A Hybrid Method for Planning and Scheduling

John Hooker
Carnegie Mellon University

CP 2004
Toronto, September

The Problem

- Allocate tasks to facilities.
- Schedule tasks assigned to each facility.
  - Subject to deadlines.
  - Facilities may run at different speeds and incur different costs.
- Cumulative scheduling
  - Several tasks may run simultaneously on a facility.
  - But total resource consumption must never exceed limit.

Approach

- In practice, problem is often solved by give-and-take.
  - If schedule doesn’t work, schedulers telephone planners and ask for a different allocation
  - Repeat until everyone can live with the solution.
- **Benders decomposition** is a mathematical formalization of this process.
  - Planning is the **master problem**.
  - Scheduling is the **subproblem**.
  - Telephone calls are **Benders cuts**.
- Use **logic-based** Benders.
  - Classical Benders requires that the subproblem be a linear or nonlinear programming problem.

Approach

- Decomposition permits hybrid solution:
  - Apply MILP to planning master problem.
    - MILP is generally better at resource allocation.
  - Apply CP to scheduling subproblem.
    - CP is generally better at scheduling.

Previous Work

1995 (JH & Yan) – Apply logic-based Benders to circuit verification.

- Better than BDDs when circuit contains error.

1995, 2000 (JH) – Formulate general logic-based Benders.

- Specialized Benders cuts must be designed for each problem class.
- Branch-and-check proposed.

2001 (Jain & Grossmann) – Apply logic-based Benders to multiple-machine scheduling using CP/MILP.

- Substantial speedup.
- But… easy problem for Benders approach

2001 (Thorsteinsson) – Apply branch-and-check to CP/MILP.

- 1-2 orders of magnitude speedup on multiple machine scheduling.

2003 (JH, Ottosson) – Apply logic-based Benders to IP (and SAT).

Today – Apply logic-based Benders to resource-constrained planning/scheduling problems.

- Multiple facilities, cumulative scheduling on each facility.
- Minimize cost, makespan, or total tardiness.

Also at this meeting (Cambazard et al.) – Logic-based Benders applied to real-time task allocation & scheduling

Logic-Based Benders Decomposition

$$\begin{aligned} \min \quad & f(x,y) \\ \text{subject to} \quad & C(x,y) \\ & x \in D_x,\ y \in D_y \end{aligned}$$

Basic idea: Search over values of x in **master problem**.

For each $x = \bar{x}$ examined, solve **subproblem** for y.

Solution
of master
problem

Master Problem

$$\begin{aligned} \min_{x,z} \quad & z \\ \text{subject to} \quad & z \ge B_{\bar{x}^k}(x), \text{ all } k \\ & x \in D_x,\ y \in D_y \end{aligned}$$

Subproblem

$$\begin{aligned} \min_{y} \quad & f(\bar{x},y) \\ \text{subject to} \quad & C(\bar{x},y) \\ & y \in D_y \end{aligned}$$

Benders cuts for all iterations k

Logic-Based Benders Decomposition

Subproblem

$$\begin{array}{ll} \min & f(\bar{x}, y) \\ \text{subject to} & C(\bar{x}, y) \\ & y \in D_y \end{array}$$

Subproblem dual

$$\begin{array}{ll} \max & v \\ \text{s.t.} & C(\bar{x}, y) \overset{P}{\Rightarrow} f(\bar{x}, y) \geq v \\ & v \in R,\ P \in Q \end{array}$$

Solution of subproblem **dual** is a proof that cost can be no less than the optimal cost $B_{\bar{x}}(\bar{x})$ when $x = \bar{x}$

We use the *same proof schema* to derive a valid lower bound $B_{\bar{x}}(x)$ for any x.

Benders cut $z \geq B_{\bar{x}}(x)$ (a type of nogood) forces master problem to look at a value of x other than $\bar{x}$ to get a lower cost.

