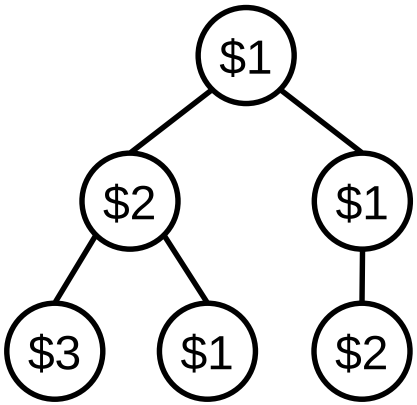
STOCHASTIC ENUMERATION WITH IMPORTANCE SAMPLING

Alathea Jensen

December 11, 2017

Tree Size Problems

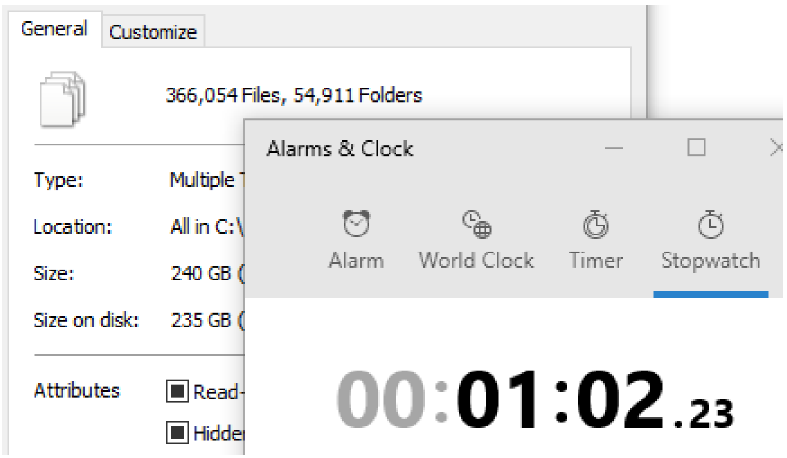
- Many problems in math, physics, and computer science boil down to the same underlying question: how big is this tree?
- Some different flavors of this question:
  - How many leaves are on this tree?
  - How many nodes are in this tree?
  - Given a cost function, what’s the total cost of this tree?
- Example
  - 3 leaves
  - 6 nodes
  - $10 cost

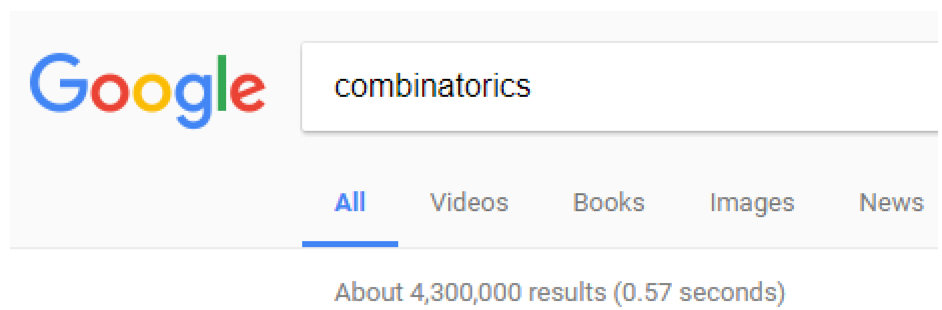


Tree Size Problems

- Counting problems
  - Any set whose elements can be constructed by a series of decisions can be modeled by a decision tree
  - E.g. graph colorings, spanning trees, set partitions, etc.
- Algorithms
  - A decision tree can model all possible ways that an algorithm might proceed
  - Statistics such as runtime and memory usage can be modeled by cost functions on the tree
- Databases
  - Most large databases are organized as trees in order to optimize searching and updating
  - E.g. personal computer files (~300,000), Amazon listings (~500 million), Facebook accounts (~2 billion), indexed pages on Google (~100 trillion)

Database Examples



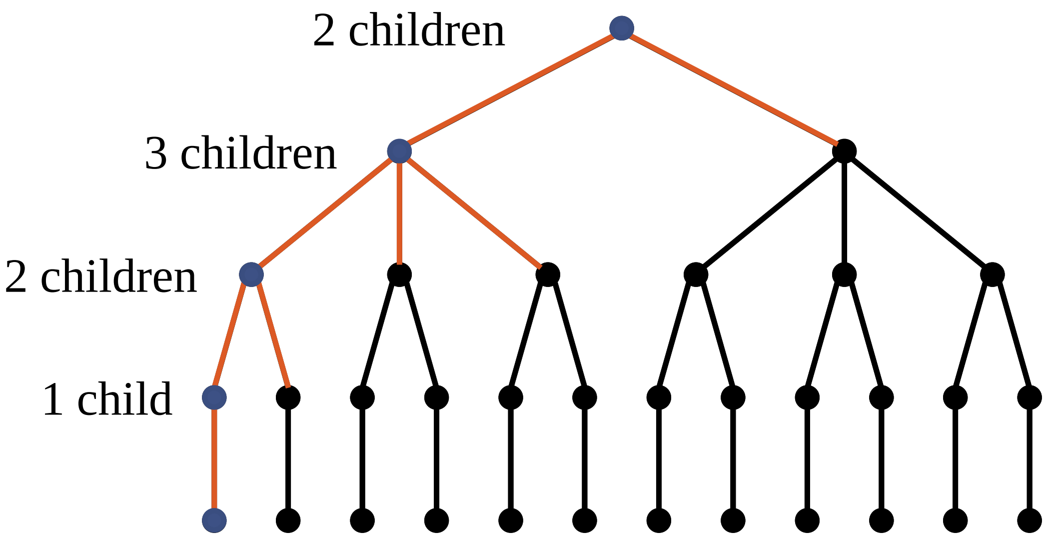


Tree Size Estimation

- Calculating tree size is $\#P$-complete in general
- Goal: efficient estimation of tree size
- Two main types of estimation algorithms
- Markov Chain Monte Carlo (MCMC)
  - Easier to bound variance of samples
  - Takes longer to get each sample
- Sequential Importance Sampling (SIS)
  - Harder to bound variance of samples
  - Get samples very quickly

