COT 6405
ANLYSIS OF ALGORITHMS

Linear Programming

Computer & Electrical Engineering and Computer Science Dept.
Florida Atlantic University

Spring 2017

Outline

- Motivation
- Example
- Terminology and definition of an linear program
- Overview of algorithms, solvers
- Integer linear program
- Overview of an linear program
- Standard and slack forms
- Formulating problems as linear programs

Reference: chapter 29.1 & 29.2 in CLRS

Linear Programming, motivation

- Used to solve optimization problems: maximizing or minimizing an objective
- Basic idea:
  - specify the objective of the problem as a linear function of certain variables
  - specify the constraints on resources as equalities or inequalities on those variables

Example: a political problem

- a politician is trying to win an election
- 3 area types: urban (100K voters), suburban (200K voters), rural (50K voters)
- Estimates on the number of votes you could win/lose from each area by spending $1,000 on advertising the following issues:

policy	urban	suburban	rural
build roads	−2	5	3
gun control	8	2	−5
farm subsidies	0	0	10
gasoline tax	10	0	−2

• each entry is the number of thousand of voters that can be won/lost(-) by the advertisement

• Find the minimum amount of money that you need to spend to win 50K urban votes, 100K suburban votes and 25K rural votes

Formalizing the problem

Variables:

x_1- thousands of $ spent on advertising on building roads

x_2 - thousands of $ spent on advertising on gun control

x_3 - thousands of $ spent on advertising on farm subsides

x_4 - thousands of $ spent on advertising on a gasoline tax

Requirements:

at least 50K urban votes:

$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$

at least 100K suburban votes:

$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$

at least 25K rural votes:

$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$

policy	urban	suburban	rural
build roads	−2	5	3
gun control	8	2	−5
farm subsidies	0	0	10
gasoline tax	10	0	−2

Objective:

minimize $x_1 + x_2 + x_3 + x_4$

Linear Programming

minimize $x_1 + x_2 + x_3 + x_4$

subject to:

$$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$$
$$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$$
$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$$
$$x_1, x_2, x_3, x_4 \geq 0$$

The solution of this linear program yields an optimal strategy

Linear Programming

A linear programming problem asks to maximize or minimize a *linear function* (objective function) subject to a finite set of *linear constraints*.

- Given a set of real numbers a_1, …, a_n and a set of variables x_1, x_2, …, x_n, we define a *linear function* as:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n$$

- *Linear constraints* can be either *linear equalities* or *linear inequalities*
  - Linear equality: $f(x_1,x_2,\ldots,x_n) = b$
  - Linear inequality: $f(x_1,x_2,\ldots,x_n) \leq b$ or $f(x_1,x_2,\ldots,x_n) \geq b$

Linear Programming

- Linear programming algorithms:
  - *Simplex algorithm*, developed by George Dantzig in 1947
    - runs fast in practice
    - solves “random” problems in cubic number of steps
    - worst-case RT is exponential in the problem size
  - *Ellipsoid algorithm*, developed by Leonid Khachian in 1979
    - worst-case RT is polynomial in the problem size $O(n^4L)$, a problem which has *n* variables and can be encoded in *L* input bits
  - *Interior point methods*
    - N. Karmarkar, 1984, worst-case RT is $O(n^{3.5}L)$
- Linear programming solver:
  - CPLEX
  - Matlab, optimization package
  - many others

