LP problem and Simplex algorithm

- Linear algebra techniques are useful in solving LP: maximize or minimize a linear function on decision variables $x_1, \dots, x_n$ under linear constraints. (Chapters 1, 2.)
- Standard form $\max z = c^T x$ subject to $Ax \le b$, $x \ge 0$. Find optimal basic feasible solution(s).
- If $x = (x_1, x_2)$, one can use graphical methods. (Chapters 3, 5.)
- Simplex algorithm for the LP: $\max z = c^T x$ subject to $Ax = b$, $x \ge 0$. (Chapter 4.)

  Set up the initial Tableau if there is a basic feasible solution:

		$c_1x_1 + \cdots + c_nx_n$	constraints
c_B	x_B	$B^{-1}A$	$B^{-1}b$
	$\tilde{C}$	$c^T - c_B^T B^{-1}A$	$Z =?$

  Our last row is the first row in the book. In the book, $c_B B^{-1}A - c$ is computed.

  Select a new basic variable x_j (with maximum relative cost) if not yet optimal.

  Replace a current basic variable x_i based on the minimum $\tilde{b}_i/\tilde{a}_{i,j}$ among those $\tilde{a}_{i,j} > 0$,

  where $\tilde{b} = B^{-1}b = (\tilde{b}_1, \dots, \tilde{b}_m)^T$ and $(\tilde{a}_{1,j}, \dots, \tilde{a}_{m,j})^T$ is the jth column of $B^{-1}A$.

  In the final tableau, we can read out $B^{-1}A, B^{-1}b$, and $\tilde{c} = c^T - c_B^T B^{-1}A$.

  If I_m appears in A, then the corresponding matrix in the final tableau is B^{-1}.

- In order to set up the LP to apply Simplex algorithm, we may need to

  * change $\min z = c^T x$ to $\max \tilde{z} = (-c)^T x$,

  * add slack variables, excess variables, artificial variables.

- If artificial variables are used, one may use big-M methods or two-phase method.

  If an artificial variable stays at the optimal solution, then the LP has no solution.

- In each step of the simplex algorithm, the submatrix $[A|b]$ in the initial tableau changes to $[B^{-1}A|B^{-1}b]$ with $B = B_k \cdots B_1$, where $B_1, \dots, B_k$ are the elementary matrices associated with the Gaussian-Jordan elimination in the previous steps.
- If at a certain step, there is a nonbasic variable x_j with positive $\tilde{c}_j$ such that column $B^{-1}A_j$ has nonpositive entries, the the problem is unbounded.
- Otherwise, we always get the optimal solution (apart from the cycling issue).
- We may set up preemptive goal programming to achieve certain goal priorities by solving

$$\min W = P_1 z_1 + \cdots + P_k z_k \qquad \text{subject to } Ax = b, x \ge 0.$$

  In practice, for $j = 1, \dots, k$, we solve $\min W_j = z_j$ subject to $Ax = b$, $x \ge 0$, with the additional constraints $z_i = \tilde{b}_i$, $i = 1, \dots, j-1$ for $\tilde{b}_1, \dots, \tilde{b}_{j-1}$ attaining the minimum in the previous steps.

Sensitivity analysis (Chapters 5 & 6)

We consider $\max Z = c^T x$ subject to $Ax = b$ with $b \ge 0$.

- If we change c to $\hat{c}$, then $\tilde{C} = c^T - c_B^T B^{-1}A$ changes to $\tilde{C} = \hat{c}^T - \hat{c}_B^T B^{-1}A$.

  * We may still have optimal solution x with different Z value.

  * We may have not have optimal, and need to apply simplex algorithm further.

  * We can examine $\tilde{C} = \hat{c}^T - \hat{c}_B^T B^{-1}A$ to determine the range of change from c to $\hat{c}$ that will not affect the optimal solution x.

