Introduction to Optimization, DIKU 2007-08
Wednesday 5 December
David Pisinger

Program of the day:

- Post-optimality analysis (Taha 4.5)
- Solving MIP models by branch-and-bound (Wolsey, chapter 7)
- Duality
- Design issues in branch-and-bound
- Strong Branching
- Local Branching

Post-optimality analysis (Taha 4.5)

Periodic recalculation of the optimum solution. Post-optimality analysis determines the new solution in an efficient way.

Condition after parameters change	Recommended action
Current solution remains optimal and feasible	No further action necessary
Current solution becomes infeasible	Use dual simplex to recover feasibility
Current solution becomes nonoptimal	Use primal simplex to recover optimality
Current solution becomes nonoptimal and infeasible	Use generalized simplex to obtain new solution

Adding a variable or constraint

Original problem

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \le b \\ & x \ge 0 \end{array}$$

optimal solution x^* optimal dual y^*

Add a variable x'

$$\begin{array}{ll} \text{maximize} & cx + c'x' \\ \text{subject to} & Ax + A'x' \le b \\ & x, x' \ge 0 \end{array}$$

Primal solution $(x^*, 0)$ is feasible, warm-start simplex.

Add a constraint $a'x \le b'$

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \le b \\ & a'x \le b' \\ & x \ge 0 \end{array}$$

Primal solution is not feasible

Consider dual problem

$$\begin{array}{ll} \text{minimize} & yb + y'b' \\ \text{subject to} & yA + y'a' \ge c \\ & y, y' \ge 0 \end{array}$$

Dual solution $(y^*, 0)$, is feasible, warm-start dual simplex.

Post-optimality analysis

TOYCO Example 4.5-3

$$\begin{array}{lrcrcrcr} \max z = & 3x_1 & + & 2x_2 & + & 5x_3 & & \\ \text{s.t.} & x_1 & + & 2x_2 & + & x_3 & \le & 430 \\ & 3x_1 & & & + & 2x_3 & \le & 460 \\ & x_1 & + & 4x_2 & & & \le & 420 \\ & \multicolumn{7}{l}{x_1, x_2, x_3 \ge 0} \end{array}$$

Solution $x_1 = 0$, $x_2 = 100$, $x_3 = 230$.

- Add constraint $3x_1 + x_2 + x_3 \le 500$
  Redundant
- Add variable x_4 with $c_4 = 9$ and $A_4 = (1, 0, 0)$
  Run primal simplex
- Add constraint $3x_1 + 3x_2 + x_3 \le 500$
  Run dual simplex

Primal simplex (Taha 7.2.2)

0) Construct a starting basis feasible solution and let A_B and c_B be its associated basis and objective.

1) Compute the inverse A_B^{-1}.

2) For each nonbasis variable $j \in N$ compute

$$\overline{c}_j = z_j - c_j = c_B A_B^{-1} A_j - c_j$$

if $\overline{c}_j \geq 0$ for all nonbasis $j \in N$ stop; optimal solution

$$x_B = A_B^{-1} b, \qquad z = c_B x_B$$

Else, apply optimality condition to find entering variable x_s

$$s = \arg\min_{j \in N} \{\overline{c}_j\}$$

3) Compute $\overline{A}_s = A_B^{-1} A_s$.
If $\overline{A}_s \leq 0$ the problem is unbounded, stop.
Else, compute $\overline{b} = A_B^{-1} b$.
Feasibility check

$$k = \arg \min_{i=1,\ldots,m} \left\{ \frac{\overline{b}_i}{\overline{a}_{is}} \middle| \overline{a}_{is} > 0 \right\}$$

Leaving variable: basis variable corresponding to row k, $r = B_k$.

4) New basis is $B := B \cup \{s\} \setminus \{r\}$. Go to step 1.

Dual Simplex (Taha 4.5)

Dual simplex starts from (better than) optimal infeasible basis solution. Searches for feasible solution.