Applying Benders to Planning & Scheduling

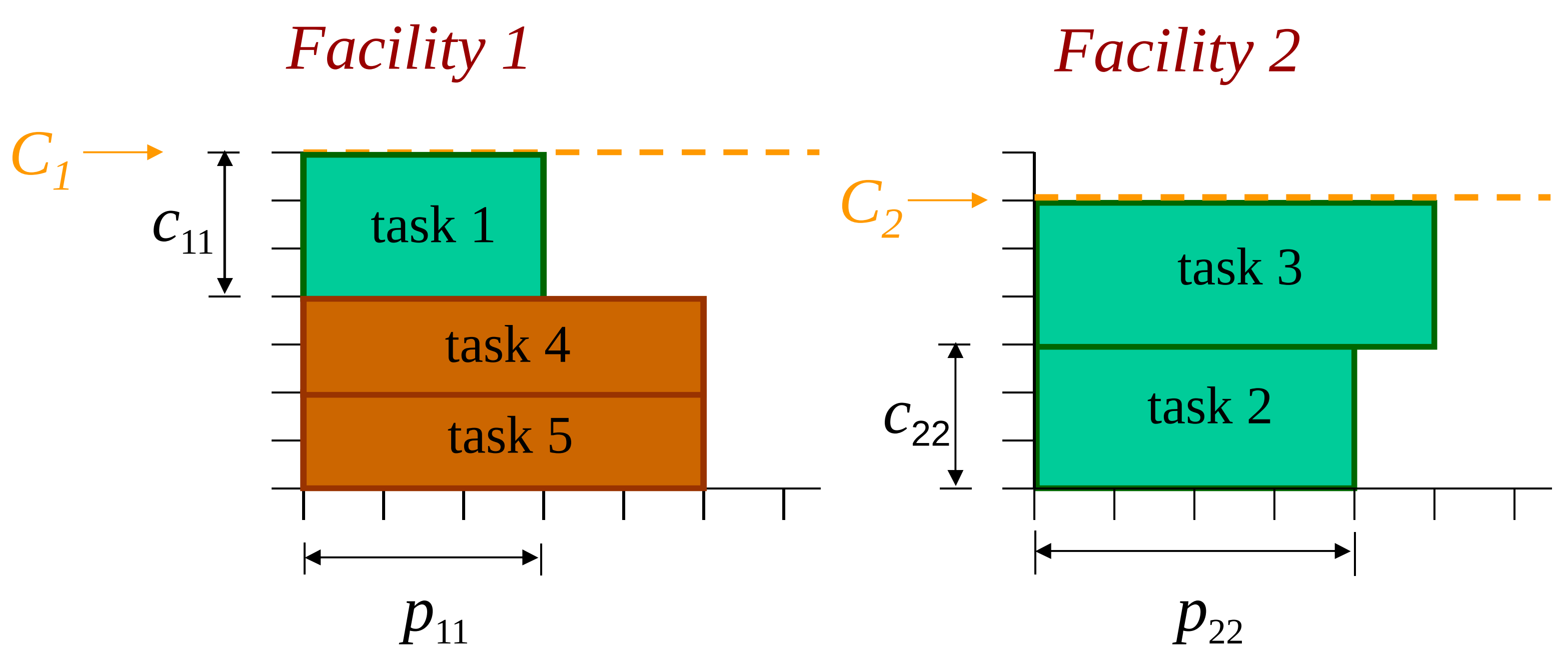
- **Decompose** problem into

assignment	+	resource-constrained scheduling
assign tasks to facilities		*schedule tasks on each facility*

- Use logic-based Benders to link these.
- Solve: master problem with **MILP**
  -- good at resource allocation
  subproblem with **Constraint Programming**
  -- good at scheduling
- We will use Benders cuts that require no information from the CP solution process.

Notation

p_{ij} = processing time of task j on facility i
c_{ij} = resource consumption of task j on facility i
C_i = resources available on facility i



Total resource consumption $\leq C_i$ at all times.

