k 最短路の探索

本田雄一郎
大阪大学情報科学研究科
情報数理学専攻 森田研究室 M1

2017/01/30

紹介する論文

D.Eppstein: “Finding the k shortest paths” SIAM Journal on Computing 28(2) (1997)

問題設定

問題

Input:

- $G = (E, V)$: 有向グラフ
- $c : E \to R_+$: 正の辺重み関数
- $s, t \in V$: 始点，終点

Output:

- $p_1, p_2, ..., p_k$: 1〜k 最短路

主結果

主結果

グラフ G における s から t への $1 \sim k$ 最短路を $O(m + n \log n + k)$ で列挙できる．

手順

大まかには以下の流れで k 最短路を求めるためのグラフ $P(G)$ を作成する．

1. 最短路木 T の作成．$(O(m + n \log n))$
2. 2 つの 2 分ヒープの族 $H_{out}(v)$，$H_T(v)$ $(v \in V)$ を作成．$(O(m)$，$O(n \log n))$
3. $H_{out}(v)$，$H_T(v)$ をマージして 3 分ヒープからなる非巡回グラフ $D(G)$ を作成．$(O(n \log n))$
4. $D(G)$ に辺と頂点を追加し，4 分ヒープからなるグラフ $P(G)$ を作成．$(O(m + n \log n))$

用語整理

- $d(u,v)$: 頂点 u から頂点 v への最短路の距離
- $l(e)$: 辺 e の長さ
- $tail(e)$，$head(e)$: 有向辺 e の始点，終点
- $\delta(e) := l(e) + d(head(e),t) - d(tail(e),t)$: e を通ったことによる最短路との差

有向グラフ G と最短路木 T

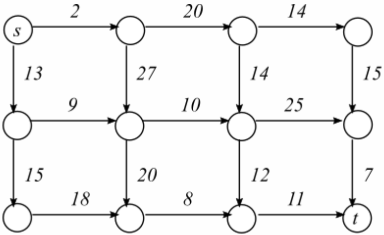


Figure: G

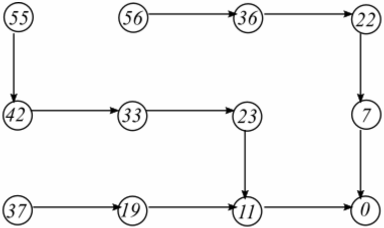


Figure: 最短路木 T

- G 内で，t から逆向きに Dijkstra 法を適用することで T を得る．$(O(m+n\log n))$
- T の各頂点の重みは t からの距離．
- T 内で頂点 v の次に来る頂点を $next_T(v)$ と表す．

$G - T$ と *sidetracks*

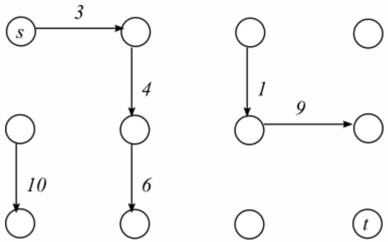


Figure: $G - T$

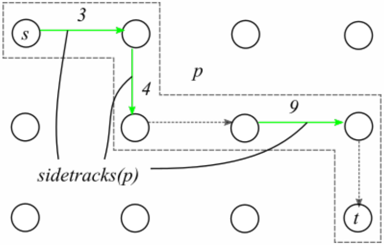


Figure: $sidetracks(p)$

G 内の $s - t$ 道 p が与えられたとき，辺集合 $sidetracks(p)$ を次のように定める．

$$sidetracks(p) = \{e| \ e \in G - T, e \in p\}$$

$G - T$ 内の $sidetracks(p)$ は G 内の $s - t$ 道 p と 1 対 1 に対応する．

2 つのヒープの作成 (1)