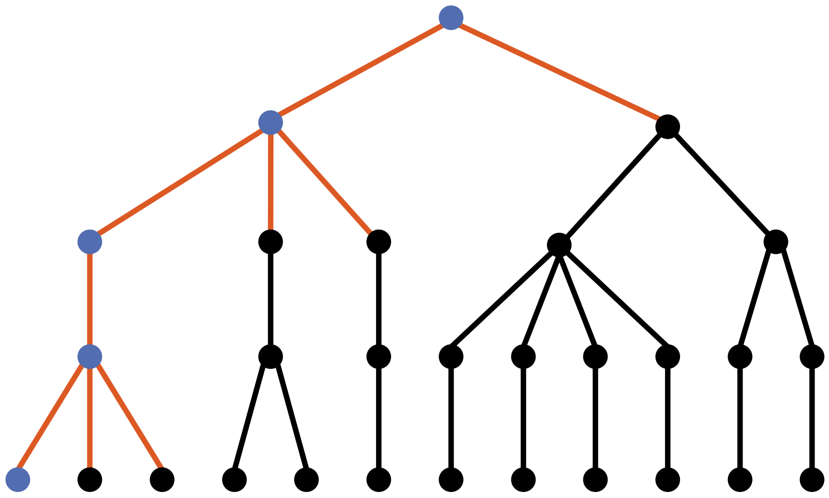
Motivation

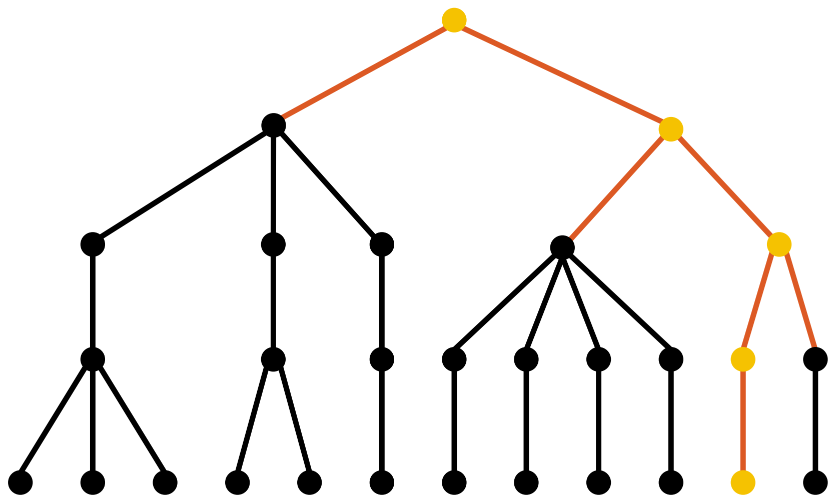
- If the number of children per node is uniform across each level…
- Then the number of leaves is the product of the number of children along any path from root to leaf
- Example
  - $2 \cdot 3 \cdot 2 \cdot 1 = 12$ leaves



Motivation

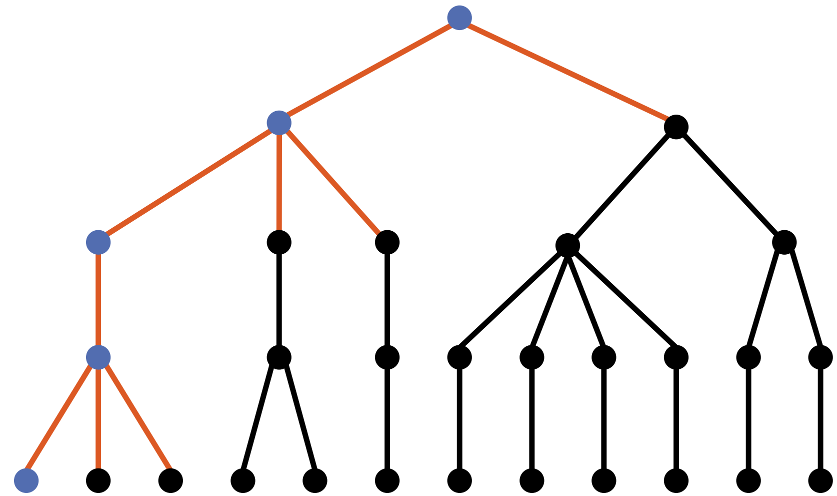
- If the number of children per node is not uniform across each level…
- Products will be different along different paths
- Example
  - Blue node path: $2 \cdot 3 \cdot 1 \cdot 3 = 18$
  - Yellow node path: $2 \cdot 2 \cdot 2 \cdot 1 = 8$
  - But the number of leaves is still 12

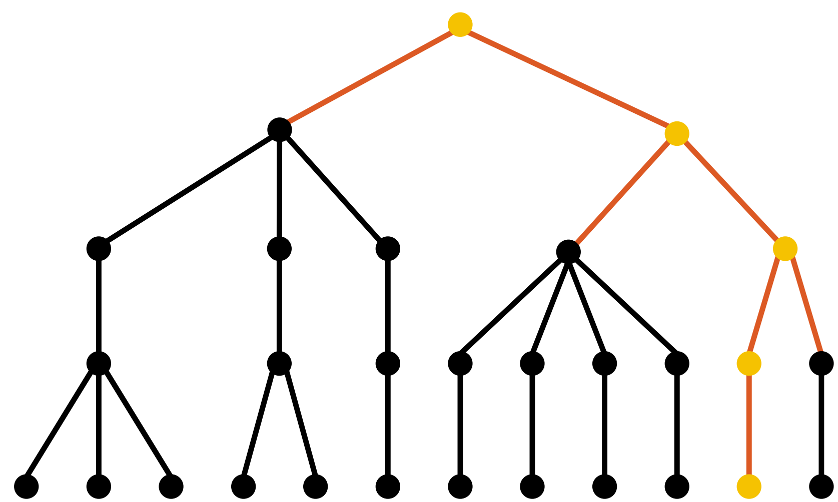




Knuth’s Algorithm

- Donald E. Knuth, 1975
- Construct a path by starting with the root and choosing a random child (uniform distribution) of the previous node to continue the path
- The estimate associated with each path is the product of the number of children seen along the path
- Example
  - The blue node path gives an estimate of $2 \cdot 3 \cdot 1 \cdot 3 = 18$
  - The probability of constructing blue node path is $\frac{1}{2} \cdot \frac{1}{3} \cdot 1 \cdot \frac{1}{3} = \frac{1}{18}$
  - The yellow node path gives as estimate of $2 \cdot 2 \cdot 2 \cdot 1 = 8$
  - The probability of constructing the yellow node path is $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot 1 = \frac{1}{8}$





Knuth's Algorithm

- Call the set of all possible paths P
- For each possible path, $p \in P$, the associated estimate, $\mathrm{est}(p)$, is the reciprocal of the probability $\mathrm{prob}(p)$ of choosing that path

$$\mathrm{est}(p) = \frac{1}{\mathrm{prob}(p)}$$

- Each possible path contributes exactly 1 to the expected value sum

$$\mathbb{E}[\mathrm{est}(p)] = \sum_{p \in P} \mathrm{est}(p) \cdot \mathrm{prob}(p) = \sum_{p \in P} 1 = |P|$$

- Hence the expected value of the estimate is the number of possible paths, which is the same as the number of leaves
- Therefore the average value of many estimates will converge to the correct answer

Sequential Importance Sampling

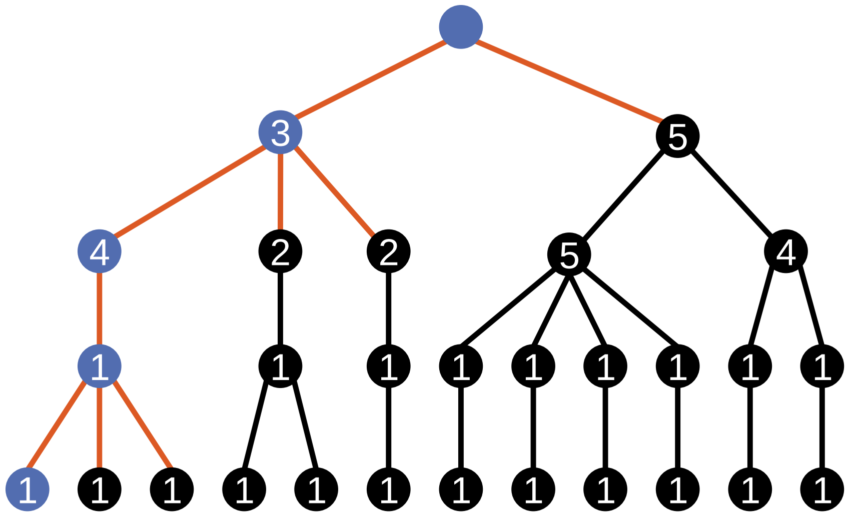
- The reason Knuth's algorithm works is because the estimate for each path is the reciprocal of the probability of that path
- New idea
  - Use a different probability distribution to choose the paths
  - Define the estimate produced to be the reciprocal of the probability used
  - Then we still have

$$\mathbb{E}[\mathrm{est}(p)] = \sum_{p \in P} \mathrm{est}(p) \cdot \mathrm{prob}(p) = \sum_{p \in P} 1 = |P|$$