Objective functions

Minimize cost = $\sum_{ij} g_{y_j j}$

facility assigned to task j

Fixed cost of assigning task j to facility y_j

Minimize makespan = $\max_{ij}\{t_j + p_{y_j j}\}$

Start time of task j

Minimize tardiness = $\sum_{ij}\left(t_j + p_{y_j j} - d_j\right)^+$

$\alpha^+ = \max\{0, \alpha\}$

Due date for task j

Minimize cost: MILP Model

= 1 if task j starts at time point t on facility i $(t = 1,\ldots,N)$

Task j starts at one time on one facility

Tasks underway at time t consume $\le C_i$ in resources

$$
\begin{array}{ll}
\min & \displaystyle\sum_{ijt} g_{ij} x_{ijt} \\
\text{subject to} & \displaystyle\sum_{it} x_{ijt} = 1, \quad \text{all } j \\
& \displaystyle\sum_{j} \sum_{\substack{t' \\ t-p_{ij} < t' \le t}} c_{ij} x_{ijt'} \le C_i, \quad \text{all } i,t \\
& x_{ijt} = 0, \quad \text{all } j,t \text{ with } d_j - p_{ij} < t \\
& x_{ijt} = 0, \quad \text{all } j,t \text{ with } t > N - p_{ij} + 1 \\
& x_{ijt} \in \{0,1\}
\end{array}
$$

Tasks observe time windows

Minimize Cost: CP Model

y_j = facility assigned to task j

start times of tasks
assigned to facility i

$$\min \quad \sum_j g_{y_j j}$$

$$\text{subject to} \quad \text{cumulative}\begin{pmatrix} (t_j \mid y_j = i) \\ (p_{ij} \mid y_j = i) \\ (c_{ij} \mid y_j = i) \\ C_i \end{pmatrix}, \quad \text{all } i$$

$$0 \le t_j \le d_j - p_{y_j j}, \quad \text{all } j$$

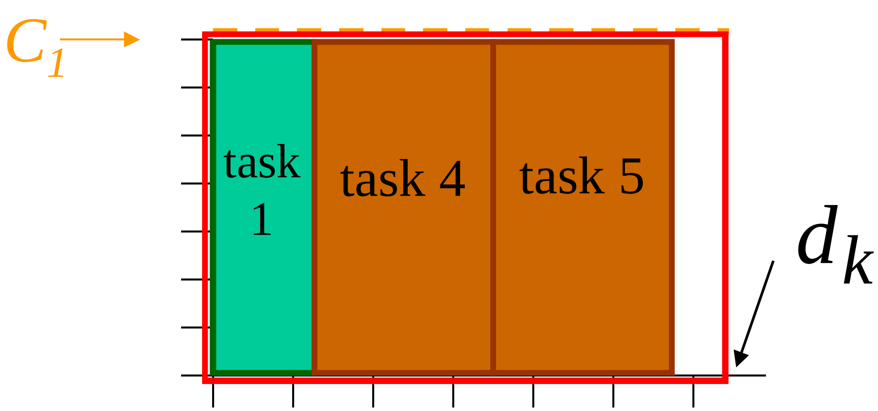
Observe time windows

Observe resource limit
on each facility

Minimize Cost: Logic-Based Benders

Master Problem: Assign tasks to facilities

$$\begin{array}{ll}
\min & \sum_{ij} g_{ij} x_{ij} \\
\text{subject to} & \sum_i x_{ij} = 1, \quad \text{all } j \\
& \sum_{\substack{j \\ d_j \le d_k}} p_{ij} c_{ij} x_{ij} \le C_i d_k, \quad \text{all } i, \text{ all distinct } d_k \\
& \text{Benders cuts}
\end{array}$$



Relaxation of subproblem:
"Area" $d_{ij} r_{ij}$ of tasks due before d_k must fit before d_k.

Subproblem: Schedule tasks assigned to each facility
Solve by constraint programming

solution of master problem

$$\left\{ \begin{array}{l} \text{cumulative}\begin{pmatrix} (t_j \mid \bar{x}_{ij} = 1) \\ (p_{ij} \mid \bar{x}_{ij} = 1) \\ (c_{ij} \mid \bar{x}_{ij} = 1) \\ C_i \end{pmatrix} \\ 0 \le t_j \le d_j \end{array} \right\}, \quad \text{all } i$$

Let J_{ih} = set of tasks assigned to facility i in iteration h.

If subproblem i is infeasible, solution of subproblem dual is a proof that not all tasks in J_{ih} can be assigned to facility i.

This provides the basis for a simple Benders cut.

