# Some Quadratic Assignment Formulations and their impact on Gurobi

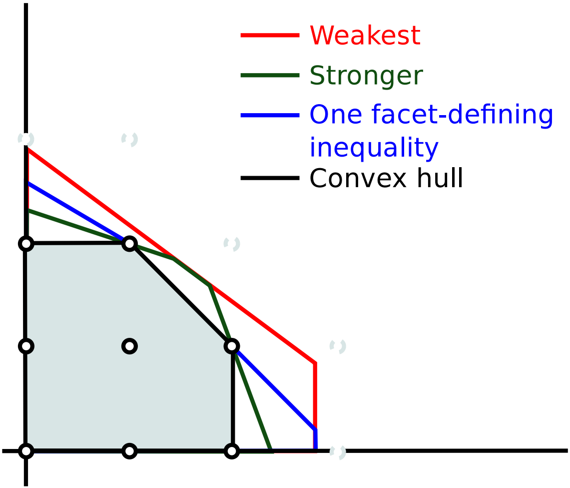
Leo Lopes[1]

[1] Monash University, School of Mathematical Sciences, Melbourne Australia

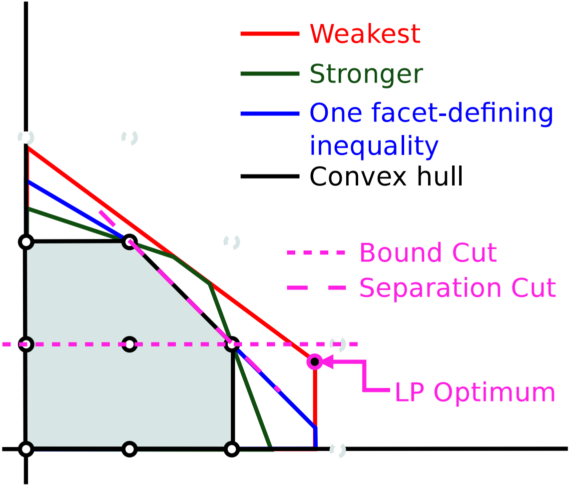
MONASH University

Monash University School of Mathematics

# The basic principle: strong and weak formulations

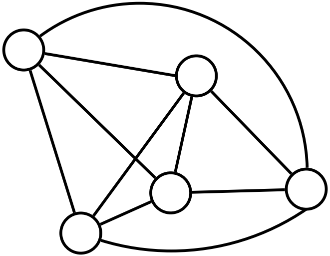


# The basic principle: formulations and branch-and-cut



# The Quadratic Assignment Problem (Input)

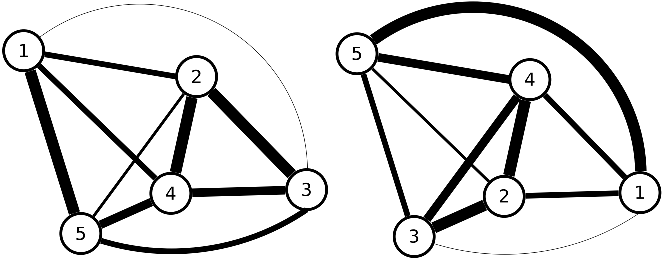
Distances:



Flows:

| | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **1** | | 2 | 0 | 2 | 4 |
| **2** | | | 4 | 4 | 1 |
| **3** | | | | 3 | 2 |
| **4** | | | | | 3 |
| **5** | | | | | |

# The Quadratic Assignment Problem (Output)



| | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **1** | | 2 | 0 | 2 | 4 |
| **2** | | | 4 | 4 | 1 |
| **3** | | | | 3 | 2 |
| **4** | | | | | 3 |
| **5** | | | | | |

# Permutation Formulation of the QAP

Let:

- $F$ be a set of facilities. Between each pair $(f_1, f_2)$ of facilities, the flow volume is $v_{f_1 f_2}$.
- $L$ be a set of locations. The distance between location $l_1$ and $l_2$ is $d_{l_1 l_2}$.

