Efficient Separation of RLT Cuts for Implicit and Explicit Bilinear Products

Ksenia Bestuzheva [1], Ambros Gleixner [1,2], Tobias Achterberg [3]

[1]Zuse Institute Berlin and [2]HTW Berlin and [3]Gurobi Optimization

The 24th Conference on Integer Programming and Combinatorial Optimization
June 22, 2023

SPONSORED BY THE
Federal Ministry of Education and Research

Mixed-Integer Programs with Bilinear Products

$$
\begin{aligned}
\min\ & \mathbf{c}^T\mathbf{x} \\
\text{s.t. } & A\mathbf{x} \le \mathbf{b}, \\
& g(\mathbf{x}, \mathbf{w}) \le 0, \\
& x_i x_j \lesseqgtr w_{ij}\ \forall (i,j) \in \mathcal{I}^w, \qquad (*) \\
& \underline{\mathbf{x}} \le \mathbf{x} \le \overline{\mathbf{x}},\ \underline{\mathbf{w}} \le \mathbf{w} \le \overline{\mathbf{w}}, \\
& x_j \in \mathbb{R} \text{ for all } j \in \mathcal{I}^c,\ x_j \in \{0,1\} \text{ for all } j \in \mathcal{I}^b,
\end{aligned}
$$

where

g - nonlinear function,
$(*)$ - bilinear product relations.

- We aim to improve the performance of **spatial branch and bound** for MIPs with bilinear products
- We focus on efficiently constructing **tight linear programming (LP) relaxations**

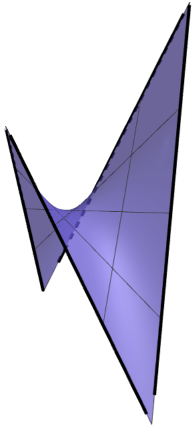
Bilinear Products

We are interested in constraints

$$x_i x_j \lesseqgtr w_{ij} \ \forall (i,j) \in \mathcal{I}^w.$$

These constraints are **nonlinear** and **nonconvex**.

Applications: pooling, packing, wastewater treatment, power systems optimisation, portfolio optimisation, etc.



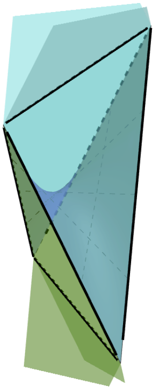
Relaxations of Bilinear Products

The convex hull of $x_i x_j = w_{ij}$ is given by the well-known **McCormick envelopes**:

$$
\begin{aligned}
w_{ij} &\geq \underline{x}_i x_j + x_i \underline{x}_j - \underline{x}_i \underline{x}_j,\\
w_{ij} &\geq \overline{x}_i x_j + x_i \overline{x}_j - \overline{x}_i \overline{x}_j,\\
w_{ij} &\leq \underline{x}_i x_j + x_i \overline{x}_j - \underline{x}_i \overline{x}_j,\\
w_{ij} &\leq \overline{x}_i x_j + x_i \underline{x}_j - \overline{x}_i \underline{x}_j.
\end{aligned}
$$

This is often a **weak relaxation**! Use other constraints to strengthen it.

RLT (Reformulation Linearization Technique): derive cuts from product relation + combinations of linear constraints/bounds.



RLT Cuts for Bilinear Products

We focus on RLT cuts derived by multiplying a constraint with a variable bound.

For example, multiply constraints of the problem by the lower bound factor of x_j (**reformulation** step):

$$\sum_{i=1}^{n} a_i x_i (x_j - \underline{x}_j) \leq b(x_j - \underline{x}_j).$$

Apply linearizations to each term $x_i x_j$ (**linearization** step):

- if relation $x_i x_j \lesseqgtr w_{ij}$ exists with the appropriate sign, replace $x_i x_j$ with w_{ij}
  - if the relation is violated in the right direction, this will increase cut violation
- otherwise, use a suitable relaxation

Motivation and Contributions

- RLT cuts can provide strong dual bounds
- Can this bounding strength of bilinear RLT also be leveraged for MILP solving?
- However, a large number of cuts is generated
  - Difficult to select which cuts to apply
  - LP sizes may increase dramatically
  - Even separation itself can be prohibitively expensive

Contributions:

- We develop a method for detecting implicit bilinear products in MILPs
- This enables us to apply bilinear RLT also to MILPs
- We propose an efficient separation algorithm that drastically reduces separation times

Implicit Bilinear Products

A bilinear product $w_{ij} = x_i x_j$, where x_i is binary, can be modeled by linear constraints:

Product	Implied relation	Big-M constraint
$w_{ij} \geq x_i x_j$	$x_i = 0 \;\Rightarrow\; w_{ij} \geq 0,$ $x_j = 1 \;\Rightarrow\; w_{ij} \geq x_j.$	$-w_{ij} + \underline{x}_j x_i \leq 0,$ $-w_{ij} + x_j + \bar{x}_j x_i \leq \bar{x}_j$
$w_{ij} \leq x_i x_j$	$x_i = 0 \;\Rightarrow\; w_{ij} \leq 0,$ $x_i = 1 \;\Rightarrow\; w_{ij} \leq x_j.$	$w_{ij} - \bar{x}_j x_i \leq 0,$ $w_{ij} - x_j - \underline{x}_j x_i \leq -\underline{x}_j.$

Implicit Products - General Form

Two constraints:

$$a_1 w_{ij} + b_1 x_i + c_1 x_j \le d_1,$$
$$a_2 w_{ij} + b_2 x_i + c_2 x_j \le d_2,$$

where

$$x_i \in \{0,1\},\ a_1 c_2 - a_2 c_1 \neq 0,\ a_1 a_2 \neq 0,$$

imply the following product relation:

$$x_i x_j \ge / \le \frac{a_1 a_2 w_{ij} + (a_2 b_1 - a_2 d_1 + a_1 d_2) x_i + a_1 c_2 x_j - a_1 d_2}{a_1 c_2 - a_2 c_1}.$$

(Derived by writing $x_i x_j \ge / \le A w_{ij} + B x_i + C x_j + D$ for unknown A, B, C, D and enforcing equivalence to the linear inequalities)

Implicit Products - Derivation

Write the general form with unknown A, B, C and D as implications:

$$\begin{aligned} x_i = 1 \quad &\Rightarrow \quad Bw_{ij} + (C-1)x_j \lesseqgtr -D - A, \\ x_i = 0 \quad &\Rightarrow \quad Bw_{ij} + Cx_j \lesseqgtr -D, \end{aligned}$$

Require equivalence to linear relations written as scaled implications:

$$\begin{aligned} x_i = 1 \quad &\Rightarrow \quad \alpha b_1 w_{ij} + \alpha c_1 x_j \lesseqgtr \alpha(d_1 - a_1), \\ x_i = 0 \quad &\Rightarrow \quad \beta b_2 w_{ij} + \beta c_2 x_j \lesseqgtr \beta d_2, \end{aligned}$$

Setting $\gamma = c_2 b_1 - b_2 c_1$ and solving the resulting system yields:

$$b_1 b_2 > 0, \; A = (1/\gamma)(b_2(a_1 - d_1) + b_1 d_2)$$
$$B = b_1 b_2/\gamma, \; C = b_1 c_2/\gamma, \; D = -b_1 d_2/\gamma, \; \gamma \neq 0,$$

where the inequality sign is '$\leq$' if $b1/\gamma \geq 0$, and '$\geq$' if $b1/\gamma \leq 0$.

Relation Types

Let $x_i \in \{0, 1\}$ and let f be a binary constant.

Implied relation	Linear relation between 2 variables activated by x_i: $x_i = f \Rightarrow \bar{a}w_{ij} + \bar{c}x_j \leq \bar{d}$;	Hashtable with 3 sorted variables as keys
Implied bound	Variable bound activated by x_i: $x_i = f \Rightarrow \bar{a}w_{ij} \leq \bar{d}$;	Sorted array per variable
Used only together with one of the above:		
Clique	If binary variables x_k, $k \in C$ and $!x_k$, $k \in C^r$ are in a clique, then: $\sum_{k \in C} x_k + \sum_{k \in C^r} (1 - x_k) \leq 1$	Clique table
Unconditional relation	Relation between x_j and w_{ij} (implied bound, clique, linear constraint with 2 nonzeroes)	Hashtable with 2 sorted variables as keys
Global bound	Global variable bound on w_{ij}	Accessed directly through the variable

Efficient data structures are crucial for performance.

Detecting Implicit Products

- **Find implied relations** $x_i = f \Rightarrow \bar{a}_1 w_{ij} + \bar{c}_1 x_j \leq \bar{d}_1$ among constraints with 3 nonzeroes and at least one binary variable.
- For each implied relation, look for the **second relation**:
  - It must be implied by $x_i = !f$ and contain w_{ij}

Product relations can also be described without a size 3 constraint:

- **For each implied bound** $x_i = f \Rightarrow w_{ij} \leq \bar{d}_1$, look for the **second relation**:
  - **unconditional relation** of w_{ij} and x_j.

Variable order matters: depending on the order, we get different products.

