Linear Programming

Adapted from CLRS, G. Blelloch, and K. Daniels

Admin

Getting Elected

- Can spend money on advertising any of four critical parts of your platform:

Motivation: A Political Problem

Goal: Win election by winning majority of votes in each region.

	100,000 voters	200,000 voters	50,000 voters
policy	urban	suburban	rural
build roads	−2	5	3
gun control	8	2	−5
farm subsidies	0	0	10
gasoline tax	10	0	−2

Thousands of voters who could be won with $1,000 of ads

Subgoal: Win majority of votes in each region while minimizing advertising cost.

Motivation: A Political Problem
(continued)

- x_1 is the number of thousands of dollars spent on advertising on building roads,
- x_2 is the number of thousands of dollars spent on advertising on gun control,
- x_3 is the number of thousands of dollars spent on advertising on farm subsidies, and
- x_4 is the number of thousands of dollars spent on advertising on a gasoline tax.

$$
\begin{array}{llrcrcrcrcr}
\text{minimize} & & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & & \\
\text{urban} & & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
\text{suburban} & & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
\text{rural} & & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & & x_1, x_2, x_3, x_4 & & & & & \geq & 0 .
\end{array}
$$

Thousands of voters representing majority.

General Linear Programs

Linear function; real numbers; real-valued variables:

$$f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n = \sum_{j=1}^{n} a_jx_j .$$

Linear constraints:

$$f(x_1, x_2, \ldots, x_n) = b$$ Linear equality

$$f(x_1, x_2, \ldots, x_n) \leq b$$

and Linear inequalities

$$f(x_1, x_2, \ldots, x_n) \geq b$$

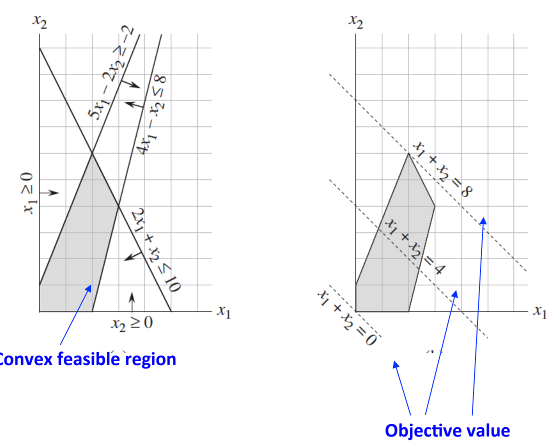
Why LP?

- 50+ software packages available
- 1300+ papers just on interior point methods
- 100+ books in the library
- Dozens of companies
- Delta Airlines claims they save $100 million a year with there optimization application

Overview of Linear Programming

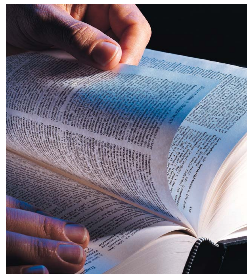
Objective function

$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& & & x_1, x_2 & \geq & 0
\end{array}
$$



Terminology

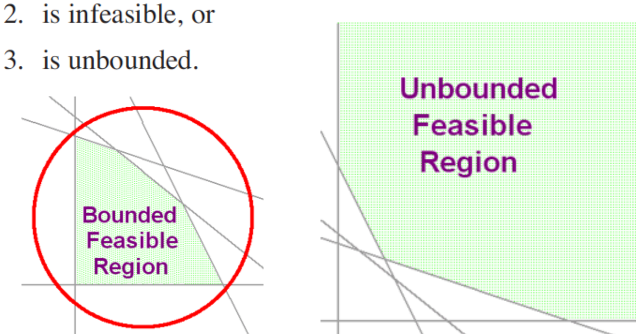
- Feasible solution:
- Infeasible solution:
- Objective value:
- Optimal solution:
- Optimal objective value:
- (In)feasible LP:
- Unbounded:



Fundamental Theorem of LP

Theorem 29.13 (Fundamental theorem of linear programming)
Any linear program L, given in standard form, either

1. has an optimal solution with a finite objective value,
2. is infeasible, or
3. is unbounded.



