Deriving intersection cuts from wide split disjunctions

Sven Wiese [1]

with Pierre Bonami [2], Andrea Lodi [1]
and Andrea Tramontani [2]

[1]DEI, University of Bologna

[2]CPLEX Optimization

Bologna, 30/04/2015

Deriving intersection cuts from wide split disjunctions

Sven Wiese [1]

with Pierre Bonami [2], Andrea Lodi [1]
and Andrea Tramontani [2]

[1]DEI, University of Bologna

[2]CPLEX Optimization

Bologna, 30/04/2015

@OR_Unibo

A motivating example...

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$D(x_{1,1}) = \{1,2,3,4,5,6,7,8,9\}$

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$D(x_{1,1}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

`mixed integer programming:`

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{1,1}) = \{1,2,3,4,5,6,7,8,9\}$$

`mixed integer programming:`

$$1 \leq x_{1,1} \leq 9$$

$$x_{1,1} \in \mathbb{Z}$$

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{1,1}) = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5, 6, \cancel{7}, \cancel{8}, \cancel{9}\}$$

`mixed integer programming:`

$$1 \leq x_{1,1} \leq 9$$

$$x_{1,1} \in \mathbb{Z}$$

A motivating example...

$x_{1,1}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

constraint programming:

$$D(x_{1,1}) = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5, 6, \cancel{7}, \cancel{8}, \cancel{9}\}$$

mixed integer programming:

$$4 \leq x_{1,1} \leq 6$$

$$x_{1,1} \in \mathbb{Z}$$

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

`mixed integer programming:`

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

`mixed integer programming:`

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

`mixed integer programming:`

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{5,5}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

`mixed integer programming:`

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{5,5}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

`mixed integer programming:`

$$1 \leq x_{5,5} \leq 9$$

$$x_{5,5} \in \mathbb{Z}$$

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

`mixed integer programming:`

$$1 \leq x_{5,5} \leq 9$$

$$x_{5,5} \in \mathbb{Z}$$

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$

`mixed integer programming:`

$1 \leq x_{5,5} \leq 8$

$x_{5,5} \in \mathbb{Z}$

A motivating example...

$x_{5,5}$

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3				$x_{5,5}$				4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

constraint programming:

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

mixed integer programming:

$$1 \leq x_{5,5} \leq 8$$

$$x_{5,5} \in \mathbb{Z}$$

$$x_{5,5} \neq 3,\ x_{5,5} \neq 4,\ x_{5,5} \neq 6,\ x_{5,5} \neq 7$$

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
		8	6		3	5		
3								4
		6	7		9	2		
9								2
	8		9		1		6	
	1		4	3	6		5	

`constraint programming:`

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

`mixed integer programming:`

$$1 \leq x_{5,5} \leq 8$$

$$x_{5,5} \in \mathbb{Z}$$

$$x_{5,5} \neq 3,\ x_{5,5} \neq 4,\ x_{5,5} \neq 6,\ x_{5,5} \neq 7$$

A motivating example...

	2		1	7	8		3	
			3		2		9	
1								6
2	7	8	6		3	5		
3				8				4
		6	7		9	2		
9						1		2
	8	4	9		1		6	
	1		4	3	6		5	

constraint programming:

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

mixed integer programming:

$$1 \leq x_{5,5} \leq 8$$

$$x_{5,5} \in \mathbb{Z}$$

$$x_{5,5} \neq 3,\ x_{5,5} \neq 4,\ x_{5,5} \neq 6,\ x_{5,5} \neq 7$$

A motivating example...

	2		1	7	8	4	3	5
			3		2		9	
1			5					6
2	7	8	6	4	3	5		
3		1		8		6	7	4
		6	7		9	2		
9					7	1		2
	8	4	9		1		6	
	1	2	4	3	6	9	5	

constraint programming:

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

mixed integer programming:

$$1 \leq x_{5,5} \leq 8$$

$$x_{5,5} \in \mathbb{Z}$$

$$x_{5,5} \neq 3, x_{5,5} \neq 4, x_{5,5} \neq 6, x_{5,5} \neq 7$$

A motivating example...

6	2	9	1	7	8	4	3	5
8	4	5	3	6	2	7	9	9
1	3	7	5	9	4	8	2	6
2	7	8	6	4	3	5	1	9
3	9	1	2	8	5	6	7	4
4	5	6	7	1	9	2	8	3
9	6	3	8	5	7	1	4	2
5	8	4	9	2	1	3	6	7
7	1	2	4	3	6	9	5	8

`constraint programming:`

$$D(x_{5,5}) = \{1, 2, \cancel{3}, \cancel{4}, 5, \cancel{6}, \cancel{7}, 8, \cancel{9}\}$$

`mixed integer programming:`

$$1 \leq x_{5,5} \leq 8$$

$$x_{5,5} \in \mathbb{Z}$$

$$x_{5,5} \neq 3, x_{5,5} \neq 4, x_{5,5} \neq 6, x_{5,5} \neq 7$$

Motivation

Motivation

In Constraint Programming, exploiting “holes” in the domains of variables is routinely done, e.g., through filtering and propagation.

Motivation

In Constraint Programming, exploiting “holes” in the domains of variables is routinely done, e.g., through filtering and propagation.

In Mixed Integer Linear Programming, holes in the domain of integer variables can be exploited for example via branching.

Motivation

In Constraint Programming, exploiting “holes” in the domains of variables is routinely done, e.g., through filtering and propagation.

In Mixed Integer Linear Programming, holes in the domain of integer variables can be exploited for example via branching.

Instead, we generate cutting planes from the disjunctions associated with the holes in the integer variable domains. We call these disjunctions wide split disjunctions.

Motivation

In Constraint Programming, exploiting "holes" in the domains of variables is routinely done, e.g., through filtering and propagation.

In Mixed Integer Linear Programming, holes in the domain of integer variables can be exploited for example via branching.

Instead, we generate cutting planes from the disjunctions associated with the holes in the integer variable domains. We call these disjunctions wide split disjunctions.

Such an approach can as well be extended to continuous variables. In this talk, we restrict to integer variables.

Outline

Outline

Outline

Outline

Outline

Outline

Variables with non-contiguous domains

Variables with non-contiguous domains

In this talk we consider a Mixed Integer Linear Program (MILP) in the general form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n, \end{aligned} \tag{M}$$

where $A \in \mathbb{R}^{m \times (p+n)}$, $b \in \mathbb{R}^m$.

Variables with non-contiguous domains

In this talk we consider a Mixed Integer Linear Program (MILP) in the general form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \qquad\qquad \text{(M)} \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n, \end{aligned}$$

where $A \in \mathbb{R}^{m \times (p+n)}$, $b \in \mathbb{R}^m$.

Furthermore, we assume that there are integer variables that have non-contiguous domains, i.e., for a given variable x_i with domain $D_i = [L_i, U_i]$, there is at least one $t \in \mathbb{Z} \cap (L_i, U_i)$ such that x_i cannot be equal to t.

Variables with non-contiguous domains

In this talk we consider a Mixed Integer Linear Program (MILP) in the general form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n, \end{aligned} \tag{M}$$

where $A \in \mathbb{R}^{m \times (p+n)}$, $b \in \mathbb{R}^m$.

Furthermore, we assume that there are integer variables that have non-contiguous domains, i.e., for a given variable x_i with domain $D_i = [L_i, U_i]$, there is at least one $t \in \mathbb{Z} \cap (L_i, U_i)$ such that x_i cannot be equal to t.

In other words, the variable has a "hole" in its domain.

Variables with non-contiguous domains

Variables with non-contiguous domains

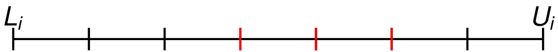
Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).

Variables with non-contiguous domains

Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).

