Properties and Implementation of Sequential Expansion of Latin Hypercube Sampling for Simulation Design

Crespi Alessandro, Gerosa Davide, Boschini Matteo

July, 2024

Contents

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin luctus finibus euismod. Quisque purus mauris, mollis sed tellus quis, congue hendrerit eros. Aliquam tempus suscipit risus non viverra. Ut pharetra mollis ante, sit amet vestibulum augue laoreet eget. Nunc tincidunt ex sit amet rutrum euismod. Maecenas feugiat, mi aliquam semper rutrum, purus justo imperdiet massa, sed feugiat leo libero sit amet sapien. Mauris sit amet sem rhoncus, hendrerit nunc sed, sagittis felis.

1 Introduction

Simulation design is a branch of Statistics focused on building better simulations to enhance the comprehension of phenomena. These simulations are widely used in mathematics, physics, economics, mechanics, and other scientific fields as tools for proving theories, interpolating real sampled values, and generating predictive models to explore uncharted traits and features, perhaps intuitively or roundly developed in the early stages of the study of a specific problem.

Since the advent of hybrid mechanical-electrical programmable calculators such as IBM's machines, the first of their kind to be really useful in engineering, scientists have used them to perform heavy computations for experiments. In 1969, Kennar and Stone's Computer Aided Design for Experiments (CADEX)[7] proposal for computer-driven experiments led to spread up a broad variety of Computer-based simulation methods.

Eventually, computer-based simulations are a set of strategies that benefit from mathematical modeling techniques based on discrete known points placed in a limited parameter space, hereafter samples, and the computer programmability advantage has been used to design, shape and enhance a specific subset of samples that satisfy the desired properties, hereafter sample sets.

The general concept of computer simulation has been defined in the past few decades, it's based on the following key ideas: taking into account a desired behavior F the experimenters have an interest in, F has to be explored through its N real parameter space; the algorithm takes samples from a standard N-dimensional hyperspace Ω and arranges the sample set for the simulation; afterwar, the simulation is carried out by evaluating F over the sample set and eventually producing a so-called surrogate model. The class of algorithms meant to implement this abstraction is commonly labeled with *sampling methods*.

To the category of sampling methods belongs also the fixed-step (or determined-step) samplers, which adopt a $h(x)$ function that space samples across the hyperspace evenly or deterministically. For instance, consider $h(x)$ constant function or a Chebyshev nodes [10]. For the matter of this paper, we have focused on the sampling methods whose points are drawn with a specific random distribution.

A critical consideration when evaluating sampling methods is the trade-off between exploration and exploitation. An exploration-oriented sample set maximizes the simulation expertise of seeking key features over the studied behavior. Exploration has been depicted as a model-free practice, so that it does not base its own actions on the model (behavior) evolution or any other on-site response. On the other hand, exploitation is an auxiliary mechanism that aims to better assist the simulation by deploying samples in strategic placements that prevent exploration from exceeding the prediction surrogate made upon a key region (such as overshooting an optima or mismatching a discontinuity for a steep slope).

The most iconic sampling method for simulations is the pseudo-random sampling, lightened of every other criteria, namely the Monte Carlo Sampling or MCS (*Metropolis et al. (1987)* [4]) which has been proposed as the fundamental design of sampling methods. Quasi-Monte Carlo methods are a class of sampling algorithms based on Monte Carlo, indeed, but without a proper random drawing of sample points from the parameter space, instead, points are sequentially extracted in order to satisfy one or multiple criteria as best as the computational time required remains acceptable. Many criteria have been theorized and tested; each of them has a proper application context. An updated, summarized list of the most remarkable ones is shown and commented in section 2.4.1.

In the scope of this matter, the authors will focus on the space-filling class of criteria and, particularly, on the one-projection property (also known as non-collapsing property or projective property); The former measures the quality of a sample set to be spread evenly across an hyperspace; the way space-filling is defined determines the final aspect of the sampling. On the same hand, a sample set, in P-dimensional space, admits the one-projection property if and only if the projections of the samples on a specific dimension fall into distinct intervals I_i. These I_i intervals are fixed-width slices of the limited volume of parameter space taken into examination (e.g., an hyper-volume $[0,1)^N$) and the number of intervals is equal to the number of samples. So, the non-collapsing property prevents samples to fall into a busy area (which has been occupied by another sample). Furthermore, given that the number of intervals and the size of the sample set is equal, it does ensure there are no empty intervals across the parameter space. This property is widely known because it is the fundamental property which Latin Hypercube Sampling sets (LHS) are based on, topic of this paper.

The authors used to work with sampling methods for simulation design, more likely LHS designs, in Astrophysics related experiments, such as simulating black hole binary mergers, which obviously requires a massive amount of computational time and many different parameters. The authors often experienced difficulties predicting how long it would take to run a full simulation given N known sampling points in a P-dimensional space. This situation forces them to reserve more machine time on a shared company supercomputer for experiments than they really need. In order to better spend the reserved machine time left after the execution of the first run of sampling points, the authors designed an algorithm that adds up points to the initial

LHS sample set using another one that has been drawn by the expansion algorithm in order to preserve the non-collapsing property of both sample sets together.
Differently, they have experienced another issue related to the accuracy of the surrogate model, result of the simulation consumed after the LHS set. Sometimes, it just happens that the model doesn't satisfy an eventual accuracy threshold. Instead of throwing the simulation away and compute another one, they would like to expand the current initial set with additional points to help, the updated model, converging.

The authors of this paper propose a algorithm called *Expansion of an LHS* that, indeed, takes a already existing Latin Hypercube Sampling' sample set and propose a new set of points samples in the same parameter hyperspace which are suppose to maintain properties stability altogether. The original sample set is referred as *starting set* and the add-on samples as *expansion set*. The whole sample set joined together by both is called *expanded set*.

The paper is structured as follows: section 2 yields a Latin Hypercube Sampling brief history and formal definition; in section 3 is shown the research results and discussion of the expansion task issued; section 4 contains metrics and experimental evaluations on the applied expansion algorithm; section 5 [...].