https://people.richland.edu/james/ictcm/2006/slopeobjective.html

Worksheet

Worksheet

- Give three feasible solutions to the linear program:

$$
\begin{array}{llcccccccccl}
\text{minimize} & & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & & \\
& & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & & \geq & 0 \,.
\end{array}
$$

Standard Form

In ***standard form***, we are given n real numbers $c_1, c_2, \ldots, c_n$; m real numbers $b_1, b_2, \ldots, b_m$; and mn real numbers a_{ij} for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. We wish to find n real numbers $x_1, x_2, \ldots, x_n$ that

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

subject to

$$
\left.\begin{array}{rcll}
\sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for } i = 1, 2, \ldots, m \\
x_j & \geq & 0 & \text{for } j = 1, 2, \ldots, n
\end{array}\right\} \text{constraints}
$$

Cormen et al.

Standard Form (compact)

n-dimensional vectors

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

mxn matrix

m-dimensional vector

Can specify linear program in standard form by (A,b,c).

Worksheet

Worksheet

- Convert the following LP into compact form:

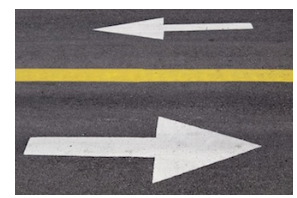
$$\begin{array}{lrcrcrcrcl} \text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\ \text{subject to} & & & & & & & & & \\ & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\ & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\ & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\ & & & x_1, x_2, x_3, x_4 & & & & & \geq & 0 . \end{array}$$

Converting to Standard Form

- Linear programs may not always fit into standard form:
  1. Objective function may be a minimization rather than a maximization
  2. There might be variables without nonnegativity constraints
  3. There might be equality constraints, rather than less-than-or-equal-to
  4. There might be inequality constraints that are greater-than-or-equal-to

Converting to Standard Form

(continued)

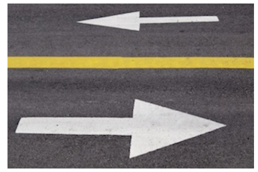


Transforming minimization to maximization

Negate coefficients

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

Converting to Standard Form
(continued)



Giving each variable a non-negativity constraint

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

x_2 has no non-negativity constraint

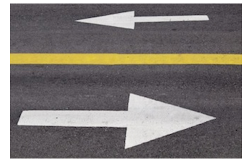
$$\begin{array}{lrcrcrcl} \text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2' & - & x_2'' & = & 7 \\ & x_1 & - & 2x_2' & + & 2x_2'' & \le & 4 \\ & x_1, x_2', x_2'' & & & & & \ge & 0 \end{array}$$

$- 3(x_2' - x_2'')$

New non-negativity constraints

If x_j has no non-negativity constraint,
replace each occurrence of x_j with $x_j' - x_j''$.

Converting to Standard Form
(continued)

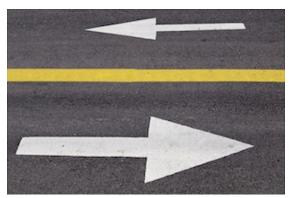


Transforming equality constraints to inequality constraints

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

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

Converting to Standard Form
(continued)



Changing sense of an inequality constraint

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

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

(Rename variables for notational consistency.)

Rationale: $\sum_{j=1}^{n} a_{ij}x_j \ge b_i$ is equivalent to $\sum_{j=1}^{n} -a_{ij}x_j \le -b_i$

Converting Linear Programs into Slack Form

for algorithmic ease, transform all constraints except non-negativity ones into equalities

for inequality constraint: $\sum_{j=1}^{n} a_{ij}x_j \le b_i$

define *slack*: $s = b_i - \sum_{j=1}^{n} a_{ij}x_j$, $s \ge 0$. (s is the slack variable)

instead of s → $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j$

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

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

basic variables: x_4, x_5, x_6

non-basic variables: x_1, x_2, x_3

Converting Linear Programs into Slack Form (continued)

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

objective function →

$$\begin{array}{rcl} z &=& 2x_1 - 3x_2 + 3x_3 \\ x_4 &=& 7 - x_1 - x_2 + x_3 \\ x_5 &=& -7 + x_1 + x_2 - x_3 \\ x_6 &=& 4 - x_1 + 2x_2 - 2x_3 \end{array}$$

