A Benders-Based Scheme for Combining Constraint Programming and Mixed Integer Programming

John Hooker
Carnegie Mellon University

Imperial College

March 2001

Some portions of this presentation represent research by:

- Ignacio Grossmann
- Vipul Jain
- Greg Ottosson
- Erlendur Thorsteinsson
- Hong Yan

Outline

- Idea behind Benders decomposition
- A simple example
- General analysis of logic-based Benders
- Classical Benders decomposition
- Logic circuit verification
- Combining MILP with CP: Machine scheduling
- An Enhancement: Branch-and-check

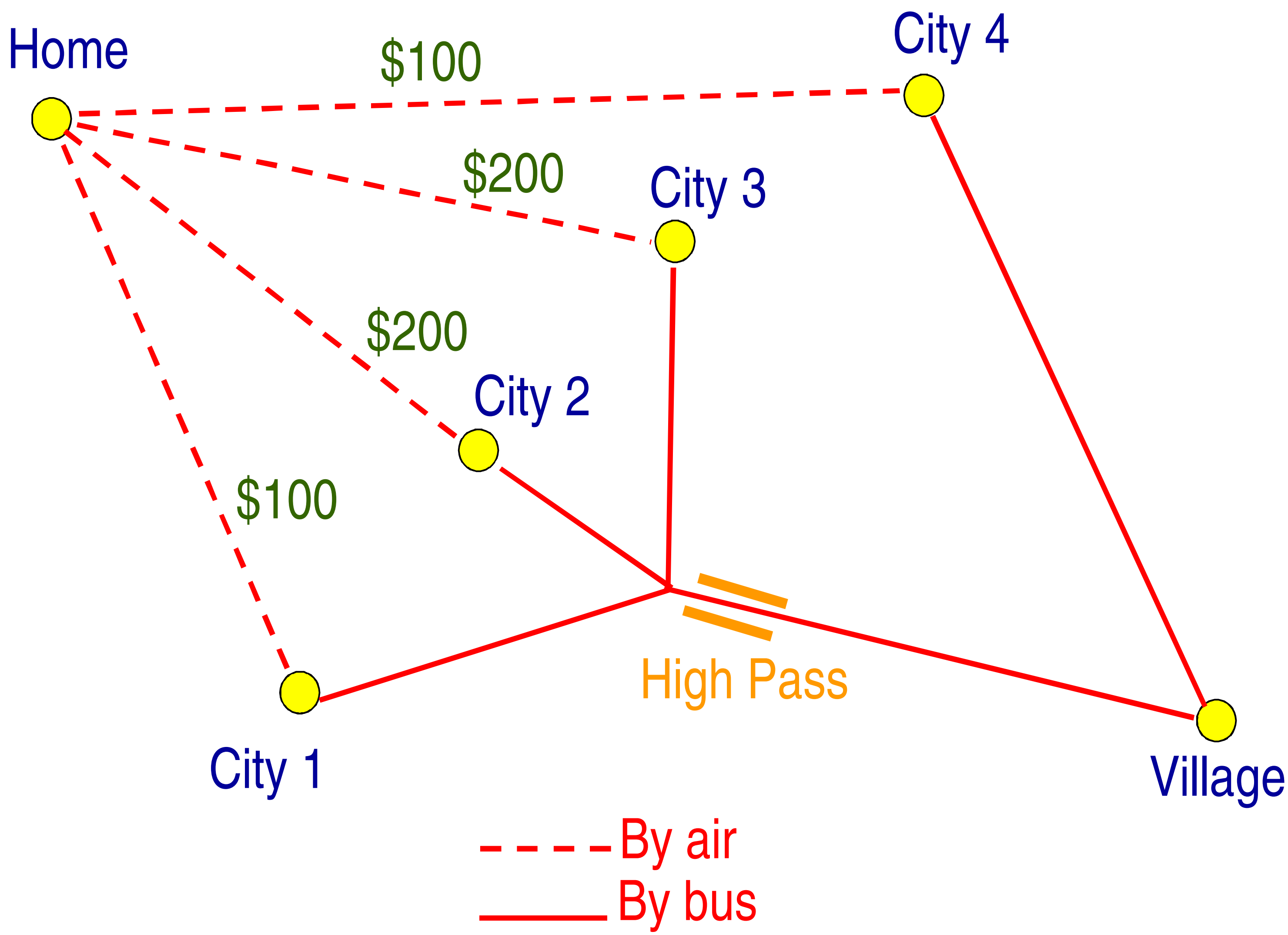
Idea Behind Benders Decomposition

“Learn from one’s mistakes.”

- Distinguish primary variables from secondary variables.
- Search over primary variables (*master problem*).
- For each trial value of primary variables, solve problem over secondary variables (*subproblem*).
- If solution is suboptimal, find out why. Design a constraint that rules out not only this solution but a large class of solutions that are suboptimal for the same reason (*Benders cut*).
- Add the Benders cut to the master problem and re-solve.

