# Duality in Optimization and Constraint Satisfaction

J. N. Hooker

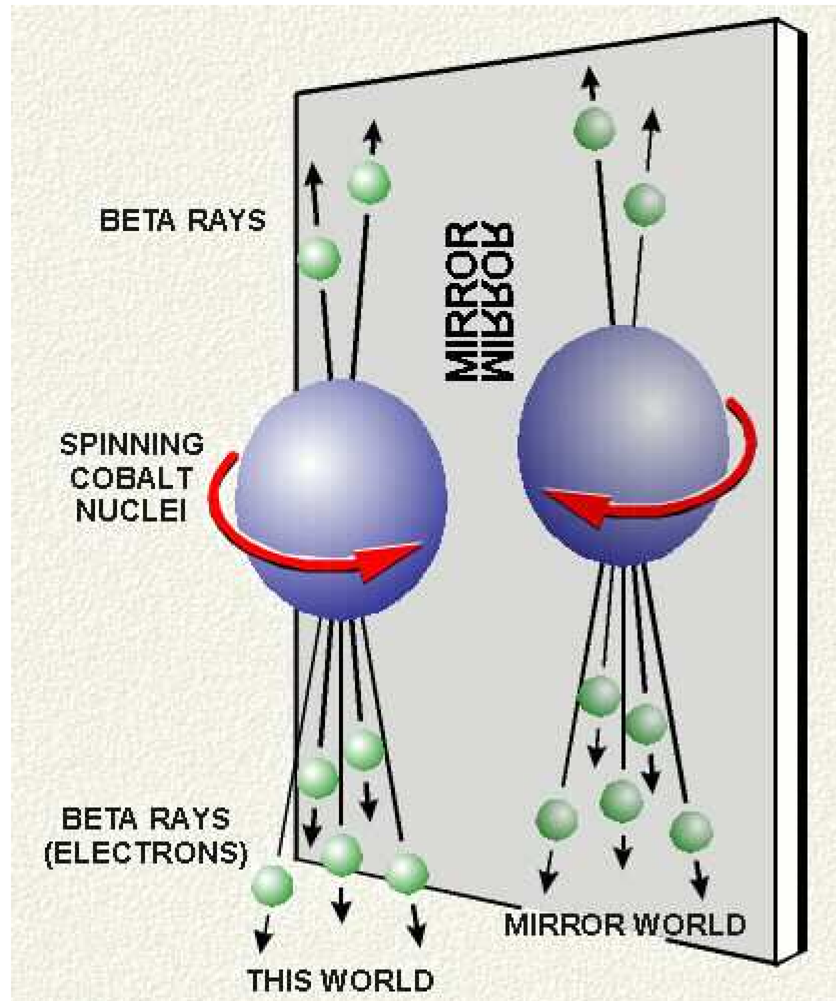
Carnegie Mellon Univ.

Pittsburgh, USA

La Sapienza, October 2006

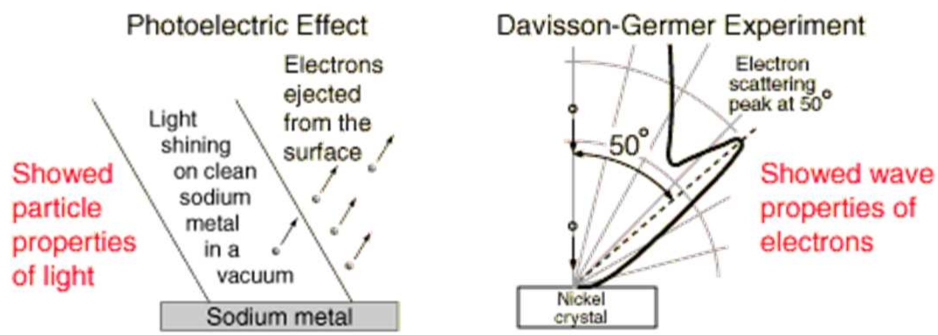
# Some Concepts of Duality

- Reflexive dual (mirror image)
  - Conservation of parity
    - Refuted by C. S. Wu, 1957



# Some Concepts of Duality

- Dual perspectives on one reality
  - Wave/particle duality



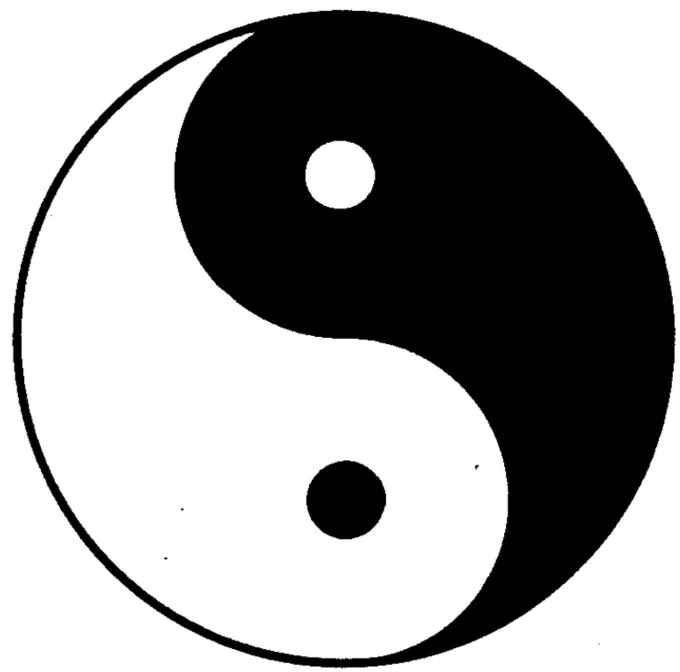
# Some Concepts of Duality

- Dual perspectives on one reality
  – Wave/particle duality
  – Mind/body duality
    - Spinoza, Brahman/Atman

Lalita Mahatripurasundari
*representing Brahman-Atman consciousness*

# Some Concepts of Duality

- Complementarity
  - Yin/yang



# Some Concepts of Duality

- Complementarity
  - Yin/yang
  - Good/evil in Zoroastrianism

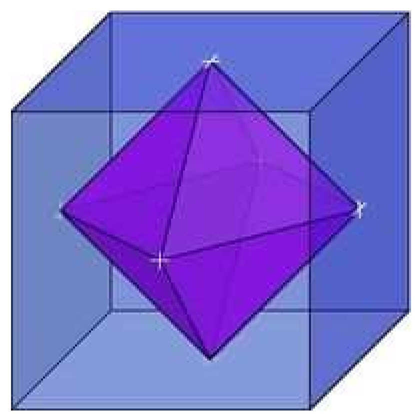
Aharamazda, Zoroastrian god

# Duality in Mathematics

- Most mathematical duals are reflexive
  - Dual of dual = original

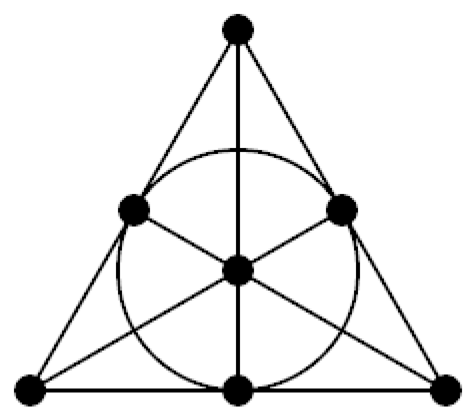
# Duality in Mathematics

- Duality of polytopes
  - $k$-dimensional faces become $(n - k - 1)$-dimensional faces



# Duality in Mathematics

- Duality of points and lines in projective geometry



7-point projective plane
(Fano plane)

# Duality in Mathematics

- Duality of rows and columns in matrix algebra.
  - Row rank of a matrix = column rank.

# Duality in Mathematics

- Duality of rows and columns in matrix algebra.
  - Row rank of a matrix = column rank.

- Generalization: dual space of a vector space.
  - Space of linear functionals on a space has same dimension as original space.
  - Dual of dual is a “natural transformation” in category theory.

# Duality in Optimization

- Most optimization duals offer dual perspectives.
  - Not reflexive in general.

# Duality in Optimization

- Reflexive:
  - Linear programming dual

# Duality in Optimization

- Reflexive:
  - Linear programming dual
- Not reflexive:
  - Surrogate dual
  - Lagrangean dual
  - Superadditive dual

# Duality in Optimization

- Reflexive:
  - Linear programming dual
- Not reflexive:
  - Surrogate dual
  - Lagrangean dual
  - Superadditive dual
- All of these are instances of **inference duality** and **relaxation duality**.

# Duality in Optimization

- **Inference duality** and **relaxation duality** are very different concepts.
  - Even though several optimization duals can be viewed as either.

# Duality in Optimization

- **Inference duality** and **relaxation duality** are very different concepts.
  - Even though several optimization duals can be viewed as either.
- Inference duality provides **sensitivity analysis** and **nogood-based search**.
  - Will focus on nogood-based search here.

# Duality in Optimization

- **Inference duality** and **relaxation duality** are very different concepts.
  - Even though several optimization duals can be viewed as either.
- Inference duality provides **sensitivity analysis** and **nogood-based search**.
  - Will focus on nogood-based search here.
- Relaxation duality provides **tight bounds**.

# Duality in Optimization

- **Inference duality** and **relaxation duality** are very different concepts.
  - Even though several optimization duals can be viewed as either.
- Inference duality provides **sensitivity analysis** and **nogood-based search**.
  - Will focus on nogood-based search here.
- Relaxation duality provides **tight bounds**.
- The **constraint dual** is neither type but gives rise to a relaxation dual.

# Duality in Optimization

- Inference duality is a duality of **search** and **inference**.
  - The original problem can be solved by **searching** over values of variables to find the best solution.
  - The dual is solved by **inferring** from the constraints the best bound on the optimal value.

# Duality in Optimization

- Inference duality is a duality of **search** and **inference**.
  - The original problem can be solved by **searching** over values of variables to find the best solution.
  - The dual is solved by **inferring** from the constraints the best bound on the optimal value.

- Relaxation duality is a duality of **restriction** and **relaxation**.
  - The original problem can be solved by enumerating **restrictions**.
  - The dual can be solved by enumerating parameterized **relaxations**.

# Duality in Optimization

- Viewing duals in this way has three benefits.

# Duality in Optimization

- Viewing duals in this way has three benefits.
- It reveals **connections** that might not otherwise be noticed.
  - For instance, surrogate and Lagrangean duals are seen to be closely related when viewed as inference duals rather than as relaxation duals.

# Duality in Optimization

- Viewing duals in this way has three benefits.
- It reveals **connections** that might not otherwise be noticed.
  - For instance, surrogate and Lagrangean duals are seen to be closely related when viewed as inference duals rather than as relaxation duals.
- It can unify **solution methods**.
  - For instance, Benders decomposition and DPL with clause learning are nogood-based search methods that result from different inference duals.

# Duality in Optimization

- It can suggest **new solution methods**.
  - For instance, any domain filter defines an inference dual with an associated nogood-based search methods.
    - Edge-finding gives rise to an effective Benders-like decomposition method for planning & scheduling, etc.
  - A relaxation dual can be defined on a constraint dual formulation so as to generalize mini-bucket elimination.

# Duality in Optimization

- Outline…

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Inference Dual

- An **optimization problem** minimizes an objective function subject to constraints.
  - It is solved by searching over **values of the variables**.

- The **inference dual** finds the tightest lower bound on the objective function that can be deduced from the constraints.
  - It is solved by searching over **proofs**.

# Inference Dual

- The optimization problem

Constraint set

$$\min_{x \in D} \left\{ f(x) \;\middle|\; C \right\}$$

"Primal" problem

has the **inference dual**

$$\max_{P \in \mathcal{P}} \left\{ v \;\middle|\; C \xrightarrow[P]{} \left( f(x) \geq v \right) \right\}$$

Family of proofs

Proof $P$ deduces $f(x) \geq v$ from $C$

# Inference Dual

- **Weak duality** always holds:

Max value of dual problem $\leq$ Min value of primal problem

Difference = **duality gap**

$$\max_{P \in \mathcal{P}} \left\{ v \;\middle|\; C \xrightarrow[P]{} \big( f(x) \geq v \big) \right\}$$

# Inference Dual

- **Strong duality** sometimes holds:

Max value of dual problem = Min value of primal problem

$\mathcal{P}$ is a **complete** proof family $\Rightarrow$ Strong duality

$$\max_{P \in \mathcal{P}} \left\{ v \mid C \xrightarrow[P]{} \left( f(x) \geq v \right) \right\}$$

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Linear Programming: Inference Dual

- A **linear programming** problem has the form

$$\min_{x \geq 0} \left\{ cx \mid Ax \geq b \right\}$$

- The inference dual is

$$\max_{P \in \mathcal{P}} \left\{ v \mid Ax \geq b \xrightarrow[P]{} (cx \geq v) \right\}$$

  – The proof family $\mathcal{P}$ consists of **domination by surrogates** (nonnegative linear combinations).

# Linear Programming: Inference Dual

- For any $u \geq 0$,
  $uAx \geq ub$ is a **surrogate** of $Ax \geq b$.

- $uAx \geq ub$ **dominates** $cx \geq v$
  if $uAx \geq ub$ implies $cx \geq v$ (for $x \geq 0$).
  - That is, $uA \leq c$ and $ub \geq v$.

- This is a complete inference method.
  - Due to **Farkas Lemma**.

# Linear Programming: Inference Dual

- So the inference dual

$$\max_{P \in \mathcal{P}} \left\{ v \mid Ax \geq b \xrightarrow[P]{} (cx \geq v) \right\}$$

becomes

$$\max_{u \geq 0} \left\{ v \mid uA \leq c, ub \geq v \right\}$$

when $Ax \geq b$, $x \geq 0$ is feasible

or

$$\max_{u \geq 0} \left\{ ub \mid uA \leq c \right\}$$

  - This is the classical linear programming dual.

# Linear Programming: Inference Dual

- Since the proof family is complete, we have strong duality:

$$\max_{u \geq 0} \left\{ ub \mid uA \leq c \right\} = \min_{x \geq 0} \left\{ cx \mid Ax \geq b \right\}$$

except when $uA \leq c$, $u \geq 0$ and $Ax \geq b$, $x \geq 0$ are both infeasible.

# Linear Programming: Inference Dual

- Since the proof family is complete, we have strong duality:

$$\max_{u\geq 0}\left\{ub \mid uA \leq c\right\} = \min_{x\geq 0}\left\{cx \mid Ax \geq b\right\}$$

except when $uA \leq c$, $u \geq 0$ and $Ax \geq b$, $x \geq 0$ are both infeasible.

  – In this case, the dual is symmetric
    - The dual of the dual is the primal.
  – The dual therefore belongs to NP and the primal to co-NP.

