University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Simulation of Healthcare Processes: Challenges, Solutions, and Benefits

Supervisor
Prof. Massimiliano De Leoni
University of Padova

Master Candidate
Marco Francesco Sommaruga

Student ID
1225007

Academic Year
2021-2022

Abstract

The emergency department of a hospital plays a key role in the incoming patient management. Therefore, it is crucial to ensure an adequate level of organization and efficiency. In this project, we aim to analyze and improve the processes at emergency departments using business process simulation. Through simulation experiments, various 'what-if' scenarios can be tested, and redesigning alternatives can be compared with respect to some key performance indicators. The input for business process simulation is a process model extended with additional information for a probabilistic characterization of the different run-time aspects (case arrival rate, task durations, routing probabilities, roles, etc.). It is thus critical that the business simulation model is accurate so as to ensure that the simulations and the various 'what-if' scenarios reflect credible alternatives with realistic outcomes.

This project aims to create emergency-department simulation models on the basis of the actual executions that are recorded in so-called event logs, which are typically extracted from the information systems that support the execution of processes at the hospital. The process model and the simulation parameters are extracted using different techniques from the field of Process Mining, which builds on an analysis of the event logs and aims to gain insights into how processes are actually carried out. In particular, this project has focused on the analysis of the emergency department at a hospital in Tuscany.

The quality of the extracted logs greatly influences the analysis that can be carried out. For example, to tackle the reduction of waiting times for patients and to optimize the resources, two of the most critical emergency-department challenges, it is necessary that the data report when each activity has started and completed, and how hospital staff participated to the execution of each activity. Unfortunately, the event logs extracted from the information systems of the Tuscany's hospital missed relevant information, including the timestamps when activity started, thus diminishing the realism of the business simulation model.

To overtake this issue, the project extended a previous technique to estimate the missing timestamps of when the activities started, and overtook some of its limitations.

The project assessed the extended technique on the emergency department of the Tuscany's hospital, and has shown on this case study how healthcare can leverage on process mining to simulate different 'what-if' scenarios, on the basis of which decisions can be made on how to improve real processes.

Contents

Listing of figures

Listing of tables

1 Introduction

The Italian health spending estimation in 2022 amounts to € 131 billion, 3% more than 2021, and an annual decrease of 0.6% is expected for the next 3 years, while SDG would averagely increase by 3.8% [1]. The tightening of government budgets is just one of the several challenges healthcare managers have to face while dealing with the increasing and ageing population, the technological progress, and the scarce resource availability. In order to handle these restrictions, healthcare managers are continuously exploring improvement opportunities to ensure high service quality to patients and to reduce inefficiencies, particularly in terms of waiting times [2].

Nowadays, the constant growth of information stored within information systems allows the collection of huge amounts of data that can be exploited to provide realistic and reliable insights on processes and to make decisions. In particular, these data enable the application of process simulation in identifying areas of improvement through the investigation of different plausible scenarios. This work presents the potentiality of process simulation in healthcare by studying the emergency department (ED) of a Tuscany hospital.

The ED represents the main gateway of a hospital, and as such has to be efficient both in terms of waiting times and resource organization.

One of the main challenges in creating a simulated process consists in dealing with poor data quality. Real-life healthcare event logs frequently suffer from a multitude of data quality issues such as missing events, incorrect timestamps and incorrect resource information [3]. Within the specific case study of the Tuscany's ED, the challenge has arisen by the fact that the supporting information system did not record the starting of activities, and hence the event log missed the corresponding start timestamp. This has initially prevented the possibility to easily compute activity durations. To develop a simulated process it is crucial to know the activity durations; hence, to address the missing start timestamps problem, a novel technique that allows their estimation has been exploited and enhanced. This method is based on the optimization of a parametric function exploited to compute the start timestamp with the highest possible accuracy. In this work, we propose an alternative approach with genetic algorithms improving the results obtained. Before applying it to discover the missing start timestamps for the ED case study, we validate it on two case studies for which we already had the start timestamps: we removed the events with

them and rediscovered, thus assessing the accuracy of their rediscovery. The results show a significant accuracy improvement with respect to the state of the art.

Another challenge of process simulation consists in the construction of a fitting process model. In this case, the main issue stands in the random nature of the emergency department, which provides medical treatments to unscheduled patients who present different types of pathologies. As a consequence, a wide variety of different paths can be performed accordingly to the type of diagnosis. In this research, we both interacted with domain experts and relied on data to find a model that accurately fits the process.

The great advantage of building an accurate simulation process stands in its exploitability in the development of 'what-if' scenarios that faithfully represent plausible alternatives to the original process. This approach is particularly crucial in healthcare systems, since it allows health managers to simulate how a strategic decision would modify the process without endangering patients' health and without incurring in high expenses to create the process in real-life.

The complexity of the healthcare processes and the limitations of simulation tools do not always allow the development of process simulation models with every policy and characteristic that we aim for, thus it is necessary to consider some simplifications and assumptions. In this work, we assume that all the patients arriving to the ED are characterized by the same degree of urgency and priority.

In line with the possibilities offered by simulation software and with the problems faced by healthcare managers, we developed simulation scenarios that aim to suggest possible implementations to solve them.

In particular, we propose a first 'what-if' scenario for the creation of a pediatric fast-track as a solution for the overcrowding emergency department. This modification to the ED structure allows a decrease in the median case durations of 30% with respect to the current one.

The second 'what-if' scenario focuses on identifying the best resource configurations in terms of waiting times and cost reduction, by solving a multi-objective optimization problem. From the solutions found by the optimization, we selected two new resource settings: the first one cuts the total costs by 16% with respect to the current one, maintaining similar case durations, whereas the second resource configuration allows a reduction of the median case durations of 55% with an expense comparable to the original one.

1.0.1 Document Outline

The present document is organized as follows:

- **Chapter 2**: Preliminaries

  In this chapter, we provide some important basic notions of the concepts elaborated in this work.

- **Chapter 3**: Related Works

  This chapter introduces some of the studies conducted on healthcare processes and the techniques exploited to face emerging challenges.

- **Chapter 4**: Estimation Of Start Timestamps: A Genetic Optimization Approach

  This chapter presents an enhancement of a novel technique for the estimation of the activities' start timestamps. The improvement is then validated by comparing the existing strategy and the new one in two case studies containing both start and completion events. To this aim, the original start timestamps have been removed to assess the accuracy of the estimated ones.

- **Chapter 5**: Case Study

  This chapter exhibits a case study on a Tuscany's emergency department. There are shown the challenges and the necessary steps to overcome them and to create an accurate simulated process. In particular, the validated technique in chapter 4 is leveraged to find the missing starting timestamps.

  Finally, two 'what-if' scenarios are presented with the aim of offering plausible solution approaches to the main issues experienced in healthcare.

- **Chapter 6**: Conclusion And Future Works

  In this final chapter, we summarize our results and propose some future research directions.

2

Preliminaries

2.1 Process mining

Process mining is a recent research field of data science related to the study of processes. The data at the basis of this discipline are represented by sequentially recorded events, namely event logs, each referring to a specific case's activity. Each event log may contain details about the activity that is performed: whenever it is possible, process mining techniques use extra information such as the resource executing or initiating the activity, the timestamp of the event, or the data elements recorded with the event [4]. The data collected can be elaborated through process mining methods with the aim to perform discovery, conformance, and enhancement. Discovery techniques are exploited to produce a model given an event log without any *a-priori* information. The process model is built automatically from the data through specific algorithms. The found graph is derived solely from the available data and it shows how the process is run in reality. Conformance methods permits to compare an existing model (either discovered or made by hand) with the event log collected from the same process. The goal of conformance checking is to detect where the given model and the event logs show different behaviors, in order to spot and explain those deviations. Enhancement allows to enrich and improve an existing model with the information collected in the event log. The goal is to modify the model to better represent the reality. There can be added information about resources, priorities in execution, quality metrics, etc. This technique is also used to repair a model that shows a specific sequence of activities that is not sequentially followed in reality. Furthermore, with process mining it is possible to focus on different data perspective. In particular there can be identified the control-flow perspective, which goal is to find the activities' sequences that characterize the paths followed by each case, the organizational perspective, that studies the information related to the resources involved in the activities, the case perspective, associated to the study of case's properties that can enrich the information about the process, and the time perspective, which is concerned with the timing and frequency of events.

	case	C02_IDUNIVOCOASSISTITO	ETA_ACCESSO	C05_ACCESSO	C05B_PATOLOGIA_TRIAGE	C06_DIAGNOSI_PRINCIPALE	C07_ESITO_DIMISSIONE	C100_EVENTO	C101_TIMESTAMP1	C103_ATTRIBUTO
0	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	ACCESSO	01/01/2017 00.17.15	AUTONOMO
1	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	DIMISSIONE	01/01/2017 01.27.39	AZZURRO
2	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	PRESTAZIONIPS	01/01/2017 01.26.38	1047, VISITA DI PS
3	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	TRIAGE	01/01/2017 00.17.51	AZZURRO
4	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	USCITA	01/01/2017 01.28.51	NaN
5	09060203-PS-2017000001	3011.0	49.0	01/01/2017	C8-MANIFESTAZIONI CUTANEE	V679-VISITA DI CONTROLLO	DIMISSIONE A DOMICILIO	VISITA	01/01/2017 01.26.38	AZZURRO
6	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	ACCESSO	01/01/2017 00.40.10	AUTONOMO
7	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	DIMISSIONE	01/01/2017 04.15.13	VERDE
8	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	LABORATORIO FINE	01/01/2017 02.38.34	ESAMI-L0000012481
9	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	LABORATORIO FINE	01/01/2017 03.31.12	ESAMI-L0000012510
10	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	LABORATORIO INIZIO	01/01/2017 01.50.56	ESAMI-L0000012481
11	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	LABORATORIO INIZIO	01/01/2017 02.42.18	ESAMI-L0000012510
12	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	PRESTAZIONIPS	01/01/2017 00.59.18	1047, VISITA DI PS
13	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	PRESTAZIONIPS	01/01/2017 01.50.56	7073, PRELIEVO VENOSO
14	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	PRESTAZIONIPS	01/01/2017 02.42.18	7073, PRELIEVO VENOSO
15	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	TRIAGE	01/01/2017 00.40.21	VERDE
16	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	USCITA	01/01/2017 04.17.17	NaN
17	09060203-PS-2017000002	3012.0	82.0	01/01/2017	CR-SINTOMI O DISTURBI UROLOGICI	78820-RITENZIONE URINARIA	DIMISSIONE A DOMICILIO	VISITA	01/01/2017 00.59.18	VERDE

Figure 2.1: Table representation of an event log.