A Simple Example

Find Cheapest Route to a Remote Village

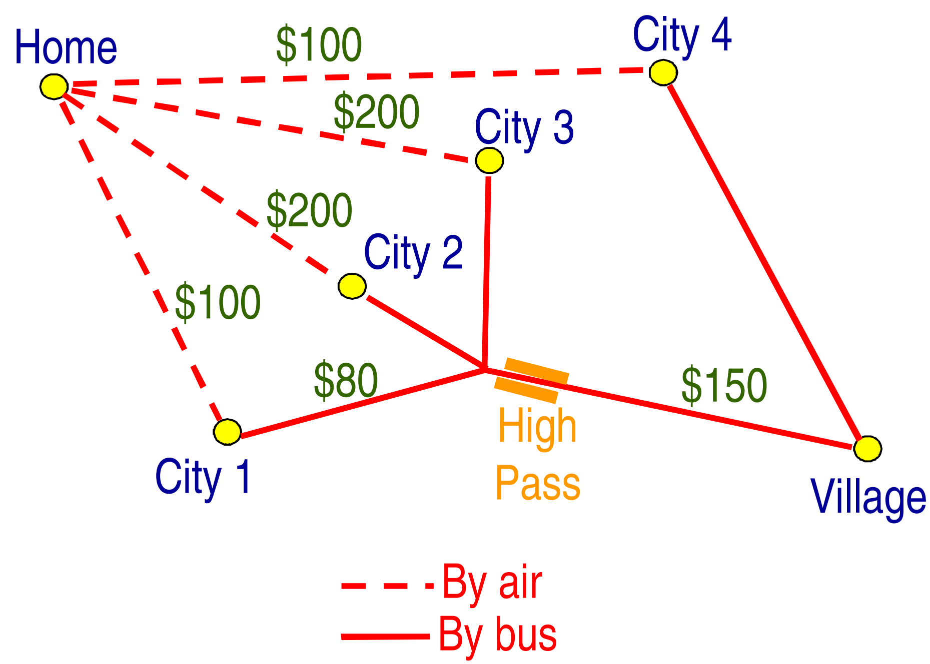


Let x = flight destination
y = bus route

Find cheapest route (x,y)

Begin with x = City 1 and pose the subproblem:

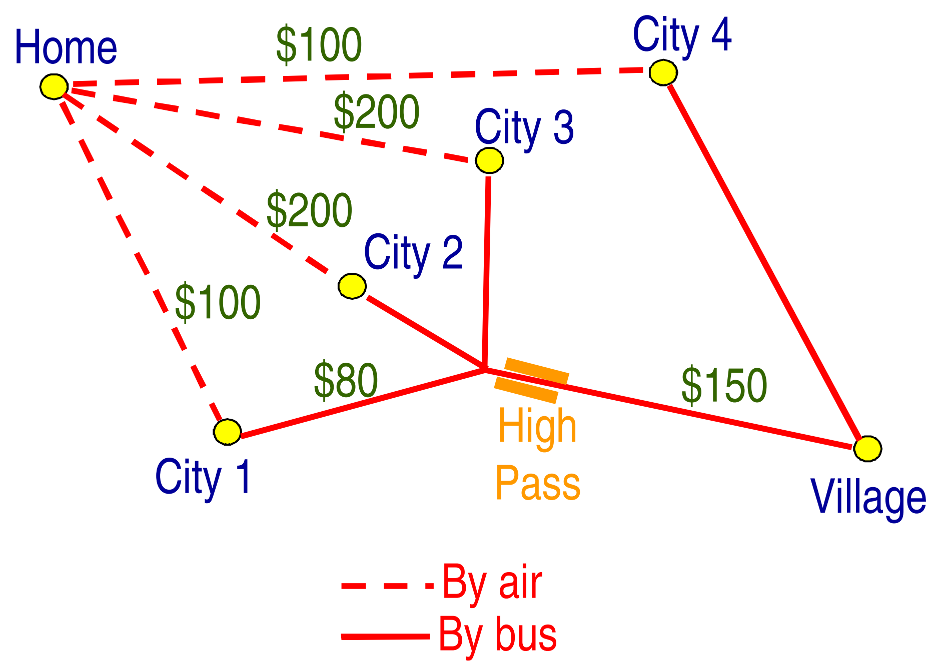
Find the cheapest route given that x = City 1.
Optimal cost is $100 + 80 + 150 = $330.



The "dual" problem of finding the optimal route is to prove optimality. The "proof" is that the route from City 1 to the village must go through High Pass. So

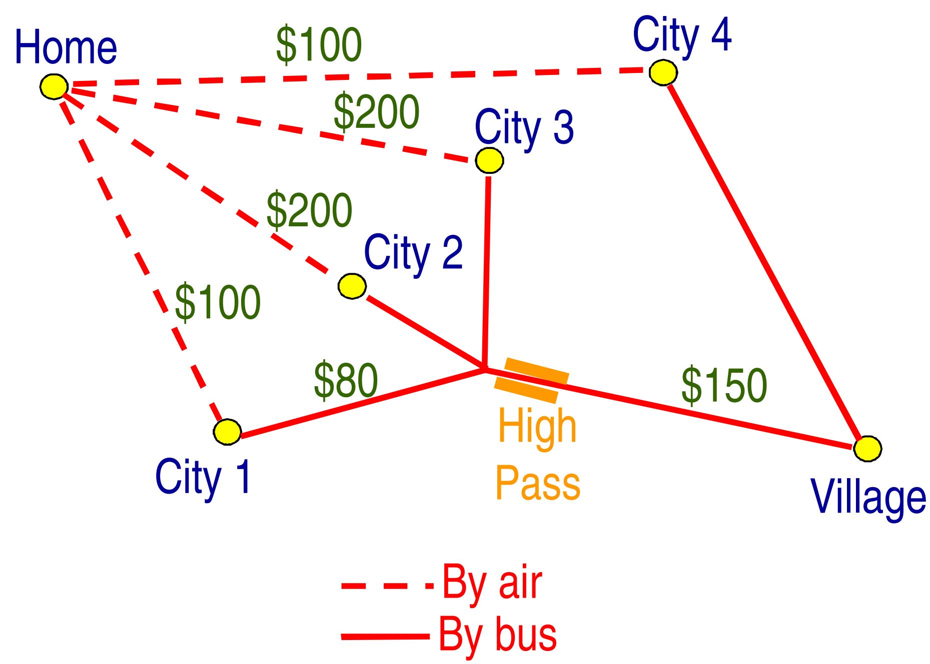
cost $\geq$ airfare + bus from city to High Pass + \$150

But *this same argument* applies to City 1, 2 or 3. This gives us the above Benders cut.



