CS630 Graduate Algorithms

October 8, 2024

by Dora Erdos and Jeffrey Considine

- Midterm reminder - this Thursday!
- Traveling Salesperson
- Open Q&A re: approximation algorithms and other midterm review

Midterm

Thursday, October 10 @ 9:30

Be on time!

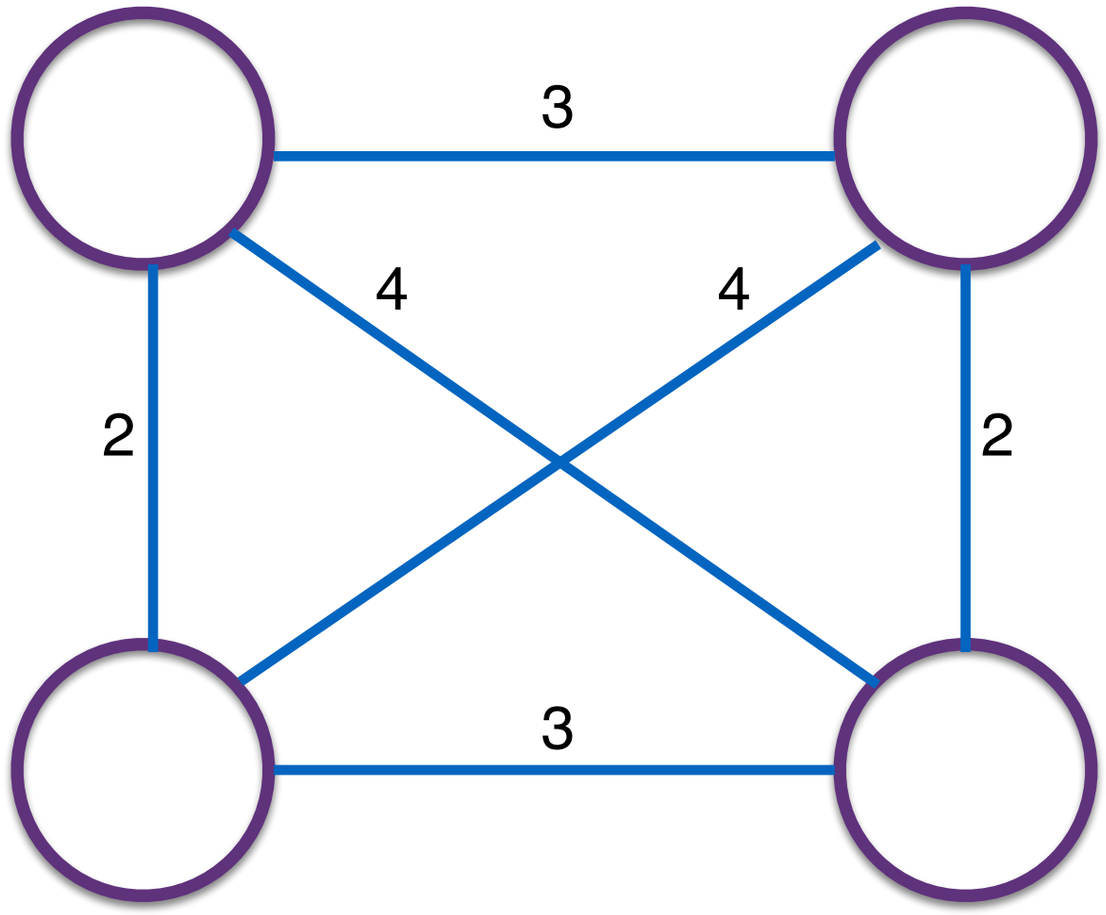
All material so far including today might be included.

You are allowed one cheatsheet.

Also, the final exam was recently scheduled for December 17, 9-11am.
Plan your travel accordingly.

Traveling Salesperson Problem (TSP)

Given a graph where every pair of vertices is directly connected, and each edge has a cost, what is the minimum total cost of a cycle visiting each vertex exactly once?

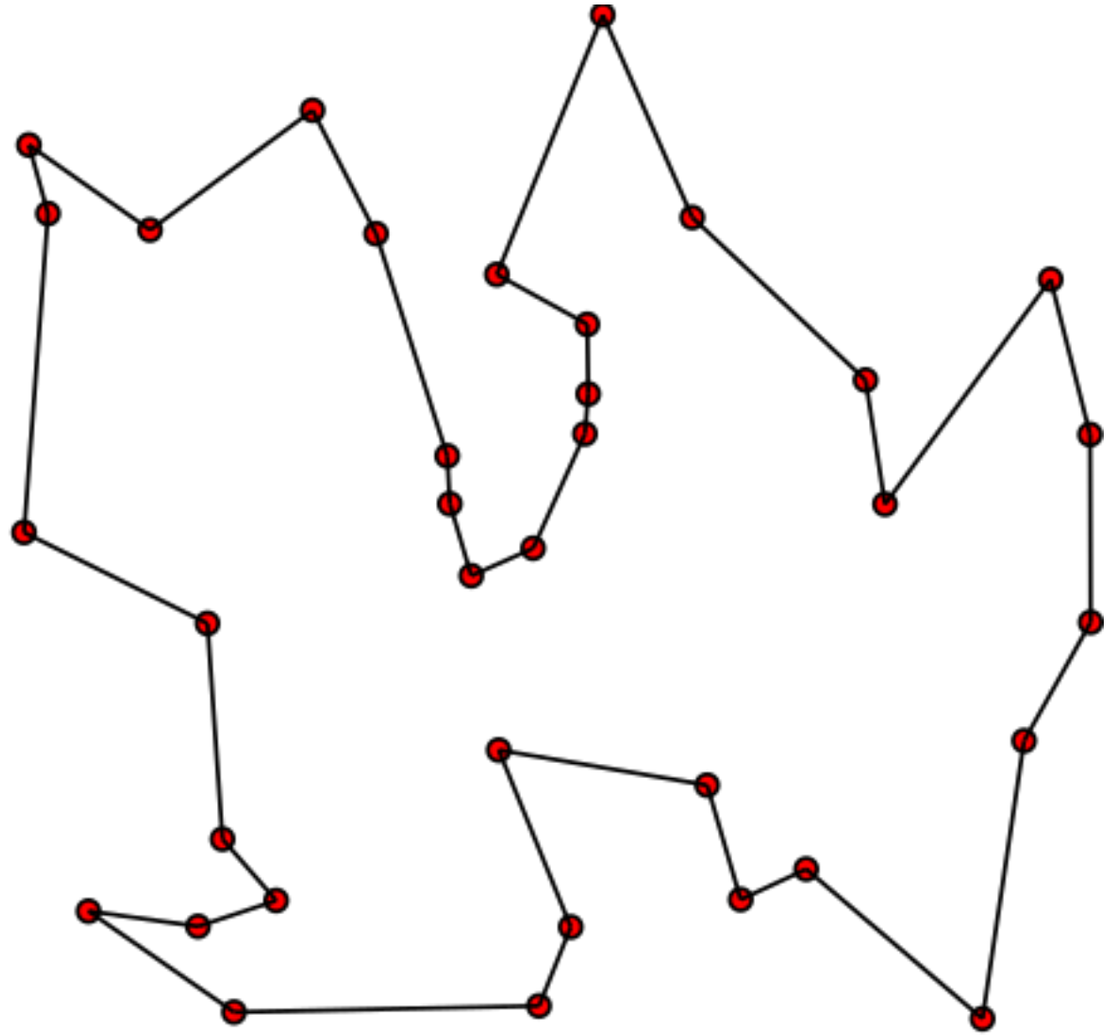


Traveling Salesperson Problem (TSP)

If you see a visualization missing costs,

- Either all the costs should be calculated by a natural distance metric, or the missing edges should be treated as having infinite cost.

- The cycle on the right is the TSP visiting all the red dots with the normal distances between them.



- Image source: Wikipedia

Traveling Salesperson Problem (1832?)

