Towards an Ant Colony Optimization algorithm for the Two-Stage Knapsack problem

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

VII ALIO/EURO - Workshop on Appl. Combinatorial Optimization
Porto, Portugal, May 4 - 6, 2011

1 The Two-Stage Knapsack Problem

Linköping University

Linköping University

Outline

Linköping University

The Deterministic Knapsack Problem

The Deterministic Knapsack Problem

- $c > 0$: Knapsack weight capacity

The Deterministic Knapsack Problem

- $c > 0$: Knapsack weight capacity
- n items

The Deterministic Knapsack Problem

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

Linköping University

The Deterministic Knapsack Problem

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

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.

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.

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.

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.

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?

Linköping University

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

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

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.

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** $\chi^1, \ldots, \chi^K$

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

Linköping University

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

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

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

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

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
- 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

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

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

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

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

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

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

Two-Stage Knapsack Problem

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

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

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

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

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

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

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

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

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

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

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

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

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

x: decision vector of 1[st] stage
$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2[nd] stage (recourse action)
$\mathbf{\bar{r}_i} < \mathbf{r_i}$, $\mathbf{d_i} > \mathbf{r_i}$

Linköping University

Two-Stage Knapsack Problem

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

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

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

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

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

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

Two-Stage Knapsack Problem

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

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

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

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

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

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

Outline

Natural idea: try metaheuristics!

But why an **ACO**-metaheuristic?

But why an **ACO**-metaheuristic?

- Possibility to use **heuristic utility measures**

But why an **ACO**-metaheuristic?

- Possibility to use **heuristic utility measures**
- Construction of solution → **no evaluation**

But why an **ACO**-metaheuristic?

- Possibility to use **heuristic utility measures**
- Construction of solution $\rightarrow$ **no evaluation**
- Obj. func. evaluation $\leftarrow$ comparison

Linköping University

Two-Stage Knapsack Problem

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

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

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

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

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

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

Linköping University

But why an **ACO**-metaheuristic?

- Possibility to use **heuristic utility measures**
- Construction of solution $\rightarrow$ **no evaluation**
- Obj. func. evaluation $\leftarrow$ comparison

Linköping University

The search graph

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items
- Add **starting vertex**

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items
- Add **starting vertex**
- Add **termination vertex**

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items
- Add **starting vertex**
- Add **termination vertex**
- Pheromone on **arcs**

Linköping University

Heuristic utility measure I: Problems

Heuristic utility measure I: Problems

- 4 factors to be considered:

Heuristic utility measure I: Problems

- 4 factors to be considered:
  - item weight
  - first-stage reward
  - second-stage reward
  - second-stage penalty

Heuristic utility measure I: Problems

- 4 factors to be considered:
  - item weight
  - first-stage reward
  - second-stage reward
  - second-stage penalty
- no "natural" certificate for termination

Heuristic utility measure I: Problems

- 4 factors to be considered:
  - item weight
  - first-stage reward
  - second-stage reward
  - second-stage penalty
- no "natural" certificate for termination
- utility measure for termination vertex?

Heuristic utility measure II: Propositions

Heuristic utility measure II: Propositions

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple utility measure:

$$\eta_i^S = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i}{\chi_i^k}$$

Heuristic utility measure II: Propositions

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple utility measure:

$$\eta_i^S = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i}{\chi_i^k}$$

- Difference utility measure:

$$\eta_i^D = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i - \overline{r}_i}{\chi_i^k}$$

Heuristic utility measure II: Propositions

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple utility measure:

$$\eta_i^S = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i}{\chi_i^k}$$

- Difference utility measure:

$$\eta_i^D = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i - \overline{r}_i}{\chi_i^k}$$

- Ratio utility measure:

$$\eta_i^R = \sum_{k \in \mathcal{K}_i} p^k \frac{r_i / \overline{r}_i}{\chi_i^k}$$

