Randomized Quasi-Monte Carlo:
Theory, Choice of Discrepancy, and Applications

Pierre L'Ecuyer and David Munger

Informatique et Recherche Opérationnelle, Université de Montréal

Randomized Quasi-Monte Carlo:
Theory, Choice of Discrepancy, and Applications
featuring randomly-shifted lattice rules

Pierre L'Ecuyer and David Munger

Informatique et Recherche Opérationnelle, Université de Montréal

1. Monte Carlo (MC), quasi-MC (QMC), randomized QMC (RQMC).
2. Lattice rules and RQMC variance.
3. Weighted discrepancies and choice of weights.
4. Several examples.

Basic Monte Carlo setting

Want to estimate

$$\mu = \mu(f) = \int_{[0,1)^s} f(\mathbf{u})\, d\mathbf{u} = \mathbb{E}[f(\mathbf{U})]$$

where $f : [0,1)^s \to \mathbb{R}$ and $\mathbf{U}$ is a uniform r.v. over $[0,1)^s$.

Standard Monte Carlo:

- Generate n independent copies of $\mathbf{U}$, say $\mathbf{U}_1, \ldots, \mathbf{U}_n$;
- estimate μ by $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{U}_i)$.

Basic Monte Carlo setting

Want to estimate

$$\mu = \mu(f) = \int_{[0,1)^s} f(\mathbf{u})\, d\mathbf{u} = \mathbb{E}[f(\mathbf{U})]$$

where $f : [0,1)^s \to \mathbb{R}$ and $\mathbf{U}$ is a uniform r.v. over $[0,1)^s$.

Standard Monte Carlo:

- Generate n independent copies of $\mathbf{U}$, say $\mathbf{U}_1, \ldots, \mathbf{U}_n$;
- estimate μ by $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{U}_i)$.

Almost sure convergence as $n \to \infty$ (strong law of large numbers).
For confidence interval of level $1-\alpha$, can use central limit theorem:

$$\mathbb{P}\left[\mu \in \left(\hat{\mu}_n - \frac{c_\alpha S_n}{\sqrt{n}},\ \ \hat{\mu}_n + \frac{c_\alpha S_n}{\sqrt{n}}\right)\right] \approx 1-\alpha,$$

where S_n^2 is any consistent estimator of $\sigma^2 = \mathrm{Var}[f(\mathbf{U})]$.

Quasi-Monte Carlo (QMC)

Replace the random points $\mathbf{U}_i$ by a set of deterministic points $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ that cover $[0,1)^s$ more evenly.
This P_n is called a highly-uniform or low-discrepancy point set if some measure of discrepancy between the empirical distribution of P_n and the uniform distribution $\to 0$ faster than for independent random points.

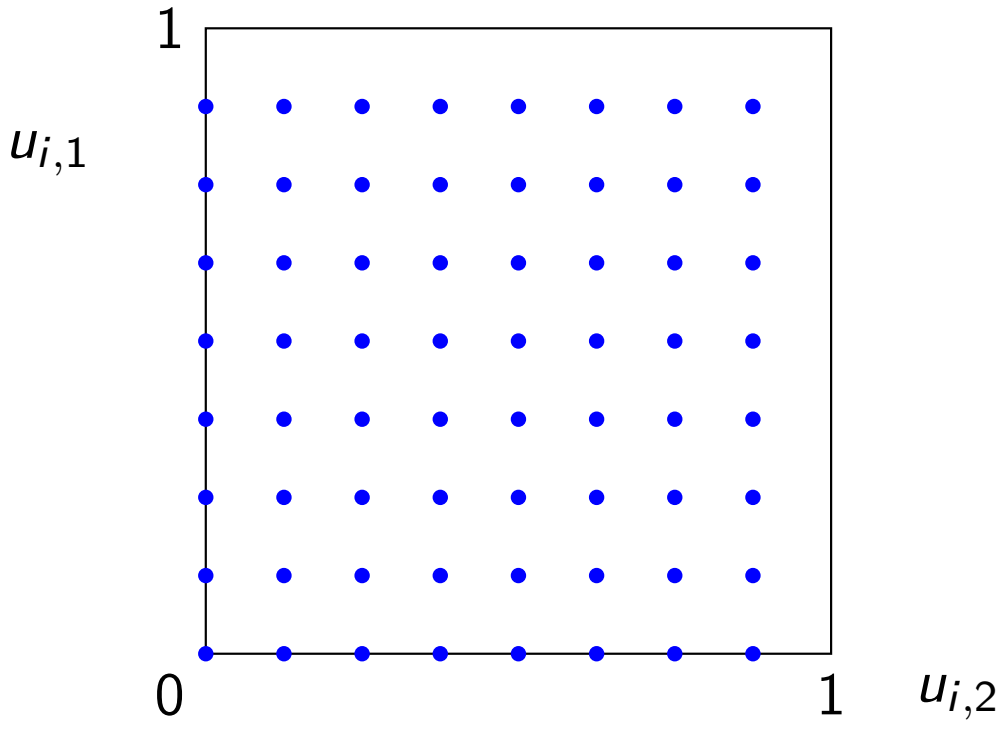
Quasi-Monte Carlo (QMC)

Replace the random points $\mathbf{U}_i$ by a set of deterministic points $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ that cover $[0,1)^s$ more evenly.
This P_n is called a highly-uniform or low-discrepancy point set if some measure of discrepancy between the empirical distribution of P_n and the uniform distribution $\to 0$ faster than for independent random points.

Main construction methods: lattice rules and digital nets
(Korobov, Hammersley, Halton, Sobol', Faure, Niederreiter, etc.)

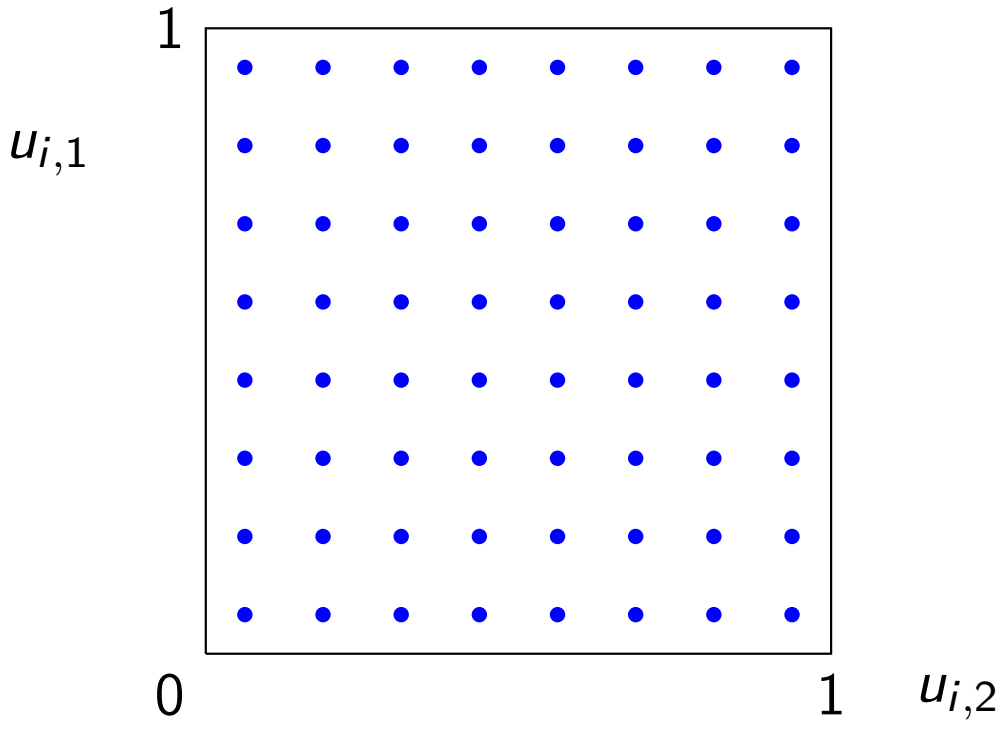
Simplistic solution: rectangular grid

$P_n = \{(i_1/d, \ldots, i_s/d) \text{ such that } 0 \leq i_j < d \quad \forall j\}$ where $n = d^s$.



Simplistic solution: rectangular grid

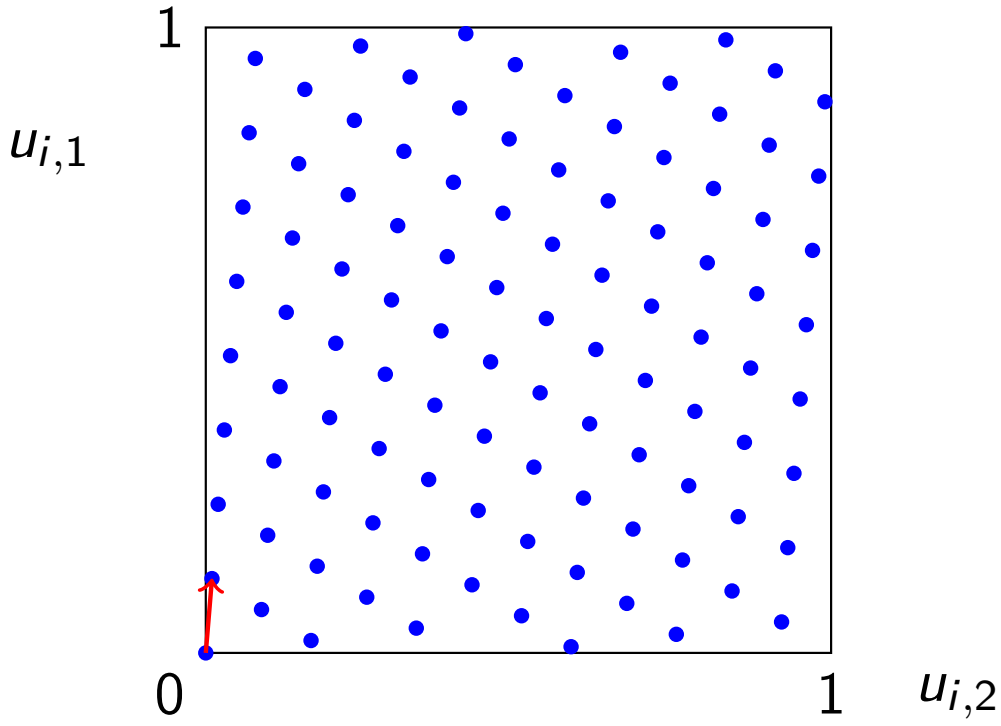
$P_n = \{(i_1/d, \ldots, i_s/d) \text{ such that } 0 \leq i_j < d \ \ \forall j\}$ where $n = d^s$.



Quickly becomes impractical when s increases.
And each one-dimensional projection has only d distinct points, each two-dimensional projections has only d^2 distinct points, etc.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

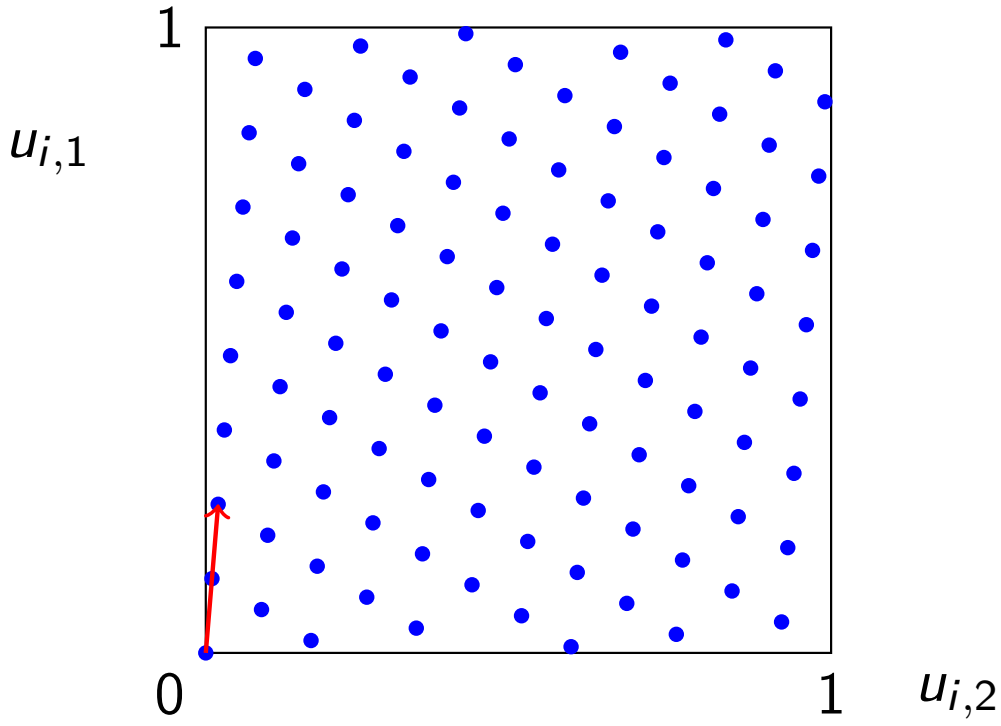
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

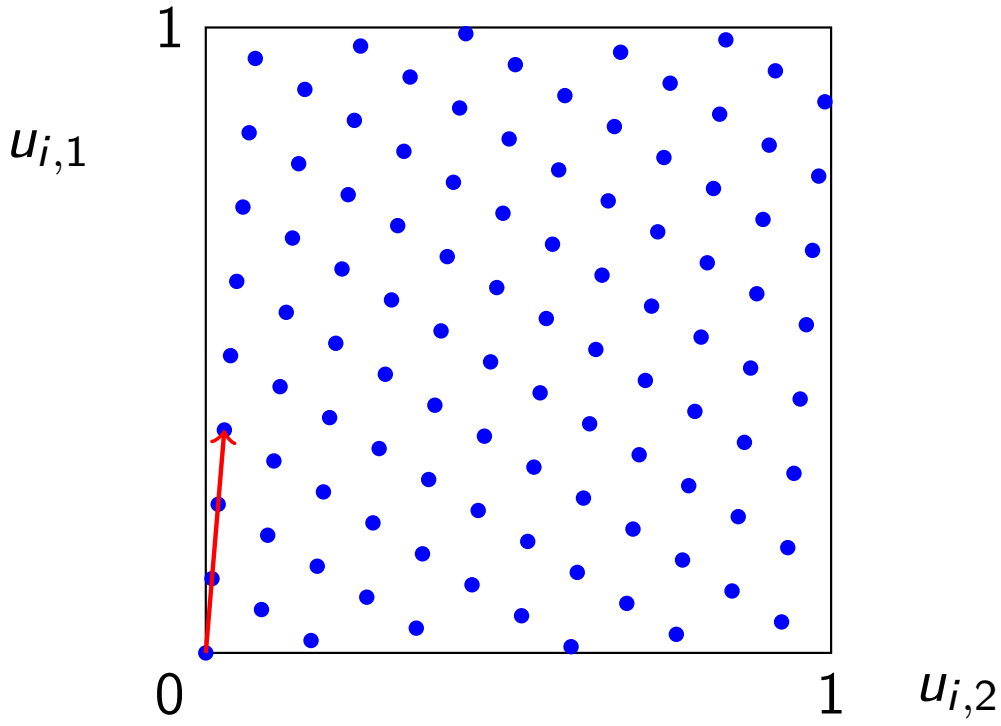
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

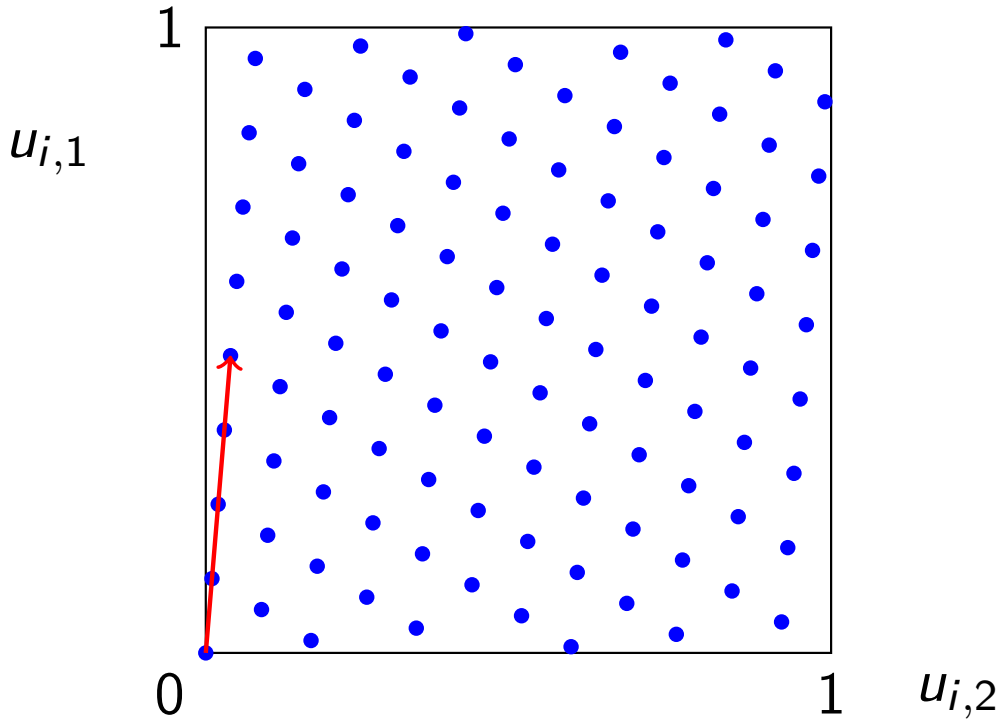
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

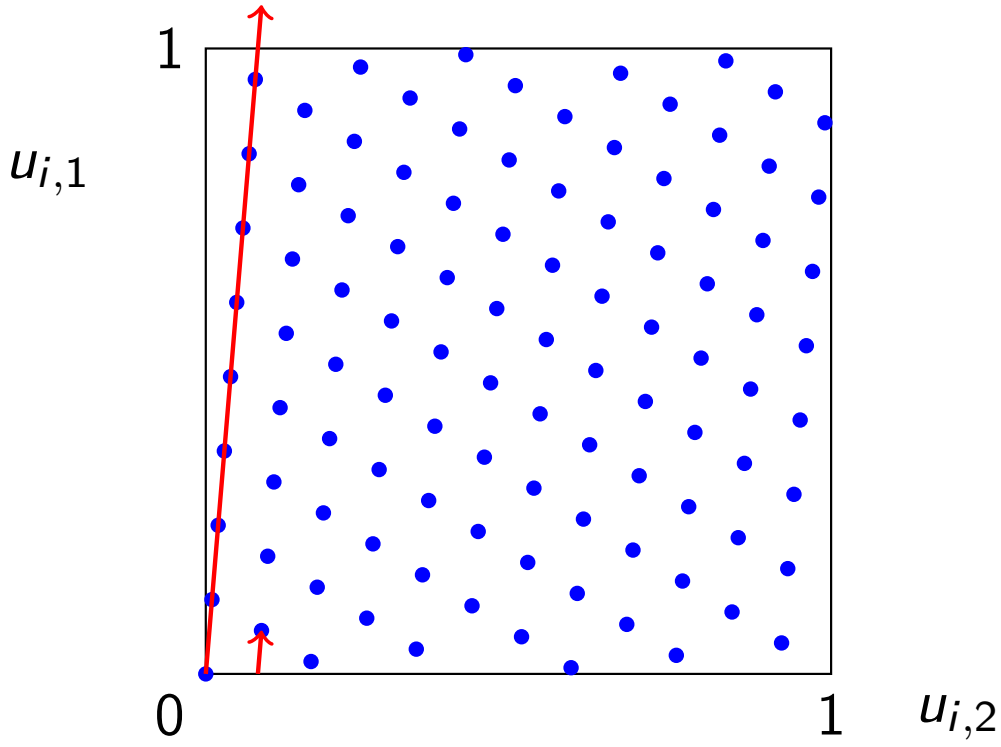
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

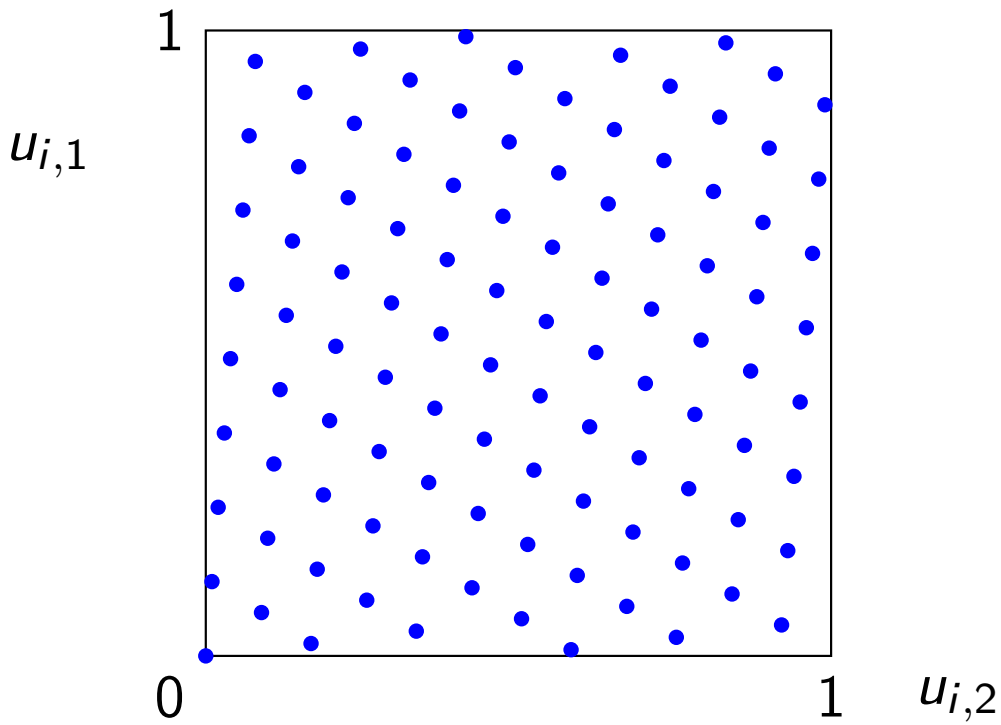
$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Example: lattice with $s = 2$, $n = 101$, $a = 12$

$$
\begin{aligned}
P_n &= \{(x/m, (ax/m) \bmod 1) : x = 0, \ldots, m-1\} \\
&= \{(x/101, (12x/101) \bmod 1) : x = 0, \ldots, 100\}
\end{aligned}
$$



Here, each one-dimensional projection is $\{0, 1/n, \ldots, (n-1)/n\}$.

Two problems: (1) point at $(0, 0)$ and (2) how to estimate the error?

Randomized quasi-Monte Carlo (RQMC)

An RQMC estimator of μ has the form

$$\hat{\mu}_{n,\mathrm{rqmc}} = \frac{1}{n}\sum_{i=0}^{n-1} f(\mathbf{U}_i),$$

with $P_n = \{\mathbf{U}_0, \ldots, \mathbf{U}_{n-1}\} \subset (0,1)^s$ an RQMC point set:

(i) each point $\mathbf{U}_i$ has the uniform distribution over $(0,1)^s$;

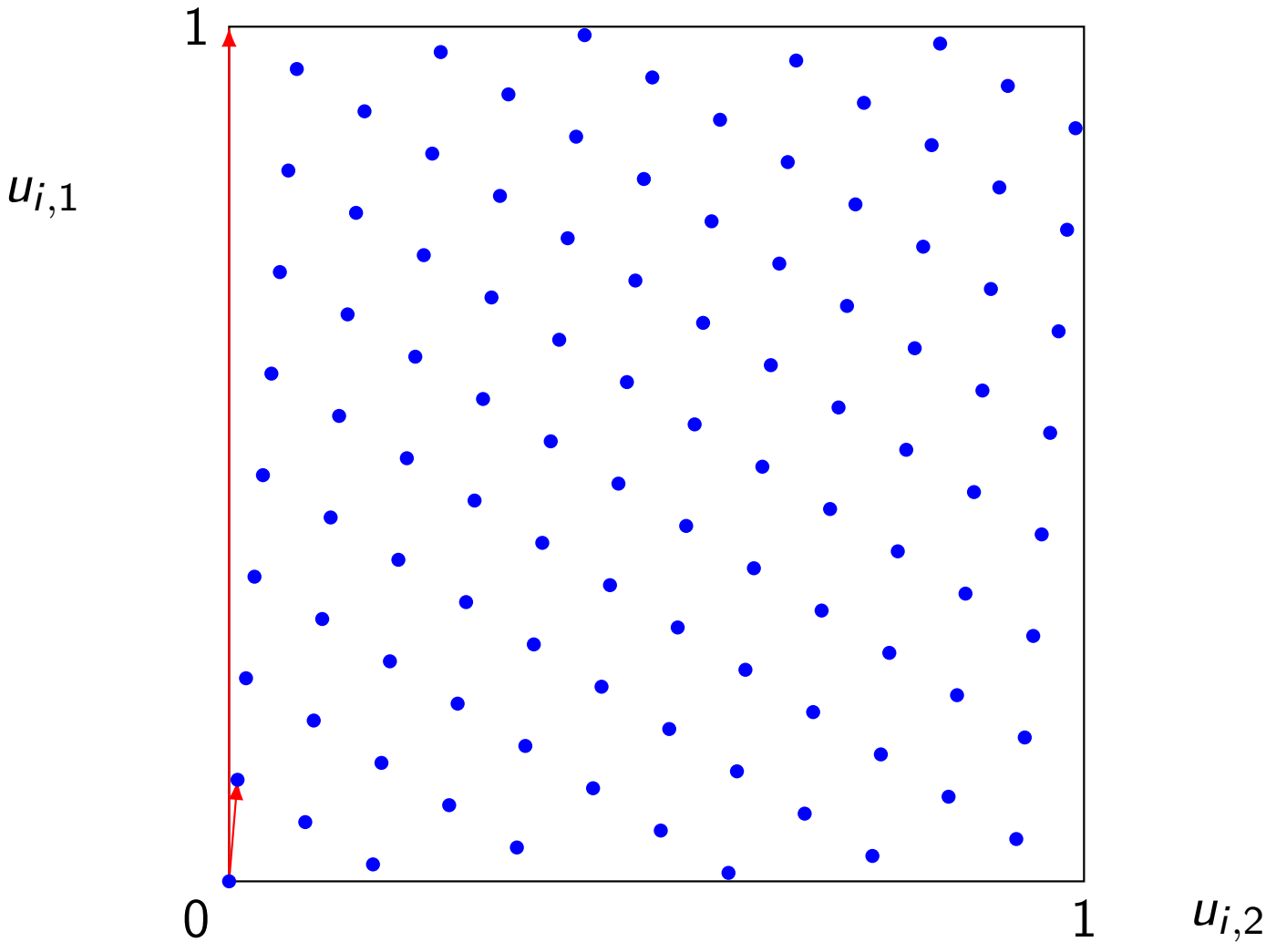
(ii) P_n as a whole is a low-discrepancy point set.

$$\mathbb{E}[\hat{\mu}_{n,\mathrm{rqmc}}] = \mu \qquad \text{(unbiased)}.$$

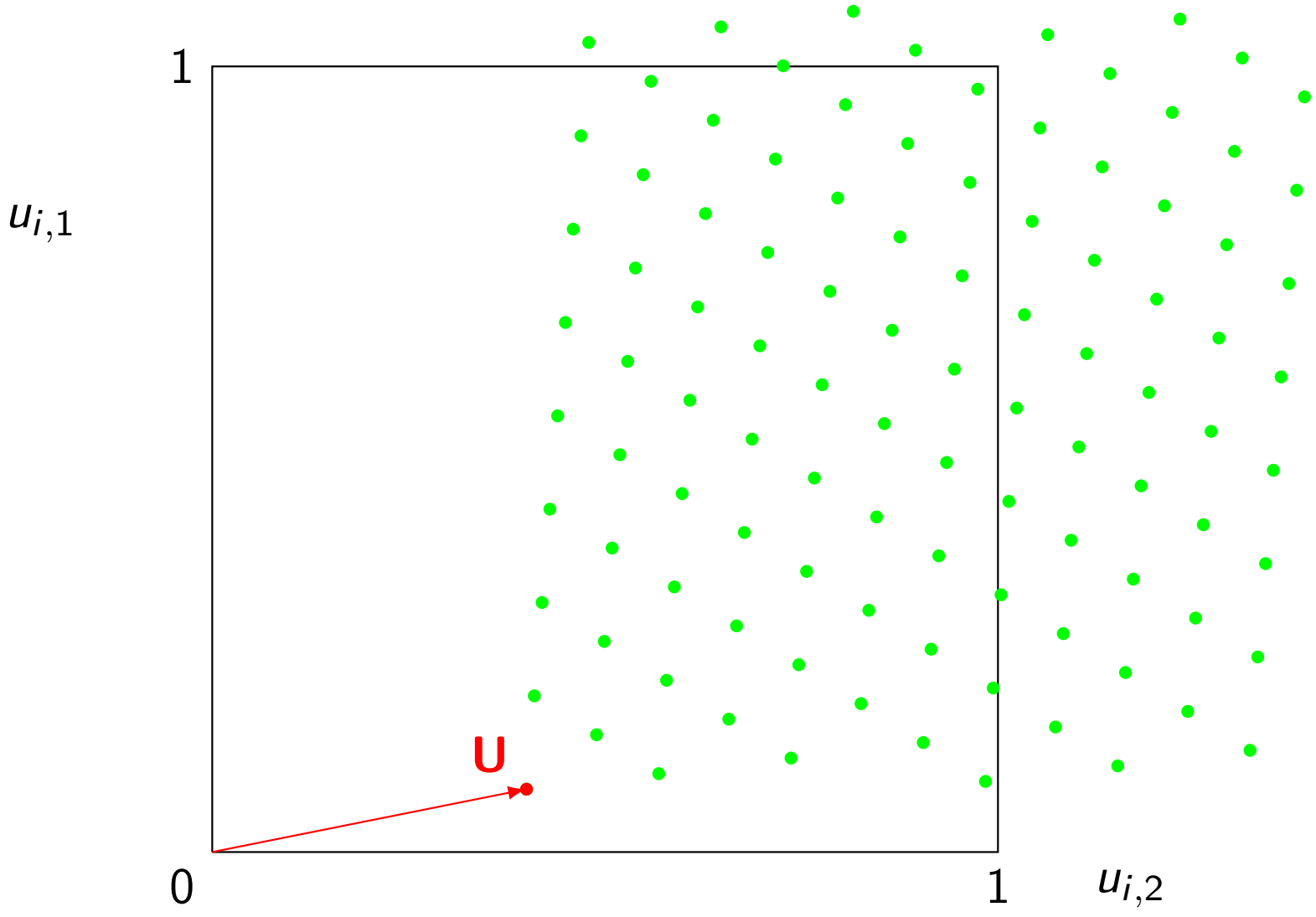
Can perform m independent realizations $X_1, \ldots, X_m$ of $\hat{\mu}_{n,\mathrm{rqmc}}$, then estimate μ and $\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}]$ by their sample mean $\bar{X}_m$ and sample variance S_m^2 (also unbiased).

Temptation: assume that $\bar{X}_m$ has the normal distribution.

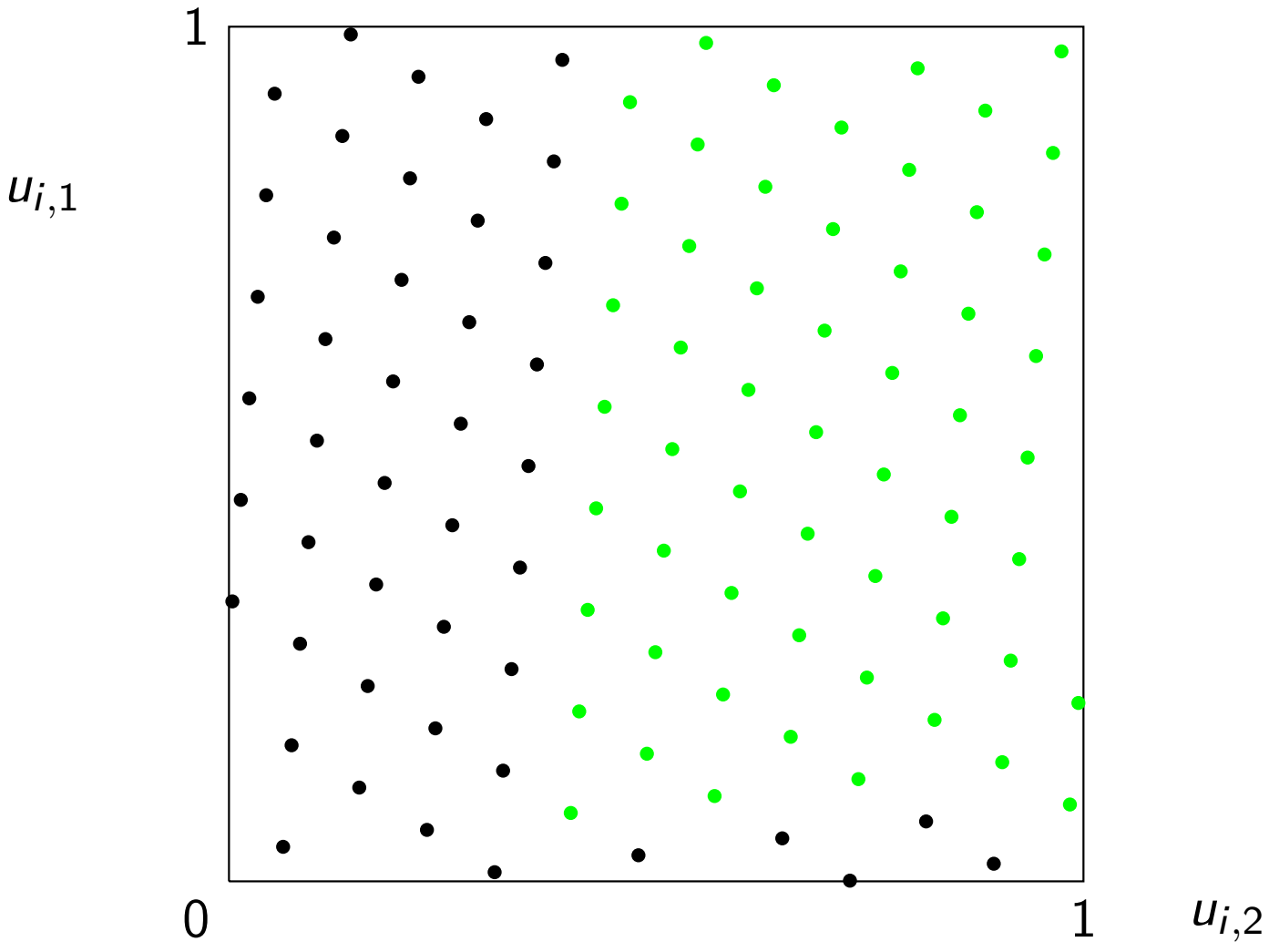
Example: lattice with $s = 2$, $n = 101$, $a = 12$



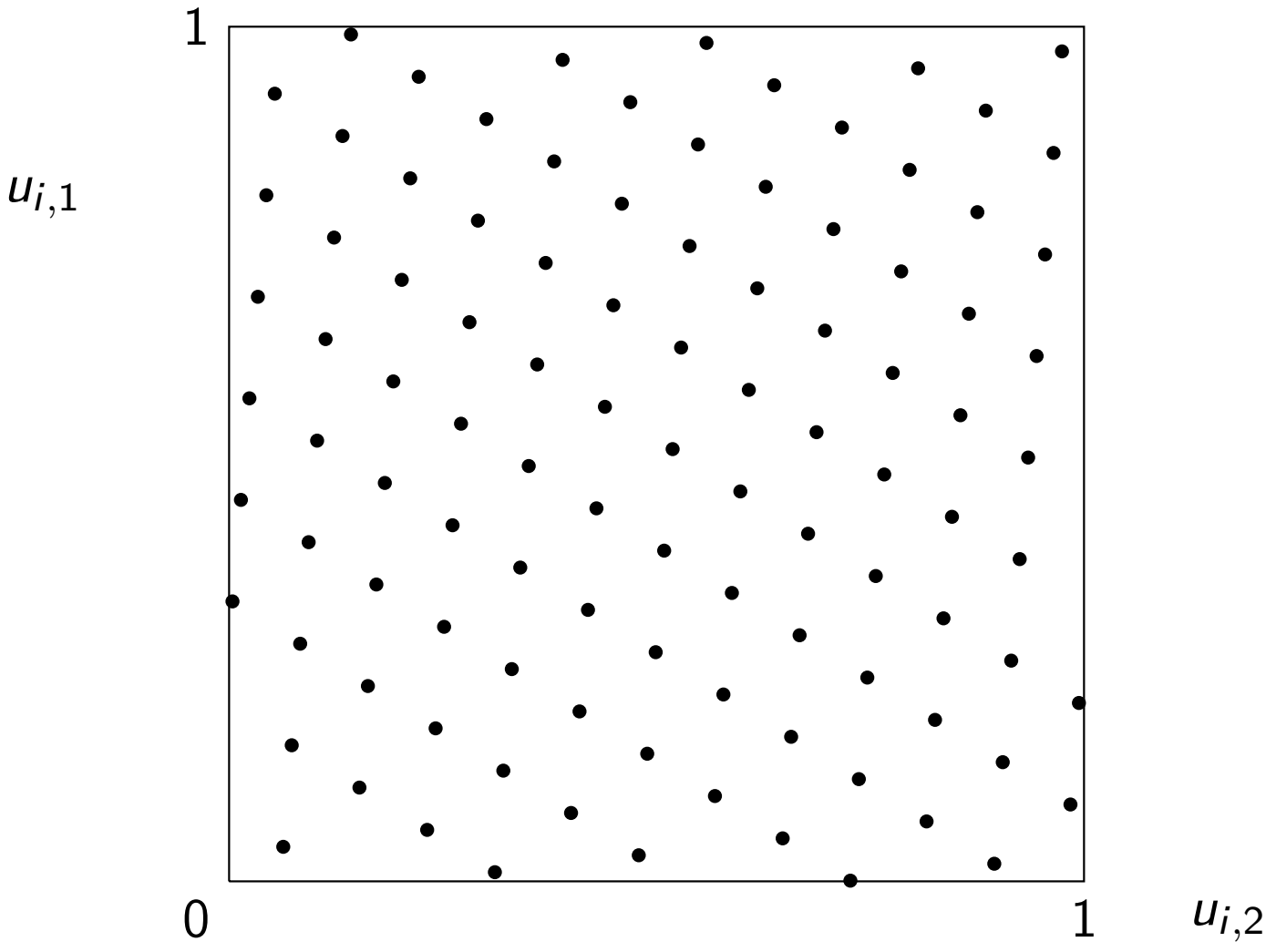
Example: lattice with $s = 2$, $n = 101$, $a = 12$



Example: lattice with $s = 2$, $n = 101$, $a = 12$



Example: lattice with $s = 2$, $n = 101$, $a = 12$



Generalized antithetic variates and RQMC

$$\begin{aligned}\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] &= \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\mathrm{Cov}[f(\mathbf{U}_i), f(\mathbf{U}_j)] \\ &= \frac{\mathrm{Var}[f(\mathbf{U}_i)]}{n} + \frac{2}{n^2}\sum_{i<j}\mathrm{Cov}[f(\mathbf{U}_i), f(\mathbf{U}_j)].\end{aligned}$$

We want to make the last sum as negative as possible.

Special cases:

- antithetic variates ($n = 2$),
- Latin hypercube sampling (LHS),
- randomized quasi-Monte Carlo (RQMC).

Lattice rules

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and where L_s contains $\mathbb{Z}^s$. Lattice rule: Take $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

Lattice rules

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and where L_s contains $\mathbb{Z}^s$. Lattice rule: Take $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$,
where $n\mathbf{v}_1 = \mathbf{z} = (z_1, \ldots, z_s) \in \{0, 1, \ldots, n-1\}$.
Korobov rule: $\mathbf{z} = (1, a, a^2 \mod n, \ldots)$.

Lattice rules

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and where L_s contains $\mathbb{Z}^s$. Lattice rule: Take $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$,
where $n\mathbf{v}_1 = \mathbf{z} = (z_1, \ldots, z_s) \in \{0, 1, \ldots, n-1\}$.
Korobov rule: $\mathbf{z} = (1, a, a^2 \mod n, \ldots)$.

For any $\mathfrak{u} \subset \{1, \ldots, s\}$, the projection $L_s(\mathfrak{u})$ of L_s is also a lattice.

Lattice rules

Integration lattice:

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

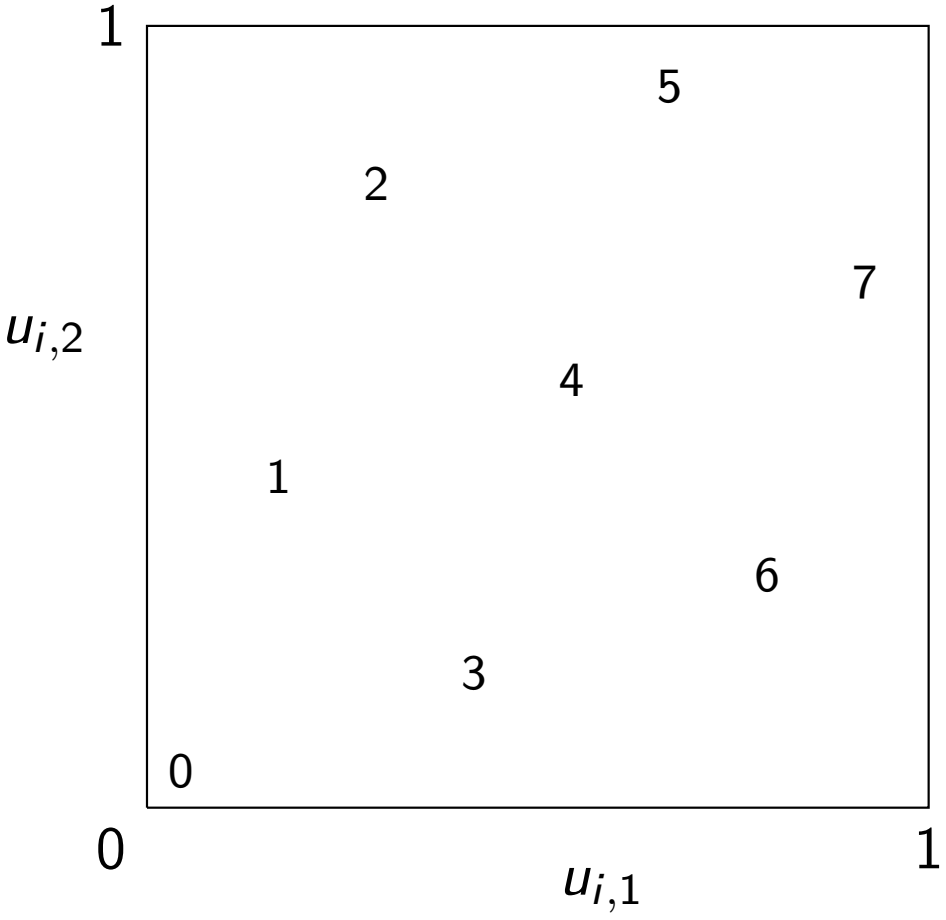
where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and where L_s contains $\mathbb{Z}^s$. Lattice rule: Take $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$.

Lattice rule of rank 1: $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$,
where $n\mathbf{v}_1 = \mathbf{z} = (z_1, \ldots, z_s) \in \{0, 1, \ldots, n-1\}$.
Korobov rule: $\mathbf{z} = (1, a, a^2 \mod n, \ldots)$.

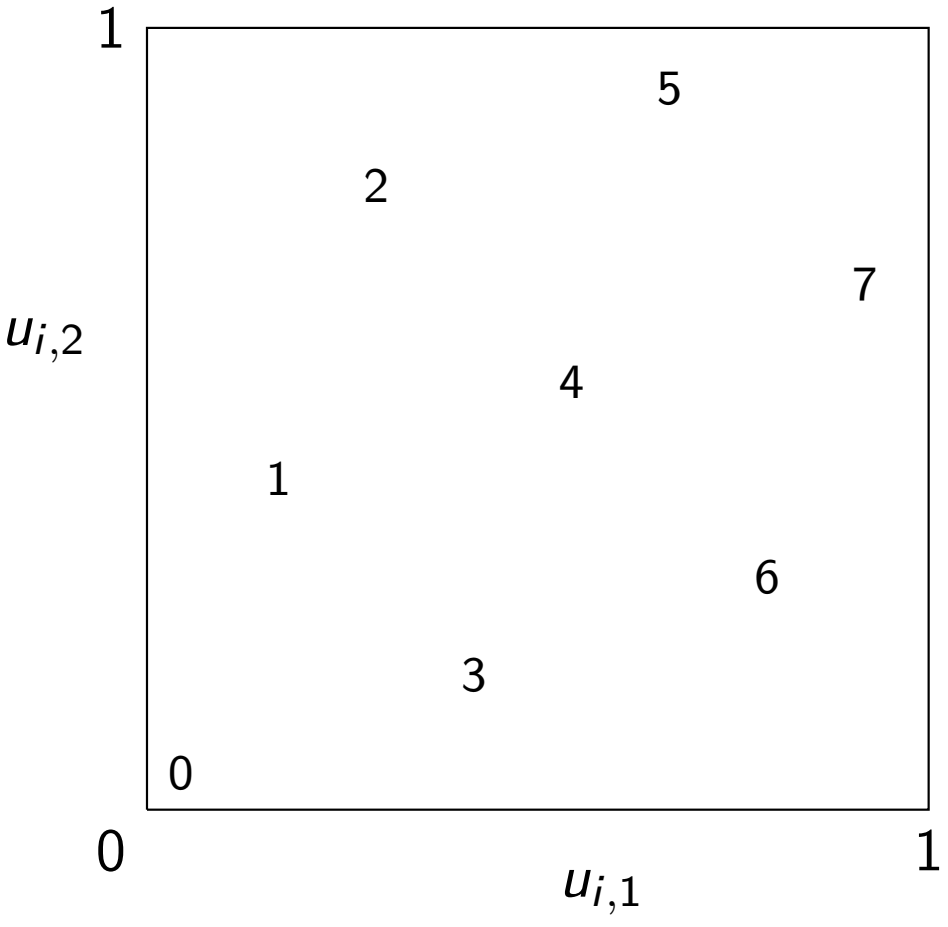
For any $\mathfrak{u} \subset \{1, \ldots, s\}$, the projection $L_s(\mathfrak{u})$ of L_s is also a lattice.

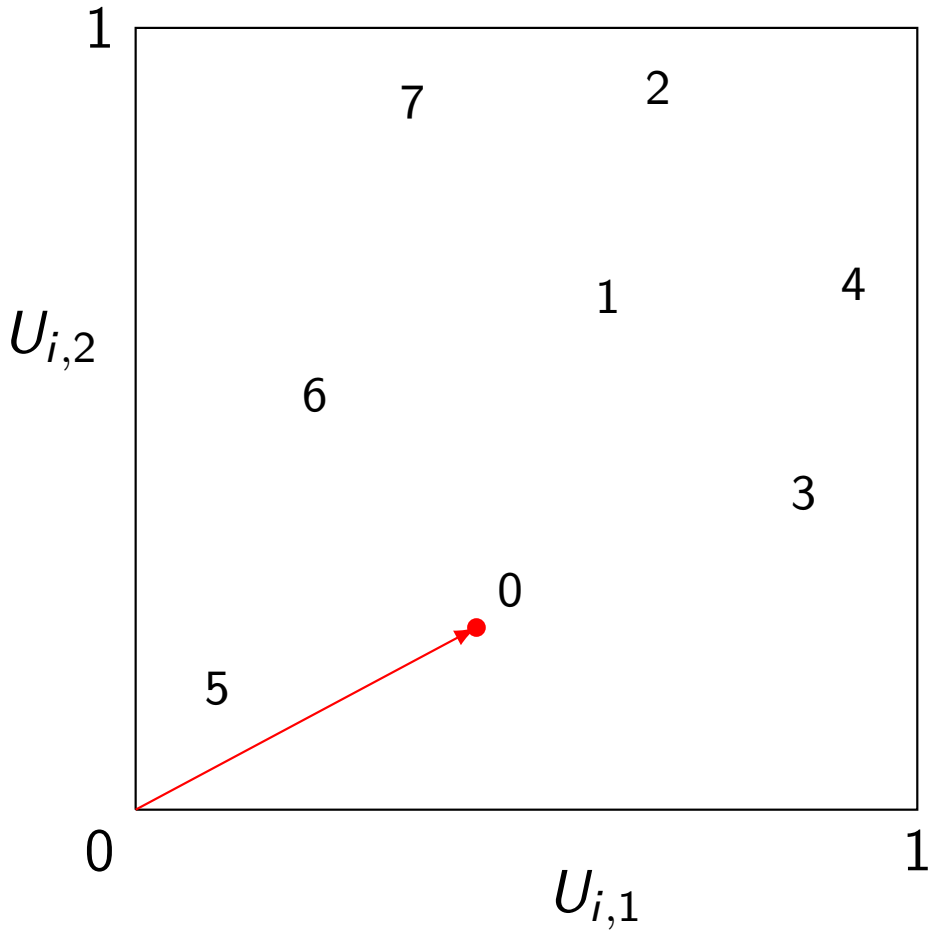
Random shift modulo 1: generate a single point $\mathbf{U}$ uniformly over $(0,1)^s$ and add it to each point of P_n, modulo 1, coordinate-wise:
$\mathbf{U}_i = (\mathbf{u}_i + \mathbf{U}) \bmod 1$. Each $\mathbf{U}_i$ is uniformly distributed over $[0,1)^s$.

A small lattice shifted by the red vector, modulo 1.

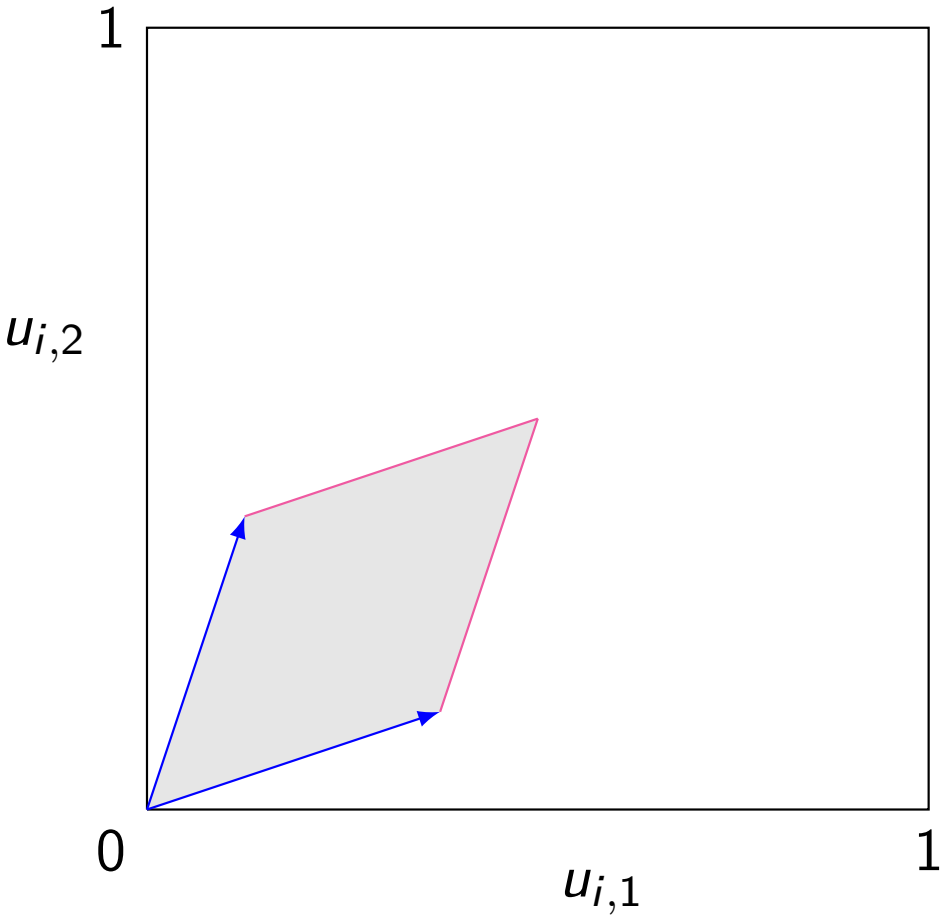


A small lattice shifted by the red vector, modulo 1.