Master Problem with Benders Cuts
Solve by MILP

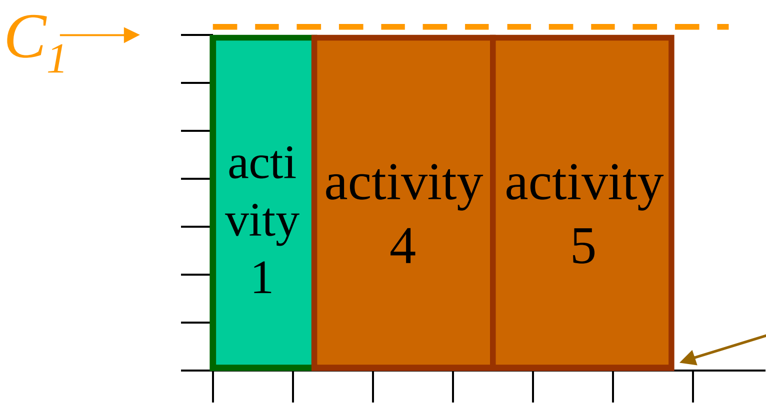
$$
\begin{aligned}
\text{min} \quad & \sum_{ij} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{i} x_{ij} = 1, \quad \text{all } j \\
& \sum_{\substack{j \\ d_j \le d_k}} p_{ij} r_{ij} x_{ij} \le C_i d_k, \quad \text{all } i, \text{all distinct } d_k \\
& \sum_{j \in J_{ih}} (1 - x_{ij}) \ge 1, \quad \text{all } i, h \\
& x_{ij} \in \{0,1\}
\end{aligned}
$$

Benders cuts

Minimize Makespan: Logic-Based Benders

Master Problem: Assign tasks to facilities

$$
\begin{array}{ll}
\min & M \quad \longleftarrow \text{makespan} \\
\text{subject to} & \sum_i x_{ij} = 1, \quad \text{all } j \\
& M \ge \dfrac{1}{C_i} \sum_j p_{ij} c_{ij} x_{ij}, \quad \text{all } i \\
& \text{Benders cuts}
\end{array}
$$



Relaxation of subproblem:
"Area" of tasks provides lower bound on makespan.

Subproblem: Schedule tasks assigned to each facility
Solve by constraint programming

$$\min \quad M$$

$$\text{subject to} \quad \left\{ \begin{array}{l} M \ge t_j + d_{ij}, \quad \text{all } j \\ \text{cumulative}\begin{pmatrix} (t_j \mid \bar{x}_{ij} = 1) \\ (p_{ij} \mid \bar{x}_{ij} = 1) \\ (c_{ij} \mid \bar{x}_{ij} = 1) \\ C_i \end{pmatrix} \\ 0 \le t_j \le d_j, \quad \text{all } j \end{array} \right\}, \quad \text{all } i$$

Let J_{ih} = set of tasks assigned to machine i in iteration h.

We get a Benders cut even when subproblem is feasible.

The Benders cut is based on:

Lemma. If we remove tasks 1, … s from a facility, the minimum makespan on that facility is reduced by at most

