Algorithms for Discrete, Non-Linear and Robust Optimization Problems with Applications in Scheduling and Service Operations

Shashi Mittal

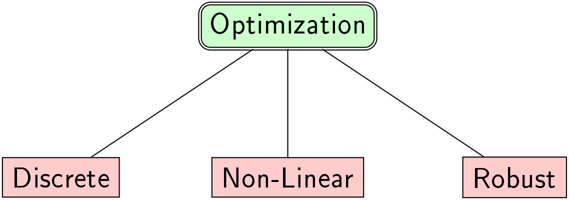
Operations Research Center, MIT

August 8, 2011

What is My Thesis All About?

Optimization

What is My Thesis All About?



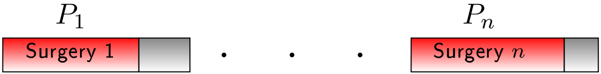
Outline

- Robust appointment scheduling (M. and Stiller 2011)
- Optimizing functions of low rank over a polytope (M. and Schulz 2010)
- Approximation schemes for combinatorial optimization problems with many objectives combined into one (M. and Schulz 2011)

The Problem

Example: Scheduling outpatient surgeries

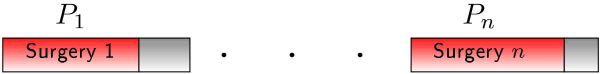
Given:



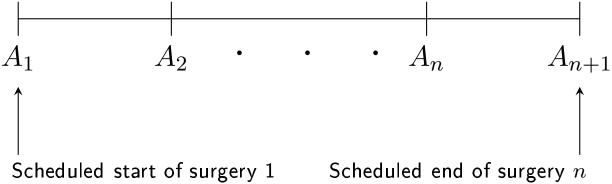
The Problem

Example: Scheduling outpatient surgeries

Given:



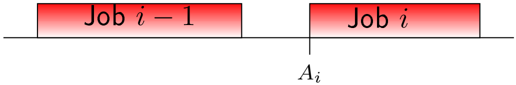
Find:



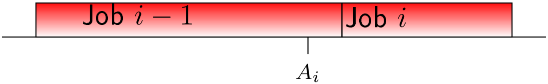
The Problem

Job processing:

- If job $i-1$ finishes before A_i, job i starts at A_i.



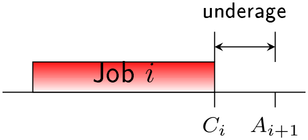
- Otherwise: job i starts immediately after completion of job i.



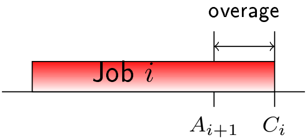
Costs

C_i = completion time of job i.

- $C_i < A_{i+1}$: underage cost $u_i(A_{i+1} - C_i)$. (Job i is *underaged*)



- $C_i > A_{i+1}$: overage cost $o_i(C_i - A_{i+1})$. (Job i is *overaged*)



Costs

P : a given realization of processing times of jobs.

Cost function

$$F(A,P) = \sum_{i=1}^{n} \max(o_i(C_i - A_{i+1}), u_i(A_{i+1} - C_i))$$

Example 1: 3 jobs, $u = 10$, $o = 1$

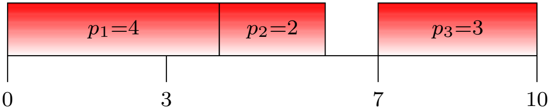
Costs

P : a given realization of processing times of jobs.

Cost function

$$F(A,P) = \sum_{i=1}^{n} \max(o_i(C_i - A_{i+1}), u_i(A_{i+1} - C_i))$$

Example 1: 3 jobs, $u = 10$, $o = 1$



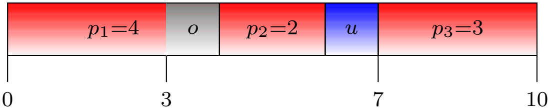
Costs

P : a given realization of processing times of jobs.

Cost function

$$F(A,P) = \sum_{i=1}^{n} \max(o_i(C_i - A_{i+1}), u_i(A_{i+1} - C_i))$$

Example 1: 3 jobs, $u = 10$, $o = 1$



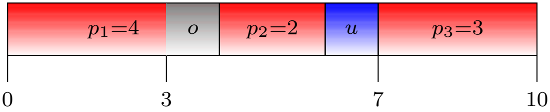
Costs

P : a given realization of processing times of jobs.

