A new criterion for Latin hypercube optimization

Timm J. Peter, Oliver Nelles

Universität Siegen, Department Maschinenbau
Institut für Mechanik und Regelungstechnik - Mechatronik
Paul-Bonatz-Str. 9-11, 57068, Siegen, Germany
E-Mail: {timm.peter,oliver.nelles}@uni-siegen.de

Abstract

In this contribution, a new approach for optimizing LH designs based on the estimation and evaluation of pdfs is presented. The proposed algorithm minimizes the mean absolute error between the estimated pdf of the LH design, evaluated solely on its data points, and the uniform distribution. To validate the functionality of the new approach, it is compared to other state-of-the-art methods to create space-filling designs. The methods are compared using the KL divergence of the resulting datasets and the uniform distribution, as well as the resulting computation times for various dimensions and number of data points. Overall, the KL divergence performance of the new approach is outstanding, but expensive in terms of the computational demand. An additional benefit of the proposed approach is that it allows higher flexibility for DoE desgins. For example, it can be extended to approach any arbitrary point distribution, not just uniform, and may be suitable for the integration of constrains.

1 Introduction

The training data point distribution in the input space, also called the experimental design, is an important influencing factor regarding the quality of data-driven models. For this reason, an assessment of the quality of the design of

DOI: 10.58895/ksp/1000124139-14 erschienen in:
Proceedings – 30. Workshop Computational Intelligence: Berlin, 26. - 27. November 2020
DOI: 10.58895/ksp/1000124139 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000124139/

experiments (DoE) is important before measurements take place. If no prior knowledge about a system exists, uniformly distributed input data should be used [2]. Furthermore, besides being space-filling in the original input space, two other properties are advantageous and concern the projection of the data onto the individual input axes: a) one-dimensional uniform distribution on each axis (1Duni) and b) non-collapsing design which means that the projected data points stay distinct in their 1D projections.

A well-known strategy used for the experimental design that places points on a grid while avoiding the curse of dimensionality is the Latin hypercube (LH) design. LH designs fulfill the 1Duni property, but do not inherently provide a uniform data distribution. Therefore, several loss functions were proposed, which try to rate uniformity of the data point distribution, such as maximin or ϕ_p [1]. These loss functions focus on the data point pair with the smallest distance which makes them suitable for optimization purposes, e.g. for the optimization of LH designs as in [3]. These approaches drive all points away from each other during optimization, thereby creating a uniform point distribution. On the downside, these local approaches are structurally not able to rate the overall distribution quality, but can be utilized for optimization. It can be carried out by local search, e.g., based on point exchanges [4, 3] and global optimization methods, e.g., particle swarm optimization [10], simulated annealing [11] or evolutionary algorithms [12].

This contribution proposes a new approach to optimize the data point distribution of an LH design based on the use of probability density function (pdf) estimation. Thus, contrary to the above-mentioned approaches, the proposed method can rate the overall distribution quality. The approach estimates the pdf for all data points. In this contribution a local search method that utilizes point exchanges is employed. In each iteration the algorithm exchanges a coordinate of a point pair, based on the difference of the estimated pdf and the uniform distribution. This procedure is compared to other state-of-the-art approaches for space-filling designs. The quality of the different approaches is evaluated using the Kullback-Leibler (KL) divergence.

The contribution is structured as follows. Section 2 introduces the most important methods to calculate space-filling designs, namely (i) Sobol sequences

and (ii) LH designs. Additionally, since LH designs are not necessarily space-filling, optimization strategies to achieve space-filling LH designs are introduced. Section 3 provides the concept of kernel density estimation and a special evaluation strategy for pdfs, before in Sec. 4 the new density-based LH design optimization strategy is presented. In Sec. 5 the performance of the algorithm is analyzed and compared to other state-of-the-art methods for space-filling designs. Finally, a conclusion is given.

2 Space-filling designs

The two main properties of a design of experiments (DoE), if no prior knowledge about the process is available, are the (i) 1Duni property and (ii) the space-filling property. This section gives a quick overview of two important methods to achieve space-filling designs.

