A fast combinatorial algorithm for the bilevel knapsack interdiction problem

Ricardo Fukasawa

ICERM - Linear and Non-Linear Mixed Integer Optimization
Feb 28, 2023
Joint work with Noah Weninger
Outline

1 Introduction

2 Our combinatorial algorithm
   - Heuristic
   - Branching
   - Bounding

3 Computational experiments

4 Conclusion
Bilevel optimization

Basic optimization problem:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad x \in U
\end{align*}
\]  

(OPT)
Bilevel optimization

Basic optimization problem:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad x \in U
\end{align*}
\] (OPT)

Key assumption:
- Decision-maker has total control of ALL decision variables.
Bilevel optimization

Basic optimization problem:

\[
\begin{align*}
\text{min} & \quad f(x) \\
\text{s.t.} & \quad x \in \mathcal{U}
\end{align*}
\]

(OPT)

Key assumption:
- Decision-maker has total control of ALL decision variables.
- Often not the case
  - Adversarial settings
  - Multiple (competing) decision makers
  - Not possible to coordinate efforts
Bilevel IP

Two decision-makers (DM):

\[
\begin{align*}
\min & \quad c^T x + d^T y \\
\text{s.t.} & \quad Ax \leq b \\
& \quad x \in \mathbb{Z}^n \\
& \quad y \in \arg\min\{f^T y : Gx + Hy \leq g, y \in \mathbb{Z}^p\}
\end{align*}
\]

- \(x\) are upper level decision variables
- \(y\) are lower level decision variables
- (DM) that controls \(x\) acts first, (DM) that controls \(y\) reacts
- Applications in military, economics, transportation, electrical grid, etc.
Knapsack Interdiction

- Interdiction idea: Upper level DM can choose to block some of the decisions from lower level DM
Knapsack Interdiction

- Interdiction idea: Upper level DM can choose to block some of the decisions from lower level DM
- Knapsack interdiction:
  - Given a set of items 1, \ldots, n
  - upper weight, lower weight, profit \((w^U_i, w^L_i, p_i)\)
  - Upper/lower knapsack capacities \(C^U, C^L\)

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} p_i y_i \\
\text{s.t.} & \quad \sum_{i=1}^{n} w^U_i x_i \leq C^U \\
& \quad x \in \{0, 1\}^n \\
\end{align*}
\]

\[
y \in \arg \max \left\{ \sum_{i=1}^{n} p_i y_i : \sum_{i=1}^{n} w^L_i y_i \leq C^L, x_i + y_i \leq 1, \forall i = 1, \ldots, n \right\}
\]

(BKP)
Knapsack Interdiction

- Interdiction idea: Upper level DM can choose to block some of the decisions from lower level DM

Knapsack interdiction:
- Given a set of items 1, \ldots, n
- upper weight, lower weight, profit \((w^U_i, w^L_i, p_i)\)
- Upper/lower knapsack capacities \(C^U, C^L\)

\[
\begin{align*}
\text{min} \quad & \sum_{i=1}^{n} p_i y_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} w^U_i x_i \leq C^U \\
& x \in \{0, 1\}^n \\
& y \in \text{arg max} \left\{ \sum_{i=1}^{n} p_i y_i : \sum_{i=1}^{n} w^L_i y_i \leq C^L, x_i + y_i \leq 1, \forall i = 1, \ldots, n \right\} \\
& y \in \{0, 1\}^n
\end{align*}
\]

(BKP)

The “Annoying Sibling Problem”
Knapsack Interdiction

- Interdiction idea: Upper level DM can choose to block some of the decisions from lower level DM

- Knapsack interdiction:
  - Given a set of items 1, . . . , n
  - upper weight, lower weight, profit \((w_i^U, w_i^L, p_i)\)
  - Upper/lower knapsack capacities \(C^U, C^L\)

\[
\begin{align*}
\text{min } & \sum_{i=1}^{n} p_i y_i \\
\text{s.t. } & \sum_{i=1}^{n} w_i^U x_i \leq C^U \\
& x \in \{0, 1\}^n
\end{align*}
\]

\[
y \in \arg \max \left\{ \sum_{i=1}^{n} p_i y_i : \sum_{i=1}^{n} w_i^L y_i \leq C^L, x_i + y_i \leq 1, \forall i = 1, \ldots, n, y \in \{0, 1\}^n \right\}
\]

(BKP)

The “Annoying Sibling Problem”

