Evolving Sampling Strategies for One-Shot Optimization Tasks

Jakob Bossek, Carola Doerr, Pascal Kerschke, Aneta Neumann, Frank Neumann

Evolving Sampling Strategies for One-Shot Optimization Tasks

Jakob Bossek[1], Carola Doerr[2], Pascal Kerschke[3], Aneta Neumann[1], and Frank Neumann[1]

[1] The University of Adelaide, Adelaide, Australia
jakob.bossek@adelaide.edu.au
[2] Sorbonne Université, CNRS, LIP6, Paris, France
[3] University of Münster, Münster, Germany

Abstract. One-shot optimization tasks require to determine the set of solution candidates prior to their evaluation, i.e., without possibility for adaptive sampling. We consider two variants, classic one-shot optimization (where our aim is to find at least one solution of high quality) and one-shot regression (where the goal is to fit a model that resembles the true problem as well as possible). For both tasks it seems intuitive that well-distributed samples should perform better than uniform or grid-based samples, since they show a better coverage of the decision space. In practice, quasi-random designs such as Latin Hypercube Samples and low-discrepancy point sets are indeed very commonly used designs for one-shot optimization tasks.
We study in this work how well low star discrepancy correlates with performance in one-shot optimization. Our results confirm an advantage of low-discrepancy designs, but also indicate the correlation between discrepancy values and overall performance is rather weak. We then demonstrate that commonly used designs may be far from optimal. More precisely, we evolve 24 very specific designs that each achieve good performance on one of our benchmark problems. Interestingly, we find that these specifically designed samples yield surprisingly good performance across the whole benchmark set. Our results therefore give strong indication that significant performance gains over state-of-the-art one-shot sampling techniques are possible, and that evolutionary algorithms can be an efficient means to evolve these.

Keywords: One-Shot Optimization · Regression · Fully Parallel Search · Surrogate-Assisted Optimization · Continuous Optimization

1 Introduction

When dealing with costly to evaluate problems under high time pressure, a decision maker is often left with the only option of evaluating a few possible decisions in parallel, in the hope that one of them proves to be a reasonable alternative. The problem of designing strategies that guarantee a fair chance of

finding a good solution is studied under the term *one-shot optimization*. One-shot optimization is studied in numerous variants and contexts, including classic Operations Research [12] and numerical analysis [26,27]. Most recently, one-shot optimization has gained momentum in the context of Machine Learning applications, including hyper-parameter optimization for deep neural networks and for heuristic optimization techniques [4,2,9].

We study in this work two variants of one-shot optimization tasks, classic one-shot optimization and one-shot regression. In **classic one-shot optimization,** n solution candidates are evaluated in parallel. We only care about the best one of them, x^{best} and measure its simple regret $f(x^{\text{best}}) - \inf f$. In **one-shot regression**, in contrast, we use all n evaluated samples to build an approximation $\hat{f}$ of the actual, unknown function f. The objective is to determine a surrogate $\hat{f}$ which resembles f as well as possible. The quality of $\hat{f}$ is measured, for example, by the mean squared error (MSE) $\sum_{x \in X} (\hat{f}(x) - f(x))^2 / |X|$. One-shot regression is also studied under the term *global surrogate modeling* [12].

Several works, in particular the one of Bousquet et al. [4] and the more recent work by Cauwet et al. [9], show that quasi-random designs of low discrepancy are more suitable for the classic one-shot optimization task than i.i.d. uniform samples or grid search. The overall recommendation propagated in [4] are randomly scrambled Hammersley point sets with random shifts. Other low-discrepancy point sets also perform well in the experiments reported there. Also for one-shot regression quasi-random constructions such as Latin Hypercube Samples (LHS [30]) and again low-discrepancy point sets [12,28] are quite common, leaving us with *the question if there is a correlation between the discrepancy of a point set and its performance in one-shot optimization.* If such a correlation existed, one could hope to find even better one-shot designs by searching for point sets of small discrepancy – a problem that is much easier (yet very hard [14]) to address than the original one-shot optimization problem. Interestingly, no such direct comparison has been attempted in the literature, although several works have investigated the suitability of various sampling designs for one-shot regression, see [28,12] for examples and although such a correlation is well known to hold in the context of numerical integration, via the Koksma-Hlawka inequality [24,21].

We compare five different experimental designs, three generalized Halton point sets, one LHS construction, and i.i.d. uniform sampling, see Sec. 2 for more details. Our test bed are the 24 noiseless BBOB functions [19,20], a standard benchmark set for numerical black-box optimization, which covers a wide range of different problems encountered in real-world optimization. We focus on *star discrepancy* [14] as diversity measure for the point sets, since this is the one that also appears in the mentioned Koksma-Hlawka inequality. For the regression task, we compare four standard regression techniques, support vector machines (SVMs) [11], decision trees [7], random forests [6], and Kriging [10], see Sec. 3.

