Approximately counting knapsack solutions in sub-quadratic time

Weiming Feng

(The University of Hong Kong)

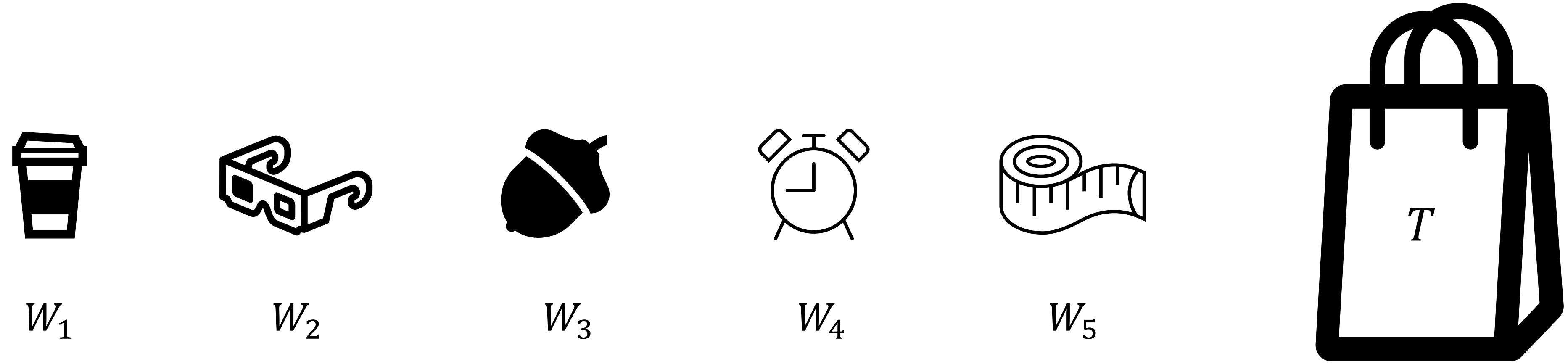
Joint work with
Ce Jin
(MIT)

Pre-AAAC Theory Workshop at HKU
29 May 2025

#Knapsack Problem

Knapsack instance: n items with weights $(W_i)_{i=1}^{n}$ and total capacity $T > 0$

Knapsack solutions: Boolean vector $x \in \{0,1\}^n$ such that $\sum_{1 \le i \le n} W_i x_i \le T$



#Knapsack Problem

Knapsack instance: n items with weights $(W_i)_{i=1}^n$ and total capacity $T > 0$

Knapsack solutions: Boolean vector $x \in \{0,1\}^n$ such that $\sum_{1 \le i \le n} W_i x_i \le T$

#Knapsack $$Z = |\Omega| = \left|\left\{x \in \{0,1\}^n : \sum_{i=1}^{n} W_i x_i \le T\right\}\right|$$

#{Boolean cube points in a half space}

- ***Exact counting***: #P-complete problem
- ***Approximate counting***: given any $\epsilon > 0$, in poly$\left(\text{input size}, \frac{1}{\epsilon}\right)$ time, output
  - **FPTAS**: a number $\hat{Z}$ s.t. $(1-\epsilon)Z \le \hat{Z} \le (1+\epsilon)Z$
  - **FPRAS**: a random number $\hat{Z}$ s.t. $\Pr\left[(1-\epsilon)Z \le \hat{Z} \le (1+\epsilon)Z\right] \ge \frac{2}{3}$

Known results

Sampling

Jerrum, Valiant, Vazirani, 1986

Approximate Counting

Sampling: draw ***uniform*** random sample from $\Omega = \{\, x \in \{0,1\}^n \mid \sum_{1\le i\le n} W_i x_i \le T \,\}$

Markov chain Monte Carlo sampling (***MCMC***) algorithm

- Dyer, Frieze, Kannan, Kapoor, Perkovic, and Vazirani 1993: $2^{O(\sqrt{n}(\log n)^{2.5})}$
- Morris and Sinclair 1999/2004: $O_\delta\left(n^{4.5+\delta}\right)$ for any $\delta > 0$

JVV86 reduction

$$T_{\text{count}} = T_{\text{sample}} \cdot O(n^2/\epsilon^2)$$

Rounding + Dynamic Programming + Rejection Sampling [Dyer 2003]: counting in time $\tilde{O}(n^{2.5} + n^2/\epsilon^2)$

FPRAS randomized

Dynamic Programming based counting algorithms

- Gopalan, Klivans, Meka, Štefankovič, Vempala, and Vigoda 2011: $\tilde{O}(n^3/\epsilon)$
- Gawrychowski, Markin, and Weimann 2018: $\tilde{O}(n^{2.5}/\epsilon)$

FPTAS deterministic

Our Result

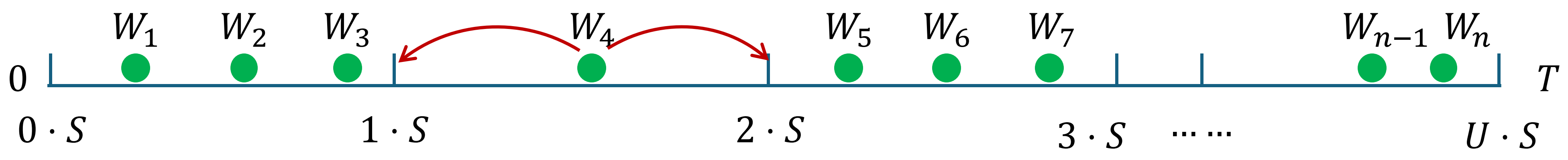
There is an ***FPRAS*** for #Knapsack in time $O\left(\frac{n^{1.5}}{\epsilon^2}\text{polylog}\left(\frac{n}{\epsilon}\right)\right)$

- Our algorithm works in ***standard word-RAM model*** with word length $\text{polylog}\left(\frac{n}{\epsilon}\right)$ bits
- We assume all W_i and T has bit length $\text{polylog}(n)$;
  otherwise the running time will be multiplied by $O(\log T)$ factor
- The dependency to n is ***sub-quadratic***, which is ***rare*** in approximate counting
  most approximate counting algorithms take quadratic time (due to counting-to-sampling reduction)
  but there some exceptions e.g. spanning trees, network unreliability, special spin systems
- The dependency to $\frac{1}{\epsilon}$ is ***near-quadratic***, which is ***common*** for Monte Carlo algorithms

Dyer's algorithm

Assume $0 < W_1 \le W_2 \le \cdots \le W_n \le T$

Dyer's algorithm



- Set the ***scale*** parameter $S = \frac{T}{2Kn}$, where $K = \sqrt{n} \cdot \text{polylog}\left(\frac{n}{\epsilon}\right)$
- For each $i \in [n]$, round W_i to a ***random nearest multiple*** $w_i \in \left\{\left\lfloor \frac{W_i}{S} \right\rfloor S, \left\lceil \frac{W_i}{S} \right\rceil S\right\}$ of S such that

$$\mathbb{E}[w_i] = W_i$$

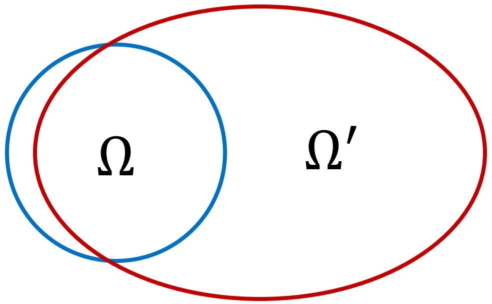
- Round capacity T to a $T' = T + KS$

Original solution space Ω

$$X \in \{0,1\}^n \text{ s.t. } W(X) = \sum_{i=1}^{n} W_i x_i \le T$$

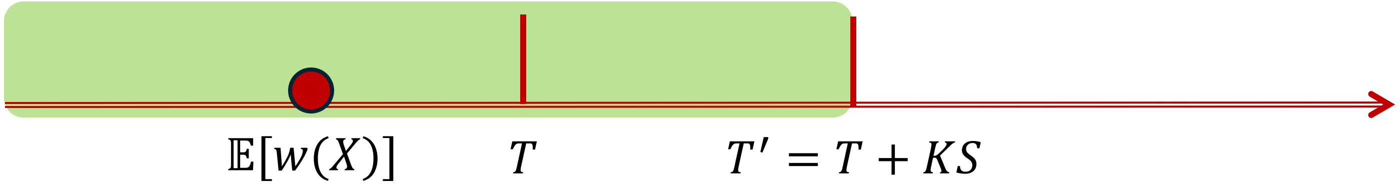
Rounded random solution space Ω'