2.2 Event log

Process mining is based on event data that may come from a variety of resources: database systems, ERP systems, etc. The collected process data are then often structured in a table format, as the example in Figure 2.1 shows. Each table's row represents an event related to a specific process activity. Events are grouped into cases (i.e., single process instances), they are sorted in ascending timestamp, and the obtained sequence of activities performed in each case is referred to as *trace*. The event log is organized in different columns: one column lists the unique identifiers for each process instance, another column refers to the activities performed in each event, and one is dedicated to the recording of the event timestamps. These represent the essential information to model the process.

Additional information can be stored in the event log: data related to the resources that perform the activity, the cost of performing such task, and attributes. For example, in Figure 2.1 there are columns referring to information not strictly necessary for describing the process, but they enrich the event log with details that can be exploited to investigate additional perspectives: resources can be studied to inspect the organizational perspective, timestamps to investigate bottlenecks, event attributes to research activities details. Depending on the process mining technique used and the question at hand, part of the information contained in the event log is selected.

Another structure used to represent the event log is the XES (eXtensible Event Stream) format [5]: the standard format supported by the majority of process mining tools. A XES document is an XML-based document that contains one log consisting of any number of traces. Each trace describes a sequential list of events corresponding to a particular case. The log, its traces, and its events may have any number of attributes [6]. Figure 2.2 shows an extract of the XES representation of the previous event log. It can be seen its correspondence with the Figure 2.1: each row of the table, representing an event, is translated into a tag <*event*>, whereas each column name here is represented as a different field to which is linked the corresponding event value.

The quality of the process mining result heavily depends on the input [4]. For this reason, it is important to carefully extract the information to build an event log. Information systems usually collect the data in several tables

```xml
<?xml version="1.0" encoding="utf-8" ?>
<log xes.version="1849-2016" xes.features="nested-attributes" xmlns="http://www.xes-standard.org/">
    <trace>
        <string key="concept:name" value="0" />
        <event>
            <string key="case" value="09060203-PS-2017000001" />
            <string key="C02_IDUNIVOCOASSISTITO" value="3011.0" />
            <string key="ETA_ACCESSO" value="49.0" />
            <string key="C05_ACCESSO" value="01/01/2017" />
            <string key="C05B_PATOLOGIA_TRIAGE" value="C8-MANIFESTAZIONI CUTANEE" />
            <string key="C06_DIAGNOSI_PRINCIPALE" value="V679-VISITA DI CONTROLLO" />
            <string key="C07_ESITO_DIMISSIONE" value="DIMISSIONE A DOMICILIO" />
            <string key="C100_EVENTO" value="ACCESSO" />
            <date key="C100_TIMESTAMP1" value="2017-01-01T00:17:15+00:00" />
            <string key="C103_ATTRIBUTO" value="AUTONOMO" />
        </event>
        <event>
            <string key="case" value="09060203-PS-2017000001" />
            <string key="C02_IDUNIVOCOASSISTITO" value="3011.0" />
            <string key="ETA_ACCESSO" value="49.0" />
            <string key="C05_ACCESSO" value="01/01/2017" />
            <string key="C05B_PATOLOGIA_TRIAGE" value="C8-MANIFESTAZIONI CUTANEE" />
            <string key="C06_DIAGNOSI_PRINCIPALE" value="V679-VISITA DI CONTROLLO" />
            <string key="C07_ESITO_DIMISSIONE" value="DIMISSIONE A DOMICILIO" />
            <string key="C100_EVENTO" value="DIMISSIONE" />
            <date key="C100_TIMESTAMP1" value="2017-01-01T01:27:39+00:00" />
            <string key="C103_ATTRIBUTO" value="AZZURRO" />
        </event>
```

Figure 2.2: Extract of XES document.

that are linked to each other. According to the analysis we want to carry out, the best strategy is to pose the right research questions in advance and then collect from these tables the needed data. Unfortunately, data often shows a lack of accuracy or even missing information (such as resources involved in an activity or the initial timestamp of the activities) that does not allow performing detailed studies, sometimes even leading to not valuable results.

2.3 Process model

Process modeling represents one of the essential procedures in process mining studies. Defining a model helps to better understand how a process is run, which are the activities performed, and gives an intuitive view of the sequence flow. There are essentially two methods to define a process model: the first one is to directly mine it from the event log: there exist several process discovery algorithms (e.g., α-algorithm) that automatically convert the data into a workflow-net. The other approach consists in drawing a model by hand; in this case, it is necessary to retrieve information about the process in order to define the model structure. To represent a process model, either drawing or mining it, several notations are available, e.g. BPMN (Business Process Modelling Notation), Petri net.

- **Petri net**: it is the oldest and best-investigated process modeling language allowing for the modeling of

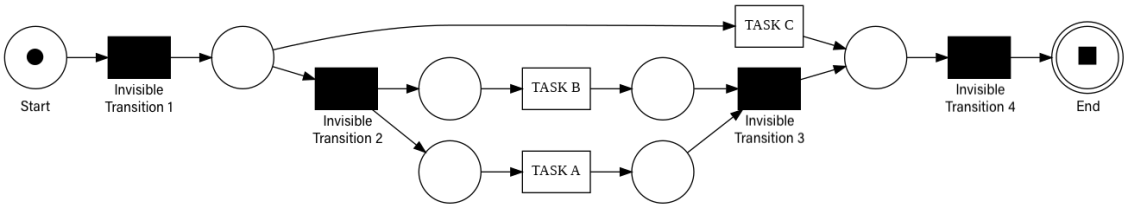


Figure 2.3: Petri net example.

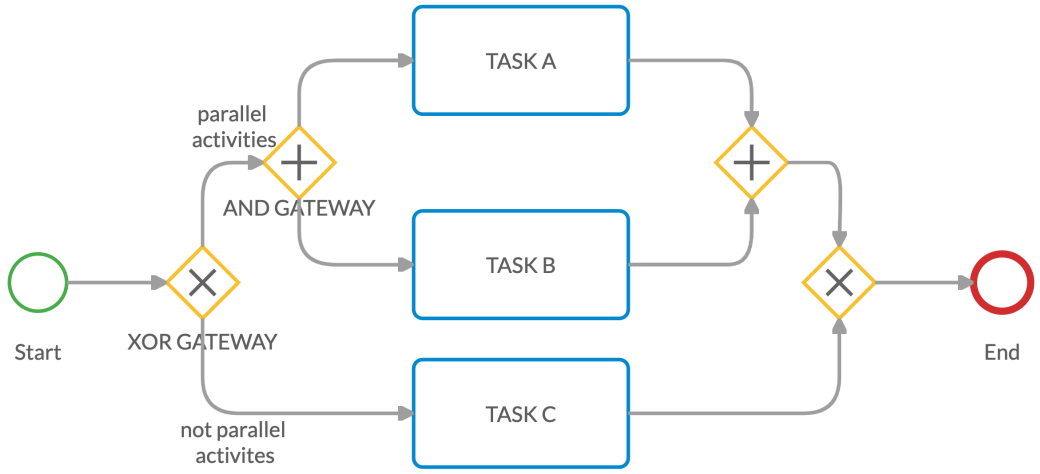


Figure 2.4: BPMN example.

concurrency. This model is a bipartite graph consisting of places and transitions [6], depicted as circles and rectangles, respectively. A place can contain multiple numbers of tokens, which are figured as black circles and represent cases. Each Petri net is characterized by one starting place and one ending place. The token begins its process in the starting place that is immediately followed by a transition and flow through the network governed by the firing rule: when all the places in input to a transition contain at least one token, the transition is enabled and it fires consuming one token from all its input places and creating tokens in the output ones. Figure 2.3 shows an example of this notation: in the initial marking, i.e., the initial token distribution over places of the Petri net, there is just one token in the Start place. This token enables the first transition, which is an invisible transition as it does not represent a real activity of the process. This transition fires and creates a new token in its output place. Here the token can flow through transition *TASK C* or through *Invisible Transition 2*. If *TASK C* is enabled, it is created a token in its output place. If, on the other hand, the token undertakes the path below, the invisible transition fires and creates two tokens, one for each output place. Here, each token enables the parallel transitions *TASK B* and *TASK A*. Afterward, the produced tokens in output enable the *Invisible Transition 3*, which creates a token in its output place. Eventually, *Invisible Transition 4* fires and creates a final token in the End place, which determines the end of the token process.

- **BPMN**: Business Process Model and Notation is a standard notation for business process modeling developed by the Object Management Group (OMG) with the primary goal to provide a notation that is readily understandable by all business users [7]. Events (things that happen instantaneously) are repre-

sented by circles and are comparable to places in Petri net, whereas activities, similarly to transitions in Petri net, by rectangles. With over 100 symbols, BPMN is a fairly complex language [8] that allows representing parallelisms, loops, and series of activities exploiting gateways (e.g. XOR, AND, OR). In this language, events cannot have multiple input or output arcs. The start events have a unique outgoing arc, similarly to end events which have a unique incoming arc. On the other hand, intermediate events are characterized by one incoming and one outgoing arc. The parallelism between activities is modeled through AND-gateways, whereas the XOR-gateways depicts the exclusivity between activities based on a condition. Figure 2.4 shows an example of this notation and replicates the same process model translated from the Petri net one.

In this project, we take advantage of the BPMN notation to draw the process model. Nonetheless, we will show that in order to exploit some process mining tools it is necessary to translate it into the Petri Net language. Several tools allow one to draw BPMN models, some examples are Camunda* and Cardanit†. Once configured the model, it is possible to download the file in XML format which represents the model in the standard language accepted by the majority of process mining tools. On the other hand, the translation into a Petri net model can be obtained by leveraging a specific function of PM4Py‡, a Python library for process mining.

2.4 Process simulation

Business process simulation refers to techniques for the simulation of business process behavior on the basis of a simulation model consisting of a business process model extended with additional information for the definition of the different run-time simulation aspects: case arrival rate, task, durations, routing probabilities, resource allocations and utilizations, etc. [9]. A simulation model allows the execution of a set of process traces that imitate the behavior of a process in a virtual setting. The main advantage of this approach is the possibility to run the process in a safe isolated environment before they are deployed [10]. Simulating a process to study its performance without effectively implementing it is extremely valuable to companies which can save costs and run the current operations without deviations. With process simulation, one could explore what-if scenarios with which it is possible to understand how the process reacts to different settings, examine the new solutions' impact on the behavior of the process, and take decisions on the base of it.

