Large-Scale Optimization and Logical Inference

John Hooker
Carnegie Mellon University

October 2014

University College Cork

Research Theme

- Large-scale **optimization** and **logical inference**.
  - Optimization on large datasets.
  - Extraction of information from large datasets.
  - Application to industry problems.

Research Theme

- Large-scale **optimization** and **logical inference**.
  - Optimization on large datasets.
  - Extraction of information from large datasets.
  - Application to industry problems.
- Why combine these?

Research Theme

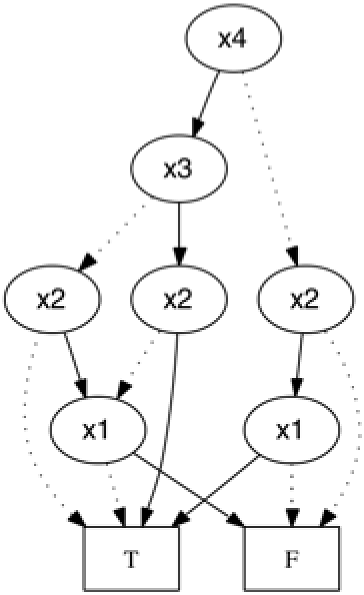
- …to exploit **underlying unity** of optimization and logical inference
  - Both are forms of **projection**.
  - Also, combine **optimization** and **constraint programming**

Research Streams

- Use **decision diagrams** for optimization and constraint programming.
  - Also for postoptimality analysis.
- Use **logic-based Benders decomposition** for optimization and inference
  - Computes a **projection**.
  - Combines integer programming and constraint programming.
- Use **optimization methods** for logical inference.
  - Probability logic, belief logics, etc.

Decision Diagrams

- Graphical encoding of a boolean function
  - Historically used for circuit design & verification
  - Adapt to optimization and constraint programming
    - T. Hadžić and JH (2007)



Decision Diagrams

- Collaborators…
  - Henrik Andersen
  - David Bergman
  - André Ciré
  - **Tarik Hadžić (now at UTRC, Cork)**
  - Samid Hoda
  - Willem-Jan van Hoeve
  - **Barry O'Sullivan (UCC, Insight Centre)**
  - Peter Tiedemann

Decision Diagrams

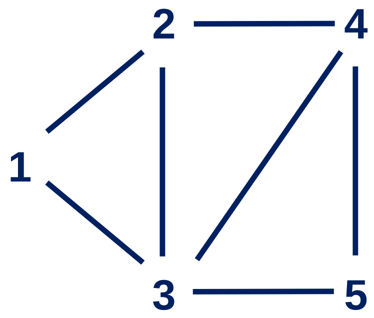
- Collaborators…
  - Henrik Andersen
  - **David Bergman***
  - **André Ciré****
  - Tarik Hadžić (now at UTRC, Cork)
  - Samid Hoda
  - Willem-Jan van Hoeve
  - Barry O'Sullivan (UCC, Insight Centre)
  - Peter Tiedemann

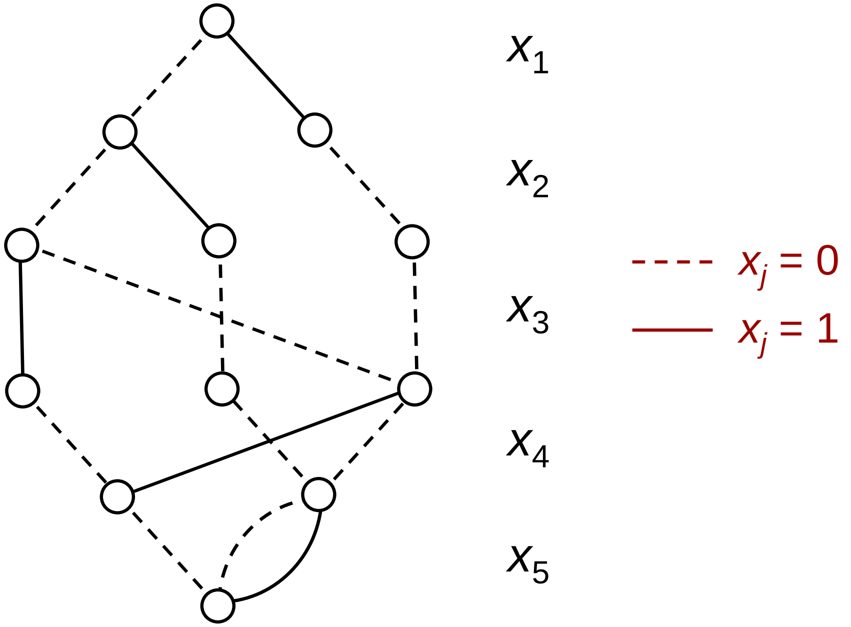
*****2014 *Doctoral Thesis Award*, Assoc. for Constraint Programming
**2014 *Best Student Paper Award,* INFORMS Computing Society

