Computer Algorithms

Lecture 11: Iterative Improvement – Ch 10

Dr. Manal Helal, Spring 2014.

http://moodle.manalhelal.com

Lecture Learning Objectives

1. Use a Iterative algorithm design strategy to solve optimisation problems

Greedy Technique

- Constructs a solution to an *optimization problem* piece by
- piece through a sequence of choices that are:
    - *feasible*
    - *locally optimal*
    - *irrevocable*
- For some problems, yields an optimal solution for every instance.
- For most, does not but can be useful for fast approximations.

Applications of the Greedy Strategy

Optimal solutions:

- change making for “normal” coin denominations
- minimum spanning tree (MST)
- single-source shortest paths
- simple scheduling problems
- Huffman codes

Approximations:

- traveling salesman problem (TSP)
- knapsack problem
- other combinatorial optimization problems

Change-Making Greedy Approach

Given 30 cents, and coins {1, 5, 10, 25}

§ Here is what a casher will do: always go with coins of highest value first

– Choose the coin with highest value 25 (1 quarter)

– Now we have 5 cents left (1 nickel)

The solution is: 2 (one quarter + one nickel)

Coins = {1, 3, 4, 5} and change required = 7 cents ?

§ Greedy solution:

– 3 coins: one 5 + two 1

§ Optimal solution:

– 2 coins: one 3 + one 4

Iterative Improvement

- Algorithm design technique for solving optimization problems

- Start with a feasible solution, greedy method for example.
- Repeat the following step until no improvement can be found:
  - change the current feasible solution to a feasible solution with a better value of the objective function
- Return the last feasible solution as optimal

- Note: Typically, a change in a current solution is "small" (*local search*)

- Major difficulty: Local optimum vs. global optimum

Important Examples

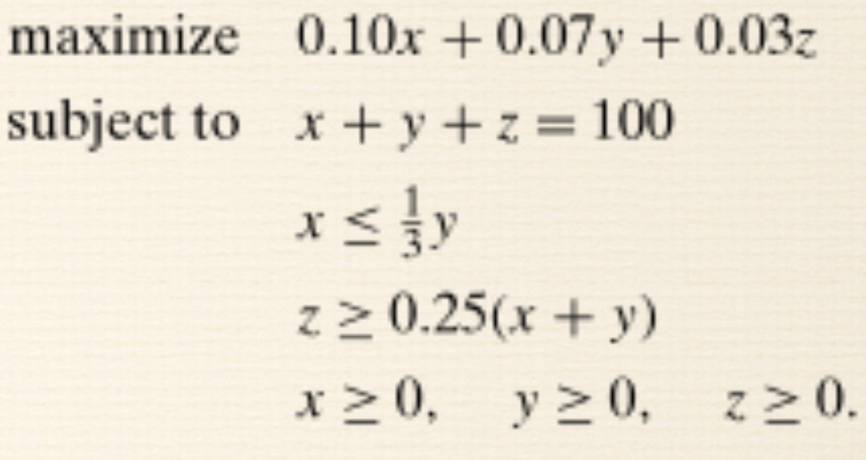
- Simplex method
- Ford-Fulkerson algorithm for maximum flow problem
- Maximum matching of graph vertices
- Gale-Shapley algorithm for the stable marriage problem
- Local search heuristics

Optimisation

EXAMPLE: Consider a university endowment that needs to invest \$100 million. This sum has to be split between three types of investments: stocks, bonds, and cash, expecting an annual return of 10%, 7%, and 3% for their stock, bond, and cash investments, respectively. Stocks are more risky than bonds, the endowment rules require the amount invested in stocks to be no more than one-third of the moneys invested in bonds. In addition, at least 25% of the total amount invested in stocks and bonds must be invested in cash. How should the managers invest the money to maximize the return?

