VIPR: Verifying Integer Programming Results

Ambros Gleixner, Zuse Institute Berlin

joint work with Kevin Cheung and Daniel Steffy

21st Combinatorial Optimization Workshop · Aussois · January 9, 2017

OUTLINE

1. Motivation and background
   - van der Waerden numbers
   - Frankl's conjecture
   - floating-point/exact integer programming

2. Verification of integer programming results
   - verification in SAT solving
   - tree-less branch-and-cut certificates
   - computational experiments

3. Conclusion

OUTLINE

1. Motivation and background
   - van der Waerden numbers
   - Frankl's conjecture
   - floating-point/exact integer programming

2. Verification of integer programming results
   - verification in SAT solving
   - tree-less branch-and-cut certificates
   - computational experiments

3. Conclusion

OUTLINE

1. Motivation and background
   - van der Waerden numbers
   - Frankl's conjecture
   - floating-point/exact integer programming

2. Verification of integer programming results
   - verification in SAT solving
   - tree-less branch-and-cut certificates
   - computational experiments

3. Conclusion

van der Waerden numbers

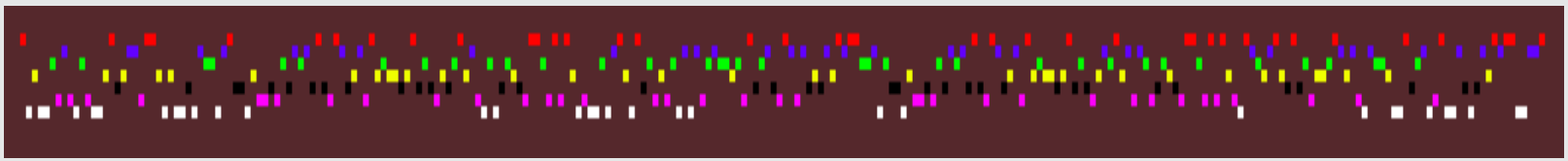
- W(r,k) := smallest M such that any r-coloring of $\{1, 2, \ldots M\}$ contains a monochromatic arithmetic progression of k integers
- 7 non-trivial numbers known, mostly by SAT solvers:
  lower bounds / show infeasibility (cannot color non-monochromatically)

van der Waerden numbers

- W(r,k) := smallest M such that any r-coloring of $\{1, 2, \ldots M\}$ contains a monochromatic arithmetic progression of k integers
- 7 non-trivial numbers known, mostly by SAT solvers:
  lower bounds / show infeasibility (cannot color non-monochromatically)
- Recent success using MIP solvers:

Theorem [Pulaj 2015]

$W(7, 3) \geqslant 258$.



Proof: 7-coloring of $\{1, 2, \ldots 257\}$ without monochromatic $\{m, m + \ell, m + 2\ell\}$.

Frankl's Conjecture (1979)
Any family of union-closed sets $\mathcal{F} \subseteq 2^{\{1,\dots,n\}}$ has an element $i \in \{1,\dots,n\}$ contained in at least half its sets.

Line of attack: Frankl-complete families $\mathcal{A} :\Leftrightarrow$ conjecture true for all $\mathcal{F} \supseteq \mathcal{A}$.

Frankl's Conjecture (1979)
Any family of union-closed sets $\mathcal{F} \subseteq 2^{\{1,\dots,n\}}$ has an element $i \in \{1,\dots,n\}$ contained in at least half its sets.

Line of attack: Frankl-complete families $\mathcal{A} :\Leftrightarrow$ conjecture true for all $\mathcal{F} \supseteq \mathcal{A}$.

