A General Framework for Designing Approximation Schemes for Combinatorial Optimization Problems with Many Objectives Combined into One

Shashi Mittal and Andreas S. Schulz

August 26, 2009

Combinatorial Optimization with Rational Objective

$$\min g(x) = \frac{f_1(x)}{f_2(x)} = \frac{a_0 + a_1 x + \ldots + a_n x}{b_0 + b_1 x + \ldots + b_n x},$$
$$x \in X \subseteq \{0, 1\}^n$$

Known: Can be solved in polynomial time, if the corresponding linear discrete program can be solved in Polynomial time (Megiddo 1979)

Question: Under what conditions does this problem admit an FPTAS?

Combinatorial Optimization with Rational Objective

$$\min g(x) = \frac{f_1(x)}{f_2(x)} = \frac{a_0 + a_1 x + \ldots + a_n x}{b_0 + b_1 x + \ldots + b_n x},$$
$$x \in X \subseteq \{0,1\}^n$$

Known: Can be solved in polynomial time, if the corresponding linear discrete program can be solved in Polynomial time (Megiddo 1979)

Question: Under what conditions does this problem admit an FPTAS?

Combinatorial Optimization with Rational Objective

$$\min g(x) = \frac{f_1(x)}{f_2(x)} = \frac{a_0 + a_1 x + \ldots + a_n x}{b_0 + b_1 x + \ldots + b_n x},$$
$$x \in X \subseteq \{0,1\}^n$$

Known: Can be solved in polynomial time, if the corresponding linear discrete program can be solved in Polynomial time (Megiddo 1979)

Question: Under what conditions does this problem admit an FPTAS?

The Santa Claus/Max-Min Fair Allocation Problem

- m agents, n resources.
- Each resource can be given to only one of the agents.
- The utility of agent i for resource k is p_{ik}.
- **Objective**: Maximize the minimum utility over all the agents.

Known: $O(\sqrt{m}\log^3 m)$ approximation algorithm for the general case (Asadpour and Saberi 2007).

Question: What happens when the number of agents is fixed?

The Santa Claus/Max-Min Fair Allocation Problem

- m agents, n resources.
- Each resource can be given to only one of the agents.
- The utility of agent i for resource k is p_{ik}.
- **Objective**: Maximize the minimum utility over all the agents.

Known: $O(\sqrt{m}\log^3 m)$ approximation algorithm for the general case (Asadpour and Saberi 2007).

Question: What happens when the number of agents is fixed?

The Santa Claus/Max-Min Fair Allocation Problem

- m agents, n resources.
- Each resource can be given to only one of the agents.
- The utility of agent i for resource k is p_{ik}.
- **Objective**: Maximize the minimum utility over all the agents.

Known: $O(\sqrt{m}\log^3 m)$ approximation algorithm for the general case (Asadpour and Saberi 2007).

Question: What happens when the number of agents is fixed?

Budgeted Resource Allocation Problem

- m agents, n objects.
- Each agent i has a budget B_i, the maximum amount of money she can spend.
- Each agent i bids b_{ik} for each object k.
- **Objective**: Allocate the objects to agents to maximize the total revenue.

Known: $3/4$-approximation algorithm for the general case (Chakrabarty and Goel (2008), Srinivasan (2008)).

Question: What happens when the number of agents is fixed?

Budgeted Resource Allocation Problem

- m agents, n objects.
- Each agent i has a budget B_i, the maximum amount of money she can spend.
- Each agent i bids b_{ik} for each object k.
- **Objective**: Allocate the objects to agents to maximize the total revenue.

Known: $3/4$-approximation algorithm for the general case (Chakrabarty and Goel (2008), Srinivasan (2008)).

Question: What happens when the number of agents is fixed?

Budgeted Resource Allocation Problem

- m agents, n objects.
- Each agent i has a budget B_i, the maximum amount of money she can spend.
- Each agent i bids b_{ik} for each object k.
- **Objective**: Allocate the objects to agents to maximize the total revenue.

Known: $3/4$-approximation algorithm for the general case (Chakrabarty and Goel (2008), Srinivasan (2008)).

Question: What happens when the number of agents is fixed?

Minimizing Product of Discrete Linear Functions

Given a graph $G = (V, E)$ and two edge weights a, b.

- Find an $s - t$ P path that minimizes $a(P).b(P)$.
- Find a spanning tree T that minimizes $a(T).b(T)$.

Known: FPTAS when objective is product of two linear functions (Kern and Woeginger (2006), Genc-Kaya, Goyal and Ravi (2008)).

Question: What happens when we have more than two (but a fixed number) of linear functions in the product?