"Der Handlungsreisende – wie er sein soll und was er zu tun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein – von einem alten Commis-Voyageur" (The travelling salesman – how he must be and what he should do in order to get commissions and be sure of the happy success in his business – by an old *commis-voyageur*)

Source: Wikipedia

Traveling Salesperson for Circuit Layouts

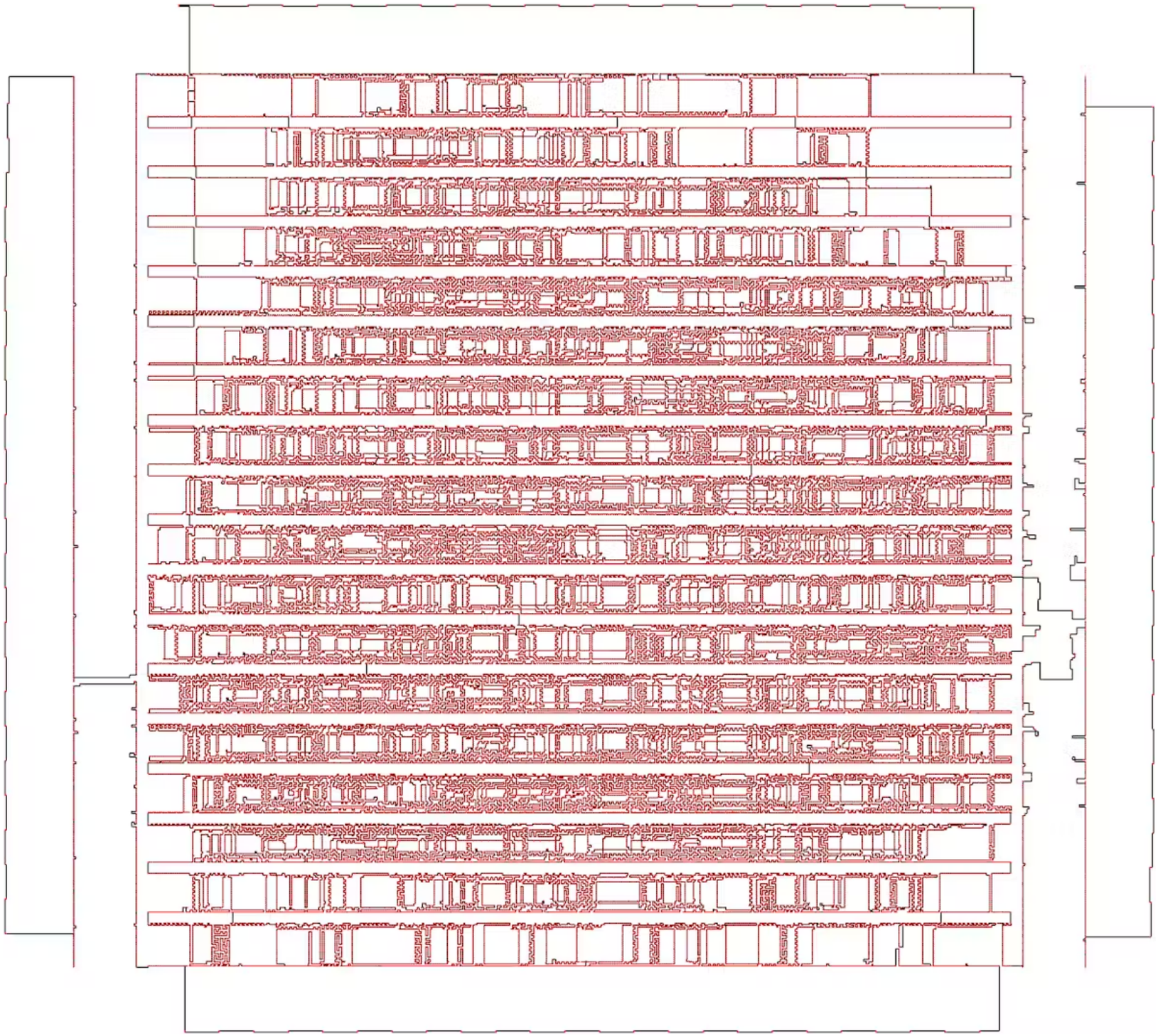


Image by William Cook et al

via https://theconversation.com/planning-the-best-route-with-multiple-destinations-is-hard-even-for-supercomputers-a-new-approach-breaks-a-barrier-thats-stood-for-nearly-half-a-century-148308

Traveling Salesperson is NP-Complete

$SATISFIABILITY \leq_P HAMILTONIAN\ CYCLE$ (Karp 1972)

Hamiltonian cycle problem: Given a graph, does there exist a cycle visiting each vertex exactly once?

How do we reduce this problem to TSP?

TopHat

TopHat

Which of the following are true when reducing the Hamiltonian Circuit problem on graph G_{HC} to the Traveling Salesperson problem with graph G_{TSP}?

- G_{TSP} must have more vertices than G_{HC}.

- One way to construct G_{TSP} is copying the vertices and setting the weight of an edge to one if it is present in G_{HC} and two otherwise.

- One way to construct G_{TSP} is copying the vertices and setting the weight of an edge to one if it is present in G_{HC} and n otherwise.

- G_{TSP} must have fewer vertices than G_{HC}.

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

Standard reduction:

- Given $G_{HC} = (V_{HC}, E_{HC})$,
  construct weighted graph $G_{TSP} = (V_{TSP}, E_{TSP}, W_{TSP})$
  such that
  - $V_{TSP} = V_{HC}$
  - $E_{TSP} = V_{HC} \times V_{HC}$
  - $W_{TSP}(e) = 1$ if $(e) \in E_{HC}$ and 2 otherwise

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

Standard reduction:

- Given $G_{HC} = (V_{HC}, E_{HC})$,
  construct weighted graph $G_{TSP} = (V_{TSP}, E_{TSP}, W_{TSP})$
  such that
  - $V_{TSP} = V_{HC}$
  - $E_{TSP} = V_{HC} \times V_{HC}$
  - $W_{TSP}(e) = 1$ if $(e) \in E_{HC}$ and 2 otherwise

- G_{HC} has a Hamiltonian cycle if and only if $OPT(G_{TSP}) = |V_{HC}|$

- Each edge that was in E_{HC} contributes 1 to the total cost, and each edge that was not in E_{HC} contributes 2 to the total cost.

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

With the standard reduction, we need to distinguish

- $OPT(G_{TSP}) = |V_{HC}|$ vs $OPT(G_{TSP}) = |V_{HC}| + 1$

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

With the standard reduction, we need to distinguish

- $OPT(G_{TSP}) = |V_{HC}|$ vs $OPT(G_{TSP}) = |V_{HC}| + 1$

In terms of approximation ratios, the approximation ratio needs to be better than

$$1 + \frac{1}{|V_{HC}|}$$

That is, a TSP approximation ratio lower than that solves Hamiltonian cycle exactly.

Does that sound likely?

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

With the standard reduction, we need to distinguish

- $OPT(G_{TSP}) = |V_{HC}|$ vs $OPT(G_{TSP}) = |V_{HC}| + 1$

In terms of approximation ratios, the approximation ratio needs to be better than

$$1 + \frac{1}{|V_{HC}|}$$

That is, a TSP approximation ratio lower than that solves Hamiltonian cycle exactly.

Does that sound likely?

Can we make that bound tougher?

Yes, if we make the “missing” edges more expensive.

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

Tweaked reduction:

- Given $G_{HC} = (V_{HC}, E_{HC})$,
  construct weighted graph $G_{TSP} = (V_{TSP}, E_{TSP}, W_{TSP})$
  such that
  - $V_{TSP} = V_{HC}$
  - $E_{TSP} = V_{HC} \times V_{HC}$
  - $W_{TSP}(u, v) = 1$ if $(u, v) \in E_{HC}$ and $|V_{HC}| + 1$ otherwise

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

Tweaked reduction:

- Given $G_{HC} = (V_{HC}, E_{HC})$,
  construct weighted graph $G_{TSP} = (V_{TSP}, E_{TSP}, W_{TSP})$
  such that
  - $V_{TSP} = V_{HC}$
  - $E_{TSP} = V_{HC} \times V_{HC}$
  - $W_{TSP}(u, v) = 1$ if $(u, v) \in E_{HC}$ and $|V_{HC}| + 1$ otherwise

- G_{HC} has a Hamiltonian cycle if and only if $OPT(G_{TSP}) = |V_{HC}|$

- If there is no Hamiltonian cycle, then $OPT(G_{TSP}) \geq 2|V_{HC}|$.

Hamiltonian Cycle $\leq_P$ Traveling Salesperson Problem

Tweaked reduction:

- Given $G_{HC} = (V_{HC}, E_{HC})$,
  construct weighted graph $G_{TSP} = (V_{TSP}, E_{TSP}, W_{TSP})$
  such that
  - $V_{TSP} = V_{HC}$
  - $E_{TSP} = V_{HC} \times V_{HC}$
  - $W_{TSP}(u, v) = 1$ if $(u, v) \in E_{HC}$ and $|V_{HC}| + 1$ otherwise

- G_{HC} has a Hamiltonian cycle if and only if $OPT(G_{TSP}) = |V_{HC}|$

- If there is no Hamiltonian cycle, then $OPT(G_{TSP}) \geq 2|V_{HC}|$.

- So an approximation ratio better than 2 for TSP solves Hamiltonian cycle exactly.
  - So no polynomial 2-approximation ratio for TSP unless P=NP.
  - Can prove for any constant by varying the “missing” edge weight.

General Traveling Salesman Problems Seem Hard

Deterministic exact solutions

- $O(n^2 2^n)$ algorithm using dynamic programming (1962)
- No known $O(1.9999^n)$ algorithms yet

Quantum algorithm

- Still takes $O(1.728^n)$ (2019)

Approximating the Traveling Salesperson Problem

Let’s try to approximate TSP anyway…

Approximating the Traveling Salesperson Problem

Let's try to approximate TSP anyway…

- Spoiler: sometimes we can approximate TSP, just not always.

