A Framework for Combining Solution Methods

John Hooker

Carnegie Mellon University

International Conference on Operations Research

Havana, September 2003

A Simple Knapsack Example
Integrating CP and MP, and Local Search
Branch Infer and Relax
Decomposition
Recent Success Stories

A Simple Knapsack Example

Solution by Constraint Programming
Solution by Integer Programming
Solution by a Hybrid Method

The Problem

$$
\begin{aligned}
\min \quad & 5x_1 + 8x_2 + 4x_3 \\
\text{subject to} \quad & 3x_1 + 5x_2 + 2x_3 \geq 30 \\
& \text{all-different}\{x_1, x_2, x_3\} \\
& x_j \in \{1, \ldots, 4\}
\end{aligned}
$$

We will illustrate how search, inference and relaxation may be combined to solve this problem by:

- constraint programming
- integer programming
- a hybrid approach

Solve as a constraint programming problem

Search: Domain splitting
Inference: Domain reduction
Relaxation: Constraint store (set of current variable domains)

Start with z = ∞.
Will decrease as feasible solutions are found.

$$5x_1 + 8x_2 + 4x_3 \leq z$$
$$3x_1 + 5x_2 + 2x_3 \geq 30$$
$$\text{all-different}\{x_1, x_2, x_3\}$$
$$x_j \in \{1, \ldots, 4\}$$

Global constraint

Constraint store can be viewed as consisting of in-domain constraints $x_j \in D_j$, which form a relaxation of the problem.

Domain reduction for inequalities

- Bounds propagation on $5x_1 + 8x_2 + 4x_3 \leq z$
  $3x_1 + 5x_2 + 2x_3 \geq 30$

For example, $3x_1 + 5x_2 + 2x_3 \geq 30$ implies

$$x_2 \geq \frac{30 - 3x_1 - 2x_3}{5} \geq \frac{30 - 12 - 8}{5} = 2$$

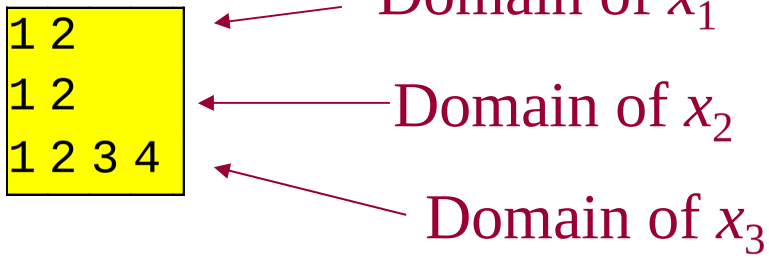
So the domain of x_2 is reduced to $\{2,3,4\}$.

Domain reduction for all-different *(e.g., Régin)*

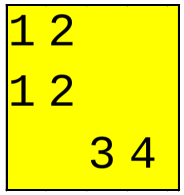
- Maintain hyperarc consistency on

$$\text{all-different}\{x_1, x_2, x_3\}$$

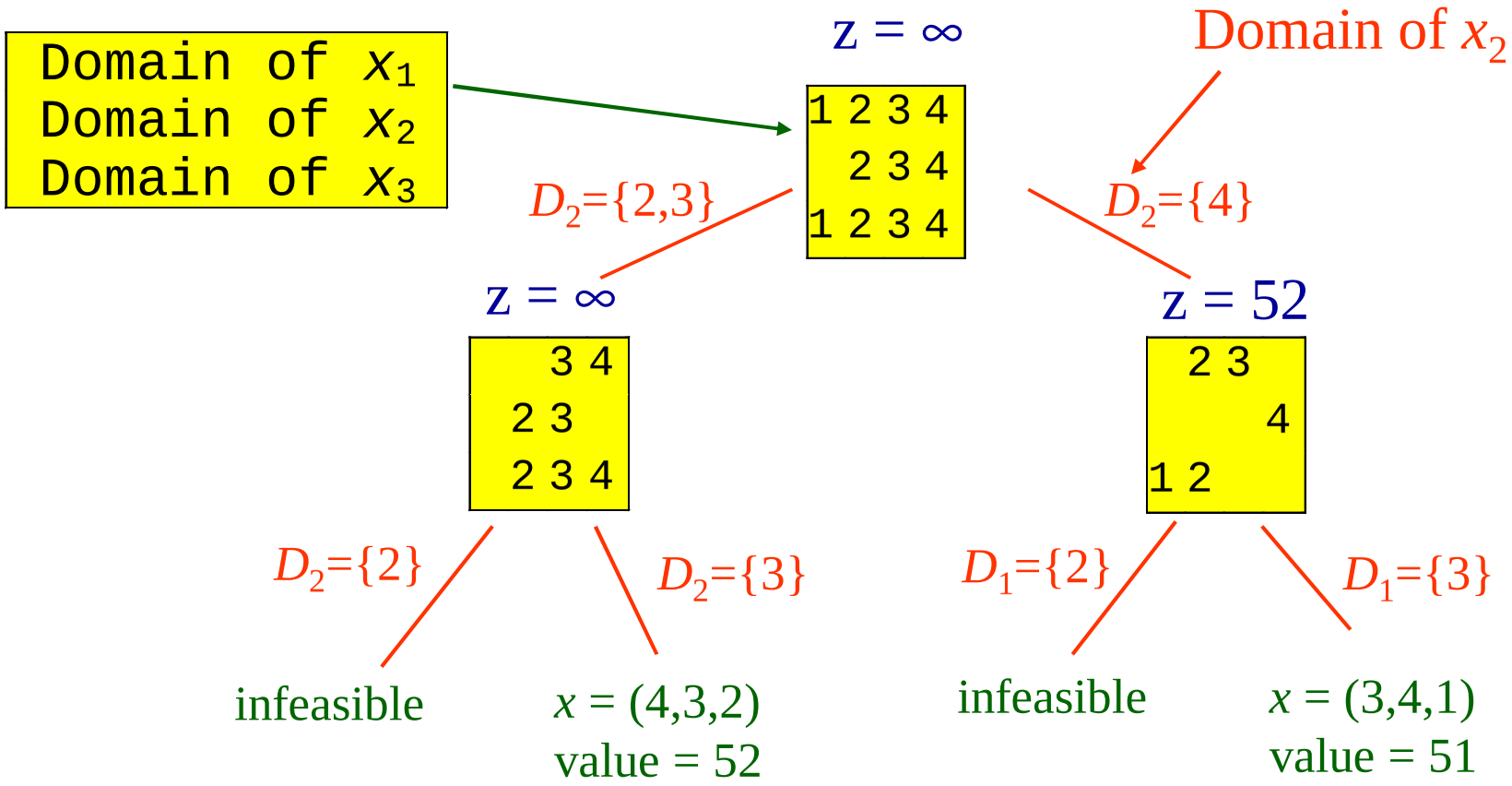
Suppose for example:



Then one can reduce the domains:



- In general, solve a maximum cardinality matching problem and apply a theorem of Berge


Domain of x_1
Domain of x_2
Domain of x_3
z = ∞
1 2 3 4
2 3 4
1 2 3 4
Domain of x_2
D_2={2,3}
D_2={4}
z = ∞
3 4
2 3
2 3 4
z = 52
2 3
4
1 2
D_2={2}
D_2={3}
D_1={2}
D_1={3}
infeasible
x = (4,3,2)
value = 52
infeasible
x = (3,4,1)
value = 51

Solve as an integer programming problem

Search: Branch on variables with fractional values in solution of continuous relaxation.
Inference: Generate cutting planes (covering inequalities).
Relaxation: Continuous (LP) relaxation.

Rewrite problem using integer programming model:

Let y_{ij} be 1 if $x_i = j$, 0 otherwise.

$$\begin{aligned}
\text{min} \quad & 5x_1 + 8x_2 + 4x_3 \\
\text{subject to} \quad & 3x_1 + 5x_2 + 2x_3 \geq 30 \\
& x_i = \sum_{j=1}^{5} j y_{ij}, \quad i = 1,2,3 \\
& \sum_{j=1}^{4} y_{ij} = 1, \quad i = 1,2,3 \\
& \sum_{i=1}^{3} y_{ij} \leq 1, \quad j = 1,\ldots,4 \\
& y_{ij} \in \{0,1\}, \quad \text{all } i, j
\end{aligned}$$

Continuous relaxation

$$\begin{aligned}
\text{min} \quad & 4x_1 + 3x_2 + 5x_3 \\
\text{subject to} \quad & 4x_1 + 2x_2 + 4x_3 \geq 17 \\
& x_i = \sum_{j=1}^{4} j y_{ij}, \quad i = 1,2,3 \\
& \sum_{j=1}^{4} y_{ij} = 1, \quad i = 1,2,3 \\
& \sum_{i=1}^{3} y_{ij} \leq 1, \quad j = 1,\ldots,4
\end{aligned}$$

Covering inequalities

$$\begin{aligned}
& x_1 + x_2 \geq 5 \\
& x_1 + x_3 \geq 4 \\
& x_2 + x_3 \geq 4 \\
& x_1 + x_2 + x_3 \geq 8
\end{aligned}$$

Relax integrality $\quad 0 \leq y_{ij} \leq 1, \quad \text{all } i, j$

Branch and bound (Branch and relax)

The *incumbent solution* is the best feasible solution found so far.

