Recent Developments on Array-RQMC

Experiments with sorting strategies

Pierre L'Ecuyer, Christian Lécot, Bruno Tuffin

DIRO, Université de Montréal, Canada
LAMA, Université de Savoie, France
Inria–Rennes, France

IMACS MCM 2015, Linz

Monte Carlo for Markov Chains

Setting: A Markov chain with state space $\mathcal{X} \subseteq \mathbb{R}^\ell$, evolves as

$$X_0 = x_0, \qquad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j),\ j \geq 1,$$

where the $\mathbf{U}_j$ are i.i.d. uniform r.v.'s over $(0,1)^d$. Want to estimate

$$\mu = \mathbb{E}[Y] \quad \text{where} \quad Y = \sum_{j=1}^{\tau} g_j(X_j)$$

for some fixed time horizon τ.

Monte Carlo for Markov Chains

Setting: A Markov chain with state space $\mathcal{X} \subseteq \mathbb{R}^\ell$, evolves as

$$X_0 = x_0, \qquad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j),\ j \geq 1,$$

where the $\mathbf{U}_j$ are i.i.d. uniform r.v.'s over $(0,1)^d$. Want to estimate

$$\mu = \mathbb{E}[Y] \quad \text{where} \quad Y = \sum_{j=1}^{\tau} g_j(X_j)$$

for some fixed time horizon τ.

Ordinary MC: For $i = 0, \ldots, n-1$, generate $X_{i,j} = \varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})$, $j = 1, \ldots, \tau$, where the $\mathbf{U}_{i,j}$'s are i.i.d. $\mathrm{U}(0,1)^d$. Estimate μ by

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{\tau} g_j(X_{i,j}) = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

$\mathbb{E}[\hat{\mu}_n] = \mu$ and $\mathrm{Var}[\hat{\mu}_n] = \frac{1}{n}\mathrm{Var}[Y_i] = \mathcal{O}(n^{-1})$.

Example 1 (very simple, one-dimensional state)

Let $Y = \theta U + (1-\theta)V$, where U, V indep. $U(0,1)$ and $\theta \in [0,1)$.
This Y has cdf G_θ.

Markov chain is defined by

$$\begin{aligned} X_0 &= U_0 \sim U(0,1); \\ X_j &= \varphi_j(X_{j-1}, U_j) = G_\theta(\theta X_{j-1} + (1-\theta)U_j), \quad j \geq 1 \end{aligned}$$

where $U_j \sim U(0,1)$. Then, $X_j \sim U(0,1)$.

We consider $g_j(X_j) = X_j - 1/2$ and $g_j(X_j) = X_j^2 - 1/3$. $\mathbb{E}[g_j(X_j)] = 0$.

Example 2: Asian Call Option (two-dim state)

Given observation times $t_1, t_2, \ldots, t_\tau$ suppose

$$S(t_j) = S(t_{j-1}) \exp[(r - \sigma^2/2)(t_j - t_{j-1}) + \sigma(t_j - t_{j-1})^{1/2} \Phi^{-1}(U_j)],$$

where $U_j \sim U[0,1)$ and $S(t_0) = s_0$ is fixed.

Running average: $\bar{S}_j = \frac{1}{j} \sum_{i=1}^{j} S(t_i)$.
Payoff at step $j = \tau$ is $Y = g_\tau(X_\tau) = \max\left[0,\ \bar{S}_\tau - K\right]$.

State: $X_j = (S(t_j), \bar{S}_j)$.
Transition:

$$X_j = (S(t_j), \bar{S}_j) = \varphi_j(S(t_{j-1}), \bar{S}_{j-1}, U_j) = \left(S(t_j), \frac{(j-1)\bar{S}_{j-1} + S(t_j)}{j}\right).$$

Plenty of applications:

Finance

Queueing systems

Inventory, distribution, logistic systems

Reliability models

MCMC in Bayesian statistics

Etc.

Classical Randomized Quasi-Monte Carlo (RQMC) for Markov Chains

One RQMC point for each sample path.

Put $\mathbf{V}_i = (\mathbf{U}_{i,1}, \ldots, \mathbf{U}_{i,\tau}) \in (0,1)^s = (0,1)^{d\tau}$. Estimate μ by

$$\hat{\mu}_{\mathrm{rqmc},n} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\tau} g_j(X_{i,j})$$

where $P_n = \{\mathbf{V}_0, \ldots, \mathbf{V}_{n-1}\} \subset (0,1)^s$ satisfies:
(a) each point $\mathbf{V}_i$ has the uniform distribution over $(0,1)^s$;
(b) P_n covers $(0,1)^s$ very evenly (i.e., has low discrepancy).

The dimension s is often very large!

Array-RQMC for Markov Chains

L., Lécot, Tuffin, et al. [2004, 2006, 2008, etc.]

Earlier deterministic versions: Lécot et al.

Simulate an "array" of n chains in "parallel."

At each step, use an RQMC point set P_n to advance all the chains by one step, with global negative dependence across the chains.

Goal: Want small discrepancy (or "distance") between empirical distribution of $S_{n,j} = \{X_{0,j}, \ldots, X_{n-1,j}\}$ and theoretical distribution of X_j.

If we succeed, these (unbiased) estimators will have small variance:

$$\mu_j = \mathbb{E}[g_j(X_j)] \approx \hat{\mu}_{\mathrm{arqmc},j,n} = \frac{1}{n}\sum_{i=0}^{n-1} g_j(X_{i,j})$$

$$\mathrm{Var}[\hat{\mu}_{\mathrm{arqmc},j,n}] = \frac{\mathrm{Var}[g_j(X_{i,j})]}{n} + \frac{2}{n^2}\sum_{i=0}^{n-1}\sum_{k=i+1}^{n-1} \mathrm{Cov}[g_j(X_{i,j}),\, g_j(X_{k,j})]\,.$$

Some generalizations

L., Lécot, and Tuffin [2008]: τ can be a random stopping time w.r.t. the filtration $\mathcal{F}\{(j, X_j), j \geq 0\}$.

L., Demers, and Tuffin [2006, 2007]: Combination with splitting techniques (multilevel and without levels), combination with importance sampling and weight windows. Covers particle filters.

L. and Sanvido [2010]: Combination with coupling from the past for exact sampling.

Dion and L. [2010]: Combination with approximate dynamic programming and for optimal stopping problems.

Gerber and Chopin [2015]: Sequential QMC.

Convergence results and applications

L., Lécot, and Tuffin [2006, 2008]: Special cases: convergence at MC rate, one-dimensional, stratification, etc. Var in $\mathcal{O}(n^{-3/2})$.

Lécot and Tuffin [2004]: Deterministic, one-dimension, discrete state.

El Haddad, Lécot, L. [2008, 2010]: Deterministic, multidimensional.

Fakhererredine, El Haddad, Lécot [2012, 2013, 2014]: LHS, stratification, Sudoku sampling, ...

Wächter and Keller [2008]: Applications in computer graphics.

Gerber and Chopin [2015]: Sequential QMC (particle filters). var in $o(n^{-1})$.

Some RQMC insight: To simplify the discussion, suppose $X_j \sim U(0,1)^\ell$. This can be achieved (in principle) by a change of variable. We estimate

$$\mu_j = \mathbb{E}[g_j(X_j)] = \mathbb{E}[g_j(\varphi_j(X_{j-1}, \mathbf{U}))] = \int_{[0,1)^{\ell+d}} g_j(\varphi_j(\mathbf{x}, \mathbf{u}))d\mathbf{x}d\mathbf{u}$$

(we take a single j here) by

$$\hat{\mu}_{\text{arqmc},j,n} = \frac{1}{n}\sum_{i=0}^{n-1} g_j(X_{i,j}) = \frac{1}{n}\sum_{i=0}^{n-1} g_j(\varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})).$$

This is (roughly) RQMC with the point set $Q_n = \{(X_{i,j-1}, \mathbf{U}_{i,j}),\ 0 \le i < n\}$.

We want Q_n to have low discrepancy (LD) (be highly uniform) over $[0,1)^{\ell+d}$.

Some RQMC insight: To simplify the discussion, suppose $X_j \sim U(0,1)^\ell$. This can be achieved (in principle) by a change of variable. We estimate

$$\mu_j = \mathbb{E}[g_j(X_j)] = \mathbb{E}[g_j(\varphi_j(X_{j-1}, \mathbf{U}))] = \int_{[0,1)^{\ell+d}} g_j(\varphi_j(\mathbf{x}, \mathbf{u}))d\mathbf{x}d\mathbf{u}$$

(we take a single j here) by

$$\hat{\mu}_{\text{arqmc},j,n} = \frac{1}{n}\sum_{i=0}^{n-1} g_j(X_{i,j}) = \frac{1}{n}\sum_{i=0}^{n-1} g_j(\varphi_j(X_{i,j-1}, \mathbf{U}_{i,j})).$$

This is (roughly) RQMC with the point set $Q_n = \{(X_{i,j-1}, \mathbf{U}_{i,j}),\ 0 \le i < n\}$.

We want Q_n to have low discrepancy (LD) (be highly uniform) over $[0,1)^{\ell+d}$.

We do not choose the $X_{i,j-1}$'s in Q_n: they come from the simulation.
To construct the (randomized) $\mathbf{U}_{i,j}$, select a LD point set

$$\tilde{Q}_n = \{(\mathbf{w}_0, \mathbf{U}_{0,j}), \ldots, (\mathbf{w}_{n-1}, \mathbf{U}_{n-1,j})\},$$

where the $\mathbf{w}_i \in [0,1)^\ell$ are fixed and each $\mathbf{U}_{i,j} \sim U(0,1)^d$.
Permute the states $X_{i,j-1}$ so that $X_{\pi_j(i),j-1}$ is "close" to $\mathbf{w}_i$ for each i (LD between the two sets), and compute $X_{i,j} = \varphi_j(X_{\pi_j(i),j-1}, \mathbf{U}_{i,j})$ for each i.

Example: If $\ell = 1$, can take $\mathbf{w}_i = (i + 0.5)/n$ and just sort the states.
For $\ell > 1$, there are various ways to define the matching (multivariate sort).

Key issues:

1. How can we preserve LD of $S_{n,j}$ as j increases?

2. Can we prove that $\mathrm{Var}[\hat{\mu}_{\mathrm{arqmc},j,n}] = \mathcal{O}(n^{-\alpha})$ for some $\alpha > 1$?
How? What α?

Intuition: Write discrepancy measure of $S_{n,j}$ as the mean square integration error (or variance) when integrating some function $\psi : [0,1)^{\ell+d} \to \mathbb{R}$ using Q_n.
Use RQMC theory to show it is small if Q_n has LD. Then use induction.

Examples of convergence-rate results:

For $\ell = d = 1$, with stratified sampling, and smooth φ_j and g_j, we proved a variance bound with $\alpha = 3/2$.

For general ℓ, Gerber and Chopin (2015) have a proof that $\mathrm{Var}[\hat{\mu}_{\mathrm{arqmc},j,n}] = o(n^{-1})$ when sorting the states with a Hilbert curve, under certain conditions.

Array-RQMC algorithm

$X_{i,0} \leftarrow x_0$ (or $X_{i,0} \leftarrow x_{i,0}$) for $i = 0, \ldots, n-1$;
for $j = 1, 2, \ldots, \tau$ **do**
  Compute the permutation π_j of the states (for matching);
  Randomize afresh $\{\mathbf{U}_{0,j}, \ldots, \mathbf{U}_{n-1,j}\}$ in $\tilde{Q}_n$;
  $X_{i,j} = \varphi_j(X_{\pi_j(i),j-1}, \mathbf{U}_{i,j})$, for $i = 0, \ldots, n-1$;
end for
Estimate μ by the average $\bar{Y}_n = \hat{\mu}_{\mathrm{arqmc},n}$.

Array-RQMC algorithm

$X_{i,0} \leftarrow x_0$ (or $X_{i,0} \leftarrow x_{i,0}$) for $i = 0, \ldots, n-1$;
for $j = 1, 2, \ldots, \tau$ **do**
  Compute the permutation π_j of the states (for matching);
  Randomize afresh $\{\mathbf{U}_{0,j}, \ldots, \mathbf{U}_{n-1,j}\}$ in $\tilde{Q}_n$;
  $X_{i,j} = \varphi_j(X_{\pi_j(i),j-1}, \mathbf{U}_{i,j})$, for $i = 0, \ldots, n-1$;
end for
Estimate μ by the average $\bar{Y}_n = \hat{\mu}_{\mathrm{arqmc},n}$.

Proposition: (i) The average $\bar{Y}_n$ is an unbiased estimator of μ.
(ii) The empirical variance of m independent realizations gives an unbiased estimator of $\mathrm{Var}[\bar{Y}_n]$.

The one-dimensional example

$$X_0 = U_0; \qquad X_j = \varphi_j(X_{j-1}, U_j) = G_\theta(\theta X_{j-1} + (1-\theta)U_j), \quad j \geq 1$$

For array-RQMC, we take $X_{i,0} = \mathbf{w}_i = (i - 1/2)/n$.

We tried different RQMC methods, for $n = 2^9$ to $n = 2^{21}$.
We did $m = 200$ independent replications for each n.
We fitted a linear regression of $\log_2 \mathrm{Var}[\bar{Y}_{n,j}]$ vs $\log_2 n$.
We also looked at $\mathbb{E}[D_j^2]$ for the Cramer von Mises statistic:

$$D_j^2 = \frac{1}{12n^2} + \frac{1}{n}\sum_{i=0}^{n-1}((i + 0.5/n) - X_{(i),j})^2.$$

Some MC and RQMC point sets:

MC:	Crude Monte Carlo
LHS:	Latin hypercube sampling
SS:	Stratified sampling
SSA:	Stratified sampling with antithetic variates in each stratum
Sobol2:	Sobol' points with linear matrix scrambling and digital random shift
Korobov:	Korobov lattice in 2 dim. with a random shift modulo 1

Slope ... vs $\log_2 n$:

RQMC points	$\log_2 \mathbb{E}[D_j^2]$	$\log_2 \mathrm{Var}[\bar{Y}_{n,j}]$		$\mathrm{Var}[\bar{Y}_{n,j}]$ for $n = 2^{21}$
		$X_j - 1/2$	$X_j^2 - 1/3$	
$j = 4$				
MC	-1.01	-1.02	-1.01	4.50×10^{-8}
LHS	-1.02	-1.05	-1.00	1.52×10^{-9}
SS	-1.50	-2.00	-2.00	2.52×10^{-14}
SSA	-1.50	-2.63	-2.60	6.73×10^{-19}
Sobol2	-1.51	-3.21	-3.16	4.22×10^{-21}
Korobov	-1.87	-2.00	-1.99	2.52×10^{-14}
$j = 20$				
MC	-1.00	-1.00	-0.99	
LHS	-1.00	-1.00	-1.00	
SS	-1.50	-2.00	-2.01	
SSA	-1.50	-2.63	-2.58	
Sobol2	-1.49	-3.17	-3.15	
Korobov	-1.88	-1.99	-1.98	

Mapping chains to points when $\ell > 2$

1. Multivariate batch sort:

Sort the states (chains) by first coordinate, in n_1 packets of size n/n_1.

Sort each packet by second coordinate, in n_2 packets of size $n/n_1 n_2$.

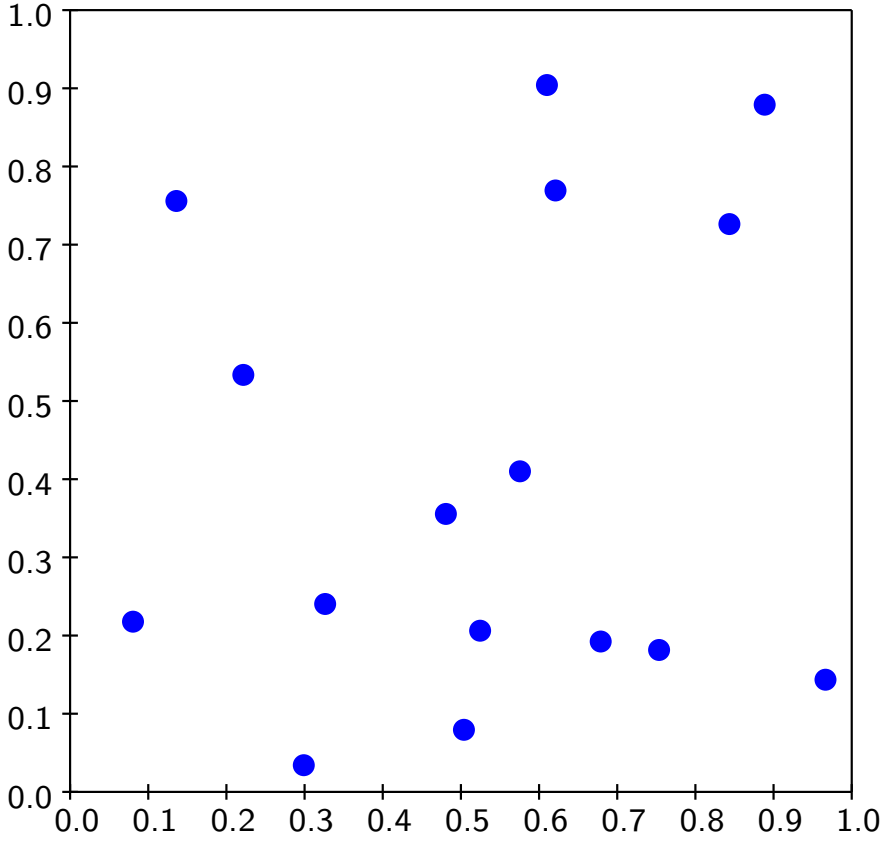
...