Specifically the Benders cut is

$$\text{cost} \geq B_{\text{City}\,1}(x) = \begin{cases} \$100 + 80 + 150 & \text{if } x = \text{City}\,1 \\ \$200 + 150 & \text{if } x = \text{City}\,2,3 \\ \$100 & \text{if } x = \text{City}\,4 \end{cases}$$



Now solve the *master problem:*

Pick the city x to minimize cost subject to

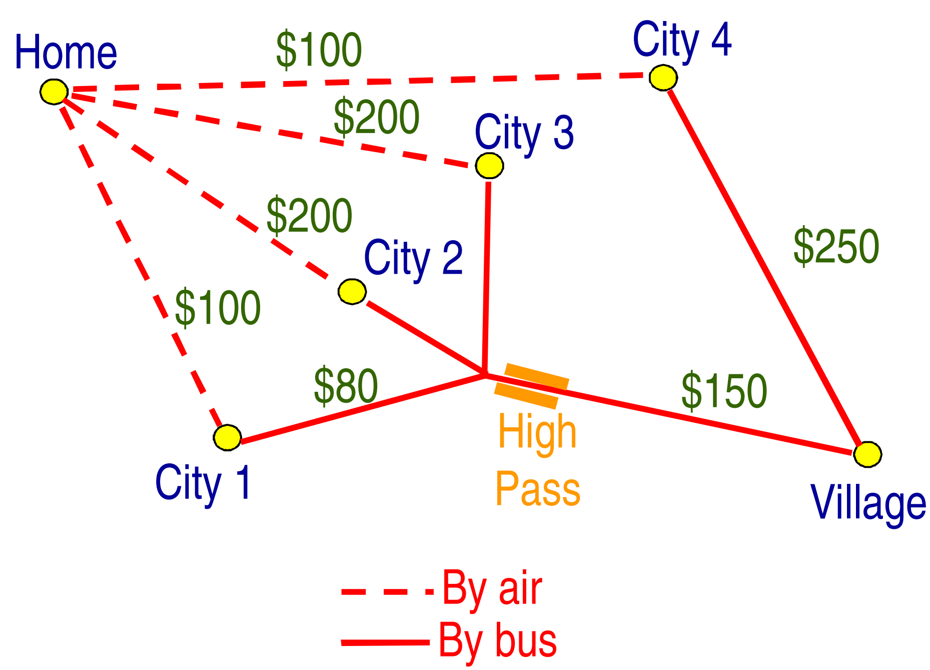
$$\text{cost} \geq B_{\text{City 1}}(x) = \left\{ \begin{array}{ll} \$100 + 80 + 150 & \text{if } x = \text{City 1} \\ \$200 + 150 & \text{if } x = \text{City 2,3} \\ \$100 & \text{if } x = \text{City 4} \end{array} \right\}$$

Clearly the solution is x = City 4, with cost \$100.

Now let x = City 4 and pose the subproblem:

Find the cheapest route given that x = City 4.

Optimal cost is \$100 + 250 = \$350.



Again solve the master problem:

Pick the city x to minimize cost subject to

$$\text{cost} \geq B_{\text{City 1}}(x) = \begin{Bmatrix} \$100+80+150 & \text{if } x = \text{City 1} \\ \$200+150 & \text{if } x = \text{City 2,3} \\ \$100 & \text{if } x = \text{City 4} \end{Bmatrix}$$

$$\text{cost} \geq B_{\text{City 4}}(x) = \begin{Bmatrix} \$350 & \text{if } x = \text{City 1} \\ \$0 & \text{otherwise} \end{Bmatrix}$$

The solution is x = City 1, with cost \$330. Because we found a feasible route with this cost, we are done.

General Analysis of Logic-Based Benders

(Hooker & Ottosson, 1999; Hooker 2000)

$$\begin{aligned} \min_{x,y} \quad & f(x, y) \\ \text{s.t.} \quad & C(x,y) \end{aligned}$$

Secondary variables

Primary variables

For a given value $\bar{x}$ of x, solve the subproblem:

$$\begin{aligned} \min_{y} \quad & f(\bar{x}, y) \\ \text{s.t.} \quad & C(\bar{x},y) \end{aligned}$$

Let y^* be an optimal solution with optimal value v^*. To find a Benders cut, consider the *inference dual:*

$$\begin{aligned} \max_{y,v} \quad & v \\ \text{s.t.} \quad & C(\bar{x}, y) \rightarrow f(\bar{x}, y) \geq v \end{aligned}$$

The inference dual clearly has the same optimal value v^*.