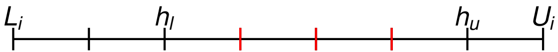
Variables with non-contiguous domains

Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).



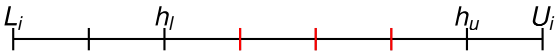
Variables with non-contiguous domains

Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).



Variables with non-contiguous domains

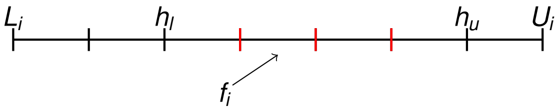
Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).



We will separate intersection cuts from the optimal simplex tableaus of the LP-relaxation of (M), provided that the fractional solution f lies in some hole.

Variables with non-contiguous domains

Formally, we say that $H = [h_l, h_u]$ with $(h_l, h_u) \in \mathbb{Z}^2$ is a hole for the integer variable x_i, if $H \subseteq D_i$ and if $\forall\ t \in \mathbb{Z} \cap \text{int}(H)$, we have that $x_i \neq t$ in any feasible (or optimal) solution of (M).



We will separate intersection cuts from the optimal simplex tableaus of the LP-relaxation of (M), provided that the fractional solution f lies in some hole.

How can holes come around?

How can holes come around?

Holes can be present in a mixed integer model explicitly, e.g., thorugh GUB-link type equalities

How can holes come around?

Holes can be present in a mixed integer model explicitly, e.g., thorugh GUB-link type equalities

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, p\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$.

How can holes come around?

Holes can be present in a mixed integer model explicitly, e.g., thorugh GUB-link type equalities

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, p\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$.

If the $\lambda_j \in \mathbb{N}$ are non-consecutive, there is a hole in the domain of the integer variable x_i.

How can holes come around?

How can holes come around?

The condition $x_i \in \bigcup_{j=1}^{q} [l_j, u_j]$ with $u_j + 1 < l_{j+1} \ \forall \ j = 1, \ldots, q$ can be modeled through bigM constraints:

How can holes come around?

The condition $x_i \in \bigcup_{j=1}^{q} [l_j, u_j]$ with $u_j + 1 < l_{j+1} \;\; \forall \; j = 1, \ldots, q$ can be modeled through bigM constraints:

$$\begin{aligned} l_j - x_j &\leq (1 - x_j) \cdot (l_j - l_1) \\ x_j - u_j &\leq (1 - x_j) \cdot (u_q - u_j) \qquad \forall \; j = 1, \ldots, q \\ \sum_{j=1}^{q} x_j &= 1. \end{aligned}$$

How can holes come around?

The condition $x_i \in \bigcup_{j=1}^{q} [l_j, u_j]$ with $u_j + 1 < l_{j+1} \ \forall\ j = 1, \ldots, q$ can be modeled through bigM constraints:

$$
\begin{aligned}
l_j - x_j &\leq (1 - x_j) \cdot (l_j - l_1) \\
x_j - u_j &\leq (1 - x_j) \cdot (u_q - u_j) \qquad \forall\ j = 1, \ldots, q \\
\sum_{j=1}^{q} x_j &= 1 .
\end{aligned}
$$

This constraint structure can be found, e.g., in straightforward models for the well-known Traveling Salesman Problem with Multiple Time Windows (TSPMTW).

How can holes come around?

How can holes come around?

It can also happen that holes are implicit.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i,

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

How can holes come around?

It can also happen that holes are implicit.

Consider for example the Traveling Salesman Problem with Time Windows (TSPTW) and its variant described in [Dash et al. 2012].

In this model, we have start variables s_i that describe the instant in which the tour leaves a certain node i, and we have to respect $s_i \in W_i = [R_i, D_i]$.

If for a $t > R_i$ there is no node k such that $t = t' + travel\_time_{ki}$ for some $t' \in W_k$, then $s_i \neq t$ in an optimal solution.

In such a situation, the time window of node i could be strengthened to $\{R_i\} \cup [K_i, D_i]$ for some $K_i > R_i + 1$.

Outline

Intersection cuts [Balas 1970]

Intersection cuts [Balas 1970]

Consider the optimal simplex tableau of the LP relaxation of (M), let the basic variables be indexed in B and the non-basic ones in N:

Intersection cuts [Balas 1970]

Consider the optimal simplex tableau of the LP relaxation of (M), let the basic variables be indexed in B and the non-basic ones in N:

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$
$$x_j \geq 0 \quad j \in N \cup B$$

Intersection cuts [Balas 1970]

Consider the optimal simplex tableau of the LP relaxation of (M), let the basic variables be indexed in B and the non-basic ones in N:

$$\begin{aligned} x_i &= f_i + \sum_{j \in N} r_i^j x_j \quad && i \in B \\ x_j &\geq 0 && j \in N \end{aligned}$$

Intersection cuts [Balas 1970]

Consider the optimal simplex tableau of the LP relaxation of (M), let the basic variables be indexed in B and the non-basic ones in N:

$$\begin{aligned} x_i &= f_i + \sum_{j \in N} r_i^j x_j \qquad && i \in B \\ x_j &\geq 0 && j \in N \end{aligned}$$

The above system defines a polyhedron $P(B)$.

Intersection cuts [Balas 1970]

Consider the optimal simplex tableau of the LP relaxation of (M), let the basic variables be indexed in B and the non-basic ones in N:

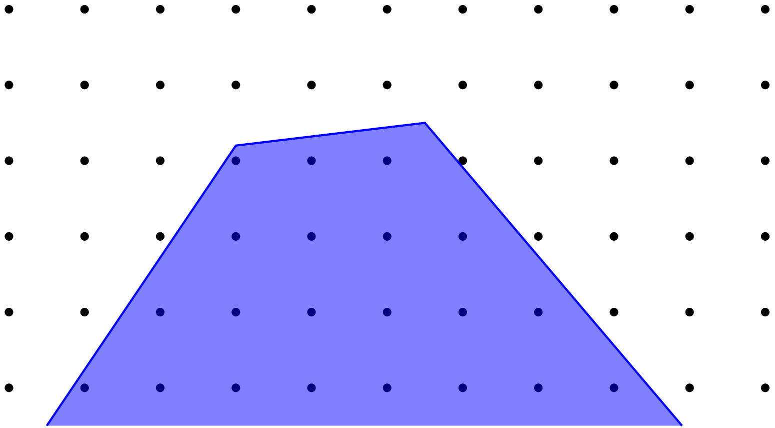
$$
\begin{aligned}
x_i &= f_i + \sum_{j \in N} r_i^j x_j && i \in B \\
x_j &\geq 0 && j \in N
\end{aligned}
$$

The above system defines a polyhedron $P(B)$.

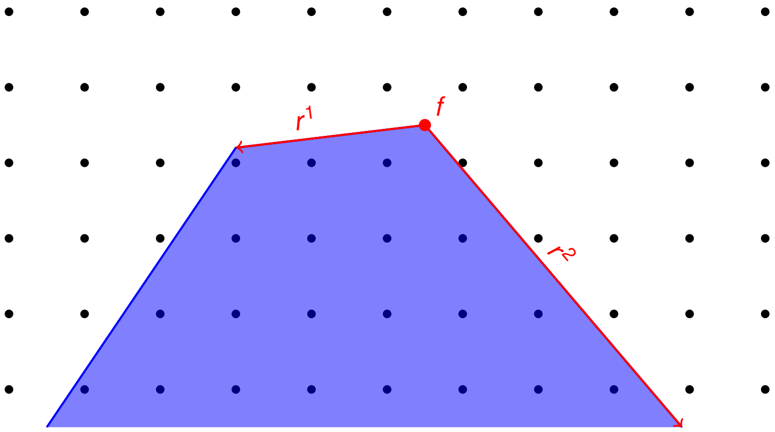
To derive a cut, take a convex lattice-free set S that has the basic solution f in its interior and compute the interesction of the extreme rays of $P(B)$ with the boundary of S.

