Planning and Scheduling to Minimize Makespan & Tardiness

John Hooker
Carnegie Mellon University

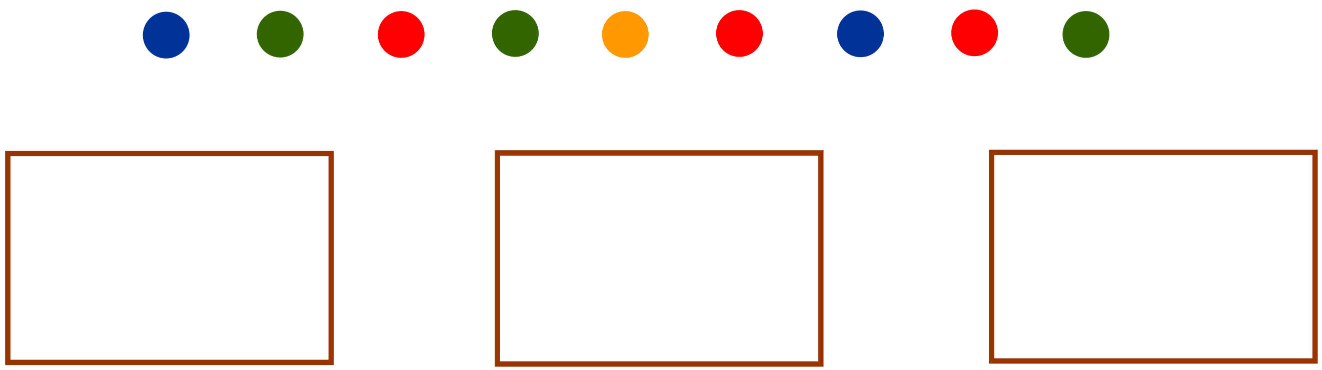
September 2006

The Problem

Given a set of tasks, each with a deadline…

The Problem

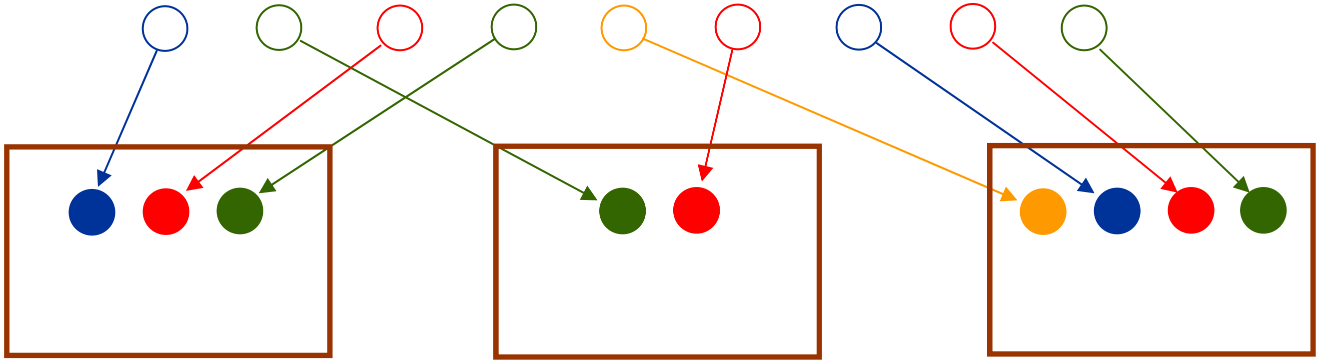
Given a set of tasks, each with a deadline…



…and processing facilities that run at different speeds.

The Problem

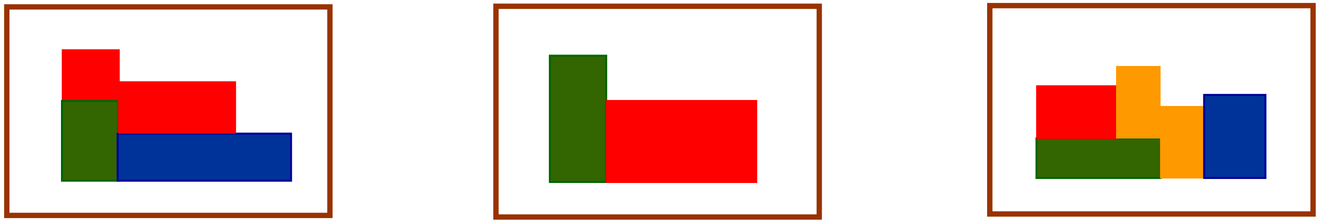
Allocate tasks to facilities.



The Problem

Allocate tasks to facilities.

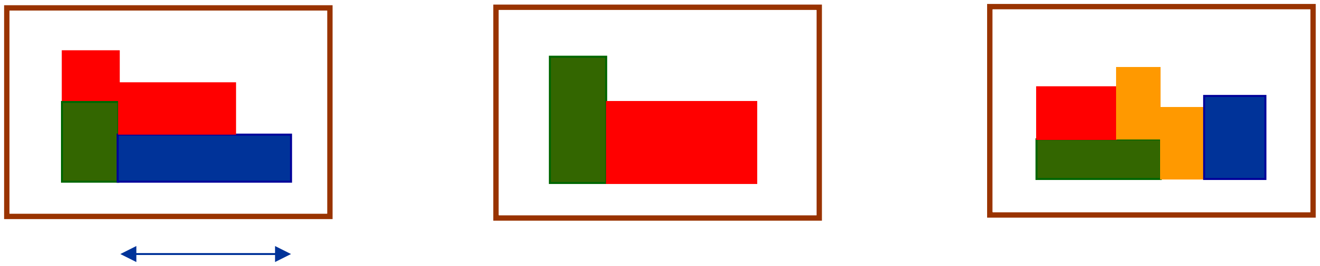
Schedule tasks on each facility
(cumulative scheduling)



The Problem

Allocate tasks to facilities.

Schedule tasks on each facility
(cumulative scheduling)

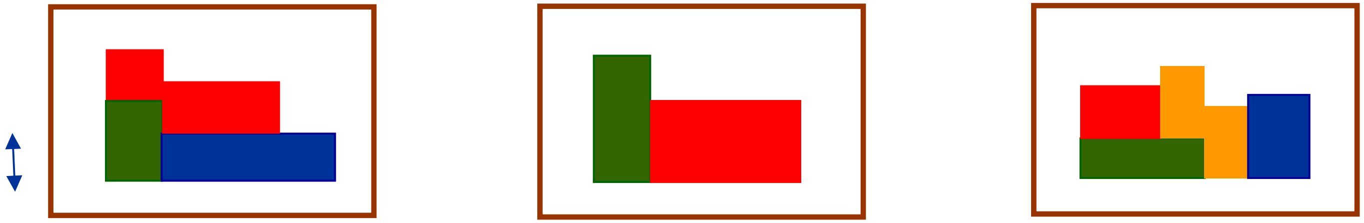


Each task has a given processing time on each facility.

The Problem

Allocate tasks to facilities.

Schedule tasks on each facility
(cumulative scheduling)



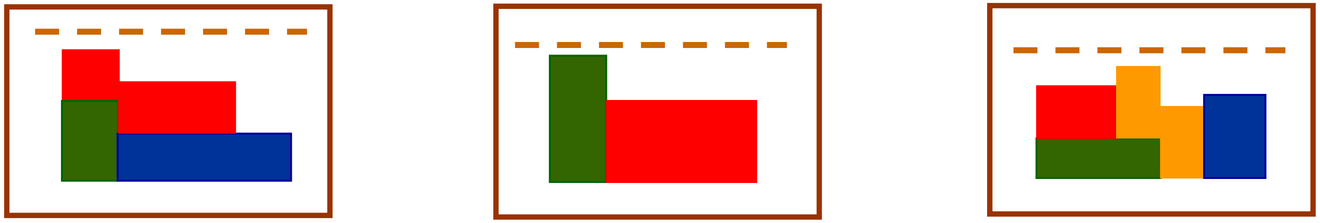
Each task has a given processing time on each facility.