The solution of the inference dual is a *proof* that $f(\bar{x}, y) \geq v^*$ follows from $C(\bar{x}, y)$

Thus when $x = \bar{x}$ we have a proof that $f(x, y)$ is at least v^*

We want to use this *same proof schema* to derive that $f(x, y)$ is at least $B_{y^*}(x)$ for any x. (In particular $B_{y^*}(\bar{x}) = v^*$.)

To find a better solution than v^* we solve the *master problem*

$$\begin{array}{ll} \min\limits_{x,v} & v \\ \text{s.t.} & v \geq B_{x^*}(x) \end{array}$$

Benders cut

At iteration K+1 the master problem is

$$\begin{aligned} \min_{x,v} \quad & v \\ \text{s.t.} \quad & v \geq B_{x^k}(x),\, k = 0,\ldots,K \end{aligned}$$

Where $x^1,\ldots,x^K$ are the solutions of the first K master problems.

Continue until the subproblem has the same optimal value as the previous master problem.

Classical Benders Decomposition *(Benders 1962)*

$$\begin{aligned} \min_{x,y} \quad & f(x) + cy \\ \text{s.t.} \quad & g(x) + Ay \geq a \\ & y \geq 0 \\ & x \in D,\ y \in R^n \end{aligned}$$

For a given $\bar{x}$ the subproblem is the LP

$$\begin{aligned} \min_{y} \quad & f(\bar{x}) + cy \\ \text{s.t.} \quad & Ay \geq a - g(\bar{x}) \quad (u) \\ & y \geq 0 \end{aligned}$$

Dual variables

With optimal solution x^* and optimal value v^*:

The inference dual in this case is the classical LP dual

$$\begin{array}{ll} \min\limits_{u} & u(a-g(\bar{x})) \\ \text{s.t.} & uA \le c \\ & u \ge 0 \end{array}$$

The dual solution u^* provides a proof that z^* is the optimal value: the linear combination $u^*Ay \ge u^*(a-g(\bar{x}))$ of the primal constraints dominates $cy \ge z^*$

Note that u^* is *dual feasible for any* x. So by weak duality, $u^*(a-g(x))$ is a lower bound on the optimal value of the subproblem for any x. So we have the Benders cut,

$$v \ge f(x) + u^*(a-g(x))$$

The master problem is

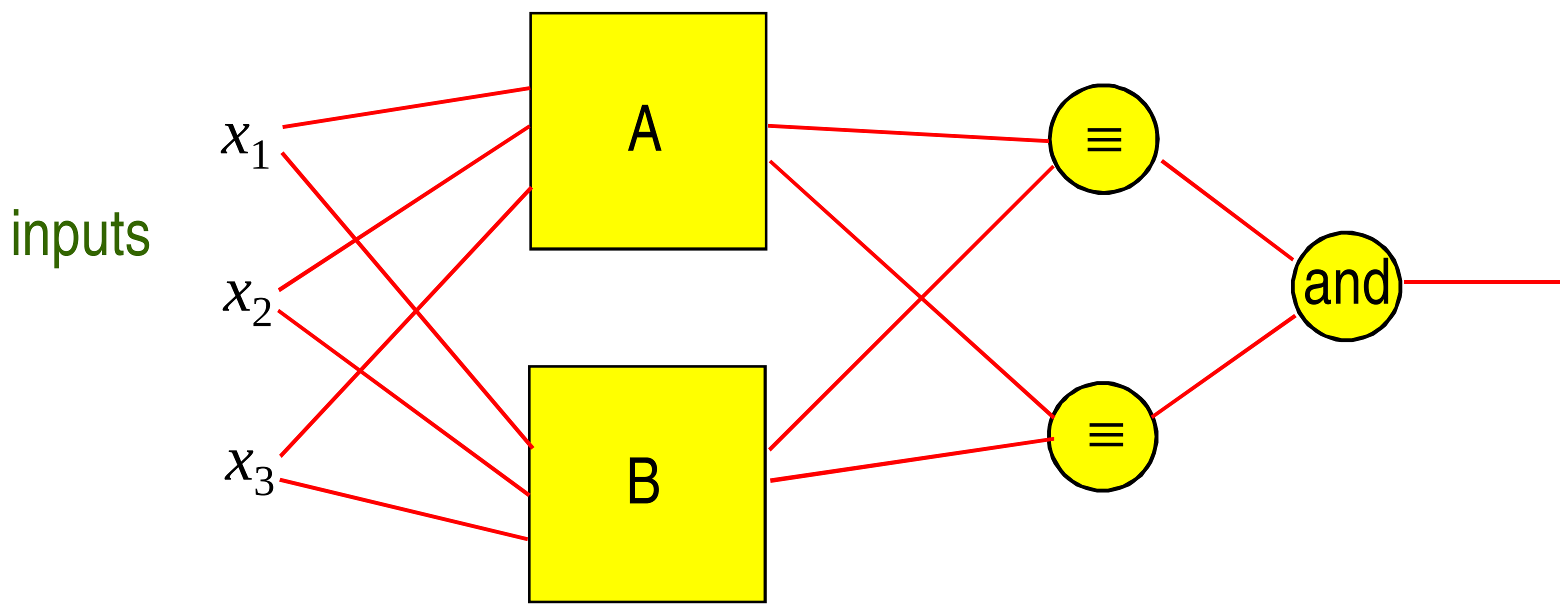
$$\begin{aligned} \min_{x} \quad & v \\ \text{s.t.} \quad & v \geq f(x) + u^k(a - g(x)),\ k = 0, \ldots, K \end{aligned}$$

Where $u^1, \ldots, u^k$ are the solutions of the first K subproblem duals. The case of an infeasible subproblem requires special treatment.

