A Tour of Modeling Techniques

John Hooker
Carnegie Mellon University

June 2008

Outline

- Mixed integer linear (MILP) modeling
  - Disjunctive modeling
    - Examples: fixed charge problems, facility location, lot sizing with setup costs.
  - Knapsack modeling
    - Examples: Freight packing and transfer
- Constraint programming models
    - Example: Employee scheduling
- Integrated Models
    - Examples: Product configuration, machine scheduling

Mixed Integer/Linear Modeling

MILP Modeling Systems

MILP Models

Disjunctive Modeling

Knapsack Modeling

MILP Modeling Systems

- Commercial modeling systems
  - AMPL
  - GAMS
  - AIMMS

MILP Modeling Systems

- Commercial modeling systems with dedicated solvers
  - OPL Studio (runs CPLEX)
  - Xpress-BCL (runs Xpress-MP)
  - Xpress-Mosel (runs Xpress-MP)
  - Excel and Quattro Pro, Frontline Systems (spreadsheet based)
  - LINGO
  - MINOPT (also nonlinear)

MILP Modeling Systems

- Non-commercial modeling systems
  - ZIMPL
  - Gnu Mathprog (GMPL)

MILP models

An **mixed integer linear programming** (MILP) model has the form

$$\begin{aligned} &\min\ cx + dy \\ &Ax + by \ge b \\ &x, y \ge 0 \\ &y \text{ integer} \end{aligned}$$

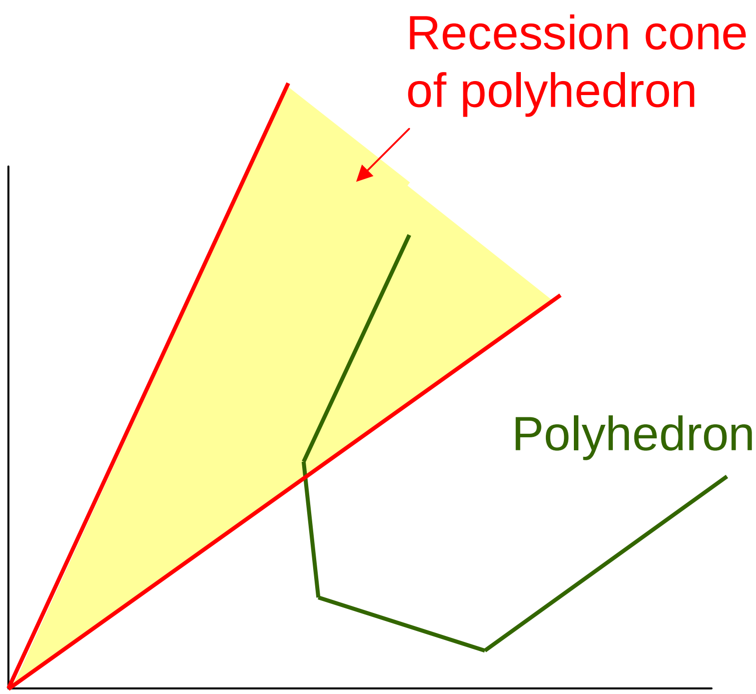
A principled approach to MILP modeling

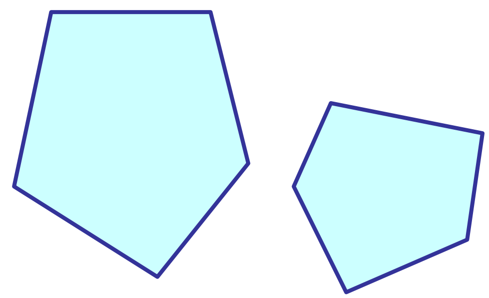
- MILP modeling combines two distinct kinds of modeling.
  - Modeling of subsets of continuous space, using 0-1 auxiliary variables.
  - Knapsack modeling, using general integer variables.
- MILP can model subsets of continuous space that are unions of polyhedra.
  - …that is, represented by disjunctions of linear systems.
- So a principled approach is to analyze the problem as

disjunctions of linear systems	+	integer knapsack inequalities

Disjunctive Modeling

Theorem. A subset of continuous space can be represented by an MILP model if and only if it is the union of finitely many polyhedra having the same recession cone.





Union of polyhedra with the same recession cone (in this case, the origin)

Modeling a union of polyhedra

Start with a disjunction of linear systems to represent the union of polyhedra.

The kth polyhedron is $\{x \mid A^k x \geq b\}$

$$\min\ cx$$
$$\bigvee_k \left(A^k x \geq b^k \right)$$

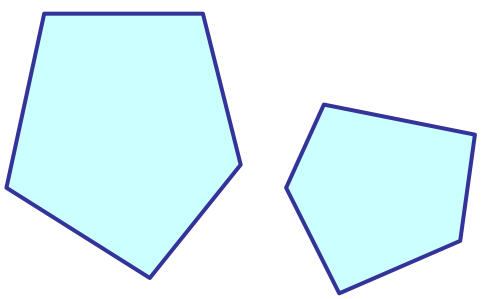
Introduce a 0-1 variable y_k that is 1 when x is in polyhedron k.

Disaggregate x to create an x^k for each k.

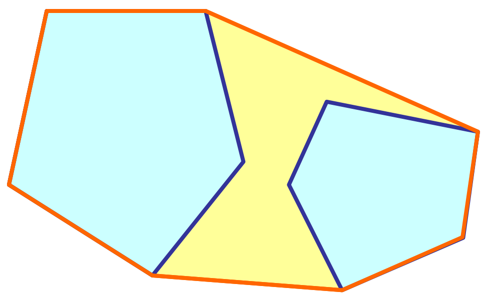
$$\min\ cx$$
$$A^k x^k \geq b^k y_k,\ \text{all } k$$
$$\sum_k y_k = 1$$
$$x = \sum_k x^k$$
$$y_k \in \{0,1\}$$

Tight Relaxations

- **Basic fact:** The continuous relaxation of the disjunctive MILP model provides a **convex hull relaxation** of the disjunction.
    - This is the tightest possible linear model for the disjunction.



Union of polyhedra



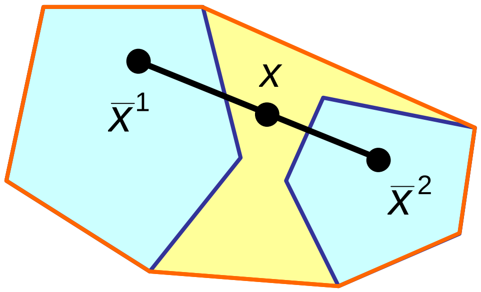
Convex hull relaxation
(tightest linear relaxation)

Tight Relaxations

To derive convex hull relaxation of a disjunction…

Write each solution as a convex combination of points in the polyhedron

$$\begin{aligned} &\min \ cx \\ &A^k \bar{x}^k \ge b^k, \text{ all } k \\ &\sum_k y_k = 1 \\ &x = \sum_k y_k \bar{x}^k \\ &y_k \in \{0,1\} \end{aligned}$$



Convex hull relaxation (tightest linear relaxation)

Tight Relaxations

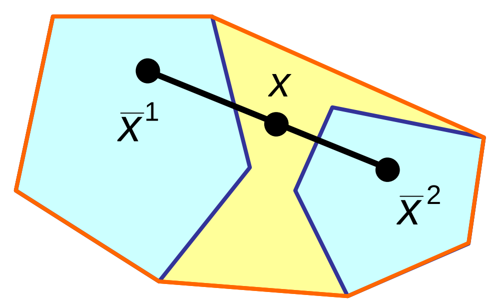
To derive convex hull relaxation of a disjunction…

Write each solution as a convex combination of points in the polyhedron

$$\begin{aligned} &\min\ cx \\ &A^k \bar{x}^k \geq b^k, \text{ all } k \\ &\sum_k y_k = 1 \\ &x = \sum_k y_k \bar{x}^k \\ &y_k \in \{0,1\} \end{aligned}$$

Change of variable $x = y_k \bar{x}^k$

$$\begin{aligned} &\min\ cx \\ &A^k x^k \geq b^k y_k, \text{ all } k \\ &\sum_k y_k = 1 \\ &x = \sum_k x^k \\ &y_k \in \{0,1\} \end{aligned}$$



Convex hull relaxation (tightest linear relaxation)