$$\sum_{j=1}^{s} p_{ij} + \max_{j \le s}\{d_j\} - \min_{j \le s}\{d_j\}$$

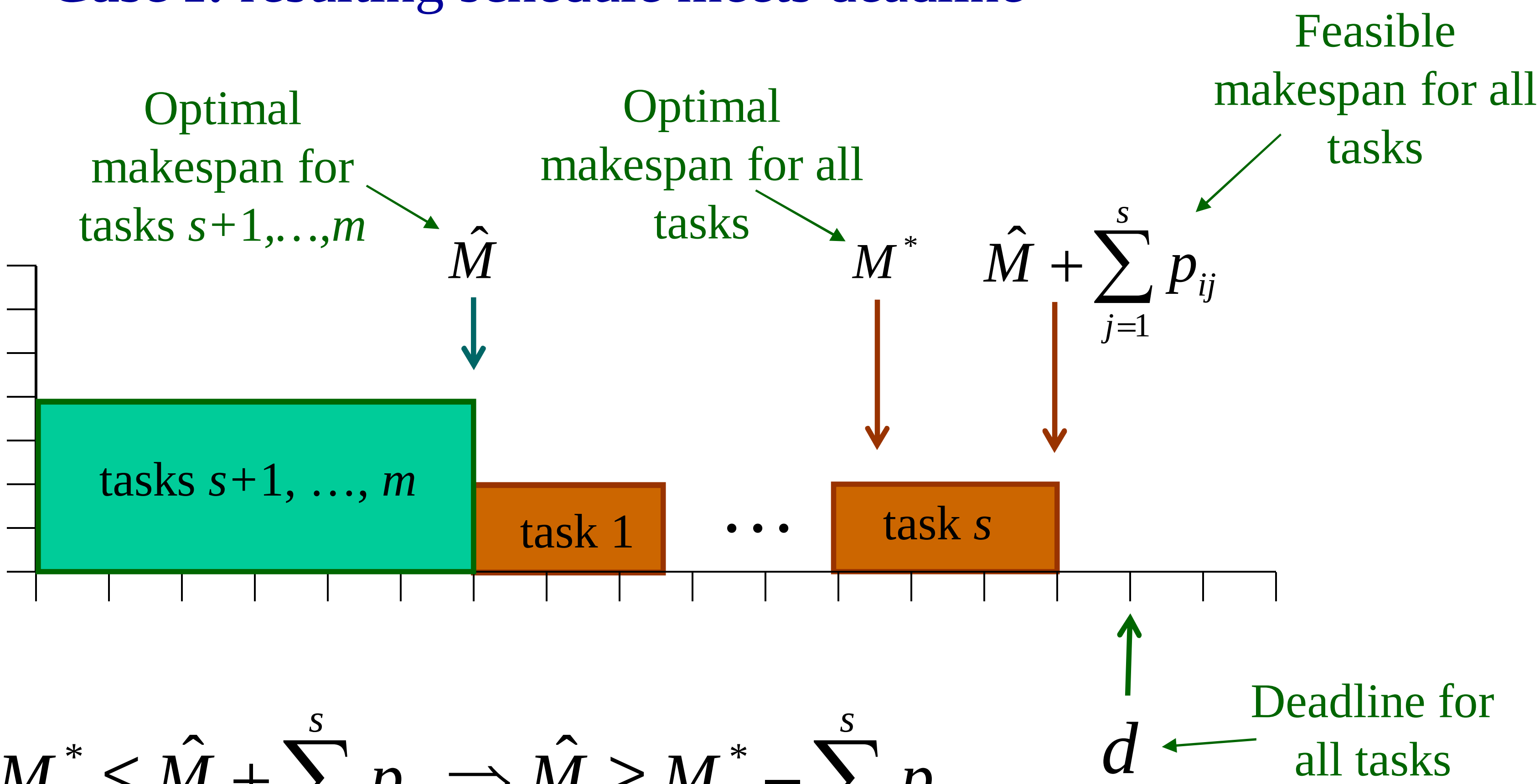
Assuming all deadines d_i are the same, we get the Benders cut

