UNIVERSITÉ LIBRE DE BRUXELLES

ÉCOLE POLYTECHNIQUE DE BRUXELLES

Business Process Analysis

Management of Data Science and Business Workflows

Fall 2022

Authors:
ABUSALEH, Ali
ULB ID: 000558778, *email*: ali.abusaleh@ulb.be
Lorencio Abril, Jose Antonio
ULB ID: 000559831, *email*: jose.lorencio.abril@ulb.be

Professor: **Sacharidis, Dimitrios**

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Business Process Analysis

Business process analysis[1] is the art and science of analyzing a process, with the aim of understanding it and being able to improve it later.

It is an **art** in the sense that a process is by nature one possibility among potentially infinity of them. Thus, having the wit to detect a step that is not really necessary, or to come up with a totally different process which achieves the same results with less resources is not something that follows a fixed set of rules. This part of process analysis is the **qualitative analysis**, and only some principles and techniques can be defined to help us in a wide range of usual situations.

In this assignment, we are focusing in the other side of the coin, the **science** side of business process analysis, i.e., **quantitative analysis**. This term encapsulates a set of techniques for analyzing business processes quantitatively in terms of process performance measures such as cycle time, waiting time, cost,... We have seen three approaches that are supported by solid mathematical theories:

- **Flow analysis**: it is a family of techniques to estimate the overall performance of a process given some knowledge about the performance of its task. Basically, statistical data for each task have to measured and analyzed, in order to obtain information of its cycle time, cost,... Then, applying different operators depending on the situation, we can approximate the total cost of the process. Important measures are **cycle time**, **cycle time efficiency** and **cost**.

- **Queues**: collection of mathematical techniques to analyze systems that have resource contention. In this case, we are focusing in the analysis of how well our process is fulfilling the requests that arrive to it. We have studied two basic models, which can be in fact encapsulated into just 1, being it the ***M/M/c* model**, which models a single-queue system, where the inter-arrival times of requests follow an exponential distribution (the first **M**), the processing times follow an exponential distribution (the second **M**) and there are c servers attending the requests in a FIFO (first-in-first-out) manner. Here, we encounter important terms such as the **mean arrival time**, λ, the **theoretical capacity per server**, μ, the **resource utilization**, ρ, the **average number of instances in the queue** is L_q, the **average time one instance spends in the queue** is W_q, W is the **average time one instance spends in the whole system** and L is the **average number of instances in the system**.

[1] Refer to [Dum+18].

Chapter 2

Solutions

2.1 Question 1

2.1.1 Overview

In this question we are asked to compute the cycle time efficiency and the cost per execution of a given process.

2.1.2 Solution

In Figure 2.1 we are showing the diagram for the first exercise, but we have annotated it with the waiting and processing times, and the cost associated to each of the tasks.

As can be seen, we have computed the cost per activity, by multiplying the processing time of each activity and the cost of the resource that performs it.

First, we are going to compute the cycle time efficiency. For this, we need both the cycle time and the processing time.

For the **cycle time** we have used all times involved in the process, both waiting and processing times. The most simplified version of the process can be seen in Figure 2.2. According to the form of the simplified model, we know that the cycle time is

$$CT = T(A) + 0.2 \cdot T(B1) + 0.8 \cdot T(B2),$$

and we know $T(A)$ and $T(B1)$ directly from the original model, so we obtain:

$$CT = (60 + 10)\ min + 0.2 \cdot 5\ min + 0.8 \cdot T(B2).$$

But $T(B2)$ requires a little bit more analysis. We have decompose it as in Figure 2.3. So, as we saw in class[1], we have

$$T(B2) = T(C) + \frac{T(D) + 0.2 \cdot T(E)}{0.8}.$$

$T(C)$ is the whole time for the first activity done by Level2, $T(D)$ is the whole time for all activities

[1]Lab session 5, exercise 4.

