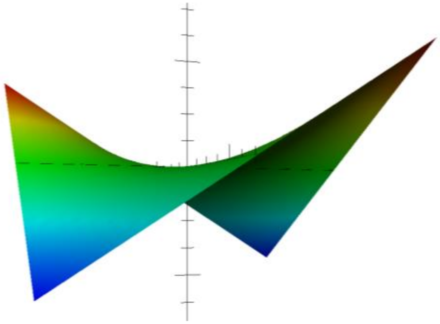


A Tighter Piecewise McCormick Relaxation for Bilinear Problems

Pedro M. Castro

Centro de Investigação Operacional
Faculdade de Ciências
Universidade de Lisboa

IF INVESTIGADOR FCT

LISBOA UNIVERSIDADE DE LISBOA

Problem definition (NLP)

- Bilinear program

(P)

$$\min f_0(x)$$

$$f_q(x) \leq 0 \quad q \in Q\backslash\{0\}$$

$$f_q(x) = \sum_{(i,j)\in BL_q} a_{ijq} x_i x_j + h_q(x) \quad q \in Q$$

$$0 \leq x^L \leq x \leq x^U$$

$$x \in \mathbb{R}^m$$

Relaxation provides a lower bound

a_{ijq} - parameters
$\boldsymbol{x}$ – vector of continuous variables
BL_q - (i,j) index set
$i \neq j$ – strictly bilinear problems
$i = j$ – can be allowed (quadratic problems)
$h_q(x)$- linear function in x

Introduction

- Bilinear problems occur in a variety of applications
  - Process network problems (Quesada & Grossmann, 1995)
  - Water networks (Bagajewicz, 2000)
  - Pooling and blending (Haverly, 1978)
- Non-convex, leading to multiple local solutions
  - Gradient based solvers unable to certify optimality
- Need for global optimization algorithms
  - What do they have in common?
    - Linear (LP) or mixed-integer linear (MILP) relaxation of (**P**) → LB
      - LP: Standard McCormick envelopes (1976)
      - MILP: Piecewise McCormick envelopes (Bergamini et al. 2005)
      - MILP: Multiparametric disaggregation (Teles et al. 2013; Kolodziej et al. 2013)
    - Solution of (**P**) with fast local solver → UB
      - Using LB as starting point
- Tight relaxation critical to ensure convergence
  - Relative optimality gap (UB-LB)/LB<ε

Piecewise McCormick relaxation $w_{ij} = x_i x_j$

- Domain of divided into N uniform partitions

x_j^L $n = 1$ $x_{j,1}^U$ $n = 2$ $x_{j,2}^U$ $\cdots$ $x_{j,N-1}^U$ $n = N$ $x_{j,N}^U$

$x_{j,1}^L$ $x_{j,2}^L$ $x_{j,N}^L$ x_j^U

$$y_{j,1} = true \quad \underline{\vee} \quad y_{j,2} = true \quad \underline{\vee} \quad \cdots \quad \underline{\vee} \quad y_{j,N} = true$$

Optimization will pick single partition

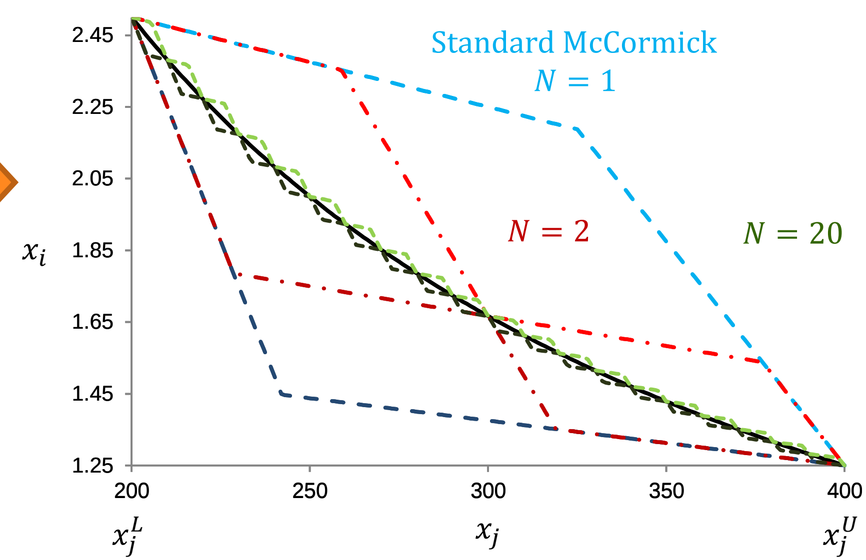
Generalized Disjunctive Program
(Balas, 1979; Raman & Grossmann, 1994)

