Advanced Topics in Networking - Fall 2006

Instructor Iordanis Koutsopoulos

Lecture 1 : Introduction - 3/10/06

Notes by : George Noutsis and George Xatziparaskevas

1 Outline of lecture:

- Introduction: Communication systems
- What is an optimization problem
    - Least Squares
    - Linear Programming
    - Convex optimization problems

2 Basics

Two ways we may view a telecommunication system are:

- Network Layer view
- Link Layer view

2.1 Network layer view

Session packets need to be transferred from sources to destinations. Each session has a source and a destination

- if the number of destinations is one, uni-cast session

- if destinations are several, multi-cast session

The problem is how we can transfer data under some optimization criteria.

Example: Given a source and a destination for each session, how can I route session packets in order to maximize network lifetime?

Network lifetime can be defined in many ways, for example it can be the time until the first node dies, i.e. its buttery finishes.

Dual Problem: Give the network lifetime l, what is the maximum number of sessions and the maximum rate that can be supported?

2.2 The mechanisms of OSI layers

The OSI layers are:

Application
Transport
Network
MAC
Physical

Mechanisms:

- Application layer: HTTP. Security is an important parameter.
- Transport layer: Flow control.
- Network layer: Routing.
- MAC link layer: Random access - protocol, channel allocation (TDMA, FDMA).
- Physical layer: Transmission power control, transmission rate control (modulation, coding).

2.3 Physical layer view (one link)

We examine a link between a source and a destination and not the entire network.

Procedures that take place before transmission:

- Source coding (e.g Quantization)

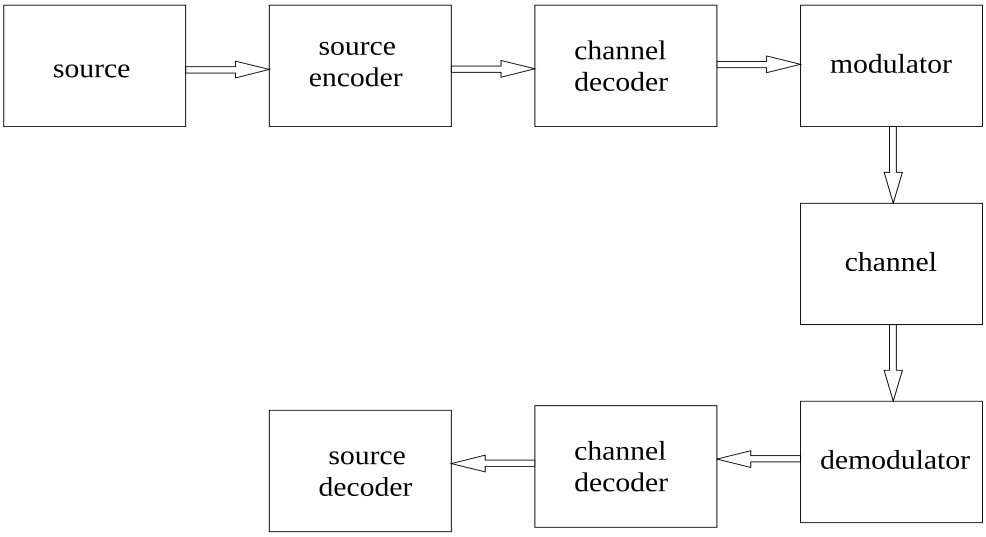


Figure 1: Transmission and reception.

- Channel coding. Some redundant bits are added to data bits. These bits can be a linear combination of the data bits (if the code is linear). Thus, if some data bits are lost due to channel errors, we can retrieve them by using the redundant bits.
- Modulator: The signal is transformed into a continuous waveform.

3 Optimization

An optimization problem has the general form:

$$\text{minimize } f_0(\mathbf{x}) \tag{1}$$

subject to:

$$f_i(\mathbf{x}) \leq b_i, \text{ for } i = 1, 2, \ldots, m \tag{2}$$

Call the problem above (P). Vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the vector of *optimization variables* of the problem. Function $f_0 : \mathcal{R}^n \longrightarrow \mathcal{R}$ is the *objective function.* Functions $f_i : \mathcal{R}^n \longrightarrow \mathcal{R}, i = 1, 2, \ldots, m$, are called *constraint functions*, and constants $b_1, \ldots, b_m$ are the limits, or bounds for the constraints.

A vector $\mathbf{x}_0$ is *feasible* or a *feasible solution* for problem (P) if its satisfies all constraints, i.e. if $f_i(\mathbf{x}_0) \leq b_i, i = 1, \ldots, m$.

A vector $\mathbf{x}^*$ is called *optimal*, or *an optimal solution* of the problem above, if it is feasible and it has the smallest objective function value among all feasible vectors. That is, for any $\mathbf{z}$ with $f_1(\mathbf{z}) \leq b_1, f_2(\mathbf{z}) \leq b_2, \ldots, f_m(\mathbf{z}) \leq b_m$, it is $f_0(\mathbf{z}) \geq f_0(\mathbf{x}^*)$.

4 Special cases of optimization problems

4.1 Least-squares problem (LS)

A least-squares problem is an optimization problem with no constraints (i.e. $m = 0$) and an objective function which is the sum of squares of terms of the form $\mathbf{a}_i^T\mathbf{x} - b_i$:

$$\text{minimize } f_0(\mathbf{x}) = \|A\mathbf{x} - b\|_2^2 = \sum_{i=1}^{k} \left(\mathbf{a}_i^T\mathbf{x} - b_i\right)^2 \tag{3}$$

where $A \in \mathcal{R}^{k\times n}$ with $(k \geq n)$ is a real matrix, $\mathbf{b} \in \mathcal{R}^k$ is a real vector, $\mathbf{a}_i^T$ for $i = 1, \ldots, k$ are the rows of A, and the vector $\mathbf{x} \in \mathcal{R}^n$ are the optimization variables.

LS problem is one of the few optimization problems that can be solved analytically. The solution of a least-squares problem can be reduced to solving a set of linear equations,

$$(A^TA)\mathbf{x} = A^T\mathbf{b}, \tag{4}$$

so we have the analytical solution $\mathbf{x}^* = (A^TA)^{-1}A^T\mathbf{b}$.

4.2 Linear Programming (LP)

Another important class of optimization problems is linear programming, in which the objective and all constraint functions are linear:

$$\text{minimize } \mathbf{c}^T\mathbf{x} \tag{5}$$

subject to

$$\mathbf{a}_i^T\mathbf{x} \leq b_i, i = 1, \ldots, m. \tag{6}$$

Vectors $\mathbf{c}, \mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathcal{R}^{\setminus}$ and scalars $b_1, \ldots, b_m \in \mathcal{R}$ are parameters that specify the objective function and constraints.

- LP problems do not have solution in analytical form.
- There exist algorithms that solve LP problems efficiently. We will see later the *Simplex* algorithm.

4.3 Convex optimization problems

$$\text{minimize } f_0(\mathbf{x}) \tag{7}$$

s.t

$$f_i(\mathbf{x}) \leq b_i, \text{for } i = 1, \ldots, m. \tag{8}$$

where $f_0(\cdot), f_1(\cdot), \ldots, f_m(\cdot)$ are convex functions i.e they satisfy:

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \tag{9}$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$ and all $\alpha, \beta \in \mathbf{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

Note: for a maximization problem, function $f_0(\cdot)$ needs to be concave.

Non-linear programming problems: all problems that are not linear programming (NP).

Linear Programming Notes

Instructor Iordanis Koutsopoulos

Lecture 2 : 09/10/06

Notes by : Eleni Galanou and Despina Koutsagia

1 Introduction

Apart from linear programming (LP), there is also Non-linear programming (NLP). Convex Optimization Problems is a special class of NLP problems. There are two methods that are used to solve a non- linear problem: local and global optimization.

1. Local Optimization: The optimum is found over a limited region (a neighborhood around the optimal point). Then, it is called local optimum.

   Advantages:

   - In order to locate a local minimum, all we need are some conditions for the first derivative of the objective function.
   - Efficient for heuristic purposes (e.g. to find a fast approximate solution).

   Disadvantages:

   - No information about the quality of the solution (how far away our solution is from the optimal one).
   - Need iterative search algorithms, that rely on starting point $\mathbf{x}_0$.

2. Global Optimization: Difficult to find the solution in general. We will focus on problems of special form whose optimal solution can be easily found (Convex Optimization).

In case the optimal solution cannot be easily found, it is worthwhile to search for bounds (upper or lower) on the value of the objective function at the optimal solution. For example, for a minimization problem, we can give bounds of the type $f(\mathbf{x}^*) \le a, \quad a \in R$.

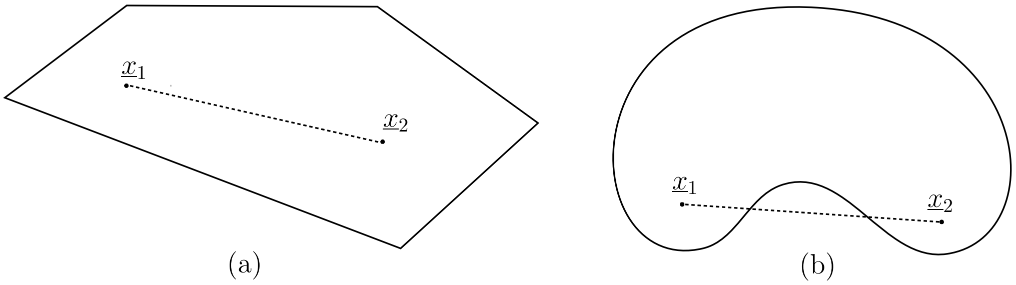
2 Convex Sets

Line Segment: *Line segment* between two points $\mathbf{x}_1$,and $\mathbf{x}_2$ is the set of points $\mathbf{x}$ which can be written as: $\mathbf{x} = \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2$ with $0 \le \theta \le 1$. Due to this special constraint on θ, $\mathbf{x}$ is not a linear but a *convex* combination of $\mathbf{x}_1$, $\mathbf{x}_2$.

Convex Set: A set of points $\mathcal{C}$ is called *convex* if all points on the line segment between any two points of the set $\mathcal{C}$ also belong in $\mathcal{C}$. That is to say, $\mathcal{C}$ is convex set if

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C} \text{ and } 0 \le \theta \le 1, \text{ it is } \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in \mathcal{C} \tag{1}$$

A set of discrete points is non-convex.



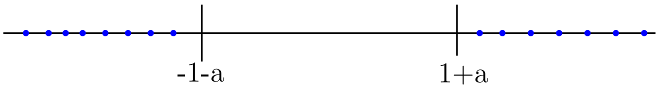
(a) (b)

Figure 1: Examples: (a)a convex set, (b)a non-convex set

For example, the set of points on the real line defined by:

$$\mathcal{C} = \{x : |x-a| \ge 1\} \tag{2}$$

and shown in Figure **??** is not a convex set.



Note: A set of points that is not convex is called non-convex.

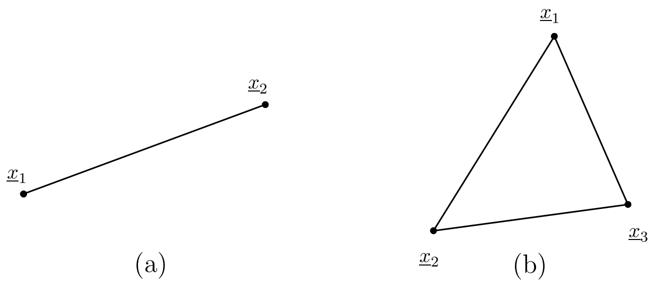
Convex Combination: *Convex combination* of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$ is every point $\mathbf{x}$ of the form:

$$\mathbf{x} = \sum_{i=1}^{K} \theta_i \mathbf{x}_i, \text{ with } \sum_{i=1}^{K} \theta_i = 1. \tag{3}$$

For a set of points $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$, the *Convex hull* of $\mathcal{C}$, $Conv(\mathcal{C})$, is the set of all convex combination of points of $\mathcal{C}$. That is,

$$\mathcal{C} = \left\{ \mathbf{x} : \mathbf{x} = \sum_{i=1}^{K} \theta_i \mathbf{x}_i, \text{ for } \mathbf{x}_i \in \mathcal{C}, \text{ with } 0 \leq \theta \leq 1, \text{ for } i = 1, \ldots, K \text{ and } \sum_{i=1}^{K} \theta_i = 1 \right\} \tag{4}$$

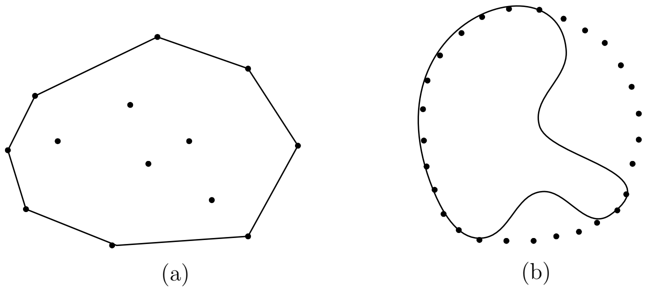
Special Cases: If $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2\}$, then $Conv(\mathcal{C})$ is the line segment connecting $\mathbf{x}_1$ and $\mathbf{x}_2$. If $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, then $Conv(\mathcal{C})$ is the triangle with these points as vertices.



(a) (b)

Figure 2: Convex hull for (a) $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2\}$ (b) $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$

Note: Convex hull of a set of points $\mathcal{C}$ is the smallest convex set that encloses all points of $\mathcal{C}$.



(a) (b)

Figure 3: Convex hull of (a) a discrete set of points, (b) a continuous set of points.

Note: $Conv(\mathcal{C})$ is always a continuous and convex set.

If $\mathcal{C}$ is convex and continuous, then $Conv(\mathcal{C}) = \mathcal{C}$.

If $\mathcal{C}$ is non-convex and continuous, then $Conv(\mathcal{C} \supset \mathcal{C}$.

If $\mathcal{C}$ is discrete set, then $Conv(\mathcal{C} \supset \mathcal{C}$.

3 Properties of Convex Sets

1. If $\mathcal{C}$ is convex set and $b \in \mathcal{R}$ is real number, then the set

   $$\mathcal{D} = b\mathcal{C} = \{\mathbf{x} : \mathbf{x} = b\mathbf{u} : \mathbf{u} \in \mathcal{C}\}. \tag{5}$$

   is also convex. This is the *scaling property*, namely: when a convex set is multiplied by a real number, the resulting set remains convex.

2. If $\mathcal{C}_1$, $\mathcal{C}_2$ are convex sets then $\mathcal{C}_1 + \mathcal{C}_2$ is also convex, where:

   $$\mathcal{C}_1 + \mathcal{C}_2 = \{\mathbf{x} : \mathbf{x} = \mathbf{u}_1 + \mathbf{u}_2,\ \text{where } \mathbf{u}_1 \in \mathcal{C}_1 \text{ and } \mathbf{u}_2 \in \mathcal{C}_2\} \tag{6}$$

   This is the *addition* property. Example for set addition: If $\mathcal{C}_1 = \{1, 2\}$ and $\mathcal{C}_2 = \{10, 15, 18\}$, then $mathcal C_1 + \mathcal{C}_2 = \{11, 16, 19, 12, 17, 20\}$.

3. If $\mathcal{C}_1, \mathcal{C}_2$ are convex sets, then $\mathcal{C}_1 \cap \mathcal{C}_2$ is convex. This is the *intersection* property.

Note: Generalization of convex combinations. The convex combination of a distinct set of points are all points $\mathbf{x}$, such that:

$$\mathbf{x} = \sum_{i=1}^{K} \theta_i \mathbf{x}_i \text{ with } \sum_{i=1}^{K} \theta_i = 1 \tag{7}$$

Generalize for continuous set of points $\mathcal{C}$ and continuous coefficients θ.

Consider functions $P : \mathcal{R} \to \mathcal{R}^n$ that satisfy $P(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{C}$ and

$$P(\mathbf{x}) = \int_{\mathbf{x} \in \mathcal{C}} P(\mathbf{x}) = 1 \tag{8}$$

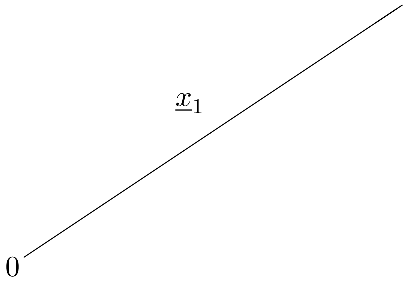
Then the convex combination is that case is given by: $\int_{\mathbf{x} \in \mathcal{C}} \mathbf{x} P(\mathbf{x}) d\mathbf{x} = E[X]$. If $\mathbf{x}$ is a *random* vector, then the above becomes the mean $E[X]$.

4 Cones, Hyperplanes, Polyhedra

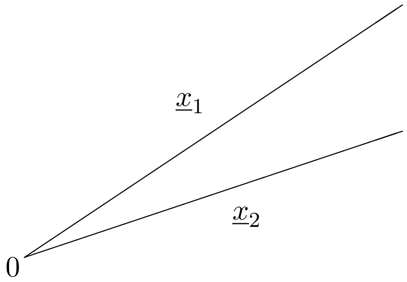
Cones: A set of points $\mathbf{C}$ is called *cone* if

$$\forall \mathbf{x} \in \mathcal{C} \quad \text{and} \quad \theta \geq 0 \quad \text{it is} \quad \theta \mathbf{x} \in \mathbf{C} \tag{9}$$

For $\mathcal{C} = \{\mathbf{x}_1\}$, the cone is the line that begins from 0 and passes through $\mathbf{x}_1$, and is shown in the figure below.



For $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2\}$, the cone consists of the two straight lines beginning from 0 and passing through $\mathbf{x}_1$ and from 0 and passing through $\mathbf{x}_2$. (but not from points in between these two lines).



Generally the cone is not a convex set. However we are interested in the cones that are convex.

A set $\mathcal{C}$ is called *convex cone* if

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C} \quad \text{and} \quad \theta_1, \theta_2 \geq 0 \quad \text{it is} \quad \theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 \in \mathcal{C} \tag{10}$$

The above is called *conic combination* of $\mathbf{x}_1, \mathbf{x}_2$, which is a special case of linear combination, since it is defined to be every linear combination of two points with positive multiplication factors.

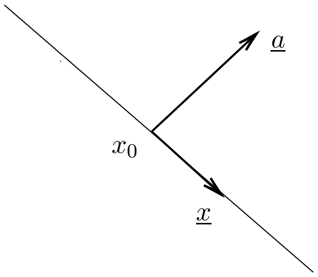
Hyperplane: A *Hyperplane* $\mathcal{P}$ is a set of points x with a constant inner product to some given vector $\mathbf{a} \neq 0$:

$$\mathcal{P} = \{\mathbf{x} \in \mathcal{R}^n : \mathbf{a}^T \mathbf{x} = b\} \tag{11}$$

where $b \in \mathcal{R}$.

$$\underline{x} = (x_1, \ldots, x_n) \quad , \underline{a} = (a_1, \ldots, a_n)$$

Alternatively, $\mathcal{P} = \{(x_1, ..., x_n) : a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b\}$. A projection of a hyperplane in a two-dimensional plane is shown in the figure below.



The hyperplane $\mathcal{P}$ equation is also written as: $\mathbf{a}^T(\mathbf{x} - \mathbf{x}_0) = 0$ where x_0 is any point on $\mathcal{P}$, (namely, it satisfies $\mathbf{a}^T\mathbf{x}_0 = b$). Vector $\mathbf{a}$ *defines* a hyperplane $\mathcal{P}$, and it is vertical to *all* points of the hyperplane.

The dimension of $mathbfx$, n, specifies the type of hyperplane.

1. If $n = 2$,the hyperplane is:

$$\mathcal{P} = \{(x_1, x_2) : a_1x_1 + a_2x_2 = b\} \tag{12}$$

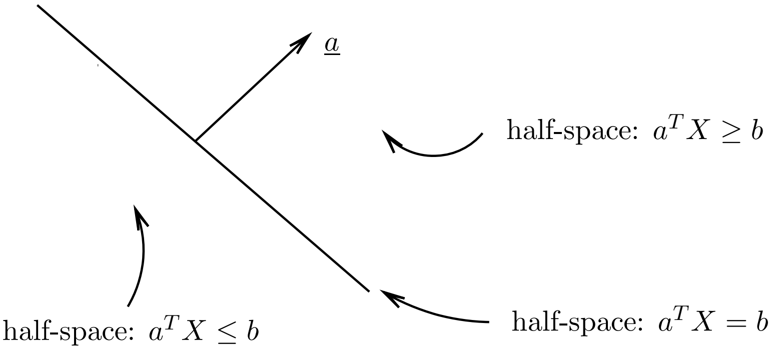
which is a straight line with slope defined by vector (a_1, a_2).

2. If $n = 3$, (three-dimensional space), then

$$\mathcal{P} = \{(x_1, x_2, x_3) : a_1x_1 + a_2x_2 + a_3x_3 = b\} \tag{13}$$

which is the usual plane.

Half-spaces: A hyperplane $\mathcal{P}$ satisfying $\mathbf{a}^T\mathbf{x} = b$ divides the space into two *Half-spaces*. The one half-space is all points $\mathbf{x}$ such that $\mathbf{a}^T\mathbf{x} \leq b$ and the other one consists of all points $\mathbf{x}$ such that $\mathbf{a}^T\mathbf{x} \geq b$.



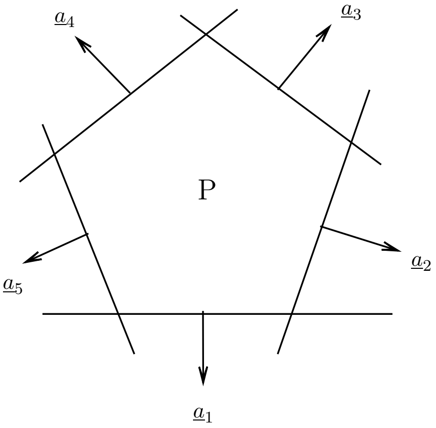
Polyhedron: A *Polyhedron* is the intersection of many half-spaces and hyper-planes

$$\mathcal{P} = \{ \mathbf{x} : \quad \mathbf{a}_j^T \mathbf{x} \leq b_j \quad , j = 1, \ldots, m \quad \text{and} \quad \mathbf{c}_j^T \mathbf{x} = d_j \quad , j = 1, \ldots, p \quad \}$$

In this definition, there are m half-spaces and p hyper-planes that make up the polyhedron $\mathcal{P}$.

Example: A polyhedron defined as intersection of 5 half-spaces

$$\mathbf{a}_j^T x \leq b_j \quad , j = 1, ..., 5 \tag{14}$$



- If the polyhedron is closed (bounded), it is called a *polytope*.

5 Application 1: Region of feasible transmission rates

Consider M BSs and N users per BS. A single-channel (slot) system is assumed.

Let G_{ij} be the channel gain for the link between BS i and user j. Gain G_{ij} shows the distance of the user from the BS,and also accounts for fading. We consider that all G_{ij} are fixed and known and will not worry about temporal evolution of them.

Each user is connected to a certain BS.

We will examine the special case $M = 2, N = 2$ for downlink and uplink and will characterize the region of feasible BS transmission rates. Each BS can transmit to one user at most in the channel.

Constraint: at the receiver of each user j, the Signal-to-Interference Ratio SIR_j should exceed a threshold γ_j, namely

$$SIR_j \geq \gamma_j \tag{15}$$

where $\gamma_j = f(r_j)$, where $f(\cdot)$ is an increasing function and r_j is the transmission rate from the corresponding BS to user j.

Function $f(r_j)$ specifies the *smallest* SIR_j at the receiver of a user j, when transmission towards him is with rate r_j, so that the bit-error-rate (BER) is at most ϵ. A usual form of f is

$$f(r_j) = \frac{-\ln(5\epsilon)}{1.5}(2^{r_j} - 1) \tag{16}$$

More about this problem, on next Lecture, Lecture 3.

Advanced Topics in Networking - Fall 2006

Instructor Iordanis Koutsopoulos

Lecture 3 : 10/10/06

Notes by : Charalampos Daskalakis and Constantinos Houmas

1 Transmission rate region for down-link

System with M Base Stations (BS) and N Users for each BS, in down-link.

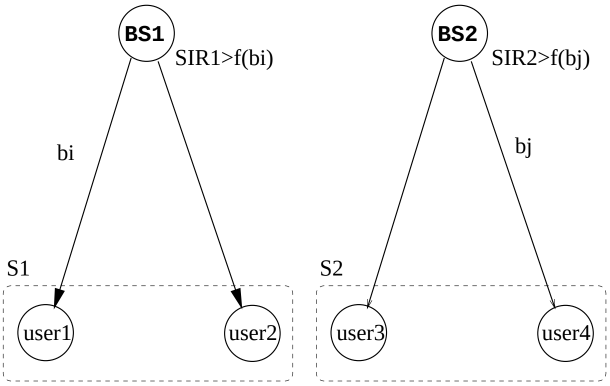


Figure 1: Two BSs with two users for each BS

Model: Each BS communicates with all users assigned to it, using one carrier frequency with TDMA. Let S_i be the set of users assigned to BS$_i$.

Our main goal is to determine the *feasible* transmission rate vector $(r_1, ..., r_M)$ for each BS and then to characterize the *region* of achievable rates. Note that r_i is the transmission rate of BS$_i$, for *some* receiving user.

A BS is called *active*, if it transmits to a user. An active BS can transmit to at most one user in S_i at the same time. A BS$_i$ can choose among different transmission rates in the set $\mathcal{B} = \{0, b_1, \ldots, b_L\}$ to transmit to a user at each time slot. For each selected rate $b_l \in \mathcal{B}$, there is a minimum required

Signal-to- Interference ratio(SIR) $\text{SIR}_{ij} = f(b_l)$, at the receiver of the selected user j by BS_i (if BS i transmits to user $j \in S_i$ with rate b_l).

Let u_i be the user to which BS_i transmits. Each BS_i should choose rate $\max\limits_{b_l \in \mathcal{B}}\{b : f(b) \leq SIR_{iu_i}\}$, for some $u_i \in S_i$. Clearly for a larger transmission rate, we need a better quality channel to keep the Bit Error Rate(BER) below a threshold ε (BER $\leq \varepsilon$). Therefore, function $f(\cdot)$ is increasing ($\uparrow$).

In our example where $M = N = 2$, a vector of transmission rates (r_1, r_2) is called *feasible* in a time slot if $r_1, r_2 \in \mathcal{B}$ (note that $r_i = 0$ if BS_i is inactive) and $SIR_{iu_i} \geq f(r_i)$, where $u_i \in S_i$, for $i = 1, 2$.

The *BS activation vector* $\mathbf{q}$ is a binary vector, for which $q_i = 1$ if BS_i is active and $q_i = 0$ otherwise. In our example, the set of possible $\mathbf{q}$ is $\mathcal{A} = \{(0,0), (0,1), (1,0), (1,1)\}$.

If $\mathbf{q} = (0,0)$ there is no communication,
else if $\mathbf{q} = (1,0)$, BS_1 can choose among $\{u_1, u_2\}$ (only BS_1 transmits),
else if $\mathbf{q} = (0,1)$, BS_2 can choose among $\{u_3, u_4\}$ (only BS_2 transmits),
else if $\mathbf{q} = (1,1)$, BS_1 and BS_2 can choose among $\{(u_1, u_3), (u_1, u_4), (u_2, u_3), (u_2, u_4)\}$ to transmit to respectively (both BS_1, BS_2 transmit simultaneously).

Selection of users for a given q. For $\mathbf{q} = (1,0)$ or $\mathbf{q} = (0,1)$ we select user u_1 (or u_2 respectively) that has the largest SNR. For activation vector $\mathbf{q} = (1,1)$, (where there is interference), intuitively we can select users u_1 and u_4 which are further from BS_1, BS_2 (Fig.1), since their SIR will be maximum. The mathematical formula which gives the SINR at the receiver of user i for given activation vector $\mathbf{q}$ is:

$$SINR_{iu_i} = \begin{cases} \dfrac{G_{iu_i}}{\sigma^2 + \sum\limits_{k \neq i, q_k = 1} G_{ku_i}}, \text{ if } \sum\limits_{k \neq i} q_k > 0 \\ \\ \dfrac{G_{iu_i}}{\sigma^2}, \text{ if } \sum\limits_{k \neq i} q_k = 0 \end{cases} \tag{1}$$

where $u_i \in S_i$, G_{ij} is the gain of the transmission channel between BS_i and user j, and σ^2 is the noise power. Note that $\sum\limits_{k \neq i} q_k > 0$ means that another BS (apart from i) transmits and causes interference to user u_i. When $\sum\limits_{k \neq i} q_k = 0$ and $q_i = 1$, then BS_i is the only active BS. In that case there is only noise at the receiver of user u_i.

Therefore, for given activation vector $\mathbf{q}$, BS_i chooses to transmit to user:

$$u_i^*(\mathbf{q}) = arg \max_{u_i \in S_i} \left\{ \max_{b \in \mathcal{B}} (b : \mathrm{SIR}_{iu_i} \geq f(b)) \right\} \tag{2}$$

We can observe that the user choice for each BS depends only on whether other BSs are active and *not* on the other BSs' choices. This happens in the downlink, but not in uplink, as we will see later.

For example, let's assume that $L = 3$ and $\mathcal{B} = \{0, 50\text{Kbps}, 100\text{Kbps}, 200\text{Kbps}\}$. Without loss of generality, we focus on BS_1's choice of user and transmission rate. Suppose that $\mathrm{SIR}_{11} = 10\text{dB}$, $\mathrm{SIR}_{12} = 20\text{dB}$ and the corresponding SIR thresholds for $b = 0, 50\text{Kbps}, 100\text{Kbps}, 200\text{Kbps}$ are $f(b) = 0, 5\text{dB}, 12\text{dB}, 15\text{dB}$. In this case BS_1 selects user 2 because it has the largest SIR, and transmits with rate that equals to 200Kbps, because it is the maximum and satisfies that $f(200\text{Kbps}) = 15\text{dB} \leq 20\text{dB}$. Note that we do not consider fairness issues in our example, but we only focus on how to maximize the transmission rate of a BS.

Assuming that the link gains do not change with time, for each activation vector $\mathbf{q}$ we can find a rate vector $\mathbf{r_q} = (r_{\mathbf{q}1}, r_{\mathbf{q}2})$, where $r_{\mathbf{q}1}, r_{\mathbf{q}2} \in \mathcal{B}$. This vector represents the transmission rates of the BSs at each slot when the activation vector is $\mathbf{q}$.

Let $t_\mathbf{q} \in [0, 1]$ be the portion of time when activation vector $\mathbf{q}$ is used. Then the BS transmission rate vector $\mathbf{R}$ over the entire (presumably long) time horizon is given by:

$$\mathbf{R} = \sum_{\mathbf{q} \in \mathcal{A}} t_\mathbf{q} \mathbf{r_q} \tag{3}$$

Different time portions $t_\mathbf{q}$ for each activation vector $\mathbf{q}$ (and therefore rate vector $\mathbf{r_q}$) result in different rate vectors $\mathbf{R}$. The set $\mathcal{Z}$ of all feasible rate vectors is:

$$\mathcal{Z} = \left\{ \sum_{\mathbf{q} \in \mathcal{A}} t_\mathbf{q} \mathbf{r_q} : \sum_{\mathbf{q} \in \mathcal{A}} t_\mathbf{q} = 1, t_\mathbf{q} \geq 0 \right\} = conv \{\mathbf{r_q} : \mathbf{q} \in \mathcal{A}\} \tag{4}$$

Thus $\mathcal{Z}$ is the convex hull of the rate vectors that correspond to different activation vectors.

For example, assume that $\mathbf{q}_0 = (0, 0)$, $\mathbf{q}_1 = (1, 0)$, $\mathbf{q}_2 = (0, 1)$, $\mathbf{q}_3 = (1, 1)$ and $\mathcal{B} = \{0, b_1, b_2, b_3\}$ $(0 < b_1 < b_2 < b_3)$. If $\mathbf{r}_{\mathbf{q}_0} = (0, 0)$, $\mathbf{r}_{\mathbf{q}_1} = (b_3, 0)$, $\mathbf{r}_{\mathbf{q}_2} = (0, b_3)$ and $\mathbf{r}_{\mathbf{q}_3} = (b_2, b_3)$, then the region of feasible transmission rates can be seen in Fig.2.

An appropriate generalization can be done when link gains vary with time.

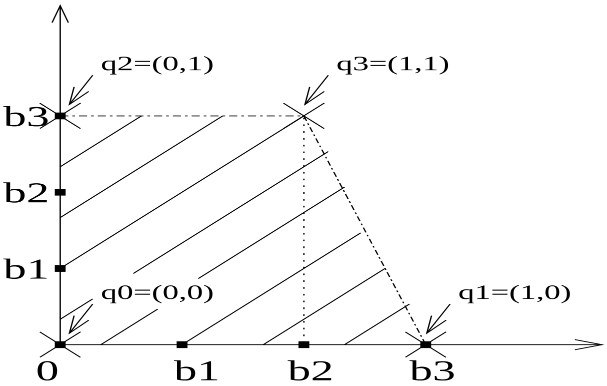


Figure 2: The region of feasible transmission rate vectors for downlink.

2 Transmission rate region for an uplink System

In the uplink, users transmit to BSs. The way of computing feasible transmission rates is quite similar to the downlink case, except for the following difference: In the uplink, the receiver is at the BS. For a BS to select a user, it is not enough to know whether the other BS is active or not, but it also needs to know *which* is the user that transmits to the other BS. Different users cause different amount of interference in the other BS (see Fig.3).

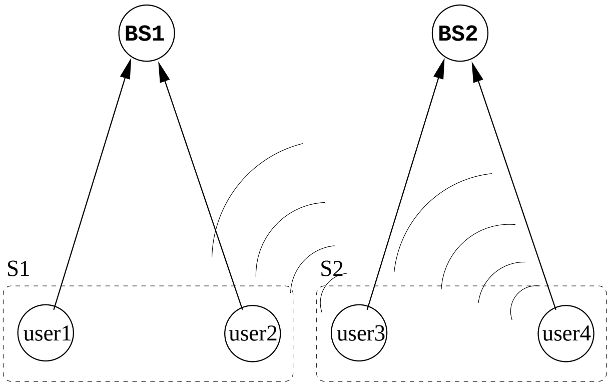


Figure 3: user 3 and user 4 cause different interference to BS_1

Therefore in the uplink, the activation vectors $\mathbf{q}_0 = (0,0)$, $\mathbf{q}_1 = (1,0)$ and $\mathbf{q}_2 = (0,1)$ (for $M = N = 2$), correspond to the same transmission rate vectors $\mathbf{r_q}$ as in the downlink case, since at most one BS is active. For $\mathbf{q}_3 = (1,1)$, we have different scenarios, and rate vectors depend on which user transmits. As we see in Fig.4, for a different pair of selected users for BS_1, BS_2 we have different rate vectors in the diagram. Of course, the region of feasible transmission rate vectors is again the convex hull of rate vectors for different BS activation vectors and different user selections.

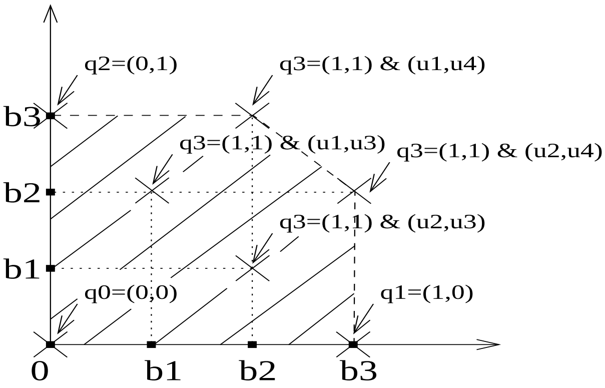


Figure 4: The region of feasible transmission rate vector for up-link

Advanced Topics in Networking - Fall 2006

Instructor: Iordanis Koutsopoulos

Lecture 4 : Power Control and Base Station Assignment - 11/10/06

Notes by : Anastasia Narou and Maria Papadopoulou

1 Models of uplink transmission

Transmission modes:

- CDMA: more than one users can communicate with one Base Station at the same slot and the same frequency.This is due to the fact that different user transmissions can be distinguished at the same BS, since each user uses a different code to pre-multiply its signal with, before transmission.
- TDMA: at most one user can communicate with each Base Station at each slot.
- FDMA: at most one user can communicate with each Base Station at the same frequency.

We will focus on CDMA transmission.In our example we will use two Base Stations and two users for each BS.

1.1 Assumptions

- M BS, N users
- Each user is assumed to be connected to one BS in order to transfer its message. We will see that users can switch between BSs with which they talk in order to achieve smaller transmission power.
- "Frozen" system with no change of link gains (defined below).

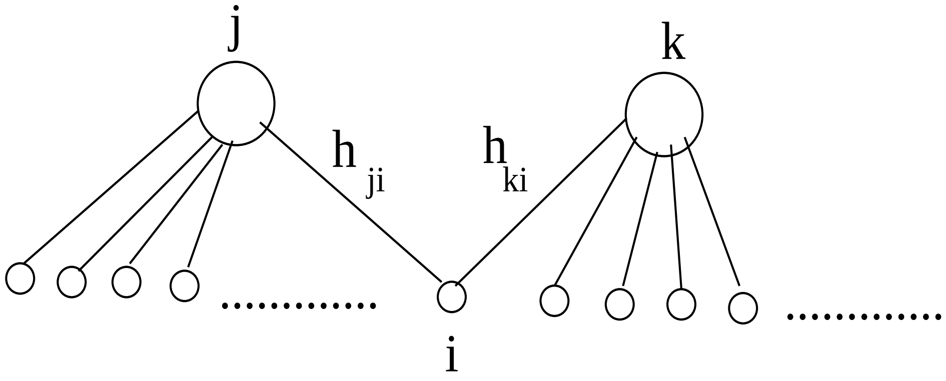


Figure 1: Users sending to BSs in an uplink scenario.

- Degree of freedom: for each user i, we define $a_i = k$ if user i communicates with BS k, $a_i \epsilon \{1, ..., M\}$.
- We assume that the system operates in a single channel.
- We define h_{ki} as the link gain between user i and BS k. The link gain is the fraction of received power over transmitted power.Thus, if i transmits with unit power, BS k receives amount of power h_{ki}.

Example: We consider M BS and n users who communicate with one of the two BSs, as shown in Figure 1.1.

As mentioned before, suppose we freeze the system. We assume all link gains $\{h_{ki}\}$, for $k = 1, 2$, $i = 1, 2$ are fixed and do not worry about temporal variations.

User i transmits power P_i. Then the SINR at the receiver corresponding to user i at BS $a_i = k$ is

$$\mathrm{SINR}_i = \frac{h_{ki}P_i}{\sum\limits_{j \neq i} h_{kj}P_j + \sigma_k^2} = \frac{h_{a_i i}P_i}{\sum\limits_{j \neq i} h_{a_i j}P_j + \sigma_{a_i}^2} \tag{1}$$

Note that SINR_i depends on the transmission power of user i, P_i and the interference created by other transmitting users $j \neq i$. This interference is created at the receiver corresponding to user i in the BS with which i talks(i.e, BS $a_i = k$).

SINR_i depends on the entire power vector $P = (P_1, ..., P_n)$ and noise power $\sigma_{a_i}^2$ at the receiver.

SINR_i also depends on assignment factor a_i (which is the Base station i talks to. Note that it is independent of BS assignments a_j of other users, $j \neq i$.

Define the assignment vector $\mathbf{a} = (a_1, ..., a_n)$ where a_i is the BS with which user i is connected. The dependencies above are expressed by denoting the SINR as: $\text{SINR}_i(a_i, \mathbf{P})$.

Thus, the following should hold:

$$\text{SINR}_i(a_i, \mathbf{P}) \geq \gamma_i \ \forall i, \tag{2}$$

where γ_i refers to the minimum SINR threshold for user i.

Now, we want to satisfy a minimum SINR threshold, γ_i for each user i, which is equivalent to saying that BER $\leq \epsilon$ at the receiver of each user. A different SINR threshold γ_i for user i may mean that different users require different minimal quality in their links to achieve BER at most ϵ. This is the case when users transmit with different rates because they carry different types of traffic (e.g, video,audio etc). Higher transmission rate for a user i means higher SINR threshold γ_i.

The objective is formulated as an optimization problem:

$$\min_{\mathbf{a},\mathbf{P}} \sum_{i=1}^{N} P_i \tag{3}$$

subject to:

$$\text{SINR}_i(a_i, \mathbf{P}) \geq \gamma_i \tag{4}$$

and also $P_i \geq 0$, for all $i = 1, ..., N$. Also, it should be $a_i \in \{1, ..., M\}$. Call the formulation above "Problem (P)".

The objective is to find the BS assignment vector $\mathbf{a}$ and the user transmission power vector $\mathbf{P}$ such that all users fulfil their $SINR$ threshold requirements $\text{SINR}_i \geq \gamma_i$ and the total transmission power of users is minimized.

Notice that two different users may talk with the same BS (because of the $CDMA$ assumption).

1.2 Problem Analysis with $M = 2$ and $N = 2$

Consider the case of $M = 2$ BSs and $N = 2$ users and let user 1 be connected to BS1. The SINR of user 1 is:

$$\text{SINR}_1 = \frac{h_{11}p_1}{h_{12}p_2 + \sigma_1^2} \geq \gamma_1 \Longrightarrow P_1 \geq \frac{\gamma_1 h_{12}}{h_{11}} P_2 + \frac{\sigma_1^2 \gamma_1}{h_{11}} \Longrightarrow P_1 \geq (0 \quad \frac{\gamma_1 h_{12}}{h_{11}})\mathbf{P} + \frac{\sigma_1^2 \gamma_1}{h_{11}}. \tag{5}$$

Observe that $SINR_1$ is not affected from the BS with which user 2 is connected (since anyway, there will be the same interference at the receiver of user 1).

Vector $\left(0 \;\; \frac{\gamma_1 h_{12}}{h_{11}}\right)$ is symbolized as $\mathbf{H}_1^{(1)}$ and shows that user 1 communicates with BS 1. The "0" in the first element of $\mathbf{H}_1^{(1)}$ is because a user cannot cause interference to himself. Also, quantity $(\frac{\sigma_1^2\gamma_1}{h_{11}})$ is symbolized as $\delta_1^{(1)}$.

Thus, the $SINR$ requirement for user 1, if it is connected to BS 1 is written as:

$$P_1 \geq \mathbf{H}_1^{(1)}\mathbf{P} + \delta_1^{(1)} \tag{6}$$

Likewise, suppose user 2 is connected to BS 1, then, the SINR requirement $\mathbf{SINR}_2 \geq \gamma_2$ is written as:

$$P_2 \geq (\frac{\gamma_2 h_{11}}{h_{12}} \;\; 0)\mathbf{P} + \frac{\sigma_1^2\gamma_2}{h_{12}} \tag{7}$$

or

$$P_2 \geq \mathbf{H}_2^{(1)}\mathbf{P} + \delta_2^{(1)} \tag{8}$$

In general vector $H_i^{(k)}$ is of dimension $1 \times N$ and has elements given by:

$$H_{ij}^k = \begin{cases} \frac{\gamma_i h_{kj}}{h_{ki}}, & \text{if } j \neq i \\ H_{ij}^{(k)} = 0, & \text{if } j = i \end{cases} \tag{9}$$

and specifies that user i is connected to BS k. Also $\delta_i^k = \frac{\gamma_i \sigma_k^2}{h_{ki}}$ for the general case.

Note that in the vector $\mathbf{H}_i^k$, element $H_{ii}^k = 0$, since a user does not cause interference to itself.

Next, define the set $\mathcal{L}$ of all possible assignments of users to BSs. For example in the case of $M = 2, N = 2$ there are 4 possible assignments:

1. user 1 → BS 1, user 2 → BS 1
2. user 1 → BS 1, user 2 → BS 2
3. user 1 → BS 2, user 2 → BS 1
4. user 1 → BS 2, user 2 → BS 2

Let ℓ be a specific assignment in $\mathcal{L}$, then $a_i(\ell)$ is the BS to which user i is assigned based on assignment ℓ. Since each user $i = 1, 2$ can communicate with a BS $k = 1, 2$, we can define $\mathbf{H}_i^{(k)}$, $\delta_i^{(k)}$, for each $i = 1, 2$ and $k = 1, 2$. Problem(P) can be divided in two subproblems:

1. Find transmission power for a given BS assignment to user.
2. Find the BS assignment for a given user transmission power.

For subproblem 2: Suppose users 1 and 2 transmit with power P_1 and P_2 respectively. Which BS assignment should we do in order to have the minimum total transmission power?

It makes sense that user 1 will connect to the BS that results in the largest $SINR$ (out of the two possible BSs). The same for user 2.