Find an assignment $g : F \rightarrow L$ of each facility to a distinct location that minimizes the total distance in the network weighted by flow:

$$\min_{g(x):F\rightarrow L} \sum_{f_i \in F} \sum_{f_2 \in F} v_{f_1 f_2} d_{g(f_1),g(f_2)}$$

# Basic Mixed Integer Programming Formulation

Let:

- $x_{fl} = \begin{cases} 1 & \text{if facility f  is assigned to location l} \\ 0 & \text{otherwise} \end{cases}$
- $y_{f_1 f_2} \geq 0$ be the distance between facilities $f_1$ and $f_2$.
- The objective: $\min_{x,y} \sum_{(f_1,f_2) \in F \times F} v_{f_1 f_2} y_{f_1 f_2}$
- Since the objective is lower when $y$ is lower, a constraint that bounds the distance between $f_1$ and $f_2$ from below to $d_{l_1 l_2}$ if and only if $f_1$ is assigned to $l_1$ and $f_2$ is assigned to $l_2$ will set $y$ to the correct value:
- $y_{f_1 f_2} \geq (x_{f_1 l_1} + x_{f_2 l_2} - 1) d_{l_1 l_2}, \forall f_1, f_2 : f_1 \neq f_2, l_1, l_2 : l_1 \neq l_2$

# In AMPL (1 of 2)

```
param n > 0;
set FAC := 1..n;
set LOC := 1..n;

param d{LOC,LOC} >= 0;
param v{FAC,FAC} >= 0;

var Assign{FAC,LOC} binary;
var Dist{FAC,FAC} >= 0;
```

# In AMPL (2 of 2)

```
minimize VolumeWeightedDistance:
  sumf1 in FAC,f2 in FAC: f2 <> f1 v[f1,f2]*Dist[f1,f2];

subject to Permute1f in FAC:
  suml in LOC Assign[f,l] = 1;

subject to Permute2l in LOC:
  sumf in FAC Assign[f,l] = 1;

subject to ComputeDistf1 in FAC, f2 in FAC,l1 in LOC, l2 in LOC:
                       f1 <> f2 and l1 <> l2:
  Dist[f1,f2] >= (Assign[f1,l1] + Assign[f2,l2] - 1)*d[l1,l2];
```

# In AMPL (2 of 2)

```
minimize VolumeWeightedDistance:
  sumf1 in FAC,f2 in FAC: f2 <> f1 v[f1,f2]*Dist[f1,f2];

subject to Permute1f in FAC:
  suml in LOC Assign[f,l] = 1;

subject to Permute2l in LOC:
  sumf in FAC Assign[f,l] = 1;

subject to ComputeDistf1 in FAC, f2 in FAC,l1 in LOC, l2 in LOC:
                       f1 <> f2 and l1 <> l2:
  Dist[f1,f2] >= (Assign[f1,l1] + Assign[f2,l2] - 1)*d[l1,l2];
```

# In AMPL (2 of 2)

```
minimize VolumeWeightedDistance:
  sumf1 in FAC,f2 in FAC: f2 <> f1 v[f1,f2]*Dist[f1,f2];

subject to Permute1f in FAC:
  suml in LOC Assign[f,l] = 1;

subject to Permute2l in LOC:
  sumf in FAC Assign[f,l] = 1;

subject to ComputeDistf1 in FAC, f2 in FAC,l1 in LOC, l2 in LOC:
                       f1 <> f2 and l1 <> l2:
  Dist[f1,f2] >= (Assign[f1,l1] + Assign[f2,l2] - 1)*d[l1,l2];
```

# Solver Results

**The math is correct, but it “doesn’t work” “in practice”!**

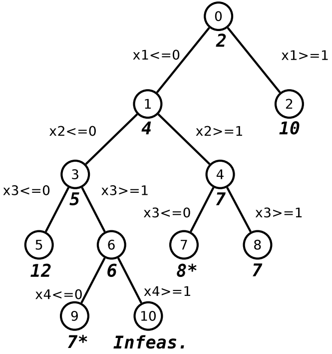
```
option solver gurobix;
option gurobi_options $gurobi_options 'outlev 1';
reset;
model qapNaive.ampl;
read n,
   {f in FAC, l in LOC} d[f,l],
   {f in FAC, l in LOC} v[f,l]
< instances/nug12.dat;
```

The branch-and-bound tree will grow until it consumes all memory available (2010).

