Approximation Algorithms for Graphic TSP in Cubic, Bipartite Graphs

Jeremy Karp
Joint work with R. Ravi

Carnegie Mellon University

Workshop on Flexible Network Design, July 2013

Background

- Graphic TSP is the Traveling Salesman Problem on shortest path metrics of undirected graphs with unit edge-lengths
- Equivalently, given an undirected, unweighted graph, G, find a spanning Eulerian subgraph with the fewest edges in $2G$
- Graphic TSP in cubic graphs captures much of the complexity of the problem in general graphs while admitting approximation algorithms with improved guarantees

Background

- Graphic TSP is the Traveling Salesman Problem on shortest path metrics of undirected graphs with unit edge-lengths
- Equivalently, given an undirected, unweighted graph, G, find a spanning Eulerian subgraph with the fewest edges in $2G$
- Graphic TSP in cubic graphs captures much of the complexity of the problem in general graphs while admitting approximation algorithms with improved guarantees

Background

- Graphic TSP is the Traveling Salesman Problem on shortest path metrics of undirected graphs with unit edge-lengths
- Equivalently, given an undirected, unweighted graph, G, find a spanning Eulerian subgraph with the fewest edges in $2G$
- Graphic TSP in cubic graphs captures much of the complexity of the problem in general graphs while admitting approximation algorithms with improved guarantees

Previous Results

- $\frac{7}{5}$-approximation for general graphs (Sebő and Vygen [2012])
- $\frac{4}{3}$-approximation for cubic and subcubic graphs (Boyd, Sitters, van der Ster, and Stougie [2011]; Mömke and Svensson [2011])
- $(\frac{4}{3} - \frac{1}{61236})$-approximation for 2-edge-connected cubic graphs (Correa, Larré, and Soto [2012])
- $(\frac{4}{3} - \frac{1}{108})$-approximation for 3-edge-connected, bipartite, cubic graphs (Correa, Larré, and Soto [to be published])

Previous Results

- $\frac{7}{5}$-approximation for general graphs (Sebő and Vygen [2012])
- $\frac{4}{3}$-approximation for cubic and subcubic graphs (Boyd, Sitters, van der Ster, and Stougie [2011]; Mömke and Svensson [2011])
- $(\frac{4}{3} - \frac{1}{61236})$-approximation for 2-edge-connected cubic graphs (Correa, Larré, and Soto [2012])
- $(\frac{4}{3} - \frac{1}{108})$-approximation for 3-edge-connected, bipartite, cubic graphs (Correa, Larré, and Soto [to be published])

Previous Results

- $\frac{7}{5}$-approximation for general graphs (Sebő and Vygen [2012])
- $\frac{4}{3}$-approximation for cubic and subcubic graphs (Boyd, Sitters, van der Ster, and Stougie [2011]; Mömke and Svensson [2011])
- $(\frac{4}{3} - \frac{1}{61236})$-approximation for 2-edge-connected cubic graphs (Correa, Larré, and Soto [2012])
- $(\frac{4}{3} - \frac{1}{108})$-approximation for 3-edge-connected, bipartite, cubic graphs (Correa, Larré, and Soto [to be published])

Previous Results

- $\frac{7}{5}$-approximation for general graphs (Sebő and Vygen [2012])
- $\frac{4}{3}$-approximation for cubic and subcubic graphs (Boyd, Sitters, van der Ster, and Stougie [2011]; Mömke and Svensson [2011])
- $(\frac{4}{3} - \frac{1}{61236})$-approximation for 2-edge-connected cubic graphs (Correa, Larré, and Soto [2012])
- $(\frac{4}{3} - \frac{1}{108})$-approximation for 3-edge-connected, bipartite, cubic graphs (Correa, Larré, and Soto [to be published])

Approach

- Find a cycle cover with k cycles
- Compress each cycle into a single node and find a spanning tree in this compressed graph
- In the original graph, add two copies of each edge in the spanning tree to the cycle cover to obtain a solution with $n + 2(k-1)$ edges
- Our algorithm finds a cycle cover with at most $\frac{3}{20}n$ cycles, giving us a solution with $\frac{13}{10}n - 2$ edges.

Approach

- Find a cycle cover with k cycles
- Compress each cycle into a single node and find a spanning tree in this compressed graph
- In the original graph, add two copies of each edge in the spanning tree to the cycle cover to obtain a solution with $n + 2(k-1)$ edges
- Our algorithm finds a cycle cover with at most $\frac{3}{20}n$ cycles, giving us a solution with $\frac{13}{10}n - 2$ edges.

Approach

- Find a cycle cover with k cycles
- Compress each cycle into a single node and find a spanning tree in this compressed graph
- In the original graph, add two copies of each edge in the spanning tree to the cycle cover to obtain a solution with $n+2(k-1)$ edges
- Our algorithm finds a cycle cover with at most $\frac{3}{20}n$ cycles, giving us a solution with $\frac{13}{10}n-2$ edges.

Approach

- Find a cycle cover with k cycles
- Compress each cycle into a single node and find a spanning tree in this compressed graph
- In the original graph, add two copies of each edge in the spanning tree to the cycle cover to obtain a solution with $n+2(k-1)$ edges
- Our algorithm finds a cycle cover with at most $\frac{3}{20}n$ cycles, giving us a solution with $\frac{13}{10}n-2$ edges.

New Result

Theorem

Given a cubic, bipartite graph G with n vertices, there is a polynomial time algorithm that computes a spanning Eulerian subgraph in $2G$ with at most $\frac{13}{10}n - 2$ edges.