$$X \in \{0,1\}^n \text{ s.t. } w(X) = \sum_{i=1}^{n} w_i x_i \le T'$$



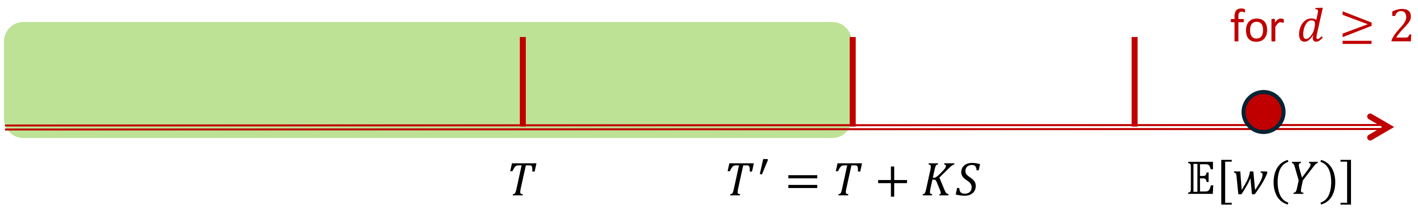
- For any ***solution*** $X \in \Omega$, w.h.p., $X \in \Omega'$ ⟹ $\Omega' \cap \Omega \approx \Omega$

$$\mathbb{E}[w(X)] = W(X) \le T$$



- For any ***non-solution*** $Y \in \Omega_d$ s.t. Y needs to throw d heaviest items to make it in Ω, then

$$\mathbb{E}[w(Y)] = W(Y) \ge T + \frac{d-1}{n} \cdot T$$



We can ignore $Y \in \Omega_d$ for $d \ge 2$. The only ***missing part*** is $Y \in \Omega_1$ but $|\Omega_1| \le n \cdot |\Omega|$. Overall,

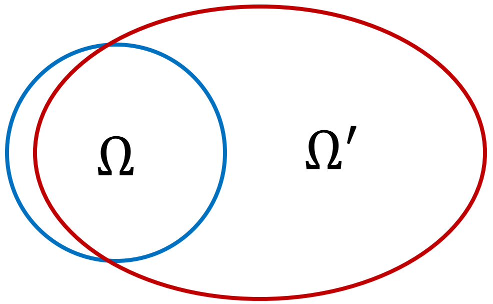
$$|\Omega' \setminus \Omega| \le O(n) \cdot |\Omega|$$

Original solution space Ω

$$X \in \{0,1\}^n \text{ s.t. } W(X) = \sum_{i=1}^{n} W_i x_i \leq T$$

Rounded random solution space Ω'

$$X \in \{0,1\}^n \text{ s.t. } w(X) = \sum_{i=1}^{n} w_i x_i \leq T'$$



- $\Omega' \cap \Omega \approx \Omega$
- $|\Omega' \setminus \Omega| \leq O(n) \cdot |\Omega|$

- We can approximately count $|\Omega \cap \Omega'| \approx |\Omega|$
- Exact count $|\Omega'|$ by ***dynamic programming***, because all w_i is a ***multiple*** of S

Running time: $O\left(n \cdot \frac{T'}{S}\right) = O\left(n \cdot \frac{T+KS}{T/(2Kn)}\right) = O(n^2 K) = \tilde{O}(n^{2.5})$

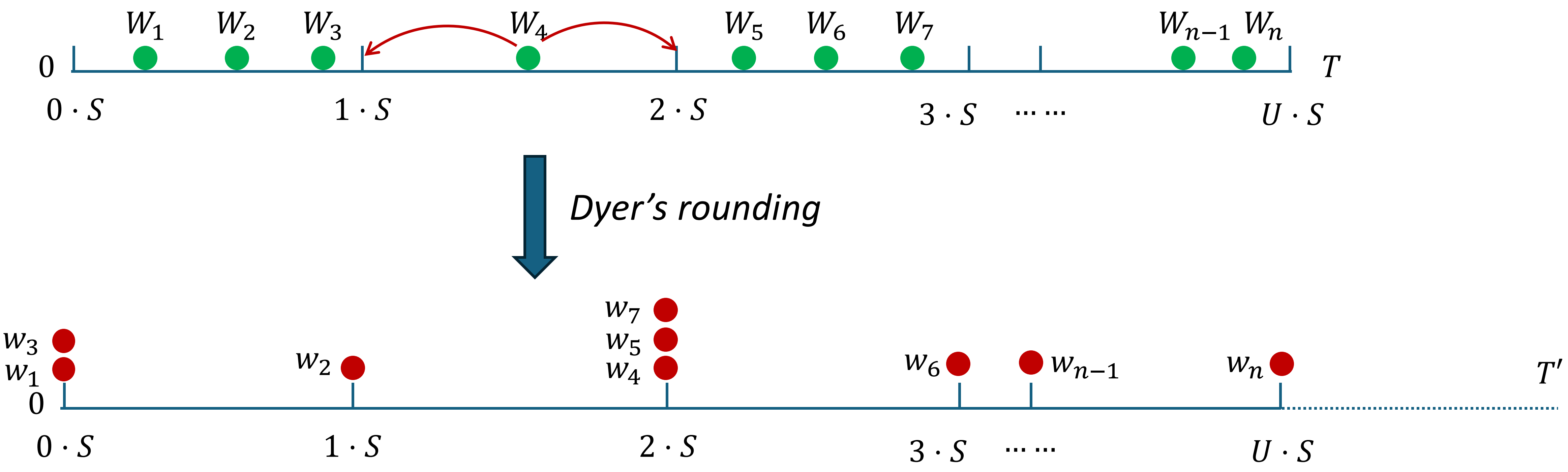
- Draw $O\left(\frac{n}{\epsilon^2}\right)$ ***uniform random sample*** X from Ω' and test whether $X \in \Omega$ to approximate $\frac{|\Omega \cap \Omega'|}{|\Omega'|} \approx \frac{|\Omega|}{|\Omega'|} = \Omega\left(\frac{1}{n}\right)$

Running time: $O\left(\frac{n^2}{\epsilon^2}\right)$

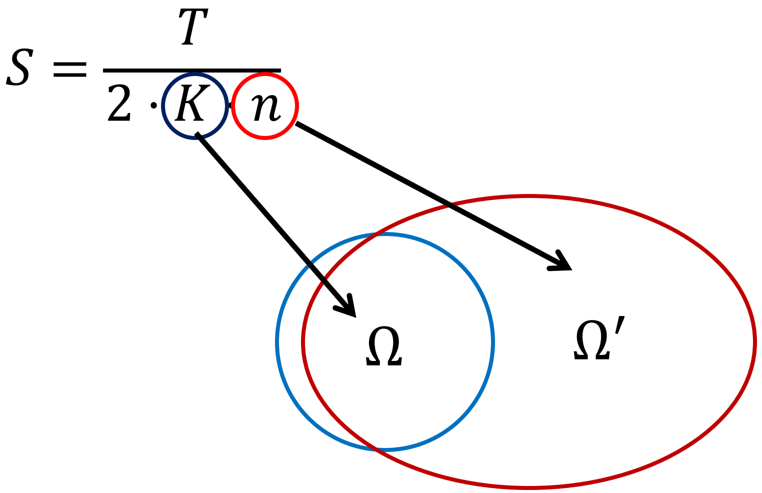
- Output: $|\Omega \cap \Omega'| = \frac{|\Omega \cap \Omega'|}{|\Omega'|} \cdot |\Omega'|$

Bounded-ratio case

- Capacity T and n items with weights $(W_i)_{i=1}^n$, where for any $W_i \in \left(\frac{T}{\ell}, \frac{2T}{\ell}\right]$ and $2 \leq \ell \leq 2n$
- Approximately count $Z = |\Omega| = |\{x \in \{0,1\}^n \mid \sum_{i=1}^n W_i x_i \leq T\}|$ with error ϵ. Assume $\epsilon = 10^{-5}$ for simplicity.



