Overview

Graph-based Approaches to Solving Binary Quadratic Programs

Ed Klotz, Ph.D.
Mathematical Optimization Specialist

May 2023

GUROBI
OPTIMIZATION

Agenda

Fundamentals

Linearizations of products of binaries
Padberg's BQP graph and cuts

The Maximum p-Dispersion Problem

The Open MIPLIB Model neos-2629914-sudost

Summary and Takeaways

Fundamentals

Fundamentals

- Problem formulation:

$$\min [\, c^T x] - [x^T Q x] \qquad // \textit{ no assumptions on convexity of } x^T Q x$$

$$[\, s.t. \;\; Ax \sim b \,]$$

$$[x^T Q_i x \sim b_i]$$

$$x_B \in \{0,1\} \qquad // \textit{ mostly interested in all binary variable case}$$

- Possibly nonconvex MIQ(C)P
- Can reformulate constraints into objective using penalties (QUBO)
  - Good formulation for Quantum Annealers, but Gurobi usually works better on original formulation
  - https://www.springerprofessional.de/en/quantum-bridge-analytics-i-a-tutorial-on-formulating-and-using-q/17436666

Fundamentals

- The Standard Linearization (Watters, 1967)
  - Linearize a convex or nonconvex MIQP into a MILP
    - Simplest linearization technique: do the following for each product of binaries in the model (PreQLinearize=1)
      - $z_{ij} = x_i x_j$

        $z_{ij} \leq x_i$

        $z_{ij} \leq x_j$ } (only need these two if objective pushes z_{ij} up)

        $z_{ij} \geq x_i + x_j - 1$ } (only need this one if objective pushes z_{ij} down)
      - Add the 3 linear constraints to the model
      - Replace each occurrence of $x_i x_j$ in the model with z_{ij}
    - We've transformed a (possibly nonconvex) MIQP into a MILP
      - Benefit from various Gurobi features available for MILP but not MIQP
      - Still no free lunch
        - We added 1-3 constraints for each product of binaries.

Fundamentals

- A less straightforward but more compact linearization (Glover, 1975)
  - Based on specialized soft knapsack constraints
    - Example (simplified from neos-911970)

Minimize
C0001 + C0002 + C0003 + C0004 + C0005 + C0006 + C0007 + C0008 + C0009
+ C0010 + C0011 + C0012 + C0013 + C0014 + C0015 + C0016 + C0017 + C0018
+ C0019 + C0020 + C0021 + C0022 + C0023 + C0024 + **C0025** + C0026 + C0027
+ C0028 + C0029 + C0030 + C0031 + C0032 + C0033 + C0034 + C0035 + C0036
+ C0037 + C0038 + C0039 + C0040 + C0041 + C0042 + C0043 + C0044 + C0045
+ C0046 + C0047 + C0048

Subject To
R0001a: **- C0025** + 4.2 B0673 + 6.5 B0697 + 5.95 B0721 <= 6.5

B0673 = B0697 = 1 --> C0025 = 4.2 + 6.5 - 6.5 = 4.2
B0673 = B0721 = 1 --> C0025 = 4.2 + 5.95 - 6.5 = 3.65
B0697 = B0721 = 1 --> C0025 = 6.5 + 5.95 - 6.5 = 5.95
B0673 = B0697 = B0721 = 1 -> C0025 = 4.2 + 5.95 = 10.15

R0001a provides a linear representation of
C0025 = 4.2 B0673*B0697 + 3.65 B0673*B0721 + 5.95 B0697*B0721 - 3.65 B0697*B0673*B0721

Penalty variable

Knapsack capacity matches largest knapsack weight

Could modify the coefficients to extend do handle larger multilinear terms

Could also modify to handle only bilinear terms

Fundamentals

- A more compact linearization for sum of bilinear terms sharing a common variable (Glover, 1975)

R0001': <u>**- C0025**</u> + 5.43 B0049 + 5.56 B0073 + 5.2 B0097 + 5.4 B0121 + 5 B0145
+ 4.39 B0169 + 4.07 B0193 + 4.56 B0217 + 4.03 B0241 + 3.3 B0265
+ 4.39 B0289 + 5.64 B0313 + 5.9 B0337 + 3.57 B0361 + 6.4 B0385
+ 3.94 B0409 + 4.5 B0433 + 4.67 B0457 + 3.88 B0481 + 4.18 B0505
+ 4.31 B0529 + 4.63 B0553 + 4.74 B0577 + 5.5 B0601 + 5.1 B0625
+ 5.1 B0649 + 4.2 B0673 + <u>**166.76 B0697**</u> + 5.95 B0721 + 5.88 B0745
+ 5.77 B0769 + 5.36 B0793 + 5.64 B0817 + 5.04 B0841 + 5.53 B0865 <= <u>**166.76**</u>

166.76 = sum of all knapsack weights except for B0697

- All sums of knapsack weights other than B0697 will be <= the rhs
  - → all multilinear expressions not involving B0697 contribute 0 violation to this soft constraint
- If B0697 = 1 and any other binary variable = 1 we get a contribution of the other binary variable's coefficient to the violation (e.g B0049 = 1 contributes 5.43 of violation).
- C0025= 5.43 B0049*B0697 + 5.56 B0073*B0697 + ... + 5.534 B0865*B0697
  - C0025 represents precisely a quadratic expression involving B0697 and other binaries
  - More compact than standard linearization when one binary variable appears in multiple bilinear terms
  - Works for bilinear terms in the objective but not in the constraints

Fundamentals

- A less straightforward but more compact linearization technique
  - C0025= 5.43 C0049*C0697 + 5.56 C0073*C0697 + ... + 5.534 C0865*C0697
    - C0025 represents precisely a quadratic expression involving C0697 and other binaries
  - Gurobi's PreQLinearize = 2 setting uses this to do a more compact linearization
  - $q_1 y * x_1 + \cdots + q_n y * x_n \;\; (y, x_j \; binary)$ is linearized as
    $q_1 x_1 + \cdots + q_n x_n + qy - p \leq q \qquad (q = \sum_{j=1}^{n} q_j) \qquad // \; q_j > 0.$