Approximating the Traveling Salesperson Problem

Let's try to approximate TSP anyway…

- Spoiler: sometimes we can approximate TSP, just not always.

What can we say about any solution to the Traveling Salesperson problem?

Approximating the Traveling Salesperson Problem

Let's try to approximate TSP anyway…

- Spoiler: sometimes we can approximate TSP, just not always.

What can we say about any solution to the Traveling Salesperson problem?

- If we remove one edge from an optimal cycle (solution), we have a spanning tree.

Approximating the Traveling Salesperson Problem

Let's try to approximate TSP anyway…

- Spoiler: sometimes we can approximate TSP, just not always.

What can we say about any solution to the Traveling Salesperson problem?

- If we remove one edge from an optimal cycle (solution), we have a spanning tree.
- Can we bound spanning tree sizes?
- Yes! Minimum spanning trees are an easy problem.

$$MST(G_{TSP}) \leq OPT(G_{TSP})$$

Approximating the Traveling Salesperson Problem

Let's try to approximate TSP anyway…

- Spoiler: sometimes we can approximate TSP, just not always.

What can we say about any solution to the Traveling Salesperson problem?

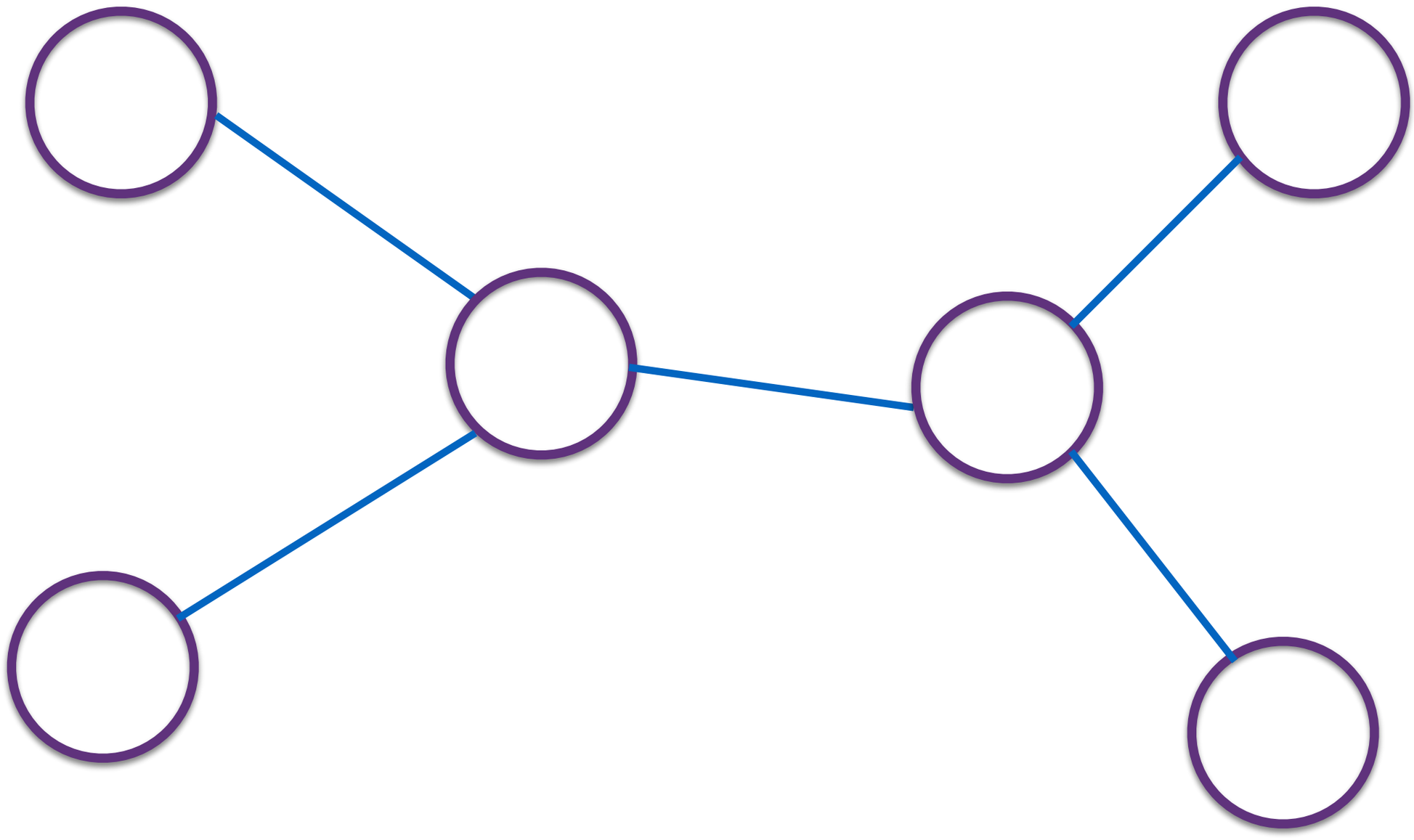
- If we remove one edge from an optimal cycle (solution), we have a spanning tree.
- Can we bound spanning tree sizes?
- Yes! Minimum spanning trees are an easy problem.

$$MST(G_{TSP}) \leq OPT(G_{TSP})$$

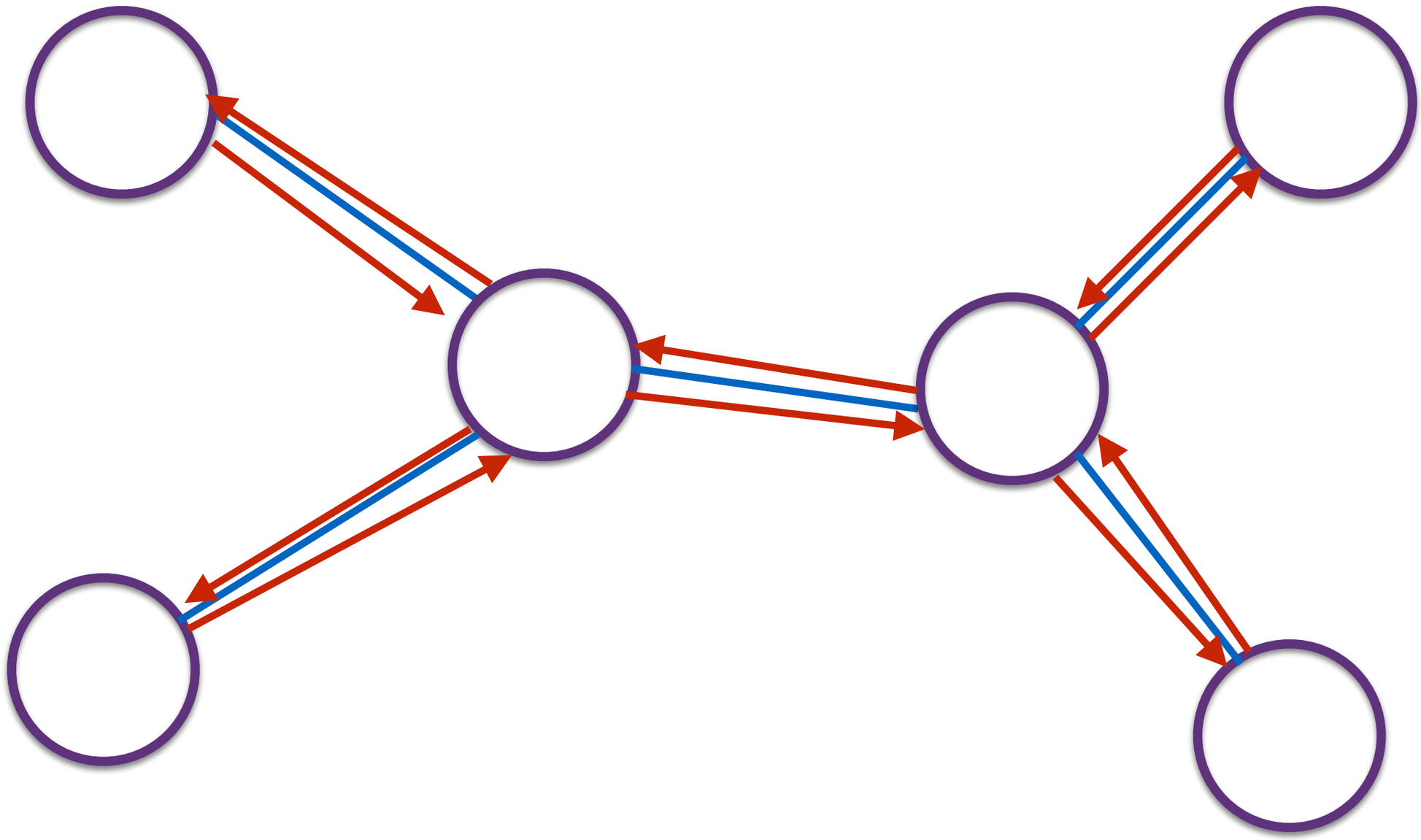
Can we upper bound value(TSP)?

