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:

\[
\begin{align*}
\text{min} & \quad cx + dy \\
Ax + by & \geq b \\
x, y & \geq 0 \\
y & \text{integer}
\end{align*}
\]
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
  \[
  \text{disjunctions of linear} \quad + \quad \text{integer knapsack systems} \quad + \quad \text{inequalities}
  \]
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.

Recession cone of polyhedron

Polyhedron

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 $k$th polyhedron is \( \{ x \mid A^k x \geq b \} \)

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 $k$.

\[
\begin{align*}
\min & \quad cx \\
\forall k & \quad (A^k x \geq b^k) \\
\min & \quad cx \\
A^k x^k & \geq b^k y_k, \text{ all } k \\
\sum_k y_k & = 1 \\
x & = \sum_k x^k \\
y_k & \in \{0,1\}
\end{align*}
\]
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...

\[
\begin{align*}
\min & \quad cx \\
A^k x^k & \geq b^k, \quad \text{all } k \\
\sum_k y_k & = 1 \\
x & = \sum_k y_k x^k \\
y_k & \in \{0,1\}
\end{align*}
\]

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

To derive convex hull relaxation of a disjunction...

\[
\begin{align*}
\text{min } & \quad cx \\
A^k x^k & \geq b^k, \text{ all } k \\
\sum_k y_k & = 1 \\
x & = \sum_k y_k x^k \\
y_k & \in \{0,1\}
\end{align*}
\]

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

\[
\begin{align*}
\text{min } & \quad cx \\
A^k x^k & \geq b^k y_k, \text{ all } k \\
\sum_k y_k & = 1 \\
x & = \sum_k x^k \\
y_k & \in \{0,1\}
\end{align*}
\]

Change of variable
\[
x = y_k x^k
\]

Convex hull relaxation (tightest linear relaxation)
Example: Fixed charge function

Minimize a fixed charge function:

\[
\min \quad x_2 \\
\begin{cases}
0 & \text{if } x_1 = 0 \\
f + cx_1 & \text{if } x_1 > 0
\end{cases}
\]

\[x_1 \geq 0\]
Fixed charge problem

Minimize a fixed charge function:

\[
\begin{align*}
\min \quad & x_2 \\
\quad \text{subject to} \quad & x_2 \geq \begin{cases} 
0 & \text{if } x_1 = 0 \\
 f + cx_1 & \text{if } x_1 > 0 
\end{cases} \\
\quad & x_1 \geq 0
\end{align*}
\]

Feasible set
Fixed charge problem

Minimize a fixed charge function:

\[
\begin{align*}
\min & \quad x_2 \\
\text{s.t.} & \quad x_2 \begin{cases} 
0 & \text{if } x_1 = 0 \\
 f + cx_1 & \text{if } x_1 > 0
\end{cases} \\
x_1 & \geq 0
\end{align*}
\]
Fixed charge problem

Minimize a fixed charge function:

\[
\begin{align*}
\min & \quad x_2 \\
\text{s.t.} & \quad x_2 \geq \begin{cases} 
0 & \text{if } x_1 = 0 \\
 f + cx_1 & \text{if } x_1 > 0 
\end{cases} \\
x_1 & \geq 0
\end{align*}
\]

Union of two polyhedra \( P_1, P_2 \)
Fixed charge problem

Minimize a fixed charge function:

\[
\begin{align*}
\min \quad & x_2 \\
\begin{cases}
0 & \text{if } x_1 = 0 \\
f + cx_1 & \text{if } x_1 > 0
\end{cases}
\end{align*}
\]

\[x_1 \geq 0\]

The polyhedra have different recession cones.
Fixed charge problem

Minimize a fixed charge function:

\[
\min x_2 \\
x_2 \begin{cases} 
0 & \text{if } x_1 = 0 \\
f + cx_1 & \text{if } x_1 > 0
\end{cases}
\]

\[0 \leq x_1 \leq M\]

The polyhedra have the same recession cone.
Fixed charge problem

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

\[
\begin{align*}
\text{min } & \quad x_2 \\
\text{subject to } & \quad \begin{cases} 
 x_1 = 0 \\
 x_2 \geq 0 \\
 x_2 \geq f + cx_1 
\end{cases} \\
& \quad 0 \leq x_1 \leq M
\end{align*}
\]
Fixed charge problem