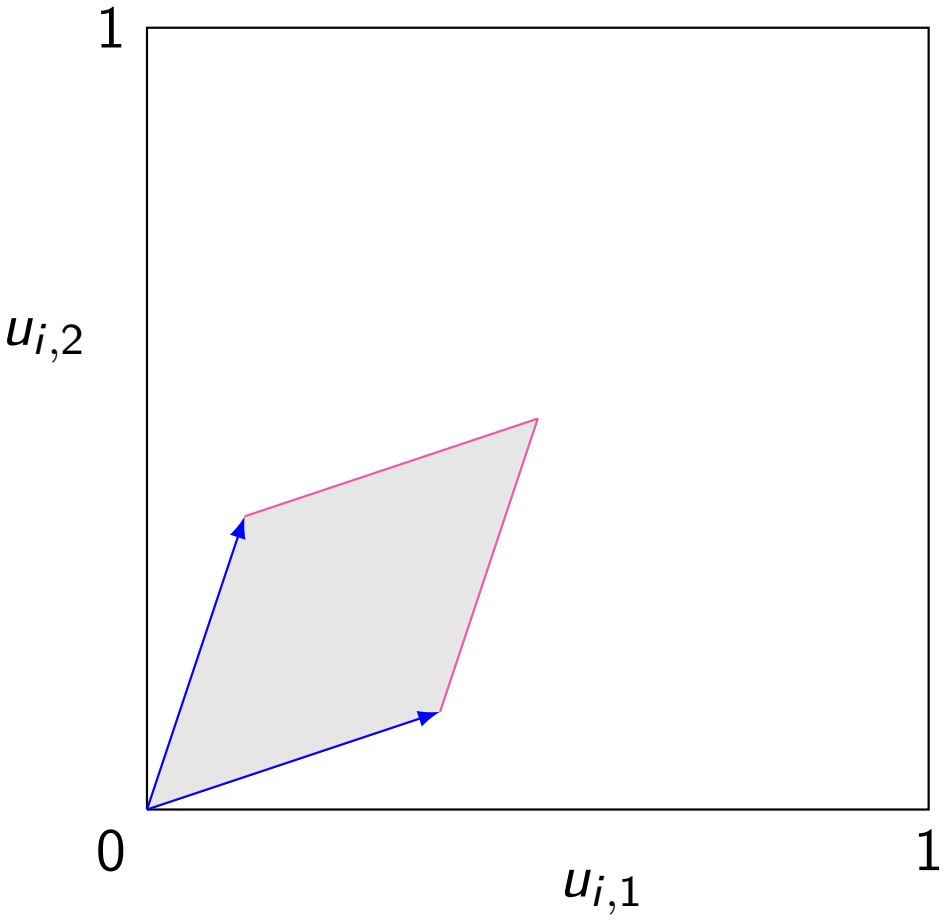


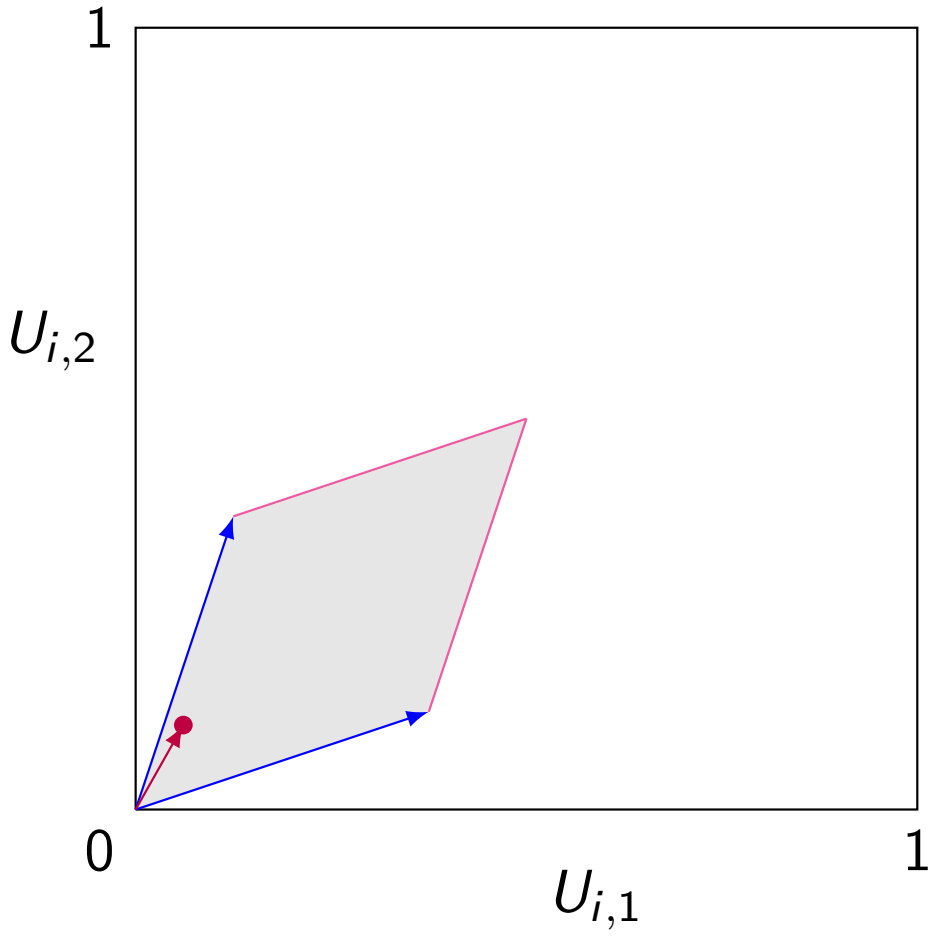


Can generate the shift uniformly in the parallelotope determined by basis vectors,

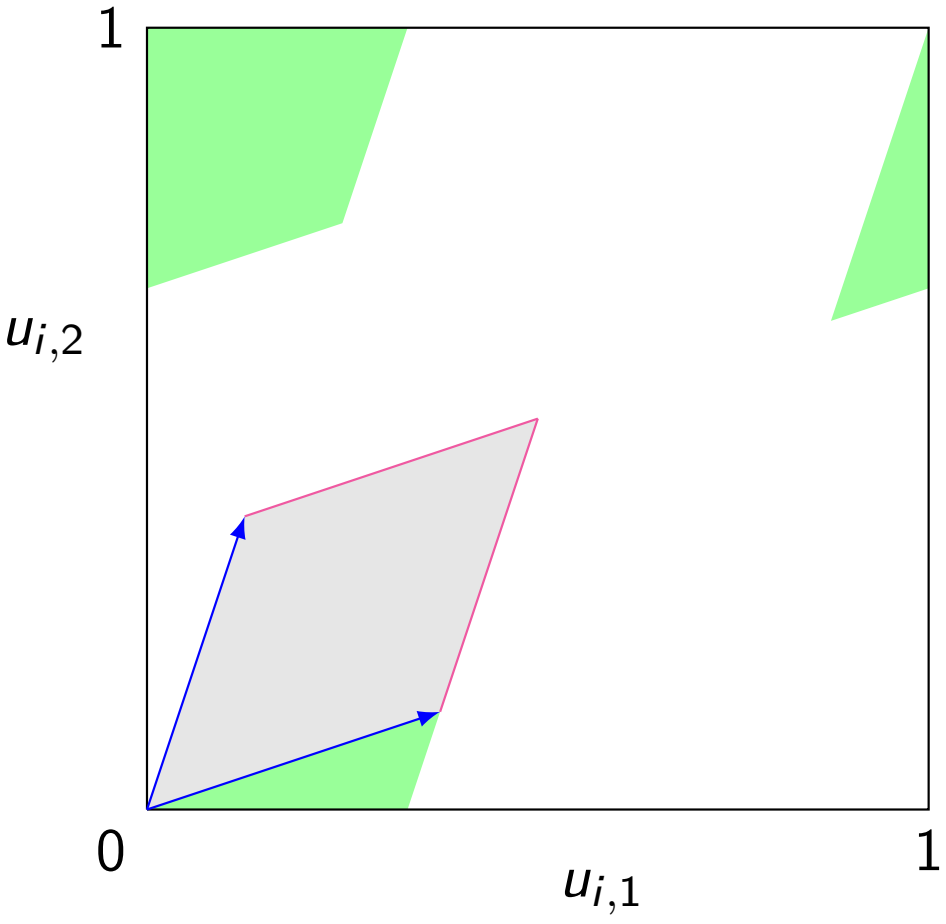


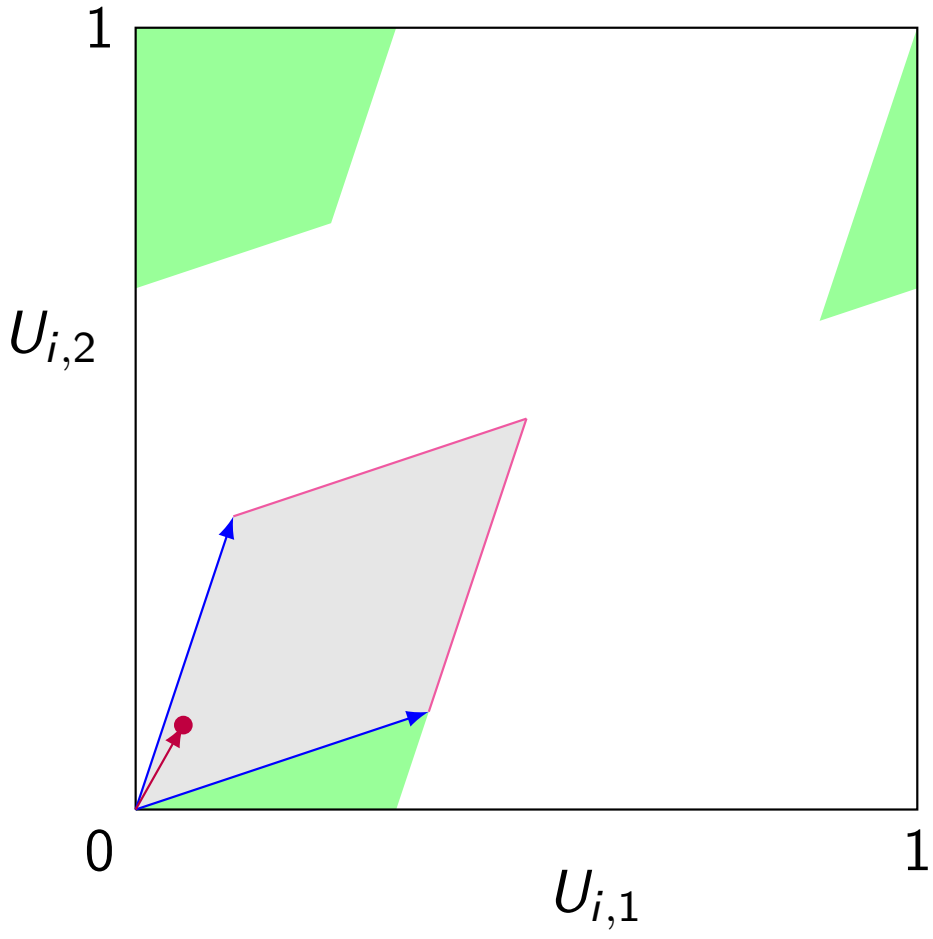
Can generate the shift uniformly in the parallelotope determined by basis vectors,



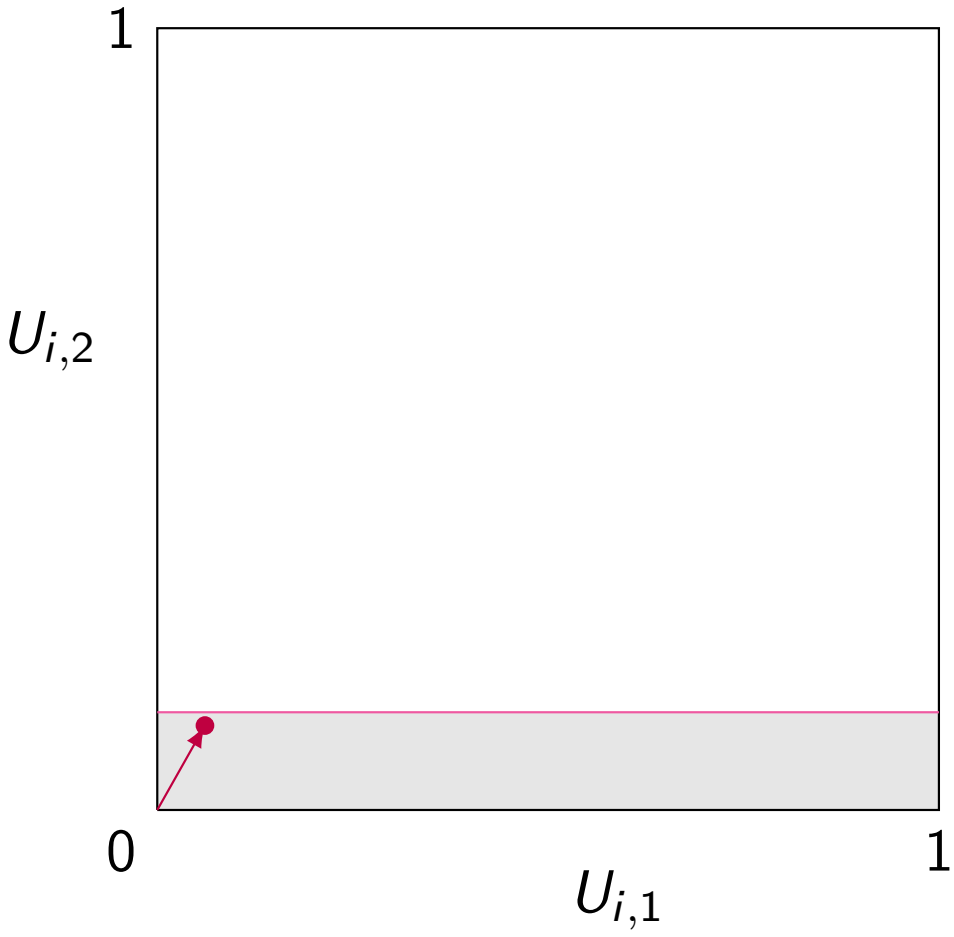


Can generate the shift uniformly in the parallelotope determined by basis vectors, or in any shifted copy of it.

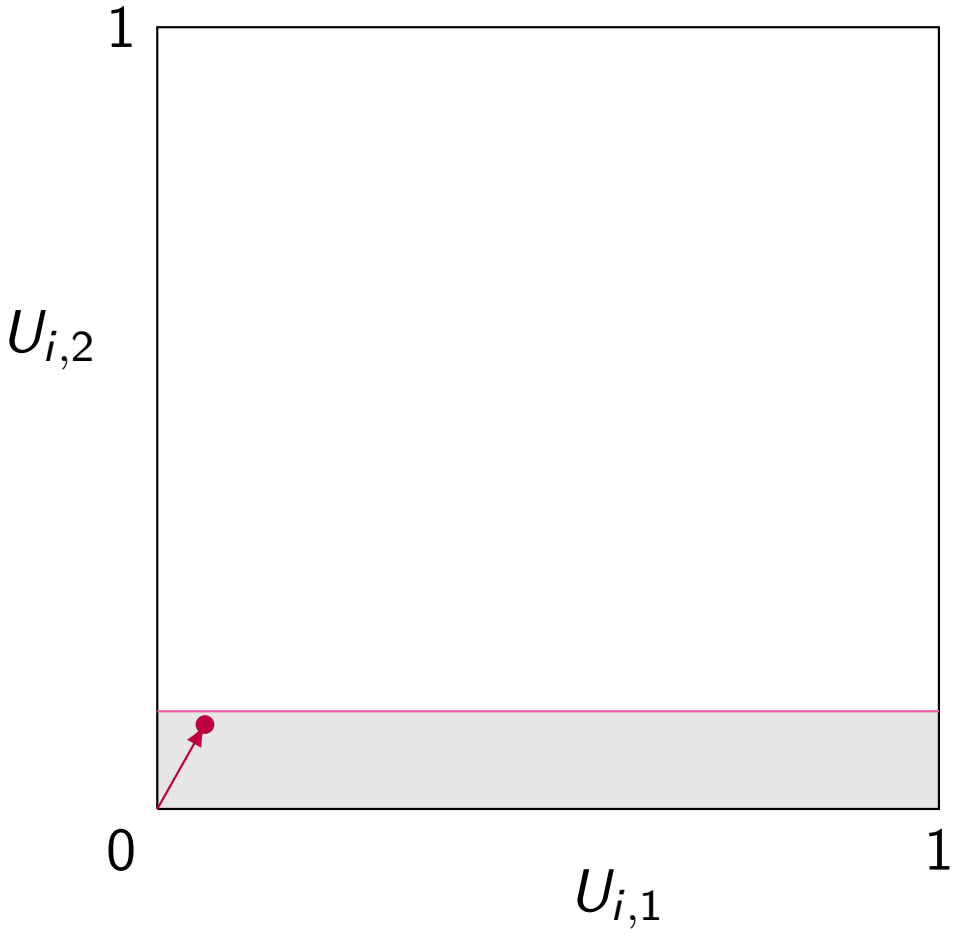


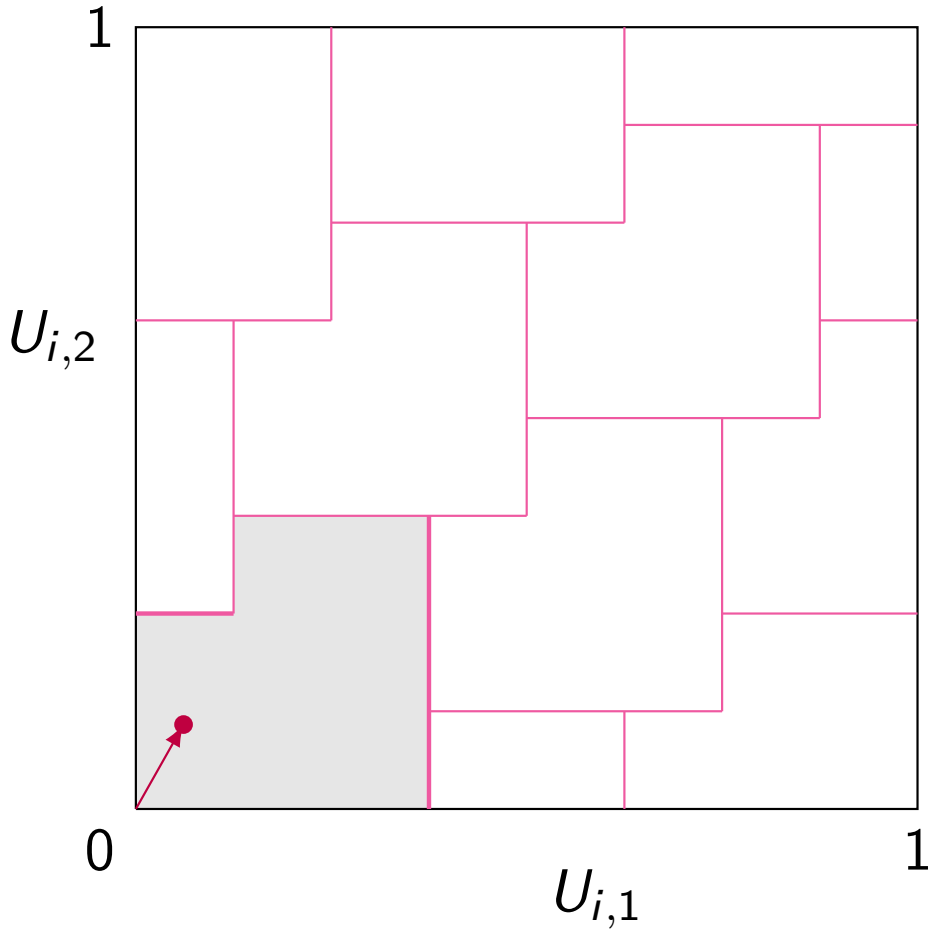


Perhaps less obvious: Can generate it in any of the colored shapes below.

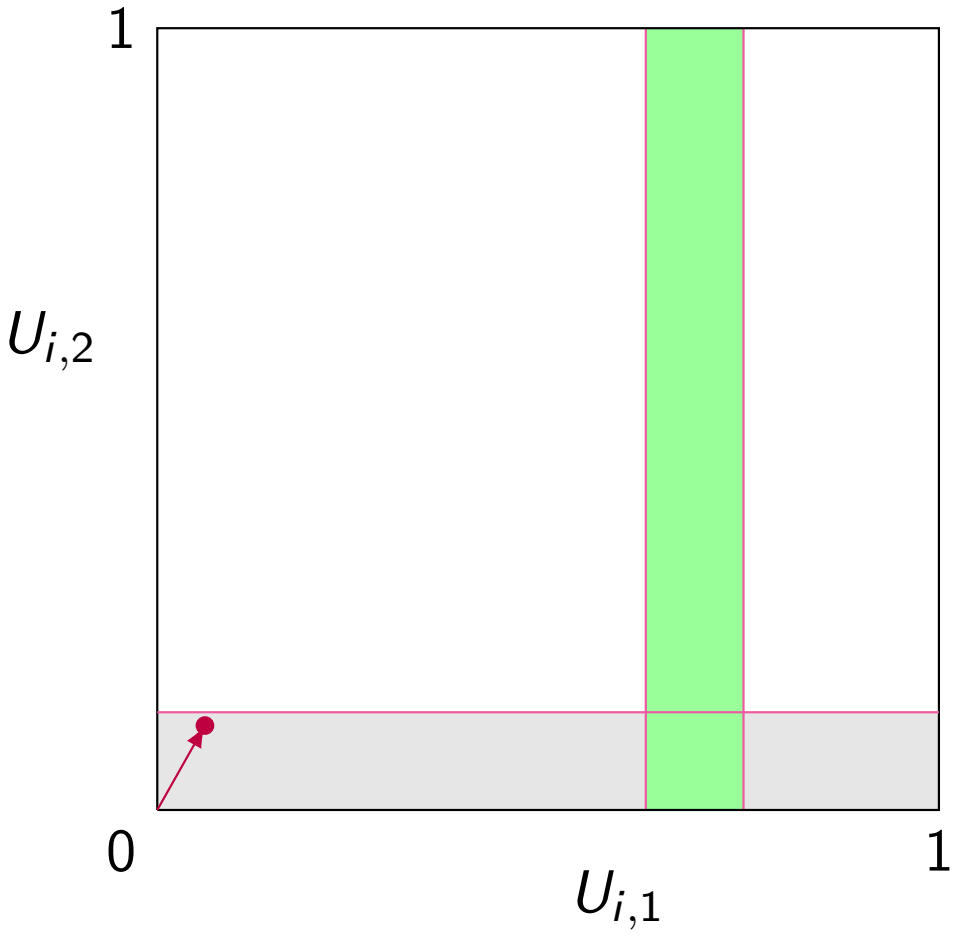


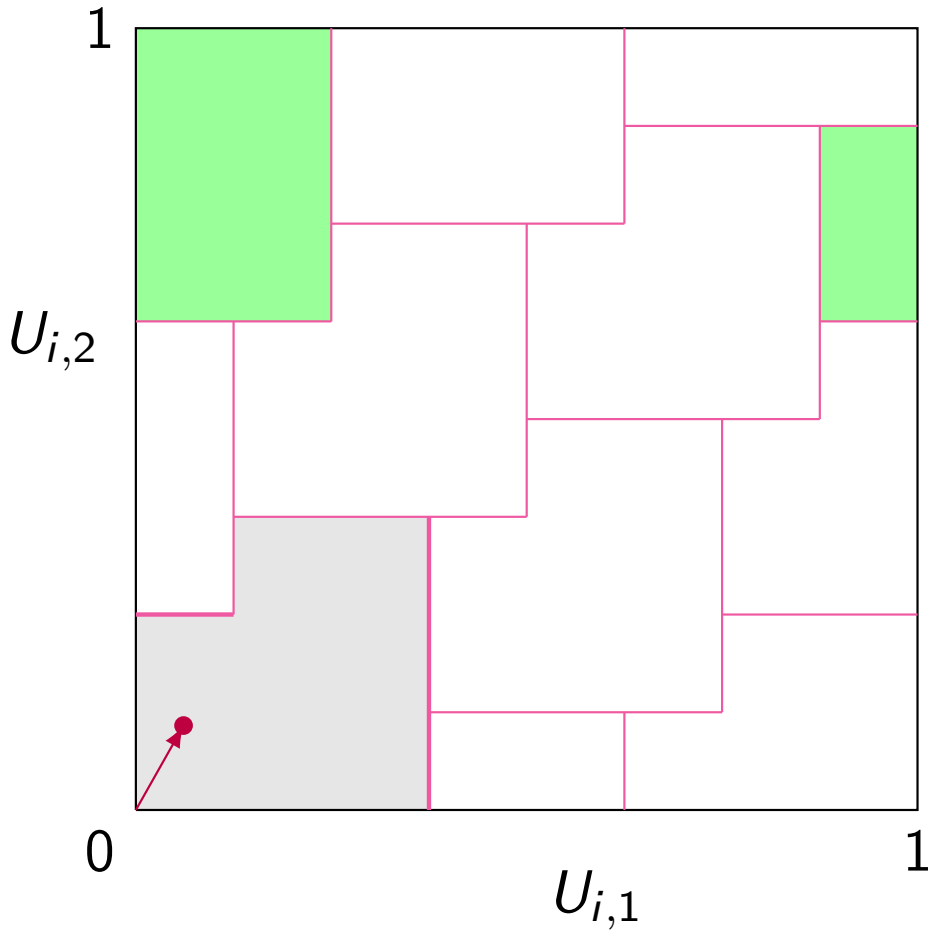
Perhaps less obvious: Can generate it in any of the colored shapes below.





Perhaps less obvious: Can generate it in any of the colored shapes below.





Generating the shift uniformly in one tile

Proposition. Let $R \subset [0,1)^s$ such that

$$\{R_i = (R + \mathbf{u}_i) \bmod 1,\ i = 0, \ldots, n-1\}$$

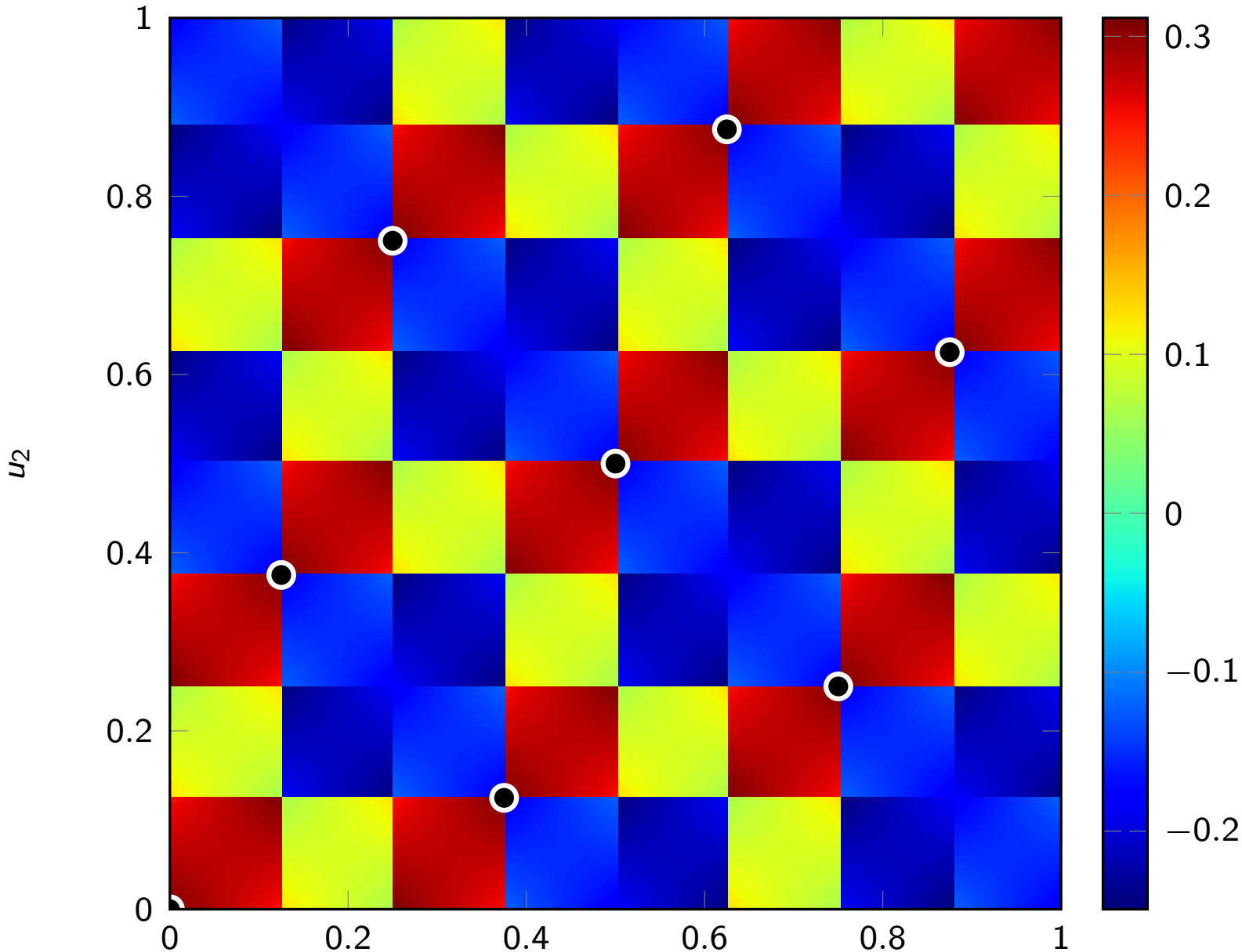
is a partition of $[0,1)^s$ in n regions of volume $1/n$.
Then, sampling the random shift $\mathbf{U}$ uniformly in any given R_i is equivalent to sampling it uniformly in $[0,1)^s$.

The error function

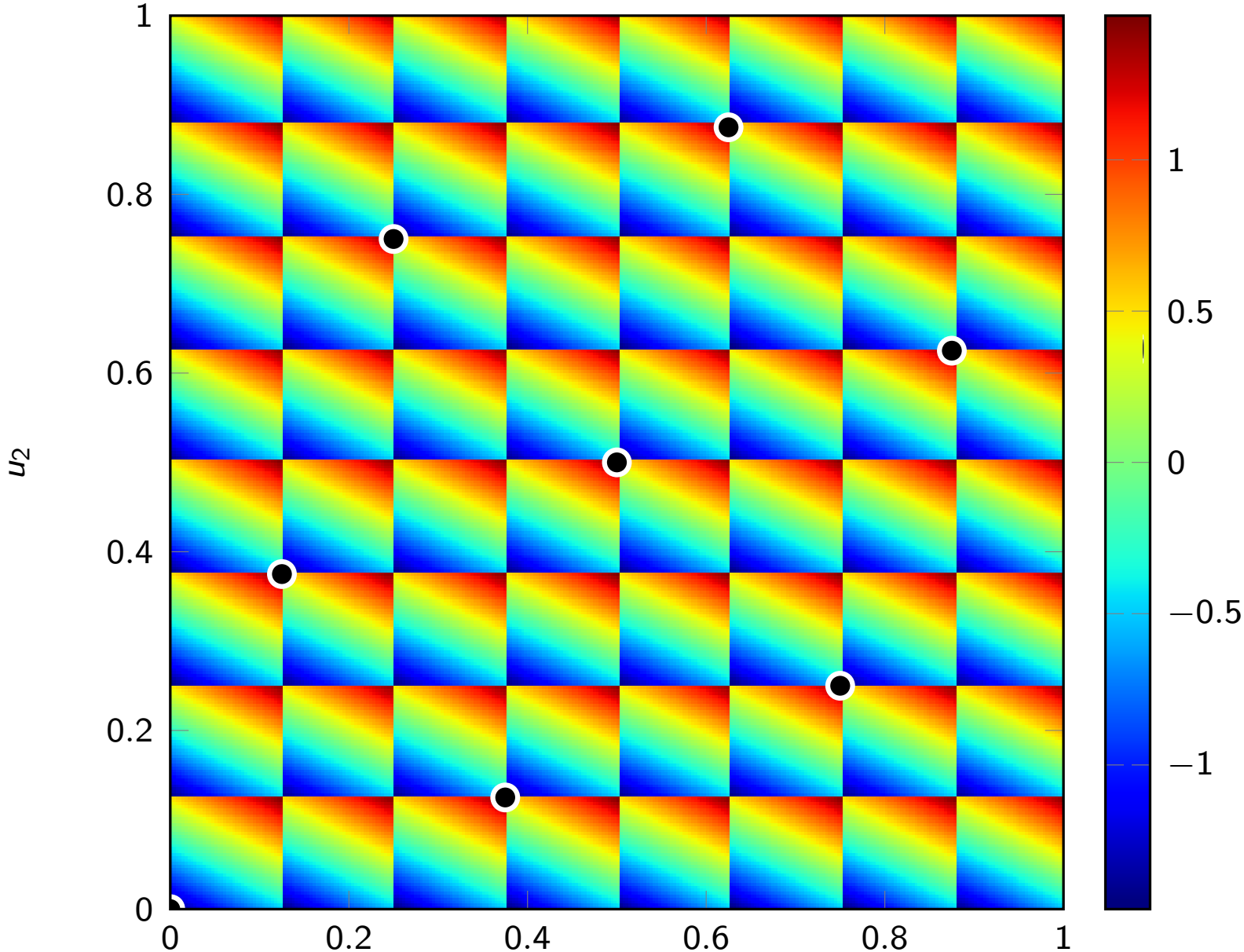
$$g_n(\mathbf{U}) = \hat{\mu}_{n,\mathrm{rqmc}} - \mu$$

over any R_i is the same as over R.

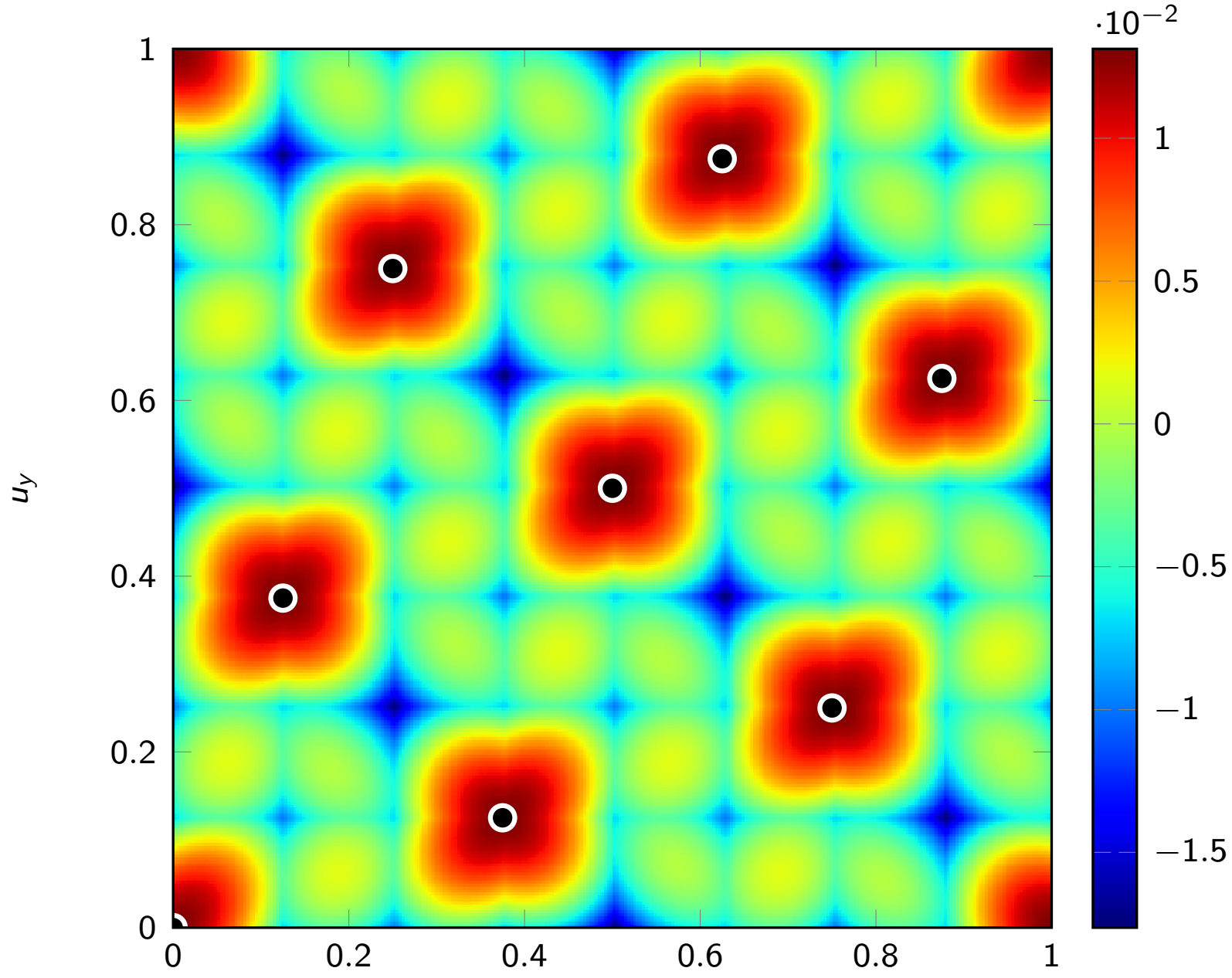
Error function $g_n(\mathbf{u})$ for $f(u_1, u_2) = (u_1 - 1/2)(u_2 - 1/2)$.



Error function $g_n(\mathbf{u})$ for $f(u_1, u_2) = (u_1 - 1/2) + (u_2 - 1/2)$.



Error function $g_n(\mathbf{u})$ for $f(u_1, u_2) = u_1 u_2 (u_1 - 1/2)(u_2 - 1/2)$.



Variance expression

Suppose f has Fourier expansion

$$f(\mathbf{u}) = \sum_{\mathbf{h}\in\mathbb{Z}^s} \hat{f}(\mathbf{h}) e^{2\pi\sqrt{-1}\mathbf{h}^t\mathbf{u}}.$$

For a randomly shifted lattice, the exact variance is (always)

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} |\hat{f}(\mathbf{h})|^2,$$

where $L_s^* = \{\mathbf{h} \in \mathbb{R}^s : \mathbf{h}^t\mathbf{v} \in \mathbb{Z} \text{ for all } \mathbf{v} \in L_s\} \subseteq \mathbb{Z}^s$ is the dual lattice.

From the viewpoint of variance reduction, an optimal lattice for f minimizes the square "discrepancy" $D^2(P_n) = \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}]$.

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} |\hat{f}(\mathbf{h})|^2.$$

If f has square-integrable mixed partial derivatives up to order α, and the periodic continuation of its derivatives up to order $\alpha - 1$ is continuous across the unit cube boundaries, then

$$|\hat{f}(\mathbf{h})|^2 = \mathcal{O}((\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha}).$$

Moreover, there is a $\mathbf{v}_1 = \mathbf{v}_1(n)$ such that

$$\mathcal{P}_{2\alpha} \stackrel{\mathrm{def}}{=} \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha} = \mathcal{O}(n^{-2\alpha+\delta}).$$

This is the variance for a worst-case f having

$$|\hat{f}(\mathbf{h})|^2 = (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha}.$$

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} |\hat{f}(\mathbf{h})|^2.$$

If f has square-integrable mixed partial derivatives up to order α, and the periodic continuation of its derivatives up to order $\alpha - 1$ is continuous across the unit cube boundaries, then

$$|\hat{f}(\mathbf{h})|^2 = \mathcal{O}((\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha}).$$

Moreover, there is a $\mathbf{v}_1 = \mathbf{v}_1(n)$ such that

$$\mathcal{P}_{2\alpha} \stackrel{\mathrm{def}}{=} \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha} = \mathcal{O}(n^{-2\alpha+\delta}).$$

This is the variance for a worst-case f having

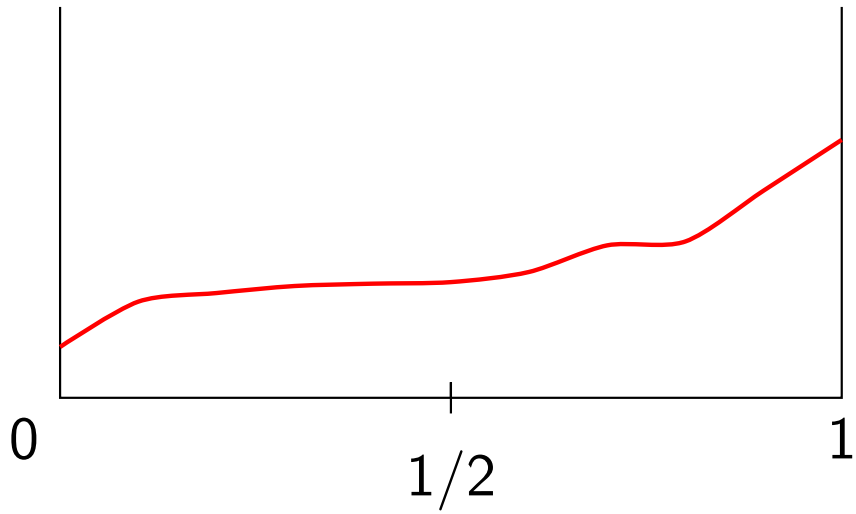
$$|\hat{f}(\mathbf{h})|^2 = (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha}.$$

Beware of hidden factor in $\mathcal{O}$ when s is large.
This worst-case function may be far from representative in applications.

Baker's transformation

Want to make the periodic continuation continuous.

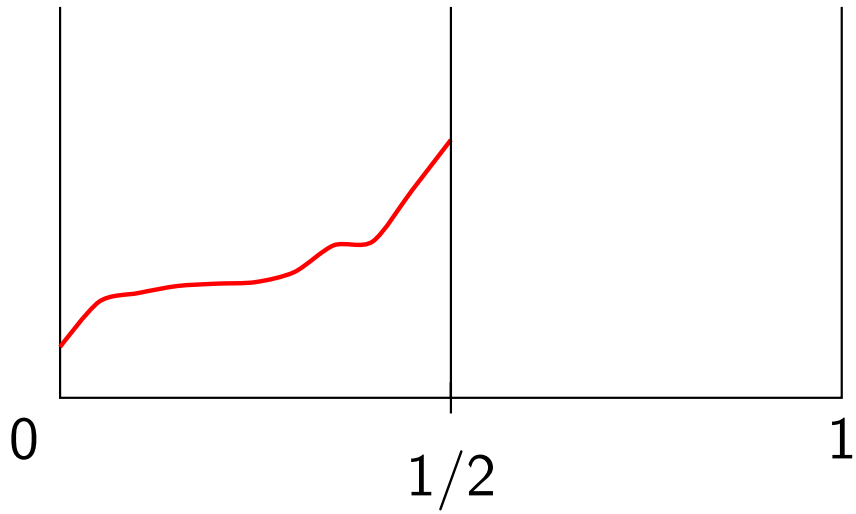
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's transformation

Want to make the periodic continuation continuous.

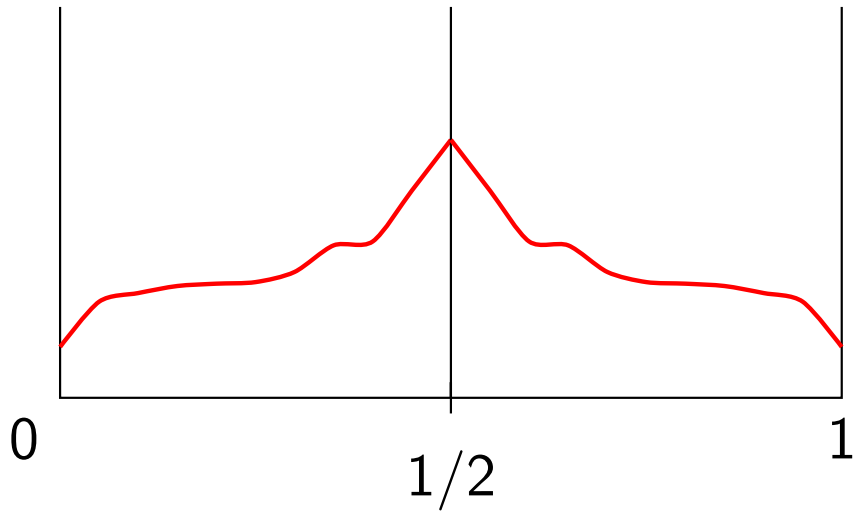
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's transformation

Want to make the periodic continuation continuous.

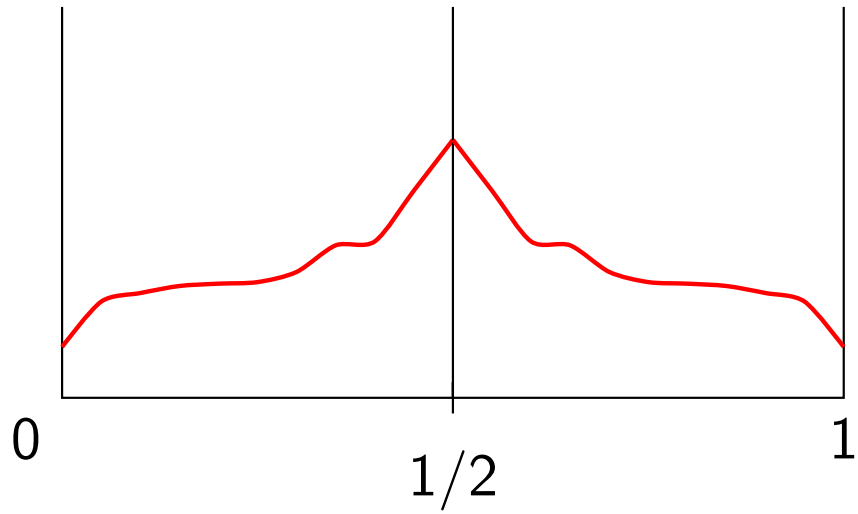
If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.



Baker's transformation

Want to make the periodic continuation continuous.

If $f(0) \neq f(1)$, define $\tilde{f}$ by $\tilde{f}(1-u) = \tilde{f}(u) = f(2u)$ for $0 \leq u \leq 1/2$.
This $\tilde{f}$ has the same integral as f and $\tilde{f}(0) = \tilde{f}(1)$.

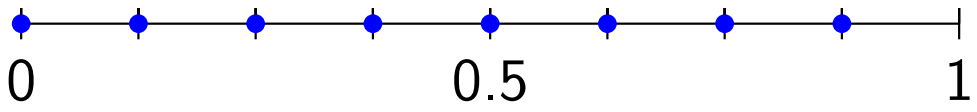


For smooth f, can reduce the variance to $O(n^{-4+\delta})$ (Hickernell 2002).
The resulting $\tilde{f}$ also symmetric with respect to $u = 1/2$.

In practice, we transform the points $\mathbf{U}_i$ instead of f.

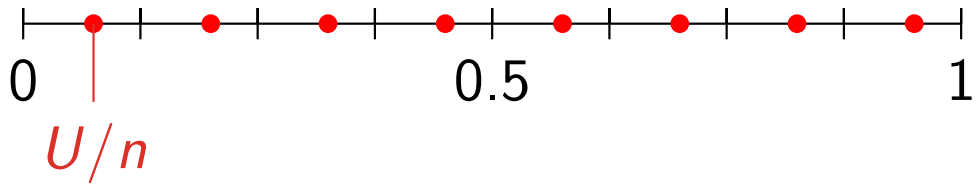
One-dimensional case

Random shift followed by baker's transformation.
Along each coordinate, stretch everything by a factor of 2 and fold.
Same as replacing U_j by $\min[2U_j, 2(1 - U_j)]$.



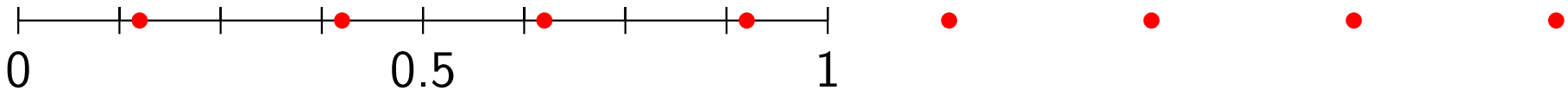
One-dimensional case

Random shift followed by baker's transformation.
Along each coordinate, stretch everything by a factor of 2 and fold.
Same as replacing U_j by $\min[2U_j, 2(1-U_j)]$.



One-dimensional case

Random shift followed by baker's transformation.
Along each coordinate, stretch everything by a factor of 2 and fold.
Same as replacing U_j by $\min[2U_j, 2(1-U_j)]$.



One-dimensional case

Random shift followed by baker's transformation.
Along each coordinate, stretch everything by a factor of 2 and fold.
Same as replacing U_j by $\min[2U_j, 2(1-U_j)]$.
Gives locally antithetic points in intervals of size $2/n$.

Searching for a lattice that minimizes

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} |\hat{f}(\mathbf{h})|^2$$

is unpractical, because:

- the Fourier coefficients are usually unknown,
- there are infinitely many,
- must do it for each f.

We nevertheless want to see how far we can go in that direction.

Searching for a lattice that minimizes

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} |\hat{f}(\mathbf{h})|^2$$

is unpractical, because:

- the Fourier coefficients are usually unknown,
- there are infinitely many,
- must do it for each f.

We nevertheless want to see how far we can go in that direction.

We start with a simple function for which we know the Fourier expansion. Even then, the discrepancy involves an infinite number of terms!

Possible ideas: Truncate the sum to a finite subset B:

$$\sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*\cap B} |\hat{f}(\mathbf{h})|^2,$$

or to the largest q square coefficients $|\hat{f}(\mathbf{h})|^2$. But hard to implement!

Dual-space exploration

The following makes sense if the $|\hat{f}(\mathbf{h})|^2$ tend to decrease with each $|h_j|$.

Start with a large set $\mathcal{L}$ of lattices (or generating vectors $\mathbf{v}_1$, for given n).

Search for vectors $\mathbf{h}$ with large weights $w(\mathbf{h}) = |\hat{f}(\mathbf{h})|^2$, via a neighborhood search starting at $\mathbf{h} = \mathbf{0}$, keeping a sorted list (as in Dijkstra's shortest path algorithm), and eliminate (successively) from $\mathcal{L}$ the lattices whose dual contains $\mathbf{h}$ for the next largest $w(\mathbf{h})$, until a single lattice remains.

Example of neighborhood $\mathcal{N}(\mathbf{h})$: only one coordinate differs, by one unit.

Dual-space exploration

The following makes sense if the $|\hat{f}(\mathbf{h})|^2$ tend to decrease with each $|h_j|$.

Start with a large set $\mathcal{L}$ of lattices (or generating vectors $\mathbf{v}_1$, for given n).

Search for vectors $\mathbf{h}$ with large weights $w(\mathbf{h}) = |\hat{f}(\mathbf{h})|^2$, via a neighborhood search starting at $\mathbf{h} = \mathbf{0}$, keeping a sorted list (as in Dijkstra's shortest path algorithm), and eliminate (successively) from $\mathcal{L}$ the lattices whose dual contains $\mathbf{h}$ for the next largest $w(\mathbf{h})$, until a single lattice remains.

Example of neighborhood $\mathcal{N}(\mathbf{h})$: only one coordinate differs, by one unit.

Component-by-component version: For $j = 1, 2, \ldots, s$, we apply the algorithm for a set $\mathcal{L}$ of j-dimensional lattices with common (fixed) $j - 1$ first coordinates, and determine the jth coordinate by visiting the j-dimensional vectors $\mathbf{h}$.

When the $|\hat{f}(\mathbf{h})|$ are unknown, we may think of estimating them as needed, dynamically.

Algorithm Dual-Space-Exploration(lattice set $\mathcal{L}$, weights w);

$\mathcal{Q} \leftarrow \mathcal{N}(\mathbf{0})$ // vectors $\mathbf{h}$ to be visited, sorted by weight $w(\mathbf{h})$;
$\mathcal{M} \leftarrow \mathcal{N}(\mathbf{0})$ // vectors $\mathbf{h}$ who already entered $\mathcal{Q}$;
while $|\mathcal{L}| > 1$ **do**
  $\mathbf{h} \leftarrow$ remove first from $\mathcal{Q}$;
  for all lattices $L_s \in \mathcal{L}$ such that $\mathbf{h} \in L_s^*$ **do**
    remove L_s from $\mathcal{L}$;
    if $|\mathcal{L}| = 1$ **then**
      return the single lattice $L_s \in \mathcal{L}$ and exit;
    end if
  end for
  for all $\mathbf{h}' \in \mathcal{N}(\mathbf{h}) \setminus \mathcal{M}$ **do**
    add $\mathbf{h}'$ to $\mathcal{M}$ and to $\mathcal{Q}$ with priority (weight) $w(\mathbf{h}')$;
  end for
end while

An example

Take the product V-shaped function

$$f(\mathbf{u}) = \prod_{j=1}^{s} \frac{|4u_j - 2| + c_j}{1 + c_j},$$

so

$$\hat{f}(\mathbf{h}) = \prod_{\{j\,:\,h_j \text{ is odd}\}} \frac{4}{(1 + c_j)\pi^2 h_j^2}.$$

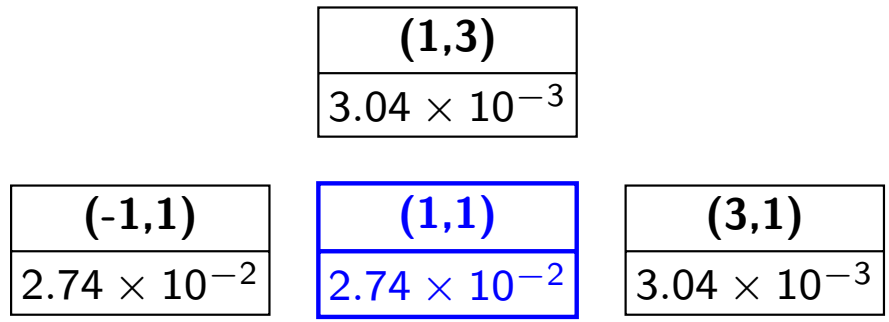
Dimensions $s = 5$ and 10.
Constants $c_j = 1, j, j^2, j^3$.

