IBM Analytics

<Ed Klotz>

<October 15, 2017>

Performance Tuning for Cplex's Spatial Branch-and-Bound Solver for Global Nonconvex (Mixed Integer) Quadratic Programs

Objective

- Performance Tuning for MILP, convex MIQP has some well established concepts that work effectively
- Spatial B&B algorithm has some fundamental differences that create different performance tuning challenges
- Construct a set of performance tuning tactics for nonconvex (MI)QP for spatial B&B algorithm
  - Distinguish tactics that extend from MILP and convex MIQP cases from those that don't
  - Consider tactics specific to nonconvex (MI)QP

Outline

- Nonconvex (MI)QP fundamentals
- Very brief review of MILP/convex MIQP tuning tactics
- Parameters to set (or leave alone)
  - Parameters that extend from MILP/convex MIQP
  - Parameters that don't extend well
  - Parameters specific to nonconvex (MI)QP
- Nonconvex (MI)QP model categories
- Examples
- Conclusions

Nonconvex (MI)QP fundamentals

Problem definition

$$\text{Minimize} \quad c^T x + x^T Q x$$

$$\text{Subject to} \quad Ax = b$$

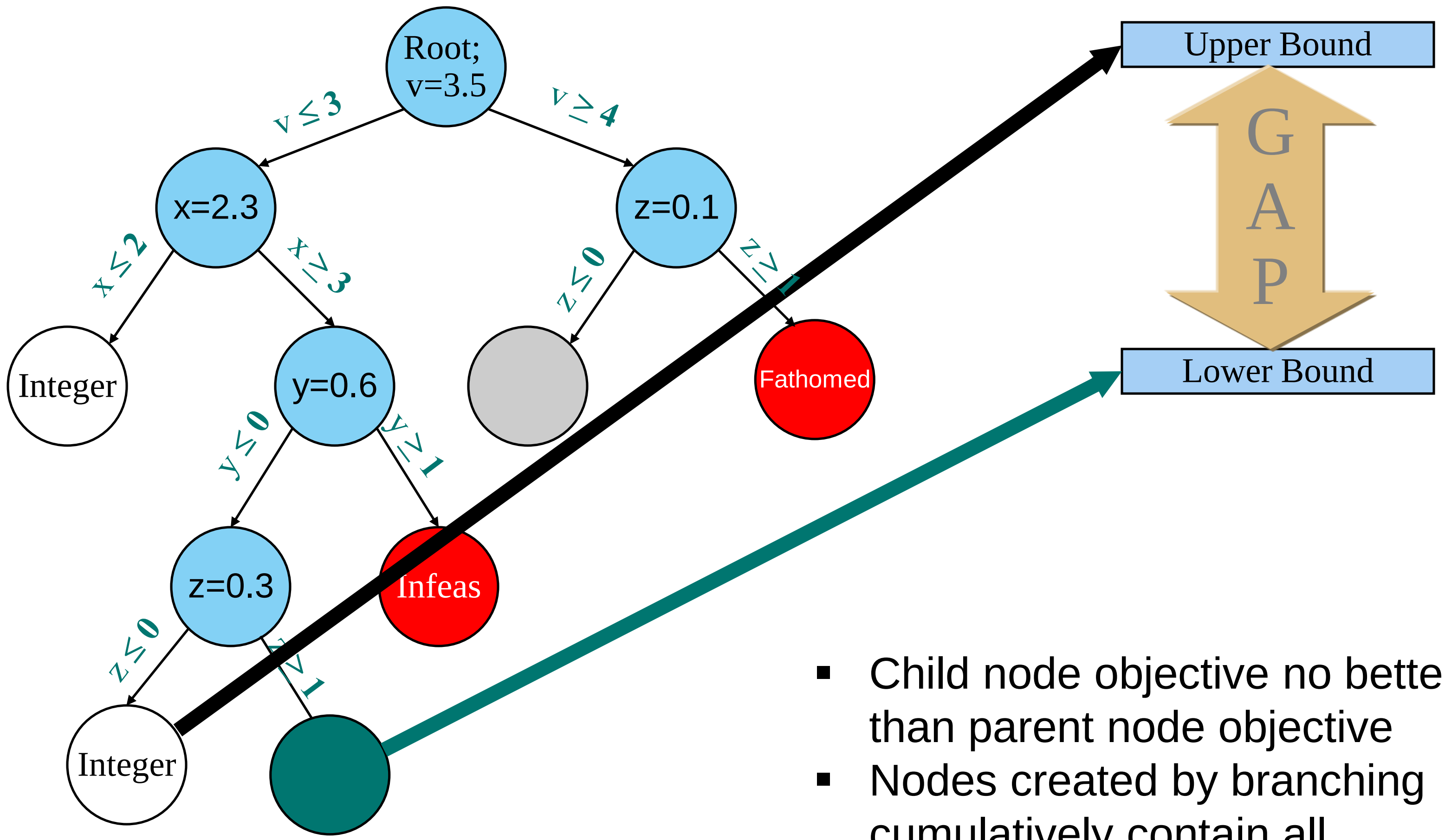
$$l \le x \le u$$

$$Q \text{ indefinite}$$

$$x_S \in Z^{|S|}$$

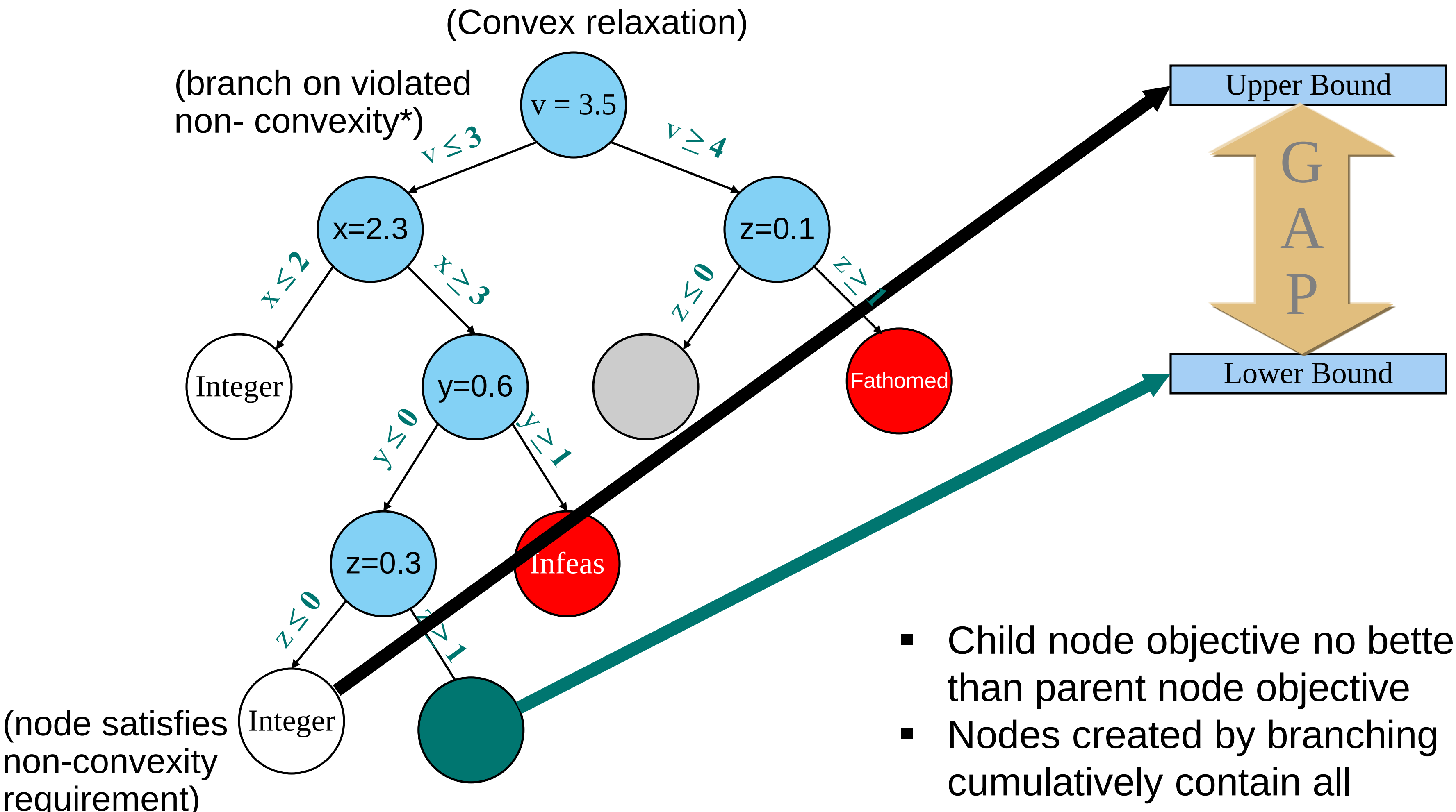
- S=∅: nonconvex QP; otherwise nonconvex MIQP
- Either way, we’ll solve it by branching, but presence of integer variables may affect tuning tactics

Branch and Bound for MILP



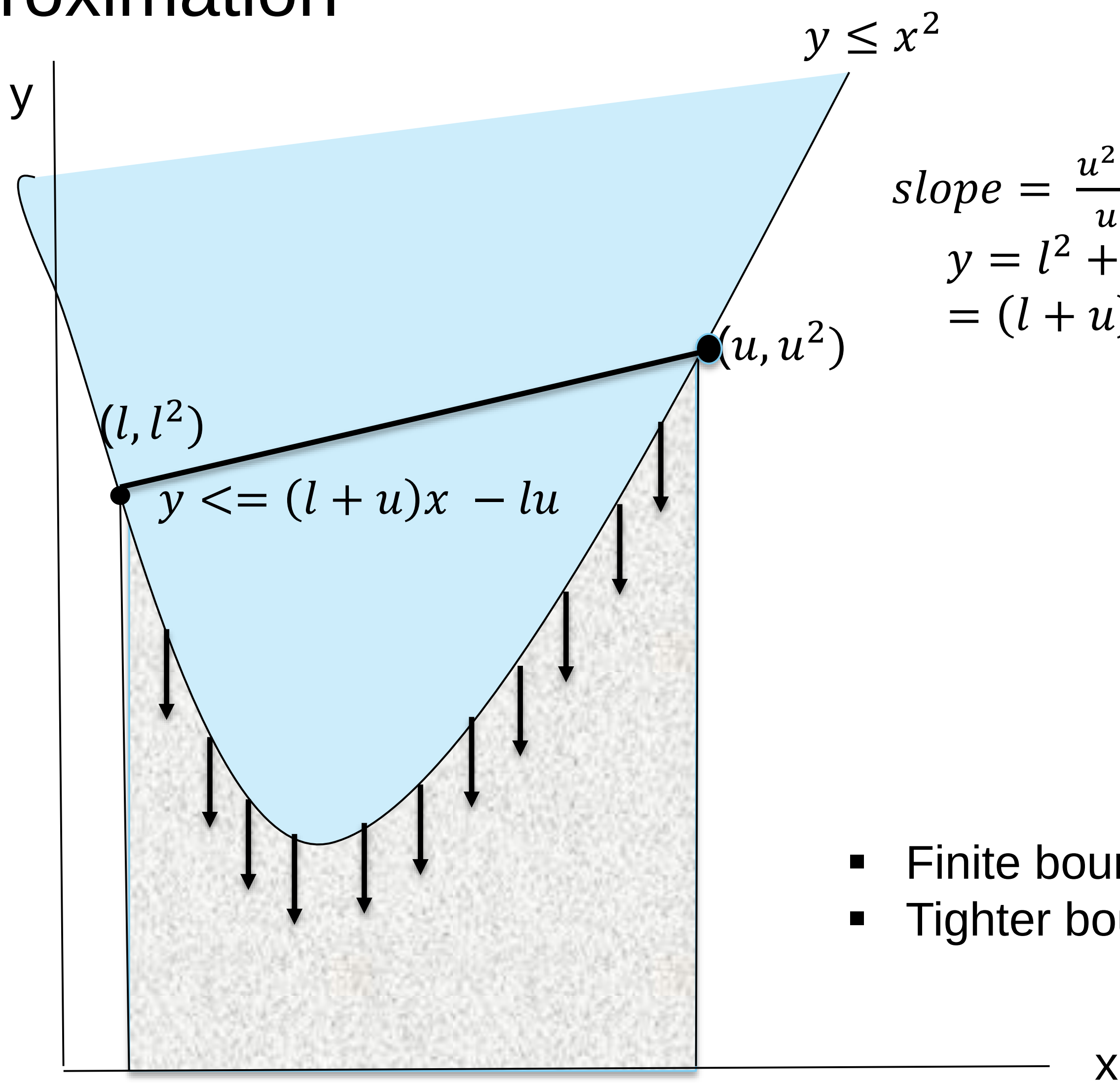
- Child node objective no better than parent node objective
- Nodes created by branching cumulatively contain all feasible solutions.

Branch and Bound for Nonconvex problems



- Child node objective no better than parent node objective
- Nodes created by branching cumulatively contain all feasible solutions.

Secant approximation



$$slope = \frac{u^2 - l^2}{u - l} = u + l$$

$$y = l^2 + (u + l)(x - l)$$
$$= (l + u)x - lu$$

- Finite bounds essential
- Tighter bound better

McCormick relaxation (algebraic interpretation)

$$x_1 x_2$$

$$x_1 \in [l_1, u_1], x_2 \in [l_2, u_2]$$

$$\overbrace{(x_1 - l_1)}^{\geq 0}\ \overbrace{(x_2 - l_2)}^{\geq 0} = x_1 x_2 - l_1 x_2 - l_2 x_1 + l_1 l_2 \geq 0$$

$$\rightarrow x_1 x_2 \geq l_1 x_2 + l_2 x_1 - l_1 l_2$$

$$(u_1 - x_1)\ (u_2 - x_2) = x_1 x_2 - u_1 x_2 - u_2 x_1 + u_1 u_2 \geq 0$$

$$\rightarrow x_1 x_2 \geq u_1 x_2 + u_2 x_1 - u_1 u_2$$

$$\rightarrow x_1 x_2 \geq \max(l_1 x_2 + l_2 x_1 - l_1 l_2, u_1 x_2 + u_2 x_1 - u_1 u_2)$$

$$(x_1 - l_1)\ (u_2 - x_2) = u_2 x_1 + l_1 x_2 - l_1 u_2 - x_1 x_2 \geq 0$$

$$\rightarrow x_1 x_2 \leq u_2 x_1 + l_1 x_2 - l_1 u_2$$

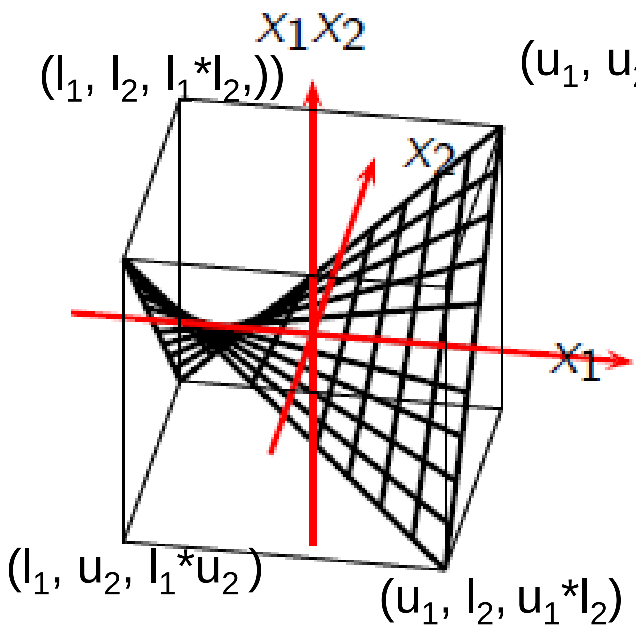
$$(u_1 - x_1)\ (x_2 - l_2) = u_1 x_2 + l_2 x_1 - u_1 l_2 - x_1 x_2 \geq 0$$

$$\rightarrow x_1 x_2 \leq u_1 x_2 + l_2 x_1 - u_1 l_2$$

$$\rightarrow x_1 x_2 \leq \min(u_2 x_1 + l_1 x_2 - l_1 u_2, u_1 x_2 + l_2 x_1 - u_1 l_2)$$

Source: https://optimization.mccormick.northwestern.edu/index.php/McCormick_envelopes

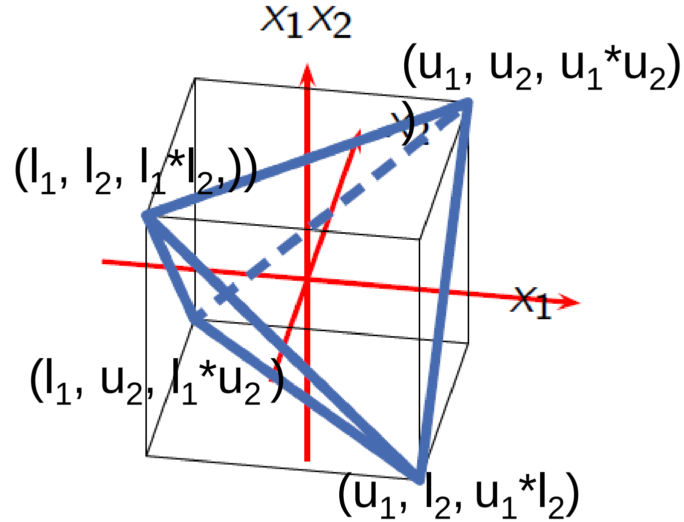
McCormick relaxation (geometric interpretation)



The convex hull relaxations of a single product $x_1 x_2$ [McCormick, 1976]

$$x_1 x_2 \geq y_{12}^{-} := \max \left\{ \begin{array}{r} u_2 x_1 + u_1 x_2 - u_1 u_2 \\ l_2 x_1 + l_1 x_2 - l_1 l_2 \end{array} \right\}$$

$$x_1 x_2 \leq y_{12}^{+} := \min \left\{ \begin{array}{r} u_2 x_1 + l_1 x_2 - l_1 u_2 \\ l_2 x_1 + u_1 x_2 - u_1 l_2 \end{array} \right\}$$



- Depending on the sign of q_{ij} we only need y^+ or y^-.
- For simplicity, we assume we put all in the remainder.

Nonconvex (MI)QP fundamentals

- We've defined some relaxation methods, but how do we use them, and how do we branch?

$$Minimize \quad c^T x + x^T Qx$$

$$< \text{constraints} >$$

$$Minimize \ldots + q_{ii} x_i^2 + q_{ij} x_i x_j + \ldots \Leftrightarrow$$

$$Minimize \ldots + q_{ii} y_i + q_{ij} y_{ij} + \ldots$$

$$y_i - x_i^2 = 0$$ (Secant Approximation)

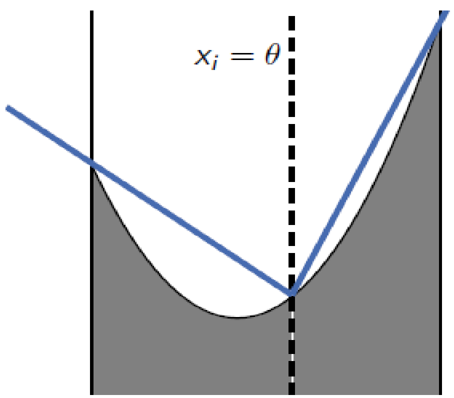
$$y_{ij} - x_i x_j = 0$$ (McCormick Relaxation)

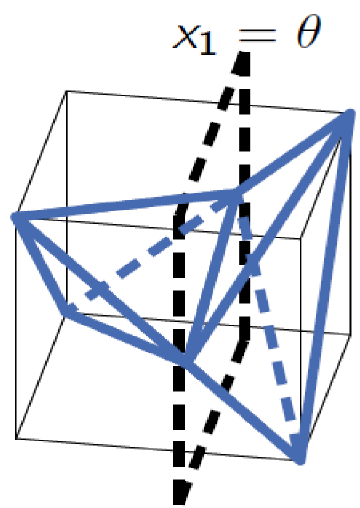
$$< \text{constraints} >$$ (Regular branching on IRs)

Do only for nonconvex parts of Q

Branching

- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick such an index i, choose a value θ between $\frac{l_i + u_i}{2}$ and $\overline{x}_i$.
- Branch by changing the bound to θ and updating all Secant and McCormick approximations involving this bound.