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

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 $k$.

\[
\begin{align*}
\min & \quad x_2 \\
\text{s.t.} & \quad (x_1 = 0) \lor (0 \leq x_1 \leq M) \\
& \quad x_2 \geq 0 \lor (x_2 \geq f + cx_1) \\
\end{align*}
\]

\[
\begin{align*}
\min & \quad x_2 \\
\text{s.t.} & \quad x_1^1 = 0, \quad 0 \leq x_1^2 \leq My_2 \\
& \quad x_2^1 \geq 0, \quad -cx_1^2 + x_2^2 \geq fy_2 \\
& \quad y_1 + y_2 = 1, \quad y_k \in \{0, 1\} \\
& \quad x_1 = x_1^1 + x_1^2, \quad x_2 = x_2^1 + x_2^2
\end{align*}
\]
To simplify, replace $x_1^2$ with $x_1$ since $x_1 = 0$

\[
\begin{align*}
\text{min } & \quad x_2 \\
& x_1^1 = 0 \quad 0 \leq x_1^2 \leq My_2 \\
& x_2^1 \geq 0 \quad -cx_1^2 + x_2^2 \geq fy_2 \\
& y_1 + y_2 = 1, \quad y_k \in \{0,1\} \\
& x_1 = x_1^1 + x_1^2, \quad x_2 = x_2^1 + x_2^2
\end{align*}
\]
To simplify, replace $x_1^2$ with $x_1$ since $x_1 = 0$

\[
\begin{align*}
\min & \quad x_2 \\
 \text{subject to} & \quad 0 \leq x_1 \leq My_2 \\
 & \quad x_2^1 \geq 0, \quad -cx_1 + x_2^2 \geq fy_2 \\
 & \quad y_1 + y_2 = 1, \quad y_k \in \{0, 1\} \\
 & \quad x_2 = x_2^1 + x_2^2
\end{align*}
\]
Replace $x_2^2$ with $x_2$

because $x_2^1$ plays no role in the model

\[
\begin{align*}
\text{min} & \quad x_2 \\
& 0 \leq x_1 \leq M y_2 \\
x_2^1 \geq 0 & \quad -c x_1 + x_2^2 \geq f y_2 \\
y_1 + y_2 &= 1, \quad y_k \in \{0, 1\} \\
x_2 &= x_2^1 + x_2^2
\end{align*}
\]
Replace $x_2^2$ with $x_2$

Because $x_2^1$ plays no role in the model

\[
\begin{align*}
\text{min} & \quad x_2 \\
0 & \leq x_1 \leq My_2 \\
-cx_1 + x_2 & \geq fy_2 \\
y_1 + y_2 & = 1, \quad y_k \in \{0,1\}
\end{align*}
\]
Replace $y_2$ with $y$

Because $y_1$ plays no role in the model

$$\begin{align*}
\min \ x_2 \\
0 & \leq x_1 \leq My_2 \\
-cx_1 + x_2 & \geq fy_2 \\
y_1 + y_2 &= 1, \ y_k \in \{0, 1\}
\end{align*}$$
Replace $y_2$ with $y$

Because $y_1$ plays no role in the model

$$\begin{align*}
\min & \quad x_2 \\
0 & \leq x_1 \leq My \\
x_2 & \geq cx_1 + fy \\
y & \in \{0,1\}
\end{align*}$$

or

$$\begin{align*}
\min & \quad cx + fy \\
0 & \leq x \leq My \\
y & \in \{0,1\}
\end{align*}$$

“Big $M$”
Example: Uncapacitated facility location

$m$ possible factory locations

$n$ markets

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

Disjunctive model:

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

$$\left(0 \leq x_{ij} \leq 1, \text{ all } j\right) \lor \left(z_i \geq f_i \land \sum_i x_{ij} = 1, \text{ all } j\right)$$

$$\sum_i x_{ij} = 0, \text{ all } j$$

$$z_i = 0, \text{ all } i$$
Uncapacitated facility location

Disjunctive model:

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

$$\left(0 \leq x_{ij} \leq 1, \text{ all } j\right) \lor \left(x_{ij} = 0, \text{ all } j\right)$$

$$z_i \geq f_i$$

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

MILP formulation:

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

$$0 \leq x_{ij}^1 \leq y_i, \text{ all } i, j$$

$$x_{ij}^2 = 0, \text{ all } i, j$$

$$z_i^1 \geq f_i y_i, \text{ all } i$$

$$z_i^2 = 0, \text{ all } i$$

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

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

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

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

MILP formulation:

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

$$0 \leq x^1_{ij} \leq y_i, \text{ all } i, j \quad x^2_{ij} = 0, \text{ all } i, j$$

$$z^1_i \geq f_i y_i, \text{ all } i \quad z^2_i = 0, \text{ all } i$$

$$x_{ij} = x^1_{ij} + x^2_{ij}, \quad z_i = z^1_i + z^2_i, \quad y_i \in \{0,1\}$$

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

\[
\begin{align*}
\min & \quad \sum_i z_i + \sum_{ij} c_{ij} x_{ij} \\
0 \leq & \quad x_{ij} \leq y_i, \text{ all } i, j \\
& \quad z_i \geq f_i y_i, \text{ all } i \\
y_i \in & \quad \{0,1\} \\
& \quad \sum_i x_{ij} = 1, \text{ all } j
\end{align*}
\]
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:

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

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

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

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

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

or

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

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

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

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

MILP formulation:
\[
\min \sum_i f_i y_i + \sum_{ij} c_{ij} x_{ij} \\
0 \leq x_{ij} \leq y_i, \text{ all } i, j \\
y_i \in \{0,1\} \\
\sum_i x_{ij} = 1, \text{ all } j
\]

Beginner’s model:
\[
\min \sum_i f_i y_i + \sum_{ij} c_{ij} x_{ij} \\
\sum_j x_{ij} \leq \boxed{ny_i}, \text{ all } i \\
y_i \in \{0,1\} \\
\sum_i x_{ij} = 1, \text{ all } j
\]

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

\begin{align*}
& v_t \geq f_t \\
& 0 \leq x_t \leq C_t
\end{align*}

Fixed cost

Production capacity

Production level

\begin{align*}
& v_t \geq 0 \\
& 0 \leq x_t \leq C_t
\end{align*}

\begin{align*}
& v_t \geq 0_t \\
& x_t = 0
\end{align*}

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 $t - 1$
Convex hull MILP model of disjunction:

\[
\begin{align*}
(1) & \quad \text{Start production} \\
(2) & \quad \text{Continue production} \\
(3) & \quad \text{Produce nothing}
\end{align*}
\]

\[
\begin{align*}
& \begin{cases}
  v_t \geq f_t \\
  0 \leq x_t \leq C_t
\end{cases} \lor 
\begin{cases}
  v_t \geq 0 \\
  0 \leq x_t \leq C_t
\end{cases} \lor 
\begin{cases}
  v_t \geq 0_t \\
  x_t = 0
\end{cases}
\end{align*}
\]

Convex hull MILP model of disjunction:

\[
\begin{align*}
& \begin{cases}
  v^1_t \geq f_t y_{t1} \\
  0 \leq x^1_t \leq C_t y_{t1}
\end{cases} \lor 
\begin{cases}
  v^2_t \geq 0 \\
  0 \leq x^2_t \leq C_t y_{t2}
\end{cases} \lor 
\begin{cases}
  v^3_t \geq 0 \\
  x^3_t = 0
\end{cases}
\end{align*}
\]

\[
\begin{align*}
& v_t = \sum_{k=1}^{3} v^k_t, \\
& x_t = \sum_{k=1}^{3} x^k_t, \\
& y_t = \sum_{k=1}^{3} y_{tk}
\end{align*}
\]

\[
y_{tk} \in \{0,1\}, \quad k = 1,2,3
\]
To simplify, define
\[ z_t = y_{t1} \]
\[ y_t = y_{t2} \]

