Improved Approximations for Graph-TSP in Regular Graphs

R Ravi

Carnegie Mellon University

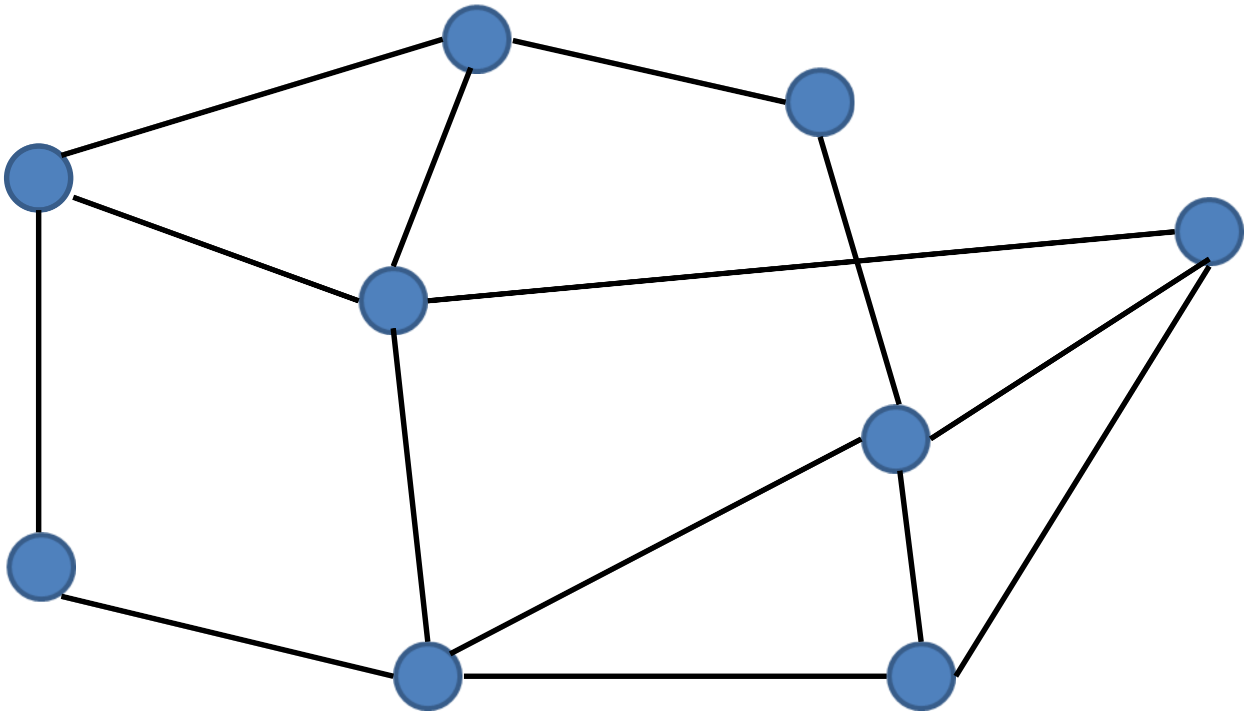
Joint work with Uriel Feige (Weizmann), Satoru Iwata (U Tokyo), Jeremy Karp (CMU), Alantha Newman (G-SCOP) and Mohit Singh (MSR)

Graph TSP

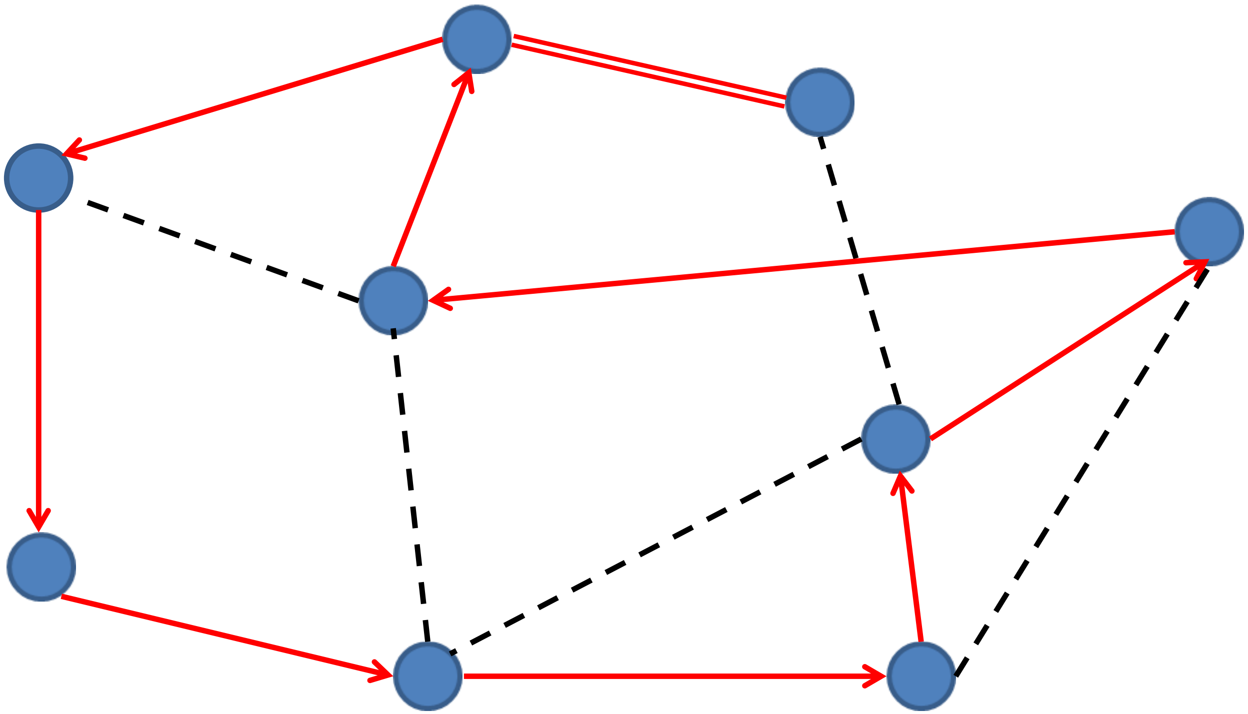
Given a connected unweighted graph, a tour is a closed walk that visits every vertex at least once.

Objective: find shortest tour.

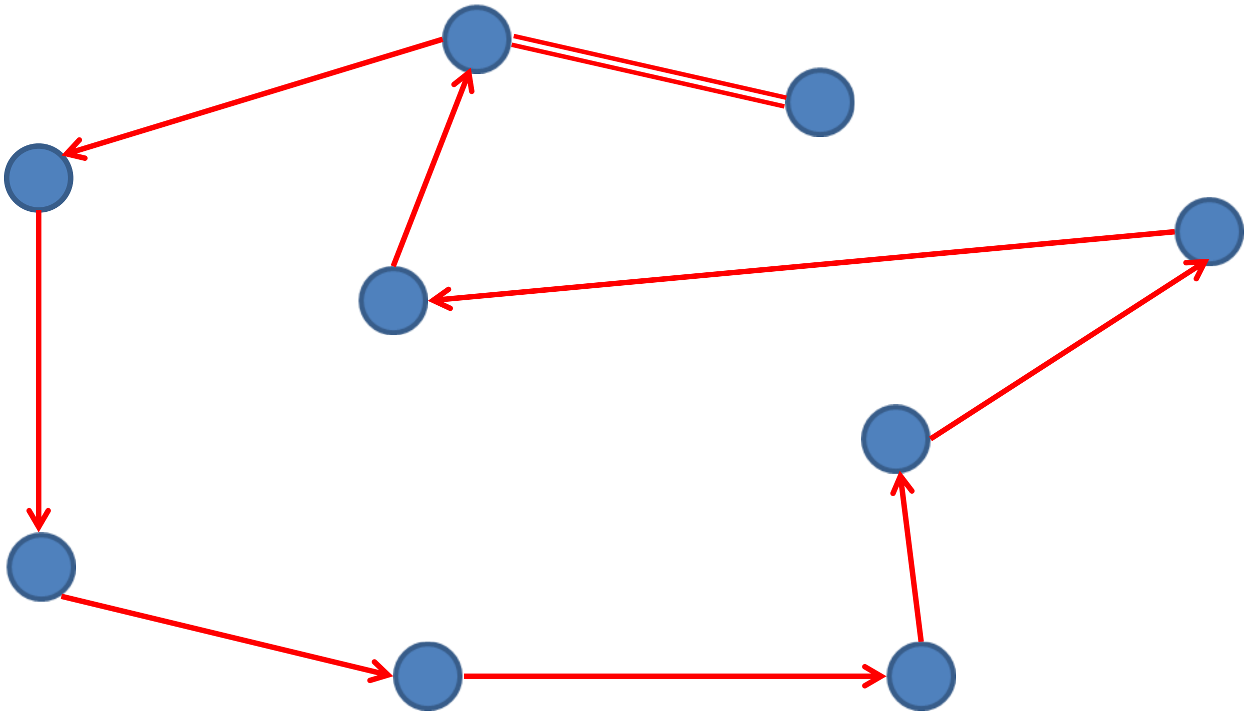
Ideally – a Hamiltonian cycle.



A tour may use the same edge twice



A tour may use the same edge twice
length 8x1 + 1x2 = 10



Results

1. Size 4n/3 in a graph with a spanning tree and a simple cycle on its odd nodes

(WG14, joint with Satoru Iwata and Alantha Newman)

"Regular graphs have short tours"

2. Size 9n/7 in cubic bipartite graphs

(APPROX14, joint with Jeremy Karp)

3. Size $(1 + O(\frac{1}{\sqrt{d}}))n$ in d-regular graphs

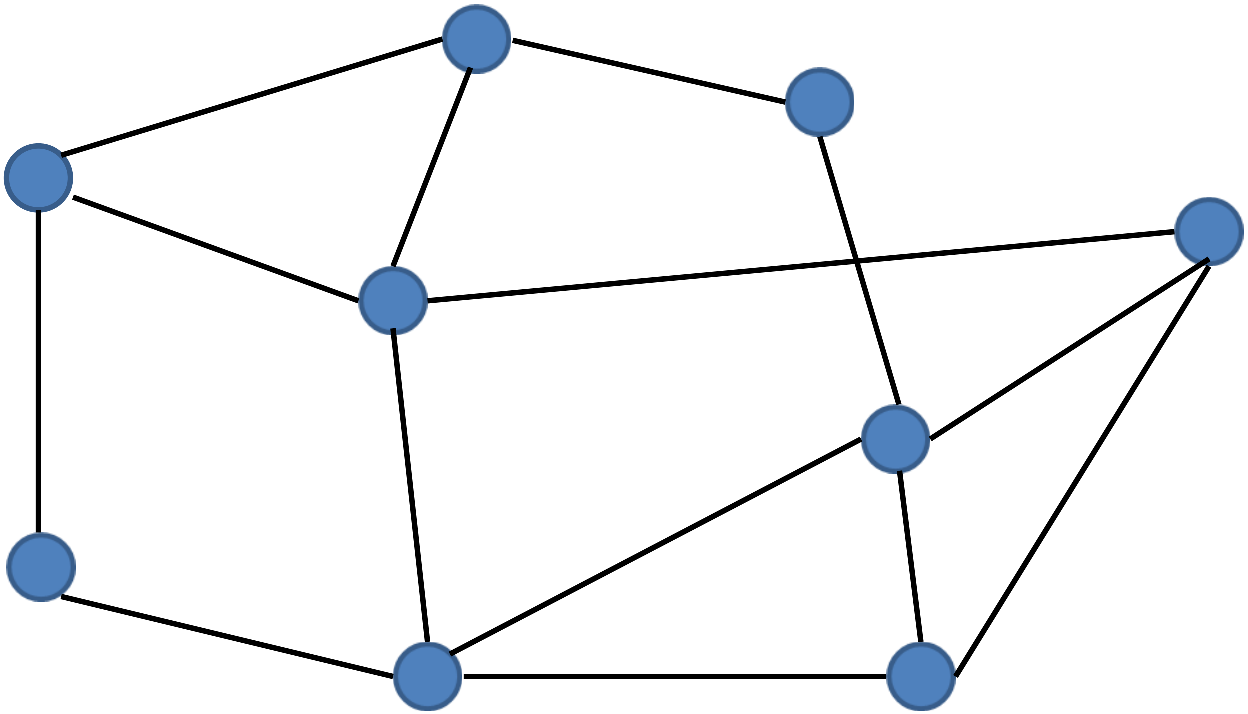
(IPCO14, joint with Uri Feige and Mohit Singh)

Main ideas for short tours

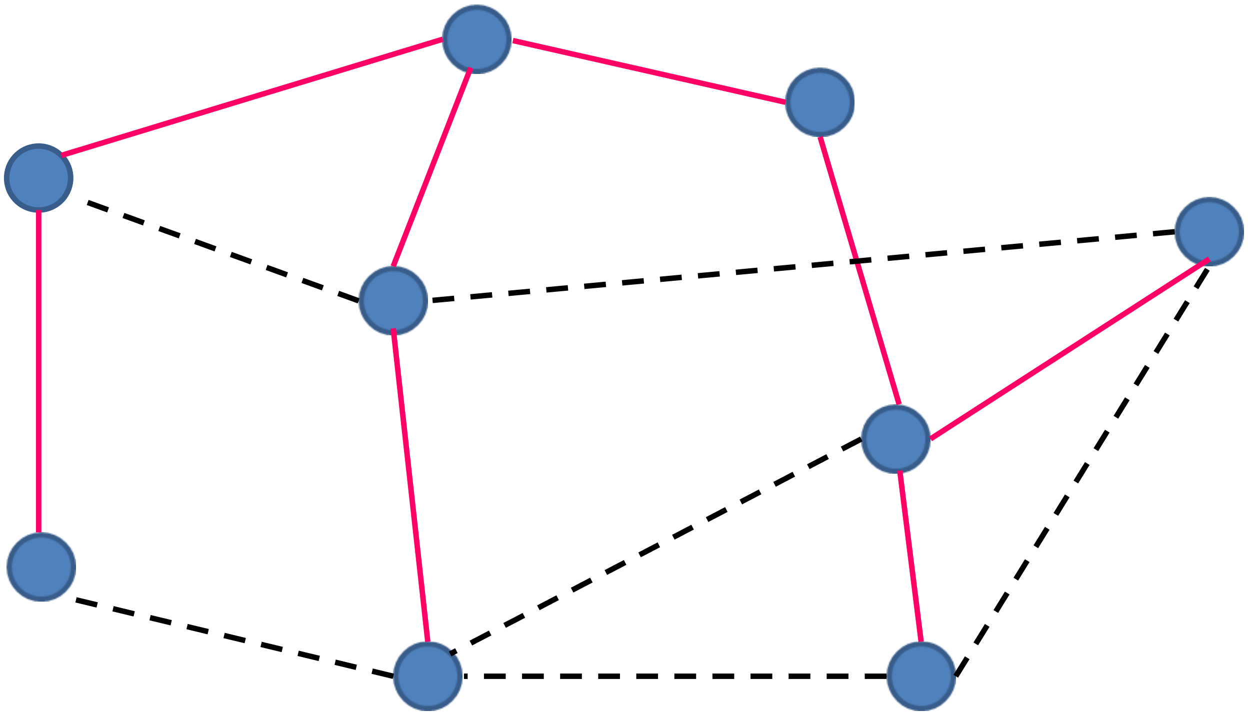
1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

General bounds

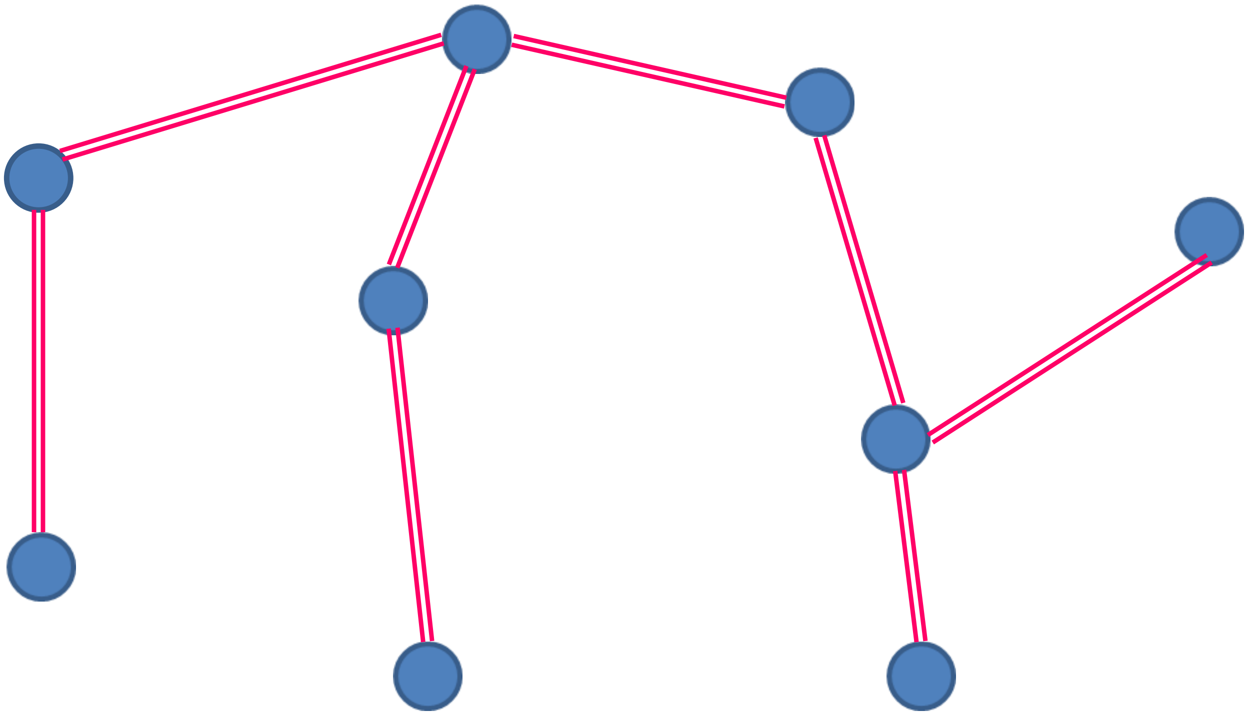
In every connected n-vertex graph, the length of the shortest tour is between n and 2n-2.



Spanning tree lower bound



Double spanning tree edges, drop remaining edges



Christofides 1976

- A 3/2-approximation to graph TSP (and more generally, metric TSP).

Tour composed of union of:

- (minimum) Spanning tree.
- Minimum T-join on odd-degree vertices.

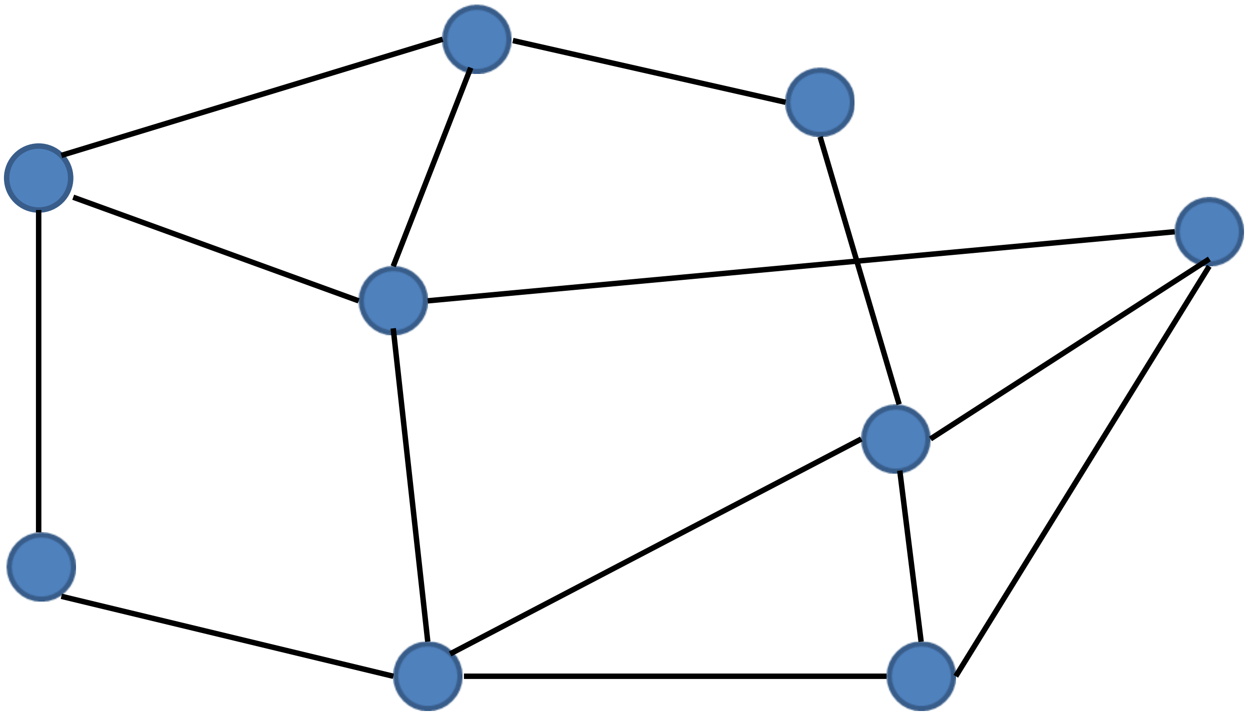
Gives a connected Eulerian graph (= tour).