Review of MILP/Convex MIQP Tuning Tactics

- MILP
  - Examine node log for source(s) of performance problems
    - Is lack of progress in best integer, best node or both?
  - Choose parameters based on source(s) of problems
  - Additional insights obtained by looking at model to either select parameters or tighten the formulation
- Convex MIQP
  - Similar to MILP, with a few exceptions and additions
    - Node relaxations need not have vertex solutions, potentially rendering some cuts ineffective
    - Solve models with alternate objectives to find good starting solutions
      - Linear term only, linear approximation to quadratic objective, simpler quadratic objective

Nonconvex (MI)QP Tuning Tactics

- Distinguish among 3 Different Problem Types
  - All quadratic objective terms have at ≥ 1 integer variable
    - Can linearize the objective
      - CPLEX does so if at least one variable is binary
      - Otherwise can express general integer as linear combination of binaries
      - Can instead convexify the objective if all QP objective terms have pairs of binaries
    - MILP or convex MIQP instead of nonconvex MIQP
      - No spatial branching needed
      - But problem size is larger and/or more relaxed
    - Standard MILP performance tuning tactics apply
      - But certain parameters may be more or less effective for the linearization constraints

Nonconvex (MI)QP Tuning Tactics

- 3 Different Problem Types (ctd.)
  - Nonconvex QP, all continuous variables
    - Spatial branching only
    - Parameters that exploit integrality to improve performance will be ineffective
      - Probing, cuts, most heuristics (e.g. RINS, feasibility pump)
      - Concepts need to be extended from integrality restrictions to more general non-convexity restrictions
    - Parameters that tighten bounds more likely to help
      - Bound strengthening already aggressive
      - Can force node presolve on (but node probing won't help)
      - Try to provide tightest bounds possible

Nonconvex (MI)QP Tuning Tactics

- 3 Different Problem Types (ctd.)
  - Nonconvex MIQP that can't be completely linearized or convexified
    - Spatial branching and IR branching
      - Tactics depend on whether spatial or IR branching makes the model challenging
      - Consider related problems with same constraints but quadratic objective that only requires spatial or only requires IR branching
    - Parameters that exploit integrality to improve performance may help
      - Probing and heuristics
      - Most cuts
        - But not those that rely on simplex tableaus (e.g. Gomory)

Nonconvex (MI)QP Specific Parameters

- Problem type control
  - Qtolin parameter
    - Controls whether CPLEX linearizes bilinear terms with at least one binary variable
      - Tradeoff between easier MILP problem type and size of problem
  - Qpmakepsd parameter
    - Controls whether CPLEX convexifies the objective
      - Obtain convex MIQP without increase in problem size
      - But tends to yield weaker relaxation than linearization
      - CPLEX default performance improved by moving from convexification to linearization

Nonconvex (MI)QP Specific Parameters

Qtolin parameter:

```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
x1,x2,z12 binary

This linearization transforms the nonconvex MIQP into an
 MILP
```

Nonconvex (MI)QP Specific Parameters

Qpmakepsd parameter to convexify quadratic objective involving pairs of binaries:

Min x1*x2 + x2*x3

$$x1 \cdot x2 + x2 \cdot x3 = x1 \cdot x2 + x2 \cdot x3 + d1(x1^2 - x1) + d2(x2^2 - x2) + d3(x3^2 - x3)$$

$$= (d1x1^2 + d2x2^2 + d3x3^2 + x1 \cdot x2 + x2 \cdot x3) - d1x1 - d2x2 - d3x3$$

- **Can increase d1, d2, d3 to make quadratic terms convex**
  - **The bigger the increase, the weaker the relaxation**
  - **Fast heuristic to calculate good choices for d1, d2, d3**
- **Change nonconvex MIQP into convex MIQP**

Nonconvex (MI)QP Specific Parameters

- Cuts specific to nonconvex (MI)QP
  - BQP (Boolean Quadric Polytope) cuts
    - Aggressive settings more likely to be effective when number of linear constraints is modest
  - RLT (Relaxation Linearization Technique) cuts
    - More likely to be effective with more linear constraints
  - Either disable or use more aggressively
    - Tradeoff between improved progress in best node value, slower node throughput due to larger size of node relaxations

Transform arbitrary QP to Box QP

$$\text{Minimize} \quad c^T x + .5 x^T Q x$$
$$Ax \sim b$$
$$l \le x \le u$$

$$x' = x - l$$

$$\text{Minimize} \quad (c^T + l^T Q) x' + .5 x'^T Q x'$$
$$Ax' \sim b - Al$$
$$0 \le x' \le u - l$$

$$x'' = x' / (u - l)$$

$$\text{Minimize} \quad c''^T x'' + .5 x''^T Q'' x''$$
$$A'' x'' \sim b''$$
$$0 \le x'' \le 1$$

Box QP

$$\min \ \frac{1}{2} x^T Q\, x + \ c^T x$$

$$s.t. \qquad\qquad \text{(box-QP)}$$

$$0 \le x \le 1$$

- Box-QP is interesting in itself:
  - Bounds 0 and 1 are w.l.o.g. (every box QP can be scaled to those bounds)
  - Still NP-hard
  - Has some academic interest [Vandenbussche and Nemhauser, 2005, Burer and Vandenbussche, 2009, Chen and Burer, 2012]
  - Also some applications [Moré and Toraldo, 1989] (usually huge size)
- Assuming finite bounds on x, (box-QP) is a valid relaxation for every non-convex (MI)QP (but additional linear constraints are removed in the relaxation process)

Box QP and Boolean Quadratic Optimization

Proposition [Burer and Letchford, 2009]

Let Y^Q be the set where variables y represent the products in Q (excluding diagonal terms Q_{ii}):

$$E = \{(i,j) : i \neq j \text{ and } q_{ij} \neq 0\},$$
$$Y^Q = \{(x,Y) : y_{ij} = x_i x_j \; \forall \, (i,j) \in E\}.$$

We then have:

$$\text{BQP} := \text{conv}\left((x,Y) \in Y^Q : x \in \{0,1\}^n\right) = \text{conv}\left((x,Y) \in Y^Q : x \in [0,1]^n\right)$$

Corollary

- BQP is the Boolean Quadratic Polytope [Padberg, 1989].
- BQP gives a valid relaxation of box-QP (but tends to be weaker as number of linear constraints in original QP increases).
- Every valid cut for BQP is valid for the box-QP.

BQP cuts for non-convex (MI)QPs

[joint project with IBM Research (Günlük and Linderoth)]

- BQP can be rewritten as

$$\mathrm{BQP} := \mathrm{conv}\Big((x, Y) \in \{0,1\}^{n+|E|} : \\ y_{ij} \leq x_i,\ \ y_{ij} \leq x_j,\ \ y_{ij} \geq x_i + x_j - 1, \forall (i,j) \in E \Big)$$

- CPLEX has a lot of technology available to optimize over BQP
  - In particular, the 0 – 1/2 Chvátal-Gomory cut separator finds strong cutting planes for BQP
  - When used in the context of a non-convex MIQP, we call these BQP-cuts
  - By scaling and shifting, separation can always be reduced to the 0 – 1 case
  - After branching, rescaling using tighter local bounds leads to tighter local cuts.
  - BQP arise from Box-QP but can be used to strengthen any non-convex (MI)QP

RLT cuts for non-convex (MI)QPs

- Details
  - **A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems** by Hanif D. Sherali and W. P. Adams, Springer 1999
- Key idea: Multiply a linear constraint (ax-b) by individual $(x_j - l_j)$ values, then use McCormick inequalities in bilinear terms, update the lower and upper bounds in the linear expressions as they get tighter during the tree search to create locally valid cuts

$$x_1 x_2 \geq \max(l_1 x_2 + l_2 x_1 - l_1 l_2, u_1 x_2 + u_2 x_1 - u_1 u_2)$$

$$x_1 x_2 \leq \min(u_2 x_1 + l_1 x_2 - l_1 u_2, u_1 x_2 + l_2 x_1 - u_1 l_2)$$

  - More likely to be effective on models with more constraints

Nonconvex (MI)QP General Parameters

- Parameters that can help regardless of the category of nonconvex (MI)QP
  - Improve node throughput
    - Start/Subalgorithm parameters to chose node relaxation algorithm
- Parameters that only apply when integer branching is present
  - Variable selection parameter
  - Probing
- MIP starts not currently supported
  - Difficult mapping from original to transformed model

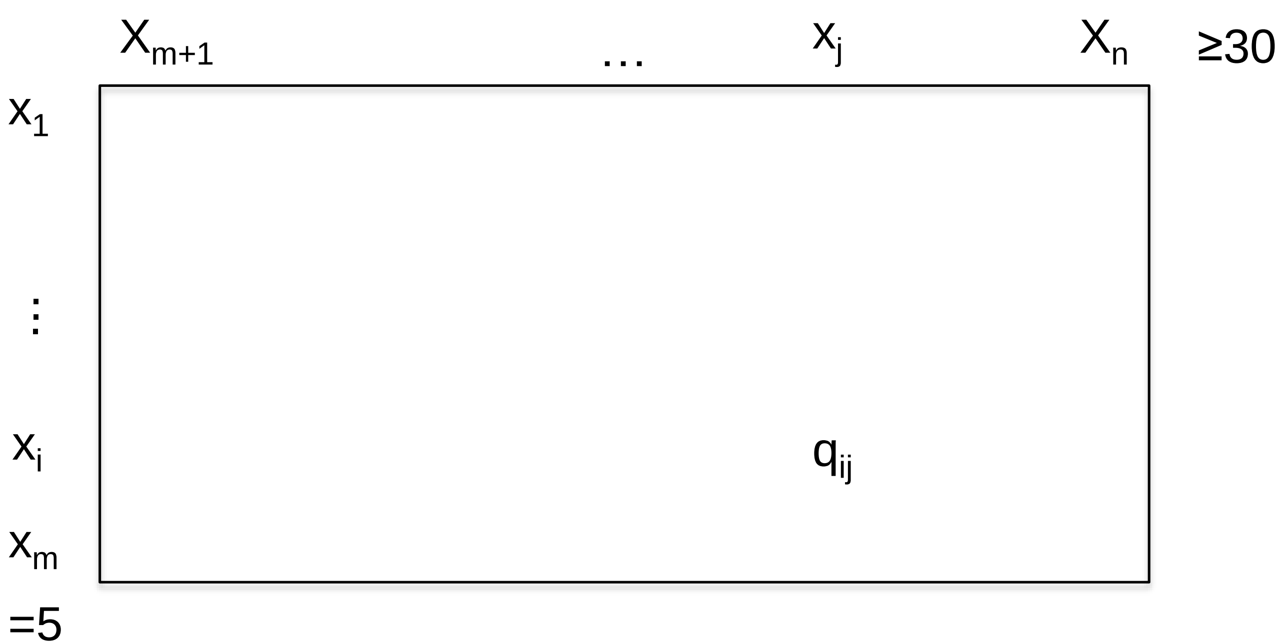
Example 1

- Nonconvex QP with all binaries and cardinality constraints:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j \qquad (x \text{ binary}, q_{ij} \geq 0)$$

$$\sum_{i=1}^{m} x_i = 5$$

$$\sum_{j=m+1}^{n} x_j \geq 30$$



Example 1

- Nonconvex QP with all binaries and cardinality constraints
- To linearize or not to linearize?

$$Minimize \quad \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j \qquad (x \text{ binary})$$

$$\sum_{i=1}^{m} x_i = 5$$

$$\sum_{j=m+1}^{n} x_j \geq 30$$

$$Minimize \quad z_{12} \qquad (x \text{ binary})$$

$$z_{12} \leq x_1$$

$$z_{12} \leq x_2$$

$$z_{12} \geq x_1 + x_2 - 1$$

$x_1 = x_2 = .5, z_{12} = 0$ feasible in the linearized LP relaxation

In original MIQP, $x_1 = x_2 = .5$ forces a positive objective value in the nonconvex QP relaxation, but not in the relaxations actually solved by the spatial B & B solver

Example 1

- Nonconvex QP with all binaries and cardinality constraints
- To linearize or not to linearize?
- Node log with qtolin parameter turned off:

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
	0	0	-7.12168e+08	109		-7.12168e+08	31	
*	0+	0			6204.9094	-7.12168e+08		---
*	0+	0			4967.7765	-7.12168e+08		---
	0	2	-7.12168e+08	109	4967.7765	-7.12168e+08	31	---
...								
	7901	5570	-6.94050e+08	0	1403.6893	-7.12137e+08	101618	---
	7902	5571	-6.87463e+08	6	1403.6893	-7.12137e+08	101623	---

Elapsed time = 712.68 sec. (540121.82 ticks, tree = 2402.25 MB, solutions = 267)

Nodefile size = 204.31 MB

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
	7942	5593	infeasible		1403.6893	-7.12137e+08	102200	---
	7950	5597	-6.87455e+08	3	1403.6893	-7.12137e+08	102377	---

Example 1

- Nonconvex QP with all binaries and cardinality constraints
- To linearize or not to linearize?
- Node log with qtolin parameter left on at default:

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
*	0+	0			494.8192	0.0000		100.00%
*	0+	0			131.3468	0.0000		100.00%
	0	0	0.0000	70	131.3468	0.0000	233	100.00%

...

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
	18233	12810	7.7423	178	17.0050	0.0000	2079862	100.00%
	18235	12812	1.7563	141	17.0050	0.0000	2080171	100.00%

Elapsed time = 6154.42 sec. (1368022.84 ticks, tree = 930.16 MB, solutions = 13)

	Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
*	18530+	13096			16.5953	0.0000		100.00%
*	18540+	13106			16.3024	0.0000		100.00%

Example 1

- Nonconvex QP with all binaries and cardinality constraints
- Node log based performance tuning
  - Linearization to MILP better
  - Progress in the best node may be challenging with parameter settings
    - Cuts and probing may not address the weakness of the formulation involving the linearization variables
      - Model only has cardinality and linearization constraints
      - Zero half cuts might help
  - Branching up more powerful when number of variables dramatically exceeds rhs values of cardinality constraints
      - Branching down more powerful in the reverse case
  - Model well suited to local search/local improvement heuristics

Example 1

- Nonconvex QP with all binaries and cardinality constraints
- Well suited to local search

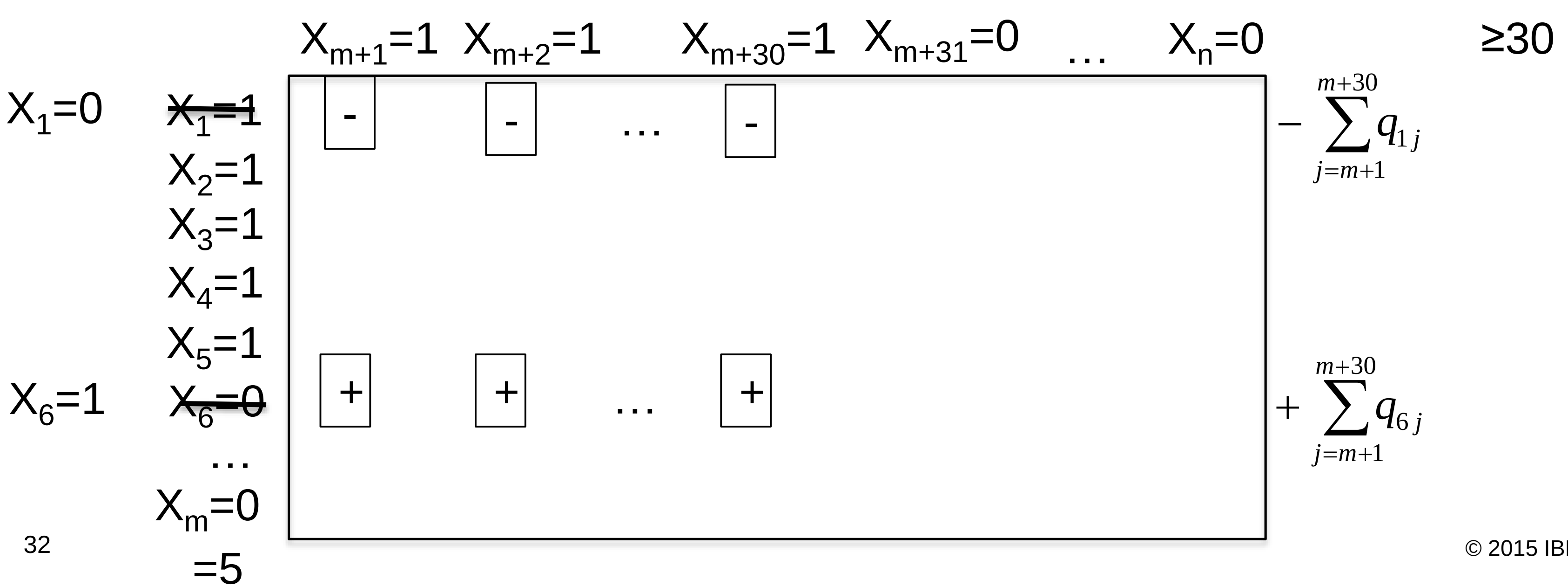
  local improvement:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j \qquad (x \text{ binary})$$

$$\sum_{i=1}^{m} x_i = 5$$

$$\sum_{j=m+1}^{n} x_j \geq 30$$

X_{m+1}



Example 1

- Nonconvex QP with all binaries and cardinality constraints
- Node log branch up, RINS frequency 200, solution polishing after 30 minutes:

Node	Nodes Left	Objective	IInf	Best Integer	Cuts/ Best Bound	ItCnt	Gap
…							
60	14	11.7385	251	11.9074	0.0000	16170	100.00%
70	14	11.7879	226	11.9074	0.0000	16292	100.00%
Elapsed time = 545.57 sec. (114453.69 ticks, tree = 0.53 MB, solutions = 6)							
79	13	cutoff		11.9074	0.0000	16407	100.00%
84	14	1.7133	173	11.9074	0.0000	16697	100.00%
…							
1708	124	cutoff		9.0886	0.0000	135562	100.00%
Elapsed time = 1696.83 sec. (240881.08 ticks, tree = 95.36 MB, solutions = 10)							
1760	127	0.0008	551	9.0886	0.0000	142195	100.00%
…							
* 1782+	118			6.8981	0.0000		100.00%
1782	120	0.3247	115	6.8981	0.0000	159376	100.00%
1783	119	0.2436	560	6.8981	0.0000	159183	100.00%
Elapsed time = 4391.19 sec. (770332.08 ticks, tree = 166.33 MB, solutions = 38)							

Example 2

- Nonconvex QP with all continuous variables
- Default settings:

```
        Nodes                                              Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt        Gap

*     0+    0                           0.3931       -7668.0051                 ---
      0     0     -153.3420     0       0.3931        -153.3420       9         ---
*     0+    0                          -6.3860        -153.3420                 ---
      0     0       -9.5378     0      -6.3860       RLT: 50        1856     49.35%
…
Elapsed time = 46.85 sec. (22852.38 ticks, tree = 125.39 MB, solutions = 632)
   7817  5294       -6.4361     0      -6.3860       -9.0699     1546050   42.03%
…
  10113  6704       -6.8908     0      -6.3860       -9.0699     2081962   42.03%
Elapsed time = 67.55 sec. (32405.73 ticks, tree = 170.23 MB, solutions = 837)
  10352  6877       -6.5928     0      -6.3860       -7.7057     2145588   20.67%
…
  99292  4408        cutoff            -6.3860       -6.4377    22808661    0.81%
 103430   703        cutoff            -6.3860       -6.4002    22946508    0.22%
…
MIP - Integer optimal, tolerance (0.0001/1e-06):  Objective = -6.3860149815e+00
Current MIP best bound = -6.3866525512e+00 (gap = 0.00063757, 0.01%)
```

Slow progress in best node

Try alternate QP rel. algorithms

<u>Solution time = 802.48 sec. Iterations = 22955491 Nodes = 104187 (44)</u>

Example 2

- Nonconvex QP with all continuous variables
- Bqp cut parameter set to most aggressive setting of 3:

```
         Nodes                                        Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                         0.3931      -7668.0051              ---
      0     0     -153.3420     0     0.3931       -153.3420      9       ---
*     0+    0                        -6.3860       -153.3420              ---
      0     0       -9.5378     0    -6.3860      RLT: 50      1856    49.35%
…
 11809  1344       cutoff           -6.3860        -6.5202  4888403    2.10%
Elapsed time = 168.14 sec. (80678.33 ticks, tree = 21.89 MB, solutions = 633)
 12164  1069       cutoff           -6.3860        -6.4920  4948613    1.66%
 12616   778       cutoff           -6.3860        -6.4664  4999328    1.26%
 13413   185       cutoff           -6.3860        -6.4096  5056713    0.37%
…
MIP - Integer optimal, tolerance (0.0001/1e-06):  Objective = -6.3860149815e+00
Current MIP best bound = -6.3866035011e+00 (gap = 0.00058852, 0.01%)
```

<u>Solution time = 174.66 sec. Iterations = 5063193 Nodes = 13690 (11)</u>

Tightening the formulation

- What happens when adjusting parameters is insufficient?
  - Examine model, figure out how to tighten the formulation
  - Start with non-convex MIQPs that linearize to MILPs, since we can draw on knowledge from the easier MILP case
    - Simplify the model if necessary
      - Remove any constraints and integrality restrictions not involved in the performance trouble
      - Try to reproduce the trouble in a smaller data instance
    - Find out how relaxing integrality allows you to cheat
      - How do fractional solutions in the node relaxations allow the objective to improve?
      - In terms of the physical system being modelled, does relaxing integrality introduce valuable new processes at little nor no cost than are unavailable in the MILP version of the model?
    - Use fractional solutions to identify the constraints and variables that will motivate additional cuts

Example: A nonconvex MIQP with a cardinality constraint

Model description:

$$\text{Maximize} \quad \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \neq j}}^{n} \overbrace{d_{ij}}^{>0} * x_i * x_j$$

$$\text{Subject to} \quad \sum_{j=1}^{n} x_j \leq k$$

$$x_j \text{ binary}$$

Source: Klotz, Newman. Practical Guidelines for Solving Difficult Mixed Integer Linear Programs

Example: A nonconvex MIQP with a cardinality constraint

Node Log with n = 28, k = 10:

```
      Nodes                                             Cuts/
Node     Left  Objective  IInf  Best Integer   Best Node    ItCnt  Gap

 0         0    13.0405    28                   13.0405      29
*0+        0                0       5.0000      13.0405      29   160.81%
*0+        0                0       5.6250      13.0405      29   131.83%
100       82     11.7307    25       5.6250      11.7307     129   108.55%
…
2222400 329     5.6256     5       5.6250       5.6256 2371768    0.01%
MIP - Integer optimal, tolerance (0.0001/1e-06):
Objective =  5.6250000000e+00
Solution time =  408.94 sec.
```

Example: A nonconvex MIQP with a cardinality constraint

- Review the guidelines and tactics
  - Simplify the model if necessary
  - **Determine how fractional solutions affect objective**
  - Use fractional solutions to motivate additional cuts
  - Linear or logical combinations of constraints
  - Disjunctions
  - Solve one or more related models
  - Use infeasibility
  - Use solution objective value

