Computational testing of exact separation for mixed-integer knapsack problems

Pasquale Avella*

(joint work with Maurizio Boccia* and Igor Vasiliev**)

* *DING - Università del Sannio*

** *Russian Academy of Sciences - Siberian Branch*

MIP 2008 - Columbia University

MIP solvers & cutting planes

MIP solvers include cut generation routines looking at single-row relaxations:

MIP solvers & cutting planes

MIP solvers include cut generation routines looking at single-row relaxations:

- Knapsack $\Rightarrow$ Lifted Cover Inequalities

MIP solvers & cutting planes

MIP solvers include cut generation routines looking at single-row relaxations:

- Knapsack $\Rightarrow$ Lifted Cover Inequalities
- Mixed knapsack $\Rightarrow$ Mixed-Integer Rounding (MIR) inequalities

MIP solvers & cutting planes

MIP solvers include cut generation routines looking at single-row relaxations:

- Knapsack $\Rightarrow$ Lifted Cover Inequalities
- Mixed knapsack $\Rightarrow$ Mixed-Integer Rounding (MIR) inequalities
- Tableau rows $\Rightarrow$ Gomory Cuts

“Cuts outside the template paradigm”

Can we do anything more to tighten MIP formulations by looking at single-row relaxations?

“Cuts outside the template paradigm”

Can we do anything more to tighten MIP formulations by looking at single-row relaxations?

- We can try to generate “cuts outside the template paradigm” (local cuts: Applegate, Bixby, Chvátal and Cook, 2000)

"Cuts outside the template paradigm"

Can we do anything more to tighten MIP formulations by looking at single-row relaxations?

- We can try to generate "cuts outside the template paradigm" (local cuts: Applegate, Bixby, Chvátal and Cook, 2000)
- Local cuts proved to be successful for the TSP

"Cuts outside the template paradigm"

Can we do anything more to tighten MIP formulations by looking at single-row relaxations?

- We can try to generate "cuts outside the template paradigm" (local cuts: Applegate, Bixby, Chvátal and Cook, 2000)
- Local cuts proved to be successful for the TSP
- Based on exact separation.

Exact separation

- Given: a polyhedron $P \subset \mathbb{R}^n$ and a point $\bar{x} \in \mathbb{R}^n$.

Exact separation

- Given: a polyhedron $P \subset \mathbb{R}^n$ and a point $\bar{x} \in \mathbb{R}^n$.
- A separation algorithm is said exact if it either guarantees to provide a valid inequality for P cutting off $\bar{x}$ or concludes that $\bar{x} \in P$.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}_+^n : \ \boldsymbol{ay} \le b, \ \boldsymbol{y} \le \boldsymbol{u}\}$$

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}^n_+ : \ \boldsymbol{ay} \le b, \ \boldsymbol{y} \le \boldsymbol{u}\}$$

The exact separation LP *SEPLP*(X^K):

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 & \\
& \boldsymbol{w\pi} \le \pi_0, \quad \boldsymbol{w} \in X^K & (1)\\
& \boldsymbol{1\pi} = 1 & (2)\\
& \boldsymbol{\pi}, \pi_0 \ge 0 &
\end{aligned}$$

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}_+^n : \ \ \boldsymbol{ay} \le b, \ \boldsymbol{y} \le \boldsymbol{u}\}$$

The exact separation LP $SEPLP(X^K)$:

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 \\ & \boldsymbol{w\pi} \le \pi_0, \quad \boldsymbol{w} \in X^K && (1) \\ & \mathbf{1}\boldsymbol{\pi} = 1 && (2) \\ & \boldsymbol{\pi}, \pi_0 \ge 0 \end{aligned}$$

$\bar{\boldsymbol{y}} \in \mathbb{R}^n$ is the fractional point to cut-off.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}_+^n : \;\; \boldsymbol{ay} \leq b, \; \boldsymbol{y} \leq \boldsymbol{u}\}$$

The exact separation LP *SEPLP*(X^K):

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 & \\ & \boldsymbol{w}\boldsymbol{\pi} \leq \pi_0, \quad \boldsymbol{w} \in X^K & (1) \\ & \mathbf{1}\boldsymbol{\pi} = 1 & (2) \\ & \boldsymbol{\pi}, \pi_0 \geq 0 & \end{aligned}$$

Inequalities (1) ensure that the inequality is satisfied from every feasible solution in X^K.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}_+^n : \ \ \boldsymbol{ay} \le b, \ \boldsymbol{y} \le \boldsymbol{u}\}$$

The exact separation LP *SEPLP*(X^K):

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 & \\ & \boldsymbol{w\pi} \le \pi_0, \quad \boldsymbol{w} \in X^K & (1) \\ & \boldsymbol{1\pi} = 1 & (2) \\ & \boldsymbol{\pi}, \pi_0 \ge 0 & \end{aligned}$$

(2) is a normalization constraint.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}_+^n : \ \boldsymbol{a}\boldsymbol{y} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

The exact separation LP $SEPLP(X^K)$:

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 \\ & \boldsymbol{w}\boldsymbol{\pi} \leq \pi_0, \quad \boldsymbol{w} \in X^K & (1) \\ & \mathbf{1}\boldsymbol{\pi} = 1 & (2) \\ & \boldsymbol{\pi}, \pi_0 \geq 0 \end{aligned}$$

Let $\boldsymbol{\pi}^*, \pi_0^*$ be the optimal solution of $SEPLP(X^K)$.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\boldsymbol{y} \in \mathbb{Z}^n_+ : \ \ \boldsymbol{ay} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

The exact separation LP *SEPLP*(X^K):

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \pi_0 \\ & \boldsymbol{w\pi} \leq \pi_0, \quad \boldsymbol{w} \in X^K && (1) \\ & \boldsymbol{1\pi} = 1 && (2) \\ & \boldsymbol{\pi}, \pi_0 \geq 0 \end{aligned}$$

The inequality $\boldsymbol{\pi}^* y \leq \pi_0^*$ is valid for $conv(X^K)$.

Exact separation of valid inequalities for the knaspack polytope

The knapsack set (Boyd, 1988)

$$X^K = \{\mathbf{y} \in \mathbb{Z}_+^n : \quad \mathbf{a}\mathbf{y} \le b, \ \mathbf{y} \le \mathbf{u}\}$$

The exact separation LP *SEPLP*(X^K):

$$\begin{aligned}
\max \quad & \bar{\mathbf{y}}\boldsymbol{\pi} - \pi_0 \\
& \mathbf{w}\boldsymbol{\pi} \le \pi_0, \quad \mathbf{w} \in X^K && (1) \\
& \mathbf{1}\boldsymbol{\pi} = 1 && (2) \\
& \boldsymbol{\pi}, \pi_0 \ge 0
\end{aligned}$$

Extreme points of *SEPLP*(X^K) are in one-to-one correspondence with the facets of *conv*(X^K).