For subproblem 1: If we have fixed BS assignment **a** then we will have the classical problem of power control. For a particular assignment $\ell \in \mathcal{L}$ the SINR requirement for user i is:

$$\min_{\mathbf{P}} \sum_{i=1}^{N} P_i \tag{10}$$

subject to:

$$\mathbf{SINR}_i(\mathbf{P}) \geq \gamma_i \tag{11}$$

Now, consider again the original problem. We symbolize $\mathbf{G}_i^{(\ell)} = \mathbf{H}_i^{(a_i(\ell))}$. If we write inequalities for each user i and assignment $\ell \in \mathcal{L}$, we have,

$$P_i \geq \mathbf{G}_i^{(\ell)}\mathbf{P} + \delta_i^{(\ell)} \tag{12}$$

and in vector form:

$$\mathbf{P} \geq \mathbf{G}^{(\ell)}\mathbf{P} + \boldsymbol{\delta}^{(\ell)} \tag{13}$$

where matrix:

$$\mathbf{G}^{(\ell)} = [\mathbf{G}_1^{(\ell)}, \mathbf{G}_2^{(\ell)}, ..., \mathbf{G}_N^{(\ell)}] \tag{14}$$

and vector

$$\boldsymbol{\delta}^{(\ell)} = (\delta_1^{(\ell)}, ..., \delta_N^{(\ell)})^T \tag{15}$$

Consequently the set of feasible transmission powers for a particular assignment ℓ is

$$\mathcal{P}^{(\ell)} = \{\mathbf{P} \geq \mathbf{0} : \mathbf{P} \geq \mathbf{G}^{(\ell)}\mathbf{P} + \boldsymbol{\delta}^{(\ell)}\} \tag{16}$$

One can show that set $\mathcal{P}^{(\ell)}$ is a cone. Indeed, suppose $\mathbf{P} \in \mathcal{P}^{\ell}$, then $\alpha\mathbf{P} \in \mathcal{P}^{\ell}$, for $\alpha \geq 1$.

Consider $M = 2$ and $N = 2$. For a fixed assignment ℓ, the set of feasible powers $\mathcal{P}^{(\ell)}$ is the cone that is defined by inequalities:

$$P_1 \geq \mathbf{G}_1^{(ell)}\mathbf{P} + \delta_1^{(\ell)} \tag{17}$$

$$P_2 \geq \mathbf{G}_2^{(\ell)}\mathbf{P} + \delta_2^{(\ell)} \tag{18}$$

From the theory of power control, the power vector $\mathbf{P}^{*(\ell)} = (P_1^{(\ell)}, P_2^{(\ell)})$ that solves the power minimization problem is the intersection of straight lines

$$P_1 = \mathbf{G}_1^{(\ell)}\mathbf{P} + \delta_1^{(\ell)} \tag{19}$$

$$P_2 = \mathbf{G}_2^{(\ell)}\mathbf{P} + \delta_2^{(\ell)} \tag{20}$$

which is the vertex of the cone, $\mathbf{P}^{*(\ell)} = \mathbf{v}^{(\ell)}$ and, for a specific assignment is shown in the figure below (Figure 2.

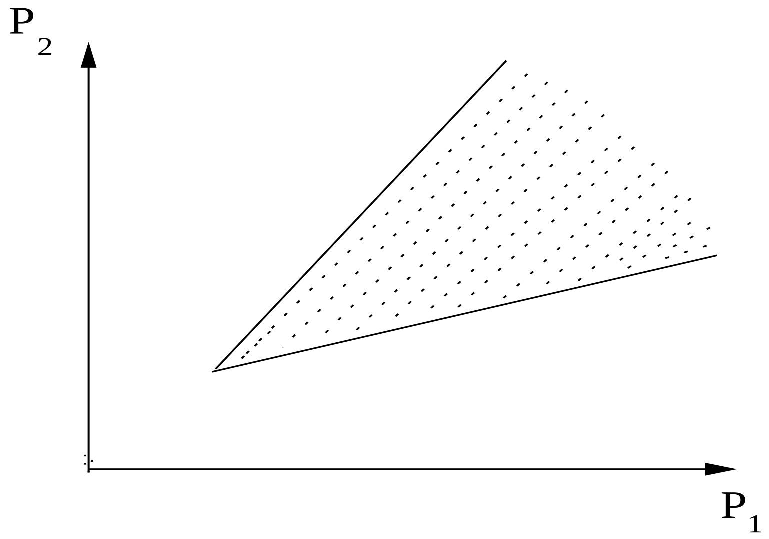


Figure 2: Region of feasible powers for a fixed BS assignment.

In figure 3 below, we show all 4 possible assignments of BS to users. In fact, two out of the 4 straight lines with equations shown above, correspond to a specific BS assignment $\ell = 1, 2, 3, 4$.

$\longrightarrow$ straight line (a) represents communication between user 2 and BS 1 and has equation

$$P_2 = \mathbf{H}_2^{(1)}\mathbf{P} + \delta_2^{(1)} \tag{21}$$

$\longrightarrow$ straight line (b) represents communication between user 1 and BS 1 and has equation

$$P_1 = \mathbf{H}_1^{(1)}\mathbf{P} + \delta_1^{(1)} \tag{22}$$

$\longrightarrow$ straight line (c) represents communication between user 2 and BS 2 and has equation

$$P_2 = \mathbf{H}_2^{(2)}\mathbf{P} + \delta_2^{(2)} \tag{23}$$

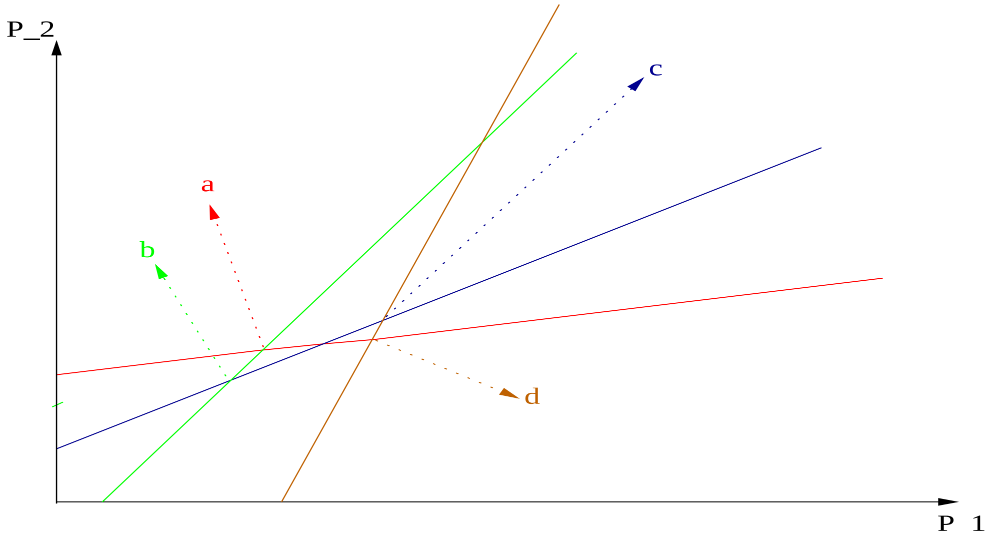


Figure 3: 4 lines showing 4 possible assignments to BSs.

$\longrightarrow$ straight line (d) represents communication between user 1 and BS2 and has equation

$$P_1 = \mathbf{H}_1^{(2)}\mathbf{P} + \delta_1^{(2)} \tag{24}$$

$\Diamond$ Now as far as the points of intersection of these lines are concerned :

$\longrightarrow$ Lines (b) and (c) represent assignment $\ell = 1$ (user 1 $\rightarrow$ BS 1, user2 $\rightarrow$ BS 2). The point of intersection of these lines is denoted as $\mathbf{v}^{(1)}$.

$\longrightarrow$ Lines (a) and (d) represent assignment $\ell = 2$ (user 1 $\rightarrow$ BS 2,user 2 $\rightarrow$ BS 1). The point of intersection of these lines is denoted as $\mathbf{v}^{(2)}$.

$\longrightarrow$ Lines (a) and (b) represent assignment $\ell = 3$ (user 1 $\rightarrow$ BS 1, user 2 $\rightarrow$ BS 1). The point of intersection of these lines is denoted as $\mathbf{v}^{(3)}$.

$\longrightarrow$ Lines (c) and (d) represent assignment $\ell = 4$ (user 1 $\rightarrow$ BS 2, user 2 $\rightarrow$ BS 2). The point of intersection of these lines is denoted as $\mathbf{v}^{(4)}$.

Theorem: Out of the cone vertices $\mathbf{v}^{(1)}, ..., \mathbf{v}^{(|\mathcal{L}|)}$, there exists a vertex $\mathbf{v}^* \in \{\mathbf{v}^{(1)}, ..., \mathbf{v}^{(|\mathcal{L}|)}\}$, such that $\mathbf{v}^* \leq \mathbf{v}^{(\ell)}$ for all $\ell = 1, ..., |\mathcal{L}|$. This vertex shows the BS assignment and the power vector (the power vector is the one that corresponds to that intersection point) that form the solution to the joint BS assignment and power control problem.

We make a note about the algorithm that solves the problem.

Define the vector

$$\mathbf{M}_i^{(k)}(\mathbf{P}) = \mathbf{H}_i^{(k)}\mathbf{P} + \delta_i^{(k)} \tag{25}$$

where $\mathbf{M}_i^{(k)}(\mathbf{P})$ actually represents P_i, depends only on P_j,$j \neq i$ and it is the minimum power needed by user i to transmit to BS k *if other users $j \neq i$ keep their powers fixed.*

Define vector

$$\mathbf{M}_i(\mathbf{P}) = \min_k \mathbf{M}_i^{(k)}(\mathbf{P}). \tag{26}$$

This is the minimum power needed by user i to transmit to *any* BS.

$\underline{\underline{\text{Algorithm}}}$:

Start with an initial assignment $\mathbf{a}^0$ and power vector $\mathbf{P}^0$. that is vector $(a_1^0, a_2^0), (P_1^0, P_2^0)$.

Users take turns, one at a time (user 1, then user 2, etc.) and produce assignments and powers (a_1^n, P_1^n) and (a_2^n, P_2^n).

At each iteration, each user i selects the *best* BS out of BS 1, BS 2, namely the one that allows the user to achieve γ_i with the least power P_i. For example, user 1 does the following:

$$P_1^{(n+1)} = \min\left\{\mathbf{M}_1^{(1)}(\mathbf{P}^{(n)}), \mathbf{M}_1^{(2)}(\mathbf{P}^{(n)})\right\} \tag{27}$$

where $\mathbf{M}_1^{(1)}(\mathbf{P}^n), \mathbf{M}_1^{(2)}(\mathbf{P}^n)$ depend $\underline{\text{only}}$ on the power of the other user P_2^n (at the previous iteration).

Specifically:

$$P_1^{(n+1)} = \min\left\{\mathbf{H}_1^{(1)}\mathbf{P}^{(n)} + \delta_1^{(1)}, \mathbf{H}_1^{(2)}\mathbf{P}^{(n)} + \delta_1^{(2)}\right\} \tag{28}$$

or in even more detailed form:

$$P_1^{(n+1)} = \min\left\{\frac{\gamma_1 h_{12}}{h_{11}}P_2^{(n)} + \frac{\sigma_2^2\gamma_1}{h_{11}}, \frac{\gamma_1 h_{22}}{h_{21}}P_2^{(n)} + \frac{\sigma_2^2\gamma_1}{h_{21}}\right\}. \tag{29}$$

Thus, we get assignment $a_1^{(n+1)} \in \{1, 2\}$ and the power of user 1 is fixed to the minimum needed so as to achieve SINR γ_1. Similarly by the update $P_2^{(n+1)}$, and user 2 selects BS and power.

By doing this iteration, the algorithm converges to the optimal solution (optimal BS assignment and power vector) that was represented by vertex $\mathbf{v}^*$.

Contraction mapping theory helps us understand better the convergence of such iterative algorithms. A basic notion in contraction mapping theory is the *fixed point* of a function.

The point $\mathbf{x}^*$ is called fixed point of function $f(\cdot)$, if $f(\mathbf{x}^*) = \mathbf{x}^*$ namely, if the point is mapped to itself. Based on this theory, one can show that the iteration defined by:

$$\mathbf{x}^{(n+1)} = f(\mathbf{x}^{(n)}). \tag{30}$$

converges fixed point of f, $\mathbf{x}^*$ so that $\mathbf{x}^* = f(\mathbf{x}^*)$, under some assumptions. We can use contraction mapping theory in our case, since our function has the structure below:

$$\mathbf{P}^{(n+1)} = \mathbf{G}\mathbf{P}^{(n)} + \delta. \tag{31}$$

Iteration converges to point $\mathbf{P}^*$, the fixed point of $f(\mathbf{P}) = \mathbf{GP} + \boldsymbol{\delta}$, if the eigenvalue of matrix $\mathbf{G}$, $\lambda(\mathbf{G}) < 1$. Then,

$$\mathbf{P}^* = \mathbf{G}\mathbf{P}^* + \boldsymbol{\delta}. \tag{32}$$

Linear Programming Notes

Instructor : Iordanis Koutsopoulos

Lecture 5 : Convexity Definitions - 17/10/06

Notes by : Eleni Anagnostopoulou and Nena Xanthopoulou

1 Outline of lecture:

- Various definitions
  - Convex function and concave function
  - Neighborhood $N(\mathbf{x})$of point $\mathbf{x}$
  - Local and global solutions
  - Gradient and Hessian Matrix
- Convexity Conditions (first and second order)

2 Definitions:

2.1 Convex Function

Given function $f : \Omega \rightarrow \mathcal{R}$, we say that f is *convex* if and only if

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in \ \mathcal{R}^n \text{ and } \theta \in \ [0,1] \tag{1}$$

The schematic representation of a function $f(\mathbf{x})$ of one variable is shown below in Figure 1. In the definition above, $\Omega \subseteq \mathcal{R}$ is the set of points at with $f(\cdot)$ is defined.

Inequality (1) can be interpreted as follows: Given any two points $x, y \in \ \Omega$ with values $f(x), f(y)$ respectively, the chord between points $(x, f(x))$ and $(y, f(y))$ lies above the graph of f (see Figure 1).

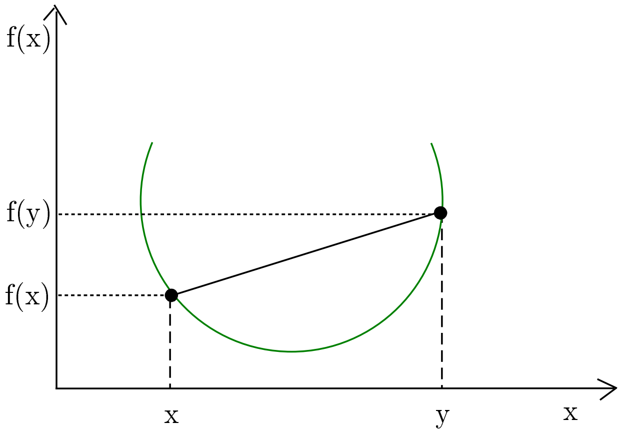


Figure 1: A convex function of one variable, $f(x)$.

2.2 Concave function

Given function $f : \Omega \rightarrow \mathcal{R}$, we say that f is *concave* if and only if

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \geq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in \ \mathcal{R}^n \text{ and } \theta \in \ [0, 1]. \tag{2}$$

The schematic representation of a concave function $f(\mathbf{x})$ of one variable is shown below (Figure 2). Inequality (2) can be interpreted as follows: Given any 2 points $x, y \in \ \Omega$ with values $f(x), f(y)$ respectively, the chord between points $(x, f(x))$ and $(y, f(y))$ lies below the graph of f (see Figure 2).

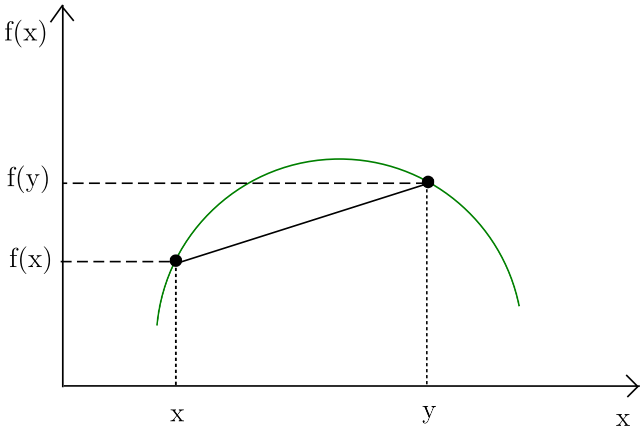


Figure 2: A concave function of one variable, $f(x)$.

Note :

If we have

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) = \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{R}^n \text{ and } \theta \in [0,1], \tag{3}$$

then function f is linear. A linear function can be considered both convex and concave.

2.3 Neighborhood

Given a point $\mathbf{x} \in \mathcal{R}^n$, the set of points $\{\mathbf{y} \in \mathcal{R}^n : \|\mathbf{y} - \mathbf{x}\|_2 < \epsilon\}$ is called *neighborhood* of x and is symbolized as $N(\mathbf{x})$, where $\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T\mathbf{a}}$ is the quadratic norm of vector $\mathbf{a}$ and ϵ is a small positive constant.

Depending on the dimension, n of set $\mathcal{R}^n$, the neighborhood can be as follows:

- In one dimension $(n = 1)$: the neighborhood of $x \in \mathcal{R}$ is shown in Figure 3 and is the set of points $N(x) = \{y : y \in (x - \epsilon, x + \epsilon)\}$.

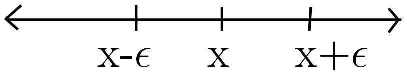


Figure 3: Neighborhood of x in one dimension.

- In two dimensions $(n = 2)$: the neighborhood of $\mathbf{x}$ is a disk centered at $\mathbf{x}$ with radius ϵ and is shown in Figure 4.

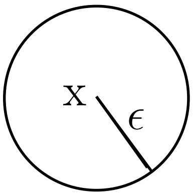


Figure 4: Neighborhood of $\mathbf{x}$ in two dimensions.

- In three dimensions $(n = 3)$: the neighborhood of $\mathbf{x}$ is a sphere centered at $\mathbf{x}$ with radius ϵ.

  ⋮

- In n dimensions $(n > 3)$, the neighborhood of $\mathbf{x}$ is a ”hyper-sphere”.

2.4 Local and global solutions

A feasible solution $\mathbf{x}^*$ is called *local optimum (minimum)* of function f in a neighborhood $N(\mathbf{x}_0)$ of point $\mathbf{x}_0$ if

$$f(\mathbf{x}^*) \le f(\mathbf{x}), \forall \mathbf{x} \in\ N(\mathbf{x}_0). \tag{4}$$

A feasible solution $\mathbf{x}^*$ is called *global optimum (minimum)* of function f if

$$f(\mathbf{x}^*) \le f(\mathbf{x}), \forall \mathbf{x} \in\ \Omega \text{ (definition set of } f). \tag{5}$$

Reminder: The *first derivative* $f'(x_0)$ of function f of one variable at point x_0 is defined as : $\lim_{x \to x_0} \frac{f(x)-f(x_0)}{x-x_0} = f'(x_0)$. For a function $f : \mathcal{R}^n \to \mathcal{R}$ of more than one variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the first derivative of a function of one variable corresponds to the *gradient* of $f(\cdot)$ at point $\mathbf{x}$, it is denoted as *grad*$(f)(\mathbf{x})$ or $\nabla f(\mathbf{x})$ and is defined as the $n \times 1$ vector:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Note that the gradient can be defined at point $\mathbf{x}$ only if all partial derivatives of f at $\mathbf{x}$ with respect to its variables exist.

The i-th component of $\nabla f(\mathbf{x})$ denotes the rate of of f when only variable x_i changes and the others remain fixed.

Example: Given $f(\mathbf{x}) = x^2 + y^2$, calculate the gradient $\nabla f(\mathbf{x})$ at point $\mathbf{x_0} = (1, 2)$.

$$\nabla f(1,2) = \begin{pmatrix} 2x \\ 2y \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}.$$

2.5 Hessian Matrix

The second-order derivative of a function of one variable corresponds to the *Hessian Matrix* of a function of several variables $f : \mathcal{R}^n \to \mathcal{R}$ at point $\mathbf{x}$ and is defined as:

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f^2}{\partial x_1^2}(\mathbf{x}) & \frac{\partial f^2}{\partial x_2 \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f^2}{\partial x_n \partial x_1}(\mathbf{x}) \\ \frac{\partial f^2}{\partial x_1 \partial x_2}(\mathbf{x}) & \frac{\partial f^2}{\partial x_2^2}(\mathbf{x}) & \cdots & \frac{\partial f^2}{\partial x_n \partial x_2}(\mathbf{x}) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f^2}{\partial x_1 \partial x_n}(\mathbf{x}) & \frac{\partial f^2}{\partial x_2 \partial x_n}(\mathbf{x}) & \cdots & \frac{\partial f^2}{\partial x_n^2}(\mathbf{x}) \end{pmatrix} = F(\mathbf{x})$$

If f has continuous second derivatives, then the Hessian Matrix is symmetric, namely $\partial f/\partial x_i \partial x_j = \partial f/\partial x_j \partial x_i, i \neq j$.

Example:

Given $\mathbf{x}$ an $N \times 1$ vector and $\mathbf{A}$ a $N \times N$ matrix, the following properties hold:

1. $\nabla(\mathbf{x}^T\mathbf{A}\mathbf{x}) = (\mathbf{A}+\mathbf{A}^T)\mathbf{x}$, and if $\mathbf{A}$ is symmetric matrix ($\mathbf{A} = \mathbf{A}^T$) , then $\nabla(\mathbf{A}+\mathbf{A}^T)\mathbf{x} = 2\mathbf{A}\mathbf{x}$.
2. $\nabla(\mathbf{y}^T\mathbf{x}) = \mathbf{y}$.
3. $\nabla(\mathbf{x}^T\mathbf{x}) = 2\mathbf{x}$.
4. $\nabla(\mathbf{y}^T\mathbf{A}\mathbf{x}) = \mathbf{A}^T\mathbf{y}$

For example suppose $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$ and $\mathbf{A}$ is 2×2 matrix , then we have:

$$\nabla(x_1 y_1 + x_2 y_2) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{y}.$$

$$\nabla(x_1^2 + x_2^2) = 2\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2\mathbf{x}.$$

3 Convexity Conditions

3.1 First-order Convexity Conditions

If the first-order derivatives of $f(\mathbf{x})$ exist, then

$$f \text{ is convex } \Leftrightarrow f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y}-\mathbf{x}), \forall \mathbf{x}, \mathbf{y} \in \mathcal{R}^n. \tag{6}$$

It is useful to note that the right part of inequality (6) consists of the first two terms of Taylor's expansion formula of $f(\mathbf{y})$ around point $\mathbf{x}$.

$$\text{Taylor's formula} : f(\mathbf{y}) = f(\mathbf{x}) + \nabla^T(f(\mathbf{x}))(\mathbf{y}-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^T \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) + \dots \tag{7}$$

The schematic representation is shown below for a function of one variable (Figure 5). The equation of the tangent line of f at point x is $f(x) + f'(x)(y-x)$. For more than one dimensions, the equation of the tangent plane of f at point $\mathbf{x}$ is $f(\mathbf{x}) + \nabla^T f(\mathbf{x})(\mathbf{y}-\mathbf{x})$.

Note : Generally we could say that because f is convex, we can use a local information (the right term of inequality (7), $f(\mathbf{x}) + \nabla^T(f(\mathbf{x}))(\mathbf{y}-\mathbf{x})$, which holds "locally" around $\mathbf{x}$, in order to

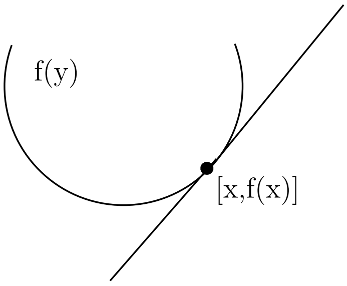


Figure 5: Tangent line of f at point x.

obtain a global information (a global lower bound on $f(\mathbf{y})$. Thus, from local information (a value and a derivative of a convex function at a given point $\mathbf{x}$) we can derive global information (global lower bound).

If $\nabla f(\mathbf{x}^*) = 0$ for point $\mathbf{x}^*$ then from inequality (6) we have : $f(\mathbf{y}) \geq f(\mathbf{x}^*), \forall \mathbf{y} \in \mathcal{R}^n \Rightarrow \mathbf{x}^*$ is global optimum (minimum).

Note: If function f depends on *only one variable* then the inequality (6) can be written as:

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0). \tag{8}$$

3.2 Second-order Convexity Conditions

Suppose f is twice differentiable (all second derivatives exist). Then, f is convex if and only if

$$\nabla^2 f(\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in \Omega \subseteq \mathcal{R}^n. \tag{9}$$

That is to say the Hessian matrix is positive semi-definite, symbolized as $\nabla^2 f(\mathbf{x}) \geq 0$.

Note : A matrix $\mathbf{A}$ is positive semi-definite ($\mathbf{A} > 0$) if and only if : $\mathbf{x}^T A\mathbf{x} \ \forall \mathbf{x} \in \ \mathcal{R}^n$. If $\mathbf{A} > 0$ then $\mathbf{A}$ has all its eigenvalues positive.

If function f depends on *only one variable* then condition (9) is written as:

$$f(x) \text{ convex } \text{ if and only if } f''(x) \geq 0. \tag{10}$$

Similarly a function f is concave if and only if $\nabla^2 f(\mathbf{x}) \leq 0 \ \forall \mathbf{x} \in \Omega \subseteq \mathcal{R}^n$.

Examples of convex and concave functions (of one variable):

1.

$$f(x) = e^{\alpha x} \text{ is convex } \text{ on } \mathcal{R}, \forall \alpha \in \mathcal{R}.$$

2.

$$f(x) = x^{\alpha} = \begin{cases} \text{ convex on } \ R_{+} \ \text{ if } \ a \geq 1 \text{ and } a \leq 0, \\ \text{concave if } \ a \in [0,1] \end{cases}$$

3.

$$f(x) = x^{\alpha} = \begin{cases} \text{concave } \ \text{on } R_{-} \ \text{ if } \ a \geq 1 \ \text{ and } a \leq 0, \\ \text{convex } \ \text{if } a \in [0,1] \end{cases}$$

4.

$$f(x) = \log x \ \text{ is concave on } R_{+}.$$

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 6 : 18/10/06

Notes by : Evagelos Galanis and Panagiotis Theodosiou

1 Outline Of Lecture:

- Utility functions
- Convex and concave functions of several variables
- Jensen's Inequality
- Feasible Directions
- Conditions for existence of minimum

2 From previous lecture: Utility function

Given the function of one variable $f(x) = \log x$, we take its second derivative, $f''(x) = -\frac{1}{x^2} < 0$, $\forall x > 0$. Thus, $f(\cdot)$ is a concave function. The log-function is useful in defining capacity and solving relevant problems.

Definition of *Capacity*(Information-theoretic) : the largest number of bits per second that can be transmitted over a link with arbitrarily small probability of error,

$$C(P) = \log(1 + \text{SNR}) = \log(1 + \frac{P}{N}), \tag{1}$$

where P is the transmission power and N is the noise power. Function $C(P)$ is a concave function of P and is depicted in figure 1.

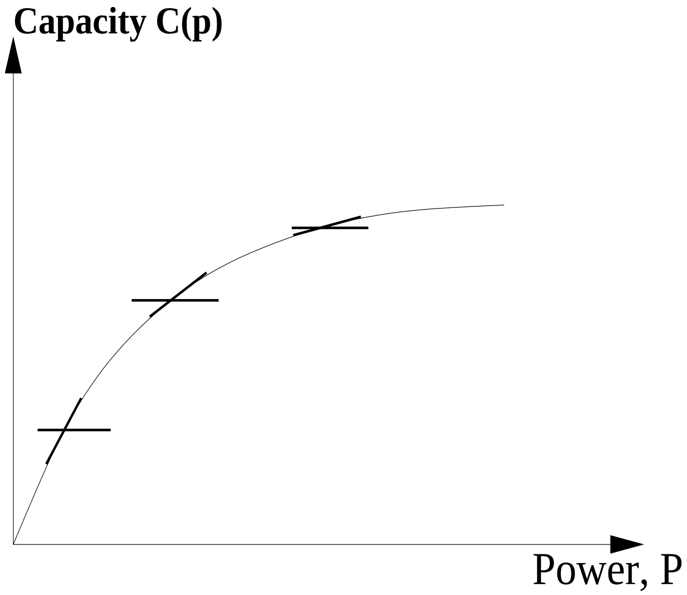


Figure 1: Capacity Function.

The first derivative in a point x of the curve gives the slope of the tangent line at x. Since $f''(x) < 0$, then $f'(x)$ is decreasing ($\downarrow$). So, as power$P$ increases, the rate of increase of capacity dC/dP decreases (this attribute holds only for concave functions) as figure 1 shows. For large enough values of power, the capacity is "saturated", in the sense that there are marginal returns. Thus, the higher the power, the smaller the rate of increase of capacity. We have the following approximations:

- For large power P, $C(P) = \log(1 + \frac{P}{N}) \approx \log \frac{P}{N}$
- For small P, $C(P) \approx \frac{P}{N}$, since $\log(1 + x) \approx x$ for small x.

A similar phenomenon is observed if we define the more general notion of utility function (used e.g. in Pricing theory of Networks), which is depicted in figure 2.

Function $U(x)$ denotes the amount of satisfaction of a user as a function of the amount of allocated resources to him. Resources can be power of bandwidth.

3 Convex functions of several variables

- $f(\mathbf{x}) = \sqrt{\mathbf{x}^T\mathbf{x}} = ||\mathbf{x}||_2$ is a convex function of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
- $f(\mathbf{x}) = \max\{x_1, x_2, \ldots, x_n\}$ is a convex function of $\mathbf{x} = (x_1, \ldots, x_n)$.

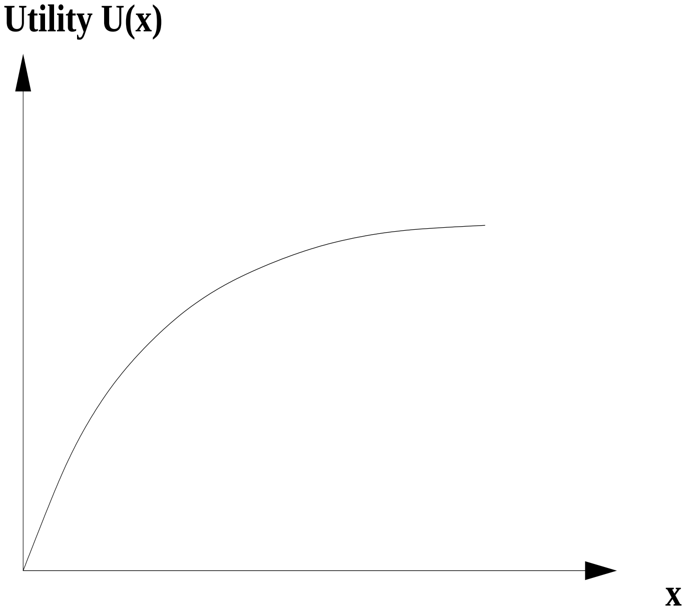


Figure 2: Utility Function.

- $f(\mathbf{x}) = \log(e^{x_1} + \ldots + e^{x_n})$ is a convex function of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ (although $f(\mathbf{x}) = \log(\mathbf{x})$ is a concave function of $\mathbf{x}$ in $\mathbb{R}_+$).

  Note: We will use this later to show that function

  $$f(\mathbf{P}) = \sum_{i=1}^{N} q_i \log \frac{G_{ii}P_i}{\sum\limits_{j\neq i} G_{ji}P_j} = \sum_{i=1}^{N} q_i \log(\mathrm{SIR}_i(\mathbf{P}) \tag{2}$$

  is a concave function of $\mathbf{P}$, where SIR is the signal-to-interference ratio. Then, we will solve the corresponding maximization problem.

- $f(\mathbf{x}) = \left(\prod_{i=1}^{n} x_i\right)^{1/n}$ is a concave function of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in $\mathbb{R}_+$.

4 Jensen's Inequality

If f is a convex function then

$$f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{f(\mathbf{x}) + f(\mathbf{y})}{2} \tag{3}$$

If f is a convex function and X is a random variable, we can generalize the above to show that $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.

Sketch of Jensen's Inequality: First, we use the definition of convex function f for two points x, y : $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$.

Then extend the inequality to more points: $x_1, x_2, \ldots, x_k$ as

$$f(\theta_1 x_1 + \ldots + \theta_k x_k) \le \theta_1 f(x_1) + \ldots + \theta_k f(x_k), \tag{4}$$

where $\sum_{i=1}^{k} \theta_i = 1,\ \forall\ \theta_i \ge 0$ (convex combination of k discrete points).

Considering that $\int_S p(x)dx = 1$, where S is the set of points where f is defined, the inequality for continuous set of points becomes

$$f\left(\int_S p(x)x dx\right) \le \int_S f(x)p(x)dx. \tag{5}$$

Thus the probability distribution $p(\cdot)$ is "similar" to a continuous distribution of θ's and declares the convex combinations. Eventually, we conclude that $f(\mathbb{E}[X]) \le \mathbb{E}[f(X)]$.

Notation: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Omega} f(x) \iff f(\mathbf{x}^*) \le f(\mathbf{x})\forall\mathbf{x} \in \Omega$. ("arg" stands for argument of a function)

5 Feasible Directions

Suppose that we have a convex set Ω as in figure 3. Assume that $\mathbf{x} \in \Omega$ (Omega in the figure) is a feasible point. A feasible direction $\mathbf{d}$ at a point $\mathbf{x}$ is a vector $\mathbf{d} \ne 0$ such that $\mathbf{x} + \alpha\mathbf{d}$ is feasible $\forall\, \alpha > 0$ which are small enough. Given a point $\mathbf{x}$, all the directions towards which we can move from $\mathbf{x}$ constitute the set of feasible directions. We show some of these directions in figure 2.

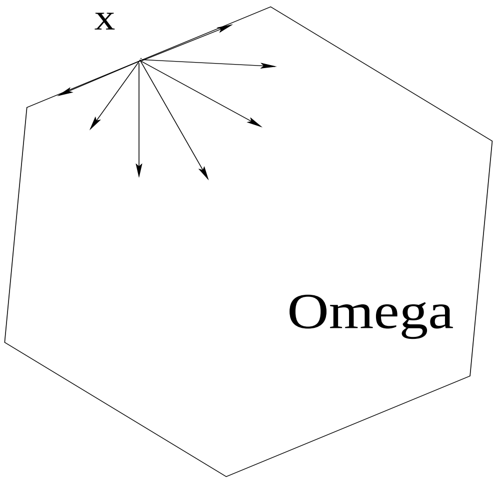


Figure 3: Feasible Direction.

Equivalently, we say that a vector $\mathbf{d} \neq 0$ is a feasible direction $\mathbf{x} \in \Omega$ if $\exists\ \alpha_0 > 0 :\ \mathbf{x} + \alpha \mathbf{d} \in \Omega\ \forall\ \alpha \in [0, \alpha_0]$.

In Linear Programming, we will see that Ω will be polyhedron that will arise from the constraints of the problem, the possible solutions are the vertices of the polyhedron and the feasible directions are the edges of the polyhedron.

We can imagine we stand in a point $\mathbf{x}$ and choose to move towards a direction $\Delta\mathbf{x}$. The deference between the value of f at the new point $\mathbf{x} + \Delta\mathbf{x}$ minus the value of f at the old point $\mathbf{x}$ can be approximated as follows: (note $\Delta\mathbf{x} = (\Delta x_1, \Delta x_2, \ldots, \Delta x_N)$

1) First Order Approximation : $f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x}) \approx \nabla^T f(\mathbf{x})\ \Delta\mathbf{x} = \sum_{i=1}^{N} \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \Delta\mathbf{x}_i$. For example, for direction $\mathbf{d}$ such that $\mathbf{d} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, with $||\mathbf{d}|| = 1$, this means that we move in a direction that forms angle of 45° with the tangent line of f at $\mathbf{x}$. In general, we can move with a step ε towards direction $\mathbf{d}$.

2) Second Order Approximation : $f(\mathbf{x} + \Delta\mathbf{x}) - f(\mathbf{x}) \approx \nabla^T f(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2}(\Delta\mathbf{x})^T \nabla^2 f(\mathbf{x})(\Delta\mathbf{x})$ (we have used Taylor's Theorem).

6 Conditions for existence of minimum point

6.1 First Order - Necessary Condition

Assume $\mathbf{x}^*$ is local minimum of function f. By using the first order approximation we have that $f(\mathbf{x}^* + \Delta\mathbf{x}) - f(\mathbf{x}^*) \approx \nabla^T f(\mathbf{x}^*)\ \Delta\mathbf{x} \geq 0\ \forall \Delta\mathbf{x}$. This means that $\sum_{i=1}^{N} \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \Delta x_i \geq 0$ for all $\Delta\mathbf{x} = (\Delta x_1, \ldots, \Delta x_N)$. Clearly, the value of function f increases wherever we move to, since currently we are at $\mathbf{x}^*$, the minimum point. We want to prove that if $\mathbf{x}^*$ is a local minimum of function f, then the gradient of f is 0 ($\nabla f(\mathbf{x}^*) = 0$). Take $\Delta\mathbf{x}$ to be positive and negative multiples of the unit coordinate vectors $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)$. The positive ones are: $\Delta\mathbf{x}^{(1)} = \varepsilon(1, 0, \ldots, 0)$
$\Delta\mathbf{x}^{(2)} = \varepsilon(0, 1, \ldots, 0)$
$\vdots$
$\Delta\mathbf{x}^{(n)} = \varepsilon(0, 0, \ldots, 1)$
and the negative ones are :
$\Delta\mathbf{x}^{(1)} = -\varepsilon(1, 0, \ldots, 0)$
$\Delta\mathbf{x}^{(2)} = -\varepsilon(0, 1, \ldots, 0)$

$\vdots$

$\Delta \mathbf{x}^{(n)} = -\varepsilon(0, 0, \ldots, 1)$.

Since $\sum_{i=1}^{N} \frac{\partial f(\mathbf{x}^*)}{\partial x_i} \Delta x_i \geq 0$ for all $\Delta \mathbf{x} = (\Delta x_1, \ldots, \Delta x_N)$, we have for example for x_1:

$$\frac{\partial f(\mathbf{x}^*)}{\partial x_1} \varepsilon \geq 0 \text{ and } \frac{\partial f(\mathbf{x}^*)}{\partial x_1} \varepsilon \leq 0. \tag{6}$$

These lead us to $\frac{\partial f(\mathbf{x}^*)}{\partial x_1} \varepsilon = 0$. Similarly for the rest of x_i, $i = 2, \ldots, N$, we have: $\frac{\partial f(\mathbf{x}^*)}{\partial x_i} \varepsilon = 0$ and finally, since $\varepsilon > 0$, we get $\frac{\partial f(\mathbf{x}^*)}{\partial x_i} = 0 \; \forall i$. So we have proved the following:

If $\mathbf{x}^*$ is a local minimum of function f, then $\nabla f(\mathbf{x}^*) = 0$ (assuming that function f has first derivative at $\mathbf{x}^*$). This is the First Order Necessary Condition for existence of local minimum.

6.2 Second Order Necessary Condition

From the Second Order Taylor approximation we saw: If $\mathbf{x}^*$ is a local minimum, then $f(\mathbf{x} + \Delta \mathbf{x}) - f(\mathbf{x}) \approx \nabla^T f(\mathbf{x}) \Delta \mathbf{x} + \frac{1}{2} (\Delta \mathbf{x})^T \nabla^2 f(\mathbf{x}) (\Delta \mathbf{x}) \geq 0$. Since $\nabla^T f(\mathbf{x}^*) = 0$, the above becomes: $f(\mathbf{x} + \Delta \mathbf{x}) - f(\mathbf{x}) \approx \frac{1}{2} (\Delta \mathbf{x})^T \nabla^2 f(\mathbf{x}) (\Delta \mathbf{x}) \geq 0 \, \forall \Delta \mathbf{x}$.

Thus, we proved that if $\mathbf{x}^*$ is local minimum of function f, then $\nabla^2 f(\mathbf{x}^*) \geq 0$ (assuming that f has all partial second derivatives at $\mathbf{x}$). This means that if $\mathbf{x}^*$ is a local minimum, then the Hessian matrix of f at $\mathbf{x}^*$, $\nabla^2 f(\mathbf{x}^*)$ is positive semi-definite.

6.3 Sufficient Conditions

Assume that function $f : \mathbb{R}^n \;\; \rightarrow \mathbb{R}^n$ has two derivatives. If $\mathbf{x}^* \in \Omega$ satisfies $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*) > 0$, then $\mathbf{x}^*$ is a local minimum of function f in Ω.

Similarly, for a local maximum, we can prove that if $\mathbf{x}^* \in \Omega$ satisfies $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*) < 0$, then $\mathbf{x}^*$ is a local maximum of function f in Ω.

Note: If f is convex function, every local minimum is also global minimum, while if f is a concave function every local maximum is also global maximum.

Remark: Necessary and sufficient condition meaning:

N is necessary condition for A or $A \Rightarrow N$.

S is sufficient condition for A or $S \Rightarrow A$.

For a pictorial representation, see figure 4.

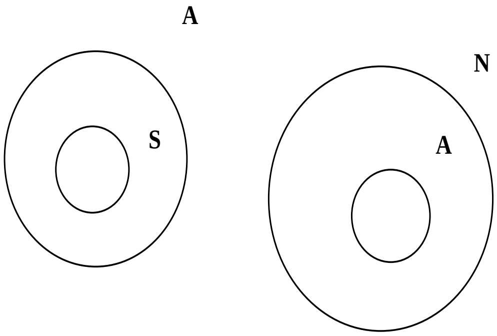


Figure 4: Necessary and Sufficient conditions.

Exercise: We are given a set of vectors $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}\}$, with $\mathbf{x}^{(i)} \in \mathbb{R}^n$ for $i = 1, \ldots, p$. Find the vector $\mathbf{x} \in \mathbb{R}^n$ such that the average squared distance (norm) between $\mathbf{x}$ and all $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, given by

$$\frac{1}{p}\sum_{i=1}^{p} ||\mathbf{x} - \mathbf{x}^{(i)}||^2 \tag{7}$$

is minimized. Is the local minimum a global minimum as well?

The solution to the above is

$$\mathbf{x}^* = \frac{1}{p}\sum_{i=1}^{p} \mathbf{x}^{(i)}. \tag{8}$$

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 7 : Minimum of functions of one variable and examples of Convex Optimization problems 24/10/06

Notes by : Katerina Mamoura and Eleana Parlavantza

1 Outline of lecture 7:

- Methods for minimization:
  - Bisection method
  - A direct method for finding the minimum
  - Newton's method
- Two examples of Convex Optimization Problems.

2 Finding the minimum of functions of one variable

2.1 First method

- Given a function $f : \Re \rightarrow \Re$ that has a first derivative, a point $x^\star$ that minimizes $f(x)$ has the property that:

  $$f'(x^\star) = 0. \tag{1}$$

  Define $g(x) = f'(x)$. Then, we need to find a point $x^\star$ such that

  $$g(x^\star) = 0, \tag{2}$$

  i.e. a root of $g(\cdot)$. In order to find $x^\star$, we can use the *bisection* method.

The pseudo-algorithm for the bisection method is the following:

STEP 1: Find two points a, b such that $g(a) \cdot g(b) < 0$ (this means that $g(a)$, $g(b)$ have opposite signs. Otherwise, function $g(\cdot)$ is increasing, or decreasing and the minimum within interval $[a, b]$ coincides with one of the two end points.

STEP 2: Compute intermediate point $y = \frac{b-a}{2} + a$.

STEP 3:

(a) If $g(a) \cdot g(y) < 0$, then set $b = y$,

(b) If $g(y) \cdot g(b) < 0$, then set $a = y$,

(c) If $g(y) \cdot g(a) = 0$ or $g(y) \cdot g(b) = 0$, then y is the minimum. **STOP**.

STEP 4: If $|b - a| < \delta$ (where $\delta << 1$) **STOP**.

`Note` : If we search for the minimum in a closed interval $[c, d]$, the minimum is either c or d, depending on whether f is increasing or decreasing.

In the algorithm above, either the root is found (hit exactly) or the searching range is progressively narrowed, until the minimizer is found with some specified accuracy δ.

The first step of the algorithm is depicted in figure 1.

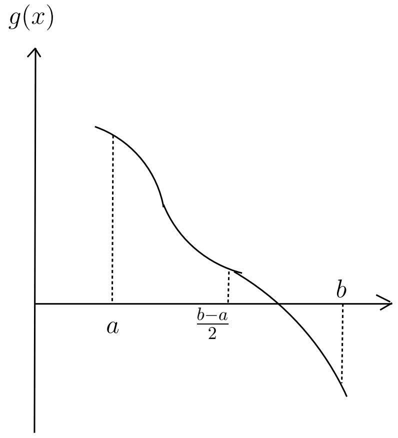


Figure 1: The first step of bisection method.