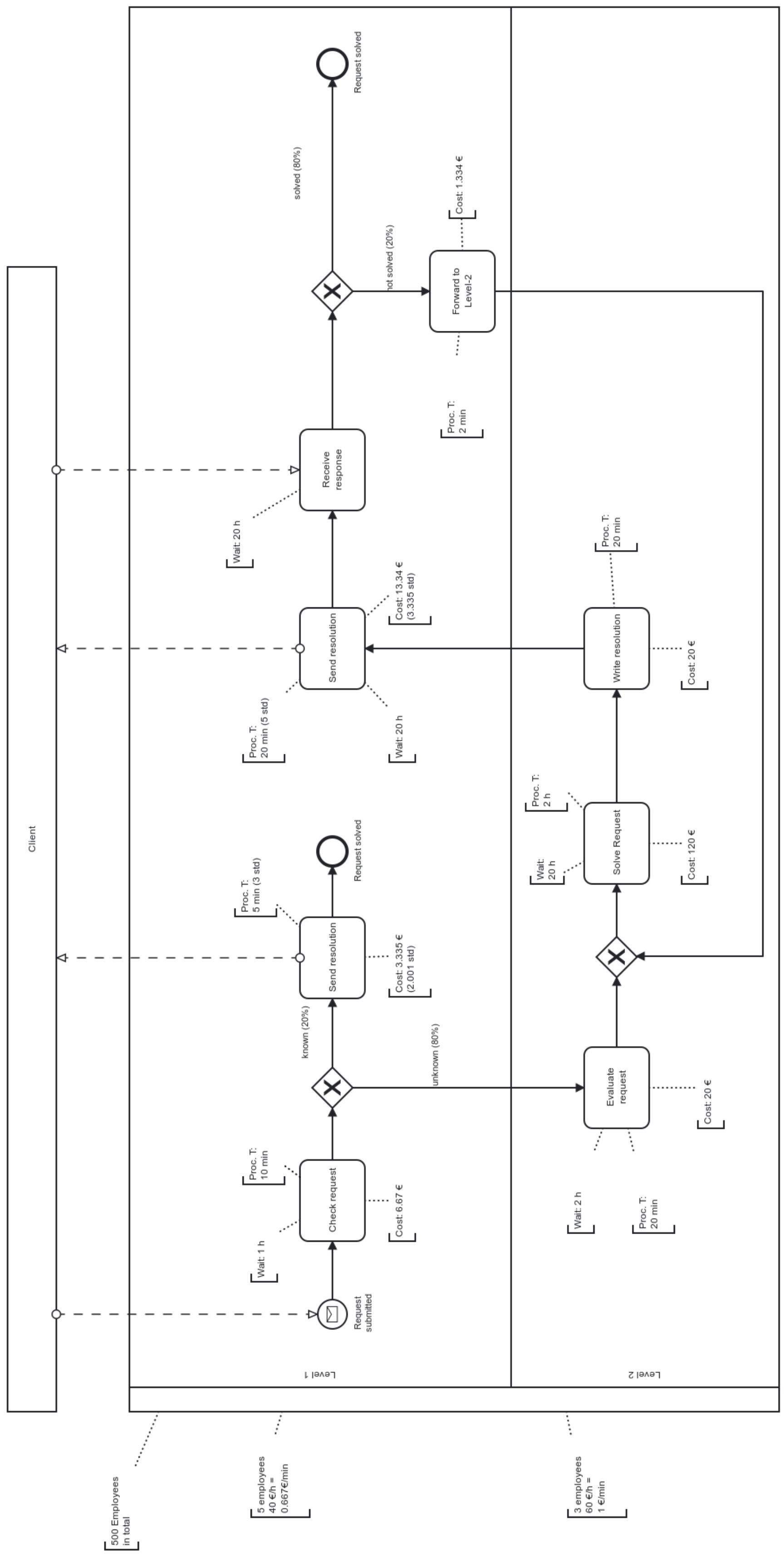

Figure 2.1: Question 1 BPMN model annotated.

inside the loop, except the activity *Forward to Level2*, whose time corresponds to $T(E)$. Thus, we get

$$\begin{aligned} T(B2) =& (120+20)\ min + \frac{(1200+120+20+1200+20+1200)\ min + 0.2 \cdot 2min}{0.8} \\ =& 140\ min + \frac{3760\ min + 0.4\ min}{0.8} \\ =& 140\ min + 4700.5\ min \\ =& 4840.5\ min. \end{aligned}$$

