Fast Approximations for Metric TSP

BIRS workshop, 2018

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

- Fast Approximations for metric TSP via Linear programming

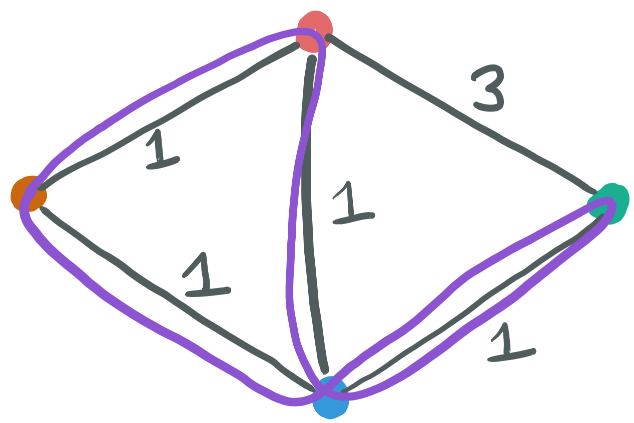
and some unpublished extensions to path TSP

Metric TSP - sparse / implicit version

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

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

Goal: find tour* of G of minimum total cost

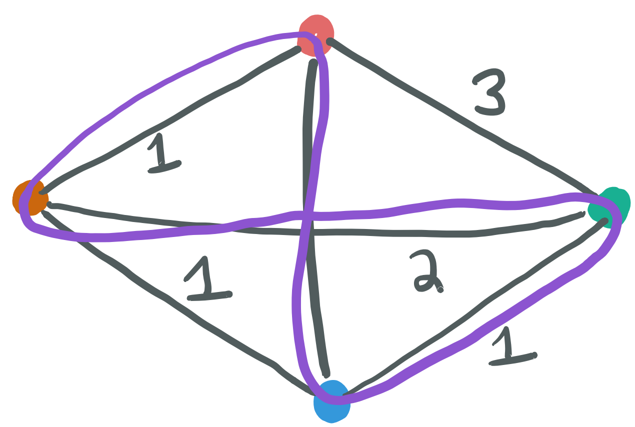


* closed walk visiting all vertices at least once

Metric TSP - explicit/dense version

Input: metric $d: V \times V \to \mathbb{R}_{\geq 0}$ 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