Intersection cuts [Balas 1970]

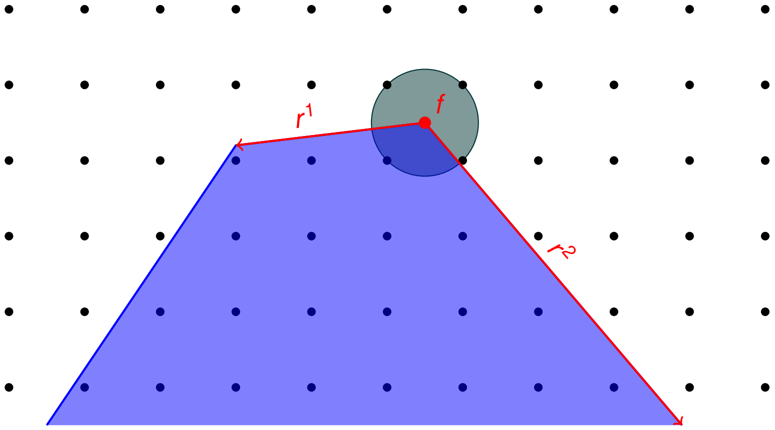
Intersection cuts [Balas 1970]



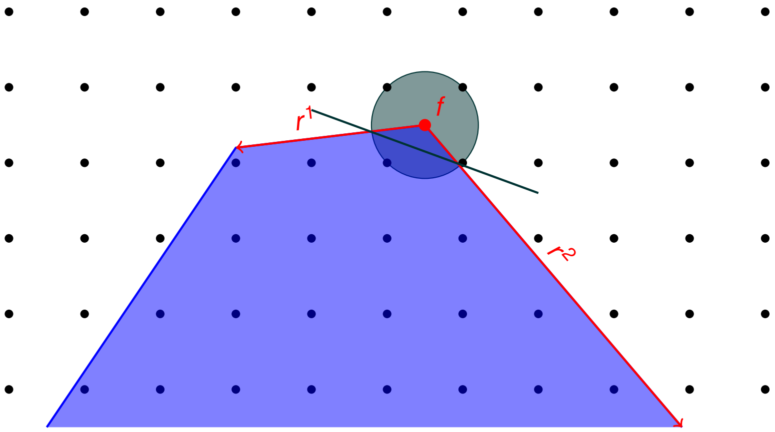
Intersection cuts [Balas 1970]



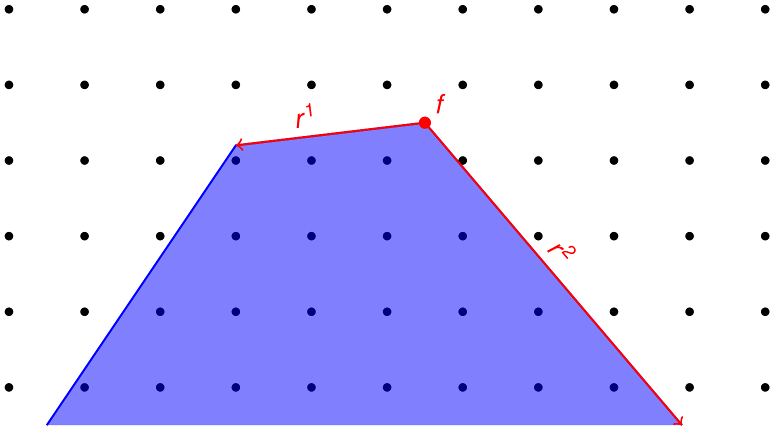
Intersection cuts [Balas 1970]



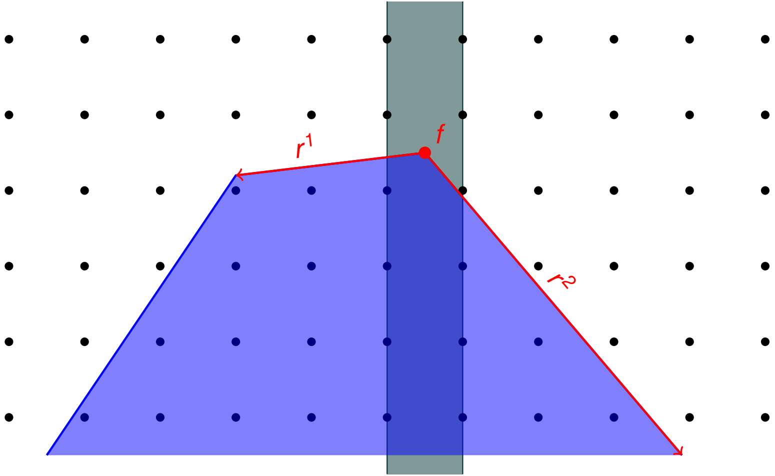
Intersection cuts [Balas 1970]



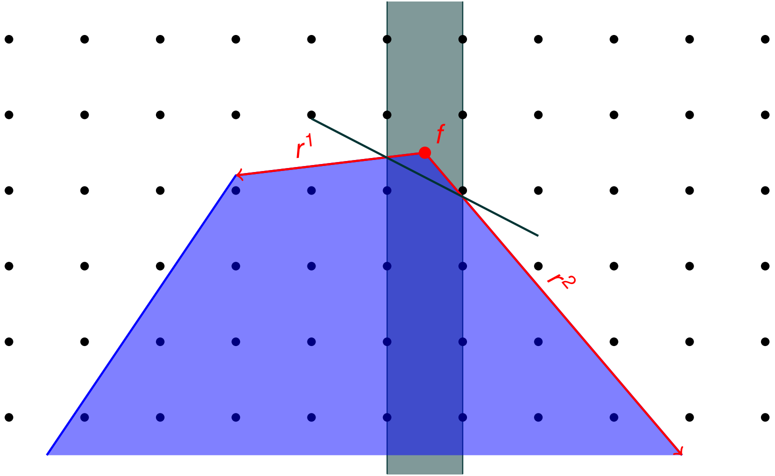
Intersection cuts [Balas 1970]



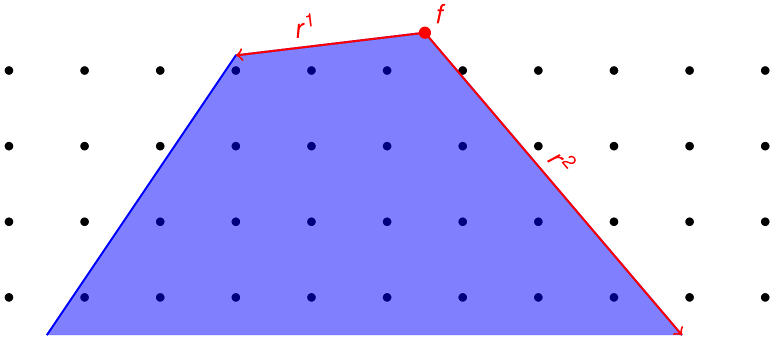
Intersection cuts [Balas 1970]



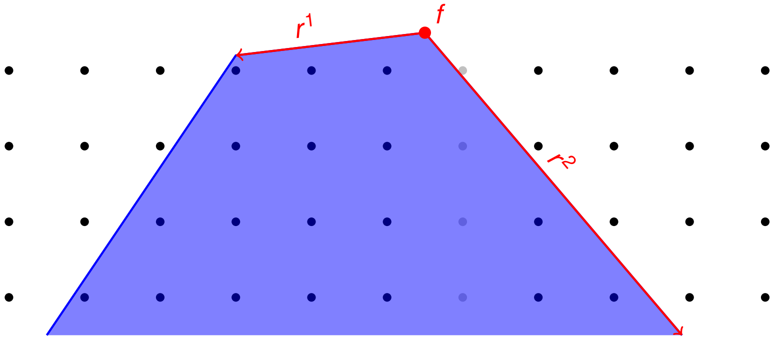
Intersection cuts [Balas 1970]



