Unfortunately, the number of decision variables can only be influenced indirectly.

4. The possibly most advanced problems are in the fourth category. Gallagher and Yuan [47] developed a test problem generator that combines randomly drawn Gaussians by taking the maximum of them. A great variety of landscapes can be created by randomizing several parameters as position, height, and the number of Gaussians. Very similar, but based on polynomials, are the generators of Preuss and Lasarczyk [115], Rönkkönen et al. [122] ("quadratic family"), and Gaviano et al. [48]. In contrast to the others, the approach by Gaviano et al. even guarantees continuous differentiability everywhere, but also creates one large funnel structure.

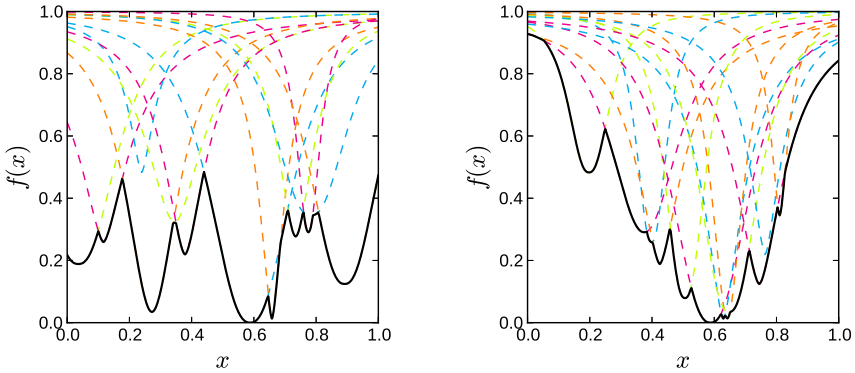


Figure 2.4: One-dimensional MPM2 functions (solid black) with their individual peaks (dashed). These example functions have ten optima each.

2.3.1 Multiple Peaks Model 2

For our experiments, we need to select some reasonable test problems, preferably from the fourth category. Those in the CEC 2013 benchmark suite [80] for multimodal optimization do not supply enough information to (efficiently) apply all indicators. Only the positions for a few global optima are known and the corresponding attraction basins would have to be estimated by steepest descent algorithms. Furthermore, many instances are only trivial one-dimensional problems. Eiben and Jelasity [38] suggest to use parametrized problem *generators* that can produce test instances randomly, to exercise fine control over problem difficulty. Such generators were also explicitly proposed for multimodal optimization by Rönkkönen et al. [122]. Thus, our choice is a hybrid version of the test problem generators by Preuss and Lasarczyk [115] and Gallagher and Yuan [47], which is very similar to the "quadratic family" in [122]. It shall be named multiple peaks model 2 (MPM2), in recognition of the similarity to extensions to the generator in [115] made in [114] under the name multiple peaks model. The generator produces multimodal problem instances by combining several randomly distributed peaks. Hence, the problems are irregular and non-separable, which are also important features of difficult real-world problems. The problem is defined by the following formulas:

$$f(\boldsymbol{x}) = 1 - \max\{g(\boldsymbol{x}, \boldsymbol{p}) \mid \boldsymbol{p} \in P\} \tag{2.11}$$

$$g(\boldsymbol{x}, \boldsymbol{p}) = \frac{h_{\boldsymbol{p}}}{1 + \frac{\mathrm{md}(\boldsymbol{x}, \boldsymbol{p})^{s_{\boldsymbol{p}}}}{r_{\boldsymbol{p}}}} \tag{2.12}$$

$$\mathrm{md}(\boldsymbol{x}, \boldsymbol{p}) = \sqrt{(\boldsymbol{x} - \boldsymbol{p})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} (\boldsymbol{x} - \boldsymbol{p})} \tag{2.13}$$

The objective function is given in (2.11). It takes the minimum of $N_{\text{peaks}} = |P|$ unimodal functions (2.12) around peak positions $\boldsymbol{p} \in P$. This has the advantage that local optima with known positions are created, which is in turn necessary to calculate some quality indicators (see Section 2.1.3). The principle is illustrated in

Figure 2.4. Rönkkönen et al. [122] criticize the exponential decay of the Gaussians used in [47]. Therefore, we are using the polynomial form in (2.12). Each of these functions is associated with parameters $h_{\boldsymbol{p}}$, $s_{\boldsymbol{p}}$, and $r_{\boldsymbol{p}}$ for height, shape, and radius, respectively. The idea of random shape and radius parameters is taken from [115]. By slightly deviating from locally quadratic behavior ($s_{\boldsymbol{p}} = 2$), we intend to increase the difficulty for local search algorithms. Radii $r_{\boldsymbol{p}}$ influence the size of attraction basins, and thus the probability to place a starting point in the basin. Optima with small attraction basins will be difficult to find [141, 122]. Additionally, a randomly drawn covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}}$ belongs to each peak. This matrix is used to create the optima's basins as rotated hyperellipsoids, by calculating the Mahalanobis distance in (2.13). All mentioned parameters are drawn randomly during initialization and then stored. By convention, we will always set $\max\{h_{\boldsymbol{p}} \mid \boldsymbol{p} \in P\} = 1$ and $\mathcal{X} = [0, 1]^n$. The former has the advantage that objective function values are always in $[0, 1]$, which makes it easy to convert from maximization to minimization and to calculate the basin inaccuracy. The latter provides some protection from numerical problems, which would appear when calculating small differences of large numbers. The box constraints shall be strictly obeyed anytime (scenario S1 in [81]). The matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}}$ is generated in the same way as in [47], by creating a random rotation matrix $\mathbf{R}$ first, according to the method of Rudolph [121]. Then a vector $\boldsymbol{v}$ of random values $v_i \sim U(0.0025, 0.0525)$ is used to create $\boldsymbol{\Sigma}_{\boldsymbol{p}} = \mathbf{R}^\top(\boldsymbol{v}\mathbf{I})\mathbf{R}$.

Note that $\nu \leq N_{\text{peaks}}$, because peaks can be masked by others. Both original landscape generators [115, 47] suffer from this problem. We therefore employ a sophisticated initialization procedure, to obtain problems with a given number of optima. This heuristic is shown in Algorithm 2 (with a slightly overloaded notation, as $\boldsymbol{p}$ rather denotes the peak object including parameters than only the position). First, it generates a problem instance with ν peaks and iteratively reduces the radii until at least 80% of all peaks are local optima. Then, the still missing optima are created by adding random peaks by rejection sampling. This means that a randomly drawn peak is only accepted if it increases the number of optima by one. Unfortunately, determining the number of optima ν takes $O(N_{\text{peaks}}^2)$ time, because we have to test for each $\boldsymbol{p} \in P$ if $f(\boldsymbol{p}) = 1 - h_{\boldsymbol{p}}$. If the condition is fulfilled, an optimum is located at $\boldsymbol{p}$. In practice, this makes the use of test problems of this kind infeasible for more than a few hundred optima, because the complete initialization procedure has cubic worst-case time complexity.

The algorithm is also capable of generating two different global structures. The first structure represents a more or less stationary case, where there is no trend in the depths of local optima and locations of optima are distributed uniformly over the search space. This structure is called the random topology. The other one contains one large funnel, that is, optima are clustered around the global one and the depths are negatively correlated with distance from the global optimum. These two topologies are chosen, because it is expected that they represent two extremes on the difficulty scale of multimodal problems – at least for global optimization [143, p. 11]. This belief is reiterated in [141], where the authors speak of isolated and embedded global optima, and in [86].

Algorithm 2 Initialization of MPM2

Input: number of optima ν, number of variables n, topology
Output: test problem instance

$h_{\min} \leftarrow 0.5$; $h_{\max} \leftarrow 0.99$
$s_{\min} \leftarrow 1.5$; $s_{\max} \leftarrow 2.5$
$r_{\min} \leftarrow 0.25\sqrt{n}$; $r_{\max} \leftarrow 0.5\sqrt{n}$
$\boldsymbol{p}_1 \leftarrow$ randomUniformPeak(1, $U(s_{\min}, s_{\max})$, $U(r_{\min}, r_{\max})$) // global optimum
$P \leftarrow \{\boldsymbol{p}_1\}$
if topology is random **then**
  $P \leftarrow P \cup \{\nu - 1$ additional random uniform peaks$\}$
else if topology is funnel **then**
  $P \leftarrow P \cup \{\nu - 1$ additional randomly clustered peaks around $\boldsymbol{p}_1\}$
end if
$f \leftarrow$ createInstance(P, topology)
while |localOptima(f)| $< 0.8 \cdot |P|$ **do**
  for all $\boldsymbol{p} \in P$ **do**
    $r_{\boldsymbol{p}} \leftarrow 0.95 \cdot r_{\boldsymbol{p}}$ // reduce radii of all peaks
  end for
end while
while |localOptima(f)| $< \nu$ **do**
  $\nu_{\text{prev}} \leftarrow$ |localOptima(f)|
  while $\nu_{\text{prev}} + 1 \neq$ |localOptima(f)| **do** // do rejection sampling
    if topology is random **then**
      $\boldsymbol{p} \leftarrow$ randomUniformPeak($U(h_{\min}, h_{\max})$, $U(s_{\min}, s_{\max})$, $U(r_{\min}, r_{\max})$)
    else if topology is funnel **then**
      $\boldsymbol{p} \leftarrow$ clusteredPeak($U(h_{\min}, h_{\max})$, $U(s_{\min}, s_{\max})$, $U(r_{\min}, r_{\max})$, $\boldsymbol{p}_1$)
    end if
    $\nu_{\text{prev}} \leftarrow$ |localOptima(f)|
    $f \leftarrow$ createInstance($P \cup \{\boldsymbol{p}\}$, topology)
  end while
  $P \leftarrow P \cup \{\boldsymbol{p}\}$
end while
return f

Clustered peaks are drawn from a normal distribution $N(\boldsymbol{p}_1, \frac{n}{36}\mathbf{I})$. More complicated arrangements with more than one funnel are of course possible in principle. The function createInstance in Algorithm 2 takes the set of peaks and the desired topology as inputs. The topology is necessary as argument, because if we want a funnel structure, additionally to the clustering also the height values are redistributed among the peaks so that they shrink with increasing Euclidean distance from the global optimum $\boldsymbol{p}_1$. Figure 1.2 shows examples of the resulting problems for $n = 2$ and $\nu = 100$.

To evaluate the indicators basin ratio and basin inaccuracy, we are using the

Algorithm 3 getBasinOptimum($\boldsymbol{x}$)

Input: solution $\boldsymbol{x}$
Output: position of local optimum

1: $\boldsymbol{p}_{\text{curr}} \leftarrow \boldsymbol{x}$
2: **repeat**
3: $\quad \boldsymbol{p}_{\text{prev}} \leftarrow \boldsymbol{p}_{\text{curr}}$
4: $\quad \boldsymbol{p}_{\text{curr}} \leftarrow \arg\max\{g(\boldsymbol{p}_{\text{curr}}, \boldsymbol{p}) \mid \boldsymbol{p} \in P\}$
5: **until** $\boldsymbol{p}_{\text{prev}} = \boldsymbol{p}_{\text{curr}}$ // if fulfilled, we have arrived at local optimum
6: **return** $\boldsymbol{p}_{\text{curr}}$

heuristic displayed in Algorithm 3 to identify the basin a point belongs to. From an arbitrary position $\boldsymbol{x} \in \mathcal{X}$, the algorithm jumps to the peak $\boldsymbol{p}_{\text{curr}}$ which is responsible for $f(\boldsymbol{x})$. As $\boldsymbol{p}_{\text{curr}}$ itself may be masked by another peak, the procedure is iterated until $\boldsymbol{p}_{\text{curr}}$ is an optimum. Thus, the set basin($\boldsymbol{x}^*$) for an optimum position $\boldsymbol{x}^* \in \mathcal{O}$, which is mentioned in Section 2.1.3, is approximated as

$$\text{basin}(\boldsymbol{x}^*) \approx \{\boldsymbol{x} \in \mathcal{X} \mid \text{getBasinOptimum}(\boldsymbol{x}) = \boldsymbol{x}^*\}\,.$$

Note that this realization via Algorithm 3 is not completely accurate in the sense of Definition 2, because we may jump over small basins that a descent algorithm would have converged to. However, when assessing an approximation set $\mathcal{P}$ by BR or BI, the error introduced by this approach should decrease with increasing amount of local search that has been spent on $\mathcal{P}$.

The test problem generator MPM2 fulfills the same desirable properties as the max-set of Gaussians by Gallagher and Yuan [47]. They characterize appropriate test functions as difficult to solve using simple methods such as hill climbing algorithms (P1); nonlinear, nonseparable, and nonsymmetric (P2); scalable in terms of problem dimensionality (P3); scalable in terms of time to evaluate the objective function (P4); tunable by a small number of user parameters (P5); able to be generated at random and difficult to reverse engineer (P6); and exhibiting an array of landscape-like features (P7). Property P4 is of course not really required; it can be imitated by considering different budgets of function evaluations. A corollary of P6 is that the global optimum should be distributed uniformly in $\mathcal{X}$, to avoid unintentional advantages for some solution strategies. This requirement is fulfilled by MPM2, but according to the BBOB authors, it is not given in the BBOB setup [54].

2.4 Experimentation

Simultaneously to the complexity of used test problems, also the complexity of experimental analyses has increased over the years, at least in the area of evolutionary computation. Therefore, we will give a short introduction to design and analysis of experiments, loosely based on [34, pp. 3–16]. Figure 2.5 shows a categorization of variables of real-world processes. These variables can be divided into independent

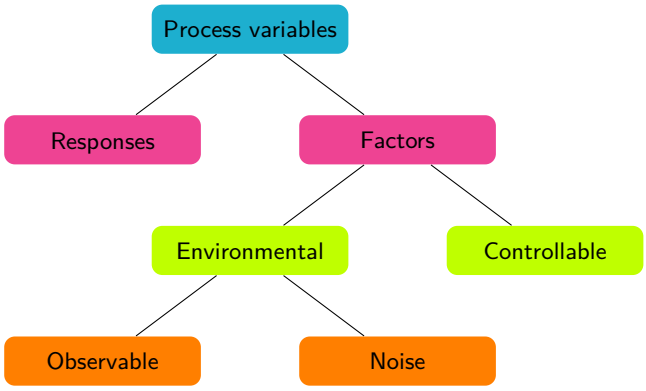


Figure 2.5: Different categories of process variables.

variables (factors) and dependent variables (responses), which are the inputs and outputs, respectively, of the system/process under consideration. Responses are in our case usually performance measures as the ones we defined in Section 2.2. An experiment simply encompasses the runs of the process at a set of test locations. A *designed* experiment is an experiment in which the test locations are planned by the experimenter, and this set of locations is called the experimental design. Naturally, the planning is only possible for controllable factors. In real-world applications, some of the factors cannot be controlled or even observed. This applies especially to most features of optimization problems. However, as we are conducting computer experiments, all of the factors can be actually controlled by us for benchmarking purposes. For example, Section 2.3 already discussed the control of problem features in detail. Generally, also all random aspects of our experiments can be modeled by using pseudorandom numbers, which are produced by an actually deterministic generator. By setting a *seed* for this random number generator (RNG), every experiment is exactly reproducible. Furthermore, by using the same set of seeds for the stochastic replications of each configuration, we can achieve a variance reduction of the responses. This technique of *common random numbers* (CRN) [93] is standard in all experiments in this work.

It should be noted that although we are conducting computer experiments, our experiments do not adhere to the assumptions usually made under the term *design and analysis of computer experiments* (DACE). The assumptions of the DACE paradigm encompass a deterministic response [123] or at least a very low measurement error [75]. In this situation replications do not make much sense. Furthermore, the response is assumed to be multimodal. As a consequence, it is important to choose an experimental design that samples in the interior of the region of interest (*space-filling*) and, if the response is to be modeled by a meta-model, a global model such as Kriging must be used [123]. Interestingly, the DACE assumptions correspond quite closely to the situation in global and multimodal optimization. So, methods from this field will appear repeatedly in our optimization algorithms. The experiments to analyze these algorithms, however, exhibit the properties of conventional

experiments [101], most importantly a high noise whose distribution is not very dependent on the factors considered in the experiment. Thus, we are using full-factorial designs (which are not space-filling) and many replications in these experiments.

A general guiding principle, promoted by Taguchi (see, e. g., [34, pp. 223–234] or [123]), says to incorporate uncontrollable factors into the experimental design, to obtain results exhibiting robustness later in the real world. This abstract principle can be instantiated in different ways, depending on the design paradigm governing the experiment. However, in any case it greatly increases the computational effort and complicates the experimental analysis. Furthermore, an important distinction of uncontrollable factors is if they are (easily) observable or not, because this has implications for their treatment in real-world applications. A problem's number of decision variables n, for example, is always known to us and therefore an optimization algorithm is not required to possess a good performance over all possible values of n. Instead we can always choose the most appropriate algorithm for the current n before we begin the optimization (given that we somehow possess this expert knowledge). This room for choice will also always be reflected in our experimental analyses. The general approach is to select the best configuration among the control factors, regarding all possible levels of noise factors, but depending on particular levels of the observable factors.

It is important to note that we are always benchmarking concrete implementations and not the mathematical algorithms themselves. Therefore, the origin of the used software should be ideally communicated along with the experimental setup. The reporting on experiments will be done largely according to the scheme of Preuss [113], which consists of research question, pre-experimental planning, task, setup, results, observations, and discussion. This clear structure aims to facilitate the distinction of objective results and subjective comments, and to improve the reproducibility of results without access to the implementation. Regarding the analysis of the obtained data, Montgomery emphasizes the importance of visualizations in his practical guidelines, especially for presenting the results to others [101, p. 16]. Thus, we will mostly use trellis graphics to investigate on the interactions of the various factors [15].

3 Summary Characteristics for the Assessment of Experimental Designs

The global stages of optimization algorithms usually rely on certain sampling algorithms to produce starting points for the local stage. Interestingly, the design of computer experiments has quite similar requirements (see Section 2.4). In both cases, a diverse (initial) sample does not seem to be a bad idea in general, but there are certainly more things to consider. In this chapter, we will concentrate on the assessment of the produced point sets. We give a motivation why diversity is needed and discuss some additional criteria, which are related more closely to sequential designs and model-based approaches. Based on topics that seem to recur frequently in the literature and the authors own preferences, the following aspects shall receive the most attention:

Diversity As already seen in Section 2.1.1, *diversity* is an umbrella term for several slightly different concepts. The diversity in the design should be high, because for deterministic functions, repeated sampling of points or sampling in the immediate vicinity of neighboring points makes not much sense. Additionally, if no prior information is available on the response, the whole region of interest should be covered. These two requirements together lead to a desire for a *uniform* distribution of points. In Monte Carlo methods for numerical integration, where it is generally necessary to give each area equal influence, it is common to measure the deviation from uniformity, called discrepancy. Low-discrepancy point sets are also explicitly known as uniform designs in the area of computer experiments [42]. *Space-filling* designs are a broader class, because they also cover the whole region of interest, but not necessarily uniformly [118]. They are also called *exploratory* designs [126, p. 125].

Low-dimensional projections Projections of the design into lower dimensions should not contain redundant points, to always obtain the best possible performance. The argument is the same as for diversity, because if some variable has no or only weak influence on f, points which only differ in this variable essentially collapse into the same point. Again, as we are dealing with deterministic functions, this yields no or little additional information. Stinstra et al. [133] call a good design in this regard *non-collapsing*. Törn and Žilinskas [143, p. 33] call the collapsing of points the *shadow effect*. In principle, the argumentation also holds if there is only a lack of interaction between decision variables, i. e., if f can be written as a sum of lower-dimensional functions [88, 109]. More generally speaking, not only redundant points should be avoided, but

also the diversity of the low-dimensional projections should be maximal. This would also improve the estimation of generalized main and interaction effects in the analysis of experiments [153], which is another application of numerical integration.

Irregularity Some authors in the literature already describe point sets as *regular* if there is a noticeable repulsion between points [63, p. 2]. Others use it in a stricter sense, meaning that points of a design form a lattice [128, 78]. We will adopt the latter interpretation, while the former case rather corresponds to (high) uniformity. Regarding meta-modeling, some irregularity is desired to avoid aliasing (e. g., Moiré patterns or similar effects). It is also widely believed that irregularity aids the estimation of model parameters, when a meta-model is fitted on the design points. Irregularity can, e. g., be measured by considering the variation of inter-point distances and thus is somewhat conflicting to diversity.

Run time As stated in the goals defined in Section 1.5, we plan to invest a relatively large budget of function evaluations and also to investigate high numbers of decision variables. In this case, of course a large number of points is necessary in the global stage to obtain a sufficiently dense sample. Therefore, the run times of the sampling algorithms and the quality indicators should be low, so that also relatively large designs can be generated and evaluated quickly.

Sequentiality We want to use sampling in the global stage of an iterative optimization algorithm. Therefore, the sampling algorithm should be able to take existing data of the previous iterations into account, e. g., by identifying less explored regions. This topic receives traditionally a lot of attention in the area of design and analysis of computer experiments [123], but so far not so much in model-free optimization. As sequential designs do not require any new performance measures, the topic is deferred to Chapter 5, where we deal with the sampling algorithms themselves.

Except for the last item, the different mentioned aspects guide the definition of corresponding indicators. In the following, possible candidates are discussed (apart from diversity indicators, which were already dealt with in Section 2.1.1) and for each respective topic, one representative is chosen. Additionally, cheap linear-time indicators are investigated. Especially, a workaround is sought to indirectly characterize designs with low covering radius, because its high run time of $O((nN)^{\lfloor n/2 \rfloor})$ prohibits a reasonable application of the indicator in practice.

3.1 Low-dimensional Projections

In recent years, low-dimensional projections of point sets received increased attention in the quasi-Monte Carlo literature [60, 150]. This is based on the insight that numerical integration of high-dimensional functions f can only be successful when

the effective dimension is low, that is, if f can be approximated well by a sum of low-dimensional functions [88]. Consequently, it is advisable to adapt discrepancy measures to only or additionally consider low-dimensional subspaces. Such measures are proposed by Hickernell [60] and Wang and Sloan [150].

In the area of design and analysis of computer experiments, the topic is present, too. Maybe the issue is even more pressing here due to the smaller budgets [105]. Santner et al. [126, p. 141] propose the average projection design criterion

$$\mathrm{av}_q(\mathcal{P}) = \left(\frac{1}{\sum_{j \in J} \binom{n}{j}} \sum_{j \in J} \sum_{k=1}^{\binom{n}{j}} [D(\mathcal{P}_{kj})]^q \right)^{1/q}$$

to assess the diversity of low-dimensional projections of the design, where $\mathcal{P}_{kj}$ is the k-th j-dimensional projection of $\mathcal{P}$ and D is some diversity indicator. Of course, we again encounter the question which diversity to measure. Santner et al. propose AID. Although SPD may be even more favorable, it is out of the question due to its high run time. av_q is expensive to compute anyway, as there are in general exponentially many low-dimensional projections. Therefore, usually only the two-dimensional projections are considered. It seems admissible to favor a maximin-type diversity over discrepancy here, because the edge effects are less severe in low-dimensional spaces. Thus, we choose av_1 with AID as diversity indicator (in accordance to the original definition), as it does not rule out true uniformity in the original $\mathcal{X}$.

3.2 Irregularity

There seems to be considerable debate about whether an experimental design should be uniform or not. Dette and Pepelyshev [35] show in several experiments, that designs with a higher point density in boundary regions achieve a lower integrated mean squared error (IMSE, estimated by a large Monte Carlo sample) when using Kriging as meta-model and three to eight input dimensions, although this seems to be in contradiction with Santner et al. [126, p. 170], who say that IMSE-optimal designs tend to locate points away from the boundary. Pronzato and Müller [118] define "estimation designs", optimizing different criteria depending on whether activity parameters for the model have to be estimated or not. Generally, it seems advisable to use more irregular designs when parameters have to be estimated. Santner et al. [126, p. 160] (who assume here that Sobol' sequences are more irregular than optimized LHDs) also venture this guess:

> "If a greater variety of inter-point distances provides more information about the correlation parameters (presumably improving prediction accuracy), then designs based on a Sobol' sequence (or other types of sequences that have been used in numerical integration) may be preferable to the LHD."

Pronzato and Müller [118] note that these irregular designs may also be those of Dette and Pepelyshev [35]. However, all these considerations are usually aimed at the mean squared error over the whole region of interest for meta-modeling, which is not necessarily decisive for global or multimodal optimization.

One possible measure of irregularity could be the sample variance of nearest-neighbor distances

$$s^2_{\mathrm{nn},\mathcal{P}} = \frac{1}{N-1} \sum_{i=1}^{N} \left(d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{P}) - \bar{d}_{\mathrm{nn},\mathcal{P}} \right)^2 , \tag{3.1}$$

where $\bar{d}_{\mathrm{nn},\mathcal{P}} = 1/N \sum_{i=1}^{N} d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{P})$ is the mean nearest-neighbor distance. A high variance indicates high irregularity. Also PDNN might be a useful measure, because for a given $\bar{d}_{\mathrm{nn},\mathcal{P}}$ the product is maximal when all nearest-neighbor distances are equal. However, the relationship between irregularity and variance seems more obvious, so $s^2_{\mathrm{nn},\mathcal{P}}$ is chosen.

3.3 Distance to the Boundary

Johnson et al. [66] point out that the main difference between minimax and maximin designs is that the former tend to avoid the boundaries, while the latter do not. Therefore, it seems promising to carry out some experimental analysis regarding the boundary behavior of different sampling methods. It would be even more desirable to have sampling methods with controllable behavior. As a first step in this direction, let us define a measure to quantify the proximity of a point set to the boundary.

Proposition 4. *The distance between a point $\boldsymbol{x} \in \mathcal{X}$ and the nearest neighbor on the boundary $\mathcal{B} = \{\boldsymbol{x} \in \mathcal{X} \mid \exists i \in \{1, \ldots, n\} : x_i = u_i \vee x_i = \ell_i\}$ is under every L_p norm*

$$d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B}) = \min_{1 \le i \le n} \left\{ \min\{x_i - \ell_i, u_i - x_i\} \right\}$$

Proof. Because the boundaries of $\mathcal{X}$ are paraxial, there exists a $\boldsymbol{y} \in \mathcal{B}$ with $d(\boldsymbol{x}, \boldsymbol{y}) = d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$, which only differs from $\boldsymbol{x}$ in one variable i. As in one-dimensional space all L_p distances are identical to the absolute difference, the distance must be the smaller one of $x_i - \ell_i$ and $u_i - x_i$. □

Proposition 5. *The expected distance between a random uniform point X in $[0,1]^n$ and the boundary $\mathcal{B}$ is*

$$\delta_n := \mathrm{E}(d_{\mathrm{nn}}(X, \mathcal{B})) = \frac{1}{2(1+n)} .$$

Proof. The expected distance to the lower boundaries is identical to the first order statistic $X_{(1)}$ (the minimum) of a random sample $X_1, \ldots, X_n$ from $U(0,1)$. $X_{(1)}$ belongs to a $\mathrm{Beta}(1,n)$ distribution [7, pp. 13–14], whose mean is $1/(1+n)$. To account for the upper boundaries, too, it is sufficient to consider $Y_i \sim U(0, 0.5)$

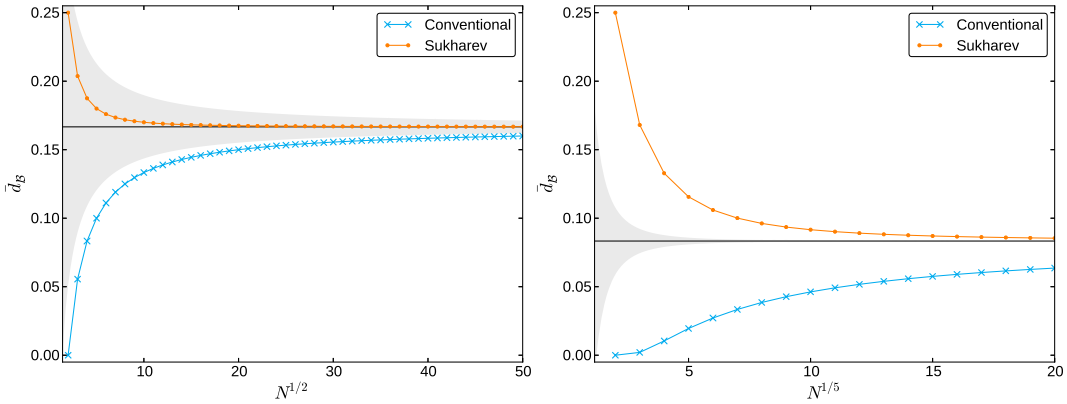


Figure 3.1: Distance to the boundary for conventional and Sukharev grids in two (left) and five dimensions (right). The horizontal black line indicates δ_n. The gray area represents a 95% confidence interval for δ_n under the conditions of Proposition 6.

instead, because $0 \leq Y_i = \min\{X_i - \ell_i, u_i - X_i\} = \min\{X_i, 1 - X_i\} \leq 0.5$. Therefore, the sought quantity is $\mathrm{E}(Y_{(1)}) = \mathrm{E}(0.5 \cdot X_{(1)}) = 0.5 \cdot \mathrm{E}(X_{(1)}) = 0.5 \cdot 1/(1+n)$, due to linearity of expectation [18, p. 77]. □

As we can see, the expected distance to the boundary decreases with increasing dimension. This is just another manifestation of the curse of dimensionality and shows us that in high dimensions, almost all of the space is in the boundary regions [63, p. 183]. We can now use the sample mean

$$\bar{d}_{\mathcal{B}} = \frac{1}{N} \sum_{i=1}^{N} d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{B}) \tag{3.2}$$

to estimate how much emphasis a point set puts on the boundary in comparison with the uniform distribution. The interesting thing about this criterion is that we are using a Monte Carlo estimate to assess the quality of our point set. Although it alone is not sufficient for getting the whole picture, it is attractive because it is a necessary condition for uniformity and can be computed in linear time. Finally, we formalize the observation of Johnson et al. [66], regarding the boundary behavior, in the following conjecture.

Conjecture 2. *Point sets with maximal separation distance (maximin designs) possess $\bar{d}_{\mathcal{B}} < \delta_n$ and point sets with minimal covering radius (minimax designs) exhibit $\bar{d}_{\mathcal{B}} > \delta_n$.*

This conjecture may be only a rule of thumb, but it is certainly true for the conventional grid and the Sukharev grid (cf. Figure 1.1), which are the optimal solutions under the L_∞ norm regarding separation distance and covering radius, respectively [136][79, pp. 202–203]. The distance to the boundary is shown in Figure 3.1, where

we can also see that $\bar{d}_{\mathcal{B}}$, the Monte Carlo estimate for δ_n, becomes more precise with increasing N, but this estimate is often less accurate than what we would expect in 95% of the cases with random uniform points. For other norms, the optimal point sets can in the general case only be approximated, but Sections 4.4 and 4.5 also indirectly support the conjecture.

Niederreiter [108, p. 152] shows that every low-discrepancy point set also is a low-dispersion point set (but not vice versa). This raises hope that combining a diversity indicator as DISC (or one of the maximin family) with a side constraint $\bar{d}_{\mathcal{B}} - \delta_n \geq \varepsilon$ yields a good criterion to obtain point sets with a small covering radius. If, on the other hand, an exactly uniform point set is sought, we can compare the deviation $|\bar{d}_{\mathcal{B}} - \delta_n|$ to the expected value for random uniform point sets.

Proposition 6. *For a set of random uniform points $\mathcal{P}$ in $\mathcal{X} = [0,1]^n$, $|\mathcal{P}| = N$, with $\bar{d}_{\mathcal{B}}$ computed as in (3.2), the mean absolute deviation around the mean δ_n is for large N*

$$\mathrm{E}(|\bar{d}_{\mathcal{B}} - \delta_n|) \approx \sqrt{\frac{2}{\pi}} \cdot 0.5 \cdot \sqrt{\frac{n}{(n+1)^2(n+2)N}}\,.$$

Proof. As shown in the proof of Proposition 5, $d_{\mathrm{nn}}(X, \mathcal{B}) \sim 0.5 \cdot \mathrm{Beta}(1, n)$ for $X \sim U(0,1)$ and therefore its standard deviation is [67, p. 217]

$$\sigma(d_{\mathrm{nn}}(X, \mathcal{B})) = 0.5 \cdot \sqrt{n} / \sqrt{(n+1)^2(n+2)}\,.$$

As N is typically large, we can now apply the central limit theorem, which says that the mean of N independent identically distributed random variables with standard deviation σ is asymptotically normally distributed with standard deviation $\sigma/\sqrt{N}$ [18, p. 357]. Therefore, $\bar{d}_{\mathcal{B}}$ converges in distribution to $N(\delta_n, \sigma^2/N)$. The mean absolute deviation around the mean then approximately follows the half-normal distribution, whose expected value in this case is $\sqrt{2/\pi} \cdot \sigma/\sqrt{N}$. (The half-normal distribution is the special case of the χ distribution with $n = 1$ [68, p. 417], see the proof of Proposition 8 for the general formula of the expected value.) □

3.4 Distance between Center of Mass and Centroid of the Hypercube

Also the distance of the sample's center of mass $\bar{\boldsymbol{c}}_{\mathcal{P}} = 1/N \sum_{i=1}^{N} \boldsymbol{x}_i$ from the centroid of the hypercube $\boldsymbol{c}_{\mathcal{X}} = (\boldsymbol{\ell} + \boldsymbol{u})/2$ may be used as a quality measure. It is another Monte Carlo estimate that can be computed in linear time and we are able to analytically derive the expected value for random uniform sets as a reference.

