INTRODUCTION

Transportation, traffic, communication and energy networks form the backbone of our modern society. To deal with the uncertainty, variation, unpredictability, size and complexity inherent in these networks, we need to develop radically new ways of thinking. The ultimate goal is to build self-organising and intelligent networks. The NWO-funded Gravitation programme NETWORKS started in the Summer of 2014 and covers a broad range of topics dealing with stochastic and algorithmic aspects of networks.

In fall 2024 the fifth “NETWORKS goes to school” event was organised. The aim of the event is to provide secondary education students and teachers a first mathematical introduction on network science. This book collects the material realised for the “NETWORKS goes to school” event. This year the theme of the masterclass is Game Theory, a modern discipline which is very important in economics and mathematics. Techniques from mathematical modelling, probability theory, functions, networks, algorithms come together in this field where researchers try to understand how ”good” decisions can be made.

In Chapter 1, all the necessary background material that is required for Chapters 2 and 3 is presented. In Chapter 2, we introduce queueing theory by showing how to model and analyse a queue with mathematical techniques. In this chapter, we show how mathematical models of real life applications can be made. We show how concepts from game theory can be applied to make decisions about queues. Chapter 3 focuses on road traffic networks, and discusses how we can use concepts from game theory to optimize road traffic. Chapter 4 contains exercises on these two topics and in Chapter 5 we provide the corresponding solutions. Chapters 2, 3, 4, and 5 were written with the help of Artem Tsikiridis (Centrum Wiskunde en Informatica, Amsterdam) and Jiesen Wang (University of Amsterdam).

For more information and the booklets of the first four masterclasses “NETWORKS goes to school”, please visit https://onderwijs.networkpages.nl/masterclass/. This masterclass is sponsored by the Netherlands Organisation for Scientific Research (NWO) through the Gravitation grant ”NETWORKS”, and by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant no. 945045.

On behalf of the NETWORKS programme,

the organising committee of “NETWORKS goes to school”
Koen van den Berk (Eindhoven University of Technology)
Nicos Starreveld (University of Amsterdam)

Contents

Chapter 1

Preliminaries

1.1. Basic notation

We start by introducing some notation we will use in the sequel:

(1) $\mathbb{N}$ for the set of natural numbers, that is $\mathbb{N} = \{1, 2, 3, \cdots\}$;
(2) $\mathbb{Z}$ for the set of integer numbers, that is $\mathbb{Z} = \{\cdots, -3, -2, -1, 0, 1, 2, 3, \cdots\}$.
(3) $\mathbb{R}$ for the set of real numbers, that is all integer numbers and all the decimal numbers between them.
(4) We will use the symbol $\leq$ when we want to say "less or equal to". For example, $a \leq b$ means that a is less or equal to b.
(5) We will use the symbol $\approx$ when we want to say " almost equal to". For example. $\pi \approx 3.14159$.
(6) We will use the symbol $\lceil\rceil$ for the *ceiling* of a number, which is the largest integer number of that number. For example $\lceil 2.5 \rceil = 3$, and $\lceil 4.1 \rceil = 5$.
(7) The number of elements in some set A is denoted by $|A|$.
(8) We will use the notation *max* and *min* for the maximum and minimum value in some set or of some function. For example

$$\max_{i \in \{1,2,3\}} i^2 = 9 \qquad \text{and} \qquad \min_{j \in \{-1,0,1\}} (i+1) = 0.$$

(9) You are familiar with functions of one variable, like the function f given by $f(x) = x^2$. We will also see functions of functions. For instance, we know that the *y-intercept* (the place where a function meets the y-axis) of the function f is its value at $x = 0$, so $f(0)$. We could define the function y-intercept as

$$\text{y-intercept}(f) = f(0),$$

for any function f.

On sums

Mathematicians always want to write down mathematics as compactly as possible. But the notation used should also be clear and representative of what it describes. The typical notation you encounter when doing mathematics involves the symbol used for summation: $\sum$. Below we show how $\sum$ is used to describe a sum. Suppose you want to use the summation symbol to describe the sum $1 + 2 + 3 + 4 + 5 + 6$, then you can write this down compactly as

$$\sum_{k=1}^{6} k = 1 + 2 + 3 + 4 + 5 + 6 = 21. \tag{1.1.1}$$

The advantage of this notation is that you can write down large sums very compactly. For example, if you want the sum of the first 100 natural numbers, instead of only up to 6, then you can write this down as

$$\sum_{k=1}^{100} k. \tag{1.1.2}$$

By playing with the value at which the sum starts or ends you see that you can represent many sums or products using this notation. The general notation is the following:

$$\sum_{k=m}^{n} a_k, \tag{1.1.3}$$

where k is the index of summation; a_k are indexed variables representing each term of the sum; m is the lower bound of summation, and n is the upper bounds of summation.

In all the expressions above, either of summations or products, you can remark that the index k in every step increases by one, it starts from a number m, then takes the value $m + 1$, $m + 2$ until it reaches the number n. It is also possible to choose the indices from some set of values. Say for example that you want to compute the sum of the squares of all even numbers greater or equal to 4 and less or equal to 20. You can define the set of indexes you want to sum over, in this case

$$I = \{N \in \mathbb{N} : 4 \leq N \leq 20 \text{ and } N \text{ is even}\} = \{4, 6, 8, 10, 12, 14, 16, 18, 20\}.$$

Then the desired sum can be written as

$$\sum_{k \in I} k^2 = 16 + 36 + 64 + 100 + 144 + 296 + 324 + 400 = 1380.$$

This sum could also be written compactly as follows

$$\sum_{k \in I} k^2 = \sum_{4 \leq k \leq 20,\, k \text{ even}} k^2.$$

For two quantities x and y, we say x is a *lower bound* for y if $x \leq y$. Similarly, we say x is an *upper bound* for y if $x \geq y$.

1.2. Probability theory

Probability theory is the area of mathematics that studies random phenomena. For example if the experiment is tossing a coin, then there are two possible outcomes, either *heads* or *tails*. Each outcome occurs with probability 0.5. In order to study such a random experiment we use random variables.

Random variable

A **random variable** X is a variable whose possible values are outcomes of a random experiment. We will also use the term *stochastic* as a synonym for random.

We define a random variable by giving the *state space*, i.e. the set of all possible values the variable can take, and the *probability function*, which yields the corresponding probability that a given outcome will occur. For the coin toss for example we can define a random variable by assigning to the outcome *heads* the value 1 and to the outcome *tails* the value 0. In this case we have

$$X = \begin{cases} 1 & \text{if the outcome is heads,} \\ 0 & \text{if the outcome is tails.} \end{cases}$$

The probability function for this random variable is given by

$$\mathbb{P}(X = 1) = \mathbb{P}(\textit{heads}) = 0.5,$$

and

$$\mathbb{P}(X = 0) = \mathbb{P}(\textit{tails}) = 0.5,$$

where for a possible set of outcomes A, $\mathbb{P}(A)$ denotes the probability that A occurs. A random variable can be *discrete* or *continuous*.

Discrete random variables

A random variable X is called discrete when it can take countable many values, for simplicity we can just say that its values are the integer numbers, that is $X \in \mathbb{Z}$.

Continuous random variables

A random variable X is called continuous when it can take continuously many values, for simplicity we can just say that its values are the real numbers that is $X \in \mathbb{R}$.

For a discrete random variable, we can write down the probability that it equals a specific value. For a continuous random variable, this is not possible, as there is a continuum of possible values. We can however specify the probability that a continuous random variable falls in a range of values by using the **density function**. The probability that a continuous random variable X assumes values in the interval $[a, b]$ is given by the integral of the density function, denoted by f_X, over that interval:

$$\int_a^b f_X(x)\mathrm{d}x = \mathbb{P}(X \in [a, b]).$$

The result of this integration gives the area delimited by the graph of the density function f_X, the x-axis and the vertical lines given by $x = a$ and $x = b$.
Integrals: If you are not familiar with integration don't worry, you won't need it in Chapters 2 and 3. We present it here to give a complete picture and to show that they are very important in probability theory.

Expectation of a random variable

For a random variable X, discrete or continuous, we define the expectation, or expected value, as the average of all independent realisations of the random variable. We denote the expectation of X by $\mathbb{E}[X]$.

For a discrete random variable its expectation is defined by

$$\mathbb{E}[X] = \sum_{k=0}^{\infty} k\mathbb{P}(X = k). \tag{1.2.1}$$

For a continuous random variable its expectation is defined by

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x)\mathrm{d}x, \tag{1.2.2}$$

where f_X denotes the density function of the random variable, this means that

$$f_X(x)\mathrm{d}x = \mathbb{P}(X \in \mathrm{d}x). \tag{1.2.3}$$

As we will see in the sequel, if the random variable takes only positive values then the integral in the expectation starts from 0 instead of $-\infty$.

Bernoulli random variable

Bernoulli random variable

A **Bernoulli random variable** describes the outcome of any single random experiment that asks a yes-no question, like tossing a coin.

It takes the value 1 with probability p and the value 0 with probability $1-p$. Consider for example a coin where one side is heavier, then this is a biased coin where one side is favoured. We will use $B(p)$ to denote a Bernoulli random variable with probability p. A Bernoulli random variable has expectation given by

$$\mathbb{E}[B(p)] = 1 \cdot \mathbb{P}(B(p) = 1) + 0 \cdot \mathbb{P}(B(p) = 0) = p. \tag{1.2.4}$$

1.2.1. Geometric random variable

Geometric random variable

A **geometric random variable** describes the number of failures in a sequence of random experiments, each asking a yes-no question, until the first success.

We make the following assumptions:

- each observation is independent of the other observations;
- each observation represents one of two outcomes: success or failure;
- the probability p of success is exactly the same for each trial.

Under these assumptions, we can describe each geometric distribution by using the parameter p, we will denote a geometric random variable by $G(p)$. The geometric random variable has state space $\{0, 1, 2, \ldots\}$, and the probability that $G(p)$ is equal to k is given by

$$\mathbb{P}(G(p) = k) = (1-p)^k p.$$

When the random variable $G(p)$ is equal to k then we know that k failures have occurred before the first success. The probability of a failure is equal to $1-p$ and by the assumptions above the experiments we perform are independent of each other. The geometric random variable has expectation equal to

$$\mathbb{E}[G(p)] = \sum_{k=0}^{\infty} k\mathbb{P}(G(p) = k) = \sum_{k=0}^{\infty} k(1-p)^k p = \frac{1-p}{p}. \tag{1.2.5}$$

Again the exact derivation of the formula is far away from the scope of this booklet. For some more details on this formula we refer to Exercise 2.

EXAMPLE 1.2.1. Consider a coin toss, where possible outcomes are heads or tails. Suppose that we have an unfair coin, i.e., the probability for heads is $\frac{1}{3}$ and the probability for tails is $\frac{2}{3}$. Then the probability to get five times tails before the first heads is equal to

$$\mathbb{P}\left(G\left(\frac{1}{3}\right) = 5\right) = \left(\frac{2}{3}\right)^5 \frac{1}{3} \approx 0.044.$$

1.2.2. Exponential random variable

Exponential random variable

The **exponential random variable** is a continuous random variable and describes the time elapsed between events that occur continuously and independently at a constant intensity.

An exponential random variable is characterised by a parameter λ, called the intensity. The larger this parameter is the higher the frequency of the arriving events. A random variable having the exponential distribution with parameter λ, denoted by $E(\lambda)$, has the following probability distribution function

$$\mathbb{P}(E(\lambda) \leq x) = 1 - e^{-\lambda x}, \qquad x \geq 0, \tag{1.2.6}$$

and a probability density function given by

$$f_\lambda(x) = \lambda e^{-\lambda x}, \qquad x \geq 0. \tag{1.2.7}$$

The expectation of the exponential random variable is equal to

$$\mathbb{E}[E(\lambda)] = \int_0^\infty x f_{E(\lambda)}(x)\mathrm{d}x = \int_0^\infty x\lambda e^{-\lambda x}\mathrm{d}x = \frac{1}{\lambda}. \tag{1.2.8}$$

The exponential random variable has the *memoryless property*, i.e. that means that

$$\mathbb{P}(E(\lambda) > x + y | E(\lambda) > y) = \mathbb{P}(E(\lambda) > x), \qquad x, y \geq 0. \tag{1.2.9}$$

The probability on the left-hand side in the equation above is called a *conditional probability*. The symbol | is read as " given that", we will not need this concept during the masterclass but in general it is a very important concept in probability theory. For more details we refer to Exercise 1. This memoryless property is quite remarkable, so let's look at it from a practical side. Suppose the time until the bus arrives is exponentially distributed. If that would be the case, then if the bus didn't arrive for an hour, then it would still take the same amount of time until the bus arrives. But in reality we expect that if the bus didn't arrive for an hour, then it will probably arrive soon.