**Producing good bounds is essential**

Let’s see the details.

# A Branch and Bound Tree



| After node | Lower Bound | Upper Bound |
|---|---|---|
| 0 | 2 | ∞ |
| 1 | 2 | ∞ |
| 2 | | |
| 3 | 4 | ∞ |
| 4 | 5 | ∞ |
| 5 | 5 | ∞ |
| 6 | | ∞ |
| **7*** | | |
| 8 | 6 | 8 |
| 9 | 6 | 7 |
| **10*** | | |

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node     |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42       582.00000  499.14837  14.2%   20.9   230s
 321368 85757  575.74359   48    46 582.00000  500.51202  14.0%   20.8   235s
 331149 86576  523.66648   43    65 582.00000  501.93019  13.8%   20.7   240s
*332324 85008              39     578.0000000  502.07250  13.1%   20.6   240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes          |   Current Node      |      Objective Bounds        |     Work
Expl Unexpl    | Obj  Depth IntInf  | Incumbent     BestBd   Gap   | It/Node Time
...
 311942 84869  cutoff      42        582.00000  499.14837  14.2%   20.9   230s
 321368 85757  575.74359   48     46 582.00000  500.51202  14.0%   20.8   235s
 331149 86576  523.66648   43     65 582.00000  501.93019  13.8%   20.7   240s
*332324 85008              39     578.0000000  502.07250  13.1%   20.6   240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node      |      Objective Bounds        |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap   | It/Node Time
...
 311942 84869  cutoff       42        582.00000  499.14837  14.2%   20.9   230s
 321368 85757  575.74359    48     46 582.00000  500.51202  14.0%   20.8   235s
 331149 86576  523.66648    43     65 582.00000  501.93019  13.8%   20.7   240s
*332324 85008               39      578.0000000  502.07250  13.1%   20.6   240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node     |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42      582.00000  499.14837  14.2%   20.9  230s
 321368 85757  575.74359   48   46 582.00000  500.51202  14.0%   20.8  235s
 331149 86576  523.66648   43   65 582.00000  501.93019  13.8%   20.7  240s
*332324 85008              39    578.0000000  502.07250  13.1%   20.6  240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node     |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42      582.00000  499.14837  14.2%   20.9  230s
 321368 85757  575.74359   48   46 582.00000  500.51202  14.0%   20.8  235s
 331149 86576  523.66648   43   65 582.00000  501.93019  13.8%   20.7  240s
*332324 85008              39    578.0000000  502.07250  13.1%   20.6  240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes          |   Current Node       |       Objective Bounds        |     Work
Expl Unexpl    | Obj   Depth IntInf   | Incumbent      BestBd    Gap   | It/Node Time
...
 311942 84869   cutoff      42          582.00000  499.14837  14.2%    20.9   230s
 321368 85757   575.74359   48      46  582.00000  500.51202  14.0%    20.8   235s
 331149 86576   523.66648   43      65  582.00000  501.93019  13.8%    20.7   240s
*332324 85008               39        578.0000000  502.07250  13.1%    20.6   240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node    |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42       582.00000  499.14837  14.2%   20.9  230s
 321368 85757  575.74359   48    46 582.00000  500.51202  14.0%   20.8  235s
 331149 86576  523.66648   43    65 582.00000  501.93019  13.8%   20.7  240s
*332324 85008              39     578.0000000  502.07250  13.1%   20.6  240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node     |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42      582.00000  499.14837  14.2%   20.9  230s
 321368 85757  575.74359   48   46 582.00000  500.51202  14.0%   20.8  235s
 331149 86576  523.66648   43   65 582.00000  501.93019  13.8%   20.7  240s
*332324 85008              39    578.0000000  502.07250  13.1%   20.6  240s
...
```

# Reading the Gurobi Log

```
Gurobi 3.0.0: outlev 1
Optimize a model with 17448 Rows, 276 Columns and 52560 NonZeros
Presolve removed 5544 rows and 42 columns
Presolve time: 0.57s
Presolved: 11904 Rows, 234 Columns, 47808 Nonzeros
Objective GCD is 1
Found heuristic solution: objective 868.0000000
Found heuristic solution: objective 724.0000000

