Randomized Algorithms - Package Presentation 2

Nitisha Bharathi, Sanjay Seetharaman

March 2020

Paper Details

Title	A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees
Authors	Karger, David R and Klein, Philip N and Tarjan, Robert E
Journal	Journal of the ACM (JACM)
Year	1995
Categories	Analysis of Algorithms and Problem Complexity, Discrete Mathematics, Probability and Statistics, Pattern Recognition
URL	`http://cs.brown.edu/research/pubs/pdfs/1995/Karger-1995-RLT.pdf`
Subject Descriptors	computations on discrete structures, graph algorithms, network problems, trees, probabilistic algorithms

Assumptions

- Input graph with no isolated vertices
- Edge weights are distinct

MST Properties

- **Cycle Property:** For any cycle $\mathcal{C}$ in a graph, the heaviest edge in $\mathcal{C}$ does not appear in the minimum spanning forest.
- **Cut Property:** For any proper nonempty subset $\mathcal{X}$ of the vertices, the lightest edge with exactly one endpoint in $\mathcal{X}$ belongs to the minimum spanning forest.

Borůvka's algorithm

1. For each vertex, select the minimum-weight edge incident to the vertex.
2. Contract all the selected edges, replacing by a single vertex each connected component defined by the selected edges and deleting all resulting isolated vertices, loops (edges both of whose endpoints are the same), and all but the lowest-weight edge among each set of multiple edges.

Randomization

How randomization can help us achieve a better time complexity?
Boruvka + Cycle Property + Randomization = Linear time with

very high probability

F-heavy Edge

Let $\mathcal{G}$ be a graph with weighted edges.
$\mathcal{W}$(x, y) : the weight of edge x, y

If $\mathcal{F}$ is a forest of a subgraph in $\mathcal{G}$,
$\mathcal{F}(x, y)$: The path connecting x and y in $\mathcal{F}$
$\mathcal{W}_{\mathcal{F}}$(x, y): The maximum weight of an edge on $\mathcal{F}(x, y)$, with the convention that $\mathcal{W}_{\mathcal{F}}(x, y) = \infty$ if x and y are not connected in $\mathcal{F}$