1.2.3. Poisson process

Finally, we introduce the Poisson process. This represents a sequence of events where events happen once every while. The time between events is exponentially distributed. Since the exponential distribution is memoryless, the Poisson process has a very remarkable property. If no event happened for a while, it doesn't imply that some event will occur soon. As an example, consider the time until you hit a specific number on a roulette wheel. If that specific number didn't show up for a while, that doesn't make it more likely for the number to show up sooner than normal. In other words: the history of the process has no influence on the future.

1.3. Graph theory

An intuitive definition of a network would be a 'collection of objects that are interconnected in some way'. Think for example of a collection of people, who can be interconnected by friendships; or a collection of cities, which can be interconnected by roads. To make this idea precise, we turn to graph theory.

Graph

An (undirected) **graph** is a pair $G = (V, E)$, where
- V is the set of nodes or vertices;
- E is the set of edges, connecting the nodes.

Typically, we number the nodes from $\{1, 2, 3, \ldots, \}$. We denote an edge between two nodes i and j by $\{i, j\}$. To define a graph, we can write down the sets V and E.

EXAMPLE 1.3.1. Consider

$$V = \{1, 2, 3, 4, 5, 6\}, \quad E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}.$$

Then $G = (V, E)$ is a graph with six nodes and seven edges.

It may be very useful to have a graphical representation of a graph. We do this by typically drawing nodes as a circle with a label in it, and edges as a line between nodes. However, you are free to choose any representation you may like! In fact, the location of the nodes is also arbitrary, it only matters the way in which the edges connect the nodes together.

EXAMPLE 1.3.1 (Continued). In Figure 1.3.1 we see two ways in which the graph G can be drawn.

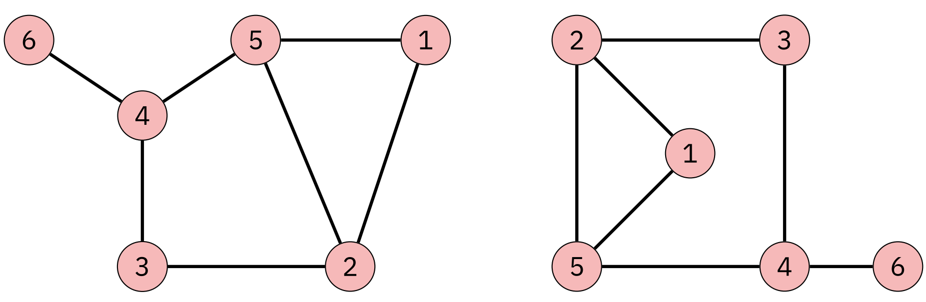


Figure 1.3.1. Two different representations of the graph in Example 1.3.1.

Degree of a node in a graph

The degree of a node v in a graph $G = (V, E)$, denoted by $d(v)$, is the number of neighbors of v. In the graph above for example the degree of node 1 is $d(1) = 2$, and of node 5 is $d(5) = 3$.

Path between two nodes in a graph

A path between two nodes in a graph, say v and w, is a sequence of edges which joins a sequence of nodes from v to w. In the graph in Figure 1.3.1 for example, the sequence $6 \to 4 \to 3 \to 2$ forms a path from 6 to 2. On the other side, the sequence $6 \to 4 \to 3 \to 1$ is not a path since $\{3, 1\}$ is not an edge in the graph. A shortest path between two nodes is a path using the least amount of edges. The shortest path from node 6 to node 1 for example has length 3.

Flows on networks

Undirected graphs, like those we have just seen, are useful in many situations. In other cases, it makes sense if the edges have a **direction**.

Directed graph

A **directed graph** is a pair $G = (V, A)$ where A is the set of *arcs*, which are directed edges. We denote an arc from node i to node j by (i, j).

To get an idea of when directed graphs can be useful, suppose we are given a sewage network from an industrial estate. We have depicted it in Figure 1.3.2. The nodes in this graph are the Companies, the intersections (I1, I2, I3) and the Reservoir. The arcs in this network are sewage pipes, represented by arrows. The sewage water should go from the companies to the reservoir, where the sewage water is purified. To harness the power of gravity, the pipes are typically slanted downwards, so the water flows in the right direction. Sometimes, intersections are at the same height, like I1 and I2, so water can flow both ways.

In reality, if the sewage system is overloaded or there is some congestion, water might flow in any direction. For the purpose of this example, we do not consider these scenarios. Today, Company 2 and Company 3 are not producting any waste water at all. Imagine Company 1 flushes 3 cubic metres of water into the sewage system every minute. We can describe the resulting *flow* as follows:

- For every arc $a \in A$, we define the *flow value* f_a, describing the amount of sewage that flows through this pipe every minute.
- At any node (except Company 1 and Reservoir) the amount of flow going in, should equal the amount flowing out.

Observe that in this way we obtain a flow along the path *Company 3 - I2 - I3 - Reservoir*. A flow along a path of this graph, starting at Company 3 and ending in Reservoir, is called a *Company 3 - Reservoir flow*. In general we will use the notation $s-t$-flow for a flow on a path in the graph starting from a *source* s and ending in a *sink* t. This ideas leads us to the general definition of an $s-t$-flow on a graph.

s-t flow

Let s and t be nodes in a directed graph $G = (V, A)$. Then an $s-t$ flow is a collection of values $f_a,\ a \in A$ that defines a flow value for every arc such that *flow conservation* holds. This means:

- For every node v (except s and t), the total flow value going into v equals the total flow value going out of v.

For simplicity, we assume s has only outgoing arcs, and t has only ingoing arcs. The *value* of an $s-t$ flow is the total flow going out of s.

You, the reader, are invited to write a flow value next to every arc in Figure 1.3.2 in such that these values constitute a valid $s-t$ flow of value 3.

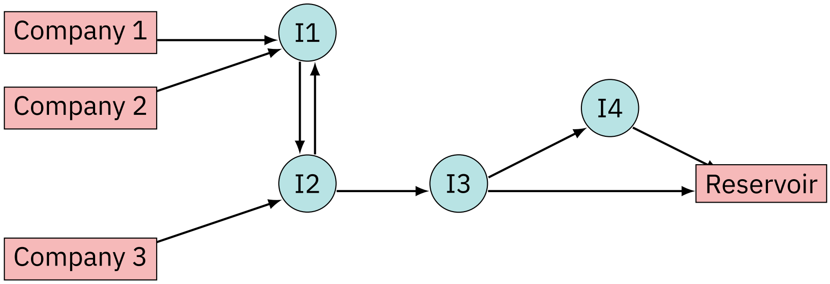


Figure 1.3.2. An imaginary sewer network as a directed graph

Flows usually become interesting when the arcs each have a certain *capacity*. Denote the capacity of an arc a by c_a. In Figure 1.3.3 we have drawn a flow from Company 1 to the Reservoir with value 3. The capacities are drawn in red, after the '/'. Note how the pipe from I3 to the reservoir has capacity 1, so the remaining flow has to be sent through I4.

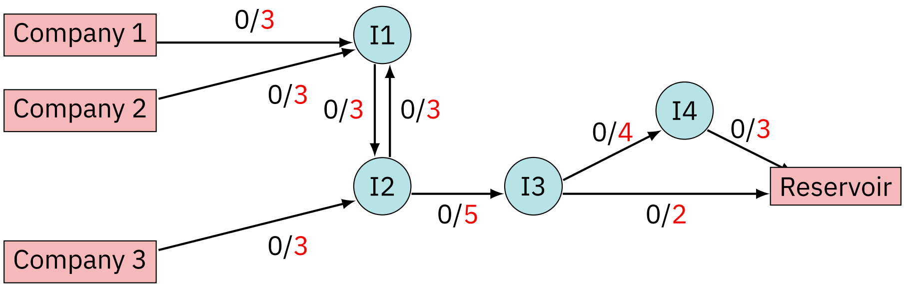


Figure 1.3.3. An imaginary sewer network with capacities as a directed graph, with a flow of value 3 from Company 1 to the Reservoir

Constructing flows with paths

One way of constructing a flow is by taking an $s-t$ path P, and choosing an amount of flow to send along this path. As long as this amount does not exceed the capacity of any arc on the path, this results in a valid $s-t$ flow.

Taking it one step further, we can choose several $s-t$ paths $P_1, P_2, \dots, P_n$ and assign each path a flow value. Look again at the flow f in $1.3.3$. This is a flow composed of flows on two paths, namely

- $\mathbf{P_1}$: Company 1 $\rightarrow$ I1 $\rightarrow$ I2 $\rightarrow$I3 $\rightarrow$Reservoir,
- $\mathbf{P_2}$: Company 1 $\rightarrow$ I1 $\rightarrow$ I2 $\rightarrow$I3 $\rightarrow$I4 $\rightarrow$Reservoir.

Notice how the paths overlap! In the example, the flow induced by each path is

$$\begin{aligned} f_{P_1} &= 1, \\ f_{P_2} &= 2. \end{aligned} \tag{1.3.1}$$

In a way, this is easier: we get flow conservation essentially for free! But finding the flow on each arc is a bit more tedious: give an arc a, we need to find all paths that include a, and sum the flow values of these paths.

Finding paths in a network

We have seen that a way of viewing $s-t$ flows is using $s-t$ paths. Small networks can easily be drawn. We can often easily find paths between two nodes. For large networks, this is time consuming and tedious. Luckily, we can let a computer find paths between nodes using an algorithm.

Algorithm

An algorithm is a step-by-step procedure to perform a given task. Algorithms can be executed by computers, but also by persons.

Below we give an algorithm to find *one* $s-t$ path in an undirected graph. However, the algorithm can be adjusted to find a path in a directed graph, or even to find *all* paths in a graph. For the latter, one needs to be careful; what if the graph has *cycles*?

Breadth First Search

Let $G=(V,E)$ be given, as well as nodes s and t. The following algorithm finds an $s-t$ path, or tells you one doesn't exist.

1. Put s in a list Q and mark s as visited
2. Repeat the following steps until Q is empty:
3.   **remove** the **first** vertex v from list Q
4.   for every neighbour w of the node v, do the following:
5.   put w at the end of list Q
6.   mark w as visited
7.   set the predecessor of w to v

When the algorithm is done, there are two options:

- Node t has no predecessor. This means there is no $s-t$ path in the graph.
- Node t has some predecessor w. This node w has again a predecessor. And so on, until we eventually find s. By tracing the predecessors like this, we find an $s-t$ path.

It turns out that if every edge in the graph has equal "length" (or "weight"), then this path is a "shortest" path (or "minimum weight path"). Furthermore, we do not only find a path to t (if it exists), but also one to every node that is *reachable* from s.

It may be hard to visualize what this algorithm does. Perhaps the video that this QR code leads you to will help. You can also click this link.

1.4. Game theory

During World War I, peace broke out. It was Christmas 1914 on the Western Front. Despite

strict orders not to chillax with the enemy, British and German soldiers left their trenches, crossed No Man's Land, and gathered to bury their dead, exchange gifts, and play games.

Meanwhile: it's 2024, the West has been at peace for decades, and wow, we suck at trust. Surveys show that, over the past forty years, fewer and fewer people say they trust each other. So here's our puzzle:

Why, even in peacetime, do friends become enemies? And why, even in wartime, do enemies become friends?

We think game theory can help explain our epidemic of distrust – and how we can fix it! So, to understand all this...
*Text taken from Evolution of Trust (https://ncase.me/trust/).

Game theory

Game theory is the study of mathematical models of strategic interactions. In general, you have a situation where participants need to make a decision. Each decision results in a profit or a penalty, and you want to understand how all the parties involved should behave if you want an optimal decision to be made.

Game theory has applications in many fields of social sciences, and is used extensively in economics, logic, systems science and computer science. Let us see some popular examples.

1.4.1. Cooperation vs Competition

A game is *cooperative* if the players are able to form alliances and work together towards an optimal scenario for the whole team. In this case they try to cooperate and maximize together the profit of the whole group.

A game *is non-cooperative* if players cannot form alliances or if all agreements need to be self-enforcing. In this case players are selfish and are competing with each other to maximize their individual profits.

