

Fast Approximations for Metric TSP

ISMP 2018, Bordeaux, France

Kent Quanrud, UIUC

joint work w/ Chandra Chekuri, UIUC

Results from 2 papers

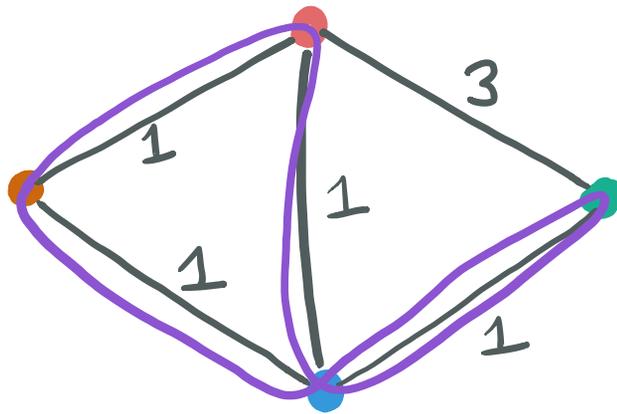
- Approximating the Held-Karp Bound for Metric TSP in nearly linear time (FOCS 2017)
- Fast Approximations for metric TSP via Linear programming (Arxiv, 2018)

Metric TSP - sparse / implicit version

Input: undirected graph $G=(V,E)$,
edge costs $c: E \rightarrow \mathbb{R}_{>0}$

$$\begin{bmatrix} m = |E| \\ n = |V| \end{bmatrix}$$

Goal: find tour* of G of minimum total cost

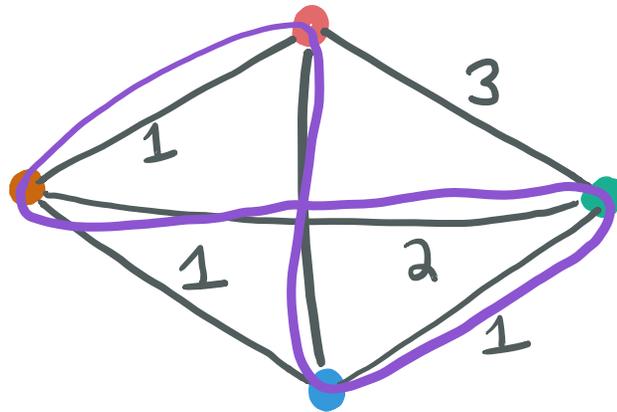


* closed walk visiting all vertices at least once

Metric TSP - explicit/dense version

Input: metric $d: V \times V \rightarrow \mathbb{R}_{>0}$ on K_n

Goal: Find a Hamiltonian cycle* of min. total cost



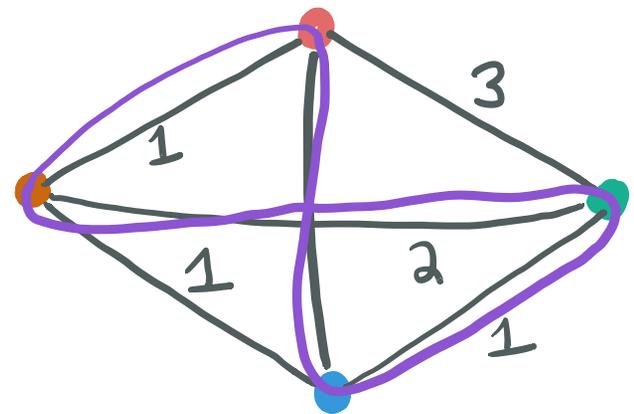
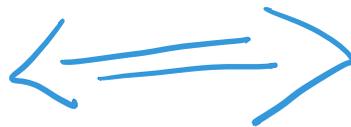
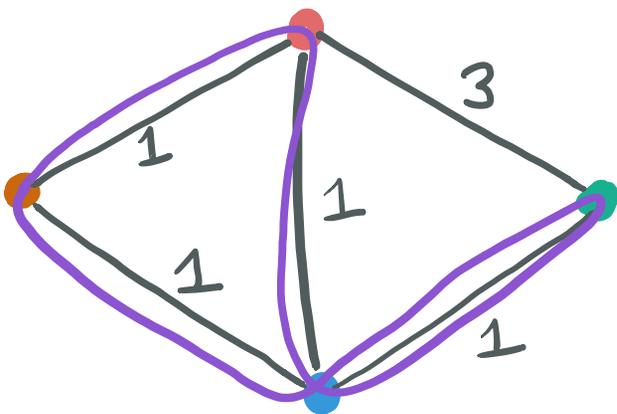
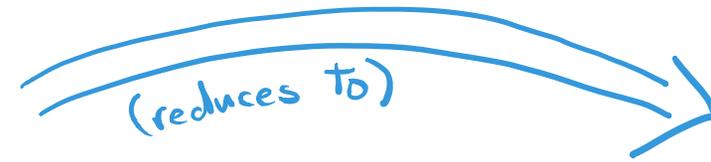
* cycle visiting each vertex exactly once