Heuristic utility measure II: Non-utility measures

Heuristic utility measure II: Non-utility measures

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple non-utility measure:

$$\nu_i^S = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i}{\chi_i^k} \qquad \nu_i^S = \sum_{k=1}^{K} p^k \frac{\bar{r}_i}{\chi_i^k}$$

versity

Heuristic utility measure II: Non-utility measures

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple non-utility measure:

$$\nu_i^S = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i}{\chi_i^k} \qquad \nu_i^S = \sum_{k=1}^{K} p^k \frac{\bar{r}_i}{\chi_i^k}$$

- Difference utility measure:

$$\nu_i^D = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i - r_i}{\chi_i^k}$$

Heuristic utility measure II: Non-utility measures

$\mathcal{K}_i$: set of scenarios where item i still fits

- Simple non-utility measure:

$$\nu_i^S = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i}{\chi_i^k} \qquad \nu_i^S = \sum_{k=1}^{K} p^k \frac{\bar{r}_i}{\chi_i^k}$$

- Difference utility measure:

$$\nu_i^D = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i - r_i}{\chi_i^k}$$

- Ratio utility measure:

$$\nu_i^R = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i / r_i}{\chi_i^k}$$

Heuristic utility measure II: Non-utility measures

$\mathcal{K}_i$: set of scenarios where item i still fits

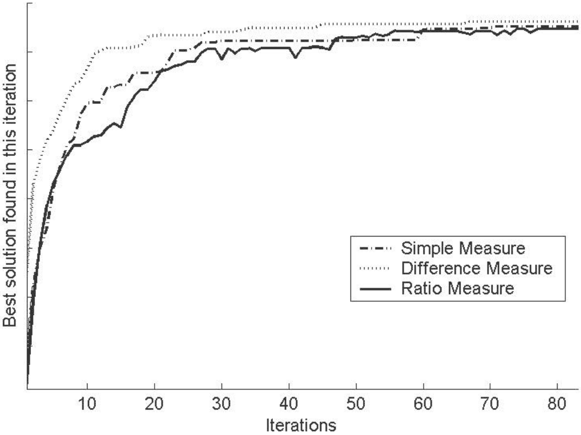
- Simple non-utility measure:
  $$\nu_i^{S_1} = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i}{\chi_i^k} \quad \text{or} \quad \nu_i^{S_2} = \sum_{k=1}^{K} p^k \frac{\bar{r}_i}{\chi_i^k}$$
- Difference non-utility measure:
  $$\nu_i^D = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i - r_i}{\chi_i^k}$$
- Ratio non-utility measure:
  $$\nu_i^R = \sum_{k \notin \mathcal{K}_i} p^k \frac{d_i / r_i}{\chi_i^k}$$

Utility of termination:

$$\eta_{n+1} = min_{i \in \{1,\dots,n\}} \nu_i$$

Outline

Comparison of the different utility measures I



Comparison of the different utility measures II

		Difference measure			Simple measure		
n-K-t	Inst.	Runs	Gap	CPU(s)	Runs	Gap	CPU(s)
100-5-0.25	3/3	57%	0.02%	35	13%	0.05%	30
100-5-0.5	2/3	28%	0.01%	57	1%	0.03%	52
100-5-0.75	1/3	1%	0.02%	69	0%	0.02%	71
100-10-0.25	3/3	93%	0.06%	47	63%	0.01%	34
100-10-0.5	2/3	23%	0.01%	72	0%	0.03%	63
100-10-0.75	1/3	15%	0.02%	85	0%	0.04%	85
100-30-0.25	2/3	58%	0.02%	147	0%	0.12%	107
100-30-0.5	3/3	63%	0.01%	232	8%	0.02%	179
100-30-0.75	1/3	25%	0.01%	295	0%	0.03%	183

Linköping University

Outline

Linköping University

Future work

Future work

- Improve utility measure for higher n

Future work

- Improve utility measure for higher n
- Consider sampling for higher K

Linköping University

Future work

- Improve utility measure for higher n
- Consider sampling for higher K
- Consider using approximate knapsack algorithm for higher n

Future work

- Improve utility measure for higher n
- Consider sampling for higher K
- Consider using approximate knapsack algorithm for higher n
- **Comparison with other metaheuristics**

Linköping University

Thank you!

:-)

Obrigada!