At the last level, sort each packet of size n_ℓ by the last coordinate.

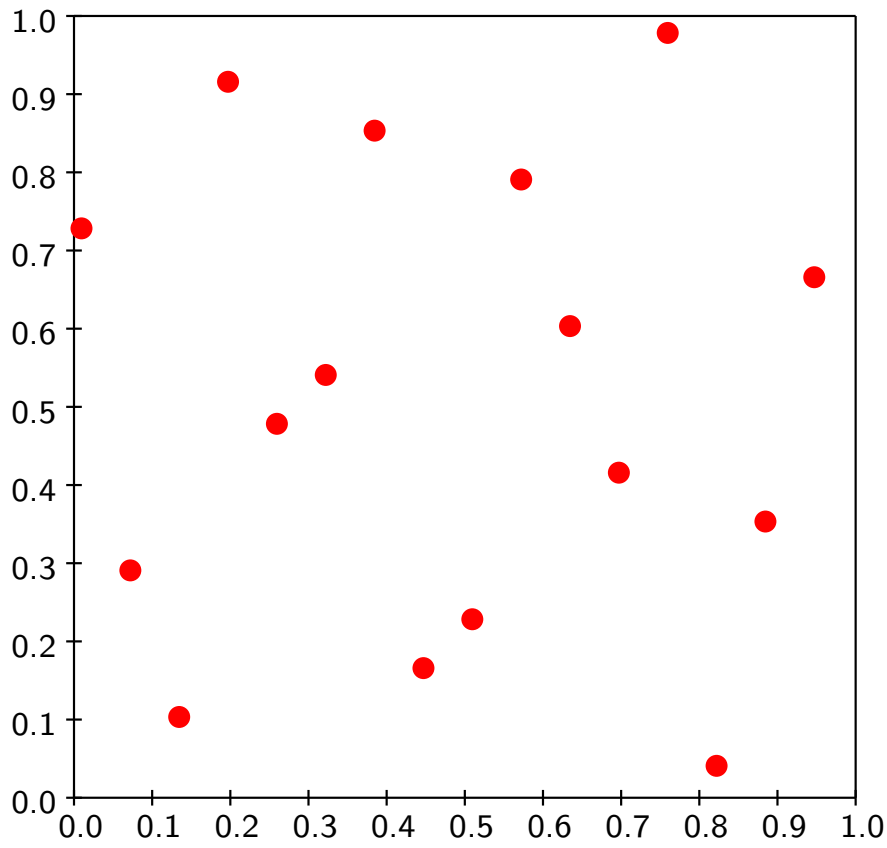
Choice of $n_1, n_2, ..., n_\ell$?

