Train Tracks with Gaps
Applying the Probabilistic Method to Trains

Seminar Algorithmentechnik · November 10, 2023
Robert Krause

Institute of Theoretical Informatics · Algorithmics Group

Tains, Tracks and Gaps

train, length $4f$
Tains, Tracks and Gaps

train, length 4f
Tains, Tracks and Gaps

Train, length $4f$

back

$\text{n wheels}$

$\text{n wheels}$

front
Tains, Tracks and Gaps

- Trains drive on uninterrupted tracks, very wasteful.
- Train, length \(4f\).
- Track length \(\ell\).
- \(n\) wheels at the back and \(n\) wheels at the front.

Diagram:
- Train on a track.
- Arrows indicating front and back.
- Track length labeled as \(\ell\).
- \(n\) wheels at both ends.
- Train length labeled as \(4f\).

Institute of Theoretical Informatics
Algorithmics Group

Robert Krause – Train Tracks with Gaps
trains drive on uninterrupted tracks
very wasteful
⇒ build as little track as necessary
How much track do we really need?

- front and back quarter have to always be supported

\[ O\left(\frac{\ell}{n}\right) \text{ track for equally spaced wheels} \]

\[ O\left(\ell \ln n\right) \text{ track for arbitrary wheel arrangements} \]
How much track do we really need?

- front and back quarter have to always be supported

\[ O\left(\frac{\ell}{n}\right) \text{ track for equally spaced wheels} \]
\[ \mathcal{O}(\ell \ln n) \text{ track for arbitrary wheel arrangements} \]
How much track do we really need?

- front and back quarter have to always be supported

\[ O\left(\frac{\ell}{n}\right) \text{ track for equally spaced wheels} \]
\[ O\left(\frac{\ell \ln n}{n}\right) \text{ track for arbitrary wheel arrangements} \]
How much track do we really need?

- front and back quarter have to always be supported
How much track do we really need?

- front and back quarter have to always be supported
- we will show:
How much track do we really need?

- Front and back quarter have to always be supported.
- We will show:
  - $O(\ell/n)$ track for equally spaced wheels.
  - $O(\ell \ln n / n)$ track for arbitrary wheel arrangements.
Probabilistic Method

- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater 0 then it must exist

Lemma: It is possible to flip a coin three times so that the number of tails is at least 2

Proof:
expected value is 1.5
outcome is integer
there exists an outcome ≥ 1.5 ⇒ there have to be at least 2
Probabilistic Method

- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater than 0 then it must exist

Lemma: It is possible to flip a coin three times so that the number of tails is at least 2

Proof:
expected value is 1.5
outcome is integer
there exists an outcome $\geq 1.5 \Rightarrow$ there have to be at least 2
Probabilistic Method

- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater 0 then it must exist

**Lemma**: It is possible to flip a coin three times so that the number of tails is at least 2

**Proof**:
- expected value is 1.5
- outcome is integer
- there exists an outcome $\geq 1.5 \Rightarrow$ there have to be at least 2
Evenly spaced wheels: Lower Bound

Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal
Evenly spaced wheels: Lower Bound

assume a portion < \( \frac{1}{n} \) of the track is built

probability \( w_i \) for wheel \( i \) to be supported by track is < \( \frac{1}{n} \)

Placing a track of length \( \frac{1}{4n} \) every quarter of a train length is optimal

⇒ there exists a position where the train falls through
Evenly spaced wheels: Lower Bound

- assume a portion $< \frac{1}{n}$ of the track is built
- probability $w_i$ for wheel $i$ to be supported by track is $< \frac{1}{n}$
- only looking at wheels in the back
- using union bound: $\mathbb{P}(\bigcup_{i=1}^{n} w_i) \leq \sum_{i=1}^{n} \mathbb{P}(w_i) < \sum_{i=1}^{n} \frac{1}{n} = 1$
Evenly spaced wheels: Lower Bound

Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal

- Assume a portion $< \frac{1}{n}$ of the track is built
- Probability $w_i$ for wheel $i$ to be supported by track is $< \frac{1}{n}$
- Only looking at wheels in the back
- Using union bound: $P(\bigcup_{i=1}^{n} w_i) \leq \sum_{i=1}^{n} P(w_i) < \sum_{i=1}^{n} \frac{1}{n} = 1$

$\implies$ There exists a position where the train falls though
The Setting formalised

- Length $f$
- $n$ wheels
- Integer steps $1, 2, \ldots$
- Track length $\ell$

Diagram:

- Train track
- Wheels
- Integer steps
- Track length
The Setting formalised

- $C = \text{set of wheel positions, from the rear quarter, of size } n$
- $T \subseteq \{1, 2, \ldots, \ell\} = \text{set of pillars}$
- $C + r = \{c + r | c \in C\}$
- $T$ is valid $\iff (C + k) \cap T \neq \emptyset$, for $k = 1, \ldots, \ell - f$
Arbitrary Wheel Arrangements

\[
\text{every pillar is built with probability } \frac{\ln n}{n} \quad \text{train falls with probability } (1 - \frac{\ln n}{n})^n \leq 1 - e^{\frac{\ln n}{n}} = \frac{1}{n}
\]

out of all the possible train placements only \(\frac{1}{n}\)-th is problematic

fix these instances by adding a pillar to support the train

\[
\ln \frac{n}{n} \cdot \ell + \frac{1}{n} \cdot \ell = \left(1 + \ln \frac{n}{n}\right) \cdot \ell
\]

expected tracks

\[O(\ell n)\] runtime
Arbitrary Wheel Arrangements

- every pillar is built with probability \( \frac{\ln n}{n} \)
Arbitrary Wheel Arrangements

- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probability $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
Arbitrary Wheel Arrangements

- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probability $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^\ln n} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train
Arbitrary Wheel Arrangements

- every pillar is built with probability \( \frac{\ln n}{n} \)
- train falls with probability \((1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}\)
- out of all the possible train placements only \(\frac{1}{n}\)-th is problematic
- fix these instances by adding a pillar to support the train
Arbitrary Wheel Arrangements

- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probability $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train
Arbitrary Wheel Arrangements

- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probability $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train
Arbitrary Wheel Arrangements

- every pillar is built with probability \( \frac{\ln n}{n} \)
- train falls with probability \( (1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n} \)
- out of all the possible train placements only \( \frac{1}{n} \)-th is problematic
- fix these instances by adding a pillar to support the train

\[
\frac{\ln n}{n} \cdot \ell + \frac{1}{n} \cdot \ell = (\frac{1+\ln n}{n}) \cdot \ell \text{ exp. tracks}
\]

\( \mathcal{O}(\ell n) \) runtime
Why is it enough to look at the rear quarter?

- keep the calculated tracks
- check front quarter for every position
- running time still $O(\ell n)$
Why is it enough to look at the rear quarter?

- Keep the calculated tracks
- Check front quarter for every position
- Running time still $O(\ell n)$
Why is it enough to look at the rear quarter?

- keep the calculated tracks
- check front quarter for every position
- running time still $O(\ell n)$
Why is it enough to look at the rear quarter?

- keep the calculated tracks
- check front quarter for every position
- running time still $O(\ell n)$
Why is it enough to look at the rear quarter?

- keep the calculated tracks
- check front quarter for every position
- running time still \( O(\ell n) \)
- train falls with probability \( (1 - \frac{1+\ln n}{n})^n \leq \frac{1}{e^{1+\ln n}} = \frac{1}{ne} \)
- build additional exp. \( \frac{\ell}{ne} \) tracks
what we have seen so far:

<table>
<thead>
<tr>
<th></th>
<th>lower-bound</th>
<th>upper-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>even</td>
<td>$\frac{\ell}{n}$</td>
<td>$\frac{\ell}{n}$</td>
</tr>
<tr>
<td>arbitrary</td>
<td>$\left(\frac{\ell \ln n}{n}\right)$</td>
<td>$\frac{\ell \ln n}{n}$</td>
</tr>
</tbody>
</table>
Further Algorithms

- what we have seen so far:

<table>
<thead>
<tr>
<th></th>
<th>lower-bound</th>
<th>upper-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>even</td>
<td>( \frac{\ell}{n} )</td>
<td>( \frac{\ell}{n} )</td>
</tr>
<tr>
<td>arbitrary</td>
<td>( \frac{\ell \ln n}{n} )</td>
<td>( \frac{\ell \ln n}{n} )</td>
</tr>
</tbody>
</table>

- Goals for algorithms:
  1. correctness: Find set \( T \) such that for each \( k \in \{0, 1, \ldots, \ell - f\} \), the set \((C + k) \cap T \neq \emptyset\)
  2. runtime: exp. \( O(n\ell) \)
  3. track length: \( |T| \in \text{exp. } O\left(\frac{\ell \ln n}{n}\right) \)
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
- replace random root-to-leaf walk with deterministic walk
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1

![Diagram of a binary tree with probabilities and conditional probabilities of failure.]
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic algorithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1
Deterministic Algorithm: Overview

Algorithm 1: deterministic algorithm

1. initially no pillars are built;
2. initialize objective function value;
3. for offset $k = 1, \ldots, \ell$ do
4. calculate $\Delta_0$ for not building pillar at $k$;
5. calculate $\Delta_1$ for not building pillar at $k$;
6. choose option minimizing obj. function;
Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{i \mid x_i = 1\} \cup \{j \mid X_j = 1\}$
- $T$ adds a pillar for each position $j$ where the train falls through the track
- $F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell) = |T|$ be the objective function
Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\Pr[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{ i \mid x_i = 1 \} \cup \{ j \mid X_j = 1 \}$
- $p_i = \text{probability that train falls through track at position } i$
Deterministic Algorithm

- Let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- Let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{i \mid x_i = 1\} \cup \{j \mid X_j = 1\}$
- $p_i$ = probability that train falls through track at position $i$
- $T$ adds a pillar for each position $j$ where the train falls through the track
Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{ i | x_i = 1 \} \cup \{ j | X_j = 1 \}$
- $p_i = \text{probability that train falls through track at position } i$
- $T$ adds a pillar for each position $j$ where the train falls through the track
- let $F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell) = |T|$ be the objective function
Deterministic Algorithm

\[ \mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{i \mid x_i = 1\} + \frac{\ln n}{n} (\ell - k) + \sum_i p_i \]

- Already built pillars
- exp. additional pillars in \( T_k \)
- exp. \#positions where train falls through

\[ \mathbb{E}[|T_k|] \]

\[ \mathbb{E}[:T \setminus T_k:] \]
Deterministic Algorithm

- \( \mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_{\ell})] = \{ i \mid x_i = 1 \} + \frac{\ln n}{n} (\ell - k) + \sum_i p_i \)
  - Already built pillars
  - exp. additional pillars in \( T_k \)
  - \( \mathbb{E}[|T_k|] \)
  - exp. #positions where train falls through
  - \( \mathbb{E}[T \setminus T_k] \)

- choosing 1: \( \mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}, p_i = 0 \) where \( k + 1 \in (C + i) \)
Deterministic Algorithm

- \( \mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{ i | x_i = 1 \} + \frac{\ln n}{n} (\ell - k) + \sum_i p_i \)

Already built pillars

\( \mathbb{E}[|T_k|] \)

exp. additional pillars in \( T_k \)

exp. #positions where train falls through

- choosing 1: \( \mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}, p_i = 0 \) where \( k + 1 \in (C + i) \)

- choosing 0: \( \mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n} \), update affected \( p_i \) with \( \frac{p_i}{1-(\ln n)/n} \)
Deterministic Algorithm

\[ \mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{i | x_i = 1\} + \frac{\ln n}{n} (\ell - k) + \sum_i p_i \]

choosing 1: \( \mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}, \ p_i = 0 \) where \( k + 1 \in (C + i) \)

choosing 0: \( \mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n}, \) update affected \( p_i \) with \( \frac{p_i}{1 - (\ln n)/n} \)

\[ \mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] \leq \frac{1 + \ln n}{n} \]

correctness: ✔

runtime: \( \ell \) iterations with \( \mathcal{O}(n) \) ✔

track length: arbitrary wheel arrangement proof showed

exp. #positions where train falls through

exp. additional pillars in \( T_k \)