- An upper bound close to $MST(G_{TSP})$ would be nice.
- Proposal: Construct a cycle based on the minimum spanning tree.

Minimum Spanning Tree Approximation of Traveling Salesperson

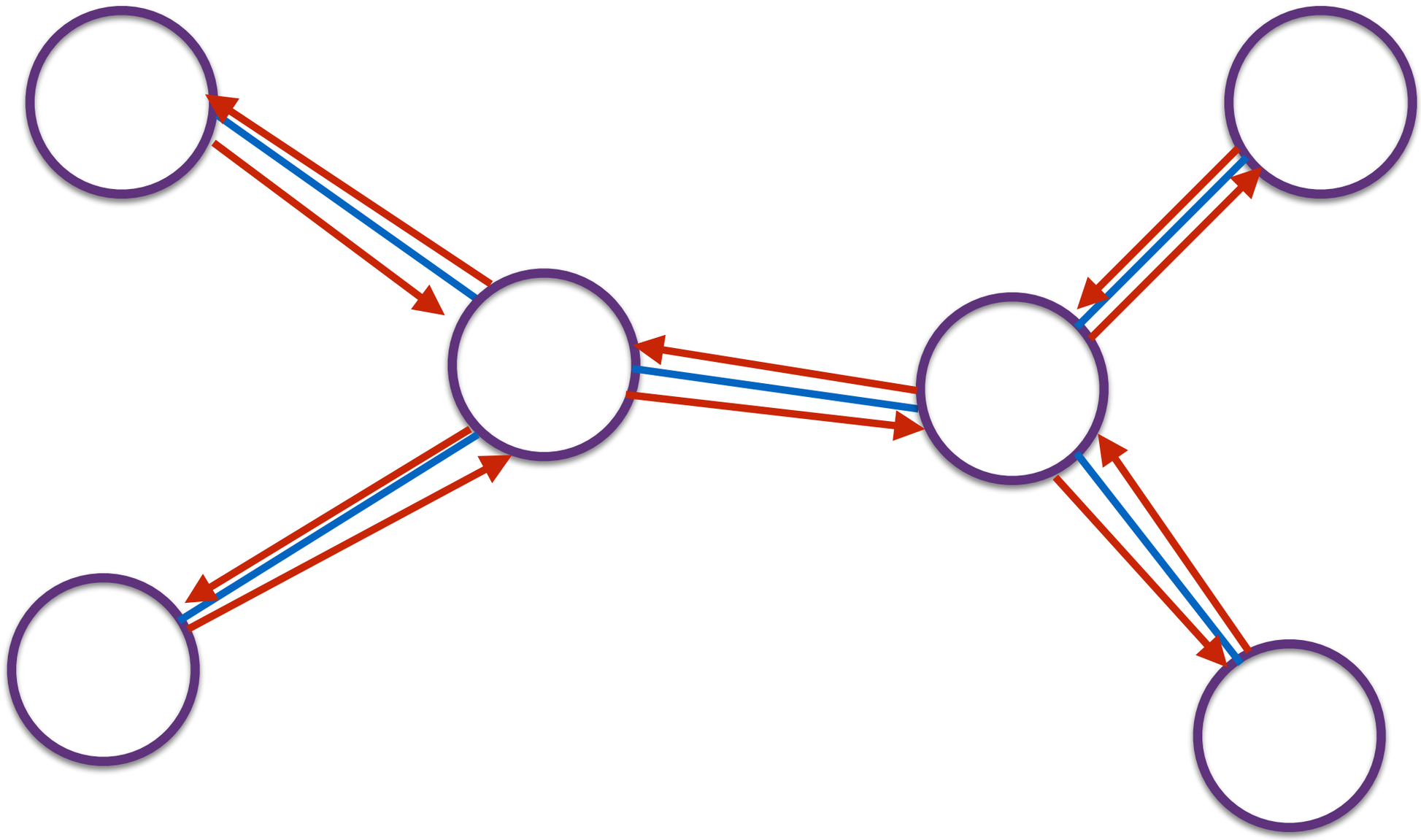


Minimum Spanning Tree Approximation of Traveling Salesperson



Replace undirected edge of the MST with a directed edge each way.

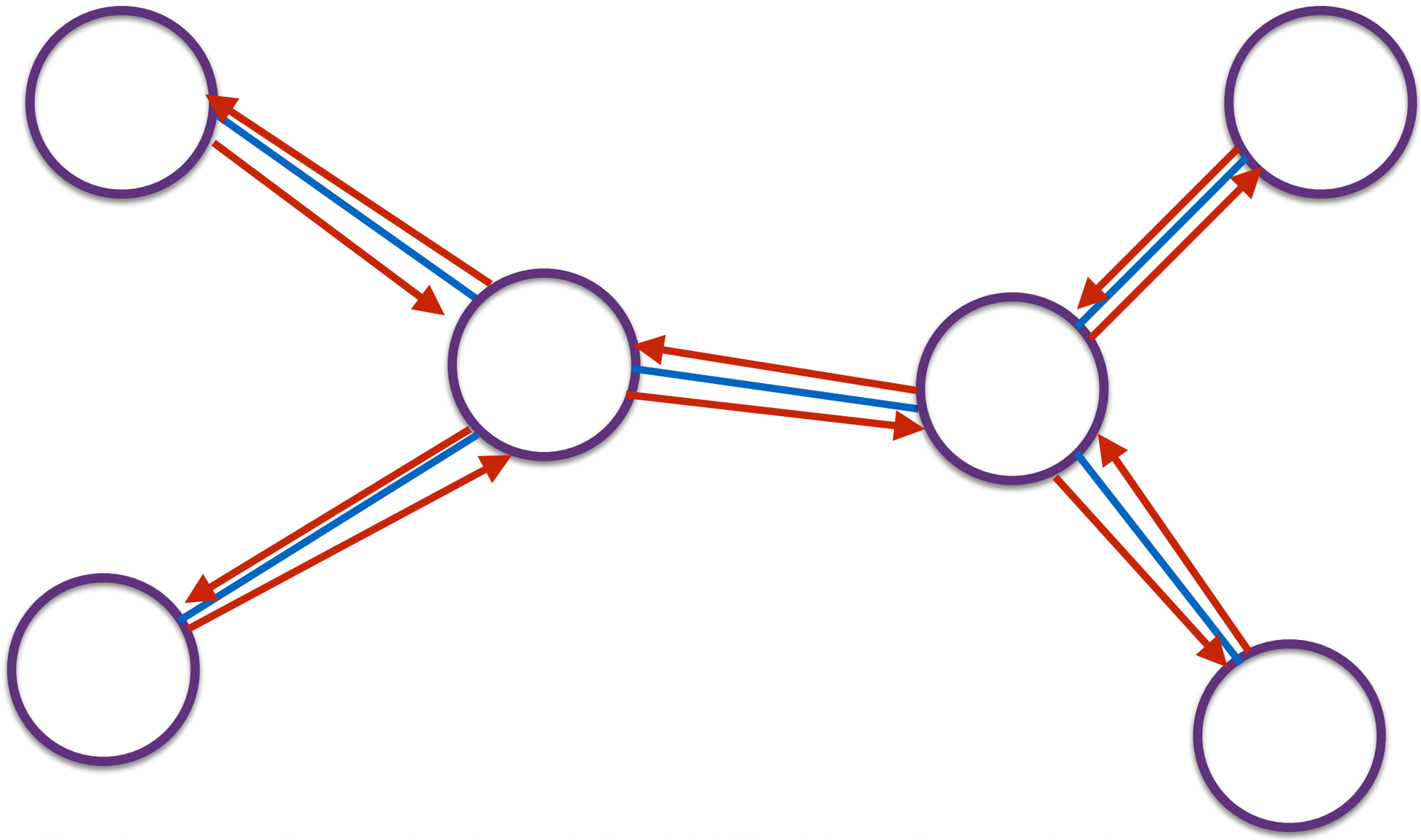
Minimum Spanning Tree Approximation of Traveling Salesperson



Replace undirected edge of the MST with a directed edge each way.

Can find an “Eulerian circuit” trivially since all vertex degrees are even.

