Using MIP to Model Midstream Energy Assets

Levi DeLissa

GUROBI OPTIMIZATION

The World's Fastest Solver

EAST DALEY CAPITAL

Welcome to the Webinar

Using MIP to Model Midstream Energy Assets

GUROBI OPTIMIZATION

The World's Fastest Solver

EAST DALEY CAPITAL

Speaker Introduction

GUROBI OPTIMIZATION

EAST DALEY CAPITAL

Levi DeLissa

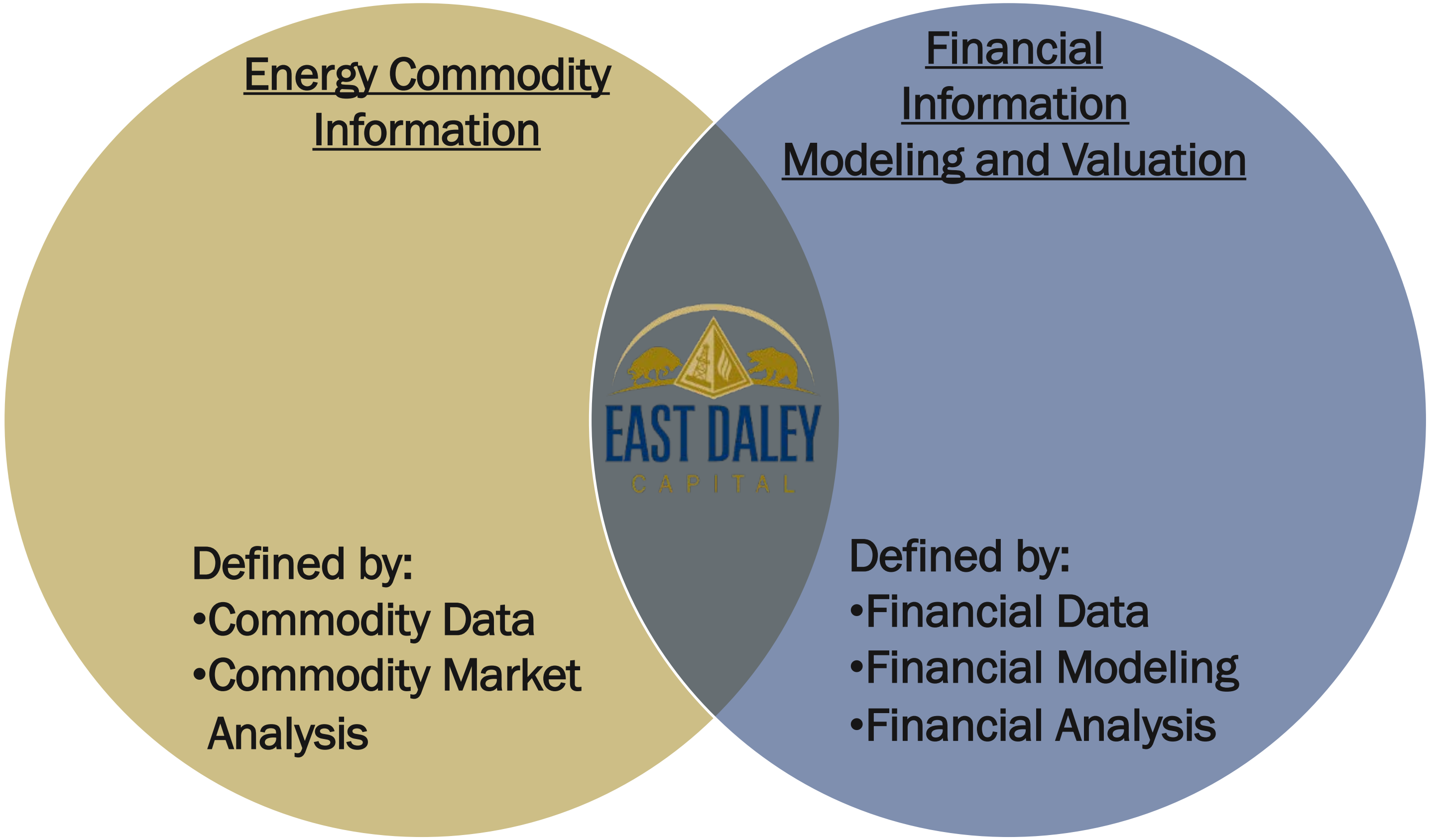
- Data Scientist at East Daley Capital
- MS in Industrial Engineering and BS in Industrial Engineering from Kansas State University
- Spent 5 years in Colorado working with East Daley Capital, helping people use data to quantify risk in the midstream energy sector
- Passionate about solving challenging problems

Using MIP to Model Midstream Energy Assets

Levi DeLissa

Who is East Daley?

Reshaping the way markets use data to quantify risk for North American energy infrastructure, midstream, and E&P companies



Energy Commodity Information

Defined by:
- Commodity Data
- Commodity Market Analysis

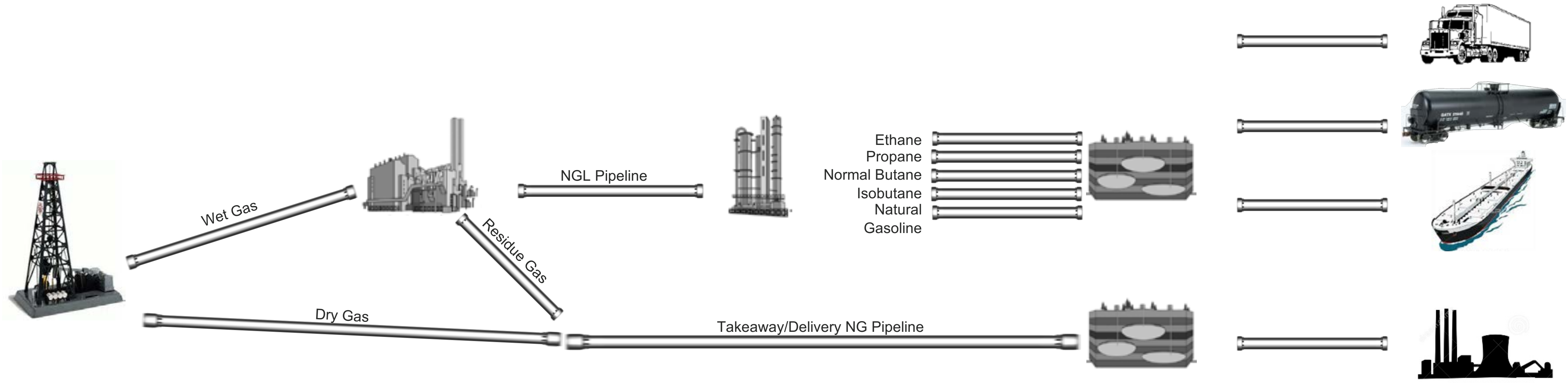
Financial Information Modeling and Valuation