2.1 Sobol sequences

The most commonly used strategy to create space-filling datasets is introduced in [9]. The so-called Sobol sequences are low-discrepency sequences, which are uniformly distributed for $N = 2^x$ data points with $x \in \mathbb{N}$. The foundation to create Sobol sequences is the successive subdivision of each dimension in halves and the reordering of the coordinates in each dimension. This procedure is computationally very cheap, even for high numbers of data points and high dimensions and thus used commonly. On the downside, Sobol sequences do not fulfill the 1Duni property.

2.2 Latin hypercubes

Originally being used in the field of computer experiments, LH designs fulfill the 1Duni property due to their structure, but they do not fulfill the space-filling property intrinsically. Therefore, a subsequent optimization of the LH design has to be performed, if a space-filling dataset is desired.

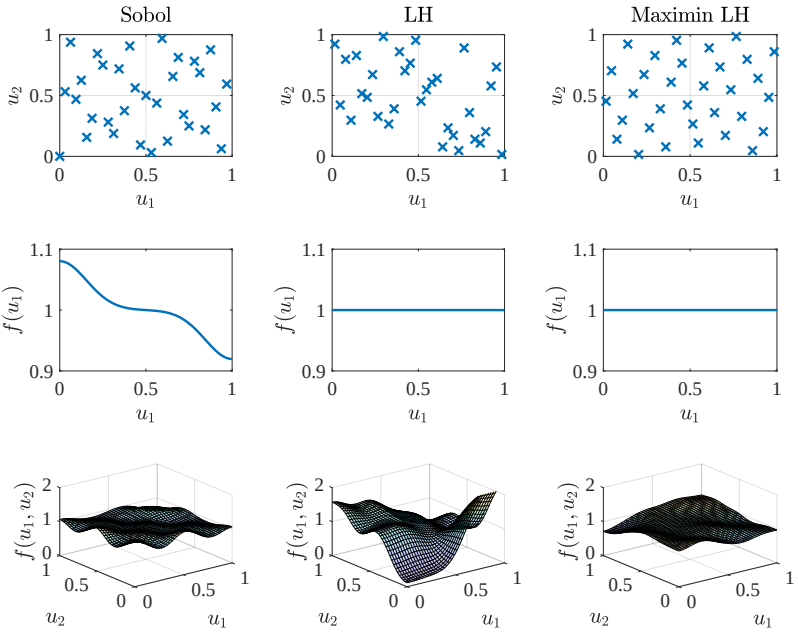

Figure 1: The first row of subplots visualizes the point distribution, the second row the 1Duni property and the last row the space-filling property for $N = 32$ data points and $n = 2$.

To create an LH design, the number of samples N and the dimension n have to be fixed by the user. Each input $u_i, i = 1, 2, ..., n$ is partitioned into N levels. Thus, for N samples and n input dimensions, N^n grid points are constructed. Out of these N^n grid points, N are occupied by data points. Here, each level of the N levels is occupied once, establishing the non-collapsing property of the LH design.

2.3 Optimization of Latin hypercubes using deterministic local seach

This subsection focuses on the optimization of LH designs using the deterministic local search (DLS) [4] and the extended deterministic local search (EDLS) [3] algorithms.

The procedure of the DLS Algorithm is as follows. A dataset, here an LH design, with N data points in n dimensions is given. The goal is to maximize the distance of the two nearest neighbors in the dataset. To achieve this, the first step is to calculate the nearest neighbor distance between all data points. The two data points with minimal distance are labeled as the critical points, the rest of the dataset is labeled as potential swap partners. Now the coordinates of a point pair are swapped in one dimension. This point pair consisting of one out of the two critical points and one swap partner point. If the coordinate swap reduces the minimum nearest neighbor distance, the procedure starts again with the first step. If not, the algorithm tries all possible dimensions and all potential swap partners. If no further improvements can be achieved, the algorithm is terminated. This procedure effectively drives all data points away from each other, thereby creating a space-filling design. At the same time, the 1Duni property is preserved.

The DLS algorithm has different positive properties. On the one hand, it preserves the LH design, if the initial set is also an LH design. On the other hand, the algorithm can optimize any collapsing design as well. It is deterministic, thus the results are reproducible and it ensures improvement in each iteration.

