Intersection cuts from bilinear disjunctions

Matteo Fischetti, University of Padova
(joint work with Michele Monaci, University of Bologna)

MIQP as a MILP with bilinear eq.s

- We consider the Mixed-Integer Quadratic Problem (MIQP)

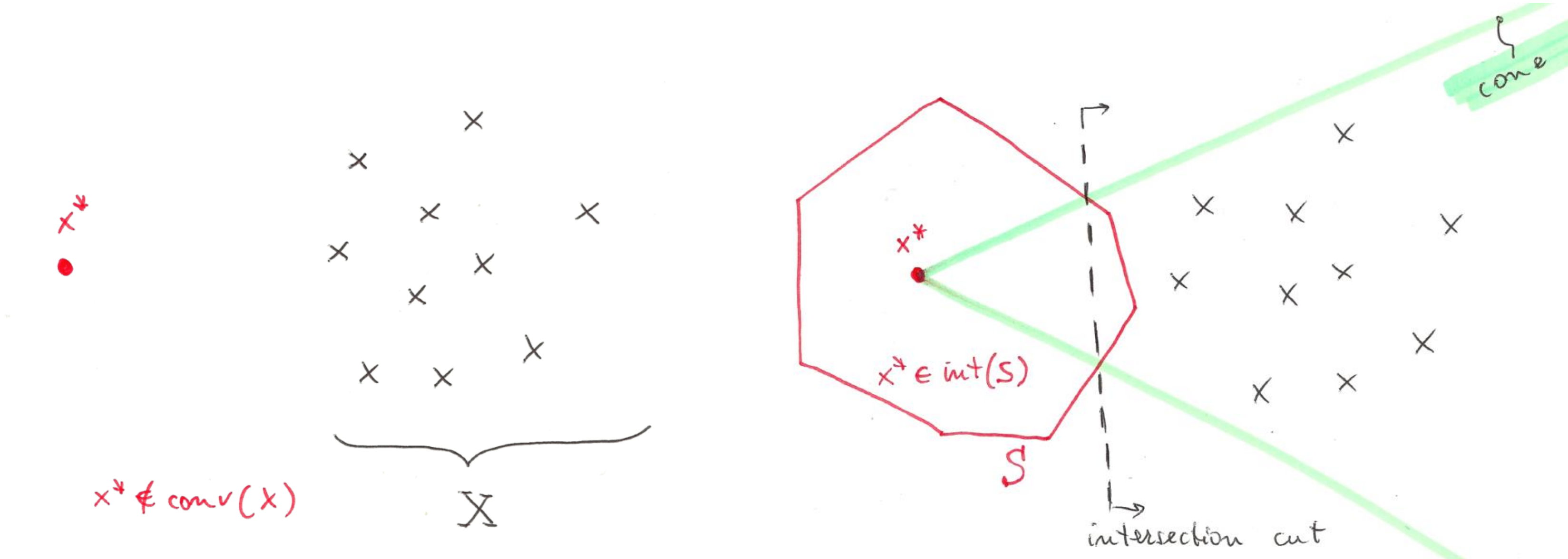
$$
\begin{array}{lrl}
(MIQP) & \min a_0^T x + x^T Q^0 x & \\
& a_k^T x + x^T Q^k\, x \,@\, b, & k = 1, \dots, m \\
& \ell_j \le x_j \le u_j, & j = 1, \dots, n \\
& x_j \text{ integer}, & j \in \mathcal{I}, \\
& x_j \text{ continuous}, & j \in \mathcal{C},
\end{array}
$$

restated as Mixed-Integer Bilinear Problem (MIBLP)

$$
\begin{array}{lrl}
(MIBLP) & \min_x c^T x & \\
& Ax = b & \\
& \ell_j \le x_j \le u_j, & j = 1, \dots, n \\
& x_j \text{ integer}, & j \in \mathcal{I} \\
& x_j \text{ continuous}, & j \in \mathcal{C} \\
& x_{r_k} = x_{p_k} x_{q_k}, & k = 1, \dots, K,
\end{array}
$$

Intersection Cuts (ICs)

- **Intersection cuts** (Balas, 1971): a powerful tool to separate a point **x*** from a set **X** by a liner cut



- All you need is
  - a **cone** pointed at **x*** containing all **x ε X**
  - a **convex set S** with x* (but no **x ε X**) in its <u>**interior**</u>
- If x* **vertex** of an LP relaxation, a suitable cone comes for the **LP basis**

Bilinear-free sets

- **Observation**: given an infeasible point x*, any **branching** disjunction violated by x* implicitly defines a **convex set S** with x* (but no feasible x) in its **interior**

$$\bigvee_{i=1}^{k} (g_i^T x \geq g_{i0}) \quad \rightarrow \quad S = \{x : g_i^T x \leq g_{0i}, \; i = 1, \ldots, k\}$$

- Thus, **in principle,** one could always generate an IC instead of branching → not always advisable because of numerical issues, slow convergence, tailing off, cut saturation, etc. **#LikeGomoryCuts**

- **Candidate branching disjunctions** (supplemented by MC cuts) are the 1- and 2-level (possibly shifted) **spatial branching** conditions:

$$(x \leq x^*) \vee (x \geq x^*)$$

$$(x \leq x^*, y \leq y^*) \vee (x \leq x^*, y \geq y^*) \vee (x \geq x^*, y \leq y^*) \vee (x \geq x^*, y \geq y^*)$$

IC separation issues

- IC separation can be probematic, as we need to read the cone rays from the LP tableau → **numerical accuracy** can be a big issue here!

