Duality & Sensitivity

$$
\boxed{\begin{array}{llll}
(P) & \max & Z = & \sum_{j=1}^{n} c_j x_j \\
 & \text{s.t.} & & \sum_{j=1}^{n} a_{ij} x_j \quad \le b_i, \quad \forall i \\
 & & & x_j \quad \ge 0, \quad \forall j
\end{array}}
$$

$$
\boxed{\begin{array}{llll}
(D) & \min & W = & \sum_{i=1}^{m} b_i y_i \\
 & \text{s.t.} & & \sum_{i=1}^{m} a_{ij} y_i \quad \ge c_j, \quad \forall j \\
 & & & y_i \quad \ge 0, \quad \forall i
\end{array}}
$$

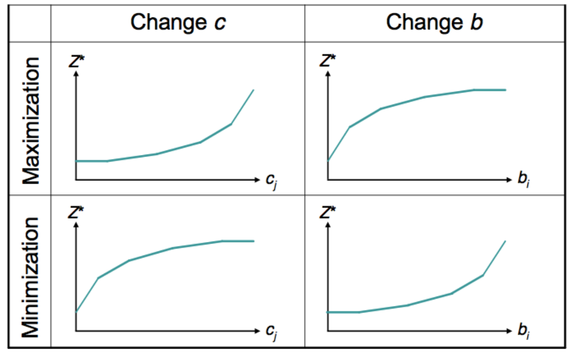
- The Weak Duality Property: $\boldsymbol{cx} \le \boldsymbol{yb}$
- The Strong Duality Property: $\boldsymbol{cx}^* = \boldsymbol{yb}^*$
- Complementary Solution Property: the simplex method simultaneously identifies a CPF solution $\boldsymbol{x}$ and a complementary solution $\boldsymbol{y}$ such that $\boldsymbol{cx} = \boldsymbol{yb}$. ($\boldsymbol{y}$ may be infeasible)
- Duality Theorem: (P) *Feasible* and *Bounded* $\longrightarrow$ (D) *Feasible* and *Bounded*; (P) *Feasible* and *Unbounded* $\longrightarrow$ (D) *Infeasible*; (P) *Infeasible* $\longrightarrow$ (D) *Infeasible* or *Unbounded*
- Shadow Prices y_i^* (0 if constrain not tight)
- Complementary Slackness Conditions:

  $$\sum_{j=1}^{n}(c_j - \sum_{i=1}^{m} y_i^* a_{ij})x_j^* + \sum_{i=1}^{m} y_i^*(\sum_{j=1}^{n} a_{ij}x_j^* - b_i) = 0$$

  For every j, either $x_j^* = 0$ or $\sum_{i=1}^{m} y_i^* a_{ij} = c_j$;
  For every i, either $y_i^* = 0$ or $\sum_{j=1}^{n} a_{ij} x_j^* = b_i$;
  x^* and y^* are optimal $\Longleftrightarrow$ CSCs hold
- Complementary Slackness Theorem: Let $\boldsymbol{x}^*$ be feasible for (P) and $\boldsymbol{y}^*$ be feasible for (D). Then $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are optimal if and only if the complementary slackness conditions hold.
- Universal Transform:

Type	Primal (max)	Dual (min)
Sensible	$\le$	$y_i \ge 0$
Odd	$=$	Unrestricted
Bizarre	$\ge$	$y_i \le 0$
Sensible	$x_j \ge 0$	$\ge$
Odd	Unrestricted	$=$
Bizarre	$x_j \le 0$	$\le$

- Sensitivity Analysis:



Network Optimization

- Network (G), Nodes (V), Arcs (A): $G = (V, A)$
- Tree: connected network with no cycles
- Dijkstra's Algorithm for SPP:

n	Conn. Solved Nodes	Closest Conn. Unsolved	Total Dist.	nth Nearest Node	Min. Dist.	Last Conn.
1	3	5	180	5	180	3-5
2	3	8	195	8	195	3-8
	5	6	272			
3	3	9	246	9	246	3-9
	5	6	272			
	8	7	274			
...	...	...	...	...	...	...