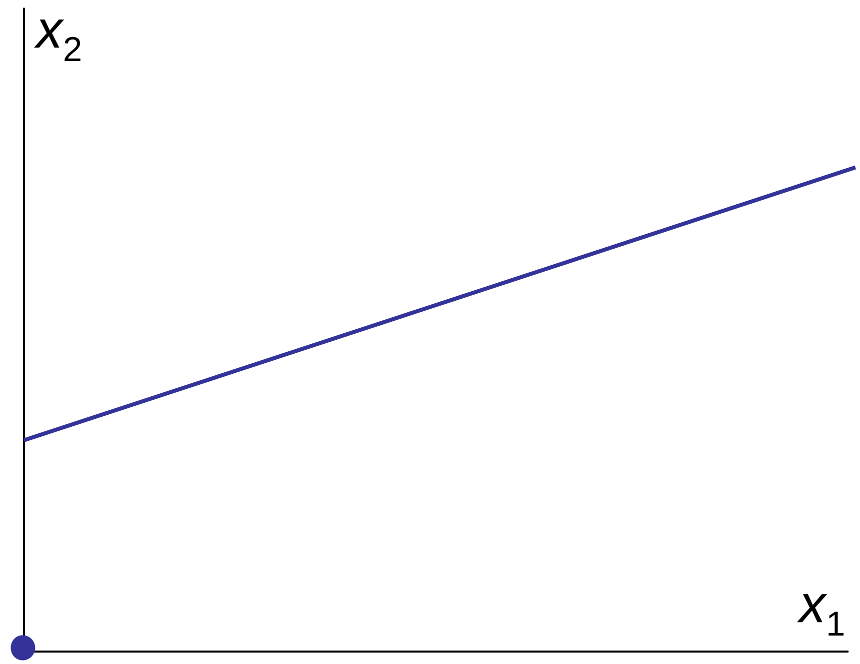
Example: Fixed charge function

Minimize a fixed charge function:

$$\min \; x_2$$

$$x_2 \geq \begin{cases} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{cases}$$

$$x_1 \geq 0$$



Fixed charge problem

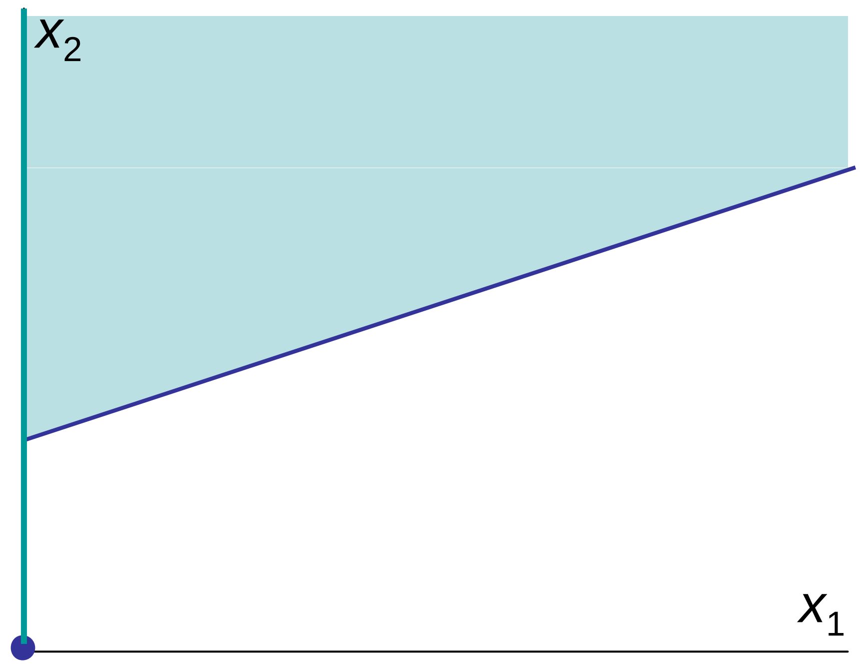
Minimize a fixed charge function:

$$\min \ x_2$$

$$x_2 \geq \begin{Bmatrix} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{Bmatrix}$$

$$x_1 \geq 0$$

Feasible set

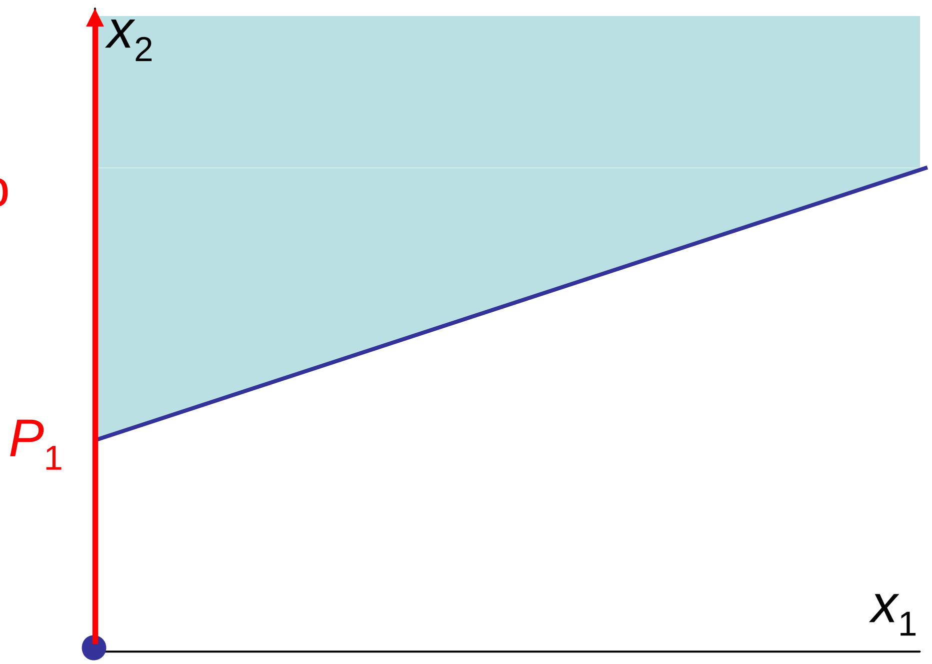


Fixed charge problem

Minimize a fixed charge function:

$$\min\ x_2$$
$$x_2 \geq \begin{Bmatrix} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{Bmatrix}$$
$$x_1 \geq 0$$

Union of two polyhedra P_1, P_2

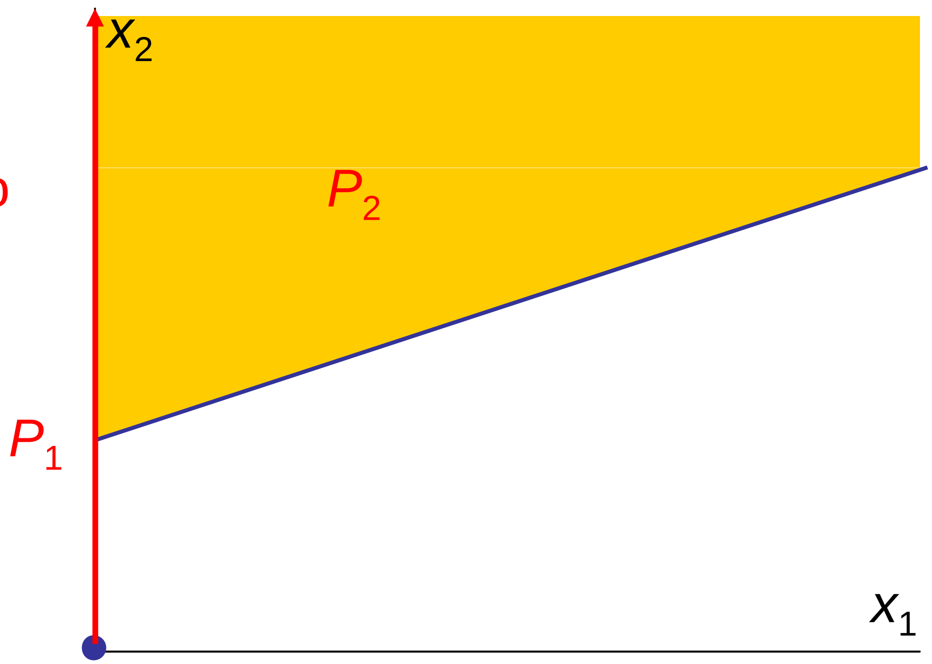


Fixed charge problem

Minimize a fixed charge function:

$$\min \ x_2$$
$$x_2 \geq \begin{Bmatrix} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{Bmatrix}$$
$$x_1 \geq 0$$

Union of two polyhedra
P_1, P_2

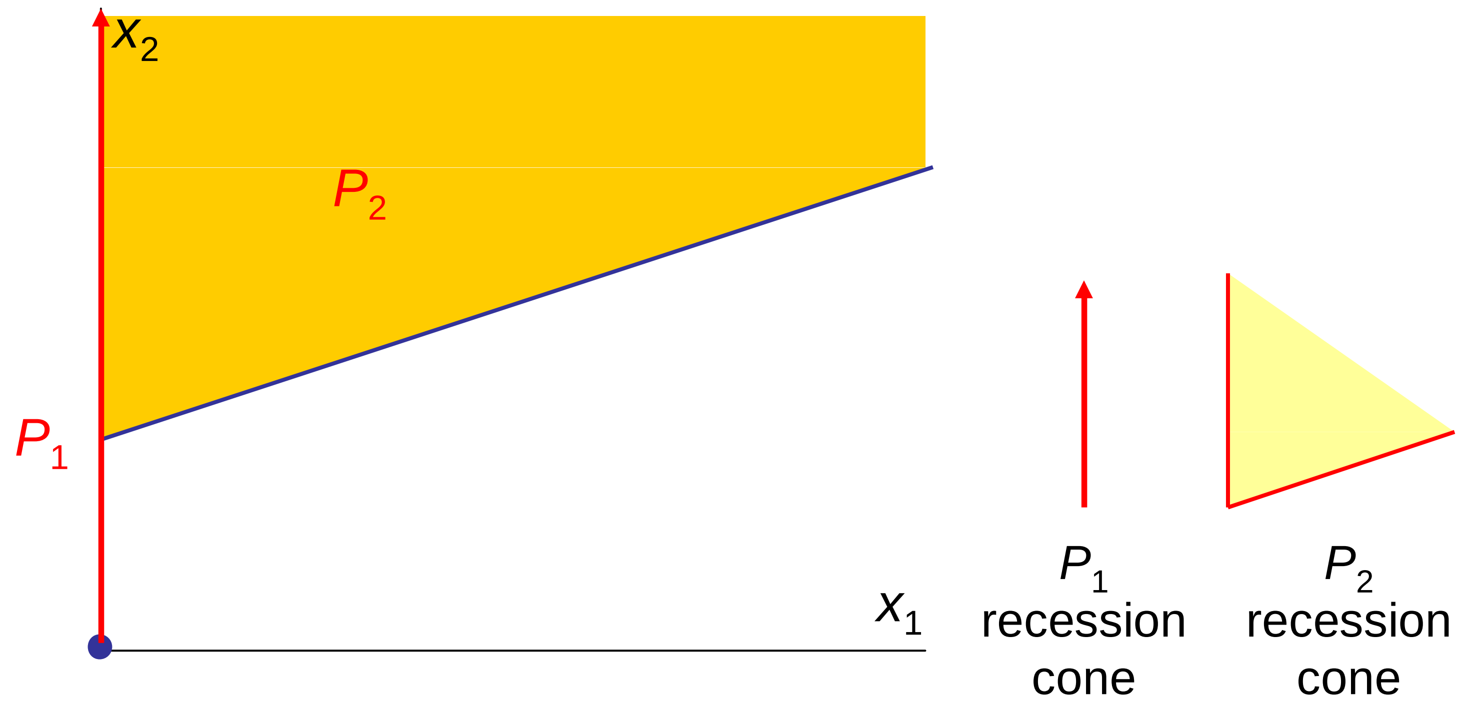


Fixed charge problem

Minimize a fixed charge function:

$$\min \; x_2$$
$$x_2 \geq \begin{Bmatrix} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{Bmatrix}$$
$$x_1 \geq 0$$

The polyhedra have different recession cones.