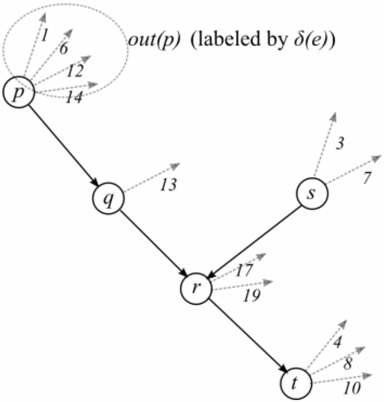


Figure: 最短路木 T の部分グラフ

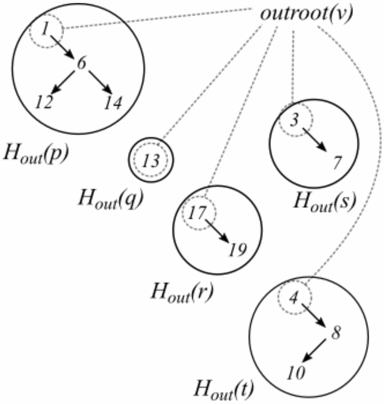


Figure: $H_{out}(v)$

- 各頂点 v について，$(v, w) \in G - T$ となる最小の辺 (以後，$outroot(v)$ と呼ぶ) を取り出し，残りの辺でヒープを作る．作成したヒープの根を $outroot(v)$ の右の子とする．($O(m)$)

2 つのヒープの作成 (2)

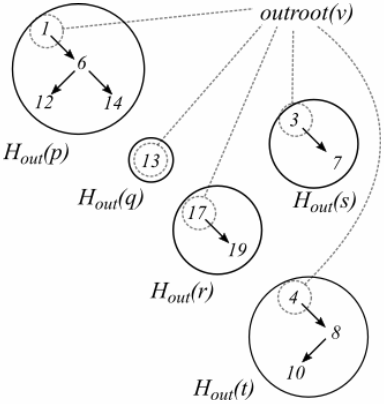


Figure: $H_{out}(v)$

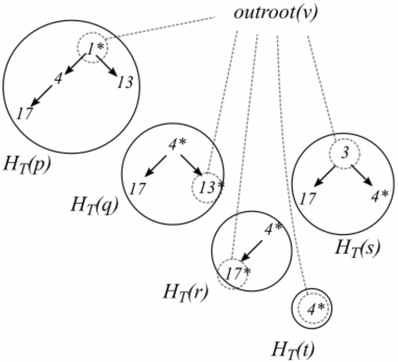


Figure: $H_T(v)$

- $H_T(t) := outroot(t)$．$H_T(next_T(v))$ に $outroot(v)$ を挿入し $H_T(v)$ を構成．($O(n \log n)$)
- $H_{out}(v)$，$H_T(v)$ の頂点はグラフ G の辺と 1 対 1 に対応することに注意．

