Reformulation-Linearisation Technique for Implicit Bilinear Relations

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

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

June 24, 2021

Mixed-Integer Programs with Bilinear Products

$$
\begin{aligned}
\min\ & c \cdot x \\
\text{s.t. } & Ax - b \leq 0, \\
& g(x) \leq 0, \\
& \pm (x_i x_j - w_{ij}) \leq 0 \ \forall i, j \in P, \qquad (*) \\
& x_j \in \mathbb{Z}, \ j \in I,
\end{aligned}
$$

where

c, b - constant vectors,
x - variable vector,
A - $m \times n$ matrix,

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

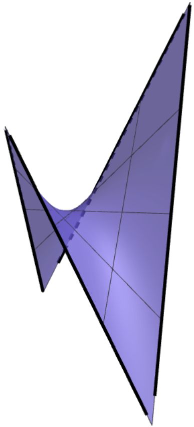
Bilinear Products

We are interested in constraints

$$\pm(x_i x_j - w_{ij}) \leq 0.$$

These constraints are nonlinear and nonconvex.

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



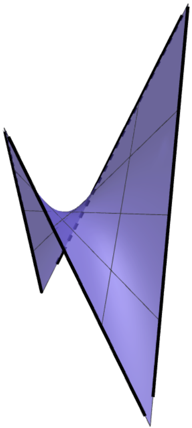
McCormick Envelopes

Bilinear products of bounded variables are linearized by McCormick envelopes:

$xy = w$

is replaced by:

$$w \geq x^l y + x y^l - x^l y^l$$
$$w \geq x^u y + x y^u - x^u y^u$$
$$w \leq x^l y + x y^u - x^l y^u$$
$$w \leq x^u y + x y^l - x^u y^l$$



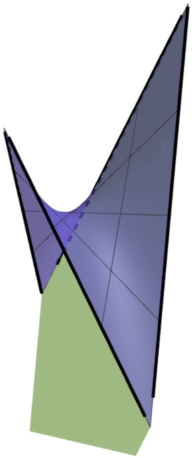
McCormick Envelopes

Bilinear products of bounded variables are linearized by McCormick envelopes:

$$xy = w$$

is replaced by:

$$w \geq x^l y + x y^l - x^l y^l$$
$$w \geq x^u y + x y^u - x^u y^u$$
$$w \leq x^l y + x y^u - x^l y^u$$
$$w \leq x^u y + x y^l - x^u y^l$$



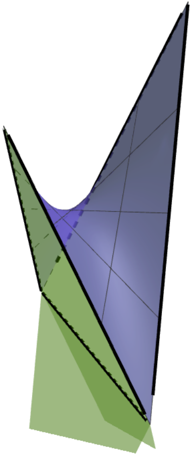
McCormick Envelopes

Bilinear products of bounded variables are linearized by McCormick envelopes:

$$xy = w$$

is replaced by:

$$w \geq x^l y + x y^l - x^l y^l$$
$$w \geq x^u y + x y^u - x^u y^u$$
$$w \leq x^l y + x y^u - x^l y^u$$
$$w \leq x^u y + x y^l - x^u y^l$$



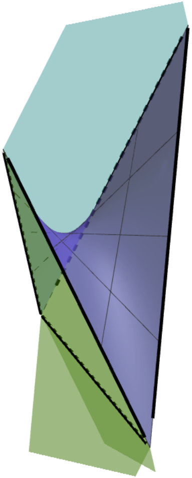
McCormick Envelopes

Bilinear products of bounded variables are linearized by McCormick envelopes:

$xy = w$

is replaced by:

$$w \geq x^l y + x y^l - x^l y^l$$
$$w \geq x^u y + x y^u - x^u y^u$$
$$w \leq x^l y + x y^u - x^l y^u$$
$$w \leq x^u y + x y^l - x^u y^l$$



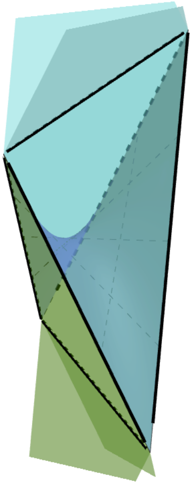
McCormick Envelopes

Bilinear products of bounded variables are linearized by McCormick envelopes:

$xy = w$

is replaced by:

$$w \geq x^l y + x y^l - x^l y^l$$
$$w \geq x^u y + x y^u - x^u y^u$$
$$w \leq x^l y + x y^u - x^l y^u$$
$$w \leq x^u y + x y^l - x^u y^l$$



RLT Cuts

McCormick relaxation describes the convex hull, but the LP solution often violates the bilinear constraints!

