Finite Samplings on $[0,1]^n$

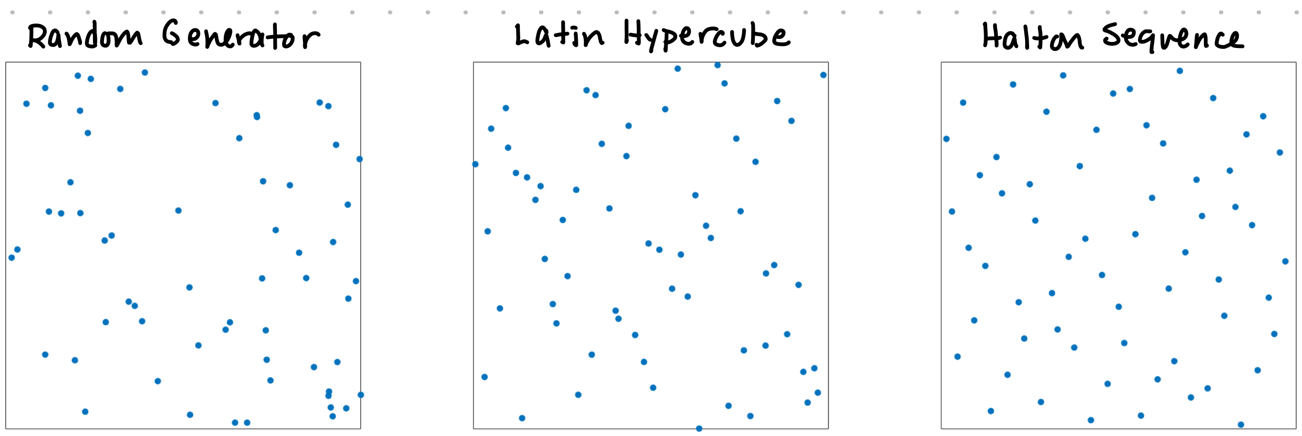
Task: Find a set of points $\{x_k\}_{k=1}^{P}$ in $[0,1]^n$ satisfying

1. $\{x_k\}_{k=1}^{P}$ are uniformly distributed.
2. $\{x_k\}_{k=1}^{r}$ are uniformly distributed for all $r < P$.
3. The process is repeatable.
4. The process is independent of computing platform.

Motivation:

- I have a total budget of P function evaluations and sampling these points will be my entire algorithm.
- my algorithm needs P points to initiate (e.g. Nelder Mead) and it would be good to start with a good sampling.
- Every so often, I would like to check P points in my domain to help me escape local minimizer basins.
- I need non-stochastic methods for repeatability and testing purposes.
- If I try for P points, but only get $r < P$ points, I want the sampling to still be uniform (useful).

There are many strategies and research is ongoing. We will consider only three options - not all of which satisfy our desired properties. (example: $n=2$, $P=24$)



Random Generator

Given : n, p, s

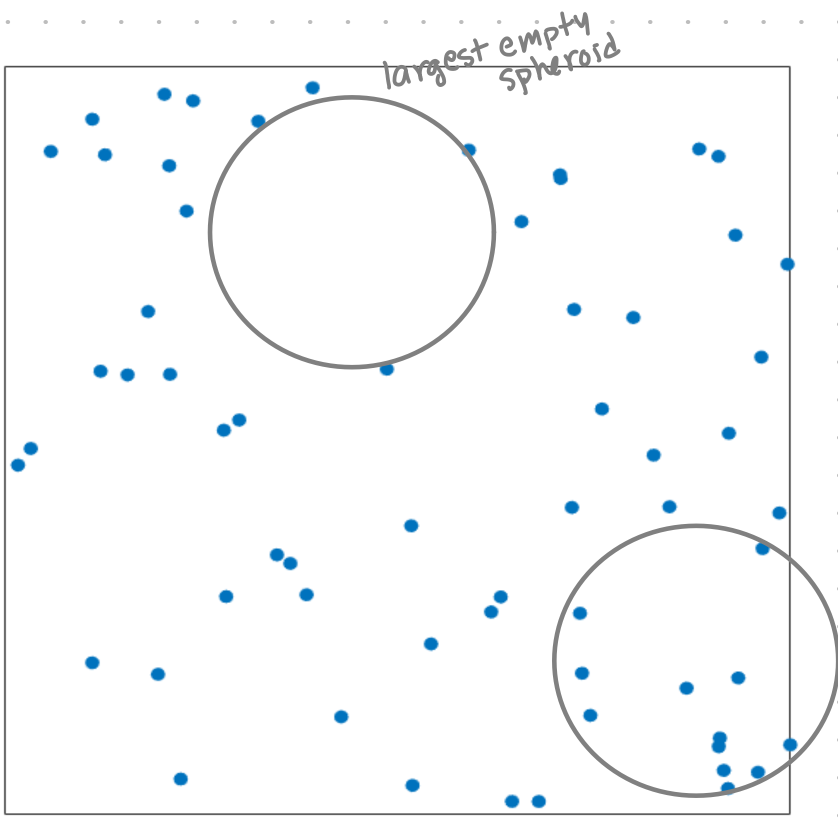
Set random number generator seed to s.