Example: A nonconvex MIQP with a cardinality constraint

Determine how fractional solutions affect objective

Simplify the model if necessary

```
max 3x1*x2 + 4x1*x3 + 5x2*x3
subject to
x1 + x2 + x3 <= 2
x1, x2, x3 binary
MIQP optimal solution: x2 = x3 = 1; obj = 5
QP optimal solution: x1 = x2 = x3 = 2/3; obj = 16/3
```

Extreme point (x2 = x3 = 1)

Non-vertex, fractional (x1 = x2 = x3 = 2/3)

We cannot tighten this formulation with linear constraints since the integer solutions are extreme points of the relaxation polyhedron

Example 3: A nonconvex MIQP

Solve one or more related models

```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z12 binary
```

This linearization transforms the nonconvex MIQP into an MIP

Example 3: A nonconvex MIQP

Node log with n=28, k=10 for the linearized problem:

```
     Nodes                            Cuts/
Node   Left    Objective IInf   Best Integer  Best Node ItCnt Gap

 0       0       12.8720   402                 12.8720    617
*0+      0                   0     -0.0000     12.8720    617   ---
*0+      0                   0      0.1250     12.8720    617   ---
…
*85     74                   0      5.0417     11.8962   2345 135.96%
…
137900 23      cutoff               5.6250      5.6333  2707056 0.15%

MIP - Integer optimal solution:  Objective =  5.6250000000e+00
Solution time =  251.66 sec.  Iterations = 2707079  Nodes = 137923
```

Example 3: A nonconvex MIQP

Determine how fractional solutions affect objective

```
max 3z12 + 4z13 + 5z23        subject to
x1 + x2 + x3 <= 2
z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z13 <= x1
z13 <= x3
z13 + 1 >= x1 + x3
z23 <= x2
z23 <= x3
z23 + 1 >= x2 + x3
All variables binary
MIP optimal: z23 = x2 = x3 = 1, obj = 5.
LP optimal: all variables = 2/3, obj = 8
```

zij = 1 ⟺ xi = xj = 1 (true for MIP, LP)

zij < 1 ⟺ xi = 0 or xj = 0 (true for MIP)

Example 3: A nonconvex MIQP

Use Infeasibility

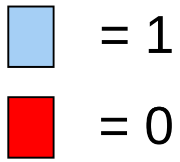
```
max 3z12 + 4z13 + 5z23        subject to
x1 + x2 + x3 <= 2
z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z13 <= x1
z13 <= x3
z13 + 1 >= x1 + x3
z23 <= x2
z23 <= x3
z23 + 1 >= x2 + x3
All variables binary
z23 = z12 = 1  → x1 + x2 + x3 = 3
→ z12 + z13 + z23 <= 1  (cuts off z12 = z13 = z23 = 2/3 in LP)
```

zij = 1 ⟺ xi = xj = 1

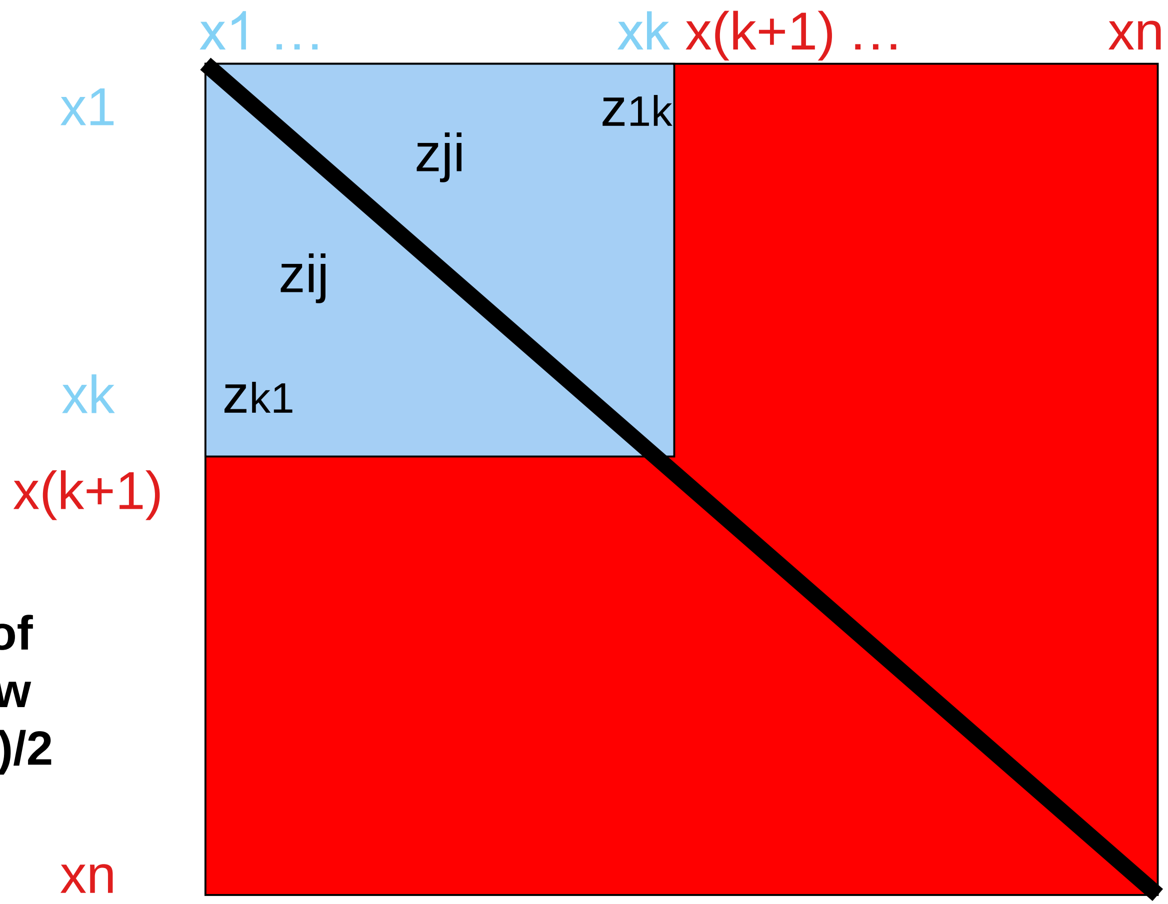
Example 3: A nonconvex MIQP

Extend result to general model with arbitrary n, k:

$z_{ij} = 1 \iff x_i = x_j = 1$



Z =



Count the number of blue elements below the diagonal: k*(k-1)/2

Example: A nonconvex MIQP with a cardinality constraint

Tightened linearized model:

$$\text{Maximize} \quad \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i<j}}^{n} d_{ij} * z_{ij}$$

$$\text{Subject to} \quad \sum_{j=1}^{n} x_j \leq k$$

$$z_{ij} - x_j \leq 0$$

$$z_{ij} - x_i \leq 0$$

$$x_i + x_j \leq 1 + z_{ij}$$

$$\sum_{j=1}^{n} \sum_{\substack{i=1 \\ i<j}}^{n} z_{ij} \leq k * (k-1) / 2$$

Cut

$$x_j, z_{ij} \ \text{binary}$$

Example 3: A nonconvex MIQP

Node log with n=28, k=10 for the linearized problem with cut:

	Nodes				Cuts/		
Node	Left	Objective	IInf	Best Integer	Best Node	ItCnt	Gap
0	0	6.4792	110		6.4792	312	
*0+	0		0	-0.0000	6.4792	312	---
…							
*50+	19		0	5.5000	6.0833	2277	10.61%
…							
300	20	cutoff		5.6250	5.6919	9882	1.19%
…							

```
MIP - Integer optimal solution:  Objective =  5.6250000000e+00
Solution time =     2.89 sec.  Iterations = 10154  Nodes = 330
```

Example: A nonconvex MIQP with a cardinality constraint

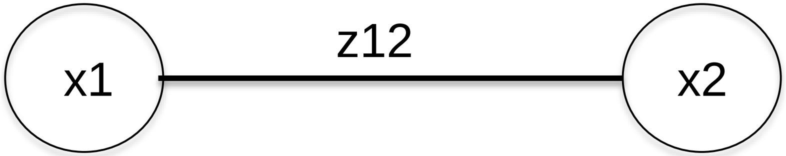
- Treated the linearized MIQP is a generic MILP
  - Drew on well established MILP guidelines to tighten the formulation
    - Linearization constraints were involved
  - Can we do this more efficiently by looking at the generic underlying structure when linearization of products of binary variables are involved?

Useful graph structure present in all linearized nonconvex MIQP with bilinear terms of binaries

- Associate nodes with binary variables in the problem, draw edges for every pair of binaries that have a nonzero quadratic objective coefficient (and hence are linearized)

```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z12 binary
```



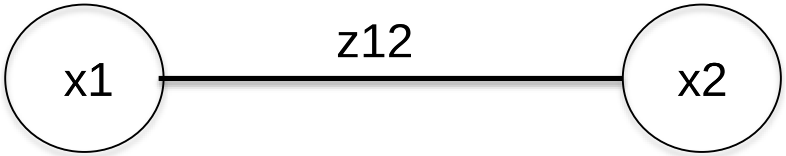
($x_1 = x_2 = .5, z_{12} = 0$ feasible in the linearized LP relaxation)

Useful graph structure present in all linearized nonconvex MIQP with bilinear terms of binaries

- Padberg examined this graph in detail in his 1989 paper **THE BOOLEAN QUADRIC POLYTOPE: SOME CHARACTERISTICS, FACETS AND RELATIVES**
  - Examined the polytope associated with the LP relaxation of the linearized Binary QP
    - For a Binary QP with one bilinear term, its vertices and facets are the same as those of its relaxation

```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z12 binary
```



Useful graph structure present in all linearized nonconvex MIQP with bilinear terms of binaries

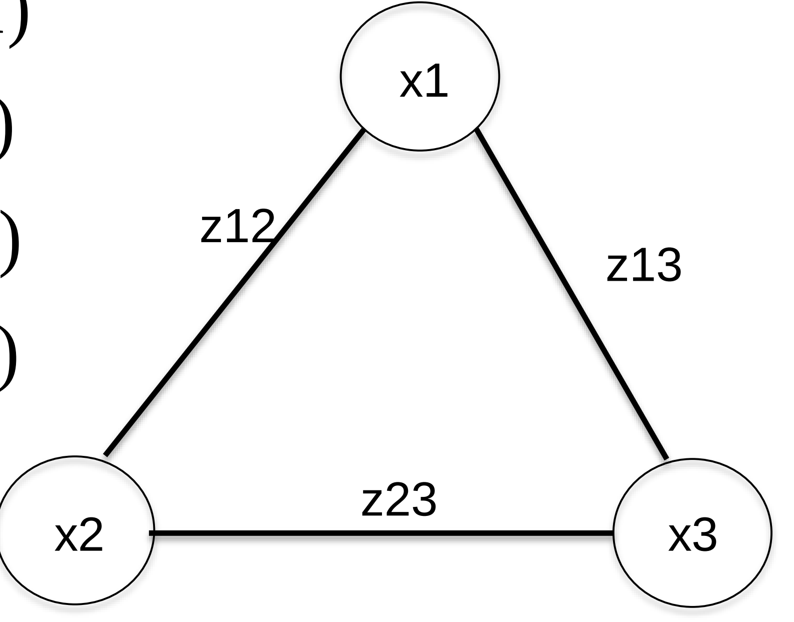
- However, for a Binary QP with 3 binaries and 3 linearized bilinear terms, Padberg derived some facet defining inequalities that cut off fractional solutions in the relaxation

$$x_1 + x_2 + x_3 - (z_{12} + z_{13} + z_{23}) \leq 1 \quad (11)$$

$$z_{12} + z_{13} - (x_1 + z_{23}) \leq 0 \quad (12)$$

$$z_{12} + z_{23} - (x_2 + z_{13}) \leq 0 \quad (13)$$

$$z_{13} + z_{23} - (x_3 + z_{12}) \leq 0 \quad (14)$$



Useful graph structure present in all linearized nonconvex MIQP with bilinear terms of binaries

- Straightforward to prove these cuts are globally valid:

$$x_1 + x_2 + x_3 - (z_{12} + z_{13} + z_{23}) \le 1$$

$$\text{Suppose } \; x_1 + x_2 + x_3 - (z_{12} + z_{13} + z_{23}) \ge 2$$

$$\text{Then } x_1 + x_2 + x_3 \ge 2 \text{ since all variables} \ge 0$$

$$x_1 + x_2 + x_3 = 2 \Rightarrow z_{12} + z_{13} + z_{23} = 1$$

$$x_1 + x_2 + x_3 = 3 \Rightarrow z_{12} + z_{13} + z_{23} = 3$$

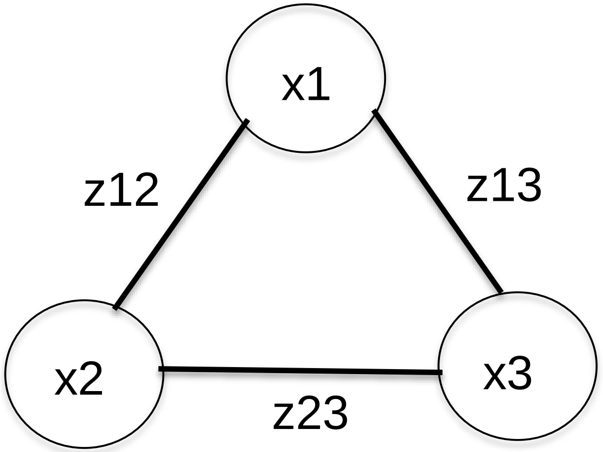
```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z12 binary
```

- Analogous approach to prove the validity of (12)-(14)
- Inductive approach to extend cut (11) above from cliques of size 3 to cliques of arbitrary size
  - But the cut may weaken as the clique size grows

Implications of Padberg's graph for linearized nonconvex MIQPs containing bilinear terms of binaries

- Create the graph with nodes corresponding to binaries, edges corresponding to z variables associated with the linearization of bilinear terms of binaries
- Find the cliques in the graph
- Generate cuts from the cliques
- Can also create similar graphs from constraints involving variables associated with linearizations, use those to generate additional cuts




```
Let x1*x2 = z12

z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z12 binary
```

Example: Minimizing overlap conditions

x11	x12	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	x33	x34	x35	= 1
x41	x42	x43	x44	x45	= 1
x51	x52	x53	x54	x55	= 1
x61	x62	x63	x64	x65	= 1

Overlap conditions: For a given column j in the grid, define an overlap variable for any two rows i1 < i2 such that the overlap variable is 1 if both x(i1,j) and x(i2,j) = 1:

z(i1,i2,j) + 1 >= x(i1,j) + x(i2,j)

6 binaries must be 1 → at least one column has 2 or more binaries set to 1 in any feasible solution → at least one overlap variable z(i1,i2,j) must be 1. Pretty easy for humans familiar with the pigeonhole principle to see

Example: Modeling overlap conditions

Surely CPLEX's numerous cut generators will pick this up…

x11	x12	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	x33	x34	x35	= 1
x41	x42	x43	x44	x45	= 1
x51	x52	x53	x54	x55	= 1
x61	x62	x63	x64	x65	= 1

- Not a rank-1 Chvatal Gomory cut
  - LP feasible solutions with all overlap variable set to 0
- Not a rank-1 Lift and Project Cut
  - Optimize small instance over L&P closure
- Probably not a rank-1 Split Cut
  - Optimize small instance over Split closure

Example: Minimizing overlap conditions

Cheating by relaxing integrality:

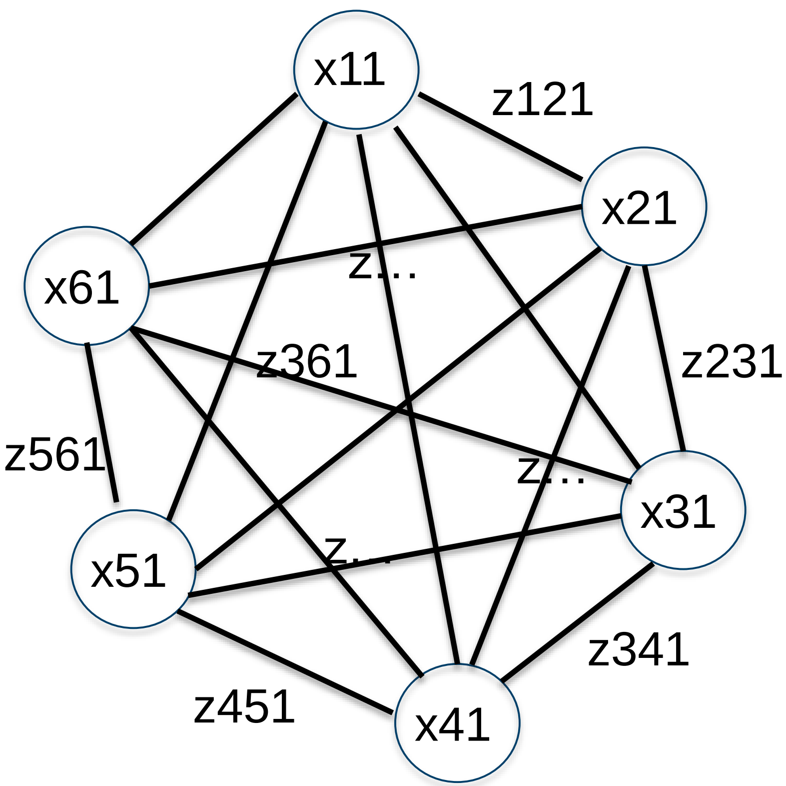
1/5	1/5	1/5	1/5	1/5	= 1
1/5	1/5	1/5	1/5	1/5	= 1
1/5	1/5	1/5	1/5	1/5	= 1
1/5	1/5	1/5	1/5	1/5	= 1
1/5	1/5	1/5	1/5	1/5	= 1
1/5	1/5	1/5	1/5	1/5	= 1

```
z(i1,i2,j) + 1 >= x(i1,j) + x(i2,j)->z(*,*,j) = 0
```

The Padberg Graph consists of 5 complete graphs with 6 vertices each (corresponding to the binaries in one column of the grid), and edges associated with the overlap variables.

Example: Minimizing overlap conditions

Padberg graph, column 1:



x11	x12	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	x33	x34	x35	= 1
x41	x42	x43	x44	x45	= 1
x51	x52	x53	x54	x55	= 1
x61	x62	x63	x64	x65	= 1

Padberg cut, column 1:

$$\underbrace{x_{11} + x_{21} + \ldots + x_{61}}_{\text{All node variables}} - \underbrace{\sum_{i=1}^{6} \sum_{k=i+1}^{6}}_{\text{All edge variables}} z_{ik}^{1} \leq 1$$

All node variables All edge variables

Example: Minimizing overlap conditions

Padberg cut, column 1:

$$x_{11} + x_{21} + \ldots + x_{61} - \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{1} \leq 1$$

Padberg cut, column 2:

$$x_{12} + x_{22} + \ldots + x_{62} - \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{2} \leq 1$$

$$x_{13} + x_{23} + \ldots + x_{63} - \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{3} \leq 1$$

$$x_{14} + x_{24} + \ldots + x_{64} - \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{4} \leq 1$$

Padberg cut, column 5:

$$x_{15} + x_{25} + \ldots + x_{65} - \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{5} \leq 1$$

x11	x12	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	x33	x34	x35	= 1
x41	x42	x43	x44	x45	= 1
x51	x52	x53	x54	x55	= 1
x61	x62	x63	x64	x65	= 1

(aggregate)

$$6 - \sum_{j=1}^{5} \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{j} \leq 5$$

(rearrange)

$$\sum_{j=1}^{5} \sum_{i=1}^{6} \sum_{k=i+1}^{6} z_{ik}^{j} \geq 1$$

(at least one z variable must be 1)

Results for Overlap models

- 5 overlap models, each with 10 grids of varying sizes and aspect ratios
  - CPLEX versions before 12.7 failed to solve any model to optimality within 2 hours

Model	CPLEX 12.8 Defaults, 12 threads	Padberg cuts added to model
10grids	6.62	3.47
10grids_a	3.42	1.03
10grids_b	11.14	7.32
10grids_c	Timeout (2.65%)	Timeout (2.65%)
10grids_big	Timeout (96.93%)	Timeout (38.83%)

(excluding aggregated cut)

- Padberg cuts improved performance, but don't seem to scale up particularly well
  - But were very easy to generate from the Padberg graph