A (4,4) mapping

States of the chains

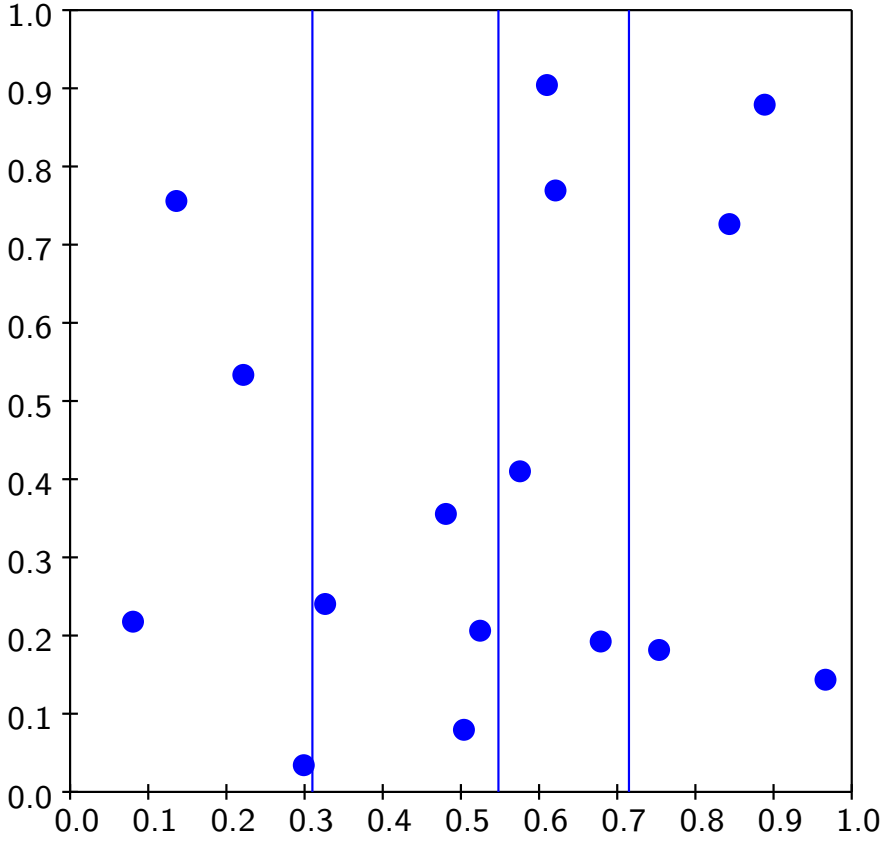


Sobol' net in 2 dimensions after random digital shift

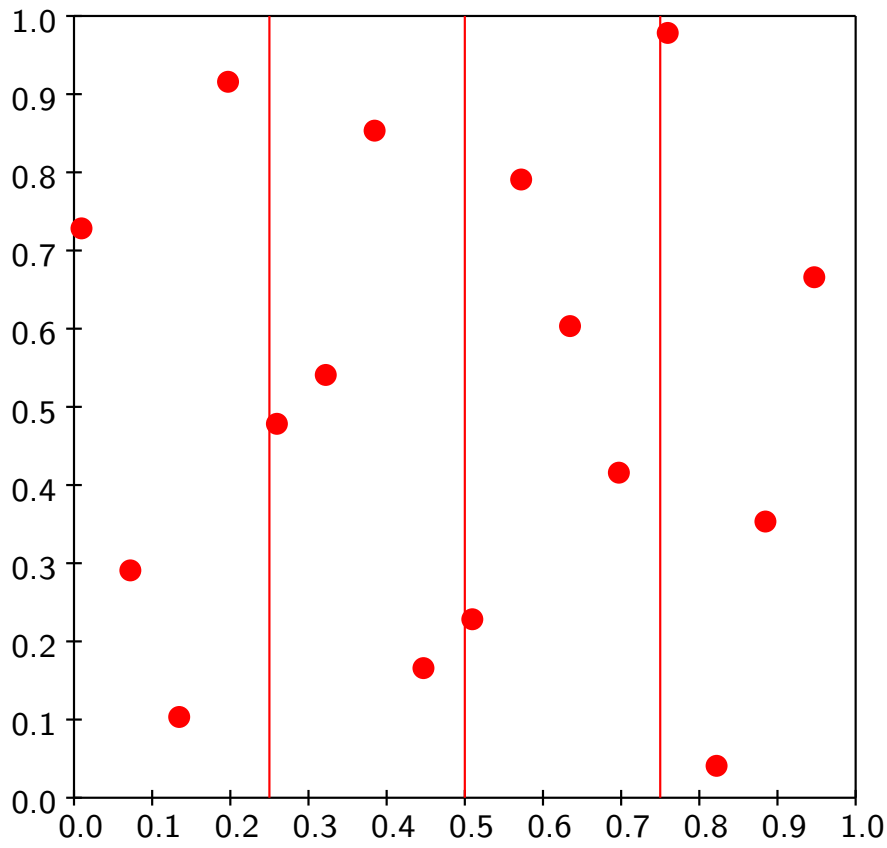


A (4,4) mapping

States of the chains

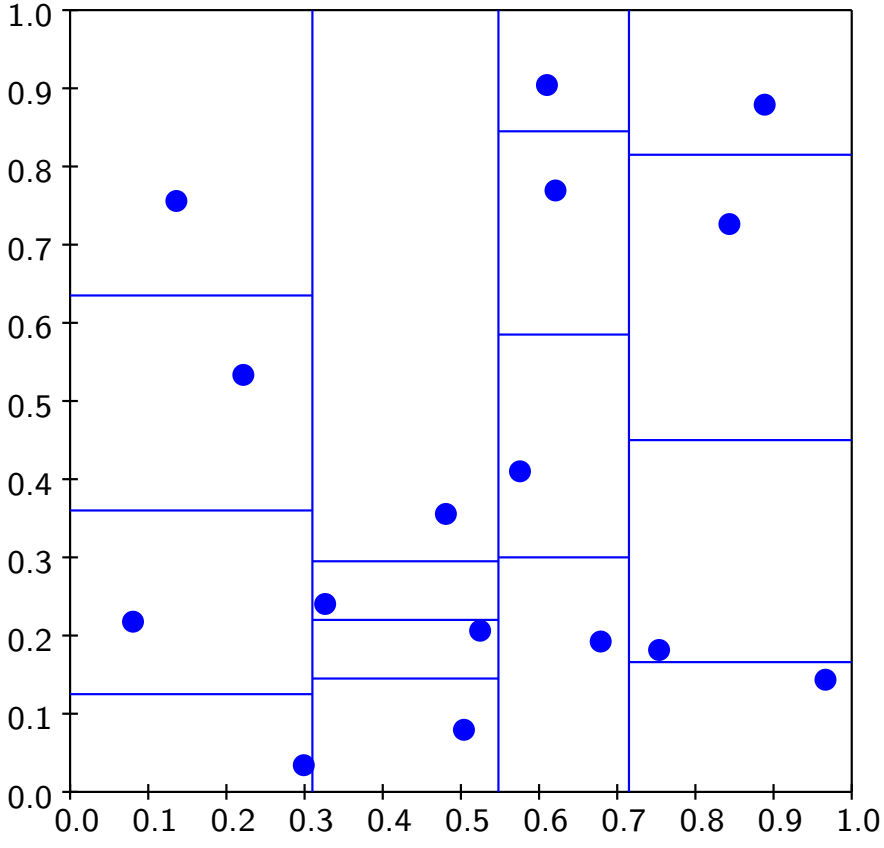


Sobol' net in 2 dimensions after random digital shift

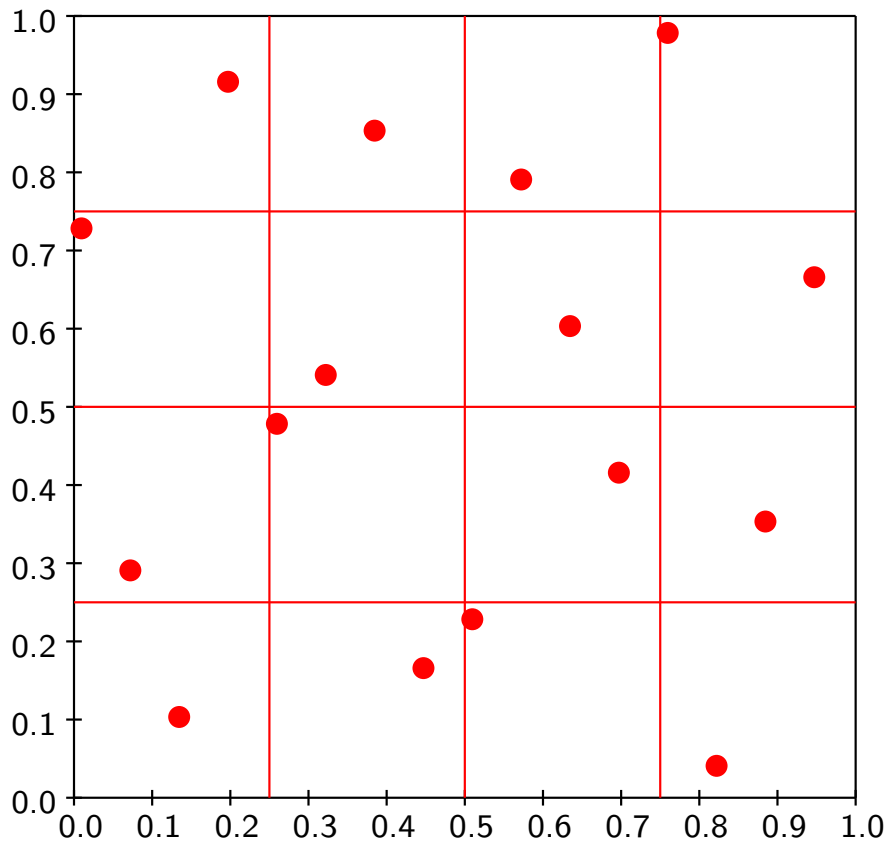


A (4,4) mapping

States of the chains

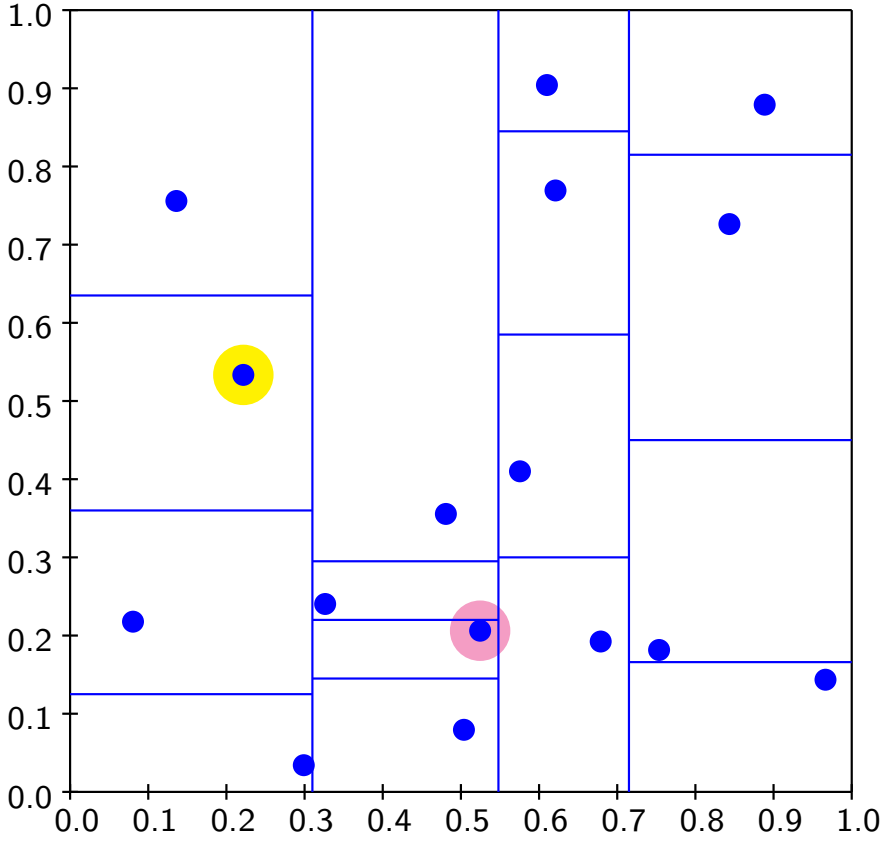


Sobol' net in 2 dimensions after random digital shift

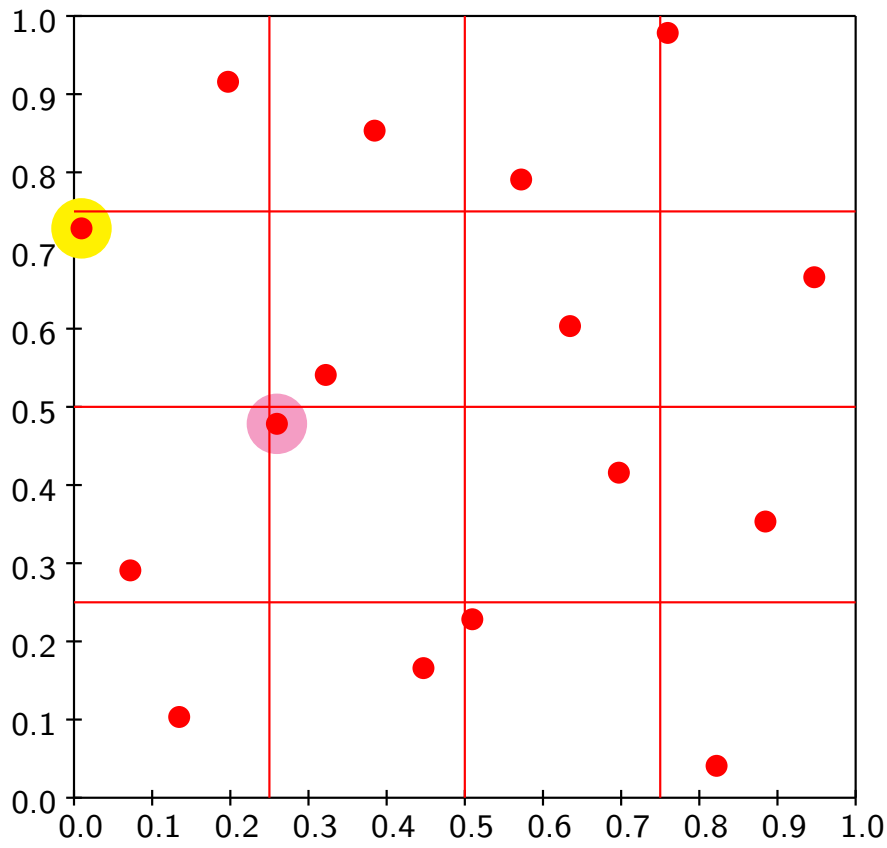


A (4,4) mapping

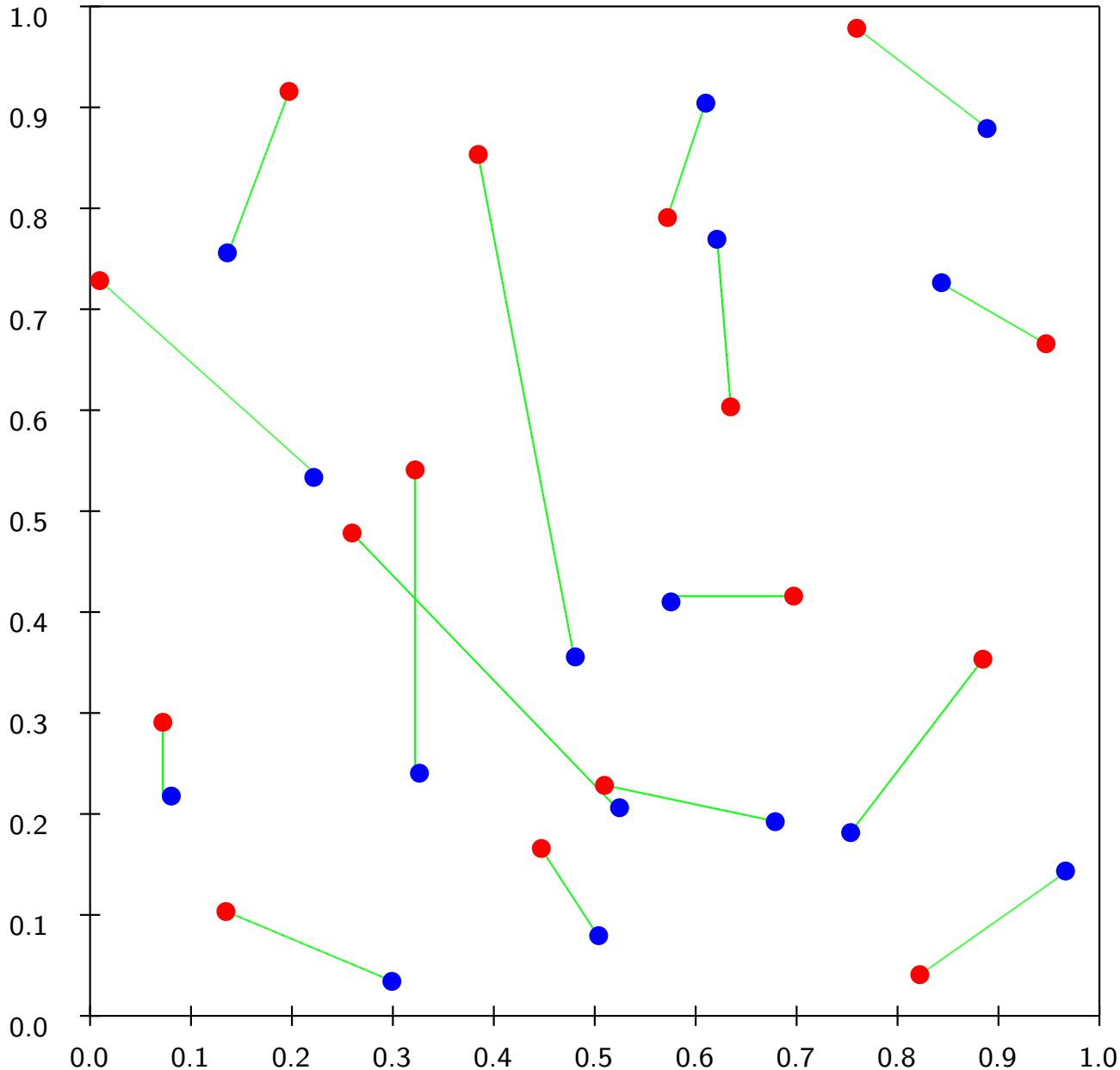
States of the chains



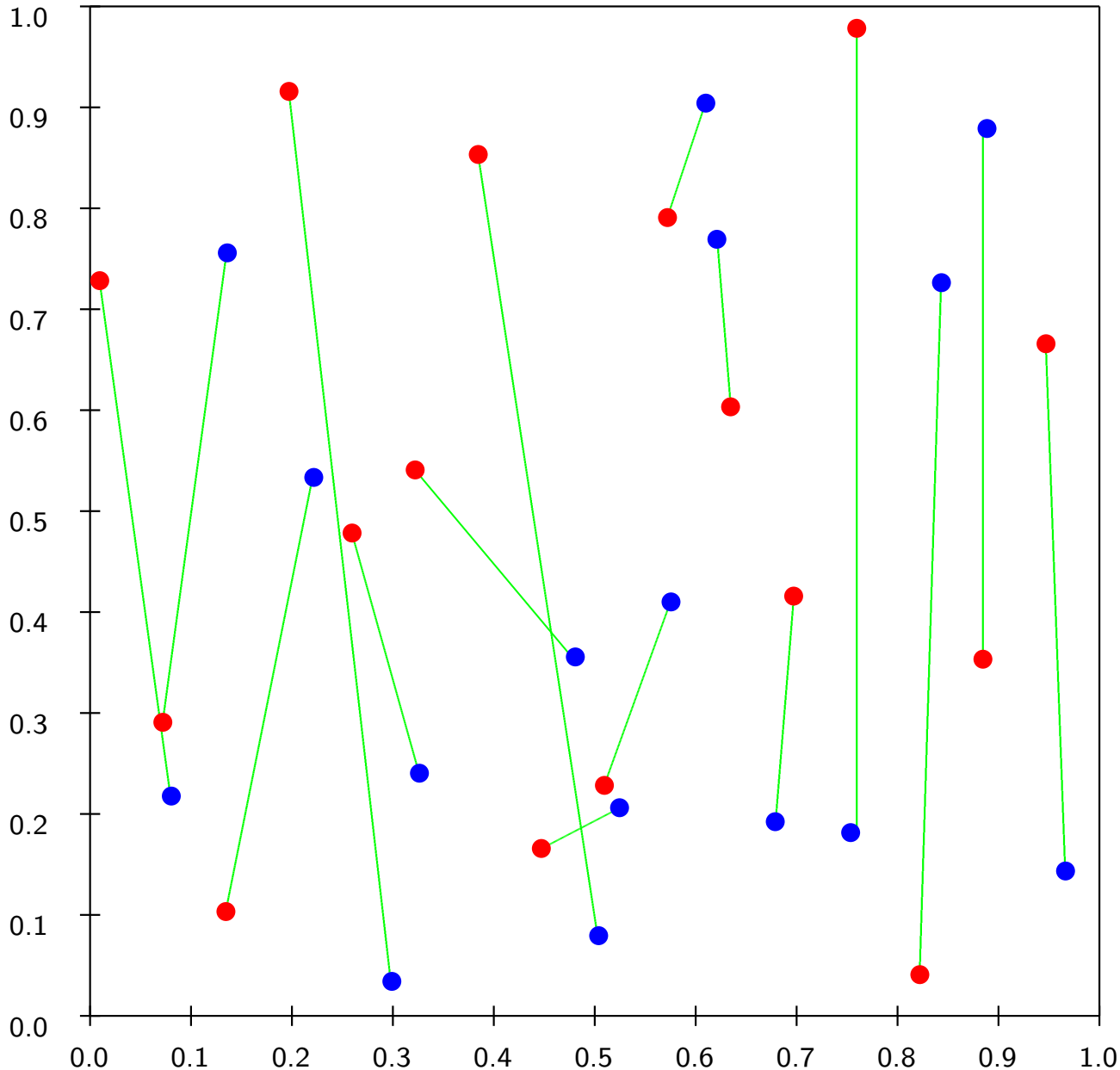
Sobol' net in 2 dimensions after random digital shift



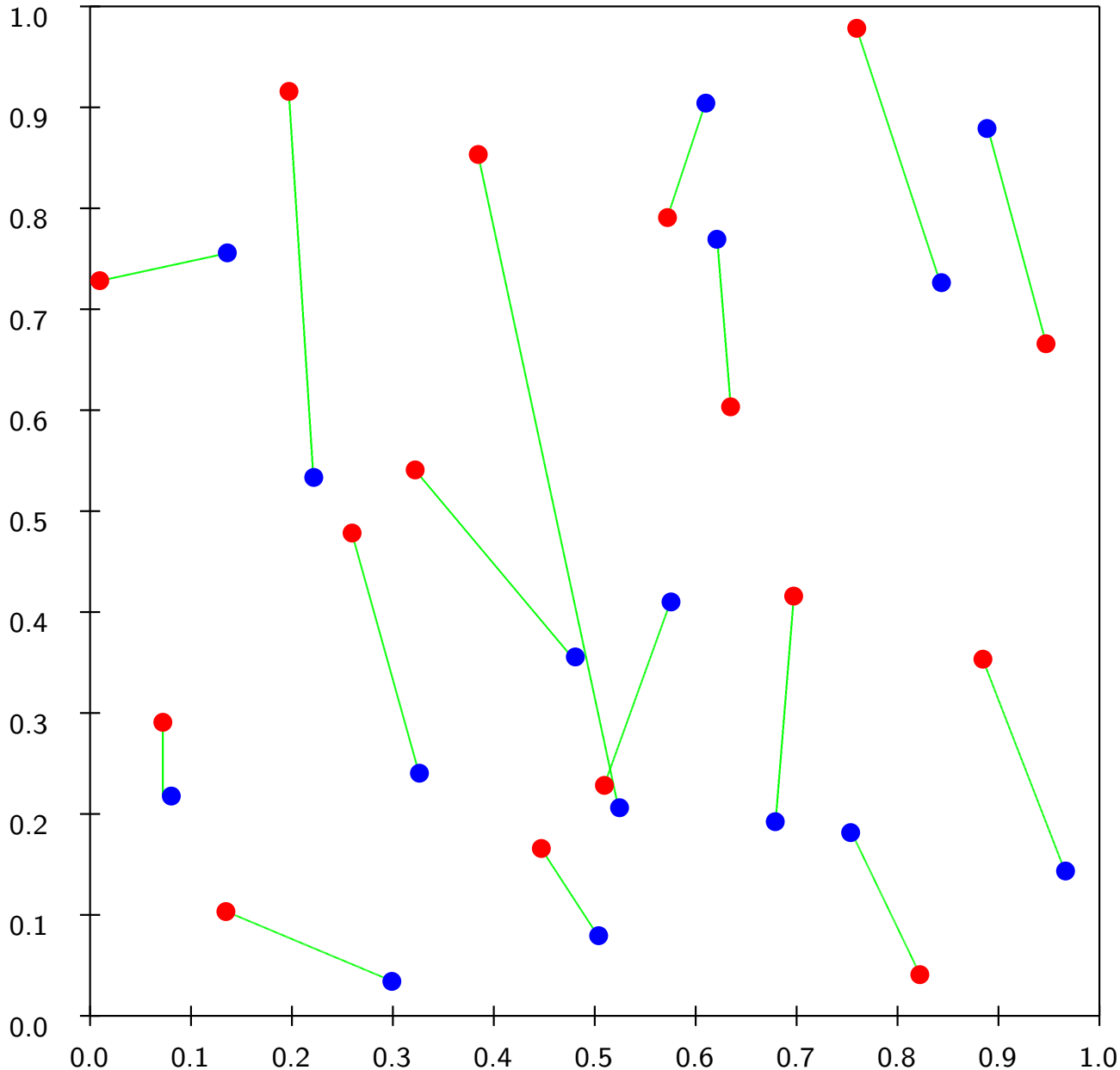
A (4,4) mapping



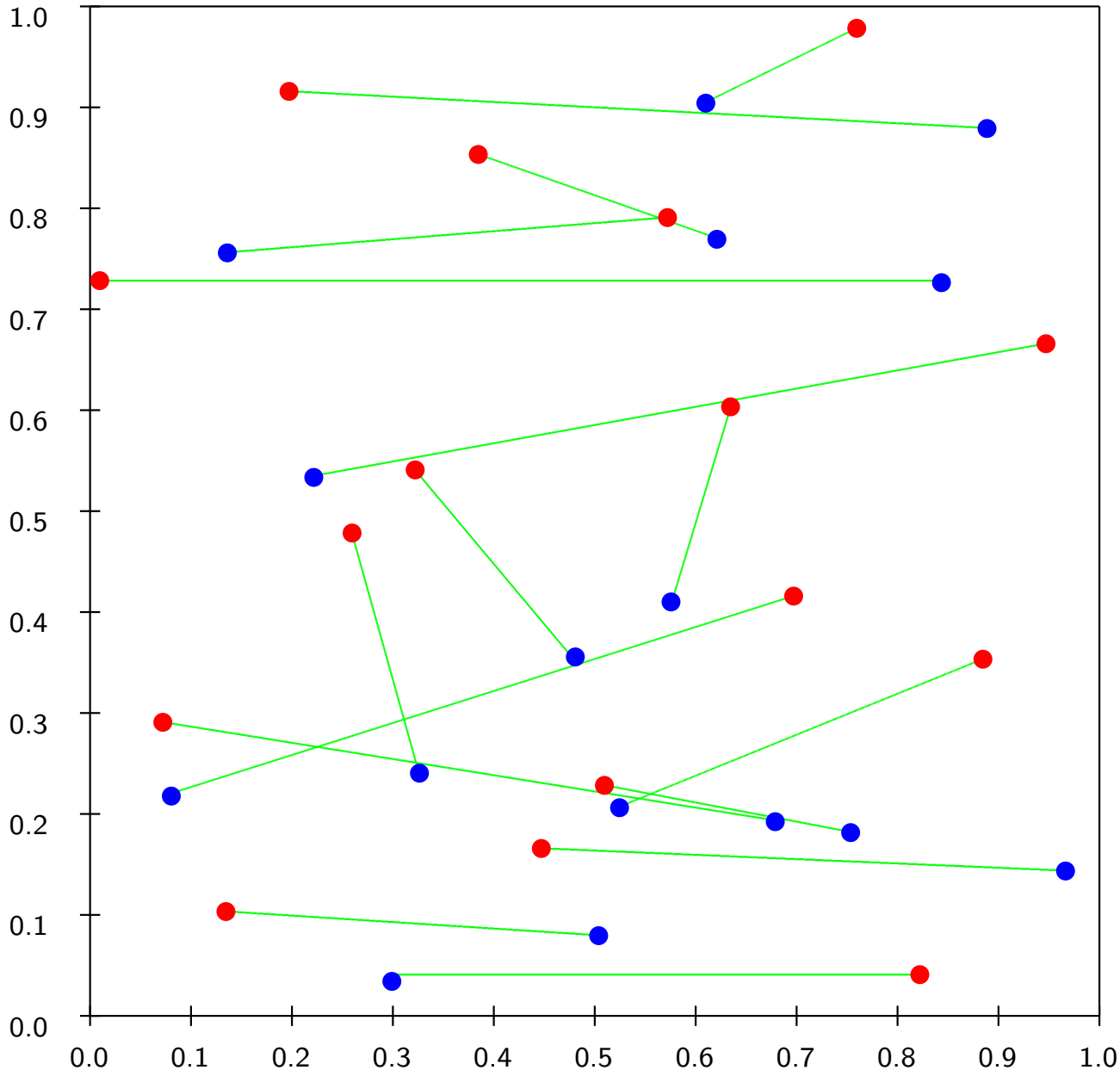
A (16,1) mapping, sorting along first coordinate



A (8,2) mapping



A (1,16) mapping, sorting along second coordinate



Mapping chains to points when $\ell > 2$

1. Multivariate split sort:

$n_1 = n_2 = \cdots = 2$.