Recent results

Extension of the "local cuts" technique to MIP problems

- Espinoza (2006)

Recent results

Extension of the “local cuts” technique to MIP problems

- Espinoza (2006)

MIPLIB instances

- Kaparis and Letchford (2007) yielded tighter lower bounds for several MIPLIB instances

Recent results

Extension of the "local cuts" technique to MIP problems

- Espinoza (2006)

MIPLIB instances

- Kaparis and Letchford (2007) yielded tighter lower bounds for several MIPLIB instances

Generalized Assignment problem

- Medium-size Generalized Assignment instances *d*10200 and *d*20200 solved to optimality for the first time.
- Integrality gap reduced on many larger benchmark instances (up to 80*x*1600) (A., Boccia and Vasilyev, 2007).

Recent results (cont.)

Single Source Capacitated Facility Location Problems

- Reformulation based on dicut inequalities + exact separation (Boccia, 2007).
- Many benchmark instances solved to optimality (MIP solvers failed).

Recent results (cont.)

Single Source Capacitated Facility Location Problems

- Reformulation based on dicut inequalities + exact separation (Boccia, 2007).
- Many benchmark instances solved to optimality (MIP solvers failed).

Set Covering

- Exact separation for subsets of formulation constraints (A., Boccia and Vasyliev, 2007).
- *seymour* solved to optimality on a single workstation.

A step further: the mixed-integer knapsack set X^{MI}

We consider single-row mixed-integer knapsack relaxations of MIP problems:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}$$

A step further: the mixed-integer knapsack set X^{MI}

We consider single-row mixed-integer knapsack relaxations of MIP problems:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \quad \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}$$

- Atamturk (2002) studied the polyhedral structure of $conv(X^{MI})$.

A step further: the mixed-integer knapsack set X^{MI}

We consider single-row mixed-integer knapsack relaxations of MIP problems:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \quad \boldsymbol{ay} + \boldsymbol{gx} \leq b, \; \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}$$

- Atamturk (2002) studied the polyhedral structure of $conv(X^{MI})$.
- Fukasawa and Goycoolea (2007) proposed an exact separation routine for X^{MI}. The core of their separation procedure is a sophisticated Branch-and-Bound algorithm for the mixed-integer knapsack problem.

The knapsack set with a single continuous variable X^{MK}

If in

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : \ \ \boldsymbol{ay} + \boldsymbol{gx} \le b, \ \boldsymbol{y} \le \boldsymbol{u}, \boldsymbol{x} \le \boldsymbol{v}\}$$

we remove bounds $\boldsymbol{v}$ and aggregate the continuous variables we get the "weaker" knapsack set with a single continuous variable X^{MK}:

$$X^{MK} = \{(\boldsymbol{y}, s) \in \mathbb{Z}^n_+ \times \mathbb{R}_+ : \ \ \boldsymbol{ay} - s \le b, \ \boldsymbol{y} \le \boldsymbol{u}\}$$

The knapsack set with a single continuous variable X^{MK}

If in

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : \quad \boldsymbol{ay} + \boldsymbol{gx} \leq b, \; \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}$$

we remove bounds $\boldsymbol{v}$ and aggregate the continuous variables we get the "weaker" knapsack set with a single continuous variable X^{MK}:

$$X^{MK} = \{(\boldsymbol{y}, s) \in \mathbb{Z}^n_+ \times \mathbb{R}_+ : \quad \boldsymbol{ay} - s \leq b, \; \boldsymbol{y} \leq \boldsymbol{u}\}$$

Why we focus on X^{MK}

The set X^{MK} is a better candidate for a "lightweight" exact separation routine.

A few remarks on $conv(X^{MK})$

- The polyhedron $conv(X^{MK})$ was investigated by Marchand and Wolsey (1999)

A few remarks on $conv(X^{MK})$

- The polyhedron $conv(X^{MK})$ was investigated by Marchand and Wolsey (1999)
- They showed that Mixed-Integer Rounding (MIR) inequalities

$$\sum_{j=1}^{n} \left(\lfloor a_j \rfloor + \frac{(f_{a_j} - f_b)^+}{1 - f_b} \right) x_j \leq \lfloor b \rfloor + \frac{s}{1 - f_b}$$

(where $f_d = d - \lfloor d \rfloor$) can be easily derived from X^{MK}.

A few remarks on $conv(X^{MK})$

- The polyhedron $conv(X^{MK})$ was investigated by Marchand and Wolsey (1999)
- They showed that Mixed-Integer Rounding (MIR) inequalities

$$\sum_{j=1}^{n} \left(\lfloor a_j \rfloor + \frac{(f_{a_j} - f_b)^+}{1 - f_b} \right) x_j \leq \lfloor b \rfloor + \frac{s}{1 - f_b}$$

(where $f_d = d - \lfloor d \rfloor$) can be easily derived from X^{MK}.

- They characterized several other classes of valid inequalities for $conv(X^{MK})$

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi} \boldsymbol{y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi} \boldsymbol{y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 & \\
& \boldsymbol{w}\boldsymbol{\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} & (3) \\
& \mathbf{1}\boldsymbol{\pi} + \sigma = 1 & (4) \\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0 &
\end{aligned}$$

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 \\
& \boldsymbol{w\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} && (3) \\
& \boldsymbol{1\pi} + \sigma = 1 && (4) \\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0
\end{aligned}$$

$(\bar{\boldsymbol{y}}, \bar{\boldsymbol{s}}) \in \mathbb{R}^n$ is the fractional point to cut-off.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$
\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 \\
& \boldsymbol{w\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} & (3) \\
& \mathbf{1}\boldsymbol{\pi} + \sigma = 1 & (4) \\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0
\end{aligned}
$$

Inequalities (3) ensure that the inequality is satisfied from every feasible solution in X^{MK}.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 & \\
& \boldsymbol{w\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} & (3) \\
& \boldsymbol{1\pi} + \sigma = 1 & (4) \\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0 &
\end{aligned}$$

(4) is a normalization constraint.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi}\boldsymbol{y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 \\
& \boldsymbol{w}\boldsymbol{\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} && (3) \\
& \boldsymbol{1}\boldsymbol{\pi} + \sigma = 1 && (4) \\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0
\end{aligned}$$

Let $\boldsymbol{\pi}^*, \sigma^*, \pi_0^*$ be the optimal solution of $SEPLP(X^{MK})$.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi}\boldsymbol{y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned} \max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 \\ & \boldsymbol{w}\boldsymbol{\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} && (3) \\ & \boldsymbol{1}\boldsymbol{\pi} + \sigma = 1 && (4) \\ & \boldsymbol{\pi}, \sigma, \pi_0 \geq 0 \end{aligned}$$

The inequality $\boldsymbol{\pi}^*\boldsymbol{y} - \sigma^* \boldsymbol{s} \leq \pi_0^*$ is valid for $conv(X^{MK})$.