Root relaxation: objective 0.000000e+00, 768 iterations, 0.05 seconds

Nodes        |   Current Node     |      Objective Bounds       |     Work
Expl Unexpl  | Obj  Depth IntInf  | Incumbent     BestBd   Gap  | It/Node Time
...
 311942 84869  cutoff      42      582.00000 499.14837  14.2%   20.9  230s
 321368 85757  575.74359   48    46 582.00000 500.51202  14.0%   20.8  235s
 331149 86576  523.66648   43    65 582.00000 501.93019  13.8%   20.7  240s
*332324 85008              39    578.0000000 502.07250  13.1%   20.6  240s
...
```

# Diagnosing weak bounds

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

     0     0    0.00000    0   37  724.00000    0.00000  100%     -    1s
     0     0    0.00000    0   40  724.00000    0.00000  100%     -    1s
...
     0     0    0.00000    0   31  724.00000    0.00000  100%     -    4s
H    0     0                     682.0000000    0.00000  100%     -    4s
     0     0    0.00000    0   31  682.00000    0.00000  100%     -    4s
...
H 1238   974                     618.0000000   15.97409  97.4%  53.8   32s
  1320  1020  142.41196   15  107  618.00000   27.18269  95.6%  53.1   35s
H 1369  1048                     606.0000000   27.18269  95.5%  52.7   35s
  1470  1131  377.68530   33   40  606.00000   27.75623  95.4%  52.3   41s
  1473  1135   27.75623   16   49  606.00000   27.75623  95.4%  53.6   47s
  1474  1135   27.75623   16   96  606.00000   27.75623  95.4%  53.7   50s
...
```

# Diagnosing weak bounds

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

     0     0    0.00000    0   37  724.00000    0.00000   100%     -    1s
     0     0    0.00000    0   40  724.00000    0.00000   100%     -    1s
...
     0     0    0.00000    0   31  724.00000    0.00000   100%     -    4s
H    0     0                     682.0000000    0.00000   100%     -    4s
     0     0    0.00000    0   31  682.00000    0.00000   100%     -    4s
...
H 1238   974                     618.0000000   15.97409  97.4%  53.8   32s
  1320  1020  142.41196   15  107  618.00000   27.18269  95.6%  53.1   35s
H 1369  1048                     606.0000000   27.18269  95.5%  52.7   35s
  1470  1131  377.68530   33   40  606.00000   27.75623  95.4%  52.3   41s
  1473  1135   27.75623   16   49  606.00000   27.75623  95.4%  53.6   47s
  1474  1135   27.75623   16   96  606.00000   27.75623  95.4%  53.7   50s
...
```

# Observation on variable bounds

No matter what facility gets assigned where, the distance between any two facilities must be at least the minimal distance between any two locations:

- $y_{f_1 f_2} \geq \min_{(l_1, l_2) \in L \times L : l_1 \neq l_2} d_{l_1 l_2}$
- In AMPL:

```
var Dist{FAC,FAC} >=
  min{l1 in LOC, l2 in LOC: l1 <> l2} d[l1,l2];
```

# Solver Results

At least it solves, but...

```
Explored 10,180,956 nodes (160,414,000 simplex iterations)
in 13,664.80 seconds
Thread count was 4 (of 4 available processors)
```

Let’s see the details.

# Initial bounds vs. the muddled middle

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
     0     0  348.00000    0   27  724.00000  348.00000  51.9%     -    0s
...
H 5117  3068                     622.0000000  353.58269  43.2%  28.2   61s
* 6157  3790              38     612.0000000  354.34464  42.1%  27.6   64s
  6179  3811  382.04491   29   78  612.00000  354.34464  42.1%  27.6   65s
...
 146583 88509  447.72035   31   66  590.00000  402.72441  31.7%  23.6  600s
 147934 89260  549.35810   41   65  590.00000  402.91520  31.7%  23.6  605s
 149758 90263  516.43027   44   67  590.00000  403.18118  31.7%  23.6  610s
...
 1660356 685638  563.64870   59   55  586.00000  469.62771  19.9%  19.7 3000s
 1664829 687024     cutoff   48       586.00000  469.71047  19.8%  19.7 3005s
...
```