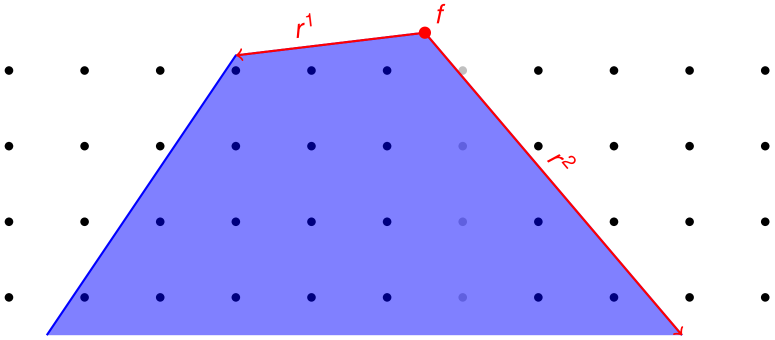
Intersection cuts [Balas 1970]



Intersection cuts [Balas 1970]

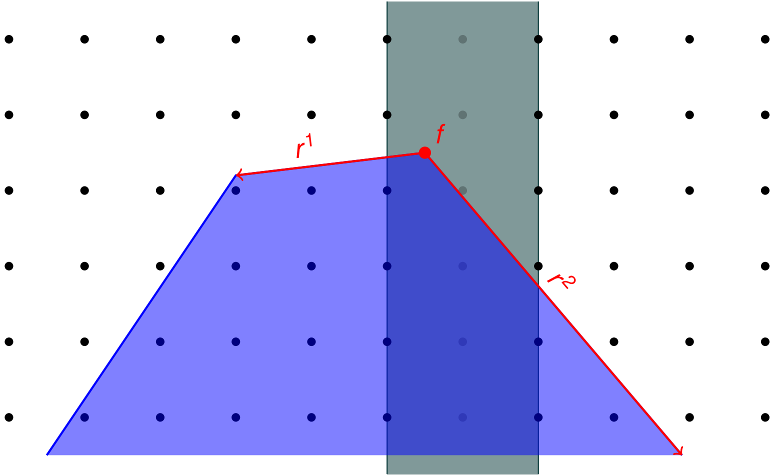


Intersection cuts [Balas 1970]



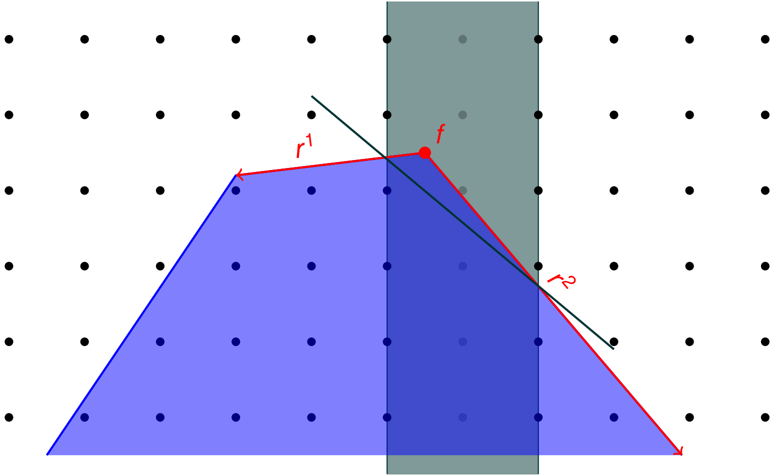
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



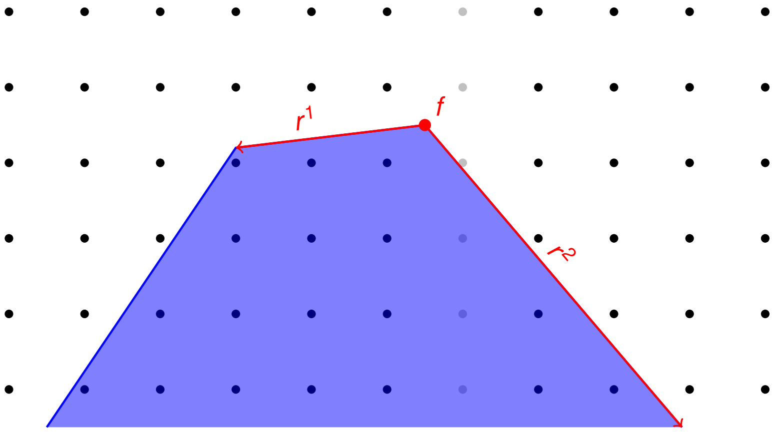
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



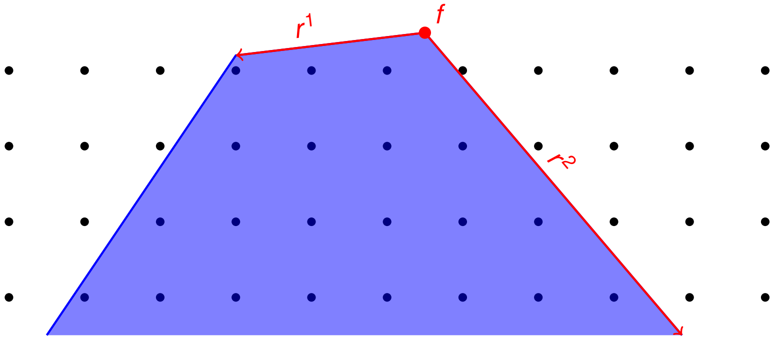
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



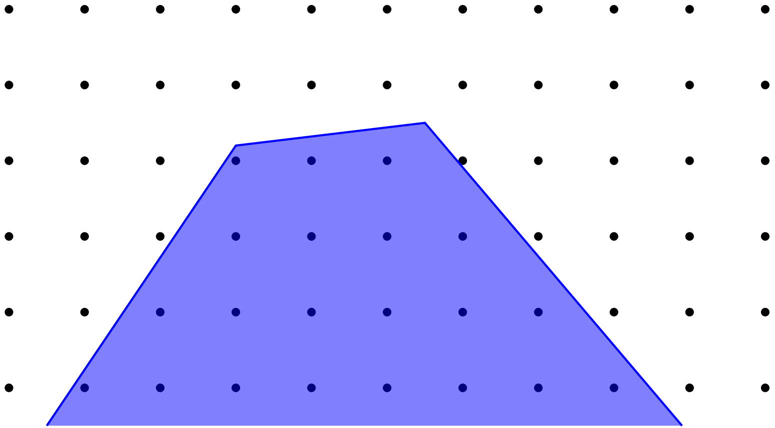
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



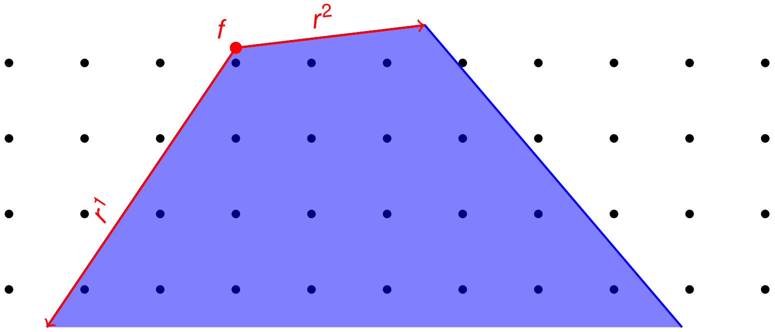
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