Assembling all this information, we finally obtain

$$\begin{aligned} CT =& 70\ min + 1\ min + 0.8 \cdot 4840.5\ min \\ =& 3943.4\ min. \end{aligned}$$

For the **processing time**, we repeat the same process, but this time we don't take into account the waiting times. Thus, we get:

$$PT = 10\ min + 1\ min + 0.8 \cdot PT(B2).$$

Let's tackle $PT(B2)$:

$$\begin{aligned} PT(B2) =& 20\ min + \frac{(120+20+20)\ min + 0.2 \cdot 2\ min}{0.8} \\ =& 20\ min \frac{160.4\ min}{0.8} \\ =& 220.5\ min. \end{aligned}$$

So,

$$PT = 187.4\ min.$$

Now, we can compute the **cycle time efficiency**, as

$$CTE = \frac{PT}{CT} = \frac{187.4}{3943.4} = 0.0475,$$

or, as a percentage, 4.75%. This means that, in this process, for every minute of work needed to complete a request by a client, this one has to wait for 21 minutes, approximately[2]. For example, if a task needs 3h to be completed, the client would need to wait for approximately 63 hours for his task to be completed once it started.

For computing the **Cost**, we do exactly the same computations as for the Processing Time, but we take into account the cost of each task, as shown in Figure 2.1. This way, we get:

$$C = 6.67\ euro + 0.2 \cdot 3.335\ euro + 0.8 \cdot C(B2).$$

For $C(B2)$, we get

$$\begin{aligned} C(B2) =& 20\ euro + \frac{(120+20+13.34)\ euro + 0.2 \cdot 1.334\ euro}{0.8} \\ =& 20\ euro + \frac{153.6068\ euro}{0.8} \\ =& 212.0085\ euro. \end{aligned}$$

So, all in all:

$$C = 176.9438\ euro.$$

[2]This is obtained by the inverse of CTE, and tells us how many units of time we have to wait for each processing unit of time needed for the process.

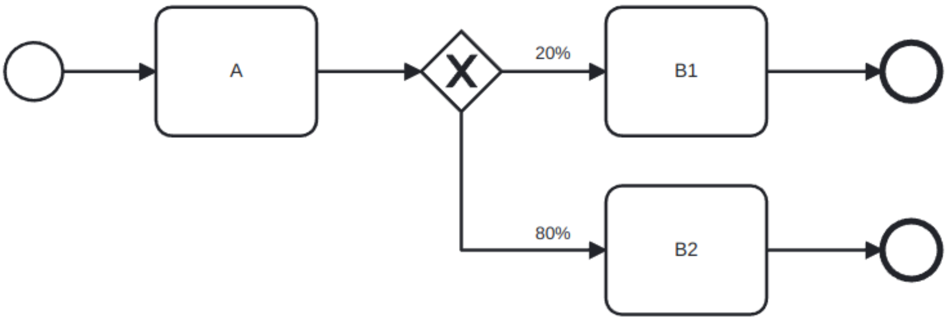

Figure 2.2: Simplified process for Question 1.