Example: Minimizing overlap conditions

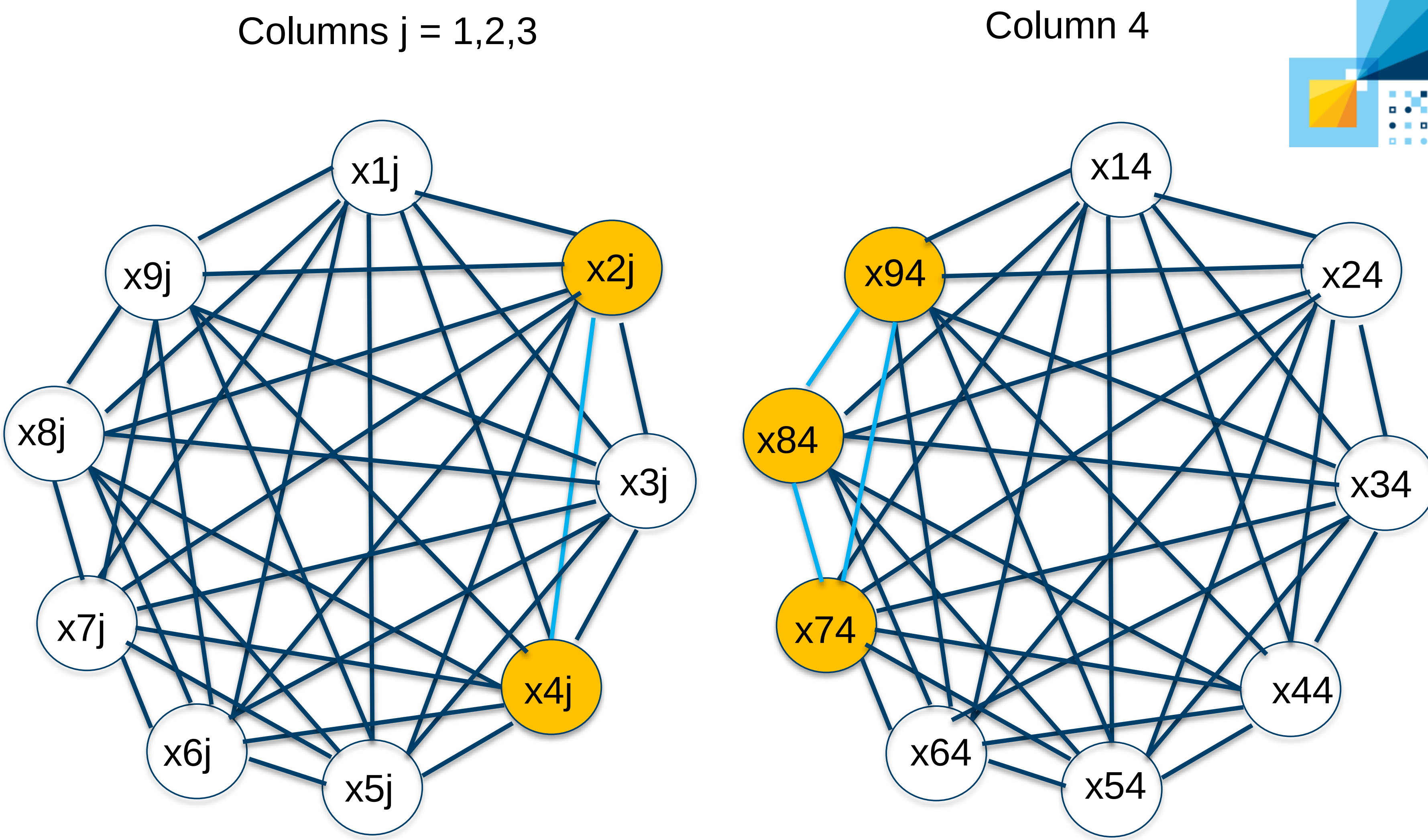
- Padberg graph can derive stronger cuts than aggregating individual Padberg cuts
  - Consider a grid with a smaller aspect ratio:
    - How many z variables must be 1?
      - Fewest occurs when grid elements set to 1 are as evenly distributed as possible
    - Aggregate 4 Padberg cuts:

$$\sum_{j=1}^{4}\sum_{i=1}^{9}\sum_{k=i+1}^{9} z_{ik}^{j} \geq 5$$

    - Use Padberg graph on all 4 columns:

$$\sum_{j=1}^{4}\sum_{i=1}^{9}\sum_{k=i+1}^{9} z_{ik}^{j} \geq 6$$

x11	x12	x13	x14	= 1
x21	x22	x23	x24	= 1
x31	x32	x33	x34	= 1
x41	x42	x43	x44	= 1
x51	x52	x53	x54	= 1
x61	x62	x63	x64	= 1
x71	x72	x73	x74	= 1
x81	x82	x83	x84	= 1
x91	x92	x93	x94	= 1


Columns j = 1,2,3
x1j
x2j
x3j
x4j
x5j
x6j
x7j
x8j
x9j
Column 4
x14
x24
x34
x44
x54
x64
x74
x84
x94

Results for Overlap models

- 5 overlap models, each with 10 grids of varying sizes and aspect ratios
  - CPLEX versions before 12.7 failed to solve any model to optimality within 2 hours

Model	CPLEX 12.8 defaults, 12 threads	Both	Overlap only	Padberg only
10grids	6.62	0.54	0.7	3.47
10grids_a	3.42	0.3	0.29	1.03
10grids_b	11.14	2.67	1.57	7.32
10grids_c	timeout (2.65%)	2.09	1.39	timeout(2.65%)
10grids_big	timeout(96.93%)	timeout (2.72%)	4188.89	timeout(38.83%)

- Overlap cuts stronger than Padberg cuts
  - But Padberg graph made them easier to find
  - Effectiveness significant, but diminishes as grid size grows

Example: Nonconvex MIQP with cardinality constraint

- Back to another model with cardinality constraint:
- Padberg graph is the complete graph with n vertices
  - Could generate Padberg cuts from cliques of size 3,4,5,…,n
  - Experiments indicate Padberg cuts don't help.
  - Instead, use Padberg graph and the cardinality constraint…

$$\textit{Maximize} \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \neq j}}^{n} \overbrace{d_{ij}}^{>0} * x_i * x_j$$

$$\textit{Subject to} \sum_{j=1}^{n} x_j \leq k$$

$$x_j \ \textit{binary}$$

```
Let xi*xj = zij

zij <= xi
zij <= xj
zij + 1 >= xi + xj
zij binary
```

Example: A nonconvex MIQP

Determine how fractional solutions affect objective

```
max 3z12 + 4z13 + 5z23        subject to
x1 + x2 + x3 <= 2
z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z13 <= x1
z13 <= x3
z13 + 1 >= x1 + x3
z23 <= x2
z23 <= x3
z23 + 1 >= x2 + x3
All variables binary
MIP optimal: z23 = x2 = x3 = 1, obj = 5.
LP optimal: all variables = 2/3, z12 + z13 + z23 = 2, obj = 8
```

zij = 1 ⇔ xi = xj = 1 (true for MIP, LP)

zij < 1 ⇔ xi = 0 or xj = 0 (true for MIP, not for LP)

Example 3: A nonconvex MIQP

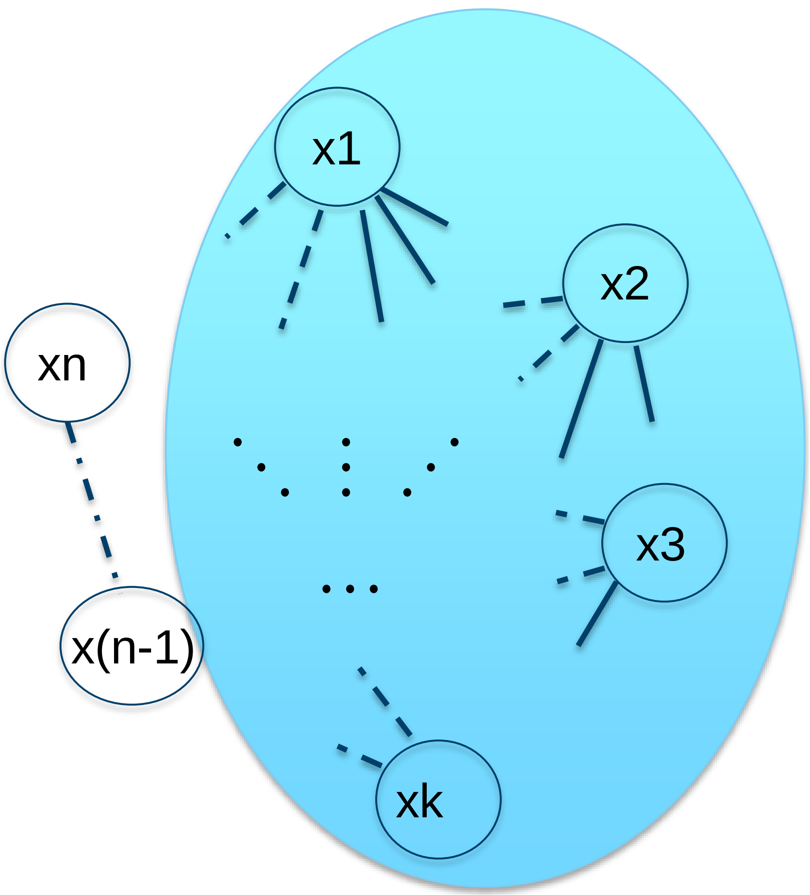
Use Infeasibility

```
max 3z12 + 4z13 + 5z23        subject to
x1 + x2 + x3 <= 2
z12 <= x1
z12 <= x2
z12 + 1 >= x1 + x2
z13 <= x1
z13 <= x3
z13 + 1 >= x1 + x3
z23 <= x2
z23 <= x3
z23 + 1 >= x2 + x3
All variables binary
z23 = z12 = 1  → x1 + x2 + x3 = 3
→ z12 + z13 + z23 <= 1  (cuts off z12 = z13 = z23 = 2/3 in LP)
```

zij = 1 ⟺ xi = xj = 1

Example: A nonconvex MIQP with a cardinality constraint

Padberg graph (complete graph, size n):



- How many zij variables can be set to 1?
  - At most k x variables can be 1
  - Look at any subgraph consisting of k nodes (complete graph, size k)
  - Number of edges of subgraph = # of z variables that can be set to 1
  - k*(k-1)/2 edges, hence sum of all z variables is at most k*(k-1)/2

(dashed edges correspond to those in the original Padberg graph with n nodes but not in the subgraph of k nodes)

Example: A nonconvex MIQP with a cardinality constraint

Tighttened linearized model:

$$\textit{Maximize} \quad \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i<j}}^{n} d_{ij} * z_{ij}$$

$$\textit{Subject to} \quad \sum_{j=1}^{n} x_j \le k$$

$$z_{ij} - x_j \le 0$$

$$z_{ij} - x_i \le 0$$

$$x_i + x_j \le 1 + z_{ij}$$

$$\sum_{j=1}^{n} \sum_{\substack{i=1 \\ i<j}}^{n} z_{ij} \le k * (k-1) / 2$$

Cut

$$x_j, z_{ij} \ \textit{binary}$$

Example: A nonconvex MIQP with a cardinality constraint

- Computational results (maximization problems)
  - All times in seconds, TO = 2 hour timeout

	CPLEX 12.8 defaults, 12 threads			CPLEX 12.8 card. cuts, 12 threads		
Model (n,k)	Time,Nodes	Initial, Final Upper bound	Final Gap	Time,Nodes	Initial, Final Upper bound	Final Gap
28,10	21.7, 102k	750.7,*	0	0.7,1363	383.8,*	0
60,20	TO, 4656788	6245.3,3962.5	61.5%	458.8,17837	3224.4,*	0
100,7	TO, 1332669	3613, 1658	364.4%	49.8, 2054	410.11,*	0
150,30	TO, 214202	23537.6, 20458.78	284.4%	TO, 18264	7920.2, 6952.7	25.1%
200,14	TO, 100369	14553.4, 11012.8	859.2%	TO, 120866	1765.1, 1739.9	29.2%
280,100	TO, 14306	76541.4, 74982.1	158.2%	TO, 6931	41597,1,40027.8	41.4%

  - Much better performance with cardinality cuts on all 6 instances, but advantage diminishes as we scale up
    - Still get superior gaps, bounds as we scale up
      - Single (dense) cut slows node throughput

Extend to arbitrary MIQPs with products of binaries in the objective

- Construct Padberg graph for all products of binaries in the objective
  - May be disjoint collection of connected components
  - Look for cardinality constraints involving only subset of binaries associated with nodes in the Padberg graph
    - Explicit or implicit constraints
  - Generate one cut for each such cardinality constraint
  - As branch and cut continues, at any node that has one or more binary in a cardinality constraint fixed to 0, can generate a tighter locally valid cut
  - Other speedups may exist as well

Other examples

- QUBOs (QUadratic Binary Optimization)
  - Our internal test set of such models have Padberg graphs with no cliques of size 3 or more
    - Nothing to be done
  - QUBOs from max clique reformulation
    - Modest improvement in root bound, but unfavorable overall
    - QUBO formulation significantly worse than direct formulation anyways (for branch and cut, not necessarily for other algorithms)
- What about QAP models?

QAPs

- Basic QAP of dimension n
  - n^2 binary variables, 2n constraints:

x_1	x_2	…	x_{n-1}	x_n	=1
x_{n+1}	x_{n+2}	…	x_{2n-1}	x_{2n}	=1
…	…	…	…	…	=1
$x_{(n\ 2)n+1}$	…	…	…	$x_{(n-1)n+1}$	=1
$x_{(n-1)n+1}$	…	…	…	x_{n^2}	=1
=1	=1	=1	=1	=1	

$$\text{Minimize} \sum_{i=1}^{n^2} \sum_{j=1}^{n^2} \overbrace{q_{ij}}^{>0} * x_i * x_j$$

$$\text{Subject to} \sum_{j=1}^{n} x_{kn+j} = 1 \qquad k = 0,\ldots,n-1$$ (Variable grid row)

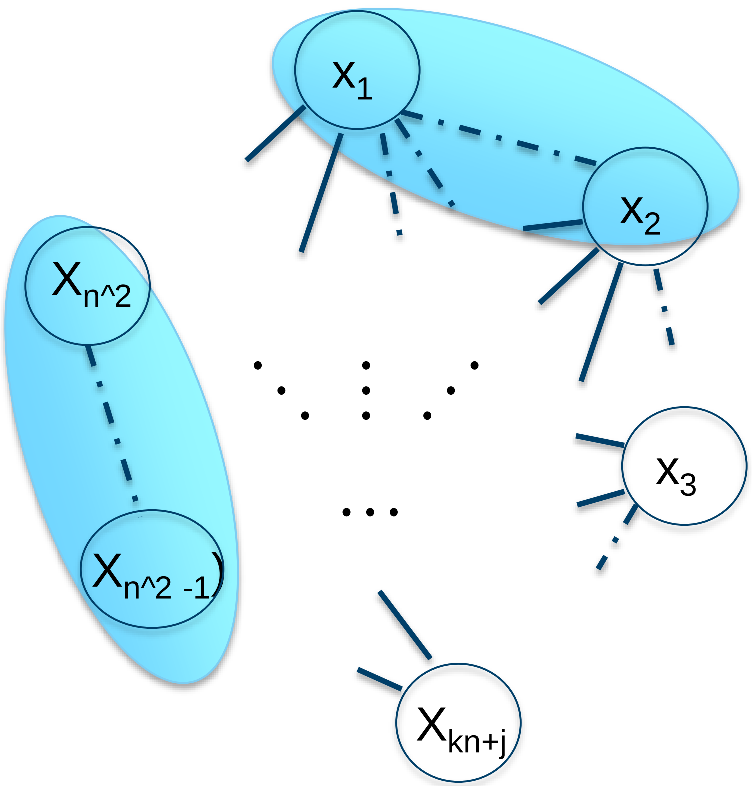
$$\sum_{k=0}^{n-1} x_{kn+j} = 1 \qquad j = 1,\ldots,n$$ (Variable grid column)

$$x_j \text{ binary}, j = 1,\ldots,n$$

$$\sum_{j=1}^{n^2} x_j = n$$ Implied cardinality constraint

Example: QAP of size n with dense off diagonals

Padberg graph (complete graph, size n^2):



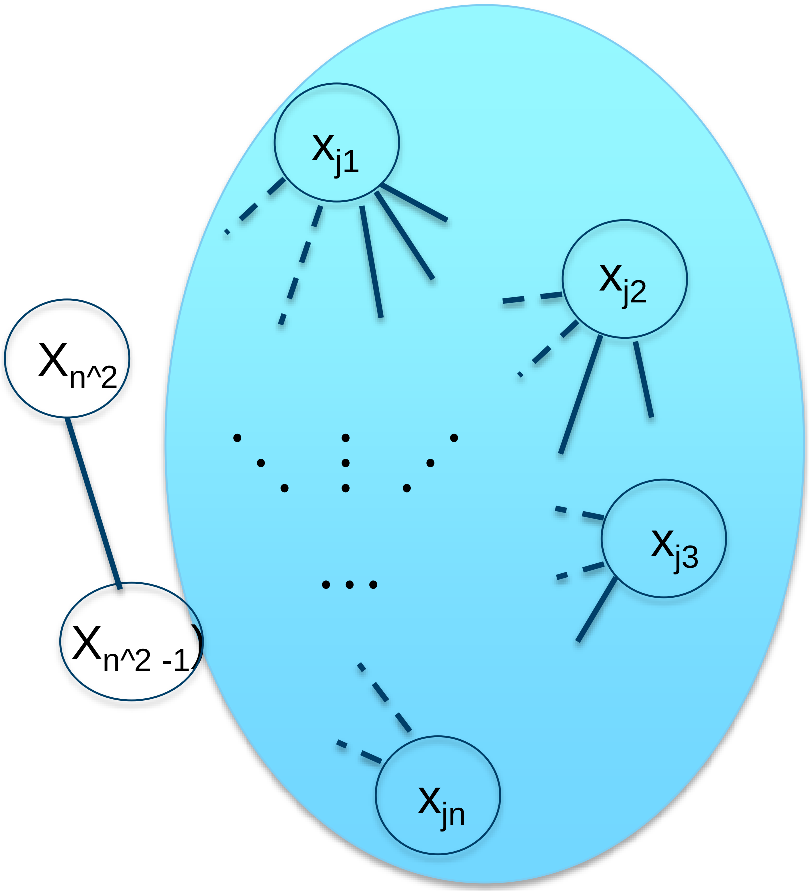
x_1	x_2	…	x_{n-1}	x_n	=1
x_{n+1}	x_{n+2}	…	x_{2n-1}	x_{2n}	=1
…	…	…	…	…	=1
$x_{(n\ 2)n+1}$	…	…	…	$x_{(n-1)n+1}$	=1
$x_{(n-1)n+1}$	…	…	x_{n^2-1}	x_{n^2}	=1
=1	=1	=1	=1	=1	

- First cardinality constraints to consider
  - $x_1 + \ldots + x_n = 1$
  - Cardinality cut, k = 1:

$$z_{ir} = 0 \qquad i = kn + j, r = j+1,\ldots,n$$
$$k = 0,\ldots,n-1,\ j = 1,\ldots,n$$

Example: QAP of size n with dense off diagonals

Padberg graph (complete graph, size n^2):



- Second cardinality constraints to consider
  - $x_1 + \ldots + x_{n^2} = n$
  - WLOG, any n variables that satisfy the grid constraints
  - Cut: sum z_{ij} <= n*(n-1)/2

(dashed edges correspond to those in the original Padberg graph with n^2 nodes but not in the subgraph of n nodes)

QAP models

- QAP models of dimension n
- Start with models with Q matrices with all off diagonal nonzeros positive
- Padberg graph provides cardinality based cuts without any significant computational effort.
- CPLEX's probing already picked off the low hanging fruit
  - Cardinality constraints involving sums of binaries = 1
    - Associated z variables must all be 0
- Cardinality constraint that sum of all assignment binaries must be n yields an additional cut on the sum of the remaining z variables.

QAPs with dense off diagonal Q matrices

- Computational results (minimization problems)

	CPLEX 12.8 defaults, 12 threads			CPLEX 12.8 card. cuts, 12 threads		
Model (n)	Time,Nodes	Initial, Final Lower bound	Final Gap	Time,Nodes	Initial, Final Lower bound	Final Gap
12	164.8, 10654	26735.1, *	0	11.4, 0	209653.4, *	0
15	3746.9,62265	6737.57,*	0	361.7,1072	558121.5,*	0
20	TO, 13389	156910.0, 349972.1	93.4%	TO, 3967	2365485.4, 3391021.5	10.5%
25	TO, 3867	500553.5, 819447.3	94.9%	TO, 795	7069139.9, 7654860.7	46.3%

  - Cut gives much tighter root node bound
  - Node throughput becomes a challenge
  - Barrier at the nodes, turn off zero half cuts.