Logic circuit verification

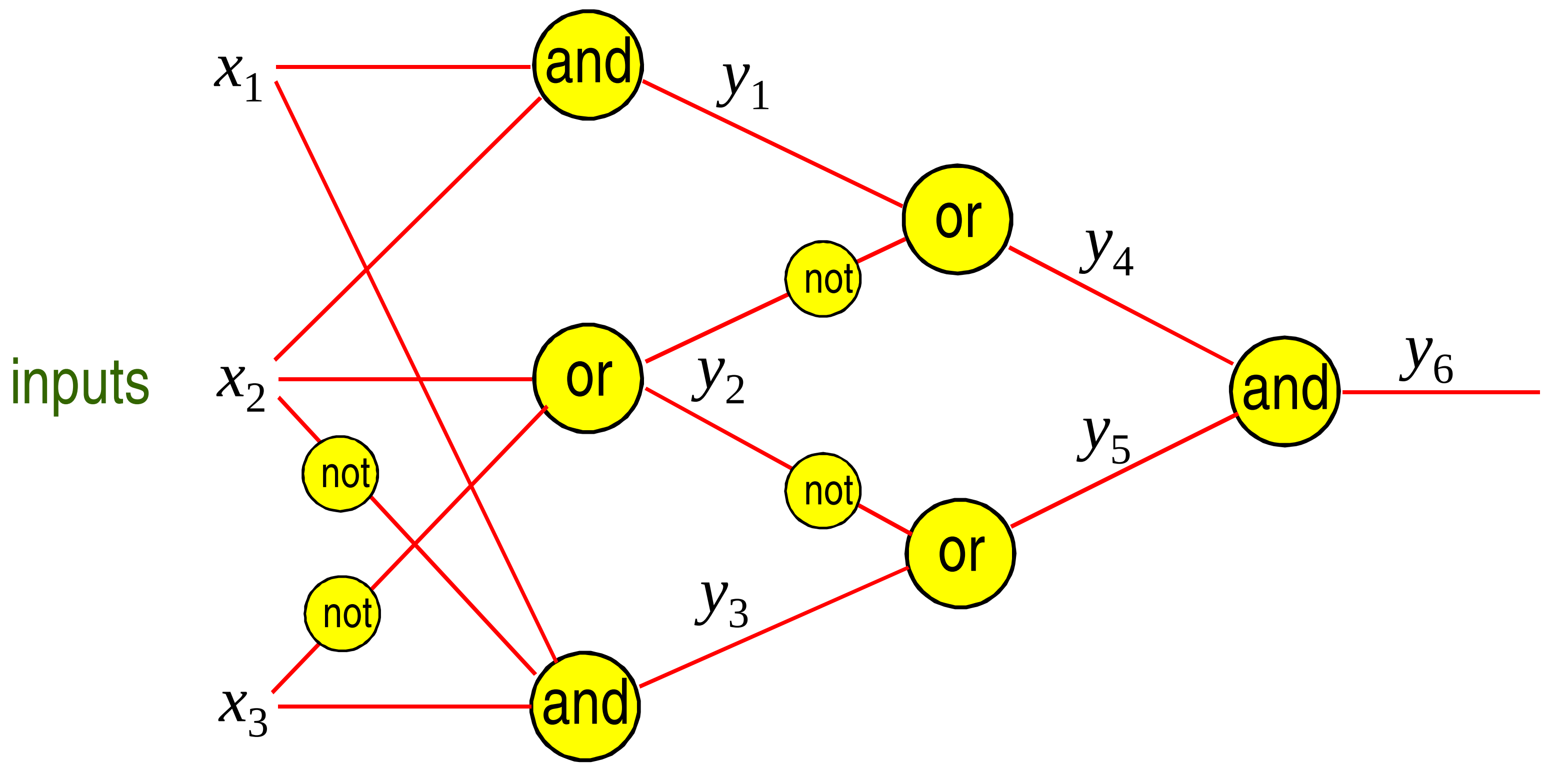
(Hooker & Yan 1994; Hooker 1999)

Logic circuits A and B are equivalent when the following circuit is a tautology:



The circuit is a tautology if the minimum output over all 0-1 inputs is 1.