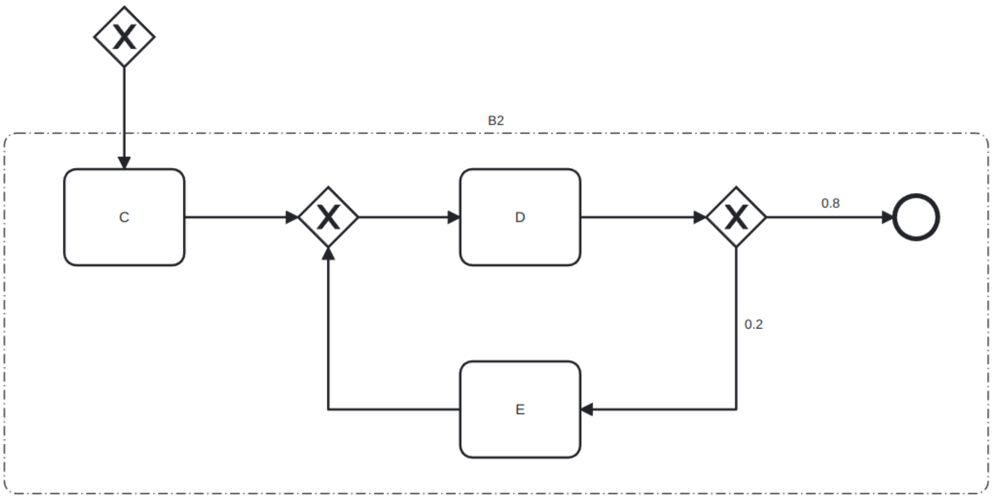

Figure 2.3: $B2$ decomposed.

2.2 Question 2

2.2.1 Overview

Now we are asked to compute how many Level-2 staff members are required to ensure a mean waiting time of less than two hours for those requests reaching Level2 and assuming these come at a rate of one per hour.

2.2.2 Solution

In this case we have to obtain the processing time for which Level-2 staff members account. To do this, we can use the same model as in Figure 2.3, but disregarding those activities done by Level-1 staff members. So, we obtain a unit load, u_l of

$$u_l = 20\ min + \frac{(120+20)\ min}{0.8} = 195\ min/instance = 3.25\ h/instance.$$

This means that Level-2 members are able to process

$$\mu = \frac{1}{u_l} = 0.308\ instance/h.$$

μ is called the **theoretical capacity per server**. On the other hand, the statement of the problem tell us that the **mean arrival rate** is $\lambda = 1\ instance/h$.

This is a **M/M/c queue model**, in which we want to compute c so that the average time spent in the queue, W_q, is less than 2 hours. To do this, we have developed a simple python script as shown in Listing 2.1.

```
def fact(n): #simple factorial implementation
    if n == 0 or n == 1:
        return 1
    return n*fact(n-1)

def Lq(c, l, m, r): #computation of Lq
    num = (l/m)**c*r
    div = fact(c)*(1-r)**2
    s = 0
    for i in range(c):
        s = s + (l/m)**i/fact(i)
    div = div*((l/m)**c/(fact(c)*(1-r)) + s)
    return num/div

def Wq(c, l, m, r): #computation of W_q
    return Lq(c, l, m, r)/l

l = 1, m = 0.308, c = 3 #lambda, mu and c

print("start")

wq = 5
while wq >= 2: #stop when W_q meets the requirements of teh exercise
    c = c + 1
    r = l/(c*m)
    wq = Wq(c,l,m,r)

print("Finished! Solution is:")
print("c="+str(c)+", Wq="+str(wq))
```

Listing 2.1: compute_wq.py

As can be seen, the logic is to increase c until the computed W_q is less than 2 hours. Note that the first value used for computation is 4, because the resource utilization factor is $\rho = \frac{\lambda}{c\mu} = \frac{3.247}{c}$, and this value has to be less than 1, so $c \geq 4$.

When we run the script, we obtain the result presented in Listing 2.2, which tells us that we need 5 Level-2 staff members to fulfill all the requirements of the exercise and that the average waiting time for the instances reaching Level-2 will be 0.559 hours.

```
Finished! Solution is:
c=5, Wq=0.5585783349206068
```

Listing 2.2: Result of the script.

2.3 Question 3

2.3.1 Overview

In this exercise we have to simulate the process, assuming a mean of 4 request submitted per hour and ignoring all internal waiting times except the waiting time for receiving the response from the client. The resources can be modified to gain insights of the process.

2.3.2 Solution

First we need to calculate the inter arrival time, which can be calculated using the following equation

$$Arrival - rate = \frac{1}{\lambda} = \frac{1}{4} = 0.25 \cdot 60 = 15\ minutes$$

For simulating the waiting time from the client, we created an artificial resource on the simulator with cost of 0 Euro/Hour.
The Resources setting was as in Table 2.1.