For each \(x \in \mathcal{U} := \left\{ x \in \{0, 1\}^n : \sum_{i=1}^{n} w_i^U x_i \leq C^U \right\} \), let \(\ell^*(x)\) be an optimizer of the lower level problem.
Knapsack interdiction

- $\Sigma^p_2$-complete problem (likely does not admit a poly-sized IP formulation) (Caprara et al., 2014)
- No pseudopolytime algorithm exists (unless $P = NP$)
- *DeNegre (2011) introduced the problem. Solved instances with $\leq 15$ items
- Caprara, Carvalho, Lodi and Woeginger (2016): Solved instances with $\leq 50$ items
- *Tang, Richard and Smith (2016): Solved instances with $\leq 30$ items
- *Fischetti, Ljubic, Monaci, and Sinnl (2019). Solved instances with $\leq 55$ items
- Lozano, Bergman and Cire (2022). Solved instances with $\leq 50$ items.
- Della Croce and Scatamacchia (2018). Solved instances with $\leq 500$ items
Knapsack interdiction

- $\Sigma_2^P$-complete problem (likely does not admit a poly-sized IP formulation) (Caprara et al., 2014)
- No pseudopolytime algorithm exists (unless $P = NP$)
- *DeNegre (2011) introduced the problem. Solved instances with $\leq 15$ items
- Caprara, Carvalho, Lodi and Woeginger (2016): Solved instances with $\leq 50$ items
- *Tang, Richard and Smith (2016): Solved instances with $\leq 30$ items
- *Fischetti, Ljubic, Monaci, and Sinnl (2019). Solved instances with $\leq 55$ items
- Lozano, Bergman and Cire (2022). Solved instances with $\leq 50$ items.
- Della Croce and Scatamacchia (2018). Solved instances with $\leq 500$ items

All approaches rely on MIP solvers
Outline

1 Introduction

2 Our combinatorial algorithm
   - Heuristic
   - Branching
   - Bounding

3 Computational experiments

4 Conclusion
Greedy Heuristic

Procedure GREEDY:

1. Obtain $x'$ by solving $\max \{ \sum_{i=1}^{n} p_{i}x_{i} : \sum_{i=1}^{n} w_{i}^{U}x_{i} \leq C^{U} \}$

3. $(x', y')$ is a feasible solution to our problem.

Notation:

$f(S) := \sum_{i \in S} f_{i}$

We will use interchangeably $X := \{ i : x_{i} = 1 \}$ and $x_{T} := \{ 1, \ldots, n \} \setminus T$
Procedure GREEDY:

1. Obtain \( x' \) by solving \( \max \{ \sum_{i=1}^{n} p_i x_i : \sum_{i=1}^{n} w_i U_i x_i \leq C^U \} \)

2. Obtain \( y' = \ell^*(x') \)
Greedy Heuristic

Procedure GREEDY:

1. Obtain $x'$ by solving $\max\{\sum_{i=1}^{n} p_i x_i : \sum_{i=1}^{n} w_i^U x_i \leq C^U\}$

2. Obtain $y' = \ell^*(x')$

3. $(x', y')$ is a feasible solution to our problem.
Greedy Heuristic

Procedure GREEDY:

1. Obtain $x'$ by solving \[ \max \left\{ \sum_{i=1}^{n} p_i x_i : \sum_{i=1}^{n} w_i^U x_i \leq C^U \right\} \]

2. Obtain $y' = \ell^*(x')$

3. $(x', y')$ is a feasible solution to our problem.

Notation:

- $f(S) := \sum_{i \in S} f_i$
- We will use interchangeably $X := \{i : x_i = 1\}$ and $x$
- $\overline{T} := \{1, \ldots, n\} \setminus T$
When is GREEDY optimal?

Della Croce and Scatamacchia (2018) and Caprara, Carvalho, Lodi and Woeginger (2016) note classes of easy instances.

**Lemma (Weninger and F. ’22)**

*GREEDY returns an optimal solution if there exists an optimal solution \((X^*, Y^*)\) for BKP where \(Y^* = \overline{X}^*\).*

**Proof.**

Suppose \(\hat{X} \in U\) with \(p(\hat{X}) > p(X^*)\). Let \(\hat{Y} = \ell^*(\hat{X})\)

\[
p(\hat{Y}) \leq p(\overline{\hat{X}}) < p(\overline{X}^*) = p(Y^*)
\]

which contradicts optimality of \((X^*, Y^*)\).

Moreover,

\[
p(Y') \leq p(\overline{X'}) = p(\overline{X}^*) = p(Y^*) \leq p(Y')
\]
When is GREEDY optimal?

