Necklace Splitting and Multiobjective Optimization [1]

Christian Reitwießner

CCTVal and UTFSM, Valparaíso, Chile

Seminario conjunto AGCO y Matemáticas Discretas
Universidad de Chile,
9. January 2013

[1] joint with Christian Glaßer and Maximilian Witek at Würzburg, Germany

Splitting Golden Necklaces

Problem Statement

Two thieves (A and B) want to split a stolen golden necklace into parts with as few cuts as possible such that both obtain the same amount of gold.

Splitting Golden Necklaces

Problem Statement

Two thieves (A and B) want to split a stolen golden necklace into parts with as few cuts as possible such that both obtain the same amount of gold.

Model

necklace:	$\mathbb{S}^1$ (unit circle)
gold density:	integrable $f\colon \mathbb{S}^1 \to \mathbb{R}^+$
necklace part:	"interval" (connected subset) in $\mathbb{S}^1$
value of part $I \subseteq \mathbb{S}^1$:	$\int_I f(x)dx$

Splitting Golden Necklaces

Problem Statement

Two thieves (A and B) want to split a stolen golden necklace into parts with as few cuts as possible such that both obtain the same amount of gold.

Model

necklace:	$\mathbb{S}^1$ (unit circle)
gold density:	integrable $f\colon \mathbb{S}^1 \to \mathbb{R}^+$
necklace part:	"interval" (connected subset) in $\mathbb{S}^1$
value of part $I \subseteq \mathbb{S}^1$:	$\int_I f(x)dx$

Observation

intermediate value theorem: two cuts suffice
even works for arbitrary fractions

Splitting More Complex Necklaces

Problem Extension

Necklace consists of k materials (gold, silver, diamond, ...).
1.: Each thief wants exactly half of each material.
2.: A fraction of α of each material for A and $1 - \alpha$ for B.

Splitting More Complex Necklaces

Problem Extension

Necklace consists of k materials (gold, silver, diamond, ...).
1.: Each thief wants exactly half of each material.
2.: A fraction of α of each material for A and $1-\alpha$ for B.

Theorem (Stromquist, Woodall 1985)

Let $k \geq 2$ and $f_1, \ldots, f_k : \mathbb{S}^1 \to \mathbb{R}^+$ integrable. For each $0 \leq \alpha \leq 1$ there are $k-1$ intervals $I_1, I_2, \ldots, I_{k-1} \subseteq \mathbb{S}^1$ such that for all $i \in \{1, \ldots, k\}$

$$\int_{I_1 \cup I_2 \cup \cdots \cup I_{k-1}} f_i(x)dx = \alpha \cdot \int_{\mathbb{S}^1} f_i(x)dx.$$

Proof ingredients: Ham-Sandwich- or Borsuk-Ulam-Theorem for $\alpha \Rightarrow \frac{1}{2}\alpha$, closedness of solution space

Combining Necklaces

We want to combine r necklaces consisting of k materials:

Corollary (Convex Combinations of Necklaces)

For each $r, k \geq 1$ there is a (polynomial) bound $t \in \mathbb{N}$ such that the following holds: For $j = 1, \ldots, r$ and $i = 1, \ldots, k$ let $f_{j,i} \colon \mathbb{S}^1 \to \mathbb{R}$ be integrable and let $\alpha \in [0,1]^r$ such that $||\alpha||_1 = 1$. There is a function $c \colon \mathbb{S}^1 \to \{1, \ldots, r\}$ with at most t jumps such that for all $i \in \{1, \ldots, k\}$

$$\int_{\mathbb{S}^1} f_{c(x),i}(x)dx = \sum_{j=1}^{r} \alpha_j \cdot \int_{\mathbb{S}^1} f_{j,i}(x)dx.$$

Multiobjective Optimization - Motivation

Typical Problem: Buying a Car

Objectives can be:

- price
- gas consumption
- maximal speed
- capacity
- age

Most real-world-problems have multiple conflicting objectives that cannot be easily converted into a single measure like money

