Wednesday, November 3

Program of the day: (Wolsey chapter 7)

- Solving MIP models by branch-and-bound
- Duality
- Design issues in branch-and-bound
- Strong Branching
- Local Branching
- Applications: Knapsack Problem (demo)
Techniques for MIP

- Preprocessing
- Branch-and-bound
- Valid cuts

Development

1960 Breakthrough: branch-and-bound

1970 Small problems ($n < 100$) may be solved. Exponential growth, many important problems cannot be solved.

1983 Crowder, Johnson, Padberg: new algorithm for pure BIP. Sparse matrices up to ($n = 2756$).

1985 Johnson, Kostreva, Sahl: further improvements.

1987 Van Roy, Wolsey: Mixed IP. Up to 1000 binary variables, several continuous variables.

Now Preprocessing, addition of cuts, good branching strategies
Solving IP by enumeration

- **Binary IP**

  maximize \( 2x_1 + 3x_2 - 1x_3 + 5x_4 \)
  subject to \( 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \)
  \( x_1, x_2, x_3, x_4 \in \{0, 1\} \)

- **Integer IP**

  maximize \( 2x_1 + 3x_2 - 1x_3 + 5x_4 \)
  subject to \( 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \)
  \( x_1, x_2, x_3, x_4 \in \mathbb{N}_0 \)

- **Mixed integer IP**

  maximize \( 2x_1 + 3x_2 - 1x_3 + 5x_4 \)
  subject to \( 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \)
  \( x_1, x_2 \in \mathbb{R} \)
  \( x_3, x_4 \in \{0, 1\} \)
Elements of Branch-and-bound

Problem

\[
\begin{align*}
\text{maximize} & \quad cx \\
\text{subject to} & \quad x \in S
\end{align*}
\]

- **Divide and conquer** (Wolsey prop. 7.1)
  \[S = S_1 \cup S_2 \cup \ldots \cup S_K\] and \[z^k = \max \{ cx : x \in S_k \}\]

\[z = \max_{k=1,\ldots,K} z^k\]

Overlap between \(S_i\) and \(S_j\) is allowed

Often: decompose by splitting on decision variable
Elements of Branch-and-bound

- **Upper bound function** (Wolsey prop. 7.2)
  \[ \bar{z}^k = \sup\{ cx : x \in S_k \} \]
  then
  \[ \bar{z} = \max \bar{z}^k \]
  is an upper bound on \( S \)

- **Lower bound** (so far best solution) \( \underline{z} \)

- **Upper bound test**
  \[ \text{if } \bar{z}^k \leq \underline{z} \text{ then } x^* \not\in S_k \]

Relaxation (Wolsey 2.1)

\[
\begin{align*}
\max \{ cx : x \in S \} \quad & \text{(IP)} \\
\max \{ f(x) : x \in T \} \quad & \text{(RP)}
\end{align*}
\]

RP is a relaxation of IP if
- \( S \subseteq T \)
- \( f(x) \geq cx \) for all \( x \in S \)
Branch-and-bound

A systematical enumeration technique for solving IP/MIP problems, which apply bounding rules to avoid to examine specific parts of the solution space.

\[
\text{maximize } \quad cx \\
\text{subject to } \quad Ax \leq b \\
\quad x' \geq 0 \\
\quad x'' \geq 0, \text{ integer}
\]

- Branching tree enumerates all integer variables.
- Once all integer variables are fixed, remaining problem is solved by LP.
- General MIP algorithm does not know structure of problem
- Upper bounds \( \bar{z} \) are derived in each node by LP-relaxation.
- If \( \bar{z} \leq \overline{z} \) then descendant nodes need not to be examined
**Branch-and-bound for MIP**

Recursive procedure which at each node:

- If infeasible, backtrack
- Solve LP-relaxation, getting $\bar{x}$ and $\bar{z}$
- If $\bar{z} \leq \underline{z}$ then backtrack
- If all $x$ are integral: update $\underline{z}$, backtrack
- Choose a fractional variable $\bar{x}_i = d$
- Branch on
  $$\bar{x}_i \leq \lfloor d \rfloor \quad \bar{x}_i \geq \lceil d \rceil$$

