Computational Game Theory	Spring Semester, 2005/6

Lecture 3: Price of Anarchy (PoA): Routing

Lecturer: Yishay Mansour *Scribe: Maya Ben-Ari, Lotem Kaplan*[1]

3.1 Lecture Overview

In this lecture we consider the problem of routing traffic to optimize the performance of a congested and unregulated network. We are given a network, a rate of traffic between each pair of nodes and a latency function specifying the time needed to traverse each edge given its congestion. The goal is to route traffic while minimizing the *total latency*. In many situations, network traffic cannot be regulated, thus each user minimizes his latency by choosing among the available paths with respect to the congestion caused by other users. We will see that this "selfish" behavior does not perform as well as an optimized regulated network.
We investigate the *price of anarchy* by exploring characteristics of *Nash Equilibrium* and *minimal latency optimal flow*.
We prove that if the latency of each edge is a linear function, then the PoA is at most 4/3, while in unsplitable routing the PoA is bounded by 8/3. We also show that if the latency function is only known to be continuous, nondecreasing and differentiable, then there is no bounded coordination ratio.

3.2 Introduction

Last lecture we observed the problem of Job Scheduling (or Parallel Lines Routing) where each player wants to send a particular amount of traffic along a path from source to destination, and has to choose exactly one line to pass his traffic along.

Today, we shall investigate the problem of routing traffic in a network. The problem is defined as follows: Given a rate of traffic between pairs of nodes in the network, find an assignment of the traffic to paths so that the total latency is minimized. Each link in the network is associated with a latency function which is typically load-dependent, i.e. the latency increases as the link becomes more congested.
In many domains (such as the internet or road networks) it is impossible to impose regulation of traffic, and therefore we are interested in those settings where each user acts according to his own selfish interests. We assume that each user will always select the minimum latency path to its destination. In other words, we assume all users are rational and nonmalicious. This can actually be viewed as a noncooperative game where each user plays the best response given the state of all other users, and

[1]This scribe is based in part on the scribe notes of Anat Axelrod, Eran Werner 2004

thus we expect the chosen routes to form a Nash equilibrium.

3.2.1 Motivation for the Model

Each edge in the network is assigned a latency function, that specifies the delay of the edge as a function of the congestion on that edge.

- **The Player Model**
  - Many users, where each user holds only a negligible portion of the total traffic.
  - A finite number of users that are allowed to split their load between different paths.
- **The Global Target Function** is to minimize the average (or total) latency suffered by all users.

Let *Nash* denote the maximum latency among all feasible flows that are Nash Equilibrium (NE).
Let OPT denote the minimum latency among all feasible flows.
The *price of anarchy* (PoA) is defined as the ratio $PoA = \frac{Nash}{OPT}$.
Our goal is to bound the PoA.

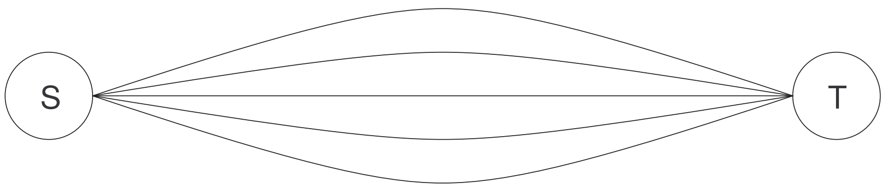

Figure 3.1: Routing on Parallel Lines

Example: Routing on Parallel Lines We use nearly the same model as in the last lecture (see Figure 3.1): a set of n players with weights $w_i,\ i = 1, ..., n$ $(w_i > 0)$. m lines (machines) with speeds $s_i,\ i = 1, ..., m$.
Our goal is to minimize the congestion on the lines. The players are allowed split their flow between different lines.

Nash Equilibrium is achieved when the Load on each line is $L_i(a) = \frac{\sum_{i=1}^{n} w_i}{\sum_{j=1}^{m} s_j} = \frac{W}{S}$ (this is NE because if there is a line with more than $\frac{W}{S}$ then there is a line with less than $\frac{W}{S}$ so any player using the more loaded line will benefit from passing flow to the less loaded line).

