# Why MIP?

Dr. Edward Rothberg

GUROBI OPTIMIZATION

The World's Fastest Solver

# About the Speaker

## Ed Rothberg

CEO and Co-Founder

Gurobi Optimization

# Why MIP?

GUROBI OPTIMIZATION

The World's Fastest Solver

# A Brief History of MIP

# MIP – Similar History to Machine Learning

GUROBI OPTIMIZATION

**Many parallels between AI in 2010's and MIP in 1970's**

**Powerful technologies developed decades before**

- MIP:
  - Simplex method, 1947; Branch-and-bound, 1960
- ML:
  - Neural networks, 1951; Perceptron, 1957

**Big initial successes**

- MIP:
  - Optimal solution to a 48-city Traveling Salesman Problem, 1954
  - Oil companies use LP/MIP to save $Ms, 1960s
- ML:
  - Watson Jeopardy victory 2011
  - Speech recognition, face recognition, self-driving cars, 2010s

# MIP – Similar History to Machine Learning

GUROBI OPTIMIZATION

## Broad applicability

- MIP:
  - Significant modeling advances in the 1960’s and 1970’s
  - Important problems from many commercial application domains formulated as LP/MIP models
- ML:
  - “800 Million jobs to be replaced by AI worldwide by 2030” – McKinsey Global Institute

## Ready for mass adoption

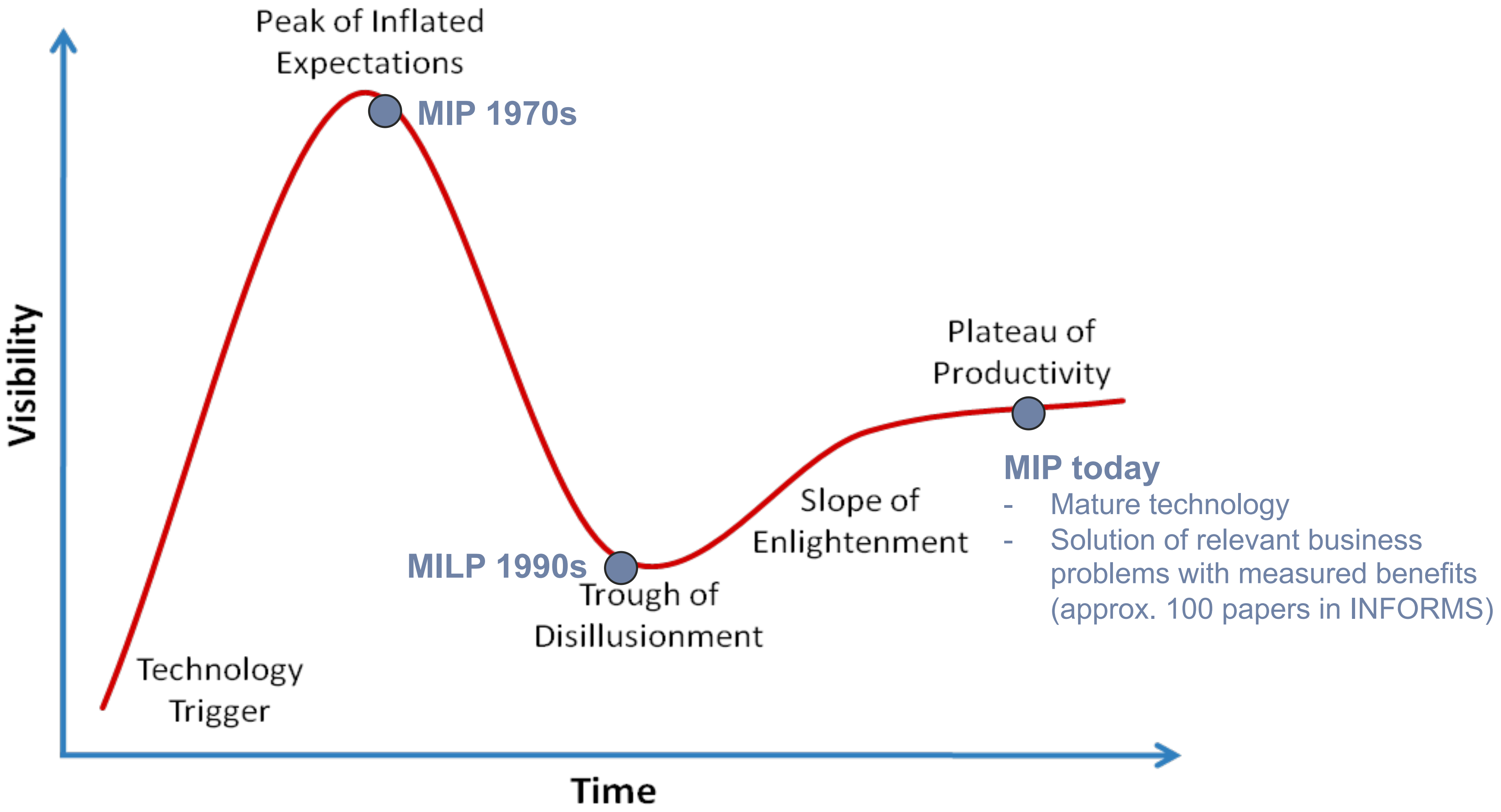
- MIP:
  - Commercial codes, running on “widely” available mainframe computers, available in 1970s
- ML
  - Software libraries galore (scikit-learn, PyTorch, TensorFlow)
  - Make it ”easy” to build machine learning models