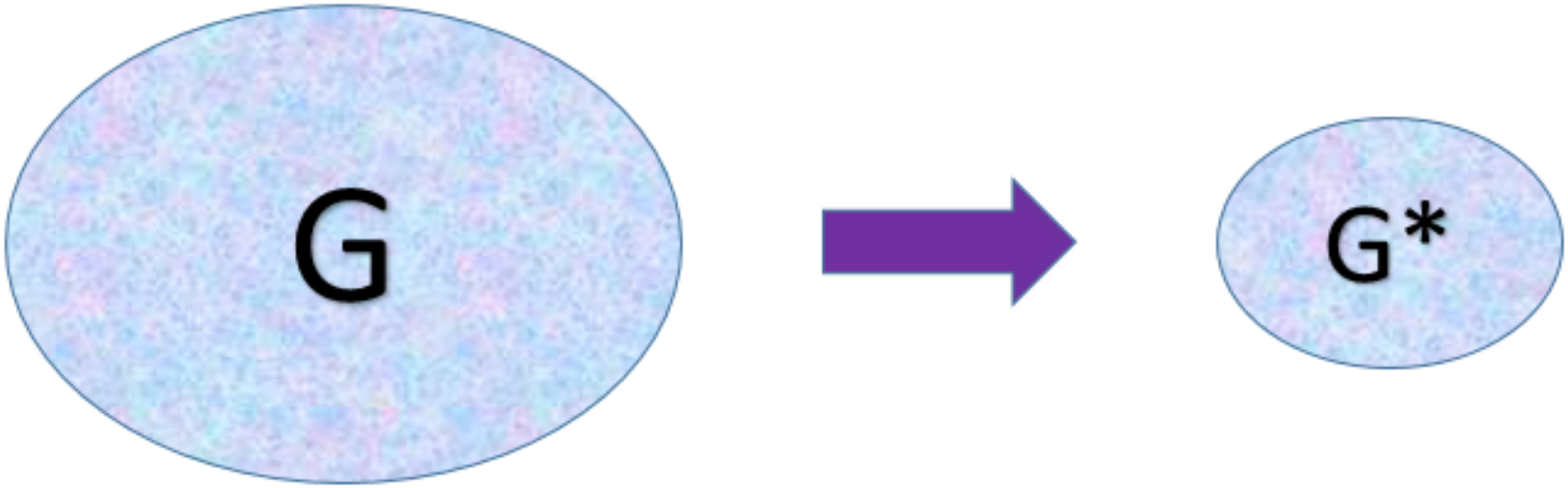
F-heavy Edge

An edge is $\mathcal{F}$- heavy if $\mathcal{W}$(x, y) $>$ $\mathcal{W}_{\mathcal{F}}$(x, y) otherwise, {x,y} is light

For any forest F, no F-heavy edge can be in the minimum spanning forest of G.

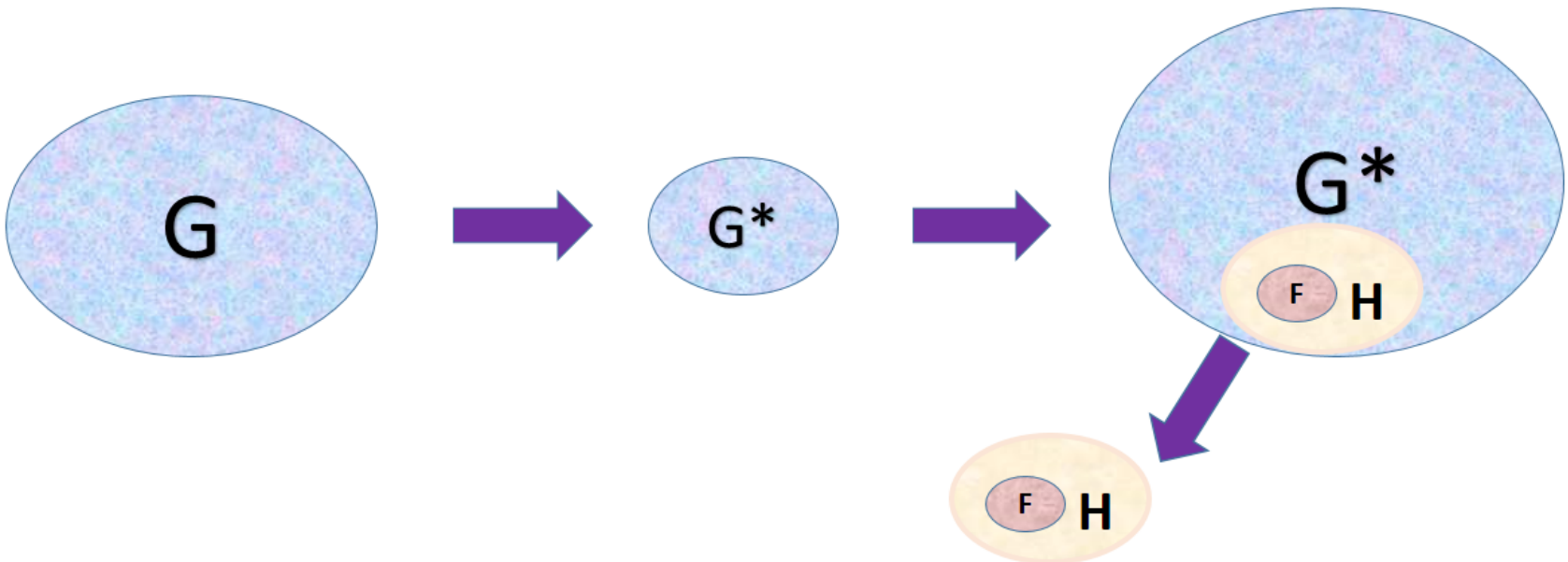
The Algorithm - Step 1

Apply two successive Borůvka steps to the graph, thereby reducing the number of vertices by at least a factor of four.