Idea similar from Della Croce and Scatamacchia (2018).

\[
LB(c) = \min \sum_{i=1}^{c-1} p_i (1 - x_i) \\
\text{s.t.} \quad \sum_{i=1}^{c-1} w^U_i x_i \leq C^U \\
C^L - w^L_c + 1 \leq \sum_{i=1}^{c-1} w^L_i (1 - x_i) \leq C^L \\
x \in [0, 1]^n
\]

If the LP is infeasible for some \(c\), we define \(LB(c) = \infty\).

**Lemma (Weninger and F. ’22)**

*Suppose GREEDY returns \((X', Y')\) with value \(z'\). If \(z' \leq \min \{LB(c) : 1 \leq c \leq n\}\) then \((X', Y')\) is optimal for BKP.*

**Proof.**

Let \((X^*, Y^*)\) be an optimal solution to BKP. Let \(k = \min \{c : \sum_{i : i \leq c; x_i^* = 0} w^L_i > C^L\}\)

If \(k = \infty\), then by previous lemma, \((X', Y')\) is optimal for BKP.

Let \(x^* := (x_1^*, \ldots, x_{k-1}^*)\)

Then \(x^*\) is feasible for \(LB(k)\), so \(LB(k) \leq \sum_{i=1}^{k-1} p_i (1 - x^*_i) \leq p(y^*)\)

So \(z' \leq \min \{LB(c) : 1 \leq c \leq n\} \leq LB(k) \leq p(y^*) \leq z'\)
Outline

1 Introduction

2 Our combinatorial algorithm
   - Heuristic
   - Branching
   - Bounding

3 Computational experiments

4 Conclusion
Branching

Assume variables are ordered such that $\frac{p_1^1}{w_1^1} \geq \ldots \geq \frac{p_n^n}{w_n^n}$

- Branch at depth $i$ will be done on variable $x_{i+1}$
- Done in DFS, branching first on the right branch
- Check for infeasibility before branching
- At leaf (depth $n$), we get a possible $\hat{x}$: Use $(\hat{x}, \ell^*(\hat{x}))$ to update primal (upper) bound if needed

\[
\begin{array}{c}
\text{depth } i-1 \\
\end{array}
\]

\[
\begin{array}{c}
\text{depth } i-1 \\
\end{array}
\]

\[
\begin{array}{c}
\text{depth } i-1 \\
\end{array}
\]
Lower bounds

- BB node identified by $X \subseteq \{1, \ldots, i - 1\}$ that has been picked by upper level: Identified as $(X, i)$.
- Compute lower bound on node $(X, i)$ by:
  1. solving a lower-level knapsack on items \(\{1, \ldots, i - 1\} \setminus X\),
  2. computing a lower bound for BKP restricted to items $\{i, \ldots, n\}$, and
  3. combining (1) and (2) into a lower bound for the descendants of $(X, i)$.
Lower bounds

- BB node identified by $X \subseteq \{1, \ldots, i-1\}$ that has been picked by upper level: Identified as $(X, i)$.
- Compute lower bound on node $(X, i)$ by:
  1. solving a lower-level knapsack on items $\{1, \ldots, i-1\} \setminus X$,
  2. computing a lower bound for BKP restricted to items $\{i, \ldots, n\}$, and
  3. combining (1) and (2) into a lower bound for the descendants of $(X, i)$.

Let $K(X, c) := \max \{p(Y) : Y \subseteq \{1, \ldots, n\} \setminus X \text{ and } w^L(Y) \leq c\}$. 

Lower bounds

- BB node identified by $X \subseteq \{1, \ldots, i - 1\}$ that has been picked by upper level: Identified as $(X, i)$.
- Compute lower bound on node $(X, i)$ by:
  1. solving $K(X \cup \{i, \ldots, n\}, c)$,
  2. computing a lower bound for BKP restricted to items $\{i, \ldots, n\}$, and
  3. combining (1) and (2) into a lower bound for the descendants of $(X, i)$.

Let $K(X, c) := \max \{p(Y) : Y \subseteq \{1, \ldots, n\} \setminus X \text{ and } w^L(Y) \leq c\}$. 
Lower bounds

- BB node identified by $X \subseteq \{1, \ldots, i - 1\}$ that has been picked by upper level:
  Identified as $(X, i)$.
- Compute lower bound on node $(X, i)$ by:
  1. solving $K(X \cup \{i, \ldots, n\}, c)$,
  2. computing a lower bound for BKP restricted to items $\{i, \ldots, n\}$, and
  3. combining (1) and (2) into a lower bound for the descendants of $(X, i)$.