# Initial bounds vs. the muddled middle

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
     0     0  348.00000    0   27  724.00000  348.00000  51.9%     -    0s
...
H 5117  3068                     622.0000000  353.58269  43.2%  28.2   61s
* 6157  3790              38     612.0000000  354.34464  42.1%  27.6   64s
  6179  3811  382.04491   29   78  612.00000  354.34464  42.1%  27.6   65s
...
 146583 88509  447.72035   31   66  590.00000  402.72441  31.7%  23.6  600s
 147934 89260  549.35810   41   65  590.00000  402.91520  31.7%  23.6  605s
 149758 90263  516.43027   44   67  590.00000  403.18118  31.7%  23.6  610s
...
 1660356 685638  563.64870   59   55  586.00000  469.62771  19.9%  19.7 3000s
 1664829 687024     cutoff   48       586.00000  469.71047  19.8%  19.7 3005s
...
```

# Initial bounds vs. the muddled middle

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
     0     0  348.00000    0   27  724.00000  348.00000  51.9%     -    0s
...
H 5117  3068                    622.0000000  353.58269  43.2%  28.2   61s
* 6157  3790              38    622.0000000  354.34464  42.1%  27.6   64s
  6179  3811  382.04491   29   78  612.00000  354.34464  42.1%  27.6   65s
...
 146583 88509  447.72035   31   66  590.00000  402.72441  31.7%  23.6  600s
 147934 89260  549.35810   41   65  590.00000  402.91520  31.7%  23.6  605s
 149758 90263  516.43027   44   67  590.00000  403.18118  31.7%  23.6  610s
...
 1660356 685638  563.64870   59   55  586.00000  469.62771  19.9%  19.7 3000s
 1664829 687024    cutoff   48       586.00000  469.71047  19.8%  19.7 3005s
...
```

# Initial bounds vs. the muddled middle

```
...
   Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
     0     0  348.00000    0   27  724.00000  348.00000  51.9%     -    0s
...
H 5117  3068                      622.0000000  353.58269  43.2%  28.2   61s
* 6157  3790               38     612.0000000  354.34464  42.1%  27.6   64s
  6179  3811  382.04491   29   78  612.00000  354.34464  42.1%  27.6   65s
...
 146583 88509  447.72035   31   66  590.00000  402.72441  31.7%  23.6  600s
 147934 89260  549.35810   41   65  590.00000  402.91520  31.7%  23.6  605s
 149758 90263  516.43027   44   67  590.00000  403.18118  31.7%  23.6  610s
...
 1660356 685638  563.64870   59   55  586.00000  469.62771  19.9%  19.7 3000s
 1664829 687024     cutoff   48       586.00000  469.71047  19.8%  19.7 3005s
...
```

# Initial bounds vs. the muddled middle

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
     0     0  348.00000    0   27  724.00000  348.00000  51.9%     -    0s
...
H 5117  3068                    622.0000000  353.58269  43.2%  28.2   61s
* 6157  3790              38    612.0000000  354.34464  42.1%  27.6   64s
  6179  3811  382.04491   29   78  612.00000  354.34464  42.1%  27.6   65s
...
 146583 88509  447.72035   31   66  590.00000  402.72441  31.7%  23.6  600s
 147934 89260  549.35810   41   65  590.00000  402.91520  31.7%  23.6  605s
 149758 90263  516.43027   44   67  590.00000  403.18118  31.7%  23.6  610s
...
 1660356 685638  563.64870   59   55  586.00000  469.62771  19.9%  19.7 3000s
 1664829 687024     cutoff   48       586.00000  469.71047  19.8%  19.7 3005s
...
```

# Rule of thumb: use more variables in each constraint

If we use more variables in the constraints, when some of them get fixed, the resulting LP bounds tend to be tigher.

**Observation:**

If we assume that $f_1$ was assigned to $l_1$, then for each $f_2$: the distance between $f_1$ and $f_2$ must be at least the sum of the distances out of $l_1$ minus the sum of all the other distances except that between $l_1$ and $l_2$ when $f_2$ is assigned to $l_2$:

- $y_{f_1 f_2} \geq \sum_{l_2 \neq l_1} d_{l_1 l_2} x_{f_1 l_1} - \sum_{f_2 \neq f_1} (\sum_{l_3 \notin \{l_1, l_2\}} d_{l_1 l_3}) x_{f_2 l_2}, \forall f_1, f_2, l_1 : f_1 \neq f_2$
- In AMPL:

```
subject to ComputeDist2{f1 in FAC, f2 in FAC,l1 in LOC: f1 <> f2}:
  Dist[f1,f2] >= Assign[f1,l1]*sum{l2 in LOC: l2 <> l1} d[l1,l2]
    - sum{l2 in LOC: l2 <> l1}(Assign[f2,l2] *
        sum{l3 in LOC: l3 <> l1 and l3 <> l2} d[l1,l3]
      ) ;
```

# Solver Results

Much better...

```
Explored 2,855,769 nodes (19,376,754 simplex iterations)
in 800.96 seconds
Thread count was 4 (of 4 available processors)
```

Much better. Let's see the details.

# A healthy tree

```
...
   Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 148565 58745  472.38928   42   41   578.00000  426.17068  26.3%  11.0   60s
 162715 63069  550.49347   51   41   578.00000  428.39964  25.9%  10.9   65s
 176814 67294  493.66237   43   41   578.00000  430.60198  25.5%  10.8   70s
...
 2352964 205490     cutoff   32       578.00000  525.24585  9.13%   7.3  600s
 2374875 201750     cutoff   47       578.00000  526.14261  8.97%   7.3  605s
 2398390 197345  536.17206   41   43  578.00000  527.17244  8.79%   7.3  610s
...
```

# A healthy tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 148565 58745  472.38928   42    41   578.00000  426.17068  26.3%  11.0    60s
 162715 63069  550.49347   51    41   578.00000  428.39964  25.9%  10.9    65s
 176814 67294  493.66237   43    41   578.00000  430.60198  25.5%  10.8    70s
...
 2352964 205490     cutoff   32        578.00000  525.24585  9.13%   7.3   600s
 2374875 201750     cutoff   47        578.00000  526.14261  8.97%   7.3   605s
 2398390 197345  536.17206   41   43  578.00000  527.17244  8.79%   7.3   610s
...
```

# A healthy tree

```
...
   Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 148565 58745  472.38928   42   41  578.00000  426.17068  26.3%  11.0   60s
 162715 63069  550.49347   51   41  578.00000  428.39964  25.9%  10.9   65s
 176814 67294  493.66237   43   41  578.00000  430.60198  25.5%  10.8   70s
...
 2352964 205490     cutoff   32       578.00000  525.24585  9.13%   7.3  600s
 2374875 201750     cutoff   47       578.00000  526.14261  8.97%   7.3  605s
 2398390 197345  536.17206   41   43  578.00000  527.17244  8.79%   7.3  610s
...
```

# A healthy tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 148565 58745  472.38928   42   41  578.00000  426.17068  26.3%  11.0   60s
 162715 63069  550.49347   51   41  578.00000  428.39964  25.9%  10.9   65s
 176814 67294  493.66237   43   41  578.00000  430.60198  25.5%  10.8   70s
...
 2352964 205490     cutoff   32       578.00000  525.24585  9.13%   7.3  600s
 2374875 201750     cutoff   47       578.00000  526.14261  8.97%   7.3  605s
 2398390 197345  536.17206   41   43  578.00000  527.17244  8.79%   7.3  610s
...
```

# A healthy tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 148565 58745  472.38928   42   41  578.00000  426.17068  26.3%  11.0   60s
 162715 63069  550.49347   51   41  578.00000  428.39964  25.9%  10.9   65s
 176814 67294  493.66237   43   41  578.00000  430.60198  25.5%  10.8   70s
...
 2352964 205490     cutoff   32      578.00000  525.24585  9.13%   7.3  600s
 2374875 201750     cutoff   47      578.00000  526.14261  8.97%   7.3  605s
 2398390 197345  536.17206   41   43  578.00000  527.17244  8.79%   7.3  610s
...
```