Each task consumes resources at a given rate on each facility.

The Problem

Allocate tasks to facilities.

Schedule tasks on each facility
(cumulative scheduling)



Each task has a given processing time on each facility.

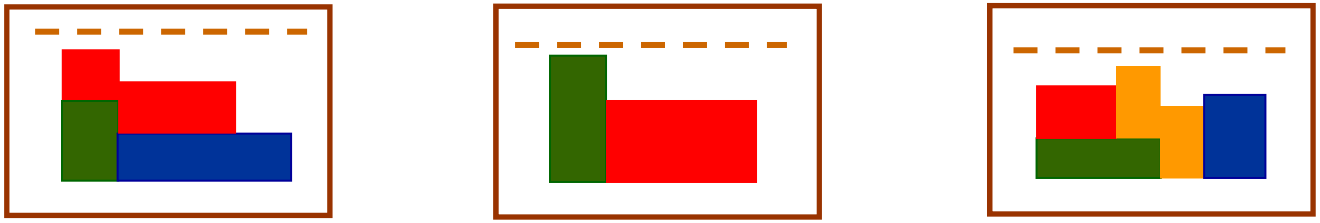
Each task consumes resources at a given rate on each facility.

Each facility has a resource limit.

The Problem

Allocate tasks to facilities.

Schedule tasks on each facility
(cumulative scheduling)

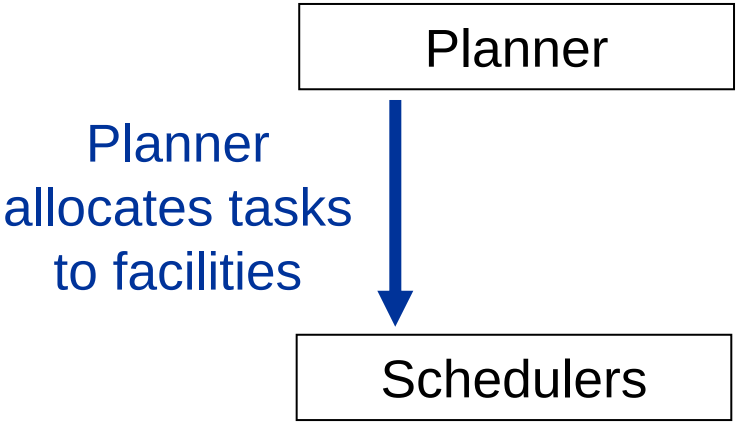


Objectives:

- Minimize makespan
- Minimize number of late tasks.
- Minimize total tardiness

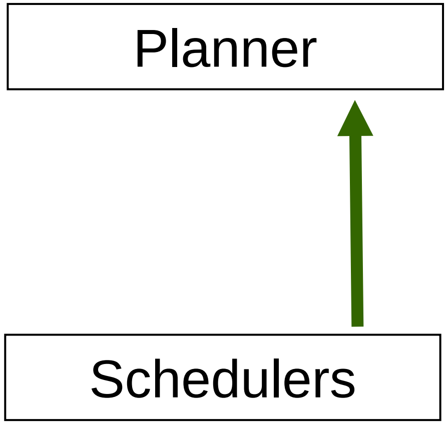
Approach

- In practice, problem is often solved by give-and-take.



Approach

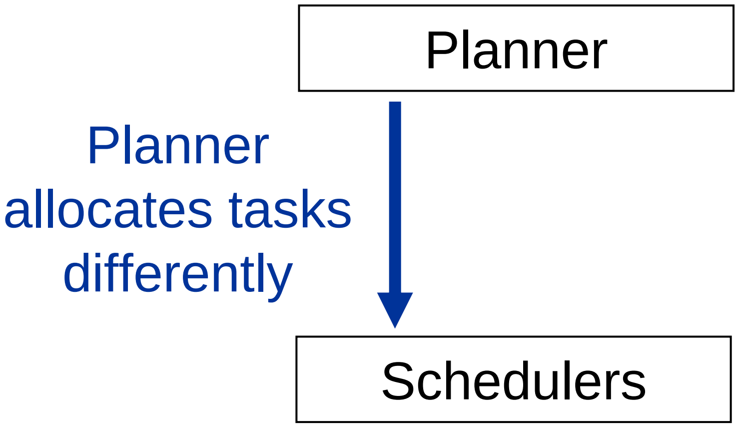
- In practice, problem is often solved by give-and-take.



If there are problems, schedulers telephone planners and ask for a different allocation.

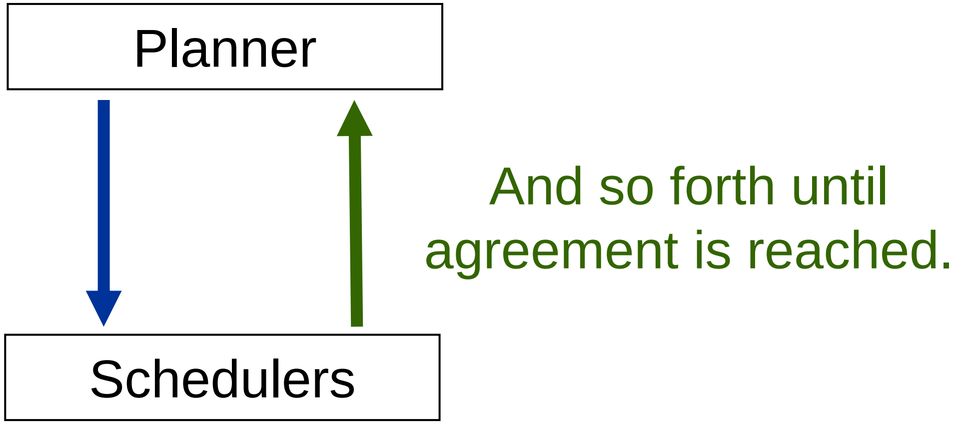
Approach

- In practice, problem is often solved by give-and-take.



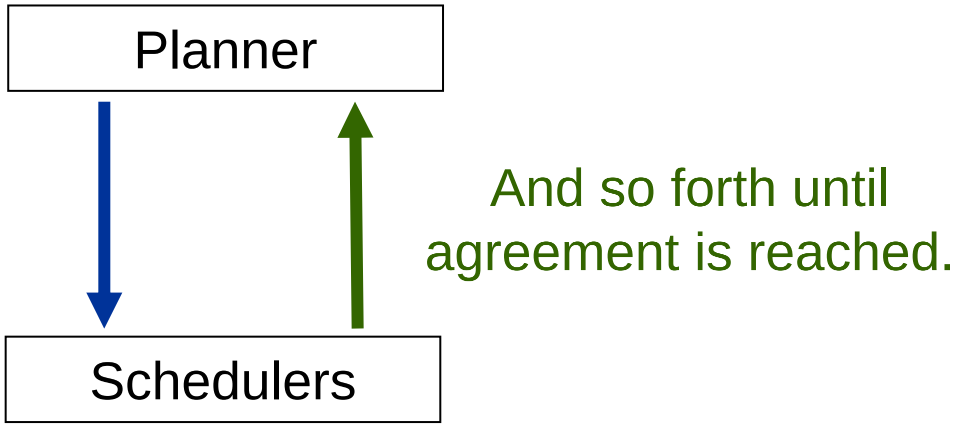
Approach

- In practice, problem is often solved by give-and-take.