In order to develop a simulation model, insights into the process behavior should be gathered. This relates, amongst others, to the order of activities, their duration, and the availability of resources [11]. This information can be inferred from the collected data ([12], [13], [14]). Unfortunately, frequently the information systems do not store all the details needed to build the entire simulation model and the event logs alone suffer from data quality issues. Traditionally, the event logs are thus enriched with domain experts' interviews. The necessity to rely on human sources may lead to information that does not fully reflect reality since it can be based on subjective opinions, and this has to be taken into account when developing the simulation model.

Once gathered the needed information, the simulation model, such as a Petri net or a BPMN, that shows the control-flow perspective, can be extracted. This model can then be enhanced with additional details about resources, timestamps, costs, attributes, etc. The final step consists in defining the simulation parameters that

*https://camunda.com/

†https://www.cardanit.com/

‡https://pm4py.fit.fraunhofer.de/

characterize the process such as calendars, inter-trigger-timers, processing times, waiting times, etc. All this information is then combined to generate a XML file: this document has to be compliant with the Business Process Simulation specification (BPSim), a standardized format by the Workflow Management Coalition (WfMC) [10]. BPSim format is readable by several software simulation tools which, taking the document in input, are able to generate the simulated process.

In this work, in order to create a simulated process, we leveraged the LANNER L-Sim simulator[§] which supports BPMN 2.0 Interchange format and the WfMC BPSim standard to enable simulation of BPMN based models and diagrams and a structured return of statistical results. L-Sim offers an extensive API for rapid integration, simple simulation control, visualization options and is able to operate entirely through the exchange of serialized XML data. This tool takes in input the BPSim file and gives as output the simulated process.

2.5 BPSim and BPSimpy

BPSim framework is a specification defining the standardized rules to augment business process models captured in BPMN with the information needed to run simulations. This standard format allows for the creation of an XML file that guarantees interpretability by several simulation software. It defines a specification for the parametrization and interchange of process analysis data allowing structural and capacity analysis of a process model providing for pre-execution and post-execution optimization [10].

This framework provides for the definition of scenarios that are used to capture the parameters needed to define the simulation process. Each scenario is composed of a collection of element parameters. Typically, there are defined the scenario's starting time, the duration, the number of replications of that scenario, etc. There are then declared additional details for each BPMN element. Specifically, each process element is enriched with information about the time, resource, and control perspective. Time perspective typical parameters are the duration of the activity, processing time, setup time, etc. When defining the resources, firstly there are declared the roles of each resource, then the parameters which determine the resource quantity and availability. Concerning the control perspective, BPSim standard allows for the definition of the inter-trigger timer, which specifies the time interval between two start events, the trigger count, that gives the maximum number of times to trigger an event, and the outgoing probabilities of exclusive gateways. In addition, the parameters may be attached to specific calendars, which set the time range in which they are enabled.

The XML file generation can be automated by leveraging the Python BPSimpy library [9]. When this library is called, it is run a Python constructor giving in input a BPMN model and the simulation parameters, and it is generated the BPSim simulation document. In our studies, we exploited this Python package to compile the input simulation files, which contain control parameters such as the starting time and the inter-trigger timer, time parameters regarding the distribution characterizing the activity durations, and specifications about the resource perspective.

Figure 2.5 shows an example of the Python code needed to implement a BPSim file. The BPMN model taken as input is displayed in Figure 2.4. Firstly, it is imported the BPSimpy library and a BPSim object is created by taking in input the BPMN. In this example, it is defined a single scenario, which is identified by a specific id and name.

[§] `https://www.lanner.com/en-us/technology/l-sim-bpmn-simulation-engine.html`

```python
import BPSimpy

example = BPSimpy.BPSim('Example.xml', verbosity = None)
# Setup Scenario
scenario = example.addScenario(id = 'S', name = 'Scenario', description = 'Example', author = 'M')

# Add scenario parameters
from datetime import datetime, timedelta
scenario.addScenarioParameters(replication = 1, baseTimeUnit = 's')
scenario.addStart(value = datetime(2022,1,1))
scenario.addDuration(value = timedelta(days = 365))

# Add control parameters
# Start
start = scenario.getElementParameters(example.getNameById('Start'))
start.addInterTriggerTimer(nameDistribution = 'NegativeExponentialDistribution', mean = 1,
                           validFor = 'Case Arrival Calendar')
start.addTriggerCount(value = 1000)

# Add probability to XOR-Gateways
xor_gateway_ab = scenario.getElementParameters(example.getNameById('parallel activities'))
xor_gateway_ab.addProbability(value = 0.9)
xor_gateway_c = scenario.getElementParameters(example.getNameById('not parallel activities'))
xor_gateway_c.addProbability(value = 0.1)

# Add time parameters
task_A = scenario.getElementParameters(example.getIdByName('TASK A'))
task_A.addProcessingTime(nameDistribution = 'TruncatedNormalDistribution', mean = 0,
                         standardDeviation = 3, min = 0, max = 10)
task_B = scenario.getElementParameters(example.getIdByName('TASK B'))
task_B.addProcessingTime(value = 5)
task_C = scenario.getElementParameters(example.getIdByName('TASK C'))
task_C.addProcessingTime(value = 3)

# Add Calendar
from icalendar import Calendar
cal_start = Calendar()
cal_start['begin'] = 'VEVENT'
cal_start['dtstart'] = '20220101T080000'
cal_start['end'] = '20221231T240000'
cal_start.add('rrule', {'freq': 'daily', 'byday': ('MO', 'TU', 'WE', 'TH', 'FR', 'SA',' SU')})
cal_start['end'] = 'VEVENT'
cal_start['version'] = '2.0'
scenario.addCalendar(name = 'Case Arrival Calendar', id = 'Case Arrival Calendar', calendar = cal_start)
```

Figure 2.5: Python code to implement the simulation for the BPMN model in Figure 2.4 with BPSimpy.

The following lines of code define: firstly, the inter-trigger timer distribution, which tells the tokens' interval arrival time, and the number of tokens to be processed (method addInterTriggerCount); secondly, the branch probabilities for each XOR-gateway; from line 27 to 33, the time distribution of the tasks; finally, the calendar, which regularizes the time interval in which tokens can arrive. This example shows just some of the simulation parameters made available by the BPSim library, which can can be downloaded from a Github repository[¶].

[¶] https://github.com/claudiafracca/BPSimpyLibrary

2.6 Estimation of start timestamps of activities

Several process mining techniques, including to build process simulation models, rely on the presence of both the starting and ending timestamps of the activities. Unfortunately, as already mentioned in section 2.4, the majority of information systems only provide the completion of the process activities. One way to face this problem is assuming that an event starts as soon as the previous completed and a suitable resource is available. Another possibility is to approximate the starting timestamp of an activity as the average between its end and the end of the previous activity. Nonetheless, these approaches may result inaccurate and unrealistic. The first approach does not account for waiting times and it assumes that a resource is available as soon as it completes the previous task, without considering possible breaks during the working days, multi-activities carried out by the same resource in other processes, and the presence of additional tasks not recorded in the event log. The second method suffers from precision in the start timestamp estimation: it relies on the strong assumption that an activity lasts as much as the waiting time before starting it, which in general is not true.

Recently it has been introduced a new technique [12] to estimate the timestamps of the start events. The starting point is an event log of a given process and a simulation model of this process. The idea consists in enriching the event log with the missing start timestamps using n different activity duration profiles, thus obtaining a set of n event logs. Every event log is then exploited to compute the activity duration probability to include in the simulation model, leading to the construction of n different simulation models. Each simulation model is used to generate a simulated event log and these are then compared with the real event log. An error measure is computed by comparing the trace durations and the waiting times between each simulated event log and the original one augmented with the starting timestamps. Finally, it is returned the simulated log characterized by the smallest error, which represents the most realistic event log, augmented with the starting events' timestamps.

In particular, given the original event log $\mathcal{L}$, the technique aims to estimate the timestamp for each start event, based on the assumption that waiting times for different instances of the same activity are similar, i.e., these instances are executed by the same type of resources, which exhibit similar behavior. Given an activity a and the corresponding event e, the problem is formulated as finding a value for a parameter $\alpha(e) \in [0,1]$ related to the event e such that:

$$time(e') = \alpha(a) * mintime(e) + (1 - \alpha(a)) * time(e) \tag{2.1}$$

where *e'* corresponds with the matching start event. The *mintime(e)* depicts the minimum timestamp when the activity could have started, considering the different process constraints, e.g. on resource and control flow. As shown in Equation 2.2, it is found as the maximum between the ending timestamp of the previous (non parallel) activity, $time(prev\_t_{\mathcal{L}}(e))$, and the timestamp in which the needed resource is available $time(prev\_r_{\mathcal{L}}(e))$:

$$mintime(e) = max(time(prev\_t_{\mathcal{L}}(e), time(prev\_r_{\mathcal{L}}(e)) \tag{2.2}$$

Notice that if the event log has no resource information, *mintime(e)* only depends on the previous activity completion timestamp.

From Equation 2.1 follows that if $\alpha(e) = 0$, the activity is performed instantaneously, without any waiting

time. Conversely, if $\alpha(e) = 1$, the activity begins as soon as the previous one is concluded.

The technique is developed as follows: given as input a process model (e.g., BPMN), the event log $\mathcal{L}$ and a set $\alpha_1, \alpha_2, ..., \alpha_i$, where i is the number of activities, firstly $\mathcal{L}$ is enriched with the estimated starting timestamps e', computed using Equation 2.1, creating the new event log $\mathcal{L}^{\alpha}$. From the latter log, it is possible to compute the probability distribution $\mathcal{D}^{\alpha}$ of trace duration as well as the waiting time probability distribution d_i^{α} for each activity. Furthermore, once defined the necessary simulation parameters (e.g., inter-trigger-timer, trigger-counter, resources, etc.), it is created the corresponding simulation model. The processing times are estimated from the probability distribution of activities' durations, $durat_i^{\alpha}$, computed on $\mathcal{L}^{\alpha}$. The obtained simulation model can be run so to obtain an event log $\mathcal{L}^{sim}$. Log $\mathcal{L}^{sim}$ is then compared with $\mathcal{L}^{\alpha}$ in order to find the error $\Delta(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})$, computed as the sum of the distance of the trace duration distribution and the sum of the distance of the waiting times distributions of each activity. The final error function is shown in Equation 2.3,

$$\Delta(\mathcal{L}^{sim}, \mathcal{L}^{\alpha}) = \varepsilon_{(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} + \sum_i \varphi_{i(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} \tag{2.3}$$

where $\varepsilon_{(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} = \int_0^{+\infty} |\mathcal{D}^{sim}(x) - \mathcal{D}^{\alpha}(x)| dx$ represents the distance between the trace duration distributions of the two event logs, whereas $\varphi_{i(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} = \int_0^{+\infty} |d_i^{sim}(x) - d_i^{\alpha}(x)| dx$ depicts the distance between the waiting distributions of activity i.