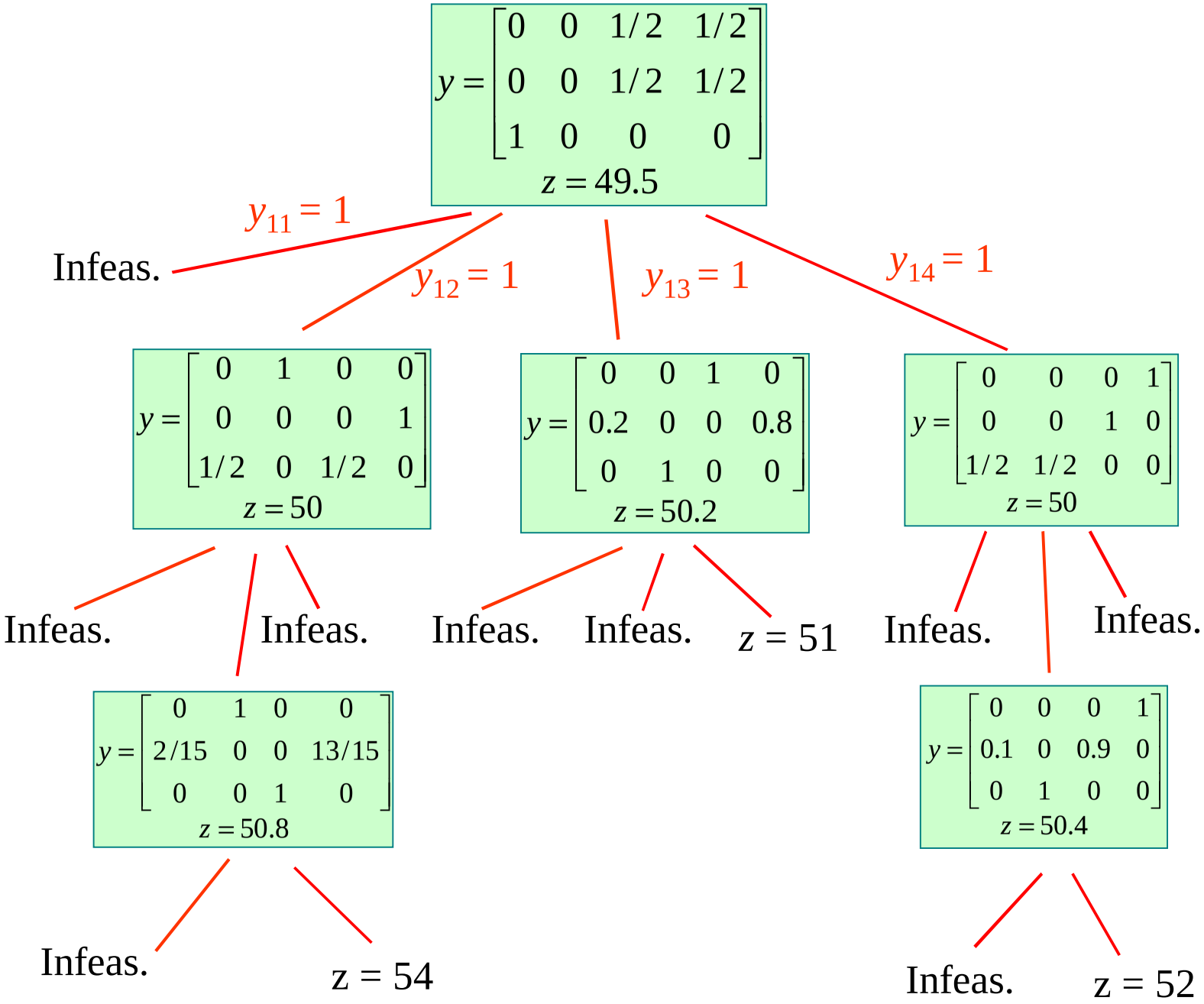
At each node of the branching tree:

- If Optimal value of relaxation $\geq$ Value of incumbent solution

There is no need to branch further.

- No feasible solution in that subtree can be better than the incumbent solution.

- Use SOS-1 branching.


$y = \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \end{bmatrix}$
$z = 49.5$
$y_{11} = 1$
Infeas.
$y_{12} = 1$
$y_{13} = 1$
$y_{14} = 1$
$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix}$
$z = 50$
$y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0.2 & 0 & 0 & 0.8 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
$z = 50.2$
$y = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix}$
$z = 50$
Infeas.
Infeas.
Infeas.
Infeas.
$z = 51$
Infeas.
Infeas.
$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2/15 & 0 & 0 & 13/15 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
$z = 50.8$
$y = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.1 & 0 & 0.9 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
$z = 50.4$
Infeas.
z = 54
Infeas.
z = 52

Solve using a hybrid approach

Search:

- Branch on fractional variables in solution of relaxation.
  - Drop constraints with y_{ij}'s. This makes relaxation too large without much improvement in quality.
  - If variables are all integral, branch by splitting domain.
- Use branch and bound.

Inference:

- Use bounds propagation for all inequalities.
- Maintain hyperarc consistency for all-different constraints.

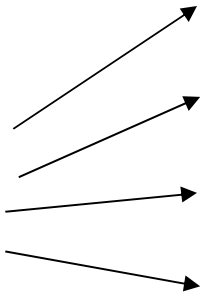
Relaxation:

- Put knapsack constraint in LP.
- Put covering inequalities based on knapsack/all-different into LP.

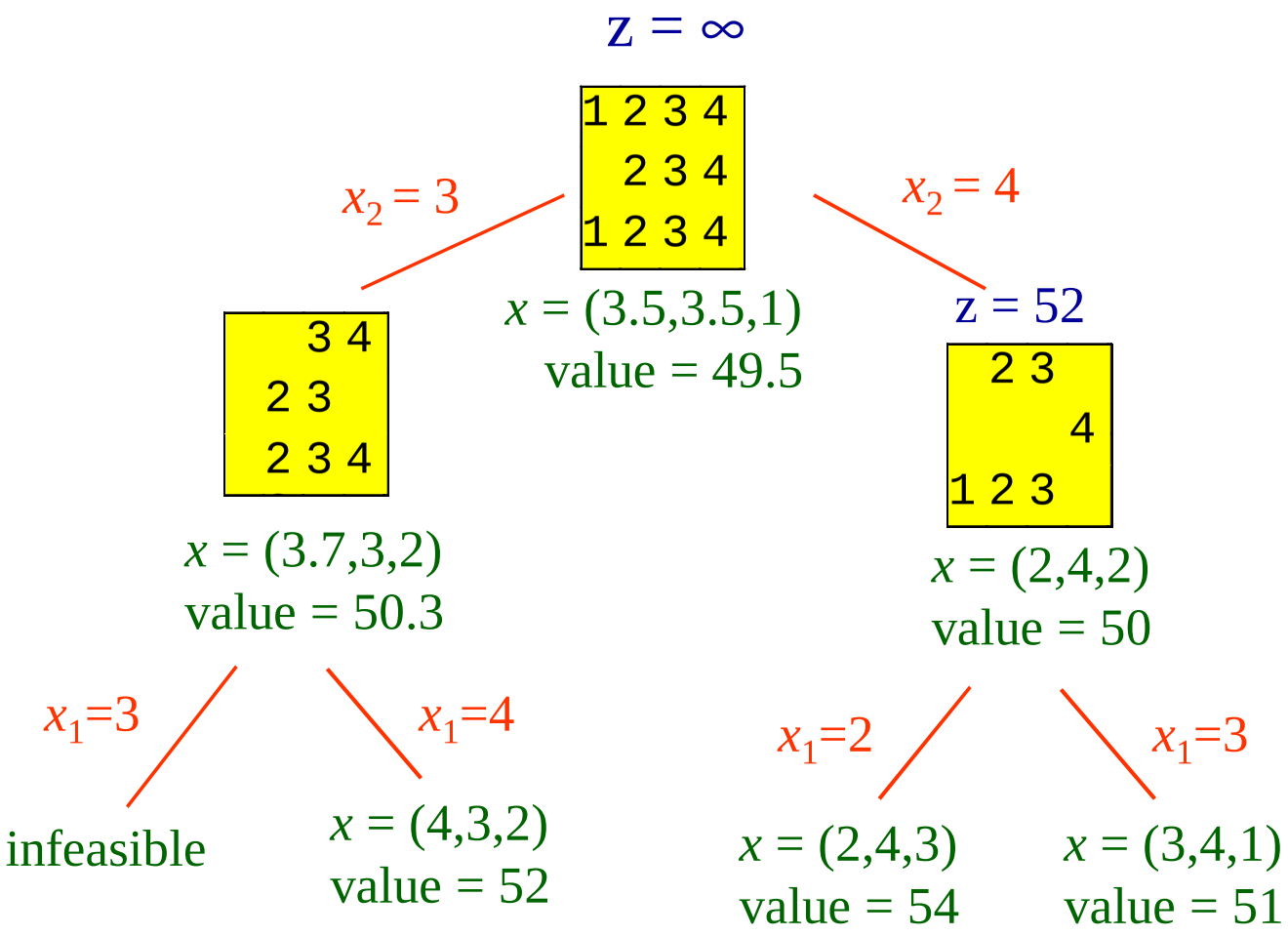
Model for hybrid approach

$$\begin{aligned}
\min \quad & 5x_1 + 8x_2 + 4x_3 \le z \\
\text{s.t.} \quad & 3x_1 + 5x_2 + 2x_3 \ge 30 \\
& \text{all-different}\{x_1, x_2, x_3\} \\
& x_1 + x_2 \ge 5 \\
& x_1 + x_3 \ge 4 \\
& x_2 + x_3 \ge 4 \\
& x_1 + x_2 + x_3 \ge 8 \\
& x_j \in \{1, \ldots, 4\}
\end{aligned}$$

Covering inequalities



Generate and propagate covering inequalities at each node of search tree


z = ∞
1 2 3 4
2 3 4
1 2 3 4
x = (3.5,3.5,1)
value = 49.5
x2 = 3
x2 = 4
3 4
2 3
2 3 4
x = (3.7,3,2)
value = 50.3
z = 52
2 3
4
1 2 3
x = (2,4,2)
value = 50
x1=3
x1=4
infeasible
x = (4,3,2)
value = 52
x1=2
x1=3
x = (2,4,3)
value = 54
x = (3,4,1)
value = 51

Integrating CP, MP, and Local Search

Motivation

Integration Schemes

Motivation to Integrate CP and MP

- CP's inference techniques tend to be effective when constraints contain few variables.
  - Misleading to say CP is effective on "highly constrained" problems.
- MP's relaxation techniques tend to be effective when constraints or objective function contain many variables.
  - For example, cost and profit.

Integration Schemes

Recent work can be broadly seen as using two integrative ideas:

- *Branch-infer-and-relax:* View CP and MP methods as special cases of a branch-infer-and-relax method.
- *Decomposition:* Decompose problems into a CP part and an MP part, perhaps using a Benders scheme.
- *General scheme:* These are special cases of a general scheme for integrating CP, MP and local search.

Branch-infer-and-relax

- *Branching* – enumerate solutions by branching on variables or violated constraints.
- *Inference* – deduce new constraints
  - CP: domain reduction
  - MP: cutting planes.
- *Relaxation* – remove some constraints before solving
  - CP: the constraint store (variable domains)
  - MP: continuous relaxation

Decomposition

- Some problems can be decomposed into a master problem and subproblem.
  - Master problem searches over some of the variables.
  - For each setting of these variables, subproblem solves the problem over the remaining variables.
  - One scheme is a generalized Benders decomposition.
    - CP is natural for subproblem, which can be seen as an inference (dual) problem.