# The Path Forward

## Some reasons for caution…

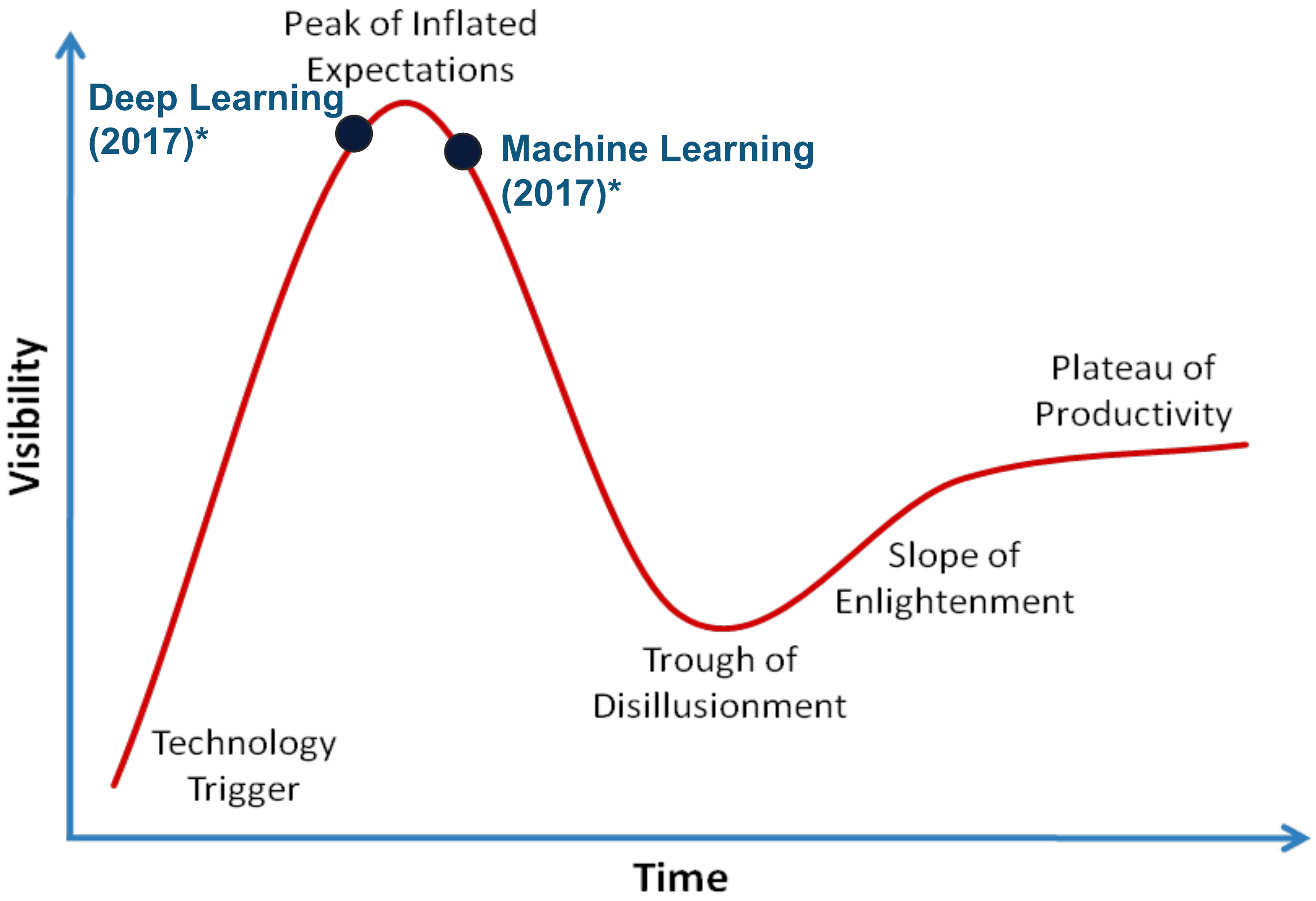
- MIP from 1970-2000
  - Difficult to build production applications
  - Resulting optimization models weren't solvable with then-current technology
  - Result: disillusionment with LP and MIP, lasted into 2000s
- ML from 2020-
  - Some recent calming of expectations (self-driving cars)
  - ???

# The Hype Cycle




Source: Gartner, Gurobi

# The Hype Cycle




Source: Gartner, Gurobi

# Progress: LP (1988-2004)

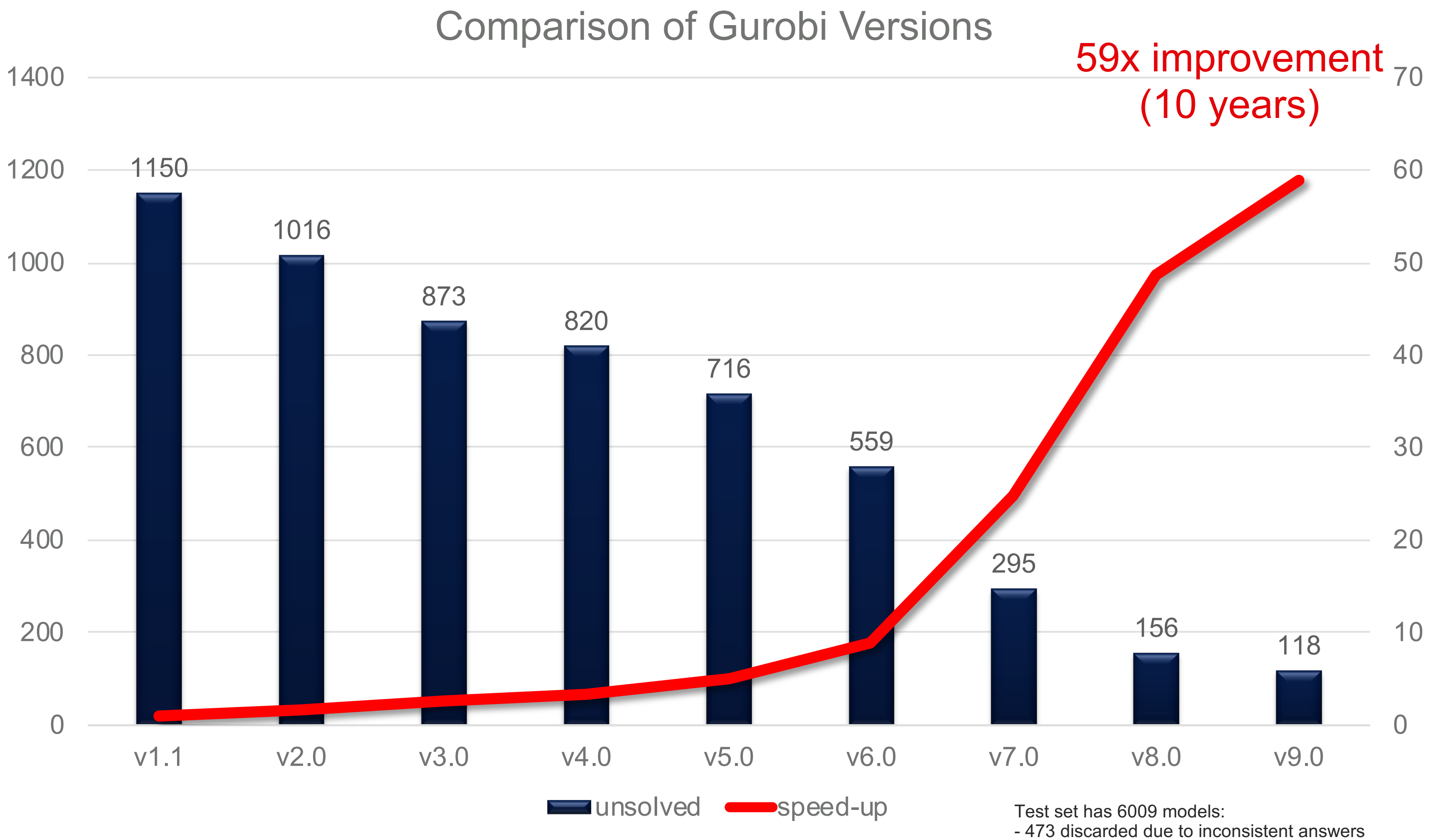
**From 1988-2004:**

- Improvement in algorithms:
  - Primal simplex in 1988 versus best of primal/dual/barrier: 3,300X
- Improvement in machines: 1,600X
- Net improvement: 5,300,000X
- Source: *Bixby*, Progress in Linear Programming, ORSA Journal on Computing