Dyer's rounding: the scale parameter is $S = \frac{T}{2Kn}$ and new capacity $T' = T + KS$ for $K = \widetilde{\Theta}(\sqrt{n})$



- $\Omega' \cap \Omega \approx \Omega$
- $|\Omega' \setminus \Omega| \leq O(n) \cdot |\Omega|$

Dyer's rounding: the scale parameter is $S = \frac{T}{2Kn}$ and new capacity $T' = T + KS$ for $K = \widetilde{\Theta}(\sqrt{n})$

- for any solution $X \in \Omega$, $|X| \le n$, we require $X \in \Omega'$, the total rounding error is at most

$$\textbf{Rounding Error} \le \underbrace{\sum_{i \in X} (w_i - W_i)}$$

Sum of $\le n$ independent random variables in range $[-S, S]$

Dyer's rounding: the scale parameter is $S = \frac{T}{2Kn}$ and new capacity $T' = T + KS$ for $K = \widetilde{\Theta}(\sqrt{n})$

- for any solution $X \in \Omega$, $|X| \le n$, we require $X \in \Omega'$, the total rounding error is at most

$$\textbf{Rounding Error} \le \underbrace{\sum_{i \in X} (w_i - W_i)}_{\text{Sum of } \le n \text{ independent random variables in range } [-S, S]} =_{\text{whp}} S \cdot \tilde{O}(\sqrt{n})$$

Sum of $\le n$ independent random variables in range $[-S, S]$

Dyer's rounding: the scale parameter is $S = \frac{T}{2Kn}$ and new capacity $T' = T + KS$ for $K = \widetilde{\Theta}(\sqrt{n})$

- for any solution $X \in \Omega$, $|X| \le n$, we require $X \in \Omega'$, the total rounding error is at most

$$\textbf{Rounding Error} \le \underbrace{\sum_{i \in X} (w_i - W_i)}_{\text{Sum of } \le n \text{ independent random variables in range } [-S, S]} =_{\text{whp}} S \cdot \tilde{O}(\sqrt{n}) \le T' - T \ \textbf{= Slack of Capacity}$$

$$= KS = \widetilde{\Theta}(S\sqrt{n})$$

Bounded ratio case: $W_i \in \left(\frac{T}{\ell}, \frac{2T}{\ell}\right]$ ⟹ for any solution $X \in \Omega$, $|X| \le \ell$

In the bounded ratio case, one can round ***more aggressively*** by setting a ***larger scale***

$S \approx \frac{T}{n\sqrt{n}}$ ⟹ $S \approx \frac{T}{\ell\sqrt{\ell}}$

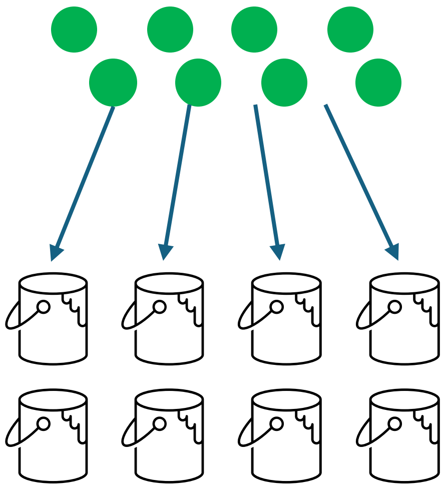
Such simple improvement does not help if $\ell = \Theta(n)$

Our algorithm: bounded-ratio case

Bounded ratio case: $W_i \in \left(\frac{T}{\ell}, \frac{2T}{\ell}\right]$ ⟹ for any solution $X \in \Omega$, $|X| \le \ell$

Step-I: **Balls-into-Bins Hashing** [Bringmann 2017]

n items (balls)



ℓ bins: $B_1, B_2, \ldots, B_\ell$

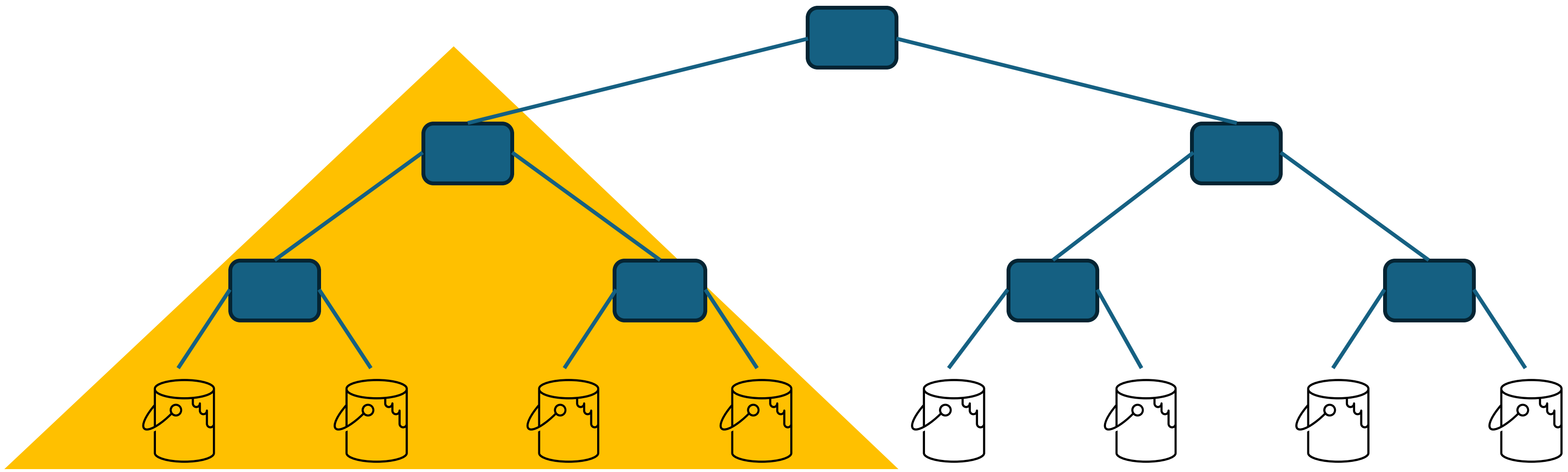
For each item $i \in [n]$, sample a bin $j \in [\ell]$ u.a.r., and throw i to B_j

For any solution $X \in \Omega$, w.h.p. $|X \cap B_i| \le B = O\left(\frac{\log n}{\log \log n}\right)$

Define collection of ***good subsets*** $\widehat{\Omega} = \{X \subseteq [n] \mid \forall b \in [\ell], |B_b \cap X| \le B\}$

⟹ $|\Omega| \approx \left|\widehat{\Omega} \cap \Omega\right|$ ⟸ **Approximately Count**