Algorithm Complexity : It takes at most $\log_2 \frac{|b-a|}{\delta}$ steps to find the minimum. Thus, the algorithm complexity depends on the initial search range $[a, b]$ and the tolerance parameter δ.

2.2 Second method

In the second method, we attempt to minimize function $f : \Re \rightarrow \Re$ directly, namely without finding the root of the derivative of f.

Definition of Unimodal function : Given a function $f : \Re \rightarrow \Re$ defined in a closed interval $[a, b]$, $f : [a, b] \subset \Re$, f is called *unimodal* in $[a, b]$ if f has only one local minimizer in $[a, b]$. Specifically, f is unimodal if, given a $x^\star \in [a, b]$, f is increasing for $x \geq x^\star$ and decreasing for $x \leq x^\star$, $x \in [a, b]$.

Conditions that are satisfied by unimodal functions: Given points x_1, x_2 such that $a \leq x_1 < x_2 \leq b$, there exists a point $x^\star$ that:

(a) If $x_1 > x^\star$, then $f(x_1) < f(x_2)$. Thus, f is decreasing, as x moves from x_1 towards $x^\star$ (Figure 2).

(b) If $x_2 < x^\star$, then $f(x_1) > f(x_2)$. Thus, f is increasing, as x moves from $x^\star$ towards x_2 (Figure 3).

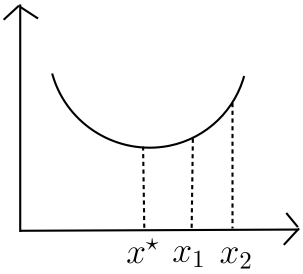


Figure 2: Conditions satisfied by a unimodal function.

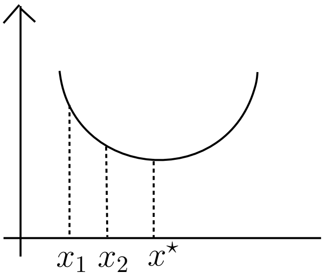


Figure 3: Conditions satisfied by a unimodal function.

Note that The bisection method cannot be used in this case, because we cannot make any conclusions about the sign of the intermediate point. For example f can have one of the two forms shown in figure 4, and the minimum can be either in the first interval or in the second interval.

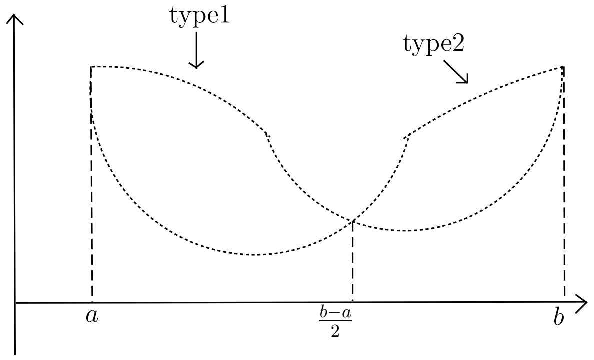


Figure 4: Bisection method cannot give any hint on where the minimum is.

Thus, we proceed as follows: We find two points $\hat{x}_1, \hat{x}_2 \in [a, b]$ such that $a < \hat{x}_1 < \hat{x}_2 < b$ and we find $f(\hat{x}_1), f(\hat{x}_2)$. Then, we can distinguish the following cases.

(I) If $f(a) > f(\hat{x}_1) > f(\hat{x}_2)$, the interval $[a, \hat{x}_1]$ is excluded, as there is no way the minimum is located in that interval. Thus, we set the right point of searching interval $\hat{x}_1 \leftarrow a$. The minimum should be somewhere in $[\hat{x_1}, b]$. This case is shown is figure 5.

(II) If $f(b) > f(\hat{x}_2) > f(\hat{x}_1)$, the interval $[\hat{x}_2, b]$ is being excluded, as there is no way the minimum is located in that interval. The minimum should be somewhere in $[a, \hat{x}_2]$. Thus, we set the left point of searching interval $\hat{x}_2 \leftarrow b$. This case is shown is figure 6.

We proceed in that fashion, until we find the minimum $x^\star$ with some accuracy. The question that arises is the following: *How do we choose the points* $\hat{x}_1, \hat{x}_2$ in each iteration? We need to choose them so that the number of iterations and the searching interval at every step are reduced fast. The following methods exist for choosing $\hat{x}_1, \hat{x}_2$: Golden ratio search and Fibbonacci search.

2.3 Third method - Newton's method

Newton's method is an iterative method which uses the second derivative of f. Thus, it assumes that f is twice differentiable.

Given a function $f(x)$ and a point $x^{(k)}$ the idea is to approximate $f(x)$ through a quadratic function, namely a second degree $q(x)$. Instead of minimizing f we attempt to minimize its approximation

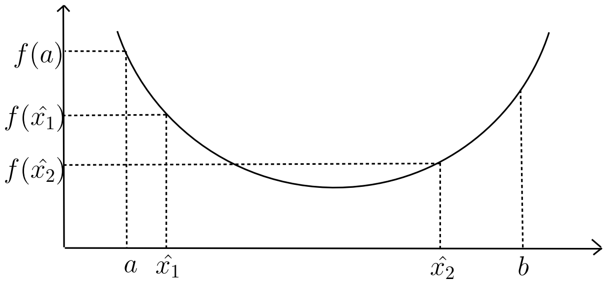


Figure 5: Case II in the method of directly finding the minimum.

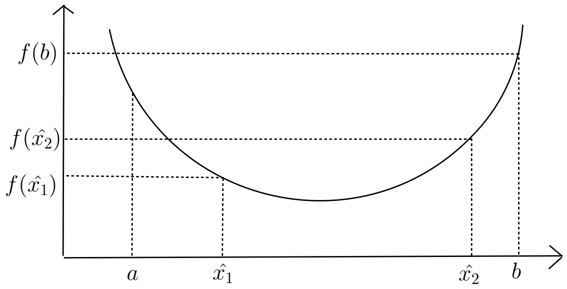


Figure 6: Case I in the method of directly finding the minimum.

$q(x)$, which has the form:

$$q(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2}f''(x^{(k)})(x - x^{(k)})^2 \tag{3}$$

Note that approximation $f(x) \approx q(x)$ is selected so that $q(x)$ satisfies:

$$q(x^{(k)}) = f(x^{(k)}), \tag{4}$$

$$q'(x^{(k)}) = f'(x^{(k)}), \tag{5}$$

$$q''(x^{(k)}) = f''(x^{(k)}). \tag{6}$$

By minimizing function $q(\cdot)$ we get: $q'(x) = 0 \Rightarrow f'(x^{(k)}) + f''(x^{(k)})(x - x^{(k)}) = 0$. Solving this equation to find the x that minimizes f,we get:

$$x = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})} \tag{7}$$

By setting the next point to be that minimizing x, i.e, $x \leftarrow x^{(k+1)}$ the equation above becomes:

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})} \tag{8}$$

and gives the form of iteration of Newton method.

Note that Newton method is a special form of gradient method.

- This iterative method, starts from an initial point x_0 and terminates either if $f'(x^{(k)}) = 0$ (in which case $x^{(k+n)} = x^{(k)}$, for $n > 0$) or when $|x^{(k+1)} - x^{(k)}| < \varepsilon$
- In the stopping condition $|x^{(k+1)} - x^{(k)}| < \varepsilon$, there exists a tradeoff: if ε is too small, then the result is more accurate, but it takes more iterations to reach that. On the other hand, if ε is larger, then, despite the fact that the result is found faster, the error is also larger.
- For functions of several variables, we will see that the Newton iteration becomes:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\nabla^2 f(x^{(k)})]^{-1} \nabla f(\mathbf{x}^{(k)}) \tag{9}$$

  where $\nabla^2 f(\mathbf{x}^{(k)})$ is the Hessian matrix of f at point $\mathbf{x}^{(k)}$.

3 Examples of Convex Optimization Problems

3.1 Routing with transmission rate control

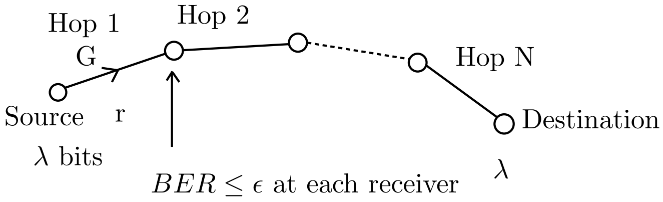


Fig.3 : Multi-hop transmission in a sensor network

Figure 7: Bit transfer from a source to a destination in multi-hop in a wireless sensor network.

Consider a sensor network, in which λ bits have to be transmitted from a source (S) to a destination (D) which can be a processing center. The λ bits come as the output of sensing and measurement at the sensor in (S). So, in each hop in the path from S to D, there happens a transmission from the transmitter to the corresponding receiver.

There exist L possible data transmission rates, $r \in \{r_1, ...r_L\}$, and each transmitter has to choose an $r_i \in \{r_1, ...r_L\}$.

The λ bits need to be transferred within a certain period of time. Then, λ can be written as function of the transmission rate as follows

$$\lambda = \frac{\text{bits}}{\text{symbol}} \cdot \frac{\text{symbols}}{\text{sec}} \cdot \text{sec} \tag{10}$$

$$\Rightarrow \lambda = r \cdot s \cdot t \tag{11}$$

where s is the symbol rate $(\frac{symbols}{sec})$ and $s = \frac{1}{T}$, where T is the symbol (pulse) duration. Bits are transferred on symbol pulses (which can be square pulses or sigmoid ones). The rate r determines the number of transmitted bits per symbol, namely it determines the modulation level. Assume t is a fixed known time interval.

In this problem, the modulation level for the transmitter of each hop is controllable. At each receiver along the S-to-D path, the SNR should satisfy:

$$\text{SNR} \geq -\frac{\ln(5\epsilon)}{1.5}(2^r - 1) \tag{12}$$

if rate r is used at the transmitter, so that BER $\leq \epsilon$ at each receiver, where ϵ is a fixed number (e.g.$10^{-6}, 10^{-7}$) that shows the maximum tolerable bit error rate (BER) at each transmission hop. The inequality above results from the following approximate formula for BER:

$$\text{BER} \approx \frac{1}{5} e^{-\frac{1,5 \cdot SNR}{2^r - 1}} \tag{13}$$

The receiver is aware of transmitter rate and tries to decode the signal according to that. If r is too large, this means that the signal points in the constellation diagram are too close to each other, so there is a difficulty in distinguishing what has been sent. Thus, in order to reduce the number of errors, a better channel is needed.

The SNR is related to transmission power as follows:

$$\text{SNR} = \frac{G \cdot P}{\sigma^2} \tag{14}$$

where G is the link gain between transmitter-receiver and σ^2 is the noise power at the receiver. From (12) and (14) by using equality and solving for P we have the following:

$$P \approx k \cdot 2^r \tag{15}$$

where k is a proportionality constant. Thus P is an exponential function of rate r. The transmission energy at some hop is given by:

$$\text{Energy} = \text{Power} \times \text{time} \tag{16}$$

From equation (10) the energy is :

$$E = P \cdot \frac{\lambda}{s} r \tag{17}$$

$$\Rightarrow \qquad E = \frac{k \cdot 2^r}{\frac{sr}{\lambda}}$$

$$\Rightarrow \qquad E = k \cdot 2^r \cdot \frac{\lambda}{sr}$$

Thus, for example for hop 1 (h_1) we have from above that the consumed energy for transmission is:

$$\Rightarrow \qquad E_1 \approx k_1 \cdot \frac{2^{r_1}}{r_1}$$

and in general for hop i, $i = 1, \ldots, N$:

$$\Rightarrow \qquad E_i \approx k_i \cdot \frac{2^{r_i}}{r_i} \tag{18}$$

where k_i is a constant and r_i is the transmission rate at hop i (transmitter i to receiver i).

What is the problem that arises ?

(a) Given a route from S to D, what is the transmission rate r_i in each hop (h_i) so that the total energy $\sum_{i=1}^{N} E_i$ is minimized.

If there are no constraints on the time by which the bits need to be transferred and since

$$E_i \approx k_i \cdot \frac{2^{r_i}}{r_i} \tag{19}$$

and there are L choices for the rate, $r \in \{b_1, ...b_L\}$ with $b_1 < b_2 < ... < b_L$, the optimal choice is to operate with the minimum rate for each hop, namely $\forall$ hops i, choose $r_i = b_i$.

(b) *What happens when there is a deadline by which bits need to be transferred to the destination?* Then λ bits need to be transferred from source to destination within some deadline time T. We also make the assumption that only one hop is active at a time, the one that transmits. Is the solution still the same?

There are two conflicting arguments : The sum $\sum_{i=1}^{N} E_i$ needs to be low (because in a sensor network battery consumption has to be low). That means that energy transmission in each hop has to be low. Thus,in order to achieve low energy consumption, each r_i has to be low. But if each r_i is low then λ bits are transmitted in a longer period of time. Hence $\frac{\lambda}{sr_i}$ increases and it is very likely for the end-to-end transmission to last longer and not satisfy the deadline T.

Therefore, we can formulate the following optimization problem (assuming that the rates are continuous variables):

$$\min_{\mathbf{r}} \sum_{i=1}^{N} k_i \cdot \frac{2^{r_i}}{r_i},$$

$$\text{subject to the constraint: } \sum_{i=1}^{N} \frac{1}{r_i} \leq \frac{T_s}{\lambda}$$

with $\mathbf{r} = \{r_1, ... r_N\}$.

Objective function $f(\mathbf{r}) = \sum_{i=1}^{N} k_i \frac{2^{r_i}}{r_i}$ is convex in $\mathbf{r}$, since it is the sum of convex functions of the form $\frac{e^x}{x}$. Also the constraints are convex. Therefore, we have a convex optimization problem.

3.2 Convexity issues in an M/M/1 queue

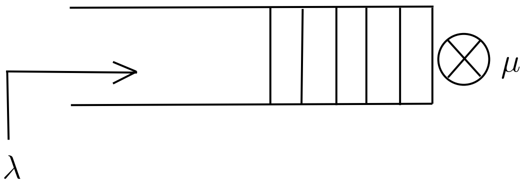


Fig.4 : M/M/1 queue

Figure 8: An M/M/1 queue.

In the depicted M/M/1 queue let λ be the average customer arrival rate (in customers/sec) and μ be the customer service rate.

From Little's theorem for the average number of customers N in the system we have $N = \lambda \cdot T$, where T is the average waiting time in the queue for a customer and $T = \frac{1}{\mu - \lambda}$ for an M/M/1 queue.

Is function $N = \frac{\lambda}{\mu - \lambda}$ *convex or concave ?* For fixed λ, it is $N''(\mu) > 0$. Thus, for fixed λ, function $N(\cdot)$ is convex for μ. For fixed μ, it is $N''(\lambda) < 0$, thus function $N(\cdot)$ is concave for λ.

Advanced Topics in Networking - Fall 2006

Instructor: Iordanis Koutsopoulos

Lecture 8 : Gradient Methods and Gradient Descent Method - 25/10/06

Notes by : Nikolaos Kapetanios and Konstantinos Sotiropoulos

1 Gradient Methods

The level set of a function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$ at level c is the set of points

$$\mathcal{S} = \{\mathbf{x} : f(\mathbf{x}) = c\} \tag{1}$$

If a point $\mathbf{x}_0$ is on the level set $\mathcal{S}$ at level c, then $f(\mathbf{x}_0) = c$.

Fact: Assuming that f is continuously differentiable, the vector $\nabla f(\mathbf{x}_0)$ is orthogonal to the tangent vector to an arbitrary smooth curve passing through $\mathbf{x}_0$ at the level set $f(\mathbf{x}) = f(\mathbf{x}_0)$. A curve lying on $\mathcal{S}$ can be parameterized by a continuously differentiable function $\mathbf{x} : \mathcal{R} \longrightarrow \mathcal{R}^n$.

Definition of a curve: A curve γ on $\mathcal{S}$ is defined as $\gamma = \{\mathbf{x}(t) : t \in (a, b)\} \subseteq \mathcal{S}$, where a denotes the start and b denotes the end of the curve. The beginning and ending points of the curve are $\mathbf{x}(a)$ and $\mathbf{x}(b)$. Now suppose also that $\mathbf{x}(t_0) = \mathbf{x}_0$ and $\mathbf{x}'(t_0) \neq \mathbf{0}$ is the tangent vector to curve γ at $\mathbf{x}_0$ (see Figure 1). We will show:

$$\nabla f(\mathbf{x}_0) \perp \mathbf{x}'(t_0). \tag{2}$$

Proof: Let $\mathbf{g}(t) = f(\mathbf{x}(t))$ with $t, t_0 \in (a, b)$ and $\mathbf{x}_0 = \mathbf{x}(t_0)$. From the level set of f we have : $f(\mathbf{x}(t)) = c$.

$$\frac{df(\mathbf{x}(t))}{dt} = 0 \Leftrightarrow \nabla_{\mathbf{x}}^T f(\mathbf{x}(t))\ \mathbf{x}'(t) = 0 \text{ from the chain rule} \tag{3}$$

and for $t = t_0$ we have,

$$\nabla_{\mathbf{x}}^T f(\mathbf{x}(t_0))\ \mathbf{x}'(t_0) = 0. \tag{4}$$

The direction of $\nabla f(\mathbf{x})$: $\nabla f(\mathbf{x})$ actually is always the direction of maximum rate of increase of f at point $\mathbf{x}$.

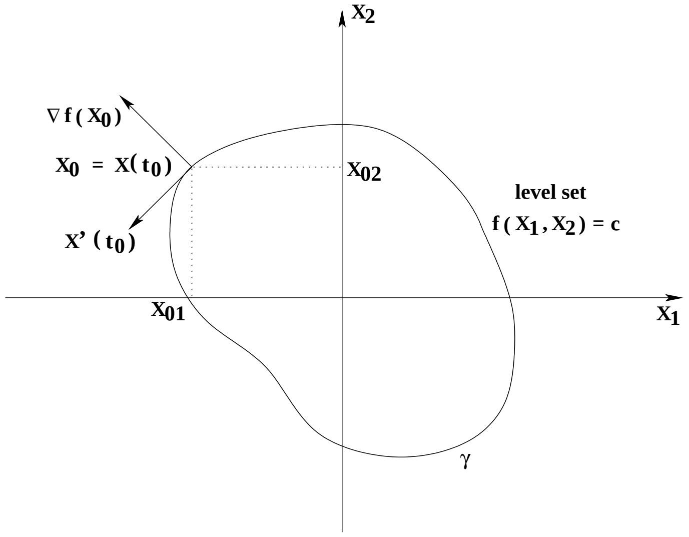


Figure 1: Orthogonality of the gradient to the level set

Directional derivative: Define as $\frac{\partial f(\mathbf{x}}{\partial \mathbf{d}}$ the directional derivative of function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$ in the direction $\mathbf{d}$ at point $\mathbf{x}$, as

$$\frac{\partial f}{\partial \mathbf{d}} = \lim_{\alpha \to 0} \frac{f(\mathbf{x} + \alpha\, \mathbf{d}) - f(\mathbf{x})}{\alpha} = \mathbf{d}^T \nabla f(\mathbf{x}). \tag{5}$$

If $||\mathbf{d}|| = 1$, then $\frac{\partial f}{\partial \mathbf{d}}(\mathbf{x})$ is the rate of increase of f at direction $\mathbf{d}$ at point $\mathbf{x}$. If we apply the Cauchy-Schwartz inequality for vectors $\mathbf{a}$ and $\mathbf{b}$: $((\mathbf{a}^T\mathbf{a})^2 \leq (\mathbf{a}^T\mathbf{a})(\mathbf{b}^T\mathbf{b}) = ||\mathbf{a}||^2||\mathbf{b}||^2)$ we have that:

$$(\mathbf{d}^T \nabla f(\mathbf{x}))^2 \leq ||\mathbf{d}||^2 \ ||\nabla f(\mathbf{x})||^2 \Rightarrow \mathbf{d}^T \nabla f(\mathbf{x}) \leq ||\nabla f(\mathbf{x})||. \tag{6}$$

Thus the maximum value of the left side of expression (6) is $||\nabla f(\mathbf{x})||$. For direction

$$\mathbf{d} = \frac{\nabla f(\mathbf{x})}{||\nabla f(\mathbf{x})||}, \tag{7}$$

namely the unit-norm direction of gradient, we get that $\frac{\partial f}{\partial \mathbf{d}} = ||\nabla f(\mathbf{x})||$, i.e, the maximum value of rate of increase.

Note: The amount of increase of f to the direction $\mathbf{d}$ is $\alpha \ \frac{\partial f}{\partial \mathbf{d}}$.

A simple problem: Suppose that we have function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$ and we are looking at the maximum (or minimum) value $p^* = f(\mathbf{x}^*)$. The solution to this problem is given by solving equation $\nabla f(\mathbf{x}^*) = 0$, which includes n unknown variables.

A simple example: Given function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x} + r$, where $Q > 0$, vector $\mathbf{q} \in \mathcal{R}^n$ *and* $q \in \mathcal{R}$, compute the value $\mathbf{x}^*$ at which $f(\mathbf{x})$ is optimized (minimized in this case).

Solution: We have $\nabla f(\mathbf{x}) = \frac{1}{2} 2 Q\mathbf{x} + \mathbf{q} = Q\mathbf{x} + \mathbf{q}$, which we set equal to 0 and we get the result $\mathbf{x} = -Q^{-1}\mathbf{q} = \mathbf{x}^*$.

2 Descent Methods

2.1 General Form

We will consider iterative methods for finding the minimum of a function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$.

- **Definitions**
  - **Descent method** A method is called *descent method*, if it satisfies the following relation

    $$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}), \tag{8}$$

    where k is the iteration number, and when $\mathbf{x}^{(k)}$ has not reached the optimal point. Thus, the method is called *descent method*, if it leads to decrease of the value of the objective function.
  - **Descent direction** In this section, we will consider algorithms that produce a sequence of points $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}, ...,$ where

    $$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}, \tag{9}$$

    where $\Delta\mathbf{x}^{(k)}$ is a vector in $\mathcal{R}^n$, called the search direction of iteration k, and scalar $t^{(k)} > 0 \ \in \mathcal{R}$ is the step size at iteration k. In a *descent* method, the search direction must be such that $\nabla f(\mathbf{x}^{(k)})^T \Delta\mathbf{x}^{(k)} < 0$ for all k so that the value of the function decreases with iteration k. In that case, the direction is called *descent direction* for f at $\mathbf{x}^{(k)}$. That is, the search direction should make acute angle with the negative direction of the gradient at $\mathbf{x}^{(k)}$ (otherwise $f(\mathbf{x}^{(k+1)}) > f(\mathbf{x}^{(k)})$).

- **The algorithmic steps for a General Descent Method**

  Start with initial point $\mathbf{x}^{(0)}$.

  At each iteration k:

  1. Determine a descent direction $\Delta\mathbf{x}^{(k)}$.
  2. Determine a step $t^{(k)}$.
  3. Update the point according to equation: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\Delta\mathbf{x}^{(k)}$

  STOP if $\|\nabla f(\mathbf{x}^{(k)})\| \leq \varepsilon$, where ε a small positive number. More on stopping conditions later.

2.1.1 Question:

If $\nabla^T f(\mathbf{x}^{(k)})\, \Delta\, \mathbf{x}^{(k)} < 0$ like above and f is a convex function, show that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$.

Answer: Since f is a convex function, we have $f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)}) + \nabla^T f(\mathbf{x}^{(k)})\, (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$ for two points $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}$.
Since $\Delta\mathbf{x}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ and $\nabla^T f(\mathbf{x}^{(k)})\, \Delta\, \mathbf{x}^{(k)} < 0$, we have $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$.

2.2 Steepest descent method

The *steepest* descent method is a special case of descent methods, where the step size is selected based as follows. Assuming a step $t^{(k)}$ and a descent direction $\Delta \mathbf{x}^{(k)}$, the step for steepest descent method is found as

$$t^{(k+1)} = \arg\min_{s \geq 0} f(\mathbf{x}^{(k)} + s\,\Delta \mathbf{x}^{(k)}) \tag{10}$$

where $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + s\,\Delta \mathbf{x}^{(k)}$ is the $(k+1)$-th point in the descent direction method. The steepest descent method optimizes the step, so that the value $f(\mathbf{x}^{(k+1)})$ is as small as possible. This method accelerates the search of global minimum $\mathbf{x}^*$ for a convex function f.

There is also another descent method with constant step t at all iteration, which is simpler to implement but it is not as efficient as the method above.

3 Upper and lower bounds on Hessian matrix of f

Suppose that f is defined in a domain Ω and is a strongly convex function ($\nabla^2 f(\mathbf{x}) > 0$). Then there exists $m > 0$ such that

$$\nabla^2 f(\mathbf{x}) \geq m\,I \tag{11}$$

and also there exists $M \geq m > 0$ so that

$$\nabla^2 f(\mathbf{x}) \leq MI, \tag{12}$$

where I is the identity matrix of size n (assuming $\mathbf{x} \epsilon \mathcal{R}^n$).

The lower bound is easily derived.

Proof of upper bound : Assume that $\lambda(\mathbf{x})$ is the eigenvalue of $\nabla^2 f(\mathbf{x})$ and $\mathbf{y}(\mathbf{x})$ is the corresponding eigenvector of $\nabla^2 f(\mathbf{x})$. This means that

$$\nabla^2 f(\mathbf{x})\,\mathbf{y}(\mathbf{x}) = \lambda(\mathbf{x})\,\mathbf{y}. \tag{13}$$

Note that $\lambda(\mathbf{x})$ and $\mathbf{y}(\mathbf{x}$ depend on the particular point $\mathbf{x}$, since $\nabla^2 f(\mathbf{x})$ depends on $\mathbf{x}$ and that $\mathbf{y}(\mathbf{x}) > 0$, since $\nabla^2 f(\mathbf{x})$ is positive definite.

Define Λ to be the maximum eigenvalue of $\nabla^2 f(\mathbf{x})$ in domain Ω :

$$\Lambda = \max_{\mathbf{x} \in \Omega} \lambda(\mathbf{x}) \tag{14}$$

which leads us to inequality:

$$\nabla^2 f(\mathbf{x})\,\mathbf{y} \;\leq\; \Lambda\,\mathbf{y}. \tag{15}$$

Multiply from the left by $\mathbf{y}(\mathbf{x}) > 0$ to get:

$$\mathbf{y}^T(\mathbf{x})\,\nabla^2 f(\mathbf{x})\,\mathbf{y}(\mathbf{x}) \;\leq\; \mathbf{y}^T(\mathbf{x})\,\Lambda\,\mathbf{y}(\mathbf{x}) = \mathbf{y}^T(\mathbf{x})\,\Lambda\,I\,\mathbf{y}(\mathbf{x}) \tag{16}$$

and finally set $M = \Lambda$ to get:

$$\nabla^2 f(\mathbf{x}) \leq MI. \tag{17}$$

More information about the upper and lower bounds can be found in section 9.1.2 of the book "Convex Optimization" by Boyd and Vandenberghe.

3.1 Condition Number

From the strong convexity inequality and inequality (11) we have

$$m\,I \leq \nabla^2 f(\mathbf{x}) \leq M\,I. \tag{18}$$

The ratio $\frac{M}{m}$ is said to be an upper bound on the *condition number* of matrix $\nabla^2 f(\mathbf{x})$. The condition number of this matrix is defined as the ratio of its largest eigenvalue to its smallest eigenvalue.

$$\text{ConditionNumber} = \frac{\lambda_{\text{max}}(\mathbf{x})}{\lambda_{\text{min}}(\mathbf{x})} \tag{19}$$

and from the bounds above we get:

$$\text{ConditionNumber} \leq \frac{M}{m}. \tag{20}$$

The condition number gives a measure of "eccentricity" of domain Ω. If it is close to one, it means that the set has approximately the same "width" in all directions, i.e. it is nearly spherical. We will see later the role that condition number plays on convergence of descent methods.

One can show the following bound on the optimal value of objective function :

$$f(\mathbf{x}) - \frac{1}{2m}\|\nabla f(\mathbf{x})\|^2 \leq p^* \leq f(\mathbf{x}) - \frac{1}{2M}\|\nabla f(\mathbf{x})\|^2, \tag{21}$$

where $p^* = f(\mathbf{x}^*)$ is the value of the objective function f at the global optimum $\mathbf{x}^*$.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 9 : 30/10/06

Notes by : Aggelos-Christos Anadiotis, Giannis Dimitropoulos, Odysseas Kalamiotis

1 Gradient Descent Methods

In the gradient descent method, we choose as search direction $\Delta \mathbf{x}^{(k)} = - \bigtriangledown f(\mathbf{x}^{(k)})$. In figure 1 we depict some descent directions. The iteration for the gradient descent method is:

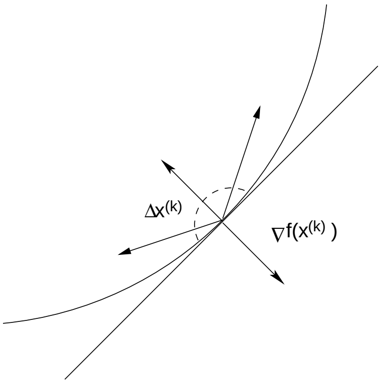


Figure 1: Descent directions for the Gradient Descent method.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t^{(k)} \bigtriangledown f(\mathbf{x}^{(k)}). \tag{1}$$

Variable step $t^{(k)}$: (Steepest descent method)

The step $t^{(k)}$ at each iteration is found such that the value of the objective function at the next iteration is as small as possible, namely

$$t^{(k)} = \arg\min_{s \geq 0} f(\mathbf{x}^{(k+1)}) = \arg\min_{s \geq 0} f(\mathbf{x}^{(k)} - s\nabla f(\mathbf{x}^{(k)})) \tag{2}$$

We can view $f(\mathbf{x}^{(k)} - t^{(k)}\nabla f(\mathbf{x}^{(k)}))$ as a function of one variable (the step), and its optimum is found by taking the derivative equal to 0. Thus, at each iteration, the step is optimized so that it causes the largest decrease in the value of the objective function at iteration $k+1$, $f(\mathbf{x}^{(k+1)}$.

$$\frac{\mathrm{d}f(\mathbf{x}^{(k)} - t^{(k)}\nabla f(\mathbf{x}^{(k)}))}{\mathrm{d}t^{(k)}} = 0 \tag{3}$$

Fixed Step *t*: We can use a fixed step at each iteration and thus have

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t\nabla f(\mathbf{x}^{(k)}). \tag{4}$$

Function $f(\cdot)$ achieves its minimum p^* with accuracy $\varepsilon > 0$,(i.e. $|f(\mathbf{x}^{(k)}) - p^*| < \varepsilon$)
in at most

$$k = \frac{\log(\dfrac{|f(\mathbf{x}^{(0)}) - p^*|}{\varepsilon})}{\log\dfrac{1}{1-\frac{m}{M}}} \tag{5}$$

steps. The number of steps it takes the algorithm to converge depends on:

1. Accuracy ε (the stricter the accuracy, i.e. the smaller the ε, the more the steps).

2. Initial point, $\mathbf{x}^{(0)}$ and initial distance of $f(\mathbf{x}^{(0)})$ from the optimal.

3. Parameters m, M that appear in the bound: $mI \leq \nabla^2 f(\mathbf{x}) \leq MI$. If $\dfrac{m}{M} \ll 1$ we can make the approximation: $\log\frac{1}{1-\frac{m}{M}} = -\log(1-\frac{m}{M}) \overset{\frac{m}{M}\ll 1}{\approx} \dfrac{m}{M}$
since $\log(1+x) \approx x$, for $x \ll 1$
Then we get that the number of steps is at most $k \approx \dfrac{M}{m}\dfrac{d(f(\mathbf{x}^{(0)}), p^*)}{\epsilon}$, where $d(a,b)$ is the distance between points a, b.

2 Properties of Gradient descent

1. For each iteration k, the points produced by the iteration are such that: $(\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}) \perp (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$. Thus, vector $(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$ is orthogonal to vector $(\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)})$, which equivalently means $\nabla f(\mathbf{x}^{(k+1)}) \perp \nabla f(\mathbf{x}^{(k)})$, or $\nabla f(\mathbf{x}^{(k)})$ is parallel to the tangent plane to the level set $\{f(\mathbf{x}) = f(\mathbf{x}^{(k+1)})\}$ at $\mathbf{x}^{(k+1)}$.

2. For each new point generated by the Gradient descent method, the value of function f decreases namely $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ (this is clear, since gradient descent is a special case of a descent method)

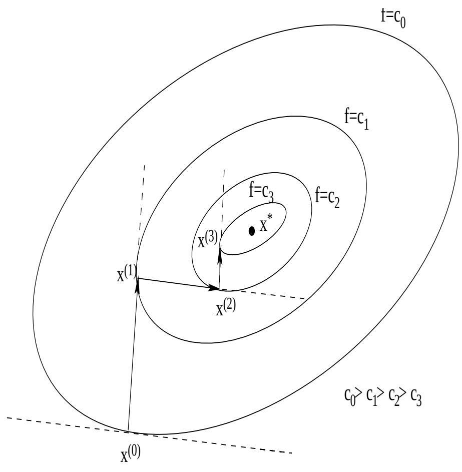


Figure 2: Point Sequence in the gradient descent method.

3 Example

Consider the quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{b}^T\mathbf{x}$, where Q is a $n \times n$ symmetric ($Q = Q^T$) positive-definite matrix, $\mathbf{b} \in \Re^n$, $\mathbf{x} \in \Re^n$. We want to find the form of iteration of gradient descent.

Thus, we need to find the step $t^{(k)}$ and the gradient $\nabla f(\mathbf{x}^{(k)}) = \mathbf{g}^{(k)}$ at each step, and then the iteration will be:

- $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t^{(k)}\nabla f(\mathbf{x}^{(k)})$.

We have:

- $\nabla f(\mathbf{x}^{(k)}) = Q\mathbf{x}^{(k)} - \mathbf{b} = \mathbf{g}^{(k)}$
- $$\begin{aligned} t^{(k)} &= \arg\min_t f(\mathbf{x}^{(k+1)}) \\ &= \arg\min_t f(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)}) \\ &= \arg\min_t \left\{ \frac{1}{2}(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})^T Q(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)}) \right\} \end{aligned}$$
- Define $\Phi(t) = \frac{1}{2}(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})^T Q(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})$
- By differentiating $\Phi(t)$ with respect to step size t, we get:

  $$\begin{aligned} \Phi'(t) &= \frac{1}{2}[(-\mathbf{g}^{(k)})^T Q(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})] + \frac{1}{2}[(\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})^T Q(-\mathbf{g}^{(k)})] + \mathbf{b}^T\mathbf{g}^{(k)} \\ &= (\mathbf{x}^{(k)} - t\mathbf{g}^{(k)})^T Q(-\mathbf{g}^{(k)}) + \mathbf{b}^T\mathbf{g}^{(k)}. \end{aligned}$$

  In order to find t which minimizes $\Phi(t)$, we take the derivative of $\Phi'(t)$ equal to zero:

  $$\Phi'(t) = 0 \Leftrightarrow t{\mathbf{g}^{(k)}}^T Q\mathbf{g}^{(k)} = ({\mathbf{x}^{(k)}}^T Q - \mathbf{b}^T)\mathbf{g}^{(k)} = 0$$

Observe that:

$$(\mathbf{x}^{(k)^T}Q - \mathbf{b}^T) = \mathbf{g}^{(k)^T} \tag{6}$$

Then we get that

$$t^{(k)} = \frac{\mathbf{g}^{(k)^T}\mathbf{g}^{(k)}}{\mathbf{g}^{(k)^T}Q\mathbf{g}^{(k)}} = \frac{||\mathbf{g}^{(k)}||^2}{\mathbf{g}^{(k)^T}Q\mathbf{g}^{(k)}}. \tag{7}$$

Finally, the iteration is:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{||\mathbf{g}^{(k)}||^2}{\mathbf{g}^{(k)^T}Q\mathbf{g}^{(k)}}\mathbf{g}^{(k)}. \tag{8}$$

4 Stopping criteria

The following stopping criteria for the iterative algorithm $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t^{(k)}\nabla f(\mathbf{x}^{(k)})$ are valid:

1. $||\nabla f(\mathbf{x}^{(k)}|| \leq \epsilon$, where $\epsilon > 0$ is a positive constant.
2. $|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| \leq \epsilon$ (the value of the objective function does not change significantly between two iterations).
3. $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}|| < \epsilon$ (the point does not change between two iterations).
4. $\dfrac{|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|}{f(\mathbf{x}^{(k)})} < \epsilon$ (the relative value of the object does not change much between two iterations)

5 Convergence of descent Algorithms

The sequence of points produced by the descent algorithm converges to a minimum $\mathbf{x}^*$, namely $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$. We will now study the rate of convergence for the following:

- Descent method,
- Gradient Descent method with fixed step t,
- Quadratic functions of the form $f(\mathbf{x}) = \dfrac{1}{2}\mathbf{x}^TQ\mathbf{x} - \mathbf{b}^T\mathbf{x}$. Note that $f(\cdot)$ is convex function since $\nabla^2 f(\mathbf{x}) = Q > 0$.
- In order to make analysis easier, we define the quadratic function $V(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}\mathbf{x}^{*^T}Q\mathbf{x}^* = \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^TQ(\mathbf{x}-\mathbf{x}^*)$, where $\mathbf{x}^* = Q^{-1}\mathbf{b}$ is the point at which $f(\mathbf{x})$ is minimized ($\nabla f(\mathbf{x}) = 0 \Rightarrow \mathbf{x}^* = Q^{-1}\mathbf{b}$).

- If $\mathbf{x} = \mathbf{x}^*$, which means we have reached the minimum of $f(\mathbf{x})$, then $V(\mathbf{x}^*) = 0$.

We start from the quadratic function property:

Property 1: The gradient descent iteration $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t^{(k)}\nabla f(\mathbf{x}^{(k)})$ for $f(x)$ satisfies the equation: $V(\mathbf{x}^{(k+1)}) = (1-\gamma_k)V(\mathbf{x}^{(k)})$, where:

$$\gamma_k = \begin{cases} 1, & \text{if } \mathbf{g}^{(k)} = \mathbf{0} \\ t^{(k)} \dfrac{\mathbf{g}^{(k)T} Q \mathbf{g}^{(k)}}{\mathbf{g}^{(k)T} Q^{-1} \mathbf{g}^{(k)}} (2 \dfrac{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}}{\mathbf{g}^{(k)T} Q \mathbf{g}^{(k)}} - t^{(k)}), & \text{if } \mathbf{g}^{(k)} \neq \mathbf{0} \end{cases}$$

where $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)}) = \nabla V(\mathbf{x}^{(k)})$. Note that, $\gamma_k = 1 - \dfrac{V(\mathbf{x}^{(k+1)})}{V(\mathbf{x}^{(k)})} \leq 1$, since $V(\mathbf{x}^{(k+1)}) < V(\mathbf{x}^{(k)})$ from the gradient descent method. Also γ_k 0. Therefore, $0 \leq \gamma_k \leq 1$.

If $\gamma_k = 1$ for some k, then $V(\mathbf{x}^{(k+1)}) = 0$, and this means we have reachead the optimum, $\mathbf{x}^{(k+1)} = \mathbf{x}^*$. Then, $\forall i \geq k+1$, it is $\mathbf{x}^{(i)} = \mathbf{x}^*$.

Property 2: Let $\mathbf{x}^{(k)}$ be the the sequence of points for the gradient descent method $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - a_k \mathbf{g}^{(k)}$. Let γ_k be defined as above. Then, $\mathbf{x}^{(k)} \to \mathbf{x}^*$, for any initial point $\mathbf{x}^{(0)}$, if and only if γ_k satisfies

$$\sum_{k=0}^{\infty} \gamma_k = \infty. \tag{9}$$

We now state some additional properties:

1. *Rayleigh Inequality:*For the $n \times n$ matrix $Q = Q^T > 0$, the following holds:

   $$\lambda_{\min}(Q)||\mathbf{x}||^2 \leq \mathbf{x}^T Q \mathbf{x} \leq \lambda_{\max}(Q)||\mathbf{x}||^2$$

   where $\lambda_{\min}(Q), \lambda_{\max}(Q) > 0$ are the minimum and maximum eigenvalues of Q.

2. For $Q = Q^T$, it is

   $$\lambda_{\min}(Q^{-1}) = \frac{1}{\lambda_{\max}(Q)},\ \lambda_{\max}(Q^{-1}) = \frac{1}{\lambda_{\min}(Q)} \tag{10}$$

   For $Q = Q^T$ and any $\mathbf{x} \in \Re^n$, it is:

   $$\frac{\lambda_{\min}(Q)}{\lambda_{\max}(Q)} \leq \frac{(\mathbf{x}^T\mathbf{x})^2}{(\mathbf{x}^T Q \mathbf{x})(\mathbf{x}^T Q^{-1} \mathbf{x})} \leq \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)} \tag{11}$$

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 10 : 31/10/06

Notes by : Konstantinos Gerogiokas and Alexandra Xamilothori

We continue with the convergence of quadratic function $V(\mathbf{x})$. The sequence of points $\{\mathbf{x}^{(k)}\}$ converges to the optimal point $\mathbf{x}^{\star}$.

Proof:

- if $\mathbf{g}^{(k)} = 0 \Rightarrow \mathbf{x}^{(k)} = \mathbf{x}^{*}$ and the result holds.
- if $\mathbf{g}^{(k)} \neq 0$, the best step was found to be: $t^{(k)} = \frac{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}{\mathbf{g}^{(k)T}Q\mathbf{g}^{(k)}}$, from which we get for γ_k:

$$\gamma_k = \frac{(\mathbf{g}^{(k)T}\mathbf{g}^{(k)})^2}{(\mathbf{g}^{(k)T}Q\mathbf{g}^{(k)})(\mathbf{g}^{(k)T}Q^{-1}\mathbf{g}^{(k)})} \geq \quad \frac{\lambda_{\min(Q)}}{\lambda_{\max(Q)}}$$

Where the last inequality is found by applying inequality:

$$\frac{\lambda_{\min(Q)}}{\lambda_{\max(Q)}} \leq \frac{(\mathbf{x}^T\mathbf{x})^2}{(\mathbf{x}^TQ\mathbf{x})(\mathbf{x}^TQ^{-1}\mathbf{x})} \leq \frac{\lambda_{\max(Q)}}{\lambda_{\min(Q)}}, \quad \texttt{for} \quad \mathbf{x} = \mathbf{g}^{(k)}.$$

Thus:

$$\sum_{k=0}^{\infty} \gamma_k \to \infty$$

and the result follows from Property 2 of previous lecture.

1 Gradient descent method with fixed step

For the gradient descent algorithm with fixed step t,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t\mathbf{g}^{(k)}$$

we have convergence $\quad \mathbf{x}^{(k)} \to \mathbf{x}^{\star} \quad$ if and only if $\quad 0 < t < \frac{2}{\lambda_{\max(Q)}}$.

2 Rate of convergence of gradient descent for quadratic function with variable step $t^{(k)}$

For the gradient descent with variable step $t^{(k)}$ we have:

$$V(\mathbf{x}^{(k+1)}) \leq \left(\frac{\lambda_{\max(Q)} - \lambda_{\min(Q)}}{\lambda_{\max(Q)}}\right) V(\mathbf{x}^{(k)}) \tag{1}$$

Proof:

From a previous property: $V(\mathbf{x}^{(k+1)}) = (1 - \gamma_k) \cdot V(\mathbf{x}^{(k)})$ and from above we have shown that $\gamma_k \geq \frac{\lambda_{\min(Q)}}{\lambda_{\max(Q)}}$. Thus (1) can be easily derived.

- Define $r = \frac{\lambda_{\max(Q)}}{\lambda_{\min(Q)}}$ as the condition number of matrix Q. We get from (1) by substituting r, that: $V(\mathbf{x}^{(k+1)}) \leq (1 - \frac{1}{r}) \cdot V(\mathbf{x}^{(k)})$, where $0 < r \leq 1$ since $0 < 1 - \frac{1}{r} \leq 1$.

The parameter $(1 - \frac{1}{r})$ is called convergence ratio and shows how fast the value of the quadratic function $V(\mathbf{x})$ decreases at each iteration.

If $(1 - \frac{1}{r}) \downarrow \;\Rightarrow r \downarrow \;\Rightarrow V(\mathbf{x}^{(k)})$ converges faster to zero and thus at the optimal point.

If $r = 1$ (then $\lambda_{\max(Q)} = \lambda_{\min(Q)}$), we get convergence to the optimum $\mathbf{x}^*$ in one step.

Example: Consider function $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2$, that has minimum value 0, achieved at $\mathbf{x}^* = (0, 0)$.

▷ $x_1^2 + x_2^2$ can be written in the form $\frac{1}{2}\mathbf{x}^T Q\mathbf{x}$ as $\frac{1}{2}(x_1 \quad x_2)\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

▷ $Q = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ has $\operatorname{rank}(Q) = 2$. It has two positive eigenvalues, both equal to 2. Thus: $\lambda_{\max}(Q) = \lambda_{\min}(Q) = 2$ and $r = 1$.