Fixed charge problem

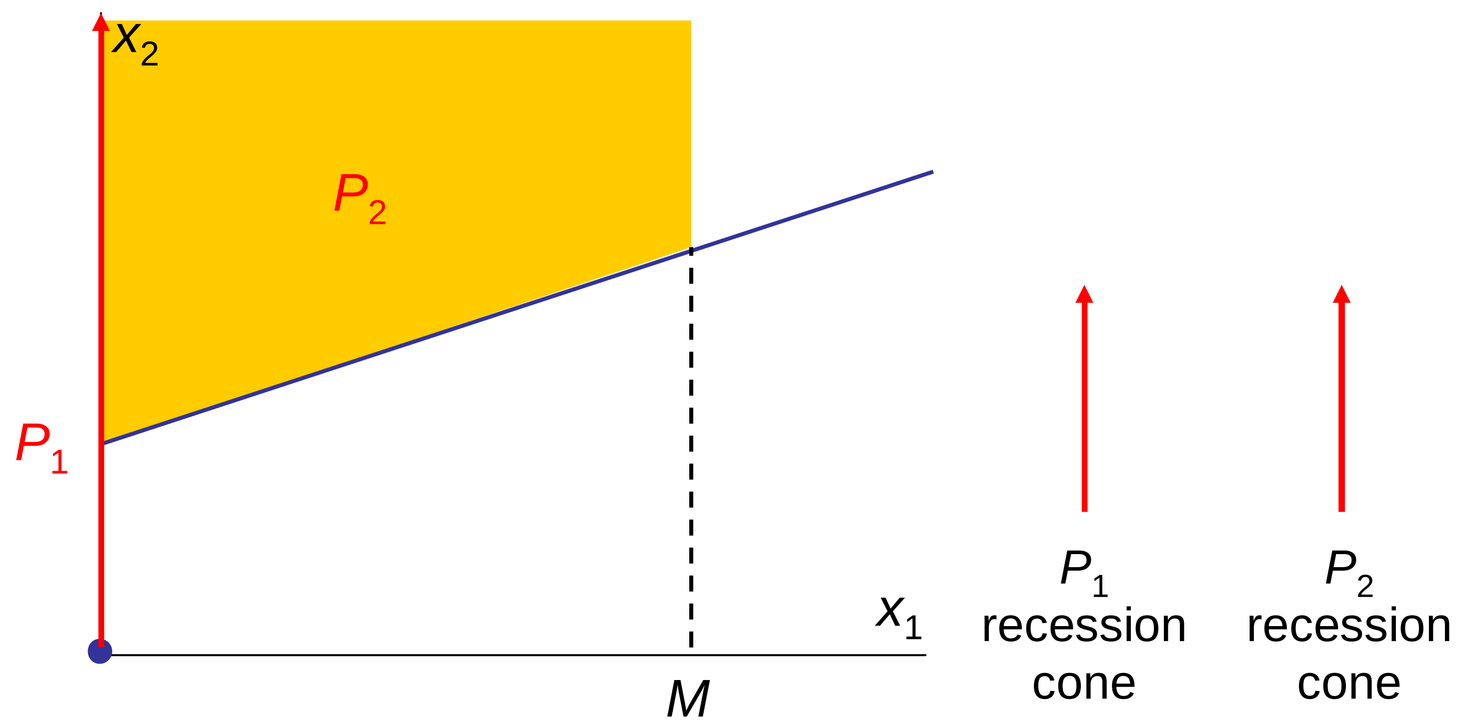
Minimize a fixed charge function:

$$\min \; x_2$$

$$x_2 \geq \begin{Bmatrix} 0 & \text{if } x_1 = 0 \\ f + cx_1 & \text{if } x_1 > 0 \end{Bmatrix}$$

Add an upper bound on x_1

$$0 \leq x_1 \leq M$$

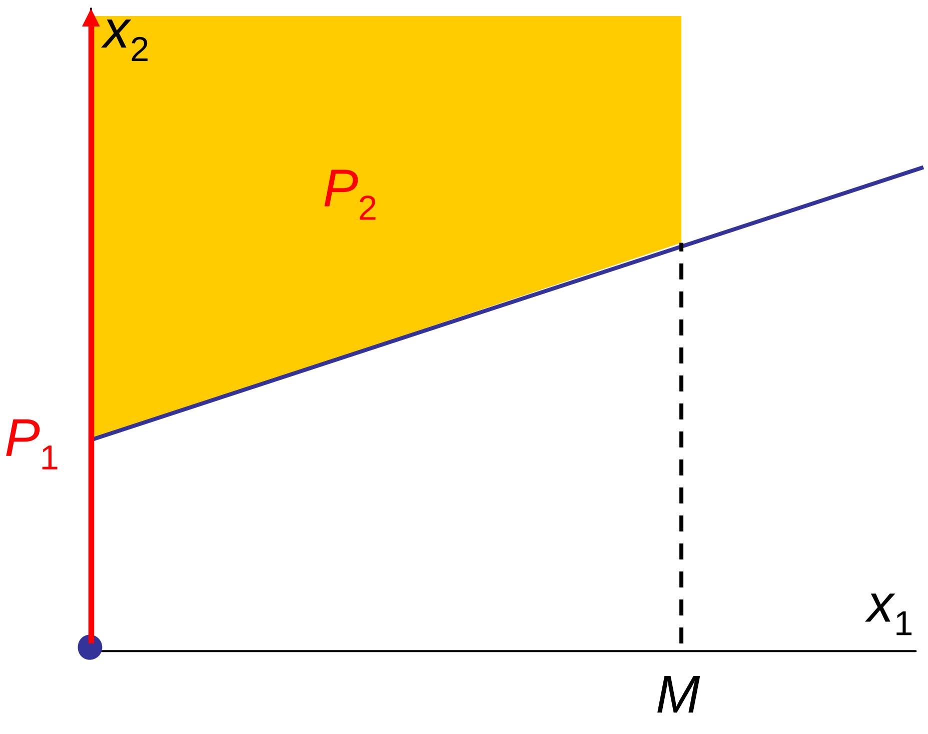


The polyhedra have the same recession cone.

Fixed charge problem

Start with a disjunction of linear systems to represent the union of polyhedra

$$\min \; x_2$$
$$\begin{pmatrix} x_1 = 0 \\ x_2 \geq 0 \end{pmatrix} \vee \begin{pmatrix} 0 \leq x_1 \leq M \\ x_2 \geq f + cx_1 \end{pmatrix}$$



Fixed charge problem

Start with a disjunction of linear systems to represent the union of polyhedra

$$\min \; x_2$$
$$\begin{pmatrix} x_1 = 0 \\ x_2 \geq 0 \end{pmatrix} \vee \begin{pmatrix} 0 \leq x_1 \leq M \\ x_2 \geq f + cx_1 \end{pmatrix}$$

Introduce a 0-1 variable y_k that is 1 when x is in polyhedron k.

Disaggregate x to create an x^k for each k.

$$\min \; x_2$$
$$x_1^1 = 0 \qquad 0 \leq x_1^2 \leq My_2$$
$$x_2^1 \geq 0 \qquad -cx_1^2 + x_2^2 \geq fy_2$$
$$y_1 + y_2 = 1, \;\; y_k \in \{0,1\}$$
$$x_1 = x_1^1 + x_1^2, \quad x_2 = x_2^1 + x_2^2$$

To simplify, replace x_1^2 with x_1 since $x_1^1 = 0$

$$\min \; x_2$$

$$x_1^1 = 0 \qquad 0 \le x_1^2 \le My_2$$

$$x_2^1 \ge 0 \qquad -cx_1^2 + x_2^2 \ge fy_2$$

$$y_1 + y_2 = 1, \;\; y_k \in \{0,1\}$$

$$x_1 = x_1^1 + x_1^2, \quad x_2 = x_2^1 + x_2^2$$

To simplify, replace x_1^2 with x_1
since $x_1^1 = 0$

$$\min \ x_2$$

$$0 \le x_1 \le My_2$$

$$x_2^1 \ge 0 \qquad -cx_1 + x_2^2 \ge fy_2$$

$$y_1 + y_2 = 1, \ \ y_k \in \{0,1\}$$

$$x_2 = x_2^1 + x_2^2$$

Replace x_2^2 with x_2

because x_2^1 plays no role in the model

$$\min \; x_2$$

$$0 \le x_1 \le My_2$$

$$x_2^1 \ge 0 \qquad -cx_1 + x_2^2 \ge fy_2$$

$$y_1 + y_2 = 1, \;\; y_k \in \{0,1\}$$

$$x_2 = x_2^1 + x_2^2$$

Replace x_2^2 with x_2

Because x_2^1 plays no role in the model

$$\begin{aligned} &\min \;\; x_2 \\ &\qquad 0 \le x_1 \le My_2 \\ &\quad -cx_1 + x_2 \ge fy_2 \\ &y_1 + y_2 = 1, \;\; y_k \in \{0,1\} \end{aligned}$$