Scenario	Resource Name	Number or Resources	Cost Per hour
1	Level 1	5	40
	Level 2	3	60
	Artificial	100	0
2	Level 1	3	40
	Level 2	2	60
	Artificial	100	0
3	Level 1	10	40
	Level 2	6	60
	Artificial	100	0

Table 2.1: Resources setup for each scenario.

The settings for each task was as in Table 2.2.

Task Name	Resource	Duration distribution	Value (Minutes)
Check Request	Level 1	Exponential	10
Send Resolution[3]	Level 1	Normal	5 (std. 3)
Send Resolution[4]	Level 1	Normal	20 (std. 5)
Receive Response	Artificial	Fixed	20
Forward to Level-2	Level 1	Fixed	2
Evaluate Request	Level 2	Exponential	20
Solve Request	Level 2	Exponential	2
Write Resolution	Level 2	Exponential	20

Table 2.2: Tasks setup.

And the setting for the Probability of XOR gateways were as in Table 2.3:

We are going to run the three scenarios with a total number of cases of 1000, so we can see the behavior of the process in the long term.

XOR Gateway	First Option	Second Option
Known Request	Known (20%)	Unknown (80%)
Client Response	Solved (80%)	Not Solved (20%)

Table 2.3: XOR gateways setup.

Scenario 1

For the first scenario, we should get similar results to the theoretical ones obtained (but we will not get exactly the same because we are not modeling all the waiting times involved in the process).

The results are summarized in Figure 2.4. We can observe that the cycle time is on average 2.06 weeks. This is 4944 minutes, which is higher than the value we obtained theoretically, 3943.4 minutes. This is probably due to queue waiting times, which are not taken into account in the theoretical computations. For instance, in Figure 2.5 we can see that the Level2 employees are very saturated, which makes the process last longer than it should. We can see also that the waiting time is in average 1.93 weeks, which is too long.

Regarding the cost, we can see that it is pretty near to the one we computed before. It is important to notice that the cost is only dependent on the processing time, so we should see similar results for all the scenarios.

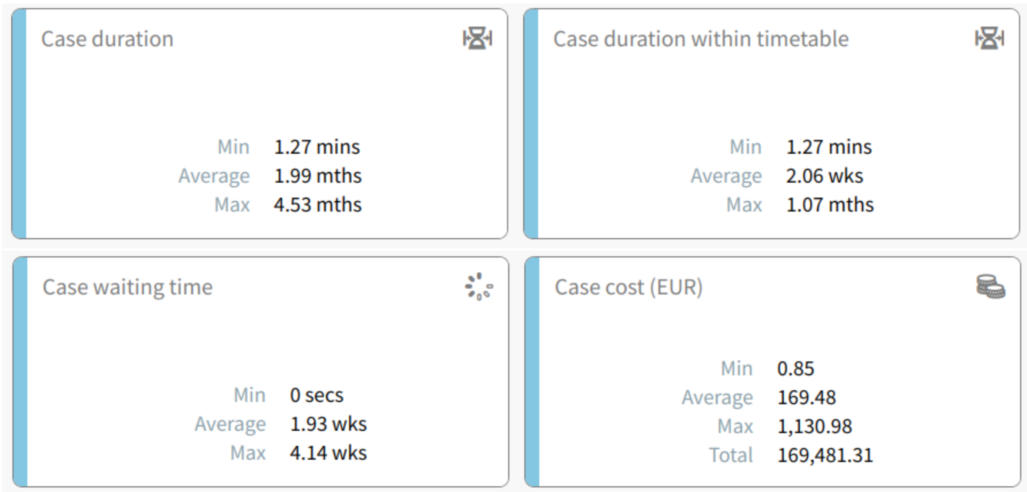

Figure 2.4: Results for the simulation of scenario 1.