- If we change b to $\hat{b}$, we change $x_B = B^{-1}b$ changes to $x_B = B^{-1}\hat{b}$.
  * We still have the optimality condition determined by $\tilde{C}$.
  * If $B^{-1}\hat{b}$ is nonnegative, then we can use the same optimal basis, but a different optimal solution $x_B = B^{-1}\hat{b}$ and a different Z value accordingly.
  * If $B^{-1}\hat{b}$ has negative values, we use dual simplex method.
  * We can compare $B^{-1}(b + e_i)$ and $B^{-1}b$ for $i = 1, \ldots, m$, to determine the range of change of b_i that will affect the optimal solution if $B^{-1}(b + e_i)$ remain feasible.
  * The value $c_B^T[B^{-1}(b + e_i) - B^{-1}b] = c_B^T B^{-1}e_i$ is the shadow price for the ith constraint.

- If we add a variable x_{n+1}, then we add a column A_{n+1} to the tableau and c_{n+1} to the initial tableau so that the final tableau has a column $B^{-1}A_{n+1}$ and $\tilde{c}_{n+1} = c_{n+1} - B^{-1}A_{n+1}$.
  * If $\tilde{c}_{n+1} \le 0$, the x_B remains to be optimal with the same value Z.
  * Else, we apply simplex algorithm to move x_{n+1} to the optimal basis.

- If we add a constraint $a_{m+1,1}x_1 + \cdots + a_{m+1,n}x_n = b_{m+1}$, then we add a row to A to get $\tilde{A}$.

  We may add a slack, excess, or artificial variable x_{n+1} and apply Gaussian-Jordan elimination to get

  * If $\hat{b} \ge 0$, then the same optimal solution with the same Z as $c_{n+1} = 0$.
  * If $\hat{b} < 0$, apply the dual simplex method.

		$c_1x_1 + \cdots + c_nx_n + 0x_{n+1}$	constraints
c_B	x_B	$B^{-1}A$	$B^{-1}b$
0	x_{n+1}	$\hat{R}_{m+1}$	$\hat{b}$
	$\tilde{C}$	$c^T - c_B^T B^{-1}A \quad 0$	$Z = ?$

Dual simplex method (Chapter 6)

- Consider primal LP and the dual LP. There is a standard conversion table.

Primal (Maximize) $\max Z = c^T x$	Dual (Minimize) $\min W = b^T y$
A: coefficient matrix	A^T: coefficient matrix
b: Right-hand-side vector	Cost vector
c: Price vector	Right-hand-side vector
ith constraint is an equation	The dual variable y_i has urs
ith constraint is $\le$ type	The dual variable $y_i \ge 0$
ith constraint is $\ge$ type	The dual variable $y_i \le 0$
x_j has urs	jth dual constraint is an equation
$x_j \ge 0$	jth dual constraint is $\ge$ type
$x_j \le 0$	jth dual constraint is $\le$ type

- Solving the problem by "luck". If x_0, y_0 are primal and dual feasible such that $C^T x_0 = Z = W = b^T y_0$,, then they are the optimal solutions.

- At the optimal solution x_0 and y_0, we have the complementary slackness principle:
$$(y_0^T A - c^T)x_0 = 0 = y_0^T(b - Ax_0).$$

- Solve the primal LP $\max z = c^T x$, $Ax = b$, $x \ge 0$.
  * If we get an optimal solution, then the dual LP has solution $y^T = c_B^T B^{-1}$.
  * If the primal LP is unbounded, then the dual is infeasible.
  * If the primal LP is infeasible, then the dual is infeasible or unbounded.

- If the tableau of the primal attains optimal, but infeasible, then we can apply the dual simplex method to find the feasible solution.

- If $\tilde{b}_j < 0$, choose $\tilde{a}_{ij} < 0$, determine i so that $|\tilde{c}_i/\tilde{a}_{ij}|$ is minimum, and use a_{ij} as the pivoting entry.

Chapter 7 Transportation problem and transshipment problem

Transportation problem $\min z = \sum_{ij} c_{ij}x_{ij}$ s.t. $\sum_j x_{ij} \le s_i, \sum_i x_{ij} \ge d_j,\ x_{ij} \ge 0.$

1. Convert it to a balanced problem. Add dummy demand column with 0 cost;

   add dummy supply with penalty cost (in different form).