Minimum Spanning Tree Approximation of Traveling Salesperson

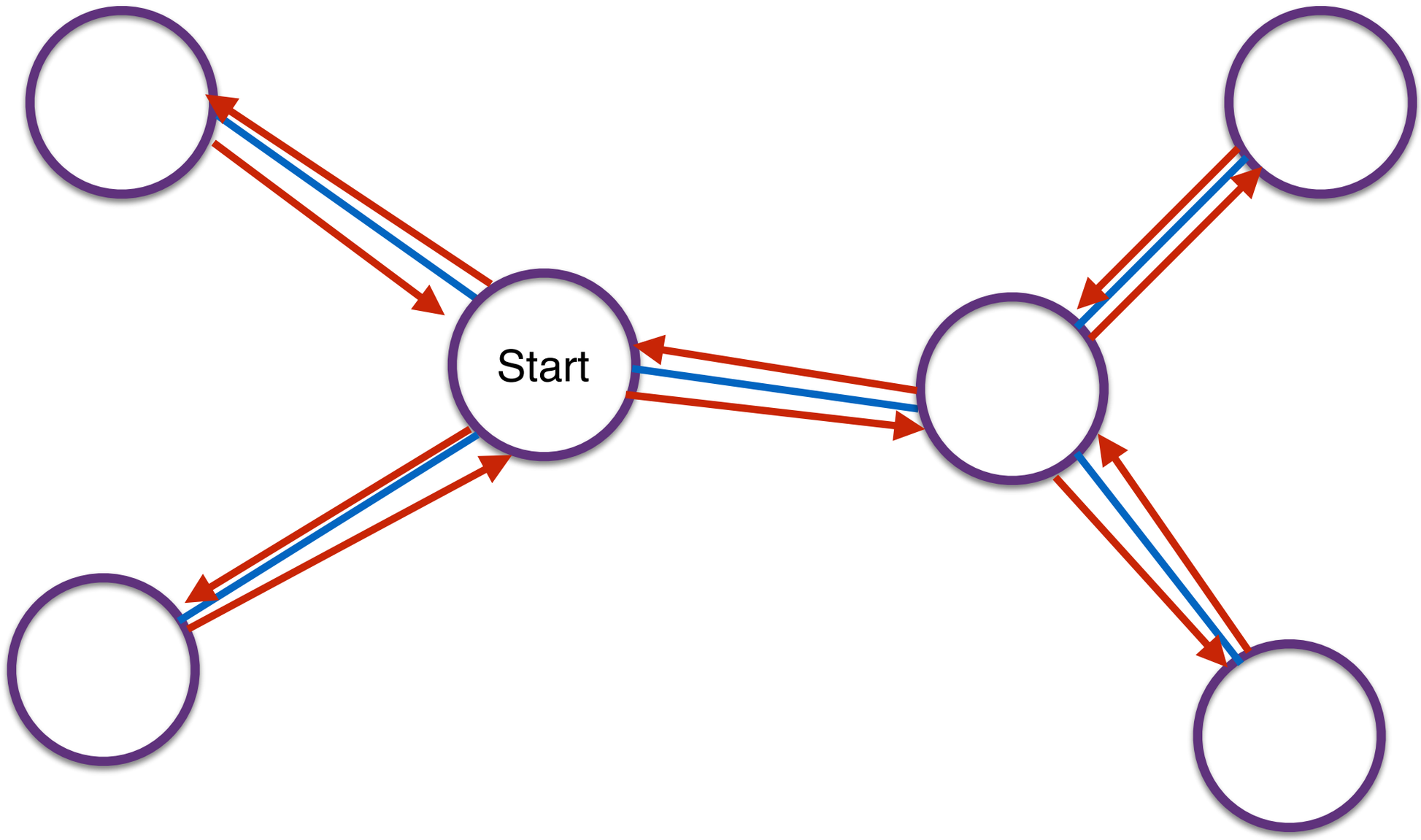


Replace undirected edge of the MST with a directed edge each way.

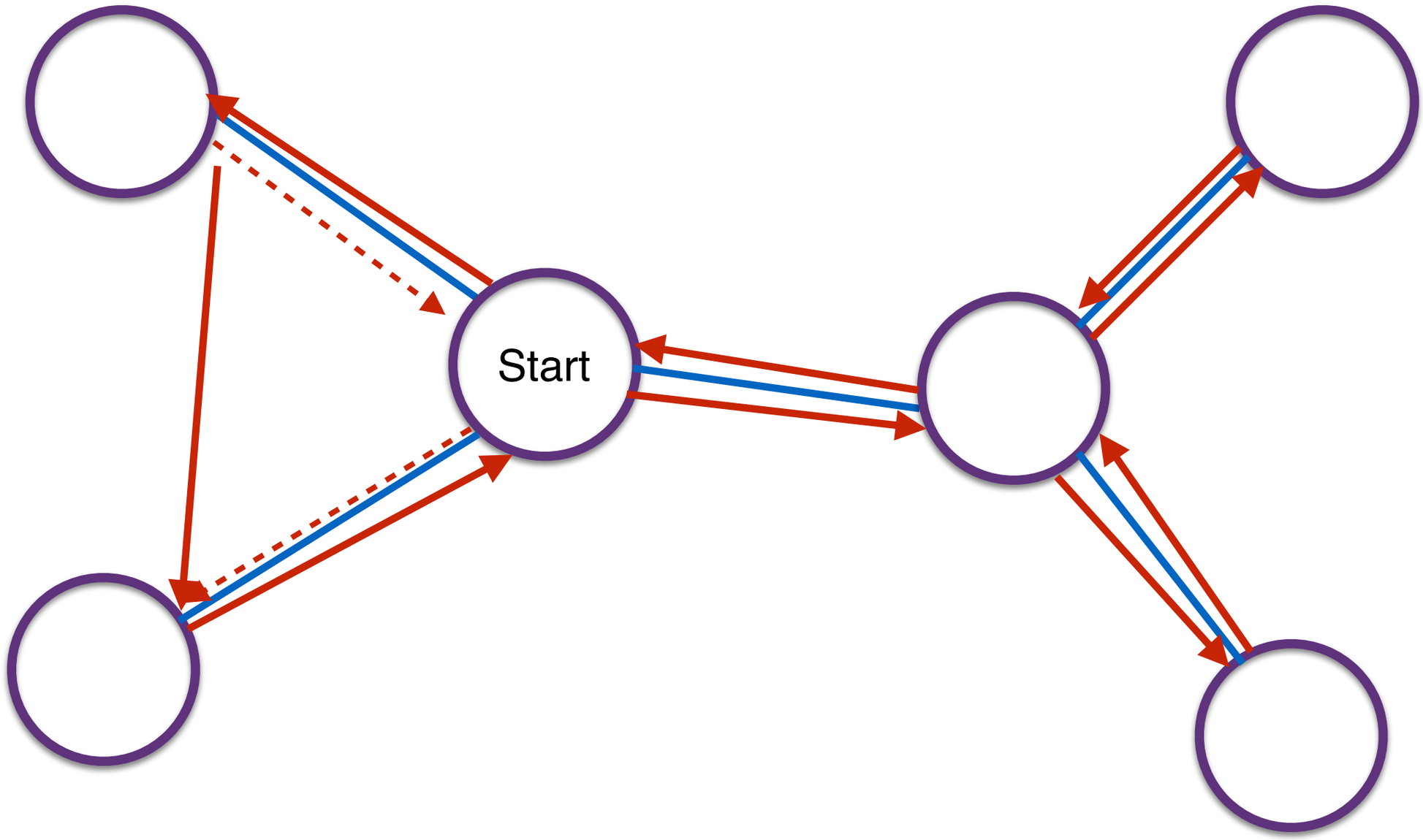
Can find an “Eulerian circuit” trivially since all vertex degrees are even.

But there will be duplicate vertices.