The same problem...

compute shortest-path
("SP") metric

IMPLICIT
Min-cost tour
 $G = (V, E)$,
 $c: E \rightarrow \mathbb{R}_{>0}$

EXPLICIT
Min Hamiltonian
cycle in metric
 $d: V \times V \rightarrow \mathbb{R}$



Approximating metric TSP

[abbr. previous results]

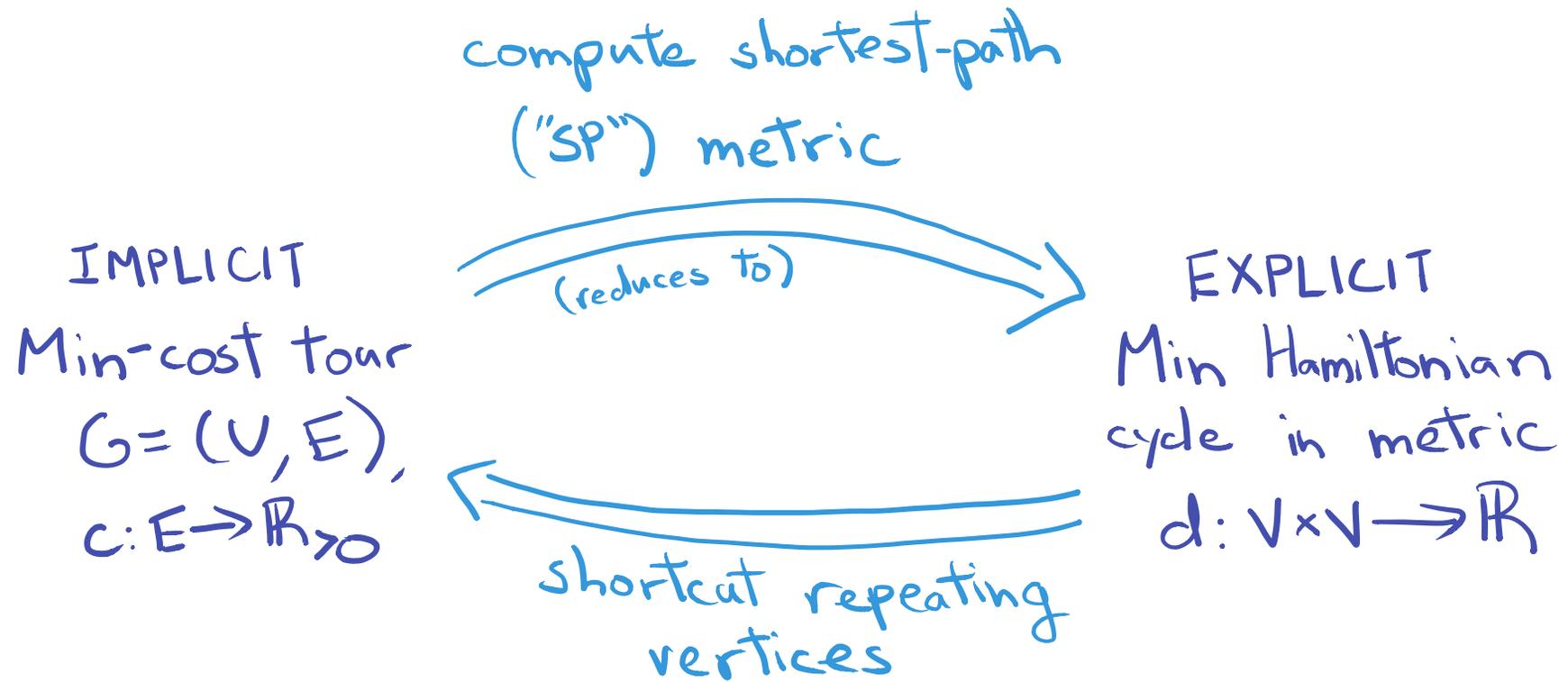
- Held-Karp ("HK") relaxation [1970]
 - lower bound on OPT w/ $< \frac{3}{2}$ integrality gap
 - solvable by ellipsoid algorithm
 - $(1+\epsilon)$ -approx in $\tilde{O}(\frac{m^2}{\epsilon^2})$ time [Garg & Khandekar, 2004]

