The Power of Vertex Sparsifiers in Dynamic Graph Algorithms

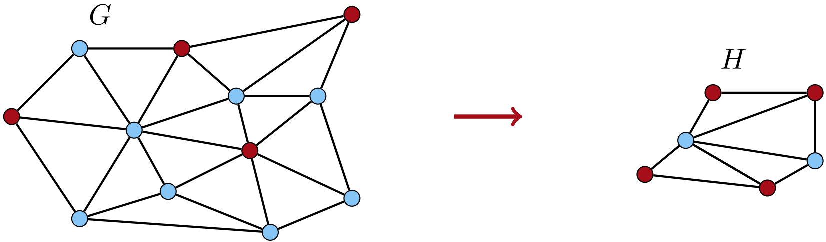
Gramoz Goranci[1], Monika Henzinger[1], Pan Peng[2]

[1]University of Vienna, Austria [2]University of Sheffield, UK

HALG 2018

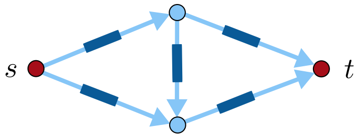
Vertex Sparsification [Moitra '09]

- **input:** a large undirected/directed graph G and some important vertices (a.k.a. **terminals**).
- **goal:** remove **non-terminals** while preserving some **property** among terminals.
- **properties:** minimum cuts, distances, reachability



Effective Resistance

- undirected graph $G = (V, E)$, edge **resistances** r_e
- **source** vertex s, **sink** vertex t
- **Laplacian**: $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- **psuedo-inverse** of $\mathbf{L}$: $\mathbf{L}^\dagger$



$s - t$ Effective Resistance $R_G(s, t)$

- voltage difference across s and t when a unit current source is applied to them:

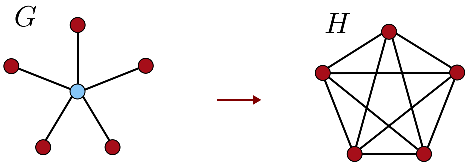
$$R_G(s, t) := (\mathbf{1}_s - \mathbf{1}_t)^\intercal \mathbf{L}^\dagger (\mathbf{1}_s - \mathbf{1}_t)$$

Vertex Resistance Sparsifiers

- graph $H = (V', E', r')$ with $T \subset V'$ is a $\alpha-$**vertex resistance sparsifier** of a graph $G = (V, E, r)$ if for all $u, v \in T$

$$\alpha \cdot R_G(u,v) \leq R_H(u,v) \leq R_G(u,v)$$

- α denotes **quality**, $|V'|$ denotes **size** of sparsifier



Dynamic Effective Resistance Problem

Given initial graph G, build a data-structure that supports the following operations:

- $\textsc{Insert}(u, v, r)$: insert the edge (u, v) with resistance r in G
- $\textsc{Delete}(u, v)$: delete the edge (u, v) from G
- $\textsc{EffectiveResistance}(s, t)$: return the exact (approximate) effective resistance $R_G(s, t)$ between s, t in the current graph G

Motivation

- natural, fundamental problem
- recent work in the static setting, e.g., laplacian solvers [ST '04], and many others afterwards...
- can dynamic effective resistance (or, electrical flow) help solving dynamic max-flow? (similar to the static setting [CKMST '13])

Our Results

Graph	Approx.	Update time	Query time	Ref.
general	exact	$\mathcal{O}(n^{2.373})$	$\mathcal{O}(1)$	Naïve
general	exact	$\mathcal{O}(n^{1.575})$	$\mathcal{O}(n^{0.575})$	[San '04]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(n)$	[ADKKP '16]

Our Results

Graph	Approx.	Update time	Query time	Ref.
general	exact	$\mathcal{O}(n^{2.373})$	$\mathcal{O}(1)$	Naïve
general	exact	$\mathcal{O}(n^{1.575})$	$\mathcal{O}(n^{0.575})$	[San '04]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(n)$	[ADKKP '16]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{2/3})$	$\tilde{\mathcal{O}}(n^{2/3})$	[GHP '17]

Our Results

Graph	Approx.	Update time	Query time	Ref.
general	exact	$\mathcal{O}(n^{2.373})$	$\mathcal{O}(1)$	Naïve
general	exact	$\mathcal{O}(n^{1.575})$	$\mathcal{O}(n^{0.575})$	[San '04]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(n)$	[ADKKP '16]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{2/3})$	$\tilde{\mathcal{O}}(n^{2/3})$	[GHP '17]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{1/2})$	$\tilde{\mathcal{O}}(n^{1/2})$	[GHP '18]

Our Results

Graph	Approx.	Update time	Query time	Ref.
general	exact	$\mathcal{O}(n^{2.373})$	$\mathcal{O}(1)$	Naïve
general	exact	$\mathcal{O}(n^{1.575})$	$\mathcal{O}(n^{0.575})$	[San '04]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(n)$	[ADKKP '16]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{2/3})$	$\tilde{\mathcal{O}}(n^{2/3})$	[GHP '17]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{1/2})$	$\tilde{\mathcal{O}}(n^{1/2})$	[GHP '18]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(m^{4/5})$	$\tilde{\mathcal{O}}(m^{4/5})$	[DGGP '18]

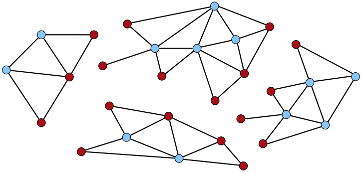
Our Results

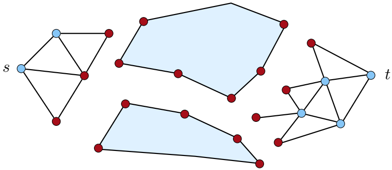
Graph	Approx.	Update time	Query time	Ref.
general	exact	$\mathcal{O}(n^{2.373})$	$\mathcal{O}(1)$	Naïve
general	exact	$\mathcal{O}(n^{1.575})$	$\mathcal{O}(n^{0.575})$	[San '04]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(n)$	[ADKKP '16]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{2/3})$	$\tilde{\mathcal{O}}(n^{2/3})$	[GHP '17]
planar	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(n^{1/2})$	$\tilde{\mathcal{O}}(n^{1/2})$	[GHP '18]
general	$(1-\varepsilon)$	$\tilde{\mathcal{O}}(m^{4/5})$	$\tilde{\mathcal{O}}(m^{4/5})$	[DGGP '18]

- [GHP '18] Assuming **OMv** conjecture, no algorithm can maintain **exact** $R_G(s,t)$ in $O(n^{1-\delta})$ update and $O(n^{2-\delta})$ query time.

Techniques

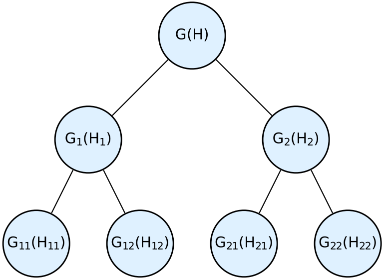
- Graph Clustering
- Vertex Sparsification





$\mathcal{O}(n^{2/3})$

- Nested Dissection
- Vertex Sparsification



$\mathcal{O}(n^{1/2})$

Thank You!