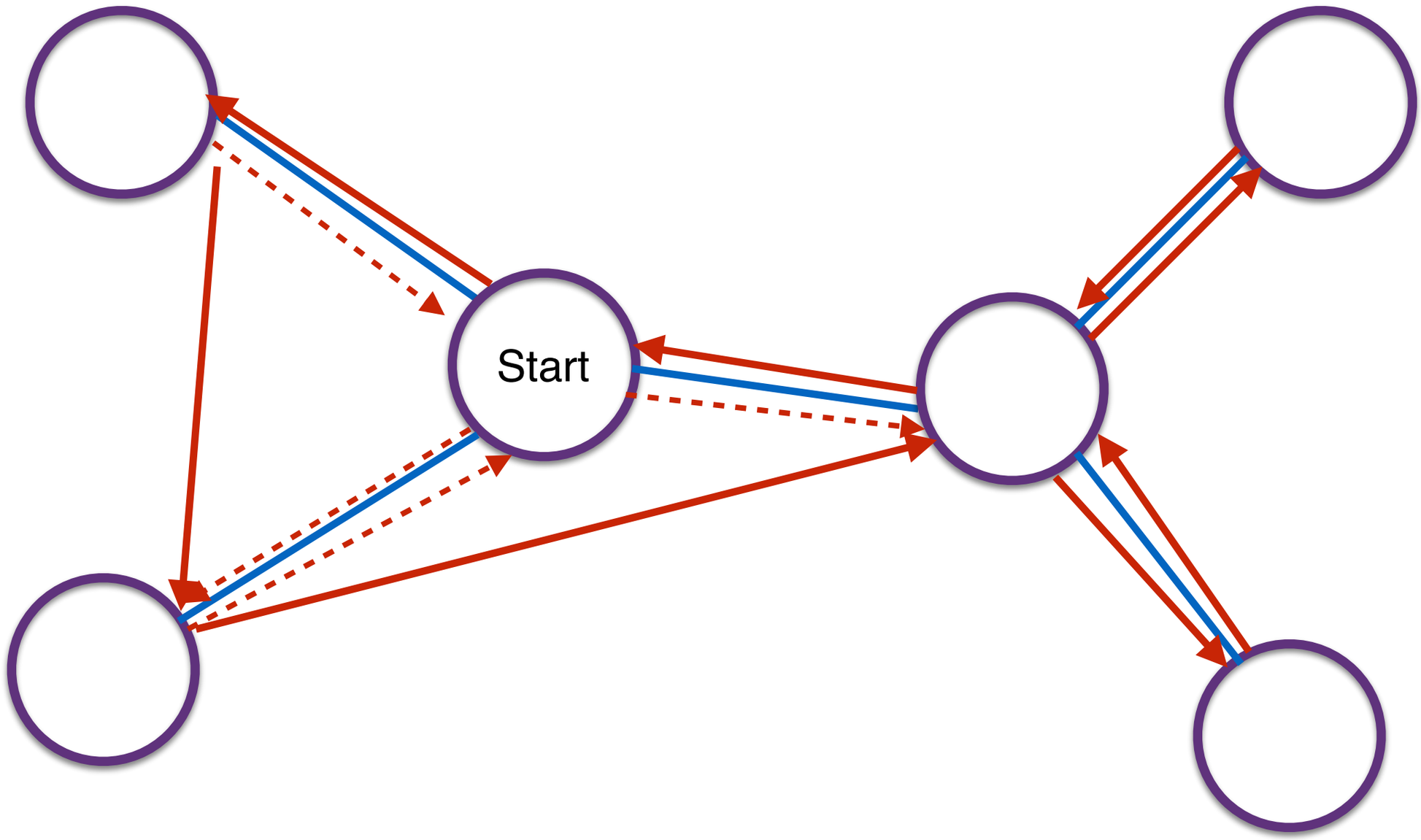
Minimum Spanning Tree Approximation of Traveling Salesperson



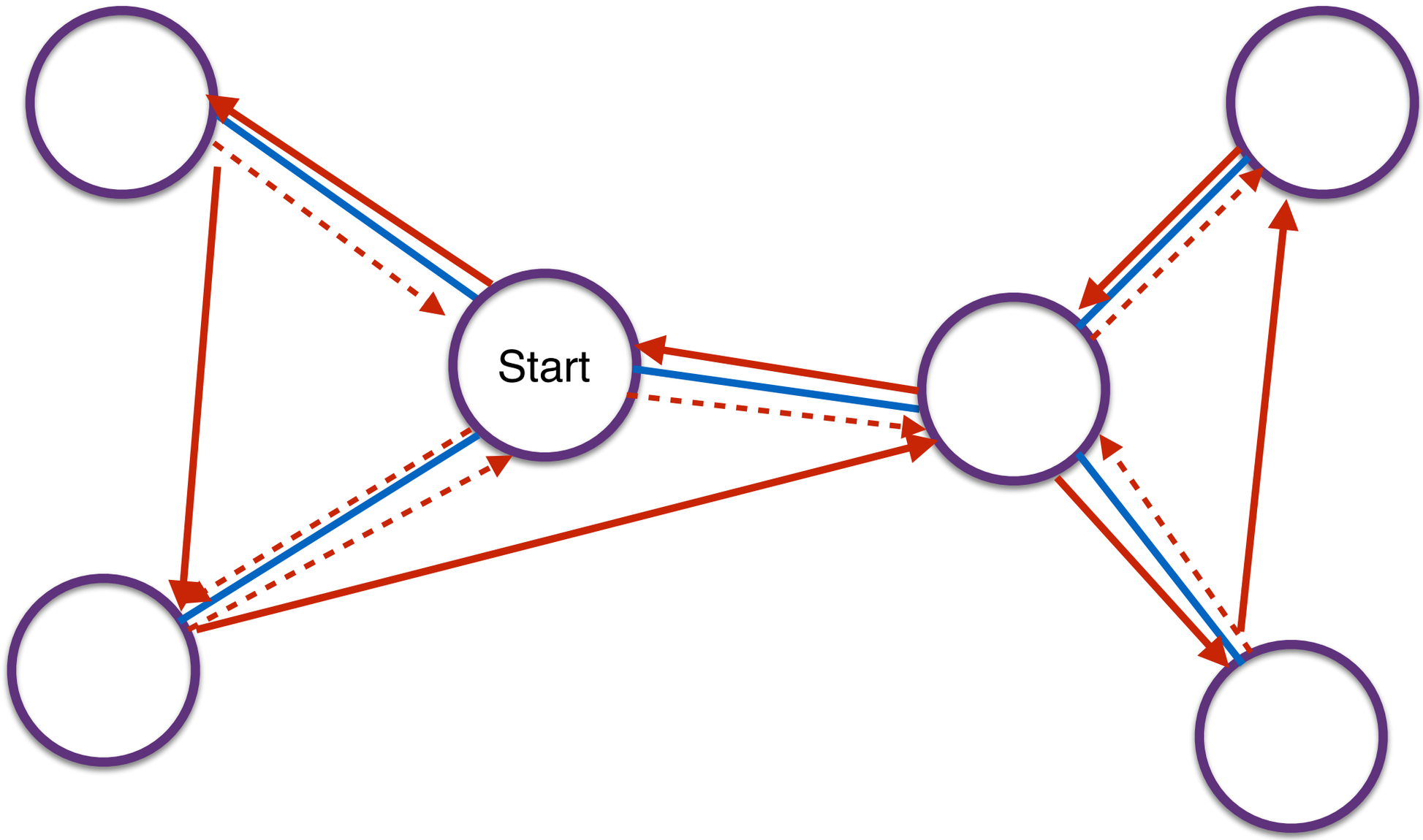
Minimum Spanning Tree Approximation of Traveling Salesperson



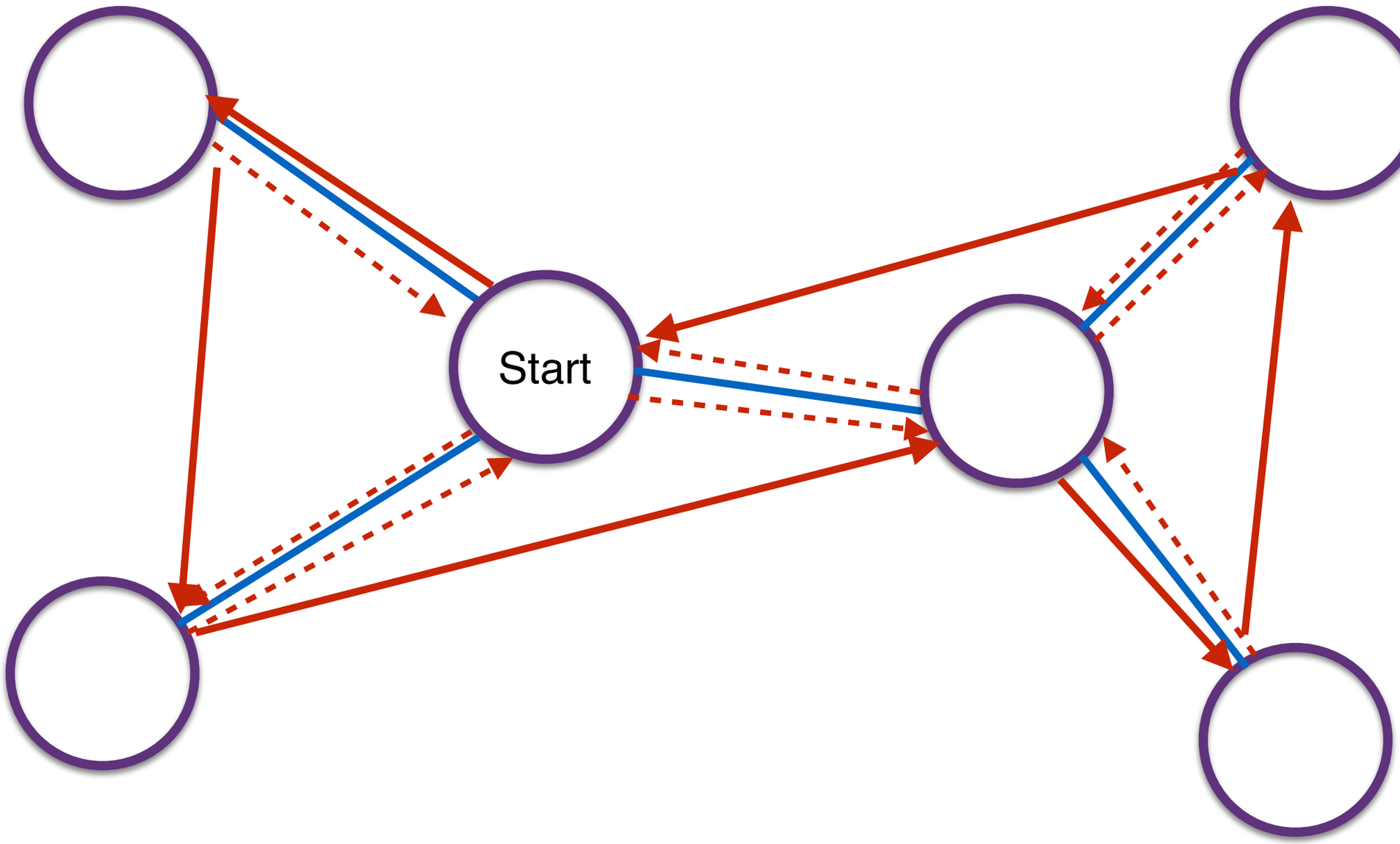
Minimum Spanning Tree Approximation of Traveling Salesperson



Minimum Spanning Tree Approximation of Traveling Salesperson



Minimum Spanning Tree Approximation of Traveling Salesperson



How Can We Bound Cost of Minimum Spanning Tree Approximation?

