Convex Optimization II

Lecture 1: Linear Programming

Hamed Shah-Mansouri

Department of Electrical Engineering

Sharif University of Technology

1400-2

Outline

- Preliminaries
- Linear Programming
- Problem Types and Equivalent Form
- Application 1: Multicommodity Flow Problem
- Application 2: Lifetime Maximization Problem in Wireless Sensor Networks
- Basic Properties
- Summary

Acknowledgement: Vincent Wong and Stephen Boyd. Some materials and graphs are from Boyd and Vandenberghe.

Preliminaries and History

- Programming: Used traditionally to describe the process of operations planning and resource allocation.
- In 1940s, it was realized that planning process could be aided by solving optimization problems involving linear objective and constraints.
- Initial impetus, in the aftermath of World War II, within the context of military planning problems.
- In 1947, Dantzig proposed simplex method to solve linear programming (LP) problems.
- Early work goes back to Fourier, who in 1824 developed an algorithm for solving systems of linear inequalities.
- In late 1930s, Kantorovich worked on problems on resource allocation and developed LP formulations. He also provided a solution method, but his work was not widely known at that time.
- Others included Koopmans, who shared a Nobel Prize in economic science with Kantorovich in 1975.

Linear Programming (LP)

- Minimize a linear objective function of a variable $\mathbf{x} \in \mathbf{R}^n$ over linear inequality and equality constraints

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b} \end{array}$$

  The problem data are vectors $\mathbf{c} \in \mathbf{R}^n$, $\mathbf{h} \in \mathbf{R}^m$, $\mathbf{b} \in \mathbf{R}^p$, as well as matrices $G \in \mathbf{R}^{m \times n}$ and $A \in \mathbf{R}^{p \times n}$.

- Standard form LP

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \succeq \mathbf{0} \end{array}$$

- Computationally more convenient representation.
- Can solve dense problems with thousand of variables and ten thousand constraints.

Equivalent Problems

- Informally, two problems are equivalent if the solution of one problem is readily obtained from the solution of the other, and vice versa.
- Given a feasible solution of one problem, we can construct a feasible solution to the other, with the same cost.
- In particular, the two problems have the same optimal cost and given an optimal solution to one problem, we can construct an optimal solution to the other.

Transformation To Standard Form

- Elimination of inequality constraints: Introduce slack variables s_i for inequality constraints.

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & G\mathbf{x} + \mathbf{s} = \mathbf{h} \\ & A\mathbf{x} = \mathbf{b} \\ & \mathbf{s} \succeq \mathbf{0} \end{aligned}$$

- Elimination of free variables: Express $\mathbf{x}$ as difference between two nonnegative vectors $\mathbf{x}^+, \mathbf{x}^- \succeq \mathbf{0}$. That is, $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$.

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x}^+ - \mathbf{c}^T\mathbf{x}^- \\ \text{subject to} \quad & G\mathbf{x}^+ - G\mathbf{x}^- + \mathbf{s} = \mathbf{h} \\ & A\mathbf{x}^+ - A\mathbf{x}^- = \mathbf{b} \\ & \mathbf{x}^+, \mathbf{x}^-, \mathbf{s} \succeq \mathbf{0} \end{aligned}$$

- Now in LP standard form with variables $\mathbf{x}^+$, $\mathbf{x}^-$, and $\mathbf{s}$.

Piecewise Linear Function

- A function of the form $\max_{i=1,\dots,m}(\mathbf{c}_i^T\mathbf{x} + d_i)$ is called a piecewise linear function.

  Example: absolute value function $f(x) = |x| = \max\{x, -x\}$

- Consider a generalization of LP, where the objective function is piecewise linear rather than linear

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{i=1,\dots,m}(\mathbf{c}_i^T\mathbf{x} + d_i) \\ \text{subject to} & A\mathbf{x} \succeq \mathbf{b} \end{array}$$

- Note that $\max_{i=1,\dots,m}(\mathbf{c}_i^T\mathbf{x} + d_i)$ is equal to the smallest number z that satisfies $z \geq \mathbf{c}_i^T\mathbf{x} + d_i$ for all i.

