# What's New in the Gurobi Infeasibility Finder

## Gurobi 9.1 Product Launch

GUROBI OPTIMIZATION

The World's Fastest Solver

Ed Klotz, Ed Rothberg, Yuriy Zinchenko, Ph.Ds

December 8-9, 2020

# Outline

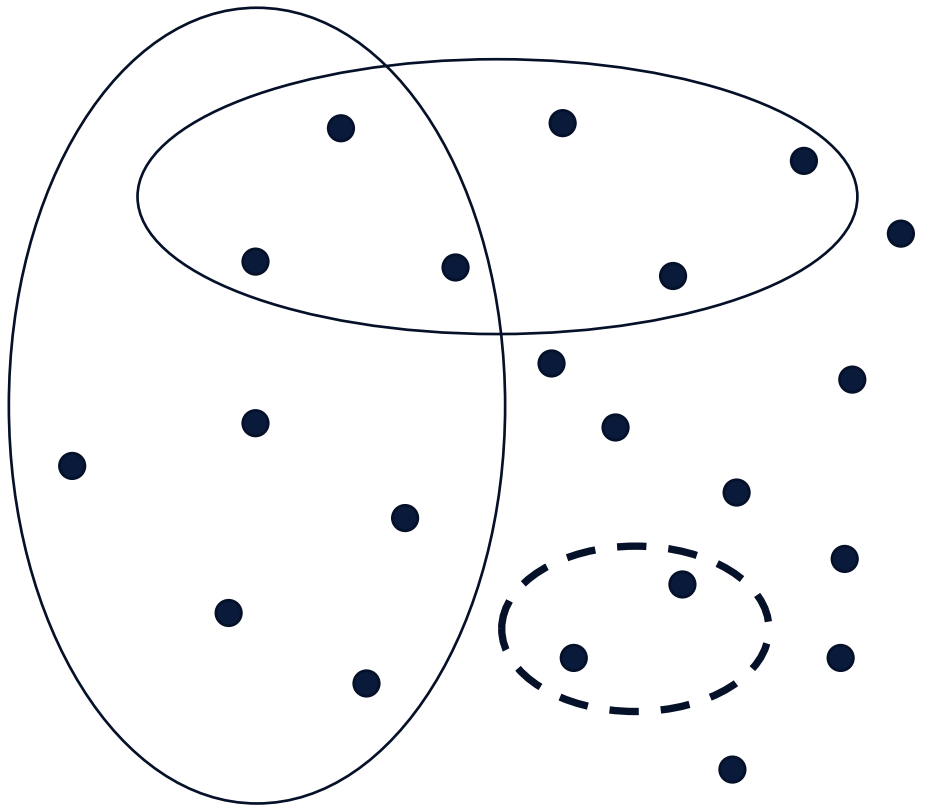
- Problem Statement
- IIS Fundamentals
  - Deletion
  - Certificates
  - Addition
- Performance
- Examples

# Problem Statement

- **Given an infeasible system of constraints…**
  - Find a single Irreducible Inconsistent Subsystem (IIS)
    - Ax=b is infeasible
    - Removing any constraint renders the result feasible
  - IIS is *minimal*, not *minimum*
- **Meant to guide a human to the source of the infeasibility**
  - The smaller, the better
- **Cost**
  - Cheap for LP, very expensive for MIP

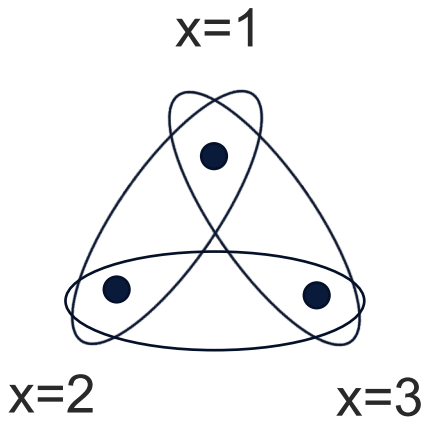
# Depicting an IIS

- **Graphical representation**
  - One node = one constraint
  - One oval = one IIS
  - All 3 are minimal, but only the dashed one is minimum