**Impact:**

- What would take 2 months before now takes 1 second
- LP is now (mostly) considered a solved problem
  - Regularly solve models with millions of variables and constraints

# Progress: MIP (2008-2020)




Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 6009 models:
- 473 discarded due to inconsistent answers
- 1580 discarded that none of the versions can solve
- speed-up measured on >100s bracket: 2197 models

# Industries Transformed by MIP – Supply Chain

In the 1980's, software dominated by rules of thumb

- Example: theory of constraints (The Goal, Goldratt)

MIP widely adopted in the 1990's

Now the standard technology for supply-chain

- SAP, Oracle, JDA, Manhattan Associates, …

# Industries Transformed by MIP – Electrical Power

Electrical power deregulated in the late 1990's

Need to create a market for electricity

Early solution techniques:

- Heuristics (Lagrangean relaxation)
- MIP (lots of models; no real usage)

EPRI report, June 1989:

- "Mixed-integer programming (MIP) is a powerful modeling tool. 'They are, however, theoretically complicated and computationally cumbersome'"

DIMACS meeting 1999:

- Bob Bixby demonstrated that MIP had improved to the point where practical power models could be solved

Within a few years, nearly every grid operator in the world was using MIP to solve these models

# Industries Transformed by MIP – Sports Scheduling

GUROBI
OPTIMIZATION

**Computing sports schedules quite complicated**

- Stadium constraints, travel constraints, TV schedules, …

**Done by hand for decades**

- Example: Henry and Holly Stephenson scheduled Major League Baseball “by hand” from 1981-2004

**Schedules now done using MIP:**

- MLB 2004-2019
- NFL 2007-

# Franz Edelman Award in Operations Research

Annual award for the OR project with the biggest impact on society

Measured financial impact of finalists: $257 billion

Large fraction use LP or MIP to achieve results

https://www.informs.org/Recognizing-Excellence/INFORMS-Prizes/Franz-Edelman-Award

# Customer Applications of MIP

## (2011-2012)

Accounting
Advertising
Agriculture
Airlines
ATM provisioning
Compilers
Defense
Electrical power
Energy
Finance
Food service
Forestry
Gas distribution
Government
Internet applications
Logistics/supply chain
Medical
Mining

National research labs
Online dating
Portfolio management
Railways
Recycling
Revenue management
Semiconductor
Shipping
Social networking
Sourcing
Sports betting
Sports scheduling
Statistics
Steel Manufacturing
Telecommunications
Transportation
Utilities
Workforce Management

Shell
TransUnion
at&t
SCHNEIDER LOGISTICS
BOEING
DEPARTMENT OF DEFENSE UNITED STATES OF AMERICA
ORACLE
WM WASTE MANAGEMENT
Nokia Siemens Networks
STARBUCKS COFFEE
U·S AIRWAYS
LEANLOGISTICS
Marriott
ProCom bringt Transparenz
CSX
FCC Federal Communications Commission
Google
HUAWEI
SAP
NFL
DAIMLER
YUTONG
Ferrari
AMD
Prudential
Walmart Save money. Live better.
ESTĒE LAUDER
ABB
nielsen
BM&FBOVESPA The New Exchange
SIEMENS
Bank of America
Microsoft
SNOWDEN
NYISO
bhpbilliton
Lufthansa

# Progress: MIP

## Impact: widespread adoption

- Wide variety of industries
- Wide variety of time-scales
  - Planning – make recommendations over very long time horizons
  - Real-time – make decisions in fractions of a second
- Wide variety of deployment environments
  - In-house desktops and servers
  - Cloud machines
  - Embedded systems
- Wide variety of skill levels
  - OR PhDs
  - Computer-savvy computer scientists, biologists, statisticians, …