Replace y_2 with y

Because y_1 plays no role in the model

$$\begin{aligned} \min\ & x_2 \\ & 0 \le x_1 \le My_2 \\ & -cx_1 + x_2 \ge fy_2 \\ & y_1 + y_2 = 1,\ \ y_k \in \{0,1\} \end{aligned}$$

Replace y_2 with y

Because y_1 plays no role in the model

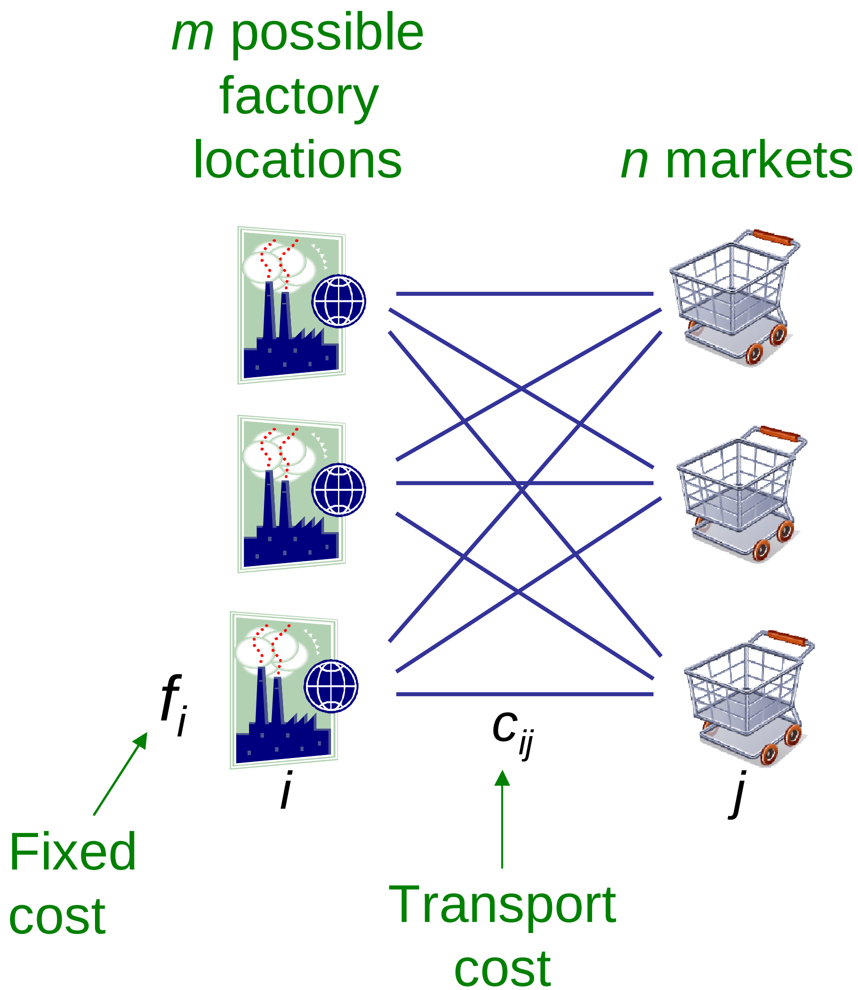
$$\begin{aligned} &\min \;\; x_2 \\ &0 \le x_1 \le My \\ &x_2 \ge cx_1 + fy \\ &y \in \{0,1\} \end{aligned}$$

or

$$\begin{aligned} &\min \;\; cx + fy \\ &0 \le x \le \boxed{M}y \\ &y \in \{0,1\} \end{aligned}$$

"Big *M*"

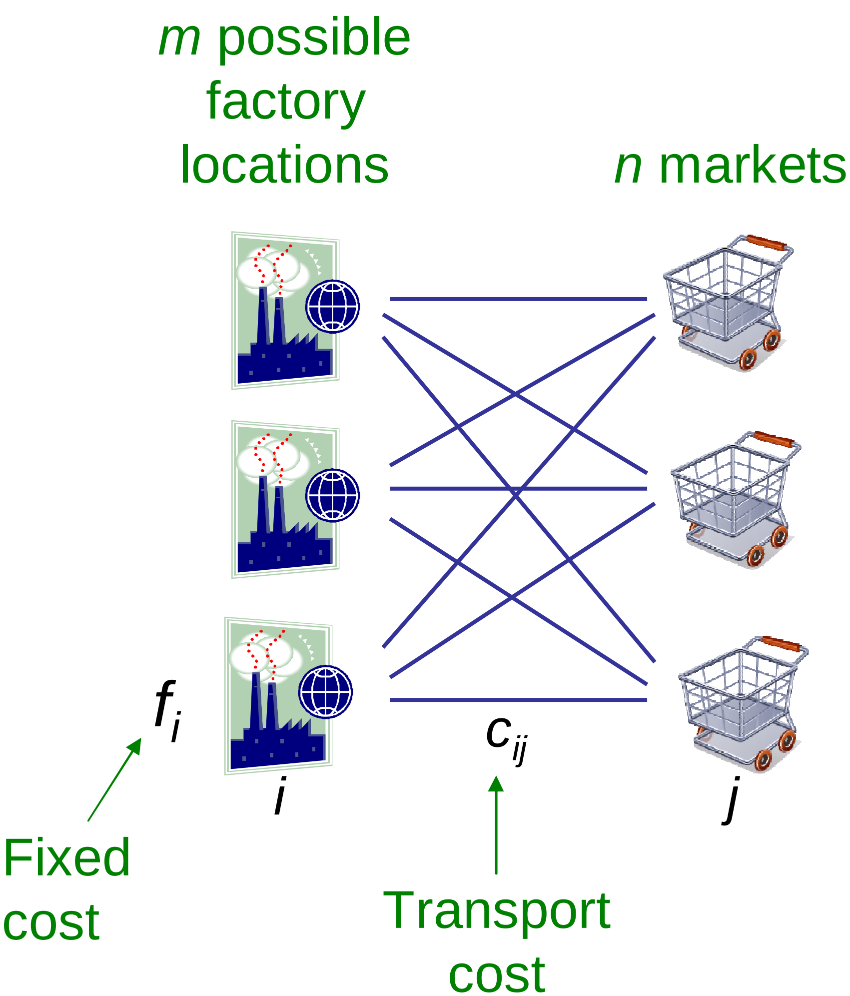
Example: Uncapacitated facility location



Locate factories to serve markets so as to minimize total fixed cost and transport cost.

No limit on production capacity of each factory.

Uncapacitated facility location



Fraction of market *j*'s demand satisfied from location *i*

Disjunctive model:

$$\min \sum_i z_i + \sum_{ij} c_{ij} \boxed{x_{ij}}$$

$$\begin{pmatrix} 0 \le x_{ij} \le 1, \text{ all } j \\ z_i \ge f_i \end{pmatrix} \vee \begin{pmatrix} x_{ij} = 0, \text{ all } j \\ z_i = 0 \end{pmatrix}, \quad \text{all } i$$

$$\sum_i x_{ij} = 1, \quad \text{all } j$$

Factory at location *i*

No factory at location *i*

Uncapacitated facility location

Disjunctive model:

$$\min \sum_i z_i + \sum_{ij} c_{ij} x_{ij}$$

$$\begin{pmatrix} 0 \le x_{ij} \le 1, \ \text{all } j \\ z_i \ge f_i \end{pmatrix} \vee \begin{pmatrix} x_{ij} = 0, \ \text{all } j \\ z_i = 0 \end{pmatrix}, \quad \text{all } i$$

$$\sum_i x_{ij} = 1, \quad \text{all } j$$

MILP formulation:

$$\min \sum_i z_i + \sum_{ij} c_{ij} x_{ij}$$

$$0 \le x_{ij}^1 \le y_i, \ \text{all } i, j \qquad x_{ij}^2 = 0, \ \text{all } i, j$$

$$z_i^1 \ge f_i y_i, \ \text{all } i \qquad z_i^2 = 0, \ \text{all } i$$

$$x_{ij} = x_{ij}^1 + x_{ij}^2, \ \ z_i = z_i^1 + z_i^2, \ \ y_i \in \{0,1\}$$

$$\sum_i x_{ij} = 1, \ \text{all } j$$

Uncapacitated facility location

$$\text{Let } x_{ij}^1 = x_{ij} \quad \text{since } x_{ij}^2 = 0$$

$$\text{Let } z_i^1 = z_i \quad \text{since } z_i^2 = 0$$

MILP formulation:

$$\min \sum_i z_i + \sum_{ij} c_{ij} x_{ij}$$

$$0 \le x_{ij}^1 \le y_i, \text{ all } i, j \qquad x_{ij}^2 = 0, \text{ all } i, j$$

$$z_i^1 \ge f_i y_i, \text{ all } i \qquad z_i^2 = 0, \text{ all } i$$

$$x_{ij} = x_{ij}^1 + x_{ij}^2, \;\; z_i = z_i^1 + z_i^2, \;\; y_i \in \{0,1\}$$

$$\sum_i x_{ij} = 1, \text{ all } j$$

Uncapacitated facility location

Let $x_{ij}^1 = x_{ij}$ since $x_{ij}^2 = 0$

Let $z_i^1 = z_i$ since $z_i^2 = 0$

MILP formulation:

$$\min \sum_i z_i + \sum_{ij} c_{ij} x_{ij}$$

$$0 \le x_{ij} \le y_i, \text{ all } i, j$$

$$z_i \ge f_i y_i, \text{ all } i$$

$$y_i \in \{0,1\}$$

$$\sum_i x_{ij} = 1, \text{ all } j$$

Uncapacitated facility location

Let $x_{ij}^1 = x_{ij}$ since $x_{ij}^2 = 0$

Let $z_i^1 = z_i$ since $z_i^2 = 0$

MILP formulation:

$$
\begin{aligned}
&\min \sum_i z_i + \sum_{ij} c_{ij} x_{ij} \\
&0 \le x_{ij} \le y_i, \text{ all } i, j \\
&z_i \ge f_i y_i, \text{ all } i \\
&y_i \in \{0,1\} \\
&\sum_i x_{ij} = 1, \text{ all } j
\end{aligned}
$$

or

$$
\begin{aligned}
&\min \sum_i f_i y_i + \sum_{ij} c_{ij} x_{ij} \\
&0 \le x_{ij} \le y_i, \text{ all } i, j \\
&y_i \in \{0,1\} \\
&\sum_i x_{ij} = 1, \text{ all } j
\end{aligned}
$$

Uncapacitated facility location

Maximum output from location i

MILP formulation:

$$\min \sum_i f_i y_i + \sum_{ij} c_{ij} x_{ij}$$

$$0 \le x_{ij} \le y_i, \text{ all } i, j$$

$$y_i \in \{0,1\}$$

$$\sum_i x_{ij} = 1, \text{ all } j$$

Beginner's model:

$$\min \sum_i f_i y_i + \sum_{ij} c_{ij} x_{ij}$$

$$\sum_j x_{ij} \le \boxed{n y_i}, \text{ all } i$$

$$y_i \in \{0,1\}$$

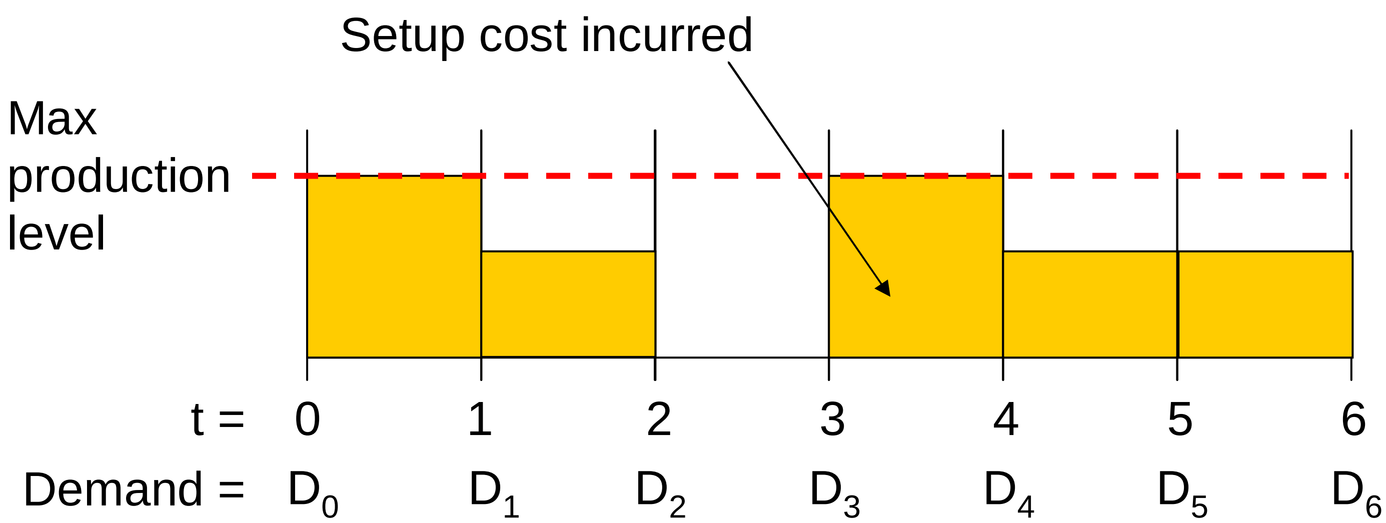
$$\sum_i x_{ij} = 1, \text{ all } j$$

Based on capacitated location model.

It has a **weaker continuous relaxation**

This beginner's mistake can be avoided by starting with disjunctive formulation.

Example: Lot sizing with setup costs



Determine lot size in each period to minimize total production, inventory, and setup costs.

Fixed-cost variable | Fixed cost | Production capacity | Production level

$$\begin{pmatrix} v_t \ge f_t \\ 0 \le x_t \le C_t \end{pmatrix} \vee \begin{pmatrix} v_t \ge 0 \\ 0 \le x_t \le C_t \end{pmatrix} \vee \begin{pmatrix} v_t \ge 0_t \\ x_t = 0 \end{pmatrix}$$

(1)	(2)	(3)
Start production (incurs setup cost)	Continue production (no setup cost)	Produce nothing (no production cost)

Logical conditions:

(2) In period t $\Rightarrow$ (1) or (2) in period $t - 1$

(1) In period t $\Rightarrow$ neither (1) nor (2) in period $t - 1$

(1)	(2)	(3)
Start production	Continue production	Produce nothing

$$\begin{pmatrix} v_t \ge f_t \\ 0 \le x_t \le C_t \end{pmatrix} \vee \begin{pmatrix} v_t \ge 0 \\ 0 \le x_t \le C_t \end{pmatrix} \vee \begin{pmatrix} v_t \ge 0_t \\ x_t = 0 \end{pmatrix}$$

Convex hull MILP model of disjunction:

$$\begin{array}{ccc} v_t^1 \ge f_t y_{t1} & v_t^2 \ge 0 & v_t^3 \ge 0 \\ 0 \le x_t^1 \le C_t y_{t1} & 0 \le x_t^2 \le C_t y_{t2} & x_t^3 = 0 \end{array}$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad x_t = \sum_{k=1}^{3} x_t^k, \quad y_t = \sum_{k=1}^{3} y_{tk}$$

$$y_{tk} \in \{0,1\}, \quad k = 1,2,3$$

To simplify, define

$$z_t = y_{t1}$$

$$y_t = y_{t2}$$

Convex hull MILP model of disjunction:

$$v_t^1 \geq f_t y_{t1} \qquad v_t^2 \geq 0 \qquad v_t^3 \geq 0$$

$$0 \leq x_t^1 \leq C_t y_{t1} \qquad 0 \leq x_t^2 \leq C_t y_{t2} \qquad x_t^3 = 0$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad x_t = \sum_{k=1}^{3} x_t^k, \quad y_t = \sum_{k=1}^{3} y_{tk}$$

$$y_{tk} \in \{0,1\}, \quad k = 1,2,3$$

To simplify, define

$z_t = y_{t1}$

$y_t = y_{t2}$

Convex hull MILP model of disjunction:

$$v_t^1 \geq f_t z_t \qquad v_t^2 \geq 0 \qquad v_t^3 \geq 0$$

$$0 \leq x_t^1 \leq C_t z_t \qquad 0 \leq x_t^2 \leq C_t y_t \qquad x_t^3 = 0$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad x_t = \sum_{k=1}^{3} x_t^k, \quad z_t + y_t \leq 1$$

$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$

= 1 for startup

= 1 for continued production

$$\text{Since } x_t^3 = 0$$
$$\text{set } x_t = x_t^1 + x_t^2$$

Convex hull MILP model of disjunction:

$$v_t^1 \geq f_t z_t \qquad v_t^2 \geq 0 \qquad v_t^3 \geq 0$$
$$0 \leq x_t^1 \leq C_t z_t \qquad 0 \leq x_t^2 \leq C_t y_t \qquad x_t^3 = 0$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad x_t = \sum_{k=1}^{3} x_t^k, \quad z_t + y_t \leq 1$$
$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$

$$\boxed{\begin{array}{l}\text{Since } x_t^3 = 0 \\ \text{set } x_1 = x_1^1 + x_2^2\end{array}}$$

Convex hull MILP model of disjunction:

$$v_t^1 \geq f_t z_t \qquad v_t^2 \geq 0 \qquad v_t^3 \geq 0$$

$$0 \leq x_t \leq C_t(z_t + y_t)$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad z_t + y_t \leq 1$$

$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$

Since v_t occurs positively in the objective function, and v_t^2, v_t^3 do not play a role, let $v_t = v_t^1$

Convex hull MILP model of disjunction:

$$v_t^1 \geq f_t z_t \qquad v_t^2 \geq 0 \qquad v_t^3 \geq 0$$

$$0 \leq x_t \leq C_t(z_t + y_t)$$

$$v_t = \sum_{k=1}^{3} v_t^k, \quad z_t + y_t \leq 1$$

$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$

Since v_t occurs positively in the objective function, and v_t^2, v_t^3 do not play a role, let $v_t = v_t^1$

Convex hull MILP model of disjunction:

$$
\begin{aligned}
& v_t \geq f_t z_t \\
& 0 \leq x_t \leq C_t (z_t + y_t) \\
& z_t + y_t \leq 1 \\
& z_t, y_t \in \{0,1\}, \quad k = 1,2,3
\end{aligned}
$$

Formulate logical conditions:

(2) In period t $\Rightarrow$ (1) or (2) in period $t - 1$

(1) In period t $\Rightarrow$ neither (1) nor (2) in period $t - 1$

$$v_t \geq f_t z_t$$
$$0 \leq x_t \leq C_t(z_t + y_t)$$
$$z_t + y_t \leq 1$$
$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$
$$y_t \leq z_{t-1} + y_{t-1}$$
$$z_t \leq 1 - z_{t-1} - y_{t-1}$$

Add objective function

Unit production cost

Unit holding cost

$$\min \quad \sum_{t=1}^{n} (p_t x_t + h_t s_t + v_t)$$

$$v_t \geq f_t z_t$$

$$0 \leq x_t \leq C_t (z_t + y_t)$$

$$z_t + y_t \leq 1$$

$$z_t, y_t \in \{0,1\}, \quad k = 1,2,3$$

$$y_t \leq z_{t-1} + y_{t-1}$$

$$z_t \leq 1 - z_{t-1} - y_{t-1}$$

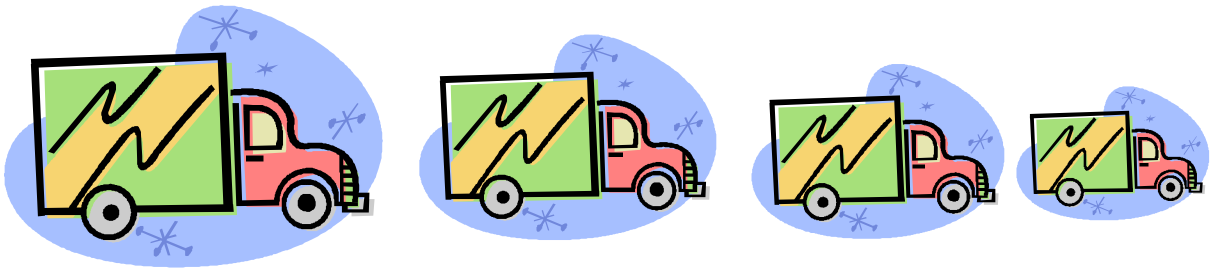
Knapsack Models

Integer variables can also be used to express counting ideas.

This is totally different from the use of 0-1 variables to express unions of polyhedra.

Example: Freight Transfer

- Transport 42 tons of freight using 8 trucks, which come in 4 sizes…



Truck size	Number available	Capacity (tons)	Cost per truck
1	3	7	90
2	3	5	60
3	3	4	50
4	3	3	40

Number of trucks of type 1

$$\min\ 90x_1 + 60x_2 + 50x_3 + 40x_4$$

$$7x_1 + 5x_2 + 4x_3 + 3x_4 \geq 42$$

$$x_1 + x_2 + x_3 + x_4 \leq 8$$

$$x_i \in \{0,1,2,3\}$$

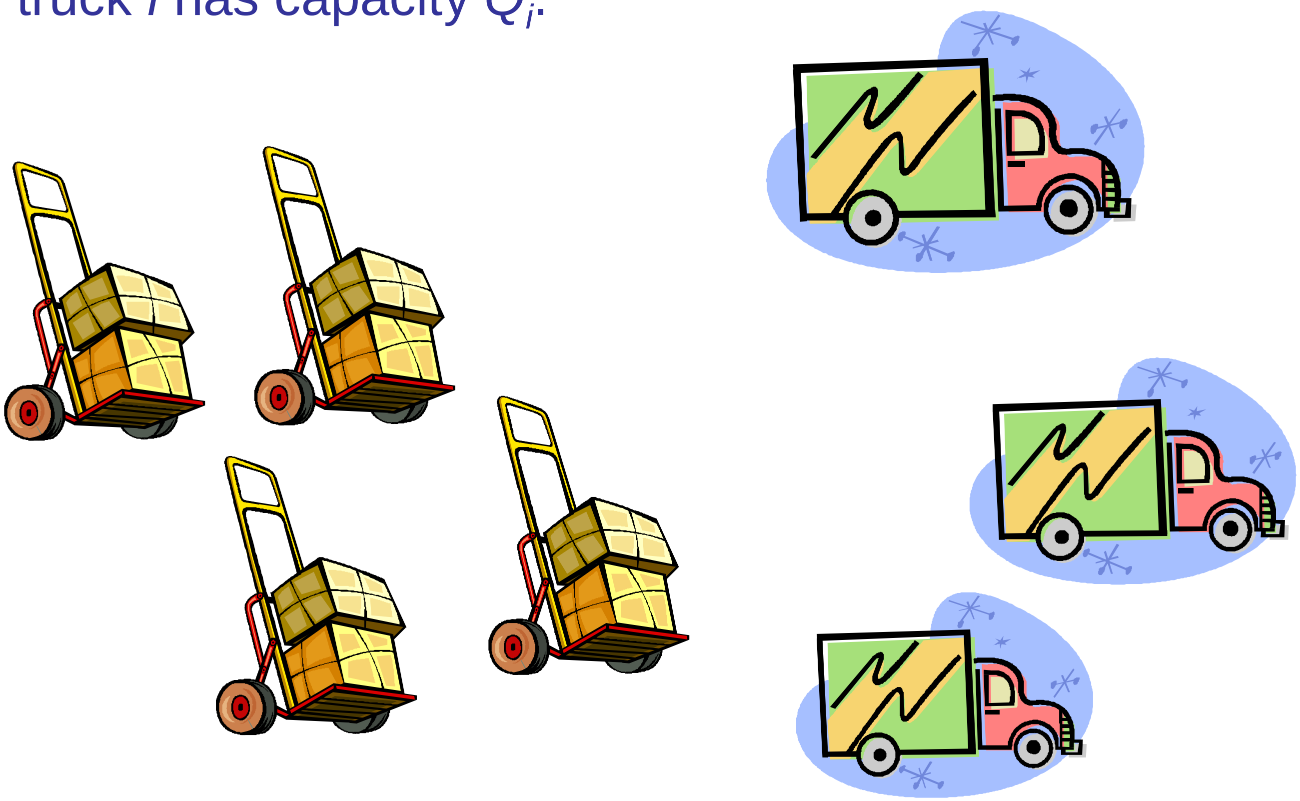
Knapsack covering constraint

Knapsack packing constraint

Truck type	Number available	Capacity (tons)	Cost per truck
1	3	7	90
2	3	5	60
3	3	4	50
4	3	3	40

Example: Freight Packing and Transfer

- Transport packages using *n* trucks
- Each package *j* has size a_j.
- Each truck *i* has capacity Q_i.



Knapsack component

The trucks selected must have enough capacity to carry the load.

$$\sum_{i=1}^{n} Q_i \boxed{y_i} \geq \sum_j a_j$$

= 1 if truck *i* is selected

Disjunctive component (with embedded knapsack constraint)

Truck i selected

Truck i not selected

Cost variable

Cost of operating truck i

$$\begin{pmatrix} \boxed{z_i} \ge \boxed{c_i} \\ \sum_j a_j x_{ij} \le Q_i \\ \boxed{0 \le x_{ij} \le 1, \text{ all } j} \end{pmatrix} \vee \begin{pmatrix} z_i \ge 0 \\ \boxed{x_{ij}} = 0 \end{pmatrix}$$

Use continuous relaxation because we want a disjunction of linear systems

= 1 if package j is loaded on truck i

Disjunctive component (with embedded knapsack constraint)

Truck i selected	Truck i not selected

$$\begin{pmatrix} z_i \ge c_i \\ \sum_j a_j x_{ij} \le Q_i \\ 0 \le x_{ij} \le 1, \text{ all } j \end{pmatrix} \vee \begin{pmatrix} z_i \ge 0 \\ x_{ij} = 0 \end{pmatrix}$$

Convex hull MILP formulation

$$z_i \ge c_i y_i$$

$$\sum_j a_j x_{ij} \le Q_i y_i$$

$$0 \le x_{ij} \le y_i$$

The resulting model

$$\min \sum_{i=1}^{n} c_i y_i$$

$$\boxed{\begin{array}{l} \sum_j a_j x_{ij} \le Q_i y_i, \quad \text{all } i \\ 0 \le x_{ij} \le y_i, \quad \text{all } i, j \end{array}}$$

Disjunctive component

$$\boxed{\sum_{i=1}^{n} x_{ij} = 1, \quad \text{all } j}$$

Logical condition
(each package must be shipped)

$$\boxed{\sum_{i=1}^{n} Q_i y_i \ge \sum_j a_j}$$

$$x_{ij}, y_i \in \{0,1\}$$

Knapsack component

The resulting model

$$\min \sum_{i=1}^{n} c_i y_i$$

$$\sum_j a_j x_{ij} \le Q_i \boxed{y_i}, \quad \text{all } i$$

$$0 \le x_{ij} \le y_i, \quad \text{all } i, j$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \text{all } j$$

$$\sum_{i=1}^{n} Q_i y_i \ge \sum_j a_j$$

$$x_{ij}, y_i \in \{0,1\}$$

The y_i is redundant but makes the continuous relaxation tighter.

This is a modeling “trick,” part of the folklore of modeling.

The resulting model

$$\min \sum_{i=1}^{n} c_i y_i$$

$$\sum_j a_j x_{ij} \le Q_i \boxed{y_i}, \quad \text{all } i$$

$$0 \le x_{ij} \le y_i, \quad \text{all } i, j$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \text{all } j$$

$$\boxed{\sum_{i=1}^{n} Q_i y_i \ge \sum_j a_j}$$

$$x_{ij}, y_i \in \{0,1\}$$

The y_i is redundant but makes the continuous relaxation tighter.

This is a modeling "trick," part of the folklore of modeling.