Recall definitions

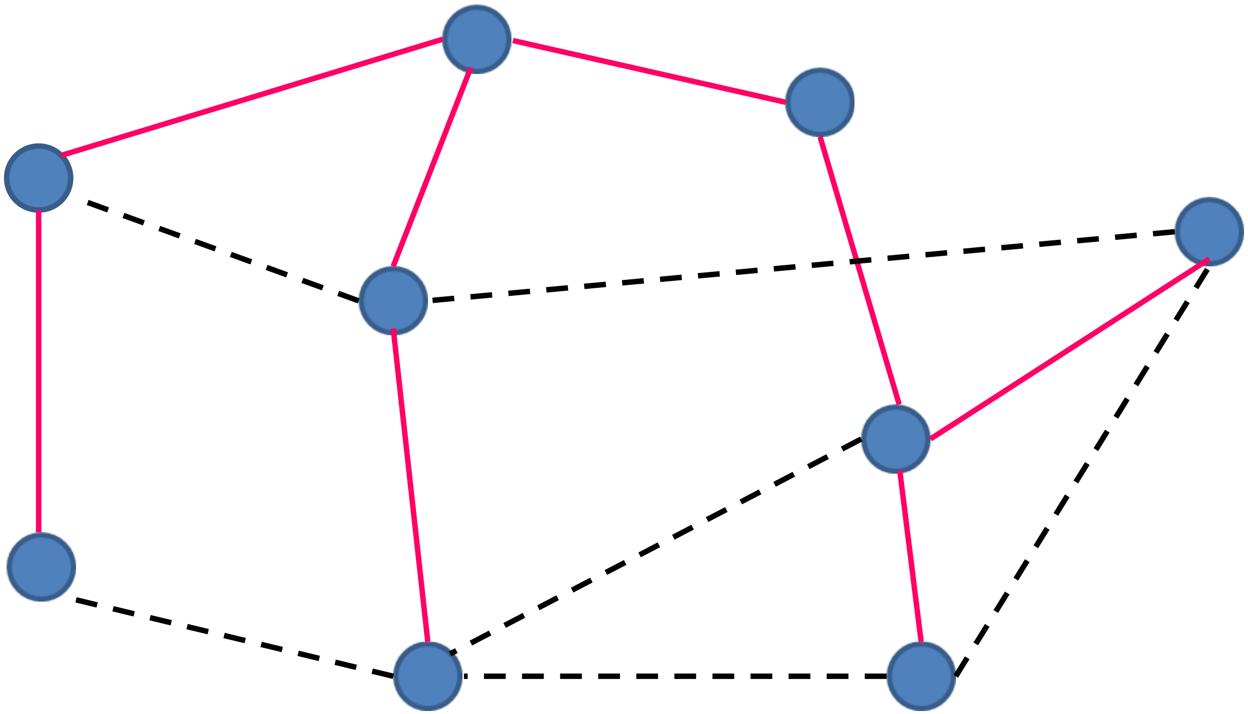
Let *T* be a subset of the vertex set of a graph. An edge set is called a ***T*-join** if in the induced subgraph of this edge set, the collection of all the odd-degree vertices is *T*.

A graph is Eulerian if all degrees are even. A connected Eulerian (multi)-graph has an Eulerian circuit: a walk that uses every edge exactly once.

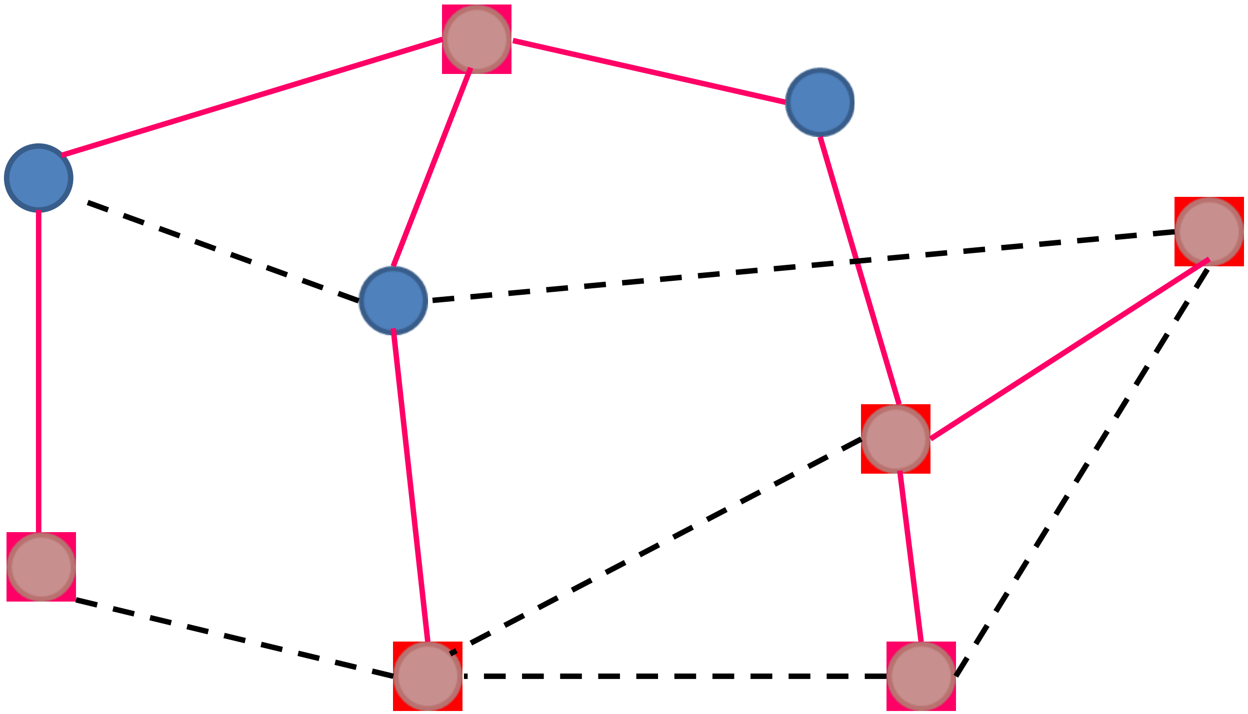
Christofides for graph TSP



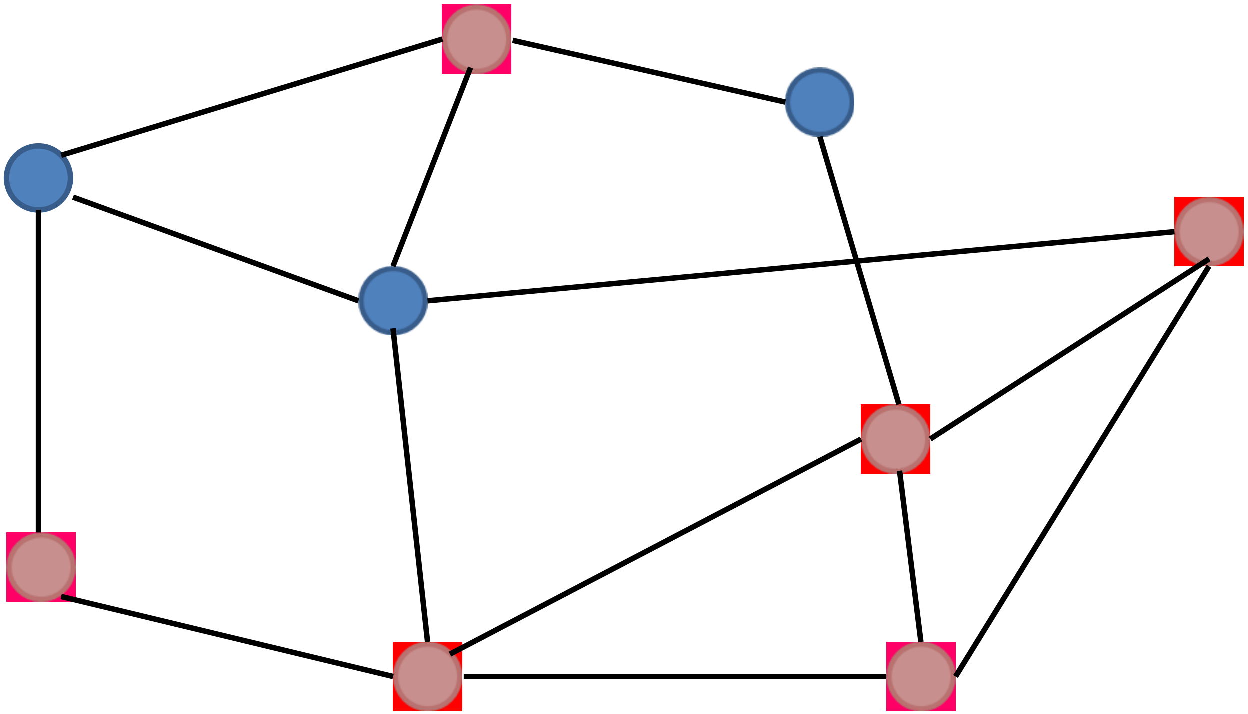
Arbitrary spanning tree



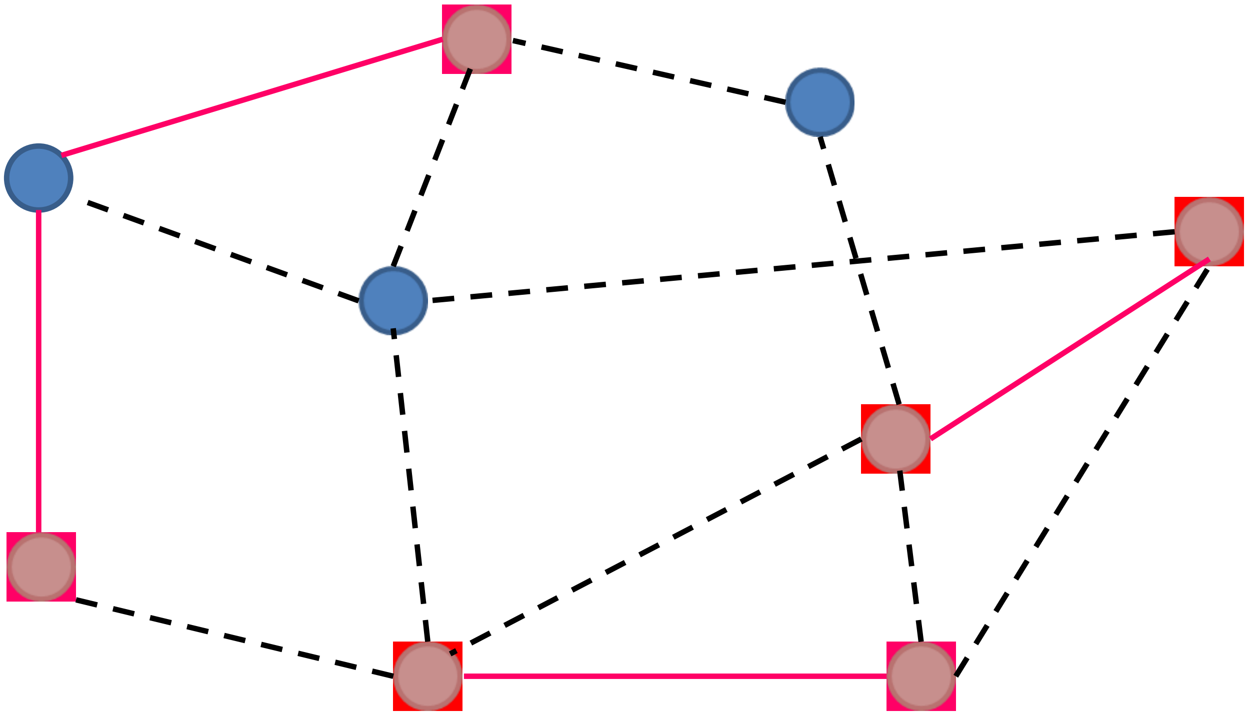
Odd degree vertices



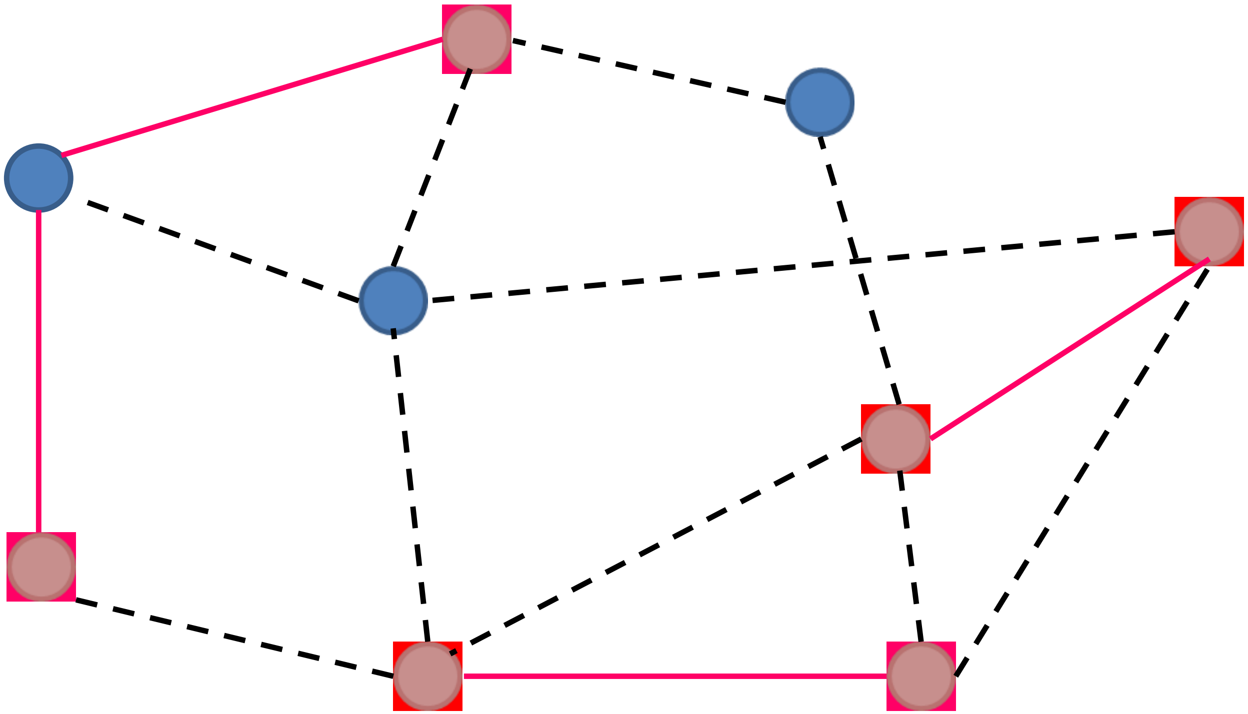
Odd degree vertices



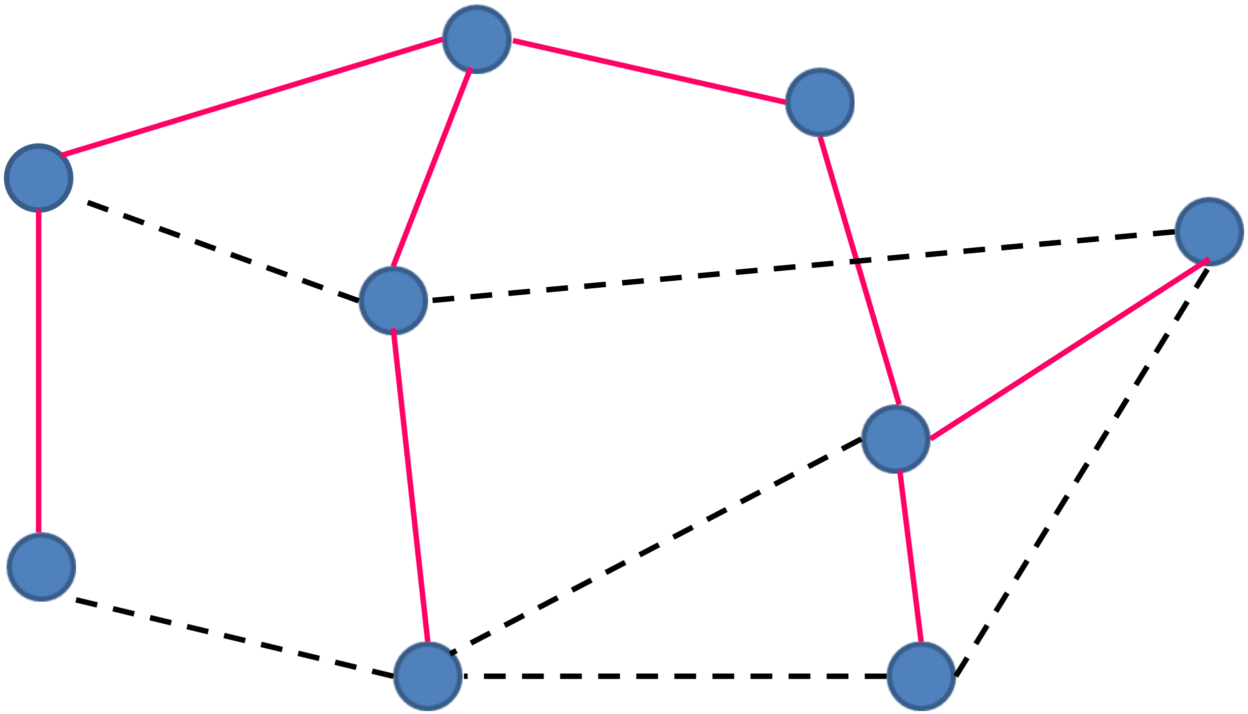
Find minimum T-join



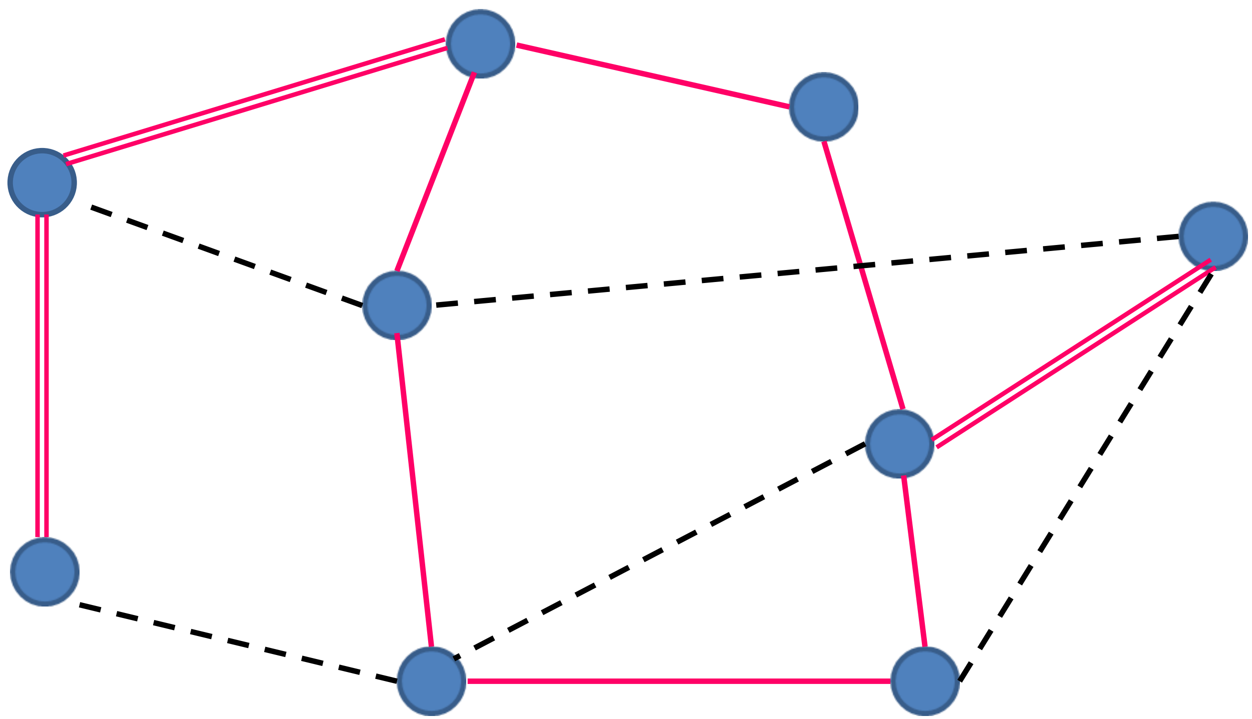
Union of the T-join…



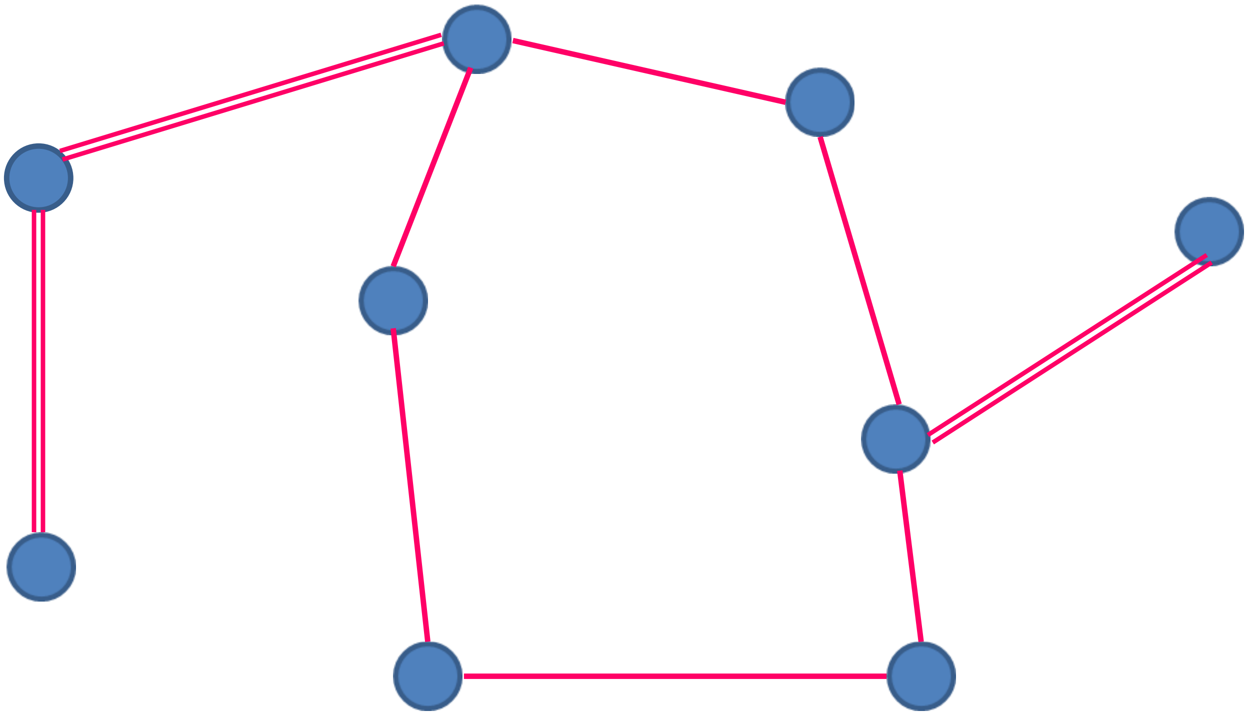
… and the spanning tree



Union of T-join and the spanning tree



Drop remaining edges



Analysis

The algorithm gives a connected multi-graph with even degrees. It has an Eulerian tour.

