Overview LP

1. Examples: politics, flow, shortest paths
2. General form and converting to it
3. Simplex algorithm
   - Iterating over slack form

1/ Politics

How to campaign to win an election? -> estimate vote obtained per dollar spent advertising in support of a particular issue.

Policy	Urban	Suburban	Rural
Build roads [x1]	-2	5	3
Gun Control [x2]	8	2	-5
Farm Subsidies [x3]	0	0	10
Gasoline Tax [x4]	10	0	2

want a win by spending the min. amount of money.

Policy	Urban	Suburban	Rural
population	100,000	200,000	50,000
majority	50,000	100,000	25,000

Algebraic Setup

Let x1, x2, x3, x4 denote $ spent per issue.
Minimize x1+x2+x3+x4
subject to

1. $-2x1 + 8x2 + 0x3 + 10x4 >= 50,000$
2. $5x1 + 2x2 + 0x3 + 0x4 >= 100,000$
3. $3x1 - 5x2 + 10x3 + 2x4 >= 25,000$

and x1, x2, x3, x4 >= 0

(n variables) (n constraints)

optimum:
x1 = 2050 000/111
x2 = 425 000/111
x3 = 0
x4 = 625 000/111

x1+x2+x3+x4 = 3,100,000/111
xi. are real numbers

2/ standard form for LP

-> minimize or maximize linear objective fn subject to linear inequaloties (or equations)

variable : <x> = <x1 x2 x3 .. xn>
objective fn : <c>.<x> = c1x1 + ... + cnxn
inequalities : A.<x> <= <b>
max. <c>.<x> s.t. A.<x> <= <b> and <x> >= 0

3/ Certificate of Optimality (but why ?)

Is there a short certificate that shows LP solu. is optimal?
=> Consider 25/222*(1) + 36/222*(2) + 14/222*(3) [equations]
=> x1 + x2 + 140/222*x3 + x4 >= 3100000/111*
*SINCE, x1+x2+x3+x4 >= x1+x2+140/222*x3+x4 >= 3100000/111

4/ LP Duality

Theorem:
** Equivalent **

primal form	dual form
max <c>.<x>	min <b>.<y>
s.t. A.<x> <= <b>	s.t. A^T.<y> >= <c>
<x> >= 0	<y> >= 0

5/ Converting to Standard Form

1. Minimize -2x1 + 3x2, negate to 2x1 - 3x2, maximize
2. Suppose xj does NOT have a non-negative constraint, xj replace with xj' - xj'', xj' >= 0 xj'' >= 0
3. equality constraint x1 + x2 = 7, x1 + x2 <= 7, -x1 - x2 <= -7
4. >= constraint translates to <= by (-1) multiply

6/ Maximum Flow

max SUM[v in V]{f(s,v) = |f|}

1. Skew symmetry: f(u, v) = -f(v,u) [for all u,v in V]
2. Conservation : SUM[v in V]{f(u,v) = 0} [for all u in V - {s, t}]
3. Capactiy : f(u, v) <= c(u,v) [for all u,v in V]

Two Commodities (1 & 2)
f1, c1, f2, c2 (two distinct disjoint opts)
f1, f2, single capacity c, w1*f1(u, v) + w2*f2(u, v) <= c(u, v)

7/ Shortest Path

From Vertex S \

[triangular inequality] \ d[v] - d[u] <= w(u,v) [for all (u,v) in E] \

d[s] = 0
w(u1, v) & w(u2, v)
u1 -> v <- u2

d[v] - d[u1] <= w(u1, v)
d[v] - d[u2] <= w(u2, v)
d[v] = minimum { ... }

objective: max SUM[V]{d[v]}
** change minimize to maximize ** (becoz all constraints serve as & operator)

8/ Simplex Algorithm | worst case (m+n Choose m)

Flow: Represent LP in slack form
convert one slack form into an equivalent slack form
whose objective value has not decreased, and has likely
increased. Keep going till the optimal solution become
obvious.

Example

Maximize 3x1 + x2 + x3
s.t.
x1 + x2 + 3x3 <= 30
2x1 + 2x2 + 5x3 <= 24
4x1 + x2 + 2x3 <= 36
x1, x2, x3 >= 0

slackform

z = 3x1 + x2 + x3 [non-basic variables]
[basic variables][equation I]
x4 = 30-x1-x2-3x3
x5 = 24-2x1-2x2-2x3
x6 = 36-4x1-x2-2x3 \

x1, x2, x3, x4, x5, x6

Basic Solution

set all non-basic vairbales to 0
calculate values of baisc variables
obj fn: 3(0) + 1(0) + 2(0) = 0
<0, 0, 0, 30, 24, 36> \

Pivoting

1. Select a non-bacic x_e whose coefficient in the obj. fn is positive
2. Increase the value of x_e as much as posible without volating any of the constraints
3. Variable x_e become basic
4. x_e becomes basic and other variable becomes non-basic. (value of other basic variabl)

Iteration

select x1 nonbasic, increase the value of x1
3rd constraint is tightest one:
x1 = 9 - x2/4 - x3/2 - x6/4
Rewrite the other equations with x6 on the R.H.S.
i.e. replace x1 with above equation \

z = 27 + x2/4 +x3/2 - 3x6/4
x1 = 9 - x2/4 - x3/2 - x6/4
x4 = 21 - 3x2/4 -5x3/2 + x6/4
x5 = 6 - 3x2/2 - 4x3 + x1/2

original basic solution: <0, 0, 0, 30, 24, 36>
satisfies [equation II] and has
objective value: 27 + 1/4 * 0 + 1/2 * 0 - 3/4 * 36 = 0 \

basic solution II: set nonbasic values on 0
< 9, 0, 0, 21, 6, 0 >
objective value: 9*3 = 7