Let $K(X, c) := \max \{ p(Y) : Y \subseteq \{1, \ldots, n\} \setminus X \text{ and } w^L(Y) \leq c \}.$

Let $\omega(i, c^U, c^L) \leq \min \{ K(X' \cup \{1, \ldots, i - 1\}, c^L) : X' \subseteq \{i, \ldots, n\}, w^U(X') \leq c^U \}. $
Lower bounds

- BB node identified by $X \subseteq \{1, \ldots, i-1\}$ that has been picked by upper level: Identified as $(X, i)$.

- Compute lower bound on node $(X, i)$ by:
  1. solving $K(X \cup \{i, \ldots, n\}, c)$,
  2. computing $\omega(i, c^U, c^L)$, and
  3. combining (1) and (2) into a lower bound for the descendants of $(X, i)$.

Let $K(X, c) := \max \{ p(Y) : Y \subseteq \{1, \ldots, n\} \setminus X \text{ and } w^L(Y) \leq c \}$.

Let $\omega(i, c^U, c^L) \leq \min \{ K(X' \cup \{1, \ldots, i-1\}, c^L) : X' \subseteq \{i, \ldots, n\}, w^U(X') \leq c^U \}$. 
Combining (1) and (2)

Lemma

Let \((X, i)\) be a subproblem. For all \(c \in \{0, \ldots, C^L\}\),

\[
K(X \cup \{i, \ldots, n\}, c) + \omega\left(i, C^U - w^U(X), C^L - c\right) \leq \\
\min \left\{ p(Y') : (X', Y') \text{ is feasible for BKP and } X' \cap \{1, \ldots, i - 1\} = X \right\}.
\]
Combining (1) and (2)

Lemma

Let \((X, i)\) be a subproblem. For all \(c \in \{0, \ldots, C^L\}\),

\[
K(X \cup \{i, \ldots, n\}, c) + \omega\left(i, C^U - w^U(X), C^L - c\right) \leq \\
\min \{p(Y') : (X', Y') \text{ is feasible for BKP and } X' \cap \{1, \ldots, i - 1\} = X\}.
\]

Proof:

Fix \(X \subseteq \{1, \ldots, i - 1\}\) and \(X' \subseteq \{i, \ldots, n\}\)

\[
\{1, \ldots, i - 1\} \quad \{i, \ldots, n\}
\]

Then

\[
K(X \cup \{i, \ldots, n\}, c) + K(X' \cup \{1, \ldots, i\}, C^L - c) \leq K(X \cup X', C^L)
\]
Combining (1) and (2)

**Lemma**

Let \((X, i)\) be a subproblem. For all \(c \in \{0, \ldots, C^L\}\),

\[
K(X \cup \{i, \ldots, n\}, c) + \omega\left(i, C^U - w^U(X), C^L - c\right) \leq \\
\min \left\{p(Y') : (X', Y') \text{ is feasible for BKP and } X' \cap \{1, \ldots, i - 1\} = X\right\}.
\]

**Proof:**

Fix \(X \subseteq \{1, \ldots, i - 1\} \) and \(X' \subseteq \{i, \ldots, n\}\)

\[
K(X \cup \{i, \ldots, n\}, c) + K(X' \cup \{1, \ldots, i\}, C^L - c) \leq K(X \cup X', C^L)
\]

So:

\[
K(X \cup \{i, \ldots, n\}, c) + \min\{K(X' \cup \{1, \ldots, i\}, C^L - c) : X' \subseteq \{i, \ldots, n\}\} \leq \\
\leq \min\{K(X \cup X', C^L) : X' \subseteq \{i, \ldots, n\}\}
\]
Obtaining $\omega$

Recall: $\omega(i, c^U, c^L) \leq \min\{K(X' \cup \{1, \ldots, i - 1\}, c^L) : X' \subseteq \{i, \ldots, n\}, w^U(X') \leq c^U\}$
Obtaining $\omega$

Recall: $\omega(i, c^U, c^L) \leq \min\{K(X' \cup \{1, \ldots, i - 1\}, c^L) : X' \subseteq \{i, \ldots, n\}, w^U(X') \leq c^U\}

Idea: Apply greedy policy to both children nodes, and pick the lowest.
Obtaining $\omega$

Recall: $\omega(i, c^U, c^L) \leq \min\{K(X' \cup \{1, \ldots, i - 1\}, c^L) : X' \subseteq \{i, \ldots, n\}, w^U(X') \leq c^U\}$