Spanning trees and minimum T-joins can be found in polynomial time.

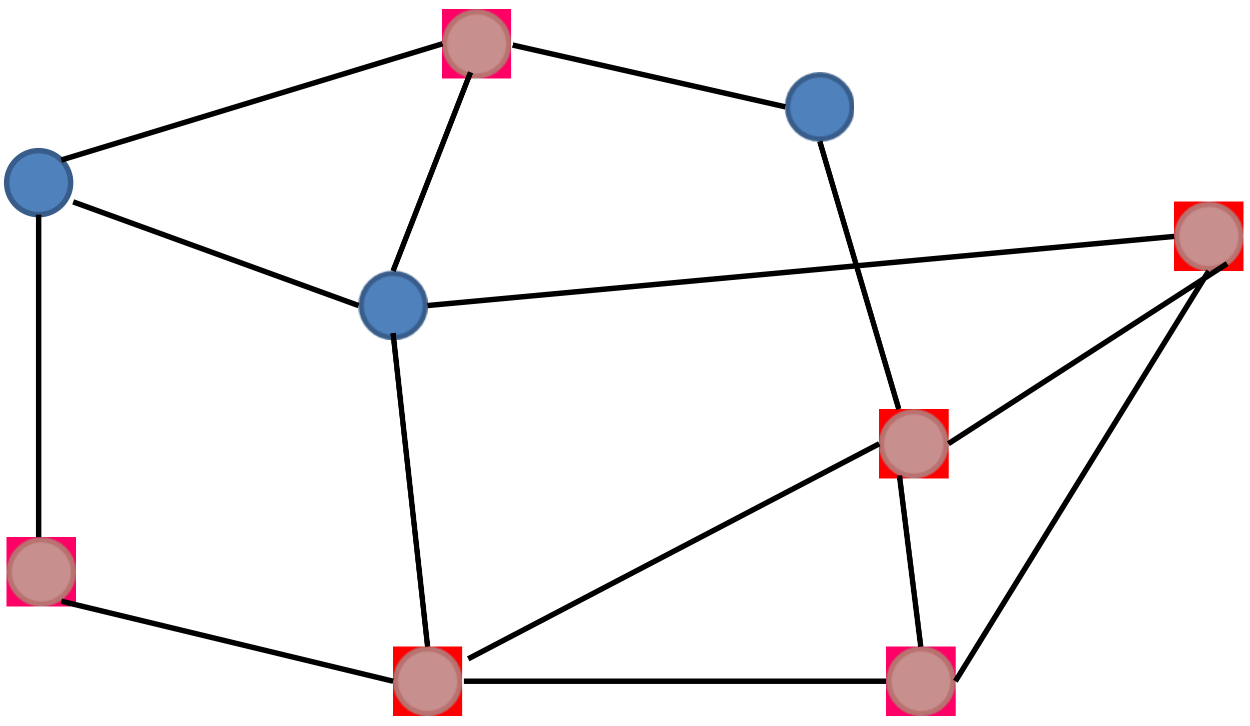
Approximation ratio:

Spanning tree < opt

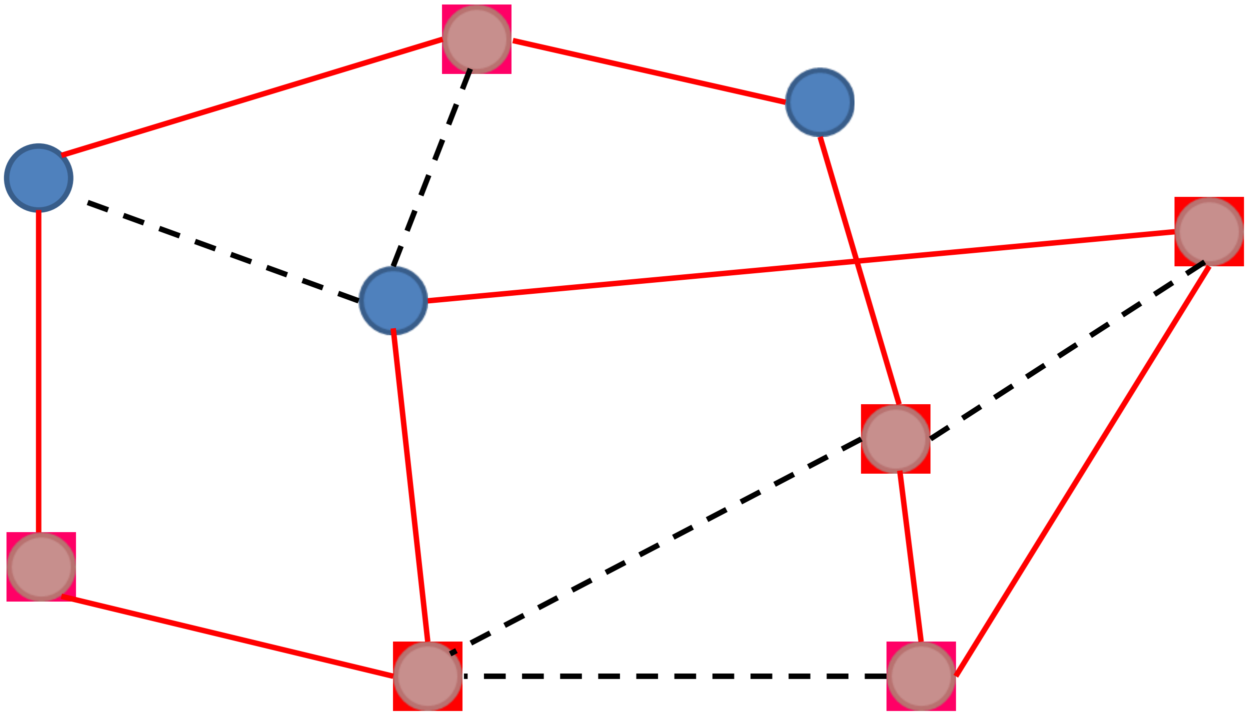
Minimum T-join ≤ opt/2

Christofides tour < 3opt/2

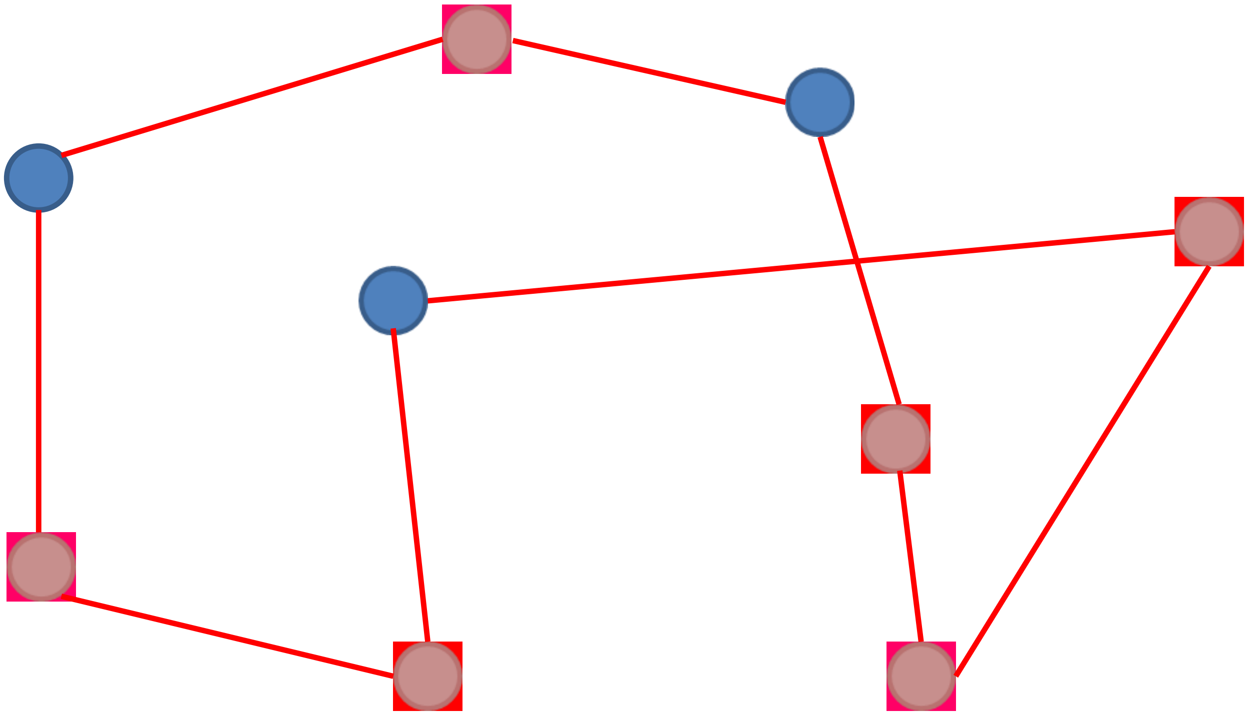
Upper bound on T-join



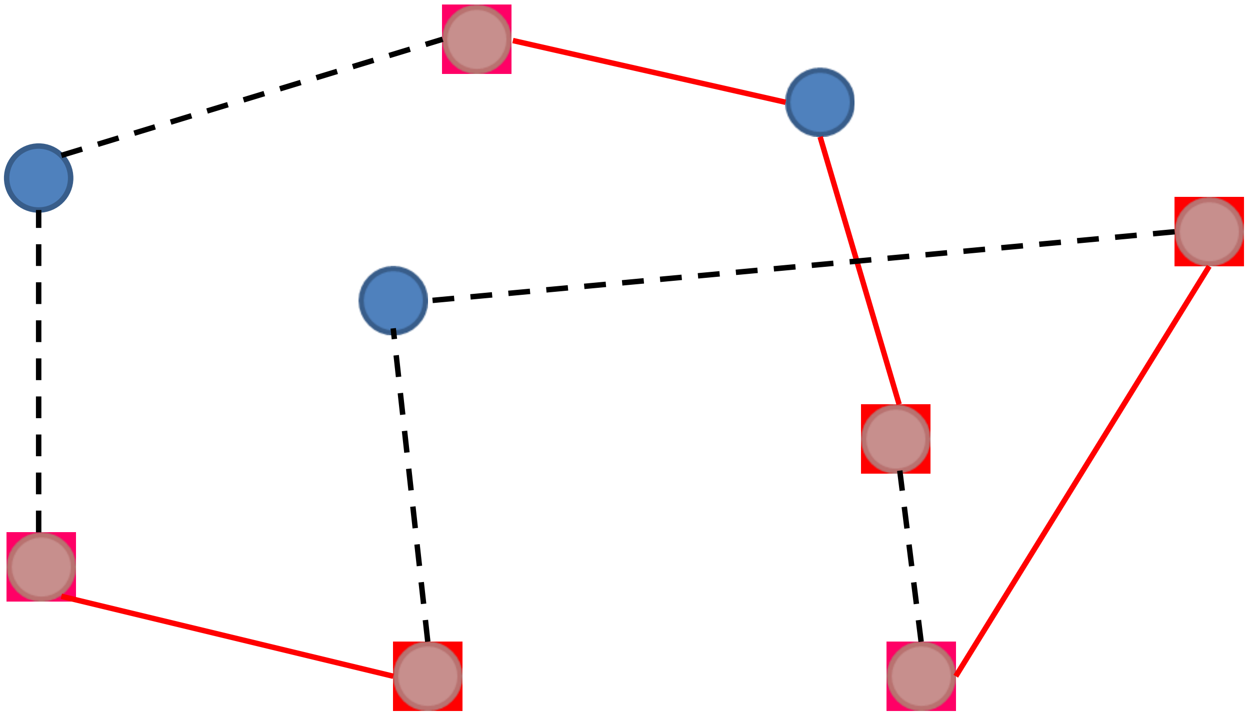
Consider optimal tour



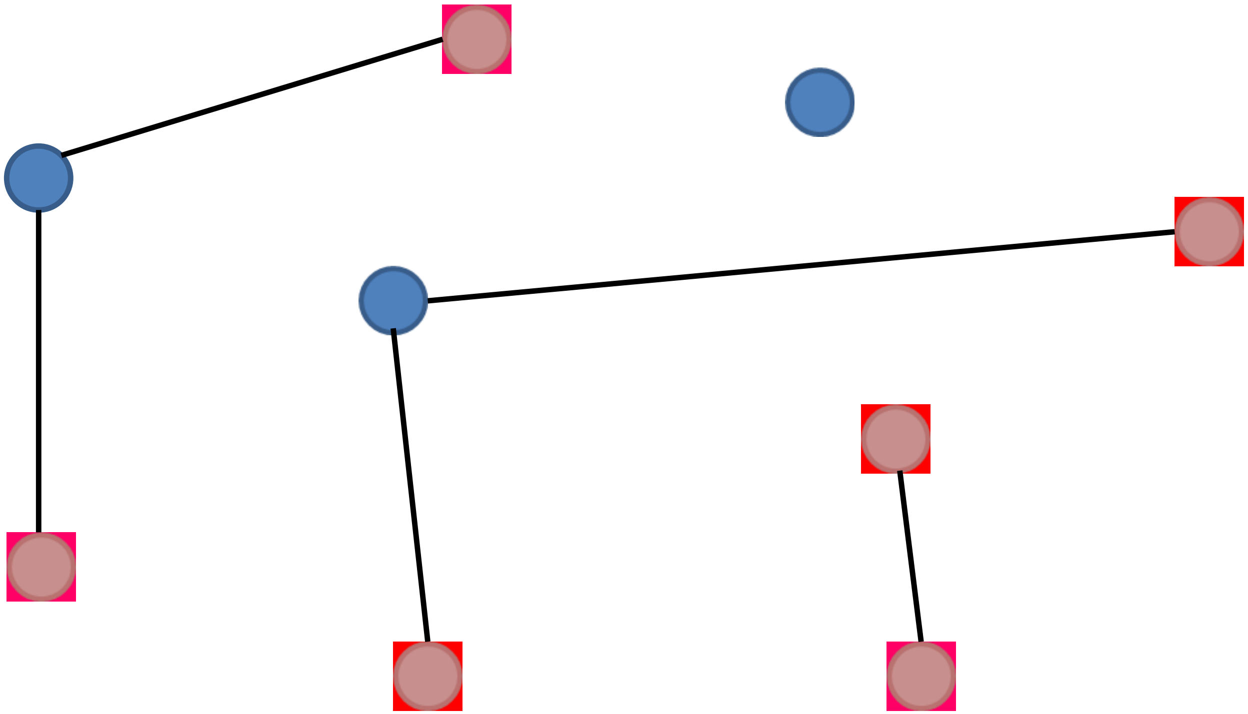
Ignore other edges



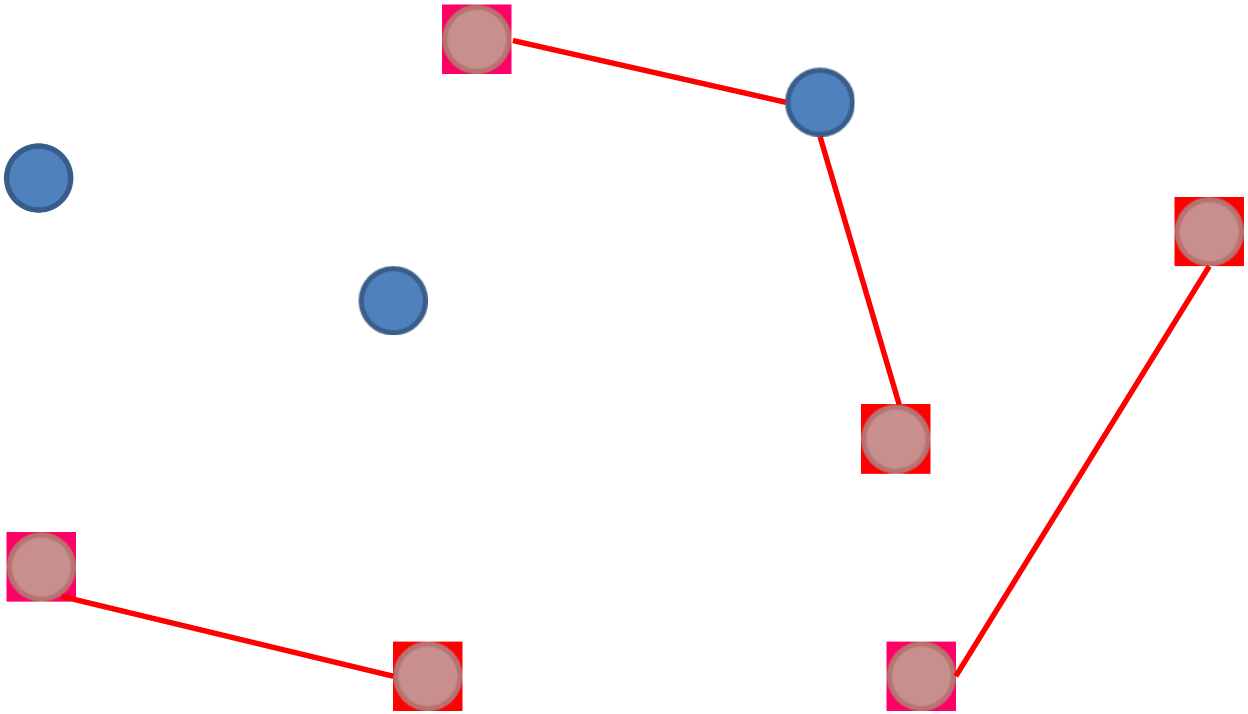
Take either even or odd segments



The even segments



The odd segments



Steiner Cycles

Obs: If G has a spanning tree T and a simple cycle C on the odd nodes of T, then it has a tour of length $4n/3$

- $|C| > 2n/3$ Contract cycle and double remaining spanning tree
- $|C| < 2n/3$ Use Christofides' idea and take shorter of even and odd segments to get T-join of size at most $n/3$

Cor: If G has a Hamiltonian path, then it has a tour of length $4n/3$

Approximating TSP

- NP-hard and APX-hard.
- 3/2 still best approx ratio for metric TSP.
- Dantzig, Fulkerson, Johnson (aka Held-Karp) linear program gives at least as good approximation. Moreover, worst integrality gap example known is 4/3, on an instance of graph TSP of max degree 3.
- For graph TSP, there has been substantial progress in recent years, leading to 7/5 approximation [Sebo and Vygen 2012]

Oveis-Gharan, Saberi and Singh 2011

Thm: Graph TSP can be approximated within a ratio better than 3/2.

Proof idea: Rather than starting from arbitrary spanning tree, start with one that would give a cheaper T-join than OPT/2.
Use fractional solution of LP to define a distribution over spanning trees, sample one at random, and it is likely to have a cheap T-join.

Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

Mömke and Svensson 2011

- A different approach, giving a 1.461 approximation for graph-TSP.
- Every 3-regular 2-vertex connected graph has a tour of length at most 4n/3. *(Also proven independently and differently by Aggarwal, Garg, Gupta '11 and Boyd, Sitters, van der Ster, Stougie '11)*
- Improvement of analysis to 13/9 *(Mucha '12)*

Mömke and Svensson 2011

Every 3-regular 2-vertex connected graph has a tour of length at most 4n/3.