(reduces to) $\Longrightarrow$

IMPLICIT

Min-cost tour

$G=(V,E)$,

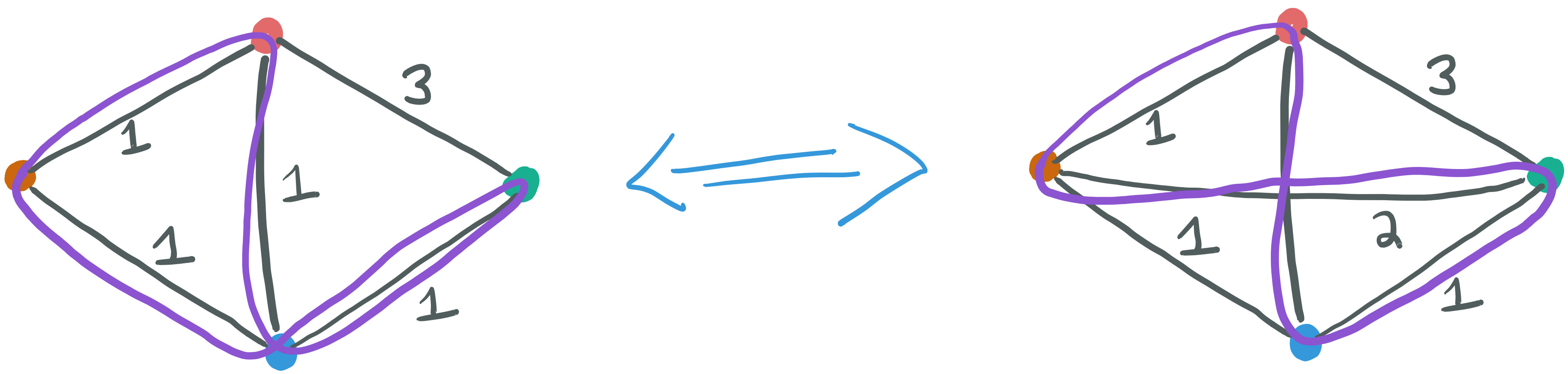
$c: E \to \mathbb{R}_{\geq 0}$

EXPLICIT

Min Hamiltonian cycle in metric

$d: V \times V \to \mathbb{R}$

$\Longleftarrow$ shortcut repeating vertices



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
  - $(1+\varepsilon)$-approx in $\tilde{O}(\frac{m^2}{\varepsilon^2})$ 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 ←---- [Gabow & Tarjan, 1991]
  - implicit shortest-path metric: $\left\{ \begin{array}{l} \tilde{O}(mn + n^{2.5}) \\ \tilde{O}(m^{1.5}) \end{array} \right.$ time ←----[Berman et al 1999]

Notable recent breakthroughs for variants and special cases

The same problem...

compute shortest-path
("SP") metric
(reduces to)

IMPLICIT	EXPLICIT
Min-cost tour	Min Hamiltonian cycle in metric
$G=(V,E)$,	$d: V \times V \to \mathbb{R}$
$c: E \to \mathbb{R}_{>0}$	

shortcut repeating
vertices

or different: min-cost tour version is <u>sparse</u>

- 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 $c: E \to \mathbb{R}_{>0}$ $\quad [m=|E|,\ n=|V|]$

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? $[\tilde{O}(mn)]$
- writing down the shortest path metric? $[O(n^2)]$

Primary Results

1. $(1+\varepsilon)$-approximation for the Held-Karp bound in $\tilde{O}(\frac{m}{\varepsilon^2})$ randomized time
   - improves $\tilde{O}(m^2/\varepsilon^2)$ time
   - gives LP certificate

uses LP certificate

2. $(1+\varepsilon)\frac{3}{2}$-approximate min-cost tour in $\tilde{O}(\frac{m}{\varepsilon^2}+\frac{n^{1.5}}{\varepsilon^3})$ randomized time
   - faster than writing down (let alone computing) SP metric
   - $\tilde{O}(\frac{n^2}{\varepsilon^2}+\frac{n^{1.5}}{\varepsilon^3})$ = nearly-linear time for explicit metrics
   - improves $\left\{\begin{array}{l}\tilde{O}(mn+n^{2.5}),\ \tilde{O}(m^{1.5}) \\ \tilde{O}(n^{2.5})\end{array}\right.$ for $\begin{array}{l}\text{implicit} \\ \text{explicit}\end{array}$ metrics

Plan for the rest of the talk

I. Held-Karp bound in nearly-linear time
(+ path TSP)

II Accelerating Christofides' Heuristic

III Conclusion

Subtour Elimination LP (for explicit metrics)

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

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

[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 (2ECSS)

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

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

[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 $c: E \to \mathbb{R}_{\geq 0}$

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

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

✗ (BAD) "$x: \mathcal{C} \to \mathbb{R}_{\geq 0}$" has exponentially large dimension

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

"MWU framework" in a nutshell

Iteratively increase $x \in \mathbb{R}^{e}_{\geq 0}$ along solutions

to Lagrangian relaxations ①

of the cut packing LP w/r/t

dynamically chosen weights ②

1

Lagrangian relaxations: packing cuts → min-cuts

Packing cuts	⇒	Lagr. relax. (LR)	⇒	min-cuts
$\max \sum_c x_c$		$\max \sum_c x_c$		$\min_{C \in \mathcal{C}} \sum_{e \in C} w_e$
s.t. $\sum_{C \ni e} x_C \leq c_e \ (e \in E)$		s.t. $\sum_e w_e \sum_{C \ni e} x_C \leq \sum_e w_e c_e$		

Given any weights $w \in \mathbb{R}^E_{>0}$ (one per packing constraint)

- many packing constraints → 1 packing constraint in LR
- In LR, w acts as edge weights,

  cost of cut = weight of cut w/r/t w

- LR is solved by (scaling) the min-cut w/r/t w.