Proposition 7. *The expected L_1 distance between $\bar{\boldsymbol{c}}_{\mathcal{P}}$ and $\boldsymbol{c}_{\mathcal{X}} = (0.5, \ldots, 0.5)^\top$ for a set of random uniform points $\mathcal{P}$ in $\mathcal{X} = [0,1]^n$, $|\mathcal{P}| = N$, is for large N*

$$\mathrm{E}(\|\bar{\boldsymbol{c}}_{\mathcal{P}} - \boldsymbol{c}_{\mathcal{X}}\|_1) \approx n \cdot \sqrt{\frac{2}{\pi}} \cdot \frac{1}{\sqrt{12N}}\,.$$

Proof. First of all, we note that it suffices to regard the one-dimensional distances, because

$$\mathrm{E}(\|\bar{\boldsymbol{c}}_{\mathcal{P}} - \boldsymbol{c}_{\mathcal{X}}\|_1) = \mathrm{E}\left(\sum_{i=1}^{n} |\bar{c}_{\mathcal{P},i} - 0.5|\right) = \sum_{i=1}^{n} \mathrm{E}(|\bar{c}_{\mathcal{P},i} - 0.5|) \ .$$

The standard deviation of a random uniform variable on $[0, 1]$ is $1/\sqrt{12}$ [67, p. 279]. Again using the central limit theorem, we obtain that $\forall i \in \{1, \ldots, n\}$, $\bar{c}_{\mathcal{P},i}$ converges in distribution to $N(0.5, \sigma^2_{\bar{c}_{\mathcal{P},i}})$, where $\sigma_{\bar{c}_{\mathcal{P},i}} = 1/\sqrt{12N}$ is both the standard error of the mean and the standard deviation of the estimate $\bar{c}_{\mathcal{P},i}$. Finally, $|\bar{c}_{\mathcal{P},i} - 0.5|$ again approximately follows a half-normal distribution with expected value $\sqrt{2/\pi} \cdot \sigma_{\bar{c}_{\mathcal{P},i}}$, which we only have to multiply by n to obtain the expected L_1 distance in n dimensions. □

In the light of the results of Aggarwal et al. [5], this reference value may be useful in high dimensions. On the other hand, it seems inappropriate to assess a $\bar{\boldsymbol{c}}_{\mathcal{P}}$ with radially symmetric distribution using the L_1 distance, which is not radially symmetric. Luckily, we can also derive a similar result for the L_2 distance:

Proposition 8. *The expected value for the L_2 distance $d_{\bar{\boldsymbol{c}}\boldsymbol{c}} := \|\bar{\boldsymbol{c}}_{\mathcal{P}} - \boldsymbol{c}_{\mathcal{X}}\|_2$ between $\bar{\boldsymbol{c}}_{\mathcal{P}}$ and $\boldsymbol{c}_{\mathcal{X}} = (0.5, \ldots, 0.5)^\top$ for a set of random uniform points $\mathcal{P}$ in $\mathcal{X} = [0, 1]^n$, $|\mathcal{P}| = N$, is for large N*

$$\mathrm{E}(d_{\bar{\boldsymbol{c}}\boldsymbol{c}}) \approx \epsilon_{N,n} := \sqrt{2} \cdot \frac{\Gamma((n+1)/2)}{\Gamma(n/2)} \cdot \frac{1}{\sqrt{12N}} \ ,$$

where $\Gamma(\cdot)$ is the Gamma function.

Proof. As we already know from the proof of Proposition 7, $\bar{\boldsymbol{c}}_{\mathcal{P}}$ approximately follows a multinormal distribution with covariance matrix $1/12N \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The Euclidean norm of a vector of n independent standard normally distributed variables follows a χ distribution with n degrees of freedom [68, pp. 415–417]. Thus, its expected value is $\sqrt{2}\Gamma((n+1)/2)/\Gamma(n/2)$ [68, p. 421], which only has to be scaled with the appropriate standard deviation to obtain the result. □

For $n = 1$, Propositions 7 and 8 yield the same formula, as required. Again, note that there is no point in directly optimizing $d_{\bar{\boldsymbol{c}}\boldsymbol{c}}$ or $|\bar{d}_{\mathcal{B}} - \delta_n|$, because they are not sufficient conditions for uniformity. However, it should be useful to apply these measures to given point sets to detect potentially undesired deviations from uniformity.

3.5 Testing the Linear-time Indicators

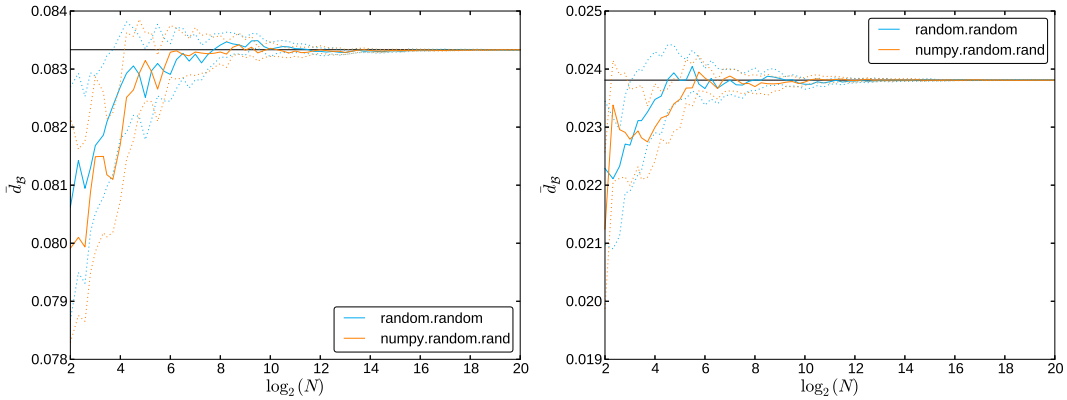
To verify the correctness of the proposed measures (and the pseudorandom numbers), we carry out an experimental analysis. Random uniform point sets are generated with two commonly available pseudorandom number generators in the Python

programming language. One is the built-in RNG of Python 2.7 ("random.random"), the other is the function "random.rand" of the NumPy library 1.6. Both use independent implementations of the Mersenne twister algorithm, with a period length of $2^{19937} - 1$. Their empirical properties are compared to the theoretical values for the distance to the boundary and the deviation from the centroid of the hypercube in five and twenty dimensions. Figure 3.2 shows the results of this experiment. In this figure, the black horizontal lines indicate the respective expected values that were derived in the previous propositions for random uniform points. The measured values largely exhibit a good agreement with the predictions, only $|\bar{d}_{\mathcal{B}} - \delta_n|$ in Figure 3.2b slightly deviates downwards. This may be due to the highly skewed Beta distribution involved, which slows down the convergence to the normal distribution.

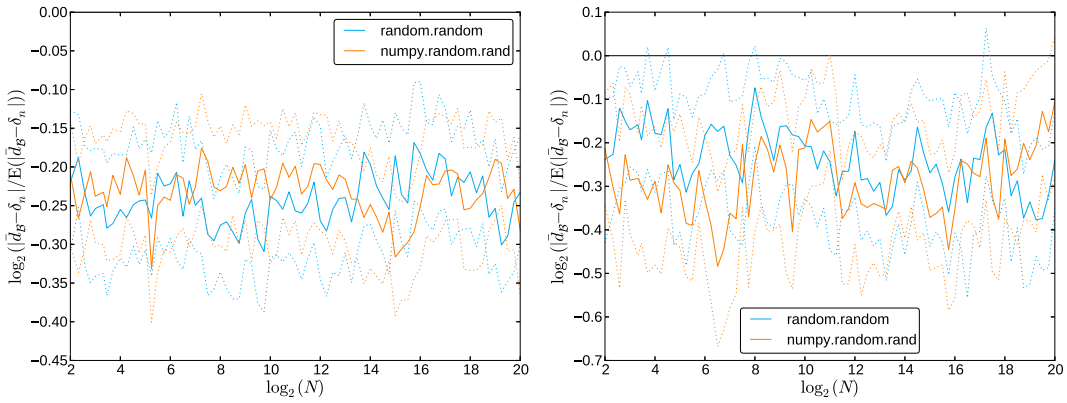
3.6 Other Criteria for Experimental Designs

There exist many optimality criteria for designs of experiments, which are mostly relevant for "classical" response surface methodology. Some of them, i. e., G-, D- and A-optimality, have been adapted to space-filling designs [66]. To do this, the original definitions have been changed slightly, which makes the situation a bit confusing because of the overloaded notation. Santner et al. [126, pp. 69–76] advise against using D-optimal designs, based on a parameter study regarding different meta-models and configurations. On the other hand, they show that maximum entropy designs actually minimize the D-criterion [126, pp. 164–167]. Note that in this case entropy regarding the responses is meant. Johnson et al. [66] in turn show that for vanishingly small correlation between sample locations, maximin designs tend to be asymptotically D-optimal and minimax designs tend to be G-optimal. Pronzato and Müller [118] discuss entropy in the search space as a measure for uniformity. However, their entropy has to be estimated by taking detours via kernel density estimation or minimum spanning trees, which makes it rather uninteresting for our purposes. This is also the reason for not including entropy as an indicator in Section 2.1.1. Saka et al. [124] use four uniformity measures based on Voronoi tessellations, among them the covering radius. In their experiments, also these measures are approximated by Monte Carlo methods, which is a huge disadvantage. Bursztyn and Steinberg [21] propose an "alias sum of squares criterion", which is reportedly related to the IMSE of a polynomial regression model. This, and other model-based criteria, e. g., by Pronzato and Müller [118], seem too specific for a general summary characteristic.

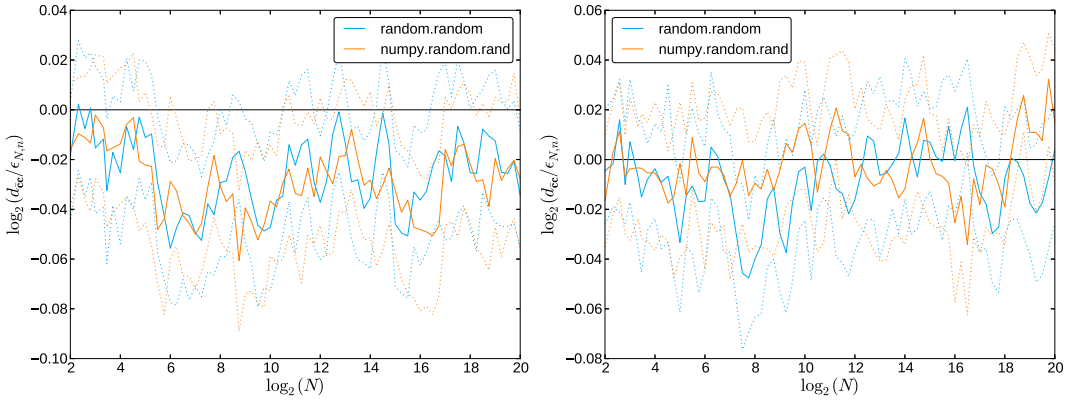
Lagae and Dutré [78] use the Fourier transform to obtain two functional summary characteristics, namely the *radially averaged power spectrum* and *anisotropy*. However, these functions are averages over a number of point samples, and thus merely useful for comparing the distributional properties of different sampling algorithms. Another numerical summary characteristic proposed by these authors is briefly mentioned in Section 4.4.



(a) Mean distance to the boundary.



(b) Relative deviation from the expected distance to the boundary.



(c) Relative Euclidean distance between $\mathcal{P}$'s center of mass and the centroid of the hypercube.

Figure 3.2: Pseudorandom numbers for $n = 5$ (left) and $n = 20$ (right). The number of replications for each subfigure was chosen as $\lfloor 10^4/n \rfloor$. Solid lines mark the median and dotted lines 95% confidence intervals.

4 Sampling

In this chapter, possible algorithms to generate designs are discussed. Unfortunately, optimizing design criteria is in general expensive and we want to avoid having another expensive optimization problem inside our optimization algorithm. So, we will focus on rather simple algorithms for generating experimental designs. Sacks et al. [123] would probably call them "less sophisticated", but they explicitly acknowledge that there is a need for such algorithms.

4.1 Latin Hypercube Designs

A classical approach of generating space-filling designs for computer experiments are latin hypercube designs (LHD). An LHD is defined as a set of points $\mathcal{D} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, where each set $\{z_{1,j}, \ldots, z_{N,j}\}$, $j = 1, \ldots, n$, is a random permutation of the numbers $1, \ldots, N$. Depending on the user's preferences, one can use different approaches to normalize $\mathcal{P}$ to the region of interest $[0, 1]^n$:

1. McKay et al. [94] combine the approach with random sampling, so that $x_{i,j} = (z_{i,j} - U(0, 1))/N$. The perturbation protects us against misleading samples in case of a periodic signal [46, p. 29]. The approach is also called latin hypercube sampling (LHS) or N-rooks sampling [130].

2. Alternatively, one can generate a centered LHD by choosing $x_{i,j} = (z_{i,j} - 0.5)/N$.

3. Finally, we could arrange points in a way that the edges are covered, i. e., $x_{i,j} = (z_{i,j} - 1)/(N - 1)$ (see [46, p. 17]).

This construction guarantees that the one-dimensional projections of the design are perfectly uniform (apart from perturbations), because each dimension is partitioned into N bins. As already mentioned in Section 3.1, this is a desirable property for several applications. Saka et al. [124] observe that enforcing the LHD property on arbitrary point sets by post-processing improves the L_∞ star discrepancy of the point sets. Another argument in favor of LHDs is that generating them from scratch is cheap, requiring only $O(Nn)$ time. Stein [132] even shows theoretically that in certain cases, LHDs have superior properties to random uniform samples. Santner et al. [126, pp. 148–149] summarize the situation:

> "LHDs are superior, in many situations, to designs based on simple random sampling for estimating mean responses. [...], for large sample sizes

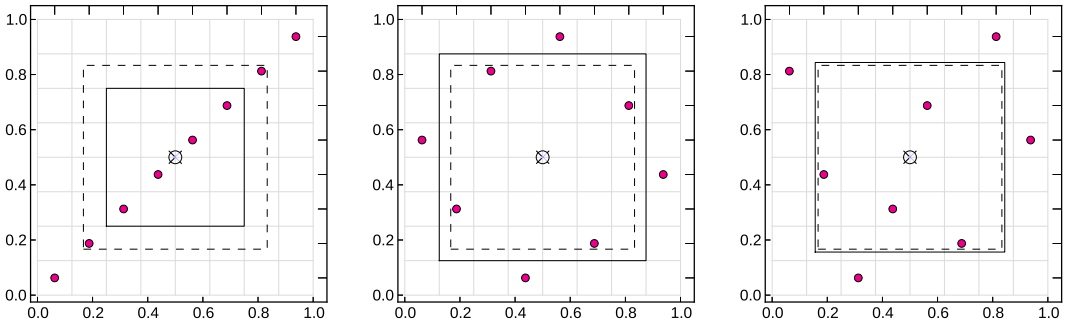
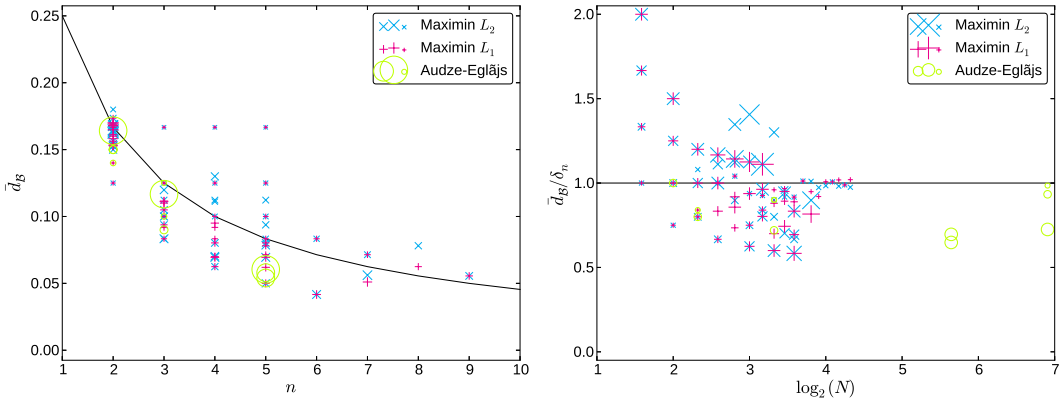


Figure 4.1: Three examples of centered LHDs. Each row and column contains exactly one point. The solid cube indicates $\bar{d}_{\mathcal{B}}$ and the dashed cube indicates δ_n. Cross and circle mark $\bar{\boldsymbol{c}}_{\mathcal{P}}$ and $\boldsymbol{c}_{\mathcal{X}}$, respectively. One-dimensional projections of the points are indicated at the top and right axis.

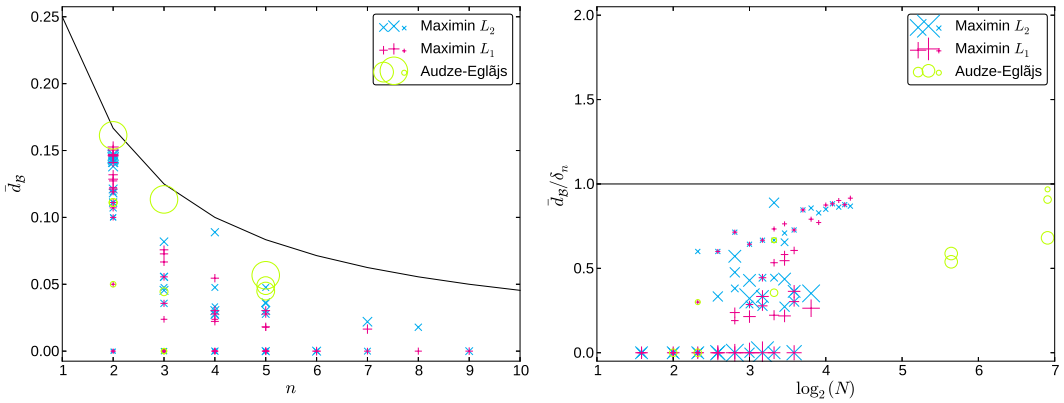
> [...] this is true unless no main effects are present. For small sample sizes, these results hold if the response is a monotonic function of the inputs, and the function of the response that is of interest is also monotonic in the response. [...] In sum, it has not been demonstrated that LHDs are superior to any designs other than simple random sampling (and they are only superior to simple random sampling in some cases). It is our belief that LHDs are popular because (i) LHDs are relatively easy to generate and (ii) their projections onto one-dimensional subspaces are evenly spread (or can be tailored to a desired target spread over a given dimension)."

However, Figure 4.1 shows that the LHD property does neither guarantee a space-filling design nor prevent large deviations from δ_n, the expected distance to the boundary for random uniform points.

The idea of latin hypercube sampling was apparently first mentioned by Audze and Eglãjs [8] and McKay et al. [94]. The former authors are also widely credited for introducing the potential energy E_2, see (2.2), as additional uniformity criterion to avoid situations like in Figure 4.1, where the points are aligned along a diagonal of the hypercube. Another self-evident approach would be to choose an LHD that maximizes the separation distance. If we consider non-perturbed LHDs, this criterion yields many optimal designs. To further discriminate among them, one could also regard the second-smallest, third-smallest, etc. distance, as proposed by Morris and Mitchell [105]. However, for the optimization they revert to the general potential energy definition E_q, with different values for q. While [105] predates the result of Hardin and Saff [57], that $q \geq n$ must hold for obtaining an asymptotically uniform distribution, by nine years, it seems that this fact has not even been recognized at any later point by the LHD community [118]. Figure 4.2 shows the distance to the boundary for some LHDs taken from the literature, namely maximin LHDs published by Morris and Mitchell [104] and Audze-Eglãjs LHDs published by Bates



(a) Centered LHDs.



(b) Edge LHDs.

Figure 4.2: Distance to the boundary for some LHDs in the literature. Left and right panels show the same data. The left one shows the distance to the boundary versus the dimension, the right one versus the number of points. The respective other number is indicated by marker size. The black lines indicate the expected values for random uniform points.

et al. [12]. The distances have been computed for centered and edge LHDs, although the original papers seem to describe only edge LHDs. This was done to show that the decision influences especially the smaller designs. Remarkably, all Audze-Eglãjs LHDs satisfy $\bar{d}_{\mathcal{B}} \leq \delta_n$, indicating a strong drift towards the boundary. Also all "edge variants" of the maximin LHDs satisfy $\bar{d}_{\mathcal{B}} \leq \delta_n$, in accordance with Conjecture 2.

To optimize all the mentioned criteria, one has to search for adequate instances within the class of LHDs. For this task, e. g., local searches, simulated annealing, and several flavors of evolutionary algorithms have been applied [62, 82, 105, 32]. As already one objective function evaluation takes $O(N^2 n)$ time, this search can become quite costly. However, the argumentation in the literature is that this effort is negligible because LHDs are mostly used for expensive optimization and $N \geq 1000$

is already considered as large. Moreover, once the optimal designs are found, they could be stored and then be reused any number of times [62, 82]. This perspective is less valid when we consider model-based optimization. First, designs that explicitly consider a certain meta-model can achieve a better performance (and are likely non-uniform) and second, randomized designs may make our optimization method generally more robust (see [118] for both arguments). We therefore strive for "throw-away" designs that are randomized and relatively cheap to generate.

Beachkofski and Grandhi [13] propose an algorithm to generate LHDs with improved n-dimensional distribution. The proposal is a greedy heuristic that begins with a single random point and sequentially adds the best point of a random sample until the LHD is complete. We will call it improved latin hypercube sampling (ILHS) here. In each iteration i, $f_{\text{dup}}(N-i)$ candidates are considered. $f_{\text{dup}} \in \mathbb{N}$ is a duplication factor that is usually chosen smaller than ten. The candidates are randomly chosen points not violating the LHD property if added to the design. The selection criterion in this case minimizes $|d_{\text{nn}}(\boldsymbol{x}, \mathcal{P}) - d_{\text{opt}}|$, the deviation from a supposedly ideal distance $d_{\text{opt}} = N/\sqrt[n]{N}$ to the already chosen points $\mathcal{P}$. Beachkofski and Grandhi make no mention of asymptotic run times, but the approach clearly leads to a cubic number of distance computations, as the outer loop runs $N-2$ times and in each iteration a distance matrix of size $i \cdot f_{\text{dup}}(N-i)$ is computed. The approach has another drawback: For the last points added to the design, only very few alternatives are considered. While it is true that there is only one possibility to choose the last point, for the second to last point there are 2^n alternatives, yet the point is chosen from only $2f_{\text{dup}}$ candidates. Still, Saka et al. [124] observed much better results for ILHS compared to conventional LHS.

An implementation of ILHS is, e. g., available for the R programming language in the package lhs[1]. The package is employed by various other packages and also contains a function to generate approximate maximin LHDs based on the same heuristic approach, which shows the relative popularity the algorithm has gained.

4.1.1 Fast Improved Latin Hypercube Sampling

We will now consider a modification of ILHS that runs in $O(N^2 n)$, while simultaneously improving its distribution properties. The pseudocode for this variant called *fast* ILHS is shown in Algorithm 4. This code is identical to ILHS except for lines 9–13. Instead of specifying a fixed duplication factor, we require a fixed number of candidates c as input parameter, which is crucial for obtaining quadratic run time. It also ensures that the same number of candidates is tested in each iteration. The duplication factor in iteration i is then calculated as $f_{\text{dup}} = \lceil c/(N-i) \rceil$.

Figure 4.3 compares ILHS and FILHS regarding run times and the diversity measures separation distance, average inverse distance, and discrepancy. The task in this case is to generate sets of 1000 points in five and ten dimensions. ILHS is tested with $f_{\text{dup}} = 1, \ldots, 5$ while FILHS uses $c = 2^i$, $i = 0, \ldots, 9$. (Note that $c = 1$ corresponds

[1] `http://cran.r-project.org/web/packages/lhs/`

Algorithm 4 Fast improved latin hypercube sampling

Input: number of points N, number of variables n, number of candidates c
Output: latin hypercube design with improved n-dimensional distribution

$\mathcal{P} \leftarrow \emptyset$
for all $j \in \{1, \dots, n\}$ **do**
  $A_j \leftarrow \{1, \dots, N\}$ // initialize sets of available bins
  $u_j \leftarrow$ random element of A_j
  $A_j \leftarrow A_j \setminus \{u_j\}$ // remove used bin
end for
$\mathcal{P} \leftarrow \mathcal{P} \cup \{(u_1, \dots, u_n)^\top\}$ // add first point to LHD
for all $i \in \{1, \dots, N-2\}$ **do**
  $f_{\text{dup}} \leftarrow \lceil c/(N - |\mathcal{P}|) \rceil$ // calculate variable duplication factor
  for all $j \in \{1, \dots, n\}$ **do**
    create multiset A'_j which contains each element of A_j f_{dup} times
  end for
  for all $k \in \{1, \dots, c\}$ **do**
    for all $j \in \{1, \dots, n\}$ **do**
      $u_j \leftarrow$ random element of A'_j // drawing without replacement
      $A'_j \leftarrow A'_j \setminus \{u_j\}$ // remove used bin from intermediate set
    end for
    $\boldsymbol{u}_k \leftarrow (u_1, \dots, u_n)^\top$ // $\boldsymbol{u}_k$ is the next candidate
    calculate $d_{\text{nn}}(\boldsymbol{u}_k, \mathcal{P})$ // min of all distances between $\boldsymbol{u}_k$ and points in $\mathcal{P}$
  end for
  $\boldsymbol{u}^* \leftarrow$ best of $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_k\}$ according to selection criterion
  for all $j \in \{1, \dots, n\}$ **do**
    $A_j \leftarrow A_j \setminus \{u_j^*\}$ // remove used bin
  end for
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\boldsymbol{u}^*\}$ // add best point to the LHD
end for
$\mathcal{P} \leftarrow \mathcal{P} \cup \{(v_1, \dots, v_n)^\top\}$, where v_j is the only possible choice left in A_j
return $\mathcal{P}$

to the conventional LHD.) Centered LHDs are created here, to avoid introducing additional noise into the designs. Both algorithms are implemented in Python 2.7, using the mathematical library NumPy 1.6. The run times are measured on an Intel Core i5-4670 with 3.4 GHz. As we can see, FILHS achieves the same or a slightly better performance in less time, while allowing much finer control through its parameter c. Please note that although the difference seems small for the tested $N = 1000$, many software packages can benefit from this simple improvement. And of course the gap between the algorithms grows steadily with increasing sample size.

Unfortunately, in high dimensions the optimization of the FILHS criterion seems to be counterproductive for discrepancy. After gaining more experience with discrepancy in the following sections, we will finally get back to this observation in

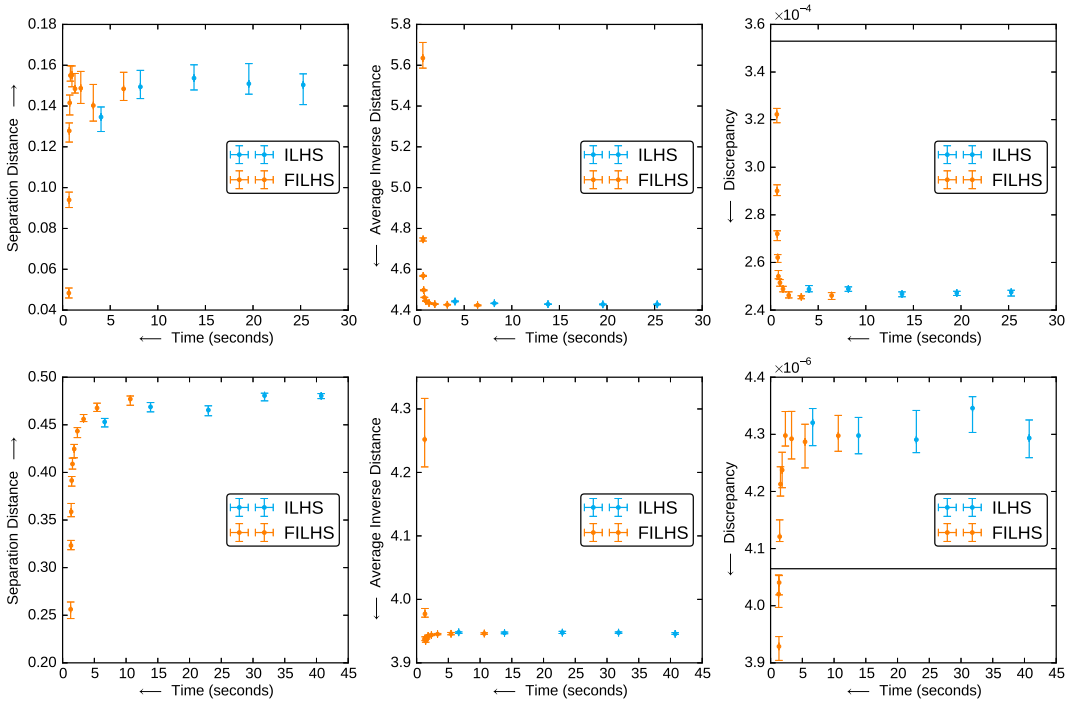


Figure 4.3: Performance of ILHS and FILHS. Each point denotes the median of 100 repeats, with errorbars depicting 95% confidence intervals for the median. The top row shows $n = 5$, the bottom row $n = 10$. In the right panels, the horizontal line indicates the expected discrepancy for random uniform designs.

Section 4.5.1, where it becomes clear that not only FILHS, but also the behavior of the indicator changes in high dimensions.

4.2 Quasirandom Sequences

The term *quasirandom sequence* describes an actually deterministic rule for the generation of uniformly distributed points. These sequences are popular tools for multidimensional numerical integration. Classical representatives are Halton, Faure, and Sobol' sequences. As this research area ist vast and only remotely connected to optimization, it is only touched briefly in the following. For the foundations of quasi-Monte Carlo methods, including the definitions of the mentioned quasirandom sequences, we refer to Niederreiter [108].

Due to the relation between discrepancy and the integration error (2.4), quasirandom sequences are of course designed to have low discrepancy. Formally, we speak of low discrepancy sequences when their discrepancy has a convergence order of $O(N^{-1}\log(N)^n)$ [108, p. 32]. However, Morokoff and Caflisch [103] estimate that this asymptotic property only holds after a transition phase, which might not ap-

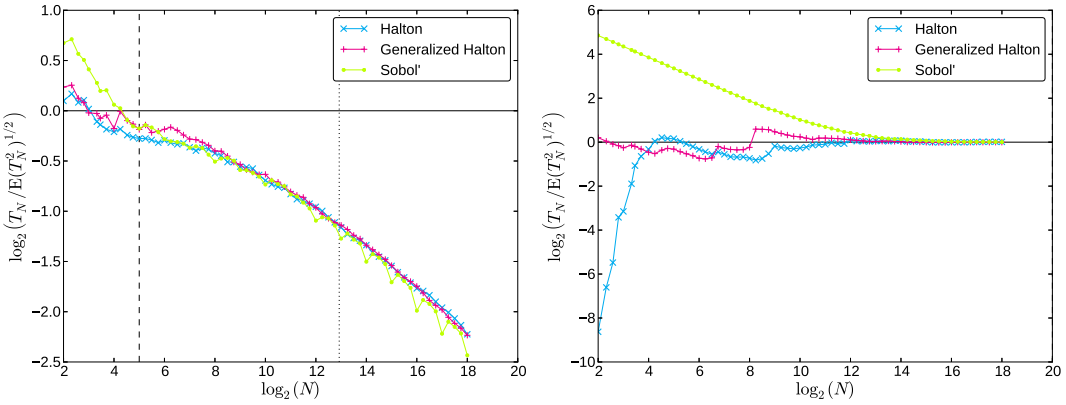
pear until $N \approx 6^n$. These results are in principle confirmed by their experiments, testing the practical usability of conventional Faure and Sobol' sequences in up to 16 dimensions with T_N as a measure. Jäckel [72, pp. 91–96] carries out more extensive experiments for T_N^* with up to 2^{16} points in 100 dimensions. In these tests, Sobol' sequences with "pattern-breaking" initializations could always provide at least the same, and in low dimensions a significantly lower L_2 star discrepancy than pseudo-random numbers. As a consequence of such findings, it generally became state-of-the-art to employ specially optimized quasirandom sequences. The most advanced initialization numbers for Sobol' sequences seem to be due to Joe and Kuo [65], who explicitly optimize the diversity of the two-dimensional projections.

Other quasirandom sequences that experienced a renaissance in recent years are the generalized (or scrambled) Halton sequences [43]. These arise from inserting certain permutations into the definition of the original Halton sequence. De Rainville et al. [32] optimize the two-dimensional projections of their sequence with an evolutionary algorithm. Figure 4.4 compares this optimized generalized Halton sequence[2] with the original Halton sequence and Joe and Kuo's Sobol' sequence[3] regarding discrepancy and the two new summary characteristics $\bar{d}_{\mathcal{B}}$ and $d_{\bar{\boldsymbol{c}}\boldsymbol{c}}$ in five and twenty dimensions. (The comparison has also been carried out for $n = 2$, 3, 10, 40.) The good news is that a reasonable estimation of $\boldsymbol{c}_{\mathcal{X}}$ seems to be possible with all sequences with as few as 2^{12} to 2^{14} points – even in forty dimensions. However, the Sobol' sequence seems to have a burn-in period where other indicator values considerably deviate from uniformity. In forty dimensions, an estimated number of 2^{23} points is necessary to even reach the discrepancy of random uniform points (not shown here). Results in [23] indicate that this contradiction to Jäckel's results stems from the different discrepancy formulation and not from the different Sobol' sequence.

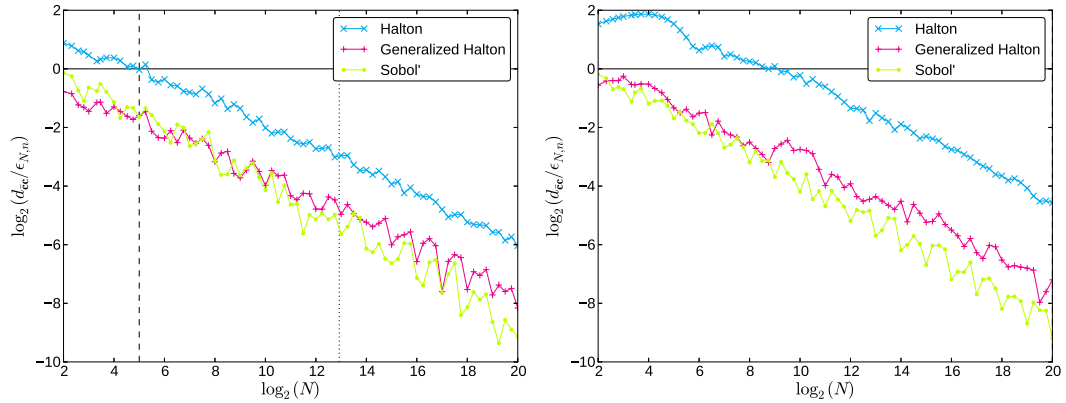
Morokoff and Caflisch are right with their predicted transition phase around $N = 6^n$, insofar that the discrepancy of the Sobol' sequence exhibits local optima at powers of two, starting approximately at this value. In high dimensions, these optima are also visible for $d_{\bar{\boldsymbol{c}}\boldsymbol{c}}$. The Halton sequence initially obtains a suspiciously low discrepancy in high dimensions. This may indicate that T_N has the same problems with reliability as T_N^* for $N < 2^n$, because simultaneously the $d_{\bar{\boldsymbol{c}}\boldsymbol{c}}$ values of the Halton sequence are bad. (Recall the discussion of discrepancy in Section 2.1.1, where we learned that T_N^* can give highly unintuitive results if $N < 2^n$ [91].) Finally, the results regarding $\bar{d}_{\mathcal{B}}$ are surprising. While the absolute values of $\bar{d}_{\mathcal{B}}$ exhibit a nice progression towards δ_n (not shown), the impression changes when we put the deviation from δ_n into relation with the expected deviation for random uniform point sets. In five dimensions, the results seem noisy, but acceptable. For $n = 20$, however, we see a wavy pattern with deep spikes (which is even more pronounced in forty dimensions), and also the convergence rate seems no better than the random uniform expectation. Especially the $\bar{d}_{\mathcal{B}}$ of the Sobol' sequence is too large for low N, which

[2] Available in the ghalton Python library at `https://github.com/fmder/ghalton`, Version 0.6.

