# Sequential Importance Sampling for Counting Linear Extensions

Isabel Beichl

NIST, Applied & Computational Mathematics Division

work done in collaboration with
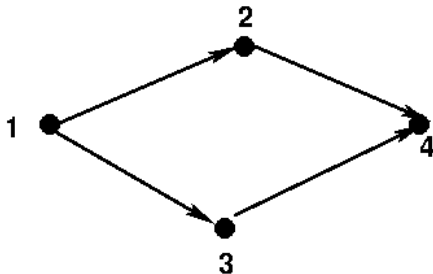Francis Sullivan
Center for Computing Sciences
Alathea Jensen
George Mason University

# Counting Linear Extensions

Counting Linear Extensions of a Poset How many ways to order
the vertices of a poset or DAG consistent with the order.

# Counting Linear Extensions of a Poset

- ▶ How many ways to order the vertices of a poset or DAG consistent with the order.
- ▶ = the number of topological sorts of the directed acyclic graph
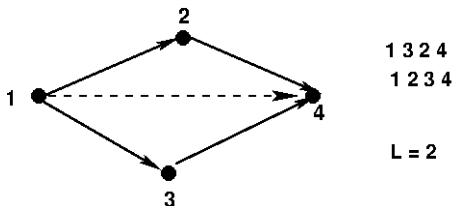


- ▶ poset = partially ordered set, DAG = directed acyclic graph.
- ▶ For this problem, number of extensions of a poset = number of topological sorts of a DAG = number of orderings of vertices that preserves DAG structure.
- ▶ As we start by taking the transitive closure of the DAG, these are the same for our purposes.

# Why approximation is important

- ▶ Scheduling - Processor, construction
- ▶ A measure of the number of choices remaining = A measure of degrees of freedom
- ▶ Brightwell Winkler, proved NP hard to do exactly
- ▶ Karzanov Kachiyan MCMC approximation, high degree polynomial Not practical

# Relation to linear algebra

- Sometimes want a matrix upper triangular with row and column permutations.
- $v \to w$ iff $a(v, w) \neq 0$
- Can do iff when thought of as a DAG, can get a linear extension.

This method gives a fast way to evaluate the number of ways to make upper triangular with row & column interchanges.

For optimization in scheduling, want to know that there are many possibilities to choose from.
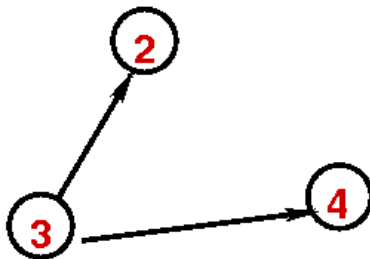
We investigated Sequential Importance Sampling (SIS) as an approximation method.

# Recall Classical Topological Sort
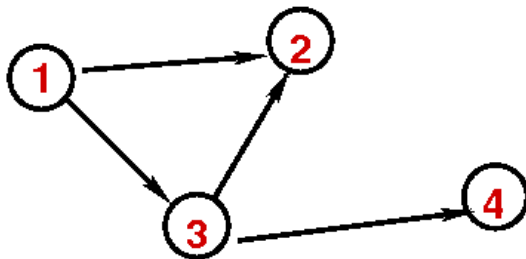


Choose any vertex with no predecessors, put it in the list.
Then delete that vertex.

# Recall Classical Topological Sort



Choose any vertex with no predecessors, put it in the list.
Then delete that vertex.

# Recall Classical Topological Sort



Choose any vertex with no predecessors, put it in the list.
Then delete that vertex.

# Recall Classical Topological Sort

$(4)$

**1   3   2**

Choose any vertex with no predecessors, put it in the list.
Then delete that vertex.