Decision Diagrams

Example:
Stable set problem

Find max-weight subset
of nonadjacent vertices

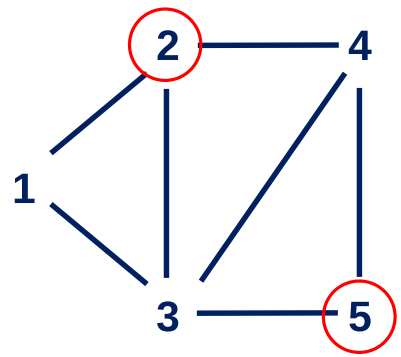


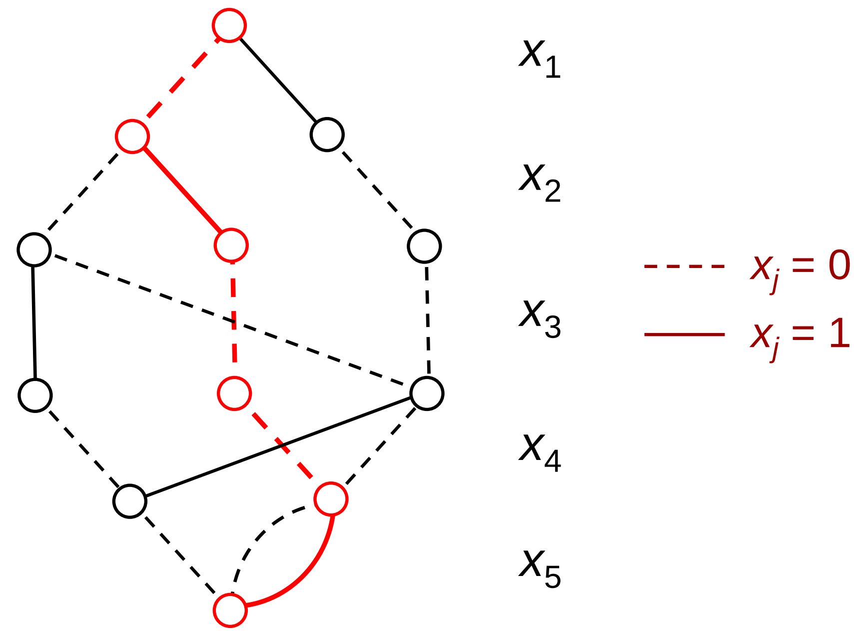


Decision Diagrams

Example:
Stable set problem

Find max-weight subset
of nonadjacent vertices



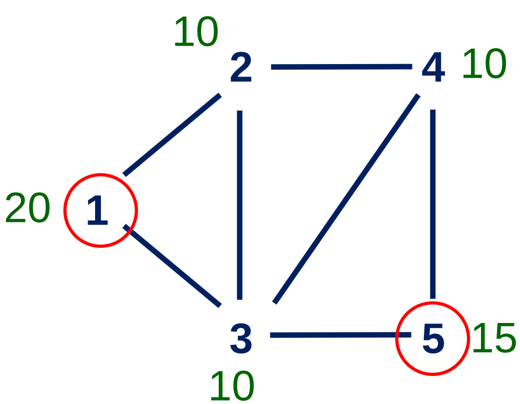


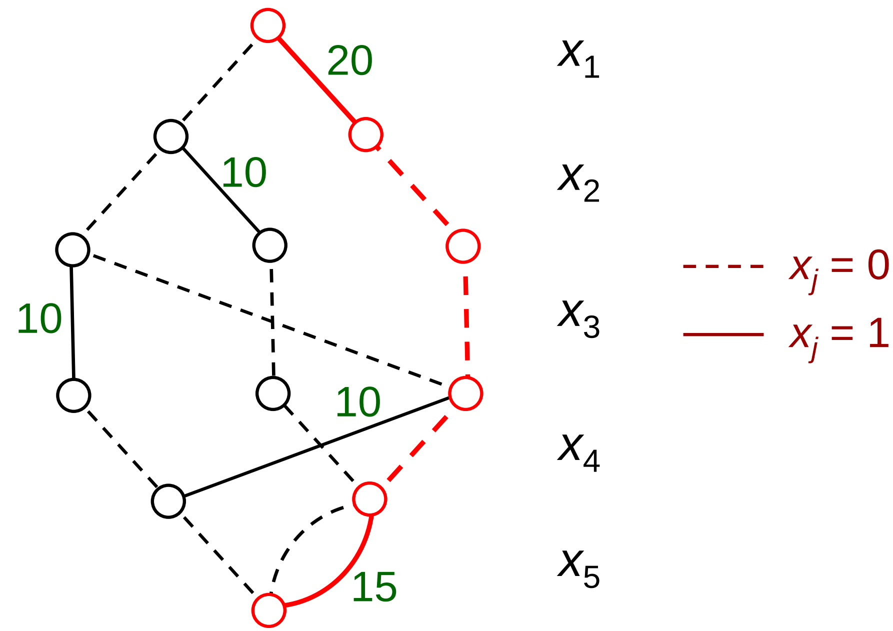
Each path corresponds to a feasible solution.

Decision Diagrams

Example:
Stable set problem

Find max-weight subset
of nonadjacent vertices





Longest path corresponds to an optimal solution.

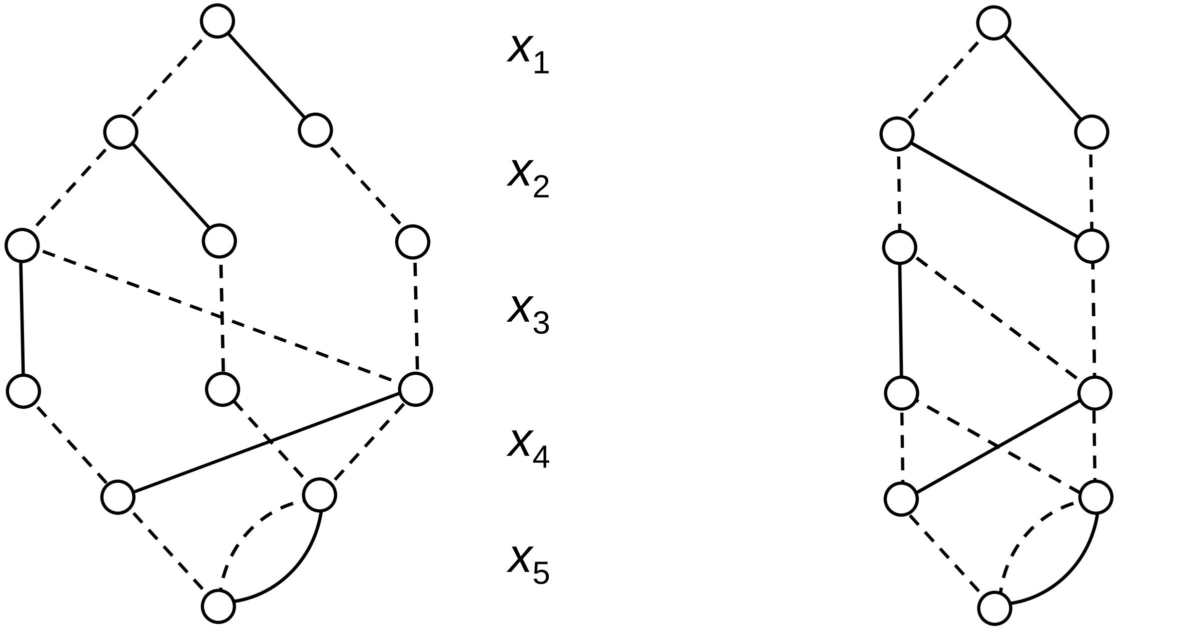
Decision Diagrams

- Key idea: Use **relaxed** decision diagrams
  - Represent a superset of the feasible set
    - … with a **limited-width** diagram.
  - The idea was first used to strengthen **propagation** in constraint programming solvers.
    - Andersen, Hadžić, JH, Tiedemann (2007)
    - Reduced 1 million node search trees to 1 node.