Cost before removing duplicates = $2 \cdot MST(G_{TSP})$

Already have $MST(G_{TSP}) \leq OPT(G_{TSP})$

How Can We Bound Cost of Minimum Spanning Tree Approximation?

Cost before removing duplicates = $2 \cdot MST(G_{TSP})$

Already have $MST(G_{TSP}) \leq OPT(G_{TSP})$

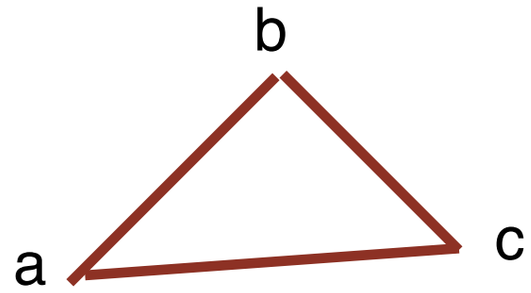
Tricky bit:

- When we remove duplicate nodes from the cycle, what happens to the total path cost?
  - Can it go up?
  - What will it take to bound that cost?

Triangle Inequality Bound

A distance function $d()$ obeys the triangle inequality if and only if for all a, b, c, $d(a,b) + d(b,c) \geq d(a,c)$.

- In other words, going direct is always shorter.



- A distance function obeying the triangle inequality is also called a metric.

Δ-Traveling Salesperson Problem

Given a graph where every pair of vertices is directly connected, each edge has a cost, and all edge costs follow the triangle inequality, what is the minimum total cost of a cycle visiting each vertex exactly once?

$$MST(G_{TSP}) \leq OPT(G_{TSP}) \leq MST\_TSP(G_TSP) \leq 2 \cdot MST(G_{TSP})$$

∴ MST-TSP is a 2-approximation to Δ-TSP.

Sanity Check

Does the 2-approximation of MST-TSP conflict with previous TSP approximation hardness?

Sanity Check

Does the 2-approximation of MST-TSP conflict with previous TSP approximation hardness?

- No, because the Hamiltonian path reduction does not follow triangle inequality.

How Hard is Δ-Traveling Salesperson?

The only change to the problem definition is that the triangle inequality is obeyed.

Should this make this harder or easier than normal TSP?

Δ-Traveling Salesperson is NP-Complete

TSP $\leq_P$ Δ-TSP

Let $w_{\max} = \max_{e \in E} W(e)$

Replace $W_{TSP}(e)$ with $W_{\Delta - TSP}(e) = W_{TSP}(e) + w_{\max}$

Δ-Traveling Salesperson is NP-Complete

TSP $\leq_P$ Δ-TSP

Let $w_{\max} = \max_{e \in E} W(e)$

Replace $W_{TSP}(e)$ with $W_{\Delta - TSP}(e) = W_{TSP}(e) + w_{\max}$

All cycle costs increase by $w_{\max} \mid V \mid$ so optimal cycles are unchanged.

But now the triangle inequality is satisfied.

Δ-Traveling Salesperson is NP-Complete

TSP $\leq_P$ Δ-TSP

Let $w_{\max} = \max_{e \in E} W(e)$

Replace $W_{TSP}(e)$ with $W_{\Delta - TSP}(e) = W_{TSP}(e) + w_{\max}$

All cycle costs increase by $w_{\max} \mid V \mid$ so optimal cycles are unchanged.

But now the triangle inequality is satisfied.

But this additive “distortion” keeps approximation ratio from applying to original TSP problem.

Further Δ-Traveling Salesperson Approximations (1976)

3/2 approximation Christofides-Serdyukov (1976)

Constructs cycle using MST + perfect matching of the MST's odd degree nodes…

Δ-Traveling Salesperson Approximation Lower Bounds (2015)

There are a few lower bounds on the approximation factors for Δ-TSP.

The most recent ones are 123/122 (symmetric distances) and 75/74 (asymmetric distances…

https://arxiv.org/abs/1303.6437

Further Δ-Traveling Salesperson Approximations (2021)

A (Slightly) Improved Approximation Algorithm for Metric TSP

Anna R. Karlin, Nathan Klein, Shayan Oveis Gharan

For some $\epsilon > 10^{-36}$ we give a randomized $3/2 - \epsilon$ approximation algorithm for metric TSP.

https://arxiv.org/abs/2007.01409

1.499999999999999999999999999999999-approximation

(you don't need to know it either)

Euclidean Traveling Salesman Problem

Last TSP variant for today…

- Also NP-hard.
- NP status is complicated by irrational numbers…
- NP-complete if you scale and round to integers.

But it actually has a fully polynomial time approximation scheme.

For d dimensions and any c,

$O\left(n(\log n)^{O(c\sqrt{d})^{d-1}}\right)$ algorithm to get (1+1/c)-approximate solution

Technically a polynomial for fixed d and c. But a pretty big one.

TSP Summary

All discussed TSP variants are NP-hard

General case:

- Cannot approximate within a constant factor or P=NP

Metric / Δ case:

- Easy 2-approximation based on minimum spanning trees
- Somewhat harder 1.5-approximation

Euclidean distance case:

- Fully polynomial time approximation scheme
- But often big polynomials

Looking for Intuition re: Approximation Algorithms?

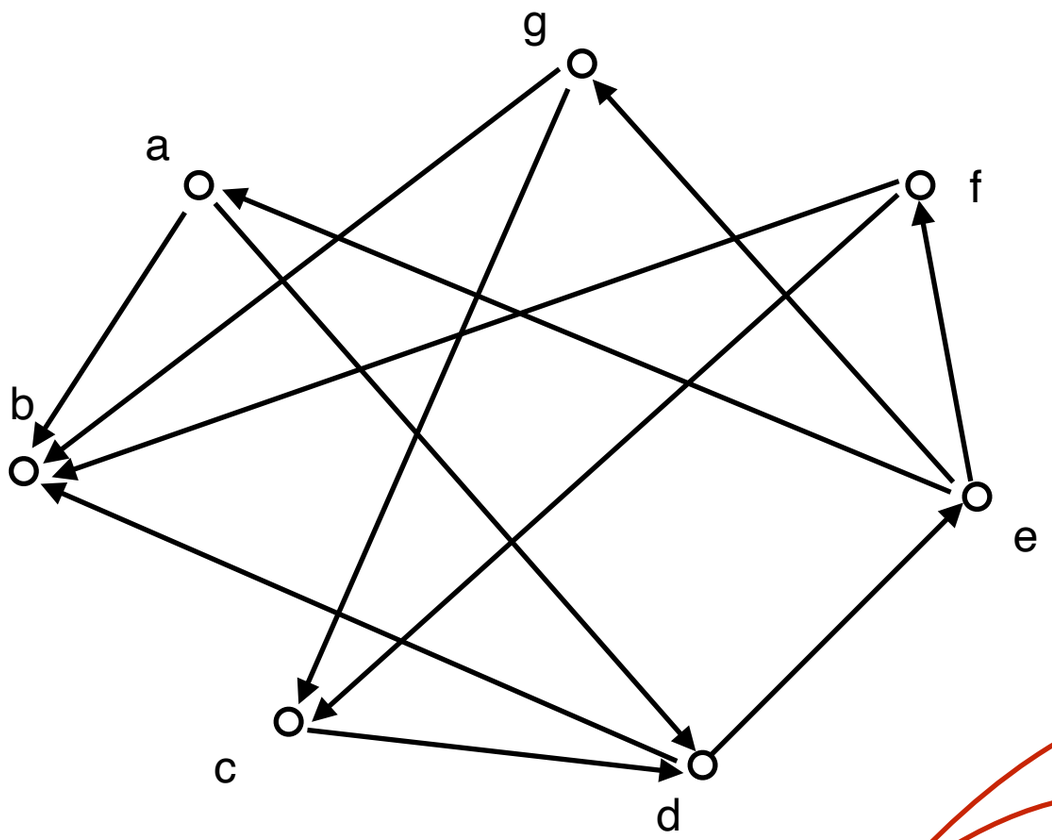
Where do the algorithms come from?

