Approximability of the Two-Stage Knapsack problem with discretely distributed weights

Stefanie Kosuch
Postdoc Technical Computer Science Lab
Linköpings Universitet (Sweden)

10-th Cologne-Twente Workshop
on graphs and combinatorial optimization
Rom, Italy, June 14 - 16, 2011

1 The Two-Stage Knapsack Problem

Linköping University

Linköping University

Outline

The Deterministic Knapsack Problem

- $c > 0$: Knapsack weight capacity
- n items
- $r_i > 0$: reward of item i
- w_i: weight of item i

Objective

Maximize the total reward of chosen items whose total weight respect knapsack capacity.

Applications

Logistics - Resource allocation - Scheduling - Network Optimization etc.

Linköping University

The **Stochastic** Knapsack Problem **with Random Weights**

- $c > 0$: Knapsack weight capacity
- n items
- $r_i > 0$: reward of item i
- χ_i: **random** weight of item i

Objective

Maximize the total reward of chosen items whose total weight respect knapsack capacity.

The **Stochastic** Knapsack Problem **with Random Weights**

- $c > 0$: Knapsack weight capacity
- n items
- $r_i > 0$: reward of item i
- χ_i: **random** weight of item i
  weight unknown when decision has to be made

Objective

Maximize the total reward of chosen items whose total weight respect knapsack capacity.

Linköping University

The **Stochastic** Knapsack Problem **with Random Weights**

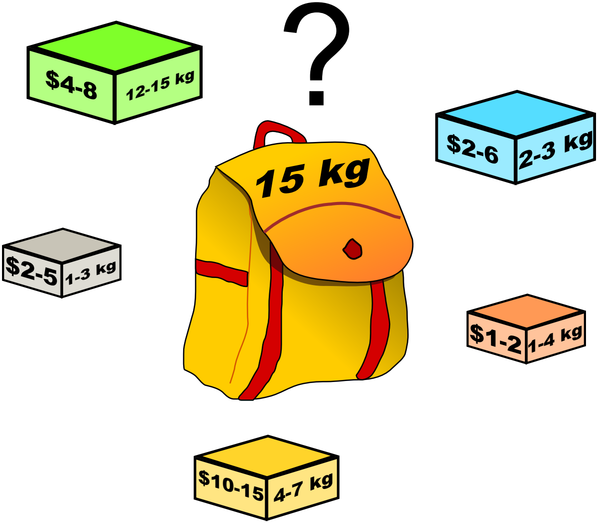
- $c > 0$: Knapsack weight capacity
- n items
- $r_i > 0$: reward of item i
- χ_i: **random** weight of item i
  weight unknown when decision has to be made

Objective

Maximize the total reward of chosen items whose total weight respect knapsack capacity.

Question

How to handle the fact that chosen items might not respect knapsack capacity?


$4-8
12-15 kg
?
$2-6
2-3 kg
15 kg
$2-5
1-3 kg
$1-2
1-4 kg
$10-15
4-7 kg

Two-Stage Setting

Two-Stage Setting

- First stage: items can be put in the knapsack

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: The decision can/has to be corrected

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.
- Correction of the decision causes penalty

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.
- Correction of the decision causes penalty

Assumption: Discretely distributed weights

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.
- Correction of the decision causes penalty

Assumption: Discretely distributed weights

- K **scenarios**

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.
- Correction of the decision causes penalty

Assumption: Discretely distributed weights

- K **scenarios**
- K **realizations** $\chi^1, \dots, \chi^K$

Linköping University

Two-Stage Setting

- First stage: items can be put in the knapsack
- First stage $\longleftrightarrow$ second stage: item weights are revealed
- Second stage: Items
  ...have to be removed in case of an overweight
  ...can be added if capacity sufficient
  ...can be exchanged to increase gain.
- Correction of the decision causes penalty

Assumption: Discretely distributed weights

- K **scenarios**
- K **realizations** $\chi^1, \ldots, \chi^K$
- $\mathbb{P}\{\chi = \chi^k\} = p^k$

Application: Travel Agency

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size
- Randomness e.g., cancellations

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size
- Randomness e.g., cancellations
- Agency allows overbooking

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size
- Randomness e.g., cancellations
- Agency allows overbooking
- Number of beds insufficient

Linköping University

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size
- Randomness e.g., cancellations
- Agency allows overbooking
- Number of beds insufficient
  $\rightarrow$ groups have to be relocated in other hotels