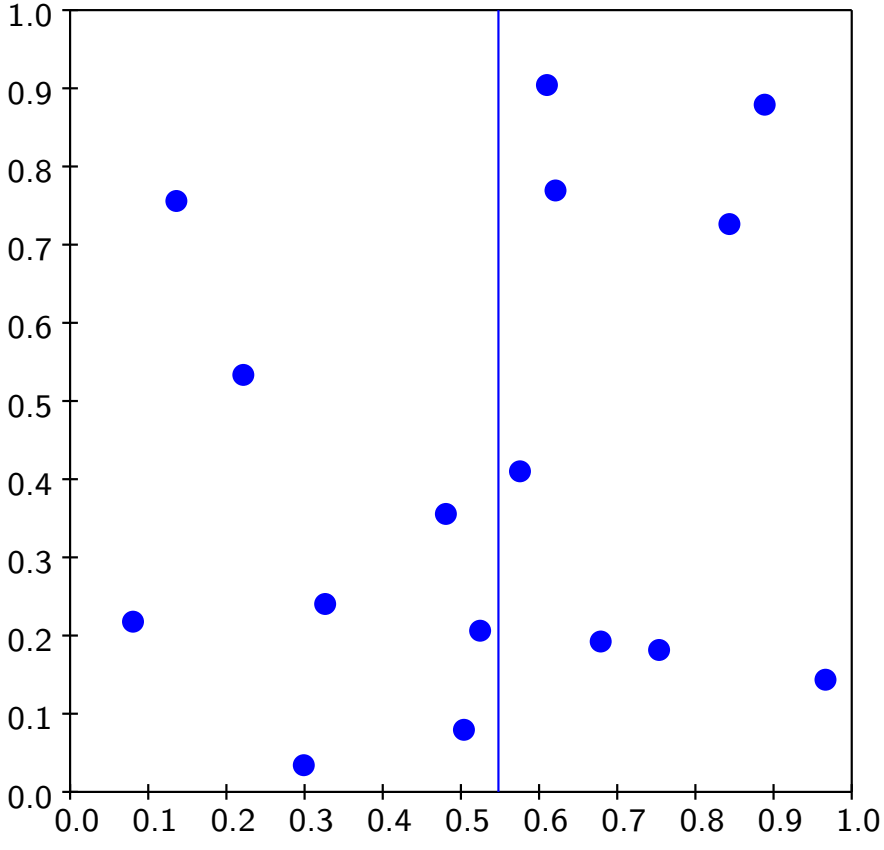
Sort by first coordinate in 2 packets.

Sort each packet by second coordinate in 2 packets.

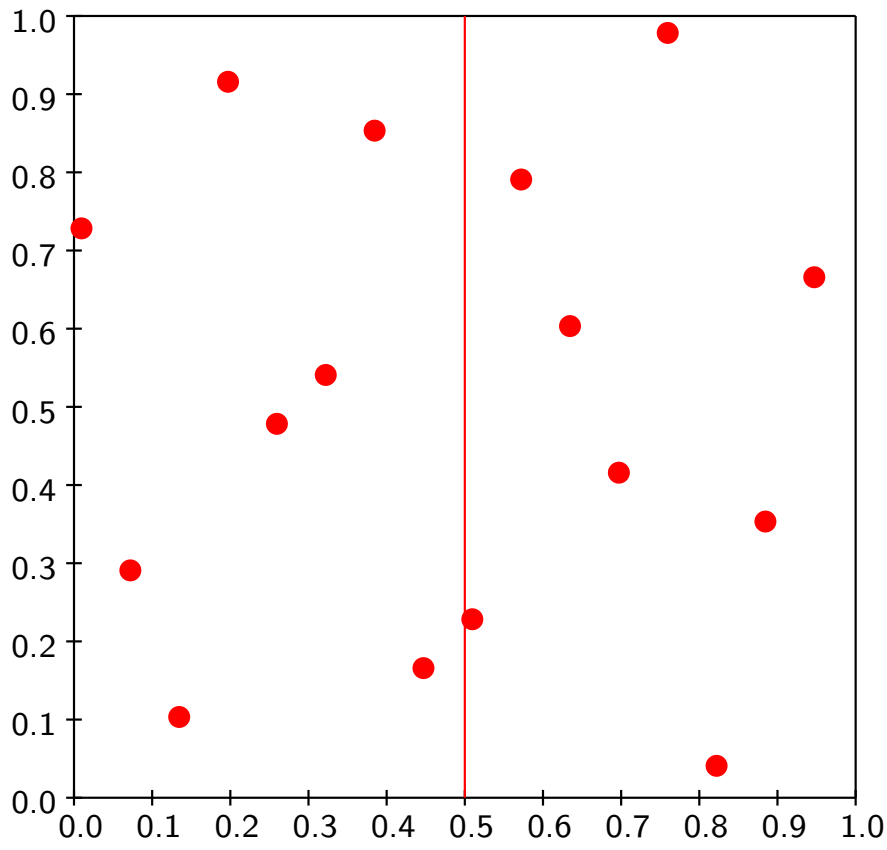
etc.

