Analysis of Algorithms

V. Adamchik

Lecture 10

CSCI 570

University of Southern California

Network Flow

Linear Programming

Reading: chapter 8

Discussion Problem 1

The computer science department course structure is represented as a directed acyclic graph $G = (V, E)$ where the vertices correspond to courses and a directed edge (u, v) exists if and only if the course u is a prerequisite of the course v. By taking a course w, you gain a benefit of p_w which could be a positive or negative number. Note, to take a course, you have to take all its prerequisites. Design an efficient algorithm that picks a subset $S \subset V$ of courses such that the total benefit is maximized.

benefit = $\sum p_w$, where $w \in S$.

goal: max benefit

Discussion Problem 2

CSCI 570 is a large class with n TAs. Each week TAs must hold office hours in the TA office room. There is a set of k hour-long time intervals $I_1, I_2, \ldots I_k$ in which the office room is available. The room can accommodate up to 3 TAs at any time. Each TA provides a subset of the time intervals he or she can hold office hours with the minimum requirement of l_j hour per week, and the maximum m_j hours per week. Lastly, the total number of office hours held during the week must be H. Design an algorithm to determine if there is a valid way to schedule the TA's office hours with respect to these constraints.

Linear Programming

Linear Programming

In this lecture we describe linear programming that is used to express a wide variety of different kinds of problems. LP can solve the max-flow problem and the shortest distance, find optimal strategies in games, and many other things.

We will primarily discuss the setting and how to code up various problems as linear programs.

Solving by Reduction

Formally, to reduce a problem Y to a problem X (we write $Y \leq_p X$) we want a function f that maps Y to X such that:

- f is a polynomial time computable
- $\forall$ *instance* $y \in Y$ is solvable if and only if $f(y) \in X$ is solvable.

A Production Problem

A company wishes to produce two types of souvenirs: type-A will result in a profit of $1.00, and type-B in a profit of $1.20.
To manufacture a type-A souvenir requires 2 minutes on machine I and 1 minute on machine II.
A type-B souvenir requires 1 minute on machine I and 3 minutes on machine II.
There are 3 hours available on machine I and 5 hours available on machine II.
How many souvenirs of each type should the company make in order to maximize its profit?

A Production Problem

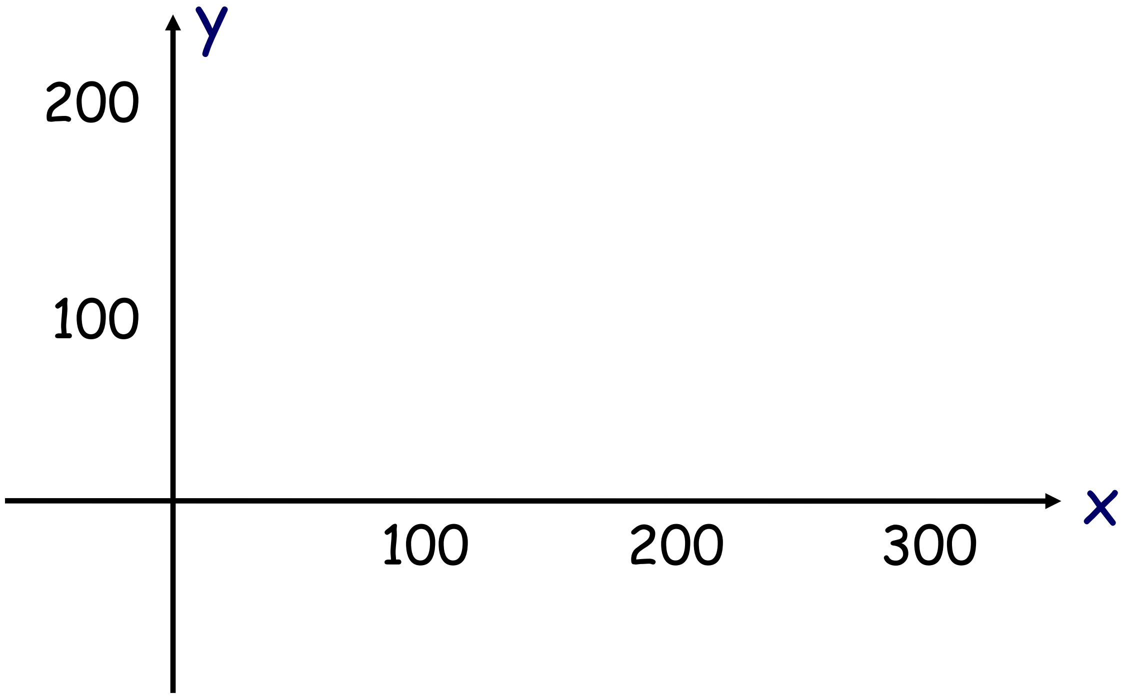
	Type-A	Type-B	Time Available
Profit/Unit	$1.00	$1.20	
Machine I	2 min	1 min	180 min
Machine II	1 min	3 min	300 min

