Lecture 8

Discrete optimization modeling II

Content

- Discrete optimization modeling (2)
  - Linear programming: the simplex method

3. Linear programming: the simplex method

Recall: what kind of problem are we solving

- Focus here on the following type of discrete optimization problem:
  - Optimize (i.e. minimize or maximize) linear function $c_1x_1 + c_2x_2 + \dots + c_nx_n$ subject to linear constraints:
    - $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \le b_1$
    - $a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \le b_2$
    - $\dots$
    - $a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \le b_m$
    - $x_1, x_2, \dots, x_n$ are the variables of the model
  - For the objective function (to be optimized):
    - Discrete optimization problems are solvable with integer optimization
    - Integer optimization problems are sometimes solvable with linear optimization

Recall from the last lecture

- Without loss of generality, we can assume that the problem to solve is of the form:

Maximize/minimize $c_1x_1 + c_2x_2 + \dots + c_nx_n$

subject to constraints $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \le b_i \forall i \in I$

where $x_1, x_2, x_3, \dots, x_n \ge 0$

- Simply discount trying inequalities in equations through new non-negative variables, for example:
  - For variables with a lower bound other than 0, say $x_1 \ge l_1$, introduce a new variable $x_1' = x_1 - l_1$ and replace x_1 by $x_1' + l_1$
  - Replace each unrestricted variable, say x_1, with $x_1' - x_1''$

Note: The case of minimization problems is similar, requires some minor modifications to the algorithm, skip it here.

Recall from the last lecture: the geometric approach

- Consider again the parameter example:
  - The objective function to maximize has $c_1 = 1, c_2 = 2$
  - Solve the ones below:
    - Inequalities $x_1 \ge 0$ and $x_2 \ge 0$ (variables cannot take negative values)
    - $x_1 + x_2 \le 4$ (we can use at most 4 units of each of the two resources)
    - $x_1 \le 2$ (we can use at most 2 units of the first resource)

Feasible region for the example of the simplex method (the shaded area)

Standard form

- The standard form of a linear program:
  - Maximize **c.x**
  - Subject to **A.x=b, x≥0**

where
  - $x=(x_1, \dots, x_n)$ is the vector of variables,
  - $c=(c_1, \dots, c_n)$ is the coefficient of the objective function,
  - $b = (b_1, \dots, b_m)$ and
  - $A=(a_{ij})_{1 \le i \le m, 1 \le j \le n}$ are constant matrices with sizes

Note: The case of minimization problems is similar, requires some minor modifications to the algorithm, skip it here.

Recall from the last lecture: variables

- Enumerating all intersection points for n=2 variables
  - Without loss of generality assume that the constraints are linear inequalities $a_1x_1 + a_2x_2 \le b$, where $x_1, x_2 \ge 0$
  - Add slack variables $s_1, s_2 \ge 0$ to transform inequalities above into a set of equations $a_1x_1 + s_1 = b$ and $a_2x_2 + s_2 = b$
    - x_1 and x_2 represent a point of the border of the solutions
  - Repeat for all other constraints (two variable for each)
    - Some of the points will be outside the feasible region, but this is fine since we can negative images
  - For all possible pairs of 2 variables, set the variables to 0 or 0 and solve the resulting equations and 2 variables, with n variables to pick and intersection points
  - Select the best one of the intersection points, as well as the optimum

The algebraic approach yields intersection points that may be unfeasible. Improvements

- Step 1. Drop all the computation of unfeasible points
  - Intersection points outside the feasible region
- Step 2. Drop all the computation of intersection points that cannot improve the current optimum
  - Step 2 of the simplex proof is not optimal. Show that there is a better intersection point, or the current one is optimal. Use the properties of the sliding parallel line given by the linear objective function
  - Intersection points
- Step 3 (cycle prevention). Improve the optimum as much as possible in every step of the algorithm
  - Step 2 of the simplex algorithm, we will now take the optimum adjacent intersection point to improve the optimum. Avoid to go back to previous intersection points