Idea: Apply greedy policy to both children nodes, and pick the lowest.

$$
\omega_g(i, c^U, c^L) = \begin{cases} 
\infty & \text{if } c^U < 0, \\
0 & \text{if } c^U \geq 0, c^L \geq 0 \text{ and } i > n, \\
\omega_g(i + 1, c^U, c^L) & \text{if } c^U \geq 0, w_i^L > c^L \text{ and } i \leq n, \\
\min \left\{ \omega_g(i + 1, c^U - w_i^U, c^L), \omega_g(i + 1, c^U, c^L - w_i^L) + p_i \right\} & \text{if } c^U \geq 0, w_i^L \leq c^L \text{ and } i \leq n.
\end{cases}
$$
Improving $\omega_g$

Greedy: Follower always takes item $i$ if possible.
Improving $\omega_g$

Greedy: Follower always takes item $i$ if possible.

Improvement: Allow follower a different choice

$$
\omega(i, c^U, c^L) = \begin{cases}
+\infty, \\
\ldots \\
\min \left\{ \begin{array}{l}
\omega(i + 1, c^U - w_i^U, c^L) \\
\max \left\{ \omega(i + 1, c^U, c^L), \\
\omega(i + 1, c^U, c^L - w_i^L) + p_i \right\} \end{array} \right. \\
\right. 
\right. 
\right. 
$$

if $c^U < 0$

$\omega(i, c^U, c^L) = \begin{cases}
+\infty, \\
\ldots \\
ymin \left\{ \begin{array}{l}
\omega(i + 1, c^U - w_i^U, c^L) \\
\max \left\{ \omega(i + 1, c^U, c^L), \\
\omega(i + 1, c^U, c^L - w_i^L) + p_i \right\} \end{array} \right. \\
\right. 
\right. 
\right. 
$$

if $\ldots$

$x_i = 0$

$x_i = 1$
Validity of lower bound

**Theorem**

For all $1 \leq i \leq n$, $c^U \geq 0$ and $c^L \geq 0$,

$$
\omega(i, c^U, c^L) \leq \min_{X' \subseteq \{i, \ldots, n\} : w^U(X') \leq c^U} K(X \cup \{1, \ldots, i - 1\}, c^L).
$$

Another interpretation of what is happening:

Leader and follower are taking turns:

After leader chooses $x_i = 0$ or $1$, follower reacts.

Becomes a $2n$-stage game (where each stage is very simple).
Validity of lower bound

**Theorem**

For all $1 \leq i \leq n$, $c^U \geq 0$ and $c^L \geq 0$,

$$\omega(i, c^U, c^L) \leq \min_{X' \subseteq \{i, \ldots, n\} : w(U(X')) \leq c^U} K(X \cup \{1, \ldots, i - 1\}, c^L).$$

Another interpretation of what is happening:

- Leader and follower are taking turns:
  - After leader chooses $x_i = 0$ or $1$, follower reacts

Becomes a $2n$-stage game (where each stage is very simple).
How to use good solutions for the follower?

Consider the problem

\[ z^* := \min_{x \in U} \max_{y \in \mathcal{L}(x)} c(x, y) \quad (1) \]

For each \( x \in U \), let \( \ell^*(x) \) be an optimizer of \( c(x, y) \) for \( y \in \mathcal{L}(x) \) (we assume it exists).

**Lemma (Weninger and F., 22)**

Suppose we have a function \( f(x) \) such that for all \( x \in U \):

- \( f(x) \in \mathcal{L}(x) \)
- \( c(x, f(x)) \leq c(x, \ell^*(x)) \leq \alpha c(x, f(x)), \text{ for some } \alpha \geq 1 \)

Let \( \tilde{x} \in \arg \min c(x, f(x)) \).

Then

\[ c(\tilde{x}, \ell^*(\tilde{x})) \leq \alpha z^* \]
How to use good solutions for the follower?

Consider the problem

\[ z^* := \min_{x \in U} \max_{y \in L(x)} c(x, y) \] (1)

For each \( x \in U \), let \( \ell^*(x) \) be an optimizer of \( c(x, y) \) for \( y \in L(x) \) (we assume it exists).

**Lemma (Weninger and F., 22)**

Suppose we have a function \( f(x) \) such that for all \( x \in U \):

1. \( f(x) \in L(x) \)
2. \( c(x, f(x)) \leq c(x, \ell^*(x)) \leq \alpha c(x, f(x)), \text{ for some } \alpha \geq 1 \)

Let \( \tilde{x} \in \arg \min c(x, f(x)) \).