Examples

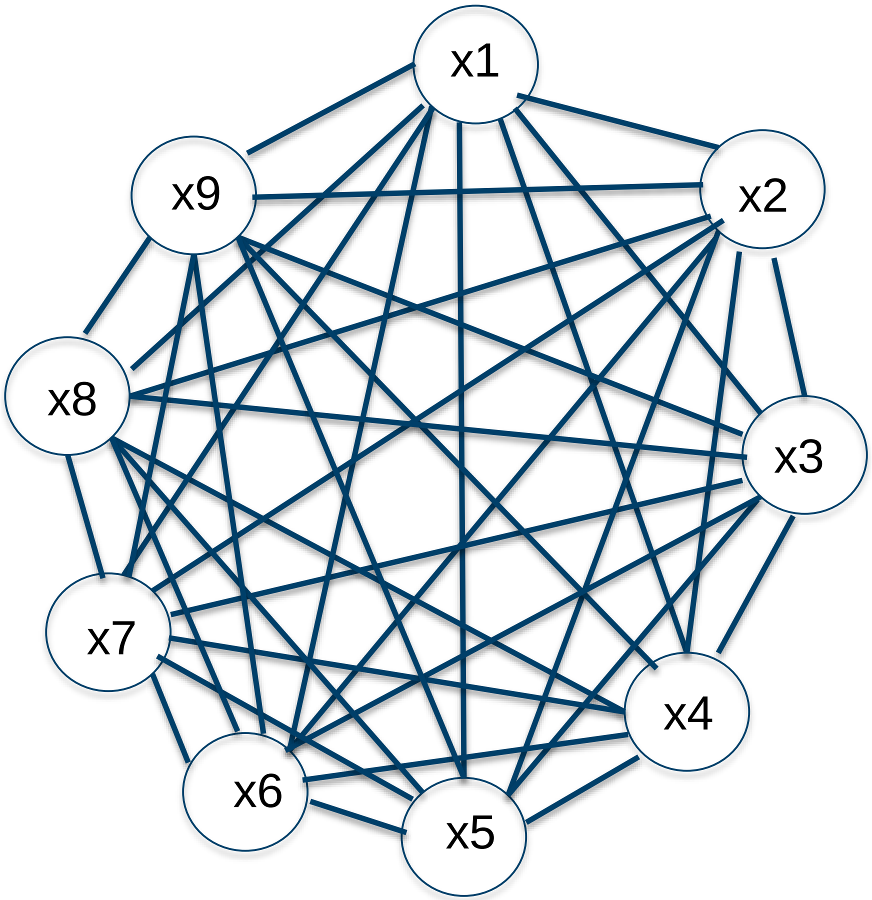
- QAP models from QAPLIB
  - Wide range, from mostly sparse Q matrix to 100% dense
  - Cardinality cut computation time for sparse Q matrix can be quite significant
    - Could have a feasible solution with all zero costs

12*11*11 =
n * (n-1)^2

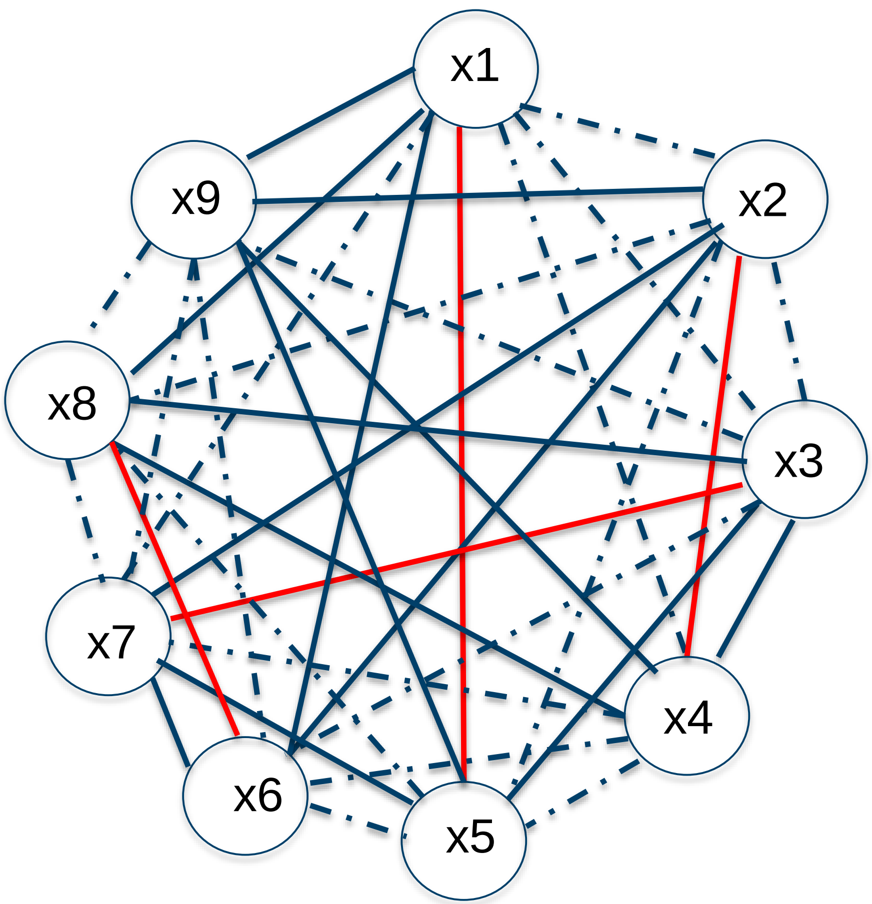
Name	n, n^2	Q nonzeros	Q density
had12	12 144	17424	100%
rou12	12 144	17160	98.48
tai12a	12 144	16896	96.97%
nug12	12 144	11880	68.18%
scr12	12 144	7392	42.42%
chr12a	12 144	2860	15.38%

Examples

x_1	x_2	x_3	=1
x_4	x_5	x_6	=1
x_7	x_8	x_9	=1
=1	=1	=1	



- Cardinality cuts for sparse QAP models from QAPLIB
  - Start small with dense QAP with n = 3, then systematically introduce sparsity.
  - Padberg graph for dense QAP, n = 3



x_1	x_2	x_3	=1
x_4	x_5	x_6	=1
x_7	x_8	x_9	=1
=1	=1	=1	

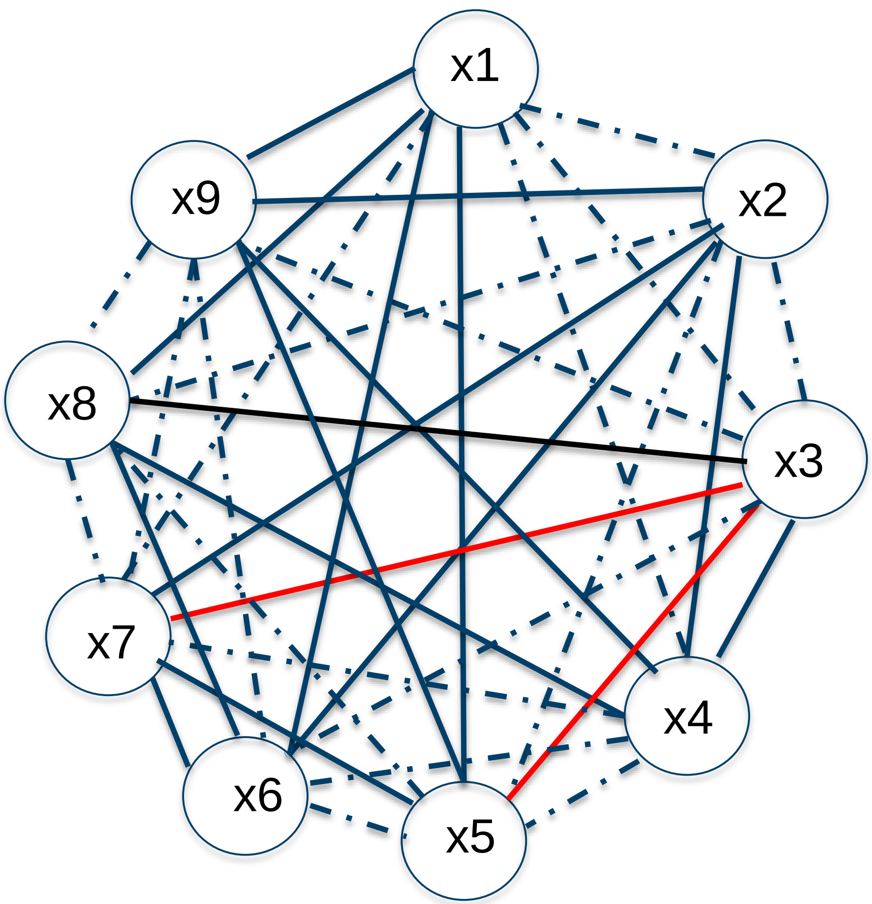
Edge with zij = 0 due to constraint on xi and xj; thus qij = 0

Edge with qij = 0

Edge with qij > 0

At least 2 z variables must be 1
Red edges are a maximum matching

x_1	x_2	x_3	=1
x_4	x_5	x_6	=1
x_7	x_8	x_9	=1
=1	=1	=1	



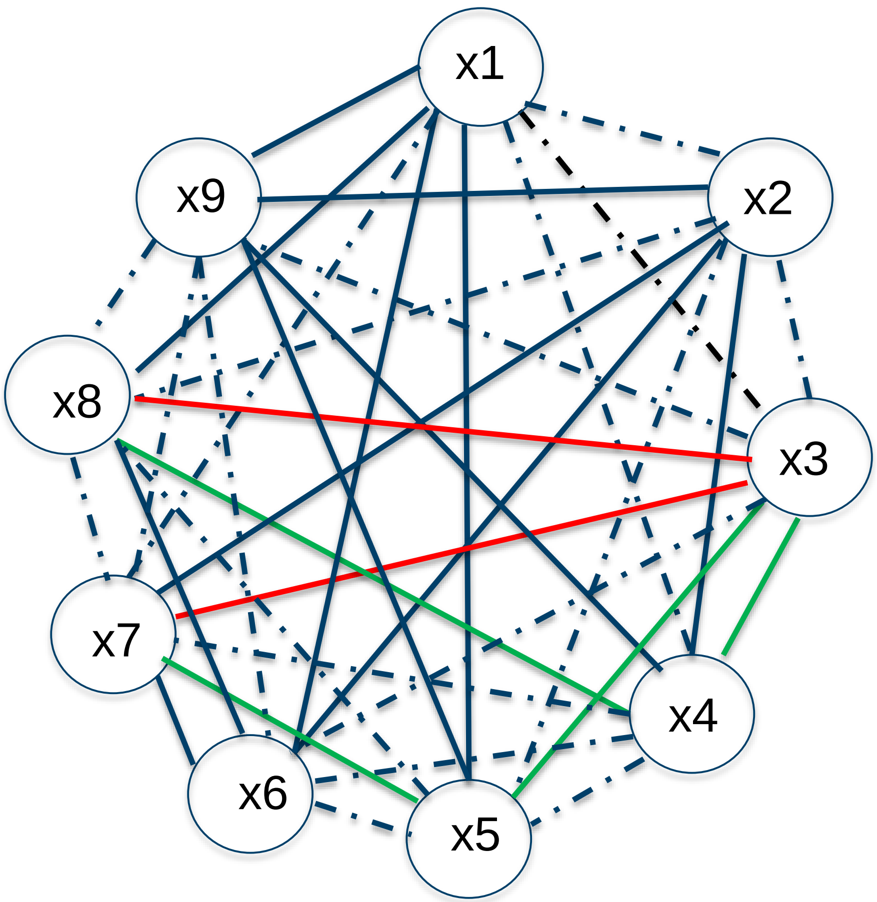
Edge with zij = 0 due to constraint on xi and xj; thus qij = 0

Edge with qij = 0

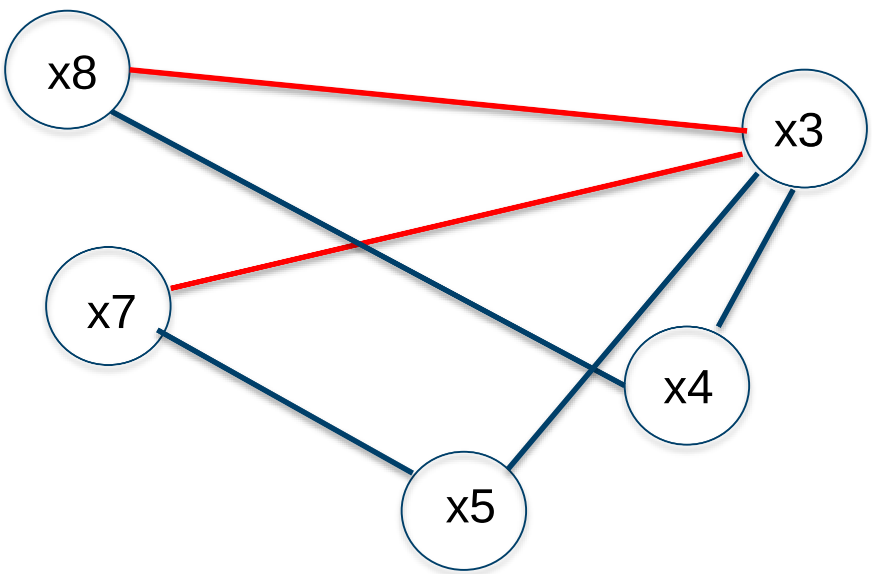
Edge with qij > 0

At least 1 z variable must be 1

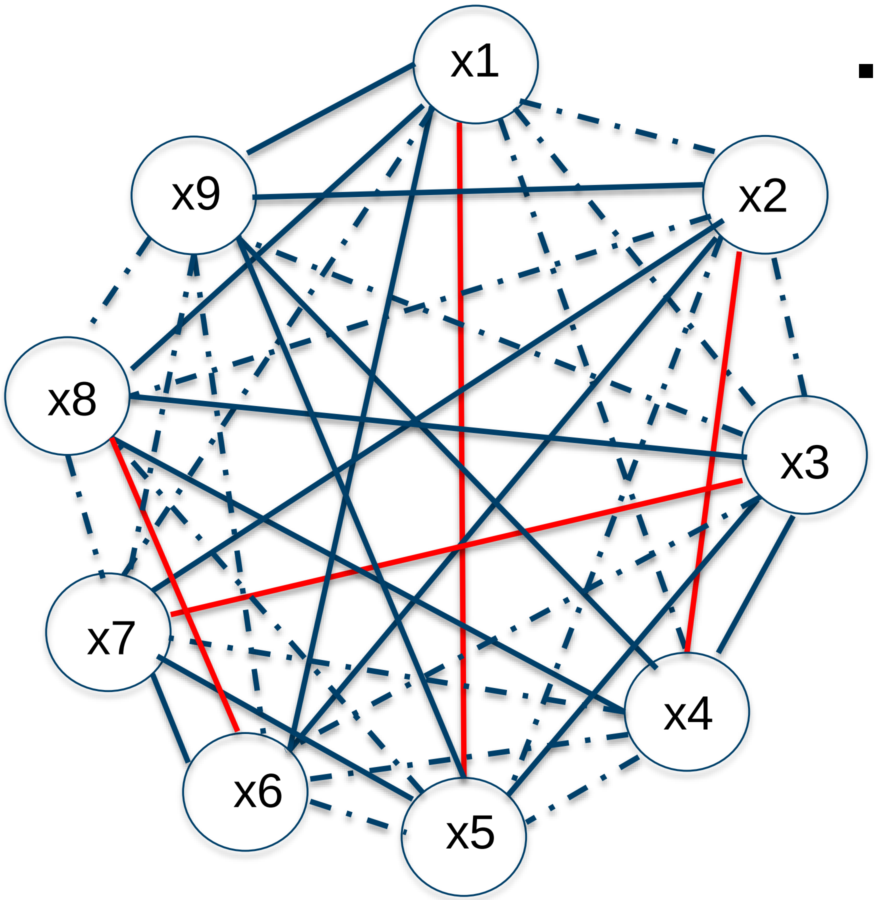
x_1	x_2	x_3	=1
x_4	x_5	x_6	=1
x_7	x_8	x_9	=1
=1	=1	=1	



- QAPs with sparse Q matrices
  - Any node not adjacent to both nodes of at least one zero coeff. edge can be discarded
    - Green edge nodes are adjacent to both nodes
  - From the remaining subgraph, find the clique with the most red edges
    - The # of edges in that clique provides an offset by which we reduce the rhs of the cardinality constraint derived for the dense QAP



- QAPs with sparse Q matrices
  - Rhs of dense cardinality constraint: 3*2/2 = 3
  - Max # of red edges in any clique: 1
  - At least 3 – 1 = 2 z variables must be 1



- QAPs with sparse Q matrices
  - Unfortunately, the subgraph may not be smaller than the original graph
    - Nodes 1 – 8 must be retained since they are part of a red edge
    - Node 9 is adjacent to nodes 2 and 4
  - The QAPLIB models have Padberg graphs that allow few, if any, nodes to be discarded
  - If we have to enumerate all the n-cliques in the graph, that will provide the optimal solution to the original QAP (Junger & Kaibel, 1997)

QAPLIB models of dimension 12

- Computational results
- Dual bounds are not as good as others found in the literature (Junger & Kaibel, 1997)

		CPLEX 12.8 defaults, 12 threads		CPLEX 12.8 card. cuts, 12 threads	
Model	Density	Time,Nodes	Initial Lower bound	Time,Nodes	Initial Lower bound
had12	100%	24444.5, 12385907	0	26325.6, 408249	381.5
rou12	98.48	11295.8, 4185669	0	14030.1, 165636	14822.3
tai12a	96.97%	3876.7, 1527100	0	6703.0, 58048	6774.8
nug12	68.18%	3955.3, 2782290	0	624.7, 10155	134.7
scr12	42.42%	20.2, 20660	0	161.7, 8623	4633.0
chr12	15.38%	1.2, 4699	0	6.26, 1897	842.3

- Adding cardinality cut degrades performance on 5 of 6 models
  - Cut provides better lower bound, much smaller node count
  - One (dense) cut dramatically slows node throughput

Cardinality cut derived from Padberg graph.

- We examined 3 different type of models
  - Grid models with overlap to be minimized
  - Choosing k of n points to maximize diversity
  - For these two model types, we saw big speedups
    - But speedups declined as model size scaled up
      - Still got significant improvement in dual bound, MIP gap for the larger models

Cardinality cut derived from Padberg graph.

- We examined 3 different type of models (ctd)
  - Quadratic assignment problems (QAPs)
    - Easy to generate the cut on dense QAPs
      - Big speedups or gap improvements, but hard to solve to optimality as QAP dimension increases
    - More challenges for QAPs with sparsity
      - Better root dual bounds, much smaller node counts
        - But time per node increased, need to speed up node LP solve times to get major speedups
      - Cut generation time much more significant
        - We may have just moved the heavy lifting elsewhere (finding n-cliques in the Padberg graph)
        - Different sparsity patterns in Q may profoundly affect work to generate the cut.

Testing implementation

- DoCPLEX Python modeling layer in conjunction with Networkx package for graph algorithms
  - DoCPLEX is open source, included at no additional charge with versions 12.8 and later of CPLEX
  - Networkx (https://networkx.github.io/) is also open source, available at no charge
    - Numerous graph algorithms and tools beyond the clique calculations used here
- Cut validation
  - Test validity of cuts by confirming adding the reverse of the cut makes the resulting model is infeasible
    - If ax >= b is the cut, confirm that adding ax <= b – Δ to the original model is infeasible
    - Feasopt may be much faster than mipopt to prove infeasibility

Testing implementation

- DoCPLEX Python modeling layer in conjunction with Networkx package for graph algorithms, clique calculation:

```
Nodelist = range(1,len(Varlist))            # Varlist contains variables in the model
G = nx.Graph()
G.add_nodes_from(Nodelist)
Q = qubomodel.get_objective_expr()
Qiter = Q.iter_quad_triplets()
while True:
    try:
        vi,vj,qij = next(Qiter)
        if (qij != 0.0):
            i1 = Vardict[vi]                # Networkx requires hashable objects
            i2 = Vardict[vj]                # for nodes and edges
            G.add_edge(i1,i2)
    except StopIteration as e:
        break
cliques = nx.find_cliques(G)
cliquenum = nx.graph_clique_number(G)
```

Testing implementation

- DoCPLEX Python modeling layer in conjunction with Networkx package for graph algorithms, linearization of bilinear terms:

```
while True:
  try:
    vi,vj,qij = next(Qiter)
    if (qij != 0.0):
        Zvarname = "zlin_" + vi.name + "_" + vj.name
        Zvar = qubomodel.binary_var(Zvarname)
        Zvarlist.append(Zvar)
        tup = (Vardict[vi], Vardict[vj])
        ZVardict[tup] = Zvar
        ctname = "Linearize_" + vi.name + "_" + vj.name + "_a"
        qubomodel.add_constraint(Zvarlist[Zcount] - vi <= 0, ctname)
        ctname = "Linearize_" + vi.name + "_" + vj.name + "_b"
        qubomodel.add_constraint(Zvarlist[Zcount] - vj <= 0, ctname)
        ctname = "Linearize_" + vi.name + "_" + vj.name + "_c"
        qubomodel.add_constraint(vi + vj - 1 <= Zvarlist[Zcount], ctname)
        Qlinobjexpr += qij*Zvarlist[Zcount]
        Zcount += 1
```

Testing implementation

- DoCPLEX Python modeling layer in conjunction with Networkx package for graph algorithms, linearization of bilinear terms:

```
while True:
    try:
        thisclique = next(cliques)                    # Networkx clique iterator
        if len(thisclique) > 2:
            if cliquesize == 0 or len(thisclique) == cliquesize:
                cutcount += 1
                if cliquelim > 0 and cutcount > cliquelim:
                    break
                else:
                    add_padberg_cut(qubomodel, thisclique, Varlist, ZVardict,
                                    cutcount)

    except StopIteration as e:
        break
```

Summary and Conclusions

- Spatial Branch and Bound generalizes MILP branching
  - Less developed than MILP B&B, but improving rapidly
- 3 different types of nonconvex (MI)QPs
  - Those that can be linearized to MILP (quadratic objective involves only discrete variables
  - Nonconvex QPs (all continuous variables)
  - Nonconvex MIQPs (some integer restrictions, but some continuous variables in the quadratic objective)
  - Different strategies apply depending on problem type
    - Ratio of continuous to integer variables in objective more important that overall ratio in the model
- Integrality based cuts, other integrality based reductions and heuristics, may not be effective
- Tightest bounds possible can favorably impact performance

Summary and Conclusions

- Padberg's BQP graph has additional uses beyond those in his original paper
  - Extend his cuts beyond the 3 cliques
    - Initial computational results modest
  - Use in conjunction with constraints to generate additional cuts
    - Cardinality cuts yielded significant performance boosts in grid overlap, maximum diversity models and dense QAPs
      - Less dramatic but still significant as problem size increased
    - Sparse QAPs more challenging
      - But even then, the Padberg graph provided additional insight on the nature of these models

Sources/References

- Spatial B&B deep dive: http://ibm.co/1gtDdlj
- General optimization resources, including MINLP:

https://optimization.mccormick.northwestern.edu/index.php/Main_Page

- RLT:**A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems** Sherali and Adams, Springer 1999
- More details in the methods in the CPLEX global (MI)QP solver: http://cerc-datascience.polymtl.ca/wp-content/uploads/2016/06/Technical-Report_DS4DM-2016-001-1.pdf
- MILP performance tuning (including linearized MIQP):

  http://www.sciencedirect.com/science/journal/18767354/18

  or

  http://inside.mines.edu/~anewman/MIP_practice120212.pdf

Sources/References (ctd)

- Padberg, **The boolean quadric polytope: Some characteristics, facets and relatives,** Mathematical Programming August 1989, Volume 45, Issue 1–3, pp 139–172
- Junger, Kaibel, **Box-inequalities for quadratic assignment polytopes,** Mathematical Programming October 2001, Volume 91, Issue 1, pp 175–197

Backup

Example: Minimizing overlap conditions

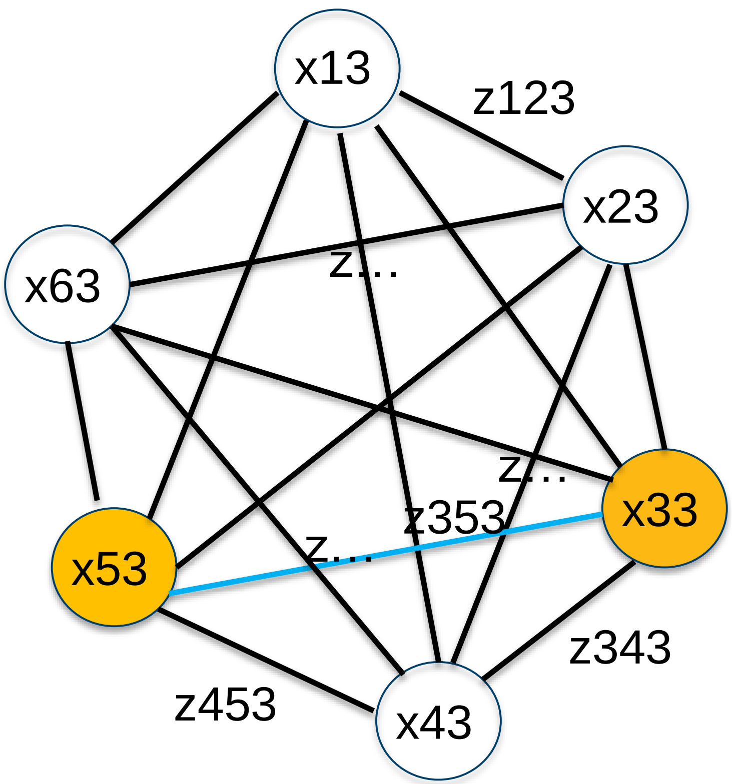
- Padberg graph can also be used to derive stronger cuts.
  - Cardinality constraint: $\sum_{j=1}^{5}\sum_{i=1}^{6} x_{ij} = 6$
  - Padberg graph consists of a disjoint complete graph for each grid column with 6 distinct x_{ij} variables as nodes
  - Any integer feasible solution must have at least one complete graph with at least 2 node variables set to 1 → the overlap variable in the corresponding edge must be 1 → the sum of all the overlap variables must be at least 1

x11	**x12**	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	**x33**	x34	x35	= 1
x41	x42	x43	x44	**x45**	= 1
x51	x52	**x53**	x54	x55	= 1
x61	x62	x63	**x64**	x65	= 1

Example: Minimizing overlap conditions

- Padberg graph can also be used to derive stronger cuts.