Multiobjective Optimization - Motivation

Typical Problem: Buying a Car

Objectives can be:

- price
- gas consumption
- maximal speed
- capacity
- age

Most real-world-problems have multiple conflicting objectives that cannot be easily converted into a single measure like money

Multiobjective Optimization - Motivation

Typical Problem: Buying a Car

Objectives can be:

- price
- gas consumption
- maximal speed
- capacity
- age

Most real-world-problems have multiple conflicting objectives that cannot be easily converted into a single measure like money

Multiobjective Optimization - Motivation

Typical Problem: Buying a Car

Objectives can be:

- price
- gas consumption
- maximal speed
- capacity
- age

Most real-world-problems have multiple conflicting objectives that cannot be easily converted into a single measure like money

Multiobjective Optimization - Motivation

Typical Problem: Buying a Car

Objectives can be:

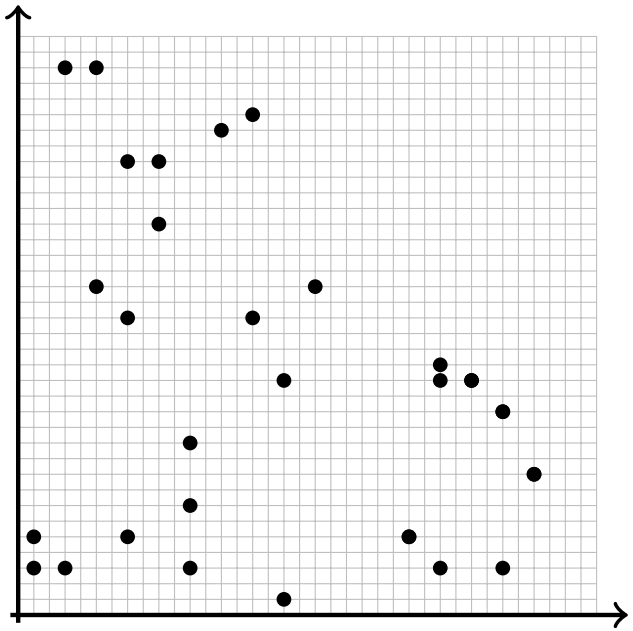
- price
- gas consumption
- maximal speed
- capacity
- age

Most real-world-problems have multiple conflicting objectives that cannot be easily converted into a single measure like money

Multiobjective Optimization - Algorithmic View

Algorithmic Tasks

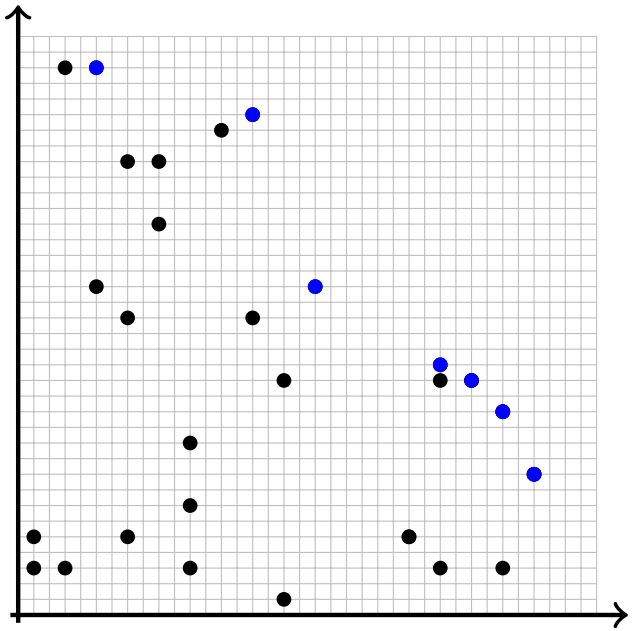
- find all Pareto-optimal solutions (unnecessarily hard if there are many Pareto-optimal solutions)
- find some solution that satisfies a given constraint
- find approximation of such a solution



