Multi-Objective Combinatorial Optimization: The Traveling Salesman Problem and Variants

C. Glaßer[1] C. Reitwießner[1] H. Schmitz[2] M. Witek[1]

[1]University of Würzburg, Germany

[2]Trier University of Applied Sciences, Germany

March 26, 2010, Turku, Finland

Two-Objective Traveling Salesman Problem (2-TSP)

Example

Pizza delivery man wants to deliver pizzas to various addresses and return afterwards.
He wants to minimize the travel time and the risk of a traffic ticket.

Two-Objective Traveling Salesman Problem (2-TSP)

Example

Pizza delivery man wants to deliver pizzas to various addresses and return afterwards.
He wants to minimize the travel time and the risk of a traffic ticket.
(He always drives at full speed.)

Two-Objective Traveling Salesman Problem (2-TSP)

Example

Pizza delivery man wants to deliver pizzas to various addresses and return afterwards.
He wants to minimize the travel time and the risk of a traffic ticket.
(He always drives at full speed.)

Definition (Two-Objective Traveling Salesman Problem (2-TSP))

Instances:	(Multi-)Graph (V, E, c) with edge-labeling $c\colon E \to \mathbb{N}^2$
Solutions:	closed spanning walk W of (V, E)
Costs:	$c(W) = \sum_{e \in W} c(e) \in \mathbb{N}^2$

Two-Objective Traveling Salesman Problem (2-TSP)

Example

Pizza delivery man wants to deliver pizzas to various addresses and return afterwards.
He wants to minimize the travel time and the risk of a traffic ticket.
(He always drives at full speed.)

Definition (Two-Objective Traveling Salesman Problem (2-TSP))

Instances:	(Multi-)Graph (V, E, c) with edge-labeling $c\colon E \to \mathbb{N}^2$
Solutions:	closed spanning walk W of (V, E)
Costs:	$c(W) = \sum_{e \in W} c(e) \in \mathbb{N}^2$

Two-Objective Traveling Salesman Problem (2-TSP)

Example

Pizza delivery man wants to deliver pizzas to various addresses and return afterwards.
He wants to minimize the travel time and the risk of a traffic ticket.
(He always drives at full speed.)

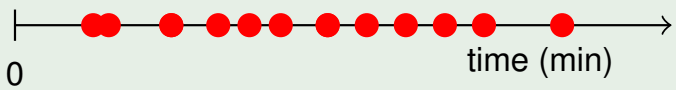
Definition (Two-Objective Traveling Salesman Problem (2-TSP))

Instances:	(Multi-)Graph (V, E, c) with edge-labeling $c\colon E \to \mathbb{N}^2$
Solutions:	closed spanning walk W of (V, E)
Costs:	$c(W) = \sum_{e \in W} c(e) \in \mathbb{N}^2$

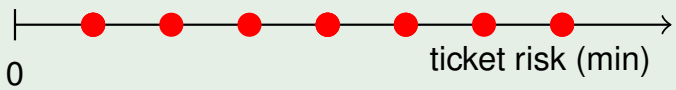
Independent Single-Objective Optimization

Example

Solutions to specific instance, time only:



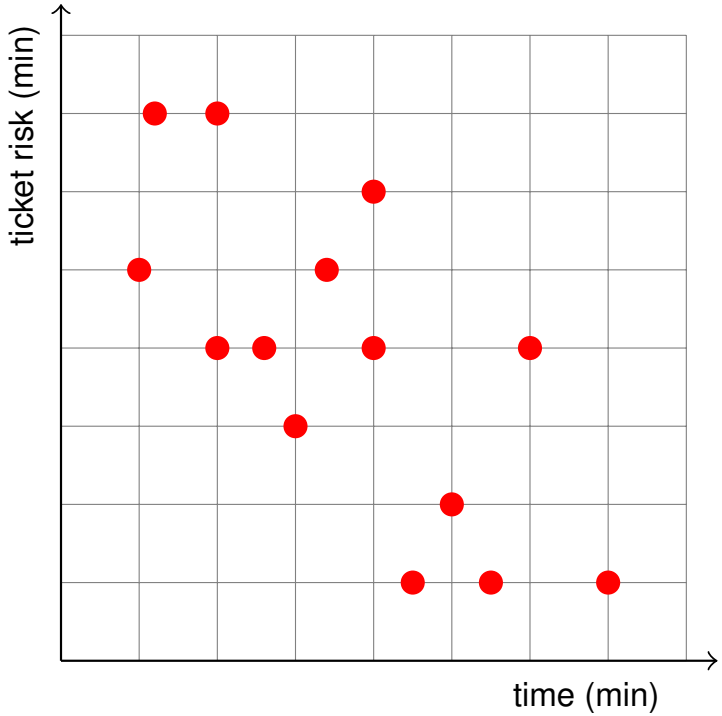
Same solutions, ticket risk only:



$\longrightarrow$ if objectives are optimized independently, we get arbitrary cost value for non-optimized objective

Pareto Optimization

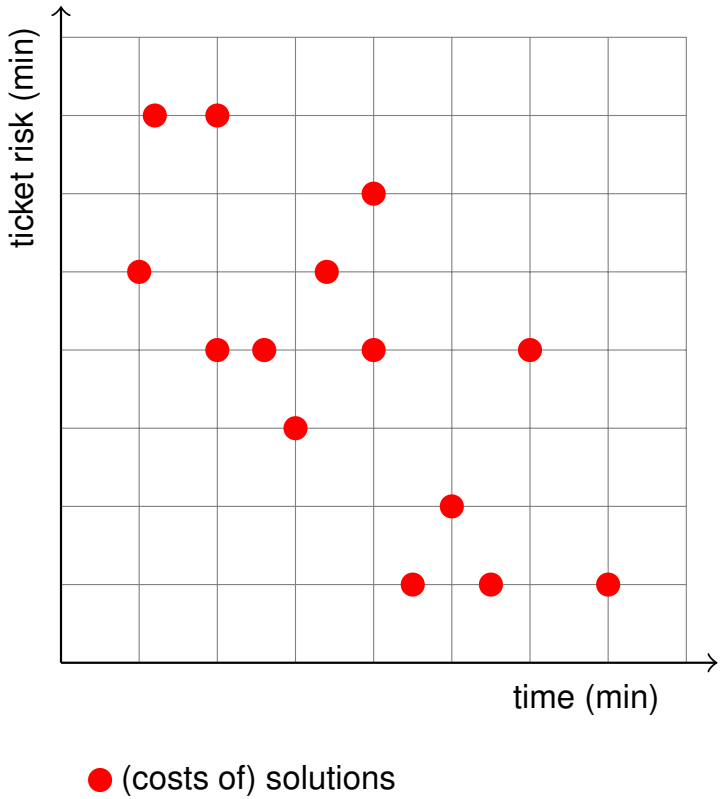
solution cost space for specific instance