Converting Linear Programs into Slack Form (continued)

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

$$z = v + \sum_{j \in N} c_j x_j$$

$$x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B$$

set of indices of basic variable (B)

set of indices of non-basic variables (N)

Compact Form: (N, B, A, b, c, v)

Slack Form Example → Compact Form

$$\begin{array}{rcl} z &=& 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\ x_1 &=& 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\ x_2 &=& 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\ x_4 &=& 18 - \frac{x_3}{2} + \frac{x_5}{2} , \end{array}$$

we have $B = \{1, 2, 4\}$, $N = \{3, 5, 6\}$,

$$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix} ,$$

negative of slack form coefficients

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix} ,$$

$c = (c_3 \;\; c_5 \;\; c_6)^T = (-1/6 \;\; -1/6 \;\; -2/3)^T$, and $v = 28$.

Applications

- Selecting a mix: Oil mixtures: portfolio selection, …
- Distribution: How much of a commodity should distribute to different locations
- Allocation: How much of a resource should we allocate to difference tasks
- **Network flows**!

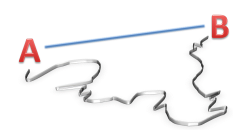
Shortest Paths

What problem is this?

Single-pair shortest path: minimize "distance" from source s to sink t.

$$\begin{array}{ll} \text{maximize} & d_t \\ \text{subject to} & \\ & d_v \leq d_u + w(u, v) \quad \text{for each edge } (u, v) \in E , \\ & d_s = 0 . \end{array}$$

Why don't we want minimize?

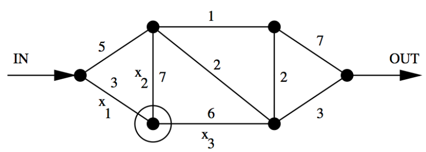


Maximum Flow

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

Capacity constraints

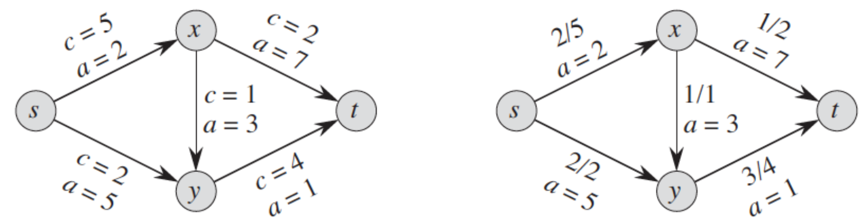
Flow conservation



Minimum Cost Flow

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

Flow target is prespecified.



Multicommodity Flow

$$\begin{aligned}\text{minimize} \quad & 0 \\ \text{subject to} \quad & \\ & \sum_{i=1}^{k} f_i(u,v) \le c(u,v) \quad \text{for each } u,v \in V\,, \\ & f_i(u,v) = -f_i(v,u) \quad \text{for each } i = 1,2,\ldots,k \text{ and for each } u,v \in V\,, \\ & \sum_{v\in V} f_i(u,v) = 0 \quad \text{for each } i = 1,2,\ldots,k \text{ and for each } u \in V - \{s_i,t_i\}\,, \\ & \sum_{v\in V} f_i(s,v) = d_i \quad \text{for each } i = 1,2,\ldots,k\,.\end{aligned}$$

should be s_i

$$f(u,v) = \sum_{i=1}^{k} f_i(u,v)$$

Worksheet: Write the LP!