Edge weights exponentiate load

2

weights $w \in \mathbb{R}^m_{>0}$ depend on current packing $x \in \mathbb{R}^{\mathcal{C}}_{\geq 0}$

for edge e,

$$\boxed{w_e = e^{\eta \, \mathrm{load}(e)}, \quad \text{where } \mathrm{load}(e) = \frac{\sum_{c \ni e} x_c}{c_e} \qquad \left(\eta \approx \frac{\log m}{\varepsilon}\right)}$$

- monotonically increasing in x
- $1 \leq w_e \leq e^{\eta} \approx m^{O(1/\varepsilon)}$ for feasible x

MWU-Naive Running Time

1. $x = \mathbb{0}^{\mathcal{C}}$, $w = \mathbb{1}^{E}$
2. repeatedly
   a. $C \leftarrow$ 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.

      $w_e = \exp(\eta \, \mathrm{load}(e)) \quad \forall e \in E$
3. output x

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

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

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

$\Rightarrow \tilde{O}\left(\frac{m^3}{\varepsilon^2}\right)$ 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
$(1+\varepsilon)$-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

Incremental min-cut data structure

With $\tilde{O}(m/\varepsilon^2)$ total overhead,

- $\tilde{O}(1)$ amort. time*: returns $(1+\varepsilon)$-APX min-cut
- $\tilde{O}(1)$ amort. time*: simulates/registers a weight update along APX-min-cut,
- $\tilde{O}(1)$ time: increase edge weight by $(1+\varepsilon)$-mult factor

*amortized against invariants of MWU analysis

Quick word on finding min-cuts

- APX-min-cuts induced by $\tilde{O}(1/\varepsilon^2)$ trees from tree packings [Karger]

- Tree structure of cuts allows for efficient "bulk updates"

Omitting details, these techniques lead to $\tilde{O}(1)$ time per iteration, $\tilde{O}(\frac{m}{\varepsilon^2})$ 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 $\tilde{O}(\frac{1}{\varepsilon^2})$ trees $\Rightarrow$ fast and compressed min-cuts, efficient weight update data structures

Extension to Path TSP

(s,t)-Path TSP

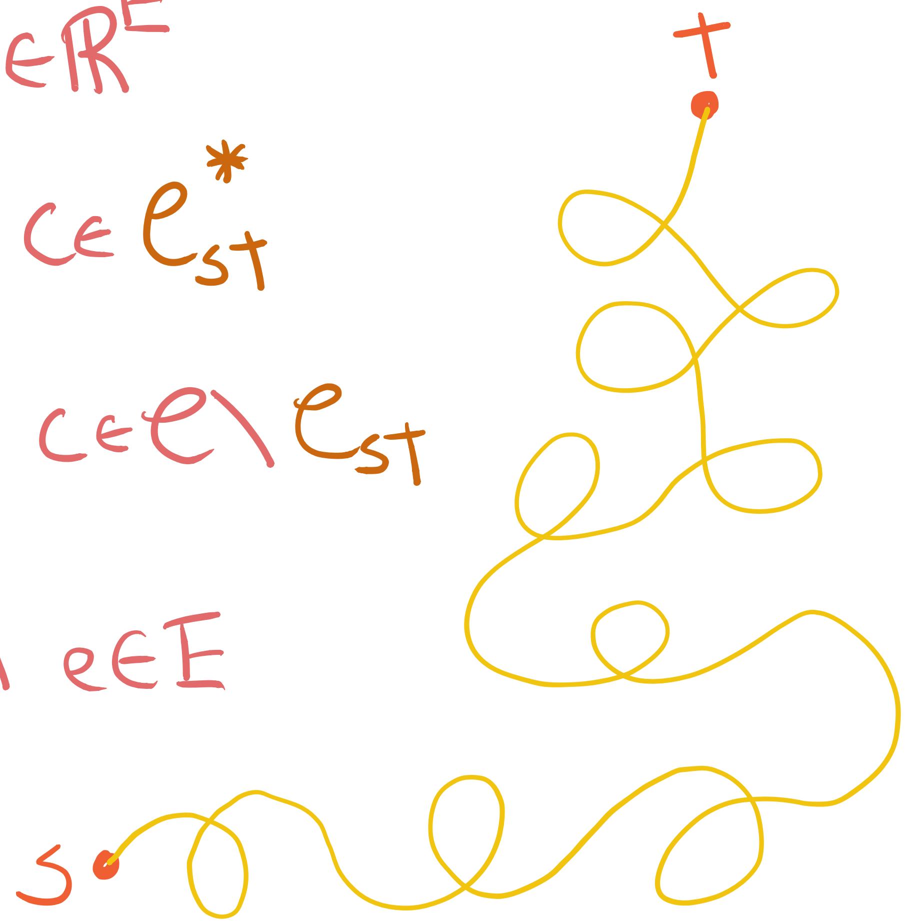
Goal: for fixed s and t, find minimum cost walk from s to t visiting all vertices