● (costs of) solutions

Pareto Optimization

solution cost space for specific instance

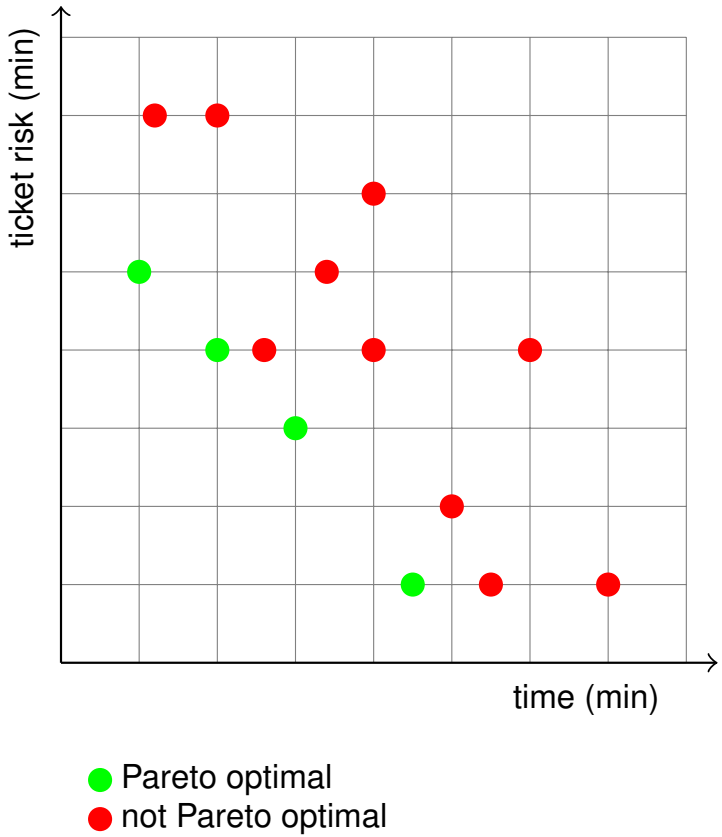


Comparing Solutions

Some pairs of solutions are comparable, some are not.

Pareto Optimization

solution cost space for specific instance



Comparing Solutions

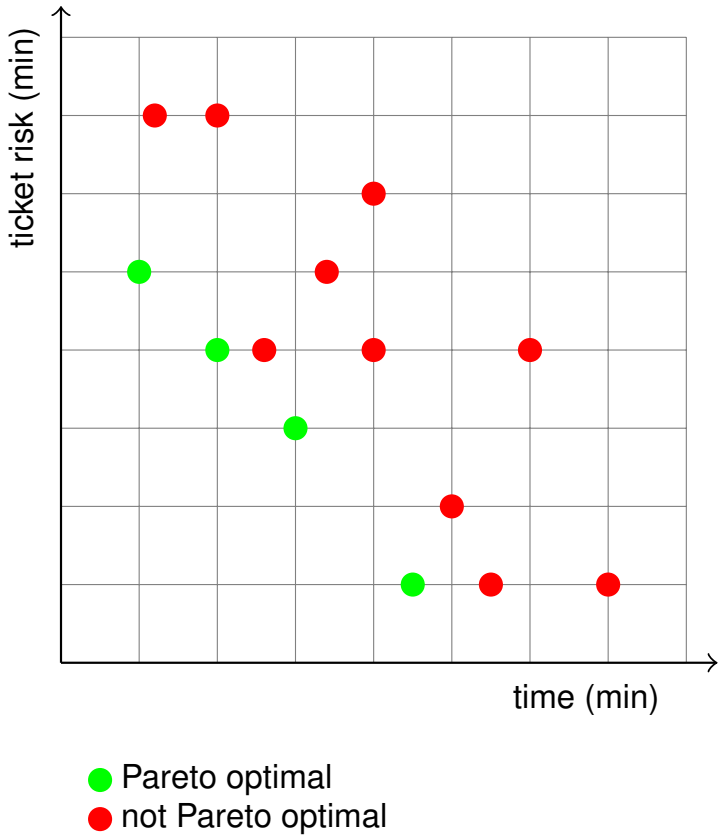
Some pairs of solutions are comparable, some are not.

Definition

A solution is Pareto optimal if no comparable solution is better.
Pareto set: set of Pareto optimal solutions.