Step-II: partition-and-convolve with multi-level rounding



For every node u at level h, define $B_u = \cup_k B_k$ for ***leaves*** k in subtree rooted at u

The node u (implicitly) defines a ***random weight*** $w_u : \widehat{\Omega} \cap 2^{B_u} \to \mathbb{R}_{\geq 0}$ such that $\forall X \in \widehat{\Omega} \cap 2^{B_u}$,

- $w_u(X)$ is a ***multiple*** of the ***scale*** $S_h \approx \frac{T}{\ell \cdot 2^{h/2}}$ such that $0 \leq w_u(X) \leq L_h S_h$
- $\mathbb{E}[w_u(X)] = W(X) = \sum_{i \in X} W_i$ and $w_u(X)$ is ***concentrated*** around its mean

The node u explicitly ***maintains*** $f_u : \{0S_h, 1S_h, 2S_h, \ldots L_h S_h\} \to \mathbb{R}_{\geq 0}$ such that $\forall\ 0 \leq i \leq L_h$,

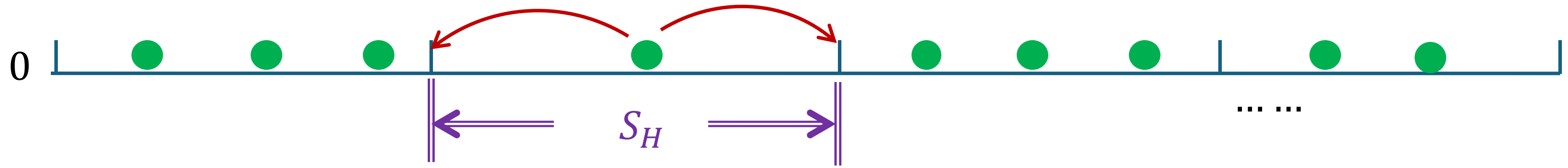
$$\underbrace{\sum_{j \leq i} f_u(jS_h)}_{\text{Prefix Sum to } i} \approx \left|\left\{X \in \widehat{\Omega} \cap 2^{B_u} \mid w_u(X) \leq iS_h\right\}\right|$$

Prefix Sum to i

Base Case: Rounding at leaf nodes u

For leaf node, set ***the scale*** $S_H \approx \frac{T}{\ell\sqrt{\ell}}$, where $H = \log \ell$

For each item $i \in B_u$, round W_i to $w_u(i) = \left\{ S_H \left\lfloor \frac{W_i}{S_H} \right\rfloor, S_H \left\lceil \frac{W_i}{S_H} \right\rceil \right\}$ such that $\mathbb{E}[w_u(i)] = W_i$



$$\forall \text{ good subset } X \in \widehat{\Omega} \cap 2^{B_u}, \qquad w_u(X) = \sum_{i \in X} w_u(i)$$

Compute the function $f_u: \{0, S_H, 2S_H, \dots, L_H S_H\} \to \mathbb{R}_{\geq 0}$, ***via dynamic programming***,

$M(i, j, k)$: ***how many*** subsets X of first i items with $|X| = j$ and total weight $k \cdot S_H$

$i \leq |B_u|$

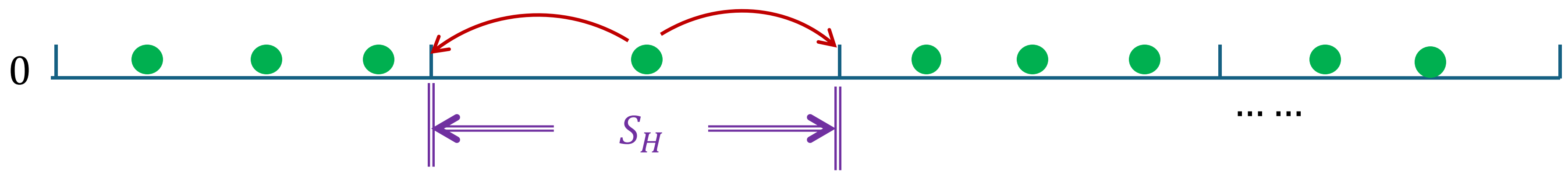
$j = O\left(\frac{\log n}{\log \log n}\right)$

$k \leq \frac{2T}{\ell S_H} \cdot B = \tilde{O}(\sqrt{\ell})$

Base Case: Rounding at leaf nodes u

For leaf node, set ***the scale*** $S_H \approx \frac{T}{\ell\sqrt{\ell}}$, where $H = \log \ell$

For each item $i \in B_u$, round W_i to $w_u(i) = \left\{S_H \left\lfloor \frac{W_i}{S_H} \right\rfloor, S_H \left\lceil \frac{W_i}{S_H} \right\rceil\right\}$ such that $\mathbb{E}[w_u(i)] = W_i$



$$\forall \text{ good subset } X \in \widehat{\Omega} \cap 2^{B_u}, \qquad w_u(X) = \sum_{i \in X} w_u(i)$$

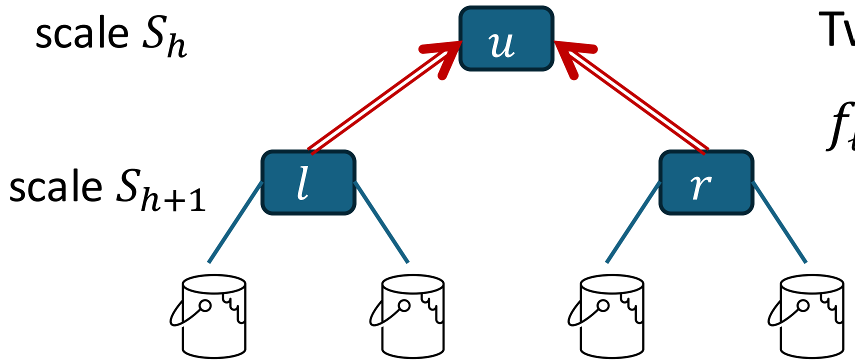
Compute the function $f_u: \{0, S_H, 2S_H, \dots, L_H S_H\} \to \mathbb{R}_{\geq 0}$, ***via dynamic programming***, in time

$$O\left(|B_u| \cdot B \cdot \frac{T}{\ell \cdot S_H}\right) = \tilde{O}(|B_u| \cdot \sqrt{\ell})$$

The total complexity contributed by all leaf nodes is

$$\sum_u \tilde{O}\left(|B_u| \cdot \sqrt{\ell}\right) = \tilde{O}\left(n\sqrt{\ell}\right) = \tilde{O}(n^{1.5})$$

Induction Step: convolution and rounding



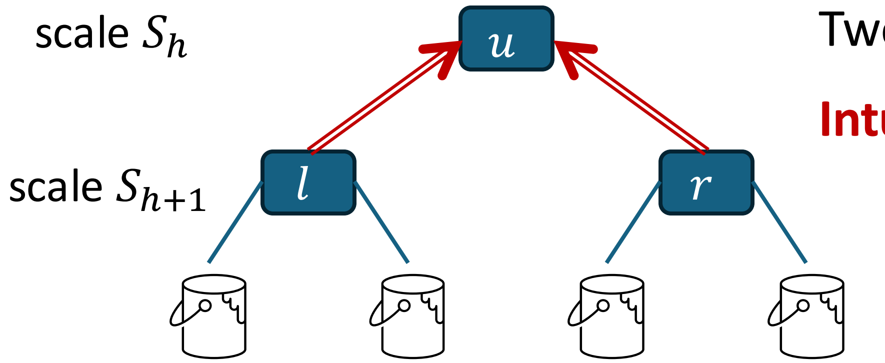
Two functions $f_l, f_r: \{0, S_{h+1}, 2S_{h+1}, \ldots, L_{h+1}S_{h+1}\} \to \mathbb{R}$ at nodes l and r

f_ℓ achieves ***prefix-sum approximation*** such that for any i

$$\sum_{j \le i} f_u(jS_{h+1}) \approx \left|\left\{X \in \widehat{\Omega} \cap 2^{B_\ell} \mid w_\ell(X) \le iS_{h+1}\right\}\right|$$

approximate the number of subsets with weight $\le iS_{h+1}$

Induction Step: convolution and rounding



Two functions $f_l, f_r : \{0, S_{h+1}, 2S_{h+1}, \ldots, L_{h+1}S_{h+1}\} \to \mathbb{R}$ at nodes l and r