New ideas:

- Use probability distribution over T-joins to fix up a single tree
- *Delete* carefully chosen edges from T-join

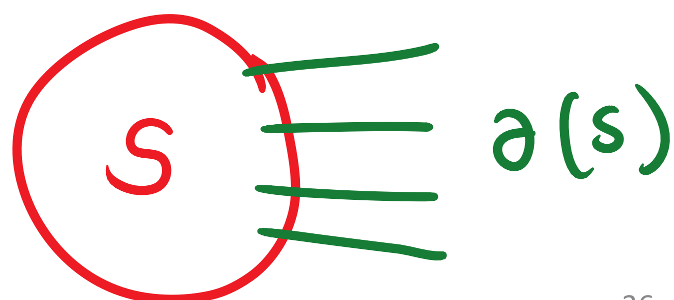
Naddef and Pulleyblank 1981

Assigning every edge in a 3-regular 2-vertex connected graph a value of 1/3 puts it in the perfect matching polytope

Theorem [Edmonds 1964]: Perfect Matching polytope characterization

$$x(\partial(v)) = 1 \quad \forall v$$

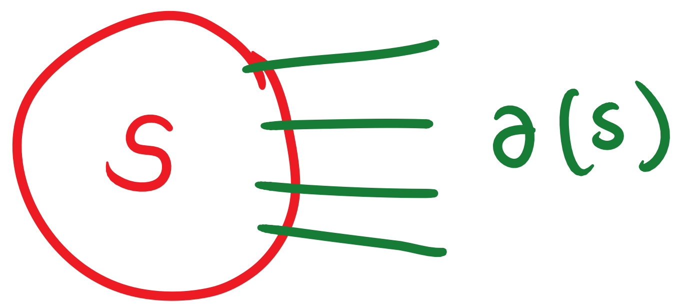
$$x(\partial(S)) \geq 1 \quad \forall S : |S| \text{ odd}$$



Naddef and Pulleyblank 1981

Assigning every edge in a 3-regular 2-vertex connected graph a value of 1/3 puts it in the perfect matching polytope

$$x(\partial(S)) \geq 1 \quad \forall S : |S| \text{ odd}$$



- |S| odd and cubic graph implies |δ(S)| odd
- 2-connectivity implies |δ(S)| ≥ 3

Naddef and Pulleyblank 1981

Assigning every edge in a 3-regular 2-vertex connected graph G a value of 1/3 puts it in the perfect matching polytope

(Caratheodory's theorem): G with 1/3 on every edge can be written as a convex combination of a polynomial number of perfect matchings M1, M2, …, Mk

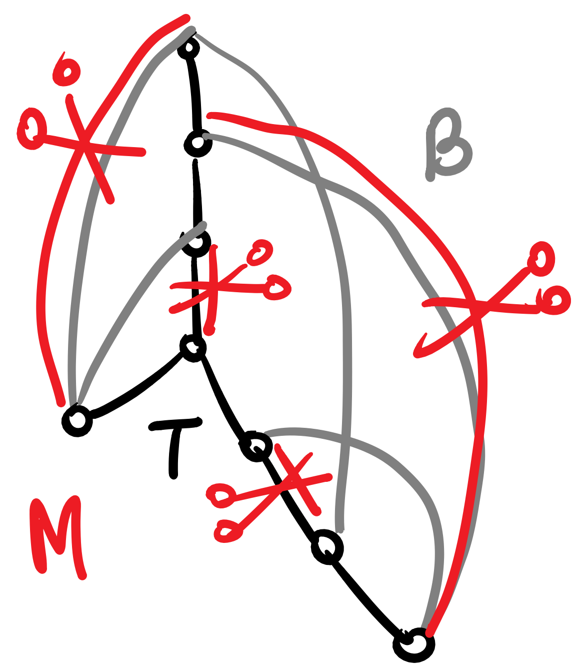
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in $M \cap B$, delete it from H
  - For all edges in $M \cap P$, delete it from H
  - For all edges in $M \cap Q$, double it in H

Claim: H is an Eulerian connected graph (and hence contains a tour)

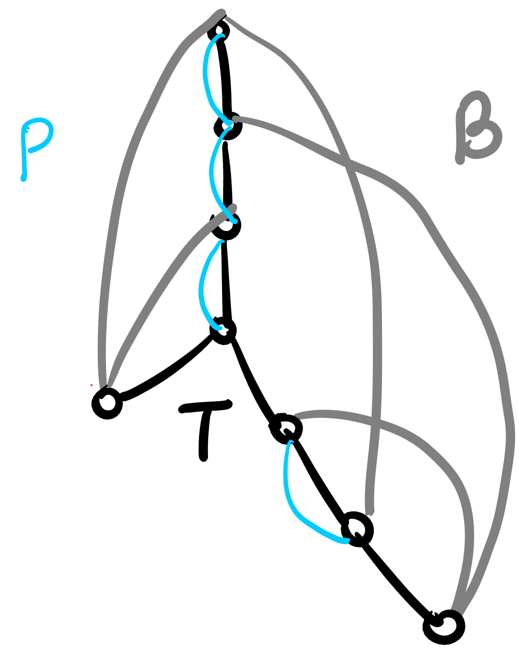
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



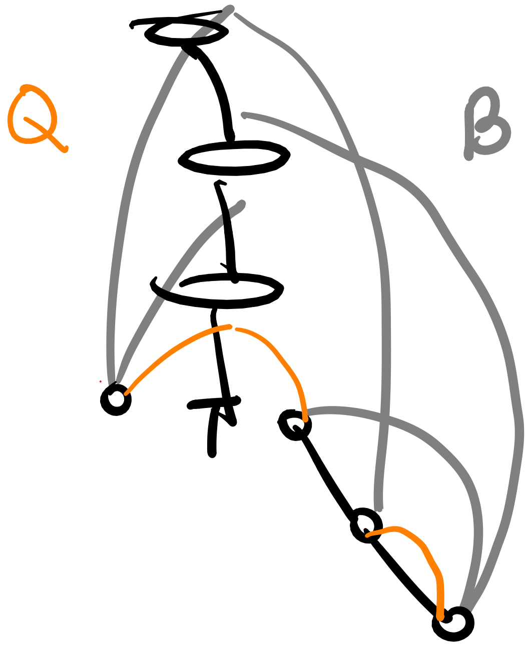
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P



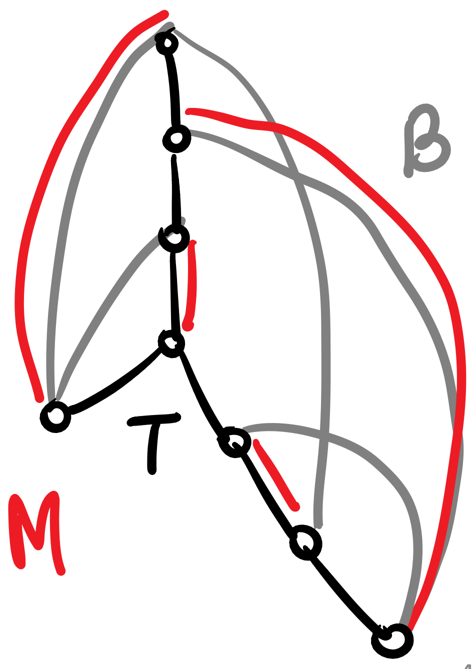
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P



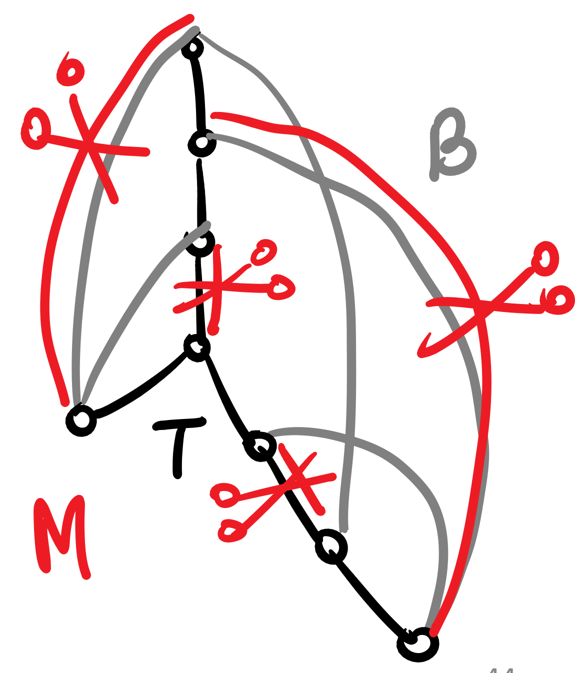
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



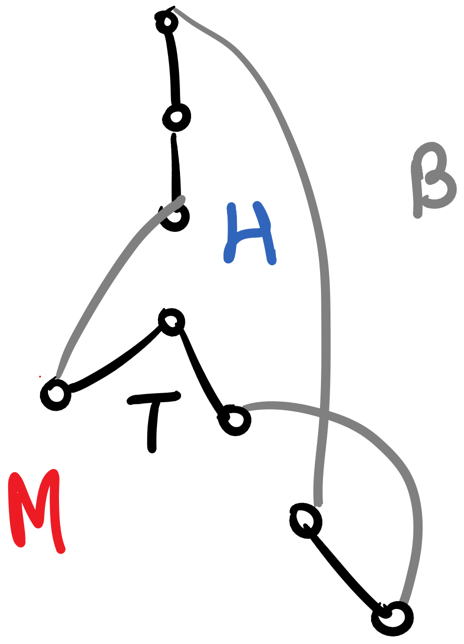
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



Mömke and Svensson 2011

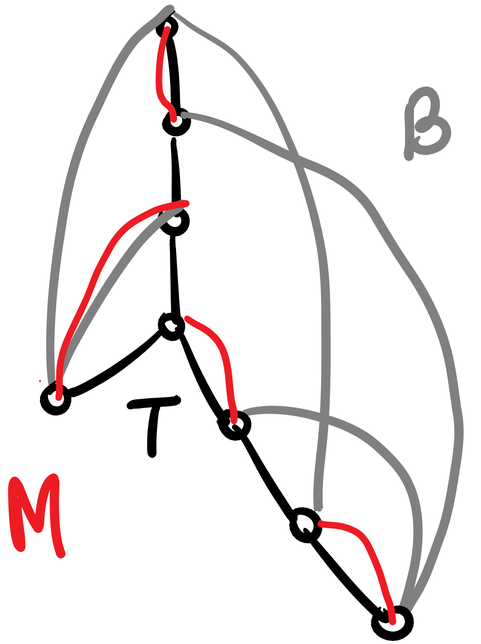
- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



Claim: H is a connected Eulerian graph

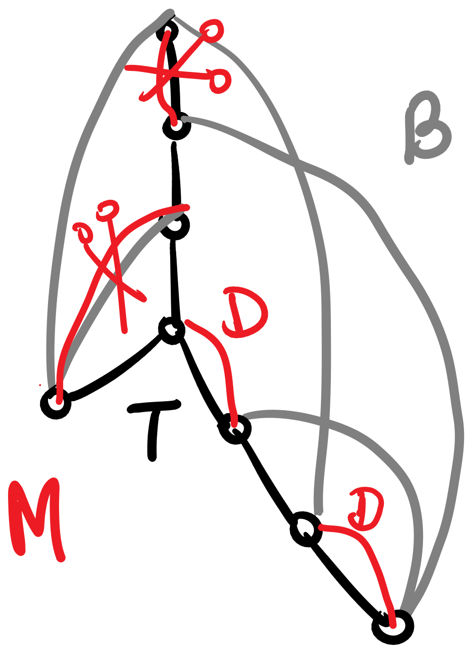
Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



Mömke and Svensson 2011

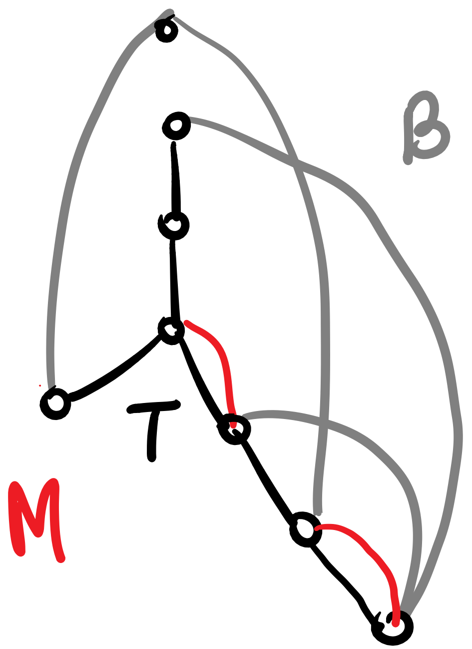
- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H



Mömke and Svensson 2011

- Pick a DFS tree T with a set of back edges B
  - P: Tree edges with back edge hanging from parent
  - Q = T\P
- Pick a matching M randomly from the distribution defined by x=1/3 on E(G)
- Initialize solution H to whole graph G
  - For all edges in M ∩ B, delete it from H
  - For all edges in M ∩ P, delete it from H
  - For all edges in M ∩ Q, double it in H

Claim: H is a connected Eulerian graph



Mömke and Svensson 2011

Claim: H is an Eulerian connected graph (and hence contains a tour)

- Eulerian: Every node has initial degree 3. One matching edge incident is deleted or doubled making degree 2 or 4
- Connected (Bottom up induction)
  - If tree edge in P deleted, back edge hanging from parent connects subtree to upper part
  - If back edge in B is deleted, its sibling tree edge in P connects both sides

Mömke and Svensson 2011

$E[|H|] = |G| - E[|M \cap B|] - E[|M \cap P|] + E[|M \cap Q|]$

$= 3n/2 - (1/3)\ |B| - (1/3)\ |P| + (1/3)\ |Q|$

$\approx 3n/2 - (1/3)(n/2) - (1/3)\ (n/2) + (1/3)\ (n/2)$

$= 9n/6 - n/6$

$= 4n/3$

Theorem: Every 3-regular 2-vertex connected graph has a tour of length at most 4n/3

Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

Sebo and Vygen 2012

- Find a ‘nice’ ear decomposition of G based on Frank’s min-max theorem for min number of even ears
- Pick a set of edges based on the decomposition (ear-muff) to form a connected sugraph
- Extend chosen subgraph to an even supergraph by inductively adding edges from the pendant ears
  - If many pendant ears, add T-join on odd nodes (augment)
  - If few pendant ears, use Mömke and Svensson’s method (deletion)

Theorem: 7/5-approximation for graph-TSP (best known currently)

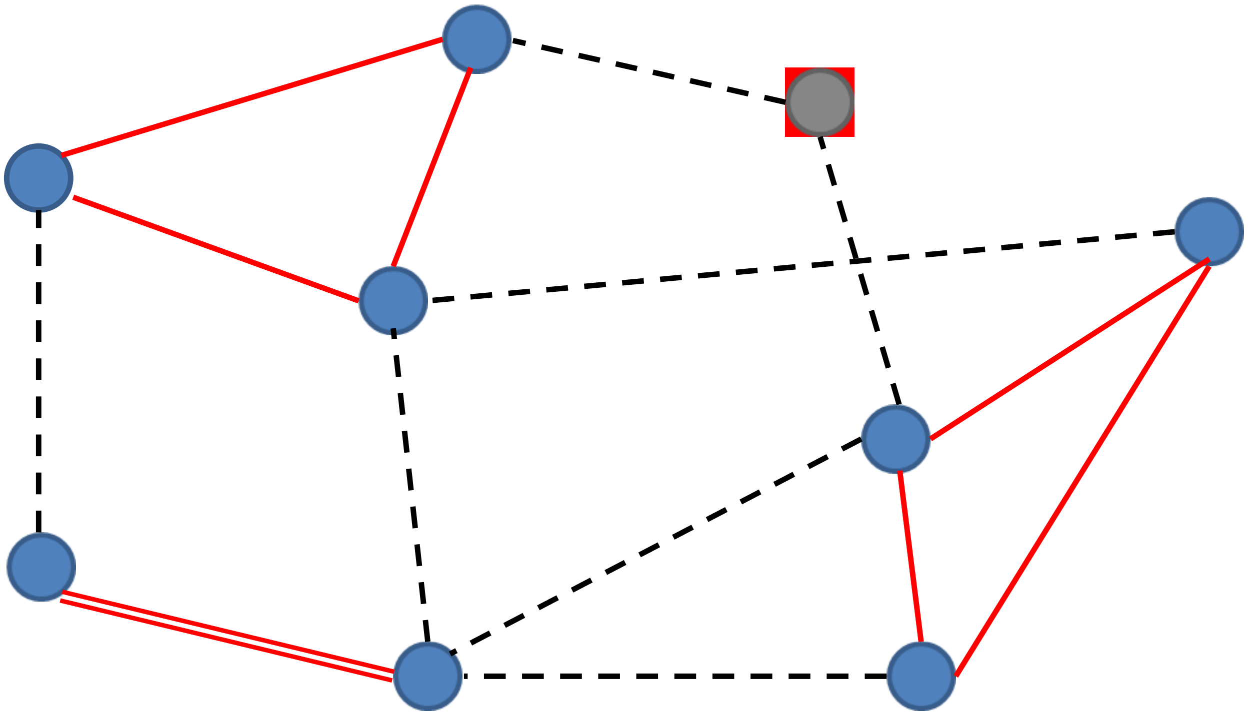
Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

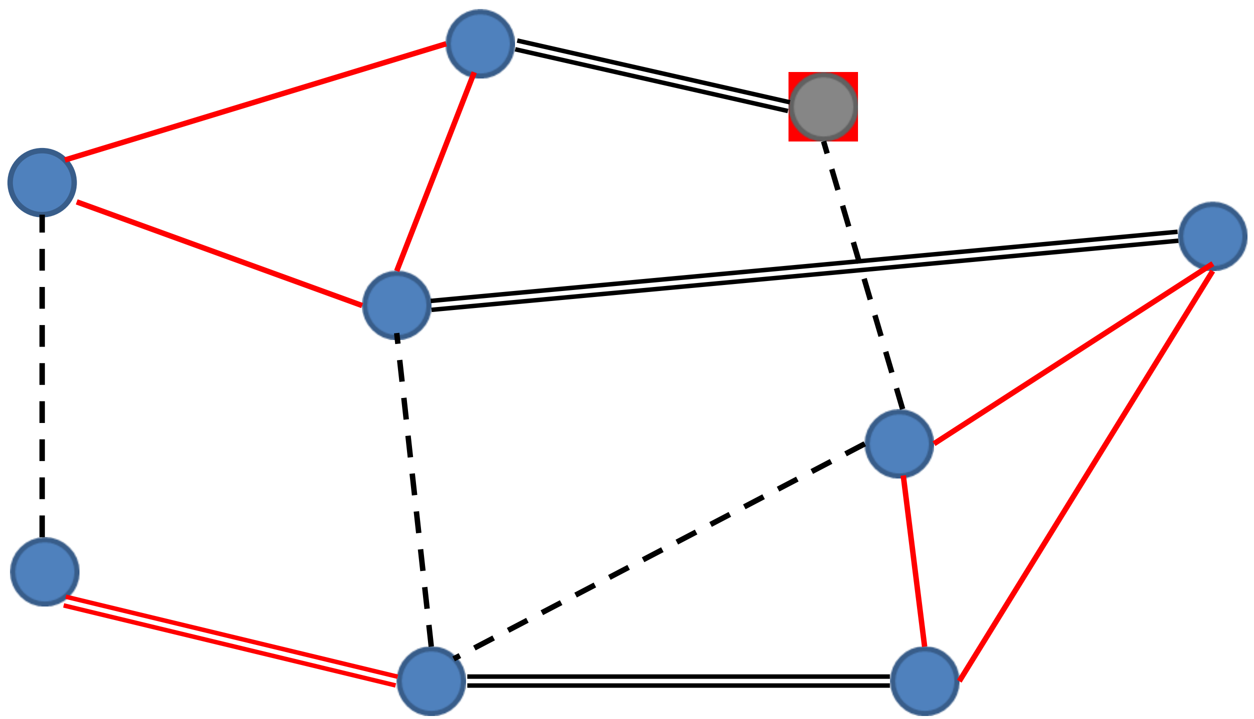
Augment cycle cover with few cycles

- Find a cycle cover with few cycles.
- Connect it by doubled edges to get a connected Eulerian multi-graph.
- If the cycle cover has c cycles, the tour length is at most $n + 2(c-1)$.
- Need c to be small, i.e., average cycle length to be large.

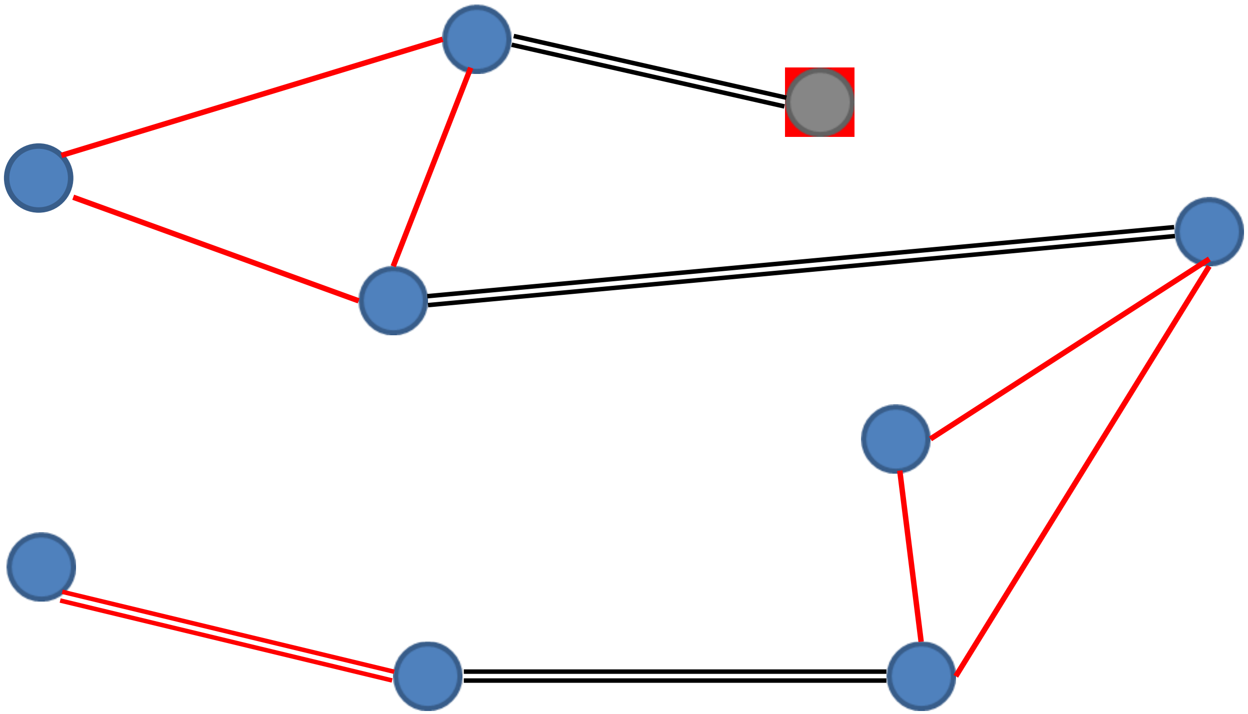
Cycle cover



Connecting the cycle cover



Drop remaining edges



Simple 3n/2 tour in cubic bipartite graphs

1. Find a cycle cover with no parallel edges (e.g. union of any two disjoint perfect matchings)

Bipartite implies no odd cycles → min cycle length is 4 → At most n/4 cycles

2. Add doubled spanning tree connecting cycles

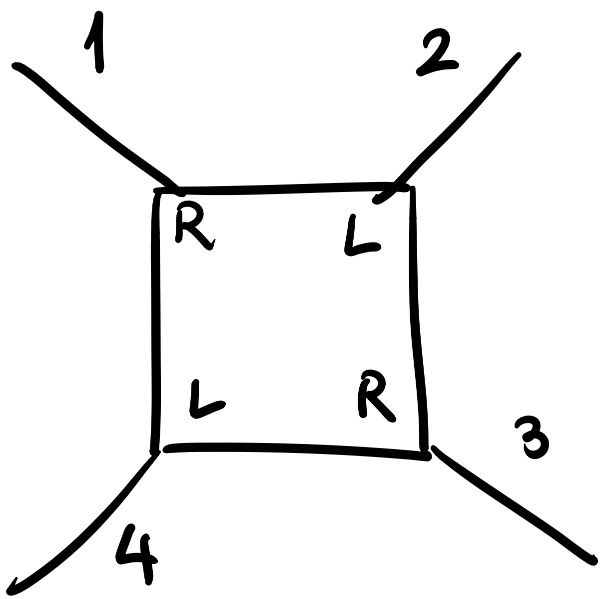
Total no of edges ≈ $n + 2(n/4)$

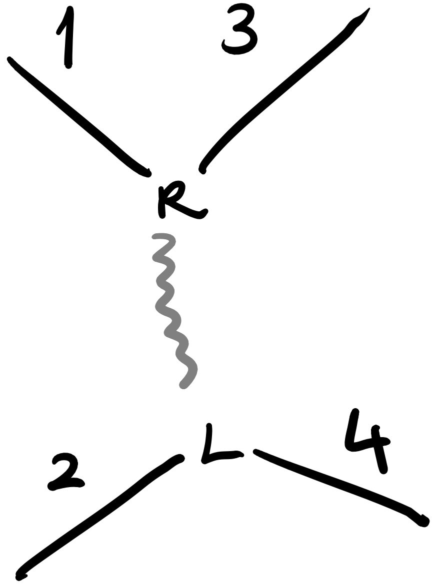
Simple 4n/3 tour in cubic bipartite graphs