Defined by:
- Financial Data
- Financial Modeling
- Financial Analysis

Midstream 101

*Simplified

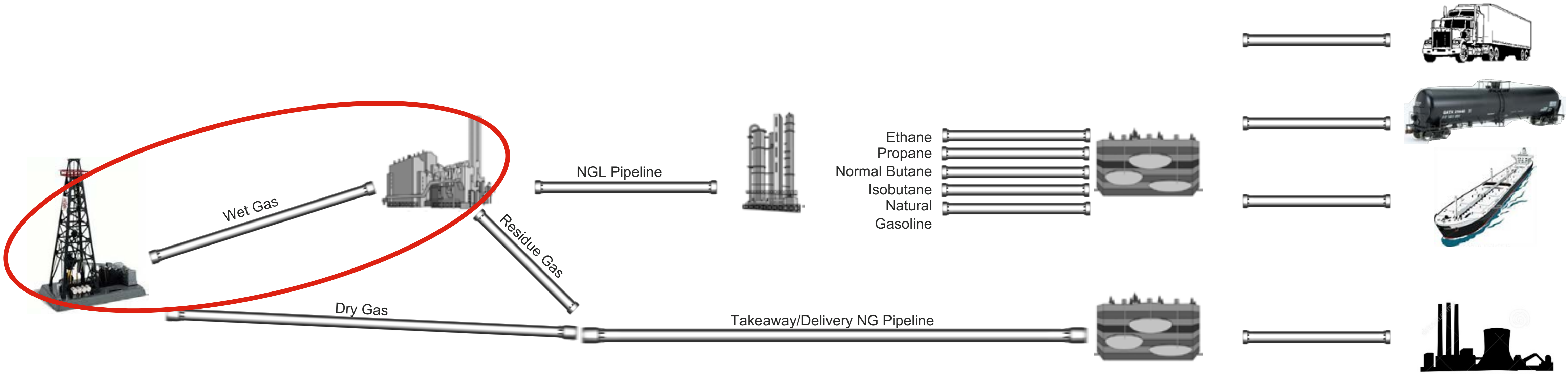
**Oil Not Included



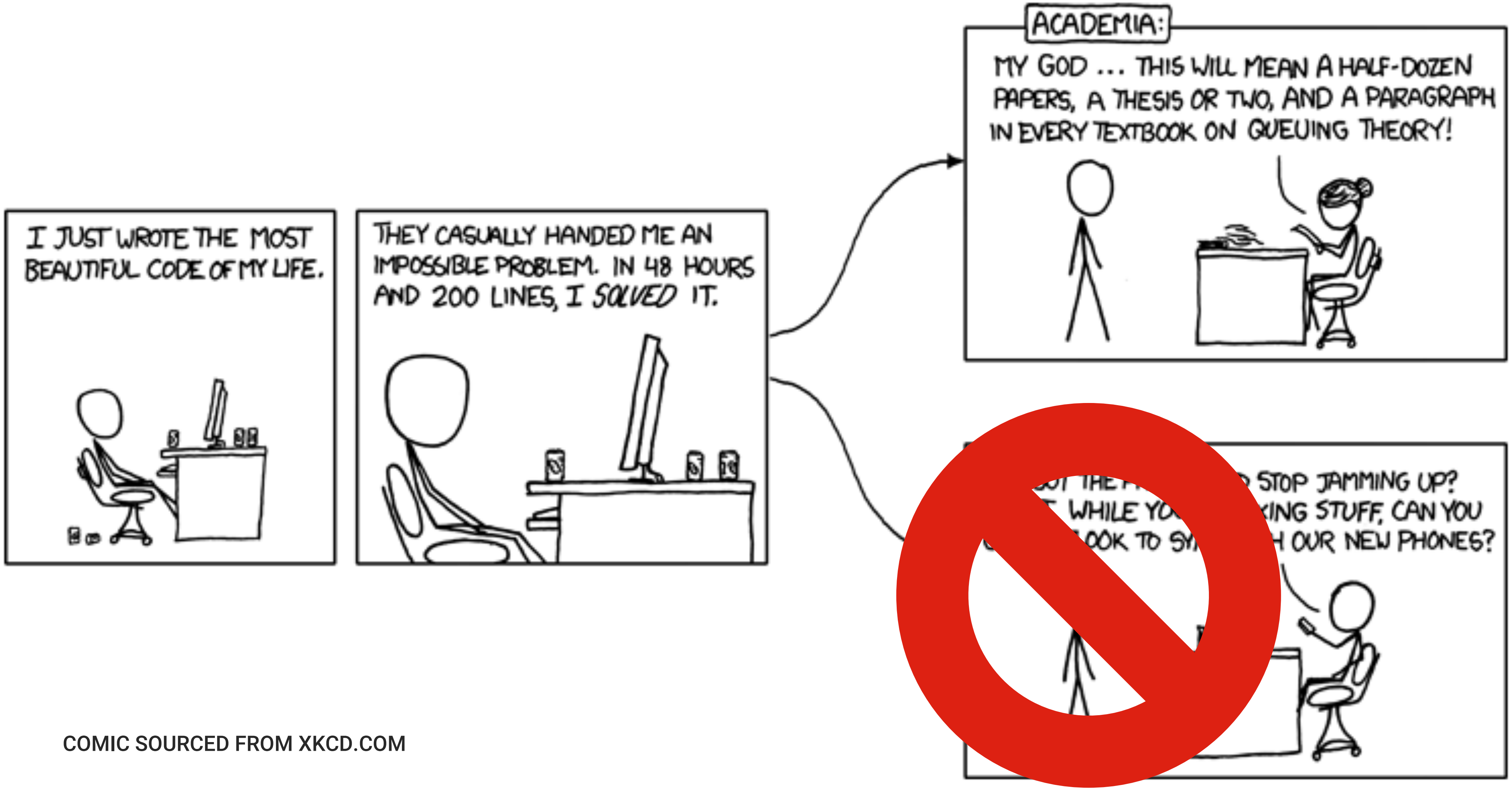
Exploration and Production	Gathering and Processing	Fractionation	Transportation and Storage	Consumption and Distribution

Some questions

- If the price of oil goes to $30 what happens?
- If there is a hurricane in South Texas what happens?
- If Chesapeake Energy is at risk of bankruptcy what happens?
- If a drilling rig drills a new well at 31.7° N, -101.9° W what happens?