Intuitively, one may think that

$$f_\ell(jS_{h+1}) \approx \left|\{X \in \widehat{\Omega} \cap 2^{B_\ell} \mid w_\ell(X) = jS_{h+1}\}\right|$$

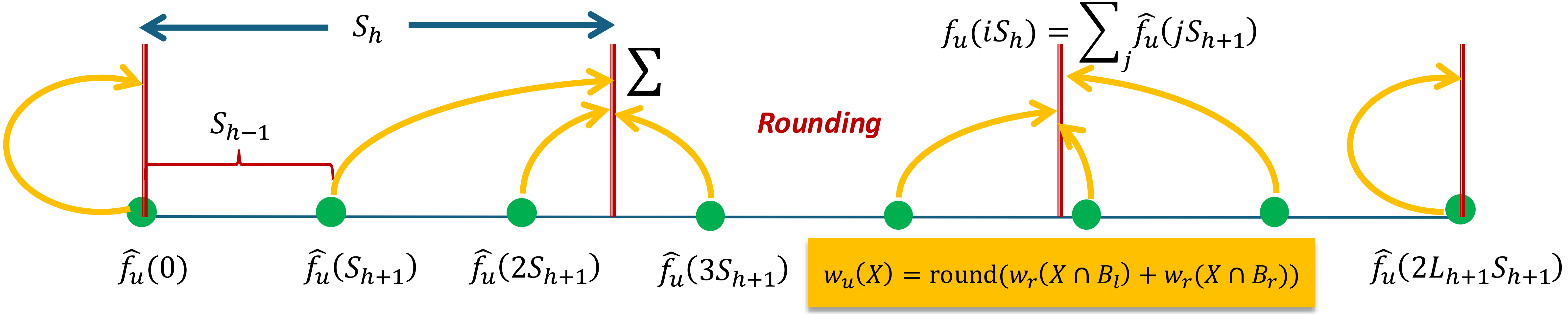
for the approximation measured by the prefix-sum

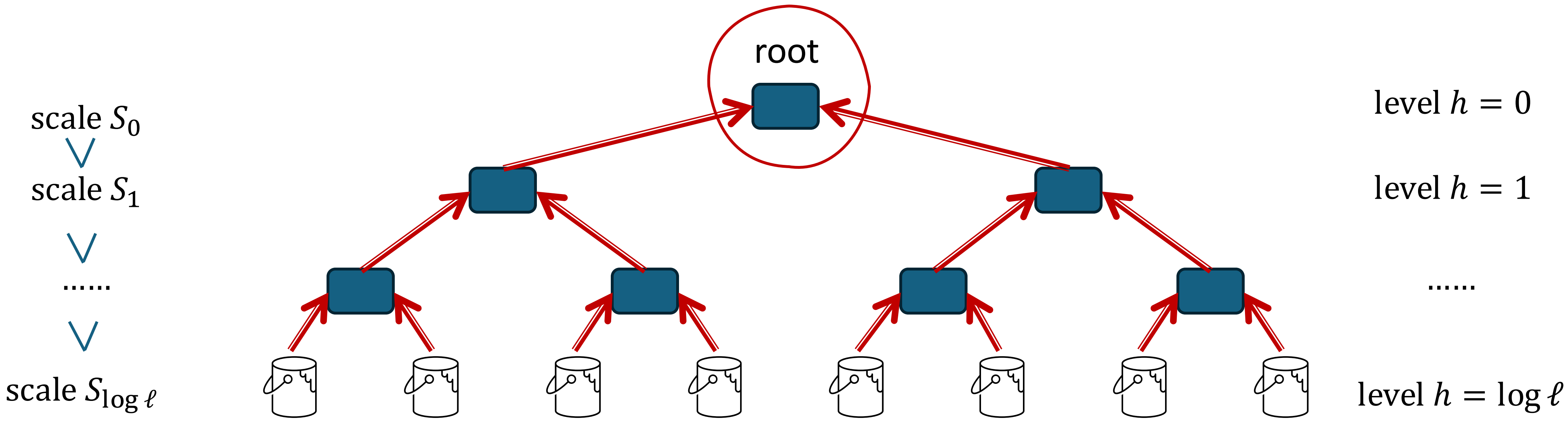
$\widehat{f_u} = f_l * f_r$ is a ***convolution***

$$f_u(xS_{h+1}) = \sum_{0 \le i \le x} f_l(iS_{h+1}) f_r((x-i)S_{h+1})$$

- We only require $f_\ell * f_r$ achieve ***prefix-sum approximation***
- The values in f_ℓ and f_r are at most $n^{O(2^{H-h}\log n)} = 2^M$

➡ The approximate convolution $\widehat{f_u}$ takes time $\tilde{O}(L_{h+1}\sqrt{M})$





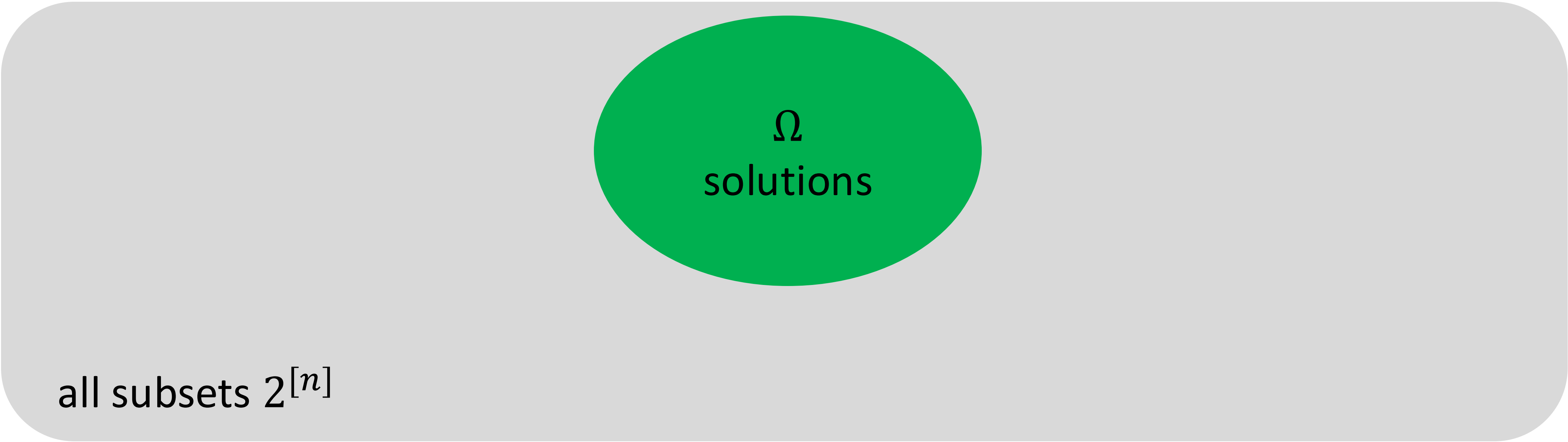
- Compute the function f_u for each node from ***bottom to up***
- The total computational complexity is $\tilde{O}(n^{1.5})$