Pareto Optimization

solution cost space for specific instance



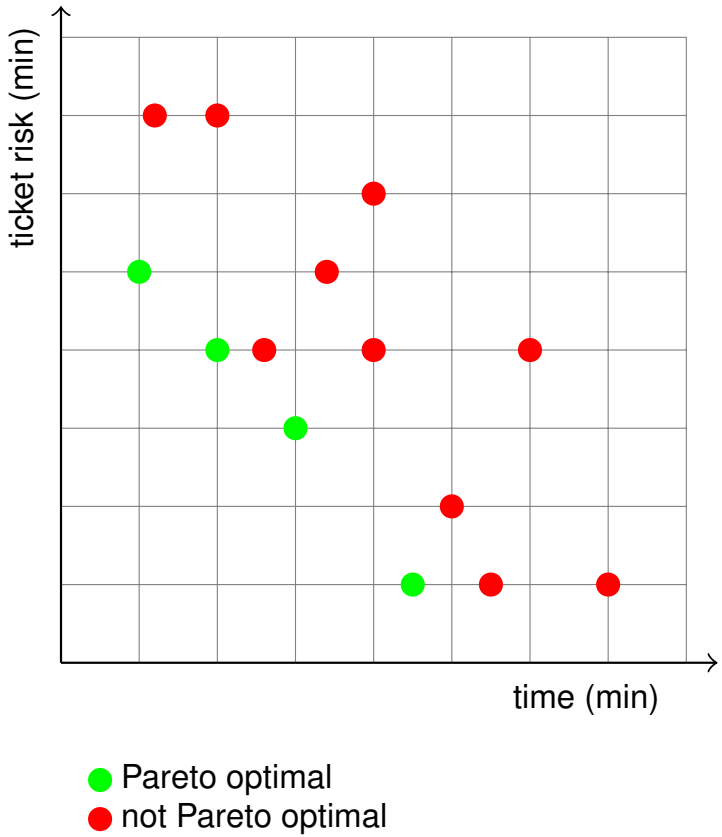
Problem

Computing the Pareto set can be hard:

1. exponential number of Pareto optimal solutions
2. some solutions hard to find

Pareto Optimization

solution cost space for specific instance



Problem

Computing the Pareto set can be hard:

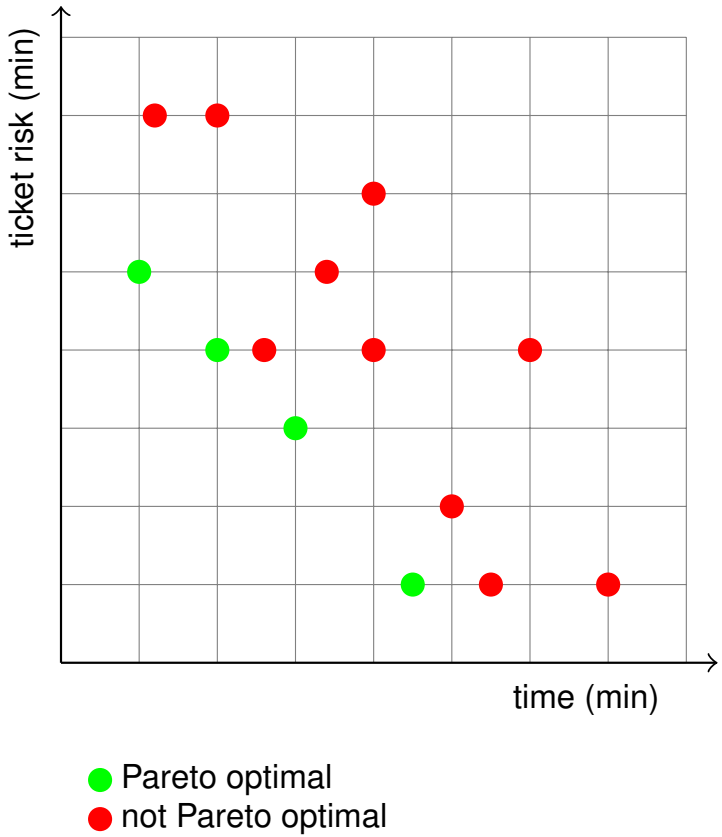
1. exponential number of Pareto optimal solutions
2. some solutions hard to find

Theorem (PY00)

For any multi-objective NP-optimization problem and any $\varepsilon > 1$ there are polynomial-size sets of solutions that ε-approximate the Pareto sets.

