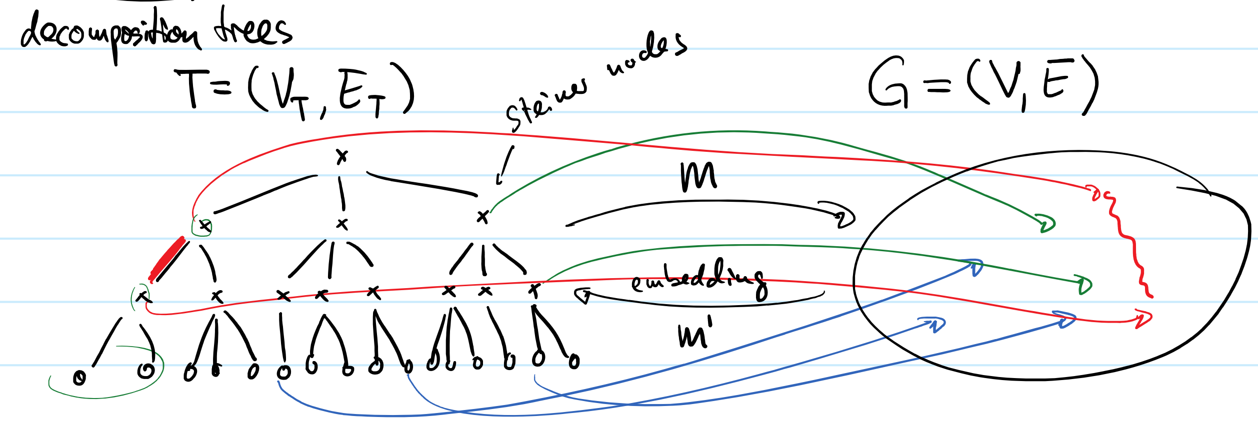
Cut-Based Tree Decompositions

10 May 2019 09:40

last week: distance-based tree decompositions [Räcke 2008]

decomposition trees

$T = (V_T, E_T)$ $\qquad\qquad$ $G = (V, E)$



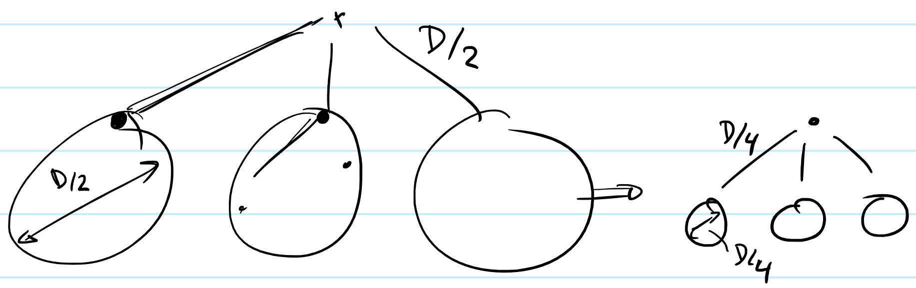
$m_V : V_T \to V$ (one-to-one mapping between the leaves of T and V)

$m_E : E_T \to 2^E$ for $\{u_t, v_t\} \in E_T$ $\;m_E(\{u_t, v_t\}) \to$ shortest path between $m_V(u_t)$ and $m_V(v_t)$

$m'_V : V \to V_T$ (maps to leaves in one-to-one manner)

$m'_E : E \to 2^{E_T}$ for $\{u, v\} \in E$ $\;m'_E(\{u, v\}) \to$ unique path between $m'_V(u)$ & $m'_V(v)$

from now on: $d_T(u_t, v_t)$: length of corresponding path in G



Mapping Multicommodity Flows

10 May 2019 09:40

Given $G=(V,E,c)$

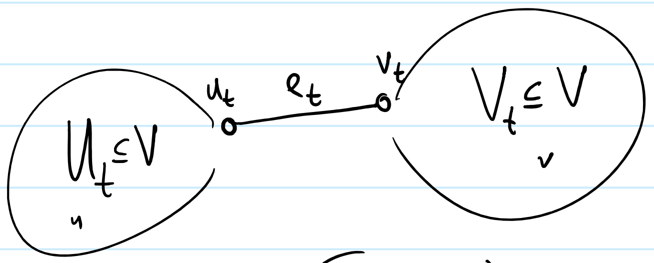
multicomm. flow instance given by requirements $r(u,v)$ for $\{u,v\} \in \binom{V}{2}$

when solving a multicomm. flow problem on T:

(requirements are only between leaves)

each comm. $\{u,v\}$ is routed on the unique path conn. the 2 leaves

Load of an edge $e \in E$ when mapping the flow from T to G



$$r(e_t) = \sum_{u \in U_t,\, v \in V_t} r(u,v)$$

$$\mathrm{load}_T(e) := \sum_{e_t \in E_T :\, e \in \underbrace{m_E(e_t)}_{\text{path repr. } e_t \text{ in } G}} r(e_t)$$

ultimate goal:

for multicomm. flow inst., find tree T

s.t. $\max_e \frac{\mathrm{load}_T(e)}{c(e)}$ is minimized

The Minimum Communication Cost Tree (MCCT) Problem

undir. $G=(V,E,\ell)$ $\quad \ell(e)$: length of e $\quad \ell(e) \geq 0$ $\quad d_G(u,v)$: length of shortest path

$\forall \{u,v\} \in \binom{V}{2}$: $\underset{\text{requirement}}{r(u,v)}$: amount of traffic that has to go between u & v

Goal: Find decomp. tree T that minimizes

$$\mathrm{cost}(T) = \sum_{\{u,v\} \in \binom{V}{2}} d_T(u,v) \cdot r(u,v) \quad \geq \sum_{\{u,v\} \in \binom{V}{2}} d_G(u,v)\, r(u,v)$$

Lemma: MCCT can be $O(\log n)$-approximated.

Proof: use random tree constr. from last lecture:

$$\mathbb{E}[\mathrm{cost}(T)] = \mathbb{E}\Big[\sum_{u,v} d_T(u,v) \cdot r(u,v)\Big] = \sum_{u,v} r(u,v) \cdot \underbrace{\mathbb{E}[d_T(u,v)]}_{\underbrace{O(\log n) \cdot d_G(u,v)}_{\text{last lecture}}} = O(\log n) \cdot \underbrace{\sum_{u,v} d_G(u,v) \cdot r(u,v)}_{\text{lower bd. on opt. cost}}.$$

MCCT Problem

10 May 2019 09:40

$$\mathrm{cost}(T) = \sum_{u,v} d_T(u,v) \cdot r(u,v)$$

Alternative formulation of MCCT

Lemma:
$$\mathrm{cost}(T) = \sum_{e \in E} \mathrm{load}_T(e) \cdot \ell(e)$$

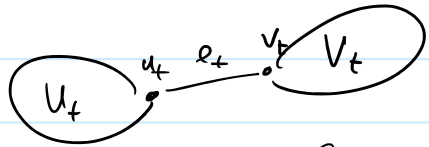
Proof:
$$\sum_{u,v} d_T(u,v)\, r(u,v) = \sum_{u,v} r(u,v) \cdot \sum_{e_t \in P_{uv}} \sum_{e \in M_E(e_t)} \ell(e)$$

P_{uv}: path conn. u & v in T

$$= \sum_{e} \ell(e) \cdot \underbrace{\sum_{e_t : e \in M_E(e_t)} \underbrace{\sum_{\substack{u,v \text{ s.t. } \\ u,v \text{ are sep. in} \\ T \text{ by } e_t}} r(u,v)}}_{\mathrm{load}_T(e)}$$

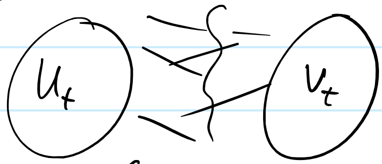
Approximating Cuts / Bottlenecks of a Graph by a Tree

