

Reoptimization Problems: Hardness and Algorithms

Tobias Mömke

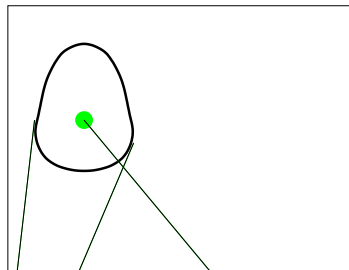
Department of Computer Science, ETH Zurich, Switzerland

Réunion Alpage
31 January 2008

Classical Optimization Problems

NP-hard Problem U

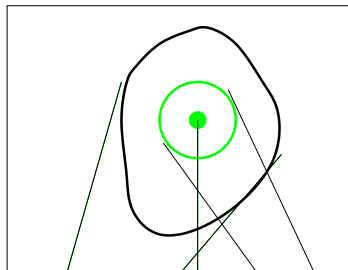
Input:



Input instances

Input (i)

Output:



Solutions

Opt. Solution

Approx. Solutions

Limits of the Classical Problems

- Isolated instances
- Calculation from the scratch
- No use of prior knowledge

Local Modifications

Scenario:

- Optimal (or very good) solution is given

Local Modifications

Scenario:

- Optimal (or very good) solution is given
- Now the input changes slightly (local modification)

Local Modifications

Scenario:

- Optimal (or very good) solution is given
- Now the input changes slightly (local modification)
- Realistic: e.g. railway schedules
 - Good schedule exists

Local Modifications

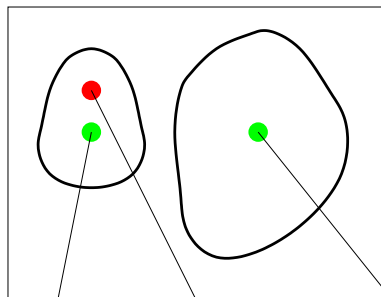
Scenario:

- Optimal (or very good) solution is given
- Now the input changes slightly (local modification)
- Realistic: e.g. railway schedules
 - Good schedule exists
 - Railway station closes down

Reoptimization Problems

Problem Im- U

Input:



Input (i)

Input (ii)

Opt. Solution for (i)

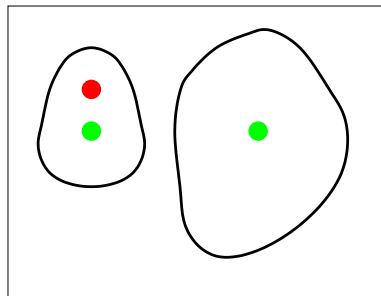
Output:



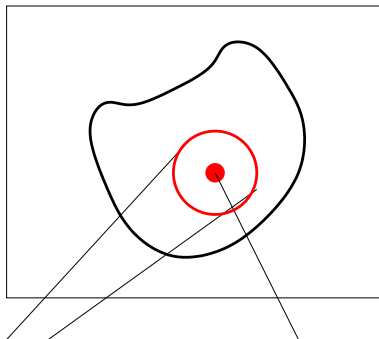
Reoptimization Problems

Problem Im- U

Input:



Output:



Approx. Solutions for (ii)

Opt. Solution for (ii)

Prospects

- Question: Can we use the additional information for hard problems?

Prospects

- Question: Can we use the additional information for hard problems?
 - For faster algorithms?

Prospects

- Question: Can we use the additional information for hard problems?
 - For faster algorithms?
 - For achieving better approximation ratios?

Prospects

- Question: Can we use the additional information for hard problems?
 - For faster algorithms?
 - For achieving better approximation ratios?
- Answer:
 - Sometimes we get a PTAS
 - Sometimes we get a better constant approximation
 - Sometimes the problem stays as hard as before

TSP:

- C. Archetti, L. Bertazzi, M. Speranza:
Reoptimizing the Travelling Salesman Problem.
- H.-J. Böckenhauer, L. Forlizzi, J. Hromkovič, J. Kneis, J. Kupke, G. Proietti, P. Widmayer:
On the approximability of TSP on local modifications of optimally solved instances.
- G. Ausiello, B. Escoffier, J. Monnot, V. Th. Paschos:
Reoptimization of minimum and maximum traveling salesman's tours.
- R. Královič, T. Mömke:
Approximation hardness of the traveling salesman reoptimization problem.