Minimizing Product of Discrete Linear Functions

Given a graph $G = (V, E)$ and two edge weights a, b.

- Find an $s - t$ P path that minimizes $a(P).b(P)$.
- Find a spanning tree T that minimizes $a(T).b(T)$.

Known: FPTAS when objective is product of two linear functions (Kern and Woeginger (2006), Genc-Kaya, Goyal and Ravi (2008)).

Question: What happens when we have more than two (but a fixed number) of linear functions in the product?

Minimizing Product of Discrete Linear Functions

Given a graph $G = (V, E)$ and two edge weights a, b.

- Find an $s - t$ P path that minimizes $a(P).b(P)$.
- Find a spanning tree T that minimizes $a(T).b(T)$.

Known: FPTAS when objective is product of two linear functions (Kern and Woeginger (2006), Genc-Kaya, Goyal and Ravi (2008)).

Question: What happens when we have more than two (but a fixed number) of linear functions in the product?

Our Results

- FPTAS for combinatorial optimization with rational objective under very general conditions.
- FPTAS for the Santa Claus problem and the budgeted resource allocation problem with fixed number of agents.
- FPTAS for the product optimization problem when the number of linear functions in the product is fixed.

Our Results

- FPTAS for combinatorial optimization with rational objective under very general conditions.
- FPTAS for the Santa Claus problem and the budgeted resource allocation problem with fixed number of agents.
- FPTAS for the product optimization problem when the number of linear functions in the product is fixed.

Our Results

- FPTAS for combinatorial optimization with rational objective under very general conditions.
- FPTAS for the Santa Claus problem and the budgeted resource allocation problem with fixed number of agents.
- FPTAS for the product optimization problem when the number of linear functions in the product is fixed.

The General Framework

Designing FPTAS for the following general problem:

$$\text{minimize } g(x) = h(f_1(x), \ldots, f_m(x)), \qquad x \in X.$$

- Each f_i non-negative function computable in polynomial time.
- h is a function used for combining $f_1, \ldots, f_m$ (e.g. l_p norm).
- m **is fixed.**

Main Idea of the Algorithm

- Each function $f_i(x)$ is considered as a separate objective.
- Compute approximate Pareto-optimal curve in polynomial time (Safer and Orlin (1995), Papadimitriou and Yannakakis (2000), Safer, Orlin and Dror (2004)).
- Find an approximate solution in the approximate Pareto-optimal curve.

Special Case: l_∞ norm

$$\text{minimize } g(x) = \max(f_1(x), \ldots, f_m(x)), \qquad x \in X.$$

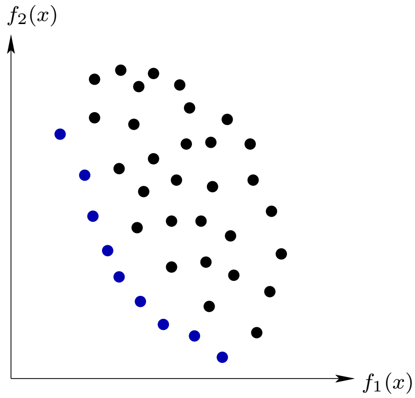
- Each f_i non-negative function computable in polynomial time.
- X is a discrete set.
- m **is fixed.**

π : specific instance of a problem.

$|\pi|$: the input problem size.

Pareto-optimal Front

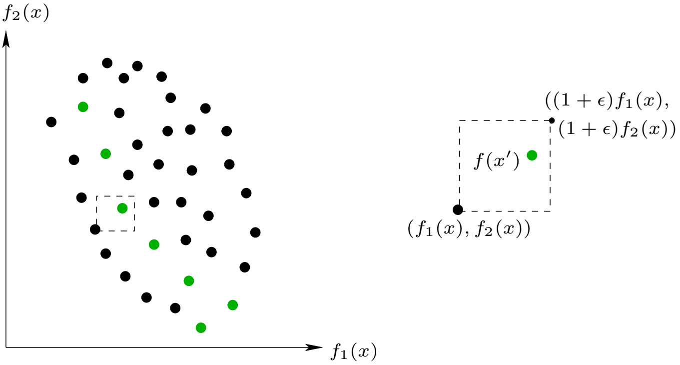
Pareto-optimal front ($P(\pi)$) is the set of all non-dominated solution points.



Approximate Pareto-optimal Front

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

for all $x \in X$, there is $x' \in P_\epsilon(\pi)$ such that $f_i(x') \leq (1+\epsilon) f_i(x)$.



Why Approximate Pareto Front?