Exact separation for $conv(X^{MK})$

Any valid inequality for $conv(X^{MK})$ has the form:

$$\boldsymbol{\pi y} - \sigma \boldsymbol{s} \leq \pi_0,$$

with $\boldsymbol{\pi}$, σ and π_0 nonnegative.

Solve $SEPLP(X^{MK})$:

$$\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{\boldsymbol{s}}\sigma - \pi_0 & \\
& \boldsymbol{w\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in X^{MK} & (3)\\
& \mathbf{1}\boldsymbol{\pi} + \sigma = 1 & (4)\\
& \boldsymbol{\pi}, \sigma, \pi_0 \geq 0 &
\end{aligned}$$

Extreme points of $SEPLP(X^{MK})$ are in one-to-one correspondence with the facets of $conv(X^{MK})$.

Solving $SEPLP(X^{MK})$ by row generation

Step 1 Let $S \subset X^{MK}$ be a subset of the feasible solutions.

Solving $SEPLP(X^{MK})$ by row generation

Step 1 Let $S \subset X^{MK}$ be a subset of the feasible solutions.

Step 2 Solve the *partial separation* problem $SEPLP(S)$:

$$
\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{s}\sigma - \pi_0 \\
& \boldsymbol{w}\boldsymbol{\pi} - t\sigma \le \pi_0, \quad (\boldsymbol{w}, t) \in S \\
& \boldsymbol{\pi} + \sigma = 1 \\
& \boldsymbol{\pi}, \pi_0 \ge 0
\end{aligned}
$$

Let $(\boldsymbol{\pi}^*, \sigma^*, \bar{\pi}_0^*)$ be the optimal solution of $SEPLP(S)$.

Solving $SEPLP(X^{MK})$ by row generation

Step 1 Let $S \subset X^{MK}$ be a subset of the feasible solutions.

Step 2 Solve the *partial separation* problem $SEPLP(S)$:

$$
\begin{aligned}
\max \quad & \bar{\boldsymbol{y}}\boldsymbol{\pi} - \bar{s}\sigma - \pi_0 \\
& \boldsymbol{w}\boldsymbol{\pi} - t\sigma \leq \pi_0, \quad (\boldsymbol{w}, t) \in S \\
& \boldsymbol{\pi} + \sigma = 1 \\
& \boldsymbol{\pi}, \pi_0 \geq 0
\end{aligned}
$$

Let $(\boldsymbol{\pi}^*, \sigma^*, \bar{\pi}_0^*)$ be the optimal solution of $SEPLP(S)$.

Step 3 Solve the mixed-integer knapsack problem *MKNAP*

$$
\begin{aligned}
\max \quad & \boldsymbol{\pi}^* \boldsymbol{w} - \bar{\sigma}^* t \\
& (\boldsymbol{w}, t) \in X^{MK}
\end{aligned}
$$

to check whether the "candidate inequality" $\boldsymbol{\pi}^* \boldsymbol{y} - \sigma^* \boldsymbol{s} \leq \pi_0^*$ is valid for $conv(X^{MK})$.

Solving $SEPLP(X^{MK})$ by row generation (cont.)

Step 4 Let $(\hat{\boldsymbol{w}}, \hat{t})$ be the optimal solution of *MKNAP*. If $\boldsymbol{\pi}^* \hat{\boldsymbol{w}} - \sigma^* \hat{t} > \pi_0^*$ then set $S = S \cup \{(\hat{\boldsymbol{w}}, \hat{t})\}$ and goto **Step 1**.

Solving $SEPLP(X^{MK})$ by row generation (cont.)

Step 4 Let $(\hat{\boldsymbol{w}}, \hat{t})$ be the optimal solution of *MKNAP*. If $\boldsymbol{\pi}^* \hat{\boldsymbol{w}} - \sigma^* \hat{t} > \pi_0^*$ then set $S = S \cup \{(\hat{\boldsymbol{w}}, \hat{t})\}$ and goto **Step 1**.

Step 5 $(\boldsymbol{\pi}^*, \sigma^*, \pi_0^*)$ is the optimal solution of $SEPLP(X^{MK})$ and the inequality $\boldsymbol{\pi}^* \boldsymbol{y} - \sigma^* s \leq \pi_0^*$ is valid for $conv(X^{MK})$.

Solving *MKNAP* efficiently

Solving *MKNAP* efficiently

- The mixed-integer knapsack problem *MKNAP*:

$$\begin{aligned} \max \quad & \boldsymbol{\pi}^* \boldsymbol{w} - \sigma^* t \\ & \boldsymbol{a}\boldsymbol{w} - t \leq b \\ & \boldsymbol{w} \in \mathbb{Z}^n \\ & t \geq 0 \end{aligned}$$

  must be solved repeatedly.

Solving *MKNAP* efficiently

- The mixed-integer knapsack problem *MKNAP*:

$$\begin{aligned} \max \quad & \boldsymbol{\pi}^* \boldsymbol{w} - \sigma^* t \\ & \boldsymbol{a}\boldsymbol{w} - t \leq b \\ & \boldsymbol{w} \in \mathbb{Z}^n \\ & t \geq 0 \end{aligned}$$

  must be solved repeatedly.
- We need a very efficient algorithm.

Solving *MKNAP* efficiently

- The mixed-integer knapsack problem *MKNAP*:

$$\begin{aligned} \max \quad & \pi^* \boldsymbol{w} - \sigma^* t \\ & \boldsymbol{a}\boldsymbol{w} - t \leq b \\ & \boldsymbol{w} \in \mathbb{Z}^n \\ & t \geq 0 \end{aligned}$$

  must be solved repeatedly.
- We need a very efficient algorithm.

Proposition

For any optimal solution $(\hat{\boldsymbol{w}}, \hat{t})$ *of MKNAP we have* $\hat{t} = \max(0, \boldsymbol{a}\hat{\boldsymbol{w}} - b)$.