Pareto Optimization

solution cost space for specific instance

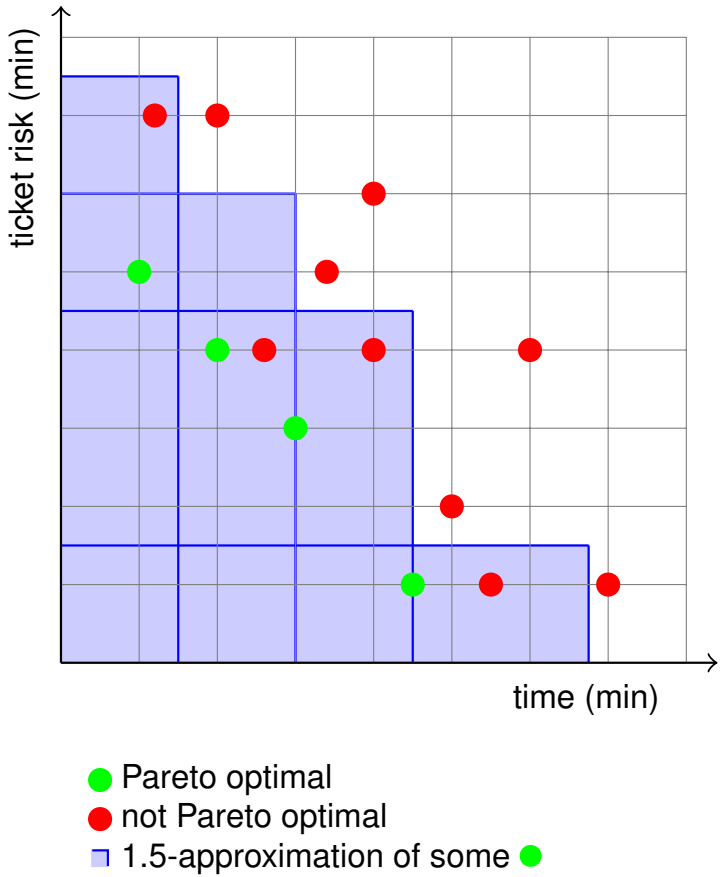


Definition (ε-approximation)

A set of solutions S is an ε-approximation of the Pareto-set P if for every $s \in P$ there is some $s' \in S$ such that for any objective i, $c_i(s') \leq \varepsilon \cdot c_i(s)$. ($\varepsilon > 1$)

Pareto Optimization

solution cost space for specific instance

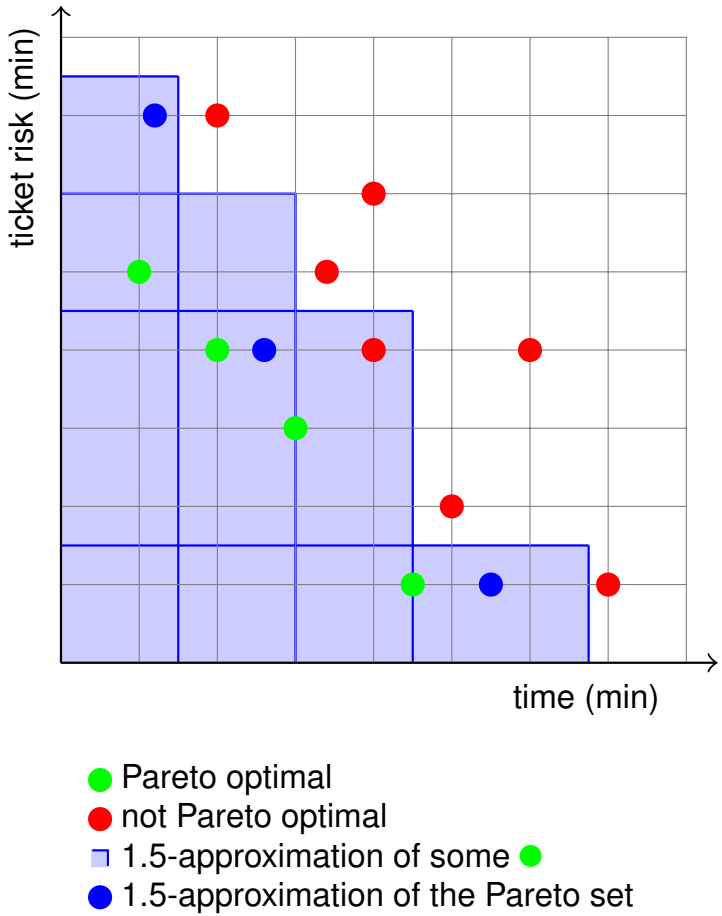


Definition (ε-approximation)