Proposition [Pulaj 2016]

$\mathcal{A}$ is Frankl-complete if $\exists$ weights $c \in \mathbb{N}_0^n$, not all zero,

$$FC(\mathcal{A}, c)_n := \left\{ \begin{array}{ll} \displaystyle\sum_{S \in 2^{[n]}} \Big(\sum_{i \in S} c_i - \sum_{i \notin S} c_i \Big) x_S \leqslant -1 & \\ x_T + x_U \leqslant 1 + x_S & \forall\, T \cup U = S \in 2^{\{1,\dots,n\}} \\ x_T \leqslant x_U & \forall\, S \in \mathcal{A}, S \cup T = U \in 2^{\{1,\dots,n\}} \\ x_S \in \{0,1\} & \forall\, S \in 2^{\{1,\dots,n\}} \end{array} \right\} = \emptyset.$$

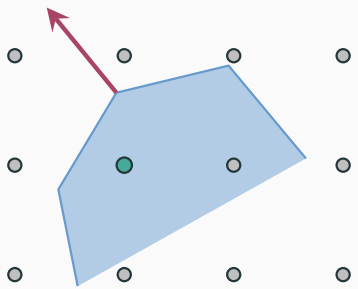
Proof: Rewrite Poonen's Theorem (1992).

Mixed-Integer Linear Programming

$$\text{minimize } c^T x \text{ subject to } Ax \leqslant b \text{ and } x_i \in \mathbb{Z} \text{ for some } i$$

LP-based branch-and-cut

- tree of dual bounds from LP solutions



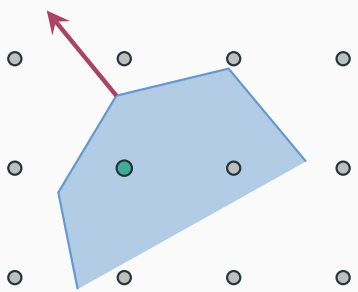
exact solution

Mixed-Integer Linear Programming

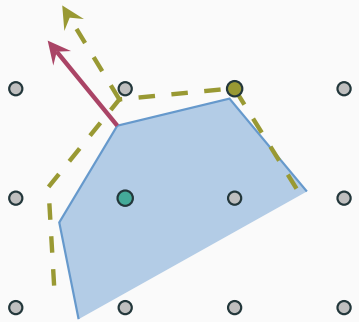
$$\text{minimize } c^T x \text{ subject to } Ax \leqslant b \text{ and } x_i \in \mathbb{Z} \text{ for some } i$$

LP-based branch-and-cut

- tree of dual bounds from LP solutions
- floating-point errors $\rightsquigarrow$ tolerances for feasibility and numeric comparisons



exact solution



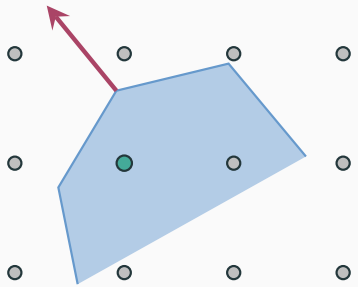
approximate solution

Mixed-Integer Linear Programming

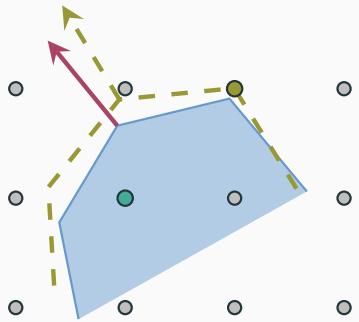
$$\text{minimize } c^T x \text{ subject to } Ax \leqslant b \text{ and } x_i \in \mathbb{Z} \text{ for some } i$$

LP-based branch-and-cut

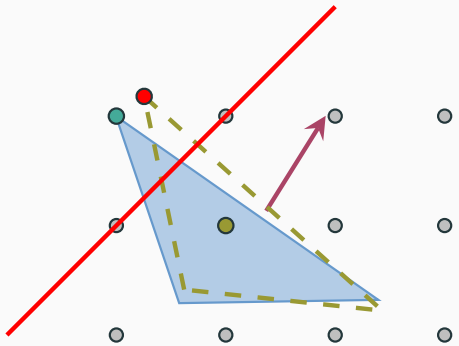
- tree of dual bounds from LP solutions
- floating-point errors $\rightsquigarrow$ tolerances for feasibility and numeric comparisons
- aggressive problem modification during the algorithm



exact solution



approximate solution



invalid model strengthening

Exact rational solvers exist

- LP: Qsopt_ex [Espinoza 2006; Applegate, Cook, Dash, Espinoza 2007]
- LP: SoPlex [G. 2015; G., Steffy, Wolter 2016]
- MIP: SCIP [Cook, Koch, Steffy, Wolter 2013]

Exact rational solvers exist

- LP: Qsopt_ex [Espinoza 2006; Applegate, Cook, Dash, Espinoza 2007]
- LP: SoPlex [G. 2015; G., Steffy, Wolter 2016]
- MIP: SCIP [Cook, Koch, Steffy, Wolter 2013]

But...