Cost function

$$F(A,P) = \sum_{i=1}^{n} \max(o_i(C_i - A_{i+1}), u_i(A_{i+1} - C_i))$$

Example 1: 3 jobs, $u = 10$, $o = 1$

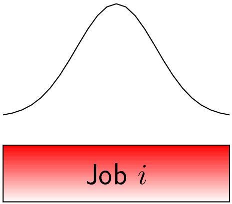


Total Cost = $1 + 10 + 0 = 11$

Other Applications

- Project scheduling (Bendavid and Golany, 2009)
- Serial production systems (Elhafsi 2002)
- Servicing ships at seaports (Sabria and Daganzo 1989)
- Professors scheduling meeting with grad students
- ...

Existing Models



P_i: random variable

Cost function

$$F(A) = \mathbb{E}_P[F(A, P)]$$

Optimization problem: Minimize expected cost

Known Methods

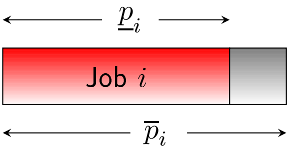
- Sequential bounding algorithm (Denton and Gupta 2003)
- Monte Carlo techniques (Robinson and Chen 2003)
- Local search (Kandoorp and Koole 2007)
- Submodular function minimization (Begen and Queyranne 2009)

Our Contributions

1. A robust optimization framework
2. Closed form optimal solution
3. Near-optimal sequencing of jobs when jobs can be re-arranged

The Robust Model

Given: minimum and maximum possible execution time of each job.



$\mathcal{P}$: Set of all possible realization of processing times of jobs

Robust Model

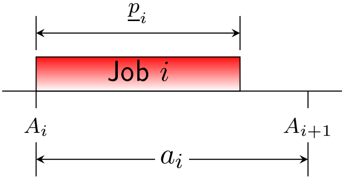
$$F(A) = \max_{P \in \mathcal{P}} F(A, P)$$

Optimization problem: minimize worst-case scenario(s) cost.

The Global Balancing Heuristic

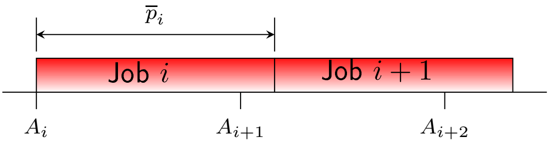
Main idea: Balance between maximum underage cost of job i, and maximum overage cost *due to* job i.

Maximum possible underage cost of job $i = u_i(a_i - \underline{p}_i)$.



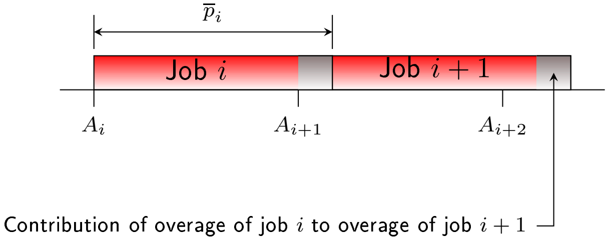
The Global Balancing Heuristic

Maximum possible contribution of job i to overage costs of all jobs succeeding i:



The Global Balancing Heuristic

Maximum possible contribution of job i to overage costs of all jobs succeeding i: $(\sum_{j=i}^{n} o_j)(\overline{p}_i - a_i)$.



The Global Balancing Heuristic

Equating maximum possible underage and overage costs:

$$u_i(a_i - \underline{p}_i) = (\sum_{j=i}^{n} o_j)(\overline{p}_i - a_i)$$

We get:

$$a_i^G = \frac{u_i \underline{p}_i + (\sum_{j=i}^{n} o_j)\overline{p}_i}{u_i + \sum_{j=i}^{n} o_j}$$

The Main Theorem

Theorem (M. and Stiller 2011)

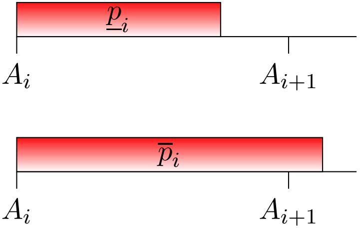
The global balancing schedule is optimal for the robust version when the underage costs of the jobs are non-decreasing ($u_i \leq u_{i+1}$).