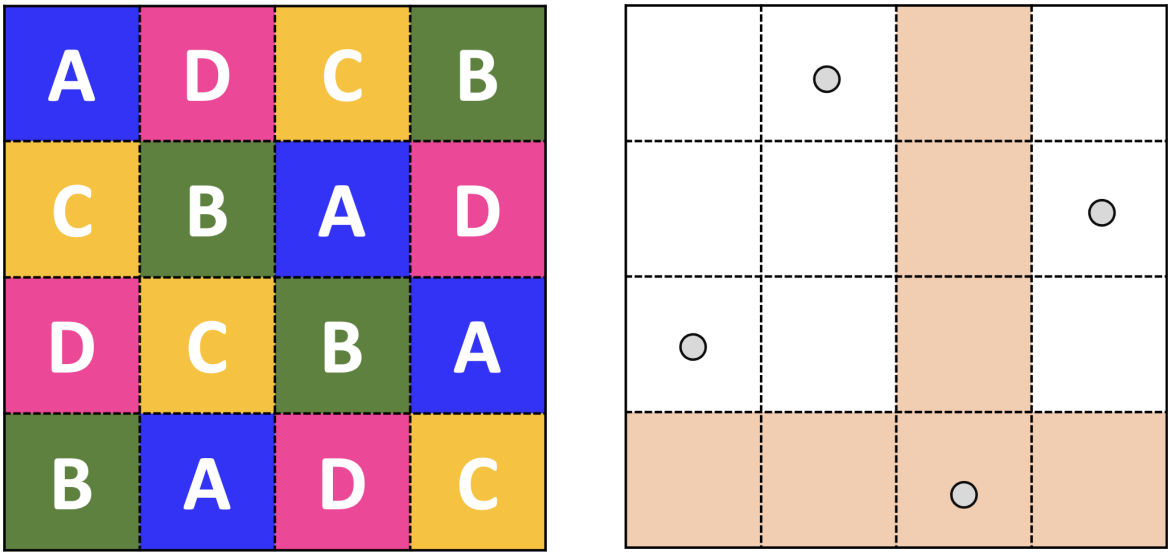
2 Latin Hypercube Sampling

2.1 What is an LHS?

According to the Handbook of Combinatorial Designs [1], the first appearance in history of the "Latin Square" has to be attributed to the Korean mathematician Choi Seok-jeong, who described it, using modern terminologies, as a $N \times N$ matrix with N distinct symbols, appearing N times each but precisely once per type for each row and column. The suffix "Latin" has been inspired by the efforts that Leonhard Euler has put into this topic while defining a general theory for Latin Squares [11] [working on replacing wikipedia] and using Latin letters as symbols to fill the square up with. See Fig.1 for an example with 4 objects.

In the scope of this paper, which doesn't aim to study combinatorial properties of N symbols, we consider the placement of a sample set, which has to obey to the non-collapsing property. The symbols must occur in the matrix exactly N times each. Instead of consider all symbols, we will look to a single one only.

Generally, we can speak of modern Latin Hypercube designs as the Euler's Latin Square's matrix concept (depicted as a grid in Fig.1b) but deeming a more complex multidimensional matrix. N symbols are placed in the same way of earlier, multidimensionally talking: such that each one lies exactly once on each fiber (in literature, a "fiber" of a multidimensional matrix is the general term for a one-dimensional substructure in any dimension [or mode] of a tensor. e.g. In a 2D matrix, the fibers of the first and second dimensions are respectively "rows" and



(a) A Latin 4x4 Square with 4 distinct symbols, both letters or colors either, arranged so that no letter occurs more than once in a row or a column

(b) The skeleton of (a) Latin Square highlighting only the D (or PINK) symbol positions, it's easily noticeable how the positions does not overlap onto each other's row and column both

Figure 1: [RIFAI E NON CITARE] The images were kindly taken from *Sheikholeslami and Razavi et al. (2017)* [9]

"columns"). Moreover, LHS is no longer supposed to place symbols in matrices but, instead, put N random samples on the hyperspace. The *hyperspace* represents the examined multi-parameter space of the interested model, whereof each axis is associated with a different parameter.

By definition of LHS, the parameter hyperspace has a limited span, which is the hypervolume $[0,1)^P$, commonly used in literature. However, some texts use a different standard, and they let the parameter space take place in a $[-1,1)^P$ hypercube. Every parameter axis is sliced into smaller consecutive intervals of width $\frac{1}{N}$ that consequently depict the sub-region where each coordinate of the points is sampled randomly. In anticipation for further definitions, we provide a indexing system for intervals; in any dimension of the hypercube with N intervals, for each i from 0 to $N-1$, the i-th interval's boundaries are:

$$I_i = \left[\frac{i}{N}, \frac{i+1}{N}\right) \tag{1}$$

for further usage, the right term $\frac{i+1}{N}$ of the interval was called *frontier of the i-th interval* which is shared with the left hand term of I_{i+1}. With no loss of generality, the examples and considerations designed by the authors of this paper assess the random distribution to be uniform in all intervals. The uniformly distributed samples can be transformed by associated transformation functions for any other distribution (e.g. Gaussian distribution).

2.2 How to build a Latin Hypercube Sample Set

In this section it has been marked out mathematically the construction of an LHS sample. This description is widely used for introducing the topic on many textbooks and lectures and it takes inspiration from the work of X.Kong at al.[3]. Let $S = \{S_1, S_2, ..., S_N\}$ be the Latin Hypercube sample set with N number of samples, where $\|S_i\| = P$ number of dimensions. It is comfortable to use the matrix notation S_{ij}, whereof rows are the i-th sample and columns, instead, represents the projection of every sample on each j-th dimension. Then, we introduce the sorted index matrix $A = \{a_{ij} = i\}$ of $N \times P$ dimension as a tool for trace the intervals index, its purpose will be clearer soon.
Given an A index matrix, the preliminary design matrix S is given by:

$$S_{ij} = \frac{u_{ij} + a_{ij} - 1}{N} \tag{2}$$

where u_{ij} is a uniform distributed variable U[0,1). This preliminary design has the peculiarity to have the samples always placed on the diagonal of the hypercube $[0,1)^P$. Now, the indexes matrix comes into use, typically shuffling the original A we attain $B = \{b_{ij}\}$ random permutation, which plugging it into eq. (2) describe the uniform random variable S_{ij} living inside the statistical bin identified by the interval index b_{ij}. We explicitly write down the uniform random variable S_{ij} involving eq. (1) boundaries for the ij-th interval as well:

$$S_{ij} \sim U\Big[\frac{b_{ij} - 1}{N}, \frac{b_{ij}}{N}\Big) \tag{3}$$

Along this paper, the authors have used the notation $R = MC(N,P)$ where MC is the Monte-Carlo sample set space of N points in P, hence R is a Monte-Carlo sample set.
Similarly, the expression $S \in LHS(N,P)$ states that S is a Latin Hypercube sample set of the same parameters. By definition of LHS, which stress that it is a quasi-Monte-Carlo sample set distribution, we can write that:

$$LHS(N,P) \subset MC(N,P)$$

2.3 Grade of a Sample Set

For the purpose of this research, the authors introduced a tool to measure how much a Monte Carlo sample set is close to a Latin Hypercube one, namely *grade of a sample set*. This metric (Eq.6) is designed to assign to a sample set S of N elements in P dimensions an index that ranges from the worst possible distributions of points when approaching 0 (it happens if samples overcrowd into the same very intervals and let many more empty) to a fully conventional LHS when grade equals to 1, such that:

$$0 < gr(S) \leq 1 \ , \ S \in MC(N,P) \tag{4}$$

such that

$$gr(S) = 1 \ \Leftrightarrow \ S \in LHS(N,P) \tag{5}$$

The grade formula cast the arithmetical average of the presence (with numerical value 1) of the projection of every sample in each interval for each dimension:

$$gr(S) = \frac{\sum_{j=1}^{P} \sum_{q=1}^{N} min(\sum_{i=1}^{N} \mathbf{1}_{[\frac{q-1}{N}, \frac{q}{N})}(S_{ij}), 1)}{P \cdot N} \tag{6}$$

where $\mathbf{1}$ is the indicator function (see Appendix 6.1), the variable q is another way to represent the sorted index matrix $A = \{a_{ij} = i\}$ for a more immediate comprehension. The *min* operator states that the presence of several samples' projections in a specific interval q doesn't weight up the whole term, which would be at most 1 even if multiple samples lie in q. Hence, the grade formula ignores the overlapping samples onto the same interval, hereafter only *overlaps*.

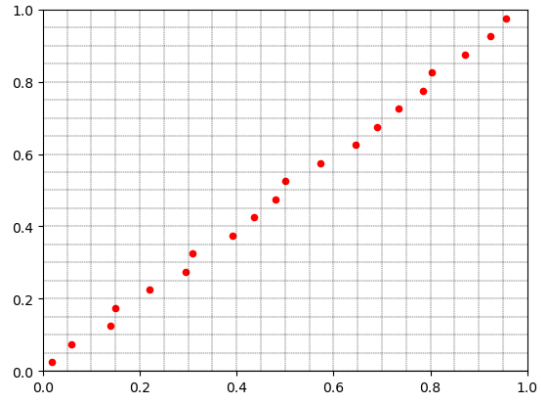
2.4 Additional properties (Work in progress)

In this section, the authors present some very important additional properties that could greatly improve the accuracy - or the quality in other terms - of the surrogate model produced by the consumed simulation.

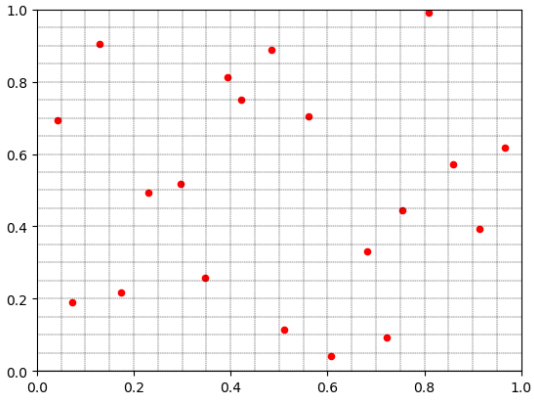
A well known issue with LHS designs is well depicted in Fig.2a. It's obvious that it must be considered a poor simulation design for the most of the experiments. Usually, implementations of LHS add some peculiar properties to the design. We can highlight two superclasses of properties which LHS may be joined with: model-free properties and model-based properties.

2.4.1 Model-free and model-based simulation designs

As the names may suggest, the latter implies the experimental design to involve some the peculiarities of the function to evaluate e.g. the shape that is expected, the initial or boundaries conditions, any well known critical region. Given that this approach is knowledge-driven, "What do we already know about?", can knock down pretty much the computational time required for the whole experiment with eventually high accuracy, but it can be equally easy that results may be biased, based on assumptions that may prove wrong later, or make hard any further effort from other scientists to retake the experiment and confirm the results if the assumptions are not completely clear. On the other hand, the model-free is the formal way to depict a fully inde-



(a) A poor Latin Hypercube sampling in 2 dimension. Despite it fulfills the one-projection rule, it busts the experiment.
The diagonal sampling depicts how insufficient a pure LHS is to achieve a good simulation.



(b) LHS with space-filling property prevents samples to cluster together, low-discrepancy reduces unintentional patterns. This sampling has been generated using scipy.stats.qmc.LatinHypercube utility class. The sample generator bears random permutations of coordinates to lower the centered discrepancy [6].

Figure 2

pendent sampling of the parameter space from the experiment which it has been designed for. Because of not many criteria have been left, all of the properties in this class are based on inter-element relationships, which convey that for each of them has a heuristic meant to quantify how much each sample is well-placed among the others. The shorthand for such a class of criteria is *space-filling properties*.

Beside the many interesting considerations and suggestions that the model-based class of criteria has to offer to mathematicians, the space-filling properties has highly excited the experimenter community through the decades, producing creative and curious features that the samples could experience among each others. A very short list of the most representative ones is shown in Tab.**??**. The most simple one is the [L2? phi di P?] criterion where the assumed metric is the Euler distance between each point in the hyperspace. The non-trivial issue comes along with the computational effort necessary to run through the search tree generated by the maximin (or minimax?) algorithm.

By the way, the characteristic non-collapsing property of the LHS itself is actually a space-filling property. It ensures that, for example, the average distance between the consecutive projections of a group of 3 samples on the same axis is at least 1/2 distance units and at most 3/2 d.u.

Authors	Year	Algorithm	Criteria
Audze and Eglajs	1977	Coordinates exchange	Potential energy

Table 1: Riempio la tabella piano piano che leggo i paper. Ho visto che molte ricerche su LHS usano dare una cronologia sull'utilizzo dei criteri utilizzati throught history

3 *Expansion of a Latin Hypercube Sampling*

The LHS paradigm allows to implement several criteria over a rigorous grid, which forms the basis for the sequential creation of samples. Thus it's frequently utilized in engineering environments for surrogate manufacture of complex systems - see also section 2.4. For example, the LHS is used in hyperparameter tuning/optimization of Machine Learning models (*Koch et al. et al. (2018)* [2]), environmental and water system analysis (*Sheikholeslami and Razavi et al. (2017)* [9]), structural reliability analysis (*Olsson, Sandberg, and Dahlblom et al. (2003)* [5]).