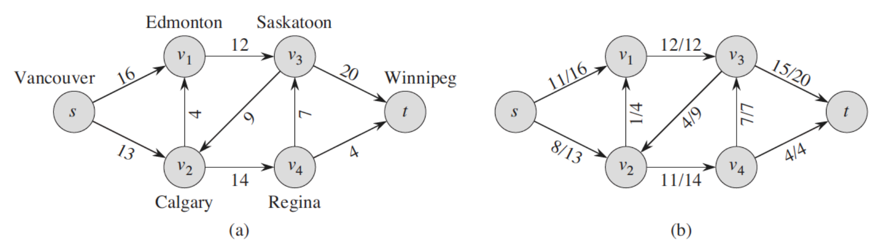
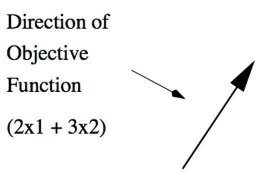
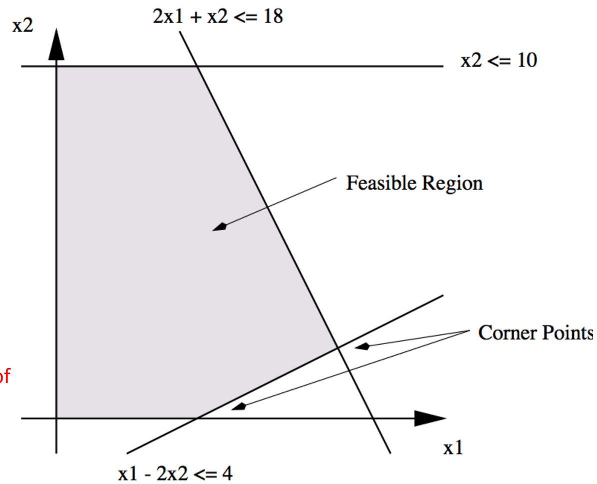


Figure 26.1 **(a)** A flow network $G = (V, E)$ for the Lucky Puck Company's trucking problem. The Vancouver factory is the source s, and the Winnipeg warehouse is the sink t. The company ships pucks through intermediate cities, but only $c(u,v)$ crates per day can go from city u to city v. Each edge is labeled with its capacity. **(b)** A flow f in G with value $|f| = 19$. Each edge (u,v) is labeled by $f(u,v)/c(u,v)$. The slash notation merely separates the flow and capacity; it does not indicate division.

Geometric Interpretation

maximize $z = 2x_1 + 3x_2$
subject to $x_1 - 2x_2 \le 4$
$2x_1 + x_2 \le 18$
$x_2 \le 10$
$x_1, x_2 \ge 0$

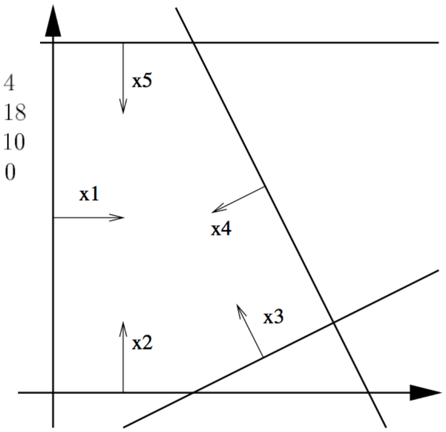




Where is the maximal point?
What are possible properties of the feasible region?

Geometric Interpretation: slack

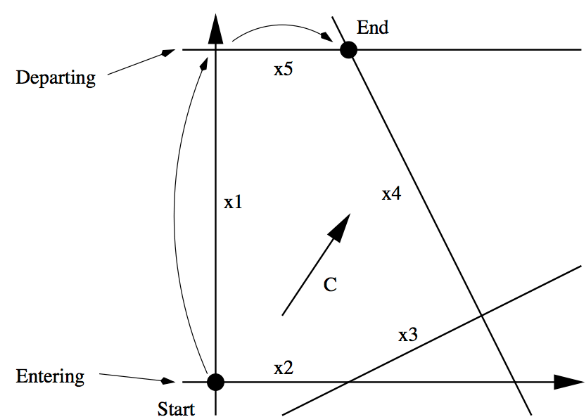
maximize $z = 2x_1 + 3x_2$
subject to $x_1 - 2x_2 + x_3 = 4$
$2x_1 + x_2 + x_4 = 18$
$x_2 + x_5 = 10$
$x_1, x_2, x_3, x_4, x_5 \ge 0$



Simplex Algorithm

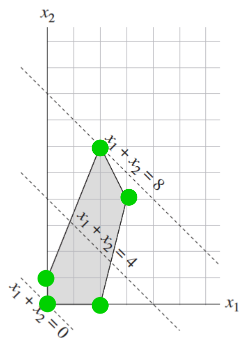
1. Find any corner of the feasible region (if one exists)
2. Corner exists at the intersection of n hyperplanes. For each plane, calculate the dot product of the objective function cost vector *c* with the unit vector normal to the plan (facing inward).
3. If all dot products are negative, then DONE (problem is maximized)
4. Else, select plane with max dot product.
5. Intersection of remaining n-1 hyperplanes forms a line. Move along this line until next corner is reached.
6. Goto 2.