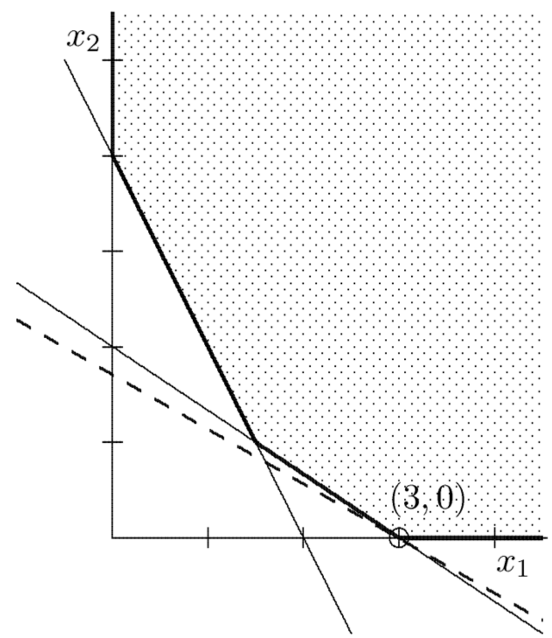
# Linear Programming: Inference Dual

- Example:

$$\min 4x_1 + 7x_2$$
$$2x_1 + 3x_2 \ge 6 \quad (u_1)$$
$$2x_1 + x_2 \ge 4 \quad (u_2)$$
$$x_1, x_2 \ge 0$$

Dual:

$$\max 6u_1 + 4u_2$$
$$2u_1 + 2u_2 \le 4 \quad (x_1)$$
$$3u_1 + u_2 \le 7 \quad (x_2)$$
$$u_1, u_2 \ge 0$$



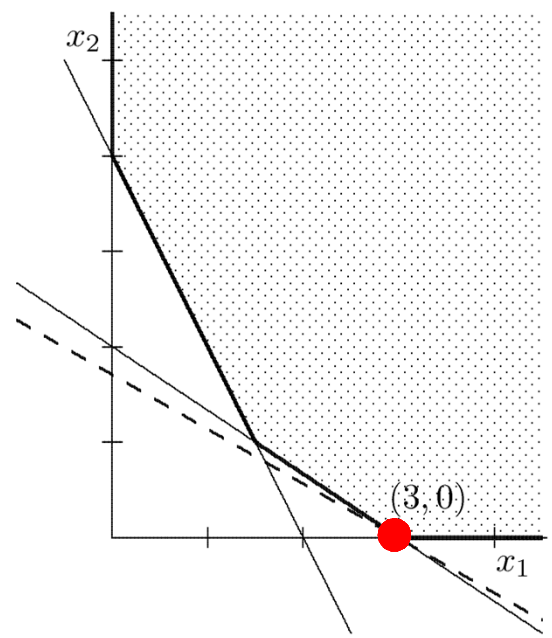
# Linear Programming: Inference Dual

- Example:

$$\min 4x_1 + 7x_2 \quad = 12$$
$$2x_1 + 3x_2 \geq 6 \quad (u_1 = 2)$$
$$2x_1 + x_2 \geq 4 \quad (u_2 = 0)$$
$$x_1, x_2 \geq 0$$

Dual:

$$\max 6u_1 + 4u_2 \quad = 12$$
$$2u_1 + 2u_2 \leq 4 \quad (x_1 = 3)$$
$$3u_1 + u_2 \leq 7 \quad (x_2 = 0)$$
$$u_1, u_2 \geq 0$$




– Solution:

$$2x_1 + 3x_2 \geq 6 \quad (2)$$
$$2x_1 + x_2 \geq 4 \quad (0)$$

Surrogate → $4x_1 + 6x_2 \geq 12$

Dominates → $\Downarrow$

$$4x_1 + 7x_2 \geq 12$$

# Relaxation Dual

- A **relaxation dual** parameterizes relaxations of the optimization problem.
  - The relaxation parameters are **dual variables**.
  - Each relaxation provides a **lower bound** on the optimal value of the primal.

- It seeks the relaxation that provides the **tightest bound**.
  - It is solved by searching over **values of the dual variables**.

# Relaxation Dual

- Given the optimization problem

$$\min_{x\in D}\left\{ f(x) \mid C \right\}$$

a **parameterized relaxation** is

$$\min_{x\in D}\left\{ f(x,u) \mid C(u) \right\}$$

$u$ = vector of **dual variables**

Lower bound on $f(x)$ for all $x \in D$ satisfying $C$.

Relaxation of $C$.

# Relaxation Dual

- Given the optimization problem

$$\min_{x \in D} \left\{ f(x) \mid C \right\}$$

a **parameterized relaxation** is

$$\min_{x \in D} \left\{ f(x,u) \mid C(u) \right\}$$

  – It provides a lower bound on the optimal value of the original problem, for any *u*.

# Relaxation Dual

- Given the optimization problem

$$\min_{x\in D}\left\{ f(x) \mid C \right\}$$

a **relaxation dual** is

$$\max_{u\in U}\left\{ \min_{x\in D}\left\{ f(x,u) \mid C(u) \right\} \right\}$$

Find relaxation that provides the largest lower bound.

# Relaxation Dual

- **Weak duality** always holds:

Max value of dual problem ≤ Min value of primal problem

Difference = **duality gap**

$$\max_{u \in U} \left\{ \min_{x \in D} \left\{ f(x,u) \mid C(u) \right\} \right\}$$

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Linear Programming: Relaxation Dual

- A **surrogate relaxation** of

$$\min_{x \geq 0} \left\{ cx \mid Ax \geq b \right\}$$

  replaces $Ax \geq b$ with a surrogate $uAx \geq ub$.
  - The relaxation is parameterized by $u$.
  - The objective function is left unchanged.

# Linear Programming: Relaxation Dual

- This yields the relaxation dual

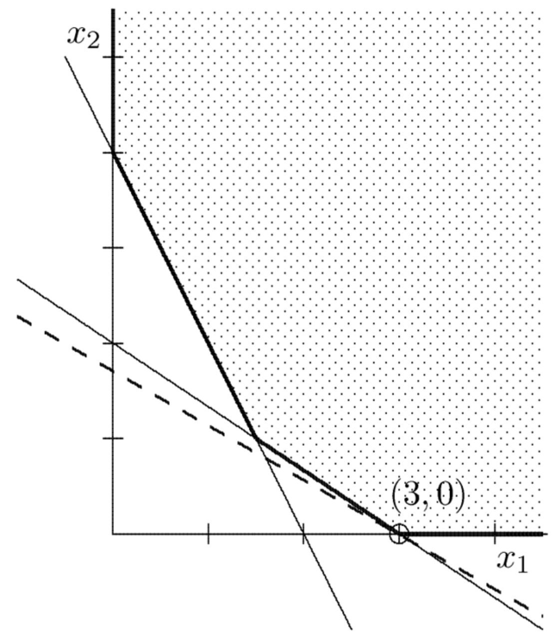
$$\max_{u\geq 0}\left\{ \min_{x\geq 0}\left\{ cx \mid uAx \geq ub \right\}\right\}$$

  - It is equivalent to the classical linear programming dual.
  - So the classical dual is both an inference and a relaxation dual.

# Linear Programming: Relaxation Dual

- Example:

$$\begin{aligned} &\min 4x_1 + 7x_2 \\ &2x_1 + 3x_2 \geq 6 \quad (u_1) \\ &2x_1 + x_2 \geq 4 \quad (u_2) \\ &x_1, x_2 \geq 0 \end{aligned}$$



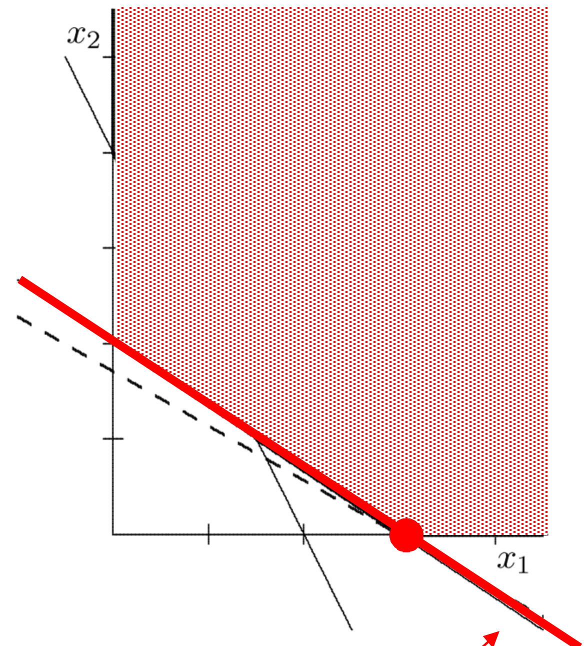

- Relaxation dual:

$$\max_{u_1,u_2 \geq 0} \left\{ \min_{x_1,x_2 \geq 0} \left\{ 4x_1 + 7x_2 \;\middle|\; (2u_1 + 2u_2)x_1 + (3u_1 + u_2)x_2 \geq 6u_1 + 4u_2 \right\} \right\}$$

# Linear Programming: Relaxation Dual

- Example:

$$\begin{aligned} &\min 4x_1 + 7x_2 \\ &2x_1 + 3x_2 \geq 6 \quad (u_1) \\ &2x_1 + x_2 \geq 4 \quad (u_2) \\ &x_1, x_2 \geq 0 \end{aligned}$$




- Relaxation dual:

$$\max_{u_1,u_2 \geq 0} \left\{ \min_{x_1,x_2 \geq 0} \left\{ 4x_1 + 7x_2 \mid (2u_1 + 2u_2)x_1 + (3u_1 + u_2)x_2 \geq 6u_1 + 4u_2 \right\} \right\}$$

- Solution: Let $(u_1, u_2) = (2, 0)$

$$\min_{x_1,x_2 \geq 0} \left\{ 4x_1 + 7x_2 \mid 4x_1 + 6x_2 \geq 12 \right\} = 12$$

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Surrogate Dual

- The **surrogate dual** is a relaxation dual of a general inequality-constrained problem:

$$\min_{x \in D} \left\{ f(x) \mid g(x) \le 0 \right\}$$

- The parameterized relaxation is

$$\min_{x \in D} \left\{ f(x) \mid ug(x) \le 0 \right\}$$

- So the dual is

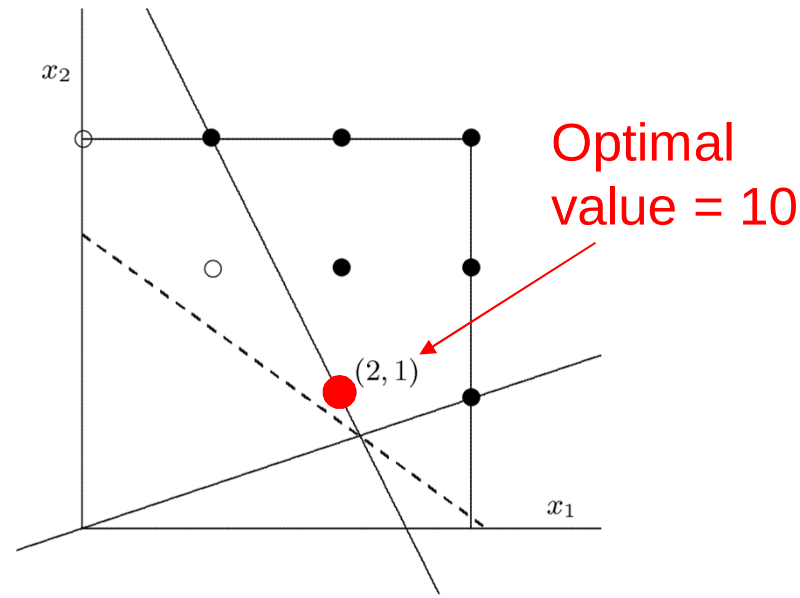
$$\max_{u \ge 0} \left\{ \min_{x \in D} \left\{ f(x) \mid ug(x) \le 0 \right\} \right\}$$

  – In general there is a duality gap.

# Surrogate Dual

- Example: $\min 3x_1 + 4x_2$

$$x_1 - 3x_2 \le 0$$
$$5 - 2x_1 - x_2 \le 0$$
$$x_1, x_2 \in \{0,1,2,3\}$$



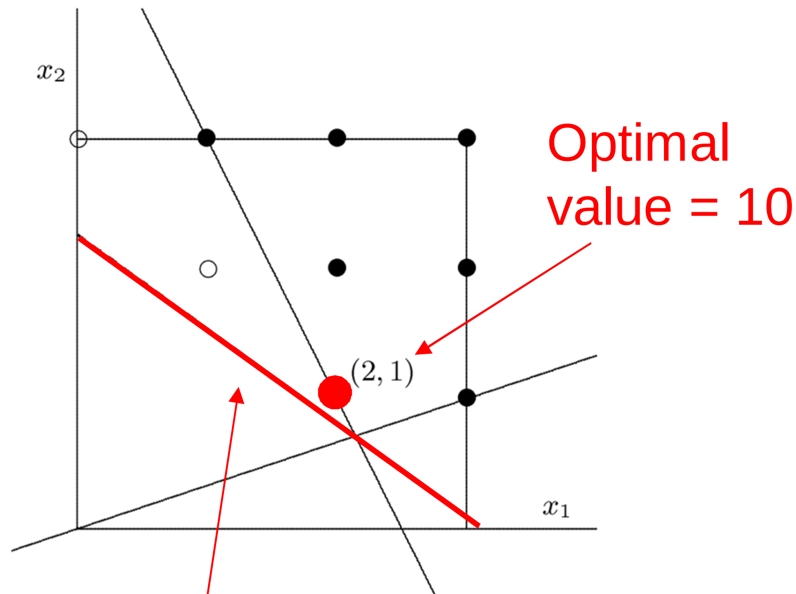

- The surrogate dual is

$$\max_{u \ge 0} \left\{ \min_{x_j \in \{0,1,2,3\}} \left\{ 3x_1 + 4x_2 \mid u_1(x_1 - 3x_2) + u_2(5 - 2x_1 - x_2) \le 0 \right\} \right\}$$

# Surrogate Dual

- Example: $\min 3x_1 + 4x_2$

$$x_1 - 3x_2 \le 0$$
$$5 - 2x_1 - x_2 \le 0$$
$$x_1, x_2 \in \{0,1,2,3\}$$




- The surrogate dual is

$$\max_{u \ge 0} \left\{ \min_{x_j \in \{0,1,2,3\}} \left\{ 3x_1 + 4x_2 \mid u_1(x_1 - 3x_2) + u_2(5 - 2x_1 - x_2) \le 0 \right\} \right\}$$

- Solution: $(u_1, u_2) = (2,5)$, which yields
  the surrogate $8x_1 + 11x_2 \ge 25$
  and bound 10 (no duality gap in this case)

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Surrogate Dual

- The surrogate dual is also an **inference dual**.
- It uses the same proof system as the linear programming dual.
- Deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$.
  - i.e., when $ug(x) \leq 0$ **implies** $f(x) \geq v$.
  - An incomplete proof system.

# Surrogate Dual

- This yields the **inference dual**

$$\max_{u \geq 0} \left\{ v \,\middle|\, ug(x) \leq 0 \rightarrow f(x) \geq v \text{ for } x \in D \right\}$$

  – which is equivalent to the relaxation dual

$$\max_{u \geq 0} \left\{ \min_{x \in D} \left\{ f(x) \,\middle|\, ug(x) \leq 0 \right\} \right\}$$

# Aside: Subadditive Dual

- Suppose multiplication by nonnegative vector $u$ is replaced by a general **subadditive function** $u(\,)$.
  - $u(a + b) \leq u(a) + u(b)$
- The surrogate dual becomes the **subadditive dual**.
  - Repeated linear combinations and rounding give rise to a family of subadditive functions for integer programming.
    - They provide a complete inference method.
    - There is no duality gap.