The integral absolute difference between the distributions is computed via the Riemann integral technique and calculates the distance between the two areas under the curve. Given that the two curves represent density functions, each area sum to 1. Hence, the result of the integral is a number $\in [0, 2]$: 0 if the two distributions are exactly the same, 2 if the two distributions are not overlapping.

This process is run for j different configuration of $\alpha_1, \alpha_2, ..., \alpha_i$ with the objective to find a (sub)optimal minimum of the error function. Finally, it is accepted the lowest $\Delta(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})$, yielded by the best set A^j=$\alpha_1^j, \alpha_2^j, ..., \alpha_i^j$. The generated $\mathcal{L}^{\alpha^j}$ represents the event log with the best-estimated start timestamps.

The algorithm exploited to explore the $\mathcal{A}$ space (space spanned by all the $\alpha_1, \alpha_2, ..., \alpha_i$) and figure out the best solution, is a local search based algorithm. This method takes as input an event log $\mathcal{L}$, an initial simulation model, a $mintime_{\mathcal{L}}$ function, and a parameter $\delta \in (0, 1)$. The δ parameter defines the succession $alpha\_succ(\delta)$ = $\{x_t | x_t = x_{t-1} + \delta, x_0 = 0, x_t \leq 1\}$, in such way it is possible to obtain a different configuration obtained via funcion $\alpha(a) = x_t$ for each activity.

The algorithm start initializing the set $\alpha_1, \alpha_2, ..., \alpha_i$ with random values. Then it is selected an activity i of the log and try to optimize α_i using local search. In particular, for each activity and the corresponding value $\alpha_i = x_t \in alpha\_succ(\delta)$, for the next values of α_i in $\mathcal{Q}_{next}$, it is added the previous value $x_{t-1} \in alpha\_succ(\delta)$ and the consecutive value $x_{t+1} \in alpha\_succ(\delta)$. Once found the error, if it is smaller then the previous one, it is stored, and the algorithm keeps going in the next updates in the direction with decreasing logs distance until no improvements are permitted.

In this work we aim to improve this technique in the accuracy of timestamp estimation, presenting a modified version of the error and two different optimization algorithms. The next two paragraphs introduce MOGAII and NSGAII, two genetic approaches that we will exploit to find the error (sub)optimal.

2.7 Genetic optimization algorithms

An optimization algorithm is a procedure aiming to discover the best values satisfying a given problem. It usually consists of a series of steps iteratively repeated in order to minimize/maximize the function at hand. Research in this field is constantly evolving and several techniques have been proposed.

Optimization problems can be solved through the use of genetic algorithms, i.e., computational models inspired by biological evolution. The main attractivenesses of genetic algorithms are that they are usually robust algorithms and can tolerate even approximate or noisy objective evaluation, they can be parallelized and can therefore take full advantage of the massively parallel computer architecture, and they can directly approach a multi-objective optimization problem.

The basic requirements of these algorithms are a genetic representation of the solutions and a fitness function. Originally, genetic algorithms worked with the binary coding of the solutions, inspired by DNA sequences. Subsequently, new structure representations have been introduced and nowadays it is possible to deal with continuous variables as well.

The main idea of these algorithms is to start from a random population of possible solutions and exploit genetic inspired techniques in order to find the best results, either local or global optimums. This approach can be leveraged both in single and multi-objective optimizations.

In general, given a population of solutions, at each evolutionary step of the algorithm, the fitness (i.e., the value of the objective functions) of every element is computed. Then, designs are selected with probability proportional to their fitness from the population and transformed by genetic operators to form a new population, namely the offspring. Examples of operators are:

- **Selection**: a design is copied to the next generation without any modifications;
- **Classical crossover**: two designs of the parent population exchange their genetic material to form a new design in the next generation. If the solutions are represented in binary structures, this corresponds to cutting two sequences and exchanging their tails.
- **Directional crossover**: it assumes that the direction of improvement can be detected in the fitness values of individuals in the same generation. The direction of improvement is evaluated by comparing the fitness of three stochastically selected individuals of the same generation. The new design in the next generation is created by moving in a randomly weighted direction computed based on the direction vector of the three parent individuals;
- **Mutation**: the genetic material of a design is randomly modified to create a new design in the next generation.

The offspring produced will then be evaluated and if its fitness is higher than the parent population will replace the less fitted values coming from it.

Besides these genetic modifications of the parent population to create new designs, modern multi-objective algorithms have introduced other techniques to improve convergence and become more robust. An example is *elitism*, an evolutionary algorithm in which the best solutions, called 'elites', are directly inserted into the new generation without being modified. This operation ensures that the best solutions are preserved during evolution.

The algorithm terminates when one stopping condition is satisfied. Three possible mechanisms are:

- The algorithm has reached the pre-defined number of iterations;

- No improvements have been generated after x iterations;
- The fitness has reached the desired value.

The set of best solutions forms the Pareto frontier, i.e., the set of non-dominated solutions. A solution is said to be 'non-dominated' if it is not possible to improve the value of a target function without reducing the quality of the other objective functions. The objective of a genetic algorithm is to find solutions as near as possible to the Pareto front.

When dealing with single-objective optimization, the Pareto optimal solution is unique, whereas in multi-objective optimization the Pareto front may contain several (possibly infinite) solutions. According to the type of problem one is dealing with, the solution to be picked from the set of possible ones represents a trade-off between all the objective functions.

In this work two evolutionary algorithms, MOGAII and NSGAII, are used as alternative to the local-based search approach to improve the estimation of the start timestamps. The following subsections describe how they work.

2.7.1 MOGAII

The Multi-Objective Genetic Algorithm (MOGAII) [15] starts from an initial generation of N random solutions encoded by binary strings. Using the genetic operators already mentioned, it is generated the first offspring. In order to create a new design, an operator is chosen with the following probabilities: 0.5 for the directional crossover, 0.1 for mutation, and 0.05 for selection. The probability of the classical crossover is automatically set to one minus all the other probabilities.

Then, a first evaluation of the individuals' fitness is carried out. All non-dominated designs are copied and stored in the elite set. Each time this step is repeated replacing those individuals that have lower fitness. The collection of elite solutions is limited to be equal to the initial size N of the population. Thus, if the non-dominated solutions exceed the boundary, the elite set is reduced to N solutions by randomly removing designs in excess. On the contrary, if it does not reach the necessary numerosity, the gap is filled with the highest fitness old solutions.

Once the population is evaluated, the MOGAII operators generate the new offspring by considering the parent population and the elite set and applying the genetic operators. In particular, the highest the fitness of an individual, the higher the probability to be involved in a genetic operation and to be part of the new generation. These steps are iteratively repeated until a stopping criterion is met.

The environment in which the algorithm is run allows to choose between *autonomous*, *self-initializing* and *manual configuration* modes.

In this work, the majority of the optimizations have been run with the autonomous mode, because it does not require parameters, and it uses the information gathered from the problem analysis to drive the optimization in the right direction. In this case, the algorithm terminates when it is not able to find enough dominating designs or the designs that it finds are neither dominating nor dominated. In the remaining cases, the only parameter manually set has been the number of iterations, limiting the optimization to terminate when reached the considered threshold.

2.7.2 NSGAII

The Non-dominated Sorting Genetic Algorithm [16], as the name suggests, is a genetic algorithm that introduces a smart non-dominated sorting approach. In general, the computations to find if a solution dominates the others are very demanding since they require comparing the considered design with all the remaining population. NSGAII exploits an original technique in which it is calculated firstly the domination count n_p, i.e., the number of solutions which dominate solution p, and then the set, S_p, of solutions that the solution p dominates. All the designs with an empty set S are those non-dominated, i.e., the best available solutions. For each individual p belonging to this set is then explored its members of set S_p, i.e., those designs dominated by the considered solutions p. The domination count of these solutions is reduced by one and those in which become zero are marked as the second non-dominated front. This procedure continues until all fronts are identified. This mechanism speeds up the sorting procedure from $O(MN^3)$ (where M is the number of objective functions and N the population size) to $O(MN^2)$.

Besides this, one of the requirements is that the algorithm is able to maintain a good spread of solutions in the Pareto front. To this aim, NSGAII introduces the technique called *crowding distance*. This mechanism relies on computing the average distance of two points on either side of this point along each of the objectives. The computed quantity serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices. In order to find this quantity, it is necessary to sort the population according to each objective function value in ascending order of magnitude. To those solutions having the smallest and largest magnitude is assigned an infinite distance value, whereas to the intermediate solutions it is assigned a distance equal to the absolute normalized difference in the function values of two adjacent solutions. Finally, the overall crowding-distance value is computed as the sum of individual distance values corresponding to each objective.

This technique allows discovering which are regions less populated of the front, which will be characterized by a high value of crowding-distance.

Thus, when during the optimization steps it has to be selected a solution, two information are considered: firstly, its non-domination rank, and secondly, its crowding-distance. Between two solutions belonging to the same front (i.e., with the same non-domination rank), it is preferred the design that is located in a less crowded region.

NSGAII algorithm starts with a random population of N solutions. Differently from MOGAII, solutions are sorted and marked considering their non-domination. The first ranked designs are selected and the offspring of size N is generated through modifications of the solutions given by the genetic operators.

Once the first iteration concludes, the following ones introduce also the elitism mechanism. In each consecutive step, the parent population and the generated offspring are combined, generating a population of size $2N$ and ranked according to their non-domination. Since all the previous and current population members are considered, elitism is ensured. Now, exploiting the non-domination rank and the crowding-distance operator, a new population of size N is created. This procedure continues until one of the stopping criteria is met.

As for the case of MOGAII, the environment in which the algorithm is launched allows choosing between three different modalities: *autonomous*, *self-iniziatilizing*, and *manual configuration*.

In this work, it has been exploited the autonomous mode, which does not require any parameter setting and stops when the Pareto frontier does not improve any further.