- **Notation**: consider w.l.o.g. an LP in standard form (no var. ub's) and let

$\min\{\hat{c}^T \xi : \hat{A}\xi = \hat{b}, \xi \geq 0\}$ be the LP relaxation at a given node

$S = \{\xi : g_i^T \xi \leq g_{0i},\ i = 1, \ldots, k\}$ be a given bilinear-free set

$\bigvee_{i=1}^{k} (g_i^T \xi \geq g_{i0})$ be the disjunction to be satisfied by all feas. sol.s

Numerically safe ICs

$$\bigvee_{i=1}^{k} (g_i^T \xi \geq g_{i0})$$

A **single** valid inequality can be obtained by taking, for each variable, the worst LHS Coefficient (and RHS) in each disjunction

$$\bigvee_{i=1}^{k} (\overline{g}_i^T \xi \geq \overline{g}_{i0})$$

To be applied to a **reduced form** of each disjunction where the coefficient of all basic variables is zero (kind of LP reduced costs)

$$\bigvee_{i=1}^{k} (\frac{\overline{g}_i^T}{\overline{g}_{i0}} \xi \geq 1)$$

Algorithm 1: Intersection cut separation

Input : An LP vertex ξ^* along with its a associated LP basis $\hat{B}$;
the feasible-free polyhedron $S = \{\xi : g_i^T \xi \leq g_{0i},\ i = 1, \ldots, k\}$ and the associated valid disjunction $\bigvee_{i=1}^{k} (g_i^T \xi \geq g_{i0})$ whose members are violated by ξ^*;

Output: A valid intersection cut violated by ξ^*;

for $i := 1$ **to** k **do**
    $(\overline{g}_i^T, \overline{g}_{i0}) := (g_i^T, g_{i0}) - u_i^T(\hat{A}, \hat{b})$, where $u_i^T = (g_i)_{\hat{B}}^T \hat{B}^{-1}$
end
for $j := 1$ **to** n **do** $\gamma_j := \max\{\overline{g}_{ij} / \overline{g}_{i0} : i \in \{1, \ldots, k\}\}$;
return *the violated cut* $\gamma^T \xi \geq 1$

Computational analysis

- **Three** algorithms under comparison

  - ✓ **SCIP**: the general-purpose solver SCIP (vers. 5.0.1 using CPLEX 12.8 as LP solver + IPOPT 3.12.9 as nonlinear solver)
  - ✓ **basic**: our branch-and-cut algorithm <u>without</u> intersection cuts
  - ✓ **with-IC**: intersection cuts separated at each node where the LP solution is integral

- Single-thread runs (parallel runs not allowed in SCIP) with a time limit of **1 hour** on a standard PC Intel @ 3.10 GHz with 16 GB ram