$$M \ge M_{hi}^{*} - \sum_{j \in J_{hi}} (1 - x_{ij})\, p_{ij}$$

Min makespan on facility i in last iteration

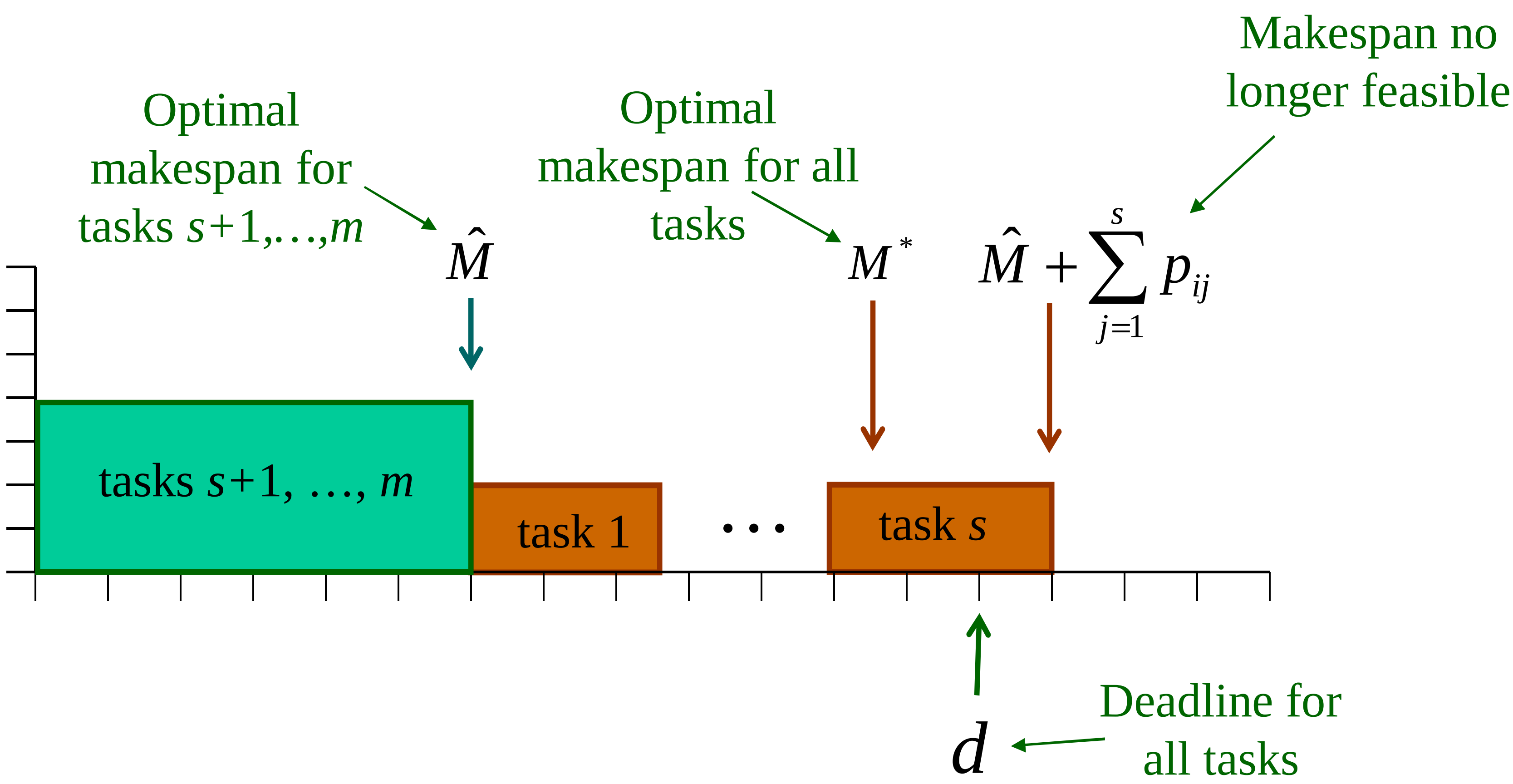
Why does this work? Add tasks 1,…,s sequentially at end of optimal schedule for other tasks…

Case I: resulting schedule meets deadline



$$M^* \le \hat{M} + \sum_{j=1}^{s} p_{ij} \Rightarrow \hat{M} \ge M^* - \sum_{j=1}^{s} p_{ij}$$

Case II: resulting schedule exceeds deadline



$$M^* \le d \ \text{ and } \ \hat{M}+\sum_{j=1}^{s} p_{ij} > d \Rightarrow \hat{M} \ge M^* - \sum_{j=1}^{s} p_{ij}$$

Master Problem: Assign tasks to facilities
Solve by MILP

$$
\begin{aligned}
\min \quad & M \\
\text{subject to} \quad & \sum_i x_{ij} = 1, \quad \text{all } j \\
& M \geq \frac{1}{C_i} \sum_j p_{ij} c_{ij} x_{ij}, \quad \text{all } i \\
& M \geq M^*_{hi} - \sum_{j \in J_{ik}} (1 - x_{ij}) p_{ij}, \quad \text{all } i, h \\
& x_{ij} \in \{0,1\}
\end{aligned}
$$

Relaxation

Benders cuts

Makespan on facility *i* in iteration *h*

Minimize Tardiness: Logic-Based Benders

Master Problem: Assign tasks to facilities

tardiness

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \sum_i x_{ij} = 1, \quad \text{all } j \\
& \text{relaxation of subproblem} \\
& \text{Benders cuts}
\end{array}
$$

Relaxation of subproblem

Lemma. Consider a min tardiness problem that schedules tasks 1, …, n on facility i, where $d_1 \le \ldots \le d_n$. The min tardiness T^* is bounded below by

$$L = \sum_{k=1}^{n} L_k$$

where

$$L_k = \left(\frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} - d_k \right)^+$$

and π is a permutation of 1, …, n such that

$$p_{\pi_i(1)} c_{\pi_i(1)} \le \cdots \le p_{\pi_i(n)} c_{\pi_i(n)}$$

Idea of proof

For a permutation σ of $1,\ldots,n$ let $L(\sigma) = \sum_{k=1}^{n} L_k(\sigma)$

where $$L_k(\sigma) = \left(\frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} - d_{\sigma(k)} \right)^{+}$$