Linear Programming Solution

x is amount invested in Stocks
y is amount invested in Bonds
z is amount invested in Cash.



$$\begin{array}{ll} \textbf{maximize} & 0.10x + 0.07y + 0.03z \\ \textbf{subject to} & x + y + z = 100 \\ & x \le \frac{1}{3}y \\ & z \ge 0.25(x + y) \\ & x \ge 0, \quad y \ge 0, \quad z \ge 0. \end{array}$$

Knapsack LP Formalisation

EXAMPLE 2: Given a knapsack of capacity W and n items of weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$, find the most valuable subset of the items that fits into the knapsack.

In Fractional representation: let x_j, j = 1, . . . , n, be a variable representing a fraction of item j taken into the knapsack, subject to $0 \le x_j \le 1$.

$$\text{maximize} \quad \sum_{j=1}^{n} v_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le W$$

$$0 \le x_j \le 1 \quad \text{for } j = 1, \ldots, n.$$

In Integer LP representation (discrete or 0-1) we are only allowed either to take a whole item or not to take it at all.

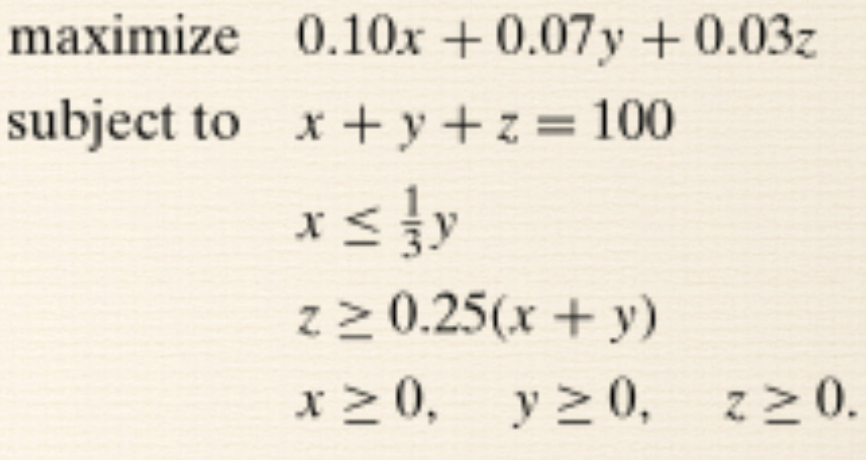
$$\text{maximize} \quad \sum_{j=1}^{n} v_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le W$$

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

Linear Programming Solution

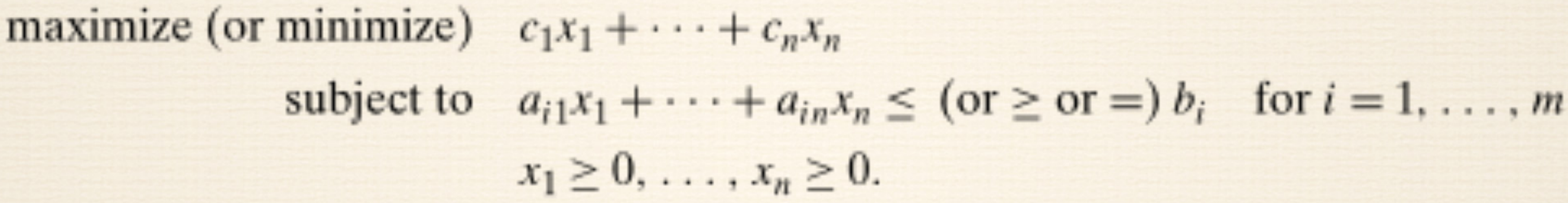
x is amount invested in Stocks
y is amount invested in Bonds
z is amount invested in Cash.



$$
\begin{array}{ll}
\textbf{maximize} & 0.10x + 0.07y + 0.03z \\
\textbf{subject to} & x + y + z = 100 \\
 & x \le \frac{1}{3}y \\
 & z \ge 0.25(x + y) \\
 & x \ge 0, \quad y \ge 0, \quad z \ge 0.
\end{array}
$$

Linear Programming

Linear programming (LP) problem is to optimize a linear function of several variables subject to linear constraints:

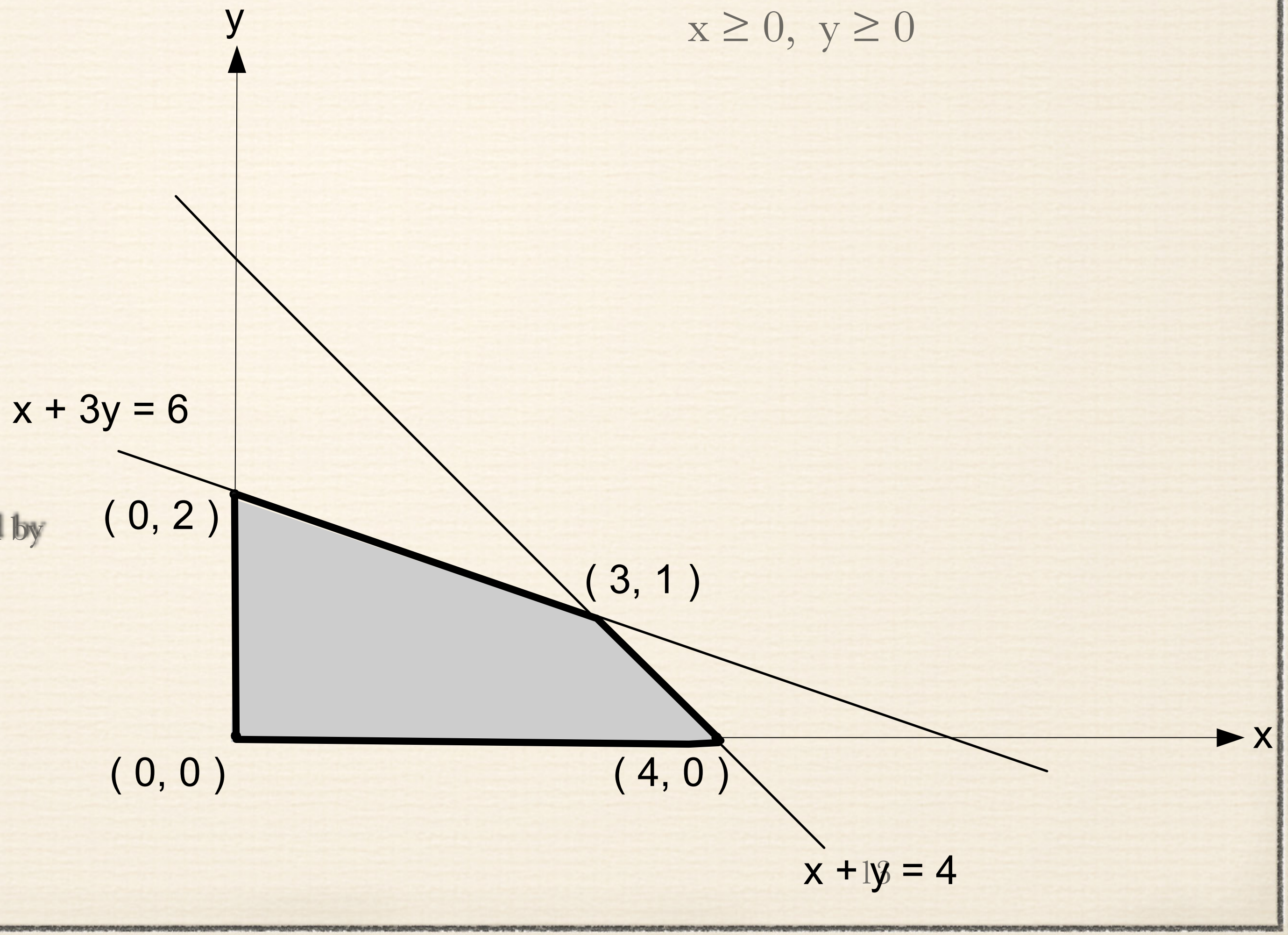


$$\begin{array}{rl} \text{maximize (or minimize)} & c_1x_1 + \cdots + c_nx_n \\ \text{subject to} & a_{i1}x_1 + \cdots + a_{in}x_n \le \text{ (or } \ge \text{ or } =) \; b_i \quad \text{for } i = 1, \ldots, m \\ & x_1 \ge 0, \ldots, x_n \ge 0. \end{array}$$