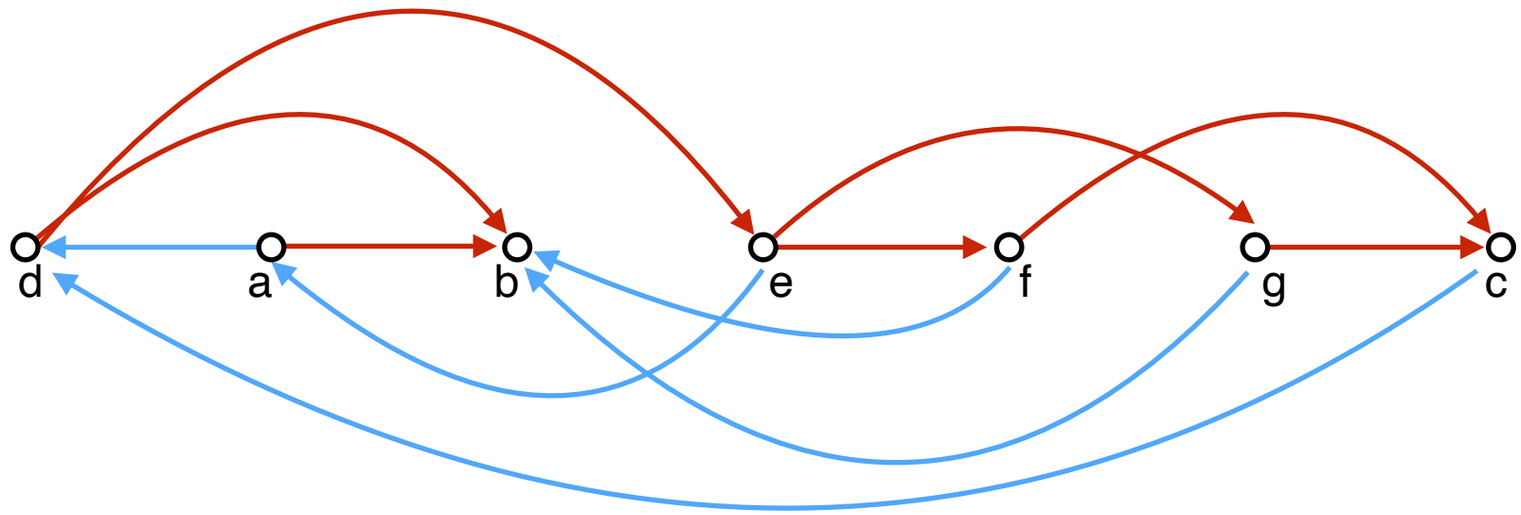
Where do the factors come from?

No general answer, but some recurring themes -

- Bound optimal solution quality?
- Bound approximate solution quality?
- Linking structure of optimal answer to structure of approximate answer.

Acyclic Subgraph





Given graph G, find it's largest acyclic subgraph:

delete some of its edges, such that the remaining graph doesn't contain any directed cycles and has the maximum number of edges.

Vertex Cover 2x-optimal greedy algorithm

Vertex Cover: Given a graph G(V,E) find the smallest subset of vertices S, such that it forms a vertex cover. That is, every edge (u,v) **has at least one of its nodes** in S.

Algorithm 1: GreedyVC($G(V, E)$)

```
1 S ← empty set of vertices;
2 for (u, v) is an edge do
3 |   if u ∉ S AND v ∉ S then
4 |   |   S ← S ∪ {u, v};
5 return S;
```

Clue: At least one of its nodes in S.

This is not a one node per edge lower bound because the edges can have overlapping nodes.

What if we pick a set of edges E' with non-overlapping nodes? Then lower bound is $|E'|$ nodes. And this algorithm picks $2|E'|$ nodes.

Greedy approximation algorithm for Independent Set

Independent Set: Given a graph G(V,E), an independent set is a subset of its vertices S*, such that for each edge (u,v) at most one of u or v is in S*

We do not know S. But we can reason about its relationship with V—S because nodes added to S block other nodes from being added.

Analysis of Any Maximal Independent Set

Goal: find a lower bound on ISI

- a node u is in V—S (thus not in S) because it has a neighbor v in s
- Each v in S has at most D neighbors
- we get $|V - S| \leq D \cdot |S|$
- Adding up the two we get $|V| = |V - S| + |S| \leq D \cdot |S| + |S| = (D + 1)|S|$
- in conclusion $|OPT| \leq |V| \leq (D + 1)|S|$

Set Cover greedy algorithm

Set Cover: Given a universe U of items $i_1, i_2, \ldots, i_n$ and subsets of items $S_1, S_2, \ldots, S_m$, select a minimum number of the subsets so that their union contains every item in U.

Greedy algorithm:

In each iteration select the set that covers the most additional items.

Algorithm 1: GreedySC$(U, S_1, \ldots S_m)$

```
1 X ← U /* uncovered elements in U                    */
2 C ← empty set of subsets;
3 while X is not empty do
4 |   Select S_i that covers the most items in X;
5 |   C ← C ∪ S_i;
6 |   X ← X \ S_i;
7 return C;
```

Pretty standard greedy heuristic. Even has proven bounds for submodular functions (but those are maximizing coverage vs minimizing covering set).

Set Cover Analysis

This is the trickiest example we've covered?

Hard part is reasoning how much progress we make with each greedy choice.

Key insight:

If I tell you that the minimum set cover has size k and there are n' items left, then you know you can eliminate n'/k of the items with at least one choice.

Why? Because there exist k sets that cover everything.

At any point, if you started picking those k sets, you could finish with those k picks.

So those k picks must total n' now.

And the biggest one must be at least n'/k.

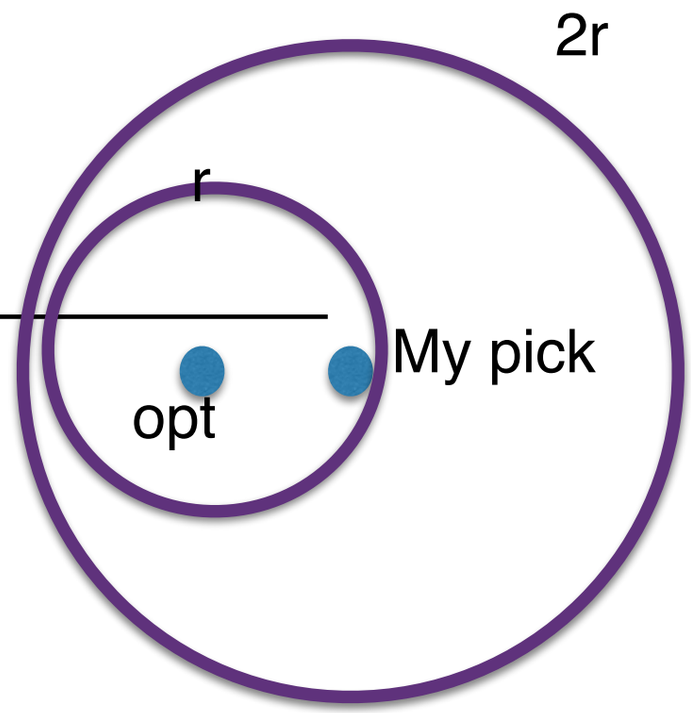
2-approximation greedy algorithm

Let C^* be the solution to the k-center problem, suppose $r(C^*)=r$.

select the k centers among the sites $s_1,\ldots,s_n$

Algorithm 1: kCenterWithR$(k, s_1, s_2, \ldots, s_n, r)$

```
   /* s_i are the sites, r is the optimal radius                    */
 1 S ← s_1, ... s_n /* uncovered sites                               */
 2 C ← ∅ /* centers                                                 */
 3 for j = 1 to k do
 4 |   s ← random element from S;
 5 |   C ← C ∪ {s};
 6 |   remove sites from S with distance ≤ 2r from s;
 7 return C
```



Next Fit Bin Packing

```
Start with one empty bin and call it "current".
For each item:
    If there is room in the current bin,
        Put the item in the current bin.
    Otherwise,
        Start a new bin and make it current.
        Put the item in the new current bin.
```

Lower bound before algorithm:

$$\left\lceil \sum s_i \right\rceil$$

If we started a new bin, the "there is room" check failed. So the total between the old bin and new bin is more than one.

This turns into pairs of bins having too much for one bin.

Turns into 2-approx

Open Questions aka Unofficial Midterm Review

What still seems hard re: approximation algorithms?

- Approximation ratios?
- Finding connections between optimum solutions and approximations?

Other course topics so far?