Combinatorial Optimization

Zdeněk Hanzálek
hanzalek@fel.cvut.cz

CTU FEE Department of Control Engineering

May 3, 2013

European Social Fund Prague & EU: We invests in your future.

Traveling Salesman Problem

Zdeněk Hanzálek
hanzalek@fel.cvut.cz

CTU FEE Department of Control Engineering

April 16, 2013

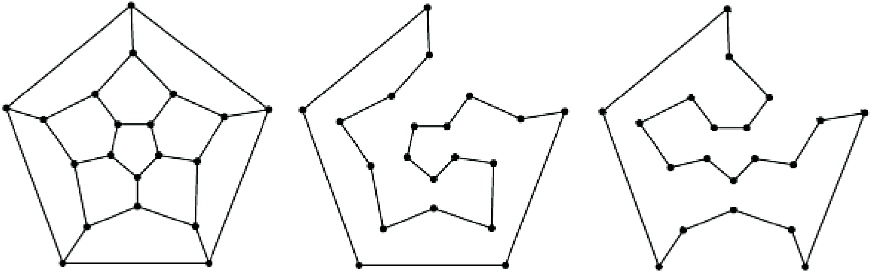
Related Decision Problem

Existence of Hamiltonian Circuit (HC)

- **Instance:** Undirected graph G.
- **Goal:** Decide if Hamiltonian circuit (circuit visiting every node exactly once) exists in graph G.

- NP-complete problem
- The directed version of this problem is: (**Hamiltonian cycle**) for a directed graph
- HC belongs to NP problems. For each yes-instance G we take any Hamiltonian circuit of G as a certificate. To check whether a given edge set is in fact a Hamiltonian circuit of a given graph is obviously possible in polynomial time.

Hamilton's Puzzle



The Hamiltonian circuit is named after William Rowan Hamilton who invented the Icosian game, now also known as Hamilton's puzzle, which involves finding a Hamiltonian circuit in the edge graph of the dodecahedron. (picture can be viewed as a look inside the dodecahedron through one of its **twelve faces**). Like all platonic solids, the dodecahedron is Hamiltonian.

Problem Formulation

Traveling Salesman Problem - TSP

- **Instance:** A complete undirected graph K_n ($n \geq 3$) and weights $c : E(K_n) \rightarrow \mathbb{Q}_0^+$.
- **Goal:** Find a Hamiltonian circuit T whose weight $\sum_{e \in E(T)} c(e)$ is minimum.

- Nodes correspond to cities and weights to distances or travel costs
- This problem is called **symmetric TSP**, since it is given by a complete undirected graph
- If the distance form city A to city B differs form the one from B to A, we have to use a directed graph and we deal with an **asymmetric TSP**

Strongly NP-hard Problems

Let L be an optimization problem.
For a polynomial p let L_p be the restriction of L to such instances x that consist of nonnegative integers with $largest(x) \leq p(size(x))$, i.e. numerical parameters of L_p are bounded by a polynomial in the length of the input.

L is called **strongly NP-hard** if there is a polynomial p such that L_p is NP-hard.

If L is strongly NP-hard, then L **cannot be solved by a pseudopolynomial** time algorithm unless P = NP.

In the following we will study the case, where:

- L ... **TSP**
- L_p ... **TSP** with restriction $c(e) \in \{1, 2\}$

TSP Complexity and Likely Nonexistence of Pseudopolynomial Algorithm

Proposition

TSP is strongly NP-hard.

Proof: We show that the **TSP** is NP-hard even when restricted to instances where all distances are 1 or 2 using polynomial transformation from the **HC** problem

- Let G be an undirected graph in which we want to find the Hamiltonian circuit.
- Create a **TSP** instance such that every node from G is associated to one node in the complete undirected graph K_n. Weight of $\{i,j\}$ in K_n equals:

$$c(\{i,j\}) = \begin{cases} 1 & \textit{if } \{i,j\} \in E(G); \\ 2 & \textit{if } \{i,j\} \notin E(G). \end{cases}$$

- G has a Hamiltonian circuit iff optimal **TSP** solution is equal to n.

Likely Nonexistence of r-approximation Algorithm for General TSP

Theorem

If we believe P≠NP, than there is no r-approximation algorithm for **TSP** for r ≥ 1.

Proof by contradiction:
Assume there exists a polynomial r-approximation algorithm $\mathcal{A}$ for **TSP**.
We further show that we can solve the **HC** problem while using such an "inaccurate" algorithm $\mathcal{A}$.
Since **HC** is NP-complete, P=NP.

In other words: if there exists a polynomial r-approximation algorithm $\mathcal{A}$ solving **TSP**, then the NP-complete **HC** problem can be solved in polynomial time by $\mathcal{A}$.