Mapping by split sort

States of the chains

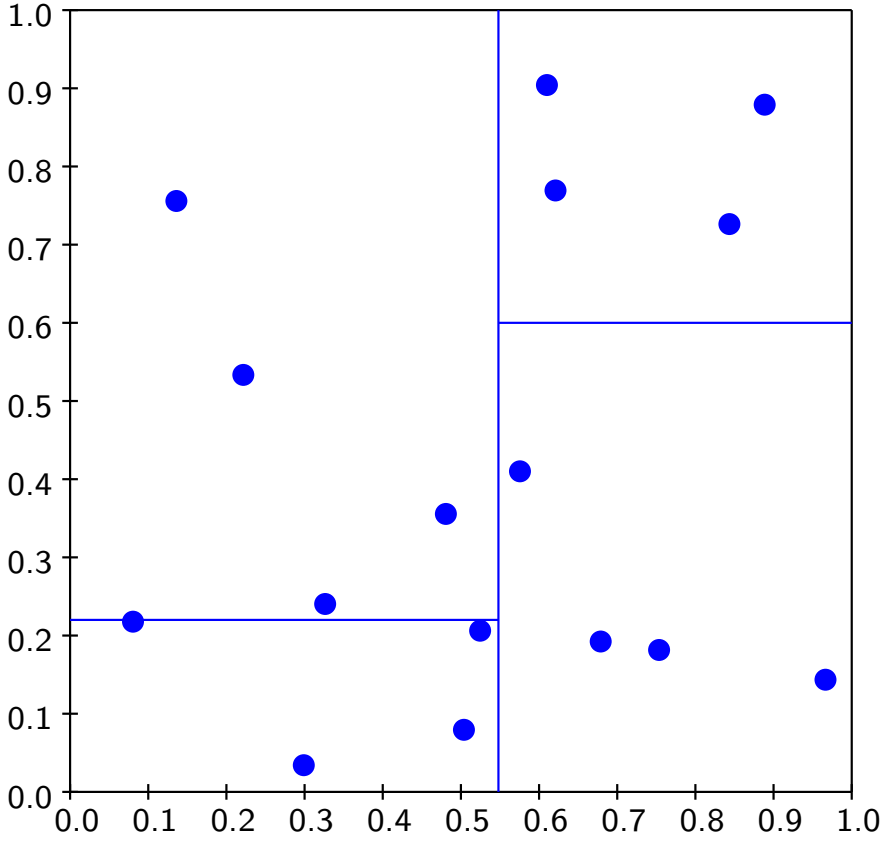


Sobol' net in 2 dimensions after random digital shift

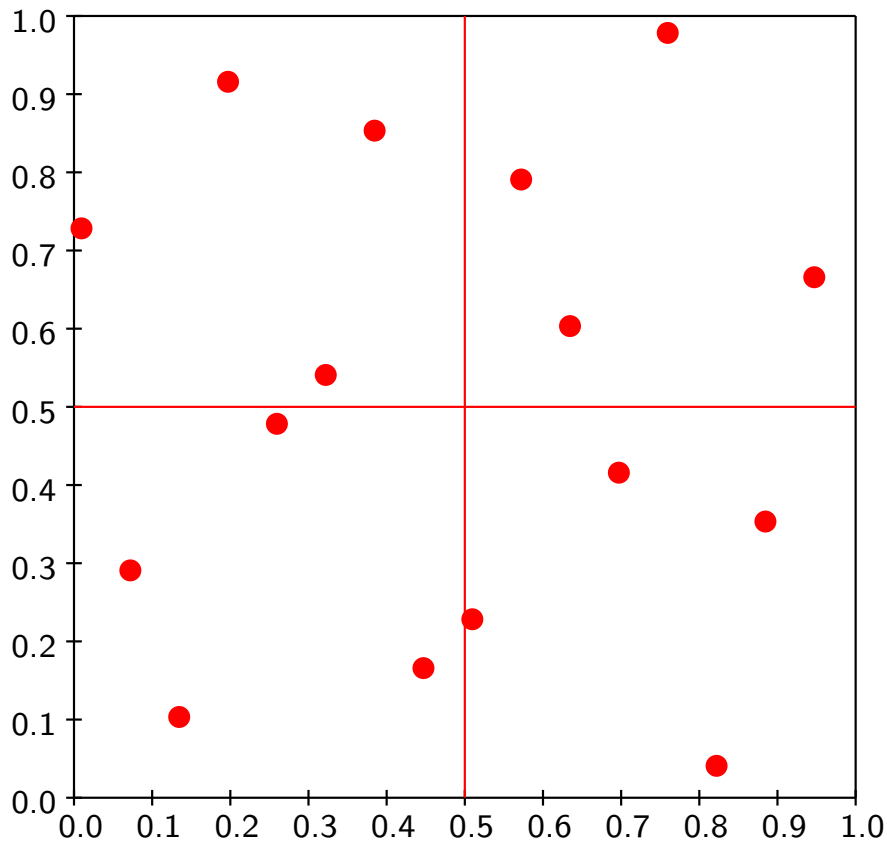


Mapping by split sort

States of the chains

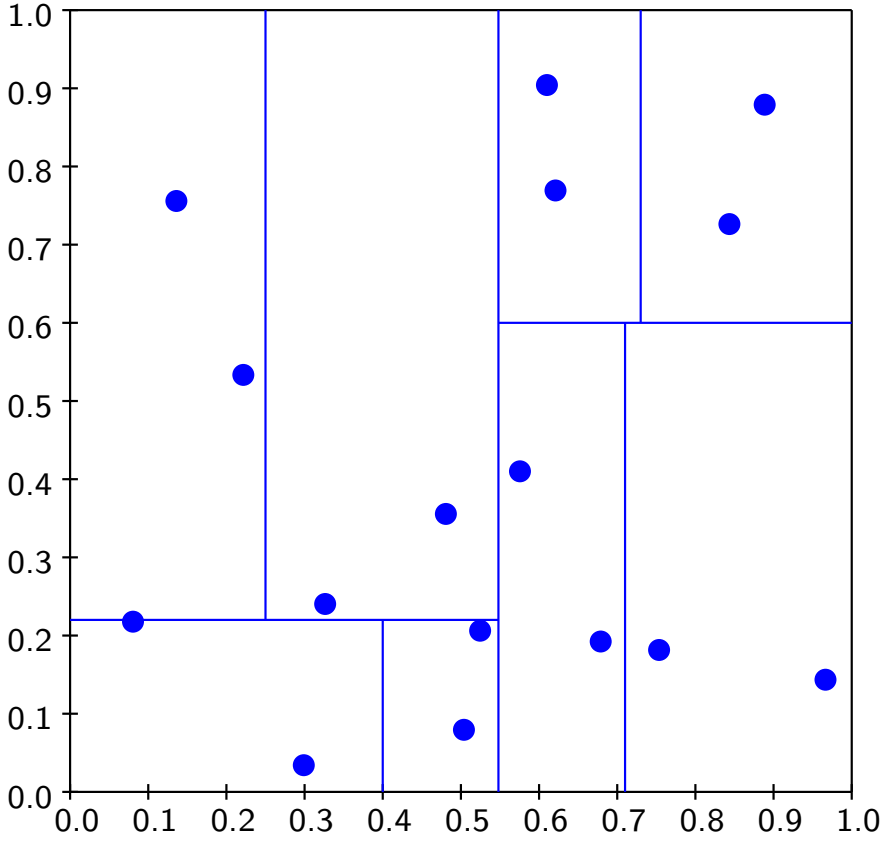


Sobol' net in 2 dimensions after random digital shift

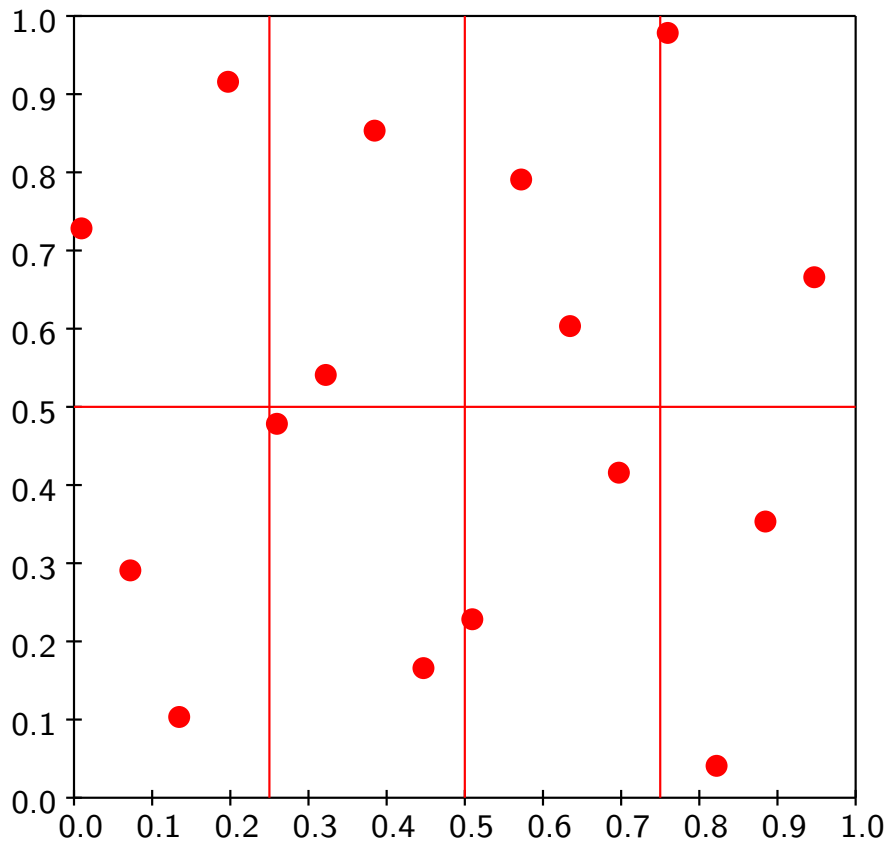


Mapping by split sort

States of the chains

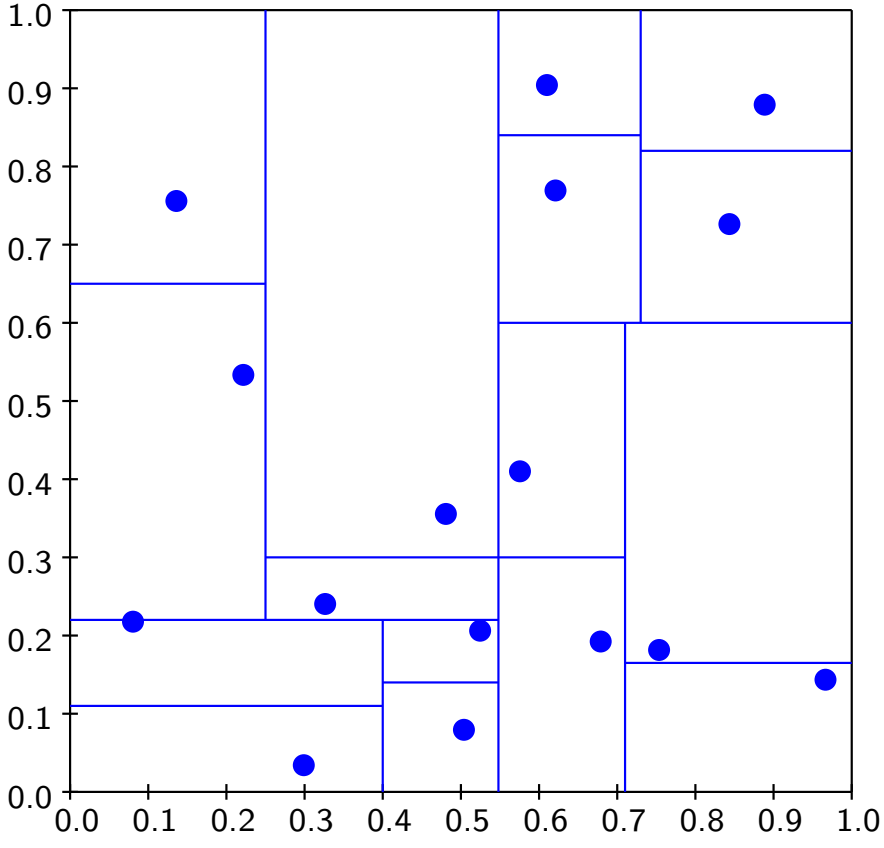


Sobol' net in 2 dimensions after random digital shift

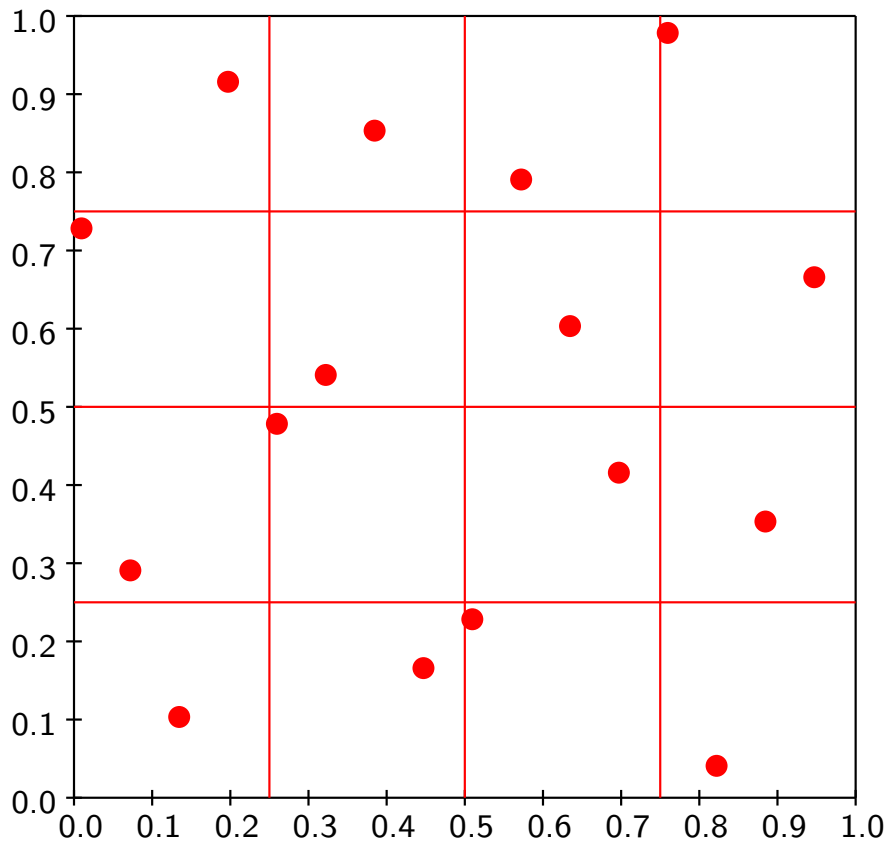


Mapping by split sort

States of the chains

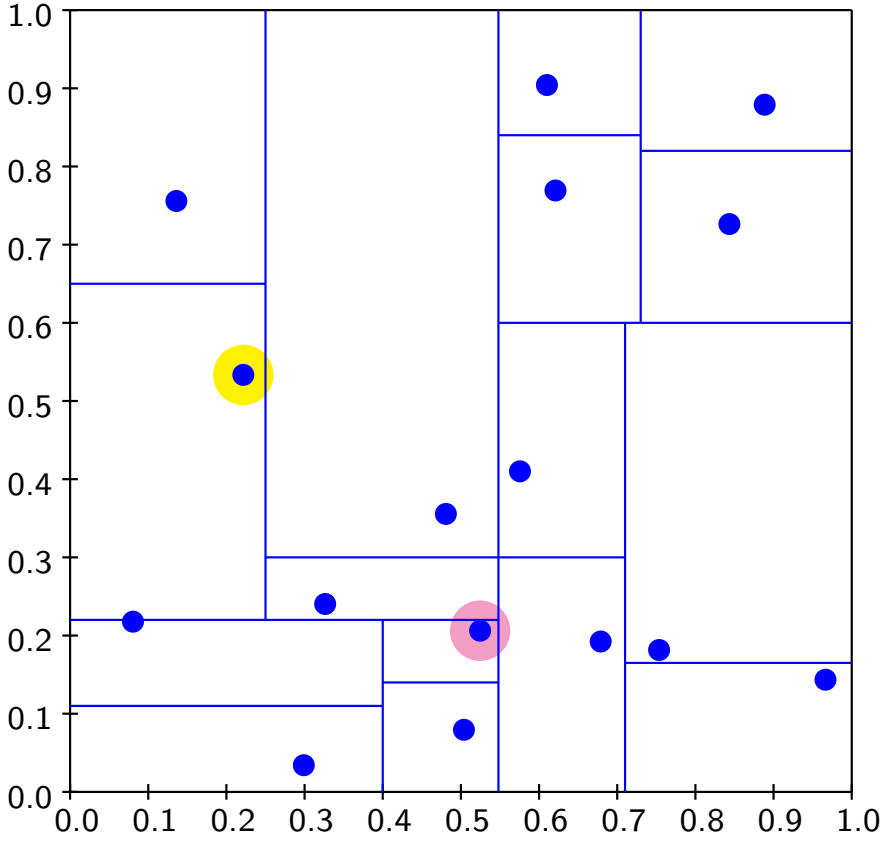


Sobol' net in 2 dimensions after random digital shift

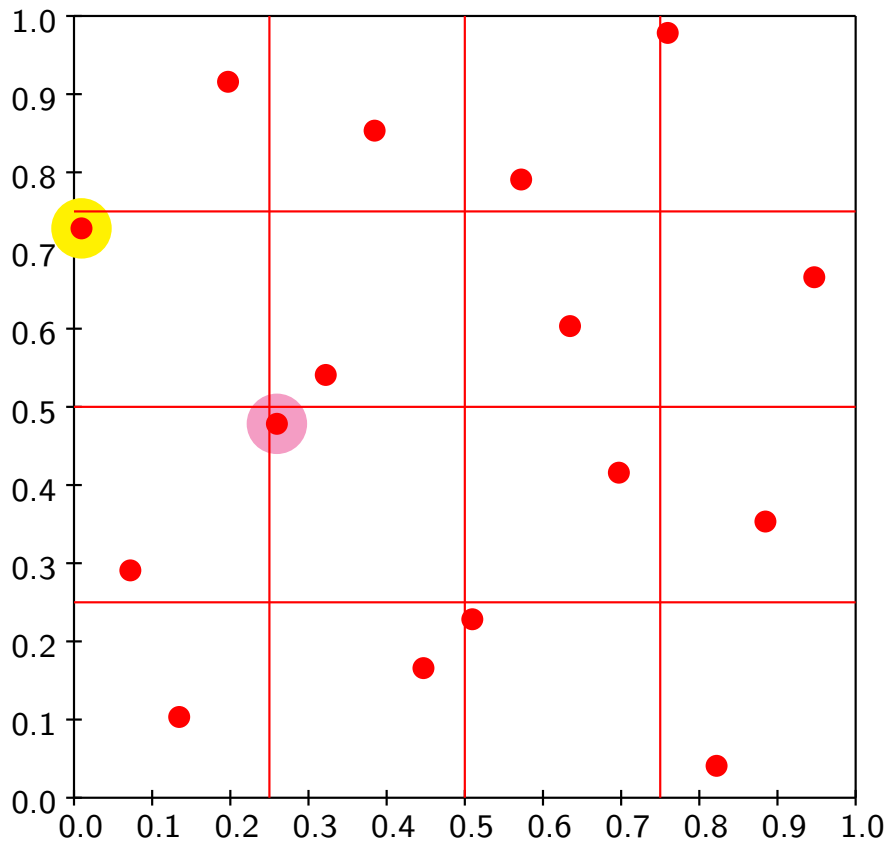


Mapping by split sort

States of the chains



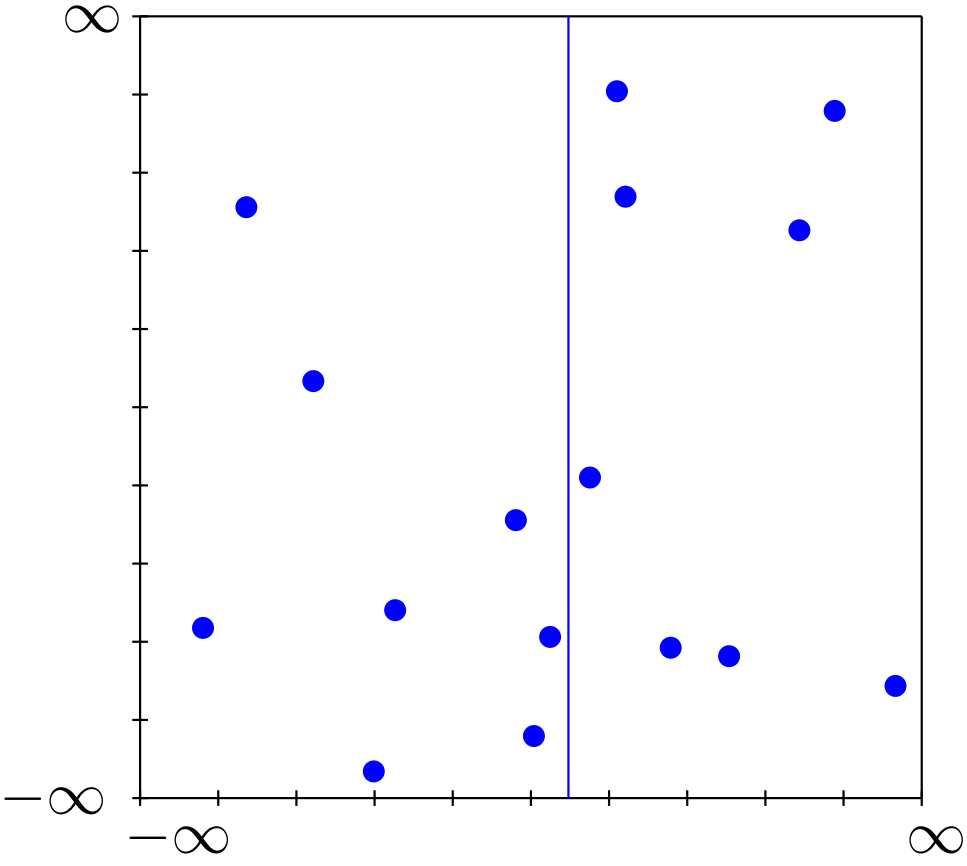
Sobol' net in 2 dimensions after random digital shift



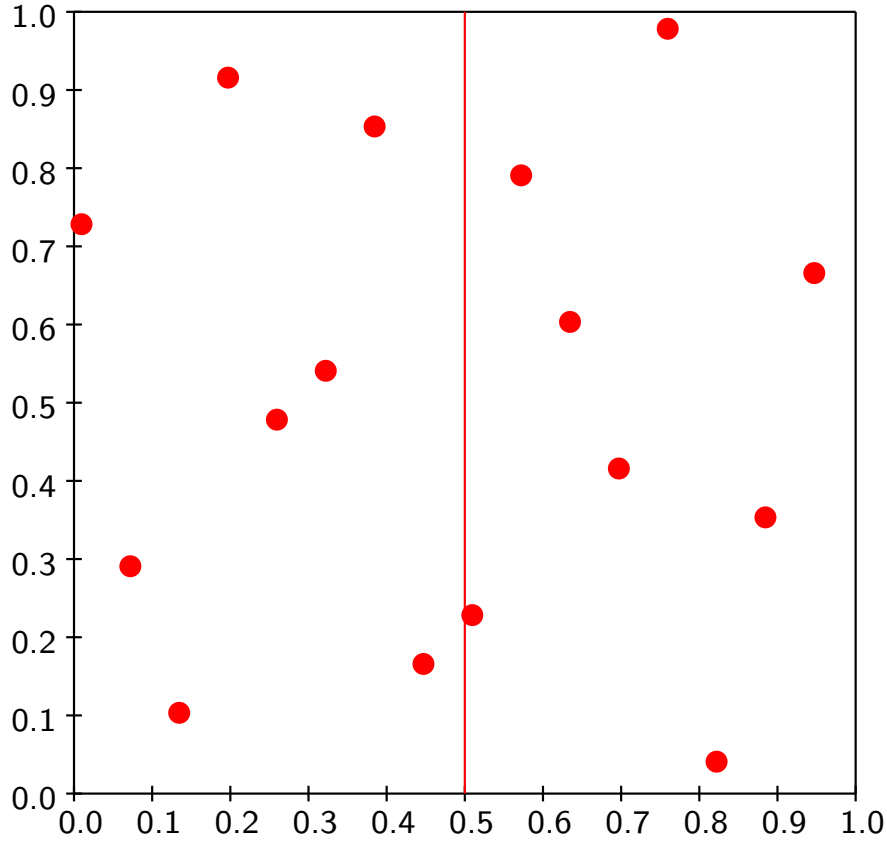
Mapping by batch sort and split sort

Good news: The state space does not have to be $[0,1)^d$:

States of the chains



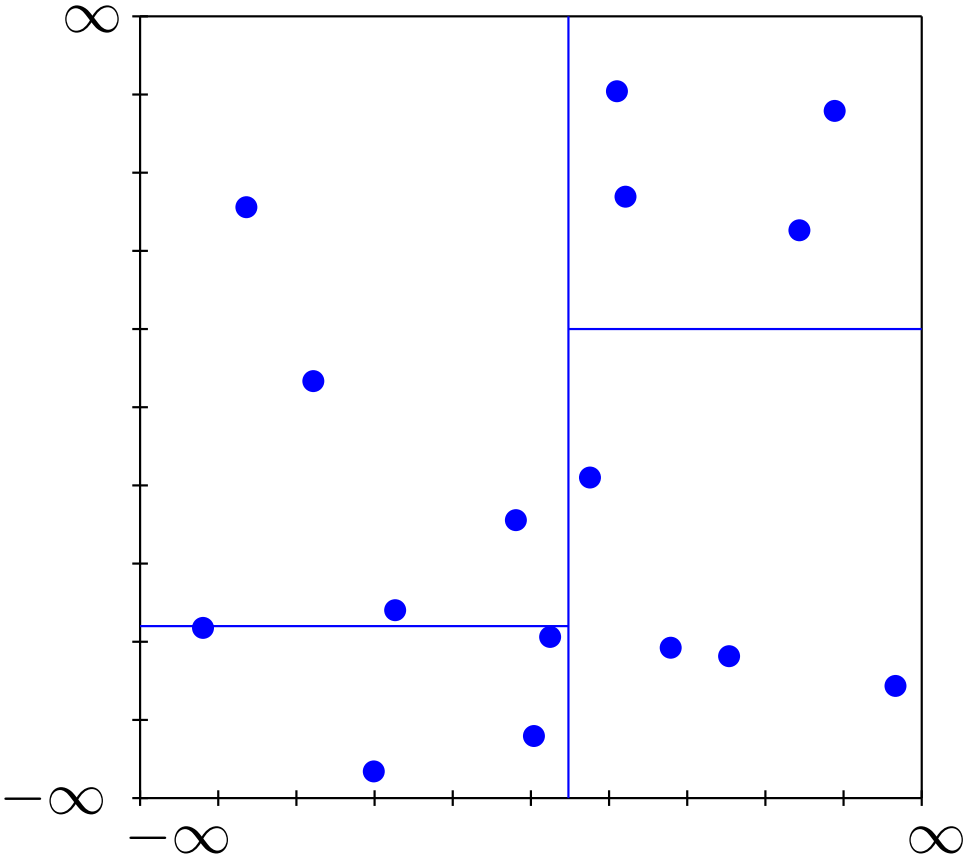
Sobol' net + digital shift



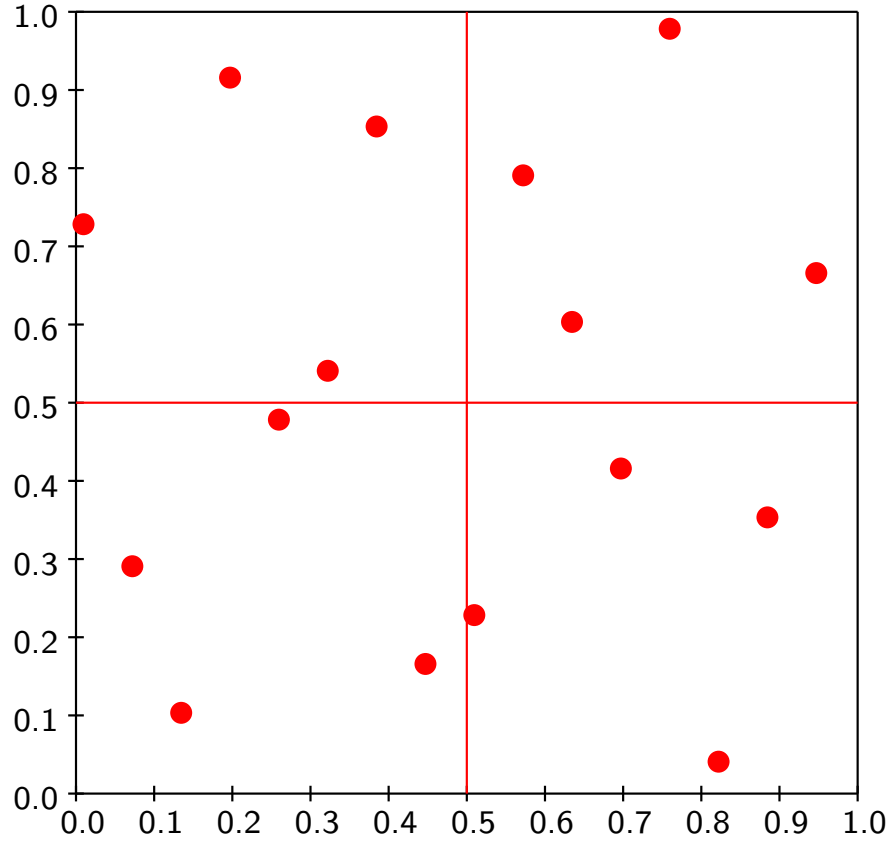
Mapping by batch sort and split sort

Good news: The state space does not have to be $[0,1)^d$:

States of the chains



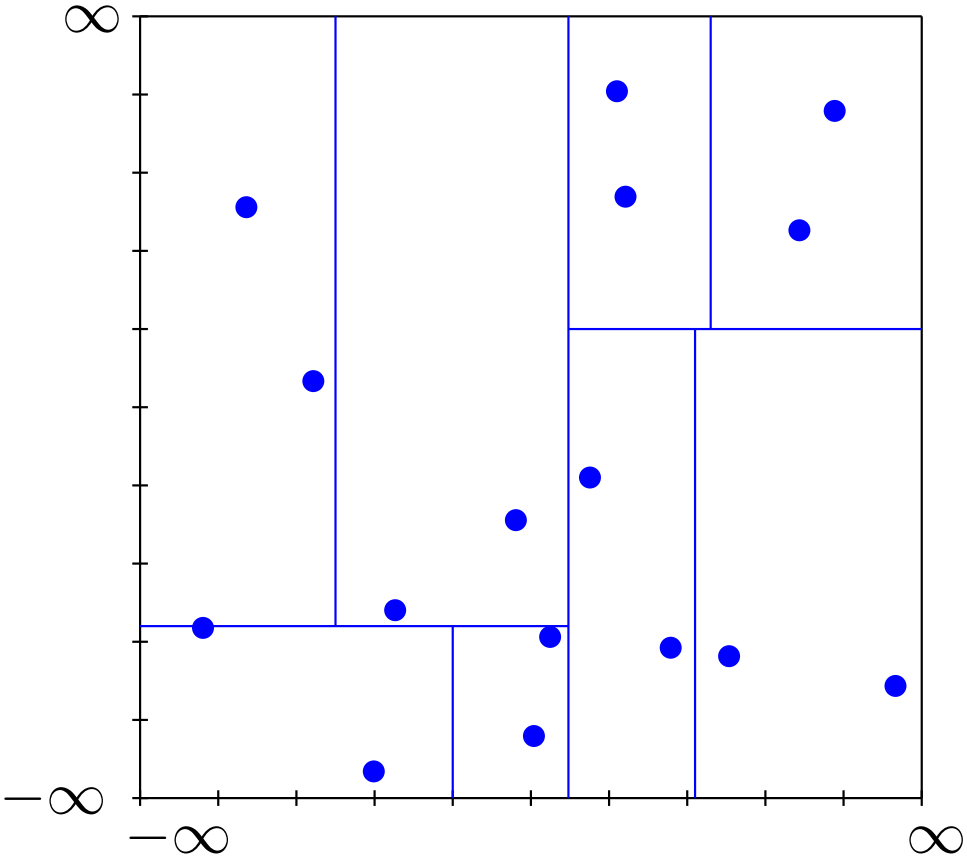
Sobol' net + digital shift



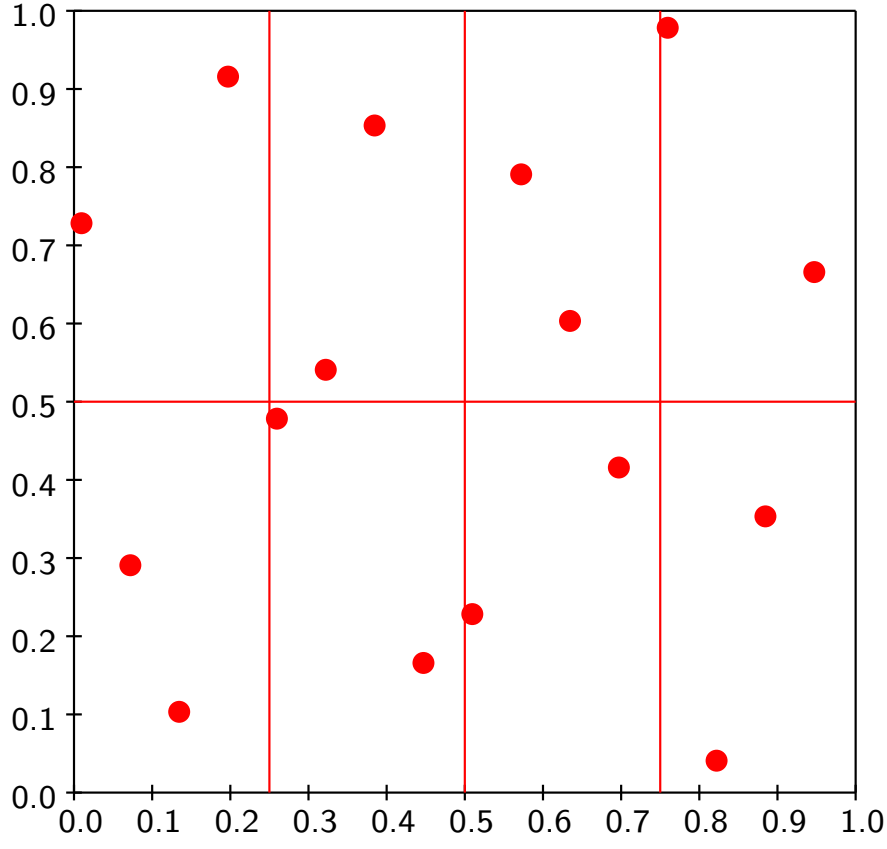
Mapping by batch sort and split sort

Good news: The state space does not have to be $[0,1)^d$:

States of the chains



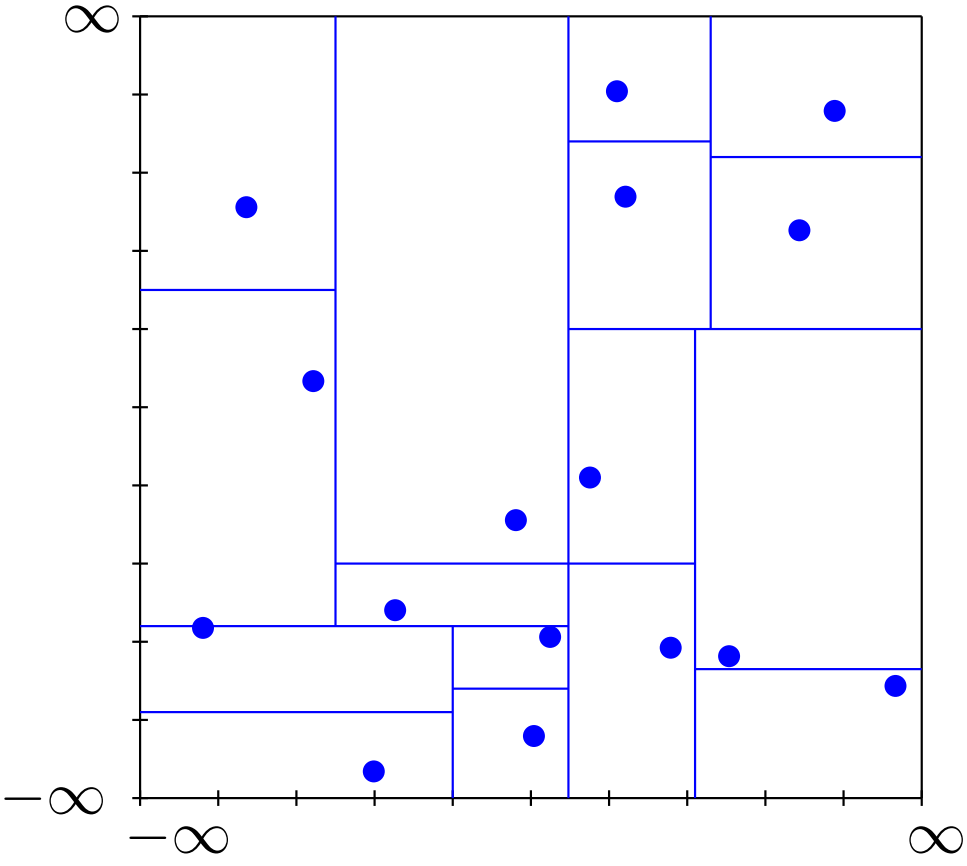
Sobol' net + digital shift



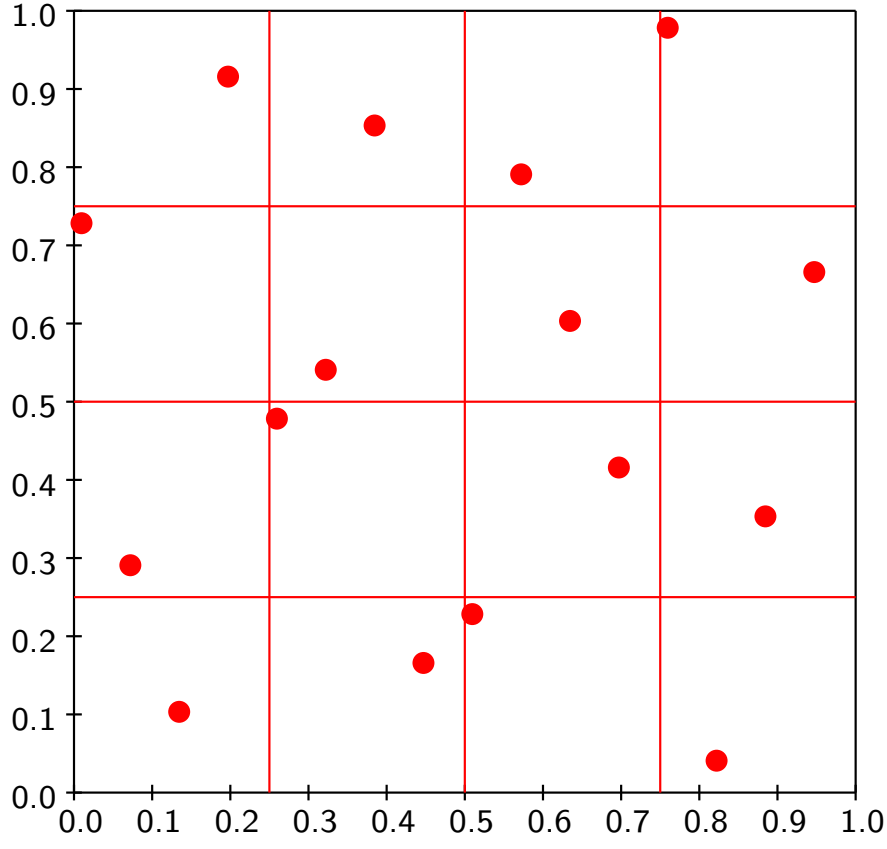
Mapping by batch sort and split sort

Good news: The state space does not have to be $[0,1)^d$:

States of the chains



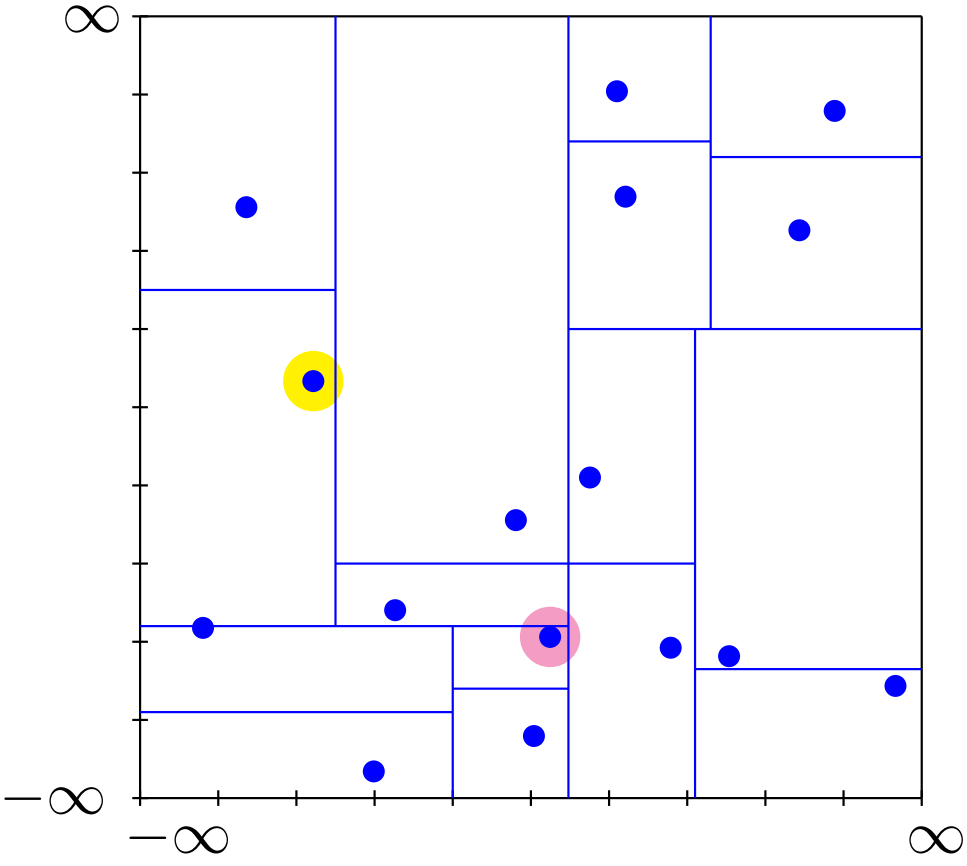
Sobol' net + digital shift



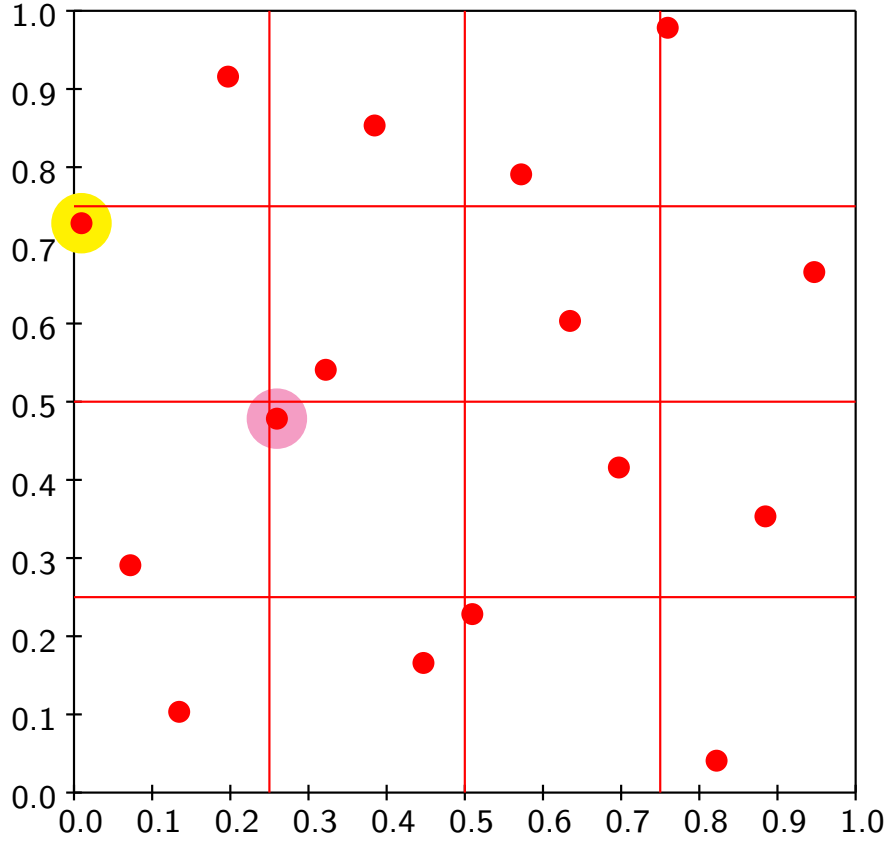
Mapping by batch sort and split sort

Good news: The state space does not have to be $[0,1)^d$:

States of the chains



Sobol' net + digital shift

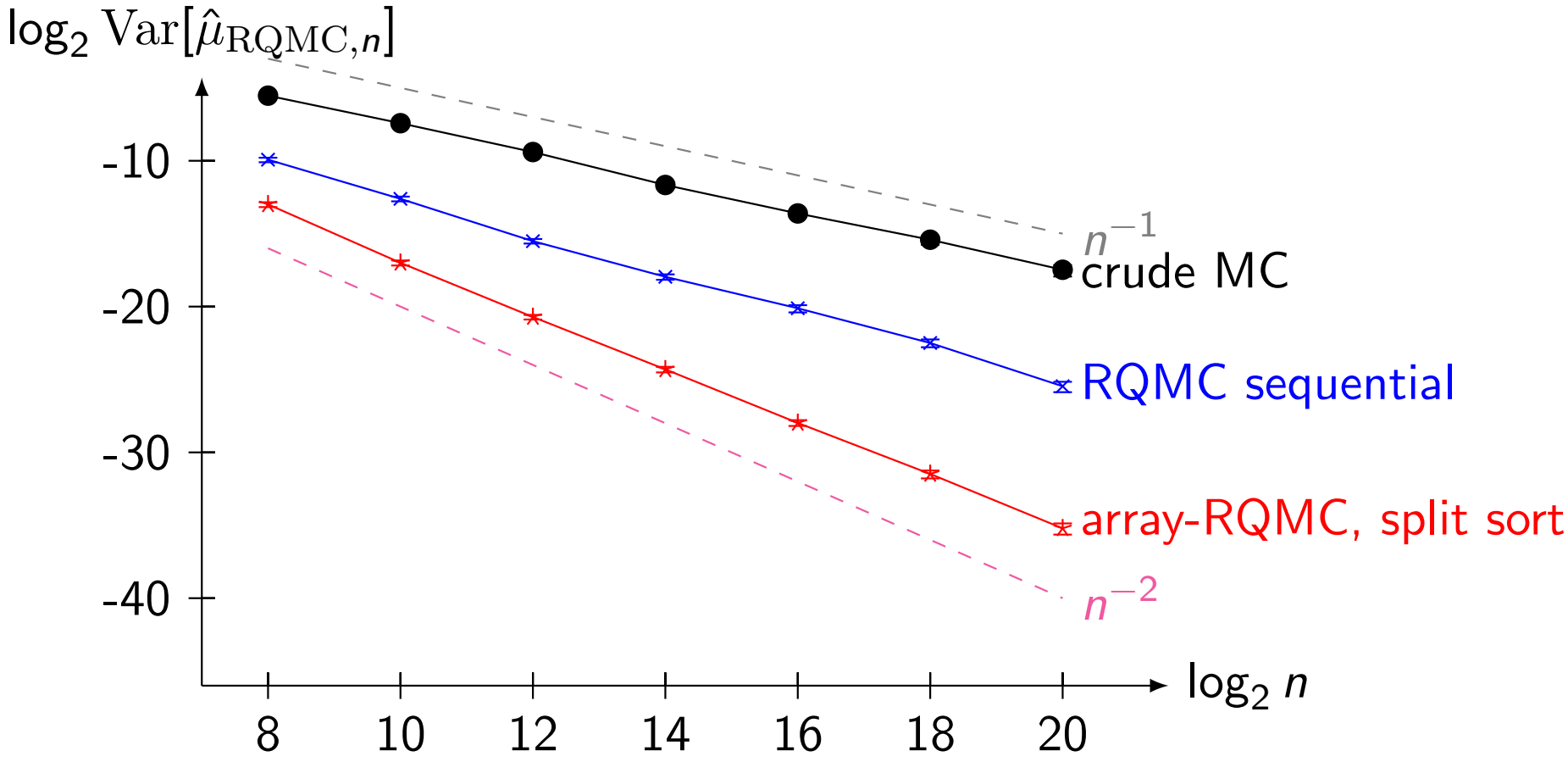


Example: Asian Call Option

$S(0) = 100$, $K = 100$, $r = 0.05$, $\sigma = 0.15$, $t_j = j/52$, $j = 0, \ldots, \tau = 13$.

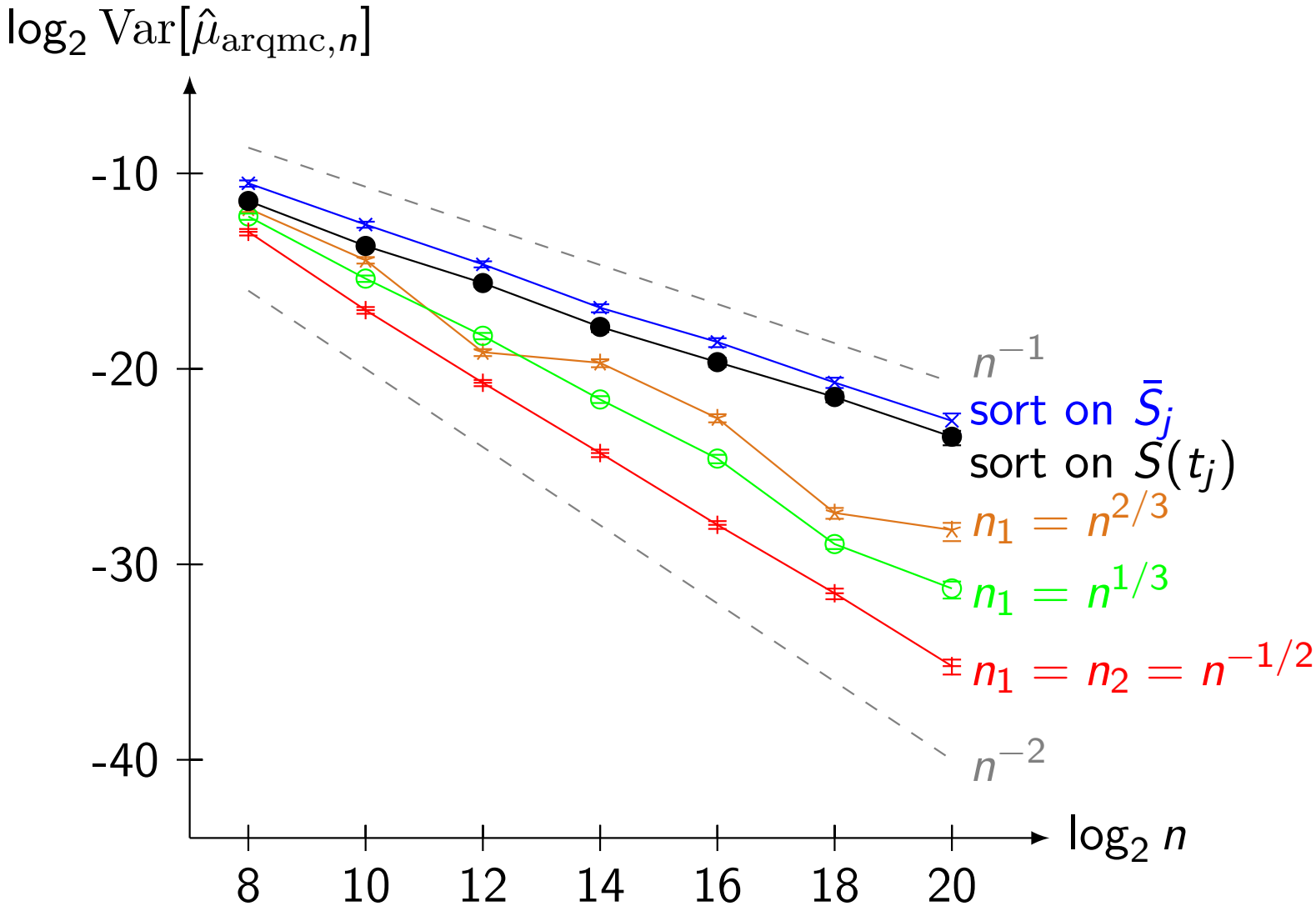
RQMC: Sobol' points with linear scrambling + random digital shift.

Similar results for randomly-shifted lattice + baker's transform.



Array-RQMC for Asian option, 2-dim. batch sort

Sort in n_1 packets based on $S(t_j)$, then sort the packets based on $\bar{S}_j$.



Lowering the state dimension

For large ℓ: Define a transformation $v : \mathcal{X} \to [0,1)^c$ for $c < \ell$.
Sort the transformed points $v(X_{i,j})$ in c dimensions.
Now we only need $c + d$ dimensions for the RQMC point sets;
c for the mapping and d to advance the chain.

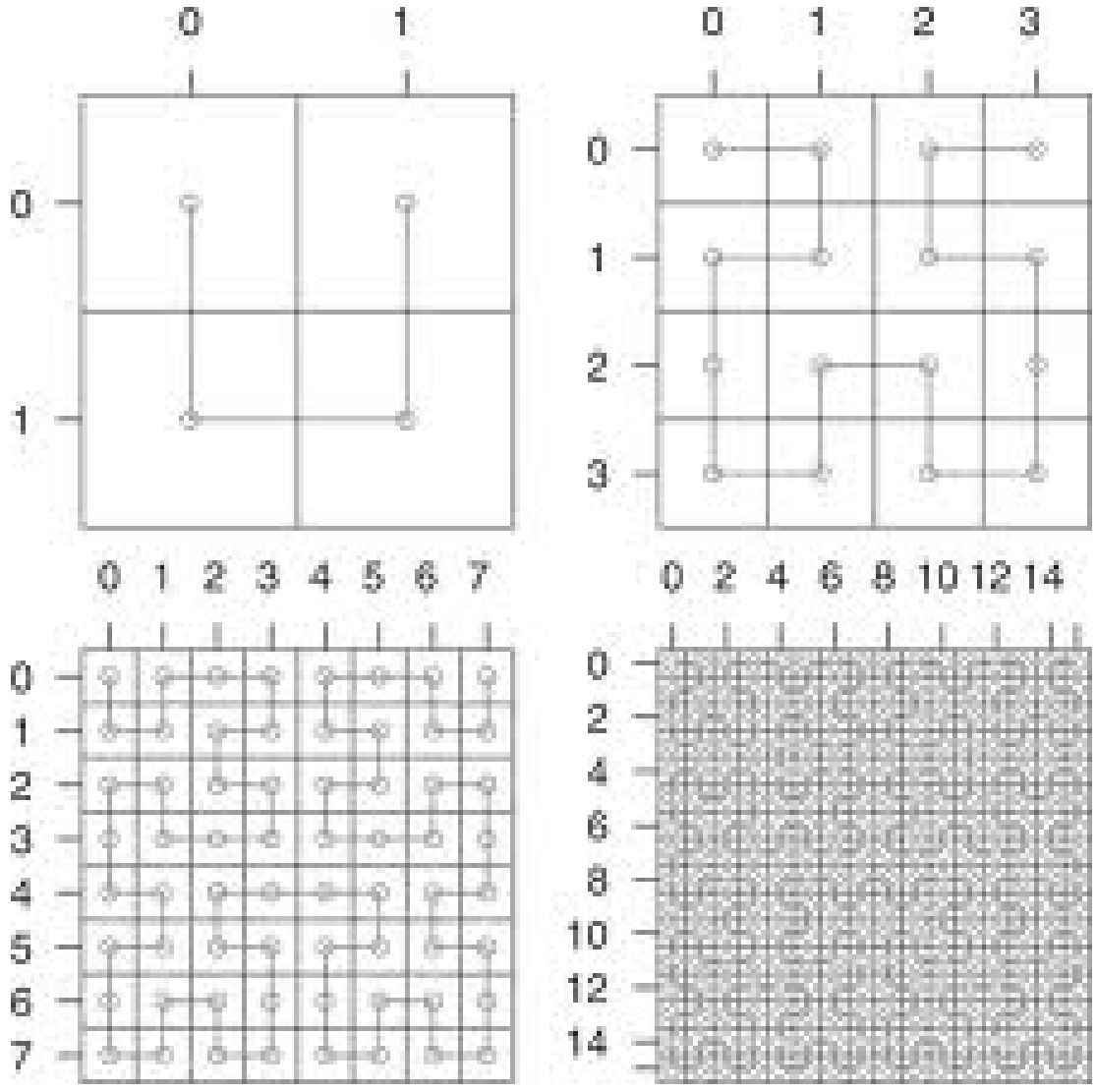
Choice of v: states mapped to nearby values should be nearly equivalent.

For $c = 1$, $\mathcal{X}$ is mapped to $[0,1)$, which leads to a one-dim sort.

The mapping v with $c = 1$ can be based on a space-filling curve:
Z-curve, in Wächter and Keller [2008];
Hilbert curve (also mentioned by W. and K.), is used in Gerber and Chopin [2015], who prove $o(n^{-1})$ convergence for the variance.

We only need a good pairing between states and RQMC points.
Any good way of doing this is welcome!

Hilbert curve sort



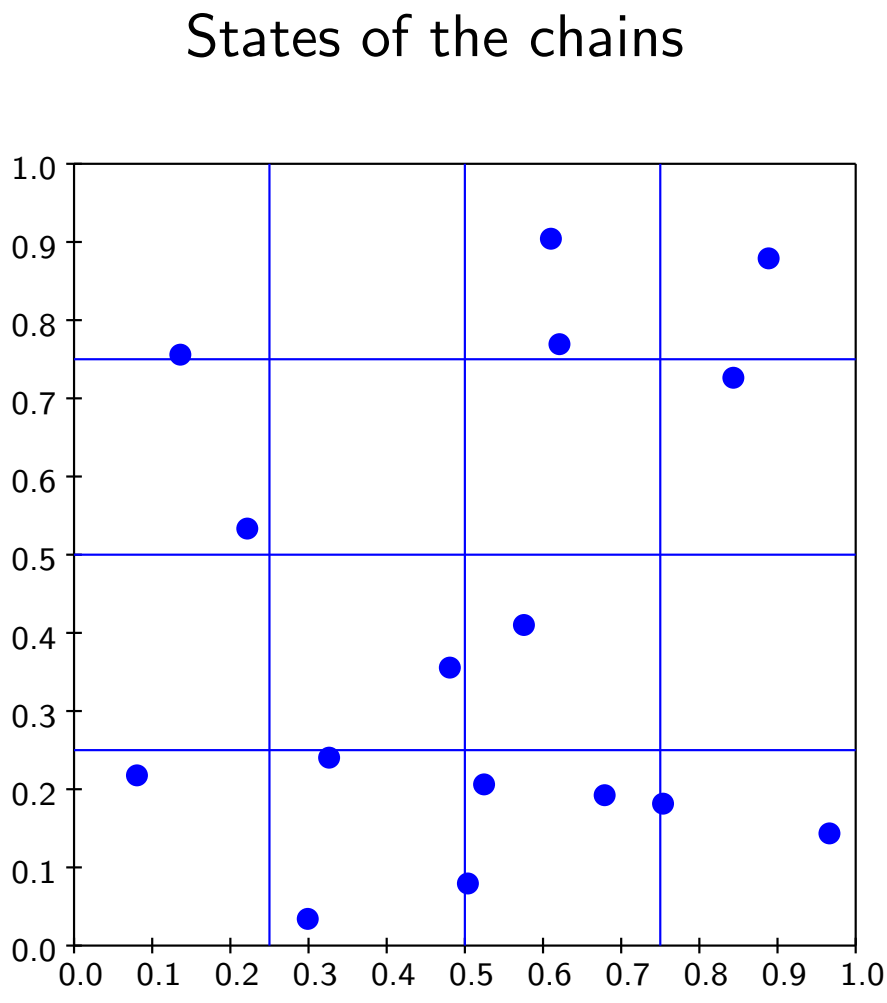
Sorting by a Hilbert curve

Suppose the state space is $\mathcal{X} = [0,1)^\ell$.
Partition this cube into $2^{k\ell}$ subcubes of equal size.
While any subcube contains more than one point, partition it in 2^ℓ.