# Example

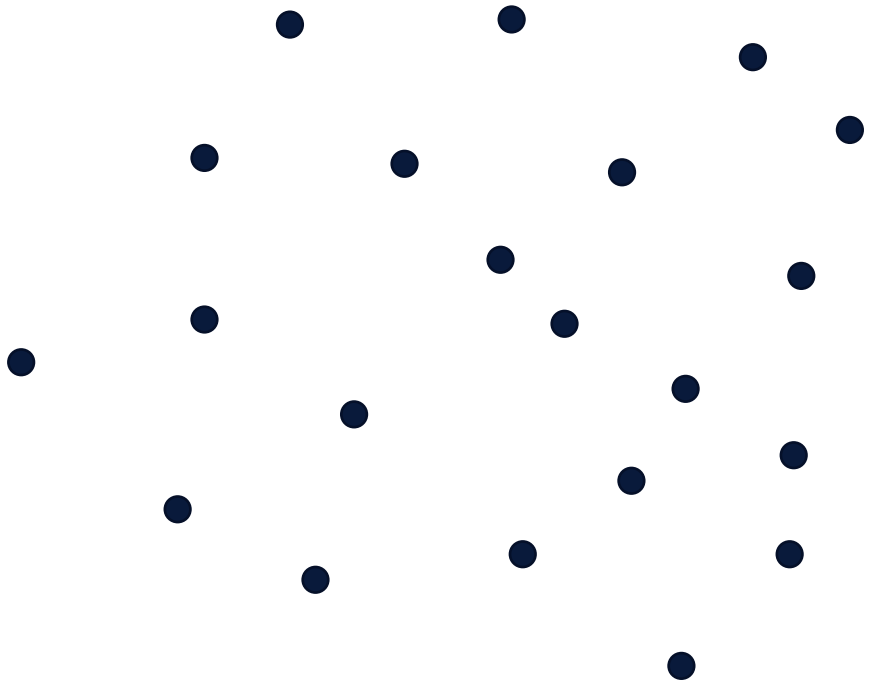
- **Model will typically have multiple IISs**
- **A trivial example:**




  - But it will come back to haunt us later…

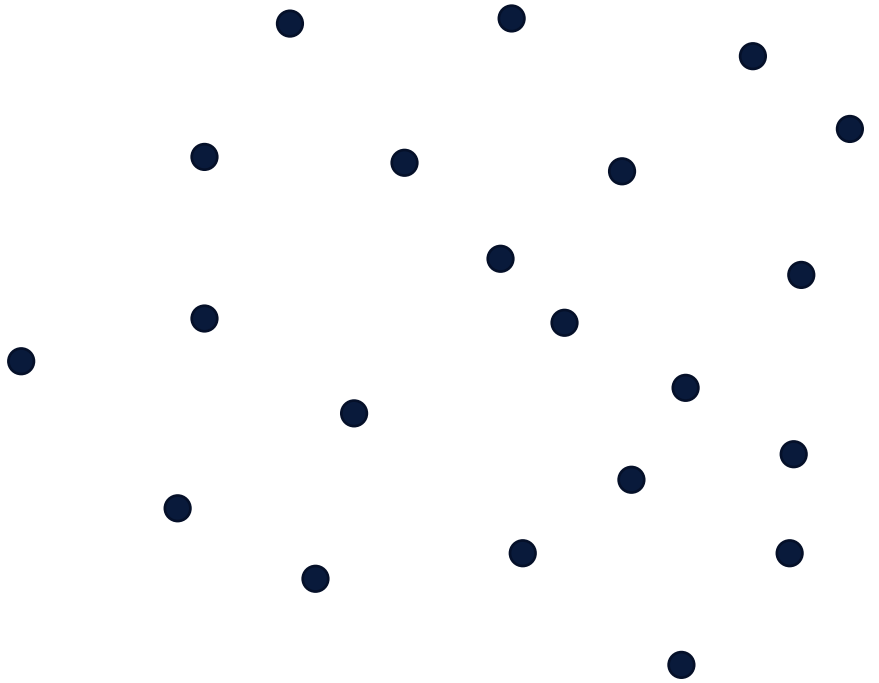
# IIS Computation

- **Maintain**
  - C: infeasible constraint set, IIS candidate
    - Build down
  - K: known IIS members ($K \subseteq C$)
    - Build up
  - Stop when K = C