General idea: use other constraints too.

RLT (Reformulation Linearization Technique): derive cuts from the product relation + variable bound + a linear constraint:

$$x_1 x_2 = w_{12},$$
$$a_1 x_1 + a^r \cdot x^r = b,$$
$$x_2 \in [x_2^l, x_2^u].$$

(a^r, x^r - coefficient and variable vectors representing the remaining part of the linear constraint.)

RLT Cuts - Construction

Multiply the constraint with $(x_2 - x_2^l)$ or $(x_2^u - x_2)$: ← Reformulation

$$a_1 x_1 (x_2 - x_2^l) + a^r x^r (x_2 - x_2^l) = b(x_2 - x_2^l),$$

Linearize the nonlinear terms: ← Linearization

$$a_1 w_{12} - a_1 x_1 x_2^l + L^{under}(a^r x^r x_2) - a^r x^r x_2^l \leq b(x_2 - x_2^l),$$
$$a_1 w_{12} - a_1 x_1 x_2^l + L^{over}(a^r x^r x_2) - a^r x^r x_2^l \geq b(x_2 - x_2^l),$$

Relaxations of $x_i x_2$, if product $x_i x_2 = w_{i2}$ does not exist:

- McCormick for bilinear products
- Tangent/secant for quadratic terms

Contributions

RLT cuts:

+ Can provide strong dual bounds

- A large number of cuts is generated, difficult to select which cuts to apply; even separation itself can be expensive

In this work we develop:

- Detection of hidden products in MILPs
- Separation algorithm

Hidden Bilinear Products

Bilinear relations can be modeled by linear constraints with at least one binary variable:

Product	Implied relation	Big-M constraint
$w \geq xy$	$x = 0 \Rightarrow w \geq 0,$ $x = 1 \Rightarrow w \geq y.$	$w - w^l x \geq 0,$ $w - y + (w^l - y^u)x \geq w^l - y^u.$
$w \leq xy$	$x = 0 \Rightarrow w \leq 0,$ $x = 1 \Rightarrow w \leq y.$	$w - w^u x \leq 0,$ $w - y + (w^u - y^l)x \leq w^u - y^l.$

(x binary)

Hidden Products - General Form

Two constraints:

$$a_1 w + b_1 x + c_1 y \leq d_1,$$
$$a_2 w + b_2 x + c_2 y \leq d_2,$$

where

$$x \in \{0,1\},\ \ a_1 c_2 - a_2 c_1 \neq 0,$$
$$b_1 > 0, b_2 \leq 0,\ \ a_1 a_2 \neq 0,$$

imply the following product relation:

$$xy \geq / \leq \frac{a_1 a_2 w + (a_2 b_1 - a_2 d_1 + a_1 d_2)x + a_1 c_2 y - a_1 d_2}{a_1 c_2 - a_2 c_1}$$

(Derived by writing $xy \geq / \leq Aw + Bx + Cy + D$ for unknown A, B, C, D and enforcing the equivalence to the linear inequalities)

Relation Types

For product detection, we use several types of relations:

Implied relation	Linear relation between 2 variables activated by a binary variable: $x = f \Rightarrow \bar{a}w + \bar{c}y \leq \bar{d}$;	Hashtable with 3 sorted variables as keys
Implied bound	Variable bound activated by a binary variable: $x = f \Rightarrow \bar{a}w \leq \bar{d}$;	Sorted array for each variable
Clique	If binary variables x_i, $i \in C$ and $!x_i$, $i \in C^r$ are in a clique, then: $\sum_{i \in C} x_i + \sum_{i \in C^r} (1 - x_i) \leq 1$	Clique table

Also linear constraints with 2 or 3 nonzeros and global variable bounds.