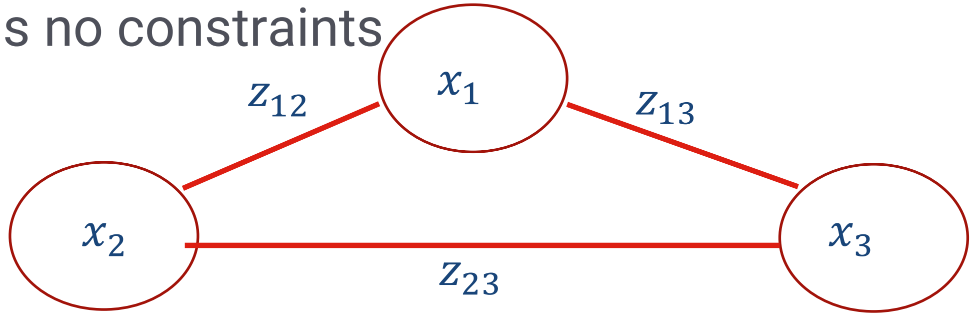
  - Replace every occurrence of $q_1 y * x_1 + \cdots + q_n y * x_n$ with p
  - Add the soft knapsack constraint
  - Added one constraint for the n bilinear terms associated with y

Fundamentals

- Summary of Gurobi PreQLinearize settings: No Free Lunch
  - PreQLinearize = 0: Convexify the nonconvex quadratic objective
    - Move from nonconvex MIQP to convex MIQP👍
    - No additional constraints👍
    - Miss out on MILP features absent from convex MIQP solver👎
    - Counterintuitive dual bound values that suggest possibly weak relaxations👎
  - PreQLinearize = 1: Linearize the nonconvex quadratic objective with new variable and constraints for each bilinear objective term
    - Move from nonconvex MIQP to MILP👍💪
      - Fairly strong MILP formulation
    - Each bilinear term in the quadratic objective introduces one new variable and 1-3 additional linear constraints👎
  - PreQLinearize = 2: Use Glover's Linearization
    - Move from nonconvex MIQP to MILP👍
    - Multiple bilinear terms modelled with one additional variable and constraint👍
    - Weaker MILP formulation 👎

Fundamentals

- **Padberg, The Boolean Quadric Polytope: Some Characteristics, Facets and Relatives**
  - Product graph associated with products of binary variables
  - Generate cuts even when the original problem has no constraints
    - Example: $\boldsymbol{x_1 + x_2 + x_3 - (z_{12} + z_{23} + z_{13}) \leq 1}$
    - Can prove by contradiction
    - Or by induction
      - Extends to cliques of larger size
    - Or by deriving as a zero half cut
      - But Padberg figured it out first
    - Or via facet defining inequalities
  - **Gurobi's BQP cut feature makes use of this with cliques of size 3**
    - **Traction for cut generation when model has few or no constraints**
    - **Gurobi 9.5 and later also considers cliques of size 4 or more**



The Maximum p-Dispersion Problem

The Maximum p-Dispersion Problem

- Given a set of n points with distances dij between points i and j, find the subset of k points that maximizes the sum of the distances

$$\begin{aligned} Max \ & \textstyle\sum_{i<j} d_{ij} x_i x_j \\ s.t. \ & \textstyle\sum_{j=1}^{n} x_j = k \\ & x_j \in \{0,1\} \end{aligned}$$

- Example discussed in Practical Guidelines for Solving Difficult MILPs (https://www.sciencedirect.com/science/article/abs/pii/S1876735413000020
- Broader discussion in http://yetanothermathprogrammingconsultant.blogspot.com/2019/06/maximum-dispersion.html

The Maximum p-Dispersion Problem

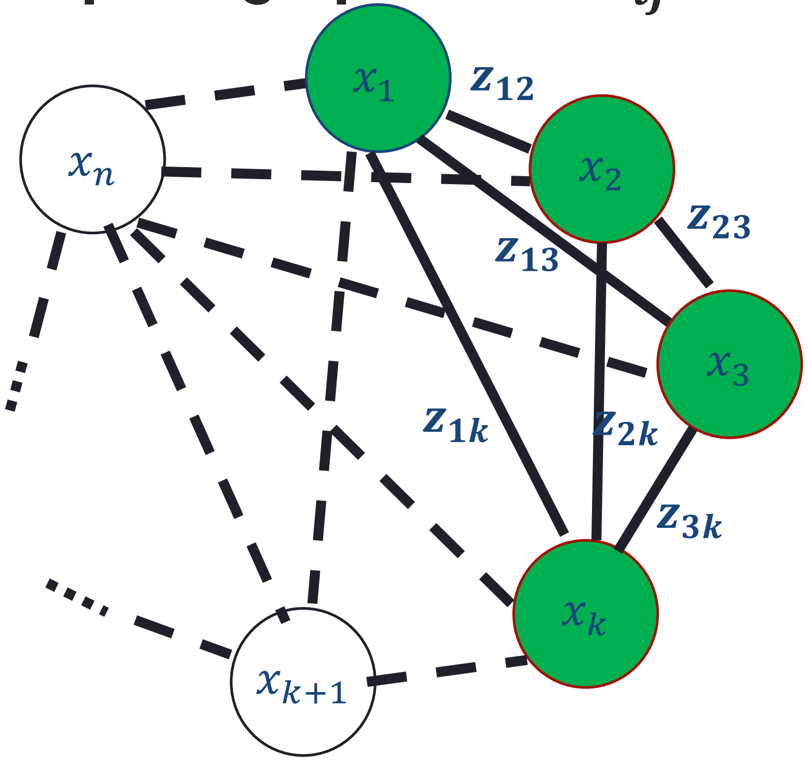
- LP relaxation feasible region is the convex hull of the integer feasible points
  - Won't be able to use LP-based polyhedral cuts on this direct formulation
  - Previous results indicate just linearizing (PreQLinearize=1) is better, but still not particularly effective given the problem size
- http://yetanothermathprogrammingconsultant.blogspot.com/2019/06/maximum-dispersion.html describes multiple ways to derive a single cut that uses both original x binary variables and the linearization variables z
- $$\begin{aligned} Max \ & \textstyle\sum_{i<j} d_{ij} x_i x_j \\ s.t. \ & \textstyle\sum_{j=1}^{n} x_j = k \\ & x_j \in \{0,1\} \end{aligned}$$

  $$\begin{aligned} Max \ & \textstyle\sum_{i<j} d_{ij} z_{ij} \\ s.t. \ & \textstyle\sum_{j=1}^{n} x_j = k \\ & \text{<linearization constraints>} \\ & \textstyle\sum_{i<j} z_{ij} = k*(k-1)/2 \\ & x_j \in \{0,1\} \end{aligned}$$
- Run time with cut drops from hours to < a minute
- Blog describes multiple ways to derive this cut, but how do we do it generically in a way that extends to other models?