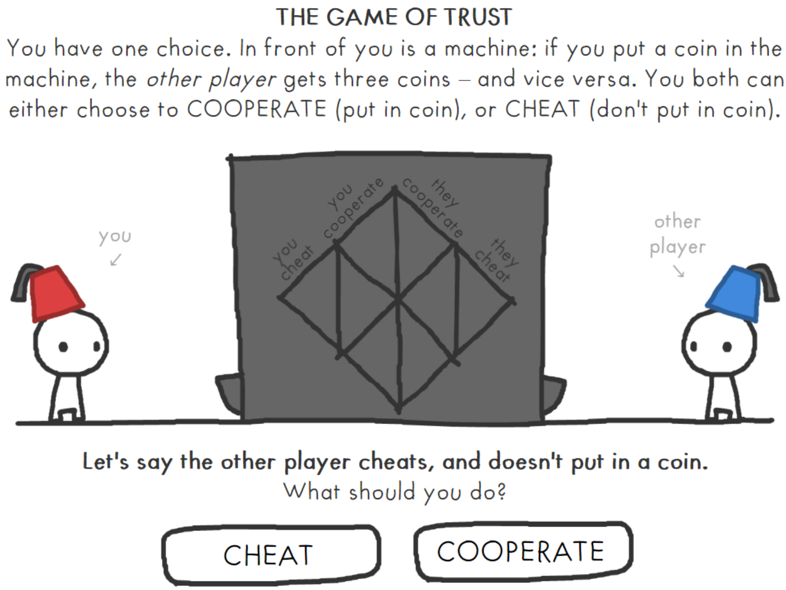


Figure 1.4.1. Picture taken from the online game Evolution of trust, you can play the game here https://ncase.me/trust/

1.4.2. Prisoner's dilemma

The prisoner's dilemma is a famous example of a non-cooperative game. We read on Wikipedia the following:

William Poundstone described this "typical contemporary version" of the game in his 1993 book Prisoner's Dilemma:

Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of speaking to or exchanging messages with the other. The police admit they don't have enough evidence to convict the pair on the principal charge. They plan to sentence both to a year in prison on a lesser charge. Simultaneously, the police offer each prisoner a Faustian bargain. If he testifies against his partner, he will go free while the partner will get three years in prison on the main charge. Oh, yes, there is a catch ... If both prisoners testify against each other, both will be sentenced to two years in jail. The prisoners are given a little time to think this over, but in no case may either learn what the other has decided until he has irrevocably made his decision. Each is informed that the other prisoner is being offered the very same deal. Each prisoner is concerned only with his own welfare—with *minimizing his own prison sentence.*

This leads to four different possible outcomes for prisoners A and B:

(1) If A and B both remain silent, they will each serve one year in prison.
(2) If A testifies against B but B remains silent, A will be set free while B serves three years in prison.
(3) If A remains silent but B testifies against A, A will serve three years in prison and B will be set free.
(4) If A and B testify against each other, they will each serve two years.

You can visually represent this situation as follows:

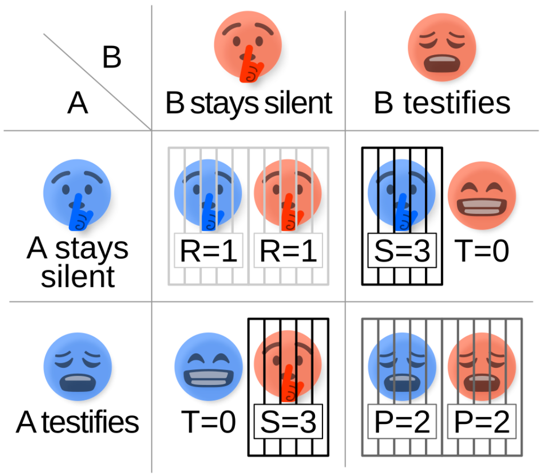


Figure 1.4.2. Visual representation of Prisoner's dilemma, made by cmglee under Creative Commons for Wikipedia, Prisoner's Dilemma.

You see that these "players" need to make a decision, without knowing the decision of the other player. Obviously they can't cooperate (what should they decide if cooperation was allowed?), and they decision has a big impact on both players. Game theory offers tools to understand how such decisions could be made. This brings us to the concept of a Nash equilibrium!

1.4.3. Nash equilibrium

The Nash equilibrium, named after American mathematician John Nash, who won both the Nobel Prize in Economics (1994) and the Abel Prize in mathematics (2015), is the most commonly-used solution concept for non-cooperative games. A Nash equilibrium is a situation where no player could gain by changing their own strategy (holding all other players' strategies fixed). Let's go back to the prisoner's dilemma.

Nash equilibrium

A Nash equilibrium is a situation where no player could gain by changing their own strategy, holding all other players' strategies fixed.

What is the Nash equilibrium in this game? We need to find the pair of decisions (A, B) such that each player won't change their decision if they know the decision of the other player. There are four possible pairs of decisions, namely (A stays silent, B testifies), (A stays silent, B stays silent), (A testifies, B testifies), (A testifies, B stays silent). We want to find in which of these cases a player doesn't change their decision if they know the decision of the other player. Consider the pair (A stays silent, B testifies). If player A knows the decision of player B, then of course they change their decision since by testifying A will go to prison for 2 years instead of 3 years. On the other hand player B wouldn't want to change their decision in this situation. So the pair (A stays silent, B testifies) is not a Nash equilibrium! Work out the other three cases in Exercise 4 to find the answer.

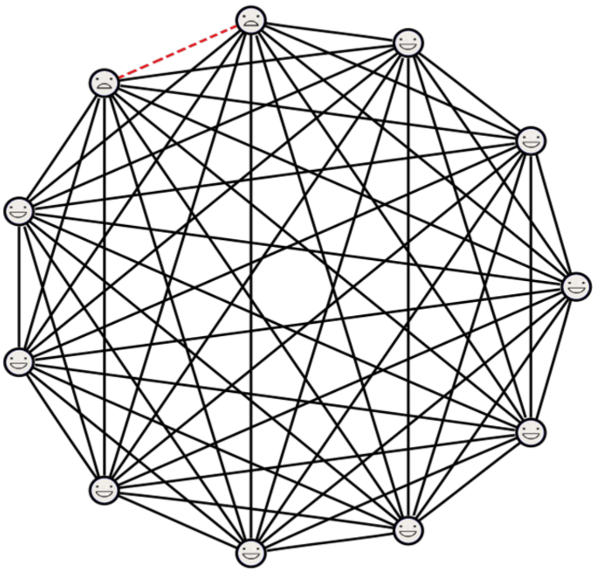


Figure 1.4.3. A graph from our article on the Friendship paradox: networkpages.nl/the-friendship-paradox-and-how-it-might-produce-a-biased-world/.

On the Network Pages

For further reading on probability theory, game theory, algorithms, and graph theory have a look at

(1) Trust and other wonderful mistakes humans make, by Ben Maylahn. network-pages.nl/trust-and-other-wonderful-mistakes-humans-make/

(2) No winner without a second place, by Ruben Brokkelkamp. network-pages.nl/no-winner-without-a-second-place/

(3) Our collection of articles on graph theory, networkpages.nl/tag/graph-theory/

(3) Our collection of articles on probability theory, network-pages.nl/tag/probability-theory/

Chapter 2

To be a social or a selfish driver?

Imagine a scenario where every driver in the city starts from the same location and needs to reach the same destination. Each driver wants to get there as fast as possible, choosing routes they think will minimize their travel time. But what happens when everyone makes these choices independently? Often, this can lead to heavy congestion, with everyone's trip taking longer than it would if they coordinated. This raises an interesting question: how much does everyone acting in their own best interest actually make things worse for everyone? In this chapter, we model this scenario using flows on graphs and attempt to answer this question using tools from Game Theory. Moreover, we will consider some options the government may have to improve traffic and how do these affect the network.

2.1. A mathematical model of selfish routing

A road traffic network consists of a collection of roads connecting various cities. The users of this network travel from a common starting location, called the *source*, to a common *target*[1]. These users drive vehicles, so we will refer to them as the *drivers* in the network. Throughout this chapter, we will take the perspective of a *social planner* (think of a person in the government responsible for traffic!) analyzing how traffic congestion builds up on city roads. In particular:

1. To represent the roads of a city, we use a directed graph $G = (V, E)$, where V is the set of vertices (representing intersections or cities) and E is the set of edges (rep-

[1] This is indeed a simplification. In the real world, drivers may of course have different origins and destinations. However, many of the conclusions we will arrive to, hold even for the general case!

resenting roads between the intersections or cities). If two vertices u and $v \in V$ are connected by a road going from u to v, we represent this road as an edge $(u, v) \in E$. Note that we allow *parallel* edges, meaning that for two vertices $u, v \in V$, there may be more than one road (directed edge) from u to v. Finally, we denote the origin location by a special vertex we call s and the destination by a vertex we call t. Naturally, we assume that every given $G = (V, E)$ with $s, t \in V$ we consider has at least one path from s to t.

2. To focus on the bigger picture of this large-scale system, we will assume that each driver contributes only a small amount to the overall congestion. That is, if there are 10 cars in the network, each driver controls $\frac{1}{10}$ of the traffic, or if there are 100 cars, then each one of them controls $\frac{1}{100}$ of the traffic. In general, if there are N cars and N is large, the fraction $\frac{1}{N}$ becomes very small, and therefore, we can assume that the impact of an individual driver is negligible, although a large group of drivers is still impactful. Therefore, we choose to model traffic as a continuous *flow*, rather than tracking individual drivers. More precisely, in a graph $G = (V, E)$ we assume that we want to transfer a flow of 1 (representing the entire population of drivers) from $s \in V$ to $t \in V$.

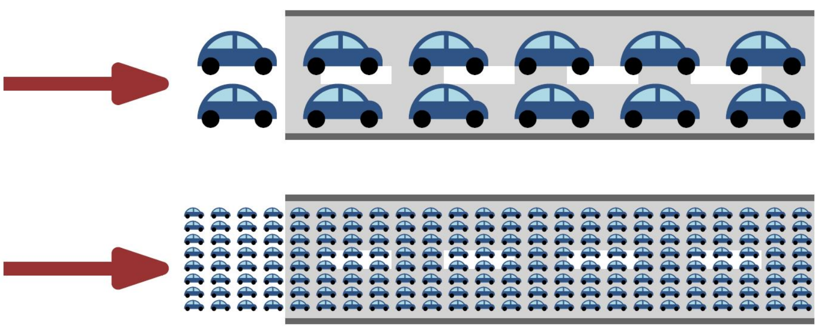


Figure 2.1.1. When the number of cars is large the impact of an individual driver is negligible. We study traffic as a continuous flow!

3. Every edge $e \in E$ has a non-negative number ℓ_e which represents the travel time on the road depending on the traffic. We can describe these numbers as a function on the edges of the graph, which we call the *latency function* ℓ_e.
4. Finally, we assume that drivers have full information about the state of the network. This allows them to make informed (selfish) decisions. One way to interpret this, is that drivers have access to navigation systems which inform them of the current traffic in real-time.

By summarizing the above, we can define an instance of a simple selfish routing game.

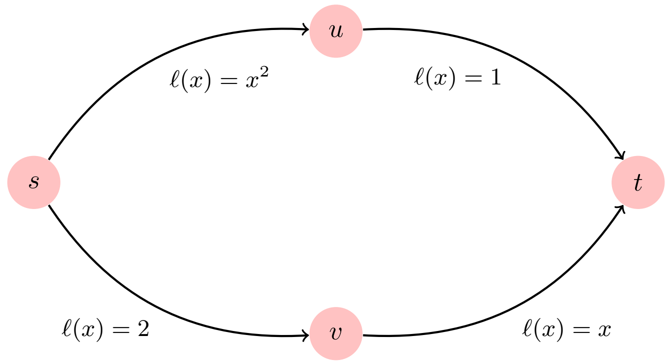


Figure 2.1.2. An example of a selfish routing game.

Selfish Routing Game

An instance of a selfish routing game is given by

- a directed graph $G = (V, E)$ with a vertex set V and an edge set E.
- a non-negative and continuously differentiable latency function ℓ_e for each edge $e \in E$
- a connected pair $(s, t) \in V \times V$ of source and target nodes associated with a flow demand of 1 (a flow of 1 must be sent from s to t)

For a selfish routing game with k paths from s to t, let $P_1, \ldots, P_k$ be the set of available paths from s to t. A feasible flow is a vector of numbers $f = (f_{P_1}, \ldots, f_{P_k})$ such that $f_{P_i} \geq 0$ for $i = 1, \ldots, k$, and the total flow across all paths satisfies $\sum_{i=1}^{k} f_{P_i} = 1$. For a given feasible flow f, we define for every $s - t$ path P its *total latency* as $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$, where f_e is the flow put by f on edge e by all the paths that include it. Finally, we define as $SC(f) = \sum_{i=1}^{k} \ell_{P_i}(f) f_P$ the *social cost* (or total latency) of a flow.

EXAMPLE 2.1.1. Consider the selfish routing game depicted by the graph in Figure 2.1.2. There are $k = 2$ paths from s to t: the first path is $s \to u \to t$ (denoted P_1) and the second path is $s \to v \to t$ (denoted P_2).