The Dual Exploration Algorithm in Action

(1,1)
2.74×10^{-2}

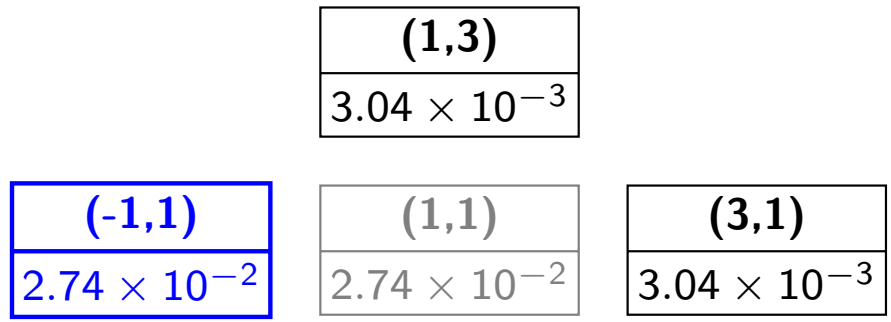
Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action



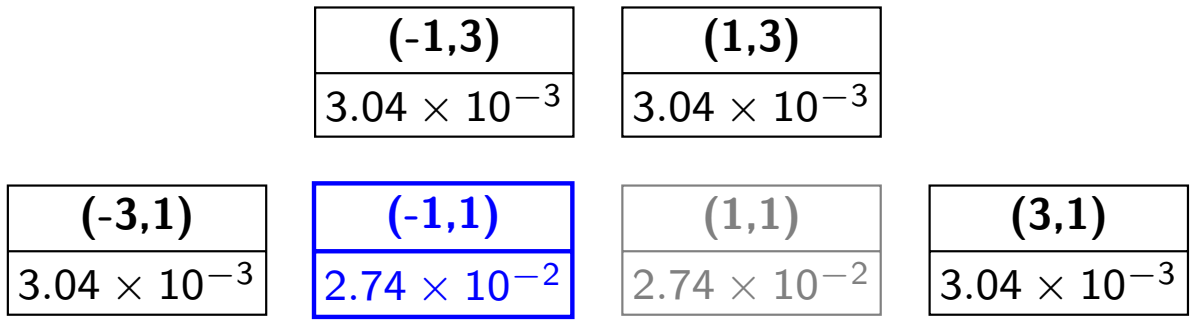
Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action



Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action



Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

	(-1,3)	**(1,3)**	
	3.04×10^{-3}	3.04×10^{-3}	
(-3,1)	(-1,1)	(1,1)	**(3,1)**
3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}

Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

	(-3,3)	**(-1,3)**	**(1,3)**		
	3.38×10^{-4}	3.04×10^{-3}	3.04×10^{-3}		
(-5,1)	**(-3,1)**	(-1,1)	(1,1)	**(3,1)**	
1.10×10^{-3}	3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}	

Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

	(-3,3)	**(-1,3)**	**(1,3)**	
	3.38×10^{-4}	3.04×10^{-3}	3.04×10^{-3}	
(-5,1)	**(-3,1)**	**(-1,1)**	**(1,1)**	**(3,1)**
1.10×10^{-3}	3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}

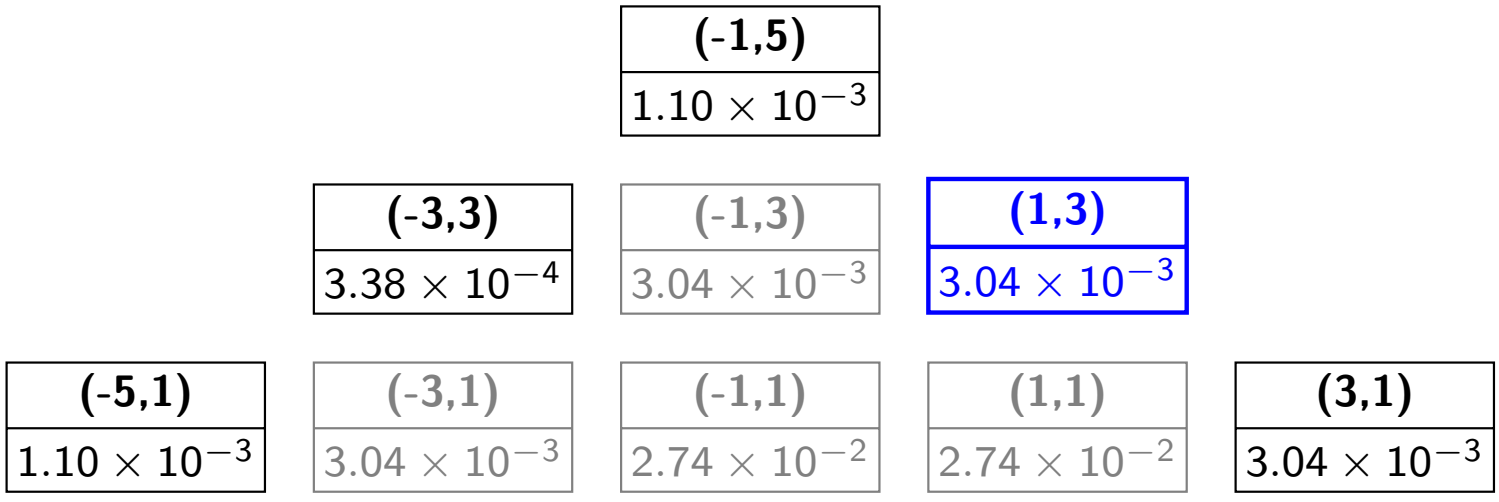
Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

		(-1,5)		
		1.10×10^{-3}		
	(-3,3)	**(-1,3)**	**(1,3)**	
	3.38×10^{-4}	3.04×10^{-3}	3.04×10^{-3}	
(-5,1)	**(-3,1)**	**(-1,1)**	**(1,1)**	**(3,1)**
1.10×10^{-3}	3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}

Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action



Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

		(-1,5) 1.10×10^{-3}	**(1,5)** 1.10×10^{-3}	
	(-3,3) 3.38×10^{-4}	(-1,3) 3.04×10^{-3}	**(1,3)** 3.04×10^{-3}	**(3,3)** 3.38×10^{-4}
(-5,1) 1.10×10^{-3}	(-3,1) 3.04×10^{-3}	(-1,1) 2.74×10^{-2}	(1,1) 2.74×10^{-2}	**(3,1)** 3.04×10^{-3}

Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

		(-1,5) 1.10×10^{-3}	**(1,5)** 1.10×10^{-3}	
	(-3,3) 3.38×10^{-4}	(-1,3) 3.04×10^{-3}	(1,3) 3.04×10^{-3}	**(3,3)** 3.38×10^{-4}
(-5,1) 1.10×10^{-3}	(-3,1) 3.04×10^{-3}	(-1,1) 2.74×10^{-2}	(1,1) 2.74×10^{-2}	**(3,1)** 3.04×10^{-3}

Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

		(-1,5)	**(1,5)**		
		1.10×10^{-3}	1.10×10^{-3}		
	(-3,3)	(-1,3)	(1,3)	**(3,3)**	
	3.38×10^{-4}	3.04×10^{-3}	3.04×10^{-3}	3.38×10^{-4}	
(-5,1)	(-3,1)	(-1,1)	(1,1)	**(3,1)**	**(5,1)**
1.10×10^{-3}	3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}	1.10×10^{-3}

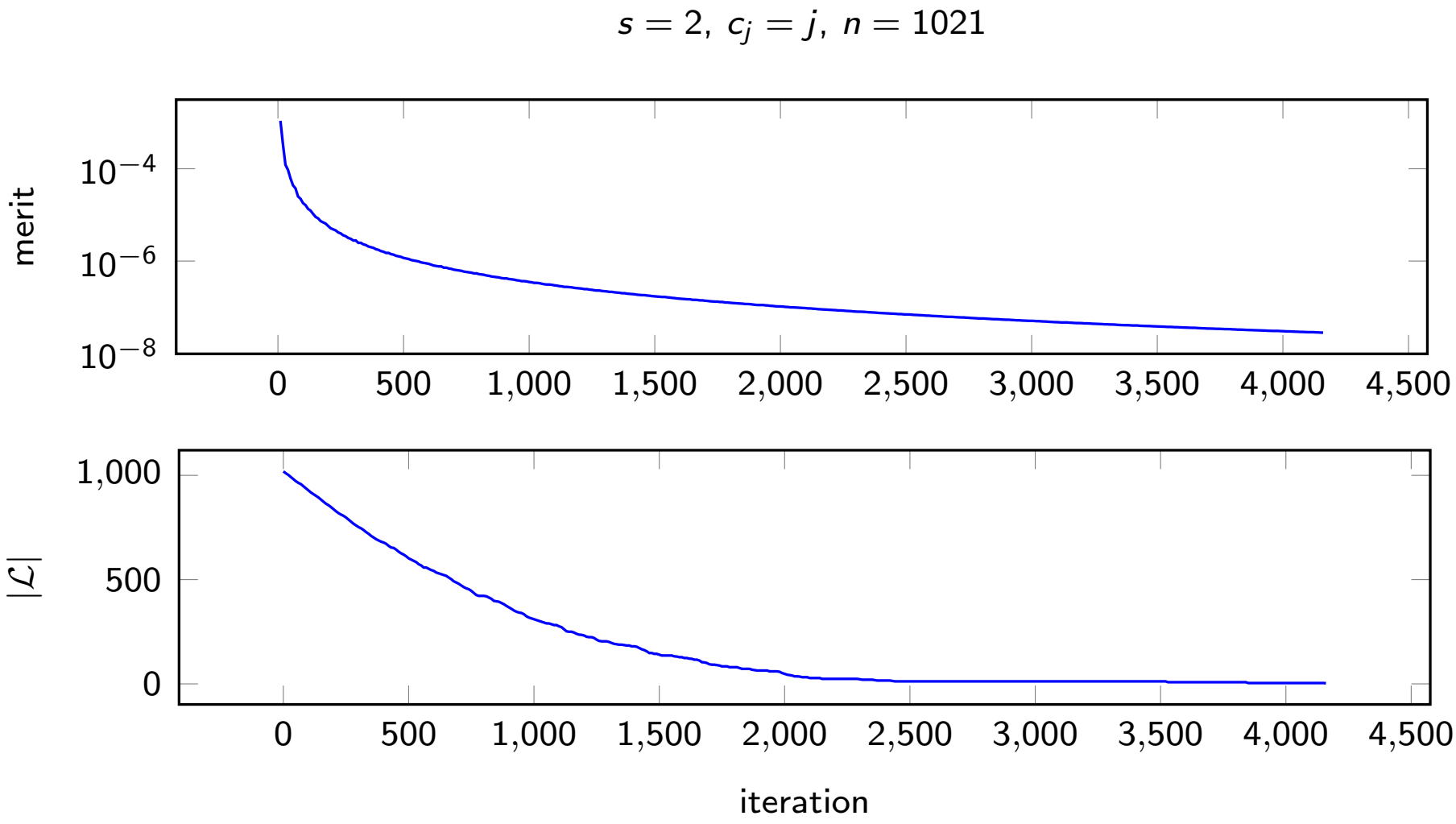
Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

The Dual Exploration Algorithm in Action

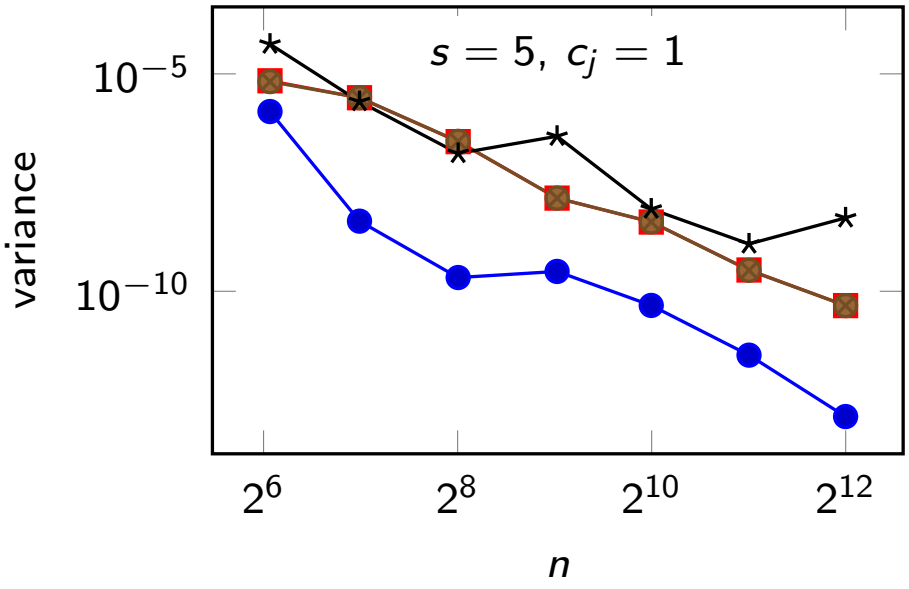
		(-1,5)	**(1,5)**			
		1.10×10^{-3}	1.10×10^{-3}			
	(-3,3)	(-1,3)	(1,3)	**(3,3)**		
	3.38×10^{-4}	3.04×10^{-3}	3.04×10^{-3}	3.38×10^{-4}		
(-5,1)	(-3,1)	(-1,1)	(1,1)	(3,1)	**(5,1)**	
1.10×10^{-3}	3.04×10^{-3}	2.74×10^{-2}	2.74×10^{-2}	3.04×10^{-3}	1.10×10^{-3}	

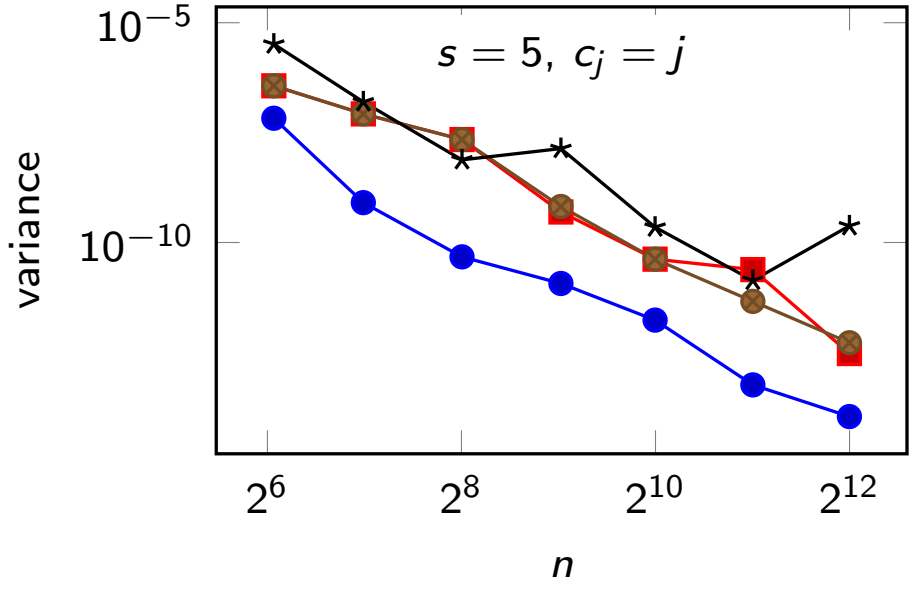
Colors: [in $\mathcal{Q}$] [in $\mathcal{M} \setminus \mathcal{Q}$] [visiting]

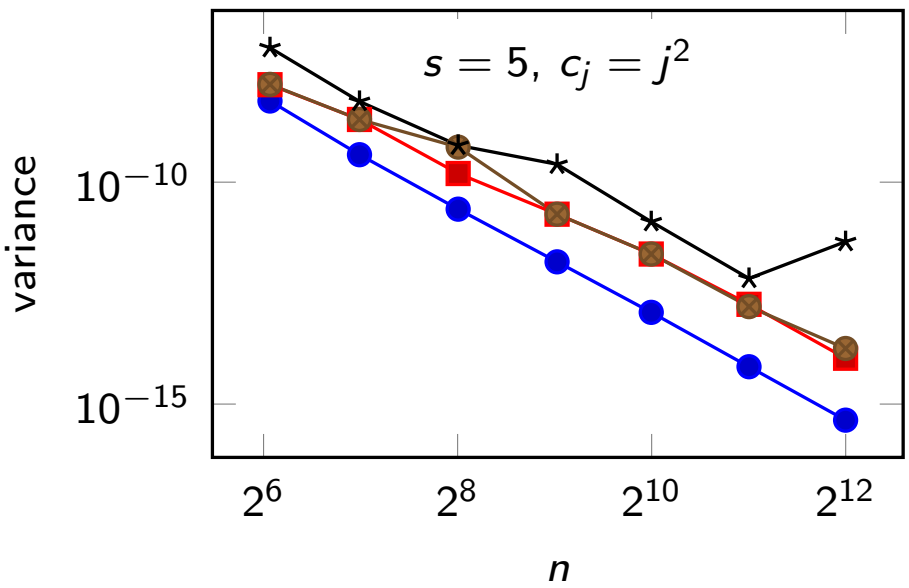
Evolution of the Dual Exploration Algorithm

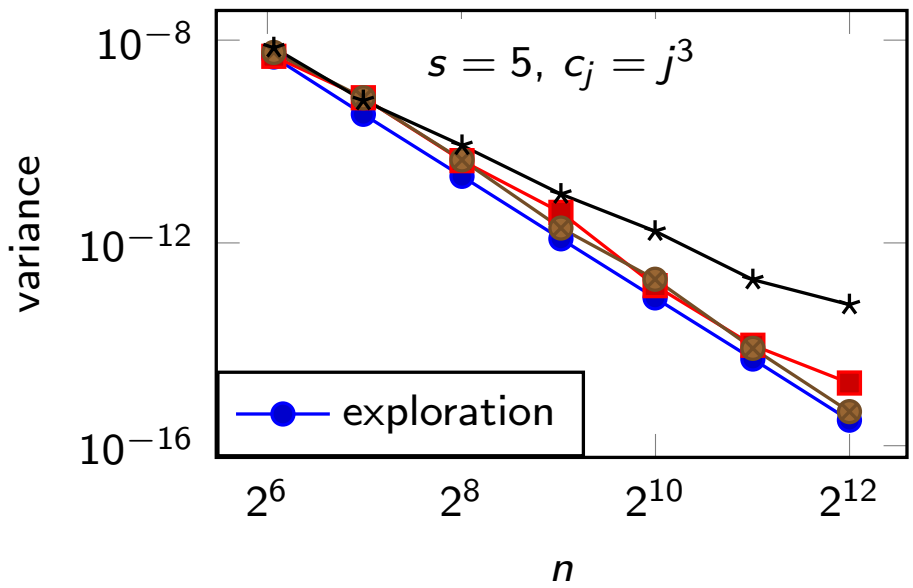


Estimated variance vs n for $s = 5$

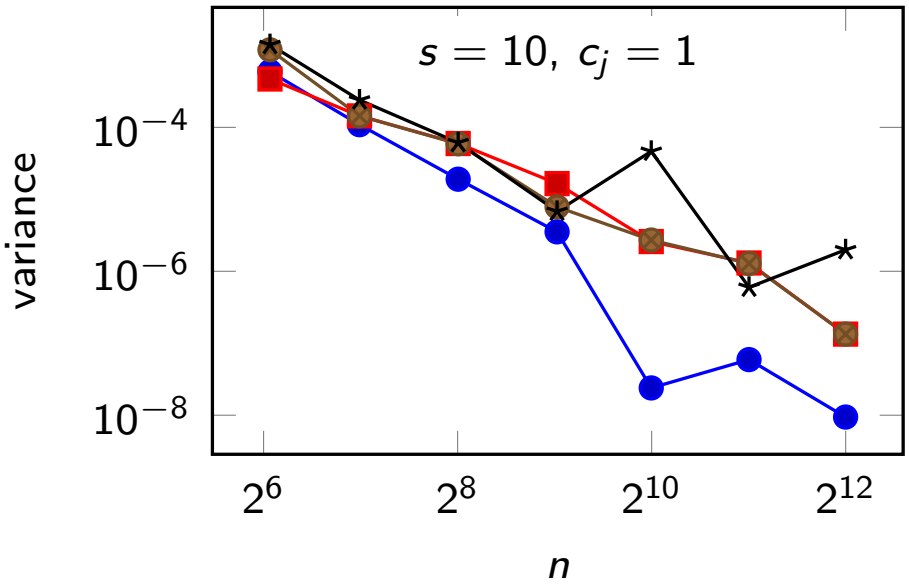


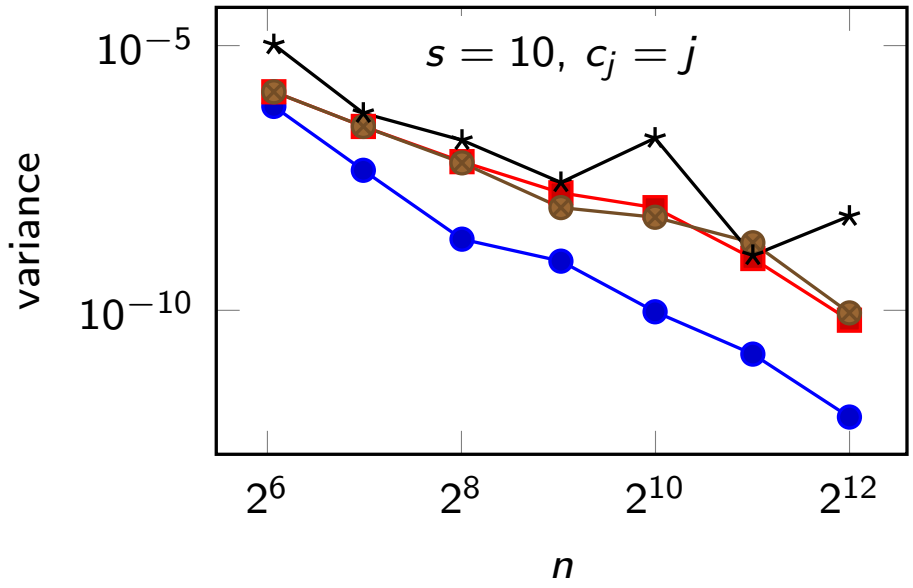


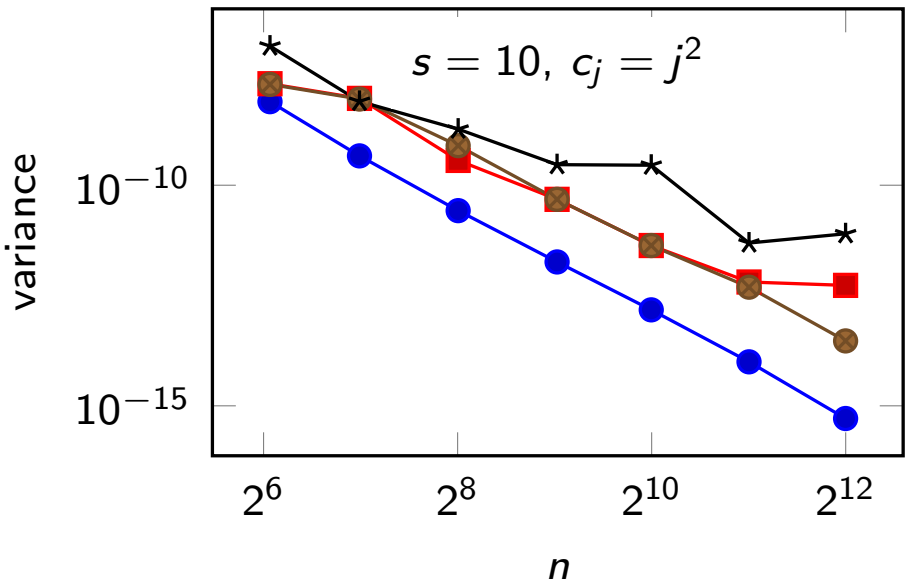


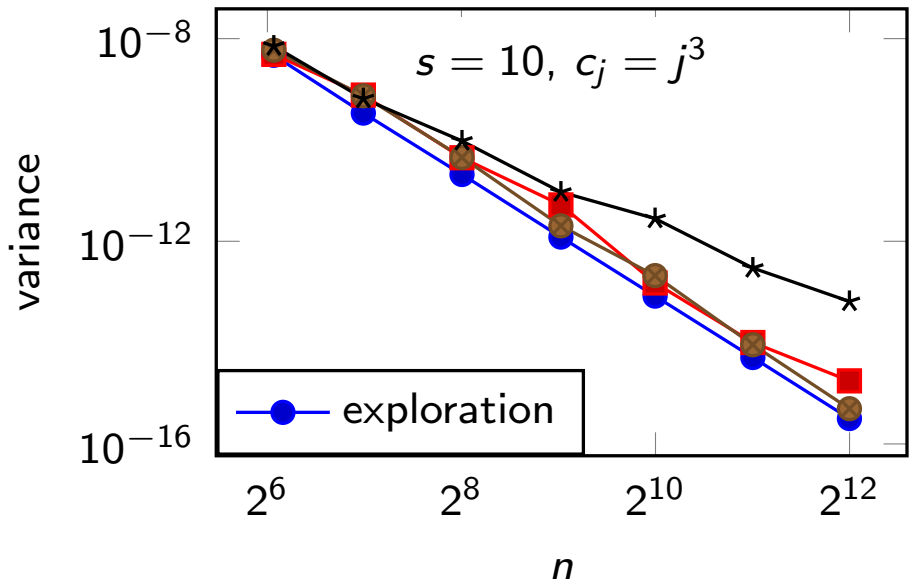


Estimated variance vs n for $s = 10$









ANOVA decomposition

The Fourier expansion has too many terms to handle. As a cruder expansion, we can write $f(\mathbf{u}) = f(u_1, \ldots, u_s)$ as:

$$f(\mathbf{u}) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} f_{\mathfrak{u}}(\mathbf{u}) = \mu + \sum_{i=1}^{s} f_{\{i\}}(u_i) + \sum_{i,j=1}^{s} f_{\{i,j\}}(u_i, u_j) + \cdots$$

where

$$f_{\mathfrak{u}}(\mathbf{u}) = \int_{[0,1)^{|\bar{\mathfrak{u}}|}} f(\mathbf{u})\, \mathrm{d}\mathbf{u}_{\bar{\mathfrak{u}}} - \sum_{\mathfrak{v} \subset \mathfrak{u}} f_{\mathfrak{v}}(\mathbf{u}_{\mathfrak{v}}),$$

and the Monte Carlo variance decomposes as

$$\sigma^2 = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \sigma_{\mathfrak{u}}^2, \qquad \text{where } \sigma_{\mathfrak{u}}^2 = \mathrm{Var}[f_{\mathfrak{u}}(\mathbf{U})].$$

Sensitivity indices: $S_{\mathfrak{u}} = \sigma_{\mathfrak{u}}^2/\sigma^2$. Can be estimated by MC or RQMC.

Heuristic intuition: Make sure the projections of P_n are very uniform for the important subsets $\mathfrak{u}$ (i.e., with large $S_{\mathfrak{u}}$).

Shift-invariant discrepancy

In a reproducing kernel Hilbert space (RKHS) with kernel K, and randomly-shifted points, the relevant discrepancy corresponds to the shift-invariant kernel

$$K_{\mathrm{sh}}(\mathbf{u}_i, \mathbf{u}_j) := \mathbb{E}[K(\mathbf{U}_i, \mathbf{U}_j)] = \mathbb{E}[K(\mathbf{u}_i + \mathbf{U}, \mathbf{u}_j + \mathbf{U})] = \mathbb{E}[K(\mathbf{u}_i - \mathbf{u}_j + \mathbf{U}, \mathbf{U})].$$

The mean square discrepancy can be written as

$$\mathbb{E}[D^2(P_n)] = \frac{1}{n} \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{Z}^s} w(\mathbf{h}) \sum_{i=0}^{n-1} e^{2\pi\sqrt{-1}\mathbf{h}^{\mathrm{t}}\mathbf{u}_i} = \sum_{\mathbf{0} \neq \mathbf{h} \in L_*^s} w(\mathbf{h})$$

(for a lattice).

Key issue: choice of the weights $w(\mathbf{h})$.

Regrouping by projections: projection weights

Denote $\mathfrak{u}(\mathbf{h}) = \mathfrak{u}(h_1, \ldots, h_s)$ the set of indices j for which $h_j \neq 0$. We have

$$\mathbb{E}[D^2(P_n)] = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \sum_{\mathbf{h}:\mathfrak{u}(\mathbf{h})=\mathfrak{u}} w(\mathbf{h}) \frac{1}{n} \sum_{i=0}^{n-1} e^{2\pi\sqrt{-1}\mathbf{h}^t\mathbf{u}_i} = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} D^2_{\mathfrak{u}}(P_n).$$

The RKHS decomposes as a direct sum and the RQMC variance has a corresponding decomposition.

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}(f_{\mathfrak{u}})].$$

Restriction on the weights: take $w(\mathbf{h}) = \gamma_{\mathfrak{u}(\mathbf{h})} \prod_{j \in \mathfrak{u}} h_j^{-2\alpha}$ for all $\mathbf{h}$.
Those projection weights are the so-called general weights.

Regrouping by projections: projection weights

Denote $\mathfrak{u}(\mathbf{h}) = \mathfrak{u}(h_1, \ldots, h_s)$ the set of indices j for which $h_j \neq 0$. We have

$$\mathbb{E}[D^2(P_n)] = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \sum_{\mathbf{h}:\mathfrak{u}(\mathbf{h})=\mathfrak{u}} w(\mathbf{h}) \frac{1}{n} \sum_{i=0}^{n-1} e^{2\pi\sqrt{-1}\mathbf{h}^t\mathbf{u}_i} = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} D^2_{\mathfrak{u}}(P_n).$$

The RKHS decomposes as a direct sum and the RQMC variance has a corresponding decomposition.

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}(f_{\mathfrak{u}})].$$

Restriction on the weights: take $w(\mathbf{h}) = \gamma_{\mathfrak{u}(\mathbf{h})} \prod_{j \in \mathfrak{u}} h_j^{-2\alpha}$ for all $\mathbf{h}$.
Those projection weights are the so-called general weights.
Anyhow, how should we choose them?

Example: a weighted Sobolev space

Space of functions with integrable partial derivatives. RKHS with kernel

$$K(\mathbf{u},\mathbf{x}) = \sum_{u\subseteq\{1,\ldots,s\}} \gamma_u \prod_{j\in u} 2\pi^2 \left[B_2((u_j - x_j) \bmod 1)/2 + (u_j - 0.5)(x_j - 0.5)\right]$$

where $B_2(u) = u^2 - u + 1/6$. The shift-invariant kernel is

$$K_{\mathrm{sh}}(\mathbf{u},\mathbf{x}) = \sum_{u\subseteq\{1,\ldots,s\}} \gamma_u \prod_{j\in u} 2\pi^2 B_2((u_j - x_j) \bmod 1)$$

and the corresponding mean square discrepancy for a randomly-shifted lattice rule with $\mathbf{v}_1 = (v_1,\ldots,v_s) = \mathbf{z}/n$ is

$$\mathbb{E}[D^2(P_n)] = \frac{1}{n}\sum_{i=1}^{n} \sum_{u\subseteq\{1,\ldots,s\}} \gamma_u \prod_{j\in u} 2\pi^2 B_2((i\,v_j) \bmod 1).$$

From the Fourier expansion of B_2, we also have

$$\begin{aligned}\mathbb{E}[D^2(P_n)] &= \frac{1}{n}\sum_{i=1}^{n}\sum_{u\subseteq\{1,\ldots,s\}}\gamma_u\prod_{j\in u}\sum_{h_j\neq 0} h_j^{-2}e^{2\pi\sqrt{-1}ih_jv_j}\\ &= \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*}\gamma_{u(\mathbf{h})}\prod_{j\in u(\mathbf{h})}h_j^{-2} = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} w(\mathbf{h}).\end{aligned}$$

For those weights, we have $w(\mathbf{h}) = |\hat{f}(\mathbf{h})|^2$ for the function

$$f(\mathbf{u}) = \sum_{u\subseteq\{1,\ldots,s\}}(2\pi)^{|u|}\gamma_u^{1/2}\prod_{j\in u}(u_j - 0.5),$$

so $\mathbb{E}[D^2(P_n)]$ is the RQMC variance for this f.

On the other hand, the ANOVA variance components for this f are

$$\sigma_u^2 = (4\pi^2)^{|u|}\gamma_u\prod_{j\in u}\mathrm{Var}[U-0.5] = (4\pi^2/12)^{|u|}\gamma_u = (\pi^2/3)^{|u|}\gamma_u.$$

The optimal weights for this f are then

$$\gamma_u = (3/\pi^2)^{|u|}\sigma_u^2 \approx (0.30396)^{|u|}\sigma_u^2.$$

Using the same kernel and a different heuristic argument, Wang and Sloan (2006) come up with weights that generalize to (they do this for product weights only):

$$\gamma_{\mathfrak{u}}^2 = (45/\pi^4)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}}^2,$$

that is,

$$\gamma_{\mathfrak{u}} = (\sqrt{45}/\pi^2)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}} \approx (0.6797)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}},$$

Using the same kernel and a different heuristic argument, Wang and Sloan (2006) come up with weights that generalize to (they do this for product weights only):

$$\gamma_{\mathfrak{u}}^2 = (45/\pi^4)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}}^2,$$

that is,

$$\gamma_{\mathfrak{u}} = (\sqrt{45}/\pi^2)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}} \approx (0.6797)^{|\mathfrak{u}|}\sigma_{\mathfrak{u}},$$

With $\gamma_{\mathfrak{u}} = 1$, we obtain the classical (unweighted) $\mathcal{P}_2$.

Weighted $\mathcal{P}_{2\alpha}$:

$$\mathcal{P}_{2\alpha} = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} \gamma_{\mathfrak{u}(\mathbf{h})}(\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha}$$

Variance for a worst-case function whose square Fourier coefficients are

$$|\hat{f}(\mathbf{h})|^2 = \gamma_{\mathfrak{u}(\mathbf{h})}(\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha}.$$

This is the RQMC variance for the function

$$f(\mathbf{u}) = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \sqrt{\gamma_{\mathfrak{u}}} \prod_{j\in\mathfrak{u}} \frac{(2\pi)^\alpha}{\alpha!} B_\alpha(u_j).$$

We also have for this f:

$$\sigma_{\mathfrak{u}}^2 = \gamma_{\mathfrak{u}}\left[\mathrm{Var}[B_\alpha(U)]\frac{(4\pi^2)^\alpha}{(\alpha!)^2}\right]^{|\mathfrak{u}|} = \gamma_{\mathfrak{u}}\left[|B_{2\alpha}(0)|\frac{(4\pi^2)^\alpha}{(2\alpha)!}\right]^{|\mathfrak{u}|}.$$

Weighted $\mathcal{P}_{2\alpha}$:

$$\mathcal{P}_{2\alpha} = \sum_{\mathbf{0}\neq\mathbf{h}\in L_s^*} \gamma_{u(\mathbf{h})}(\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha}$$

Variance for a worst-case function whose square Fourier coefficients are

$$|\hat{f}(\mathbf{h})|^2 = \gamma_{u(\mathbf{h})}(\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha}.$$

This is the RQMC variance for the function

$$f(\mathbf{u}) = \sum_{u\subseteq\{1,\ldots,s\}} \sqrt{\gamma_u}\prod_{j\in u}\frac{(2\pi)^\alpha}{\alpha!}B_\alpha(u_j).$$

We also have for this f:

$$\sigma_u^2 = \gamma_u\left[\mathrm{Var}[B_\alpha(U)]\frac{(4\pi^2)^\alpha}{(\alpha!)^2}\right]^{|u|} = \gamma_u\left[|B_{2\alpha}(0)|\frac{(4\pi^2)^\alpha}{(2\alpha)!}\right]^{|u|}.$$

For $\alpha = 1$, we should take $\gamma_u = (3/\pi^2)^{|u|}\sigma_u^2 \approx (0.30396)^{|u|}\sigma_u^2$.
For $\alpha = 2$, we should take $\gamma_u = [45/\pi^4]^{|u|}\sigma_u^2 \approx (0.46197)^{|u|}\sigma_u^2$.

The ratios weight / variance should decrease exponentially with $|u|$.

Heuristics for choosing the weights

Idea 1: take $\gamma_u \approx \sigma_u^2$ or $\gamma_u \approx S_u$ for each u. Too simplistic.

Heuristics for choosing the weights

Idea 1: take $\gamma_{\mathfrak{u}} \approx \sigma^2_{\mathfrak{u}}$ or $\gamma_{\mathfrak{u}} \approx S_{\mathfrak{u}}$ for each $\mathfrak{u}$. Too simplistic.

Idea 2: Just take simple order-dependent weights. For example, $\gamma_{\mathfrak{u}} = 1$ for $|\mathfrak{u}| \leq d$ and $\gamma_{\mathfrak{u}} = 0$ otherwise. Wang (2007) recommends this with $d = 2$.

Heuristics for choosing the weights

Idea 1: take $\gamma_{\mathfrak{u}} \approx \sigma^2_{\mathfrak{u}}$ or $\gamma_{\mathfrak{u}} \approx S_{\mathfrak{u}}$ for each $\mathfrak{u}$. Too simplistic.

Idea 2: Just take simple order-dependent weights. For example, $\gamma_{\mathfrak{u}} = 1$ for $|\mathfrak{u}| \leq d$ and $\gamma_{\mathfrak{u}} = 0$ otherwise. Wang (2007) recommends this with $d = 2$.

Idea 3: In general, one can define a simple parametric model for the weights and then estimate the parameters by matching the ANOVA variances (e.g., Wang and Sloan 2006).

For example, $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for some constants $\gamma_j \geq 0$ (product weights).

Fewer parameters: take $\gamma_j = a\beta^j$ for $a, \beta > 0$ (geometric).

With a weighted $\mathcal{P}_\alpha$-type criterion, we should have $\gamma_{\mathfrak{u}} = \rho^{|\mathfrak{u}|}\sigma^2_{\mathfrak{u}}$ for some $\rho > 0$.

Proposal 4: A strategy for order-dependent weights.

Assume $\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|}$. Need to select $\Gamma_1, \ldots, \Gamma_s$.

For each $\mathfrak{u}$, let $v_{\mathfrak{u}}^2$ be an estimate of the optimal $\gamma_{\mathfrak{u}}$.

Strategy: take Γ_r as the average

$$\Gamma_r = \binom{s}{r}^{-1} \sum_{\{\mathfrak{u}:|\mathfrak{u}|=r\}} v_{\mathfrak{u}}^2.$$

Here, scaling all weights by the same factor changes nothing.

Proposal 5: A strategy for product weights.

Ignore one-dimensional projections; they are the same for all lattices.

The idea is to fit the estimated "optimal weights" over all two-dimensional projections via a least-squares procedure. Then we rescale all the weights by a constant factor to match the ratio of average estimated "optimal weights" over the three-dimensional projections to that over the two-dimensional projections.

Let τ_j be the unscaled weight for projection j. We first minimize

$$R = \sum_{k=1}^{s} \sum_{j=1}^{k-1} \left(\tau_j \tau_k - v_{\{j,k\}}^2 \right)^2 .$$

Differentiating w.r.t. τ_j and equaling to 0, we obtain, for each j,

$$\tau_j \sum_{k=1,\, k \neq j}^{s} \tau_k^2 = \sum_{k=1,\, k \neq j}^{s} \tau_k v_{\{j,k\}}^2 .$$

This can be solved by an iterative fixed-point algorithm:

$$\tau_j^{(0)} = \max_{k,l=1,\ldots,s} v_{\{k,l\}}, \qquad \tau_j^{(i+1)} = \frac{\sum_{k=1,\, k\neq j}^{s} \tau_k^{(i)} v_{\{j,k\}}^2}{\sum_{k=1,\, k\neq j}^{s} \left(\tau_k^{(i)}\right)^2},$$

for $i = 1, 2, \ldots$.

We then rescale the weights via $\gamma_j = c\tau_j$ where the constant c satisfies

$$\frac{\sum_{k=1}^{s}\sum_{j=1}^{k-1} \tau_j\tau_k}{\sum_{k=1}^{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j-1} \tau_j\tau_k\tau_l} = c\, \frac{\sum_{k=1}^{s}\sum_{j=1}^{k-1} v_{\{j,k\}}^2}{\sum_{k=1}^{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j-1} v_{\{j,k,l\}}^2}.$$

Idea 6: Control the shortest vector in dual lattice, for each projection.

Spectral test for LCGs (Knuth, Fishman, etc.):

$$\min_{2 \le r \le t_1} \frac{\ell_{\{1,\ldots,r\}}}{\ell_r^*(n)}$$

where $\ell_{\mathfrak{u}}$ is the length of a shortest vector in $L_s^*(\mathfrak{u})$ and $\ell_r^*(n)$ is a theoretical upper bound on this length, in r dimensions.

Advantages: Computing time of $\ell_{\mathfrak{u}}$ are almost independent of n, although exponential in $|\mathfrak{u}|$. Poor lattices can be eliminated quickly: search is fast.

Idea 6: Control the shortest vector in dual lattice, for each projection.

Lemieux and L'Ecuyer (2000, etc.) maximize

$$M_{t_1,\ldots,t_d} = \min \left[\min_{2\le r\le t_1} \frac{\ell_{\{1,\ldots,r\}}}{\ell_r^*(n)}, \min_{2\le r\le d} \min_{\substack{\mathfrak{u}=\{j_1,\ldots,j_r\}\subset\{1,\ldots,s\} \\ 1=j_1<\cdots<j_r\le t_r}} \frac{\ell_{\mathfrak{u}}}{\ell_r^*(n)} \right],$$

where $\ell_{\mathfrak{u}}$ is the length of a shortest vector in $L_s^*(\mathfrak{u})$ and $\ell_r^*(n)$ is a theoretical upper bound on this length, in r dimensions.

Advantages: Computing time of $\ell_{\mathfrak{u}}$ are almost independent of n, although exponential in $|\mathfrak{u}|$. Poor lattices can be eliminated quickly: search is fast.

This can of course be generalized by adding weights to projections.

Searching for lattice parameters

Korobov lattices.

Search for $\mathbf{z} = (1, a, a^2, \ldots, \ldots)$ over all admissible integers a.

Searching for lattice parameters

Korobov lattices.

Search for $\mathbf{z} = (1, a, a^2, \ldots, \ldots)$ over all admissible integers a.

Component by component (CBC) construction.

Let $z_1 = 1$;

For $j = 2, \ldots, s$, find $z_j \in \{1, \ldots, n-1\}$, $\gcd(z_j, n) = 1$, such that $(z_1, \ldots, z_{j-1}, z_j)$ minimizes the selected discrepancy for the first j dimensions.

Searching for lattice parameters

Korobov lattices.

Search for $\mathbf{z} = (1, a, a^2, \ldots, \ldots)$ over all admissible integers a.

Component by component (CBC) construction.

Let $z_1 = 1$;
For $j = 2, \ldots, s$, find $z_j \in \{1, \ldots, n-1\}$, $\gcd(z_j, n) = 1$, such that $(z_1, \ldots, z_{j-1}, z_j)$ minimizes the selected discrepancy for the first j dimensions.

Partial randomized CBC construction.

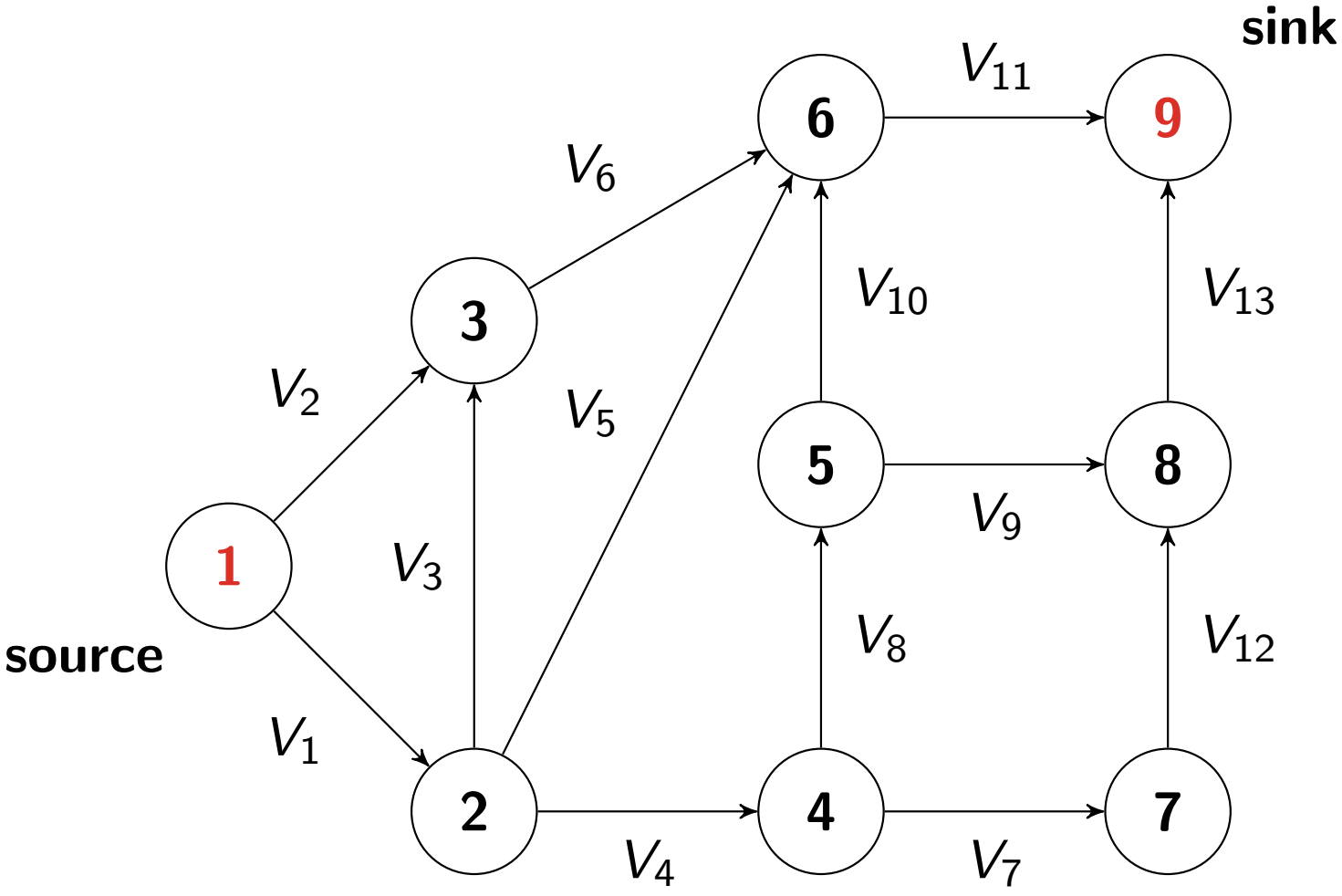
Let $z_1 = 1$;
For $j = 2, \ldots, s$, try r random $z_j \in \{1, \ldots, n-1\}$, $\gcd(z_j, n) = 1$, and retain the one for which $(z_1, \ldots, z_{j-1}, z_j)$ minimizes the selected discrepancy for the first j dimensions.

Example: stochastic activity network

[Elmaghraby 1977]. Each arc j has random length $V_j = F_j^{-1}(U_j)$.

Let $T = f(U_1, \ldots, U_{13})$ = length of longest path from node 1 to node 9.

Want to estimate $q(x) = \mathbb{P}[T > x]$ for a given constant x.



To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and take $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

For **RQMC**, we replace the n realizations of $(U_1, \ldots, U_{13})$ by the n points of a randomly-shifted lattice.

To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and take $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

For **RQMC**, we replace the n realizations of $(U_1, \ldots, U_{13})$ by the n points of a randomly-shifted lattice.

Illustration: $V_j \sim \text{Normal}(\mu_j, \sigma_j^2)$ for $j = 1, 2, 4, 11, 12$, and $V_j \sim \text{Exponential}(1/\mu_j)$ otherwise.

The μ_j: 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.

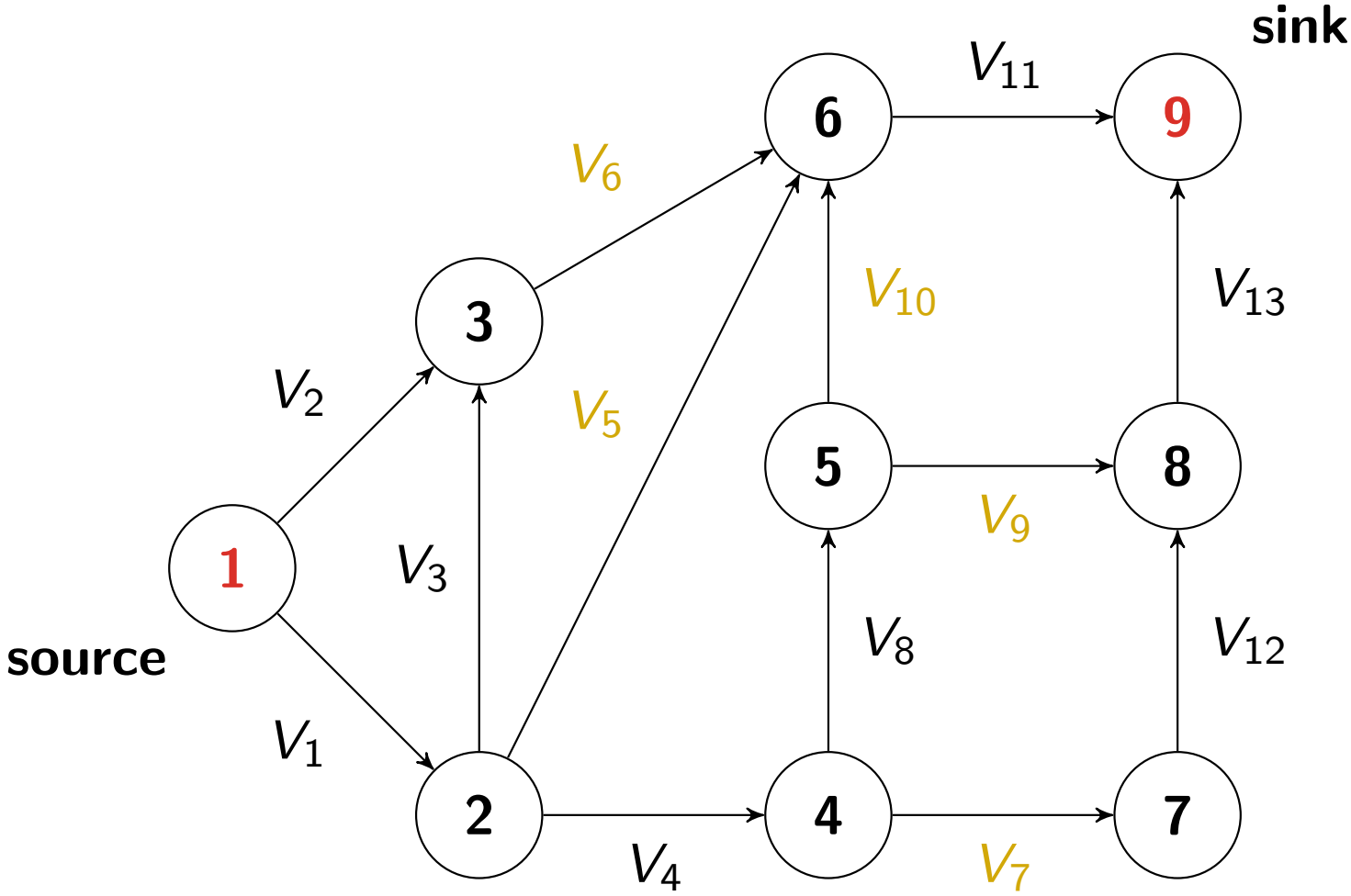
To estimate $q(x)$ by **MC**, we generate n independent realizations of T, say $T_1, \ldots, T_n$, and take $(1/n)\sum_{i=1}^{n} \mathbb{I}[T_i > x]$.

For **RQMC**, we replace the n realizations of $(U_1, \ldots, U_{13})$ by the n points of a randomly-shifted lattice.