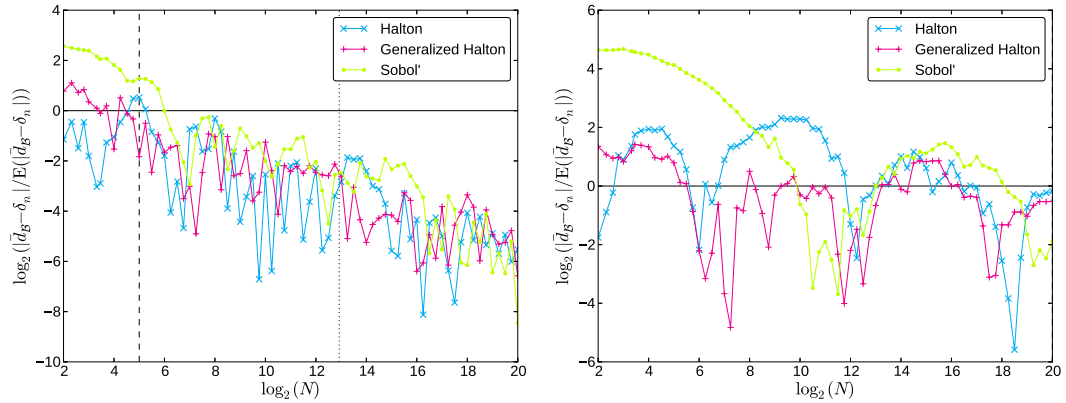
[3] Generated with the direction numbers "new-joe-kuo-6.21201" and software from `http://web.maths.unsw.edu.au/~fkuo/sobol/`, Version from 16 September 2010.



(a) Relative discrepancy.



(b) Relative Euclidean distance between $\mathcal{P}$'s center of mass and the centroid of the hypercube.



(c) Relative deviation from the expected distance to the boundary.

Figure 4.4: Quasirandom points evaluated with three summary characteristics for $n = 5$ (left) and $n = 20$ (right). The horizontal line indicates the reference value for each indicator. Dashed and dotted vertical lines mark 2^n and 6^n, respectively.

causes the bad performance in Figure 4.4c. In the following, only the generalized Halton sequence will be considered, because its performance seems to be the most stable considering all three indicators.

4.3 Subset Selection Methods

Discrete versions of our sampling problem may be obtained by sampling a larger number of points M in a random uniform fashion and then trying to find an optimal subset of N points. We will call this the subset selection approach to sampling. Several selection problems corresponding to diversity indicators in Section 2.1.1 have been defined and investigated in the literature. The selection problem corresponding to SD is known as the maximum diversity problem [77] or the maxisum diversity problem [49]. MD is represented by the maximin diversity problem [49], also called N-dispersion problem [41]. The minimax diversity problem or N-center problem [61] optimizes CR. The corresponding decision problems to all mentioned subset selection problems have been proved to be NP-hard [77, 41, 49, 51, 61]. Thus, various heuristics have been developed to obtain approximate solutions. Greedy construction heuristics exist for each problem with a run time of $O(MNn)$. They generally do forward selection, beginning with an initial point chosen at random from the set of candidates. For SD and MD, in each iteration the distances from the remaining candidates to the last chosen point are calculated. These new distances are used to calculate updated objective values for the candidates and the one with the best value is transferred to the selected points. Afterwards, the next iteration begins.

For CR, the algorithm begins in the same way but follows a slightly different approach. It is assumed that all points initially belong to one cluster, with an arbitrarily chosen representative point. In each iteration, a new cluster is opened for the point that has maximal distance to its representative. Then, all points which are not representatives yet become members of the new cluster, if their distance to the new representative is not larger than to their current one. Gonzalez [51] shows that this algorithm guarantees to find a solution for the minimax diversity problem that is not worse than twice the optimal value if the triangle inequality holds and that no better approximation bound can be found unless P = NP. Hochbaum and Shmoys [61] independently prove the same result for a more sophisticated heuristic with run time $O(M^2 n \log_2 M)$. If this higher effort is spent wisely, the better practical performance should be expected for this algorithm.

Erkut [41] considers the maximin diversity problem including fixed points, which is important for generating sequential designs. Regarding the distance calculations, the fixed points count as selected except that distances among fixed points are irrelevant. Erkut shows that the problem including fixed points is not more difficult than the original problem and a heuristic with run time $O(M^2 n \log_2 M)$ is proposed with a construction stage based on backward elimination and an improvement stage based on exchanges of single points between the selected and remaining points (pseudocode can be found in [95]). It is straightforward to extend at least the greedy heuristics

Algorithm 5 Partitioning a set $\mathcal{P}$ into N subsets

Input: points $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$
Output: clustering of $\mathcal{P}$

1: $\mathcal{C}_1 \leftarrow \mathcal{P}$
2: calculate $s_1 = \Delta_{p_1}$ and store $(s_1, p_1, \mathcal{C}_1)$ in an archive
3: $i \leftarrow 2$
4: **while** $i < N$ **do**
5: find $\mathcal{C}_j$ such that $s_j = \Delta_{p_j} = \max\{s_h \mid h = 1, \ldots, i-1\}$
6: part $\mathcal{C}_j$ into subsets $\mathcal{C}_{j_1}, \mathcal{C}_{j_2}$:
$\mathcal{C}_{j_1} \leftarrow \{\boldsymbol{x} \in \mathcal{C}_j \mid x_{p_j} \leq (u_{p_j} + \ell_{p_j})/2\}$
$\mathcal{C}_{j_2} \leftarrow \{\boldsymbol{x} \in \mathcal{C}_j \mid x_{p_j} > (u_{p_j} + \ell_{p_j})/2\}$
7: calculate $s_{j_1} = \Delta_{p_{j_1}}$ and $s_{j_2} = \Delta_{p_{j_2}}$
8: remove $(s_j, p_j, \mathcal{C}_j)$ from, and store $(s_{j_1}, p_{j_1}, \mathcal{C}_{j_1})$ and $(s_{j_2}, p_{j_2}, \mathcal{C}_{j_2})$ in archive
9: $i \leftarrow i + 1$
10: **end while**
11: **return** $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$

for the other problems to regard fixed points as well. However, all subset selection algorithms share the disadvantage of a relatively high memory consumption for storing the M points, as M should be probably chosen a multiple of N to obtain reasonable results. This also holds for the algorithm in the next section, although it has a much better average-case run time than all subset selection heuristics presented so far.

4.3.1 Part and Select Algorithm

Salomon et al. [125] proposed the part and select algorithm (PSA). This algorithm creates a partitioning of $M > N$ points in n dimensions into N clusters, beginning with a single cluster containing all points. PSA then repeatedly divides the cluster containing the greatest dissimilarity, which is defined as follows: for a cluster $\mathcal{C}_j = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\} \subset \mathbb{R}^n$, the minimal and maximal values for each dimension i, $\ell_i = \min\{x_{1,i}, \ldots, x_{k,i}\}$ and $u_i = \max\{x_{1,i}, \ldots, x_{k,i}\}$, have to be calculated. The difference between these two is $\Delta_i := u_i - \ell_i$, which is used to obtain $s_j := \max\{\Delta_i \mid i = 1, \ldots, n\}$, the largest spread in any dimension.

Intended or not, PSA is closely related to several algorithms for vector quantization, as the median-cut algorithm by Heckbert [59], the mean-split algorithm by Wu and Witten [159], and the method of Wan et al. [149]. These approaches are all divisive clustering algorithms using hyperboxes to describe the clusters. The main difference between them is the criterion determining where a cluster is split in two. In contrast to the other algorithms, PSA does not aim to minimize the quantization error, but simply to obtain a uniform subset of the original data. It was developed originally for subset selection in multiobjective optimization, but there are no special assumptions that prevent a universal application. The outline of the partitioning

part is given in Algorithm 5. It has a worst-case run time of $O(MnN)$ [125], using an array as a data structure to store the clusters. If we want to employ PSA to generate an initial design, we can choose $|\mathcal{P}| = M := cN$ for some constant c. The initial point set $\mathcal{P}$ would then be obtained by random uniform sampling, taking $O(Mn)$ time. Thus, the initialization does not increase the worst-case performance. Moreover, now that we have control over the input distribution, we can also make a run time analysis for the average case.

Theorem 2. *The average-case performance of Algorithm 5 on uniformly distributed point sets is* $O(Mn \log_2 N)$.

Proof. When a cluster $\mathcal{C}_j$ is divided into two parts, both new ones have exactly half the volume of the old one. Therefore, the expected number of points in each new cluster is $|\mathcal{C}_j|/2$, thanks to the uniform distribution we chose for the initial point set. We also have to assume that the cluster chosen as $\mathcal{C}_j$ in each iteration is the one containing the most points, because the number of points is proportional to the cluster's volume, again due to the uniform distribution [63, p. 60]. This leads to an estimated cost of

$$\begin{aligned} Mn \sum_{h=1}^{N} \frac{1}{2^{\lfloor \log_2 h \rfloor}} &\leq Mn \sum_{i=0}^{\lfloor \log_2 N \rfloor} 2^i \frac{1}{2^i} \\ &= Mn \sum_{i=0}^{\lfloor \log_2 N \rfloor} 1 \\ &\leq Mn(\log_2 N + 1) = O(Mn \log_2 N) \end{aligned} \tag{4.1}$$

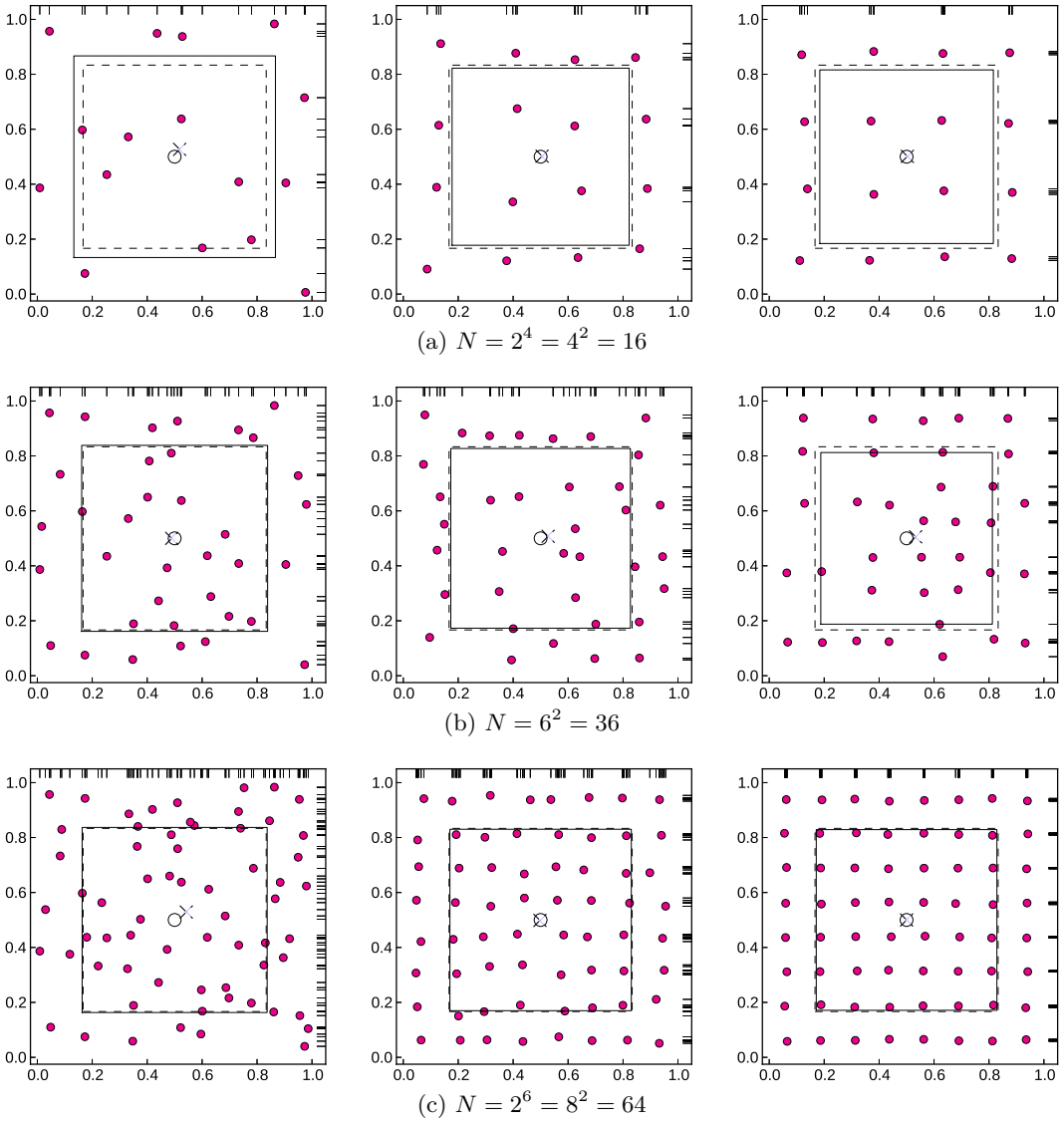
for N times partitioning the largest cluster and calculating the information for the new ones (steps 6 and 7). The series on the left hand side of inequality (4.1) is super-harmonic, which can be seen by unrolling the infinite variant

$$\sum_{h=1}^{\infty} \frac{1}{2^{\lfloor \log_2 h \rfloor}} = 1 + \underbrace{\frac{1}{2} + \frac{1}{2}}_{=2\frac{1}{2}} + \underbrace{\frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}}_{=4\frac{1}{4}} + \underbrace{\frac{1}{8} + \cdots + \frac{1}{8}}_{=8\frac{1}{8}} + \cdots \ .$$

Now it is easy to see that (4.1) holds, because the number of terms $1/2^i$ is greater or equal on the right hand side.

Using an array to store the clusters would cause total costs of $O(N^2)$ for N executions of steps 5 and 8. To obtain the bound $O(Mn \log_2 N)$ for the whole algorithm, a binary heap has to be used instead. It changes the cost of searching for $\mathcal{C}_j$ (step 5) from $O(N)$ to $O(1)$ and for the replacement operation in step 8 from $O(1)$ to $O(\log_2 N)$. Thus, the total cost for archiving is now only $O(N \log_2 N)$. □

Although it does not change the worst case run time, we assume it is generally advisable in practice to use the binary heap instead of the array data structure, because the run time of PSA is probably often low enough for it to make a relevant



(a) $N = 2^4 = 4^2 = 16$

(b) $N = 6^2 = 36$

(c) $N = 2^6 = 8^2 = 64$

Figure 4.5: Points generated by PSA with $M = 2N, 20N, 200N$ (columns left to right) and center point selection.

difference. Also note that $O(Mn \log_2 N)$ is identical to the run time of the mean-split algorithm [149].

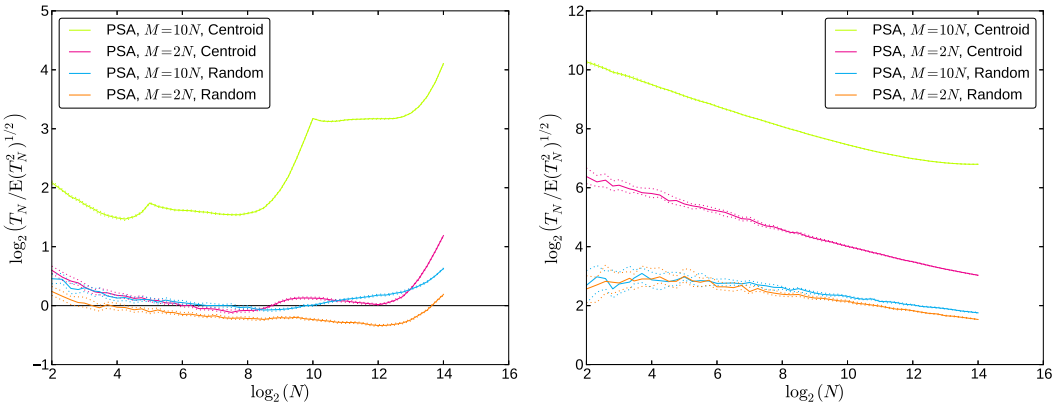
After the partitioning, a representative has to be chosen for each cluster. The authors in [125] propose *center point selection*, i. e., to choose for each cluster $\mathcal{C}_j$ the point of the discrete set that has the smallest Euclidean distance to the centroid of the hypercube defined by $\boldsymbol{u}_j$ and $\boldsymbol{\ell}_j$. The complexity of this approach is $O(Mn)$. Figure 4.5 illustrates in two dimensions, that for $M \to \infty$ the point set obtained by center point selection seems to tend to the Sukharev grid (cf. Figure 1.1f) for

some choices of N. Similar effects might also be observed in higher dimensions and therefore, it may be worthwhile to investigate some other approaches for choosing a representative.

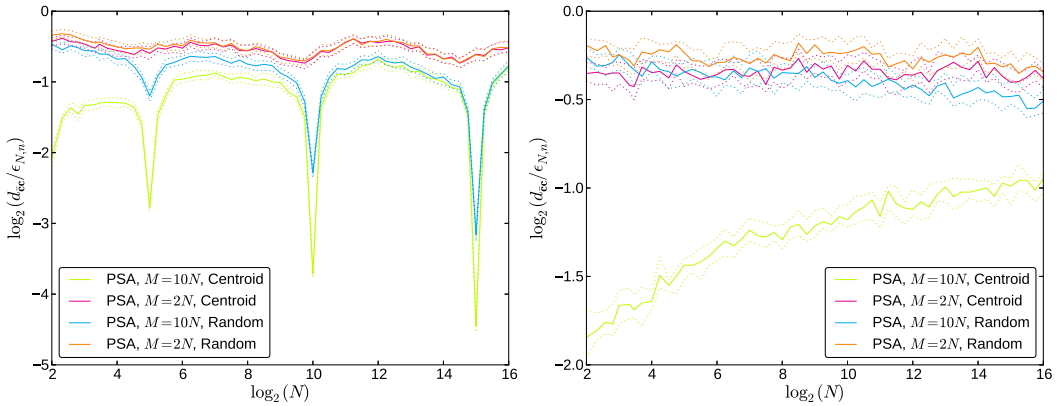
First, we will discuss approaches following the subset selection paradigm, with identical run time to center point selection. One simple idea would be determining the target point according to a uniform distribution in the cluster. This has the effect that a point's probability of being chosen is proportional to the size of its Voronoi cell. As points in sparsely populated regions of the cluster have larger Voronoi cells, they also have a higher probability of being selected. The opposite effect can be achieved by calculating the center of mass of the points in the cluster and using this as the target point. Both approaches presumably would create more variation in the target points, and therefore avoid or slow down convergence to the Sukharev grid. Instead of changing the target point, we could also change the selection method to not consider all points in the cluster. One possible approach would be k-ary tournament selection, which means that a random sample of size k is drawn from the points, and the best point according to some criterion is selected. For $k = |\mathcal{C}_j|$ the selection remains deterministic and for $k = 1$ the approach is identical to random selection, where each point has equal probability of being chosen. Thus, a changeover between the two extremes can be specified through k. In high dimensions, however, it must be expected that there are no points in the interior of the cluster, and therefore the chosen point will probably lie on the boundary of the cluster.

Luckily we do not have to stick to the subset selection scheme if our aim is just to generate an initial design. Then, we are free to use any of the target points mentioned above directly as the representative. This reduces the cost for the whole selection step to $O(Nn)$ if we use the centroid or a random point of the hypercube. These latter two selection approaches are visualized in Figure 4.6. Apparently, PSA is not very good at generating low-discrepancy point sets. While the figures in two and three dimensions look not as bad, the discrepancy performance deteriorates in higher dimensions (see Figure 4.6a). Additionally, the behavior seems to be quite dependent on the chosen number of points. At values of $N = 2^{ni}$, $i \in \mathbb{N}$, local extrema appear in the indicator values. The extrema are more pronounced for larger M and for the variant with the centroid of each cluster's hypercube as representative. In combination with Figure 4.5, this observation suggests that the convergence to the Sukharev grid also happens in higher dimensions, but not at every n-th power of the natural numbers. As a (perturbed) Sukharev grid is much easier generated directly, this setting of N should probably be avoided for PSA.

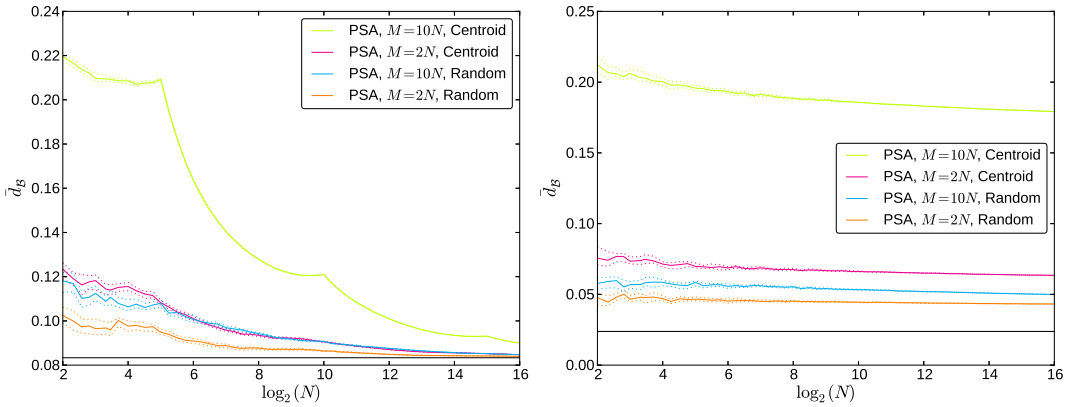
In summary, PSA may be an option to generate point sets with low covering radius due to the large distances to the boundary, but otherwise its properties seem to be rather bad in comparison to other methods. Perhaps it can serve as a construction heuristic for an initial point set, which is then refined by a more expensive method. Of the tested variants, the one with $M = 10N$ and random representative seems to be the most attractive, because of its relatively well-balanced behavior. This variant, which should be similar to stratified sampling [94], also yields the lowest discrepancy in low dimensions and the least irregularity in forty dimensions (not shown here).



(a) Relative discrepancy.



(b) Relative Euclidean distance between $\mathcal{P}$'s center of mass and the centroid of the hypercube.



(c) Mean distance to the boundary.

Figure 4.6: Behavior of PSA variants. The left column shows $n = 5$, the right one $n = 20$. The number of replications for each subfigure was chosen as $\lfloor 1600/n \rfloor$. Solid lines mark the median and dotted lines 95% confidence intervals.

4.4 Point Processes

Illian et al. [63, p. 23] state that "[p]oint processes are stochastic models of irregular point patterns". For example, a random uniform design with a fixed number of points is called a *binomial point process* in the language of spatial statistics [63, p. 59]. Point process statistics (a subcategory of spatial statistics) is used in a wide range of research areas, such as astronomy, forestry, biology, physics, or materials science, to analyze observed point patterns [63, pp. 5–17]. Of course, we are especially interested in *simulating* samples from certain point processes with favorable properties for experimental designs. In principle, also any of the previously considered algorithms could be interpreted as a model for some corresponding point process. Point processes have also been considered by Matérn [92], who is the inventor of the Matérn correlation function frequently used in Kriging. This serves as a further inspiration for us to consider point processes for generating space-filling designs.

Another area that focuses on the simulation of point patterns is computer graphics. There, the research is mainly concerned with fast algorithms and aesthetics. Uniformly distributed point sets are for example required for procedural texture generation or for antialiasing by supersampling [111, pp. 280–367]. Thus, usually only the case $n = 2$ is considered. Generating the point sets is also known as Poisson disk sampling there, because it is imagined that each point denotes the center of a disk with radius r. Interestingly, the requirements of applications in computer graphics seem quite similar to our own ones. The disks shall be densely packed but non-overlapping, which means that r cannot be larger than half the minimal pairwise distance in the set. Thus, many considered algorithms there simulate hard-core processes, which just means that r is fixed in advance and the points are located so that disks are non-intersecting. Perhaps the simplest algorithm of this kind is random sequential adsorption (RSA), which begins with $\mathcal{P} = \emptyset$ and sequentially adds random candidates $\boldsymbol{x}_i$ if $d_{\text{nn}}(\boldsymbol{x}_i, \mathcal{P}) \geq 2r$. The algorithm may be terminated after a fixed number of iterations or when no further points can be added. Consequentially, the number of points N is a random variable. We, however, are interested in specifying N in advance. RSA is also known as dart throwing [78] or simple sequential inhibition [63, p. 393]. Lagae and Dutré [78] use spectral analysis and the *relative radius* to evaluate the point sets. The relative radius $\rho = r/r_{\max}$ relates the absolute radius r to the maximum possible disk radius of N disks in the plane,

$$r_{\max} = \sqrt{\frac{1}{2\sqrt{3}N}} \,,$$

which is achieved by the hexagonal lattice. Lagae and Dutré [78] comment that "[p]oisson disk distributions should have a relative radius that is large ($\rho \geq 0.65$), but not too large ($\rho \leq 0.85$), because regular configurations must be avoided." So, also in this area, a compromise between uniformity and irregularity is sought.

One of the methods mentioned in [78] for the generation of poisson disk distributions is Lloyd's algorithm [83], which is the most commonly used variant of k-means

Algorithm 6 MacQueen's algorithm

Input: initial points $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
Output: generators of an approximately centroidal Voronoi tesselation

1: set weights $w_i = 1$, $i = 1, \ldots, N$
2: **repeat**
3: $\quad \boldsymbol{r} \leftarrow$ random uniform point in $\mathcal{X}$
4: $\quad \boldsymbol{x}_{i^*} \leftarrow \mathrm{nn}(\boldsymbol{r}, \mathcal{P})$ // determine nearest neighbor of $\boldsymbol{r}$ in $\mathcal{P}$
5: $\quad \boldsymbol{x}_{i^*} \leftarrow (w_{i^*}\boldsymbol{x}_{i^*} + \boldsymbol{r})/(w_{i^*} + 1)$ // update $\boldsymbol{x}_{i^*} \in \mathcal{P}$
6: $\quad w_{i^*} \leftarrow w_{i^*} + 1$ // update corresponding weight
7: **until** termination
8: **return** $\mathcal{P}$

clustering. As is well-known, it requires a predefined number of clusters, which is an advantage in our case. Lagae and Dutré mention that the algorithm creates the highest relative radius among the ones they tested. This is not surprising, as the algorithm converges to a centroidal Voronoi tesselation (CVT) of the space [37]. The same approach is used by Suzuki and Drezner [137] as a heuristic for the minimax distance design problem (or, in their terms, the N-center location problem in an area). This illustrates that the CVT property is quite interesting, because it is associated with low covering radii, and therefore also minimax distance designs. Unfortunately, if Lloyd's method is applied to continuous spaces, it requires the explicit construction of Voronoi cells, which is expensive [37]. However, there is a much simpler algorithm called random k-means by MacQueen [89], which will be discussed in the next section.

4.4.1 MacQueen's Algorithm

MacQueen's algorithm [89] is a very simple, probabilistic algorithm to generate CVTs. Its pseudocode is shown in Algorithm 6. The points of an initial design $\mathcal{P}$ are regarded as generators of a Voronoi tesselation. $\mathcal{P}$ is then improved sequentially, by updating one point per iteration as the weighted mean of itself and a random member of its Voronoi cell. Although this procedure can be proven to converge almost surely to a centroidal Voronoi tesselation [89], its convergence seems to be quite slow in practice [71]. As an alternative, Ju et al. [71] propose a parallel algorithm that can be seen as a generalization of MacQueen's and Lloyd's algorithm. It, too, avoids the explicit construction of Voronoi tesselations by Monte Carlo sampling and is therefore also probabilistic. However, none of the algorithms discussed in [89, 83, 71] is able to incorporate existing points into the calculations in a straightforward manner. Such an algorithm is presented in the next section.

4.4.2 Maximin Reconstruction

The basic idea in this section is to imitate CVTs by maximizing the separation distance under additional constraints for the boundary regions. The proposed algorithm

Algorithm 7 Maximin reconstruction algorithm

Input: initial points $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, distance criterion $d(\cdot)$
Output: uniformly distributed points

$A \leftarrow \{1, \ldots, N\}$ // indices of candidates for replacement
$i \leftarrow$ random element of A // choose arbitrary candidate
$A \leftarrow A \setminus \{i\}$ // remove used index
repeat
  $\boldsymbol{y} \leftarrow$ random point in $\mathcal{X}$ // sample potential substitute
  if $d(\boldsymbol{y}) \geq d(\boldsymbol{x}_i)$ **then** // if improvement found
    $\boldsymbol{x}_i \leftarrow \boldsymbol{y}$ // replace the point in $\mathcal{P}$
    $A \leftarrow \{1, \ldots, N\} \setminus \{i\}$ // distances have changed, reset the available indices
  else if $A \neq \emptyset$ **then** // try to find point that is easier to replace
    $i' \leftarrow$ random element of A
    $A \leftarrow A \setminus \{i'\}$
    if $d(\boldsymbol{x}_{i'}) \leq d(\boldsymbol{x}_i)$ **then** // if $\boldsymbol{x}_{i'}$ is easier to replace
      $i \leftarrow i'$ // use it as new candidate for replacement
    end if
  end if
until termination
return $\mathcal{P}$

is basically a variation of the "reconstruction algorithm" [63, pp. 407-417]. The general idea of the reconstruction approach is to imitate a measured point process by minimizing the deviation of summary characteristics between the measured and the simulated point process. The measured points may also be a part of the final point set. In this case, we seek a set of ("simulated") points that augments the existing ("measured") sample in the most plausible way. Note that this task is generally an optimization problem, which could be tackled with various algorithms. A common approach, however, is to use a local search that exchanges one point per iteration and accepts the modification if the objective is improved. The references [123, 118] contain some more pointers to such exchange algorithms for experimental designs.

The idea is adopted and slightly modified in the following. The whole pseudocode is outlined in Algorithm 7. As in Section 4.4.1, the number of points is fixed in advance. The algorithm then iteratively tries to replace one of the current points with a randomly chosen one. Instead of imitating an existing point set, we want to simply maximize uniformity. Thus, potentially existing fixed points are not taken as a reference set, but assumed to belong to the whole set, whose uniformity is to be maximized. Every improvement of the separation distance is accepted. In case no improvement is found, there is another extension of the basic algorithm. Instead of choosing the candidate for replacement randomly, we compare the current candidate with another one of the remaining, untested points in $\mathcal{P}$. The one with the smaller nearest-neighbor distance is chosen as the candidate for replacement in the next iteration. If the sequence of failed attempts is long enough, we will eventually find

the point in $\mathcal{P}$ with the (currently) minimal nearest-neighbor distance, and replace it in one of the next iterations. Thus, we have a true maximin approach, and therefore the algorithm shall be called maximin reconstruction (MmR). The attractiveness of the algorithm lies in its relatively economical use of distance computations without the need for a sophisticated data structure. Note, however, that we do not intend to produce the exact optimum, because we want to retain some irregularity of the point set. The irregularity hopefully helps to improve the diversity of the low-dimensional projections. If we disregarded these aspects, we would approach the topic of optimal sphere packing, for which certainly better local optimization algorithms could be devised [4].

We still have to discuss the concrete definition of the distance criterion $d(\cdot)$ used in Algorithm 7. In its most basic form, $d(\boldsymbol{x}) = d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{Q})$, where $\mathcal{Q} = \mathcal{P}$, or, if additionally a set of fixed points $\mathcal{A}$ has to be considered, $\mathcal{Q} = \mathcal{P} \cup \mathcal{A}$. Unfortunately, maximin approaches are known for a drift towards the boundary [66], which means that the point density at the boundary is higher than in the interior [63, p. 145]. One possible remedy is to use periodic edge-correction (PEC) [63, p. 184]. For this purpose, an L_p torus distance

$$d_{\mathrm{torus}}(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{n} \min\{|x_i - y_i|, u_i - \ell_i - |x_i - y_i|\}^p \right)^{1/p}$$

is assumed as the internal distance in d_{nn}. In this case, the resulting sample is expected to be uniform everywhere, because edge effects are eliminated. To imitate CVTs, which have a *lower* density in boundary regions, we choose $d(\boldsymbol{x}) = \min\{d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{Q}), 2d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B}) \cdot \sqrt[p]{n}\}$. The term $2d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$ is motivated by a hypothesized mirror point on the other side of the closest boundary, corresponding to reflection edge-correction (REC) [63, p. 184]. However, we slightly modify the conventional REC because of the following argumentation: In case of a Sukharev grid $\mathcal{P}$ (cf. Figure 1.1f), which has optimal covering radius for $p = \infty$ [79, pp. 202–203], it holds that

$$\mathrm{CR}(\mathcal{P}) = \min\{d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B}) \mid \boldsymbol{x} \in \mathcal{P}\} = \frac{1}{2} \mathrm{MD}(\mathcal{P}) \,.$$

The orthogonal mirroring of conventional REC perfectly fits into this case. If we now keep the grid $\mathcal{P}$ fixed and reduce p in a thought experiment, it is obvious that CR increases while MD stays the same. The increase of CR is caused (among others) by points in the corners of $\mathcal{X}$, which are the first to be uncovered because their distance to some point $\boldsymbol{x} \in \mathcal{P}$ is larger than $d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$ for $p < \infty$. Thus, our assumption is that reflection edge correction should rather be based on diagonally mirrored points than orthogonally mirrored ones, to obtain a less extreme behavior. So, we multiply $2d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$ by $\sqrt[p]{n} = \|\mathbf{1}\|_p \geq 1$, to obtain the distance to a diagonally mirrored point, which is also illustrated in Figure 4.7. In reality also the arrangement of the points plays a role, so this rule is only a very rough guideline. However, two observations can be made in this situation:

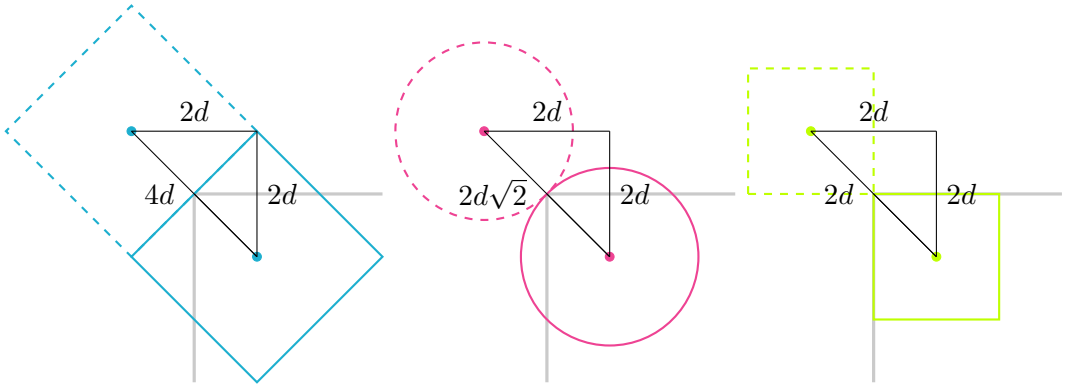


Figure 4.7: An assumed diagonal mirroring of points is used to influence the distances of points close to the boundary. From left to right, the cases $p = 1$, 2, ∞ are considered. The distance to the closest boundary is assumed to be d. The gray lines indicate a (hypothetical) corner of the search space.