Application: Travel Agency

- Knapsack $\simeq$ Hotel Complex
- Weight capacity $\simeq$ Total number of beds
- Items $\simeq$ Travel groups
- Item weights $\simeq$ Group size
- Randomness e.g., cancellations
- Agency allows overbooking
- Number of beds insufficient
  $\rightarrow$ groups have to be relocated in other hotels
- Vacant beds filled with last minute offers

Two-Stage Knapsack Problem

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} r_i x_i$$

s.t.

x: decision vector of 1st stage

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x}, \chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x, \chi) = \max_{\mathbf{y}^+, \mathbf{y}^- \in \{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_\mathbf{i} \mathbf{y}_\mathbf{i}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_\mathbf{i} \mathbf{y}_\mathbf{i}^-,$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x\in\{0,1\}^n} \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x},\chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x,\chi) = \max_{\mathbf{y}^+,\mathbf{y}^-\in\{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_\mathbf{i}\mathbf{y}_\mathbf{i}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_\mathbf{i}\mathbf{y}_\mathbf{i}^-,$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x \in \{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x}, \chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x, \chi) = \max_{\mathbf{y}^+, \mathbf{y}^- \in \{0,1\}^n} \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \bar{\mathbf{r}}_\mathbf{i} \mathbf{y}_\mathbf{i}^+ - \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathbf{d}_\mathbf{i} \mathbf{y}_\mathbf{i}^-,$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x}, \chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x, \chi) = \max_{\mathbf{y}^+, \mathbf{y}^- \in \{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_{\mathbf{i}} \mathbf{y}_{\mathbf{i}}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_{\mathbf{i}} \mathbf{y}_{\mathbf{i}}^-,$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_{\mathbf{i}} < \mathbf{r}_{\mathbf{i}}$, $\mathbf{d}_{\mathbf{i}} > \mathbf{r}_{\mathbf{i}}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x \in \{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x}, \chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x, \chi) = \max_{\mathbf{y}^+, \mathbf{y}^- \in \{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_\mathbf{i} \mathbf{y}_\mathbf{i}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_\mathbf{i} \mathbf{y}_\mathbf{i}^-,$$

$$\text{s.t.} \quad y_j^+ \leq 1 - x_j, \quad j = 1, \ldots, n,$$

$$y_j^- \leq x_j, \quad j = 1, \ldots, n,$$

x: decision vector of 1^{st} stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2^{nd} stage (recourse action)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x\in\{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x},\chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x,\chi) = \max_{\mathbf{y}^+,\mathbf{y}^-\in\{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_{\mathbf{i}}\mathbf{y}_{\mathbf{i}}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_{\mathbf{i}}\mathbf{y}_{\mathbf{i}}^-,$$

$$\text{s.t.} \quad y_j^+ \leq 1 - x_j, \quad j = 1,\ldots,n,$$

$$y_j^- \leq x_j, \quad j = 1,\ldots,n,$$

$$\sum_{i=1}^{n}(x_i + y_i^+ - y_i^-)\chi_i \leq c.$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_{\mathbf{i}} < \mathbf{r}_{\mathbf{i}}$, $\mathbf{d}_{\mathbf{i}} > \mathbf{r}_{\mathbf{i}}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x\in\{0,1\}^n} \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(\mathbf{x},\chi)]$$

$$\text{s.t.} \quad \mathcal{Q}(x,\chi) = \max_{\mathbf{y}^+,\mathbf{y}^-\in\{0,1\}^n} \sum_{\mathbf{i=1}}^{\mathbf{n}} \bar{\mathbf{r}}_\mathbf{i}\mathbf{y}_\mathbf{i}^+ - \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d}_\mathbf{i}\mathbf{y}_\mathbf{i}^-,$$

$$\text{s.t.} \quad y_j^+ \leq 1 - x_j, \quad j = 1,\dots,n,$$

$$y_j^- \leq x_j, \quad j = 1,\dots,n,$$

$$\sum_{i=1}^{n} (x_i + y_i^+ - y_i^-)\chi_i \leq c.$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Linköping University