# MIP is not an Algorithm

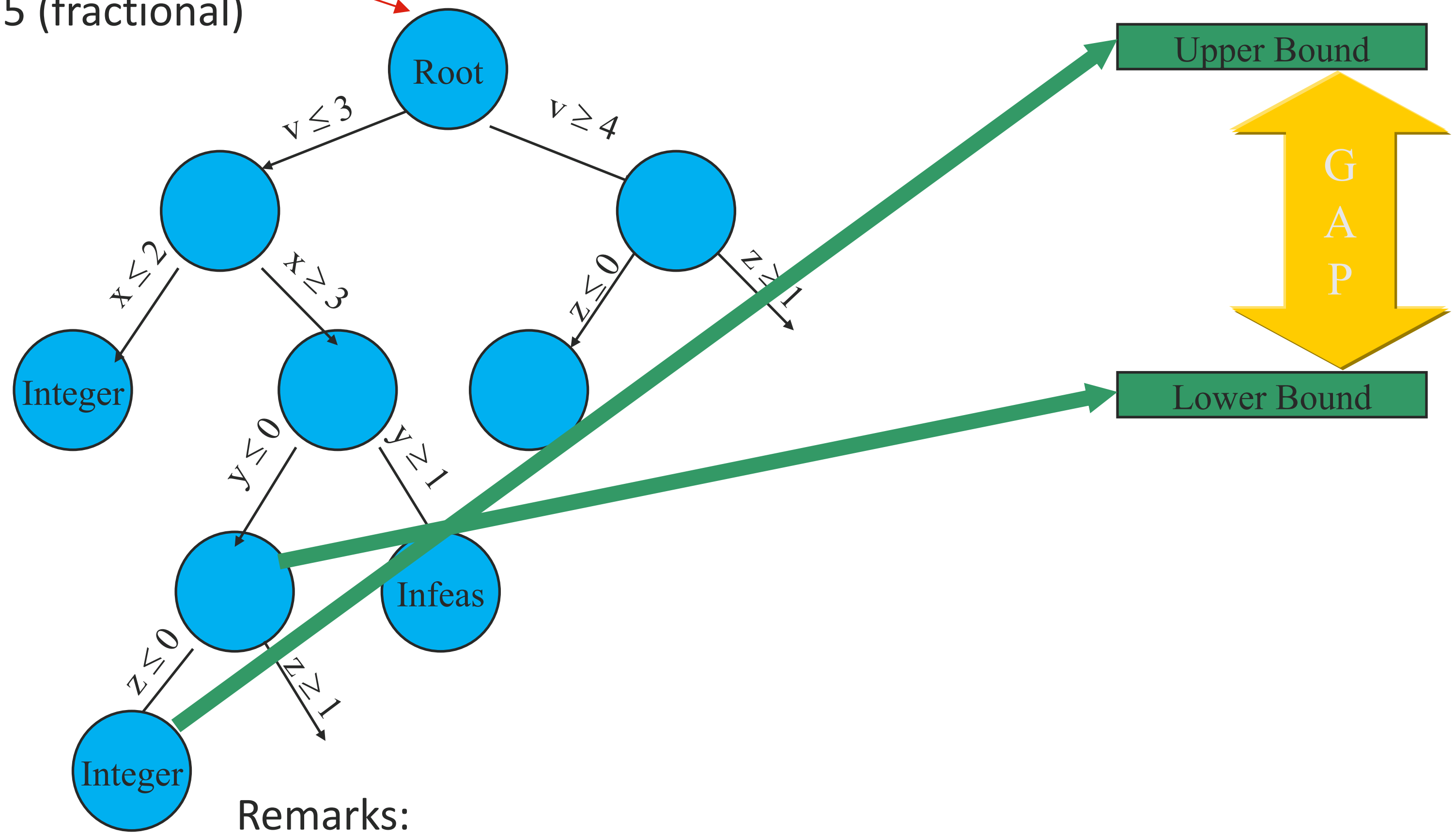
# MIP Problem Statement

A *mixed-integer program* (MIP) is an optimization problem of the form

$$\boxed{\begin{aligned} &\mathit{Minimize} && c^T x \\ &\mathit{Subject\ to} && Ax = b \\ & && l \le x \le u \\ &\text{some or all } x_j \text{ integer} \end{aligned}}$$

# MIP Solution Framework: LP based Branch-and-Bound

Solve LP relaxation:
v=3.5 (fractional)




Remarks:

(1) GAP = 0 $\Rightarrow$ Proof of optimality

(2) In practice: Often good enough to have good solution

# MIP is not an Algorithm

MIP is typically thought of as an algorithm

- Relaxation-based branch-and-bound

A limiting point of view

Better to think of it as a declarative framework for stating optimization problems

- Backed by a rich mathematical foundation
  - Linear programming, duality, polyhedral theory, etc.

Allows for a variety of algorithms to be applied

- A giant “bag of tricks”
- Applied systematically and automatically

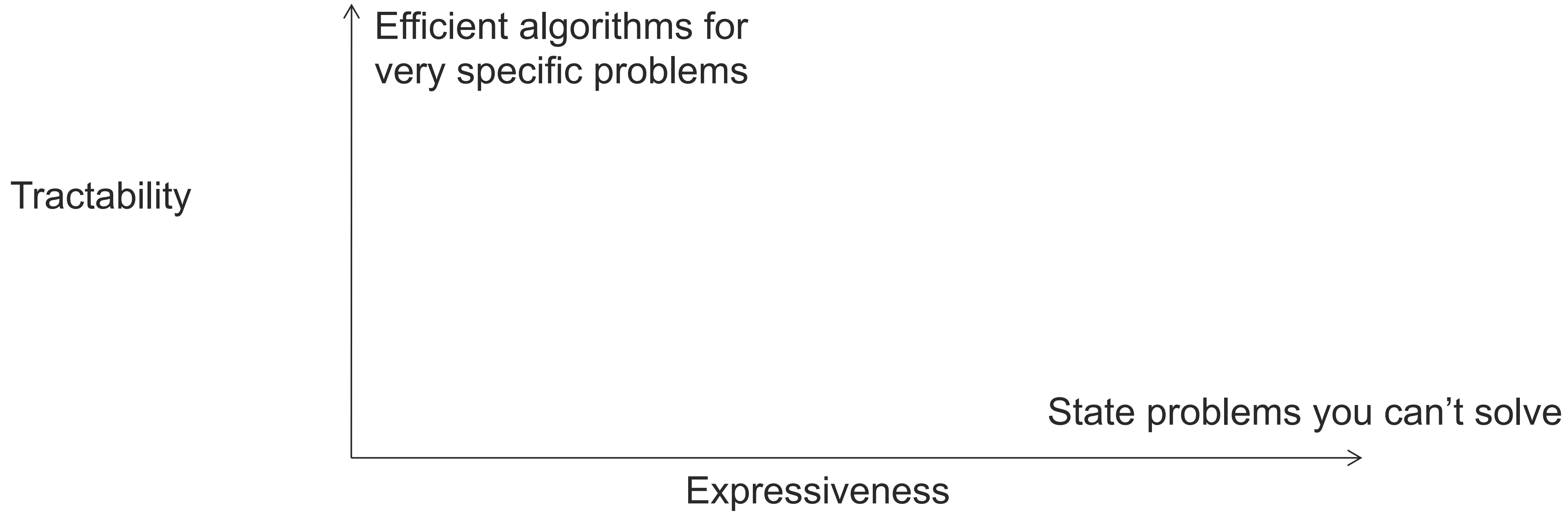
Only when considered together you get a robust and efficient method