General Scheme

- Decomposition, Branch-infer-and-relax, and local search are special cases of a general scheme:
  - Enumerate problem restrictions.
    - Leaf nodes of tree.
    - Benders subproblems
    - Local search neighborhoods.
  - Solve relaxation of each restriction.
    - LP relaxation.
    - Benders master problem.
    - Relaxation of neighborhood search

General Scheme

- Solution of relaxation guides search.
  - Fractions in LP solution help determine branching.
  - Solution of Benders master problem determines next subproblem.
  - Solution of neighborhood search is center of next neighborhood.

Branch Infer and Relax

The knapsack example illustrated this.

Decomposition

Idea Behind Benders Decomposition
Machine Scheduling

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*).
- Can be viewed as solving a subproblem to generate *Benders cuts* or "nogoods."
- Add the Benders cut to the master problem to require next solution to be better than last, and re-solve.

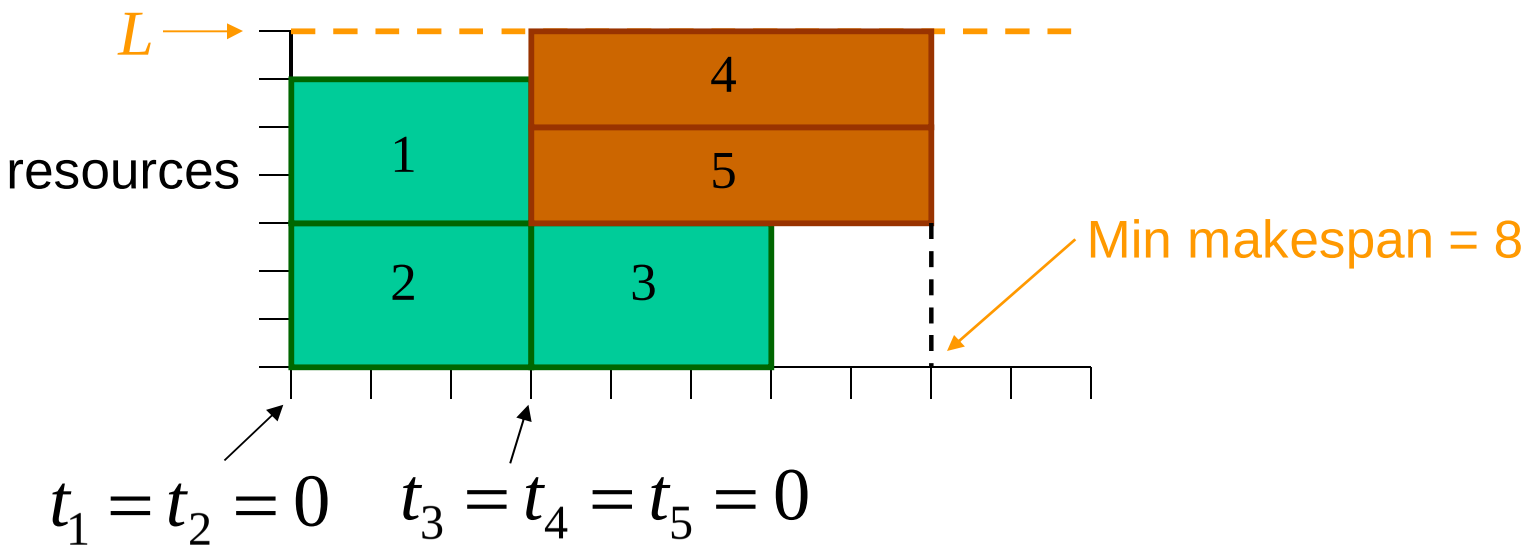
Machine scheduling

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.

Modeling resource-constrained scheduling with *cumulative*

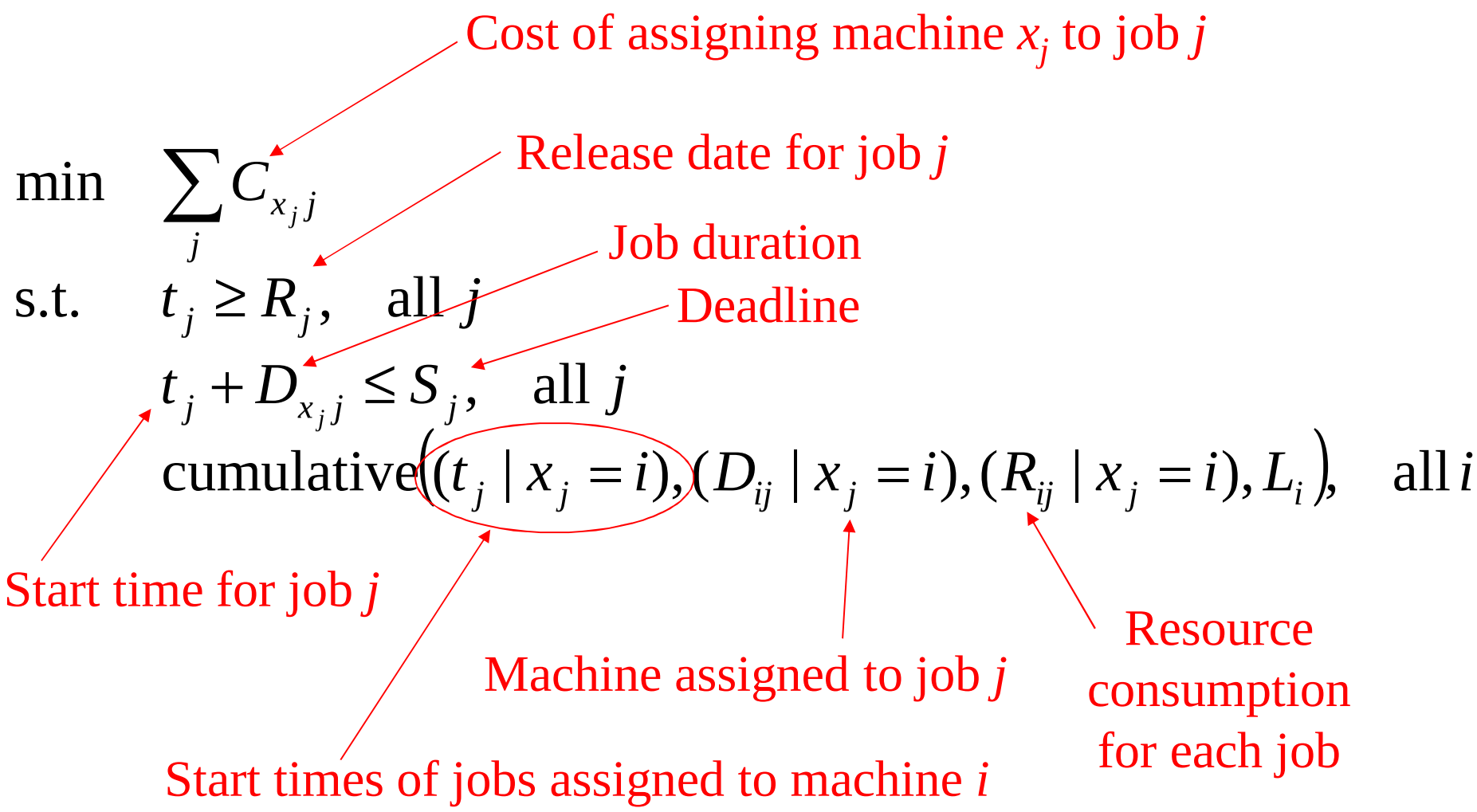
Jobs 1,2,3 consume 3 units of resources.
Jobs 4,5 consume 2 units.
Maximum $L = 7$ units of resources available.



$$\text{cumulative}\big((t_1,\ldots,t_5),(3,3,3,5,5),(3,3,3,2,2),7\big)$$

Start times — Durations — Resource consumption — L

A model for the machine scheduling problem:



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

$$\text{cumulative}\left((t_j \mid \bar{x}_j = i), (D_{ij} \mid \bar{x}_j = i), (R_{ij} \mid \bar{x}_j = i), L_i\right), \text{all } i$$

Feasibility of each problem is checked by constraint programming.

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

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

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:

$$\min \quad \sum_{ij} C_{ij} y_{ij}$$

$$\text{s.t.} \quad t_j \geq R_j, \quad \text{all } j$$

$$t_j + \sum_i D_{ij} y_{ij} \leq S_j, \quad \text{all } j$$

$$\sum_i y_{ij} \geq 1, \quad \text{all } j$$

$$\sum_j (1 - y_{ij}) \geq 1, \quad \text{all } i, \quad k = 1, \ldots, K$$

$$x_j^k = i$$

Valid constraint added to improve performance →

$$\sum_j D_{ij} y_{ij} \leq \max_j \{S_j\} - \min_j \{R_j\}, \quad \text{all } i$$