Two-Stage Knapsack Problem

$$(TSKP) \quad \max_{x\in\{0,1\}^n} \sum_{i=1}^{n} r_i x_i + \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{K}} \mathbf{p}^{\mathbf{k}} \mathcal{Q}(\mathbf{x}, \chi^{\mathbf{k}})$$

$$\text{s.t.} \quad \mathcal{Q}(x,\chi) = \max_{\mathbf{y}^+,\mathbf{y}^-\in\{0,1\}^n} \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \bar{\mathbf{r}}_{\mathbf{i}} \mathbf{y}_{\mathbf{i}}^+ - \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathbf{d}_{\mathbf{i}} \mathbf{y}_{\mathbf{i}}^-,$$

$$\text{s.t.} \quad y_j^+ \leq 1 - x_j, \quad j = 1, \dots, n,$$

$$y_j^- \leq x_j, \quad j = 1, \dots, n,$$

$$\sum_{i=1}^{n} (x_i + y_i^+ - y_i^-)\chi_i \leq c.$$

x: decision vector of 1st stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2nd stage (recourse action)
$\bar{\mathbf{r}}_{\mathbf{i}} < \mathbf{r}_{\mathbf{i}}$, $\mathbf{d}_{\mathbf{i}} > \mathbf{r}_{\mathbf{i}}$

Linköping University

Deterministic Equivalent Two-Stage Knapsack Problem

Linköping University

Deterministic Equivalent Two-Stage Knapsack Problem

$$
\begin{aligned}
(TSK^D) \quad \max \quad & \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} \overline{r}_i (\mathbf{y}^+)_\mathbf{i}^\mathbf{k} - \sum_{i=1}^{n} d_i (\mathbf{y}^-)_\mathbf{i}^\mathbf{k} \right) \\
\text{s.t.} \quad & (\mathbf{y}^+)_\mathbf{j}^\mathbf{k} \leq 1 - x_j && j = 1, \dots, n, \ \mathbf{k} = \mathbf{1}, \dots, \mathbf{K}, \\
& (\mathbf{y}^-)_\mathbf{j}^\mathbf{k} \leq x_j && j = 1, \dots, n, \ \mathbf{k} = \mathbf{1}, \dots, \mathbf{K}, \\
& \sum_{i=1}^{n} (x_i + (\mathbf{y}^+)_\mathbf{i}^\mathbf{k} - (\mathbf{y}^-)_\mathbf{i}^\mathbf{k}) \chi_i^k \leq c && \mathbf{k} = \mathbf{1}, \dots, \mathbf{K}, \\
& x \in \{0,1\}^n, \\
& (\mathbf{y}^+)^\mathbf{k}, (\mathbf{y}^-)^\mathbf{k} \in \{0,1\}^n && \mathbf{k} = \mathbf{1}, \dots, \mathbf{K}.
\end{aligned}
$$

x: decision vector of 1st stage

Deterministic Equivalent Two-Stage Knapsack Problem

$$
\begin{aligned}
(TSK^D) \quad \max \quad & \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} \overline{r}_i (\mathbf{y}^+)_{\mathbf{i}}^{\mathbf{k}} - \sum_{i=1}^{n} d_i (\mathbf{y}^-)_{\mathbf{i}}^{\mathbf{k}} \right) \\
\text{s.t.} \quad & (\mathbf{y}^+)_{\mathbf{j}}^{\mathbf{k}} \leq 1 - x_j \qquad j = 1, \ldots, n, \ \mathbf{k} = \mathbf{1}, \ldots, \mathbf{K}, \\
& (\mathbf{y}^-)_{\mathbf{j}}^{\mathbf{k}} \leq x_j \qquad j = 1, \ldots, n, \ \mathbf{k} = \mathbf{1}, \ldots, \mathbf{K}, \\
& \sum_{i=1}^{n} (x_i + (\mathbf{y}^+)_{\mathbf{i}}^{\mathbf{k}} - (\mathbf{y}^-)_{\mathbf{i}}^{\mathbf{k}}) \chi_i^k \leq c \qquad \mathbf{k} = \mathbf{1}, \ldots, \mathbf{K}, \\
& x \in \{0,1\}^n, \\
& (\mathbf{y}^+)^{\mathbf{k}}, (\mathbf{y}^-)^{\mathbf{k}} \in \{0,1\}^n \qquad \mathbf{k} = \mathbf{1}, \ldots, \mathbf{K}.
\end{aligned}
$$

x: decision vector of 1^{st} stage
$(y^+)^k, (y^-)^k$: decision vectors in scenario k