Geometric View of Simplex



Solving a Linear Program

- Simplex algorithm
  - Geometric interpretation
    - Visit vertices on the boundary of the simplex representing the convex feasible region
  - Transforms set of inequalities using process similar t Gaussian elimination (Ch. 28)
  - Run-time
    - not polynomial in worst-case
    - often *<u>very</u>* fast in practice
- Ellipsoid method
  - Run-time
    - polynomial
    - slow in practice
- Interior-Point methods
  - Run-time
    - polynomial
    - for large inputs, performance can be competitive with simplex method
  - Moves through interior of feasible region



Simplex Algorithm: Example
Basic Solution

Standard Form

$$\begin{array}{lrcrcrcl} \text{maximize} & 3x_1 & + & x_2 & + & 2x_3 & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\ & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\ & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\ & & & x_1, x_2, x_3 & & & \geq & 0 \end{array}$$

Slack Form

$$\begin{array}{rcrcrcrcr} z & = & & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Basic Solution: $(\overline{x_1}, \overline{x_2} \ldots \overline{x_6}) = (0,0,0,30,24,36)$

Basic Solution: set each nonbasic variable to 0.

Simplex Algorithm: Example
Reformulating the LP Model

<u>Main Idea</u>: In each iteration, reformulate the LP model so basic solution has larger objective value

Select a nonbasic variable whose objective coefficient is positive: x_1

Increase its value as much as possible.

Identify tightest constraint on increase.

For basic variable x_6 of that constraint, swap role with x_1.

Rewrite other equations with x_6 on RHS.

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}$$

$$\begin{aligned} x_4 &= 30 - x_1 - x_2 - 3x_3 \\ &= 30 - \left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) - x_2 - 3x_3 \\ &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4}. \end{aligned}$$

entering variable (x_1); leaving variable (x_6)

$$\begin{array}{rcrcrcrcr} z & = & & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

PIVOT →

new objective value (27)

$$\begin{array}{rcrcrcrcr} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

Simplex Algorithm: Example
Reformulating the LP Model

Next Iteration: select x_3 as entering variable.

entering variable (x_3); leaving variable (x_5)

$$\begin{array}{rcrcrcrcr} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

PIVOT →

new objective value ($\frac{111}{4}$)

$$\begin{array}{rcrcrcrcr} z & = & \frac{111}{4} & + & \frac{x_2}{16} & - & \frac{x_5}{8} & - & \frac{11x_6}{16} \\ x_1 & = & \frac{33}{4} & - & \frac{x_2}{16} & + & \frac{x_5}{8} & - & \frac{5x_6}{16} \\ x_3 & = & \frac{3}{2} & - & \frac{3x_2}{8} & - & \frac{x_5}{4} & + & \frac{x_6}{8} \\ x_4 & = & \frac{69}{4} & + & \frac{3x_2}{16} & + & \frac{5x_5}{8} & - & \frac{x_6}{16}. \end{array}$$

New Basic Solution: $(\overline{x_1}, \overline{x_2} \ldots \overline{x_6}) = (33/4, 0, 3/2, 69/4, 0, 0)$

Simplex Algorithm: Example

Reformulating the LP Model

Next Iteration: select x_2 as entering variable.

entering variable

$$
\begin{aligned}
z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\
x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} .
\end{aligned}
$$

leaving variable

PIVOT →

new objective value

$$
\begin{aligned}
z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2} .
\end{aligned}
$$

New Basic Solution: $(\bar{x}_1, \bar{x}_2 \ldots \bar{x}_6) = (8,4,0,18,0,0)$

Simplex Algorithm: Pivoting

leaving variable → l; entering variable → e

PIVOT(N, B, A, b, c, v, l, e)

Rewrite the equation that has x_l on LHS to have x_e on LHS

// Compute the coefficients of the equation for new basic variable x_e.
let $\hat{A}$ be a new $m \times n$ matrix
$\hat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
  $\hat{a}_{ej} = a_{lj} / a_{le}$
$\hat{a}_{el} = 1/a_{le}$

Update remaining equations by substituting RHS of new equation for each occurrence of x_e.

// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
  $\hat{b}_i = b_i - a_{ie}\hat{b}_e$
  for each $j \in N - \{e\}$
    $\hat{a}_{ij} = a_{ij} - a_{ie}\hat{a}_{ej}$
  $\hat{a}_{il} = -a_{ie}\hat{a}_{el}$

Do the same for objective function.

// Compute the objective function.
$\hat{v} = v + c_e\hat{b}_e$
for each $j \in N - \{e\}$
  $\hat{c}_j = c_j - c_e\hat{a}_{ej}$
$\hat{c}_l = -c_e\hat{a}_{el}$

Update sets of nonbasic, basic variables.

// Compute new sets of basic and nonbasic variables.
$\hat{N} = N - \{e\} \cup \{l\}$
$\hat{B} = B - \{l\} \cup \{e\}$
return $(\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v})$

Simplex Algorithm: Pseudocode

SIMPLEX(A, b, c)

$(N, B, A, b, c, v) =$ INITIALIZE-SIMPLEX(A, b, c) ← **initial basic solution**
let Δ be a new vector of length n
while some index $j \in N$ has $c_j > 0$
  choose an index $e \in N$ for which $c_e > 0$
  for each index $i \in B$
    if $a_{ie} > 0$
      $\Delta_i = b_i / a_{ie}$
    else $\Delta_i = \infty$
  choose an index $l \in B$ that minimizes Δ_i
  if $\Delta_l == \infty$
    return "unbounded" ← **detects unboundedness**
  else $(N, B, A, b, c, v) =$ PIVOT(N, B, A, b, c, v, l, e)
for $i = 1$ **to** n
  if $i \in B$
    $\bar{x}_i = b_i$
  else $\bar{x}_i = 0$
return $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ ← **optimal solution**

Correctness: Roadmap (Key Pieces)

Lemma 29.3: Algebraic lemma

Lemma 29.4: Slack form uniqueness

Lemma 29.8: Weak LP duality

Corollary 29.9: Conditions for which feasible solutions for primal, dual programs are optimal

Lemma 29.1: Pivot results

Lemma 29.6: Tie-breaking

Lemma 29.5: $\binom{n+m}{m}$ iteration bound, cycling

Lemma 29.11: L_{aux}

Lemma 29.2: Basic solution feasible -> if SIMPLEX finds solution it is feasible; if reports unbounded, then model is unbounded

Lemma 29.7: Basic solution feasible -> SIMPLEX either reports unbounded or finds feasible solution in $\binom{n+m}{m}$ iterations

Lemma 29.12: Infeasibility detection

Theorem 29.10: LP duality: SIMPLEX primal result is optimal & dual is optimal

Theorem 29.13: Fundamental Theorem of Linear Programming
For LP model in standard form, either:
1. exists optimal solution with finite objective function value & SIMPLEX returns one, or
2. infeasible & SIMPLEX returns INFEASIBLE, or
3. Unbounded & SIMPLEX returns UNBOUNDED

Simplex: Further reading

- See 29.5 to answer important remaining questions:
  - How do we determine whether a linear program is feasible?
  - What do we do if the initial basic solution is not feasible?
  - How do we determine whether a linear program is unbounded?
  - How do we choose the entering and leaving variables?
  - **How do we know it's the optimal value?**

Linear Programming Duality

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

subject to

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

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

max becomes min

x variables go away

sense changes

minimize $\sum_{i=1}^{m} b_i y_i$

y variables appear

subject to

$$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad \text{for } j = 1, 2, \ldots, n ,$$

$$y_i \geq 0 \quad \text{for } i = 1, 2, \ldots, m .$$

Duality Example

maximize $3x_1 + x_2 + 2x_3$

subject to

$$\begin{aligned} x_1 + x_2 + 3x_3 &\leq 30 \\ 2x_1 + 2x_2 + 5x_3 &\leq 24 \\ 4x_1 + x_2 + 2x_3 &\leq 36 \\ x_1, x_2, x_3 &\geq 0 . \end{aligned}$$

minimize $30y_1 + 24y_2 + 36y_3$

subject to

$$\begin{aligned} y_1 + 2y_2 + 4y_3 &\geq 3 \\ y_1 + 2y_2 + y_3 &\geq 1 \\ 3y_1 + 5y_2 + 2y_3 &\geq 2 \\ y_1, y_2, y_3 &\geq 0 . \end{aligned}$$