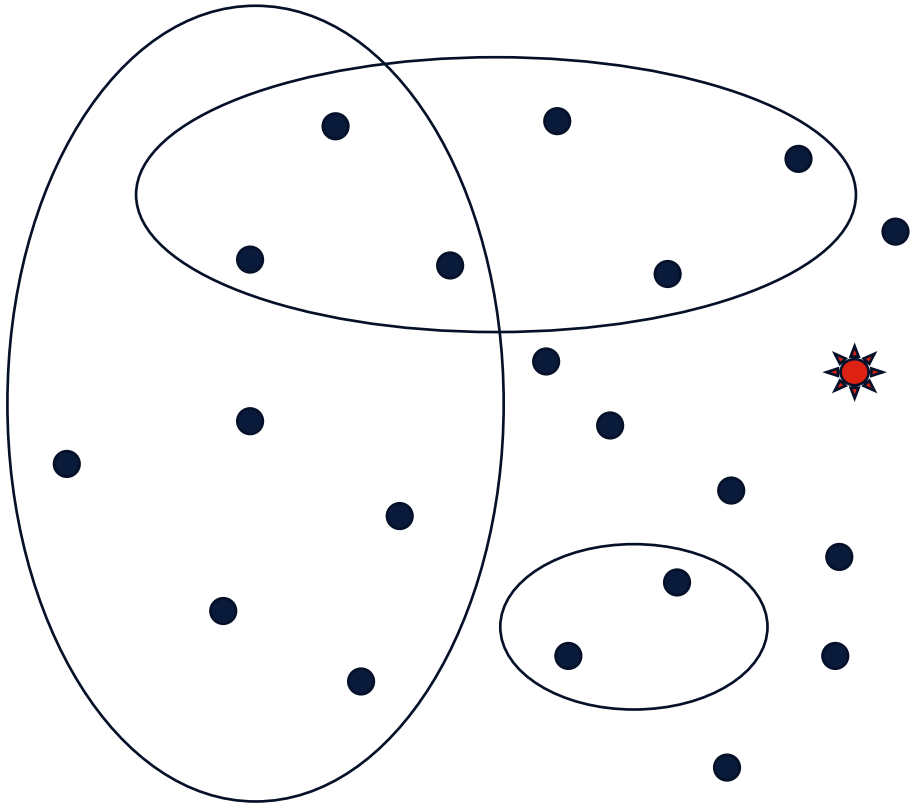
# IIS Primitives

- **IIS primitives (Chinneck, 1991)**
  - Single-constraint deletion
  - Multi-constraint deletion
  - Independent (parallel) deletion
  - Addition
- **Computing IISs easier for LP since we can also use duality**



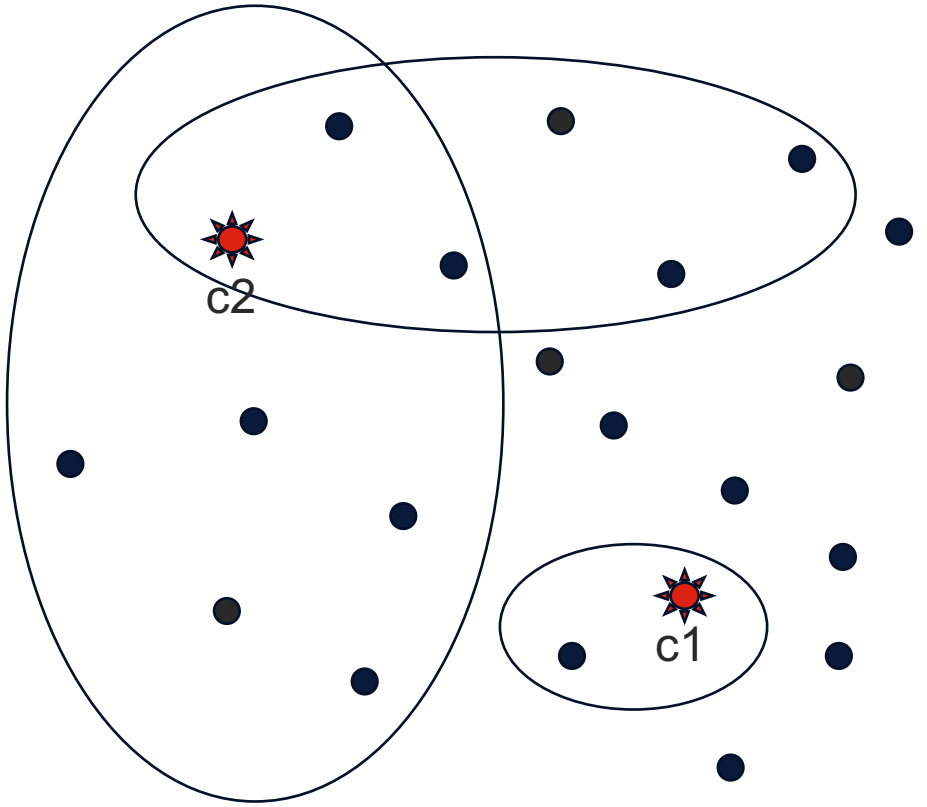
# Single-Constraint Deletion

- **Choose a constraint *c***
  - Perform (truncated) MIP solve on C\{c}
- **Three possible outcomes:**
  - Still infeasible
    - Constraint can be removed from candidate set (C = C\{c})
  - Feasible
    - Constraint belongs to every IIS in *C* ($\mathrm{K} = \mathrm{K} \cup \{c\}$)
      - No such constraint exists for first iteration in example on this slide
  - MIP didn't finish
    - Probably infeasible (?)
    - No useful conclusion