A Hybrid Branch-and-Bound Approach for Exact Rational Mixed-Integer Programming

William Cook · Thorsten Koch · Daniel E. Steffy · Kati Wolter

[2] Of course, even with a very carful implementation and extensive testing, a certain risk of an implementation error remains (also in the underlying exact LP solver and the software package for rational arithmetic). So, the exact objective values reported here come with no warranty.

Math. Prog. Comput., 5(3):305–344, 2013

OUTLINE

1. Motivation and background
   - van der Waerden numbers
   - Frankl's conjecture
   - floating-point/exact integer programming

2. Verification of integer programming results
   - verification in SAT solving
   - tree-less branch-and-cut certificates
   - computational experiments

3. Conclusion

Verification in UNSAT

- many SAT solvers produce a trace: certificate file for infeasible problems
- DRAT-trim: formally verified trace checker [Welzer, Heule, Hunt 2014]
- May 2016: 200 TB proof for the Pythagorean Triples problem [Heule, Kullmann, Marek 2016]

Verification in UNSAT

- many SAT solvers produce a trace: certificate file for infeasible problems
- DRAT-trim: formally verified trace checker [Welzer, Heule, Hunt 2014]
- May 2016: 200 TB proof for the Pythagorean Triples problem [Heule, Kullmann, Marek 2016]

Verification in MIP

- MIP solvers do not output an optimality certificate

Verification in UNSAT

- many SAT solvers produce a trace: certificate file for infeasible problems
- DRAT-trim: formally verified trace checker [Welzer, Heule, Hunt 2014]
- May 2016: 200 TB proof for the Pythagorean Triples problem [Heule, Kullmann, Marek 2016]

Verification in MIP

- MIP solvers do not output an optimality certificate, but:
- verification of a TSP tour through 85,900 cities [Applegate et al. 2009]
- formal verification of LP-based proof of Kepler's Conjecture [Flyspeck: Obua 2008, Hales et al. 2015]

Difficulties

- MIP solvers use complex and diverse algorithms
- formal verification involving rational arithmetic is prohibitively slow
- certificate form unclear: tree, cuts, superadditive function, ...

Difficulties

- MIP solvers use complex and diverse algorithms
- formal verification involving rational arithmetic is prohibitively slow
- certificate form unclear: tree, cuts, superadditive function, ...

Goals of our certificate format

- *expressivity*: encode all (or most important) MIP techniques
- *simplicity*: allow verification by short checker code

LP optimality can be certified by a dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
`obj`$:$	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

LP optimality can be certified by a dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
`obj`$:$	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
```

LP optimality can be certified by a dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
obj :	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
```

LP optimality can be certified by a dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
obj :	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 inf
```

LP optimality can be certified by a dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
obj :	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2
  x y
OBJ min
  2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 inf
DER 1
  C2  G 1  2  0 2  1 1  { lin 2  0 1  1 -1 }
```

LP optimality can be certified by a primal-dual solution

- e.g.

$$\begin{aligned} \min \quad & 2x + y \\ \text{s.t.} \quad & \\ C0: \quad & 5x - y \geqslant 2 \\ C1: \quad & 3x - 2y \leqslant 1 \end{aligned}$$

Given		
$C0:$	$5x - y \geqslant 2$	
$C1:$	$3x - 2y \leqslant 1$	
Derived		**Reason**
obj :	$2x + y \geqslant 1$	$\{1 \times C0 + (-1) \times C1\}$

- plain text syntax:

```
VAR 2 x y
OBJ min 2  0 2  1 1
CON 2 0
  C0  G 2  2  0 5  1 -1
  C1  L 2  2  0 3  1 -2
RTP range 1 1
SOL 1
  2  0 3/7  1 1/7
DER 1
  C2  G 1  2  0 2  1 1  { lin 2  0 1  1 -1 }
```