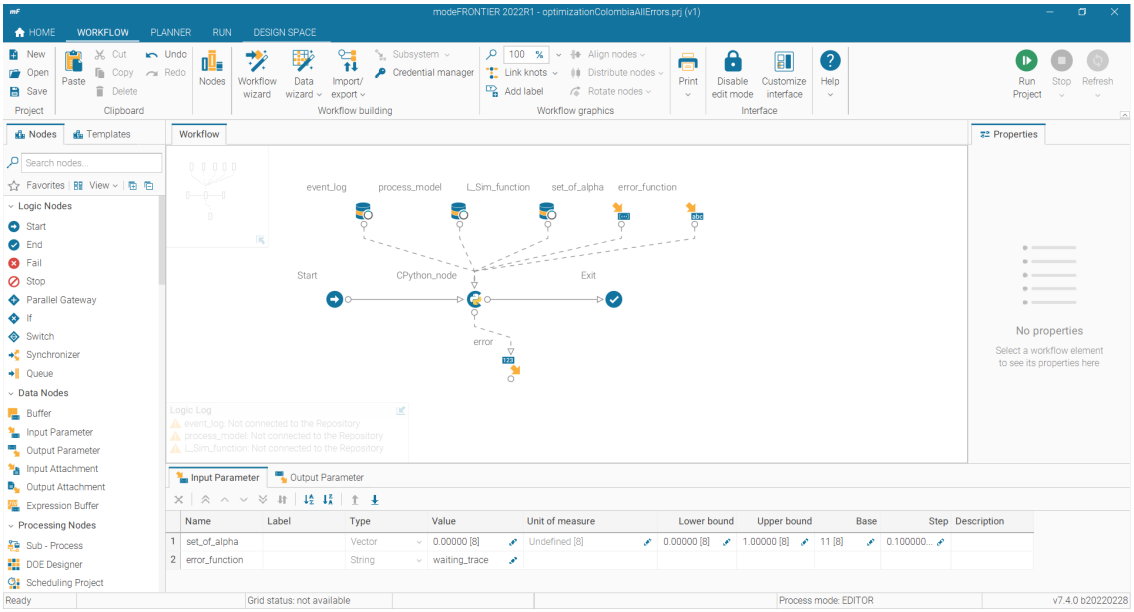


Figure 2.6: modeFRONTIER environment.

2.8 modeFRONTIER and VOLTA: optimization setup

modeFRONTIER [17] is the software environment (Figure 2.6) in which the optimization processes have been carried out in this work. The software requires defining a workflow for each optimization project. The workflow describes both the sequence of internal and external computations that transform input into output and the mathematical definitions of variables, constraints, and objectives. The basic building blocks of a workflow are links and nodes. Nodes represent the interface that integrates third-party software with modeFRONTIER (e.g. CPython node in which there are uploaded the Python classes) and describes data (variables, parameters and the objectives of the problem (i.e., the error functions)). Two basic nodes, present in every workflow, are the Start and End nodes. Links are lines connecting the nodes and indicating the direction in which information is exchanged between nodes. Once created the workflow, the project is saved and can be run locally or on a dedicated machine.

In this work, modeFRONTIER environment has been exploited to perform optimization of the objective error functions at hand, leveraging two genetic optimization algorithms: MOGAII and NSGAII.

On the other hand, VOLTA is a web platform for multidisciplinary business process optimization and the management of enterprise simulation data [18] developed by ESTECO S.p.A.[‖]. Its main objective is to offer a collaborative environment installed in companies' servers where to store simulation data in session tables. The peculiarity of this platform is that, once uploaded a modeFRONTIER workflow, it offers the possibility of sharing with external people the necessary tools to work with data optimization and visualization. In this work, it has been created a dedicated VOLTA environment thus to be able to exploit the genetic algorithm without activating a new modeFRONTIER license.

[‖] https://www.esteco.com/

3
Related Works

Since process mining is a novel branch of data science, research in this field is continuously evolving. One important area of research concerns process simulation, a technique leveraged to build processes that emulate the behavior of a real process in a virtual setting. This has the advantage of being able to test process modifications without having to implement these changes in practice [19]. This technique is applied in a variety of research fields, including healthcare.

Literature offers several examples in which process mining has been used to analyze and enhance hospital services. Duma and Aringhieri in [20] focus on discovering a model capable to properly replicate the possible patient paths and on predicting the next activities on the basis of their characteristics and their activities performed until that moment. Similarly to our research, they work on emergency department data, but they extensively rely on domain experts' information in pre-processing the data and in building the process model. Furthermore, they limit their studies to process discovery without developing simulation processes and plausible what-if scenarios. Van Hulzen et al. in [11] present a real-life case study at the radiology department of a Belgium hospital, aiming to analyze solutions for merging two of their radiology facilities in a new hospital site. Since the variety of sub-processes emerging in a radiology department is much lower than those in an emergency department, where only unplanned patients with very different symptomatologies are received, the development of a process model is simplified. Furthermore, as in [20], they largely interact with domain experts to face data quality issues in the event log.

These research examples differ from our work in that the latter does not rely on human opinions for the data pre-processing, but it tries to estimate the missing relevant data with an algorithmic approach thus not introducing a bias error given by subjective opinions.

Indeed, to develop a simulated process that faithfully reflects reality, an event log has to present essential information. Hence, since it often happens that real-world event logs contain noisy or corrupted data records [21], many research techniques focus on repairing the event logs from missing information.

One of the main information event logs frequently lacks is the start events of the activities [19]. A naïve ap-

proach [22] suggests facing this problem by estimating the start timestamp of activities by assuming that it begins as soon as the previous one concludes, but this is often unrealistic in real processes. Duma and Aringhieri, in their work [20], estimate the missing start timestamps by subtracting the average service time from the final timestamp of the activity or relying on time reported by the direction of the ED staff. Given the huge variety of hospital performance, the former approach may result imprecise, whereas the latter one may introduce a human opinion that does not fully reflect reality. Van Hulzen et al. in [11] deal with the missing timestamps problem by asking domain experts to identify the minimum, the maximum, and the most likely time required to complete a particular activity, exploiting that information as input for a triangular distribution. Again, this approximation may result inaccurate in describing a real-life scenario.

Denisov et al. [23] use linear programming over token trajectories to derive the timestamps of unobserved events, but they limit its applicability to acyclic processes without concurrency. Another technique [24] aims to restore missing events, but it assumes that the recorded events have no missing and correct timestamps, which does not always hold.

Camargo et al. in [25] proved that is possible to create a simulation model combining deep learning and data-driven simulation methods. The idea is to discover the process model and the branching probabilities with an automatic technique and to delegate the generation of activity start and end times to a deep learning model. Their proposal represents a valid alternative for the creation of a simulated process to the one used in this work, but it is limited in creating a simulated process without resource attributes. This turns out to be restrictive in developing 'what-if' scenarios where one of the research areas is the impact of the process modification on the resources' degree of utilization. Furthermore, although they demonstrated the goodness of the simulation model created, they also admitted that the accuracy of the deep learning model degrades in developing 'what-if' scenarios.

4

Estimation Of Start Timestamps: A Genetic Optimization Approach

In Section 2.6 we presented a technique that optimizes a parametric function to find out an estimate of the activities' start timestamps. Here we introduce an enhancement of this technique with the aim to improve the accuracy of the estimation compared with the approach by Fracca et Al. This chapter introduces the necessary steps to compute the best alpha parameters. Experiments were carried out using two genetic algorithms, MOGAII and NSGAII, on two different case studies. The comparison of the results with those obtained from the approach discussed in Section 2.6 shows that our techniques outperform the local-based search strategy developed by Fracca et Al.

4.1 Weighting of the error

The main problem emerging from the technique proposed by Fracca et Al. in computing the error function through Equation 2.3 is the lacking of weighting parameters that give different relevance to its terms. As mentioned in Section 2.6, the absolute integral difference of two density functions is a number that stands in the interval $[0, 2]$. If two distributions are similar, this number will tend to 0. On the other hand, when there is a remarkable distance between the two density functions, it will tend to 2. This leads to give similar importance to all the terms in the final error function, without taking into account the time scale of the waiting times/case durations.

In particular, if the waiting time of an activity lasts some seconds, an absolute integral difference resulting near to 2 leads to a relevant rise in the total error function, comparable to the contribution that would give the same error computed on an activity that has a waiting time of hours. Thus, when optimizing the function, the

algorithm will try to diminish the two errors giving them the same importance, and this will not allow finding an alpha value that actually relevantly reduces the error for the worst timestamped activity, in terms of absolute time.

This turns out to be detrimental especially in process simulation. When developing a simulation model, it is required to define the distribution of the activity durations. If, on the one hand, a high estimation error on the start timestamp of an activity that is characterized by a short waiting time does not heavily influence the total duration of the simulated process, on the other hand, the same estimation error for an activity with longer waiting time can end up to make the simulation last days, months, or even years more than it should do. Indeed, in the latter situation, an imprecise start timestamp estimation leads to defining an activity duration that is drastically different from the real one in terms of absolute time.

The solution to this problem lies in multiplying each term composing the final error by a scaling factor. In order to give more relevance to errors in activities with higher waiting times, we decided to weight the error terms by the absolute difference between the median waiting times of the original log augmented with the start timestamps and the simulated log. The same is applied to the term referred to the trace duration. Thus the reformulation of the error is:

$$\Delta(\mathcal{L}^{sim}, \mathcal{L}^{\alpha}) = \varepsilon_{(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} * \gamma + \sum_{i} \varphi_{i(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} * \theta_i \tag{4.1}$$

where:

- $\varepsilon_{(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} = \int_0^{+\infty} |\mathcal{D}^{sim}(x) - \mathcal{D}^{\alpha}(x)| dx$ is the distance between the trace duration distributions of the two event logs;
- $\gamma = |median(\mathcal{D}^{sim}(x)) - median(\mathcal{D}^{\alpha}(x))|$ is the absolute difference between the median trace duration of $\mathcal{D}^{sim}$ and $\mathcal{D}^{\alpha}$;
- $\varphi_{i(\mathcal{L}^{sim}, \mathcal{L}^{\alpha})} = \int_0^{+\infty} |d_i^{sim}(x) - d_i^{\alpha}(x)| dx$ is the distance between the waiting distributions of activity i;
- $\theta_i = |median(d_i^{sim}(x)) - median(d_i^{\alpha}(x))|$ is the absolute difference between the median waiting time of activity i in $\mathcal{D}^{sim}$ and $\mathcal{D}^{\alpha}$.

This new error formulation allows the optimization method to concentrate on minimizing those terms related to items with larger distances, in terms of absolute time, between the simulated and the real log enriched by the start timestamps.

4.2 Best alpha computation

To find the best alpha configuration, a workflow in modeFRONTIER environment has been developed (example in Figure 4.1), which allows the automatic launch of the optimization methods (MOGAII and NSGAII) on the objective function, which is Equation 4.1 in our settings. It receives in input the event log, the process model, the error function, a given set of alpha, and a function to run the LANNER L-Sim simulator. Besides this, the environment allows the declaration of the δ parameter, which defines the distance between consecutive α values. Furthermore, the workflow has a central CPython node in which there are imported the two Python classes needed to create the BPSim document and compute the error.