Padberg graph, column 3:



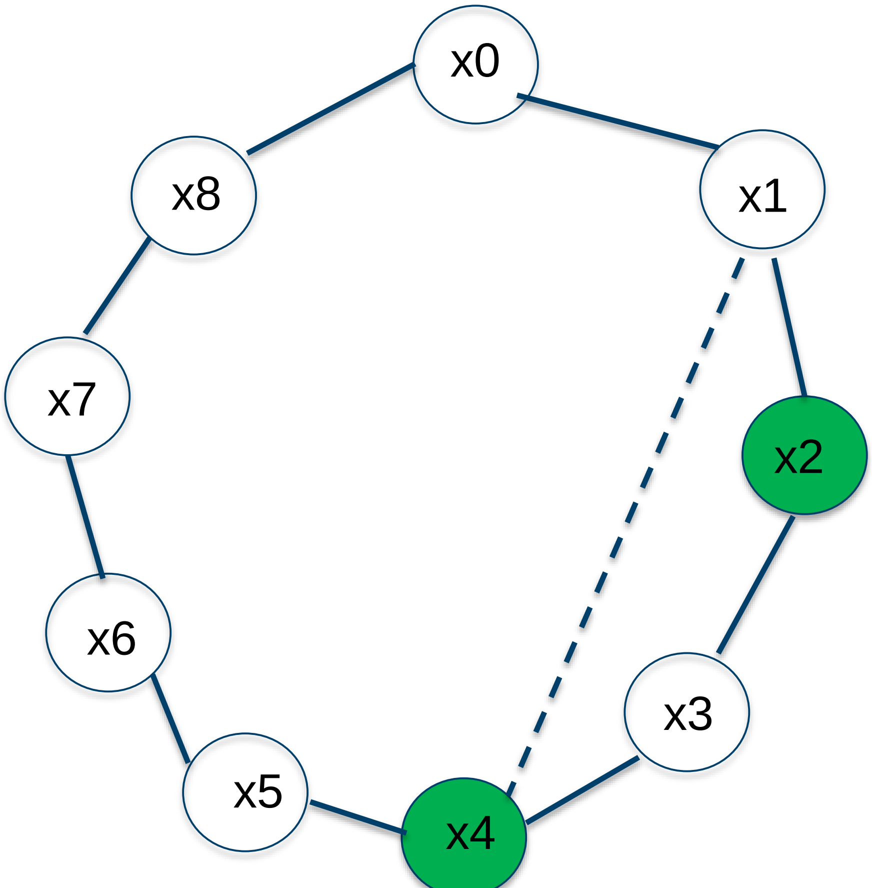
$$\sum_{j=1}^{5}\sum_{i=1}^{6} x_{ij} = 6$$

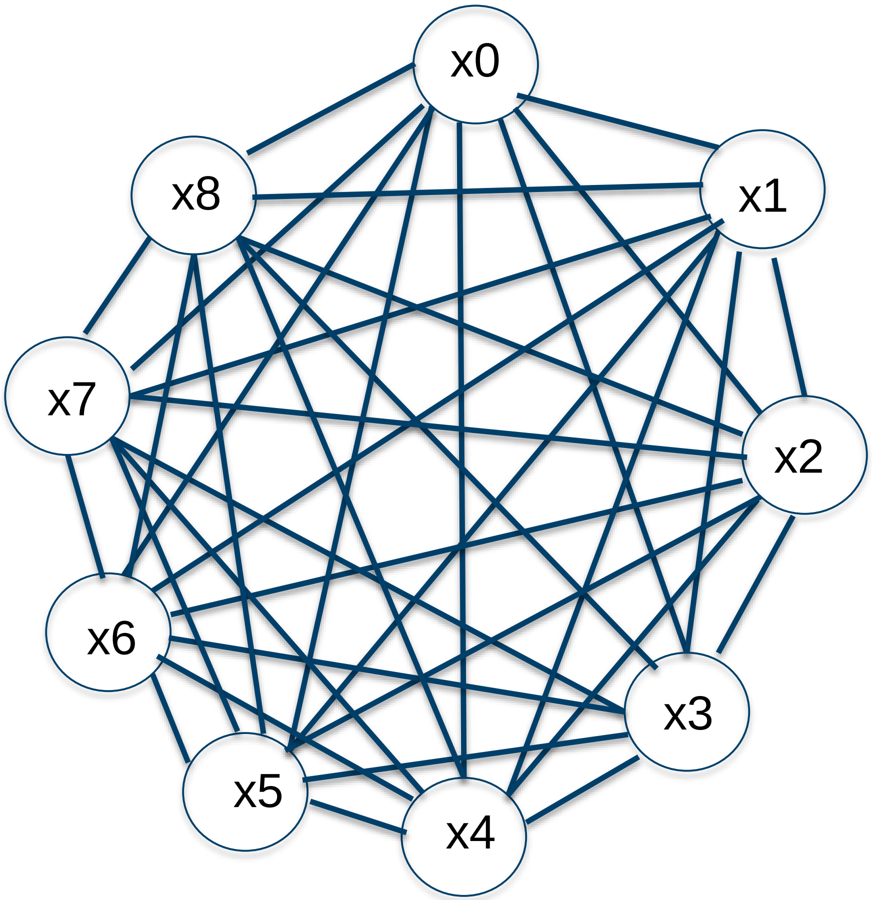
x11	x12	x13	x14	x15	= 1
x21	x22	x23	x24	x25	= 1
x31	x32	x33	x34	x35	= 1
x41	x42	x43	x44	x45	= 1
x51	x52	x53	x54	x55	= 1
x61	x62	x63	x64	x65	= 1

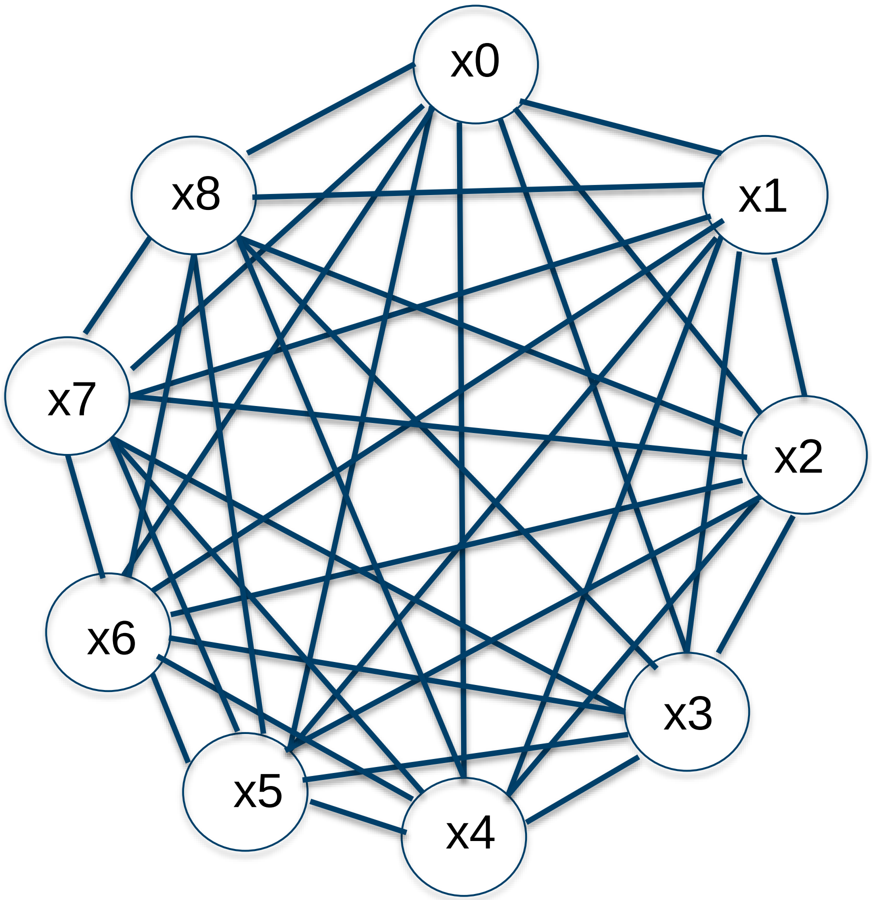
$$\sum_{j=1}^{5}\sum_{i=1}^{6}\sum_{k=i+1}^{6} z_{ik}^{j} \geq 1$$

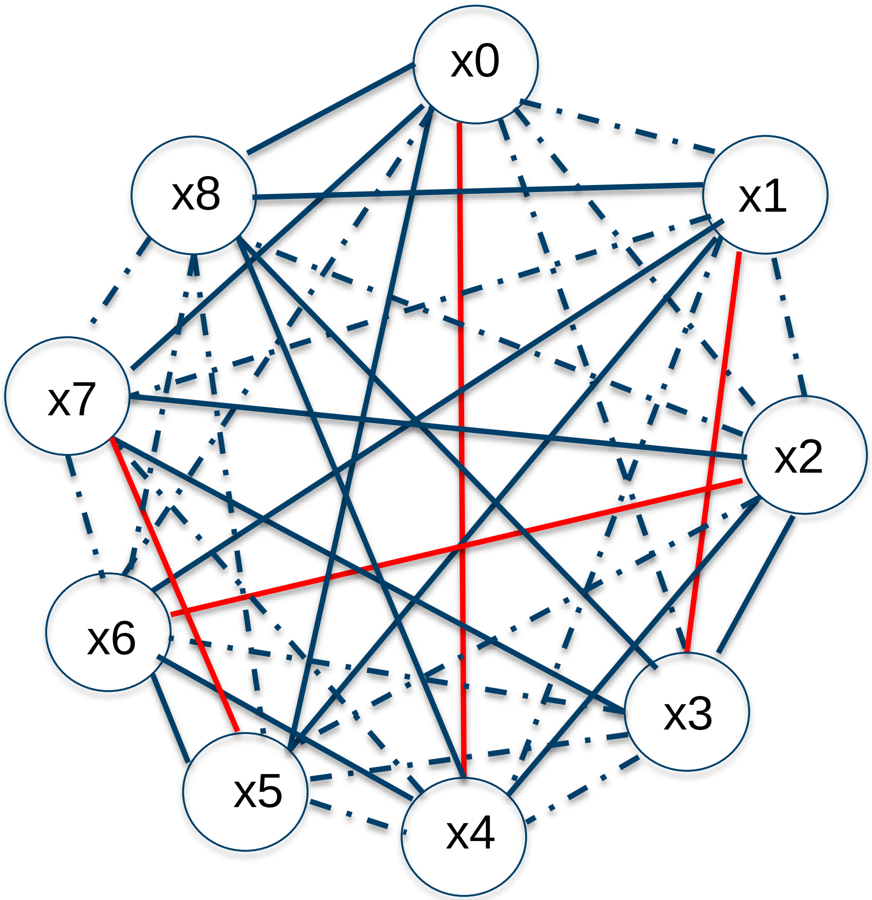
Same cut, no aggregation needed

$$x_1 + x_2 + x_3 + x_4 - (z_{12} + z_{23} + z_{34} + z_{41}) = 2$$

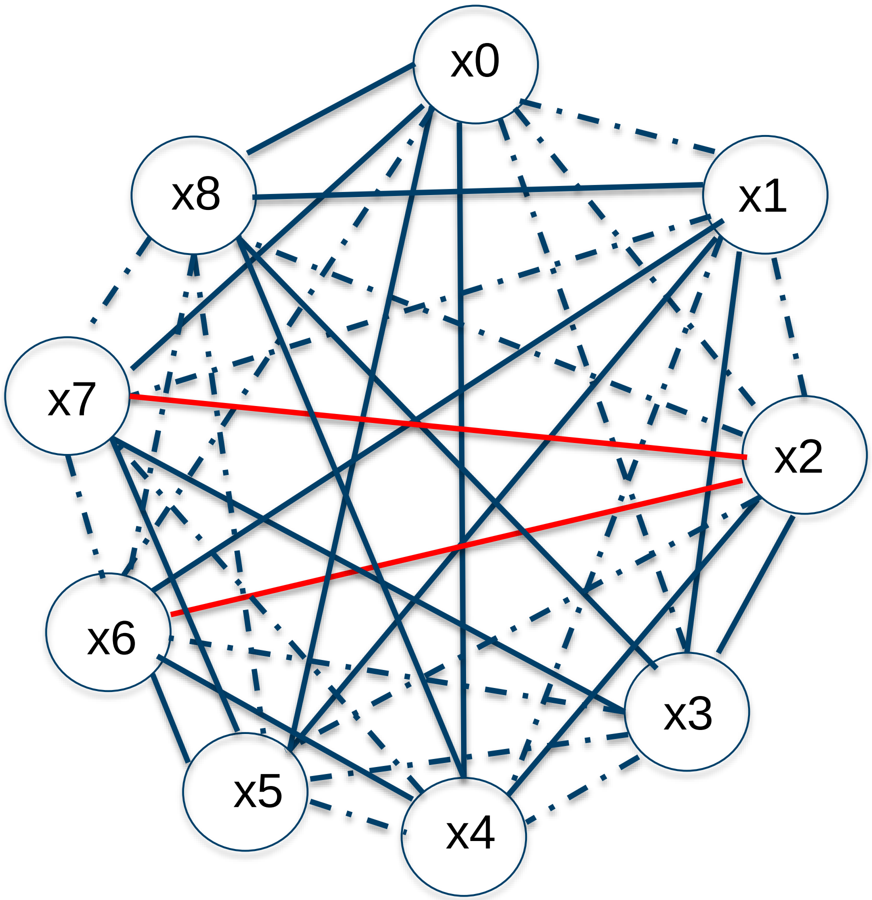



x0
x8
x1
x7
x2
x6
x3
x5
x4

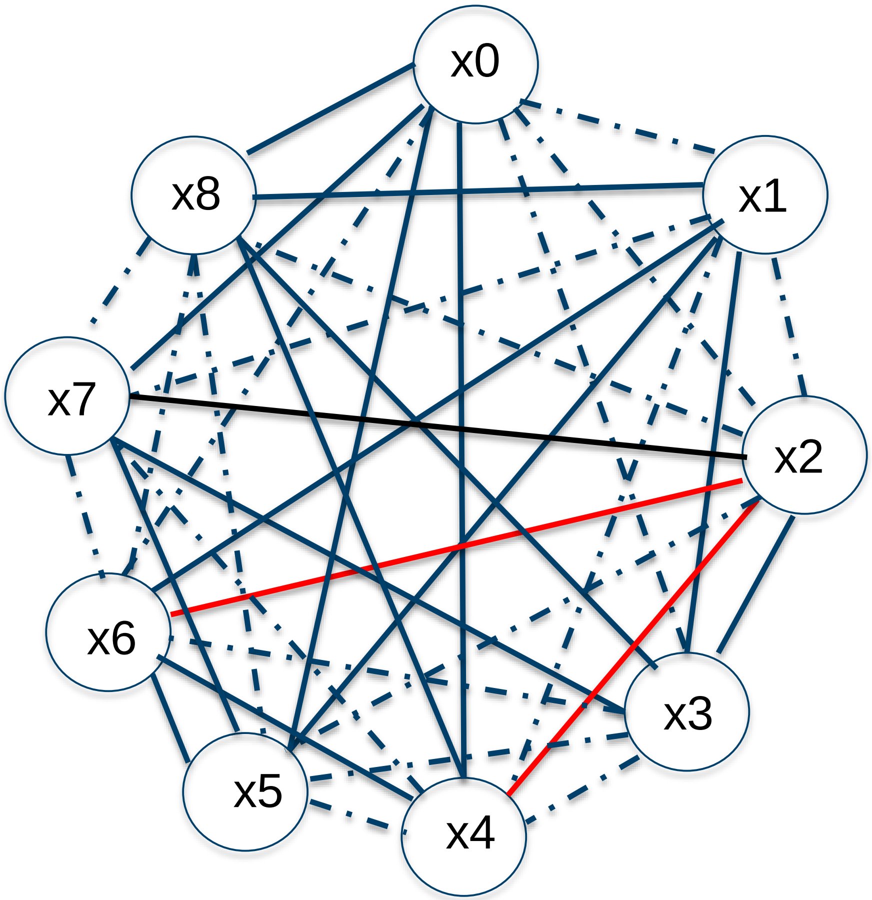

x0
x1
x2
x3
x4
x5
x6
x7
x8



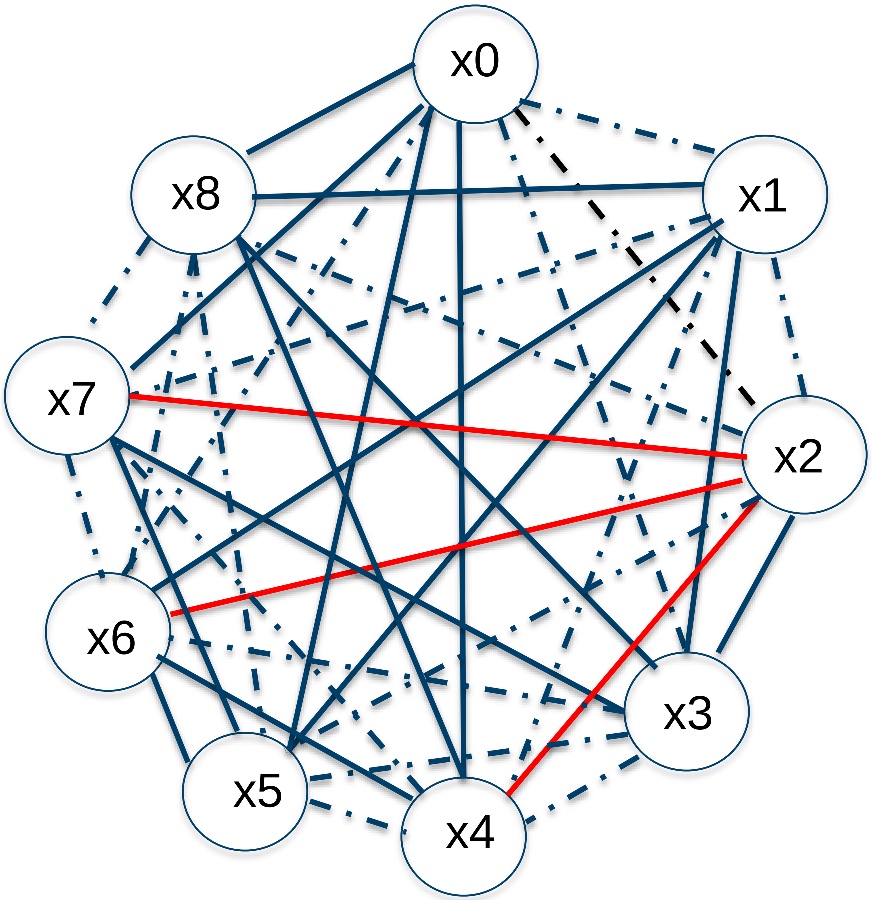
At least 2 z variables must be 1
Red edges are a maximum matching