Decision Diagrams

- Key idea: Use **relaxed** decision diagrams
  - Represent a superset of the feasible set
    - … with a **limited-width** diagram.
  - The idea was first used to strengthen **propagation** in constraint programming solvers.
    - Andersen, Hadžić, JH, Tiedemann (2007)
    - Reduced 1 million node search trees to 1 node.
- Shortest (longest) path in the decision diagram provides a **bound** on optimal value.
  - Leads to general-purpose **optimization** method.

Decision Diagrams

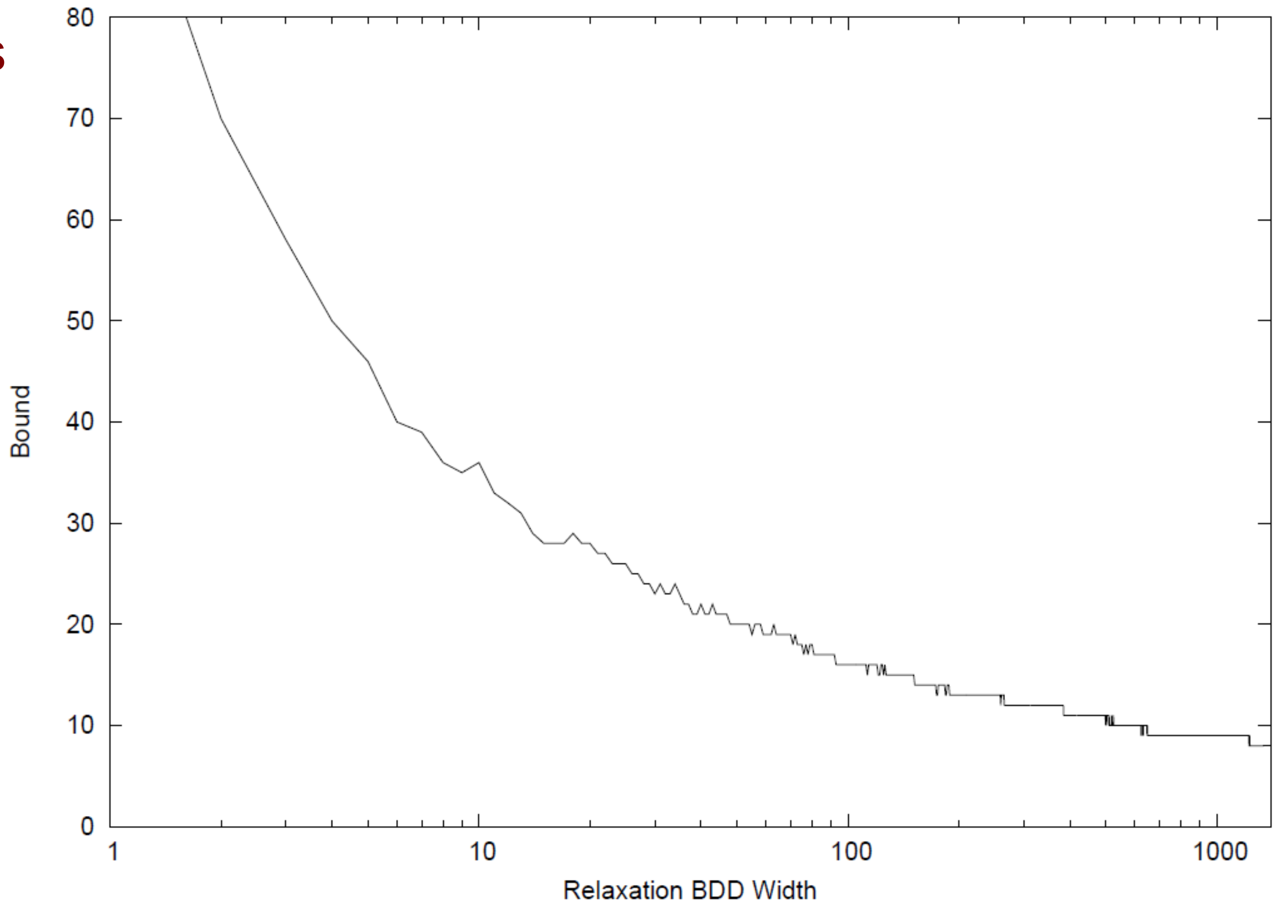


Exact diagram
Width = 3

Relaxed diagram
Width = 2

Decision Diagrams

- Wider diagrams yield tighter bounds
  - But take longer to build.
  - Adjust width dynamically.



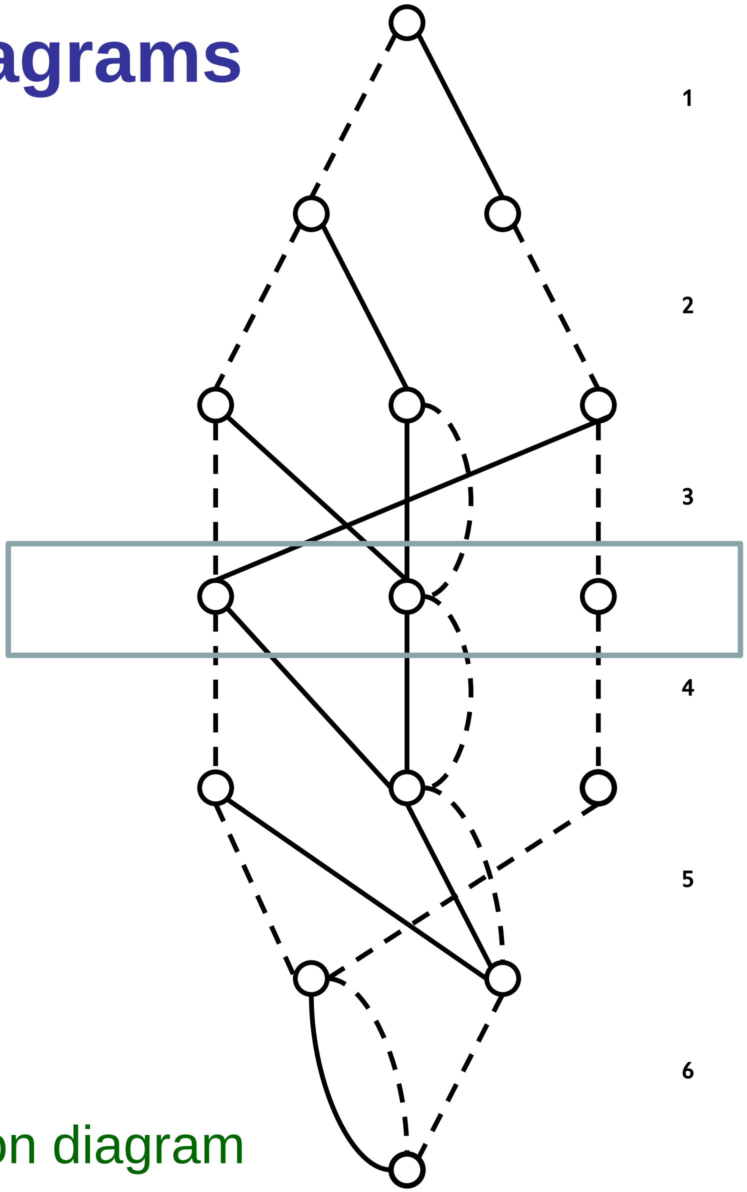
Decision Diagrams

- Solve using a novel branch-and-bound algorithm.
  - Branch on nodes in **last exact layer** of relaxed decision diagram.
    - …rather than branching on variables.