- **Testbed**: all quadratic instances in **MINLPlib** (700+ instances) …

  … but some instances removed as root LP was **unbounded**

  → **620** instances left, **408** of which solved by all methods in 1 hour

Results

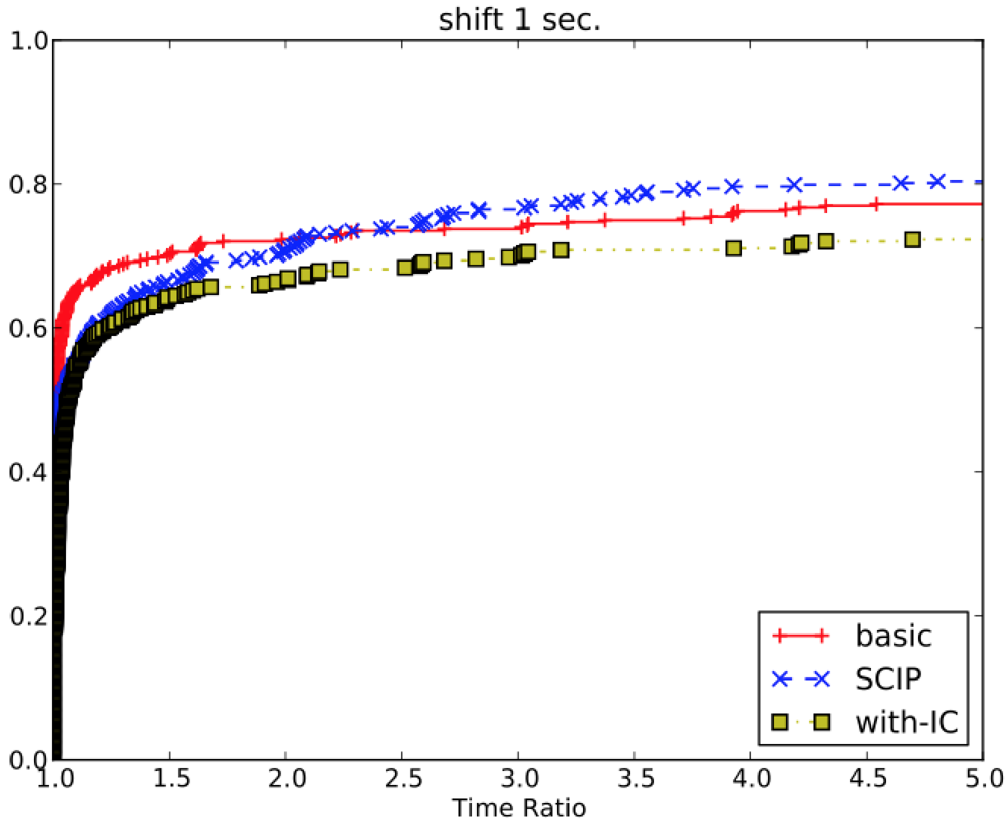


Figure 1: Performance profile comparison of **basic**, **SCIP** and **with-IC**, on the 408 MINLPlib instances that could be solved by at least one method in the 1-hour time limit (time shift of 1 sec.)

Results (without small instances)

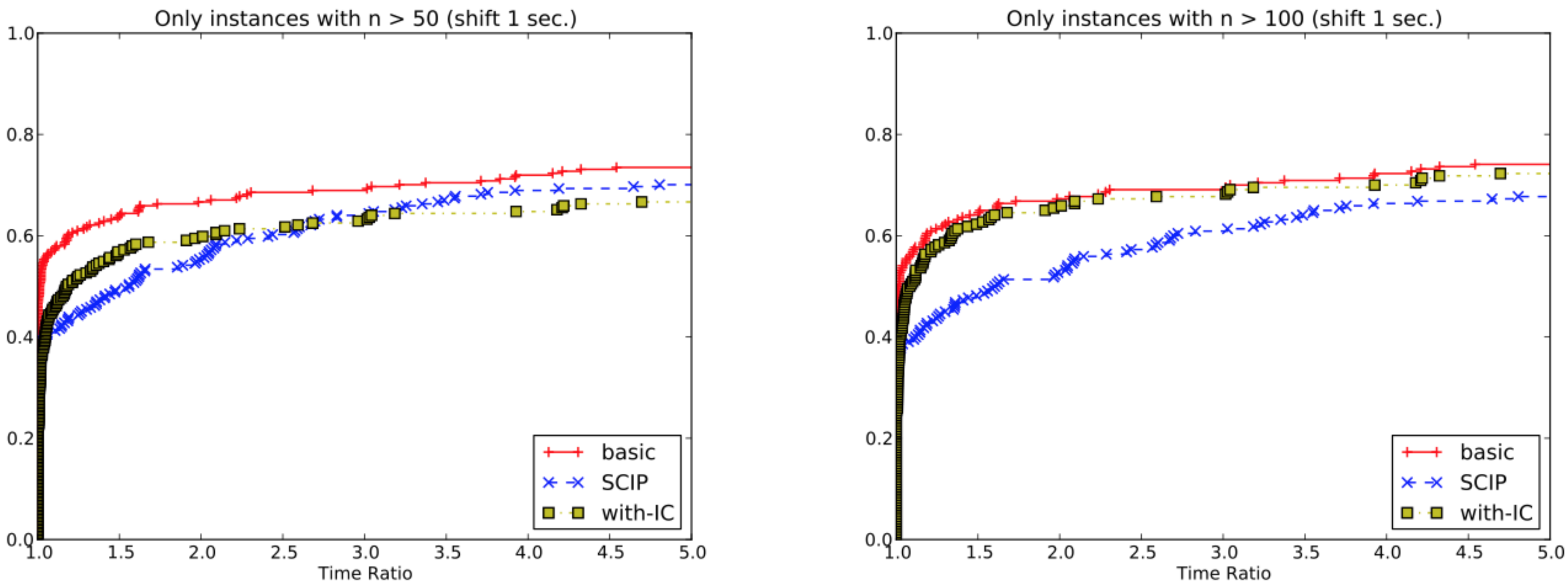


Figure 3: Performance profile comparison of **basic**, **SCIP** and **with-IC** as in Figure 1, when small instances are removed (time shift of 1 sec.)

ICs can make a difference!

Instance	SCIP	basic	with-IC
blend531	234.21	3600.00	31.05
crudeoil_lee4_09	89.12	9.83	2.21
portfol_classical050_1	57.03	54.37	33.26
powerflow0009r	3600.00	3600.00	969.12
powerflow0014r	3600.00	3600.00	302.77
sporttournament14	3600.00	182.41	125.50
squfl015-080	3600.00	238.53	137.32
squfl025-030	3600.00	44.46	18.72
turkey	61.19	3600.00	0.11

Table 4: Selected instances for which adding intersection cuts is highly beneficial.

Thanks for your attention!

Paper available at http://www.dei.unipd.it/~fisch/papers/

Slides available at http://www.dei.unipd.it/~fisch/papers/slides/

.