- To get a different probability distribution…
  - Assign an importance value to each node using an importance function
  - Choose each node with probability proportional to its relative importance amongst its siblings

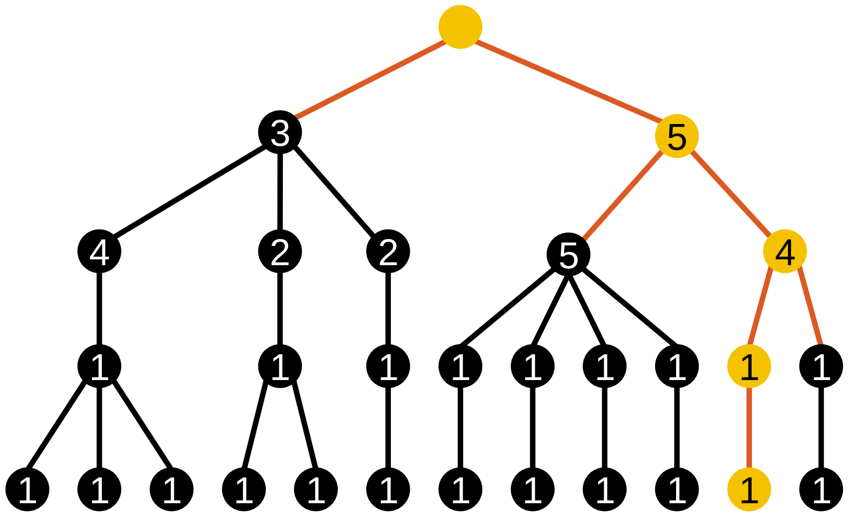
Sequential Importance Sampling Example

- Labeled numbers are the importance values of the nodes
- Blue path probabilities
  - First node $\frac{3}{3+5} = \frac{3}{8}$
  - Second node $\frac{4}{4+2+2} = \frac{1}{2}$
  - Third node 1
  - Fourth node $\frac{1}{1+1+1} = \frac{1}{3}$
  - Complete probability $\frac{3}{8} \cdot \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{16}$
- Blue path estimate is 16
- Better than the previous blue path estimate, 18



Sequential Importance Sampling Example

- Labeled numbers are the importance values of the nodes
- Yellow path probabilities
  - First node $\frac{5}{3+5} = \frac{5}{8}$
  - Second node $\frac{4}{5+4} = \frac{4}{9}$
  - Third node $\frac{1}{1+1} = \frac{1}{2}$
  - Fourth node 1
  - Complete probability $\frac{5}{8} \cdot \frac{4}{9} \cdot \frac{1}{2} = \frac{20}{144}$
- Yellow path estimate is $\frac{144}{20} = 7.2$
- Worse than the previous yellow path estimate, 8



Sequential Importance Sampling Variance

- Sequential importance sampling has the potential to make the variance better, but also the potential to make it worse
- The optimal importance function is the number of leaves beneath a node, which gives zero variance
- The closer the importance function comes to approximating the number of leaves, the better the variance
  - Call the optimal importance function r_* and the actual importance function r
  - Assume both r and r_* have been normalized so the sum of importance for all sibling sets is 1
  - If we always have $\frac{r_*(n)}{r(n)} \leq a_i$ for all nodes n at level i of the tree, then the relative variance will be less than the product of a_i over all levels i of the tree

Ideas for Improving Variance

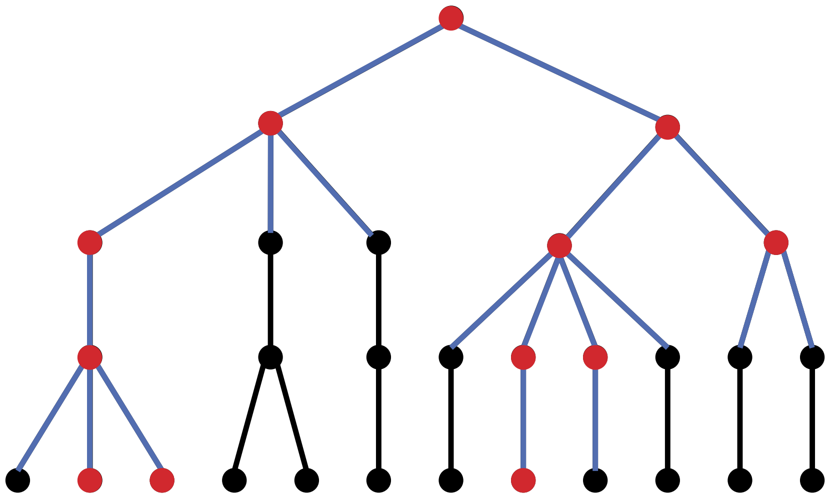
- Better importance functions
  - Use all available information
- Conditioning the input
  - Prune uniform regions of the tree (if any exist) and handle separately
  - Structure the decision process so as to make the tree more uniform
- Improving the algorithm itself
  - Visit more regions of the tree (stratified sampling)
  - Use more paths (stochastic enumeration)

Stochastic Enumeration

- Reuven Rubinstein, 2013
- Given a fixed budget, B, instead of 1 node per level, we select B nodes per level
- At each level, the new set of nodes is chosen with uniform probability from the children of the previous set of nodes
- This means some paths dead end while other paths split
- If there are fewer than B children, we take all the children
- At each level, instead of multiplying the estimate by the number of children, we multiply by the average number of children over all chosen nodes at that level

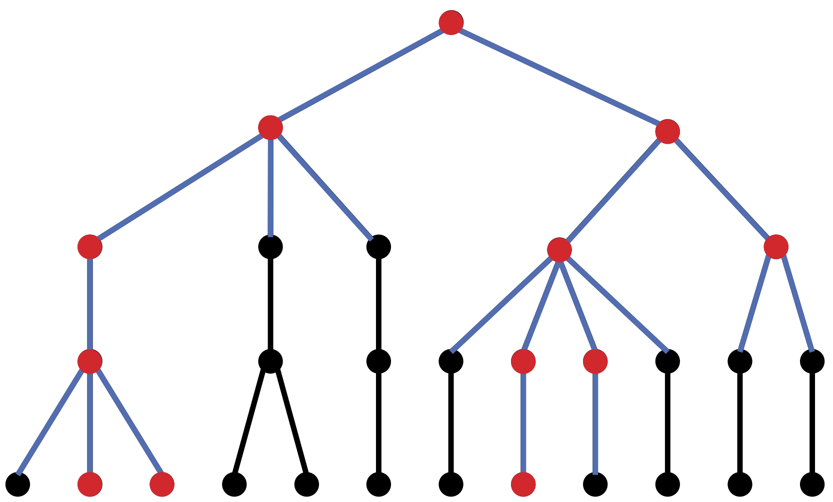
Stochastic Enumeration Example

- Let $B = 3$
- Blue edges = available children
- Red nodes = chosen nodes
- Average number of children per level?
  - Level 1: $\frac{2}{1} = 2$
  - Level 2: $\frac{3+2}{2} = \frac{5}{2}$
  - Level 3: $\frac{1+4+2}{3} = \frac{7}{3}$
  - Level 4: $\frac{3+1+1}{3} = \frac{5}{3}$
- Estimate is $2 \cdot \frac{5}{2} \cdot \frac{7}{3} \cdot \frac{5}{3} = 19.\overline{4}$