Closed form optimal solution for robust model

Intuitive Interpretation

If job i alone is scheduled:

$$a_i^* = \frac{u_i \underline{p}_i + o_i \overline{p}_i}{u_i + o_i}$$

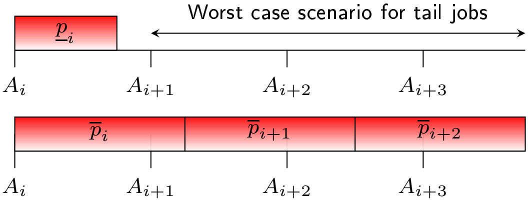


Intuitive Interpretation

However, if jobs $i+1, \ldots, n$, are to be scheduled after job i:

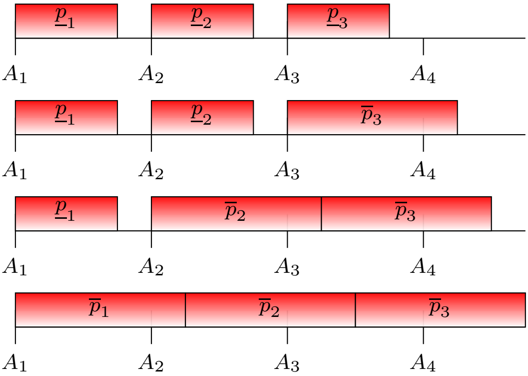
$$a_i^G = \frac{u_i \underline{p}_i + o_{\geq i} \overline{p}_i}{u_i + o_{\geq i}}$$

where $o_{\geq i} = \sum_{j=i}^{n} o_j$.



Worst Case Scenarios for the Optimal Schedule

Sequence of min-length jobs followed by max-length jobs.



Comparison with Stochastic Model

Cost parameters: $u = 10$, $o = 1$

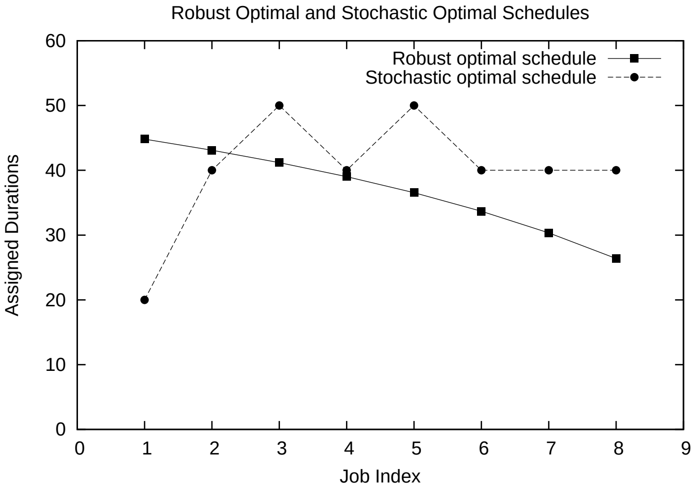
Stochastic model:

- Job durations: Discrete version Weibull distribution with $\mu = 48$ and $\sigma = 26$
- Stochastic optimal solution found using local search

Robust model:

- $\underline{p} = \mu - \sigma = 22$, $\overline{p} = \mu + \sigma = 74$

Comparison with Stochastic Model



Ordering Problem

Uniform underage cost ($u_i = 1$)

Let $\Delta_i = \overline{p}_i - \underline{p}_i$.

- Same overage cost:

Schedule in *increasing* order of Δ

Ordering Problem

Uniform underage cost ($u_i = 1$)

Let $\Delta_i = \overline{p}_i - \underline{p}_i$.

- Same overage cost:

  Schedule in *increasing* order of Δ

- Same Δ:

  Schedule in *decreasing* order of o.

Ordering Heuristic

Sequence jobs in increasing order of Δ/o values.

Ordering Heuristic

Sequence jobs in increasing order of Δ/o values.

