An Ant Colony Optimization Algorithm for the Two-Stage Knapsack Problem

Stefanie Kosuch
Postdoc at the Theoretical Computer Science Lab
Linköpings Universitet (Sweden)

Évolution Artificielle 2011
Angers, France, October 24 - 26, 2011

1 The Two-Stage Knapsack Problem

Linköping University

Linköping University

Outline

Linköping University

The Deterministic Knapsack Problem

Linköping University

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

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.

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?

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

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

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

Linköping University

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 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

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

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

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

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)
$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

Two-Stage Knapsack Problem

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

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

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

Linköping University

Two-Stage Knapsack Problem

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

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

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

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

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

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

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

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

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

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

x: decision vector of 1[st] stage

$\mathbf{y}^+, \mathbf{y}^-$: decision vectors of 2[nd] stage (recourse action)

$\bar{\mathbf{r}}_\mathbf{i} < \mathbf{r}_\mathbf{i}$, $\mathbf{d}_\mathbf{i} > \mathbf{r}_\mathbf{i}$

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

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

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

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

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

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

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

Linköping University

Outline

Linköping University

Natural idea: try metaheuristics!

But why an **ACO**-metaheuristic?

Linköping University

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

Two-Stage Knapsack Problem

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

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

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

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

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

x: decision vector of 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

The search graph

The search graph

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

Linköping University

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items
- Pheromone laid on **arcs** (directed edges)

The search graph

- **Complete directed** search graph: n vertices $\simeq$ n items
- Pheromone laid on **arcs** (directed edges)
- Add **starting vertex**

The search graph

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

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

Linköping University

Heuristic utility measure I: Problems

- 4 factors to be considered:
  - item weight
  - first-stage reward
  - second-stage reward
  - second-stage penalty
- K different weights per item

Heuristic utility measure I: Problems

- 4 factors to be considered:
  - item weight
  - first-stage reward
  - second-stage reward
  - second-stage penalty
- K different weights per item
- 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
- K different weights per item
- no "natural" certificate for termination
- utility measure for termination vertex $\longrightarrow$ via non-utility measure

Heuristic utility measure II: Simple measure

Heuristic utility measure II: Simple measure

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

- Simple utility measure ($i \in \{1, \ldots, n\}$):

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

Heuristic utility measure II: Simple measure

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

- Simple utility measure ($i \in \{1, \ldots, n\}$):

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

- Simple non-utility measure ($i \in \{1, \ldots, n\}$):

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

Heuristic utility measure II: Simple measure

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

- Simple utility measure ($i \in \{1, \ldots, n\}$):

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

- Simple non-utility measure ($i \in \{1, \ldots, n\}$):

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

Utility of termination:

$$\eta_{n+1}^S = min_{i \in \{1,\ldots,n\}} \nu_i^S$$

Heuristic utility measure III: Difference measure

Heuristic utility measure III: Difference measure

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

- Difference utility measure ($i \in \{1, \ldots, n\}$):

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

Heuristic utility measure III: Difference measure

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

- Difference utility measure ($i \in \{1, \ldots, n\}$):

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

- Difference non-utility measure ($i \in \{1, \ldots, n\}$):

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

Heuristic utility measure III: Difference measure

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

- Difference utility measure ($i \in \{1, \ldots, n\}$):

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

- Difference non-utility measure ($i \in \{1, \ldots, n\}$):

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

Utility of termination:

$$\eta_{n+1}^D = min_{i \in \{1, \ldots, n\}} \nu_i^D$$

Outline

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure

Linköping University

Numerical results

	Difference measure			Simple measure		
n-K-t	Succesf. runs	Av. gap	Time (s)	Succesf. runs	Av. gap	Time (s)
100-5-0.25	57 %	0.02 %	35	13 %	0.05 %	30
100-5-0.5	28 %	0.01 %	57	1 %	0.03 %	52
100-5-0.75	1 %	0.02 %	69	0 %	0.02 %	71
100-10-0.25	93 %	0.06 %	47	63 %	0.01 %	34
100-10-0.5	23 %	0.01 %	72	0 %	0.03 %	63
100-10-0.75	15 %	0.02 %	85	0 %	0.04 %	85
100-30-0.25	58 %	0.02 %	147	0 %	0.12 %	107
100-30-0.5	63 %	0.01 %	232	8 %	0.02 %	179
100-30-0.75	25 %	0.01 %	295	0 %	0.03 %	183
250-30-0.25	0 %	0.04 %	414	N/T	N/T	N/T
250-30-0.5	0 %	0.06 %	592	N/T	N/T	N/T
250-30-0.75	0 %	0.06 %	835	N/T	N/T	N/T

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K

Linköping University

Numerical results

	Difference measure			Simple measure		
n-K-t	Succesf. runs	Av. gap	Time (s)	Succesf. runs	Av. gap	Time (s)
100-5-0.25	57 %	0.02 %	35	13 %	0.05 %	30
100-5-0.5	28 %	0.01 %	57	1 %	0.03 %	52
100-5-0.75	1 %	0.02 %	69	0 %	0.02 %	71
100-10-0.25	93 %	0.06 %	47	63 %	0.01 %	34
100-10-0.5	23 %	0.01 %	72	0 %	0.03 %	63
100-10-0.75	15 %	0.02 %	85	0 %	0.04 %	85
100-30-0.25	58 %	0.02 %	147	0 %	0.12 %	107
100-30-0.5	63 %	0.01 %	232	8 %	0.02 %	179
100-30-0.75	25 %	0.01 %	295	0 %	0.03 %	183
250-30-0.25	0 %	0.04 %	414	N/T	N/T	N/T
250-30-0.5	0 %	0.06 %	592	N/T	N/T	N/T
250-30-0.75	0 %	0.06 %	835	N/T	N/T	N/T

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K

Linköping University

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Observations $n = 250$

versity

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Observations $n = 250$

- Still small average gaps ($\leq 0.06\%$)

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Observations $n = 250$

- Still small average gaps ($\leq 0.06\%$)
- None of instances "solved"

Numerical Tests

- $n \in \{100, 250\}$, $K \in \{5, 10, 30\}$, $t \in \{0.25, 0.5, 0.75\}$
- 36 instances, 50 runs per instance
- hard instances (CPLEX $> 2h$)

Observations $n = 100$

- Difference measure "outperforms" Simple measure
- Very small average gaps ($\sim 0.02\%$)
- Relative performance to CPLEX increases with K
- Average Time $<$ 1min ($<$ 1.5min, $<$ 5min)

Observations $n = 250$

- Still small average gaps ($\leq 0.06\%$)
- None of instances "solved"
- Ratio Running time/n increases slightly

Outline

Future work

Future work

- Improve utility measure for higher n

Future work

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

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**

Thank you!

:-)

Merci!