$$y_{ij} \in \{0,1\}$$

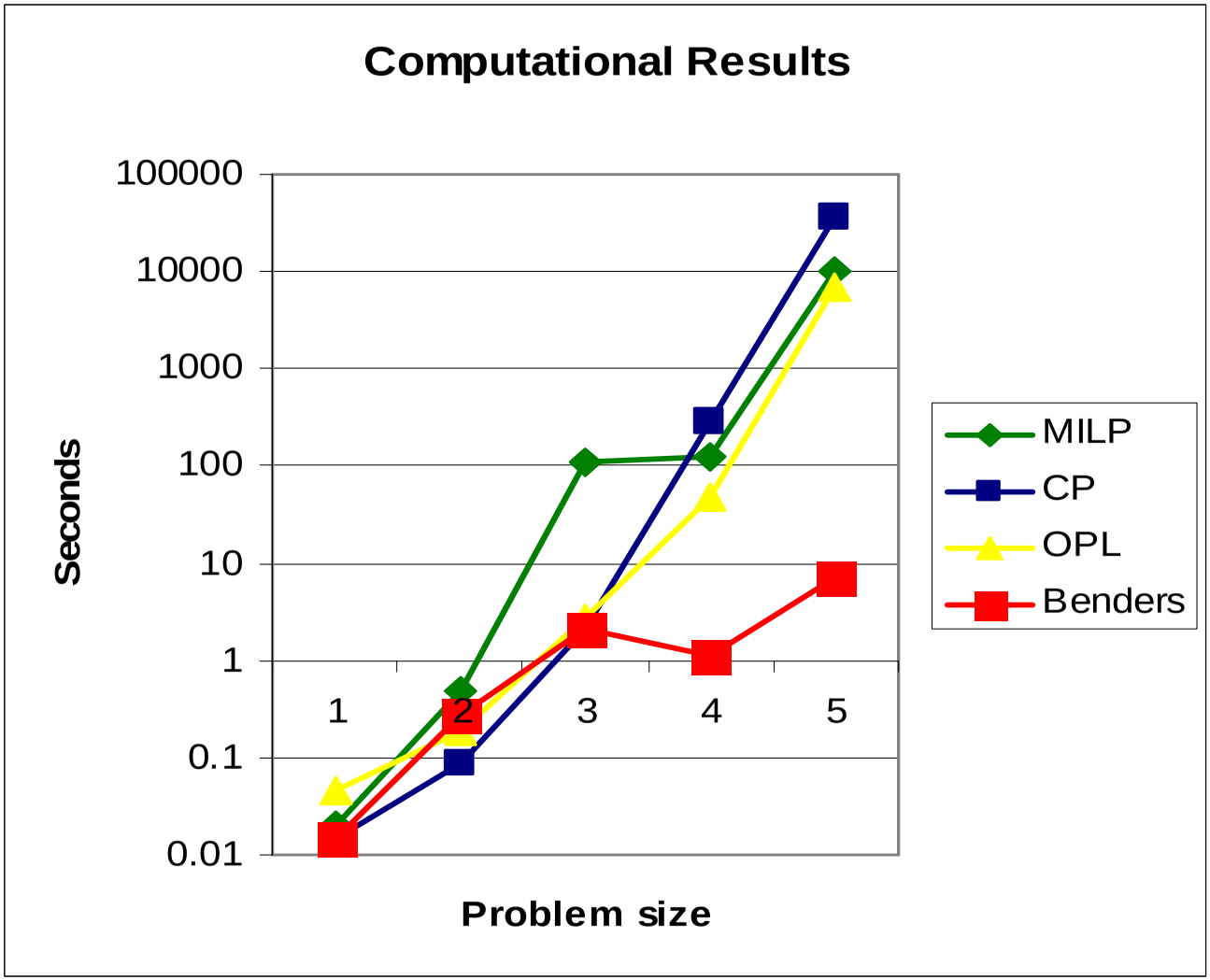
Computational Results *(Jain & Grossmann)*

Subproblems are simple 1-machine scheduling problems

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

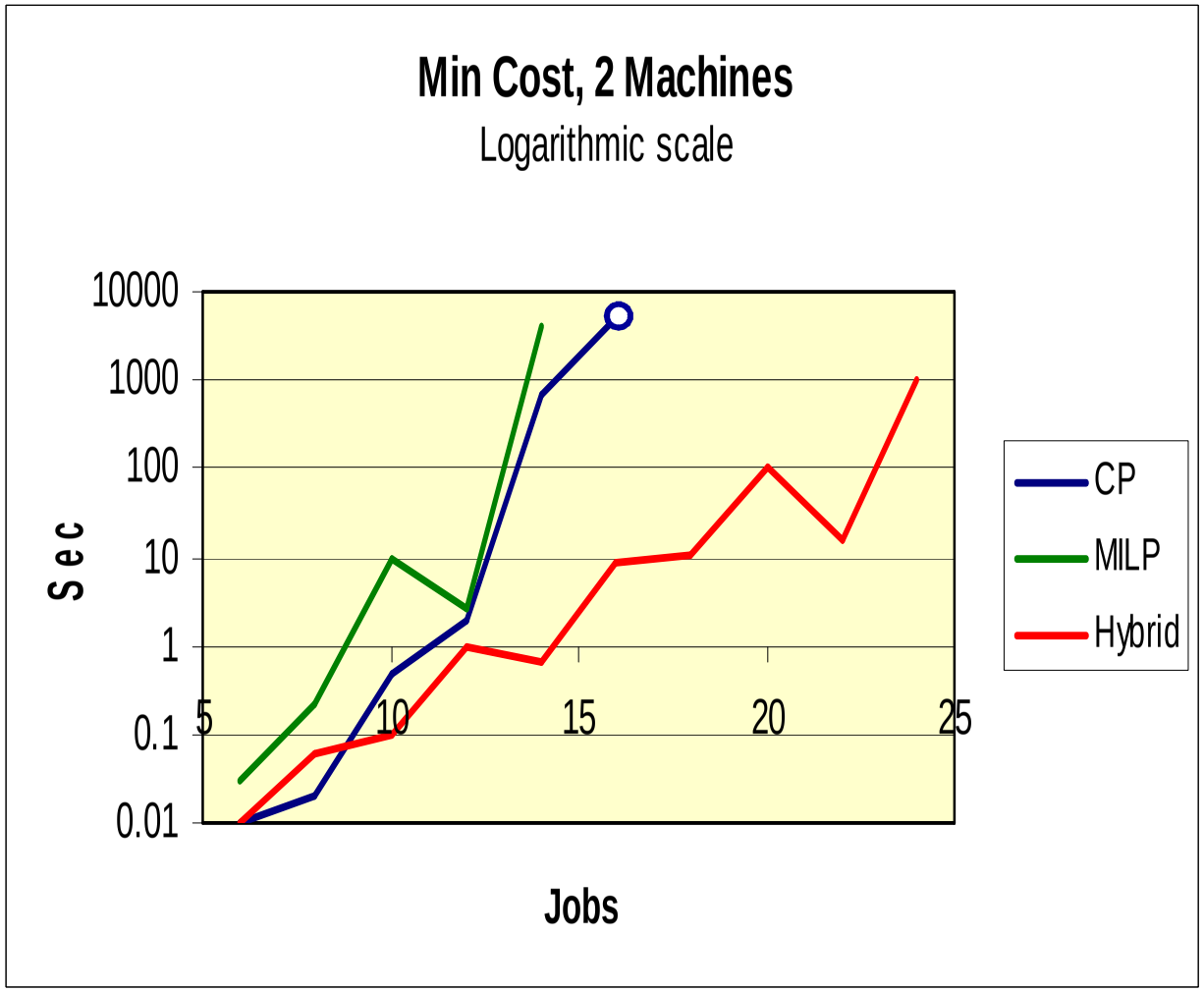
Each data point represents an average of 2 instances