The function $\boldsymbol{z = c_1 x_1 + \ldots + c_n x_n}$ is called the *objective function*;
constraints $\boldsymbol{x_1 \geq 0, \ldots , x_n \geq 0}$ are called *nonnegativity constraints*

Example

$$\begin{aligned} \text{maximize} \quad & 3x + 5y \\ \text{subject to} \quad & x + y \leq 4 \\ & x + 3y \leq 6 \\ & x \geq 0, \ y \geq 0 \end{aligned}$$



Feasible region is the set of points defined by the constraints

Geometric solution

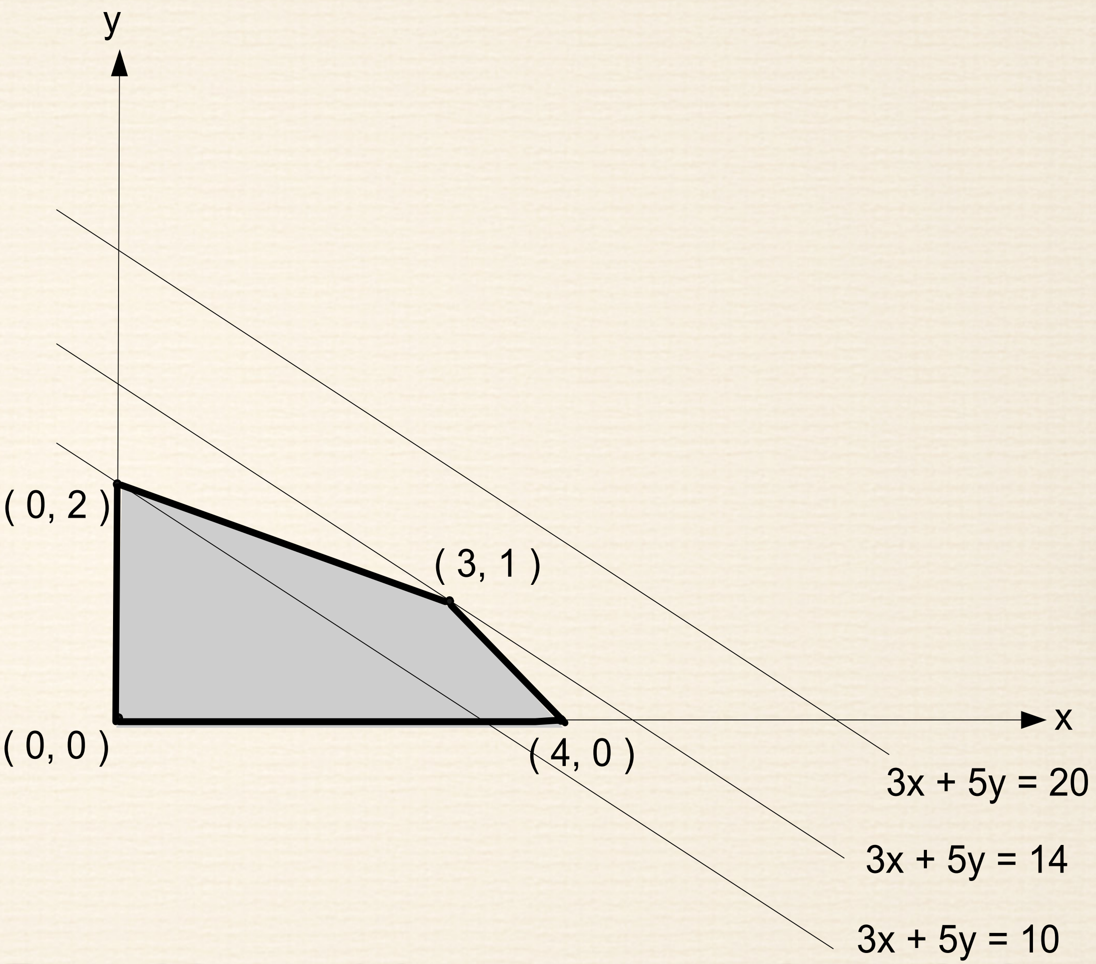
maximize $3x + 5y$

subject to $x + y \leq 4$

$x + 3y \leq 6$

$x \geq 0, \; y \geq 0$

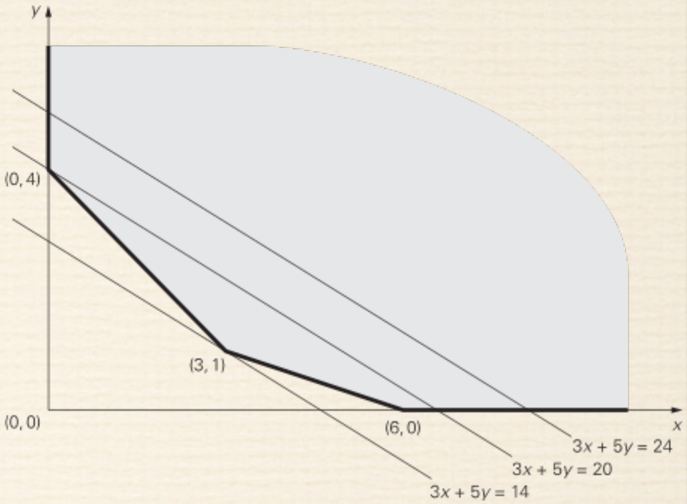
Optimal solution: $x = 3, y = 1$



<u>Extreme Point Theorem</u> Any LP problem with a nonempty bounded feasible region has an optimal solution; moreover, an optimal solution can always be found at an *extreme point* of the problem's feasible region.

3 possible outcomes in solving an LP problem

- has a finite optimal solution, which may not be unique



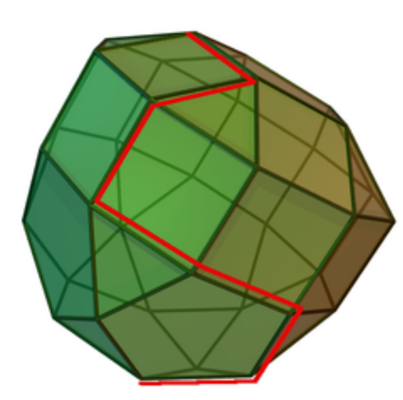
- *unbounded*: the objective function of maximization (minimization) LP problem is unbounded from above (below) on its feasible region

- *infeasible*: there are no points satisfying all the constraints, i.e. the constraints are contradictory

The Simplex Method

The classic method for solving LP problems;
one of the most important algorithms ever invented

Invented by George Dantzig in 1947



Based on the iterative improvement idea:

Generates a sequence of adjacent points of the problem's feasible region with improving values of the objective function until no further improvement is possible

Standard form of LP problem

1. must be a maximization problem
2. all constraints (except the nonnegativity constraints) must be in the form of linear equations
3. all the variables must be required to be nonnegative

Thus, the general linear programming problem in standard form with m constraints and n unknowns ($n \geq m$) is

$$\begin{aligned} &\text{maximize } c_1 x_1 + \ldots + c_n x_n \\ &\text{subject to } a_{i\,1} x_1 + \ldots + a_{in} x_n = b_i, \quad i = 1,\ldots,m, \\ &\qquad x_1 \geq 0, \ldots, x_n \geq 0 \end{aligned}$$

Every LP problem can be represented in such form

Transforming to Standard Form

1. min f (x) max[−f (x)].