Conventional modeling wisdom would not use this constraint, because it is the sum of the first constraint over i.

But it radically reduces solution time, because it generates knapsack cuts.

This argues for a principled approach to modeling.

Constraint Programming Models

CP Modeling Systems

Global Constraints

Employee Scheduling

CP Modeling Systems

- Commercial modeling systems with dedicated solvers
  - OPL Studio (runs ILOG Solver, ILOG Scheduler)
  - CHIP (runs CHIP solver)
  - Mosel (runs Xpress-Kalis)
  - Mozart (uses Oz language)

- Non-commercial modeling system with dedicated solvers
  - ECLiPSe (runs ECLiPSe CP solver)

Global constraints

- A **global constraint** represents a set of constraints with special structure.
- The structure is exploited by **filtering** algorithms in the CP solver.

Some general-purpose global constraints

Alldiff - Requires that all the listed variables take different values.

Among - Bounds the number of listed variables that take one of the values in a list.

Cardinality - Bounds the number of listed variables that take each of the values in a list.

Element - Requires that a given variable take the *y*th value in a list, where *y* is an integer variable.

Path - Requires that a given graph contain a path of at most a given length.

Some global constraints for scheduling

Disjunctive - Requires that no two jobs overlap in time.

Cumulative - Limits the resources consumed by jobs running at any one time. In particular, it can limit the number of jobs running at any one time.

Stretch - Bounds the length of a stretch of contiguous periods assigned the same job.

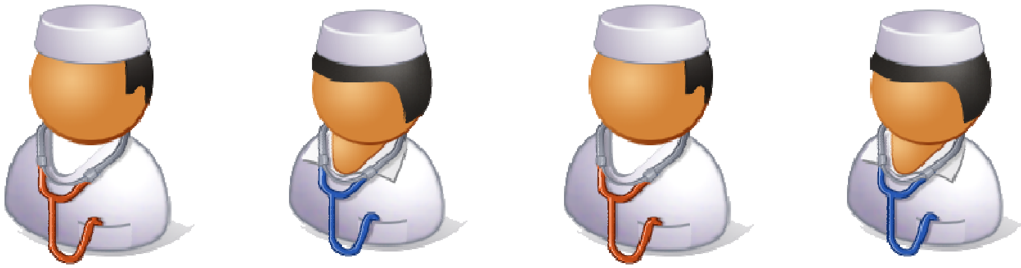
Sequence – A set of overlapping **among** constraints.

Regular – Generalizes **stretch** and **sequence**.

Diffn - Requires that no two boxes in a set of multidimensional boxes overlap. Used for space or space-time packing.

Example: Employee Scheduling

- Schedule four nurses in 8-hour shifts.
- A nurse works at most one shift a day, at least 5 days a week.
- Same schedule every week.
- No shift staffed by more than two different nurses in a week.
- A nurse cannot work different shifts on two consecutive days.
- A nurse who works shift 2 or 3 must do so at least two days in a row.



Two ways to view the problem

Assign nurses to shifts

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Shift 1	A	B	A	A	A	A	A
Shift 2	C	C	C	B	B	B	B
Shift 3	D	D	D	D	C	C	D

Assign shifts to nurses

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Nurse A	1	0	1	1	1	1	1
Nurse B	0	1	0	2	2	2	2
Nurse C	2	2	2	0	3	3	0
Nurse D	3	3	3	3	0	0	3

0 = day off

Use **both** formulations in the same model!

First, assign nurses to shifts.

Let w_{sd} = nurse assigned to shift s on day d

$$\text{alldiff}(w_{1d}, w_{2d}, w_{3d}), \quad \text{all } d$$

The variables w_{1d}, w_{2d}, w_{3d} take different values

That is, schedule 3 different nurses on each day

Use **both** formulations in the same model!

First, assign nurses to shifts.

Let w_{sd} = nurse assigned to shift s on day d

$$\text{alldiff}(w_{1d}, w_{2d}, w_{3d}), \quad \text{all } d$$
$$\text{cardinality}\big(w \mid (A,B,C,D),(5,5,5,5),(6,6,6,6)\big)$$

A occurs at least 5 and at most 6 times in the array w, and similarly for B, C, D.

That is, each nurse works at least 5 and at most 6 days a week

Use **both** formulations in the same model!

First, assign nurses to shifts.

Let w_{sd} = nurse assigned to shift s on day d

$$\text{alldiff}\left(w_{1d}, w_{2d}, w_{3d}\right), \quad \text{all } d$$

$$\text{cardinality}\left(w \mid (A, B, C, D), (5,5,5,5), (6,6,6,6)\right)$$

$$\text{nvalues}\left(w_{s,\text{Sun}}, \ldots, w_{s,\text{Sat}} \mid 1, 2\right), \quad \text{all } s$$

The variables $w_{s,\text{Sun}}$, …, $w_{s,\text{Sat}}$ take at least 1 and at most 2 different values.

That is, at least 1 and at most 2 nurses work any given shift.

Remaining constraints are not easily expressed in this notation.

So, assign shifts to nurses.

Let y_{id} = shift assigned to nurse i on day d

$$\text{alldiff}\left(y_{1d}, y_{2d}, y_{3d}\right),\ \ \text{all } d$$

Assign a different nurse to each shift on each day.

This constraint is redundant of previous constraints, but redundant constraints speed solution.

Remaining constraints are not easily expressed in this notation.

So, assign shifts to nurses.

Let y_{id} = shift assigned to nurse i on day d

$$\text{alldiff}\left(y_{1d}, y_{2d}, y_{3d}\right), \text{ all } d$$
$$\text{stretch}\left(y_{i,\text{Sun}}, \ldots, y_{i,\text{Sat}} \mid (2,3),(2,2),(6,6),P\right), \text{ all } i$$

Every stretch of 2's has length between 2 and 6.
Every stretch of 3's has length between 2 and 6.

So a nurse who works shift 2 or 3 must do so at least two days in a row.

Remaining constraints are not easily expressed in this notation.

So, assign shifts to nurses.

Let y_{id} = shift assigned to nurse i on day d

$$\text{alldiff}\left(y_{1d}, y_{2d}, y_{3d}\right), \ \text{all } d$$
$$\text{stretch}\left(y_{i,\text{Sun}}, \ldots, y_{i,\text{Sat}} \mid (2,3),(2,2),(6,6),P\right), \ \text{all } i$$

Here $P = \{(s,0),(0,s) \mid s = 1,2,3\}$

Whenever a stretch of a's immediately precedes a stretch of b's, (a,b) must be one of the pairs in P.

So a nurse cannot switch shifts without taking at least one day off.

Now we must connect the w_{sd} variables to the y_{id} variables.

Use **channeling constraints**:

$$w_{y_{id}d} = i, \quad \text{all } i,d$$

$$y_{w_{sd}d} = s, \quad \text{all } s,d$$

Channeling constraints increase propagation and make the problem easier to solve.

The complete model is:

$$\text{alldiff}\left(w_{1d}, w_{2d}, w_{3d}\right), \ \text{ all } d$$
$$\text{cardinality}\left(w \mid (A,B,C,D),(5,5,5,5),(6,6,6,6)\right)$$
$$\text{nvalues}\left(w_{s,\text{Sun}},\ldots,w_{s,\text{Sat}} \mid 1,2\right), \ \text{ all } s$$

$$\text{alldiff}\left(y_{1d}, y_{2d}, y_{3d}\right), \ \text{ all } d$$
$$\text{stretch}\left(y_{i,\text{Sun}},\ldots,y_{i,\text{Sat}} \mid (2,3),(2,2),(6,6),P\right), \ \text{ all } i$$

$$w_{y_{id}d} = i, \ \text{ all } i,d$$
$$y_{w_{sd}d} = s, \ \text{ all } s,d$$

Integrated Models

Modeling Systems

Product Configuration

Machine Assignment and Scheduling

Integrated Modeling Systems

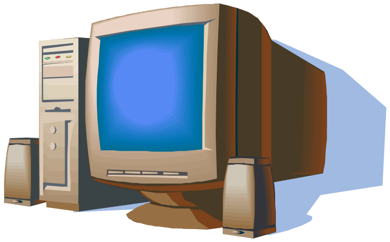
- Commercial modeling systems with dedicated solvers
  - OPL Studio (runs CPLEX, ILOG Solver/Scheduler)
  - Mosel (runs Xpress-MP, Xpress-Kalis)

- Non-commercial modeling systems with dedicated solvers
  - ECLiPSe (runs ECLiPSe CP solver, Xpress-MP)
  - SIMPL (just released, open source)

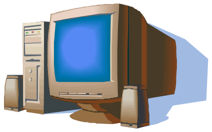
Example: Product Configuration

This example combines **MILP modeling** with **variable indices**, used in constraint programming.