Convex hull MILP model of disjunction:

\[ v_t^1 \geq f_t y_{t1} \quad v_t^2 \geq 0 \quad v_t^3 \geq 0 \]
\[ 0 \leq x_t^1 \leq C_t y_{t1} \quad 0 \leq x_t^2 \leq C_t y_{t2} \quad x_t^3 = 0 \]

\[ v_t = \sum_{k=1}^{3} v_t^k \quad x_t = \sum_{k=1}^{3} x_t^k \quad y_t = \sum_{k=1}^{3} y_{tk} \]
\[ y_{tk} \in \{0,1\}, \quad k = 1,2,3 \]
To simplify, define

\[ z_t = y_{t1} \]
\[ y_t = y_{t2} \]

Convex hull MILP model of disjunction:

\[ v^1_t \geq f_t z_t \]
\[ v^2_t \geq 0 \]
\[ v^3_t \geq 0 \]
\[ 0 \leq x^1_t \leq C_t z_t \]
\[ 0 \leq x^2_t \leq C_t y_t \]
\[ x^3_t = 0 \]

\[ v_t = \sum_{k=1}^{3} v^k_t \]
\[ x_t = \sum_{k=1}^{3} x^k_t \]
\[ z_t + y_t \leq 1 \]

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

= 1 for startup
= 1 for continued production
Convex hull MILP model of disjunction:

\[ v_t^1 \geq f_t z_t \quad v_t^2 \geq 0 \quad v_t^3 \geq 0 \]

\[ 0 \leq x_t^1 \leq C_t z_t \quad 0 \leq x_t^2 \leq C_t y_t \quad x_t^3 = 0 \]

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

\[ z_t, y_t \in \{0,1\}, \quad k = 1,2,3 \]
Convex hull MILP model of disjunction:

\[
\begin{align*}
    v_t^1 &\geq f_t z_t & v_t^2 &\geq 0 & v_t^3 &\geq 0 \\
    0 &\leq x_t & C_t(z_t + y_t) &\leq C_t(z_t + y_t) \\
    v_t &= \sum_{k=1}^{3} v_t^k & z_t + y_t &\leq 1 \\
    z_t, y_t &\in \{0,1\}, & k = 1,2,3
\end{align*}
\]
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:

$$v_t^1 \geq f_t z_t$$
$$v_t^2 \geq 0$$
$$v_t^3 \geq 0$$
$$0 \leq x_t \leq C_t(z_t + y_t)$$

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

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

\[
\begin{align*}
    v_t &\geq f_t z_t \\
    0 &\leq x_t \leq C_t (z_t + y_t) \\
    z_t + y_t &\leq 1 \\
    z_t, y_t &\in \{0, 1\}, \quad k = 1, 2, 3
\end{align*}
\]
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 $t - 1$

\[
\begin{align*}
    v_t &\geq f_t z_t \\
    0 &\leq x_t \leq C_t (z_t + y_t) \\
    z_t + y_t &\leq 1 \\
    z_t, y_t &\in \{0,1\}, \quad k = 1,2,3 \\
    y_t &\leq z_{t-1} + y_{t-1} \\
    z_t &\leq 1 - z_{t-1} - y_{t-1}
\end{align*}
\]
Add objective function

Unit production cost

Unit holding cost

\[
\min \sum_{t=1}^{n} \left( p_t x_t + h_t s_t + v_t \right)
\]

\[v_t \geq f_t z_t\]

\[0 \leq x_t \leq C_t (z_t + y_t)\]

\[z_t + y_t \leq 1\]

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

\[y_t \leq z_{t-1} + y_{t-1}\]

\[z_t \leq 1 - z_{t-1} - y_{t-1}\]
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...

<table>
<thead>
<tr>
<th>Truck size</th>
<th>Number available</th>
<th>Capacity (tons)</th>
<th>Cost per truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>40</td>
</tr>
</tbody>
</table>
Number of trucks of type 1

\[ \begin{align*}
\text{min} & \quad 90x_1 + 60x_2 + 50x_3 + 40x_4 \\
7x_1 + 5x_2 + 4x_3 + 3x_4 & \geq 42 \\
x_1 + x_2 + x_3 + x_4 & \leq 8 \\
x_i & \in \{0, 1, 2, 3\}
\end{align*} \]

Knapsack packing constraint

Knapsack covering constraint

<table>
<thead>
<tr>
<th>Truck type</th>
<th>Number available</th>
<th>Capacity (tons)</th>
<th>Cost per truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>40</td>
</tr>
</tbody>
</table>
Example: Freight Packing and Transfer

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

Slide 49
Knapsack component

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

\[ \sum_{i=1}^{n} Q_i y_i \geq \sum_{j} a_j \]

= 1 if truck \(i\) is selected
Disjunctive component (with embedded knapsack constraint)

\[
\begin{align*}
z_i & \geq c_i \\
\sum_j a_j x_{ij} & \leq Q_i \\
0 \leq x_{ij} & \leq 1, \text{ all } j
\end{align*}
\]

Use continuous relaxation because we want a disjunction of linear systems.