- Root defines a ***random weight function***

$$\forall X \in \widehat{\Omega}, \qquad \mathbb{E}[w_{\text{root}}(X)] = W(X) \text{ and w.h.p.}\, w_{\text{root}}(X) \approx W(X) \pm \frac{T}{\ell}$$

- The ***set of solutions*** at the root

$$\Omega' = \left\{ X \in \widehat{\Omega} \,\middle|\, w_{\text{root}}(X) \le T + \tilde{O}\left(\frac{T}{\ell}\right) \right\}$$

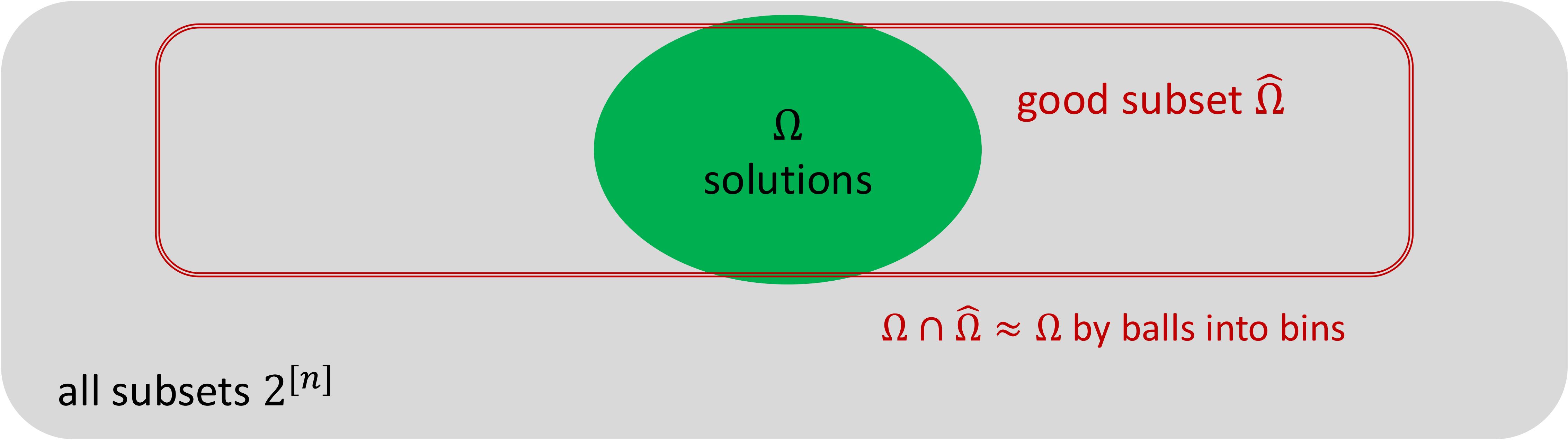


- Root defines a ***random weight function***

$$\forall X \in \widehat{\Omega}, \qquad \mathbb{E}[w_{\mathrm{root}}(X)] = W(X) \text{ and w.h.p. } w_{\mathrm{root}}(X) \approx W(X) \pm \frac{T}{\ell}$$

- The ***set of solutions*** at the root

$$\Omega' = \left\{ X \in \widehat{\Omega} \;\middle|\; w_{\mathrm{root}}(X) \leq T + \tilde{O}\left(\frac{T}{\ell}\right) \right\}$$

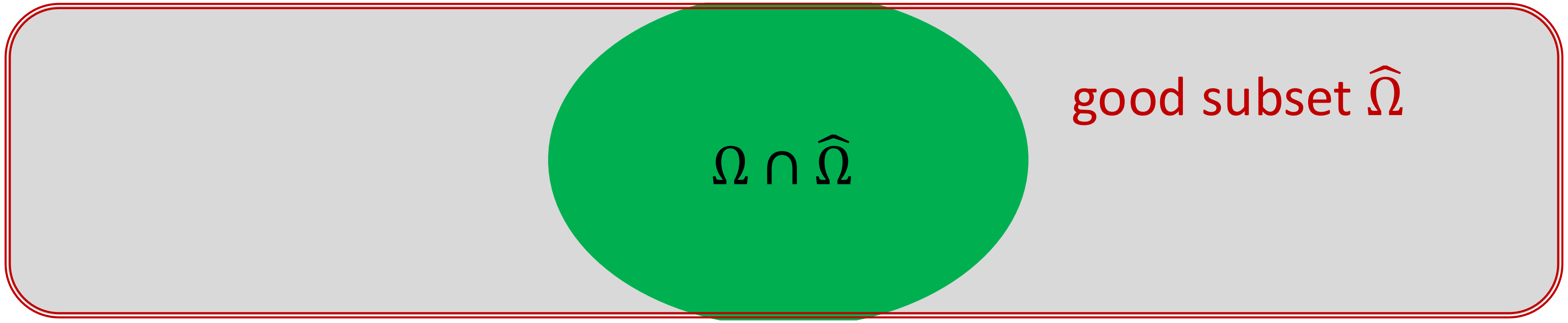


- Root defines a ***random weight function***

$$\forall X \in \widehat{\Omega}, \qquad \mathbb{E}[w_{\mathrm{root}}(X)] = W(X) \text{ and w.h.p. } w_{\mathrm{root}}(X) \approx W(X) \pm \frac{T}{\ell}$$

- The ***set of solutions*** at the root

$$\Omega' = \left\{ X \in \widehat{\Omega} \;\middle|\; w_{\mathrm{root}}(X) \le T + \tilde{O}\left(\frac{T}{\ell}\right) \right\}$$

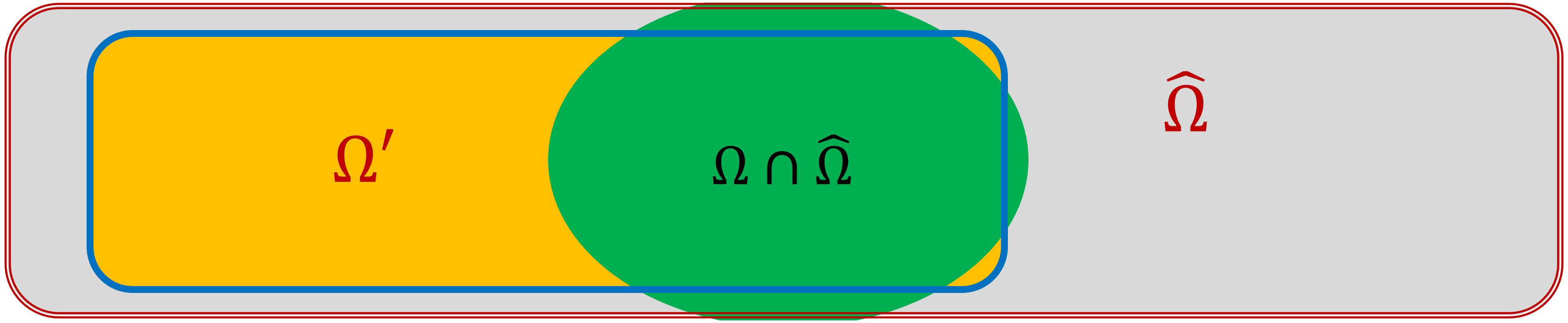


- Root defines a ***random weight function***

$$\forall X \in \widehat{\Omega}, \qquad \mathbb{E}[w_{\text{root}}(X)] = W(X) \text{ and w.h.p. } w_{\text{root}}(X) \approx W(X) \pm \frac{T}{\ell}$$

- The ***set of solutions*** at the root

$$\Omega' = \left\{ X \in \widehat{\Omega} \,\middle|\, w_{\text{root}}(X) \leq T + \tilde{O}\left(\frac{T}{\ell}\right) \right\}$$