Many cutting planes require a rounding argument

- e.g.

$$
\begin{aligned}
\min \quad & x + y \\
\text{s.t.} \quad & \\
C0: \quad & 4x + y \geqslant 1 \\
C1: \quad & 4x - y \leqslant 2 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

Given		
	$x, y \in \mathbb{Z}$	
$C0:$	$4x + y \geqslant 1$	
$C1:$	$4x - y \leqslant 2$	
Derived		**Reason**
$C2:$	$y \geqslant -\frac{1}{2}$	$\{\frac{1}{2} \times C0 + (-\frac{1}{2}) \times C1\}$
$C3:$	$y \geqslant 0$	{round up $C2$}

Many cutting planes require a rounding argument

- e.g.

$$
\begin{aligned}
\min \quad & x + y \\
\text{s.t.} \quad & \\
C0: \quad & 4x + y \geqslant 1 \\
C1: \quad & 4x - y \leqslant 2 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

Given		
	$x, y \in \mathbb{Z}$	
$C0:$	$4x + y \geqslant 1$	
$C1:$	$4x - y \leqslant 2$	
Derived		**Reason**
$C2:$	$y \geqslant -\frac{1}{2}$	$\left\{\frac{1}{2} \times C0 + \left(-\frac{1}{2}\right) \times C1\right\}$
$C3:$	$y \geqslant 0$	{round up $C2$}
$C4:$	$x + y \geqslant \frac{1}{4}$	$\left\{\frac{1}{4} \times C0 + \frac{3}{4} \times C3\right\}$
$C5:$	$x + y \geqslant 1$	{round up $C4$}

Many cutting planes require a rounding argument

- e.g.

$$
\begin{array}{rl}
\min & x + y \\
\text{s.t.} & \\
C0: & 4x + y \geqslant 1 \\
C1: & 4x - y \leqslant 2 \\
 & x, y \in \mathbb{Z}
\end{array}
$$

Given		
	$x, y \in \mathbb{Z}$	
$C0:$	$4x + y \geqslant 1$	
$C1:$	$4x - y \leqslant 2$	
Derived		**Reason**
$C2:$	$y \geqslant -\frac{1}{2}$	$\{\frac{1}{2} \times C0 + (-\frac{1}{2}) \times C1\}$
$C3:$	$y \geqslant 0$	{round up $C2$}
$C4:$	$x + y \geqslant \frac{1}{4}$	$\{\frac{1}{4} \times C0 + \frac{3}{4} \times C3\}$
$C5:$	$x + y \geqslant 1$	{round up $C4$}

- plain text syntax:

```
…
DER 4
  C2  G -1/2  1  1 1       { lin 2  0 1/2  1 -1/2 }
  C3  G 0  1  1 1          { rnd 2 }
  C4  G 1/4   2  0 1  1 1  { lin 2  0 1/4  3 3/4 }
  C5  G 1     2  0 1  1 1  { rnd 4 }
```

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	{assume}	
$A1:$	$x_2 \leqslant 0$	{assume}	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	$\{\text{assume}\}$	
$A1:$	$x_2 \leqslant 0$	$\{\text{assume}\}$	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$
$A2:$	$x_2 \geqslant 1$	$\{\text{assume}\}$	
$C4:$	$0 \geqslant 1$	$\left\{\left(-\frac{1}{3}\right) \times C2 + \left(-\frac{1}{3}\right) \times A0 + 2 \times A2\right\}$	$A0, A2$

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	{assume}	
$A1:$	$x_2 \leqslant 0$	{assume}	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$
$A2:$	$x_2 \geqslant 1$	{assume}	
$C4:$	$0 \geqslant 1$	$\left\{\left(-\frac{1}{3}\right) \times C2 + \left(-\frac{1}{3}\right) \times A0 + 2 \times A2\right\}$	$A0, A2$
$A3:$	$x_1 \geqslant 1$	{assume}	
$C5:$	$x_2 \geqslant \frac{1}{4}$	$\left\{\left(-\frac{1}{4}\right) \times C1 + \left(\frac{3}{4}\right) \times A3\right\}$	$A3$
$C6:$	$x_2 \geqslant 1$	{round up $C5$}	$A3$