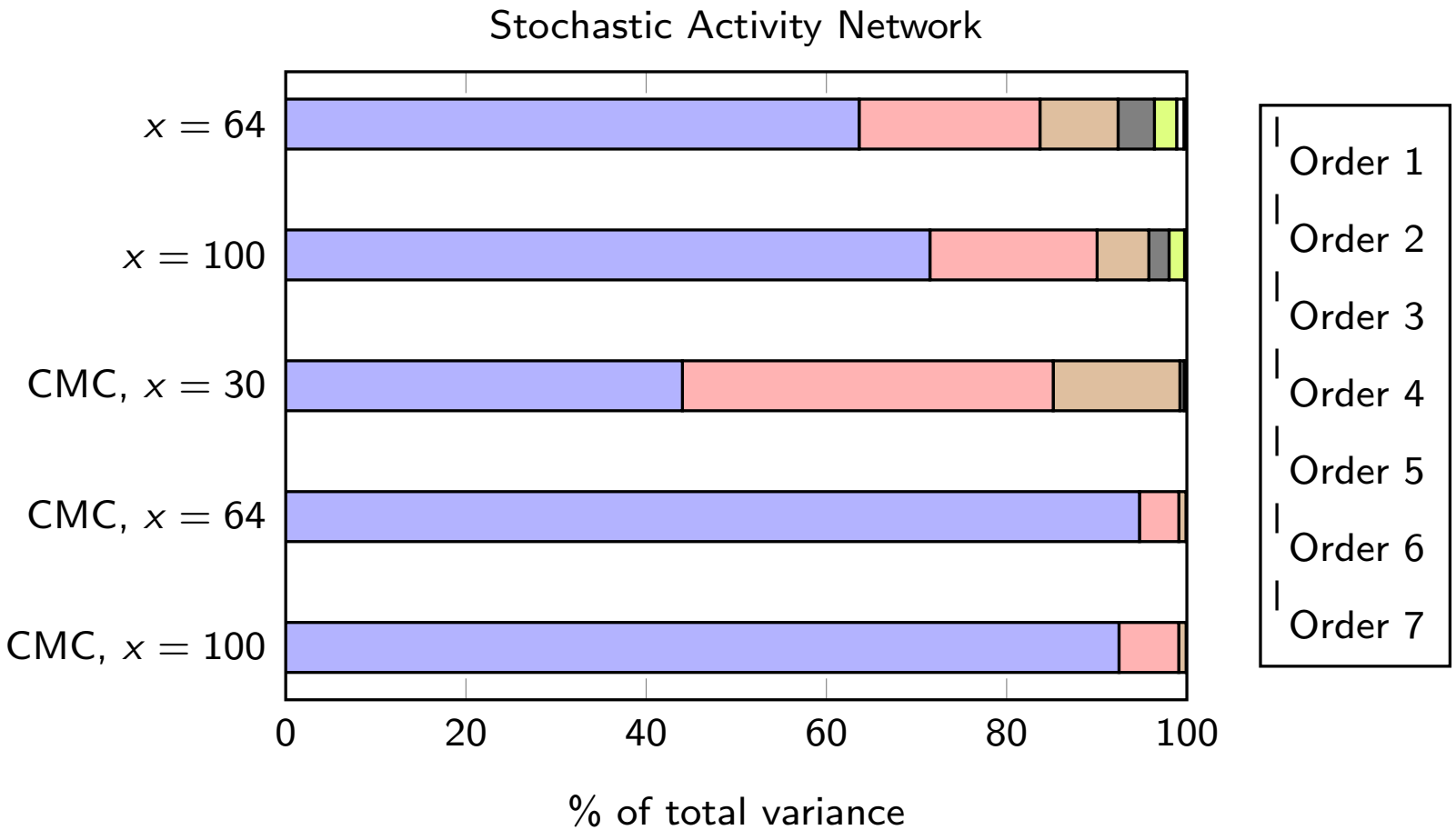
Illustration: $V_j \sim \text{Normal}(\mu_j, \sigma_j^2)$ for $j = 1, 2, 4, 11, 12$, and $V_j \sim \text{Exponential}(1/\mu_j)$ otherwise.

The μ_j: 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5.

CMC estimator. Generate the V_j's only for the 8 arcs that do not belong to the cut $\mathcal{L} = \{5, 6, 7, 9, 10\}$, and replace $\mathbb{I}[T > x]$ by its conditional expectation given those V_j's, $\mathbb{P}[T > x \mid \{V_j, j \notin \mathcal{L}\}]$.
This makes the integrand continuous in the U_j's.


sink
source
1
2
3
4
5
6
7
8
9
V_1
V_2
V_3
V_4
V_5
V_6
V_7
V_8
V_9
V_{10}
V_{11}
V_{12}
V_{13}

ANOVA Variances for the Stochastic Activity Network



ANOVA decomposition

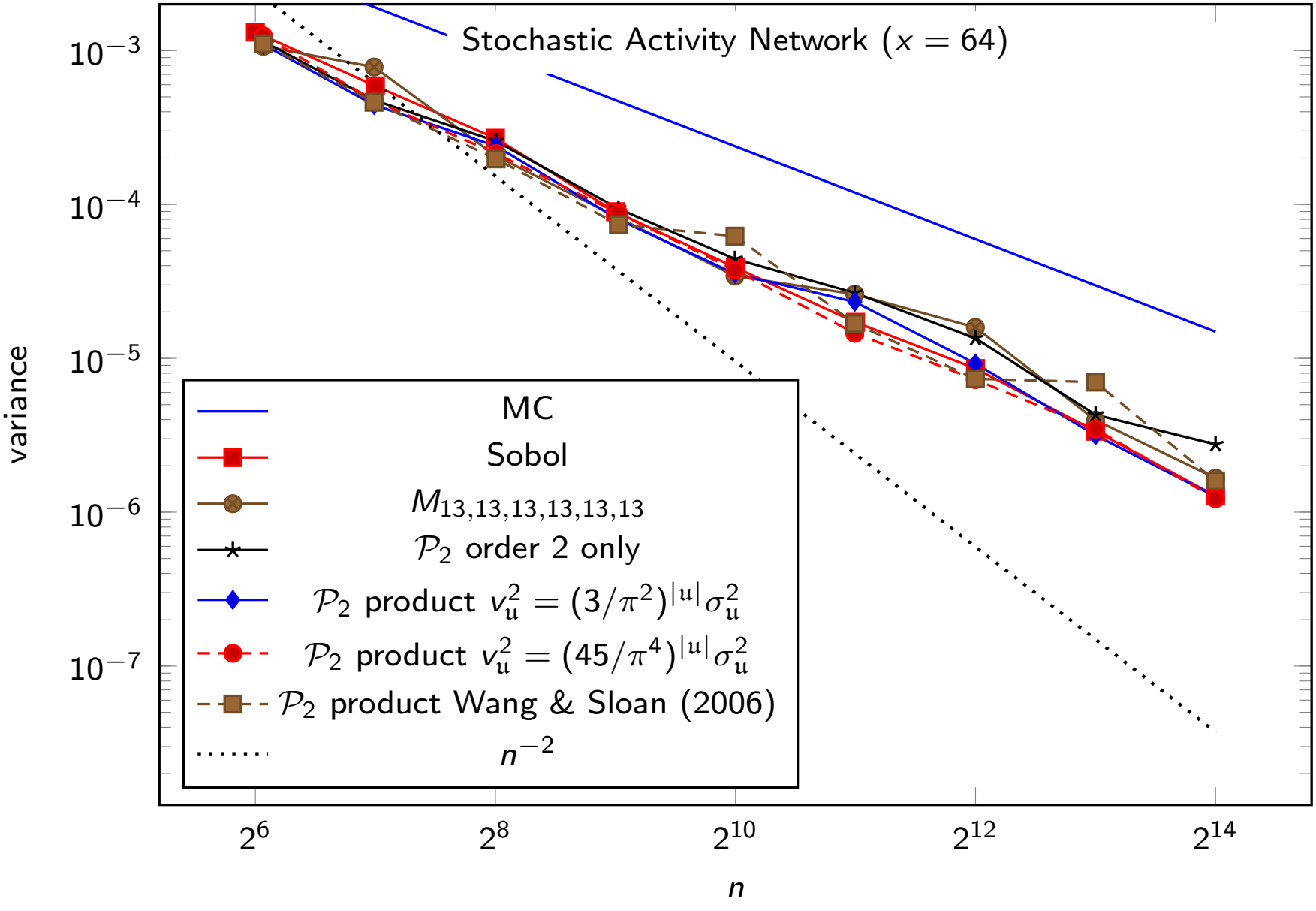
There are six paths from 1 to 9:

$$\{\{1,5,11\},\{2,6,11\},\{1,3,6,11\},\{1,4,7,12,13\},\{1,4,8,9,13\},\{1,4,8,10,11\}\}.$$

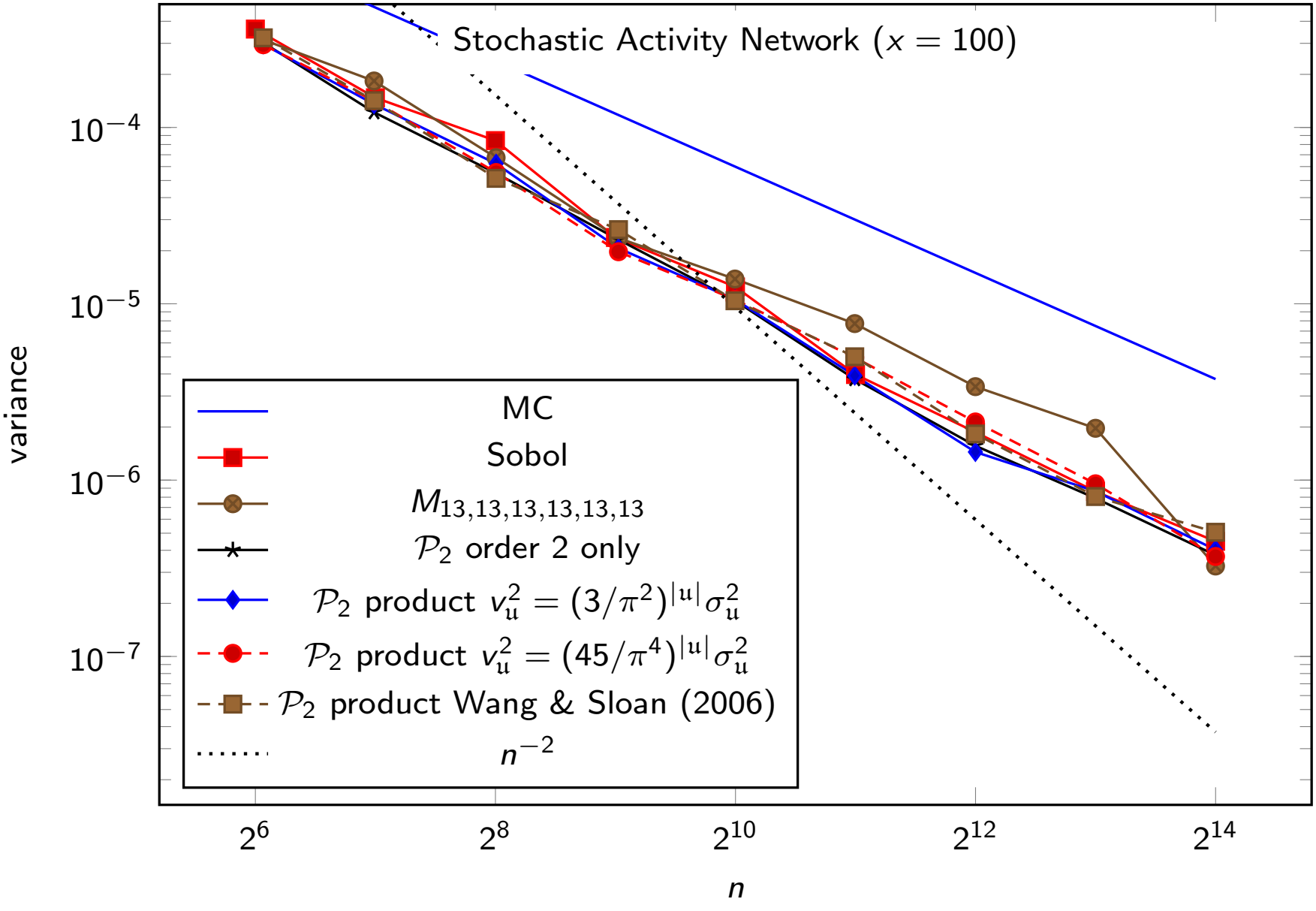
Intuition: the important projections should be only the subsets of those paths.
Fraction of the total variance that lies in these projections:

	$x = 30$	$x = 64$	$x = 100$
crude MC		80.6 %	96.3 %
conditional MC	88.8 %	99.5 %	100 %

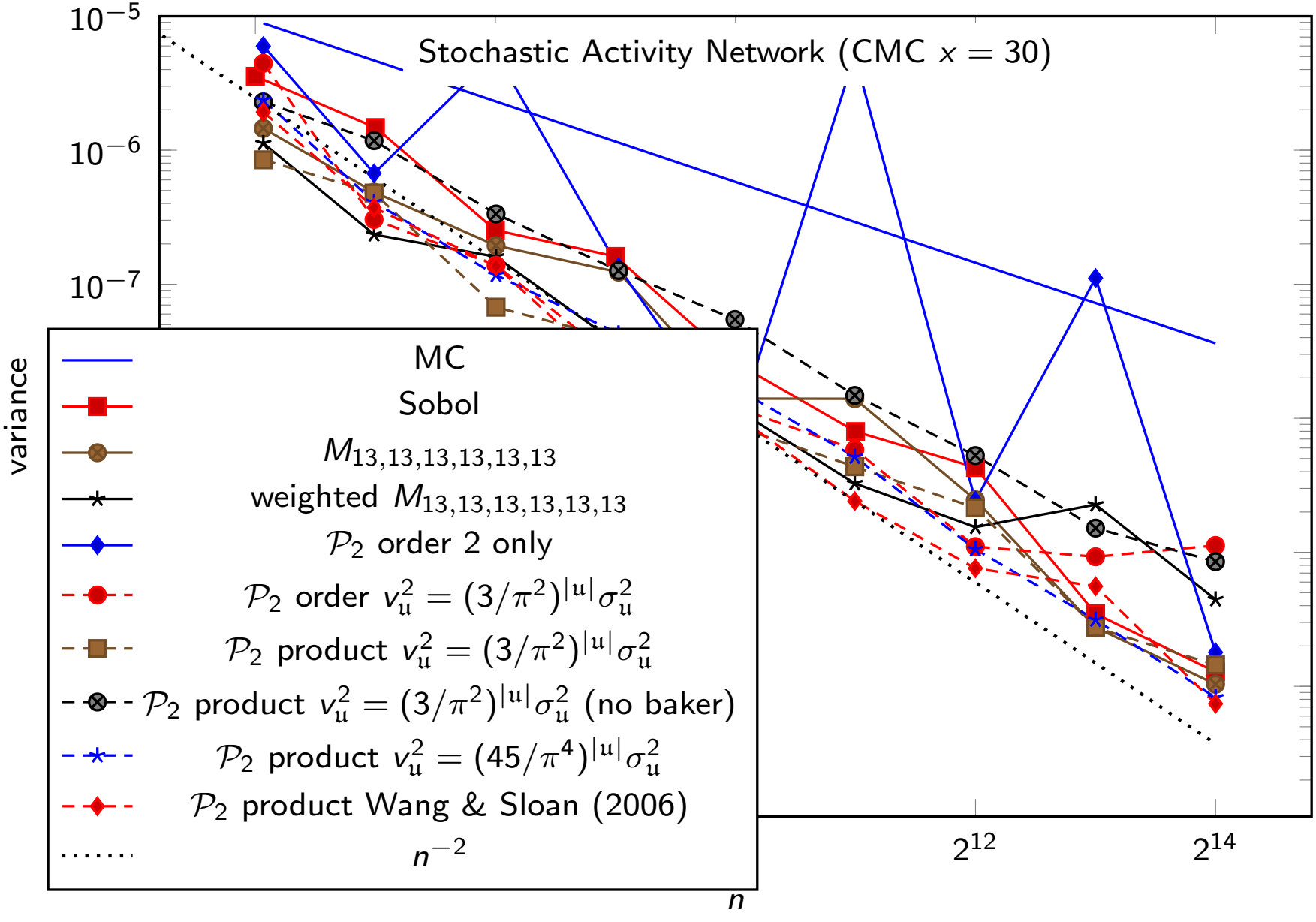
Lattices of Rank 1 with CBC



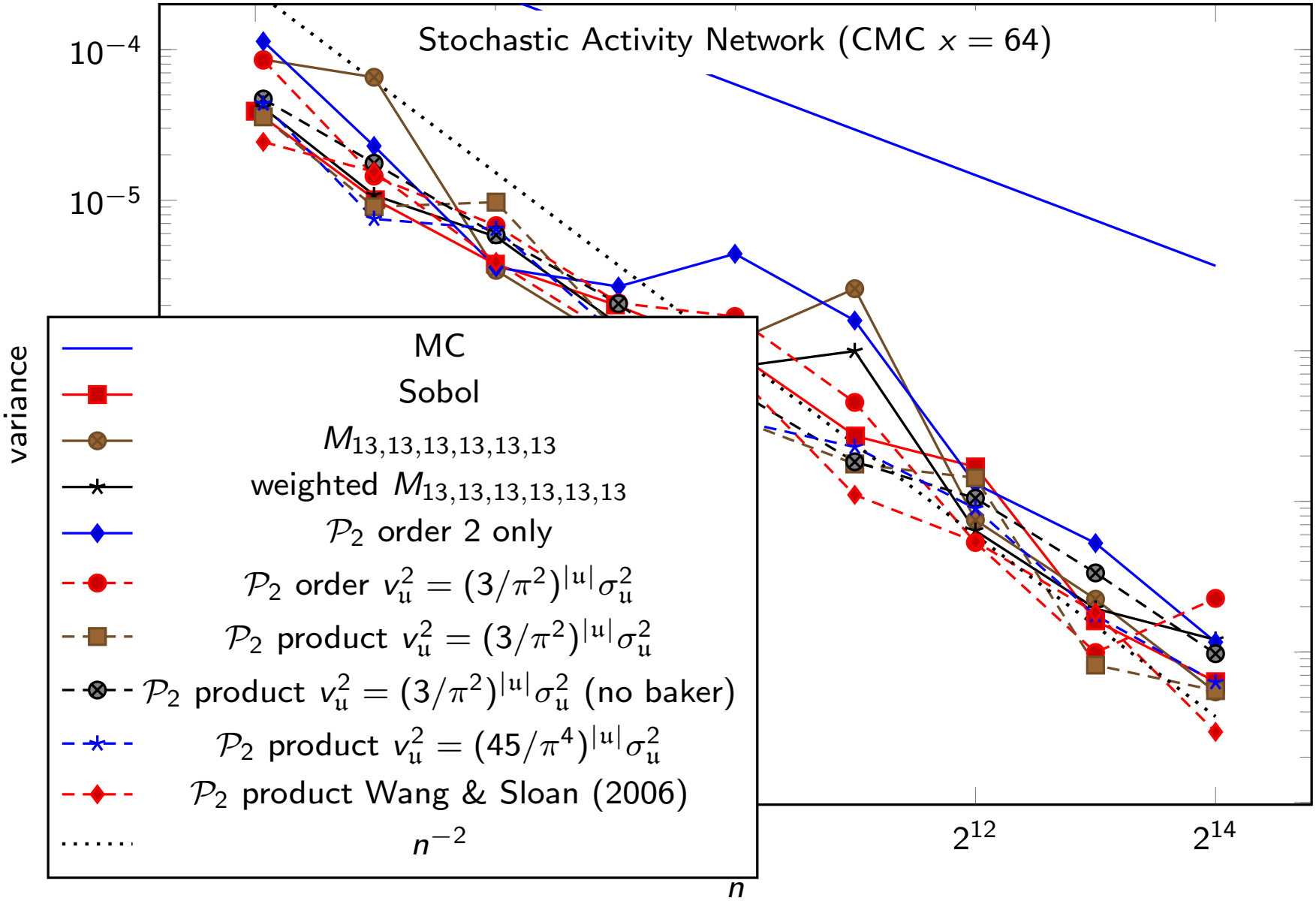
Lattices of Rank 1 with CBC



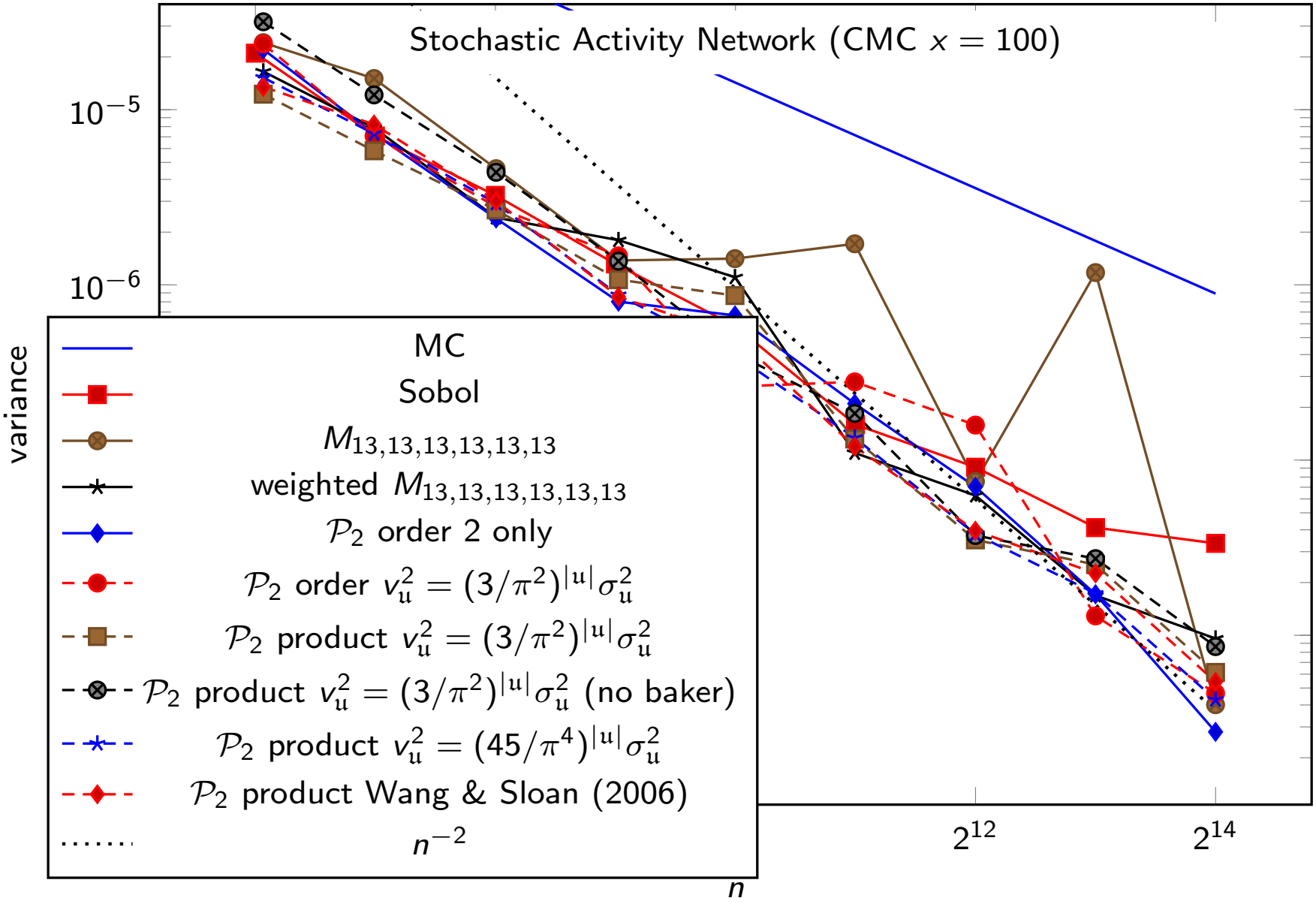
Lattices of Rank 1 with CBC



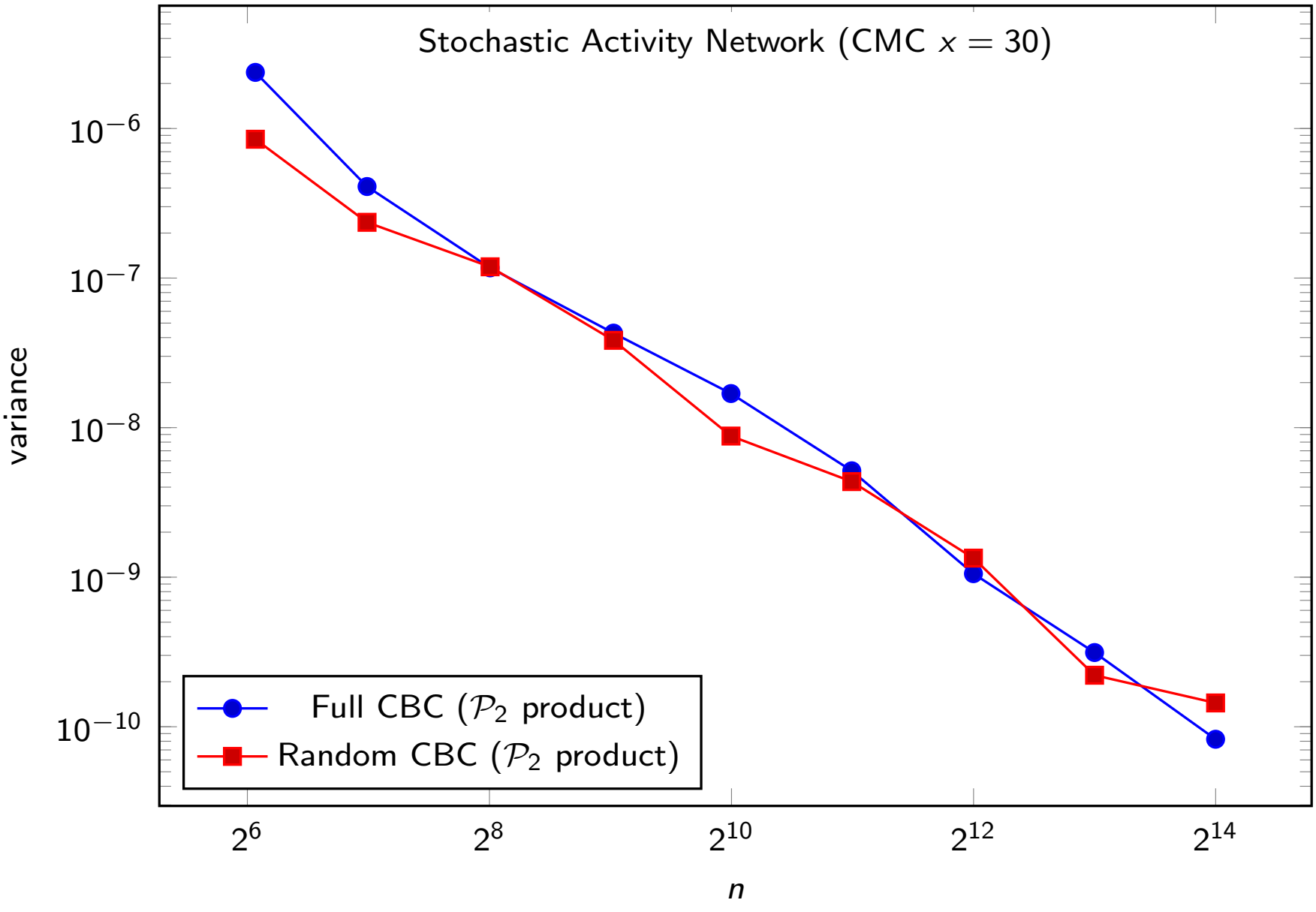
Lattices of Rank 1 with CBC



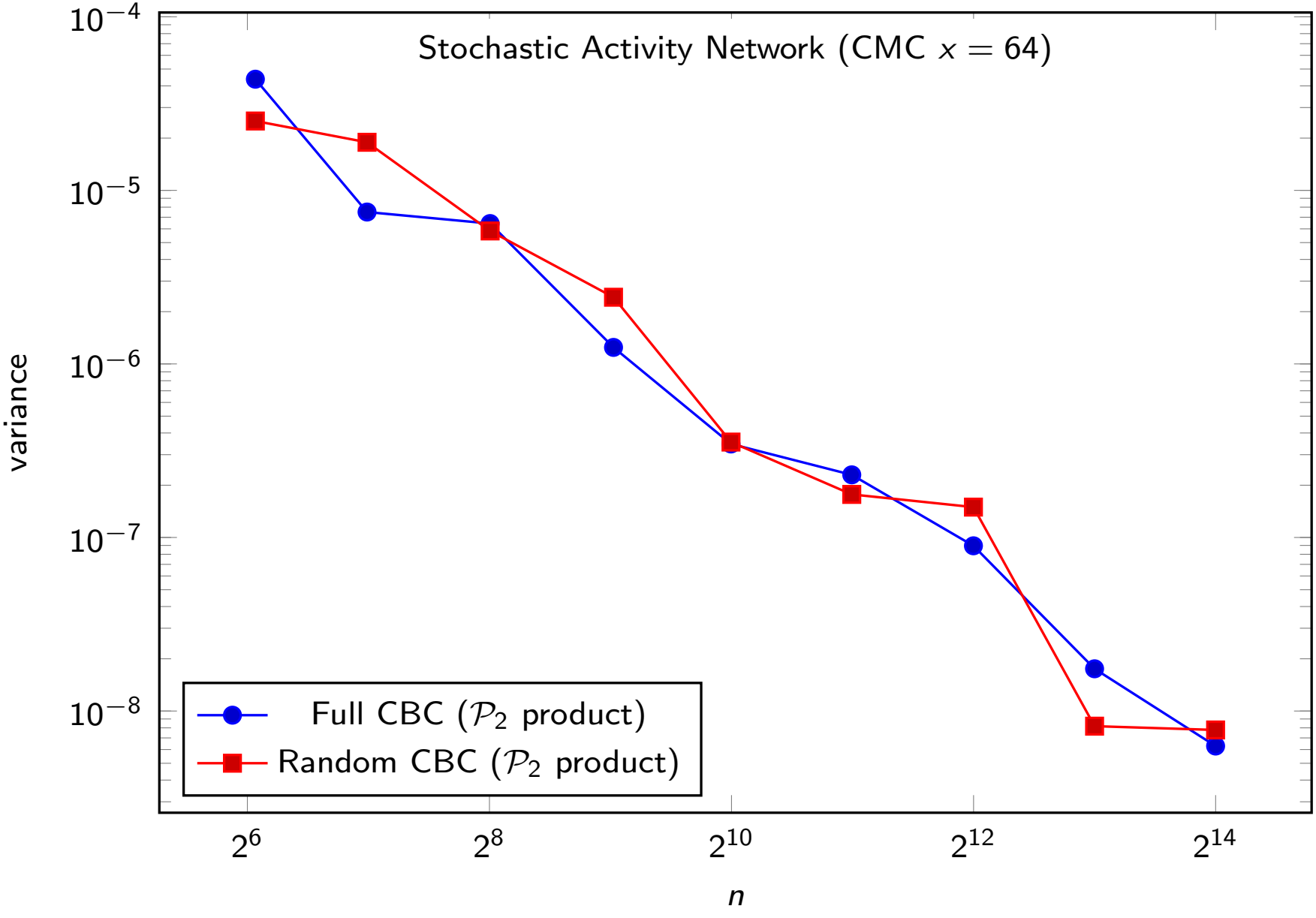
Lattices of Rank 1 with CBC



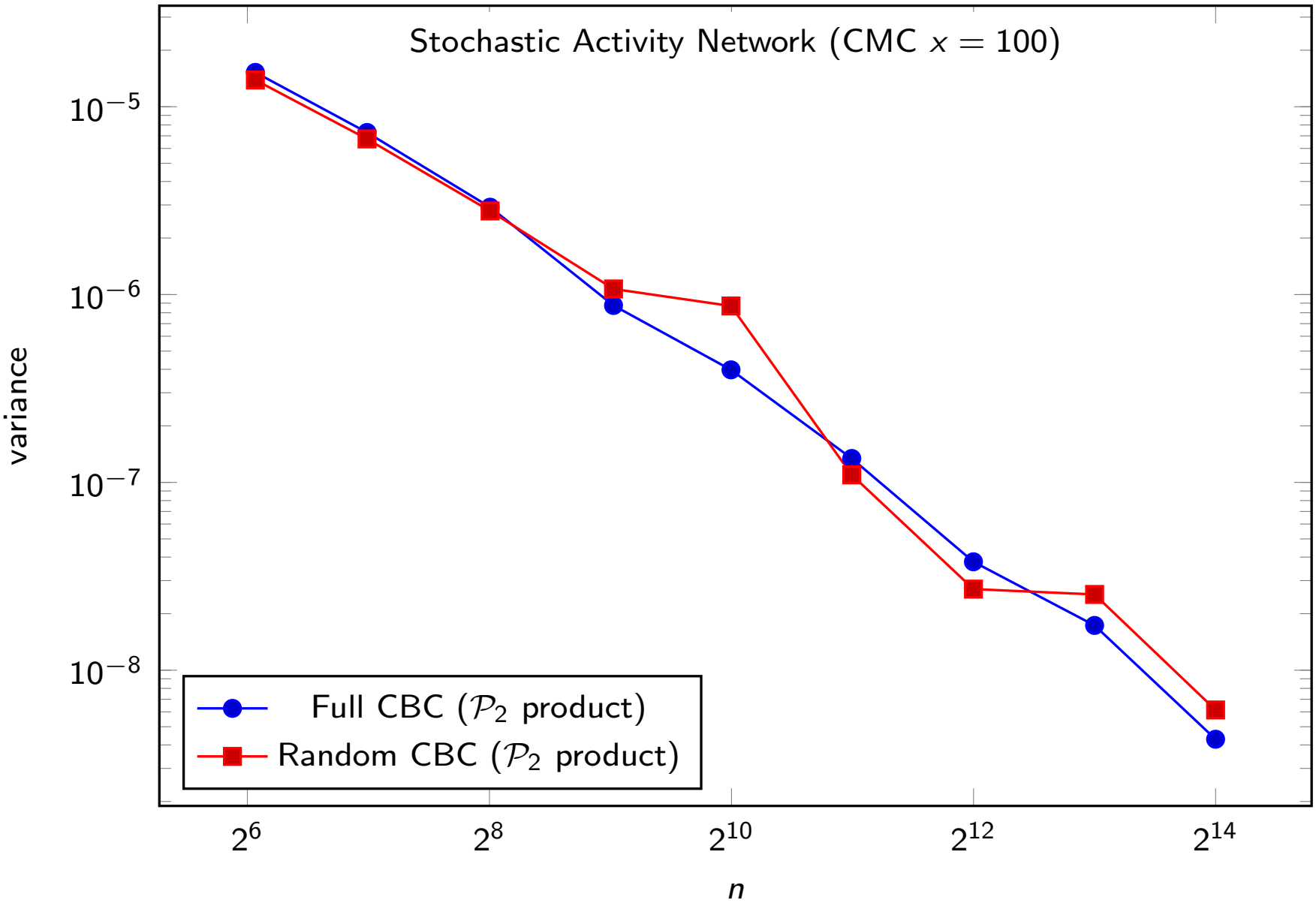
Random vs. Full CBC



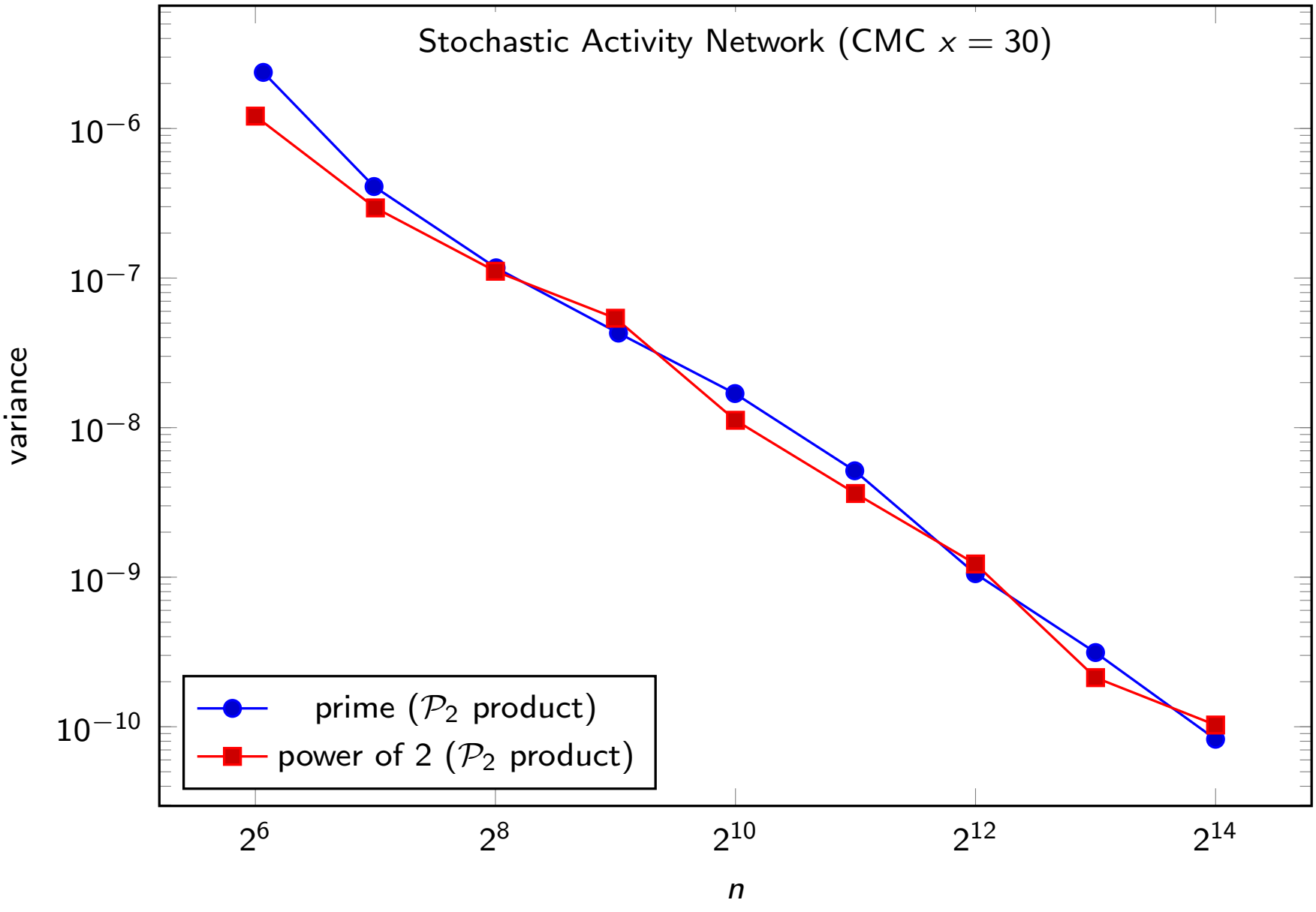
Random vs. Full CBC



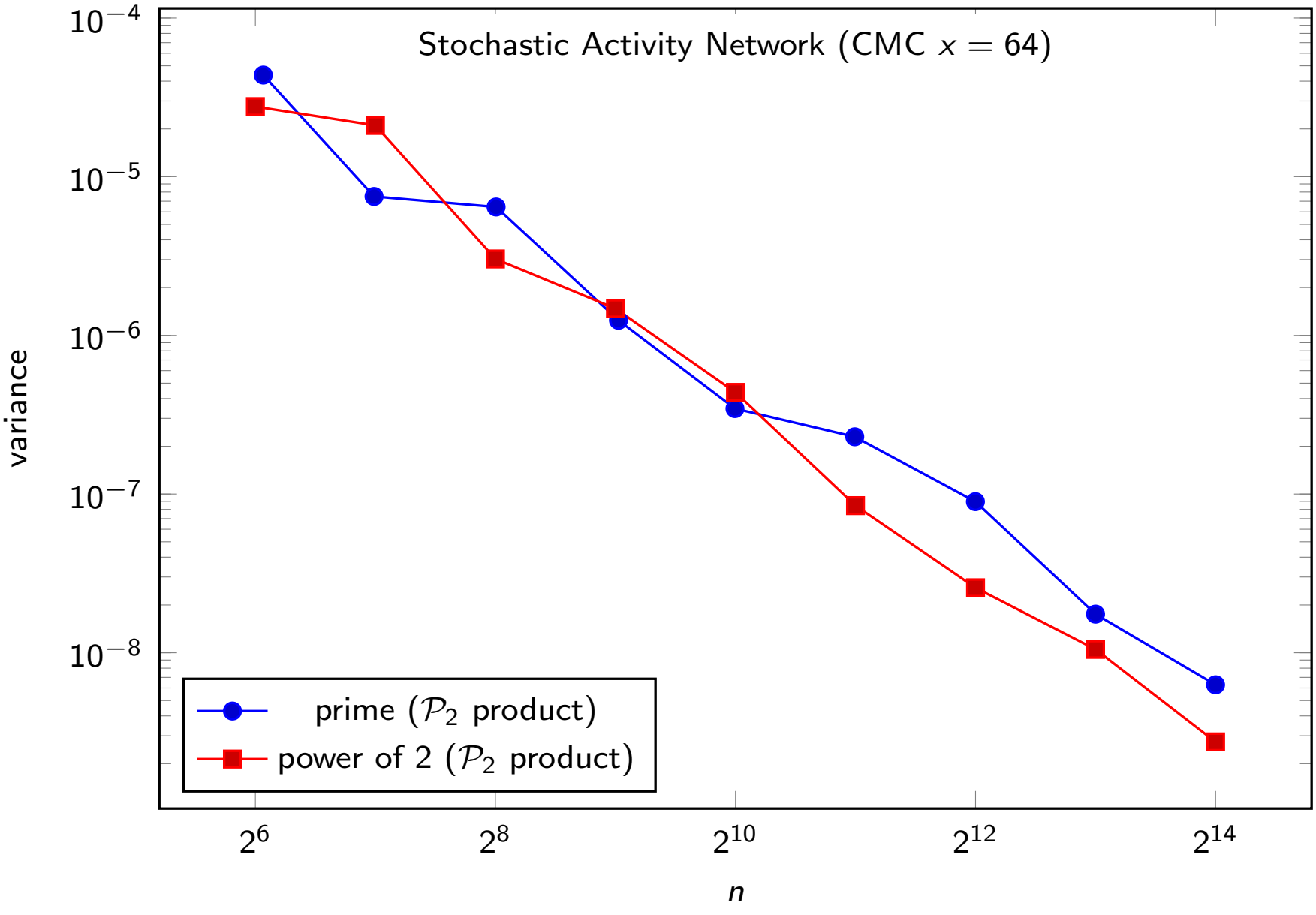
Random vs. Full CBC



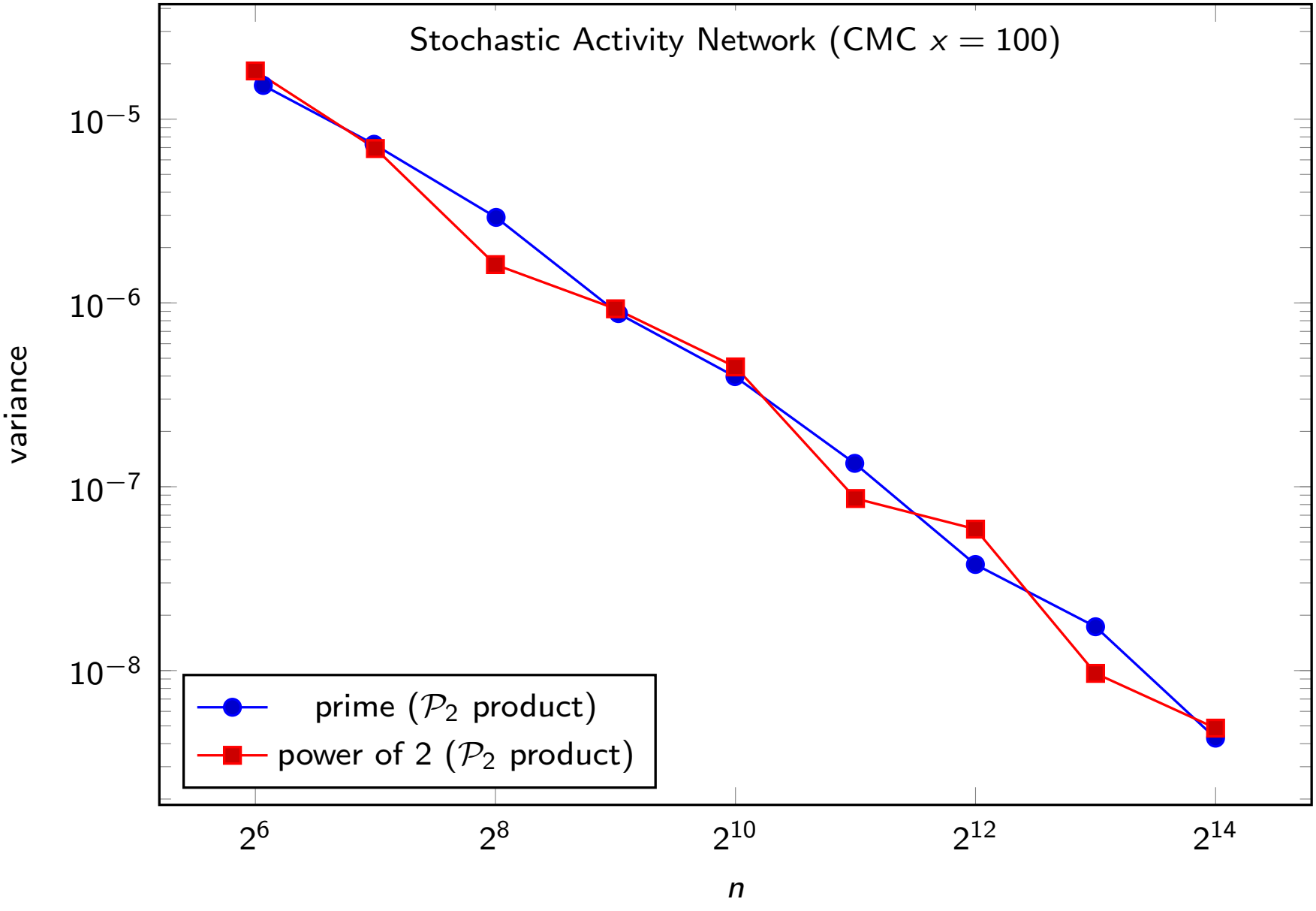
Prime vs. Power-of-2 Number of Points



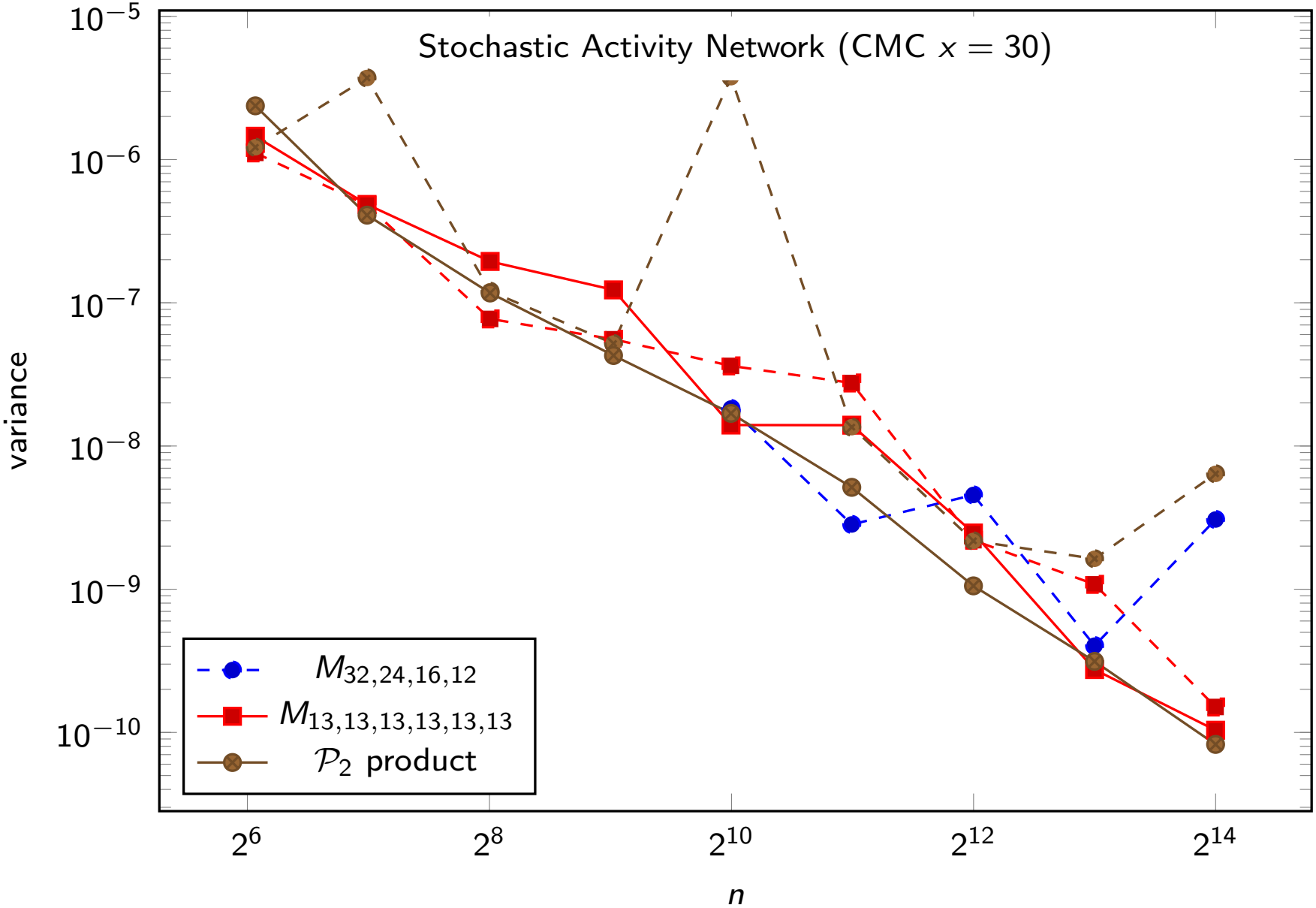
Prime vs. Power-of-2 Number of Points



Prime vs. Power-of-2 Number of Points

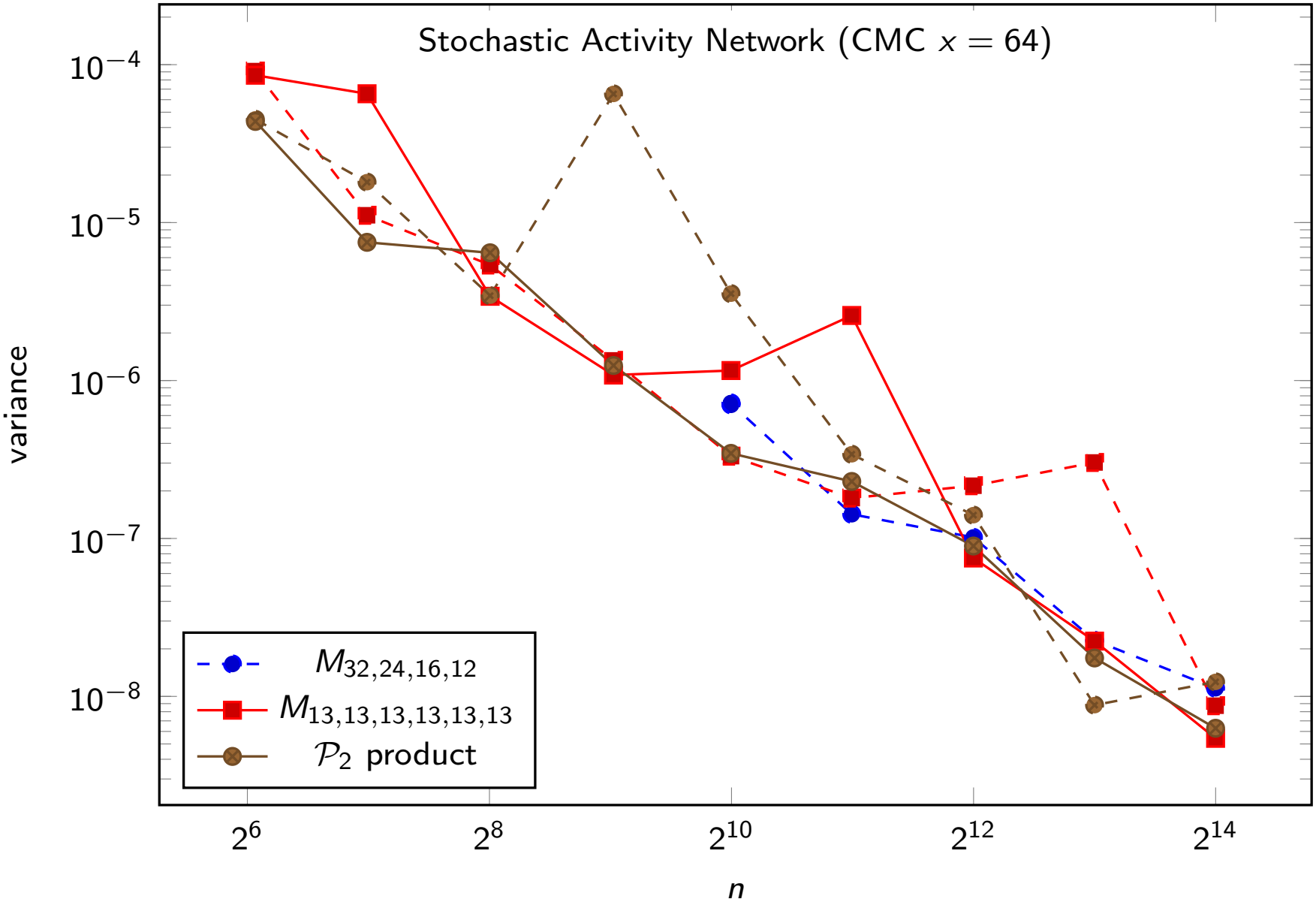


Korobov vs. CBC



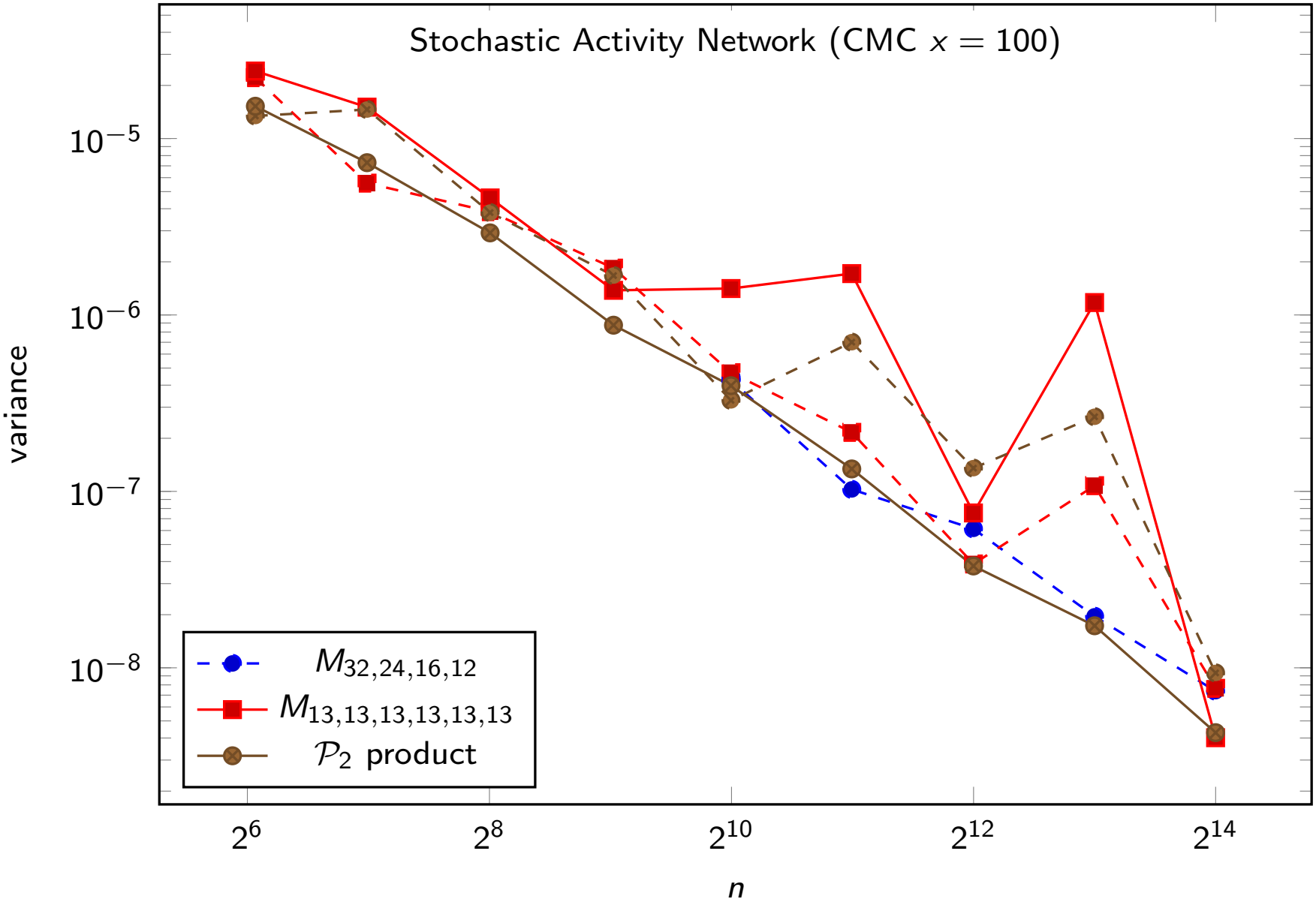
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