# Feasible Outcome Requires a Cover

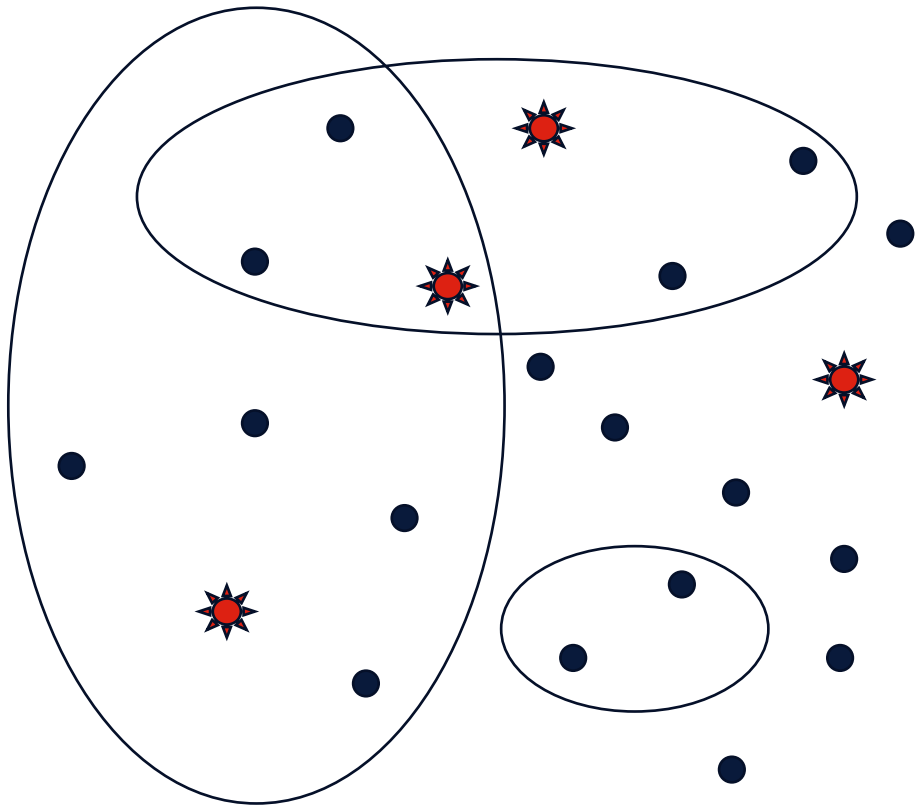
- **To get a feasible outcome, removed constraint(s) must form a *cover* on all IISs in the model**
  - Computing one IIS is cheaper than computing an IIS cover
  - After correcting the infeasibility in the computed IIS, additional IISs may remain
    - Correct the computed IIS, then compute another one
  - Correcting infeasibility doesn't always mean removal of constraint(s)
    - Could be relaxing constraint, adding new activities, or other changes




Remove c1 first → c2 ∈ $K$ (and vice versa)