# Expressiveness Versus Tractability

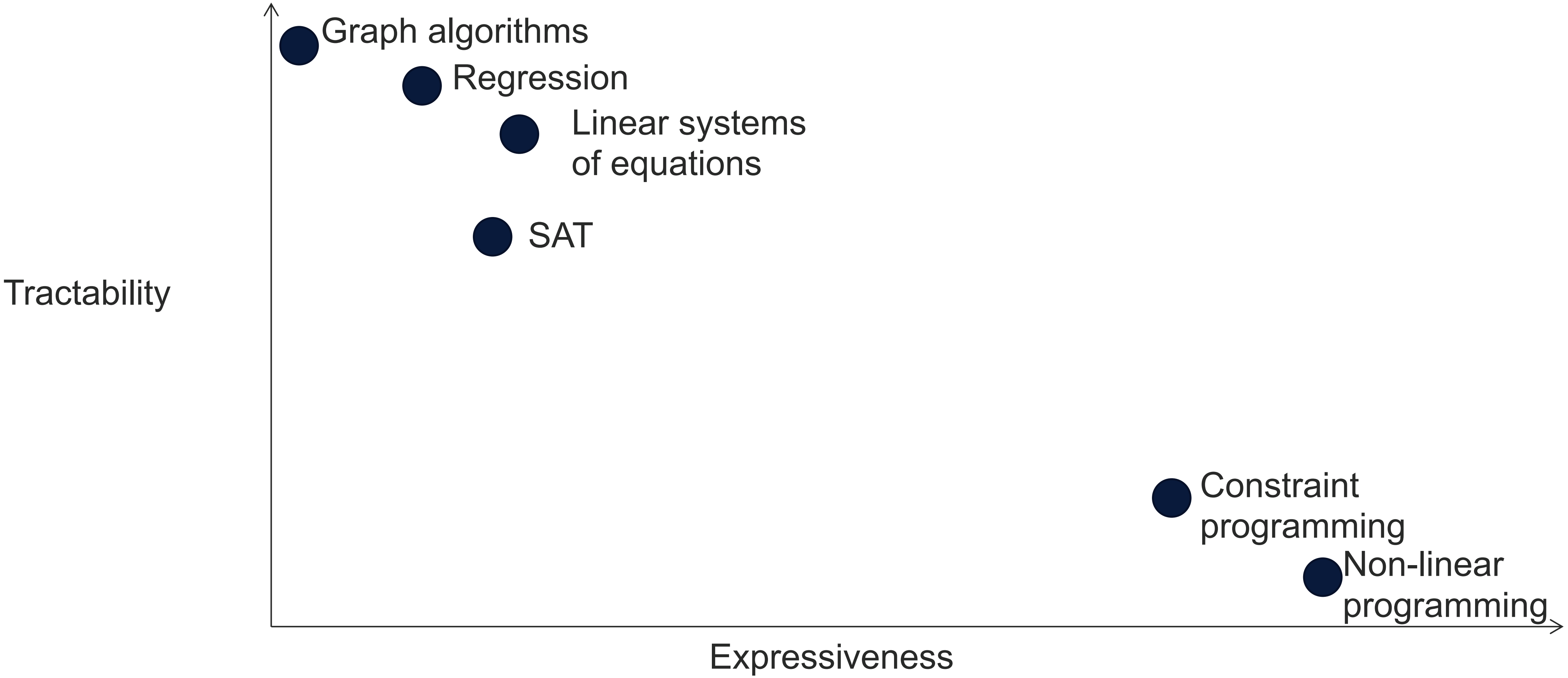
# Problem and Algorithms

GUROBI OPTIMIZATION

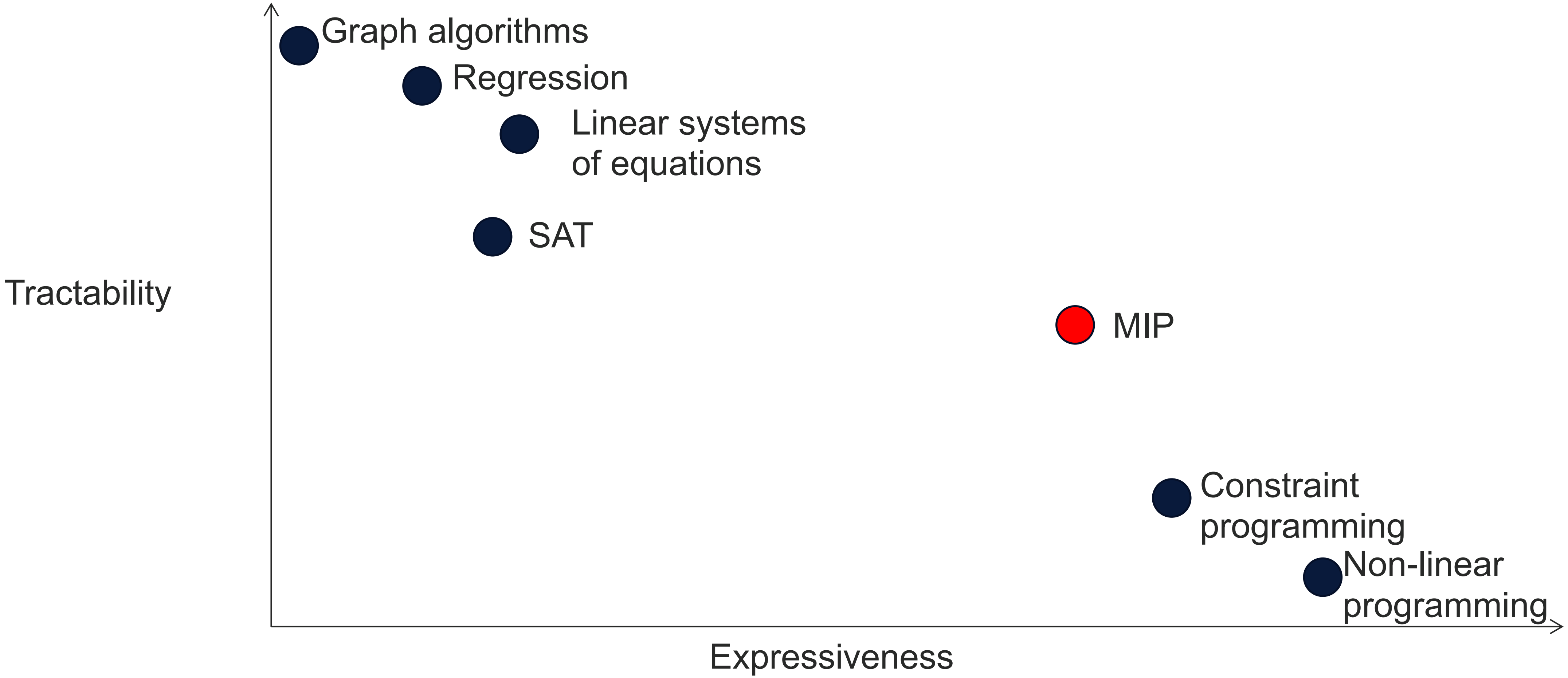
## Trade-off between expressiveness and tractability