Given: $G = (V, E, c)$ & $T = (V_T, E_T)$



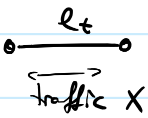
capacity of tree edge $\{u_t, v_t\} \in E_T$

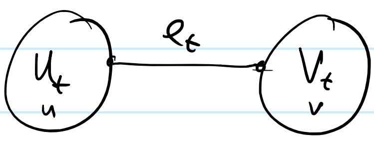
cap. $C_T(u_t, v_t) := \sum_{u \in U_t, v \in V_t} c(u,v)$

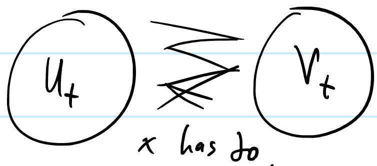


$C_T(u_t, v_t)$ = total cap. over this cut

Theorem: Given multicomm. flow problem, let C_G be the opt. congestion on G and let C_T be the opt. cong. on T. Then, $C_T \leq C_G$.







x has to go over cut

Load on edge $e \in E$ for tree T

10 May 2019 09:40

Assume that each edge $e_t \in E_T$ is fully utilized

↳ $c_T(e_t)$ units of traffic on e_t

$$\mathrm{load}_T(e) = \sum_{e_t \in E_T : e \in m_E(e_t)} c_T(e_t)$$

↳ the same as when routing a multicomm. flow with $r(u,v) = c(u,v)$

$$\mathrm{rload}_T(e) := \frac{1}{c(e)} \cdot \mathrm{load}_T(e)$$

Goal: Find a distribution over trees T s.t. max (over all e) expected $\mathrm{rload}_T(e)$ is minimized.

trees $T_1, \dots, T_k$ and $0 \le \lambda_1, \dots, \lambda_k \le 1$ s.t. $\sum \lambda_i = 1$

$$\text{minimize} \quad \beta := \max_{e \in E} \left\{ \sum_{i=1}^{k} \lambda_i \, \mathrm{rload}_{T_i}(e) \right\}$$

Claim: Given multicomm. flow f_i for each T_i s.t. congestion on T_i is ≤ 1

When mapping $\sum \lambda_i f_i$ to G → cong. on G is $\le \beta$

Proof: f_i uses each $e_t \in E_{T_i}$ at most to its cap. $c_{T_i}(e_t)$

f_i incurs load on $e \in E$ a load of at most $\lambda_i \, \mathrm{load}_{T_i}(e)$

$$\text{total rel. load on } e \le \sum_{i=1}^{k} \lambda_i \, \mathrm{rload}_{T_i}(e) \overset{!}{\le} \beta$$

Why is this what we need?

Multicomm. flow instance $r(u,v)$, assume opt. cong. on G is C_G

flow on $e \le C_G \cdot c(e)$

We showed that for each T_i : opt. cong. on T_i is $C_T \le C_G$

flow on T_i has cong. $\le C_G$

convex comb. of flows on T_i mapped to G → cong. $\le \beta \cdot C_G$ on G.

Finding a distribution on trees

~~10 May 2019 09:40~~

Goal: $\min \beta$

s.t. $\forall e \in E: \quad \sum_i \lambda_i \cdot \mathrm{rload}_{T_i}(e) \le \beta$

$$\sum \lambda_i \ge 1$$

$$\lambda_i \ge 0$$

think of this as a distribution over all possible trees $T_i \in \mathcal{T}$

matrix formulation $|E| \times |\mathcal{T}|$ matrix M, $M_{e,T} = \mathrm{rload}_T(e)$

comp.-wise

edge constr. $M \cdot \vec{\lambda} \le \beta \cdot \mathbb{1}$ ← all-1-vector

$\max(\vec{x})$: maximum component $\qquad \max(M\vec{\lambda}) \le \beta$

replace max by a smooth approx.

$$\mathrm{lmax}(\vec{x}) := \ln\Big(\sum_{e \in E} e^{x_e}\Big) \ge \max(\vec{x}) = \max_{e \in E}\{x_e\}$$

replace $\max(M\vec{\lambda}) \le \beta$ by the stronger cond. $\boxed{\mathrm{lmax}(M\vec{\lambda}) \le \beta}$