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	{assume}	
$A1:$	$x_2 \leqslant 0$	{assume}	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$
$A2:$	$x_2 \geqslant 1$	{assume}	
$C4:$	$0 \geqslant 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A2\}$	$A0, A2$
$A3:$	$x_1 \geqslant 1$	{assume}	
$C5:$	$x_2 \geqslant \frac{1}{4}$	$\{(-\frac{1}{4}) \times C1 + (\frac{3}{4}) \times A3\}$	$A3$
$C6:$	$x_2 \geqslant 1$	{round up $C5$}	$A3$
$C7:$	$0 \geqslant 1$	$\{(-\frac{1}{3}) \times C1 + (-1) \times C2 + \frac{14}{3} \times C6\}$	$A3$

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	{assume}	
$A1:$	$x_2 \leqslant 0$	{assume}	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$
$A2:$	$x_2 \geqslant 1$	{assume}	
$C4:$	$0 \geqslant 1$	$\left\{\left(-\frac{1}{3}\right) \times C2 + \left(-\frac{1}{3}\right) \times A0 + 2 \times A2\right\}$	$A0, A2$
$A3:$	$x_1 \geqslant 1$	{assume}	
$C5:$	$x_2 \geqslant \frac{1}{4}$	$\left\{\left(-\frac{1}{4}\right) \times C1 + \left(\frac{3}{4}\right) \times A3\right\}$	$A3$
$C6:$	$x_2 \geqslant 1$	{round up $C5$}	$A3$
$C7:$	$0 \geqslant 1$	$\left\{\left(-\frac{1}{3}\right) \times C1 + (-1) \times C2 + \frac{14}{3} \times C6\right\}$	$A3$
$C8:$	$0 \geqslant 1$	{unsplit $C3, C4$ on $A1, A2$}	$A0$

A tree-less branch-and-cut certificate

Given			
	$x, y \in \mathbb{Z}$		
$C0:$	$2x_1 + 3x_2 \geqslant 1$		
$C1:$	$3x_1 - 4x_2 \leqslant 2$		
$C2:$	$-x_1 + 6x_2 \leqslant 3$		
Derived		**Reason**	**Assumptions**
$A0:$	$x_1 \leqslant 0$	{assume}	
$A1:$	$x_2 \leqslant 0$	{assume}	
$C3:$	$0 \geqslant 1$	$\{C0 + (-2) \times A0 + (-3) \times A1\}$	$A0, A1$
$A2:$	$x_2 \geqslant 1$	{assume}	
$C4:$	$0 \geqslant 1$	$\{(-\frac{1}{3}) \times C2 + (-\frac{1}{3}) \times A0 + 2 \times A2\}$	$A0, A2$
$A3:$	$x_1 \geqslant 1$	{assume}	
$C5:$	$x_2 \geqslant \frac{1}{4}$	$\{(-\frac{1}{4}) \times C1 + (\frac{3}{4}) \times A3\}$	$A3$
$C6:$	$x_2 \geqslant 1$	{round up $C5$}	$A3$
$C7:$	$0 \geqslant 1$	$\{(-\frac{1}{3}) \times C1 + (-1) \times C2 + \frac{14}{3} \times C6\}$	$A3$
$C8:$	$0 \geqslant 1$	{unsplit $C3, C4$ on $A1, A2$}	$A0$
$C9:$	$0 \geqslant 1$	{unsplit $C7, C8$ on $A3, A0$}	