For implicit products, the linear expression of (w_{ij}, x_i, x_j) is used in place of w_{ij}.

Standard Separation Algorithm

Context - separation in spatial BB solvers:

- LP-based spatial BB builds LP relaxations of node subproblems
- $(\mathbf{x}^*, \mathbf{w}^*)$ - solution of an LP relaxation
- Suppose that $(\mathbf{x}^*, \mathbf{w}^*)$ violates the relation $x_i x_j \lesseqgtr w_{ij}$ for some $(i, j) \in \mathcal{I}^w$
- Need to generate cuts that separate $(\mathbf{x}^*, \mathbf{w}^*)$ from the feasible region

RLT cut separation we use as a baseline:

- Iterate over all linear constraints
- For each constraint, iterate over all x_j that participate in bilinear relations
- Generate RLT cuts using bound factors of x_j

Row Marking

Observation:

- Consider reformulated constraint $a_i x_i x_j + \mathbf{a}_{\setminus i}^T \mathbf{x}_{\setminus i} x_j \leq b x_j$
- Replace with $a_i w_{ij} + L^{under}(\mathbf{a}_{\setminus i}^T \mathbf{x}_{\setminus i} x_j) \leq b x_j$
- The cut can be violated only if $a_i x_i^* x_j^* < a_i w_{ij}^*$

Algorithm:

- Create data structures to enable efficient access to
  - all variables appearing in bilinear products together with a given variable
  - the bilinear product relation involving two given variables
- For each variable x_i, create a sparse array to store marked rows
- For each j such that $(i,j) \in \mathcal{I}^w$, iterate over linear rows containing x_j
- Store the rows in the marked rows array with the following marks:
  - LE: the row contains a term $a_{rj} x_j$ such that $a_{rj} x_i^* x_j^* < a_{rj} w_{ij}^*$
  - GE: the row contains a term $a_{rj} x_j$ such that $a_{rj} x_i^* x_j^* > a_{rj} w_{ij}^*$
  - BOTH: the row contains terms fitting both cases above

The Use of Row Marks

Generate cuts only for the following combinations of rows and bound factors $(x_i - \underline{x}_i)$ and $(\bar{x}_i - x_i)$:

- $mark$ = LE:
  - "$\leq$" constraints are multiplied with $(x_i - \underline{x}_i)$
  - "$\geq$" constraints are multiplied with $(\bar{x}_i - x_i)$
- $mark$ = GE:
  - "$\leq$" constraints are multiplied with $(\bar{x}_i - x_i)$
  - "$\geq$" constraints are multiplied with $(x_i - \underline{x}_i)$
- $mark$ = BOTH:
  - both "$\leq$" and "$\geq$" constraints are multiplied with both $(x_i - \underline{x}_i)$ and $(\bar{x}_i - x_i)$
- marked equality constraints are always multiplied with x_i itself

Efficient Separation of RLT Cuts

- For each variable x_i that appears in products:
  - For each violated product relation with $x_i x_j$, mark and store constraints with nonzero a_{rj}
  - Iterate over marked rows:
    - For each marked row, construct cuts with suitable sides and multipliers
    - If a cut is violated, add it to the cut pool

For example:

$$x_1 \leq 0, \; x_1 x_2 \leq w, \; x_2 \in [1, 2]$$

Reformulations are: $x_1(x_2 - 1) \leq 0, \; x_1(2 - x_2) \leq 0$

If at LP solution $x_1^* x_2^* > w^*$, use only the second reformulation.

If several linearizations are available: use the most violated.

Term Linearization

- $x_i x_j \to \ell(w_{ij}, x_i, x_j)$ if relation $x_i x_j \lesseqgtr \ell(w_{ij}, x_i, x_j)$ exists with the appropriate sign,
- if $i = j \in \mathcal{I}^b$, then $x_i x_j = x_i$,
- if $i = j \notin \mathcal{I}^b$, then $x_i x_j = x_j^2$ is outer approximated by a secant or tangent,
- if $i \neq j$, $i, j \in \mathcal{I}^b$ and a clique constraint exists, then:
  $x_i + x_j \leq 1 \Rightarrow x_i x_j = 0$; $x_i - x_j \leq 0 \Rightarrow x_i x_j = x_i$;
  $-x_i + x_j \leq 0 \Rightarrow x_i x_j = x_j$; $-x_i - x_j \leq -1 \Rightarrow x_i x_j = x_j + x_j - 1$,
- otherwise, use the McCormick relaxation.

Projection

McCormick is tight if at least one of the variables is at bound $\Rightarrow$ replacing such a product does not add to the violation.