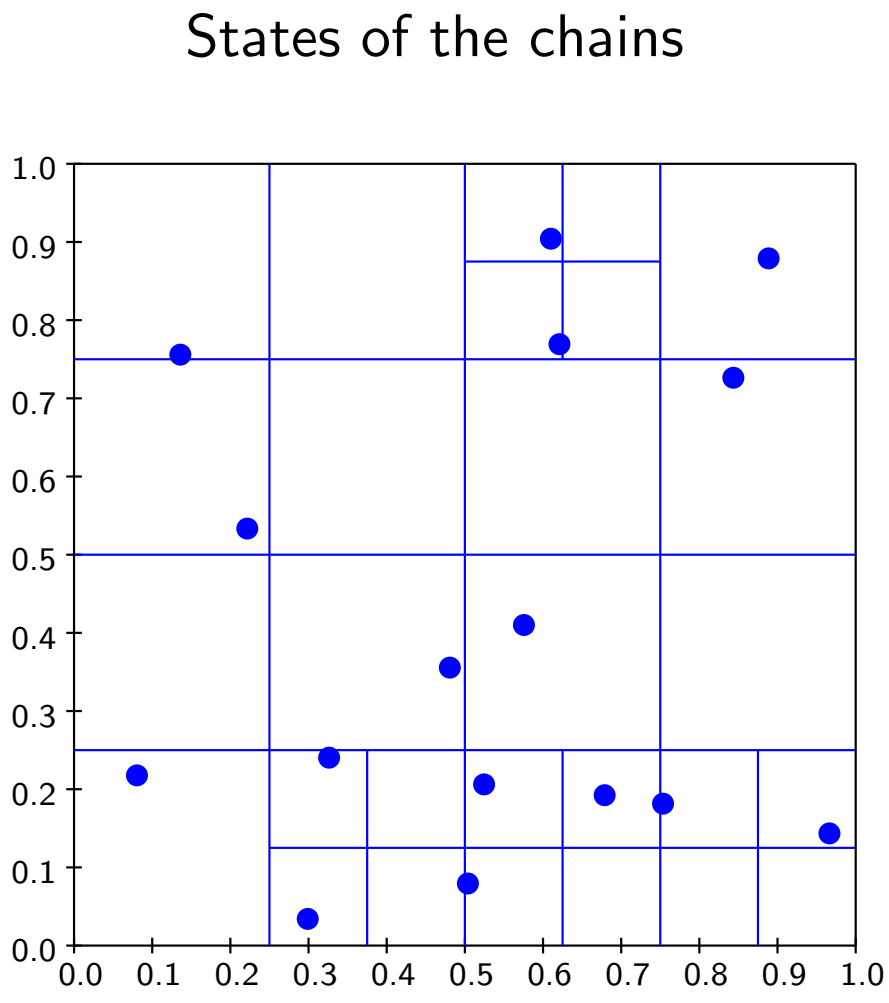
The Hilbert curve defines a way to enumerate (order) the subcubes so that successive subcubes are always adjacent.
This gives a way to sort the points.
Reasonable "traveling salesman" path!

Then we can map states to points as if the state had one dimension.
We use RQMC points in $1 + d$ dimensions, ordered by first coordinate, which is used to match the states, and d (randomized) coordinates are used to advance the chains.

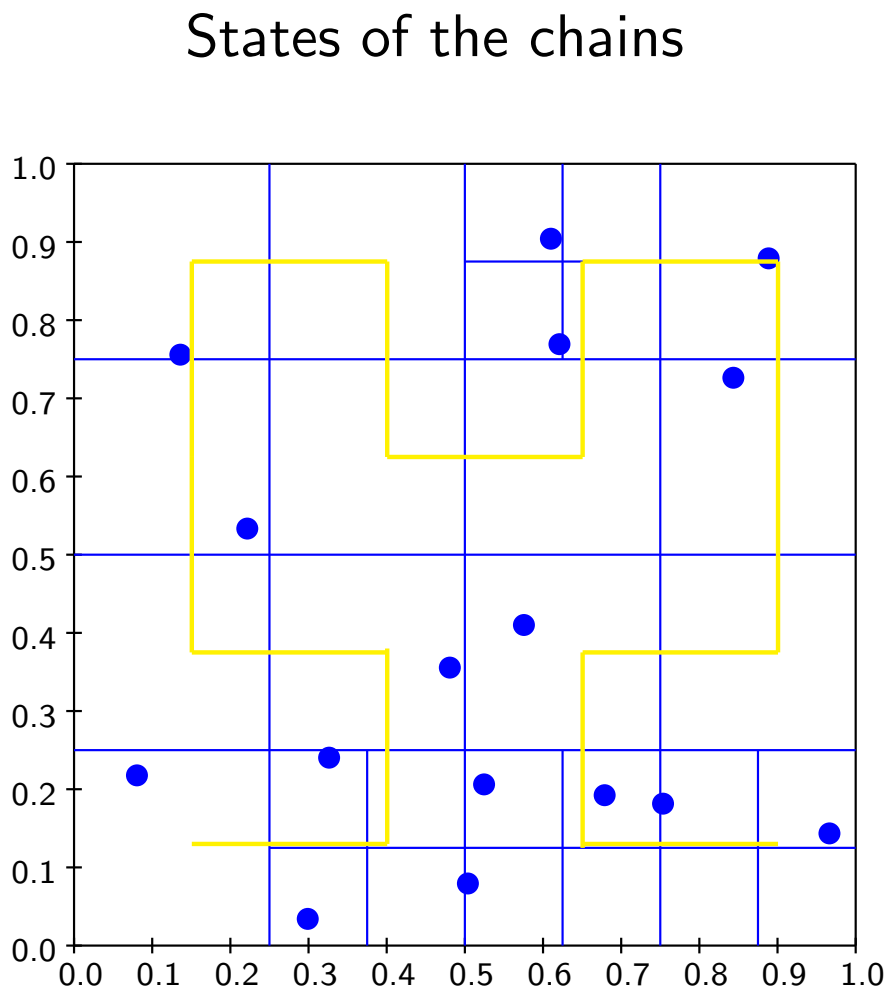
Hilbert curve sort



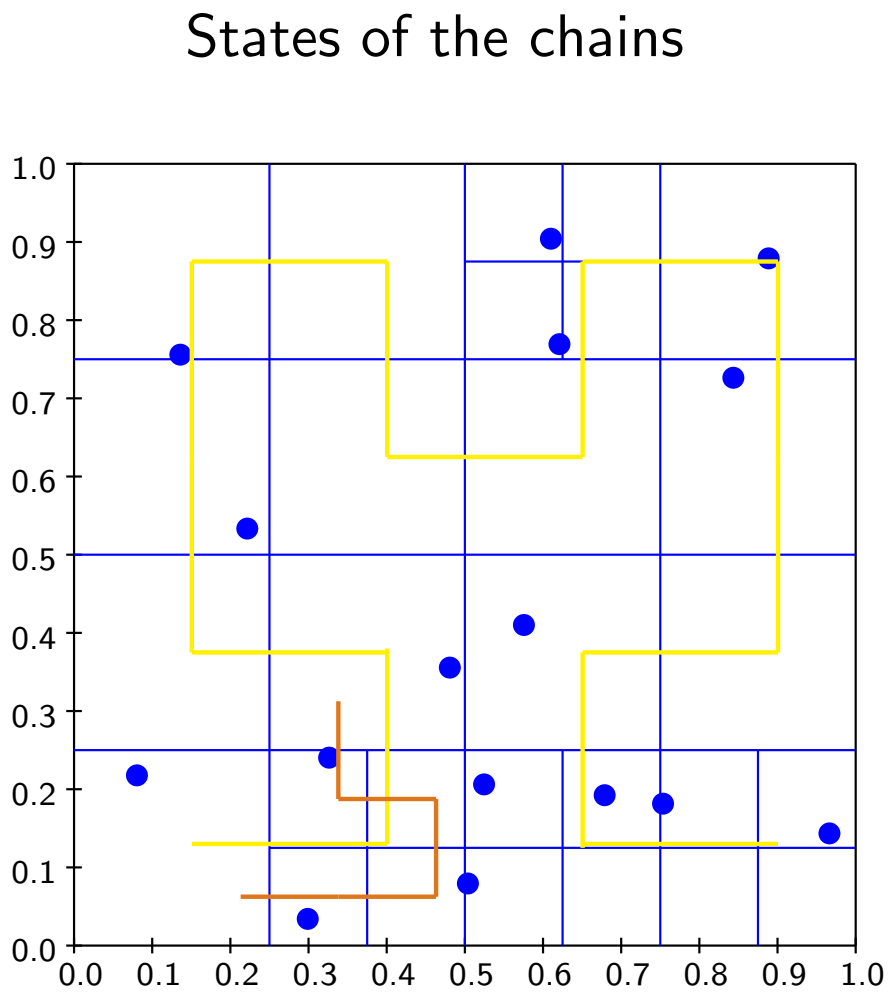
Hilbert curve sort



Hilbert curve sort



Hilbert curve sort



What if state space is not $[0,1)^\ell$?

Ex.: For the Asian option, $\mathcal{X} = [0,\infty)^2$.

Then one must define a transformation $\psi : \mathcal{X} \to [0,1)^\ell$ so that the transformed state is approximately uniformly distributed over $[0,1)^\ell$.

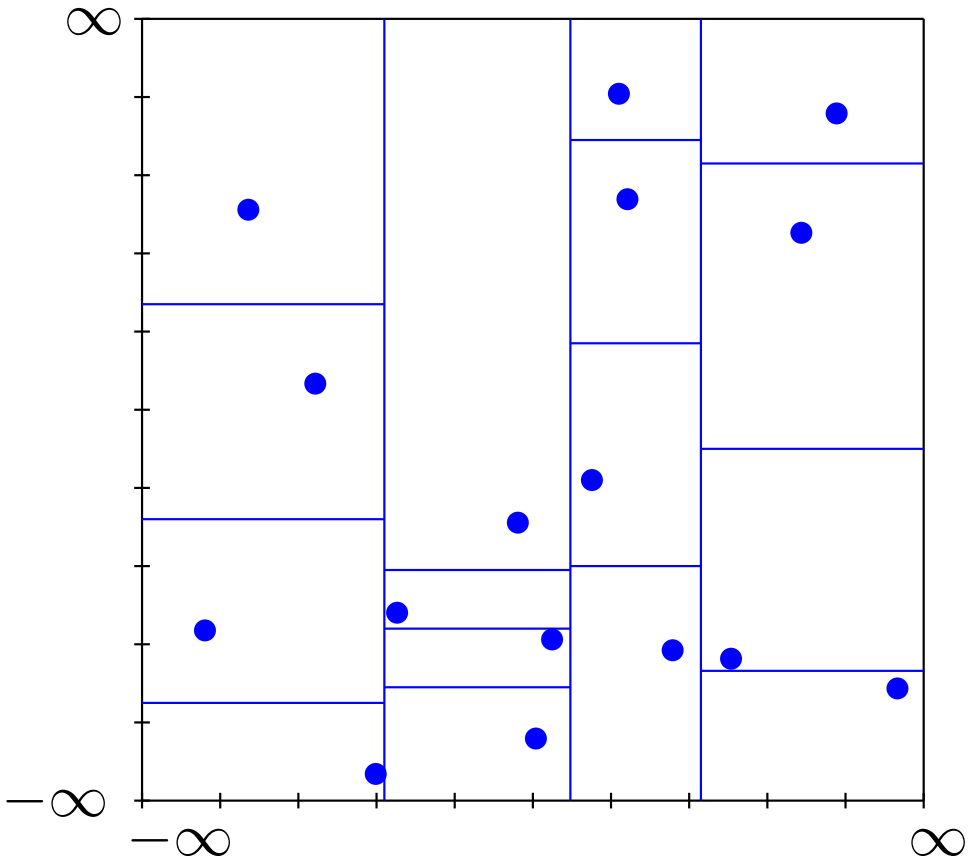
Not easy to find a good ψ in general!
Gerber and Chopin [2015] propose basically trial and error.

A lousy choice can kill efficiency.

Hilbert curve batch sort

Perform a multivariate batch sort, or a split sort, and then enumerate the boxes as in the Hilbert curve sort.

Big advantage: the state space can be $\mathbb{R}^\ell$.



Example: Asian Call Option

$S(0) = 100$, $K = 100$, $r = \ln(1.09)$, $\sigma = 0.2$, $t_j = (230 + j)/365$, $j = 1, \ldots, \tau = 10$.

Example: Asian Call Option

Slope ... vs $\log_2 n$:

RQMC points	$\log_2 \mathrm{Var}[\bar{Y}_{n,j}]$	RQMC VRF for $n = 2^{18}$
Split sort		
Sobol2	-1.86	2.7×10^4
Korobov+baker	-1.91	4.7×10^5
Batch sort (equal)		
Sobol2	-1.87	2.4×10^4
Korobov+baker	-1.91	4.7×10^5
Hilbert batch sort		
Stratif.	-1.56	1.4×10^3
Sobol2	-1.80	1.4×10^4
Korobov+baker	-1.89	4.6×10^5

Convergence results and proofs

L., Lécot, Tuffin [2008] + some extensions.

Simple case: suppose $\ell = d = 1$, $\mathcal{X} = [0,1]$, and $X_j \sim U(0,1)$. Define

$$\Delta_j = \sup_{x \in \mathcal{X}} |\hat{F}_j(x) - F_j(x)| \qquad \text{(discrepancy of states)}$$

$$V(g_j) = \int_0^1 |dg_j(x)| \qquad \text{(variation of } g_j\text{)}$$

Theorem. $\left|\bar{Y}_{n,j} - E[g_j(X_j)]\right| \leq \Delta_j V(g_j)$.

Convergence results and proofs

Assumption 1. $\varphi_j(x, u)$ non-decreasing in u. That is, we use inversion to generate next state from cdf $F_j(z \mid \cdot\,) = P[X_j \le z \mid X_{j-1} = \cdot\,]$.

$$\text{Let} \qquad \Lambda_j \;=\; \sup_{0 \le z \le 1} V(F_j(z \mid \cdot\,]).$$

Convergence results and proofs

Assumption 1. $\varphi_j(x, u)$ non-decreasing in u. That is, we use inversion to generate next state from cdf $F_j(z \mid \cdot\,) = P[X_j \le z \mid X_{j-1} = \cdot\,]$.

$$\text{Let} \qquad \Lambda_j \;=\; \sup_{0 \le z \le 1} V(F_j(z \mid \cdot\,]).$$

Assumption 2. Each square of $\sqrt{n} \times \sqrt{n}$ grid has one RQMC point.

Proposition. (Worst-case error.) Under Assumptions 1 and 2,

$$\Delta_j \le n^{-1/2} \sum_{k=1}^{j} (\Lambda_k + 1) \prod_{i=k+1}^{j} \Lambda_i.$$

Corollary. If $\Lambda_j \le \rho < 1$ for all j, then

$$\Delta_j \le \frac{1+\rho}{1-\rho} n^{-1/2}.$$

Conclusion

We have convergence proofs for special cases, but not yet for the rates we observe in examples.

Many other sorting strategies remain to be explored.

Other examples and applications. Higher dimension.

Array-RQMC is good not only to estimate the mean more accurately, but also to estimate the entire distribution of the state.