6.889 — Lecture 15: Traveling Salesman (TSP)

Christian Sommer csom@mit.edu
(figures by Philip Klein)

November 2, 2011

Traveling Salesman Problem (TSP) given $G = (V, E)$ find a *tour* visiting each[1] node $v \in V$.
NP–hard optimization problem, hard even for planar graphs
Polynomial-time approximation for general graphs: Christofides' algorithm achieves $3/2$ approximation

Assumption (all of Lecture 15) undirected planar G, $\ell : E \to \mathbb{R}^+$

2**–approximation** simple algorithm, bound approximation ratio in terms of *minimum spanning tree*

- compute *minimum spanning tree* T. let $\ell(T) := \sum_{e \in T} \ell(e)$
- duplicate all edges $\rightsquigarrow$ *Eulerian* graph
- find Eulerian cycle, length at most $2\ell(T)$
- (if G is the complete graph K_n, Eulerian cycle can be converted into Hamiltonian cycle by skipping already visited nodes)

any tour needs to visit all nodes, total length at least $\ell(T)$, hence 2–approximation

Recall: Linear-Time Approximation Schemes for Planar Graphs (L. 8)

Example min VERTEXCOVER

Algorithm given G and approximation parameter $\epsilon \in (0, 1)$, let $k = 1/\epsilon$

1. BFS in G
2. $G_{ij} \leftarrow$ graph induced by $k + 1$ BFS levels $jk + i$ to $(j + 1)k + i$ (*shift* i, $0 \leqslant i < k$, and *slice* j)
3. $S_{ij} \leftarrow$ min VERTEXCOVER of G_{ij} (dynamic programming on graph with tree-width $\mathcal{O}(k)$)
4. $S_i \leftarrow \bigcup_j S_{ij}$
5. RETURN best S_i (best shift i, $0 \leqslant i < k$, smallest $|S_i|$)

Running Time dynamic program runs in time $2^{\mathcal{O}(k)}|V(G_{ij})|$, overall $2^{\mathcal{O}(k)}n$

Correctness and Approximation Ratio two properties used

3. part of OPT in G_{ij} is a feasible solution for G_{ij}. $\rightsquigarrow$ consequence: $|\mathsf{OPT} \cap V(G_{ij})| \geqslant |S_{ij}|$
   optimum solution OPT induces solution on subgraph G_{ij}
   for at least one shift i overlap $|\mathsf{OPT}_i|$ is small[2] ($\leqslant |\mathsf{OPT}|/k = \epsilon|\mathsf{OPT}|$)
4. solutions in G_{ij} together form a feasible solution for G, $\bigcup_j S_{ij}$ is a solution for G (for any i)

[1] visiting only a subset $U \subseteq V$ to be discussed in Lectures 16 and 17
[2] define $\mathsf{OPT}_i = \mathsf{OPT} \cap \{\text{all nodes on BFS level } i \mod k\}$

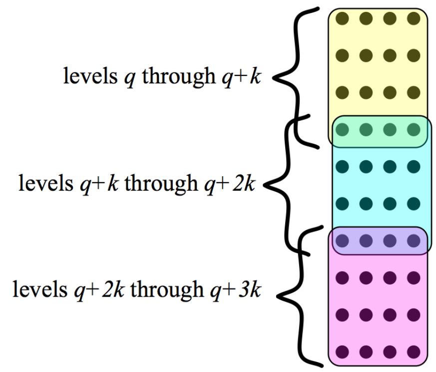

Figure 1: Consecutive slices (subgraphs) overlap by one single level (boundary).

Apply framework to TSP?

Approximation Ratio two properties used

✗ 3. part of OPT in G_{ij} is not necessarily a feasible solution for G_{ij}, maybe not a tour!
optimum solution does not induce a solution on subgraph G_{ij}

✓ 4. solutions in G_{ij} form a feasible solution for G, $\bigcup_j S_{ij}$ can be combined into a solution for G (for any i)
ok for TSP since tour visits *all* nodes $\rightsquigarrow$ also visits *all* boundary nodes $\rightsquigarrow$ can connect subtours

Modified framework for unit-length graphs

Algorithm given G and approx. parameter $\epsilon \in (0, 1)$, let $k = 3 \cdot 2/\epsilon$ (factor 3 from Euler's $|E| \leqslant 3|V| - 6$)

1. BFS in G^* (determine levels in dual, conn. components of G_{ij} have cycle boundary, cut–cycle duality)
2. $G_{ij} \leftarrow$ graph induced by $k+1$ BFS levels $jk+i$ to $(j+1)k+i$ (*shift* i, $0 \leqslant i < k$, and *slice* j)
choose shift i such that number of edges in level $i \mod k$ is at most $|E|/k \leqslant \epsilon|V|/2$
3. $S_{ij} \leftarrow$ TSP of each component of G_{ij} (dynamic programming on graph(s) with tree-width $\mathcal{O}(k)$)
4. $S_i \leftarrow \bigcup_j S_{ij}$, "find Eulerian cycle"
5. Return best S_i (best shift i, $0 \leqslant i < k$, however, we already chose good i in Step 2)