2. Use NW corner method, minimum cost, or Vogel;s method to find a bfs.

3. Set $u_1 = 0$ and solve $u_i + v_j = c_{ij}$ for the basic variables.

4. If $(u_i + u_j) \le c_{ij}$ for all i, j, then we have optimal.

5. Else choose x_{ij} to be the entering basic variable, where maximum $u_i - u_j - c_{ij} > 0$.

6. Choose a loop and Δ to determine the leaving variable; determine the new feasible solution.

Remarks a) Do suitable adjustment if one considers $\max z = c_{ij}x_{ij}$.
b) One can do sensitivity analysis by changing c_{ij} or changing s_i, d_j simultaneously.
c) One can use the techniques do inventory problem, and transshipment problem, etc.
d) One can also do assignment problem; the Hungarian method is preferred.

Chapter 8 Network models

Shortest path problem Dijkstra's algorithm.

Remarks a) One can formulate the problem as a transshipment problem.
b) One can use the method to solve equipment replacement problem.

Maximum flow / minimum cut problem Ford-Fulkerson Algorithm.

1. Find an initial flow; then find $so - si$ chain to improve the flow.

2. If there is no $s0 - si$ chain then the flow is maximum; one can determine the minimum cut.

Remark Special case include the optimal matching.

Critical Path Method and Project Management and Review Techniques

1. Formulate the network problem (label the activities as edges).

2. Find the critical path (by minimum path algorithm (in terms of the total flow).

3. One may crash the project in m days by solving the LP problem with n nodes:

   $\min z = \sum_{ij} c_{ij}A_{ij}$ s.t. $x_j - x_i \ge d_{ij} - A_{ij},\quad x_n - x_1 \le m,\qquad x_j$ urs, $\quad 1 \le A_{ij} \le m_{ij}.$

4. Note that one can set $x_1 = 0$ so that $x_j \ge 0$.

5. Using statistical techniques, one can evaluate and estimate the completion day. But there are limitation.

Mimumum spanning tree An easy greedy algorithm.

Minimum cost network problem $\min z = c_{ij}x_{ij}, \quad \sum_j x_{ij} - x_{ki} = b_i, \quad L_{ij} \le x_{ij} \le U_{ij}.$

Network Simplex method

1 Determine a starting bfs. The $n-1$ basic variables will correspond to a spanning tree.

Indicate nonbasic variables at their upper bound by dashed arcs.

2 Compute $y_1, y_2, ...y_n$ (often called the simplex multipliers) by solving

$y_1 = 0$, $y_i - y_j = c_{ij}$ for all basic variables x_{ij}.

For all nonbasic variables, determine the first / last row coefficient $\tilde{c}_{ij} = y_i - y_j - c_{ij}$.

The current bfs is optimal if $\tilde{c}_{ij} \le 0$ for all $x_{ij} = L_{ij}$ and $\tilde{c}_{ij} \ge 0$ for all $x_{ij} = U_{ij}$.

If the bfs is not optimal, choose the nonbasic variable that most violates the optimality conditions as the entering basic variable.

3 Identify the cycle (there will be exactly one!) created by adding the arc corresponding to the entering variable to the current spanning tree of the current bfs.

Use conservation of flow to determine the new values of the variables in the cycle.

The variable that exits the basis will be the variable that first hits its upper or lower bound as the value of the entering basic variable is changed.

4 Find the new bfs by changing the flows of the arcs in the cycle found in step 3. Now go to step 2.

Integer Programming

- Many LP program may require some or all the variables $x_i$to be integers.
- We have mixed or pure IP. Sometimes, we require $x_i \in \{0, 1\}$.
- One may change all IP constraints to 0-1 constraints using binary numbers $x = u_n 2^n + \cdots + u_0$.

Branch and bound method

1. Solve Subproblem - LP relaxation.
2. Branch at the variables assuming fractional values.
3. Each branching remove some non-integral points in subproblem 1.
4. For each subprogram, one may fathom the corresponding node if:

   a) an integer optimal solution (a candidate solution) for the subproblem is found,

   b) the subproblem is infeasible,

   c) the subproblem has optimal solution less than or equal to a candidate solution.

Remarks One may consider the special cases:

a) knapsack problem (use c_i/a_i ratios to find an initial solution),

b) TSL problem, and 0-1 (use assignment problem to solve subproblem),

c) 0-1 problem, use optimal choices of x_i for $\max z = \sum_i c_i x_j$ and $\sum_j a_{ij} x_j \le b_i$;
implicit enumeration method can be employed.