# Smaller constants tend to be better

- The sums over all distances are likely making many of the constraints non-binding in the LP relaxations. A formulation with smaller numbers will *usually* perform better.
- Since $\sum_l x_{fl} = 1$, then either $x_{fl} = 1$ or $\sum_{l_2 \neq l} x_{fl_2} = 1$. In the former case, $y_{f_1 f_2}$ will be $d_{ll_2}$. In the latter case, then discounting $\max_{l_2 \neq l} d_{ll_2}$ from $y_{f_1 f_2}$ is just enough to make the constraint non-binding.
- In algebra:
  $y_{f_1 f_2} \geq \sum_{l_2 \neq l_1} d_{l_1 l_2} x_{f_2 l_2} - (\max_{l_2 \neq l_1} d_{l_1 l_2}) \sum_{l_2 \neq l_1} x_{f_1 l_2}, \forall f_1, f_2, l_1 : f_1 \neq f_2$
- Rearranging and reorganizing terms: $M_l = \max_{l_2 \neq l} d_{ll_2}$ and
  $y_{f_1 f_2} \geq \sum_{l_2 \neq l} (d_{ll_2} x_{f_2 l_2} - M_l x_{f_1 l_2}), \forall f_1, f_2, l : f_1 \neq f_2$
- In AMPL:

```
param maxd{l in LOC} := max{l2 in LOC: l2 <> l} d[l,l2];
subject to ComputeDist3{f1 in FAC, f2 in FAC,l in LOC: f1 <> f2}:
  Dist[f1,f2] >= sum{l2 in LOC: l2 <> l} (
    d[l,l2]*Assign[f2,l2] - maxd[l]* Assign[f1,l2]
  );
```

# Solver Results

Total time cut in half again...

```
Explored 723,627 nodes (12,293,156 simplex iterations)
in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

Much better again. Let's see the details.

# A healthier tree

```
...
   Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35       578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29       578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39       578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50       578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30       578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# A healthier tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35        578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29        578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39        578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50        578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30        578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# A healthier tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35       578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29       578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39       578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50       578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30       578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# A healthier tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35         578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29         578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39         578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50         578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30         578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# A healthier tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35       578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29       578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39       578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50       578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30       578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# A healthier tree

```
...
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
...
 41971 20620  578.17341   42   57  586.00000  426.84741  27.2%  26.7   60s
 49310 23712  506.64911   34   68  586.00000  430.28747  26.6%  26.2   65s
 56491 26513  580.49821   55   64  586.00000  433.86992  26.0%  25.9   70s
...
 443823 85925     cutoff   35        578.00000  517.55057  10.5%  19.6  300s
 454929 85247  572.52381   38   46  578.00000  518.97396  10.2%  19.5  305s
 461098 84930     cutoff   29        578.00000  519.80133  10.1%  19.4  310s
...
 676266 34174     cutoff   39        578.00000  552.77727  4.36%  17.6  410s
 687701 27147     cutoff   50        578.00000  556.03894  3.80%  17.5  415s
 699749 18846     cutoff   30        578.00000  560.31293  3.06%  17.4  420s
 713063  8885  568.11389   37   59  578.00000  566.75240  1.95%  17.2  425s

Explored 723627 nodes (12293156 simplex iterations) in 428.08 seconds
Thread count was 4 (of 4 available processors)
```

# Modeling Tools and solvers are powerful – but not automatic *yet*

- I am not endorsing these formulations as practical ways for solving QAPs
    - Even stronger bounds computed using more sophisticated machinery are needed for larger instances
- Modeling Languages allow experimentation with new formulations at very low cost
- The general principle is to define constraints as tightly as possible.
    - Dramatic improvements can be obtained with very little extra programming effort.
    - Your real world “branch” problem likely contains a “core” problem (sequencing, coloring, QAP). Check the academic literature for strong formulations of those problems.
    - Analytical results showing that one constraint is dominated by another are common and appreciated in the academic literature.
    - This formulation can still be improved. How?

## Thanks for listening!

Parting words: put some thought into the problem and experiment with different formulations. It produces results (and is fun to do!)