Linköping University

Deterministic Equivalent Two-Stage Knapsack Problem

$$(TSK^D) \quad \max \quad \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} \overline{r}_i (\mathbf{y}^+)_\mathbf{i}^\mathbf{k} - \sum_{i=1}^{n} d_i (\mathbf{y}^-)_\mathbf{i}^\mathbf{k} \right)$$

$$\text{s.t.} \quad (\mathbf{y}^+)_\mathbf{j}^\mathbf{k} \le 1 - x_j \qquad j = 1, \dots, n, \ \mathbf{k} = \mathbf{1}, \dots, \mathbf{K},$$

$$(\mathbf{y}^-)_\mathbf{j}^\mathbf{k} \le x_j \qquad j = 1, \dots, n, \ \mathbf{k} = \mathbf{1}, \dots, \mathbf{K},$$

$$\sum_{i=1}^{n} (x_i + (\mathbf{y}^+)_\mathbf{i}^\mathbf{k} - (\mathbf{y}^-)_\mathbf{i}^\mathbf{k}) \chi_i^k \le c \qquad \mathbf{k} = \mathbf{1}, \dots, \mathbf{K},$$

$$x \in \{0,1\}^n,$$

$$(\mathbf{y}^+)^\mathbf{k}, (\mathbf{y}^-)^\mathbf{k} \in \{0,1\}^n \qquad \mathbf{k} = \mathbf{1}, \dots, \mathbf{K}.$$

x: decision vector of 1st stage
$(y^+)^k, (y^-)^k$: decision vectors in scenario k

Deterministic Reformulation...

Deterministic Reformulation...

- ...has linear objective and constraints.

Deterministic Reformulation...

- ...has linear objective and constraints.
- ...has $(2n+1)K$ constraints.

Linköping University

Deterministic Reformulation...

- ...has linear objective and constraints.
- ...has $(2n+1)K$ constraints.
- ...has $(2K+1)n$ binary decision variables.

Deterministic Reformulation...

- ...has linear objective and constraints.
- ...has $(2n+1)K$ constraints.
- ...has $(2K+1)n$ binary decision variables.
- ...can have **"exponential size"**.

Deterministic Reformulation...

- ...has linear objective and constraints.
- ...has $(2n+1)K$ constraints.
- ...has $(2K+1)n$ binary decision variables.
- ...can have **"exponential size"**.
- ...in general intractable with exact solvers.

Published and Working Papers treating the TSK^D

Published and Working Papers treating the TSK^D

- R. Lopez (2009): **Stochastic Quadratic Knapsack Problems and Semidefinite Programming**, *Thesis at the LRI, Université Paris Sud, France*

Published and Working Papers treating the TSK^D

- R. Lopez (2009): **Stochastic Quadratic Knapsack Problems and Semidefinite Programming**, *Thesis at the LRI, Université Paris Sud, France*
- A. Gaivoronski, A. Lisser, R. Lopez and X. Hu (2010): **Knapsack problem with probability constraints**, *Journal of Global Optimization 49(3)*

Published and Working Papers treating the TSK^D

- R. Lopez (2009): **Stochastic Quadratic Knapsack Problems and Semidefinite Programming**, *Thesis at the LRI, Université Paris Sud, France*
- A. Gaivoronski, A. Lisser, R. Lopez and X. Hu (2010): **Knapsack problem with probability constraints**, *Journal of Global Optimization 49(3)*
- S. Kosuch (2011): **Towards an Ant Colony Optimization algorithm for the Two-Stage Knapsack problem**, *Proc. of the VII. ALIO/EURO Workshop on Applied Combinatorial Optimization*

Linköping University

Outline

Theorem (Main Non-Approximability Result)

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Definition

Definition

We denote *AddTSKP* ($AddTSK^D$) the variant of *TSKP* ($AddTSK^D$) where in the second stage items can only be added.

Definition

We denote *AddTSKP* (*AddTSK*^{D}) the variant of *TSKP* (*AddTSK*^{D}) where in the second stage items can only be added.

Observations

- No relatively complete recourse.

Definition

We denote *AddTSKP* ($AddTSK^D$) the variant of *TSKP* ($AddTSK^D$) where in the second stage items can only be added.