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

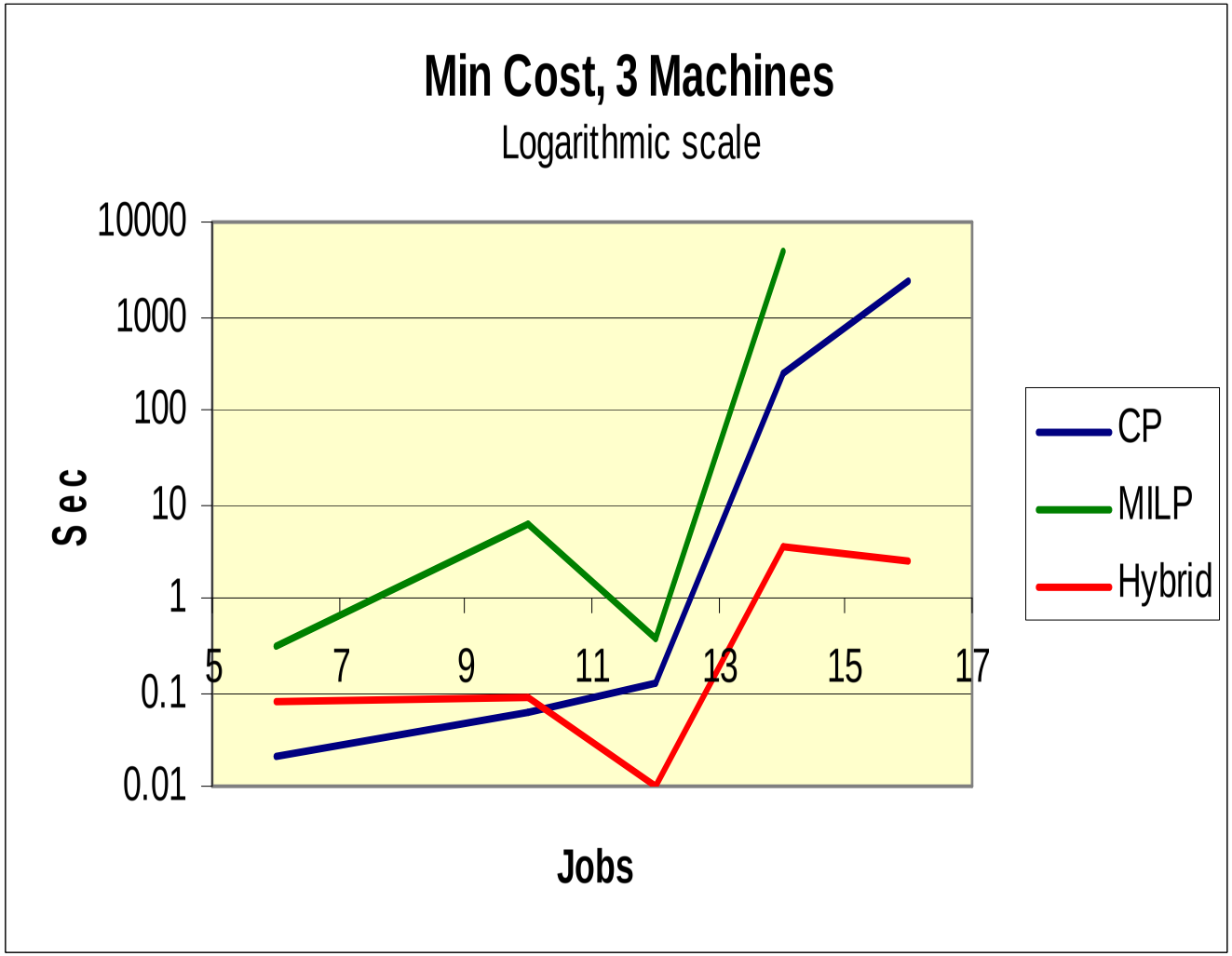


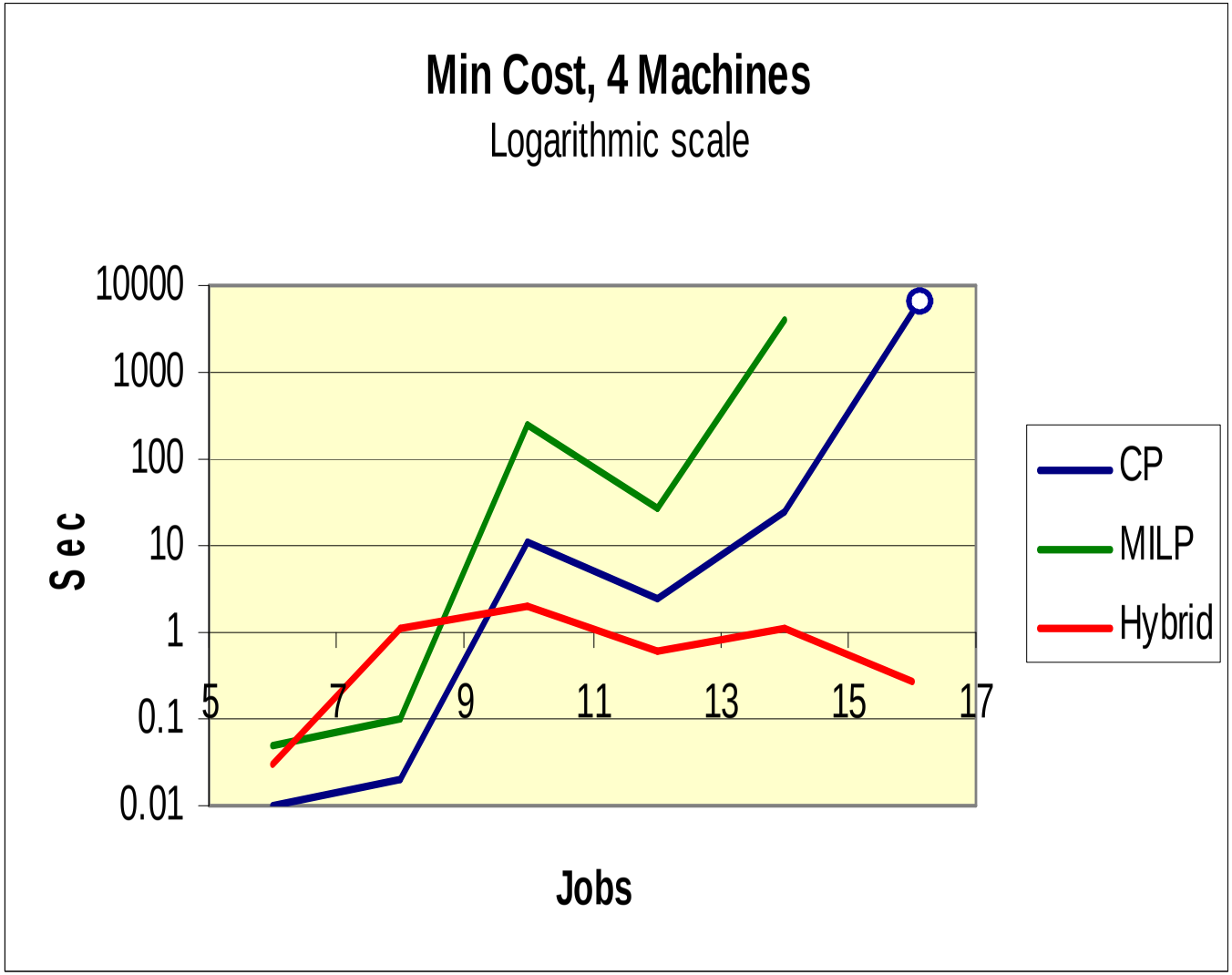
Computational Results *(JNH)*

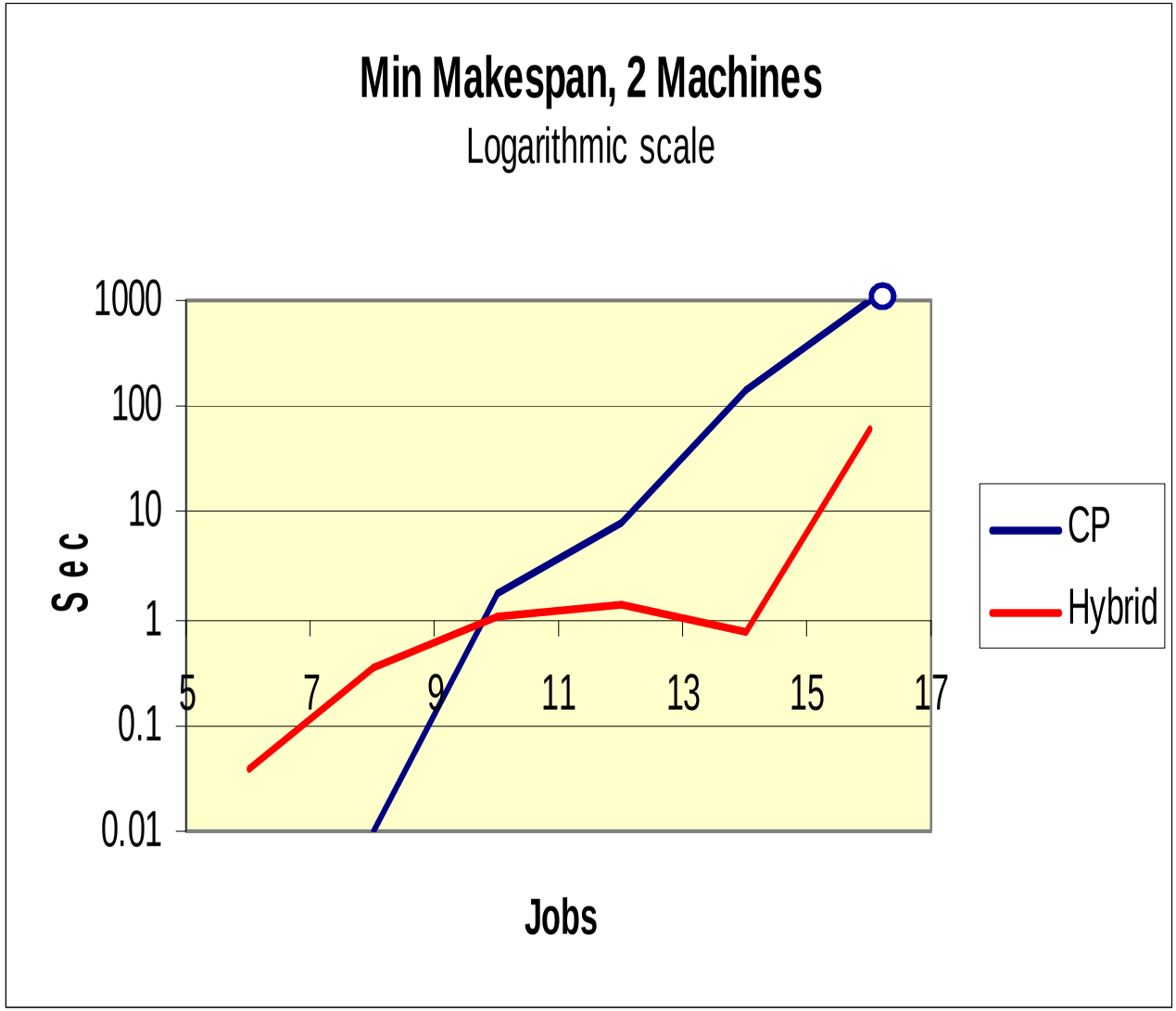
Subproblems are full resource-constrained scheduling problems