The Maximum p-Dispersion Problem

- **Padberg graph for our dispersion problem**

Complete graph since $d_{ij} > 0$



$$Max \sum_{i<j} d_{ij} z_{ij}$$

$$s.t. \quad \sum_{j=1}^{n} x_j = k$$

<linearization constraints>

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

$$x_j \in \{0, 1\}$$

- **Given that k of the x variables must be 1, how many of the z variables must be 1?**
  - WLOG, set the first k x variables to 1
  - Induces a complete subgraph on the green nodes associated with $x_1, \ldots, x_k$
  - Each edge in the subgraph identifies a z variable that must be 1
  - There are $\boldsymbol{k} * (\boldsymbol{k} - \mathbf{1})/2$ such edges

Source: http://orwe-conference.mines.edu/files/IOS2018SpatialPerfTuning.pdf

The Open MIPLIB Model neos-2629914-sudost

neos-2629914-sudost

Info:

	Original	Presolved
Variables	496	496
Constraints	51872	51872
Binaries	256	256
Integers	0	0
Continuous	240	240

	ID	Objective	Exact	Int. Viol	Cons. Viol	Obj. Viol	Submitter	Date	Description
2	2	48180		0	0	0	Ed Klotz	2019-11-15	Found with a customized approach using the Optimization Direct Heuristic and CPLEX 12.9
1	1	48212	48212	0	0	0	-	2018-10-16	Solution found during MIPLIB2017 problem selection.

Gurobi 10.0 finds this in ~18 minutes with MIPFocus = 1, aggressive symmetry and presolve, RINS every 50 nodes

Robert Ashford reports recent ODH/CPLEX with similar settings finds the solution in ~26 minutes

neos-2629914-sudost

Gurobi 10.0, default parameters:

```
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 39622.0000    0   56 48528.0000 39622.0000  18.4%     -    1s
     0     0 39622.0000    0   58 48528.0000 39622.0000  18.4%     -    1s
     0     0                    48454.000000 39622.0000  18.2%     -    2s
     0     0 39622.0000    0   49 48454.0000 39622.0000  18.2%     -    4s
     0     0 39622.0000    0   59 48454.0000 39622.0000  18.2%     -    5s


3916280 2894735 44540.5192   49  173 48220.0000 42081.3281  12.7%  89.5 123981s
3916790 2895156 45421.8383   97  163 48220.0000 42081.3686  12.7%  89.5 123997s
3917460 2895510 46202.5125   71  158 48220.0000 42081.3908  12.7%  89.5 124012s
3917920 2895952 42768.4602   52  183 48220.0000 42081.4286  12.7%  89.5 124028s
```

As we shall see, modest progress in gap is not as meaningful as it might seem.

Rapid rate of growth in active node list indicates unlikely to prove optimality in our lifetimes

Examine the model

Objective involves minimizing the 240 continuous variables

Minimize
13 C0001 + 14 C0002 + 16 C0003 + 18 C0004 + 15 C0005 + 16 C0006 + 16 C0007
+ 15 C0008 + 11 C0009 + 14 C0010 + 16 C0011 + 11 C0012 + 15 C0013
+ 14 C0014 + 15 C0015 + 13 C0016 + 16 C0017 + 13 C0018 + 17 C0019 + …
… + 15 C0224 + 18 C0225 + 15 C0226 + 12 C0227 + 16 C0228 + 18 C0229
+ 13 C0230 + 14 C0231 + 16 C0232 + 14 C0233 + 15 C0234 + 15 C0235
+ 18 C0236 + 15 C0237 + 14 C0238 + 13 C0239 + 18 C0240

Bounds
C0001 >= 11
C0002 >= 11
…
C0239 >= 11
C0240 >= 11

Even with no constraints in the model, the objective value will be $\sum_{j=1}^{240} c_j l_j$ = 39622

Examine the model

Constraint group 1: sums of binaries = 1

R0001: B0241 + B0242 + B0243 + B0244 + B0245 + B0246 + B0247 + B0248
+ B0249 + B0250 + B0251 + B0252 + B0253 + B0254 + B0255 + B0256 = 1
R0002: B0257 + B0258 + B0259 + B0260 + B0261 + B0262 + B0263 + B0264
+ B0265 + B0266 + B0267 + B0268 + B0269 + B0270 + B0271 + B0272 = 1
R0003: B0273 + B0274 + B0275 + B0276 + B0277 + B0278 + B0279 + B0280
+ B0281 + B0282 + B0283 + B0284 + B0285 + B0286 + B0287 + B0288 = 1
…
R0015: B0465 + B0466 + B0467 + B0468 + B0469 + B0470 + B0471 + B0472
+ B0473 + B0474 + B0475 + B0476 + B0477 + B0478 + B0479 + B0480 = 1
R0016: B0481 + B0482 + B0483 + B0484 + B0485 + B0486 + B0487 + B0488
+ B0489 + B0490 + B0491 + B0492 + B0493 + B0494 + B0495 + B0496 = 1
R0017: B0241 + B0257 + B0273 + B0289 + B0305 + B0321 + B0337 + B0353
+ B0369 + B0385 + B0401 + B0417 + B0433 + B0449 + B0465 + B0481 = 1
R0018: B0242 + B0258 + B0274 + B0290 + B0306 + B0322 + B0338 + B0354
+ B0370 + B0386 + B0402 + B0418 + B0434 + B0450 + B0466 + B0482 = 1
…
R0031: B0255 + B0271 + B0287 + B0303 + B0319 + B0335 + B0351 + B0367
+ B0383 + B0399 + B0415 + B0431 + B0447 + B0463 + B0479 + B0495 = 1
R0032: B0256 + B0272 + B0288 + B0304 + B0320 + B0336 + B0352 + B0368
+ B0384 + B0400 + B0416 + B0432 + B0448 + B0464 + B0480 + B0496 = 1