1. While there is a square, replace it with a gadget
2. Find cycle cover in square-free graph
3. Expand gadgets maintaining square-free cycle cover

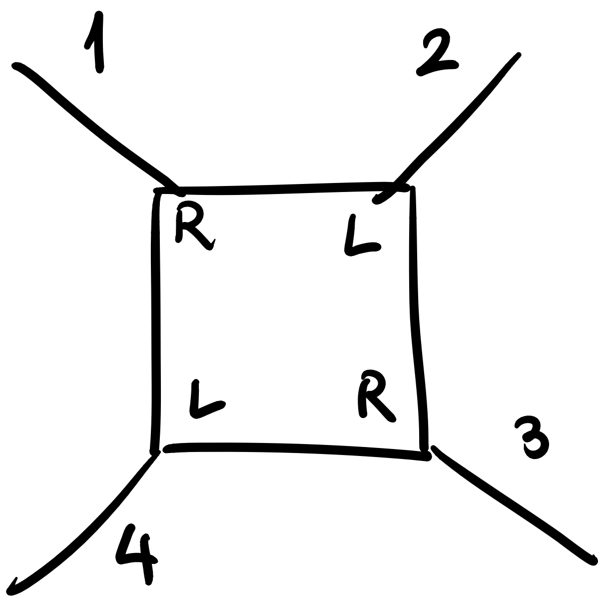
Total no of edges ≈ $n + 2(n/6)$

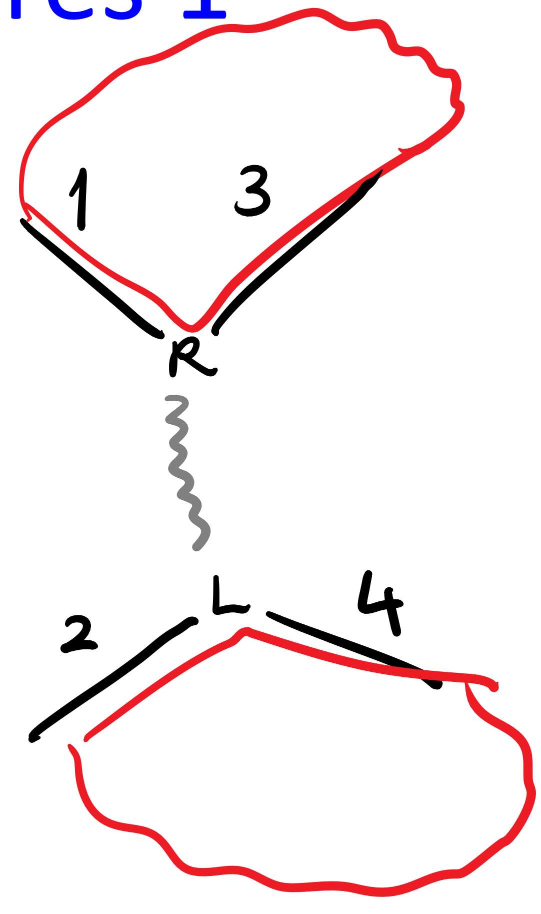
Replacing Squares



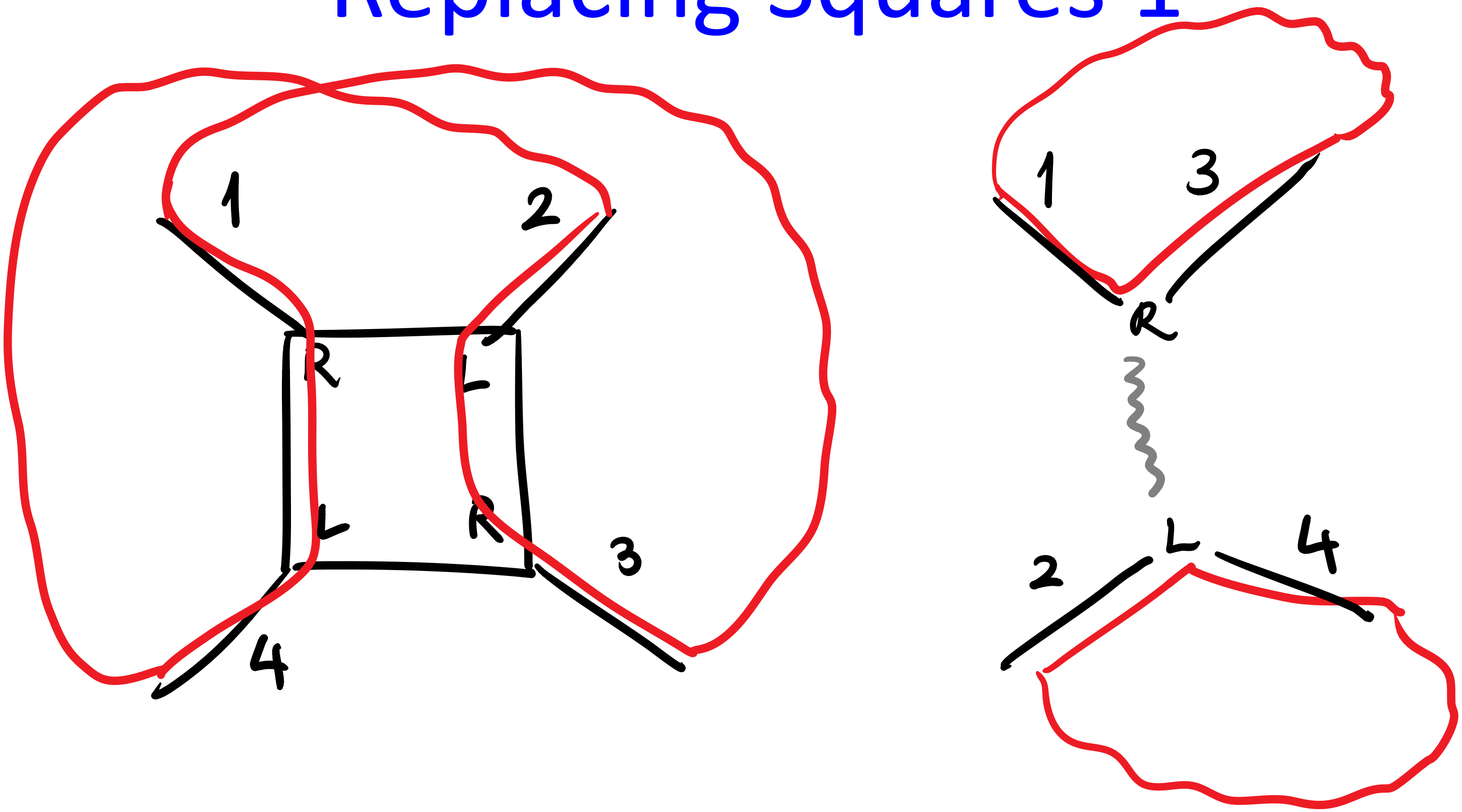


Replacing Squares 1

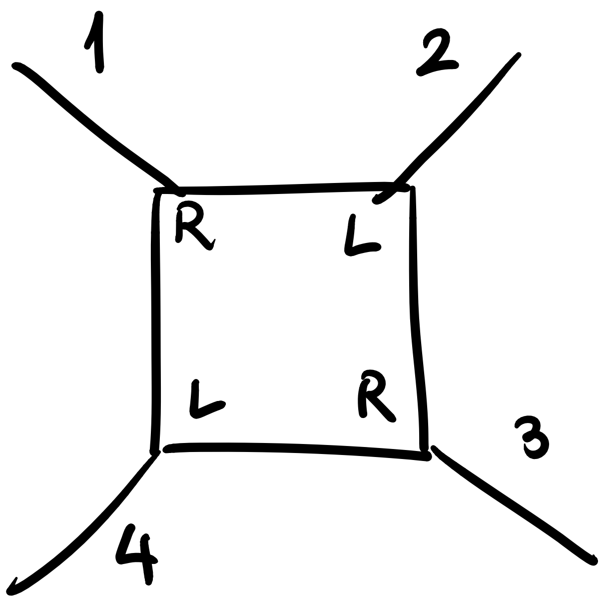


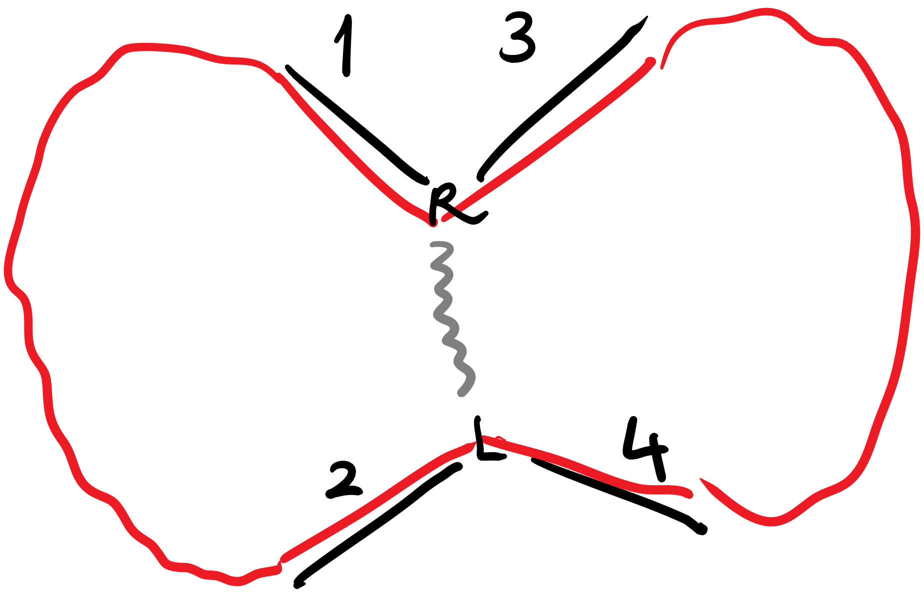


Replacing Squares 1

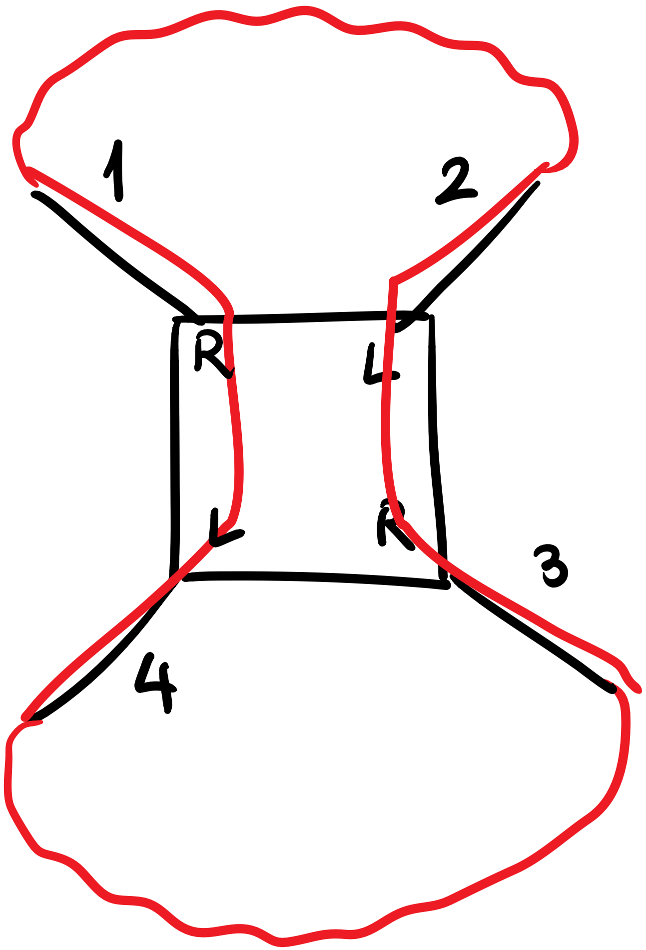


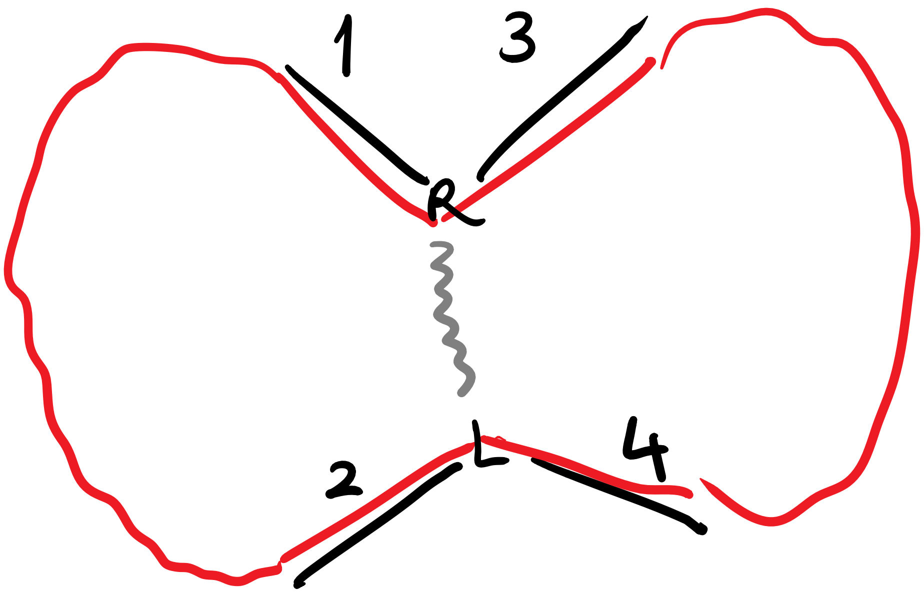
Replacing Squares 2



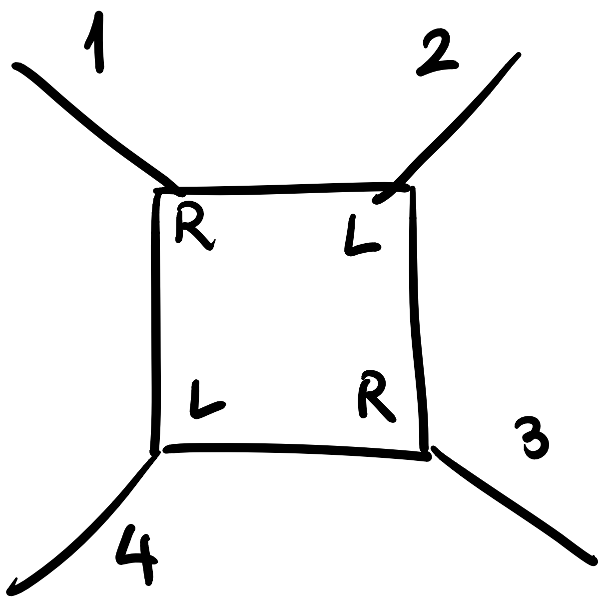


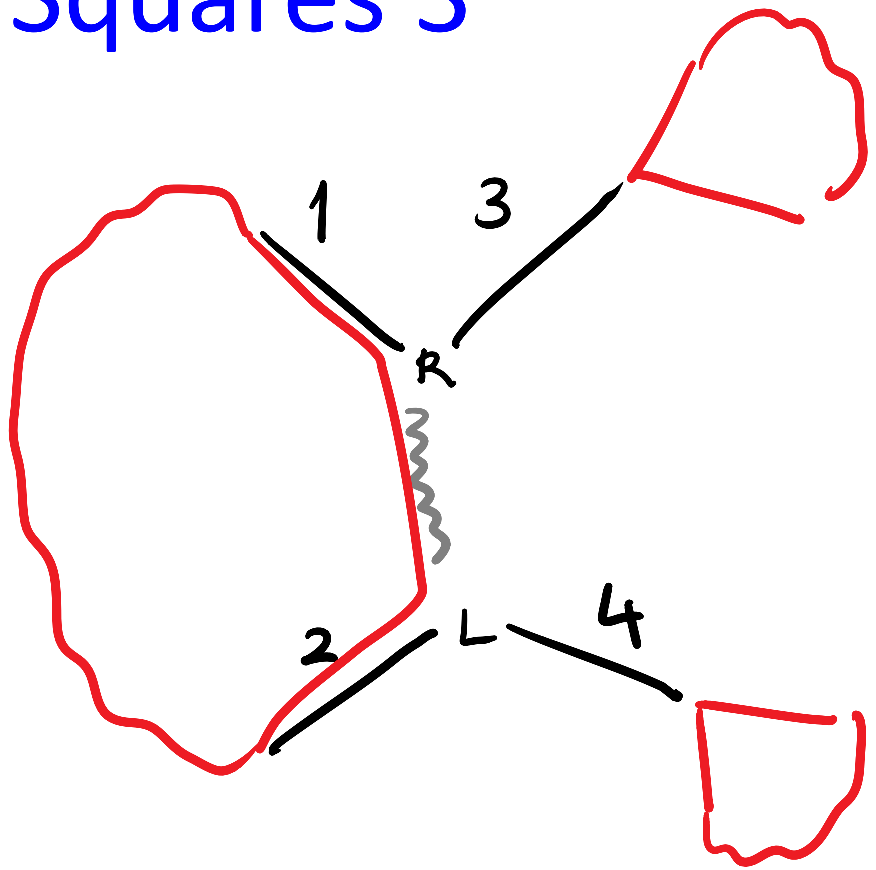
Replacing Squares 2



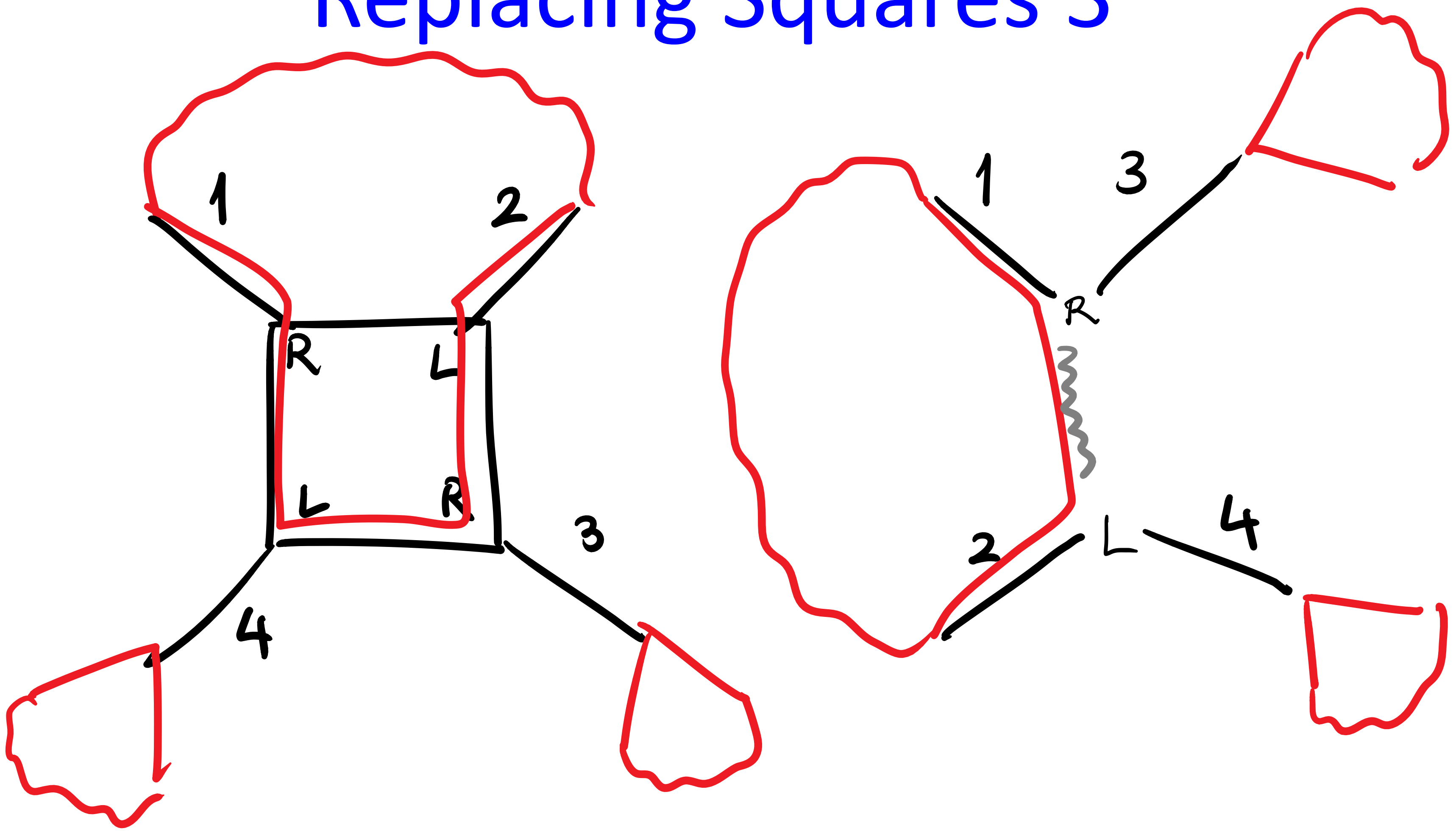


Replacing Squares 3



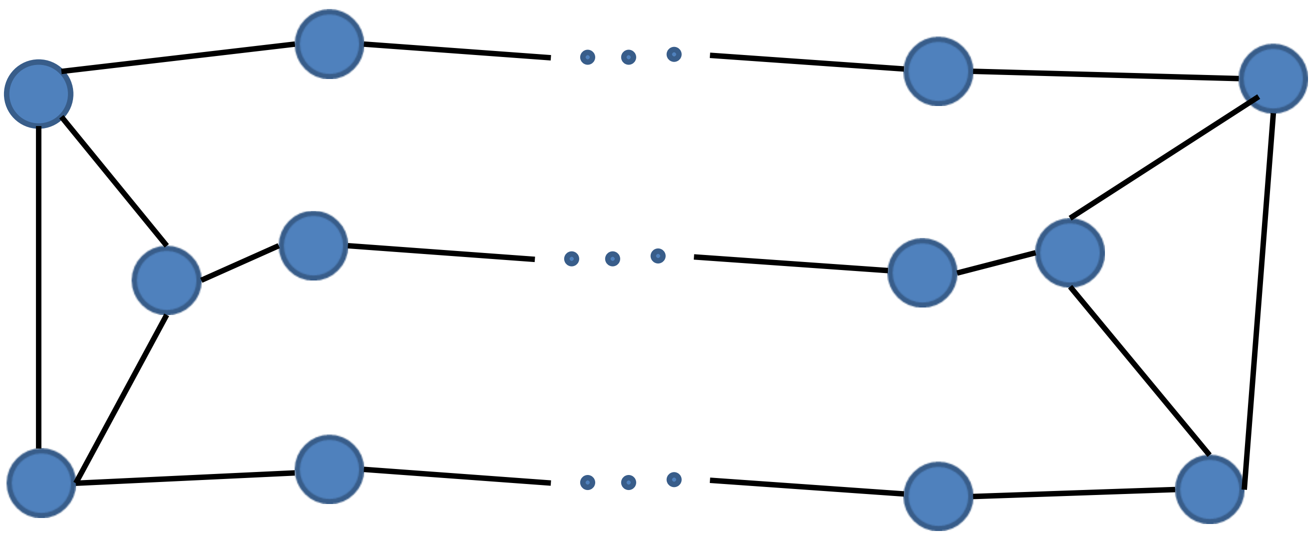


Replacing Squares 3



Better than 4/3 approximation?

