Avoiding L_∞ Discrepancy Optimization

François Clément

joint work with Carola Doerr, Luís Paquete and Kathrin Klamroth

UNIVERSITY of WASHINGTON

UNIVERSIDADE D COIMBRA

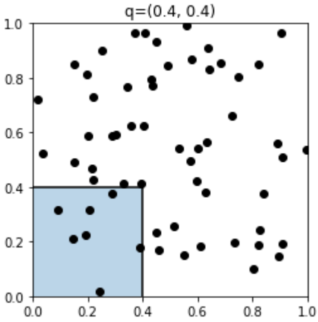
BERGISCHE UNIVERSITÄT WUPPERTAL

The L_∞ star discrepancy

L_∞ star discrepancy

For P a point set in $[0;1]^d$,

$$d_\infty^*(P) = sup_{q \in [0;1)^d} \left| \frac{|P \cap [0,q)|}{|P|} - \lambda([0,q)) \right|.$$



Local discrepancy:

$D(q,P) = 0.044$

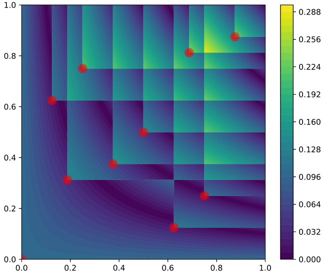
Very small instances: optimal values

- The minimal star discrepancy, $d_\infty^*(n,d)$, is the best possible L_∞ star discrepancy value for a point set of size n in dimension d.
- [White, 1977] gave point sets for $n \leq 6$ in dimension 2
- 1-point sets for any d have been solved by [Pillard, Cools and Vandewoestyne, 2006], extended to 2 points by [Larcher and Pillichshammer, 2007]
- For the periodic L_2 discrepancy, [Hinrichs and Oettershagen, 2016] solved the problem for $n \leq 16$

Can we provide point sets matching $d_\infty^*(n,d)$?

Computing the star discrepancy

Calculating the discrepancy is a discrete problem, maximal values can only be reached on a grid defined by the points.



Computing the star discrepancy

- From the discrete "positions-grid": $O(n^d)$, $O(n^d/d!)$ if we only count **critical boxes**
- Best known algorithm: $O(n^{1+d/2})$ by Dobkin, Eppstein and Mitchell (1996)
- Best heuristic in higher dimensions: **Threshold Accepting** algorithm by Gnewuch, Wahlström and Winzen (2012)

Optimal constructions[1]

Optimal L_∞^* star discrepancy set

Given an integer $n \geq 1$ and a dimension $d \geq 2$, find a set P of size n in dimension d of discrepancy $d_\infty^*(n,d)$.

- Our two non-linear programming formulations rely on the grid structure of the discrepancy calculation.

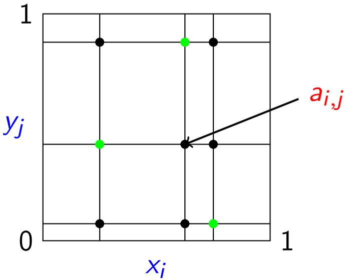
[1] Constructing Optimal L_∞ Star Discrepancy Sets, F.C, C. Doerr, K. Klamroth and L. Paquete, Proceedings of the American Mathematical Society Series B, 2025

The assignment formulation

- We are trying to determine where the n points should be to minimize the discrepancy value, by placing the underlying grid and deciding which points should generate it.
- The objective z represents the discrepancy of the selected set.
- Variables x_i correspond to the ordered x coordinates of the points. The y_j to the ordered y coordinates.
- The binary variables $a_{i,j}$ correspond to the selected grid-points.

The assignment formulation

We split the problem in two parts: finding the coordinates and finding an assignment.

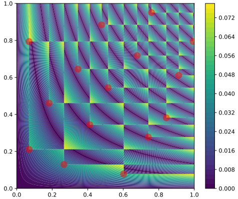


First formulation

$$\min \; z$$

$$\text{s.t.} \; \frac{1}{n}\sum_{u=1}^{i}\sum_{v=1}^{j} a_{uv} - x_i y_j \leq z$$

$$\frac{-1}{n}\sum_{u=1}^{i-1}\sum_{v=1}^{j-1} a_{uv} + x_i y_j \leq z$$

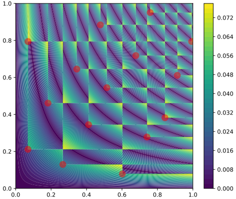


For each box, we need:

- the number of points inside: $\sum_{u=1}^{i}\sum_{v=1}^{j} a_{uv}$
- its volume: $x_i y_j$