- The above problem can be transformed as

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & z \geq \mathbf{c}_i^T\mathbf{x} + d_i, \quad i = 1, \dots, m \\ & A\mathbf{x} \succeq \mathbf{b} \end{array}$$

  where the variables are scalar z and vector $\mathbf{x}$.

Norm Minimization Problems

- l_1 norm: $||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$
- Minimize $||A\mathbf{x} - \mathbf{b}||_1$ is equivalent to the following LP in $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{s} \in \mathbf{R}^p$

$$
\begin{array}{ll}
\text{minimize} & \mathbf{1}^T\mathbf{s} \\
\text{subject to} & A\mathbf{x} - \mathbf{b} \preceq \mathbf{s} \\
& A\mathbf{x} - \mathbf{b} \succeq -\mathbf{s}
\end{array}
$$

- l_∞ norm: $||\mathbf{x}||_\infty = \max_i\{|x_i|\}$
- Minimize $||A\mathbf{x} - \mathbf{b}||_\infty$ is equivalent to the following LP in $\mathbf{x} \in \mathbf{R}^n$, $t \in \mathbf{R}$

$$
\begin{array}{ll}
\text{minimize} & t \\
\text{subject to} & A\mathbf{x} - \mathbf{b} \preceq t \\
& A\mathbf{x} - \mathbf{b} \succeq -t
\end{array}
$$

Linear Fractional Programming

- Minimize the ratio of linear functions

$$\begin{aligned} \text{minimize} \quad & \frac{\mathbf{c}^T\mathbf{x} + d}{\mathbf{e}^T\mathbf{x} + f} \\ \text{subject to} \quad & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b} \end{aligned}$$

- Domain of the objective function: $\{\mathbf{x} \mid \mathbf{e}^T\mathbf{x} + f > 0\}$
- Not an LP. If the feasible set is non-empty, we can transform it into an equivalent LP with variables $\mathbf{y} = \frac{\mathbf{x}}{\mathbf{e}^T\mathbf{x}+f}$ and $z = \frac{1}{\mathbf{e}^T\mathbf{x}+f}$

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{y} + dz \\ \text{subject to} \quad & G\mathbf{y} - \mathbf{h}z \preceq \mathbf{0} \\ & A\mathbf{y} - \mathbf{b}z = \mathbf{0} \\ & \mathbf{e}^T\mathbf{y} + fz = 1 \\ & z \geq 0 \end{aligned}$$

- See [pp. 151, Boyd and Vandenberghe] for the proof of equivalence.

Application 1: Multicommodity Flow Problem

- Consider a communication network with N nodes.
- Nodes are connected by communication links.
- A link from node i to node j is an ordered pair (i, j).
- Let $\mathcal{A}$ be the set of all links.
- Each link $(i, j) \in \mathcal{A}$ can carry up to u_{ij} bits per second.
- Positive charge c_{ij} per bit transmitted along link (i, j).
- Each source node k generates data, at the rate of b^{kl} bits per second, that have to be transmitted to destination node l.
- Problem: Choose paths along which all data reach their intended destinations, while minimizing the total cost.
- We allow data with the same origin/source and destination to be split and be transmitted along different paths.

Multicommodity Flow Problem (cont.)

- Let variables x_{ij}^{kl} denote the amount of data with source k and destination l that traverse link (i, j).
- b_i^{kl} is the net flow at node i, of data with source k and destination l

$$b_i^{kl} = \begin{cases} b^{kl}, & \text{if } i = k, \\ -b^{kl}, & \text{if } i = l, \\ 0, & \text{otherwise}. \end{cases}$$

- We have the following LP formulation

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{(i,j)\in\mathcal{A}} \sum_{k=1}^{N} \sum_{l=1}^{N} c_{ij} x_{ij}^{kl} \\ \text{subject to} & \displaystyle\sum_{\{j \mid (i,j)\in\mathcal{A}\}} x_{ij}^{kl} - \sum_{\{j \mid (j,i)\in\mathcal{A}\}} x_{ji}^{kl} = b_i^{kl}, \quad i, k, l = 1, \ldots N, \\ & \displaystyle\sum_{k=1}^{N} \sum_{l=1}^{N} x_{ij}^{kl} \le u_{ij}, \qquad (i, j) \in \mathcal{A} \\ & x_{ij}^{kl} \ge 0, \qquad (i, j) \in \mathcal{A},\ k, l = 1, \ldots N. \end{array}$$

Multicommodity Flow Problem (cont.)