For many multi-objective combinatorial problems,

- It may not be tractable to compute $P(\pi)$.
- $P(\pi)$ may contain exponentially many points.

Example: two-objective shortest path problem (Hansen 1979).

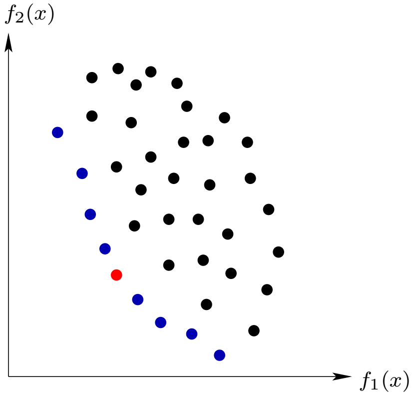
However:

For a **fixed number** m **of objectives**, there is a $P_\epsilon(\pi)$ whose cardinality is bounded by a polynomial in $|\pi|$ and $1/\epsilon$.

(Papadimitriou and Yannakakis 2000).

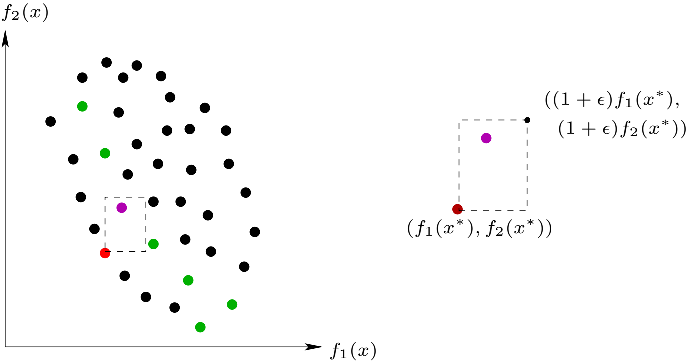
A Lemma

An optimal solution of the problem $\min g(x) = \max_{i=1,\ldots,m} f_i(x)$ lies on $P(\pi)$.



Another Lemma

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



Usefulness of the Lemma

- Holds true for any l_p norm.
- Holds true if $g(x) = f_1(x) \ldots f_p(x)$.
- Holds true if $g(x) = \frac{f_1(x)}{f_2(x)}$.

So, how do we compute the approximate Pareto-optimal front?

Usefulness of the Lemma

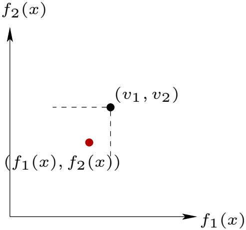
- Holds true for any l_p norm.
- Holds true if $g(x) = f_1(x) \dots f_p(x)$.
- Holds true if $g(x) = \frac{f_1(x)}{f_2(x)}$.

So, how do we compute the approximate Pareto-optimal front?

The Gap Theorem [PY'00]

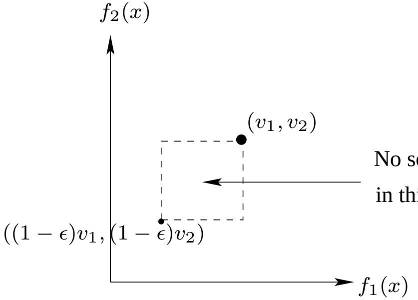
For a fixed m, it is possible to find a $P_\epsilon(\pi)$ in time polynomial in $|\pi|$ and $1/\epsilon$ *if and only if* the following "gap problem" can be solved in polynomial time.

Gap problem: Given an m vector of values $(v_1, \ldots, v_m)$, either (i) return a solution $x \in X$ such that $f_i(x) \leq v_i$ for all $i = 1, \ldots, m$, or .....



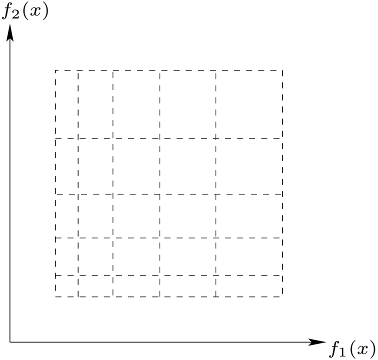
The Gap Theorem (contd.)

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



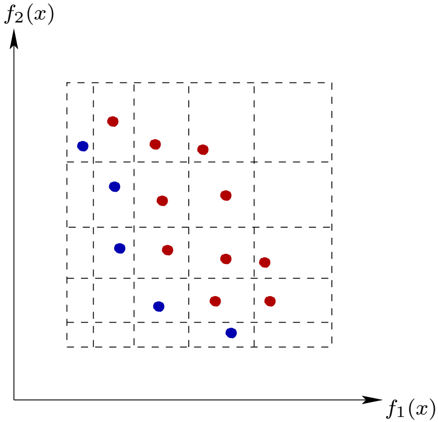
Proof for the 'If' Part

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



Proof for the 'If' Part (contd.)