Where

- objective function should be maximized
- $\underline{z}$ is so far best solution (incumbent solution)
- $\bar{z}$ is upper bound at node
- $\bar{x}$ is LP-solution to current problem
Branch-and-bound for MIP

Example:

\[
\begin{align*}
\text{maximize} & \quad x_1 + x_2 \\
\text{subject to} & \quad x_1 + 5x_2 \leq 20 \\
& \quad 5x_1 + 3x_2 \leq 20 \\
& \quad 6x_1 + x_2 \leq 21 \\
& \quad x_1, x_2 \geq 0, \text{ integer}
\end{align*}
\]

Branch on most fractional variable, best-first search
Root node

- LP-solution $x_1 = \frac{20}{11} = 1.8181, x_2 = \frac{40}{11} = 3.6363$.
- Lower bound $\bar{z} = -\infty$.
- Two nodes: $x_2 \leq 3$ and $x_2 \geq 4$ with upper bounds $\bar{z} = 5.2$ and $\bar{z} = 4$.

Node 1

- Add constraint $x_2 \leq 3$, getting LP-solution $x_1 = \frac{11}{5} = 2.2$ and $x_2 = 3$.
- Two nodes: $x_1 \leq 2$ and $x_1 \geq 3$ with upper bounds $\bar{z} = 5$ and $\bar{z} = \frac{14}{3} = 4.6667$.

Node 2

- Add constraint $x_1 \leq 2$, getting LP-solution $x_1 = 2$ and $x_2 = 3$. Upper bound $\bar{z} = 5$. Feasible solution $\underline{z} = 5$.

Node 3

- Add constraint $x_1 \geq 3$, getting LP-solution $x_1 = 3$ and $x_2 = \frac{5}{3} = 1.6667$. Upper bound $\bar{z} = 4.6667 < \underline{z}$.

Node 4

- Add constraint $x_2 \geq 4$, getting LP-solution $x_1 = 0$ and $x_2 = 4$. Upper bound $\bar{z} = 4 < \underline{z}$.
Design issues

\[
\text{maximize } cx \\
\text{subject to } x \in S
\]

Pruning rules (Wolsey 7.2)

- Prune by optimality \( z^k = \max \{ cx : x \in S_k \} \)
- Prune by bound \( \underline{z}_k \leq \underline{z} \)
- Prune by infeasibility \( S_k = \emptyset \)

Branching rules (Wolsey 7.4)

- most fractional variable \( j \) i.e. \( x_j - \lfloor x_j \rfloor \) close to \( \frac{1}{2} \)
- least fractional variable
- greedy approach

Selecting next problem

- Depth-first-search
  (quickly find solution, small changes in LP, space)
- Best-first-search
  (greedy approach)
Duality

Branch-and-bound, economics: upper bound on LP.

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + x_2 + 5x_3 + 3x_4 \\
\text{subject to} & \quad x_1 - x_2 - x_3 + 3x_4 \leq 1 \\
& \quad 5x_1 + x_2 + 3x_3 + 8x_4 \leq 55 \\
& \quad -x_1 + 2x_2 + 3x_3 - 5x_4 \leq 3 \\
& \quad x_1, x_2, x_3, x_4 \geq 0
\end{align*}
\]

(1)

Multiplying the second constraint by two

\[
10x_1 + 2x_2 + 6x_3 + 16x_4 \leq 110
\]

thus \( z^* \leq 110 \).

Linear combination of some constraints: second and third constraint

\[
4x_1 + 3x_2 + 6x_3 + 3x_4 \leq 58
\]

thus \( z^* \leq 58 \).