$$\underline{\bigvee_n} \begin{bmatrix} y_{jn} \\ w_{ij} \geq x_i \cdot x_{jn}^L + x_i^L \cdot x_j - x_i^L \cdot x_{jn}^L \\ w_{ij} \geq x_i \cdot x_{jn}^U + x_i^U \cdot x_j - x_i^U \cdot x_{jn}^U \\ w_{ij} \leq x_i \cdot x_{jn}^L + x_i^U \cdot x_j - x_i^U \cdot x_{jn}^L \\ w_{ij} \leq x_i \cdot x_{jn}^U + x_i^L \cdot x_j - x_i^L \cdot x_{jn}^U \\ x_{jn}^L \leq x_j \leq x_{jn}^U \end{bmatrix}$$

$$x_i^L \leq x_i \leq x_i^U$$
$$x_{jn}^L = x_j^L + (x_j^U - x_j^L) \cdot (n-1)/N$$
$$x_{jn}^U = x_j^L + (x_j^U - x_j^L) \cdot n/N$$

Convex hull

(PR-MILP)



New tighter piecewise relaxation

- Basic idea
  - Use partition-dependent bounds also for variable x_i
    - x_{ijn}^L / x_{ijn}^U lower/upper bound when x_j constrained to n
  - Better bounds $\Rightarrow$ tighter relaxation $\Leftrightarrow$ lower gap

$$\bigvee_n \left[\begin{array}{c} y_{jn} \\ w_{ij} \geq x_i \cdot x_{jn}^L + x_{ijn}^L \cdot x_j - x_{ijn}^L \cdot x_{jn}^L \\ w_{ij} \geq x_i \cdot x_{jn}^U + x_{ijn}^U \cdot x_j - x_{ijn}^U \cdot x_{jn}^U \\ w_{ij} \leq x_i \cdot x_{jn}^L + x_{ijn}^U \cdot x_j - x_{ijn}^U \cdot x_{jn}^L \\ w_{ij} \leq x_i \cdot x_{jn}^U + x_{ijn}^L \cdot x_j - x_{ijn}^L \cdot x_{jn}^U \\ x_{ijn}^L \leq x_i \leq x_{ijn}^U \\ x_{jn}^L \leq x_j \leq x_{jn}^U \end{array} \right]$$

$$x_{jn}^L = x_j^L + (x_j^U - x_j^L) \cdot (n-1)/N$$
$$x_{jn}^U = x_j^L + (x_j^U - x_j^L) \cdot n/N$$

Convex hull

(PRT-MILP)

$$\min z^R = f_0(x) = \sum_{(i,j) \in BL} a_{ij0} w_{ij} + h_0(x)$$

$$f_q(x) = \sum_{(i,j) \in BL} a_{ijq} w_{ij} + h_q(x) \leq 0 \quad \forall q \in Q \backslash \{0\}$$