• Christofides' heuristic [1976]

- computes tour of length $< \frac{3}{2}$ times HK bound
- explicit metric: $\tilde{O}(n^{2.5})$ time \leftarrow [Gabow & Tarjan, 1991]
- implicit shortest path metric: $\begin{cases} \tilde{O}(mn + n^{2.5}) \\ \tilde{O}(m^{1.5}) \end{cases}$ time \leftarrow [Berman et al 1999]

Notable recent breakthroughs for variants and special cases

The same problem...



or different: min-cost tour version is sparse

- computing shortest path metric takes $\tilde{O}(mn)$
- writing down shortest path metric takes $O(n^2)$

Metric TSP - sparse version

Input: undirected graph $G=(V,E)$,
edge costs $c: E \rightarrow \mathbb{R}_{\geq 0}$

$$\begin{cases} m = |E| \\ n = |V| \end{cases}$$

Goal: find tour of G of minimum total cost

Q: How fast can we approximate metric TSP?

Can we do better than:

- computing all pairs shortest paths? $[O(mn)]$
- writing down the shortest path metric? $[O(n^2)]$

Primary Results

1. $(1+\epsilon)$ -approximation for the Held-Karp bound
in $\tilde{O}\left(\frac{m}{\epsilon^2}\right)$ randomized time

- improves $\tilde{O}\left(\frac{m^2}{\epsilon^2}\right)$ time
- gives LP certificate

uses LP
certificate

2. $(1+\epsilon)^{\frac{3}{2}}$ -approximate min-cost tour in $\tilde{O}\left(\frac{m}{\epsilon^2} + \frac{n^{1.5}}{\epsilon^3}\right)$
randomized time

- faster than writing down (let alone computing) SP metric
- $\tilde{O}\left(\frac{n^2}{\epsilon^2} + \frac{n^{1.5}}{\epsilon^3}\right)$ = nearly-linear time for explicit metrics
- improves $\left\{ \begin{array}{l} \tilde{O}(mn + n^{2.5}) \\ \tilde{O}(n^{2.5}) \end{array} \right\}$ for $\left\{ \begin{array}{l} \text{implicit} \\ \text{explicit} \end{array} \right\}$ metrics

Plan for the rest of the talk

I. Held-Karp bound in nearly-linear time

II Accelerating Christofides' Heuristic

III Conclusion

Subtour Elimination LP (for explicit metrics)