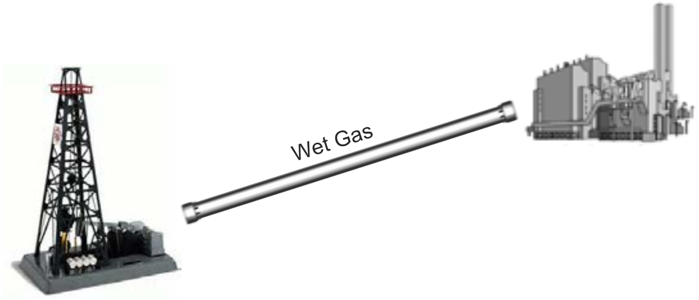
Wait… I though this was about optimization?

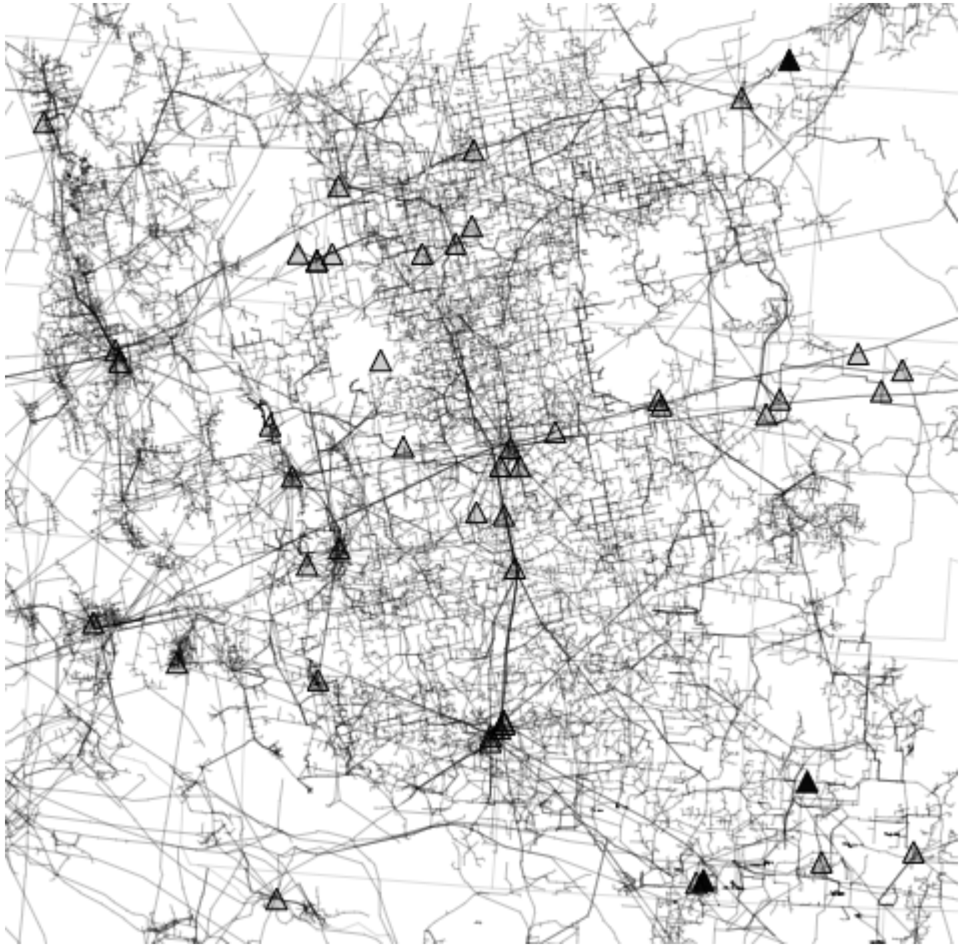

I JUST WROTE THE MOST BEAUTIFUL CODE OF MY LIFE.
THEY CASUALLY HANDED ME AN IMPOSSIBLE PROBLEM. IN 48 HOURS AND 200 LINES, I SOLVED IT.
ACADEMIA:
MY GOD … THIS WILL MEAN A HALF-DOZEN PAPERS, A THESIS OR TWO, AND A PARAGRAPH IN EVERY TEXTBOOK ON QUEUING THEORY!
STOP JAMMING UP?
WHILE YO XING STUFF, CAN YOU
OOK TO SY H OUR NEW PHONES?