$$\left. \begin{array}{l} w_{ij} \geq \sum_n \left(\hat{x}_{ijn} \cdot x_{jn}^L + x_{ijn}^L \cdot \hat{x}_{jn} - x_{ijn}^L \cdot x_{jn}^L \cdot y_{jn} \right) \\ w_{ij} \geq \sum_n \left(\hat{x}_{ijn} \cdot x_{jn}^U + x_{ijn}^U \cdot \hat{x}_{jn} - x_{ijn}^U \cdot x_{jn}^U \cdot y_{jn} \right) \\ w_{ij} \leq \sum_n \left(\hat{x}_{ijn} \cdot x_{jn}^L + x_{ijn}^U \cdot \hat{x}_{jn} - x_{ijn}^U \cdot x_{jn}^L \cdot y_{jn} \right) \\ w_{ij} \leq \sum_n \left(\hat{x}_{ijn} \cdot x_{jn}^U + x_{ijn}^L \cdot \hat{x}_{jn} - x_{ijn}^L \cdot x_{jn}^U \cdot y_{jn} \right) \\ x_i = \sum_n \hat{x}_{ijn} \end{array} \right\} \forall (i,j)$$

$$\left. \begin{array}{l} x_j = \sum_n \hat{x}_{jn} \\ \sum_n y_{jn} = 1 \end{array} \right\} \forall \{j | (i,j) \in BL\}$$

$$\left. \begin{array}{l} x_{jn}^L = x_j^L + \dfrac{(x_j^U - x_j^L) \cdot (n-1)}{N} \\ x_{jn}^U = x_j^L + \dfrac{(x_j^U - x_j^L) \cdot n}{N} \\ x_{jn}^L \cdot y_{jn} \leq \hat{x}_{jn} \leq x_{jn}^U \cdot y_{jn} \end{array} \right\} \forall \{j | (i,j) \in BL\}, n$$

$$x_{ijn}^L \cdot y_{jn} \leq \hat{x}_{ijn} \leq x_{ijn}^U \cdot y_{jn} \quad \forall (i,j) \in BL, n \in \{1, \ldots, N\}$$

How to generate bounds x_{ijn}^L & x_{ijn}^U?

- Optimality bound contraction with McCormick envelopes
  - Solve multiple instances of LP problem (**BC**), two per:
    - Non-partitioned variable x_i
    - Partitioned variable x_j
    - Partition n
  - Focus on regions that can actually improve current UB
    - z' obtained from solving (**P**) with local solver
  - (**BC**) infeasible? Remove partition n^* of j^* from (**PRT-MILP**)
  - Computation can be time consuming

(**BC**)

$$x_{i^*j^*n^*}^L := \min x_{i^*} \quad (x_{i^*j^*n^*}^U := \max x_{i^*})$$

$$f_0(x) = \sum_{(i,j)\in BL_0} a_{ij0} w_{ij} + h_0(x) \le z'$$

$$f_q(x) = \sum_{(i,j)\in BL_q} a_{ijq} w_{ij} + h_q(x) \le 0 \quad \forall q \in Q\backslash\{0\}$$