For each corner point, solve the gap problem, and keep only the undominated set of solutions.



A Sufficient Condition for Solving Gap Problem

The setting:

- the functions $f_i(x)$ are linear functions
- X is a discrete set ($\{0, 1\}^d$)

then 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

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

A Sufficient Condition for Solving Gap Problem

The setting:

- the functions $f_i(x)$ are linear functions
- X is a discrete set ($\{0, 1\}^d$)

then 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

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

Proof Outline for Sufficiency of Given Condition

- Scale the coefficients in f_i to get f_i', in which the maximum magnitude of coefficients is $r = \lceil d/\epsilon \rceil$.
- Find x such that $f_i' \leq r$ for all i, or assert no such x exists.
- $(r+1)^m$ ways of having $f_i'(x) = b_i$, $b_i \leq r$
- Check for each case by finding an x such that $\sum_{i=1}^{m} M^{i-1} f_i'(x) = \sum_{i=1}^{m} M^{i-1} b_i$, M is maximum value f' can take.

Proof Outline for Sufficiency of Given Condition

- Scale the coefficients in f_i to get f_i', in which the maximum magnitude of coefficients is $r = \lceil d/\epsilon \rceil$.
- Find x such that $f_i' \leq r$ for all i, or assert no such x exists.
- $(r+1)^m$ ways of having $f_i'(x) = b_i$, $b_i \leq r$
- Check for each case by finding an x such that $\sum_{i=1}^{m} M^{i-1} f_i'(x) = \sum_{i=1}^{m} M^{i-1} b_i$, M is maximum value f' can take.

Proof Outline for Sufficiency of Given Condition

- Scale the coefficients in f_i to get f_i', in which the maximum magnitude of coefficients is $r = \lceil d/\epsilon \rceil$.
- Find x such that $f_i' \leq r$ for all i, or assert no such x exists.
- $(r+1)^m$ ways of having $f_i'(x) = b_i$, $b_i \leq r$
- Check for each case by finding an x such that $\sum_{i=1}^{m} M^{i-1} f_i'(x) = \sum_{i=1}^{m} M^{i-1} b_i$, M is maximum value f' can take.

Proof Outline for Sufficiency of Given Condition

- Scale the coefficients in f_i to get f'_i, in which the maximum magnitude of coefficients is $r = \lceil d/\epsilon \rceil$.
- Find x such that $f'_i \leq r$ for all i, or assert no such x exists.
- $(r+1)^m$ ways of having $f'_i(x) = b_i$, $b_i \leq r$
- Check for each case by finding an x such that $\sum_{i=1}^{m} M^{i-1} f'_i(x) = \sum_{i=1}^{m} M^{i-1} b_i$, M is maximum value f' can take.

Exact Problem for the Santa Claus Problem

Given an integer C, does there exist a $0/1$-vector such that

$$\begin{aligned} \sum_{k=1}^{n}\sum_{j=1}^{m} c_{jk}x_{jk} &= C, \\ \text{subject to} \sum_{j=1}^{m} x_{jk} &= 1, \quad \text{for} \quad k = 1, \ldots, n. \end{aligned}$$

- This exact problem can be formulated as a reachability problem in a directed graph.
- By using depth first search we can solve the exact problem in *pseudo-polynomial time*.

Exact Problem for the Santa Claus Problem

Given an integer C, does there exist a $0/1$-vector such that

$$\begin{aligned} \sum_{k=1}^{n}\sum_{j=1}^{m} c_{jk}x_{jk} &= C, \\ \text{subject to} \sum_{j=1}^{m} x_{jk} &= 1, \quad \text{for} \quad k = 1, \ldots, n. \end{aligned}$$

- This exact problem can be formulated as a reachability problem in a directed graph.
- By using depth first search we can solve the exact problem in *pseudo-polynomial time*.

Solving Exact Problem for Other Cases

- Spanning Tree: Barahona and Pulleyblank (1987)
- $s - t$ path: Dynamic Programming
- Knapsack problem: Dynamic Programming

Overview of the FPTAS

- Divide the solution space into (polynomially many) smaller hyper-rectangles.
- For each corner point, solve the gap problem and keep only the undominated set of solutions.
- Each gap problem can be solved by making (polynomially many) calls to the (pseudo-polynomial time) algorithm for exact problem. Each input to this algorithm has numbers whose magnitude is polynomial in input size.

Thank You!

Questions??