COMIC SOURCED FROM XKCD.COM

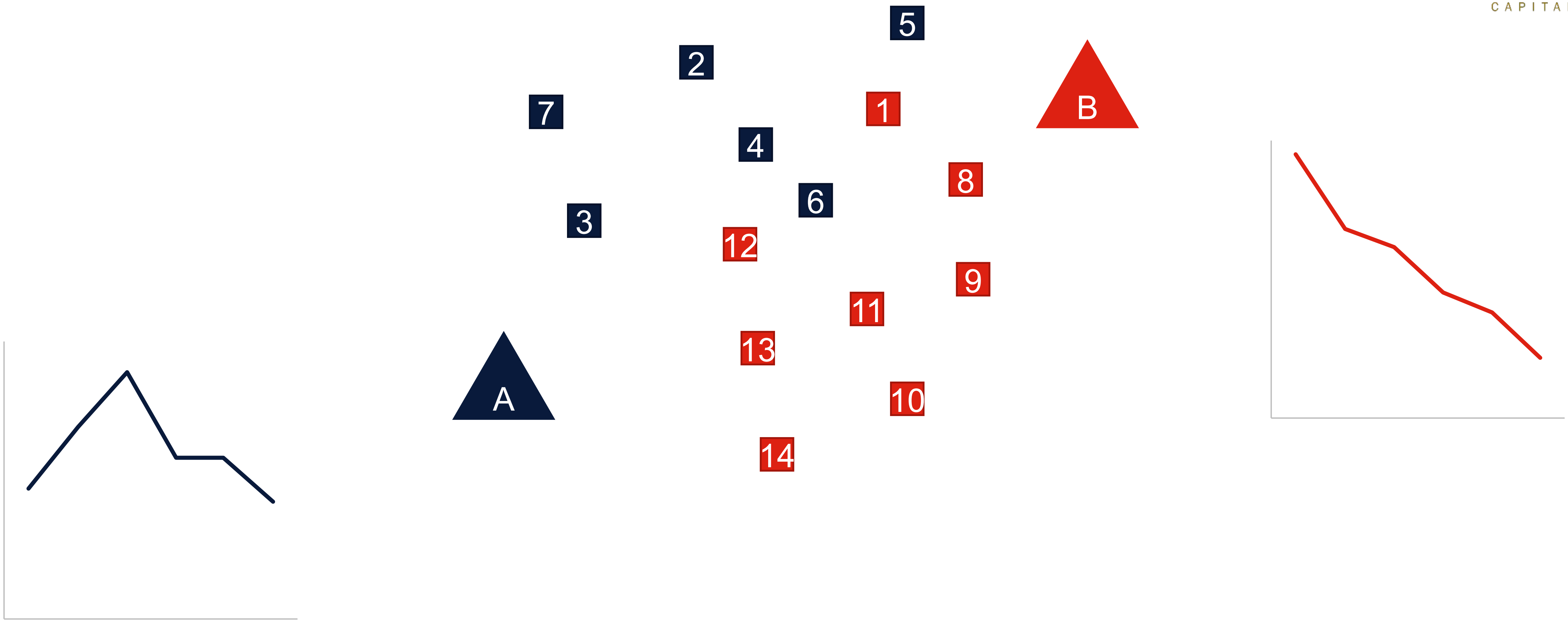
Gas Gathering & Processing

- Complex Pipeline Networks
- Hundreds of Processing Plants
  - Grouped into Systems
- Millions of Wells

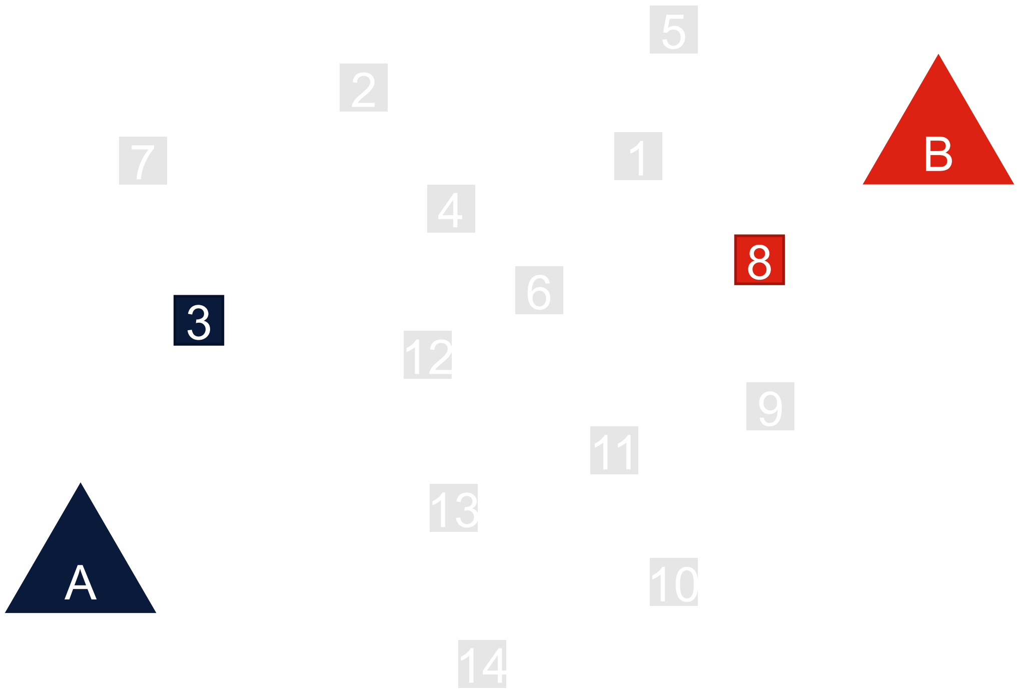


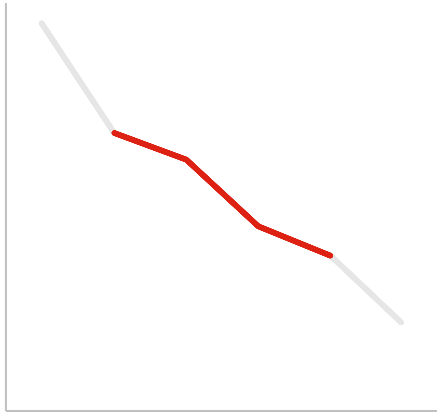


In a perfect world



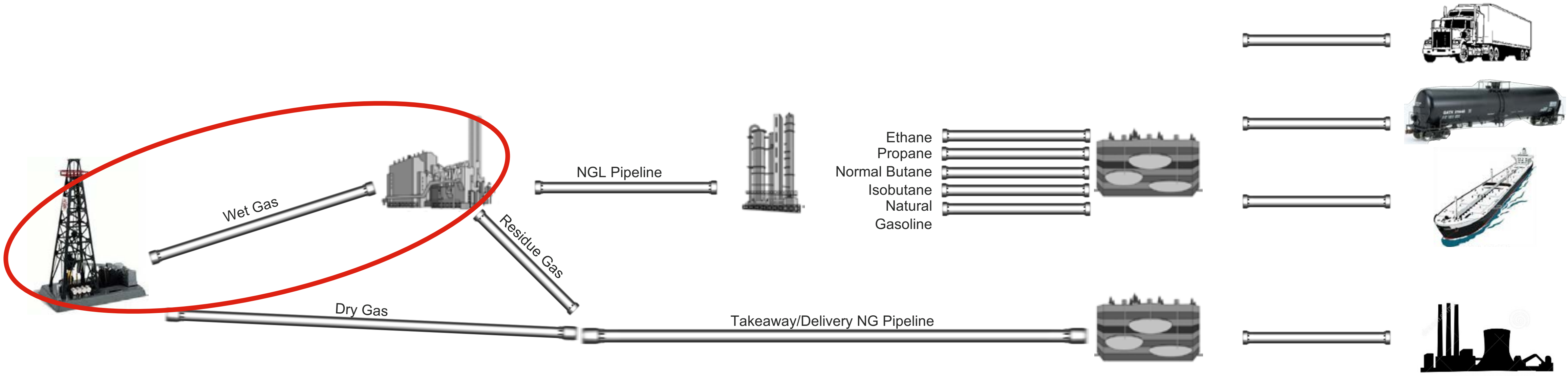
In reality





The goal

- Answer detailed questions about the midstream supply chain
- Systematically model which wells are connected to each processing system



The model

- Determine which wells provide input volumes to which G&P systems.