Approach

- In practice, problem is often solved by give-and-take.



- Benders decomposition is a mathematical formulation of this process.
  - Planning is the **master problem**.
  - Scheduling is the **subproblem**.
  - Telephone calls are **Benders cuts**.

Approach

- Use **logic-based** Benders.
  - Since classical Benders requires that the subproblem be a linear or nonlinear programming problem.
- 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.

Related Work

1995 (JH & Yan) – Logic-based Benders.

- Applied to logic circuit verification.
- Faster than BDDs when circuit contains error

2000 (JH) – Theory of logic-based Benders.

- Branch-and-check proposed.

2000 (JH) – Application to planning & scheduling proposed.

- Combine CP & MILP.
- Application to integer programming, SAT.

2001 (Jain & Grossmann) – Application to machine assignment and disjunctive scheduling.

- Simple Benders cuts, since subproblem is a feasibility problem.
- 20 to 1000 times faster than CP, MILP

Related Work

2001 (Thorsteinsson) – Branch and check applied to Jain & Grossmann problems.

- Update solution of master problem, factor of 10 further speedup

2002 (Harjunkowski & Grossmann) – Generalization of Jain & Grossmann to multistage.

2002 (Timpe) – Polypropylene batch scheduling at BASF.

- Solved previously insoluble problem in 10 mins.
- Similar approach applied to automobile assembly (Peugeot/Citroën), paint mixing (Barbot).

2003 (JH, Ottosson) – Integer programming.

- Benders cuts less obvious, since subproblem is an optimization problem.
- Suitable for stochastic integer programming – breakpoint at 20-30 scenarios

Related Work

2004 (JH) – Min-cost and min-makespan planning & cumulative scheduling.

- Benders cuts less obvious in min makespan, since subproblem is optimization problem.
- 100 to 1000 times faster than CP, MILP.

2004 (Cambazard, Jussien et al.) – Real-time scheduling of computer processes.

- CP master problem.

2004 (Chiu & Xia) – Integer programming.

- Used min-conflict + classical Benders cuts.

Related Work

2004 (Maravelias and Grossmann) – Batch scheduling in chemical plants.

2004 (Correa, Langevin & Rousseau) – Automated guided vehicles.

2005 (Rasmussen & Trick) – Sports scheduling (min # consecutive home and away games)

- Orders of magnitude speedup over state of the art.

2005 (JH) – Min-tardiness planning and cumulative scheduling.

- 10 - >1000 times faster than CP, MILP when minimizing # late jobs.
- ~10 times faster when minimizing total tardiness, much better suboptimal solutions.

Logic-Based Benders Decomposition

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

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

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

Master Problem

$$\begin{array}{ll} \min\limits_{x,z} & z \\ \text{subject to} & z \ge B_{\bar{x}^h}(x),\ \text{all } h \\ & x \in D_x,\ y \in D_y \end{array}$$

Subproblem

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

Solution of master problem

Benders cuts for all iterations *h*

Logic-Based Benders Decomposition

Subproblem

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

After solving the subproblem, generate a **Benders cut** $z \geq B_{\bar{x}}(x)$ where $B_{\bar{x}}(\bar{x})$ is the optimal value of the subproblem.

The Benders cut is based on a logical analysis of subproblem solution.

Re-solve the master problem and continue until it has the same optimal value as the subproblem.

Applying Benders to Planning & Scheduling

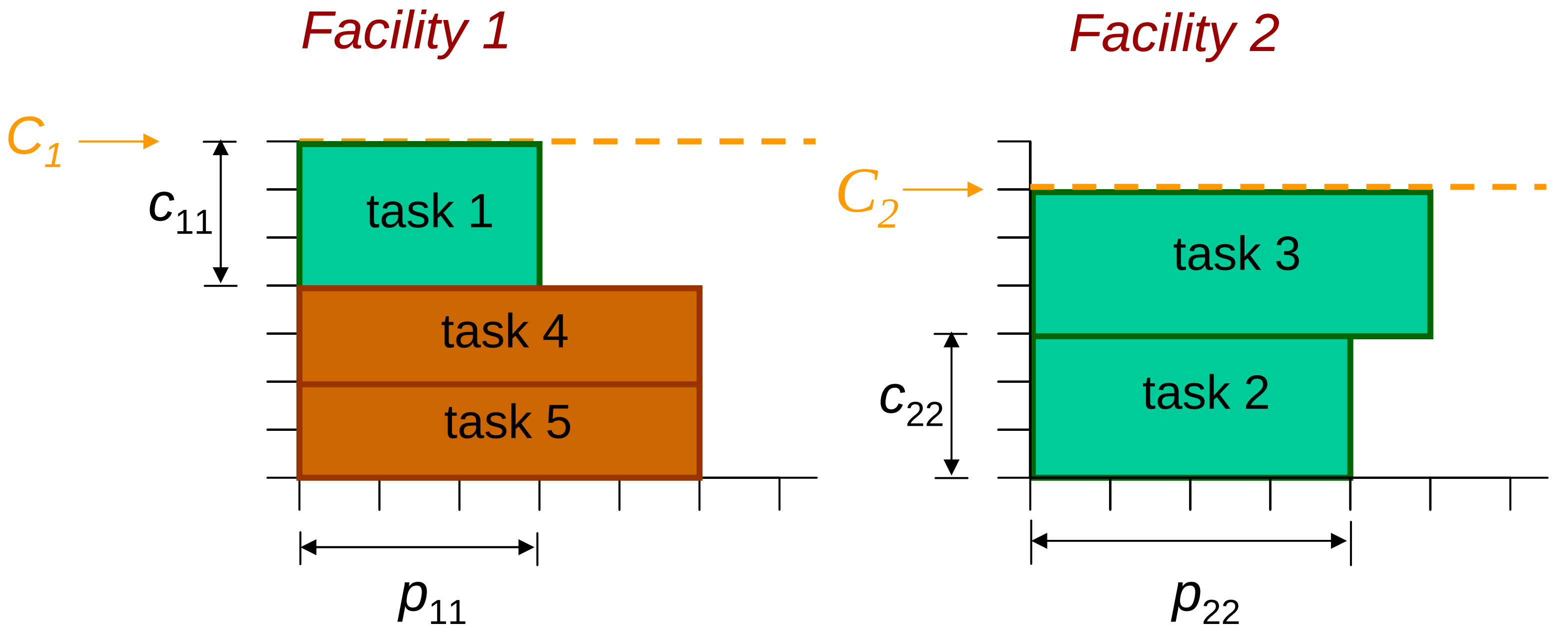
- **Decompose** problem into

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 internal information from the CP solver.

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 ≤ C_i at all times.