Examine the model

Constraint group 1: sums of binaries = 1

- Easier to visualize as a 16 x 16 grid of binaries whose rows and columns sum to 1

B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255	B256	= 1
B257	B258	B259	B260	B261	B262	B263	B264	B265	B266	B267	B268	B269	B270	B271	B272	= 1
B273	B274	B275	B276	B277	B278	B279	B280	B281	B282	B283	B284	B285	B286	B287	B288	
B289	B290	B291	B292	B293	B294	B295	B296	B297	B298	B299	B300	B301	B302	B303	B304	
B305	B306	B307	B308	B309	B310	B311	B312	B313	B314	B315	B316	B317	B318	B319	B320	
B321	B322	B323	B324	B325	B326	B327	B328	B329	B330	B331	B332	B333	B334	B335	B336	
B337	B338	B339	B340	B342	B342	B343	B344	B345	B346	B347	B348	B349	B350	B351	B352	.
B353	B354	B355	B356	B357	B358	B359	B360	B361	B362	B363	B364	B365	B366	B367	B368	.
B369	B370	B371	B372	B373	B374	B375	B376	B377	B378	B379	B380	B381	B382	B383	B384	.
B385	B386	B387	B388	B389	B390	B391	B391	B393	B394	B395	B396	B397	B398	B399	B400	
B401	B402	B403	B404	B405	B406	B407	B408	B409	B410	B411	B412	B413	B414	B415	B416	
B417	B418	B419	B420	B421	B422	B423	B424	B425	B426	B427	B428	B429	B430	B431	B432	
B433	B434	B435	B436	B437	B438	B439	B440	B441	B442	B443	B444	B445	B446	B447	B448	
B449	B450	B451	B452	B453	B454	B455	B456	B457	B458	B459	B460	B461	B462	B463	B464	
B465	B466	B467	B468	B469	B470	B471	B472	B473	B474	B475	B476	B477	B478	B479	B480	
B481	B482	B483	B484	B485	B486	B487	B488	B489	B490	B491	B492	B493	B494	B495	B496	= 1
= 1	= 1						...								= 1	

Examine the model

Constraint group 2: look familiar?

R0033: C0001 - 12 B0241 - 12 B0259 >= -12 ⟷ C0001 >= 12(B0241 + B0259 – 1) ⟷ C0001 >= 12 B0241*B0259

R0034: C0001 - 12 B0241 - 12 B0260 >= -12

R0035: C0001 - 13 B0241 - 13 B0261 >= -13

$z_{ij} \geq x_i + x_j - 1$ } (only need this one if objective pushes z_{ij} down)

…

R0231: C0001 - 13 B0256 - 13 B0268 >= -13

R0232: C0001 - 12 B0256 - 12 B0269 >= -12

R0233: C0002 - 12 B0241 - 12 B0275 >= -12

R0234: C0002 - 12 B0241 - 12 B0276 >= -12

The standard linearization

…

R48031: C0240 - 13 B0476 - 13 B0496 >= -13

R48032: C0240 - 12 B0477 - 12 B0496 >= -12

Examine the model

Constraint group 3: look (sort of) familiar?

R48033: C0001 - **224 B0241** + 213 B0258 + 212 B0259 + 212 B0260 + 211 B0261
+ 210 B0262 + 210 B0263 + 209 B0264 + 211 B0265 + 209 B0266 + 208 B0267
+ 207 B0268 + 206 B0269 + 205 B0270 + 207 B0271 + 206 B0272 >= 0
R48034: C0001 - **210 B0242** + 199 B0257 + 199 B0259 + 199 B0260 + 198 B0261
+ 197 B0262 + 197 B0263 + 196 B0264 + 198 B0265 + 196 B0266 + 195 B0267
+ 194 B0268 + 193 B0269 + 192 B0270 + 194 B0271 + 193 B0272 >= 0
R48035: C0001 - **200 B0243** + 188 B0257 + 189 B0258 + 188 B0260 + 189 B0261
+ 188 B0262 + 188 B0263 + 187 B0264 + 189 B0265 + 187 B0266 + 186 B0267
+ 185 B0268 + 184 B0269 + 183 B0270 + 185 B0271 + 184 B0272 >= 0
…
R48048: C0001 - **212 B0256** + 194 B0257 + 195 B0258 + 196 B0259 + 196 B0260
+ 197 B0261 + 198 B0262 + 198 B0263 + 199 B0264 + 195 B0265 + 199 B0266
+ 200 B0267 + 199 B0268 + 200 B0269 + 201 B0270 + 201 B0271 >= 0
R48049: C0002 - 224 B0241 + 213 B0274 + 212 B0275 + 212 B0276 + 211 B0277
+ 210 B0278 + 210 B0279 + 209 B0280 + 211 B0281 + 209 B0282 + 208 B0283
+ 207 B0284 + 206 B0285 + 205 B0286 + 207 B0287 + 206 B0288 >= 0

Each objective variable C0001,…,C0240 appears in 16 such constraints

Examine the model

Constraint group 3: look (sort of) familiar?