- Align with any known facts.
  - Volumes
  - Geography

- Given multiple choices, choose the ‘best’ one

Decision variables

- **Determine which wells provide input volumes to which G&P systems.**
  - Assume 100% volume allocation
  - Assume constant within the time period in question

- **Set of wells** W

- **Set of systems** S

- **Let** $x_{ij} \in \{0,1\}$ **be** 1 **if well** $\mathrm{i} \in W$ **is allocated to system** $j \in S$
  - $|W| * \ |S|$ binary variables
  - Logical constraint $\sum_{j \in S} x_{ij} \leq 1 \ \forall\, i \in W$

Constraints

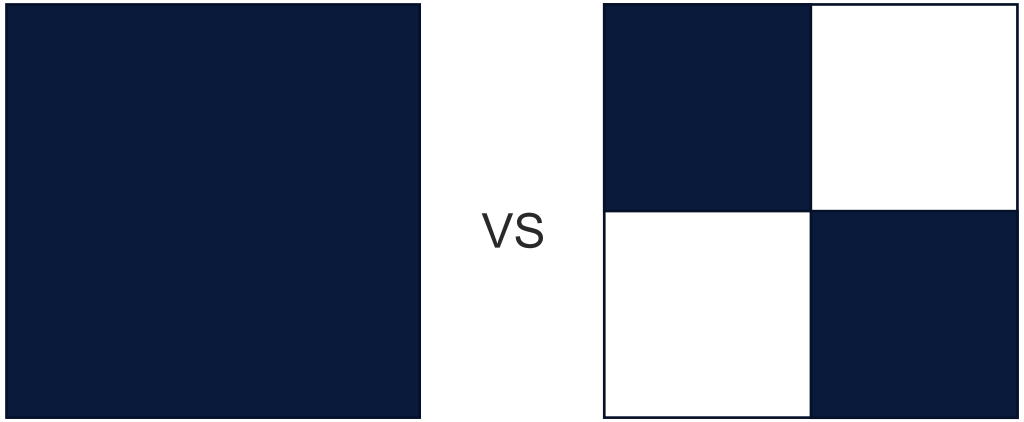
- Align with any known facts.
  - Volumes
  - Geography

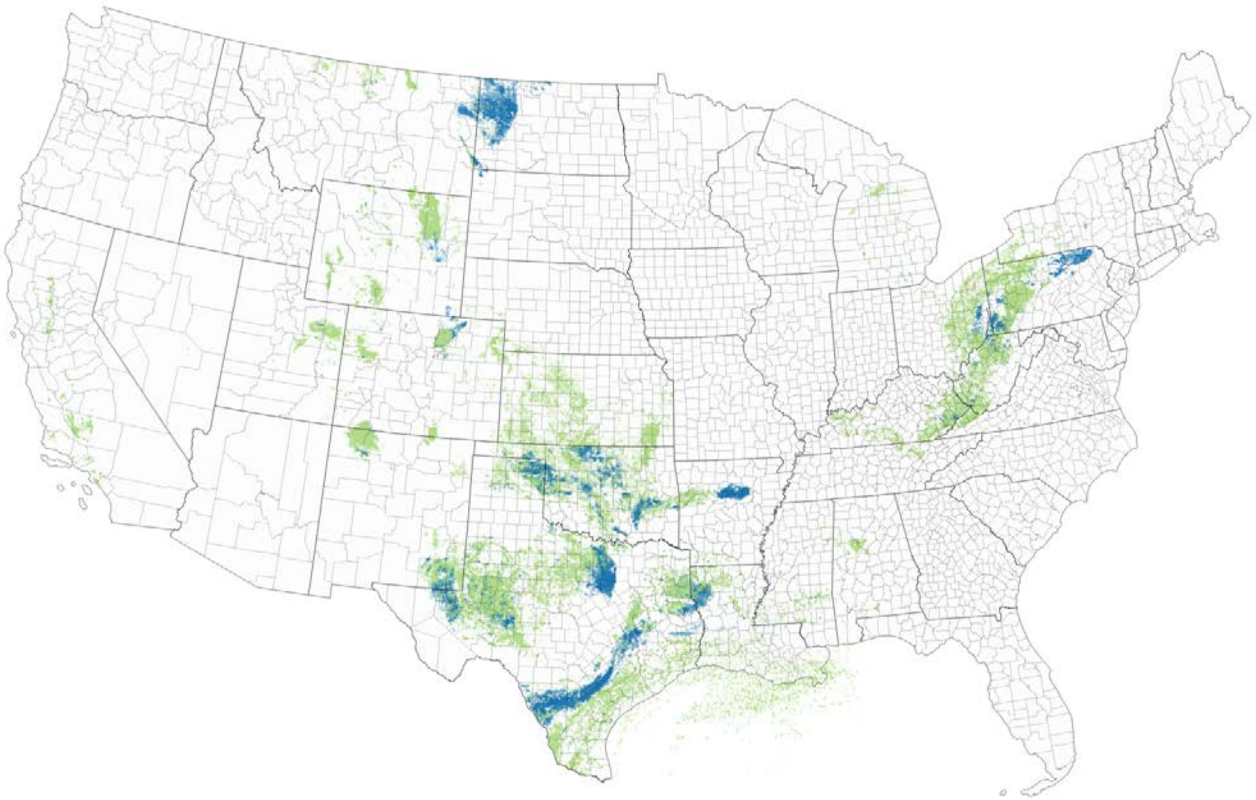
- Set of time periods T
- Set of well volumes WV
- Set of system volumes SV

- $\sum_{i \in W} x_{ij} * WV_{it} = SV_{jt} \forall\, j \in S\; t \in T$
  - $|S| * |T|$ constraints

Constraints cont.

- Align with any known facts.
  - Volumes
  - Geography





Objective

- Given multiple choices, choose the 'best' one

- Set of well to system costs C
  - Another series of models to calculate costs

- Minimize $\sum_{i \in Wells\ j \in Systems} x_{ij} * C_{ij}$

- Model is complete, but not likely to be feasible (or useful)

Revisiting infeasible constraints

- $\sum_{i \in W} x_{ij} * WV_{it} \enclose{circle}{=} SV_{jt} \forall\, j \in S\ t \in T$

- **Reformulate as a ‘soft’ constraint(s)**
  - Add variable $f \in \mathbb{R}^{+}$ which represents units of constraint violation for system j in time period t
  - $\sum_{i \in W} x_{ij} * WV_{it} \; - f_{ijt} \leq SV_{jt} \forall\, j \in S\ t \in T$
  - $\sum_{i \in W} x_{ij} * WV_{it} + f_{ijt} \geq SV_{jt} \forall\, j \in S\ t \in T$
  - Add $\sum_{i \in W\ j \in S\ t \in T} f_{ijt}$ to objective function to minimize violation

Linear constraints

- **Linear constraints can exactly model many things**
  - If then
  - Either or
  - Max and Min