There are ongoing efforts to improve the LHS capability. As previously stated, add criterions first. Then, by rethinking the foundations of the algorithm. The reader has to acquaint that Latin Hypercube technique, widely implemented, is labeled as a *one-stage* algorithm in literature. The fact that all samples are distributed and assessed "on the first run" bestows this adjective. It is relevant to clear out that the actual creation and propagation of points is not properly implemented as the resulting of a single sampling random variables in agreement with Eq.3, but instead it is a sequential drawing of points - or, possibly, a parallel drawing of several ones for optimization reasons - given a desired N_1 number of samples, the newest one has to be pulled out from a pool of optimal candidates in order to improve the criteria applied, such as maximin space-filling distance (see section 2.4.1). The feature "one-stage" highlights that it could be possible to have many more "stages" of the algorithm. It's a game of perspectives: inside a current stage, the policy for drawing a fixed number of data points is aimed at pulling out point given the other ones; instead, a multistage policy is related to a more evolutionary approach, aiming to enhance the sample set stage by stage. Manipulating an already instantiated LHS may sound difficult because LHS is not designed to add points - or, at least, it is not supposed to consider it. Indeed, it is reasonable to assess that by adding points, one by one, over a full grid of N_1 intervals of an LHS, and then reshaping such a grid in $N_2 > N_1$ intervals, will lead to collisions (overlaps), which represents an issue - see section 3.1.

The process that embodies the evolution from a precursor LHS to his next-stage has been called *Expansion* by the authors; the resultant of the expansion process is the so-called *expanded set*. Please refer to Fig.5 for the visual explanation of expansion.

Along this section, the authors used to refer several times to the expansion process without precising how the new samples are going to be placed, the actual process is described in section section 3.3.3.

3.1 The task of multistaging sampling

The multistage approach raises concerns regarding the consistency of the one-projection property, which is valid for one-staged setups. Fig.3 depicts an experiment carried out upon scipy's $S \in LHS(N_1 = 10, P = 2)$ with samples displayed over its appropriate grid of N_1 intervals per dimension (Fig.3a). The experiment consists in evaluating the behavior of the fixed S sample set while the grid is "growing" - intended as creating a brand new grid with a greater number of intervals - one by one for three times. Light grey-colored rows and columns are vacant, meaning that no projections are located there; rows and columns marked in red indicate the location of overlaps. After the first add-on (Fig.3b), the grid shows two overlaps in row #3 and in column number #3 as well; the next stage (Fig.3c) shows overlaps too, four in total, more than before. In the end, Fig.3d has no overlaps at all. The distribution of occurrences of overlaps when the grid grows, depends on the initial sample set. The authors discuss about this topic later in section 4 [⟵ be more specific].

Regarding the LHS directive, the multistage experiment proposed shows that growing the interval grid could lead to a downgrade of the one-projection property in some extent (Fig.3b and Fig.3c) or give a new opportunity to fill the new empty space with a set of brand new samples without compromise the whole set optimality.

3.2 Grade of an Expansion

A precise heuristic that expresses how near the current expansion is to a perfect expansion, given the desired expansion magnitude M and the beginning state, is necessary for developing a new sample set. Since the metric in Eq.6 is no longer sufficient, the variation *expanded grade* has been offered. The purpose of the *expanded grade* is to convert the sample set S of N elements into an index when the sample set is compared against a P-dimensional hypercube grid of $N+M$ intervals per axis - this is where it deviates from Eq.6. Here it follows:

$$gr(S,M) := \frac{\sum_{j=1}^{P} \sum_{q=1}^{N+M} min(\sum_{i=1}^{N} \boldsymbol{1}_{[\frac{q-1}{N+M}, \frac{q}{N+M})}(S_{ij}), 1)}{P \cdot (N+M)} \tag{7}$$

Each I interval contributes to Eq.7 value with a share of:

$$\mu_I = \begin{cases} \frac{1}{P \cdot (N+M)} & \text{if any } x \in I \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

which makes Eq.7 a discrete quantity that ranges from $\frac{1}{N+M}$, if every sample of a non-empty set lies in one single interval per axis, to 1, which represents the perfect LHS expansion.

A perfect expansion for $S \in MC(N,P)$ over a $N+M$ space grid of a number M of intervals (per dimension) empty, given that, by definition of LHS, N samples fall in N distinct intervals