Input: metric $d: V \times V \rightarrow \mathbb{R}$

LP: $\min \sum_{e \in E} c_e x_e$ over $x: E \rightarrow \mathbb{R}_{\geq 0}$

[degree constraints] s.t. $\sum_{e \in \partial(v)} x_e = 2$ for all vertices $v \in V$

[subtour elimination] $\sum_{e \in \partial(U)} x_e \geq 2$ for all $\emptyset \subsetneq U \subsetneq V$

$\partial(U) =$ edges cut
by U

$0 \leq x_e \leq 1$ for all edges $e \in E$

Equivalent to "one-tree" lower bound of Held and Karp

2-Edge Connected Spanning Subgraph (2ECS)

INPUT: Graph $G=(V,E)$, edge costs $c: E \rightarrow \mathbb{R}_{\geq 0}$

LP: $\min \sum_{e \in E} c_e y_e$ over $y: E \rightarrow \mathbb{R}_{\geq 0}$

[2-edge connectivity] s.t. $\sum_{e \in C} y_e \geq 2$ for all cuts $C \in \mathcal{C}$
[\mathcal{C} = family of all cuts]

• equivalent to subtour elimination LP on

metric completion of (G,c) [Cunningham] [Goemans & Bertsimas]

• pure covering LP (unlike subtour elim.)

Dual of 2ECSS: Packing Cuts

INPUT : $G=(V,E)$ w/ cuts \mathcal{C} , edge costs $c: E \rightarrow \mathbb{R}_{>0}$

$$\text{LP: } \max \sum_{C \in \mathcal{C}} x_C \quad \text{over } \underline{x: \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}}$$

$$\text{s.t. } \underline{\sum_{C \ni e} x_C \leq c_e} \quad \text{for all edges } e \in E$$

X_{BAD} " $x: \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ " has exponentially large dimension

+_{GOOD} " $\sum_{C \ni e} x_C$ " is a $\{0,1\}$ incidence matrix

Basic Idea: Simpler problems via Lagrangian relax.

$$w \in \mathbb{R}_{>0}^m$$

$$\begin{aligned} \max \langle \mathbb{1}, x \rangle \\ \text{s.t. } Ax \leq c \end{aligned}$$

\Rightarrow

$$\begin{aligned} \max \langle \mathbb{1}, x \rangle \\ \langle w, Ax \rangle \leq \langle w, \mathbb{1} \rangle \end{aligned}$$

\Rightarrow

$$\operatorname{argmin}_j \langle w, Ae_j \rangle$$

many
packing
constraints

1 packing
constraint

greedy
solution

- positive weights $w \in \mathbb{R}_{>0}^m$ collapse many packing constraints to 1 \Rightarrow trivial greedy sol'n
- each w_i controls "importance" of i th constraint

From packing cuts to finding min-cuts

$$\max \sum_c x_c$$

$$\text{s.t. } \sum_{c \ni e} x_c \leq c_e \quad (e \in E)$$

Packing cuts into
capacitated edges

$$\max \sum_c x_c$$

$$\text{s.t. } \sum_e w_e \sum_{c \ni e} x_c \leq \sum_e w_e c_e$$

Lagr. relaxation

$$\min_{w \in \mathbb{R}_{>0}} \sum_{e \in E} w_e$$

min-cut
w/r/t w

- Lagrangian duals $w \in \mathbb{R}_{>0}$ look like edge weights
- "greedy coordinate" is the min-cut w/r/t w .

MWU type algorithm (sketch)

1. $x = \mathbf{0}^c, w = \mathbf{1}^E$

2. repeatedly

a. $C \leftarrow \text{min-cut w/r/t } w$

b. $x_c \leftarrow x_c + \delta$ for some $\delta > 0$

c. update edge weights w_e s.t.

$$w_e = \exp\left(\frac{\log m}{\epsilon} \text{load}(e)\right) \forall e \in E$$

3. output \tilde{x}

$$\text{load}(e) = \frac{\sum_{C \ni e} x_c}{c_e}$$

At the end:

• x is a convex comb. of solutions to Lagr. relaxations

• $\sum_{C \ni e} x_c \leq (1+\epsilon) c_e$ for each edge $e \Rightarrow \tilde{O}\left(\frac{m}{\epsilon^2}\right)$ iterations

• $(1+\epsilon)$ -APX min-cut suffices

• weight w_e is exponential in the load on edge e

• $\delta > 0$ chosen large as possible s.t. no w_e increases by more than $(1+\epsilon)$ -factor

$$\sum_{C \in \mathcal{C}} x_c \geq \text{OPT}$$

MWU-Naive Running Time

1. $x = \mathbf{0}^c, w = \mathbf{1}^E$

2. repeatedly

a. $C \leftarrow \text{min-cut w/r/t } w$

b. $x_c \leftarrow x_c + \delta$ for some $\delta > 0$

c. update edge weights w_e st.

$$w_e = \exp\left(\frac{\log m}{\epsilon c_e} \sum_{c \ni e} x_c\right) \quad \forall e \in E$$

3. output x

$\tilde{O}\left(\frac{m}{\epsilon^2}\right)$ iterations

$\tilde{O}(m)$ time per min-cut

$O(m)$ edge weights per min-cut

$\Rightarrow \tilde{O}\left(\frac{m^3}{\epsilon^2}\right)$ running time

• even having to write down m cuts (w/ up to m edges per cut) suggests $\Omega(m^2)$ time is necessary.

from m^2 to m

we have

we need

$(1+\epsilon)$ -APX
min-cut

$\tilde{O}(m)$ per
iteration

$\tilde{O}(1)$ amortized
per iteration

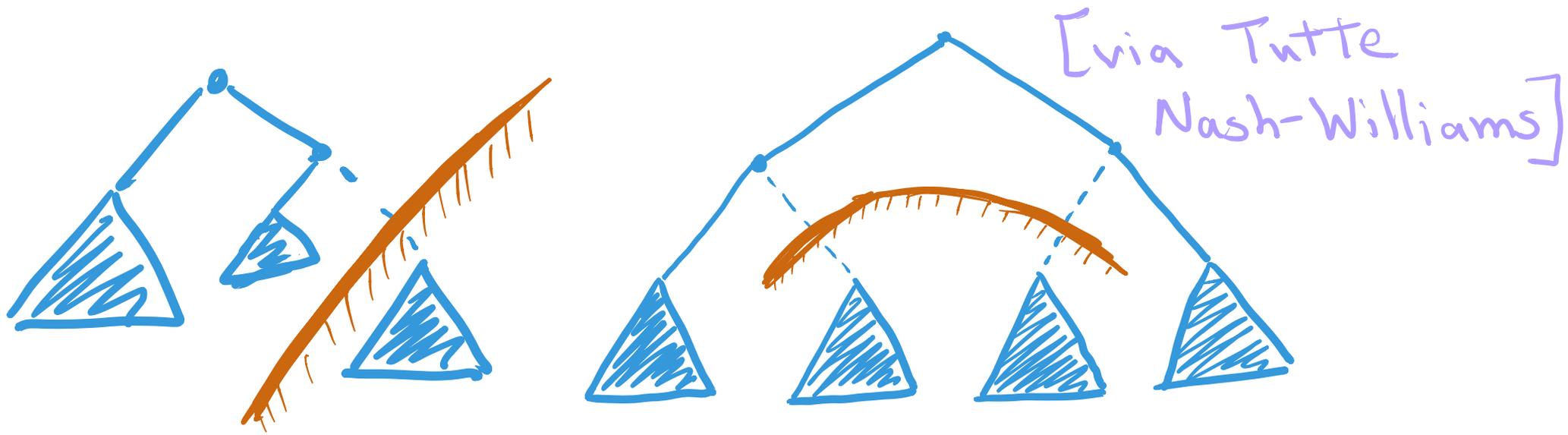
weight
update

$O(m)$ per
iteration

$\tilde{O}(1)$ amortized
per iteration

Spanning trees to the rescue

Karger: If one samples $O(\log n)$ trees from APX-max tree packing^{*}, then w/h/p every $(1+\epsilon)$ -min cut is induced by removing 1 or 2 edges from some sampled tree.