(f - binary constant)

Detecting Hidden Products

- **Find implied relations** $x = f \Rightarrow \bar{a}_1 w + \bar{c}_1 y \leq \bar{d}_1$ 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 $x =! f$ and contain w or w and y

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

- **For each implied bound** $x = f \Rightarrow w \leq \bar{d}_1$, look for the **second relation**:
  - **unconditional relation** of w and y (implied bound, clique, linear constraint with 2 nonzeroes).

Variable order matters: depending on the choice of w, x and y, we get different products.

Separation of RLT Cuts

When replacing $a_1 x_1 x_2 + a^r \cdot x^r x_2 \le b x_2$ with

$$a_1 w_{12} + L^{under}(a^r \cdot x^r x_2) \le b x_2,$$

the cut can be violated if $a_1 x_1 x_2 < a_1 w_{12}$.

Separation Algorithm

- For each variable x_j that appears in products:
  - For each violated product relation with $x_i x_j$, mark constraints of x_i where a_i has the right sign.
  - The marks tell us which combinations of bound factors and sides to use for cut generation.
  - For each marked row:
    - Construct cuts with suitable sides and multipliers;
    - If a cut is violated, add it to the cut pool.

If several linearisations are available: use the most violated.

For example:

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

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

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

Term Linearisation

- Replace $x_i x_j$ by:
  - linearisation $L(w_{ij}, x_i, x_j)$, if it exists in the right direction;
  - clique relationship if x_i and x_j are binary and clique exists:
    - $x_i + x_j \leq 1 \Rightarrow x_i x_j = 0$,
    - $x_i + (1 - x_j) \leq 1 \Rightarrow x_i x_j = x_i$,
    - $(1 - x_i) + x_j \leq 1 \Rightarrow x_i x_j = x_j$,
    - $(1 - x_i) + (1 - x_j) \leq 1 \Rightarrow x_i x_j = x_i + x_j - 1$;
  - McCormick envelopes otherwise.
- Replace x_j^2 by:
  - x_j if x_j is binary
  - Tangent for underestimation and secant for overestimation otherwise

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:

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

$!B$ - 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 Results - Setup

- Using a development version of SCIP;
- Linear solver SoPlex;
- Time limit $1/2$ hour.
- Testsets: subsets of MINLPLib and MIPLIB2017 where (either explicit or implicit) bilinear products exist.

Computational Results - Implicit Products

Settings:

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

Linear instances:

Setting	Solved	T	N	T_100	N_100	T_1000	N_1000
Off	174	**46.35**	**1363**	**412.3**	10022	1104.9	24052
Hidden	**177**	48.76	1392	414.2	**9491**	**1069.9**	**19470**

Nonlinear instances:

Setting	Solved	T	N
Off	869	51.38	2698
Existing	890	**47.01**	**2331**
Hidden	**892**	48.82	2396

Computational Results - Separation Algorithm

Settings:

- RLT cuts for both existing and hidden products are enabled
- SimpleSepa: a straightforward separation algorithm is used
- NewSepa: the new separation algorithm is used

Linear instances:

Setting	Solved	T	N
SimpleSepa	154	94.57	**1180**
NewSepa	**174**	**49.82**	1448

Nonlinear instances:

Setting	Solved	T	N
SimpleSepa	**902**	**45.61**	**2188**
NewSepa	898	46.23	2373

Computational Results - Projection

Settings:

- RLT cuts for both existing and hidden products are enabled
- NoProject: the projected LP is not used
- Project: the projected LP is used

Linear instances:

Setting	Solved	T	N
NoProject	173	49.69	1463
Project	**176**	**48.90**	**1424**

Nonlinear instances:

Setting	Solved	T	N
NoProject	886	**48.00**	**2374**
Project	**888**	48.43	2424

Summary

- Implicit product relations are detected by analysing MILP constraints.
- Use row marking to speed up separation.
- Use a projected LP to speed up separation and filter out cuts that are less promising.
- RLT cuts improve performance for difficult MILP instances; the separation algorithm is crucial.
- RLT cuts help on MINLP instances; however, adding cuts derived from implicit products as well as utilising the new separation algorithm appears to be slightly detrimental to performance.