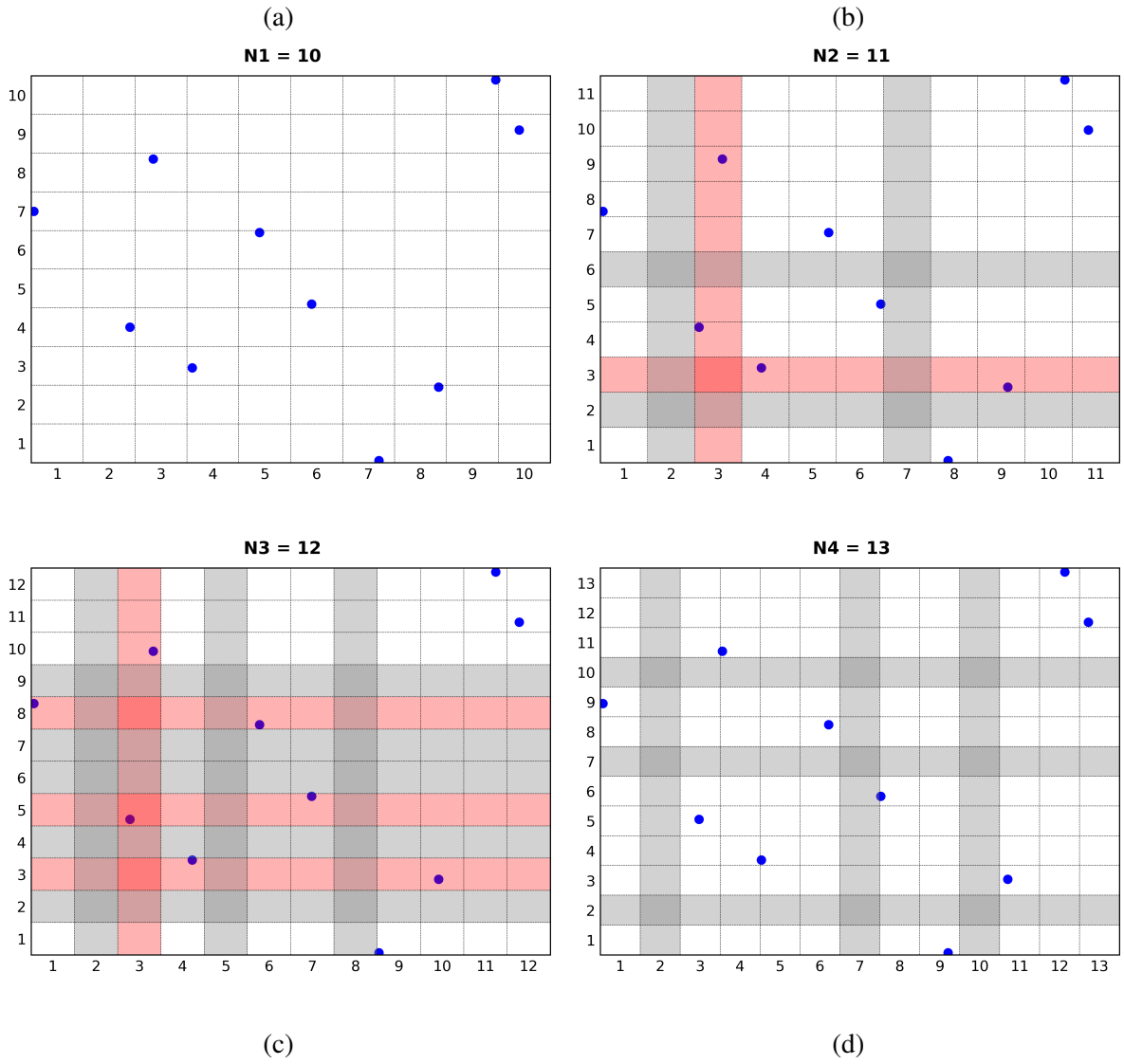


Figure 3: To demonstrate the behavior of the grid, $S \in LHS(N_1 = 10, P = 2)$ sample set is allocated and projected over the grid as it grows one by one. Rows and columns marked in red indicate the location of overlaps, while light grey-colored rows and columns are vacant. In (a) $N_1 = 10$ LHS samples are displayed over a grid of N_1 intervals per axis; (b) shows the first step of the growing grid: two overlaps occur in the horizontal and vertical intervals #3, two intervals per dimension are vacant one of which derives from the expansion vacancy and the other one is caused by the overlap; (c) samples displayed over $N_3 = N_1 + 2$ grid: on the vertical axis there are 3 overlaps and 2 (from the expansion) + 3 (from the overlaps) vacancies; (d) the grown $N_4 = N_1 + 3$ grid have no collisions, then S is a perfect expansion with $M = 3$ expansion magnitude

(per dimension). Then, the only valuable μ shares are given by N intervals while the other M set of intervals are empty. From this statement, we can provide an upper limit for the expanded grade by starting from 1 - the maximum possible index of a perfect LHS sample set - minus the total weight lost during the growing of the grid, which corresponds to the total number of *voids* times Eq.8. The total number of *voids* is equal to $M \cdot P$ because they are evenly spread across each dimension. So, Z is a *perfect expansion* of S with M additional intervals if and only if:

$$gr_{max}(S,M) = 1 - \frac{M}{N+M} \tag{9}$$

The *initial set* $S \in MC(N,P)$ plotted over P hypercube of $M+N$ intervals (per dimension) is called *perfect expansion* if and only if S expanded grade Eq.7 is equal to the *upper expanded grade limit* Eq.9.
As the name may suggest, the perfect expansion is the best possible candidate to produce an optimal non-collapsing sample set, so an LHS, as result of an expansion process described in section 3.3.2.

3.3 The expansion process

The expansion task was initially handed at the very beginning of section 3.1 as a evolutionary process which augments the S starting sample set to a next-stage state with increased number of elements M, placed as better as it can to maximize criteria. However, we show in section 3.2 that an expansion may demote the non-collapsing property of the resultant expanded set Z, which can be measured with the metric Eq.7. In this section the authors propose how to place the new samples to achieve maximum "LHS-ness" over any M expansion magnitude needed and introduce its potentialities for future researches - see more in section 5.

In first place, the proposal is delivered by studying the basic case of a perfect expansion in section 3.3.1, then extend to the general case with any non-perfect expansion outcome.

3.3.1 State case - Perfect Expansion

An initial instanced sample set S of N elements can be perfect expanded if and only if S after the M-th growing step of the P-hypercube grid has the maximum grade (Eq.9). Furthermore, Fig.3d well depicts what the experimenters would expect before setting down an expansion set E. We notice the best candidate extent is likely the empty intervals (in light grey) because they do not interfere with other well-formed intervals. It is also imperative that the new samples do not overlap onto each other over the vacant space (*vacancies* or *voids*, as we previously called them).

First of all, it's necessary to trace each void's index. Here it comes into use again the permuted index matrix, previously used in section 2.2 to scatter the samples across the hyper-

parameter space. The matrix of voids $P \times M$ is composed by the row vectors:

$$\mathbf{V}_j = \left(q : \nexists\, x \in S_{ij}\ s.t.\ x \in \left[\frac{q-1}{N+M}, \frac{q}{N+M} \right) \right) \quad , \quad \forall j = 1...P \tag{10}$$

which should be element-wise permuted along each row to prevent Fig.2a diagonalized situation. Then, the expansion set E collects all the newly generated samples which are a distribution likewise Eq.3 but using V_{ji}.

In the end, the initial S set is concatenated with expansion set E, originated from the voids of the $N+M$ growing grid, in a perfect expansion, that guarantees the fulfilling of the non-collapsing property (grade = 1).

A perfect expansion is a intuitively rare event: each sample has to avoid overlaps. This could eventually happen if two sample projections are spaced more than $\frac{1}{N+M}$ apart (new size of the intervals), But if they are closer, another condition has to be fulfilled: for any M close enough to N, a *critical span* exists across the *i-th frontier* (see right after Eq.1 for definition) - which shares its boundary with the next interval. The critical span is the intersection area between two neighbor samples intervals and the eventual interval raised across them (Fig.4b). Given that LHS is not supposed to consider further additional points, it might happen that samples are placed in critical areas (after a M growing step). An example of how can a couple of collapsing samples can occur is shown in Fig.4. In the example, the first sample (in Fig.4b) has been generated in the critical area that is the intersection between $\sim$*2* and #1 original interval. On its hand, the second sample lies upon the same critical area between $\sim$*2* and #2 original subspace. Hence, in the example the two samples are overlapped.