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Lagrangean Dual

- The **Lagrangean dual** is another relaxation dual for inequality-constrained problems

$$\min_{x \in D} \left\{ f(x) \,\middle|\, g(x) \le 0 \right\}$$

# Lagrangean Dual

- The **Lagrangean dual** is another relaxation dual for inequality-constrained problems

$$\min_{x \in D} \left\{ f(x) \;\middle|\; g(x) \le 0 \right\}$$

- This time, remove the constraints entirely but "penalize" constraint violations in the objective function:

$$\max_{u \ge 0} \left\{ \min_{x \in D} \left\{ f(x) + ug(x) \right\} \right\}$$

# Lagrangean Dual

- The **Lagrangean dual** is another relaxation dual for inequality-constrained problems

$$\min_{x \in D} \left\{ f(x) \;\middle|\; g(x) \le 0 \right\}$$

- This time, remove the constraints entirely but "penalize" constraint violations in the objective function:

$$\max_{u \ge 0} \left\{ \min_{x \in D} \left\{ f(x) + u g(x) \right\} \right\}$$

  - This is a relaxation since

$$f(x) + \underbrace{u}_{\ge 0}\, \underbrace{g(x)}_{\le 0} \le f(x) \text{ for all feasible } x$$

# Lagrangean Dual

- The Lagrangean dual provides a weaker bound than than the surrogate dual.
  – But it can be solved by steepest ascent search.

- Write the dual

$$\max_{u \geq 0} \left\{ \min_{x \in D} \{ f(x) + ug(x) \} \right\}$$

as

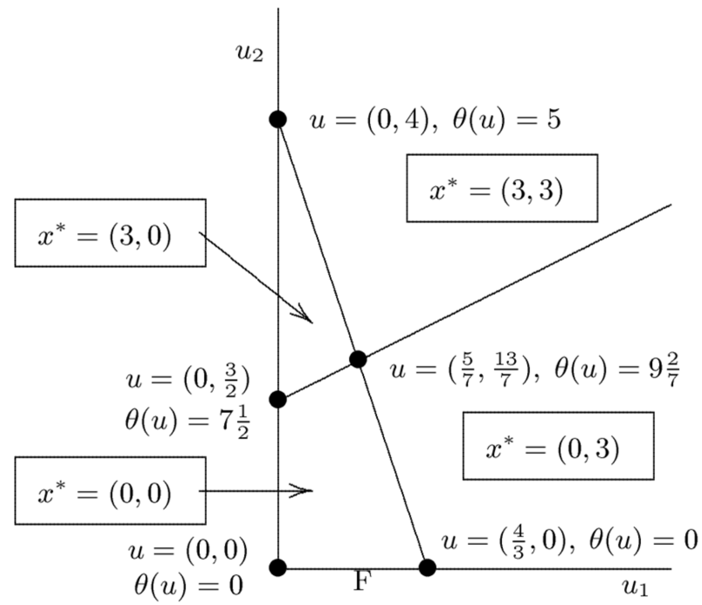
$$\max_{u \geq 0} \{\theta(u)\} \text{ where } \theta(u) = \min_{x \in D} \{f(x) + ug(x)\}$$

  – Then $\theta(u)$ is **concave,** and its subgradient is $g(x)$.

# Lagrangean Dual

- Example: $\min 3x_1 + 4x_2$

$$x_1 - 3x_2 \le 0$$
$$5 - 2x_1 - x_2 \le 0$$
$$x_1, x_2 \in \{0,1,2,3\}$$




- The dual is

$$\max_{u_1,u_2 \ge 0} \{\theta(u_1,u_2)\}$$

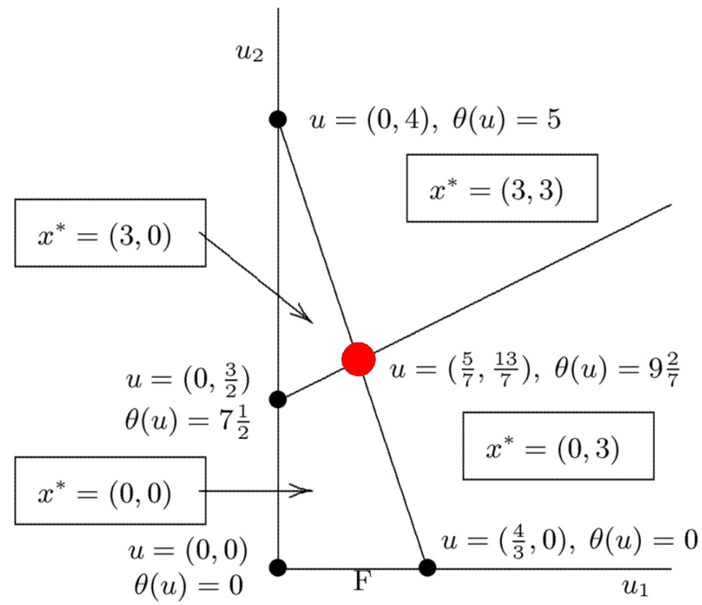
where

$$\theta(u_1,u_2) = \min_{x_j \in \{0,1,2,3\}} \{ 3x_1 + 4x_2 + u_1(x_1 - 3x_2) + u_2(5 - 2x_1 - x_2) \}$$

# Lagrangean Dual

- Example: $\min 3x_1 + 4x_2$

$$x_1 - 3x_2 \le 0$$
$$5 - 2x_1 - x_2 \le 0$$
$$x_1, x_2 \in \{0,1,2,3\}$$




- The dual is $\max_{u_1,u_2 \ge 0} \{\theta(u_1,u_2)\}$

  where

$$\theta(u_1,u_2) = \min_{x_j \in \{0,1,2,3\}} \{ 3x_1 + 4x_2 + u_1(x_1 - 3x_2) + u_2(5 - 2x_1 - x_2) \}$$

- Solution: $(u_1,u_2) = (\frac{5}{7}, \frac{13}{7})$ with optimal value $9\frac{2}{7}$

  Duality gap of $10 - 9\frac{2}{7}$

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
| --- | --- | --- | --- |
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Lagrangean Dual

- The Lagrangean and surrogate duals appear unrelated.
  - But when viewed as inference duals, they are closely related.
- The Lagrangean dual uses a slightly weaker proof system.
  - It follows that it yields a weaker bound.

# Lagrangean Dual

- Deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$.
  - i.e., some surrogate $ug(x) \leq 0$ dominates $v - f(x) \leq 0$.

# Lagrangean Dual

- Deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$.
  - i.e., some surrogate $ug(x) \leq 0$ dominates $v - f(x) \leq 0$.
- In the surrogate dual, "dominates" means

$$\boldsymbol{ug(x) \leq 0 \text{ implies } v - f(x) \leq 0}.$$

# Lagrangean Dual

- Deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$.
  - i.e., some surrogate $ug(x) \leq 0$ dominates $v - f(x) \leq 0$.
- In the surrogate dual, "dominates" means

$$\boldsymbol{ug(x) \leq 0 \ \ \text{implies} \ \ v - f(x) \leq 0}.$$

- In the Lagrangean dual, it means something stronger:

$$\boldsymbol{ug(x) \geq v - f(x)}.$$

  - i.e., $v \leq f(x) + ug(x)$

# Lagrangean Dual

- Deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$.
  - i.e., some surrogate $ug(x) \leq 0$ dominates $v - f(x) \leq 0$.
- In the surrogate dual, "dominates" means

$$\boldsymbol{ug(x) \leq 0 \text{ implies } v - f(x) \leq 0}.$$

- In the Lagrangean dual, it means something stronger:

$$\boldsymbol{ug(x) \geq v - f(x)}.$$

  - i.e., $v \leq f(x) + ug(x)$
- So the resulting inference dual is the Lagrangean dual:

$$\max_{u \geq 0} \left\{ v \;\middle|\; v \leq f(x) + ug(x) \text{ for all } x \in D \right\}$$

$$= \max_{u \geq 0} \left\{ \min_{x \in D} \left\{ f(x) + ug(x) \right\} \right\}$$

# Nogood-Based Search

- The inference dual provides the basis for **nogood-based search**.
  - Nogoods rule out solutions already examined.
    - …and possibly other solutions that are no better.

# Nogood-Based Search

- The inference dual provides the basis for **nogood-based search**.
  - Nogoods rule out solutions already examined.
    - …and possibly other solutions that are no better.
  - Search proceeds by enumerating partial assignments to the variables.
  - The next partial assignment must satisfy nogoods generated so far.

# Nogood-Based Search

- The inference dual provides the basis for **nogood-based search**.
  - Nogoods rule out solutions already examined.
    - …and possibly other solutions that are no better.
  - Search proceeds by enumerating partial assignments to the variables.
  - The next partial assignment must satisfy nogoods generated so far.
- The nogoods are obtained by **analyzing the inference dual** of the restricted problem that results from each partial assignment.

# Nogood-Based Search

- Solve: $\min_{x \in D} \left\{ f(x) \mid C \right\}$

- In each iteration, formulate a **restriction** of the problem:

$$\min_{x \in D} \left\{ f(x) \mid C \cup \mathcal{B} \right\}$$

Partial assignment that fixes some of the variables $x_j$

# Nogood-Based Search

- Solve: $\min_{x \in D} \{ f(x) \mid C \}$
- In each iteration, formulate a **restriction** of the problem:

$$\min_{x \in D} \{ f(x) \mid C \cup \mathcal{B} \}$$

  - $\mathcal{B}$ may contain:
    - Branching constraints (in branching methods)
    - Solution of master problem (in Benders decomposition)

# Nogood-Based Search

- Let $(\bar{v}, \bar{P})$ solve the inference dual of the restriction:

$$\max_{p \in \mathcal{P}} \left\{ f(x) \,\middle|\, C \cup \mathcal{B} \xrightarrow[P]{} \big(f(x) \geq v\big) \right\}$$

  - Let $\bar{\mathcal{B}}$ fix the variables in $\mathcal{B}$ that serve as premises in $\bar{P}$

# Nogood-Based Search

- Let $(\bar{v}, \bar{P})$ solve the inference dual of the restriction:

$$\max_{p \in \mathcal{P}} \left\{ f(x) \,\middle|\, C \cup \mathcal{B} \xrightarrow[P]{} \left( f(x) \geq v \right) \right\}$$

  - Let $\bar{\mathcal{B}}$ fix the variables in $\mathcal{B}$ that are needed as premises in $\bar{P}$
  - Create the **nogood** $\bar{\mathcal{B}} \rightarrow \left( f(x) \geq \bar{v} \right)$

# Nogood-Based Search

- Let $(\bar{v}, \bar{P})$ solve the inference dual of the restriction:

$$\max_{p \in \mathcal{P}} \left\{ f(x) \,\middle|\, C \cup \mathcal{B} \xrightarrow[P]{} \left( f(x) \geq v \right) \right\}$$

  - Let $\bar{\mathcal{B}}$ fix the variables in $\mathcal{B}$ that serve as premises in $\bar{P}$
  - Create the **nogood** $\bar{\mathcal{B}} \rightarrow \left( f(x) \geq \bar{v} \right)$

- The next partial assignment $\mathcal{B}$ must be consistent with all nogoods generated so far.
  - This avoids partial assignments already examined.
    - …and others that are no better.

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Linear Programming: Nogood-Based Search

- Solve $$\min_{\substack{x_F \in D_F \\ x_U \geq 0}} \left\{ f(x_F) + cx_U \;\middle|\; g(x_F) + Ax_U \geq b \right\}$$

Problem becomes linear subproblem when $x_F$ is fixed to $\bar{x}_F$.

# Linear Programming: Nogood-Based Search

- Solve $$\min_{\substack{x_F \in D_F \\ x_U \geq 0}} \left\{ f(x_F) + c x_U \;\middle|\; g(x_F) + A x_U \geq b \right\}$$

Problem becomes linear subproblem when $x_F$ is fixed to $\bar{x}_F$.

- Nogood-based search becomes **Benders decomposition**.
  - The nogoods are formed by solving the linear programming dual of the problem that results when $x_F$ is fixed.

# Linear Programming: Nogood-Based Search

- Solve $\min\limits_{\substack{x_F \in D_F \\ x_U \geq 0}} \left\{ f(x_F) + cx_U \;\middle|\; g(x_F) + Ax_U \geq b \right\}$

- Let $\bar{u}$ solve LP dual of the subproblem:

$$\max_{u \geq 0} \left\{ u(b - g(\bar{x}_F)) \;\middle|\; uA \leq c \right\}$$

# Linear Programming: Nogood-Based Search

- Solve $$\min_{\substack{x_F \in D_F \\ x_U \geq 0}} \left\{ f(x_F) + cx_U \;\middle|\; g(x_F) + Ax_U \geq b \right\}$$

- Let $\bar{u}$ solve LP dual of the subproblem:

$$\max_{u \geq 0} \left\{ u\left(b - g(\bar{x}_F)\right) \;\middle|\; uA \leq c \right\}$$

  – Then by strong duality ($\bar{z}$ = optimal value):

$$\bar{z} - f(\bar{x}_F) = \bar{u}\left(b - g(\bar{x}_F)\right)$$

# Linear Programming: Nogood-Based Search

- Solve $$\min_{\substack{x_F \in D_F \\ x_U \geq 0}} \left\{ f(x_F) + cx_U \;\middle|\; g(x_F) + Ax_U \geq b \right\}$$

- Let $\bar{u}$ solve LP dual of the subproblem:

$$\max_{u \geq 0} \left\{ u\left(b - g(\bar{x}_F)\right) \;\middle|\; uA \leq c \right\}$$

  – Then by strong duality ($\bar{z}$ = optimal value):

$$\bar{z} - f(\bar{x}_F) = \bar{u}\left(b - g(\bar{x}_F)\right)$$

  – And since $\bar{u}_F$ is dual feasible for any $x_F$, weak duality implies the Benders cut

$$z \geq f(x_F) + \bar{u}\left(b - g(x_F)\right)$$

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# SAT: Nogood-Based Search

- Consider the propositional satisfiability problem (SAT), which can be written

$$\min_{x_j \in \{T,F\}} \left\{ 0 \;\middle|\; C \right\}$$

Set of logical clauses

- Branching search (DPL) with clause learning can be viewed as a nogood-based search.
  - Unit resolution (unit clause rule) is applied at every node of the search tree.
  - The nogoods are **conflict clauses**…