Solving *MKNAP* efficiently

- The mixed-integer knapsack problem *MKNAP*:

$$\begin{aligned} \max \quad & \pi^* \boldsymbol{w} - \sigma^* t \\ & \boldsymbol{aw} - t \leq b \\ & \boldsymbol{w} \in \mathbb{Z}^n \\ & t \geq 0 \end{aligned}$$

  must be solved repeatedly.
- We need a very efficient algorithm.

Proposition

For any optimal solution $(\hat{\boldsymbol{w}}, \hat{t})$ *of MKNAP we have* $\hat{t} = \max(0, \boldsymbol{a}\hat{\boldsymbol{w}} - b)$.

It follows that:

$$(\hat{t} = 0) \vee (\hat{t} = \boldsymbol{a}\hat{\boldsymbol{w}} - b > 0)$$

Solving *MKNAP* efficiently (cont.)

Proposition

The optimal solution of MKNAP is the best between the optimal solutions of the two following knapsack problems:

Solving *MKNAP* efficiently (cont.)

Proposition

The optimal solution of MKNAP is the best between the optimal solutions of the two following knapsack problems:

KNAP1 $(t = 0)$:

$$\begin{aligned} \max \quad & \pi^* \mathbf{w} \\ & \mathbf{a}\mathbf{w} \leq b \\ & \mathbf{w} \in \mathbb{Z}^n \end{aligned}$$

Solving *MKNAP* efficiently (cont.)

Proposition

The optimal solution of MKNAP is the best between the optimal solutions of the two following knapsack problems:

KNAP1 $(t = 0)$:

$$\begin{aligned} \max \quad & \pi^* \mathbf{w} \\ & \mathbf{a}\mathbf{w} \leq b \\ & \mathbf{w} \in \mathbb{Z}^n \end{aligned}$$

KNAP2 $(t = \mathbf{a}\mathbf{w} - b)$:

$$\begin{aligned} \min \quad & (\bar{\sigma}^* \mathbf{a} - \pi^*)\mathbf{w} \\ & \mathbf{a}\mathbf{w} \geq b + 1 \\ & \mathbf{w} \in \mathbb{Z}^n \end{aligned}$$

Solving *MKNAP* efficiently (cont.)

Proposition

The optimal solution of MKNAP is the best between the optimal solutions of the two following knapsack problems:

KNAP1 $(t = 0)$:

$$\begin{aligned} \max \quad & \pi^* \boldsymbol{w} \\ & \boldsymbol{a}\boldsymbol{w} \leq b \\ & \boldsymbol{w} \in \mathbb{Z}^n \end{aligned}$$

KNAP2 $(t = \boldsymbol{a}\boldsymbol{w} - b)$:

$$\begin{aligned} \min \quad & (\bar{\sigma}^* \boldsymbol{a} - \pi^*) \boldsymbol{w} \\ & \boldsymbol{a}\boldsymbol{w} \geq b + 1 \\ & \boldsymbol{w} \in \mathbb{Z}^n \end{aligned}$$

!!

Both the knapsack problems can be solved very fast by dynamic programming (Pisinger, 2004).

Implementation details

When embedded into a cutting plane algorithm, $SEPLP(X^{MK})$ is applied to each row defining a mixed-integer knapsack set:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}.$$

Some operations are required to put the row in the “right” form:

Implementation details

When embedded into a cutting plane algorithm, $SEPLP(X^{MK})$ is applied to each row defining a mixed-integer knapsack set:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}.$$

Some operations are required to put the row in the “right” form:

Bound substitution: replace a subset of continuous variable by their simple or variable bounds.

Implementation details

When embedded into a cutting plane algorithm, $SEPLP(X^{MK})$ is applied to each row defining a mixed-integer knapsack set:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}.$$

Some operations are required to put the row in the “right” form:

Bound substitution: replace a subset of continuous variable by their simple or variable bounds.

Preprocessing: transform the mixed integer set X^{MI} into the mixed-integer knapsack set X^{MK}.

Implementation details

When embedded into a cutting plane algorithm, $SEPLP(X^{MK})$ is applied to each row defining a mixed-integer knapsack set:

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{x} \leq \boldsymbol{v}\}.$$

Some operations are required to put the row in the “right” form:

Bound substitution: replace a subset of continuous variable by their simple or variable bounds.

Preprocessing: transform the mixed integer set X^{MI} into the mixed-integer knapsack set X^{MK}.

Convert coefficients into integers (required to use dynamic programming)

Bound substitution

- Consider the mixed-integer set

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{lx} \leq \boldsymbol{v}\}.$$

Bound substitution

- Consider the mixed-integer set

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \boldsymbol{a}\boldsymbol{y} + \boldsymbol{g}\boldsymbol{x} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{l}\boldsymbol{x} \leq \boldsymbol{v}\}.$$

- The MIP formulation can also include some additional variable bounds on the continuous variables.

Bound substitution

- Consider the mixed-integer set

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{lx} \leq \boldsymbol{v}\}.$$

- The MIP formulation can also include some additional variable bounds on the continuous variables.
- Bound substitution consists of replacing some continuous variables by their respective simple/variable bounds. It is done heuristically by performing one of the following substitutions:

$$x_j = l_j + x_j'; x_j = v_j - x_j'; x_j = \tilde{l}_j y_i + x_j'; w_j = \tilde{v}_j y_k - x_j'$$

Bound substitution

- Consider the mixed-integer set

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{lx} \leq \boldsymbol{v}\}.$$

- The MIP formulation can also include some additional variable bounds on the continuous variables.
- Bound substitution consists of replacing some continuous variables by their respective simple/variable bounds. It is done heuristically by performing one of the following substitutions:

$$x_j = l_j + x'_j; x_j = v_j - x'_j; x_j = \tilde{l}_j y_i + x'_j; w_j = \tilde{v}_j y_k - x'_j$$

- Let $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ be the current fractional solution. The bound with smallest slack is selected for substitution. That is, let

$$\mu = \min\{\bar{x}_j - l_j, \ v_j - \bar{x}_j, \ \bar{x}_j - \tilde{l}_j \bar{y}_i, \ \tilde{v}_j \bar{y}_k - \bar{x}_j\}.$$

Bound substitution

- Consider the mixed-integer set

$$X^{MI} = \{(\boldsymbol{y}, \boldsymbol{x}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : \ \ \boldsymbol{ay} + \boldsymbol{gx} \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}, \boldsymbol{lx} \leq \boldsymbol{v}\}.$$

- The MIP formulation can also include some additional variable bounds on the continuous variables.
- Bound substitution consists of replacing some continuous variables by their respective simple/variable bounds. It is done heuristically by performing one of the following substitutions:

$$x_j = l_j + x'_j; x_j = v_j - x'_j; x_j = \tilde{l}_j y_i + x'_j; w_j = \tilde{v}_j y_k - x'_j$$