- Greedy Algorithm for Minimum Spanning Tree Problem: 1. choose any node and connect its nearest neighbor; 2. choose node that is closest to the connected nodes until all are connected.
- Augmenting Path: path from source to sink that every arc has a *strictly positive residual capacity.*
- Augmenting Path Algorithm for Maximum Flow Problem: 1. identify an augmenting path; 2. find the minimum residual capacity (c^*) of all arcs on the path; 3. decrease the capacity of each arc by c^*; 4. repeat until no augmenting path exists.
- Max-Flow Min-Cut Theorem: for any network, the maximum feasible flow from the source to the sink *equals* the minimum cut value for all cuts in the network.
- Integer Solutions Property: for minimum-cost flow problems in which every b_i and u_{ij} have in-

teger values, all basic variables in every basic feasible solution, including every optimal one, also have integer values.

- The Min-Cost Flow Problem

$$\begin{array}{rl} \min & \sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{s.t.} & \sum_j x_{ij} - \sum_k x_{ki} = b_i,\ \forall i \\ & l_{ij} \le x_{ij} \le u_{ij} \\ & x_{ij} \ge 0,\ \forall (i,j)\ inE \end{array}$$

- The Balanced Transportation Problem

$$\begin{array}{rl} \min & \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = a_i,\ \forall i \\ & \sum_{i=1}^m x_{ij} = b_j,\ \forall j \\ & x_{ij} \ge 0,\ \forall i,j \end{array}$$

- The Assignment Problem

$$\begin{array}{rl} \min & \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = 1,\ \forall i \\ & \sum_{i=1}^m x_{ij} = 1,\ \forall j \\ & x_{ij} = \{0,1\},\ \forall i,j \end{array}$$

- The Maximum Flow Problem

$$\begin{array}{rl} \min & v \\ \text{s.t.} & \sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} v & \text{if } i = s, \\ -v & \text{if } i = t, \\ 0 & \text{otherwise} \end{cases} \\ & 0 \le x_{ij} \le u_{ij},\ \forall (i,j) \in E \end{array}$$

- The Shortest Path Problem

$$\begin{array}{rl} \min & \sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{s.t.} & \sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = t, \\ 0 & \text{otherwise} \end{cases} \\ & x_{ij} \ge 0,\ \forall (i,j) \in E \end{array}$$

Integer Programming

- LP is easy to solve because there is always an optimal solution that is a CPF solution, which is not necessarily true in IP.

- Fixed-Charge Problems:

$$\min Z = \sum_{j=1}^n (c_j x_j + k_j y_j)$$

- Contingent Decisions:

$$x_j \le M y_j, \quad \forall j = 1, \dots, n$$

in whichi M is a huge number.

- Either-Or Constraints:

$$\text{ConsONE} \le My, \text{ and ConsTWO} \le M(1-y)$$

- K-out-of-N Constraints:

$$\sum_j a_{ij}x_{ij} \le b_i + My_i, \text{ and } \sum_i y_i = N - K$$

- Constraints with m Possible RHSs:

$$\sum_{i=1}^n a_i x_i = \sum_{k=1}^m d_k y_k, \text{ and } \sum_{k=1}^m y_k = 1$$

- $z = x$ `O_R` y: $z \ge x, z \ge y, z \le x + y$
  $z = x$ `AND` y: $z \le x, z \le y, z \ge x + y - 1$

- If-Then Constraints: if $f(x) > 0$, then $g(x) > 0$

$$-g(x) \le My, f(x) \le M(1-y)$$

- Piecewise Linear Approach ($x = \sum_i z_i b_i$):

$$z_1 \le y_1, z_k \le y_{k-1} + y_k, z_n \le y_{n-1}$$

$$\sum_{i=1}^{n-1} y_i = 1,\ \sum_{i=1}^n z_i = 1,\ y_i \in \{0,1\},\ z_i \ge 0$$

- Minimax or Maximin: replace the inner max{} (min{}) with a new decision variable t, which is an upper (lower) bound for all the terms in max{} (min{}).