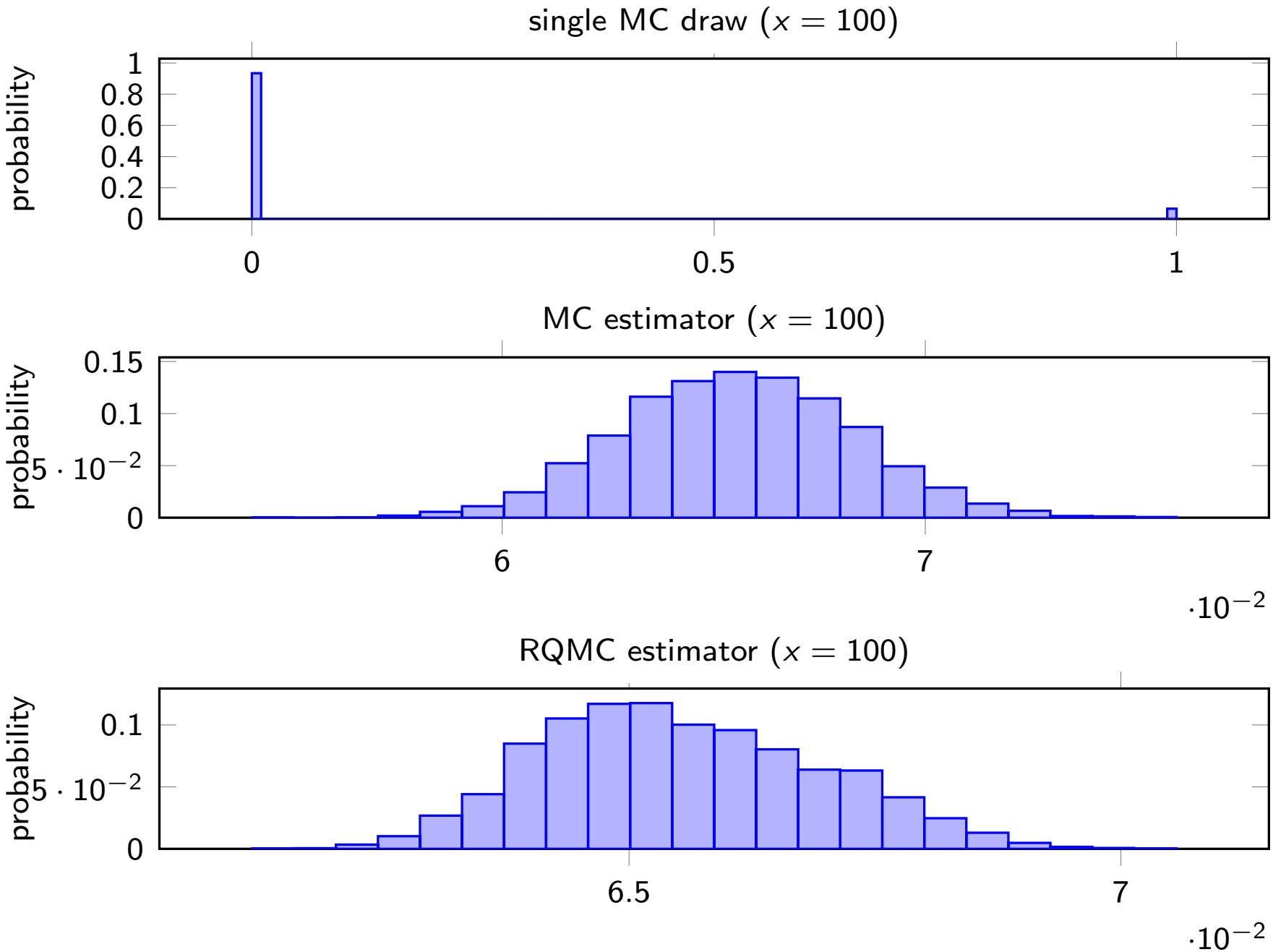
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC

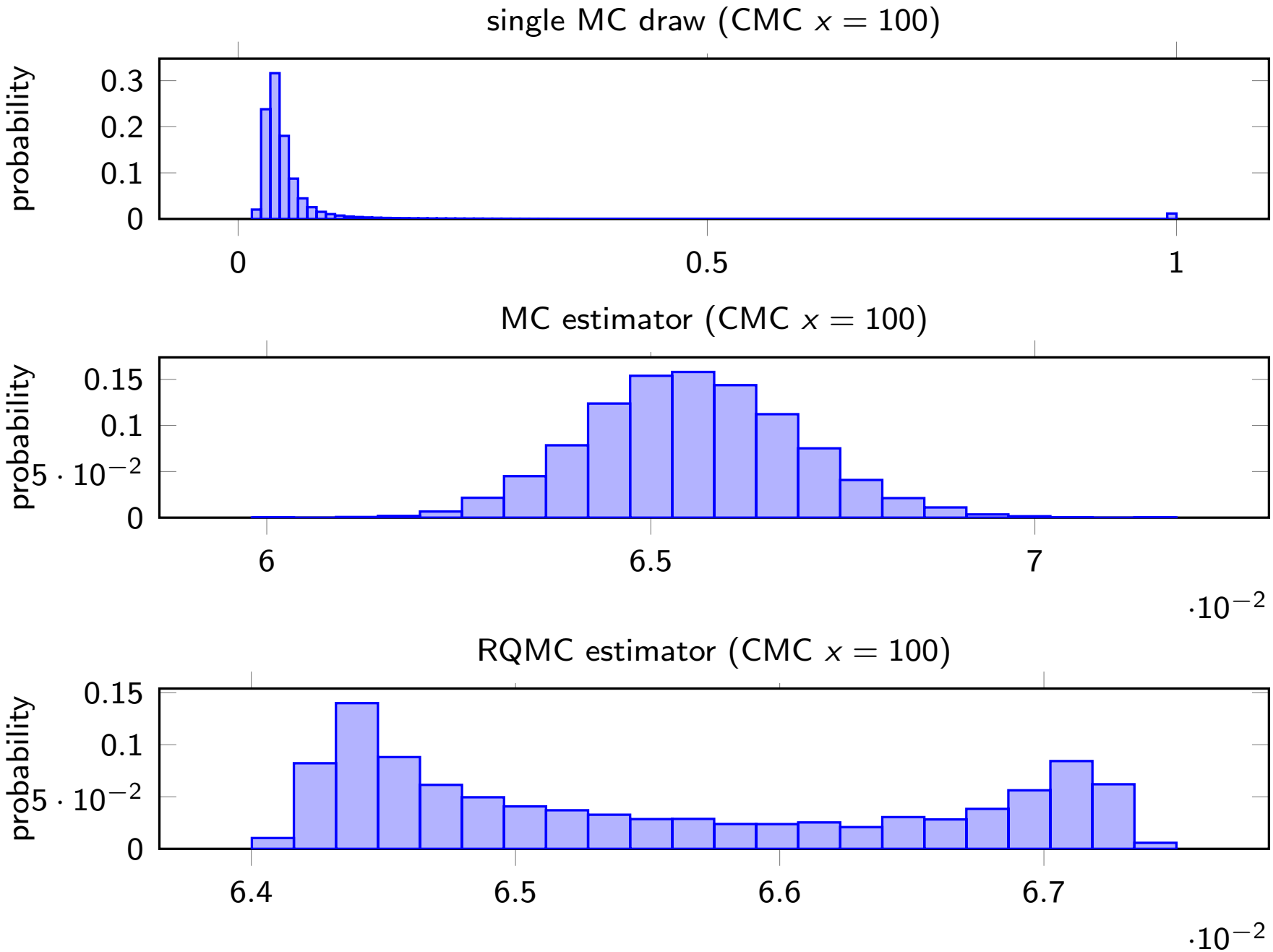


Solid: CBC. Dashed: Korobov.

Histograms, for $n = 8191$, $m = 10^4$ replications



Histograms


single MC draw (CMC $x = 100$)
probability
0.3
0.2
0.1
0
0
0.5
1
MC estimator (CMC $x = 100$)
probability
0.15
0.1
$5 \cdot 10^{-2}$
0
6
6.5
7
$\cdot 10^{-2}$
RQMC estimator (CMC $x = 100$)
probability
0.15
0.1
$5 \cdot 10^{-2}$
0
6.4
6.5
6.6
6.7
$\cdot 10^{-2}$

Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c-dimensional geometric Brownian motions (GMB) at d observations times $0 < t_1 < \cdots < t_d = T$, then we have $s = cd$.

Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c-dimensional geometric Brownian motions (GMB) at d observations times $0 < t_1 < \cdots < t_d = T$, then we have $s = cd$.

To generate $\mathbf{Y}$: Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$, generate $\mathbf{Z} = (Z_1, \ldots, Z_s) = (\Phi^{-1}(U_1), \ldots, \Phi^{-1}(U_s)) \sim N(\mathbf{0}, \mathbf{I})$ and return $\mathbf{Y} = \mathbf{A}\mathbf{Z}$.

Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c-dimensional geometric Brownian motions (GMB) at d observations times $0 < t_1 < \cdots < t_d = T$, then we have $s = cd$.

To generate $\mathbf{Y}$: Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$, generate
$\mathbf{Z} = (Z_1, \ldots, Z_s) = (\Phi^{-1}(U_1), \ldots, \Phi^{-1}(U_s)) \sim N(\mathbf{0}, \mathbf{I})$ and return
$\mathbf{Y} = \mathbf{A}\mathbf{Z}$.
Choice of $\mathbf{A}$?

Function of a Multinormal vector

Let $\mu = E[f(\mathbf{U})] = E[g(\mathbf{Y})]$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

For example, if the payoff of a financial derivative is a function of the values taken by a c-dimensional geometric Brownian motions (GMB) at d observations times $0 < t_1 < \cdots < t_d = T$, then we have $s = cd$.

To generate $\mathbf{Y}$: Decompose $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^{\mathrm{t}}$, generate
$\mathbf{Z} = (Z_1, \ldots, Z_s) = (\Phi^{-1}(U_1), \ldots, \Phi^{-1}(U_s)) \sim N(\mathbf{0}, \mathbf{I})$ and return
$\mathbf{Y} = \mathbf{A}\mathbf{Z}$.
Choice of $\mathbf{A}$?

Cholesky factorization: $\mathbf{A}$ is lower triangular.

Principal component decomposition (PCA):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors.

Principal component decomposition (PCA):

$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors. With this $\mathbf{A}$, Z_1 accounts for the maximum amount of variance of $\mathbf{Y}$, then Z_2 for the maximum amount of variance conditional on Z_1, and so on.

Principal component decomposition (PCA):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors. With this $\mathbf{A}$, Z_1 accounts for the maximum amount of variance of $\mathbf{Y}$, then Z_2 for the maximum amount of variance conditional on Z_1, and so on.

Function of a Brownian motion:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\, t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d$.

Principal component decomposition (PCA):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors. With this $\mathbf{A}$, Z_1 accounts for the maximum amount of variance of $\mathbf{Y}$, then Z_2 for the maximum amount of variance conditional on Z_1, and so on.

Function of a Brownian motion:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\, t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d$.

Sequential (or random walk) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Principal component decomposition (PCA):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors. With this $\mathbf{A}$, Z_1 accounts for the maximum amount of variance of $\mathbf{Y}$, then Z_2 for the maximum amount of variance conditional on Z_1, and so on.

Function of a Brownian motion:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d$.

Sequential (or random walk) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Brownian bridge (BB) sampling: Suppose $d = 2^m$.
Generate $\mathbf{X}(t_d)$, then $\mathbf{X}(t_{d/2})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_d))$,
then $\mathbf{X}(t_{d/4})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_{d/2}))$, and so on.

The first few $N(0,1)$ r.v.'s already sketch the path trajectory.

Principal component decomposition (PCA):
$\mathbf{A} = \mathbf{P}\mathbf{D}^{1/2}$ where $\mathbf{D} = \mathrm{diag}(\lambda_s, \ldots, \lambda_1)$ (eigenvalues of $\mathbf{\Sigma}$ in decreasing order) and the columns of $\mathbf{P}$ are the corresponding unit-length eigenvectors. With this $\mathbf{A}$, Z_1 accounts for the maximum amount of variance of $\mathbf{Y}$, then Z_2 for the maximum amount of variance conditional on Z_1, and so on.

Function of a Brownian motion:
Payoff depends on c-dimensional Brownian motion $\{\mathbf{X}(t),\ t \geq 0\}$ observed at times $0 = t_0 < t_1 < \cdots < t_d$.

Sequential (or random walk) method: generate $\mathbf{X}(t_1)$, then $\mathbf{X}(t_2) - \mathbf{X}(t_1)$, then $\mathbf{X}(t_3) - \mathbf{X}(t_2)$, etc.

Brownian bridge (BB) sampling: Suppose $d = 2^m$.
Generate $\mathbf{X}(t_d)$, then $\mathbf{X}(t_{d/2})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_d))$,
then $\mathbf{X}(t_{d/4})$ conditional on $(\mathbf{X}(0), \mathbf{X}(t_{d/2}))$, and so on.

The first few $N(0,1)$ r.v.'s already sketch the path trajectory.

Each of these methods corresponds to some matrix $\mathbf{A}$.
Choice has large impact on the ANOVA decomposition of f.

Example: Pricing an Asian option

Single asset, s observation times $t_1, \ldots, t_s$. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rt_s} \max\left[0,\ \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right]$$

and $\{S(t),\ t \geq 0\}$ is a geometric Brownian motion.
We have $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Example: Pricing an Asian option

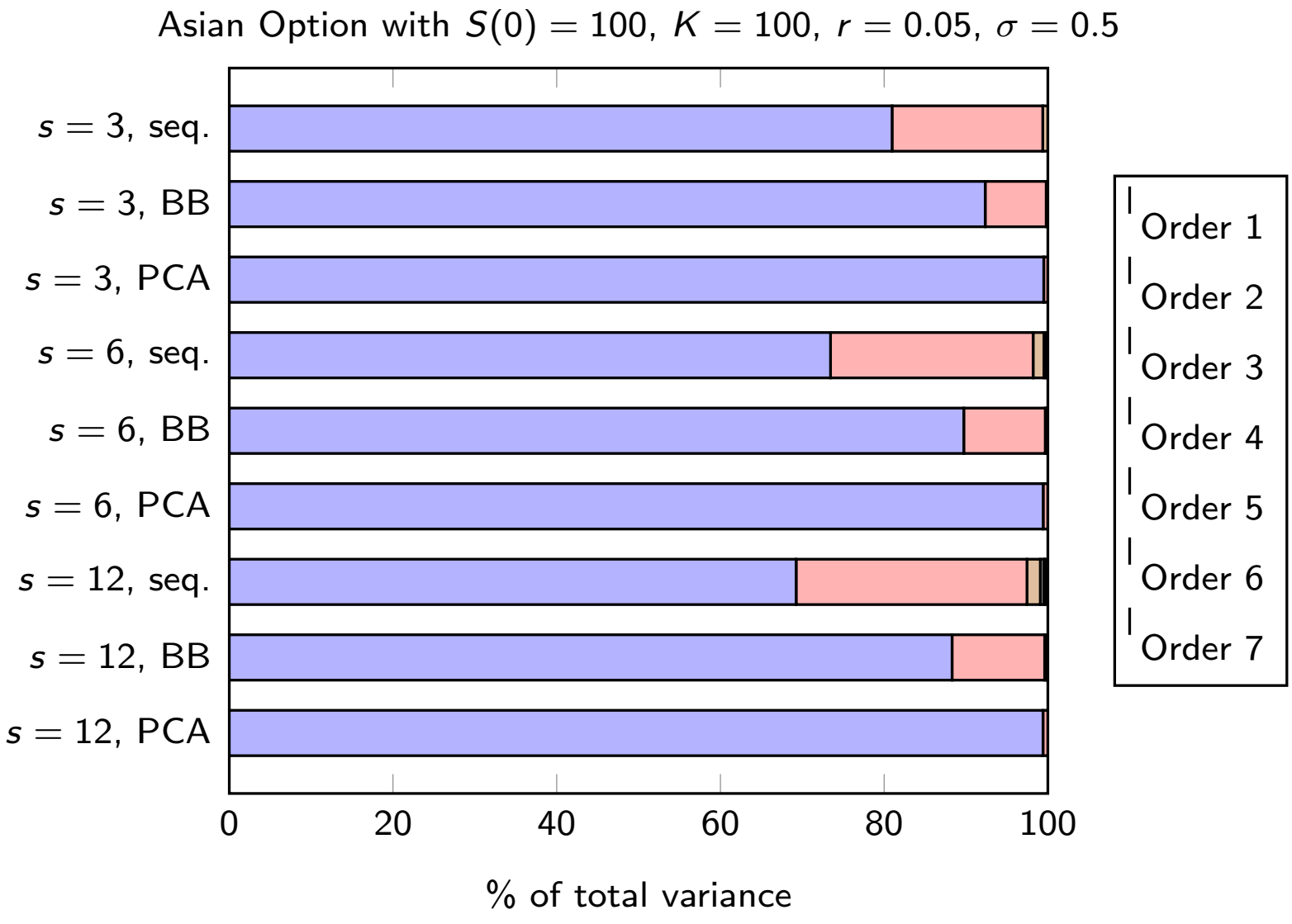
Single asset, s observation times $t_1, \ldots, t_s$. Want to estimate $\mathbb{E}[f(\mathbf{U})]$, where

$$f(\mathbf{U}) = e^{-rt_s} \max\left[0,\ \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right]$$

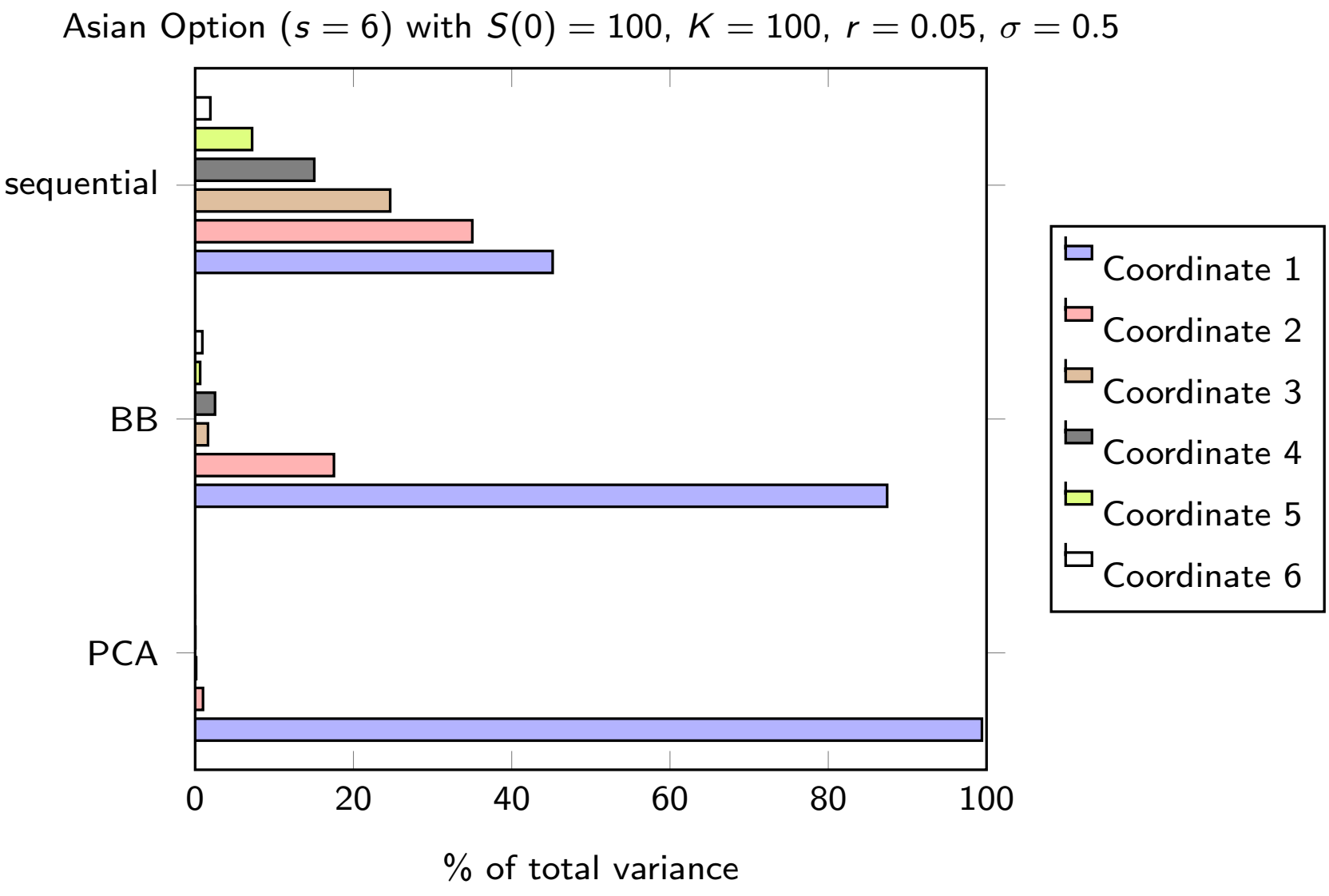
and $\{S(t),\ t \geq 0\}$ is a geometric Brownian motion.
We have $f(\mathbf{U}) = g(\mathbf{Y})$ where $\mathbf{Y} = (Y_1, \ldots, Y_s) \sim N(\mathbf{0}, \mathbf{\Sigma})$.

Let $S(0) = 100$, $K = 100$, $r = 0.05$, $t_s = 1$, and $t_j = jT/s$ for $1 \leq j \leq s$.
We consider $\sigma = 0.2$, 0.5 and $s = 3, 6, 12$.

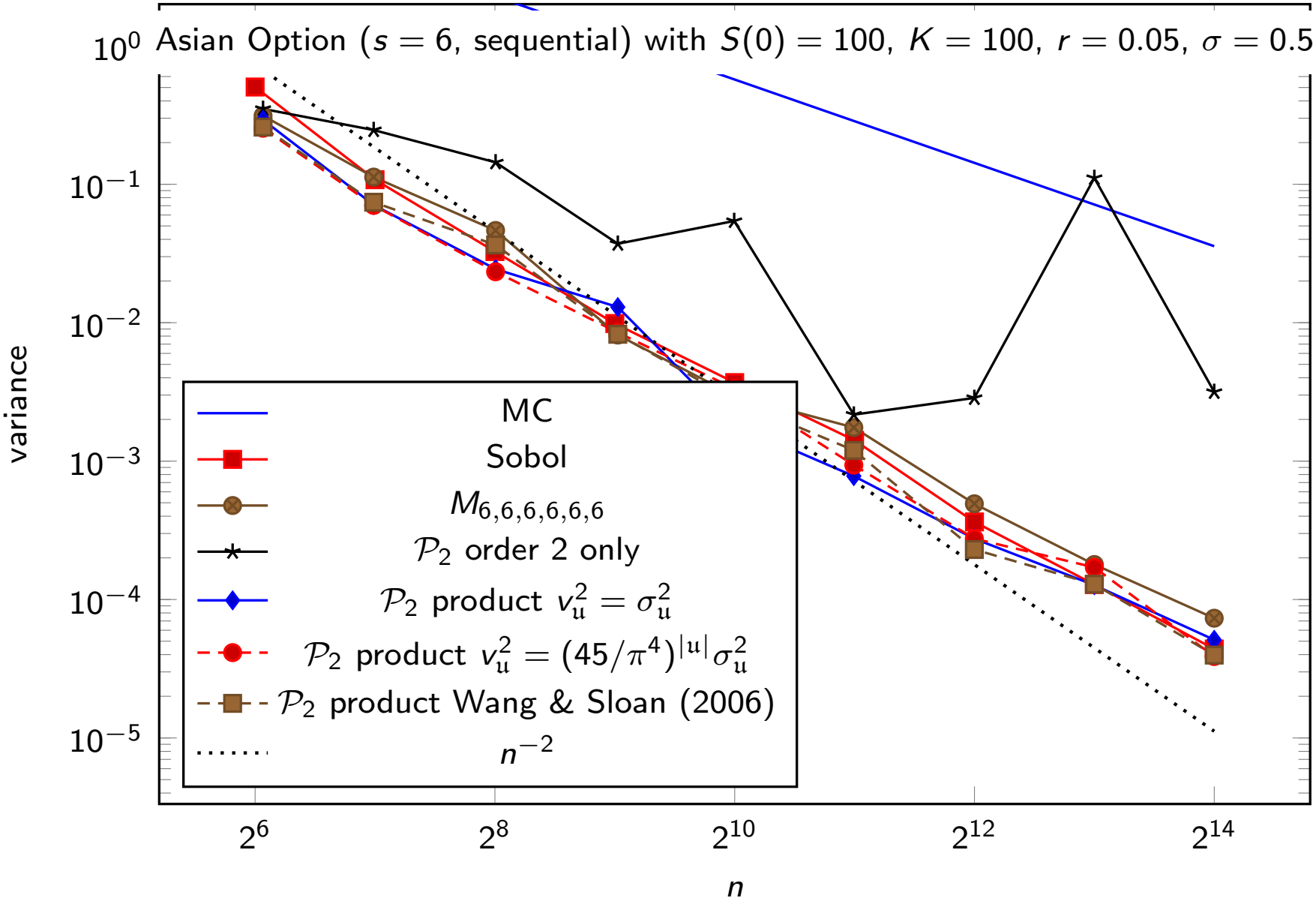
ANOVA Variances for the Asian Option



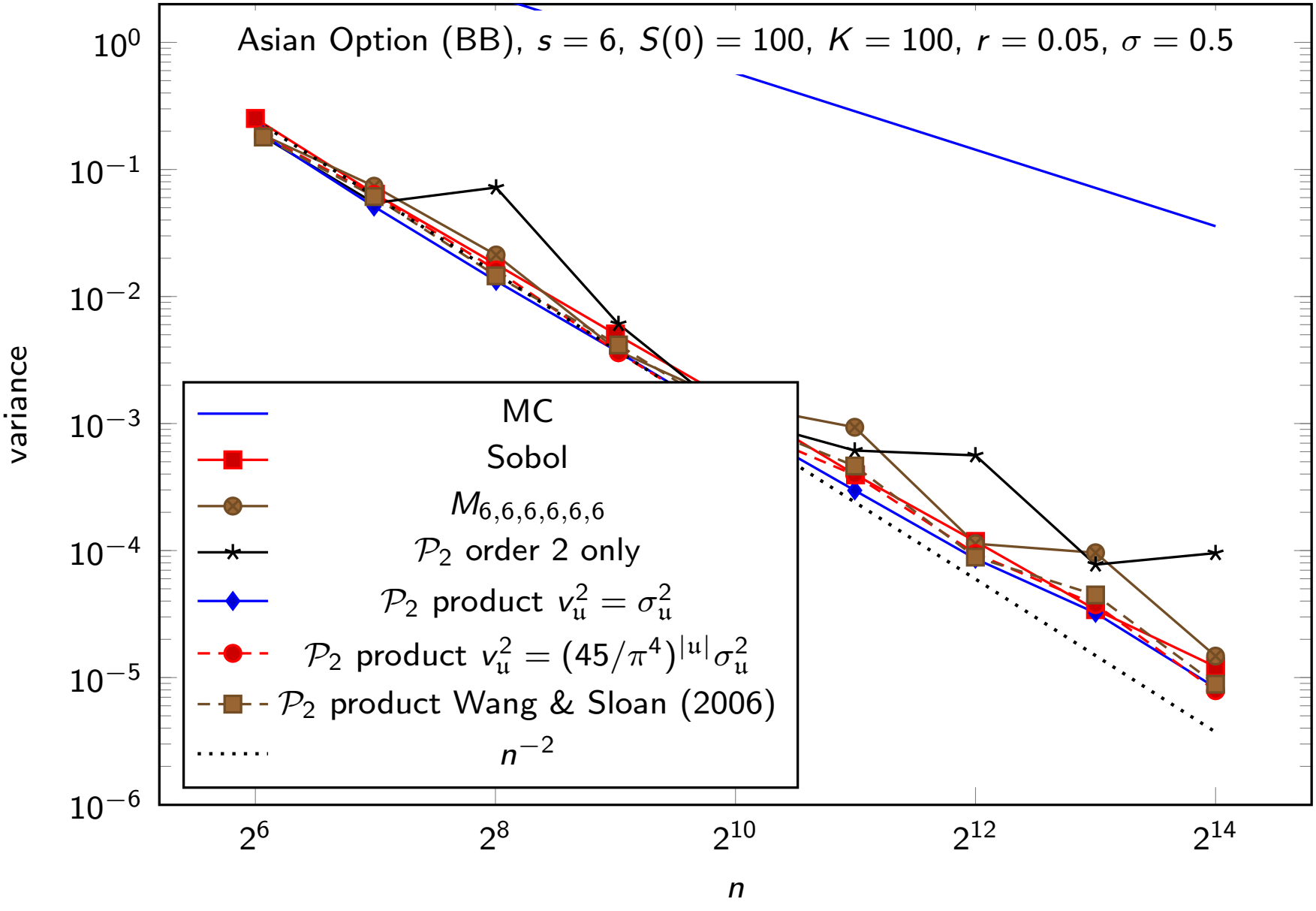
Total Variance per Coordinate for the Asian Option



