Minimim Spanning Trees

Overview

Consider a undirected graph $G = (V, E)$, with edge weights $w : E \to \mathbb{R}_+$.

An MST is a set of edges $T \subseteq E$ such that (V, T) is a connected graph and $\sum_{e \in T} w(e)$ is minimal.

- Some deterministic algorithms
- Sampling for edge reduction
- Boruvka's node reduction algorithm
- A randomized linear time algorithm

1 Some Deterministic Algorithms

1.1 The Cut Property

Lemma 1. *Consider $S \subseteq V$ and $e \in \{(s,t) : (s,t) \in E, s \in S, t \in V \setminus S\}$ with minimal weight.*

$\exists$ MST T of G that contains e.

1.2 A Generic Algorithm Using the Cut Propery

$T := \emptyset$ -- MST edges found so far
while $|T| < n - 1$ **do** -- no spanning tree yet
  pick an edge $e \in E \setminus T$ that fulfills the cut property
  $T := T \cup \{e\}$ -- Grow tree
return T

1.3 The Jarník-Prim Algorithm

$T := \emptyset$
$S := \{s_0\}$ for arbitrary start node s
repeat $n - 1$ times
  find (s,t) fulfilling the cut property for S
  $S := S \cup \{t\}$
  $T := T \cup \{(s,t)\}$

Time $\mathcal{O}(m + n \log n)$ using Fibonacci heap storing $V \setminus S$

1.4 Kruskal's Algorithm

$T := \emptyset$ −− subforest of the MST
foreach $(u, v) \in E$ in ascending order of weight **do**
  if u and v are in different subtrees of T **then**
    $T := T \cup \{(u, v)\}$ −− Join two subtrees
return T

Time $\mathcal{O}(m \log m)$. Faster for integer weights.

2 Edge Reduction Using Sampling

2.1 The Cycle Property

Lemma 2.
Consider any cycle $C \subseteq E$,
and an edge $e \in C$ with maximal weight.
MST$(E \setminus \{e\})$ is an MST of G.

2.2 The Algorithm

$R :=$ random sample of r edges from E
$F :=$MST(R) −− Wlog assume that F spans V
$L := \emptyset$ −− "light edges" with respect to R
foreach $e \in E$ **do** −− Filter
  $C :=$ the unique cycle in $\{e\} \cup F$
  if e is not heaviest in C **then**
    $L := L \cup \{e\}$
return MST$((L))$

2.3 Analysis

T. Chan, Backward Analysis of the Karger-Klein-Tarjan algorithm for minimum spanning trees, Information Processing Letters 67, 303–304, 1998.

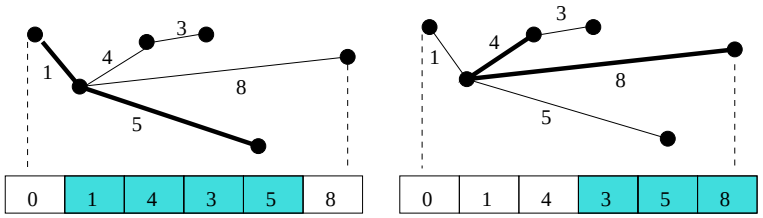
Observation: $e \in L$ only if $e \in \mathrm{MST}(R \cup \{e\})$.
(Otherwise e could replace some heavier edge in F).

Lemma 3. $\mathbb{E}[|E''|] \leq \frac{mn}{r}$

Proof. Consider a random edge $e \in E$ choosen independently of R.
It suffices to show that $\mathbb{P}[e \in \mathrm{MST}(R \cup \{e\})] < n/r$.
Let $R' = R \cup \{e\}$.
Now we delete e from R'
e is equally likeley to be any element of R'
MST(R') has $n - 1$ edges, i.e.,
$\mathbb{P}[e \in \mathrm{MST}(R \cup \{e\})] \leq (n-1)/r < n/r$. □

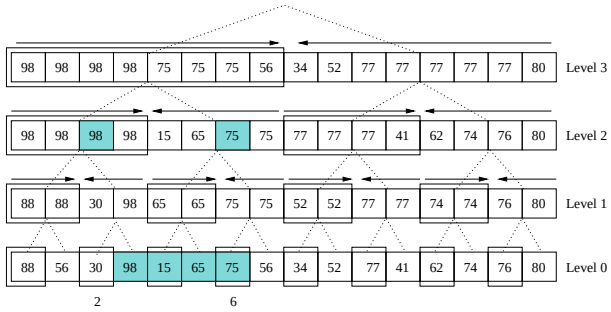
2.4 MST Verification by Interval Maxima

- Number the nodes by the order they were added to the MST by Prim's algorithm.
- $w_i =$ weight of the edge that inserted node i.
- Largest weight on path$(u,v) = \max\{w_j \| u < j \leq v\}$.



2.5 Interval Maxima

Preprocessing: build $n \log n$ size array PreSuf.



To find $\max a[i..j]$:

- Find the level of the LCA: $\ell = \lfloor \log_2(i \oplus j) \rfloor$.
- Return $\max(\mathrm{PreSuf}[\ell][i], \mathrm{PreSuf}[\ell][j])$.
- Example: $2 \oplus 6 = 010 \oplus 110 = 100 \Rightarrow \ell = 2$

2.6 A Simple Filter Based Algorithm

Choose $r = \sqrt{mn}$.

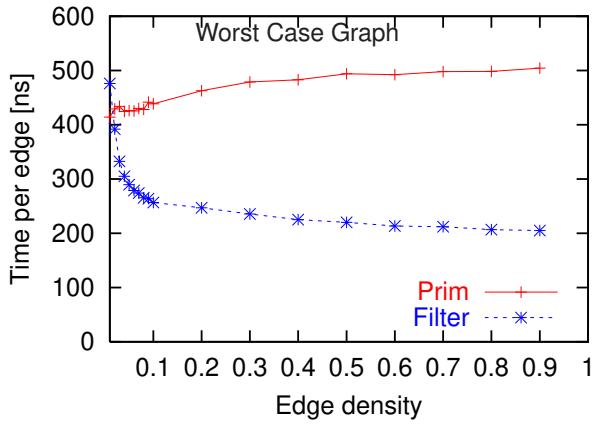
We get expected time

$$T_{\mathrm{Prim}}(sqrtmn) + \mathcal{O}(n \log n + m) + T_{\mathrm{Prim}}(\tfrac{mn}{\sqrt{mn}}) = \mathcal{O}(n \log n + m)$$

The constant factor in front of the m is very small.

2.7 Results

10 000 nodes, SUN-Fire-15000, 900 MHz UltraSPARC-III+



Boruvka's Node Reduction Algorithm

For each edge find the lightest incident edge.

Include them into the MST (cut property)

contract these edges,

Time $\mathcal{O}(m)$

At least halves the number of remaining nodes

Randomized Linear Time Algorithm

1. Apply 3 Boruvka steps (node reduction). $\mathcal{O}(m+n)$.
2. $R \Leftarrow m/2$ random edges. $\mathcal{O}(m+n)$.
3. $F \Leftarrow MST(R)$ [Recursively].
4. Find light edges L (edge reduction). $\mathcal{O}(m+n)$
   $\mathbb{E}[|L|] \leq \frac{mn/8}{m/2} = n/4$.
5. $T \Leftarrow MST(L)$ [Recursively].

$T(n,m) \leq T(n/8, m/2) + T(n/8, n/4) + c(n+m)$
$T(n,m) \leq 2c(n+m)$ fulfills this recurrence.