- The flow $f = \left(\frac{1}{2}, \frac{1}{2}\right)$ is feasible. This is because sending a flow of $\frac{1}{2}$ along P_1 and a flow of $\frac{1}{2}$ along P_2 results in the transfer of a total flow of $\frac{1}{2} + \frac{1}{2} = 1$ from s to t. The social cost of this flow, $SC\left(\frac{1}{2}, \frac{1}{2}\right)$, is calculated as:

$$\begin{aligned}
SC\left(\frac{1}{2}, \frac{1}{2}\right) &= \ell_{P_1}\left(\frac{1}{2}, \frac{1}{2}\right) \cdot \frac{1}{2} + \ell_{P_2}\left(\frac{1}{2}, \frac{1}{2}\right) \cdot \frac{1}{2} \\
&= \left(\ell_{(s,u)}\left(\frac{1}{2}\right) + \ell_{(u,t)}\left(\frac{1}{2}\right)\right) \cdot \frac{1}{2} + \left(\ell_{(s,v)}\left(\frac{1}{2}\right) + \ell_{(v,t)}\left(\frac{1}{2}\right)\right) \cdot \frac{1}{2} \\
&= \left(\frac{1}{4} + 1\right) \cdot \frac{1}{2} + \left(2 + \frac{1}{2}\right) \cdot \frac{1}{2} \\
&= \frac{15}{8} = 1.875.
\end{aligned}$$

- The flows $f = \left(\frac{1}{3}, \frac{2}{3}\right)$, $f = (1, 0)$, and $f = (0, 1)$ are also feasible for similar reasons.
- In general, for every $x \in [0, 1]$, the flow $f = (x, 1 - x)$ is feasible for this graph.

Computing the social cost for these flows is left as an exercise (see Exercise 1).

The social cost $SC(f)$ can be interpreted as the average travel time in the network when the flow f represents the traffic. Clearly, the goal of the social planner is to design the network in a way that minimizes this cost.

Optimal flow

Given a selfish routing game, we say that a feasible flow f^* is *optimal* if it minimizes the social cost. Formally, for every feasible flow f, it holds that $SC(f^*) \leq SC(f)$.

For general networks, computing an optimal flow may be challenging because it requires solving an optimization problem, typically a *convex program*, which might require help from software. However, for simple networks, it is relatively straightforward to find by hand. For example, can you compute the optimal flow for the selfish routing game in Figure 2.1.2? Unfortunately, in practice, the social planner cannot always expect to achieve traffic that follows the optimal flow. This is because drivers are selfish and will typically choose routes based on what they perceive to be the fastest. In fact, if the social planner suggested routes to simulate the optimal flow, drivers might refuse to follow them. Nevertheless, calculating the optimal flow and the minimum social cost of a selfish routing game is useful, as it provides a "best-case" scenario for the social planner.

2.2. The Nash flow and the price of anarchy

But how does traffic actually look for selfish drivers? What kind of flow do we expect to see on a network? To understand their behavior, let's consider an example commonly referred to in the literature as the Pigou[2] network.

[2] Arthur Cecil Pigou (1877–1959) was a professor of economics at the University of Cambridge.

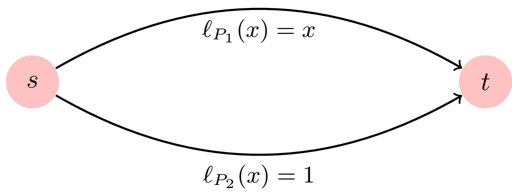


Figure 2.2.1. The Pigou network.

Let's first try to interpret the Pigou network qualitatively. There are two paths from s to t: the upper path (denoted P_1) and the lower path (denoted P_2), with each of them being a single edge or road. The latency function of P_1 is $\ell_{P_1}(x) = x$, which implies that the congestion on this road increases linearly with the number of cars using it. The traffic will only become "bad" if most of the cars in the network choose this path. You can think of P_1 as a highway with multiple lanes—it's generally fast unless it becomes overly crowded. On the other hand, the latency function of P_2 is $\ell_{P_2}(x) = 1$. This means that the congestion on this road is constant and does not depend on how many cars use it. However, this constant latency is relatively high, even if only a few cars take this route. We can think of P_2 as a narrow or old road that cannot handle heavy traffic efficiently.

Why is 1 very high?

We said that in the Pigou network path P_2 has a high latency value of 1. As a number 1 is of course not so large, but remember that there is high traffic and every driver is responsible for a fraction of the total traffic. Hence a latency value of 1 is equal to maximum traffic! When have the drivers choose P_1 then $\ell_{P_1}(\frac{1}{2}) = \frac{1}{2}$. When all drivers choose P_1 then $\ell_{P_1}(1) = 1$, which equals the constant value of P_2. Hence the experience of a driver taking P_2 is always similar to full traffic on P_1.

Consider the perspective of a driver following the narrow road P_2. What is on their mind? Well, unless every other car in the city has decided to follow P_1 to go from s to t (in which case P_1 is heavily congested, and the driver is indifferent about which road to take), they are probably regretting following P_2. This is because they could improve their travel time by switching to P_1. Mathematically, if there is a flow of $x > 0$ moving through P_2, the following condition must hold for the network:

$$\ell_{P_2}(x) \leq \ell_{P_1}(1 - x).$$

If this inequality holds then using P_2 was a good choice! This is an *equilibrium* condition. Such conditions must hold for all paths from s to t even in more general graphs. In a sense, these conditions describe the Nash equilibrium of this selfish routing game. Applying equilibrium conditions to all paths leads us to a special type of flow, which we call the Nash flow, or equilibrium flow.

Nash flow

Given a selfish routing game, we say that a feasible flow f_{nash} is a *Nash flow* (or equilibrium flow) if, for every pair of $s-t$ paths P_i and P_j that carry positive flow under f_{nash}, it holds that $\ell_{P_i}(f_{\text{nash}}) = \ell_{P_j}(f_{\text{nash}})$.

By applying the above definition to the selfish routing instance on the Pigou network (Figure 2.2.1), we find that the Nash flow is $f_{\text{nash}} = (1, 0)$, meaning all the traffic is routed through P_1. The social cost of this flow is

$$SC(f_{\text{nash}}) = SC(1,0) = \ell_{P_1}(1) \cdot 1 + \ell_{P_2}(0) \cdot 0 = 1.$$

To get a quantitative sense of how good or bad the Nash flow is, it makes sense to compare its social cost with the social cost of the optimal flow. To do this, we need to find an $x \in [0,1]$ such that

$$SC(x, 1-x) = \ell_{P_1}(x) \cdot x + \ell_{P_2}(1-x) \cdot (1-x) = x^2 + 1 - x$$

is minimized. Let $g(x) = x^2 + 1 - x$. The first derivative of $g(x)$ is $g'(x) = 2x - 1$. Therefore, we need to find an $x^* \in [0,1]$ such that $g'(x^*) = 0$ to find the global minimum[3]. Solving the equation, we obtain that $x^* = \frac{1}{2}$.
Therefore, the optimal flow for our Pigou network is $(x^*, 1-x^*) = (\frac{1}{2}, \frac{1}{2})$, and the minimum social cost is

$$SC(x^*, 1-x^*) = SC\left(\frac{1}{2}, \frac{1}{2}\right) = \ell_{P_1}\left(\frac{1}{2}\right) \cdot \frac{1}{2} + \ell_{P_2}\left(\frac{1}{2}\right) \cdot \frac{1}{2} = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}.$$

From the above, we can infer that the Nash flow of the Pigou network is $4/3$ times worse than the optimal flow in terms of social cost. In fact, the ratio

$$\frac{SC(f_{\text{nash}})}{SC(f^*)} = \frac{1}{\frac{3}{4}} = \frac{4}{3}$$

has an important meaning: it serves as a lower bound on the *price of anarchy* for a large class of selfish routing games. Here is the formal definition.

Price of Anarchy

The price of anarchy of a class of selfish routing games is the *worst-case ratio* of the social cost at equilibrium to the optimal social cost, across all games in the class and all Nash equilibria of those games.

[3] Note that x^* is indeed a global minimum and not a global maximum since $g''(x) = 2 > 0$ (by the second derivative test).

This metric is commonly used by theoretical computer scientists who are interested in analyzing the worst-case performance of a system. It is one of the key concepts in *algorithmic game theory*, a field that lies at the intersection of theoretical computer science and economics.

Remark 2.2.1. With our analysis, we were only able to prove a lower bound on the price of anarchy by examining a specific game (the Pigou network in Figure 2.2.1) and its specific Nash flow. Proving an upper bound is much more technically involved and will not be covered in this mini-lecture. However, it is remarkable that the bound of $4/3$ on the Price of Anarchy we found is the *worst-possible* for the class of games with latency functions of the form $\ell(x) = \alpha x + \beta$, where $\alpha, \beta \geq 0$ (affine functions)!

2.3. Improving Traffic with Tolls

Now imagine that the social planner is looking for a way to decrease congestion at equilibrium. Let's return to our example (Figure 2.2.1). Recall that at the equilibrium flow, all cars take path P_1. Since the drivers are free to choose which road they will take, they cannot be influenced directly. To reduce the total number of cars on P_1, the social planner would like to provide an incentive for drivers to take P_2 instead. To achieve this, let's assume that the social planner can impose tolls: a toll of $\tau_1 \geq 0$ on P_1 and a toll of $\tau_2 \geq 0$ on P_2 (see Figure 2.3.1).

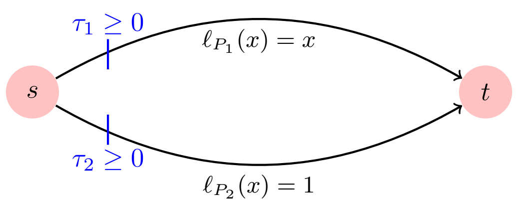


Figure 2.3.1. Tolls on the Pigou network.

Once the tolls τ_1 and τ_2 are introduced, the costs incurred by drivers are influenced not only by the latency functions, but also by these tolls. Therefore, for each selfish routing game and for each edge e with toll τ_e, the cost for a flow f is defined as:

$$c_e(f_e) = \ell_e(f_e) + \gamma \cdot \tau_e,$$

where $\gamma > 0$ is a parameter fixed in the model.

At first glance, this definition might seem strange because the functions ℓ represent latency (i.e., time), while each $\tau_e \geq 0$ is a monetary cost. This is where the parameter γ comes in: it represents how much drivers care about tolls. A high value of γ means that drivers are highly sensitive to paying tolls, while a low value means they are more tolerant. While drivers may have different tolerances for tolls, we assume here for simplicity that all drivers are have the same reaction to tolls.

Another important assumption is that the social planner (for example, the government) is not interested in maximizing profits from the toll system; their goal is still to minimize the average travel time. There are several ways to justify this. One explanation is that the government needs the toll revenue to cover the cost of implementing and maintaining the toll system and they can get this money by running the toll system for a sufficient time (e.g. first few months). After this, they are indifferent regarding revenue from tolls. Alternatively, the government might be using average travel time as a proxy for reducing emissions, aiming for an environmental goal, such as lowering greenhouse gas emissions.

Recall that without tolls, the social cost of the optimal flow $f^* = \left(\frac{1}{2}, \frac{1}{2}\right)$ is $SC\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{3}{4}$, whereas the Nash flow is $f_{\text{nash}} = (1, 0)$ with a social cost of $SC(1, 0) = 1$. Our goal now is to answer the following question:

Question

How can we set τ_1 and τ_2 in the network of Figure 2.3.1 so that the optimal flow $\left(\frac{1}{2}, \frac{1}{2}\right)$ becomes a Nash flow?

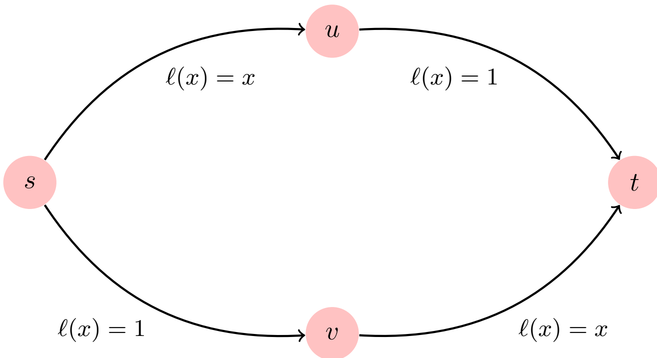
Keeping in mind that drivers now have cost functions affected by tolls, and the definition of a Nash flow, we conclude that to find a pair τ_1 and τ_2 with this property, it must hold that

$$\ell_{P_1}\left(\frac{1}{2}\right) + \gamma \cdot \tau_1 = \ell_{P_2}\left(\frac{1}{2}\right) + \gamma \cdot \tau_2 \iff \frac{1}{2} + \gamma\tau_1 = 1 + \gamma\tau_2 \iff \tau_1 = \frac{1}{2\gamma} + \tau_2.$$

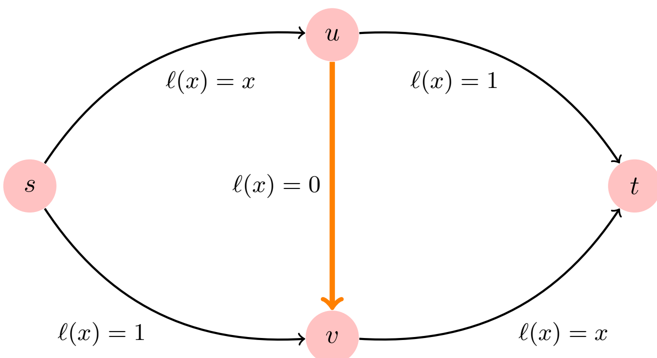
As we can see above, there are infinitely many such pairs. Notice that by introducing these tolls, the price of anarchy of this system becomes 1, meaning that the selfish behavior of drivers no longer negatively impacts the system at all.