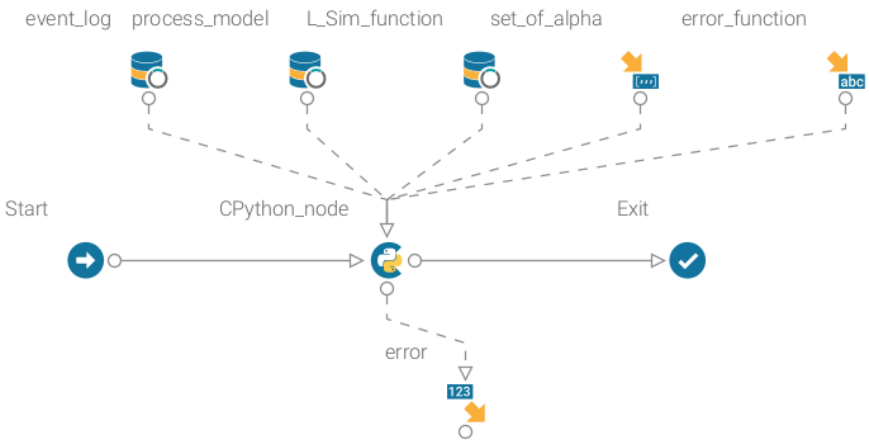


Figure 4.1: Example modeFRONTIER workflow.

When the workflow is enabled, the following pipeline is executed for all the alpha sets explored by the optimization algorithm:

1. Enrichment of the original log with the starting timestamps computed on the base of the set of alpha received in input as shown in Equation 2.1;
2. Definition of the simulation parameters:
    - Start datetime;
    - Process duration;
    - Trigger count;
    - Inter-trigger timer;
    - Resources roles;
    - Calendars;
    - Branch probabilities;
    - Activities' durations distributions computed on the original log augmented with the starting timestamp, obtained at pass 1.
3. Creation of the BPSim file exploiting the BPSimpy library;
4. Call of the LANNER L-Sim simulator giving in input the BPSim file to run the simulation model;
5. Transformation of the simulator output in XES format;
6. Computation of the error between the simulation model and the original log enriched with the starting timestamps as shown in Equation 4.1.

At each step, the optimization algorithm extracts new values of the alpha set. Once the single pipeline round returns the result, the error and the respective alpha are stored in a VOLTA session table. At the end, it is possible to inspect all the configurations found by the algorithm and choose the one leading to the lowest error.

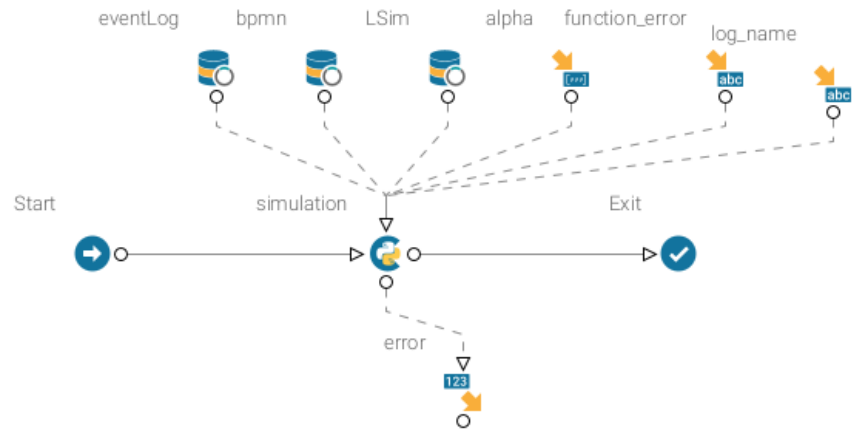


Figure 4.2: modeFRONTIER workflow for the two process case studies: process for student credential recognition, purchase process.

4.3 Optimization methods comparison

In this section it is assessed the improvement on the start timestamps estimation provided by optimizing the objective function with genetic algorithms. In particular, the aim is to determine if the optimization is able to converge to a (sub)optimal point leading to a smaller error with respect to the local-based strategy. The choice of exploiting genetic algorithms as optimization approach is due to their ability to explore a larger set of solutions space than that of the approach proposed by Fracca et al., and their higher capability to produce global optimums. The error function at hand is, indeed, a non-convex function which presents several local minimums and genetic algorithms should be able to overtake the optimization performance of the local-based search strategy used in [12].

Given the non-convexity of the error function, each genetic algorithm, accordingly to its parameter settings, can discover different solutions. Often, the final extracted point corresponds to a sub-optimum, which however represents a satisfactory solution to the problem.

In this section, there will be shown experiments conducted on the same event logs used in [12], comparing the results reached by the new techniques and the simpler local-based search one.

The event logs contain both the start and the end timestamps for each activity. In order to test the capacity of the technique, the start events have been removed. For both the case studies the same settings used in the article have been developed. The process models have been mined with the *Visual inductive miner* [26] Prom plug-in, which uses the inductive miner algorithm [27] to discover the Petri net models, which consecutively is translated into BPMN language. Then, the simulation parameters have been discovered exploiting both Prom packages (e.g. *Multi-perspective process explorer* [28]) and the Python library PM4Py. Finally, it has been complemented the BPMN model with this information exploiting the Python BPSimpy library. All the simulation models are generated using the LANNER L-Sim simulator.

In order to estimate the best alpha parameters, a modeFRONTIER workflow has been shaped (Figure 4.2) and the steps described in Section 4.2 followed. Besides the already cited input parameters, the workflow is character-

ized by another node (*log_name*) that allows specifying the name of the case study at hand so to easily launch it on both the examples presented in this chapter. The optimization step δ is set to 0.05, i.e. the same step considered in [12], in order to carry out fair performance comparisons. Then, for the case of MOGAII and NSGAII, it has been performed another optimization run, considering δ=0.001 and studying how a larger searching space impacts the final convergence results.

Since the purpose of this research is to improve the accuracy with which the starting timestamps of the activities are estimated, the main effort consists in analyzing if a genetic approach leads to a higher reduction of the objective function. However, it has to be noticed that genetic approaches have a stochastic behavior, hence, in order to state their absolute efficiency, it should be conducted a deeper analysis running several times the optimizations and carrying out statistics on the final results. Nonetheless, it will be shown that the results obtained are able to overcome the local-based search strategy ones.

Notice that, in order to reduce the deterministic settings given by the single simulation run, it should be defined at each optimization step a batch of simulations, in which each one is created with a different random seed, and an average of the outcomes should be computed. In this work, for the sake of time, we decided to perform just one simulation for each alpha configuration. Hence, the results obtained contain a bias term due to the absence of randomness in the development of the simulation.

All the optimizations performed in this section have been run on a dedicated machine with Common KVM processor 2.67 GHz and 8 GB of RAM. For both the case studies and both for MOGAII and NSGAII the entire optimization cycle lasted ca. 4 hours for each case study.

On the other hand, the optimization on the two case studies which exploits the local-based search strategy requires less computational power as it explores a smaller number of possible alpha configurations, thus it has been run locally on a machine equipped with Intel Core i7-6700HQ 2.6 GHz and 16GB of RAM, and lasted ca. 3 hours.

4.3.1 Case study: process for student credential recognition

The real-life event log* describes the students' credential recognition process of a Colombian University. It contains 954 traces, 6870 events, and 18 different activities. The log provides also 561 resources who take part in each activity. To reduce the number of available resources, they have been grouped into 10 respective roles.

The BPMN model describing the process is shown in Figure 4.3 and it contains just the most frequent activities in the log. The simulation parameters are reported in Table 4.1, Figure 4.3 (branch probabilities), and Table 4.2. Notice that the latter table represents just the resources involved in the activities considered in the process model. The duration of the simulation is set to a high number of days to ensure that all the tokens processes complete. Hence, the simulation life span is determined by only the number of tokens and their trace durations.

*The event log is available at `https://github.com/AutomatedProcessImprovement/`

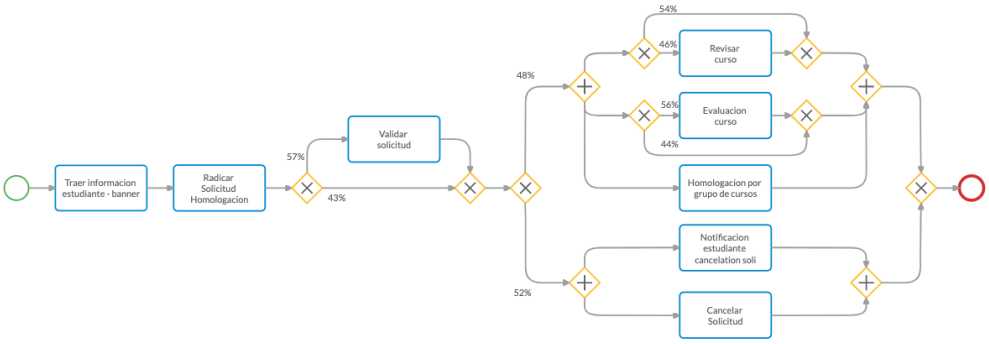


Figure 4.3: BPMN model and branch probabilities student credential recognition process case study.

Start Timestamp	Duration	Trigger Count	Inter-trigger Timer
1/2/2016	10000 days	954	NegativeExponentialDistribution mean 13408

Table 4.1: Simulation parameters student credential recognition process case study.

Role 1	Role 3	Role 4	Role 5	Role 6	Role 10
Cancelar solicitud / Notificacion estudiante cancelacio soli	Evaluacion curso	Homologacion por grupo de cursos	Traer information estudiante - banner / Radicar solicitud homologacion	Revisar curso	Validar solicitud
24h/7 8 resources	24h MO-TU-WE-TH-FR-SU 1 resource	24h MO-TU-WE-TH-FR-SA 5 resources	24h/7 1 resource	24h/7 5 resources	24/7 5 resources

Table 4.2: Number of resources per work shift and activities they are involved in student credential recognition process.

Given this information, the modeFRONTIER workflow has been launched twice: first exploiting MOGAII, then NSGAII, both in autonomous mode and with $\partial = 0.05$. The number of alpha parameters to be estimated corresponds to the number of activities involved in the process; thus, given that the process model consists of 8 activities, the same number of alpha parameters are introduced.