Stochastic Enumeration Example

- Probability of selecting these nodes?
  - $1 \cdot 1 \cdot \frac{1}{\binom{5}{3}} \cdot \frac{1}{\binom{7}{3}} \cdot \frac{1}{\binom{5}{3}} = \frac{1}{3500}$
- Estimates and probabilities are no longer reciprocal
- Sample space is now hyperpaths, not paths, so it's no longer in 1-1 correspondence with tree leaves
- Proving estimates are unbiased now requires induction on tree height



Stochastic Enumeration with Importance

- My project's goal: introduce an importance function
- With sequential importance sampling, our estimates look like

$$\prod_i \frac{total\ importance\ of\ available\ nodes\ on\ level\ i}{importance\ of\ chosen\ node\ on\ level\ i}$$

- This worked because it was the reciprocal of the probability with which we chose the nodes
- With stochastic enumeration with an importance function, we want our estimates to look similar

$$\prod_i \frac{total\ importance\ of\ available\ nodes\ on\ level\ i}{importance\ of\ chosen\ nodes\ on\ level\ i}$$

- But the probabilities here are more complicated, so…

Stochastic Enumeration with Importance

- For level i of the tree
  - Let A_i be the set of available nodes to choose from
  - Let $C_i \subset A_i$ be the chosen set of nodes
  - Let r be the importance function
- We want our estimates to look like

$$\prod_i \frac{\sum_{a \in A_i} r(a)}{\sum_{c \in C_i} r(c)}$$

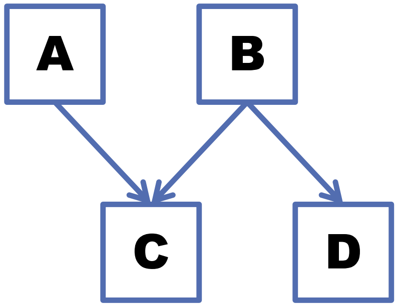
- It turns out this will only be an unbiased estimate if we choose each C_i from A_i with probability

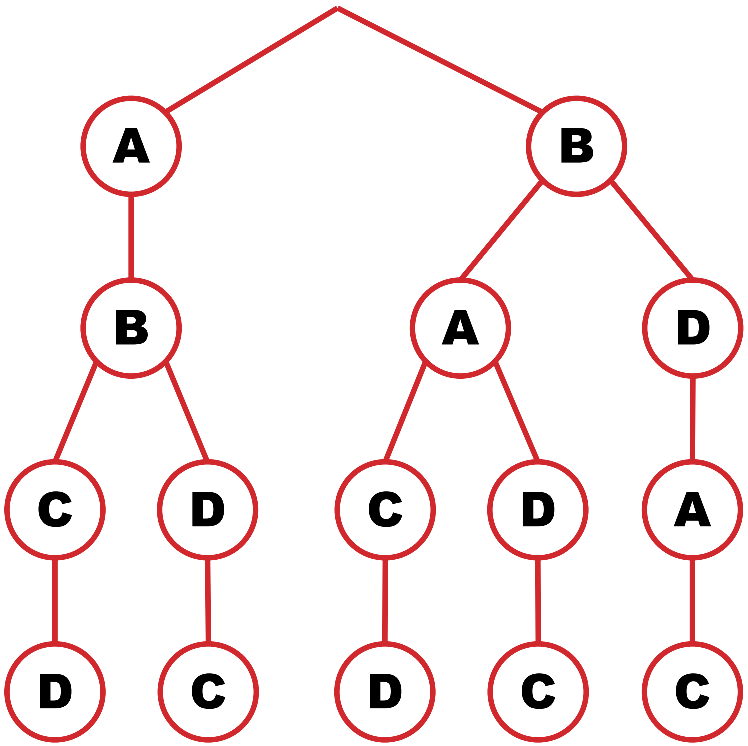
$$\mathrm{prob}(C_i) = \frac{\sum_{c \in C_i} r(c)}{\sum_{a \in A_i} r(a)} \cdot \frac{1}{\binom{|A_i| - 1}{|C_i| - 1}}$$

- To achieve this probability…
  - Choose one element x from A_i with probability proportional to its importance
  - Choose the other elements with uniform probability from the remaining elements in A_i

Numerical Testing

- Applied stochastic enumeration with importance sampling to the problem of counting linear extensions of posets
- A linear extension of a poset is a total ordering of the poset that is consistent with the partial order
- One procedure for getting a linear extension is
  - Select a maximal element, then delete it from the poset
  - Repeat until poset is empty
- We can make a decision tree representing all possible ways to do this