Theorem (M. and Stiller 2011)

The heuristic gives an α-approximate solution to the ordering problem, where

$$\alpha = \min\left(\frac{1+o_{\geq 1}}{1+o_{\min}} \quad , \quad \frac{o_{\geq 1}}{1+o_{\geq 1}} \cdot \frac{1+o_{\min}}{o_{\min}}\right)$$

$$o_{\geq 1} = \sum_{i=1}^{n} o_i, \qquad o_{\min} = \min_{i=1,\dots,n} o_i$$

Gives a reasonable approximation factor when $o_{\geq 1}$ is not too big compared to $o_{\min}$.

Insights

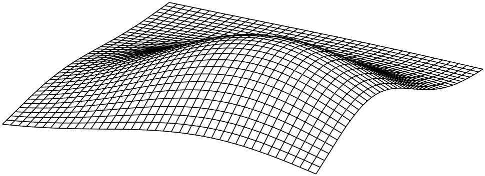
- Time allowances should be greater for the jobs in the start, and smaller for the jobs in the end.
- Sequencing in increasing order of Δ/o ratio gives a near-optimal ordering of the jobs.

Outline

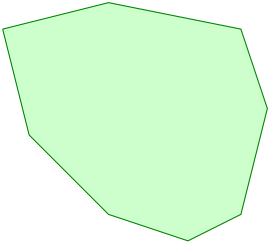
- Robust appointment scheduling (M. and Stiller 2011)
- Optimizing functions of low rank over a polytope (M. and Schulz 2010)
- Approximation schemes for combinatorial optimization problems with many objectives combined into one (M. and Schulz 2011)

Question

What is the complexity of minimizing a non-convex function over a polytope?



$\min f(x)$



$x \in P$

NP-Hardness Results

The following problems are NP-hard (Matsui 1996):

Problem 1

$$\begin{aligned} \min \quad & x_1 x_2 \\ \text{s.t.} \quad & Cx \geq d \end{aligned}$$

Problem 2

$$\begin{aligned} \max \quad & \frac{1}{x_1} + \frac{1}{x_2} \\ \text{s.t.} \quad & Cx \geq d \end{aligned}$$

Hardness of Approximation

Theorem (M. and Schulz 2010)

The optimal solution of the problem

$$\min f(x)$$
$$x \in [0,1]^n$$

where $f(x)$ is a concave function, cannot be approximated to within any factor unless P = NP.

Low Rank Functions

Definition

A function $f : \mathbb{R}^n \to \mathbb{R}$ is of rank k if

$$f(x) = g(a_1^T x, \dots, a_k^T x),$$

where $a_1, \dots, a_k$ are k linearly independent vectors.

Low rank: k fixed.

The Optimization Problem

Problem

$$\begin{aligned} \min / \max \quad & f(x) = g(a_1^T x, \dots, a_k^T x) \\ \text{s.t.} \quad & x \in P \end{aligned}$$

Examples:

- $f(x) = (a_1^T x) \cdot (a_2^T x)$ (multiplicative)
- $f(x) = (a_1^T x) \cdot (b_1^T x) + (a_2^T x) \cdot (b_2^T x)$ (bi-linear)
- $f(x) = \frac{a_1^T x}{b_1^T x} + \frac{a_2^T x}{b_2^T x}$ (sum-of-fractions)

Challenges

- Can have multiple local optima, so any global minimization algorithm must avoid getting stuck into a local optimum.
- Results known mostly for minimizing quasi-concave functions of low rank over a polytope (e.g. Goyal and Ravi (2009), Kelner and Nikolova (2007), Porembski (2004))
- Bi-linear functions and sum-of-ratios functions are neither quasi-concave nor quasi-convex.

Fully Polynomial Time Approximation Scheme (FPTAS)

Consider a family of minimization problems:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Definition

FPTAS: A family of algorithms A_ϵ, such that for any $\epsilon > 0$, the algorithm A_ϵ

- is a $(1 + \epsilon)$-approximation algorithm.
- has running time polynomial in input size and $1/\epsilon$.

Our Result

FPTAS for the following optimization problem for a fixed k:

Problem

$$\begin{aligned} \min / \max \quad & f(x) = g(a_1^T x, \ldots, a_k^T x) \\ \text{s.t.} \quad & x \in P \end{aligned}$$

Assumptions

- $g(y) \leq g(y')$ for all $y \leq y'$.
- $g(\lambda y) \leq \lambda^c g(y)$ for all $\lambda > 1$ and some constant c.
- $a_i^T x > 0$ for all $i = 1, \ldots, k$ over the given polytope.