Let $\sigma_0(1), \ldots, \sigma_0(n)$ be order of jobs in any optimal solution, so that $t_{\sigma_0(1)} \leq \cdots \leq t_{\sigma_0(n)}$ and min tardiness is T^*

Consider bubble sort on $\sigma_0(1), \ldots, \sigma_0(n)$ to obtain $1,\ldots,n$. Let $\sigma_0,\ldots, \sigma_S$ be resulting sequence of permutations, so that σ_s, σ_{s+1} differ by a swap and $\sigma_s(j) = j$.

since $t_k \geq \frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)}$

Now we have

swap k and $k+1$

$$T^* \geq L(\sigma_0) \geq \cdots \geq L(\sigma_s) \geq L(\sigma_{s+1}) \geq \cdots \geq L(\sigma_S) = L$$

$$L(\sigma_s) = \sum_{j=1}^{k-1} L_j(\sigma_s) + L_k(\sigma_s) + L_{k+1}(\sigma_s) + \sum_{j=k+2}^{n} L_j(\sigma_s)$$

$$L(\sigma_{s+1}) = \sum_{j=1}^{k-1} L_j(\sigma_s) + L_k(\sigma_{s+1}) + L_{k+1}(\sigma_{s+1}) + \sum_{j=k+2}^{n} L_j(\sigma_s)$$

So
$$\begin{aligned} L(\sigma_s) - L(\sigma_{s+1}) &= L_k(\sigma_s) + L_{k+1}(\sigma_s) - L_k(\sigma_{s+1}) - L_{k+1}(\sigma_{s+1}) \\ &= (a-A)^+ + (A-b)^+ - (a-b)^+ - (A-B)^+ \geq 0 \end{aligned}$$

since $A \geq a, \ B \geq b$

From the lemma, we can write the relaxation

$$T \geq \sum_i \sum_{k=1}^{n} L'_{ik} x_{ik}$$

where $L'_{ik} \geq \dfrac{1}{C_i} \displaystyle\sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} x_{i\pi_i(j)} - d_k$

To linearize this, we write $T \geq \displaystyle\sum_i \sum_{k=1}^{n} L_{ik}$

and $L_{ik} \geq \dfrac{1}{C_i} \displaystyle\sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} x_{i\pi_i(j)} - d_k - (1 - x_{ik}) M_{ik}$

where $M_{ik} = \dfrac{1}{C_i} \displaystyle\sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} - d_k$

Benders cuts

To extract some “dual” information, re-solve the scheduling subproblem a few times with some tasks removed.

Let J_{hi}^0 = {tasks that can be individually removed without reducing min makespan}

Δ_{hi} = reduction in min makespan if all tasks in J_{hi}^0 are removed simultaneously

This yields Benders cuts:

$$T \ge T_{hi}^* - \Delta_{hi} - U \sum_{j \in J_{hi} \setminus J_{hi}^0} (1 - x_{ij}), \quad \text{all } i, h$$

$$T \ge T_{hi}^* - U \sum_{j \in J_{hi} \setminus J_{hi}^0} (1 - x_{ij}), \quad \text{all } i, h$$

Computational Results

- Random problems on 2, 3, 4 facilities.
- Facilities run at different speeds.
- All release times = 0.
  - Min cost and makespan problems: all tasks have same deadline.
  - Tardiness problems: random due date parameters set so that a few tasks tend to be late.
- No precedence or other side constraints.
  - Makes problem harder.
- Implement with OPL Studio
  - CPLEX for MILP.
  - ILOG Scheduler for CP. Use AssignAlternatives & SetTimes.