1.1 Summary of Results

Results for Standard Sampling Designs In the context of *classic one-shot optimization,* our experiments confirm the superiority of low-discrepancy point sets over random sampling. However, no clear correlation could be identified between the star discrepancy value of a point set and its performance as one-shot optimizer, somewhat refuting our hope that point sets with optimized discrepancy values could substantially boost performance in one-shot optimization.

For the *one-shot regression* task, we observe that there is no clear winning design, nor any obvious correlation between discrepancy and performance, indicating that we cannot rely on simple recommendations suggesting to use a specific design and/or surrogate model. Rather, we observe that competence maps, which provide recommendations based on some high-level features of the problem, can be crucial to achieve peak performance in one-shot optimization.

Constructing High-Performing Designs with Evolutionary Algorithms In the absence of theoretical bounds, we investigate in Sec. 6 how the performances obtained by the tested (design, surrogate) combinations in the one-shot regression task compare against sampling strategies that are explicitly designed for minimizing the MSE individually for each of the 24 benchmark problems.[1] To this end, we apply an off-the-shelf evolutionary algorithm and evolve designs of low MSE for each BBOB function. To our surprise, we find that some of these designs perform very well not only on the problem that they have been designed for, but across all 24 functions, indicating that substantial performance gains over the state-of-the-art one-shot optimization strategies might exist.

Discussion While our results might appear negative with respect to the original question about the correlation between the discrepancy of a sampling strategy and its performance in one-shot optimization, they reveal a clear need and may pave a way for identifying other diversity measures showing a better correlation with the performance results. The evolved designs clearly indicate that such investigations could significantly improve the state of the art. We note that previous attempts to construct low discrepancy samples [15] or well-performing LHS designs [28,22] can be found in the literature. A wider application of such constructions, however, seems to be lagging behind its potential. We therefore believe more research is needed to test these methods in various applications, and to make them easily applicable and accessible.

Finally, while we only focus on one-shot optimization in this work, we note that good one-shot optimization designs are likely to be useful in the context of *sequential model-based optimization (SMBO)* [23]. SMBO is also studied under the notion of *global optimization* or *surrogate-based optimization*, sand entails iterative methods for the optimization of black-box functions that are computationally expensive to evaluate. In SMBO, one uses the evaluated samples of an

[1] Note here that for the classic one-shot optimization task, this question is not meaningful, as the design $\{x\}$ with $x = \arg\min f$ is optimal with zero regret.

initial design to build a model of the true objective function, which is computationally fast or at least much faster to evaluate than the true objective function. In a sequential process this initial design is augmented by injecting further design points in order to improve the function approximation. So-called infill criteria or acquisition functions which usually balance exploitation of the current model and exploration of areas with high model uncertainty are used to decide which point(s) seem(s) adequate to evaluate next with the true objective function. Classic model-based approaches, such as the efficient global optimization algorithm (EGO) by Jones et al. [23], typically use well-distributed, space-filling point sets to initialize the search (see, e.g., [17]).

Reproducibility. We can only show a small set of results here in this extended abstract. Detailed data for both one-shot optimization tasks, for all 29 designs, the 4 surrogate models, 5 sample sizes, and each of the 24 tested BBOB functions is available on our public GitHub repository [3].

2 Low-Discrepancy Designs

The discrepancy of a point set measures how far it deviates from a perfectly distributed set. Various discrepancy measures exist, providing different performance guarantees in quasi-Monte Carlo integration and other applications [1,25,29]. The arguably most common discrepancy metric is the *star discrepancy*, which measures the largest absolute difference between the volume V_y of any origin-anchored box $[0, y] := \prod_{i=1}^{d} [0, y_i]$ and the fraction of points contained in this box. Hence, the star discrepancy of a point set $\{x^1, \ldots, x^n\} \in [0, 1]^d$ is defined as $D^*(X) := \sup_{y \in [0,1]^d} \left| V_y - \left| [0, y] \cap X \right| / n \right|$.

Low-discrepancy designs provide a proven guarantee on their asymptotic discrepancy value. They are well-studied objects in numerical analysis, because of the good error guarantees that they provide for numerical integration. The interested reader is referred to the survey [14], which covers in particular the computational aspects of star discrepancies relevant to our work. In our experiments we consider four different designs of low discrepancy, and we compare them to uniform sampling. More precisely, we study Latin Hypercube Samples (LHS [30]; we use the maximin LHS implementation of the R package `lhs` [8]) and three variants of so-called Halton sequences: the original one suggested by Halton [18], an improved version introduced by Braaten and Wellter [5], and a third design which we obtain from a full enumeration and evaluation of all generalized Halton sequences (for the sample sizes $n \in \{125, 1\,000\}$). For our four-dimensional setting, these are 34 560 different designs each. Those are evaluated using the algorithm by Dobkin et al. [13], which has running time $n^{1+d/2}$. This exact approach becomes infeasible for larger sample sizes, and we use the best generator for $n = 1\,000$ instead. The so-minimized Halton designs are referred to as "Best" in the remainder of this work. The discrepancy values of the different designs used in this paper are summarized in Table 1.

n	Best	BW	Halton	LHS	UNIFORM	EVOLVED
125	0.056	12%	48%	49%	185%	156%
1000	0.013	20%	35%	109%	316%	343%
2500	0.008	0%	6%	295%	371%	–
5000	0.005	2%	6%	376%	413%	–

Table 1. Discrepancy value of the best design and the relative overhead of the other designs. Values for LHS, UNIFORM, and EVOLVED designs are averaged.