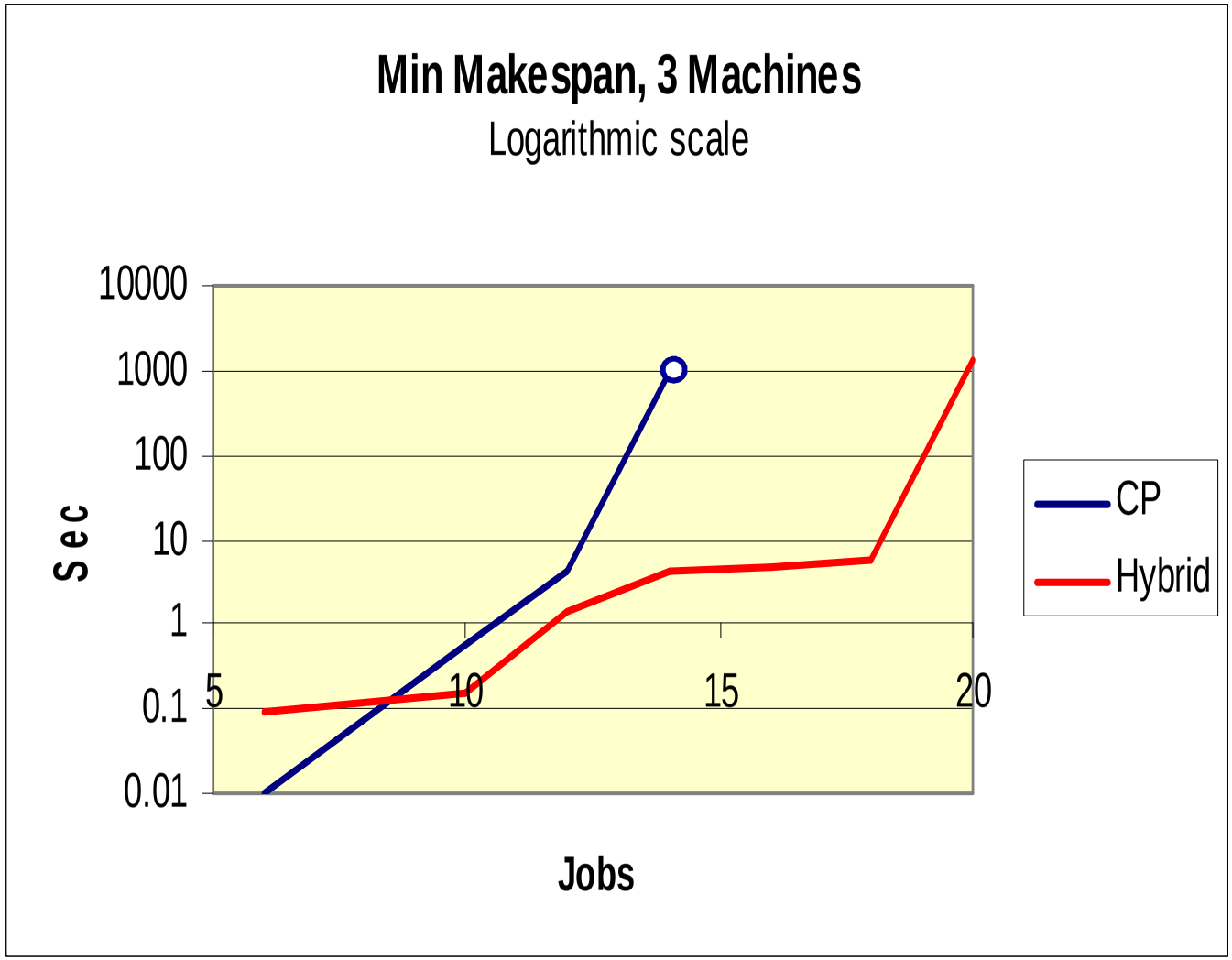


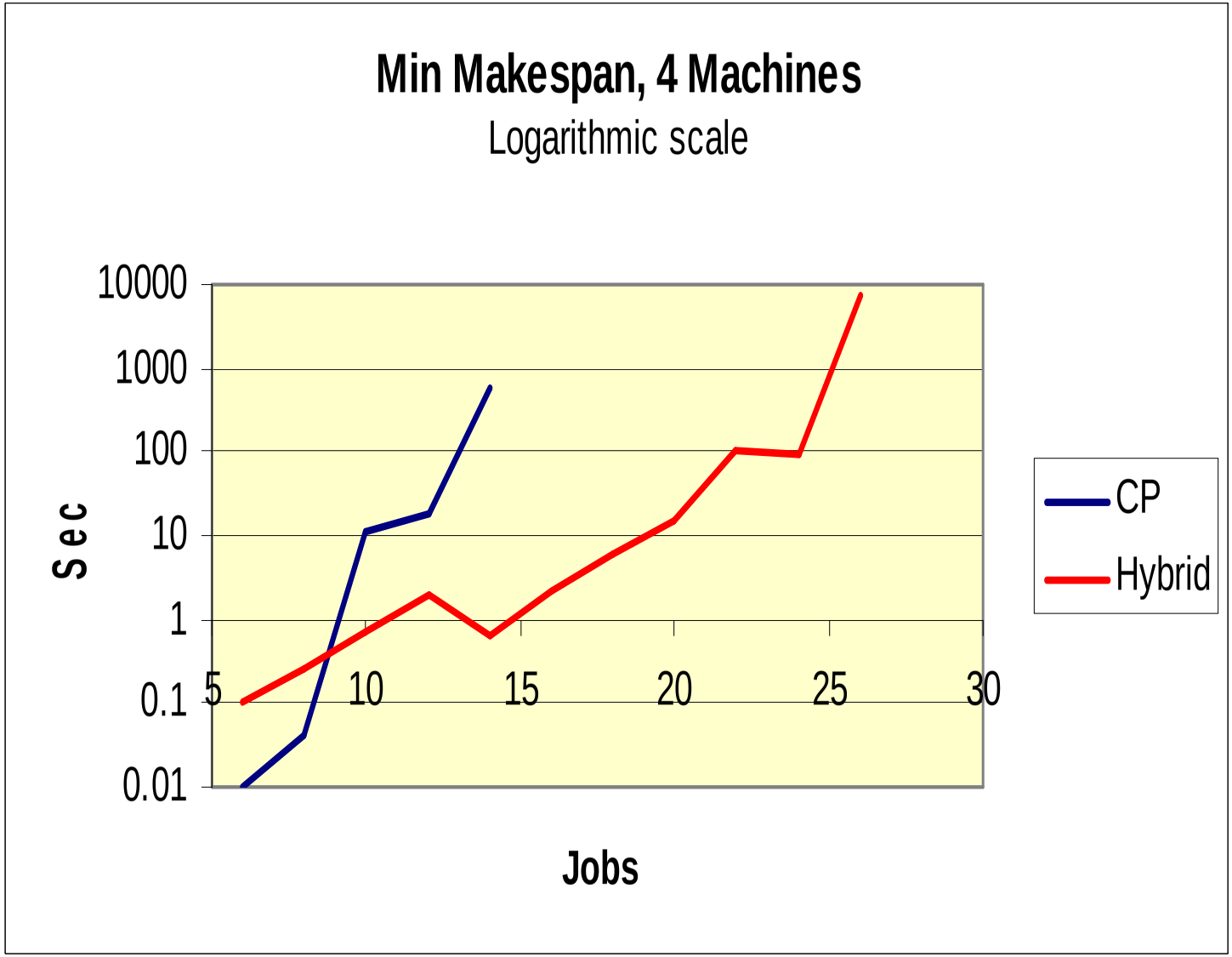
= computation terminated


Min Cost, 3 Machines
Logarithmic scale
10000
1000
100
10
1
0.1
0.01
Sec
5
7
9
11
13
15
17
Jobs
CP
MILP
Hybrid


Min Cost, 4 Machines
Logarithmic scale
10000
1000
100
10
1
0.1
0.01
Sec
5
7
9
11
13
15
17
Jobs
CP
MILP
Hybrid


Min Makespan, 2 Machines
Logarithmic scale
1000
100
10
1
0.1
0.01
Sec
5
7
9
11
13
15
17
Jobs
CP
Hybrid


Min Makespan, 3 Machines
Logarithmic scale
10000
1000
100
10
1
0.1
0.01
Sec
5
10
15
20
Jobs
CP
Hybrid


Min Makespan, 4 Machines
Logarithmic scale
10000
1000
100
10
1
0.1
0.01
Sec
5
10
15
20
25
30
Jobs
CP
Hybrid

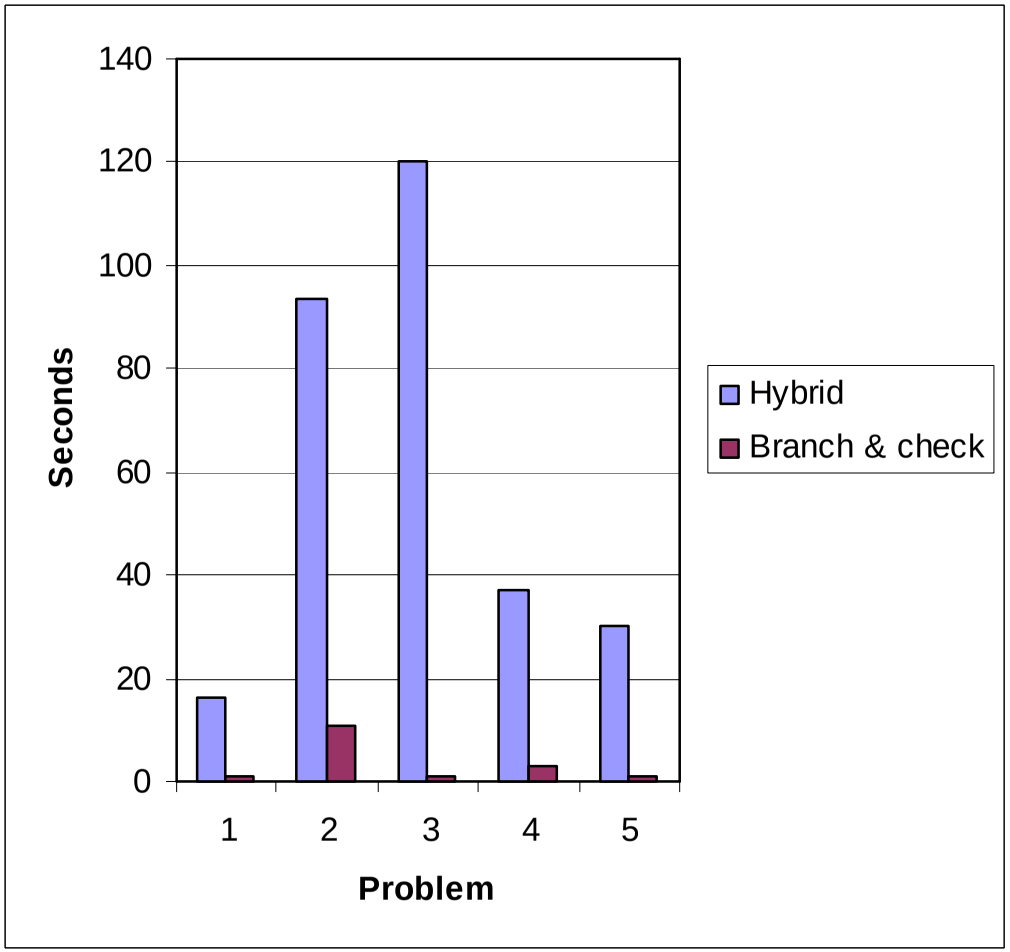
An Enhancement: Branch and Check

(JNH, Thorsteinsson)

- Solve the master problem only once but continually update it with Benders cuts when feasible solutions are found.
- This was applied to the machine scheduling problem described earlier.