Likely Nonexistence of r-approximation Algorithm for General TSP

Every **HC** instance can be polynomially reduced to a TSP instance "inaccurately" solved by r-approximation algorithm $\mathcal{A}$:

- Let G be an undirected graph in which we want to find the Hamiltonian circuit.
- Create a **TSP** instance such that every node from G is associated to one node (city) in the complete undirected graph K_n. Weight (distance) of $\{i, j\}$ in K_n equals:

$$c(\{i,j\}) = \begin{cases} 1 & \textit{if } \{i,j\} \in E(G); \\ 2 + (r-1) * n & \textit{if } \{i,j\} \notin E(G). \end{cases}$$

- We use $\mathcal{A}$ to solve the instance.
  - if the result is equal to n, then the Hamiltonian circuit exists,
  - otherwise the result is greater or equal to $(n-1) + 2 + (r-1) * n = r * n + 1$.
    Because $\mathcal{A}$ is an r-approximation algorithm, $r * n + 1 \leq r * OPT(K_n, c)$.
    Hence $OPT(K_n, c) > n$ shows that G has no Hamiltonian circuit.

Triangle Inequality

In most common applications the distances of the **TSP** satisfies the triangle inequality.

Metric TSP

- **Instance:** Complete undirected graph K_n $(n \geq 3)$ with weights $c : E(K_n) \rightarrow \mathbb{R}_0^+$ such that $c(\{i,j\}) + c(\{j,k\}) \geq c(\{k,i\})$ for all $i,j,k \in V(K_n)$.
- **Goal:** Find the Hamiltonian circuit T such that $\sum_{e \in E(T)} c(e)$ is minimal.

- The **metric TSP** is strongly NP-hard. Can be proved in the same way as the complexity of **TSP** because weights 1 and 2 preserve the triangle inequality. Therefore the psudopolynomial algorithms do not exist.
- But approximation algorithms do exist.

Nearest Neighbor - Heuristic Algorithm

Input: An instance (K_n, c) of **metric TSP**.
Output: Hamiltonian circuit H.
Choose arbitrary node $v_{[1]} \in V(K_n)$;
for $i := 2$ **to** n **do**
  choose $v_{[i]} \in V(K_n) \setminus \{v_{[1]}, \cdots, v_{[i-1]}\}$ such that $c(\{v_{[i-1]}, v_{[i]}\})$ is minimal;
end
Hamiltonian circuit H is defined by the sequence $\{v_{[1]}, \cdots, v_{[n]}, v_{[1]}\}$;

- The nearest unvisited city is chosen in each step
- This is not an approximation algorithm
- Time complexity is $O(n^2)$

Double-tree Algorithm

Input: An Instance (K_n, c) of the **metric TSP**.
Output: Hamiltonian circuit H.

1. Find a **minimum weight spanning tree** T in K_n;
2. By **doubling every edge** in T we get multigraph in which we find the **Euler tour** L;
3. **Transform the Euler tour** L **to the Hamiltonian circuit** H **in the complete graph** K_n:
   - create a sequence of nodes on the Euler tour L;
   - we **skip nodes that are already in the sequence**;
   - the rest creates the Hamiltonian circuit H;

Double-tree Algorithm is 2-approximation Algorithm

Time complexity is $O(n^2)$
It is a 2-approximation algorithm for the **metric TSP**:

- 1. due to the triangle inequality, the skipped nodes don't prolong the route, i.e. $c(E(L)) \geq c(E(H))$
- 2. while deleting one edge in the circuit, we create the tree. Therefore, inequality $OPT(K_n, c) \geq c(E(T))$ holds
- 3. $2c(E(T)) = c(E(L))$ holds due to the creation of L by doubling edges in T
- above points imply $2OPT(K_n, c) \geq c(E(H))$ since: $2OPT(K_n, c) \overset{2.}{\geq} 2c(E(T)) \overset{3.}{=} c(E(L)) \overset{1.}{\geq} c(E(H))$

Christofides' Algorithm [1976]

Input: An instance (K_n, c) of **metric TSP**.
Output: Hamiltonian circuit H.

1. Find a minimum weight spanning tree T in K_n;
2. Let W be the set of vertices having an **odd degree** in T;
3. Find a **minimum weight matching** M of nodes from W in K_n;
4. Merge of T and M forms a multigraph $(V(K_n), E(T) \cup M)$ in which we find the Eulerian circuit L;
5. Transform the Eulerian circuit L into the Hamiltonian circuit H in the complete graph K_n ;

Observation: Each edge connects 2 nodes $\Rightarrow$ the sum of the degree of all nodes is $2|E|$ $\Rightarrow$ there are an even number of nodes with an odd degree in every graph (and an arbitrary number of nodes with an even degree).
With respect to the previous observation and completeness of K_n, it follows that it is possible to find the perfect matching.