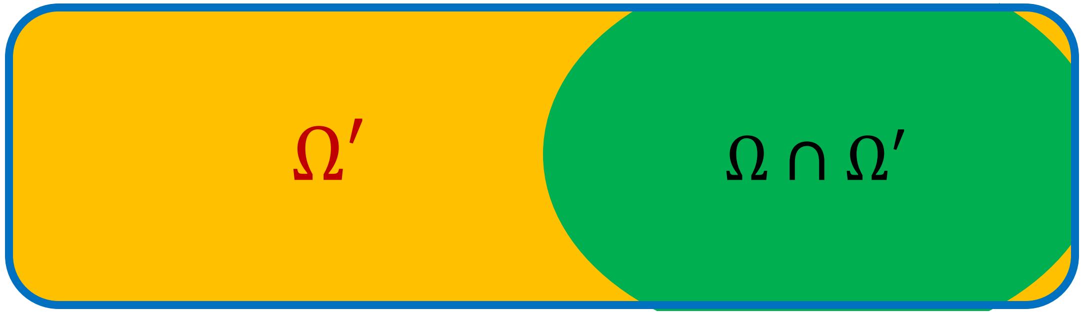


- Root defines a ***random weight function***

$$\forall X \in \widehat{\Omega}, \qquad \mathbb{E}[w_{\mathrm{root}}(X)] = W(X) \text{ and w.h.p. } w_{\mathrm{root}}(X) \approx W(X) \pm \frac{T}{\ell}$$

- The ***set of solutions*** at the root

$$\Omega' = \left\{ X \in \widehat{\Omega} \,\middle|\, w_{\mathrm{root}}(X) \le T + \tilde{O}\left(\frac{T}{\ell}\right) \right\}$$

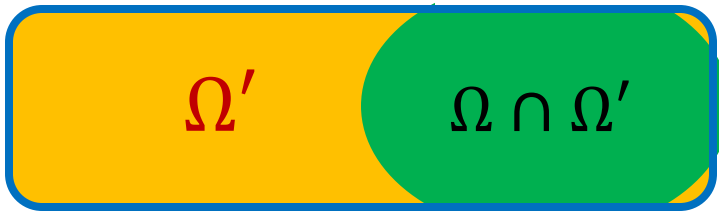


- W.h.p., if $X \in \Omega$ then $X \in \Omega'$. ➡ $\Omega \cap \Omega' \approx \Omega$.
- W.h.p., for most $X \in \Omega' \setminus \Omega$, it holds that $T < W(X) \le T + \frac{T}{\ell}$ ➡ $\Omega' \setminus \Omega$ is not too large

Since $W_i \in \left(\frac{T}{\ell}, \frac{2T}{\ell}\right]$, X becomes a solution by throwing an *arbitrary* item

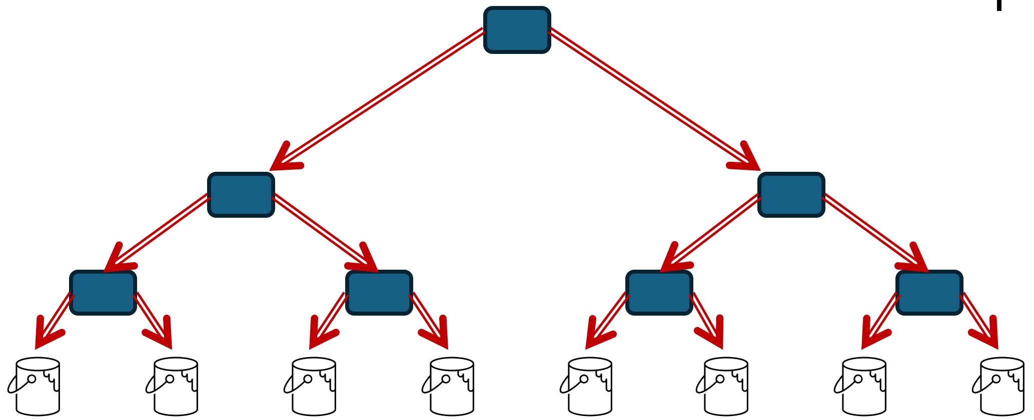
- Using the function f_{root} to approximate the size of Ω'

$$|\Omega'| \approx \sum_{jS_0 \le T'} f_{\text{root}}(jS_0)$$



- Draw uniform random samples $X \in \Omega'$ and test whether $X \in \Omega \cap \Omega'$ to estimate

$$\frac{|\Omega \cap \Omega'|}{|\Omega'|}$$



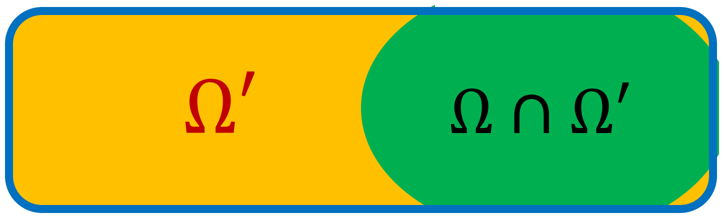
Sample Complexity

$$\#\text{samples} = O\left(\frac{|\Omega'|}{|\Omega \cap \Omega'|}\right)$$

- Draw samples using a ***top-down*** process
- Each sample takes time $\tilde{O}(\ell^{1.5})$

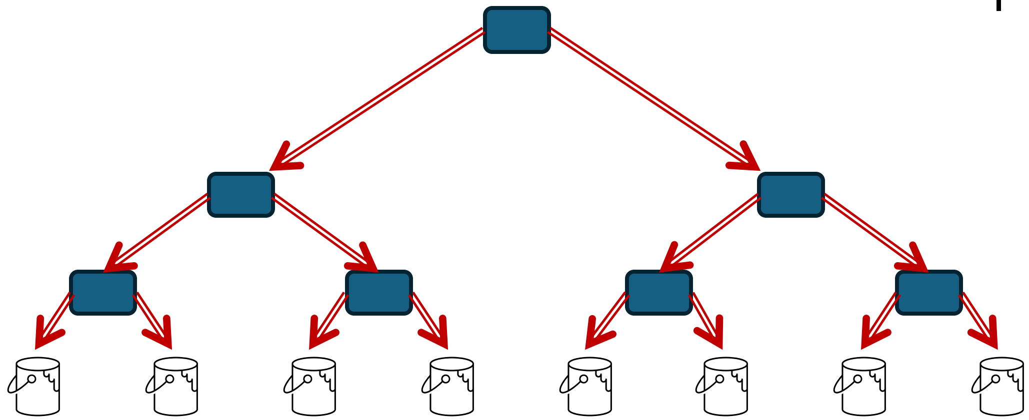
- Using the function f_{root} to approximate the size of Ω'

$$|\Omega'| \approx \sum_{jS_0 \le T'} f_{\mathrm{root}}(jS_0)$$



- Draw uniform random samples $X \in \Omega'$ and test whether $X \in \Omega \cap \Omega'$ to estimate

$$\frac{|\Omega \cap \Omega'|}{|\Omega'|}$$



Sample Complexity

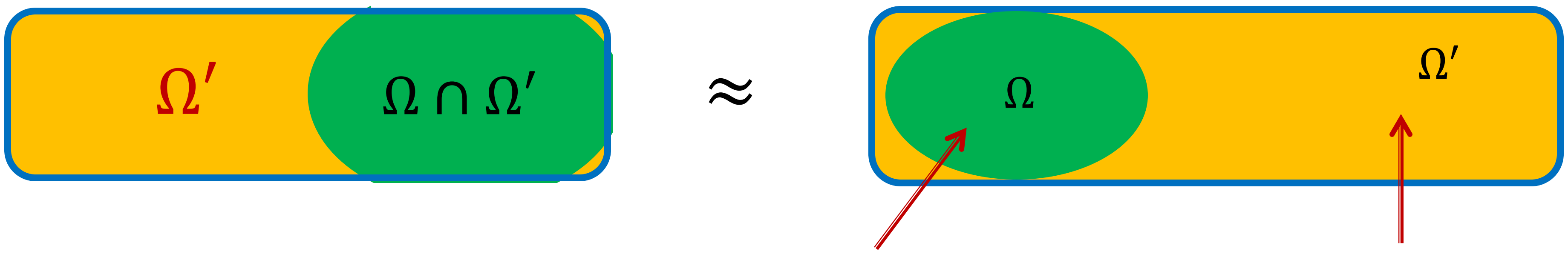
$$\#\text{samples} = O\left(\frac{|\Omega'|}{|\Omega \cap \Omega'|}\right) = \tilde{O}\left(\frac{n}{\ell}\right)$$

hitting set

- Draw samples using a ***top-down*** process
- Each sample takes time $\tilde{O}(\ell^{1.5})$