Basic Idea

- start with $\lambda_1 = \lambda_2 = \dots = 0$, $\mathrm{lmax}(M\vec{\lambda}) = O(\log n)$
- in each step, choose a tree T_i and $\delta_i > 0$ and update $\lambda_i := \lambda_i + \delta_i$

$\mathrm{lmax}(M\vec{\lambda})$ increases by at most $\delta_i \cdot \beta$ (for $\beta = O(\log n)$)

- stop when $\sum \lambda_i \ge 1$

Implementing basic idea

need to understand how lmax changes when some λ_i

What is $\mathrm{lmax}(\vec{x} + \vec{\varepsilon}) \approx \mathrm{lmax}(\vec{x}) + \vec{\varepsilon}^{T} \cdot \nabla \mathrm{lmax}(\vec{x}) = \mathrm{lmax}(\vec{x}) + \sum_e \varepsilon_e \cdot \mathrm{partial}_e(\vec{x})$

$$\mathrm{partial}_e(\vec{x}) := \frac{\partial}{\partial x_e} \mathrm{lmax}(\vec{x}) = \frac{e^{x_e}}{\sum_e e^{x_e}}$$

10 May 2019 09:40

Lemma: For $\vec{\varepsilon}$ s.t. $\varepsilon_e \in [0,1]$ for all $e \in E$,

$$\mathrm{lmax}(\vec{x}+\vec{\varepsilon}) \le \mathrm{lmax}(\vec{x}) + 2 \cdot \sum_{e\in E} \varepsilon_e \cdot \mathrm{partial}'_e(\vec{x})$$

Assume, we set $\lambda_i = \lambda_i + \delta_i$ how does $\mathrm{lmax}(M\vec{\lambda})$ change

$$\vec{\delta} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \delta_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$\mathrm{lmax}(M(\vec{\lambda}+\vec{\delta}))$

$= \mathrm{lmax}(M\vec{\lambda} + M\vec{\delta})$

if $0 \le (M\vec{\delta})_e \le 1$

$\le \quad + 2 \sum_{e\in E} (M \quad e \cdot \mathrm{partial}'_e(M\vec{\lambda})$

$= \mathrm{lmax}(M\vec{\lambda}) + 2\delta_i \cdot \sum_e \mathrm{rload}_{T_i}(e) \cdot \mathrm{partial}'_e(M\vec{\lambda})$

$(M\vec{\delta})_e = \delta_i \, \mathrm{rload}_{T_i}(e)$

define

$$\delta_i := \frac{1}{\max_e \mathrm{rload}_{T_i}(e)}$$

$$= \mathrm{lmax}(M\vec{\lambda}) + 2\delta_i \cdot \underbrace{\sum_e \mathrm{load}_{T_i}(e) \frac{e^{(M\vec{\lambda})_e}}{c(e) \underbrace{\sum_{e'} e^{(M\vec{\lambda})_{e'}}}_{l(e)}}}$$

find a tree that min. this

MCCT problem ← can find a T_i s.t. this whole thing is $O(\log n)$

can compute sol. s.t. $\sum_e \mathrm{load}_{T_i}(e)\, l(e) \le O(\log n) \cdot \sum_{e\in E} l(e) \cdot c(e)$

$$\sum_{e\in E} c(e) \cdot l(e) = \sum_{e\in E} c(e) \cdot \frac{e^{(M\vec{\lambda})_e}}{c(e) \cdot \sum_{e'} e^{(M\vec{\lambda})_{e'}}} = 1$$

Termination / # iterations

Lemma: $\#\mathrm{iter} = O(|E| \cdot \log n)$

$\phi := \sum_e \underbrace{\sum_i \lambda_i \, \mathrm{rload}_{T_i}(e)}$ → we know $\phi = O(|E| \cdot \log n)$ at all times

potential diff. in one step:

$$\delta_i = \frac{1}{\max_e \mathrm{rload}_{T_i}(e)}$$

ϕ grows by ≥ 1 in each step.