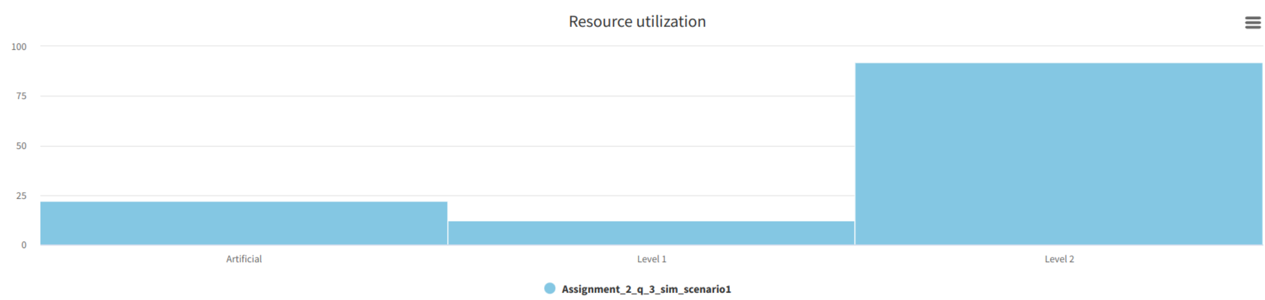

Figure 2.5: Resource utilization for the simulation of scenario 1.

Scenario 2

In this scenario we are reducing the available resources, so we expect to see an increase in the cycle time and in the waiting time. On the other hand, the cost should remain with a similar value. This corresponds exactly to the obtained results, depicted in Figure 2.6, where we can see how the case duration has almost double and the waiting time has more than doubled. The cost remains the same. It is curious that the resource utilization of Level2 has decreased a little bit, as can be seen in Figure 2.7.

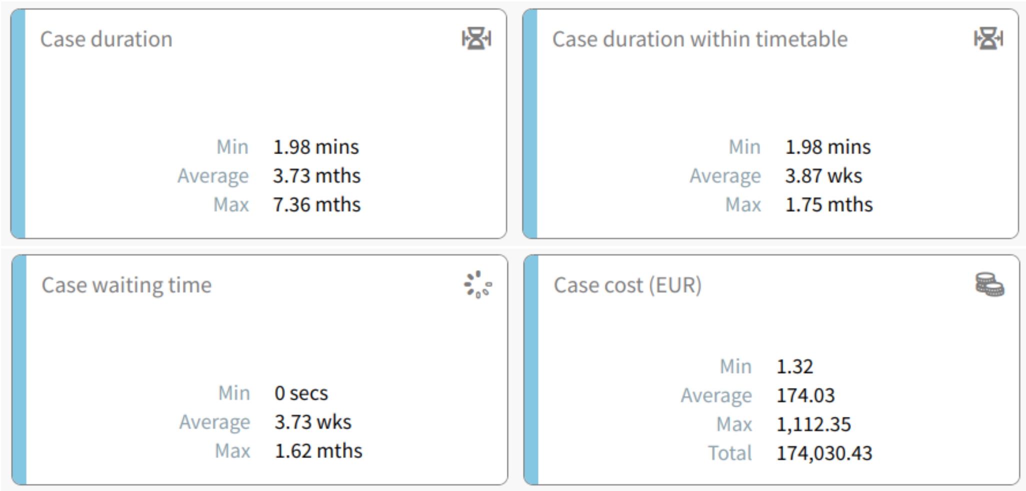

Figure 2.6: Results for the simulation of scenario 2.

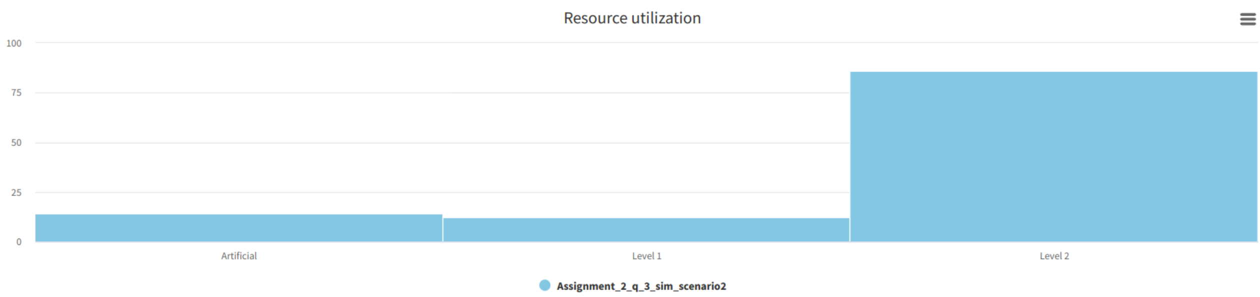

Figure 2.7: Resource utilization for the simulation of scenario 2.