Our Result

Examples of functions satisfying the above conditions:

- Multiplicative forms: $f(x) = \prod_{i=1}^{k}(a_i^T x)$
- Bi-linear forms: $f(x) = \sum_{i=1}^{k}(a_i^T x) \cdot (b_i^T x)$

The monotonicity assumption can be relaxed:

- For example, the sum-of-ratios form: $f(x) = \sum_{i=1}^{k} \frac{a_i^T x}{b_i^T x}$

The Solution Approach

Problem π

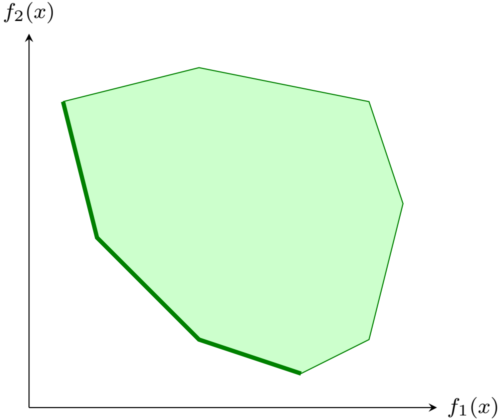
$$\begin{aligned} \min \quad & f(x) = (a_1^T x) \cdot (a_2^T x) \\ \text{s.t.} \quad & Cx \geq d \end{aligned}$$

Solution

- Let $f_i(x) = a_i^T x$.
- Compute an *approximate Pareto-optimal* frontier of the functions f_i.
- Return the best solution from the approximate Pareto-optimal frontier.

Pareto-optimal Frontier

Pareto-optimal front ($P(\pi)$) is the set of all non-dominated solution points.

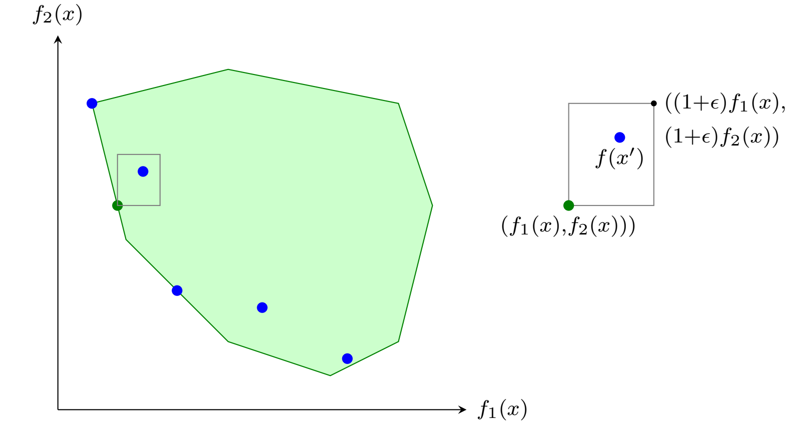


Approximate Pareto-optimal Frontier

Set of solutions $P_\epsilon(\pi)$ such that:

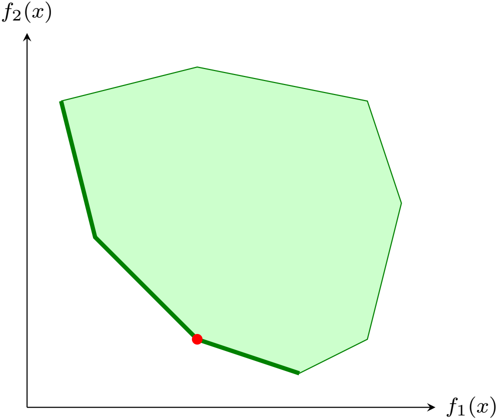
for all feasible x, there is $x' \in P_\epsilon(\pi)$ such that

$f_i(x') \leq (1+\epsilon) f_i(x)$.