Steiner tree problem:

- H.-J. Böckenhauer, J. Hromkovič, R. Královič, T. Mömke, P. Rossmanith:
Reoptimization of Steiner trees: changing the terminal set.

Considered Problems I

Definition (IncEdge- Δ TSP, DecEdge- Δ TSP)

Given:

- Complete graph G

Considered Problems I

Definition (IncEdge- Δ TSP, DecEdge- Δ TSP)

Given:

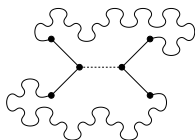
- Complete graph G
- Two metric cost functions $c_O, c_N : E \rightarrow \mathbb{Q}^+$
 - Both functions coincide in all edges but one edge e
 - $c_O(e) < c_N(e)$ ($c_O(e) > c_N(e)$ resp.)

Considered Problems I

Definition (IncEdge- Δ TSP, DecEdge- Δ TSP)

Given:

- Complete graph G
- Two metric cost functions $c_O, c_N : E \rightarrow \mathbb{Q}^+$
 - Both functions coincide in all edges but one edge e
 - $c_O(e) < c_N(e)$ ($c_O(e) > c_N(e)$ resp.)
- A minimum-cost Hamiltonian tour according to c_O

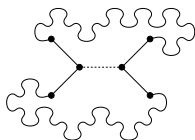


Considered Problems I

Definition (IncEdge- Δ TSP, DecEdge- Δ TSP)

Given:

- Complete graph G
- Two metric cost functions $c_O, c_N : E \rightarrow \mathbb{Q}^+$
 - Both functions coincide in all edges but one edge e
 - $c_O(e) < c_N(e)$ ($c_O(e) > c_N(e)$ resp.)
- A minimum-cost Hamiltonian tour according to c_O



The Problem:

- Find a minimum-cost Hamiltonian tour according to c_N

Considered Problems II

Definition (IncTerm-STP, DecTerm-STP)

Given:

- A complete graph G



Considered Problems II

Definition (IncTerm-STP, DecTerm-STP)

Given:

- A complete graph G
- A cost function $c : E \rightarrow \mathbb{Q}^+$



Considered Problems II

Definition (IncTerm-STP, DecTerm-STP)

Given:

- A complete graph G
- A cost function $c : E \rightarrow \mathbb{Q}^+$
- Terminal sets $S_O, S_N \subseteq V$
 - $S_O \subset S_N$ and $|S_O| + 1 = |S_N|$ ($S_N \subset S_O$ and $|S_O| - 1 = |S_N|$ resp.)



Considered Problems II

Definition (IncTerm-STP, DecTerm-STP)

Given:

- A complete graph G
- A cost function $c : E \rightarrow \mathbb{Q}^+$
- Terminal sets $S_O, S_N \subseteq V$
 - $S_O \subset S_N$ and $|S_O| + 1 = |S_N|$ ($S_N \subset S_O$ and $|S_O| - 1 = |S_N|$ resp.)
- A minimum Steiner tree for (G, S_O, c)



Considered Problems II

Definition (IncTerm-STP, DecTerm-STP)

Given:

- A complete graph G
- A cost function $c : E \rightarrow \mathbb{Q}^+$
- Terminal sets $S_O, S_N \subseteq V$
 - $S_O \subset S_N$ and $|S_O| + 1 = |S_N|$ ($S_N \subset S_O$ and $|S_O| - 1 = |S_N|$ resp.)
- A minimum Steiner tree for (G, S_O, c)



The Problem:

- Find a minimum Steiner tree for (G, S_N, c)

Hardness of TSP Reoptimization

Theorem

The problem $IncEdge-\Delta TSP$ is NP-hard.

- Reduction from restricted Hamiltonian cycle
- Use technique of standard reduction from HC to TSP
 - Edge costs 1 and 2

General Framework for Hardness Proof

Lemma

- *NP-hard problem U , instance I*
- *Local modification Im :*
 - *Efficiently solvable instance I'*
 - *I' transformable to any instance I*
 - *Polynomial number of reoptimizations*

⇒ *$Im-U$ is NP-hard*



Theorem

The problems IncTerm-STP and DecTerm-STP are strongly NP-hard.

- Efficiently solvable: $S = \emptyset$ and $S = V$
- Successively add vertices to S (remove vertices from S resp.)
- At most n reoptimizations

Theorem

The problems IncTerm-STP and DecTerm-STP are strongly NP-hard.

