Safe Verified Gomory Mixed Integer Cuts for Exact Rational MIP

Leon Eifler [1], Ambros Gleixner [1,2]

[1]Zuse Institute Berlin [2]HTW Berlin

eifler@zib.de

SCIP 20th anniversary

Berlin · Nov 04 2022

SPONSORED BY THE
Federal Ministry of Education and Research

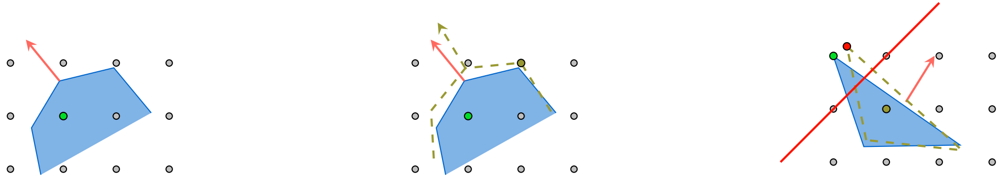
How exact are MIP solvers?

Proven optimality guarantees are a unique selling point of MIP solvers:

$$L \leq c^T x^* \leq U$$

with primal-dual gap $U - L \to 0$.

In a strict mathematical sense, this promise is **compromised by rounding errors**



exact solution

fp solution

invalid model strengthening

Closing the gap

Proven optimality guarantees are a unique selling point of MIP solvers:

$$L \leq c^T x^* \leq U$$

with primal-dual gap $U - L \to 0$.

In a strict mathematical sense, this promise is compromised by rounding errors

Goal: close this gap between theory and practice of MIP by providing

(a) a roundoff-error-free MIP solver with
(b) with comparable performance and
(c) solver-independent verification of results.

Outline

1. Previous work: hybrid-precision algorithms for LP and MIP

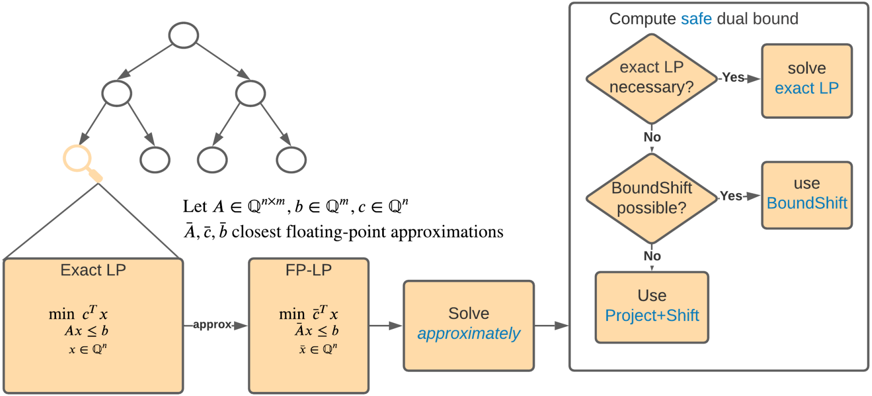
2. Safe Gomory mixed-integer cuts and their performance

3. Verification / Proof-logging

Hybrid-Precision solving of exact MIPs

Cook, Koch, Steffy, Wolter (2013) [2]

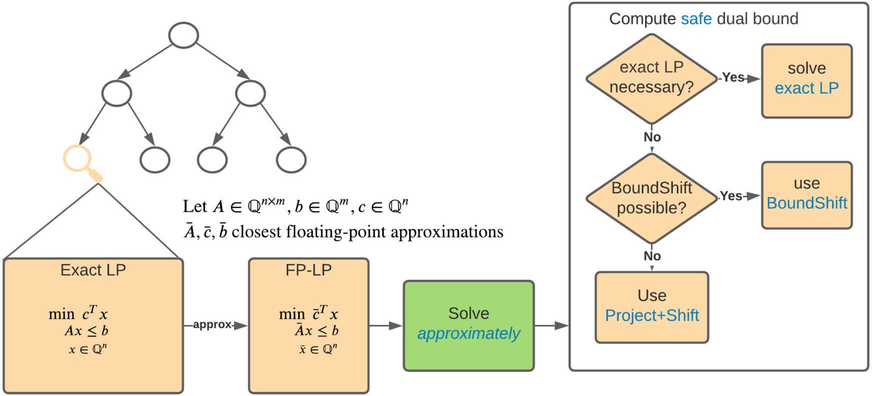
Given $A \in \mathbb{Q}^{n\times m}$, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, consider the clostest floating-point representable approximation $\bar{A}, \bar{c}, \bar{b}$



Hybrid-Precision solving of exact MIPs

Cook, Koch, Steffy, Wolter (2013) [2]