Consider the gradient method $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t\mathbf{g}^{(k)}$,where fixed step t. Let $t = \frac{1}{2}$. With this step, we can go to (0,0) in one step. Indeed, we have $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - t\mathbf{g}^{(0)}$, where $\nabla f(\mathbf{x}) = 2\mathbf{x}$ and $\mathbf{g}^{(0)} = 2\mathbf{x}^{(0)} \quad \Rightarrow$

$\Rightarrow \mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \mathbf{x}^{(0)} = \mathbf{0}$ in just one iteration.

Note that for different steps, we get to $(0, 0)$ in more iterations. As can be shown, the level sets of f are circles of different radii. The convergence of the gradient descent algorithm is depicted in figure 1.

Example: Show the convergence of gradient method with fixed step for function $f(x_1, x_2) = \frac{x_1^2}{5} + x_2^2$.

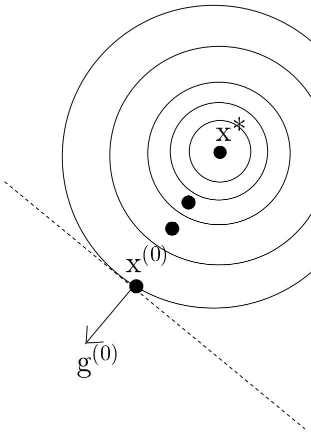


Figure 1: Convergence to the optimal point $\mathbf{x}^*$.

The level sets are ellipses and the convergence to the optimal point $(0, 0)$ is realized in more steps.

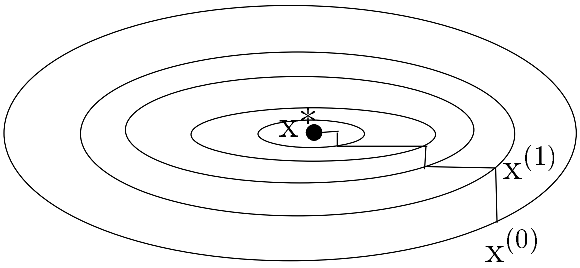


Figure 2: Convergence of the gradient method.

3 Newton method for finding the minimum of f

Given a function $f(\mathbf{x})$, the purpose of Newton method is to approximate $f(\mathbf{x})$ through another function $q(\mathbf{x})$ and find the minimum of q. The approximation should be such that:

$$f(\mathbf{x}^{(k)}) = q(\mathbf{x}^{(k)}) \tag{2}$$

$$\nabla f(\mathbf{x}^{(k)}) = \nabla q(\mathbf{x}^{(k)}) \tag{3}$$

$$F(\mathbf{x}^{(k)}) = Q(\mathbf{x}^{(k)}) \tag{4}$$

for all points $\mathbf{x}^{(k)}$, where F($\cdot$), Q($\cdot$) are the Hessian matrices of f, q. A square function $q(\mathbf{x})$ that fulfills the above criteria is:

$$q(\mathbf{x}) = f(\mathbf{x}^{(k)}) + (\mathbf{x} - \mathbf{x}^{(k)})^T \mathbf{g}^{(k)} + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^T F(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) \tag{5}$$

where $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$ and $F(\mathbf{x}^{(k)})$ is the Hessian matrix of $f(\mathbf{x})$ at point $\mathbf{x}^{(k)}$. We can verify that:

$$f(\mathbf{x}^{(k)}) = q(\mathbf{x}^{(k)}) \tag{6}$$

$$\nabla f(\mathbf{x}^{(k)}) = \nabla q(\mathbf{x}^{(k)}) \tag{7}$$

$$F(\mathbf{x}^{(k)}) = Q(\mathbf{x}^{(k)}). \tag{8}$$

Calculating the gradient of $q(\mathbf{x})$ and setting it to 0 ,we get to the point:

$$\nabla q(\mathbf{x}) = \mathbf{0} \Rightarrow \mathbf{g}^{(k)} + F(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{x}^{(k)} - F^{-1}(\mathbf{x}^{(k)})\mathbf{g}^{(k)}$$

For the next iteration we set $\mathbf{x} = \mathbf{x}^{(k+1)}$, so the iteration becomes:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - F^{-1}(\mathbf{x}^{(k)})\mathbf{g}^{(k)} \Rightarrow \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\nabla^2 f(\mathbf{x}^{(k)})]^{-1}\nabla f(\mathbf{x}^{(k)}).$$

Note that, as a special case, if $[\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$ is a diagonal matrix with the vector of fixed step $\{t^{(k)} \dots t^{(k)}\}$ in its diagonal, then we have the gradient descent method. Note that the Newton method at each iteration goes toward direction $\mathbf{u}^{(k)} = -F^{-1}(\mathbf{x}^{(k)})\nabla f(\mathbf{x}^{(k)})$.

- **Disadvantages** of Newton method are:

1. Calculation of the inverse table ($[\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$) at each step, which is of complexity O(n^3) for $n \times n$ matrix.
2. The inverse of $\nabla f(\mathbf{x}^{(k)})$ has to exist and be positive definite.

However, the **advantage** of Newton method is that it converges faster to the optimal point.

4 Convergence Order

Given a sequence $\mathbf{x}^{(k)}$, $k = 1...$ we say that it converges to vector $\mathbf{x}^*$, if

$$\lim_{k\to\infty} \parallel \mathbf{x}^{(k)} - \mathbf{x}^* \parallel = 0.$$

The *convergence order* of $\mathbf{x}^{(k)}$ is $p \in \mathcal{R}$ $(1 \leq p \leq \infty)$, if

$$0 < \lim_{k \to \infty} \frac{\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \|}{\| \mathbf{x}^{(k)} - \mathbf{x}^* \|^p} < \infty$$

Then we get

$$\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \| \sim c \| \mathbf{x}^{(k)} - \mathbf{x}^* \|^p \qquad \text{for some} \quad c \in \mathcal{R},\, 0 < c < \infty$$

The larger the p, the faster the convergence. It turns out that:

- The Gradient descent method has $p = 1$ (linear convergence)
- The Newton method has $p = 2$ (quadratic convergence), which is faster than the gradient method.

5 Throughput maximization with power control in wireless communications

Example: There are N transmitters, N receivers, and each transmitter i is connected to a receiver i.

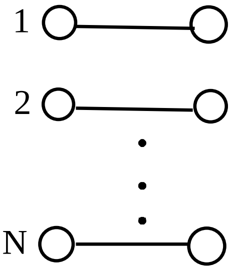


Figure 3: N transmitters connected to N receivers. The different transmitter-receiver links can have different relative positions.

The Signal to Interference and Noise Ratio at each receiver i as a function of the transmitter power vector $\mathbf{P} = (P_1, \ldots, P_N)$ is given by

$$SINR_i(\mathbf{P}) = \frac{G_{ii} P_i}{\sum\limits_{j=1, j \neq i}^{N} G_{ji} P_j + N_i} \tag{9}$$

where N_i is the noise power at receiver i and P_i is the transmission power of transmitter i.

The *capacity* for each link i is $C_i = \log_2(1 + SINR_i(\mathbf{P})) \approx \log_2(SINR_i(\mathbf{P}))$ for large enough SINRs.

We assume that at each transmitter i, where packets arrive and wait in a queue before being transmitted. Let q_i be the number of packets (size of queue) that are waiting for transmission. The product $q_i C_i(\mathbf{P})$ is called *throughput* of link i.

Our purpose: We wish to maximize the sum of $q_i C_i(\mathbf{P})$, namely the total system throughput, or

$$\max_{\mathbf{P}} \sum_{i=1}^{N} q_i C_i(\mathbf{P}),$$

by appropriately controlling transmission power vector $\mathbf{P}$.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 11 : An optimization problem - 6/11/06

Notes by : George Noutsis and George Xatziparaskevas

1 An optimization problem solved by the gradient method

In the previous class, we had the problem of optimizing the following function:

$$\max_{(P_1,\ldots,P_N)} \sum_{l=1}^{N} q_l \log(SINR_l) = \max_{(P_1,\ldots,P_N)} \sum_{l=1}^{N} q_l \log\left(\frac{G_{ll}P_l}{\sum\limits_{k=1,k\neq l}^{N} G_{kl}P_k + N_l}\right) = \max_{(P_1,\ldots,P_N)} f(\mathbf{P}) \tag{1}$$

by controlling transmission power vector $\mathbf{P} = (P_1, \ldots, P_N)$, where q_i is the queue size of transmitter i and $C_i(P)$ is the capacity of link i.

Although $f(P)$ does not seem to be a concave function of P, we will transform it to a concave function. If we prove that $f(\mathbf{P})$ is concave the local maximum is global as well.

Consider the transformation: $\widetilde{P_l} = \ln \mathbf{P}_l$,for $l = 1, \ldots, N$ and the vector $\widetilde{P} = (\widetilde{P_1}, \ldots, \widetilde{P_N})$.

We have:

$$f(\widetilde{P}) = \sum_{l=1}^{N} q_l \log\left(\frac{G_{ll}e^{\widetilde{P_l}}}{\sum\limits_{k\neq l}^{N} G_{kl}e^{\widetilde{P_k}} + N_l}\right) = \sum_{l=1}^{N} q_l \left[\underbrace{\log\left(G_{ll}e^{\widetilde{P_l}}\right)}_{\text{term 1}} - \underbrace{\log\left(\sum_{k\neq l}^{N} G_{kl}e^{\widetilde{P_k}} + N_l\right)}_{\text{term 2}}\right] \tag{2}$$

Term 1 is linear in $\widetilde{\mathbf{P}}$. We will prove that term 2 is convex in $\widetilde{\mathbf{P}}$. In order to do that, we shall examine first under which conditions a function $f(\cdot)$ that arises as composition of functions $h(\cdot)$ and $g(\cdot)$ is convex or concave.

Let $f(x) = (h \circ g)(x)$, with $f(x) = h(g(x))$. For functions of one variable x, $h : \mathbf{R} \longrightarrow \mathbf{R}$ and g: $\mathbf{R^n} \longrightarrow \mathbf{R}$, assume that h and g are twice differentiable. In this case convexity of $f(\cdot)$ means

$f''(x) \geq 0$ for all $x \in \mathbf{R}$. The first and the second derivatives of $f(\cdot)$, $f = h \circ g$ are:

$$f'(x) = h'(g(x))g'(x),$$

$$f''(x) = h''(g(x))(g'(x))^2 + h'(g(x))g''(x)$$

Function f is concave if:

(i.) h is concave, increasing and g is concave, or

(ii.) h is concave, decreasing and g is convex.

Function f is convex if:

(i.) h is convex, increasing and g is convex, or

(ii.) h is convex, decreasing and g is convex.

Vector Composition: Consider now the case where $f(.)$ is composition of several functions, that is:

$$f(x) = h(g_1(x), \ldots, g_k(x)) = h(\mathbf{g}(x))$$

with $h : \mathbf{R}^{\mathbf{k}} \longrightarrow \mathbf{R}$, $g_i : \mathbf{R} \longrightarrow \mathbf{R}$.

The first and the second derivatives of $f(.)$ are as follows:

$$f'(x) = \nabla h^T(\mathbf{g}(x))\mathbf{g}'(x)$$

$$f''(x) = \mathbf{g}'^T(x)\nabla^2 h(\mathbf{g}(x))\mathbf{g}'(x) + \nabla h^T(\mathbf{g}(x)\mathbf{g}''(x)$$

where $\mathbf{g}'(x) = (g_1'(x), \ldots, g_k'(x))$. Function f is convex if:

(i.) h is convex, increasing in each argument and g_i is convex.

(ii.) h is convex, decreasing in each argument and g_i is concave

Proof of concavity of $f(\widetilde{P})$: Consider function $h(\mathbf{z}) = \log(\sum_{i=1}^{k} e^{z_i})$. As a first step for proving concavity of $f(\widetilde{P})$, we will prove that $h(\mathbf{z})$ is convex, or that its Hessian matrix $H(\mathbf{z}) > 0$.

The first derivative of the function $h(\mathbf{z})$ is:

$$\frac{\vartheta h(\mathbf{z})}{\vartheta z_i} = \frac{e^{z_i}}{\sum\limits_{k=1}^{k} e^{z_i}}$$

The second derivative with respect to the i^{th} component:

$$\frac{\vartheta^2 h(\mathbf{z})}{\vartheta z_i^2} = \frac{e^{z_i}}{\sum\limits_{j=1}^{k} e^{z_j}} - \frac{e^{2z_i}}{(\sum\limits_{j=1}^{k} e^{z_j})^2}$$

The second derivative with respect to component z_i, z_j with $(i \neq j)$ is:

$$\frac{\vartheta^2 h(\mathbf{z})}{\vartheta z_i \vartheta z_j} = -\frac{e^{z_i+z_j}}{(\sum\limits_{j=1}^{k} e^{z_i})^2}$$

Consider the case of $N = 2$ to better visualize the situation. Let $A = e^{z_1} + e^{z_2}$. Then for any $\mathbf{v} \geq 0$, $\mathbf{v} = (v_1, v_2)$ the quadratic for $\mathbf{v}^H H \mathbf{v}$, with H the Hessian of $h(z)$, should be shown non-negative. Thus:

$$\mathbf{v}^T H \mathbf{v} \geq 0 \Leftrightarrow \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} \frac{Ae^{z_1} - e^{2z_1}}{A^2} & -\frac{e^{z_1+z_2}}{A^2} \\ -\frac{e^{z_1+z_2}}{A^2} & \frac{Ae^{z_2} - e^{2z_2}}{A^2} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \geq 0 \Leftrightarrow$$

$$\Leftrightarrow v_1^2 \left(\frac{e^{z_1}}{A} - \frac{e^{2z_1}}{A^2} \right) - 2v_1 v_2 \frac{e^{z_1+z_2}}{A^2} + v_2^2 \left(\frac{e^{z_2}}{A} - \frac{e^{z_2}}{A^2} \right) =$$

$$= \frac{A(v_1^2 e^{z_1} + v_2^2 e^{z_2}) - (v_1 e^{z_1} + v_2 e^{z_2})^2}{A^2} \overset{?}{\geq} 0. \tag{3}$$

In order to prove the above, we use the Cauchy-Schwartz inequality for vectors $\mathbf{q}, \mathbf{b}$:

$$(\mathbf{a}^T \mathbf{a})(\mathbf{b}^T \mathbf{b}) \geq (\mathbf{a}^T \mathbf{b})^2$$

for

$$\mathbf{a} = \begin{pmatrix} e^{\frac{z_1}{2}} & e^{\frac{z_2}{2}} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} v_1 e^{\frac{z_1}{2}} & v_2 e^{\frac{z_2}{2}} \end{pmatrix}.$$

Once we proved that inequality (3) holds, we have proved that $h(\mathbf{z}) = \log(\sum_{i=1}^{k} e^{z_i})$ is convex, $h(.)$ is increasing ($\nearrow$) in its argument z_i. Thus, function $h(\mathbf{g}(x)) = \log(\sum_{i=1}^{k} e^{g_i(x)})$ is convex if $g_i(x)$ is convex [rule (i.)].

Now, we have:

$$f(\widetilde{P}) = \sum_{l=1}^{N} q_l \left[\underbrace{\log\left(G_{ll} e^{\widetilde{P}_l} \right)}_{\text{term}\,1} \underbrace{- \log\left(\sum_{k \neq l} e^{\ln G_{kl} + \widetilde{P}_k} + N_l \right)}_{\text{term}\,2} \right]$$

Term 1 is linear in $\tilde{\mathbf{P}}$, as we said before. Also, $g_k(x) = \ln G_{kl} + \widetilde{P}_k$ is linear in $\tilde{\mathbf{P}}_k$, so it is convex as well. Thus, $\log \sum_{k \neq l} e^{\ln G_{kl} + \widetilde{P}_k} + N_l$ is convex in $\tilde{\mathbf{P}}$ and thus $(- \log \sum_{k \neq l} e^{\ln G_{kl} + \widetilde{P}_k} + N_l)$ is concave and the whole $f(\tilde{\mathbf{P}})$ is concave in $\tilde{\mathbf{P}}$.

Now, we use the gradient ascent method to find the global maximum of $f(\mathbf{P})$ (we come back to the initial notation with $\mathbf{P}$, since we have used the transformation to $\tilde{\mathbf{P}}$ only to show the concavity of $f(\cdot)$).

$$f(\mathbf{P}) = \sum_{l=1}^{N} q_l \log \frac{G_{ll}P_l}{\sum\limits_{k\neq l}^{N} G_{kl}P_k + N_l}$$

$$\frac{\vartheta f(\mathbf{P})}{\vartheta P_l} = \frac{q_l}{P_l} - \sum_{j\neq l} \frac{q_j G_{jl}}{\sum_{k\neq j} G_{jk}P_k + N_j}$$

1. Start with an initial vector $\mathbf{P}^{(0)} = (P_1^{(0)}, P_2^{(0)}, \ldots, P_N^{(0)})$.

2. At the $(k+1)$-th step of the algorithm, we have the iteration: $\mathbf{P}^{(k+1)} = \mathbf{P}^{(k)} + \beta\nabla f(\mathbf{P}^{(k)})$, where β is the constant step size.

Note that since we want to maximize $f(\mathbf{P})$, we have a gradient ascent method, with $f(\mathbf{P}^{(k+1)}) > f(\mathbf{P}^{(k)})$ and we move towards the direction of maximum increase of $f(\cdot)$, (i.e towards the direction of the gradient)

Each transmitter l updates its power according to rule:

$$P_l^{(t+1)} = P_l^{(t)} + \beta \frac{\vartheta f P}{\vartheta P_l} \Longrightarrow$$

$$\Longrightarrow P_l^{(t+1)} = P_l^{(t)} + \beta \left(\frac{q_l}{P_l^{(t)}} - \sum_{j\neq l} \underbrace{\frac{q_j G_{il}}{\sum_{k\neq j} G_{jk}P_k^{(t)} + N_j}}_{m_j^{(t)}} \right)$$

Let $m_j^{(t)}$ given as noted above. By multiplying numerator and denominator by G_{jj} and $P_j^{(t)}$ we have:

$$m_j^{(t)} = \frac{q_j SINR_j^{(t)}}{G_{jj}P_j^{(t)}}$$

Consequently, at the $(t+1)$-th step of the iteration we have:

$$P_l^{(t+1)} = P_l^{(t)} + \beta \left(\frac{q_l}{P_l^{(t)}} - \sum_{j\neq l} G_{jl}\, m_j^{(t)} \right)$$

Thus $m_j^{(t)}$ can be considered as a message pertaining to transmitter j, for $j = 1, \ldots, N$. Each transmitter knows its queue q_j, its gain to its receiver G_{jj} and its transmission power $P_j^{(t)}$. It also receives channel state information (CSI) from the receiver in the form of SINR_j.

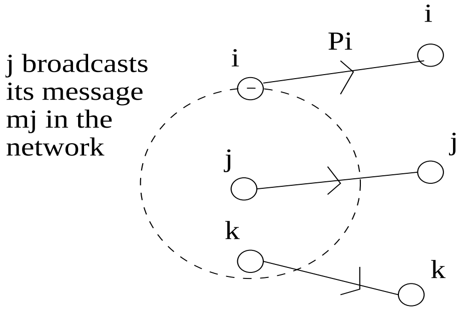


Figure 1: Explanation of message broadcasting in the network for distributed algorithm operation.

Each node j broadcasts this message to every other node, $l \neq j$ which then updates its power according to the rule above. It turns out that the gradient ascent method $\mathbf{P}^{(t+1)} = \mathbf{P}^{(t)} + \beta \nabla f(\mathbf{P}^{(t)})$ above (independently of the initial power vector $\mathbf{P}^{(0)} = (P_1^{(0)}, \ldots, P_N^{(0)})$ and the sequence in which the iteration will be executed by the users) converges to the optimal vector $\mathbf{P}^*$. This is the vector that maximizes $f(\mathbf{P})$. The algorithm is distributed, since each transmitter uses quantities that he only knows and needs to know only messages $m_j(t)$.

Observation: In general, if the objective function is separable in its variables, namely if

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$$

can be written as sum of functions, where each function depends only on one variable, i.e if

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \underbrace{f_i(x_i)}_{concave}$$

then the iteration of gradient ascent method

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \beta \nabla f(\mathbf{x}^{(t)})$$

becomes:

$$x_i^{(t+1)} = x_i^{(t)} + \beta \frac{\vartheta f_i(x_i)}{\vartheta x_i}$$

and thus there is no message passing needed among nodes. Simply, each node computes $\frac{\vartheta f_i(x_i)}{\vartheta x_i}$ (since it knows $f_i(x_i)$, but not $f_j(x_j)$) and does the updates of its variables independently from others.

The independent iterations for each node will again lead to the optimal vector $\mathbf{x}^*$, namely the vector that maximizes $f(\mathbf{x})$.

Advanced topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 12 : Linear Programming - 07/11/06

Notes by Eleni Galanou and Despina Koutsagia

1 Linear Programming (LP)

LP problems originally appeared in Operations Research. The form of an LP problem is as follows:

$$\text{minimize } \mathbf{c}^T\mathbf{x}, \tag{1}$$

subject to the constraints:

$$A\mathbf{x} = \mathbf{b} \text{ or } A\mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}, \tag{2}$$

with $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{R}^n$, $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{R}^n$, $\mathbf{b} = (b_1, b_2, \ldots, b_m) \in \mathcal{R}^m$.

Function $\mathbf{c}^T\mathbf{x} : \mathcal{R}^n \to R$ is called the *objective function* and $A\mathbf{x} \leq \mathbf{b}$ are called constraints. More specifically:

- c_i is the cost per unit of variable x_i.
- The total cost can be represented by $\mathbf{c}^T\mathbf{x} = c_1x_1 + \ldots + c_nx_n$.
- x_i is the i-th variable i.

The constraints and the objective function are linear to vector of variables $\mathbf{x}$. Matrix $A \in \mathcal{R}^{mxn}$ is a $m \times n$ matrix, $\mathbf{b} \in \mathcal{R}^m$, and

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$

Therefore the problem can be written as:

$$\text{minimize } c_1x_1 + \cdots + c_nx_n$$

$$\text{s.t.} \quad a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$

$$\vdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

or

$$\text{minimize } c_1x_1 + \cdots + c_nx_n,$$

$$\mathbf{a}_i^T\mathbf{x} = b_i, \tag{3}$$

where $\mathbf{a}_i$ is the i-th row of matrix A, for $i = 1, \ldots, m$. We will focus on formulating and solving LP problems.

2 Example 1 : A company with 4 products

A company constructs four products : $\Pi_1, \Pi_2, \Pi_3, \Pi_4$.The resources that are needed are: man-weeks, kg of material A and quantity of material B (in packages).

Resources	Π_1	Π_2	Π_3	Π_4	Resources
man-weeks	1	2	1	2	20
kg of material A	6	5	3	2	100
packages of material B	3	4	9	12	75

Each cell (i, j) in the table above contains the number of units of resource i which are necessary to produce one unit of product j. Thus, for example Π_2,Π_4 are the most demanding ones in man-weeks. Also,6 kilograms of material A are needed to make one unit of Π_1.

The last column of the table shown the availability of resources. Availability shows the amounts of the resources that the company can waste to produce the products. So availability is going to be vector **b** and the table is going to be matrix A of the Linear Program.

There is also a cost vector [6475], where each cost coefficient c_i expresses the benefit of the company for each unit of product $\Pi_i, i = 1, 2, 3, 4$ that is sold. Thus $c_1 = 6$ is the profit per unit of product Π_1.

The company's objective is to find the vector $\mathbf{x} = (x_1, x_2, x_3, x_4)$ with x_i the quantity of Π_i that must be constructed so as to maximize the total benefit from all products. The problem is stated as follows:

$$\text{maximize } \mathbf{c}^T\mathbf{x} = 6x_1 + 4x_2 + 7x_3 + 5x_4 \tag{4}$$

subject to the constraints :

$$x_1 + 2x_2 + x_3 + 2x_4 = 20 \tag{5}$$

$$6x_1 + 5x_2 + 3x_3 + 2x_4 = 100 \tag{6}$$

$$3x_1 + 4x_2 + 9x_3 + 12x_4 = 75 \tag{7}$$

and $\mathbf{x} = (x_1, x_2, x_3, x_4) \geq \mathbf{0}$.

In the formulation, we assumed that all available resources are used.

3 Example 2 : Diet Problem

There are n different kinds of food and m vitamins. Each unit of food j costs c_j, $j = 1, \ldots, n$. To achieve balanced diet, we must receive at least b_i units of vitamin i per day, $i = 1, \ldots, m$.

A unit of food j contains a_{ij} units of vitamin i. Elements a_{ij} form matrix A . $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of variables, where x_j is the amount of food j in the diet. We want to find the quality x_j of each food j that should be consumed per day, so that all necessary vitamins are received and the cost is minimized. This is the min-cost diet problem which can be formulated as follows:

$$\text{minimize } \sum_{i=1}^{n} c_i x_i = \mathbf{c}^T\mathbf{x}$$

subject to

$$a_{11}x_1 + \cdots + a_{1n}x_n \geq b_1$$

$$\vdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n \geq b_m$$

and $\mathbf{x} = (x_1, \ldots, x_n) \geq (0, 0, \ldots, 0)$ or,

$$\min \quad \mathbf{c}^T\mathbf{x} \tag{8}$$

subject to:

$$A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}. \tag{9}$$

4 Geometrical interpretation of LP Problems

Consider the problem:

$$\max \begin{pmatrix} 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\text{s.t.} \quad \begin{pmatrix} 5 & 6 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 30 \\ 12 \end{pmatrix} \text{with } x_1, x_2 \geq 0.$$

Infinitely many points (x_1, x_2) satisfy the two constraints and the set of feasible solutions is all such points. Now, we draw the region of feasible solutions. This is shaded area OABC in the figure below.

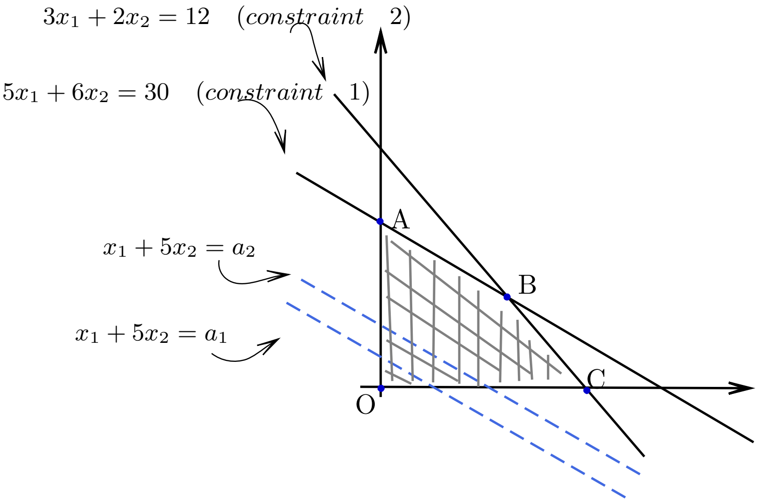

$3x_1 + 2x_2 = 12$ (*constraint* 2)
$5x_1 + 6x_2 = 30$ (*constraint* 1)
$x_1 + 5x_2 = a_2$
$x_1 + 5x_2 = a_1$
A
B
C
O

Figure 1: The feasible region of an LP problem.

Later we will see that the optimal solution $\mathbf{x}^* = (x_1^*, x_2^*)$, i.e. the one that maximizes the objective $x_1 + 5x_2$ is *always* one of the four vertices O,A,B or C.

Geometrically, maximizing $\mathbf{c}^T\mathbf{x} = x_1 + 5x_2$ subject to $\mathbf{x} \in (OABC)$ amounts to finding a straight line $x_1 + 5x_2 = a$ that intersects with the shaded region and has the largest value, a.

Thus, we can draw the lines $x_1 + 5x_2 = a$ and consecutively increase a to values $a_1 < a_2 < \ldots$. Thus, we form the parallel lines

$$x_1 + 5x_2 = a_1$$

$$x_1 + 5x_2 = a_2$$

$$\vdots$$

$$x_1 + 5x_2 = a_{\max},$$

until we reach the value $a_{\max}$, beyond which if we increase a further, we will go out of the feasible region. Then $a_{\max}$ is the maximum value of the objective function, and the point of intersection of $x_1 + 5x_2 = a_{\max}$ with region (OABC) is the optimal solution.

In general the LP problem is of the form:

$$\text{minimize } \mathbf{c}^T\mathbf{x} \tag{10}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{P}. \tag{11}$$

Then $\mathcal{P}$ is called set of feasible solutions of the LP problem and is a polyhedron.

Objective $\mathbf{c}^T\mathbf{x}$ is linear in the vector of variables $\mathbf{x}$, so its level curves are *hyperplanes* orthogonal to $\mathbf{c}$ (shown by dashed lines).

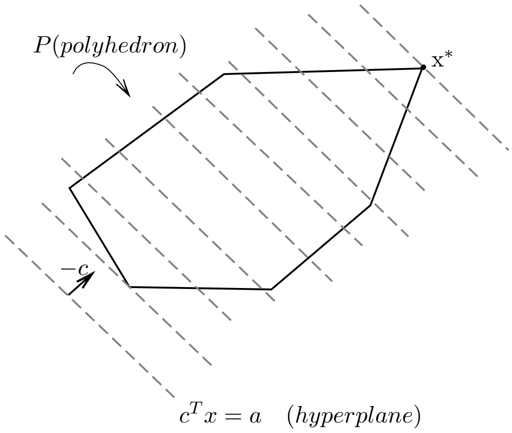


Figure 2: Feasible region of LP problems is a polyhedron.

The optimal solution $\mathbf{x}^*$ is the point in $\mathcal{P}$ as far as possible in direction $-\mathbf{c}$. Sometimes the optimal solution is not only one point but several.

Example: Consider the LP problem

$$\text{minimize } c_1x_1 + c_2x_2 \tag{12}$$

subject to:

$$-x_1 + x_2 \le 1, \text{ with } x_1 \ge 0, \ , x_2 \ge 0, \ A = [-1 \ 1] \ b = [1]. \tag{13}$$

We have the following cases with regard to a solution:

1. An LP problem may have a unique solution, e.g. when $\mathbf{c} = (1,1)$, then $\Rightarrow x_1^* = 0, x_2^* = 0 \Rightarrow$ $\mathbf{x}^* = [0,\ 0]$ is the optimal solution.
2. The problem may have multiple optimal solutions.
   - If $\mathbf{c} = (1,0)$, then any vector $(0, x_2)$ is optimal with $x_2 \in [0,1]$. The set of the optimal solutions is infinite but bounded, since $0 \le x_2 \le 1$.
   - If $\mathbf{c} = (0,1)$, then there exist several optimal solutions of the form$(x_1, 0)$ with $x_1 \in [0, \infty]$. The set of the optimal solutions is infinite and unbounded in this case.
3. An LP problem has optimal cost $-\infty$ and no finite feasible solution. For example, if $\mathbf{c} = (-1,-1)$, then for the problem

$$\begin{gathered} \min\ (-x_1 - x_2) \\ \text{s.t. } x_2 \le 1 + x_1 \end{gathered}$$

for any feasible solution (x_1, x_2), we can produce another feasible solution with less cost by simply increasing x_1. By considering vectors with increasing values of x_1, x_2, we obtain a sequence of feasible solutions that goes to $-\infty$.

5 Standard form of an LP Problem

An LP problem is said to be in standard form if it is of the form:

$$\begin{gathered} \min\ \mathbf{c}^T\mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \text{ with } \mathbf{x} \ge \mathbf{0} \\ A \in \mathcal{R}^{m\times n},\ m < n,\ \operatorname{rank}(A) = m,\ \mathbf{b} \ge \mathbf{0}. \end{gathered}$$

Namely, an LP is said to be in *standard* form has equality constraints, and is a minimization problem.

An LP problem is in *inequality* form if it is of the form:

$$\begin{gathered} \min\ \mathbf{c}^T\mathbf{x} \\ \text{s.t. } A\mathbf{x} \ge \mathbf{b} \ \text{ with } \ \mathbf{x} \ge \mathbf{0} \\ A \in \mathcal{R}^{m\times n},\ m < n,\ \operatorname{rank}(A) = m,\ \mathbf{b} \ge \mathbf{0}. \end{gathered}$$

Those two forms are equivalent in the sense that starting from a feasible solution of a standard form problem we can produce a feasible solution of an inequality form problem with the same cost (and vice versa)

Consider a non-standard LP problem:

$$\min \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \geq \mathbf{b} \ \text{ with } \ \mathbf{x} \geq \mathbf{0}.$$

In order to convert it to standard form, we need to convert the inequalities to equalities. We subtract a positive quantity y_i out of each constraint i. We call $y_i, i = 1, 2 \ldots, m$ *surplus variables*, with $\mathbf{y} \geq \mathbf{0}$. Then, we have:

$$\begin{gathered} a_{11}x_1 + \ldots + a_{1n}x_n \geq b_1 \ \Rightarrow \ a_{11}x_1 + \ldots + a_{1n}x_n - y_m = b_1 \\ \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \geq b_m \ \Rightarrow \ a_{m1}x_1 + \ldots + a_{mn}x_n - y_m = b_m \end{gathered}$$

or in matrix form it is written as: $(A \quad -I_m) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b}$ where $(A \quad -I_m)$ is a block matrix and I_m is the $m \times m$ unit matrix. Thus, the non-standard LP problem is transformed into a standard LP problem:

$$\min \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} - \mathbf{y} = [A \quad -I_m] \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}.$$

Note that the vector of variables is $(\mathbf{x}, \mathbf{y})$ but the cost is the same as above, $\mathbf{c}^T\mathbf{x} + \mathbf{0}^T\mathbf{y} = \mathbf{c}^T\mathbf{x}$.

If the problem is in the non-standard form:

$$\min \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \leq \mathbf{b} \ \text{ with } \ \mathbf{x} \geq \mathbf{0},$$

we need to define positive variables $y_i, i = 1, 2 \ldots, m$, to add to each constraint, which we call *slack variables* and $\mathbf{y} \geq \mathbf{0}$. The new form of the problem is:

$$\min \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} + \mathbf{y} = [A \quad I_m] \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0}, \, \mathbf{y} \geq \mathbf{0}.$$

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 13 LP in standard form, LP BFSs - 13/11/06

Notes by : Charalampos Daskalakis and Constantinos Houmas

1 Exercise 15.10. Expressing a minimization problem in its typical form

We start from the problem:

$$\begin{aligned} max \quad & x_2 - x_1 \\ s.t. \quad & 3x_1 = x_2 - 5 \\ & |x_2| \leq 2 \\ & x_1 \leq 0 \end{aligned}$$

In order to bring an LP problem to a standards form, we change "max" to "min" and we need to transform all variables to non-negative ones. Also we transform all inequalities in the problem into equalities. So, the variable x_1 will be replaced by the variable $x_1' = -x_1$ and the two inequalities implied by $|x_2| \leq 2$ will be converted to equalities using slack variables x_3 and x_4. Now the problem is expressed as:

$$\begin{aligned} min \quad & -x_2 - x_1' \\ s.t. \quad & -3x_1' = x_2 - 5 \\ & x_2 + x_3 = 2, \ \ -2 + x_4 = x_2 \\ & x_3, x_4, x_1' \geq 0 \end{aligned}$$

In the original form of the problem, we have inequality $-2 \leq x_2 \leq 2$, x_2 should be redefined as $u - v$ with $u, v \geq 0$. The reason is that a variable which is unrestricted in sign (such as x_2) can be written in general as the difference of two positive variables. So in its standard form, the problem above becomes:

$$
\begin{aligned}
min \quad & -x_2 - x_1' \\
s.t. \quad & 3x_1' = 5 - x_2 \\
& u - v + x_3 = 2, \;\; v - u + x_4 = 2 \\
& x_3, x_4, x_1', v, u \geq 0
\end{aligned}
$$

2 Routing as an LP problem

Routing in communication networks means selecting paths for transferring traffic from given source(s) to given destination(s).

Consider a network which is abstracted as a directed graph $G(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of edges of the graph. Let $\mathcal{N} = n$ be the number of nodes of the network. For each edge $(i, j) \in \mathcal{A}$ we define u_{ij} to be the *capacity* of edge junction (i, j), which is the maximum amount of traffic (in bps) that can be carried over the edge. Also, let c_{ij} be the cost per unit of transmitted traffic over the edge (i, j).

For each source k and destination l, define as b^{kl} the amount of traffic (bps) that is generated by node k and needs to be transferred to l. In this example, all nodes may be sources and destinations (if not, then $b^{kl} = 0$). **Problem:**

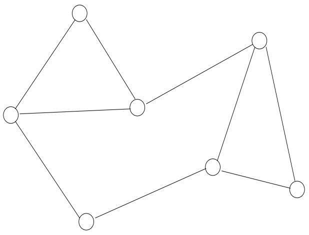


Figure 1: A graph, depicting a network (note here the graph is directed).

Choose the paths for routing traffic for each source $k = 1 \ldots n$ *to each destination* $l = 1 \ldots n$ *while minimizing total cost.*

The variables in this problem are defined as x_{ij}^{kl}, the amount of traffic with origin k and destination l that traverses link $(i, j) \in \mathcal{A}$. For each node $i = 1 \ldots n$ we define:

$$
b_i^{kl} = \begin{cases} b^{kl} & if \;\; i = k \\ -b^{kl} & if \;\; i = l \\ 0 & \text{otherwise} \end{cases} \tag{1}
$$

Thus, b_i^{kl} denotes the net inflow at node i of traffic originated at k and destined at l.

So we have the problem:

$$\begin{array}{lll}
min & \displaystyle\sum_{(i,j)\in\mathcal{A}}\sum_{k=1}^{n}\sum_{l=1}^{n} c_{ij}\ x_{ij}^{kl} & \\
s.t. & x_{ij}^{kl} \geq 0 & \forall k,l = 1\ldots n \text{ and } \forall (i,j)\in\mathcal{A} \\
 & \displaystyle\sum_{j:(i,j)\in\mathcal{A}} x_{ij}^{kl} - \sum_{j:(i,j)\in\mathcal{A}} x_{ji}^{kl} = b_i^{kl} & \forall i = 1\ldots n \\
 & \displaystyle\sum_{k=1}^{n}\sum_{l=1}^{n} x_{ij}^{kl} \leq u_{ij} & \forall (i,l)\in\mathcal{A}
\end{array}$$

In the problem there is a set of constraints, one for each node that reflect the *flow conservation constraint* at each node. Also, there is a constraint for each link that denotes the capacity constraint for each link.

The problem is called *minimum cost network flow problem*, and as we will see later, there are several known problems that emerge as special cases of this, such as the shortest path, the Max flow and the assignment problem.

3 BFS (Basic Feasible Solution)

Consider the system of inequalities $A\mathbf{x} = \mathbf{b}$, with $\mathbf{x} \geq 0$ and matrix A of dimension $m \times n$, $m \leq n$ and $rank(A) = m$. Matrix A can be written in a block matrix form as $A = [B\ \ D]$, where (i) the $m \times m$ matrix B includes all m linearly independent columns of A and (ii) the $m \times (n-m)$ matrix D includes the rest of the columns of A.

By definition, B is non-singular ($|B| \neq 0$) where $|B|$ is the determinant of matrix B. Then, matrix B is said to be the *basis* for the system. The columns of B called *basic columns*. Then, the system of equations $B\mathbf{x}_B = \mathbf{b}$ has a unique solution, $\mathbf{x}_B = B^{-1}\mathbf{b}$. Vector $\mathbf{x}_B$ is of dimension $m \times 1$ and consists of those variables that correspond to the columns of B (these are called *basic variables*.

Thus, for example if A is 2×4 and the first and third column of A are linearly independent, then $\mathbf{x}_B = (x_1, x_3)^T$.

Note that vector $\mathbf{x} = (\mathbf{x}_B\ \ \mathbf{0})^T$ solves the original system $A\mathbf{x} = \mathbf{b}$.

Definition 1: We call a vector of the form $(\mathbf{x}_B\ \ \mathbf{0})^T$ a *Basic solution* with respect to the basis B. Thus, all vectors of values variables that can be divided into a non-zero and a zero variable part (the non-zero part corresponding to the columns of a basis) are called *Basic solutions* (Note here that this definition does not imply feasibility).

Defintion 2: A basic solution $\mathbf{x}$ that satisfies $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ (i.e. it is feasible) is called *Basic Feasible Solution (BFS).*

If the BFS $\mathbf{x}_B > \mathbf{0}$ has all m components positive, the BFS is called *non-degenerate BFS.* Otherwise, if some of the components of $\mathbf{x}_B$ are zero (i.e the positive components of $\mathbf{x}_B$ are fewer than m, the BFS is called *degenerate BFS.*

An alternative definition: Consider an LP problem with constraints $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$ and A a $m \times n$ matrix, $m \leq n$ (with m linear independent rows). Then $\mathbf{x} \in \mathcal{R}^n$ is a *Basic Feasible Solution* if there exist indices $B(1), ..., B(m)$ such that columns $\mathbf{a}_{B(1)}, \mathbf{a}_{B(2)}, \ldots, \mathbf{a}_{B(m)}$ of matrix A are linearly independent and $\forall i \neq B(1), B(2), ..., B(m)$ it is $x_i = 0$. Also it should be $A\mathbf{x} = \mathbf{b}$. In addition if $x_i > 0 \ \ \forall i \in \{B(1), B(2), ..., B(m)\} \Rightarrow \mathbf{x}$ is a non-degenerate BFS.

An Example Consider the set of constraints:

$$\begin{array}{ccccccc} x_1 & + & x_2 & + & 2x_3 & \leq 8 \\ & & x_2 & + & 6x_3 & \leq 12 \\ x_1 & & & & & \leq 4 \\ & & x_2 & & & \leq 6 \\ x_1 & , & x_2 & , & x_3 & \geq 0 \end{array}$$

After converting them to a standard form, we get:

$$\begin{array}{ccccccccc} x_1 & + & x_2 & + & 2x_3 & + & x_4 & = 8 \\ & & x_2 & + & 6x_3 & + & x_5 & = 12 \\ x_1 & & & & & + & x_6 & = 4 \\ & & x_2 & & & + & x_7 & = 6 \\ x_1 & , & x_2 & , & \ldots & , & x_7 & \geq 0 \end{array}$$

So the constraints correspond to the following representation: $A\mathbf{x} = \mathbf{b}$ where:

$\mathbf{x} = [x_1, x_2, \ldots, x_7]^T$,

$\mathbf{b} = [8, 12, 4, 6]^T$,

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let the basis be $B = [\ \mathbf{a}_1\ \mathbf{a}_3\ \mathbf{a}_4\ \mathbf{a}_7\ ]$. If we consider the system of equations $B\mathbf{x}_B = \mathbf{b}$, the solution to that is $\mathbf{x}_B = [4, 2, 0, 6]^T$ and $\mathbf{x} = [4, 0, 2, 0, 0, 0, 6]^T$ is a BFS which is degenerate (since $x_4 = 0$).

- In the case that we choose the basis to be $B = [\ \mathbf{a}_4\ \mathbf{a}_5\ \mathbf{a}_6\ \mathbf{a}_7\ ]$ then we have solution $\mathbf{x}_B = [8, 12, 4, 6]^T$ and $\mathbf{x} = [0, 0, 0, 8, 12, 4, 6]^T$ is a BFS that is non-degenerate.

Clearly, there are several ways of choosing the basis for an LP problem. For a matrix A of dimension $m \times n$, $m \leq n$ with $rank(A) = m$, we can choose among $C(n, m) = \frac{n!}{(n-m!)m!}$ different bases B, and so we can have so many basic solutions.

The basic theorem in LP is that in order to solve a LP problem, we will only need to check the BFS and among them find the optimal BFS, i.e the BFS that minimizes $\mathbf{c}^T\mathbf{x}$.

Theorem 15

- (a) If there exists a feasible solution in an LP problem, then there exists a BFS.
- (b) If there exists an optimal feasible solution in an LP problem, then there exists an optimal BFS.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 14 : Rationale of Simplex algorithm for LP problems - 14/11/06

Notes by : Anastasia Narou and Maria Papadopoulou

1 Introduction

Definition: Consider a polyhedron $\mathcal{P}$, defined by linear equality and inequality constraints as defined in the first lectures. Then, the point (represented by vector) $\mathbf{x}_0$ is a *vertex* of $\mathcal{P}$ if there exists $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{x}_0 < \mathbf{c}^T\mathbf{y}$ for all $\mathbf{y} \in P$ with $\mathbf{y} \neq \mathbf{x}_0$.

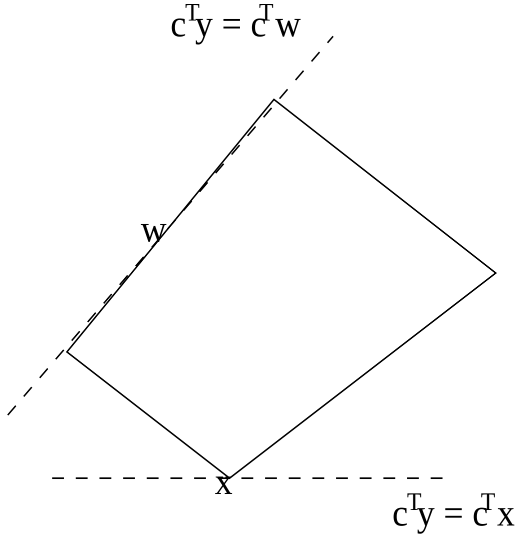


Figure 1: A vertex point **x** and a point **w** that is not a vertex.