A Linear Program

We want to maximize the objective function

subject to the system of inequalities:

A Production Problem



We need to find the feasible point that is farthest in the "objective" direction

Fundamental Theorem

If a linear programming problem has a solution, then it must occur at a vertex, or corner point, of the feasible set S associated with the problem.

If the objective function P is optimized at *two* adjacent vertices of S, then it is optimized at *every point* on the line segment joining these vertices, in which case there are infinitely many solutions to the problem.

Existence of Solution

Suppose we are given a LP problem with a feasible set S and an objective function P. There are 3 cases to consider

Standard LP form

We say that a maximization linear program with n variables is in standard form if for every variable x_k we have the inequality $x_k \geq 0$ and all other m linear inequalities.

An LP in standard form is written as

$$\max\ (c_1x_1 + \ldots + c_nx_n)$$

subject to

$$a_{11}x_1 + \ldots + a_{1n}x_n \leq b_1$$

$$\vdots$$

$$a_{m1}x_1 + \ldots + a_{mn}x_n \leq b_m$$

$$x_1 \geq 0, \ldots, x_n \geq 0$$

Standard LP in Matrix Form

The vector c is the column vector $(c_1, \ldots, c_n)$.
The vector x is the column vector $(x_1, \ldots, x_n)$.
The matrix A is the $n \times m$ matrix of coefficients of the left-hand sides of the inequalities, and
$b = (b_1, \ldots, b_m)$ is the vector of right-hand sides of the inequalities.

$$\max\ (c^T x)$$

subject to

$$A x \le b$$
$$x \ge 0$$

Exercise: Convert to Matrix Form

$\max(x_1 + 1.2\ x_2)$
$2x_1 + x_2 \leq 180$
$x_1 + 3x_2 \leq 300$
$x_1 \geq 0$
$x_2 \geq 0$

Algorithms for LP

The standard algorithm for solving LPs is the Simplex Algorithm, due to Dantzig, 1947.

This algorithm starts by finding a vertex of the polytope, and then moving to a neighbor with increased cost as long as this is possible. By linearity and convexity, once it gets stuck it has found the optimal solution.

Unfortunately, simplex does not run in polynomial time it does well in practice, but poorly in theory.

Algorithms for LP

In 1974 Khachian has shown that LP could be done in polynomial time by something called the Ellipsoid Algorithm (but it tends to be fairly slow in practice).

In 1984, Karmarkal discovered a faster polynomial-time algorithm called "interior-point". While simplex only moves along the outer faces of the polytope, "interior-point" algorithm moves inside the polytope.

MATLAB

https://www.mathworks.com/help/optim/ug/linprog.html

linprog

Linear programming solver

Finds the minimum of a problem specified by

$$\min_x f^T x \text{ such that } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub. \end{cases}$$

f, *x*, *b*, *beq*, *lb*, and *ub* are vectors, and *A* and *Aeq* are matrices.

Description

`x = linprog(f,A,b)` solves min `f'*x` such that `A*x ≤ b`.

`x = linprog(f,A,b,Aeq,beq)` includes equality constraints `Aeq*x = beq`. Set `A = []` and `b = []` if no inequalities exist.