The idea of the DLS algorithm is amplified in [3] to the extended deterministic local search (EDLS). A small modification of the DLS leads to a vast change in the algorithm's behavior. Instead of considering only the *smallest* nearest neighbor distance, all neighbor distances are inspected and maximized. This has the effect of even more homogeneous data distributions in comparison to the DLS. On the downside, the computing time increases considerably.

3 Estimation and evaluation of probability density functions

For a n-dimensional dataset consisting of the data points $\underline{u}(i) = [u_1(i)\ \ u_2(i)\ \ \cdots\ \ u_n(i)]^T,\ \ i = 1,2,\ldots,N$ summarized in a matrix $\underline{U} = [\underline{u}(1)\ \underline{u}(2),...,\underline{u}(N)]$ a density estimator can be used to estimate its pdf $q(\underline{u})$. The pdf is based on placing a kernel on each data point. This procedure is known as kernel density or Parzen estimator. In areas, where two or more

kernels overlap, the corresponding values are added. In most cases, a Gaussian kernel is used. For other kernel types, see [6, 8].

The pdf is estimated by

$$\hat{p}(\underline{u}) = \frac{1}{N}\sum_{i=1}^{N} K(\underline{u},\underline{u}(i)),\ K(\underline{u},\underline{u}(i)) = \frac{\exp\left(-\frac{1}{2}[\underline{u}-\underline{u}(i)]^T\Sigma^{-1}[\underline{u}-\underline{u}(i)]\right)}{\sqrt{(2\pi)^n|\Sigma|}} \quad (1)$$

typrically with a diagonal covariance matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$ of n standard deviations for each dimension. Estimator (1) can be interpreted (i) as a sum of N kernels with height one normalized by their integral sum or (ii) as an average of N normal distributions.

One commonly used strategy to determine standard deviations is to use Silverman‘s rule-of-thumb [8]

$$\sigma_i = \sigma_{ui}\left(\frac{4}{n+2}\right)^{\frac{1}{n+4}} N^{-\frac{1}{n+4}} \quad (2)$$

with the standard deviation of the data σ_{ui} in dimension i. For alternative approaches to determine standard deviations, see [7].

Typically, Monte Carlo sampling is used to evaluate pdf estimates of datasets. This is a computationally demanding task since the number of sampling points has to be high “enough”. In a recent publication, a different, less time-consuming strategy to estimate and evaluate pdfs was proposed [5]. Here, a pdf is estimated using kernel density estimation and evaluated solely on the data points of the dataset itself, to select a subset of an original dataset. Compared to Monte Carlo sampling the computational effort decreases dramatically, without decreasing the subset selection performance.

In terms of calculation, this strategy can be visualized using a symmetric $N \times N$ matrix. Figure 2 shows the combinations of each point of a dataset $\underline{u}$ used for evaluation (rows) with the kernels of the estimated density (columns). The dataset’s pdf value evaluated at one data point can be calculated by taking the mean of the associated row of this calculation matrix.

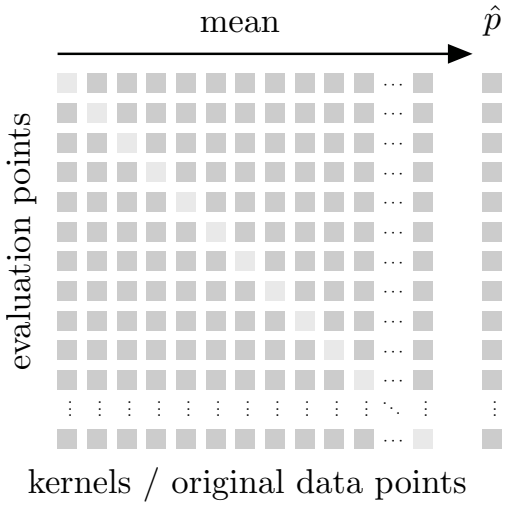

Figure 2: Pdf visualization in form of a matrix $\underline{X}$.