# A Few Examples



# A Few Examples



# Important Properties of MIP

**Declarative**

- No need to rewrite algorithm to accommodate additional constraints

**Expressive**

- Adding side constraints typically leaves the problem type unchanged

**Regular problem structure (*Ax=b*)**

- Allows for fairly simple recognition of useful/special structure

# MIP Improvements and Where They Come From

# MIP Improvements in Gurobi 7.5 (by Category)

**Presolve - 7.8%**

- 8 improvements, largest 2.0%

**Node presolve - 4.2%**

- 5 improvements, largest 1.3%

**Symmetry - 7.0%**

- 3 improvements, largest 3.1%

**LP - 17.5%**

- Many small improvements

**Cutting planes - 14.3%**

- 6 improvements, largest 6.3%

**Branching - 2.8%**

**Heuristics - 4.6%**

- 5 improvements, largest 1.4%

# Opening the Bag of Tricks

GUROBI OPTIMIZATION

**MIP draws from many different disciplines**

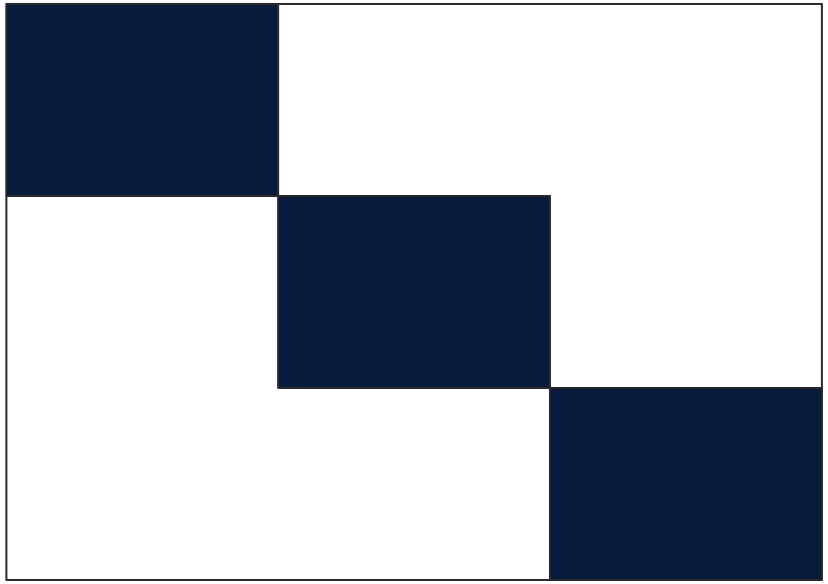
**A few examples:**

- Graph algorithms:
  - Bi-connected components
- Number theory
  - Modular multiplicative inverse
- Meta-heuristics
  - Relaxation Induced Neighborhood Search (RINS)

# Bi-Connected Components

# Disconnected Components

A =

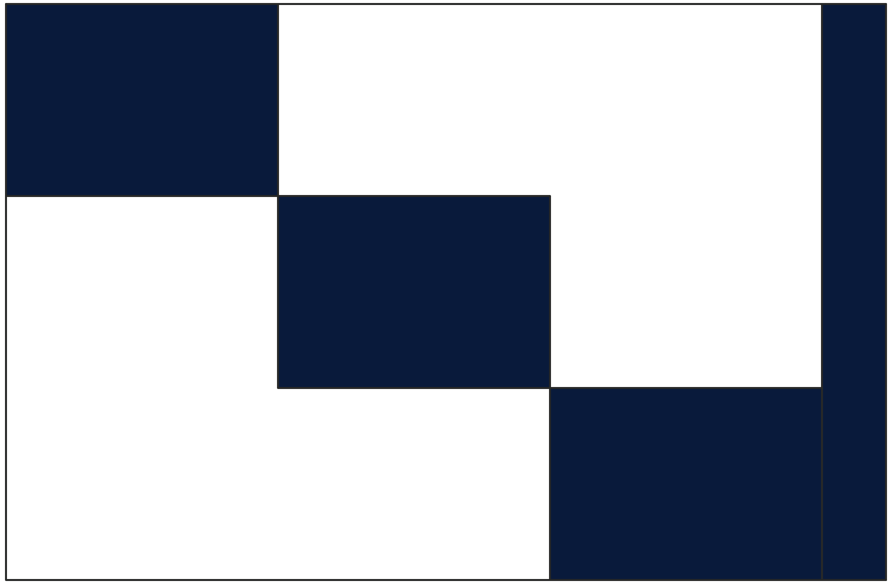


As a single model, runtime grows as #nodes$^n$ for n components

When split, runtime grows as #nodes*n

# Nearly Disconnected Components

$A =$



Identify variables that disconnect the model to *create* disconnect components

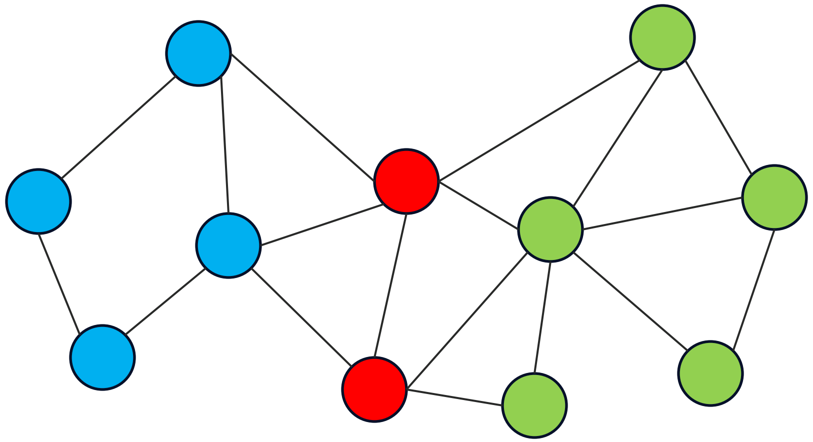
# Identifying Variables that Disconnect the Model

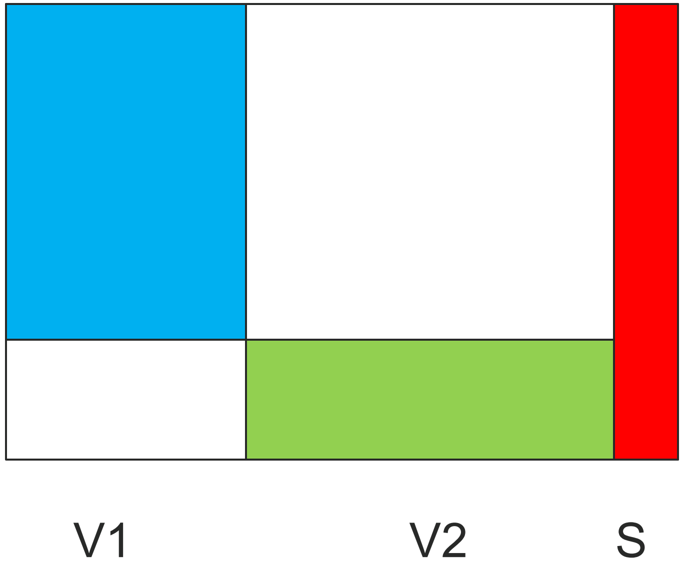
Problem is equivalent to finding a *vertex separator* in the *intersection graph* of A