- Let $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ be the current fractional solution. The bound with smallest slack is selected for substitution. That is, let

$$\mu = \min\{\bar{x}_j - l_j, \ v_j - \bar{x}_j, \ \bar{x}_j - \tilde{l}_j \bar{y}_i, \ \tilde{v}_j \bar{y}_k - \bar{x}_j\}.$$

- Let:

$$x_j = \begin{cases} l_j + x'_j & \text{if } \mu = x_j - l_j \\ v_j - x'_j & \text{if } \mu = v_j - \bar{x}_j \\ \tilde{l}_j y_i + x'_j & \text{if } \mu = \bar{x}_j - \tilde{l}_j \\ \tilde{v}_j y_k - x'_j & \text{if } \mu = \tilde{v}_j \bar{y}_k - \bar{x}_j \end{cases}$$

Preprocessing

Let

$$\sum_{i \in I} a'_i y_i + \sum_{j \in P} g'_j x'_j \leq b',$$

with $0 \leq y_i \leq u_i \; \forall j \in I$ and $x'_j \geq 0 \; \forall j \in P$, be the mixed-integer inequality after bound substitution.

Preprocessing

Let

$$\sum_{i \in I} a'_i y_i + \sum_{j \in P} g'_j x'_j \leq b',$$

with $0 \leq y_i \leq u_i \; \forall j \in I$ and $x'_j \geq 0 \; \forall j \in P$, be the mixed-integer inequality after bound substitution.

- All the continuous variables with positive coefficients can be discarded (Atamturk, 2000).

Preprocessing

Let

$$\sum_{i \in I} a'_i y_i + \sum_{j \in P} g'_j x'_j \leq b',$$

with $0 \leq y_i \leq u_i \; \forall j \in I$ and $x'_j \geq 0 \; \forall j \in P$, be the mixed-integer inequality after bound substitution.

- All the continuous variables with positive coefficients can be discarded (Atamturk, 2000).
- All the continuous variables with negative coefficients are aggregated into the same variable s:

$$s = -\sum_{j \in P^-} g'_j x'_j,$$

where $P^- = \{j \in P : \; g'_j < 0\}$.

Preprocessing

Let

$$\sum_{i \in I} a_i' y_i + \sum_{j \in P} g_j' x_j' \leq b',$$

with $0 \leq y_i \leq u_i \ \forall j \in I$ and $x_j' \geq 0 \ \forall j \in P$, be the mixed-integer inequality after bound substitution.

- All the continuous variables with positive coefficients can be discarded (Atamturk, 2000).
- All the continuous variables with negative coefficients are aggregated into the same variable s:

  $$s = -\sum_{j \in P^-} g_j' x_j',$$

  where $P^- = \{j \in P : \ g_j' < 0\}$.
- All the integer variables with negative coefficients are complemented:

  $$y_j = \begin{cases} u_j - y_j' & \text{if } a_j' < 0 \\ y_j' & \text{otherwise} \end{cases}$$

Convert all the coefficients into integers

- The integer knapsack problems of *MKNAP* are solved by the dynamic programming algorithm of Pisinger (2001).

Convert all the coefficients into integers

- The integer knapsack problems of *MKNAP* are solved by the dynamic programming algorithm of Pisinger (2001).
- Dynamic programming is fast, but there is a price to pay: it requires that all the knapsack coefficients are integers.

Convert all the coefficients into integers

- The integer knapsack problems of *MKNAP* are solved by the dynamic programming algorithm of Pisinger (2001).
- Dynamic programming is fast, but there is a price to pay: it requires that all the knapsack coefficients are integers.
- The coefficients of the integer variables must be converted into suitably small integers before running exact separation.

Convert all the coefficients into integers

- The integer knapsack problems of *MKNAP* are solved by the dynamic programming algorithm of Pisinger (2001).

- Dynamic programming is fast, but there is a price to pay: it requires that all the knapsack coefficients are integers.

- The coefficients of the integer variables must be converted into suitably small integers before running exact separation.

- We adopt a brute-force approach: enumerate all the $q \in \mathbb{N}$ in the interval $[1, 10^4]$, stopping when $qb'' - \lfloor qb'' \rfloor \leq \varepsilon$ and $qa_j'' - \lfloor qa_j'' \rfloor \leq \varepsilon$ for each $j \in I$. In our experiments we set $\varepsilon = 10^{-5}$.

Convert all the coefficients into integers

- The integer knapsack problems of *MKNAP* are solved by the dynamic programming algorithm of Pisinger (2001).
- Dynamic programming is fast, but there is a price to pay: it requires that all the knapsack coefficients are integers.
- The coefficients of the integer variables must be converted into suitably small integers before running exact separation.
- We adopt a brute-force approach: enumerate all the $q \in \mathbb{N}$ in the interval $[1, 10^4]$, stopping when $qb'' - \lfloor qb'' \rfloor \leq \varepsilon$ and $qa''_j - \lfloor qa''_j \rfloor \leq \varepsilon$ for each $j \in I$. In our experiments we set $\varepsilon = 10^{-5}$.
- If the procedure fails, we discard the inequality since too large coefficients may cause numerical problems.

Lifting

- Exact separation runs over the fractional support.

Lifting

- Exact separation runs over the fractional support.
- Then standard sequential lifting is used to get globally valid inequalities.

Lifting

- Exact separation runs over the fractional support.
- Then standard sequential lifting is used to get globally valid inequalities.
- Computing a lifting coefficient amounts to solve a knapsack problem with a single continuous variable. The problem can be solved by splitting into two integer knapsack problems.

Computational results

- Computational experiments were carried out on a 64bit Pentium Quad-core 2.6 GHz processor with 4 Gb RAM. The LP solver was Xpress 2007B.

Computational results

- Computational experiments were carried out on a 64bit Pentium Quad-core 2.6 GHz processor with 4 Gb RAM. The LP solver was Xpress 2007B.
- The test bed consists of all the MIPLIB 2003 mixed-integer instances and of the “Mittleman” instances *bc1, bienst1, bienst2, binkar10_1, dano3-4, dano3-5*. We set a limit of 300 CPU secs for the time spent in separation.

Computational results (cont.)

- We compare the lower bounds returned by exact separation with those provided by Mixed-Integer Rounding (MIR) inequalities

Computational results (cont.)

- We compare the lower bounds returned by exact separation with those provided by Mixed-Integer Rounding (MIR) inequalities
- We compare with the MIR separation procedure of K. Wolter (2007), embedded into SCIP version 1.00, linked to SoPlex 1.3.2 and ZIMPL 2.07 (Achterberg, 2007).