Simplicity

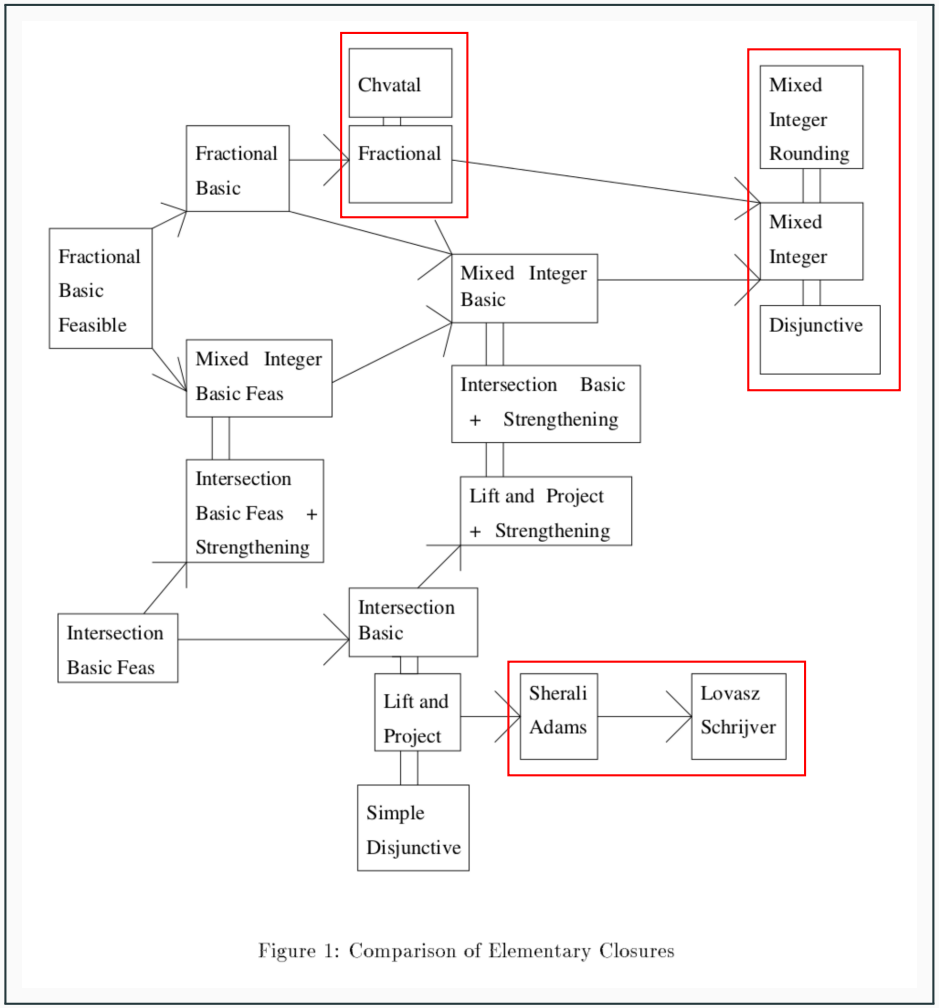
- only 4 reason types:
  `lin, rnd, asm, uns`
- no explicit tree structure
- allows for sequential checking

Simplicity

- only 4 reason types: `lin, rnd, asm, uns`
- no explicit tree structure
- allows for sequential checking