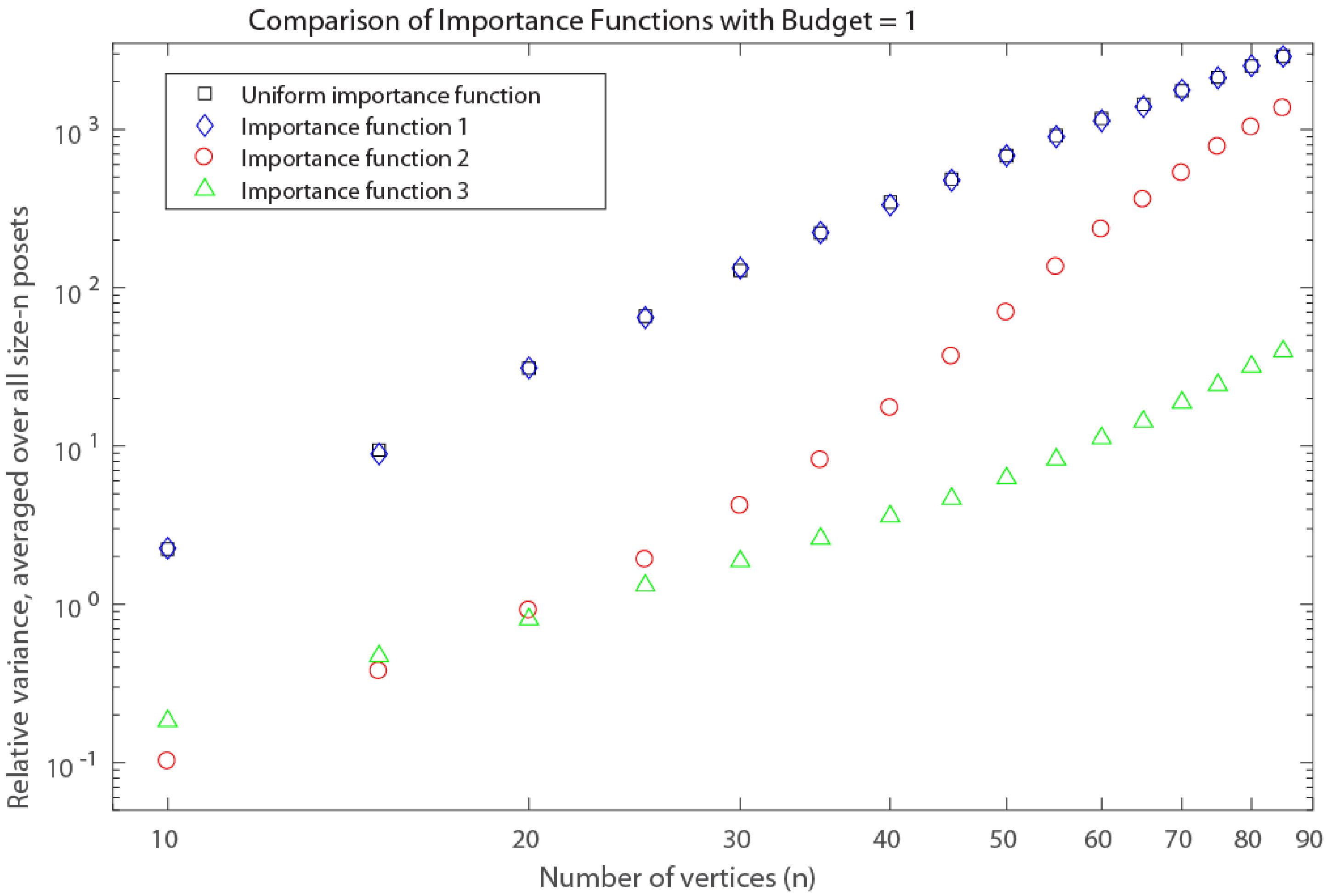
Numerical Testing

- Tested three importance functions and compared them to the uniform importance function
- Notation
  - n is the number of elements in the poset
  - $\mathrm{sib}(x)$ is the number of siblings of node x in the decision tree
  - $\mathrm{desc}(x)$ is the number of descendants of node x in the poset, including x itself
  - $\mathrm{level}(x)$ is the level of node x in the decision tree
- Importance function 1: $r(x) = \mathrm{sib}(x)^3$
- Importance function 2: $r(x) = \mathrm{sib}(x)^3 \cdot \mathrm{desc}(x)$
- Importance function 3: $r(x) = \mathrm{sib}(x)^3 \cdot \frac{n - \mathrm{level}(x) + 1 + \mathrm{desc}(x)}{n - \mathrm{level}(x) + 1 - \mathrm{desc}(x)}$

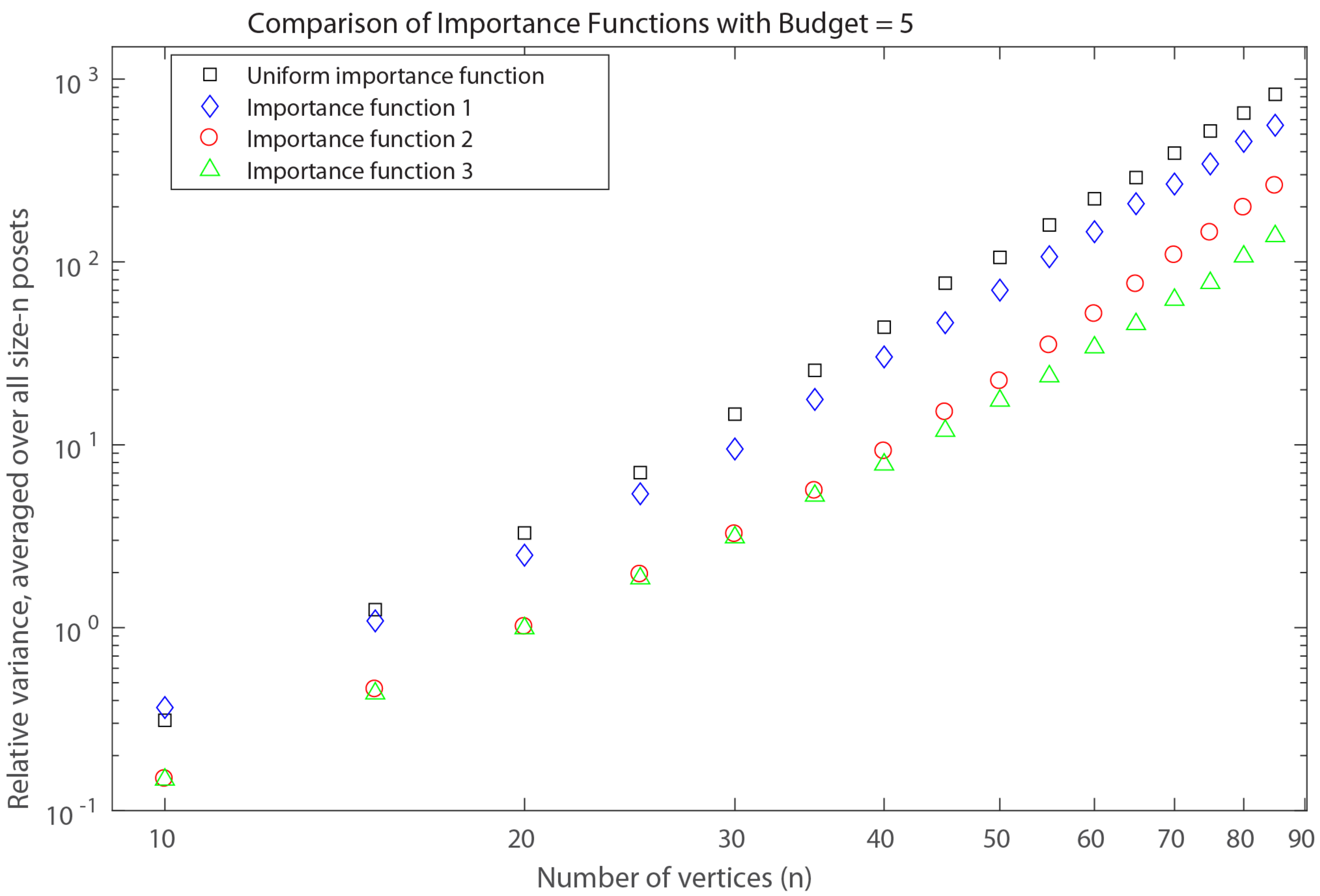
Numerical Testing

- The first set of tests kept B fixed and let n run through the values $n = 10,15,20, \dots, 85$
- For each value of n, n^2 random posets of size n were generated
  - For each pair of poset elements p_i and p_j with $i > j$, the relation $p_i > p_j$ was given a 20% chance to exist
  - The poset was then transitively completed
- n^2 estimates were performed on each poset and relative variance calculated
- Relative variance was averaged for each value of n
- Results are plotted on a log-log scale