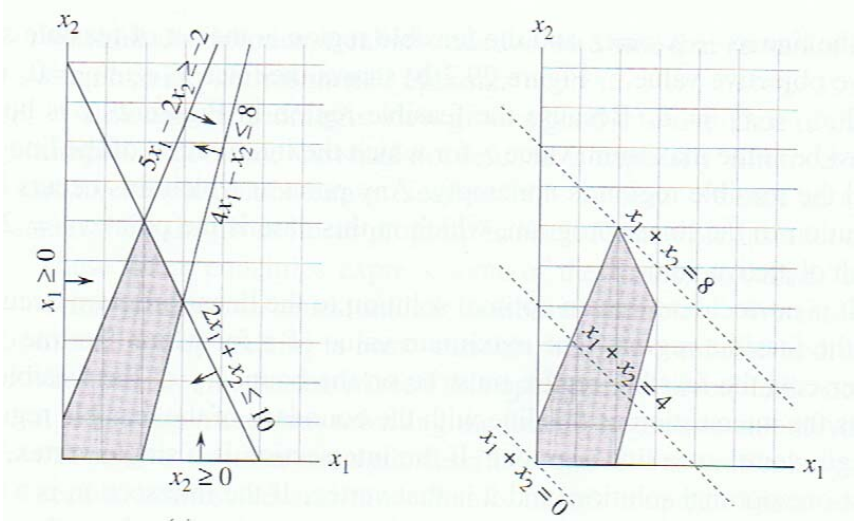
Integer Linear Program

- when the variables take integer values
- finding a feasible solution to an integer linear program is NP-hard
  - no known polynomial-time algorithm

Overview

Linear program with 2 variables:

maximize $x_1 + x_2$
subject to
$4x_1 - x_2 \leq 8$
$2x_1 + x_2 \leq 10$
$5x_1 - 2x_2 \geq -2$
$x_1, x_2 \geq 0$



- *feasible solution*: any solution x_1, x_2 that satisfies all the constraints
- the set of feasible solutions form a convex region, called the *feasible region*
- this example, optimal solution is $x_1 = 2$, $x_2 = 6$

Standard and slack forms

- **Standard form**: all the constraints are inequalities
- **Slack form**: all constraints are equalities (except those that require the variables to be nonnegative)

Standard form

Given:

- n real numbers $c_1, c_2, \ldots, c_n$
- m real numbers $b_1, b_2, \ldots, b_m$
- mn real numbers a_{ij}, for i = 1,…,m, j = 1,…,n

We wish to find n real numbers $x_1, x_2, \ldots x_n$ such that:

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

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \quad \text{for } j = 1, 2, \ldots, n$$

objective function

constraints

nonnegativity constraints

Standard form

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \\ & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

- Express it in a compact form:

$$\begin{aligned} &\text{maximize } c^T x \\ &\text{subject to} \\ &\qquad Ax \leq b \\ &\qquad x \geq 0 \end{aligned}$$

where $A = (a_{ij})$ is an $m \times n$ matrix
$b = (b_i)$ is an m – vector
$c = (c_j)$ is an n – vector
$x = (x_j)$ is an n – vector

- Then, we can specify a linear program in standard form by a tuple (A,b,c)

Converting a LP into standard form

A LP might not be in standard form because:

1. The objective function might be a minimization rather than a maximization
2. There might be variables w/o nonnegativity constraints
3. There might be equality constraints
4. There might be inequality constraints, but instead of $\leq$ sign, they have $\geq$ sign

Equivalent LP

Two maximization LP L and L' are ***equivalent*** if:

- for each feasible solution x to L with objective value z, there is a corresponding feasible solution x' to L' with objective value z, and
- for each feasible solution x' to L' with objective value z, there is a corresponding feasible solution x to L with objective value z

Convert a minimization LP to an equivalent maximization LP

- Negate the coefficients in the objective function
- Example:

$$
\begin{array}{lrcrcl}
\text{minimize} & -2x_1 & + & 3x_2 & & \\
\text{subject to} & & & & & \\
 & x_1 & + & x_2 & = & 7 \\
 & x_1 & - & 2x_2 & \leq & 4 \\
 & x_1 & & & \geq & 0
\end{array}
$$

Convert to an equivalent maximization LP:

$$
\begin{array}{lrcrcl}
\text{maximize} & 2x_1 & - & 3x_2 & & \\
\text{subject to} & & & & & \\
 & x_1 & + & x_2 & = & 7 \\
 & x_1 & - & 2x_2 & \leq & 4 \\
 & x_1 & & & \geq & 0
\end{array}
$$