A set of solutions S is an ε-approximation of the Pareto-set P if for every $s \in P$ there is some $s' \in S$ such that for any objective i, $c_i(s') \leq \varepsilon \cdot c_i(s)$. ($\varepsilon > 1$)

Pareto Optimization

solution cost space for specific instance



Definition (ε-approximation)

A set of solutions S is an ε-approximation of the Pareto-set P if for every $s \in P$ there is some $s' \in S$ such that for any objective i, $c_i(s') \leq \varepsilon \cdot c_i(s)$. ($\varepsilon > 1$)

α-Gap Problem

Theorem (PY00,GRSW10)

If for some multi-objective NP-optimization problem and some $\alpha > 1$, the α-Gap-Problem is solvable in polynomial time then an $\alpha + \varepsilon$-approximation for the Pareto set can be computed in polynomial time for every $\varepsilon > 0$.

Problem (α-Gap Problem)

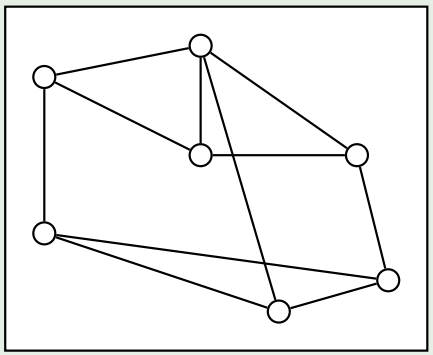
Input: instance, cost vector v
Output: solution s such that $\forall i(c_i(s) \leq \alpha \cdot v_i)$ or
"no" if there is no solution s such that $\forall i(c_i(s) \leq v_i)$

Approximating the Single-Objective TSP

Christofides' Algorithm (1976)

- calculate minimum spanning tree T
- calculate minimum (path) matching M of T's odd-degree vertices
- in $M \uplus T$ each edge has even degree
- spanning walk S' can be extracted by directing each edge

Example

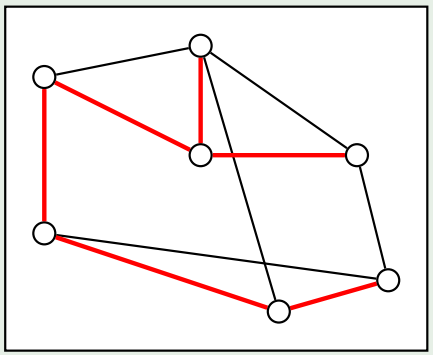


Approximating the Single-Objective TSP

Christofides' Algorithm (1976)

- calculate minimum spanning tree T
- calculate minimum (path) matching M of T's odd-degree vertices
- in $M \uplus T$ each edge has even degree
- spanning walk S' can be extracted by directing each edge

Example



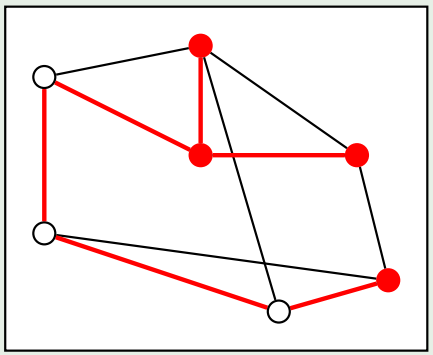
↳ minimum spanning tree

Approximating the Single-Objective TSP

Christofides' Algorithm (1976)

- calculate minimum spanning tree T
- calculate minimum (path) matching M of T's odd-degree vertices
- in $M \uplus T$ each edge has even degree
- spanning walk S' can be extracted by directing each edge

Example



minimum spanning tree

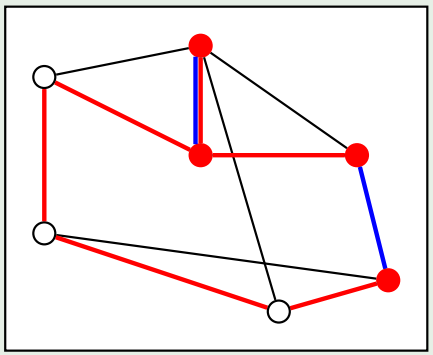
odd degree vertex

Approximating the Single-Objective TSP

Christofides' Algorithm (1976)