^{*} like a cut packing, but w/ trees

$O(\log n)$ trees induce all APX min-cuts:

1. Finding min-cuts

- search efficiently through trees w/ dynamic programming + dynamic tree data structures [Karger]
- when no more min-cuts left, repack spanning trees in $\tilde{O}(m/\epsilon^2)$ time [Karger] [LQ SODA 17]

2. Updating edge weights along cuts

- Preprocess trees to allow efficient bulk updates via data structures from [LQ SODA 17]
- Incorporate weight updates back into min-cut search

Omitting details, these techniques lead to $\tilde{O}(1)$ time per iteration, $\tilde{O}(\frac{m}{\epsilon^2})$ time overall

- finds m min-cuts faster than writing down m cuts explicitly!

———— Important factors in hindsight ————

Monotonicity: weight updates, dynamic min-cuts in $\tilde{O}(1)$ amortized fails if w can decrease

MWU invariants: subroutines amortized against standard invariants of the MWU framework

Trees, trees, trees: all cuts come from $\tilde{O}(\frac{1}{\epsilon^2})$ trees
 \Rightarrow fast and compressed min-cuts, efficient weight update data structures

II Accelerating Christofides' Heuristic

1. Christofides' heuristic
2. Related LPs and polytopes
3. Graph sparsification
4. 1 + 2 + 3

Christofides' Heuristic [1976]

1. $M \leftarrow$ minimum spanning tree [$c(T) \leq (1 - \frac{1}{h}) \text{OPT}$]

2. $T \leftarrow \{\text{odd degree vertices of } M\}$

3. $J \leftarrow$ min cost T -join* [$c(J) \leq \frac{1}{2} \text{OPT}$]

* subgraph w/ odd degree vertices T

4. $M + J$ is Eulerian. [$c(M+J) \leq (1 - \frac{1}{h}) \frac{3}{2} \text{OPT}$]

Return Eulerian walk on $M+J$

$\Rightarrow (1 - \frac{1}{h}) \frac{3}{2}$ -APX min cost tour

Best APX for metric TSP 740 years later!

3. $J \leftarrow$ min cost T-join*

* subgraph w/ odd degree vertices S

Two approaches:

$[|T| = O(n)$ in worst case]

1. Minimum cost perfect matching on shortest path metric between vertices in T .

$$\left[\begin{array}{l} \text{All-pairs} \\ \text{shortest} \\ \text{paths} \end{array} \right] + \left[\begin{array}{l} \text{min-cost} \\ \text{perfect} \\ \text{matching} \\ \text{on } K_n \end{array} \right] = \tilde{O}(mn) + \tilde{O}(n^{2.5})$$

[Gabow & Tarjan '91]

2. Gadget-based reduction to min-cost perfect matching on aux graph w/ $O(m)$ vertices, $O(m)$ edges

$$\left[\begin{array}{l} \text{min-cost perfect} \\ \text{matching on } m \\ \text{vertices, } m \text{ edges} \end{array} \right] = \tilde{O}(m^{1.5})$$

[Berman, Kahng, Vidhani,
Zelikovsky 1999]

Christofides' Heuristic [1976] Running Time

1. $M \leftarrow$ minimum spanning tree

$\tilde{O}(m)$

2. $T \leftarrow \{\text{odd degree vertices of } M\}$

3. $J \leftarrow$ min cost T -join*

$\min \begin{cases} \tilde{O}(mn + n^{2.5}), \\ \tilde{O}(m^{1.5}) \end{cases}$

* subgraph w/ odd vertices T

4. $M+J$ is Eulerian.