# What is the Knuth method

- Used to find the size of a tree you don't have explicitly
- Take a sample from the root to the leaves: At each node make a choice of which child to take and note the number of possibilities
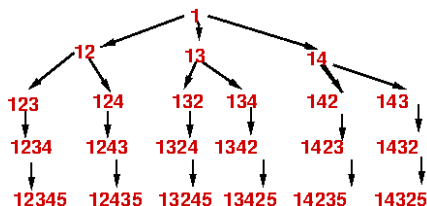- The estimate is the product of the possibilities
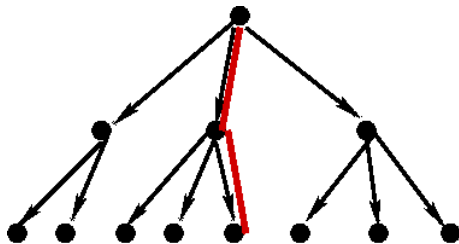


**Estimate = 3 * 3 = 9**

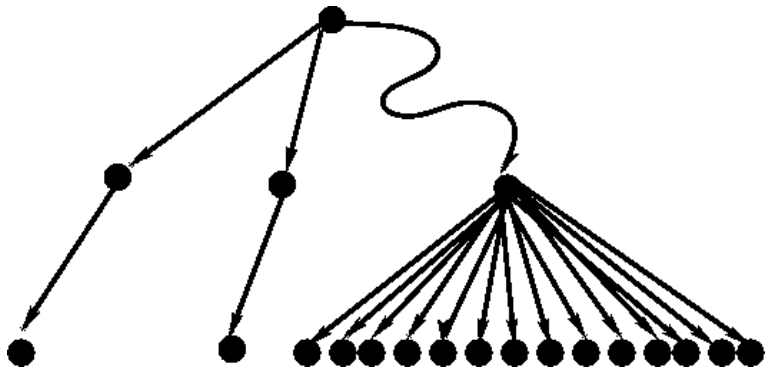# Knuth method applied to our problem
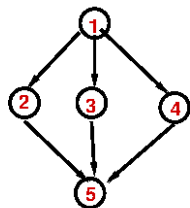


DAG

Knuth Tree

The tree is made by all possible partial extensions.

# Problem with the Knuth method



Variance can be large

# Sequential Importance Sampling

▶ A variance reduction technique

Suppose we wish to estimate a sum:

$$F(N) = \sum_{j=1}^{N} f(\sigma_j)$$

A basic approach is to take a sample of size $M << N$

$$F(N) \approx \frac{N}{M} \sum_{j=1}^{M} f(\sigma_j)$$

where each $\sigma_j$ is uniformly generated
Note that $1/N$ is the probability of selecting $\sigma_j$ so

$$F(N) \approx \left( \sum_{j=1}^{M} \frac{f(\sigma_j)N}{M} \right) = \frac{1}{M} \left( \sum_{j=1}^{M} \frac{f(sigma_j)}{p(\sigma_j)} \right)$$

## Sequential Importance Sampling

Importance sampling says you can select $\sigma_j$ NON-uniformly if you divide by $p(\sigma_j)$.

$$\sum_{j=1}^{M} \frac{f(\sigma_j)}{p(\sigma_j)} \frac{1}{M} \rightarrow \sum_{\sigma} \frac{f(\sigma)}{p(\sigma)} p(\sigma) = \sum_{\sigma} f(\sigma) = F(N)$$

because in the long run $\sigma$ will be chosen $p(\sigma) * M$ times. This limit holds for any probability distribution $p(\sigma)$.

▶ But how to choose $p()$?

## Sequential Importance Sampling

The *ideal* choice of importance function is

$$p(\sigma) = \frac{f(\sigma)}{F(N)}$$

i.e. the weight assigned to $\sigma$ is its relative contribution to the desired sum. SO

$$var = \frac{1}{M}\left(\sum_{j=1}^{M}\left(\frac{f(\sigma_j)}{p(\sigma_j)}\right)^2\right) - F^2$$

$$\rightarrow \sum_{\sigma}\frac{f^2(\sigma)}{p^2(\sigma)}p(\sigma) - F^2 = \sum_{\sigma}\frac{f^2(\sigma)}{p(\sigma)} - F^2 = F^2 - F^2 = 0.$$
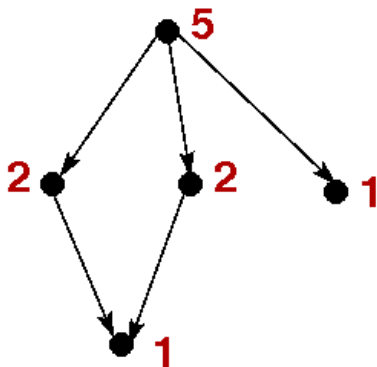
But this requires knowledge of the answer!!!