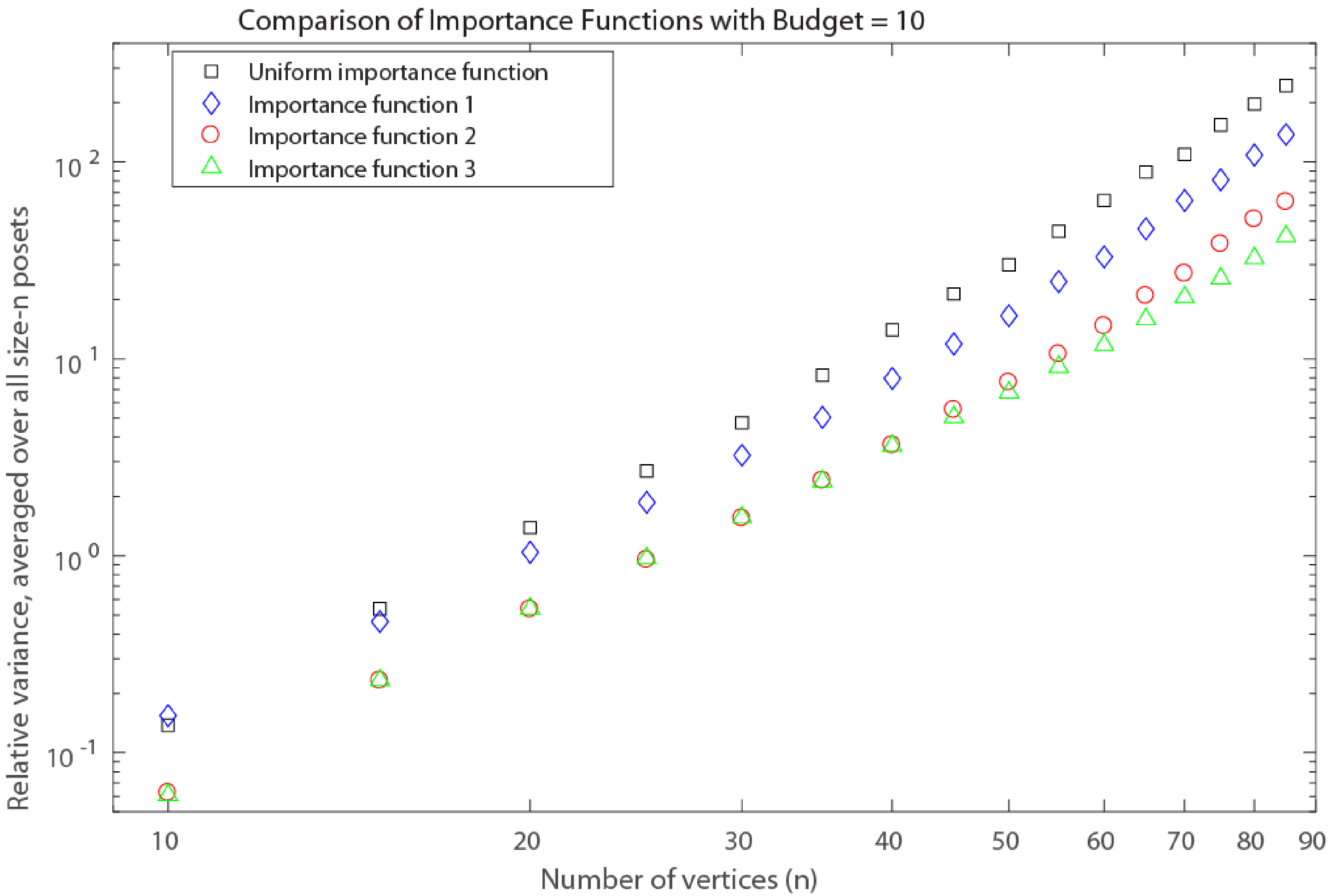
Numerical Results ($B = 1$)



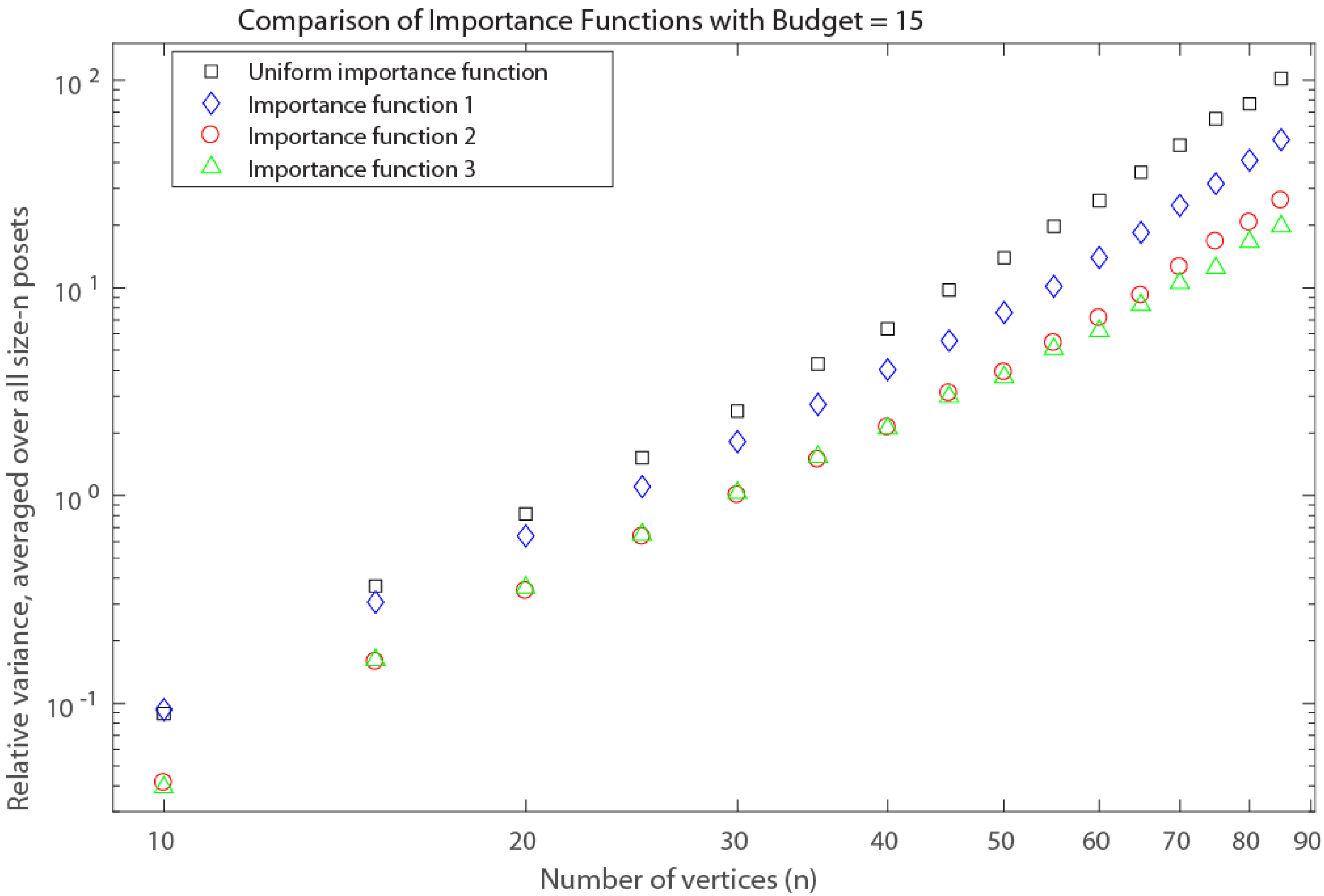
Numerical Results ($B = 5$)