2.4. The Braess Paradox

Consider the selfish routing game represented by the network in Figure 2.4.1a. There are two paths from s to t in this network: $s \to u \to t$ and $s \to v \to t$. Since both paths have the same latency functions (though applied in a different order), it is easy to see that the optimal flow f^* is $\left(\frac{1}{2}, \frac{1}{2}\right)$. Furthermore, by recognizing the symmetry in the latency functions and recalling the definition of Nash flows, we can conclude that the optimal flow f^* is also the Nash flow, i.e., $f^* = f_{\text{nash}}$. Thus, the social cost of this flow is $SC\left(\frac{1}{2}, \frac{1}{2}\right) = \left(\frac{1}{2} + 1\right) \cdot \frac{1}{2} + \left(1 + \frac{1}{2}\right) \cdot \frac{1}{2} = \frac{3}{2}$.



(a) A selfish routing game



(b) Adding a new highway.

Figure 2.4.1. The Braess Paradox instance.

Suppose now that the government decides to build a new road from u to v (see Figure 2.4.1b) in an attempt to further improve congestion in the network. In fact, suppose that this new road is built using state-of-the-art techniques in road engineering, and it guarantees that there will be no congestion. That is, for any flow of cars x on this road, we have $\ell(x) = 0$. Notice that now, in the new version of the network, there are *three* paths from s to t: the paths $s \to u \to t$ and $s \to v \to t$, as before, but also the new path $s \to u \to v \to t$. The government expects that traffic will now spread across the three paths, and thus congestion will improve.

But does this actually happen? Let's reconsider the situation from the perspective of a selfish driver. With the introduction of the new road from u to v, drivers who were considering paths that included $u \to t$ or $s \to v$ now have the option to follow the "zig-zag" route (see Figure 2.4.2) and reduce their travel time. However, this shift increases congestion on the roads $s \to u$ and $v \to t$. This process continues until all drivers have moved to the "zig-zag" path.

The new Nash flow

In the Nash flow for the network in Figure 2.4.1b, the entire flow of 1 starting from s and ending at t uses the path $s \to u \to v \to t$ (see Figure 2.4.2).

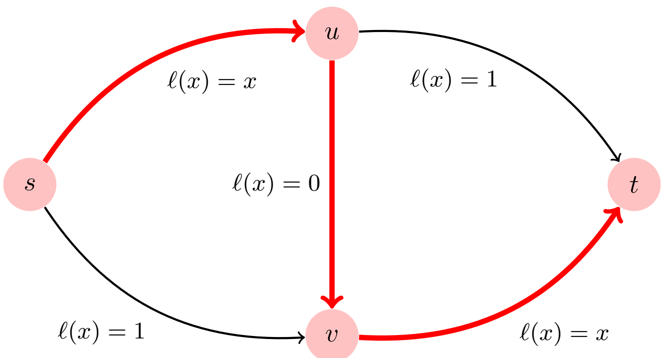


Figure 2.4.2. The new Nash flow

Notice that the social cost of this new Nash flow is 2, while before the introduction of the new road, it was $\frac{3}{2}$. Therefore, the overall traffic in the system has increased by 33%! There is a common belief in the mathematics and engineering community that when there is no traffic congestion in a network, adding new road segments will not negatively affect the average travel time. However, when congestion is already present, adding a shortcut may actually increase the overall travel time. The phenomenon we just analyzed is known as *Braess's paradox*, named after *Dietrich Braess*, a German mathematician who first discovered it in 1968. Braess observed that the introduction of a new road could paradoxically worsen traffic conditions, as drivers, acting selfishly, might individually optimize their own travel time but collectively increase congestion.

This raises an important question: in a congested network, are there edges that negatively impact the average travel time? This is a crucial factor that experts must consider when designing road networks. Although solving this problem exactly is difficult, there are algorithmic methods that can help identify potentially problematic edges.

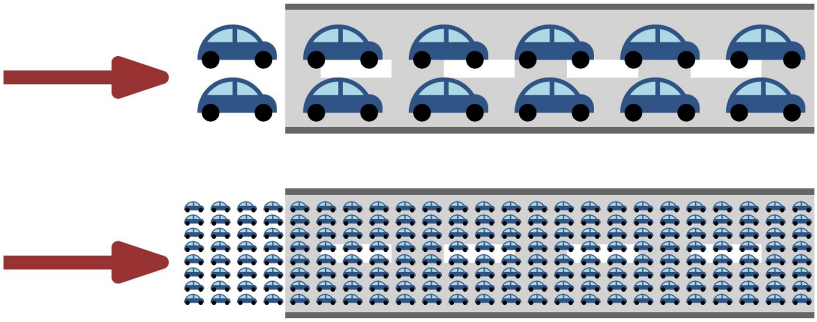


Figure 2.4.3. Viewing traffic as a flow, from the article of Peter Kleer "Traffic Congestion: Pigou's Example"

On the Network Pages

For further reading on game theory and road traffic have a look at these articles written by Peter Kleer:

(1) *Traffic Congestion: Pigou's Example,* networkpages.nl/equilibrium-congestion-models-pigous-example/.
(2) *Traffic Congestion: Tolls*, networkpages.nl/traffic-congestion-model-part-ii-tolls/.
(3) *Braess paradox,*networkpages.nl/traffic-congestion-iv-braess-paradox/.
(4) *Traffic lights no longer needed: back to the future* by Rik Timmerman, *networkpages.nl/traffic-lights-no-longer-needed-back-to-the-future/*.

Chapter 3

Queueing Theory - Waiting in an efficient way

Queueing theory

Queueing theory is a branch of *operations research* that studies waiting lines or queues from a mathematical perspective.

Some typical everyday examples of queueing systems can be found in supermarkets, industrial production systems and hospitals. In a supermarket customers arrive to the counters, they may have to wait in the queue until their turn comes, they are served and then leave the supermarket. In an industrial production system, like a factory producing cars, the products also have to undergo multiple stages until they are assembled and the servers may be either machines or individuals. Finally, patients arriving to a hospital often need access to resources like doctors, beds, medicine and equipment. A new patient can go into treatment only when the hospital has the necessary resources available, for example only if there are free beds. In short, queueing theory helps us to analyse such systems and make important decisions about the layout, capacity and control.

3.1. A mathematical model of a queue

To study any kind of system or real-life situation we first have to construct a mathematical model. To illustrate what a mathematical model is, you can think of a toy car of a Ferrari (which is also called a *model car*). Such a toy car is not precisely like the Ferrari, since it does not contain a working engine, is made of different material, is much smaller, and so on. However, it does give you a good idea of the shape, how it looks when it is driving, and how it compares to other toy cars. As another example, architects make models of their buildings on a small scale (also called a *scale model*) to study how they would look, how much light will enter the building, how much material is needed, and so on. In a similar way, mathematical models describe a real-life phenomenon, using mathematical concepts and language. The model will not resemble reality perfectly, but can be used to learn from.

A mathematical model

When we speak of a mathematical model we mean a description of a system or some real life situation, in this case a queue, using mathematical concepts.

In our setting we are interested in **queueing models**. The idea behind such a model is to replicate the behavior of a queue as accurately as possible, so that the model can be used to make predictions on how the system will behave. Among others, a queueing model is characterised by:

- **How customers arrive.**
  Customers arrive to a system at, possibly random, points in time, we call this the *arrival process*. The time between two consecutive arrivals is called the interarrival time and is usually described by a random variable. We assume that the interarrival times between customers are independent and have a common probability distribution. In many practical situations customers arrive according to a Poisson stream (i.e., the interarrival times have an exponential distribution). Customers may arrive one by one, or in batches. An example of batch arrivals is the customs once at the border where travel documents of bus passengers have to be checked.
- **The behavior of customers.**
  Customers may be patient and willing to wait. Or customers may be impatient and leave after a while. For example, in call centres, customers will hang up when they have to wait too long before an operator is available, and they possibly try again after a while.
- **How customers are served.**
  Each customer needs some time to be served by the server. This time is called the service time of that customer. Often customers don't have exactly the same service time, hence in many cases we consider the service time to be a random variable. Usually we assume that the service times are independent and have a common distribution function, and that they are independent of the interarrival times. For example, the service times can be deterministic or exponentially distributed. It can also occur that service times depend on the queue length. For example, the processing rates of the machines in a production system can be increased once the number of jobs waiting to be processed becomes too large.
- **The service discipline.**

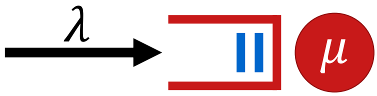


Figure 3.1.1. Illustration of an M/M/1 queue

Customers can be served one by one or in batches. We have many possibilities for the order in which customers can be served. We mention:
  - first come first served, i.e., in order of arrival;
  - random order;
  - last come first served (e.g., in a computer stack or a shunt buffer in a production line);
  - priorities (e.g., rush orders first, shortest processing time first);
  - processor sharing (in computers that equally divide their processing power over all jobs in the system).
- **The service capacity.**
  There may be a single server or a group of servers helping the customers.
- **The waiting room.**
  There can be limitations with respect to the number of customers in the system, i.e. the customer being served, if any, and the number of customers waiting in the queue. For example, in a data communication network, only finitely many cells can be buffered in a switch. The determination of good buffer sizes is an important issue in the design of these networks.

All these different aspects of queueing systems result in a huge variety of queueing models, which means that an efficient way to characterise queueing models based on its properties is vital. Luckily, D.G. Kendall introduced in 1953 a shorthand notation to characterise queueing models. We explain the notation via the simplest model denoted by $M/M/1$. In the $M/M/1$ queueing model each letter represents a property of the system, in particular:

- **The first letter:** the interarrival time between arriving customers has an exponential distribution with parameter λ. The *M* stands for Memoryless.
- **The second letter:** the service time distribution has an exponential distribution with parameter μ. Other models that are often studied are $M/D/1$ which stands for deterministic service times or $M/G/1$ which stands for general service times.
- **The number:** the number of servers in the queueing model.

3.2. The $M/M/1$ queue

After all this notation we can finally start analysing the $M/M/1$ queue. Let Q be a random variable that denotes the number of customers in the system. We start analysing the Q by constructing a *flow diagram* for Q. Below we explain how to construct this flow diagram.

Figure 3.2.1. State space for Q.

In this figure the numbers denote the state of the system, i.e., how many customers are in the system. Suppose that $L = i$, that is there are i customers in the system. Then two things can occur: the customer who is being served departs from the system before a new customer arrives, or a new customer arrives before the customer who is being served departs the system. The first event corresponds to the transition $\{Q = i\} \to \{Q = i - 1\}$ since a customer departs. The service time has an exponential distribution with parameter μ, hence we say that the transition $\{Q = i\} \to \{Q = i - 1\}$ occurs with rate μ. On the other side the second event corresponds to the transition $\{Q = i\} \to \{Q = i + 1\}$ since a customer arrives to the system. The interarrival time has an exponential distribution with parameter λ, hence the transition $\{Q = i\} \to \{Q = i + 1\}$ occurs with rate λ. We can illustrate these transitions using the following flow diagram, where we have chosen the case i = 1.

Figure 3.2.2. Flow from state $\{Q = 1\}$.

Doing this for all possible states we obtain the following flow diagram

Figure 3.2.3. Flow diagram for the $M/M/1$ queue.

Since the customers arrive to the system according to a Poisson process, that is at random times, and have a service time which is also random, i.e. exponentially distributed, we observe that Q will also be a random variable. Thus we want to know the probabilities that at an arbitrary point in time there will k customers in the system (which means 1 customer in service, and $k-1$ customers waiting for service). We denote this probability by p_k. We are going to compute the probabilities $p_k = \mathbb{P}(Q = k)$, for $k = 0, 1, 2, \ldots$ using a *flow conservation argument*.