3 Experimental Setup and Availability of Data

For our experiments we have chosen the 24 noiseless problems from the *black-box optimization benchmark (BBOB)* by Hansen et al. [20]. For computational reasons, we limit our attention to the first instance of each problem. The BBOB functions assume $[-5, 5]^d$ as search space. We therefore scale our designs, which are initially constructed in $[0, 1]^d$, accordingly.

Our study summarizes the results from a total of 124 080 scenarios. We considered designs of three (deterministic) Halton sequences, as well as LHS and random uniform samples. As the latter two are stochastic, we generated ten samples each to account for their stochasticity. Moreover, each design was generated for the five sample sizes $n \in \{125, 1\,000, 2\,500, 5\,000, 10\,000\}$. For each design, we then computed surrogates using the following four machine learning algorithms: support vector machines (SVMs) [11], decision trees [7], random forests [6], and Kriging [10]. Note that the latter could not be computed on designs of size 10 000 due to memory issues. Also, as (except for the decision trees) the considered algorithms are stochastic – or at least contain stochastic elements within their R implementations – we replicated all experiments ten times. In addition to these 104 880 scenarios we further evaluated a total number of 1 920 "evolved" designs, which will be introduced in detail in Section 5 (one-shot regression).

4 Classic One-Shot Optimization

In the classic one-shot optimization scenario we are asked to provide a point set $\{x^1, \ldots, x^n\}$ for which the quality $f(x^{\text{best}})$ of the best point $x^{\text{best}} := \arg\min_{x^i \in \{x^1, \ldots, x^n\}} f(x^i)$ is as good as possible.

In line with the machine learning literature, where the one-shot problem originates from, we consider simple regret $f(x^{\text{best}}) - f^*$ as performance measure, where $f^* = \inf_x f(x)$ denotes the best function value. In optimization, this measure is referred to as the *target precision* of the best design point. Of course, this performance criterion requires that f^* is known. This is usually not the case for real-world applications, but for the BBOB benchmarks these values are available [20], so that the regret can be computed straightforwardly. Minimizing simple regret is also the standard objective in other related domains, including