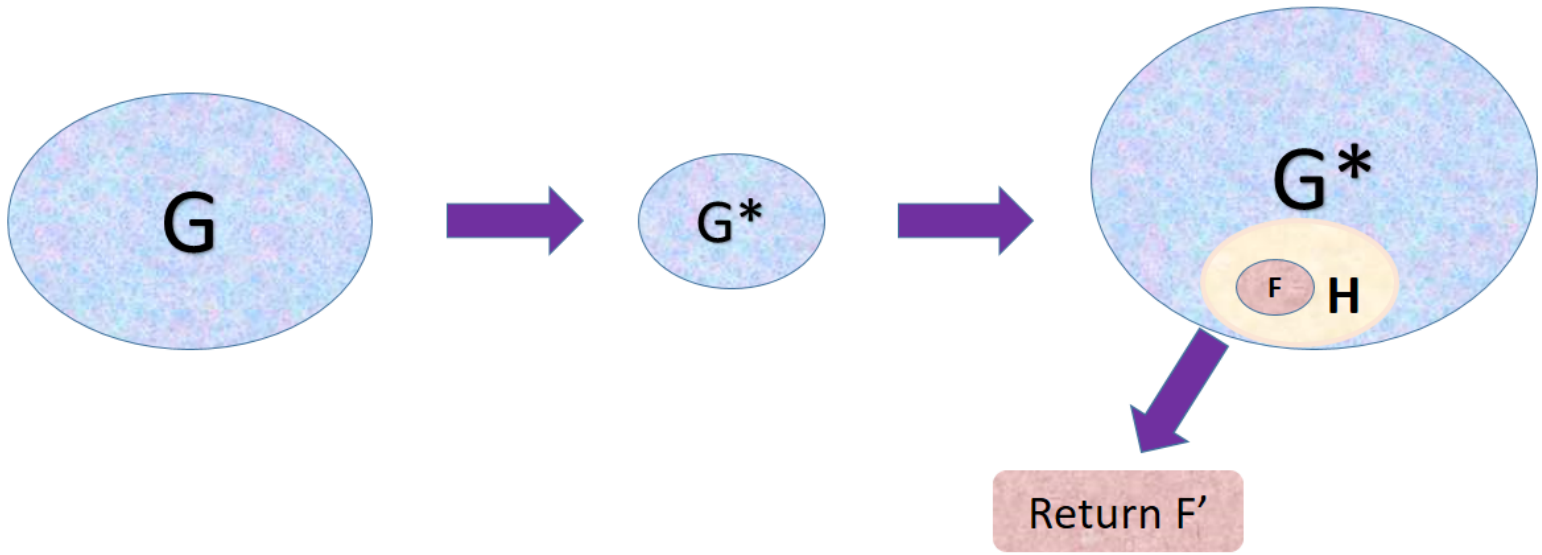
The Algorithm - Step 2

In the contracted graph, choose a subgraph $\mathcal{H}$ by selecting each edge independently with probability $1/2$. Apply the algorithm recursively to $\mathcal{H}$, producing a minimum spanning forest $\mathcal{F}$ of $\mathcal{H}$.Find all the $\mathcal{F}$-heavy edges (both those in $\mathcal{H}$ and those not in $\mathcal{H}$) and delete them from the contracted graph.



The Algorithm - Step 3

Apply the algorithm recursively to the remaining graph to compute a spanning forest $\mathcal{F}'$. Return those edges contracted in Step (1) together with the edges of $\mathcal{F}'$.



Analysis of the Algorithm

- Correctness
- Worst - case time complexity
- Expected time complexity

Correctness - Completeness

By the cut property, every edge contracted during Step 1 is in the minimum spanning forest. Hence the remaining edges of the minimum spanning forest of the original graph form a minimum spanning forest of the contracted graph.