Request $n*p$ random values uniformly drawn from $(0,1)$.

Arrange the values (by predetermined order) into $n \times p$ array A.

The ordered columns of A represent $\{x_k\}_{k=1}^{p} \subseteq (0,1)$.

- Repeatable through seed s
- May not be platform independent
- uniform in large p limit



62 points on $[0,1]^2$

- distributed across domain
- not visually uniform with large empty spaces and point clusters.

Same size spheroid but containing 12 points.

Latin Hypercube Sampling

Given : n, p, s

Set random number generator seed to s.

Create n random permutations of $[1,2,3,\dots,p]$.

Arrange these row vectors, by predetermined order, into $n \times p$ array B.

Create (by previous algorithm) $n \times p$ array R.

Compute $A = \frac{1}{p}(B-R)$.

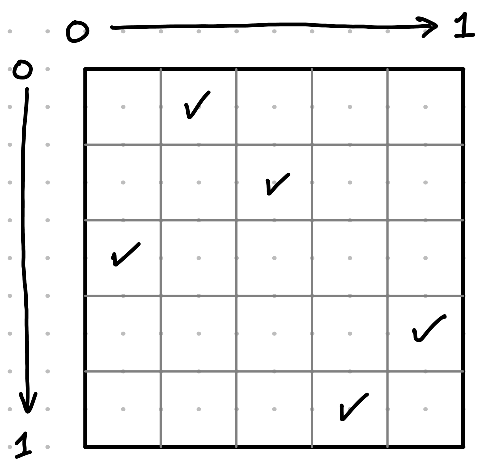
The ordered columns of A represent $\{x_k\}_{k=1}^{p} \subseteq (0,1)$.

This algorithm partitions $[0,1]^n$ into a $p \times p \times p \times \cdots \times p$ hyper Rubik's cube and guarantees that each column contains one point, each row contains one point, each hyperrow contains one point. Then in each small subcube, a random point is chosen.

Example : ($n=2$, $p=5$) Five points on unit square

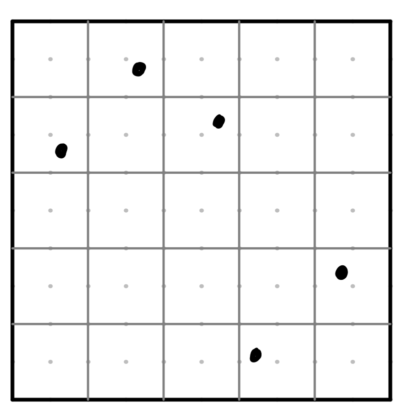
$$B = \begin{bmatrix} 2 & 3 & 5 & 1 & 4 \\ 3 & 1 & 4 & 2 & 5 \end{bmatrix}$$

represents :

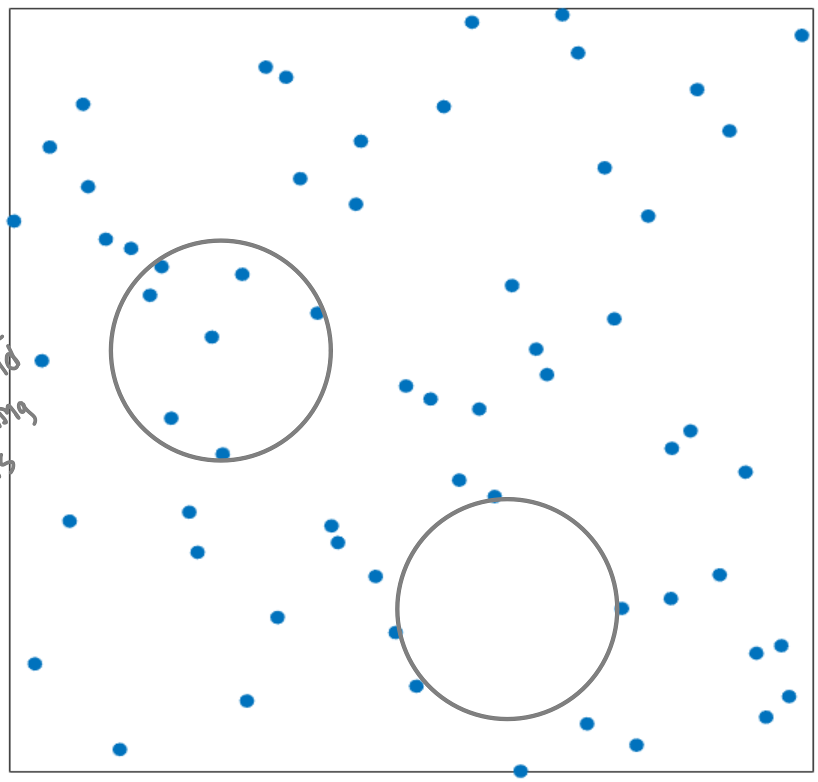


The columns of $A = \frac{B-R}{p}$ are then the points :