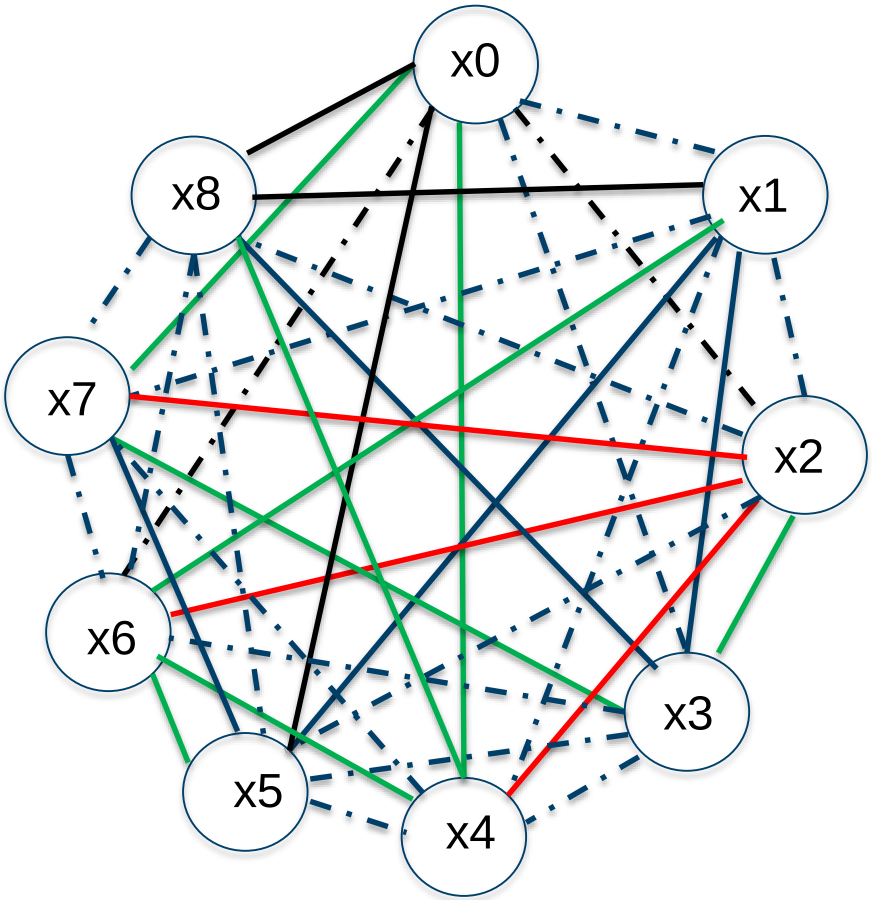
At least 2 z variables must be 1



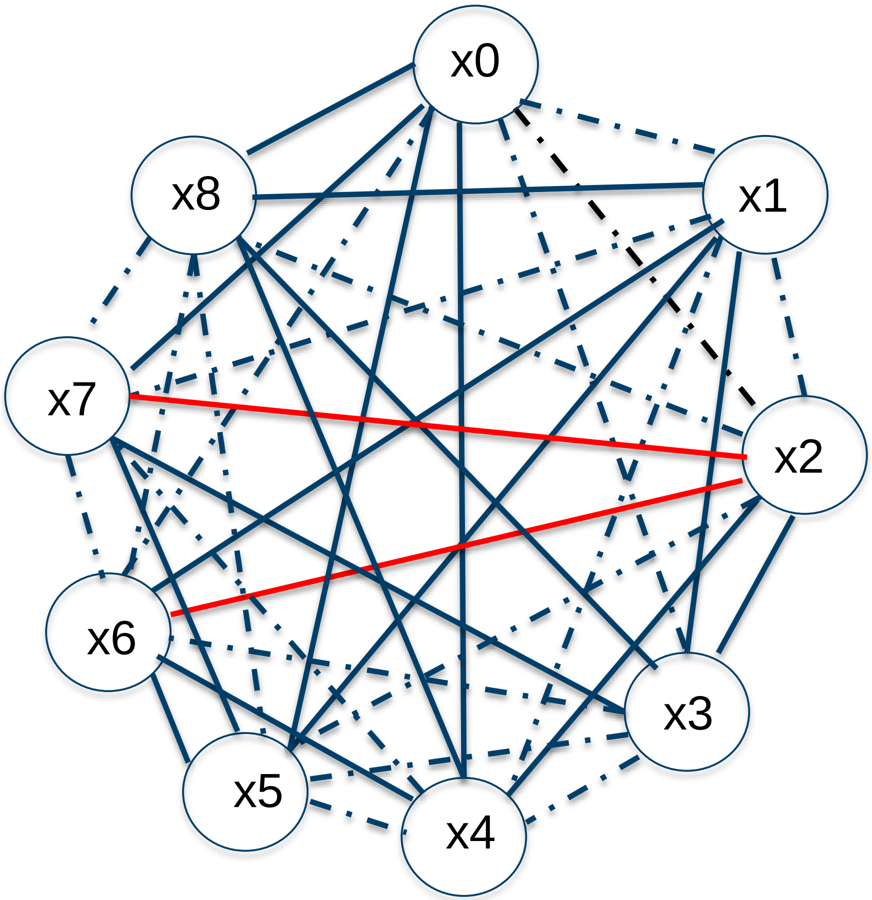
At least 1 z variable must be 1



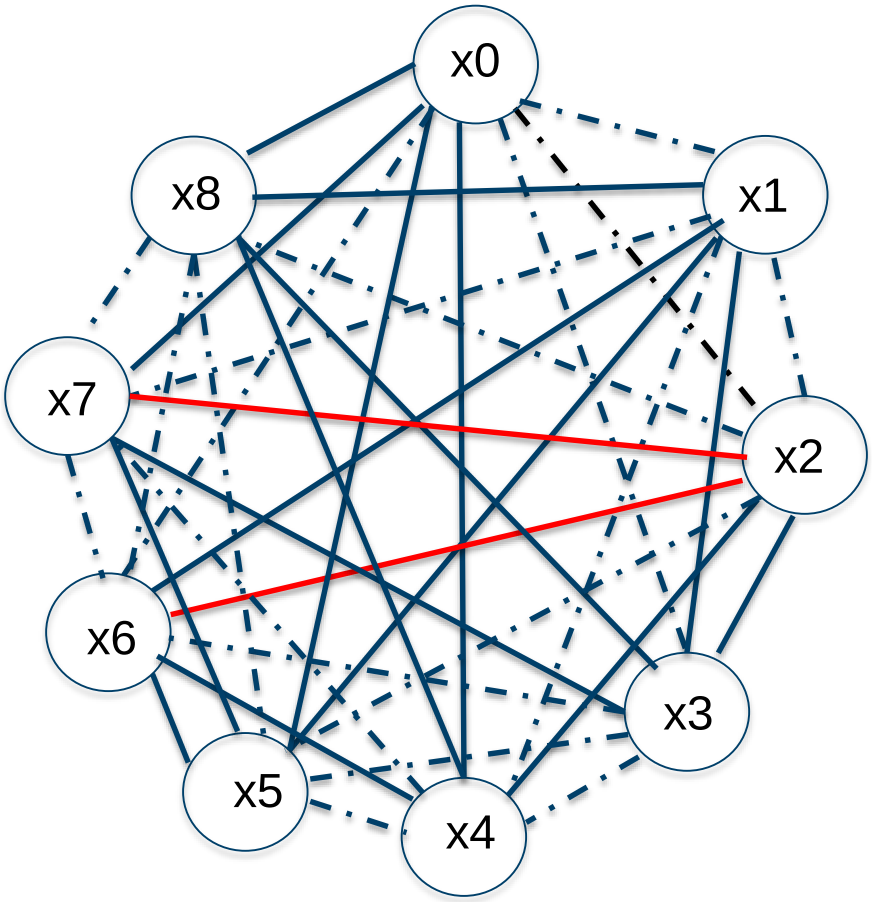
At least 1 z variable must be 1



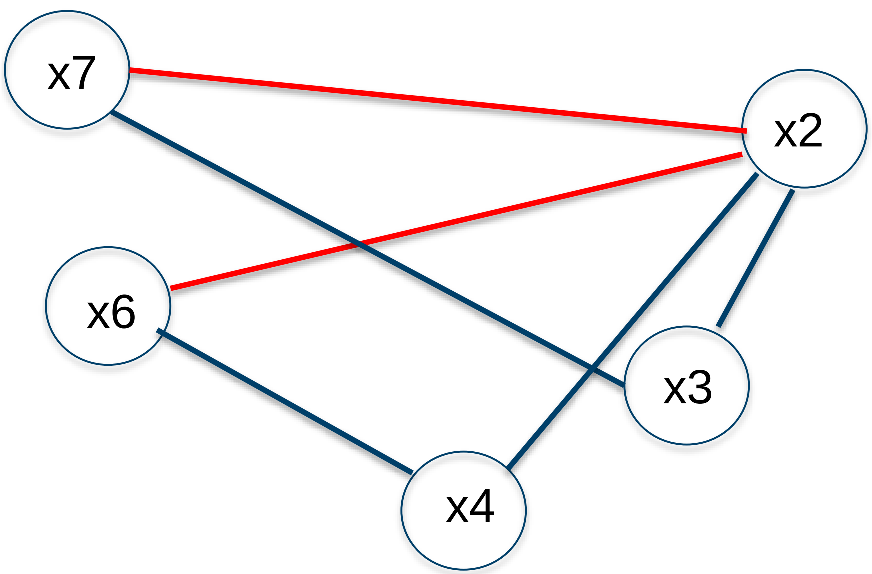
At least 1 z variable must be 1



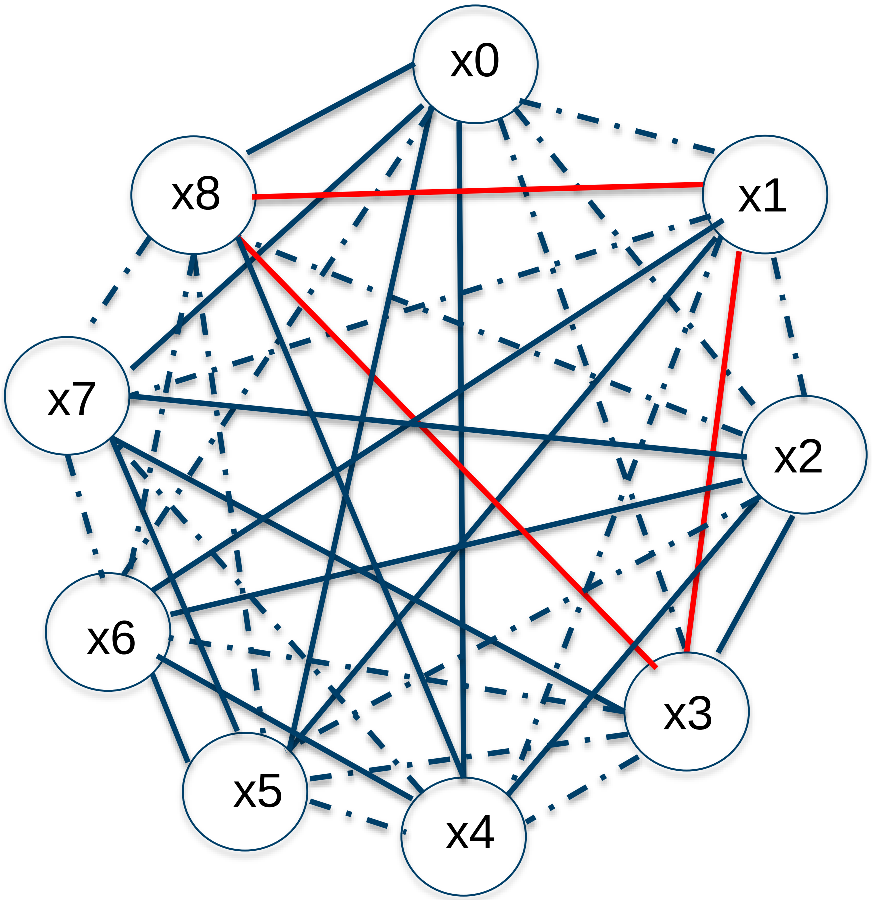
At least 2 z variables must be 1



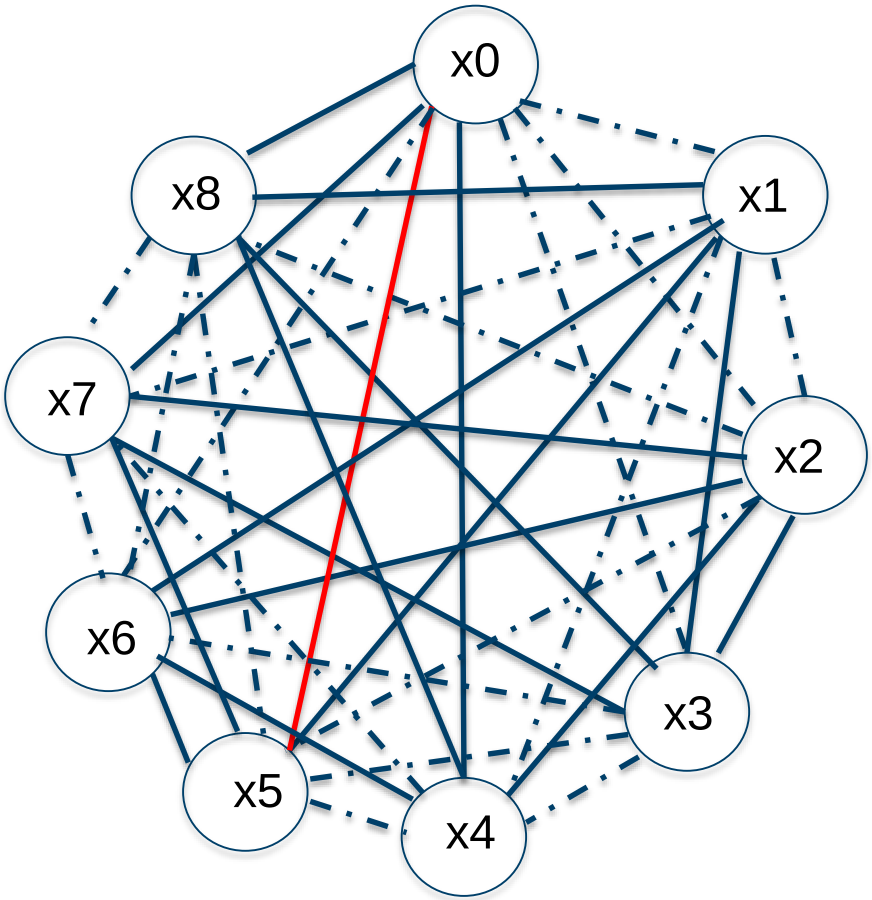
At least 2 z variables must be 1

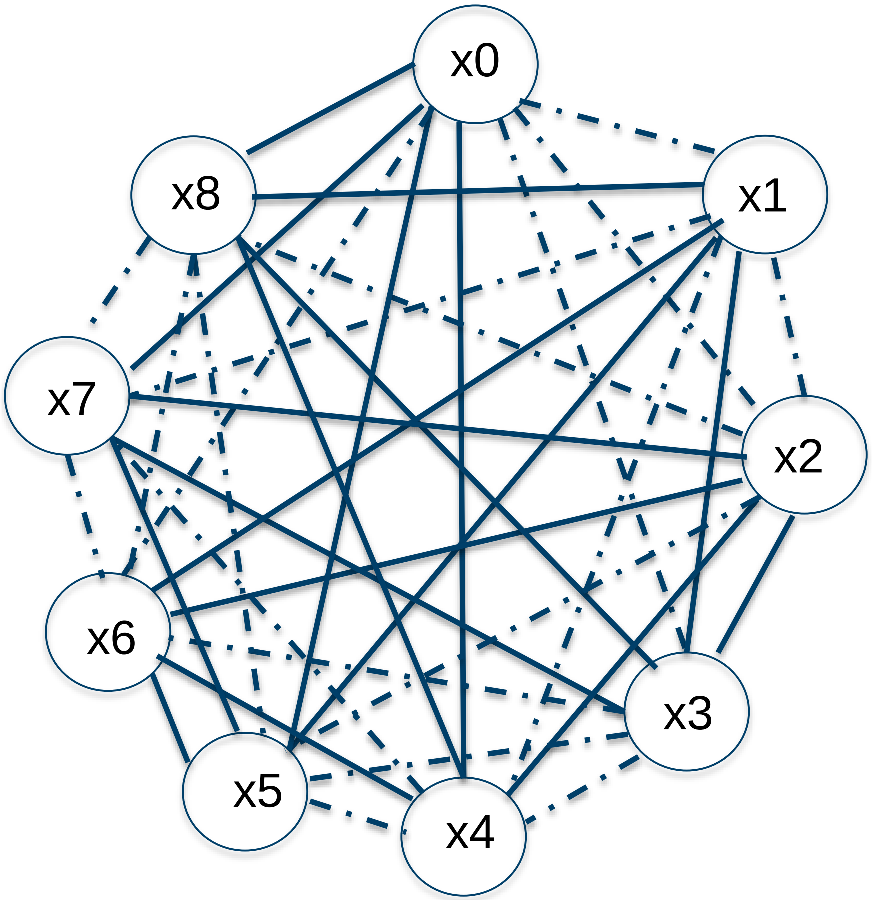


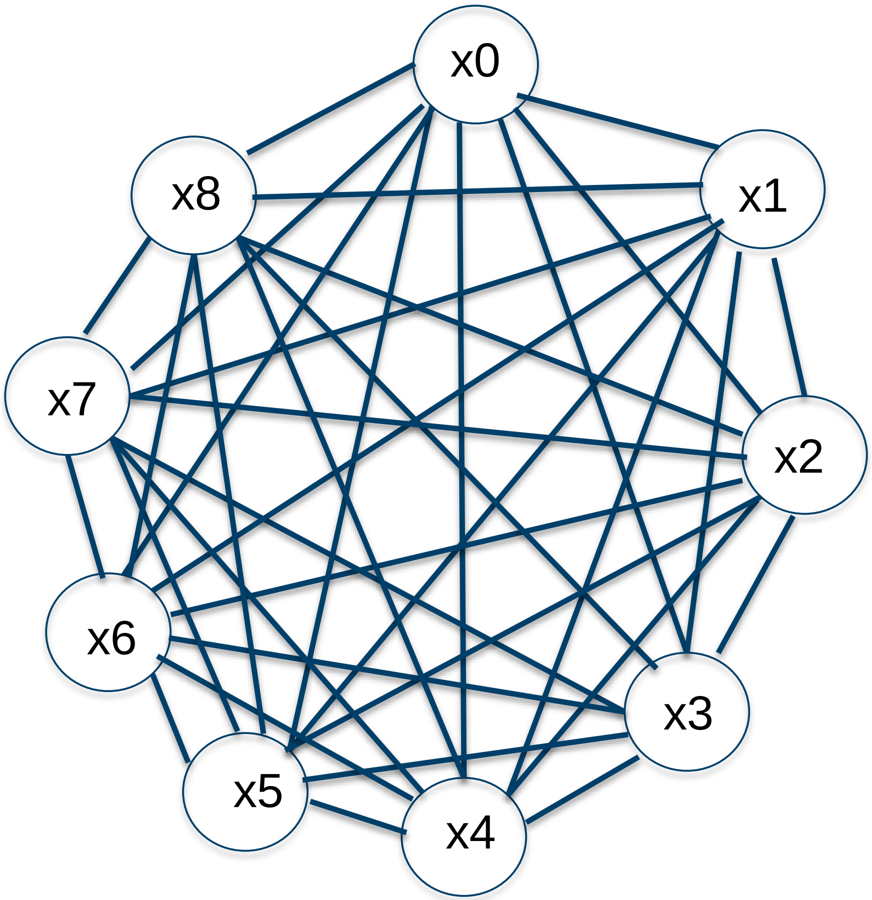
At least 2 z variables must be 1

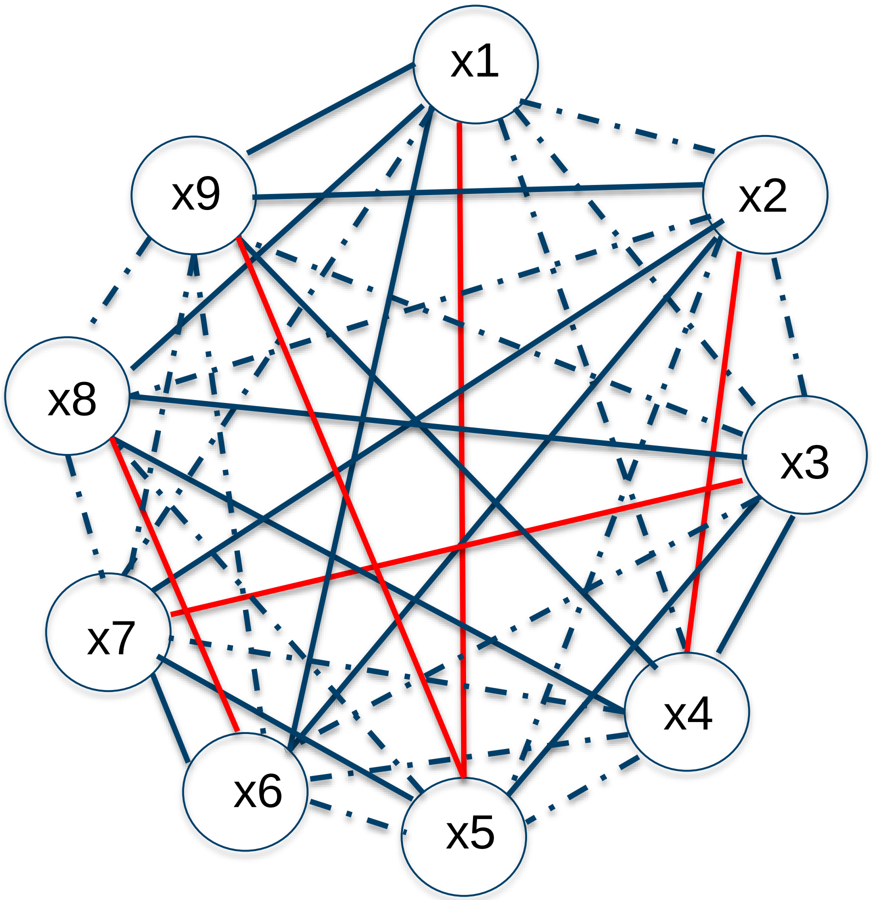


Feasible solution with all 3 associated z variables = 0; no cut available.