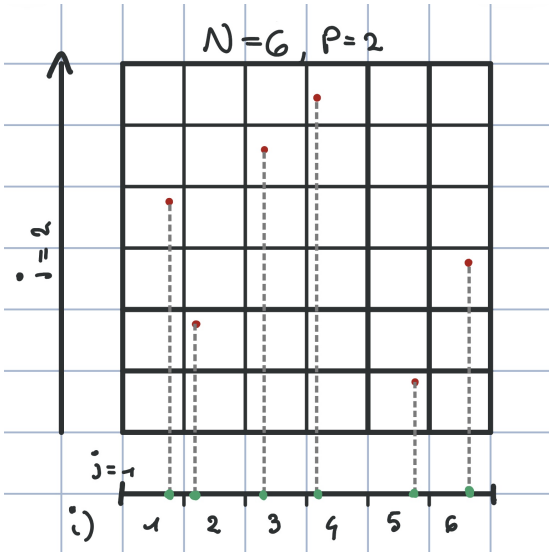
The number and displacement of overlaps and vacancies are pretty hard to predict: they are strongly correlated to the Eq.2 sample random distribution. However, we experimentally observed that some expansions are peculiar: exactly all expansions with multiples of N are always perfect. That's because critical spans - again, that are given by the intersection of the specific new intervals that cross the old frontiers with the previous intervals - are always void. Take Fig.4b but with grown binning size equals to $\frac{1}{12}$ instead of $\frac{1}{8}$ (so, with $N = 6$ then $M = 12$): new intervals divide perfectly the old ones, without ever crossing any frontier. This concept is properly explained in Appendix 6.2. Explicitly:

$$\forall K \in \mathbb{N}^+ \ : \ S \in LHS(N,P) \Rightarrow gr(S, K \cdot N) = gr_{max} \tag{11}$$

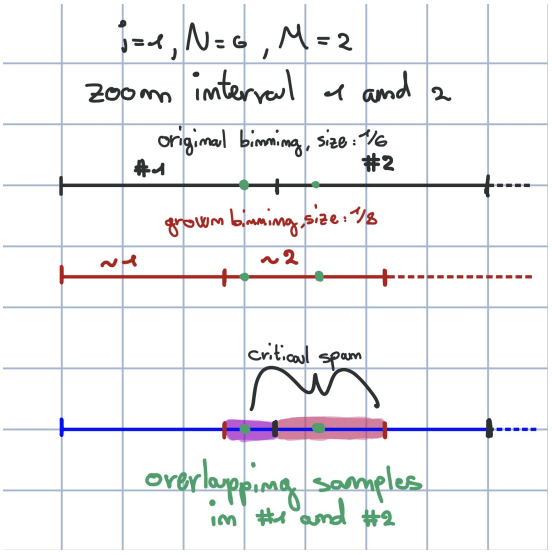
[I'm not exactly sure if it should be $\Rightarrow$ or $\Leftrightarrow$]

3.3.2 State case - General Expansion

During the upscale of S from N to $N+M$ number of elements, it comes along with a variable number of overlaps on j-th axis O_j. In section 3.2 the authors stress the relation between tuples



(a) An LHS of $N = 6$ samples expanded for $M = 2$. on the horizontal axis are projected the horizontal component of all samples. [SKETCH - ne farò in digitale di migliori]



(b) Zooming in #1 and #2 interval, it's show how both shares a critical area (on the bottom) whereof each one may have been distributed in it. [SKETCH - ne farò in digitale di migliori]

Figure 4

$\mathbf{V}_j$, M, and the amount of collisions O_j on each j-th dimension. Specifically:

$$\| \mathbf{V}_j \| = O_j + M \tag{12}$$

whereof the overlaps count equals to zero, the expansion has been perfect (see section 3.1). In a general case of expansion, the overlaps amount is most likely not equal to zero in some dimensions.

The case study implies that the creation of the expansion set iE is not trivial anymore because of the irregularity of the vacancies set. By referring to Eq.12, unlike what was stated before, V would probably be no matrix at all but, instead, a set of heterogeneous tuples. If V components was homogeneous, it would have been a inherited matrix. The number of $\mathbf{V}_j$ interval indexes would likely be more than M sample's projections to commit. The expansion algorithm has to pick up a reasonable subset of M void entries, and thus to discard an amount of intervals equal to the number of overlaps O_j. Therefore, given the sub-hyperspace settled by the joined selected voids, in order to plot an M amount of new samples, it will pick up an $P \times M$ submatrix (that mimics Eq.10) of V set. The submatrix should be handled being aware that it would effect the samples layout which may better improve another coherent criteria chosen (such as low-discrepancy or Maximin space-filling).

The selection process of vacancies from an irregular V voids set is described by a function $\sigma : N \times P,\ M \to P \times M$, namely *perfectify* or *vacancy reduction* (for the matter of giving names

to anything).
In this section, σ reduce function trivially picks up M intervals randomly per dimension and build up a permuted Eq.10 vacancies matrix, which will be plugged into Eq.2 to produce an expansion set.

In other words, σ criterion extracts from each j-th axis of the vacancies set $\mathbf{V}_j$ a fixed M number of elements which are going to compose the sub-hyperspace where M samples will be placed using at least the non-collapsing property. The function σ discards O_j number of intervals (Eq.12), then creating an amount of voids of the same quantity. Hence, the number of overlaps O_j determines the number of void intervals after the expansion is consumed.

In this paper, the term *quasi-LHS* refers to a non full-graded in-th stage of expansion descended from a proper LHS.

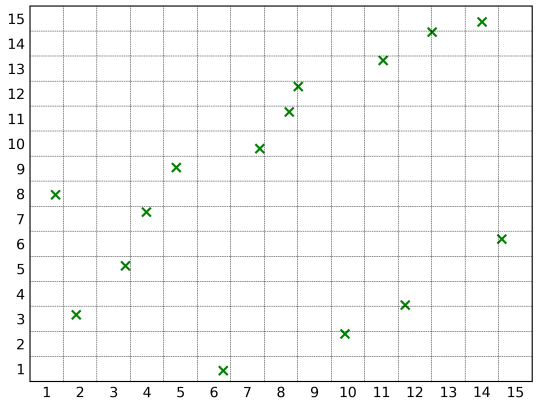
3.3.3 The eLHS algorithm

The LHS expansion algorithm, namely *eLHS*, push a starting Latin Hypercube S to the next stage $Z = eLHS(S,M)$ that maximizes at most the non-collapsing property, along with other eventual criterions.