Min cost, 2 facilities

Computation time in seconds
Average of 5 instances shown

Jobs	MILP*	CP	Benders
10	1.9	0.14	0.09
12	199	2.2	0.06
14	1441	79	0.04
16	3605+	1511	1.1
18		7200+	7.0
20			85
22			1674+

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Min cost, 3 facilities

Computation time in seconds
Average of 5 instances shown

Tasks	MILP*	CP	Benders
10	0.9	0.13	0.37
12	797	2.6	0.55
14	114	35	0.34
16	678*	1929	4.5
18		7200+	15
20			2.9
22			23
24			53

*CPLEX ran out of memory on 1 or more problems.

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Min cost, 4 facilities

Computation time in seconds
Average of 5 instances shown

Jobs	MILP*	CP	Benders
10	2.0	0.10	0.6
12	7.2	1.4	4.0
14	158	72	2.8
16	906*	344	0.8
18		6343+	5.2
20			2.6
22			22
24			114
26			76

*CPLEX ran out of memory on 1 or more problems.

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Min makespan, 2 facilities

Average of 5 instances shown

Jobs	MILP	CP	Benders
10	3.4	0.8	0.08
12	12	4.0	0.39
14	2572+	299	7.8
16	5974+	3737	30
18		7200+	461
20			2656

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Min makespan, 3 facilities

Average of 5 instances shown

Jobs	MILP	CP	Benders
10	3.9	0.9	0.06
12	12	7.5	0.3
14	524	981	0.7
16	1716+	4414	6.5
18	4619+	7200+	13.3
20			34
22			3084+

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Min makespan, 4 facilities

Average of 5 instances shown

Jobs	MILP	CP	Benders
10	1.0	0.07	0.09
12	5.0	1.9	0.09
14	24	524	0.8
16	35	3898	0.9
18	3931+	7200+	14
20			25
22			472
24			1131

+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Scaling up the Benders Method

Average of 5 instances shown

Tasks	Facilities	Min cost (sec)	Min makespan (sec)
10	2	0.1	0.2
15	3	0.7	1.6
20	4	50	13
25	5	2.9	213
30	6	4.8	3373+
35	7	128	6404+
40	8	1792+	7200+

\+ At least one problem in the 5 exceeded 7200 sec (2 hours)

Bounds Provided by Benders

Min makespan problems unsolved after 2 hours

Tasks	Facilities	Best solution value	Lower bound
30	6	13	12
35	7	11	10
35	7	15	13
40	8	14	11
40	8	15	12
40	8	16	13
40	8	10	9
40	8	13	11

Min tardiness, 3 facilities

Smaller problems

Tasks	Time (sec)			Min tardiness
	CP	MILP	Benders	
10	13	4.7	2.8	10
	1.1	6.4	1.5	10
	1.4	6.4	1.9	16
	4.6	32	4.5	17
	8.1	33	23	24
12	4.7	0.7	0.3	0
	14	0.6	0.3	0
	25	0.7	0.2	1
	19	15	2.3	9
	317	25	11	15
14	838	7.0	2.2	1
	7159	34	4.2	2
	1783	45	20	15
	>7200	73	47	19
	>7200	>7200	3123	26
16	>7200	19	1.2	0
	>7200	46	1.6	0
	>7200	52	5.3	4
	>7200	1105	187	20
	>7200	3424	752	31

Min tardiness,
3 facilities

Larger problems

On all problems:

average time ratio
MILP/Benders = 20

Tasks	Time (sec) MILP	Time (sec) Benders	Best solution MILP	Best solution Benders
18	187	4.0	**0**	**0**
	15	8.1	**3**	**3**
	46	53	**5**	**5**
	256	54	**11**	**11**
	>7200	1146	**14**	**14**
20	105	11	**0**	**0**
	4141	16	**1**	**1**
	39	28	**4**	**4**
	1442	305	**8**	**8**
	>7200	>7200	**75**	**75**
22	6	20	**0**	**0**
	584	36	**2**	**2**
	>7200	>7200	120	40
	>7200	>7200	162	46
	>7200	>7200	375	128
24	10	661	**0**	**0**
	>7200	53	20	**0**
	>7200	72	57	**0**
	>7200	>7200	20	5
	>7200	>7200	25	7

Boldface = optimality proved

Future Research

- Implement branch-and-check for Benders problem.
- Exploit dual information from the subproblem solution process (e.g. edge finding).
- Explore other problem classes.
    - Integrated long- and short-term scheduling
    - Vehicle routing
    - SAT (subproblem is renamable Horn)
    - Stochastic IP