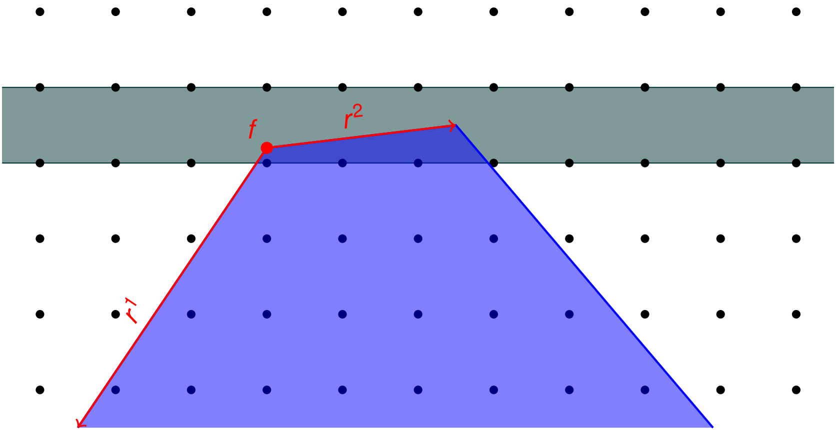
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



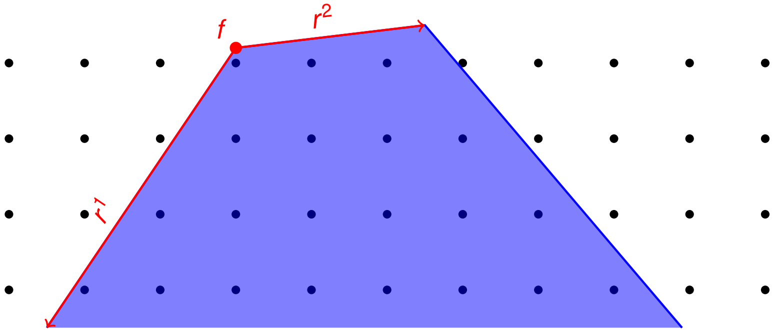
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



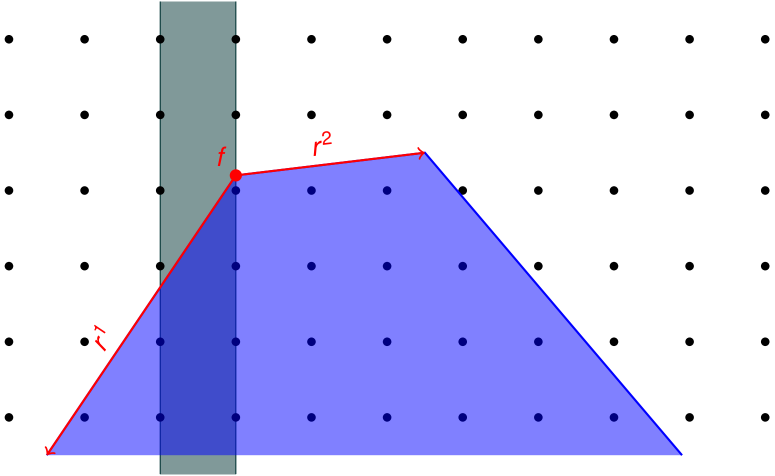
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



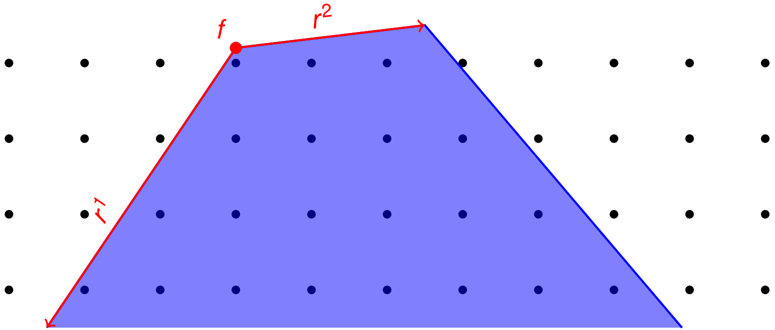
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



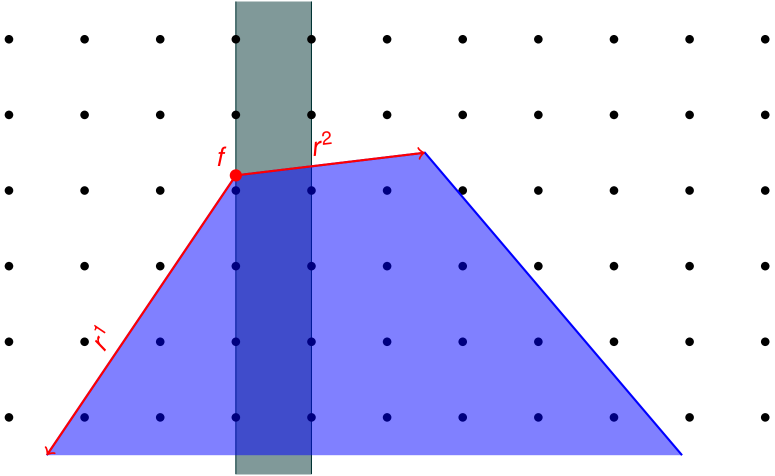
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



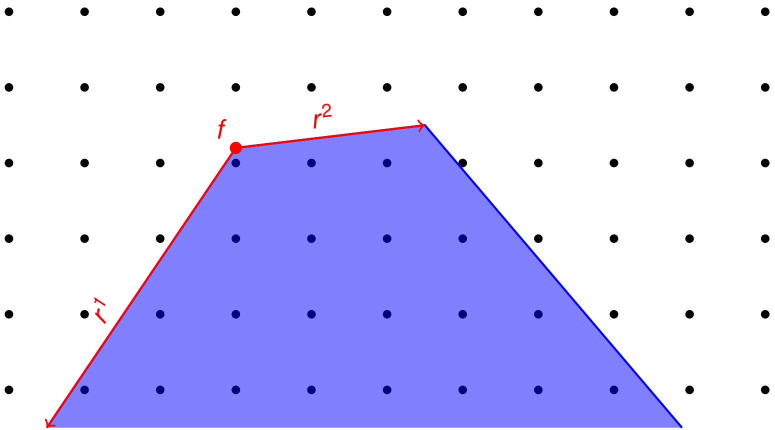
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



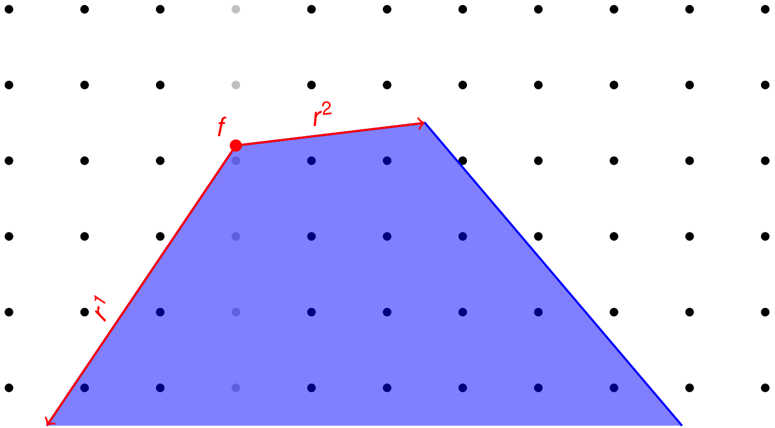
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