B241	**B242**	**B243**	**B244**	**B245**	**B246**	**B247**	**B248**	**B249**	**B250**	**B251**	**B252**	**B253**	**B254**	**B255**	**B256**	= 1
B257	B258	B259	B260	B261	B262	B263	B264	B265	B266	B267	B268	B269	B270	B271	B272	= 1

R48033: C0001 - **224 B0241** + 213 B0258 + 212 B0259 + 212 B0260 + 211 B0261
+ 210 B0262 + 210 B0263 + 209 B0264 + 211 B0265 + 209 B0266 + 208 B0267
+ 207 B0268 + 206 B0269 + 205 B0270 + 207 B0271 + 206 B0272 >= 0

R48034: C0001 - **210 B0242** + 199 B0257 + 199 B0259 + 199 B0260 + 198 B0261
+ 197 B0262 + 197 B0263 + 196 B0264 + 198 B0265 + 196 B0266 + 195 B0267
+ 194 B0268 + 193 B0269 + 192 B0270 + 194 B0271 + 193 B0272 >= 0

…

R48048: C0001 - **212 B0256** + 194 B0257 + 195 B0258 + 196 B0259 + 196 B0260
+ 197 B0261 + 198 B0262 + 198 B0263 + 199 B0264 + 195 B0265 + 199 B0266
+ 200 B0267 + 199 B0268 + 200 B0269 + 201 B0270 + 201 B0271 >= 0

Examine the model

Constraint group 3: look (sort of) familiar?

B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255	B256	= 1
B257	B258	B259	B260	B261	B262	B263	B264	B265	B266	B267	B268	B269	B270	B271	B272	= 1

R48033: C0001 >= **224 B0241** - 213 B0258 - 212 B0259 - 212 B0260 - 211 B0261
- 210 B0262 - 210 B0263 - 209 B0264 - 211 B0265 - 209 B0266 - 208 B0267
- 207 B0268 - 206 B0269 - 205 B0270 - 207 B0271 - 206 B0272

Constraint Group 2:

			Bounds
B0241 = 1, B0258 = 1 → C0001 >= 11	⟷ C0001 >= 11 B0241*B0258	---	C0001 >= 11
B0241 = 1, B0259 = 1 → C0001 >= 12	⟷ C0001 >= 12 B0241*B0259	R0033: C0001 - 12 B0241 - 12 B0259 >= -12	
		R0034: C0001 - 12 B0241 - 12 B0260 >= -12	
…			
B0241 = 1, B0272 = 1 → C0001 >= 18	⟷ C0001 >= 18 B0241*B0272	R0046: C0001 - 18 B0241 - 18 B0272 >= -18	

C0001 >= max{11 B0241*B0258, 12 B0241*B0259,…, 18 B0241*B0272}

neos-2629914-sudost

Model belongs in QPLIB, not MIPLIB

Summary

- Minimization problem
- All objective variables have lower bounds of 11
- Constraint group 1: 16x16 grid of binaries whose row and column sums = 1
- Constraint groups 2 and 3 express linearizations of products of binary variables analogous to Gurobi's PreQLinearize = 1 and 2 settings
  - Removing one of these groups has no impact on the dual bound
  - Removing both of these groups enables the model to solve instantly with all objective variables at their lower bounds of 11
- Model appears to minimize the sum of 240 different minimax functions
  - Or does it?

neos-2629914-sudost

Additional simplifications

- Does model really minimize the sum of 240 different minimax functions
    - Let's take a closer look at the minimax functions
        - Consider group 3 constraints for C0001

R48033: C0001 - **224 B0241** + 213 B0258 + 212 B0259 + 212 B0260 + 211 B0261
+ 210 B0262 + 210 B0263 + 209 B0264 + 211 B0265 + 209 B0266 + 208 B0267
+ 207 B0268 + 206 B0269 + 205 B0270 + 207 B0271 + 206 B0272 >= 0
R48034: C0001 - **210 B0242** + 199 B0257 + 199 B0259 + 199 B0260 + 198 B0261
+ 197 B0262 + 197 B0263 + 196 B0264 + 198 B0265 + 196 B0266 + 195 B0267
+ 194 B0268 + 193 B0269 + 192 B0270 + 194 B0271 + 193 B0272 >= 0
…
R48048: C0001 - **212 B0256** + 194 B0257 + 195 B0258 + 196 B0259 + 196 B0260
+ 197 B0261 + 198 B0262 + 198 B0263 + 199 B0264 + 195 B0265 + 199 B0266
+ 200 B0267 + 199 B0268 + 200 B0269 + 201 B0270 + 201 B0271 >= 0

Additional Simplifications

R48033: C0001 - **224 B0241** + 213 B0258 + 212 B0259 + 212 B0260 + 211 B0261 + 210 B0262 + 210 B0263 + 209 B0264 + 211 B0265 + 209 B0266 + 208 B0267 + 207 B0268 + 206 B0269 + 205 B0270 + 207 B0271 + 206 B0272 >= 0

R48034: C0001 - **210 B0242** + 199 B0257 + 199 B0259 + 199 B0260 + 198 B0261 + 197 B0262 + 197 B0263 + 196 B0264 + 198 B0265 + 196 B0266 + 195 B0267 + 194 B0268 + 193 B0269 + 192 B0270 + 194 B0271 + 193 B0272 >= 0

…

R48048: C0001 - **212 B0256** + 194 B0257 + 195 B0258 + 196 B0259 + 196 B0260 + 197 B0261 + 198 B0262 + 198 B0263 + 199 B0264 + 195 B0265 + 199 B0266 + 200 B0267 + 199 B0268 + 200 B0269 + 201 B0270 + 201 B0271 >= 0

Each constraint selects one variable from row 1 of the grid, considers its bilinear terms with all variable in row 2 of the grid

Exactly one bilinear term from all 16 constraints will have xi*xj = 1
(xi from grid row 1; xj from grid row 2)

B241	**B242**	**B243**	**B244**	**B245**	**B246**	**B247**	**B248**	**B249**	**B250**	**B251**	**B252**	**B253**	**B254**	**B255**	**B256**	= 1
B257	B258	B259	B260	B261	B262	B263	B264	B265	B266	B267	B268	B269	B270	B271	B272	= 1

Additional Simplifications

$\text{C0001} \geq q_{ij}x_ix_j \quad i \in G1, j \in G2$

(only 1 $x_ix_j = 1$)

$\text{C0001} \geq \sum_{i \in G1} \sum_{j \in G2} q_{ij}x_ix_j$

[11,inf) [0,U]