Multiobjective Optimization - Algorithmic View

Algorithmic Tasks

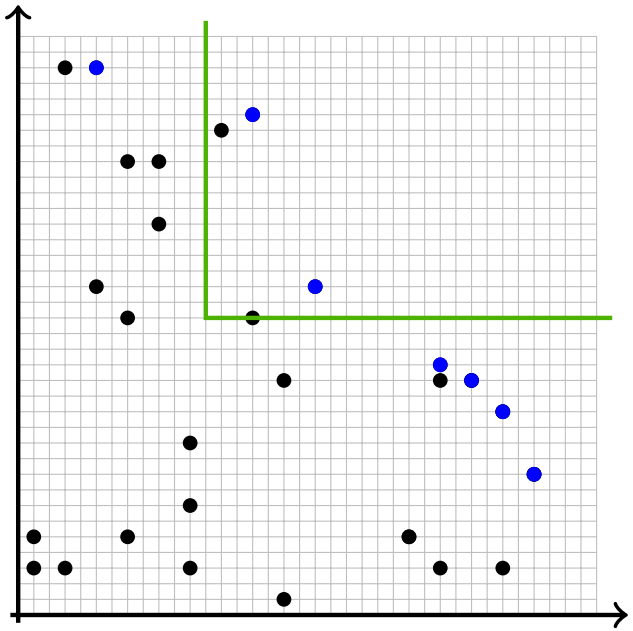
- find all Pareto-optimal solutions (unnecessarily hard if there are many Pareto-optimal solutions)
- find some solution that satisfies a given constraint
- find approximation of such a solution



Multiobjective Optimization - Algorithmic View

Algorithmic Tasks

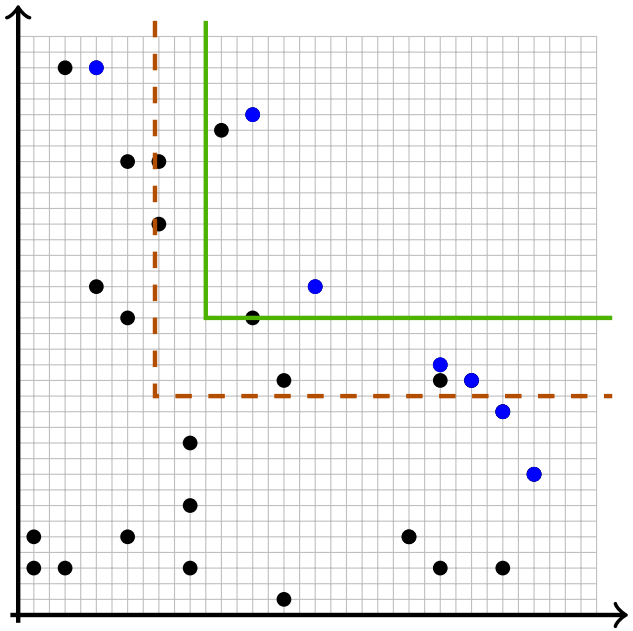
- find all Pareto-optimal solutions (unnecessarily hard if there are many Pareto-optimal solutions)
- find some solution that satisfies a given constraint
- find approximation of such a solution



Multiobjective Optimization - Algorithmic View

Algorithmic Tasks

- find all Pareto-optimal solutions (unnecessarily hard if there are many Pareto-optimal solutions)
- find some solution that satisfies a given constraint
- find approximation of such a solution



Multiobjective Optimization - Definitions

Definition (k-Objective Maximization Problem)

A k-objective maximization problem has the form $(S, f_1, \ldots, f_k)$, where

- $S(x)$: set of solutions for instance x
- $f_i(x, s)$: value of solution $s \in S(x)$ in i-th objective

Multiobjective Optimization - Definitions

Definition (k-Objective Maximization Problem)

A k-objective maximization problem has the form $(S, f_1, \ldots, f_k)$, where

- $S(x)$: set of solutions for instance x
- $f_i(x, s)$: value of solution $s \in S(x)$ in i-th objective