MOGAII explored 1001 different configurations of alpha values. Figure 4.4 shows the behavior of the optimization with a parallel coordinate graph; here, each line corresponds to a specific configuration of the alpha parameters and the color with which the line is painted illustrates the resulting final error: the darker the color is, the lower the error is (see also the legend on the right). In this specific case, 1001 different lines are drawn, one for each alpha configuration explored. To have a clearer view of the best alpha sets, we filtered the visualization considering only the configurations leading to the 20% best errors. To this aim, we firstly considered all the alpha sets explored, sorted them in ascending order according to their resulting error, and considered just the top 20%, which

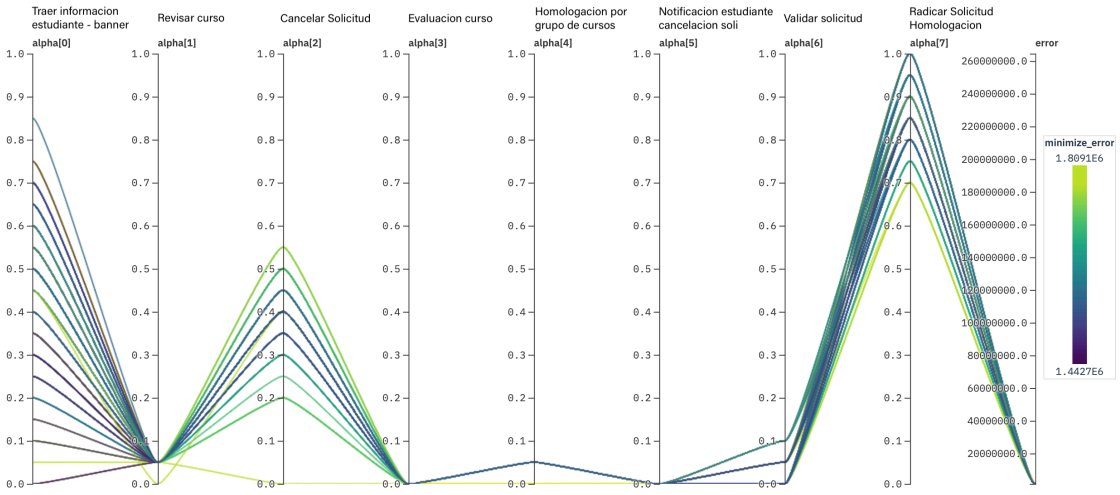


Figure 4.4: Alpha behavior filtered on the 20% best errors for MOGAII optimizer with $\partial = 0.05$ applied on the student credential recognition process.

in this specific case amount to ca. 200 alpha configurations. While it depicts a high concordance in choosing the values of $\alpha_3, \alpha_4, \alpha_5, \alpha_6$, it is more uncertain on α_0, α_2 and α_7. α_0 is the parameter related to the start activity and as such can assume any value without influencing the final error outcome. Indeed, for the first activity of each case, it is not possible to estimate the starting timestamp since there is no information about the time range in which the activity can start. Lacking any previous activity, the *mintime(e)* in Equation 2.2, necessary to the estimation of the start timestamp, can not be computed. Concerning the two other alpha parameters, i.e. α_2 and α_7, even if the filtered 20% best errors shown in the parallel graph include a higher range of their possible values, the color gradient shows how the algorithm reached convergence, determining specific values also for those parameters. An additional optimization has been run considering just those two alpha variables and keeping fixed all the others, setting them to their final best outcome. This second optimization aimed to explore if it is possible to furtherly tighten the admissible value range of α_2 and α_7 leading to a higher convergence and a smaller error.

Figure 4.5 shows how, with a second optimization, the algorithm is able to define with higher accuracy the best solutions for those alpha parameters. In fact, the indecision decreased and following the color gradient it is possible to spot a darker line that identifies a specific configuration that performs better than the others. The final alpha configuration identified by MOGAII optimization is reported in Table 4.3.

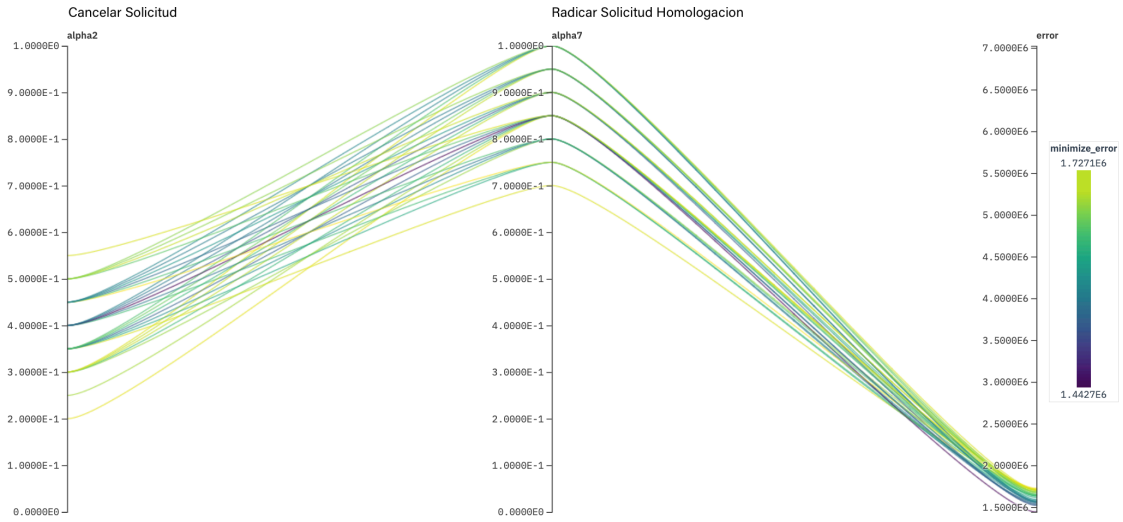


Figure 4.5: Second optimization of α_2 and α_7 with $\delta = 0.05$ keeping fixed all the other parameters to their best values.

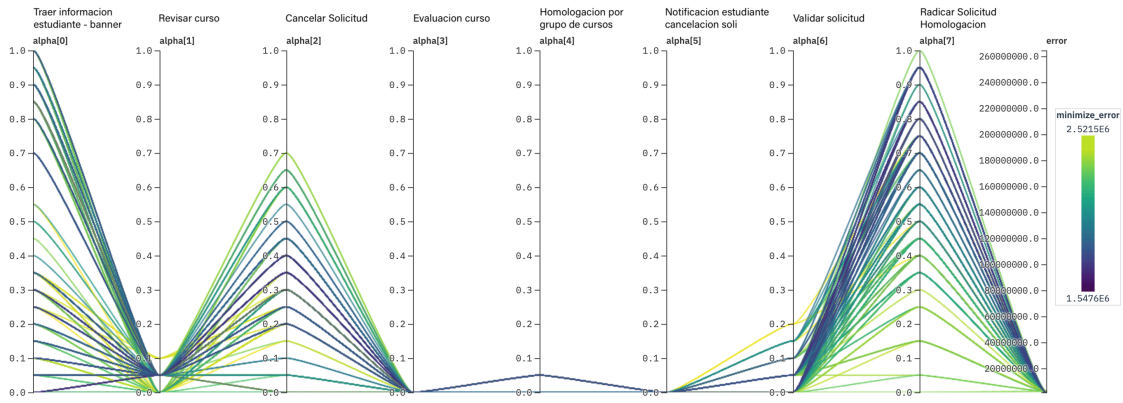


Figure 4.6: Alpha behavior filtered on the 20% best errors for NSGAII optimizer with $\delta = 0.05$ applied on the student credential recognition process.

The second genetic algorithm, NSGAII, explored 1876 alpha configurations and found similar results to those of MOGAII, depicted in Figure 4.6. The indecision previously encountered in defining α_2 and α_7 is presented also with this approach. Again, by performing a second optimization, it is possible to achieve better accuracy.

As it appears in Table 4.3, which represents the final outcomes, the three sub-optimal points reached by each algorithm are different, thus the derived start timestamps will profile distinct activity durations.

To assess the accuracy with which the three algorithm perform, three boxplots have been compared in Figure 4.7, one for each optimization algorithm. In order to build each boxplot, it has been computed the start timestamps through Equation 2.1 considering the alpha sets in Table 4.3. Then it has been calculated the differ-

	∂	Traer informacion estudiante -banner	Revisar curso	Cancelar solicitud	Evaluacion curso	Homologacion por grupo de cursos	Notificacion estudiante cancelation soli	Validar solicitud	Radicar Solicitud Homologacion
Local based search	0.05	0	0.1	0.05	0.05	0.05	0.05	0.55	0.95
MOGAII	0.05	0	0.05	0.4	0	0.05	0	0.05	0.8
	0.001	0	0.026	0.017	0.008	0.018	0.013	0.018	0.15
NSGAII	0.05	0	0.05	0.4	0	0.05	0	0	0.95
	0.001	0	0.051	0.344	0.017	0.027	0.006	0.014	0.867

Table 4.3: Final alpha configuration for each optimization technique student credential recognition process.

ence between those timestamps and the real ones, and finally the outcomes have been normalized on the activity durations (computed on the original log):

$$\frac{|t_s - \widetilde{t_s}|}{t_c - t_s} \tag{4.2}$$

where t_s and t_c are the actual timestamp when the activity started or completed, and $\widetilde{t_s}$ is the estimated value.

Figure 4.7 clearly shows how the genetic approaches outperform the local-base search one. Even if the median errors of the three algorithms are the same, the average error is reduced to 27% of the local-based search approach by MOGAII and to 23% by NSGAII. Studying the standard deviation of the estimation errors, it appears that the variation of the error is drastically reduced: MOGAII has a standard deviation amounting to 30% of the approach by Fracca et al. one, while NSGAII to 60%, meaning that the genetic approaches are able to decrease the variability of the estimation error.

Figure 4.7, 4.8, 4.9 do not show the outliers: there are cases (ca. 20% of the elements for all the optimization approaches) with an estimation error even beyond 100%. This behavior is due to the fact that, in order to keep the complexity of the problem feasible, the technique researches a single alpha parameter for each activity, based on the assumption that the waiting times for the same activity are similar. Nonetheless, this situation may not be reflected in real-life and for some of them, the estimation of the start timestamp does not coincide with reality.