First formulation

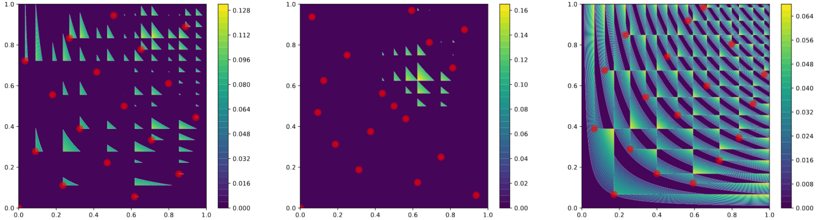
$$\min \; z$$
$$\text{s.t. } \frac{1}{n}\sum_{u=1}^{i}\sum_{v=1}^{j} a_{uv} - x_i y_j \leq z$$
$$\frac{-1}{n}\sum_{u=1}^{i-1}\sum_{v=1}^{j-1} a_{uv} + x_i y_j \leq z$$



For each box, we need:

- the number of points inside: $\sum_{u=1}^{i}\sum_{v=1}^{j} a_{uv}$
- its volume: $x_{2i-1}x_{2j}$

Fibonacci vs Sobol' vs Optimal



Left: Fibonacci 18; **Middle**: Sobol' 18; **Right**: Optimal 18

A computational obstacle

- We can solve these models to optimality for up to around 20 points in 2d.
- Similar models can be found for other discrepancy measures.
- Going further: focus on half of the problem! Either we fix the grid preemptively, or the $a_{i,j}$.

The correct choice: Fixing the permutation [2]

- Fixing the coordinates does not help at all.
- Fixing the assignment (equiv. permutation) makes the problem *much* easier to solve.
- Can solve for up to 500 points, and the main obstacle is reading the model and presolving.

How to choose the correct permutation?

[2] F. C., Carola Doerr, Kathrin Klamroth, Luís Paquete. Searching permutations for constructing uniformly distributed point sets, PNAS 2025

A natural candidate: shifted Fibonacci sets

For some shift $j \in \mathbb{N}$, $P = \left\{ \left(i/n, \{\phi(i+j)\} \right) : i \in \{1, \ldots, n\} \right\}$

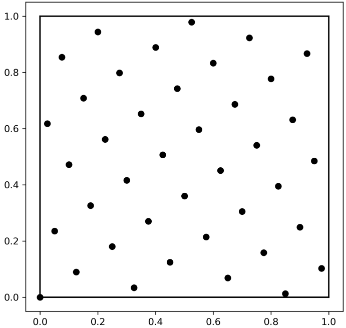


Figure: The unshifted Fibonacci set for 40 points

A natural candidate: shifted Fibonacci sets

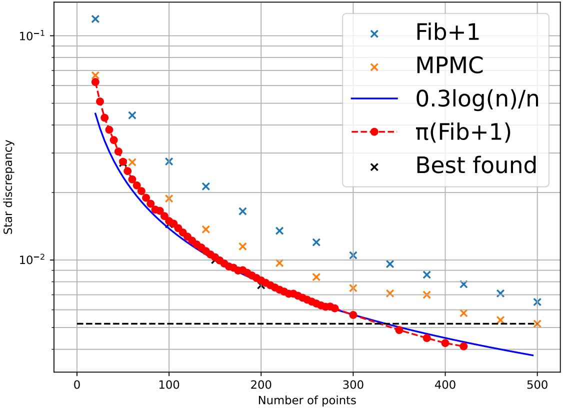


Figure: Best L_∞ star discrepancy values obtained by taking the permutation from the Fibonacci set *offset by 1*.

Structure of the Fibonacci permutations

- Cycle structure of the permutation is *very* regular.
- For $n = F_k$ many points, either there are only cycles of length 2 and fixed points, or cycles of length 4 and a unique fixed points/cycle of length 2.
- Similar observations are possible for quadratic irrationals.
- Is this even relevant for discrepancy?

Key questions

- How can we determine if a given permutation can lead to a low-discrepancy point set?
- Can we replace the discrepancy optimization by an optimization on the permutation? What kind of structure are we looking for?
- What should be the starting point in higher dimensions? for other discrepancy measures?

A separate path: the L_2 discrepancy

L_2 star discrepancy

For P a point set in $[0;1]^d$,

$$d_2^*(P) = \left(\int_{[0,1)^d} D(q,P)^2 dq \right)^{1/2},$$

where $D(q,P)$ is the local discrepancy.