# Sequential Importnace Sampling

But sometimes we DO know something about the tree

In our case the path through the tree is made by one extension.

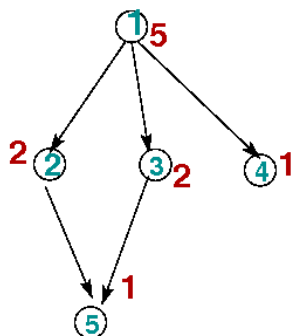# An Importance Function for Counting Extensions

The number of descendants of a node $+ 1$.



Does not change during the course of the computation.
The row sums of the adjacency matrix $+ 1$.

# How does our SIS work?



Ext = 1 3 2 4 5

p = 5/5 * 2/(2+2+1) * 2/(2+1) * 1/2 * 1/1

$$= \frac{r_1 \quad r_3 \quad r_2 \quad r_4 \quad r_5}{(r_1) \; *(r_2 + r_3 + r_4)*(r_2 + r_4)*(r_4 + r_5)*r_5}$$

ESTIMATE = 1/p

# EXACT for G = a tree!
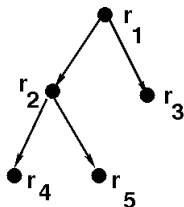


$$\text{Est} = \frac{r_1 * (r_2 + r_3) * (r_3 + r_4 + r_5) * (r_4 + r_5) * r_5}{r_1 \, r_2 \, r_3 \, r_4 \, r_5}$$

$$= \frac{n * (n-1) * (n-2) * {}^* 2 * 1}{\text{prod } r}$$

All samples are the *same*

## Variance for SIS

Let $s$ be a random variable. $X$ a poset. Want to know:

$$\frac{\langle s^2 \rangle}{\langle s \rangle^2}$$

Our $s$ is $1/p$ sampled with probability $p$.

$$\langle s \rangle_p = \sum_{all\,i} s_i * p_i = \sum_i \frac{1}{p_i} * p_i = \sum_{L(X)} 1 = L(X)$$

$$\langle s^2 \rangle_p = \sum_i s_i^2 p_i = \sum_i \frac{1}{p_i^2} * p_i = \sum \frac{1}{p_i} = L(X) \langle \frac{1}{p} \rangle_u$$

where $\langle . \rangle_u$ is the uniform average.

These look the same but they are not because the frequencies are different.

## Sequential Importance Sampling

For the special case of linear extensions

$$\frac{1}{p} = \frac{x}{\prod_k r_k}$$

where $x$ is the product of all those numerators. So in our case

$$\frac{\langle s^2 \rangle_p}{\langle s \rangle_p^2} = \frac{L(X) * \langle \frac{x}{\prod r} \rangle_u}{L(X)^2} = \frac{\langle \frac{x}{\prod r} \rangle_u}{L(X)} = \frac{\langle x \rangle_u}{\prod r L(X)}$$

## Recursion for better Variance

For a disjoint union of subgraphs $X = Y \cup Z$, there is a formula

$$L(X) = L(Y) * L(Z) \binom{s + t}{s}$$

where $Y$ has a size $s$ and $Z$ has size $t$.
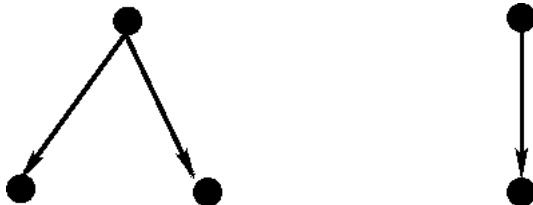


Recursion using connected components also reduces variance.