- The smaller p is, the larger is the distance between $\boldsymbol{x}$ and the closest corner in relation to $d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$. (Also recall that in connection with this, the triangle inequality is violated for $p < 1$.)
- The larger the distance to the mirrored point, the weaker is the selection pressure at the boundary.

As a consequence, the influence of this correction factor is the strongest for $p = \infty$ and decreases together with p. The lower p is, the easier it is for the point set to approach the boundary.

MmR's asymptotic run time is identical to that of MacQueen's algorithm: Suppose we want to generate N new points, while there are already $|\mathcal{A}|$ existing points. Then the algorithm needs $N + |\mathcal{A}|$ distance computations in successful iterations and at most $2(N + |\mathcal{A}|)$ in unsuccessful iterations. Thus, the overall run time is $O(t(N + |\mathcal{A}|)n)$ if t is the number of iterations. If no existing points are present ($|\mathcal{A}| = 0$), this is just the run time of MacQueen's algorithm.

Experimental Analysis of MmR

PEC can not only be applied to MmR, but also MacQueen's algorithm can be adapted to a torus straightforwardly. To do this, the nearest neighbor cluster center to the current random point has to be identified according to torus distance. Then, the random point is replaced by its "virtual" counterpart that obtained the minimal torus distance. As this virtual point may be located outside of $\mathcal{X}$, also the new $\boldsymbol{x}_{i^*}$ may be "pulled out" by the update operation. Mapping $\boldsymbol{x}_{i^*}$ back into $\mathcal{X}$ by a modulo operation then is all that is necessary to complete the edge-corrected iteration of MacQueen's algorithm. Periodic edge correction is also mentioned in other sources,

e. g. for Lloyd's algorithm in [78], for CVTs in [124], or for image sampling in [111, pp. 334–338]. Finally, also FILHS can employ torus distances in its optimization criterion, without any further changes to the algorithm.

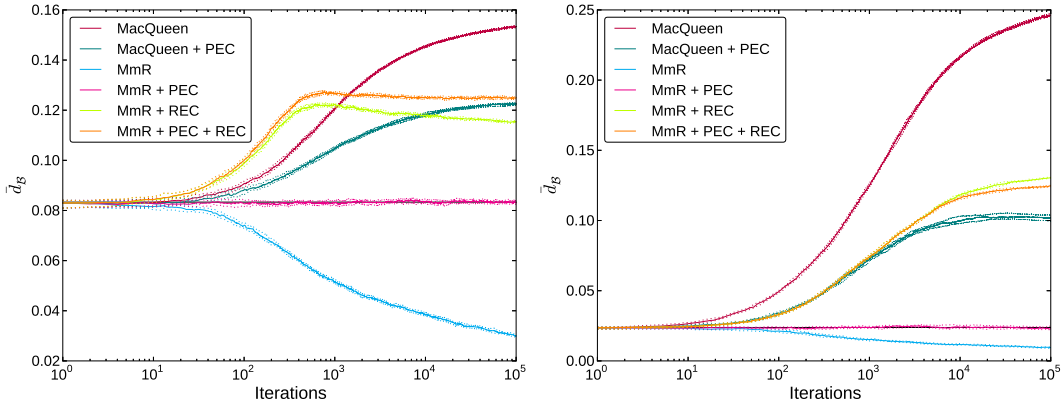
To check if our edge corrections work as intended, we make some experiments comparing MmR (with $p = 2$) and MacQueen's method. Figure 4.8a shows the development of $\bar{d}_{\mathcal{B}}$ in dependency of the number of iterations used for optimizing the point sets. As expected, pure MmR obtains a very low $\bar{d}_{\mathcal{B}}$ while its combination with PEC leads to an exactly uniform distribution. The coupling of MmR and our variant of REC yields a higher $\bar{d}_{\mathcal{B}}$ than for uniform point sets (as intended), but the value it converges to is generally not the same as that of MacQueen's method. Normally, the $\bar{d}_{\mathcal{B}}$ for PEC and REC together should be higher than for each of them alone. This behavior, however, can only be observed in lower dimensions. The biggest surprise is that MacQueen's algorithm with PEC does not create uniform point sets, but ones with a similar $\bar{d}_{\mathcal{B}}$ as MmR with REC.

The irregularity of MacQueen's method seems to be quite dependent on the dimension (see Figure 4.8b). Apart from that, MmR with PEC produces the most irregular point sets. The differences between the other variants are negligible in low dimensions. For twenty dimensions and higher, MmR without any edge correction has the most regular sets.

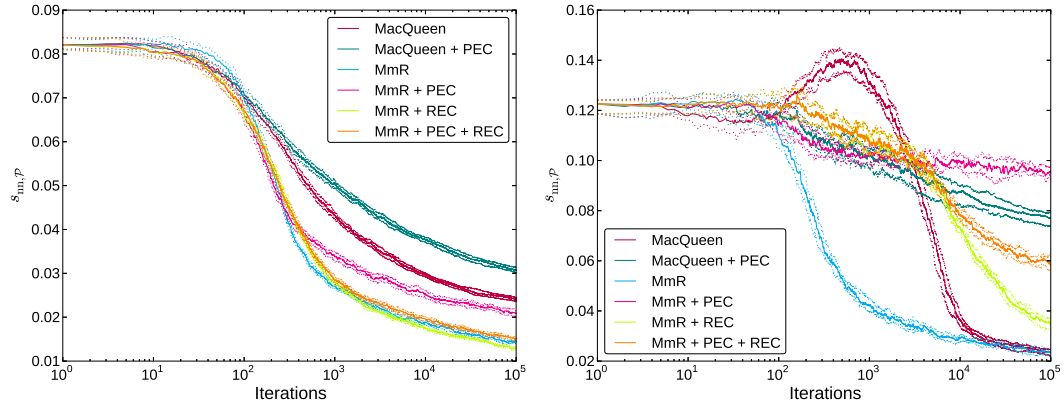
Discrepancy favors MmR without edge correction in high dimensions (see Figure 4.8c), which is surprising because it is a pure maximin design and not perfectly uniform. This is another indication that discrepancy does not work with small N. In five dimensions and lower, however, MmR with PEC obtains the lowest discrepancy, as desired.

Figure 4.9 shows the influence of p on the point sets created by MmR (without edge correction) in two dimensions. If we use a low value of p, the pattern of points is rotated against the axes of the hypercube (strings of points are perceived to run along diagonals, Figure 4.9a). Euclidean distance ($p = 2$), on the other hand, does not encourage this behavior (Figure 4.9b). In this case, the edges have more influence and cause a stronger impression of axis-alignment. As a consequence, the lower-dimensional projections are less diverse, which can also be measured by av_1. This is done in Figure 4.10, where a lower p gives a clear advantage in terms of the AID value averaged over all two-dimensional projections. Simultaneously, the irregularity (3.1) – measured with Euclidean distance – behaves anti-proportional to av_1. This may or may not be desired, depending on the application.

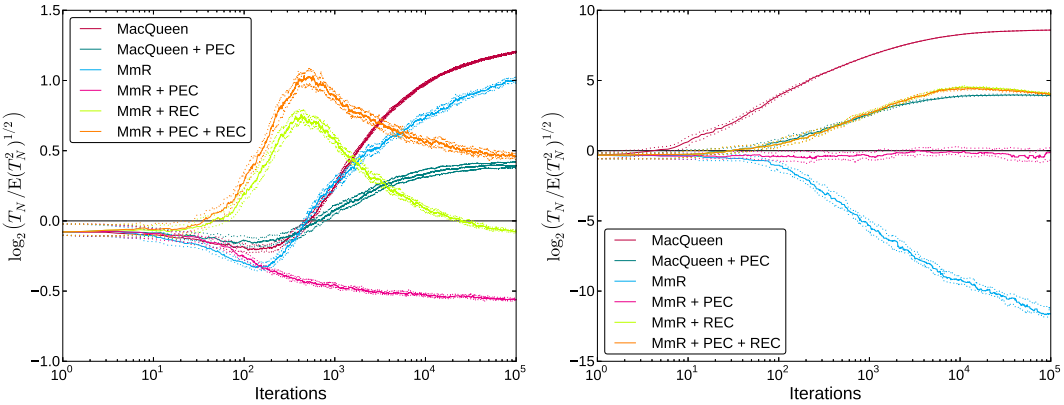
Figure 4.9c illustrates the sequential aspect of MmR. The lime green dots represent 150 existing points (with random uniform distribution). MmR is able to uniformly fill the gaps in the existing pattern. Here, also both edge corrections have been enabled to show how the distance to the boundary is increased in comparison to Figure 4.9b. Again, if we chose a lower p value, the pattern of the filled in points would look less axis-aligned (not shown here).



(a) Mean distance to the boundary.

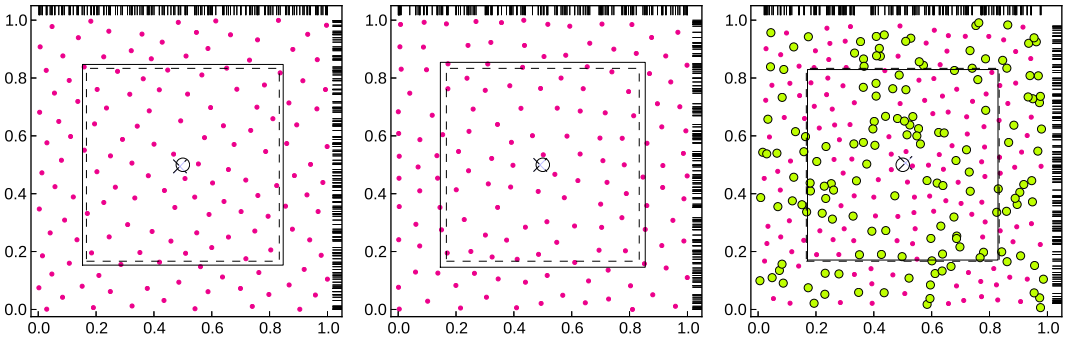


(b) Irregularity.



(c) Relative discrepancy.

Figure 4.8: Selected indicators during optimization of point sets with $N = 100$. The left column shows $n = 5$, the right one $n = 20$. Number of replications is $\lfloor 800/n \rfloor$. Solid lines mark the median, dotted lines 95% confidence intervals.



(a) $p = 0.5$ (b) $p = 2$ (c) +PEC, +REC, $p = 2$

Figure 4.9: Examples of MmR with $N = 150$. The algorithm was run for $10^4 N$ iterations to create the point sets. Fixed points are marked in lime green.

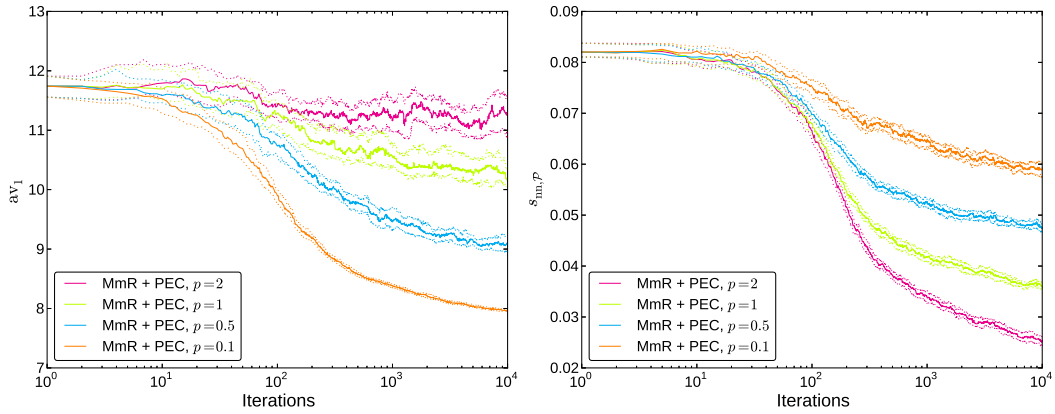


Figure 4.10: The influence of p on the average projection criterion and irregularity of $N = 100$ points in $[0, 1]^5$ created by MmR with periodic edge correction.

4.5 Comparison of Sampling Methods

Table 4.1 summarizes some properties of the mentioned sampling algorithms. Most of the considered algorithms are stochastic. In spite of their name, quasirandom methods were originally designed and used as deterministic methods. Nowadays, randomizations do exist [88], but they may influence the performance of the sequences. The attribute "sequential" means that the algorithm is designed to take arbitrary existing points into account for obtaining a better result than random uniform sampling. Quasirandom sequences do this automatically by continuing "their own pattern" with low discrepancy, but not arbitrary ones. While in theory the same could be said about grids, they are terribly inflexible for practical use in high dimensions (see Figure 3.1). In the strict sense, of the methods we considered, only most subset selection approaches [41] and MmR have this capability. The property "online" means that the algorithm does not need to keep all N points in memory while generating the point set. Instead, online algorithms only need some state in-

Table 4.1: Properties of sampling algorithms.

Algorithm	Run time	Stochastic	Sequential	Online
Random uniform	$O(Nn)$	✔	✘	✔
Grid	$O(Nn)$	✘	(✘)	✔
Quasirandom	$O(Nn)$	(✘)	(✘)	✔
Latin Hypercube				
LHS	$O(Nn)$	✔	✘	✘
ILHS	$O(N^3n)$	✔	✘	✘
FILHS	$O(N^2n)$	✔	✘	✘
Subset selection				
Erkut	$O(M^2n \log_2 M)$	✔	✔	✘
Gonzalez	$O(MNn)$	✔	✔	✘
Hochbaum & Shmoys	$O(M^2n \log_2 M)$	✔	✔	✘
PSA	$O(Mn \log_2 N)$	✔	✘	✘
Point Process				
MacQueen	$O(tNn)$	✔	✘	✘
MmR	$O(t(N + \lvert\mathcal{A}\rvert)n)$	✔	✔	✘

formation and their generating rule to create the next point. However, the budgets considered in this work are not large enough for this property to become important.

The classification in Table 4.1 should not be seen as the only alternative. For example, a category for clustering was omitted, although several algorithms have roots in this domain. In the following experiment, some of the more promising sampling algorithms are compared regarding their performance in multimodal optimization.

4.5.1 Experiment on Sampling Algorithms

Research Question Which sampling algorithm yields the best performance in multimodal optimization?

Pre-experimental Planning The discussions of the sampling algorithms in Chapter 4 contain most of the preliminary investigations. Of these algorithms (and variants), a subset is selected for inclusion in this experiment. Based on Figure 4.3, the number of candidates for FILHS is set to $c = 100$. The number of iterations for MmR is set to $t = 100N$. PSA is only granted an initial set size of $M = 10N$, because these points all have to be stored in memory simultaneously. A preliminary experiment was conducted to estimate the influence of PEC on FILHS and maximin LHS. (The latter were also created by Algorithm 4, but with a maximin criterion in line 21.) In both cases, the randomized approach of McKay et al. [94] was used to scale the designs to $[0, 1]^n$, to avoid any bias towards or away from the boundary. Surprisingly, these (approximate) maximin LHS do not seem to behave as the ones examined in Figure 4.2: The distance to the boundary is constantly higher than expected, as can

be seen in Figure 4.11b. The behavior of FILHS seems to be dimension-dependent, as $\bar{d}_{\mathcal{B}}$ is lower than δ_n in five dimensions and higher in twenty. To avoid unforseen effects in this regard, using PEC is again an option, because it ensures a good uniformity and eliminates the differences between the two heuristics. However, as we will also test several other truly uniform designs, only FILHS without edge correction is included in the main experiment, to keep the effort endurable and to have more diversity in the setup.

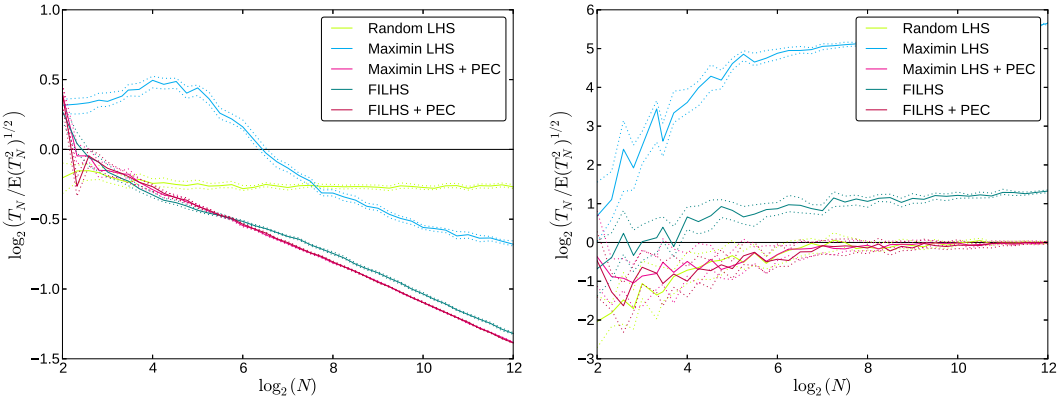
Task We are especially interested in the worst-case performance of sampling algorithms and their ability to yield starting points for potentially following local searches. Therefore, point sets are evaluated by quality indicators that reward exploration as in Figure 2.3a. No local search is actually performed and no subset selection step is included in the evaluation.

Several quality indicators are possible candidates for the assessment. The basin ratio is directly connected to our argumentation regarding numerical integration in Section 1.2 and the observation that it should be easy to find an optimum once we have a starting point in its region of attraction [143, p. 8]. For peak distance and peak inaccuracy we know theoretical worst-case bounds that depend on the covering radius (Propositions 1 and 2). Therefore it would be interesting to know if they give different evaluations than BR. As a fourth indicator, basin inaccuracy is evaluated. Using AHD would not make sense in this setting, as it does not solely reward exploration.

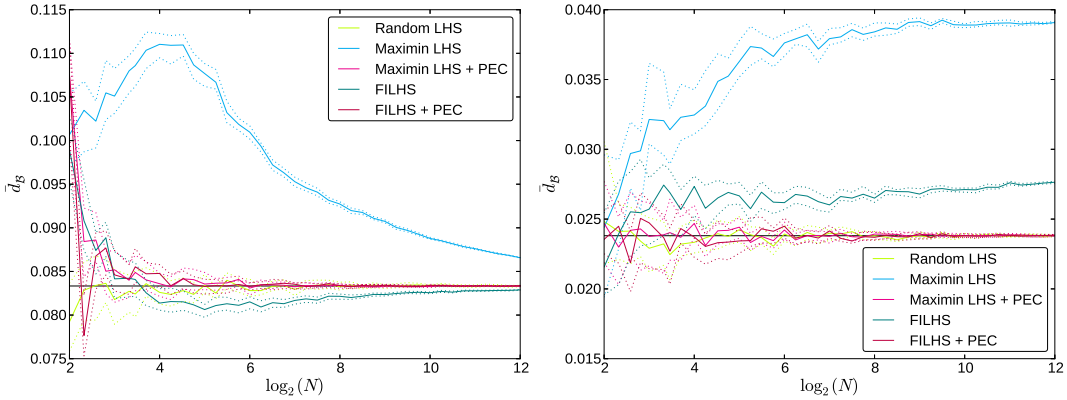
Pairwise (two-sided) sign tests are used to identify those algorithms which are non-dominated in the sense that no other algorithm is significantly better according to this statistical test. This approach, inspired by Bischl et al. [19], is very conservative as the sign test makes very few assumptions about the distribution of the data. It does, however, require paired samples, which we support by using common random numbers [93]. To account for the multiple comparisons, we are also being very conservative and use Bonferroni correction. Thus, the used significance level is $\frac{\alpha}{c(c-1)/2}$, with $\alpha = 0.05$ and $c = 12$ being the number of contestants. A consensus is achieved by counting how many algorithms are better than another one regarding this test. As a second performance measure, a ranking of the sampling algorithms is established for each block of the experiment and an algorithm's mean rank over all blocks is calculated (separately for each n). This approach belongs to the class RT-2 in Conover and Iman's survey of rank transformations [26].

Neither of these two consensus methods is *independent of irrelevant alternatives*. This means that removing some bad algorithms from the competition could alter the ranking of better ones [98, 97]. However, independence of irrelevant alternatives is generally impossible to attain without sacrificing other properties [98], so we will simply accept this deficiency.

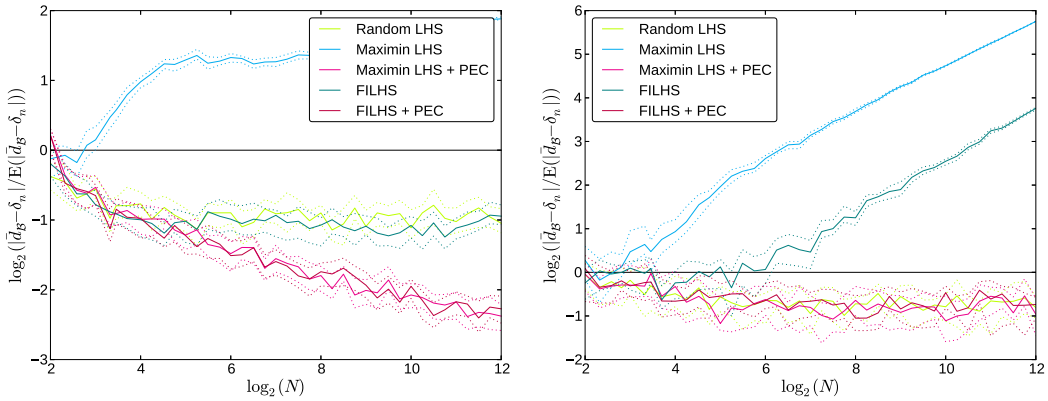
Setup The high-level experimental factors are listed in Table 4.2. For MmR, actually eight variants are tested, resulting from the combination of all four boundary treatments with $p = 1, 2$. For the other sampling algorithms, only one representative is chosen, based on the preliminary investigations. These are the generalized Halton



(a) Relative discrepancy.



(b) Mean distance to the boundary.



(c) Relative deviation from the expected distance to the boundary.

Figure 4.11: Indicator values for several LHDs. The left column shows $n = 5$, the right one $n = 20$. The number of replications was $\lfloor 1600/n \rfloor$. Solid lines mark the median of these replications and dotted lines 95% confidence intervals.

Table 4.2: High-level factors for the experiment in Section 4.5.

Factor	Type	Symbol	Levels
Problem topology	non-observable		{random, funnel}
Number of local optima	non-observable	ν	$\{5, 20, 100, 500\}$
Number of variables	observable	n	$\{2, 3, 5, 10, 20, 40\}$
Budget	observable	N_f	$\{10^1 n, 10^2 n\}$
Algorithm	control		{SRS, GHalton, PSA, FILHS, MmR}

sequence (now with random initialization), PSA with a random representative for each cluster, FILHS with randomized transformation to $\mathcal{X}$, and finally the simple random uniform sampling (SRS) as a baseline method. Together with the environmental factors, a full factorial experiment is generated. We assume that the maximal budget is determined by the application, just like the number of variables, and thus counts as an "observable" factor. Each configuration is replicated 100 times with different random numbers.

Results Figure 4.12 shows some example Box plots for peak distance. In these and also in all following Box plots, the notched areas denote the median value of the configurations with its 95% confidence interval. For PD it is undoubted that the indicator should be minimized, which is indicated by an arrow pointing downwards. If the case is not so clear, the arrow may be omitted in other Box plots. Mean values are indicated by a magenta asterisk with eight spokes and the blue points denote outliers.

Figure 4.13 summarizes the algorithms' behavior regarding the problem-independent measures irregularity and distance to the boundary. Finally, Table 4.3 gives an overview of the performance results obtained for the four multimodal quality indicators in this experiment.

Observations All indicator values become worse by trend with increasing dimension. For example, it can be observed that even if $N_f \gg \nu$, the basin ratio can be very low in high dimensions (not shown here). The results for peak distance are the clearest and can be illustrated easily by Box plots (see Figure 4.12). In high dimensions, PD is strongly correlated with the approximation set's distance to the boundary, which can be seen by comparing Figures 4.12 and 4.13. This effect is independent of the number of optima ν and the number of points N_f. It can also be observed for PI, but not for BR and BI.

SRS consistently produces the most irregular point sets. Also GHalton and PSA are always quite irregular, while FILHS cannot be classified definitely. In low dimensions, irregularity of the MmR variants is largely determined by p. In high dimensions, MmR variants without edge correction obtain the lowest irregularity,

Table 4.3: Aggregated results of the sampling experiment. Column "D" denotes the number of pairwise tests where another algorithm was significantly better, "R" is the algorithm's mean rank. The best results are printed in bold.

n	Indicator	SRS		GH		PSA		FILHS		MmR $p=1$		MmR $p=2$		MmR +PEC $p=1$		MmR +PEC $p=2$		MmR +REC $p=1$		MmR +REC $p=2$		MmR +PEC +REC $p=1$		MmR +PEC +REC $p=2$	
		D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R
2	PD	11	8.8	10	8.1	9	4.4	4	7.0	7	10.0	7	10.6	4	7.5	4	7.5	**0**	3.7	**0**	**3.3**	2	3.8	**0**	3.5
2	PI	11	7.9	8	7.7	7	4.3	2	6.5	7	10.0	7	10.4	3	7.4	1	7.4	**0**	4.3	**0**	**4.0**	**0**	4.1	**0**	4.0
2	BR	11	7.3	9	6.7	9	6.9	**0**	6.8	**0**	6.2	**0**	6.3	**0**	6.3	**0**	6.3	**0**	**6.2**	**0**	6.2	**0**	6.4	**0**	6.5
2	BI	11	7.5	9	6.9	9	6.5	3	6.8	4	7.4	6	7.6	1	6.4	1	6.4	**0**	**5.6**	**0**	5.6	**0**	5.6	**0**	5.8
3	PD	11	10.3	9	8.7	10	6.4	4	6.0	7	7.7	7	8.0	6	6.6	4	6.4	1	4.5	**0**	**4.4**	2	4.7	**0**	4.4
3	PI	10	8.7	7	7.5	6	5.7	3	6.2	9	7.9	9	8.4	4	6.7	4	6.6	**0**	5.1	**0**	5.1	**0**	5.2	**0**	**5.0**
3	BR	11	8.5	9	7.1	9	7.3	4	6.3	**0**	**5.8**	**0**	5.9	4	6.0	2	6.2	**0**	6.2	**0**	6.2	4	6.3	**0**	6.3
3	BI	11	9.1	9	7.3	9	7.4	4	6.2	7	6.2	7	6.4	4	5.9	4	6.1	**0**	**5.7**	**0**	5.9	**0**	6.0	**0**	5.9
5	PD	10	10.5	8	8.2	5	8.5	5	5.9	9	7.5	10	7.5	4	6.5	4	5.9	1	4.5	**0**	**4.1**	3	4.6	1	4.2
5	PI	9	8.4	7	7.2	4	7.0	4	6.0	10	7.7	11	7.8	5	6.5	4	6.2	1	5.5	**0**	5.2	1	5.4	**0**	**5.2**
5	BR	11	9.5	9	7.5	9	7.8	4	6.2	**0**	5.7	**0**	**5.6**	4	6.1	4	6.1	**0**	5.8	**0**	5.7	4	6.0	4	5.9
5	BI	11	9.8	7	7.6	5	7.8	4	6.1	7	6.7	8	6.8	4	6.2	4	6.0	1	5.3	**0**	5.2	2	5.4	2	**5.1**
10	PD	8	9.6	6	7.9	4	6.9	9	6.8	10	9.0	11	9.5	5	6.6	5	6.3	2	3.9	**0**	**3.3**	2	4.5	1	3.7
10	PI	7	7.6	6	6.9	4	6.2	5	6.5	10	8.9	11	9.5	5	6.6	5	6.3	3	5.1	1	**4.7**	1	5.1	**0**	4.7
10	BR	10	8.9	3	7.2	2	7.4	11	6.5	2	5.6	2	5.7	3	6.6	2	6.5	**0**	5.6	**0**	**5.6**	2	6.2	2	6.3
10	BI	8	9.1	5	7.2	2	7.0	10	6.5	8	7.4	10	7.7	5	6.7	5	6.2	**0**	4.9	**0**	**4.6**	2	5.3	2	5.2
20	PD	8	7.8	5	7.2	4	5.0	9	8.8	10	10.6	11	11.3	5	7.0	5	6.9	2	3.7	**0**	**2.7**	2	3.9	1	3.0
20	PI	6	7.3	6	7.2	4	5.3	5	6.9	10	10.3	11	11.0	6	6.9	6	7.0	3	4.7	**0**	3.9	2	4.1	1	**3.5**
20	BR	**0**	7.1	**0**	6.6	**0**	6.4	9	8.1	**0**	6.2	1	6.4	**0**	6.6	**0**	6.5	**0**	**5.6**	9	5.7	**0**	6.4	9	6.4
20	BI	2	7.2	2	6.7	1	6.0	9	8.1	9	7.6	10	8.2	2	6.7	2	6.7	**0**	5.0	**0**	**4.8**	**0**	5.6	1	5.5
40	PD	6	7.9	6	7.6	**0**	4.3	5	8.4	10	11.1	11	11.8	6	7.7	6	7.6	1	3.4	1	**2.2**	1	3.5	1	2.5
40	PI	6	7.6	7	7.9	**0**	4.5	5	6.4	10	10.8	11	11.5	6	7.7	6	7.7	1	4.1	1	**2.8**	1	4.0	1	3.0
40	BR	**0**	6.2	**0**	**6.1**	7	6.4	**0**	7.3	**0**	6.3	**0**	6.5	**0**	6.1	**0**	6.1	**0**	6.4	1	7.1	**0**	6.4	1	7.1
40	BI	5	6.4	5	6.5	**0**	5.7	5	7.4	10	7.8	10	8.1	5	6.5	5	6.5	**0**	**5.5**	**0**	6.0	**0**	5.6	**0**	6.0

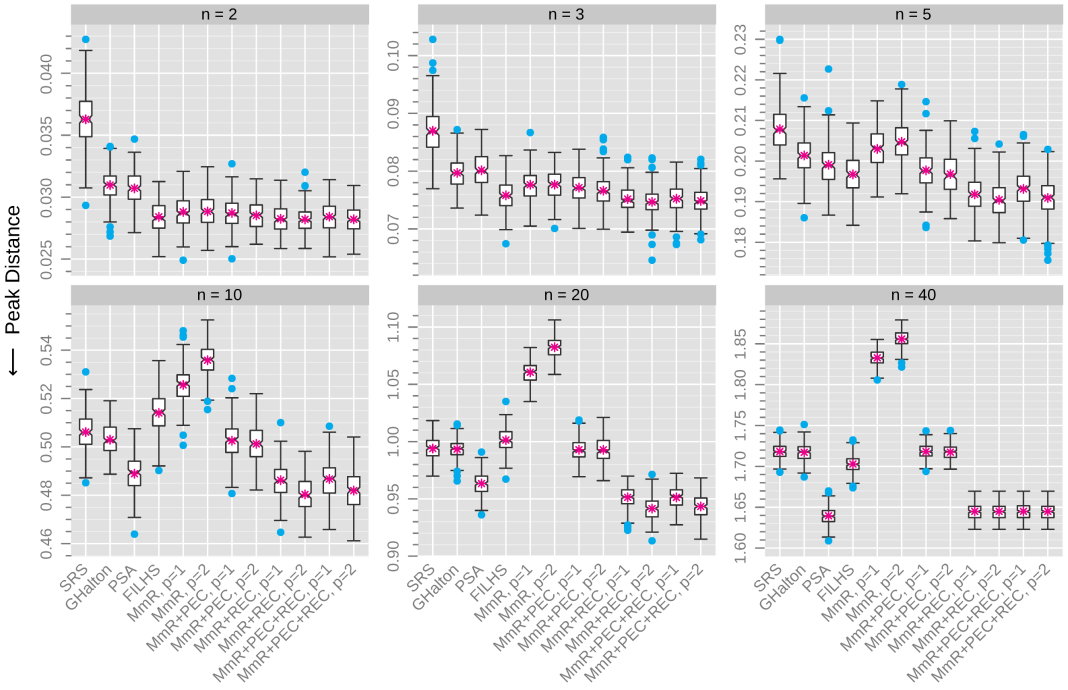


Figure 4.12: Peak distance values for $\nu = 100$ and $N_f = 10^2 n$. The mean values are indicated by ✳.

which indicates that edge effects gain a higher influence there.

On a side note, the complicated Algorithm 2 for creating the problem instances is only necessary in two and three dimensions. In higher dimensions, $\nu = N_{\text{peaks}}$ always holds from the start, and thus it is guaranteed that the optima are uniformly distributed.

Discussion The number of points N_f did not seem to exercise significant influence onto any sampling algorithm. Also the number of optima ν and problem topology seem to be irrelevant in the setting of this experiment, which is why they are omitted from the presentation. The $\bar{d}_{\mathcal{B}}$ values obtained for the sampling algorithms are as expected and confirm our measurements in previous sections.