LP: $\min \sum_e c_e y_e$ over $y \in \mathbb{R}^E$

s.t. $\sum_{e \in C} y_e \geq 1$ for all $C \in \mathcal{C}_{st}^{*}$

$\sum_{e \in C} y_e \geq 2$ for all $C \in \mathcal{C} \setminus \mathcal{C}_{st}$

$y_e \geq 0$ for all $e \in E$



$^{*}\mathcal{C}_{st} = \{\text{(s,t)-cuts}\}$

Dual LP

$$\max \sum_{C \in \mathcal{C}_{st}} x_C + 2 \sum_{C \in \mathcal{C} \setminus \mathcal{C}_{st}} x_C \quad \text{over } x \in \mathbb{R}^E$$

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

$$x_C \geq 0 \quad \text{for all } C \in \mathcal{C}$$

Packing cuts, except $\{s,t\}$-cuts worth half as much as non-$\{s,t\}$-cuts.

Lagrangian dual

Given $w(e) > 0$ for $e \in E$,

$$\underset{C \in \mathcal{C}}{\text{minimize}} \begin{cases} \sum_{e \in C} w(e) & \text{if } C \in \mathcal{C}_{st} \\ \frac{1}{2} \sum_{e \in C} w(e) & \text{if } C \notin \mathcal{C}_{st} \end{cases}$$

Goal: approximate and then update edge weights in $\tilde{O}(1)$ amortized time

Reduce to (global) min-cut

We maintain underestimate $\lambda < OPT$ and 2 instances of incremental min-cut datastructure

1. One instance of inc-min-cuts over entire graph (returns global min-cuts)

2. One instance of inc-min-cuts over graph w/ s and t contracted.

returns APX-min non-$\{s,t\}$-cuts

Reduction to global min-cut

$\tilde{O}(1)$ amortized

$\Rightarrow \tilde{O}(m/\varepsilon^2)$ overall

Given $\lambda \leq OPT$:

1. Query min non-$\{s,t\}$-cut. If $\leq (1+O(\varepsilon))2\lambda$ take and update along this cut.
2. Failing 1, query min cut. If $\leq (1+O(\varepsilon))\lambda$, take and update along this cut.
3. Failing 1 and 2, set $\lambda \leftarrow (1+\varepsilon)\lambda$

   happens $\leq \tilde{O}(1/\varepsilon^2)$ times

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

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

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

   * subgraph w/ odd degree vertices T

4. $M+J$ is Eulerian. $[c(T+J) \le (1-\frac{1}{n})\frac{3}{2} OPT]$

   Return Eulerian walk on $M+J$

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

Best APX for metric TSP >40 years later!

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

2 algorithms:

* subgraph w/ odd degree vertices S

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

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

[Gabow & Tarjan '91]

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

$$\begin{bmatrix}\text{min-cost perfect} \\ \text{matching on } m \\ \text{vertices, } m \text{ edges}\end{bmatrix} = \tilde{O}(m^{1.5})$$

[Berman, Kahng, Vidhani, Zelikovsky 1999]