`x = linprog(f,A,b,Aeq,beq,lb,ub)` defines a set of lower and upper bounds on the design variables, `x`, so that the solution is always in the range `lb ≤ x ≤ ub`. Set `Aeq = []` and `beq = []` if no equalities exist.

Discussion Problem 1

A cargo plane can carry a maximum weight of 100 tons and a maximum volume of 60 cubic meters. There are three materials to be transported, and the cargo company may choose to carry any amount of each, up to the maximum available limits given below.

	Density	Volume	Price
Material 1	2 tons/m^3	40 m^3	\$1,000 per m^3
Material 2	1 tons/m^3	30 m^3	\$2,000 per m^3
Material 3	3 tons/m^3	20 m^3	\$12,000 per m^3

Write a linear program that optimizes revenue within the constraints.

Discussion Problem 2

There are n people and n jobs. You are given a cost matrix, C, where c_{ij} represents the cost of assigning person i to do job j. You need to assign all the jobs to people and also only one job to a person. You also need to minimize the total cost of your assignment. Write a linear program that minimizes the total cost of your assignment.

Discussion Problem 3

Convert the following LP to standard form

$\max\ (5x_1 - 2x_2 + 9x_3)$

$3x_1 + x_2 + 4x_3 = 8$
$2x_1 + 7x_2 - 6x_3 \le 4$
$x_1 \le 0,\ x_3 \ge 1$

Discussion Problem 4

Explain why LP cannot contain constrains in the form of *strong* inequalities.