This calculation matrix additionally offers an interesting opportunity for the estimation of pdfs. If a matrix is build-up for a dataset and one data point is replaced by a new data point, the pdf can easily be adjusted, instead of being forced to recalculate the whole matrix from scratch. One simply has to calculate the effect of a kernel, placed on the new data point, on all other data points, and vice versa. The effect of the kernel on all data points equals the row of the particular data point in the calculation matrix, whereas the effect of all kernels on the data point equals the column. If the row and column of the new data point are updated, the matrix represents the pdf of the new dataset.

4 New algorithm

The proposed algorithm effectively combines elements from the DLS algorithm and the efficient pdf calculation strategy proposed in [5]. While the DLS algorithm maximizes the minimum nearest neighbor distance to find optimal LH designs, the new method has a more global approach for the loss function. It minimizes the mean absolute error of the difference between the uniform

distribution and the estimated pdf of a given LH design. For the flow chart of the new algorithm, see Fig. 3. The proposed algorithm proceeds as follows.

1. Initialize the algorithm with an arbitrary n-dimensional LH design $\underline{U} = [\underline{u}(1), \underline{u}(2), \ldots, \underline{u}(N)]$ consisting of the data points $\underline{u}(i) = [u_1(i)\ u_2(i)\ \cdots\ u_n(i)]^T$, $i = 1, 2, \ldots, N$.

2. In order to estimate the pdf $\hat{p}(\underline{u})$ of the LH design, kernel density estimation is used. The $N \times N$ calculation matrix $\underline{X}$ is set up for the LH design $\underline{U}$. The matrix $\underline{X}$ is symmetric, since data points and evaluation points are the same. Take the mean of all rows of $\underline{X}$ to receive $\hat{p}(\underline{u})$.

3. Select the data point with the maximum pdf value as the point to swap. This represents the point where data is most dense and therefore should be moved away by swapping with a partner point.

4. Go through all non-selected points $i = 1 : N-1$ in each dimension $ii = 1 : n$ and initialize a swap matrix $\underline{X}_s = \underline{X}$ for each point-dimension combination and an associated matrix $\underline{U}_s = \underline{U}$ to store the altered LH design. Swap the coordinates of the swap point and point i in dimension ii. Instead of recalculating the whole pdf, this merely equals a recalculation of two rows and two columns of $\underline{X}_s$ and the update of two points of $\underline{U}_s$. Calculate the pdf at all data points. Calculate the error $E(i, ii) = \frac{1}{N} \sum_{k=1}^{N} (|1 - \hat{p}(\underline{u}(k))|)$. For $N-1 \times n$ swaps, this results in the $N-1 \times n$ error matrix $\underline{E}$.

5. Execute the swap with the minimal value of the error matrix $\underline{E}$. Update the corresponding matrices $\underline{X} = \underline{X}_s$ and $\underline{U} = \underline{U}_s$ as well as $\hat{p}(\underline{u}) = \hat{p}(\underline{u}_s)$. Terminate the algorithm if no improvement is possible, otherwise continue at step 3.

5 Performance analysis

In this section, the performance of the new density-based LH design optimization approach is analyzed and compared to other state-of-the-art methods. For this purpose, space-filling designs for different dimensions ($n = 2, ..., 5$)