Flow Conservation Argument

The probability flux in any subset of states is equal to the probability flux out of that subset of states. Intuitively, this means that you enter a state just as many times as you leave a state.

Figure 3.2.4. Flow diagram of the probability flux for the $M/M/1$ queue.

Consider for example the set consisting of the state 0, i.e., where no customers are present in the system. Then the probability flux out of this set is λp_0, because we are in state 0 with probability p_0 and we leave it with rate λ. The probability flux into the set $\{0\}$ is equal to μp_1, because we can reach state 0 only from state 1 in which we are with probability p_1 and the transition from state 1 to 0 happens with rate μ. Then we get the first equation

$$\lambda p_0 = \mu p_1,$$

which we can rewrite to

$$p_1 = \frac{\lambda}{\mu} p_0 = \rho p_0, \tag{3.2.1}$$

where $\rho = \lambda/\mu$. If $\rho < 1$, then ρ is called the *occupation rate*, because it is the fraction of time the server is working. Intuitively, $\rho < 1$ means that there are on average more departures than arrivals so the queue will not keep growing. Suppose now that we consider the subset $\{1\}$, then we obtain the equation

$$(\lambda + \mu) p_1 = \lambda p_0 + \mu p_2,$$

which, after substituting p_1 from (3.2.1), can be rewritten to

$$p_2 = \rho^2 p_0.$$

In general we obtain the equations

$$(\lambda + \mu)p_k = \lambda p_{k-1} + \mu p_{k+1}, \qquad k = 1, 2, \ldots$$

and

$$p_k = \rho^k p_0, \qquad k = 0, 1, 2, \ldots.$$

Hence it suffices to compute p_0, which denotes the probability that there are no customers waiting, and there is nobody being served. We know that the sum of all the probabilities has to be equal to one, hence

$$\sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} \rho^k p_0 = 1.$$

Solving this equation yields

$$p_0 = 1 - \rho.$$

For derivation of this result have a look at Exercise 9. Hence we obtain the following result for the desired probabilities

$$p_k = \rho^k (1 - \rho), \quad k = 0, 1, 2, \ldots$$

Hence, the number of customers in an M/M/1 system is a geometric random variable with success probability $1 - \rho$ (see Section 1.2.1). With this result already some quantities can be computed. For example the average number of customers in the system is equal to

$$\mathbb{E}[L] = \sum_{k=0}^{\infty} k p_k = \sum_{k=0}^{\infty} k \rho^k (1 - \rho) = \frac{\rho}{1 - \rho}. \tag{3.2.2}$$

See also Exercise 9 for more details on how to derive this result.

3.2.1. Little's law

Little's law is the most important relation between $\mathbb{E}[L^Q]$, the mean number of customers in the queue (there may be one more in service), $\mathbb{E}[W]$, the mean waiting time of a customer and λ, the average number of customers entering the system.

Little's Law

Little's law states that

$$\mathbb{E}[L^Q] = \lambda \mathbb{E}[W].$$

Intuitively, this result can be understood as follows. Suppose that all customers pay 1 euro per unit time while in the queue. This money can be earned in two ways.

- The first possibility is to let pay all customers continuously in time. Then the average reward earned by the system equals $\mathbb{E}[L^Q]$ euro per unit time.
- The second possibility is to let customers pay 1 euro per unit time for their residence in the queue when they leave. In equilibrium, the average number of customers leaving the system per unit time is equal to the average number of customers entering the system. So the system earns an average reward of $\lambda\mathbb{E}[W]$ euro per unit time.

The system earns the same in both cases. Hence these two quantities are equal which is that Little's law states.

3.3. The M/M/1/K Queue

The M/M/1/K queue is a finite-capacity queueing system. The system has a finite capacity of K (including the customer being served). In the M/M/1/K queue, only one job receives service at a time. At any moment, there can be at most K jobs in this system. Whenever a job arrives and the system is full, it will be blocked and it will leave forever. The possible states are $Q = 0, 1, 2, \ldots, K$. State K means the system is full, no more customers can enter.

Why is M/M/1/K so interesting? Let us consider the following situation. Each arriving customer can choose to join or not join the queue. When such decisions become part of the problem we are studying then the problem becomes a *game theoretical* problem. Imagine the following situation: Upon completion of service, the customer is endowed with a reward R (expressible in monetary units). All customer rewards are equal. The cost per unit time to a customer for staying in a queue (i.e. for queueing) is C monetary units per unit time. All customer costs are equal. Each arriving customer weighs the net gains associated with joining or not joining. The net gain, in the first case if $Q = i$, meaning there are i customers waiting upon arrival, is on average equal to

$$G_i = R - C\frac{i}{\mu},$$

In Exercise 11 you are asked to explain why this formula is correct. In the alternative case the net gain is zero. We see that as the number of customers in the systems, i.e. i, increases the net gain decreases, at some point it may reach zero or even become negative. At that point it is not profitable to wait and people no longer join. Hence there exists an integer n_s that satisfies the two inequalities

$$R - C\frac{n_s}{\mu} \geq 0 \qquad\qquad R - C\frac{n_s + 1}{\mu} < 0$$

If the number of customers in the system encountered an arriving customer is less or equal to n_s, then the arriving customer will choose to join, otherwise they will balk. We can incorporate the above two equations into one expression

$$n_s \leq \frac{R\mu}{C} = v_s < n_s + 1\,.$$

If we use the ceil function $\lceil\cdot\rceil$, then $n_s = \lceil v_s \rceil$ is the largest integer not exceeding v_s. Note that n_s depends on μ, R, C, but not on the arrival rate λ.

3.4. Optimization

A queueing system can also be analyzed from a social perspective. In this context, the objective of controlling the system is to maximize social welfare, defined here as the total expected net benefit to society, encompassing both customers and servers. Under this approach, any payment exchanged between individuals within the population has no net impact on social welfare and, consequently, does not affect the system's optimization. Therefore, the social objective is to maximize the total benefits from service while minimizing waiting and operational costs.

Maximizing social welfare

Social welfare is defined here as the total expected net benefit to society, encompassing both customers and servers. The social objective is to maximize the total benefits from service while minimizing waiting and operational costs.

Overall optimization. Consider a queue with the following setup:

- The arrival rate is λ,
- The maximum capacity of the system is K, so there will be at most K customers in the system.
- Customers will join the queue as long as the number of customers is less than K,
- Hence the joining probability is 1 minus the probability that there are K people in the system upon arrival, which is $1 - p_K$.
- Each joining customer will get a reward of R. [1]
- The cost of queueing is still C monetary units per unit time.
- The expected waiting time for each customer **that joins the queue** is denoted by $\mathbb{E}[W]$.

Imagine a customer arrives at the queue and chooses to join the queue. Then they eventually get a reward of R. It also costs them, on average, $\mathbb{E}[W]$ times C monetary units for

[1] Note that the server may charge a price for the service, but since this is a transaction between the server and customers, so it will not be calculated in the social welfare.

waiting. The expected payoff per customer that joins the queue is therefore

$$R - C\mathbb{E}[W].$$

We can use Little's Law to exchange mean waiting time for mean queue length. We must be careful though; we cannot simply use λ as the arrival rate, because the mean waiting time was defined for the customers that join the queue. So we need the arrival rate of customers that actually join the queue. Since there is a probability of p_K that an arriving customer stumbles upon a full queue, the arrival rate of customers that join the queue is $\lambda(1 - p_K)$. Little's Law tells us

$$E[L] = (1 - p_K)\lambda E[W].$$

We then find

$$R - C\mathbb{E}[W] = R - \frac{1}{(1 - p_K)\lambda} C\mathbb{E}[L]$$

Note that the payoff for customers that don't join the queue is simply zero. This means that the social welfare (the total payoff per unit time) is

$$\mathsf{SW} = \lambda(1 - p_K)\left(R - \frac{1}{1 - p_K} C\mathbb{E}[L]\right) = \lambda R(1 - p_K) - C\,\mathbb{E}[L].$$

In Exercise 12, you will calculate the optimal threshold that maximizes SW. Compare it with n_s. The conclusion we have in this section the optimal individual strategy is not optimal for society as a whole. One way to regulate it is by requiring a payment for the service.

Revenue maximization. Now consider a scenario where the server imposes an admission fee θ. Unlike the social perspective, where funds collected are viewed as transfer payments, here they are treated as the server's profits. In this model, the fee θ is made known to customers, who then decide whether to join the queue based on this fee. Suppose a customer who observes i customers already in the system will only enter if the reward R is at least equal to the expected total cost $\theta + C\frac{i}{\mu}$.
Just like before, there is a certain threshold, say k, at which customers stop joining. If a customer arrives, there is a p_k chance they stumble upon a full queue. If so, they don't join. The rate at which people join the queue is therefore $\lambda(1-p_k)$. Denote by M the rate of profit for the server. Then

$$\mathsf{M} = \lambda(1 - p_k)\theta.$$

We have that k is is the smallest number such that it is not profitable to join the queue any more, because the costs outweigh the reward. So k is the smallest number for for which

$$\theta + C\frac{k}{\mu} > R.$$

For numbers smaller than k, the '$>$' becomes '$\leq$'. Let us assume for simplicity that k satisfies

$$\theta + C\frac{k}{\mu} = R.$$

Then we can rewrite M as follows:

$$\begin{aligned} \mathsf{M} &= \lambda\,(1-p_k)\,\theta \\ &= \lambda\,(1-p_k)\,\left(R - C\frac{k}{\mu}\right) \\ &= \lambda\,(1-p_k)\,R\left(1 - C\frac{k}{R\mu}\right) \\ &= \lambda\,(1-p_k) R\left(1 - \frac{k}{v_s}\right). \end{aligned} \tag{3.4.1}$$

We can view our queue as an $M/M/1/k$ queue (note the small k rather than big K). In Exercise 9, you are tasked to deduce the blocking probability of such queue. It turns out that is

$$p_k = \frac{(1-\rho)\rho^k}{1-\rho^{k+1}}.$$

Hence

$$\begin{aligned} 1 - p_k &= 1 - \frac{(1-\rho)\rho^k}{1-\rho^{k+1}} \\ &= \frac{1-\rho^{k+1}}{1-\rho^{k+1}} - \frac{(1-\rho)\rho^k}{1-\rho^{k+1}} \\ &= \frac{1-\rho^k}{1-\rho^{k+1}}. \end{aligned} \tag{3.4.2}$$

This gives us the final result for M, which is

$$\mathsf{M} = \lambda R\,\frac{1-\rho^k}{1-\rho^{k+1}}\left(1 - \frac{k}{v_s}\right).$$

In Exercise13, calculate the integer n_r that maximizes the toll revenue, and compare it with n_s.

Figure 3.4.1. Consult a mathematician before you visit Disneyland, because queues can be large! By Ellen Cardinaels.

On the Network Pages

For further reading on queueing theory and its applications have a look at:

(1) *The quest for a beter Internet* by Mark van der Boor,
networkpages.nl/the-quest-for-a-better-internet/.

(2) *Consult a mathematician before you visit Disneyland* by Ellen Cardinaels,
networkpages.nl/consult-a-mathematician-before-you-visit-disneyland/.

(3) *Can flipping the queue spare you time* by Youri Raaijmakers,
networkpages.nl/can-flipping-the-queue-spare-you-time/.

(4) *Traffic lights no longer needed: back to the future* by Rik Timmerman,
networkpages.nl/traffic-lights-no-longer-needed-back-to-the-future/.

Chapter 4

Exercises

4.1. Probability theory

Conditional probabilities and expectations

EXERCISE 1. A conditional probability is denoted by $\mathbb{P}(A|B)$, which corresponds to

the probability of A happening, given that B happens.

Let's look at a few simple examples. We denote by X the random variable that represents the number that you roll with a six-sided die.

1. What is the probability that you roll a 6 with a six-sided die? In a formula: $\mathbb{P}(X=6)$.
2. What is the probability that you roll a 6, given that you roll at least a 4; $\mathbb{P}(X=6|X\geq 4)$?
3. You can use the following formula to compute conditional probabilities:

$$\mathbb{P}(A|B)=\frac{\mathbb{P}(A \text{ and } B)}{\mathbb{P}(B)}. \tag{4.1.1}$$

   Check that this formula works by solving the second question again, but now with the formula.
4. Similarly to probabilities, we can also look at expectations. What is the expected number you roll with a six-sided die? In formulas: $\mathbb{E}(X)$.
5. What is the expected number that you roll, given that you roll at least a 4; $\mathbb{E}(X|X\geq 4)$?

EXERCISE 2. The geometric random variable has expectation equal to

$$\mathbb{E}[G(p)]=\sum_{k=0}^{\infty}k\mathbb{P}(G(p)=k)=\sum_{k=0}^{\infty}k(1-p)^k p=\frac{1-p}{p}. \tag{4.1.2}$$

Hint: Use the identity

$$\sum_{k=0} x^k = \frac{1}{1-x}, x < 1$$

and take derivatives with respect to x on the left and right-hand sides.