Enhancement Using "Branch and Check" *(Thorsteinsson)*

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



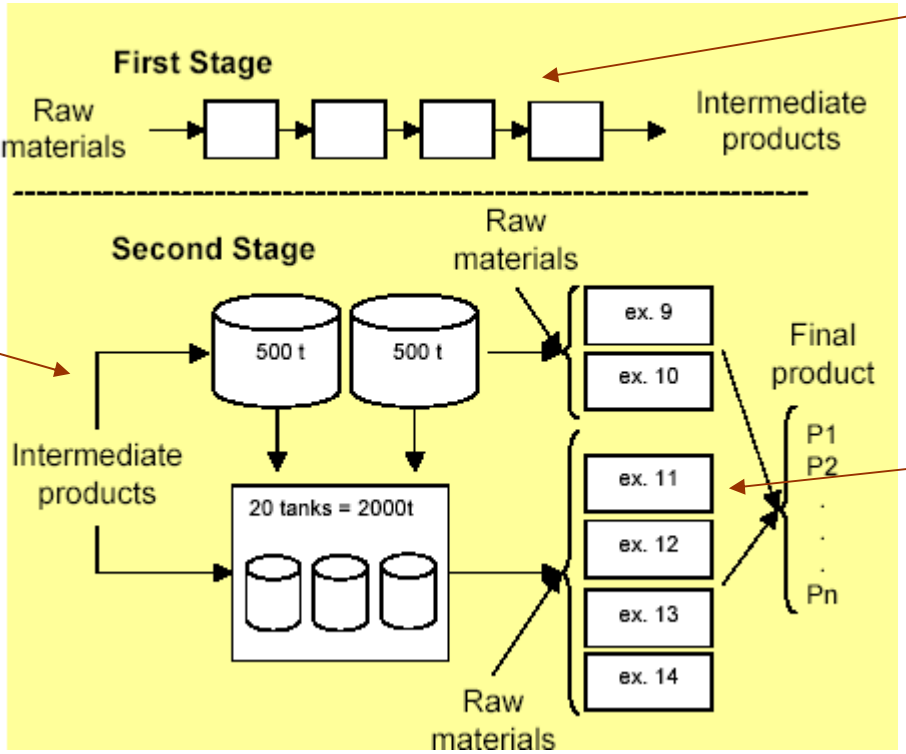
Recent Success Stories

Process Scheduling at BASF
Paint Production at Barbot
Production Line Sequencing at Peugeot/Citroën
Line Balancing at Peugeot/Citroën

Process Scheduling and Lot Sizing at BASF

Manufacture of polypropylenes in 3 stages

polymerization



intermediate storage

extrusion

Process Scheduling and Lot Sizing at BASF

- Manual planning (old method)
  - Required 3 days
  - Limited flexibility and quality control
- 24/7 continuous production
  - Variable batch size.
  - Sequence-dependent changeover times.

Process Scheduling and Lot Sizing at BASF

- Intermediate storage
  - Limited capacity
  - One product per silo
- Extrusion
  - Production rate depends on product and machine

Process Scheduling and Lot Sizing at BASF

- Three problems in one
  - Lot sizing – based on customer demand forecasts
  - Assignment – put each batch on a particular machine
  - Sequencing – decide the order in which each machine processes batches assigned to it

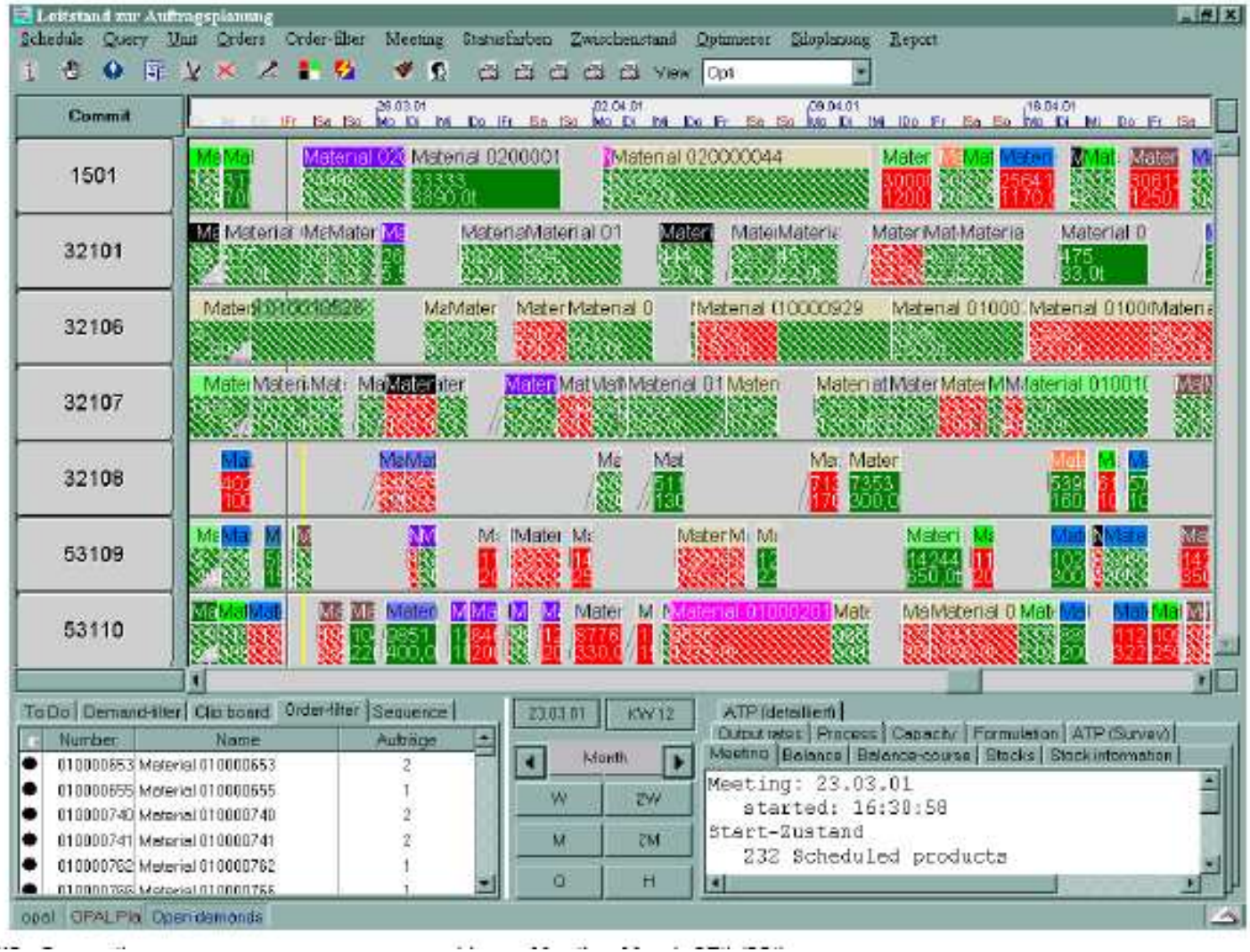
Process Scheduling and Lot Sizing at BASF

- The problems are interdependent
  - Lot sizing depends on assignment, since machines run at different speeds
  - Assignment depends on sequencing, due to restrictions on changeovers
  - Sequencing depends on lot sizing, due to limited intermediate storage

Process Scheduling and Lot Sizing at BASF

- Solve the problems simultaneously
  - *Lot sizing:* solve with MIP (using XPRESS-MP)
  - *Assignment:* solve with MIP
  - *Sequencing:* solve with CP (using CHIP)
- The MIP and CP are linked mathematically.
  - Use logic-based Benders decomposition, developed only in the last few years.

Sample schedule, illustrated with Visual Scheduler (AviS/3)

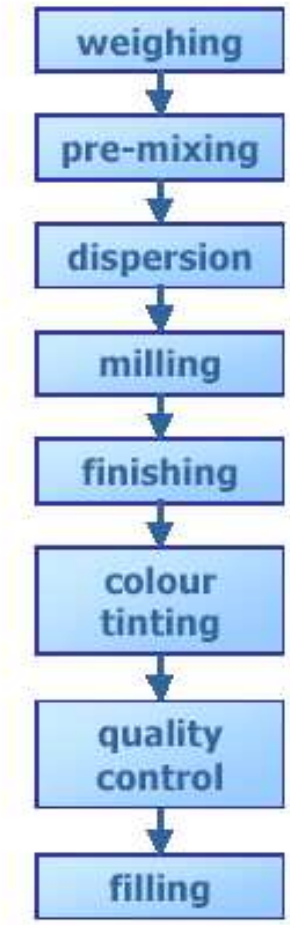


Source: BASF

Process Scheduling and Lot Sizing at BASF

- Benefits
  - Optimal solution obtained in 10 mins.
  - Entire planning process (data gathering, etc.) requires a few hours.
  - More flexibility
  - Faster response to customers
  - Better quality control

Paint Production at Barbot



- Two problems to solve simultaneously
  - Lot sizing
  - Machine scheduling
- Focus on solvent-based paints, for which there are fewer stages.
- Barbot is a Portuguese paint manufacturer.

Several machines of each type

Paint Production at Barbot

- Solution method similar to BASF case (MIP + CP).
- Benefits
  - Optimal solution obtained in a few minutes for 20 machines and 80 products.
  - Product shortages eliminated.
  - 10% increase in output.
  - Fewer cleanup materials.
  - Customer lead time reduced.

Production Line Sequencing at Peugeot/Citroën

- The Peugeot 206 can be manufactured with 12,000 option combinations.
- Planning horizon is 5 days

PSA PEUGEOT CITROËN

Production Line Sequencing at Peugeot/Citroën

- Each car passes through 3 shops.

- Objectives
  - Group similar cars (e.g. in paint shop).
  - Reduce setups.
  - Balance work station loads.

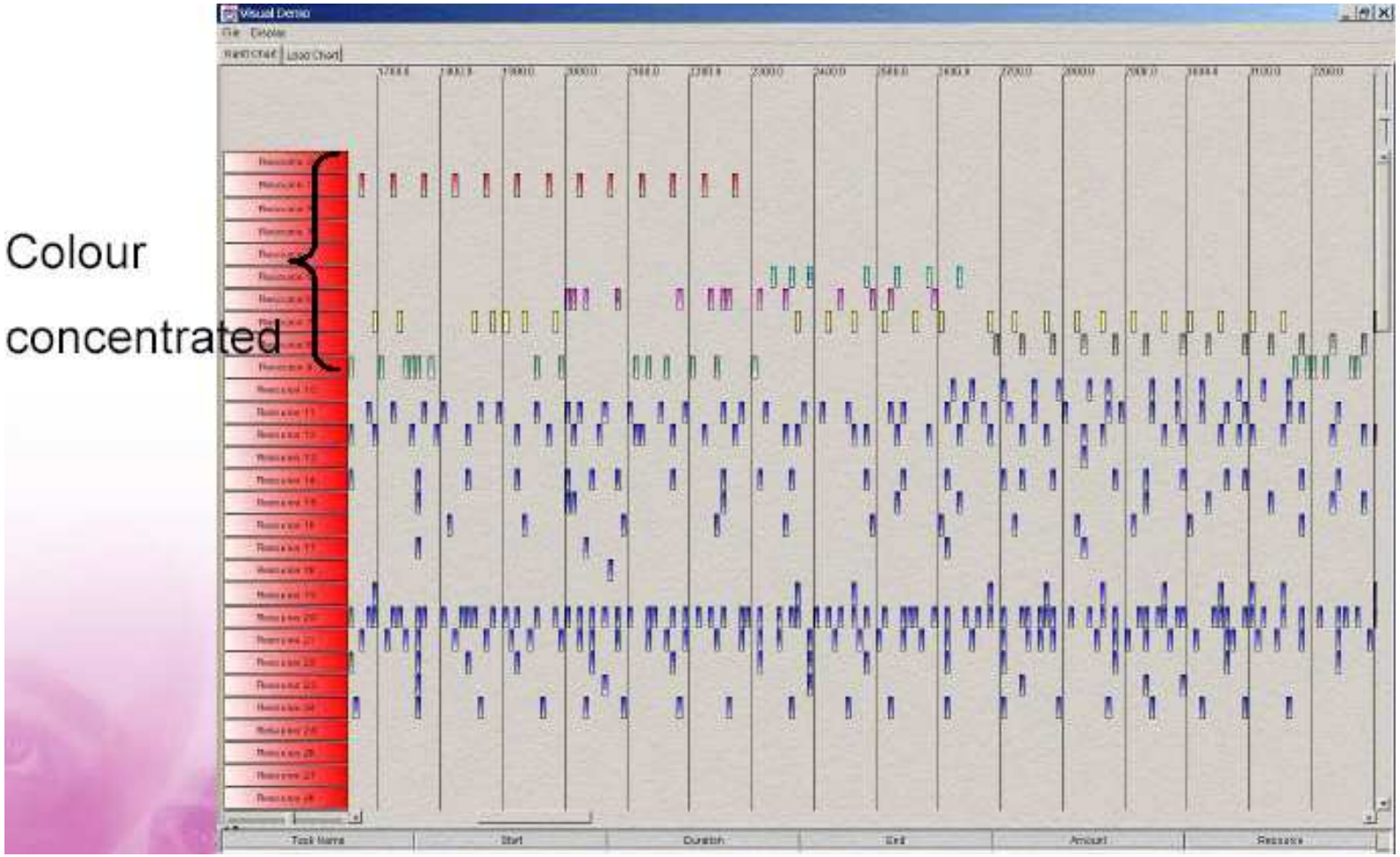
Production Line Sequencing at Peugeot/Citroën