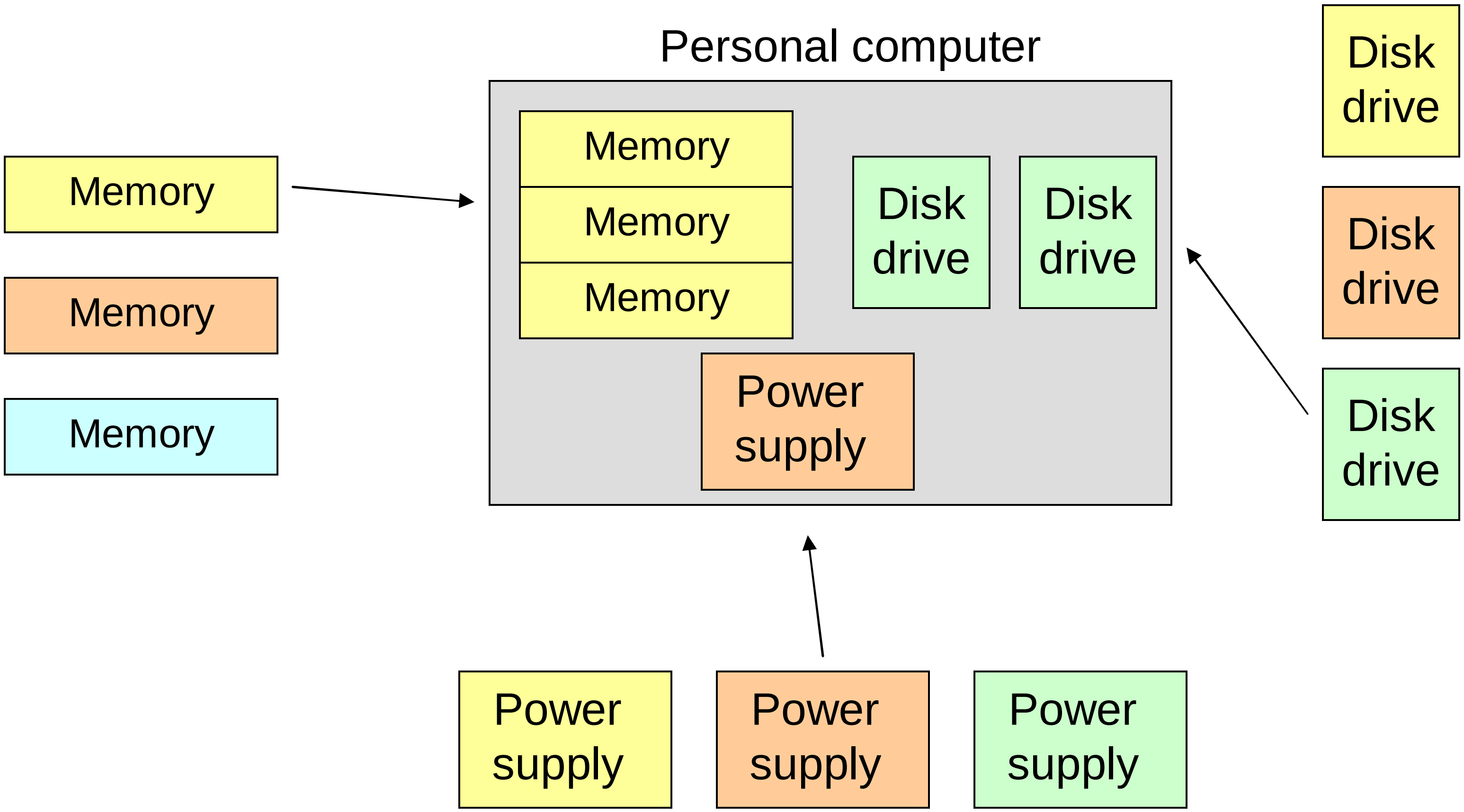
• It can be solved by combining MILP and CP techniques.



The problem



Choose what type of each component, and how many



Integrated model

Unit cost of producing attribute j

Amount of attribute j produced by type t_i of component i

$$\min \sum_j c_j v_j$$

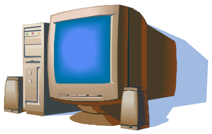
Amount of attribute j produced
(< 0 if consumed):
memory, heat, power, weight, etc.

$$v_j = \sum_{ik} q_i A_{ijt_i}, \text{ all } j$$

$$L_j \le v_j \le U_j, \text{ all } j$$

Quantity of component i installed

Integrated model



$$\min \sum_j c_j v_j$$

$$\boxed{v_j = \sum_{ik} q_i A_{ijt_i}, \text{ all } j}$$

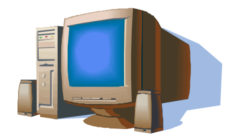
$$L_j \le v_j \le U_j, \text{ all } j$$

This is reformulated

$$\boxed{\begin{array}{l} v_j = \sum_i z_i, \text{ all } j \\ \text{element}\left(t_i, (q_i A_{ij1}, \ldots, q_i A_{ijn}), z_i\right), \text{ all } i, j \end{array}}$$

t_i is a **variable index**

Integrated model



$$\min \sum_j c_j v_j$$

$$\boxed{v_j = \sum_{ik} q_i A_{ijt_i}, \text{ all } j}$$

$$L_j \le v_j \le U_j, \text{ all } j$$

This
is reformulated

t_i is a **variable index**

$$\boxed{\begin{array}{l} v_j = \sum_i z_i, \text{ all } j \\ \text{element}\left(t_i, (q_i A_{ij1}, \ldots, q_i A_{ijn}), z_i\right), \text{ all } i, j \end{array}}$$

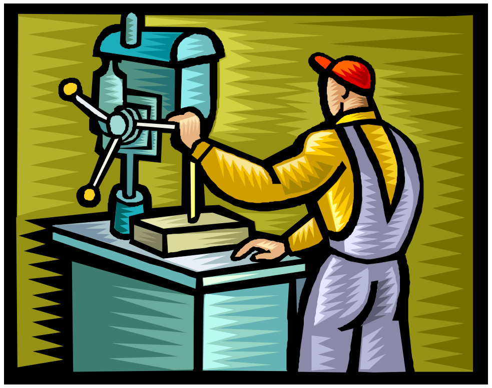
Set z_i equal to the t_i^{th} item in the red list.

Machine Assignment and Scheduling

• Assign jobs to machines and schedule the machines assigned to each machine within time windows.

• The objective is to minimize **makespan**.

Time lapse between start of first job and end of last job.



• Combine MILP and CP modeling

Machine Scheduling

The model is

Processing time of job j
on machine $\boxed{x_j}$

Machine
assigned to job j

$$\min \; M$$

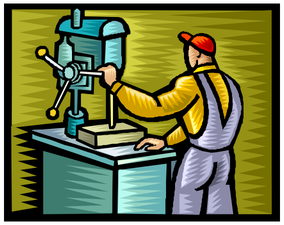
$$M \ge \boxed{s_j} + \boxed{p_{x_j j}}, \text{ all } j$$

Makespan

Start time
variable for
job j

$$r_j \le s_j \le d_j - p_{x_j j}, \text{ all } j$$

$$\text{disjunctive}\big((s_j \mid x_j = i), (p_{ij} \mid x_j = i)\big), \text{ all } i$$



Machine Scheduling

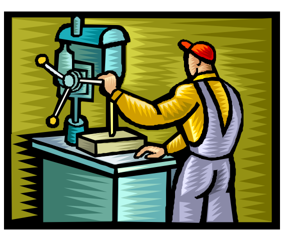
The model is

Release time
for job *j*

$$\begin{aligned}
&\min \; M \\
&M \geq s_j + p_{x_j j}, \text{ all } j \\
&\boxed{r_j} \leq s_j \leq \boxed{d_j} - p_{x_j j}, \text{ all } j \\
&\text{disjunctive}\big((s_j \,|\, x_j = i), (p_{ij} \,|\, x_j = i) \big), \text{ all } i
\end{aligned}$$

Time windows

Deadline
for job *j*



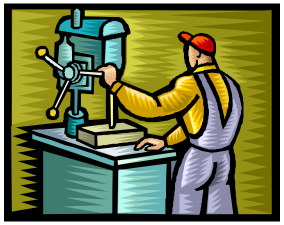
Machine Scheduling

The model is

$$
\begin{aligned}
&\min \ M \\
&M \geq \boxed{s_j} + p_{x_j j}, \ \text{all } j \\
&r_j \leq s_j \leq d_j - p_{x_j j}, \ \text{all } j \\
&\text{disjunctive}\Big(\boxed{(s_j \mid x_j = i)}, (p_{ij} \mid x_j = i)\Big), \ \text{all } i
\end{aligned}
$$

Start times of jobs assigned to machine i

Disjunctive global constraint requires that Jobs do not overlap



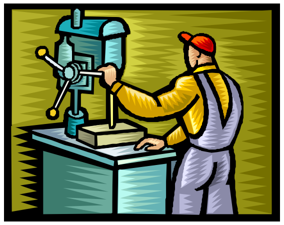
Machine Scheduling

The problem can be solved by logic-based Benders decomposition.

$$\begin{aligned}&\min\ M\\ &M \ge \boxed{s_j} + p_{x_j j},\ \text{all } j\\ &r_j \le s_j \le d_j - p_{x_j j},\ \text{all } j\end{aligned}$$

$$\text{disjunctive}\big((s_j \mid x_j = i),(p_{ij} \mid x_j = i)\big),\ \text{all } i$$

Master problem is this plus Benders cuts, solved as an MILP



Machine Scheduling

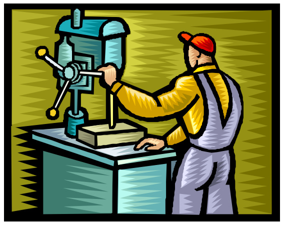
The problem can be solved by logic-based Benders decomposition.

$$\min\ M$$
$$M \ge \boxed{s_j} + p_{x_j j},\ \text{all } j$$
$$r_j \le s_j \le d_j - p_{x_j j},\ \text{all } j$$

Master problem is this plus Benders cuts, solved as an MILP

$$\text{disjunctive}\big((s_j \mid x_j = i),(p_{ij} \mid x_j = i)\big),\ \text{all } i$$

Subproblem is this, solved by CP



Proposal

- Replace atomistic modeling with modeling based on global constraints.
  - Including specially-structured families of inequalities.
- Build solvers with constraint-based control.
  - Each global constraint invokes specialized filters, relaxations, cutting planes.