Then

\[ c(\tilde{x}, \ell^*(\tilde{x})) \leq \alpha z^* \]

**Proof:**

Let \((x^*, \ell^*(x^*))\) be the optimal solution to (1).

\[ \frac{1}{\alpha} c(\tilde{x}, \ell^*(\tilde{x})) \leq c(\tilde{x}, f(\tilde{x})) \leq c(x^*, f(x^*)) \leq c(x^*, \ell^*(x^*)) = z^* \]
Instances from the literature

Instances:

- **CCLW**: From Caprara et al (2016). \( n \in [35, 55] \)
- **DCS**: From DellaCroce and Scatamacchia (2020). \( n \in [100, 500] \)
- **DeNegre**: From DeNegre (2011). \( n \in [10, 50] \)
- **FMS**: From Fischetti et al (2018). \( n \in [100, 500] \)
- **TRS**: Tang et al (2016). \( n \in [15, 30] \)

Algorithms:

- **DCS**: Previous best from literature
- **Comb**: Ours

<table>
<thead>
<tr>
<th>Group</th>
<th>#Inst</th>
<th>#Opt</th>
<th>#Best</th>
<th>DCS Avg</th>
<th>DCS Max</th>
<th>#Opt</th>
<th>#Best</th>
<th>Comb Avg</th>
<th>Comb Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>1,340</td>
<td>1,271</td>
<td>55</td>
<td>200.49</td>
<td>3,600</td>
<td>1,324</td>
<td>1,269</td>
<td>44.11</td>
<td>3,600</td>
</tr>
<tr>
<td>CCLW</td>
<td>50</td>
<td>50</td>
<td>2</td>
<td>0.21</td>
<td>1.27</td>
<td>50</td>
<td>48</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>DCS</td>
<td>500</td>
<td>500</td>
<td>0</td>
<td>3.87</td>
<td>15.73</td>
<td>500</td>
<td>500</td>
<td>0.7</td>
<td>8.59</td>
</tr>
<tr>
<td>DeNegre</td>
<td>160</td>
<td>160</td>
<td>50</td>
<td>0.17</td>
<td>1.62</td>
<td>160</td>
<td>110</td>
<td>0.08</td>
<td>1.73</td>
</tr>
<tr>
<td>FMS-easy</td>
<td>150</td>
<td>150</td>
<td>0</td>
<td>13.35</td>
<td>79.85</td>
<td>150</td>
<td>150</td>
<td>0.38</td>
<td>7.1</td>
</tr>
<tr>
<td>FMS-hard</td>
<td>300</td>
<td>231</td>
<td>0</td>
<td>882.2</td>
<td>3,600</td>
<td>284</td>
<td>284</td>
<td>195.61</td>
<td>3,600</td>
</tr>
<tr>
<td>TRS</td>
<td>180</td>
<td>180</td>
<td>3</td>
<td>0.14</td>
<td>2.04</td>
<td>180</td>
<td>177</td>
<td>0.04</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**Table**: Summary of results for all instances from the literature.
Results per instance type

<table>
<thead>
<tr>
<th>Group</th>
<th>#Inst</th>
<th>DCS</th>
<th>Avg</th>
<th>Max</th>
<th>Comb</th>
<th>Root</th>
<th>Nodes</th>
<th>Root gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#Opt</td>
<td>#Best</td>
<td>Avg</td>
<td>Max</td>
<td>Avg</td>
<td>Max</td>
<td>Root</td>
</tr>
<tr>
<td>uncorrelated</td>
<td>940</td>
<td>940</td>
<td>64</td>
<td>2.32</td>
<td>15.73</td>
<td>0.31</td>
<td>6.74</td>
<td>0.3</td>
</tr>
<tr>
<td>weak correlated</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>13.49</td>
<td>72.64</td>
<td>0.26</td>
<td>3.59</td>
<td>0.25</td>
</tr>
<tr>
<td>strong correlated</td>
<td>50</td>
<td>41</td>
<td>0</td>
<td>689.58</td>
<td>3,600</td>
<td>0.34</td>
<td>3.89</td>
<td>0.26</td>
</tr>
<tr>
<td>inverse strong corr.</td>
<td>50</td>
<td>38</td>
<td>0</td>
<td>919.91</td>
<td>3,600</td>
<td>1.06</td>
<td>34.24</td>
<td>0.38</td>
</tr>
<tr>
<td>almost strong corr.</td>
<td>50</td>
<td>40</td>
<td>0</td>
<td>815.4</td>
<td>3,600</td>
<td>0.24</td>
<td>3.17</td>
<td>0.24</td>
</tr>
<tr>
<td>subset-sum</td>
<td>50</td>
<td>35</td>
<td>0</td>
<td>1,087.18</td>
<td>3,600</td>
<td>42</td>
<td>586.33</td>
<td>0.29</td>
</tr>
<tr>
<td>even-odd subset-sum</td>
<td>50</td>
<td>36</td>
<td>0</td>
<td>1,033.98</td>
<td>3,600</td>
<td>42</td>
<td>581.43</td>
<td>0.29</td>
</tr>
<tr>
<td>even-odd strong corr.</td>
<td>50</td>
<td>41</td>
<td>0</td>
<td>747.12</td>
<td>3,600</td>
<td>50</td>
<td>0.6</td>
<td>16.86</td>
</tr>
<tr>
<td>similar weight uncorr.</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>22.89</td>
<td>79.85</td>
<td>50</td>
<td>4.64 \cdot 10^{-2}</td>
<td>7.92 \cdot 10^{-2}</td>
</tr>
</tbody>
</table>