Approximation Ratio different property used

3. tour in G_{ij} may be longer than OPT $\cap\, G_{ij}$
how much longer? *cannot* bound distortion by $\epsilon\ell(\text{OPT} \cap G_{ij})$, error is not *multiplicative* but *additive*
no bound on overlap with OPT. let ∂G_{ij} be boundary of G_{ij}. then TSP$(G_{ij}) \leqslant \ell(\text{OPT} \cap G_{ij}) + \ell(\partial G_{ij})$
4. $\ell\left(\bigcup_j S_{ij}\right) \leqslant \ell(\text{OPT}) + 2 \cdot \ell(\partial G_i)$, where $\partial G_i := \bigcup_j \partial G_{ij}$ is total boundary
unit-length: total boundary is at most k–fraction, $\ell(\partial G_i) = |\partial G_i| \leqslant |E|/k \leqslant \epsilon|V|/2$, total error term is $\epsilon|\text{OPT}|$ since $|V| \leqslant$ OPT
arbitrary lengths? $\ell(\partial G_i) \leqslant \ell(E)/k$, no connection to $\ell(\text{OPT})$

Spanner

Problem if graph G has edge lengths $\ell : E \to \mathbb{R}^+$, boundary cycles may be too long, solution in G_{ij}?

Algorithm given G and approx. parameter $\epsilon \in (0,1)$, let $k = 18/\epsilon^2$ (factor is *not* from Euler's formula)

0. compute spanner (sparsify G, obtain subgraph H)
1. BFS in H^*

   ...

Goal sparse subgraph of G, called *spanner* H, shall have two properties

- total length of edges $V(H)$ proportional to total length of minimum spanning tree in G (trivial for unit-length graphs, since $|E| \leqslant 3|V| - 6$; for arbitrary non-negative lengths we obtain total length $\leq (1+2/\epsilon)\ell(\text{MST})$, factor $3/\epsilon$)
  $\rightsquigarrow$ can bound total length of edges in the overlap/boundary
- distances preserved: $d_H(v,w) \leqslant (1+\epsilon)d_G(v,w)$ (cf. *stretch* in Lecture 13)
  $\rightsquigarrow$ paths and distances not distorted by too much, $(1+\epsilon')^2 \leqslant (1+\epsilon)$ for $\epsilon' = \epsilon/3$, factor 3