2.

$$\begin{array}{l} \text{maximize } 3x + 5y \\ \text{subject to } x + y \le 4 \\ \qquad\qquad x + 3y \le 6 \\ \qquad\qquad x\ge 0, \ \ y\ge 0 \end{array} \quad \longrightarrow \quad \begin{array}{l} \text{maximize } 3x + 5y + 0u + 0v \\ \text{subject to } x + y + u = 4 \\ x + 3y + v = 6 \\ x\ge 0, \ y\ge 0, \ u\ge 0, \ v\ge 0 \end{array}$$

Variables u and v, transforming inequality constraints into equality constrains, are called *slack variables*

3. Unconstrained x_j can be replaced by two non-negative variables as follows $x_j = x' - x''$, $x' \ge 0$, $x'' \ge 0$

Basic feasible solutions

A *basic solution* to a system of m linear equations in n unknowns ($n \geq m$) is obtained by setting $n - m$ variables to 0 and solving the resulting system to get the values of the other m variables. The variables set to 0 are called *nonbasic*; the variables obtained by solving the system are called *basic*.
A basic solution is called *feasible* if all its (basic) variables are nonnegative.

Example
$$\begin{aligned} x + y + u &= 4 \\ x + 3y + v &= 6 \end{aligned}$$

(0, 0, 4, 6) is basic feasible solution
(x, y are nonbasic; u, v are basic)

There is a 1-1 correspondence between extreme points of LP's feasible region and its basic feasible solutions.

Simplex Tableau

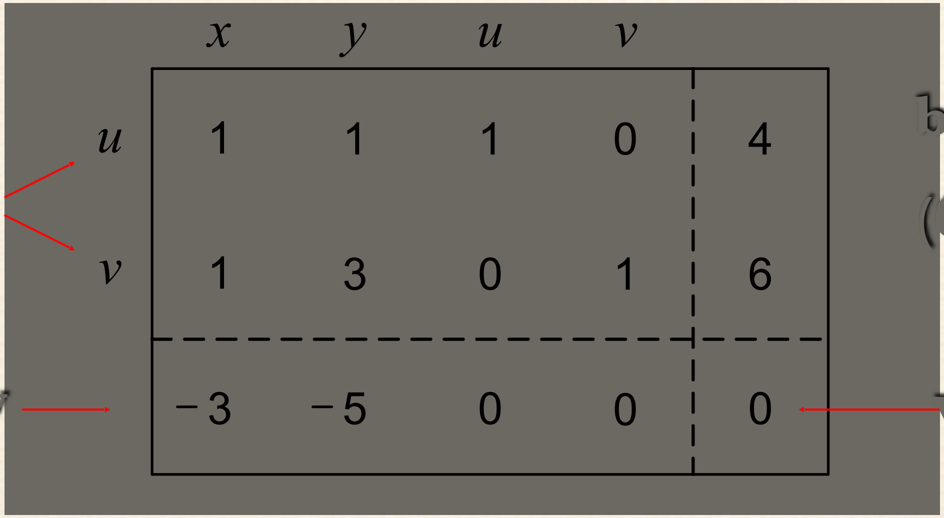
maximize $\quad z = 3x + 5y + 0u + 0v$

subject to $\quad x + y + u \qquad = 4$

$$x\begin{bmatrix}1\\1\end{bmatrix}+y\begin{bmatrix}1\\3\end{bmatrix}+u\begin{bmatrix}1\\0\end{bmatrix}+v\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}4\\6\end{bmatrix}$$

$\qquad x + 3y \qquad + v = 6$

$\qquad x\geq 0,\ y\geq 0,\ u\geq 0,\ v\geq 0$



	x	y	u	v	
u	1	1	1	0	4
v	1	3	0	1	6
	−3	−5	0	0	0

basic variables

objective row

basic feasible solution

(0, 0, 4, 6)

value of *z* at (0, 0, 4, 6)