Sub-cubic graph instances already require 4n/3 edges.



Better than 4/3 approximation?

Correa, Larre, Soto 2011, 2012

2-edge-connected cubic graphs have a tour of length

$(4/3 - 1/61236)n$

3-edge-connected bipartite cubic graphs have a tour of length $(4/3 - 1/108)n$

Jeremy Karp, Ravi 2013

Cubic bipartite graphs have a tour of length $9n/7$

Van Zuylen 2015

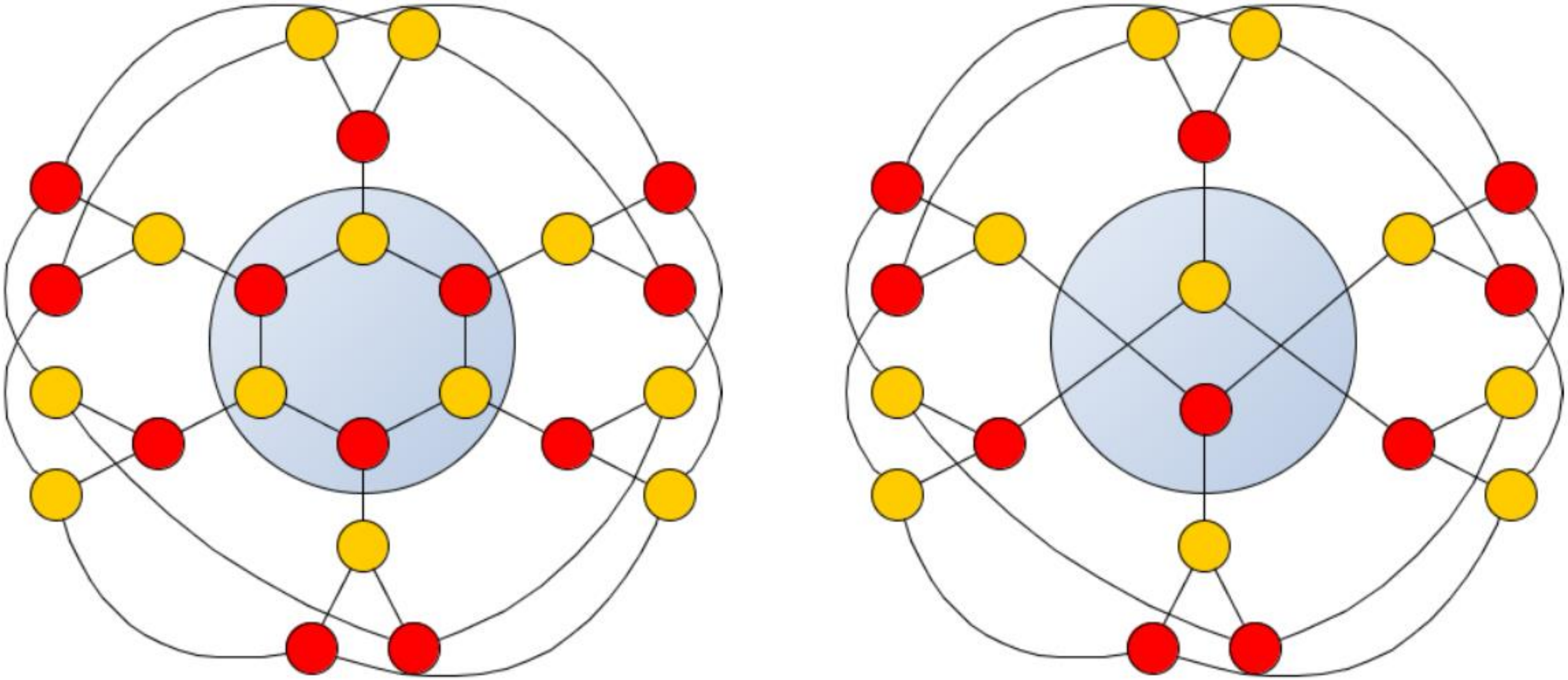
Cubic bipartite graphs have a tour of length $5n/4$

Open: Barnette's Conjecture : 3-connected planar bipartite cubic graphs have a Hamiltonian cycle

Newman 2014

Momke-Svensson gives tour of length $46n/33$ in subquartic graphs

Main idea: Replace short cycles



Algorithm Sketch

"Organic": made up of original nodes and edges (not resulting from earlier replacements)

1. While graph contains 4-cycle or organic 6-cycle, COMPRESS by replacing cycle with gadget
2. Find cycle cover in final compressed graph
3. EXPAND compressed cycles in reverse order rewiring cycle cover to span all deleted nodes

Algorithm Example

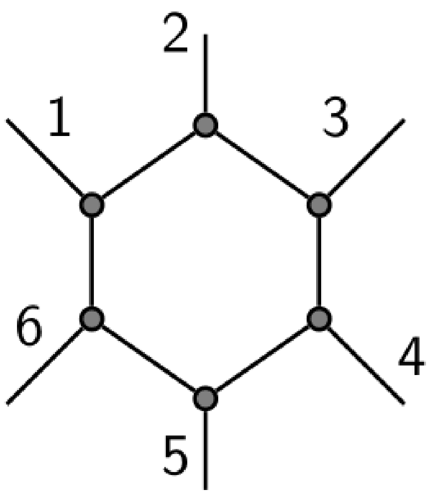


Figure : An organic 6-cycle

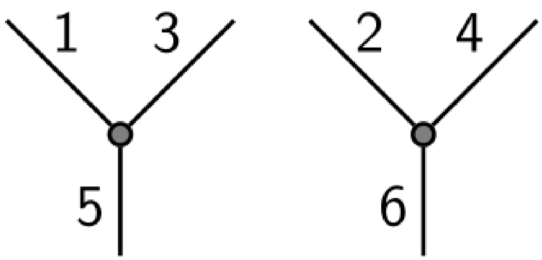


Figure : The gadget which replaces the 6-cycle

Algorithm Example

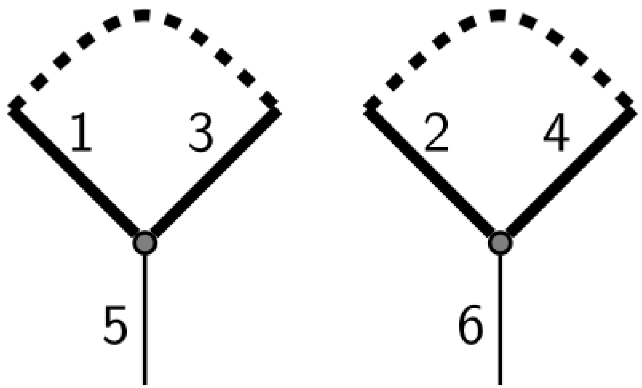


Figure : Part of the cycle cover in the compressed graph

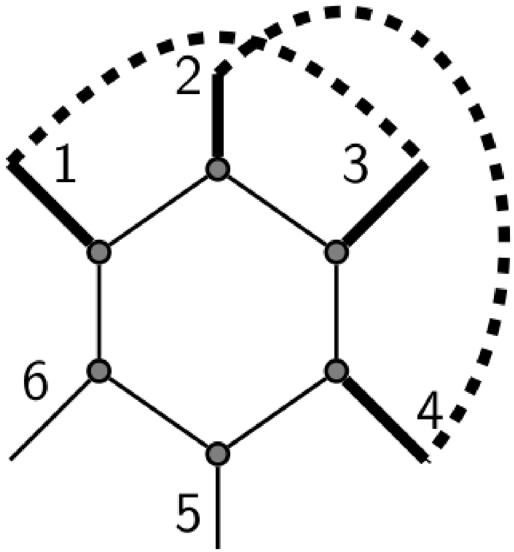
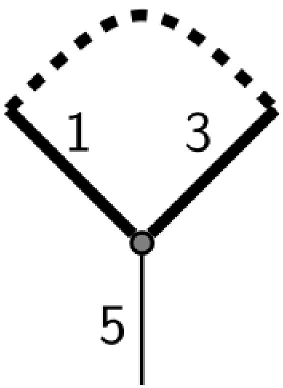


Figure : The same portion of the cycle cover, after expanding the graph

Algorithm: Good Expansion



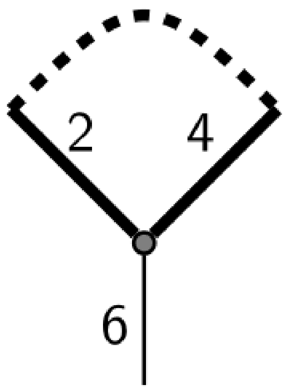


Figure : Part of the cycle cover in the compressed graph

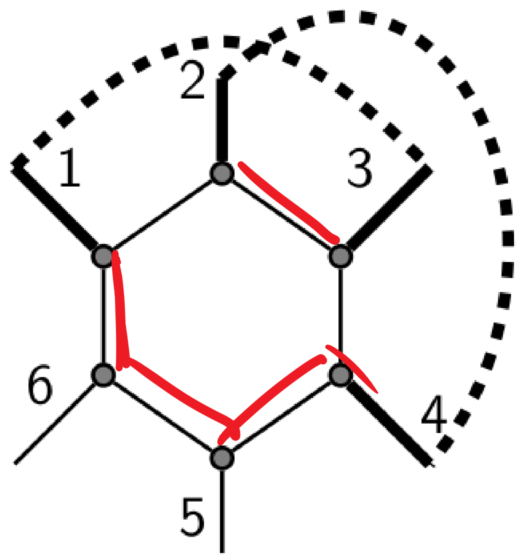


Figure : The same portion of the cycle cover, after expanding the graph

Algorithm: Bad Expansion

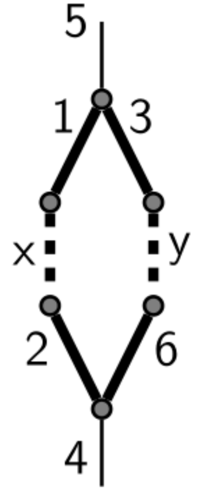


Figure : A cycle of length $x + y + 4$ that passes through a gadget that replaced a 6-cycle

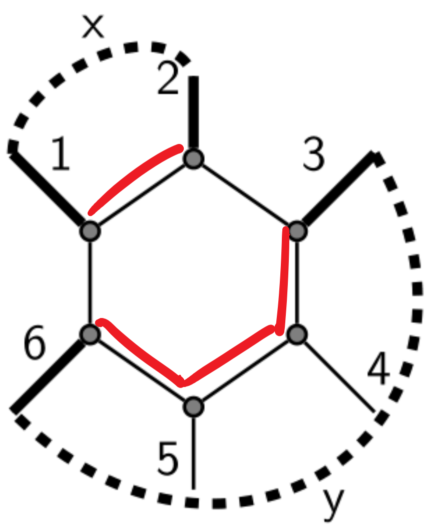


Figure : The cycle from the previous figure, after expanding the graph

Handling Bad Expansions

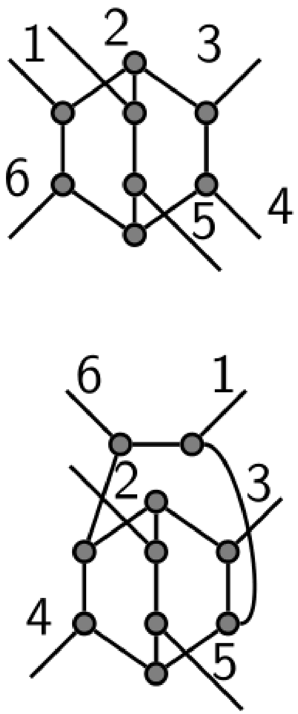


Figure : Two other “bad” subgraphs

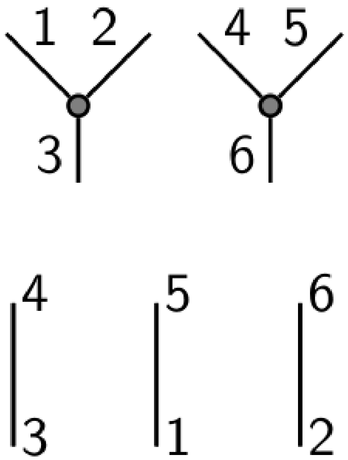


Figure : The gadgets which replace the “bad” subgraphs

After handling, y ≥ 3

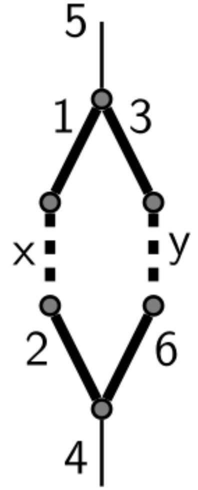


Figure : A cycle of length $x + y + 4$ that passes through a gadget that replaced a 6-cycle

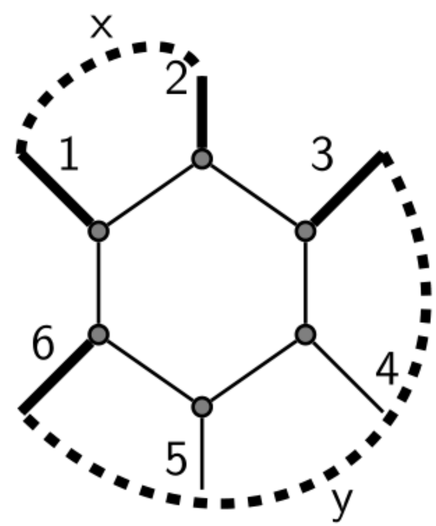
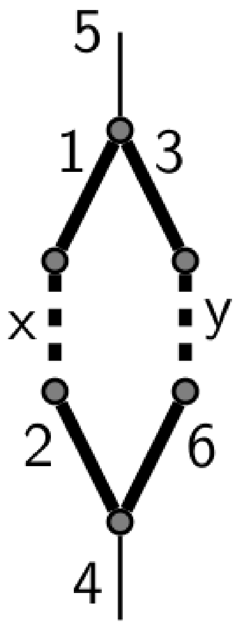
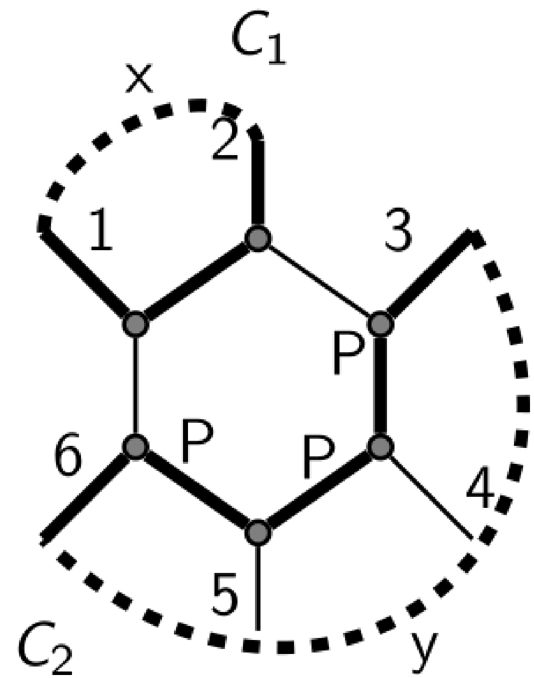


Figure : The cycle from the previous figure, after expanding the graph

Analysis: Protected Edges





Analysis Outline

- Every 6-cycle can be disjointly charged towards 2 or 3 protected edges
- If a final cycle has P protected edges, at most P/3 6-cycles are charged to it
- Final amortized average cycle length ≥ 7
- Total number of edges ≤ n + 2 (n/7)

Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

Nisheeth Vishnoi 2012

- Thm: Every n-vertex d-regular graph has a tour of length $(1 + o(1))n$, where the $o(1)$ term tends to 0 as d grows.
- Moreover, such a tour can be found in random polynomial time.

Vishnoi's approach

- Find a cycle cover with few cycles.
- Connect it by doubled edges to get a connected Eulerian multi-graph.
- If the cycle cover has c cycles, the tour length is at most $n + 2(c-1)$.
- Need c to be small.

Key questions

- Why would a d-regular graph have a cycle cover with few cycles?
- Even if such a cycle cover exists, how can it be found?
(A Hamiltonian cycle is a cycle cover with one cycle, but it is NP-hard to find it.)

Matrix representation of cycle covers

Given G, consider its n by n adjacency matrix A.

An all-1 permutation is a cycle cover.

	1	1		1			
1			1				1
1			1		1		
	1	1		1			
1			1			1	
		1				1	1
				1	1		1
	1				1	1	

Permanents and cycle covers

- The permanent of the adjacency matrix is precisely the number of cycle covers.
- For d-regular graphs, the matrix is doubly stochastic (after scaling by 1/d).
- Van-der-Warden's conjecture [1926] (proved by Egorychev and by Falikman [1981]) implies that the permanent is large.

Permanents and cycle covers [Vishnoi 2012; ~Noga Alon 2003]

- Van-der-Warden's conjecture implies that d-regular graphs have many different cycle covers.
- There are only few permutations with linearly many cycles (a random permutation has O(log n) cycles).
- A random cycle cover in a d-regular graph has $O(\frac{n}{\sqrt{\log d}})$ cycles.

Vishnoi's algorithm

Use the approximation algorithm of Jerrum Sinclair and Vigoda [2004] for the permanent to find a random cycle cover.

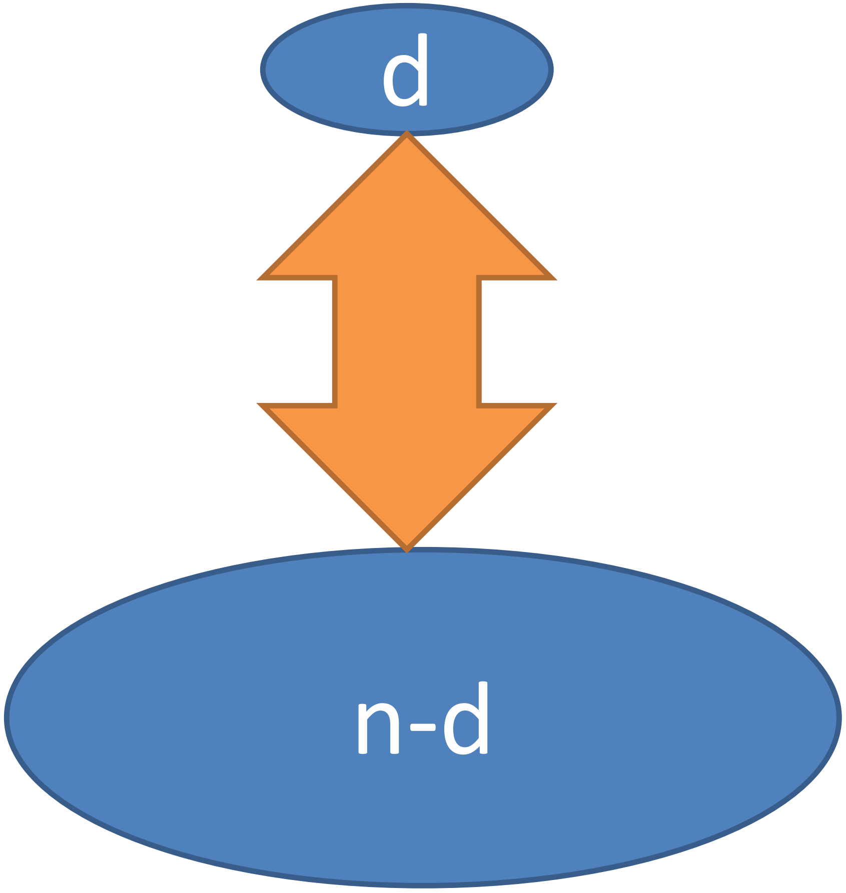
Connect it using double edges to get a connected Eulerian subgraph.