The simplex method

- Solution: the simplex method
  - Proposed by George Dantzig (1947)
  - Very efficient in practice (normally) and secretly (a Leonid Kantorovich Prize)
  - Exponential time complexity in the worst case (rare)
  - Historical note: Dantzig's solutions solved open problems in statistics while a graduate student at Berkeley (1947)

Simplex algorithm: input

- The linear program is in standard form:
  - Maximize **c.x**
  - Subject to **A.x=b, x≥0**
- Assume the objective function into a constraint-like condition:
  - Introducing c_0 to maximize (we variables c_0 and x_0 to zero, or in other words $c_0 = c.x$)
  - With correspondence to point $(x_1, x_2, \dots, x_n) = (x_1, x_2, \dots, x_n)$
- Represent now the linear program as a tableau (table):
  - Use equations to express both the objective function and the slack variables (initially, the slack variables are equal to the constraints)
  - Use one for each of the equations (one is each column the coefficient of the variables and the constant in the last column)
  - Place equations according to the slack variable of independent variables, that are those with a coefficient of 1 in one equation and 0 in the other equations
    - Use variables added to the objective function (initially it depends variables for the slack variables)

An example

- $c_0 = c_1x_1 + c_2x_2 + c_3x_3$
- $s_1 = b_1 - a_{11}x_1 - a_{12}x_2 - a_{13}x_3$
- $s_2 = b_2 - a_{21}x_1 - a_{22}x_2 - a_{23}x_3$
- $s_3 = b_3 - a_{31}x_1 - a_{32}x_2 - a_{33}x_3$

$c_0 = 3x_1 + 5x_2 + 6x_3$
$s_1 = 4 - x_1 - x_2 - x_3$
$s_2 = 7 - 2x_1 - x_2 - x_3$
$s_3 = 9 - x_1 - 2x_2 - 2x_3$

$c_0 = 3x_1 + 5x_2 + 6x_3$
$s_1 = 4 - x_1 - x_2 - x_3$
$s_2 = 7 - 2x_1 - x_2 - x_3$
$s_3 = 9 - x_1 - 2x_2 - 2x_3$

$$\begin{pmatrix} 0 & 3 & 5 & 6 \\ 4 & 1 & 1 & 1 \\ 7 & 2 & 1 & 1 \\ 9 & 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} c_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

An example (2)

- $c_0 = c_1x_1 + c_2x_2 + c_3x_3$
- $x_1 = 4 - s_1 - x_2 - x_3$
- $s_2 = 7 - 2x_1 - x_2 - x_3$
- $s_3 = 9 - x_1 - 2x_2 - 2x_3$

- Same as before, will move to an intersection point that increases the objective function, in this case the variable of a 1 in the coefficient of the last equation is 0 (the variable of a 1 in the coefficient of the last equation is 1 (the same for the 1)
  - $c_0 = c_1x_1 + 5x_2 + 6x_3$
  - $x_1 = 4 - s_1 - x_2 - x_3$
  - $s_2 = 7 - 2x_1 - x_2 - x_3$
  - $s_3 = 9 - x_1 - 2x_2 - 2x_3$
- Some do pivoting with respect to an intersection point that increases the objective function, in this case the variable of a 1 in the coefficient of the last equation is 0 (the variable of a 1 in the coefficient of the last equation is 1 (the same for the 1)
  - Solve the first coefficient c_1 by 2 in equations 2 and 3 and choose the lowest one, in this case the third one
  - $c_0 = 3x_1 + 5x_2 + 6x_3$
  - $x_1 = 4 - s_1 - x_2 - x_3$
  - $x_2 = 3/2 - s_2/2 - x_1/2 - x_3/2$
  - $x_3 = 3/2 - s_3/2 - x_1/2 - x_2$
  - Solve the second equation because $x_2 = 4/0$!