Christofides' Algorithm is a $\frac{3}{2}$ Approximation

Time complexity is $O(n^3)$
It is a $\frac{3}{2}$ approximation algorithm for the **metric TSP**:

- 1. due to the triangle inequality the skipped nodes do not prolong the route, i.e $c(E(L)) \geq c(E(H))$
- 2. while deleting one edge in the circuit, we create the tree. Therefore, inequality $OPT(K_n, c) \geq c(E(T))$ holds
- 3. since the perfect matching M considers every second edge in the alternating path and being the minimum weight matching it chooses the smaller half, $\frac{OPT(K_n,c)}{2} \geq c(M)$ holds
- 4. due to the construction of L it holds $c(M) + c(E(T) = c(E(L))$
- hence 2.,3.,4. and 1. imply $\frac{3}{2} OPT(K_n, c) \geq c(E(H))$

Tour Improvement Heuristics - Local Search *k*-OPT

One of the most successful techniques for **TSP** instances in practice.
A simple idea which can be used to solve other optimization problems as well:

- Find any Hamiltonian circuit by some heuristic
- Improve it by "local modifications" (for example: delete 2 edges and reconstruct the circuit by some other edges).

Local search is an algorithmic principle based on two decisions:

- Which modifications are allowed
- When to modify the solution (one possibility is, to allow improvements only)

Example of local search is *k*-OPT algorithm for TSP

k-OPT algorithm for TSP

Input: An instance (K_n, c) of **TSP**, number $k \geq 2$.
Output: Hamiltonian circuit H.
1. Let H be any Hamiltonian circuit;
2. Let $\mathcal{S}$ be the family of k-element subsets of $E(H)$;

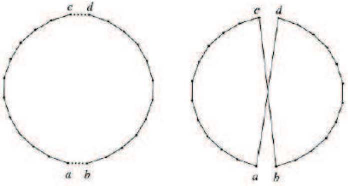
for all $S \in \mathcal{S}$ **do**
    for all Ham.circuits $H' \neq H$ such that $E(H') \supseteq E(H) \setminus S$ **do**
        if $c(E(H')) < c(E(H))$ **then** $H := H'$ a **go to** 2.;
    end
end
Stop

Note:

- H' is constructed, so that it is Hamiltonian circuit as well.
- When $k = 2$, the inner loop, which creates the Hamiltonian circuits H' from the remaining edges of H, executes only once, since there is just one way to construct the new Hamiltonian circuit.

Examples of 2-opt and 3-opt for TSP

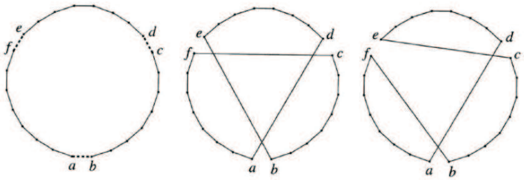
2-opt



just one way to construct the new Hamiltonian circuit:

- the gain if the improvement is: $c(E(H')) - c(E(H)) =$ (a,d)+(b,c)-(c,d)-(a,b) and path (b,...,d) has changed orientation

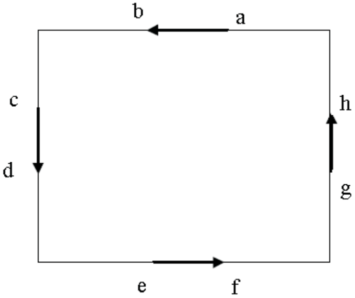
3-opt

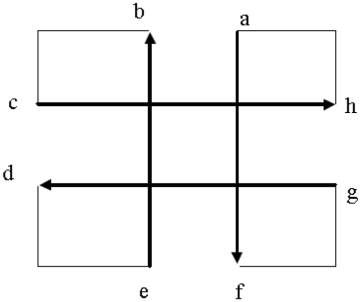


two ways to construct the new Hamiltonian circuit:

- $c(E(H')) - c(E(H)) =$ (a,d)+(e,b)+(c,f)-(a,b)-(c,d)-(e,f) no path has changed orientation
- $c(E(H'')) - c(E(H)) =$ (a,d)+(e,c)+(b,f)-(a,b)-(c,d)-(e-f) path (c,...,b) has changed orientation

One possible solution called the “double bridge” - no path has changed orientation:





TSP - Summary

- One of the "most popular" NP-hard problems
- 49 - 120 – 550 - 2,392 - 7,397 – 19,509 - 24,978 cities from year 1954 to year 2004
- Lot of constraints must be added when solving real life problems
  - CVRP - Capacitated Vehicle routing Problem - limited number of cars and limited load capacity of cars, every customer buys a different amount of the product
  - VRPTW - Time Windows - customers define time windows in which they accept cargo
  - VRPPD - Pick-up and Delivery - customers return some amount of the product (or wrapping) that takes place in the car

References

David L. Applegate, Robert E. Bixby, Vasek Chvátal, and William J. Cook.
The Traveling Salesman Problem: A Computational Study.
Princeton University Press, 2007.

B. H. Korte and Jens Vygen.
Combinatorial Optimization: Theory and Algorithms.
Springer, fourth edition, 2008.