The results according to sign tests are largely in good agreement with the mean ranks (see Table 4.3). On the whole, MmR variants with edge correction seem to obtain the best performance averaged over all indicators. Especially reflection edge correction is advantageous. Among the indicators, PD and PI form a similar pair, and BR and BI form another. For the latter ones (especially BR) the completely uniform samplings seem to do relatively better. However, the evaluations by basin ratio produced many ties and thus are the least reliable. For PD and PI, the optimal $\bar{d}_{\mathcal{B}}$ apparently is larger than δ_n. This would be in agreement with our theoretical results that "peak" indicators are related to covering radius (Propositions 1 and 2), as well as with our working hypothesis that "basin" indicators pose a task similar

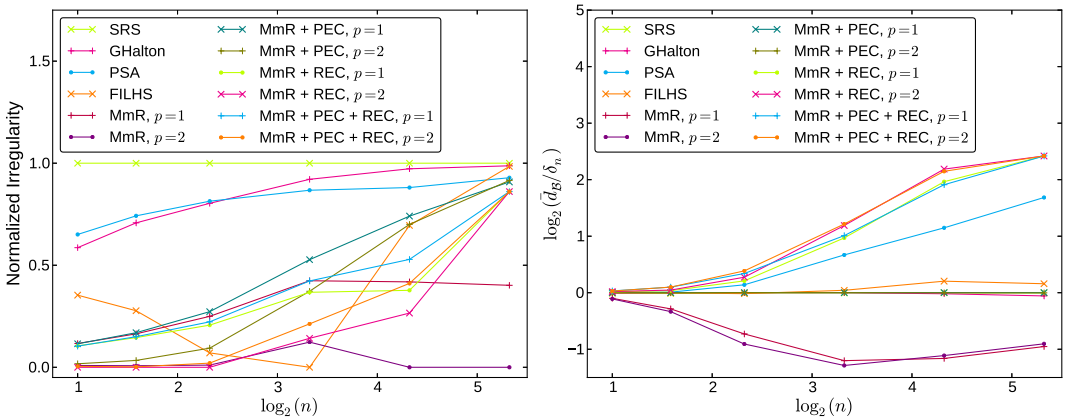


Figure 4.13: Normalized irregularity (left) and the relative distance to the boundary (right) for all analyzed sampling algorithms. The curves depict mean values of all configurations with $N_f = 10^2 n$.

to numerical integration (Section 1.2).

It is not surprising that L_1 distances yield no improvement over Euclidean distances, because we included no problems with weakly dependent variables in our experimental setup. To put it positively, one also does not loose much by using L_1 distances. Furthermore, we may have biased the experiment in favor of larger distances to the boundary, because all optima are located in the interior of the region of interest, and none on the boundary. This assumption of an *essentially unconstrained* problem is common in global optimization [143, p. 10] [141], but seems unrealistic if we suppose that the region of interest was chosen by the experimenter from a larger region. A test problem with optima also on the boundaries could be produced by doing just that: create the instance as usual and define the region of interest as a subset of the original one. Similar approaches in spatial statistics are called plus- and minus-sampling [63, pp. 183–186]. However, determining the locations of the emerging optima at the boundaries would be difficult in this approach if we use rotated basins.

The experiment generally verifies that an improvement of worst-case performance in multimodal optimization is possible by applying several different stratified samplings as variance reduction methods. As diversity indicators were not recorded in this experiment due to their individual deficiencies, the complete relationship between performance on the one hand, and diversity and distance to the boundary on the other hand, could not be revealed yet. Irregularity was measured as a surrogate for diversity to some extent, but can of course not capture all of its aspects. However, in our setting it does seem to correlate with performance in low dimensions, while in high dimensions the distance to the boundary gains importance.

5 Optimization

In this chapter, we will deal with two-stage methods whose conceptual origin are simple combinations of global and local search, called singlestart and multistart by Törn and Žilinskas [143, p. 66]. These approaches start with a space-filling sample of the region of interest and then conduct local searches either from only the best solution in the sample, or from all solutions. If we execute these algorithms in a loop, we have representatives for two-stage methods as defined in Section 1.3. Our focus is on the determination of the starting point for each local search. The goal is to place each starting point in a different attraction basin, so that the corresponding local search explores a new region, instead of revisiting an already known attraction basin. In a sense, this topic could be regarded as *basin discovery*, and it is a prerequisite for *basin identification* and *basin recognition* as described by Preuss [114, Sec. 3.2].

Explicit basin identification is for example appropriate when the optimization begins with a global stage yielding k points, and $1 < s < k$ local searches are planned to follow. It usually involves some kind of clustering [143, pp. 95–116], enabling an economical use of local searches. Each cluster should correspond to one attraction basin. Törn and Žilinskas [143, p. 66] argue that the two extreme cases singlestart ($s = 1$) and multistart ($s = k$) are inferior for global optimization, because they disregard the problem structure and thus may spend too few or too many resources on local searches. However, as clustering is affected by the curse of dimensionality, this does not have to hold generally [76].

One of the currently successful clustering approaches is due to Preuss [114], who developed a two-stage method combining a CMA-ES with a sophisticated restart management. The algorithm, called *niching evolutionary algorithm 2* (NEA2), won the CEC 2013 multimodal optimization competition [80]. In each iteration, it draws a random uniform sample of the search space, determines a variable number of clusters by a procedure described in Section 5.2.1, and starts one local search from each cluster. Due to the batch-sequential fashion of the approach, one might not only try to identify basins once, but also recognize the previously identified basins again. Such behavior was investigated by Preuss, but later discarded, because it did not fulfill the expectations regarding improvement of performance.

The downside of the method is the requirement for additional function evaluations to assess a space-filling sample prior to every (re-)start of the local stage. This cost of around $50n$ per iteration [114, Sec. 6.2] outweighs its benefits in higher dimensions as the basin identification becomes more difficult. Therefore, NEA2 looses its advantage over CMA-ES with independent restarts at about $n \geq 20$ [114, Sec. 6.4], making it seem advisable to begin with the local stage and completely avoid the cost for basin identification in high dimensions.

In principle, all of the mentioned two-stage approaches can benefit from an improved basin discovery. Using our (sequential) sampling algorithms from Chapter 4, this improvement comes at no additional cost in terms of function evaluations. However, it is of course inevitable that several points are sampled in the same basin while others remain undetected (see the results for the basin ratio in Section 4.5). We cannot even guarantee to start only one local search per basin, because even an explicit basin identification can never work perfectly.

Previous improvements for basin discovery have mainly been developed for problems with funnel structures, where it is beneficial to adapt the sampling to focus on the vicinity of previously found optima. One attempt was made in connection with EAs by Cuccu et al. [28], who use the novelty criterion $d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P}, k)$ to determine promising start locations. The candidates $\boldsymbol{x}$ are only the already visited points, because the authors deal with unconstrained search spaces. Another approach relies on using small perturbations of found local optima as starting points. In real-valued optimization, this idea appears in basin hopping [148] and its variants [4]. The concept is also known as iterated local search, especially in combinatorial optimization [85]. Note that although this name sounds quite general, it is normally associated exclusively with the strategy of perturbing local optima. Furthermore, we remark that all of the algorithms in [28, 148, 4, 85] fit into the two-stage paradigm.

As we will not presuppose funnel structures, our global stage will rely on uniform sampling, which is also necessary to ensure a theoretical convergence to the global optimum. It is, however, possible to combine uniform sampling, adaptive sampling, and local search in one algorithm [1]. The rationale behind this approach is that of obtaining a "globalized" local search [127], which is something that CMA-ES provides implicitly, given an appropriate parametrization [86].

Topics as (low-level) parameter control are ignored in this chapter, although especially for the CMA-ES, sophisticated heuristics exist to adapt algorithm parameters between restarts [53, 156]. Also Preuss does not use these parameter control capabilities, except for a custom-built heuristic to set the initial mutation strength of his NEA2 [114, Sec. 6.1.3]. Omitting this topic now is not a severe problem, as the later incorporation of such heuristics is straightforward and should not interact much with our global stage, thanks to the modular structure of two-stage methods [85].

5.1 Restarted Local Search

Beginning with local search is especially advisable when it is unknown if the problem is really multimodal. If it is not, there is only one optimum, which can be found straightforwardly with one local search. So it would be wasteful to invest function evaluations into a global exploration first. Likewise, executing local searches in a strictly sequential order facilitates the optimal utilization of the available information: While the (potentially) parallel multistart assumes independent local searches, the sequential execution enables parameter adaptation. For the CMA-ES, this was originally proposed for the population size by Hansen and Kern [55] and

later also much more sophisticated approaches appeared [53, 156]. Sequential execution also enables the seamless incorporation of information obtained by previous local searches, e. g., already found local optima.

Pošík and Huyer [112] compare independent restart variants of several other derivative-free optimization algorithms on the BBOB testbed, concluding that restarted local search (RLS) is highly competitive, but there is no single best local search algorithm. So, we will not try to identify one in our experiments, either. But it seems that no attention has ever been paid to an improved determination of starting points for restarted local search in box-constrained search spaces, which is why this is our next topic.

5.1.1 Experiment on Restarted Local Search

Research Question Is there an advantage from using a sequential sampling with optimized diversity as a global strategy in a two-stage algorithm consisting of restarted local searches?

Pre-experimental Planning In Section 4.5 we saw that the samples' mean distance to the boundary can have a strong influence on performance. It is quite possible that this effect can also be observed for the starting points of local searches. At least Hansen et al. [54] recommend for the BBOB testbed to choose starting points from the interior of the search space, namely from $\mathcal{Y} = [-4, 4]^n$ when the search space is $\mathcal{X} = [-5, 5]^n$. This choice is extreme, as $\operatorname{vol}(\mathcal{Y})/\operatorname{vol}(\mathcal{X})$ tends to zero for $n \to \infty$. Also multilevel single linkage (MLSL), a historic two-stage algorithm employing clustering, only starts local searches from points $\boldsymbol{x}$ for which $d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{B})$ is larger than some fixed threshold [119, 120]. According to Schoen [127, p. 165], this also leads to problems in high dimensions because the feasible space for starting points may tend to zero. Our edge corrections instead provide an adaptive behavior, depending on both n and N.

Task The main objective of this experiment is not to compare different local search algorithms, but to investigate the influence of different sequential sampling algorithms in the restart mechanism. Nonetheless, several local methods are tested, because they can have a strong influence on performance. The assessment is done by only considering the final outcomes of the local searches. This way, we avoid the overhead of selecting a subset from all points. The sets are evaluated with PR, F1P, and AHD. The name F1P shall indicate that we are using (2.6) in the formulas for precision and recall, which are then aggregated as in (2.10) for the F_1 measure. The parameter of PR and F1P, determining if an optimum has been approximated, is set to the moderate value of $r = 10^{-3}$.

The selection of indicators is guided by two contradictory considerations: On the one hand, we should employ indicators penalizing larger approximation sets, following the argumentation in Section 2.2. On the other hand, larger approximation sets are achievements of the global sampling algorithms, and thus should be rewarded. Moreover, the approximation sets are all quite small anyway, because their size is

Table 5.1: High-level factors for the experiment in Section 5.1.

Factor	Type	Symbol	Levels
Problem topology	non-observable		{random, funnel}
Number of local optima	non-observable	ν	$\{5, 20, 100, 500\}$
Number of variables	observable	n	$\{2, 3, 5, 10, 20, 40\}$
Budget	observable	N_f	$\{10^3 n, 10^4 n\}$
Global algorithm	control		{SRS, MmR}
Archive	control	$\mathcal{A}$	$\{\mathcal{S}, \widehat{\mathcal{O}}, \mathcal{S} \cup \widehat{\mathcal{O}}\}$
Local search	control		{Nelder-Mead, L-BFGS-B, CMA-ES}

limited by the number of local searches that can be afforded with the given budget. So, penalization is not really necessary. Both arguments are represented in this set of indicators.

The obtained indicator values are aggregated to a consensus ranking with the same two approaches as in Section 4.5. The Bonferroni correction now uses $c = 9$, because we have only this number of competing sampling algorithms left.

Setup Table 5.1 contains the high-level factors for this experiment. The setup is largely identical to Section 4.5, with the exceptions of larger budgets of function evaluations N_f and more factors concerning optimization algorithms. Funnel topologies are included in the setup mainly to rule out performance deteriorations for the new algorithms, and not to identify especially well-suited configurations for this special case. The number of stochastic replications is set to fifty. Again, the eight MmR variants plus SRS as baseline method are tested. The other sampling algorithms cannot be used reasonably because they do not provide a sequential sampling.

As it would be too computationally expensive to consider all previously visited points in the sequential sampling, we test three more economic strategies to fill the archive $\mathcal{A}$: using the preceding starting points $\mathcal{S}$, the approximation set $\widehat{\mathcal{O}}$ consisting of the results of the previous local searches, and the union of both, $\mathcal{S} \cup \widehat{\mathcal{O}}$. (We have to admit that if only $\mathcal{S}$ is considered, also quasirandom sequences would be viable alternatives, especially if many starting points are needed. Compared to MmR they have the advantage of low run time, but the disadvantages of not having control over the distance to the boundary and not yielding good results for small sample sizes. We leave them out of the experiment to reduce the computation time.)

As local searches we employ L-BFGS-B by Byrd et al. [22], the downhill simplex method by Nelder and Mead [107], and CMA-ES by Hansen and Ostermeier [56]. These three are important representatives of quasi-Newton methods, direct search methods, and evolutionary algorithms, respectively. For L-BFGS-B and Nelder-Mead, implementations from the Python library SciPy (version 0.9.0) are employed. For CMA-ES, version 1.1.02 of the implementation available at `https://pypi.python.org/pypi/cma` is used. Note that function evaluations are counted directly

in the objective function, because at least the number L-BFGS-B reports is smaller than what is actually used. (Apparently the evaluations needed for approximating the gradient are not accounted for.) L-BFGS-B and CMA-ES come with their own constraint handling. For Nelder-Mead we use Baldwinian reflection to repair violations of the box constraints [154]. "Baldwinian" means that after the constraint violation is repaired, the objective function value of the repaired individual is assigned to the original, infeasible individual. This way, the optimization algorithm does not have to know about any constraints, which allows this approach to be added very conveniently to existing implementations. "Reflection" means that each decision variable $x \in (-\infty, \infty)$ is mapped into its feasible range $[\ell, u]$ by using the recursive function

$$T(x) = \begin{cases} x & \ell \leq x \leq u\,, \\ T(u + (u - x)) & x > u\,, \\ T(\ell + (\ell - x)) & x < \ell\,. \end{cases}$$

This repair method has a plus factor of not adding discontinuities to the objective function.

With very few exceptions we are using the default parameters of the algorithms. The modified parameters all correspond to stopping criteria. For L-BFGS-B, "pgtol" is decreased to 10^{-8} to suppress this criterion. This decision was taken after some manual experimentation on unimodal MPM2 instances, where higher values more often led to premature convergence in flat areas of the search space. For CMA-ES, "tolfun" is increased to 10^{-6} to induce an earlier stopping as in [114, Sec. 6.3]. The initial mutation strength for CMA-ES has no consistent default value. With $\sigma_0 = 10^{-2} \cdot 0.5(u - \ell)$ it is chosen very small, in accordance with [9]. So, the globalized search capabilities of CMA-ES are rather underused, to avoid finding optima in large attraction basins disproportionately often.

Results Several Box plots are used for a first explorative data analysis: Figure 5.1 shows the number of starts depending on the local search algorithm, the dimension, and the sampling algorithm to determine the starting points. Figure 5.2 divides the peak ratio performance according to the problem topology and the used archive points, while Figure 5.3 focuses on the effects of the sampling algorithms on the peak ratio. Table 5.2 contains the aggregated data of the experiment in the same fashion as in Section 4.5.

On a side note, the three optimization algorithms CMA-ES, L-BFGS-B, and Nelder-Mead attain median peak ratios of 0.15, 0.26, and 0.16 on the whole experiment, respectively. The corresponding median precision values are 0.4, 0.14, and 0.16.

Observations Regarding local search, the experiment confirms previous results that Nelder-Mead is quite successful in low dimensions, but its performance deteriorates quickly with an increasing number of decision variables n [97, 112]. CMA-ES and L-BFGS-B are more stable in the sense that their required number of function evaluations until convergence grows slower with n. While L-BFGS-B achieves much

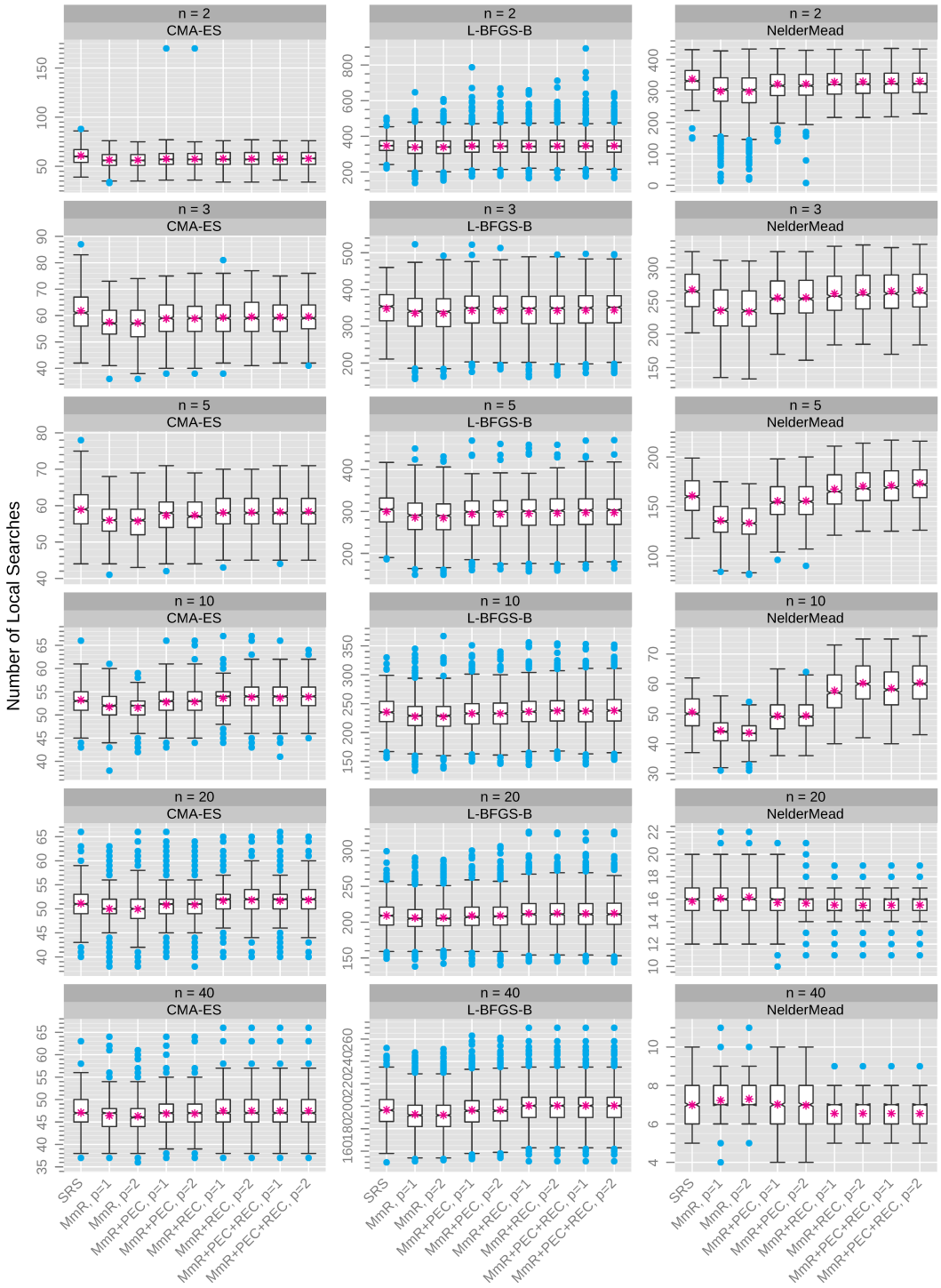


Figure 5.1: Number of local searches for $N_f = 10^4 n$, depending on sampling algorithms.

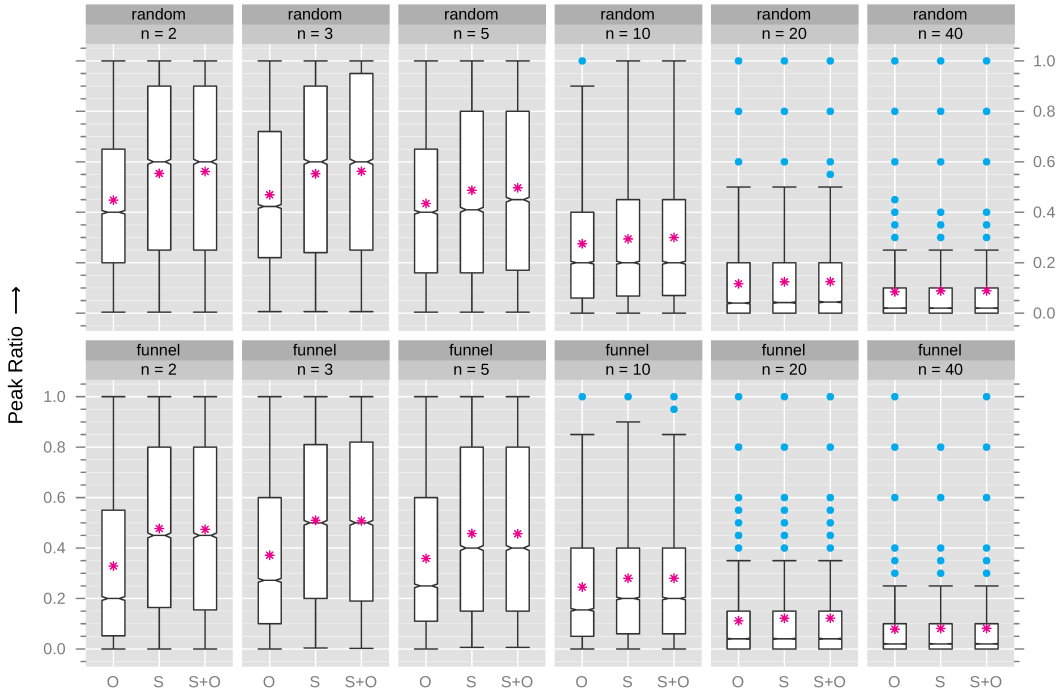


Figure 5.2: Influence of the used archive points on the peak ratio.

faster convergence, its precision is also constantly worse than that of CMA-ES. Consequently, CMA-ES catches up with increasing dimension and is the best algorithm for $n = 40$.

For the smaller budget, $N_f = 10^3 n$, only very few restarts can be afforded, so Figure 5.1 only focuses on the larger one. The global sampling algorithm has a slight influence on the length of local searches, and thus also on the number of searches with a fixed budget (see Figure 5.1). Interestingly, the influence is not the same for every local search algorithm and not always positive (especially in low dimensions). For example, starting close to the boundary seems to be harmful for Nelder-Mead, which produces many outliers with very few restarts if MmR is used without edge correction. Also the largest positive effect can be found for Nelder-Mead in ten dimensions, where an improvement of up to 20% can be achieved, compared to SRS. Two lone outliers are produced by MmR+PEC with CMA-ES on a two-dimensional instance with five optima and $\mathcal{A} = \widehat{\mathcal{O}}$. Here, the number of restarts is very high, but the results are of inferior quality. The ultimate reason for this behavior could not be identified, but Figure 5.2 clearly shows that if we use MmR as sampling algorithm, the archive should definitely contain the starting points $\mathcal{S}$. Especially for the random topology, it may be advisable to also include the found optima $\widehat{\mathcal{O}}$, but using $\widehat{\mathcal{O}}$ alone should be disregarded in any case. Thus, the remaining analysis is restricted to configurations including the starting points.

Figure 5.3 shows that there is a statistically significant benefit in terms of peak

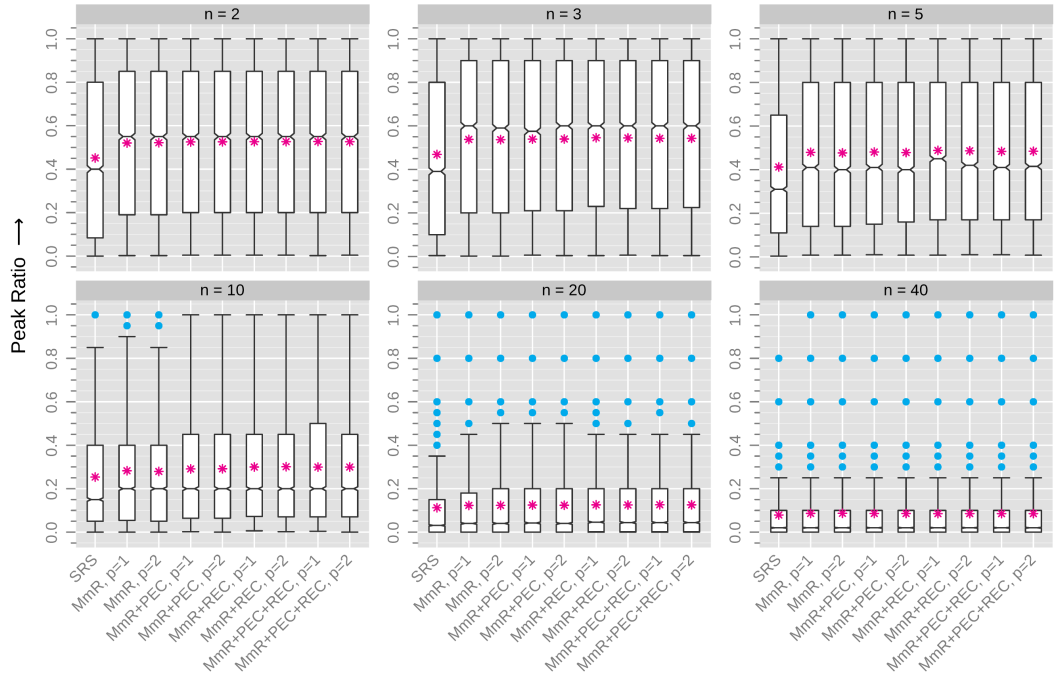


Figure 5.3: Peak ratio values for different sampling algorithms, using $\mathcal{S} \cup \widehat{\mathcal{O}}$ or only $\mathcal{S}$ as archive points.

ratio from using MmR as global sampling algorithm. Naturally, the effect diminishes with increasing dimension. Table 5.2 confirms these observations also for the two other indicators.

Discussion In total the improvements by using MmR seem to be moderate but consistent. This may be due to the relatively low number of local searches conducted even with the larger budget. It is to be expected that the advantage of MmR increases with the budget and that the same effect can be achieved also in higher dimensions, given a sufficiently large budget is used. Unfortunately, the curse of dimensionality suggests that the necessary budget increases exponentially with dimension.

The distance to the boundary does not seem to have much influence in the view taken in Figure 5.3. However, Table 5.2 does suggest such an influence: While F1P favors the pure maximin approach, PR and AHD reward the variant with reflection edge correction. So, apparently both the quality of the distribution and the mean distance to the boundary play a role. In accordance with this theory, periodic edge correction always ranges somewhere in the middle of all MmR configurations, but is steadily better than SRS. Overall, MmR with reflection edge correction and L_1 distances seems to be the recommended variant in this experiment. A remaining question is why the assessment of F1P is so different from the other indicators.

To further improve the performance, the local search methods should be screened for tunable algorithm parameters, which could then be optimized with established

Table 5.2: Aggregated results of the restarted local search experiment for $N_f = 10^4 n$ and disregarding configurations with $\widehat{\mathcal{O}}$ as archive points. Column "D" denotes the number of pairwise tests where another algorithm was significantly better, "R" is the algorithm's mean rank. The best results are printed in bold.

n	Indicator	SRS		MmR $p=1$		MmR $p=2$		MmR +PEC $p=1$		MmR +PEC $p=2$		MmR +REC $p=1$		MmR +REC $p=2$		MmR +PEC +REC $p=1$		MmR +PEC +REC $p=2$	
		D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R
2	PR	8	7.2	2	4.9	3	4.9	**0**	4.7	**0**	4.7	**0**	**4.6**	**0**	4.7	**0**	4.6	**0**	4.6
2	F1P	8	7.3	**0**	4.1	**0**	**4.1**	2	4.7	2	4.7	4	5.0	4	5.2	4	5.1	4	5.1
2	AHD	8	7.1	4	4.9	3	4.9	**0**	4.7	**0**	4.8	**0**	4.6	**0**	4.7	**0**	4.6	**0**	**4.6**
3	PR	8	7.1	6	5.1	6	5.1	3	4.8	3	4.8	**0**	4.5	**0**	**4.4**	**0**	4.6	**0**	4.5
3	F1P	8	7.2	1	4.0	**0**	**3.8**	2	4.6	2	4.6	4	5.0	4	5.1	6	5.3	6	5.4
3	AHD	8	7.2	6	5.3	6	5.3	4	4.8	4	4.8	**0**	4.3	**0**	**4.2**	1	4.5	**0**	4.5
5	PR	8	7.2	6	5.2	6	5.3	4	5.0	4	4.9	**0**	4.2	**0**	**4.2**	2	4.6	2	4.5
5	F1P	8	7.1	1	3.9	**0**	**3.8**	2	4.8	2	4.7	2	4.8	5	5.0	6	5.4	6	5.5
5	AHD	8	7.3	6	5.4	6	5.5	4	5.1	4	5.0	**0**	4.0	**0**	**3.9**	2	4.4	2	4.3
10	PR	8	6.7	6	5.4	7	5.6	4	5.1	4	5.0	**0**	**4.1**	**0**	4.2	2	4.4	2	4.5
10	F1P	8	6.6	**0**	4.7	**0**	4.8	2	4.9	1	4.8	**0**	**4.5**	1	4.7	1	4.9	5	5.1
10	AHD	8	6.8	4	5.0	3	5.1	4	5.1	4	5.1	**0**	**4.3**	**0**	4.3	1	4.6	2	4.7
20	PR	8	6.3	6	5.2	6	5.2	4	4.9	4	5.0	**0**	4.5	1	4.7	**0**	**4.5**	**0**	4.6
20	F1P	8	6.2	**0**	4.8	**0**	4.8	**0**	4.8	**0**	4.9	**0**	4.8	2	5.0	**0**	**4.7**	1	4.9
20	AHD	8	6.3	**0**	4.8	**0**	4.9	**0**	4.8	**0**	4.8	**0**	**4.7**	1	5.0	**0**	4.8	1	5.0
40	PR	8	5.7	**0**	4.8	**0**	**4.7**	**0**	4.8	**0**	4.9	1	5.0	1	5.0	1	5.0	1	5.0
40	F1P	8	5.7	**0**	4.3	**0**	**4.2**	2	4.7	2	4.8	4	5.4	4	5.4	4	5.4	4	5.4
40	AHD	8	6.1	4	5.0	**0**	5.0	**0**	5.0	**0**	5.0	**0**	4.7	**0**	4.7	**0**	**4.7**	**0**	4.7

tuning software [11]. An even more promising but also more challenging approach would be to incorporate and further develop parameter control strategies, which adapt parameters between local searches, as the ones already existing for the CMA-ES [53]. Besides, a more detailed analysis of the behavior of Nelder-Mead would be in order, as there seems to be an issue with its initialization: The algorithm uses a simplex consisting of $n + 1$ points. In the SciPy implementation, the points $\boldsymbol{x}_i$, $i = 2, \ldots, \mathrm{n} + 1$ are constructed from the starting point $\boldsymbol{x}_1$ by multiplying coordinate $i-1$ by 1.05, respectively. So, if the algorithm is started close to the upper boundaries, the constraint handling will come into operation right at the beginning. Even worse, if it is started close to the lower boundaries, the initial simplex will be extremely small and/or degenerated. Altogether, it seems that the difficulty of properly normalizing the search space and choosing an initial simplex size has been hidden from the user by employing a questionable heuristic.

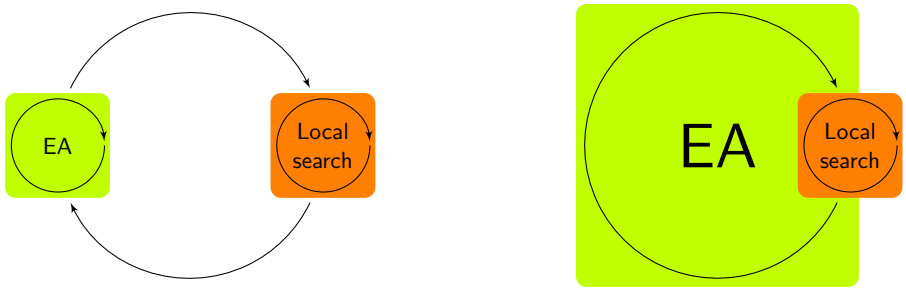
5.2 Clustering Methods

As already explained in the introduction of Chapter 5, there are situations where clustering methods (CM) outperform restarted local searches by putting more emphasis on the global sampling part. An interesting question for us is if clustering methods can also be improved more than restarted local searches by using a more sophisticated global sampling strategy. Of course, we will test our usual suspects, the variants of MmR.

There are mixed results in the literature regarding the combination of optimization algorithms and "improved" sampling. (Apparently, only low-discrepancy point sets and latin hypercube sampling have been considered so far.) Preuss does not find any significant differences between LHS and SRS for his NEA2 [114, Sec. 4.6.4]. Also Omran et al. [110] do not find any significant difference to SRS by using low-discrepancy point sets when initializing population-based optimization algorithms (differential evolution and particle swarm optimization). This, however, is not surprising, as in their case the used population size of fifty is negligible in comparison to the employed budgets of 10^5 objective function evaluations. Our budget for the global stage will be several magnitudes smaller. On the positive side, it was shown even twice that multilevel single linkage can be improved by using quasirandom point sets (Halton or Sobol') in the global stage [6, 76].

There are also arguments for deviating from uniformity. One justification to abandon random uniform samples are investigations by Morgan and Gallagher [102], who show that the curse of dimensionality may be alleviated by using a different sampling, e. g., random walks. Additionally, it seems promising to incorporate Törn's [143, p. 95] old idea of investing a few optimization steps into concentrating the global point sample around the optima. At the time, this was mandatory to obtain a clustering corresponding to the attraction basins at all, because only "conventional" clustering methods were available [143, pp. 95–116]. But it also had the advantage that the local stage was disburdened from some of the work. Nowadays, we are able to cluster arbitrary point sets, e. g., via MLSL [120], topographical clustering [142], or nearest-better clustering (NBC) [116], but the benefits of good starting points may have been partly forgotten. Exceptions are the works of Lourenço et al. [85] in combinatorial optimization and Addis et al. [4] in real-valued optimization, which show that run lengths of local searches can be reduced significantly by using starting points that are already close to local optima. In their cases, the points are obtained by simply perturbing local optima. As already indicated earlier, the usefulness of this approach is limited to problems with a single funnel.

A more disruptive generation of starting points, which is useful for problems with multiple funnels [1], could be achieved by applying variation operators (especially recombination) of EAs to a population of local optima. Locatelli and Schoen describe such ideas in a survey paper [84]. This approach of course constitutes a hybridization of EAs with local search, which is well known in evolutionary computation under the term *memetic algorithm* [39, pp. 173–188]. The most common variant of memetic algorithms applies an improvement step to the outcome of the variation operators of



(a) Two-stage algorithm (b) Memetic algorithm

Figure 5.4: Differing concepts of employing EAs in two-stage and memetic algorithms.

the EA in each generation [39, pp. 181-182]. This tight integration of the two modules is illustrated in Figure 5.4b. Although this concept could also be interpreted as a two-stage optimization algorithm, we want to stay closer to Törn's idea of concentrating a global sample, by letting an EA run independently for several generations. This stronger separation of EA and local search, as shown in Figure 5.4a, gives us a good opportunity to cluster the EA's outcome and only run the local search for selected points. It is assumed that this finer control also enables a better performance.