Already built pillars

\[ \mathbb{E}[|T_k|] \]

\[ \mathbb{E}[|T \setminus T_k|] \]
Lovász Local Lemma (LLL)

- connected $X_i$ determine event outcome
- $d = \#\text{events connected via path of length 2}$
- each event depends on at most $d$ other events
Lovász Local Lemma (LLL)

- connected $X_i$ determine event outcome
- $d = \#\text{events connected via path of length 2}$
- each event depends on at most $d$ other events

**Lemma:** Given $p$ with $\mathbb{P}[E_i] \leq p$ and $pde \leq 1$, then $\mathbb{P}[\text{none of the events } E_i \text{ occur}] > 0$
Algorithmic Lovász Local Lemma

- given the LLL holds
- the **fix-it algorithm** resamples an event $E_i$ at most $\exp\left(\frac{1}{d}\right)$ times

---

**Algorithm 2: fix-it algorithm**

**Data:** independent random variables $X_1, \ldots, X_s$, events $E_1, \ldots, E_m$;

1. Independently sample each $X_1, \ldots, X_s$;
2. while $\exists E_i$ that holds do
   3. select $E_i$;
   4. resample all $X_j$, $E_i$ depends on;

$\Rightarrow \exp\left(\frac{m}{d}\right)$ iterations
Fix-it Algorithm for Train Tracks

- $\Pr[X_i = 1] = \frac{1+2\ln n}{n}$ decides whether to build pillar $i$
- $E_i = \text{event that train falls through tracks at offset } i$
- what is $d$?
Fix-it Algorithm for Train Tracks

- \( \mathbb{P}[X_i = 1] = \frac{1 + 2 \ln n}{n} \) decides whether to build pillar \( i \)
- \( E_i = \) event that train falls through tracks at offset \( i \)
- what is \( d \)?

![Diagram of train tracks and pillar placements](image)
Fix-it Algorithm for Train Tracks

- \( P[X_i = 1] = \frac{1+2\ln n}{n} \) decides whether to build pillar \( i \)
- \( E_i = \) event that train falls through tracks at offset \( i \)
- what is \( d \)?
Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1 + 2 \ln n}{n}$ decides whether to build pillar $i$
- $E_i = \text{event that train falls through tracks at offset } i$
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1 + 2 \ln n}{n})^n < \frac{1}{e^{1 + 2 \ln n}} \leq \frac{1}{en^2} = \rho$
- $pde = \frac{1}{en^2} n^2 e = 1$
Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides whether to build pillar $i$
- $E_i =$ event that train falls through tracks at offset $i$
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✅
Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides whether to build pillar $i$
- $E_i = \text{event that train falls through tracks at offset } i$
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✔
- runtime: Algorithmic LLL $\Rightarrow \frac{l}{n^2}$ iterations each naively $\mathcal{O}(n^3)$ ✔

\[ \text{track length: worst case each resample gives } n \text{ ones } \Rightarrow O(\ell n) \]

\[ \text{initial: } O(\ell + 2\ell \ln n) \]
Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides whether to build pillar $i$
- $E_i =$ event that train falls through tracks at offset $i$
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✓
- runtime: Algorithmic LLL $\Rightarrow \frac{\ell}{n^2}$ iterations each naively $\mathcal{O}(n^3)$ ✓
- track length:
  - worst case each resample gives $n$ ones $\Rightarrow \mathcal{O}\left(\frac{\ell}{n}\right)$
  - initial: $\mathcal{O}\left(\frac{\ell+2\ell\ln n}{n}\right)$ ✓
Min-Hash

- given collection of sets $S$
- technique for sampling one element for each set $S \in S$
- hash elements $h(s) \in (0, 1)$
- select element with minimum hash
Min-Hash

- given collection of sets $S$
- technique for sampling one element for each set $S \in S$
- hash elements $h(s) \in (0, 1)$
- select element with minimum hash