- leaving variable: x_r is the basic variable having most negative value.
- optimality criteria: all basic variables are nonnegative
- entering variable: nonbasic variable with $\overline{a}_{rj} < 0$

$$\min \left\{ \left| \frac{\overline{c}_j}{\overline{a}_{rj}} \right|, \overline{a}_{rj} < 0 \right\}$$

Example

$$\begin{array}{lrcrcrcr}
\min z = & 3x_1 & + & 2x_2 & + & x_3 & & \\
\text{s.t.} & -3x_1 & - & x_2 & - & x_3 & \leq & -3 \\
 & 3x_1 & - & 3x_2 & - & x_3 & \leq & -6 \\
 & x_1 & + & x_2 & + & x_3 & \leq & 3 \\
 & \multicolumn{7}{l}{x_1, x_2, x_3 \geq 0}
\end{array}$$

Simplex table after adding slack variables

basic	x_1	x_2	x_3	x_4	x_5	x_6	solution
z	-3	-2	-1	0	0	0	0
x_4	-3	-1	-1	1	0	0	-3
x_5	3	-3	-1	0	1	0	-6
x_6	1	1	1	0	0	1	3

All reduced costs $\overline{c}_j \leq 0$.
Infeasible

Leaving variable x_5 as -6 smallest.

$$\min \left\{ \left| \frac{\overline{c}}{\overline{a}_{rj}} \right|, \overline{a}_{rj} < 0 \right\} = \min \left\{ -, \frac{2}{3}, 1 \right\} = \frac{2}{3}$$

Entering variable x_2

Iteration 1:

basic	x_1	x_2	x_3	x_4	x_5	x_6	solution
z	-5	0	$-\frac{1}{3}$	0	$-\frac{2}{3}$	0	4
x_4	-4	0	$-\frac{2}{3}$	1	$-\frac{1}{3}$	0	-1
x_2	-1	1	$\frac{1}{3}$	0	$-\frac{1}{3}$	0	2
x_6	2	0	$\frac{2}{3}$	0	$\frac{1}{3}$	1	1

Leaving variable x_4 as -1 smallest

$$\min \left\{ \frac{5}{4}, \frac{1}{2}, 2 \right\} = \frac{1}{2}$$

Entering variable x_3

Iteration 2:

basic	x_1	x_2	x_3	x_4	x_5	x_6	solution
z	-3	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{9}{2}$
x_3	1	0	1	$-\frac{3}{2}$	$\frac{1}{2}$	0	$\frac{3}{2}$
x_2	0	0	0	$\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{3}{2}$
x_6	0	1	0	1	0	1	0

Optimal and feasible

Solution techniques for MIP

- Preprocessing
- Branch-and-bound
- Valid cuts
- Branch-price
- Branch-cut
- Branch-cut-price

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

$$\begin{array}{ll} \text{maximize} & 2x_1 + 3x_2 - 1x_3 + 5x_4 \\ \text{subject to} & 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \\ & x_1, x_2, x_3, x_4 \in \{0,1\} \end{array}$$

- Integer IP

$$\begin{array}{ll} \text{maximize} & 2x_1 + 3x_2 - 1x_3 + 5x_4 \\ \text{subject to} & 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \\ & x_1, x_2, x_3, x_4 \in \mathbb{N}_0 \end{array}$$

- Mixed integer IP

$$\begin{array}{ll} \text{maximize} & 2x_1 + 3x_2 - 1x_3 + 5x_4 \\ \text{subject to} & 4x_1 + 1x_2 + 2x_3 + 3x_4 \leq 8 \\ & x_1, x_2 \in \mathbb{R} \\ & x_3, x_4 \in \{0,1\} \end{array}$$

Elements of Branch-and-bound

Problem

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & x \in S \end{array}$$

- **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)

  $$\overline{z}^k = \sup\{cx : x \in S_k\}$$

  then

  $$\overline{z} = \max \overline{z}^k$$

  is an upper bound on S

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

- **Upper bound test**

  $$\text{if } \overline{z}^k \leq \underline{z} \text{ then } x^* \notin S_k$$

Relaxation (Wolsey 2.1)

$$\max\{cx : x \in S\} \qquad (IP)$$
$$\max\{f(x) : x \in T\} \qquad (RP)$$

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.