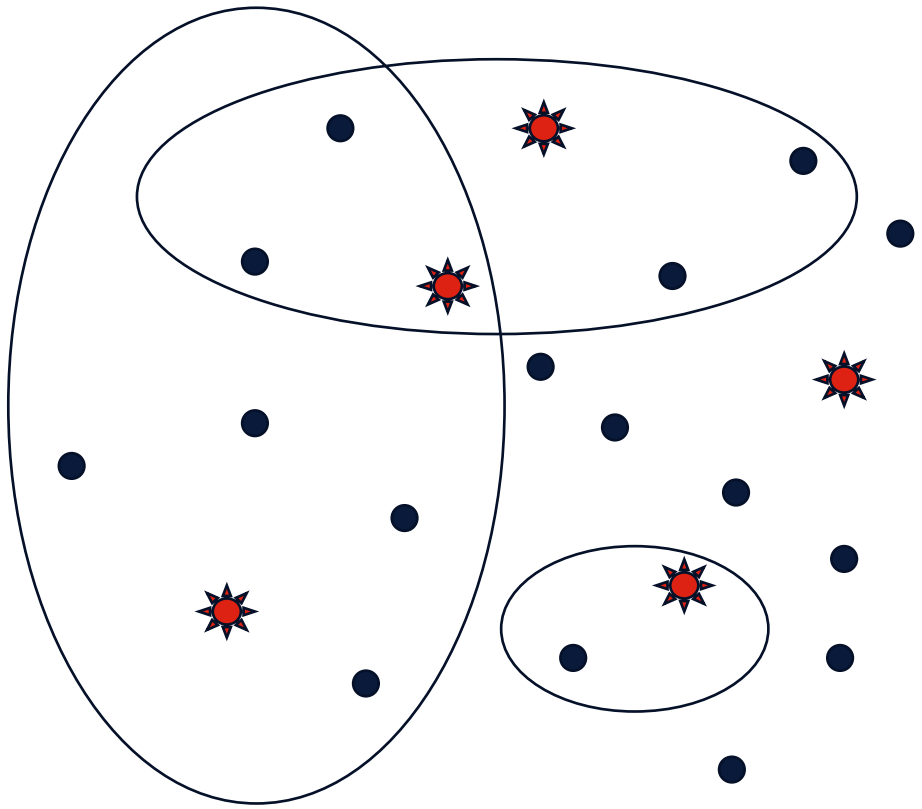
# Multi-Constraint Deletion

- **Remove multiple constraints *S***
- **Two possible outcomes:**
  - Still infeasible
    - Constraints can be removed from candidate set (C = C \ S)
  - Feasible or incomplete
    - No useful conclusion
    - Unlike single constraint deletion, cannot augment the set of known members K



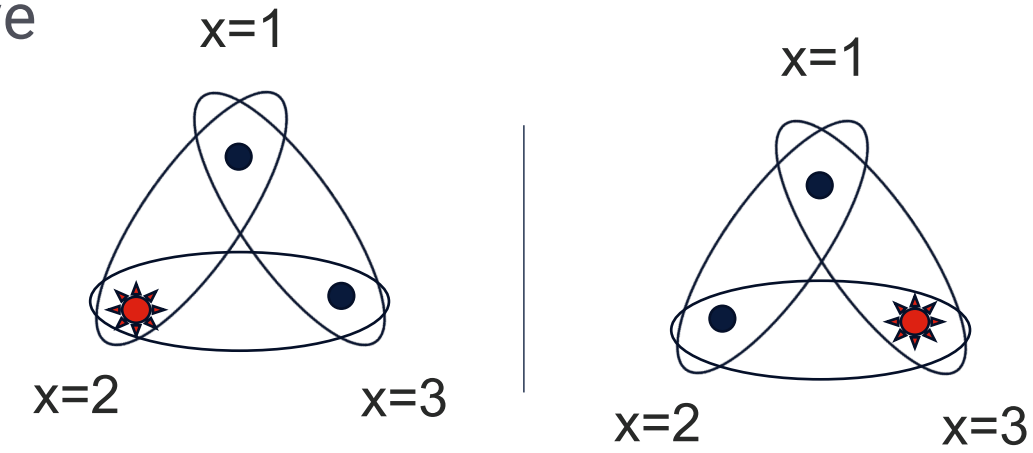
# Multi-Constraint Deletion

- **Remove multiple constraints *S***
- **Two possible outcomes:**
  - Still infeasible
    - Constraints can be removed from candidate set (C = C \ S)
  - Feasible or incomplete
    - No useful conclusion
    - Unlike single constraint deletion, cannot augment the set of known members K