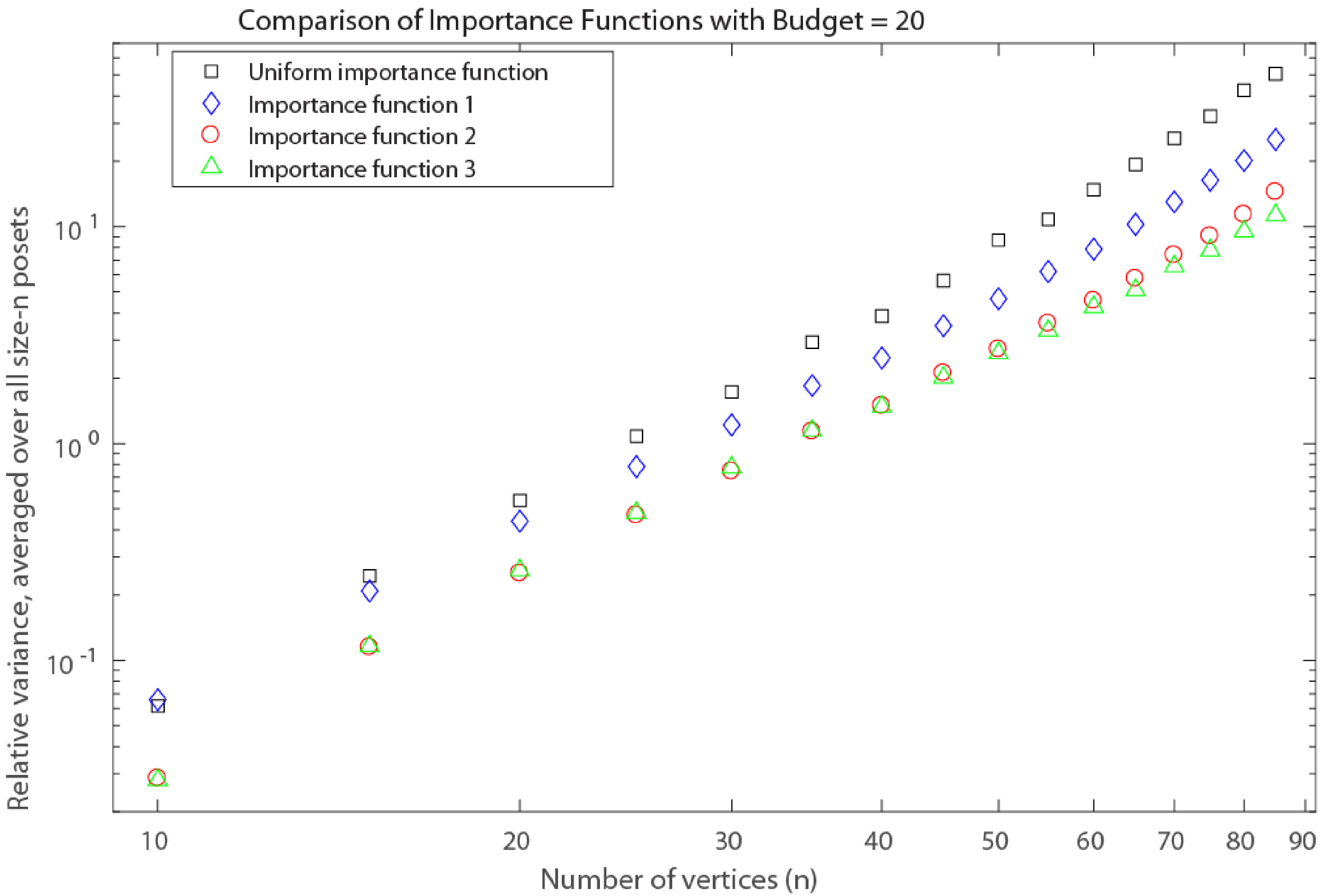
Numerical Results ($B = 10$)



Numerical Results ($B = 15$)



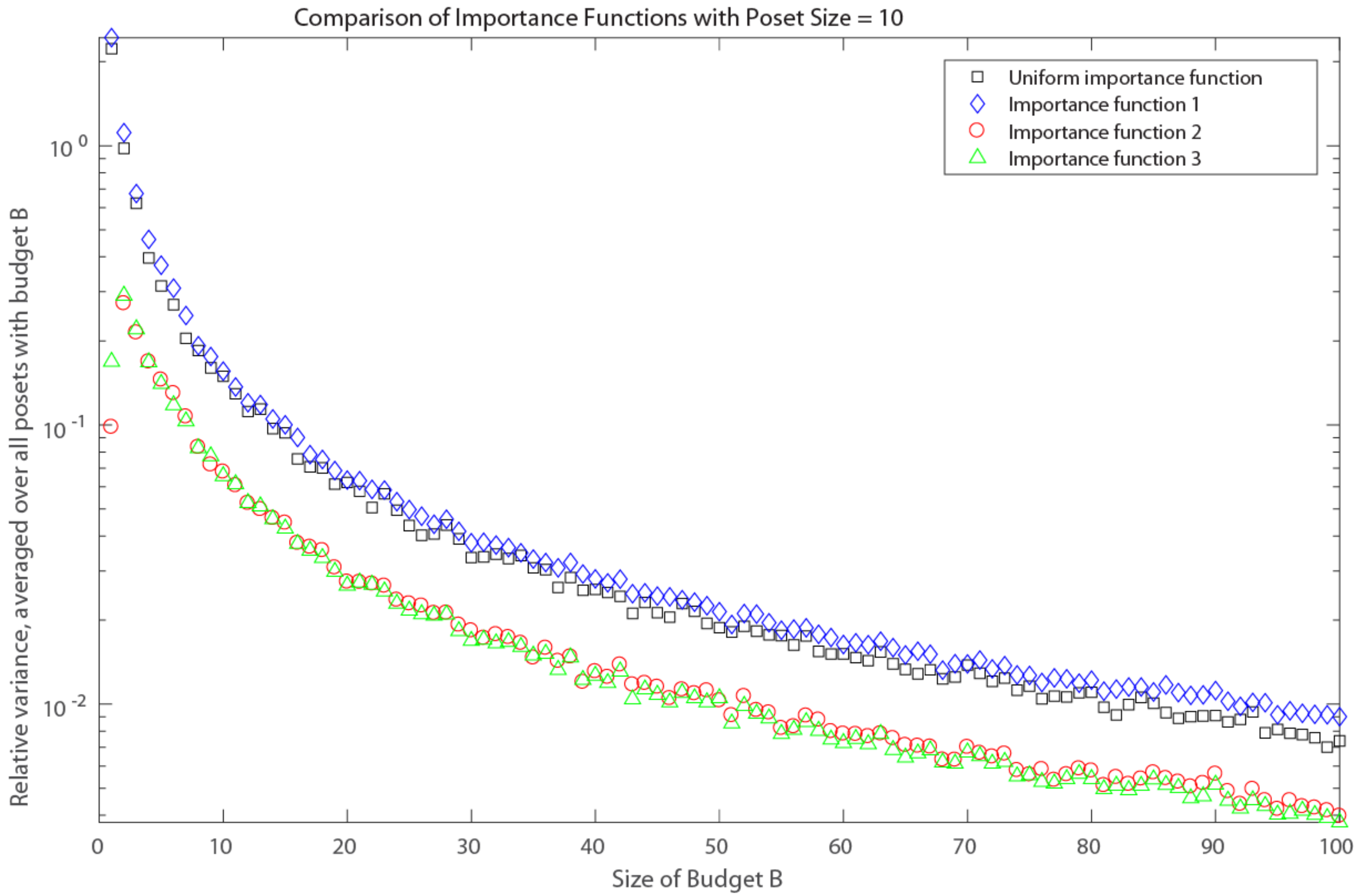
Numerical Results ($B = 20$)