Construct a smaller system by fixing all variables that are at bound:

$$\sum_{i=1}^{n} a_i x_i \leq b \quad \text{becomes} \quad \sum_{i \in !B} a_i x_i \leq b - \sum_{i \in B} a_i x_i^*,$$

$!B$ - indices of variables not at bound,
B - indices of variables at bound.

Check violation for projected cuts first.

However...
if McCormick constraints are dynamically added as cuts, the above does not hold $\Rightarrow$ some violated cuts might be ignored.

Computational Setup

- Using a development version of SCIP
- Linear solver SoPlex
- Time limit one hour
- Testsets: subsets where (either explicit or implicit) bilinear products exist chosen from
  - 1846 MINLPLib instances for MINLP
  - a testset comprised of 666 instances from MIPLIB3, MIPLIB 2003, 2010 and 2017, and Cor@l

- At most 20 unknown bilinear terms that a reformulated constraint can have in order to be used
- Frequency: every 10 nodes
- 1 separation round in tree nodes, 10 separation rounds in the root node
- Implicit product detection and projection filtering enabled until specified otherwise

Impact of RLT Cuts: MILP

Settings:

- Off: RLT cuts are disabled
- IERLT: RLT cuts are added for both explicit and implicit products

		Off			IERLT			IERLT/Off	
Subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
All	971	905	**45.2**	1339	**909**	46.7	**1310**	1.03	0.98
Affected	581	571	**48.8**	1936	**575**	51.2	**1877**	1.05	0.97
[0,tilim]	915	905	**34.4**	1127	**909**	35.6	**1104**	1.04	0.98
[1,tilim]	832	822	**47.2**	1451	**826**	49.0	**1420**	1.04	0.98
[10,tilim]	590	580	**126.8**	3604	**584**	133.9	**3495**	1.06	0.97
[100,tilim]	329	319	439.1	9121	**323**	**430.7**	**8333**	0.98	0.91
[1000,tilim]	96	88	1436.7	43060	**92**	**1140.9**	**31104**	0.79	0.72
All-optimal	899	899	**31.9**	**1033**	899	34.1	1053	1.07	1.02

Impact of RLT Cuts Derived From Explicit Products: MINLP

Settings:

- Off: RLT cuts are disabled
- ERLT: RLT cuts are added only for products that exist explicitly in the problem
- IERLT: RLT cuts are added for both explicit and implicit products

		Off			ERLT			ERLT/Off	
Subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
All	6622	4434	67.5	3375	**4557**	**57.5**	**2719**	0.85	0.81
Affected	2018	1884	18.5	1534	**2007**	**10.6**	**784**	0.57	0.51
[0,timelim]	4568	4434	10.5	778	**4557**	**8.2**	**569**	0.78	0.73
[1,timelim]	3124	2990	28.3	2081	**3113**	**20.0**	**1383**	0.71	0.67
[10,timelim]	1871	1737	108.3	6729	**1860**	**63.6**	**3745**	0.59	0.56
[100,tilim]	861	727	519.7	35991	**850**	**196.1**	**12873**	0.38	0.36
[1000,tilim]	284	150	2354.8	196466	**273**	**297.6**	**23541**	0.13	0.12
All-optimal	4423	4423	8.6	627	4423	**7.5**	**518**	0.87	0.83

Impact of RLT Cuts Derived From Implicit Products: MINLP

Settings:

- Off: RLT cuts are disabled
- IERLT: RLT cuts are added for both explicit and implicit products

		ERLT			IERLT			ERLT/IERLT	
Subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
All	6622	4565	**57.0**	2686	**4568**	57.4	**2638**	1.01	0.98
Affected	1738	1702	**24.2**	1567	**1705**	24.8	**1494**	1.02	0.95
[0,timelim]	4601	4565	**8.5**	587	**4568**	8.6	**576**	1.01	0.98
[1,timelim]	3141	3105	**21.1**	1436	**3108**	21.4	**1398**	1.01	0.97
[10,timelim]	1828	1792	**74.1**	4157	**1795**	75.4	**4012**	1.02	0.97
[100,tilim]	706	670	**359.9**	**22875**	**673**	390.4	24339	1.09	1.06
[1000,tilim]	192	156	**1493.3**	**99996**	**159**	1544.7	107006	1.03	1.07
All-optimal	4532	4532	**7.7**	540	4532	7.8	**529**	1.02	0.98

Impact of the Separation Algorithm

Settings:

- RLT cuts for both explicit and implicit products are enabled
- Marking-off: a straightforward separation algorithm is used
- Marking-on: the new separation algorithm is used