- Intersection graph:
  - One node for each variable
  - An edge (i,j) whenever variables i and j ever appear in the same constraint
- Vertex separator
  - A set of vertices whose removal disconnects the graph

# Intersection Graph

GUROBI OPTIMIZATION

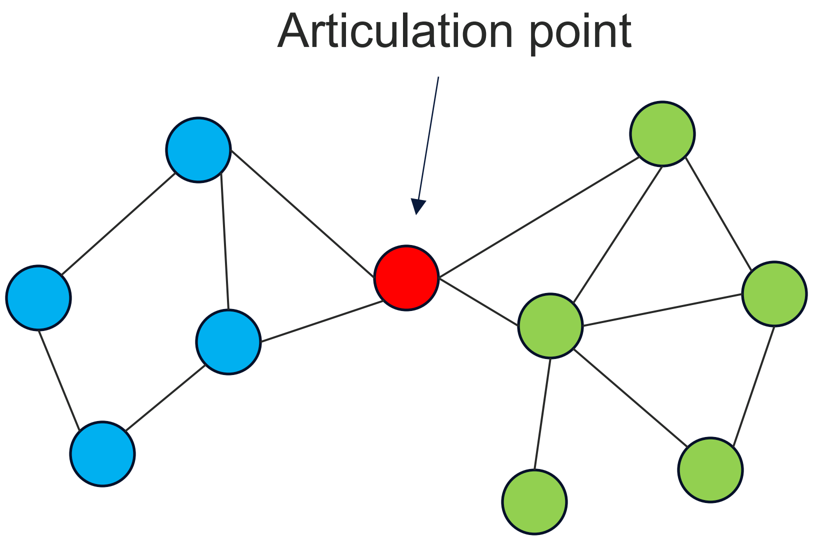


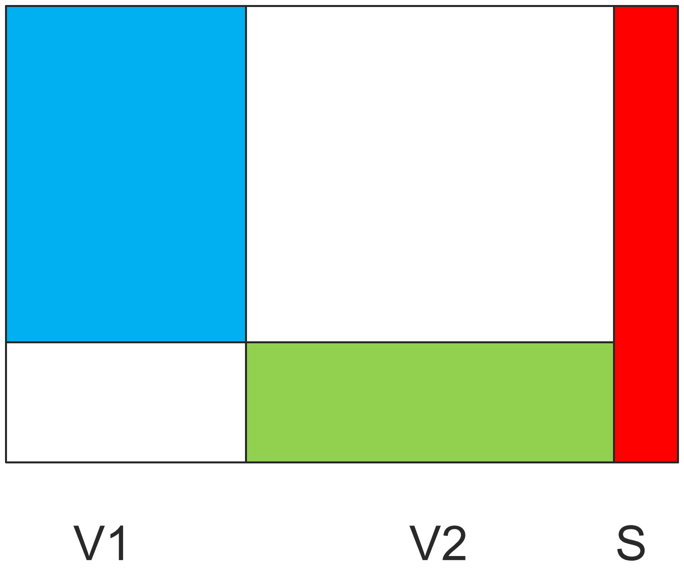



**No edges between vertices in V1 and V2 implies…**

- No pair of variables in V1 and V2 participate in the same constraint

# Bi-Connected Components



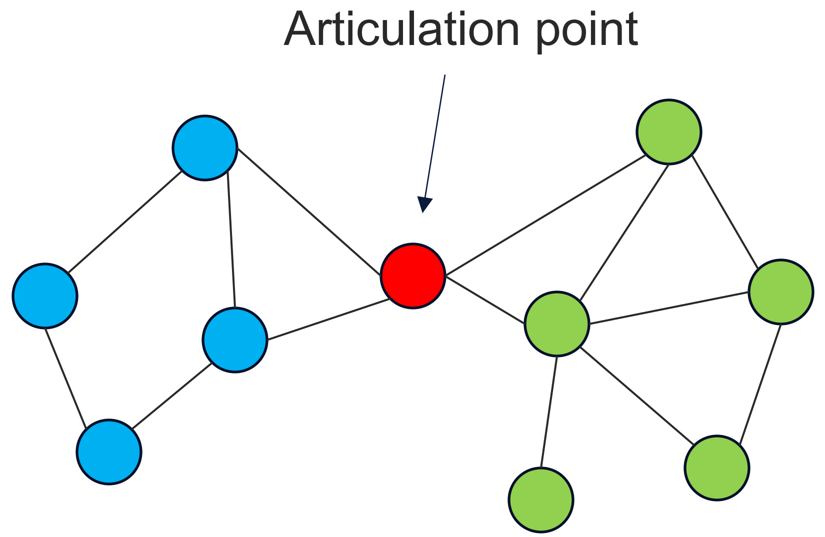

**Identify a single vertex that disconnects the graph**

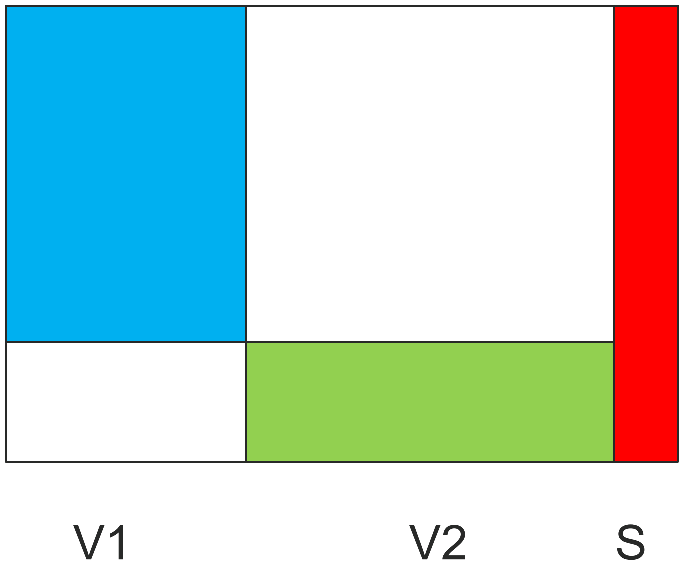
**Algorithm by Lipton and Tarjan**

**Linear time in # of edges in graph**

- Identifies all vertices that disconnect graph

# Bi-Connected Components




**If articulation point is a binary variable**

- Fix variable to 1 – solve remaining pieces independently
- Fix variable to 0

**Exploits bi-connected structure to create disconnected components**

**Often much faster than solving whole problem**

# Number Theory

# Modular Inverse Reduction

## Consider

- $a x + b y = c$ (possibly after aggregation)
- $x, y$ are integer variables
- $a$, $b$ and $c$ are integers, $a > 1$
- Assume $GCD(a,b) = 1$
  - Divide through by GCD otherwise