Decision Diagrams

Branching in a relaxed decision diagram

Diagram is exact down to here

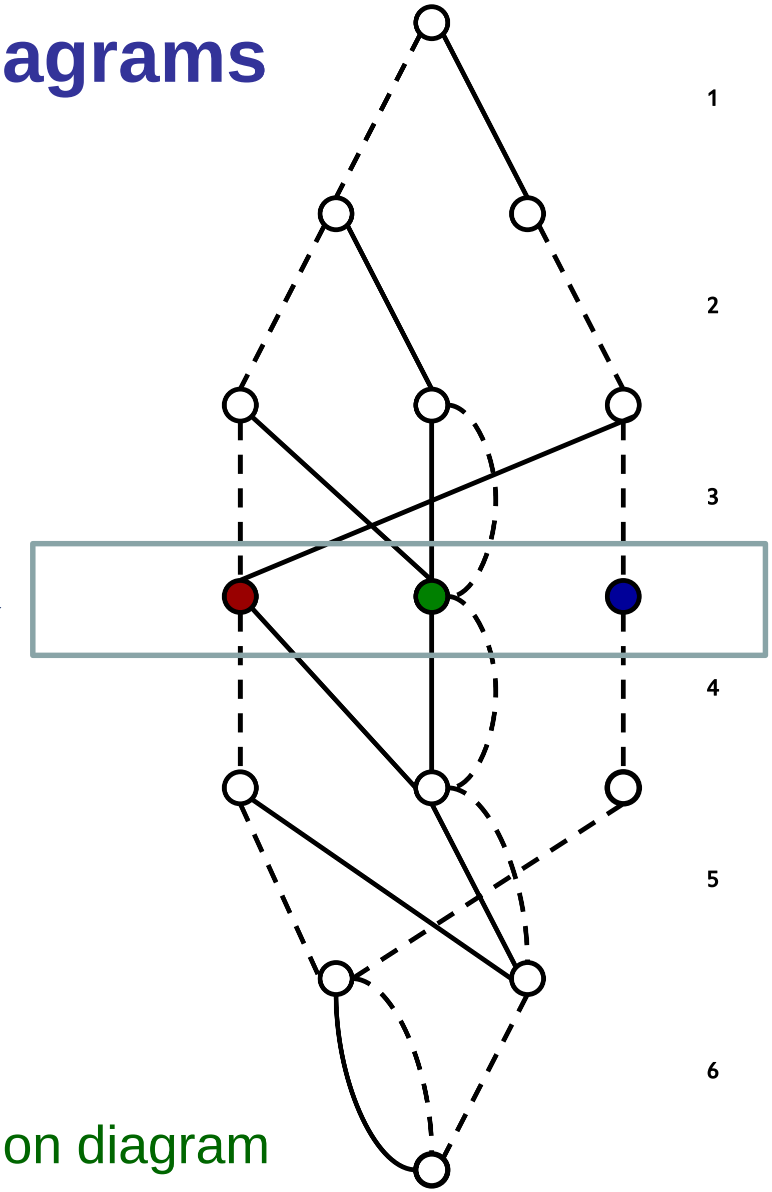


Relaxed decision diagram

Decision Diagrams

Branching in a relaxed decision diagram

Branch on nodes in this layer

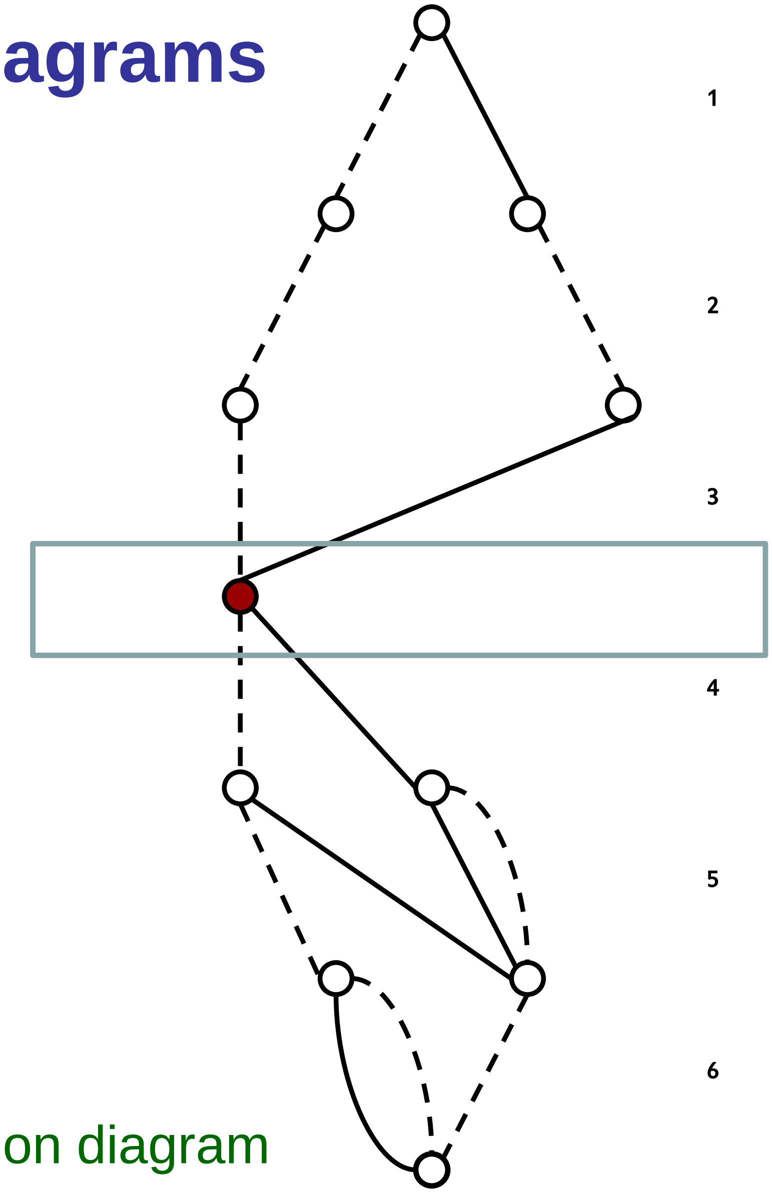


Relaxed decision diagram

Decision Diagrams

Branching in a relaxed
decision diagram

First branch

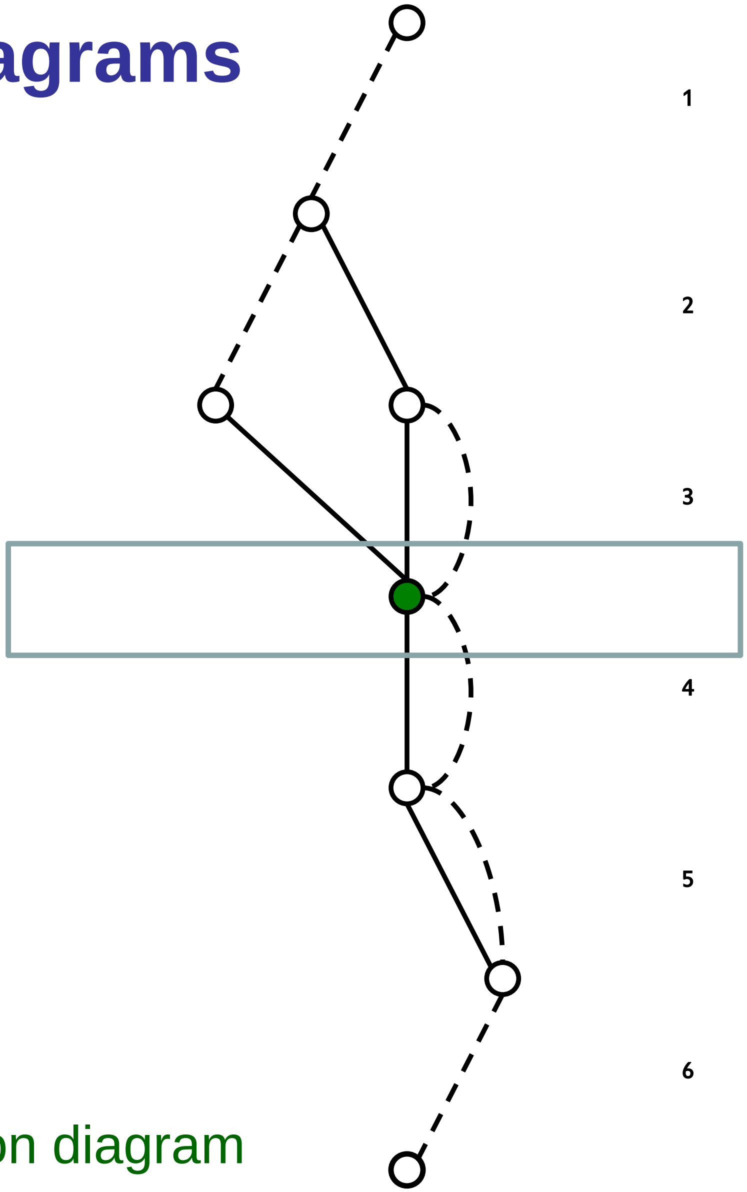


Relaxed decision diagram

Decision Diagrams

Branching in a relaxed decision diagram

Second branch



Relaxed decision diagram

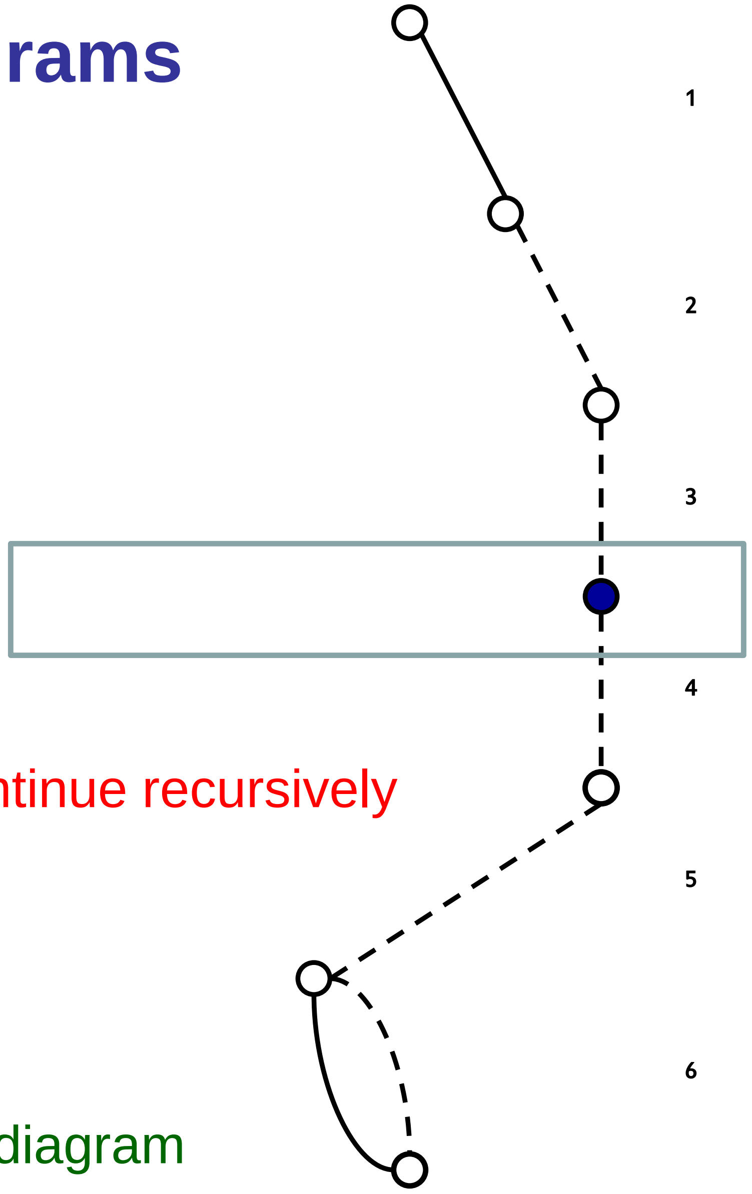
Decision Diagrams

Branching in a relaxed decision diagram

Third branch

Continue recursively

Relaxed decision diagram



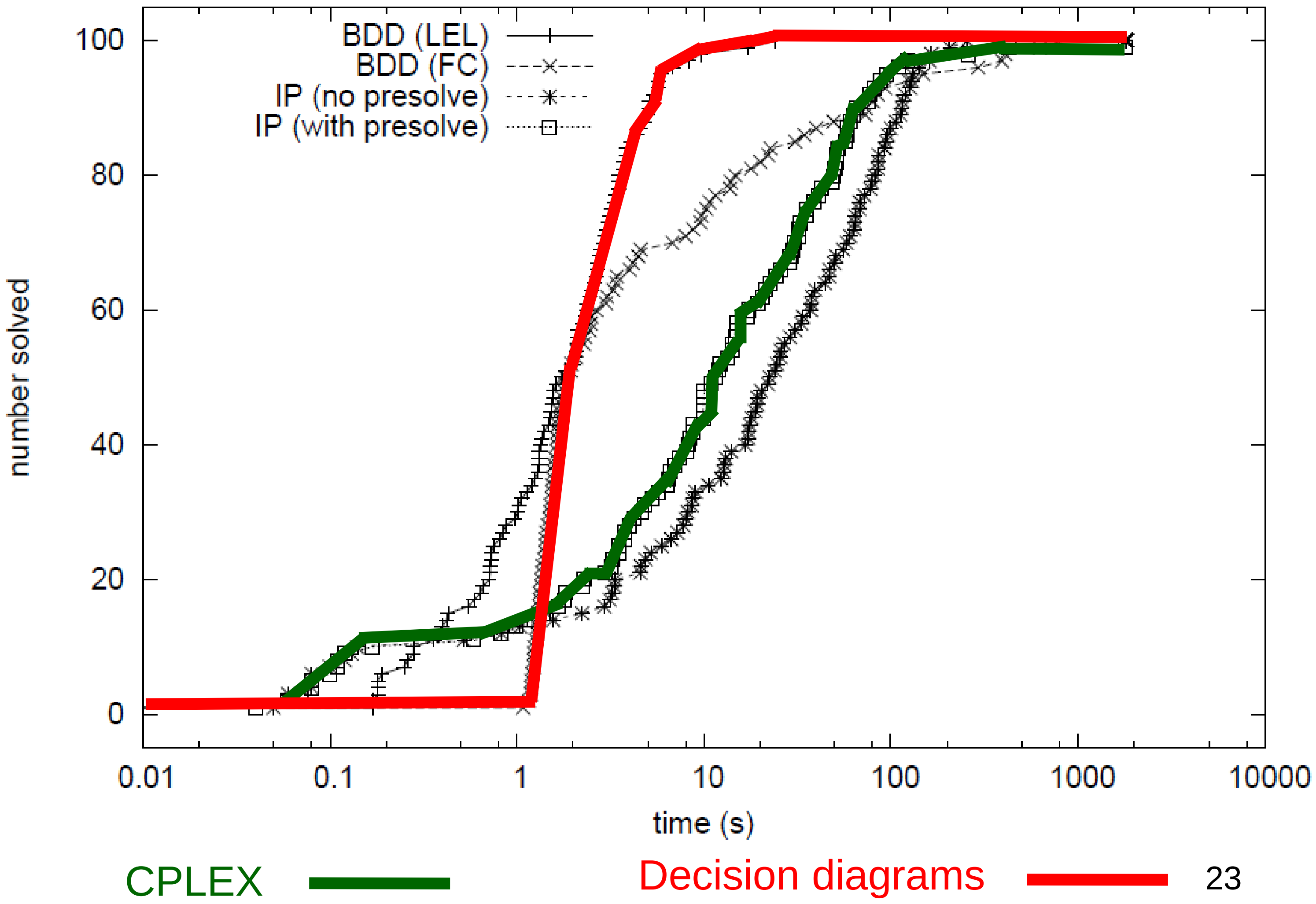
Decision Diagrams

- Computational results…
  - Superior to state-of-the-art mixed integer solver on most instances tested.
    - Even when the problem has a natural MIP model.
    - Stable set, max cut, max 2-SAT.
    - Obtained best known solution on several max cut instances.

Decision Diagrams