Computational results (cont.)

- We compare the lower bounds returned by exact separation with those provided by Mixed-Integer Rounding (MIR) inequalities
- We compare with the MIR separation procedure of K. Wolter (2007), embedded into SCIP version 1.00, linked to SoPlex 1.3.2 and ZIMPL 2.07 (Achterberg, 2007).
- We set SCIP parameters to perfom Wolter's procedure on single rows, i.e. to forbid constraint aggregation. Separation of Lifted Cover inequalities is enabled too.

Computational results (cont.)

- We compare the lower bounds returned by exact separation with those provided by Mixed-Integer Rounding (MIR) inequalities
- We compare with the MIR separation procedure of K. Wolter (2007), embedded into SCIP version 1.00, linked to SoPlex 1.3.2 and ZIMPL 2.07 (Achterberg, 2007).
- We set SCIP parameters to perfom Wolter's procedure on single rows, i.e. to forbid constraint aggregation. Separation of Lifted Cover inequalities is enabled too.
- For simplicity of comparison, separation routines run on the original (i.e. not preprocessed) instances.

Computational results

Name	SCIP LB	SCIP %Gap	SCIP Time	*MK*-SEP LB	*MK*-SEP % Gap	*MK*-SEP Time
10teams	917.00	0.00	0.08	917	0.00	0.96
a1c1s1	997.53	0.00	0.14	997.53	0.00	2.20
aflow30a	983.16	0.00	0.00	1053.29	40.11	10.07
aflow40b	1005.50	0.00	0.03	1058.32	32.50	10.67
arki001	7579599.81	0.00	0.46	7579599.81	0.00	0.89
atlanta-ip	81.25	0.11	11.14	82.46	13.91	300.00
dano3mip	576.23	0.00	0.56	576.23	0.00	7.40
danoint	62.63	0.00	0.01	62.66	0.88	3.59
fiber	385094.10	91.66	0.27	390493.82	93.82	9.26
fixnet6	3192.04	71.57	0.09	3442.60	80.58	196.21
gesa2	25691081	71.28	0.44	25701859	74.86	4.29
gesa2-o	25476489	0.0	0.06	25588105	37.02	7.79
glass4	800002400	0.00	0.01	800002400	0.00	0.23
liu	385.00	4.92	0.64	385.00	4.92	8.76
markshare1	0.00	0.00	0.00	0.00	0.00	43.79
markshare2	0.00	0.00	0.00	0.00	0.00	26.82
mas74	10482.79	0.00	0.00	10482.79	0.00	0.25
mas76	38901.02	0.64	0.00	38901.02	0.64	0.16
misc07	1415.00	0.00	0.00	1415	0.00	0.54
mkc	-607.18	9.73	4.62	-605.83	12.54	56.40
modglob	20430947.60	0.00	0.02	20431515.90	0.18	9.20
msc98-ip	19538746.75	5.58	16.65	19559084.16	11.97	169.54
net12	31.55	7.27	7.97	32.08	7.54	106.53
nsrand-ipx	49851.43	41.87	4.92	49877.59	43.00	80.75
roll3000	12072.71	54.41	2.13	12073.49	54.46	23.06
swath	334.50	0.00	0.53	334.5	0.00	9.18
timtab1	195605.34	22.68	0.07	229628.78	27.30	3.23
timtab2	250004.21	16.43	0.16	270295.07	18.43	6.84
tr12-30	18124.17	3.36	0.01	84403.46	60.27	8.23
vpm2	10.40	13.21	0.02	11.21	33.94	1.59
binkar10_1	6701.56	61.42	1.33	6720.55	79.54	9.06
bienst1	11.72	0.00	0.01	14.01	6.54	2.15
bienst2	11.72	0.00	0.00	14.88	7.41	3.18
dano3-4	576.23	0.00	0.41	576.23	0.00	2.76
dano3-5	576.23	0.00	0.52	576.23	0.00	3.15
rgn	68.00	57.49	0.00	68.00	57.49	1.14

The overall effect

Some preliminary tests on non-trivial instances (Cplex 11.1)

timtab1 After 30000 B&B nodes: the original formulation returned a relative gap of 12.30%. Using exact separation the relative gap is 10.23%.

The overall effect

Some preliminary tests on non-trivial instances (Cplex 11.1)

timtab1 After 30000 B&B nodes: the original formulation returned a relative gap of 12.30%. Using exact separation the relative gap is 10.23%.

timtab2 After 20000 B&B nodes: with the original formulation the gap is 37.6%. Using exact separation the relative gap is 34.5%.

The overall effect

Some preliminary tests on non-trivial instances (Cplex 11.1)

- **timtab1** After 30000 B&B nodes: the original formulation returned a relative gap of 12.30%. Using exact separation the relative gap is 10.23%.
- **timtab2** After 20000 B&B nodes: with the original formulation the gap is 37.6%. Using exact separation the relative gap is 34.5%.
- **aflow40** Solved in 46826 nodes by Cplex 11.1 after exact separation. Solved in 96400 nodes on the original formulation.

The overall effect

Some preliminary tests on non-trivial instances (Cplex 11.1)

- **timtab1** After 30000 B&B nodes: the original formulation returned a relative gap of 12.30%. Using exact separation the relative gap is 10.23%.
- **timtab2** After 20000 B&B nodes: with the original formulation the gap is 37.6%. Using exact separation the relative gap is 34.5%.
- **aflow40** Solved in 46826 nodes by Cplex 11.1 after exact separation. Solved in 96400 nodes on the original formulation.
- **tr12-30** After 20000 B&B nodes: with the original formulation the relative gap is 2.76%. Using exact separation the relative gap is 0.29%.

The overall effect

Some preliminary tests on non-trivial instances (Cplex 11.1)

- **timtab1** After 30000 B&B nodes: the original formulation returned a relative gap of 12.30%. Using exact separation the relative gap is 10.23%.
- **timtab2** After 20000 B&B nodes: with the original formulation the gap is 37.6%. Using exact separation the relative gap is 34.5%.
- **aflow40** Solved in 46826 nodes by Cplex 11.1 after exact separation. Solved in 96400 nodes on the original formulation.
- **tr12-30** After 20000 B&B nodes: with the original formulation the relative gap is 2.76%. Using exact separation the relative gap is 0.29%.
- **nsrand-ip** After 5000 B&B nodes: with the original formulation the gap is 1.5%. Using exact separation the relative gap is 0.37%.

Some considerations

- Computational experience shows that exact separation for $conv(X^{MK})$ is effective in tightening MIP formulations.