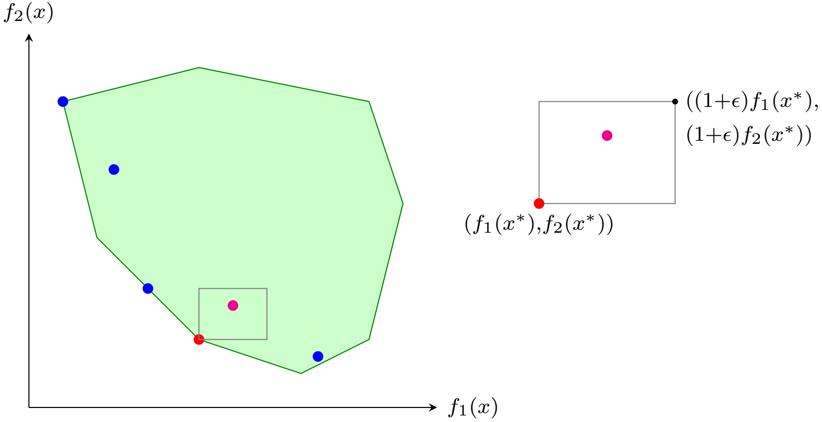
Lemma 1

An optimal solution of the problem π lies on the Pareto-optimal front.



Lemma 2

Let $\hat{x}$ be the solution in $P_\epsilon(\pi)$ that minimizes $f(x)$ over all the points in $P_\epsilon(\pi)$. Then $\hat{x}$ is a $(1+\epsilon)^2$-approximate solution.



The Gap Theorem (Papadimitriou and Yannakakis 2000)

For a fixed k, it is possible to find a $P_\epsilon(\pi)$ in time polynomial in $|\pi|$ and $1/\epsilon$ *iff* the following "gap problem" can be solved in polynomial time.

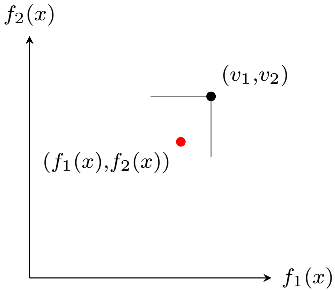
The Gap Theorem (Papadimitriou and Yannakakis 2000)

For a fixed k, it is possible to find a $P_\epsilon(\pi)$ in time polynomial in $|\pi|$ and $1/\epsilon$ *iff* the following "gap problem" can be solved in polynomial time.

Gap problem

Given a k vector of values $(v_1, \ldots, v_k)$, either

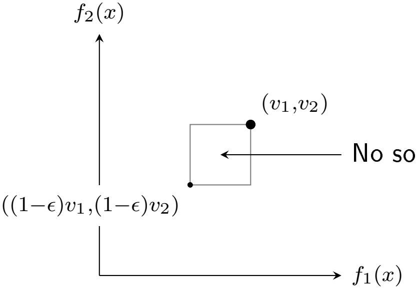
(i) return a feasible x such that $f_i(x) \leq v_i$ for all $i = 1, \ldots, k$, or ..



The Gap Theorem (Papadimitriou and Yannakakis 2000)

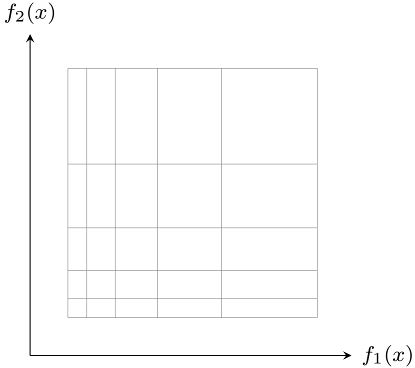
Gap problem

(ii) assert that there is no feasible x' such that $f_i(x') \leq (1-\epsilon)v_i$ for all $i = 1, \ldots, k$.



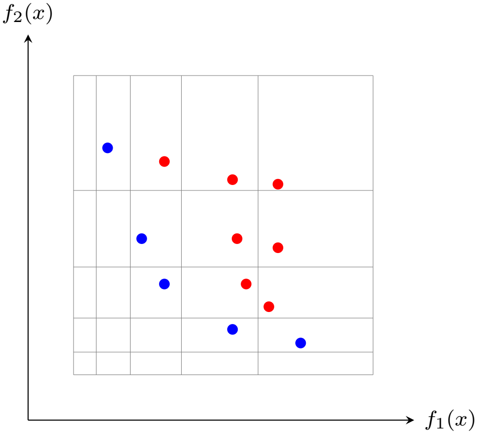
The Approximation Scheme

Divide the solution space into smaller hyper-rectangles, such that in each dimension, the ratio of successive divisions is equal to $1 + \epsilon'$. (ϵ' depends on ϵ).