As shown in figure 1, **w** in the figure is not a vertex because there is no hyperplane that meets $\mathcal{P}$ only at **w**. That is, a point **w** of a polyhedron is a vertex of the polyhedron if and only if a hyperplane passing through the point divides the space in two half-spaces and *all points of the hyperplane except from* $\mathbf{w}$ *lie on the same side (same half-space)*. In the example of figure 1, there exist all the points $\mathbf{y}$ on the hyperplane $\mathbf{c}^T\mathbf{y} = \mathbf{c}^T\mathbf{w}$ that make the definition of $\mathbf{w}$ being vertex of $\mathcal{P}$ not hold.

Defintion: A point $\mathbf{x}$ is called *extreme point* of $\mathcal{P}$ if there are no distinct points $\mathbf{x}_1$,$\mathbf{x}_2$ of $\mathcal{P}$ such that $\mathbf{x} = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$. In other words, if $\mathbf{x}$ is an extreme point and it is $\mathbf{x} = \alpha\mathbf{x}_1 + (1-\alpha\mathbf{x}_2)$ for some $\alpha \in (0,1)$, then it must be $\mathbf{x} = \mathbf{x}_1 = \mathbf{x}_2$. Thus, an extreme point of a polyhedron cannot be represented as a convex combination of two other points of the polyhedron.

Theorem (Fundamental theorem of Linear Programming) For a linear programming problem with constraints that define the polyhedron $\mathcal{P}$ of feasible points we have: The point $\mathbf{x}_0$is a vertex of $\mathcal{P} \Leftrightarrow \mathbf{x}_0$ is an extreme point of $\mathcal{P} \Leftrightarrow \mathbf{x}_0$ is a BFS (basic feasible solution) of the LP problem.

We now demonstrate part of the proof which will help us in the subsequent discussion about the Simplex Algorithm. We will show that if a point $\mathbf{x}$ is an extreme point of the polyhedron of feasible points $\mathcal{P}$, then $\mathbf{x}$ is BFS of the LP problem.

Assume that $\mathbf{x}$ is extreme point of $\mathcal{P}$. Then it satisfies $\mathbf{x} \in \mathcal{P}$ and $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. Assume that $\mathbf{x}$ is of the form $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_p, 0, \dots, 0)^T$, with $p \leq n$. Namely it has some of its elements non-zero. Point $\mathbf{x}$ satisfies the equation

$$x_1\mathbf{a}_1 + \ldots + x_p\mathbf{a}_p = \mathbf{b} \tag{1}$$

where $\mathbf{a}_i$ is the ith column of matrix A. Note that A can be written as $A = [\mathbf{a}_1 \ldots \mathbf{a}_n]$.

Define numbers y_i, for $i = 1, ..., p$ such that:

$$y_1\mathbf{a}_1 + \ldots + y_p\mathbf{a}_p = \mathbf{0}. \tag{2}$$

In order to show that $\mathbf{x}$ is a BFS of the LP problem, it suffices to show that columns $\mathbf{a}_i, i = 1 \ldots p$ of matrix A are linearly independent. If they are, then they will form a basis and the solution $\mathbf{x} = (x_1, \dots, x_p)$ will be a BFS.

To show $\mathbf{a}_i, i = 1 \ldots p$ of matrix A are linearly independent, we will show that $y_i = 0$ for $i = 1, \dots, p$. We multiply (2) by $\varepsilon > 0$ and add and subtract it from (1). We get:

$$(x_1 + \varepsilon y_1)\mathbf{a}_1 + \ldots + (x_p + \varepsilon y_p)\mathbf{a}_p = \mathbf{b}, \tag{3}$$

and

$$(x_1 - \varepsilon y_1)\mathbf{a}_1 + \ldots + (x_p - \varepsilon y_p)\mathbf{a}_p = \mathbf{b}. \tag{4}$$

Since $x_i > 0$, $\varepsilon > 0$ can be we chosen arbitrarily small arbitrary very small so that $x_i + \varepsilon y_i \geq 0$, and $x_i - \varepsilon y_i \geq 0$. It can be easily deduced that we can choose

$$\varepsilon = \min\left\{\left|\frac{x_i}{y_i}\right|, i = 1, \ldots, p, y_i \neq 0\right\}. \tag{5}$$

Then, for the vectors

$$\mathbf{z}_1 = (x_1 + \varepsilon y_1, \ldots, x_p + \varepsilon y_p, 0, \ldots, 0) \tag{6}$$

and

$$\mathbf{z}_2 = (x_1 - \varepsilon y_1, \ldots, x_p - \varepsilon y_p, 0, \ldots, 0) \tag{7}$$

we have $A\mathbf{z}_1 = \mathbf{b}$, $A\mathbf{z}_2 = \mathbf{b}$, $\mathbf{z}_1, \mathbf{z}_2 \geq \mathbf{0}$, $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{P}$.

Now observe that $\mathbf{x} = \frac{1}{2}\mathbf{z}_1 + \frac{1}{2}\mathbf{z}_2$, but the fact that $\mathbf{x}$ is extreme point results in $\mathbf{x} = \mathbf{z}_1 = \mathbf{z}_2$ which leads to $y_i = 0$. Thus vectors $\mathbf{a}_1, \ldots, \mathbf{a}_p$ are linearly independent and thus $\mathbf{x}$ is a BFS.

As a result of the theorem : if we want to solve an LP problem, we need to search for the optimal solution only among the extreme points of $\mathcal{P}$.

2 Useful facts

Assume we have the system of linear equations $A\mathbf{x} = \mathbf{b}$. Let matrix A have some m linearly independent columns, A is of dimension $m \times n$, $m < n$, and $\text{rank}(A) = m$. Denote by B the submatrix that consists of these columns. Let D be the submatrix with the rest of the columns. The augmented matrix of this system is $[A\ \mathbf{b}]$. We want to bring this matrix to the form $[I\ D\ \tilde{\mathbf{b}}]$.

We know from the methodology of solving linear systems of equations that the augmented matrix can be brought in that form with elementary operations on it:

- Interchanging any two rows of the matrix,
- Multiplying one of its rows by a real, non-zero number
- Multiplying one of its rows by a real, non-zero number and adding to another row

If the augmented matrix is brought in that form, then $\mathbf{x} = \tilde{\mathbf{b}}$ is the solution of the linear system $A\mathbf{x} = \mathbf{b}$. Because A is of dimension $m \times n$, $m < n$, and $\text{rank}(A) = m$ the system has infinite solutions.

Matrix A is brought in the form $[I\ D\ \tilde{\mathbf{b}}]$ and then the linear system of equations can be written as:

$$x_1 + y_{1,m+1}x_{m+1} \cdots y_{1n}x_n = y_{10} \tag{8}$$

$$x_2 + y_{2,m+1}x_{m+1} + \ldots + y_{2n}x_n = y_{20} \tag{9}$$

$\vdots$

$$x_m + y_{m,m+1}x_{m+1} + \ldots + y_{mn}x_n = y_{m0} \tag{10}$$

where the first factor in each equation is reflected in the unit matrix I, the remaining factors in each equation represent matrix D and the right side of the equations represent $\tilde{\mathbf{b}}$.

2.1 Example

We have the following constraints:

$$x_1 + x_2 - x_3 + 4x_4 = 8 \tag{11}$$

$$x_1 - 2x_2 - x_3 + x_4 = 2 \tag{12}$$

The augmented matrix is:

$$\begin{pmatrix} A \, \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 & 4 & | & 8 \\ 1 & -2 & -1 & 1 & | & 2 \end{pmatrix}$$

Multiply the first row by -1 and add to the second row:

$$\begin{pmatrix} A \, \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 & 4 & | & 8 \\ 0 & -3 & 0 & -3 & | & -6 \end{pmatrix}$$

Divide the second row by -3:

$$\begin{pmatrix} A \, \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 & 4 & | & 8 \\ 0 & 1 & 0 & 1 & | & 2 \end{pmatrix}$$

Multiply the second row by -1 and add to the first row.

$$\begin{pmatrix} A \, \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 & 3 & | & 6 \\ 0 & 1 & 0 & 1 & | & 2 \end{pmatrix}$$

If we choose as basis matrix $B = [\mathbf{a}_1, \mathbf{a}_2]$, an obvious solution is $\mathbf{x} = (6, 2, 0, 0)$. This solution is feasible (since it satisfies $A\mathbf{x} = \mathbf{b}$), basic and non-degenerate.

If we choose as basis $B = [\mathbf{a}_3, \mathbf{a}_4]$, the solution $\mathbf{x} = (0, 0, 0, 2)$ is feasible, basic and degenerate. Another possible solution is $\mathbf{x} = (3, 1, 0, 1)$ which is feasible but not basic.

If we choose as basis $B = [\mathbf{a}_2, \mathbf{a}_3]$, the solution $\mathbf{x} = (0, 2, -6, 0)$ is basic but non-feasible, since we do not admit negative solutions.

3 Introduction to Simplex Algorithm

In order to solve a Linear Programming problem, we move from one BFS to another BFS until we find the optimal one, which will be the one with the property that if I try to move to whatever other BFs, the value of the objective function is not improved. This is precisely what the Simplex algorithm does.

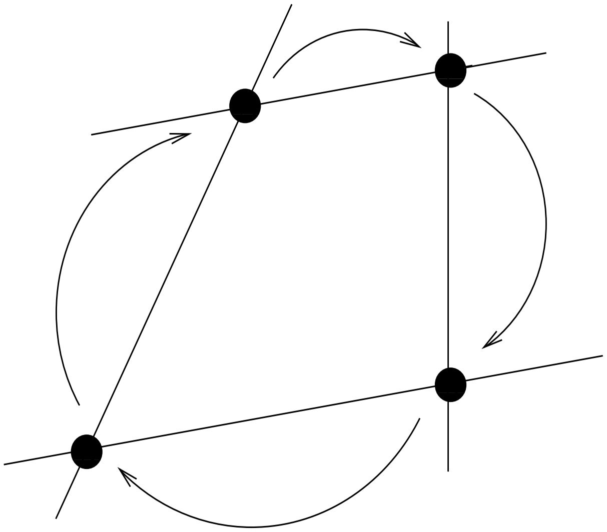


Figure 2: In LP, the Simplex algorithm moves from one BFS to another.

In the following, we will assume that we have non-degenerate solutions. We will treat the cases of degenerate solutions separately.

A given BFS at some step of the algorithm

Consider that at some stage of the algorithm, we have the BFS

$$\mathbf{x} = (y_{10}, \ldots, y_{m0}, 0, \ldots, 0), \; y_{i0} > 0, i = 1, \ldots, m. \tag{13}$$

The way, we move from one BFS (vertex, or extreme point of the polyhedron defined by the constraints $A\mathbf{x} = \mathbf{b}$) to another is as follows: In each step we change a non-basic variable to basic and a basic variable to non-basic. This operation is called *pivoting*. The non-basic variables are set to zero while the basic variables are found to be non-negative values. Talking in columns, we insert to the basis a column that is currently not in the basis and we take out of the basis a column that used to be in the basis.

Suppose we have chosen the column $\mathbf{a}_q$, $q > m$ (non-basic column now) and we want to have it inside the base. We have the column $\mathbf{a}_q$ as a linear combination of the current basis $\{\mathbf{a}_1, \ldots, \mathbf{a}_p\}$:

$$\mathbf{a}_q = y_{1q}\mathbf{a}_1 + y_{2q}\mathbf{a}_2 + \ldots + y_{mq}\mathbf{a}_m \tag{14}$$

where matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_m, \mathbf{a}_{m+1}, \ldots, \mathbf{a}_q, \ldots, \mathbf{a_n}]$. Also note that $B = [\mathbf{a}_1 \ldots \mathbf{a}_m]$ and $D = [\mathbf{a}_{m+1}, \ldots, \mathbf{a}_n]$

We want to move the non-basic column $\mathbf{a}_q$ in the basis. Multiply the left- and the right-hand side of the equation above with $\varepsilon > 0$ to get:

$$\varepsilon\mathbf{a}_q = \varepsilon(y_{1q}\mathbf{a}_1 + \ldots + y_{mq}\mathbf{a}_m) \tag{15}$$

Now we know that the current BFS (the vector $\tilde{\mathbf{b}}$ we saw before) satisfies $A\mathbf{x} = \mathbf{b}$ and can be written as a linear combination of the basic columns as:

$$y_{10}\mathbf{a}_1 + \ldots + y_{m0}\mathbf{a}_m = \mathbf{b} \tag{16}$$

Subtract the one equation from the other to get:

$$(y_{10} - \varepsilon y_{1q})\mathbf{a}_1 + (y_{20} - \varepsilon y_{2q})\mathbf{a}_2 + \ldots + (y_{m0} - \varepsilon y_{mq})\mathbf{a}_m + \varepsilon\mathbf{a_q} = \mathbf{b} \tag{17}$$

Notice that since the equation $A\mathbf{x} = \mathbf{b}$ is satisfied again, the coefficients correspond to a new solution,

$$\begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ y_{20} - \varepsilon y_{2q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \varepsilon \\ 0 \end{pmatrix}$$

where ε appears in the q-th position, $q > m$.

This operation can be understood as follows: currently $x_q = 0$. Assume we start increasing the (currently non-basic) variable x_q to some positive value ε, so as to make it basic. Equivalently, column $\mathbf{a}_q$ will enter the basis. When ε increases, then q-th component increases too. The variables $x_1, \ldots, x_m$ 1 decrease if $y_{iq} > 0$ and increase if $y_{iq} < 0$.

The question that arises now is: Up to which value ε can x_q be increased so that we go from one BFS to another? The answer is the following: the maximum value that ε can take is determined by that component of the solution that becomes zero first. This value of ε is clearly

$$\varepsilon = \min_i \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0 \right\} \tag{18}$$

Suppose that p is the index that is first zeroed, $1 \leq p \leq m$. Then, p corresponds to the basic variable that will now become non-basic. Specifically, it is

$$p = \arg \min_{i=1,\dots,m} \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0 \right\} \tag{19}$$

Therefore, variable x_p now becomes zero, or equivalently column $\mathbf{a}_p$ exits the basis. The new basis is therefore,

$$\{\mathbf{a}_1, \dots, \mathbf{a}_{p-1}, \mathbf{a}_{p+1}, \dots, \mathbf{a}_m, \mathbf{a}_q\} \tag{20}$$

and the new BFS is:

$$\begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ \vdots \\ y_{p-1,0} - \varepsilon y_{p-1,q} \\ 0 \\ y_{p+1,0} - \varepsilon y_{p+1,q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix}$$

with ε in the q-th position.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 15 : Simplex algorithm and an example - 15/11/06

Notes by : Eleni Anagnostopoulou and Nena Xanthopoulou

1 Outline of lecture:

- Case of degenerate BFS
- Critical questions:
  - When does Simplex algorithm stop?
  - How can I choose which non-basic variable will become basic variable?
- Simplex algorithm steps
- Example

In the previous lecture, we saw the switch from one BFS to another BFS, or in other words specified what it means for a currently non-basic column $\mathbf{a}_q$ to enter the basis and for a currently basic column $\mathbf{a}_p$ to exit the basis. The Simplex Algorithm that we will see in today's lecture uses this principle of moving among different BFSs.

2 Cases of degenerate BFS

It may happen that the new BFS is degenerate, namely a variable that is basic is zero. This can happen in the following cases:

1. The coefficients are such that two basic variables can become zeroe when we want to change the base. For the example above, we may have two indices $0 \leq p_1, p_2 \leq m$ such that $x_{p_1} = 0$ and $x_{p_2} = 0$.
2. It may happen that $\varepsilon = 0$. Then, the variable we want to turn to basic and place it in the basis cannot take a larger value and the BFS remains the same. When the current BFS $\mathbf{x}$ is degenerate, then it may be that $\varepsilon = 0$ and the new BFS remains the same as the current BFS (especially this is the case if some basic variable is 0 and the corresponding denominator is positive).
3. Cycling phenomenon : It may happen that the a sequence of basis changes lead us through a sequence of changes of bases back to the initial basis.

Also note that in the case that none of coefficients y_{iq} is positive, we can move further from the current BFS, but we cannot discover any new BFS for the problem ⇒ the polyhedron P is unbounded and the LP problem is said to be unbounded.

3 Critical questions

3.1 When does Simplex algorithm stop?

Suppose that we have a BFS

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} y_{10} \\ y_{20} \\ \vdots \\ y_{m0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

What is the cost of the solution? In other words, we find the value of the objective function $\mathbf{c}^T\mathbf{x}$ for this solution. We have the value

$$z = \mathbf{c}^T\mathbf{x} = c_1 y_{10} + c_2 y_{20} + \ldots + c_m y_{m0} = \mathbf{c}_B^T\mathbf{x}, \tag{1}$$

where $\mathbf{c}_B^T$ is the part of the cost vector $\mathbf{c}$ that corresponds to the basis. Suppose we get to a new BFS,

$$\mathbf{x}^{'} = \begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ y_{20} - \varepsilon y_{2q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix}$$

where ε comes at the q-th component of the solution. The new cost is:

$$z^{'} = \sum_{i=1}^{m} c_i(y_{i0} - \varepsilon y_{iq}) + c_q \varepsilon, \tag{2}$$

where q denotes the new variable x_q that became a basic variable. We have

$$z^{'} = z + \varepsilon[c_q - (c_1 y_{1q} + c_2 y_{2q} + \ldots + c_m y_{mq})]. \tag{3}$$

Now set

$$z_q = c_1 y_{1q} + c_2 y_{2q} + \ldots + c_m y_{mq}. \tag{4}$$

Then we get

$$z^{'} = z + \varepsilon(c_q - z_q) \tag{5}$$

or $z^{'} - z = \varepsilon(c_q - z_q)$.

In order for the new solution to be "better" than the current one, its objective function value has to be smaller than the one of the current solution. If $z^{'} - z < 0$, then the new BFS (that corresponds to the non-basic column $\mathbf{a}_q$ entering the basis) has a lower objective function value. Since $\varepsilon > 0$, this happens when $c_q - z_q < 0$ is true. Define $c_q - z_q = r_q$ as the *reduced cost coefficient* corresponding to the newly entered variable x_q. If $c_q - z_q < 0$ then by entering column $\mathbf{a}_q$ in the basis, we have arrived to a better BFS (one with a lower cost).

Fact: The solution

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ 0 \end{pmatrix}$$

is *optimal* if $\forall q = m+1,\dots,n$ it is $c_q - z_q > 0$. In other words, this means that I am currently at the optimal BFS if I try to change the basis by all possible means (i.e, insert to the basis *any* non-basic column $\mathbf{a}_q$) and I will never manage to reduce the value of the objective function. Note that the reduced cost variables $c_q - z_q = r_q$ are defined for each variable. As an exercise, we can easily show the following:

Problem: Show that for the basic variables $x_i,\ i = 1,\dots,m$, it is $r_i = 0$.

Remark: The change of basis performed between columns $\mathbf{a}_p$ and $\mathbf{a}_q$ is called (p,q) *pivoting operation.*

3.2 How do I choose which non-basic variable will become basic?

How can I select the column q that will enter the base? There are three possible ways to do that:

1. If $z' - z = \epsilon(c_q - z_q)$, we choose q such that the difference $c_q - z_q$ takes the minimum value (the rate of cost reduction becomes as large as possible). To be more specific,
$$q = \arg \max_{l \in \{m+1,\dots,n\}} |r_l| = \arg \min_{l \in \{m+1,\dots,n\}} r_l. \tag{6}$$
Thus, we choose to make basic the variable that leads to the largest rate of cost reduction (cost reduction per unit of non-basic variable increase, $c_q - z_q = \frac{z'-z}{\varepsilon}$).

2. Choose
$$q = \arg \max_{l \in \{m+1,\dots,n\}} \varepsilon_l |r_l| = \arg \min_{l \in \{m+1,\dots,n\}} \varepsilon_l r_l. \tag{7}$$
In this way of choosing q we take into account the total change is minimum, which also depends on ε. Note that ε_l is the value of ε that turns the non-basic variable x_l to basic and thus clearly ε has a different value for different q.

3. Choose $q = \arg\min_{i \in \{m+1,\dots,n\}} \{r_i : r_i < 0\}$. That is, we examine all non-basic variables starting from the lowest-indexed one and select to place at the basis the first one that has negative reduced cost coefficient. The disadvantage with this approach is that it does not guarantee that the value of the objective function at the new BFS will be the smallest possible.

If we choose q in that fashion (3) above, and choose $p = \min\{j : \frac{y_{jo}}{y_{jq}}\} = \min_i\{\frac{y_{io}}{y_{iq}}\} : y_{iq} > 0\}$ (in other words, we choose to put out of the basis the lowest-indexed variable out of the ones that become 0), then, even though I have a non-degenerate new BFS, the cycling phenomenon mentioned above is avoided. This is known as *lexicographic pivoting rule.*

4 Simplex algorithm : steps

1. Begin with an initial BFS. If the LP problem is defined using inequalities, we define slack variables and bring it to the standard form and find the BFS.
2. Calculate the coefficients r_q for each non-basic variable x_q.
3. - If $r_q \geq 0$ for all non-basic variables, then STOP the algorithm. We have found the optimal solution.
   - Else, choose q according to one of the rules that we described in question (2) previously. If none of $y_{iq} > 0$ then STOP(unbounded LP problem) Else, calculate $p = \arg\min_i\{\frac{y_{i0}}{y_{iq}} : y_{iq} > 0\}$ (this selection rule for the variable that exits the basis eliminates the cycling phenomenon).
4. Pivot(p,q) and find new BFS.
5. Go to step 2.

5 Example of Simplex Algorithm for an LP problem

Given the LP problem

$$\max 7x_1 + 6x_2$$

subject to the constraints:

$$\begin{aligned} 2x_1 + x_2 &\leq 3 && (8) \\ x_1 + 4x_2 &\leq 4 && (9) \\ x_1, x_2 &\geq 0. && (10) \end{aligned}$$

solve it (find the optimal BFS).

Solution: We convert the problem to the standard form by defining slack variables x_3, x_4, so the new problem (P) is:

$$\min -7x_1 - 6x_2$$

subject to:

$$2x_1 + x_2 + x_3 = 3 \tag{11}$$

$$x_1 + 4x_2 + x_4 = 4 \tag{12}$$

$$x_1, x_2, x_3, x_4 \geq 0 \tag{13}$$

The objective function value z is : $z = -7x_1 - 6x_2 + 0x_3 + 0x_4$.

We start running the Simplex Algorithm. Start with initial BFS: $\mathbf{x} = (0, 0, 3, 4)$. The basis is $B = [\mathbf{a}_3, \mathbf{a}_4]$.

At each step, we will express the cost as a function of the non-basic variables. We will also write the basic variables as functions of the non-basic variables to facilitate computation of ε and the pivoting.

Initial cost:$z = -7x_1 - 6x_2 + 0x_3 + 0x_4$ with value $z_0 = 0$ for the current BFS. Write the basic variables as functions of the non-basic ones:

$$x_3 = 3 - 2x_1 - x_2 \tag{14}$$

$$x_4 = 4 - x_1 - 4x_2 \tag{15}$$

STEP 1: Our goal is to change the basis, so that we find a new BFS with lowest cost value. The question is "which (non-basic) variable x_p should we choose to make basic?". We choose it according to the first rule case out of the three we described at question (2) in this lecture.

We see that $r_1 = -7$, $r_2 = -6$ (reduced cost coefficients). In other words, if I increase x_1 or x_2 by making one of the two basic, I observe that the increase of x_1 causes the largest decrease in z. That is, since $|r_1| > |r_2|$, we choose to make variable x_1 basic (equivalently put the first column $\mathbf{a}_1$ of matrix A in the basis. Thus it is $q = 1$. If we increase x_1, we observe that, out of the basic variables x_3, x_4, the first that becomes zero is x_3 and this occurs for $x_1 = \varepsilon = 3/2$. Thus $p = 3$ and x_3 will become non-basic.

Pivot$(3, 1)$. New basis: $B = [\mathbf{a}_1, \mathbf{a}_4]$

New BFS: $\mathbf{x} = (\frac{3}{2}, 0, 0, \frac{5}{2})$ and new cost value: $z = -\frac{21}{2}$ (observe that we reduced the value of the objective).

We now write the new basic variables as functions of non-basic variables:

$$x_1 = \frac{3}{2} - \frac{1}{2}x_2 - \frac{1}{2}x_3 \tag{16}$$

$$x_4 = \frac{5}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 \tag{17}$$

and the cost:

$$z = -7x_1 - 6x_4 = -\frac{21}{2} - \frac{5}{2}x_2 + \frac{7}{2}x_3 \tag{18}$$

STEP 2: Now again we will have to choose which non-basic variable to make basic. Observe that if we make variable x_3 basic, the cost will be increased, which is undesirable. So we choose to make basic the variable x_2. Thus $q = 2$.

If we increase x_2, we observe that, out of the basic variables x_1, x_4, the first that becomes zero is x_4 and this occurs for $x_2 = \varepsilon = 5/7$. Thus $p = 4$ and x_4 will become non-basic.

Pivot$(4, 2)$. New basis: $B = [\mathbf{a}_1, \mathbf{a}_2]$

New BFS: $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$ and new cost value: $z_2 = -\frac{81}{7}$ (observe that with the change of basis we have reduced the objective function value more). From equations

Again, we write the basic variables as a function of non-basic variables.

$$x_1 = \frac{8}{7} - \frac{4}{7}x_3 + \frac{1}{7}x_4 \tag{19}$$

$$x_2 = \frac{5}{7} + \frac{1}{7}x_3 - \frac{2}{7}x_4 \tag{20}$$

and the cost:

$$z = -7x_1 - 6x_2 = -\frac{86}{7} + \frac{22}{7}x_2 + \frac{5}{7}x_4 \tag{21}$$

STEP 3: Now gain we will have to choose which non-basic variables to make basic. However, observe that if make either x_3 or x_4 basic (i.e try to increase them from 0), the cost value will be increased. So the algorithm stops here and we say that we found the optimal BFS and we solved the LP problem.

Optimal solution: $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 16 : 27/11/06

Notes by : Evaggelos Galanis and Panagiotis Theodosiou

1 Outline Of Lecture:

- Wireless Sensor Networks - an LP problem
- Carrier assignment in OFDM - an LP problem

2 Maximum lifetime routing in wireless sensor networks

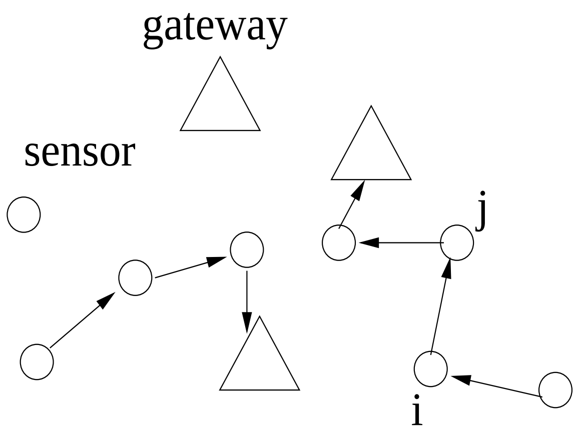


Figure 1: A wireless sensor network with a set of sensor nodes and a set of gateways.

We will now see an example of a problem from wireless sensor networks, that is formulated as an LP problem.

A sensor network consists of a set of miniature-sized sensor nodes that sense and monitor processes such as vibration, sound, light, temperature, movement. The information needs to be transferred from the sensors to a set of information processing centers, called gateways. The data will be transferred with multi-hop routing as shown in figure 1. Of course the different routes may intersect (something that is not shown in the figure). Sensor nodes have the ability to forward their data, as well as other nodes' data.

All sensors can potentially produce information (data. Gateways are connected with optical fiber and we will assume that the data from a sensor will have reached its destination if it reaches *any* of the gateways. We will also not deal with interference from multiple ongoing transmissions (which can be here assume to be reduced or eliminated by means of an appropriate scheduling protocol). We assume there exist several *kinds* (or *commodities*) of traffic (e.g. temperature, sound, etc)

The transmitted energy from a sensor node can be adjusted to a level appropriate for a receiver within its transmission range to the able to receive the data correctly. We will discuss later this issue.

Upon arrival of new information at a node (either generated by the node itself or forwarded from other nodes) a routing decision needs to be made so that the data is forwarded to an appropriate neighbor. We will see that routing accounts to finding the way to split the traffic streams across different routes, so as to "balance" energy consumption among nodes and thus increase network lifetime.

The topology is considered to be static, namely with no mobility. Note that mobility either of the sensor nodes or of the gateways or both may further help in reducing the energy consumption and improve network lifetime. However, we will not consider such an issue here.

We define the following quantities:

P_{ij} : the minimum power needed for sensor i to send information to to sensor j. This os proportional to the distance d_{ij} between the nodes i and j and is given as $P_{ij} = \gamma d_{ij}^a$, where a is a constant that specifies the type of wireless propagation environment and γ is the minimum required SNR at the receiver such that reception is acceptable. sensors

Specifically, at the receiver we have SNR $\geq \gamma \Rightarrow \dfrac{P_{ij}}{d_{ij}^a \sigma^2} \geq \gamma$, and thus the minimum power is $P_{ij} = \gamma d_{ij}^a \sigma^2$. Note that σ^2 is the noise power.

e_{ij} : The amount of energy consumed by sensor i for the transport of one unit of information (bit or packet) to sensor j. It is measured in Joules per bit. Now, we determine e_{ij} in more detail and note that e_{ij} is known to each sensor i only for its neighbors j (let $\mathcal{N}_i$ be the set of neighbors of sensor node i).

Energy and power are related as $E = Pt$, which has units (energy/bit) = (power/bit) × (sec). We understand that energy per bit is the product of power and transmitting rate, $e_{ij} = P_{ij} r_{ij}$. Thus, the parameter e_{ij} captures *both* changing the power level and the transmission rate, e.g changing the modulation level to reach j.

Each sensor has an initial amount of energy E_i. During a sensor operation, energy can be consumed to perform the following tasks:

- Transmission of information (this is the most energy-consuming task).
- Reception of information (since the reception circuits have to be on and process the received information).
- CPU operation (battery is consumed to perform numerical operations and tasks. Thus, the algorithms for sensor networks need to be simple and of low computational load.
- Sensing (the sensing module consumes energy)
- ON-time of circuits. Even if not involved in any of the operations above, a sensor consumes energy even by being ON (awake as we say).

In this problem, we will assume that energy is consumed only for transporting data. The network can be represented as a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of edges. An edge exists whenever a node j is within the transmission range of node i (can be reached for a constant power P)

We now define the following quantities that will help us construct the model in our problem. Let $c = 1, \ldots, C$ denote the C different kinds of information transferred in the network.

- $Q_i^{(c)}$: Rate of generation of kind c of traffic at node i in units bits/sec.
- $Q_i = \sum_{c=1}^{C} Q_i^{(c)}$: total rate traffic generation at node i.
- $q_{ij}^{(c)}$: The rate at which information of kind c is transferred from node i to j (in bits/sec).
- $q_{ij} = \sum_{c=1}^{C} q_{ij}^{(c)}$: The total rate of information transfer from sensor i to sensor j (in bits/sec).

- $O_c = \{i \in \mathcal{N} : Q_i^{(c)} > 0\}$. The set of origin nodes of traffic of type c.
- D_c : the set of destinations of traffic of type c

The variables are the q_{ij}'s. The problem of routing is equivalent to finding flows q_{ij} or equivalently the flow vector $\mathbf{q} = \{q_{ij} \forall (i,j) \in \mathcal{A}\}$. Vector $\mathbf{q}$ shows how information flows in each edge (i,j) and the way that information streams are split in each node.

2.1 Network Lifetime

The sensor network lifetime is defined as time between the beginning of network operation and the time when first node "dies", namely its energy vanishes and its battery is drained. Here, we should note that there are several different definitions of network lifetime. For example, we lifetime can be alternatively defined as the time until transfer of information from the sources to the destinations is still feasible no matter how many nodes have zero battery. Or network lifetime, can be the time when the battery of some percentage $k\%$ of sensor nodes becomes zero. However, we will consider the definition of network lifetime we said before. This is a meaningful definition in the following sense: the network operates normally until the first node's battery finishes. Then, this node cannot handle traffic any more and additional re-routing algorithms need to be applied in the network to circumvent that node and find alternative routes. Hence, much more additional energy is needed and the rate at which nodes' batteries will be emptying will be higher from then on. Therefore, the time when the battery of one node vanishes is a benchmark and can be defined as the network lifetime.

Let us express the network lifetime as a function of flow vector $\mathbf{q}$. First, we express the *node lifetime* as a function of the flow vector. In the problem formulation from now on, we will assume that the network carries one type of traffic. The generalization to more than one types of traffic is easy.

For a flow vector $\mathbf{q}$, the node lifetime is:

$$T_i(\mathbf{q}) = \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{1}$$

where the denominator shows the total rate of decrease of energy for node i.

The *Network lifetime* for flow vector $\mathbf{q}$ is defined as:

$$T_N(\mathbf{q}) = \min_{i \in \mathcal{N}} T_i(\mathbf{q}) = \min_{i \in \mathcal{N}} \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{2}$$

The maximum lifetime routing problem is defined as follows: Find the flow vector $\mathbf{q}$ so as to maximize network lifetime:

$$\max_{\mathbf{q}} T_N(\mathbf{q}) = \max_{\mathbf{q}} \min_{i \in \mathcal{N}} \frac{E_i}{\sum\limits_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{3}$$

for $q_{ij} \geq 0, \forall\ i \in \mathcal{N},\ \forall j \in \mathcal{N}_i$.

There is also the following constraint for the problem:

$$Q_i + \sum_{j: i \in \mathcal{N}_j} q_{ij} = \sum_{k \in \mathcal{N}_i} q_{ik},\ \forall\ i \in \mathcal{N} \setminus D^c \tag{4}$$

This equation expresses the flow conservation principle at each node i. The way the problem is formulated now is difficult to solve. With a change of variables and the definition of a new variable, we will show that the formulation above is actually equivalently to a Linear Programming problem. Define the network lifetime as a new variable,

$$T = \min_{i \in \mathcal{N}} \frac{E_i}{\sum\limits_{j \in \mathcal{N}_i} e_{ij} q_{ij}}. \tag{5}$$

Then, clearly,

$$T \leq \frac{E_i}{\sum\limits_{j \in \mathcal{N}_i} e_{ij} q_{ij}}, \forall i. \tag{6}$$

Multiply the flow conservation equation with T to get: multiplying

$$TQ_i + T \sum_{j: i \in\ \mathcal{N}_j} q_{ij} = T \sum_{k \in \mathcal{N}_i} q_{ik} \tag{7}$$

We define new variables $\hat{q}_{ij} = T q_{ij}$ where $\hat{q}_{ij} \geq 0$. Then, the flow conservation equation becomes:

$$TQ_i + \sum_{j: i \in \mathcal{N}_j} \hat{q}_{ij} = T \sum_{k \in \mathcal{N}_i} \hat{q}_{ik}. \tag{8}$$

Also, there appears the inequality constraint:

$$\sum_{j \in \mathcal{N}_i} e_{ij} \hat{q}_{ij} \leq E_i. \tag{9}$$

The objective function is now linear in the variable vector $(\hat{\mathbf{q}}, T)$ and the objective is now stated as

$$\max_{\hat{\mathbf{q}}, T} T \tag{10}$$

Thus, we converted our problem into a linear programming problem, since the objective is linear in the variable vector and the constraints are linear in the variables as well.

In general if we have a problem of the form min-max (ours was of the max-min form)

$$\min_{\mathbf{x}} \max_{i=1,\ldots,m} \mathbf{a}_i^T \mathbf{x} + b_i, \tag{11}$$

the idea is to define an extra variable,

$$t = \max_{i=1,\ldots,m} \mathbf{a}_i^T \mathbf{x} + b_i. \tag{12}$$

and the problem will be:

$$\min_{\mathbf{x},t} t \tag{13}$$

subject to the constraints:

$$t \geq \mathbf{a}_i^T \mathbf{x} + b_i, \forall\ i = 1, \ldots, m \tag{14}$$

When we have several linear functions of $\mathbf{x}$, the min-max problem is converted into a linear programming problem.

3 Carrier assignment in OFDM systems

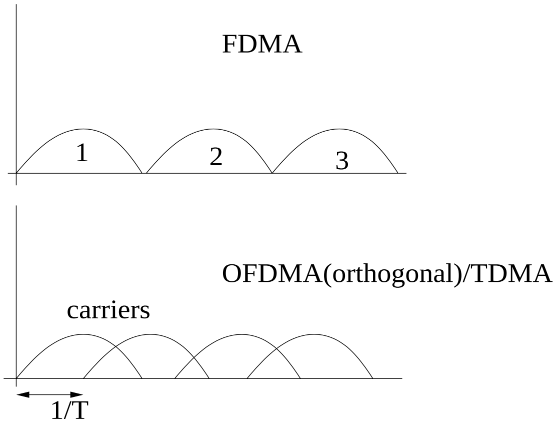


Figure 2: OFDM versus conventional FDMA.

We will now examine and formulate the problem of carrier assignment in OFDM systems as a linear programming problem. OFDM (Orthogonal Frequency Division Multiplexing) is different from conventional FDMA systems in the following sense: in OFDM systems, the spectrum is divided into several sub-carriers with overlapping spectra (see figure 2). Note than in FDMA, the spectrum is divided into non-overlapping spectra.

The innovation in OFDM is that, due to a well-known property of the Fourier transform, although the spectra are overlapping they are *orthogonal* to each other, that is, they do not cause interference to each other. Thus, more efficient use of spectrum is achieved. The OFDM is said to have higher *spectral efficiency* than FDMA. Another innovation is that each user can split its bit stream and use several sub-carriers in parallel (each sub-carrier corresponds e.d. to a different frequency. Note that in FDMA, each user was allocated one frequency.

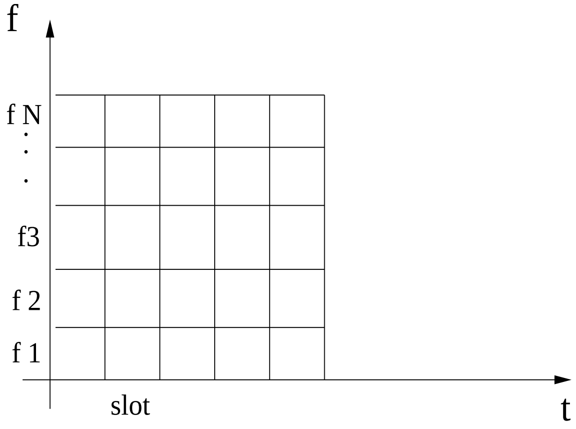


Figure 3: Resources in the system : Timeslots making up one frame and sub-carriers

We will consider here an OFDM/TDMA system. There exist K users and N subcarriers. Time is divided into time slots and C time slots make up one time frame. There are two kinds of resources to be allocated to users: the frequencies (subcarriers $1, \ldots, N$) and the timeslots (slots $1, \ldots, C$ within a frame). See figure 3 for how resources are organized. In the next lecture, we will formulate a sub-carrier assignment problem as an LP problem.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 17 LP example, Introduction to duality in LP: 28/11/06

Notes by : Katerina Mamoura and Eleana Parlavantza

1 Lecture Outline

- Carrier assignment in OFDM systems (continued from lecture 16)
- Introduction to duality in LP

2 Carrier assignment in OFDM systems

This example is about carrier assignment to users in one cell. the BS transmits to K users with N subcarriers in the down-link.

There exist a set of subcarrier frequencies and a set of C time slots that need to be assigned to users in an OFDM system. The timeslots make up a time frame of duration T_f sec. In general, a user can be assigned several time slots in several different subcarriers. The figure below shows the resources (subcarriers/timeslots) that are assigned to two users A and B. Each user i perceives each subcarrier j to be of different quality. There exist two main reasons for that:

- Co-channel interference. Different users are located in different regions within a cell and thus experience different amounts of co-channel interference in each sub-carrier due to different amounts of subcarrier reuse in neighboring cells. For example, a user may be in a location that is close to a cell in which subcarriers 1 and 2 are reused which subcarrier 3 is not. In that case, that user perceives subcarrier 3 as being of very good quality, while subcarriers 1 and 2 are of lower quality. Also, different users perceive the *same* subcarrier as being of different quality for

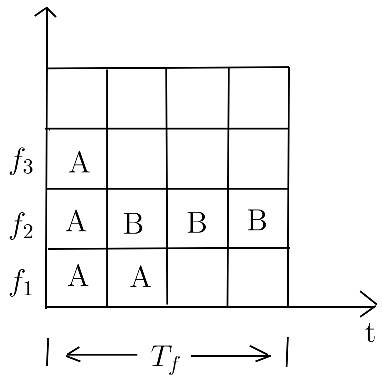


Figure1 : Example
T_f : frame duration

the same reason. For example, if another user is in a location close to a cell where subcarrier 3 is used, then this user perceives subcarrier 3 as of lower quality than the first user does.

- Frequency selectivity. Even in the absence of interference whatsoever, a user has different channel gain in different frequencies. That is, a transmitted signal has different channel attenuation in different frequencies. Frequency selectivity results in different *frequency response function* $H(f)$ in different subcarrier frequencies f. The phenomenon of frequency selectivity is attributed to multi-path: the same signal follows several different paths while traveling from the transmitter to the receiver and each path is of different length (and thus arrives with different delay at the receiver). These delay differences give rise to a frequency dependence on the amplitude and phase of the signal (for more details, look in the class of Wireless Communications notes).

In the problem, we assume that the time frame duration is small enough, so that each user has the same quality across all time slots of a subcarrier. Therefore, we do not consider temporal channel quality variations in our problem.

Given the fact that different users experience different quality in a subcarrier, there comes the question: *Which user is the most suitable to be assigned to a subcarrier?*

As mentioned above, each user i experiences different quality in different subcarriers j. If the user utilizes only one carrier j, the achieved transmission rate r_i is:

$$r_i = \frac{S}{T_f} \cdot b_{ij} \cdot a_{ij}, \qquad (1)$$

where S is the number of symbols of the user transmitted in a time slot, a_{ij} is the number of time slots that are used in a frame in subcarrier j and b_{ij} is the maximum tolerable modulation level (in bits/symbol) that can be assigned to a user in subcarrier j. Rate r_i has units of bits/sec. Clearly, the achievable rate for a user i in a subcarrier j depend on b_{ij}, a_{ij}. The more slots the user uses, the more bits it can transfer in a frame duration. Also, the larger the modulation level, the larger the achievable rate. The quality of each subcarrier j for a user i is reflected on the Signal-to-Interference and Noise ratio (SINR) of a user.

As we have said before, if the SINR is large (the subcarrier is of good quality), the BS can afford to transmit with a large modulation level (and still maintain the BER below a threshold ϵ. On the other hand, if the SINR for a user in a subcarrier is not good, the BS cannot transmit with a high modulation level. Instead, it needs to use lower modulation level, so as to maintain BER below ϵ.

Each user sends a message to the BS and informs it about the quality in each subcarrier. The user receiver can easily measure the quality in different subcarriers at its receives and then it can feed back this information at the BS. The user i thus essentially indirectly informs the BS of the modulation level vector $(b_{i1}, ..., b_{iN})$ that the BS can give in the different subcarriers. Now, if the user has also declared its requirements in rate to the BS, the BS gets informed about the user's preferences and can estimate how many slots are needed for every user in order to fulfil its rate requirements (if the user uses exclusively one subcarrier). Thus, the number of slots that are needed by a user i in order to fulfil its rata requirements if it used only one subcarrier j is:

$$a_{ij} = \left\lceil \frac{r_i \cdot T_f}{S \cdot b_{ij}} \right\rceil \tag{2}$$

Thus we make the following observations:

- The more the user rate requirements, the more the slots that the user needs in order to fulfil them.
- The better the channel quality in a subcarrier for a user, the higher the achievable modulation level and thus the fewer the number of slots that are required in order for the user to fulfil its rate requirements.

The BS faces the following problem: Given a number of users with some rate requirements and given a number of carriers, allocate carriers and timeslots to users, such that the user rate requirements are satisfied and the minimum total number of time slots are used. This optimization objective is meaningful, since the BS would like to have as many free slots are possible in case they are needed:

- in order to serve a burst of many new arriving users. Hence the free slots help to serve sudden increased in user loads.
- in order to cope with sudden subcarrier quality deterioration for many users. This often occurs in wireless systems. If the quality of one or more subcarriers deteriorates for users, then the users need additional time slots in order to fulfil rate requirements.