An example (3)

- $c_0 = c_1x_1 + c_2x_2 + c_3x_3$
- $x_1 = 4 - s_1 - x_2 - x_3$
- $x_2 = 3/2 - s_2/2 - x_1/2 - x_3/2$
- $x_3 = 3/2 - s_3/2 - x_1/2 - x_2$

- There is no more variable with a negative coefficient in the last equation, so we are done. The optimum is achieved for variables s_1, s_2 and s_3 and maximum our objective function:
- Solution: (12,15,0,0,7/2), i.e., maximum value is 795, for $x_1=12$, $x_2=15$

Simplex algorithm: tableau format

- The linear program is in standard form:
  - Maximize **c.x**
  - Subject to **A.x=b, x≥0**
- Assume the objective function into a constraint-like condition:
  - Introducing c_0 to maximize (we variables c_0 and x_0 to zero, or in other words $c_0 = c.x$)
- Represent now the linear program as a tableau (table):
  - Use equations to express both the objective function and the slack variables (initially, the slack variables are equal to the constraints)
  - Use one for each of the equations (one is each column the coefficient of the variables and the constant in the last column)
  - Place equations according to the slack variable of independent variables, that are those with a coefficient of 1 in one equation and 0 in the other equations
    - Use variables added to the objective function initially it depends variables for the slack variables

Tableau format – example

- Example: the carpenter's problem: Maximize $2x_1+3x_2$ such that:
  - $x_1 + x_2 \le 4$
  - $x_1 \le 2$
- Transform the problem into the set of:
  - $c_0 = 2x_1 + 3x_2$
  - $s_1 = 4 - x_1 - x_2$
  - $s_2 = 2 - x_1$
- Tableau:

Objective function	c_0	-2	-3	0
	s_1	1	1	4
	s_2	1	0	2

Objective function: $c_0 = 0 + 2x_1 + 3x_2$

Dependent variables: $s_1 = 4 - x_1 - x_2$, $s_2 = 2 - x_1$

Independent variables: x_1, x_2

Value of the objective function: $c_0 = 0$

Value of all the dependent variables: $s_1 = 4$, $s_2 = 2$

Outline of the Simplex Method

- Step 0 (initialization). Preprocess a given linear programming problem so that it is in the standard form, and construct the initial tableau.
- Step 1 (optimality test). Examine the objective function in the current tableau. If it is a constant, then it is the optimum. Stop.
- Step 2 (find entering variable). Select the smallest negative entry in the objective function. The corresponding variable is the entering variable. If there is no such variable, the current solution is optimal. Stop.
- Step 3 (find leaving variable). For each entry in the column of the entering variable, calculate the ratio of the constant in the last column to the entry. Choose the row with the smallest positive ratio. If there is no such row, the problem is unbounded. Stop.
- Step 4 (Pivot the new tableau). Divide all the entries in the pivot row by the pivot element. Add multiples of the pivot row to other rows so that all other entries in the pivot column become zero. Replace the dependent variable in the pivot row by the entering variable. Go back to Step 1.

First step: find one initial extreme point

- The algorithm starts implicitly by choosing the origin as the initial extreme point
  - all independent variables equal to 0

Next step: apply the optimality test

- Examine the last equation (corresponding to the objective function)
  - If all coefficients are nonnegative, then stop: the current extreme point is optimal
  - Otherwise, choose the variable with the smallest negative coefficient (or larger absolute value) to be the variable that will be added to the dependent variables
    - Example: the variable x_1 has the smallest negative coefficient, so it is the variable that will be added to the dependent variables

c_0	-2	-3	0	0
s_1	1	1	1	4
s_2	1	0	0	2

Next step: apply the feasibility test

- For each row of the tableau, divide its right-hand side value by the coefficient shown in the column of x_1
  - Choose the entry variable (that has the smallest ratio) as the pivot
  - Example: the ratio for the first row is 4/1 and for the second one is 2/1 (the coefficient in the row)
    - Choose the second row as the pivot