- Efficiently solvable: $S = \emptyset$ and $S = V$
- Successively add vertices to S (remove vertices from S resp.)
- At most n reoptimizations
- Strongly NP-hard:
Construction works for 2-IncTerm-STP and 2-DecTerm-STP

Approximation Algorithm for Im-TSP

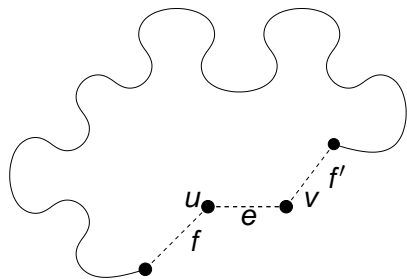
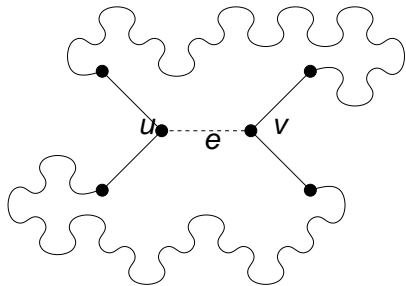
Theorem

- *Polynomial time approximation algorithm for IncEdge- Δ TSP, DecEdge- Δ TSP*
- *Approximation ratio: 1.4*

Observation:

- The modified edge e cannot be in both, the old and the new optimal solution
- Edges adjacent to e are expensive ($\geq 1/2 \cdot |c_O(e) - c_N(e)|$ due to triangle inequality)

Proof Idea for DecEdge- Δ TSP



Approximation Algorithm for DecTerm-STP

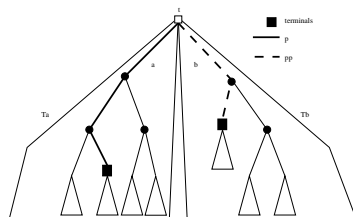
Theorem

There is a 1.5-approximation algorithm for DecTerm-STP.

- We can assume metric costs (construct metric closure)
- Degree of removed terminal is crucial

Main Idea

- Remove shortest paths to terminals in both subtrees
- High cost, length at most $\log_2(n)$
- Connect components by an optimal Steiner tree
 - Possible due to small number of components



PTAS for 2-DecTerm-STP

Theorem

There exists a polynomial time approximation scheme for 2-DecTerm-STP.

PTAS for 2-DecTerm-STP

Theorem

There exists a polynomial time approximation scheme for 2-DecTerm-STP.

- Less than $2 \cdot \lceil 1/\varepsilon \rceil$ terminals: Calculate exact solution (e.g. Dreyus-Wagner algorithm)
- Otherwise keep old solution
- Result costs at most $1 + \varepsilon \text{ OPT}$

PTAS for 2-DecTerm-STP

Theorem

There exists a polynomial time approximation scheme for 2-DecTerm-STP.

- Less than $2 \cdot \lceil 1/\varepsilon \rceil$ terminals: Calculate exact solution (e.g. Dreyus-Wagner algorithm)
- Otherwise keep old solution
- Result costs at most $1 + \varepsilon \text{ OPT}$

Observations:

- Adding one edge to T_N yields feasible solution for (G, S_O, c) .

$$\Rightarrow \text{Opt}_O \leq \text{Opt}_N + r$$

Approximation ratio:

$$\frac{T_A}{\text{Opt}_N} \leq \frac{\text{Opt}_N + r}{\text{Opt}_N} = 1 + \frac{r}{\text{Opt}_N} \leq 1 + \frac{r}{r \cdot k} \leq 1 + \varepsilon.$$

Conclusion

- Most reoptimization problems stay NP-hard

Question: Are there reoptimization problems that allow subexponential algorithms?

Conclusion

- Most reoptimization problems stay NP-hard

Question: Are there reoptimization problems that allow subexponential algorithms?

- Some reoptimization problems allow improved approximation algorithms
- Some reoptimization problems of APX-hard problems allow PTAS

Conclusion

- Most reoptimization problems stay NP-hard

Question: Are there reoptimization problems that allow subexponential algorithms?

- Some reoptimization problems allow improved approximation algorithms
- Some reoptimization problems of APX-hard problems allow PTAS

Question: Which type of problems becomes easier by reoptimization?

Thank you!