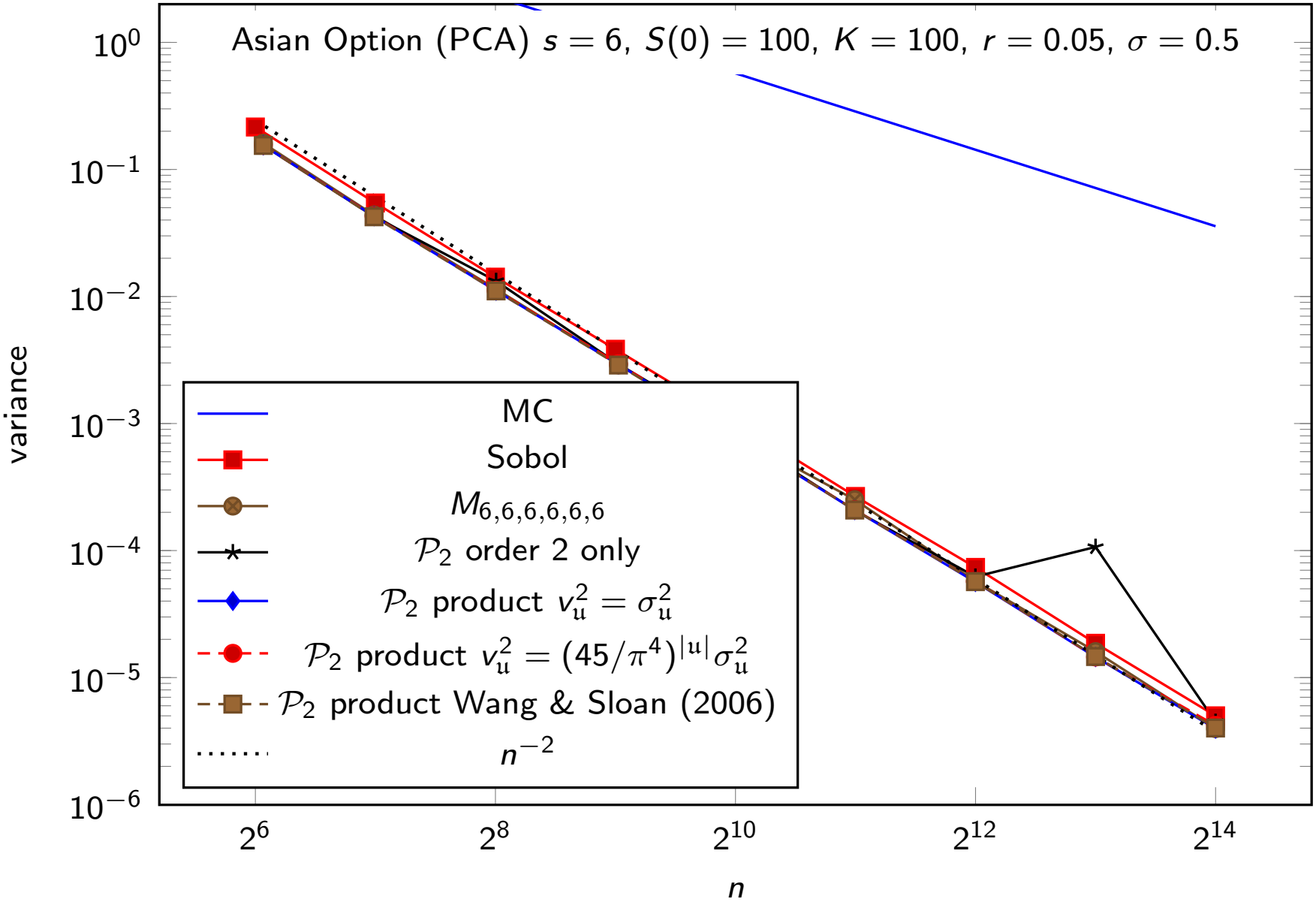
Lattices of Rank 1 with CBC



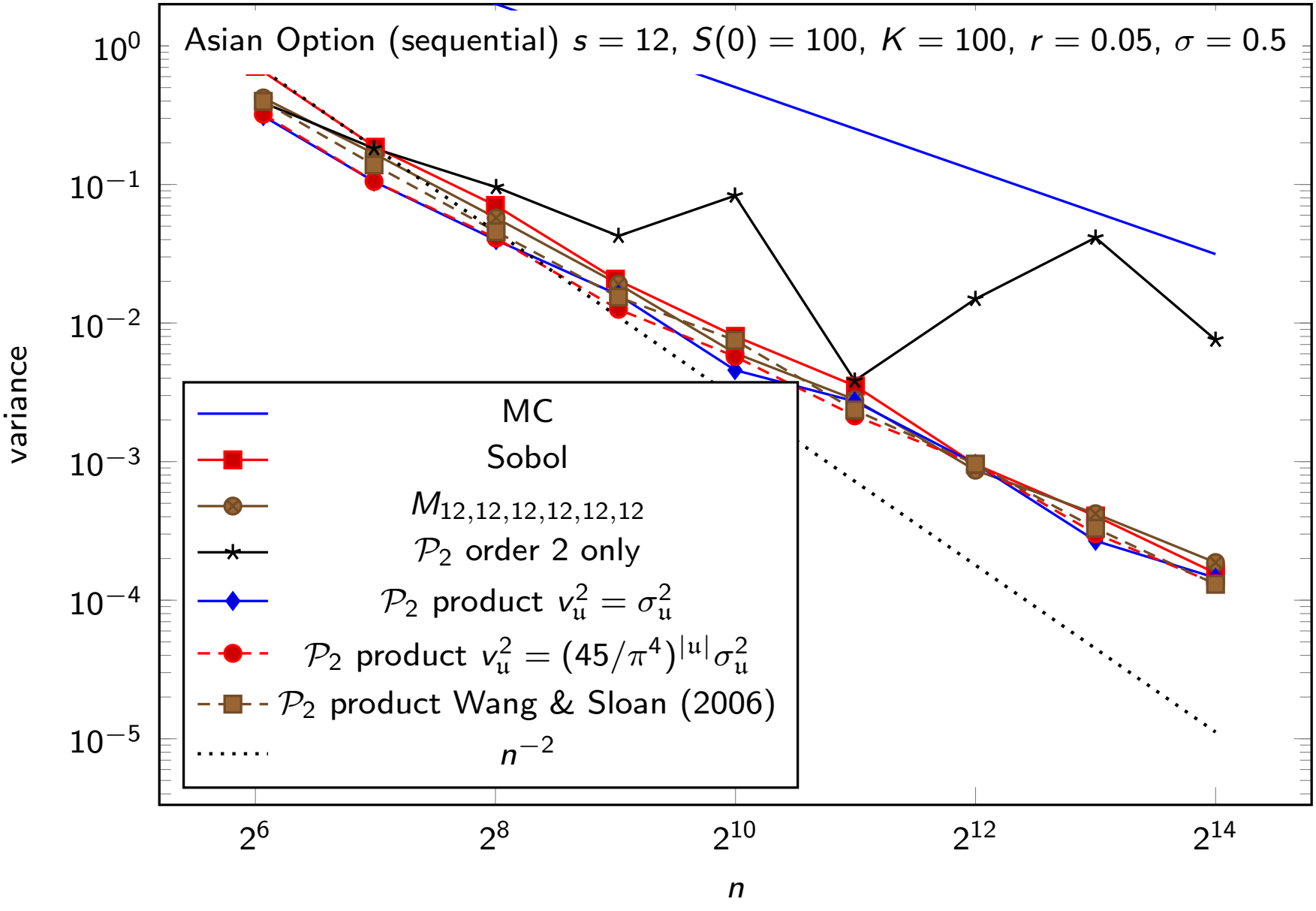
Lattices of Rank 1 with CBC



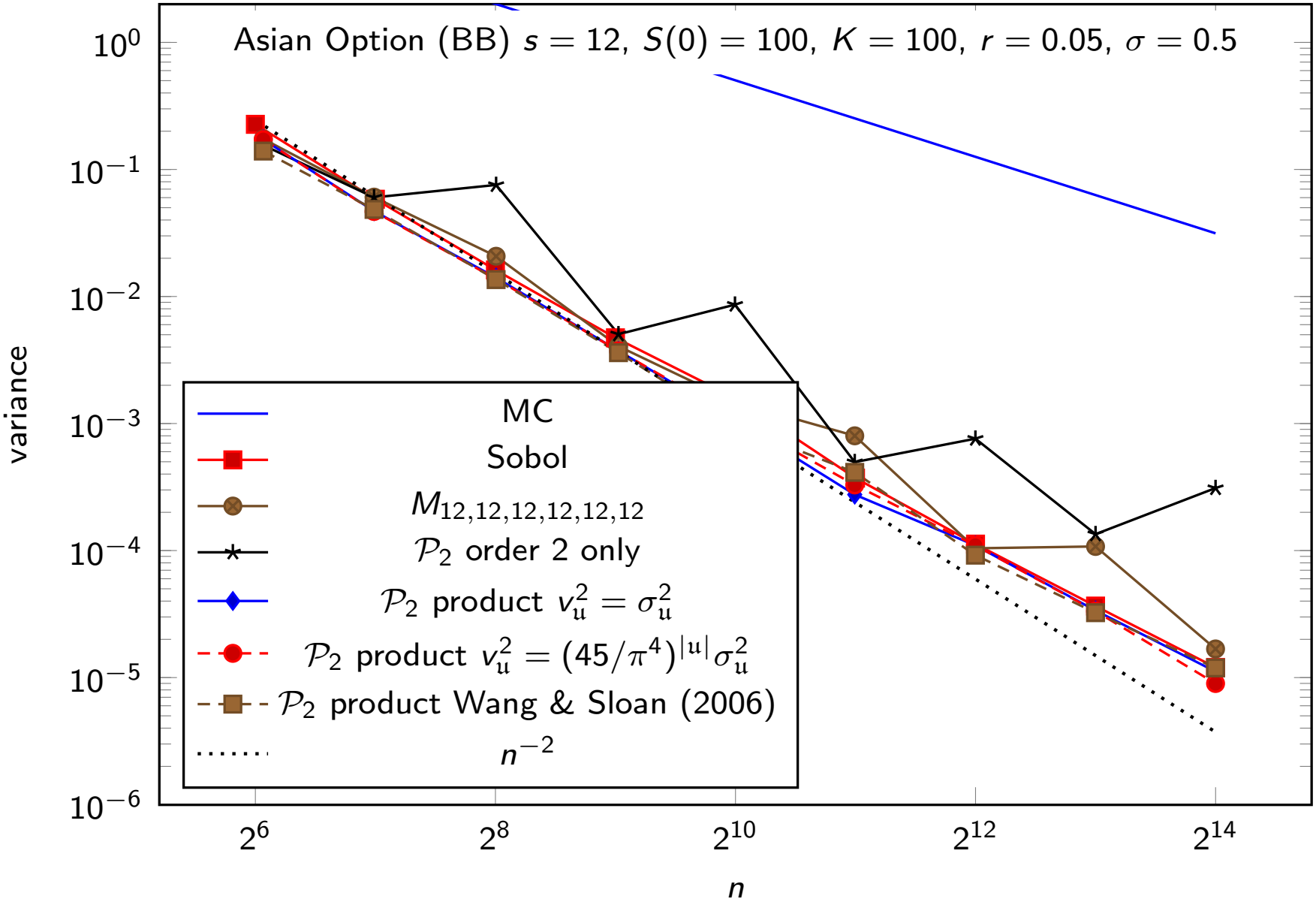
Lattices of Rank 1 with CBC



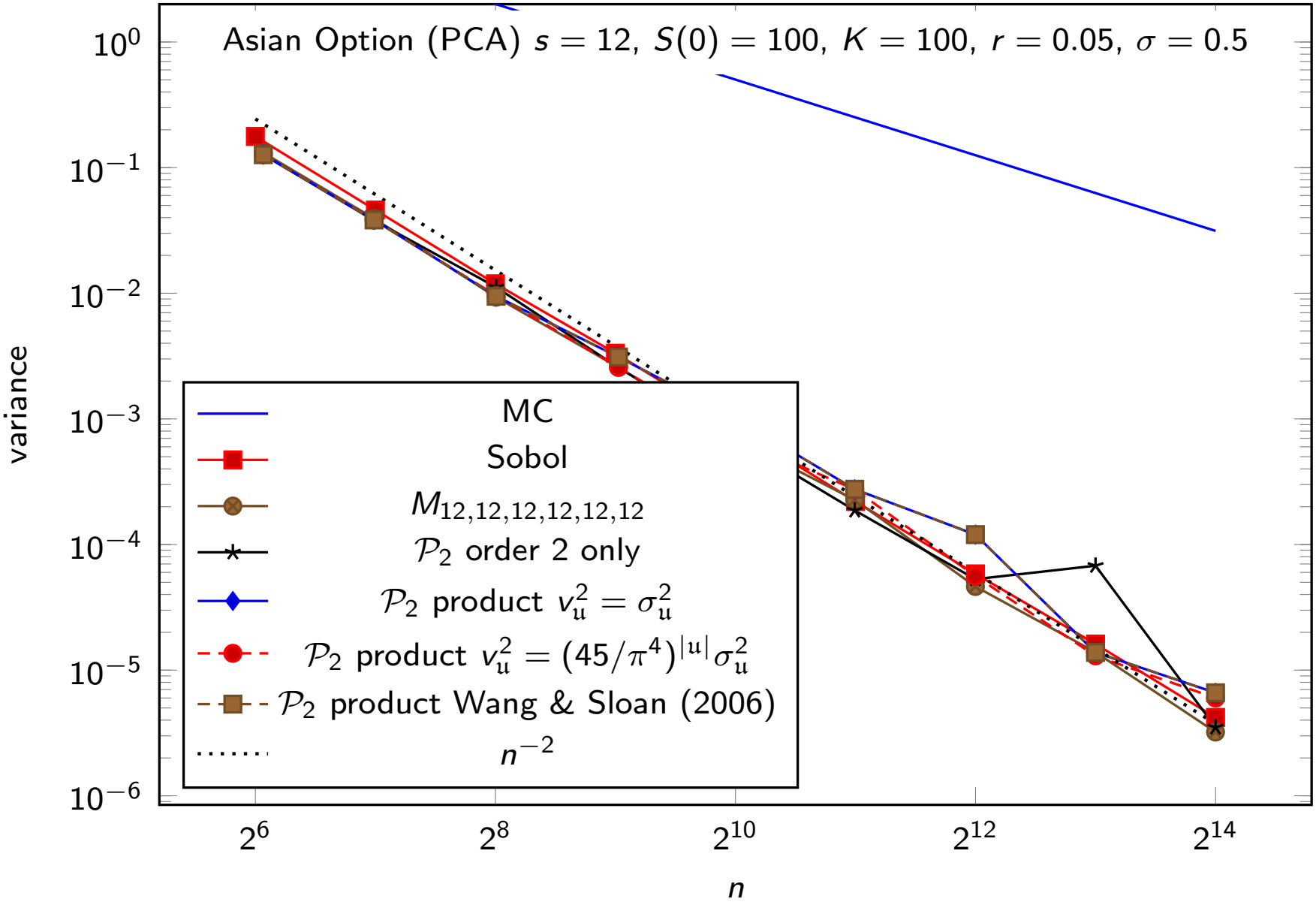
Lattices of Rank 1 with CBC



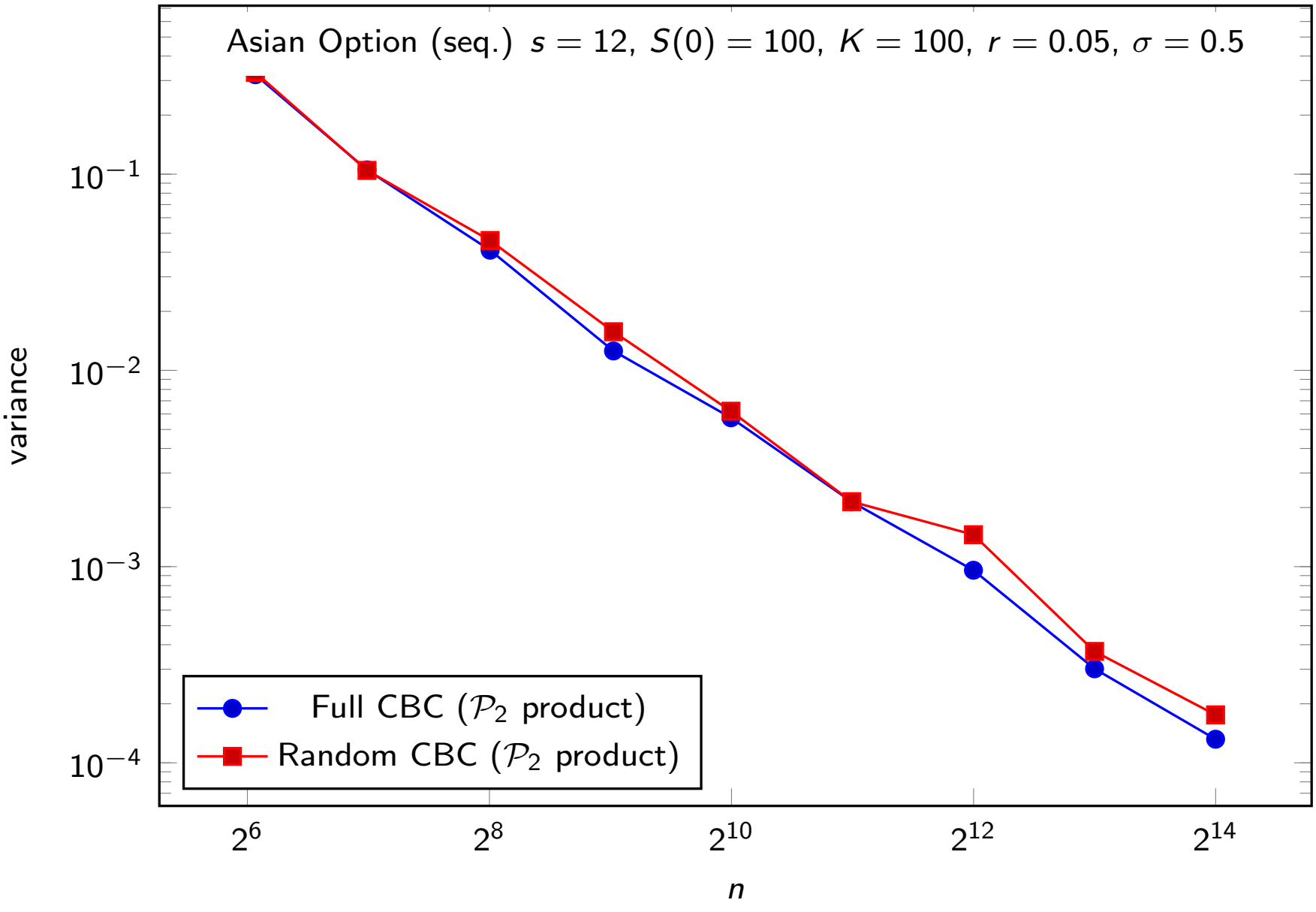
Lattices of Rank 1 with CBC



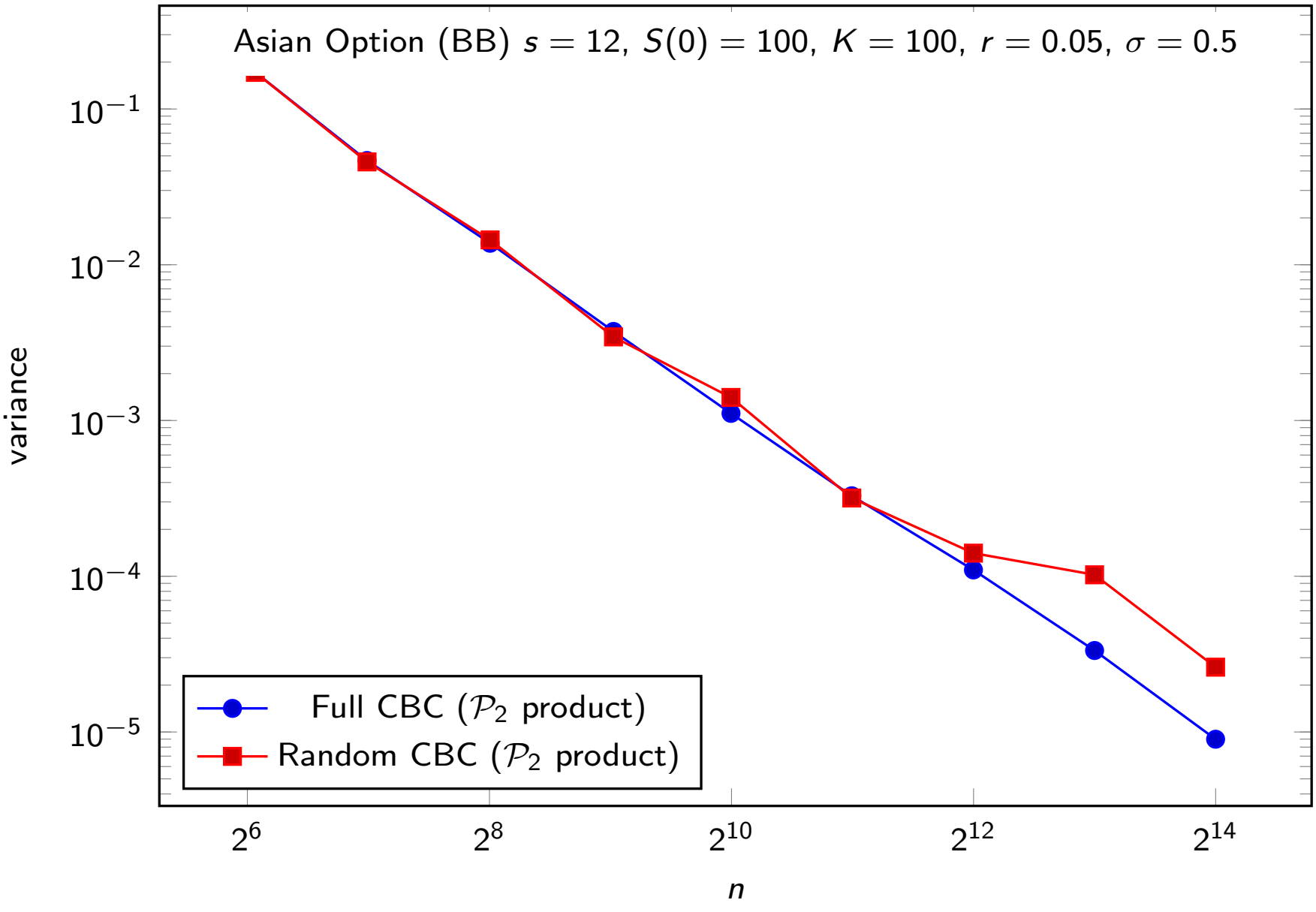
Lattices of Rank 1 with CBC



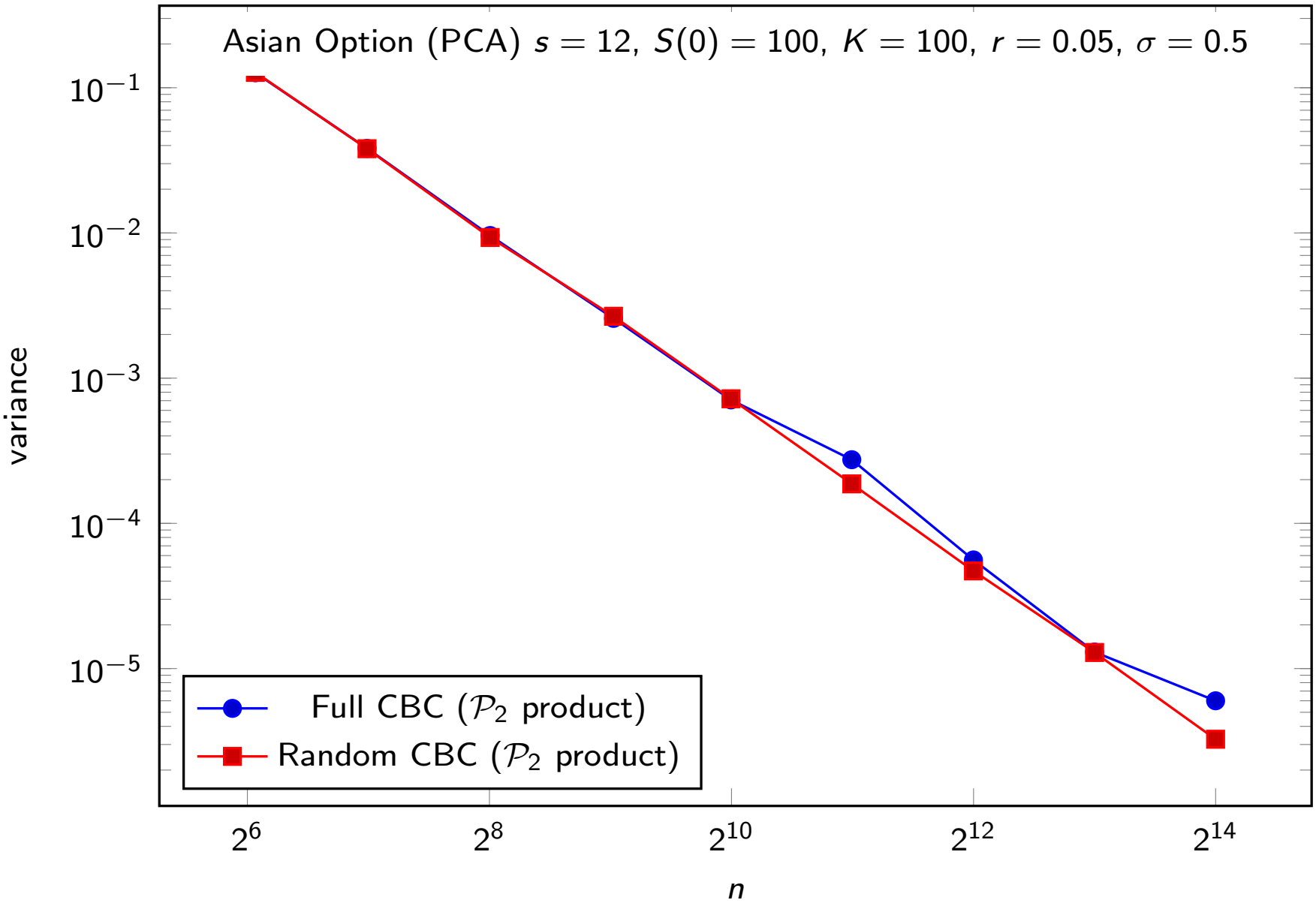
Random vs. Full CBC



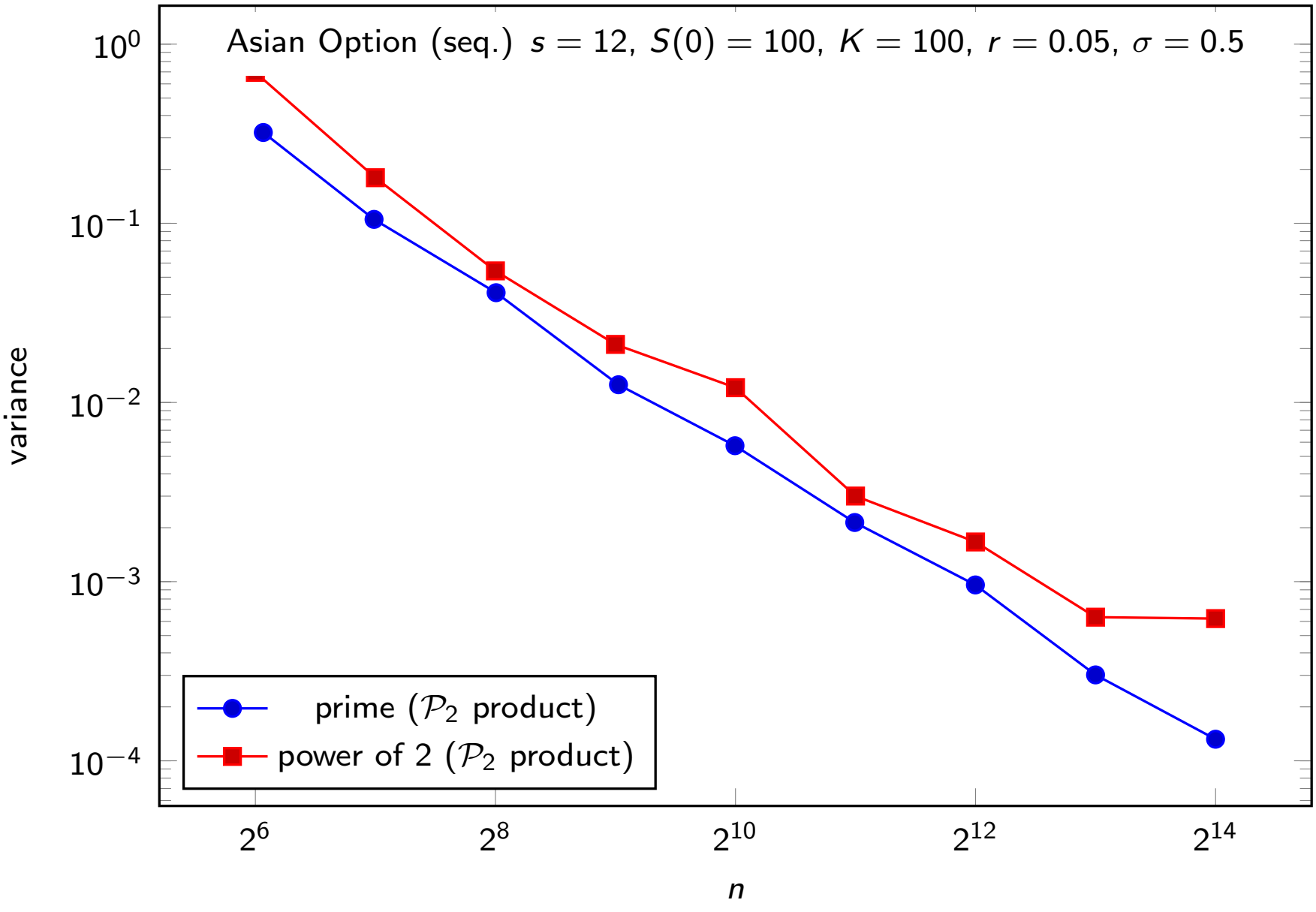
Random vs. Full CBC



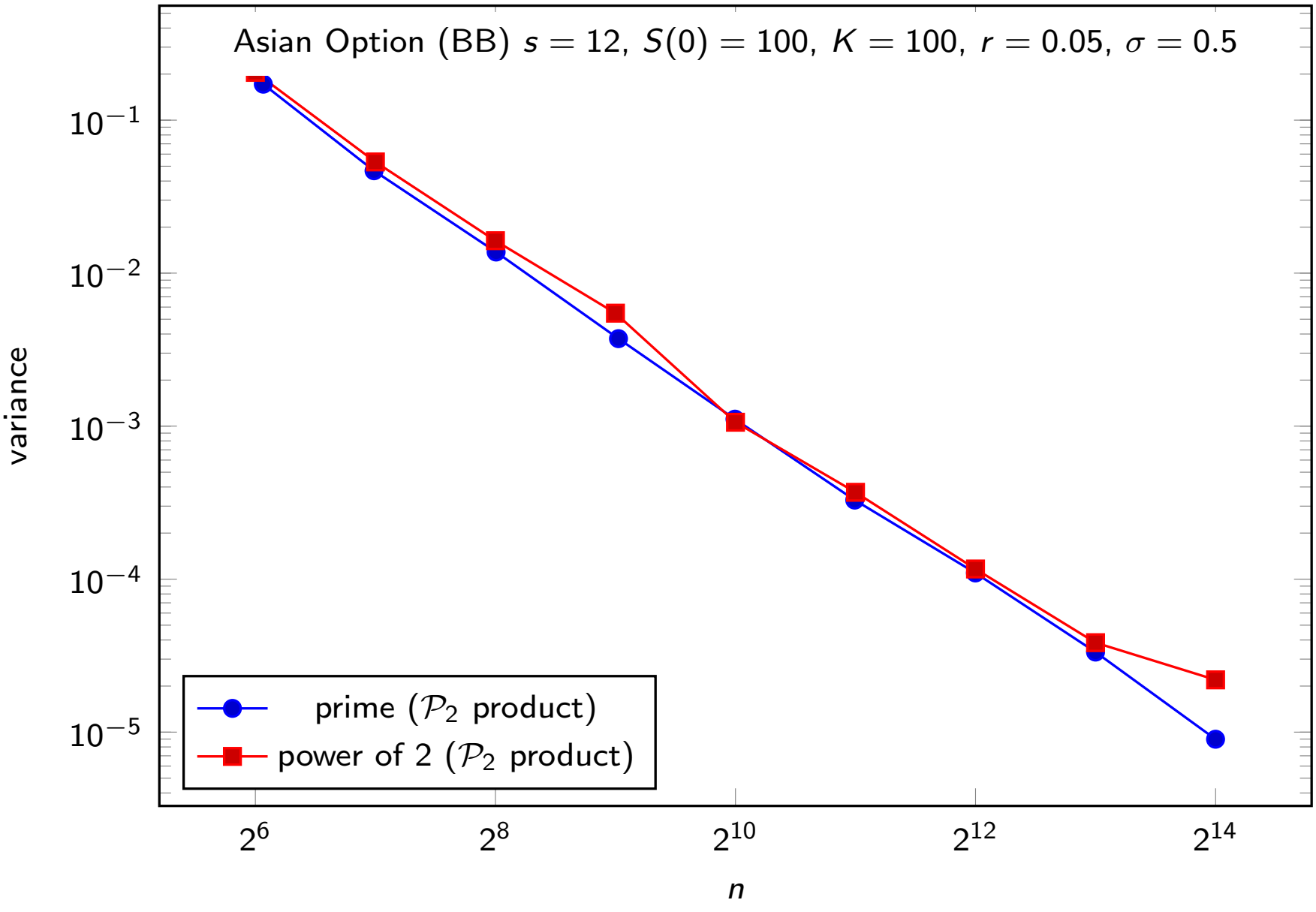
Random vs. Full CBC



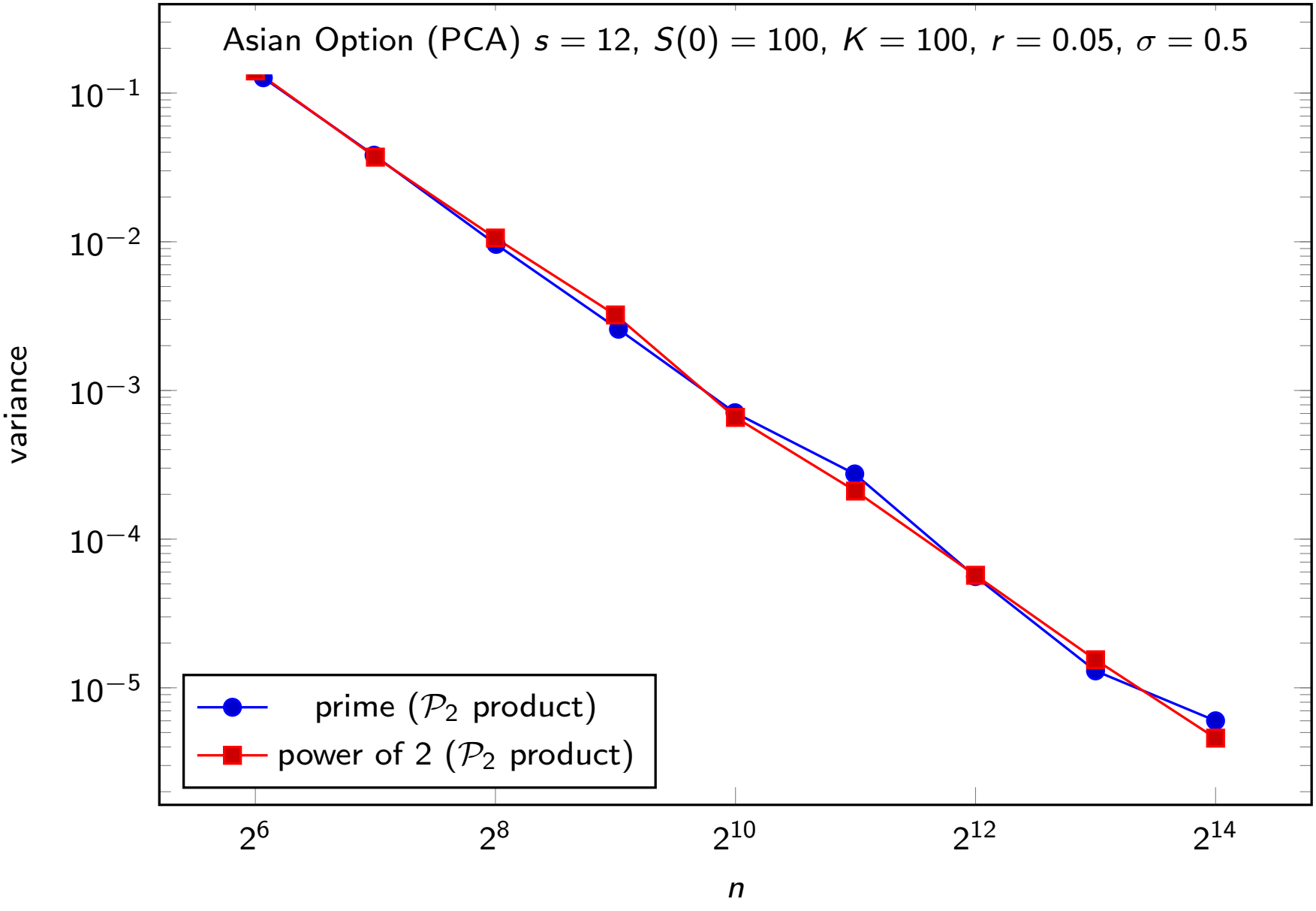
Prime vs. Power-of-2 Number of Points



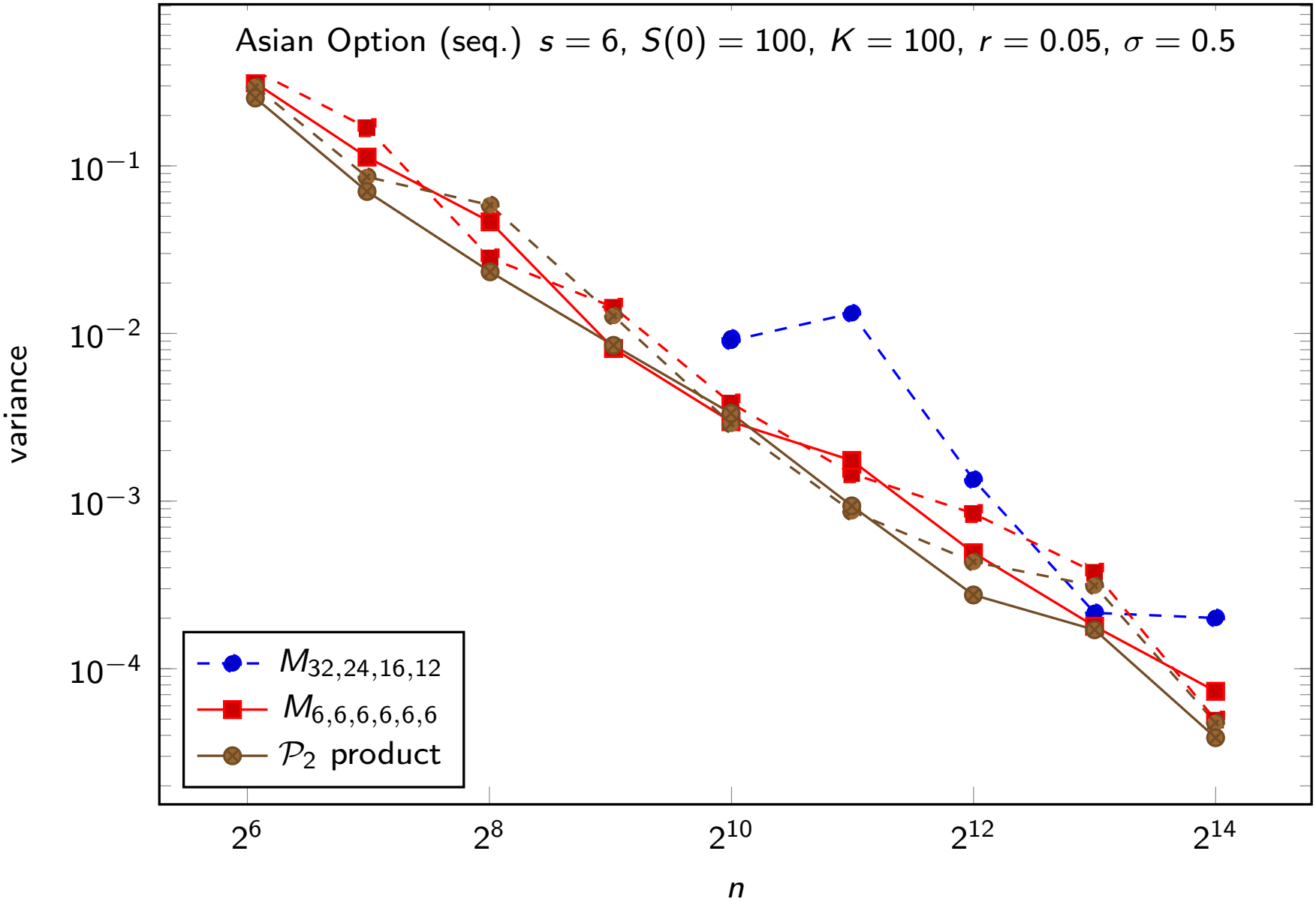
Prime vs. Power-of-2 Number of Points



Prime vs. Power-of-2 Number of Points

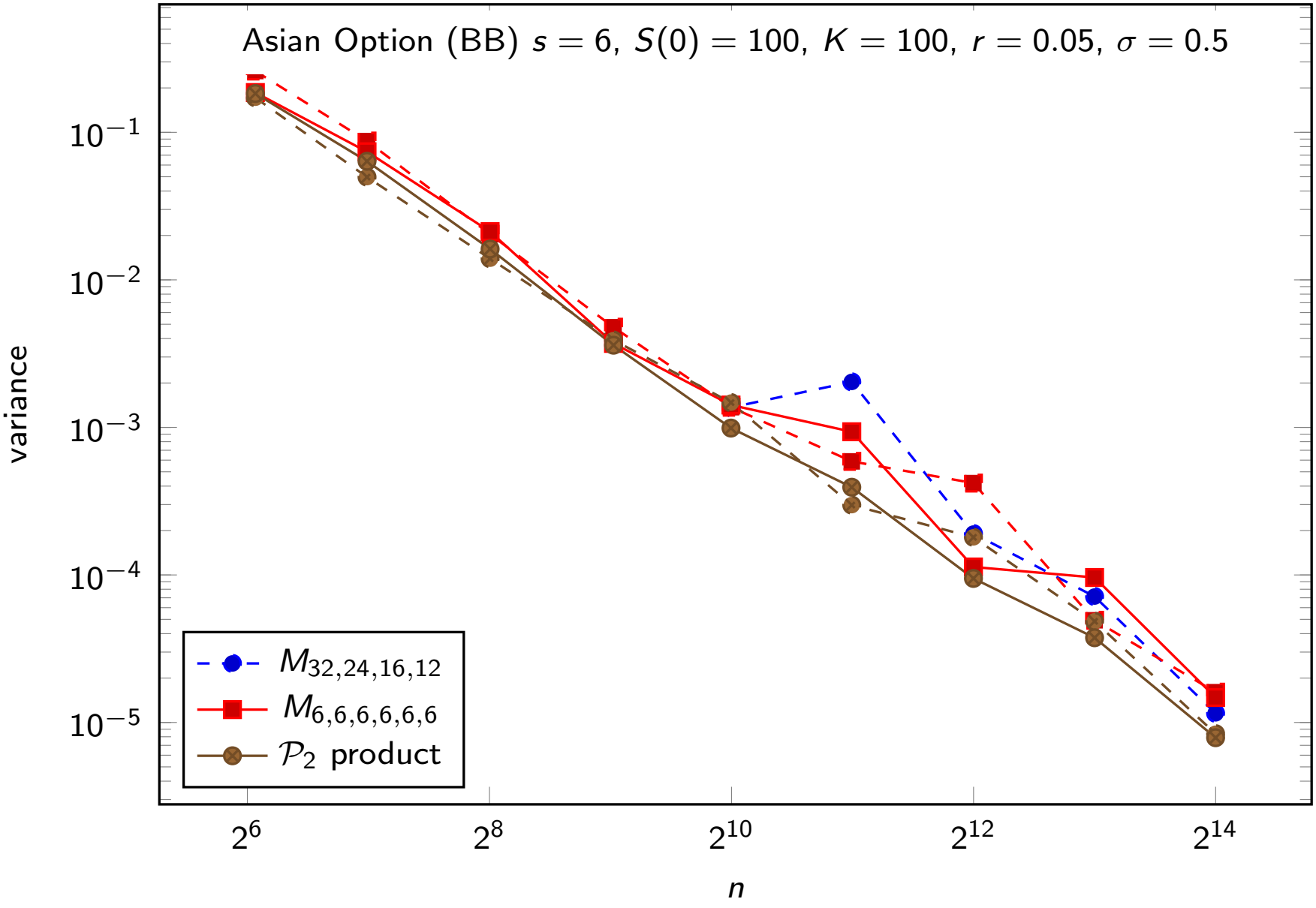


Korobov vs. CBC



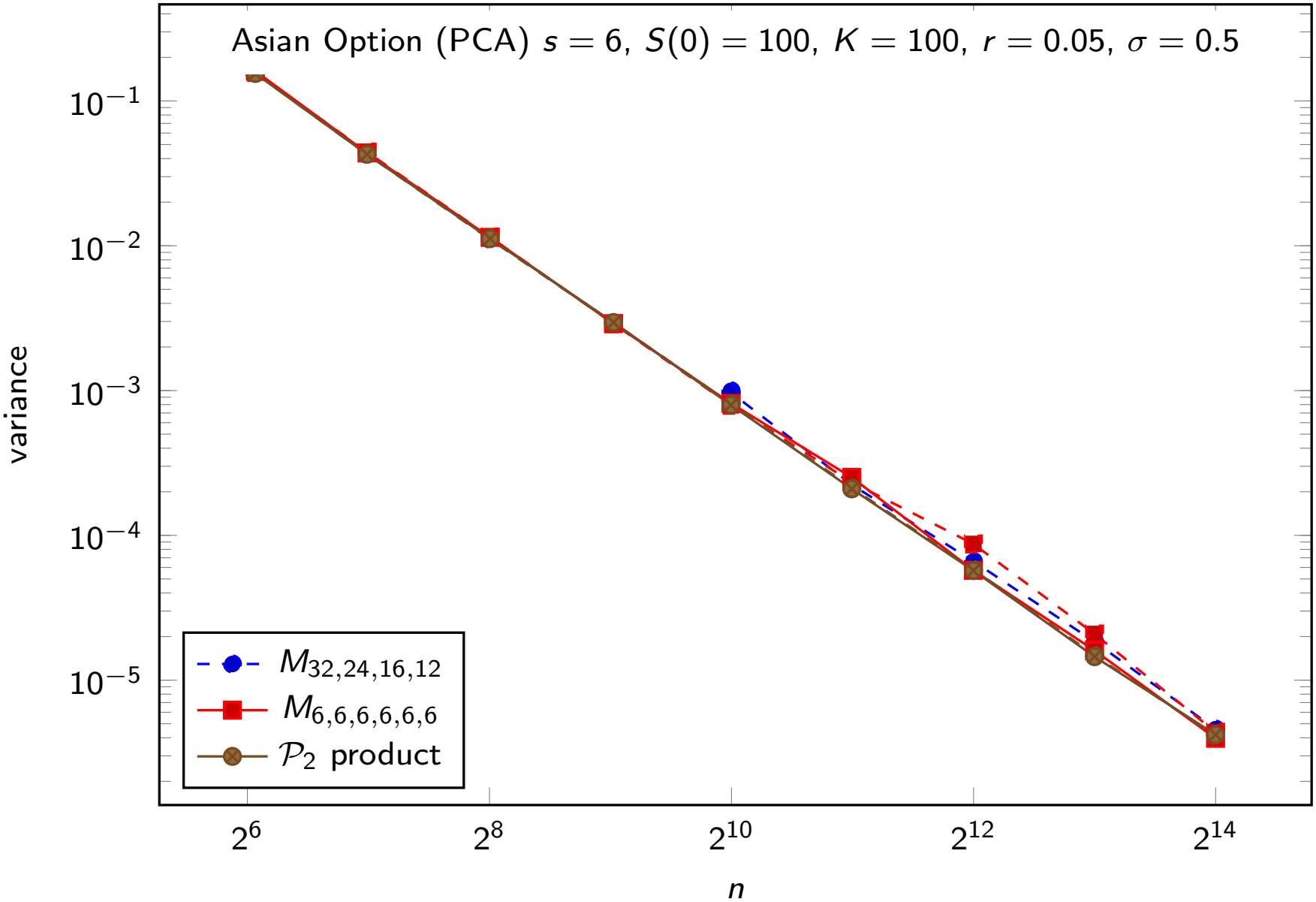
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



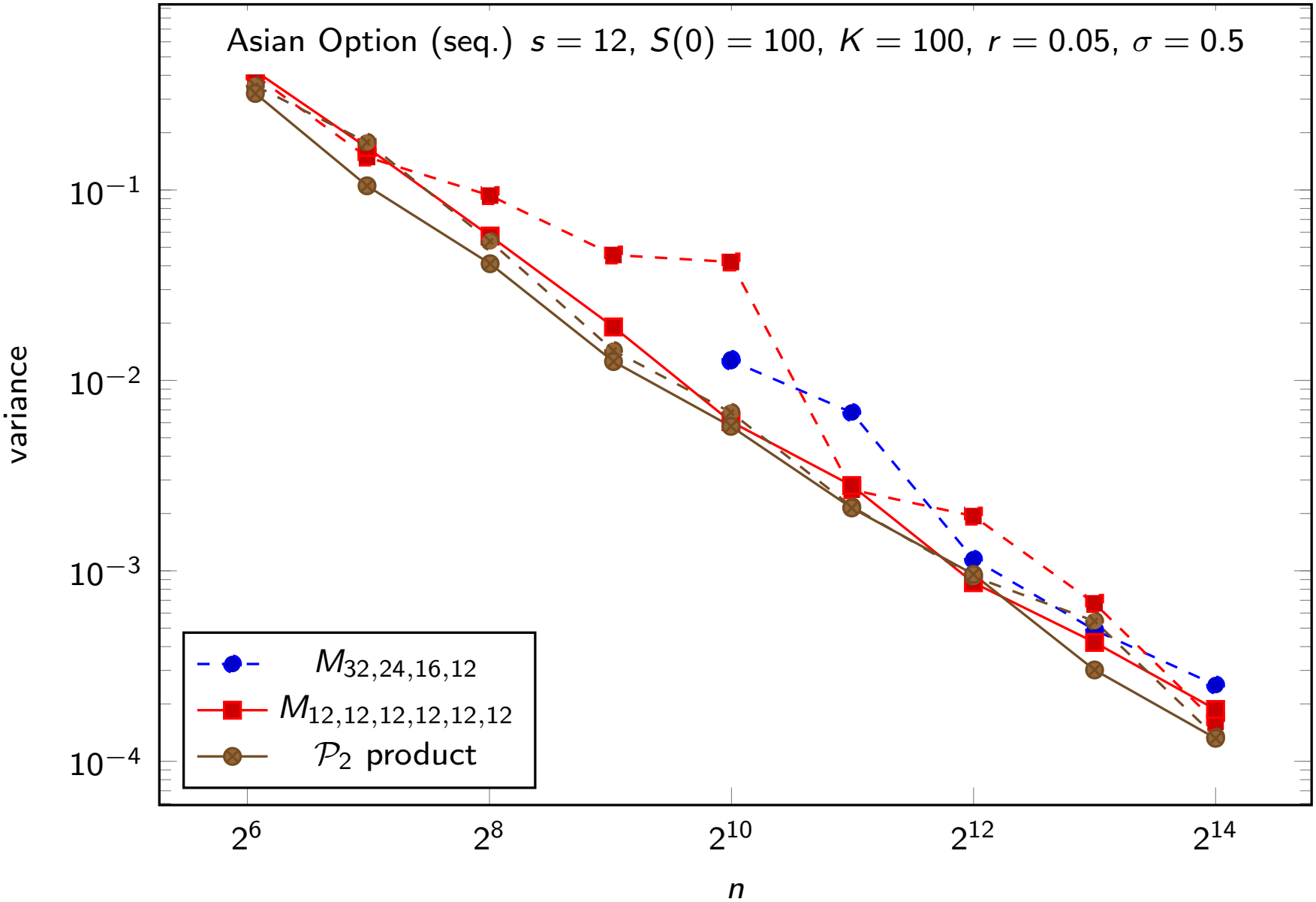
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



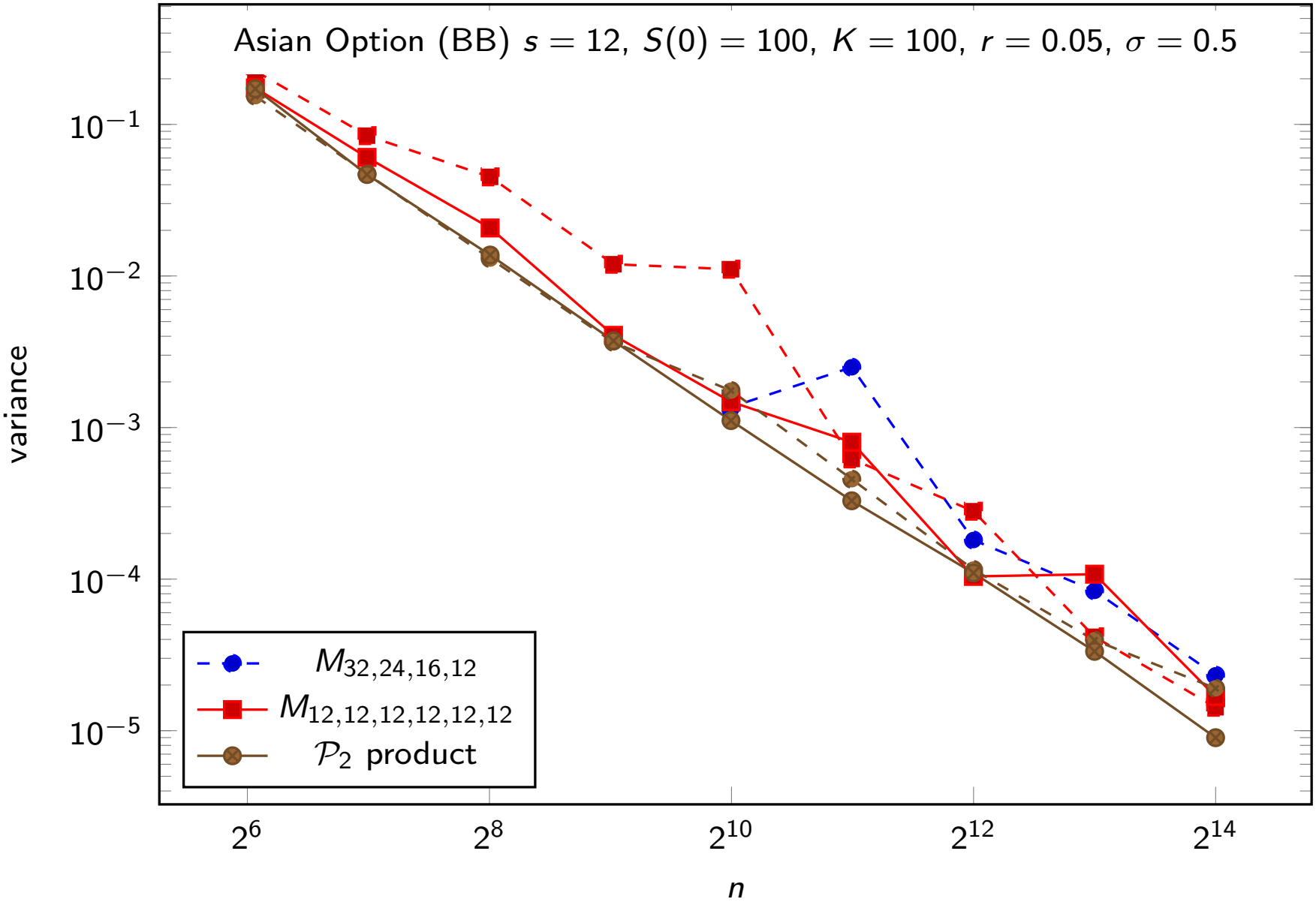
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



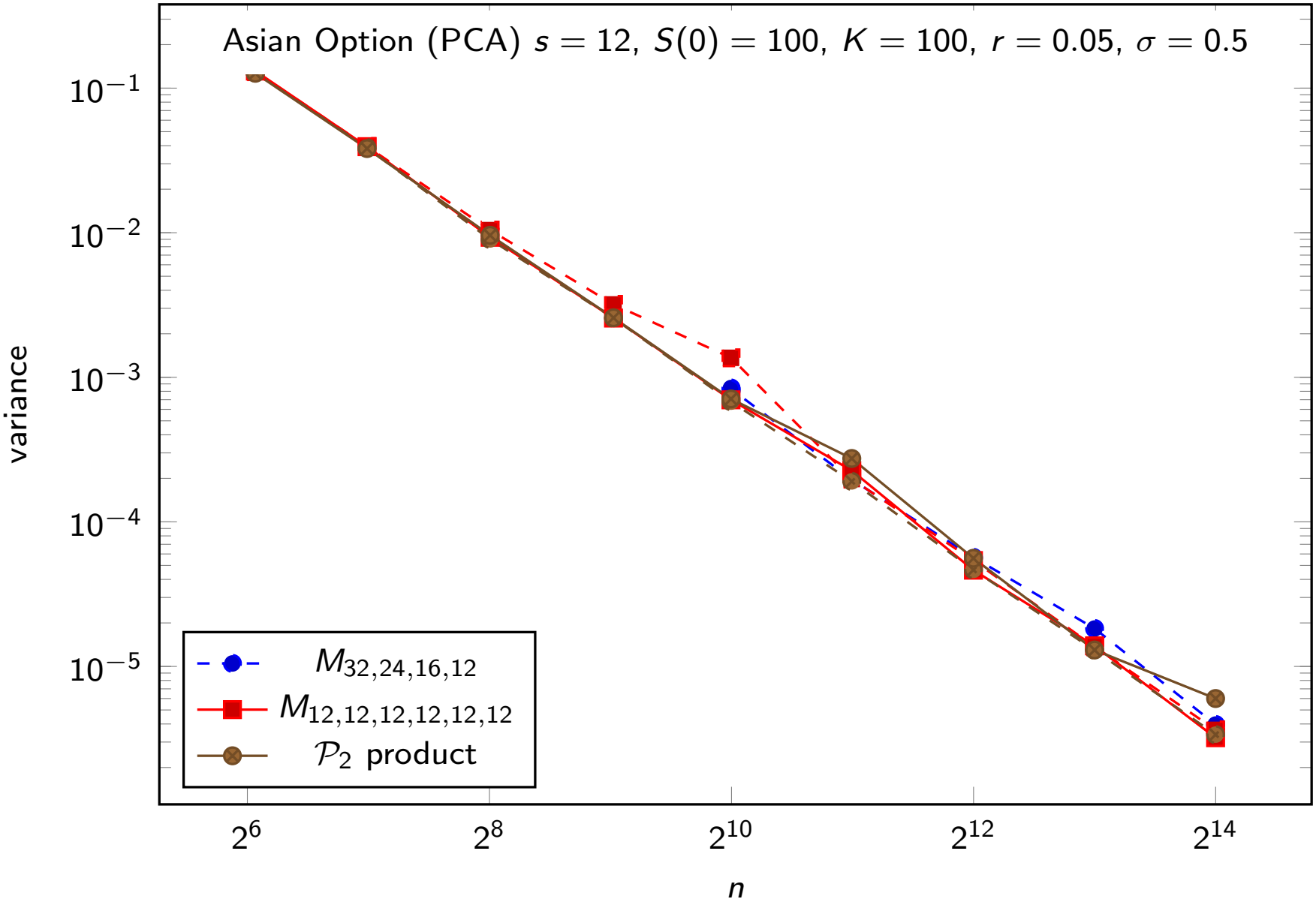
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



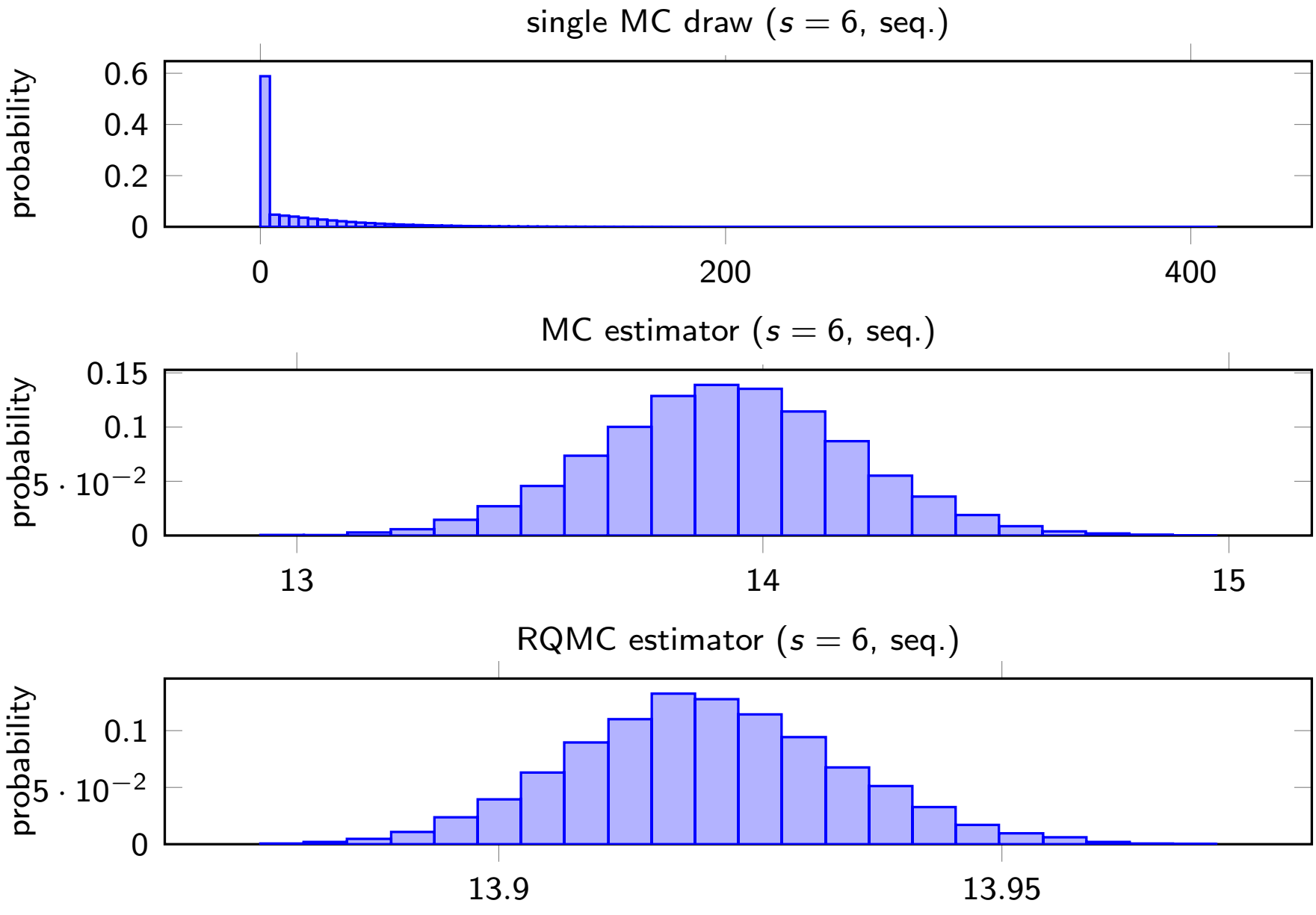
Solid: CBC. Dashed: Korobov.

Korobov vs. CBC

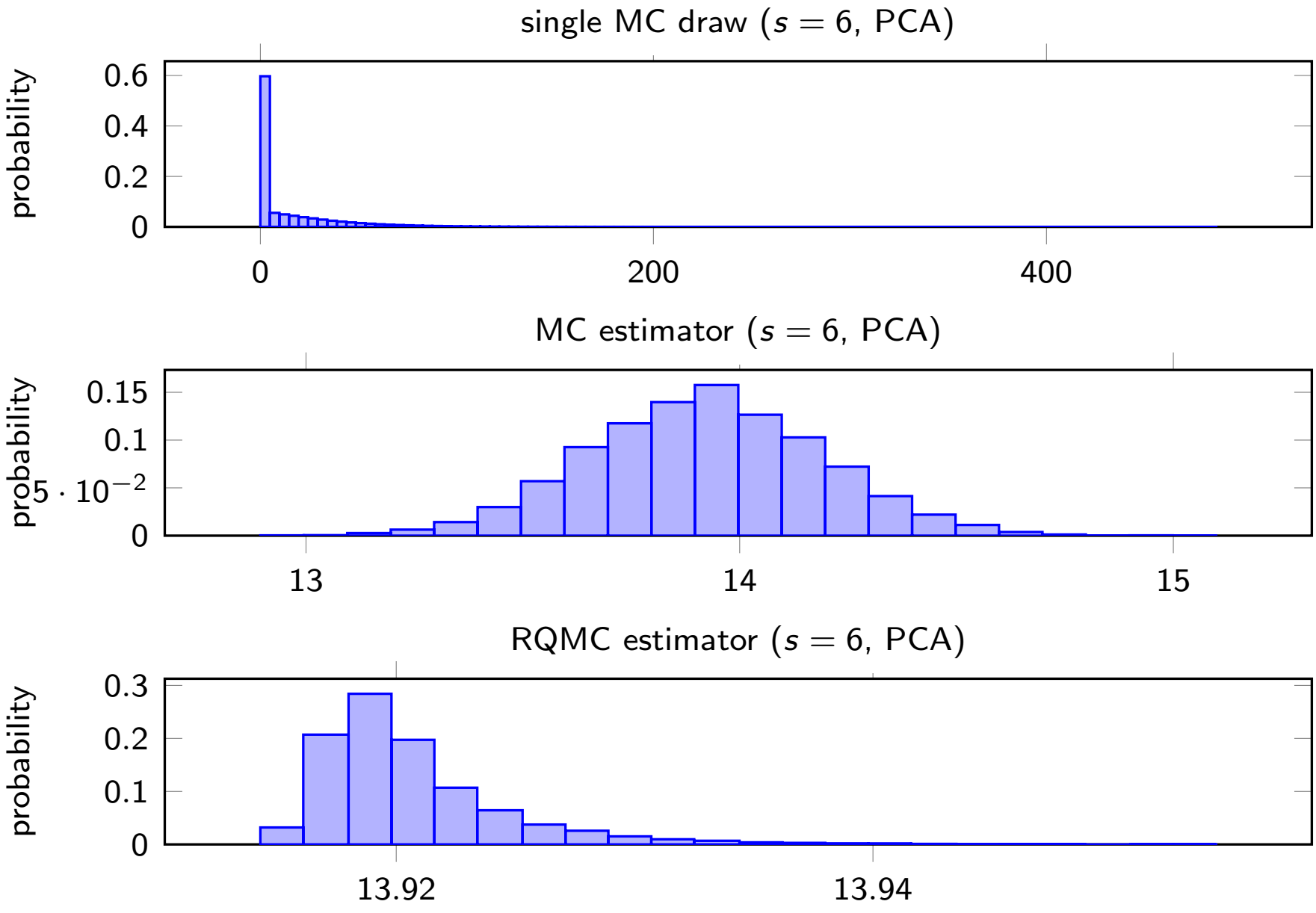


Solid: CBC. Dashed: Korobov.