Objective functions

Minimize makespan = $\max_j \left\{ \left(t_j + p_{y_j j} - d_j \right)^+ \right\}$

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

facility assigned to task *j*

start time of task *j*

Minimize # late tasks = $\sum_j \delta(t_j + p_{y_j j} - d_j)$

$$\delta(\alpha) = \begin{cases} 1 \text{ if } \alpha > 0 \\ 0 \text{ if } \alpha = 0 \end{cases}$$

due date for task *j*

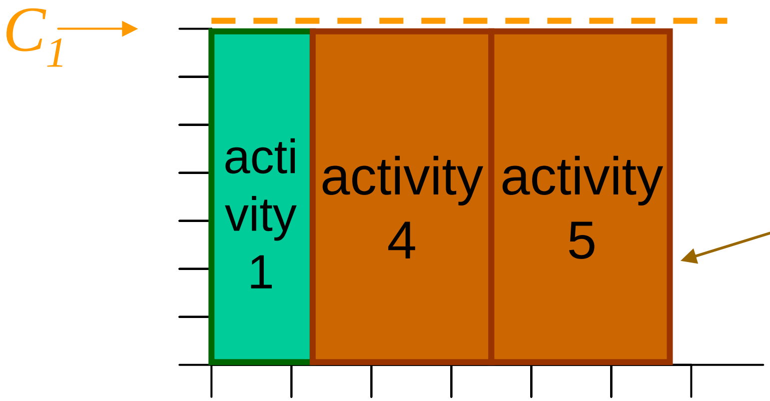
Minimize tardiness = $\sum_j \left(t_j + p_{y_j j} - d_j \right)^+$

Minimize Makespan

Master Problem: Assign tasks to facilities
Formulate as MILP problem

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

makespan



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

Subproblem: Schedule tasks assigned to each facility

Separates into an independent scheduling problem on each facility *i*.
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.*

Let M^* be min makespan. We get a Benders cut stating that any future assignment that puts jobs in J_{ih} on this machine must have makespan of at least M^*

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

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

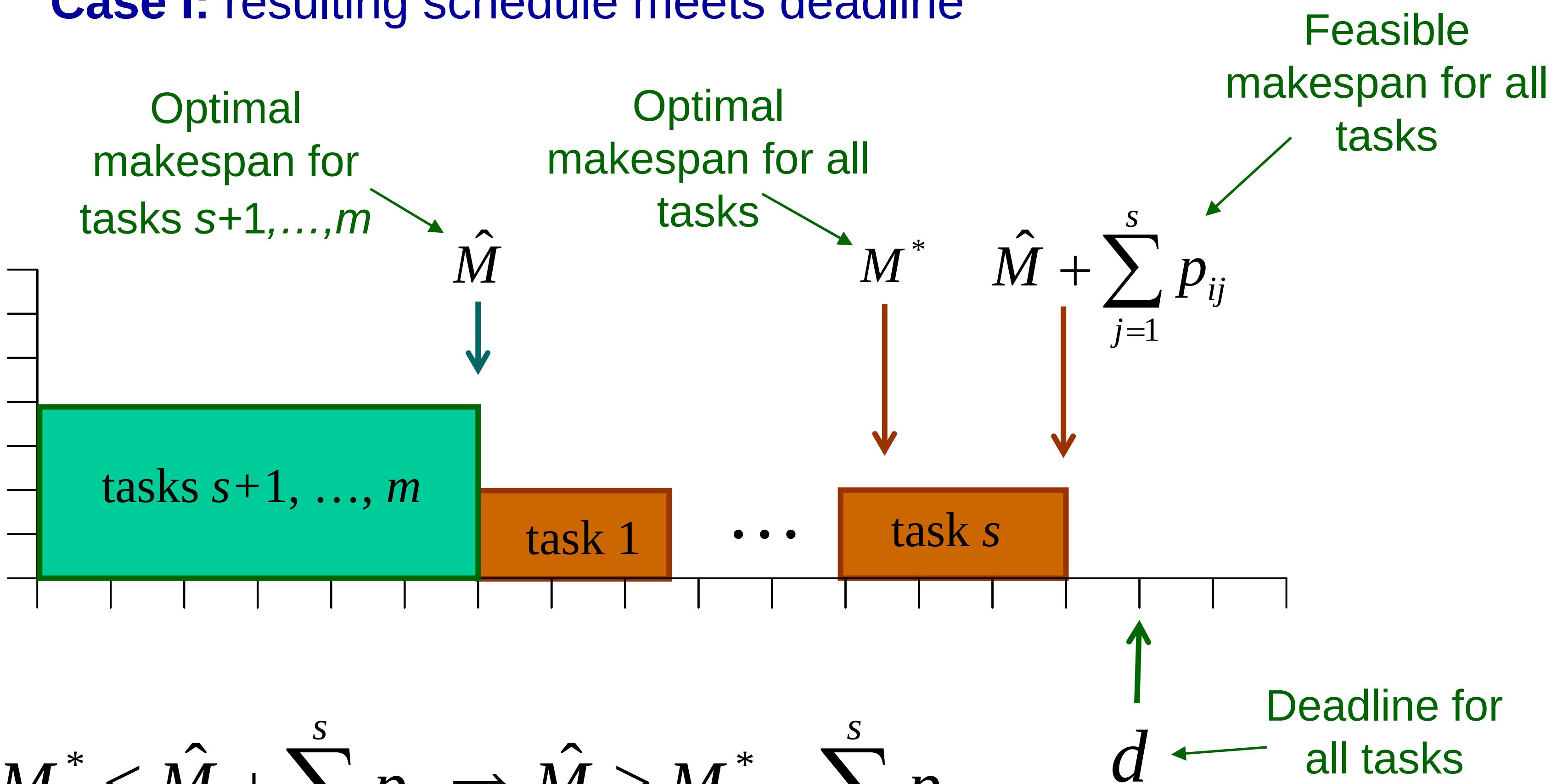
Assuming all deadlines d_i are the same, we get the Benders cut

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

Min makespan on facility *i* in last iteration

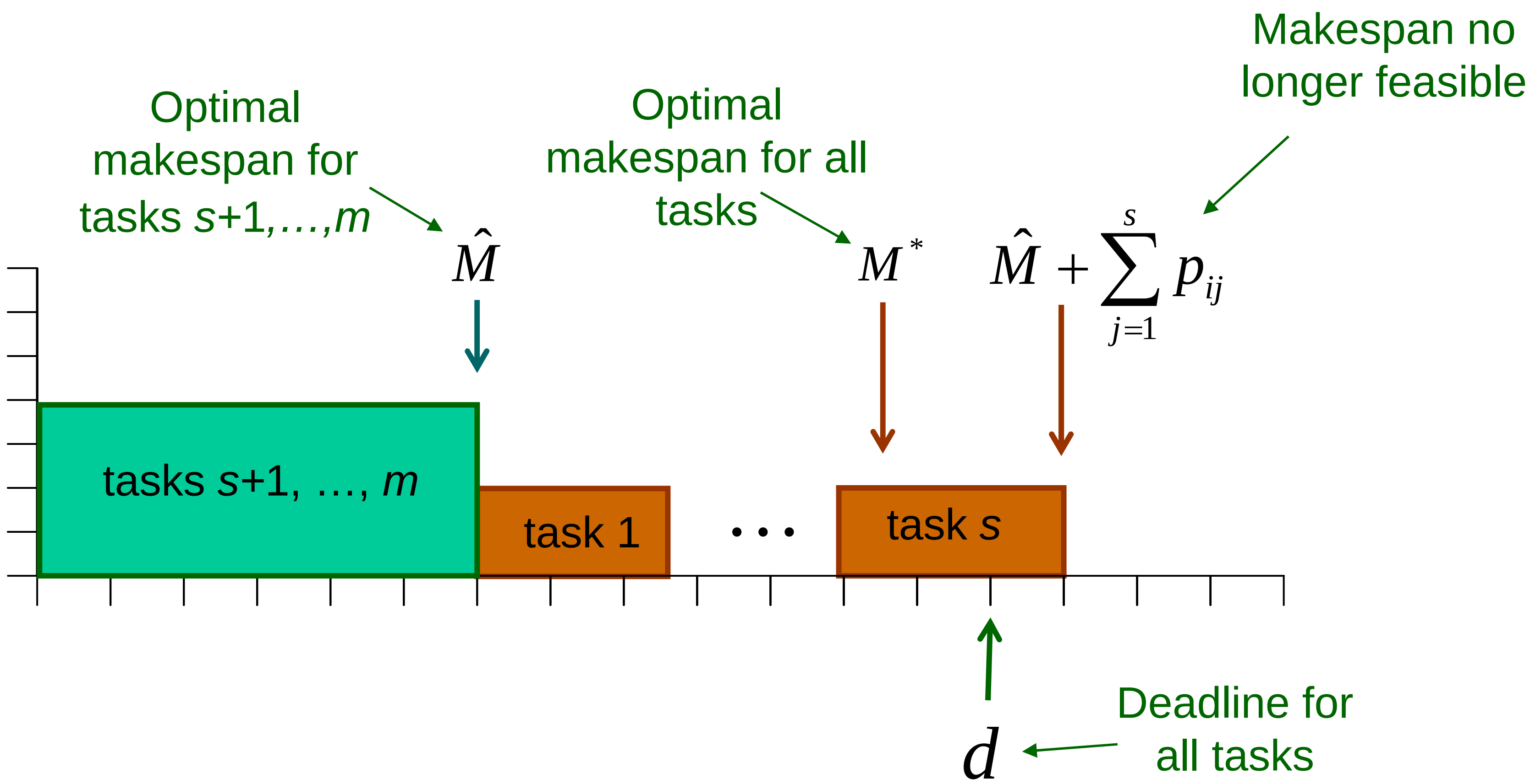
Why does this work? Assume all deadlines are the same. Add tasks 1,…,*s* sequentially at end of optimal schedule for other tasks…