In a d-regular graph, this gives a tour of length

$$\left(1 + O\left(\frac{1}{\sqrt{\log d}}\right)\right)n$$

Regularity is essential

Graphs of minimum degree d need not have short tours.



Improved bounds

Thm: Every n-vertex d-regular graph has a tour of length (1 + o(1))n, where the o(1) term tends to 0 as d grows.

Moreover, such a tour can be found in random polynomial time.

Vishnoi 2012 $o(1) = O(\frac{1}{\sqrt{\log d}})$

Feige, Ravi, Singh 2014 $o(1) = O(\frac{1}{\sqrt{d}})$

Our proof approach

- Find a spanning tree with a small set T of odd degree vertices.
- Find a small T-join, of size O(|T|) + O(n/d).

The union of the spanning tree and T-join is a connected Eulerian subgraph, hence a tour of length n + O(|T|) + O(n/d).

How small can we make |T|?

In our proof $|\mathrm{T}| = O\left(\frac{n}{\sqrt{d}}\right)$.

Proof approach

- Find a spanning tree with a small set T of odd degree vertices.
- Find a small T-join, of size $O(|T|) + O(n/d)$.

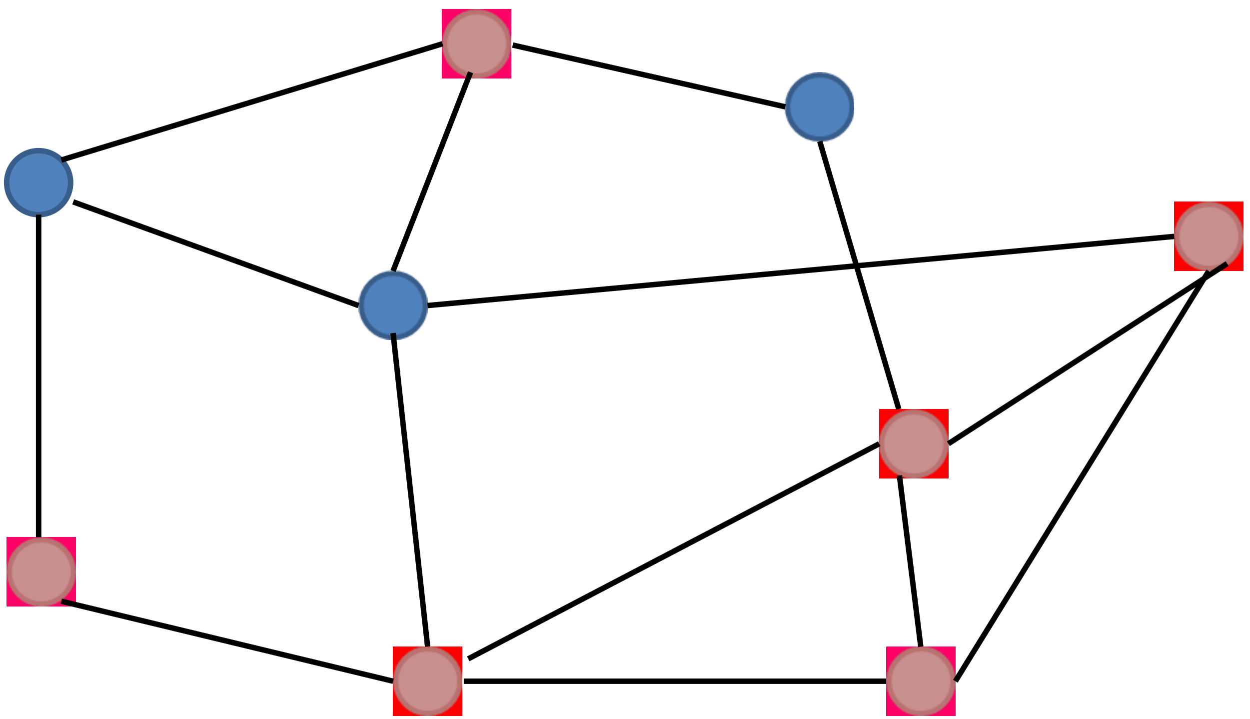
The union of the spanning tree and T-join is a connected Eulerian subgraph, hence a tour of length $n + O(|T|) + O(n/d)$.

Why is there a T-join of size $O(|T|) + O(n/d)$?

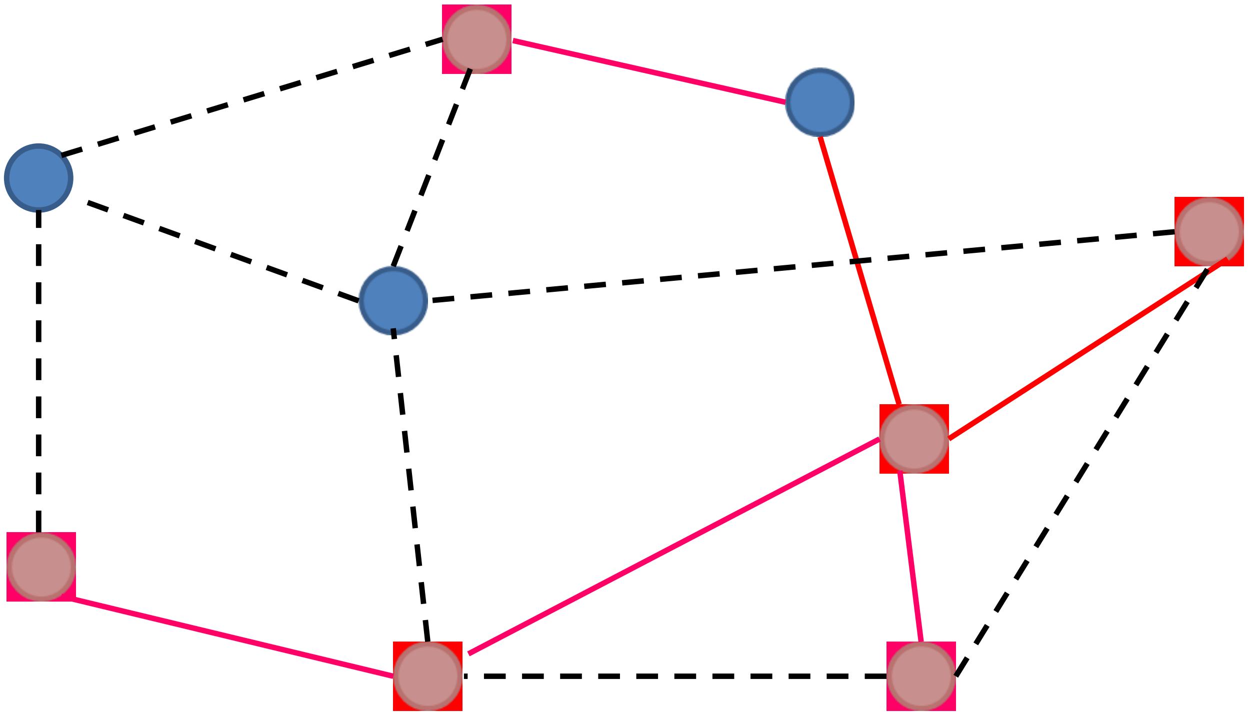
Let T’ be a tree spanning T.

Claim: There is a T-join supported only on edges of T’, and hence of size at most $|T'|-1$.

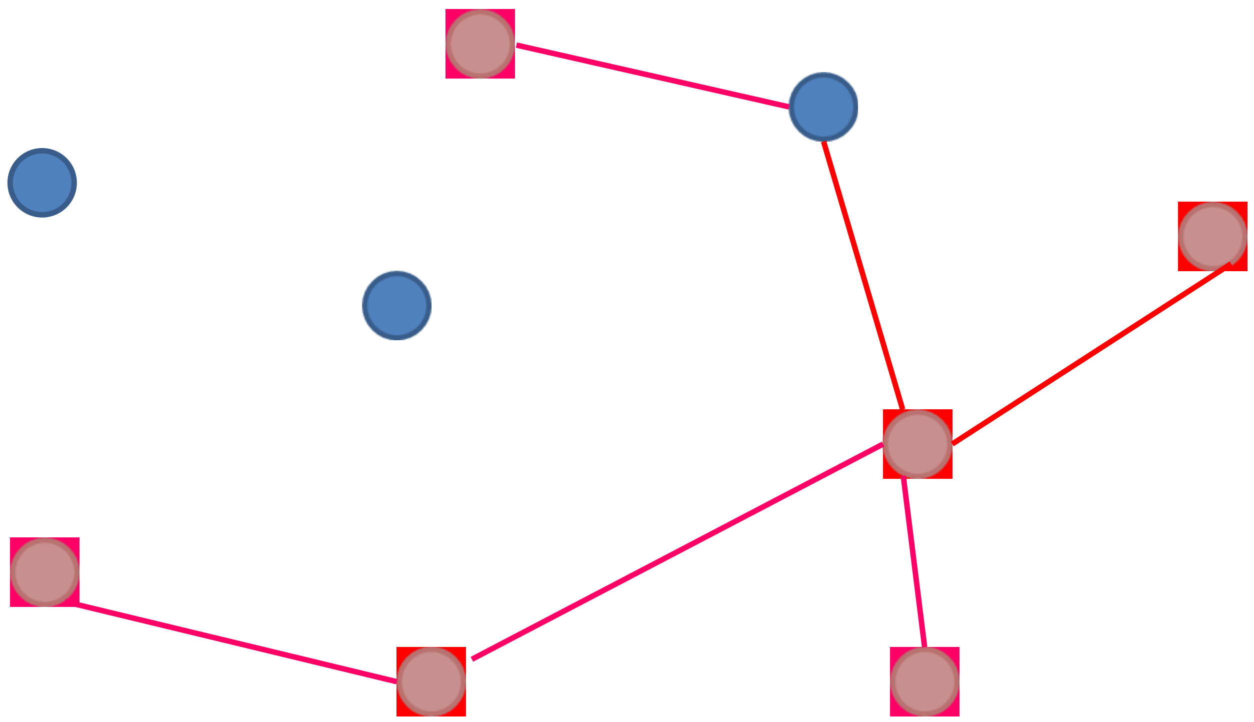
Upper bound on T-join



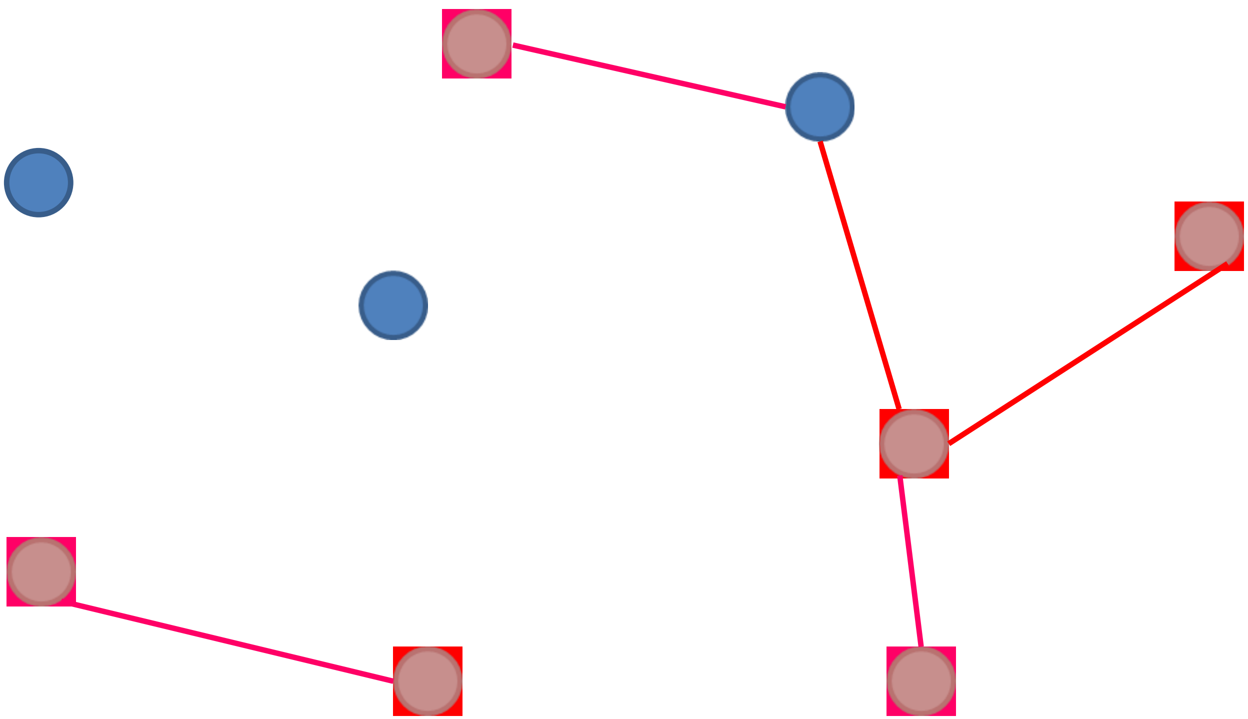
A tree T’ spanning T



A tree T' spanning T



A T-join



Tree T' of size at most $2|T| + 3n/(d+1)$ spanning a set T

- A 3-net: maximal set of vertices, no two of which at distance less than 3 from each other.
- 3-net has at most $n/(d+1)$ vertices.
- Every vertex from T at distance at most 2 from 3-net.
- All of T plus the 3-net can be connected by $2|T| + 3n/(d+1)-3$ edges.

Proof approach

- Find a spanning tree with a small set T of odd degree vertices.
- Find a small T-join, of size $O(|T|) + O(n/d)$.

The union of the spanning tree and T-join is a connected Eulerian subgraph, hence a tour of length $n + O(|T|) + O(n/d)$.

Proof approach

- Find a spanning tree with a small set T of odd degree vertices.
- Find a small T-join, of size $O(|T|) + O(n/d)$.

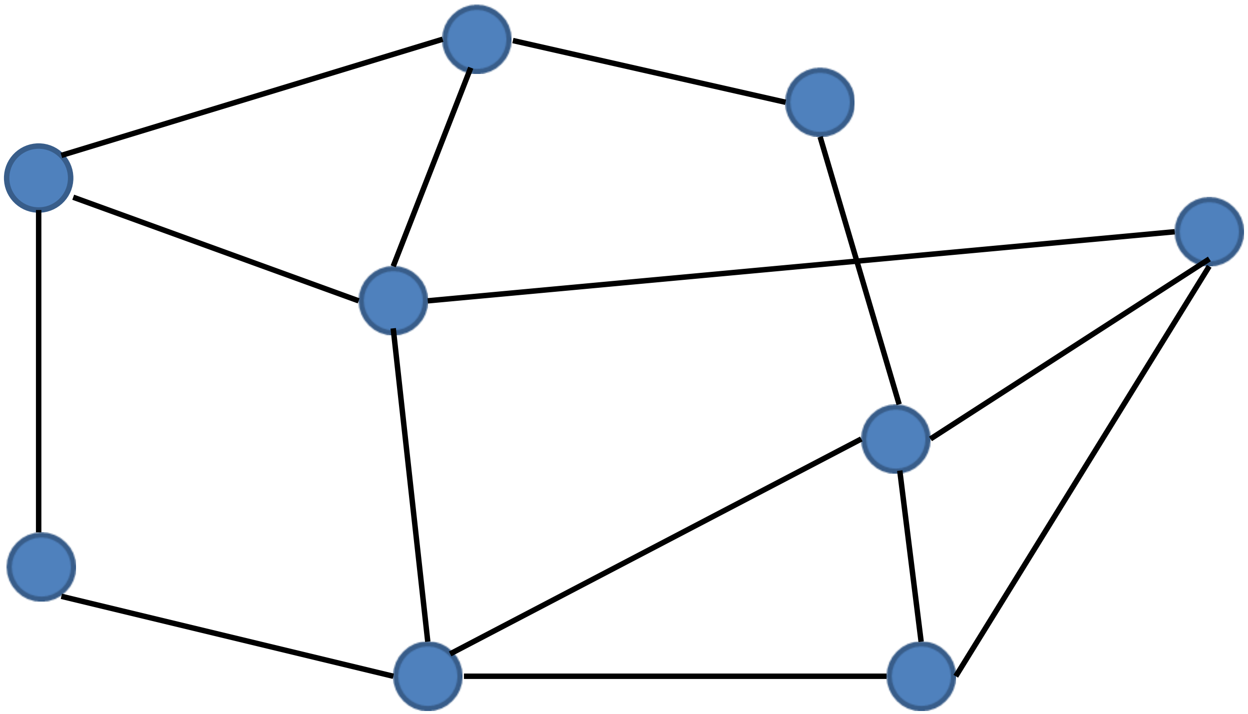
The union of the spanning tree and T-join is a connected Eulerian subgraph, hence a tour of length $n + O(|T|) + O(n/d)$.

Spanning tree with few odd degree vertices

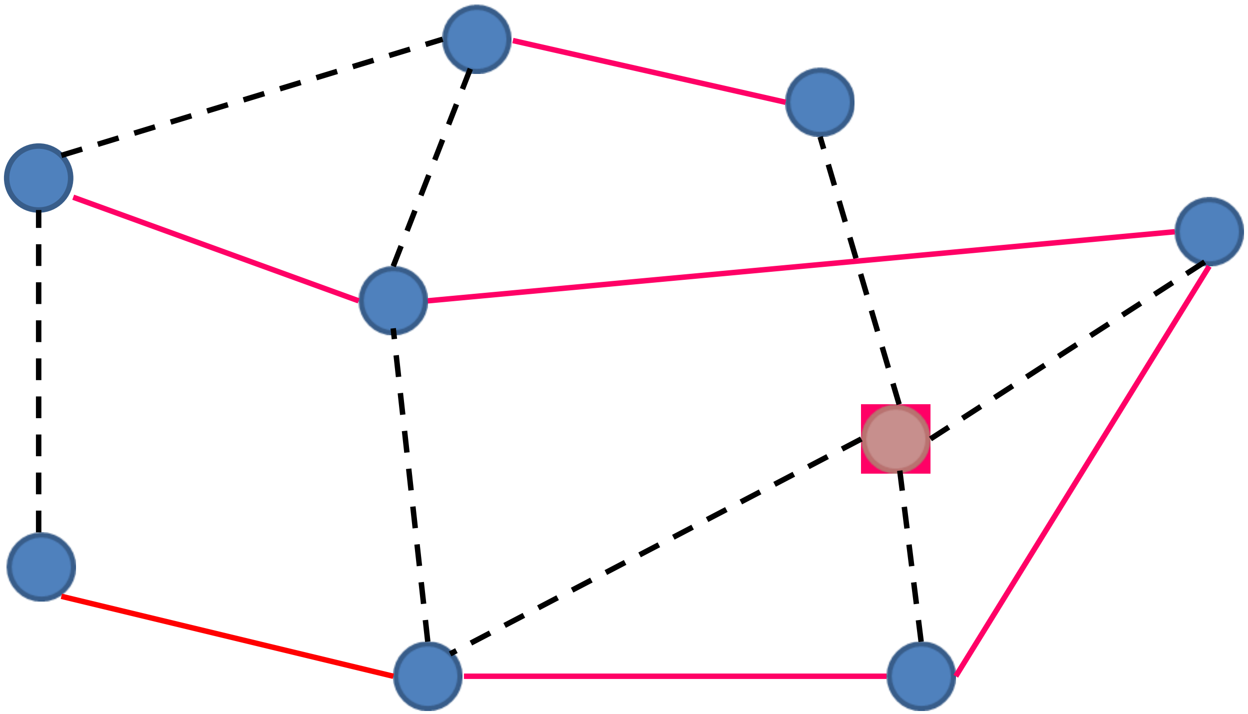
Thm: Every connected d-regular graph has a spanning tree with $O(\frac{n}{\sqrt{d}})$ odd degree vertices.

Proof approach: cover all vertices in G by a small number of paths (a spanning linear forest). Complete to a spanning tree arbitrarily.

