LP and FFT

EECS 477

Lecture 19, 11/19/2002

A Linear Programming Example

- Politician trying to win a election
- 3 Types of areas: urban, suburban, rural
- Primary issues:
  build roads, gun control, farm subsidies
- To win, need:
  50,000 urban votes
  100,000 suburban votes
  25,000 rural votes.

Money Spent on Campaign Ads

Policy	Urban	Suburban	Rural
Build Roads	-2	5	3
Gun Control	8	2	-5
Farm Subsidies	0	0	10
Votes Req.	50	100	25

Shown: 1000s of votes won by spending $1000 an ads

- Goal: spend min $$ and <u>win the elections</u>

Formalization

- Introduce variables:

  x_1 = $$ spent on building roads

  x_2 = $$ spent on gun control

  x_3 = $$ spent on farm subsidies

- Express constraints through equations:

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

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

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

Finally…

- Minimize the function:

  $x_1 + x_2 + x_3$
- Subject to constraints

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

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

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

  and $x_1, x_2, x_3 \geq 0$
- (Any) solution of this linear program is an optimal strategy for the politician

Formal Definitions

- Linear Function:
  $f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n$
  where $x_1 \ldots x_n$ are variables and $a_1 \ldots a_n$ are constants.
- Linear Equality:
  $f(x_1, \ldots, x_n) = b$ where b is a real number.
- Linear Inequalities:
  $f(x_1, \ldots, x_n) \leq b$ and
  $f(x_1, \ldots, x_n) \geq b$

More Definitions…

- Linear Programming Problem:
  - Minimizing or maximizing linear function
    - E.g., $x_1 + x_2 + x_3$ (Objective Function)
  - Subject to a finite set of linear constraints (=,≥,≤)
- Three cases
  - Max *f(…)*
  - Min *f(…)*: can turn into Max –*f(…)*
  - (Pure) constraint satisfaction (no objective function)
    - E.g., $f(x)=0\,x$
- Standard Form: max subject to linear inequalities
- Slack Form: max subject to linear equalities

More Definitions (2)

- Feasible Solution
  - Values $x_1, x_2 \ldots x_n$ that satisfy all constraints
- Feasible Region (in the *n*-dim space)
  - Graph all the constraint equations
  - Their intersection is the feasible region
- Objective Value: value of the objective function at a particular point in the feasible region.

Standard Form

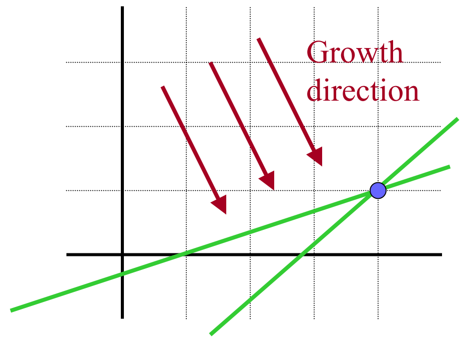
- n real numbers $c_1 \ldots c_n$;
  m real numbers $b_1, b_2, \ldots b_m$; and mn real numbers a_{ij} for $i = 1 \ldots m$ and $j = 1 \ldots n$.
- Find $x_1 \ldots x_n$ that
  maximize $\sum c_j x_j$, $j = 1 \ldots n$ (objective function)
  subject to
  $\sum a_{ij} x_j \leq b_i$ for $i = 1 \ldots m$ and $j = 1 \ldots n$ (Eq1)
  $x_j \geq 0$ for $j = 1 \ldots n$ (Eq 2)
- Eq 1 and Eq 2 are constraints
- Eq 2 : non negativity constraints

Summary of Standard Form

- a_{ij} = elements of m x n matrix A
- b_i = m-dim vector b
- c_j = n-dim vector c
- x_i = n-dim vector x
- Linear Program is a tuple (A,b,c), interpreted as
  - Max $c^T x$
  - Subject to $Ax \leq b$
    $x \geq 0$

Example

- max ($x_1 - 2 x_2$)
- $x_1 >= 0$, $x_2 >= 0$
- $x_1 - 3 x_2 <= 1$
- $x_1 - x_2 <= 3$



- Draw a picture
- Find the solution via simplex method

Scheduling and LP

- m=1..M machines
- j=1..J jobs
- p[m,j] – processing time

  x[m,j] = 0 or 1 -- assignment

  $\Sigma_{m=1..M}$ x[m,j] = 1,   j=1..J

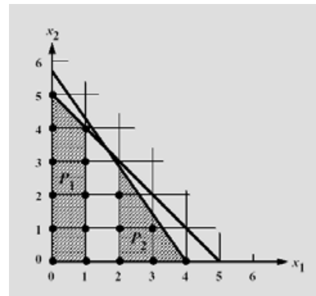
  $\Sigma_{j=1..J}$ x[m,j] p[m,j] <= t,   m=1..M

  min t

  LP Relaxation x[m,j]>=0

Integer programming

- Maximize 17x+12y
- Subject to
    - 10x+7y <= 40
    - x+y<=5
    - x,y>=0, integers
  - from Vanderbei ‘2001
    - Optimal solution (x,y)=(1.67,3.33)
    - Rounding (2,3) infeasible, closest feasible (1,3)
    - Can use Branch and Bound for exact solution



DFT

- w := exp(-2πi/N)
  - N-th roots of unity $(w^k)^N = 1$
  - Complex number exp(iu) = cos(u) + i sin(u)
  - Taking squares get n/2 roots of unity
  - Fourier transform: what applications

  $a^{\wedge}[k] = \Sigma_{j=0..N-1}$ a[j] w^{kj}, for all k=0..N-1
- Simple algorithm

  N times N-term sums = $\Theta(N^2)$

FFT

- Use the fact that

$\Sigma_{j=0..N-1}$ a[j] x^j = $\Sigma_{j=0..N/2-1}$ a[2*j] x^{2j} + x $\Sigma_{j=0..N/2-1}$ a[2*j+1] x^{2j}

- Recursive procedure

```
fft(a[0..n-1]) {
    ye = fft((a[0],a[2],..,a[n-2]));
    yo = fft((a[1],a[3],..,a[n-1]));
    for(k=0; k<n/2; ++k) {
        y[k] = ye[k] + w^k yo[k];
        y[k+n/2] = ye[k] - w^k yo[k];
    }
    return y;
}
```

FFT

- T(N) = 2 T(N/2) + Θ(N)
  – Case ? of the Master Theorem
- Fast algorithm

  T(N) = Θ(N log N)

[0,1,2,3,4,5,6,7,8]

[0,2,4,6] [1,3,5,7]

[0,4] [2,6] [1,5] [3,7]

[0] [4] [2] [6] [1] [5] [3] [7]