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



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

Master Problem: Assign tasks to facilities

Assume all deadlines are the same

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 # Late Tasks

Master Problem: Assign tasks to facilities
Iteration h

$$\begin{aligned} \min \quad & L \\ \text{subject to} \quad & \sum_i x_{ij} = 1, \quad \text{all } j \\ & \text{Benders cuts generated in iterations } 1, \ldots, h-1 \\ & \text{relaxation of subproblem} \\ & x_{ij} \in \{0,1\} \end{aligned}$$

= 1 if task *j* is assigned to facility *i*

Subproblem: Schedule tasks assigned to each facility

Separates into an independent scheduling problem on each facility *i*.
Solve by constraint programming.

Set of tasks assigned to facility *i* by solution of master problem.

$$\begin{array}{ll} \min & \displaystyle\sum_{j \in J_{hi}} L_j \\ \text{subject to} & (t_j + p_{ij} > d_j) \Rightarrow (L_j = 1), \text{ all } j \in J_{hi} \\ & \text{cumulative}\begin{pmatrix} (t_j \mid j \in J_{hi}) \\ (p_{ij} \mid j \in J_{hi}) \\ (c_{ij} \mid j \in J_{hi}) \\ C_i \end{pmatrix} \end{array}$$

Benders Cuts

Lower bound on # late tasks on facility *i*

Min # late tasks on facility *i* (solution of subproblem)

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

Benders Cuts

Lower bound on # late tasks on facility i

Min # late tasks on facility *i* (solution of subproblem)

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

subset of J_{hi} for which min # late tasks is still L_{hi}^* (found by heuristic that repeatedly solves subproblem on facility *i*)

Benders Cuts

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

Min # late tasks on facility *i* (solution of subproblem)

To reduce # late tasks, must remove one of the tasks in J^0_{hi} from facility *i*.

Benders Cuts

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

Min # late tasks on facility *i* (solution of subproblem)

subset of J_{hi} for which min # late tasks is still L_{hi}* (found by heuristic that repeatedly solves subproblem on facility *i*)

Smaller subset of J_{hi} for which min # late tasks is L_{hi}* – 1 (found while running same heuristic)

Benders Cuts

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

Min # late tasks on facility *i* (solution of subproblem)

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

To reduce # late tasks by more than 1, must remove one of the tasks in J^1_{hi} from facility *i*.

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

Benders Cuts

$$L \geq \sum_i \hat{L}_{hi}$$

$$\hat{L}_{hi} \geq L^*_{hi} - L^*_{hi} \sum_{j \in J^0_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq L^*_{hi} - 1 - L^*_{hi} \sum_{j \in J^1_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{L}_{hi} \geq 0, \quad \text{all } i$$

These Benders cuts are added to the master problem in each iteration *h.*

Relaxation of Subproblem

Lower bound on # late tasks on facility i

$$L \geq \sum_i L_i$$

$$L_i \geq \frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{p_{ik}\}}, \text{ all } j$$

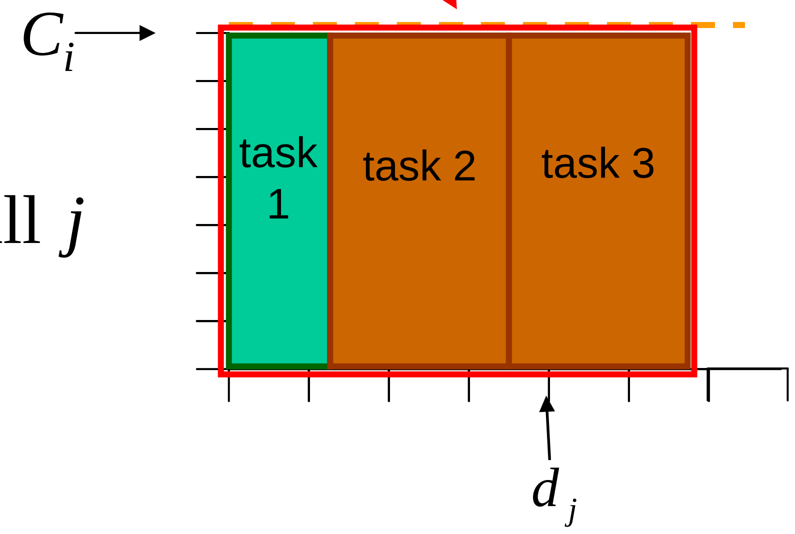
Relaxation of Subproblem

Lower bound on # late tasks on facility i

$$L \geq \sum_i L_i$$

Set of tasks assigned to facility *i* with deadline at or before d_j

$$L_i \geq \frac{\frac{1}{C_i} \sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\max_{k \in J(d_j)} \{ p_{ik} \}}, \text{ all } j$$



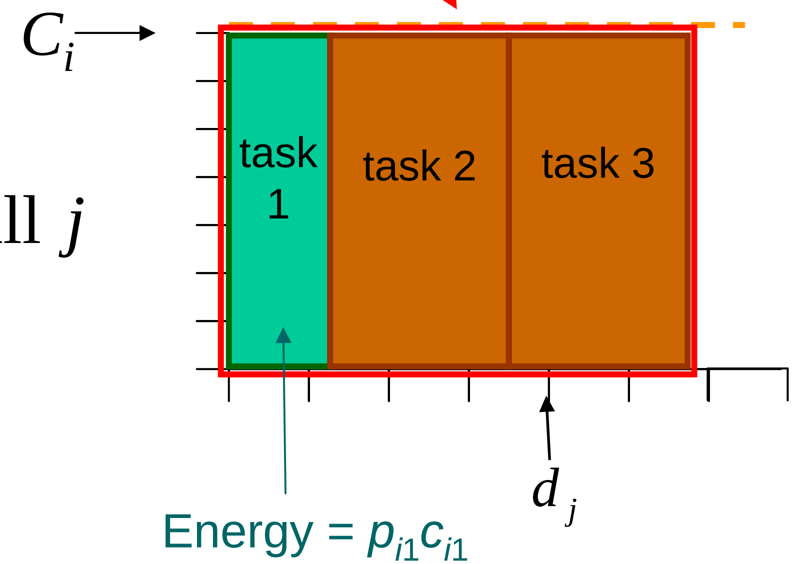
Relaxation of Subproblem

Lower bound on # late tasks on facility i

$$L \geq \sum_i L_i$$

Set of tasks assigned to facility *i* with deadline at or before d_j

$$L_i \geq \frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{ p_{ik} \}}, \text{ all } j$$