Note that although being an evolutionary algorithm, the CMA-ES is rather not suited as a global stage, because its strategy is aimed at quickly converging to a local optimum. Instead we examine two evolutionary algorithms that are better able to maintain diversity. Both feature a special selection component, which relies on a notion of the distance from one solution to the nearest other solution offering a better objective value.

5.2.1 Utilizing Nearest-better Distances

As sophisticated two-stage methods necessarily spend objective function evaluations in the global stage, the question arises how to use the gained information effectively. One possible answer of course is to use a meta-model of all the available data [46], to get a comprehensive model of the problem's landscape and thus to find out which attraction basin a solution belongs to. The model would even allow to predict positions of unvisited local optima without spending additional function evaluations. However, as meta-modeling is computationally quite expensive, this approach may be rather suited for smaller budgets of function evaluations than we consider here. One remedy may be to build the meta-model only on the local optima obtained so far [84]. This has the advantages that the lower number of training points implies a lower run time for the model fit and that the resulting model would presumably be smoother.

An even less demanding approach is probably obtained by more directly trying to transfer the goals of multimodal optimization, as expressed through the quality

indicators in Section 2.1.3, to the level of single solutions. For this purpose, it seems beneficial to seek some biological inspiration, because natural evolution is obviously able to maintain diversity while creating better adapted species. An important mechanism in this regard is *niche differentiation*. Roughly speaking, this term refers to the fact that, if two species occupy the same ecological niche, the competition between them (interspecific competition) forces them to become more different from each other over time. If this differentiation does not occur (fast enough), then one species will become extinct. This observation from nature is formulated in the *competitive exclusion principle*, which in short states that "complete competitors cannot coexist" [58]. The mechanism has (vaguely) inspired a whole area of research in evolutionary computation [114, Sec. 5.3]. The competition between individuals of the same species (intraspecific competition) has similar effects. Here, the goal of a social animal may be to become the alpha animal of the pack, while a solitary animal may be rather looking for its own territory.

The actual details of all these biological examples are irrelevant to us, because our optimization problem possesses neither temporal nor spatial (apart from $\mathcal{X}$, which we interpret as genotypic space) aspects. The crucial aspect for us is the competition and the important question is: How can we model the avoidance of competition by other individuals in our optimization algorithm? Transferring the concepts from the biological examples above, a possible answer could be to take a solution's distance to better solutions as fitness criterion. This criterion would still promote the approximation of the global optimum, because its distance to the (nonexistent) better solutions can be seen as infinite. However, the second best solution would not be in the ε-environment of the global optimum, but rather at a separate locally optimal position.

Definition 7. *The distance to the nearest better neighbor of $\boldsymbol{x} \in \mathcal{X}$ in $\mathcal{P} \subset \mathcal{X}$ is defined as*

$$d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) = \min\{d(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{P} \wedge f(\boldsymbol{y}) < f(\boldsymbol{x})\}\,. \tag{5.1}$$

A first reference to these distances can be found in [116]. It necessarily holds that $d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) \geq d_{\mathrm{nn}}(\boldsymbol{x}, \mathcal{P})$, because the considered distances to better solutions are a subset of the distances to all solutions. We define $d_{\mathrm{nb}}(\boldsymbol{x}^*, \mathcal{P}) := \infty$, because the best solutions $\boldsymbol{x}^*$ regarding the objective value have no nearest better neighbor. The nearest better neighbor itself shall be denoted $\mathrm{nbn}(\boldsymbol{x}, \mathcal{P})$ and is obtained by using $\arg\min$ instead of min in (5.1).

The distance d_{nb} can be used to sort a population and thus to select the fittest individuals. Algorithm 8 describes a multiobjective and a single-objective instance of such a selection mechanism. Both consider $m = 2$ objective values per individual, but process them differently. The former one uses $f(\boldsymbol{x})$ as first and $-d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P})$ as a second objective (both to be minimized). The ranking is established by calculating non-dominated fronts [33] and then sorting each front by objective value $f(\boldsymbol{x})$. The latter selection variant establishes a ranking by lexicographic ordering according to the tuples $(-d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}), f(\boldsymbol{x}))$. Thus, the multiobjective selection treats the objectives (although not completely) more on equal terms than the single-objective

Algorithm 8 Nearest-better selection

Input: population $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\mu+\lambda}\}$, archive $\mathcal{A}$
Output: μ surviving individuals

$\mathcal{Q} \leftarrow \mathcal{P} \cup \mathcal{A}$
for all $\boldsymbol{x} \in \mathcal{P}$ **do**
  calculate $d_{\text{nb}}(\boldsymbol{x}, \mathcal{Q})$
end for
if multiobjective is true **then**
  compute non-dominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_k$
  for all $\mathcal{F}_i \in \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ **do**
    sort $\mathcal{F}_i$ ascending by objective values $f(\boldsymbol{x})$
  end for
  $\mathcal{P} \leftarrow$ concatenate $\mathcal{F}_1, \ldots, \mathcal{F}_k$
else if multiobjective is false **then**
  sort $\mathcal{P}$ by $(-d_{\text{nb}}(\boldsymbol{x}, \mathcal{Q}), f(\boldsymbol{x}))$ in ascending lexicographic order
end if
remove last λ elements of $\mathcal{P}$ // truncation
return $\mathcal{P}$

one. Both variants finally apply a truncation selection to the ranked population to obtain a subset. The worst-case run time of both approaches is $O(N^2 n)$ for naïve implementations due to the distance computations.

These selections are very similar to those proposed in [157] under the names SV4 and SV7, respectively. In the original definition, only one point at a time was removed, leading to a run time of $O(N^3 n)$. However, experiments in [155], where also a detailed overview of other closely related variants can be found, showed that this precaution is not necessary. As we are only considering these two cases, we will simply call them multiobjective nearest-better selection (MNBS) and single-objective nearest-better selection (SNBS). MNBS always guarantees to retain the best solution in the population and generally puts a high focus on exploitation. In [158] an EA using this selection always converged to a single local optimum. SNBS only guarantees to retain the best solution if no better one is present in the archive [155]. Here, experiments suggest that the population becomes highly dispersed while simultaneously approximating several optima [158]. If we would set $\lambda = 1$ and used no archive points, then SNBS would even have almost the same effect as maximizing the minimal distance [155].

The selection components will be employed in a very simple EA shown in Algorithm 9. It generates λ offspring per generation by adding a multivariate normally distributed random vector to a randomly chosen parent individual. Recombination is not used, because it has been shown to have adverse influence at maintaining diversity [116] (unless specialized variants are employed). There is no maximum age for individuals, so the approach corresponds to what is called $(\mu + \lambda)$ in the literature [16]. Self-adaptation of the mutation strength σ, as it was used in [158], is

Algorithm 9 Non-recombinant $(\mu + \lambda)$ evolutionary algorithm

Input: initial population $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mu\}$, archive $\mathcal{A}$
Output: database $\mathcal{D}$ of all generated points

for all $\boldsymbol{x} \in \mathcal{P}$ **do**
  evaluate $f(\boldsymbol{x})$
end for
$\mathcal{D} \leftarrow \mathcal{P}$
$t \leftarrow 1$
repeat
  $\mathcal{F}_t \leftarrow \emptyset$ // filial generation t
  for all $i \in \{1, \ldots, \lambda\}$ **do**
    $\boldsymbol{x} \leftarrow$ randomChoice$(\mathcal{P})$ // choose parent
    $\boldsymbol{x}_i \leftarrow \boldsymbol{x} + \sigma N(\mathbf{0}, \mathbf{I})$ // generate mutated offspring
    evaluate $f(\boldsymbol{x}_i)$
    $\mathcal{F}_t \leftarrow \mathcal{F}_t \cup \{\boldsymbol{x}_i\}$
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{F}_t$
  $\mathcal{P} \leftarrow$ selection$(\mathcal{P} \cup \mathcal{F}_t, \mathcal{A})$ // determine survivors, Alg. 8
  $t \leftarrow t + 1$
until termination
return $\mathcal{D}$

not considered so far, because a budget of around $50n$ is probably too small for this to have an effect anyway. Except for the actual μ and λ values, the resulting EA is identical to the one employed in [155]. As we do not use the EA purely for optimization, but as a vehicle to obtain a non-uniform sample of the search space, all generated points are recorded and finally returned. For the whole two-stage algorithm, the property of theoretical convergence to the global optimum is retained with this approach, because the initial population of the EA, which was sampled uniformly, is included in this set. A comprehensive introduction to EAs in general is given by Beyer and Schwefel [16].

To describe the complete global stage, we finally have to explain nearest-better clustering, NEA2's method to identify promising starting points for the local searches. This clustering algorithm is applied to a space-filling sample of the search space and relies on the nearest-better distances we already saw in Definition 7. The actual operation of NBC is described in Algorithm 10. In a first step, it creates a spanning tree consisting of edges from points to their nearest better neighbors. Afterwards, the tree is divided into several connected components by removing "long" edges. The run time is again governed by the quadratic number of distance computations necessary for building the graph.

For characterizing edges as long, two heuristics exist, which are called rule 1 and rule 2 in the pseudocode. Rule 1 simply removes all edges whose length exceeds the mean length of all edges by more than a factor ϕ. Rule 2 was added later. It is only

Algorithm 10 Nearest-better clustering

Input: points $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
Output: clusters in form of connected components of a graph

create a weighted, directed graph $G = (V, E)$ with $V = \{v_1, \ldots, v_N\}$ and $E = \emptyset$
for all $i \in \{1, \ldots, N\}$ **do**
  if $\mathrm{nbn}(\boldsymbol{x}_i, \mathcal{P})$ exists **then**
    $\boldsymbol{x}_j \leftarrow \mathrm{nbn}(\boldsymbol{x}_i, \mathcal{P})$
    $e_i \leftarrow (v_i, v_j)$ // create edge
    $w_{e_i} \leftarrow d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ // set weight equal to distance
    $E \leftarrow E \cup \{e_i\}$ // add edge to graph
  end if
end for // G is now a spanning tree
$w_{\max} \leftarrow \phi \cdot 1/|E| \sum_{i=1}^{|E|} w_{e_i}$ // calculate weight threshold for rule 1
$E' \leftarrow E$
for all $e_i \in E'$ **do** // apply rule 2
  let $e_i = (v_i, v_j)$
  if $\deg^-(v_i) \geq 3$ **then**
    let $e_1^-, \ldots, e_k^-$ be the incoming edges of v_i
    if $w_{e_i} / \mathrm{median}\{w_{e_1^-}, \ldots, w_{e_k^-}\} > b$ **then**
      $E \leftarrow E \setminus \{e_i\}$
    end if
  end if
end for
for all $e_i \in E'$ **do** // apply rule 1
  if $w_{e_i} > w_{\max}$ **then**
    $E \leftarrow E \setminus \{e_i\}$
  end if
end for
return G

applied to edges e whose tail v has an indegree $\deg^-(v) \geq 3$. The rule states to cut such an edge e if its length w_e is more than b times longer than the median of the incoming edges of v. The parameter b has been derived by extensive experimentation and is actually dependent on the number of points and the dimension [114, Sec. 4.5]:

$$b(N, n) = (-4.69 \cdot 10^{-4} n^2 + 0.0263n + 3.66n^{-1} - 0.457) \cdot \log_{10}(N) + 7.51 \cdot 10^{-4} n^2 - 0.0421n - 2.26n^{-1} + 1.83 \,.$$

The aim of this involved rule 2 is to produce a correction yielding more clusters for large random uniform samples on highly multimodal functions, while not detecting more than 1.1 clusters on average in the case of unimodal functions [114, Sec. 4.5]. While this rule may have been overfitted to random uniform samples, it seems that it cannot do too much harm, because its condition is only seldom satisfied. Also note

that under the assumption of unique objective values for all solutions in $\mathcal{P}$, rule 2 can never come into effect on one-dimensional problems. We will show this using the kissing number τ_n, the maximal number of non-overlapping unit hyperspheres in Euclidean space that can be arranged such that they all touch another central unit hypersphere [27, p. 21]. In other words, τ_n is the highest number of points that can have a common nearest neighbor.

Proposition 9. *Let $\mathcal{P} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $N < \infty$, and $V = \{v_1, \ldots, v_N\}$ the corresponding nodes in the minimum spanning tree constructed by Algorithm 10. If $\forall i, j \in \{1, \ldots, N\}, i \neq j : f(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_j)$, it holds for every node v_i that*

$$\deg^-(v_i) \leq \tau_n \,.$$

Proof. The proof is by contradiction. Assume a point $\boldsymbol{x}_i$ has a set of neighbors $\mathcal{Q} \subset \mathcal{P}$, with $|\mathcal{Q}| > \tau_n$ and $\forall \boldsymbol{z} \in \mathcal{Q} : \mathrm{nbn}(\boldsymbol{z}, \mathcal{P}) = \boldsymbol{x}_i$, meaning that $\deg^-(v_i) = |\mathcal{Q}|$. From the definition of τ_n it follows that at least two of these points must be closer to each other than $d_{\mathrm{nn}}(\boldsymbol{x}_i, \mathcal{Q})$. Let these points be $\boldsymbol{z}'$ and $\boldsymbol{z}''$. But according to our premise we also have $f(\boldsymbol{z}') \neq f(\boldsymbol{z}'')$. Thus one of the two must be the nearest better neighbor of the other, in contradiction to our assumption that $\mathrm{nbn}(\boldsymbol{z}', \mathcal{P}) = \mathrm{nbn}(\boldsymbol{z}'', \mathcal{P}) = \boldsymbol{x}_i$. □

In one dimension the kissing number is two, so rule 2 in its current definition cannot be used for $n = 1$. Other exactly known values for the kissing number are $\tau_2 = 6$ and $\tau_3 = 12$. The number appears to grow exponentially, but the exact value is known only for a few other dimensions [27, p. 23].

In our use case we are actually not interested in the connected components of the obtained graph, but only in the sinks, i. e., nodes with an outdegree of zero. These points are used as starting points of local searches. The whole global stage is illustrated in Algorithm 11. It produces a sample $\mathcal{P}$ by either using an EA or a conventional sampling algorithm. In both cases, all evaluated points are passed to NBC, which selects a variable, typically small number of them. For each point, a local search is started and its final result recorded, before the next iteration starts again with the global stage.

We will now conduct an experiment with the described two-stage algorithm to determine its promising configurations.

5.2.2 Experiment on Clustering Methods

Research Question How does the global stage affect a clustering method for multimodal optimization?

Pre-experimental Planning In this experiment the global stage is more sophisticated than previously. It consists at least of a sampling algorithm and a clustering method and for some configurations, also a population-based optimization algorithm is part of it. All three components offer the possibility to regard archive points in the distance calculations. For example, NBC could be applied to a set $\mathcal{P} \cup \mathcal{A}$, where $\mathcal{A}$

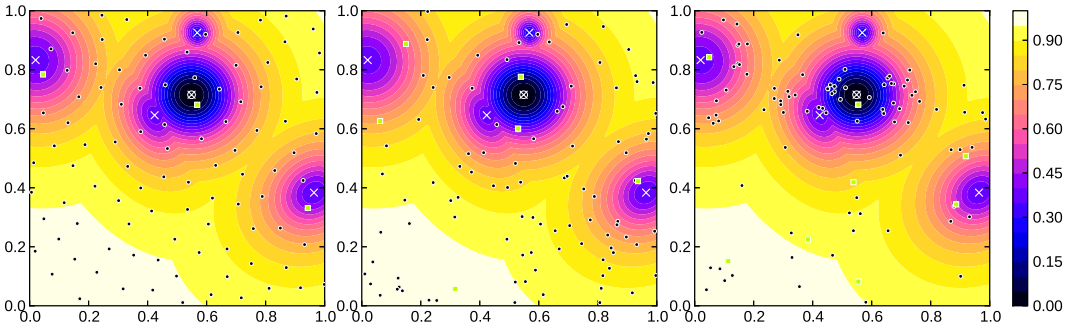
Algorithm 11 Global stage for clustering methods

Input: budget B, archive $\mathcal{A}$, objective function f
Output: variable number of starting points

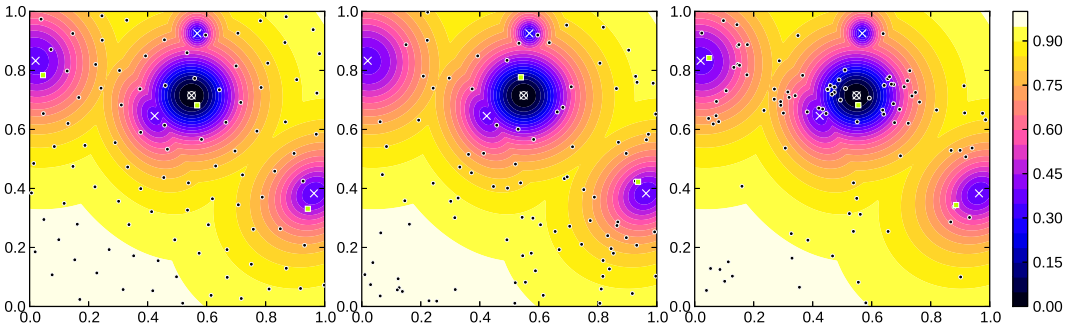
```
if samplingAlgorithm is an EA then
    choose μ ≪ B                          // population size μ should be a fraction of B
    P ← MmR(μ, A, n)                      // generate initial population
    P ← EA(P, A, B, f)                    // execute evolutionary algorithm, Alg. 9
else
    P ← samplingAlgorithm(B, A, n)        // spend whole budget on global sampling
    for all x ∈ P do
        evaluate f(x)                     // add objective values for NBC
    end for
end if
G ← NBC(P)                                // obtain clustering, Alg. 10
return points corresponding to sinks in G, sorted ascending by objective value
```

are the archive points, and from the resulting graph only the nodes belonging to newly sampled points $\mathcal{P}$ would come into consideration as starting points. This approach was taken with a different clustering by Ali and Storey [6], as a part of their "topographical" MLSL algorithm. In their case the archive contained previously encountered local optima. The results of this variant do look good, but the influence of the archive alone was not measured. Other investigations did not always observe benefits from using archives. In [155] the effect was negative, but the archive was used differently: it was initially empty and simply recorded all solutions encountered during optimization with Algorithm 9. For NEA2, which is more similar to our clustering method, no clear conclusions could be drawn [114, Sec. 6.1.3]. Testing all possible combinations of employing archive points or not in the three algorithm components of the global stage would multiply the computational effort of our full-factorial experiment by 2^3, so we make a preliminary choice to either disregard archive points completely ($\mathcal{A} = \emptyset$) or to regard them in the sampling and the optimization algorithm (if used), but not in NBC. The latter decision is taken under the assumption that it should not be made too difficult to determine starting points – which might happen by incorporating knowledge about local optima into NBC. A follow-up experiment may investigate further details if using an archive has proven to be promising in principle.

It is well known that NBC produces too many clusters if the point set deviates from uniformity. An example for this effect can be found in Figure 5.5a, where the uniformity decreases from left to right. The reason for this behavior is that outliers tend to be selected simply because of their large nearest-neighbor distances. A possible resolution to this problem is applying a correction factor to the distance threshold, as done by Preuss [114, Sec. 4.4]. Also MmR was designed with this problem in mind, taking care of it indirectly by reducing the variance of nearest-neighbor distances. Thus, problematic outliers in the sample are avoided from the



(a) Rules 1 and 2



(b) Rule 3

Figure 5.5: Subset selection with different NBC rules. $N = 100$ points are generated with MmR+PEC, SRS, and MNBS-EA (from left to right). Selected points are marked with green squares.

outset. However, we also have to deal with the very non-uniform samples produced by SNBS-EA and MNBS-EA. Therefore, we introduce a third rule that may replace rules 1 and 2. To describe this rule, we first need to define yet another measure related to nearest-better distances. Let $B(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathcal{X} \mid d(\boldsymbol{x}, \boldsymbol{y}) < r\}$ be the open ball of radius r around $\boldsymbol{x} \in \mathcal{X}$. $B_{\mathrm{nb}}(\boldsymbol{x}) := B(\boldsymbol{x}, d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}))$ shall be called the nearest-better ball of $\boldsymbol{x}$.

Definition 8. *The cardinality of the subset of $\mathcal{P}$ lying in the nearest-better ball of $\boldsymbol{x} \in \mathcal{X}$ is defined as*

$$\begin{aligned} N_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) &= |\mathcal{P} \cap B_{\mathrm{nb}}(\boldsymbol{x})| \\ &= |\{\boldsymbol{y} \in \mathcal{P} \mid d(\boldsymbol{x}, \boldsymbol{y}) < d_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P})\}| \,. \end{aligned}$$

It should be noted that besides the problem with non-uniformity, d_{nb} also suffers from a drift to the boundary, because boundary points have fewer neighbors and thus a potentially higher nearest-better distance. N_{nb} inherently employs edge correction, because the effects of the higher distance and lower number of neighbors cancel each other out. We will assume that the point $\boldsymbol{x}$ is also counted in its nearest-better ball, so $1 \leq N_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) \leq |\mathcal{P}|$. In our biological examples in the beginning of Section 5.2.1,

maximizing N_{nb} could be viewed as the strategy of the social animal, and maximizing d_{nb} as the strategy of the solitary one. N_{nb} is also an alternative for incorporation into Algorithm 8, but this investigation has to be deferred to some other time.

The distributions of N_{nb} and d_{nb} values are generally right-skewed, but the effect seems to be much stronger for the former. A quick visual inspection of experimental data suggests that the skewness of both distributions increases with the number of points and decreases with the number of optima. Figure 5.6 shows the effect for N_{nb} values of points generated by SRS on problems with random topology. This has adverse consequences for simple rules based on the mean of the sample (e. g., rule 1). The distribution of N_{nb} values is always restricted to the domain $\{1, \ldots, N\}$. If the number of optima increases, also the sample mean increases, making us select fewer points although we would rather want to select more points. Thus, finding a general rule is complicated considerably. We therefore apply a Box-Cox transformation [20]

$$\tilde{x} = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \lambda \neq 0, \\ \ln x & \lambda = 0 \end{cases}$$

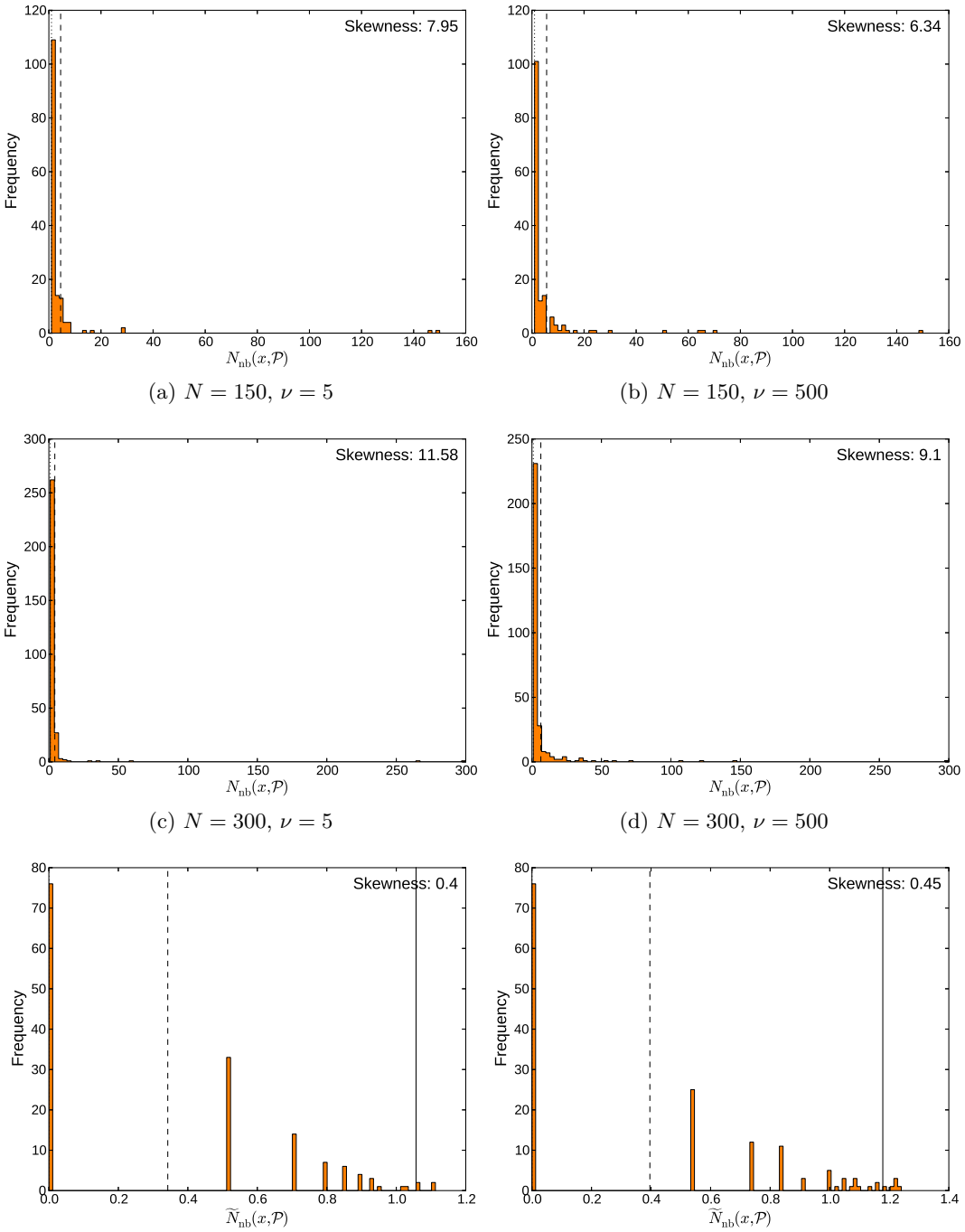
to the data, which optimizes the fit with the normal distribution by maximum likelihood estimation of the parameter λ. Now we have a less skewed distribution, shown in Figures 5.6e and 5.6f. However, also the common criterion to identify outliers based on the interquartile range as a robust measure of scale is unsuitable for our purposes. This criterion, usually used in Box-Whisker plots, identifies every point above $Q_3 + 1.5(Q_3 - Q_1)$ as (upper) outlier. Here, Q_1 and Q_3 denote the first and third quartile of the data. The resulting threshold is all too often close to the minimum in our case. To have a lower dependence on the actual distribution, we choose a threshold

$$\theta = \widetilde{N}_{\min} + 0.95(\widetilde{N}_{\max} - \widetilde{N}_{\min}) \,,$$

where $\widetilde{N}_{\min} = \min\{\widetilde{N}_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) \mid \boldsymbol{x} \in \mathcal{P}\}$, $\widetilde{N}_{\max} = \max\{\widetilde{N}_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) \mid \boldsymbol{x} \in \mathcal{P}\}$, and $\widetilde{N}_{\mathrm{nb}}$ means the Box-Cox-transformed values. Using this threshold θ, we can now formulate rule 3, which simply says to select a point $\boldsymbol{x}$ if $\widetilde{N}_{\mathrm{nb}}(\boldsymbol{x}, \mathcal{P}) > \theta$. To apply this selection rule, we even do not have to construct the spanning tree. It suffices to calculate the mentioned distances and count for each solution how many others are closer than the nearest better neighbor (for the best solution, this are all points). However, the rule *can* be incorporated into NBC. In this case, the conclusion of rule 3 says to remove the outgoing edge of the node corresponding to a point for which the antecedent holds. (There can be at most one such edge, due to the way G is constructed.)

Task The task in this experiment is largely identical to the task in Section 5.1.1. Again, the three indicators PR, F1P, and AHD are used for performance assessment and again the focus is on the global stage.

Setup Table 5.3 contains the high-level factors for this experiment. Some changes are made to the setup in comparison to Section 5.1.1, to reduce the computational



(a) $N = 150$, $\nu = 5$

(b) $N = 150$, $\nu = 500$

(c) $N = 300$, $\nu = 5$

(d) $N = 300$, $\nu = 500$

(e) Box-Cox transformation of Fig. 5.6a.

(f) Box-Cox transformation of Fig. 5.6b.

Figure 5.6: Histograms of N_{nb} on MPM2 instances in three dimensions. The dashed line marks the sample mean, the solid line the selection threshold. The median (dotted line) is always identical to the minimum of the data in these examples.

Table 5.3: High-level factors for the experiment in Section 5.2.

Factor	Type	Symbol	Levels
Problem topology	non-observable		{random, funnel}
Number of local optima	non-observable	ν	$\{5, 20, 100, 500\}$
Number of variables	observable	n	$\{2, 3, 5, 10, 20, 40\}$
Budget	observable	N_f	$\{10^3 n, 10^4 n\}$
Sampling algorithm	control		{SRS, MmR, SNBS-EA, MNBS-EA}
Archive	control	$\mathcal{A}$	$\{\emptyset, \mathcal{S}, \mathcal{S} \cup \widehat{\mathcal{O}}\}$
NBC rules	control		$\{\{1, 2\}, \{3\}\}$

effort. First of all, only MmR variants with $p = 2$ are considered, and the option MmR+PEC+REC is completely disregarded because of its great similarity to MmR+REC. On the other hand, two new candidates based on EAs are added to the sampling algorithms, meaning that we have six in total. The local optima approximations $\widehat{\mathcal{O}}$ alone are not considered as archive points any more, because of the bad performance they obtained previously. But now also the empty archive is a sensible factor level, because we are not only choosing a single starting point but draw a larger sample. In other words, there is a difference between SRS and MmR now, even when the archive is empty.

Two choices will be considered for the NBC rules, namely the original setup including rules 1 and 2, and the novel rule 3, which has to compete on its own. While the parameter ϕ of rule 1 could be subject to further tuning, it is set to $\phi = 2$, a relatively conservative value [114, Sec. 4.6.4] and the original default value [116]. A budget of $N_g = 50n$ is reserved for each call to the global stage. This is the same value as in [114, Sec. 6.2]. If an evolutionary algorithm is used in the global stage, an initial population of size $10n$ is drawn with MmR+REC. The EA is then run with parameters $\mu = 10n$, $\lambda = 2n$, and $\sigma = 10^{-1} \cdot 0.5(u - \ell)$, enabling 20 generations with said budget.

As a local search, only CMA-ES is used. This algorithm is chosen because it provided the highest reliability (best precision) in the previous experiment. Thus, it should provide the least confounding of the results. Other experimental settings remain unchanged (also on the low-level).

Results Figures 5.7 and 5.8 focus on the archive $\mathcal{A}$. The former illustrates the general effect of the archive points on AHD, depending on the number of variables and problem topology. The latter figure concentrates on a smaller subset of the data where the number of decision variables is $n = 10$ and $N_f = 10^4 n$. Here the influence of the individual sampling algorithms and NBC rules is examined. The other figures and the table are restricted to the data with $N_f = 10^4 n$ and $\mathcal{A} \neq \emptyset$. Figure 5.9 investigates the number of iterations conducted by the two-stage algorithm. Figure 5.10 presents the peak ratio data depending on dimension, sampling algorithm, and NBC

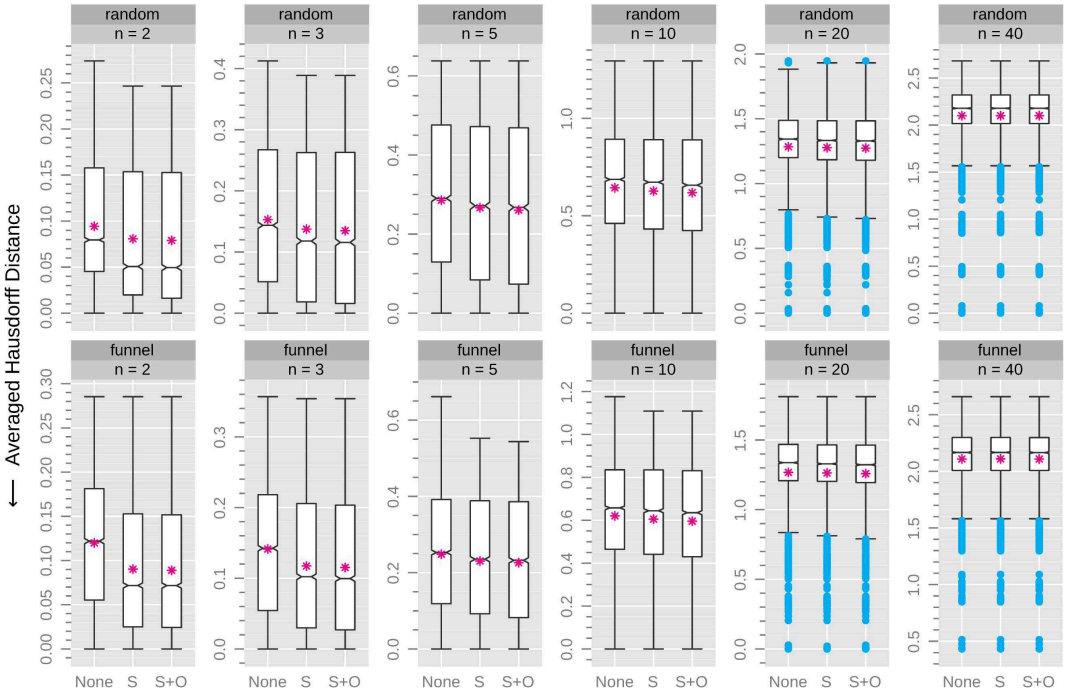


Figure 5.7: Influence of the used archive points on AHD.

rules. Table 5.4 contains the already familiar evaluation based on mean ranks and pairwise sign tests.

The median peak ratio of all variants of this CMA-ES-based clustering method is 0.18, while the median precision is 0.43.

Observations Figure 5.7 is used for a small factor screening. It shows that $\mathcal{A} = \emptyset$ clearly yields a worse performance and no interaction with dimension or topology can be detected. The impression is similar in Figure 5.8, where $\mathcal{A} = \mathcal{S} \cup \widehat{\mathcal{O}}$ obtains the best results. One can also see that the improved sampling alone does not yield much benefit, but the consideration of archive points is crucial. Thus, the further analysis is restricted to configurations for which $\mathcal{A} \neq \emptyset$. Hardly any significant differences can be found for the low budget $N_f = 10^3 n$ (not shown), so these runs are also excluded completely from the analysis to avoid a weakening of the effects. Also the differences between $\mathcal{S}$ and $\mathcal{S} \cup \widehat{\mathcal{O}}$ are relatively small, so we will again not differentiate between them in the comparison of global stages, to obtain larger sample sizes. Thus, we are left with $6 \cdot 2$ competitors in Table 5.4, leading to a Bonferroni correction of $c = 12$ in this experiment.