The *Optimum* is achieved (as seen in the last lecture) by dividing the flow equally between the lines. Therefore we achieve $PoA = 1$.

Before we continue, let's examine an example setting which has inspired much of the work in this traffic model. Consider the network in Figure 3.2(a). There are

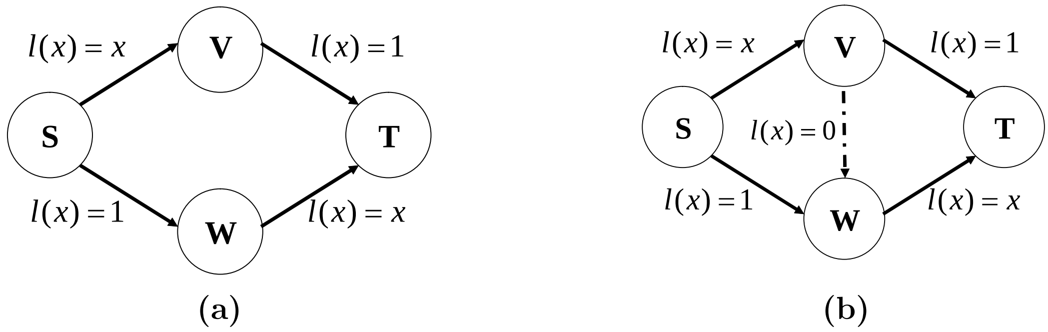

Figure 3.2:

two disjoint paths from S to T. Each path follows exactly two edges. The latency functions are labeled on the edges. Suppose one unit of traffic needs to be routed from S to T. The optimal flow coincides with the Nash equilibrium such that half of the traffic takes the upper path and the other half takes the lower path. In this manner, the latency perceived by each user is $\frac{3}{2}$. In any other (unequal) distribution of traffic among the two paths, there will be a difference in the total latency of the two paths and users will be motivated to reroute to the less congested path.

Note Incidentally, we will soon realize that in any scenario in which the flow at Nash Equilibrium is split over more than a single path, the latency of all the chosen paths must be equal.

Now, consider Figure 3.2(b) where a fifth edge of latency zero is added to the network. While the optimum flow has not been affected by this augmentation and stays $\frac{3}{2}$, Nash will only occur by routing the entire traffic on the single S $\rightarrow$ V $\rightarrow$ W $\rightarrow$ T path, hereby increasing the latency each user experiences to 2 (because if we split the flow to the upper and lower paths, then the user will be motivated to reroute to the less congested path, using the new edge. However, if the entire traffic is routed trough S $\rightarrow$ V $\rightarrow$ W $\rightarrow$ T no user will benefit from a change, and therefore this is a Nash Equilibrium).
Amazingly, adding a new zero latency link had a negative effect for all agents. This counter-intuitive impact is known as *Braess's paradox.*

Anecdote 1 *Two live and well known examples of Braess's paradox occurred when 42nd street was closed in New York City and instead of the predicted traffic gridlock, traffic flow actually improved. In the second case, traffic flow worsened when a new road was constructed in Stuttgart, Germany, and only improved after the road was closed.*

3.2.2 Formal Definition of the Problem

The problem of routing flow in a network with flow dependent latencies is defined as follows:

- We consider a directed graph $G = (V, E)$.
- Input
  - k pairs of source and destination vertices (s_i, t_i).
  - Demand r_i (the amount of required flow between s_i and t_i).
    We may assume that $r_i > 0$.
  - Each edge $e \in E$ is given a load-dependent latency function denoted by $\ell_e(\cdot)$. We restrict our discussion to nonnegative, differentiable and nondecreasing latency functions.
- Output
  Flow f - A function that defines for each path p a flow f_p.
  f induces *flow on edge* e, $f_e = \sum_{p:e\in p} f_p$ (a flow on an edge is the sum of flows of all the paths that contains that edge).
- We denote the set of simple paths connecting the pair (s_i, t_i) by $\mathcal{P}_i$.
  And let $\mathcal{P} = \bigcup_i \mathcal{P}_i$.
- A solution is *feasible* if $\forall i$, $\sum_{p\in\mathcal{P}_i} f_p = r_i$ (for all i - the sum of the flow over all paths between s_i to t_i is equal to the demand r_i).
- The latency of a path ℓ_p is defined as the sum of latencies of all edges in the path. $\ell_p(f) = \sum_{e\in p} \ell_e(f_e)$.
- The *total cost* of a flow f, $C(f) \triangleq \sum_{p\in\mathcal{P}} \ell_p(f) \cdot f_p = \sum_{e\in E} \ell_e(f_e) \cdot f_e$ (the two formulas express the same value, since they differ only in the summation order).
- (G, r, ℓ) - A triple which defines an *instance* of the routing problem.
- Our goal is to find a feasible flow f that will minimize the total cost $C(f) = \sum_e \ell_e(f_e) \cdot f_e$.

We denote this problem by *GAME*.

3.3 Characterizations of Nash & OPT Flows

3.3.1 Flows at Nash Equilibrium

Lemma 3.3.1 *A feasible flow f for instance (G, r, ℓ) is Nash Equilibrium if for every $i \in \{1, ..., k\}$ and $p_1 \in \mathcal{P}_i$*

- *if* $f_{p_1} > 0$ *then* $\ell_{p_1}(f) = \min_{p\in\mathcal{P}_i} \ell_p(f)$ *(equivalently* $\forall p_2 \in \mathcal{P}_i,\ \ell_{p_1}(f) \leq \ell_{p_2}(f)$*)*
- *if* $f_{p_1} = 0$ *then* $\ell_{p_1}(f) \geq \min_{p\in\mathcal{P}_i} \ell_p(f)$

From the lemma it follows that flow at Nash Equilibrium will be routed only through best response (BR) paths. Consequently, all paths assigned with a positive flow between (s_i, t_i) have equal latency denoted by $L_i(f)$. Namely, $L_i(f) \triangleq \min_{p\in\mathcal{P}_i} \ell_p(f)$.

Corollary 3.1 *If f is a flow at a Nash Equilibrium for instance (G, r, ℓ) then $\forall_i,\ C(f) = \sum_{i=1}^{k} L_i(f) \cdot r_i$.*

3.3.2 Optimal Solution - Flow

Our goal is to find the optimal solution, that is to find a feasible flow f that will minimize the total cost $C(f) = \sum_{e\in E} \ell_e(f_e) f_e$.

Observation 3.2 *The following (possibly non-linear) program solves the minimum latency flow problem:*

Let $c_e(x) = \ell_e(x) \cdot x$, *and then* $C(f) = \sum_{e\in E} c_e(f_e)$

$$\min \sum_{e\in E} c_e(f_e)$$

$$s.t.$$

$$\text{(SYSTEM)} \qquad \begin{array}{ll} \sum_{p\in\mathcal{P}_i} f_p = r_i & \forall i \in \{1, ...k\} \\ f_e = \sum_{p\in\mathcal{P}: e\in p} f_p & \forall e \in E \\ f_p \geq 0 & \forall p \in \mathcal{P} \end{array}$$

The optimal solution for (SYSTEM) is the same as the optimal solution for (GAME), therefore we will refer to them both as (GAME).

Note For simplicity the above formulation of (SYSTEM) is given with an exponential number of variables (there can be an exponential number of paths). This formulation can be easily modified with decision variables only on edges giving a polynomial number of variables and constraints.

The program (SYSTEM) has linear constrains, however, it's objective function may be too general to allow an efficient algorithm for optimal solution. We therefore consider a restricted case of, called *convex programming.*

Convex Programming
Let $F(x)$ be a convex function, and S a convex set.
A convex Programming is of the form: $\min F(x),\ s.t.\ x \in S$.