Dual argument; C0001 intersects no other group 3 constraints

$\text{C0001} = \sum_{i \in G1} \sum_{j \in G2} q_{ij}x_ix_j$

Remove group 3 constraints for C0001, put this equality in the objective

Minimize

$13 \sum_{i \in G1} \sum_{j \in G2} q_{ij}x_ix_j$ + 14 C0002 + 16 C0003 + 18 C0004 + 15 C0005 + 16 C0006 + 16 C0007
+ 15 C0008 + 11 C0009 + 14 C0010 + 16 C0011 + 11 C0012 + 15 C0013
+ 14 C0014 + 15 C0015 + 13 C0016 + 16 C0017 + 13 C0018 + 17 C0019 + …
… + 15 C0224 + 18 C0225 + 15 C0226 + 12 C0227 + 16 C0228 + 18 C0229
+ 13 C0230 + 14 C0231 + 16 C0232 + 14 C0233 + 15 C0234 + 15 C0235
+ 18 C0236 + 15 C0237 + 14 C0238 + 13 C0239 + 18 C0240

Repeat for C0002, C0003, …, C0240

Group 2 constraints linearize the same QCs, so they too can be removed

Additional Simplifications

$n = 16, N = \{1 \ldots n^2\}$

Minimize $\sum_{i \in N} \sum_{j \in N} q'_{ij} x_i x_j.$

s.t.

<Grid constraints>

$\sum_{i \in Gk} \sum_{j \in Gl} q_{ij} x_i x_j \geq 11$ // bounds on objective variables that have been substituted out

x binary

Model is a quadratic assignment problem (QAP) except for these constraints

We can get remove these constraints by first doing a change of variables C’I = Ci – 11 for the objective variables in the original model

Generic QAP

- After the change of variables:

$$Minimize \sum_{i=1}^{n^2} \sum_{j=1}^{n^2} q_{ij} x_{ij}$$

s.t.

$$\sum_{j=1}^{n} x_{kn+j} = 1 \quad k = 0, \ldots, n-1$$

$$\sum_{k=0}^{n-1} x_{kn+j} = 1 \quad \mathrm{j} = 1, \ldots, \mathrm{n}$$

$$x \in \{0,1\}^{n^2}$$

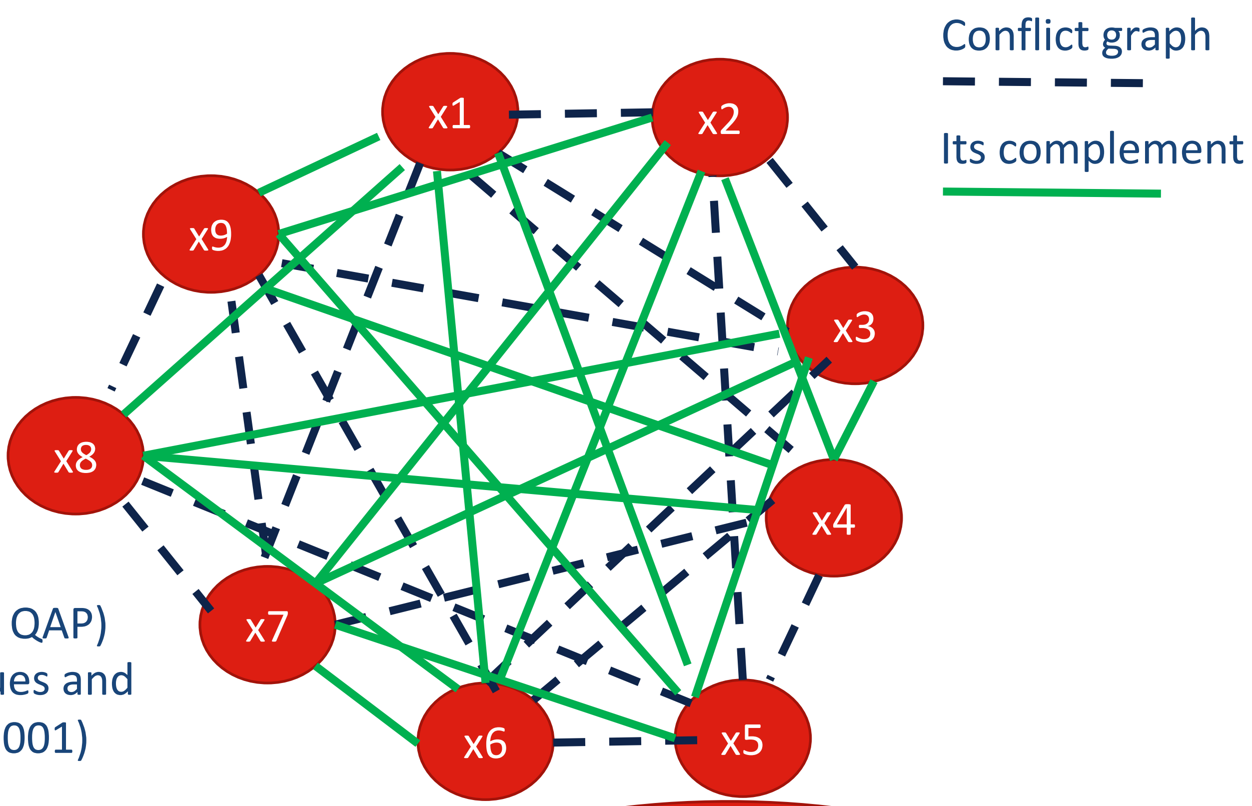
B241	B242	B243	B244	B245	B246	B247	B248	B249	B250	B251	B252	B253	B254	B255	B256	= 1
B257	B258	B259	B260	B261	B262	B263	B264	B265	B266	B267	B268	B269	B270	B271	B272	= 1
B273	B274	B275	B276	B277	B278	B279	B280	B281	B282	B283	B284	B285	B286	B287	B288	
B289	B290	B291	B292	B293	B294	B295	B296	B297	B298	B299	B300	B301	B302	B303	B304	
B305	B306	B307	B308	B309	B310	B311	B312	B313	B314	B315	B316	B317	B318	B319	B320	
B321	B322	B323	B324	B325	B326	B327	B328	B329	B330	B331	B332	B333	B334	B335	B336	
B337	B338	B339	B340	B342	B342	B343	B344	B345	B346	B347	B348	B349	B350	B351	B352	.
B353	B354	B355	B356	B357	B358	B359	B360	B361	B362	B363	B364	B365	B366	B367	B368	.
B369	B370	B371	B372	B373	B374	B375	B376	B377	B378	B379	B380	B381	B382	B383	B384	.
B385	B386	B387	B388	B389	B390	B391	B391	B393	B394	B395	B396	B397	B398	B399	B400	
B401	B402	B403	B404	B405	B406	B407	B408	B409	B410	B411	B412	B413	B414	B415	B416	
B417	B418	B419	B420	B421	B422	B423	B424	B425	B426	B427	B428	B429	B430	B431	B432	
B433	B434	B435	B436	B437	B438	B439	B440	B441	B442	B443	B444	B445	B446	B447	B448	
B449	B450	B451	B452	B453	B454	B455	B456	B457	B458	B459	B460	B461	B462	B463	B464	
B465	B466	B467	B468	B469	B470	B471	B472	B473	B474	B475	B476	B477	B478	B479	B480	
B481	B482	B483	B484	B485	B486	B487	B488	B489	B490	B491	B492	B493	B494	B495	B496	= 1
= 1	= 1						...								= 1	