Two important Properties
- if sets $S_1$ and $S_2$ are similar, then their min-hash is likely to be the same
- if $s \in S$ is the minimum-hashed element in one set it is likely to be the minimum-hashed element in other sets
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- Assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar.
- Iterate over all possible positions.
- Choose the pillar with the smallest hash that is at a wheel position.
The Algorithm

- assign random real numbers \( r_1, \ldots, r_\ell \in (0, 1) \) to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
**The Algorithm**

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
- correctness: ✓
- runtime $\mathcal{O}(\ell n)$: ✓
The Algorithm

- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

**Correctness:**

**Runtime:** $O(\ell n)$

**Track length:** ?
Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset $k$
Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset $k$
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small $r_s$
Proof: Min-Hashing Algorithm

- \( S_k = (C + k) \) the wheel positions of the rear quarter at offset \( k \)
- selected pillar: \( \arg \min_{s \in S_k} r_s \)
- key observation: almost all sampled pillars have small \( r_s \)
- let \( r_s > \frac{\ln n}{n} \)
  \[ \Rightarrow \] all other \( s' \in S \) have larger values
- prob. of selecting such a pillar: \( (1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n} \)
- the expected number of pillars then is \( \frac{\ell - f}{n} \leq \frac{\ell}{n} \)
Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset $k$
- selected pillar: $\arg \min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small $r_s$
- let $r_s > \frac{\ln n}{n}$
  - $\Rightarrow$ all other $s' \in S$ have larger values
- prob. of selecting such a pillar: $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- the expected number of pillars then is $\frac{\ell - f}{n} \leq \frac{\ell}{n}$
- exp number of pillars with $r_s \leq \frac{\ln n}{n}$ is at most $\frac{\ell \ln n}{n}$
Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset $k$
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small $r_s$
- let $r_s > \frac{\ln n}{n}$
  $\Rightarrow$ all other $s' \in S$ have larger values
- prob. of selecting such a pillar: $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- the expected number of pillars then is $\frac{\ell - f}{n} \leq \frac{\ell}{n}$
- exp number of pillars with $r_s \leq \frac{\ln n}{n}$ is at most $\frac{\ell \ln n}{n}$
  $\Rightarrow$ overall $\leq \frac{\ell(1 + \ln n)}{n}$
What have we learned?

- Probabilistic Method and its applications for train tracks
What have we learned?

- Probabilistic Method and its applications for train tracks
- $O(\ell/n)$ track for equally spaced wheels
What have we learned?

- Probabilistic Method and its applications for train tracks
- $O(\ell/n)$ track for equally spaced wheels
- $O(\ell \ln n/n)$ track for arbitrary wheel arrangements
What have we learned?

- Probabilistic Method and its applications for train tracks
- $O(\ell/n)$ track for equally spaced wheels
- $O(\ell \ln n/n)$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities
What have we learned?

- Probabilistic Method and its applications for train tracks
- $O(\ell/n)$ track for equally spaced wheels
- $O(\ell \ln n / n)$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities
- Fix-it algorithm based on algorithmic Lovász Local Lemma
What have we learned?

- Probabilistic Method and its applications for train tracks
- $O(\ell/n)$ track for equally spaced wheels
- $O(\ell \ln n/n)$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities
- Fix-it algorithm based on algorithmic Lovász Local Lemma
- Min-Hash based algorithm
Appendix
Deterministic Algorithm: Correctness

- Why is the result not $F(0, \ldots, 0) \leq \frac{1 + \ln n}{n}$?
- Let $X$ be the set of pillars the event $E_i$ depends on.
- Let $0 = x_i \in X$, $i \neq k$.
- $\Delta_0 = -\frac{\ln n}{n} + \Delta \sum p_i \geq -\frac{\ln n}{n} + (1 - \frac{\ln n}{n}) > 0$
- $\Delta_1 = 1 - \frac{\ln n}{n} - (1 - \frac{\ln n}{n}) = 0$

Choosing 1: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}$ and zeroing out all $p_i$ where $x_k \in (C + i)$.

Choosing 0: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n}$ and updating affected $p_i$ with $\frac{p_i}{1 - (\ln n)/n}$. 