Lemma 3.3.2 *If $F(\cdot)$ is strictly convex then the solution is unique.*

Proof. Assume that $x \neq y$ are both minimum solutions. Let $z = \frac{1}{2}x + \frac{1}{2}y$. Because S is a convex set, $z \in S$.
Since $F(\cdot)$ is strictly convex: $F(z) < \frac{1}{2}F(x) + \frac{1}{2}F(y)$, contradicting the minimality of $F(x)$ and $F(y)$. □

Lemma 3.3.3 *If $F(\cdot)$ is convex then the solution set U is convex.*

Lemma 3.3.4 *If $F(\cdot)$ is convex and y is not optimal ($\exists x : F(x) < F(y)$) then y is not a local minimum. Consequently, any local minimum is also a global minimum.*

Proof. Assume that y is not optimal, i.e. $\exists x : F(x) < F(y)$. Let $Z = \lambda x + (1-\lambda)y$. Since $F(\cdot)$ is convex $F(z) \leq \lambda F(x) + (1-\lambda)F(y) < F(y)$, for every $0 < \lambda < 1$.
□

Note Lemma 3.3.4 implies that the "gradient method" converges to an optimal solution in convex programming.

In our case of network routing, we assume that for each edge $e \in E$ the function $c_e(x) = \ell_e(x) \cdot x$ is a convex function, and therefore, our target function $C(f)$ is also convex.

Our assumption on $\ell_e(x)$ implies that $c_e(x)$ is differentiable for every x.

Let $c'_e(x) = \frac{d}{dx}c_e(x)$
Let $c'_p(x) = \sum_{e \in p} c'_e(x)$

Lemma 3.3.5 *(The optimality condition) A flow f is optimal for (GAME) iff*

$$\forall p_1, p_2 \in \mathcal{P}_i,\ f_{p_1} > 0 \Rightarrow\ c'_{p_1}(f) \leq c'_{p_2}(f)$$

Notice the resemblance between the characterization of optimality conditions (Lemma 3.3.5), and Nash Equilibrium (Lemma 3.3.1). In fact, an optimal flow can be interpreted as a Nash equilibrium with respect to a different edge latency functions. We will use this resemblance to reach the bound on PoA.

Let

$$\begin{aligned} \ell^*_e(x) &\triangleq c'_e(x) = (\ell_e(x) \cdot x)' = \ell_e(x) + x \cdot \ell'_e(x) \\ \ell^*_p(x) &\triangleq \textstyle\sum_{e \in p} \ell^*_e(x) \end{aligned}$$

Corollary 3.3 *Flow f is an optimal flow for (G, r, ℓ) iff f is a Nash Equilibrium for the instance (G, r, ℓ^*).*

Proof. Flow f is OPT for $\ell \Leftrightarrow$ (optimallity conditions) $\forall p_1, p_2 \forall f_{p_1} > 0,\ c'_{p_1}(f) \leq c'_{p_2}(f) \Leftrightarrow$ (by def.) $\forall p_1, p_2 \forall f_{p_1} > 0,\ \ell^*_{p_1}(f) \leq \ell^*_{p_2}(f) \Leftrightarrow f$ is Nash Eq. for ℓ^*
$(\forall i\ \forall p_1, p_2 \in \mathcal{P}_i)$. □

3.3.3 Existence of Flows at Nash Equilibrium

We exploit the similarity between the characterizations of Nash and OPT flows to establish that a Nash equilibrium indeed exists and its cost is unique.
For the outline of the proof we define an edge cost function $h_e(x) \triangleq \int_0^x \ell_e(t)dt$.
By definition $h'_e(x) = \frac{d}{dx}h_e(x) = \ell_e(x)$ thus h_e is differentiable with non decreasing derivative ℓ_e and therefore convex.
Next, we consider the following convex program:

$(GAME^*)\ \min \sum_{e \in E} h_e(f_e)\ s.t.\ f\ feasible$

Observation 3.4 *The optimal solution for $GAME^*$ is Nash $GAME$.*

Proof. The proof follows directly from Lemma 3.3.1 and the optimality condition in Lemma 3.3.5 where $\ell_e(x) = h'_e(x)$. □

Since Nash is an optimal solution for a different convex setting we conclude that:

- Nash equilibrium exists.
- The cost at Nash equilibrium is unique.
- The cost of all the paths used in a Nash equilibrium jave the same cost.