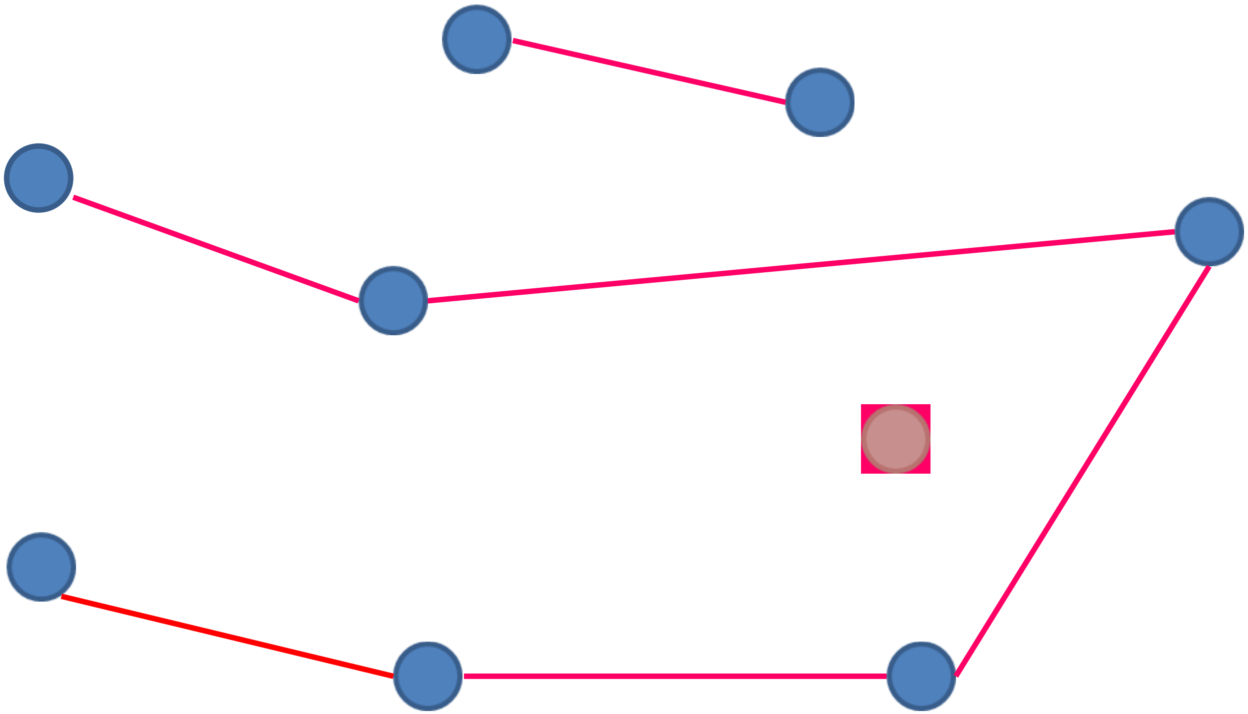
If the number of paths is P, then the number of odd degree vertices in the resulting spanning tree is at most 2P-2.



A spanning linear forest



A spanning linear forest



Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths

Path cover number

The fewest number of components in a linear forest is the path cover number of the graph.

Conjecture [Magnant and Martin, 2009]: the path cover number of a d-regular graph is at most $n/(d+1)$.

Proved for $d < 6$.

If true for all d, would imply a tour of length $(1 + O(1/d))n$. Better than what we know how to prove.

Linear arboricity conjecture

Arboricity – covering all edges by forests.

Linear arboricity – covering all edges by linear forests.

Conjecture [Akiyama, Exoo, Harary, 1981]: in d-regular graphs, $\left\lceil \frac{d+1}{2} \right\rceil$ linear forests suffice.

If true, one of these linear forests has at least n – O(n/d) edges, and hence at most O(n/d) components.

Results on linear arboricity conjecture

Alon, Teague and Wormald 2001 (see also Alon and Spencer): Linear arboricity is at most $\frac{d+\tilde{O}(d^{\frac{2}{3}})}{2}$.

Implies linear forest of size $\left(1-\tilde{O}\left(\left(\frac{1}{d}\right)^{\frac{1}{3}}\right)\right)n$, and

by our results, a tour of length $\left(1+\tilde{O}\left(\left(\frac{1}{d}\right)^{\frac{1}{3}}\right)\right)n$.

Improved bounds

Thm: every d-regular graph has a path cover with $O(\frac{n}{\sqrt{d}})$ paths.

Corollary: every d-regular graph has a tour of length $\left(1 + O(\frac{1}{\sqrt{d}})\right)n$.

Inductive construction of a path cover

Single vertex v: any two of its edges can be used as part of a path.

Path between s and t: only one edge from each endpoint can be used as part of a longer path.

Cannot contract the path to a single vertex.

Inductive construction of a path cover
Easier for directed graphs

Path between s and t: only one edge from each endpoint can be used as part of a longer path.

Cannot contract the path to a single vertex.

If edges are oriented, directed path from s to t can be contracted to single vertex, leaving incoming edges to s and outgoing edges from t.

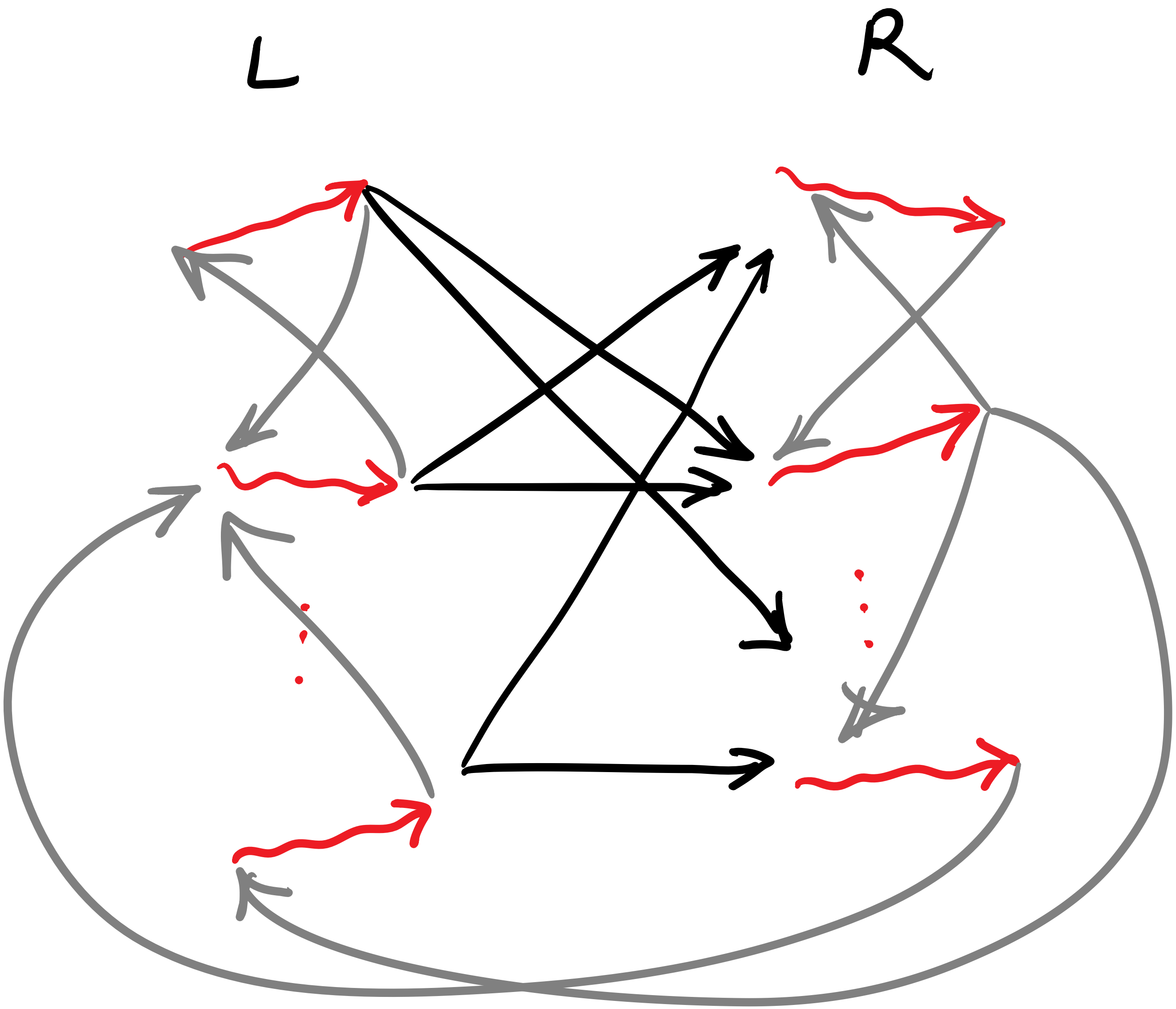
Orienting the edges

- Suppose for simplicity that d is even.
- Take an Euler tour through all edges.
- Orient the edges according to tour.
- Gives a directed graph with in and out degrees $d/2$.

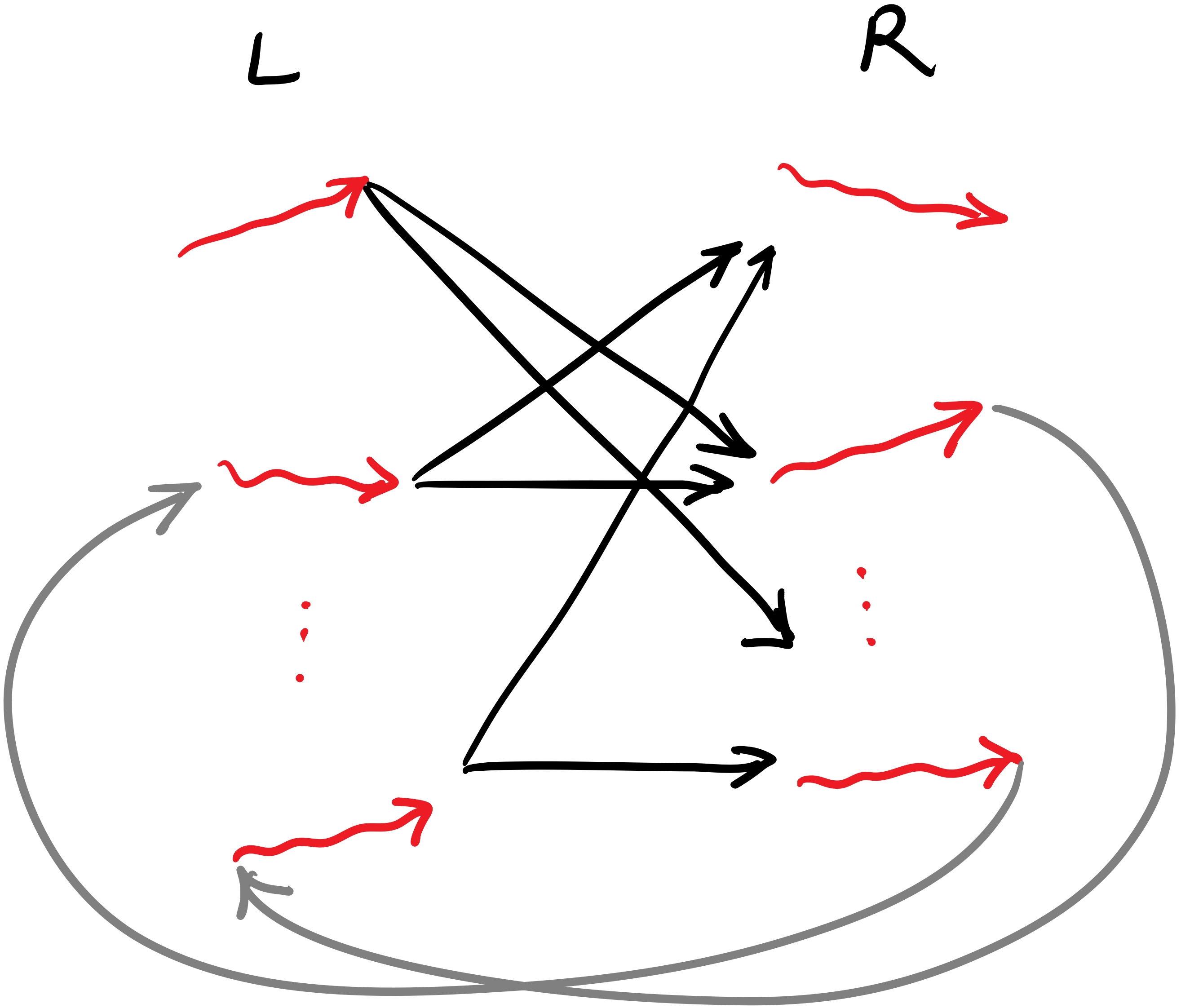
Algorithm Sketch: One phase

- Pair up nodes arbitrarily
- Put the two nodes of a pair in (L,R) or (R,L) respectively at random
- Find max matching of directed edges from L to R; Use “dummy” edges to complete to a perfect matching M
- Delete non matching arcs from L to R and all arcs within L and within R
- Use matching arc to “extend” paths and reduce number of paths by half

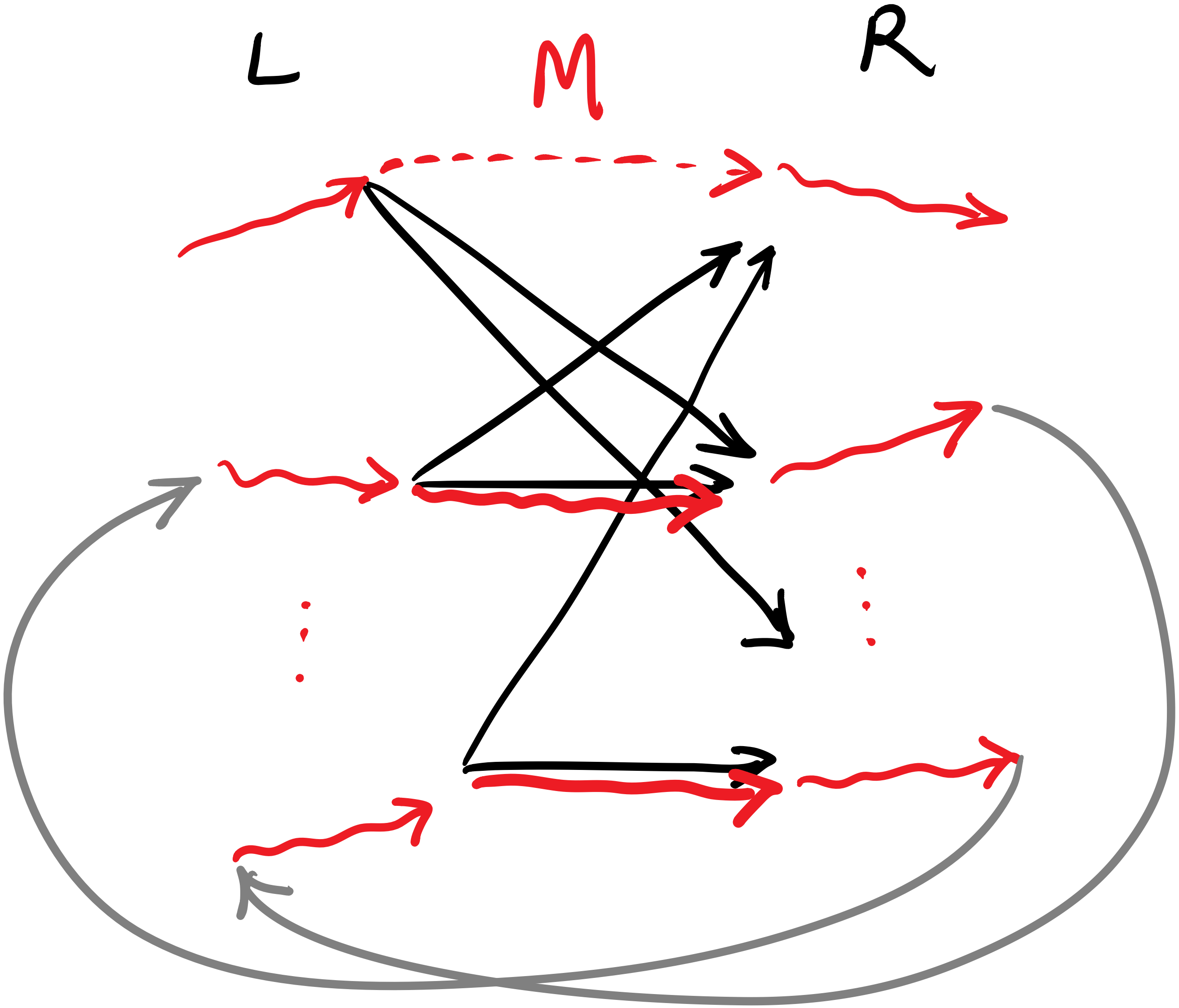
Algorithm Sketch: One phase



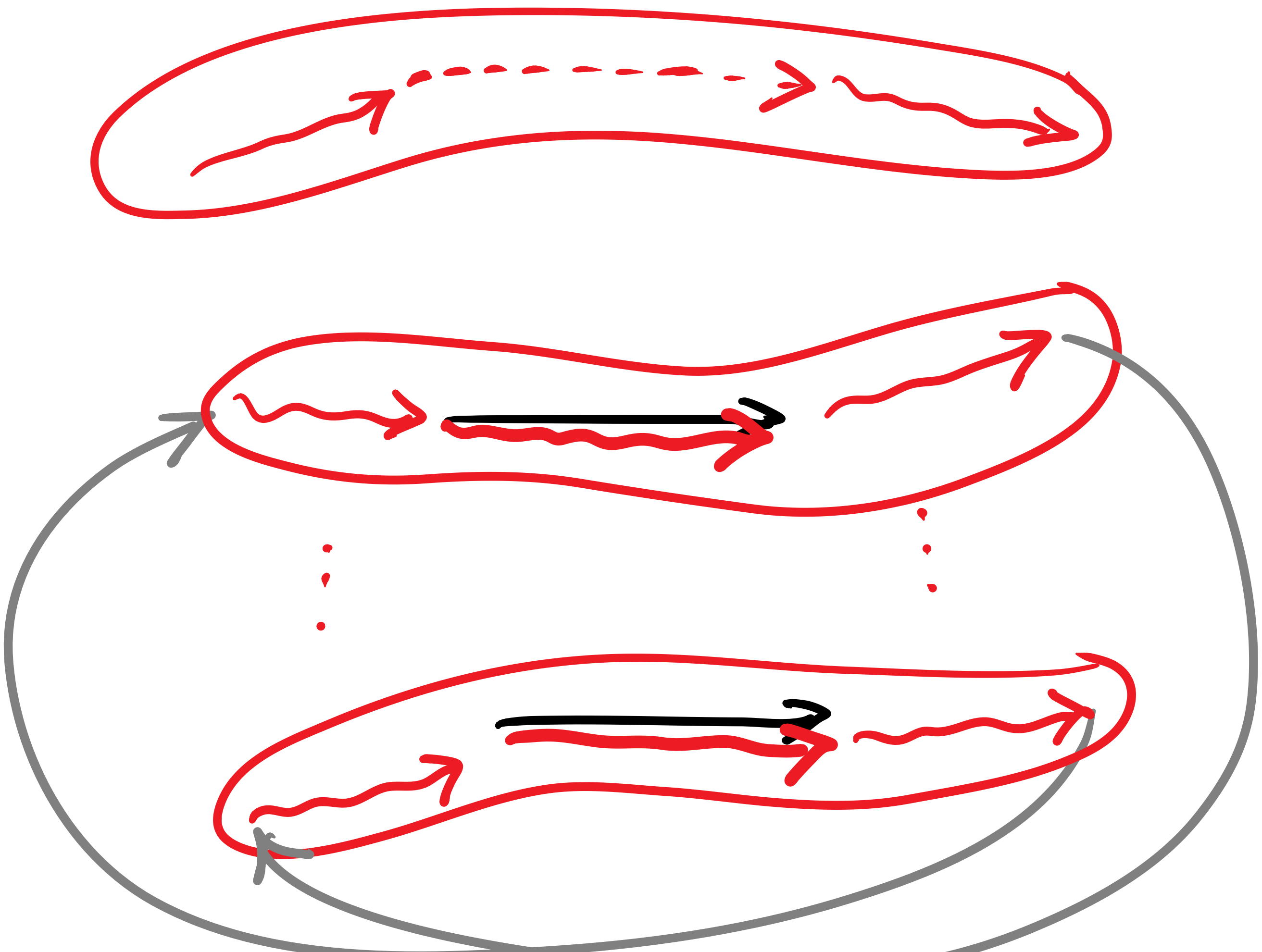
Algorithm Sketch: One phase



Algorithm Sketch: One phase



Algorithm Sketch: One phase



Algorithm: Analysis Sketch

- Assume n is power of 2 → vertex in phase t is a path of 2^t nodes; Exactly log n phases
- Show that max matching in phase t is of size ≥ (1 – err(t)) (n/2^t)

Size of final linear forest $\geq \sum_t |M_t|$

$$\geq \sum_t \left(1 - err(t)\right)\left(\frac{n}{2^t}\right)$$

$$\geq n - n \sum_t \frac{err(t)}{2^t}$$

Key technical claim: $err(t) \leq c 2^{\frac{t}{2}} \frac{\sqrt{\log d}}{\sqrt{d}}$

Final linear forest size $\geq n - n \sum_t \frac{err(t)}{2^t} \geq n - cn \frac{\sqrt{\log d}}{\sqrt{d}} \sum_t \frac{1}{2^{\frac{t}{2}}}$

$$\geq n - c' n \frac{\sqrt{\log d}}{\sqrt{d}}$$

Summary of our approach

- Every d-regular digraph can be covered by $O(\frac{n}{\sqrt{d}})$ paths. (via improved algorithm)
- Every d-regular graph has a spanning tree with $|\mathrm{T}| = O(\frac{n}{\sqrt{d}})$ odd degree vertices.
- There is a T-join with $O(|T|) + O(\frac{n}{d})$ edges.
- There is a tour of length $\mathrm{n} + O(\frac{n}{\sqrt{d}})$.

More generally

Let G be a connected n-vertex graph with max degree Δ, avg degree d, min degree δ. Then there is a tour of length

$$\left(1+\frac{\Delta-d}{\Delta}+O\left(\frac{1}{\sqrt{\Delta}}\right)+O\left(\frac{1}{\delta}\right)\right)n$$

Moreover, such a tour can be found in random polynomial time.

Results

1. Size 4n/3 in a graph with a spanning tree and a simple cycle on its odd nodes

(WG14, joint with Satoru Iwata and Alantha Newman)

"Regular graphs have short tours"

2. Size 9n/7 in cubic bipartite graphs

(APPROX14, joint with Jeremy Karp)

3. Size $(1 + O(\frac{1}{\sqrt{d}}))$n in d-regular graphs

(IPCO14, joint with Uri Feige and Mohit Singh)

Open

- Use Steiner cycles approach for graph-TSP
- Prove Barnette's conjecture
- Improve additive bound in d-regular graphs from $O\left(\frac{n}{\sqrt{d}}\right)$ to $O\left(\frac{n}{d}\right)$
- 4/3-approximation for general graph-TSP

Main ideas for short tours

1. Augment spanning tree with carefully chosen edges
2. Delete carefully chosen edges from the whole graph
3. Augment cycle cover with few cycles
4. Augment path cover with few paths