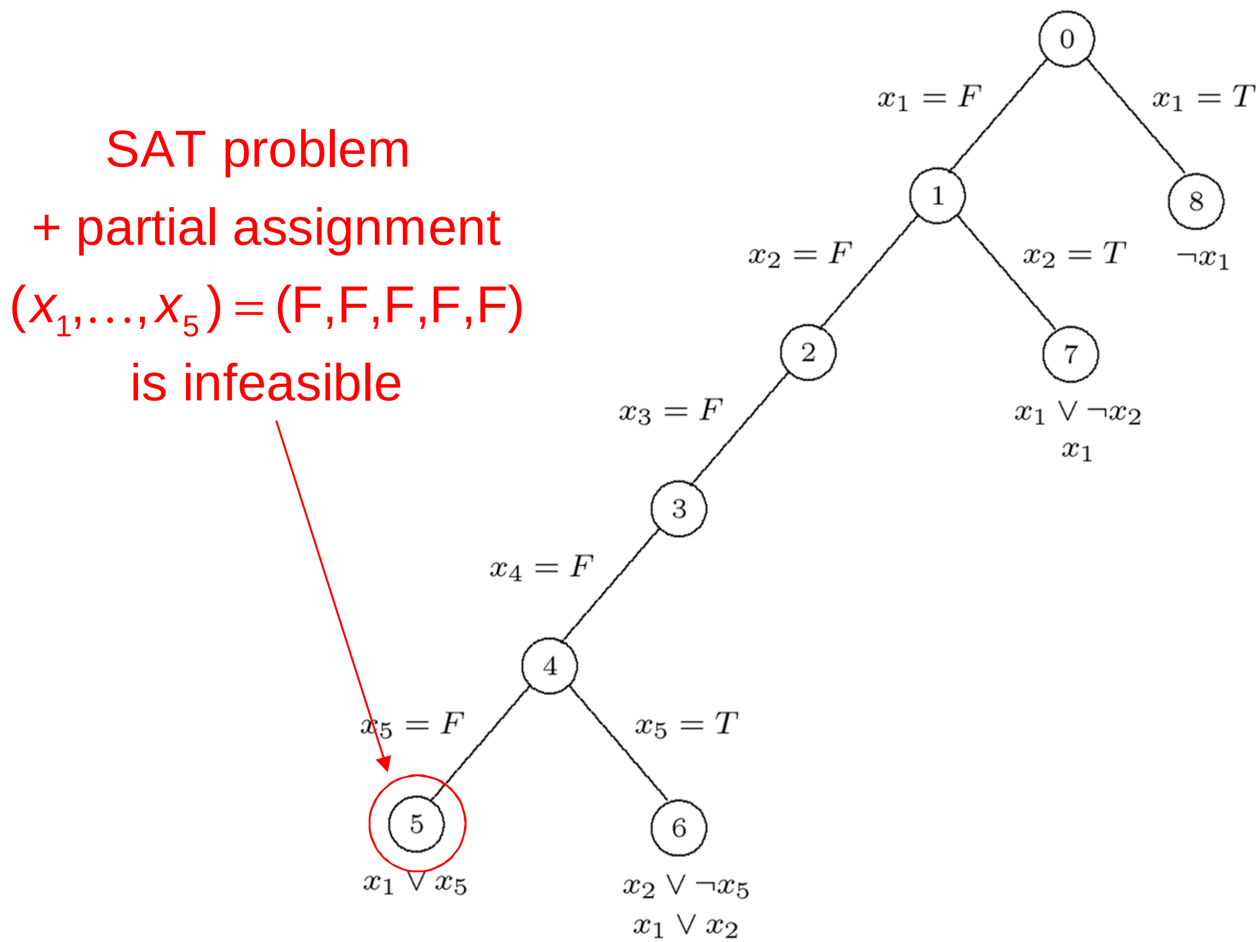
# SAT: Nogood-Based Search

- Example:

$$\begin{array}{llllll}
x_1 & & & & \vee \ x_5 & \vee \ x_6 \\
x_1 & & & & \vee \ x_5 & \vee \neg x_6 \\
 & x_2 & & & \vee \neg x_5 & \vee \ x_6 \\
 & x_2 & & & \vee \neg x_5 & \vee \neg x_6 \\
\neg x_1 & & \vee \ x_3 & \vee \ x_4 & & \\
 & \neg x_2 & \vee \ x_3 & \vee \ x_4 & & \\
\neg x_1 & & \vee \neg x_3 & & & \\
\neg x_1 & & & \vee \neg x_4 & & \\
 & \neg x_2 & \vee \neg x_3 & & & \\
 & \neg x_2 & & \vee \neg x_4 & &
\end{array}$$

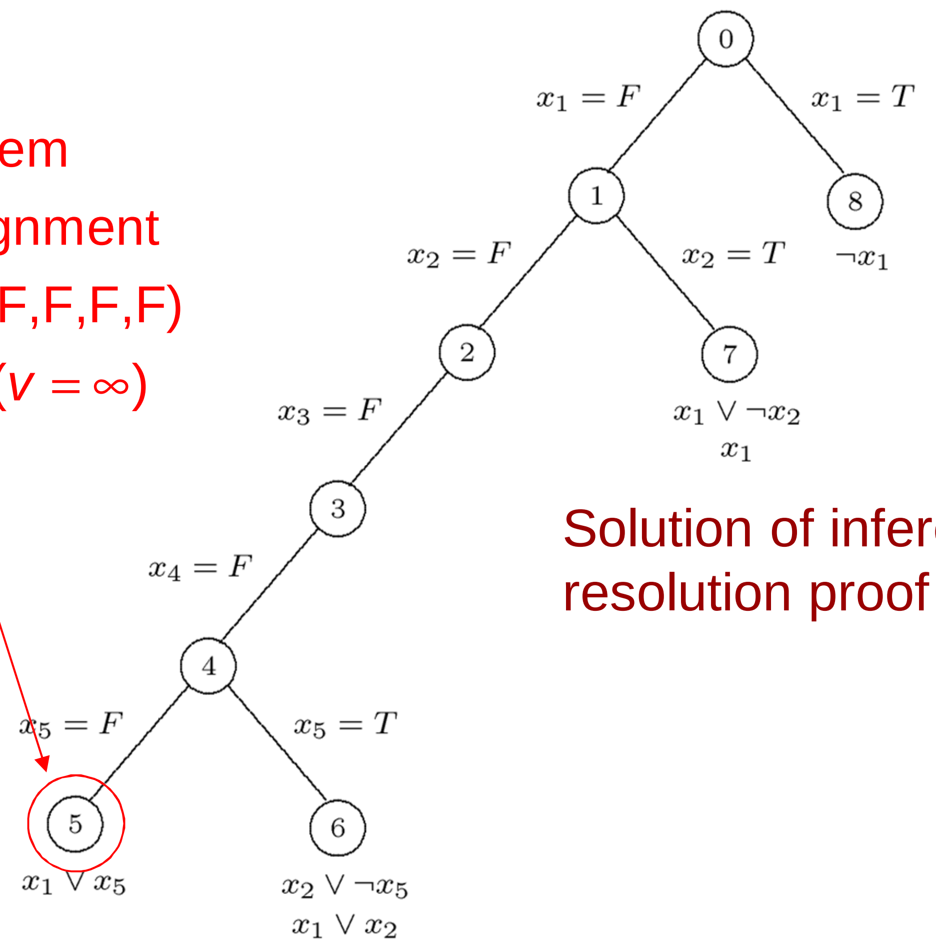
  – Solve by branching on $x_j$'s…

# SAT: Nogood-Based Search



# SAT: Nogood-Based Search

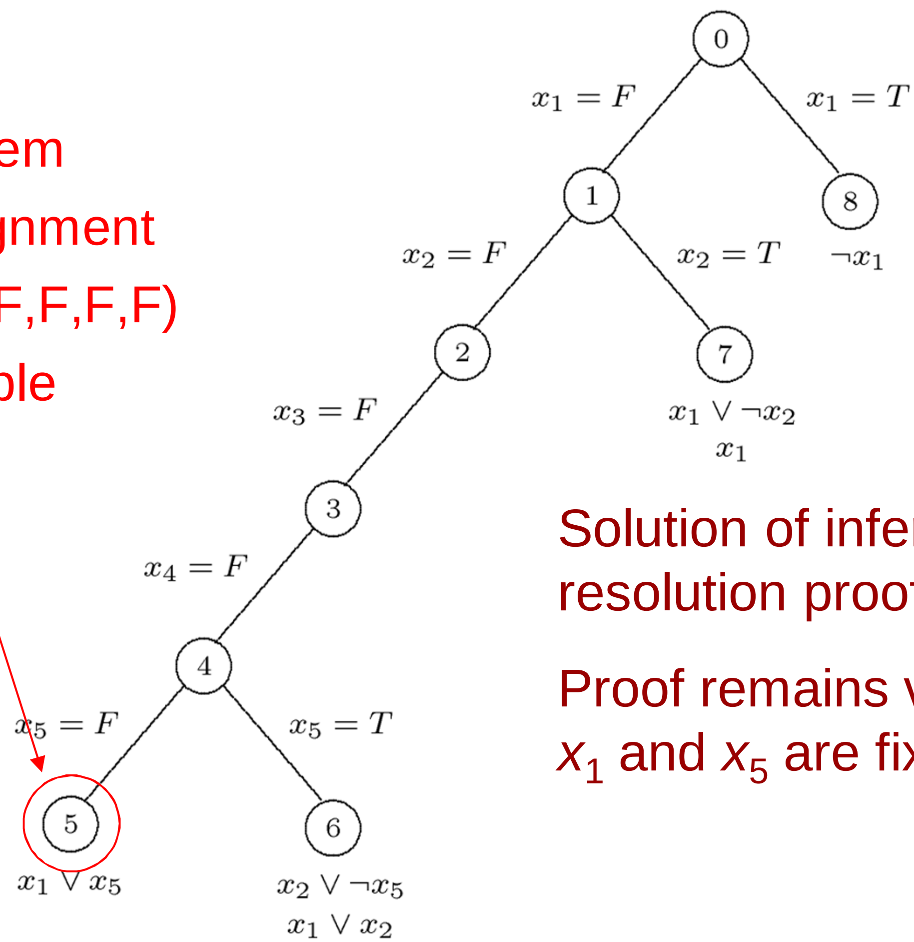
SAT problem
+ partial assignment
$(x_1,\ldots,x_5) = (F,F,F,F,F)$
is infeasible ($v = \infty$)




Solution of inference dual is a unit resolution proof of infeasibility.

# SAT: Nogood-Based Search

SAT problem
+ partial assignment
$(x_1,\ldots,x_5) = (F,F,F,F,F)$
is infeasible

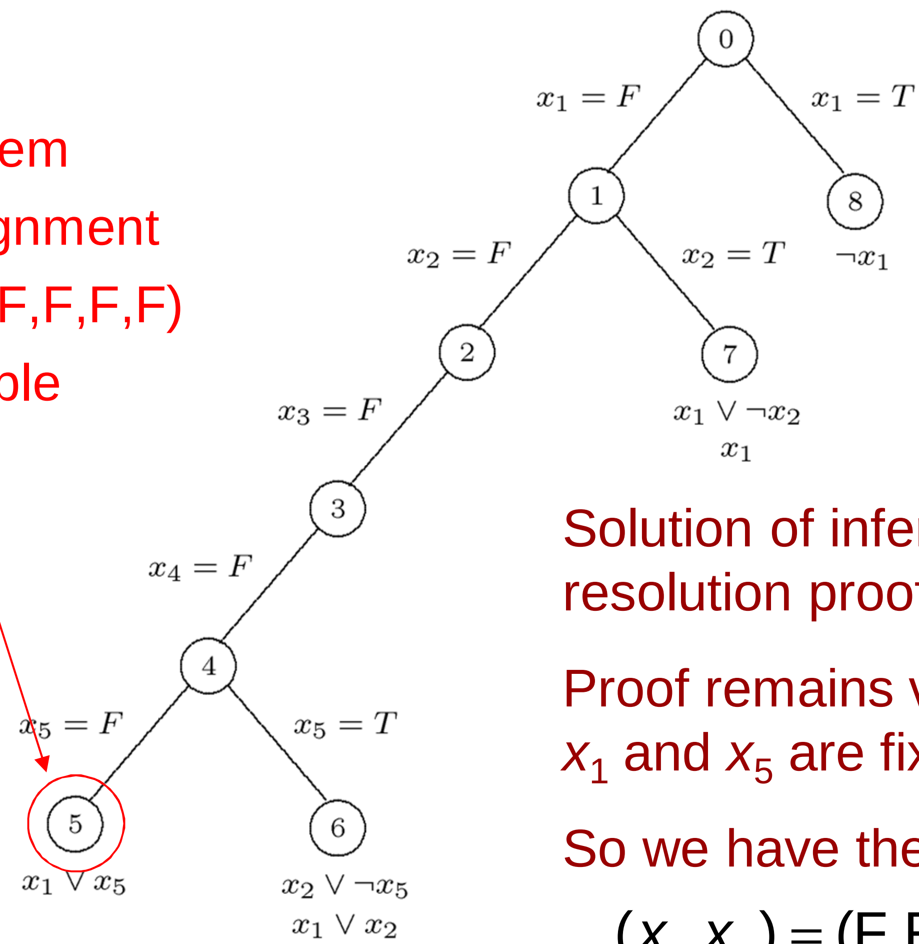



Solution of inference dual is a unit resolution proof of infeasiblity.

Proof remains valid when only $x_1$ and $x_5$ are fixed to F.

# SAT: Nogood-Based Search

SAT problem
+ partial assignment
$(x_1,\ldots,x_5) = (F,F,F,F,F)$
is infeasible




Solution of inference dual is a unit resolution proof of infeasiblity.

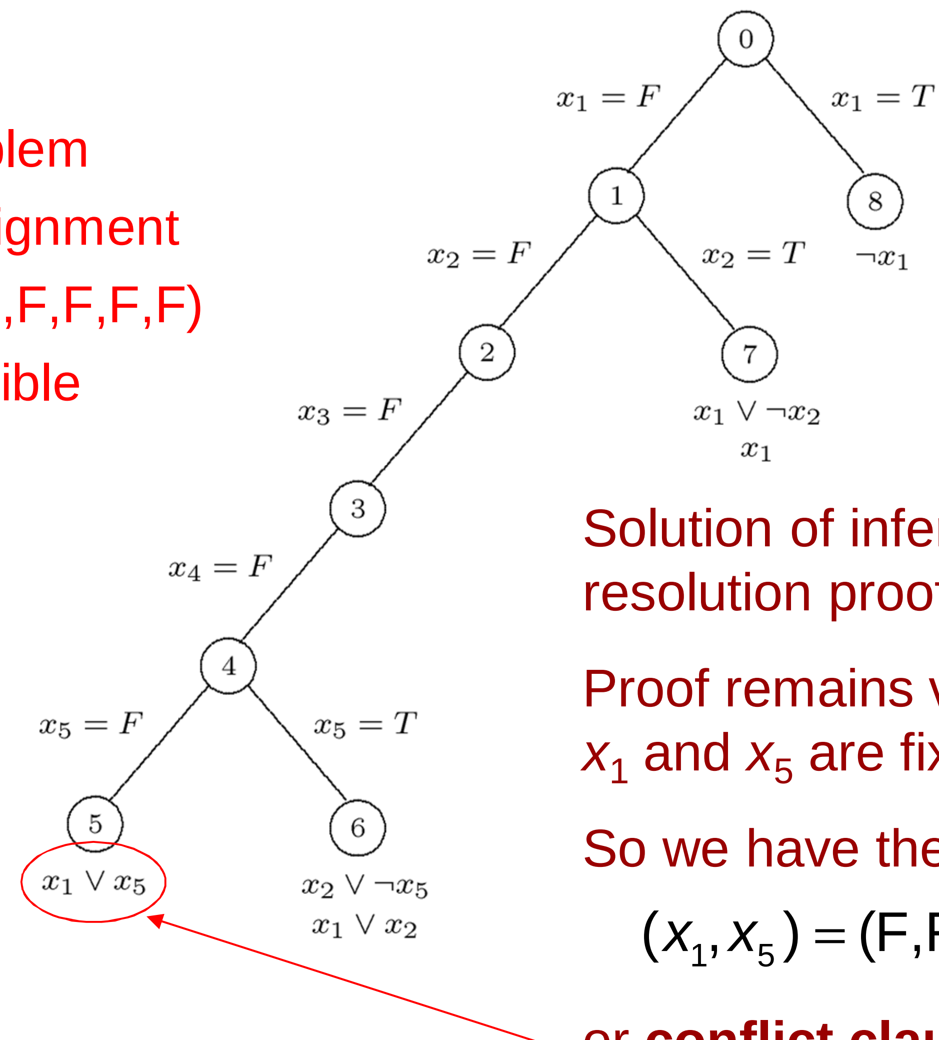
Proof remains valid when only $x_1$ and $x_5$ are fixed to F.

So we have the nogood

$$(x_1, x_5) = (F,F) \rightarrow f(x) \geq \infty$$

# SAT: Nogood-Based Search

SAT problem
+ partial assignment
$(x_1,\ldots,x_5) = (F,F,F,F,F)$
is infeasible




Solution of inference dual is a unit resolution proof of infeasiblity.