3.3.4 Bounding the Price of Anarchy

The relationship between Nash and OPT characterizations provide a general method for bounding the price of anarchy $PoA = \frac{C(f)}{C(f^*)} = \frac{Nash}{OPT}$.
$\ell_e(\cdot)$ is non-decreasing, therefore $h_e(x) = \int_0^x \ell_e(t)dt \leq x\ell_e(x) = c_e(x)$

Theorem 3.5 *If there exists a constant $\alpha > 0$ such that $\forall x,\ \alpha h_e(x) \geq c_e(x)$ then $PoA \leq \alpha$.*

Proof.

$$\begin{aligned} Nash = C(f) &= \sum_{e \in E} c_e(f_e) \\ &\leq \alpha \sum_{e \in E} h_e(f_e) \\ &\leq \alpha \sum_{e \in E} h_e(f_e^*) \\ &\leq \alpha \sum_{e \in E} c_e(f_e^*) \\ &= \alpha \cdot C(f^*) = \alpha \cdot OPT \end{aligned}$$

The first inequality follows from the hypothesis, the second follows from the fact that Nash flow f is optimal for the function $h_e(f_e)$ and the final inequality follows from $h_e(x) \leq c_e(x)$. □

Corollary 3.6 *If the latency function $\ell_e(\cdot)$ is a polynomial function of degree d, $\ell_e(x) = \sum_{i=0}^d a_{e,i}x^i$, then $PoA \leq d+1$.*

Note From the corollary, an immediate coordination ratio of 2 is established for linear latency functions. Later, we will show a tighter bound of $\frac{4}{3}$.

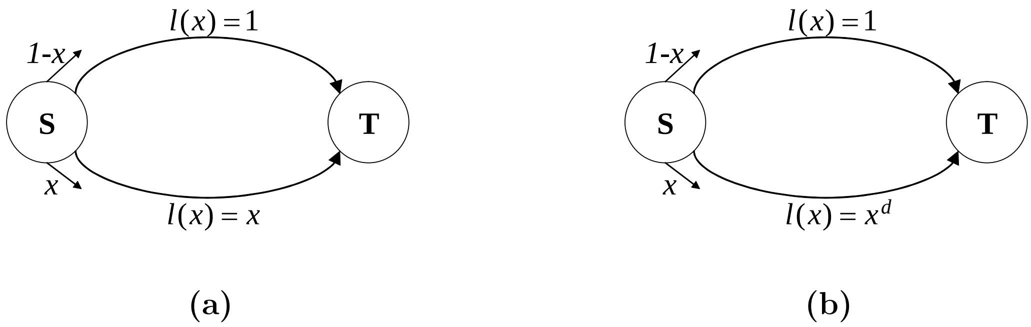

Figure 3.3: bounded and unbounded PoA

Figure 3.3(a) shows an example for which Nash flow will only traverse in the lower path, and reach the value 1, while OPT will divide the flow equally among the two paths. The target function is $1 \cdot (1-x) + x \cdot x$ and it reaches minimum with value $\frac{3}{4}$ when x = $\frac{1}{2}$, giving a coordination ratio of $\frac{4}{3}$ for this example. Combining the example with the tighter upper bound to be shown, we demonstrate a tight bound of $\frac{4}{3}$ for linear latency functions.

In Figure 3.3(b) the flow at Nash will continue to use only the lower path, with the value 1, but OPT will minimize the cost function $x \cdot x^d + (1-x) \cdot 1$. at $x = 1 - \frac{1}{\sqrt{d+1}}$ $OPT \leq (1 - \frac{1}{\sqrt{d+1}})^{d+1} + \frac{1}{\sqrt{d+1}} \leq e^{-\sqrt{d+1}} + \frac{1}{\sqrt{d+1}} \rightarrow_{d\rightarrow\infty} 0$. This is not the x that reaches OPT but it gives an upper bound. This implies that $\lim_{d\rightarrow\infty} PoA = \infty$ meaning, PoA cannot be bounded from above for any polynomial latency function, independent of the degree.