## Compute *modular multiplicative inverse* of a

- Integer *m* such that $a m = 1 \pmod b$
  - Computed using *extended Euclidean algorithm*
- Transformation
  - $a x + b y = c$
  - $m a x + m b y = m c$
  - $x = m c \pmod b$

# Modular Inverse Reduction - Example

**$5\ x_1 + 10\ x_2 + 3\ x_3 + 10\ x_4 = 31$ (all integer variables)**

- Substitute $y = x_1 + 2\ x_2 + 2\ x_4$
- $5\ y + 3\ x_3 = 31$

**Modular inverse of 3 mod 5 is 2 ($2 * 3 = 1 \pmod 5$)**

- $10\ y + 6\ x_3 = 62 \pmod 5$
- $x_3 = 2 \pmod 5$
- $x_3 = \{ 2, 7, 12, 17, \ldots \}$

**Exploiting this observation**

- In MIP search tree, if $x_3 = 3.5$ in relaxation
    - Normal branch: x <= 3 or x >= 4
    - Better branch: x <= 2 or x >= 7

**Impact**

- Runtime drops from 10000+s to 0.01s for one family of models
- Often a significant win when it can be applied

# Meta-Heuristics

# Sub-MIP Heuristics

**Meta-heuristics play a big role in MIP**

**Basic structure: local search through sub-MIPs**

- Fix a set of variables
- Solve a MIP on the remainder
  - Recursive (sub-MIPs explored in remainder model too)
- A form of Very Large Neighborhood Search (VLNS)

**Population-based meta-heuristics**

- MIP tree search naturally finds multiple solutions
- Can use approaches that combine solutions

**Most effective meta-heuristic in MIP:**

- Relaxation Induced Neighborhood Search (RINS)

# Relaxation Induced Neighborhood Search

**A population of two**

- Current incumbent solution – integer feasible but (probably) not optimal
- Current relaxation solution – optimal but not integer feasible

**Simple sub-MIP**

- Fix variables that agree in two solutions

**Diversified**

- Different relaxation at each branch-and-bound node

**Adaptive**

- Neighborhood is large when optimality gap is large, small when gap is small

**Often the best approach to find a good solution is…**

- Find a bad solution and improve it

# Meta-meta-heuristics

Feed domain-specific heuristic solutions to MIP solver

MIP solver works to improve those solutions

- Systematic exploration of space, using cutoff from provided solution
- Additional input to MIP meta-heuristics

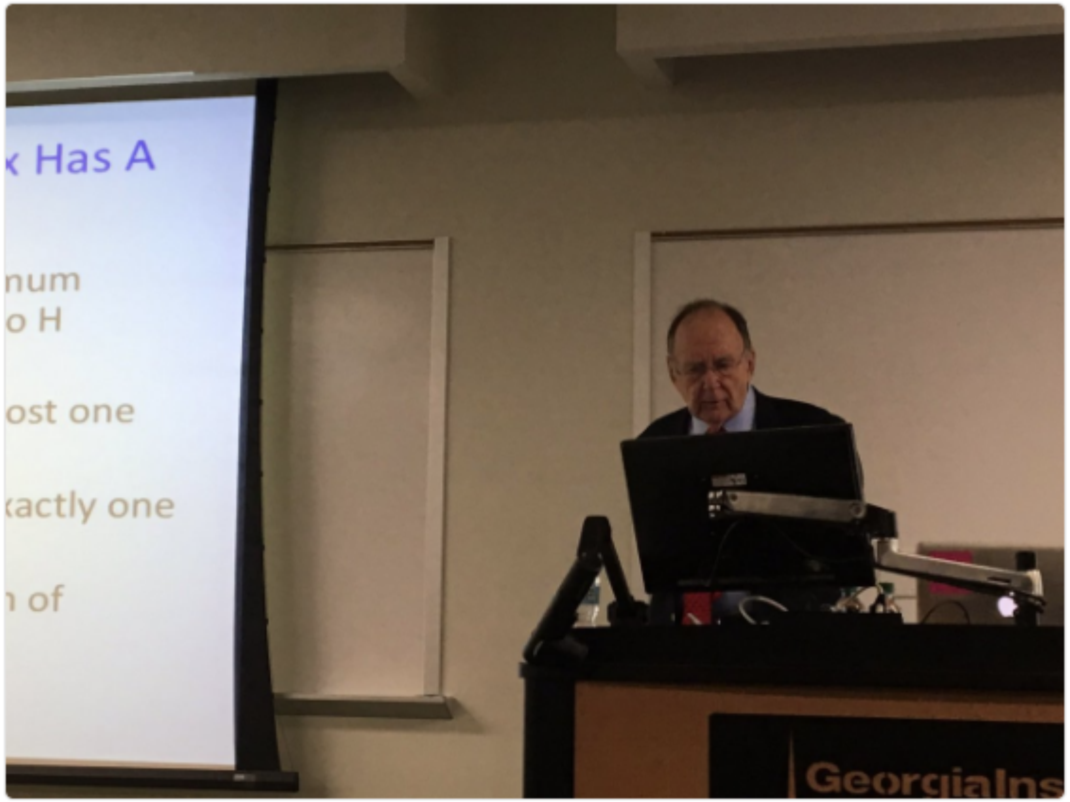
# Conclusions

# Always Try MIP

**Shabbir Ahmed**
@Shabbir0Ahmed

Follow

Richard Karp quotes a colleague "Always try integer programming, it might work"




8:27 AM - 13 Mar 2017

# Thank You

GUROBI OPTIMIZATION

The World's Fastest Solver

# Coming Up!

- **Tuesday, June 23rd**
  - Introduction to Modeling with Python
  - Advanced Modeling
  - Ask the Experts: Technician Routing & Scheduling JPME

- **Wednesday, June 24th**
  - Compute Server and Cloud Demo
  - Introduction to Algorithms
  - Advanced Algorithms
  - Ask the Experts: Gurobi R&D Leaders

- **Thursday, June 25th**
  - Customer Case Study: KPMG
  - Customer Case Study: NFL
  - Ask the Experts: Chat with Michael North, VP of Broadcast Planning & Scheduling at NFL