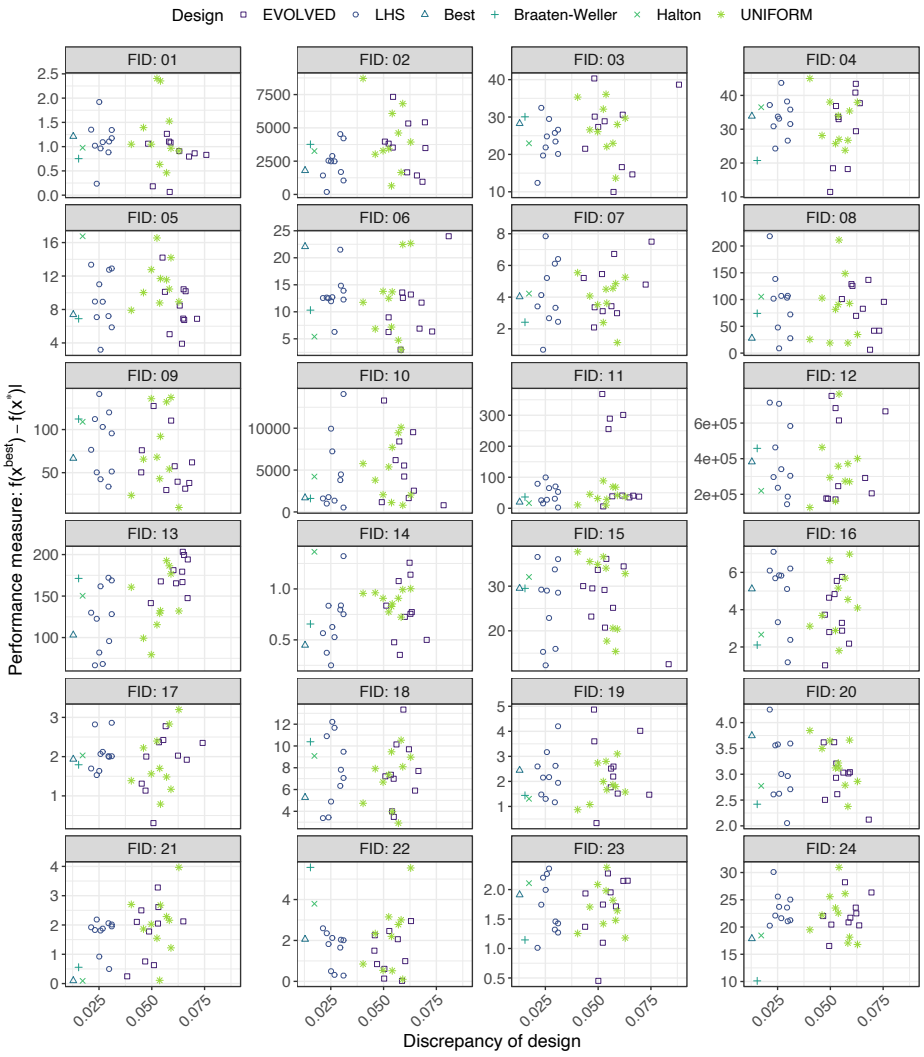

Fig. 1. Scatterplots showing the relationship between the discrepancy of designs of size $n = 1\,000$ and the one-shot performance $f(x^{\text{best}}) - f^*$ for all 24 BBOB problems. The EVOLVED instances were designed for Kriging surrogates.

evolutionary computation [19]. Since this performance depends on a single point, the variance of the results can be tremendous, and it is therefore interesting to compare different designs over different sets of problems (and to perform several independent runs in case of the stochastic designs LHS and uniform sampling).

Fig. 1 compares the average regret for each pair of function and design, and plots the respective performance (y-axis) in dependence of the design's discrepancy (x-axis). Due to different scales of the problems, absolute performances

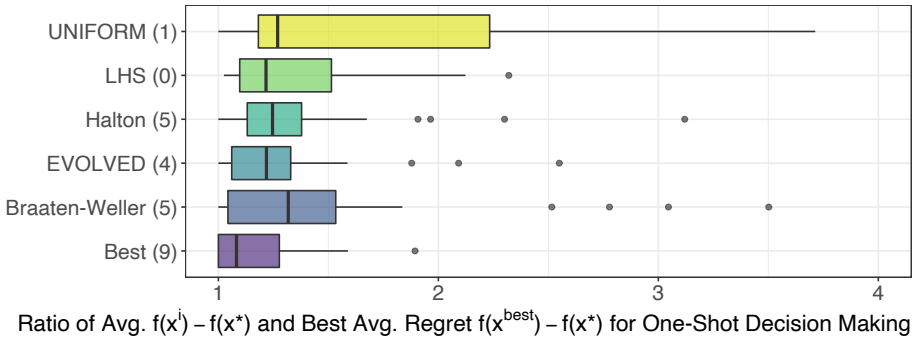

Fig. 2. Boxplots for factors by which the average one-shot result is worse than that of the best design (one data point per BBOB function). The x-axis is capped at 4 (outliers not shown in this plot: UNIFORM at 4.5 and 4.7, and Best at 7.2). Numbers in brackets indicate on how many functions the design achieved the best (average) result. All numbers are for $n = 1\,000$ points and use Kriging surrogates.

should not be directly compared across functions (we will use relative performances instead). As already mentioned before, the results for LHS, UNIFORM and EVOLVED sampling are based on ten independent designs. Note that the concept of the EVOLVED designs will be discussed in more detail later in this work, but we are already showing its results for completeness.

The plots in Fig. 1 indicate that the correlation between discrepancy and one-shot-performance is rather weak, as we do not see any obvious trend. However, one has to keep in mind that these performances depend on a single point only – similar to a *lucky punch* in sports. Therefore, we additionally analyze the aggregated performances in Fig. 2. The boxplots display the distribution of the factor by which each design is worse than the best design for the respective function. According to this aggregated view, the "Best" design – whose discrepancy is the smallest among all sets (recall Tab. 1) – is also the one achieving the smallest mean and median result. The Braaten-Weller-design had the best performance in 5 out of the 24 benchmarks. Although LHS showed good (average) performance as well, it did not achieve the best average result on any of the benchmark functions. Interestingly, uniform sampling achieves a good median score. In fact, we can see in Fig. 1 that the best uniform design often outperforms all other designs, but at the same time there is (with few exceptions) always at least one of the uniform samples which is worse than all other designs. Of course, our benchmark set is small compared to broad range of numerical problems encountered in practice. An extension to more use-cases, possibly grouped by type of application, forms an important direction for future work.In particular, we suggest to not only consider more instances of the BBOB functions, but to extend our approach to other problems, such as those provided by Nevergrad [32,33], the problems from the black-box optimization competition BBComp (https://bbcomp.ini.rub.de), and to hyper-parameter tuning.

n	Best	BW	Halton	LHS	UNIFORM	**total**
125	6	13	8	11	4	**42**
1 000	9	9	6	11	3	**38**
2 500	11	7	8	16	3	**45**
5 000	12	10	11	14	3	**50**

Table 2. Number of functions for which the respective design, together with the Kriging surrogates, achieved (on average) a MSE that is at most 5% worse than the best achieved MSE. We recall that we have 24 benchmark problems in total.