3.4 A Tight Bound for Linear Latency Functions

We will now focus on a scenario where all edge latency functions are linear $\ell_e(x) = a_e x + b_e$, for constants $a_e, b_e \geq 0$. A fairly natural example for such a model is a network employing a congestion control protocol such as TCP. We have already seen in Figure 3.3(a) an example where the coordination ratio was $\frac{4}{3}$. We have also established an upper bound of 2 according to Corollary 3.6. We shall now show that the $\frac{4}{3}$ ratio is also a tight upper bound.

Prior to this result, we examine a simple case where $\ell_e(x) = b_e$. In this case both OPT and Nash will route all the flow to the shortest paths. Thus, $Nash = OPT$.

Lemma 3.4.1

$$xy \leq x^2 + \frac{y^2}{4}$$

Proof. See Appendix. □

Theorem 3.7 *If the latency functions are all of the form $\ell_e(x) = a_e x + b_e$ then $PoA \leq \frac{4}{3}$.*

Proof. Let f be a flow at Nash equilibrium and f^* an optimal flow.
Given a flow f, we define $\ell_e^f = a_e f_e + b_e$, and $C^f(x) = \sum \ell_e^f x_e$.

$$\begin{aligned}
C^f(x) &= \sum (a_e f_e + b_e) x_e \\
&= \sum (a_e f_e x_e + b_e x_e) \\
&\leq \sum (a_e x_e^2 + b_e x_e) + \sum a_e f_e^2 \tfrac{1}{4} \quad \text{since } (xy \leq x^2 + \tfrac{y^2}{4}) \\
&\leq C(X) + \tfrac{1}{4} C(f)
\end{aligned}$$

Since $\forall x\ C^f(f) \leq C^f(x)$ we can set $x_e = f_e^*$ and derive,

$$\begin{aligned}
C^f(f) = C(f) &\leq C(f^*) + \tfrac{1}{4} C(f) \\
\tfrac{3}{4} C(f) &\leq C(f^*) \\
C(f) &\leq \tfrac{4}{3} C(f^*) \\
PoA &\leq \tfrac{4}{3}
\end{aligned}$$

□

3.5 Unsplitable Routing

Recall that each pair (s_i, t_i) has the demand r_i. In this case we have to route all the demand r_i on exactly one path $p_i \in \mathcal{P}_i$.
Let P_j be the the path assignment in Nash Equilibrium solution, and P_j^* the optimal solution.

If $\ell_e(x) = a_e x + b_e$, then from NE we obtain:

$$\sum_{e \in P_j} (a_e f_e + b_e) \leq \sum_{e \in P_j^*} [a_e (f_e + r_j) + b_e]$$

This inequality is correct $\forall j$, so we can multiply by r_j and sum up all inequalities:

$$\sum_j \sum_{e \in P_j} (a_e f_e + b_e) r_j \leq \sum_j \sum_{e \in P_j^*} (a_e (f_e + r_j) + b_e) r_j$$

Let $J(e)$, $J^*(e)$ be the set of flows that use edge e in the assignment P_j and P_j^* respectively. Changing the accumulation order:

$$\begin{aligned}\sum_e \sum_{j\in J(e)}(a_e f_e + b_e) r_j &\leq \sum_e \sum_{j\in J^*(e)}(a_e(f_e + r_j) + b_e) r_j \\ \sum_e \sum_{j\in J(e)}(a_e f_e + b_e) r_j &\leq \sum_e \sum_{j\in J^*(e)}[(a_e f_e + b_e) r_j + a_e r_j^2]\end{aligned}$$

note that $\sum_{j\in J(e)} r_j = f_j$, and $\sum_{j\in J^*(e)} r_j = f_j^*$ and therefore $\sum_{j\in J^*(e)} r_j^2 \leq f_j^{*2}$

$$\begin{aligned}\sum_e (a_e f_e + b_e) f_e &\leq \sum_e (a_e f_e + b_e) f_e^* + \sum_e a_e f_e^{*2} \\ \sum_e (a_e f_e + b_e) f_e &\leq \sum_e (a_e f_e^* + b_e) f_e^* + \sum_e a_e f_e f_e^*\end{aligned}$$