The exponential distribution

EXERCISE 3. The exponential distribution is defined in the following way. Suppose that X is exponentially distributed with parameter λ. Then $\mathbb{P}(X < t) = 1 - e^{-\lambda t}$.

1. Calculate $\mathbb{P}(X \geq t)$.
2. Calculate $\mathbb{P}(1 < X < 2)$.
3. Calculate the expectation of the exponential distribution with the following formula:

$$\mathbb{E}(X) = \int_0^\infty \mathbb{P}(X \geq t) dt.$$

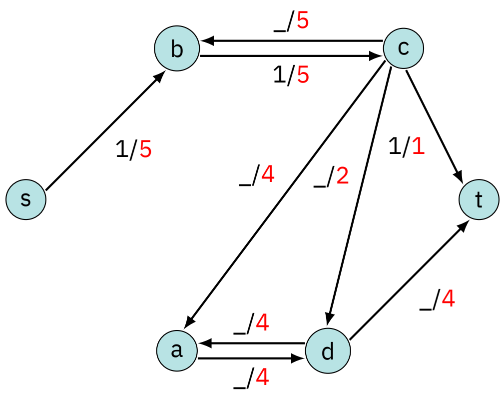
4. Use Equation (4.1.1) to prove the memoryless property of the exponential distribution:

$$\mathbb{P}(X > t + u | X > t) = \mathbb{P}(X > u).$$

4.2. Graph theory

Maximum flows

Consider the following flow network with capacities:



Suppose all arcs with flow value "_" are given value 0. It is possible to send more flow from s to t.

1. Find an $s-t$ flow with a flow value that is as high as possible. You may write the edge flow values in the booklet (you do not need to copy the flow network)

We say a flow is *maximum* if there does not exist a flow with a higher value.

2 Prove that your $s-t$ flow is a maximum flow.

Value of a flow

Recall the definition of an $s-t$ flow in Section 1.3. The definition is largely in words. It says

> The value of an $s-t$ flow is the total flow going out of s.

You can imagine that the (total) flow going out of s should equal the (total) flow going into t. After all, flow is *conserved*. We will define flow conservation mathematically, and prove that this is true.

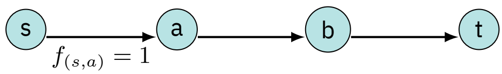
Define $\delta_+(v)$ as the set (collection) of arcs going out of v. Define $\delta_-(v)$ as the set of arcs going into v. We can write flow conservation as:

$$\sum_{a\in\delta_+(v)} f_a = \sum_{a\in\delta_-(v)} f_a, \quad \text{for all } v\in V \text{ with } v\neq s \text{ and } v\neq t. \tag{4.2.1}$$

We can rewrite this as:

$$\sum_{a\in\delta_+(v)} f_a - \sum_{a\in\delta_-(v)} f_a = 0, \quad \text{for all } v\in V \text{ with } v\neq s \text{ and } v\neq t. \tag{4.2.2}$$

Consider the following flow network with $s - t$ flow f:



1. What can you say about $f_{(a,b)}$ based on Equation (4.2.2)?
2. What can you conclude about the flow going into t? **Hint:** use 1. and Equation (4.2.2) again
3. Forget about the above flow network. Prove that for any flow network and any $s - t$ flow f it holds that

$$\sum_{a \in \delta_+(s)} f_a = \sum_{a \in \delta_-(t)} f_a. \tag{4.2.3}$$

Hint: Add all the equations given by (4.2.2).

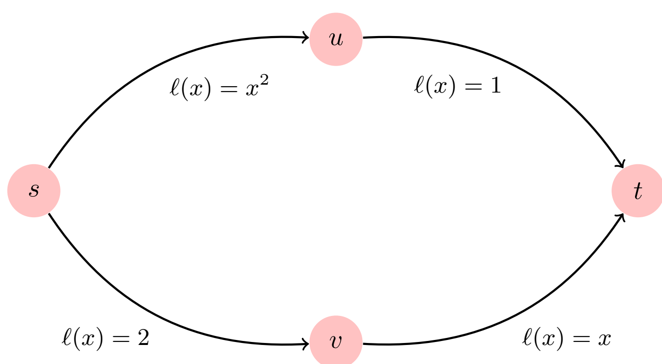
4.3. Game theory

EXERCISE 4. Find the Nash equilibrium for Prisoner's dilemma.

4.4. Selfish routing

Optimal flow

EXERCISE 5. Consider the selfish routing game below.

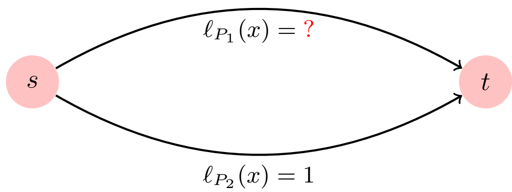


1. Compute the social cost for the flow $(\frac{1}{3}, \frac{2}{3})$.
2. Compute the social cost for the flow $(0, 1)$.
3. Given an $x \in [0, 1]$, compute the social cost for the flow $(x, 1 - x)$.

4. What is the optimal flow f^*? **Hint**: consider the function $g(x) = SC(x, 1-x)$ for $x \in [0, 1]$.
5. What is the optimal social cost?

Optimal flows versus Nash flows

EXERCISE 6. Consider the following family of Pigou networks. Compute an optimal flow



and a Nash flow when:

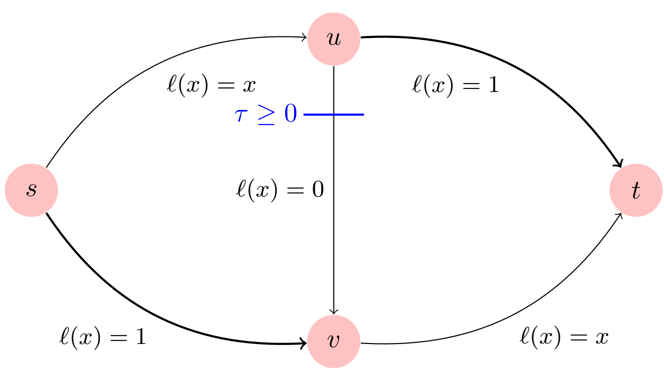
1. $\ell_{P_1}(x) = \frac{x^2+x}{2}$.
2. $\ell_{P_1}(x) = x^3$.
3. $\ell_{P_1}(x) = x^d$, for a given positive integer d.

Tolls on the Pigou network

EXERCISE 7. Recall our Pigou network instance with tolls $\tau_1, \tau_2 \geq 0$ in Figure 2.3.1. Of all the possible pairs $\tau_1, \tau_2 \geq 0$ for which the flow $(\frac{1}{2}, \frac{1}{2})$ is a Nash flow, find a payment scheme τ_1^*, τ_2^* which minimizes the money the government gets from the drivers.

Using tolls to get a Nash flow

EXERCISE 8. Consider the network below.



1. In this network, there are three paths from s to t:
   - The path $s \to u \to t$ (let's call it P_1).
   - The path $s \to v \to t$ (let's call it P_2).
   - The path $s \to u \to v \to t$ (let's call it P_3).

   Compute the social cost for the flow $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.
2. Consider adding a toll station to edge $u \to v$. For a given toll-sensitivity parameter $\gamma > 0$, what should the toll τ be so that $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is a Nash flow?

4.5. Queueing theory

Mean queue length

EXERCISE 9. We introduce $\rho = \lambda/\mu$ to make the calculations easier. In the $M|M|1$ queue we found that the probability of having i jobs in the system, in equilibrium, equals

$$p_i = (1-\rho)\rho^i.$$

1. Of course, the sum of all these probabilities should sum up to 1. Prove that

$$\sum_{i=0}^{\infty} p_i = 1.$$

Hint

Consider $S_n = \sum_{i=0}^{n} p_i$ and compute $(1-\rho)S_n$. What can you say about S_n as n goes to infinity?

2. We can calculate the mean queue length using these probabilities;

$$\mathbb{E}(L) = \sum_{i=0}^{\infty} i p_i = \sum_{i=0}^{\infty} i(1-\rho)\rho^i.$$

   Calculate $\mathbb{E}(L)$.

Hint

Take the derivative with respect to ρ of

$$\sum_{i=1}^{\infty} \rho^i$$

$M|M|1|k$

EXERCISE 10. In the lecture we drew the transition diagram and calculated the equilibrium probabilities of the $M|M|1$ queue, which is a system where 1 job can be served at a time. In this set of questions, we will consider the $M|M|1|k$ queue. Draw the transition diagram, calculate the equilibrium probabilities, and find

1. the blocking probability p_b.
2. the expected number of customers joining the queue in unit time.
3. the expected number of customers leaving the service station in unit time, and compare it with (2).
4. the degree of utilization of the service station, and compare it with ρ.
5. the expected number of customers in the system $\mathbb{E}(L)$.

EXERCISE 11. Explain why the net gain, in the case there are $Q = i$ customers in the system, is, on average, equal to

$$G_i = R - C\frac{i}{\mu}.$$

Hint: Remember that the time it takes for a customer to be served follows an exponential distribution with parameter μ. The expectation of this probability distribution is equal to $\frac{1}{\mu}$.

Optimization

EXERCISE 12. We are given that SW in its dependence on n is "discretely unimodal". This means that a local maximum is a global maximum.

1. If n_o is the integer that maximizes SW, prove that n_0 shall satisfy

$$\frac{n_0(1-\rho) - \rho\,(1-\rho^{n_0})}{(1-\rho)^2} \leq \frac{R\,\mu}{C} < \frac{(n_0+1)(1-\rho) - \rho\,(1-\rho^{n_0+1})}{(1-\rho)^2}$$

> **Hint**
>
> Treat SW as a function of n, then n_0 shall satisfy SW$(n_0) \geq$ SW(n_0+1) and SW$(n_0) \geq$ SW(n_0-1).

2. Prove that $n(1-\rho) - \rho\,(1-\rho^n)$ is increasing with n.
3. This means we have $n_0 = \lceil v_0 \rceil$, where v_0 solves

$$v_0\,(1-\rho) - \rho(1-\rho^{v_0}) = v_s\,(1-\rho)^2\,.$$

Can you see $n_0 \leq n_s$?

EXERCISE 13. We know that

$$\mathsf{M} = \lambda R\,\frac{1-\rho^n}{1-\rho^{n+1}}\left(1-\frac{n}{v_s}\right).$$

Given that M in its dependence on n is "discretely unimodal", can you show that n_r that maximize M is $n_r = \lceil v_r \rceil$, where v_r satisfies

$$v_r + \frac{(1-\rho^{v_r-1})(1-\rho^{v_r+1})}{\rho^{v_r-1}\,(1-\rho)^2} = v_s\,?$$

Chapter 5

Solutions to exercises

5.1. Probability theory

Conditional probabilities and expectations

1. $$\mathbb{P}(X=6)=\mathbb{P}(\text{you get a 6 when rolling a six-sided die})=\frac{1}{6},$$

   since it is equally probable to obtain any of the six sides.
2. This is a conditional probability. You don't know exactly what the outcome is but you know that it is at least 4. This means that the die number is either a 4 or a 5 or a 6. Yes now you have three possible outcomes, given the condition, not six. All three are equally probable, hence the desired probability is equal to

   $$\mathbb{P}(X=6|X\geq 4)=\frac{1}{3}.$$

3. $$\mathbb{P}(X=6|X\geq 4)=\frac{\mathbb{P}(\{X=6\}\text{ and }\{X\geq 4\})}{\mathbb{P}(X\geq 4)}=\frac{\frac{1}{6}}{\frac{1}{2}}=\frac{1}{3}. \tag{5.1.1}$$

4. $$\mathbb{E}(X)=\sum_{i=1}^{6} i\mathbb{P}(X=i)=\frac{1}{6}\sum_{i=1}^{6}=3.$$

5. $$\mathbb{E}(X|X\geq 4)=\sum_{i=1}^{6} i\mathbb{P}(X=i|X\geq 4)=5.$$

The exponential distribution

1. $$\mathbb{P}(X \geq t) = 1 - \mathbb{P}(X < t) = e^{-\lambda t}.$$

2. $$\mathbb{P}(1 < X < 2) = \mathbb{P}(X < 2) - \mathbb{P}(X < 1) = e^{-\lambda} - e^{-2\lambda}.$$

3. $$\mathbb{E}(X) = \frac{1}{\lambda}.$$

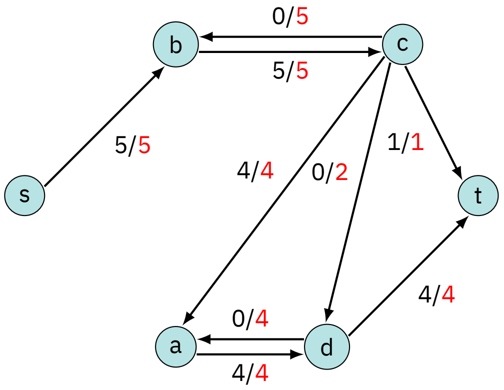
4. $$\mathbb{P}(X > t + u \text{ and } X > t) = \mathbb{P}(X > t + u),$$

   because if $X > t + u$ then it will also happen that $X > t$. The rest follows by doing one more computation.