x0
x1
x2
x3
x4
x5
x6
x7
x8


x0
x8
x1
x7
x2
x6
x3
x5
x4


x0
x1
x2
x3
x4
x5
x6
x7
x8



- QAPs with sparse Q matrices
  - Unfortunately, the subgraph may not be smaller than the original graph
    - Nodes 1 – 8 must be retained since they are part of a red edge
    - Node 9 is adjacent to nodes 2 and 4
  - The QAPLIB models have Padberg graphs that allow few, if any, nodes to be discarded
  - If we have to enumerate all the n-cliques in the graph, that will provide the optimal solution to the original QAP (Junger & Kaibel, 1997)

Example 1

- Nonconvex QP with all binaries and cardinality constraints
  - Draw a grid with the first bilinear objective term on the vertical axis, second bilinear term on the horizontal axis
  - Use cardinality constraints to tighten the formulation
  - Another example of this can be found at

http://www.sciencedirect.com/science/journal/18767354/18

or

http://inside.mines.edu/~anewman/MIP_practice120212.pdf

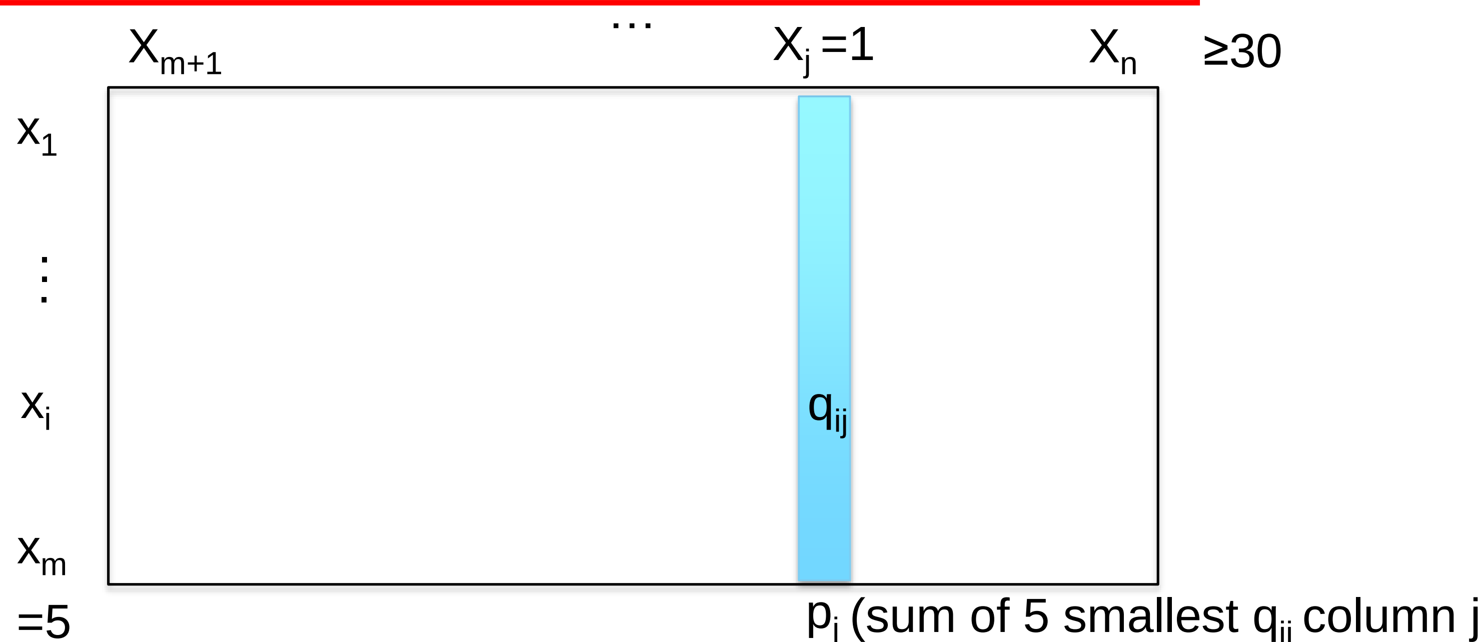
Example 1

- Nonconvex QP with all binaries and cardinality constraints:

$$Suppose\ x_j = 1,\ q_{i_1 j} \le q_{i_2 j} \le q_{i_3 j} \le q_{i_4 j} \le q_{i_5 j} \le q_{ij}$$

$$Let\ \ p_j = q_{i_1 j} + q_{i_2 j} + q_{i_3 j} + q_{i_4 j} + q_{i_5 j}$$

$$If\ \ p_{j_1} \le \ldots \le p_{j_{30}} \le \ldots p_{j_{n-m}}\ \ then \sum_{k=1}^{30} p_{j_k} \le \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j$$

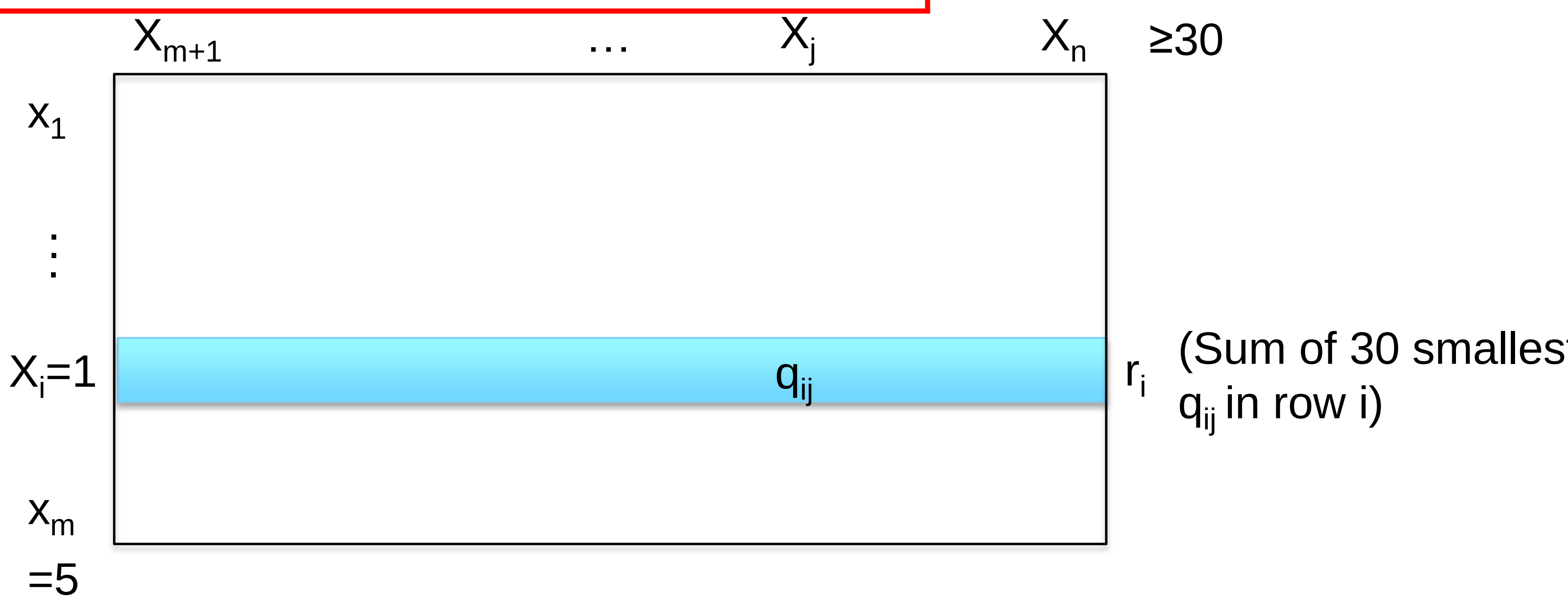


Example 1

- Nonconvex QP with all binaries and cardinality constraints
  - Analogous argument for rows in the grid

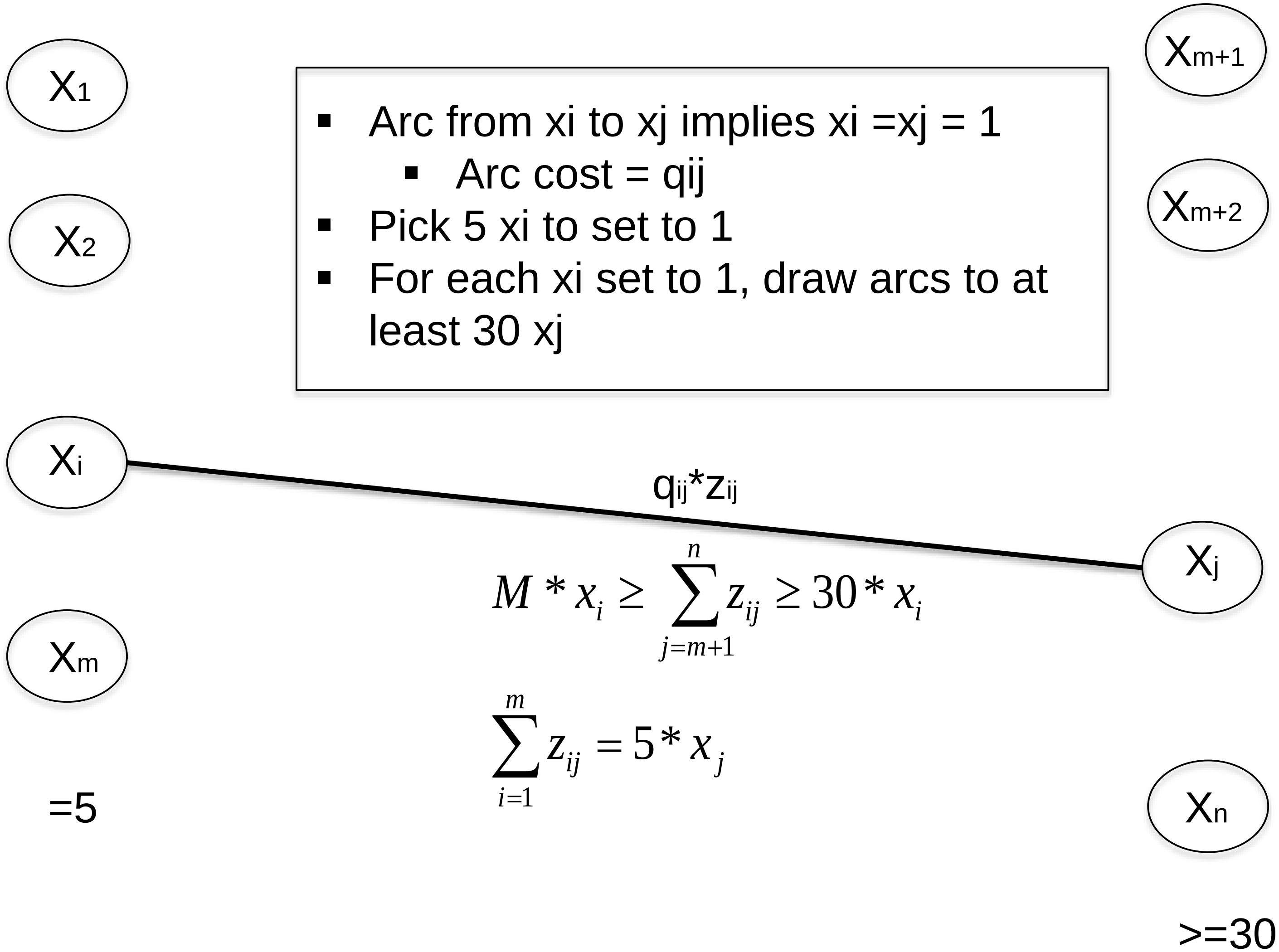
$$\textit{If } r_{i_1} \leq \ldots \leq r_{i_5} \leq \ldots r_{i_m} \textit{ then } \sum_{k=1}^{5} r_{i_k} \leq \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j$$

$$\Rightarrow \max\{\sum_{k=1}^{5} r_{i_k}, \sum_{k=1}^{30} p_{j_k}\} \leq \sum_{i=1}^{m} \sum_{j=m+1}^{n} x_i q_{ij} x_j$$



Example 1

- Alternate view of formulation: Bipartite graph



- Arc from xi to xj implies xi =xj = 1
  - Arc cost = qij
- Pick 5 xi to set to 1
- For each xi set to 1, draw arcs to at least 30 xj

$$M * x_i \geq \sum_{j=m+1}^{n} z_{ij} \geq 30 * x_i$$

$$\sum_{i=1}^{m} z_{ij} = 5 * x_j$$

Tightening the Formulation

- Common tactics for deriving additional cuts
  - Linear or logical combinations of constraints
    - Associate a graph with the model structure
  - Disjunctions
  - Solve one or more related models
  - Use infeasibility of model with superoptimal obj. value
    - CPLEX Conflict Refiner
  - Use solution objective value
    - Feasibility on model with objective value constraint added
    - Especially true for models with soft constraints

Source: Klotz, Newman. Practical Guidelines for Solving Difficult Mixed Integer Linear Programs

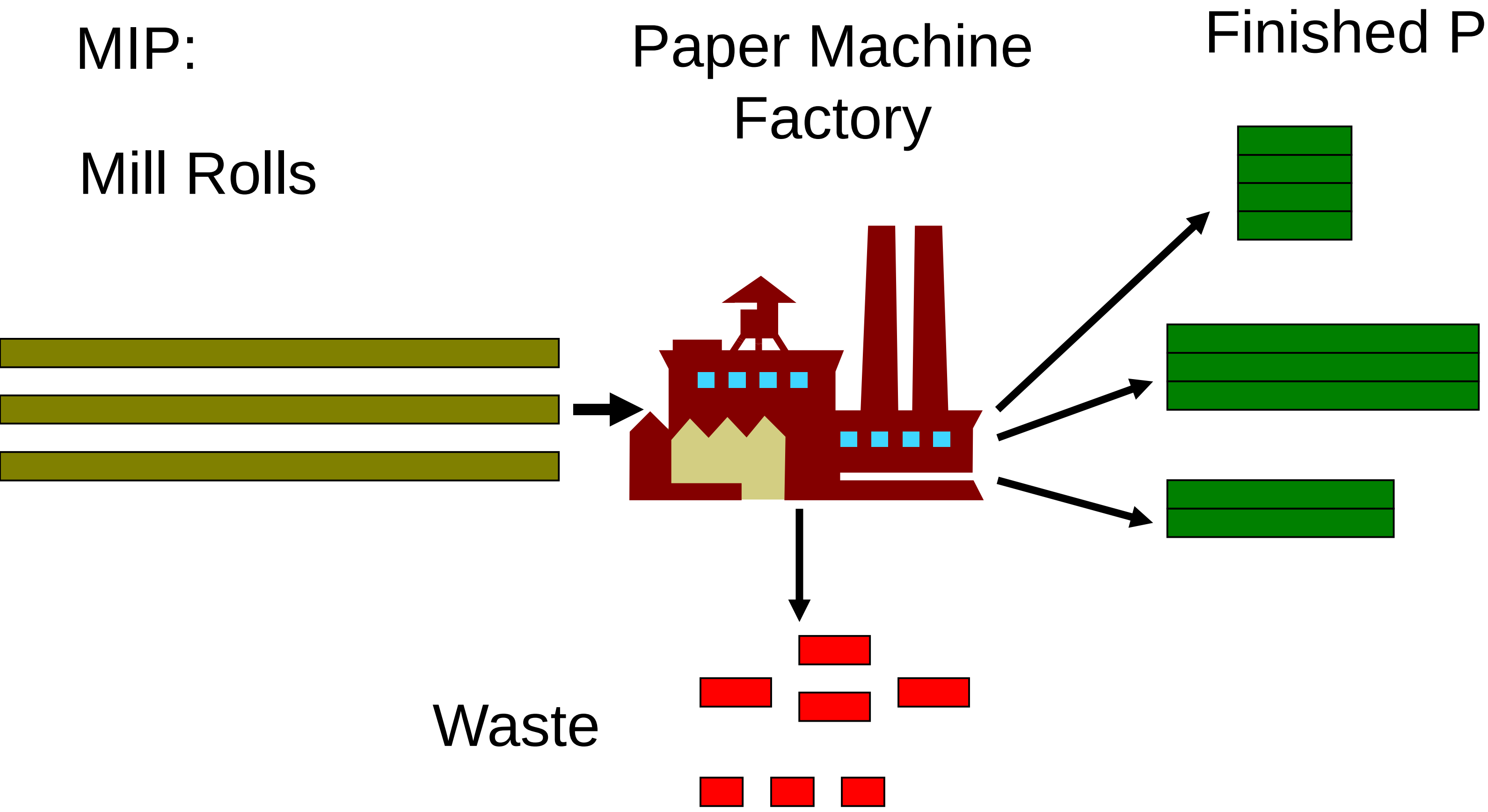
Tightening the Formulation

- Some examples of cheating by relaxing integrality

$$\begin{aligned} &Minimize\ t \\ &s.t. \\ &t \geq x \\ &4x = 3 \\ &x \geq 0, \text{integer} \end{aligned}$$

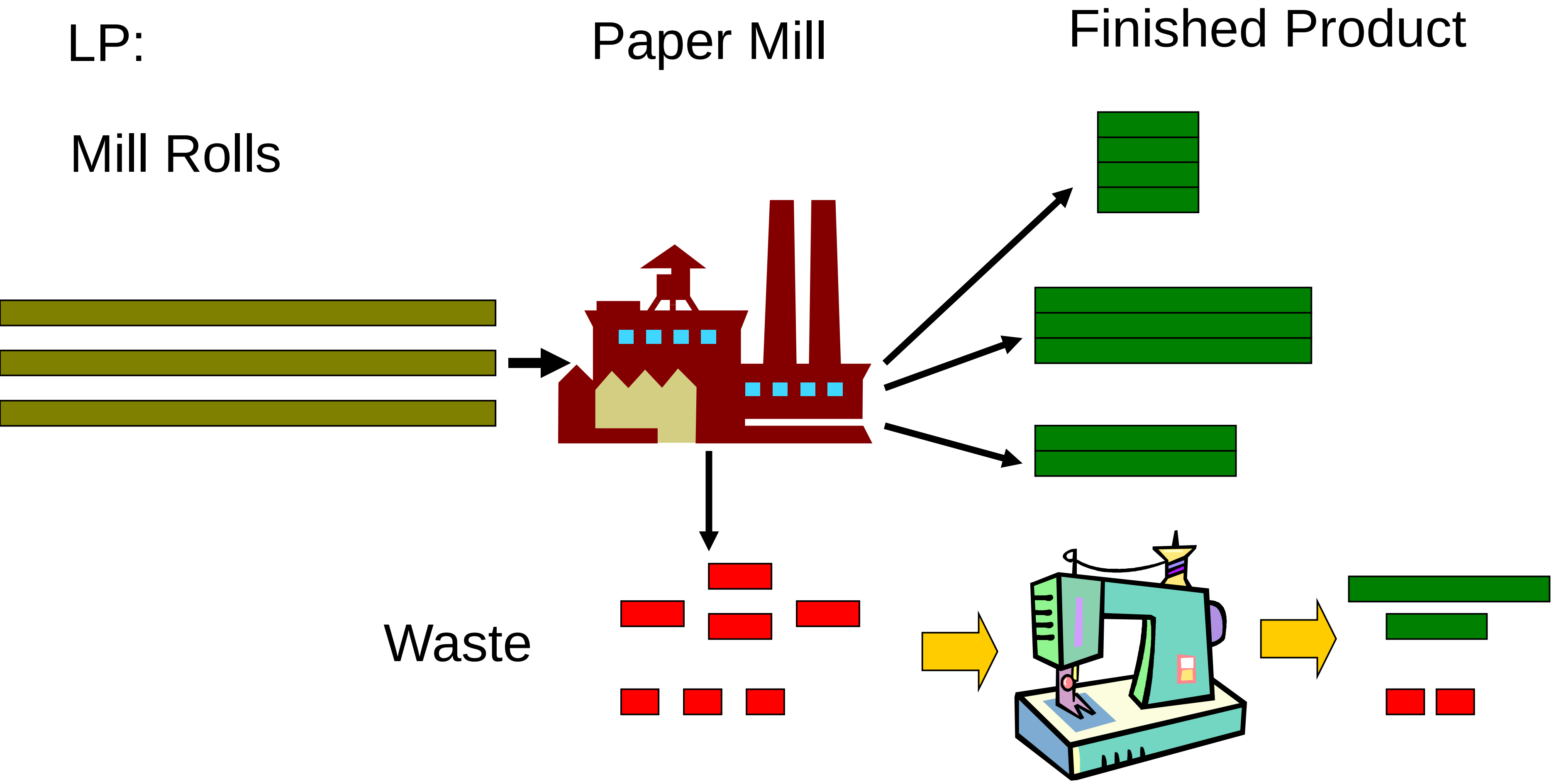
(MIP infeasible, LP has optimal objective 3/4

Another example of cheating: Cutting stock, known to have a weak compact formulation



Cutting stock(ctd)

LP:



IBM®

Legal Disclaimer

-
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:
  Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:
  All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to http://www.ibm.com/legal/copytrade.shtml for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:
  Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:
  Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:
  Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:
  Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:
  UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:
  Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.