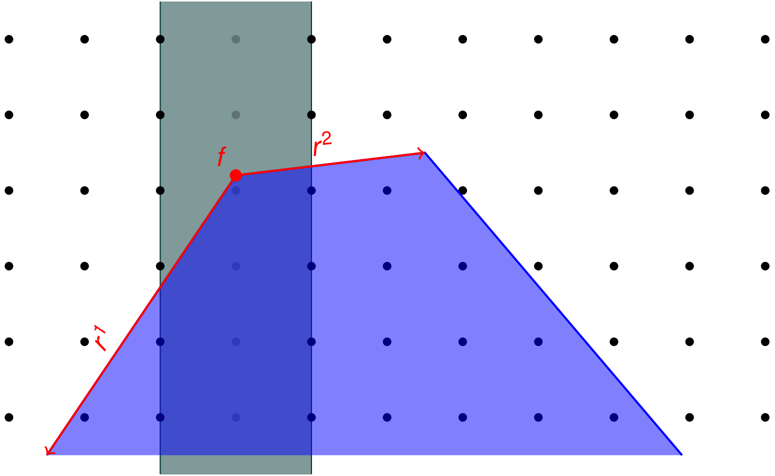
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



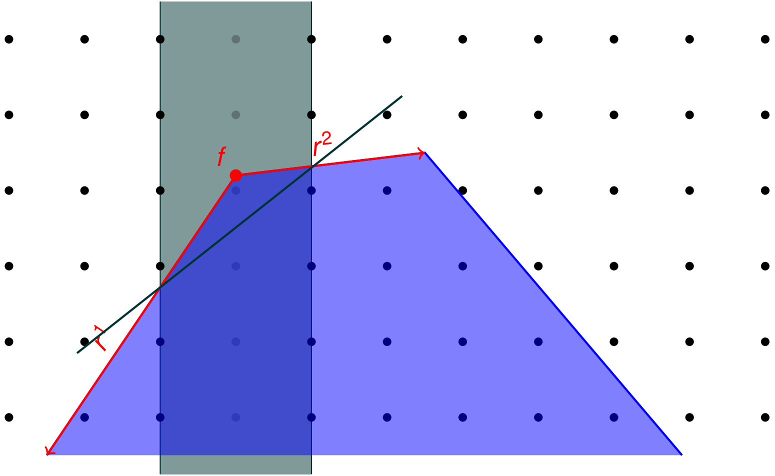
We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



We consider convex sets that are not lattice-free!

Intersection cuts [Balas 1970]



We consider convex sets that are not lattice-free!

Split cuts

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$.

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \le \pi_0 \quad \text{OR} \quad \pi x \ge \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \; \forall \; i = p+1, \ldots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \le \pi_1 \quad \text{OR} \quad \pi x \ge \pi_2,$$

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \leq \pi_1 \quad \text{OR} \quad \pi x \geq \pi_2,$$

where $(\pi, \pi_1, \pi_2) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+2} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \leq \pi_1 \quad \text{OR} \quad \pi x \geq \pi_2,$$

where $(\pi, \pi_1, \pi_2) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+2} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$ and we have $\pi_2 - \pi_1 > 1$.

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \ldots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \leq \pi_1 \quad \text{OR} \quad \pi x \geq \pi_2,$$

where $(\pi, \pi_1, \pi_2) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+2} \mid \pi_i = 0 \ \forall \ i = p+1, \ldots, p+n\}$ and we have $\pi_2 - \pi_1 > 1$.

In particular, we consider simple splits, i.e., $\pi = e_i$ for some $i \in \{1, \ldots, p\}$, that is

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \le \pi_0 \quad \text{OR} \quad \pi x \ge \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \le \pi_1 \quad \text{OR} \quad \pi x \ge \pi_2,$$

where $(\pi, \pi_1, \pi_2) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+2} \mid \pi_i = 0 \ \forall \ i = p+1, \dots, p+n\}$ and we have $\pi_2 - \pi_1 > 1$.

In particular, we consider simple splits, i.e., $\pi = e_i$ for some $i \in \{1, \dots, p\}$, that is

$$x_i \le \pi_1 \quad \text{OR} \quad x_i \ge \pi_2.$$

Split cuts

Intersection cuts derived from split sets are split cuts [Cook et al. 1990], i.e., they are cuts derived from the valid split disjunction

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1,$$

where $(\pi, \pi_0) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+1} \mid \pi_i = 0 \ \forall \ i = p+1, \ldots, p+n\}$.

Here, we consider general split disjunctions [Andersen et al. 2003], i.e., disjunctions of the form

$$\pi x \leq \pi_1 \quad \text{OR} \quad \pi x \geq \pi_2,$$

where $(\pi, \pi_1, \pi_2) \in \Pi := \{(\pi, \pi_0) \in \mathbb{Z}^{p+n+2} \mid \pi_i = 0 \ \forall \ i = p+1, \ldots, p+n\}$ and we have $\pi_2 - \pi_1 > 1$.

In particular, we consider simple splits, i.e., $\pi = e_i$ for some $i \in \{1, \ldots, p\}$, that is

$$x_i \leq \pi_1 \quad \text{OR} \quad x_i \geq \pi_2.$$

We call these disjunctions wide split disjunctions and the derived cuts wide split cuts.

Deriving a cut

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$\begin{aligned} x_i &= f_i + \sum_{j \in N} r_i^j x_j \quad && i \in B \\ x_j &\geq 0 && j \in N. \end{aligned}$$

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$
$$x_j \geq 0 \qquad j \in N.$$

Assume that $H = [h_l, h_u]$ is a hole for x_i and that $h_l < f_i < h_u$.

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$

$$x_j \geq 0 \quad j \in N.$$

Assume that $H = [h_l, h_u]$ is a hole for x_i and that $h_l < f_i < h_u$.

If we define $\alpha_j := max\{\alpha \mid h_l \leq f_i + \alpha r_i^j \leq h_u\}$,

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$

$$x_j \geq 0 \quad j \in N.$$

Assume that $H = [h_l, h_u]$ is a hole for x_i and that $h_l < f_i < h_u$.

If we define $\alpha_j := max\{\alpha \mid h_l \leq f_i + \alpha r_i^j \leq h_u\}$, then the intersection cut corresponding to the convex, not lattice-free set $S = \{x_i \in \mathbb{R} \mid h_l \leq x_i \leq h_u\}$ can be written as

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1,$$

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$

$$x_j \geq 0 \quad j \in N.$$

Assume that $H = [h_l, h_u]$ is a hole for x_i and that $h_l < f_i < h_u$.

If we define $\alpha_j := max\{\alpha \mid h_l \leq f_i + \alpha r_i^j \leq h_u\}$, then the intersection cut corresponding to the convex, not lattice-free set $S = \{x_i \in \mathbb{R} \mid h_l \leq x_i \leq h_u\}$ can be written as

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1,$$

and one can show that

$$\alpha_j = \begin{cases} \frac{f_i - h_l}{-r_i^j} & r_i^j < 0, \\ \frac{h_u - f_i}{r_i^j} & r_i^j > 0, \\ +\infty & otherwise. \end{cases}$$

Deriving a cut

Consider the polyhedron $P(B)$ derived from the optimal simplex tableau of the LP relaxation of (M),

$$
\begin{aligned}
x_i &= f_i + \sum_{j \in N} r_i^j x_j \qquad && i \in B \\
x_j &\geq 0 && j \in N.
\end{aligned}
$$

Assume that $H = [h_l, h_u]$ is a hole for x_i and that $h_l < f_i < h_u$.

If we define $\alpha_j := max\{\alpha \mid h_l \leq f_i + \alpha r_i^j \leq h_u\}$, then the intersection cut corresponding to the convex, not lattice-free set $S = \{x_i \in \mathbb{R} \mid h_l \leq x_i \leq h_u\}$ can be written as

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1,$$

and one can show that

$$
\alpha_j = \begin{cases} \frac{f_i - h_l}{-r_i^j} & r_i^j < 0, \\ \frac{h_u - f_i}{r_i^j} & r_i^j > 0, \\ +\infty & otherwise. \end{cases}
$$

$f_i - \lfloor f_i \rfloor$

$\lfloor f_i \rfloor - f_i$

Outline

Pure cutting plane approach

Pure cutting plane approach

The wide split cut dominates the intersection cut from a corresponding simple split!