Example (Buying a Car)

- x: car dealer
- $S(x)$: set of cars sold by dealer x
- $f_1(x, s)$: maximal speed of car s sold by dealer x
- $f_2(x, s)$: capacity of car s sold by dealer x

Multiobjective Optimization - Definitions

Definition (k-Objective Maximization Problem)

A k-objective maximization problem has the form $(S, f_1, \ldots, f_k)$, where

- $S(x)$: set of solutions for instance x
- $f_i(x, s)$: value of solution $s \in S(x)$ in i-th objective

Example (Maximum k-Weighted Matching)

- $x = (V, E, w_1, \ldots, w_k)$, where (V, E) is a complete undirected graph and $w_i \colon E \to \mathbb{N}$
- $S(x) = \{M \subseteq E \mid M \text{ is matching of } (V, E)\}$
- $f_i(x, M) = w_i(M) = \sum_{e \in M} w_i(e)$

Known Approximation-Results about Matching

Theorem (Papadimitriou, Yannakakis 2000)

For every k there is an FPRAS (fully polynomial-time randomized approximation scheme) for maximum k-weighted matching.

Known Approximation-Results about Matching

Theorem (Papadimitriou, Yannakakis 2000)

For every k there is an FPRAS (fully polynomial-time randomized approximation scheme) for maximum k-weighted matching.

This means, for every k there is a randomized algorithm A such that on input $(x, \varepsilon, c_1, \ldots, c_k)$, with probability at least $\frac{1}{2}$ it

- finds some $M \in S(x)$ s. t. $f_i(x, M) \geq (1 - \varepsilon)c_i$ for all i or
- states that there is no $M \in S(x)$ with $f_i(x, M) \geq c_i$ for all i

and the runtime of A is polynomial in $|x| + \frac{1}{\varepsilon}$.
Proof: Long chain of reductions, boils down to isolation lemma by Mulmuley, Vazirani, Vazirani.

PTAS for Maximum k-Weighted Matching

Theorem (Chekuri, Vondrák, Zenklusen 2011)

For every k there is a (deterministic) PTAS (polynomial-time approximation scheme) for maximum k-weighted matching.

We present an easier proof using necklace splitting results, although these results are non-constructive and in a continuous setting.
Main feature of necklace splittings: Constant number of cuts or "jump points"

1. rounding error is small (constant times largest element)
2. rounded splitting can be found by exhaustive search

PTAS for Maximum k-Weighted Matching (cont. 1)

Lemma (Convex Combination of Matchings)

Let $k, r \in \mathbb{N}$. On input of $(V, E, w_1, \ldots, w_k)$, matchings $M_1, \ldots, M_r$ and $\alpha \in [0,1]^r$, $||\alpha||_1 = 1$ we can compute a matching M in polynomial time such that for all i

$$w_i(M) \geq \left(\sum_j \alpha_j w_i(M_j) \right) - 2rt \max_{e \in E} w_i(e)$$

where t is the "jump bound" in the necklace combination result.

PTAS for Maximum k-Weighted Matching (cont. 1)

Lemma (Convex Combination of Matchings)

Let $k, r \in \mathbb{N}$. On input of $(V, E, w_1, \ldots, w_k)$, matchings $M_1, \ldots, M_r$ and $\alpha \in [0,1]^r$, $||\alpha||_1 = 1$ we can compute a matching M in polynomial time such that for all i

$$w_i(M) \geq \left(\sum_j \alpha_j w_i(M_j) \right) - 2rt \max_{e \in E} w_i(e)$$

where t is the "jump bound" in the necklace combination result.

Proof idea: Successively combine two matchings by bringing the edges in order and combining them.