Given $A \in \mathbb{Q}^{n \times m}, c \in \mathbb{Q}^n, b \in \mathbb{Q}^m$, consider the clostest floating-point representable approximation $\bar{A}, \bar{c}, \bar{b}$



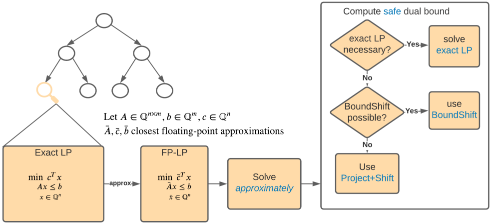
Hybrid-Precision Solving of Exact MIPs

Available in exact solver[3]

- presolving
- primal heuristics

Algorithmic gap to state-of-the-art

- no cutting planes
- no domain propagation
- no conflict analysis



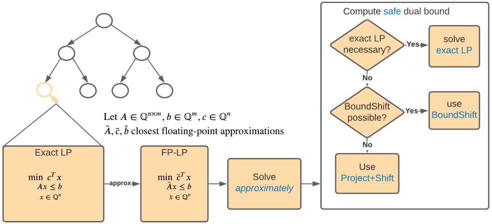
Hybrid-Precision Solving of Exact MIPs

Available in exact solver[3]

- presolving
- primal heuristics

Algorithmic gap to state-of-the-art

- no cutting planes
- no domain propagation
- no conflict analysis



Safe GMI cuts via mixed integer rounding

Given a one-row relaxation $a^T x \leq b, f := b - \lfloor b \rfloor, f_i := a_i - \lfloor a_i \rfloor$,

$$\sum_{i \in \mathcal{I}} \left(\lfloor a_i \rfloor + \frac{(f_i - f)^+}{1-f} \right) x_i \leq \lfloor b \rfloor + \sum_{i \notin \mathcal{I}} \frac{a_i^-}{1-f} x_i$$

is valid.
For GMI cuts $a^T x \leq b$ is obtained as a row of the optimal simplex tableau.

Safe GMI cuts via mixed integer rounding

Given a one-row relaxation $a^T x \leq b$, $f := b - \lfloor b \rfloor$, $f_i := a_i - \lfloor a_i \rfloor$,

$$\sum_{i \in \mathcal{I}} \left(\lfloor a_i \rfloor + \frac{(f_i - f)^+}{1 - f} \right) x_i \leq \lfloor b \rfloor + \sum_{i \notin \mathcal{I}} \frac{a_i^-}{1 - f} x_i$$

is valid.
For GMI cuts $a^T x \leq b$ is obtained as a row of the optimal simplex tableau.

Requires exact LP solution to be directly usable

Safe GMI cuts via mixed integer rounding

Given a one-row relaxation $a^T x \leq b$, $f := b - \lfloor b \rfloor$, $f_i := a_i - \lfloor a_i \rfloor$,

$$\sum_{i \in \mathcal{I}} \left(\lfloor a_i \rfloor + \frac{(f_i - f)^+}{1 - f} \right) x_i \leq \lfloor b \rfloor + \sum_{i \notin \mathcal{I}} \frac{a_i^-}{1 - f} x_i$$

is valid.
For GMI cuts $a^T x \leq b$ is obtained as a row of the optimal simplex tableau.

Requires exact LP solution to be directly usable

Instead use approximation of A_B^{-1} and generate guaranteed feasible row using safe directed rounding. (**Cook, Dash, Fukasawa, Gooycolea (2009)** [1])

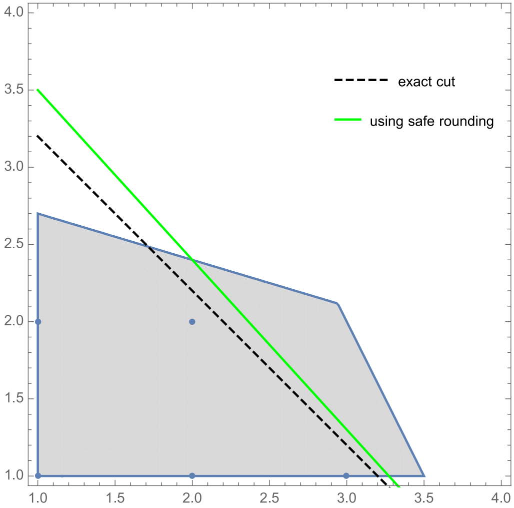
Safe rounding trick

- Assume each variable has at least upper or lower bound
- Transform to non-negative variable space:

$$x \le u \qquad x' = u - x \ge 0$$
$$x \ge l \qquad x' = x - l \ge 0$$

- Round down all coefficients on left hand side, round up on right hand side
- Transform back to original variable-space