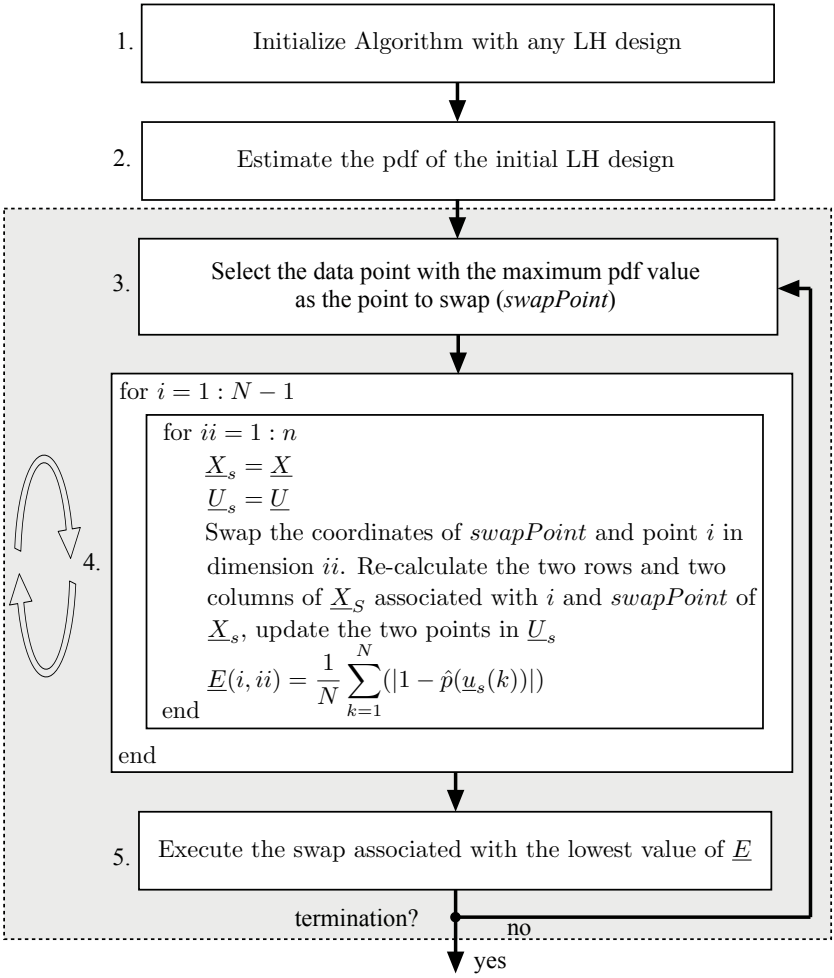

Figure 3: Procedure of the new algorithm.

and numbers of data points ($N = [20, 40, ..., 200]$) are compared. To assess the space-filling qualities of the methods, the Kullback-Leibler (KL) divergence between the estimated pdf of the datasets and the uniform distribution using Monte Carlo sampling with 10000 random points is calculated. Lower KL divergence values indicate a higher similarity of two pdfs. For a KL divergence value of zero two pdfs are identical. Furthermore, the computation time is investigated.

5.1 KL divergence evaluation

Figure 4 shows the result of the evaluation of the KL divergence over the number of data points for $n = 2, ..., 5$. For each investigated dimension, the proposed density-based LH design optimization consistently produces the best results. In most parts of the evaluation, the EDLS achieves the second-best results, followed by the DLS algorithm. The Sobol sequences exhibit the critical overall KL divergence scores and thus the critical space-filling datasets.

The difference between the diverse approaches can be attributed to the employed optimization criteria. The (extended) deterministic local search optimizes a nearest-neighbor-based criterion, thus only distances between pairs of data points are considered. On the contrary, the density-based LH design optimization directly optimizes the pdf of the LH design. Because of this global optimization approach, the density-based LH design optimization is more powerful and therefore able to achieve better KL divergence performances. Sobol sequences have the most simplistic structure out of the four analyzed methods. Furthermore, Sobol sequences work best if $N = 2^x$ with $x \in \mathbb{N}$. Thus, in a separate case not shown in this contribution, the four methods were analyzed for $N = [16, 32, 64, 128, 256]$. The performance differences were similar to the presented case.

5.2 Calculation time evaluation

Figure 5 shows the evaluation of the calculation time in dependence on the number of data points for different dimensions. Since the calculation time of