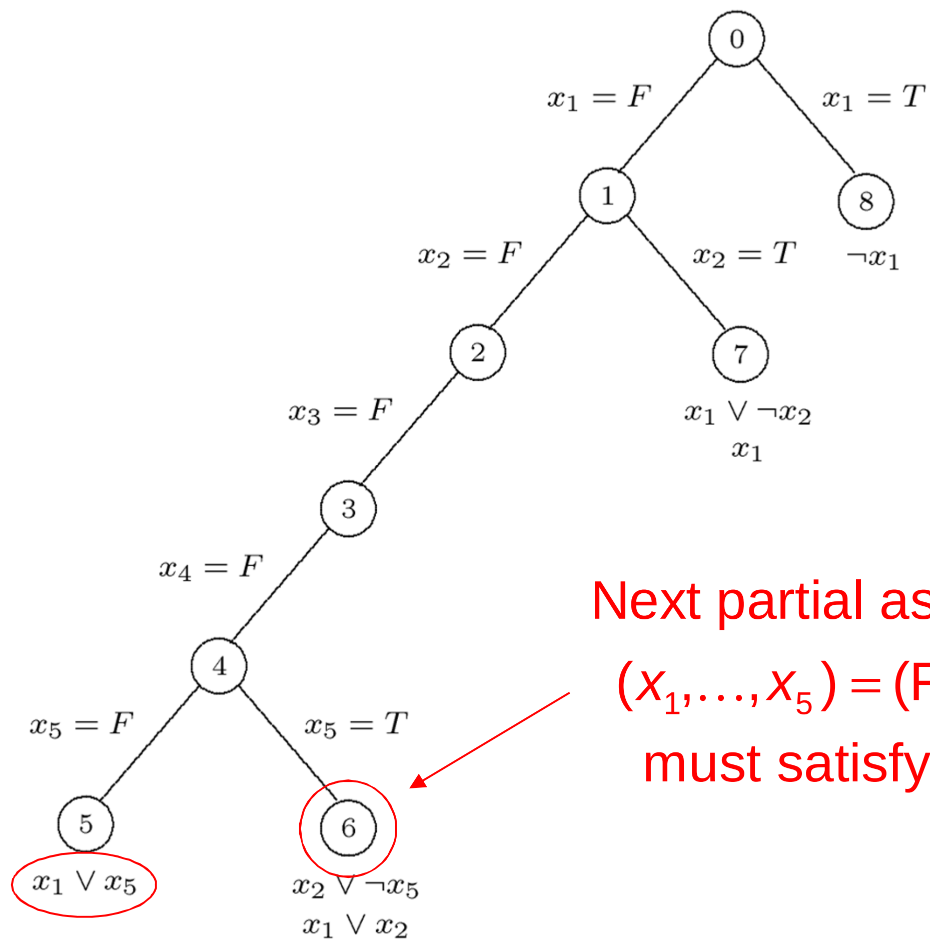
Proof remains valid when only $x_1$ and $x_5$ are fixed to F.

So we have the nogood

$$(x_1, x_5) = (F,F) \rightarrow f(x) \geq \infty$$

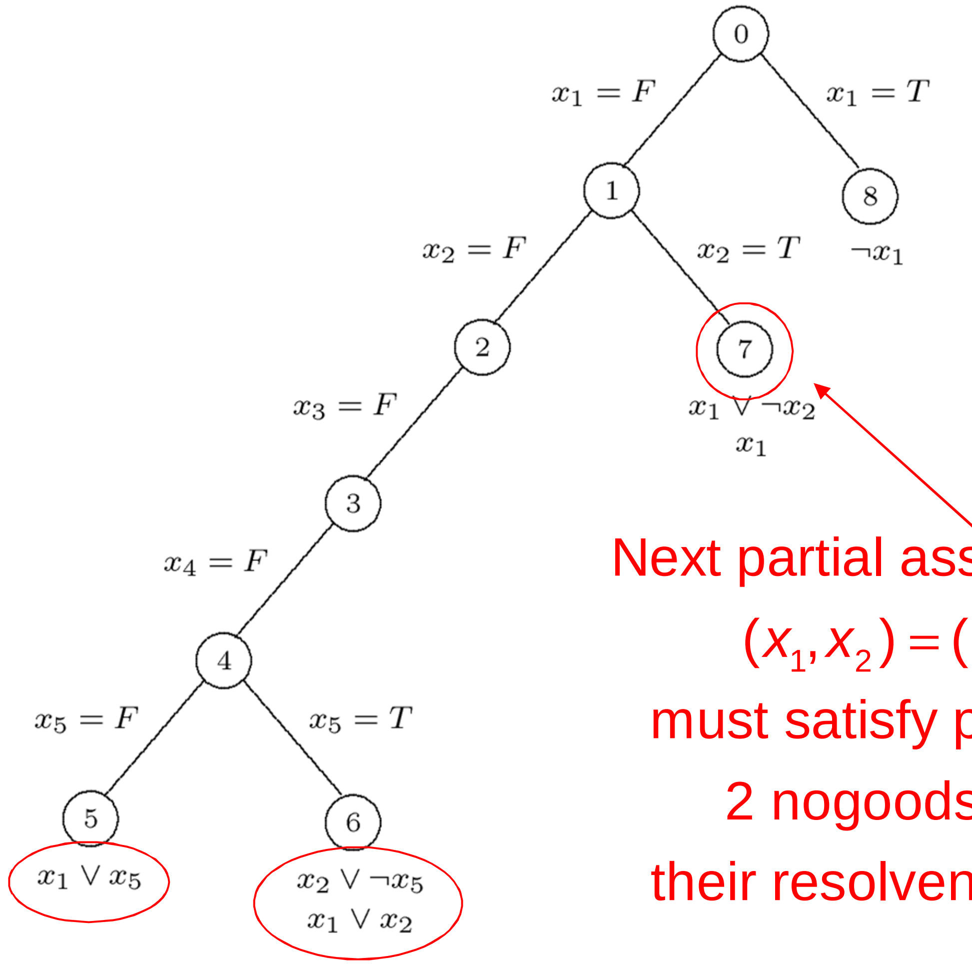
or **conflict clause** $x_1 \vee x_5$

# SAT: Nogood-Based Search



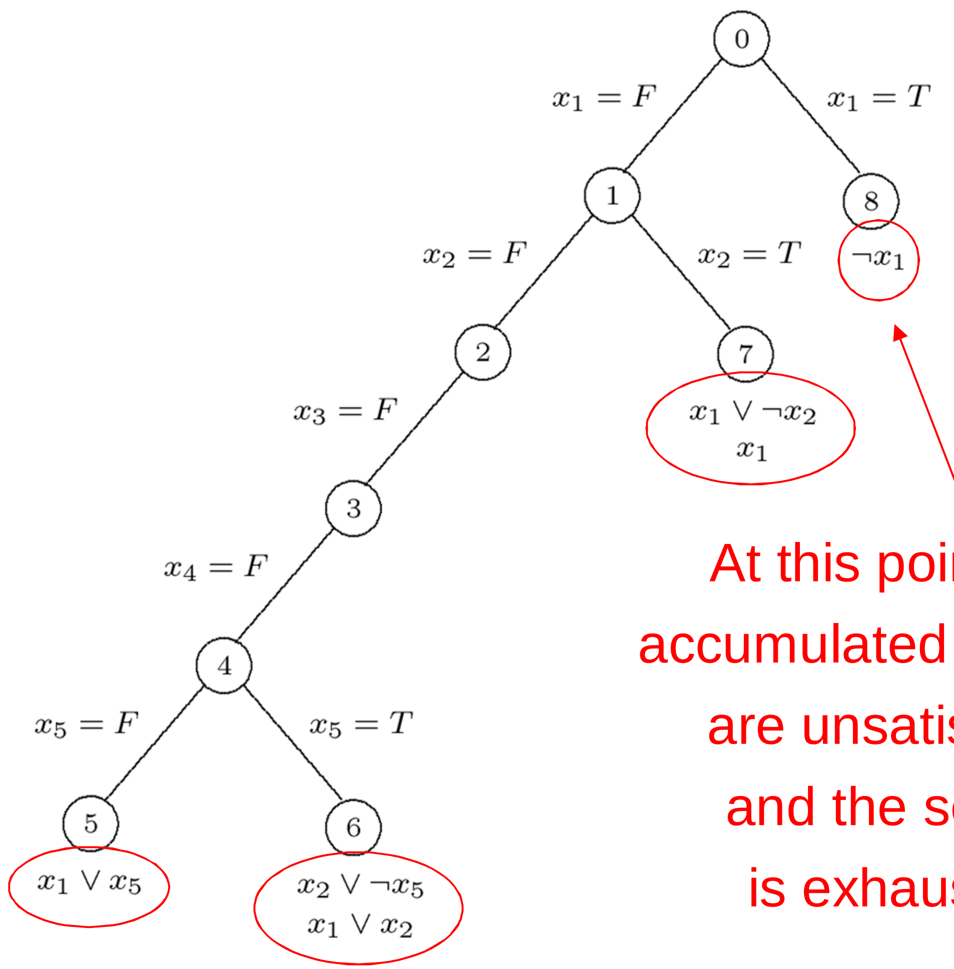

Next partial assignment $(x_1,\ldots,x_5) = (F,F,F,F,T)$ must satisfy nogood

# SAT: Nogood-Based Search




Next partial assignment
$(x_1, x_2) = (F, T)$
must satisfy previous
2 nogoods and
their resolvent $x_1 \vee x_2$

# SAT: Nogood-Based Search




At this point the accumulated nogoods are unsatisfiable and the search is exhaustive

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Domain Filtering Duals

- Since a domain filter is an inference method, it defines an **inference dual** and a **nogood-based search method**.
  - Let $[L_j,U_j]$ be the domain of $x_j$.
  - So $x \in [L,U]$ where $L = (L_1,\ldots,L_n)$, $U = (U_1,\ldots,U_n)$
  - Then the inference dual can be defined

$$\max_{p\in\mathcal{P}}\left\{ v \;\middle|\; C \underset{P}{\rightarrow} (f(x) \geq v) \right\} = \max_{P\in\mathcal{P}}\left\{ f(x) \;\middle|\; C \underset{P}{\rightarrow} (x \in [L,U]) \right\}$$

where $\mathcal{P}$ contains the domain filtering operations.

# Domain Filtering Duals

- Suppose we are branching.
  - Let $\mathcal{B}$ contain the current branching constraints.
  - Let $(\bar{v}, \bar{P})$ solve the inference dual.
  - Let $\bar{\mathcal{B}}$ contain the constraints in $\mathcal{B}$ used as premises in the filtering operation $\bar{P}$
  - We have the nogood

$$\bar{\mathcal{B}} \rightarrow \left( f(x) \geq \bar{v} \right)$$

# Domain Filtering Duals

- **Example:** assign jobs to machines and schedule them to minimize makespan.

Start time of job *j*

processing time of job *j* on machine *i*

$$\min_{\substack{x_j \in \{1,\ldots,m\},\ \text{all } j \\ s_j \in [r_j, d_j],\ \text{all } j}} \left\{ v \;\middle|\; \begin{array}{l} v \ge s_j + p_{x_j j},\ \text{all } j \\ \text{disjunctive}\big((s_j \mid x_j = i), (p_{ij} \mid x_j = i)\big),\ \text{all } i \end{array} \right\}$$

Time window for job *j*

Machine assigned to job *j*

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Planning & Scheduling: Filtering Dual

Suppose there are 5 jobs and 2 machines:

| *Job* $j$ | *Release time* $r_j$ | *Dead-line* $d_j$ | *Processing time* $p_{Aj}$ | $p_{Bj}$ |
|---|---|---|---|---|
| 1 | 0 | 10 | 2 | 2 |
| 2 | 0 | 8 | 3 | 2 |
| 3 | 2 | 7 | 3 | 2 |
| 4 | 2 | 5 | 4 | 3 |
| 5 | 4 | 7 | 2 | 2 |

We will search over machine assignments *x*.

# Planning & Scheduling: Filtering Dual

Suppose there are 5 jobs and 2 machines:

| *Job* $j$ | *Release time* $r_j$ | *Dead-line* $d_j$ | *Processing time* $p_{Aj}$ | $p_{Bj}$ |
|---|---|---|---|---|
| 1 | 0 | 10 | 2 | 2 |
| 2 | 0 | 8 | 3 | 2 |
| 3 | 2 | 7 | 3 | 2 |
| 4 | 2 | 5 | 4 | 3 |
| 5 | 4 | 7 | 2 | 2 |

We will search over machine assignments *x.*

At the current point in the search, we have assigned jobs 1,2,3,5 to machine A and job 4 to machine B.

# Planning & Scheduling: Filtering Dual

Suppose there are 5 jobs and 2 machines:

| *Job* $j$ | *Release time* $r_j$ | *Dead-line* $d_j$ | *Processing time* $p_{Aj}$ | $p_{Bj}$ |
|---|---|---|---|---|
| 1 | 0 | 10 | 2 | 2 |
| 2 | 0 | 8 | 3 | 2 |
| 3 | 2 | 7 | 3 | 2 |
| 4 | 2 | 5 | 4 | 3 |
| 5 | 4 | 7 | 2 | 2 |

We will search over machine assignments *x*.

At the current point in the search, we have assigned jobs 1,2,3,5 to machine A and job 4 to machine B.

Min makespan on machine B is 5.

# Planning & Scheduling: Filtering Dual

Suppose there are 5 jobs and 2 machines:

| *Job* $j$ | *Release time* $r_j$ | *Dead-line* $d_j$ | *Processing time* $p_{Aj}$ | $p_{Bj}$ |
|---|---|---|---|---|
| 1 | 0 | 10 | 2 | 2 |
| 2 | 0 | 8 | 3 | 2 |
| 3 | 2 | 7 | 3 | 2 |
| 4 | 2 | 5 | 4 | 3 |
| 5 | 4 | 7 | 2 | 2 |

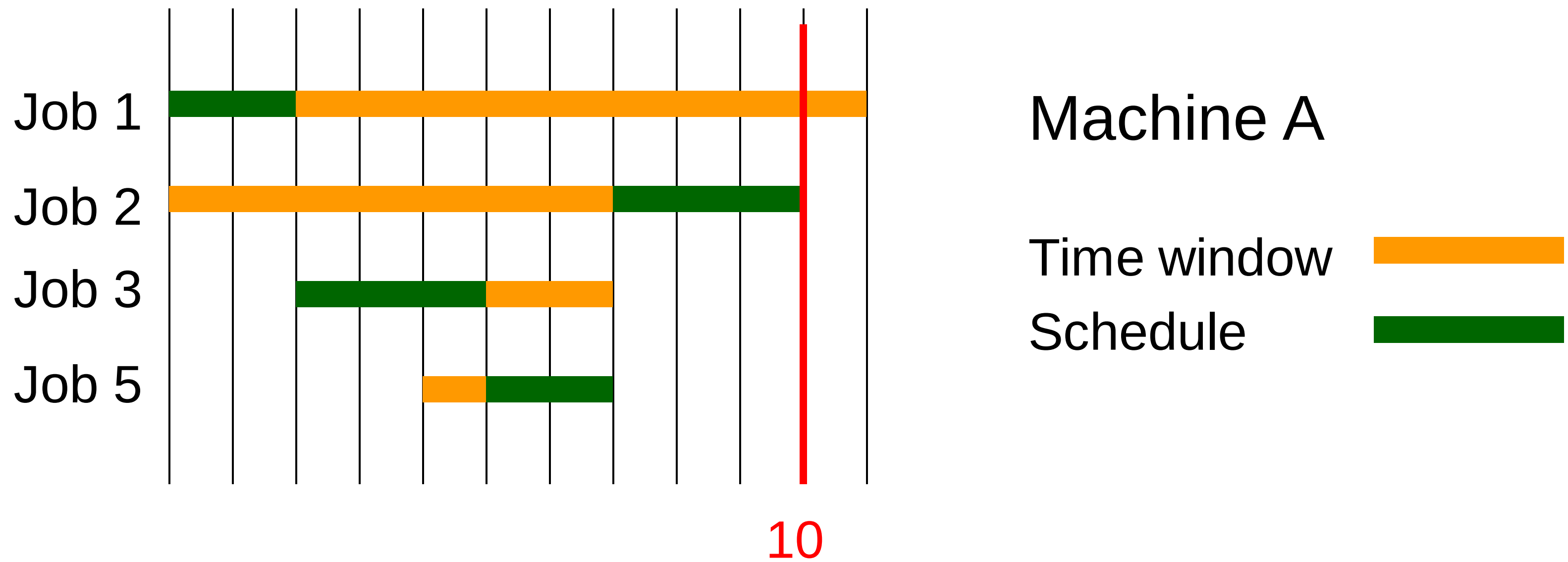
We will search over machine assignments *x.*

At the current point in the search, we have assigned jobs 1,2,3,5 to machine A and job 4 to machine B.