In general any linear combination.

multipliers \( y_1, y_2, y_3 \), demand \( y_1, y_2, y_3 \geq 0 \)

\[
\begin{align*}
y_1(x_1 - x_2 - x_3 + 3x_4) + \\
y_2(5x_1 + x_2 + 3x_3 + 8x_4) + \\
y_3(-x_1 + 2x_2 + 3x_3 - 5x_4) \leq y_1 + 55y_2 + 3y_3
\end{align*}
\]
which is equivalent to

\[ \begin{align*}
(y_1 + 5y_2 - y_3)x_1 + \\
(-y_1 + y_2 + 2y_3)x_2 + \\
(-y_1 + 3y_2 + 3y_3)x_3 + \\
(3y_1 + 8y_2 + 5y_3)x_4 & \leq y_1 + 55y_2 + 3y_3
\end{align*} \] (2)

coefficients must exceed those in (1):

\[
\begin{align*}
y_1 + 5y_2 - y_3 & \geq 4 \\
-y_1 + y_2 + 2y_3 & \geq 1 \\
-y_1 + 3y_2 + 3y_3 & \geq 5 \\
3y_1 + 8y_2 + 5y_3 & \geq 3 \\
y_1, y_2, y_3 & \geq 0
\end{align*}
\]

minimize the right-hand side of (2).

dual problem:

\[
\begin{align*}
\text{minimize} & \quad y_1 + 55y_2 + 3y_3 \\
\text{subject to} & \quad y_1 + 5y_2 - y_3 \geq 4 \\
& \quad -y_1 + y_2 + 2y_3 \geq 1 \\
& \quad -y_1 + 3y_2 + 3y_3 \geq 5 \\
& \quad 3y_1 + 8y_2 + 5y_3 \geq 3 \\
& \quad y_1, y_2, y_3 \geq 0
\end{align*}
\]
Duality

primal problem.

\[
\text{maximize } \sum_{j=1}^{n} c_j x_j \\
\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, \ldots, m \\
x_j \geq 0 \quad j = 1, \ldots, n
\]

associated dual problem

\[
\text{minimize } \sum_{i=1}^{m} b_i y_i \\
\text{subject to } \sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad j = 1, \ldots, n \\
y_i \geq 0 \quad i = 1, \ldots, m
\]

Weak duality

For every primal feasible solution \((x_1, \ldots, x_n)\) for every dual feasible solution \((y_1, \ldots, y_m)\) :

\[
\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left( \sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left( \sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \sum_{i=1}^{m} b_i y_i
\]
Deriving bounds efficiently

- At each branching node we add one constraint
- New LP-problems needs to be solved
- Can we reuse some computations?

\[
\begin{align*}
\text{maximize} & \quad cx \\
\text{subject to} & \quad Ax \leq b \\
& \quad a'x \leq b' \\
& \quad x \geq 0
\end{align*}
\]

- The previous solution is not feasible!
- Simplex needs feasible solutions in every step
- Consider dual problem

\[
\begin{align*}
\text{minimize} & \quad yb + y'b' \\
\text{subject to} & \quad yA + y'a' \geq c \\
& \quad y, y' \geq 0
\end{align*}
\]

- When primal problem gets additional constraint \( a'x \leq b' \) the dual problem gets one more variable \( y' \)
- The same \( y \) is feasible to dual problem (\( y' = 0 \))
- Same basis solution (for dual problem) can be used
- Normally, only a few steps are needed to find new LP-optimum
The use of interior-point algorithms

- Simplex runs in exponential time (worst-case)
- Interior-point algorithms solve LP-problem in polynomial time
- May be useful for solving MIP problems, if degenerate problem
- Use interior-point to find LP-relaxation at root node
- Derive dual solution (Complementary slackness)
- Use dual simplex at other branching nodes
Design issues

**Relaxation** (Wolsey 2.1)

\[
\max \{ cx : x \in S \} \quad (IP)
\]
\[
\max \{ f(x) : x \in T \} \quad (RP)
\]

RP is a relaxation of IP if

- \( S \subseteq T \)
- \( f(x) \geq cx \) for all \( x \in S \)

Which constraints should be relaxed

- Quality of bound (tightness of relaxation)
- Remaining problem can be solved efficiently
- Constraints difficult to formulate mathematically
- Constraints which are too expensive to write up
Strong branching

Applegate, Bixby, Chvatal, and Cook (1995) for TSP
Linderoth, Savelsbergh (1999) for MIP

Assume binary MIP to be maximized

- Normal branch-and-bound: choose a subproblem, choose a variable to branch at, create two new subproblems. (*sample*)

- If we decide to branch on a variable which has limited or no effect on the LP-bound on subsequent nodes, we have essentially doubled the total work.