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \leq b \\ & x' \geq 0 \\ & x'' \geq 0, \text{integer} \end{array}$$

- 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 $\overline{z}$ are derived in each node by LP-relaxation.
- If $\overline{z} \leq \underline{z}$ then descendant nodes need not to be examined

Branch-and-bound for MIP (maximization)

Maintain pool of open problems. In each iteration take S_j

- If S_j infeasible, backtrack
- Solve LP-relaxation of S_j, getting $\overline{x}$ and $\overline{z}$
- If $\overline{z} \leq \underline{z}$ then backtrack
- If all $\overline{x}$ are integral: update $\underline{z}$, backtrack
- Choose a fractional variable $\overline{x}_i = d$
- Branch on

$$\overline{x}_i \leq \lfloor d \rfloor \qquad \overline{x}_i \geq \lceil d \rceil$$

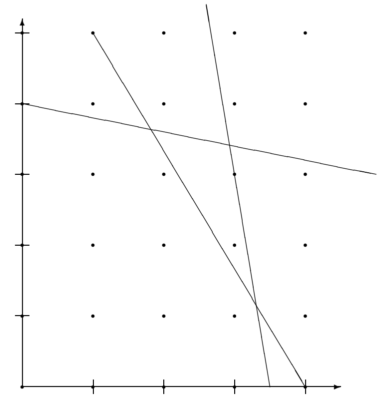
Where

- $\underline{z}$ is so far best solution (incumbent solution)
- $\overline{z}$ is upper bound at node
- $\overline{x}$ is LP-solution to current problem

Branch-and-bound for MIP

Example:

$$\begin{array}{lrcrcl} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & x_1 & + & 5x_2 & \leq & 20 \\ & 5x_1 & + & 3x_2 & \leq & 20 \\ & 6x_1 & + & x_2 & \leq & 21 \\ & x_1, & & x_2 & \geq & 0, \ \text{integer} \end{array}$$



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 $z = -\infty$.
- Two nodes: $x_2 \leq 3$ and $x_2 \geq 4$ with upper bounds $\overline{z} = 5.2$ and $\overline{z} = 4$.

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 $\overline{z} = 5$ and $\overline{z} = \frac{14}{3} = 4.6667$.

Node 2

- Add constraint $x_1 \leq 2$, getting LP-solution $x_1 = 2$ and $x_2 = 3$. Upper bound $\overline{z} = 5$. 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 $\overline{z} = 4.6667 < \underline{z}$.

Node 4

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

Design issues

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & x \in S \end{array}$$

Pruning rules (Wolsey 7.2)

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

Deriving bounds efficiently

- At each branching node we add one constraint
- New LP-problems need to be solved
- Use dual simplex to solve problem
- 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
- Use dual simplex at other branching nodes

Design issues

Relaxation (Wolsey 2.1)

$$\max\{cx : x \in S\} \qquad (IP)$$
$$\max\{f(x) : x \in T\} \qquad (RP)$$

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

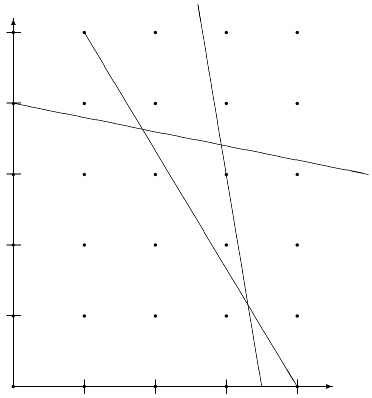
Improving performance

- 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

$$
\begin{array}{llll}
\text{maximize} & x_1 + x_2 & & \\
\text{subject to} & x_1 + 5x_2 & \leq & 20 \\
 & 5x_1 + 3x_2 & \leq & 20 \\
 & 6x_1 + x_2 & \leq & 21 \\
 & x_1, \quad x_2 & \geq & 0, \ \text{integer}
\end{array}
$$

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

Fischetti, Lodi (2003)

- 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

$$
\begin{array}{ll}
\text{maximize} & 4x_1 + 5x_2 + 6x_3 + 7x_4 + 8x_5 \\
\text{subject to} & 3x_1 + 4x_2 + 5x_3 + 6x_4 + 7x_5 \leq 10 \\
 & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}
\end{array}
$$

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

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

How large should we choose k?

$$k \approx 10$$

Local branching, exact algorithm