Min makespan on machine B is 5.

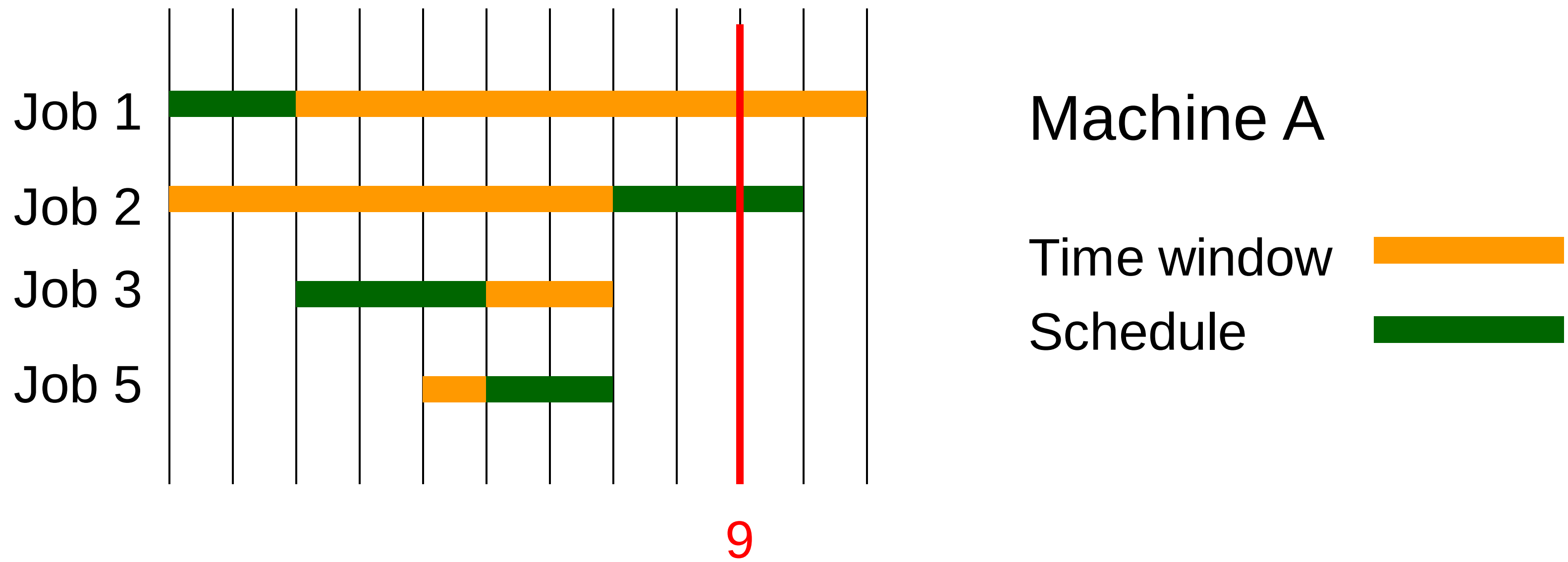
Let's compute min makespan on machine A…

# Planning & Scheduling: Filtering Dual




We find a feasible schedule with makespan 10.

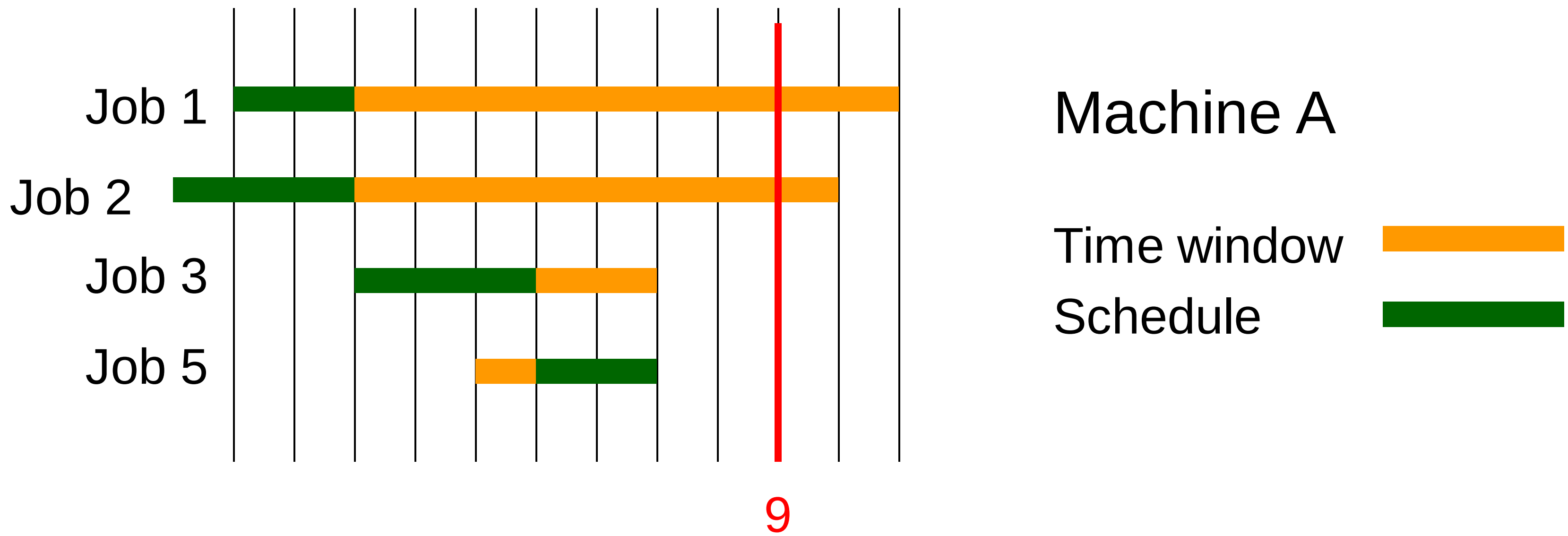
# Planning & Scheduling: Filtering Dual




We found a feasible schedule with makespan 10.

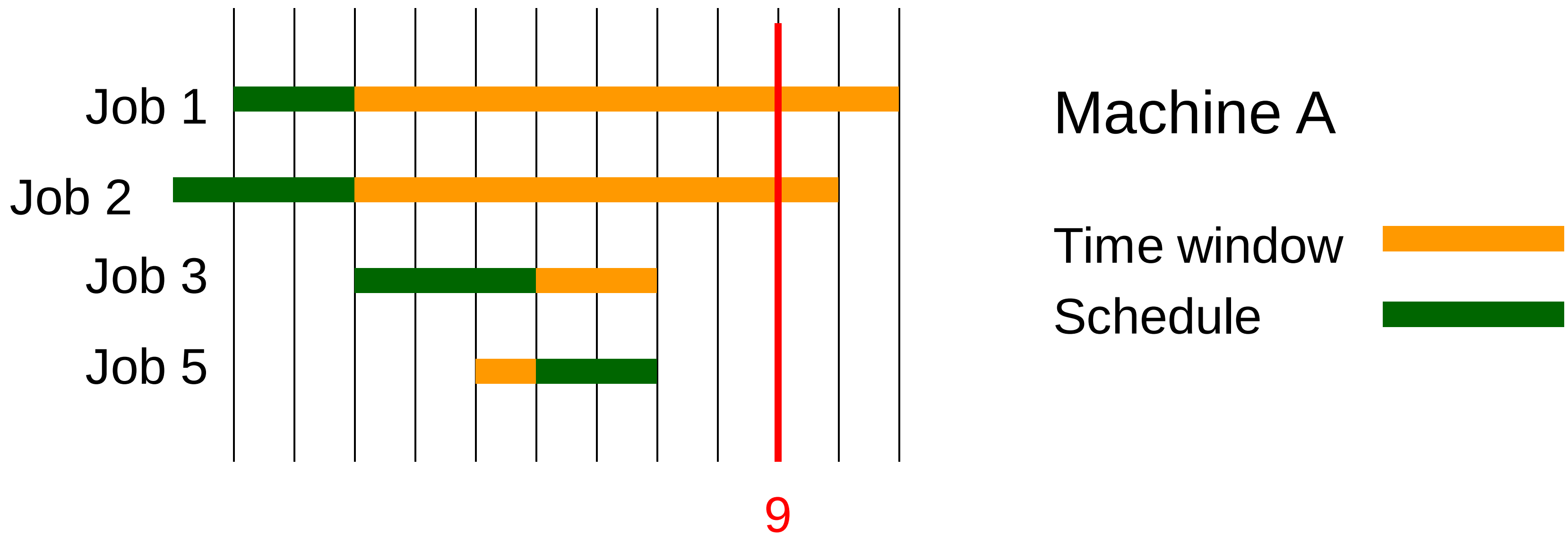
To prove optimality, we check whether a makespan of 9 is possible.

# Planning & Scheduling: Filtering Dual




Edge finding discovers that job 2 must precede jobs 3 and 5.

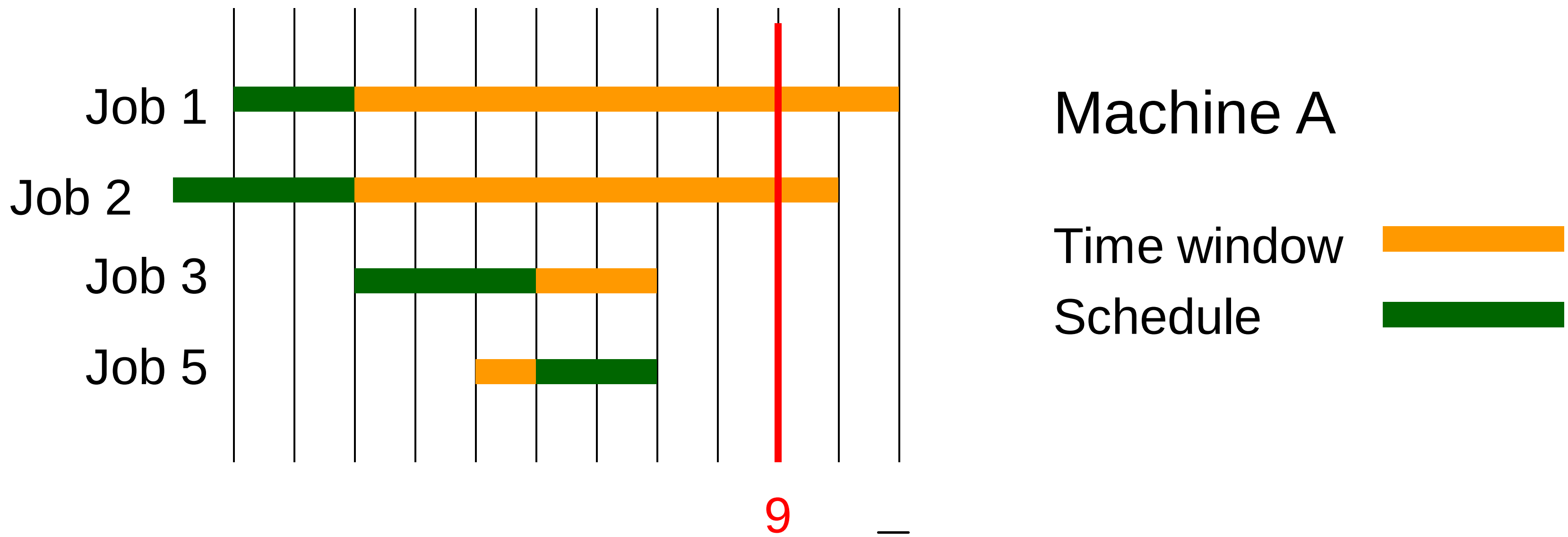
# Planning & Scheduling: Filtering Dual




Edge finding discovers that job 2 must precede jobs 3 and 5.

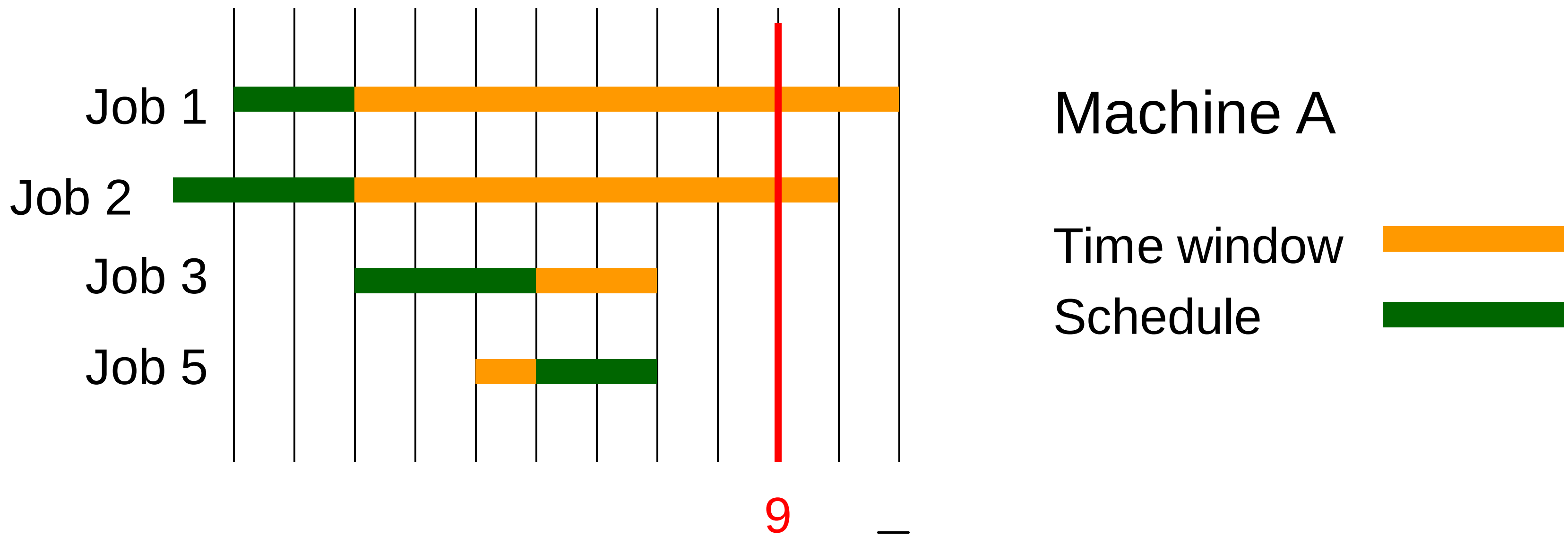
Since this reduces domain of $s_2$ to empty set, there is no feasible solution with makespan 9.