Histograms for the Asian Option, $s = 6$, sequential



Histograms for the Asian option, $s = 6$, PCA

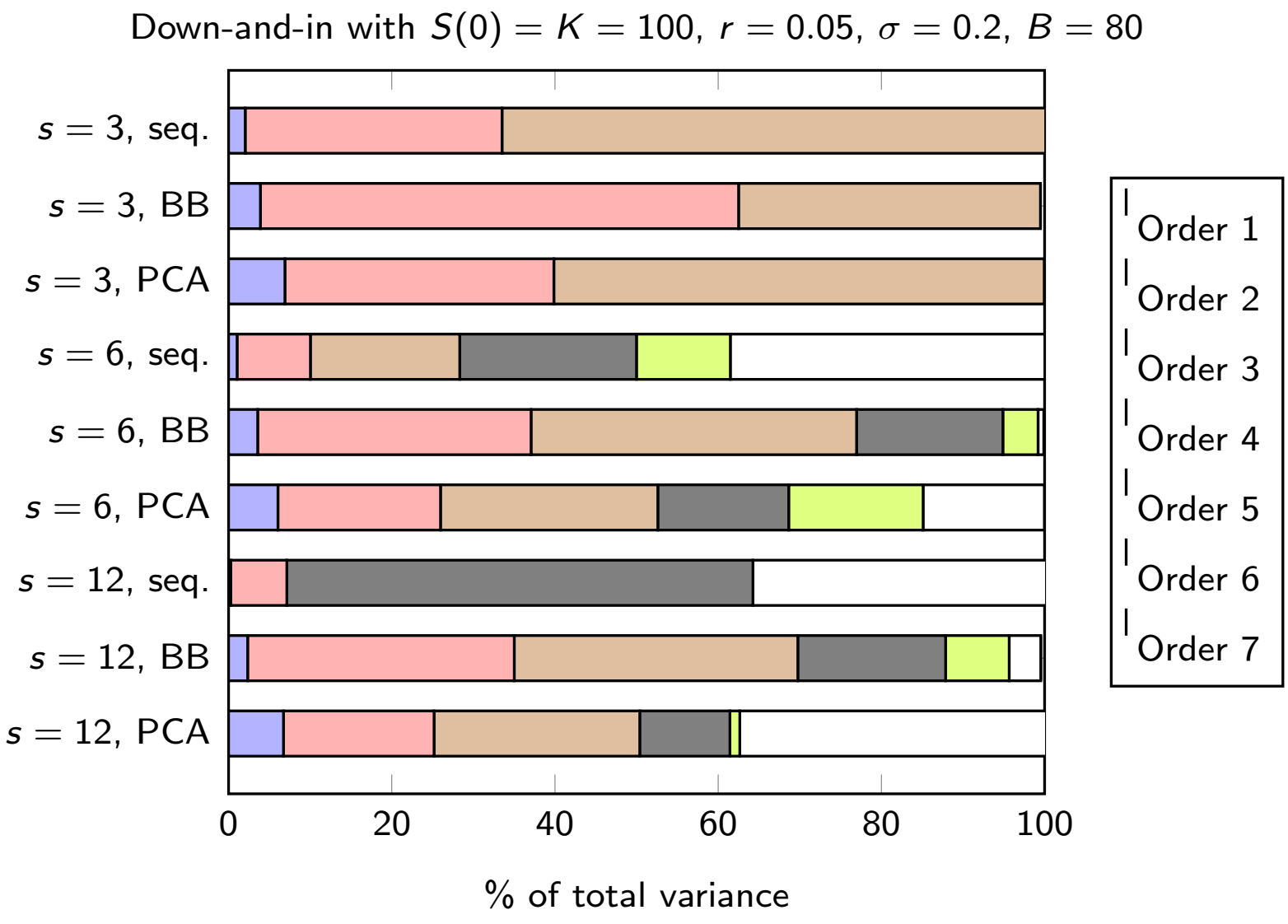


A down-and-in Asian option with barrier B

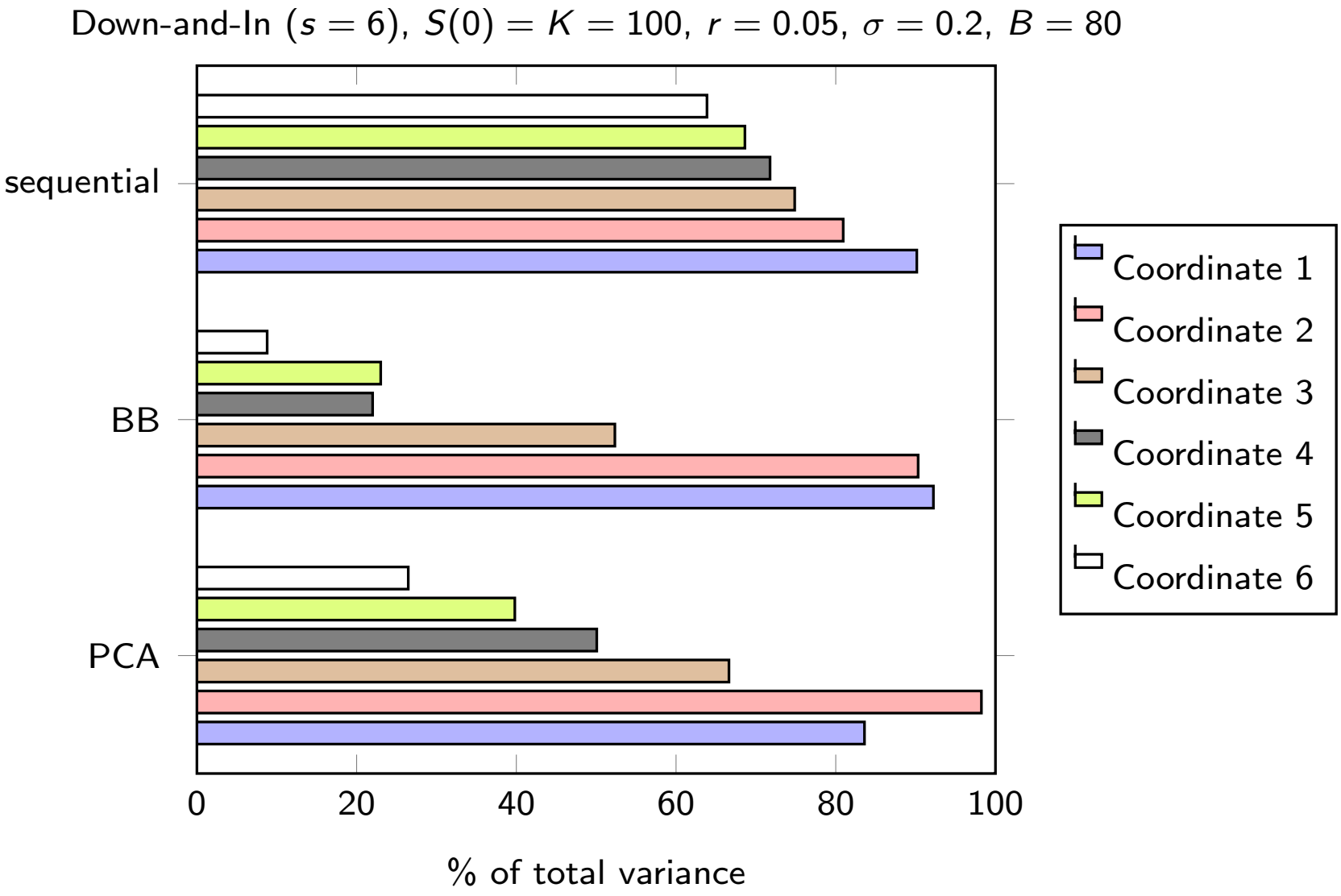
Same as for Asian option, except that payoff is zero unless

$$\min_{1 \le j \le s} S(t_j) \le 80.$$

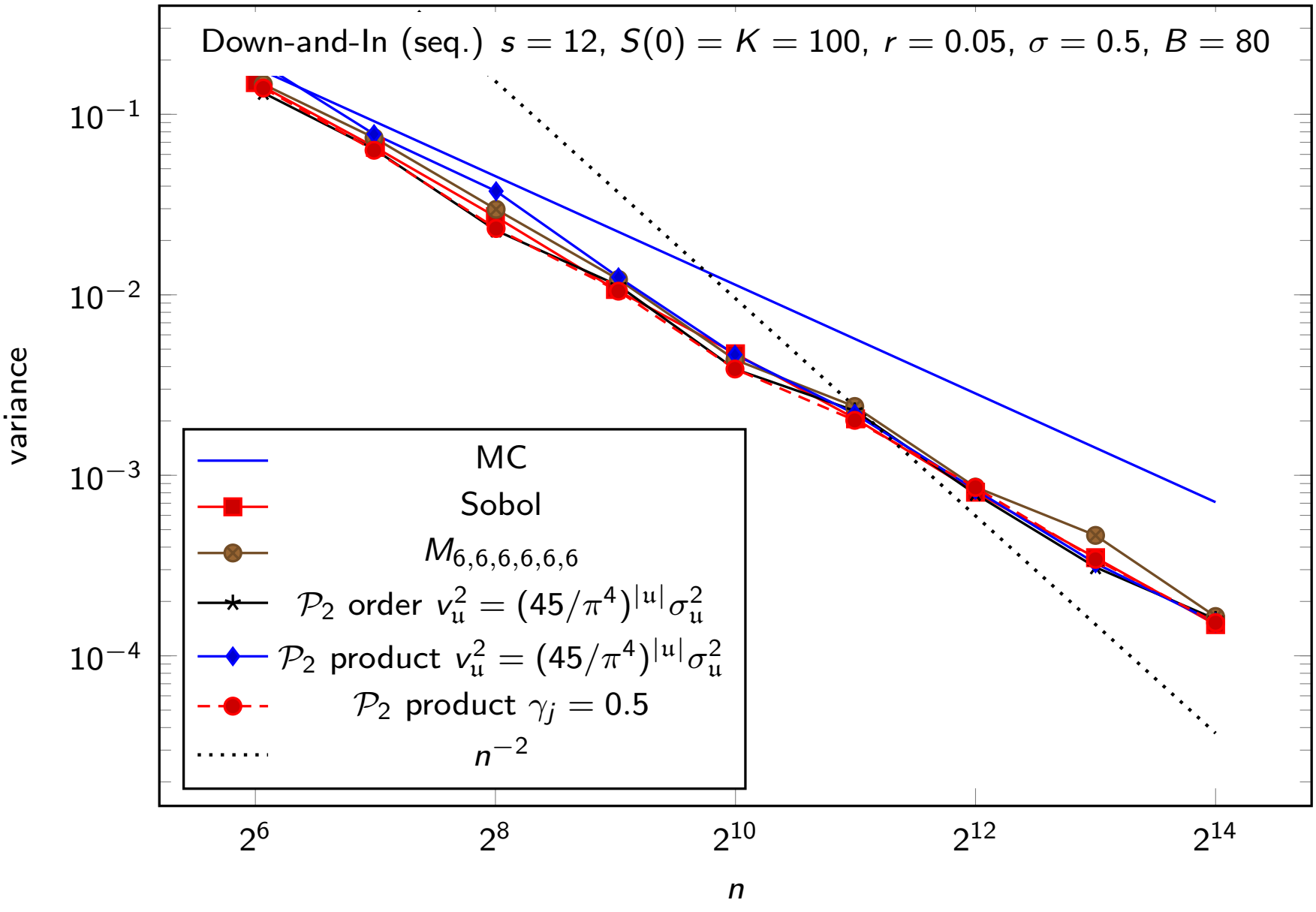
ANOVA Variances for the down-and-in Asian Option



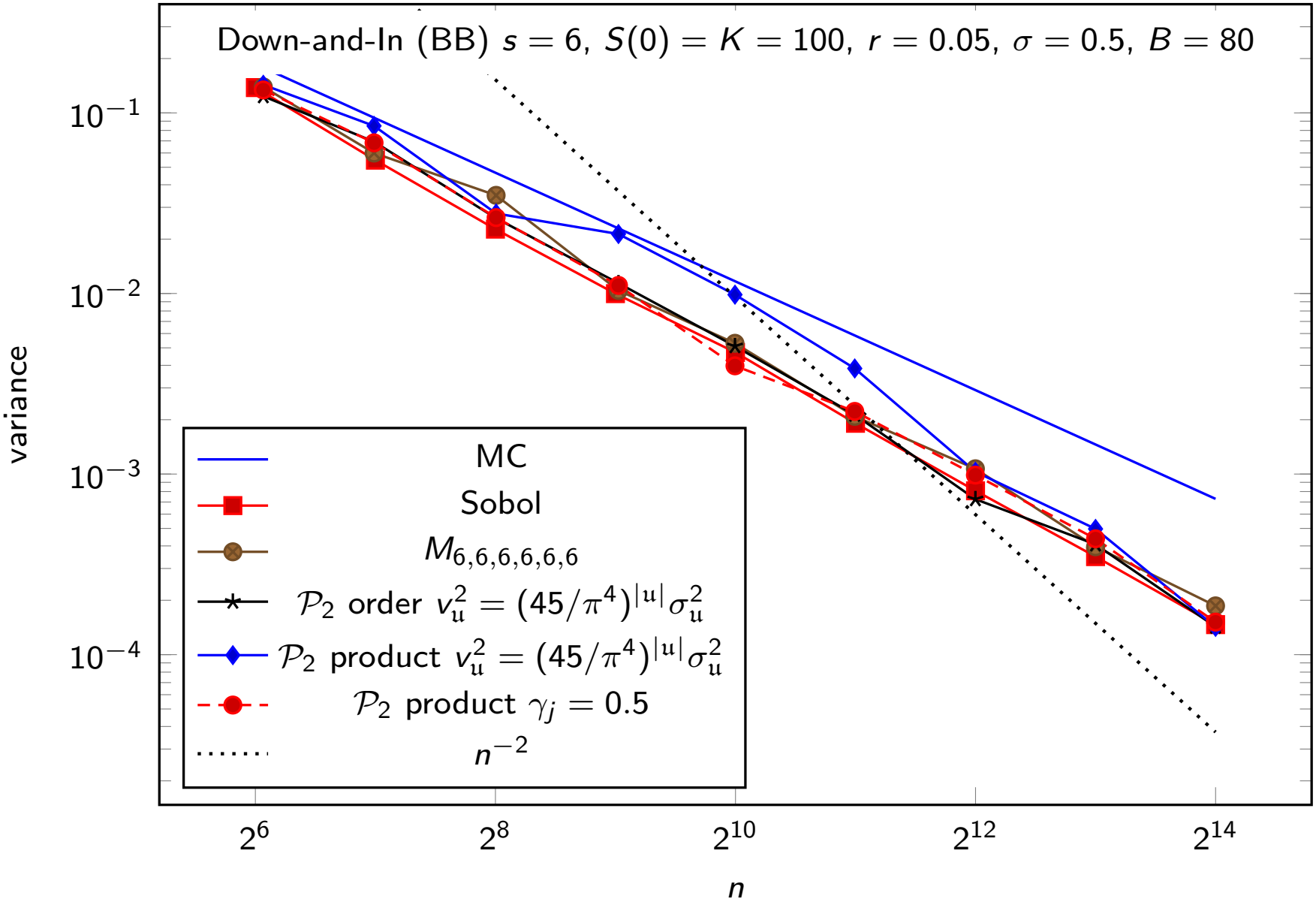
Total Variance per Coordinate for the down-and-in Asian Option



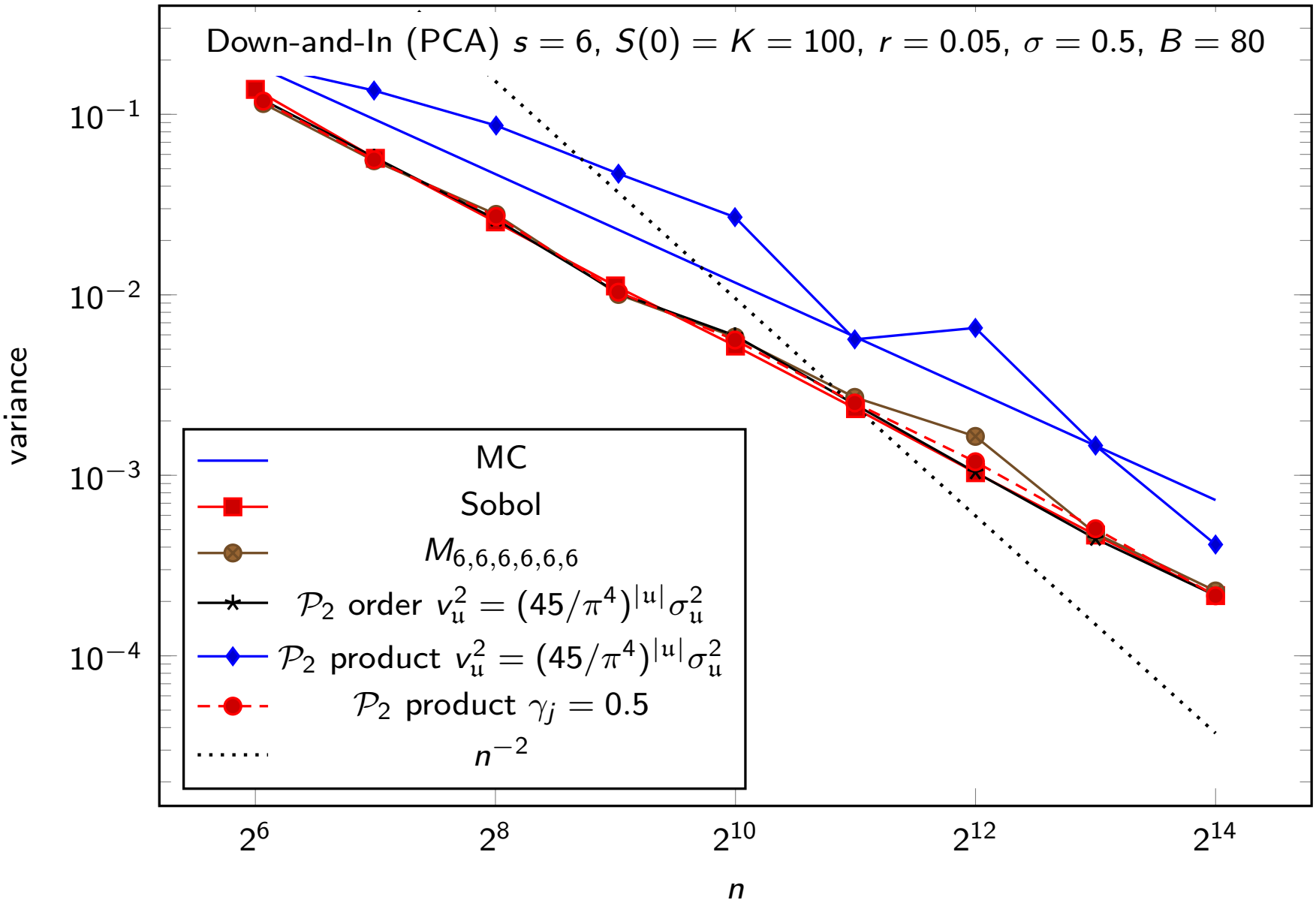
Lattices of Rank 1 with CBC



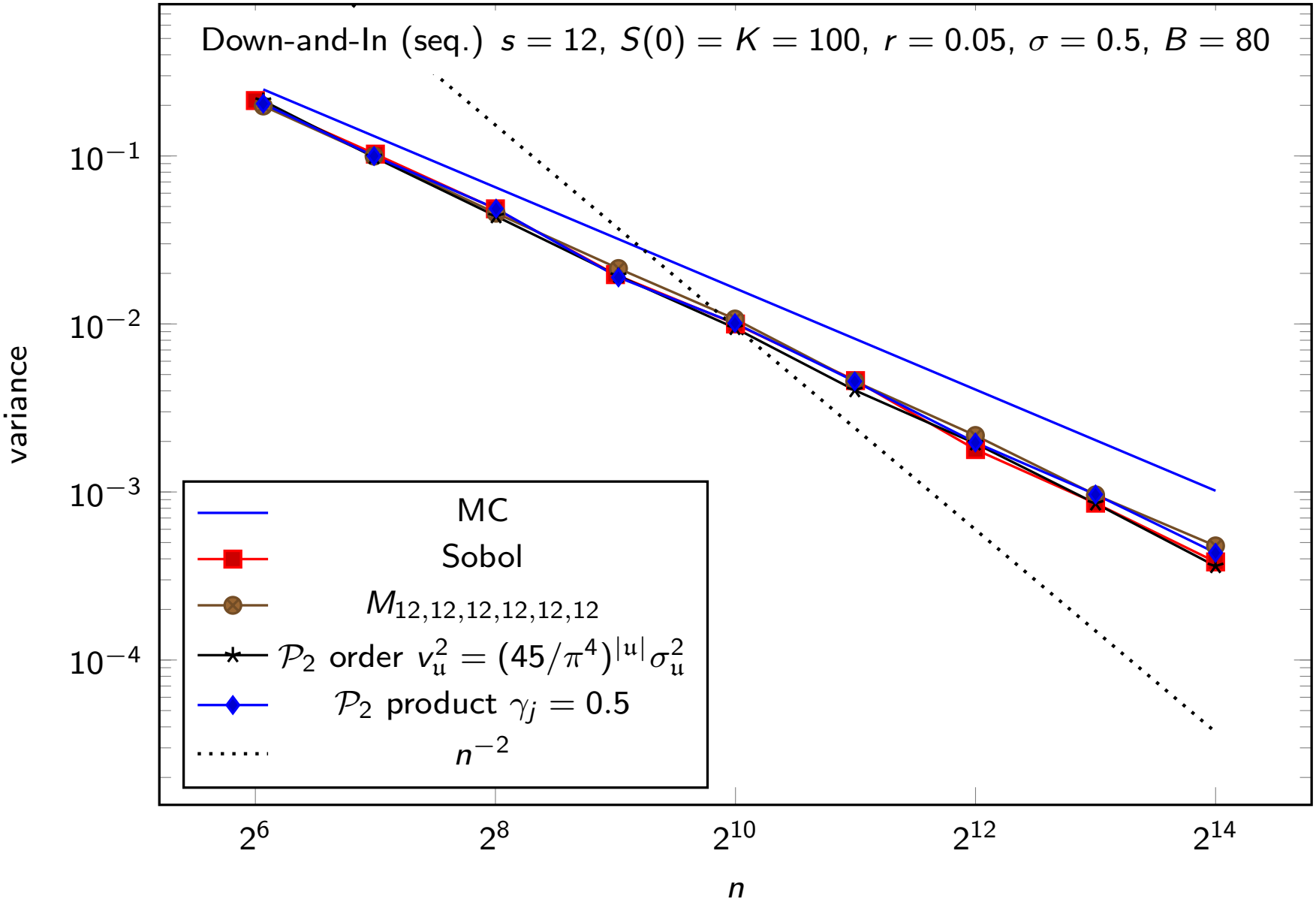
Lattices of Rank 1 with CBC



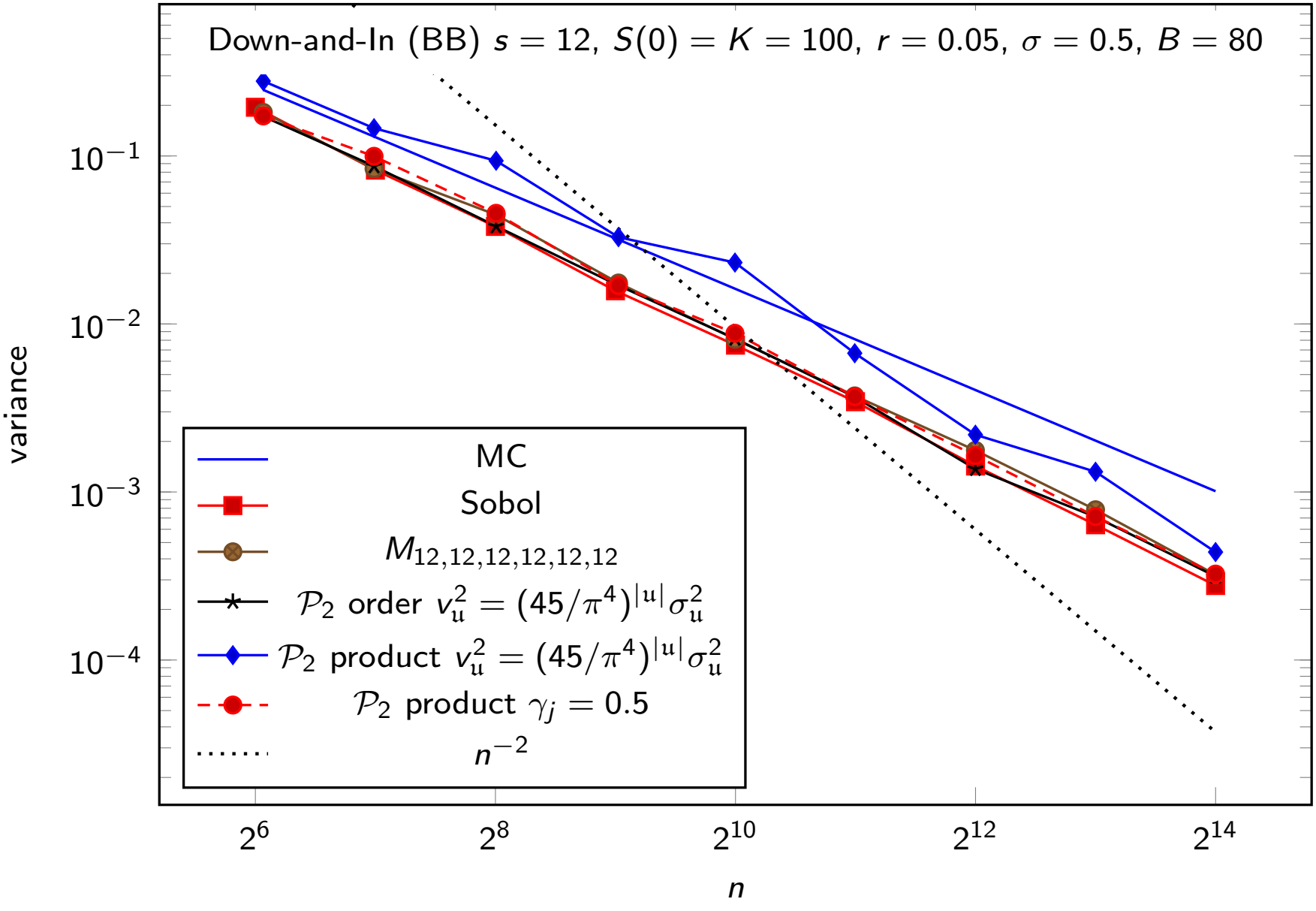
Lattices of Rank 1 with CBC



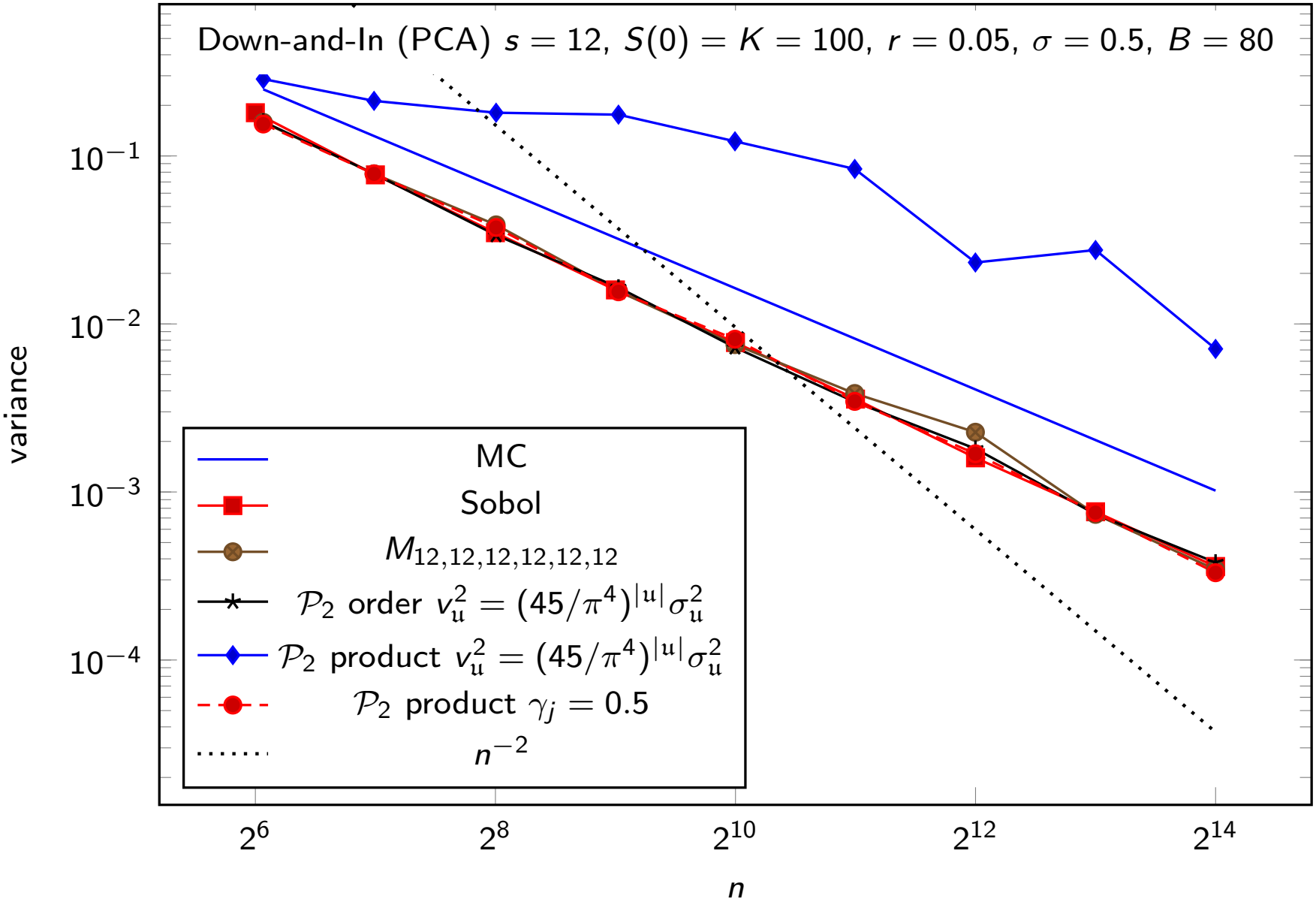
Lattices of Rank 1 with CBC



Lattices of Rank 1 with CBC



Lattices of Rank 1 with CBC

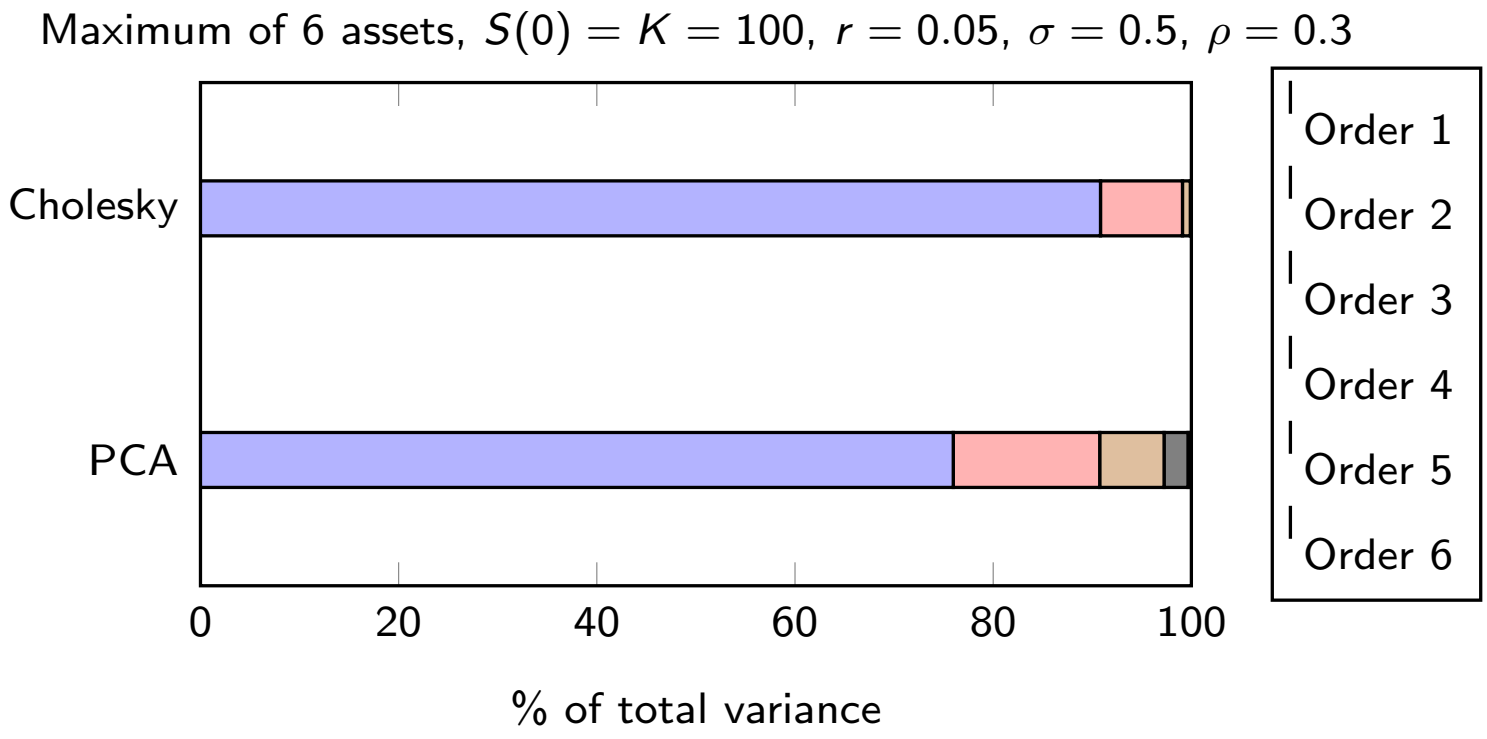


Call on the maximum of 6 assets

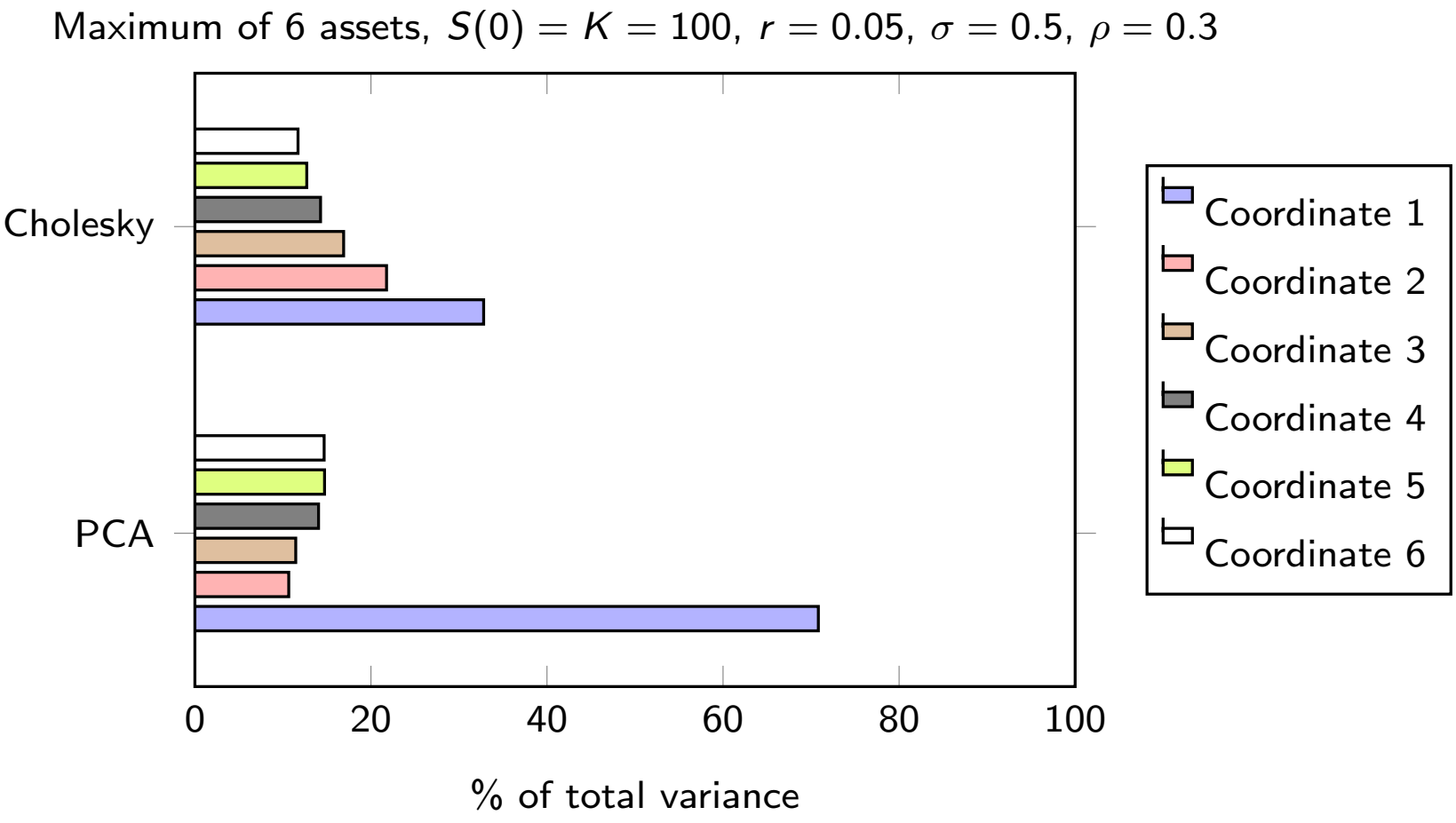
Each of 6 asset prices obeys a GBM with $s_0 = 100$, $r = 0.05$, $\sigma = 0.2$. The pairwise correlation between Brownian motions is 0.3.

The assets pay a dividend at rate 0.10, which means that the effective risk-free rate can be taken as $r' = 0.05 - 0.10 = -0.05$.

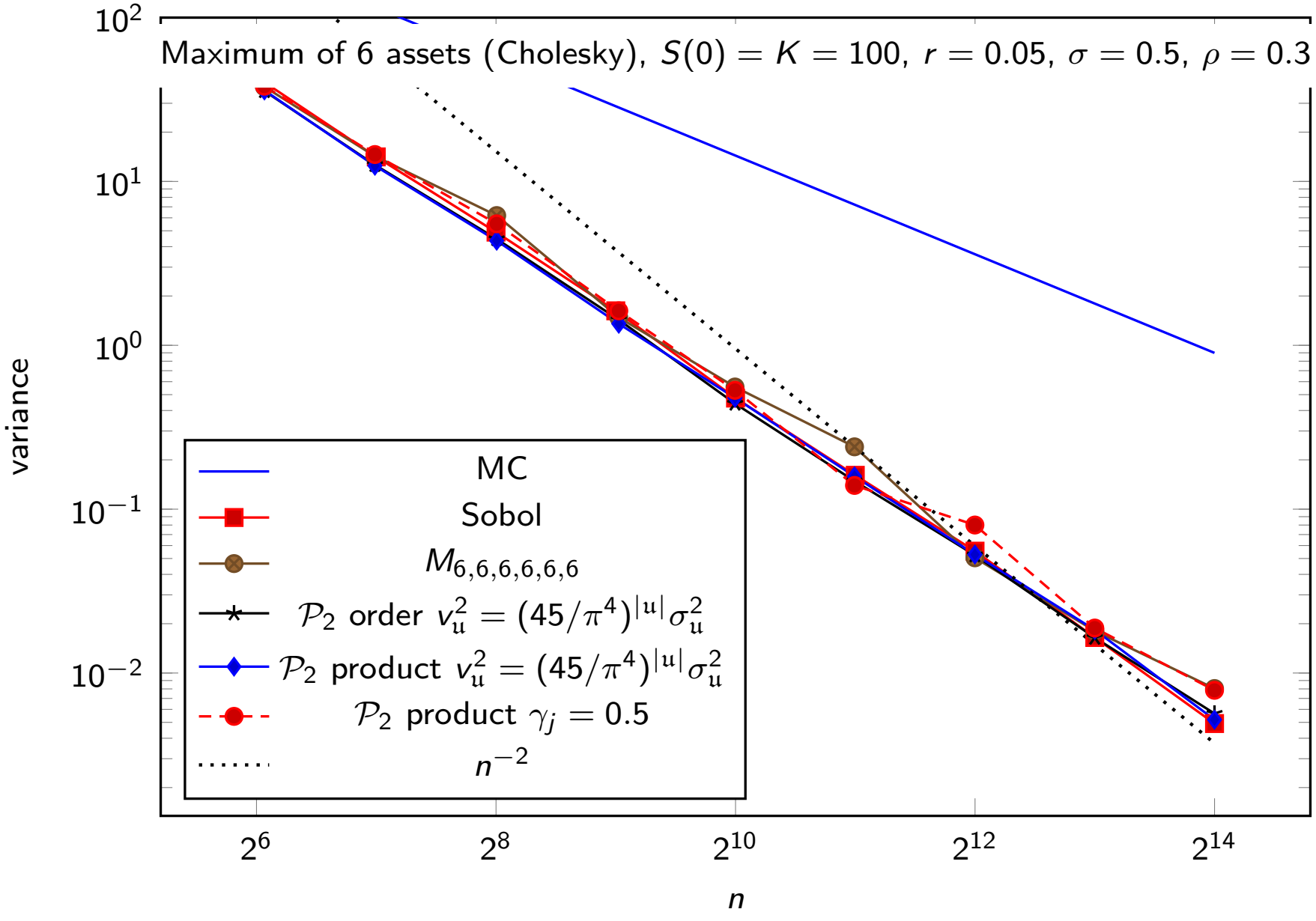
ANOVA variances for the maximum of 6 assets



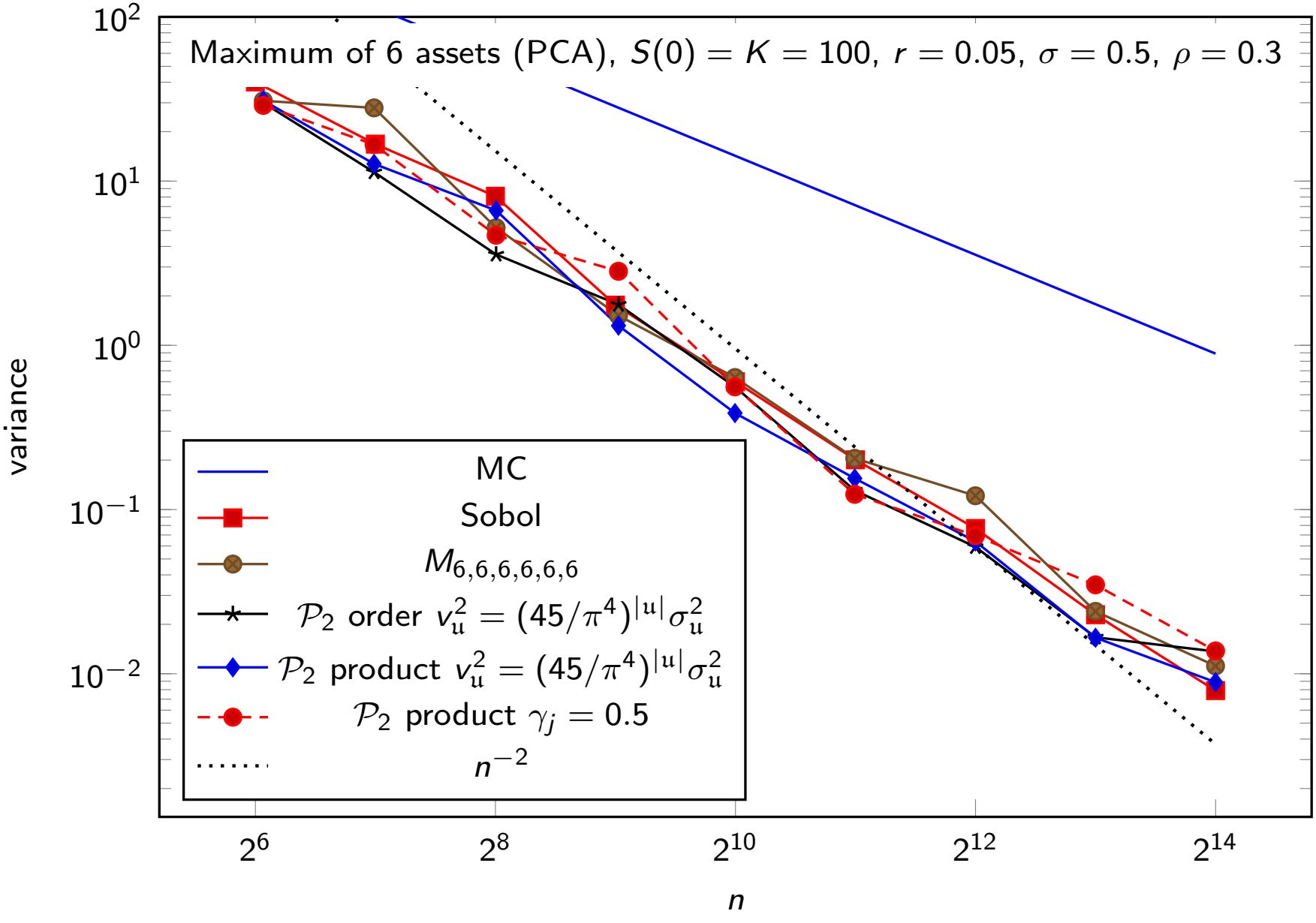
Total Variance per Coordinate for max of 6 assets



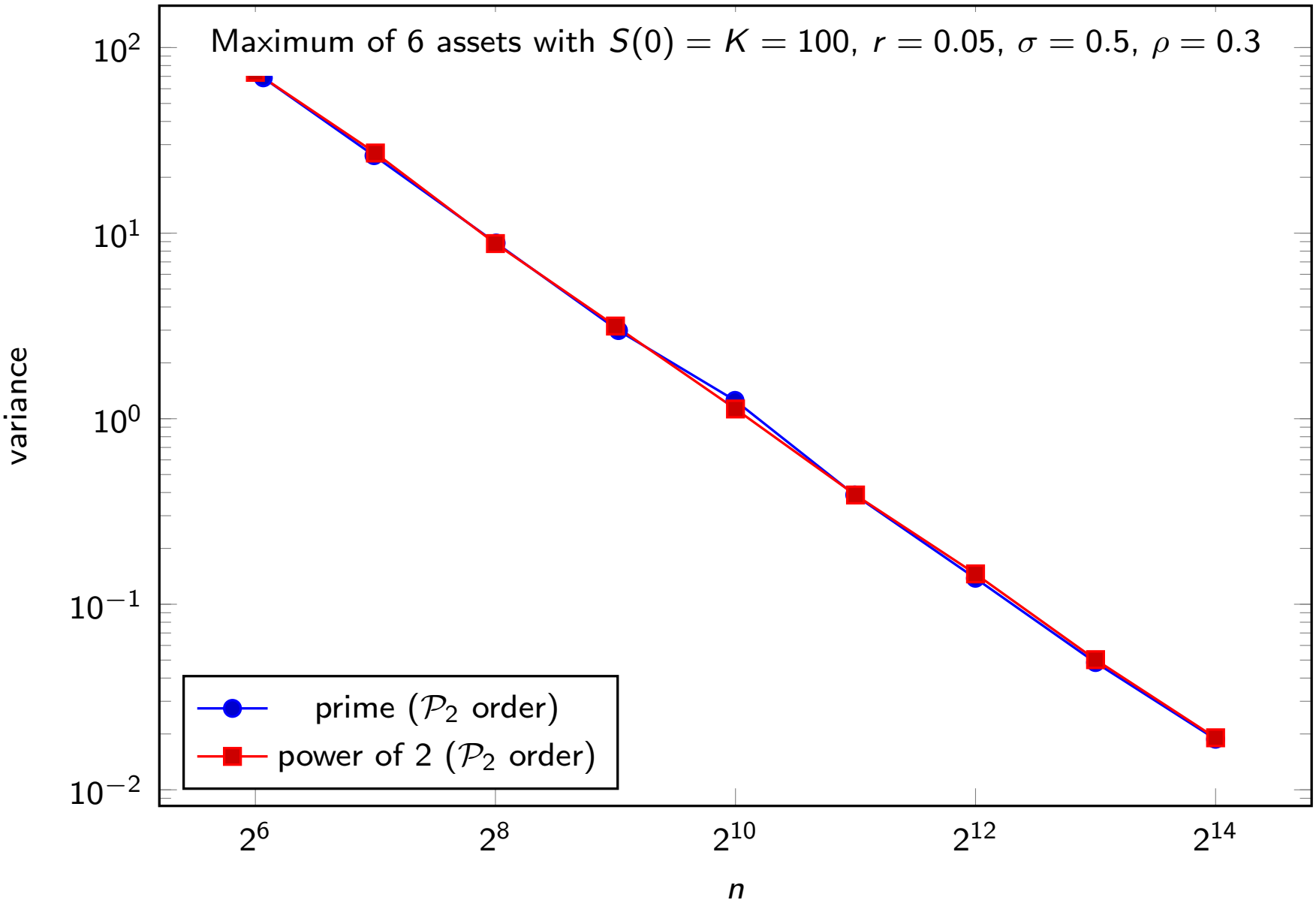
Lattices of Rank 1 with CBC



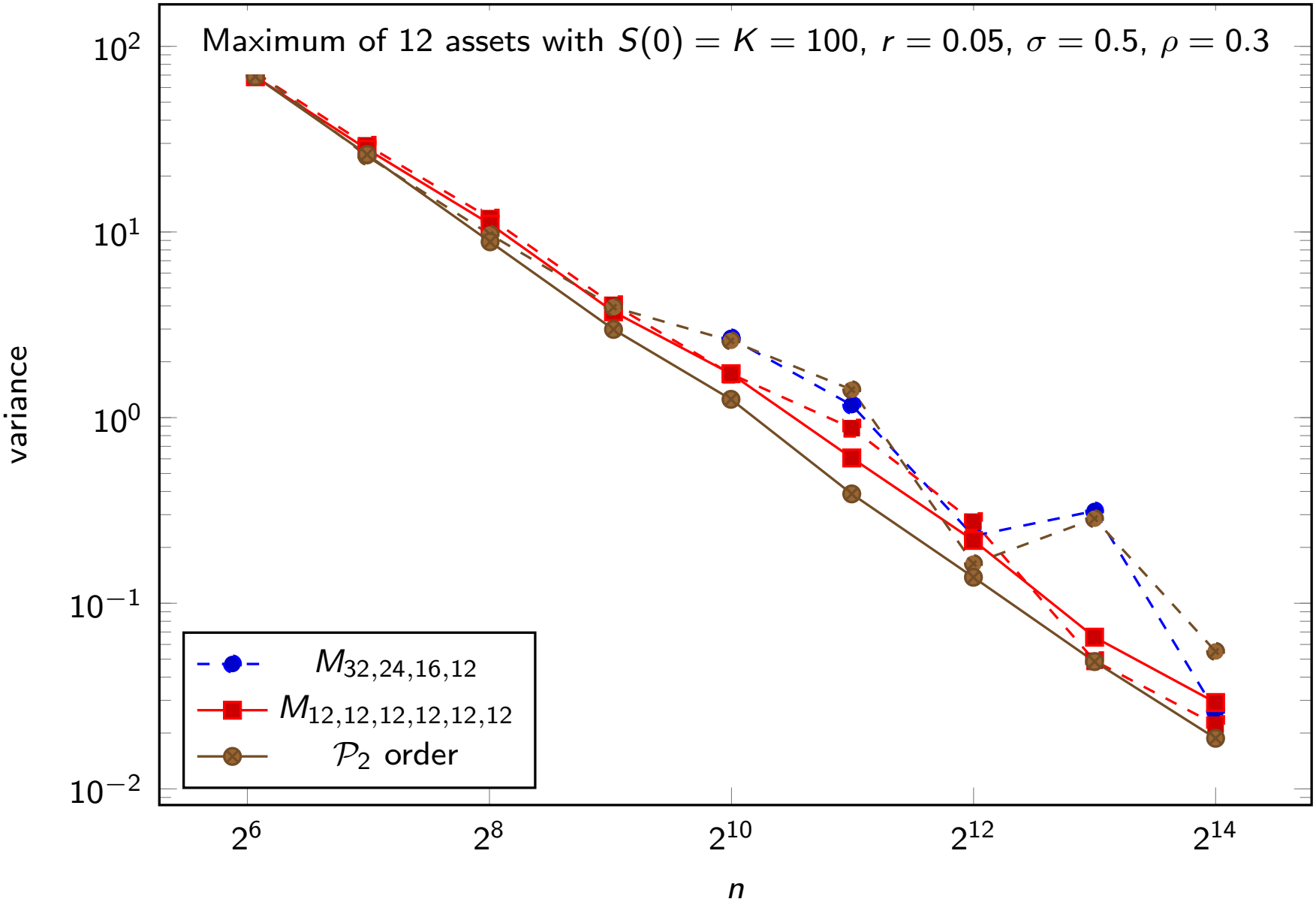
Lattices of Rank 1 with CBC



Prime vs. Power-of-2 Number of Points



Korobov vs. CBC



Solid: CBC. Dashed: Korobov.

Discrete choice with multinomial mixed logit probability, max likelihood estimation

Utility of alternative j for individual q is

$$
\begin{aligned}
U_{q,j} &= \boldsymbol{\beta}_q^{\mathsf{t}} \mathbf{x}_{q,j} + \epsilon_{q,j} = \sum_{\ell=1}^{s} \beta_{q,\ell} x_{q,j,\ell} + \epsilon_{q,j}, \text{ where} \\
\boldsymbol{\beta}_q^{\mathsf{t}} &= (\beta_{q,1}, \ldots, \beta_{q,s}) \text{ gives the tastes of individual } q, \\
\mathbf{x}_{q,j}^{\mathsf{t}} &= (x_{q,j,1}, \ldots, x_{q,j,s}) \text{ attributes of alternative } j \text{ for individual } q, \\
\epsilon_{q,j} & \quad \text{noise; Gumbel of mean 0 and scale parameter } \lambda = 1.
\end{aligned}
$$

Individual q selects alternative with largest utility $U_{q,j}$.

Can observe the $\mathbf{x}_{q,j}$ and choices y_q, but not the rest.

Logit model: for β_q fixed, j is chosen with probability

$$L_q(j \mid \beta_q) = \frac{\exp[\beta_q^{\mathrm{t}} \mathbf{x}_{q,j}]}{\sum_{a \in \mathcal{A}(q)} \exp[\beta_q^{\mathrm{t}} \mathbf{x}_{q,a}]}$$

where $\mathcal{A}(q)$ are the available alternatives for q.

Logit model: for $\boldsymbol{\beta}_q$ fixed, j is chosen with probability

$$L_q(j \mid \boldsymbol{\beta}_q) = \frac{\exp[\boldsymbol{\beta}_q^{\mathrm{t}} \mathbf{x}_{q,j}]}{\sum_{a \in \mathcal{A}(q)} \exp[\boldsymbol{\beta}_q^{\mathrm{t}} \mathbf{x}_{q,a}]}$$

where $\mathcal{A}(q)$ are the available alternatives for q.

For a random individual, suppose $\boldsymbol{\beta}_q$ is random with density $f_{\boldsymbol{\theta}}$, which depends on (unknown) parameter vector $\boldsymbol{\theta}$. We want to estimate $\boldsymbol{\theta}$ from the data (the $\mathbf{x}_{q,j}$ and y_q).

The unconditional probability of choosing j is

$$p_q(j, \boldsymbol{\theta}) = \int L_q(j \mid \boldsymbol{\beta}) f_{\boldsymbol{\theta}}(\boldsymbol{\beta}) \boldsymbol{\beta}.$$

It depends on $\mathcal{A}(q)$, j, and $\boldsymbol{\theta}$.