- The main advantage of the L_2 discrepancy is that it is very easy to compute using the Warnock formula [Warnock, 1972].

$$(d_2^*)^2(P) = \frac{1}{3^d} - \frac{n}{2^{d-1}} \sum_{i=1}^{n} \prod_{k=1}^{d} (1-(x_k^{(i)})^2) + \sum_{i,j=1}^{n} \prod_{k=1}^{d} (1-\max(x_k^{(i)}, x_k^{(j)}))$$

Optimizing the L_∞ discrepancy via the L_2 discrepancy

- Rusch et al. (2024)[3] use GNN to optimize point placement for the L_2 discrepancy, and they obtain excellent sets also for the L_∞ star discrepancy!
- Preliminary work suggests that gradient descent on a smoothed version of the L_2 discrepancy also leads to similar results as long as the starting set is "good".
- A subset selection approach also leads to low L_∞ sets.

[3] T. Konstantin Rusch, N. Kirk, M. M. Bronstein, C. Lemieux and D. Rus, Message-Passing Monte Carlo: Generating low-discrepancy point sets via Graph Neural Networks, 2024

A good surrogate in low dimensions

- The "base" L_2 discrepancy can be used to optimize for L_∞ only for $d \leq 5$.
- Strongest results in dimension 2.
- Generalized discrepancy is the better choice in higher dimensions, but not a miracle solution.

What would be the appropriate L_2 surrogate function?

A greedy approach: the Kritzinger sequence

Kritzinger, 2022

Given a starting point p_1, we define the sequence $P = (p_i)_{i \in \mathbb{N}}$, such that

$$p_k := \arg \min_{p \in [0,1)^d} d_2^*(P_{k-1} \cup \{p\}),$$

where $P_{k=1}$ is the set containing the first $k-1$ elements of P.

In 1d, this comes down to finding

$$\arg \min_{p \in [0,1)} (n+1)(1-p^2) + (1-p) + 2\sum_{i=1}^{n}(1-\max(x_i, p))$$

Constructing the sequence

- In one dimension, the next point in the sequence can only come from a "small" set of rationals.
- Linear-time to compute the next point in the sequence.
- Dimensions 2 and 3: exact MILP models, or heuristics.

A million points

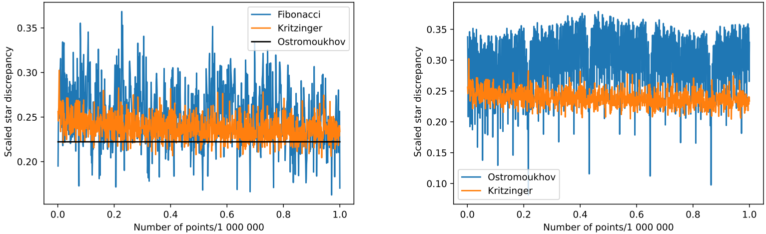


Figure: One million points with the Kritzinger sequence, compared to the Fibonacci sequence and the Ostromoukhov sequence.

5 million points!

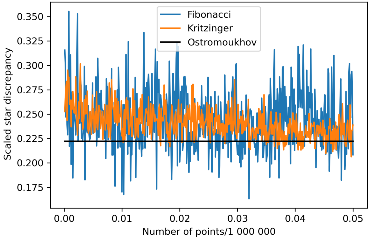


Figure: Five million points for the Kritzinger and Fibonacci sequences.

2d and 3d

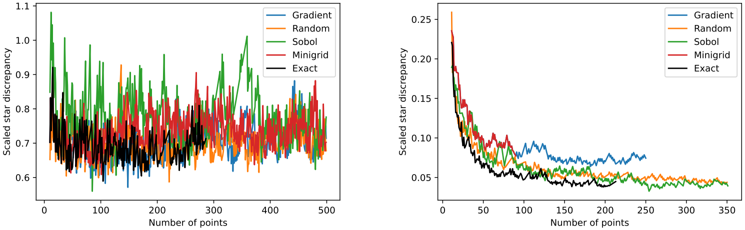


Figure: Performance of the Krizinger sequence in two and three dimensions.

The Kritzinger sequence

The Kritzinger sequence:

- has on average lower discrepancy than the best low-discrepancy sequences.
- is more stable.
- corrects a bad starting set! Even starting with 1 000 badly placed points, the sequence will become low-discrepancy *very* quickly.

It has not been shown that the sequence is low-discrepancy!

Key questions

- Can we determine what permutations lead to low-discrepancy sets? Are there specific structures we should look out for?
- Why do near-optimal L_2 discrepancy sets have near-optimal L_∞ sets?
- What function should be used to generalize this to higher dimensions?
- What makes the Kritzinger sequence so good? Can we show it is low-discrepancy?

Key questions

- Can we determine what permutations lead to low-discrepancy sets? Are there specific structures we should look out for?
- Why do near-optimal L_2 discrepancy sets have near-optimal L_∞ sets?
- What function should be used to generalize this to higher dimensions?
- What makes the Kritzinger sequence so good? Can we show it is low-discrepancy?

Thank you for your attention!