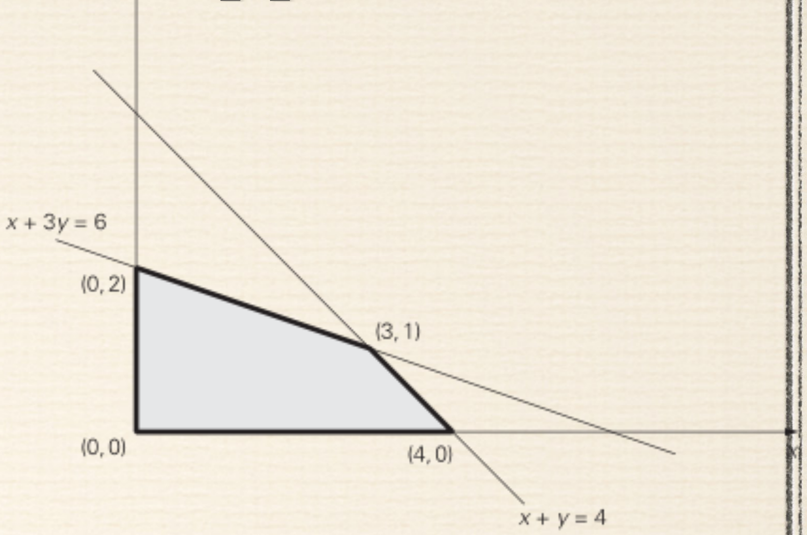
Outline of the Simplex Method

- **Step 0** [Initialization] Present a given LP problem in standard form and set up initial tableau.
- **Step 1** [Optimality test] If all entries in the objective row are nonnegative — stop: the tableau represents an optimal solution.
- **Step 2** [Find entering variable] Select (the most) negative entry in the objective row. Mark its column to indicate the entering variable and the pivot column.

- **Step 3** [Find departing variable] For each positive entry in the pivot column, calculate the θ-ratio by dividing that row's entry in the rightmost column by its entry in the pivot column. (If there are no positive entries in the pivot column — stop: the problem is unbounded.) Find the row with the smallest θ-ratio, mark this row to indicate the departing variable and the pivot row.

- **Step 4** [Form the next tableau] Divide all the entries in the pivot row by its entry in the pivot column. Subtract from each of the other rows, including the objective row, the new pivot row multiplied by the entry in the pivot column of the row in question. Replace the label of the pivot row by the variable's name of the pivot column and go back to Step 1.

Example of Simplex Method Application



maximize $z = 3x + 5y + 0u + 0v$

subject to

$$
\begin{aligned}
x + y + u &= 4\\
x + 3y + v &= 6\\
x\geq 0,\ y\geq 0,\ u\geq 0,\ v&\geq 0
\end{aligned}
$$

$$
\begin{array}{cc|cccc|c|}
 & & x & y & u & v & \\
\hline
 & u & 1 & 1 & 1 & 0 & 4\\
\leftarrow & v & 1 & 3 & 0 & 1 & 6\\
\hline
 & & -3 & -5 & 0 & 0 & 0\\
\hline
 & & & \uparrow & & &
\end{array}
\qquad
\begin{array}{cc|cccc|c|}
 & & x & y & u & v & \\
\hline
\leftarrow & u & \frac{2}{3} & 0 & 1 & -\frac{1}{3} & 2\\
 & y & \frac{1}{3} & 1 & 0 & \frac{1}{3} & 2\\
\hline
 & & -\frac{4}{3} & 0 & 0 & \frac{5}{3} & 10\\
\hline
 & & \uparrow & & & &
\end{array}
\qquad
\begin{array}{c|cccc|c|}
 & x & y & u & v & \\
\hline
x & 1 & 0 & \frac{3}{2} & -\frac{1}{2} & 3\\
y & 0 & 1 & -\frac{1}{2} & \frac{1}{2} & 1\\
\hline
 & 0 & 0 & 2 & 1 & 14\\
\hline
\end{array}
$$

basic feasible sol. (0, 0, 4, 6)

$z = 0$

basic feasible sol. (0, 2, 2, 0)

$z = 10$

basic feasible sol. (3, 1, 0, 0)

$z = 14$

Notes on the Simplex Method

- Finding an initial basic feasible solution may pose a problem
- Theoretical possibility of cycling
- Typical number of iterations is between *m* and 3*m*, where *m* is the number of equality constraints in the standard form
- Worse-case efficiency is exponential
- More recent *interior-point algorithms* such as *Karmarkar's algorithm* (1984) have polynomial worst-case efficiency and have performed competitively with the simplex method in empirical tests

Assignment 4

Can we apply the simplex method to solve the knapsack problem (see Example 2 in Section 6.6)? If you answer yes, indicate whether it is a good algorithm for the problem in question; if you answer no, explain why not.