$\max(7x_1 - x_2 + 5x_3)$

$x_1 + x_2 + 4x_3 < 8$

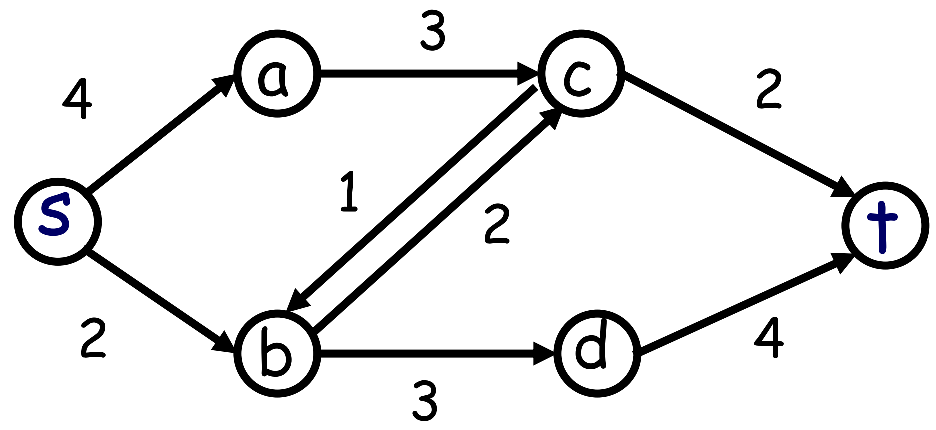
$3x_1 - x_2 + 2x_3 > 3$

$2x_1 + 5x_2 - x_3 \leq -7$

$x_1, x_2, x_3 \geq 0$

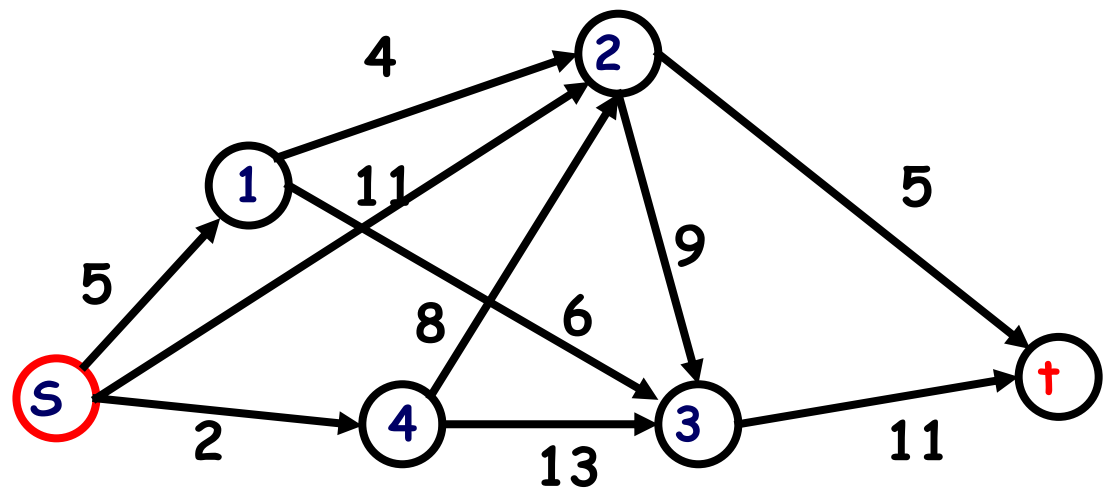
Exercise: Max-Flow as LP

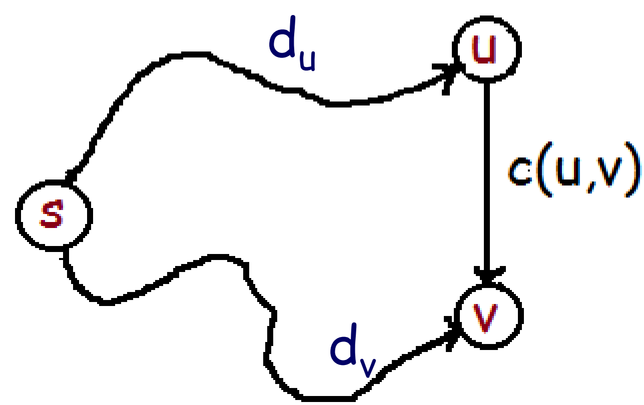
Write a max-flow problem as a linear program.

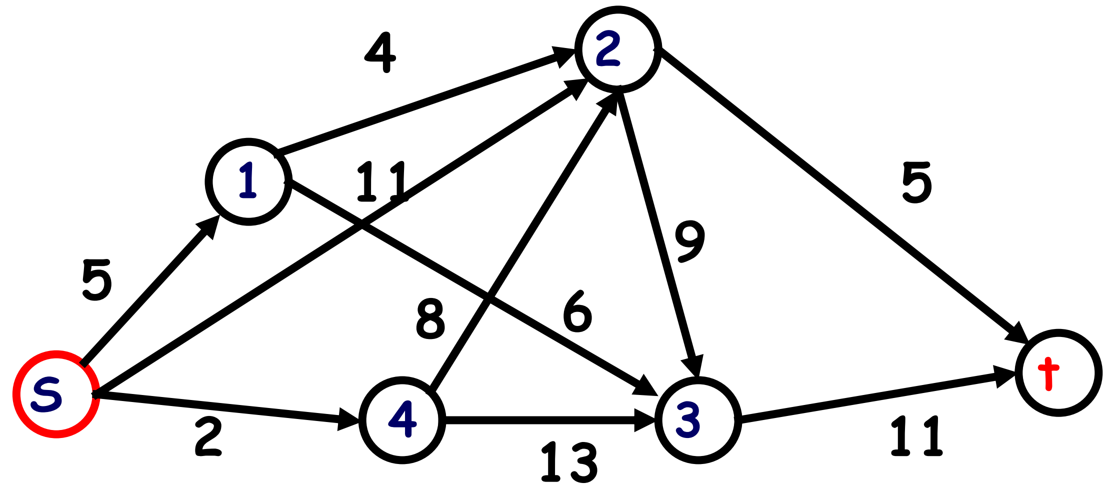


Exercise: Shortest Path as LP

Write a shortest st-path problem as a linear program.







s
1
2
3
4
t
5
4
11
5
9
8
6
2
13
11

Discussion Problem 5

Write a 0-1 Knapsack Problem as a linear program.

Given n items with weights $w_1, w_2, \ldots, w_n$ and values $v_1, v_2, \ldots, v_n$. Put these items in a knapsack of capacity W to get the maximum total value in the knapsack.

$$\text{Given } \sum_{k=1}^{m} w_k \leq W$$

$$\text{optimize } \sum_{k=1}^{m} v_k \to \max$$

Knapsack as LP