- Absolute Values ($\min Z = |x_j|$): $x_j = x_j^+ - x_j^-$ with an objective function [$\min Z = x_j^+ + x_j^-$], in which $x_j^+, x_j^- \ge 0$. Then, one of x_j^+ and x_j^- will be **zero** in a *minimization* problem.

- Absolute Values in Constraints:

$$\left|\sum a_i x_i\right| \le b \Rightarrow \sum a_i x_i \le b \text{ and } -\sum a_i x_i \le b$$

- LP Relaxation: $x_i \in \{0,1\} \Rightarrow 0 \le x_i \le 1$

- The LP relaxation of any maximization (min) problem provides an upper (lower) bound on the optimal objective value; The bound attained at child nodes is always less (greater) than or equal to the bound attained at parent nodes.

- Divide and Conquer: partition the set of feasible solutions into smaller subsets (branching), stop partitioning a particular subset if we can tell it's no good (fathoming).
- Branch-and-Bound Algrithm:
  - 0. Initialization: set $Z^* = -\infty$;
  - 1. Branching: among all unfathomed subproblems, select the one that was created most recently. (If tie, choose the one with larger bound.) Branch to create 2 subproblems setting $x_j = 0$ or 1;
  - 2. Bounding: for each new subproblem solve LP relaxation using simplex method and rounding down optimal Z (assumes objective function coefficients are integer);
  - 3. Fathoming: fathom any new subproblem that meets one of the 3 fathoming reasons. If subproblem has integer solution with larger objective value than current incumbent, replace incumbent with it. If there are no remaining (unfathomed) subproblems, stop; the current incumbent is optimal. Otherwise, go to 1.

$x_1 = 1, x_2 = 0$:

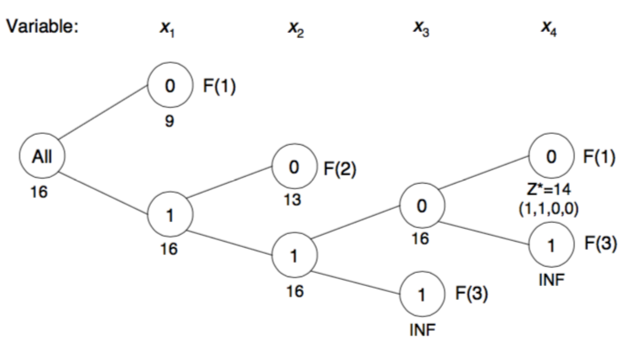
$$\begin{array}{ll} \max Z = 9 + 6x_3 + 4x_4 & \\ \text{s.t.} \quad 5x_3 + 2x_4 \le 4 & \\ \quad x_3 + x_4 \le 1 & \\ \quad x_3 \le 1 & \\ \quad x_4 \le 0 & \\ \quad 0 \le x_3, x_4 \le 1 & \end{array}$$

$\mathbf{x} = (1, 0, \frac{4}{5}, 0) \quad Z = 13\frac{4}{5}$

$x_1 = 1, x_2 = 1$:

$$\begin{array}{ll} \max Z = 14 + 6x_3 + 4x_4 & \\ \text{s.t.} \quad 5x_3 + 2x_4 \le 1 & \\ \quad x_3 + x_4 \le 1 & \\ \quad x_3 \le 1 & \\ \quad x_4 \le 1 & \\ \quad 0 \le x_3, x_4 \le 1 & \end{array}$$

$\mathbf{x} = (1, 1, 0, \frac{1}{2}) \quad Z = 16$



- Fathoming Reason: the solution to the LP relaxation is all-integer; OR the optimal objective value for this LP relaxation is less than or equal to the current incumbent value; OR the LP relaxation is infeasible.