$O(m)$

Return Eulerian walk on $M+J$

Total running time $\min \{ \tilde{O}(mn + n^{2.5}), \tilde{O}(m^{1.5}) \}$

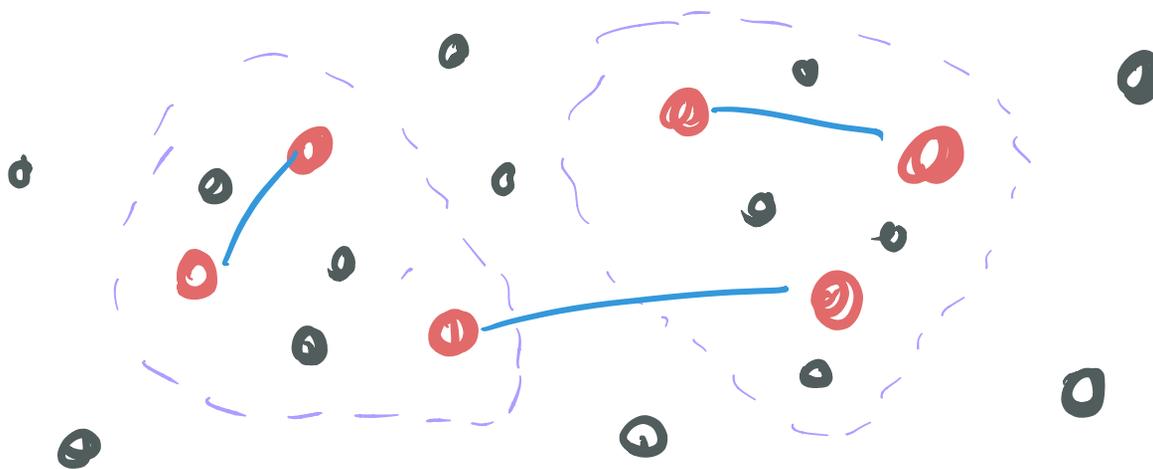
The dominant of the T-join polytope

(Edmonds & Johnson, 1973)

$x \in \mathbb{R}^E$ such that

$$\sum_{e \in S(s)} x_e \geq 1 \quad \forall S \subseteq V, \quad |S \cap T| \text{ odd}$$

$$x_e \geq 0 \quad \forall e \in E$$



$\text{red circle} = T$

$\text{black circle} = V \setminus T$

2ECSS

$$\sum_{e \in \delta(u)} y_e \geq 2 \quad \forall \phi \neq u \neq v$$

$$y_e \geq 0 \quad \forall e \in E$$

Dominant of T-join

$$\sum_{e \in \delta(s)} x_e \geq 1 \quad \forall s \subseteq V, |S \cap T| \text{ odd}$$

$$x_e \geq 0 \quad \forall e \in E$$

[Wolsey 1980]. if $y \in \mathbb{R}^n$ is feasible for 2ECSS, then $y/2$ is in dominant of the T-join polytope for any even $T \subseteq V$

\Rightarrow we can compute T-join in support of y !!

Sparsifying 2ECSS

$$\text{LP: } \min \sum_{e \in E} c_e y_e \quad \text{over } \gamma: E \rightarrow \mathbb{R}_{\geq 0}$$

$$\left[\begin{array}{l} \text{2-edge} \\ \text{connectivity} \end{array} \right] \text{ s.t. } \sum_{e \in C} y_e \geq 2 \quad \text{for all cuts } C \in \mathcal{C}$$

$[\mathcal{C} = \text{family of all cuts}]$

Given $(1+\epsilon)$ -APX solution $\gamma \in \mathbb{R}^E$ (via part I) we want a sparse feasible solution $x \in \mathbb{R}^E$ of similar cost

Interpreting γ as edge weights, we need a sparse set of edge weights maintaining 2-connectivity

Cut Sparsification

[let $\epsilon > 0$]

Given a weighted graph $G=(V, E, \gamma)$, a cut sparsifier is a reweighted sub-graph $H=(V, E', x)$ that preserves the weight of each cut up to $(1 \pm \epsilon)$ multiplicative factor.

