Minimum Spanning Tree (MST)

Hengfeng Wei

hfwei@nju.edu.cn

May 26, 2017 – May 31, 2017

Minimum Spanning Tree (MST)

A generic MST algorithm

Cut property (strong)

Cut property (strong)

- Graph $G = (V, E)$
- X is some part of an MST T of G
- Any cut $(S, V \setminus S)$ *s.t.* X does not cross $(S, V \setminus S)$
- Let e be a lightest edge across $(S, V \setminus S)$

Then $X \cup \{e\}$ is some part of an MST T' of G.

Proof.

Exchange argument

□

Cut property (strong)

Correctness of Prim's and Kruskal's algorithms.

Cut property (weak)

Cut Property [Problem: 3.6.18 (a)]

- Graph $G = (V, E)$
- Any cut $(S, V \setminus S)$ where $S, V - S \neq \emptyset$
- Let $e = (u, v)$ be *a* minimum-weight edge across $(S, V \setminus S)$

Then e must be in *some* MST of G.

"a" $\rightarrow$ "the" $\implies$ "some" $\rightarrow$ "any"

Applications of cut property

Application of cut property (Problem 6.10)

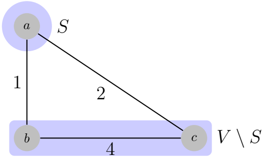
(3) $e \in G$ is a lightest edge $\implies e \in \exists$ MST of G

(4) $e \in G$ is the unique lightest edge $\implies e \in \forall$ MST of G

Applications of cut property

Wrong divide&conquer algorithm for MST (Problem 6.14)

- $G = (V, E, w)$
- $(V_1, V_2) : ||V_1| - |V_2|| \le 1$
- $T_1 + T_2 + \{e\}$: e is a lightest edge across (V_1, V_2)



Cycle property (weak)

Cycle property (Problem 6.13–2)

- $G = (V, E, w)$
- Let C be any cycle in G
- $e = (u, v)$ is *a* maximum-weight edge in C

Then $\exists$ MST T of $G : e \notin T$.

"a" $\rightarrow$ "the" $\implies$ "some" $\rightarrow$ "any"

Applications of cycle property

Anti-Kruskal algorithm (Problem 6.13–3)

Reverse-delete algorithm (wiki)

$$O(m \log n (\log \log n)^3)$$

Proof.

Invariant: If F is the set of edges remained at the end of the while loop, then there is some MST that are a subset of F. □

Reference

- "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem" by Kruskal, 1956.

Application of cycle property

(Problem 6.13–1)

(1) $e \notin$ any cycle of $G \implies e \in \forall$ MST

By contradiction.

Application of cycle property

(Problem 6.10)

(1) $|E| > |V| - 1$, e is the unique max-weight edge of $G \implies e \notin \forall$ MST

(2) $\exists C \subseteq G$, e is the unique max-weight edge of $C \implies e \notin \forall$ MST

(5) Cycle $C \subseteq G$, $e \in C$ is the unique lightest edge of $G \implies e \in \forall$ MST

Unique MST

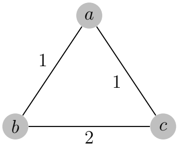
Unique MST (Problem 6.12–1)

Distinct weights $\implies$ Unique MST.

Unique MST

Unique MST (Problem 6.12–2)

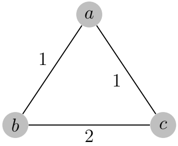
Unique MST $\not\Longrightarrow$ Equal weights.



Unique MST

Unique MST (Problem 6.12–3)

Unique MST $\not\Longrightarrow$ Minimum-weight edge across any cut is unique.



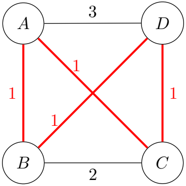
Theorem

Minimum-weight edge across any cut is unique $\Longrightarrow$ *Unique MST.*

Unique MST

Unique MST (Problem 6.12–3)

Unique MST $\not\Longrightarrow$ Maximum-weight edge in any cycle is unique.