**Table:** Results on all instances, by instance type
Variations of the algorithm

<table>
<thead>
<tr>
<th></th>
<th>Solution to ( K(X \cup {i, \ldots, n}, c) )</th>
<th>Lower bound on ( K(X' \cup {1, \ldots, i - 1}, c^L) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comb</td>
<td>Exact</td>
<td>( \omega )</td>
</tr>
<tr>
<td>Comb-weak</td>
<td>Exact</td>
<td>( \omega_g )</td>
</tr>
<tr>
<td>Comb-greedy</td>
<td>Greedy</td>
<td>( \omega )</td>
</tr>
</tbody>
</table>

Figure: Performance profile for all instances from the literature.
Influence of threads

Figure: Performance profile for all instances from the literature, different numbers of threads.

(DCS ran with 16 threads)
New instances

\( n \in [10, 10000] \), both easy and hard, also allowing correlation with \( w_i^U \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>#Inst</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>250</td>
<td>250</td>
<td>101</td>
<td>0.13</td>
<td>3.41</td>
<td>250</td>
<td>149</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>238</td>
<td>8</td>
<td>58.36</td>
<td>900</td>
<td>250</td>
<td>242</td>
<td>0.05</td>
<td>0.33</td>
</tr>
<tr>
<td>50</td>
<td>250</td>
<td>203</td>
<td>1</td>
<td>178.63</td>
<td>900</td>
<td>247</td>
<td>246</td>
<td>17.83</td>
<td>900</td>
</tr>
<tr>
<td>100</td>
<td>250</td>
<td>184</td>
<td>3</td>
<td>253.42</td>
<td>900</td>
<td>222</td>
<td>219</td>
<td>104.77</td>
<td>900</td>
</tr>
<tr>
<td>1000</td>
<td>167</td>
<td>109</td>
<td>12</td>
<td>302.26</td>
<td>900</td>
<td>136</td>
<td>124</td>
<td>169.82</td>
<td>900</td>
</tr>
<tr>
<td>10000</td>
<td>26</td>
<td>23</td>
<td>0</td>
<td>357.43</td>
<td>900</td>
<td>26</td>
<td>26</td>
<td>12.55</td>
<td>25.21</td>
</tr>
</tbody>
</table>

Class

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>uncorrelated</td>
<td>241</td>
<td>239</td>
<td>25</td>
<td>12.37</td>
<td>900</td>
<td>241</td>
<td>216</td>
<td>0.97</td>
<td>25.21</td>
</tr>
<tr>
<td>lower subset-sum</td>
<td>256</td>
<td>174</td>
<td>13</td>
<td>318.09</td>
<td>900</td>
<td>237</td>
<td>224</td>
<td>70.2</td>
<td>900</td>
</tr>
<tr>
<td>upper subset-sum</td>
<td>232</td>
<td>232</td>
<td>31</td>
<td>2.58</td>
<td>89.25</td>
<td>232</td>
<td>201</td>
<td>0.8</td>
<td>18.29</td>
</tr>
<tr>
<td>both subset-sum</td>
<td>232</td>
<td>130</td>
<td>23</td>
<td>417.68</td>
<td>900</td>
<td>189</td>
<td>166</td>
<td>175.93</td>
<td>900</td>
</tr>
<tr>
<td>equal weights</td>
<td>232</td>
<td>232</td>
<td>33</td>
<td>2.12</td>
<td>120.55</td>
<td>232</td>
<td>199</td>
<td>0.67</td>
<td>14.97</td>
</tr>
</tbody>
</table>