Figure: Visualization of safe cutting plane idea



Theorem

Given two valid, floating-point representable inequalities $a^T x \leq b$, $c^T x \leq d$, *and* $\lambda > 0$.
We can generate an approximation of the aggregated inequality $(a + \lambda c)^T x \leq b + \lambda d$ *by*

$$\sum_{i \in U} \overline{\alpha_i} x_i + \sum_{i \in L} \underline{\alpha_i} x_i \leq \overline{d + \sum_{i \in U} (\overline{\alpha_i} - \underline{\alpha_i}) u_i + \sum_{i \in L} (\underline{\alpha_i} - \overline{\alpha_i}) l_i},$$

with $\alpha_i := a_i + \lambda_i c_i$.

Theorem

Given two valid, floating-point representable inequalities $a^T x \leq b$, $c^T x \leq d$, *and* $\lambda > 0$.
We can generate an approximation of the aggregated inequality $(a + \lambda c)^T x \leq b + \lambda d$ *by*

$$\sum_{i \in U} \overline{\alpha_i} x_i + \sum_{i \in L} \underline{\alpha_i} x_i \leq \overline{d + \sum_{i \in U} (\overline{\alpha_i} - \underline{\alpha_i}) u_i + \sum_{i \in L} (\underline{\alpha_i} - \overline{\alpha_i}) l_i},$$

with $\alpha_i := a_i + \lambda_i c_i$.

Use same trick for:

- fp-approximation of problem data
- aggregation with multipliers from floating-point tableau
- transformation to nonnegative variable space
- MIR formula
- resubstitution of slack variables
- cut scaling

Performance impact

MIPLIB 2017 Benchmark / 2 h time limit

setting	solved	time (rel)	nodes (rel)	exlptime (rel)
baseline	132	873.4	12555.9	74.0
cuts	136	793.7 (0.91)	5330.1 (0.42)	102.8 (1.39)

Observe:

Plain procedure yields **drastic slow down** on **few exact LPs**

Performance impact

MIPLIB 2017 Benchmark / 2 h time limit

setting	solved	time (rel)	nodes (rel)	exlptime (rel)
baseline	132	873.4	12555.9	74.0
cuts	136	793.7 (0.91)	5330.1 (0.42)	102.8 (1.39)

Observe:

Plain procedure yields drastic slow down on few exact LPs

Performance impact

MIPLIB 2017 Benchmark / 2 h time limit

setting	solved	time (rel)	nodes (rel)	exlptime (rel)
baseline	132	873.4	12555.9	74.0
cuts	136	793.7 (0.91)	5330.1 (0.42)	102.8 (1.39)

Observe:

Plain procedure yields **drastic slow down** on **few exact LPs**

⇝ control encoding length of coefficients by rounding to lower denominators

- impose a limit on the maximal size denominators can attain ($2^{17} = 131072$)
- compute best approximations using continued fractions
- offset right hand side by multiplying difference with variable bounds to ensure cut validity

Performance impact

MIPLIB 2017 Benchmark / 2 h time limit

setting	solved	time (rel)	nodes (rel)	exlptime (rel)
baseline	132	873.4	12555.9	74.0
cuts	136	793.7 (0.91)	5330.1 (0.42)	102.8 (1.39)
cuts/d-round	145	739.9 (0.85)	6322.5 (0.50)	70.7 (0.96)

Observe:

Plain procedure yields **drastic slow down** on **few exact LPs**

⇝ control encoding length of coefficients by rounding to lower denominators

- impose a limit on the maximal size denominators can attain ($2^{17} = 131072$)
- compute best approximations using continued fractions
- offset right hand side by multiplying difference with variable bounds to ensure cut validity

Verification/Proof-logging using VIPR-Certificates

The VIPR certificate format consists of

- the problem specification
- assumptions (e.g. branching decision, split disjunctions)
- aggregated constraints (most often $c^T x \geq b$, derived by aggregation with dual-LP multipliers)
- Chvátal-Gomory rounding
- unsplitting of Assumptions