- calculate minimum spanning tree T
- calculate minimum (path) matching M of T's odd-degree vertices
- in $M \uplus T$ each edge has even degree
- spanning walk S' can be extracted by directing each edge

Example



minimum spanning tree
odd degree vertex
minimum matching

Performance Analysis of Christofides' Algorithm

Minimum Spanning Tree

$c(T) \leq c(S)$ for any spanning walk S
Proof: By removing edges from S, one obtains a spanning tree.

Performance Analysis of Christofides' Algorithm

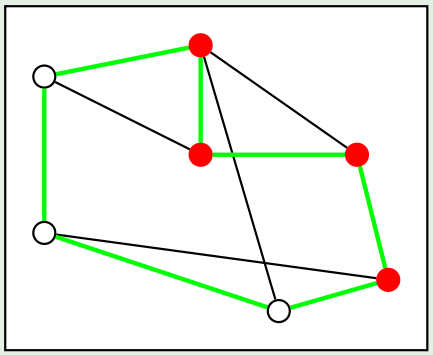
Minimum Spanning Tree

$c(T) \leq c(S)$ for any spanning walk S
Proof: By removing edges from S, one obtains a spanning tree.

Minimum Matching

$c(M) \leq \frac{1}{2}c(S)$ for any spanning walk S
Proof: S consists of two path matchings, use the better one.

Example



- spanning walk
- odd degree vertex in T

Performance Analysis of Christofides' Algorithm

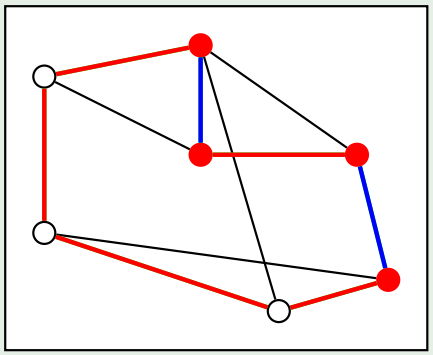
Minimum Spanning Tree

$c(T) \leq c(S)$ for any spanning walk S
Proof: By removing edges from S, one obtains a spanning tree.

Minimum Matching

$c(M) \leq \frac{1}{2}c(S)$ for any spanning walk S
Proof: S consists of two path matchings, use the better one.

Example



- spanning walk
- odd degree vertex in T
- first (path) matching
- second (path) matching

Performance Analysis of Christofides' Algorithm

Whole Algorithm

$c(S') \leq c(T) + c(M) \leq 1.5c(S)$ for any spanning walk S

I.e., Christofides' Algorithm computes a 1.5-approximation of TSP.

Generalization to Multiple Objectives

Problems

- the minimum spanning tree does not exist
- the minimum (path) matching does not exist
- the better of the two matchings in the spanning walk does not exist

Theorem

If a linear single-objective minimization problem is α-approximable, then its k-objective variant is $(k \cdot \alpha)$-approximable.
(a weighted sum of the costs is used as single-objective cost fuction)

Corollary

2-TSP is 3-approximable.

Better Approximation for 2-TSP

Theorem (PY00)

For any $\varepsilon > 0$, we can compute in time polynomial in the length of the input and $1/\varepsilon$ a $(1+\varepsilon)$-approximation for

- *k-objective minimum spanning tree*
- *k-objective minimum (path) matching (randomized)*

for any k.

Sketch of Algorithm for 2-TSP

- compute approximation $\mathcal{T}$ of minimum spanning trees
- for each $T \in \mathcal{T}$
  - compute approximation $\mathcal{M}$ of path matchings of odd-degree vertices in T
  - combine T with all $M \in \mathcal{M}$.

Analysis of 2-TSP-Approximation

Theorem

The algorithm sketched above computes a $(3/2, 2+\varepsilon)$*- and a* $(3/2+\varepsilon, 2)$*-approximation of 2-TSP in (randomized) polynomial time.*

Remark

ε is removed by deleting the heaviest edge from the spanning walk when estimating the cost of the spanning tree.
Matching can only be halved for one objective.