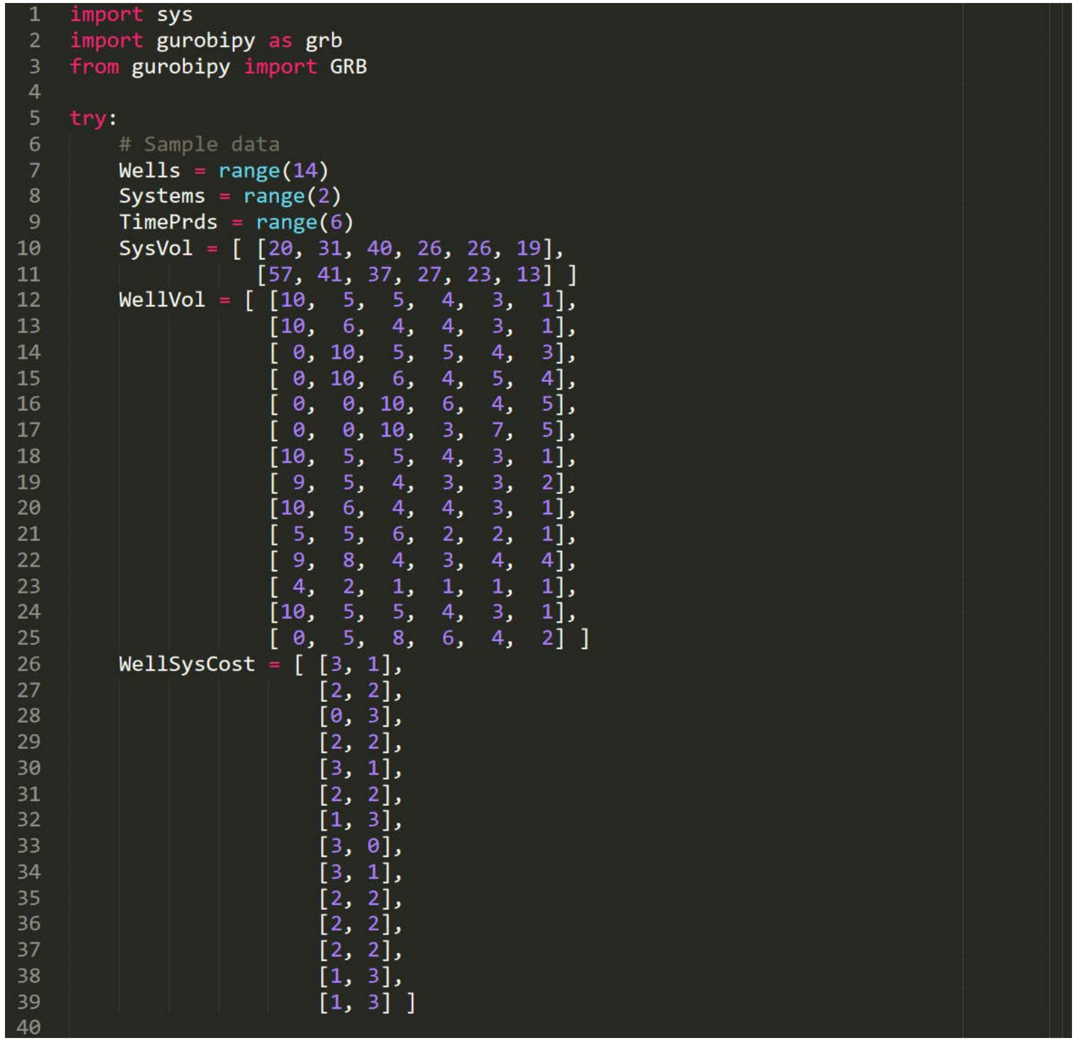
- **Linear constraints can approximate many things**
  - Piecewise functions

- **Still encounter non-linear**

Multiple objectives

- When working with ‘soft’ constraints importance can be an issue
- Two approaches
  - Weighted
  - Hierarchy
- You can do the work yourself, or you can let Gurobi take care of this under the hood

Implementation - Setup

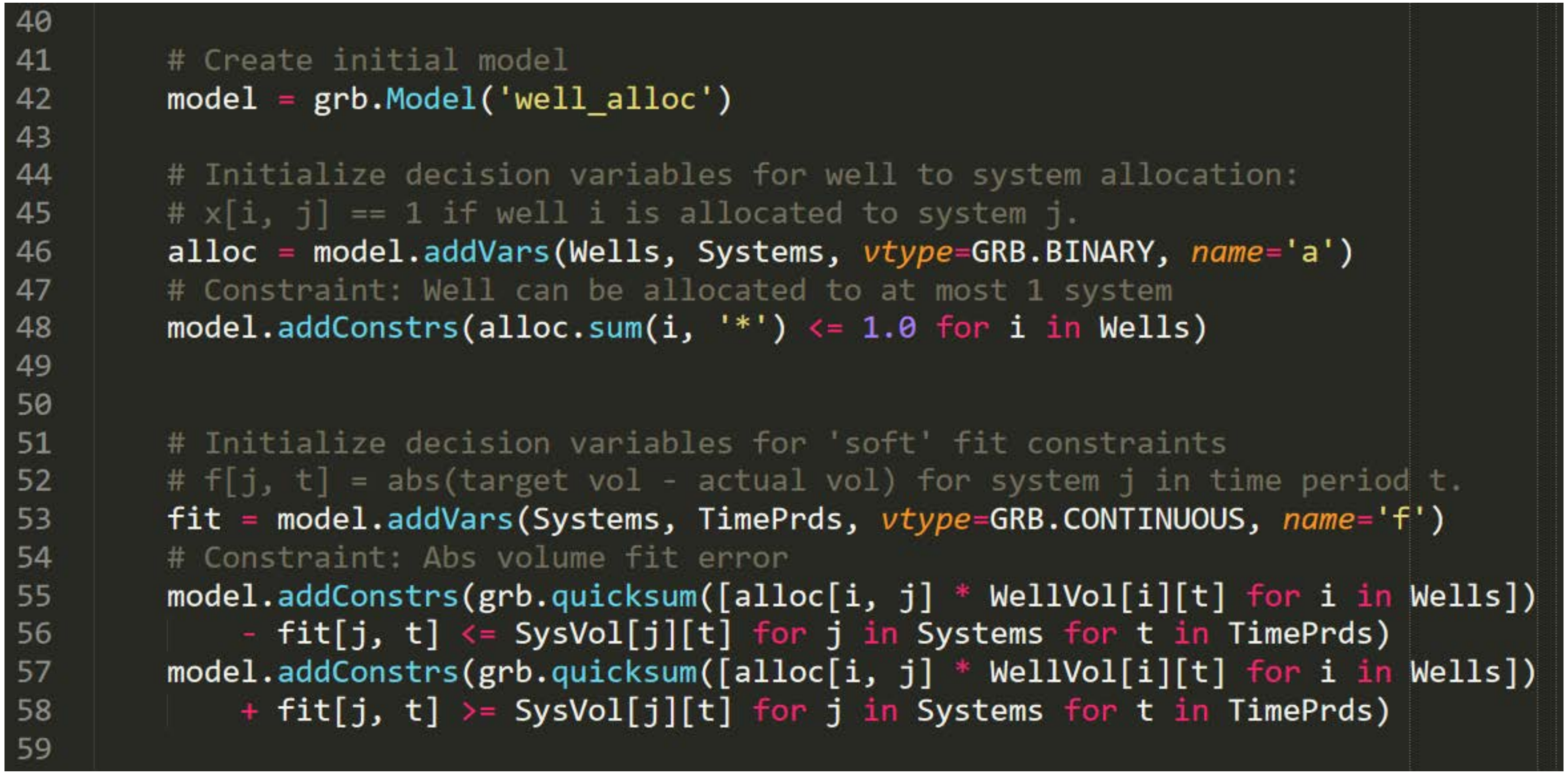


```python
import sys
import gurobipy as grb
from gurobipy import GRB

try:
    # Sample data
    Wells = range(14)
    Systems = range(2)
    TimePrds = range(6)
    SysVol = [ [20, 31, 40, 26, 26, 19],
               [57, 41, 37, 27, 23, 13] ]
    WellVol = [ [10,  5,  5,  4,  3,  1],
                [10,  6,  4,  4,  3,  1],
                [ 0, 10,  5,  5,  4,  3],
                [ 0, 10,  6,  4,  5,  4],
                [ 0,  0, 10,  6,  4,  5],
                [ 0,  0, 10,  3,  7,  5],
                [10,  5,  5,  4,  3,  1],
                [ 9,  5,  4,  3,  3,  2],
                [10,  6,  4,  4,  3,  1],
                [ 5,  5,  6,  2,  2,  1],
                [ 9,  8,  4,  3,  4,  4],
                [ 4,  2,  1,  1,  1,  1],
                [10,  5,  5,  4,  3,  1],
                [ 0,  5,  8,  6,  4,  2] ]
    WellSysCost = [ [3, 1],
                    [2, 2],
                    [0, 3],
                    [2, 2],
                    [3, 1],
                    [2, 2],
                    [1, 3],
                    [3, 0],
                    [3, 1],
                    [2, 2],
                    [2, 2],
                    [2, 2],
                    [1, 3],
                    [1, 3] ]