- Special constraints
  - Cars with a sun roof should be grouped together in assembly.
  - Air-conditioned cars should not be assembled consecutively.
  - Etc.

Production Line Sequencing at Peugeot/Citroën

- Problem has two parts
  - Determine number of cars of each type assigned to each line on each day.
  - Determine sequencing for each line on each day.
- Problems are solved simultaneously.
  - Again by MIP + CP.

Sample schedule



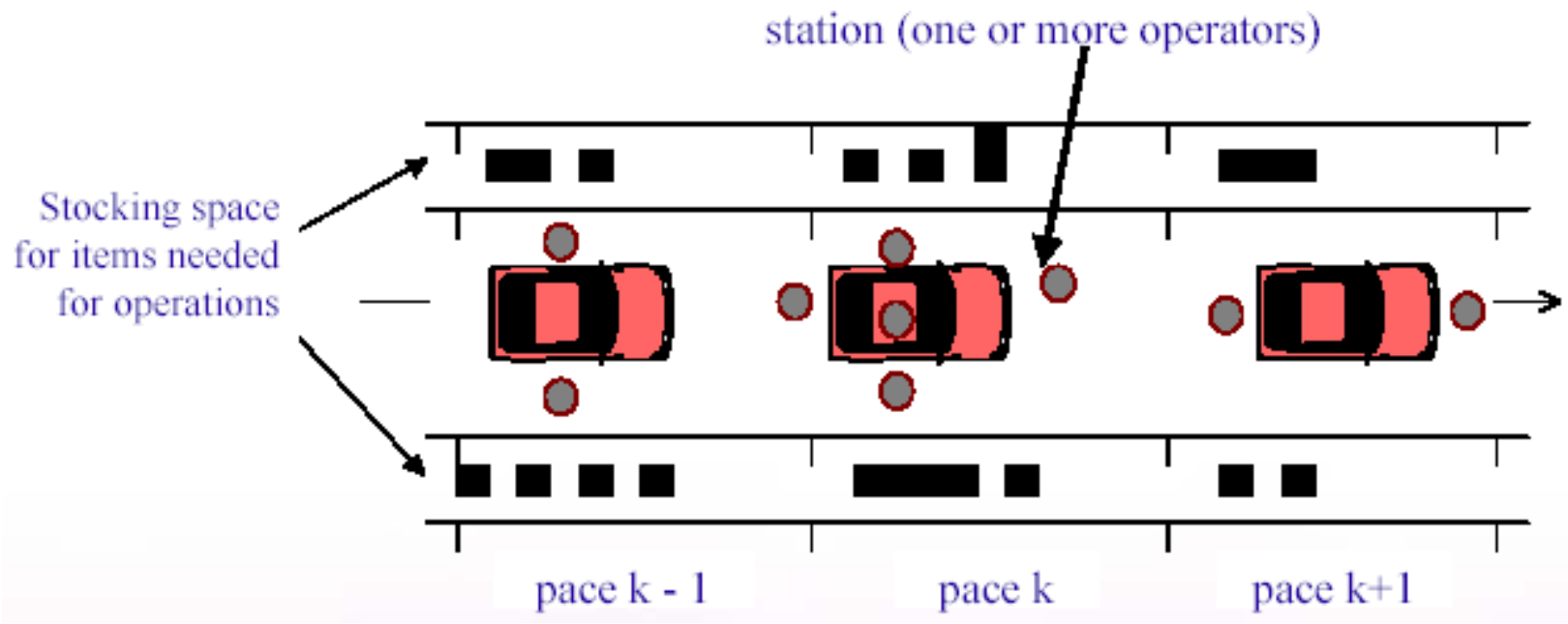
Source: Peugeot/Citroën

Production Line Sequencing at Peugeot/Citroën

- Benefits
  - Greater ability to balance such incompatible benefits as fewer setups and faster customer service.
  - Better schedules.

Line Balancing at Peugeot/Citroën

A classic production sequencing problem



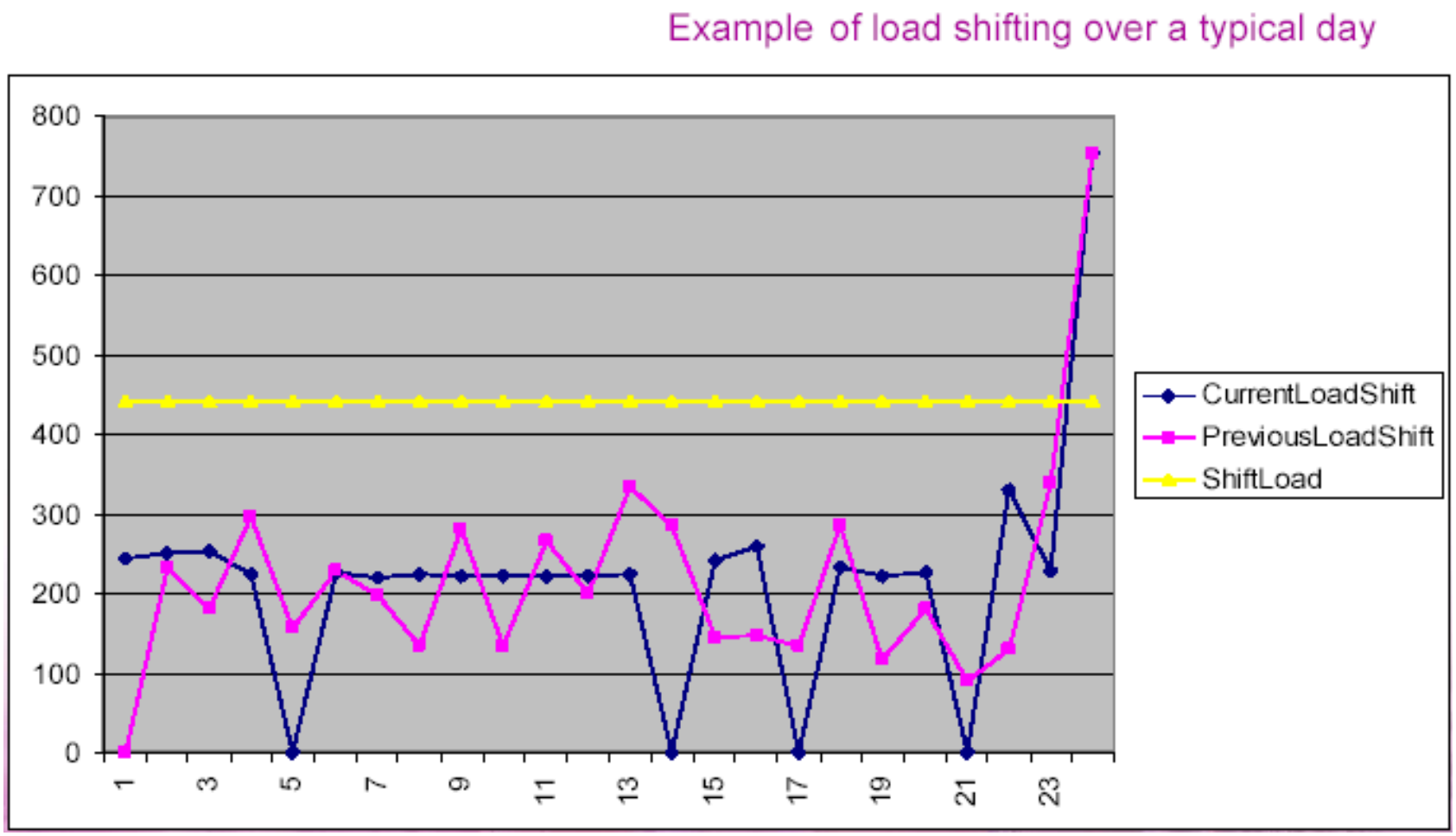
Source: Peugeot/Citroën

Line Balancing at Peugeot/Citroën

- Objective
  - Equalize load at work stations.
  - Keep each worker on one side of the car
- Constraints
  - Precedence constraints between some operations.
  - Ergonomic requirements.
  - Right equipment at stations (e.g. air socket)

Line Balancing at Peugeot/Citroën

- Solution again obtained by a hybrid method.
  - MIP: obtain solution without regard to precedence constraints.
  - CP: Reschedule to enforce precedence constraints.
  - The two methods interact.



Source: Peugeot/Citroën

Line Balancing at Peugeot/Citroën

- Benefits
  - Better equalization of load.
  - Some stations could be closed, reducing labor.
- Improvements needed
  - Reduce trackside clutter.
  - Equalize space requirements.
  - Keep workers on one side of car.