Section 21.4: Mixed Integer Programming Solvers

Example: The Knapsack Problem

Input: item values $v_1, v_2, \ldots, v_n$; item sizes $s_1, s_2, \ldots, s_n$; knapsack capacity C

Goal: max total value subject to total size $\leq C$.

Ingredients:

- decisions to be made $x_j = \begin{cases} 1 & \text{if included} \\ 0 & \text{if excluded} \end{cases}$
- constraint $\sum_{j=1}^{n} s_j x_j \leq C$ (total size of chosen subset)
- objective function $\max \sum_{j=1}^{n} v_j x_j$ (total value of chosen subset)

Example

$v_1 = 6, s_1 = 5$

$v_2 = 5, s_2 = 4$

$v_3 = 4, s_3 = 3$

$v_4 = 3, s_4 = 2$

$v_5 = 2, s_5 = 1$

$C = 10$

A MIP for Knapsack

maximize $6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5$

subject to $5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 \leq 10$

$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$

$$\sum_{j=1}^{n} w_j x_j \leq W$$

(for 2-D Knapsack)

A MIP for Knapsack

```
Maximize 6 x(1) + 5 x(2) + 4 x(3) + 3 x(4) + 2 x(5)
subject to
5 x(1) + 4 x(2) + 3 x(3) + 2 x(4) + x(5) <= 10
binary
x(1) x(2) x(3) x(4) x(5)
end
```

MIPs More Generally

Ingredients:

- decision variables (0-1, real-valued, etc.)
- linear constraints + objective function

(not allowed: x_j^2, $x_j x_k$, $\frac{1}{x_j}$, e^{x_j}, etc.)

"mixed" = can have both integer- and real-valued decision variables

MIP Input: decision variables $x_1, x_2, \ldots, x_n$; objective function coefficients $c_1, c_2, \ldots, c_n$; for each constraint $i = 1, 2, \ldots, m$, coefficients $a_{i1}, a_{i2}, \ldots, a_{in}$ & right-hand side b_i

also specify whether to min or max

MIP Output: assignment to x_j's with best-possible objective function value, subject to the constraints.

Some Starting Points

Comments: - many NP-hard problems easy to encode as MIPs (you check)

- MIPs useful for TSP (search for "subtour relaxation")
- different formulations of a problem can lead to very different run times; may need to experiment

Commercial solvers: Gurobi Optimizer; CPLEX; FICO Xpress

Non-commercial solvers: SCIP, CBC, MIPCL, GLPK

Solver-independent modeling language: e.g., CVXPY