```

Implementation – Variables & Constraints



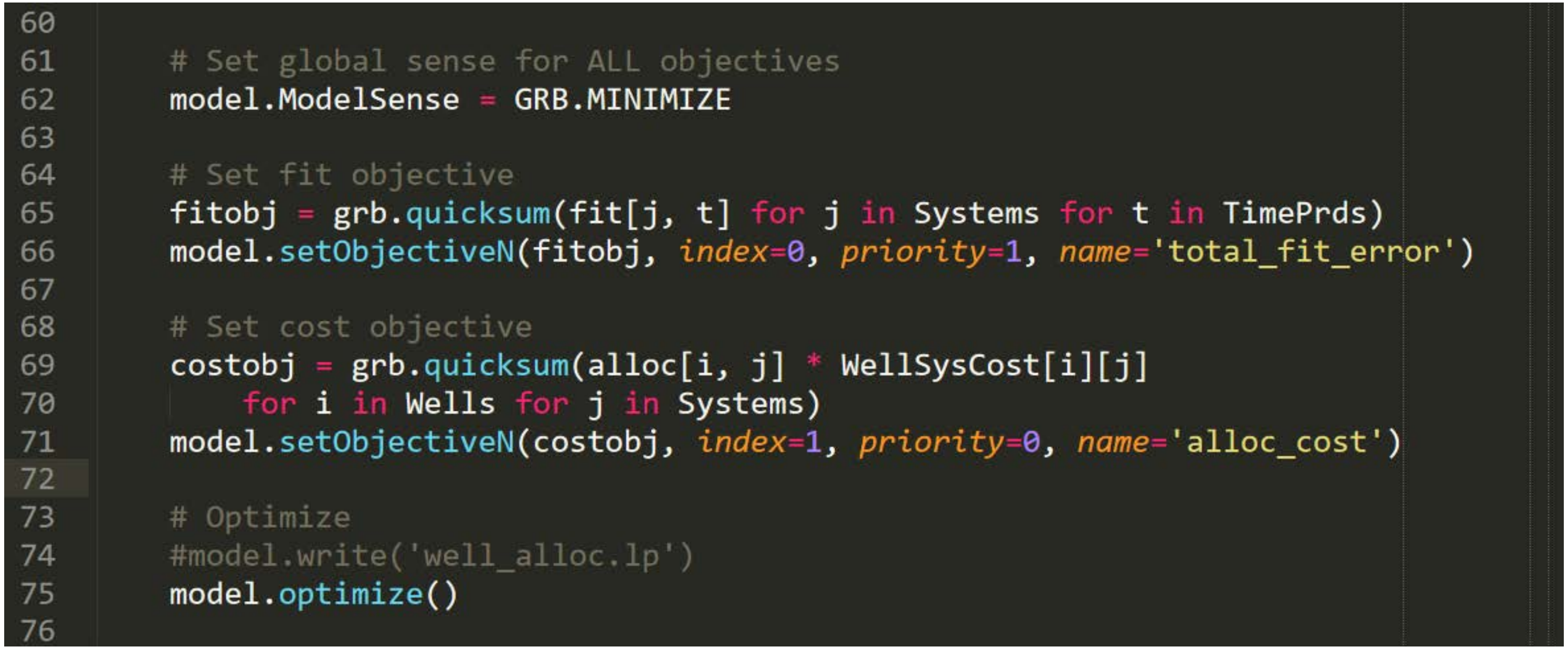
```
    # Create initial model
    model = grb.Model('well_alloc')

    # Initialize decision variables for well to system allocation:
    # x[i, j] == 1 if well i is allocated to system j.
    alloc = model.addVars(Wells, Systems, vtype=GRB.BINARY, name='a')
    # Constraint: Well can be allocated to at most 1 system
    model.addConstrs(alloc.sum(i, '*') <= 1.0 for i in Wells)


    # Initialize decision variables for 'soft' fit constraints
    # f[j, t] = abs(target vol - actual vol) for system j in time period t.
    fit = model.addVars(Systems, TimePrds, vtype=GRB.CONTINUOUS, name='f')
    # Constraint: Abs volume fit error
    model.addConstrs(grb.quicksum([alloc[i, j] * WellVol[i][t] for i in Wells])
        - fit[j, t] <= SysVol[j][t] for j in Systems for t in TimePrds)
    model.addConstrs(grb.quicksum([alloc[i, j] * WellVol[i][t] for i in Wells])
        + fit[j, t] >= SysVol[j][t] for j in Systems for t in TimePrds)