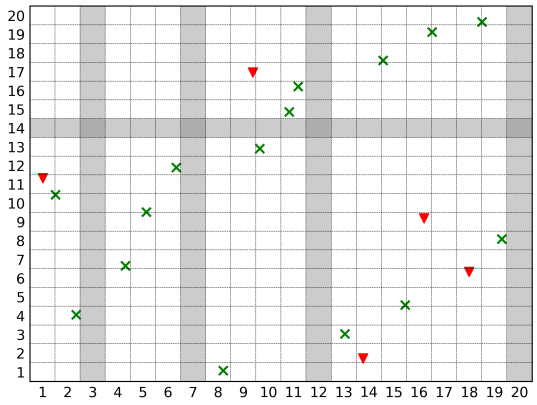
1. Instance a V vacancies set of P tuples - which may have different lengths (Eq.10) because every dimension has an arbitrary number of voids (Eq.12). The list of all indexes of the $N+M$ grid is filtered accordingly with Eq.10.
   Visually, the algorithm re-bins the original S grid to achieve $N+M$ intervals. By plotting it against S, the reader can visualize where samples overlap and where there are voids.

2. Reduce V vancancies set to a suitable indexes matrix $V' \in Matrix(P,M)$ by extracting from each $\mathbf{V}_j$ tuple M elements - which are going to compose the expansion binning grid - using σ reduction criteria (see section 3.3.1). If there are no overlaps (meaning S has maximum expanded grade Eq.9) then V is implicitly equal to matrix V', so no reduction criterion is required to be applied.
   We propose a paradigm which σ can be built on. Reduce (abuse of notation) the issue to an harder problem with integer constraints so that solutions are the void interval's indexes. Before generating optimal samples (in step 3.), select the optimal void intervals by considering a trivial "puppet" sample that lies in each possible combination of void intervals. Then apply Branch&Bound methodology [TODO: find good refs for BnB] to find the best integer solution(s).

3. Generate new points over the sub-hyperspace outlined by the permuted V' indexes matrix. Currently, Scipy doesn't implement the instancing of a LHS over a discontinuous space yet.
   The eLHS' implementers should achieve drawing optimal data from a discontinuous

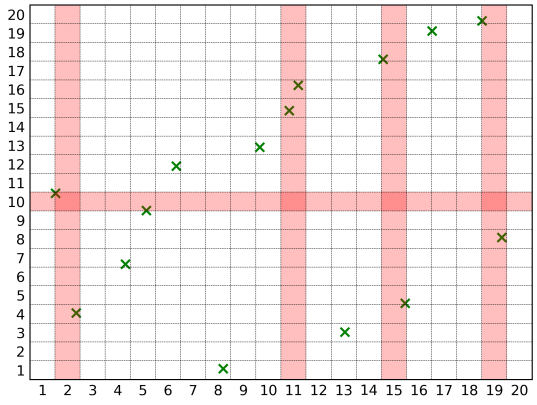


(a) First-stage LHS of $N = 15$ samples in $P = 2$ generated with scipy's qmc library.

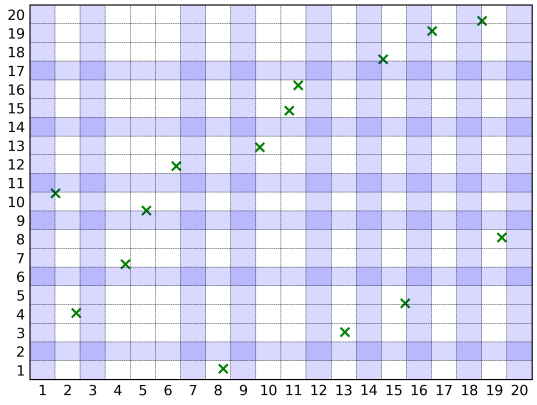


(b) Expansion of (a)'s LHS with section 3.3.3 eLHS algorithm given $M = 5$ new samples. Note that the light grey marked intervals are empty and, according to section 3.3.2, they are related with the overlaps distribution in (c)

Above is shown a two-staged quasi-LHS. (a) is the first original LHS and (b) is next stage expansion of it.



(c) Red intervals have two sample's projections in it and break the non-collapsing property.



(d) Blue intervals are empty. They represents the best candidate spots to place new LHS samples. Every interval all together has been referred as the sub-hyperspace of vacancies.

Re-binned (a)'s grid with $N + M$ intervals and plotted against the starting LHS.

Figure 5

space on their own, based on desired additional criteria and time complexity requirements. However, we suggest the following algorithm structures.

(a) This procedure was inspired by Shang et al.[8] work based on the pioneering CADEX research (*R. W. Kennard et al. (1969)* [7]). It was designed to be fast and flexible given some appropriate criteria to optimize:

i. set the variable $\Lambda = M$, iteration index $k = 1$ and initialize the expanded set E empty;

ii. generate Γ_k a random pool of, let's say, α_k number of new random sample points. The hyper-parameter α_k should be much greater than Λ. We would recommend a fast-and-reliable MCS;

iii. ignore all points that fall outside V' space;

iv. from Γ_k select a subset of optimal samples γ_k of $\lambda_k \leq \Lambda$ number of points.

† If $\lambda_k = 1$ it is like drawing the very optimal point each iteration;

† If $\lambda_k > 1$ the algorithm should yield a bunch of samples which satisfies or optimize any eventual sub-property. Definitely, the basic property that γ_k has to satisfy is the non-collapsing property after being joined with $S \cup E$;

v. pop out from V' the intervals where $gamma_k$ elements lie in;

vi. append the optimal subset: the sample set $E = E \cup \gamma_k$;

vii. set $\Lambda = \Lambda - \lambda_k$.
If $\Lambda = 0$, return E.
Otherwise, set $k = k + 1$ and go to step (ii).

(b) The other way to generate E recalls what was said in section 2.4 [not said yet, I'll do it] about search trees. Indeed, as many sampling methods implementations adopt, use a search tree enhances any desired criteria by a lot but trade-offs with time complexity.
However, following there are some further suggestions that the authors have considered notable:

i. A basic search three, branches take a random not yet used vacant space from V' (a P tuple of intervals for each dimension) and shoot K samples, then selects whose the optimal. The deeper the tree goes, the higher is the number of samples drawn. On the leaves there are every expansion sets E computed, the number of leaves is $(M!)^P$. The time complexity is $O\left(K^M \cdot (M!)^P\right)$. This search tree should came along with a reasonable branch pruning rule that reduces computational time and approximate the optimal solution.

ii. If multiple criterions are given, build a multi-agent adversarial search algorithm [drop some references from AI here] where each agent (one for each property)

tries to optimize its own criteria against the other's (without compromising the global score, the actual objective function linear composed by every criteria).