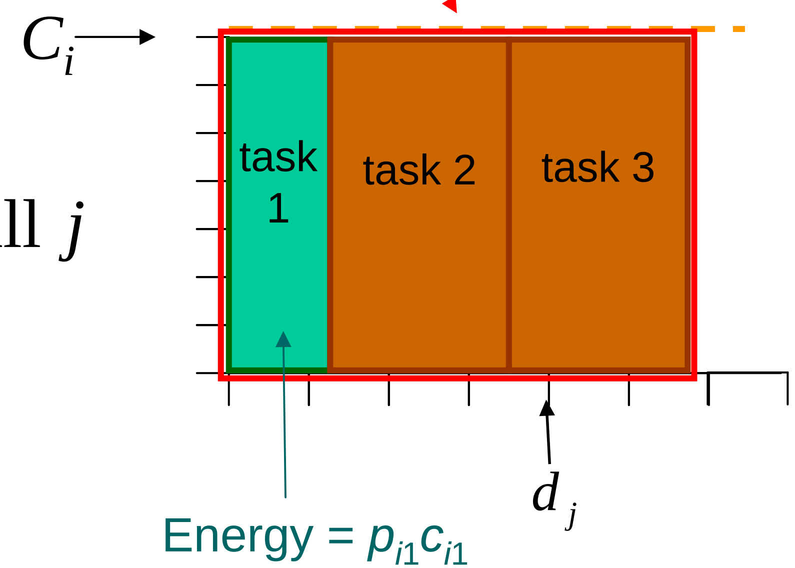
Relaxation of Subproblem

Lower bound on # late tasks on facility i

$$L \ge \sum_i L_i$$

Area of tasks assigned to facility *i* with deadline at or before d_j

$$L_i \ge \frac{\frac{1}{C_i}\sum_{k\in J(d_j)} c_{ik}\,p_{ik}\,x_{ik} - d_j}{\max\limits_{k\in J(d_j)}\{p_{ik}\}}, \text{ all } j$$

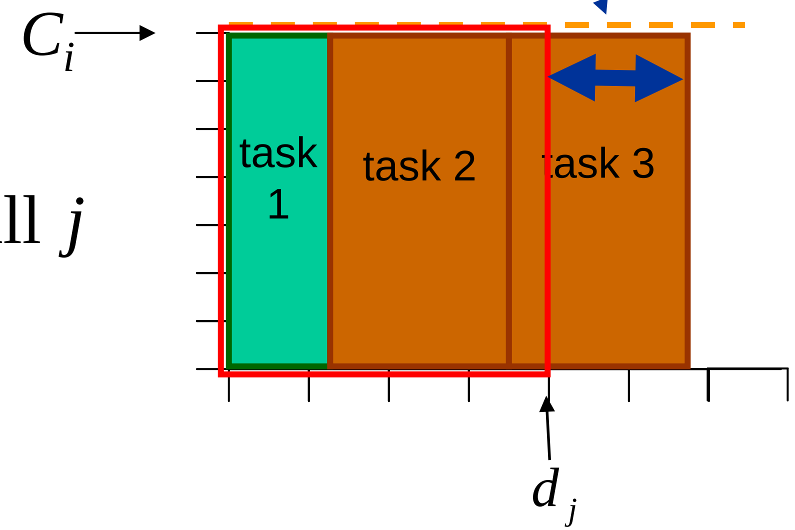


Relaxation of Subproblem

Lower bound on (makespan – latest deadline)

$$L \geq \sum_i L_i$$

$$L_i \geq \frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{p_{ik}\}}, \text{ all } j$$



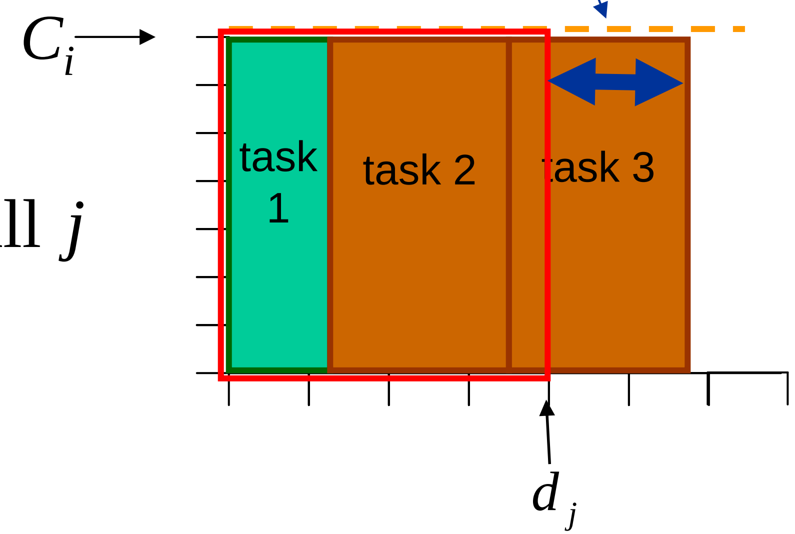
Relaxation of Subproblem

Lower bound on (makespan – latest deadline)

$$L \geq \sum_i L_i$$

$$L_i \geq \frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{p_{ik}\}}, \text{ all } j$$

Max processing time



Relaxation of Subproblem

$$L \geq \sum_i L_i$$

$$L_i \geq \boxed{\frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{p_{ik}\}}}, \text{ all } j$$

Min # of late jobs on facility *i*

Relaxation of Subproblem

$$L \geq \sum_i L_i$$

$$L_i \geq \boxed{\frac{\dfrac{1}{C_i} \displaystyle\sum_{k \in J(d_j)} c_{ik} p_{ik} x_{ik} - d_j}{\displaystyle\max_{k \in J(d_j)} \{p_{ik}\}}}, \text{ all } j$$

Min # of late jobs on facility *i*

This relaxation is added to the master problem at the outset.

Minimize Total Tardiness

Master Problem: Assign tasks to facilities

Iteration h

$$
\begin{array}{ll}
\min & L \\
\text{subject to} & \sum_i x_{ij} = 1, \quad \text{all } j \\
 & \text{Benders cuts generated in iterations } 1, \ldots, h-1 \\
 & \text{relaxation I of subproblem} \\
 & \text{relaxation II of subproblem} \\
 & x_{ij} \in \{0,1\}
\end{array}
$$

x_{ij} = 1 if task *j* is assigned to facility *i*

Subproblem: Schedule tasks assigned to each facility

Separates into an independent scheduling problem on each facility *i*.
Solve by constraint programming.

Set of tasks assigned to facility *i* by solution of master problem.

$$\begin{array}{ll} \min & \sum_{j \in J_{hi}} T_j \\ \text{subject to} & T_j \ge t_j + p_{ij} - d_j, \text{ all } j \in J_{hi} \\ & \text{cumulative}\begin{pmatrix} (t_j \mid j \in J_{hi}) \\ (p_{ij} \mid j \in J_{hi}) \\ (c_{ij} \mid j \in J_{hi}) \\ C_i \end{pmatrix} \end{array}$$

Benders Cuts

Lower bound on tardiness for facility *i*

Min tardiness on facility *i* (solution of subproblem)

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T^*_{hi} - T^*_{hi} \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq T^0_{hi} - T^0_{hi} \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

Benders Cuts

Lower bound on tardiness for facility *i*

Min tardiness on facility *i* (solution of subproblem)

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T^*_{hi} - T^*_{hi} \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

To reduce tardiness on facility *i*, must remove one of the tasks assigned to it.

$$\hat{T}_{hi} \geq T^0_{hi} - T^0_{hi} \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

Benders Cuts

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T^*_{hi} - T^*_{hi} \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq T^0_{hi} - T^0_{hi} \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

Min tardiness on facility *i* (solution of subproblem)

Set of tasks that can be removed, one at a time from facility *i* without reducing min tardiness.