```

Implementation - Objective



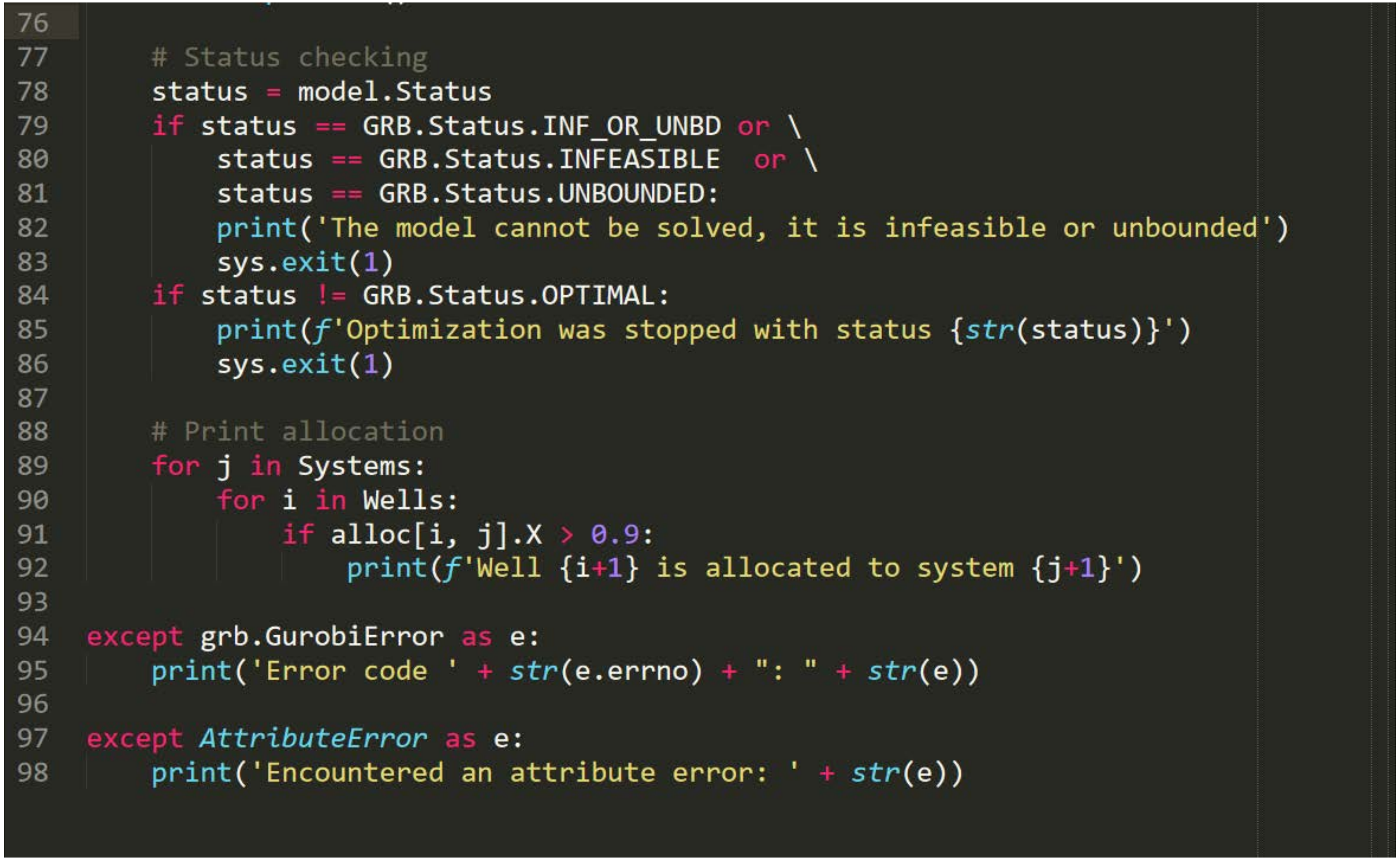
```
    # Set global sense for ALL objectives
    model.ModelSense = GRB.MINIMIZE

    # Set fit objective
    fitobj = grb.quicksum(fit[j, t] for j in Systems for t in TimePrds)
    model.setObjectiveN(fitobj, index=0, priority=1, name='total_fit_error')

    # Set cost objective
    costobj = grb.quicksum(alloc[i, j] * WellSysCost[i][j]
        for i in Wells for j in Systems)
    model.setObjectiveN(costobj, index=1, priority=0, name='alloc_cost')

    # Optimize
    #model.write('well_alloc.lp')
    model.optimize()
```

Implementation – Process Solution



```python

    # Status checking
    status = model.Status
    if status == GRB.Status.INF_OR_UNBD or \
        status == GRB.Status.INFEASIBLE  or \
        status == GRB.Status.UNBOUNDED:
        print('The model cannot be solved, it is infeasible or unbounded')
        sys.exit(1)
    if status != GRB.Status.OPTIMAL:
        print(f'Optimization was stopped with status {str(status)}')
        sys.exit(1)

    # Print allocation
    for j in Systems:
        for i in Wells:
            if alloc[i, j].X > 0.9:
                print(f'Well {i+1} is allocated to system {j+1}')

except grb.GurobiError as e:
    print('Error code ' + str(e.errno) + ": " + str(e))

except AttributeError as e:
    print('Encountered an attribute error: ' + str(e))
```

Variable hints

- Data is always changing
- Depending on region of country model runs can take from seconds to 24+ hours.
- Typically doesn't make catastrophic changes to the solution
- Variable hints can help algorithm make good decisions based on outside knowledge
- Drastically reduce time required to update models with new data

Closing thoughts

- **MIP is an excellent framework for stating a problem**
  - Not just big decision making
  - Modeling process helpful to understand problem
- **Advances in hardware/software have greatly expanded the scope of what is possible**
- **Using MIP for G&P allocation has some beneficial properties**
  - Can’t prove model wrong without making it better
  - Simple framework for adding new categories of ‘facts’
  - Balanced
  - Gap/Bounds provide some sense of opportunity
  - Patent pending
- **Gurobi features can significantly speed up development time (and accuracy)**

Thank You – Questions?

GUROBI OPTIMIZATION

The World's Fastest Solver

EAST DALEY CAPITAL

Your Next Steps

- If you haven’t already done so, please register for an account at http://www.gurobi.com
- For questions about Gurobi pricing contact sales@gurobi.com or sales@gurobi.de
- A recording of this webinar, including the slides, will be available in roughly one week