			Marking-off			Marking-on			M-on/M-off	
Test set	subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
MILP	All	949	780	124.0	**952**	**890**	**45.2**	1297	0.37	1.37
	Affected	728	612	156.6	**1118**	**722**	**46.4**	1467	0.30	1.31
	All-optimal	774	774	58.4	**823**	774	**21.2**	829	0.36	1.01
MINLP	All	6546	4491	64.5	**2317**	**4530**	**56.4**	2589	0.88	1.12
	Affected	3031	2949	18.5	**1062**	**2988**	**14.3**	1116	0.78	1.05
	All-optimal	4448	4448	9.1	**494**	4448	**7.4**	502	0.81	1.02

Impact of the Separation Algorithm on Separation Times

Settings:

- RLT cuts for both explicit and implicit products are enabled
- Marking-off: a straightforward separation algorithm is used
- Marking-on: the new separation algorithm is used

Test set	Setting	avg %	max %	N($< 5\%$)	N(5-20%)	N(20-50%)	N(50-100%)	fail
MILP	Marking-off	54.2	99.6	121	117	169	552	16
	Marking-on	**2.8**	**71.6**	853	87	31	4	**0**
MINLP	Marking-off	15.1	100.0	3647	1265	1111	685	77
	Marking-on	**2.4**	100.0	6140	376	204	49	**16**

Impact of Projection: MILP

Settings:

- No-proj: the projected LP is not used
- Proj: the projected LP is used

		No-proj			Proj			relative	
Subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
All	972	**912**	46.4	1329	911	**46.1**	**1302**	0.99	0.98
Affected	530	**523**	75.7	3092	522	**74.6**	**2964**	0.99	0.96
[0,timelim]	919	**912**	36.0	1155	911	**35.7**	**1126**	0.99	0.98
[1,timelim]	832	**825**	50.3	1504	824	**49.8**	**1462**	0.99	0.97
[10,timelim]	582	**575**	143.4	3886	574	**141.7**	**3741**	0.99	0.96
[100,tilim]	323	**316**	485.0	9601	315	**471.3**	**9065**	0.97	0.94
[1000,tilim]	96	**89**	**1483.8**	45276	88	1512.2	**43061**	1.02	0.95
All-optimal	904	904	33.5	1054	904	**33.4**	**1040**	1.00	0.99

Impact of Projection: MINLP

Settings:

- No-proj: the projected LP is not used
- Proj: the projected LP is used

		No-proj			Proj			relative	
Subset	instances	solved	time	nodes	solved	time	nodes	time	nodes
All	6637	**4582**	57.9	2689	4581	**57.7**	**2674**	1.00	0.99
Affected	2476	**2438**	23.3	1681	2437	**23.1**	**1660**	0.99	0.99
[0,timelim]	4620	**4582**	8.8	595	4581	**8.7**	**590**	0.99	0.99
[1,timelim]	3137	**3099**	22.4	1483	3098	**22.3**	**1467**	0.99	0.99
[10,timelim]	1854	**1816**	77.7	4253	1815	**76.4**	**4210**	0.98	0.99
[100,tilim]	743	**705**	377.4	23389	704	**364.4**	**22680**	0.97	0.97
[1000,tilim]	205	**167**	**1434.5**	**98443**	166	1480.7	105546	1.03	1.07
All-optimal	4543	4543	8.0	539	4543	**7.9**	**533**	0.99	0.99

Results with Gurobi

- Ran with Gurobi 10.0 beta
- Same RLT algorithms, implementation details may differ
- Internal Gurobi test set
- Time limit 10000s

	MILP			MINLP		
Subset	instances	timeR	nodeR	instances	timeR	nodeR
All	5011	0.99	0.97	806	0.73	0.57
[0,timelim]	4830	0.99	0.96	505	0.57	0.44
[1,timelim]	3332	0.98	0.96	280	0.40	0.29
[10,timelim]	2410	0.97	0.93	188	0.29	0.20
[100,timelim]	1391	0.95	0.91	114	0.17	0.11
[1000,timelim]	512	0.89	0.83	79	0.12	0.08
Solved	RLT off: +41; RLT on: +37			RLT off: +2; RLT on: +35		

Summary

- Implicit product relations are detected by analysing MILP constraints
- We use row marking to efficiently separate RLT cuts
- We use a projected LP to speed up separation and filter out less promising cuts

- RLT cuts improve performance for difficult MILP instances ([1000,timelim])
- RLT cuts for explicit products considerably improve MINLP performance
- RLT cuts derived from implicit products are slightly detrimental to MINLP performance
- The separation algorithm is crucial and enables the improvements yielded by RLT
- Projection slightly improves overall performance, but slightly worsens performance on difficult instances