Benders Cuts

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T_{hi}^* - T_{hi}^* \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq T_{hi}^0 - T_{hi}^0 \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

Set of tasks that can be removed, one at a time from facility *i* without reducing min tardiness.

Min tardiness on facility *i* when all tasks in Z_{hi} are removed *simultaneously*.

Benders Cuts

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T^*_{hi} - T^*_{hi} \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq T^0_{hi} - T^0_{hi} \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

To reduce tardiness below T^0_{hi} on facility *i*, must remove one of the tasks in $J_{hi} \setminus Z_{hi}$

Set of tasks that can be removed, one at a time from facility *i* without reducing min tardiness.

Min tardiness on facility *i* when all tasks in Z_{hi} are removed *simultaneously*.

Benders Cuts

$$T \geq \sum_i \hat{T}_{hi}$$

$$\hat{T}_{hi} \geq T_{hi}^* - T_{hi}^* \sum_{j \in J_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq T_{hi}^0 - T_{hi}^0 \sum_{j \in J_{hi} \setminus Z_{hi}} (1 - x_{ij}), \quad \text{all } i$$

$$\hat{T}_{hi} \geq 0, \quad \text{all } i$$

These Benders cuts are added to the master problem in each iteration *h.*

Subproblem Relaxation I

Lower bound on total tardiness for facility *i*

$$T \geq \sum_i T_i$$

$$T_i \geq \frac{1}{C_i} \sum_{j \in J(d_k)} c_{ij} p_{ij} x_{ij} - d_k, \text{ all } k$$

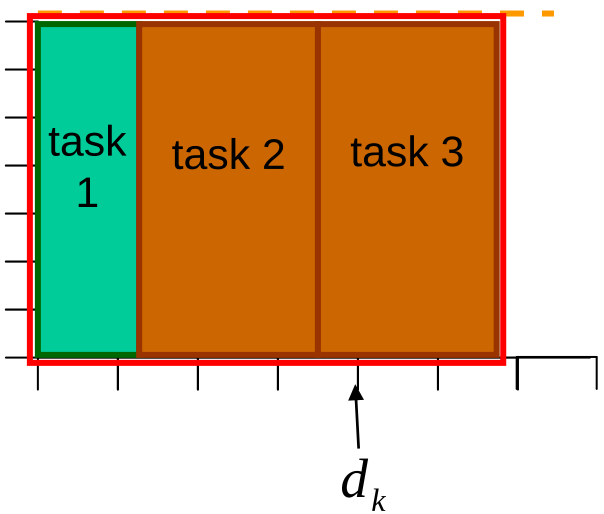
Subproblem Relaxation I

Lower bound on total tardiness for facility *i*

Set of tasks assigned to facility *i* with deadline at or before d_k

$$T \geq \sum_i T_i$$

$$T_i \geq \frac{1}{C_i} \sum_{j \in J(d_k)} c_{ij} p_{ij} x_{ij} - d_k, \ \text{all } k$$



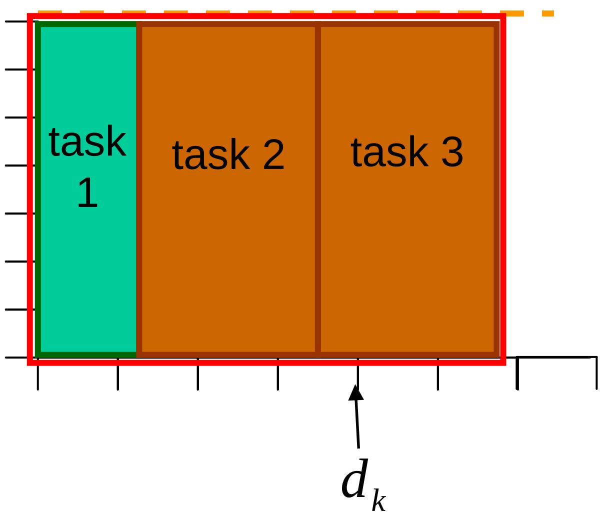
Subproblem Relaxation I

Lower bound on total tardiness for facility *i*

$$T \ge \sum_i T_i$$

Area of tasks assigned to facility *i* with deadline at or before d_k

$$T_i \ge \frac{1}{C_i} \sum_{j \in J(d_k)} c_{ij} p_{ij} x_{ij} - d_k, \ \text{all } k$$



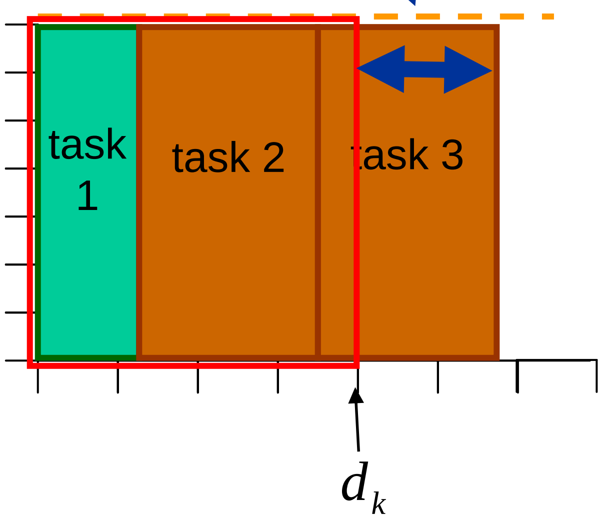
Subproblem Relaxation I

Lower bound on total tardiness for facility *i*

Lower bound on total tardiness

$$T \geq \sum_i T_i$$

$$T_i \geq \frac{1}{C_i} \sum_{j \in J(d_k)} c_{ij} p_{ij} x_{ij} - d_k, \text{ all } k$$



Subproblem Relaxation II

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

$$\overline{T} = \sum_{k=1}^{n} \overline{T}_k$$

where

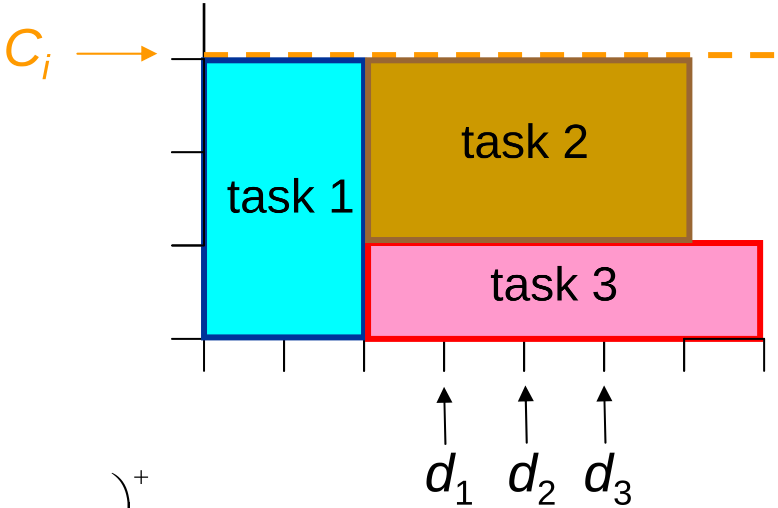
$$\overline{T}_k = \left(\frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} - d_k \right)^+$$