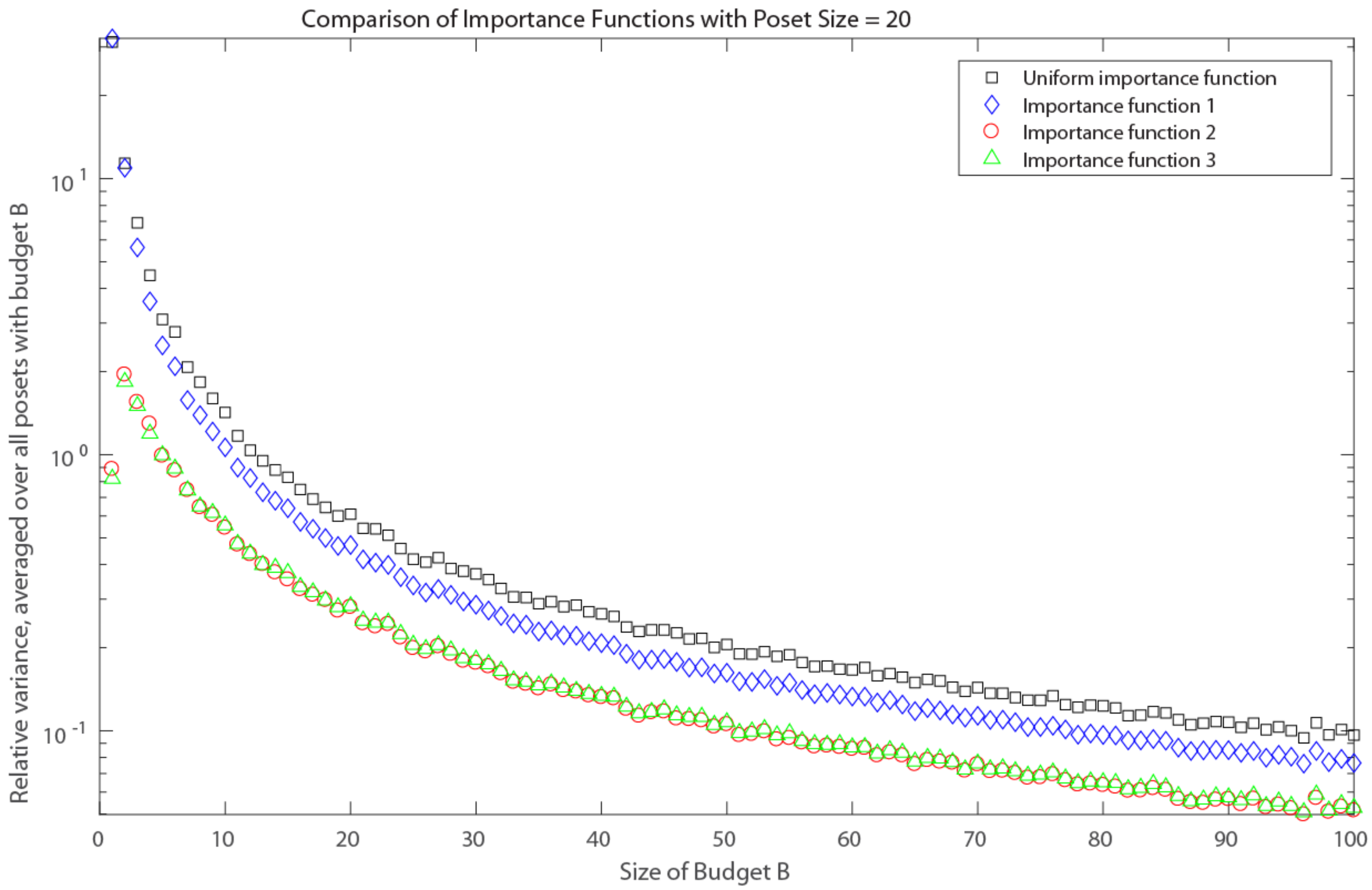
Numerical Testing

- The second set of tests kept n fixed and let B run through the values $B = 1,2,3, \ldots, 100$
- For each value of n, n^2 random posets of size n were generated
  - For each pair of poset elements p_i and p_j with $i > j$, the relation $p_i > p_j$ was given a 20% chance to exist
  - The poset was then transitively completed
- n^2 estimates were performed on each poset and relative variance calculated
- Relative variance was averaged for each value of B
- Results are plotted on a semi-log scale

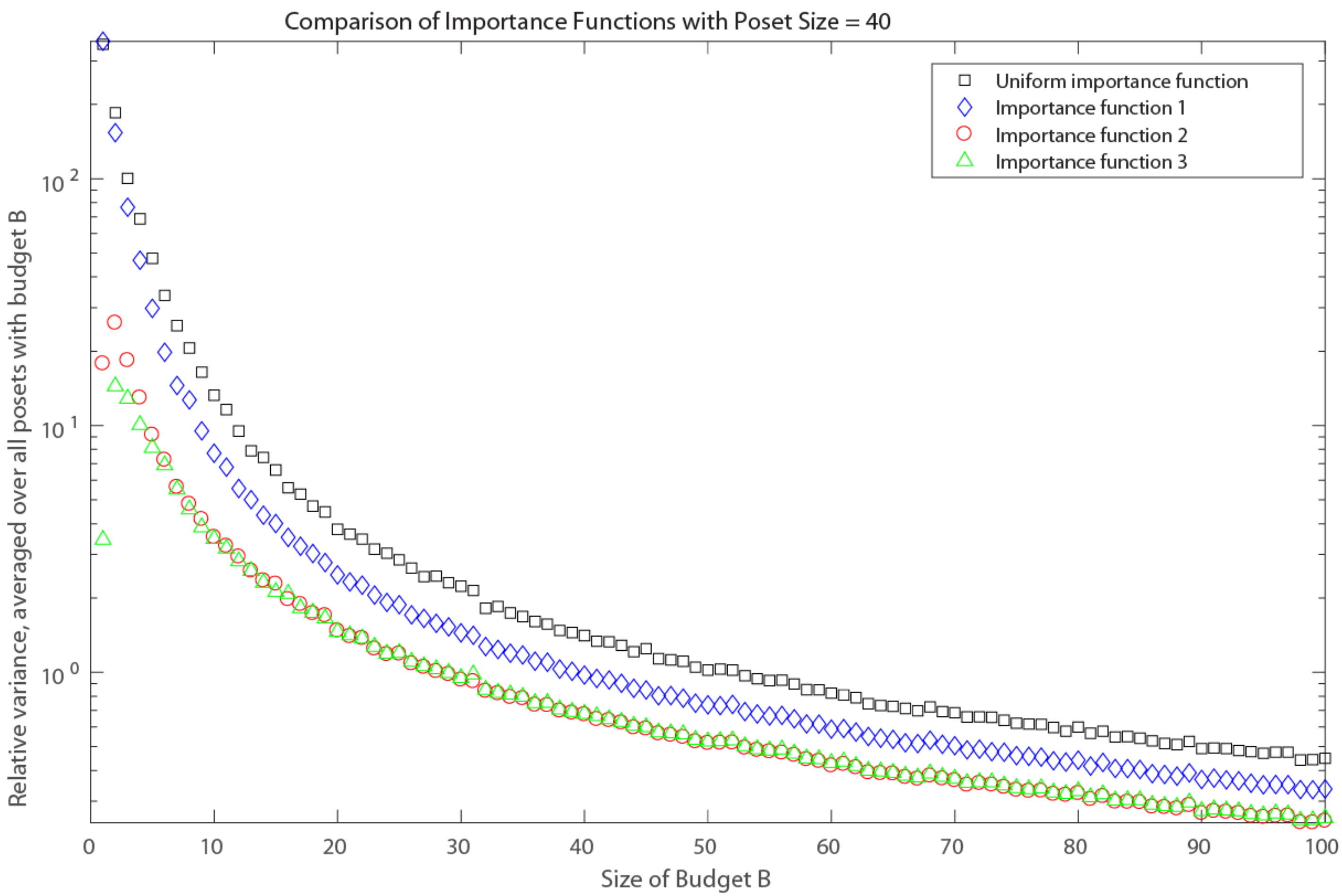
Numerical Results ($n = 10$)



Numerical Results ($n = 20$)



Numerical Results ($n = 40$)



References

- Reuven Rubinstein, *Stochastic enumeration method for counting NP-hard problems*, Methodology and Computing in Applied Probability (2013).
- Radislav Vaisman, Dirk P. Kroese, *Stochastic Enumeration Method for Counting Trees*, Methodology and Computing in Applied Probability (2015).
- Alathea Jensen, *Stochastic Enumeration with Importance Sampling*.