**Table:** Summary of results for new instances, grouped by \( n \) (upper half) and by class (lower half)
New instances

\( n \in [10, 10000] \), both easy and hard, also allowing correlation with \( w_i^U \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>#Inst</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>250</td>
<td>250</td>
<td>101</td>
<td>0.13</td>
<td>3.41</td>
<td>250</td>
<td>149</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>238</td>
<td>8</td>
<td>58.36</td>
<td>900</td>
<td>250</td>
<td>242</td>
<td>0.05</td>
<td>0.33</td>
</tr>
<tr>
<td>50</td>
<td>250</td>
<td>203</td>
<td>1</td>
<td>178.63</td>
<td>900</td>
<td>247</td>
<td>246</td>
<td>17.83</td>
<td>900</td>
</tr>
<tr>
<td>100</td>
<td>250</td>
<td>184</td>
<td>3</td>
<td>253.42</td>
<td>900</td>
<td>222</td>
<td>219</td>
<td>104.77</td>
<td>900</td>
</tr>
<tr>
<td>1000</td>
<td>167</td>
<td>109</td>
<td>12</td>
<td>302.26</td>
<td>900</td>
<td>136</td>
<td>124</td>
<td>169.82</td>
<td>900</td>
</tr>
<tr>
<td>10000</td>
<td>26</td>
<td>23</td>
<td>0</td>
<td>357.43</td>
<td>900</td>
<td>26</td>
<td>26</td>
<td>12.55</td>
<td>25.21</td>
</tr>
</tbody>
</table>

Class

<table>
<thead>
<tr>
<th>( n )</th>
<th>#Inst</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
<th>#Opt</th>
<th>#Best</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>uncorrelated</td>
<td>241</td>
<td>239</td>
<td>25</td>
<td>12.37</td>
<td>900</td>
<td>241</td>
<td>216</td>
<td>0.97</td>
<td>25.21</td>
</tr>
<tr>
<td>lower subset-sum</td>
<td>256</td>
<td>174</td>
<td>13</td>
<td>318.09</td>
<td>900</td>
<td>237</td>
<td>224</td>
<td>70.2</td>
<td>900</td>
</tr>
<tr>
<td>upper subset-sum</td>
<td>232</td>
<td>232</td>
<td>31</td>
<td>2.58</td>
<td>89.25</td>
<td>232</td>
<td>201</td>
<td>0.8</td>
<td>18.29</td>
</tr>
<tr>
<td>both subset-sum</td>
<td>232</td>
<td>130</td>
<td>23</td>
<td>417.68</td>
<td>900</td>
<td>189</td>
<td>166</td>
<td>175.93</td>
<td>900</td>
</tr>
<tr>
<td>equal weights</td>
<td>232</td>
<td>232</td>
<td>33</td>
<td>2.12</td>
<td>120.55</td>
<td>232</td>
<td>199</td>
<td>0.67</td>
<td>14.97</td>
</tr>
</tbody>
</table>

Table: Summary of results for new instances, grouped by \( n \) (upper half) and by class (lower half)

NOTE: Some instances we ran out of memory (excluded from this table)
Figure: Left: scatter plot of the lower bound approximation ratio $\omega(1, C^U, C^L)/\text{OPT}$ for all instances (sorted by approximation ratio). Right: running time of Comb (darker = longer time) as a function of $C^U$ and $C^L$, for an uncorrelated instance with 100 items.
1. Introduction

2. Our combinatorial algorithm
   - Heuristic
   - Branching
   - Bounding

3. Computational experiments

4. Conclusion
Conclusion

Key ideas:
- Exploit strong lower bounds via good heuristics to lower level
- Idea of using $2n$ rounds of a simpler game
- Lower bounds that are extendable from each other (DP)
- Synchronization with how we branch

Challenges / future research:
- Memory intensive
- Parallelization
- Extension to other problems

Code and new instances available at
https://github.com/nwoeanhinnogaehr/bkpsolver
Conclusion

Key ideas:
- Exploit strong lower bounds via good heuristics to lower level
- Idea of using $2n$ rounds of a simpler game
- Lower bounds that are extendable from each other (DP)
- Synchronization with how we branch

Challenges / future research:
- Memory intensive
- Parallelization
- Extension to other problems

Code and new instances available at https://github.com/nwoeanhinnogaehr/bkpsolver

THANK YOU!