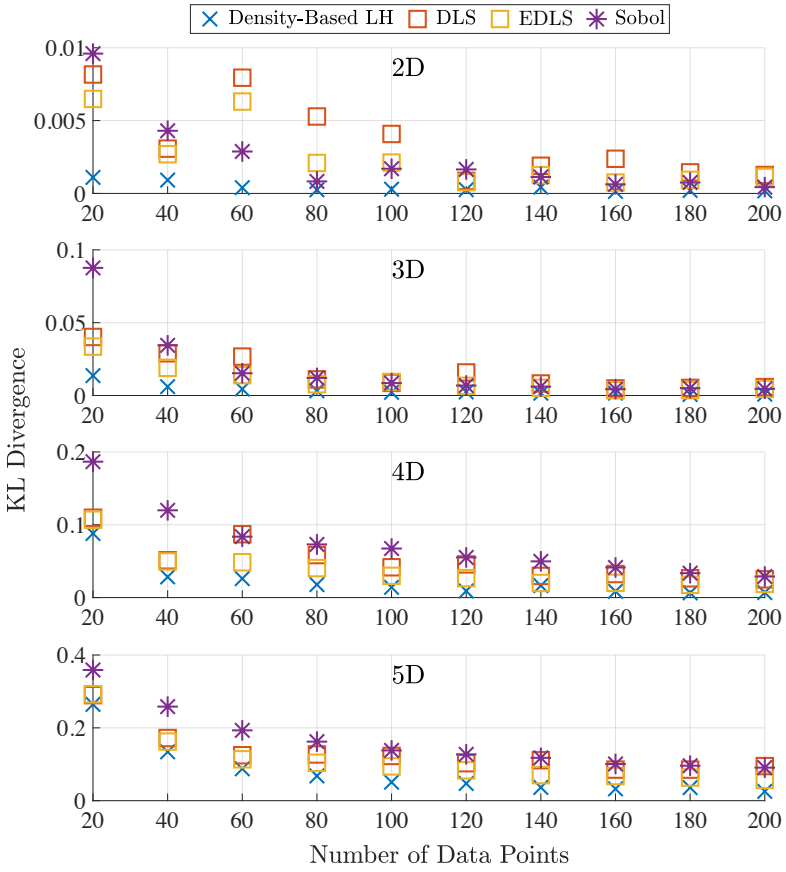

Figure 4: Comparison of KL divergences over different dimensions and number of data points.

Sobol sequences is negligibly small, it is left out of this particular evaluation. The density-based LH design optimization shows the highest calculation times, increasing with both the number of data points and the number of dimensions, followed by the EDLS and, with a big difference due to the simpler structure, the DLS. Furthermore as a reference several higher dimensional cases have been investigated. In this test case the calculation time for the new algorithm was 32 minutes in 7D and 69 minutes in 10D, both with $N = 100$ data points.

The computational effort of the density-based LH design optimization can be attributed to the time-consuming calculations of the pdfs for every possible point switch in each iteration. An alteration of the algorithm in steps 4 and 5, compare Fig. 3, could make it more efficient. In the proposed version, i runs over all $N-1$ data points and executes the swap with the lowest overall error. Alternatively, rather than performing the best swap, the algorithm could run until the *first* swap improves the error. This alteration will be subject to further research.

5.3 Extension to a stochastic algorithm

The density-based LH design optimization always performs the best possible point switch. Therefore, it represents a local search method. A simple, straightforward stochastic extension is presented to circumvent this shortcoming to possibly increase the algorithm's performance even more. On the downside, the convergence of the algorithm is not strictly decreasing.

The only necessary change takes place in step 5, compare Fig. 3. Instead of executing the *best* swap resulting in the best possible quality, a swap is executed with a probability proportional to the size of a quality measure. Here, the inverse mean absolute error is used as the measure of quality. This stochastic element helps the algorithm to escape bad local optima, see Fig. 6. Here, the density-based LH approach stops after around 100 iterations, because no swap leads to further improvement of the quality. In comparison, the density-based LH with stochastic extension is terminated after around 400 iterations through an artificial criterion, which terminates the algorithm if no improvement is achieved for 20 iterations.