# Planning & Scheduling: Filtering Dual




This edge finding procedure $\bar{P}$ solves the inference dual of the scheduling problem.

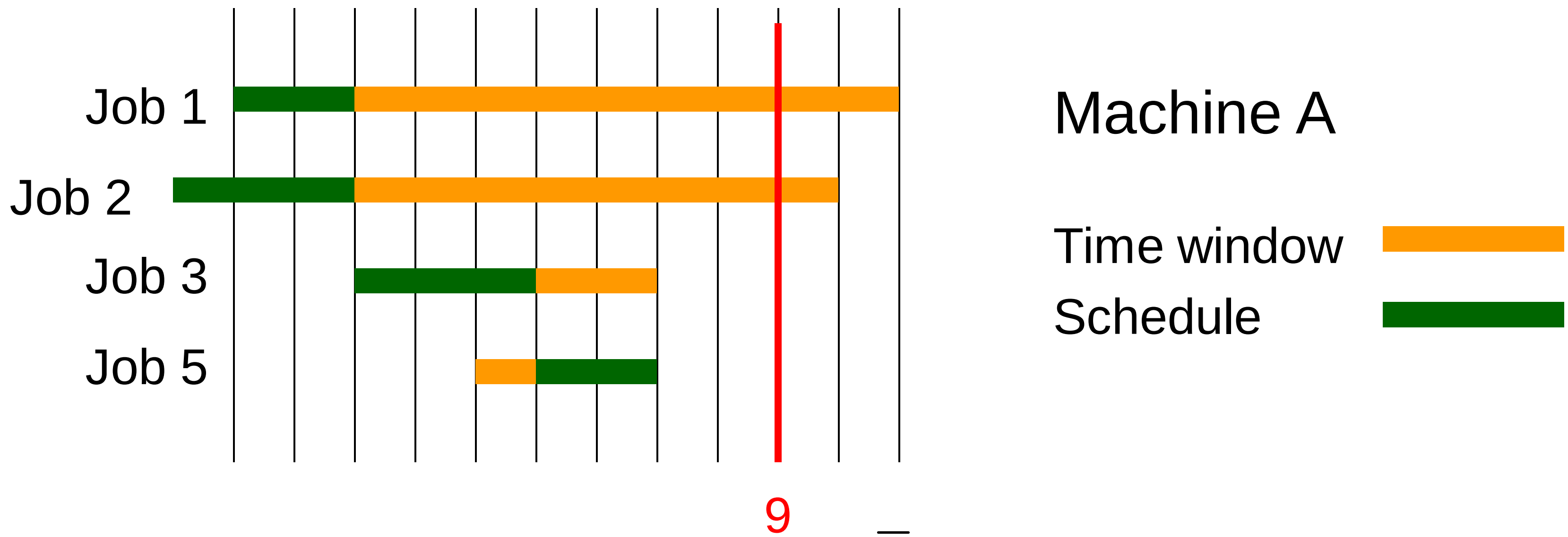
# Planning & Scheduling: Filtering Dual




This edge finding procedure $\bar{P}$ solves the inference dual of the scheduling problem.

$\bar{P}$ derives infeasibility from the assignments $\mathcal{B}$ of jobs 1,2,3,5 to machine A.

# Planning & Scheduling: Filtering Dual

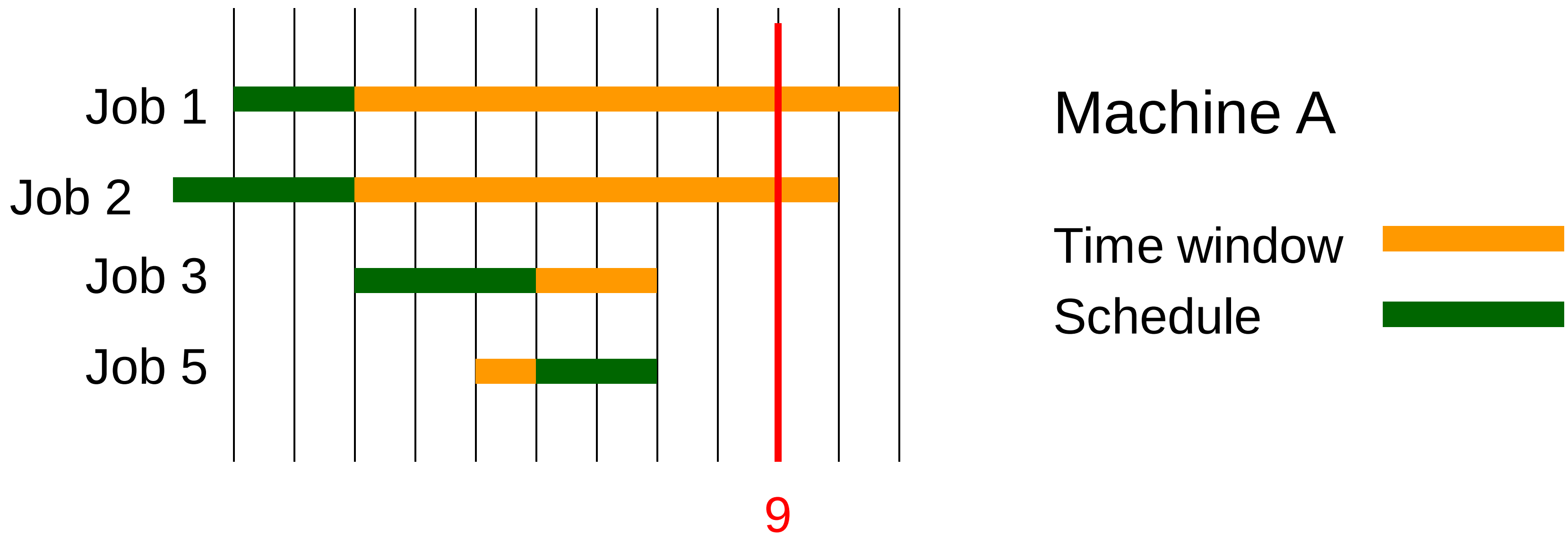



This edge finding procedure $\bar{P}$ solves the inference dual of the scheduling problem.

$\bar{P}$ derives infeasibility from the assignments $\mathcal{B}$ of jobs 1,2,3,5 to machine A.

Is there a smaller set $\bar{\mathcal{B}}$ of assignments that is sufficient to prove infeasibility?

# Planning & Scheduling: Filtering Dual

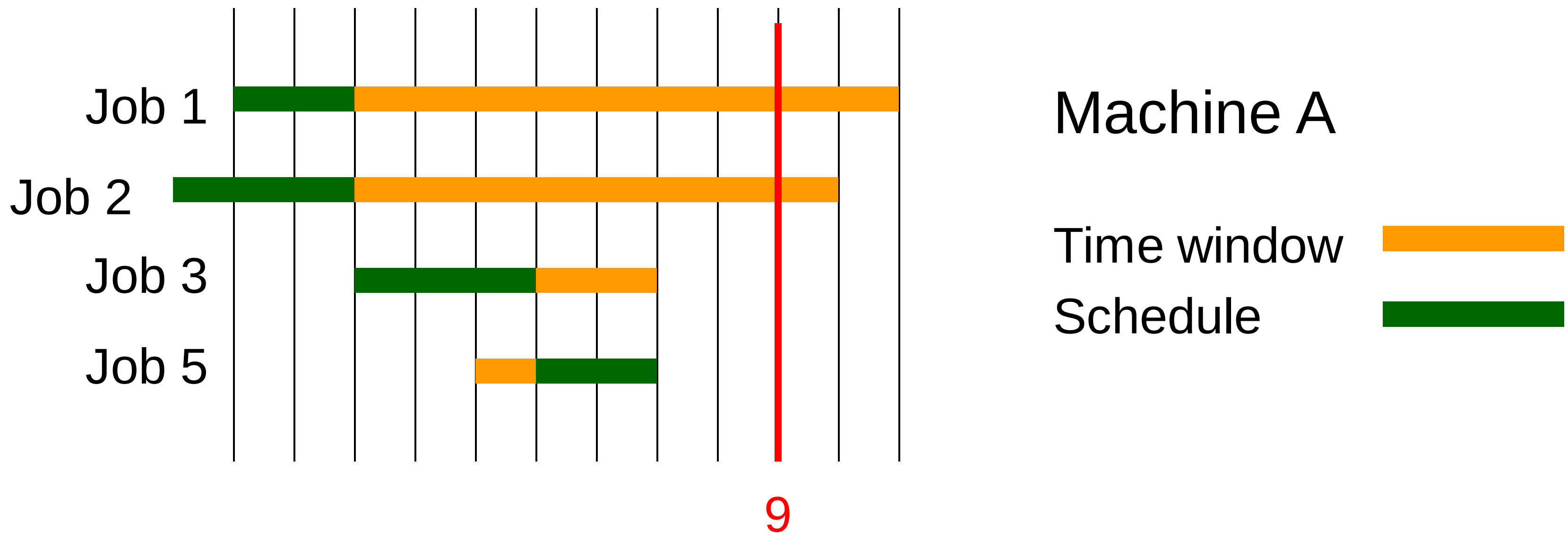



Only jobs 2, 3, 5 were involved in this deduction.
So we have the nogood:

$$(x_2 = x_3 = x_5 = A) \rightarrow (v \geq 10)$$

$\bar{\mathcal{B}}$

# Planning & Scheduling: Filtering Dual

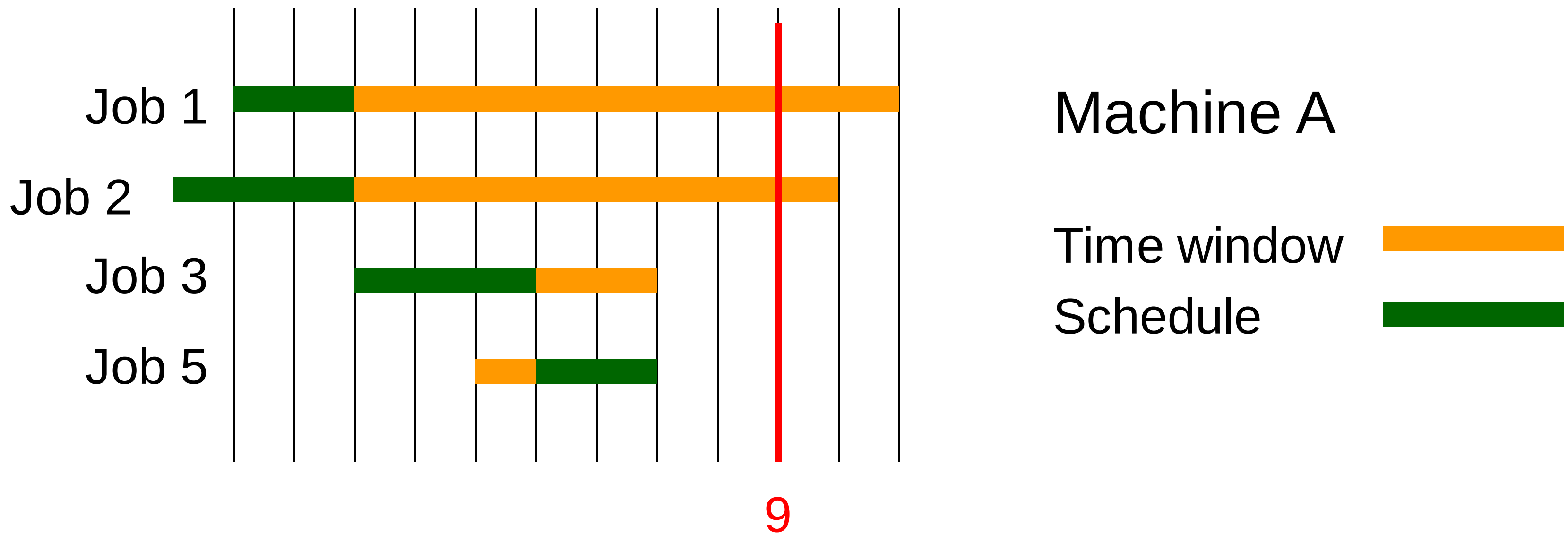



Only jobs 2, 3, 5 were involved in this deduction.
So we have the nogood:

$$(x_2 = x_3 = x_5 = A) \rightarrow (v \geq 10)$$

To improve the solution, the search must avoid this assignment.

# Planning & Scheduling: Filtering Dual




Only jobs 2, 3, 5 were involved in this deduction.
So we have the nogood:

$$(x_2 = x_3 = x_5 = A) \rightarrow (v \geq 10)$$

Since the nogoods contain a fixed set of variables $x$, this is a generalization of **Benders decomposition**

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Constraint Dual

- The **constraint dual** is neither a relaxation dual nor an inference dual.
- However, one can define a relaxation dual for a constraint dual formulation.
  - **Mini-bucket** elimination is a special case.
- This is best explained by example…

# Constraint Dual

- Consider the 0-1 problem

$$\begin{aligned} \min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1 + x_2 + x_4 \geq 1 \\ & x_2 + x_3 + x_4 \geq 1 \\ & x_j \in \{0,1\} \end{aligned}$$

# Constraint Dual

- Consider the 0-1 problem

$$
\begin{aligned}
\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 = 2 \\
& x_1 + x_2 + x_3 \geq 1 \\
& x_1 + x_2 + x_4 \geq 1 \\
& x_2 + x_3 + x_4 \geq 1 \\
& x_j \in \{0,1\}
\end{aligned}
$$

  – Solution: $(x_1, \ldots, x_4) = (0,1,0,0)$

# Constraint Dual

- One form of constraint dual (ignoring the objective function) is

$$
\begin{aligned}
& x_1^1 = x_1^2 \\
& x_2^1 = x_2^2 = x_2^3 \\
& x_3^1 = x_3^3 \\
& x_4^1 = x_4^3 \\
& (x_1^1, x_2^1, x_3^1) \in D \\
& (x_1^2, x_2^2, x_4^2) \in D \\
& (x_2^3, x_3^3, x_4^3) \in D
\end{aligned}
$$

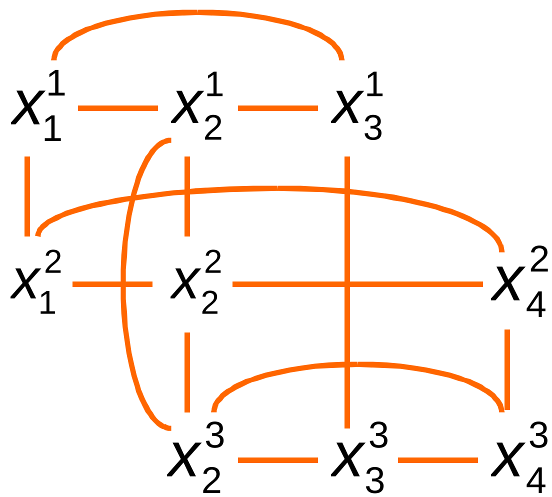
where $D = \{0,1\}^3 - \{(0,0,0)\}$

$$
\begin{aligned}
\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\
& x_1 + x_2 + x_3 \geq 1 \\
& x_1 + x_2 + x_4 \geq 1 \\
& x_2 + x_3 + x_4 \geq 1 \\
& x_j \in \{0,1\}
\end{aligned}
$$

Standardize apart variables in different constraints and equate them in binary constraints.