c_0	-2	-3	0	0
s_1	1	1	1	4
s_2	1	0	0	2

Entering variable: x_1

Ratio: $4/1 = 4$, $2/1 = 2$

Next step: pivoting

- Pivoting to solve for the new dependent variable x_1
  - Divide the pivot row by the pivot element, in this case 1, so the row remains the same (if it is not 1, divide all the coefficients by the pivot element)
  - Add multiples of the pivot row to other rows so that the entry in the column x_1 is zero (in this case, the entry is already zero coefficient)
  - Replace the dependent variable in the pivot row by the entering variable, in this case s_2 is replaced by x_1
  - Compute the value of the new dependent variable x_1 by the pivot row

c_0	-2	-3	0	0
s_1	1	1	1	4
s_2	1	0	0	2

Entering variable: x_1 Leaving variable: s_2

Dependent variables: $s_1 = 2 - x_2$, $x_1 = 2 - s_2$

Independent variables: s_2, x_2

Value of the objective function: $c_0 = 4$

Iterate

- A new intersection point has thus been found
- Repeat the whole procedure starting from this new intersection point

Example (continued)

c_0	-2	-3	0	0
s_1	1	1	1	4
s_2	1	0	0	2

Entering variable: x_2

Ratio: $4/1 = 4$, $2/1 = 2$

Dependent variables: $s_1 = 2 - x_2$, $x_1 = 2 - s_2$

Independent variables: s_2, x_2

Value of the objective function: $c_0 = 4$

c_0	-2	-3	0	0
s_1	1	1	1	4
s_2	1	0	0	2

Entering variable: x_2 Leaving variable: s_1

Dependent variables: $x_2 = 2 - s_1$, $x_1 = 2 - s_2$

Independent variables: s_1, s_2

Value of the objective function: $c_0 = 10$

Another example

maximize $z = 3x + 5y + 6z + 0r$

subject to $x + y \le 4$, $x + 2y + z \le 7$, $x + 2y + z + r = 9$

$x \ge 0$, $y \ge 0$, $z \ge 0$, $r \ge 0$

z	-3	-5	-6	0	0
x	1	1	1	0	4
y	1	2	1	0	7
r	1	2	2	1	9

Current tableau: $z = 0 + 3x + 5y + 6z$, $x = 4 - x - y - z$, $r = 9 - x - 2y - 2z$

Dependent variables: $x = 4 - x - y - z$, $r = 9 - x - 2y - 2z$

Independent variables: x, y, z

Value of the objective function: $z = 0$

z	-3	-5	-6	0	0
x	1	1	1	0	4
y	1	2	1	0	7
r	1	2	2	1	9

Current tableau: $z = 0 + 3x + 5y + 6z$, $x = 4 - x - y - z$, $r = 9 - x - 2y - 2z$

Dependent variables: $x = 4 - x - y - z$, $r = 9 - x - 2y - 2z$

Independent variables: x, y, z

Value of the objective function: $z = 0$

Computational complexity of simplex

- Simplex performs well in practice
  - It has exponential worst-case complexity
    - Here is a class of linear programs for which it runs in exponential time
    - Starting a number of vertices that has been proposed by Victor Klee and Minty
  - It has polynomial average-case complexity under various probability distributions of the input
- There are methods for solving linear programs in polynomial time
  - the ellipsoid method (1979) - polynomial but impractical: $O(n^4L)$ where n is the number of variables and L is the number of bits in the input
  - Karmarkar's algorithm (1984) – the first practical alternative polynomial time algorithm: $O(n^{3.5}L)$ – $O(n^3L)$ – $O(n^{2.5}L)$ (performance comparable to simplex on large problems)

Learning objectives

- Understand the conceptual difference between solving an optimization problem and solving a constraint satisfaction problem
- The difference holds for as well in these two cases: optimization combinatorics (with support from geometry, algebra, or algorithmics)
- Understand the concept of linear programs, integer problems
- Understand the basic concepts in the geometric approach to linear programming: intersection points, feasible regions, the sliding parallel line, objects, intersection points, combination of the intersection points
- Understand the algebraic approach to simplex: slack algorithm, intersection points, the intersection point is another along the edges of the feasible set