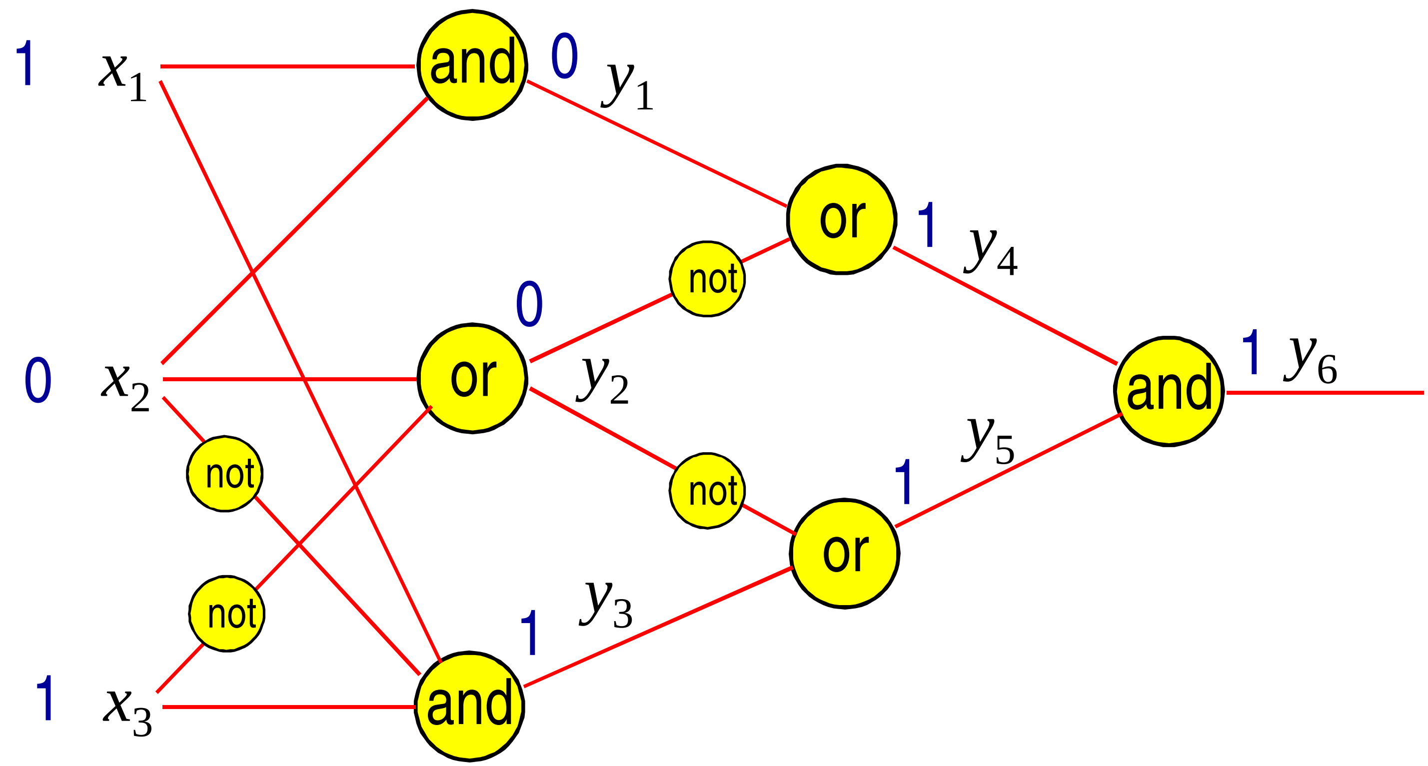
For instance, check whether this circuit is a tautology:



The subproblem is to minimize the output when the input *x* is fixed to a given value.

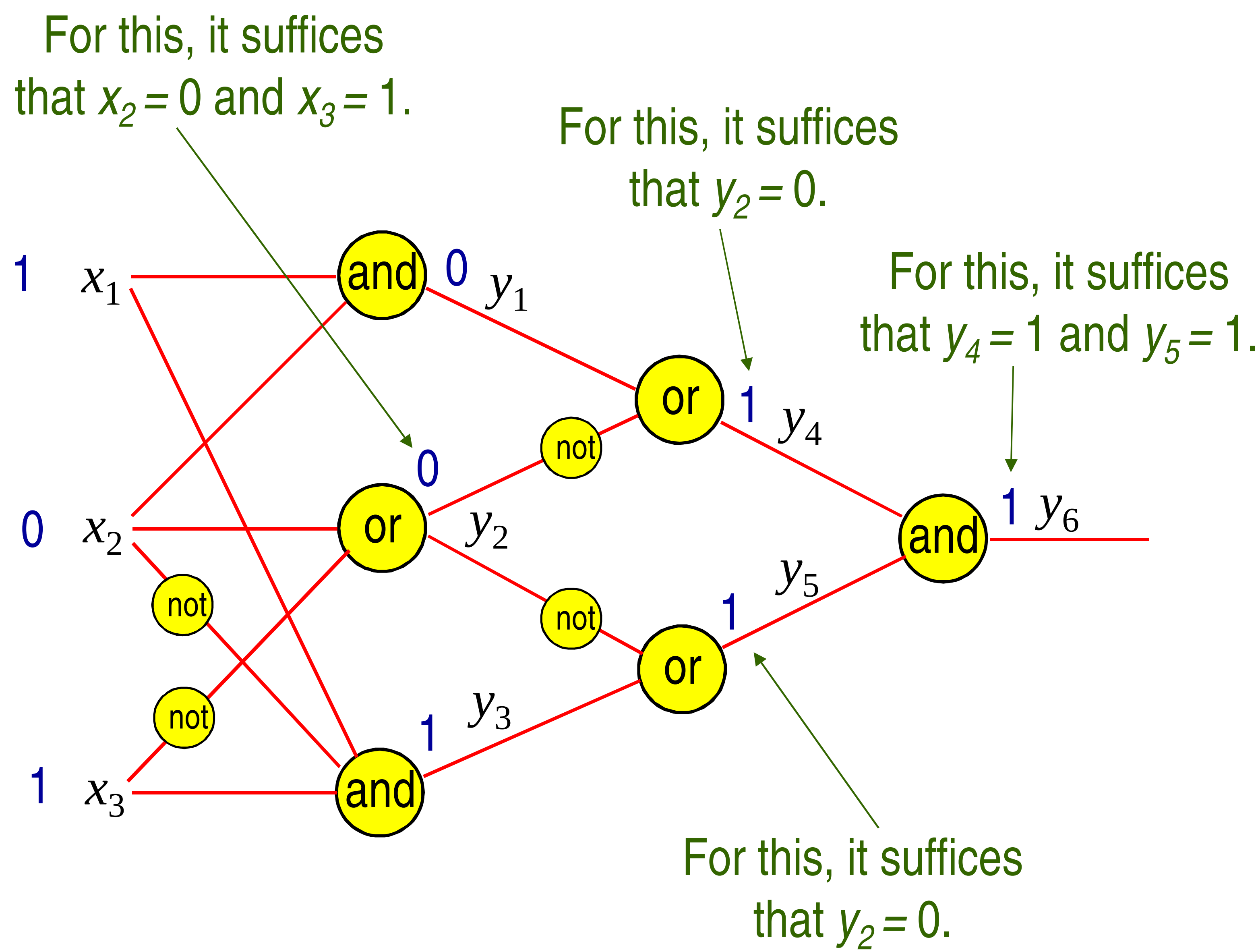
But since *x* determines the output of the circuit, the subproblem is easy: just compute the output.