Performance profile

Random max 2-SAT instances
30 variables,
100 instances



Decision Diagrams

- Potential to scale up
  - No need to load large inequality model into solver.
  - Parallelizes very effectively
    - Near-linear speedup.
    - Much better than mixed integer programming.

Decision Diagrams

- Research direction:
  - Decision diagram technology is ready for **large-scale** applications from industry
    - …which will also help develop the technology.

Decision Diagrams

- Research direction:
  - Decision diagram technology is ready for **large-scale** applications from industry
    - …which will also help develop the technology.
  - Give problems a **dynamic programming** model.
    - …as required by decision diagrams.
    - But **don't** solve problems by dynamic programming.
    - Solve them using **branch and bound algorithm**.
  - New approach to the "curse of dimensionality."
    - Solve DP models that were **previously intractable** due to exponential state space.

Decision Diagrams

- Research direction:
  - Extend decision diagrams to **stochastic dynamic programming** and **optimal control**.
    - Introduce relaxed **stochastic** decision diagrams.
    - Develop an alternative to approximate dynamic programming.
  - Specialize to…
    - **partially observable state spaces**.
    - **adaptive control** ("data driven")
  - Apply to large-scale problems.

Decision Diagrams

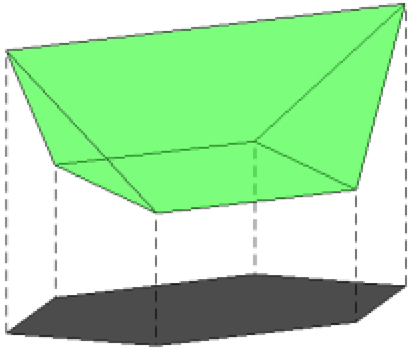
- Research direction:
  - Extend to **continuous global optimization**.
    - Use massive discretization of continuous variables.
  - Use **data driven optimization**.
    - Use sampling techniques as in “big data” analysis.
    - Develop alternatives to statistics and Lipschitz bounds for certainty guarantees.

Decision Diagrams

- Research direction:
  - Perform in-depth **postoptimality analysis**.
    - Based on the fact that a decision diagram is a logic model.
    - **Deduce information** and **answer queries** about optimal and near-optimal solutions.
    - Can construct decision diagram based on optimal value obtained from another solver.