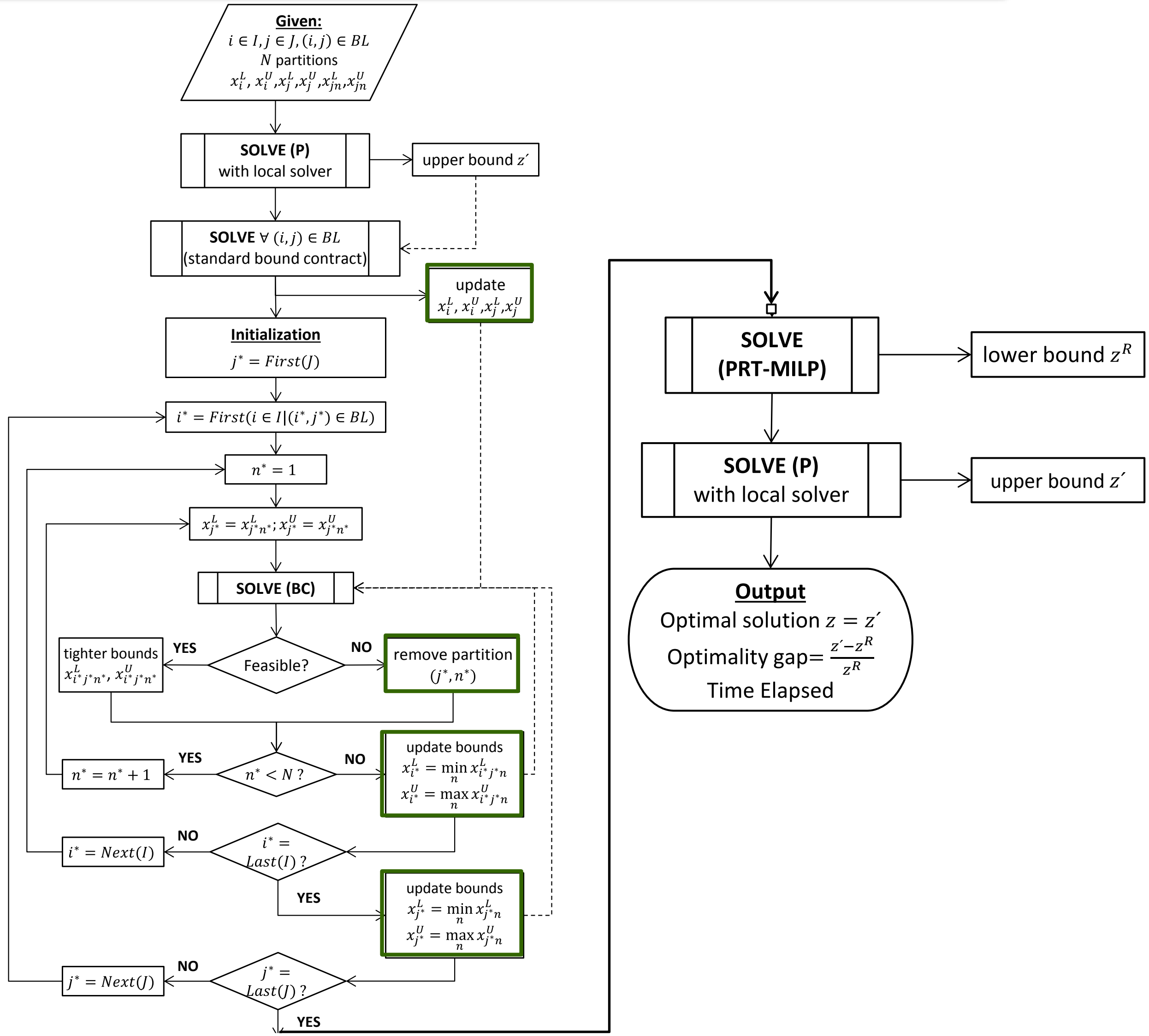
$$\left.\begin{aligned} w_{ij} &\ge x_i \cdot x_j^L + x_i^L \cdot x_j - x_i^L \cdot x_j^L \\ w_{ij} &\ge x_i \cdot x_j^U + x_i^U \cdot x_j - x_i^U \cdot x_j^U \\ w_{ij} &\le x_i \cdot x_j^L + x_i^U \cdot x_j - x_i^U \cdot x_j^L \\ w_{ij} &\le x_i \cdot x_j^U + x_i^L \cdot x_j - x_i^L \cdot x_j^U \end{aligned}\right\} \forall\,(i,j) \in BL$$

$$x_{j^*}^L = x_{j^*n^*}^L \le x_{j^*} \le x_{j^*n^*}^U = x_{j^*}^U$$

$$x^L \le x \le x^U$$

New optimization algorithm

- Key features
  - User selects:
    - N partitions
    - Partitioned variables x_j
  - Preliminary bound contraction stage
  - Bounds updated at different levels
  - LB from MILP relaxation
  - UB from NLP (single starting point)
  - Computes an optimality gap