Comparison of Importance Functions

- Importance Function = Constant
- Importance Function = # Successors
- Power Function Fit y = 0.001164x^2.917
- Power Function Fit y = 0.00003658x^3.441
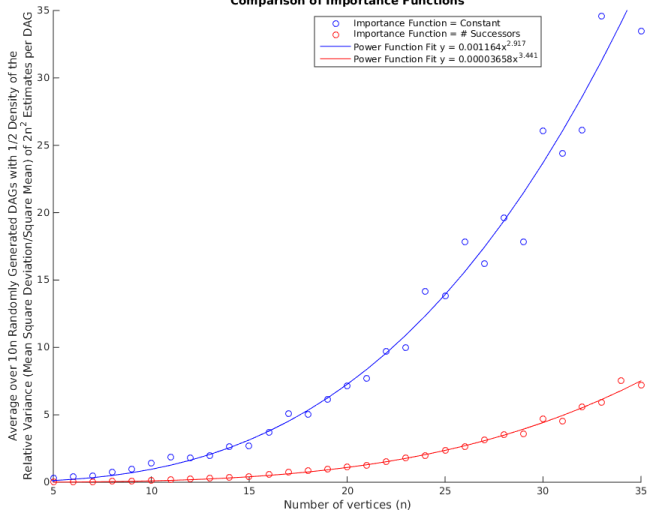
# Summary for Successors Imnportance Function

- Reduces variance over uniform importance
- Can prove it does trees exactly
- Can prove recursion reduces variance

# Another importance function



slots used  |  open slots  |  number of descendants

Want vertices with smaller "open slots" to get preference
So we put it in the denominator of the imp fcn

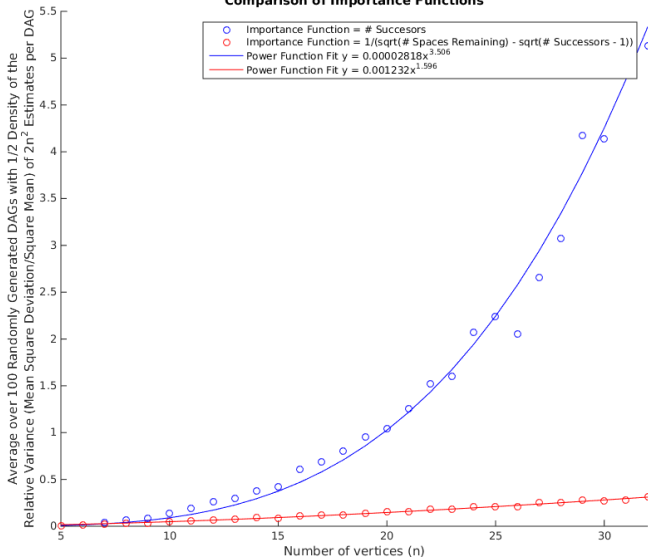Prefer to choose vertices that have smaller "open slots"
Taking sqrt is even better. (Why?)

$$p = \frac{1}{\sqrt{spaces.remaining} - \sqrt{\#successors}}$$

Comparison of Importance Functions

○ Importance Function = # Succesors
○ Importance Function = 1/(sqrt(# Spaces Remaining) - sqrt(# Successors - 1))
— Power Function Fit y = 0.00002818x^3.506
— Power Function Fit y = 0.001232x^1.596

$y$-axis: Average over 100 Randomly Generated DAGs with 1/2 Density of the Relative Variance (Mean Square Deviation/Square Mean) of $2n^2$ Estimates per DAG

$x$-axis: Number of vertices (n)

Comparison of Importance Functions

**Comparison of Importance Functions**

- ○ Importance Function = # Succesors
- ○ Importance Function = 1/(sqrt(# Spaces Remaining) - sqrt(# Successors - 1))
- Power Function Fit $y = 0.000005592x^{4.112}$
- Power Function Fit $y = 0.005617x^{1.273}$

Y-axis: Average over 100 Randomly Generated DAGs with 0.35 Density of the Relative Variance (Mean Square Deviation/Square Mean) of $2n^2$ Estimates per DAG

X-axis: Number of vertices (n)