Certifacte Example

Given

$$x, y \in \mathbb{Z}$$

$C1:$	$2x_1 + 3x_2 \geq 1$		
$C2:$	$3x_1 - 4x_2 \leq 2$		
$C3:$	$-x_1 + 6x_2 \leq 3$		
Derived		**Reason**	**Assumptions**
$A1:$	$x_1 \leq 0$	$\{\text{assume}\}$	
$A2:$	$x_1 \geq 1$	$\{\text{assume}\}$	
$A3:$	$x_2 \leq 0$	$\{\text{assume}\}$	
$C4:$	$0 \geq 1$	$\{C1 + (-2) \times A1 + (-3) \times A3\}$	$A1, A3$
$A4:$	$x_2 \geq 1$	$\{\text{assume}\}$	
$C5:$	$0 \geq 1$	$\{\left(-\frac{1}{3}\right) \times C3 + \left(-\frac{1}{3}\right) \times A1 + 2 \times A4\}$	$A1, A4$
$C6:$	$x_2 \geq \frac{1}{4}$	$\{\left(-\frac{1}{4}\right) \times C2 + \left(\frac{3}{4}\right) \times A2\}$	$A2$
$C7:$	$x_2 \geq 1$	$\{\text{round up } C6\}$	$A2$
$C8:$	$0 \geq 1$	$\{\left(-\frac{1}{3}\right) \times C2 + (-1) \times C3 + \frac{14}{3} \times C7\}$	$A2$
$C9:$	$0 \geq 1$	$\{\text{unsplit } C4, C5 \text{ on } A3, A4\}$	$A1$
$C10:$	$0 \geq 1$	$\{\text{unsplit } C8, C9 \text{ on } A2, A1\}$	

Verified GMI cuts via mixed integer rounding

Challenge: Elementary verification by

- aggregations
- disjunctions
- Chvátal-Gomory rounding

as implemented in VIPR.

Easy in theory, tricky to implement:

- avoid relying on MIR formula

Verified GMI cuts via mixed integer rounding

Challenge: Elementary verification by

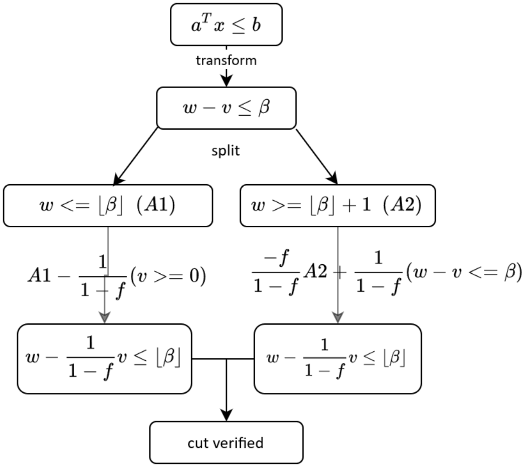
- aggregations
- disjunctions
- Chvátal-Gomory rounding

as implemented in VIPR.

Easy in theory, tricky to implement:

- avoid relying on MIR formula
- use interpretation as split cut
- post-process for “implicit” multipliers

$\rightsquigarrow$ solver-independent certificate file



Certified safe aggregation

The certificate checker will not accept the safe aggregation, since the printed constraint has coefficients $\overline{\alpha_i}$ (affected by the safe rounding routines)

Certified safe aggregation

The certificate checker will not accept the safe aggregation, since the printed constraint has coefficients $\overline{\alpha_i}$ (affected by the safe rounding routines)

Rectify this by computing and adding the rounding offset

$$
\begin{aligned}
&(\overline{\alpha_i} - \alpha_i)(x_i \leq u_i) && i \in U \\
&(\underline{\alpha_i} - \alpha_i)(x_i \geq u_i) && i \in L
\end{aligned}
$$

Implemented in an intermediate certificate-completion step to avoid extending the logic of the checker itself

Certified safe aggregation

The certificate checker will not accept the safe aggregation, since the printed constraint has coefficients $\overline{\alpha_i}$ (affected by the safe rounding routines)

Rectify this by computing and adding the rounding offset

$$
\begin{aligned}
&(\overline{\alpha_i} - \alpha_i)(x_i \leq u_i) && i \in U \\
&(\underline{\alpha_i} - \alpha_i)(x_i \geq u_i) && i \in L
\end{aligned}
$$

Implemented in an intermediate certificate-completion step to avoid extending the logic of the checker itself

Extra care has to be taken with slack variables!

Questions...? *Thank you for your attention!*

References:

William Cook, Sanjeeb Dash, Ricardo Fukasawa, and Marcos Goycoolea.
Numerically safe gomory mixed-integer cuts.
INFORMS Journal on Computing, 21(4):641–649, 2009.

William Cook, Thorsten Koch, Daniel E. Steffy, and Kati Wolter.
A hybrid branch-and-bound approach for exact rational mixed-integer programming.
Mathematical Programming Computation, 5(3):305–344, 2013.

Leon Eifler and Ambros Gleixner.
A computational status update for exact rational mixed integer programming.
In *Integer Programming and Combinatorial Optimization: 22nd International Conference, IPCO 2021, Proceedings*, 2021.