Logic-Based Benders

- **Logic-based Benders decomposition** is a generalization of classical Benders decomposition.
  - Subproblem need not have linear or inequality model.
    - JH (1995), JH and Yan (1995), JH and Ottosson (2003).
- Solves a **projection** problem.
  - ...and therefore a **logical inference** problem.



Logic-Based Benders

- Collaborators…
  - André Ciré
  - Elvin Çoban
  - Greger Ottosson
  - Erlendur Thorsteinsson*
  - Hong Yan

***2001** *Best Paper Award*, Constraint Programming conference

Logic-Based Benders

- Fundamental concept: **inference duality**
  - **Optimization** and **logical inference** are two sides of the same coin.

Primal problem: optimization

$$\min f(x)$$
$$x \in S$$

Find **best** feasible solution…
by searching over **values of *x***.

Dual problem: Inference

$$\max v$$
$$x \in S \overset{P}{\Rightarrow} f(x) \geq v$$
$$P \in \mathcal{P}$$

Find tightest bound that can be **inferred** from constraints…
by searching over **proofs *P***

Logic-Based Benders

- Popular optimization duals are special cases of the inference dual.
  - Result from different choices of inference method.
  - For example....
    - Linear programming dual
    - Lagrangean dual
    - Surrogate dual
    - Subadditive dual

Logic-Based Benders

- Decompose problem into master and subproblem.
  - Subproblem is obtained by fixing some variables x to solution values in master problem.

Master problem

$\min z$

Benders cuts

$x \in D$

Accumulated Benders cuts describe **projection** of feasible set onto x.

→ Trial value $\bar{x}$ that solves master →

← Benders cut $z \geq g(x)$ ←

Subproblem

$\min f(\bar{x}, y)$

$(\bar{x}, y) \in S$

Analysis of proof that solves inference dual yields a Benders cut

Logic-Based Benders

- Example: machine assignment and scheduling
  - Combines **mixed integer programming** and **constraint programming**

Master problem

Assign jobs to machines.

Solve problem by **mixed integer programming**.

Trial assignment $\bar{x}$

Benders cut $z \geq g(x)$

Subproblem

Schedule jobs on each machine.

Constraint programming obtains proof of optimality (dual solution).

Use same proof to deduce cost for other assignments, yielding Benders cut.

Logic-Based Benders

- Substantial speedup for many applications.
  - Several orders of magnitude relative to state of the art.

Logic-Based Benders

- Substantial speedup for many applications.
  - Several orders of magnitude relative to state of the art.
- Some applications:
  - Circuit verification
  - Chemical batch processing (BASF, etc.)
  - Steel production scheduling
  - Auto assembly line management (Peugeot-Citroën)
  - Automated guided vehicles in flexible manufacturing
  - Allocation and scheduling of multicore processors (IBM, Toshiba, Sony)
  - Facility location-allocation
  - Stochastic facility location and fleet management
  - Capacity and distance-constrained plant location

Logic-Based Benders

- Some applications…
  - Transportation network design
  - Traffic diversion around blocked routes
  - Worker assignment in a queuing environment
  - Single- and multiple-machine allocation and scheduling
  - Permutation flow shop scheduling with time lags
  - Resource-constrained scheduling
  - Wireless local area network design
  - Service restoration in a network
  - Optimal control of dynamical systems
  - Sports scheduling

Logic-Based Benders

- Research direction
  - Apply to **large-scale** problems from **industry**.
    - By incorporating subproblem relaxation into master problem.

Logic-Based Benders

- Research direction
  - Apply to **large-scale** problems from **industry**.
    - By incorporating subproblem relaxation into master problem.
- Research direction (already begun)
  - Apply to data-driven **robust optimization**.
    - …with uncertainty sets.
    - Uncertainty subproblem becomes Benders subproblem.
    - Now applying to IBM service center scheduling.

Logic-Based Benders

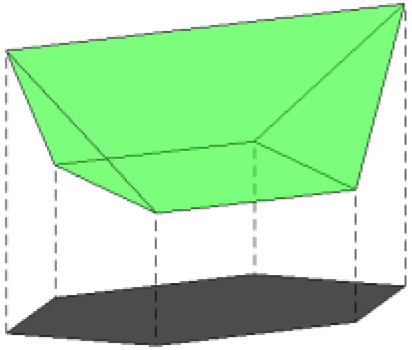
- Research direction (already begun)
  - Use **decision diagram** in master problem.
    - Add Benders cuts by applying **separation algorithm** to decision diagram
  - Now applying to home health care scheduling, etc.

Logic-Based Benders

- Research direction (already begun)
  - Use **decision diagram** in master problem.
    - Add Benders cuts by applying **separation algorithm** to decision diagram
  - Now applying to home health care scheduling, etc.
- Research direction
  - Apply to **logical inference** in large datasets…
    - Original application was to logic circuit verification.

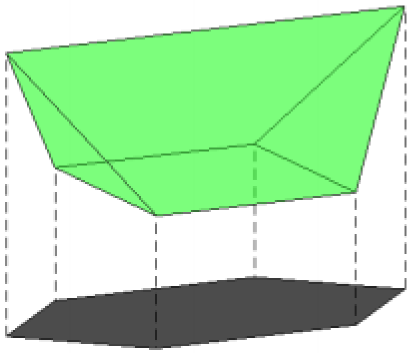
Logical Inference

- Connection between optimization and logic.
  - Both are **projection** problems.
  - Projection = extract information involving a subset of variables.



Logical Inference

- Connection between optimization and logic.
  - Both are **projection** problems.
  - Projection = extract information involving a subset of variables.
- Optimization…
  - **Project** feasible set onto a single variable representing the objective function.
- Inference…
  - **Project** knowledge base onto a subset of propositional variables.