Figure 5.10 and Table 5.4 show that MmR-based sampling algorithms have an advantage over the other candidates in most cases. However, no clear preference can be determined for any of the edge corrections. MmR without edge correction is even the most successful sampling algorithm in high dimensions (see Figure 5.8 and Table 5.4). EA-based sampling algorithms perform better than SRS for $n \leq 5$,

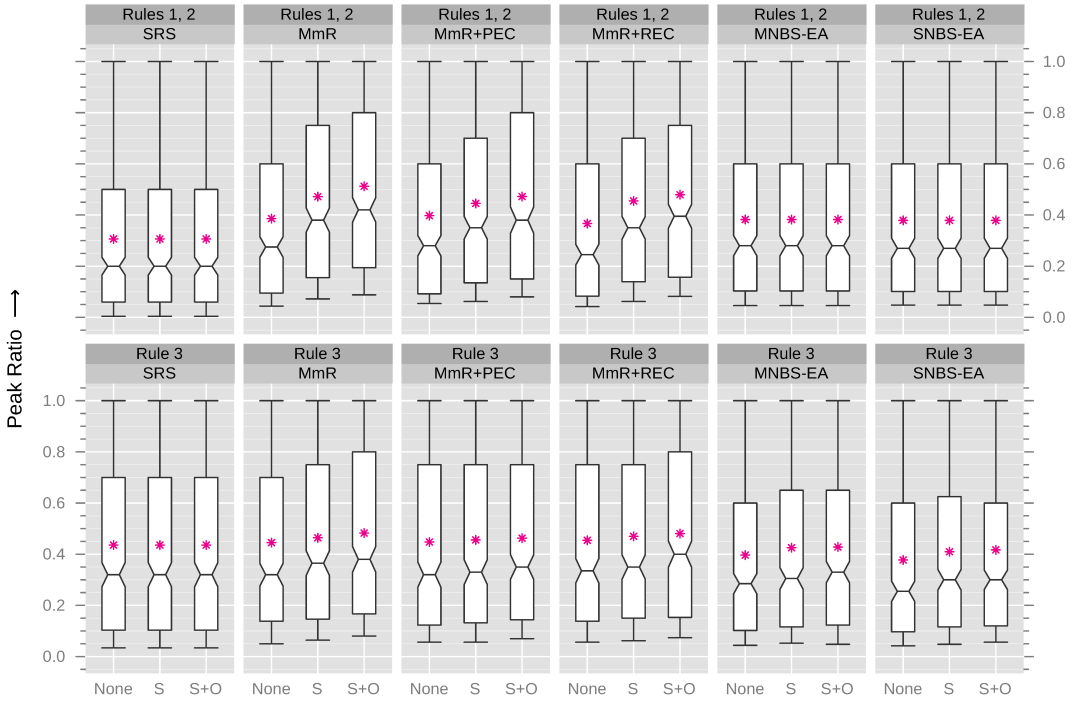


Figure 5.8: Influence of the used archive points on PR, for a subset of the data where the number of decision variables is $n = 10$ and $N_f = 10^4 n$.

but are the worst in higher dimensions. They are almost always better in combination with rule 3 than with rules 1 and 2. In this case, the improved performance corresponds to a slightly higher number of iterations (not shown). Otherwise, Figure 5.10 demonstrates that there is a high interaction between the NBC rules, the dimension, and the sampling algorithm. For example, SRS performs especially bad with rule 3 in low dimensions. Here, the bad results are associated with very high numbers of iterations (visible as outliers in Figure 5.9). AHD and PR are again in good agreement with each other, while F1P chooses a slightly different behavior.

Figure 5.9 shows that the number of iterations carried out for rule 3 is generally more dependent on the number of decision variables n than for rules 1 and 2: It is higher in low dimensions, but undergoes a steeper decline with increasing n. Contrariwise, rule 3 starts more local searches for $n \geq 5$ than rules 1 and 2 (not shown). Correspondingly, it can be observed on the whole data set that the number of iterations and the number of local searches are negatively correlated for $n \geq 10$ and $N_f = 10^4 n$ with a Pearson coefficient of -0.48.

Another interesting detail is that in terms of peak ratio, the performance of the clustering methods is slightly better on the random topologies (not shown), although funnel problems are usually regarded as easier to solve [4, 86]. However, keep in mind that we did not try to exploit the funnel structures.

Table 5.4: Aggregated results of the clustering methods experiment for $N_f = 10^4 n$ and disregarding configurations with $\mathcal{A} = \emptyset$. Column "D" denotes the number of pairwise tests where another algorithm was significantly better, "R" is the algorithm's mean rank. The best results are printed in bold.

n	Indicator	SRS {1,2}		SRS {3}		MmR {1,2}		MmR {3}		MmR +PEC {1,2}		MmR +PEC {3}		MmR +REC {1,2}		MmR +REC {3}		MNBS-EA {1,2}		MNBS-EA {3}		SNBS-EA {1,2}		SNBS-EA {3}	
		D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R	D	R
2	PR	10	10.0	11	11.7	**0**	4.6	**0**	4.7	**0**	**4.4**	3	4.9	**0**	4.6	**0**	4.8	6	6.9	6	7.0	7	7.3	6	7.0
2	F1P	10	10.8	11	12.0	3	4.6	**0**	**3.4**	3	4.7	1	3.8	4	5.0	**0**	3.6	6	7.2	8	7.6	6	7.5	9	8.1
2	AHD	10	10.0	11	11.8	**0**	4.7	**0**	4.7	**0**	**4.4**	1	5.0	**0**	4.7	1	4.9	6	6.8	6	6.9	7	7.2	6	7.0
3	PR	10	10.1	11	11.3	4	5.2	**0**	5.0	**0**	4.8	**0**	4.7	**0**	**4.7**	**0**	4.9	8	7.5	6	6.1	8	7.7	6	6.0
3	F1P	10	11.0	11	11.8	1	4.4	**0**	**3.7**	3	4.8	1	4.3	3	4.7	1	4.2	8	7.6	6	6.7	8	7.8	6	7.0
3	AHD	10	10.1	11	11.2	**0**	5.2	**0**	**4.7**	**0**	5.2	**0**	5.1	**0**	5.0	**0**	5.3	8	7.2	6	6.0	8	7.3	6	5.9
5	PR	11	10.7	10	10.4	4	5.3	1	4.7	2	5.0	1	4.9	1	4.7	**0**	**4.2**	8	8.3	6	5.5	8	8.3	7	5.9
5	F1P	11	11.2	10	11.2	**0**	**3.5**	3	4.3	1	4.0	5	5.5	1	3.6	4	4.6	8	8.6	6	6.2	8	8.5	7	6.8
5	AHD	11	10.7	10	10.4	3	5.3	1	4.8	2	5.1	2	5.2	**0**	4.7	**0**	**4.2**	8	8.1	5	5.6	8	8.0	6	5.9
10	PR	11	11.1	6	6.9	**0**	**3.5**	1	4.2	4	5.2	4	5.3	1	4.3	1	4.1	9	9.1	6	7.1	10	9.3	8	7.9
10	F1P	11	10.9	6	7.5	**0**	**2.3**	2	4.0	2	4.3	5	5.8	1	3.6	3	4.6	9	9.4	6	7.6	9	9.4	8	8.6
10	AHD	11	11.2	6	6.9	**0**	4.3	1	4.6	4	5.2	3	5.0	**0**	4.1	**0**	**3.9**	9	8.9	6	6.8	9	9.2	8	7.8
20	PR	11	9.6	2	4.8	**0**	4.5	**0**	**4.2**	5	6.0	1	4.6	6	7.2	1	4.6	8	8.1	6	7.2	9	8.6	9	8.5
20	F1P	8	8.7	2	5.2	**0**	**3.2**	1	3.9	1	4.8	2	5.0	6	6.1	3	5.3	8	8.9	7	8.0	9	9.4	9	9.5
20	AHD	11	10.0	**0**	4.8	1	5.3	**0**	4.8	5	6.2	**0**	4.6	6	6.9	**0**	**4.2**	7	7.8	6	7.1	8	8.2	8	8.1
40	PR	**0**	**5.0**	5	6.2	**0**	5.4	**0**	5.2	**0**	5.2	5	6.2	**0**	5.4	5	6.1	8	8.3	8	8.3	9	8.5	8	8.2
40	F1P	1	4.3	5	6.2	**0**	**3.8**	1	4.1	1	4.4	5	6.2	4	5.3	7	6.9	8	9.2	8	9.1	9	9.4	8	9.1
40	AHD	**0**	**5.0**	4	6.2	**0**	5.7	**0**	5.6	**0**	5.2	4	6.4	**0**	5.3	2	6.0	8	8.2	8	8.2	9	8.4	8	7.9

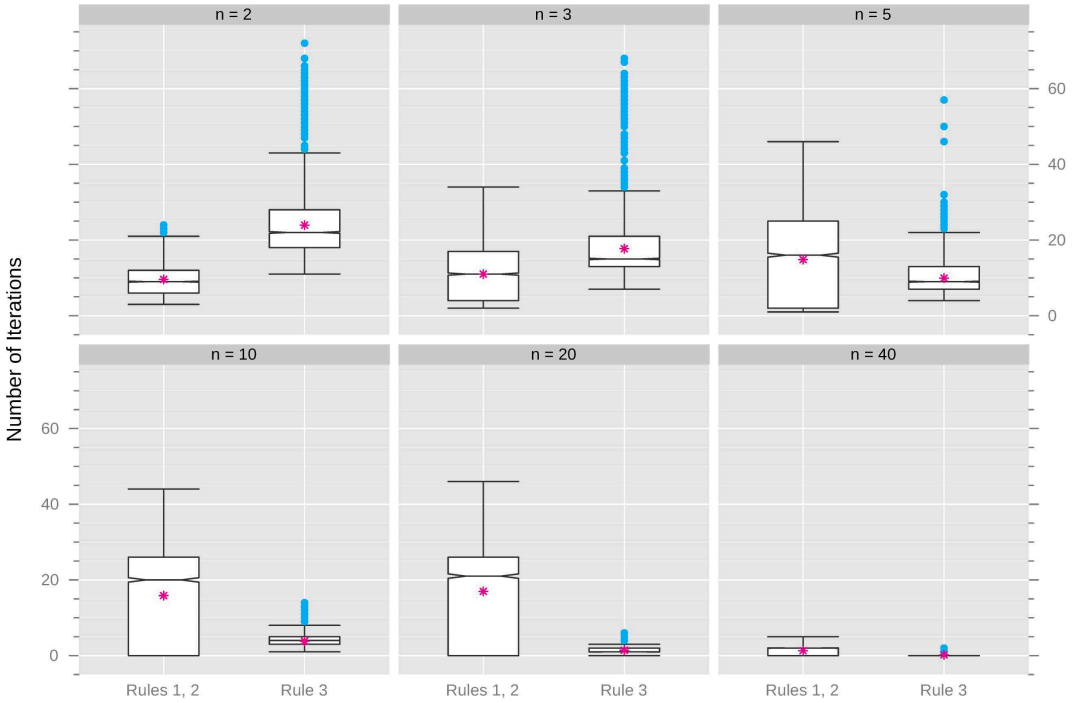


Figure 5.9: Number of completed iterations depending on the NBC rules. The data is restricted to configurations with $N_f = 10^4 n$ and $\mathcal{A} \neq \emptyset$.

Discussion The experiment shows that it is beneficial for a clustering method to record previous starting points and obtained local optima in an archive, and to exploit this information in subsequent iterations by sequential sampling. However, it is disappointing that the experiment leaves us with relatively many factor interactions we cannot explain (see, e. g., Figure 5.10 and Table 5.4), because this indicates there is an unconsidered factor in the background confounding the results. For example, it is surprising that edge correction did not turn out to be an important factor for success.

Based on Figure 5.5a, it was expected that EA-based sampling would not work well with rules 1 and 2, because many outliers are selected as starting points although they are not close to local optima. The experiment suggests that the latter statement is true also in dimensions higher than two, because the number of two-stage iterations in this case is very low (this detail of EA-based configurations is not shown here). Note that a low number of iterations means a high number of starting points must be proposed per iteration, because the total budget is fixed in our setup and the length of local searches should not vary that much. However, although the assumption holds, the actual performance of this configuration is not that bad. One might deduce that the quality of the starting points is not too important for performance, but there is also another effect in action: A lower number of iterations also means less function evaluations spent in the global stage, leaving more budget for the local searches.

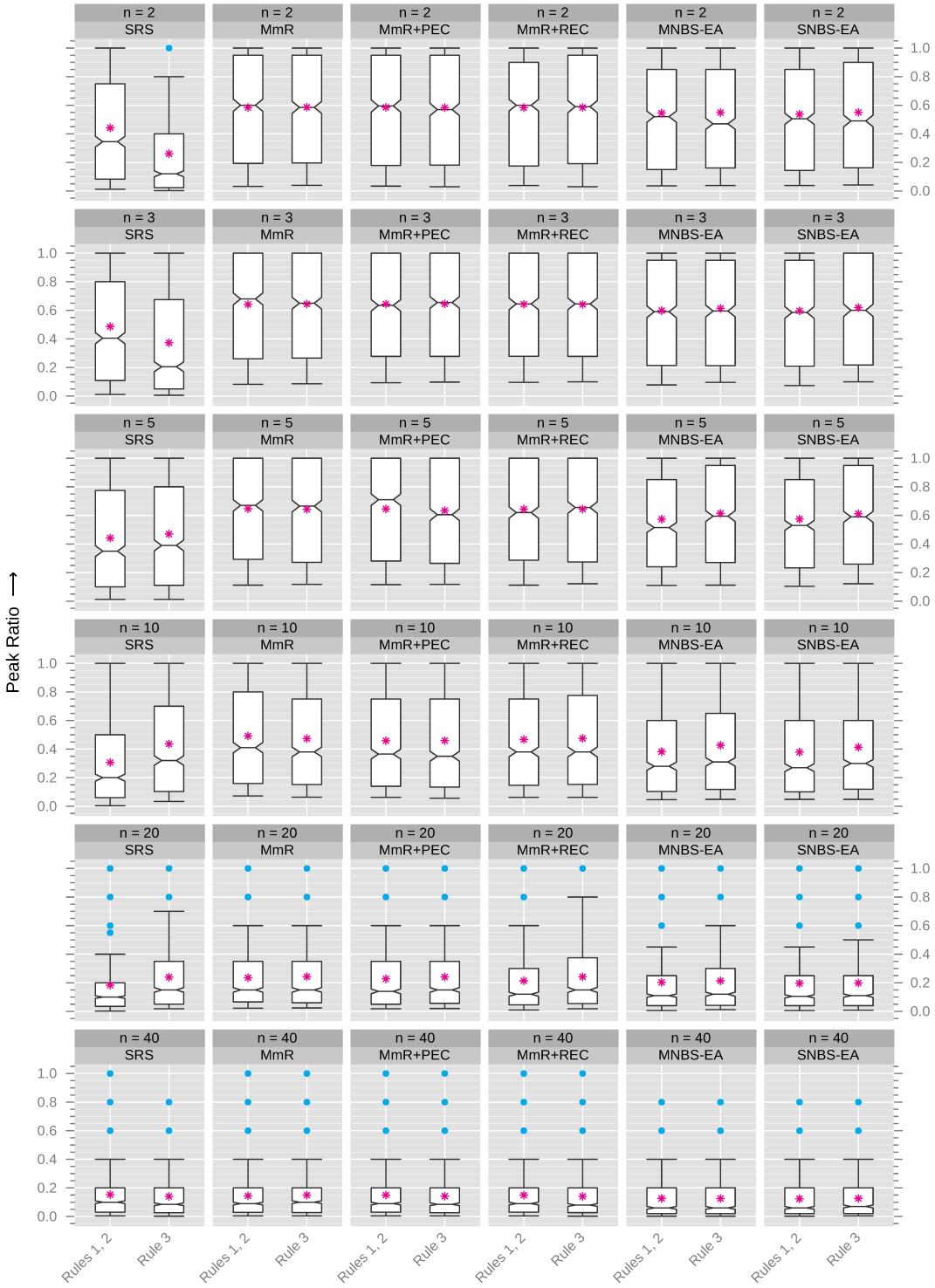


Figure 5.10: Peak ratio depending on dimension, sampling algorithm, and NBC rules. The data is restricted to values of $N_f = 10^4 n$ and $\mathcal{A} \neq \emptyset$.

The higher number of local searches of course has a positive influence on the size and thus the quality of the approximation set $\widehat{\mathcal{O}}$. It seems very difficult to balance these opposing demands of less costs of the global stage and better quality starting points. To simplify comparing the performances obtained with d_{nb} and N_{nb} values, it would be advisable to eliminate this interaction by fixing the number of selected points. However, in reality we want an adaptive rule for nearest-better clustering, selecting always an appropriate amount of starting points.

Rule 3 yields a slight improvement for the EA-based sampling, but not in general. Its performance so far is rather mixed. However, the problem seems to be the too simple method of determining the selection threshold, while N_{nb} itself still seems to be a promising measure. One could try to fix the problem by combining rules 1, 2, and 3, which would result in a consistently low but variable number of selected points, or one could add further correction factors to rule 3 as was done for the d_{nb} values [114, Sec. 4.4]. However, these attempts appear rather "kludgy". Instead, it seems advisable to design a new heuristic, operating on N_{nb} data, from scratch. This development probably should encompass trying to predict the number of optima of the problem, which is apparently an important factor for determining the right selection threshold θ (cf. Figure 5.6). If we take this route, we could go even further and conduct an extensive data-mining on the drawn sample to estimate as many problem features as possible. (As the calculations of different features can all be done on the same sample, this does not increase the costs in terms of function evaluations.) The obtained knowledge could then be employed to choose specially tailored local search algorithms. A first step into this direction is taken in the context of exploratory landscape analysis [73], where it is attempted to predict whether a problem has a funnel topology or not.

Not only the clustering, but the whole global stage could be subject to extensions. For example, we could include more points in the archive to give the sequential sampling more information. A relatively conservative approach would be to additionally include points that have been previously sampled by the global stage. This would ensure that subsequent samples are as dissimilar as possible from the previous ones. Furthermore, even all currently available data (also from the local searches) could be included in the archive, although this seems only feasible for short runs, as it increases the computational cost considerably. Apart from that, we could give up the batch-sequential character of the method and update the set of potential starting points directly after each local search. Immediately exploiting the information gained by a local search might help avoiding searches that will lead to already known optima. Similarly, a running local search could be stopped early if it is likely to converge to a known optimum.

It is surprising that there is hardly any difference between the sampling algorithms MNBS-EA and SNBS-EA, as they behave quite differently if used as optimization algorithms [158]. Perhaps the available budget of $N_g = 50n$ was too small for these differences to appear. There are many other parameters in our implementation that were chosen relatively ad hoc. Examples are ϕ, μ and λ of the EA-based samplings, or the parameter p of the distance function used in the global stage. The latter could

be especially influential in high dimensions, where the clustering method naturally has the most problems. Tuning these parameters should further improve the performance and may give additional insights into the mechanisms behind the observed interactions. Another open question is if MNBS-EA and SNBS-EA would obtain a relatively better performance in a global optimization task. This assumption suggests itself because the principle they are implementing here has been originally devised for global optimization [143, p. 95].

Note that the CMA-ES-based clustering method with SRS and NBC rules 1 and 2 investigated in this experiment is almost completely equivalent to NEA2. The only features of NEA2 omitted here are the correction factor for rule 1 [114, Sec. 4.4], dimension-dependent ϕ values [114, Sec. 4.6.4], and the initialization of the mutation strength of CMA-ES based on the estimated basin size [114, Sec. 6.1.3]. Due to this great similarity, it should be expectable that also NEA2 can be improved by replacing SRS by MmR, and using its capability of considering archive points.

Finally, we might draw a conciliatory conclusion, because the overall performance of the clustering method seems slightly better than that of the Restart-CMA-ES: the median peak ratio seems to be improved from 0.15 to 0.18 and the median precision from 0.4 to 0.43. However, in this setup the median values are not directly comparable, so the next section will investigate the performance differences between the two approaches in more detail.

5.3 Comparison of the Optimization Algorithms

The use of identical environmental factors in Sections 5.1.1 and 5.2.2 allows us to compare the two general approaches with each other. But to be as fair as possible, the control factors appearing in both experiments have to be restricted to the intersection of their tested levels. This concerns of course the local search algorithm, because Nelder-Mead and L-BFGS-B were not used in the latter experiment. As sampling algorithms, only MmR, MmR+PEC, MmR+REC (each with $p = 2$), and SRS come into question. Finally, the possibilities for archive points are reduced to $\mathcal{S}$ and $\mathcal{S} \cup \widehat{\mathcal{O}}$.

Figure 5.11 compares the AHD values obtained by the remaining configurations of the clustering method (CM) and the restarted local search (RLS) for budgets of $N_f = 10^3 n$. Figure 5.12 illustrates the same aspect for $N_f = 10^4 n$. We can now finally take a closer look at the environmental factors, although the results are again not divided into random and funnel topologies, because no significant interaction could be detected. One can see that the clustering method has an advantage in most cases. However, the higher the dimension n, the higher the number of optima ν, and the lower the budget, the better performs restarted local search in comparison: in forty dimensions it is always superior and in twenty dimensions if the budget is only $10^3 n$. Also when the number of optima is $\nu = 500$, the application of the more complicated clustering method is only worthwhile if the budget is large. The results for PR and F1P, which are not shown here visually, are in accordance with the AHD values and favor CM under the same conditions. A very similar pattern appears if

we regard the number of local searches (not shown): with the exception of $n = 20$, the better performance of CM is always associated with a higher number of local searches. Finally, we use an analysis of variance (ANOVA) on the original and on rank-transformed data to compare the influence of the factors "two-stage approach" and "sampling algorithm". These ANOVAs indicate that the effect of the two-stage approach is stronger for $N_f = 10^3 n$, while the sampling algorithm has more influence on performance for $N_f = 10^4 n$. This result could be obtained independently of the used performance measure (AHD, PR, or F1P) and the used rank transformation (none, RT-1, or RT-2 [26]).

This whole investigation shows that clustering methods do provide a significant improvement over restarted local search for many relevant problem configurations. However, they exhibit the same limitations as NEA2 in terms of problem dimensionality [114, Sec. 6.4]. For the lower budget, the area where clustering methods make sense ends somewhere between ten and twenty dimensions. For the higher budget the break-even point is probably just over twenty dimensions. Repeating conclusions from Section 5.2.2, it may be possible to push this point up by using different distance functions (e. g., Manhattan distance or even lower p values), by using more points in the archive, and by selecting starting points for local searches more intelligently. However, even when clustering becomes futile, using a sequential sampling with improved distribution is *always* advisable, because it is not associated with any cost in terms of objective function evaluations. Thus we will in the worst case simply fall back to the performance of random uniform sampling.

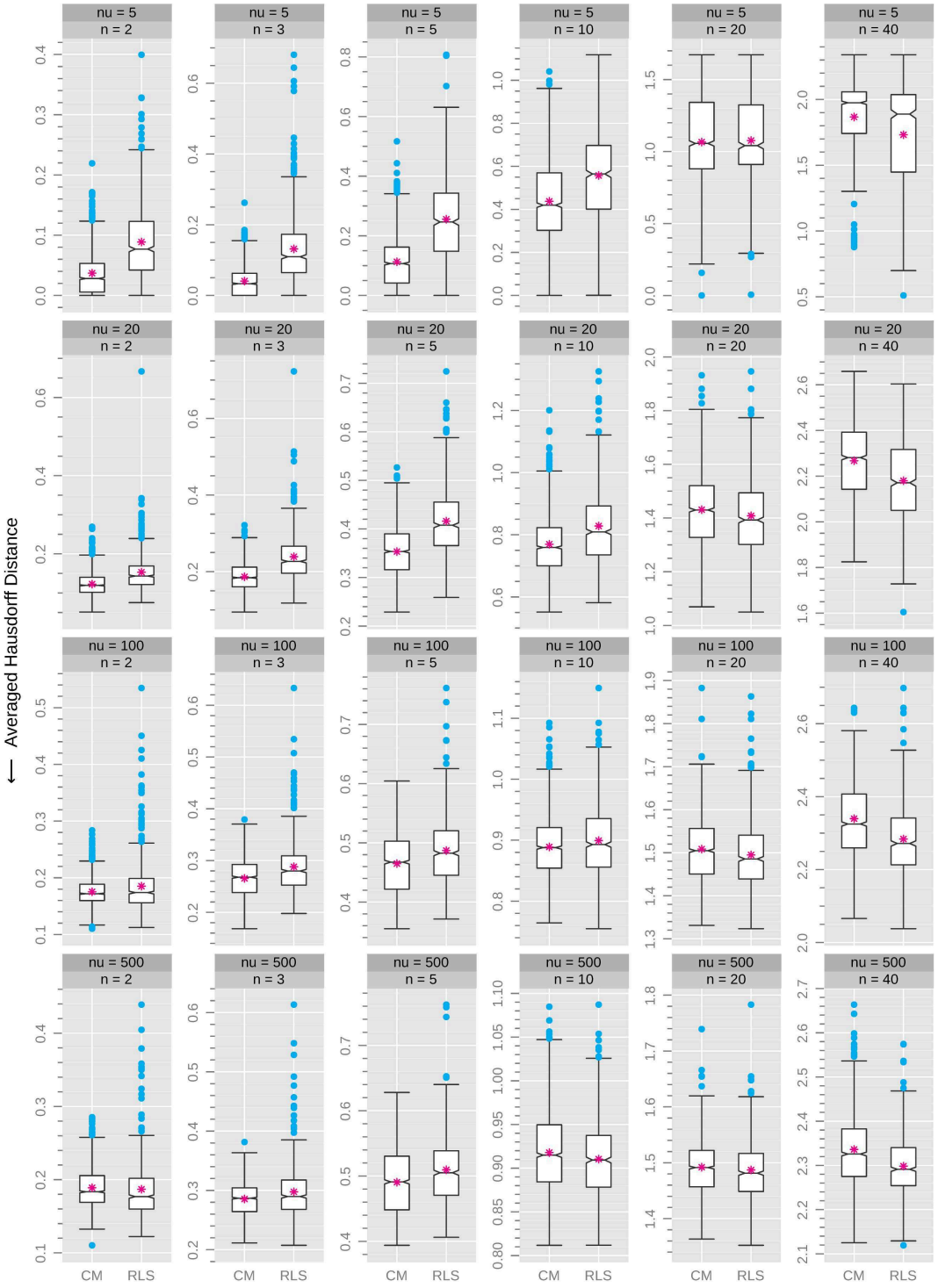


Figure 5.11: Comparison of the two-stage approaches with a budget of $N_f = 10^3 n$.

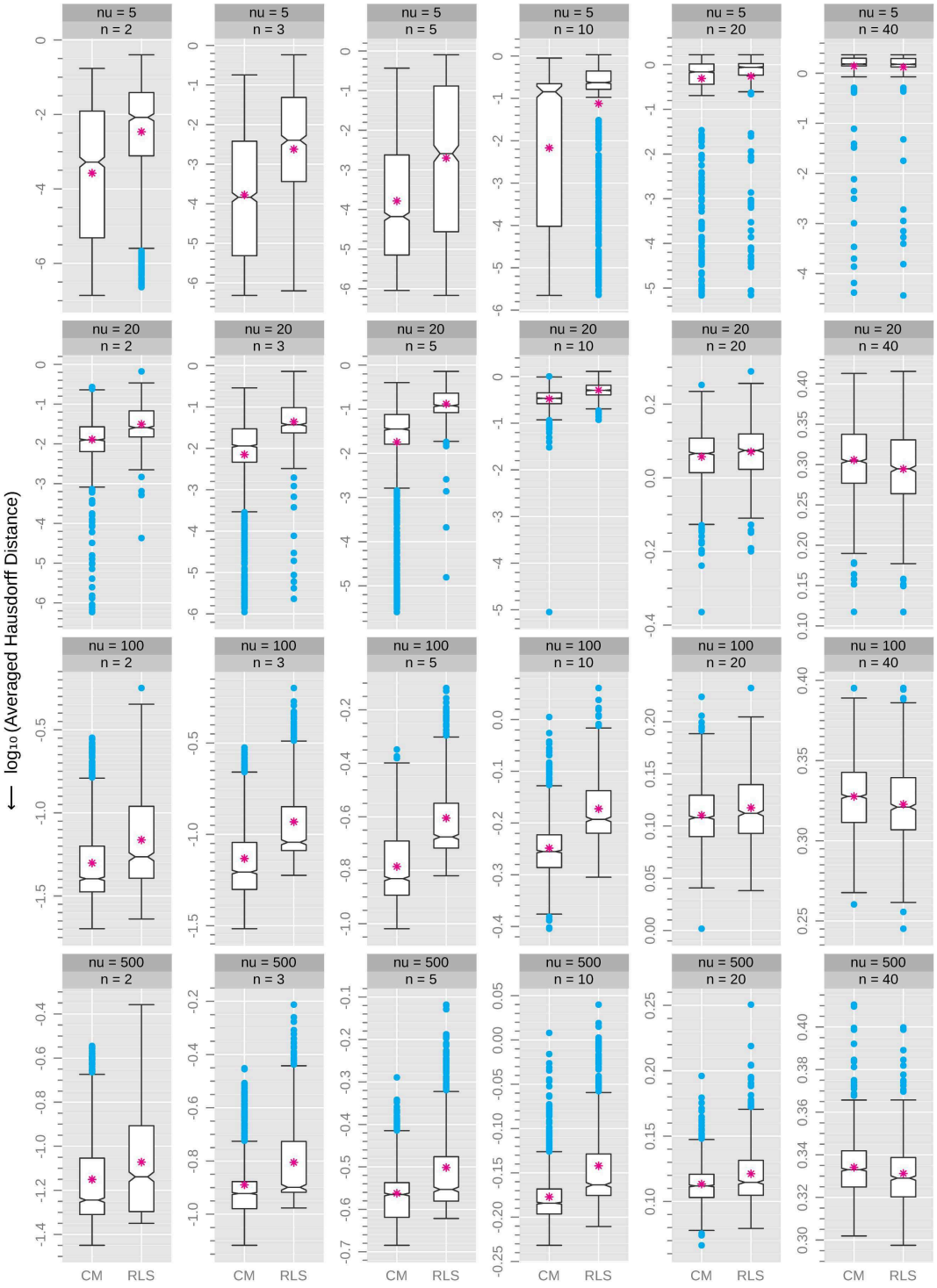


Figure 5.12: Comparison of the two-stage approaches with a budget of $N_f = 10^4 n$.

6 Conclusions and Outlook

In this work, two-stage optimization algorithms were investigated on box-constrained, multimodal problems. A comprehensive survey of performance measures was given and reasonably large experiments were conducted. In the course thereof we made the following contributions:

Compilation of quality indicators and some upper bounds The collection of quality indicators for multimodal optimization as published in [155] was extended by a larger section for diversity indicators. An axiomatic treatment of these indicators was refined and possible similarities between them were pointed out. For three problem-dependent quality indicators, upper bounds based on the diversity indicator *covering radius* were found. To the authors best knowledge, this is the first time that any upper bounds are formulated for indicators in this application area at all.

More sophisticated test problem generator With the *multiple peaks model 2* (MPM2), an advancement of existing test problem generators [47, 115] was built. The improvements pertain to the initialization of the problem instance, which now guarantees to exactly produce a certain number of local optima, and the shape of the peaks, which are now modeled by a function that avoids the disadvantages of Gallagher's Gaussians [47] and the function used by Preuss and Lasarczyk [115]. Concern regarding the latter is that the worst objective value of the problem cannot be bounded easily, while the former are criticized for their steep slope [122].

New summary characteristics to assess point sets The *mean distance to the boundary* was identified as an important characteristic of point sets. Additionally, the distance between the center of mass of a point set and the centroid of the region of interest was established as a measure for the point set's balance. Both measures are nothing more but conventional Monte Carlo estimates. They can be computed in $O(nN)$, which is a big advantage, because practically all summary characteristics used to date have run times of at least $O(nN \log N)$. As the expected values of these measures for random uniform points are useful reference values, they were derived analytically. Although neither characteristic is sufficient for measuring uniformity, they both can quickly detect certain deviations from it. This way, they can contribute to a richer description of experimental designs and point sets in general, for example as part of statistical tests for complete spatial randomness [63, pp. 83–98]. Especially the distance to the boundary may be a good characteristic to describe the main

difference between minimax and maximin designs. A working hypothesis was that the distance to the boundary and the covering radius of a low-discrepancy point set are related. This connection would be especially useful, because the distance to the boundary is an inexpensive feature, while the covering radius can only be calculated via a costly Delaunay tesselation [118]. Although no formal proof could be given, our experimental results do support this conjecture. This can be seen, e. g., by regarding Figure 4.12 together with Proposition 1.

Run time improvements for sampling algorithms While surveying existing sampling algorithms, two run time improvements could be found. The first case refers to *improved latin hypercube sampling* (ILHS), whose worst-case run time could be decreased from $O(nN^3)$ to $O(nN^2)$, while even attaining a slightly better output quality. The second case is the *part and select algorithm* (PSA), which could be shown to possess an average-case run time of $O(Mn \log_2 N)$ in our application area, while the worst-case run time is $O(MnN)$. To achieve the result, a small modification was necessary, namely replacing the array used as the internal data structure by a binary heap.

Definition of a new sampling algorithm and comparison to existing ones The *maximin reconstruction algorithm* (MmR) was designed to maximize the minimal distance in a point set under consideration of existing points and edge effects. These two items represent the novel aspects of the algorithm. In an experimental comparison with established sampling algorithms, not the uniformity but the edge correction proved to be the crucial factor for obtaining the best performance in high dimensions (see Figure 4.12). Based on this experiment, there seem to be no circumstances where replacing random uniform sampling by another truly uniform sampling with improved distribution (as, e. g., quasirandom point sets, randomized FILHS+PEC, or MmR+PEC) would yield any performance deterioration. This conjecture is in accordance with results in different contexts [100].

A more robust pruning rule for NBC We developed a new selection heuristic ("rule 3") for nearest-better clustering (NBC). This rule is based on the number N_{nb} of solutions that are closer than the nearest better neighbor of a solution (see Definition 8). In contrast to the existing approaches, the new key figure is insensitive to outliers in the sample, because it only rewards a solution for being the best in a crowded neighborhood.

Improved two-stage optimization algorithms The new sampling algorithm MmR was used to improve two-stage optimization algorithms for the task of multimodal optimization. Two general classes of two-stage algorithms were tested, namely *restarted local search* and *clustering methods*. In the two corresponding experiments, several problem and algorithm parameters were varied and several quality indicators were used for assessment, making the results very general and reliable. In these experiments, the focus was on the global stages of

the algorithms, which either employed random uniform sampling, MmR, or an evolutionary algorithm to obtain the point sample. Edge corrected variants of MmR always yielded significantly better results than simple random uniform sampling, independent of the local search algorithm, the problem topology, and the two-stage approach. Apparently, the main improvement stems from the consideration of an archive of previously used starting points and found optima when determining the next starting point. The goal is to restart far away from these points. To the authors best knowledge, it is the first time in the context of optimization that this adaptive behavior has been tested in a general form for arbitrary points. Other works investigated existing points at most in the clustering [120, 6]. On a side note, previous results that the clustering method is the better algorithm for low-dimensional problems [114, Sec. 6.4] could be verified, but the experiments also show that the used sampling algorithm often has more influence than the two-stage method.