Issue: some of the variables do not have nonnegativity constraints

- assume a variable x_j does not have a nonnegativity constraint
- solution: replace x_j by $x_j' - x_j''$ and add the nonnegativity constraints $x_j' \geq 0$ and $x_j'' \geq 0$
- example:

$$
\begin{array}{lrcrcl}
\text{maximize} & 2x_1 & - & 3x_2 & & \\
\text{subject to} & & & & & \\
 & x_1 & + & x_2 & = & 7 \\
 & x_1 & - & 2x_2 & \leq & 4 \\
 & x_1 & & & \geq & 0
\end{array}
$$

- issue: x_2 does not have a nonnegativity constraint
- replace $x_2 = x_2' - x_2''$
- convert to an equivalent LP:

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2' & - & x_2'' & = & 7 \\
 & x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
 & & & x_1, x_2', x_2'' & & & \geq & 0
\end{array}
$$

Issue: some LP may have equality constraints

- solution: replace each constraint $f(x_1,\ldots,x_n) = b$ with two constraints $f(x_1,\ldots,x_n) \leq b$ and $f(x_1,\ldots,x_n) \geq b$
- example:

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & = & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& & & x_1, x_2', x_2'' & & & \geq & 0
\end{array}
$$

- convert to an equivalent LP:

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& & & x_1, x_2', x_2'' & & & \geq & 0
\end{array}
$$

Issue: some inequality constraints are ≥

- solution: multiply these inequalities by -1

  inequality $\sum_{j=1}^{n} a_{ij} x_j \geq b_j$ is equivalent to $\sum_{j=1}^{n} -a_{ij} x_j \leq -b_i$

- example:

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
 & x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
 & x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
 & & & x_1, x_2', x_2'' & & & \geq & 0
\end{array}
$$

- convert to an equivalent LP:

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & & & x_1, x_2, x_3 & & & \geq & 0
\end{array}
$$

Converting LP into a slack form

- The only inequalities allowed are the nonnegativity constraints
- All others are equality constraints
- Assuming an inequality: $\sum_{j=1}^{n} a_{ij} x_j \le b_i$

  ➢ introduce a new variable *s* and rewrite the inequality as two constraints:

$$s = b_i - \sum_{j=1}^{n} a_{ij} x_j$$

$$s \ge 0$$

Example

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & & & x_1, x_2, x_3 & & & \geq & 0
\end{array}
$$

- to convert to the slack form, introduce variables x_4, x_5, and x_6

$$
\begin{array}{lrcrcrcrl}
\text{maximize} & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & \\
 & x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
 & x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
 & x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
 & & & x_1, x_2, x_3, x_4, x_5, x_6 & & \geq & 0 & . & &
\end{array}
$$

Slack form, cont.

maximize $2x_1 - 3x_2 + 3x_3$

subject to

$$x_1 + x_2 - x_3 + x_4 = 7$$
$$x_1 + x_2 - x_3 - x_5 = 7$$
$$x_1 - 2x_2 + 2x_3 + x_6 = 4$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

Formulating problems as linear programs

- It is important to recognize when we can formulate a problem as a linear programming
- Once we cast a problem as a LP, it can be solved in polynomial time
- We can use one of the existing software packages to solve it

Maximum Flow

- A flow is a nonnegative real-valued function f : V × V → R that satisfies the capacity constraint and flow conservation
- Linear Program:

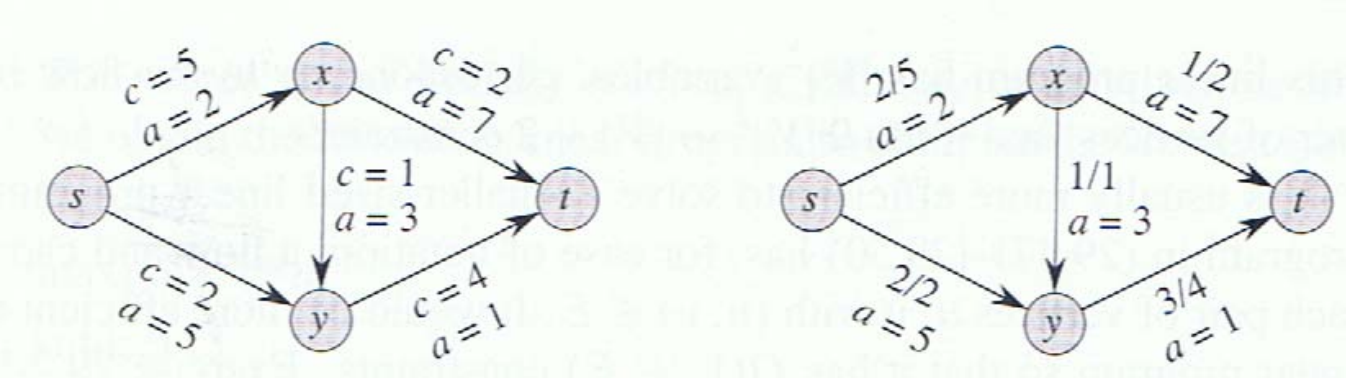
$$
\begin{aligned}
\text{maximize} \quad & \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} \\
\text{subject to} \quad & \\
& f_{uv} \leq c(u,v) && \text{for each } u, v \in V, \\
& \sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv} && \text{for each } u \in V - \{s,t\} \\
& f_{uv} \geq 0 && \text{for each } u, v \in V.
\end{aligned}
$$

- $|V|^2$ variables, f_{uv} for any pair of vertices (u,v)
- $2\,|V|^2 + |V| - 2$ constraints

Minimum-cost-flow problem

- generalization of the maximum-flow problem
- each edge (u,v) has a real-valued cost a(u,v)
- f_{uv} units of flow on (u,v) $\Rightarrow$ cost = $a(u,v)f_{uv}$
- assume c(u,v) = 0 if (u,v) $\notin$ E, and there are no antiparallel edges
- given a flow demand d
- Objective: send d units of flow from s to t while minimizing the total cost $\Sigma_{(u,v)\in E}\, a(u,v)f_{uv}$ incurred by the flow

Example



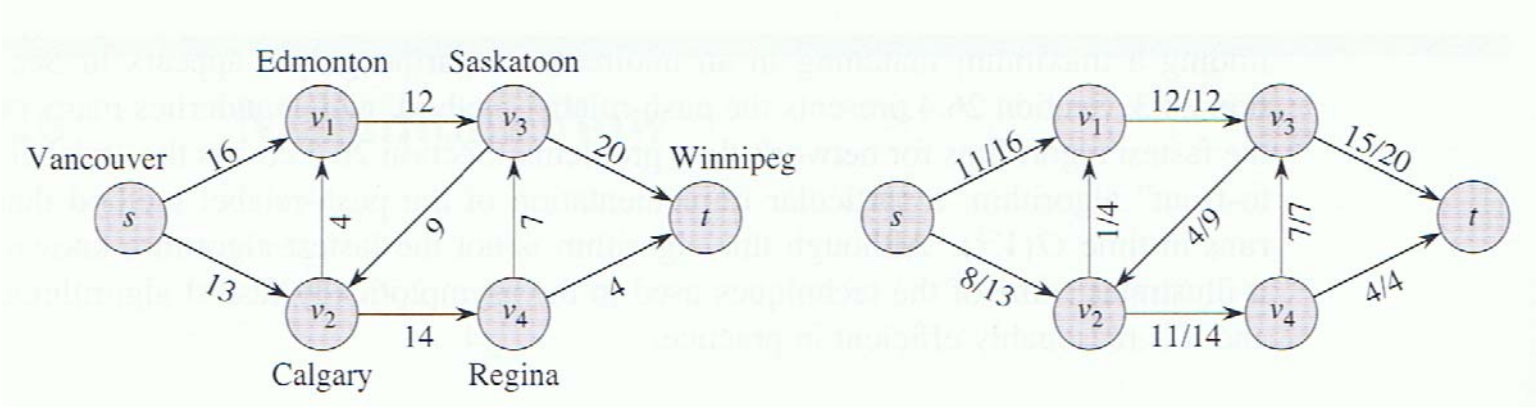
- want to send 4 units of flow s-t
- minimum total cost
- optimal solution has total cost:

  $\Sigma_{(u,v)\in E}\, a(u,v)f_{uv} = 2\cdot 2 + 5\cdot 2 + 3\cdot 1 + 7\cdot 1 + 1\cdot 3 = 27$

Linear Program

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(u,v)\in E} a(u,v)\, f_{uv} \\
\text{subject to} \quad & \\
& f_{uv} \leq c(u,v) && \text{for each } u, v \in V\ , \\
& \sum_{v\in V} f_{vu} - \sum_{v\in V} f_{uv} = 0 && \text{for each } u \in V - \{s,t\} \\
& \sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} = d\ , & & \\
& f_{uv} \geq 0 && \text{for each } u, v \in V\ .
\end{aligned}
$$

Multicommodity flow



- Suppose that LuckyPuck company ships not only hockey pucks, but also hockey sticks and hockey helmets
- Sticks must be sent from Vancouver to Saskatoon and helmets from Edmonton to Regina
- Different items (or commodities) must share the same network
- Each item K_i is specified by the triple $K_i = (s_i, t_i, d_i)$, where s_i is the source, t_i is the destination, and d_i is the demand

Multicommodity flow

Multicommodity-flow problem: Given

- directed graph G = (V, E), in which each edge (u,v) $\in$ E has capacity c(u,v) $\geq$ 0
- k different commodities $K_1, K_2, \ldots, K_k$, each specified by the triple $K_i = (s_i, t_i, d_i)$

Determine if a feasible flow exists that meets the flow-conservation and capacity constraints

Variables: f_{iuv} – flow of commodity i between vertices u and v,

minimize 0

subject to

$$\sum_{i=1}^{k} f_{iuv} \leq c(u,v) \quad \text{for each } u, v \in V,$$

$$\sum_{v \in V} f_{iuv} - \sum_{v \in V} f_{ivu} = 0 \quad \text{for each } i = 1, 2, \ldots, k \text{ and for each } u \in V - \{s_i, t_i\},$$

$$\sum_{v \in V} f_{i,s_i,v} - \sum_{v \in V} f_{i,v,s_i} = d_i \quad \text{for each } i = 1, 2, \ldots, k,$$

$$f_{iuv} \geq 0 \quad \text{for each } u, v \in V \text{ and for each } i = 1, 2, \ldots, k.$$

Problem

Formulate the following problem as an LP:

A carpenter makes tables and chairs. Each table can be sold for a profit of $30 and each chair for a profit of $10. The carpenter can afford to spend up to 40 hours per week working and takes 6 hours to make a table and 3 hours to make a chair. Customer demand requires that he makes at least three times as many chairs as tables. Tables take up four times as much storage space as chairs and there is room for at most four tables each week. The objective is to maximize the profit per week.

Hint: use two variables, X_T – number of tables made per week, and X_C - number of chairs made per week

The knapsack problem

Given:

- n objects and a knapsack
- object i (i = 1,..,n) has a positive weight w_i and a positive value v_i
- the knapsack can carry a weight $\leq W$

Objective: fill the knapsack s.t. to maximize the value of the included objects, while respecting the capacity constraints

Two variations:

- **0-1 knapsack problem**: you can only take the whole object
- **fractional knapsack problem**: you can take fractions of objects

Example: Knapsack capacity W = 16

item	weight	value
1	2	$20
2	5	$30
3	10	$50
4	5	$10

Formulate the Knapsack problem using LP

The Assignment Problem

There are *n* people who need to be assigned to *n* jobs, one person per job. The cost of assigning person *i* to job *j* is C[*i*,*j*]. Find an assignment that minimizes the total cost.

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

Formulate the Assignment problem using LP