Scenario 3

To finish the analysis, let's see what happens when we increased our available resources. We expect to reduce the case duration and the waiting times.

The results are shown in Figure 2.8, where we observe a dramatic decrease in the case duration, which goes much more in the line of the theoretical results obtained, because it is much less than the theoretical cycle time, which takes into account many more waiting times. We see also how the case waiting time has decreased yo 3.85 days and the cost remains in around the same value, as expected.

In Figure 2.9 we can see how the resource utilization for Level2 is still high, but has decreased significantly. This is exactly what makes the process decrease a lot in terms of case duration. Thus, we can conclude that the bottleneck of this process are the Level2 staff employees, because they are saturated with work.

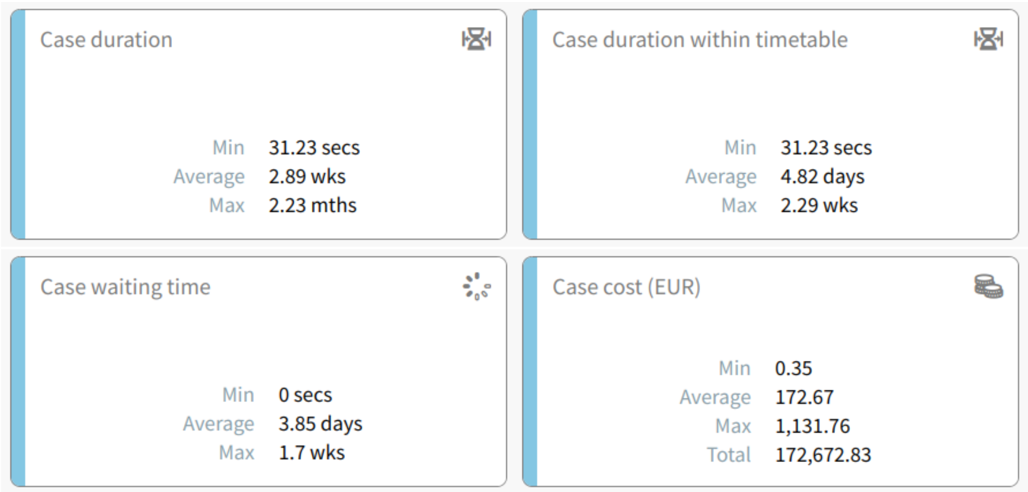

Figure 2.8: Results for the simulation of scenario 3.

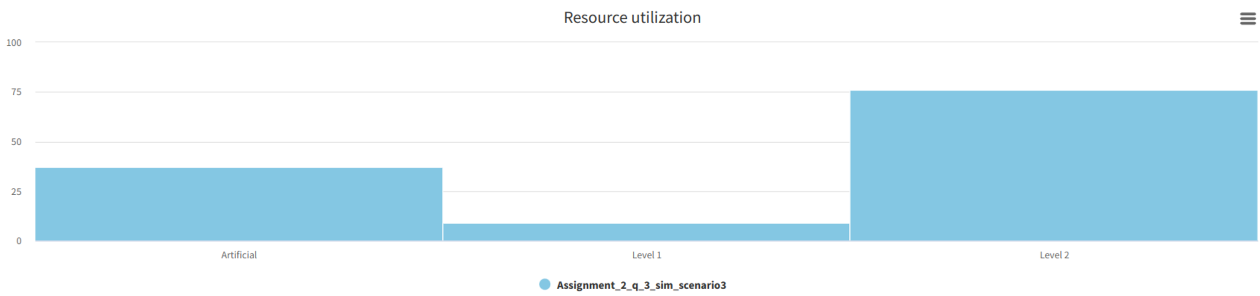

Figure 2.9: Resource utilization for the simulation of scenario 3.

Bibliography

[Dum+18] Marlon Dumas et al. *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, 2018. DOI: 10.1007/978-3-662-56509-4.