Logical Inference

- Example:
  - Knowledge base consists of logical clauses.
  - Derive all implications involving x_1, x_2.
  - This is a projection problem.

$$x_1 \vee x_2$$
$$\neg x_1 \vee x_3$$
$$\neg x_1 \vee \neg x_2 \vee \neg x_3$$
$$x_1 \vee x_3 \vee x_4$$
$$x_2 \vee x_3 \vee \neg x_4$$

$$x_1 \vee x_2$$
$$\neg x_1 \vee \neg x_2$$

Projection

Logical Inference

- Example:
  - Knowledge base consists of logical clauses.
  - Derive all implications involving x_1, x_2.
  - This is a projection problem.
- A classical projection method is resolution.
  - *Aside:* resolvents are a special case of Chvátal-Gomory cutting planes!
    - A basic optimization tool.

$$
\begin{aligned}
& x_1 \vee x_2 \\
& \neg x_1 \vee x_3 \\
& \neg x_1 \vee \neg x_2 \vee \neg x_3 \\
& x_1 \vee x_3 \vee x_4 \\
& x_2 \vee x_3 \vee \neg x_4
\end{aligned}
$$

$$
\begin{aligned}
& x_1 \vee x_2 \\
& \neg x_1 \vee \neg x_2
\end{aligned}
$$

Projection

Logical Inference

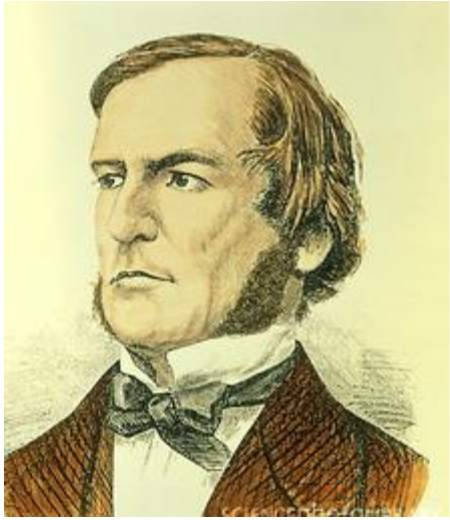
- A central problem today:
  - Associate each inference with its **probability**, **relevance**, or **confidence**.
    - For example, IBM's **Watson** (Jeopardy player).

Logical Inference

- A central problem today:
  - Associate each inference with its **probability**, **relevance**, or **confidence**.
    - For example, IBM's **Watson** (Jeopardy player).
- The problem was first posed for **probability**...

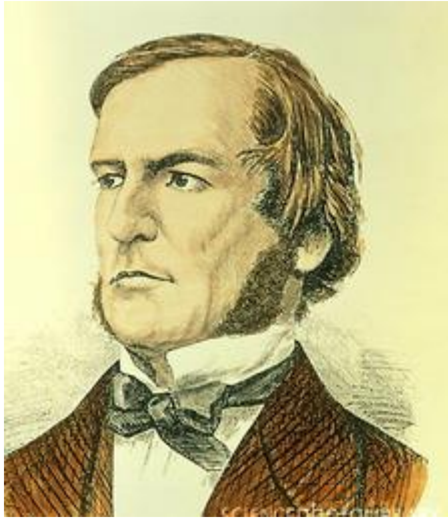
Logical Inference

- A central problem today:
  - Associate each inference with its **probability**, **relevance**, or **confidence**.
    - For example, IBM's **Watson** (Jeopardy player).
- The problem was first posed for **probability**...
  - …by George Boole.
    - He viewed **probability logic** as his most important contribution.



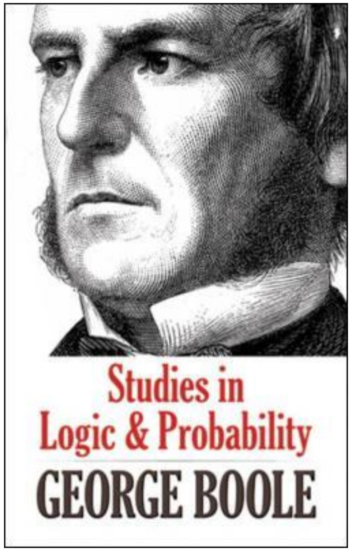
Logical Inference

- A central problem today:
  - Associate each inference with its **probability**, **relevance**, or **confidence**.
    - For example, IBM's **Watson** (Jeopardy player).
- The problem was first posed for **probability**...
  - …by George Boole.
    - He viewed **probability logic** as his most important contribution.
  - Boole also was the first to connect **optimization** with **logical inference**…



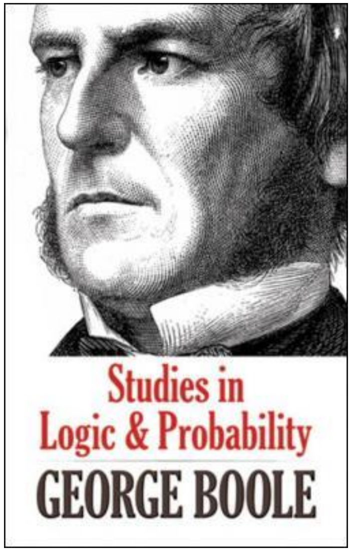
Logical Inference

- Boole used **optimization** for inference in probability logic.
  - He formulated it as a **linear programming** problem.
    - Perhaps the first application of linear programming, today the most widely used optimization model.



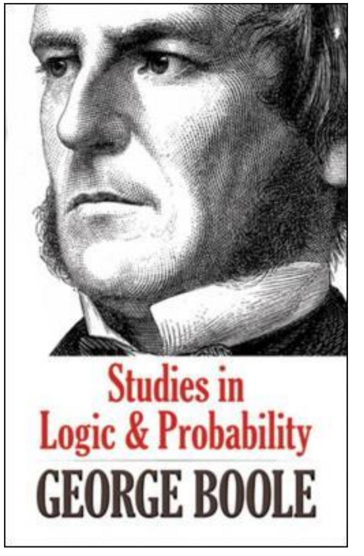
Logical Inference

- Boole used **optimization** for inference in probability logic.
  - He formulated it as a **linear programming** problem.
    - Perhaps the first application of linear programming, today the most widely used optimization model.
  - He proposed a **solution method**.
    - We now call it **Fourier-Motzkin elimination**.
    - …the best-known **projection** method for polyhedra!!



Logical Inference

- Boole used **optimization** for inference in probability logic.
  - He formulated it as a **linear programming** problem.
    - Perhaps the first application of linear programming, today the most widely used optimization model.
  - He proposed a **solution method**.
    - We now call it **Fourier-Motzkin elimination**.
    - …the best-known **projection** method for polyhedra!!
  - Boole not only connected optimization with logical inference, but connected both with **projection**.



Logical Inference

- Research direction
  - Apply **large-scale optimization** methods to logical inference.
    - They are substantially **underutilized** for this purpose.
    - 1990s research in this area is newly relevant.
  - In particular:
    - Use **logic-based Benders** for propositional logic, etc.
    - Use **column generation** for probability logic.
    - Use **linear programming** for belief logics, relevance, and variations of Dempster-Shafer theory.
    - Use **decision diagrams** for queries and what-if analysis.