Complexity in sampling step: $\tilde{O}(n\sqrt{\ell}) = \tilde{O}(n^{1.5})$

Hitting set: Improved sample complexity

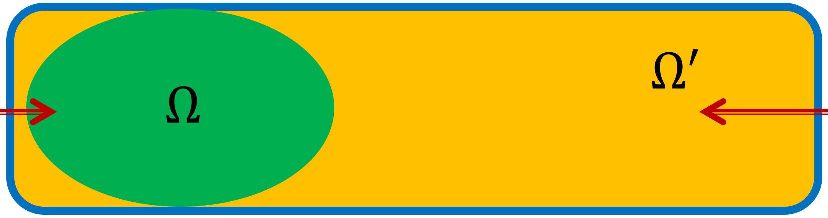


$W(X) = \sum_{i \in X} W_i \leq T$

$T < W(Y) = \sum_{i \in Y} W_i \leq T + \frac{T}{\ell}$

Hitting set: Improved sample complexity

$W(X) = \sum_{i \in X} W_i \leq T$



$T < W(Y) = \sum_{i \in Y} W_i \leq T + \frac{T}{\ell}$

- Construct a ***random*** $H \subseteq [n]$ by selecting each $i \in [n]$ with probability $\frac{1}{\ell} \cdot \mathrm{polylog}(n)$

  with prob. $\geq 99\%$, $\quad |H| \leq \tilde{O}(n/\ell)$

- For any $Y \in \Omega' \backslash \Omega$, we say Y is ***hit*** by H if $H \cap Y \neq \emptyset$.

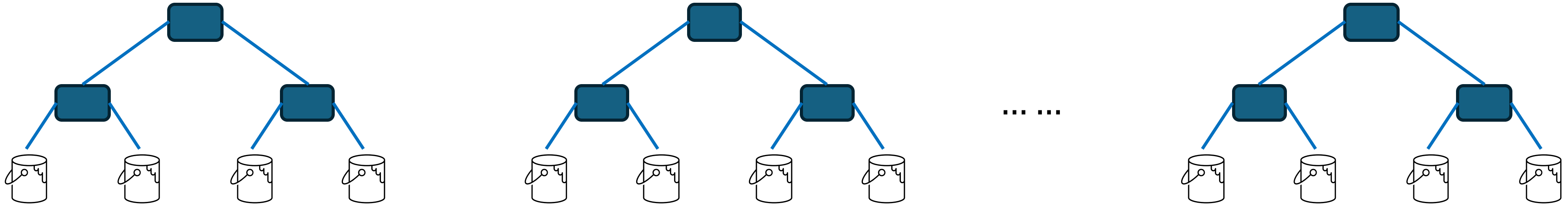
$|Y| > \Omega(\ell)$ ➡ w.h.p. Y is hit by H ➡ w.p. $\geq 99\%$, at least $1 - o(1)$ fraction in $\Omega' \backslash \Omega$ are hit by H

- By probabilistic method, $\exists H$ with size $\tilde{O}(n/\ell)$ that hits most $Y \in \Omega' \backslash \Omega$
- For any Y hit by H, throw away arbitrary $i \in H$ make $Y - i \in \Omega$

$$\frac{|\{Y \in \Omega \backslash \Omega' \mid Y \text{ hit by } H\}|}{|\Omega|} \leq |H| = \tilde{O}\left(\frac{n}{\ell}\right) \quad \Rightarrow \quad \frac{|\Omega'|}{|\Omega|} \leq (1 + o(1))|H| = \tilde{O}\left(\frac{n}{\ell}\right)$$

General case

- Partition items into ***groups*** according to the weight



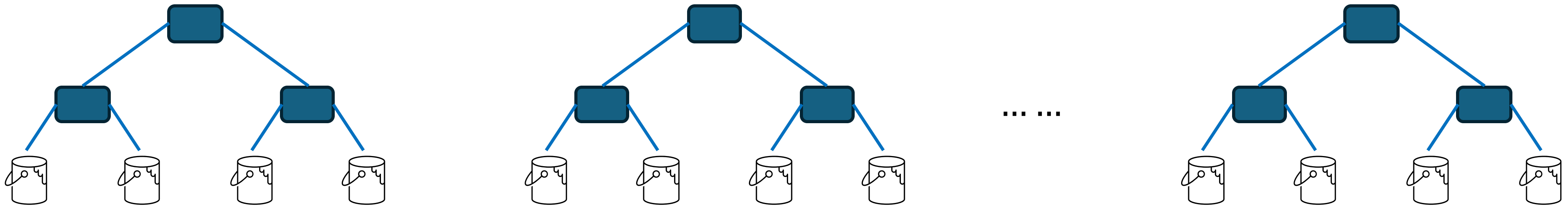
items with weight $T \sim \frac{T}{2}$ items with weight $\frac{T}{2} \sim \frac{T}{4}$ small items with weight $0 \sim \frac{T}{2n}$

- Merge all $O(\log n)$ groups together, which requires that their roots have the ***same scale*** S

  - *Compute* $\ell = O(n)$ *s.t. for all roots, set the same* ***scale*** $\widetilde{\Theta}(T/\ell)$

General case

- Partition items into ***groups*** according to the weight



items with weight $T \sim \frac{T}{2}$ | items with weight $\frac{T}{2} \sim \frac{T}{4}$ | small items with weight $0 \sim \frac{T}{2n}$

- ***Merge*** all $O(\log n)$ groups together, which requires that their roots have the ***same scale*** S
  - *most $X \subseteq [n]$ with $W_X = \Theta(T)$ contains $\widetilde{\Omega}(\ell)$ items with weight $\widetilde{\Theta}(T/\ell)$*
  - *Compute $\ell = O(n)$ s.t. for all roots, set the same **scale** $\widetilde{\Theta}(T/\ell)$*

sample complexity $\tilde{O}(n/\ell)$

- If there are many ***tiny items***, then a sample X satisfies $|X| = O(n)$. Running time $\tilde{O}\left(n \cdot \frac{n}{\ell}\right) = \tilde{O}\left(\frac{n^2}{\ell}\right)$ can be large. We reduce the running time to $\tilde{O}(n^{1.5})$ using
  1. Tiny items are typically ***rounded to weight 0***. We maintain tiny items implicitly. In other words, we draw ***partial samples*** containing only non-tiny items.
  2. Each tiny item with rounded weight 0 should be included with ½ ***probability***. (cannot implement)
  3. Construct ***another #Knapsack instance*** to take care the contribution of tiny items.

Open problems

- A near-linear time counting algorithm?
- FPTAS (***deterministic*** algorithm) with improved running time?
  - Current best algorithm $\tilde{O}(n^{2.5})$ [Gawrychowski, Markin, and Weimann 2018]
- Extensions and applications
  - Integer #Knapsack and multi-dimensional #Knapsack
  - Contingency table
  - Apply ideas to other approximate counting problems