For example, let $x = (1,0,1)$.



To construct a Benders cut, identify which subsets of the inputs are sufficient to generate an output of 1.

For instance, $(x_2, x_3) = (0,1)$ suffices.



So, Benders cut is $v \geq (\text{not } x_2) \text{ and } x_3$

Now solve the master problem

$$\begin{array}{ll} \min & v \\ \text{s.t.} & v \geq (\text{not } x_2) \text{ and } x_3 \end{array}$$

One solution is $(x_1, x_2, x_3) = (0,0,0)$

This produces output 0, which shows the circuit is not a tautology.

Note: This is actually a case of classical Benders. The subproblem can be written as an LP (a Horn-SAT problem).

Computational results:

Compare with Binary Decision Diagrams (BDDs), state-of-the-art exact method.

- When A and B are equivalent (the circuit is a tautology), BDDs are usually much better.
- When A and B are not equivalent (one contains an error), the Benders approach is usually much better.

Combining MILP with Constraint Programming:

Machine scheduling *(Hooker 2000, Jain & Grossmann 2001)*

- Assign each job to one machine so as to process all jobs at minimum cost. Machines run at different speeds and incur different costs per job. Each job has a release date and a due date.

- In this problem, the master problem assigns jobs to machines. The subproblem schedules jobs assigned to each machine.

- Classical mixed integer programming solves the master problem.

- Constraint programming solves the subproblem, a 1-machine scheduling problem with time windows.

- This provides a general framework for combining mixed integer programming and constraint programming.

A model for the problem:

Cost of assigning machine x_j to job j

Release date for job j

Job duration

Deadline

$$\begin{aligned} \min \quad & \sum_j C_{x_j j} \\ \text{s.t.} \quad & t_j \ge R_j, \quad \text{all } j \\ & t_j + D_{x_j j} \le S_j, \quad \text{all } j \\ & \text{cumulative}\big((t_j \mid x_j = i), (D_{ij} \mid x_j = i), e, 1\big), \quad \text{all } i \end{aligned}$$

Start time for job j

Machine assigned to job j

Start times of jobs assigned to machine i

For a given set of assignments $\bar{x}$ the subproblem is the set of 1-machine problems,

$$\begin{array}{ll} \min & 0 \\ \text{s.t.} & \text{cumulative}\big((t_j \mid \bar{x}_j = i), (D_{ij} \mid \bar{x}_j = i), e, 1\big), \quad \text{all } i \end{array}$$

Feasibility of each problem is checked by constraint programming.
One or more infeasible problems results in an optimal value ∞.
Otherwise the value is zero.

Suppose there is no feasible schedule for machine i. Then jobs $\{j \mid \bar{x}_j = i\}$ cannot all be assigned to machine i.

Suppose in fact that some subset $J_i(\bar{x})$ of these jobs cannot be assigned to machine i. Then we have a Benders cut

$$v \geq B_{\bar{x}}(x) = \left\{ \begin{matrix} \infty & \text{if } x_j = i \text{ for all } j \in J_i(\bar{x}) \\ 0 & \text{otherwise} \end{matrix} \right\}$$

Equivalently, just add the constraint

$$x_j \neq i \text{ for some } j \in J_i(\bar{x})$$

This yields the master problem,

$$
\begin{aligned}
\min \quad & \sum_j C_{x_j j} \\
\text{s.t.} \quad & t_j \geq R_j, \quad \text{all } j \\
& t_j + D_{x_j j} \leq S_j, \quad \text{all } j \\
& x_j \neq i \text{ for some } j \in J_i(x^k), \ \text{all } i, \ k = 1, \ldots, K
\end{aligned}
$$

This problem can be written as a mixed 0-1 problem:

$$
\begin{aligned}
\min \quad & \sum_{ij} C_{ij} y_{ij} \\
\text{s.t.} \quad & t_j \ge R_j, \quad \text{all } j \\
& t_j + \sum_i D_{ij} y_{ij} \le S_j, \quad \text{all } j \\
& \sum_i y_{ij} \ge 1, \quad \text{all } j \\
& \sum_{\substack{j \\ x_j^k = i}} (1 - y_{ij}) \ge 1, \quad \text{all } i, \ k = 1, \ldots, K \\
& \sum_j D_{ij} y_{ij} \le \max_j \{S_j\} - \min_j \{R_j\}, \quad \text{all } i \\
& y_{ij} \in \{0,1\}
\end{aligned}
$$