5 One-Shot Regression

We now turn our attention to the *one-shot regression* problem, in which we aim to build a regression model $\hat{f}$ that predicts the function values of the true function f as accurately as possible. The accuracy of the *one-shot regression models* is measured by the mean squared error (MSE), for which we evaluate both f and our proxy $\hat{f}$ in t i.i.d. points $y^1, \ldots, y^t$, which are selected from the domain $[-5, 5]^d$ uniformly at random. The MSE is then computed as

$$\mathrm{MSE}(\hat{f}) := \frac{1}{t} \sum_{j=1}^{t} \left(f(y^j) - \hat{f}(y^j) \right)^2.$$

In our evaluation, we use $t = 100\,000$ i.i.d. samples. For LHS and UNIFORM designs, we compute the MSE for each of the ten random designs, and average the results.

In Tab. 2 we compare the five designs for different sample sizes. For each sample size, we count the number of functions for which the design achieved an MSE that is at most 5% worse than the best one for the respective sample size. The displayed results are based on Kriging but results for the other surrogate models are similar. Uniform samples seem to enable less accurate regression models than the Halton and LHS designs. However, there are three cases in which the uniform design yields the best MSE: for function F16 (Weierstrass) with $n = 125$ points, F22 (Gallagher's Gaussian 21-hi Peaks) with $n = 1\,000$, and F3 (Rastrigin) with $n = 2\,500$. In the latter case no other design achieves an MSE within the 5% margin, whereas for the first two combinations the other designs achieve just slightly worse MSEs.

Fig. 3 provides a more detailed impression of the regression quality for the different (design, function)-pairs. This chart includes the EVOLVED designs, which we introduce and discuss in the next section. The results in Fig. 3 are for Kriging, but those for the other models look alike. We observe clear patterns: uniform designs, in general, produce surrogate models with high mean MSE and high variance and hence a poor global approximation of the target function f on average. An exception is F21, Gallagher's Gaussian 101-me peaks function, for which uniform samples obtain the best median results with far reaching whiskers

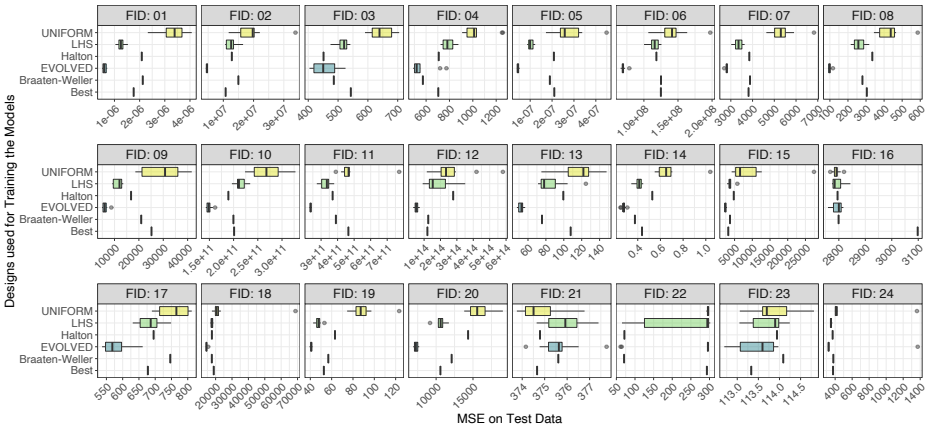

Fig. 3. MSE of the Kriging models, individually trained for each of the six designs with $n = 1\,000$ points and each of the 24 BBOB problems. Each model was assessed on a set of 100 000 i.i.d. uniform samples. Boxplots show the distribution of the 10 independent constructions.

though. We attribute this to lucky sampling. In contrast, models fitted to LHS and low-discrepancy designs tend to be much more accurate approximations of the true function. However, there is no obvious winner among the five designs, indicating that the correlation between discrepancy and performance is more complex than one might have hoped for.

6 Evolving Designs for One-Shot Regression

Given a target function f and a surrogate model we do not know what quality (w.r.t. the MSE on test data) one can achieve in the best-case with an optimal design of n points in d dimensions – a baseline is missing. In order to get an impression for the absolute quality of our tested designs, as well as for the potential of further improvement, we have approximated optimal n-point designs by means of an evolutionary algorithm (EA) [16]. That is, we evolve sampling plans in a heuristically guided stochastic manner.