Some considerations

- Computational experience shows that exact separation for $conv(X^{MK})$ is effective in tightening MIP formulations.
- Computation time is much larger than for MIR separation, but still reasonable when dealing with hard instances.

Some considerations

- Computational experience shows that exact separation for $conv(X^{MK})$ is effective in tightening MIP formulations.
- Computation time is much larger than for MIR separation, but still reasonable when dealing with hard instances.
- Exact separation not applicable to large and dense rows.

Back to the template paradigm: Mixed Knapsack Inequalities

We focus on mixed knapsack inequalities (Marchand and Wolsey, 2002), which can described by the following procedure.
Given:

$$X^{BMK} = \{(\boldsymbol{y}, s) \in \mathbb{B}^n_+ \times \mathbb{R}_+ : \ \ \boldsymbol{a}\boldsymbol{y} - s \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

Back to the template paradigm: Mixed Knapsack Inequalities

We focus on mixed knapsack inequalities (Marchand and Wolsey, 2002), which can described by the following procedure.
Given:

$$X^{BMK} = \{(\boldsymbol{y}, s) \in \mathbb{B}_+^n \times \mathbb{R}_+ : \ \boldsymbol{ay} - s \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

i) Set the $s = \bar{s}$;

Back to the template paradigm: Mixed Knapsack Inequalities

We focus on mixed knapsack inequalities (Marchand and Wolsey, 2002), which can described by the following procedure.
Given:

$$X^{BMK} = \{(\boldsymbol{y}, s) \in \mathbb{B}^n_+ \times \mathbb{R}_+ : \ \ \boldsymbol{ay} - s \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

i) Set the $s = \bar{s}$;

ii) Find a valid inequality $\alpha\boldsymbol{y} \leq \beta$ for the resulting binary knapsack polytope;

$$X^{BMK}_{\bar{s}} = \{\boldsymbol{y} \in \mathbb{B}^n_+ : \ \ \boldsymbol{ay} \leq b + \bar{s}, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

Back to the template paradigm: Mixed Knapsack Inequalities

We focus on mixed knapsack inequalities (Marchand and Wolsey, 2002), which can described by the following procedure.
Given:

$$X^{BMK} = \{(\boldsymbol{y}, s) \in \mathbb{B}_+^n \times \mathbb{R}_+ : \ \boldsymbol{ay} - s \leq b, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

i) Set the $s = \bar{s}$;

ii) Find a valid inequality $\boldsymbol{\alpha y} \leq \beta$ for the resulting binary knapsack polytope;

$$X_{\bar{s}}^{BMK} = \{\boldsymbol{y} \in \mathbb{B}_+^n : \ \boldsymbol{ay} \leq b + \bar{s}, \ \boldsymbol{y} \leq \boldsymbol{u}\}$$

iii) lift the s to get a valid inequality for X_{BMK} of the form $\boldsymbol{\alpha y} - \gamma s \leq \beta$.

Mixed Knapsack Inequalities: lifting the s

Let

$$\eta(s) = \max \quad \boldsymbol{\alpha y} \tag{5}$$
$$\boldsymbol{ay} \leq b + s \tag{6}$$
$$\boldsymbol{y} \in \{0, 1\}^n \tag{7}$$

Mixed Knapsack Inequalities: lifting the s

Let

$$\eta(s) = \max \quad \boldsymbol{\alpha y} \tag{5}$$
$$\boldsymbol{ay} \leq b + s \tag{6}$$
$$\boldsymbol{y} \in \{0,1\}^n \tag{7}$$

Proposition

The inequality

$$\boldsymbol{\alpha y} \leq \beta + \gamma s$$

is valid for conv(X^{BMK}) *if* $\eta(s) \leq \beta + \gamma s$ *for each* $s \in \mathbb{R}_+$*.*

Mixed Knapsack Inequalities: lifting the s (cont.)

A geometrical interpretation

Mixed Knapsack Inequalities: lifting the s (cont.)

A geometrical interpretation

- $\eta(s)$ is a step function

Mixed Knapsack Inequalities: lifting the *s* (cont.)

A geometrical interpretation

- $\eta(s)$ is a step function
- the line $\beta + \gamma s$ is a "valid" rhs if it defines an upper bound on the $\eta(s)$, for each $s \in \mathbb{R}_+$.

Mixed Knapsack Inequalities: lifting the s (cont.)

Mixed Knapsack Inequalities: lifting the s (cont.)

The lifting algorithm

Mixed Knapsack Inequalities: lifting the s (cont.)

The lifting algorithm

Step 0 Inizialize γ.

Mixed Knapsack Inequalities: lifting the s (cont.)

The lifting algorithm

Step 0 Inizialize γ.

Step 1 Solve the problem:

$$
\begin{aligned}
\zeta = \quad \max \quad & \boldsymbol{\alpha y} - \gamma s \\
& \boldsymbol{\alpha y} \leq b + s \\
& \boldsymbol{y} \in \{0, 1\}^N \\
& s \geq 0
\end{aligned}
$$

Mixed Knapsack Inequalities: lifting the s (cont.)

The lifting algorithm

Step 0 Inizialize γ.

Step 1 Solve the problem:

$$\begin{aligned} \zeta = \quad \max \quad & \boldsymbol{\alpha y} - \gamma s \\ & \boldsymbol{\alpha y} \leq b + s \\ & \boldsymbol{y} \in \{0,1\}^N \\ & s \geq 0 \end{aligned}$$

Step 2 If $\zeta \leq \beta$ then the inequality $\boldsymbol{\alpha y} \leq \beta + \gamma s$ is valid for $conv(X^{BMK})$. STOP.

Mixed Knapsack Inequalities: lifting the s (cont.)

The lifting algorithm

Step 0 Inizialize γ.

Step 1 Solve the problem:

$$\begin{aligned} \zeta = \quad \max \quad & \boldsymbol{\alpha y} - \gamma s \\ & \boldsymbol{\alpha y} \leq b + s \\ & \boldsymbol{y} \in \{0, 1\}^N \\ & s \geq 0 \end{aligned}$$

Step 2 If $\zeta \leq \beta$ then the inequality $\boldsymbol{\alpha y} \leq \beta + \gamma s$ is valid for $conv(X^{BMK})$. STOP.

Step 3 Increase γ and Go to Step 1.

Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.

Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.
- Let $X_1^{BMK}\{x \in X^{BMK} : s = 1\}$.

Mixed Knapsack Inequalities: lifting the s (cont.)