Good News: It's a QAP
- Take advantage of all sorts of results in the literature

Bad News: It's a QAP
- NP Hard
- Branch and Cut not particularly effective as size increases

Grid structure

x1	x2	x3	=1
x4	x5	x6	=1
x7	x8	x9	=1
=1	=1	=1	



1. Only constraints in the model are the grid (e.g., QAP)
   - One to one correspondence between cliques and integer feasible solutions (Junger, Kaibel, 2001)
2. Additional constraints besides grid
   - Cliques may be useful for heuristics
3. In both cases we may be able to use the complement of the conflict graph for cuts.
   - It is a superset of the Padberg graph

Could use this to calculate the minmum number of linearization variables that must be 1

Generic QAP

- **How many linearization variables must be 1?**
  - SubMIP solve for case where $Q_{ij} \geq 0$:

$$\begin{aligned} &\min c^T x + x^T Q x \\ &[\, s.t.\ \ Ax \sim b \,] \\ &x \in \{0,1\} \end{aligned}$$

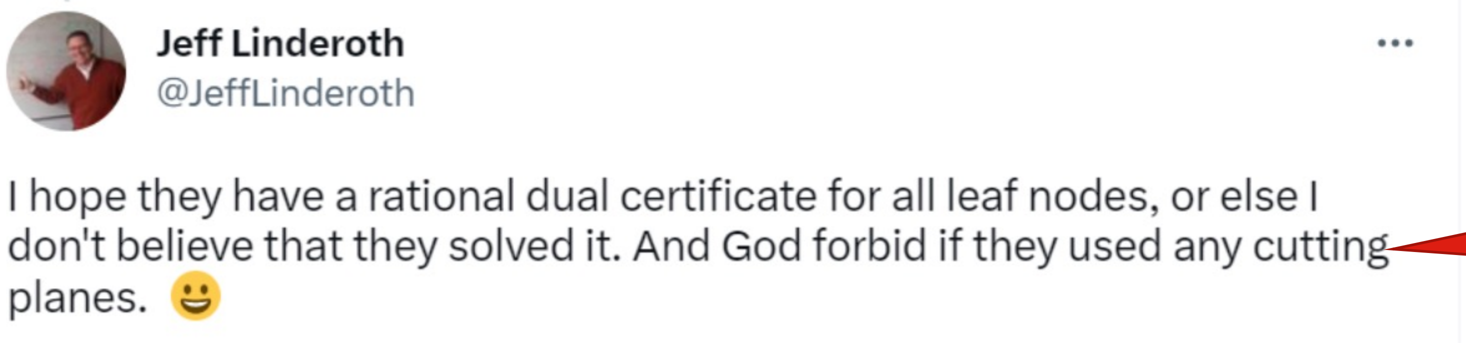
$$\begin{aligned} &\min c^T x + d^T z \\ &[\, s.t.\ \ Ax \sim b \,] \\ &E_1 x + E_2 z \leq p \\ &x, z \in \{0,1\} \end{aligned}$$

$$\begin{aligned} &\min e^T z \\ &[\, s.t.\ \ Ax \sim b \,] \\ &E_1 x + E_2 z \leq p \\ &x, z \in \{0,1\} \end{aligned}$$

  - Cardinality cut: $\boldsymbol{e}^T \boldsymbol{z} \geq \boldsymbol{z}^*$
  - Solve time for subMIP associated with reformulated QAP: 2 seconds
  - Solve time for QAP with subMIP cut (Gurobi defaults): ~34 hours, ~1800 nodes
  - Solve time with NodeMethod=2 to invoke barrier at the nodes: ~8 hours, ~1300 nodes
  - Time to demote the associated MIPLIB model from open to hard

Validation

- **We solved this model with two reformulations and some strengthening**
  - Change of variables
  - Reformulation from MIQCP to QAP
  - Added one cut to get the dual bound to move fast enough to solve to optimality
- **How can we check each step?**
  - We may have some skeptics in the audience with stringent requirements

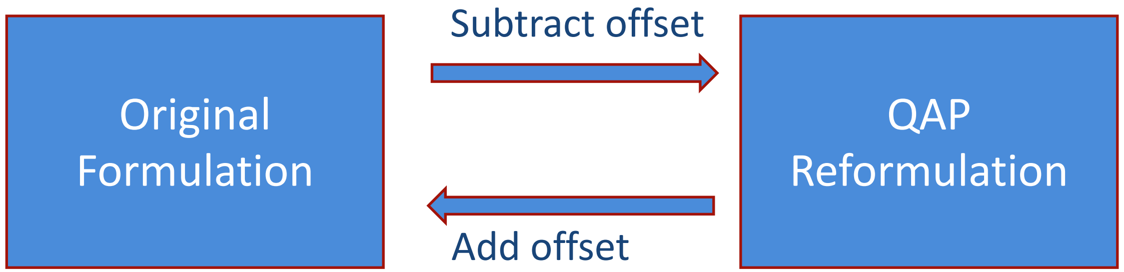


That's a pretty high bar (high jump, not limbo), but there are some simpler sanity checks we should always do