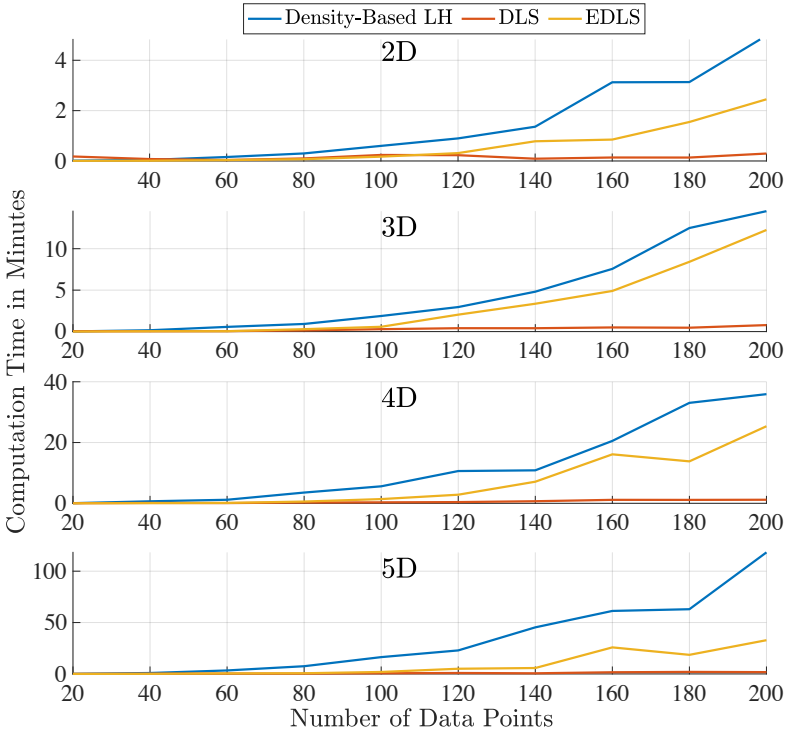

Figure 5: Comparison of computation times over different dimensions and number of data points.

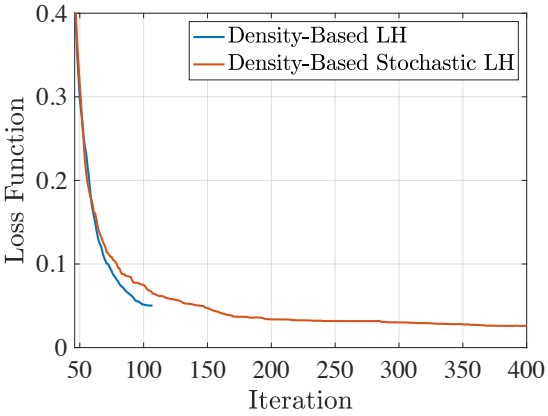

Figure 6: Exemplary comparison of the loss function progress over the iterations for $N = 120$ and $n = 3$.

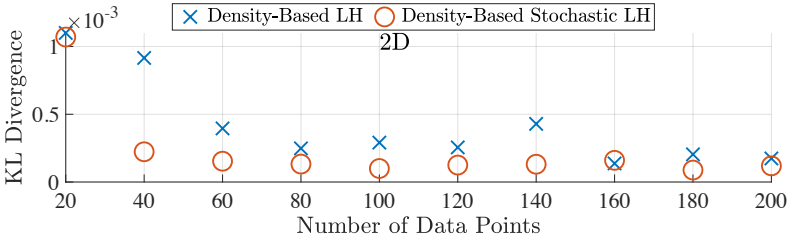

Figure 7: Comparison of KL divergences over the number of data points.

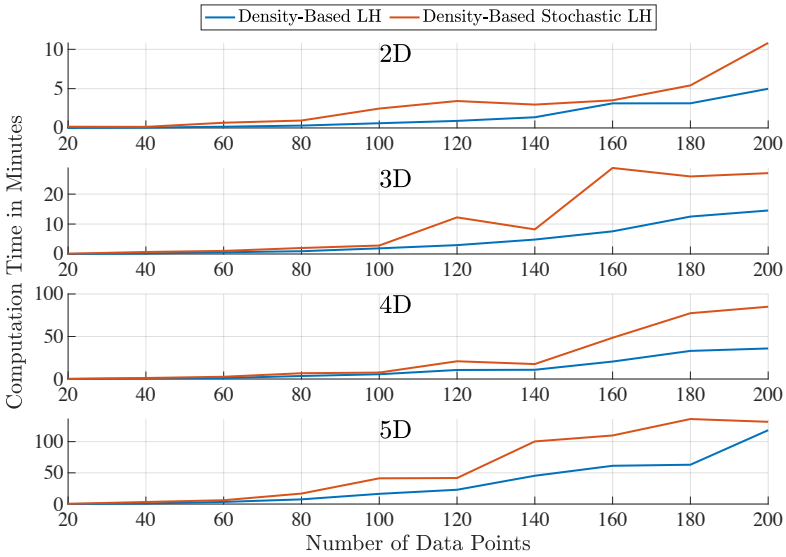

Figure 8: Comparison of calculation times over different dimensions and number of data points.