- Strong branching exploits a set of candidate variables specified by the user (*several samples*)

- For each candidate variable, test both branches, evaluate upper bounds by solving LP-relaxation (not necessarily to optimality)

- Choose the best variable for branching, and create two new subproblems
Strong branching

Which variable should we choose?

- The ones for which the upper bound of both subproblems is decreased most
- The ones for which the upper bound on average is decreased most

Improving performance

- The samples are only used as a heuristic, hence we do not need to find exact lower bounds
- Dual simplex with a limited number of iterations.
Strong branching, example

maximize \( x_1 + x_2 \)
subject to \( x_1 + 5x_2 \leq 20 \)
\( 5x_1 + 3x_2 \leq 20 \)
\( 6x_1 + x_2 \leq 21 \)
\( x_1, x_2 \geq 0, \) integer

LP-solution: \( x_1 = 1.8181, x_2 = 3.6363, \bar{z} = 5.4545 \)

Only two variables \( \rightarrow \) sample both

- \( x_1 \geq 2: \bar{z} = 5.3333 \)
  \( x_1 \leq 1: \bar{z} = 4.8 \)

- \( x_2 \geq 4: \bar{z} = 5.2 \)
  \( x_2 \leq 3: \bar{z} = 4 \)

Branching \( x_2 \): better upper bounds for both branches
Local branching


- Important to have good incumbent solution
- 2-optimal solution for TSP, QAP, KP works well
- In general: if we have a good feasible solution \( \hat{x} \) we do not want to change it too much
- At most \( k \) variables may change their value from \( \hat{x} \)
- Restrict search to \( k \)-optimal solutions

Example

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 5x_2 + 6x_3 + 7x_4 + 8x_5 \\
\text{subject to} & \quad 3x_1 + 4x_2 + 5x_3 + 6x_4 + 7x_5 \leq 10 \\
& \quad x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}
\end{align*}
\]

Greedy solution: \( \hat{x}_1 = 1, \hat{x}_2 = 1, \hat{x}_3 = 0, \hat{x}_4 = 0, \hat{x}_5 = 0 \).
Restrict to 2-opt
\[
(1 - x_1) + (1 - x_2) + x_3 + x_4 + x_5 \leq 2
\]
we get the constraint
\[
-x_1 - x_2 + x_3 + x_4 + x_5 \leq 0
\]
Other branch demands more than 2 changes
\[
(1 - x_1) + (1 - x_2) + x_3 + x_4 + x_5 \geq 3
\]
Optimal solution: \( x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0 \).
Local branching

Assume that a feasible solution $\hat{x}$ has been found

- Left branch $\Delta(x, \hat{x}) \leq k$
- Right branch $\Delta(x, \hat{x}) \geq k + 1$

Where

$$\Delta(x, \hat{x}) = \sum_{j \in N} |x_j - \hat{x}_j| = \sum_{\{j \in N \mid \hat{y}_j = 1\}} (1 - x_j) + \sum_{\{j \in N \mid \hat{y}_j = 0\}} x_j$$

How large should we choose $k$?

$$k \approx 10$$
Local branching, exact algorithm

1. solution $\hat{x}$
   - $\Delta(x, \hat{x}) \leq k$
   - $\Delta(x, \hat{x}) \geq k + 1$

2. improved $\hat{x}$
   - $\Delta(x, \hat{x}) \leq k$

3. $\Delta(x, \hat{x}) \geq k + 1$

4. improved $\hat{x}$
   - $\Delta(x, \hat{x}) \leq k$

5. $\Delta(x, \hat{x}) \geq k + 1$

6. no improved solution

7. no improved solution
Example: Knapsack Problem

Given $n$ items and a knapsack

- Item $j$ has the weight $w_j$
- Profit of item $j$ is $p_j$
- The capacity of the knapsack is $c$

maximize $\sum_{j=1}^{n} p_j x_j$

subject to $\sum_{j=1}^{n} w_j x_j \leq c$

$x_j \in \{0, 1\}, \quad j = 1, \ldots, n.$

Important problem

- Budgeting
- Transportation
- Subproblem (e.g. separation of valid inequalities)