Our algorithm starts with an initial LHS design x of n points in $[-5, 5]^d$. In each iteration, a new candidate design y is created from the current-best design x by applying Gaussian perturbations to a subset of $\lfloor n/10 \rfloor$ points. Points falling off the $[-5, 5]^d$ boundaries are repaired by projecting the violating components to the boundary, see Fig. 4 for an illustration. If y is no worse w.r.t. the fitness function, replace x by y, otherwise discard y. The process is repeated for a fixed number of 2 000 iterations. The fitness function fits a surrogate model based on the given design in a first step. Next, the quality of the surrogate is assessed by means of the MSE for ten random uniform designs with 10 000 points each. The fitness value is the average of these MSE values and is meant to be minimized.

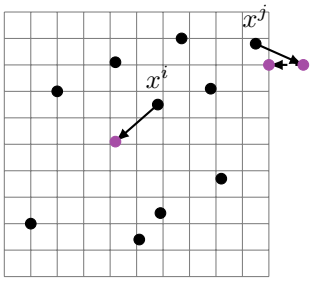

Fig. 4. Illustration of mutation step of a design with $n = 10$ points (black dots) in two dimensions. Here, the two points x^i, x^j are subject to mutation (solid arrows). The perturbation of x^j results in a point outside the bounding box. This is where a repair mechanism comes in (dashed arrow).

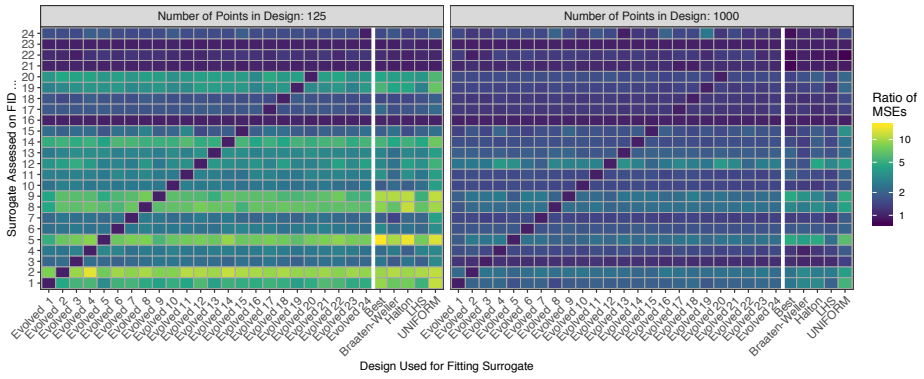

Fig. 5. Illustration of MSEs across the 24 functions from the BBOB suite. The first 24 columns correspond to the evolved designs (the i-th evolved design has been optimized for the i-th BBOB function), and the remaining five columns show the results for the five one-shot designs (Best, Braaten-Weller, Halton, LHS and UNIFORM). For each of the 29 designs, a Kriging model has been fitted to the BBOB function of the respective row and assessed by means of the MSE. The cell colors illustrate the ratio of the respective model's MSE and the MSE of the corresponding problem-tailored (i.e., evolved) design.

Note that each run of the EA produces a large set of interim solutions, but we only keep the final design for further evaluation.

We evolved ten designs (to account for randomness of the EA approach) for each combination of surrogate modelling approach, BBOB function, and size $n \in \{125, 1\,000\}$ of sampling plan resulting in 1 920 EVOLVED designs. We neglected larger sampling sizes to keep computational costs reasonable (each fitness evaluation requires fitting a surrogate on n points, which becomes computationally expensive for increasing n).

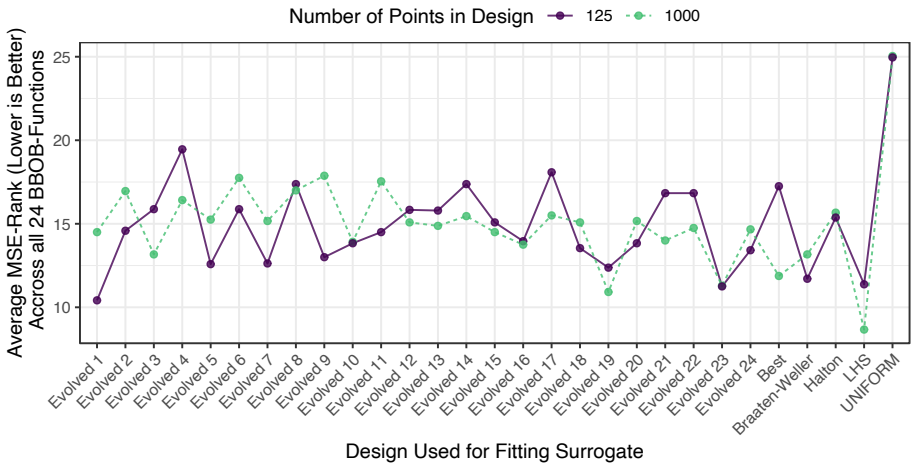

Fig. 6. Visualization of average MSE-ranks (lower is better) for all 29 designs. This figure aggregates the detailed MSE values in Fig. 5 across all 24 BBOB functions.