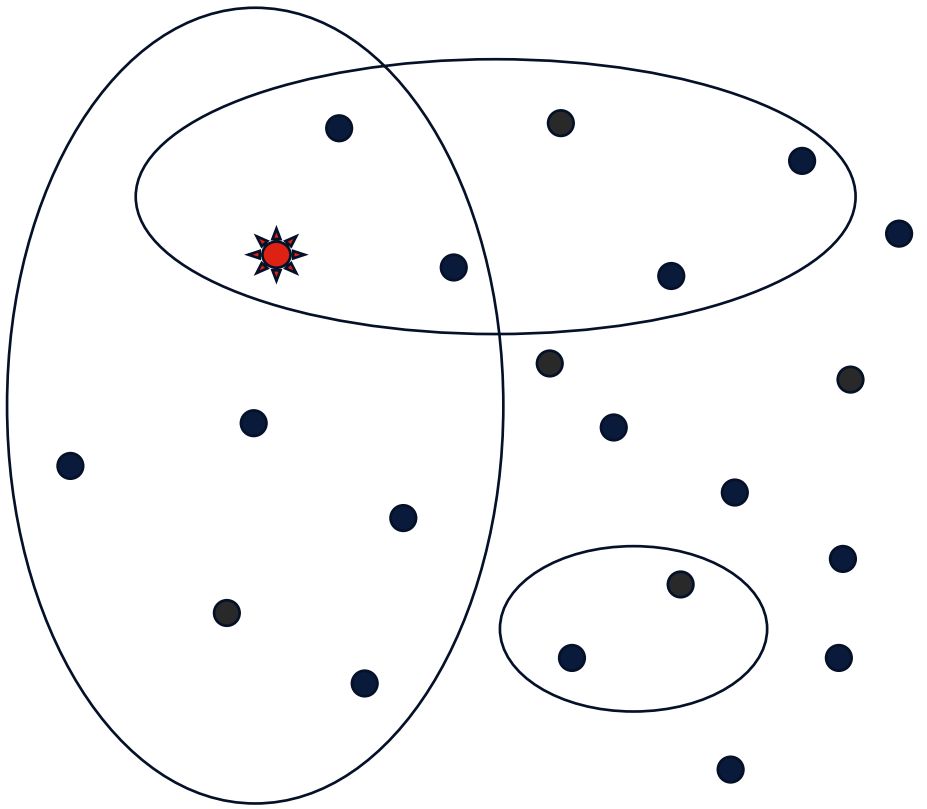
# Simultaneous, Independent Deletion

- **Parallel deletion of constraints**
  - Removing x=2 and x=3 individually preserves infeasibility
  - But can't remove both and preserve infeasibility
  - Can only remove one of the k concurrently removed constraints



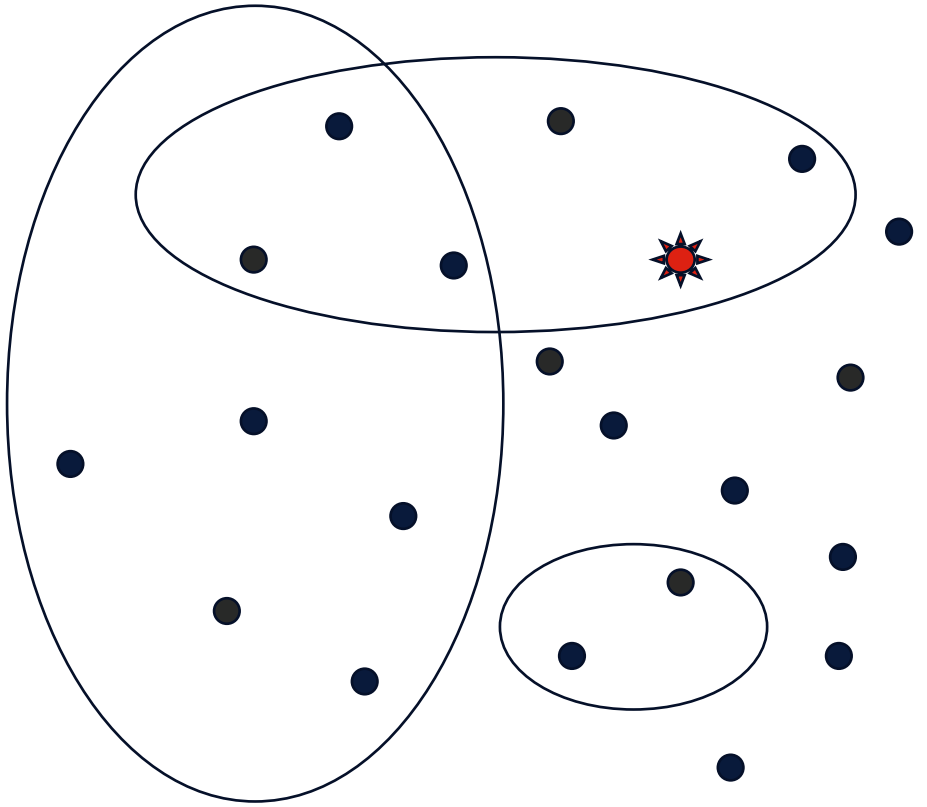
# More Observations – Pruning IISs

- Removing a constraint from the candidate set removes all IISs that contain that constraint



# Some Good News

- If you choose constraints to remove at random, smaller IISs are more likely to survive



# Relentless Focus on Performance

GUROBI OPTIMIZATION

- 9.1 versus 9.0, mean time to proven IIS:

| Full set | Count | Loss/Win | TimeR |
|---|---|---|---|
| all: | 222 | 46/ 111 | 0.501 |
| >0s: | 214 | 46/ 111 | 0.485 |
| >1s: | 156 | 41/ 107 | 0.381 |
| >10s: | 112 | 20/ 86 | 0.271 |
| >100s: | 76 | 11/ 62 | 0.175 |
| >1000s: | 53 | 7/ 45 | 0.117 |

- Nearly 2X improvement overall
- 5.7X improvement on the harder models (> 100s)