Illustrative example P1

$$\min f_0(x) = -x_1 + x_1 x_2 - x_2$$

$$-6x_1 + 8x_2 \leq 3$$
$$3x_1 - x_2 \leq 3$$
$$0 \leq x_1, x_2 \leq 1.5$$

Partitioned variable: x_1

Non-partitioned variable: x_2

$BL = \{(2,1)\}$

Standard optimality bound contraction

$$0.357143 \leq x_1 \leq 1.375$$

$$0 \leq x_2 \leq 1.26$$

Global optimum

$$z = f_0(x) = -1.083333$$

	Gap
(**PR-MILP**) relaxation	
$z^R = -1.185207$	8.60%
(**PRT-MILP**) relaxation	
$z^R = -1.169619$	7.38%

Partition-dependent bound contraction $N = 3$

Partition n	x_{1n}^L	x_{1n}^U	x_{21n}^L	x_{21n}^U
1	0.357143	0.696429	LP is infeasible	
2	0.696429	1.035714	0	1.137609
3	1.035714	1.375	0.107143	1.195150

How about using more partitions?

- Up to two orders of magnitude reduction in optimality gap
  - Major increase in total computational time
  - Still, new approach performs best!

P1

Relaxation	Partitions (N)	15	150	1500	7500
(PR-MILP)	Total CPUs	0.93	1.60	3.63	302
	Optimality gap	1.67%	0.188%	0.0189%	0.0038%
(PRT-MILP)	Optimality gap	0.77%	0.010%	0.0002%	
	Total CPUs	3.63	27.8	267	

P2

$$\min\ 6x_1^2 + 4x_2^2 - 2.5x_1x_2$$

$$x_1x_2 - 8 \geq 0$$

$$1 \leq x_1, x_2 \leq 10$$

Relaxation	Partitions (N)	9	90	900	2700
(PR-MILP)	Total CPUs	0.83	1.2	7.76	478
	Optimality gap	8.41%	0.84%	0.0830%	0.0278%
(PRT-MILP)	Optimality gap	0.84%	0.02%	0.0002%	
	Total CPUs	3.33	24.0	223	

Results for larger test problems

- 16 easiest water-using design problems
  - New (**PRT-MILP**) approach leads to lower gaps at termination (3600 CPUs)
  - Faster in 56% of the cases (finest partition level)
  - Single failure when finding best-known solution for (**P**)

Subset of problems selected for illustration

	Number of variables			(PR-MILP)			(PRT-MILP)		
	Non-partitioned $\|I\|$	Partitioned $\|J\|$	Partitions (N)	Solution	Total CPUs	Gap	Gap	Total CPUs	Solution
Ex2	20	16	10	74.4699	18.8	0.1250%	0.0484%	141	74.4699
			50		3609	0.1245%	0.0103%	1536	
Ex6	30	25	10	142.0816	44.7	0.0435%	0.0110%	205	142.0816
			50		3614	0.1379%	0.0041%	1090	
Ex8	30	20	10	164.4898	18.8	0.7134%	0.2132%	207	164.4898
			100		3613	1.9508%	0.0130%	1803	
Ex10	15	9	10	169.1173	6.69	0.0160%	0.0008%	45.3	169.1173
			100		809	0.0023%	0.0000%	323	
Ex14	32	24	10	329.9689	282	1.2259%	1.0499%	447	329.5698
			20	329.9566	3615	2.3368%	0.9206%	4270	
Ex15	40	30	10	361.5177	231	0.7963%	0.7427%	723	361.5177
			20	361.6856	3618	0.9361%	0.7779%	4486	361.6856
Ex16	24	16	10	285.9343	17.4	1.1105%	0.8861%	212	285.9343
			50		3610	1.5755%	0.0526%	2042	