Expressivity

- general split cuts



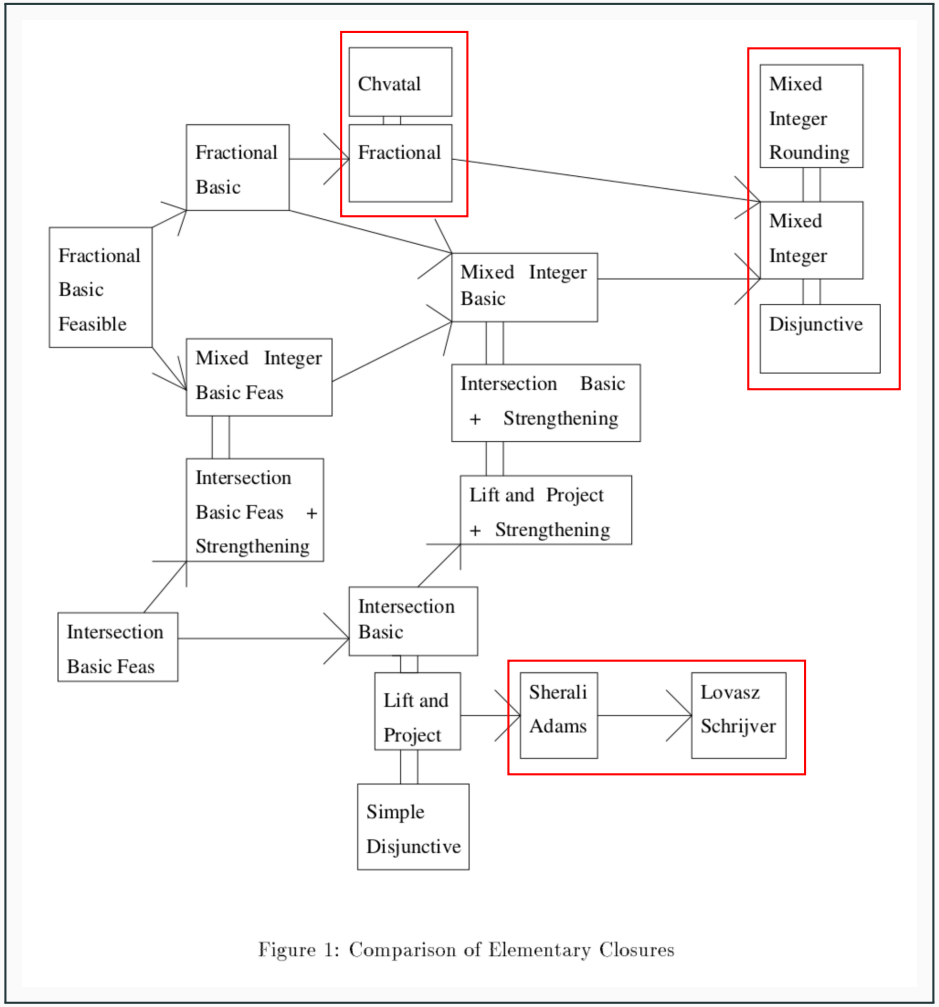
Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

Simplicity

- only 4 reason types: `lin, rnd, asm, uns`
- no explicit tree structure
- allows for sequential checking

Expressivity

- general split cuts
- presolving reductions, e.g., constraint propagation, reduced cost fixing
- global solver structures, e.g., clique graph, implication graph, variable bound graph



Cornuejols, Li. Elementary closures for integer programs. *Oper. Res. Letts*, 28:1–8, 2001

VIPR: Verifying Integer Programming Results

- C++ code to check, compress, and display `.vipr` certificates
- freely available at `http://github.com/ambros-gleixner/VIPR`

VIPR: Verifying Integer Programming Results

- C++ code to check, compress, and display `.vipr` certificates
- freely available at `http://github.com/ambros-gleixner/VIPR`

Certificates for 106 MIPs from Cook et al. 2013 (exact SCIP, 1 hour time limit)

		SCIP		VIPR				
Test set	N	N_{sol}	t_{MIP}	t_{ttn}	t_{chk}	size_{raw}	size_{ttn}	size_{gz}
easy-all	56	39	180.8	25.8	28.9	214	72	22
-solved	39	39	48.0	9.6	13.4	77	34	10
-memout	5	0	1769.4	377.5	169.8	10286	513	159
-timeout	12	0	—	83.7	97.5	1151	368	108
hard-all	50	14	976.6	31.2	15.1	372	38	11
-solved	13	13	40.8	7.1	6.3	49	15	5
-memout	10	0	1833.9	275.7	53.8	10269	146	39
-timeout	27	1	—	20.7	11.9	286	35	9

More details in Cheung, G., Steffy, *Verifying Integer Programming Results*, ZIB-Report 16-58, 2016.

OUTLINE

1. Motivation and background
   - van der Waerden numbers
   - Frankl's conjecture
   - floating-point/exact integer programming

2. Verification of integer programming results
   - verification in SAT solving
   - tree-less branch-and-cut certificates
   - computational experiments

3. Conclusion

Summary

- MIP for mathematical proofs: not the only application!
- VIPR: simple and expressive certificate format + checker
- exact SCIP can now produce `.vipr` certificates

Summary

- MIP for mathematical proofs: not the only application!
- VIPR: simple and expressive certificate format + checker
- exact SCIP can now produce `.vipr` certificates

Questions

- floating-point certificates?
- trade-off: sophistication of MIP techniques vs. ease of certification?
- proof compression?
- formal verification?
- verified verification code?