Correctness - Soundness

By the cycle property, the edges deleted in step 2 do not belong to minimum spanning forest. By the inductive hypothesis, the minimum spanning forest of the remaining graph is correctly determined in recursive call of step 3.

Analysis of the Algorithm - Worst Case

- Total running time $=$ running time in each step
- Step 1 is two steps of Boruvka - $>$ linear
- Step 2 Dixon-Rauch-Tarjan verification algorithm -$>$ linear
- Total running time is bounded by a constant factor times the total number of edges

Analysis of the Algorithm - Worst Case

Let $\mathcal{G}$ be a graph with n vertices and m edges

$$m \geq \frac{n}{2}$$

Analysis of the Algorithm - Worst Case

Let $\mathcal{G}$ be a graph with n vertices and m edges

$$m \geq \frac{n}{2}$$

Each Problem generates atmost two subproblems

Analysis of the Algorithm - Worst Case

- At depth d there is atmost 2^d nodes.

Analysis of the Algorithm - Worst Case

- At depth d there is atmost 2^d nodes.
- Each node in depth d has atmost $\frac{n}{4^d}$ vertices

$$d \leq \log_4 n$$

Analysis of the Algorithm - Worst Case

- At depth d there is atmost 2^d nodes.
- Each node in depth d has atmost $\frac{n}{4^d}$ vertices

$$d \leq \log_4 n$$

- Total number of vertices

$$\sum_{d=0}^{\infty} 2^d \left(\frac{n}{4^d}\right) = \sum_{d=0}^{\infty} \frac{n}{2^d} = 2n$$

Analysis of the Algorithm - Worst Case

Theorem

The worst-case running time of the minimum-spanning-forest algorithm is $O(min\{n^2,\ m\ log\ n\})$, the same as the bound for Borůvka's algorithm.

Proof.

There are two estimate ways

1. Since vertex number for subproblem at depth d is atmost $\frac{n}{4^d}$, the edge is atmost $(\frac{n}{4^d})^2$. Overall edge number is also bound by

$$\sum_{d=0}^{\infty} 2^d \left(\frac{n}{4^d}\right)^2 = \frac{8}{7}n^2 = O(n^2)$$

□

Analysis of the Algorithm - Worst Case

2. Let $\mathcal{G}$ be a graph with n vertices and m edges
   - After 2 Borůvka steps, atmost $\frac{n}{4}$ vertices and $m - \frac{n}{2}$ edges remain for $\mathcal{G}^*$.
   - Since $\mathcal{F} = \text{MST}(\mathcal{H}), \mathcal{F}$ has at most $\mathcal{V}_{\mathcal{H}} - 1$ edges, and thus less than $\frac{n}{4}$.
   - Edges in left child $= \mathcal{E}_{\mathcal{H}}$
   - Edges in right child $\leq \mathcal{E}_{\mathcal{G}^*} - \mathcal{E}_{\mathcal{H}} + \mathcal{E}_{\mathcal{F}}$
   - Edges in two sub problems

$$\begin{aligned}
\mathcal{E}_{\mathcal{H}} + \mathcal{E}_{\mathcal{G}^*} &\leq \mathcal{E}_{\mathcal{H}} + \mathcal{E}_{\mathcal{G}^*} - \mathcal{E}_{\mathcal{H}} + \mathcal{E}_{\mathcal{F}} \\
&= \mathcal{E}_{\mathcal{G}^*} + \mathcal{E}_{\mathcal{F}} \\
&= \mathcal{E}_{\mathcal{G}} - \frac{\mathcal{V}_{\mathcal{G}}}{2} + \frac{\mathcal{V}_{\mathcal{G}}}{4} \leq \mathcal{E}_{\mathcal{G}} \\
&= m
\end{aligned}$$

Analysis of the Algorithm - Worst Case Analysis

The depth is at most $\log_4\ n$ and each level has at most m edges, so there are at most $m\ \log\ n$ edges.
Since, running time of the algorithm is proportional to edge number, the worstt-case running time of the minimum spanning forest algorithm is $O(min\{n^2,\ m\ log\ n\})$

Analysis of the Algorithm - Average Case

Theorem

The expected running time of the minimum spanning forest algorithm is $O(m)$.