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

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

Example of Lemma

j	d_j	p_{ij}	c_{ij}	$p_{ij}c_{ij}$
1	3	2	3	6
2	4	4	2	8
3	5	5	1	5



$$\bar{T}_1 = \left(\frac{1}{C_i}(p_{i3}c_{i3}) - d_1\right)^+ = \left(\frac{1}{3}(5) - 3\right)^+ = 0$$

$$\bar{T}_2 = \left(\frac{1}{C_i}(p_{i3}c_{i3} + p_{i1}c_{i1}) - d_2\right)^+ = \left(\frac{1}{3}(5+6) - 4\right)^+ = 0$$

$$\bar{T}_3 = \left(\frac{1}{C_i}(p_{i3}c_{i3} + p_{i1}c_{i1} + p_{i2}c_{i2}) - d_3\right)^+ = \left(\frac{1}{3}(5+6+8) - 5\right)^+ = 4/3$$

$$\text{Lower bound on tardiness} = \left\lceil \bar{T}_1 + \bar{T}_2 + \bar{T}_3 \right\rceil = \lceil 4/3 \rceil = 2$$

Min tardiness = 4

Idea of proof

For a permutation σ of 1,…,*n* let $T(\sigma) = \sum_{k=1}^{n} T_k(\sigma)$

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

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

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

Now we have

swap k and $k+1$

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

since $T^* = \sum_{j=1}^{n} \left(t_{\sigma_0(j)} + p_{i\sigma_0(j)} - d_{\sigma_0(j)}\right)^+ \ge \sum_{j=1}^{n} \left(\frac{1}{C_i} \sum_{j=1}^{k} p_{i\sigma_0(j)} c_{i\sigma_0(j)} - d_{\sigma_0(j)} \right)^+ \ge \sum_{j=1}^{n} \left(\frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} - d_{\sigma_0(j)} \right)^+ = T(\sigma_0)$

areas

def. of π

$$T(\sigma_s) = \sum_{j=1}^{k-1} T_j(\sigma_s) + T_k(\sigma_s) + T_{k+1}(\sigma_s) + \sum_{j=k+2}^{n} T_j(\sigma_s)$$

$$T(\sigma_{s+1}) = \sum_{j=1}^{k-1} T_j(\sigma_s) + T_k(\sigma_{s+1}) + T_{k+1}(\sigma_{s+1}) + \sum_{j=k+2}^{n} T_j(\sigma_s)$$

So

$$\begin{aligned} T(\sigma_s) - T(\sigma_{s+1}) &= T_k(\sigma_s) + T_{k+1}(\sigma_s) - T_k(\sigma_{s+1}) - T_{k+1}(\sigma_{s+1}) \\ &= (a-A)^+ + (A-b)^+ - (a-b)^+ - (A-B)^+ \ge 0 \end{aligned}$$

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

Writing relaxation II

From the lemma, we can write the relaxation

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

where $T'_{ik} \geq \frac{1}{C_i} \sum_{j=1}^{k} p_{i\pi_i(j)} c_{i\pi_i(j)} x_{i\pi_i(j)} - d_k$

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

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

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

Computational Results

- Random problems on 2, 3, 4 facilities.
- Facilities run at different speeds.
- All release times = 0.
  - Min cost and makespan problems: deadlines same/different.
  - 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 makespan, 2 facilities

Average of 5 instances shown

Jobs	MILP	CP	Benders
10	3.4	0.8	0.24
12	12	4.0	0.31
14	2572+	299	5.0
16	5974+	3737	36
18		7200+	233
20			1268

+ 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	0.23
12	12	7.5	0.38
14	524	981	1.4
16	1716+	4414	7.6
18	4619+	7200+	30
20			8.7
22			2012+

+ 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	0.19
12	5.0	1.9	0.43
14	24	524	0.82
16	35	3898	1.0
18	3931+	7200+	6.4
20			4.4
22			28
24			945

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

Min makespan, 3 facilities
Different deadlines

Average of 5 instances shown

Jobs	MILP	CP	Benders
14	223	7.1	4.4
16	853	1620+	5.1
18	350	1928+	2.9
20	7200+	7200+	1449+
22	7200+		388
24	7200+		132

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

Min # late tasks

Smaller problems

Tasks	Time (sec) CP	MILP	Benders	Min # late tasks
14	1092	5.8	0.5	1
	382	8.0	0.7	1
	265	3.2	0.7	2
	85	2.6	1.3	2
	5228	1315	665	3
16	304	2.7	0.5	0
	?	31	0.2	1
	310	22	0.4	1
	4925	29	2.7	2
	19	5.7	24	4
18	>7200	2.0	0.1	0
	?	8.0	0.2	1
	>7200	867	8.5	1
	>7200	6.3	1.4	2
	>7200	577	3.4	2

Min # late tasks

Larger problems

Tasks	Time (sec) MILP	Time (sec) Benders	Best solution MILP	Best solution Benders
20	97	0.4	0	0
	>7200	2.3	(1)	1
	219	5.0	1	1
	>7200	11	(2)	2
	843	166	3	3
22	16	1.3	0	0
	>7200	3.7	(1)	1
	>7200	49	(3)	2
	>7200	3453	(5)	2
	>7200	>7200	(6)	(6)
24	25	0.8	0	0
	>7200	18	(1)	0
	>7200	62	(2)	0
	>7200	124	(3)	1
	>7200	234	(2)	1

( ) = optimality not proved

Effect of subproblem relaxation

Min # late tasks

Tasks	Time (sec) with relax	Time (sec) without relax
16	0.5	2.6
	0.4	1.5
	0.2	1.3
	2.7	4.2
	24	18
18	0.1	1.1
	0.2	0.7
	3.4	3.3
	1.4	15
	8.5	11
20	0.4	88
	2.3	9.7
	5.0	63
	11	19
	166	226

Min total tardiness

Smaller problems

Tasks	Time (sec) CP	MILP	Benders	Min tardiness
14	838	7.0	6.1	1
	7159	34	3.7	2
	1783	45	19	15
	>7200	73	40	19
	>7200	>7200	3269	26
16	>7200	19	1.4	0
	>7200	46	2.1	0
	>7200	52	4.2	4
	>7200	1105	156	20
	>7200	3424	3.4	31
18		187	2.8	0
		15	5.3	3
		46	49	5
		256	47	11
		>7200	1203	14

Min total tardiness

Larger problems

Tasks	Time (sec)		Best solution	
	MILP	Benders	MILP	Benders
20	105	18	0	0
	4141	23	1	1
	39	29	4	4
	1442	332	8	8
	>7200	>7200	(75)	(37)
22	6	19	0	0
	584	37	2	2
	>7200	>7200	(120)	(40)
	>7200	>7200	(162)	(46)
	>7200	>7200	(375)	(141)
24	10	324	0	0
	>7200	94	(20)	0
	>7200	110	(57)	0
	>7200	>7200	(20)	(5)
	>7200	>7200	(25)	(7)

( ) = optimality not proved

Effect of subproblem relaxation

Min total tardiness

Tasks	Time (sec) with relax	Time (sec) without relax
16	1.4	4.4
	2.1	6.5
	4.2	30
	156	199
	765	763
18	2.8	10
	5.3	17
	47	120
	49	354
	1203	5102
20	18	151
	23	1898
	29	55
	332	764
	>7200	>7200

Conclusions

- Benders is significantly faster than MILP (which is faster than CP).
    - Speedup for min makespan, # late tasks is often orders of magnitude.
    - Smaller speedup for min total tardiness.
    - Problems with more than a few late tasks are hard for all methods.
- Benders finds better suboptimal solutions.
    - Even when Benders fails to prove optimality, it obtains much better solutions than MILP in the same time period.
- Subproblem relaxation is important.
    - Linear relaxation of *cumulative* constraint can be critical to performance, especially when minimizing total tardiness.

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