Pure cutting plane approach

The wide split cut dominates the intersection cut from a corresponding simple split!

To test the impact of this dominance, we apply the following cutting plane procedure:

Pure cutting plane approach

The wide split cut dominates the intersection cut from a corresponding simple split!

To test the impact of this dominance, we apply the following cutting plane procedure:

Algorithm 4 pure cutting plane approach

relax integrality of (M) and solve, set $k = 1$
while ($k \leq$ n_rounds AND cuts_added) **do**
    for $i \in B \cap \{1, \ldots, p\}$ **do**
        if f_i is in a hole **then**
            compute the wide split cut and add it to (M)
        else if f_i is fractional **then**
            compute the simple split intersection cut and add it to (M)
        end if
    end for
    solve (M)
    $k \leftarrow k + 1$
end while

Pure cutting plane approach

Pure cutting plane approach

The MIPLIB 2010 instance `30n20b8` is a time-indexed scheduling type instance, and therefore has a significant number of GUB-link type equalities.

Pure cutting plane approach

The MIPLIB 2010 instance `30n20b8` is a time-indexed scheduling type instance, and therefore has a significant number of GUB-link type equalities.

After running the preprocessor of `cplex 12.6` on the instance, there are 30 variables with GUB-Link type equalities that certify between 3 and 7 holes per variable with medium length of 10.

Pure cutting plane approach

The MIPLIB 2010 instance `30n20b8` is a time-indexed scheduling type instance, and therefore has a significant number of GUB-link type equalities.

After running the preprocessor of `cplex 12.6` on the instance, there are 30 variables with GUB-Link type equalities that certify between 3 and 7 holes per variable with medium length of 10.

	with wide splits		without wide splits	
round	#cuts (wides)	% gap closed	#cuts	% gap closed
1	62 (10)	15.11	62	15.11
2	58 (13)	18.63	57	18.63
3	62 (14)	21.24	62	21.22
4	63 (9)	22.70	63	22.52
5	62 (14)	23.89	62	23.75
6	62 (10)	24.93	63	24.89
7	63 (6)	25.53	62	25.39
8	62 (10)	26.06	62	26.01
9	62 (11)	26.38	62	26.23
10	62 (11)	26.69	62	26.27
11	60 (11)	26.87	60	26.61
12	62 (11)	27.05	62	26.62
13	62 (10)	27.16	61	26.62
14	63 (10)	27.27	62	26.62
15	62 (11)	27.34	62	26.62

$$x_i = \sum_{j \in K} \lambda_j x_j$$

$$\sum_{j \in K} x_j = 1$$

Pure cutting plane approach

Pure cutting plane approach

To test the wide splits on holes that are not explicit in the model, we created random holes with 3 different intensities on 18 MIPLIB 2010 instances that contain bounded integer variables.

Pure cutting plane approach

To test the wide splits on holes that are not explicit in the model, we created random holes with 3 different intensities on 18 MIPLIB 2010 instances that contain bounded integer variables.

instance	int_0	int_1	int_2	int_3
50v-10	39.89	39.33	40.47	40.33
germanrr	12.80	12.58	12.26	12.28
lectsched-1	0.00	0.00	0.00	0.00
lectsched-2	0.00	0.00	0.00	0.00
lectsched-3	0.00	0.00	0.00	0.00
lectsched-4-obj	50.02	75.02	100.00	75.02
mik-250-1-100-1	70.46	75.38	71.55	77.72
mzzv11	20.44	50.65	56.78	69.06
n4-3	34.00	34.00	38.55	35.36
n9-3	23.64	23.64	24.67	24.33
neos16	17.65	29.41	29.41	23.53
neos-555424	30.71	33.03	39.00	52.92
neos-686190	7.15	7.15	7.25	13.58
noswot	0.00	0.00	0.00	0.00
rococoB10-011000	13.20	19.17	13.61	20.89
rococoC10-001000	30.90	30.18	30.66	64.86
sp98ir	7.60	9.40	10.91	10.86
timtab1	33.03	34.29	34.29	34.29
mean	27,96	33,80	36,39	39,60

percentage gap closed after 15 rounds of cuts

Branch & Cut

Branch & Cut

Assume that we have a MILP with explicit holes, modeled by the binary variables x_a,

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Dx + Ex_a = g \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n \\ & x_a \in \{0,1\}^k. \end{aligned} \tag{M'}$$

Branch & Cut

Assume that we have a MILP with explicit holes, modeled by the binary variables x_a,

$$
\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & Ax = b \\
& x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n
\end{aligned}
\tag{M'}
$$

Branch & Cut

Assume that we have a MILP with explicit holes, modeled by the binary variables x_a,

$$
\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & Ax = b \\
& Dx + Ex_a = g \\
& x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n \\
& x_a \in \{0, 1\}^k .
\end{aligned}
\tag{M'}
$$

Branch & Cut

Assume that we have a MILP with explicit holes, modeled by the binary variables x_a,

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Dx + Ex_a = g \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n \\ & x_a \in \{0,1\}^k. \end{aligned} \tag{M'}$$

Depending on the number of holes, the size of model (M') can be substantially greater than the size of (M).

Branch & Cut

Assume that we have a MILP with explicit holes, modeled by the binary variables x_a,

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Dx + Ex_a = g \qquad \text{(M')} \\ & x \in \mathbb{Z}_+^p \times \mathbb{R}_+^n \\ & x_a \in \{0,1\}^k. \end{aligned}$$

Depending on the number of holes, the size of model (M') can be substantially greater than the size of (M).

In the case of TSPMTW, relaxing (M') to (M) means relaxing to a TSPTW.

Branch & Cut

Branch & Cut

We are interested in comparing three different ways of solving (M'):

Branch & Cut

We are interested in comparing three different ways of solving (M'):

`CPXF`:

- apply `cplex 12.6` to model (M')

Branch & Cut

We are interested in comparing three different ways of solving (M'):

CPXF:

- apply `cplex 12.6` to model (M')

CPXR:

- apply `cplex 12.6` to model (M)

Branch & Cut

We are interested in comparing three different ways of solving (M'):

CPXF:

- apply `cplex 12.6` to model (M')

CPXR:

- apply `cplex 12.6` to model (M)
- whenever an incumbent x^* is found, check for satisfaction of holes

Branch & Cut

We are interested in comparing three different ways of solving (M'):

CPXF:

- apply `cplex 12.6` to model (M')

CPXR:

- apply `cplex 12.6` to model (M)
- whenever an incumbent x^* is found, check for satisfaction of holes
- if some hole $[h_l, h_u]$ for x_i is violated, branch with h_l and h_u as new upper and lower bound, respectively:

$$x_i \leq h_l \quad \text{OR} \quad x_i \geq h_u$$

Branch & Cut

We are interested in comparing three different ways of solving (M'):

`CPXF`:

- apply `cplex 12.6` to model (M')

`CPXR`:

- apply `cplex 12.6` to model (M)
- whenever an incumbent x^* is found, check for satisfaction of holes
- if some hole $[h_l, h_u]$ for x_i is violated, branch with h_l and h_u as new upper and lower bound, respectively:

$$x_i \leq h_l \quad \text{OR} \quad x_i \geq h_u$$

`CPXR+cuts`:

- the same **`CPXR`** with the separation of r rounds of wide split cuts at the root node

Branch & Cut

Branch & Cut

We compared these three methods on TSPMTW instances from [Belhaiza 2013]. We generated four testbeds based on graphs with 10 - 40 nodes.