- The first constraint is a flow conservation constraint at node i for data with source k and destination l.

- The summation below represents the amount of data with source and destination k and l, respectively, that leave node i along some link.

$$\sum_{\{j \mid (i,j)\in\mathcal{A}\}} x_{ij}^{kl}$$

- The summation below represents the amount of data with source and destination k and l, respectively, that enter node i through some link.

$$\sum_{\{j \mid (j,i)\in\mathcal{A}\}} x_{ji}^{kl}$$

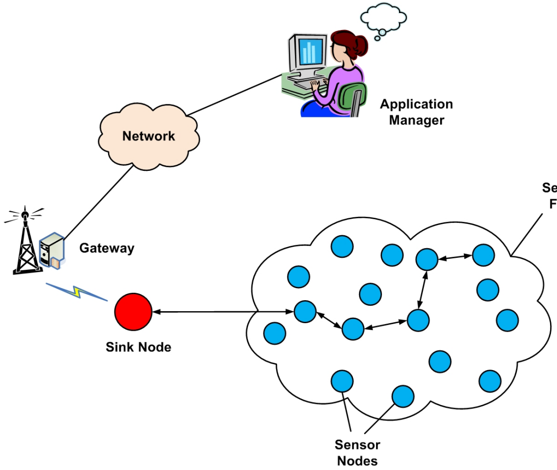
- The constraint below expresses the requirement that the total traffic through a link (i, j) cannot exceed the link's capacity.

$$\sum_{k=1}^{N}\sum_{l=1}^{N} x_{ij}^{kl} \leq u_{ij}, \quad (i,j)\in\mathcal{A}$$

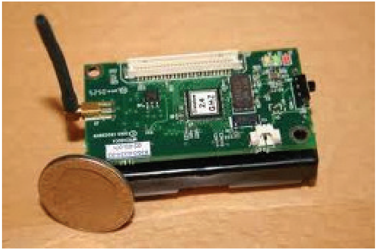
Application 2: Wireless Sensor Networks

A wireless sensor network consists of

- many wireless sensor nodes
- one/multiple sink nodes



Application 2: Wireless Sensor Networks



Application 2: Wireless Sensor Networks

- R. Madan and S. Lall, "Distributed Algorithms for Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE Trans. on Wireless Comm.*, Aug. 2006.
- Problem: Compute an optimal routing scheme that maximizes the time at which the first node in the sensor network drains out of energy.
- Consider a wireless sensor network with the set of nodes V and set of links L.
- Sensor nodes are connected by communication links.
- Each link $(i, j) \in L$ can carry up to R_{ij} bits per second.
- Each sensor node $i \in V$ has an initial battery energy B_i.
- Let S_i be the rate at which information is generated at node i; this information needs to be communicated to the sink node.
- We write $S_{\text{sink}} = -\sum_{i \in V,\ i \neq \text{sink}} S_i$.
- Energy spent by node i to transmit a unit of information directly to node j is E_{ij}.
- Variables: Let r_{ij} denote the rate of information flow from node i to node j.

Wireless Sensor Networks (Cont.)

- The lifetime of node i under flow $\mathbf{r} = \{r_{ij}\}$ is given by

$$T_i(\mathbf{r}) = \frac{B_i}{\sum_{j \in N_i} E_{ij} r_{ij}},$$

where N_i is the set of nodes connected to node i by a link.

- The network lifetime $T_{\text{net}}(\mathbf{r})$ under flow $\mathbf{r}$ is defined as the time when the first sensor node runs out of energy, i.e.,

$$T_{\text{net}}(\mathbf{r}) = \min_{i \in V} \; T_i(\mathbf{r}).$$

- We have the following lifetime maximization problem

$$\begin{array}{ll} \text{maximize} & T_{\text{net}}(\mathbf{r}) \\ \text{subject to} & \displaystyle\sum_{j \in N_i} (r_{ij} - r_{ji}) = S_i, \quad i \in V \\ & 0 \le r_{ij} \le R_{ij}, \qquad\quad i \in V,\ j \in N_i \end{array}$$

which can be transformed as an LP.