Presentations

- Type of Industry: supply chain management in micro-electronic industry (IBM Microelectronics Division).
- Type of Problem: match assets with demands, to determin which demands it could meet when, and provide manufacturing guidelines.
- Type of Formulation: linear programming (inventory model) with a traditional material resource planning algorithm and a heuristic matching process based on clues established in the explosion algorithm

Additional Lectures

- How to Solve Nonlinear Optimization: use nonlinear solver; take derivatives; numerical search.
- Some Approaches for Large-Scale LPs: column generation; dantzig wolfe decomposition; constraint generation.
- x_{ij}: pounds produced by reactor i under setting j; y_{ij}: 1 if reactor i is in setting j **or above**, 0 otherwise; c_{ij}: unit cost of reactor i in setting j; P_ij: the cumulative capacity on reactor i under setting j.

$$\begin{array}{ll} \min & \sum_{i=1}^{4}\sum_{j=1}^{3} c_{ij}x_{ij} \\ \text{s.t.} & \sum_{i=1}^{4}\sum_{j=1}^{3} x_{ij} \ge 360 \\ & P_{i1}y_{i2} \le x_{i1} \le P_{i1}y_{i1}, \forall i = 1, ..., 4 \\ & (P_{i2} - P_{i1})y_{i3} \le x_{i2} \le (P_{i2} - P_{i1})y_{i2}, \forall i = 1, ..., 4 \\ & 0 \le x_{i3} \le (P_{i3} - P_{i2})y_{i3}, \forall i = 1, ..., 4 \\ & y_{i1} \ge y_{i2} \ge y_{i3}, \forall i = 1, ..., 4 \\ & x_{ij} \ge 0, y_{ij} \in \{0, 1\}, \forall i = 1, ..., 4, j = 1, ..., 3 \end{array}$$

- Consider a multi-period production problem over the time period $t = 1, \ldots, T$. The relevant data are given below: d_t = demand in period t; c_t = fixed cost if anything is produced; p_t = unit production cost; h_t = unit holding cost.

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} (p_t x_t + h_t S_t + c_t y_t) \\
\text{s.t.} \quad & S_{t-1} + x_t = d_t + S_t, \forall t = 1, ..., T \\
& S_0 = S_T = 0 \\
& x_t \leq \left(\sum_{k=t}^{T} d_k \right) y_t, \forall t = 1, ..., T \\
& x_t \geq 0, S_t \geq 0, y_t \in \{0, 1\}, \forall t = 1, ..., T
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{T} \sum_{t=1}^{i} (p_t + \sum_{k=t}^{i-1} h_k) q_{it} + \sum_{t=1}^{T} c_t x_t \\
\text{s.t.} \quad & \sum_{t=1}^{i} q_{it} = d_i, \forall i = 1, ..., T \\
& q_{it} \leq d_i x_t, \forall i, t = 1, ..., T \\
& q_{it} \geq 0, \forall i, t = 1, ..., T, x_t \in \{0, 1\}, \forall t = 1, ..., T
\end{aligned}
$$

- x_{ij} = amount of orders shipped from region i to j using small-parcel; y_i = amount of orders shipped from region 1 to region i using carrier; $z_i = 1$ if MOM shipped orders from region 1 to region i, 0 otherwise.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{27} c_i y_i + \sum_{i=1}^{27} \sum_{j=1}^{27} p_{ij} x_{ij} \\
\text{s.t.} \quad & y_j + \sum_{i=1}^{27} x_{ij} - \sum_{i=2}^{27} x_{ji} \geq q_j, \forall j = 2, ..., 27 \\
& 1000 z_i \leq y_i \leq \left(\sum_{i=1}^{27} q_i \right) z_i, \forall i = 2, ..., 27 \\
& x_{ij}, y_i \in \mathbb{N}^+, z_i \in \{0, 1\}, \forall i, j = 2, ..., 27
\end{aligned}
$$

- XNOR Gate:

$$z_1 \leq 1 - x, z_1 \leq 1 - y, z_1 \geq 1 - x - y$$

$$z_2 \leq x, x_2 \leq y, z_2 \geq x + y - 1$$

$$z \geq z_1, z \geq z_2, z \leq z_1 + z_2$$