- Truck \( i \) selected
- Truck \( i \) not selected

Cost variable

Cost of operating truck \( i \)

= 1 if package \( j \) is loaded on truck \( i \)
Disjunctive component (with embedded knapsack constraint)

Truck $i$ selected

\[
\begin{aligned}
    z_i & \geq c_i \\
    \sum_j a_j x_{ij} & \leq Q_i \\
    0 & \leq x_{ij} \leq 1, \text{ all } j
\end{aligned}
\]

Truck $i$ not selected

\[
\begin{aligned}
    z_i & \geq 0 \\
    x_{ij} & = 0
\end{aligned}
\]

Convex hull MILP formulation

\[
\begin{aligned}
    z_i & \geq c_i y_i \\
    \sum_j a_j x_{ij} & \leq Q_i y_i \\
    0 & \leq x_{ij} \leq y_i
\end{aligned}
\]
The resulting model

\[ \begin{align*}
\min & \quad \sum_{i=1}^{n} c_i y_i \\
\sum_j a_j x_{ij} & \leq Q_i y_i, \quad \text{all } i \\
0 & \leq x_{ij} \leq y_i, \quad \text{all } i, j \\
\sum_{i=1}^{n} x_{ij} & = 1, \quad \text{all } j \\
\sum_{i=1}^{n} Q_i y_i & \geq \sum_j a_j \\
x_{ij}, y_i & \in \{0,1\}
\end{align*} \]

Disjunctive component

Logical condition (each package must be shipped)

Knapsack component
The resulting model

\[
\begin{align*}
\min & \sum_{i=1}^{n} c_i y_i \\
\sum_{j} a_j x_{ij} & \leq Q_i y_i, \quad \text{all } i \\
0 & \leq x_{ij} \leq y_i, \quad \text{all } i, j \\
\sum_{i=1}^{n} x_{ij} & = 1, \quad \text{all } j \\
\sum_{i=1}^{n} Q_i y_i & \geq \sum_{j} a_j \\
x_{ij}, y_i & \in \{0,1\}
\end{align*}
\]

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

\[
\begin{align*}
\min & \sum_{i=1}^{n} c_i y_i \\
\sum_{j} a_j x_{ij} & \leq Q_i y_i, \quad \text{all } i \\
0 & \leq x_{ij} \leq y_i, \quad \text{all } i, j \\
\sum_{i=1}^{n} x_{ij} & = 1, \quad \text{all } j \\
\sum_{i=1}^{n} Q_i y_i & \geq \sum_{j} a_j \\
x_{ij}, y_i & \in \{0,1\}
\end{align*}
\]

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

<table>
<thead>
<tr>
<th></th>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift 1</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Shift 2</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Shift 3</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>C</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Assign shifts to nurses

<table>
<thead>
<tr>
<th></th>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nurse A</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Nurse B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Nurse C</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Nurse D</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

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 $d$

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

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 \( d \)

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

\[
\text{cardinality}(w \mid (A, B, C, D), (5, 5, 5, 5), (6, 6, 6, 6))
\]

\( A \) 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 \textbf{both} formulations in the same model!

First, assign nurses to shifts.