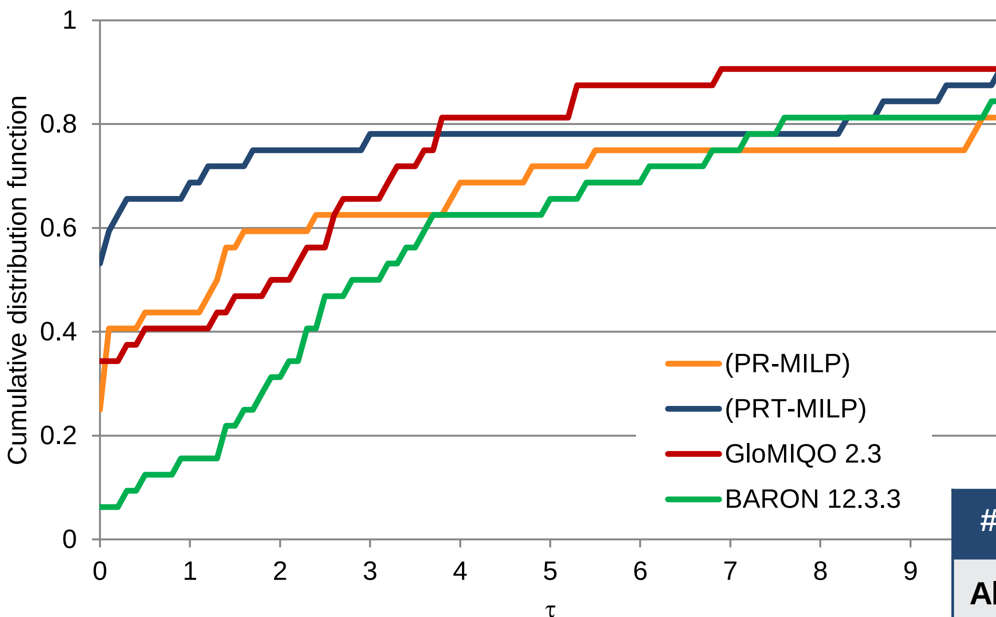
How to determine the number of partitions?

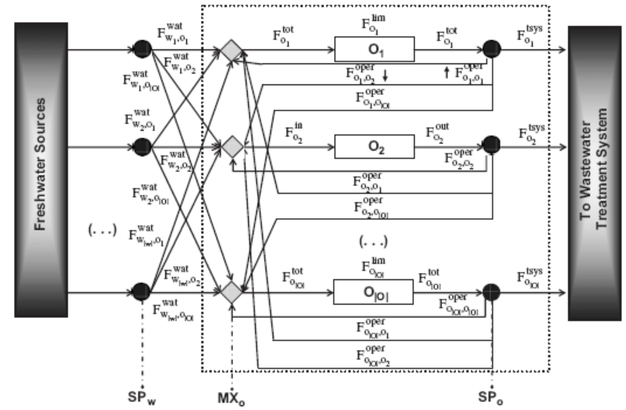
- Quality of relaxation increases with N
  - Careful not to generate intractable MILPs (Gounaris et al. 2009)
- Several works identify most appropriate value for a particular problem
  (Misener et al. 2011; Wittmann-Holhbein and Pistikopoulos, 2013; 2014)
  - No method to estimate N as a function of problem complexity
  - Needed to provide a fair comparison with commercial solvers
- Proposed formula meets important criteria:
  - Returns an integer value
  - Ensures minimum of two partitions
    - So as to benefit from piecewise relaxation scheme
  - Reduces N with increase in problem size
    - Roughly inversely proportional (r^2=0.76)

$$N = 1 + \left\lceil \frac{\alpha}{|I| \cdot |J|} \right\rceil \qquad \alpha = 1.8E4$$

Comparison to commercial solvers

- Performance profiles
  (Dolan & Moré, 2002)
  – KPI: Optimality gap





- (**PRT-MILP**) better (**PR-MILP**)
- GloMIQO outperforms BARON

GAMS 24.1.3, CPLEX 12.5.1, GloMIQO 2.3,
BARON 12.3.3, Intel Core i7-3770 (3.07 GHz),
8 GB RAM, Windows 7 64-bit
Termination criteria: gap=0.0001%, Time=3600 CPUs