An example with three users A,B and C and three subcarrier frequencies f_1, f_2, f_3 is given in the figure below (denoted as Figure 4). The matrix element (i, j) shows the number of required time slots by user i in subcarrier j in order to fulfil its rate requirements by using *exclusively* this subcarrier. Thus for example user A prefers to use subcarrier f_1 since it will occupy fewer slots there. User C also prefers subcarrier f_1 to the other two for the same reason.

	f_1	f_2	f_3
A	2	5	3
B	4	1	2
C	3	4	5

Figure 4 : User requirements in frequencies

However, if a subcarrier is preferable by several users, it may happen that the number of slots in that subcarrier is not adequate to accommodate all users. For instance, if subcarrier f_1 has $C = 4$ time slots in a frame, and user A is assigned to subcarrier f_1, user B to subcarrier f_2 and user C to subcarrier f_1, then 5 slots are needed to satisfy users A and C in subcarrier f_1 (but only 4 are available!). This is shown in the figure below (denoted as figure 5)

So,only 2 slots out of the 3 needed can be given to user C at subcarrier f_1. Thus the rest of its requirements need to be fulfilled by assigning to the user slots by lesser quality subcarrier (e.g the next more preferable subcarrier for user C is f_2). However note that if user C is given slots from

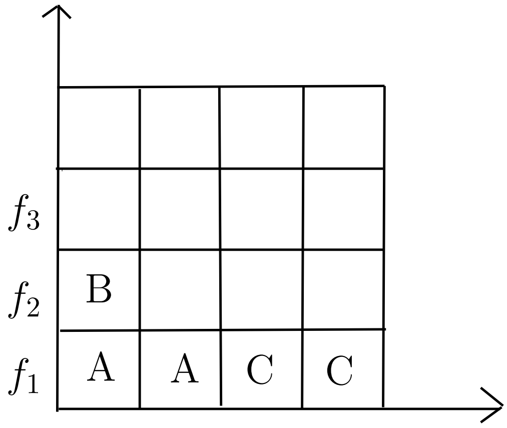


Figure 5

subcarrier f_2 the 1 remaining needed slot (if it was assigned to subcarrier f_1) is equivalent to more than one (actually $\lceil 4/3 \rceil = 2$ slots, if assigned to subcarrier f_2).

3 Linear Programming formulation

The variables of the problem are x_{ij} : portion (percentage) of rate requirements of user i that are satisfied by carrier j, $i = 1, \ldots, K$ and $j = 1, \ldots, N$. The variable vector is $\mathbf{x} = (x_{ij} : i = 1, \ldots, K, j = 1, \ldots, N)$.

A problem instance is described by the long $NK \times 1$ vector $\mathbf{a} = (a_{ij} : i = 1, \ldots, K, j = 1, \ldots, N)$. As mentioned before, the parameters a_{ij} are known to the BS for each user i and each subcarrier j and denote the number of time slots that are needed by user i to entirely fulfil its rate requirements when assigned *only* to carrier j. Let $a_{ij} \in \mathcal{R}$. Let all the subcarriers have capacity of C time slots per frame.

We want to minimize the total number of time slots that are needed to satisfy all users:

$$\min_{\mathbf{x}} \sum_{i=1}^{K} \sum_{j=1}^{N} a_{ij} x_{ij} \tag{3}$$

subject to the following constraints:

$$\sum_{i=1}^{K} a_{ij} x_{ij} \leq C, \text{ for } j = 1, \ldots, N. \tag{4}$$

and

$$\sum_{j=1}^{N} x_{ij} = 1, \text{ for } i = 1, \ldots, K. \tag{5}$$

The first constraint specifies that the available slot capacity must not be exceeded. The second constraint says that user rate requirements need to be satisfied. Also, for the variables x_{ij} it is $0 \leq x_{ij} \leq 1$. Since the objective function and the constraints are linear in the variable vector $\mathbf{x}$, the formulation above is an LP problem.

4 Introduction to Duality

We now turn our attention to a very important topic of Linear Programming, that of duality. Duality appears in LP as well as in non-LP problems.

Consider the non-linear problem:

$$\min x^2 + y^2 \tag{6}$$

subject to:

$$x + y = 1 \tag{7}$$

Eliminate the constraint and define the Lagrangian function:

$$L(x, y, \lambda) = x^2 + y^2 + \lambda(1 - x - y). \tag{8}$$

Thus, instead of enforcing the constraint $x+y = 1$, we allow it to be violated and associate a so-called Lagrange multiplier λ (price per unit of violation) with the amount $1-x-y$ by which the constraint is violated. Then, we have an unconstrained minimization problem that is solved by taking:

$$\frac{\partial L}{\partial x} = 0 \text{ and } \frac{\partial L}{\partial x} = 0 \Rightarrow x = y = \lambda/2 \tag{9}$$

From the constraint that needs to be satisfied, we obtain $\lambda = 1$ and thus $x = y = 1/2$.

When the price is appropriately chosen (as in here, $\lambda = 1$), the optimal solution to the unconstrained problem is also optimal for the constrained one. Either we take into consideration the constraint or not, the effect on the solution is the same.

In LP problem, we associate a price variable with each constraint. We then search for the prices under which the presence or absence of constraints does not affect the optimal cost. The optimal prices are found by solving a new LP problem, the *dual* problem.

4.1 Linear Programming

Consider the original LP optimization problem as we have seen it till now:

$$\min \mathbf{c}^T\mathbf{x} \tag{10}$$

subject to:

$$A\mathbf{x} = \mathbf{b},\ \text{and}\ \mathbf{x} \geq 0. \tag{11}$$

This is called *Primal* problem (P).

Suppose there exists an optimal solution $\mathbf{x}^*$. Matrix A has dimension $m \times n$ (that means that there are m constraints). Relax the problem and define the Lagrangian function

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{12}$$

and we have the unconstrained problem:

$$\min \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{13}$$

subject to $\mathbf{x} \geq \mathbf{0}$. The term $\boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x})$ is the penalty associated with violating the constraints. Define $g(\boldsymbol{\lambda})$ the optimal cost for the relaxed problem as a function of $\boldsymbol{\lambda}$,

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x} \geq \mathbf{0}} \ \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T \cdot (\mathbf{b} - A\mathbf{x}) \tag{14}$$

We have

$$g(\boldsymbol{\lambda}) \leq \min_{\mathbf{x} \geq \mathbf{0}, A\mathbf{x} = \mathbf{b}} \ \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{15}$$

since $\min_{\mathbf{x} \in \mathcal{A}} f(x) \leq \min_{\mathbf{x} \in \mathcal{B}} f(x)$ for $\mathcal{B} \subseteq \mathcal{A}$. Thus, we further get:

$$g(\boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x}^* + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}^*) = \mathbf{c}^T\mathbf{x}^* \tag{16}$$

since $\mathbf{x}^*$ is feasible solution for the primal problem.

For each $\boldsymbol{\lambda}$, $g(\boldsymbol{\lambda})$ is a lower bound on the optimal cost of the primal problem, $\mathbf{c}^T\mathbf{x}^*$. The question now in to find the best (highest) lower bound. This is the dual problem.

4.2 Dual Problem

The dual problem is a maximization problem. The primal problem without the constraints is:

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x} \geq \mathbf{0}, A\mathbf{x} = \mathbf{b}} \ \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{17}$$

and then

$$g(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \mathbf{b} + \min_{\mathbf{x} \geq \mathbf{0}} (\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} \tag{18}$$

Now,

$$\min_{\mathbf{x} \geq \mathbf{0}} (\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} = 0, \text{ if } \mathbf{c}^T - \boldsymbol{\lambda}^T A \geq \mathbf{0}^T \tag{19}$$

else it is $-\infty$. Hence, in maximizing $g(\boldsymbol{\lambda})$, we must only consider those values of $\boldsymbol{\lambda}$ for which $g(\boldsymbol{\lambda})$ is *not* $-\infty$.

The dual problem is therefore:

$$\max \boldsymbol{\lambda}^T \mathbf{b} \tag{20}$$

subject to the constraints:

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T \tag{21}$$

Notice that the dual has no constraints on the sign of $\boldsymbol{\lambda}$. The primal problem is a minimization problem, whereas the dual problem is a maximization problem. In the dual problem $\mathbf{c^T}$ (the cost vector for the primal) has become right-hand side of the constraints.

If we have an LP problem in inequality form (constraints are $A\mathbf{x} \geq \mathbf{b}$), we convert it to the standard form by using a slack variable $\mathbf{s} \geq \mathbf{0}$:

$$A\mathbf{x} - \mathbf{s} = \mathbf{b} \tag{22}$$

or

$$[A \;\; -I](\mathbf{x} \; \mathbf{s})^T = \mathbf{b} \tag{23}$$

Then, according to the previously found dual, we will have the dual constraints:

$$\boldsymbol{\lambda}^T [A \;\; -I] \leq [\mathbf{c}^T \; \mathbf{0}^T] \tag{24}$$

or

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T \tag{25}$$

and $\boldsymbol{\lambda} \geq \mathbf{0}$.

So, if in the primal problem the constraints are inequalities, in the dual, we have the constraint that the dual variables $\boldsymbol{\lambda} \geq \mathbf{0}$.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 18 : Duality in Linear Programming - 04/12/06

Notes by : Nikolaos Kapetanios and Konstantinos Sotiropoulos

1 Lecture outline

- Duality in LP
- Weak and strong duality lemmas
- Complementary slackness conditions and interpretation

2 Primal LP problems and their dual problems

2.1 Forms of the primal problem

Below we show some primal problems of linear programming and their corresponding dual.

PRIMAL PROBLEM		DUAL PROBLEM	
$\min \mathbf{c}^T\mathbf{x}, \quad \text{s.t. } A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$	$\longrightarrow$	$\max \boldsymbol{\lambda}^T\mathbf{b}, \quad \text{s.t. } \boldsymbol{\lambda}^T A \leq \mathbf{c}^T, \ \boldsymbol{\lambda}$ unrestricted.	(1)
$\min \mathbf{c}^T\mathbf{x}, \quad \text{s.t. } A\mathbf{x} = \mathbf{b},$	$\longrightarrow$	$\max \boldsymbol{\lambda}^T\mathbf{b}, \quad \text{s.t. } \boldsymbol{\lambda}^T A = \mathbf{c}^T, \ \boldsymbol{\lambda}$ unrestricted.	(2)
$\min \mathbf{c}^T\mathbf{x}, \quad \text{s.t. } A\mathbf{x} \geq \mathbf{b},$	$\longrightarrow$	$\max \boldsymbol{\lambda}^T\mathbf{b}, \quad \text{s.t. } \boldsymbol{\lambda}^T A = \mathbf{c}^T, \ \boldsymbol{\lambda} \geq \mathbf{0}.$	(3)
$\min \mathbf{c}^T\mathbf{x}, \text{ s.t. } A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$	$\longrightarrow$	$\max \boldsymbol{\lambda}^T\mathbf{b}, \quad \text{s.t. } \boldsymbol{\lambda}^T A \leq \mathbf{c}^T, \ \boldsymbol{\lambda} \geq \mathbf{0}.$	(4)

Generally, when the primal problem has inequality constraints, then in the dual we have the variables $\boldsymbol{\lambda} \geq \mathbf{0}$. When the primal problem has equality constraints, then in the dual the variables $\boldsymbol{\lambda}$ are unrestricted in sign.

Fact: The dual of the dual is the primal problem.

Proof: Assume the primal problem and its corresponding dual of equation (4) before. The dual problem can be written as

$$\min \boldsymbol{\lambda}^T(-\mathbf{b}) \text{ s.t. } \boldsymbol{\lambda}^T(-A) \geq -\mathbf{c}^T, \boldsymbol{\lambda} \geq \mathbf{0}. \tag{5}$$

The dual problem of the above is:

$$\max\,(-\mathbf{c})^T\mathbf{x},\ \text{s.t. } (-A)\mathbf{x} \leq (-\mathbf{b}),\ \mathbf{x} \geq \mathbf{0}, \tag{6}$$

which can be written as

$$\min \mathbf{c}^T\mathbf{x},\ \text{s.t. } A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}, \tag{7}$$

which is the primal problem of equation (4). Thus, we proved that the dual problem of a dual problem is its primal problem.

2.2 Example (The Diet Problem)

A diet contains m different vitamins that need to be received daily with quantities at least equal to $b_1, ..., b_m$ respectively. The diet also have n different foods. Let a_{ij} denote the amount of vitamin i per unit of j-th food.

A company intends to propose a diet that is most economical. We can compose such a diet by choosing nonnegative food quantities $\mathbf{x} = (x_1, ...x_n)$. One unit quantity of food j and has a cost of c_j . We want to determine the cheapest diet that satisfies the nutritional requirements. This problem can be formulated as the LP primal problem,

$$\min \mathbf{c}^T\mathbf{x},\ \text{s.t.}\, A\mathbf{x} \geq \mathbf{0},\ \mathbf{x} \geq \mathbf{0}. \tag{8}$$

The corresponding dual problem can be defined as

$$\max \boldsymbol{\lambda}^T\mathbf{b},\ \text{s.t.}\, \boldsymbol{\lambda}^T A < \mathbf{c}^T,\ \boldsymbol{\lambda} \geq \mathbf{0}, \tag{9}$$

where λ_i (dual problem variable) is the price of the unit quantity of vitamin $i = 1, ..., m$. In other words, this is the problem that another company (competitive to the first one) needs to solve. The company proposes a diet in which it has synthetically reproduced each food with vitamins. It then needs to find the pricing mechanism for each vitamin, i.e find the price vector $\boldsymbol{\lambda}$ to maximize its total benefit. At the same time, it has the constraints that the cost of the equivalent for food j should be,

$$\lambda_1 a_{1j} + \lambda_2 a_{2j} + ... + \lambda_m a_{mj} \leq c_j. \tag{10}$$

The inequality above should hold in order for the second company to be competitive. One unit quantity of food j has a production cost c_j and a price $\boldsymbol{\lambda}^T\mathbf{A}_j$ where $\mathbf{A}_j$ is the j-th column of matrix A, with elements a_{ij}, $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

2.3 Theorems and Lemmas in Duality

2.3.1 Duality Theorem

If the primal problem has an optimal solution $\mathbf{x}^*$, then so does the dual (it has an optimal solution $\boldsymbol{\lambda}^*$, and the optimal values of their respective objective functions are equal. In other words,

$$\mathbf{c}^T\mathbf{x}^* = \boldsymbol{\lambda}^{*T}\mathbf{b}. \tag{11}$$

Before coming to that, we will show that $\mathbf{c}^T\mathbf{x}^* \geq \lambda^T\mathbf{b}$.

2.3.2 Weak Duality Lemma

Suppose that $\mathbf{x_0}$ and λ_0 are feasible solutions to primal and dual problems respectively. Then,

$$\mathbf{c}^T\mathbf{x}_0 \geq \boldsymbol{\lambda}_0^T\mathbf{b}. \tag{12}$$

This inequality is known as the *Weak Duality Lemma.* Now, set $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_0$ and $\mathbf{x}^* = \mathbf{x}_0$ and we get

$$\mathbf{c}^T\mathbf{x}^* \geq \boldsymbol{\lambda}^{*T}\mathbf{b}. \tag{13}$$

Every feasible solution of the dual problem gives a lower bound on the value of the objective function of the primal problem. Also, every feasible solution to the primal problem gives an upper bound on the value of the objective function of the dual.

Note that as, it can be verified easily, the weak duality lemma holds for any of the four primal-dual pairs that we mentioned in the beginning. Two immediate conclusions are:

- If the primal problem is unbounded so that $\mathbf{c}^T\mathbf{x}^* = -\infty$, then the dual problem is infeasible.
- If the dual problem is unbounded so that $\boldsymbol{\lambda}^*\mathbf{b} = +\infty$, then the primal problem is infeasible.

Note: By saying a problem is infeasible, it means that its set of feasible solutions is the empty set.

2.3.3 Theorem

Suppose that $\mathbf{x}$ and λ are feasible solutions to the primal and dual problem respectively. If $\mathbf{c}^T\mathbf{x} = \boldsymbol{\lambda}^T\mathbf{b}$, then $\mathbf{x}$ and $\boldsymbol{\lambda}$ are optimal solutions to the primal and dual problems respectively.

2.3.4 Lemma

Suppose that $\mathbf{x}$ and $\boldsymbol{\lambda}$ are feasible solutions to the primal and dual problem respectively. Then,

$$\mathbf{c}^T\mathbf{x} \geq \boldsymbol{\lambda}^T A\mathbf{x} \geq \boldsymbol{\lambda}^T\mathbf{b}. \tag{14}$$

Proof: For the first inequality, we must show that

$$(\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} \geq 0, \tag{15}$$

which is true because $\mathbf{c}^T - \boldsymbol{\lambda}^T A \geq \mathbf{0}$ and $\mathbf{x} \geq \mathbf{0}$. For the second inequality, we must show that

$$\boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \geq 0, \tag{16}$$

which is also true because $\boldsymbol{\lambda} \geq \mathbf{0}$ as a feasible solution to the dual problem and $A\mathbf{x} - \mathbf{b} \geq \mathbf{0}$ as $\mathbf{x}$ is a feasible solution to the primal problem.

If the constraint $A\mathbf{x} \geq \mathbf{b}$ was replaced by $A\mathbf{x} = \mathbf{b}$ then the respective dual problem would be

$$\max \boldsymbol{\lambda}^T\mathbf{b}, \qquad \text{s.t.}\, \boldsymbol{\lambda}^T A \leq \mathbf{c}^T,\, \boldsymbol{\lambda} \text{ unrestricted}, \tag{17}$$

which shows us that $\boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \geq 0$, is always true.

2.3.5 Strong Duality Theorem

Suppose that $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ are feasible solutions to the primal and dual problem respectively and $\mathbf{c}^T\mathbf{x}^* \geq \boldsymbol{\lambda}^{*T}\mathbf{b}$. There are four options about the solutions of primal and dual problems:

- Both problems have optimal solutions (of finite value).
- Both problems are infeasible (their sets of feasible solutions are empty).
- The primal problem is unbounded and the dual problem is infeasible.
- The primal problem is infeasible and the dual problem is unbounded.

2.3.6 Theorem: Complementary Slackness Conditions

The feasible solutions $\mathbf{x}^*$ and λ^* to the primal and dual problem respectively are optimal solutions if and only if

1. $(\mathbf{c}^T - \boldsymbol{\lambda}^{*T}\mathbf{b})\mathbf{x}^* = 0$
2. $\boldsymbol{\lambda}^{*T}(A\mathbf{x}^* - \mathbf{b}) = 0.$

We omit the proofs and focus on their interpretation.

1. We know that $\mathbf{x}^* \geq \mathbf{0}$ and $\mathbf{c}^T - \boldsymbol{\lambda}^{*T}A \geq \mathbf{0}^T$. This means that,

$$(c_j - \boldsymbol{\lambda}^{*T}\mathbf{A}_j)x_j^* = 0 \;\forall\; j = 1, ..., n. \tag{18}$$

The conclusions are:

$$\text{If } x_j^* > 0 \Rightarrow \boldsymbol{\lambda}^{*T}\mathbf{A}_j = c_j, \tag{19}$$

$$\text{if } \boldsymbol{\lambda}^{*T}\mathbf{A}_j < c_j \Rightarrow x_j^* = 0, \tag{20}$$

where c_j is the j-th element of vector $\mathbf{c}$, x_j^* is the j-th element of vector $\mathbf{x}^*$ and $\mathbf{A}_j$ is the j-th column of matrix A.

Thus, if a component of the primal solution is strictly positive, the corresponding constraint in the dual must be met with equality at the optimal solution. And also, if an inequality constraint at the dual is not met with "clean" inequality at the optimal solution, the corresponding variable at the primal optimal solution is zero.

2. We know that $\boldsymbol{\lambda}^* \geq 0$ and $A\mathbf{x}^* \geq \mathbf{b}$. This means that

$$\lambda_i^{*T}(\mathbf{a}_i^T\mathbf{x}^* - b_i) = 0 \ \forall \ i = 1, ..., m. \tag{21}$$

The conclusions are:

$$\text{If } \lambda_i^* > 0 \Rightarrow \mathbf{a}_i^T\mathbf{x}^* = b_i, \tag{22}$$

$$\text{If } \mathbf{a}_i^T\mathbf{x}^* > b_i \Rightarrow \lambda_i^* = 0. \tag{23}$$

where λ_i is the i-th element of vector λ, $\mathbf{a}_i$ is the i-th row of matrix A and b_i is the i-th element of vector $\mathbf{b}$. Similar statements can be made here as those for case 1.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 19 : Dual variables and Shortest Path problem - 5/12/06

Notes by : Aggelos-Christos Anadiotis, Giannis Dimitropoulos, Odysseas Kalamiotis

1 Lecture outline

- Interpretation of dual variables
- Sensitivity analysis
- Shortest path problem
- Minimum cost flow problem

2 Interpretation of dual variables

Consider the primal LP problem

$$\min \mathbf{c}^T\mathbf{x} \tag{1}$$

subject to:

$$A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \tag{2}$$

Recall as r_i the reduced cost coefficients from the simplex algorithm. Let the optimal BFS be $\mathbf{x} = (\mathbf{x}_B\ \mathbf{0})$. At the end of the simplex algorithm, matrix A is partitioned as $[B\ D]$, where B is the matrix of linearly independent columns, corresponding to the basis. As we know, $\mathbf{x}_B = B^{-1}\mathbf{b}$. Recall also that at the end of the simplex algorithm the vector of relative coefficients corresponding to the non-basic variables is $\mathbf{r} \geq \mathbf{0}$. It can be easily shown that at the end of the simplex algorithm, it is

$$\mathbf{r}^T = \mathbf{c}_B^T - \mathbf{c}_B^T B^{-1} D, \tag{3}$$

where for the cost vector is separated in two parts corresponding to the basic and non-basic variables, as

$$\mathbf{c} = (\mathbf{c}_B \ \mathbf{c}_D). \tag{4}$$

Since at the optimal solution it is $\mathbf{r}^T \geq \mathbf{0}^T$, we have the inequality

$$\mathbf{c}_D^T \mathbf{B}^{-1} D \geq 0 \Rightarrow \mathbf{c}_B^T B^{-1} D \leq \mathbf{c}_D^T. \tag{5}$$

Now, consider the dual problem,

$$\max \boldsymbol{\lambda}^T \mathbf{b} \tag{6}$$

subject to:

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T, \tag{7}$$

and $\boldsymbol{\lambda}$ unrestricted in sign. Define $\boldsymbol{\lambda} = \mathbf{c}_B^T B^{-1}$. We are going to prove that $\boldsymbol{\lambda}^T = \mathbf{c}_B^T B^{-1}$ is the optimal solution to the dual problem.

First, we will check if $\boldsymbol{\lambda}$ is a feasible solution to the dual problem, i.e. if it satisfies $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$. Indeed, it is $\boldsymbol{\lambda}^T A = \boldsymbol{\lambda}^T [B \ D] = [\mathbf{c}_B^T \ \mathbf{c}_B^T B^{-1} D] \leq [\mathbf{c}_B^T \ \mathbf{c}_D^T] = \mathbf{c}^T$. Thus, $\boldsymbol{\lambda}$ is a feasible solution of the dual.

Next, we prove that $\boldsymbol{\lambda}$ is the optimal solution for the dual problem. We have $\boldsymbol{\lambda}^T \mathbf{b} = \mathbf{c}_B^T B^{-1} \mathbf{b} = \mathbf{c}_B^T \mathbf{x}_B = \mathbf{c}^T \mathbf{x}$, where $\mathbf{x_B}$ is the vector of the basic variables. Therefore, we proved that if the primal LP has an optimal BFS with basis B, then $\lambda^T = \mathbf{c}_B^T B^{-1}$ is optimal solution for the dual problem. Vector $\boldsymbol{\lambda}^T$ is called vector of *simplex multipliers*.

3 Sensitivity Analysis

Consider the primal problem:

$$\begin{aligned} &\min \mathbf{c}^T \mathbf{x} \\ \text{subject to: } &A\mathbf{x} = \mathbf{b} \\ &\mathbf{x} \geq \mathbf{0} \end{aligned}$$

Let the optimal basic feasible solution (BFS) be $\mathbf{x} = (\mathbf{x_B}, 0)$, with basis matrix B and $\mathbf{x}_B = B^{-1}\mathbf{b}$.

Let's assume that the right-hand side of the constraints changes by a small $\Delta\mathbf{b}$, that is $\mathbf{b} \rightsquigarrow \mathbf{b} + \Delta\mathbf{b}$. For small changes $\Delta\mathbf{b}$, the basis matrix B does not change. After the change, the new optimal solution will be $\mathbf{x}' = (\mathbf{x}_B + \Delta\mathbf{x}_B, 0)$, with $\Delta\mathbf{x}_B = B^{-1}\Delta\mathbf{b}$.

The value of the objective function before the change was $\mathbf{c}_B^T\mathbf{x}_B$. After the change, it is $\mathbf{c}_B^T(\mathbf{x}_B + \Delta\mathbf{x}_B)$. The objective function has been changed by quantity $\Delta z = \mathbf{c}_B^T\Delta\mathbf{x}_B = \mathbf{c}_B^T B^{-1}\Delta\mathbf{b} = \boldsymbol{\lambda}^T\Delta\mathbf{b}$.

Suppose there was only one constraint, $m = 1$, then: $\Delta z = \lambda\Delta b \Rightarrow \lambda = \frac{\Delta z}{\Delta b}$. So, λ can be interpreted as the rate of change of the value of the objective function for small changes of the constraints. Since the constraints often represent resources, λ can be interpreted as the *price* per unit of the resources. This is also obvious from the fact that $\Delta z = \lambda\Delta b$ if Δz is the profit delivered by quantity b of the resources. Therefore, λ specifies the change in the value of the objective function with respect to a unit change in resources. It is also called *shadow price* or *marginal cost.*

For more than one constraint, $m > 1$, it is $\lambda_i = \dfrac{\partial z}{\partial b_i}$ and λ_i is the rate of change in the value of the objective function with respect to a change in constraint i.

Next, we will study some examples of LP problems and will try to interpret duality theorems for them.

4 Shortest Path Problem

First, we consider the shortest path (SP) problem. This problem is a special case of the more general *minimum cost flow problem.*

Consider a directed graph $G = (V, E)$, where V represents the set of nodes and E represents the set of edges of the graph. Let $c_j \geq 0$ be the cost associated with each edge $e_j \in E$. The min-cost path problem is the problem of finding a directed path of minimum total cost from a source node s to a destination node t. For the special case where $c_j = 1$ for each $e_j \in E$, we have the shortest path problem, i.e the problem of finding the shortest route to th destination.

The problem arises in several applications such as routing, power control etc. The cost of an edge (i, j) may denote the required power to reach from transmitter i to receiver j. Then, the shortest path specifies the route with the minimum total power consumption. energy costs can also be similarly incorporated in that context. Costs may also denote delays in packet forwarding in a network (which may model link rates or queueing delays at nodes).

In the SP problem, the feasible set is $\mathcal{F} =$ sequences $\{P = (e_{j_1}, \ldots, e_{j_k})\}$ such that the sequence is a directed path from s to t, i.e all possible paths leading from source to destination.

Let the path cost be $c(P) = \sum_{i=1}^{k} c_{j_i}$. Define Let the *node-edge incidence matrix* $A = [A_{ij}], i =$

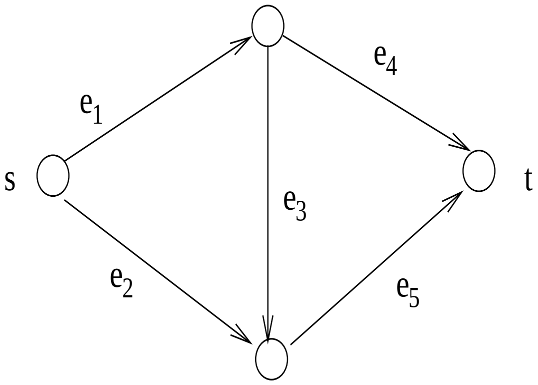


Figure 1: An example network.

$1, ..., |V|, j = 1, ..., |E|$, with

$$A_{ij} = \begin{cases} +1 & \text{if edge } e_j \text{ leaves node } i \\ -1 & \text{if edge } e_j \text{ enters node } i \\ 0 & \text{otherwise.} \end{cases}$$

Example: For the graph of Figure 4, it is

$$A = \begin{pmatrix} +1 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ -1 & 0 & +1 & +1 & 0 \\ 0 & -1 & -1 & 0 & +1 \end{pmatrix}$$

The rows, starting from the first one, stand for nodes s, t, a, b respectively. The columns, starting from the first one, stand for edges e_1, e_2, e_3, e_4, e_5 respectively.

Associate a flow variable f_j with each edge e_j to represent flow of an imaginary fluid through e_j. Consider the flow vector $\mathbf{f} = (f_j : j = 1, ..., |E|)$. The flow conservation principle at each node i can be expressed as the equation

$$\mathbf{a}_i^T \mathbf{f} = 0,\, i \neq \{s, t\}\,, \tag{8}$$

where $\mathbf{a}_i$ is the i-th row of matrix A. A path from s to t is a flow of one unit leaving s and entering t. this flow satisfies the flow conservation equations above at each intermediate node in the path, and also $\mathbf{a}_s^T\mathbf{f} = +1$ and $\mathbf{a}_t^T\mathbf{f} = -1$. Overall, the constraints are written as

$$A\mathbf{f} = \begin{bmatrix} +1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The first two rows stand for s and t respectively and the next rows stand for nodes $i \neq s, t$. The primal problem can be stated as:

$$\min \mathbf{c}^T\mathbf{f}$$
$$\text{subject to: } \mathbf{f} \geq \mathbf{0}$$

$$A\mathbf{f} = \begin{bmatrix} +1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

In its most general form, the problem is the minimum cost flow problem that has solutions $\mathbf{f} \geq \mathbf{0}$ and $\mathbf{f} \leq \mathbf{1}$. The shortest path problem is a special case of the minimum cost flow problem, where $f_e \in \{0, 1\}$. At the optimal solution,

- if $f_{e_j} = 1$, then edge e_j is part of the optimal path P^* to the destination.
- if $f_{e_j} = 0$, then edge e_j is not path of the optimal path.

In the dual problem there is one variable for each node in the network. The dual problem is:

$$\max\left(\lambda_s - \lambda_t\right)$$
$$\text{subject to: } \lambda_i - \lambda_j \leq c_{ij}, \text{ for each edge } e = (i, j)$$
$$\lambda_i \text{ unrestricted in sign.}$$

The constraints can be seeing as emerging from the dual constraints $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$. Let λ_i be the cost of having one flow unit at node i. The complementary slackness conditions for this problem in the optimal solution are written as

$$\left(\lambda_i - \lambda_j - c_{ij}\right) f_{ij} = 0 \tag{9}$$

If $\lambda_i - \lambda_j < c_{ij} \Rightarrow f_{ij} = 0$. This means that if the cost of transporting one unit of flow from node i to node j is more than the difference in costs of having the flows at i and j, then edge (i, j) is not included in the shortest path (because there can be another way of transporting one unit of flow from i to j. On the other hand, if $f_{ij} > 0 \Rightarrow \lambda_i - \lambda_j = c_{ij}$, and this means that edge (i, j) is included in the shortest path.

For the shortest path problem from a single source to a single destination and non-negative edge costs, there is the Dijkstra algorithm. A more general algorithm for multiple sources and destinations and also negative costs is the Bellman-Ford algorithm.

5 Assignment Problem

Consider the following problem. There exist n tasks / jobs to be assigned to n persons. The benefit of assigning task j to person i is a_{ij}. Alternatively, a_{ij} denotes the cost of assigning task j to person i. Depending on one or the other case, we have the min-cost or max-weight assignment problem. We will consider the second case.

Define the variables

$$x_{ij} = \begin{cases} 0 & \text{if task } j \text{ is not assigned to person } i \\ 1 & \text{if task } j \text{ is assigned to person } i \end{cases} \tag{10}$$

The maximum weight assignment problem (P) is the following:

$$\max_{\{x_{ij}\}} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij} \tag{11}$$

subject to:

$$\sum_{i=1}^{n} x_{ij} = 1, \ \forall \text{ task } j \tag{12}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \ \forall \text{ person } i, \tag{13}$$

and $x_{ij} \in \{0, 1\}$.

Define a dual variable λ_j for each constraint corresponding to a task j and a dual variable μ_i for each constraint corresponding to a person i. These can denote the cost of having the task j assigned and the cost of having occupied person i. The dual problem is:

$$\min \sum_{j=1}^{n} \lambda_j + \sum_{i=1}^{n} \mu_i \tag{14}$$

subject to:

$$\lambda_j + \mu_i \geq a_{ij}, \forall (i,j) \tag{15}$$

and $\{\lambda_j\}, \{\mu_i\}$ unrestricted in sign.

Based on complementary slackness, we have at the optimal solution: $(a_{ij} - \lambda_j - \mu_i) x_{ij} = 0$. Thus, if $\lambda_j + \mu_i > a_{ij} \Rightarrow x_{ij} = 0$. This means that if the benefit a_{ij} of assigning task j to person i is less than the incurred cost, then do not assign task j to person i. On the other hand, if $x_{ij} > 0 \Rightarrow \lambda_j + \mu_i = a_{ij}$, i.e it is valid and meaningful to assign task j to person i if the incurred benefit equals the incurred cost.

6 Minimum Cost Flow Problem

Consider a network represented by a directed graph $G(V,E)$. Let c_{ij} be the cost of transferring one unit of flow through the edge (i,j), and u_{ij} be the capacity of edge (i,j), i.e the maximum flow that can be transported through the edge (i,j). Also define

$$b_i \begin{cases} > 0 & \text{if } i \text{ is the source} \\ < 0 & \text{if } i \text{ is the destination} \\ = 0 & \text{otherwise} \end{cases}$$

For each node in the network, we have

$$b_i + \sum_{j:(j,i)\in E} f_{ji} = \sum_{j:(i,j)\in E} f_{ij}, \tag{16}$$

namely the flow conservation equation. Also, $0 \leq f_{ij} \leq u_{ij}$. The minimum cost flow problem is:

$$\min \sum_{(i,j)\in E} c_{ij} f_{ij}, \tag{17}$$

subject to the constraints above. The problem is called uncapacitated min-cost flow problem, if $u_{ij} = +\infty$ for all edges (i,j), otherwise it is called capacitated.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 20 : NLP problems with equality constraints - 6/12/06

Notes by : Aggelos-Christos Anadiotis, Giannis Dimitropoulos, Odysseas Kalamiotis

1 Lecture outline

- Other special cases of the min-cost flow problem
- Introduction to non-linear programming problems with equality constraints
- Lagrange theorem

2 Maximum Flow Problem

Another special case of the min-cost flow problem is the maximum flow problem. This is to find the largest possible amount of flow that can be sent from a given source s to a given destination t without exceeding the edge capacities.

The problem is formulated as follows:

$$\max b_s \tag{1}$$

subject to the constraints:

$$A\mathbf{f} = \mathbf{b}, \quad 0 \leq f_{ij} \leq uij$$
$$b_t = -b_s,\ b_i = 0, i \neq s, t$$

The maximum flow problem can be turned into a min-cost flow problem is we do the following. The costs of all edges are set to 0. We introduce an edge (t, s) of infinite capacity with cost $c_{ts} = -1$. Then, the min-cost flow objective $\min \sum_{(i,j)} c_{ij} f_{ij}$ becomes equivalent to max-flow objective $\max f_{ts}$

(where flow f_{ts} needs to be returned from s to t through the original network). For the max-flow problem, there is Ford-Fulkerson algorithm.

3 Transportation Problem

Another special case of the min-cost flow problem is the transportation problem. We have m suppliers and n consumers. The i-th supplier supplies s_i units of a good and the j-th consumer demands d_j units of good, where $i = 1, ..., m$ and $j = 1, ..., n$. The total amount of supply equals the total demand

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j \tag{2}$$

There is also a cost c_{ij} per unit of good carried from the i-th supplier to the j-th consumer. The problem is to transport the good from the suppliers to the consumers at minimum cost and can be formulated as follows:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} f_{ij} \tag{3}$$

subject to:

$$\sum_{i=1}^{m} f_{ij} = d_j, \ \forall\ j = 1, ..., n \tag{4}$$

$$\sum_{j=1}^{n} f_{ij} = s_i, \ \forall\ i = 1, ..., m \tag{5}$$

and $f_{ij} \geq 0$. The constraints imply that each supplier has to fully distribute the good to consumers and also that each consumer needs to fully satisfy its requirements in the good.

4 Non-linear Programming Problems with Equality Constraints

In this part of the course, we will discuss methods for solving a class of nonlinear constrained optimization problems that can be formulated as:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t} \quad & h_i(x) = 0, \quad i = 1, ..., m \\ & g_j(x) \leq 0, \quad j = 1, ..., p, \end{aligned}$$

where $\mathbf{x} \in \Re^n$, $f : \Re^n \to \Re$, $g_j : \Re^n \to \Re$, and $m \leq n$.

In particular, we will first consider the class of Non-linear Programming problems with constraints that can expressed as equalities $\{h_i(\mathbf{x}) = 0, i = 1, \ldots, m\}$. A point $\mathbf{x}_0$ is called feasible point if it satisfies the constraints.

The constraints $h_i(\mathbf{x}) = 0,\ i = 1, ..., m$ define a surface $\mathcal{S} = \{\mathbf{x} : h_i(\mathbf{x}) = 0,\ i = 1, ..., m\}$.

The tangent plane at a point $\mathbf{x}_0$ on the surface $\mathcal{S}$ is the collection of derivatives at point $\mathbf{x}_0$ of all differentiable curves on $\mathcal{S}$ passing through $\mathbf{x}_0$. A tangent plane to a surface can be visualized as generalizing the tangent line to a point on a curve.

Problem: We would like to find an explicit characterization of the tangent plane at a point $\mathbf{x}_0$ on the surface defined by the constraints of our problem,

$$\mathcal{S} = \{\mathbf{x} : h_i(\mathbf{x}) = 0,\ i = 1, ..., m\} \tag{6}$$

as a function of the gradients of the constraint functions $h_i,\ i = 1, \ldots, m$. For a point $\mathbf{x}_0 \in \mathcal{S}$, we introduce the set of points

$$\mathcal{M} = \left\{\mathbf{y} : \nabla h_i^T(\mathbf{x}_0)\mathbf{y} = 0, \forall i = 1, \ldots, m\right\} \tag{7}$$

Definition: A point $\mathbf{x}_0 \in \mathcal{S}$ is said to be a *regular point* if vectors $\nabla h_1(\mathbf{x}_0), \ldots, \nabla h_m(\mathbf{x}_0)$ are linearly independent.

Theorem: At a regular point $\mathbf{x}_0 \in \mathcal{S} = \{\mathbf{x} : h_i(\mathbf{x}) = 0,\ i = 1, ..., m\}$, the tangent plane is $\mathcal{M}$.

Thus, at regular points we can characterize the tangent plane in terms of the gradients of the constraint functions.

Example 1: Consider the surface $\mathcal{S} = \left\{\mathbf{x} \in \Re^3 : h_1(\mathbf{x}) = x_1 = 0,\ h_2(\mathbf{x}) = x_1 - x_2 = 0\right\}$. The surface is clearly $\mathcal{S} = \{\mathbf{x} = (0, 0, x_3) : x_3 \in \Re\}$, namely the x_3-axis.

At a point $\mathbf{x}_0 \in \mathcal{S}$, it is

$$\nabla h_1(\mathbf{x_0}) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$
$$\nabla h_2(\mathbf{x_0}) = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$$

So, $\nabla h_1(\mathbf{x}_0), \nabla h_2(\mathbf{x}_0)$ are linearly independent $\forall \mathbf{x}_0 \in \mathcal{S}$. So the tangent plane at $\mathcal{S}$ at point $\mathbf{x}_0$ is:

$$\mathcal{M} = \left\{\mathbf{y} : \nabla h_i^T(\mathbf{x}_0)\mathbf{y} = 0\right\} = \left\{\mathbf{y} : \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} y_1 + \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} y_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right\} \Rightarrow \tag{8}$$

$$\mathcal{M} = \{(0,0,y_3) : y_3 \in \Re\} \qquad \blacksquare$$

Definition (as reminder):

1. The vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are linearly independent if and only if the equation $\lambda_1\mathbf{u}_1 + \ldots + \lambda_n\mathbf{u}_n = \mathbf{0}$ has as solution *only* the all-zero vector $(\lambda_1, \ldots, \lambda_n) = (0, \ldots, 0)$.
2. If the equation above has more solutions (essentially non-zero) other than the all-zero one, then vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are called linearly dependent.

Note: One vector $\mathbf{u}$ by itself is linearly dependent or independent? If $\mathbf{u} \neq \mathbf{0}$ then $\lambda\mathbf{u} = \mathbf{0} \Rightarrow \lambda = 0 \Rightarrow \mathbf{u}$: linearly independent. But if $\mathbf{u} = \mathbf{0}$ then the equation $\lambda\mathbf{u} = 0$ has several (infinite) solutions. So, $\mathbf{u} = 0$ is linearly dependent.

Example 2: Consider the surface $\mathcal{S} = \left\{(x_1, x_2) : h(x_1, x_2) = x_1^2 = 0\right\} = \{(0, x_2), x_2 \in \Re\}$.

We have $\nabla h(x_1, x_2) = \left[\begin{array}{cc} 2x_1 & 0 \end{array}\right] \left\{\begin{array}{l} \text{if } x_1 \neq 0 \text{ then } \nabla h \text{ is linearly independent} \\ \text{if } x_1 = 0 \text{ then } \nabla h \text{ is linearly dependent} \end{array}\right.$

$$\mathcal{M} = \left\{ (y_1, y_2) : \left[\begin{array}{cc} 0 & 0 \end{array}\right] \left(\begin{array}{c} y_1 \\ y_2 \end{array}\right) = \mathbf{0} \right\}$$

In this example we cannot define the tangent plane at non-regular points. From now on, unless otherwise stated, we will consider surfaces $\mathcal{S} = \{\mathbf{x} : h_i(\mathbf{x}) = 0,\ i = 1, ..., m\}$ that have all their points regular.

Lemma (for one equality constraint): Let $\mathbf{x_0}$ be a regular point of the surface defined by the equality constraints, $\mathcal{S} = \{h(\mathbf{x}) = 0\}$ and $\mathbf{x}_0$ is a local minimizer of $f : \Re^n \to \Re$, subject to the constraint $h(\mathbf{x}) = 0$. Then all points $\mathbf{y}$ that satisfy $\nabla h(\mathbf{x}_0)^T\mathbf{y} = 0$ also satisfy $\nabla f(\mathbf{x_0})^T\mathbf{y} = 0$. So, that means that both vectors $\nabla h, \nabla f$ are orthogonal to vectors $\mathbf{y}$ on the tangent plane at point $\mathbf{x}_0$ of the surface $\mathcal{S}$ and this means that they are *parallel* to each other. Hence we arrive at the theorem of Lagrange for one constraint $m = 1$ which is stated as follows:

Lagrange Theorem for $m = 1$ **constraint:** Let the point $\mathbf{x}_0$ be a local minimizer of $f : \Re^n \to \Re$ subject to the constraint $h\mathbf{x} = 0$, $h : \Re^n \to \Re$. Then, $\nabla f(\mathbf{x}^*)$ and $\nabla h(\mathbf{x}^*)$ are parallel. That is, if $\nabla h(\mathbf{x}^*) \neq 0$,then there exists a scalar λ^* such that $\nabla f(\mathbf{x}^*) + \lambda^*\nabla h(\mathbf{x}^*) = \mathbf{0}$.

The theorem above provides a first-order necessary condition for a point to be a local minimizer of $f(\cdot)$ subject to an equality constraint.

An Example is shown in Figure 1 (where note that the curve $h = 0$ corresponding to the constraints should be intersecting with line $f = f_2$ at a point $\mathbf{x}_0$ so that the gradients of f and h are at the same point).

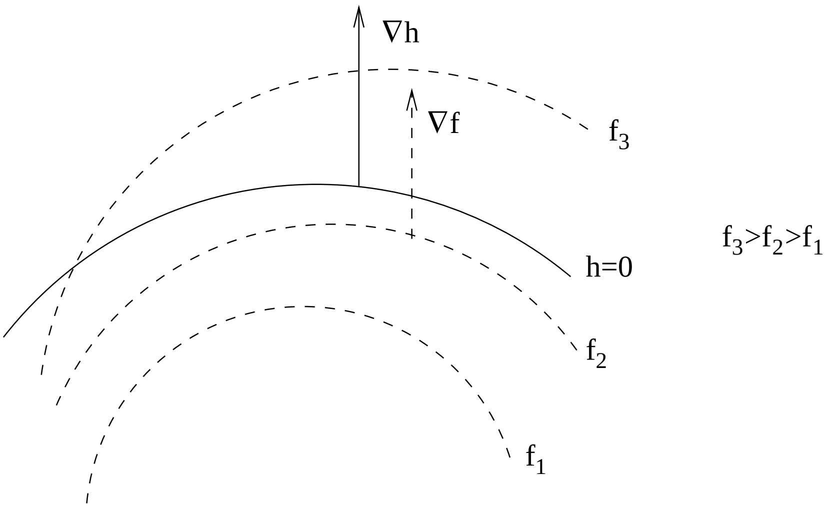


Figure 1: Example for the theorem of Lagrange.