Maximum likelihood: Maximize the log of the joint probability of the sample, w.r.t. $\boldsymbol{\theta}$:

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^{m} p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^{m} \ln p_q(y_q, \boldsymbol{\theta}).$$

Maximum likelihood: Maximize the log of the joint probability of the sample, w.r.t. $\boldsymbol{\theta}$:

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^{m} p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^{m} \ln p_q(y_q, \boldsymbol{\theta}).$$

No formula for $p_q(j, \boldsymbol{\theta})$, but can use MC or RQMC, for each q and fixed $\boldsymbol{\theta}$.

Generate n realizations of $\boldsymbol{\beta}$ from $f_{\boldsymbol{\theta}}$, say $\boldsymbol{\beta}_q^{(1)}(\boldsymbol{\theta}), \ldots, \boldsymbol{\beta}_q^{(n)}(\boldsymbol{\theta})$, and estimate $p_q(y_q, \boldsymbol{\theta})$ by

$$\hat{p}_q(y_q, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L_q(j, \boldsymbol{\beta}_q^{(i)}(\boldsymbol{\theta})).$$

Then we can find the maximizer $\hat{\boldsymbol{\theta}}$ of $\ln \prod_{q=1}^{m} \hat{p}_q(y_q, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

Maximum likelihood: Maximize the log of the joint probability of the sample, w.r.t. $\boldsymbol{\theta}$:

$$\ln L(\boldsymbol{\theta}) = \ln \prod_{q=1}^{m} p_q(y_q, \boldsymbol{\theta}) = \sum_{q=1}^{m} \ln p_q(y_q, \boldsymbol{\theta}).$$

No formula for $p_q(j, \boldsymbol{\theta})$, but can use MC or RQMC, for each q and fixed $\boldsymbol{\theta}$.

Generate n realizations of $\boldsymbol{\beta}$ from $f_{\boldsymbol{\theta}}$, say $\boldsymbol{\beta}_q^{(1)}(\boldsymbol{\theta}), \ldots, \boldsymbol{\beta}_q^{(n)}(\boldsymbol{\theta})$, and estimate $p_q(y_q, \boldsymbol{\theta})$ by

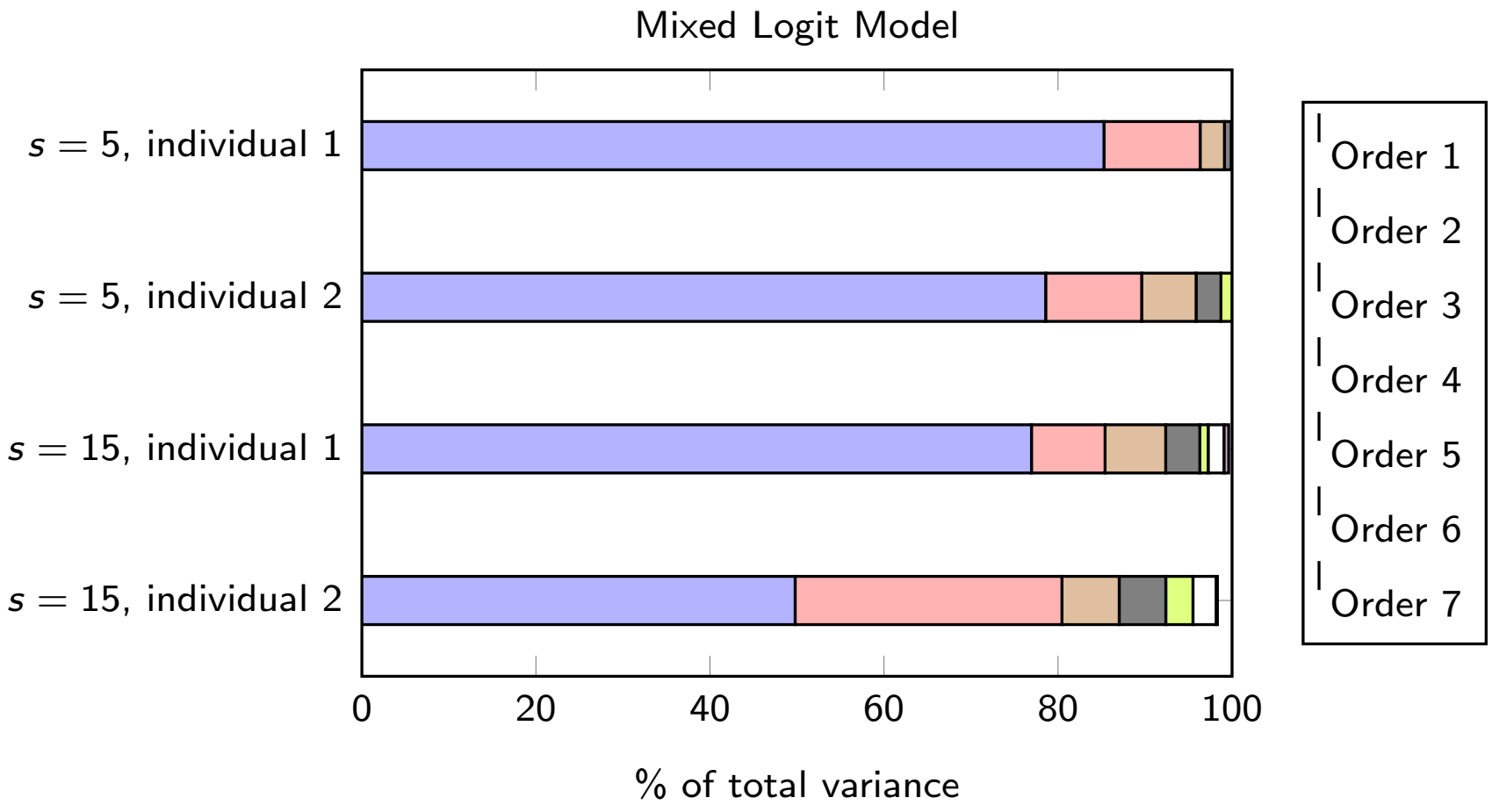
$$\hat{p}_q(y_q, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L_q(j, \boldsymbol{\beta}_q^{(i)}(\boldsymbol{\theta})).$$

Then we can find the maximizer $\hat{\boldsymbol{\theta}}$ of $\ln \prod_{q=1}^{m} \hat{p}_q(y_q, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

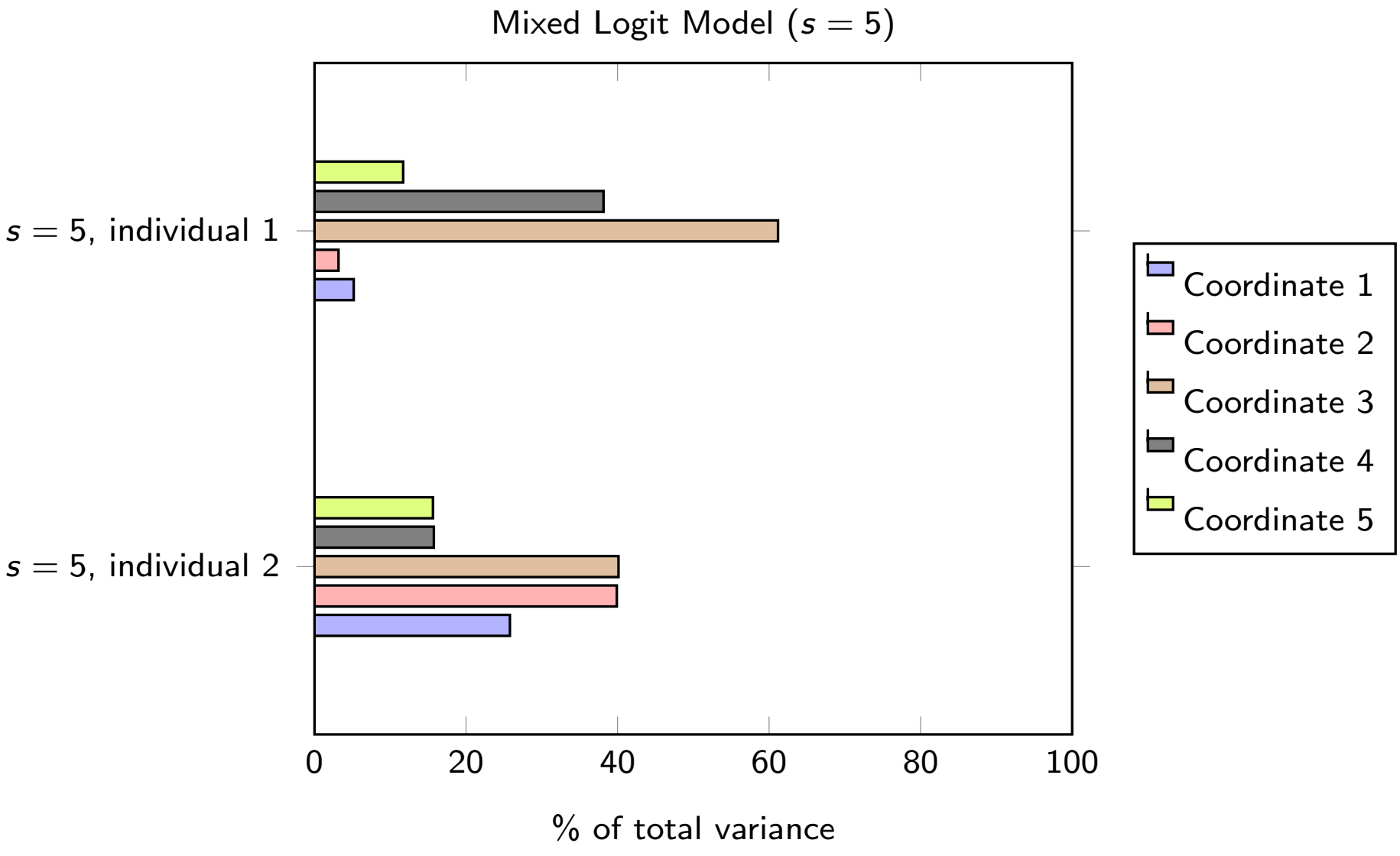
We take 4 alternatives, with indep. attributes, resp. $N(1,1)$, $N(1,1)$, $N(0.5,1)$, $N(0.5,1)$. We try $s = 5$, 10, 15.

$\boldsymbol{\beta}_q$ is a vector of s indep. $N(1,1)$ random variables.

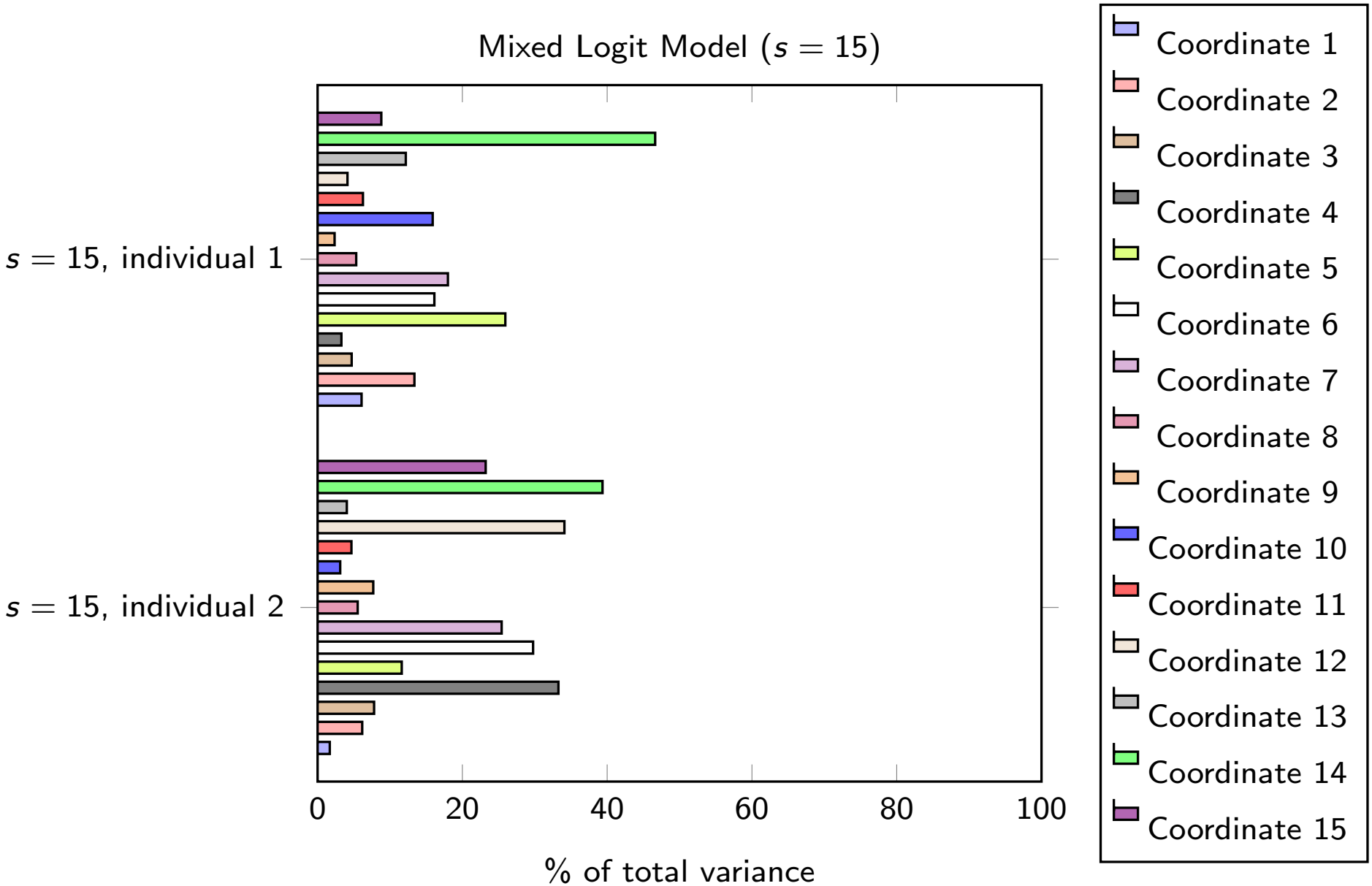
ANOVA Variances for the Mixed Logit Model



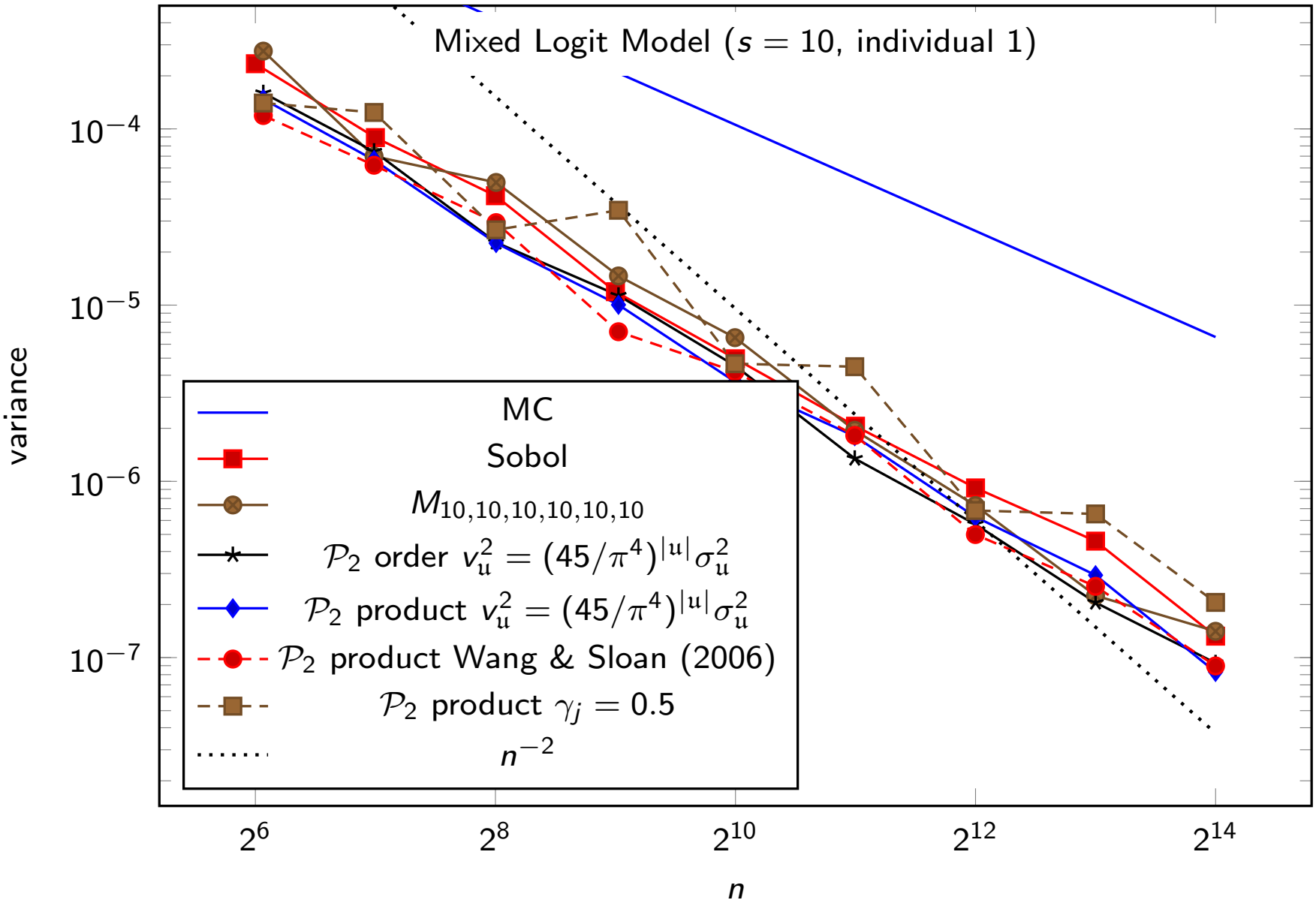
Total variance per coordinate



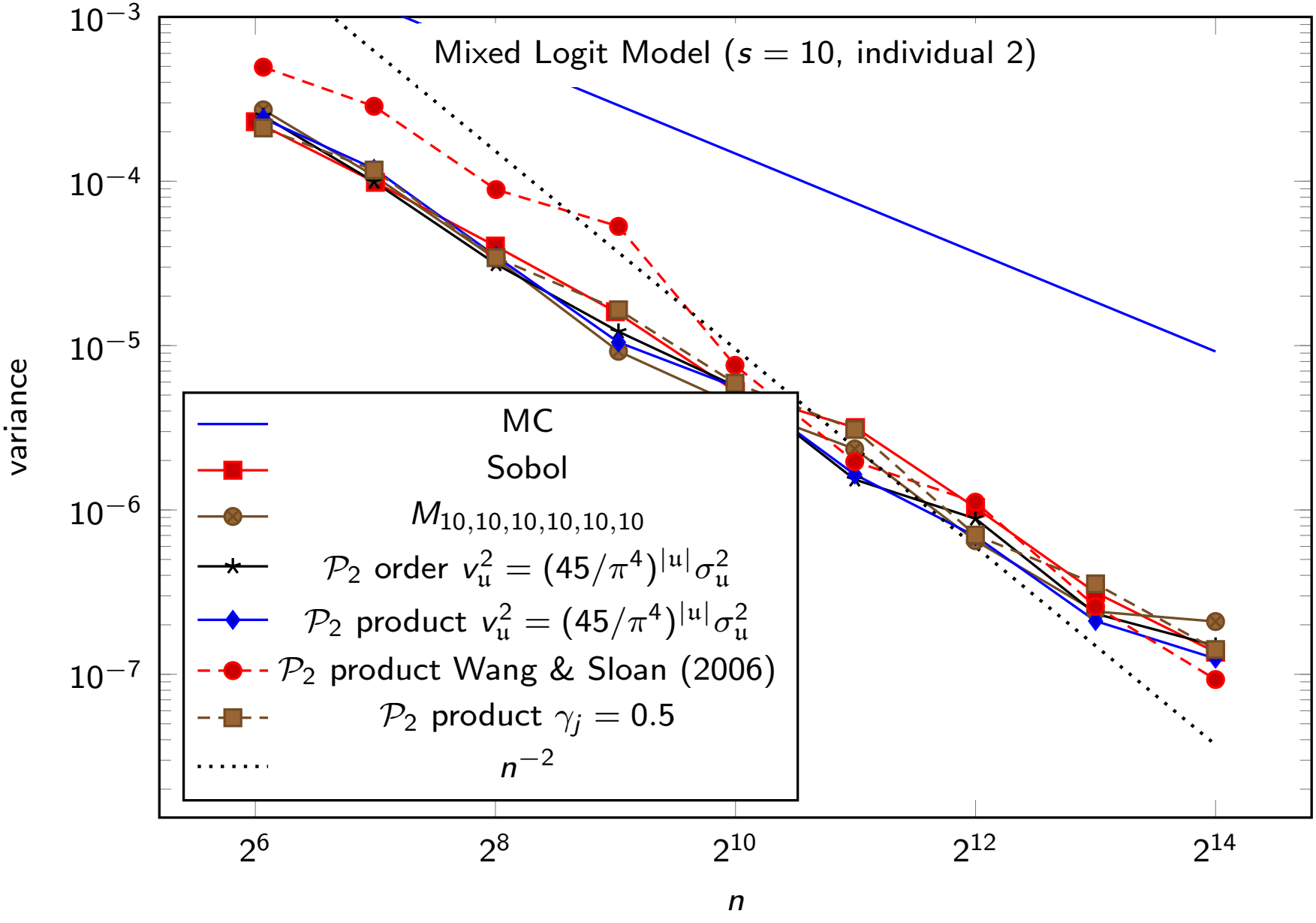
Total variance per coordinate



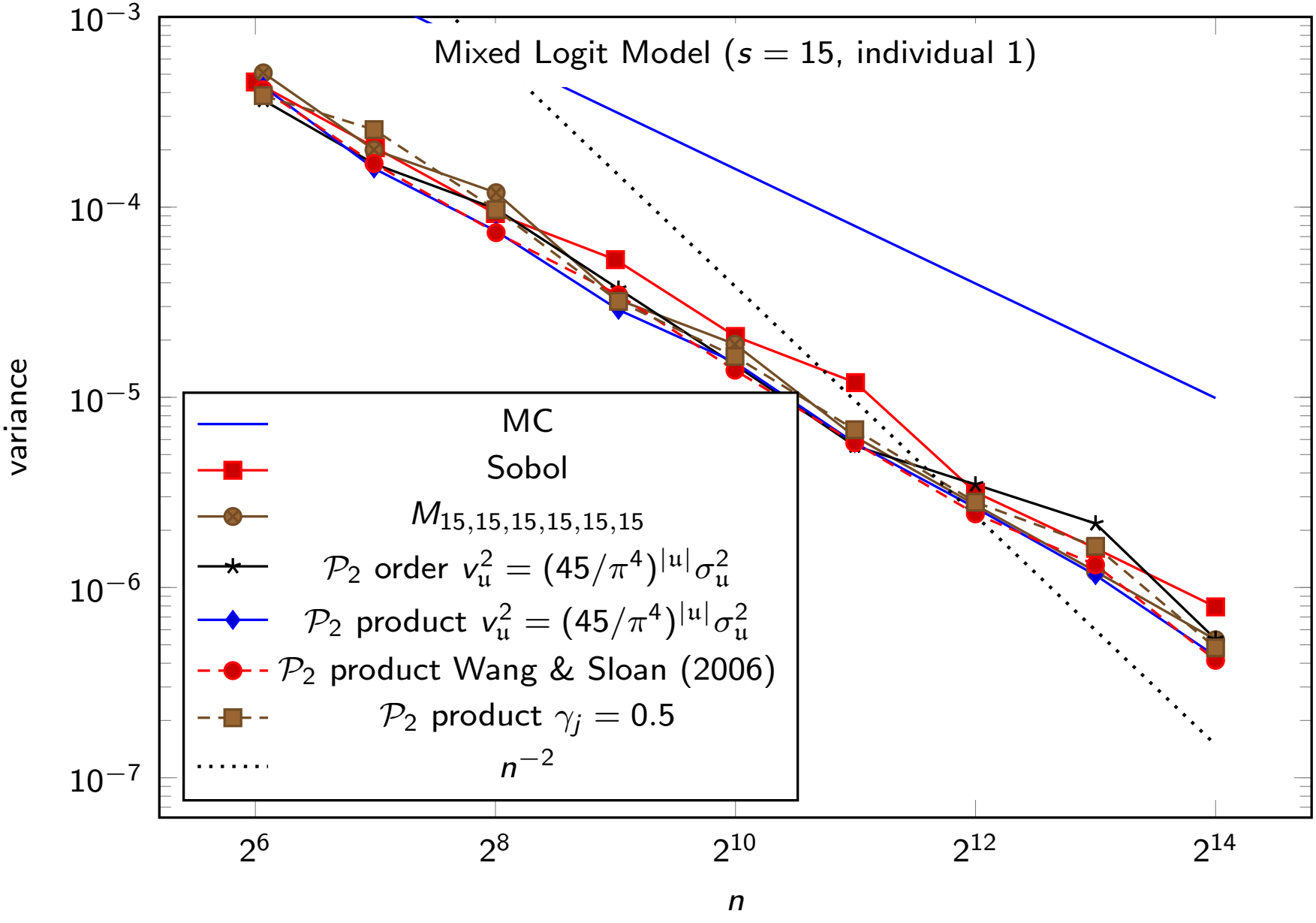
Lattices of Rank 1 with CBC



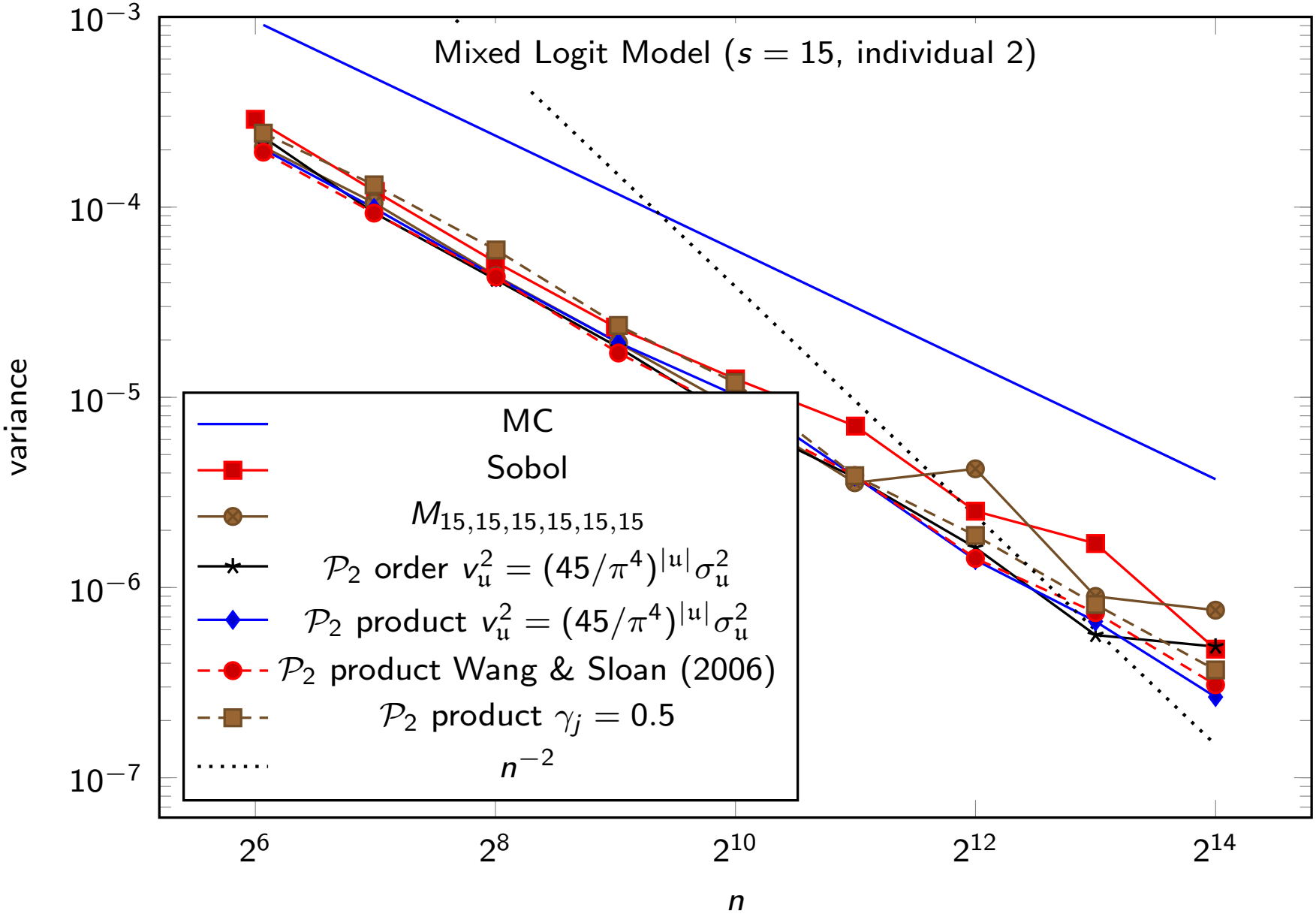
Lattices of Rank 1 with CBC



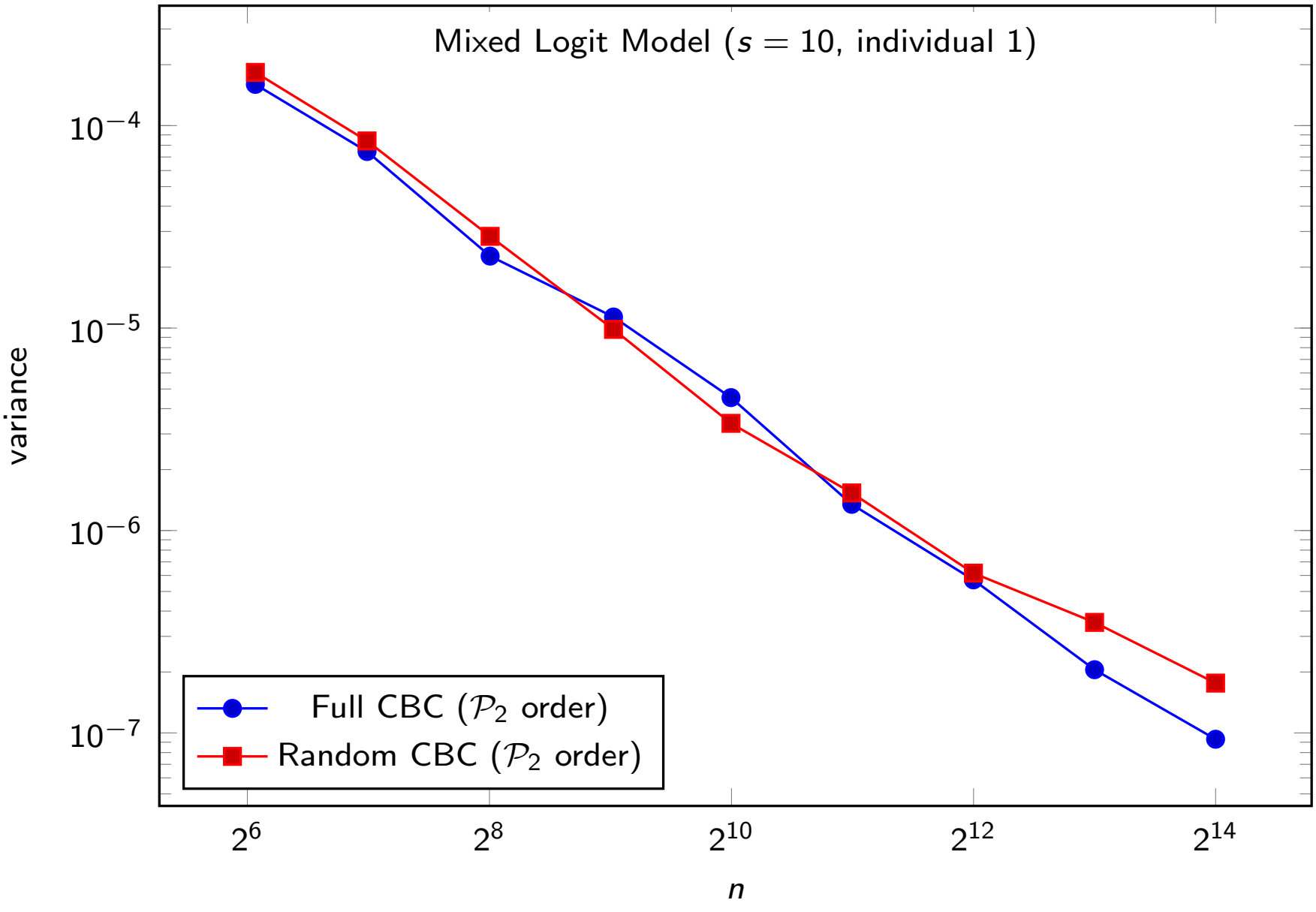
Lattices of Rank 1 with CBC



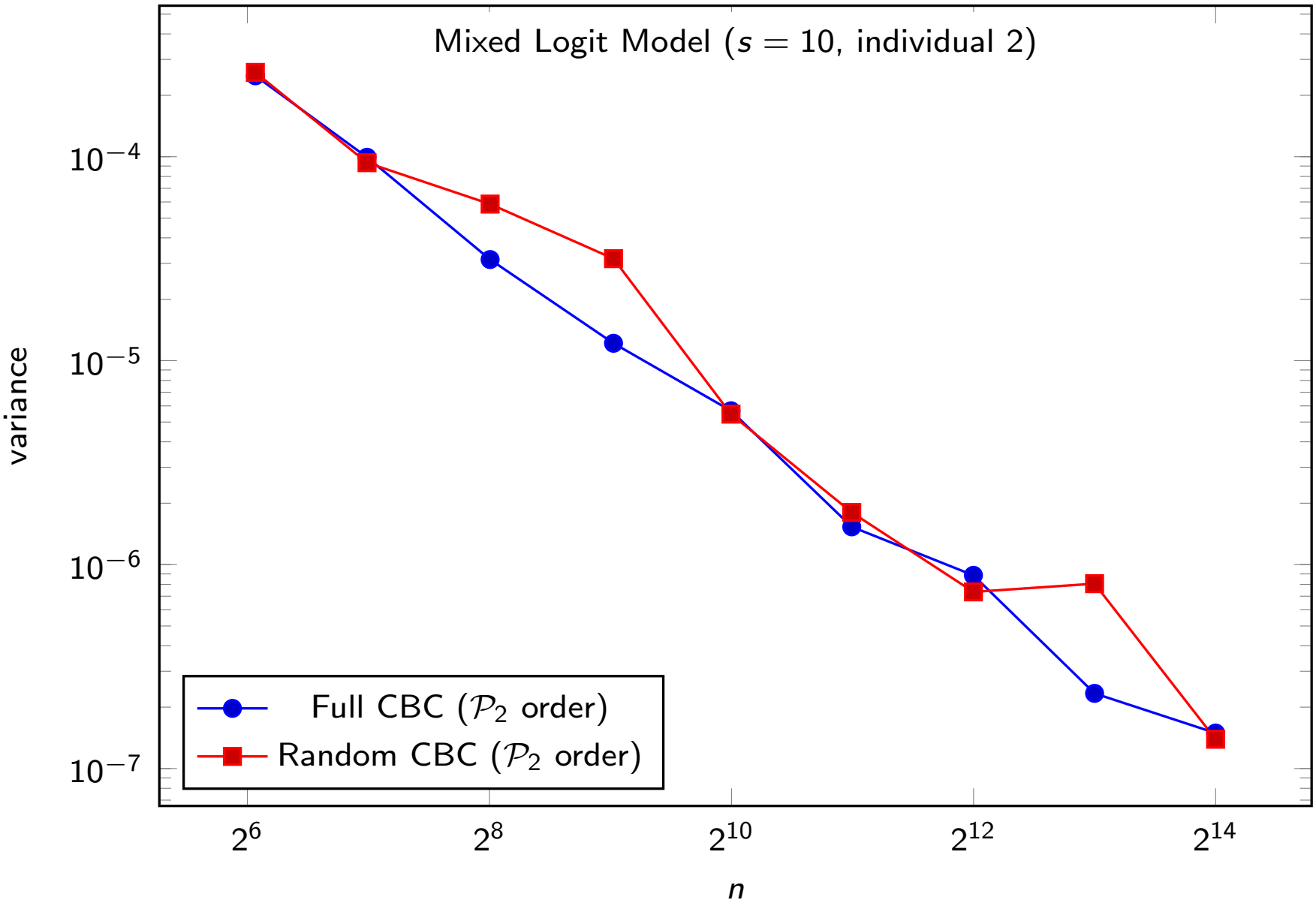
Lattices of Rank 1 with CBC



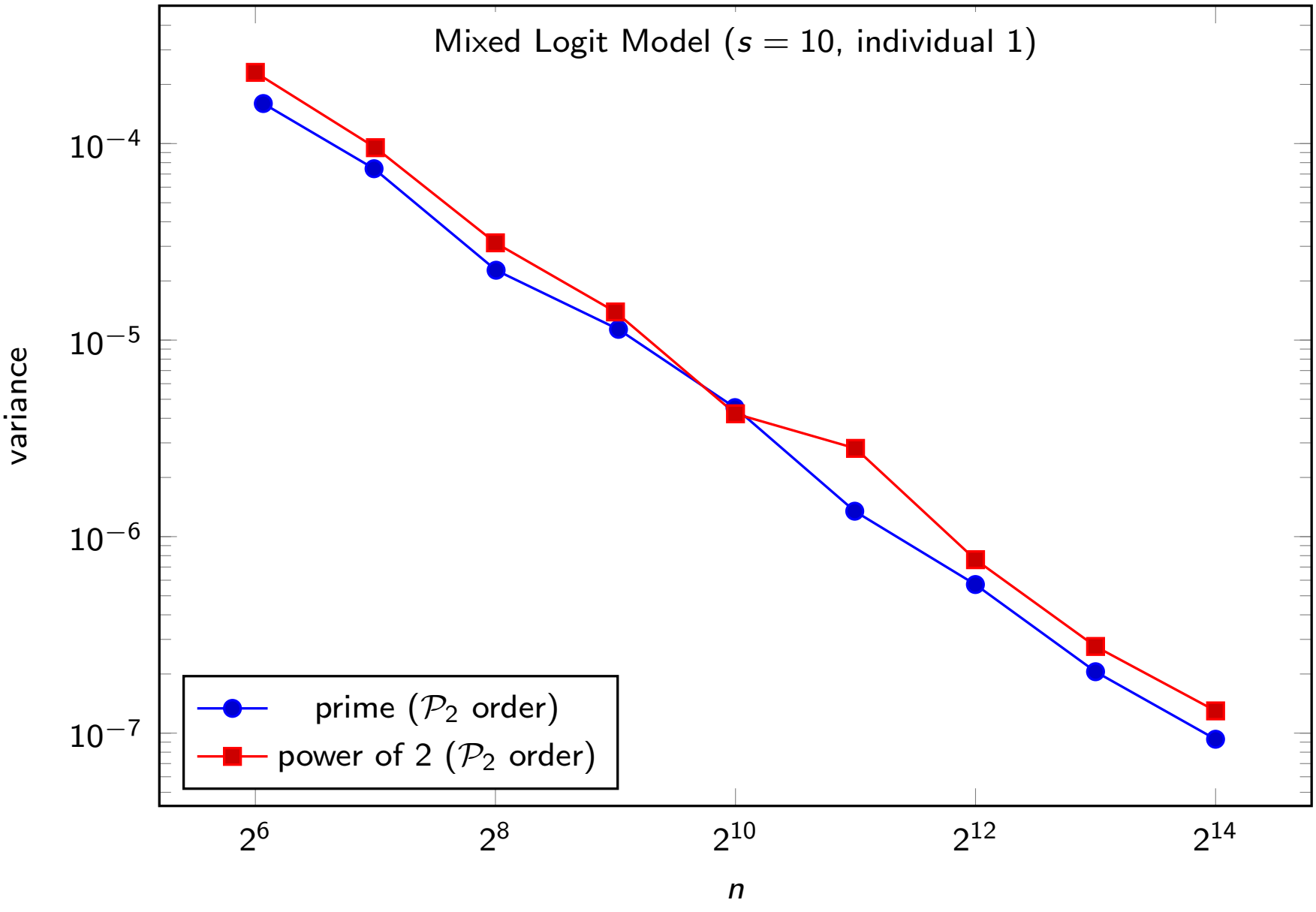
Random vs. Full CBC



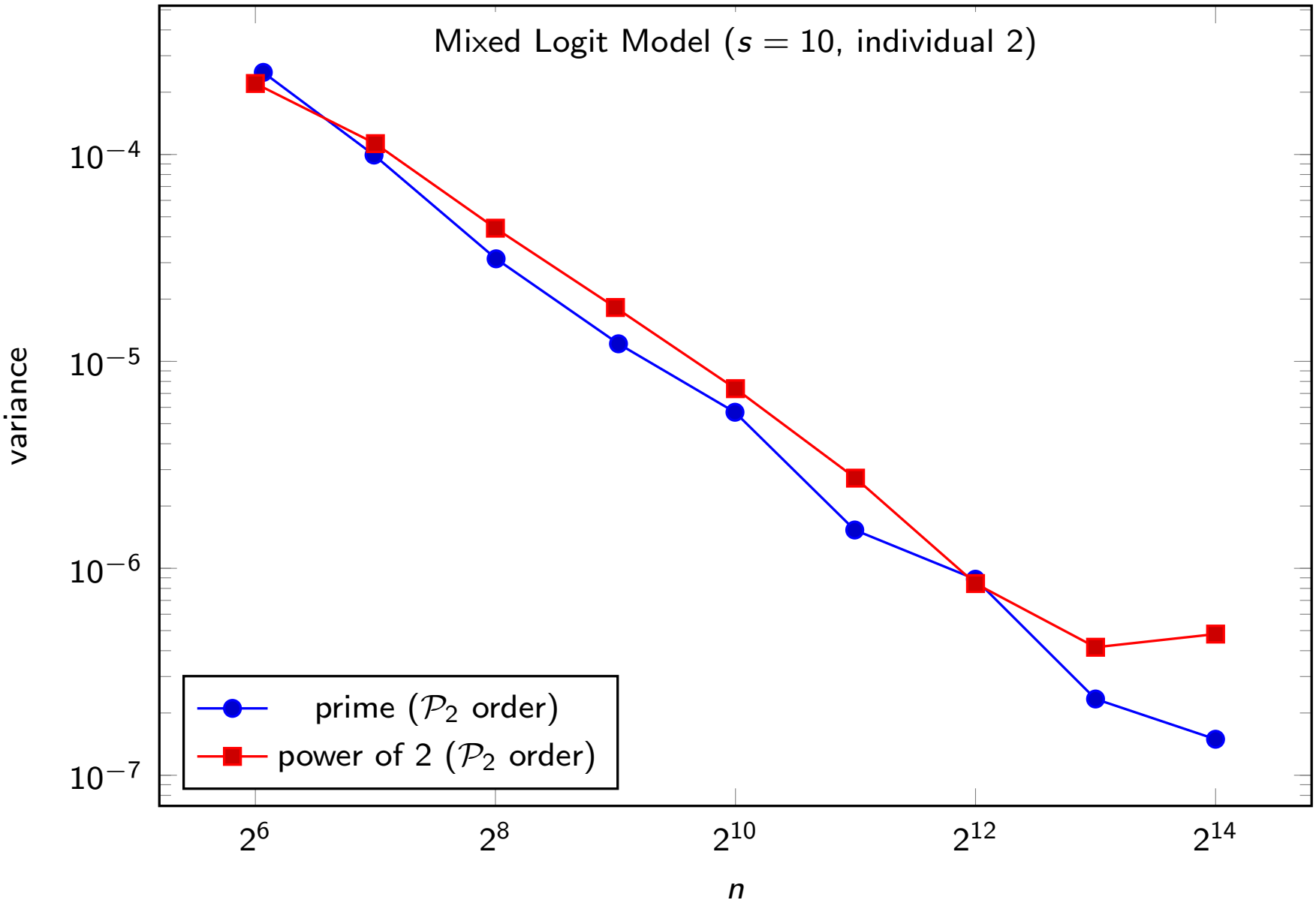
Random vs. Full CBC



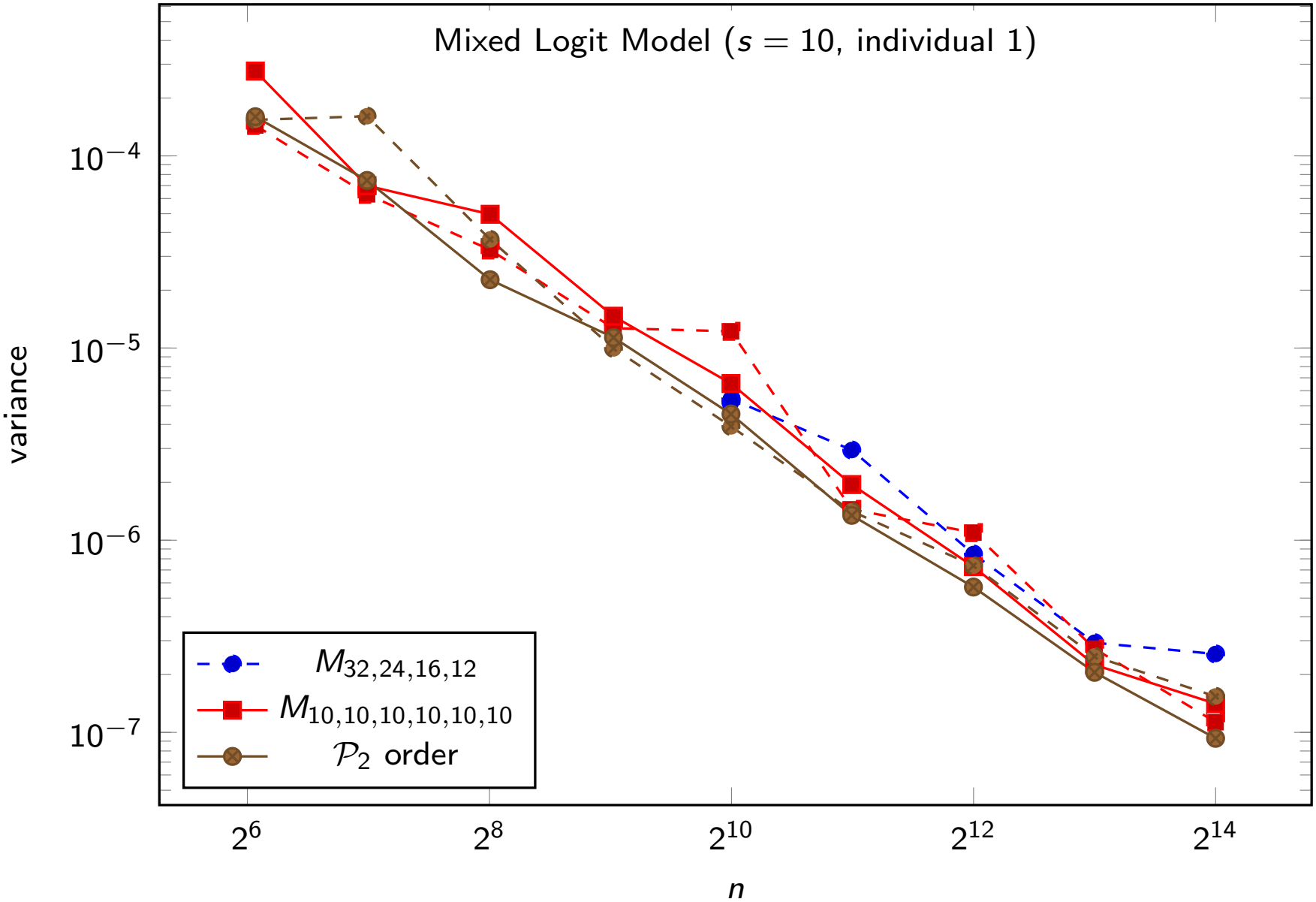
Prime vs. Power-of-2 Number of Points



Prime vs. Power-of-2 Number of Points

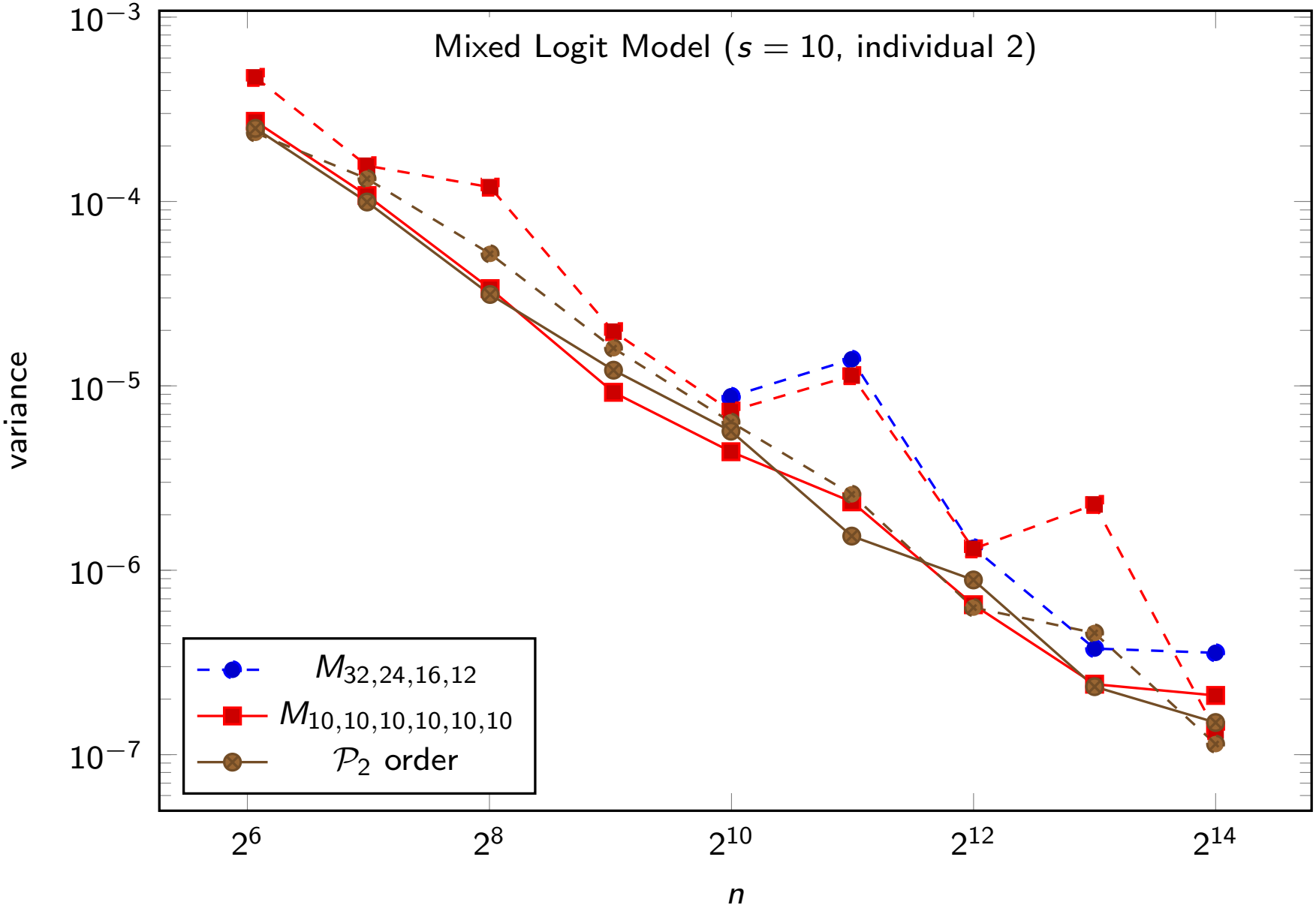


Korobov vs. CBC



Solid: CBC. Dashed: Korobov.

Korobov vs. CBC



Solid: CBC. Dashed: Korobov.

References for the material of this talk (some were added afterward):

1. P. L'Ecuyer and D. Munger, "On the Choice of Figure of Merit for Randomly-Shifted Lattice Rules", in Monte Carlo and Quasi Monte Carlo Methods 2010, H. Wozniakowski and L. Plaskota, Eds., Springer-Verlag, Berlin, 2012, 133–159.

2. P. L'Ecuyer and C. Lemieux, "Variance Reduction via Lattice Rules", Management Science **46**, 9 (2000), 1214–1235.

3. P. L'Ecuyer, "Quasi-Monte Carlo Methods in Finance", Proceedings of the 2004 Winter Simulation Conference, IEEE Press, 2004, 1645–1655.

4. P. L'Ecuyer, "Quasi-Monte Carlo Methods with Applications in Finance," Finance and Stochastics, 13, 3 (2009), 307–349.

5. P. L'Ecuyer and D. Munger, "Algorithm 958: LatticeBuilder: A General Software Tool for Constructing Rank-1 Lattice Rules", ACM Transactions on Mathematical Software, **42**, 2, Article 15, 2016.