 # Failures finding best-known solution (34 problems)

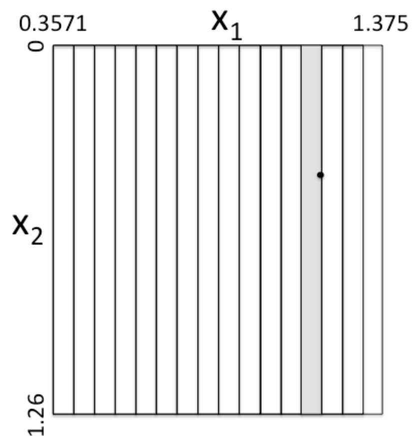
Algorithm	GloMIQO	(PRT-MILP)	BARON	(PR-MILP)
Suboptimal	3	5	2	7
No solution	0	0	4	0
Total	**3**	**5**	**6**	**7**

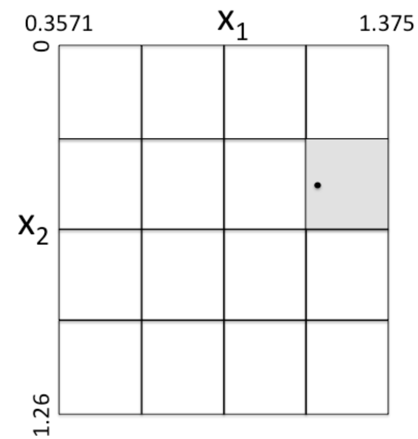
Why not bivariate partitioning?

- Domain of x_i and x_j known a priori for each 2-D partition
  - Slightly better performance than univariate partitioning for P1
  - Gap for 125^2 partitions one order of magnitude larger vs. (PRT-MILP)

(PR-BV-MILP)

$$\bigvee_{n=1}^{N}\bigvee_{n'=1}^{N}\begin{bmatrix} y_{injn'} \\ w_{ij} \geq x_i \cdot x_{jn'}^{L} + x_{in}^{L} \cdot x_j - x_{in}^{L} \cdot x_{jn'}^{L} \\ w_{ij} \geq x_i \cdot x_{jn'}^{U} + x_{in}^{U} \cdot x_j - x_{in}^{U} \cdot x_{jn'}^{U} \\ w_{ij} \leq x_i \cdot x_{jn'}^{L} + x_{in}^{U} \cdot x_j - x_{in}^{U} \cdot x_{jn'}^{L} \\ w_{ij} \leq x_i \cdot x_{jn'}^{U} + x_{in}^{L} \cdot x_j - x_{in}^{L} \cdot x_{jn'}^{U} \\ x_{in}^{L} \leq x_i \leq x_{in}^{U} \\ x_{jn'}^{L} \leq x_j \leq x_{jn'}^{U} \end{bmatrix} \quad \forall (i,j)$$





Relaxation	Partitions (N)	16	144	2500	10000	
Univariate (PR-MILP)	Total CPUs	0.88	1.42	9.11	429	
	Optimality gap	1.70%	0.197%	0.0114%	0.0029%	
	N	**4**	**12**	**50**	**100**	**125**
Bivariate (PR-BV-MILP)	Optimality gap	1.28%	0.188%	0.0078%	0.0029%	0.0018%
	Total CPUs	0.93	0.91	7.55	217	469

Conclusions

- Bilinear problems tackled through piecewise relaxation (PR)
- Novel approach with partition-dependent bounds for all bilinear variables
  - Significant improvement in relaxation quality
  - Need for extensive optimality-based bound contraction (BC)
    - Total computational time is sometimes lower
- New algorithm outperforms state-of-the-art global solvers
  - Specific class of water-using network design problems
- PR and BC schemes should be used to greater extent
  - Integrating with spatial B&B subject of future work

- See CACE paper for further details

- Acknowledgments:
  - Luso-American Foundation (2013 Portugal-U.S. Research Networks)
  - Fundação para a Ciência e Tecnologia (Investigador FCT 2013)