# Remaining Challenges

GUROBI OPTIMIZATION

- **IIS computation still can be slow**
- **No good way to add multiple constraints to the known IIS member set at once**
  - Requires MIP solve per element of set K (known IIS members)
  - Big IIS = Slow IIS
- **No good way to exploit parallelism when IIS is small**
- **Limited ability to exploit presolve**
  - Working on (almost) the whole original user model

# Some Ideas Currently on the Table

GUROBI OPTIMIZATION

- Filter on integrality "constraints”
  - Can work very well on some models
  - Not yet clear how to decide when to use
- If using FeasRelax to add new IIS members, consider examining IIS for the corresponding (infeasible) fixed LP as well
- Alternate between addition (FeasRelax) and deletion
  - Deletion to shrink candidate set
  - Addition to grow known IIS set

# Examples: Explaining the Infeasibility

- **Gurobi's infeasibility finder can be used for more than diagnosing infeasible problems**
- **It can explain any aspect of a model that can be phrased as a question about a related, infeasible model**
  - How can I reduce some large big-M values in my model?
  - Which constraints or bounds in the model (i.e. limits in the associated physical system being modelled) prevent improvement in the optimal objective value?
  - Many others

# Example: Reducing Large Big-M values

- **cdma, an open MIPLIB 2017 model**
  - Gurobi has found the best known solution, but MIPgap remains significant
  - Anything that can tighten the formulation could help
  - Wide spread of coefficients, even after presolve
    - Less important to reduce large values in original model if presolve already does so
  - Big-M style constraints (presolved model)

```
id17768: 3e+06 id16 + id36 <= 3e+06      // id16 binary
id17770: - 3e+06 id1 + id36 <= 0         // id1   binary
…
id36 <= 3e+06
```

  - Reduction in bound on id36 reduces coefficients in id17768, id17770

# Example : Reducing Large Big-M values

GUROBI OPTIMIZATION

- **Create an infeasible model whose IIS will explain how to reduce the big-M values**

```
\ id36 <= 3e+06
 id36 >= 100000
```

  - This model is infeasible
  - Immediately deduce id36 <= 100000
  - Can do better by looking at the IIS

# Example : Reducing Large Big-M values

GUROBI OPTIMIZATION

- IIS has 15 fairly dense conservation of flow constraints:

```
id17758: - id4759 - id4939 - id5299 - id5479 - id5839 - id6019 - id6379
   - id6559 - id6739 - id6919 - id7099 + id7111 + id7123 + id7147 + id7159
   + id7183 + id7195 + id7219 + id7231 + id7243 + id7255 + id7267 - id7639
...
   - id5850 - id6030 - id6390 - id6570 - id6750 - id6930 - id7110 + id7122
   + id7134 + id7158 + id7170 + id7194 + id7206 + id7230 + id7242 + id7254
   + id7266 + id7278 - id7650 - id7830 - id8190 + id50 = 0
```

- And 5 fairly dense supply or demand constraints that force flow

```
id17929: id7855 + id7879 + id7891 + id7903 + id7915 + id7927 + id7951
 + id7856 + id7880 + id7892 + id7904 + id7916 + id7928 + id7952 + id7857
...
 + id8070 + id8082 + id8094 + id8106 + id8118 + id8154 + id8166 + id8178
 + id8190 = 1000
```

# Example: Reducing Large Big-M values

GUROBI OPTIMIZATION

- Don’t try to interpret the individual constraints in the IIS when combinations may simplify the analysis
- Try summing up the 20 constraints

```
def combineconstraints(model, mcons, mmults=None, filepath = None,
showit=True):
…
if mults == None:                # sum up the constraints
        for c in mcons:

            baselhs.add(model.getRow(c))
            baserhs += c.rhs
            if c.sense != firstsense:
                showsense = False
```

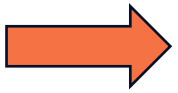
# Example: Reducing Large Big-M values

- Result of summing up the 20 constraints

```
id36 + id37 + id38 + id39 + id40 + id41 + id42 + id43 + id44 + id45 + id46 + id47 +
id48 + id49 + id50    ==    5000.0
```

- Deduce an upper bound of 5000 on id36
- Reduce coefficients on constraints that depend on upper bound
- id37,…, id50 have the same bound and constraint structure

```
id17768: 3e+06 id16 + id36 <= 3e+06
id17770: - 3e+06 id1 + id36 <= 0
…
id36 <= 3e+06
```



```
id17768: 5000 id16 + id36 <= 5000
id17770: - 5000 id1 + id36 <= 0
…
id36 <= 5000
```

# Examples

GUROBI OPTIMIZATION

- **Which constraints or bounds in the model (i.e. limits in the associated physical system being modelled) prevent improvement in the optimal objective value?**
- **Lotsize, a solved MIPLIB model on which Gurobi doesn't fare particularly well**
  - 8.97 hours to prove optimality despite having found the optimal solution in a half hour
  - Extract info about the model by creating an infeasible model by adding a constraint on the objective