As Fig. 7 shows, the impact of the stochastic extension is visible in 2D. Since the influence in dimensions 3 to 5 is negligible, the plots are left out. While the improvement is moderate to non-existing in terms of performance, it is significant in terms of computational demand, see Fig. 8. In the perspective of the author, the almost non-existing advantage in terms of performance is not worth the impact on the computational demand.

6 Conclusion

In this contribution, a new approach to optimize LH designs, based on the estimation and evaluation of pdfs, was presented. The algorithm minimizes the mean absolute error or any other criterion between the estimated pdf of the LH design, evaluated solely on its data points, and the uniform distribution. With this structure, the algorithm is capable of swapping data points of an LH design

to achieve a uniform point distribution. Alternative optimization strategies in combination with the new proposed criterion are also possible.

In order to validate the functioning of the new approach, it was compared to other state-of-the-art methods to create space-filling designs. The methods were compared using the KL divergence of the resulting datasets and the uniform distribution, as well as the resulting calculation times. Overall, the KL divergence performance of the new approach was outstanding, the new algorithm outperformed the other methods significantly. However, in terms of computation time, the new approach is expensive.

A possible alteration of the algorithm is to reduce computation time by executing not the best but the first point swap which achieves an improvement of the loss function. Furthermore, the global loss function opens up different interesting research topics. A way to implement boundaries or constrains in the input space are promising subjects for future research, as is the possibility to optimize a design to any desired distribution, not just the uniform one.

References

[1] A. Rimmel and F. Teytaud: "A survey of meta-heuristics used for computing maximin Latin hypercube". In: *European Conference on Evolutionary Computation in Combinatorial Optimization* pp. 25–36. Springer. 2014.

[2] L. Pronzanto and W. G. Müller. "Design of computer experiments: space filling and beyond". In: *Statistics and Computing* vol. 22, no. 10, pp. 681–701. 2012.

[3] T. Ebert, T. Fischer, J. Belz, T. O. Heinz, G. Kampmann and O. Nelles. "Extended deterministic local search algorithm for maximin Latin hypercube designs". In: *Computational Intelligence*, 2015 IEEE Symposium Series pp. 375–382. 2015.

[4] A. Grosso, A. Jamali, and M. Locatelli. "Finding maximin latin hypercube designs by iterated local search heuristics". In: *European Journal of Operational Research* vol. 197, no. 2, pp. 541–547. 2009

[5] T. J. Peter and O. Nelles. “Fast and simple dataset selection for machine learning”. In: *at-Automatisierungstechnik* vol. 67, no. 10, pp. 833–842. 2019

[6] D. W. Scott. “Multivariate density estimation: theory, practice, and visualization”. John Wiley & sons. 2015.

[7] S. Sheater and M. Jones. “A reliable data-based bandwidth selection method for kernel density estimation”. In: *Journal of the Royal Statistical Society* vol. 53, no. 3, pp. 683–690. 1991

[8] B. W. Silverman. “Density estimation for statistics and data analysis” vol. 26, CRC press. 1986.

[9] I. M. Sobol. “On quasi-monte carlo integrations”. In: *Mathematics and Computers in Simulation* vol. 47, no. 2, pp. 103–112. 1998

[10] R. B. Chen, D. N. Hsieh, Y. Hung and W. Wang. “Optimizing latin Hypercube designs by particle swarm”. In: *Statistics and Computing* vol. 23, no. 5, pp. 663–676. 2013

[11] M. D. Morris and T. J. Mitchell. “Exploratory designs for computational experiments”. In: *Journal of Statistical Planning and Inference* vol. 43, no. 3, pp. 381–402. 1995

[12] R. Jin, W. Chen and A. Sudjianto. “An efficient algorithm for constructing optimal design of computer experiments”. In: *Journal of Statistical Planning and Inference* vol. 134, no. 1, pp. 268–287. 2005