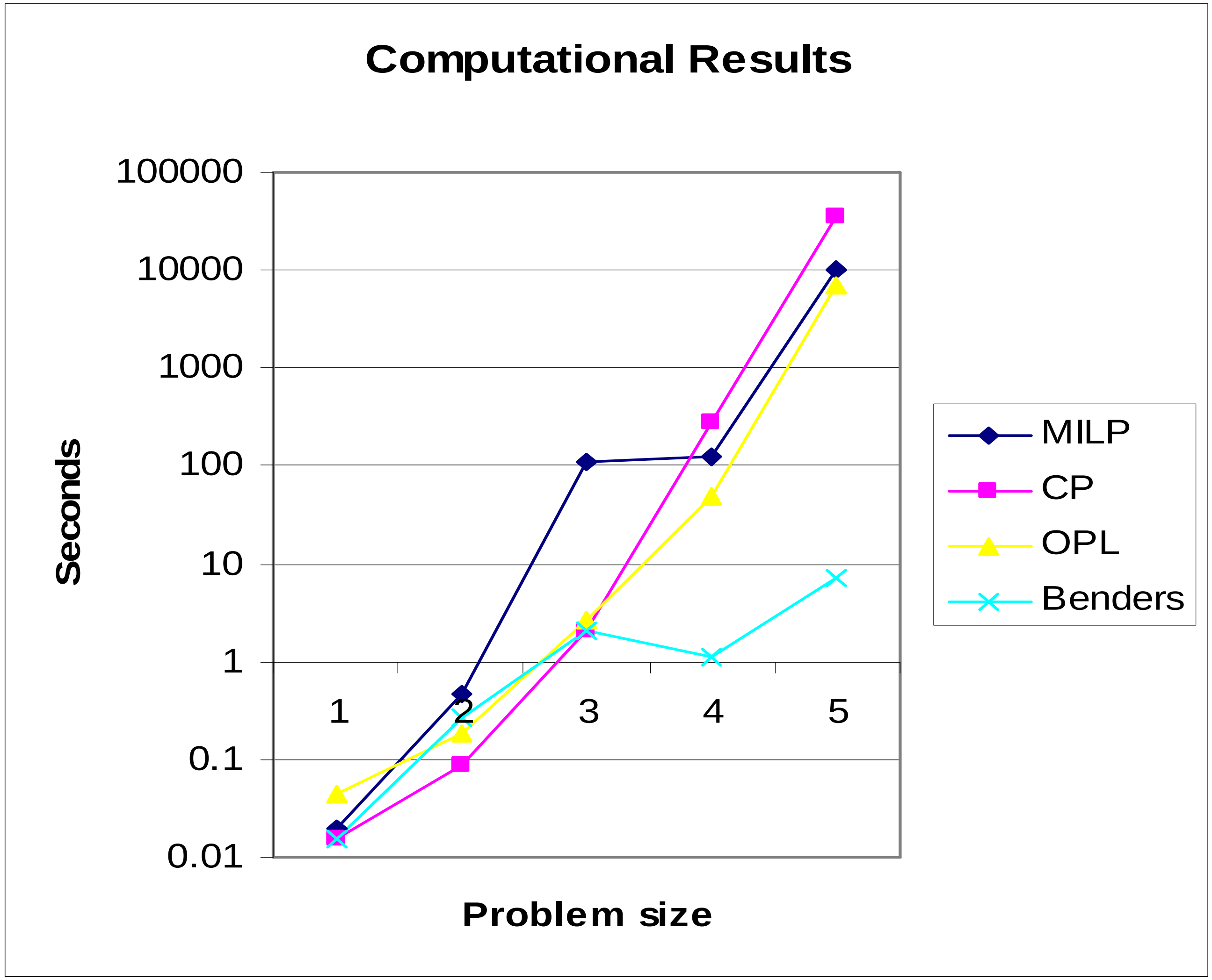
Valid constraint added to improve performance

Computational Results *(Jain & Grossmann 2001)*

Problem sizes
(jobs, machines)
1 - (3,2)
2 - (7,3)
3 - (12,3)
4 - (15,5)
5 - (20,5)

Each data point represents an average of 2 instances

MILP and CP ran out of memory on 1 of the largest instances



An Enhancement: Branch and Check

(Hooker 2000, Thorsteinsson 2001)

- Generate a Benders cut whenever a feasible solution $\bar{x}$ is found in the master problem tree search.
- Keep the cuts (essentially nogoods) in the problem for the remainder of the tree search.
- Solve the master problem only once but continually update it.
- This was applied to the machine scheduling problem described earlier.

Computational results *(Thorsteinsson 2001)*

Computation times in seconds
Problems have 30 jobs, 7 machines.

Problem	Benders	Branch and check
1	16.2	1.2
2	93.7	10.9
3	120.2	1.0
4	37.2	3.0
5	30.2	1.2

References

J. F. Benders, 1962. Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik* **4**, 238-252.

J. N. Hooker, 2000. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*, John Wiley & Sons.

J. N. Hooker and G. Ottosson, 1999. Logic-based Benders decomposition, submitted for publication.

J. N. Hooker and Hong Yan, 1995. Logic circuit by Benders decomposition, in V. Saraswat and P. Van Hentenryck, eds., *Principles and Practice of Constraint Programming: The Newport Papers*, MIT Press, 267-288.

V. Jain and I. E. Grossmann, 2001. Algorithms for hybrid MILP/CLP models for a class of optimization problems, to appear in *INFORMS Journal on Computing*.

E. Thorsteinsson, 2001. Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming, manuscript, Carnegie Mellon Univ.