Optimality

Theorem 29.10 (Linear-programming duality)

Suppose that SIMPLEX returns values $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ for the primal linear program (A, b, c). Let N and B denote the nonbasic and basic variables for the final slack form, let c' denote the coefficients in the final slack form, and let $\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m)$ be defined by equation (29.91). Then $\bar{x}$ is an optimal solution to the primal linear program, $\bar{y}$ is an optimal solution to the dual linear program, and

$$\sum_{j=1}^{n} c_j \bar{x}_j = \sum_{i=1}^{m} b_i \bar{y}_i . \quad (29.92)$$

Benchmarks

Name	Simplex (Primal)	Simplex (Dual)	Barrier + Crossover
binpacking	29.5	62.8	560.6
distribution	18,568.0	won't run	12,495,464
forestry	1,354.2	1,911.4	2,348.0
maintenance	57,916.3	89,890.9	3,240.8
crew	7,182.6	16,172.2	1,264.2
airfleet	71,292.5	108,015.0	37,627.3
energy	3,091.1	1,943.8	858.0
4color	45,870.2	won't run	44,899,242

Table 10: Running Times of Large Models in seconds.

Worksheet: Communication Network Problem

We have a network whose lines have the bandwidth shown in Figure 4. We wish to establish three calls: One between A and B (call 1), one between B and C (call 2), and one between A and C (call 3). We must give each call at least 2 units of bandwidth, but possibly more. The link from A to B pays 3 per unit of bandwidth, from B to C pays 2, and from A to C pays 4. Notice that each call can be routed in two ways (the long and the short path), or by a combination (for example, two units of bandwidth via the short route, and three via the long route). How do we route these calls to maximize the network's income?

This is also a linear program. We have variables for each call and each path (long or short); for example x_1 is the short path for call 1, and x_2' the long path for call 2. We demand that (1) no edge bandwidth is exceeded, and (2) each call gets a bandwidth of 2.

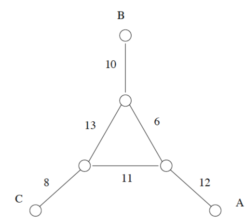


Figure 4: A communication network

Source: M. Jordan

Worksheet: Communication Network Problem

$$\max 3x_1 + 3x_1' + 2x_2 + 2x_2' + 4x_3 + 4x_3'$$

$$\begin{array}{rcl}
x_1 + x_1' + x_2 + x_2' & \leq & 10 \\
x_1 + x_1' + x_3 + x_3' & \leq & 12 \\
x_2 + x_2' + x_3 + x_3' & \leq & 8 \\
x_1 + x_2' + x_3' & \leq & 6 \\
x_1' + x_2 + x_3' & \leq & 13 \\
x_1' + x_2' + x_3 & \leq & 11 \\
x_1 + x_1' & \geq & 2 \\
x_2 + x_2' & \geq & 2 \\
x_3 + x_3' & \geq & 2 \\
x_1, x_1' \ldots, x_3' & \geq & 0
\end{array}$$

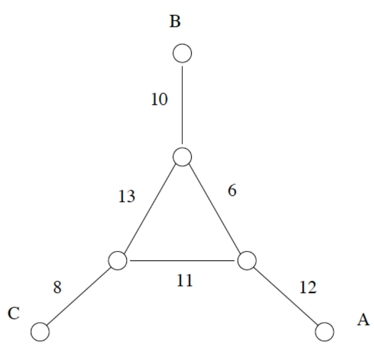


Figure 4: A communication network

The solution, obtained via simplex in a few milliseconds, is the following: $x_1 = 0, x_1' = 7, x_2 = x_2' = 1.5, x_3 = .5, x_3' = 4.5$.