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

\begin{align*}
\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
\end{align*}

The variables $w_{s,\text{Sun}}, \ldots, w_{s,\text{Sat}}$ 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 $y_{id} = \text{shift assigned to nurse } i \text{ on day } d$

\[
\text{alldiff}(y_{1d}, y_{2d}, y_{3d}), \text{ all } d
\]

\[
\text{stretch}(y_{i,\text{Sun}}, \ldots, y_{i,\text{Sat}} \mid (2,3),(2,2),(6,6),P), \text{ all } i
\]

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:

\[
\begin{align*}
w_{y_{id}d} &= i, \quad \text{all } i,d \\
y_{w_{sd}d} &= s, \quad \text{all } s,d
\end{align*}
\]

Channeling constraints increase propagation and make the problem easier to solve.
The complete model is:

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

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

$$\text{nvalues}(w_{s,\text{Sun}}, ..., w_{s,\text{Sat}} \mid 1, 2), \text{ all } s$$

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

$$\text{stretch}(y_{i,\text{Sun}}, ..., y_{i,\text{Sat}} \mid (2, 3), (2, 2), (6, 6), P), \text{ all } i$$

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

$$y_{w_{sd}d} = s, \text{ all } s, d$$
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 (< 0 if consumed): memory, heat, power, weight, etc.

Minimize $\sum_j c_j v_j$

$v_j = \sum_{ik} a_{ijt} q_i$, all $j$

$L_j \leq v_j \leq U_j$, all $j$

Quantity of component $i$ installed

Amount of attribute $j$ produced by type $t_i$ of component $i$
Integrated model

This is reformulated

\[
\begin{align*}
\min & \sum_j c_j v_j \\
\text{subject to} & \quad v_j = \sum_{ik} q_i A_{ij_t}, \text{ all } j \\
& \quad L_j \leq v_j \leq U_j, \text{ all } j \\
& \quad v_j = \sum_i z_i, \text{ all } j \\
& \quad \text{element}(t_i, (q_i A_{ij_1}, \ldots, q_i A_{ijn}), z_i), \text{ all } i, j
\end{align*}
\]

\(t_i\) is a variable index
Integrated model

\[
\min \sum_j c_j v_j
\]

\[
v_j = \sum_{ik} q_i A_{ijt}, \text{ all } j
\]

\[
L_j \leq v_j \leq U_j, \text{ all } j
\]

\[
v_j = \sum_i z_i, \text{ all } j
\]

\[
\text{element}(t_i, (q_i A_{ij1}, \ldots, q_i A_{ijn}), z_i), \text{ all } i, j
\]

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

\[
\begin{align*}
\text{min} & \quad M \\
M & \geq s_j + p_{x_j}, \quad \text{all } j \\
r_j & \leq s_j \leq d_j - p_{x_j}, \quad \text{all } j \\
\text{disjunctive}\left((s_j | x_j = i), (p_{ij} | x_j = i)\right), \quad \text{all } i
\end{align*}
\]

- Start time variable for job \( j \)
- Processing time of job \( j \) on machine \( x_j \)
- Machine assigned to job \( j \)
- Makespan
Machine Scheduling

The model is

\[ \begin{align*}
\text{min} & \quad M \\
M & \geq s_j + p_{x_{ij}}, \text{ all } j \\
r_j & \leq s_j \leq d_j - p_{x_{ij}}, \text{ all } j \\
\text{disjunctive} & \left( (s_j \mid x_j = i), (p_{ij} \mid x_j = i) \right), \text{ all } i
\end{align*} \]
Machine Scheduling

The model is

\[
\begin{align*}
\min & \quad M \\
M & \geq s_j + p_{x_{ij}}, \text{ all } j \\
r_j & \leq s_j \leq d_j - p_{x_{ij}}, \text{ all } j \\
\text{disjunctive}(s_j | x_j = i), (p_{ij} | x_j = i)), \text{ all } i
\end{align*}
\]
The problem can be solved by logic-based Benders decomposition.

\[
\begin{align*}
\text{min } & M \\
M & \geq s_j + p_{x_{ij}}, \text{ all } j \\
r_j & \leq s_j \leq d_j - p_{x_{ij}}, \text{ all } j \\
\text{disjunctive} & \left((s_j | x_j = i), (p_{ij} | x_j = i)\right), \text{ all } i
\end{align*}
\]

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

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

\[
\begin{align*}
\text{min } & \quad M \\
M & \geq s_j + p_{x_j}, \text{ all } j \\
r_j & \leq s_j \leq d_j - p_{x_j}, \text{ all } j \\
\text{disjunctive} & \left( (s_j | x_j = i), (p_{ij} | x_j = i) \right), \text{ all } i
\end{align*}
\]

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

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.