Lagrange Theorem for $m > 1$ constraints: Let the point $\mathbf{x}^*$ be a local minimizer of $f : \Re^n \to \Re$ subject to the constraints $h_1(\mathbf{x}) = 0, \ldots, h_m(\mathbf{x}) = 0$. Assume that $\mathbf{x}^*$ is a regular point. Then, there exists a real vector $\boldsymbol{\lambda} \in \Re^m : \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \nabla h(\mathbf{x}^*) = 0$.

Thus, at an optimal point (if this is optimal), the gradient of the objective function can be written as linear combination of the gradients of the constraints.

Example: Consider the following problem which is depicted in figure 2:

$$\begin{aligned} \min \quad & f(x_1, x_2) = x_1 + x_2 \\ \text{s.t} \quad & (x_1 - 1)^2 + x_2^2 - 1 = 0, \quad (h_1(x_1, x_2) = 0) \\ & (x_1 - 2)^2 + x_2^2 - 4 = 0, \quad (h_2(x_1, x_2) = 0) \end{aligned}$$

We have:

$$\nabla h_1(x_1, x_2) = (2\,(x_1 - 1)\,, 2x_2)$$

$$\nabla h_2(x_1, x_2) = (2\,(x_1 - 2)\,, 2x_2)$$

The surface $\mathcal{S}$ is the point $(0, 0)$. So we have

$$\nabla h_1\,(0, 0) = (-2, 0)$$

$$\nabla h_2\,(0, 0) = (-4, 0).$$

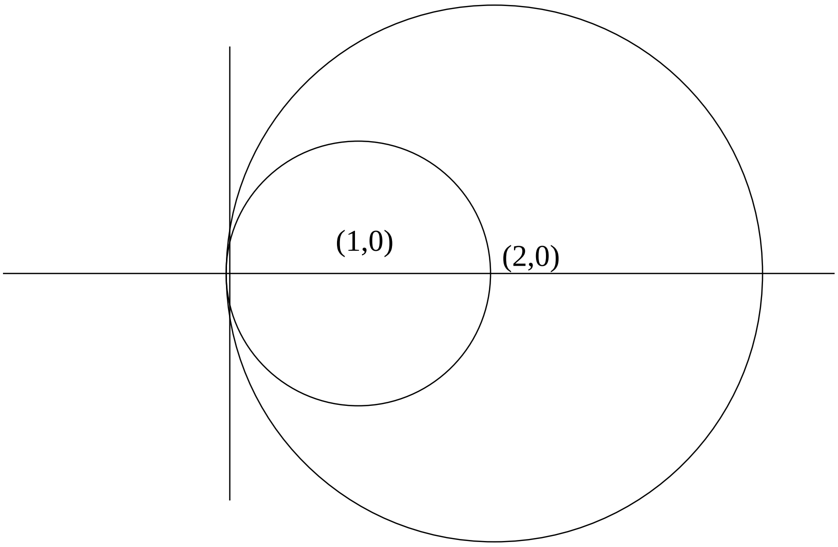


Figure 2: Lagrange theorem..

Now we try to confirm Lagrange theorem: $\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h(\mathbf{x}^*) = 0 \Rightarrow$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \lambda_1 \begin{pmatrix} -2 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} -4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow$$

$1 - 2\lambda_1 - 4\lambda_2 = 0 \Rightarrow$

$1 = 0.$

So, because of $(0,0)$ is not a regular point, we can't apply Langrange theorem here! The condition cannot hold for any λ_1, λ_2, namely the gradient of the objective function cannot be expressed as a linear combination of the constraints. ■

Recall again the form of the problem we are considering here:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t} \quad & h_1(\mathbf{x}) = 0 \\ & \vdots \\ & h_m(\mathbf{x}) = 0 \end{aligned}$$

Define the *Lagrangian function* at point $\mathbf{x}$ as:

$$L(\mathbf{x}, \lambda_1, \ldots, \lambda_m) = f(\mathbf{x}) + \lambda_1 h_1(\mathbf{x}) + \ldots + \lambda_m h_m(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) \tag{9}$$

The Hessian matrix of the Lagrangian at point $\mathbf{x}$, $\Lambda(\mathbf{x}, \lambda_1, \ldots, \lambda_m)$ is defined as

$$\Lambda(\mathbf{x}, \lambda_1, \ldots, \lambda_m) = F(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i H_i(\mathbf{x}), \tag{10}$$

where $F(\mathbf{x})$ is the Hessian matrix of the objective function f at point $\mathbf{x}$, given by

$$F(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_m} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

where the partial derivatives are evaluated at point $\mathbf{x}$ and $H_i(\mathbf{x})$ is the Hessian matrix of h_i at point $\mathbf{x}, i = 1, .., m$.

5 Second-Order Necessary and Sufficient Conditions

We now state the necessary conditions of second-order for the existence of local minimum. Then, we state second-order sufficient conditions for existence of local minimum.

Second order necessary conditions for existence of local minimum: Let point $\mathbf{x}^*$ be a local minimizer of $f : \Re^n \rightarrow \Re$ subject to the constraints $h_1(\mathbf{x}) = 0, ..., h_m(\mathbf{x}) = 0$. Suppose $\mathbf{x}^*$ is a regular point. Then, there exists vector $\boldsymbol{\lambda} \in \Re^m$ such that:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0} \tag{11}$$

Also let $\mathcal{M} = \{\mathbf{y} : \nabla h_i(\mathbf{x}^*)\mathbf{y} = 0, i = 1, ..., m\}$ be the tangent plane at point $\mathbf{x}^*$ of the surface defined by the equality constraints. Then, matrix

$$\Lambda\left(\mathbf{x}^*, \lambda\right) = F(\mathbf{x}^*) + \lambda_1 H_1(\mathbf{x}^*) + ... + \lambda_m H_m(\mathbf{x}^*) \tag{12}$$

is positive-semidefinite on $\mathcal{M}$. That is,

$$\mathbf{y}^T \Lambda\left(\mathbf{x}^*, \lambda\right) \mathbf{y} \geq 0, \; \forall \mathbf{y} \in \mathcal{M} \tag{13}$$

Second order sufficient conditions for existence of local minimum:

If there exists a real vector $\boldsymbol{\lambda} \in \Re^m$ which, at a point $\mathbf{x}^*$ satisfies

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = 0 \tag{14}$$

and if matrix

$$\Lambda\left(\mathbf{x}^*, \lambda\right) = F(\mathbf{x}^*) + \lambda_1 H_1(\mathbf{x}^*) + ... + \lambda_m H_m(\mathbf{x}^*) \tag{15}$$

is positive definite on $\mathcal{M}$ (where $\mathcal{M} = \left\{\mathbf{y} : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0, i = 1, ..., m\right\}$) that is, $\mathbf{y}^T \Lambda\left(\mathbf{x}^*, \lambda\right) \mathbf{y} > 0$, $\forall \mathbf{y} \in \mathcal{M}$,

then $\mathbf{x}^*$ is a local minimizer of $f : \Re^n \to \Re$ subject to the constraints $h_i(\mathbf{x}) = 0, i = 1, .., m$.

Note: Note that, unless otherwise stated, in the cases we will consider, we will assume that $\mathcal{M} = \Re^n$.

Advanced Topics on Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 21 : NLP Sensitivity analysis and examples - 6/12/06

Notes by : Alexandra Xamilothori and Konstantinos Gerogiokas

1 Lecture outline

- Conditions for convex functions and maximization problems
- Sensitivity analysis
- Examples

Consider the minimization problem we saw in the previous lecture

$$\min f(\mathbf{x}) \text{ such that } h_i(\mathbf{x}) = 0, \text{ for } i = 1, \ldots, m. \tag{1}$$

2 Sufficient conditions minimum for *convex* functions

Assume that the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is a *convex* function and the constraint functions $h_i(\mathbf{x})$, with $h_i : \mathbb{R}^n \to \mathbb{R}$ are also convex functions which define the surface of constraints $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n, h_i(\mathbf{x}) = 0, i = 1, \ldots, m\}$. If there exists $\mathbf{x}^* \in \mathcal{S}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = 0$$

then the $\mathbf{x}^*$ is global minimum of f.

Note 1: In the case of convex functions, the sufficient conditions for existence of global minimum are obtained only by equating the partial derivatives of the Lagrangian to zero. The Hessian matrices of functions f, h_i are positive-definite and the second condition is not needed. In the case of convex functions, the local minimum is global minimum.

Note 2: The factors $\lambda_i, i = 1, \ldots, m$ are called *Lagrange multipliers* regardless if functions are convex or not.

3 Interpretation of Lagrange multipliers

Consider again the NLP problem with one equality constraint,

$$\begin{gathered}\min f(\mathbf{x}) \\ \text{s.t. } h(\mathbf{x}) = 0\end{gathered}$$

Let λ be the Lagrange multiplier corresponding to the one equality constraint. Assume a small variation c in the right-hand side of the constraint. Note that, as in the LP case, the constraint often specifies requirements in resources. Let $\mathbf{x}^*(0)$ be the optimal solution to the problem by having the constraint $h(\mathbf{x}) = 0$ and the value of the objective function is $f(\mathbf{x}^*(0))$. Let $\mathbf{x}^*(c)$ be the optimal solution by having the constraint $h(\mathbf{x}) = c\,,\ c \in \mathbb{R}$, and let the corresponding value of the objective function be $f(\mathbf{x}^*(c))$.

We calculate the rate of change of the value of the objective function for small variations of the constraints,

$$\frac{df(\mathbf{x}(c))}{dc} = \frac{\partial f(\mathbf{x}(c))}{\partial x_1}\frac{dx_1(c)}{dc} + \ldots + \frac{\partial f(\mathbf{x}(c))}{\partial x_n}\frac{dx_n(c)}{dc} = \nabla f(\mathbf{x}(c)^T \mathbf{x}'(c) \tag{2}$$

where $\mathbf{x}'(c)$ is the vector of derivatives $(dx_1(c)/dc, \ldots, dx_n(c)/dc)$. From the first-order conditions we have

$$\begin{gathered}\nabla f(\mathbf{x}(c)) + \lambda \nabla h(\mathbf{x}(c)) = \mathbf{0} \Rightarrow \\ \frac{\partial f(\mathbf{x}(c))}{\partial x_i} + \lambda \nabla \frac{\partial h(\mathbf{x}(c))}{\partial x_i} = 0 \text{ for } i = 1, \ldots, m \Rightarrow \\ \frac{\partial f(\mathbf{x}(c))}{\partial x_i} = -\lambda \frac{\partial h(\mathbf{x}(c))}{\partial x_i} \text{ for} i = 1, \ldots, m.\end{gathered}$$

We substitute in the equation above and we have:

$$\frac{df(\mathbf{x}(c))}{dc} = -\lambda \left(\frac{\partial h(\mathbf{x}(c))}{\partial x_1}\frac{dx_1(c)}{dc} + \ldots + \frac{\partial h(\mathbf{x}(c))}{\partial x_n}\frac{dx_n(c)}{dc} \right) = -\lambda$$

since if we differentiate both sides of the constraint $h(\mathbf{x}) = c$ with respect to c, we get that $\frac{\partial h(\mathbf{x}(c))}{\partial x_1}\frac{dx_1(c)}{dc} + \ldots + \frac{\partial h(\mathbf{x}(c))}{\partial x_n}\frac{dx_n(c)}{dc} = 1$. Therefore, λ is interpreted as the rate of change in the value of the objective function per unit of change in the resources. Thus it represents the *price* of the unit of constraint requirement.

3.1 Sensitivity analysis for $m > 1$ constraints

Consider the NLP problem with more than one constraints:

$$\min f(\mathbf{x})$$
$$\text{s.t. } h_1(\mathbf{x}) = 0,\ \ldots, h_m(\mathbf{x}) = 0.$$

Note that all constraints can be collectively described by vector $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. Let $\boldsymbol{\lambda}$ be the vector of Lagrange multipliers associated with the constraints.

Now let $\mathbf{c} \in \mathbb{R}^m$ denote a vector of small changes in the right-hand sides of the constraints. Similarly with the case of one constraint, we can define $\mathbf{x}(\mathbf{0})$ the optimal solution for constraints $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{x}(\mathbf{c})$ the optimal solution for constraints $\mathbf{h}(\mathbf{x}) = \mathbf{c}$. By following the methodology for the case of one constraint, we can prove:

$$\nabla_{\mathbf{c}} f(\mathbf{x}(\mathbf{c})) = -\boldsymbol{\lambda} \Rightarrow \begin{pmatrix} \frac{\partial f(\mathbf{x}(\mathbf{c}))}{\partial c_1} \\ \vdots \\ \frac{\partial f(\mathbf{x}(c))}{\partial c_n} \end{pmatrix} = \begin{pmatrix} -\lambda_1 \\ \vdots \\ -\lambda_n \end{pmatrix}$$

where

$$\lambda_i = \frac{\partial f(\mathbf{x}(\mathbf{c}))}{\partial c_i} \tag{3}$$

is again the price per unit of the resource i, or equivalently the rate of the of the value of the objective function with regard to small changes in the resource (constraint) i.

Remark - Local and global maximum of NLP problems

For the maximization problem

$$\max f(\mathbf{x})$$
$$\text{s.t. } h_1(\mathbf{x}) = 0\ ,\ \ldots, h_m(\mathbf{x}) = 0$$

the second-order sufficient conditions for existence of local maximum are as follows: If there exist $\mathbf{x}^*, \boldsymbol{\lambda}$ such that the gradient of the Lagrangian function is zero,

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h(\mathbf{x}^*) = \mathbf{0}$$

and the Hessian matrix of the Lagrangian

$$\Lambda(\mathbf{x}^*, \boldsymbol{\lambda}) < 0,$$

i.e, the matrix is negative-definite, then $\mathbf{x}^*$ is local maximum of f under the constraints $h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$.

Note: If $f(\mathbf{x})$ is a concave function and the constraint functions $h_i(\mathbf{x})$, $i = 1, \ldots, m$ are also concave, then $\mathbf{x}^*$ is a global maximum of f in the constrained maximization problem.

Example 1: Consider the minimization problem,

$$\min \tfrac{1}{2}\mathbf{x}^T Q \mathbf{x}$$

$$\text{subject to } A\mathbf{x} = \mathbf{b}$$

where $Q > 0$ is symmetric, positive definite matrix, $A \in \mathbb{R}^{m \times n}, m < n$ and $b \in \mathbb{R}^m$.

From equation $A\mathbf{x} = b$ we get a Lagrangian multiplier vector $\boldsymbol{\lambda}$. The Lagrangian function is:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x})$$

The objective function $\frac{1}{2}\mathbf{x}^T Q\mathbf{x}$ is convex because $\nabla f(\mathbf{x}) = Q\mathbf{x}$ and the Hessian is $F(\mathbf{x}) = Q > 0$. The Lagrange condition for existence of minimum is:

$$\nabla_{\mathbf{x}}\text{Ł}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \quad \Rightarrow$$

$$Q\mathbf{x} - A^T\boldsymbol{\lambda} = \mathbf{0} \quad \Rightarrow$$

and the optimal solution satisfies:

$$\mathbf{x}^* = Q^{-1}A^T\boldsymbol{\lambda}$$

To find $\boldsymbol{\lambda}$, we use the fact that $\mathbf{x}^*$ is a feasible point, so it satisfies the constraints. So:

$$A\mathbf{x} = \mathbf{b} \quad \Rightarrow$$

$$AQ^{-1}A\boldsymbol{\lambda} = \mathbf{b} \quad \Rightarrow$$

$$\boldsymbol{\lambda} = (AQ^{-1}A)^{-1}\mathbf{b}$$

Thus,

$$\mathbf{x}^* = Q^{-1}A^T(AQ^{-1}A^T)^{-1}\mathbf{b} \tag{4}$$

is the global minimum of our problem.

Example 2: Consider the problem

$$\max f(\mathbf{x}) = x_1x_2 + x_2x_3 + x_1x_3$$

$$\text{subject to: } x_1 + x_2 + x_3 = 3.$$

Solution: The Lagrangian function is:

$$L(x_1, x_2, x_3, \lambda) = x_1x_2 + x_2x_3 + x_1x_3 + \lambda(x_1 + x_2 + x_3 - 3)$$

We equate the partial derivatives $\frac{\partial L}{\partial x_i} = 0, i = 1, 2, 3$, and we get the equations below:

$$x_2 + x_3 + \lambda = 0 \tag{5}$$

$$x_1 + x_3 + \lambda = 0 \tag{6}$$

$$x_1 + x_2 + \lambda = 0 \tag{7}$$

and we also have the constraint:

$$x_1 + x_2 + x_3 = 3 \tag{8}$$

Solving the 4×4 system of equations, we have:

$$x_1^* = 1, \quad x_2^* = 1, \quad x_3^* = 1, \quad \lambda = -2$$

For the objective function $f(\mathbf{x})$, we get the gradient vector:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} x_2 + x_3 \\ x_1 + x_3 \\ x_1 + x_2 \end{pmatrix}$$

The Hessian matrix is:

$$F(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

and does not depend on $\mathbf{x}$. The Hessian matrix for the constraint $h(x) = x_1 + x_2 + x_3 - 3$ is:

$$H(x) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Thus,

$$\Lambda(\mathbf{x}^*, \lambda) = F(\mathbf{x}^*) + \lambda H(\mathbf{x}^*) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

since $H(\mathbf{x}^*) = 0$. Now, we don't know whether the Hessian matrix is positive- or negative- definite in $\mathbb{R}^3$. As a result,we cannot claim if $\forall \mathbf{y} \in \mathbb{R}^3 : \mathbf{y}^T \Lambda \mathbf{y}$ is a positive or negative quantity. But, we take into account the *precise* formulation of the sufficient condition, which states that the Hessian should be positive-definite (negative-definite) on the tangent plane to the surface defined by the constraints, so as to have local minimum (local maximum) in the problem. The tangent plane to the surface defined by the (one and only) constraint in the problem,

$$h(\mathbf{x}) = 0 \quad \Rightarrow \quad x_1 + x_2 + x_3 - 3 = 0$$

is

$$\mathcal{M} = \{\begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix} : \nabla h(\mathbf{x})^T \mathbf{y} = 0\} = \{\mathbf{y} : \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = 0\} \Rightarrow \tag{9}$$

and finally $\mathcal{M} = \{ \quad (y_1, y_2, y_3) : y_1 + y_2 + y_3 = 0\}$. We examine if the Hessian matrix is positive- or negative-definite at the tangent plane $\mathcal{M}$. We have

$$\mathbf{y}^T \Lambda \mathbf{y} = \begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = y_1(y_2 + y_3) + y_2(y_1 + y_3) + y_3(y_1 + y_2) \tag{10}$$

We have:

$$y_2 + y_3 = -y_1, \quad y_1 + y_3 = -y_2, \quad y_1 + y_2 = -y_3$$

and thus $\mathbf{y}^T \Lambda \mathbf{y} = -y_1^2 - y_2^2 - y_3^2 \leq 0$. Thus, the Hessian matrix is negative-definite and $\mathbf{x}^* = (1, 1, 1)$ that was found above is local maximum to the problem.

Example 3: Consider the problem

$$\max \frac{\mathbf{x}^T Q \mathbf{x}}{\mathbf{x}^T P \mathbf{x}} \tag{11}$$

with matrix $Q = Q^T \geq 0$ and $P = P^T > 0$ (Q, P are symmetric and positive-definite matrices).

In the problem above, if $\mathbf{x}$ is an optimal solution, then all multiples $a\mathbf{x}$, $a \neq 0, a \in \mathbb{R}$ are optimal solutions too. In order to avoid the multiplicity of solutions, we set $\mathbf{x}^T P \mathbf{x} = 1$ and we get the constrained problem of maximizing $\mathbf{x}^T Q \mathbf{x}$ subject to $\mathbf{x}^T P \mathbf{x} = 1$.

The Lagrangian function is:

$$L(\mathbf{x}, \lambda) = \mathbf{x}^T Q \mathbf{x} + \lambda(1 - \mathbf{x}^T P \mathbf{x}) \tag{12}$$

and by the condition $\nabla L(\mathbf{x},\lambda)=\mathbf{0} \Rightarrow 2Q\mathbf{x}-2\lambda P\mathbf{x}=\mathbf{0} \Rightarrow Q\mathbf{x}=\lambda P\mathbf{x} \Rightarrow P^{-1}Q\mathbf{x}=\lambda\mathbf{x}$.

From the above, we observe that if $\mathbf{x}$ is a solution and maximizes $\mathbf{x}^TQ\mathbf{x}$ then it is an eigenvector which corresponds to some eigenvalue λ of matrix $P^{-1}Q$. Thus, suppose that $\mathbf{x}^*$ is optimal solution, then we have: $\mathbf{x}^{*T}P\mathbf{x}^*=1$ and then
$P^{-1}Q\mathbf{x}^*=\lambda^*\mathbf{x}^* \Rightarrow PP^{-1}Q\mathbf{x}^*=\lambda^*P\mathbf{x}^* \Rightarrow Q\mathbf{x}^*=\lambda^*P\mathbf{x}^* \Rightarrow \mathbf{x}^{*T}Q\mathbf{x}^*=\lambda^*\mathbf{x}^{*T}P\mathbf{x}^* \Rightarrow \lambda^*=\mathbf{x}^{*T}Q\mathbf{x}^*$ where λ^* is the Lagrange multiplier at the optimal solution.

Note that λ^* must be one of the n eigenvalues of $P^{-1}Q$, which are $\lambda_1<\lambda_2<...<\lambda_n$). In particular, λ^* is the maximum eigenvalue of matrix $P^{-1}Q$ and the optimal solution $\mathbf{x}^*$ is the eigenvector which corresponds to the maximum eigenvalue λ^* of $P^{-1}Q$.

Example 4: Solve the problem

$$\min \mathbf{c}^T\mathbf{x}, \tag{13}$$

subject to:

$$\sum_{i=1}^{n} x_i = 0 \text{ and } \sum_{i=1}^{n} x_i^2 = 1 \tag{14}$$

Solution:

$$x_i^* = \frac{c_i+\lambda^*}{\mu^*}, \tag{15}$$

with

$$\lambda^* = -\frac{1}{n}\sum_{i=1}^{n} c_i, \text{ and } \mu^* = \pm\frac{1}{2}n^2\sqrt{\frac{1}{n}\sum_{i=1}^{n} c_i^2 - \frac{1}{n}(-n\lambda^*)^2} \tag{16}$$

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 22 : Example : Beamforming - 11/12/06

Notes by : George Noutsis and George Xatziparaskevas

1 Lecture outline

- Example : Beamforming

2 Example

We continue on the topic of Non-linear programming problems with equality constraints and we will solve the problem:

$$\min \mathbf{x}^T A \mathbf{x}$$

$$\text{s.t. } \mathbf{c}^T \mathbf{x} = 1$$

where $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{c} = (c_1, \ldots, c_n)$ and A is a matrix of dimension $n \times n$. Let λ be the Lagrange multiplier associated with the constraint of the problem. Thus the Lagrangian is,

$$L(\mathbf{x}, \lambda) = \mathbf{x}^T A \mathbf{x} + \lambda(\mathbf{c}^T \mathbf{x} - 1) \tag{1}$$

We use the conditions $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0 \Leftrightarrow \frac{\partial L}{\partial x_i} = 0$ and we have:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0 \Rightarrow 2A\mathbf{x} + \lambda \mathbf{c} = 0 \Rightarrow \mathbf{x}^* = -\frac{1}{2}\lambda A^{-1}\mathbf{c}.$$

Now, we use the constraint $\mathbf{c}^T \mathbf{x} = 1$ to find the value of the Lagrange multiplier λ.

$$\mathbf{c}^T \mathbf{x} = 1 \Leftrightarrow -\frac{1}{2}\lambda \mathbf{c}^T A^{-1} \mathbf{c} = 1 \Leftrightarrow \lambda = \frac{-2}{\mathbf{c}^T A^{-1} \mathbf{c}}$$

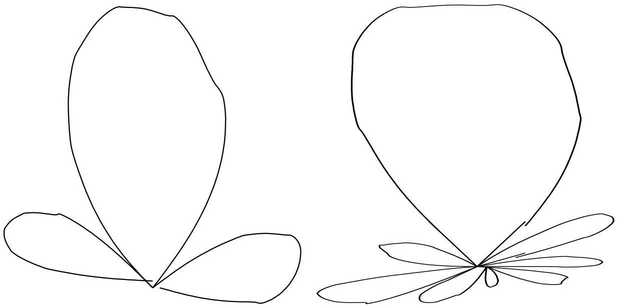


Figure 1: Different shapes of the antenna array radiation diagram as a result of controlling the electric current phases and amplitudes.

Thus, the optimal soluton is

$$\mathbf{x}^* = \frac{A^{-1}\mathbf{c}}{\mathbf{c}^T A^{-1}\mathbf{c}} \tag{2}$$

As an application of this optimization problem, we will study the fundamental problem that arises in the case of beam-forming.

3 Beamforming

In the case that we do not have a single omni-directional antenna but an array of omni-directional antennas, we can adapt the radiation diagram by changing the amplitudes and phases of the alternate electric currents that feed the antenna. Thus, for example, for an antenna array of M elements, the radiation diagram (namely the width and length of the main lobe and the angle of the lobe) is a function of of the complex numbers $\{I_i e^{j\phi_i}\}_{i=1}^M$, where I_i is the amplitude and ϕ_i is the phase of the alternate current which stimulate the antenna element i. Thus, we can control the radiation diagram and dynamically change the shape and form and make diagrams like the ones depicted in figure 1. Note that in the radiation diagram, most of the transmission power is concentrated towards a given direction, that of the main lobe. There also exist several side lobes as well.

An antenna array of controllable radiation diagram is called *adaptive antenna array* or *smart antenna* and the control of the radiation diagram is called *beam-forming.* Clearly, the radiation diagram can be controlled either in order to transmit or to receive a signal. A smart antenna can adapt its radiation diagram according e.g. to the instantaneous position of the user. The following advantages exist for an adaptive antenna array:

1. Minimization of interference. Beam-forming can take place either in the reception or in trans-

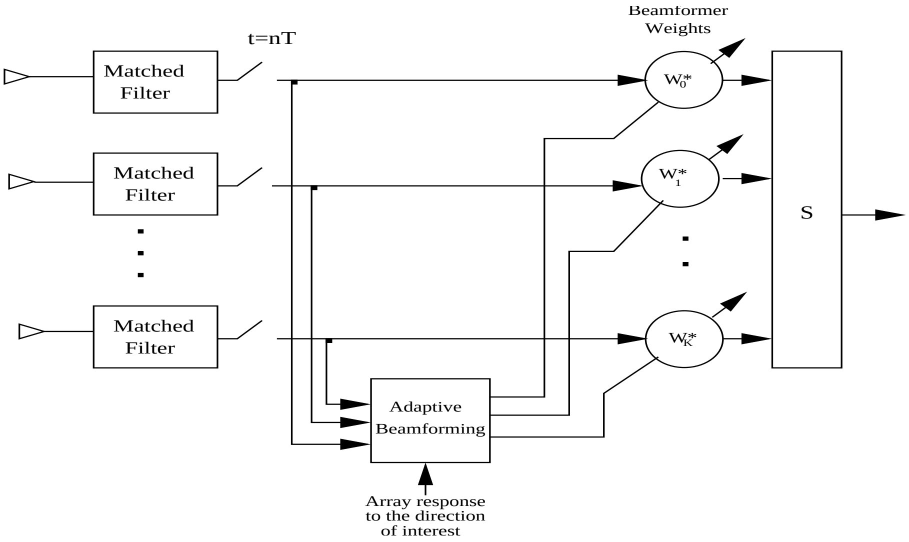


Figure 2: Block diagram of a receiver with an adaptive antenna array and a beamformer.

mission. We can shape the diagram in such a way that we can transmit or receive from a certain direction, that specified by the main lobe. For reception, the antenna array can receive signals only emitted from certain directions and attenuate signals emitted from other directions. The same holds for transmission.

2. Capacity increase. A transmitter can transmit at the same conventional channel (frequency or timeslot) to more than one users. Also, the same holds for the case of reception. In that case, a different radiation diagram is formed for each user. Clearly, there exists a M-fold increase in system capacity if the antenna array can serve at the same channel M users simultaneously (in the same frequency and time slot).

If we have M antenna elements in the antenna array, there can be $k \leq M$ radiation diagrams, one for each user. For simplicity assume that $M = 2$ here. The antenna can form at most two radiation diagrams and each diagram corresponds to a complex vector $\mathbf{w}_1 = (w_{11}, w_{12})$ where $w_{11} = I_1 e^{j\phi_1}$ and $w_{12} = I_2 e^{j\phi_2}$, and $\mathbf{w}_2 = (w_{21}, w_{22})$, defined similarly. The two vectors define the two radiation diagrams (basically the main lobes) and each radiation diagram can serve one user (if $\mathbf{w}_1$ and $\mathbf{w}_2$ are linearly independent, the antenna array can serve simultaneously both users). Each radiation diagram corresponds to a vector with dimension equal to the number of antenna elements.

We will now assume that the vectors are real numbers and we will not further worry about complex numbers. Still, the theory can be extended to cover the complex number case. We will also

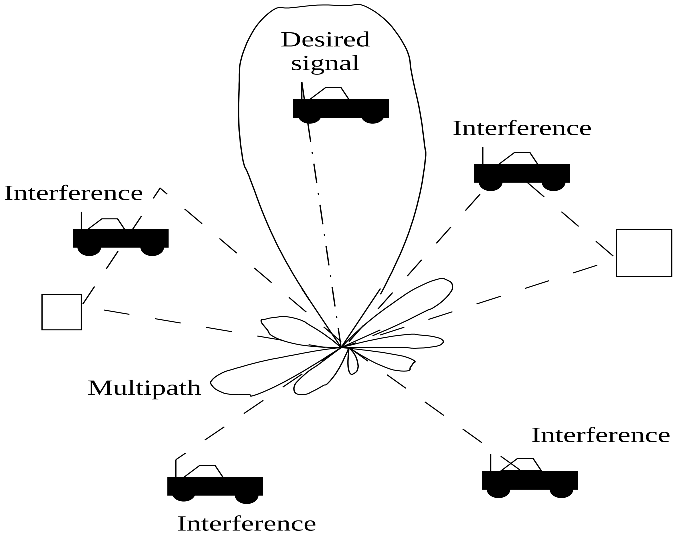


Figure 3: Example of an antenna array radiation diagram.

concentrate on the case where the signal of several users is received at a base station and the base station will attempt to discriminate the signal of only one user (the one of interest) by computing the beam-forming vector **w**.

The main signal processing segment at the receiver is the adaptive beamformer (Figure 2. The adaptive beamformer finds vector **w**. Its task is to find M numbers $(w_1, \ldots, w_M) = \mathbf{w}$. The signal y_i that has reached the i-th antenna is multiplied by w_i. Then we have to sum the above products to find the interior product $\mathbf{w}^T \mathbf{y} = w_1 y_1 + \ldots w_M y_M$ as the total outcome of *combining* all received signals at antennas by an appropriate number. That is, the output from each array element i is weighted by a weight w_i and added. The objective is to search for the w_i's such that the Signal-to-Noise Ratio (SNR) of the signal of the user of interest is maximized. The SNR is taken at the output, after the summation in figure 2.

We define the antenna array response vector to the direction of arrival θ as $\mathbf{v}(\theta) = (v_i(\theta), \ldots, v_M(\theta))$ that shows how each antenna receives a signal coming from an angle θ. The received signal (vector signal) at the M antennas at some frozen time t is

$$\mathbf{y}(t) = \sum_{j=1}^{K} \sqrt{P_j G_j} \sum_{\ell=1}^{L} a_j \mathbf{v}_j(\theta_\ell) s_j(t - \tau_j) + \mathbf{n}(t)$$

where:

K: number of users.

P_j: transmission power of the user j

G_j: path gain (denoting the distance loss) between user j and the BS. It is given by $G_j = \frac{1}{d_j^\gamma}$, where d_j is the distance from user j to the BS and γ is a constant that depends on the environment.

L: The number of paths of the multi-path (assume each user has its signal arriving through L paths). Each of these (user j) paths has:

α_j^ℓ: Attenuation factor of path ℓ of user j because of shadowing (this is a random number, usually log-normally distributed).

θ_ℓ: angle of arrival of ℓ-th path.

$\mathbf{v}_j(\theta_\ell)$: Response vector of the antenna to a signal which is sent from user j and comes from path angle θ_ℓ.

s_j: the transmitted signal of user j.

τ_j: Signal delay for the signal of user j.

$\mathbf{n}(t)$: A vector that indicates noise at the receiver in each antenna.

Each user j can be completely specified by a vector called *spatial signature* of user j,

$$\mathbf{a}_j = \sum_{\ell=1}^{L} \alpha_j^\ell \mathbf{v}_j(\theta_\ell). \tag{3}$$

As we can see, the spatial signature depends on

- position of user i,
- number of paths,
- direction of arrival (DoA) of each path,
- shadowing coefficient of each path.

Thus, we have:

$$\mathbf{y}(t) = \sum_{j=1}^{k} \sqrt{P_j G_j} \mathbf{a}_j s_j(t - \tau_j) + \mathbf{n}(t)$$

Let the transmitted signal s_j by user j be represented as:

$$s_j(t) = \sum_n b_j(n) g(t - nT)$$

where,

$g(\cdot)$: pulse shaping filter function, specifying the shape of the pulse on which the bits will be carried.

T: The symbol time.

$\{b_j(n)\}, n = 1, \ldots,$: the sequence of bits.

At the receiver, we have the matched filter receiver (matched to the pulse shaping filter function of the transmitter) that is given by $g(t) = g^*(-t)$. The output of the matched filter is sampled at discrete times $t = nT$ (once in a symbol time) and we have the received discrete-time signal at the output of the matched filter as

$$\mathbf{y}(n) = \mathbf{y}(t) * g^*(-t)|_{t=nT}$$

Convolution of the received signal with the matched filter and sampling at $t = nT, n = 1, \ldots$ is equivalent to operation

$$\int_{(n-1)T}^{nT} \sum_n b(n) g(t - nT) g^* dt = b(n)$$

The receiver calculates the above integral in each symbol time interval. Thus, the signal from continuous-time becomes discrete-time:

$$\mathbf{y}(n) = \sum_{j=1}^{K} \sqrt{P_j G_j} \mathbf{a}_j b_j(n) + \mathbf{n}(n)$$

The expected power of the output signal after the beamforming and the multiplication with factors w_i is,

$$\mathbb{E}[e^2] = \mathbb{E}\left[|\mathbf{w}^T \mathbf{y}|^2 \right] = \mathbb{E}\left[\mathbf{w}^T \mathbf{y}\mathbf{y}^T \mathbf{w} \right] = \mathbf{w}^T \mathbb{E}[\underbrace{\mathbf{y}\mathbf{y}^T}_{A}] \mathbf{w}$$

Matrix A is of dimension $M \times M$ and each element $A_{ij} = \mathbb{E}[y_i y_j]$ shows the correlation between received signals at antennas i and j. We can see from the relation above that

$$A = \sum_{j=1}^{K} P_j G_j \mathbf{a}_j \mathbf{a}_j^T + \sigma^2 I$$

under the assumptions that user signals are zero-mean ($\mathbb{E}[|b_i(n)|] = 0$), different user signals are uncorrelated ($\mathbb{E}[b_i(n) b_j(n)] = 0$, for $i \neq j$), user signals are unit-power ($\mathbb{E}[|b_i^2(n)|] = 1$). Also each random variable representing noise at each antenna is Gaussian with zero mean and variance σ^2, and the noise variables at different antennas are uncorrelated:

$$\mathbb{E}[n_i n_j] = \begin{cases} 0 & ,\text{if } i \neq j \\ \sigma^2 & , \text{ if } i = j \end{cases}$$

The base station receives data from all K users and needs to calculate the beam-forming vector $\mathbf{w}$ to distinguish the signal of a user i. We can write the matrix A as consisting of two parts, one concerning the user of interest i and another concerning all other users (which is essentially interference),

$$A = \underbrace{P_i G_i \mathbf{a}_i \mathbf{a}_i^T}_{A_i} + \underbrace{\sum_{j \neq i} P_j G_j \mathbf{a}_j \mathbf{a}_j^T + \sigma^2 I}_{A_{\text{int}}}$$

Thus, we have:

$$\mathbb{E}[\mathbf{w}^T A \mathbf{w}] = \mathbf{w}^T (P_i G_i \mathbf{a}_i \mathbf{a}_i^T + A_{\text{int}}) \mathbf{w} = \mathbf{w}^T A_{\text{int}} \mathbf{w} + P_i G_i \|\mathbf{w}^T \mathbf{a}_i\|^2 \Rightarrow \mathbb{E}[e^2] = P_i G_i \|\mathbf{w}^T \mathbf{a}_i\|^2 + \mathbf{w}^T A_{\text{int}} \mathbf{w}$$

The expected signal power at the output comprises the signal power that originates from user i and the power that originates from all other users. Thus, we have for the signal-to-interference and noise ratio:

$$\text{SINR}_i = \frac{P_j G_j \|\mathbf{w}^T \mathbf{a}_i\|^2}{\mathbf{w}^T A_{\text{int}} \mathbf{w}},$$

The receiver wants to find the vector $\mathbf{w}$ to maximize SINR_i, so it faces the problem:

$$\max_{\mathbf{w}} \text{SINR}_i = \min_{\mathbf{w}} \frac{\mathbf{w}^T A_{\text{int}} \mathbf{w}}{P_j G_j \|\mathbf{w}^T \mathbf{a}_i\|^2}$$

This is equivalent to maintaining $\mathbf{w}^T \mathbf{a}_i = 1$ (fixed) at the direction of the user of interest and trying to minimize interference. Thus, the problem becomes:

$$\min_{\mathbf{w}} \mathbf{w}^T A_{\text{int}} \mathbf{w}$$

$$\text{subject to: } \mathbf{w}^T \mathbf{a}_i = 1$$

From the solution of the problem in the beginning of the lecture, we find that the optimal beamforming vector $\mathbf{w}^*$ is:

$$\mathbf{w}^* = \frac{A_{\text{int}}^{-1} \mathbf{a}_i}{\mathbf{a}_i^T A_{\text{int}}^{-1} \mathbf{a}_i}$$

Advanced Topics in Networking - Fall 2006

Instructor Iordanis Koutsopoulos

Lecture 23 : NLP problems with inequality constraints - 12/12/06

Notes by : Anastasia Narou and Maria Papadopoulou

1 Lecture outline

- Karush-Kuhn-Tucker theorem
- Necessary and sufficient conditions for local minimum in problems with inequality constraints

2 Definitions

In previous lectures, we studied non-linear programming problems with equality constraints. We will now generalize the theory to problems which also have inequality constraints. Namely, we will consider problems of the form:

$$\min f(x) \tag{1}$$

subject to:

$$h_i(\mathbf{x}) = 0,\ i = 1, ..., m,\ \text{and}\ g_j(\mathbf{x}) \leq 0,\ j = 1, ..., p. \tag{2}$$

A point $\mathbf{x}_0$ is called *feasible* if it satisfies all constraints, namely it is $h_i(\mathbf{x}_0) = 0,\ i = 1, ..., m$ and $g_j(\mathbf{x}_0) \leq 0,\ j = 1, ..., p$.

An inequality constraint $g_j(\cdot)$ is called *active* at point $\mathbf{x_0}$ if it is satisfied with equality, namely it is $g_j(\mathbf{x}_0) = 0$, otherwise it is called *inactive*.

Let $\mathcal{J}(\mathbf{x}_0)$ be set of indices of inequality constraints that are active at point $\mathbf{x}_0$. A point $\mathbf{x}_0$ is called *regular point* of the constraints if the vectors $\nabla h_i(\mathbf{x}_0)$, for $i = 1, \ldots, m$ and $\nabla g_j(\mathbf{x}_0)$(for $j \in \mathcal{J}(\mathbf{x}_0)$ are linearly independent.

3 First-order Necessary Conditions for existence of local minimum

Define the Lagrangian function at point $\mathbf{x}$, as

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x}) \tag{3}$$

where λ_i, $i = 1, \ldots, m$ are the Lagrange multipliers corresponding to the equality constraints and μ_j, $j = 1, \ldots, p$ are the Karush-Kuhn-Tucker (KKT) multipliers corresponding to inequality constraints $g_j(\mathbf{x}) \leq 0$.

3.1 Karush-Kuhn-Tucker (KKT) theorem

Suppose that $\mathbf{x}^*$ is a regular point of constraints $h_i(\mathbf{x}) = 0$, $i = 1, ..., m$ and $g_j(\mathbf{x}) \leq 0$, $j = 1, ..., p$. If $\mathbf{x}^*$ is a local minimum of $f(\mathbf{x})$ subject to the constraints $h_i(\mathbf{x}) = 0$, $i = 1, ..., m$ and $g_j(\mathbf{x}) \leq 0$, $j = 1, ..., p$, then there exist vectors $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \geq \mathbf{0}$, $\mu \epsilon R^p$ such that:

- $$\nabla L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \tag{4}$$
- $$\sum_{i=1}^{p} \mu_j g_j(\mathbf{x}^*) = 0 \tag{5}$$

Note that in the second equation above, since $\mu_j \geq 0$ and $g(\mathbf{x}^*) \leq 0$, the fact that $\sum_{i=1}^{p} \mu_j g_j(\mathbf{x}^*) = 0$ means that each term of the summation, $\mu_j g_j(\mathbf{x}^*) = 0$. This further means the following: if the j-th KKT multiplier is $\mu_j > 0$, then the corresponding constraint $g_j(\mathbf{x}^*) = 0$, i.e it is met with equality at the optimal solution $\mathbf{x}^*$. Also, if a constraint is satisfied with strict inequality at the optimal solution, i.e, $g_j(\mathbf{x}^*) < 0$ then the corresponding KKT multiplier $\mu_j = 0$. These conditions are reminiscent of complementary slackness ones we saw at Linear Programming.

Example: Consider the problem

$$\min \ 2x_1^2 + 2x_1 x_2 + x_2^2 - 10x_1 - 10x_2 \tag{6}$$

subject to:

$$x_1^2 + x_2^2 \leq 5, \text{ and } 3x_1 + x_2 \leq 6 \tag{7}$$

The Lagrangian is:

$$L(x_1, x_2, \mu_1, \mu_2) = 2x_1^2 + 2x_1 x_2 + x_2^2 - 10x_1 - 10x_2 + \mu_1(x_1^2 + x_2^2 - 5) + \mu_2((3x_1 + x_2 - 6). \tag{8}$$

From the KKT theorem, we have that if (x_1^*, x_2^*) is local minimum, then

$$\frac{\partial L(\cdot)}{\partial x_1}(x_1^*, x_2^*) = 0 \tag{9}$$

$$\frac{\partial L(\cdot)}{\partial x_2}(x_1^*, x_2^*) = 0. \tag{10}$$

So we get the equalities:

$4x_1 + 2x_2 - 10 + 2\mu_1 x_1 + 3\mu_2 = 0$

$2x_1 + 2x_2 - 10 + 2\mu_1 x_2 + \mu_2 = 0.$

Furthermore it is $\mu_1 \geq 0$, $\mu_2 \geq 0$ and also we have the conditions:

$$\mu_1(x_1^2 + x_2^2 - 5) = 0 \tag{11}$$

$$\mu_2(3x_1 + x_2 - 6) = 0 \tag{12}$$

Then if we want to find a point that satifies the necessary conditions of KKT theorem, we should start by taking cases and try various combinations of active constraints and chack signs of the resulting KKT multipliers.

Assume that $\mu_2 = 0 \rightsquigarrow 3x_1 + x_2 - 6 < 0$ and that $\mu_1 > 0 \rightsquigarrow x_1^2 + x_2^2 - 5 = 0$.

Then, for the conditions of the partial derivatives, we replace with $\mu_2 = 0$ and we have:

$$4x_1 + 2x_2 - 10 + 2\mu_1 x_1 = 0, \text{ and } 2x_1 + 2x_2 - 10 + 2\mu_1 x_2 = 0. \tag{13}$$

Also, since $\mu_1 > 0$, the first constraint holds with equality, $x_1^2 + x_2^2 - 5 = 0$.

We soolve the system of equations and find $x_1^* = 1$, $x_2^* = 2$, $\mu_1^* = 1$ and this gives $3x_1^* + x_2^* - 6 = -1 < 0$ and thus the second constraint is satified. Similarly, we can try other cases:

$\mu_1 > 0, \mu_2 > 0,$

$\mu_1 = 0, \mu_2 > 0$

$\mu_1 = 0, \mu_2 = 0.$

3.2 Graphical interpretation of KKT theorem

Consider the problem of minimizing function $\mathbf{x}$ subject to three inequality constraints $g_1(\mathbf{x}) \leq 0$, $g_2(\mathbf{x}) \leq 0$ and $g_3(\mathbf{x}) \leq 0$. Each of the inequality constraints defines a subspace that is located on the one side of the curve $g_j(\mathbf{x}) = 0$.