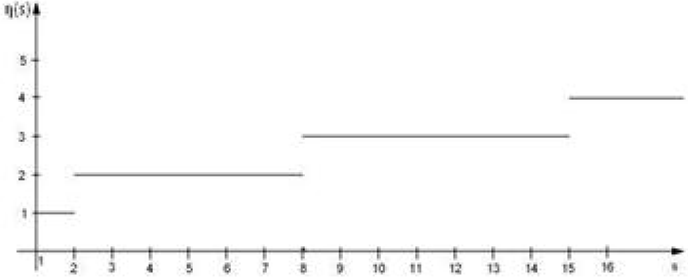
A numerical example

- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.
- Let $X_1^{BMK} \{x \in X^{BMK} : s = 1\}$.
- The inequality $y_1 + y_2 + y_3 + y_4 \leq 1$ is valid for $conv(X_1^{BMK})$.

Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

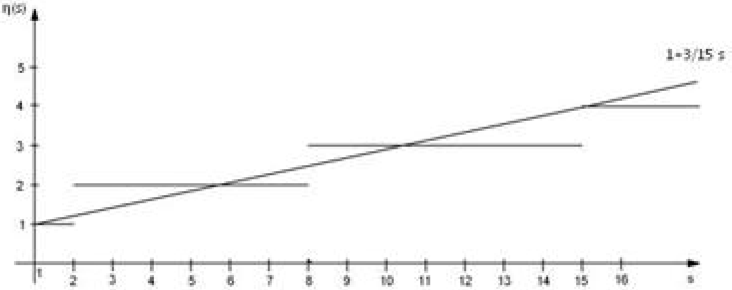
- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.
- Let $X_1^{BMK}\{x \in X^{BMK} : s = 1\}$.
- The inequality $y_1 + y_2 + y_3 + y_4 \leq 1$ is valid for $conv(X_1^{BMK})$.
- $\eta(s)$ step function.



Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

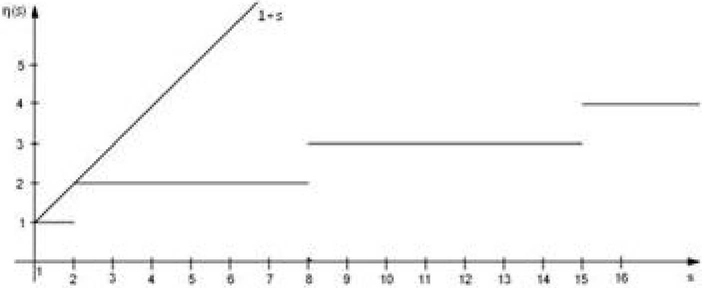
- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.
- Let $X_1^{BMK}\{x \in X^{BMK} : s = 1\}$.
- The inequality $y_1 + y_2 + y_3 + y_4 \leq 1$ is valid for $conv(X_1^{BMK})$.
- Initialization: $\gamma = 3/15$.



Mixed Knapsack Inequalities: lifting the s (cont.)

A numerical example

- Consider the set $X^{BMK} = \{7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\}$.
- Let $X_1^{BMK}\{x \in X^{BMK} : s = 1\}$.
- The inequality $y_1 + y_2 + y_3 + y_4 \leq 1$ is valid for $conv(X_1^{BMK})$.
- Iteration 1: update $\gamma = 1$; $y_1 + y_2 + y_3 + y_4 - s \leq 1$ is valid.



Computational results for Mixed Knapsack Inequalities

Name	SCIP LB	SCIP %Gap	SCIP Time	LCI LB	LCI % Gap	LCI Time
10teams	917.00	0.00	0.08	917	0.00	0.03
a1c1s1	997.53	0.00	0.14	997.53	0.00	0.66
aflow30a	983.16	0.00	0.00	1013.92	17.59	0.62
aflow40b	1005.50	0.00	0.03	1017.39	7.32	1.29
arki001	7579599.81	0.00	0.46	7579599.81	0.00	0.78
atlanta-ip	81.25	0.11	11.14	82.33	12.43	135.31
dano3mip	576.23	0.00	0.56	576.23	0.00	11.33
danoint	62.63	0.00	0.01	62.65	0.66	0.09
fiber	385094.10	91.66	0.27	385094.10	91.66	0.31
fixnet6	3192.04	71.57	0.09	3441.08	80.52	4.15
gesa2	25691081	71.28	0.44	25691081	71.28	0.11
gesa2-o	25476489	0.0	0.06	25476489	0.00	0.06
glass4	800002400	0.00	0.01	800002400	0.00	0.03
liu	385.00	4.92	0.64	385.00	4.92	0.64
markshare1	0.00	0.00	0.00	0.00	0.00	0.03
markshare2	0.00	0.00	0.00	0.00	0.00	0.01
mas74	10482.79	0.00	0.00	10482.79	0.00	0.02
mas76	38901.02	0.64	0.00	38901.02	0.64	0.08
misc07	1415.00	0.00	0.00	1415	0.00	0.01
mkc	-607.18	9.73	4.62	-611.48	0.77	1.61
modglob	20430947.60	0.00	0.02	20431458.13	0.16	0.05
msc98-ip	19538746.75	5.58	16.65	19557387.00	11.43	57.76
net12	31.55	7.27	7.97	31.91	7.45	37.92
nsrand-ipx	49851.43	41.87	4.92	49851.67	41,88	13.64
roll3000	12072.71	54.41	2.13	12072.71	54.41	0.17
swath	334.50	0.00	0.53	334.5	0.00	0.61
timtab1	195605.34	22.68	0.07	213136.28	25.06	0.03
timtab2	250004.21	16.43	0.16	250086.12	16.44	0.08
tr12-30	18124.17	3.36	0.01	84363.73	60.24	0.26
vpm2	10.40	13.21	0.02	11.31	36,79	0.08
binkar10_1	6701.56	61.42	1.33	6637.18	0.00	0.14
bienst1	11.72	0.00	0.01	14.03	6.59	0.22
bienst2	11.72	0.00	0.00	14.88	7.41	0.27
dano3-4	576.23	0.00	0.41	576.23	0.00	11.67
dano3-5	576.23	0.00	0.52	576.23	0.00	12.45
rgn	68.00	57.49	0.00	68.00	57.49	0.01

Research directions

- More efficient ways of solving the exact separation LP

Research directions

- More efficient ways of solving the exact separation LP
- How to generate new rows by constraint aggregation?

Research directions

- More efficient ways of solving the exact separation LP
- How to generate new rows by constraint aggregation?
- Looking at more complex MIP substructures than the single row.

Research directions

- More efficient ways of solving the exact separation LP
- How to generate new rows by constraint aggregation?
- Looking at more complex MIP substructures than the single row.
- How to select MIP substructures to ensure that exact separation leads to violated cuts?