Observations

- No relatively complete recourse.
- First-stage decision $\rightarrow$ infeasible second stage problem $\Rightarrow$ $f(x) = -\infty$

Definition

We denote *AddTSKP* ($AddTSK^D$) the variant of *TSKP* ($AddTSK^D$) where in the second stage items can only be added.

Observations

- No relatively complete recourse.
- First-stage decision $\rightarrow$ infeasible second stage problem $\Rightarrow$ $f(x) = -\infty$
- Optimal solution of *AddTSKP* always respects capacity

Multiply Constrained Knapsack Problem

Multiply Constrained Knapsack Problem

$$(MCKP) \quad \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} r_i x_i$$

Multiply Constrained Knapsack Problem

$$(MCKP) \quad \max_{x\in\{0,1\}^n} \sum_{i=1}^{n} r_i x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i w_i^j \leq c \quad \forall j = 1, \ldots, m.$$

Multiply Constrained Knapsack Problem

$$(MCKP) \quad \max_{x\in\{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n} x_i w_i^j \leq c \quad \forall j = 1, \ldots, m.$$

Theorem (Z. Li'ang & Z. Yin, 1999)

For any $\epsilon > 0$, the multiply-constrained knapsack problem does not admit a $m^{-\frac{1}{4}+\epsilon}$-approximation algorithm unless $\mathcal{P} = \mathcal{NP}$.

Linköping University

Multiply Constrained Knapsack Problem

$$(MCKP) \quad \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} r_i x_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n} x_i w_i^j \leq c \quad \forall j = 1, \ldots, m.$$

Corollary (Z. Li'ang & Z. Yin, 1999 + D. Zuckerman, 2006)

For any $\epsilon > 0$, the multiply-constrained knapsack problem does not admit a $m^{-\frac{1}{2}+\epsilon}$-approximation algorithm unless $\mathcal{P} = \mathcal{NP}$.

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Linköping University

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- *MCKP*

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- *MCKP*

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- *MCKP* $\leftarrow$

Linköping University

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- $MCKP \quad \leftarrow \quad AddTSK^D$

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- $MCKP \quad \leftarrow \quad AddTSK^D \quad \leftarrow$

Linköping University

Theorem (Main Non-Approximability Result)

For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the TSKP, unless $\mathcal{P} = \mathcal{NP}$.

Schema of the theorem's proof

- Reduction from the *MCKP*
- $MCKP \quad \leftarrow \quad AddTSK^D \quad \leftarrow \quad TSK^D$

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.
- Construct instance $\mathcal{I}' = (r, \bar{r}, d, p, w, c)$ of TSK^D s.t.

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.
- Construct instance $\mathcal{I}' = (r, \bar{r}, d, p, w, c)$ of TSK^D s.t.

$$(x^*, (y^+)^*, (y^-)^*) \text{ optimal for } \mathcal{I}' \Longrightarrow (y^-)^* = 0$$

Linköping University

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.
- Construct instance $\mathcal{I}' = (r, \bar{r}, d, p, w, c)$ of TSK^D s.t.

  $$(x^*, (y^+)^*, (y^-)^*) \text{ optimal for } \mathcal{I}' \implies (y^-)^* = 0$$

- Idea:

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.
- Construct instance $\mathcal{I}' = (r, \bar{r}, d, p, w, c)$ of TSK^D s.t.

$$(x^*, (y^+)^*, (y^-)^*) \text{ optimal for } \mathcal{I}' \Longrightarrow (y^-)^* = 0$$

- Idea:

Linköping University

Reformulate an instance of $AddTSK^D$ via an instance of TSK^D (idea)

- Given instance $\mathcal{I} = (r, \bar{r}, p, w, c)$ of $AddTSK^D$.
- Construct instance $\mathcal{I}' = (r, \bar{r}, d, p, w, c)$ of TSK^D s.t.

$$(x^*, (y^+)^*, (y^-)^*) \text{ optimal for } \mathcal{I}' \Longrightarrow (y^-)^* = 0$$

- Idea: Choose d_i $(i = 1, \dots, n)$ large enough!

Solve an instance of *MCKP* as an instance of $AddTSK^D$ (idea)

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.
- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of *AddTSK*^{D} s.t.

Linköping University

Solve an instance of *MCKP* as an instance of $AddTSK^D$ (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.
- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of $AddTSK^D$ s.t.

$$(x^*, (y^+)^*) \text{ optimal for } \mathcal{I}' \Longrightarrow (y^+)^* = 0$$

Linköping University

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.

- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of *AddTSK*^{D} s.t.

$$(x^*, (y^+)^*) \text{ optimal for } \mathcal{I}' \Longrightarrow (y^+)^* = 0$$

- Idea: Choose $\bar{r}_i$ $(i = 1, \ldots, n)$ small enough!

Linköping University

Solve an instance of *MCKP* as an instance of $AddTSK^D$ (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.

- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of $AddTSK^D$ s.t.

  ~~$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow (y^+)^* = 0$~~

- Idea: Choose $\bar{r}_i$ $(i = 1, \ldots, n)$ small enough!

Linköping University

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.

- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of *AddTSK*^{D} s.t.

~~$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow (y^+)^* = 0$~~

$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow x^*$ optimal for $\mathcal{I}$

- Idea: Choose $\bar{r}_i$ $(i = 1, \ldots, n)$ small enough!

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.
- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of *AddTSK*^{D} s.t.

~~$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow (y^+)^* = 0$~~

$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow x^*$ optimal for $\mathcal{I}$

- Idea: Choose $\bar{r}_i$ $(i = 1, \ldots, n)$ small enough!
  $r \in \mathbb{Z}_+^n \rightarrow \bar{r}_i = \frac{1}{n+1}$ $(i = 1, \ldots, n)$

Linköping University

Solve an instance of *MCKP* as an instance of *AddTSK*^{D} (idea)

- Given instance $\mathcal{I} = (r, w, c)$ of *MCKP*.

- Construct instance $\mathcal{I}' = (r, \bar{r}, p, w, c)$ of *AddTSK*^{D} s.t.

  ~~$(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow (y^+)^* = 0$~~

  $(x^*, (y^+)^*)$ optimal for $\mathcal{I}' \Longrightarrow x^*$ optimal for $\mathcal{I}$

- Idea: Choose $\bar{r}_i$ $(i = 1, \ldots, n)$ small enough!

  $r \in \mathbb{Z}^n_+ \rightarrow \bar{r}_i = \frac{1}{n+1}$ $(i = 1, \ldots, n)$

$$\Rightarrow \sum_{j=1}^{m} \frac{1}{m} \sum_{i=1}^{n} \frac{1}{n+1} < 1$$

Linköping University

Outline

The *TSKP* with linearly dependent first- and second-stage reward vectors

The *TSKP* with linearly dependent first- and second-stage reward vectors

Lemma

Let $\alpha \in (0,1)$ *and denote TSKP*$(\alpha, \cdot)$ *the variant of the TSKP such that* $\bar{r} = \alpha \cdot r$. *Then there exists an approximation algorithm for the TSKP*$(\alpha, \cdot)$ *with approximation ratio* α.

The *TSKP* with linearly dependent first- and second-stage reward vectors

Lemma

Let $\alpha \in (0,1)$ and denote TSKP$(\alpha, \cdot)$ the variant of the TSKP such that $\bar{r} = \alpha \cdot r$. Then there exists an approximation algorithm for the TSKP$(\alpha, \cdot)$ with approximation ratio α.

Algorithm

The *TSKP* with linearly dependent first- and second-stage reward vectors

Lemma

Let $\alpha \in (0, 1)$ and denote TSKP$(\alpha, \cdot)$ the variant of the TSKP such that $\bar{r} = \alpha \cdot r$. Then there exists an approximation algorithm for the TSKP$(\alpha, \cdot)$ with approximation ratio α.

Algorithm

Add no item in the first stage.

The *TSKP* with linearly dependent first- and second-stage reward vectors

Lemma

Let $\alpha \in (0,1)$ and denote TSKP$(\alpha,\cdot)$ the variant of the TSKP such that $\bar{r} = \alpha \cdot r$. Then there exists an approximation algorithm for the TSKP$(\alpha,\cdot)$ with approximation ratio α.

Proposition (Genaralisation of the lemma)

For any instance of the TSKP define $\alpha := \min_{i=1\ldots,n} \frac{\bar{r}_i}{r_i}$. Then adding no items in the first stage always yields a solution whose solution value is at least an α-fraction of the optimal solution value.