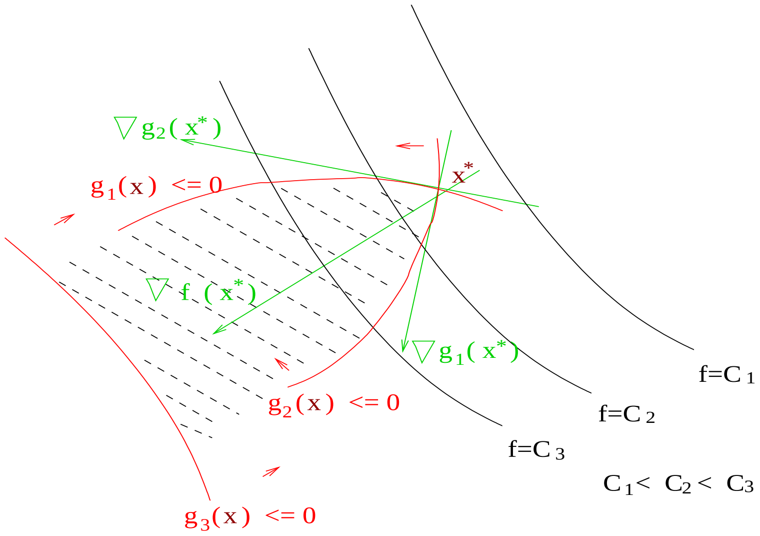


Figure 1: Graphical representation of the KKT theorem.

Assume that at the optimal point, $\mathbf{x}^*$, we have that the active constraints are the first and the second, i.e it is $g_1(\mathbf{x}^*) = 0$ and $g_2(\mathbf{x}^*) = 0$, while the third one is inactive, $g_3(\mathbf{x}^*) < 0$. This case is depicted in figure 3.1.

The KKT theorem states that if $\mathbf{x}^*$ is local minimum of $f(\cdot)$, subject to the constraints $g_j(\cdot) \leq 0$, $j = 1, 2, 3$, then:

$$\nabla f(\mathbf{x}^*) + \mu_1 \nabla g_1(\mathbf{x}^*) + \mu_2 \nabla g_2(\mathbf{x}^*) + \mu_3 \nabla g_3(\mathbf{x}^*) = \mathbf{0} \tag{14}$$

Since $g_3(\mathbf{x}^*) < 0$ (inactive) $\longrightarrow \mu_3 = 0$ and we have

$$\nabla f(\mathbf{x}^*) = -\mu_1 \nabla g_1(\mathbf{x}^*) - \mu_2 \nabla g_2(\mathbf{x}^*) \tag{15}$$

that is, the gradient of f at point $\mathbf{x}^*$ is linear combination of the gradients of the active constraints at $\mathbf{x}^*$.

3.3 Other forms of optimization problems

We saw that when we have the minimization problem

$$\min f(\mathbf{x}) \tag{16}$$

subject to: $h_i(\mathbf{x}) = 0,\ i = 1, ..., m$ and $g_j(\mathbf{x}) \leq 0, \quad j = 1, ..., p$ then we have the condition of the gradient of the Lagrangian being zero and also $\mu_j \geq 0$ for $j = 1, \ldots, p$.

What happens now if we have the problem:

$$\max f(\mathbf{x}) \tag{17}$$

subject to: $h_i(\mathbf{x}) = 0,\ i = 1, ..., m$ and $g_j(\mathbf{x}) \leq 0,\ \ j = 1, ..., p$.

Write the objective as $\max f(\mathbf{x}) = -\min f(\mathbf{x})$, subject to $h_i(\mathbf{x}) = 0,\ i = 1, ..., m$ and $g_j(\mathbf{x}) \leq 0,\ j = 1, ..., p$.

How will the KKT change? Let us apply the KKT theorem to the minimization problem of $-f(\mathbf{x})$.

If $\mathbf{x}^*$ is local maximum then

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla(-f(\mathbf{x}^*)) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \tag{18}$$

and also: $\boldsymbol{\mu} \geq \mathbf{0}$ and $\mu_j g_j(\mathbf{x}) = 0,\ \ \forall j$.

$\Rightarrow$ Multiplying with (-):

$$\nabla f(\mathbf{x}^*) - \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) - \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \Rightarrow \tag{19}$$

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*)) = \mathbf{0}, \tag{20}$$

with $\mu_j \leq 0$. Thus, if $\mathbf{x}^*$ is local maximum, then $\mathbf{x}^*$ satisfies $\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$ and now it should be $\boldsymbol{\mu} \leq \mathbf{0}$. The condition $\mu_j g_j(\mathbf{x}^*) = 0$ for each $j = 1, \ldots, p$ should also hold.

Now assume we have the problem: $\min f(\mathbf{x})$

subject to: $h_i(\mathbf{x}) = 0,\ i = 1, \ldots, m$ and $g_j(\mathbf{x}) \geq 0,\ j = 1, \ldots, p$, i.e the inequalities are now reversed. We multiply with -1, so as to bring the inequality in the usual form: $-g_j(\mathbf{x}) \leq 0$

Then, the KKT theorem is: If $\mathbf{x}^*$ local minimum of f, then:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j (-g_j(\mathbf{x}^*)) = \mathbf{0} \Rightarrow \tag{21}$$

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) - \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0},\ \ \mu_j \geq 0 \tag{22}$$

or

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0},\ \ \mu_j \leq 0 \tag{23}$$

There is also a fourth case that is treated similarly.

In conclusion, we have 4 cases:

i) $\min f(\mathbf{x}),\quad$ subject to: $h_i(\mathbf{x}) = 0,\quad i = 1, ..., m$

$g_j(\mathbf{x}) \leq 0,\quad j = 1, ..., p.$

ii) $\max f(\mathbf{x}),\quad$ subject to: $h_i(\mathbf{x}) = 0,\quad i = 1, ..., m$

$g_j(\mathbf{x}) \leq 0,\quad j = 1, ..., p.$

iii) $\min f(\mathbf{x}),\quad$ subject to: $h_i(\mathbf{x}) = 0,\quad i = 1, ..., m$

$g_j(\mathbf{x}) \geq 0,\quad j = 1, ..., p.$

iv) $\max f(\mathbf{x}),\quad$ subject to: $h_i(\mathbf{x}) = 0,\quad i = 1, ..., m$

$g_j(\mathbf{x}) \geq 0,\quad j = 1, ..., p.$

If we write the KKT condition at point $\mathbf{x}^*$, for all 4 cases we get that:

$$\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \tag{24}$$

and $\mu_j g_j(\mathbf{x}^*) = 0$ for each $j = 1, \ldots, p$. For each of the 4 cases, we have:

i) $\mu_j \geq 0,\quad j = 1, ..., p.$

ii) $\mu_j \leq 0,\quad j = 1, ..., p.$

iii) $\mu_j \leq 0,\quad j = 1, ..., p.$

iv) $\mu_j \geq 0,\quad j = 1, ..., p.$

Example: Which are the conditions of KKT theorem for the problem below:

$$\min f(\mathbf{x}) \qquad \text{subject to:} \quad \mathbf{x} \geq \mathbf{0}.$$

Solution

$$\nabla f(\mathbf{x}^*) \geq \mathbf{0}$$

$$\mathbf{x}^* \geq \mathbf{0} \text{ and}$$

$$\sum_i x_i \frac{\partial f(x^*)}{\partial x_i} = 0. \tag{25}$$

4 Second-order necessary conditions for existence of local minimum

If $\mathbf{x}^*$ is a regular point of the constraints:

$h_i(\mathbf{x}^*) = 0,\quad i = 1, ..., m$

$g_j(\mathbf{x}^*) \leq 0,\quad j = 1, ..., p$

and if $\mathbf{x}^*$ is local minimum of f subject to the constraints above, then $\exists\, \boldsymbol{\lambda} \in \mathbb{R}^m,\ \boldsymbol{\mu} \geq \mathbf{0}\ (\in \mathbb{R}_+^p)$ such that:

i) $\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$

ii) $\mu_j g_j(\mathbf{x}^*) = 0,\ j = 1, \ldots, p$

iii) The Hessian matrix of the Lagrangian function,

$$\Lambda(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = F(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i H_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j G_j(\mathbf{x}^*) \tag{26}$$

where $F(\mathbf{x}^*)$: Hessian matrix of $f(\mathbf{x})$ at $\mathbf{x}^*$,

$H_i(\mathbf{x}^*)$: Hessian matrix of $h_i(\mathbf{x})$ at $\mathbf{x}^*$,

$G_j(\mathbf{x}^*)$: Hessian matrix of $g_j(\mathbf{x})$ at $\mathbf{x}^*$

is positive semi-definite on the tangent subspace of the active constraints at $\mathbf{x}^*$.

Advanced Topics on Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 24 : Sufficient conditions for NLP problems, water-filling - 12/12/06

Notes by Eleni Galanou and Despina Koutsagia

1 Second Order Sufficient Conditions for existence of local minimum

Consider the problem:

$$\min f(\mathbf{x})$$

$$\text{s.t. } h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m,$$
$$g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, p.$$

If $\exists \;\; \boldsymbol{\lambda} \in R^m,\ \boldsymbol{\mu} \in R^p_+,\ \boldsymbol{\mu} \geq 0$ such that:

I.

$$\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \tag{1}$$

II. $\mu_j g_j(\mathbf{x}^*) = 0, \quad j = 1, \ldots, p,$

III. The Hessian matrix of the Lagrangian function,

$$\Lambda(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i H_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j G_j(\mathbf{x}^*) > 0, \tag{2}$$

is positive-definite on the subspace

$$\mathcal{M} = \{\mathbf{y} : \nabla h_i(\mathbf{x}^*)^T \mathbf{y} = 0, \nabla g_j(\mathbf{x}^*)^T \mathbf{y} = 0 \text{ for } j \in \mathcal{J}(\mathbf{x}^*)\}, \tag{3}$$

with $\mathcal{J}(\mathbf{x}^*) = \{j : g_j(\mathbf{x}^*) = 0, \mu_j > 0\}$ the set of active constraints at point $\mathbf{x}^*$. Thus, $\mathcal{M}$ is the subspace that is tangent level to the surface of the active constraints

then $\mathbf{x}^*$ is local optimum of function f.

Remark: Note that in our problems, unless otherwise specified, we will always assume that $\mathcal{M} = \mathbb{R}^m$ and thus we will not worry about finding an explicit characterization of $\mathcal{M}$. However, we will need to show that matrix $\Lambda(\cdot)$ is positive-definite, i.e for all $\mathbf{y} \in \mathbb{R}^m$ it is $\mathbf{y}^T \Lambda \mathbf{y} > 0$.

2 Sensitivity analysis

Consider the problem:

$$\min f(\mathbf{x})$$

$$\text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

Note that we have collectively described all equality constraints and all inequality constraints with two vectors $\mathbf{h}(\cdot)$ and $\mathbf{g}(\cdot)$ respectively. Now, assume we increase the right-hand side of the constraints (resources) as follows:

$$\mathbf{h}(\mathbf{x}) = \mathbf{c}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{d}$$

Let $\mathbf{x}(\mathbf{0}, \mathbf{0})$ to be the optimal solution to the initial problem and let $\mathbf{x}(\mathbf{c}, \mathbf{d})$ be the solution of the problem formed after we increased the right-hand sides of constraints. Then by following a similar reasoning as the one in the sensitivity analysis for problems with equality constraints, we have:

$$\nabla_{\mathbf{c}} f(\mathbf{x}, (\mathbf{c}, \mathbf{d})) = -\boldsymbol{\lambda}$$

$$\nabla_{\mathbf{d}} f(\mathbf{x}, (\mathbf{c}, \mathbf{d})) = -\boldsymbol{\mu}$$

and for the i-th Lagrange multiplier we have:

$$\lambda_i = -\frac{\partial f(\mathbf{x}(\mathbf{c},\mathbf{d}))}{\partial c_i}\Bigg|_{(\mathbf{c},\mathbf{d})=(\mathbf{0},\mathbf{0})}$$

namely λ_i is the rate of change of the objective function with respect to a unit of change in the i-th equality constraint, i.e it is the derivative of the cost function with respect to the quantity c_i that the i-th equality constraint changes.

For the j-th KKT multiplier we have:

$$\mu_j = -\frac{\partial f(\mathbf{x}(\mathbf{c},\mathbf{d}))}{\partial d_j}\Bigg|_{(\mathbf{c},\mathbf{d})=(\mathbf{0},\mathbf{0})}$$

namely μ_j is the rate of change of the objective function with respect to a unit of change in the j-th inequality constraint, i.e it is the derivative of the cost function with respect to the quantity d_j that the j-th inequality constraint changes.

Thus, λ_i, μ_j can be interpreted as price per unit of the corresponding resource that is described by the i-th equality or the j-th inequality constraint.

Problem: Consider the problem

$$\min f(x_1, x_2) = (x1 - 1)^2 + x_2 - 2$$

$$\text{subject to: } h(\mathbf{x}) = x_2 - x_1 - 1 = 0 \quad (\lambda)$$

$$g(\mathbf{x}) = x_1 + x_2 - 2 \leq 0 \quad (\mu \geq 0)$$

We define one Lagrange multiplier λ for the equality constraint and one KKT multiplier μ for the inequality constraint. We define the Lagrangian function and we consider:

$$\nabla L(x_1^*, x_2^*, \lambda, \mu) = \mathbf{0} \Rightarrow \begin{cases} \frac{\partial L(\cdot)}{\partial x_1} = 0 \\ \frac{\partial L(\cdot)}{\partial x_2} = 0 \end{cases}$$

and also we have the condition:

$\mu(x_1 + x_2 - 2) = 0$

$\mu \geq 0$

We have the following two cases:

a) $\mu > 0 \Rightarrow x_1 + x_2 - 2 = 0$

Thus, using the 4 equations we compute $x_1^* = 1/2, x_2^* = 3/2, \lambda^* = -1, \mu^* = 0$ and we confirm whether $\mu > 0$ is satisfied. Since $\mu = 0$ we arrive at paradox so this is not the case, and we proceed to the next case.

b) $\mu = 0 \Rightarrow x_1 + x_2 - 2 < 0$

Thus, using the 3 equations we compute $x_1^* = 1/2, x_2^* = 3/2, \lambda^* = -1$ and we confirm whether (x_1^*, x_2^*) feasible. It turns out that this is the case, and thus (x_1^*, x_2^*) is the optimal solution to the problem.

3 Second-order Sufficient Conditions for Convex Functions

Consider the problem:

$$\min f(\mathbf{x})$$

subject to:

$$h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$$
$$g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, p,$$

with functions f, h_i, g_j convex, $i = 1, \ldots, m$ and $j = 1, \ldots, p$. The second-order sufficient conditions for existence of minimum in this case are as follows:

If $\exists \quad \mathbf{x}^*, \boldsymbol{\lambda} \in R^m, \ \boldsymbol{\mu} \in R^p_+. \ \boldsymbol{\mu} \geq 0$ such that:

I.

$$\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{p} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \tag{4}$$

II. $\mu_j g_j(\mathbf{x}^*) = 0, \quad j = 1, \ldots, p$

then $\mathbf{x}^*$ is global minimum of function f subject to the constraints.

4 The water-filling algorithm

As an example of an optimization problem with equality and inequality constraints, we will consider a problem that arises in various resource allocation instances. Consider a transmitter that has at its disposal N orthogonal channels for transmission. The transmitter has a total amount of power P, assume that $P = 1$ without loss of generality.

The transmission rate (capacity) C_i for a channel i is given by $C_i = \log_2(1 + \frac{P_i}{N_i})$, where P_i is transmission power assigned to channel i and N_i is the noise power (noise variance) of channel i, $i = 1, \ldots, N$.

The objective is to allocate (split) the available power across channels so as to maximize total achieved capacity in all channels. This problem arises in OFDM systems in which the transmitter transmits in parallel using N sub-carrier frequencies. Also, the capacity can be viewed as a special case of a utility function $U(\cdot)$. The utility function $U(x)$ measures the amount of satisfaction of a user or consumer if an amount x of good (power, bandwidth, etc) is allocated to it. Here, obviously the resource is the power and the capacity is the derived utility. The problem can be formulated as follows:

$$\max \sum_{i=1}^{N} \log_2 \left(1 + \frac{P_i}{N_i}\right)$$

$$\text{subject to: } \sum_{i=1}^{N} P_i = 1,\ \ P_i \geq 0, i = 1, \ldots, N. \tag{5}$$

Note that in the most general case of utility function, we have the optimization problem of distributing a total amount of good W across users or consumers so as to maximize total utility

$$\max \sum_{i=1}^{N} U_i(x_i)$$

subject to

$$\sum_{i=1}^{N} x_i = W, \text{ and } x_i \geq 0, \tag{6}$$

where $U_i(\cdot)$ is the utility function of user i. Now, the capacity maximization problem we are dealing with can be written equivalently as

$$\min - \sum_{i=1}^{N} \log \left(1 + \frac{P_i}{N_i}\right)$$

subject to

$$\sum_{i=1}^{N} P_i = 1 \tag{7}$$

and

$$-P_i \leq 0,\ \ i = 1, \ldots, N \tag{8}$$

Define a Lagrange multiplier λ for the equality constraint and a KKT multiplier $\mu_i \geq 0$ for each inequality constraint $i = 1, \ldots, N$. Note also that in the above formulation we have omitted for simplicity the base 2 of the logarithm.

The initial problem is actually case 4 of the four cases we considered in the previous lecture, while the transformed one is of the form of case 1. Both are equivalent. The Lagrangian function is,

$$\begin{aligned} L(\mathbf{P}, \lambda, \boldsymbol{\mu}) &= - \sum_{i=1}^{N} \log \left(1 + \frac{P_i}{N_i}\right) + \lambda \left(\sum_{i=1}^{N} P_i - 1\right) + \sum_{i=1}^{N} \mu_i(-P_i) \\ &= - \sum_{i=1}^{N} \log \left(1 + \frac{P_i}{N_i}\right) + \lambda \left(\sum_{i=1}^{N} P_i - 1\right) - \sum_{i=1}^{N} \mu_i P_i \end{aligned}$$

We apply KKT conditions to solve the problem:

I.

$$\nabla_{\mathbf{P}} L(\mathbf{P}, \lambda, \boldsymbol{\mu}) = \mathbf{0} \Rightarrow \frac{\partial L(\mathbf{P}, \lambda, \boldsymbol{\mu})}{\partial P_i} = 0 \Rightarrow -\frac{1}{1+\dfrac{P_i}{N_i}} \frac{1}{N_i} + \lambda - \mu_i = 0 \; \forall\, i \tag{9}$$

II.

$$\mu_i P_i = 0, \; \forall\, i \tag{10}$$

and we also have: $P_i \geq 0$, $\sum_{i=1}^{N} P_i = 1$ and $\mu_i \geq 0$, $i = 1, \ldots, N$.

We solve (9) for μ_i:

$$\mu_i = \lambda - \frac{1}{P_i + N_i} \tag{11}$$

and substitute in $\mu_i P_i = 0$ to get:

$$\left(\lambda - \frac{1}{P_i + N_i}\right) P_i = 0 \tag{12}$$

We distinguish three cases:

i) $P_i > 0$ and $\left(\lambda - \frac{1}{P_i+N_i}\right) P_i = 0$.Then,

$$\lambda = \frac{1}{P_i + N_i} \Rightarrow P_i = \frac{1}{\lambda} - N_i \tag{13}$$

However, since we do not know λ, we do not know the sign of $\frac{1}{\lambda} - N_i$, and we can say that if $\frac{1}{\lambda} > N_i \Rightarrow P_i = \frac{1}{\lambda} - N_i$, since it has to be $P_i \geq 0$.

ii) $\left(\lambda - \frac{1}{P_i+N_i}\right) > 0$ and $P_i = 0$. Then,

$$P_i = 0 \text{ if } \frac{1}{\lambda} < N_i \tag{14}$$

iii) $\frac{1}{\lambda} = N_i$. Then,

$$\left(\frac{1}{N_i} - \frac{1}{P_i + N_i}\right) P_i = 0 \Rightarrow \tag{15}$$

$$\frac{P_i + N_i - N_i}{(P_i + N_i) N_i} P_i = 0 \Rightarrow P_i = 0 \tag{16}$$

Thus in conclusion we have:

$$P_i^* = \begin{cases} \frac{1}{\lambda^*} - N_i, & \text{if } \dfrac{1}{\lambda^*} > N_i \\ 0, & \text{if } \frac{1}{\lambda^*} \leq N_i \end{cases}$$

or equivalently,

$$P_i^* = \left(\frac{1}{\lambda^*} - N_i\right)^+ = \max\left(0, \frac{1}{\lambda^*} - N_i\right) \text{ with } \quad x^+ = \begin{cases} x & \text{, if } x > 0 \\ 0 & \text{, if } x \leq 0 \end{cases}$$

From the form of the solution above, we can see that the quantity $\frac{1}{\lambda^*}$ is common for all channels and resembles a kind of "water-level". Note that the better quality the channel is (the smaller the noise power), the more power is allocated to it. Also, the more noisy the channel, the less the power that is allocated to it. If the channel is "too noisy", i.e the noise power exceeds a certain power, then it is better from a capacity point of view not to allocate any power in the channel.

The result of water-filling can be seen in figure 1. In order to compute λ, one could argue that the constraint $\sum_{i=1}^{N} P_i^* = 1$ could be used, or $\sum_{i=1}^{N} \left(\frac{1}{\lambda^*} - N_i\right)^+ = 1$. However, it is not possible to solve the equation with regard to λ analytically, since we do not know in advance whether in different channels i the quantity $\frac{1}{\lambda^*} - N_i$ will be positive or negative.

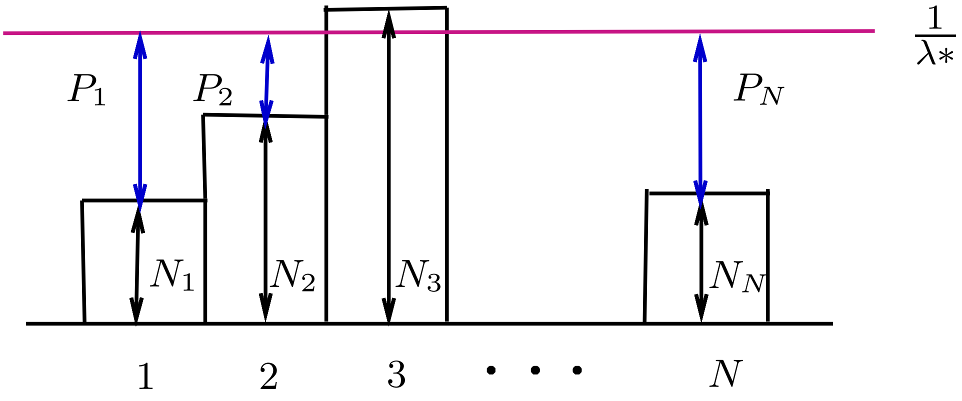


Figure 1: Water-filling algorithm with different amount of power P_i allocated to each channel i.

Instead, we use a simple algorithm. Each column represents a channel and the height of each column reflects the noise. There are N channels that are differentiated due to different noise level. Also the quantity $\frac{1}{\lambda^*}$ as we said is common for all channels.

We divide available power $P = 1$ into small quantities ϵ. We start allocating power in small quantities ϵ to the channel i_1 with the best quality (the less noise power N_{i_1}) until we reach the level of a channel i_2 of the second best quality, i.e second smallest N. From that point, we assign power ϵ to each of channels i_1, i_2 until we reach channel i_3 with the third best quality (third smallest N). Then we allocate power ϵ to these three channels. We continue in that fashion until we exhaust the available power. The point where we exhaust the power defines the final water level $\frac{1}{\lambda^*}$. The procedure is shown in figure 2.

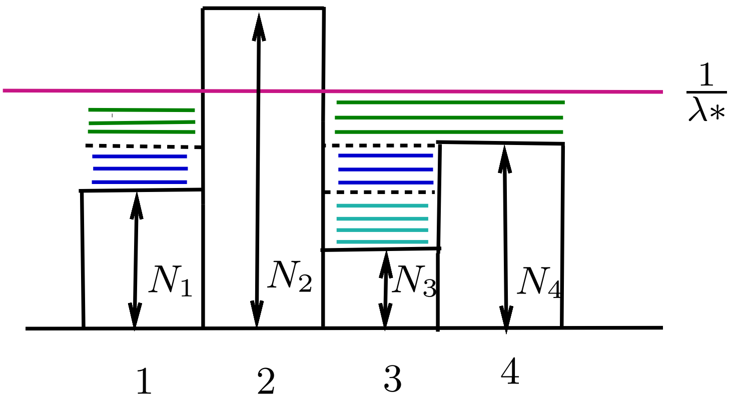


Figure 2: Successive power allocation in water-filling.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 25 : Lagrangian Duality - 18/12/06

Notes by : Charalampos Daskalakis and Constantinos Houmas

1 The Lagrangian dual function

1.1 The Lagrangian

We consider an optimization problem (P) in the form:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ s.t. \quad & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \\ & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, m \end{aligned} \tag{1}$$

$$\Omega = \left\{ \begin{array}{ll} \mathbf{x} : & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \\ & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, m \end{array} \right\}$$

with variable $\mathbf{x} \in \Omega$.

In Lagrangian duality, we start by writing the Lagrangian:

$$L(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x}) \tag{2}$$

We refer to λ_i as the *KKT multiplier* associated with the ith inequality constraint $f_i(\mathbf{x}) \leq 0$; similarly we refer to μ_i as the Lagrange multiplier associated with the ith equality constraint $h_i(\mathbf{x}) = 0$. Vectors λ and μ are called the *dual variable* vectors associated with the problem (1). We may call both kinds of variables Lagrangian variables.

1.2 The Lagrangian dual function

We define the *Lagrange dual function* l_D as the minimum value of the Lagrangian over x. That is, for $\lambda \in \mathbf{R}^m, \mu \in \mathbf{R}^p$,

$$l_D(\lambda, \mu) = \min_{\mathbf{x} \in \Omega} L(\mathbf{x}, \lambda, \mu) = \min_{\mathbf{x} \in \Omega}\{f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x})\} \tag{3}$$

1.3 Lower bounds on optimal value

The Lagrangian dual function yields lower bounds on the optimal value $f(\mathbf{x}^*)$ of the problem (1): For any $\lambda \geq 0$ and any μ we have

$$l_D(\lambda, \mu) \leq f(\mathbf{x}^*). \tag{4}$$

Suppose $\mathbf{x}_0$ is a feasible point for the problem (1), i.e., $h_i(\mathbf{x}) = 0$, $g_j(\mathbf{x}) \leq 0$, and $\lambda \geq 0$. Then we have

$$\sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}_0) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x}_0) \leq 0 \tag{5}$$

since each term in the first sum is non-positive, and each term in the second sum is zero, and therefore

$$L(\mathbf{x}_0, \lambda, \mu) = f(\mathbf{x}_0) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}_0) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x}_0) \leq f(\mathbf{x}_0) \tag{6}$$

Note that in the second sum, we have that each term is zero, since $\mu_j \geq 0$ and $g_j(\mathbf{x}_0) \leq 0$. Thus,

$$l_D(\lambda, \mu) = \min_{\mathbf{x} \in \Omega} L(\mathbf{x}, \lambda, \mu) \leq L(\mathbf{x}_0, \lambda, \mu) \leq f(\mathbf{x}_0) \tag{7}$$

So $\forall$ $\mathbf{x}_0$ which is feasible, $l_D(\lambda, \mu) \leq f(\mathbf{x}_0) \leq f(\mathbf{x}^*)$, and (4) holds.

2 Lagrangian dual problem

We saw from inequality (4), that we can get a lower bound from l_D on the optimal value of the objective function for each pair $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ with $\boldsymbol{\mu} \geq \mathbf{0}$. The next natural question is to find the *best* lower bound that can be obtained by the Lagrangian dual function. This brings us to the formulation of the Lagrangian dual problem (LD), which is given as:

$$\begin{aligned} \max \quad & l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ s.t. \quad & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned} \tag{8}$$

There are two main reasons why we prefer to solve (LD) problem instead of the original one (P). The first reason is because it may be easier to solve (LD) since it has fewer constraints, and second and most important, because $l_D(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is always concave, independently of the original (P). This last claim can be shown by the following line of thoughts:

Consider functions of one variable for simplicity. First we need to show that if $f_1(x), f_2(x)$ are concave functions, then $f(x) = \min\{f_1(x), f_2(x)\}$ is also concave. We know that a function is concave when: $f(\vartheta x + (1-\vartheta)y) \geq \vartheta f(x) + (1-\vartheta)f(y)$.

In our case:

$$\begin{aligned} f(\vartheta x + (1-\vartheta)y) = \quad & \min\{f_1(\vartheta x + (1-\vartheta)y), f_2(\vartheta x + (1-\vartheta)y)\} \;\geq \\ & \min\{\vartheta f_1(x) + (1-\vartheta)f_1(y), \;\; \vartheta f_2(x) + (1-\vartheta)f_2(y)\} \;\geq \\ & \vartheta \min\{f_1(x), f_2(x)\} \;+\; (1-\vartheta)\min\{f_1(y), f_2(y)\} = \\ & \vartheta f(x) + (1-\vartheta)f(y) \end{aligned}$$

The above result can be extended for more functions $f_1(x), f_2(x), \ldots, f_n(x)$ and is also valid for convex functions $f_1(\cdot), f_2(\cdot)$, if min is substituted by max.

So from the (LD) problem we see that $l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}}\{$ linear functions of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}\}$. Since we know that all linear functions can be considered to be concave, the proof is completed.

2.1 Week Duality

The optimal value of the Lagrangian dual problem (LD), which we denote $d^* = l_D(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, is, by definition, the best lower bound on $p^* = f(\mathbf{x}^*)$, which is the optimal value of primal problem (P). In particular, we have the important inequality:

$$d^* \leq p^*. \tag{9}$$

This property is called *Weak Duality Lemma* and can be shown as follows:

$$f(\mathbf{x}) \geq p^* = \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \geq \min_{\mathbf{x}\in\Omega}\{f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x})\} = l_D(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

$\Rightarrow f(\mathbf{x}) \geq p^* \geq l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \;\forall \boldsymbol{\lambda}, \boldsymbol{\mu}$

$\Rightarrow f(\mathbf{x}) \geq p^* \geq l_D(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = d^*$.

The weak duality inequality (9) holds even if d^* and p^* are infinite. If the primal problem (P) is unbounded from below, so that $p^* = -\infty$, we must have $d^* = -\infty$, *i.e.*, the Lagrange dual problem is infeasible. Conversely, if the dual problem (LD) is unbounded from above, so that $d^* = \infty$, we must have $p^* = \infty$, *i.e.*, the primal problem is infeasible.

We refer to the difference $p^* - d^*$ as the *optimal duality gap* of the original problem, since it gives the gap between the optimal value of the primal problem and the best (*i.e.*, greatest lower bound on it that can be obtained from the Lagrangian dual function. The optimal duality gap is always nonnegative.

2.2 Strong Duality

If the equality

$$d^* = p^* \tag{10}$$

holds, *i.e.*, the optimal duality gap is zero, then we say that *strong duality* holds. This means that the best bound that can be obtained from the Lagrange dual function is tight.

Strong duality does not, in general, hold. But if the primal problem (P) is convex, *i.e.*, of the form

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ s.t. \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, m \\ & A\mathbf{x} = \mathbf{b} \end{aligned} \tag{11}$$

with functions $f(\cdot), g_1(\cdot), \ldots, g_m(\cdot)$ convex, we usually (but not always) have strong duality. There are many results that establish conditions on the problem, beyond convexity, under which strong duality holds. These conditions are called constraint qualifications.

One such simple constraint qualification is *Slater's condition*: The condition says: if there exists $\mathbf{x} \in \Omega$ such that

$$g_j(\mathbf{x}) \leq 0, \qquad j = 1, \ldots, m, \qquad A\mathbf{x} = \mathbf{b} \tag{12}$$

then we have strong duality.

3 Solving primal problem (P) using Lagrangian dual (LD)

We transform our primal problem (P) in its Lagrangian dual form

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{13}$$

In order to find $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$ we start from an arbitrary $\boldsymbol{\lambda}_0$ and $\boldsymbol{\mu}_0$ and use the gradient ascent method:

$$\boldsymbol{\lambda}^{(t+1)} = \boldsymbol{\lambda}^{(t)} + \alpha \nabla_{\boldsymbol{\lambda}} l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{14}$$

and

$$\boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^{(t)} + \alpha \nabla_{\boldsymbol{\mu}} l_D(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{15}$$

If $\boldsymbol{\mu} < \mathbf{0}$ somewhere, then we substitute with $\mathbf{0}$. In case $l_D(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is not differentiable, we use the so called super-gradient method.

Vector $\boldsymbol{\lambda}$ is called super-gradient of function f at $\mathbf{x}_0$ if and only if:

$$f(\mathbf{x}) - f(\mathbf{x}_0) \leq \boldsymbol{\lambda}^T(\mathbf{x} - \mathbf{x}_0). \tag{16}$$

These topics are considered advanced and we will not elaborate in them more.

4 Saddle-point interpretation

In this section we give several interpretations of Lagrangian duality.

4.1 Max-min characterization of weak and strong duality

To simplify the discussion we assume there are no equality constraints. The results are easily extended to cover them. First note that

$$\max_{\boldsymbol{\mu} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\mu}) = \max_{\boldsymbol{\mu} \geq \mathbf{0}} \left(f(\mathbf{x}) + \sum_{j=1}^{p} \mu_j g_j(\mathbf{x}) \right) \tag{17}$$

We can express the optimal value of the primal problem as

$$p^* = \min_{\mathbf{x}} \max_{\boldsymbol{\mu} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\mu}). \tag{18}$$

By definition of the dual function, we also have

$$d^* = \max_{\boldsymbol{\mu} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}). \tag{19}$$

Thus, weak duality can be expressed as the inequality:

$$\max_{\boldsymbol{\mu} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}) \leq \min_{\mathbf{x}} \max_{\boldsymbol{\mu} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\mu}) \tag{20}$$

and strong duality as the equality:

$$\max_{\boldsymbol{\mu} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}) = \min_{\mathbf{x}} \max_{\boldsymbol{\mu} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\mu}) \tag{21}$$

Strong duality means that the order of the minimization over $\mathbf{x}$ and the maximization over $\boldsymbol{\mu} \geq \mathbf{0}$ can be switched without affecting the result.

In fact, the inequality (20) does not depend on any properties of function $L(\cdot)$, and therefore we have:

$$\max_{\mathbf{z}\in\mathcal{Z}} \min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w},\mathbf{z}) \leq \min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{z}\in\mathcal{Z}} f(\mathbf{w},\mathbf{z}) \tag{22}$$

for any $f : \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}$ (and any $\mathcal{W} \subseteq \mathbf{R}^n$ and $\mathcal{Z} \subseteq \mathbf{R}^m$). This general inequality is called the *max-min inequality*. When equality holds, *i.e.*,

$$\max_{\mathbf{z}\in\mathcal{Z}} \min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w},\mathbf{z}) = \min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{z}\in\mathcal{Z}} f(\mathbf{w},\mathbf{z}) \tag{23}$$

we say that f (and $\mathcal{W}$ and $\mathcal{Z}$) satisfy the *strong max-min property* or the *saddle-point property.*

We refer to a pair $\mathbf{w}_0 \in \mathcal{W}, \mathbf{z}_0 \in \mathcal{Z}$ as a *saddle-point* of function $f(\cdot)$ (and $\mathcal{W}$ and $\mathcal{Z}$) if

$$f(\mathbf{w}_0,\mathbf{z}) \leq f(\mathbf{w}_0,\mathbf{z}_0) \leq f(\mathbf{w},\mathbf{z}_0) \tag{24}$$

for all $\mathbf{w} \in \mathcal{W}$ and $\mathbf{z} \in \mathcal{Z}$.

4.2 Game theory interpretation of saddle point

We can interpret *max-min* inequality (20) the *max-min* equality (21) and the *saddle-point* property in terms of a *zero-sum game* with two players, where each players has a continuous set of strategies. If the first player (P1) chooses $\mathbf{w} \in \mathcal{W}$, and the second player (P2) selects $\mathbf{z} \in \mathcal{Z}$, then P1 pays an amount $f(\mathbf{w},\mathbf{z})$ to P2. P1 therefore wants to minimize f, while P2 wants to maximize f. In this case there are certain connections between the saddle-point of $f(\cdot)$ and the so-called *Nash equilibrium point* of the game. If we are at the Nash equilibrium, no player can do better by deviating his strategy from that defined by the equilibrium.

Advanced Topics in Networking - Fall 2006

Instructor : Iordanis Koutsopoulos

Lecture 26 : Decomposition theory and an example from pricing theory - 19/12/06

Notes by : Eleni Anagnostopoulou and Nena Xanthopoulou

1 Lecture outline

- Decomposition theory
- Example : Network pricing

Consider a (wired) network, represented as a directed graph $G = (S, A)$ where S is the set of network nodes that belong at the network ($|S|$ is the number of nodes) and A is the set of links of the network ($|A|$ is the number of links). The capacity (in bits/sec) of link $l \in A$ is defined by c_l and shows the maximum number of bits per second that can be carried over the link. We define the variable x_s (in bits/sec) as the information generation rate at every node (source) of the network as well as the utility function $U_s(x_s)$, which varies for every node, and is concave function of x_s. Note that every node in the network can potentially be the information source (and thus belong in set S). The utility function $U_s(x_s)$ shows the amount of satisfaction derived by node s if it is allowed to transmit with rate x_s. The utility function is a concave function of x_s for each node and each node may have a different utility function.

We have seen in previous lectures the physical meaning of a function being concave. It means that the rate of satisfaction with regard to change in x_s is a decreasing function of x_s. That is, the user (node) is satisfied with a higher rate for small values of x_s and this satisfaction rate decreases as we assign more resources x_s to it. The capacity function $c(P) \sim \log(P)$ as a function of the allocated amount of power p is a special case of utility function.

Each node could be considered either as:

Source : the more information rate it sends to other nodes of the network, the more it is satisfied, or

Destination : the more information rate it receives, the more it is satisfied.

Suppose now a network consisting of many nodes. Our goal is to maximize the total utility of all nodes in the network,

$$\max_{\mathbf{x}} \sum_{s=1}^{S} U_s(x_s)$$

by appropriately controlling the information generation rate x_s. The vector of variables $\mathbf{x}$ is $\mathbf{x} = (x_1, x_2, \ldots, x_{|s|})$.

Suppose source $s \in S$ uses the set of links $L(s)$ in order to transfer the information it produces. Define for each link $l \in A$ as $S(l)$ to be the set of sources that use link l, i.e that transfer their information through that link. This makes obvious the first, and essentially the only constraint of our problem which is the fact that capacity of each link should not be exceeded,

$$\sum_{s:l\in L(s)} x_s \leq c_l, \; l = 1, \ldots, |A|$$

Every source sends its information through specific paths. The above problem is an optimization problem with inequality constraints, one for each link. In order to find the solution, we use the Lagrangian function. So we define coefficient $\lambda_l \geq 0$ as the KKT multipliers that correspond to link l, $l = 1, \ldots, |A|$ and we have:

$$\begin{aligned}
L(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_{s=1}^{|S|} U_s(x_s) + \sum_{l=1}^{|A|} \lambda_l (c_l - \sum_{s:l\in L(s)} x_s) \\
&= \sum_{s=1}^{|S|} U_s(x_s) + \sum_{l=1}^{|A|} \lambda_l c_l - \sum_{l=1}^{|A|} \lambda_l \sum_{s:l\in L(s)} x_s \\
&= \sum_{s=1}^{|S|} U_s(x_s) - \sum_{s=1}^{|S|} \sum_{l=1}^{|A|} x_s \lambda_l + \sum_{l=1}^{|A|} \lambda_l c_l \\
&= \sum_{s=1}^{|S|} \left[U_s(x_s) - x_s \sum_{l\in L(s)} \lambda_l \right] + \sum_{l=1}^{|A|} \lambda_l c_l
\end{aligned}$$

Suppose that the values of λ_l are known. It is possible for the global problem to be solved by each source individually, so that each source s finds the optimal rate as

$$x_s^* = \arg\max_{x_s} \left[U_s(x_s) - x_s \sum_{l\in L(s)} \lambda_l \right] \tag{1}$$

Hence, the initial global objective is decomposed in that way into separate optimization problems, one for each source. If each source s finds the optimal rate x_s^*, then we have collectively the optimal solution $\mathbf{x}^*$ for the problem.

Now define the Lagrangian dual problem,

$$l_D(\boldsymbol{\lambda}) = \max_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{s=1}^{|S|} \max_{x_s} \left[U_s(x_s) - x_s \sum_{l \in L(s)} \lambda_s \right]$$

The value λ_l is the price paid by the source s that uses link l for each unit of flow that it sends through link l. As we see, the initial maximization decomposes into $|S|$ apart maximization problems, one for each source. Each source s solves a separate problem of maximizing the net benefit, i.e the derived utility minus the total cost of using the links in set $L(s)$,

$$\max_{x_s} \left[U_s(x_s) - x_s \sum_{l \in L(s)} \lambda_l \right]$$

Each source s can compute the optimal solution, x_s^* as the root of the equation

$$\frac{dU(x_s)}{dx_s} - \sum_{l \in L(s)} \lambda_l = 0 \Rightarrow x_s^*(\boldsymbol{\lambda})$$

for a given price vector $\boldsymbol{\lambda}$.

The Lagrangian dual problem is:

$$\min_{\boldsymbol{\lambda} \geq \mathbf{0}} l_D(\boldsymbol{\lambda}) = \min_{\boldsymbol{\lambda} \geq \mathbf{0}} \left[\sum_{s=1}^{|S|} U_s(x_s^*) - x_s^*(\boldsymbol{\lambda}) \sum_{l \in L(s)} \lambda_l \right] \tag{2}$$

In the problem setup, there is a central agent (e.g the network price controller that is ran by the network operator) that finds the price of using each link depending on its use and popularity. Thus, it computes different prices λ_l for each link l. Then, it adapts the price using the following intuitive rule : whenever a link is over-used its price has to be increased so as to discourage users from using it and thus reduce the link load. On the other hand, when a link is under-utilized, the price has to be reduced so as to make it more attractive to users to transfer their information through this link. Then, given the certain computed link price vector $\boldsymbol{\lambda}$, each source solves the separate maximization problem and finds the amount of traffic x_s^* to send through each link so as to maximize the net utility. The values of x_s are then sent to the central unit, which solves the Lagrangian dual problem in order to recalculate the prices.

Since the minimization problem of the Lagrangian dual cannot be solved analytically, the central entity can perform one iteration of the gradient descent algorithm to update the link prices. Thus, the price for link l is updated as follows:

$$\lambda_l(t+1) = \lambda_l(t) - a\frac{\partial l_D(\boldsymbol{\lambda})}{\lambda_l} \Rightarrow \lambda_l(t+1) = \max\left\{\lambda_l(t) - a(c_l - \sum_{s\in S(l)} x_s^*(\boldsymbol{\lambda}(t))), 0\right\}$$

where a is the step size for the gradient descent algorithm and we have taken care so that λ_l does not take negative values. Note that the equation above arose since

$$\frac{\partial l_D}{\partial \lambda_l} = c_l - \sum_{s:l\in L(s)} x_s^*(\boldsymbol{\lambda}(t)). \tag{3}$$

It is possible that this sum could overcome the value of c_l and then the central unit must increase the value of price $\lambda_l(t+1)$ at the next iteration. In contrast, when a link is not used very much, the value of $\lambda_l(t+1)$ decreases in order to make that link attractive and used by more sources.

The Algorithm that takes place is as follows:

1. Start with initial prices $\lambda_l(0)$, for $l = 1, \dots, |A|$.

2. Each source s solves, independently from the other sources, the separate maximization problem

   $$\max_{x_s}\left[U_s(x_s) - x_s \sum_{l\in L(s)} \lambda_l\right]$$

   and finds the optimal rate $x_s^*(\lambda)$. Each source sends the optimal values $x_s^*(\lambda)$ to the central coordinating agent.

3. The central agent updates the price for using each link as follows:

   $$\lambda_l(t+1) = \lambda_l(t) - a\frac{\partial l_D(\boldsymbol{\lambda})}{\lambda_l} \Rightarrow \lambda_l(t+1) = \max\left\{\lambda_l(t) - a(c_l - \sum_{s\in S(l)} x_s^*(\boldsymbol{\lambda})(t)), 0\right\}$$

   and broadcasts the new link prices to all sources.

4. $t \leftarrow t+1$. Go to 1. Continue until convergence.

The algorithm above can be shown to converge to the optimal rate vector $\mathbf{x}^* = (x_1^*, \dots, x_{|S|}^*)$, such that the total utility is maximized.

Note: We could view the price update mechanism as a form of congestion control for the reasons explained above.