Proof.

1. Calculating the expected number of edges for all left sub problems
   For $\mathcal{G}^*$ with k edges, after sampling with $\frac{1}{2}$ probability for each edge, $\mathrm{E}[\mathcal{E}_{\mathcal{H}}] = \frac{\mathrm{k}}{2}$. Since $\mathcal{G}^* \subseteq \mathcal{G}$, we have $\mathrm{E}[\mathcal{E}_{\mathcal{G}^*}] \leq E[\mathcal{E}_{\mathcal{G}}]$ and

$$E[\mathcal{E}_{\mathcal{H}}] = \frac{E[\mathcal{E}_{\mathcal{G}^*}]}{2} \leq \frac{E[\mathcal{E}_{\mathcal{G}}]}{2}$$

   Along the left path with starting $E[\mathcal{E}_{\mathcal{G}}] = \mathrm{k}$, the expected value of total edges is

$$E\left[\sum_{d=0}^{\infty} \mathcal{E}_{\mathcal{G}(d)}\right] \leq \sum_{d=0}^{\infty} \frac{\mathrm{k}}{2^d} = 2\mathrm{k}$$

Analysis of the Algorithm - Average Case

Given $\mathcal{V}_G^* = \mathrm{n}, \mathcal{F} = MST(\mathcal{H})$ where $\mathcal{H} \subseteq \mathcal{G}^*$

1. For each$\mathcal{F}-$light edge, there is $\frac{1}{2}$ probability of being sampled into $\mathcal{H}$.
2. Since each $\mathcal{F}-$light edge in $\mathcal{H}$ is $\mathcal{F}$ and $\mathcal{F}$ includes no edges not in $\mathcal{H}$, the chance that an $\mathcal{F}-$ light is in $\mathcal{F}$ is also $\frac{1}{2}$.
3. For edge e with weight heavier than the lightest of $\mathcal{F}$ is never $\mathcal{F}-$ light since there would be cycle with e as heaviest edge.
4. Thus the heaviest $\mathcal{F}-$light edge is always in $\mathcal{F}$. Given $\mathcal{E}_{\mathcal{F}} = \mathrm{k}, \mathcal{E}_{\mathcal{G}'}$ is the number of trials before k successes (selected into $\mathcal{H}$, and it forms a negative binomial distribution.

Analysis of the Algorithm - Average Case

2. Calculating the expected number of edges for all right sub problems
   Given $\mathcal{V}_G^* = \mathsf{n}, \mathcal{F} = MST(\mathcal{H})$ where $\mathcal{H} \subseteq \mathcal{G}^*$
   For $\mathcal{E}_{\mathcal{F}} = \mathsf{k}, \mathcal{E}_{\mathcal{G}'}$ is of negative binomial distribution with parameter $\frac{1}{2}$ and k. Thus,

$$E[\mathcal{E}_{\mathcal{G}'}] = \frac{\mathsf{k}}{1/2} = 2\mathsf{k}$$

Analysis of the Algorithm - Average Case

Summing all cases, we get

$$
\begin{aligned}
E[\mathcal{E}_{\mathcal{G}'}] &= \sum_{k=0}^{n-1} P(\mathcal{E}_{\mathcal{F}} = \mathrm{k}) E[\mathcal{E}_{\mathcal{G}'} | \mathcal{E}_{\mathcal{F}} = \mathrm{k}] \\
&= \sum_{k=0}^{n-1} P(\mathcal{E}_{\mathcal{F}} = \mathrm{k}).2\mathrm{k} \\
&< \sum_{k=0}^{n-1} P(\mathcal{E}_{\mathcal{F}} = \mathrm{k}).2\mathrm{n} \\
&= 2\mathrm{n}
\end{aligned}
$$