The Approximation Scheme

For each corner point, solve the gap problem. Return the set of undominated solution points.



Solving the Gap Problem

Same as checking the feasibility of the following LP, for each corner point $(v_1, \ldots, v_k)$:

Gap Problem LP

$$\begin{aligned} Cx &\geq d, \\ a_i^T x &\leq (1-\epsilon')v_i, \text{ for } i = 1, \ldots, k. \end{aligned}$$

Need to check feasibility of $O\left(\left(\frac{\log(M/m)}{\epsilon}\right)^k\right)$ LPs.

Applications: Sum-of-Ratios Optimization

$$\min / \max f(x) = \frac{a_1^T x}{b_1^T x} + \ldots + \frac{a_k^T x}{b_k^T x}, \text{ s.t. } Cx \geq d.$$

- Application: Multi-stage shipping problem (Falk and Palocsay, 1992).
- Sum-of-fractions is not quasi-convex/quasi-concave in general, no approximation algorithms known.
- Our result: FPTAS when k is fixed.

Minimizing Quasi-concave functions

- $f(x)$ quasi-concave function: Can get an FPTAS which returns an extreme point of the polytope as an approximate solution (M. and Schulz 2010, Goyal and Ravi 2010).

- Application: FPTAS for combinatorial optimization problems with a quasi-concave objective function.

A More General Combinatorial Optimization Problem

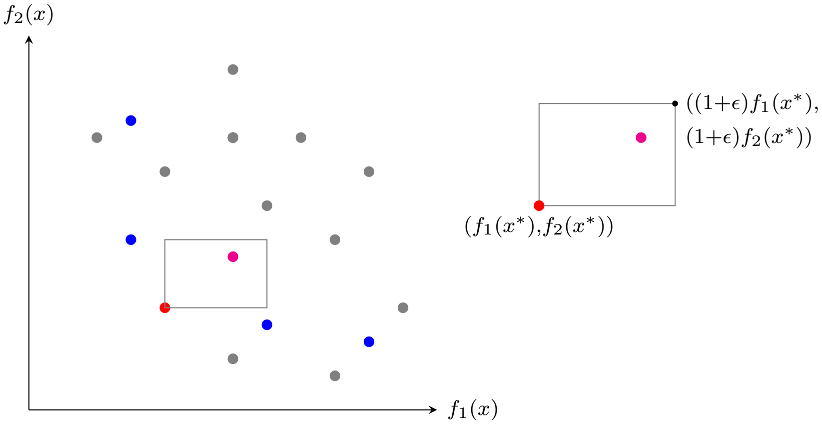
Problem

$$\begin{aligned} \min / \max \quad & f(x) = g(a_1^T x, \ldots, a_k^T x) \\ \text{s.t.} \quad & x \in X \subseteq \{0,1\}^d \end{aligned}$$

- k is fixed.
- g satisfies the same properties as before.
  (but need not be quasi-concave)

Solution Approach

Find best solution in an approximate Pareto-optimal front



Solving the Gap Problem

Theorem (Papadimitriou and Yannakakis 2000)

The gap problem can be solved in polynomial time, if the following exact problem can be solved in pseudo-polynomial time:

Given a non-negative integer C *and a vector* $(c_1, \ldots, c_d) \in \mathbb{Z}_+$*, does there exist a solution* $x \in X$ *such that*

$$\sum_{i=1}^{d} c_i x_i = C?$$

Examples

- Max-min resource allocation problem
- Scheduling problems with makespan objective
- Assortment optimization problems with logit choice model (sum-of-fractions form)

Insights

Which forms are easy to approximate?

- Product
- Bi-linear
- Sum-of-ratios

Provided: low-rank and individual terms positive.

Insights

Which forms are easy to approximate?

- Product
- Bi-linear
- Sum-of-ratios

Provided: low-rank and individual terms positive.

However:

- Difference-of-function forms are hard to approximate.

The purpose of Mathematical Programming is insight, not numbers.

- A. M. Geoffrion