Branch & Cut

We compared these three methods on TSPMTW instances from [Belhaiza 2013]. We generated four testbeds based on graphs with 10 - 40 nodes.

	CPXF	CPXR	CPXR+cuts
testbed_1	112.33	93.47	93.39
testbed_2	92.47	115.86	114.03
testbed_3	127.80	99.66	96.83
testbed_4	176.65	145.15	161.30

average comp. times in secs

Outline

1. Introduction
2. Intersection cuts and wide split cuts
3. Computational results
   3.1 Pure cutting plane approach
   3.2 Branch & Cut
4. Further topics & Outlook

Binarization splits

Binarization splits

Let us go back to the GUB-Link type equalities:

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, n\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$,

Binarization splits

Let us go back to the GUB-Link type equalities:

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, n\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$, and assume w.l.o.g. $\lambda_j < \lambda_{j+1}$.

Binarization splits

Let us go back to the GUB-Link type equalities:

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, n\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$, and assume w.l.o.g. $\lambda_j < \lambda_{j+1}$.

In the optimal simplex tableau of the LP relaxation of (M), assume that $\lambda_\ell < f_i < \lambda_{\ell+1}$ (f_i is in a hole for x_i),

Binarization splits

Let us go back to the GUB-Link type equalities:

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, n\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$, and assume w.l.o.g. $\lambda_j < \lambda_{j+1}$.

In the optimal simplex tableau of the LP relaxation of (M), assume that $\lambda_\ell < f_i < \lambda_{\ell+1}$ (f_i is in a hole for x_i), and let $L = \{1, \ldots, \ell\}$.

Binarization splits

Let us go back to the GUB-Link type equalities:

$$x_i = \sum_{j \in K} \lambda_j x_j$$
$$\sum_{j \in K} x_j = 1$$

for some $i \in \{1, \ldots, p\}$, some subset $K \subseteq \{1, \ldots, n\}$ and $\lambda_j \in \mathbb{N}$, $j \in K$, and assume w.l.o.g. $\lambda_j < \lambda_{j+1}$.

In the optimal simplex tableau of the LP relaxation of (M), assume that $\lambda_\ell < f_i < \lambda_{\ell+1}$ (f_i is in a hole for x_i), and let $L = \{1, \ldots, \ell\}$. Then the split disjunction

$$\sum_{j \in L} x_j \leq 0 \quad \text{OR} \quad \sum_{j \in L} x_j \geq 1$$

is valid for (M) and violated, and takes into account the hole.

Binarization splits

Binarization splits

Starting again from the optimal basis tableau

Binarization splits

Starting again from the optimal basis tableau

$$
\begin{aligned}
x_i &= f_i + \sum_{j \in N} r_i^j x_j \quad & i \in B \\
x_j &\geq 0 & j \in N,
\end{aligned}
$$

Binarization splits

Starting again from the optimal basis tableau

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$

$$x_j \geq 0 \quad\quad j \in N,$$

setting $\bar{B} := B \cap L, \bar{N} := N \cap L$

Binarization splits

Starting again from the optimal basis tableau

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$

$$x_j \geq 0 \quad j \in N,$$

setting $\bar{B} := B \cap L, \bar{N} := N \cap L$ and $f_0 := \sum_{i \in \bar{B}} f_i$,

$$\tilde{r}^j := \begin{cases} 1 + \sum_{i \in \bar{B}} r_i^j & j \in \bar{N}, \\ \sum_{i \in \bar{B}} r_i^j & j \in N \setminus \bar{N}, \end{cases}$$

then the intersection cut corresponding to the binarization split can be written as

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1,$$

Binarization splits

Starting again from the optimal basis tableau

$$x_i = f_i + \sum_{j \in N} r_i^j x_j \quad i \in B$$
$$x_j \geq 0 \quad j \in N,$$

setting $\bar{B} := B \cap L, \bar{N} := N \cap L$ and $f_0 := \sum_{i \in \bar{B}} f_i$,

$$\tilde{r}^j := \begin{cases} 1 + \sum_{i \in \bar{B}} r_i^j & j \in \bar{N}, \\ \sum_{i \in \bar{B}} r_i^j & j \in N \setminus \bar{N}, \end{cases}$$

then the intersection cut corresponding to the binarization split can be written as

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1,$$

where

$$\alpha_j = \begin{cases} \frac{f_0}{-\tilde{r}^j} & \tilde{r}^j < 0, \\ \frac{1-f_0}{\tilde{r}^j} & \tilde{r}^j > 0, \\ +\infty & \textit{otherwise}. \end{cases}$$

Integer Lifting

Integer Lifting

Integer Lifting is substantially weaker for the wide split than for the simple split.

Integer Lifting

Integer Lifting is substantially weaker for the wide split than for the simple split.

For the binarization split, the left-hand-side of the split disjunction can be written as

$$\sum_{j\in L} x_j = f_0 + \sum_{j\in N}\left(\sum_{i\in \bar{B}} r_i^j\right) x_j + \sum_{j\in \bar{N}} x_j .$$

Integer Lifting

Integer Lifting is substantially weaker for the wide split than for the simple split.

For the binarization split, the left-hand-side of the split disjunction can be written as

$$\sum_{j \in L} x_j = f_0 + \sum_{j \in N} \left(\sum_{i \in \bar{B}} r_i^j \right) x_j$$

Integer Lifting

Integer Lifting is substantially weaker for the wide split than for the simple split.

For the binarization split, the left-hand-side of the split disjunction can be written as

$$\sum_{j \in L} x_j = f_0 + \sum_{j \in N} \left(\sum_{i \in \bar{B}} r_i^j \right) x_j + \sum_{j \in \bar{N}} x_j.$$

All the variables in $\bar{N}$ are integer, so the lifted intersection cut obtained from the binarization split is equal to a lifted multi-row split cut, obtained from the system

$$\begin{aligned} x_i = f_i + \sum_{j \in N} r_i^j x_j, &\quad i \in \bar{B} \\ x_i \in \mathbb{Z}, &\quad i \in \bar{B} \\ x_j \geq 0, &\quad j \in N \\ x_j \in \mathbb{Z}, &\quad j = 1, \ldots, p, \end{aligned}$$

Integer Lifting

Integer Lifting is substantially weaker for the wide split than for the simple split.

For the binarization split, the left-hand-side of the split disjunction can be written as

$$\sum_{j \in L} x_j = f_0 + \sum_{j \in N} \left(\sum_{i \in \bar{B}} r_i^j \right) x_j + \sum_{j \in \bar{N}} x_j .$$

All the variables in $\bar{N}$ are integer, so the lifted intersection cut obtained from the binarization split is equal to a lifted multi-row split cut, obtained from the system

$$\begin{aligned} x_i = f_i + \sum_{j \in N} r_i^j x_j, &\quad i \in \bar{B} \\ x_i \in \mathbb{Z}, &\quad i \in \bar{B} \\ x_j \geq 0, &\quad j \in N \\ x_j \in \mathbb{Z}, &\quad j = 1, \ldots, p, \end{aligned}$$

and the split convex set $S = \{x_{\bar{B}} \in \mathbb{R}^{\bar{B}} \mid 0 \leq \sum_{i \in \bar{B}} x_i \leq 1\}$.

Outlook

Outlook

- Find more test problems!

Outlook

- Find more test problems!
- Figure out the difference between the wide split and the binarization split, non-lifted and lifted.

Outlook

- Find more test problems!
- Figure out the difference between the wide split and the binarization split, non-lifted and lifted.
- Implement them!

Outlook

- Find more test problems!
- Figure out the difference between the wide split and the binarization split, non-lifted and lifted.
- Implement them!

Thank you!