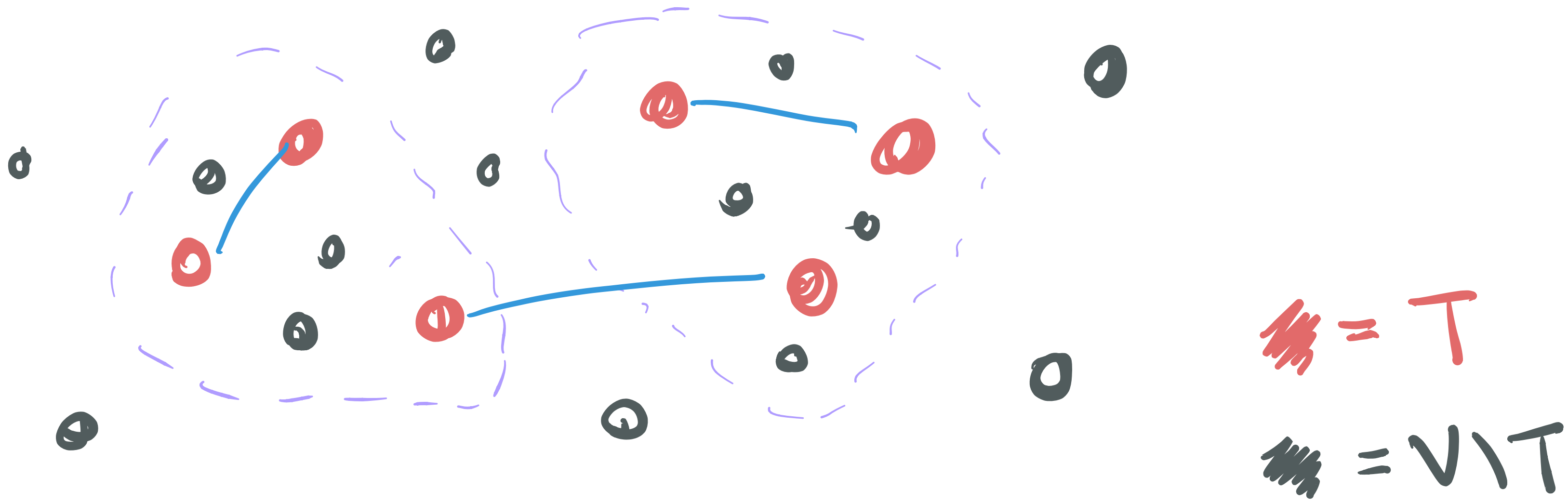
The dominant of the T-join polytope

(Edmonds & Johnson, 1973)

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

$$\sum_{e \in \delta(S)}' x_e \geq 1 \quad \forall\ S \subseteq V, \quad |S \cap T| \text{ odd}$$

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



2ECSS

$$\sum_{e\in\delta(U)}' y_e \geq 2 \quad \forall \emptyset \subsetneq U \subsetneq V$$

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

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

$\Rightarrow$ we prefer y to be <u>sparse</u>

Cut Sparsification

[let $\varepsilon > 0$]

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

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

Sparsifying with 2ECSS

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

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

1. compute $(1+\varepsilon)$-APX solution $y \in \mathbb{R}^E$ to 2ECSS

2. apply Benczur-Karger to y

   $\Rightarrow (1+\varepsilon)$-APX solution x w/ $\tilde{O}(n/\varepsilon^2)$ edges

3. Compute T-Join in support of x

   $\Rightarrow$ T-join w/ cost $\frac{(1+\varepsilon)}{2}(2ECSS)$ in $\hat{O}\left(\frac{m}{\varepsilon^2} + \frac{n^{3/2}}{\varepsilon^3}\right)$ time

APX - Christofides

1. $M \leftarrow$ minimum spanning tree [$c(T) \le (1-\frac{1}{n})\mathrm{OPT}$]
2. $T \leftarrow \{$odd degree vertices of $M\}$
3. $J \leftarrow$ APX min-cost T-join via sparsified 2ECSS [$c(J) \le \frac{1+\varepsilon}{2}\mathrm{OPT}$]
4. $M + J$ is Eulerian. [$c(T+J) \le (1+\varepsilon)\frac{3}{2}\mathrm{OPT}$]
   Return Eulerian walk on $M+J$

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

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

Running times for metric TSP

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

(Brief) Conclusion

- Faster APX for Held-Karp bound, $\frac{3}{2}$-min cost tour

- Fast LP toolkit (amortized/randomized weight updates, dynamic oracle) widely applicable

- APX-Christofides shifts focus from <u>designing</u> fast LP solvers to <u>using</u> fast LP solvers.

THANKS