Despite the progress made in this work, several important research questions do remain open. First of all, developing better diversity indicators is still an important topic. For example, various discrepancies fail to give a realistic assessment of uniformity in certain cases [126, pp. 146–148][91]. Our own experiments indicate that the unanchored L_2 discrepancy degenerates to a maximin criterion when $N < 2^n$ (see Figure 4.8c). The core deficiency of discrepancy seems to be that it compares real numbers (the volumes of subsets) with rational numbers (the fraction of points in a subset). As in high dimensions volumes of subsets can be very small (i. e., much smaller than $1/N$), the error we make by approximating these volumes by multiples of $1/N$ must be inevitably large. The wrap-around discrepancy by Hickernell [60] may be a viable alternative, although it would be important to know the expected value of this measure for random uniform point sets. Another, radical, solution would be to abandon discrepancy completely and to use a different diversity indicator instead. That this is possible has already been shown [29]. However, to reward exact uniformity, some edge correction has to be incorporated also into any other candidate indicator. We have already done this implicitly by considering the edge-corrected separation distance in MmR.

In several fields, researchers have to recognize that what they are optimizing is actually not uniformity. In the worst case this is due to the usage of spread indicators [30, 86], which do not reflect our intuitive understanding of diversity because they do not enforce any repulsion between points [131, 144, 95]. In design of computer experiments, approaches maximizing the minimal distance among points are popular [66]. These model repulsion, but produce non-uniform designs because of boundary effects and the curse of dimensionality. Also the more sophisticated maximum-entropy designs share the same property [21]. It must be noted that while latin hypercube designs (LHD) alleviate the effect, they are not the universal remedy either. In particular maximin LHDs, Audze-Eglājs LHDs for $n > 2$, and Morris-Mitchell LHDs for $q < n$ can produce unforeseen deviations from uniformity. In Figure 4.2, the former two exhibit a drift towards the boundary. The latter two actually employ an

Table 6.1: Approximate relations between concepts in design of experiments, spatial statistics, and quasi-Monte Carlo methods.

	$\bar{d}_{\mathcal{B}} < \delta_n$	$\bar{d}_{\mathcal{B}} \approx \delta_n$	$\bar{d}_{\mathcal{B}} > \delta_n$
Experimental designs	maximin [66]	uniform [42]	minimax [66]
Quasi-Monte Carlo terms		low discrepancy [108, p. 14]	low dispersion [108, p. 148]
Edge correction	none	periodic [63, p. 184]	reflection [63, p. 184]
Optimality type	D [66]	U [42]	G [66]
Measures/bounds		integration error [108, pp. 18–22]	max mean squared prediction error [126, pp. 170–171]
Examples of sampling methods	conventional grid, MmR (Sec. 4.4.2)	LHS [94], SRS, quasirandom sequences, MmR+PEC	CVT [37], PSA [125], Sukharev grid [136], MmR+REC

identical optimization criterion [118], with Audze-Eglãjs being a special case of the other with $q = 2$. If $q < n$, the drift to the boundary is even more severe than for a pure maximin LHD. However, this does not necessarily mean that these designs are completely unsuitable, as their properties may offer lower mean squared prediction errors in meta-modeling [35]. Apart from that, periodic edge correction [63, p. 184] is a straightforward technique to adjust all maximin approaches to true uniformity. Also LHDs generated by the algorithm of Beachkofski and Grandhi [13] do not seem to suffer from the drift to the boundary (see Figure 4.11), although more experiments are necessary to verify this observation.

Ideally, connections between point process statistics, experimental designs, quasirandom sequences, and centroidal Voronoi tessellations would be regarded more, because strong relations seem to exist between the concepts. Table 6.1 summarizes these considerations regarding the different terms and research areas. For example, minimax designs actually are point sets with low dispersion/low covering radius. Also centroidal Voronoi tessellations seem to play a special role for this measure.

A lot of uncertainty exists regarding the test problems. In this work we considered the problem of essentially unconstrained optimization, where all optima are located in the interior of the search space [143, p. 10][119]. If we assume that the search space is (arbitrarily) determined by a decision maker as a subset of a larger preimage of a multimodal function, this scenario seems very unlikely, at least in high dimensions. Consider, for example, two hypercubes $\mathcal{X}_1 \subset \mathcal{X}_2$ in 40 dimensions with edge lengths 1 and 1.1. While the former has a volume of 1, the latter has a volume of ap-

proximately 45. If we search only the smaller hypercube, this means that 97% of the larger hypercube are excluded in this example. Under the assumption of a uniform distribution of optima, it seems unrealistic that no optima are located directly on the boundary of $\mathcal{X}_1$, because if $\mathcal{X}_2$ is 45 times larger, then also a correspondingly large number of optima must be located outside of $\mathcal{X}_1$. By clipping the region of interest to $\mathcal{X}_1$, factitious optima appear on its bounds through the clipped outer attraction basins. Another case where optima may be (theoretically) located on bounds are periodic search spaces [45, 1], which usually arise when decision variables represent angles. However, these problems can be equally treated as unconstrained problems by making the search space a torus. Sampling with periodic edge correction (see Section 4.4.2) and local search using the repair method Baldwinian wrapping [154] should obtain very favorable results in this case.

Also the assumption that the number of optima does not increase with dimension [143, p. 11] should be challenged. However, modeling highly multimodal problems with the multiple peaks model chosen in this work is infeasible due to a function evaluation's linear run time in the number of optima. The only viable approach of constructing an MPM2 instance with an exponential number of optima seems to be as a sum of independent lower dimensional functions. This would of course result in a separable problem, which is not very difficult for optimization [109]. Non-separable problems with an exponential number of optima in n are frequently created by rotating a separable problem [122], because this is easily done. However, the author is not aware of any real-world appearances of such problems. An example for non-separable real-world problems with a (possibly) exponential number of optima are molecular conformation problems [148]. These usually possess one or even multiple funnel structures, which seem to stem from *isospectral symmetries* in the problem formulation [106]. Thanks to these properties, the effort to solve them remains acceptable despite the strong multimodality [4, 106]. The computational cost of the objective function is quadratic in the number of modeled atoms and there is only one problem instance per dimension unless the problem definition is modified in a certain way [106].

All these considerations regarding the realism of test problems have never been treated together in a systematic way, to the authors best knowledge. As the design of appropriate test problems and carrying out the corresponding experiments would be quite laborious and time-consuming, we refrained from doing this analysis ourselves. However, when such experiments are finally carried out, it should be expected that quite different results will be obtained on some of the mentioned other problem classes. Especially the chosen edge correction may interact with the positions of the optima. For separable problems it should be important to use samples with good low-dimensional projections [109].

To bridge the gap between model-based and model-free optimization algorithms regarding computational complexity, one might consider two-stage algorithms with a model-based global stage. Such an approach should be somewhere in between the pure model-free and the pure model-based algorithm regarding the number of objective function evaluations and the computational overhead of the optimization

algorithm. Another reason to use a meta-model might be greatly varying activities of the individual decision variables. In this case, the model could be used to learn an appropriate (weighted) distance function [69].

Finally, a topic where a lot of room for improvement is suspected is the clustering procedure, which was not the main focus of our work. Only nearest-better clustering was tested as a part of the optimization algorithm in Section 5.2. For a complete picture, it should be compared to MLSL [120] and topographical clustering [6]. Regarding NBC, the figure N_{nb} seems to be better suited than d_{nb}, although the current heuristics used to determine the number of returned clusters are still chosen rather ad hoc. So it should be easy to enhance this part of the algorithm. Other research directions as the further improvement of the local search procedures, fine-tuning the interplay of the local and global stage, and optimization of the algorithm parameters in general are rather self-evident.

Glossary

$\mathbf{0}$	$(0, \ldots, 0)^\top$.
$\mathbf{1}$	$(1, \ldots, 1)^\top$.
$\mathcal{A}$	archive.
$\mathcal{B}$	boundary of $\mathcal{X}$.
$\mathcal{C}$	cluster.
$\bar{\boldsymbol{c}}_{\mathcal{P}}$	measured center of mass.
$\boldsymbol{c}_{\mathcal{X}}$	centroid of the search space.
D_N	L_∞ discrepancy.
D_N^*	L_∞ star discrepancy.
d_N	covering radius.
$d_{\bar{\boldsymbol{c}}\boldsymbol{c}}$	measured distance between center of mass of a point set and the centroid of the search space.
$\bar{d}_{\mathcal{B}}$	mean distance to the boundary.
δ_n	expected distance to the boundary for uniformly distributed points.
Δ_p	averaged Hausdorff distance.
$d_{\max}$	maximal possible distance.
d_{nb}	nearest-better distance.
d_{nn}	nearest-neighbor distance.
$\epsilon_{N,n}$	expected distance between center of mass of a random uniform point set and the centroid of the search space.
$\hat{f}^*$	estimate of global optimum.
f^*	global optimum.
f	objective function.
f_{dup}	duplication factor.
$\mathcal{F}$	non-dominated front.
$\mathcal{F}_t$	filial generation at time t.
Γ	gamma function.
$\mathbf{I}$	identity matrix.
$\boldsymbol{\ell}$	lower bounds.
λ	number of offspring in an evolutionary algorithm.
m	number of objectives.
μ	number of parents in an evolutionary algorithm.
N	number of points.
$\mathbb{N}$	natural numbers.

n	number of variables.
N_f	total number of function evaluations.
N_g	number of function evaluations for the global stage.
N_{nb}	number of solutions in the nearest-better ball.
ν	number of optima.
$\widehat{\mathcal{O}}$	set of approximated local optima positions.
$\mathcal{O}$	set of local optima positions.
ω	modulus of continuity.
ϕ	threshold for NBC rule 1.
$\mathbf{R}$	correlation or rotation matrix.
$\mathbb{R}$	real numbers.
ρ	relative radius.
$\mathcal{S}$	set of starting points.
$\mathbf{\Sigma}$	covariance matrix.
σ	standard deviation, mutation strength in EAs.
T_N	L_2 discrepancy.
T_N^*	L_2 star discrepancy.
τ_n	kissing number.
$\boldsymbol{u}$	upper bounds.
$\hat{\boldsymbol{x}}^*$	estimate of global optimum position.
$\boldsymbol{x}^*$	global optimum position.
$\mathcal{X}$	search space, region of interest.

AD	Average Distance.
AHD	Averaged Hausdorff Distance.
AID	Average of Inverse Distances.
ANOVA	Analysis of Variance.
AOV	Average Objective Value.

BBOB	Black-box Optimization Benchmarking.
BI	Basin Inaccuracy.
BR	Basin Ratio.

CEC	Congress on Evolutionary Computation.
CM	Clustering Method.
CMA-ES	Covariance Matrix Adaptation Evolution Strategy.
CR	Covering Radius.
CRN	Common Random Numbers.
CVT	Centroidal Voronoi Tesselation.

DACE	Design and Analysis of Computer Experiments.
DISC	Discrepancy.

EA	Evolutionary Algorithm.
ELA	Exploratory Landscape Analysis.
FILHS	Fast ILHS.
GH	Generalized Halton Sequence.
ILHS	Improved LHS.
IMSE	Integrated Mean Squared Error.
LHD	Latin Hypercube Design.
LHS	Latin Hypercube Sampling.
MBO	Model-based Optimization.
MD	Minimal Distance.
MLSL	Multilevel Single Linkage.
MMO	Multimodal Optimization.
MmR	Maximin Reconstruction.
MNBS	Multiobjective Nearest-better Selection.
MOI	Multiobjective Infill.
MPM2	Multiple Peaks Model 2.
NBC	Nearest-better Clustering.
NEA2	Niching Evolutionary Algorithm 2.
PD	Peak Distance.
PDNN	Product of Distances to Nearest Neighbors.
PEC	Periodic Edge Correction.
PI	Peak Inaccuracy.
PR	Peak Ratio.
PSA	Part and Select Algorithm.
REC	Reflection Edge Correction.
RLS	Restarted Local Search.
RNG	Random Number Generator.
ROI	Region of Interest.
RSA	Random Sequential Adsorption.
RT	Rank Transformation.
SD	Sum of Distances.
SDCM	Sum of Distances to Center of Mass.
SDNN	Sum of Distances to Nearest Neighbor.
SNBS	Single-objective Nearest-better Selection.
SPD	Solow-Polasky Diversity.

SRS	Simple Random Uniform Sampling.
TP	Twin Property.
UB	Union of Balls.
WD	Weitzman Diversity.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Bernardetta Addis, Andrea Cassioli, Marco Locatelli, and Fabio Schoen. A global optimization method for the design of space trajectories. *Computational Optimization and Applications*, 48(3):635–652, 2011.

[2] Bernardetta Addis and Sven Leyffer. A trust-region algorithm for global optimization. *Computational Optimization and Applications*, 35(3):287–304, 2006.

[3] Bernardetta Addis and Marco Locatelli. A new class of test functions for global optimization. *Journal of Global Optimization*, 38(3):479–501, 2007.

[4] Bernardetta Addis, Marco Locatelli, and Fabio Schoen. Disk packing in a square: A new global optimization approach. *INFORMS Journal on Computing*, 20(4):516–524, 2008.

[5] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pages 420–434. Springer, 2001.

[6] Montaz M. Ali and Colin Storey. Topographical multilevel single linkage. *Journal of Global Optimization*, 5(4):349–358, 1994.

[7] Barry C. Arnold, Narayanaswamy Balakrishnan, and Haikady N. Nagaraja. *A First Course in Order Statistics*. Wiley, 1992.

[8] P. Audze and Vilnis Eglājs. New approach to the design of multifactor experiments. *Problems of Dynamics and Strengths*, 35:104–107, 1977. (in Russian).

[9] Anne Auger and Nikolaus Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1777–1784, 2005.

[10] Anne Auger and Olivier Teytaud. Continuous lunches are free plus the design of optimal optimization algorithms. *Algorithmica*, 57(1):121–146, 2010.

[11] Thomas Bartz-Beielstein, Marco Chiarandini, Luís Paquete, and Mike Preuss, editors. *Experimental Methods for the Analysis of Optimization Algorithms*. Springer, 2010.

[12] Stuart J. Bates, Johann Sienz, and D.S. Langley. Formulation of the Audze–Eglais uniform latin hypercube design of experiments. *Advances in Engineering Software*, 34(8):493–506, 2003.

[13] Brian Beachkofski and Ramana Grandhi. Improved distributed hypercube sampling. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. AIAA paper 2002-1274, American Institute of Aeronautics and Astronautics, 2002.

[14] David Beasley, David R. Bull, and Ralph R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[15] Richard A. Becker, William S. Cleveland, and Ming-Jen Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996.

[16] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[17] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In Catriel Beeri and Peter Buneman, editors, *Database Theory — ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 1999.

[18] Patrick Billingsley. *Probability and Measure*. Wiley, third edition, 1995.

[19] Bernd Bischl, Simon Wessing, Nadja Bauer, Klaus Friedrichs, and Claus Weihs. MOI-MBO: Multiobjective infill for parallel model-based optimization. In Panos M. Pardalos, Mauricio G.C. Resende, Chrysafis Vogiatzis, and Jose L. Walteros, editors, *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 173–186. Springer, 2014.

[20] George E. P. Box and David R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[21] Dizza Bursztyn and David M. Steinberg. Comparison of designs for computer experiments. *Journal of Statistical Planning and Inference*, 136(3):1103–1119, 2006.

[22] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[23] Michael Carter. A toolbox for quasirandom simulation. *The Mathematica Journal*, 13, 2011. `https://dx.doi.org/doi:10.3888/tmj.13-21`.

[24] Timothy M. Chan. Klee's measure problem made easy. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 410–419, 2013.

[25] Carlos A. Coello Coello and Nareli Cruz Cortés. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190, 2005.

[26] William J. Conover and Ronald L. Iman. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician*, 35(3):124–129, 1981.

[27] John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1988.

[28] Giuseppe Cuccu, Faustino Gomez, and Tobias Glasmachers. Novelty-based restarts for evolution strategies. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 158–163, 2011.

[29] Steven B. Damelin, Fred J. Hickernell, David L. Ragozin, and Xiaoyan Zeng. On energy, discrepancy and group invariant measures on measurable subsets of euclidean space. *Journal of Fourier Analysis and Applications*, 16(6):813–839, 2010.

[30] Emilie Danna and David L. Woodruff. How to select a small set of diverse solutions to mixed integer programming problems. *Operations Research Letters*, 37(4):255–260, 2009.

[31] Swagatam Das and Ponnuthurai N. Suganthan. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical report, Jadavpur University, India and Nanyang Technological University, Singapore, 2010. Updated 20th February 2011.

[32] François-Michel De Rainville, Christian Gagné, Olivier Teytaud, and Denis Laurendeau. Evolutionary optimization of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, 22(2):9:1–9:25, 2012.

[33] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[34] Enrique del Castillo. *Process Optimization*, volume 105 of *International Series in Operations Research & Management Science*. Springer, 2007.

[35] Holger Dette and Andrey Pepelyshev. Generalized latin hypercube design for computer experiments. *Technometrics*, 52(4):421–429, 2010.

[36] Carola Doerr, Michael Gnewuch, and Magnus Wahlström. Calculation of discrepancy measures and applications. In William Chen, Anand Srivastav, and Giancarlo Travaglini, editors, *A Panorama of Discrepancy Theory*, volume 2107 of *Lecture Notes in Mathematics*, pages 621–678. Springer, 2014.

[37] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[38] Agoston E. Eiben and Mark Jelasity. A critical note on experimental research methodology in EC. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 582–587, 2002.

[39] Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[40] Michael T. M. Emmerich, André H. Deutz, and Johannes W. Kruisselbrink. On quality indicators for black-box level set approximation. In Emilia Tantar, Alexandru-Adrian Tantar, Pascal Bouvry, Pierre Del Moral, Pierrick Legrand, Carlos A. Coello Coello, and Oliver Schütze, editors, *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, volume 447 of *Studies in Computational Intelligence*, pages 157–185. Springer, 2013.

[41] Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[42] Kai-Tai Fang, Dennis K. J. Lin, Peter Winker, and Yong Zhang. Uniform design: Theory and application. *Technometrics*, 42(3):237–248, 2000.

[43] Henri Faure and Christiane Lemieux. Generalized Halton sequences in 2008: A comparative study. *ACM Transactions on Modeling and Computer Simulation*, 19(4):15:1–15:31, 2009.

[44] Steffen Finck, Nikolaus Hansen, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[45] Roger Fletcher and Michael J. D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.

[46] Alexander Forrester, András Sóbester, and Andy Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

[47] Marcus Gallagher and Bo Yuan. A general-purpose tunable landscape generator. *IEEE Transactions on Evolutionary Computation*, 10(5):590–603, 2006.

[48] Marco Gaviano, Dmitri E. Kvasov, Daniela Lera, and Yaroslav D. Sergeyev. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Transactions on Mathematical Software*, 29(4):469–480, 2003.

[49] Jay B. Ghosh. Computational aspects of the maximum diversity problem. *Operations Research Letters*, 19(4):175–181, 1996.

[50] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 41–49. Lawrence Erlbaum Associates, Inc., 1987.

[51] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[52] Salvatore Greco, Kathrin Klamroth, Joshua D. Knowles, and Günter Rudolph. Understanding complexity in multiobjective optimization (Dagstuhl Seminar 15031). *Dagstuhl Reports*, 5(1):96–163, 2015.

[53] Nikolaus Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395. ACM, 2009.

[54] Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[55] Nikolaus Hansen and Stefan Kern. Evaluating the CMA evolution strategy on multimodal test functions. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 282–291. Springer, 2004.

[56] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[57] Douglas P. Hardin and Edward B. Saff. Discretizing manifolds via minimum energy points. *Notices of the American Mathematical Society*, 51(10):1186–1194, 2004.

[58] Garrett Hardin. The competitive exclusion principle. *Science*, 131(3409):1292–1297, 1960.

[59] Paul Heckbert. Color image quantization for frame buffer display. *SIGGRAPH Computer Graphics*, 16(3):297–307, 1982.

[60] Fred J. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67(221):299–322, 1998.

[61] Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[62] Bart G.M. Husslage, Gijs Rennen, Edwin R. Dam, and Dick Hertog. Space-filling latin hypercube designs for computer experiments. *Optimization and Engineering*, 12(4):611–630, 2011.

[63] Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan. *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley, 2008.

[64] János Izsák and László Papp. A link between ecological diversity indices and measures of biodiversity. *Ecological Modelling*, 130(1–3):151–156, 2000.

[65] Stephen Joe and Frances Y. Kuo. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, 2008.

[66] 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.

[67] Norman L. Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions*, volume 2. Wiley, second edition, 1994.

[68] Norman L. Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions*, volume 1. Wiley, second edition, 1994.

[69] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[70] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[71] Lili Ju, Qiang Du, and Max Gunzburger. Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. *Parallel Computing*, 28(10):1477–1500, 2002.

[72] Peter Jäckel. *Monte Carlo Methods in Finance*. Wiley, 2002.

[73] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. Detecting funnel structures by means of exploratory landscape analysis. In *Proceedings of the 2015 conference on Genetic and evolutionary computation*, GECCO '15. ACM, 2015. (to appear).

[74] Victor Klee. Can the measure of $\cup[a_i, b_i]$ be computed in less than $O(n \log n)$ steps? *The American Mathematical Monthly*, 84(4):284–285, 1977.

[75] Joshua 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.

[76] Sergei Kucherenko and Yury Sytsko. Application of deterministic low-discrepancy sequences in global optimization. *Computational Optimization and Applications*, 30(3):297–318, 2005.

[77] Ching-Chung Kuo, Fred Glover, and Krishna S. Dhir. Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185, 1993.

[78] Ares Lagae and Philip Dutré. A comparison of methods for generating poisson disk distributions. *Computer Graphics Forum*, 27(1):114–129, 2008.

[79] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[80] Xiadong Li, Andries Engelbrecht, and Michael G. Epitropakis. Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization. Technical report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.

[81] Tianjun Liao, Daniel Molina, Marco A. Montes de Oca, and Thomas Stützle. A note on bound constraints handling for the IEEE CEC'05 benchmark function suite. *Evolutionary Computation*, 22(2):351–359, 2014.

[82] Mattias Liefvendahl and Rafał Stocki. A study on algorithms for optimization of latin hypercubes. *Journal of Statistical Planning and Inference*, 136(9):3231–3247, 2006.

[83] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[84] Marco Locatelli and Fabio Schoen. Global optimization based on local searches. *4OR*, 11(4):301–321, 2013.

[85] Helena R. Lourenço, Olivier C. Martin, and Thomas Stützle. Iterated local search: Framework and applications. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 363–397. Springer, 2010.

[86] Monte Lunacek and Darrell Whitley. The dispersion metric and the CMA evolution strategy. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 477–484. ACM, 2006.

[87] Monte Lunacek, Darrell Whitley, and Andrew Sutton. The impact of global structure on search. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 498–507. Springer, 2008.

[88] Pierre L'Ecuyer. Comparison of point sets and sequences for quasi-Monte Carlo and for random number generation. In Solomon W. Golomb, Matthew G. Parker, Alexander Pott, and Arne Winterhof, editors, *Sequences and Their Applications – SETA 2008*, volume 5203 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2008.

[89] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, California, 1967. University of California Press.

[90] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[91] Jiří Matoušek. On the L2-discrepancy for anchored boxes. *Journal of Complexity*, 14(4):527–556, 1998.

[92] Bertil Matérn. *Spatial Variation*, volume 36 of *Lecture Notes in Statistics*. Springer, 1986.

[93] Catherine McGeoch. Analyzing algorithms by simulation: Variance reduction techniques and simulation speedups. *ACM Computing Surveys*, 24(2):195–212, 1992.

[94] Michael D. McKay, Richard J. Beckman, and William J. Conover. A 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.

[95] Thorsten Meinl, Claude Ostermann, and Michael R. Berthold. Maximum-score diversity selection for early drug discovery. *Journal of Chemical Information and Modeling*, 51(2):237–247, 2011.

[96] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 829–836. ACM, 2011.

[97] Olaf Mersmann, Mike Preuss, and Heike Trautmann. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature – PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 73–82. Springer, 2011.

[98] Olaf Mersmann, Heike Trautmann, Boris Naujoks, and Claus Weihs. Benchmarking evolutionary multiobjective optimization algorithms. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[99] Brad L. Miller and Michael J. Shaw. Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *International Conference on Evolutionary Computation*, pages 786–791, 1996.

[100] Don P. Mitchell. Consequences of stratified sampling in graphics. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 277–280. ACM, 1996.

[101] Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley, 4th edition, 1997.

[102] Rachael Morgan and Marcus Gallagher. Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements. *IEEE Transactions on Evolutionary Computation*, 18(3):456–461, 2014.

[103] William J. Morokoff and Russel E. Caflisch. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279, 1994.

[104] Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. Technical Report ORNL/TM-12045, Oak Ridge National Laboratory, Engineering Physics and Mathematics Division, 1992.

[105] Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995.

[106] Christian L. Müller and Ivos F. Sbalzarini. Energy landscapes of atomic clusters as black box optimization benchmarks. *Evolutionary Computation*, 20(4):543–573, 2012.

[107] John A. Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[108] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1992.

[109] Erich Novak and Klaus Ritter. Global optimization using hyperbolic cross points. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *State of the Art in Global Optimization*, volume 7 of *Nonconvex Optimization and Its Applications*, pages 19–33. Springer, 1996.

[110] Mahamed G. H. Omran, Salah al Sharhan, Ayed Salman, and Maurice Clerc. Studying the effect of using low-discrepancy sequences to initialize population-based optimization algorithms. *Computational Optimization and Applications*, 56(2):457–480, 2013.

[111] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.

[112] Petr Pošík and Waltraud Huyer. Restarted local search algorithms for continuous black box optimization. *Evolutionary Computation*, 20(4):575–607, 2012.

[113] Mike Preuss. Reporting on experiments in evolutionary computation. Technical Report CI-221/07, University of Dortmund, Collaborative Research Center 531, 2007.

[114] Mike Preuss. *Multimodal Optimization by Means of Evolutionary Algorithms*. Springer, 2015. (in print).

[115] Mike Preuss and Christian Lasarczyk. On the importance of information speed in structured populations. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 91–100. Springer, 2004.

[116] Mike Preuss, Lutz Schönemann, and Michael Emmerich. Counteracting genetic drift and disruptive recombination in $(\mu +/, \lambda)$-EA on multimodal fitness landscapes. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 865–872. ACM, 2005.

[117] Mike Preuss and Simon Wessing. Measuring multimodal optimization solution sets with a view to multiobjective techniques. In Michael Emmerich, Andre Deutz, Oliver Schütze, Thomas Bäck, Emilia Tantar, Alexandru-Adrian Tantar, Pierre Del Moral, Pierrick Legrand, Pascal Bouvry, and Carlos A. Coello, editors, *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, volume 227 of *Advances in Intelligent Systems and Computing*, pages 123–137. Springer, 2013.

[118] Luc Pronzato and Werner G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[119] Alexander H. G. Rinnooy Kan and Gerrit T. Timmer. Stochastic global optimization methods part I: Clustering methods. *Mathematical Programming*, 39(1):27–56, 1987.

[120] Alexander H. G. Rinnooy Kan and Gerrit T. Timmer. Stochastic global optimization methods part II: Multi level methods. *Mathematical Programming*, 39(1):57–78, 1987.

[121] Günter Rudolph. On correlated mutations in evolution strategies. In R. Männer and B. Manderick, editors, *Parallel problem solving from nature 2*, pages 105–114. Elsevier, 1992.

[122] Jani Rönkkönen, Xiaodong Li, Ville Kyrki, and Jouni Lampinen. A generator for multimodal test functions with multiple global optima. In Xiaodong Li, Michael Kirley, Mengjie Zhang, David Green, Vic Ciesielski, Hussein Abbass, Zbigniew Michalewicz, Tim Hendtlass, Kalyanmoy Deb, KayChen Tan, Jürgen Branke, and Yuhui Shi, editors, *Simulated Evolution and Learning*, volume 5361 of *Lecture Notes in Computer Science*, pages 239–248. Springer, 2008.

[123] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, November 1989.

[124] Yuki Saka, Max Gunzburger, and John Burkardt. Latinized, improved LHS, and CVT point sets in hypercubes. *International Journal of Numerical Analysis and Modeling*, 4(3-4):729–743, 2007.

[125] Shaul Salomon, Gideon Avigad, Alex Goldvard, and Oliver Schütze. PSA – a new scalable space partition based selection algorithm for MOEAs. In Oliver Schütze, Carlos A. Coello Coello, Alexandru-Adrian Tantar, Emilia Tantar, Pascal Bouvry, Pierre Del Moral, and Pierrick Legrand, editors, *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 137–151. Springer, 2013.

[126] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*. Springer, 2003.

[127] Fabio Schoen. Two-phase methods for global optimization. In Panos M. Pardalos and H. Edwin Romeijn, editors, *Handbook of Global Optimization*, volume 62 of *Nonconvex Optimization and Its Applications*, pages 151–177. Springer, 2002.

[128] Colas Schretter, Leif Kobbelt, and Paul-Olivier Dehaye. Golden ratio sequences for low-discrepancy sampling. *Journal of Graphics Tools*, 16(2):95–104, 2012.

[129] Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012.

[130] Peter Shirley. Discrepancy as a quality measure for sample distributions. In F. H. Post and W. Barth, editors, *Proceedings of Eurographics '91*, pages 183–194. Elsevier, 1991.

[131] Andrew R. Solow and Stephen Polasky. Measuring biological diversity. *Environmental and Ecological Statistics*, 1(2):95–103, 1994.

[132] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[133] Erwin Stinstra, Dick den Hertog, Peter Stehouwer, and Arjen Vestjens. Constrained maximin designs for computer experiments. *Technometrics*, 45(4):340–346, 2003.

[134] Catalin Stoean, Mike Preuss, Ruxandra Stoean, and Dumitru Dumitrescu. Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation*, 14(6):842–864, 2010.

[135] Ponnuthurai N. Suganthan, Nikolaus Hansen, Jing Jane Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, May 2005. `http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2005/Tech-Report-May-30-05.pdf`.

[136] Aleksandr G. Sukharev. Optimal strategies of the search for an extremum. *USSR Computational Mathematics and Mathematical Physics*, 11(4):119–137, 1971.

[137] Atsuo Suzuki and Zvi Drezner. The p-center location problem in an area. *Location Science*, 4(1–2):69–82, 1996.

[138] András Sóbester, Stephen J. Leary, and Andy J. Keane. A parallel updating scheme for approximating and optimizing high fidelity computer simulations. *Structural and Multidisciplinary Optimization*, 27(5):371–383, 2004.

[139] René Thomsen. Multimodal optimization using crowding-based differential evolution. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1382–1389, 2004.

[140] Aimo Törn. A search-clustering approach to global optimization. In *Towards Global Optimization 2*, pages 49–62. North-Holland, 1978.

[141] Aimo Törn, Montaz M. Ali, and Sami Viitanen. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*, 14(4):437–447, 1999.

[142] Aimo Törn and Sami Viitanen. Topographical global optimization. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Recent Advances in Global Optimization*, Princeton Series in Computer Sciences, pages 384–398. Princeton University Press, 1992.

[143] Aimo Törn and Antanas Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, 1989.

[144] Tamara Ulrich, Johannes Bader, and Lothar Thiele. Defining and optimizing indicator-based diversity measures in multiobjective search. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature – PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 707–717. Springer, 2010.

[145] Tamara Ulrich, Johannes Bader, and Eckart Zitzler. Integrating decision space diversity into hypervolume-based multiobjective search. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 455–462. ACM, 2010.

[146] Tamara Ulrich and Lothar Thiele. Maximizing population diversity in single-objective optimization. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 641–648. ACM, 2011.

[147] Rasmus K. Ursem. Multinational evolutionary algorithms. In Peter J. Angeline, editor, *Proceedings of the Congress of Evolutionary Computation (CEC 99)*, volume 3, pages 1633–1640. IEEE Press, 1999.

[148] David J. Wales and Harold A. Scheraga. Global optimization of clusters, crystals, and biomolecules. *Science*, 285(5432):1368–1372, 1999.

[149] Shijie J. Wan, S. K. Michael Wong, and Przemyslaw Prusinkiewicz. An algorithm for multidimensional data clustering. *ACM Transactions on Mathematical Software*, 14(2):153–162, 1988.

[150] Xiaoqun Wang and Ian H. Sloan. Low discrepancy sequences in high dimensions: How well are their projections distributed? *Journal of Computational and Applied Mathematics*, 213(2):366–386, 2008.

[151] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98, Proceedings of 24th International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 194–205. Morgan Kaufmann, 1998.

[152] Martin L. Weitzman. On diversity. *The Quarterly Journal of Economics*, 107(2):363–405, 1992.

[153] William J. Welch, Robert. J. Buck, Jerome Sacks, Henry P. Wynn, Toby J. Mitchell, and Max D. Morris. Screening, predicting, and computer experiments. *Technometrics*, 34(1):15–25, 1992.

[154] Simon Wessing. Repair methods for box constraints revisited. In Anna I. Esparcia-Alcázar, editor, *Applications of Evolutionary Computation*, volume 7835 of *Lecture Notes in Computer Science*, pages 469–478. Springer, 2013.

[155] Simon Wessing and Mike Preuss. On multiobjective selection for multimodal optimization. Algorithm Engineering Report TR14-2-001, Technische Universität Dortmund, Department of Computer Science, Chair of Algorithm Engineering, December 2014.

[156] Simon Wessing, Mike Preuss, and Günter Rudolph. When parameter tuning actually is parameter control. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 821–828. ACM, 2011.

[157] Simon Wessing, Mike Preuss, and Günter Rudolph. Niching by multiobjectivization with neighbor information: Trade-offs and benefits. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 103–110, 2013.

[158] Simon Wessing, Mike Preuss, and Heike Trautmann. Stopping criteria for multimodal optimization. In Thomas Bartz-Beielstein, Jürgen Branke, Bogdan Filipič, and Jim Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, volume 8672 of *Lecture Notes in Computer Science*, pages 141–150. Springer, 2014.

[159] Xiaolin Wu and Ian H. Witten. A fast k-means type clustering algorithm. Technical report, Department of Computer Science, University of Calgary, Canada, 1985.

[160] Eckart Zitzler, Joshua Knowles, and Lothar Thiele. Quality assessment of pareto set approximations. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 373–404. Springer, 2008.