Theorem (Conjecture)

Maximum-weight edge in any cycle is unique $\Longrightarrow$ *Unique MST.*

Unique MST

Unique MST (Problem 6.12–4)

Decide whether a graph has a unique MST?

1. Exchange argument based on cut property and cycle property
2. Ties breaking in Prim's and Kruskal's algorithms

Minimum Spanning Tree (MST)

Prim *vs.* Kruskal (Problem 6.4)

Prim *vs.* Kruskal (Problem 6.4)

- Array *vs.* heap
- $m = O(n)$ *vs.* $m = \Omega(n^2)$

$$T(n, m) = O(nT(\text{getMin}) + nT(\text{deleteMin}) + mT(\text{decreaseKey}))$$

MST on special graphs (Problem 6.3)

MST on special graphs (Problem 6.3)

1. K-bounded degree graph
2. Planar graph

$$(1)\ m \leq \frac{nk}{2}$$

$$(2)\ m \leq 3n - 6$$

Reference

- "Finding Minimum Spanning Trees" by David Cheriton and Robert Tarjan, 1976 (linear on planar graph).

Prim on special graphs (Problem 6.1)

Prim on special graphs (Problem 6.1)

$$E = \{v_1 v_i \mid i = 2 \ldots n\}, W(v_1 v_i) = 1$$

Prim on special graphs (Problem 6.2)

Prim on special graphs (Problem 6.2)

1. $G = K_n$
2. $W(v_i v_j) = n + 1 - i, 1 \leq i < j \leq n$

Feedback edge set

Feedback edge set (Problem 6.5)

1. Max-ST
2. (minimum) feedback edge set F:

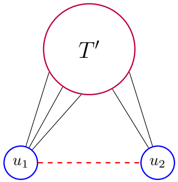
$$G' \triangleq G \setminus F \text{ is acyclic.}$$

MST with specific leaves

MST with specific leaves (Problem 6.8)

- $G = (V, E), U \subset V$
- finding an MST with U as leaves

- MST T' of $G' \triangleq G \setminus U$
- attach $\forall u \in U$ to T' (lightest edge)



MST with specific edges

MST with specific edges (Problem 6.9)

- $G = (V, E), S \subset E$ (no cycle in S)
- finding an MST with E as edges

1. Run Kruskal
2. Computing MST on graph of CCs

Edge weights

Edge weights (Problem 6.11)

$$w(e) > 0, w'(e) = (w(e))^2$$

Edge weights (Problem 6.10–7)

$$\exists e : w(e) < 0$$

Linear MST algorithms on special graphs

Linear MST algorithms on special graphs (Problem 5.25)

1. $\forall e \in E : w(e) = 1$
2. $m = n + 10$
3. $\forall e \in E : w(e) = 1 \vee w(e) = 2$

1. BFS
2. Cycle-breaking $\times 11$
3. BFS $\times 2$ ($\equiv$ Kruskal)

Updating MST

Updating MST (Problem 6.7)

Updating MST

Updating MST (Problem 6.7)

Second-best MST

Second-best MST (Problem 6.15)

Minimum Spanning Tree (MST)

Sharing edges

Sharing edges (Problem 6.6)

- $G = (V, E), w(e) > 0$
- Given s: all sssp trees from s must share some edge with some MST of G

a lightest edge leaving s

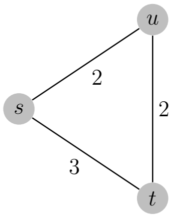
SP *vs.* MST

SP *vs.* MST (Problem 6.10–6)

✗ The shortest path between two nodes is necessarily part of some MST.

SPT *vs.* MST (Problem 6.17)

✗ The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.



Edge weights

Edge weights (Problem 6.19)

MST *vs.* SPT (from s):

$$w(e) \geq 0, w'(e) = w(e) + 1$$


THANK YOU