Analysis of the Algorithm - Average Case

For all right sub problems, expected number of edges is at most,

$$\sum_{d=1}^{\infty} \frac{2n}{4^d} . 2^{d-1} = n$$

Analysis of the Algorithm - Average Case

For each left path, the expected number total number of edges is twice of the leading subproblem, which is root or right child. So the overall expected value is at most $2(m + n)$
Since running time is proportional to the overall edge number, the expected value is $O(m) = O(m + n)$

Analysis of the Algorithm - Probability of Linearity

Theorem

The probability that time complexity is $O(m)$ is $1 - \exp(\Omega(m))$

Analysis of the Algorithm - Probability of Linearity

Chernoff Bound:
Given x_i is an i.d.d. random variable and $0 < i \leq n$, and X is the sum of all x_i, for $t > 0$, we have

$$Pr[X \geq A] \leq e^{-At} \prod_{i=1}^{n} E\left[e^{tx_i}\right]$$

Thus, the probability that less than s successes (each with chance p) within k trail is

$$Pr[X \geq s] \leq e^{-st} \prod_{i=1}^{k} E\left[e^{tx_i}\right]$$

$$= e^{-st}.(pe^t)^k$$

$$= e^{-\sigma(s)}, \text{for } t = \frac{1}{2} \text{and } p = \frac{1}{2}$$

Analysis of the Algorithm - Probability of Linearity

Given a path with leading problem $\mathcal{G}, \mathcal{E}_{\mathcal{G}} = \mathrm{k}$

- For each edge in $\mathcal{G}$, it has $\frac{1}{2}$ less chance to be kept in next sub problem and each edge contributes 1 to the total edge number. The path ends when the k-th edge move occurs.
- The probabilty that there are 3k total edges is probability there are k less edge-remove in k+3k trail. According to Chernoff Bound, the probabilty is $\exp(-\Omega(\mathrm{k}))$

Analysis of the Algorithm - Probability of Linearity

1. Given $\mathcal{V}_{\mathcal{G}^*} = \mathrm{n}^{'}$. For each edge in $\mathcal{G}^{'}$, it has $\frac{1}{2}$ chance to be in $\mathcal{F}$. Since $\mathcal{E}_{\mathcal{F}} = \mathrm{n}\prime - 1$, the probabilty that $\mathcal{E}_{\mathcal{G}'} > 3\mathrm{n}^{'}$ is probability there are $\mathrm{n}^{'} - 1$ less $\mathcal{F}$ edge in 3k trail. According to Chernoff bound, the probability is $\exp(-\Omega(n^{'}))$
2. There is at most $\frac{\mathrm{n}}{2}$ total vertices in all $\mathcal{G}^{'}$. If we take all the trail as a whole, the probability that there are more than $\frac{3\mathrm{n}}{2}$ edges in all right sub problem is $\exp(-\Omega(n))$

Analysis of the Algorithm - Probability of Linearity

Combined with previous two analysis, there is at least probability as below that total edges never exceeds $3(m + \frac{3n}{2})$, where λ it the set of all right problems:

$$(1 - e^{-\Omega(n)}).(1 - e^{-\Omega(m)}). \prod_{g \in \lambda} (1 - e^{-\Omega(\mathcal{V}_g)}) \approx 1 - e^{-\Omega(m)}$$

Thus, the probability that time complexity is $O(m)$ is $1 - \exp(\Omega(m))$

References

DR Karger.
Random sampling in matroids, with applications to graph connectivity and minimum spanning treesrandom sampling in matroids, with applications to graph connectivity and minimum spanning tree.
In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 84–93. IEEE, 1993.

Otakar Borůvka.
O jistém problému minimálním.
Práce Mor. Prırodved. Spol. v Brne (Acta Societ. Scienc. Natur. Moravicae), 3(3):37–58, 1926.