Returning to Fig. 3 we observe that the evolved designs lead to drastic improvements w.r.t. the MSE (and low variance) for the majority of BBOB functions; in particular for FIDs 1-14 (first three BBOB groups with mainly unimodal functions with global structure). Contrary, for FIDs 16 and 21-23 – i.e., functions which are characterized by a highly rugged landscape with many local optima and weak global structure – the evolving process is far less successful w.r.t. MSE improvement.

Recall that we evolved designs for specific combinations of target function, surrogate-modelling approach, and sample size. However, as depicted in Fig. 5, the problem-specifically evolved designs are not necessarily inferior to any of the established sampling strategies. While the designs resulted indeed in significantly superior performances on the problems they have been evolved on – as can be seen by the diagonal of dark blue cells – their MSE ratios are usually comparable, if not even better, than the respective ratios of Best, Braaten-Weller, etc. When comparing the average ranks obtained by the 29 designs, see Fig. 6, the design evolved for F1 (Sphere) achieves the best average score (10.4) of all 29 tested designs for $n = 125$, closely followed by LHS (11.4) and Braaten-Weller (11.7). For $n = 1\,000$ LHS has an average rank of 8.7, while the runner-ups are "Evolved 23" (11.3) and Best (11.9). Several other evolved designs obtain fine average ranks. Noticeably, the uniform design is clearly the worst, with an average rank of 25.0 for both $n = 125$ and $n = 1\,000$. That is, the MSE of uniform sampling is on average more than 5 ranks worse than any of the other 28 designs.

Fig. 5 also reveals quite noticeable differences across the functions on which the trained surrogates are assessed (rows). Non-surprisingly, we observed a decrease in the MSE ratios for an increase in sample size.

7 Conclusion

We have analyzed the question whether the promising results of low-discrepancy point sets for one-shot optimization are well correlated with the discrepancy of these sets. No strict one-to-one correlation could be identified, neither in the classic nor in the one-shot regression scenario. These results refute our hope that the challenging and resource-consuming task of designing efficient one-shot designs could be reduced to a discrepancy-minimization problem (which is also a challenging task in its own, see [15,31], but of a much smaller scale than the one-shot design one). In terms of aggregated results, however, the low-discrepancy designs performed well in the classic one-shot optimization task. In future work, we plan on investigating whether other diversity measures (such as, for example, those mentioned in [12]) show a better correlation. Among the most promising candidates are indicators measuring how "space-filling" the designs are. A related question is how well good designs for one one-shot optimization task perform on other tasks.

The decent performance of the problem-specific designs obtained through our evolutionary approach was a big surprise. Not only did they improve quite considerably over the standard designs for one-shot regression for the problem and learner they were evolved for, but some of them even rank in the top places when evaluated across the whole benchmark set. A cross-validation of the evolutionary approach on other benchmarks and an extension to other dimensions forms another line of research that seems very promising in the context of one-shot optimization.

Acknowledgments. We thank François-Michel de Rainville for help with his implementation of the generalized Halton sequences. We also thank the reviewers for providing useful comments and references. This work was financially supported by the Paris Ile-de-France Region, by ANR-11-LABX-0056-LMH, by the Australian Research Council (ARC) through grant DP190103894, and by the South Australian Government through the Research Consortium "Unlocking Complex Resources through Lean Processing". Moreover, P. Kerschke acknowledges support by the *European Research Center for Information Systems (ERCIS)*.

References

1. Beck, J.: Irregularities of Distribution. Cambridge University Press, Cambridge (1987)
2. Bergstra, J., Bengio, Y.: Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research (JMLR) **13**, 281 – 305 (2012), http://dl.acm.org/citation.cfm?id=2188395
3. Bossek, J., Doerr, C., Kerschke, P., Neumann, A., Neumann, F.: Github repository with project data. https://github.com/jakobbossek/PPSN2020-oneshot/ (2020)
4. Bousquet, O., Gelly, S., Kurach, K., Teytaud, O., Vincent, D.: Critical Hyper-Parameters: No Random, No Cry. CoRR **abs/1706.03200** (2017), http://arxiv.org/abs/1706.03200