(for some random values R)



- repeatable through seeds
- may not be platform independent
- better uniformity enforced by B
- Samples individual coordinates uniformly.
- any coordinate hyperplane projection exhibits the same uniformity.



same size spheroid but containing 7 points

largest empty spheroid

62 points on $[0,1]^2$

- distributed across domain
- better visual uniformity with more diffuse clusters
- smaller largest empty spheroid

Halton Sequence

```
q=primes(10*n);
q=q(1:n);          ← first n prime numbers
x=zeros(n,p);      ← A matrix initialization
for k=1:n
    b=q(k);
    idx=1:p;
    f=ones(1,p);
    r=zeros(1,p);
    while any(idx>0)
        f=f/b;
        r=r+f.*mod(idx,b);
        idx=floor(idx/b);
    end
    x(k,:)=r;      ← A matrix of columns x_k
end
```

A Halton sequence is defined on $(0,1)$ with integer seed $q \geq 2$.

The Halton vector sequence shown here is composed of n Halton sequences each with a distinct prime number seed.

Halton sequence for seed $q=2$:

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \ldots$$

Halton sequence for seed $q=3$:

$$\frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \frac{7}{9}, \frac{2}{9}, \frac{5}{9}, \frac{8}{9}, \frac{1}{27}, \ldots$$

$\Rightarrow$ Halton Vector Sequence for $n=2$ and seeds $2, 3$:

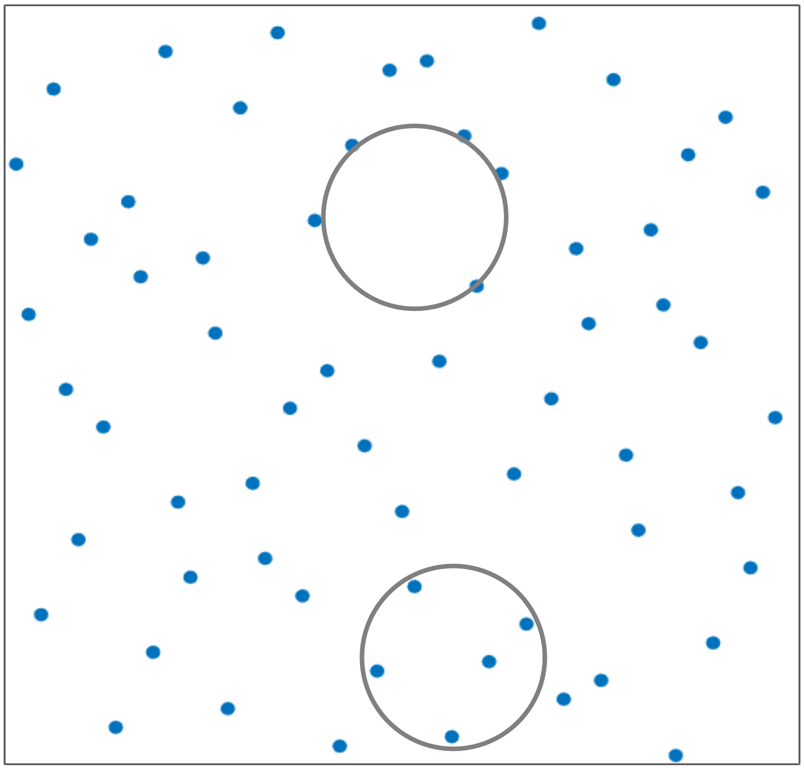
$$\begin{pmatrix} 1/2 \\ 1/3 \end{pmatrix}, \begin{pmatrix} 1/4 \\ 2/3 \end{pmatrix}, \begin{pmatrix} 3/4 \\ 1/9 \end{pmatrix}, \begin{pmatrix} 1/8 \\ 4/9 \end{pmatrix}, \begin{pmatrix} 5/8 \\ 7/9 \end{pmatrix}, \ldots$$

Interesting method for finding a sequence with seed q (example: $q=2$)

index k	k base q	reflect bits	decimal
1	1	.1	1/2
2	10	.01	1/4
3	11	.11	3/4
4	100	.001	1/8
5	101	.101	5/8
6	110	.011	3/8
7	111	.111	7/8

it's the Halton sequence!

- no random number generation
- platform independent
- maintains good uniformity at all r.



62 points on $[0,1]^2$

- distributed across domain
- excellent visual uniformity
- exhibits correlation structure if $q_1 \approx q_2 \gg 1$.