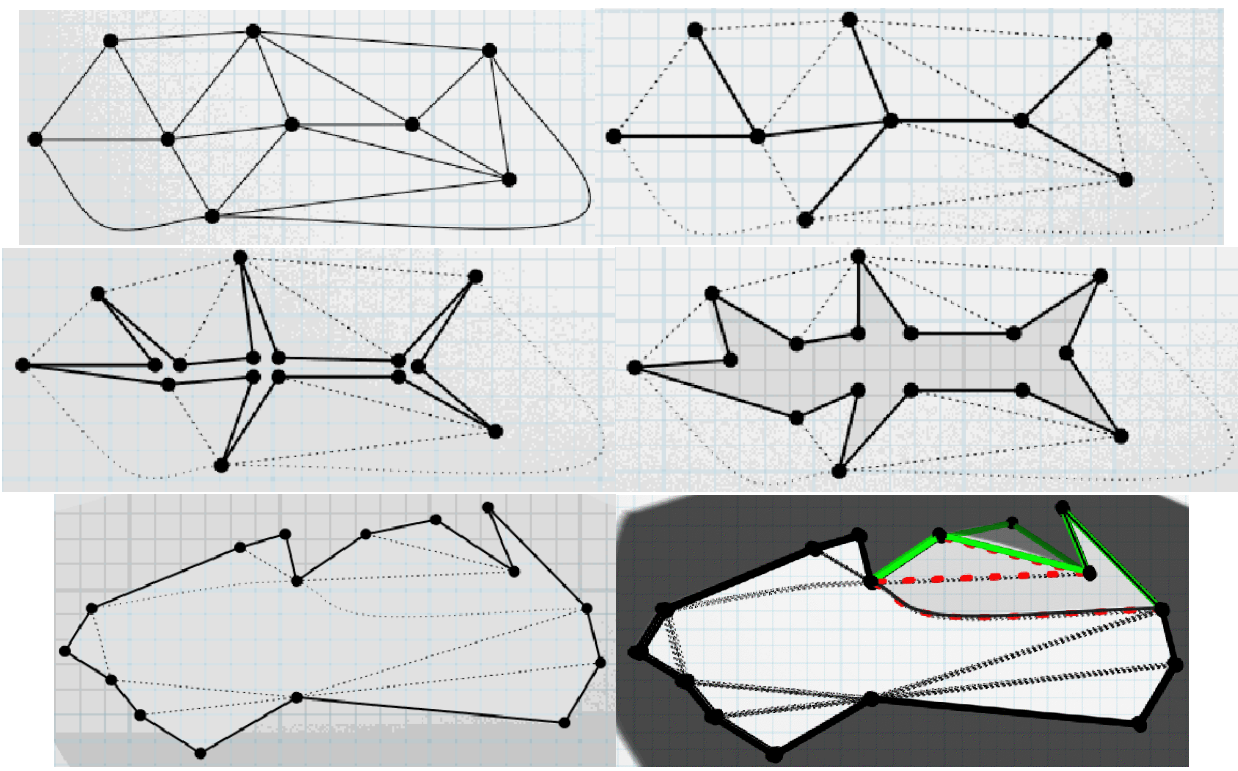

Figure 2: Spanner construction: *(i)* given an undirected planar graph *(ii)* compute a minimum spanning tree, *(iii)* "cut" along the tree to *(iv)* create a face, *(v)* make it the infinite face (conceptual step only, not actually necessary), and *(vi)* add or discard non-tree edges, starting with the leafs of the interdigitating tree.

Spanner analysis total length of spanner edges depends on ϵ and total length of minimum spanning tree. maintain cycle C (initialized by walk around spanning tree, total length twice spanning tree, $2\ell(\text{MST})$). edge $e = uv$ is added if shortest path in cycle is not a $(1+\epsilon)$–approximation, $d_C(u,v) > (1+\epsilon)\ell(u,v)$. if e is added, the total length of C decreases by at least $\epsilon\ell(u,v)$. cycle length can never be negative $\rightsquigarrow$ total length of edges added to the spanner is at most $2\ell(\text{MST})$, hence total length is at most $(1+2/\epsilon)\ell(\text{MST})$

Different perspective: contraction decomposition

different view on framework: after spanner step, slicing is *deletion decomposition* (cf. Lec. 9) in the dual (label/partition edges into $E_1, E_2, \dots$ s.t. deleting any part E_i results in a graph with small tree-width). deletion in dual ⇔ contraction in primal. ⇝ *contraction decomposition*

0. compute spanner H
1. BFS in H^*
2. contract all[3] edges on level $i \mod k$ (for best i)
3. compute solution in contracted graph (dynamic programming on graph with tree-width $\mathcal{O}(k)$)
4. lift solution (can augment solution since total boundary is small)
5. return best S_i (for TSP we already chose good i in Step 2)

[3] except for self-loops

References

The Traveling Salesman Problem is **NP**–hard even for planar graphs [GJT76]. The linear-time approximation scheme for TSP is by Klein [Kle08] (earlier algorithms in [GKP95, AGK+98]). A variant (different spanner needed) works for *Subset* TSP [Kle06]. For general undirected graphs, algorithms achieve approximation ratio roughly $3/2$ [Chr76, MS11, GSS11, Muc12]. The existence of a polynomial-time algorithm with approximation ratio better than $220/219$ would imply $\mathbf{P} = \mathbf{NP}$ [PV06].

[AGK+98] Sanjeev Arora, Michelangelo Grigni, David R. Karger, Philip N. Klein, and Andrzej Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In *9th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 33–41, 1998.

[Chr76] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[GJT76] Michael R. Garey, David S. Johnson, and Robert Endre Tarjan. The planar Hamiltonian circuit problem is **NP**–complete. *SIAM Journal on Computing*, 5(4):704–714, 1976.

[GKP95] Michelangelo Grigni, Elias Koutsoupias, and Christos H. Papadimitriou. An approximation scheme for planar graph TSP. In *36th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 640–645, 1995.

[GSS11] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011. To appear, preprint online: `http://www.stanford.edu/~saberi/tsp.pdf`.

[Kle06] Philip N. Klein. A subset spanner for planar graphs, with application to subset TSP. In *38th ACM Symposium on Theory of Computing (STOC)*, pages 749–756, 2006.

[Kle08] Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM Journal on Computing*, 37(6):1926–1952, 2008. Announced at FOCS 2005.

[MS11] Tobias Mömke and Ola Svensson. Approximating graphic TSP by matchings. In *52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011. To appear, preprint available online: `http://arxiv.org/abs/1104.3090`.

[Muc12] Marcin Mucha. 13/9–approximation for graphic TSP. In *29th Symposium on Theoretical Aspects of Computer Science (STACS)*, 2012. To appear, preprint available online at `http://arxiv.org/abs/1108.1130`.

[PV06] Christos H. Papadimitriou and Santosh Vempala. On the approximability of the traveling salesman problem. *Combinatorica*, 26(1):101–120, 2006. Announced at STOC 2000.