Validation

- **Sanity check for the reformulations**
  - Any solution from the reformulated QAP should have an objective 39622 (the implicit constant term removed by the reformulation) larger if used as a MIP start for the original model
    - Example: Optimal solution to the QAP:

      1244 0 cutoff 17 8558.00000 8541.50629 0.19% 2870 31424s
    - Add 39622 to 8558: 48180, the objective value of the QAP MIP start when given to the original formulation
    - Similar comparisons for other solutions, always consistent
    - Sorry Jeff, I didn't run the solution pool to enumerate all integer feasible solutions and test each one



Validation

- **Sanity check for primal bound**
  - Feed the optimal solution with objective 48180 to the original model
  - Run with aggressive settings to find additional feasible solutions

```
>>> m.setParam("NoRelHeurTime", 10800)
Set parameter NoRelHeurTime to value 10800
>>> m.setParam("Heuristics", .5)
Set parameter Heuristics to value 0.5
>>> m.setParam("ImproveStartTime", 144000)
Set parameter ImproveStartTime to value 144000
>>> m.setParam("MIPFocus", 1)
```

  - If Gurobi finds a better solution, we have an error in our reformulations or cut to investigate
    - That did not happen so far:

```
5602011 4148324 48119.7857  116   79 48180.0000 42200.0586  12.4%  89.7 286536s
5602481 4148966 44647.9308   60  168 48180.0000 42200.0797  12.4%  89.7 286562s
5603343 4149580 43111.6047   72  171 48180.0000 42200.1301  12.4%  89.7 286588s
```


Validation

- **Sanity check for dual bound**
  - The one cut we added was globally valid; it did not rely on dual based arguments
  - Reverse the direction of the cut from $\sum z_{ij} \geq 100$ to $\sum z_{ij} \leq 99$
  - Confirm that no integer feasible solutions are found
    - So far so good:

```
1635016 1633989 7697.31680   59 1001          - 7502.81491      -  3119 1573040s
1635024 1633997 7949.39134   87 1177          - 7502.81639      -  3119 1573052s
1635032 1634005 7939.24188  109 1036          - 7502.81800      -  3119 1573058s
1635036 1634009 8162.00000   34  267          - 7502.81818      -  3119 1573062s
```


Summary and Takeaways

Summary

- **neos-2629914-sudost can be demoted from open to hard**
  - Recognize that the original MILP formulation is a linearized version of a MIQCP
  - Change of variables to remove the implicit constant term of the objective and enable more substitutions
  - Use the grid of binary variables to visualize the model and recognize a formulation to a QAP
  - Add a cut based on a fast solving subMIP that minimizes the sum of linearization variables (for the QAP, but not for the original model)
  - But we still were somewhat fortunate that the size of the QAP wasn't bigger

Takeaways

GUROBI OPTIMIZATION

- **MILPs may have products of binary variables in disguise**
  - Models involving notion of overlap
  - Sharing info from a MILP and MIQP (and vice versa) formulation may help
- **Underutilized generic structures involving binary variables**
  - Complement of the conflict graph
  - Grids of binary variables with constraints on the rows and columns
- **Gurobi Python API well suited to the tasks associated with model reformulation, graph constructs, tightening formulations**
  - Programs used will be available at https://github.com/Gurobi/techtalks/tree/main/554 mipformulations/programs
  - Networkx Python package for graph operations
  - Other solvers and modeling language also have Python API with similar modeling constructs

Questions?

References

(Background)

- L. Watters. Reduction of integer polynomial programming to zero-one linear programming problems. Operations Research, 15(6):1171–1174, December 1967.
- F. Glover, “Improved linear integer programming formulations of nonlinear integer Problems,” Management Science, vol. 22, no. 4, pp. 455–460, 1975.
- Forrester, R.,and N. Hunt-Isaak, "Computational Comparison of Exact Solution Methods for 0-1 Quadratic Programs: Recommendations for Practitioners," *Journal of Applied Mathematics*, Vol. 2020, Article ID 5974820, 21 pages, 2020.

(Practice)

- **http://yetanothermathprogrammingconsultant.blogspot.com/2019/06/maximum-dispersion.html**
- Klotz, E. “Specialized Strategies for Products of Binary Variables”, https://www.gurobi.com/events/models-with-products-of-binary-variables/

(Theory)

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

Backup Material

Question from a Gurobi Tech Talk

GUROBI
OPTIMIZATION

Which book includes the following text?
"…working out the most efficient way to pack up and down the trucks, since saving one truck … could save something in the region of $100,000"

A) Mason, "Inside Out: A Personal History of Pink Floyd"
B) Woolsey, "Real World Operations Research: The Woolsey Papers"
C) Applegate et al., "The Traveling Salesman Problem"
D) Davenport, "Competing on Analytics"

(https://www.gurobi.com/events/holiday-tech-talk-santas-bag-of-interesting-unusual-optimization-applications/


ELEVEN

The Maximum p-Dispersion Problem

- $Max \sum_{i<j} d_{ij} x_i x_j$

  $s.t. \quad \sum_{j=1}^{n} x_j = k$

  $x_j \in \{0,1\}$

$Max \sum_{i<j} d_{ij} z_{ij}$

$s.t. \quad \sum_{j=1}^{n} x_j = k$

<linearization constraints>

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

$x_j \in \{0,1\}$

- Generic approach #1: RLT and aggregate (from the blog):

$x_i * (\sum_{j=1}^{n} x_j) = k * x_i$ → $x_i * (\sum_{j<i} x_j + \sum_{j>i} x_j) + x_i^2 = k * x_i$ ($x_i^2 = x_i$)

→ $\sum_{j<i} z_{ij} + \sum_{j>i} z_{ij} = (k-1) * x_i$

(add all n such constraints: $\sum_{i=1}^{n} (\sum_{j<i} z_{ij} + \sum_{j>i} z_{ij}) = (k-1) * \sum_{i=1}^{n} x_i$

→ $\sum_{j \neq i} z_{ij} = (k-1) * k$ → $2 * \sum_{i<j} z_{ij} = k * (k-1)$ → $\sum_{i<j} z_{ij} = k * (k-1)/2$