The left side is Nash and the first sum in the right side is OPT:

$$Nash \leq OPT + \sum_e a_e f_e f_e^*$$

Using $xy \leq x^2 + y^2 \cdot \frac{1}{4}$, we have $\sum_e a_e f_e f_e^* \leq \sum_e a_e f_e^{*2} + \sum_e a_e f_e^2 \cdot \frac{1}{4} \leq OPT + Nash \cdot \frac{1}{4}$. Therefore,

$$\begin{aligned}Nash &\leq OPT + OPT + \tfrac{1}{4} \cdot Nash \\ \tfrac{3}{4} Nash &\leq 2 \cdot OPT \\ Nash &\leq \tfrac{8}{3} \cdot OPT \\ PoA &\leq \tfrac{8}{3} \sim 2.66\end{aligned}$$

Note We can reach a bound of 2.61 by using Cauchy-Schwarz inequality instead of $xy \leq x^2 + y^2 \frac{1}{4}$.

3.6 FIN

All good things must come to an end.

3.7 APPENDIX A

Convex Set

Definition 3.7.1 *A set S is called a* convex set *if* $\forall A, B \in S$, $0 \leq \lambda \leq 1$, $\lambda A + (1 - \lambda) B \in S$.

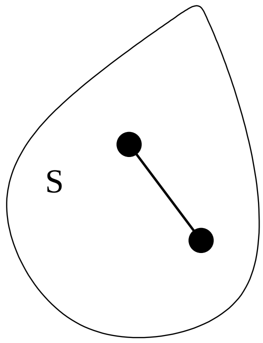

Figure 3.4: Convex Set

Intuitively, a set S is convex if the linear segment connecting two points in the set, is entirely in the set.
(see Figure 3.4)

Convex Function

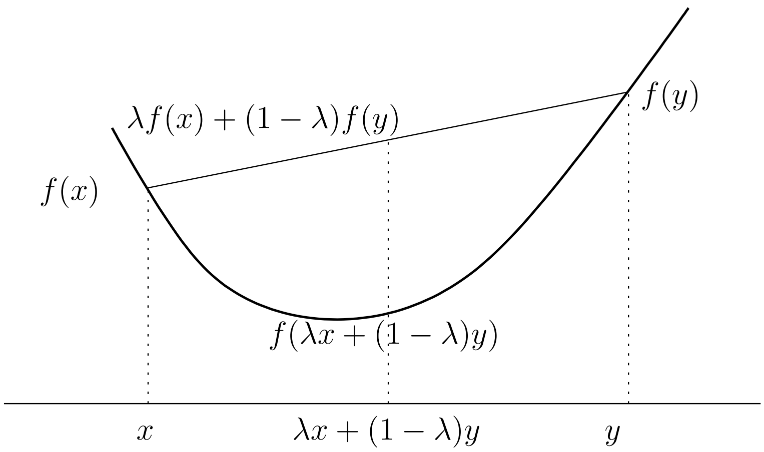

Figure 3.5: Convex Set

Definition 3.7.2 *Function f is called a* convex function *if* $\forall x, y$, $0 \leq \lambda \leq 1$, $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$.

(see Figure 3.5)

3.8 APPENDIX B

Lemma 3.8.1

$$xy \leq x^2 + \frac{y^2}{4}$$

Proof.

$$
\begin{aligned}
xy &\leq x^2 + \tfrac{y^2}{4} \\
4xy &\leq 4x^2 + y^2 \\
0 &\leq 4x^2 - 4xy + y^2 \\
0 &\leq (2x - y)^2
\end{aligned}
$$

$\square$