5.2. Graph Theory

Maximum flows

1. The following is a maximum flow:



   More answers are correct. As long as the flow has value 5 it is maximum. Of course, flow conservation must hold and arc capacities may not be exceeded.
2. Observe that any flow from s to t must pass through the arcs (c, t) and/or (d, t). So the value of any $s - t$ flow can be no more than the sum of the capacities of these arcs. This sum is 5. We have found a flow of value 5. Hence it must be a maximum flow.

Value of a flow

1. $f_{(a,b)} = f_{(s,a)}$
2. $f_{(b,t)} = f_{(a,b)} = f_{(s,a)} = 1$. The flow going into t is 1.
3. Add the flow conservation equation for all the vertices except s and t. Observe that any arc a between two vertices (that are both not s and t) appears once as an outgoing arc, and once as an ingoing arc. Hence, the value f_a will cancel out, as it is added exactly once, and subtracted once. What remains are the arcs that include s, which we assumed were only outgoing arcs, and the arcs including t, which we assumed were only ingoing arcs, so they are subtracted. The equality follows from the fact that one minus the other should be zero.

5.3. Selfish routing

Exercise 1

1. For the flow $(\frac{1}{3}, \frac{2}{3})$, the social cost is

$$\begin{aligned} SC\left(\frac{1}{3}, \frac{2}{3}\right) &= \ell_{P_1}\left(\frac{1}{3}, \frac{2}{3}\right) \cdot \frac{1}{3} + \ell_{P_2}\left(\frac{1}{3}, \frac{2}{3}\right) \cdot \frac{2}{3} \\ &= \left(\ell_{(s,u)}\left(\frac{1}{3}\right) + \ell_{(u,t)}\left(\frac{1}{3}\right)\right) \cdot \frac{1}{3} + \left(\ell_{(s,v)}\left(\frac{2}{3}\right) + \ell_{(v,t)}\left(\frac{2}{3}\right)\right) \cdot \frac{2}{3} \\ &= \left(\frac{1}{9} + 1\right) \cdot \frac{1}{3} + \left(2 + \frac{2}{3}\right) \cdot \frac{2}{3} \\ &= \frac{58}{27} \approx 2.148. \end{aligned}$$

2. For the flow $(0, 1)$, the social cost is

$$SC(0,1) = \ell_{P_1}(0,1) \cdot 0 + \ell_{P_2}(0,1) \cdot 1 = \left(\ell_{(s,v)}(1) + \ell_{(v,t)}(1)\right) \cdot 1 = (2+1) \cdot 1 = 3.$$

3. Given an $x \in [0,1]$, the social cost of the flow $(x, 1-x)$ is

$$\begin{aligned} SC(x, 1-x) &= \ell_{P_1}(x, 1-x) \cdot x + \ell_{P_2}(x, 1-x) \cdot (1-x) \\ &= \left(\ell_{(s,u)}(x) + \ell_{(u,t)}(x)\right) \cdot x + \left(\ell_{(s,v)}(1-x) + \ell_{(v,t)}(1-x)\right) \cdot (1-x) \\ &= \left(x^2 + 1\right) \cdot x + (2 + 1 - x) \cdot (1-x) = x^3 + x^2 - 3x + 3. \end{aligned}$$

4. To find the optimal flow f^*, we need to *minimize* the function $g(x) = SC(x, 1-x) = x^3 + x^2 - 3x + 3$ for $x \in [0,1]$.

   We have that $g'(x) = 3x^2 + 2x - 3 = \frac{(3x+1)^2 - 10}{3}$ and $g''(x) = 6x + 2 > 0$ for all $x \in [0,1]$. Therefore, to find the minimum of the function $g(x)$, we need to find the point $x^* \in [0,1]$ such that $g'(x^*) = 0$. This equation has two solutions, namely

$x_1^* = \frac{\sqrt{10}-1}{3}$ and $x_2^* = -\frac{\sqrt{10}-1}{3}$. Since, we are interested in the interval $[0,1]$, we keep the first solution and conclude that that the optimal flow f^* is $\left(\frac{\sqrt{10}-1}{3}, 1-\frac{\sqrt{10}-1}{3}\right)$. Computing the social cost of f^* can now be done via the definition.

Exercise 2: The Nash flow is $(1,0)$ since $\ell_{P_1}(1) = 1 \leq \ell_{P_2}(0)$ for all three functions. To compute the optimal flow we work as follows:

1. When $\ell_{P_1}(x) = \frac{x^2+x}{2}$, the social cost of $(x, 1-x)$ for $x \in [0,1]$ is

$$SC(x, 1-x) = \ell_{P_1}(x) \cdot x + \ell_{P_2}(x) \cdot (1-x) = \frac{x^3+x^2}{2} + 1 - x.$$

Consider the function $g(x) = \frac{x^3+x^2}{2} + 1 - x$. We have that $g'(x) = \frac{3}{2}x^2 + x - 1$ and $g''(x) = 3x + 1 > 0$ for all $x \in [0,1]$. We need to find an $x^* \in [0,1]$ so that $g'(x^*) = 0$ to find the minimum. This is true for $x^* = \frac{\sqrt{7}-1}{3} \approx 0.549$. Therefore, the optimal flow f^* is $\left(\frac{\sqrt{7}-1}{3}, 1-\frac{\sqrt{7}-1}{3}\right)$.

2. When $\ell_{P_1}(x) = x^3$, the social cost of $(x, 1-x)$ for $x \in [0,1]$ is

$$SC(x, 1-x) = \ell_{P_1}(x) \cdot x + \ell_{P_2}(x) \cdot (1-x) = x^4 + 1 - x.$$

Consider the function $g(x) = x^4 + 1 - x$. We have that $g'(x) = 4x^3 - 1$ and $g''(x) = 12x \geq 0$ for all $x \in [0,1]$. We need to find an $x^* \in [0,1]$ so that $g'(x^*) = 0$ to find the minimum. This is true for $x^* = \sqrt[3]{\frac{1}{4}} \approx 0.63$. Therefore, the optimal flow f^* is $\left(\sqrt[3]{\frac{1}{4}}\right)$.

3. By generalizing the derivation above, we have that

$$SC(x, 1-x) = \ell_{P_1}(x) \cdot x + \ell_{P_2}(x) \cdot (1-x) = x^{d+1} + 1 - x.$$

Consider the function $g(x) = x^{d+1} + 1 - x$. Its derivative is $g'(x) = (d+1)x^d -$ and it holds that $g''(x) = d(d+1)x^{d-1} \geq 0$. By solving for an $x^* \in [0,1]$ with $g'(x^*) = 0$ we obtain that $x^* = \sqrt[d]{\frac{1}{d+1}}$. Thus, the optimal flow f^* is $\left(\sqrt[d]{\frac{1}{d+1}}, 1 - \sqrt[d]{\frac{1}{d+1}}\right)$.

Exercise 3: Recall that in order to have the flow $(\frac{1}{2}, \frac{1}{2})$ be a Nash flow, $\tau_1 \geq 0, \tau_2 \geq 0$ need to be such that $\tau_1 = \frac{1}{2\gamma} + \tau_2$ holds for every sensitivity parameter γ. To find the pair which minimizes the money the government gets from drivers, we need to find a pair (τ_1^*, τ_2^*) which minimize $\tau_1^* + \tau_2^* = \frac{1}{2\gamma} + 2\tau_2^*$. It is easy to see that this function is minimized for $\tau_2^* = 0$. Therefore, the tolls that minimize the moeny the goverment gets are $\tau_1^* = \frac{1}{2\gamma}$ and $\tau_2^* = 0$.

Exercise 4:

1. The social cost of the flow $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ is:

$$\begin{aligned}
SC\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) &= \ell_{P_1}\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \cdot \frac{1}{3} + \ell_{P_2}\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \cdot \frac{1}{3} + \ell_{P_3}\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)\frac{1}{3} \\
&= \frac{1}{3} \cdot \left(\left(\ell_{(s,u)}\left(\frac{2}{3}\right) + \ell_{(u,t)}\left(\frac{1}{3}\right)\right) + \left(\ell_{(s,v)}\left(\frac{1}{3}\right) + \ell_{(v,t)}\left(\frac{2}{3}\right)\right)\right) \\
&\quad + \left(\ell_{(s,u)}\left(\frac{2}{3}\right) + \ell_{(u,v)}\left(\frac{1}{3}\right)\ell_{(v,t)}\left(\frac{2}{3}\right)\right) \\
&= \frac{1}{3} \cdot \left(\frac{2}{3} + 1 + 1 + \frac{2}{3} + \frac{2}{3} + 0 + \frac{2}{3}\right) = \frac{14}{9} \approx 1.556.
\end{aligned}$$

2. To guarantee that $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ is a Nash flow, the cost for each of the three paths P_1, P_2, P_3 must be equal (see Definition of Nash flow). Observe that this is already for paths P_1 and P_2 as they do not contain the edge $u \rightarrow v$ and, therefore, do not depend on the toll τ we impose. Therefore, it suffices to guarantee that the cost of P_3 equals the cost on the other two paths. In particular, τ must be such that

$$c_{(s,u)}\left(\frac{2}{3}\right) + c_{(u,t)}\left(\frac{1}{3}\right) = c_{(s,u)}\left(\frac{2}{3}\right) + c_{(u,v)}\left(\frac{1}{3}\right) + c_{(t,u)}\left(\frac{2}{3}\right),$$

which is equivalent to

$$c_{(u,v)}\left(\frac{1}{3}\right) = c_{(u,t)}\left(\frac{1}{3}\right) - c_{(t,u)}\left(\frac{2}{3}\right) = 1 - \frac{2}{3} = \frac{1}{3}.$$

By observing that $c_{(u,v)}\left(\frac{1}{3}\right) = \gamma\tau$ we obtain that

$$\tau = \frac{1}{3\gamma}.$$

5.4. Queueing theory

Mean queue length

1.

$$\sum_{i=0}^{\infty} p_i = \sum_{i=0}^{\infty}(1-\rho)\rho^i = (1-\rho)\sum_{i=0}^{\infty}\rho^i.$$

Geometric sum

For the geometric sum we have that

$$\sum_{i=0}^{n} \omega^i = \frac{1-\omega^{n+1}}{1-\omega}.$$

Hence we have that

$$\sum_{i=0}^{\infty} \omega^i = \lim_{n\to\infty} \sum_{i=0}^{n} \omega^i = \lim_{n\to\infty} \left(\frac{1-\omega^{n+1}}{1-\omega}\right),$$

and hence for $\omega \in (0,1)$

$$\sum_{i=0}^{\infty} \omega^i = \frac{1}{1-\omega}.$$

Using this result the answer follows.

2.

$$\sum_{i=0}^{\infty} i(1-\rho)\rho^i = \sum_{i=1}^{\infty} i(1-\rho)\rho^i = \rho(1-\rho)\sum_{i=1}^{\infty} i\rho^{i-1}$$
$$= \rho(1-\rho)\left(\sum_{i=0}^{\infty}\rho^i\right)' = \rho(1-\rho)\left(\frac{1}{1-\rho}\right)' = \frac{\rho}{1-\rho}.$$

$M|M|1|k$

1.

$$\sum_{i=0}^{k} p_i = \sum_{i=0}^{k} p_0\,\rho^i = 1$$

Thus

$$p_0 = \frac{1-\rho}{1-\rho^{k+1}} \qquad p_k = \frac{(1-\rho)\rho^k}{1-\rho^{k+1}}$$

2. $\lambda(1-p_k)$
3. $\mu(1-p_0)$
4. $1-p_0 < \rho$
5.

$$\mathbb{E}[L] = \frac{\rho}{1-\rho} - \frac{(k+1)\rho^{k+1}}{1-\rho^{k+1}}.$$

Optimization

1. Consider the cases when $\rho < 0$ and $\rho > 0$.

2.

$$(n+1)(1-\rho) - \rho(1-\rho^{n+1}) - n(1-\rho) - \rho(1-\rho^{n}) = (1-\rho)(1-\rho^{n+1}) > 0$$

Hint

Consider the value and derivative of $(v_0\,(1-\rho) - \rho(1-\rho^{v_0}))(1-\rho)^{-2}$ when $v_0 = 1$.

3.