Linköping University

The *TSKP* under the assumption of a polynomial scenario model

The *TSKP* under the assumption of a polynomial scenario model

Proposition

The *TSKP* under the assumption of a polynomial scenario model

Proposition

Under the assumption of a polynomial scenario model, the TSKP admits a $\frac{1}{n}$*-approximation algorithm.*

The *TSKP* under the assumption of a polynomial scenario model

Proposition

Under the assumption of a polynomial scenario model, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Algorithm

The *TSKP* under the assumption of a polynomial scenario model

Proposition

Under the assumption of a polynomial scenario model, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Algorithm

Determine the item that gives us the highest expected reward when added all alone (in first- or second stage).

The *TSKP* under the assumption of a polynomial scenario model

Proposition

Under the assumption of a polynomial scenario model, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Algorithm

Determine the item that gives us the highest expected reward when added all alone (in first- or second stage).

Corollary

If the item weights are independently distributed with polynomial number of realizations, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Linköping University

The *TSKP* under the assumption of a polynomial scenario model

Proposition

Under the assumption of a polynomial scenario model, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Algorithm

Determine the item that gives us the highest expected reward when added all alone (in first- or second stage).

Corollary

If the item weights are independently distributed with polynomial number of realizations, the TSKP admits a $\frac{1}{n}$-approximation algorithm.

Linköping University

The *AddTSKP* under the assumption of a fix number of scenarios

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

For all $\epsilon > 0$, there exists an approximation algorithm for the $K-AddTSKP$ with approximation-ratio $\frac{1}{2} - \epsilon$.

Linköping University

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

For all $\epsilon > 0$, there exists an approximation algorithm for the $K - AddTSKP$ with approximation-ratio $\frac{1}{2} - \epsilon$.

Algorithm

Linköping University

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

For all $\epsilon > 0$, there exists an approximation algorithm for the $K - AddTSKP$ with approximation-ratio $\frac{1}{2} - \epsilon$.

Algorithm

- Solve the *MCKP* (r, w, c) with *PTAS*.

Linköping University

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

For all $\epsilon > 0$, there exists an approximation algorithm for the $K-$ AddTSKP with approximation-ratio $\frac{1}{2} - \epsilon$.

Algorithm

- Solve the *MCKP* (r, w, c) with *PTAS*.
- $\forall k$, solve *KP* $(\bar{r}, w^k, c)$ with *FPTAS*.

Linköping University

The *AddTSKP* under the assumption of a fix number of scenarios

Proposition

For all $\epsilon > 0$, there exists an approximation algorithm for the $K - AddTSKP$ with approximation-ratio $\frac{1}{2} - \epsilon$.

Algorithm

- Solve the *MCKP* (r, w, c) with *PTAS*.
- $\forall k$, solve *KP* $(\bar{r}, w^k, c)$ with *FPTAS*.
- Either output solution of former, or 0.

Linköping University

Outline

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).
- Reduction from the multiply constrained knapsack problem

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).
- Reduction from the multiply constrained knapsack problem
- Approximation algorithms for special cases:

Linköping University

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).
- Reduction from the multiply constrained knapsack problem
- Approximation algorithms for special cases:
  $\rightarrow$ Linear dependency first- and second-stage rewards

Linköping University

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).
- Reduction from the multiply constrained knapsack problem
- Approximation algorithms for special cases:
  - $\rightarrow$ Linear dependency first- and second-stage rewards
  - $\rightarrow$ Polynomial scenario model

Recapitulation: Two-Stage Knapsack Problem with discrete distribution

- TSK^D cannot be approximated in polynomial time within a ratio better than $K^{-\frac{1}{2}}$ (unless $\mathcal{P} = \mathcal{NP}$).
- Reduction from the multiply constrained knapsack problem
- Approximation algorithms for special cases:
  - → Linear dependency first- and second-stage rewards
  - → Polynomial scenario model
  - → Fixed number of scenarios ($K - AddTSKP$)

Linköping University

Future Work

Future Work

- More complex approximation algorithms for special cases

Future Work

- More complex approximation algorithms for special cases
- *PTAS* for $K-(Add)TSKP$

Future Work

- More complex approximation algorithms for special cases
- $PTAS$ for $K-(Add)TSKP$
- Approximation algorithm for general case

Linköping University

Future Work

- More complex approximation algorithms for special cases
- *PTAS* for $K-(Add)TSKP$
- Approximation algorithm for general case
- Approximation in case of continuous distributions

Thank you!

:-)

Grazie!

Linköping University