$D(G)$ から $P(G)$ の構成へ

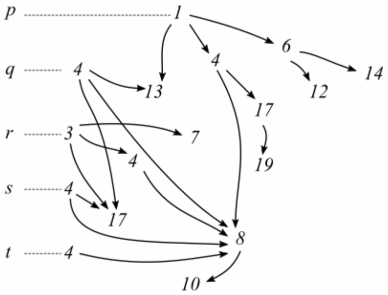


Figure: 有向非巡回グラフ $D(G)$

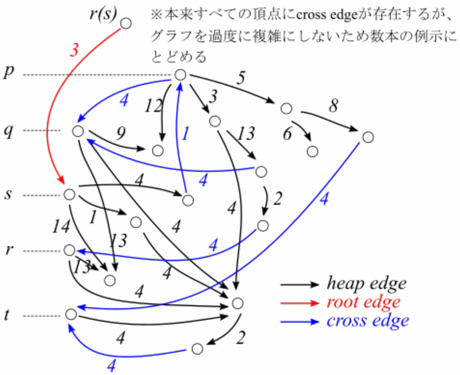


Figure: グラフ $P(G)$

- ヒープ $H_T(v)$，$H_{out}(v)$ をマージしてグラフ $D(G)$ を作成．（$D(G)$ の頂点の出次数は高々 3 となることに注意）
- $D(G)$ に根となる頂点 $r(s)$ を追加し，$r(s)$ から $h(s) := outroot(s)$ への辺重みを $\delta(h(s))$ とする．
- $D(G)$ 内の辺 (u, v) の重みを $\delta(v) - \delta(u)$ で定める．
- $D(G)$ 内の頂点 p が対応する G 内の辺 (x, y) に対し，辺 $(p, h(y))$ を $D(G)$ 内に追加し，重みを $\delta(h(y))$ で定める．
- $P(G)$ は右図の様になる (※)．なお，上記の作業は全て $O(m + n \log n)$ で行えることに注意．

$P(G)$ の性質

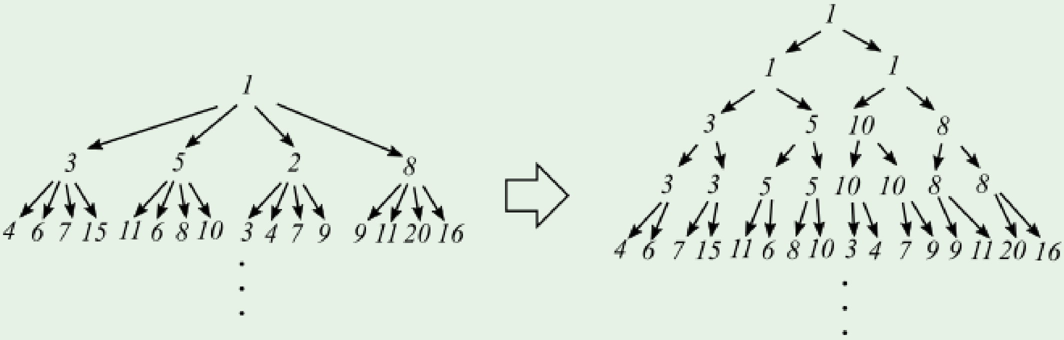
1. $P(G)$ は $O(m + n\log n)$ 個の頂点を持つ．
2. $P(G)$ の頂点の出次数は高々 4 である．
3. $P(G)$ の辺は非負の重みを持つ．
4. G 内の $s-t$ 道と $P(G)$ 内の r から出発した道は 1 対 1 に対応する．
5. $P(G)$ 内の道 p' の重みを l とすると，対応する G 内 $s-t$ 道 p の重みは $d(s,t)+l$ で表せる．

ヒープに関する補題

補題 (Frederickson, 1993)

2 分ヒープ内の k 番目に小さい要素を $O(k)$ で取得できる.

4 分ヒープから 2 分ヒープへの変形



Reference

- D.Eppstein: “Finding the k shortest paths” SIAM Journal on Computing 28(2) (1997)
- G.N.Frederickkson: “An optimal algorithm for selection in a min-heap” Information and Computation 104 (1993)

Appendix

$P(G)$ の性質の証明

1: $v \in V$ に対し，(i)$H_T(v)$ の頂点のうちラベルが付与されたものと，(ii)$H_{out}(v)$ 内の頂点全てと，(iii)$r(s)$ が $P(G)$ の頂点となる．
(i) は $O(\sum_{k=1}^{n} \log k) = O(n \log n)$，(ii) は $O(m)$，(iii) は $O(1)$ であるから，$O(V(P(G))) = O(m + n \log n)$.
2: $D(G)$ の出次数が高々 3 であることと，各頂点が 1 本だけ $crossedge$ を持つことから明らか．
3: $D(G)$ がヒープであるので，各辺 (u, v) は $\delta(v) - \delta(u) \geq 0$ を満たす．また，G の辺重みが非負であるので $\forall v \in P(G); \delta(h(v)) \geq 0$.
4，5: 省略