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a “gadget”), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no “problematic” 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a "gadget"), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no "problematic" 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a "gadget"), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no "problematic" 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a “gadget”), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no “problematic” 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a "gadget"), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no "problematic" 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a “gadget”), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no “problematic” 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

How it Works

- All cycles in a bipartite graph are even
- We can bound the number of cycles in the cycle cover if we ensure it contains no 4-cycles and relatively few 6-cycles
- If our graph contains a 4-cycle or 6-cycle, we replace it with a different, more desirable subgraph (a "gadget"), maintaining cubic- and bipartite-ness
- Eventually we have a compressed graph with no "problematic" 4-cycles or 6-cycles
- Find a cycle cover in this compressed graph
- Expand the graph by swapping the 4- and 6-cycles back into the graph
- Add edges as we go, eventually obtaining a desirable cycle cover in the original graph

A Good Example

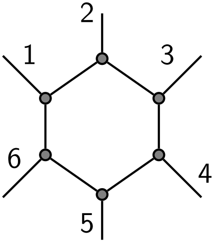


Figure : A 6-cycle

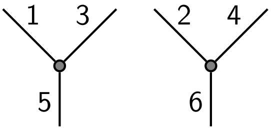


Figure : The gadget which replaces the 6-cycle

A Good Example

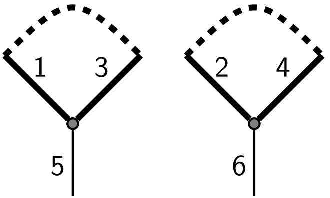


Figure : Part of the cycle cover in the compressed graph

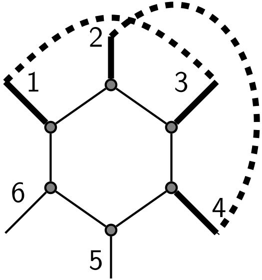


Figure : The same portion of the cycle cover, after expanding the graph

A Good Example

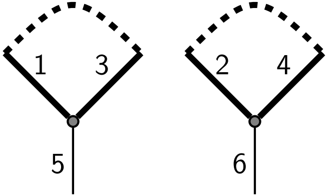


Figure : Part of the cycle cover in the compressed graph

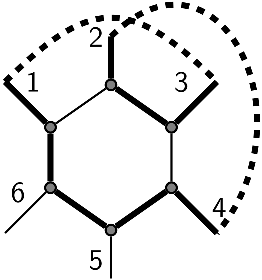


Figure : Then, we add edges to get a large cycle

A Bad Example

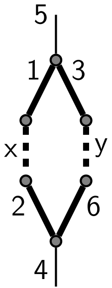


Figure : A cycle of length $x + y + 4$ that passes through a gadget that replaced a 6-cycle

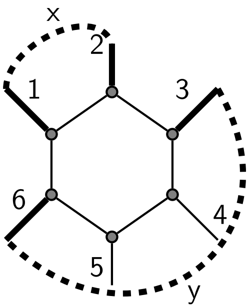


Figure : The cycle from the previous figure, after expanding the graph

A Bad Example

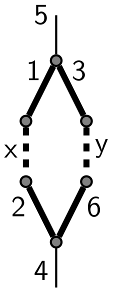


Figure : A cycle of length $x + y + 4$ that passes through a gadget that replaced a 6-cycle

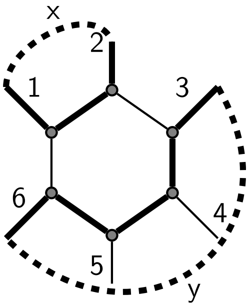


Figure : We add edges to get a cycle cover in the expanded graph, but we now have two smaller cycles of lengths $x + 3$ and $y + 5$

Handling Bad Cases

- We needed to use a few additional, more specialized gadgets in order to bound the number of 6-cycles we create as we expand the graph
- Below are the two main additional gadgets needed:

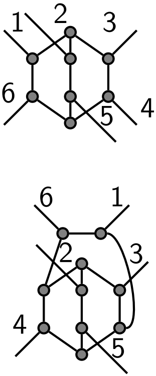


Figure : Two other “bad” subgraphs

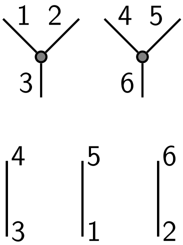
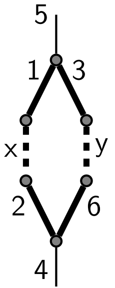
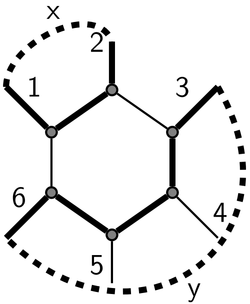


Figure : The gadgets which replace the “bad” subgraphs

Back to the Bad Example

- The additional gadgets ensure that the 6-cycle on the right is contracted only if $y \geq 3$.
- Now, this bad example can insert at most one 6-cycle, along with a longer cycle in the final cycle cover.





Future Directions

- Can we find better approximations in cubic, bipartite graphs? (Lower bound is $\frac{10}{9}n$)
- Ways to modify the algorithm so it is simpler, uses fewer gadgets
- Incorporate this technique with the local search methods used by Boyd, Sitters, van der Ster, and Stougie
- Is it possible to modify this algorithm to get a cycle cover with no 6-cycles? (implies a $\frac{5}{4}$-approximation)