Benczur-Karger algorithm can compute a cut sparsifier $H=(V, E', \gamma)$ w/ $|E'| = \tilde{O}\left(\frac{n}{\epsilon^2}\right)$ in $\tilde{O}\left(\frac{m}{\epsilon^2}\right)$ randomized time

Sparsifying 2ECSS

LP: $\min \sum_{e \in E} c_e y_e$ over $y: E \rightarrow \mathbb{R}_{\geq 0}$

[2-edge connectivity] s.t. $\sum_{e \in C} y_e \geq 2$ for all cuts $C \in \mathcal{C}$
[\mathcal{C} = family of all cuts]

Theorem In $\tilde{O}(\frac{m}{\epsilon^2})$ randomized time, one can compute $(1+\epsilon)$ -APX y to 2ECSS w/
 $|\text{support}(y)| \leq \tilde{O}(n/\epsilon^2)$ w/h/p.

Proof: Part I + Benczur - Karger
+ small adjustment to preserve cost

Computing APX T-join

1. Compute $(1+\epsilon)$ -APX x to 2ECSS

Time
 $\tilde{O}\left(\frac{m}{\epsilon^2}\right)$

2. Sparsify x w/ Benczur-Karger to get $(1+\epsilon)$ -APX y to 2ECSS w/ $\|y\|_0 \leq O\left(\frac{n \log n}{\epsilon^2}\right)$

$\tilde{O}(m/\epsilon^2)$

3. y is in dominant of T-join polytope.

$\tilde{O}\left(\frac{n^{3/2}}{\epsilon^3}\right)^*$

compute min-cost T-join in $H=(V, \text{support}(y), y)$

\Rightarrow T-join J w/ cost $\leq \frac{(1+\epsilon)}{2}$ 2ECSS

$\tilde{O}\left(\frac{m}{\epsilon^2} + \frac{n^{3/2}}{\epsilon^3}\right)$

* via sparse reduction of Berman et al

APX - Christofides

1. $M \leftarrow$ minimum spanning tree

$$[c(T) \leq (1 - \frac{1}{n}) \text{OPT}]$$

2. $T \leftarrow \{\text{odd degree vertices of } M\}$

3. $J \leftarrow$ APX min-cost T -join via
sparsified 2ECSS

$$[c(J) \leq \frac{1+\epsilon}{2} \text{OPT}]$$

4. $M + J$ is Eulerian.

$$[c(T+J) \leq (1+\epsilon) \frac{3}{2} \text{OPT}]$$

Return Eulerian walk on $M+J$

$\Rightarrow (1+\epsilon) \frac{3}{2}$ -APX min cost tour

Bottleneck is APX T -join: $\tilde{O}\left(\frac{m}{\epsilon^2} + \frac{n^{3/2}}{\epsilon^3}\right)$

Running times for metric TSP

Metric	$\frac{3}{2}$ -APX	$(1+\epsilon)^{\frac{3}{2}}$ -APX
Shortest paths	$\tilde{O}(mn + n^{2.5})$	$\tilde{O}\left(\frac{m}{\epsilon^2} + \frac{n^{1.5}}{\epsilon^3}\right)$ 
	$\tilde{O}(m^{1.5})$	
Explicit	$\tilde{O}(n^{2.5})$	$\tilde{O}\left(\frac{n^2}{\epsilon^2} + \frac{n^{1.5}}{\epsilon^3}\right)$ 

Concluding Remarks

- Faster APX for Held-Karp bound,
 $\frac{3}{2}$ -min cost tour
- Fast LP toolkit (amortized/randomized weight updates, dynamic oracle) widely applicable (see e.g. SOPA papers)
- APX-Christofides shifts focus from designing fast LP solvers to using fast LP solvers.
- Similar results hold for s-t metric TSP
(in prep.)

THANKS