4 Experiments

...

5 Conclusions

...

6 APPENDIX

6.1 Indicator function

The indicator function ***1*** of a set **A** indicates whether the input belongs to A or not, specifically:

$$\mathbf{1}_A(x) := \begin{cases} 1 & if\, x \in A \\ 0 & if\, x \notin A \end{cases} \tag{13}$$

As in the matter of sectioning a space into continuous intervals in the shape of [a, b), it is useful to redefine the indicator function as an operation that occurs with the boundaries of A using the Heaviside step function which does not involve set operators but only logical ones. It's important to remark that it doesn't matter what happens precisely on the boundaries. The Heaviside function is defined:

$$H(x) := \begin{cases} 1 & if\, x \geq 0 \\ 0 & if\, x < 0 \end{cases} \tag{14}$$

So the indicator function can be also produced:

$$\mathbf{1}_{[a,b)}(x) = H(x-a) \cdot H(b-x) \tag{15}$$

6.2 Perfect expansion case: Multiples of N

Given $S \in LHS(N,P)$ set with $N,P \in \mathbb{N}$ respectively number of samples and dimensions, by definition (Eq.5), has maximum grade. Experimentally, it has been observed that the only well-known perfect expansions (maximum expanded grade Eq.7) of S are those which the M incre-

ment of the sample space is a multiple of N. [cita figura di evoluzione del grado in Experiments section]

The reason of this behavior is linked to the unique distribution characteristic that intervals the $M = K \cdot N$ growing grid and the initial S' grid share: the former, actually, sections - with abuse of notation - *perfectly* the latter's intervals space. (*) By considering the same amount of samples N for both situations, we can state that the sum of the shares along any dimension of the regridded set is equal to the initial's one, which is equal to N.
Let's stress out the grade and expanded grade equations:

$$gr(S) = \frac{1}{P \cdot N} \cdot \sum_{j=1}^{P} \sum_{q=1}^{N} min(\sum_{i=1}^{N} \mathbf{1}_{[\frac{q-1}{N}, \frac{q}{N})}(S_{ij}), 1) = 1$$

the total share of all intervals along a fixed dimension is:

$$\sum_{q=1}^{N} min(\sum_{i=1}^{N} \mathbf{1}_{[\frac{q-1}{N}, \frac{q}{N})}(S_{ij}), 1) = N$$

Given the unique property cited above (*), also given $N + M = (K+1) \cdot N$, the expanded grid should have the exact same amount of share:

$$\sum_{q=1}^{(K+1)\cdot N} min(\sum_{i=1}^{N} \mathbf{1}_{[\frac{q-1}{(K+1)\cdot N}, \frac{q}{(K+1)\cdot N})}(S_{ij}), 1) = N$$

then, the expanded grade should be:

$$gr(S, K \cdot N) = \frac{1}{P \cdot (K+1) \cdot N} \cdot \sum_{j=1}^{P} \sum_{q=1}^{(K+1)\cdot N} min(\sum_{i=1}^{N} \mathbf{1}_{[\frac{q-1}{(K+1)\cdot N}, \frac{q}{(K+1)\cdot N})}(S_{ij}), 1) = \frac{1}{K+1}$$

which corresponds exactly to the upper limit of an $M = K \cdots N$ growing step grid for an S LHS set Eq.9:

$$gr(S, K \cdot N) = \frac{1}{K+1} = 1 - \frac{K \cdot N}{(K+1) \cdot N} = gr_{max}$$

Hence, any S expansion of $K \cdot N$ magnitude has maximum expanded grade, which makes them *perfect expansions* by definition.

References

[1] Charles J. Colbourn. *Handbook of Combinatorial Designs*. 2nd. CRC Press, 2006.

[2] Patrick Koch et al. "Autotune: A Derivative-free Optimization Framework for Hyperparameter Tuning". In: KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 443–452. ISBN: 9781450355520. DOI: 10.1145/3219819.3219837. URL: https://doi.org/10.1145/3219819.3219837.

[3] Xiangshun Kong, Mingyao Ai, and Kwok Leung Tsui. "Design for Sequential Follow-Up Experiments in Computer Emulations". In: *Technometrics* 60.1 (Apr. 2017), pp. 61–69. DOI: 10.1080/00401706.2016.1258010. URL: https://doi.org/10.1080/00401706.2016.1258010.

[4] N. Metropolis. "The beginning of the Monte Carlo Method". In: *Los Alamos Science* Special Issue (1987).

[5] A. Olsson, G. Sandberg, and O. Dahlblom. "On Latin hypercube sampling for structural reliability analysis". In: *Structural Safety* 25.1 (2003), pp. 47–68. ISSN: 0167-4730. DOI: https://doi.org/10.1016/S0167-4730(02)00039-5. URL: https://www.sciencedirect.com/science/article/pii/S0167473002000395.

[6] *Python 3.x - Scipy Documentation*. URL: docs.scipy.org/doc/scipy/reference/generated/scipy.stats.qmc.LatinHypercube.html.

[7] L. A. Stone R. W. Kennard. "Computer Aided Design of Experiments". In: *Technometrics* 11.1 (1969).

[8] Boyang Shang and Daniel W. Apley. "Fully-sequential space-filling design algorithms for computer experiments". In: *Journal of Quality Technology* 53.2 (2021), pp. 173–196. DOI: 10.1080/00224065.2019.1705207.

[9] Razi Sheikholeslami and Saman Razavi. "Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models". In: *Environmental Modelling & Software* 93 (2017), pp. 109–126. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2017.03.010. URL: https://www.sciencedirect.com/science/article/pii/S1364815216305096.

[10] G. W. Stewart. *Afternotes on Numerical Analysis*. SIAM, Jan. 1996, pp. 151–155. URL: http://books.google.ie/books?id=VEHBOUOAL-EC&printsec=frontcover&dq=Stewart,+Gilbert+W.+(1996).+Afternotes+on+Numerical+Analysis&hl=&cd=1&source=gbs_api.

[11] Wikipedia. *Latin Square*. URL: en.wikipedia.org/wiki/Latin_square.