OTHER APPLICATIONS

- A. Demiriz, K.P. Bennett, J. Shawe-Taylor, “Linear programming boosting via column generation", *Machine Learning*. Vol. 46, no.1, pp.225–254, Jan 2002.
- C.V. Rao and J.B. Rawlings, “Linear programming and model predictive control", *Journal of Process Control*. vol. 10, no. 1, pp.283–289, Apr 2000.
- K. Li and X. Wang, “Cross-Layer Design of Wireless Mesh Networks with Network Coding,” *IEEE Trans. on Mobile Computing*, Nov. 2008.
  - Network code construction scheme based on LP.

BASIC PROPERTIES

- **Definition:** $\mathbf{x}$ in polyhedron P is an extreme point if there does not exist two other points $\mathbf{y}, \mathbf{z} \in P$ such that $\mathbf{x} = \theta\mathbf{y} + (1-\theta)\mathbf{z}$ for some $\theta \in (0,1)$.

Theorem

Assume that an LP in standard form is feasible and the optimal value is finite. There exists an optimal solution which is an extreme point.

ALGORITHMS

- Simplex method
  - Very efficient in practice but specialized for LP.
  - Move from one vertex to another without enumerating all the vertices.
- Cutting-plane method
- Ellipsoid method
- Interior-point method
  - Commonly used to solve convex optimization problems as well.
- The complexity in practice is of order $n^2 m$ (assuming $m \geq n$).

Solving an LP Using MATLAB

Example:

$$\begin{aligned} \text{minimize} \quad & -5x_1 - 4x_2 - 6x_3 \\ \text{subject to} \quad & x_1 - x_2 + x_3 \le 20, \\ & 3x_1 + 2x_2 + 4x_3 \le 42 \\ & 3x_1 + 2x_2 \le 30 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

In Matlab, we have

$$f = [-5; -4; -6];$$

$$\begin{aligned} A = \quad & [1 \quad -1 \quad 1 \\ & \;\, 3 \quad\; 2 \quad\;\; 4 \\ & \;\, 3 \quad\; 2 \quad\;\; 0]; \end{aligned}$$

$$b = [20; 42; 30];$$

$$lb = zeros(3,1);$$

$$[x, fval, exitflag, output, lambda] = linprog(f, A, b, [], [], lb);$$

- http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/linprog.html

Solving an LP Using CVX

Using CVX, the same problem can be solved as follows:

```
n = 3;
cvx_begin
    variable x(n);
    minimize(f' * x);
    subject to
        A * x <= b;
        x >= lb;
cvx_end
```

- http://cvxr.com/cvx/

Summary

- Linear programming (LP) covers a wide range of interesting problems in different areas.

- There are very useful special structures in LP. But most of the important ones (computational efficiency, global optimality, Lagrange duality) can be generalized to convex optimization.

- Reading: Chapter 1 and Section 4.3 of Boyd and Vandenberghe.

Appendix

Mathematical Background

Norm

A norm is a measure of the *length* of a vector $\mathbf{x}$.

A function $f : \mathbf{R}^n \to \mathbf{R}$ with $\mathbf{dom}\, f = \mathbf{R}^n$ is called a norm if

- f is nonnegative: $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbf{R}^n$.
- f is finite and $f(\mathbf{x}) = 0$ only if $\mathbf{x} = \mathbf{0}$.
- f is homogeneous: $f(t\mathbf{x}) = |t| f(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{R}^n$ and $t \in \mathbf{R}$.
- f satisfies the triangle inequality: $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$.

Examples

- l_p-*norm* is given by $||\mathbf{x}||_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$, with $p \geq 1$.
- l_∞-*norm* is given by $||\mathbf{x}||_\infty = \max\{|x_1|, \ldots, |x_n|\}$.

  Question: Obtain l_∞-*norm* from the definition of l_p-*norm*, when $p \to \infty$.