Linear Programming

Mohammad Tehranipoor
ECE Department

Linear Programming (LP)

- What is LP?
  - A Linear Program (LP) is a problem that can be expressed as follows (the so-called Standard Form):

    minimize $cx$ // Objective Function
    subject to $Ax = b$ // Constraints
    $x >= 0$
### Linear Programming (LP)

- \( \mathbf{x} \) is the vector of variables to be solved for
- \( \mathbf{A} \) is a matrix of known coefficients
- \( \mathbf{c} \) and \( \mathbf{b} \) are vectors of known coefficients
- The expression "\( \mathbf{cx} \)" is called the objective function
- The equations "\( \mathbf{Ax=b} \)" are called the constraints
- The matrix \( \mathbf{A} \) is generally not square, hence we can’t solve an LP by just inverting \( \mathbf{A} \).

- LP tools (softwares) can handle both minimization and maximization problems.

---

### The Importance of Linear Programming

- The importance of linear programming derives in part from:
  - Its many applications
    - e.g. VLSI Physical Design
  - The existence of good general-purpose techniques (softwares) for finding optimal solutions.
    - CPLEX, LPSolve, AMPL, etc.
    - These techniques take as input only an LP in the above Standard Form, and determine a solution without reference to any information concerning the LP's origins or special structure.
    - They are fast and reliable over a substantial range of problem sizes and applications.
Integer Linear Programming (ILP)

- **Integer programming** (or integer linear programming) requires some or all of the variables to take integer (whole number) values.

- **Advantage**: Integer programs (IPs) often have the advantage of being more realistic than LPs.
  - It provides exact result
  - It is deterministic

- **Disadvantage**: IPs are much harder to solve.

- The most widely used general-purpose techniques (softwares) for solving IPs use the solutions to a series of LPs to manage the search for integer solutions and to prove optimality.

---

ILP vs. LP

- Variables are constructed to take integer values.
- ILP is very much harder problem than ordinary LP.
- In both theory and practice, ILP is more realistic.
- Many VLSI physical design problems are ILP problems, e.g. Floorplanning.
Tools/Softwares

- **Lpsolve**
  - http://www.geocities.com/lpsolve/
- **AMPL**
  - http://www.AMPL.com/
- **CPLEX**
- There is a lot more ...

---

Example

- Example from AMPL book:
  - Product A and B
  - Tons per hour:
    - A: 200
    - B: 140
  - Profit per ton:
    - A: $25
    - B: $30
Cont.

Maximum tons:
- A: 6000
- B: 4000

Question:
- For 40 hours of production time, how many tons of A and how many tons of B should be produced to bring in the maximum profit?

Answer:
Variables:
- A or $X_A$ (tons of A?)
- B or $X_B$ (tons of B?)

The total hours to produce all these tons is given by:

$\frac{1}{200} \cdot X_A + \frac{1}{140} \cdot X_B \leq 40$

$0 \leq X_A \leq 6000$
$0 \leq X_B \leq 4000$

Profit: $(\text{Profit per ton of A}) \cdot X_A + (\text{Profit per ton of B}) \cdot X_B = 25X_A + 30X_B$
Linear Program:

Maximize $25X_A + 30X_B$

Subject to:

$\frac{1}{200}X_A + \frac{1}{140}X_B \leq 40$

$0 \leq X_A \leq 6000$

$0 \leq X_B \leq 4000$
Constraint and Search Space

\[ \frac{1}{200} X_A + \frac{1}{140} X_B = 40 \]

\[ (1/200)X_A + (1/140)X_B = 40 \]

\[ (1/200)X_A + (1/140)X_B = 40 \]

\[ 25X_A + 30X_B = 177500 \]

\[ 25X_A + 30X_B = 177500 \]

\[ 25X_A + 30X_B = 192000 \]

\[ 25X_A + 30X_B = 188000 \]

\[ 25X_A + 30X_B = 188000 \]

\[ (2300, 4000) \]

\[ (6000, 1400) \]

\[ X_A \]

\[ X_B \]

Linear Program in AMPL

```
Var XA
Var XB
Maximize Profit: 25*XA + 30*XB;
Subject to Time: (1/200)*XA + (1/140)*XB <= 40;
Subject to A Limit: 0 =< XA <= 6000;
Subject to B Limit: 0 =< XB <= 4000;
```

Store it in p1.mod
Then type these commands:

```
ampl: model p1.mod
ampl: solve
```

2 iterations, objective: 192000

```
ampl: display XA and XB
XA = 6000
XB = 1400
```

```
ampl: quit
```

---

### Example Problems

- **Job Scheduling**
  - Try the example shown in previous lecture
- **Floorplanning**
- **Scheduling chips under test to control power**
- **Scheduling design with various delays**
- **Scan cell reordering**
Non-Linear Programming (NLP)

minimize $F(x)$
subject to:

\[ g_i(x) = 0 \text{ for } i = 1, ..., m_1 \text{ where } m_1 \geq 0 \]
\[ h_j(x) \geq 0 \text{ for } j = 1, ..., m \text{ where } m \geq m_1 \]

- Genetic and Simulated Annealing Algorithms are categorized in NLP.