Reminder (Convex Combinations of Necklaces):

$\exists c \colon \mathbb{S}^1 \to \{1, \ldots, r\}$ with at most t jumps s.t. $\forall i \in \{1, \ldots, k\}$

$$\int_{\mathbb{S}^1} f_{c(x),i}(x)dx = \sum_{j=1}^{r} \alpha_j \cdot \int_{\mathbb{S}^1} f_{j,i}(x)dx.$$

PTAS for Maximum k-Weighted Matching (cont. 2)

PTAS for Maximum k-Weighted Matching

Input: $(V, E, w_1, \ldots, w_k)$, $(c_1, \ldots, c_k)$, ε

1. eliminate error by guessing heavy edges in solution
2. setup linear program for matching on (V, E)
3. add constraints $w_i(x) \geq c_i$
4. find solution x (fractional matching)
5. find matchings $M_1, \ldots, M_{k+1}$ and $\alpha \in [0,1]^{k+1}$, $||\alpha||_1 = 1$ s.t. $w_i(x) = \sum_{j=1}^{k+1} \alpha_j w_i(M_i)$ for all i
6. use previous lemma to find matching M with $w_i(M) \gtrsim w_i(x)$

Max-k-TSP

Definition (Maximum k-TSP)

- Instances: $(V, E, w_1, \ldots, w_k)$, where (V, E) is a complete undirected graph and $w_i \colon E \to \mathbb{N}$
- $S(x) = \{T \subseteq E \mid T \text{ is a Hamiltonian cycle of } (V, E)\}$
- $f_i(x, T) = w_i(T)$

General idea for approximation:

1. Compute maximum cycle cover
2. Remove one edge from each cycle
3. Connect paths to Hamiltonian cycle

For $k = 1$: Removing lightest edge results in $\frac{2}{3}$-approximation (optimal tour is cycle cover, a cycle has at least three edges)
Not so easy for larger k: There is no “lightest” edge.

Known Results for Max-k-TSP

Approximability of Max-k-TSP

- Bläser, Manthey, Putz, 2008: randomized $\frac{1}{k} - \varepsilon$
- Manthey, 2009: randomized $\frac{2}{3} - \varepsilon$
- Manthey, 2011: deterministic $\frac{1}{2k} - \varepsilon$

All obtained with refinements of the basic cycle-cutting idea.

Necklace splitting shows us how to choose the edges perfectly and thus improve randomized and deterministic result to $\frac{2}{3}$ and $\frac{2}{3} - \varepsilon$, respectively.

Approximating Max-k-TSP using Necklace Splitting

Theorem

For every k and every $\varepsilon > 0$, there is a randomized $\frac{2}{3}$-approximation and a deterministic $(\frac{2}{3} - \varepsilon)$-approximation for Max-$k$-TSP.

Approximating Max-k-TSP using Necklace Splitting

Theorem

For every k and every $\varepsilon > 0$, there is a randomized $\frac{2}{3}$-approximation and a deterministic $(\frac{2}{3} - \varepsilon)$-approximation for Max-$k$-TSP.

1. eliminate error by guessing heavy edges in solution
2. compute maximum cycle cover (via matching)
3. from each cycle C_i select three edges $e_{i,1}, e_{i,2}, e_{i,3}$
4. combine the three necklaces $\{e_{1,1}, \ldots, e_{m,1}\}$, $\{e_{1,2}, \ldots, e_{m,2}\}$ and $\{e_{1,3}, \ldots, e_{m,3}\}$ in ratios $\frac{1}{3} : \frac{1}{3} : \frac{1}{3}$
5. remove resulting edges from cycles and join obtained paths to single cycle

Similarly: $\frac{1}{2}$ / $(\frac{1}{2} - \varepsilon)$ for Max-k-TSP on directed graphs (smallest cycles have two edges)

Conclusion

Applications of Necklace Splitting in Multiobjective Optimization

- PTAS for k-objective matching and cycle cover
- $\frac{2}{3}$ / $(\frac{2}{3} - \varepsilon)$-appr. for Max-$k$-TSP
- $\frac{1}{2}$ / $(\frac{1}{2} - \varepsilon)$-appr. for Max-$k$-ATSP (on directed graphs)
- $\frac{1}{2}$-approximation for k-objective weighted Max-SAT

More to come?