5. Braaten, E., Weller, G.: An Improved Low-Discrepancy Sequence for Multidimensional Quasi-Monte Carlo Integration. Journal of Computational Physics **33**(2), 249–258 (1979)
6. Breiman, L.: Random Forests. Machine Learning **45**(1), 5 – 32 (2001). https://doi.org/10.1023/A:1010933404324
7. Breiman, L., Friedman, J.H., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Wadsworth & Brooks/Cole Advanced Books & Software (1984). https://doi.org/10.1201/9781315139470
8. Carnell, R.: lhs: Latin Hypercube Samples (2020), https://CRAN.R-project.org/package=lhs, r package version 1.0.2
9. Cauwet, M., Couprie, C., Dehos, J., Luc, P., Rapin, J., Rivière, M., Teytaud, F., Teytaud, O.: Fully parallel hyperparameter search: Reshaped space-filling. CoRR **abs/1910.08406** (2019), http://arxiv.org/abs/1910.08406
10. Chilès, J.P., Desassis, N.: Fifty Years of Kriging. In: Handbook of Mathematical Geosciences: Fifty Years of Journal of the International Association for Mathematical Geology (IAMG), pp. 589 – 612. Springer (2018), https://link.springer.com/chapter/10.1007/978-3-319-78999-6_29
11. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning **20**(3), 273 – 297 (1995), https://link.springer.com/article/10.1007/BF00994018
12. Crombecq, K., Laermans, E., Dhaene, T.: Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. European Journal of Operational Research **214**(3), 683 – 696 (2011). https://doi.org/https://doi.org/10.1016/j.ejor.2011.05.032
13. Dobkin, D.P., Eppstein, D., Mitchell, D.P.: Computing the discrepancy with applications to supersampling patterns. ACM Transactions on Graphics **15**, 354–376 (1996)
14. Doerr, C., Gnewuch, M., Wahlström, M.: Calculation of discrepancy measures and applications. In: A Panorama of Discrepancy Theory, pp. 621–678. Springer (2014)
15. Doerr, C., Rainville, F.D.: Constructing low star discrepancy point sets with genetic algorithms. In: Proc. of Genetic and Evolutionary Computation Conference (GECCO). pp. 789–796. ACM (2013). https://doi.org/10.1145/2463372.2463469
16. Eiben, Á.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2015). https://doi.org/10.1007/978-3-662-44874-8
17. Forrester, A.I.J., Sobester, A., Keane, A.J.: Engineering Design via Surrogate Modelling - A Practical Guide. Wiley (2008)
18. Halton, J.H.: Algorithm 247: Radical-Inverse Quasi-random Point Sequence. Communications of the ACM **7**(12), 701 – 702 (1964). https://doi.org/10.1145/355588.365104
19. Hansen, N., Auger, A., Mersmann, O., Tušar, T., Brockhoff, D.: COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. ArXiv e-prints **arXiv:1603.08785** (2016)
20. Hansen, N., Finck, S., Ros, R., Auger, A.: Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Tech. Rep. RR-6829, INRIA (2009), https://hal.inria.fr/inria-00362633/document
21. Hlawka, E.: Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. Ann. Mat. Pura Appl. **54**, 325–333 (1961)
22. Jin, R., Chen, W., Sudjianto, A.: An efficient algorithm for constructing optimal design of computer experiments. Journal of Statistical Planning and Inference **134**(1), 268 – 287 (2005). https://doi.org/https://doi.org/10.1016/j.jspi.2004.02.014

23. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization **13**, 455 – 492 (1998). https://doi.org/10.1023/A:1008306431147
24. Koksma, J.F.: Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. Mathematica B (Zutphen) **11**, 7–11 (1942/3)
25. Kuipers, L., Niederreiter, H.: Uniform distribution of sequences. Wiley (1974)
26. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer (2009). https://doi.org/10.1007/978-0-387-78165-5
27. Leobacher, G., Pillichshammer, F.: Introduction to Quasi-Monte Carlo Integration and Applications. Springer (2014). https://doi.org/10.1007/978-3-319-03425-6
28. Liu, L.: Could enough samples be more important than better designs for computer experiments? In: Proc. of Annual Symposium on Simulation (ANSS'05). p. 107–115. IEEE (2005). https://doi.org/10.1109/ANSS.2005.17
29. Matoušek, J.: Geometric Discrepancy. Springer, Berlin, 2nd edn. (2009)
30. McKay, M.D., Beckman, R.J., Conover, W.J.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics **21**, 239–245 (1979), http://www.jstor.org/stable/1268522
31. Rainville, F.D., Gagné, C., Teytaud, O., Laurendeau, D.: Evolutionary Optimization of Low-Discrepancy Sequences. ACM Transactions on Modeling and Computer Simulation **22**, 9:1–9:25 (2012). https://doi.org/10.1145/2133390.2133393
32. Rapin, J., Gallagher, M., Kerschke, P., Preuss, M., Teytaud, O.: Exploring the MLDA Benchmark on the Nevergrad Platform. In: Proceedings of the 21st Annual Conference on Genetic and Evolutionary Computation (GECCO'19) Companion. pp. 1888 – 1896. ACM (2019). https://doi.org/10.1145/3319619.3326830
33. Rapin, J., Teytaud, O.: Nevergrad - A Gradient-Free Optimization Platform. https://GitHub.com/FacebookResearch/Nevergrad (2018)