# Constraint Dual

Dependency graph for constraint dual:



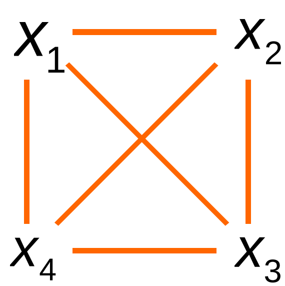
$$\begin{aligned}\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1 + x_2 + x_4 \geq 1 \\ & x_2 + x_3 + x_4 \geq 1 \\ & x_j \in \{0,1\}\end{aligned}$$

Induced width of dependency graph is 3.

This indicates complexity of solving the problem by nonserial dynamic programming.

# Constraint Dual

Dependency graph of original problem also has induced width 3:




$$
\begin{aligned}
\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\
& x_1 + x_2 + x_3 \geq 1 \\
& x_1 + x_2 + x_4 \geq 1 \\
& x_2 + x_3 + x_4 \geq 1 \\
& x_j \in \{0,1\}
\end{aligned}
$$

# Constraint Dual

- One way to relax the dynamic programming model is to remove edges from the dependency graph.
  - and form "mini-buckets"

$$\begin{aligned} \min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1 + x_2 + x_4 \geq 1 \\ & x_2 + x_3 + x_4 \geq 1 \\ & x_j \in \{0,1\} \end{aligned}$$

$$\min f_1(x_1,x_2,x_3) + f_2(x_1,x_2,x_4) + f_3(x_2,x_3,x_4)$$

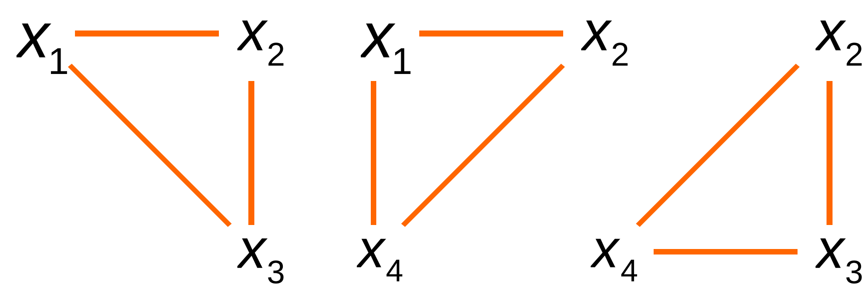
$$f_1(x_1,x_2,x_3) = \begin{cases} x_1 + 2x_2 + 3x_3 & \text{if } x_1 + x_2 + x_3 \geq 1 \\ \infty & \text{otherwise} \end{cases}$$

$$f_2(x_1,x_2,x_4) = \begin{cases} 0x_1 + 0x_2 + 4x_4 & \text{if } x_1 + x_2 + x_4 \geq 1 \\ \infty & \text{otherwise} \end{cases}$$

$$f_3(x_2,x_3,x_4) = \begin{cases} 0x_2 + 0x_3 + 0x_4 & \text{if } x_2 + x_3 + x_4 \geq 1 \\ \infty & \text{otherwise} \end{cases}$$

# Constraint Dual

- The problem separates into 3 problems with induced width 2:




$$\min f_1(x_1,x_2,x_3)+f_2(x_1,x_2,x_4)+f_3(x_2,x_3,x_4)$$

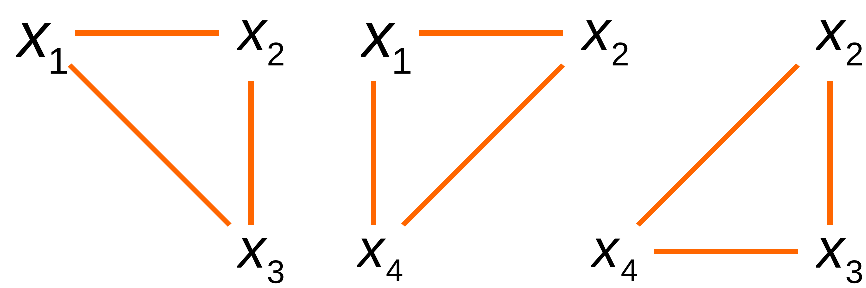
$$f_1(x_1,x_2,x_3)=\begin{cases} x_1+2x_2+3x_3 & \text{if } x_1+x_2+x_3\geq 1 \\ \infty & \text{otherwise}\end{cases}$$

$$f_2(x_1,x_2,x_4)=\begin{cases} 0x_1+0x_2+4x_4 & \text{if } x_1+x_2+x_4\geq 1 \\ \infty & \text{otherwise}\end{cases}$$

$$f_3(x_2,x_3,x_4)=\begin{cases} 0x_2+0x_3+0x_4 & \text{if } x_2+x_3+x_4\geq 1 \\ \infty & \text{otherwise}\end{cases}$$

# Constraint Dual

- But the resulting bound is weak




$$\min f_1(x_1,x_2,x_3)+f_2(x_1,x_2,x_4)+f_3(x_2,x_3,x_4)=1+0+0=1$$

$$f_1(x_1,x_2,x_3)=\begin{cases} x_1+2x_2+3x_3 & \text{if } x_1+x_2+x_3\geq 1 \\ \infty & \text{otherwise} \end{cases}$$

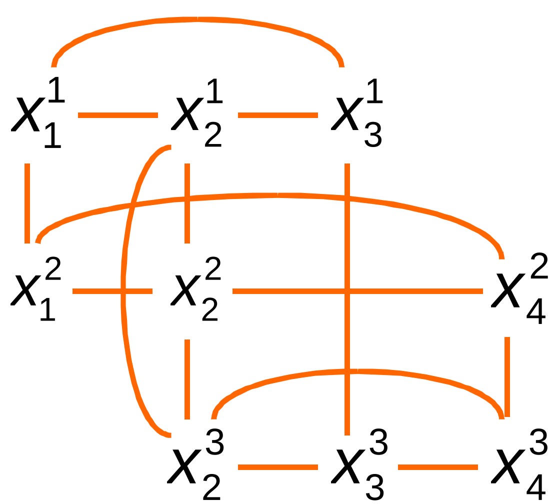
$$f_2(x_1,x_2,x_4)=\begin{cases} 0x_1+0x_2+4x_4 & \text{if } x_1+x_2+x_4\geq 1 \\ \infty & \text{otherwise} \end{cases}$$

$$f_3(x_2,x_3,x_4)=\begin{cases} 0x_2+0x_3+0x_4 & \text{if } x_2+x_3+x_4\geq 1 \\ \infty & \text{otherwise} \end{cases}$$

# Constraint Dual

Another approach is to delete edges from the dependency graph of the **constraint dual**.

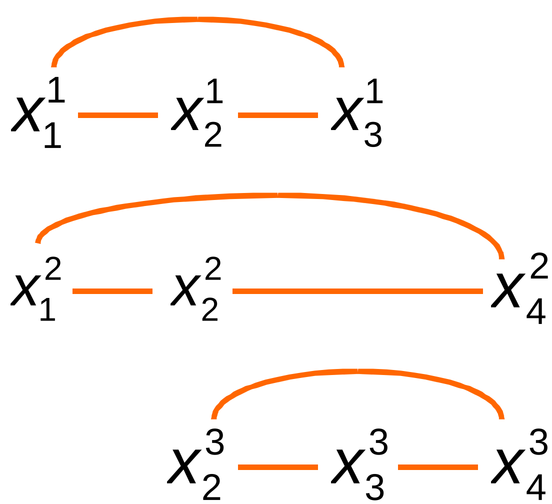
$$\begin{aligned} \min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1 + x_2 + x_4 \geq 1 \\ & x_2 + x_3 + x_4 \geq 1 \\ & x_j \in \{0,1\} \end{aligned}$$



# Constraint Dual

Another approach is to delete edges from the dependency graph of the **constraint dual**.

$$\begin{aligned} \min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\ & x_1 + x_2 + x_3 \ge 1 \\ & x_1 + x_2 + x_4 \ge 1 \\ & x_2 + x_3 + x_4 \ge 1 \\ & x_j \in \{0,1\} \end{aligned}$$

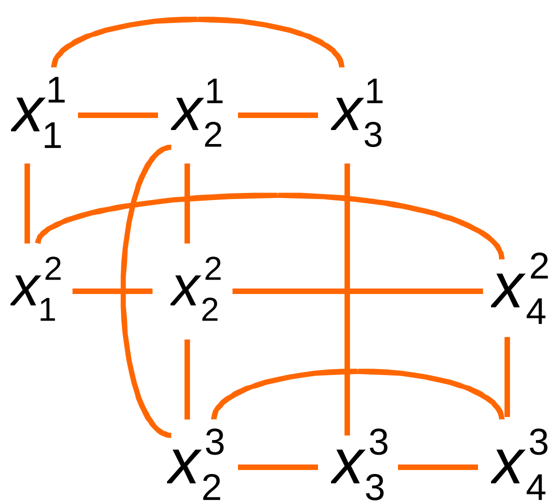



The previous mini-bucket scheme deletes all edges connecting variables.

# Constraint Dual

Another approach is to delete edges from the dependency graph of the **constraint dual**.

$$
\begin{aligned}
\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\
& x_1 + x_2 + x_3 \geq 1 \\
& x_1 + x_2 + x_4 \geq 1 \\
& x_2 + x_3 + x_4 \geq 1 \\
& x_j \in \{0,1\}
\end{aligned}
$$

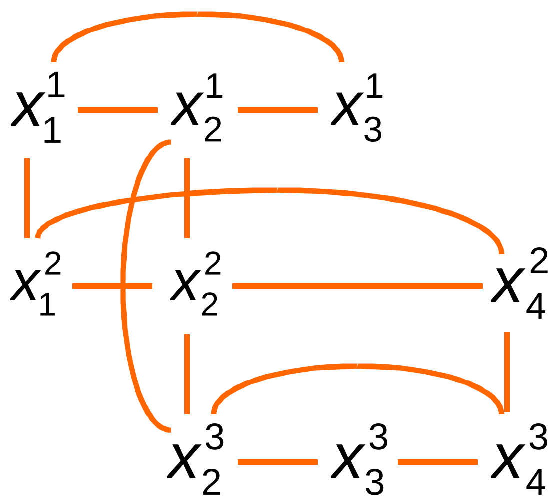



By deleting only one edge…

# Constraint Dual

Another approach is to delete edges from the dependency graph of the **constraint dual**.

$$
\begin{aligned}
\min\ & x_1 + 2x_2 + 3x_3 + 4x_4 \\
& x_1 + x_2 + y_3 \geq 1 \\
& x_1 + x_2 + x_4 \geq 1 \\
& x_2 + x_3 + x_4 \geq 1 \\
& x_j \in \{0,1\}
\end{aligned}
$$

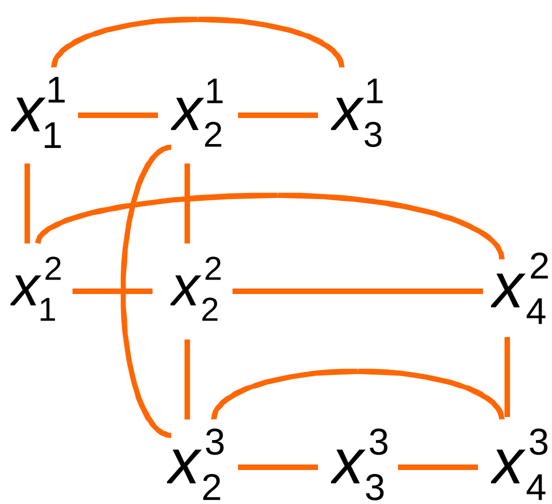



By deleting only one edge…
one can reduce the induced width to 2.

# Constraint Dual

Another approach is to delete edges from the dependency graph of the **constraint dual**.

$$\begin{aligned} \min\ & x_1 + 2x_2 + 3x_3 + 4x_4 = 2 \\ & x_1 + x_2 + y_3 \geq 1 \\ & x_1 + x_2 + x_4 \geq 1 \\ & x_2 + x_3 + x_4 \geq 1 \\ & x_j \in \{0,1\} \end{aligned}$$



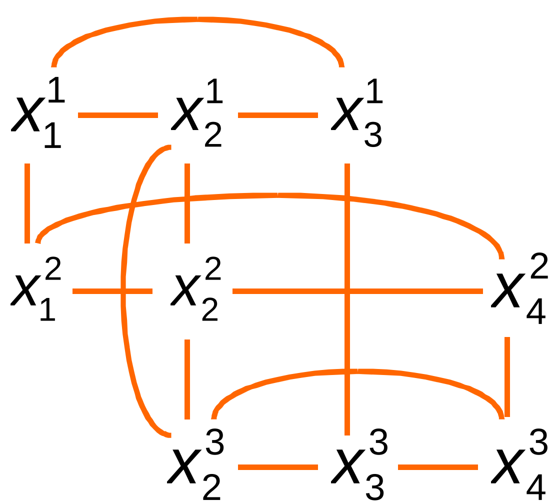

By deleting only one edge…
one can reduce the induced width to 2.

The bound is now tight.

| *Problem* | *Relaxation Dual* | *Inference Dual* | *Nogood-based search* |
|---|---|---|---|
| *Linear programming* | LP dual | LP dual | Benders decomposition |
| *Inequality-constrained problems* | Surrogate dual | Surrogate dual | |
| *Inequality-constrained problems* | Lagrangean dual | Lagrangean dual | Special case: generalized Benders |
| *SAT* | | Unit resolution | DPL with conflict clauses |
| *Domain filtering* | | Filtering dual | |
| *Planning & scheduling* | | Edge finding | Logic-based Benders |
| *Constraint dual* | State-space relaxations (e.g. mini-buckets) | | |

# Constraint Dual

- One can now define a **relaxation dual**: choose a set of eliminated edges that provides the tightest bound.
  - The choice of edges parameterizes the relaxation.




The dual can be solved by various heuristics, such as local search.

# Primal-Dual Methods

- Primal-dual **inference** methods
  - Branch-and-cut for integer programming
    - Combine branching (search over restrictions) with inferred cutting planes.
  - Standard CP methods
    - Combine branching with filtering.

# Primal-Dual Methods

- Primal-Dual Relaxation Methods
  - Branch-and-bound with Lagrangean relaxation.
    - Solve Lagrangean relaxation at each node of search tree.
  - Interleave search over restrictions with search over relaxations.
    - Dual simplex method for linear programming.
    - “Out-of-kilter” method for network flows.
    - Dual ascent method for integer programming.

# Duality of Duals?

- Why can so many duals be interpreted as **both** relaxation duals and inference duals?
  - Particularly math programming duals.
- Is there a duality of duals?

# Duality of Duals?

- Why can so many duals be interpreted as **both** relaxation duals and inference duals?
  - Particularly math programming duals.
- Is there a duality of duals?

- No.
  - But there is a formal relationship between the duals…

# Duality of Duals?

- A relaxation dual in which there is a **finite algorithm** for solving the relaxation is an inference dual.
  - The solution algorithm can be viewed as **inferring** the best bound.
- An inference dual in which the **proofs are parameterized** is a relaxation dual.
  - The proof of the best bound can be viewed as solving the tightest relaxation.

# Duality of Duals?

- Math programming duals tend to be conceived as relaxation duals.
  - There is always an algorithm for solving the relaxation, otherwise the dual would be useless.
  - So math programming duals tend to be both relaxation and inference duals.
- Inference methods are not ordinarily parameterized.
  - So inference duals are not necessarily relaxation duals.