Lower Bound Arguments by Reduction

Definition (Traveling Salesman Path Problem ($TSPP_{st}$))

Instances:	Graph (V, E, c) with edge-labeling function $c\colon E \to \mathbb{N}$, vertices $s, t \in V$
Solutions:	spanning walk W from s to t in (V, E)
Costs:	$c(W) = \sum_{e \in W} c(e) \in \mathbb{N}$

Theorem (Hoo91)

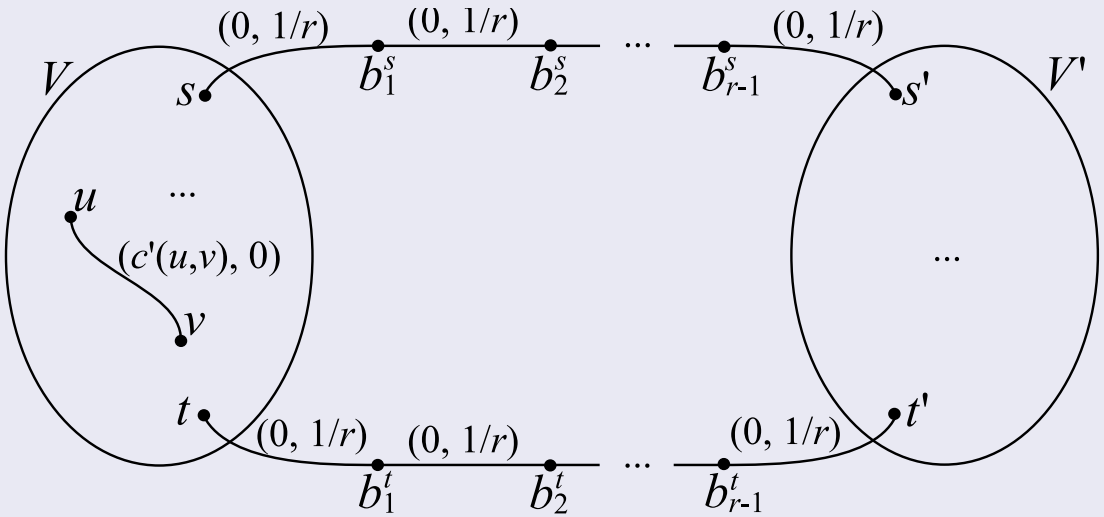
$TSPP_{st}$ is $5/3$-approximable in polynomial time.

Lower Bound Arguments by Reduction

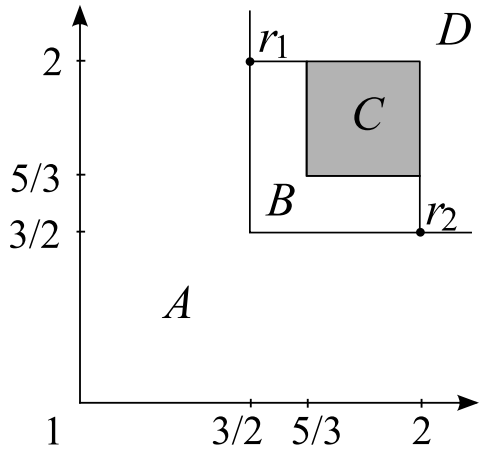
Theorem (Reduction from $TSPP_{st}$ to 2-TSP)

If 2-TSP is $(5/3 - \varepsilon, 2 - \varepsilon)$*-approximable, then* $TSPP_{st}$ *is* $(5/3 - \varepsilon)$*-approximable for any* $\varepsilon > 0$.

Proof sketch



Trade-Offs in Approximation Factors for 2-TSP



Approximation Factors for 2-TSP

- r_1, r_2: 2-TSP-approximation shown in this talk
- Approximation inside...
  - A would improve single-objective TSP
  - B would improve single-objective TSPP$_{st}$
  - C is possible without consequences
  - D is of no interest