# Examples

- Which constraints or bounds in the model (i.e. limits in the associated physical system being modelled) prevent improvement in the optimal objective value?
- Start of nodelog for lotsize:

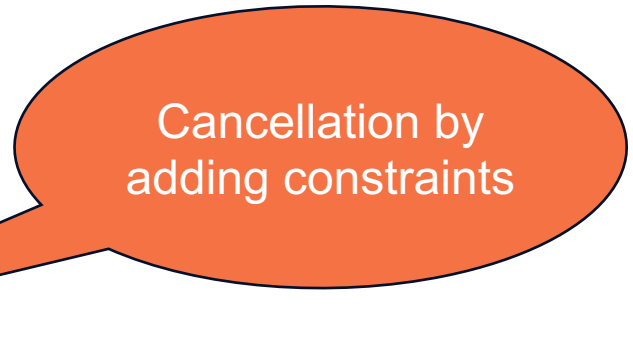
```
     Nodes    |    Current Node    |     Objective Bounds      |     Work
Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 348385.347    0  471          - 348385.347      -     -    0s
     0     0 534372.642    0  598          - 534372.642      -     -    0s
     0     0 603722.884    0  689          - 603722.884      -     -    0s
```

- No integer feasible solution exists with these dual bound values
  - Constrain the objective to be <= 400000

# Examples

- Resulting IIS




```
R0001: C0001 + C0601 - C1196 = 163
R0002: C0002 - C0601 + C0602 + C1196 - C1197 = 144
R0003: C0003 - C0602 + C0603 + C1197 - C1198 = 126
R0004: C0004 - C0603 + C0604 + C1198 - C1199 = 196
…
R0599: C0599 - C1194 + C1195 + C1789 - C1790 = 79
R0600: C0600 - C1195 + C1790 = 70

R0601: C0001 - 18298 C1791 <= 0  // C1791,…,C2390 fixed charge binaries
R0602: C0002 - 18298 C1792 <= 0
…
R1191: C0591 - 20019 C2381 <= 0
R1194: C0594 - 20019 C2384 <= 0
R1200: C0600 - 20019 C2390 <= 0
```

# Examples

- **Interpret the IIS by looking at groups of constraints**
  - Add the first 600 constraints in the IIS:

```
AGG:    C0001 + C0002  + … + C0600 = 93503 // must pay the cost ($1-10) for this flow

R0601: C0001 - 18298 C1791 <= 0           // C1791,…,C2390 fixed charge binaries
R0602: C0002 - 18298 C1792 <= 0
…
R1200: C0600 - 20019 C2390 <= 0           // Must pay some fixed charges ($5k – 40k)
```

  - Explains why we cannot have a cost below 400000
  - Added side benefit of a MIR style cut

```
93503 = C0001 + C0002  + … + C0600 ≤  18298 C1791 + 18298 C1792 + … +  20019 C2390
→ C1791 + C1792 + … + C2390 >= 5
```

# Examples

- **Improved performance**
  - MIR style cut from previous slide:  3.1 hours
  - Refinement of this cut:  1.2 hours
  - Based on the flow style of the constraints in the IIS and the cut we derived, just run original model with aggressive flow path, flow cover and MIR cuts
    - Time drops to 25 minutes
- **IISs based on overconstrained objectives can facilitate our understanding of the essential parts of the model, and thus help us tighten the formulation**
- **IISs can be large**
  - May need to interpret groups of constraints rather than individual ones

# Examples

- **Other questions we can answer by computing an IIS on the appropriate model**
  - Why does Gurobi reject my MIP start?
  - What does it mean when Gurobi support says my model is on the boundary of feasibility and infeasibility?
  - Is a particular constraint or group of constraints in my model redundant?
  - Many more
- **Pose the question in the context of an infeasible model**

# Summary

GUROBI OPTIMIZATION

- **Computing IISs for MILPs significantly more challenging than LPs**
  - Need to solve a series of subMIPs
  - Parallelization has some challenges
- **Computing IISs is more than just for infeasible models**
  - Gurobi's Infeasibility Finder can explain many aspects of the model
- **Therefore, we take improving IIS computation time as seriously as we do with other algorithms**
  - Improvements in version 9.1 confirm this
- **IISs can be large in size**
  - Consider groups of constraints rather than individual ones

# References

1. Chinneck, J.W., Feasibility and Infeasibility in Optimization, Springer.

# Thank You – Questions?

GUROBI
OPTIMIZATION

The World's Fastest Solver