Finally, it has been inspected if a larger search space for the alpha values would improve the performance of the genetic approaches. To this aim, the optimization with MOGAII and NSGAII has been repeated changing the granularity of parameter ∂ from 0.05 to 0.001. As it is shown in Table 4.3, in the case of MOGAII the smaller step allows an increase in the precision of the alpha values, furtherly improving the start timestamps estimations and reducing the average error of 12% (Figure 4.8). On the contrary, a smaller step seems to penalize the accuracy of NSGAII, which is no more able nor to reach the same sub-optimal as with ∂=0.05, neither a better one. This results can be explained by considering the multiple local minimums presented by the error function; it seems that in case of NSGAII a smaller step lets the optimization get stuck in a local minimum which does not furtherly minimize the objective function. However, as already mentioned, to say with certainty which is the best step, several optimizations should be performed and statistical studies should be carried out.

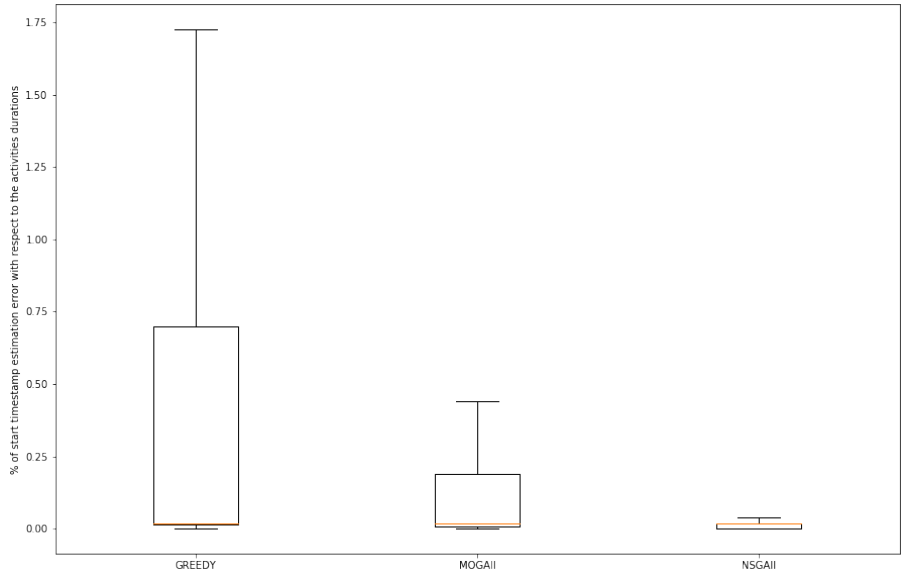


Figure 4.7: Comparison of start timestamp estimation performance of the three optimization algorithms in student recognition process.

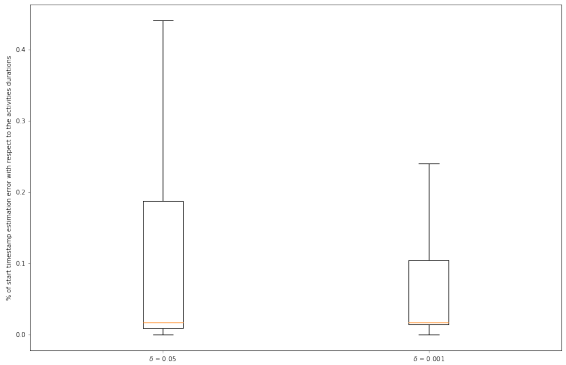


Figure 4.8: Comparison MOGAII with δ=0.05 and MOGAII with δ=0.001.

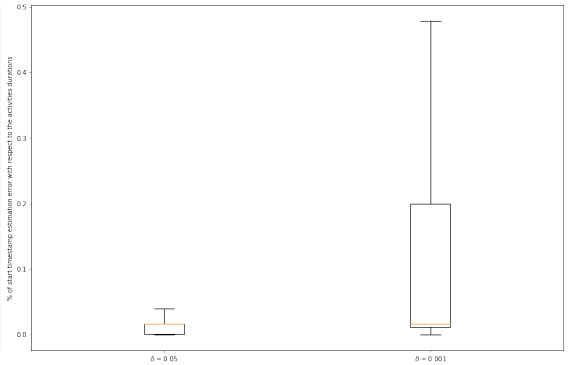


Figure 4.9: Comparison NSGAII with δ=0.05 and NSGAII with δ=0.001.

4.3.2 Case study: purchase process

The second case study is based on a synthetic event log[†] inspired by a real-life process. The number of traces, amounting to 608, is lower with respect to the previous case study, with 21 activities and 9119 events. The number of resources who contributes in the activities is 27.

[†]The event log is available at `http://fluxicon.com/academic/material/`

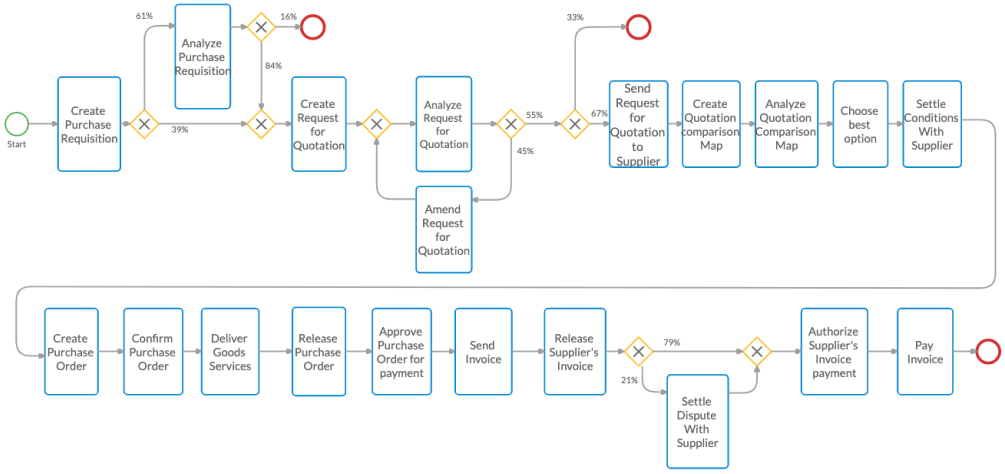


Figure 4.10: BPMN model and branch probabilities purchase process case study.

Start Timestamp	Duration	Trigger Count	Inter-trigger Timer
1/1/2017	10000	608	NegativeExponentialDistribution mean 19639.5

Table 4.4: Simulation parameters purchase process case study.

Role 0	Role 1	Role 2	Role 3	Role 4	Role 5	Role 6	Role 7
	Choose best option / Amend request for quotation / Release purchase order / Create purchase requisition / Analyze quotation comparison map	Analyze purchase requisition	Settle conditions with supplier / Create purchase order / Approve purchase order for payment / Send request for quotation to supplier / Analyze request for quotation / Create quotation comparison map	Authorize supplier's invoice payment / Pay invoice / Release supplier's invoice	Send invoice / Confirm purchase order / Deliver goods services	Create request for quotation	Settle dispute with suppliers /
24h/7 8 resources	24h/7 14 resources	24h/7 3 resources	24h/7 3 resources	24h/7 2 resources	24h/7 5 resources	24h/7 3 resources	24h/7 4 resources

Table 4.5: Number of resources per work shift and activities they are involved in purchase process.

Figure 4.10, depicts the BPMN model of the process, which has been extended with the simulation parameters, which are summerized in Tables 4.4 and 4.5.

The same modeFRONTIER workflow shown in Figure 4.2 has been exploited to run the optimization both with MOGAII and NSGAII. Each optimization has been carried out considering δ=0.05. In this case study, the search space is larger than the previous one, since it is spanned by 20 different variables (each referring to the

corresponding activity) and the granularity of the alpha is the same as before. It will be shown that, due to such complexity of the searching space and of the objective function, the algorithm will find several local minimums able to sub-optimize the target function. This complication is also linked to the dimension of the event log: given the low number of traces, fitting the distributions of activity durations can lead to inaccurate results that do not allow good accuracy in estimating waiting times and thus alpha values.

The first optimization run has been executed exploiting MOGAII, which explored 4814 different configurations. The parallel coordinate graph in Figure 4.11 obtained filtering the alpha configurations leading to the top 20% errors, shows that the algorithm converges to a specific point for the majority of the alpha parameters, but there is a small set for which it seems problematic to determine the best sub-optimal point. In fact, observing the color gradient, it appears that for α_0, α_8, α_9, α_{13}, α_{14} almost all the values available bring to the same total error. If, on the one hand, the large range of α_{14} is due to the fact that is the first activity of the traces, and as such any alpha value is acceptable since it is not possible to estimate the starting time, on the other hand, for the other activities it is necessary to run a second optimization, keeping fixed the other parameters to their best values aimed to explore a possible convergence not yet reached by the algorithm.

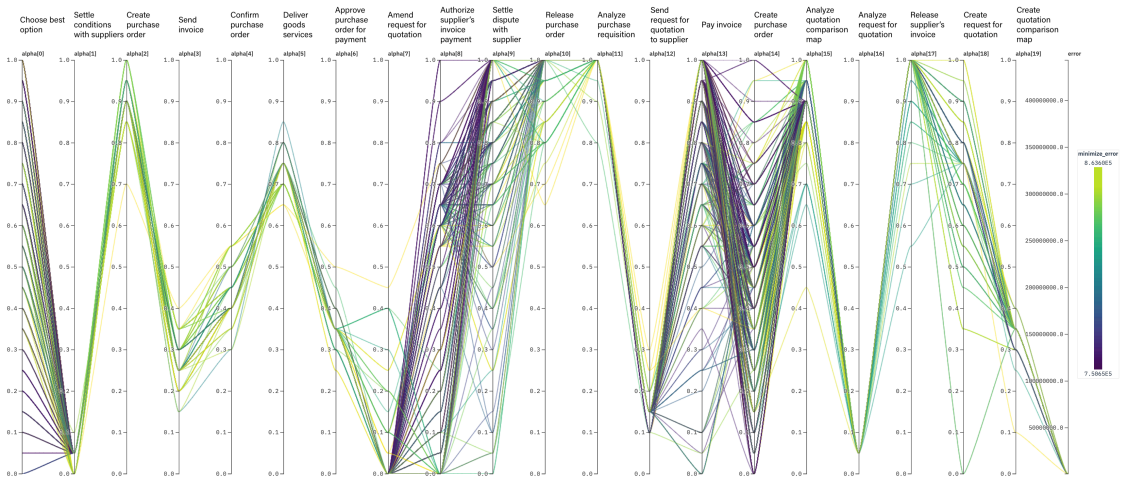


Figure 4.11: Alpha behavior filtered on the 20% best errors for MOGAII optimizer with $\delta = 0.05$ applied on the purchase process.

The second optimization run is capable to reduce the range of admissible values, as shown in Figure 4.12. The only exception is represented by the activity *Choose best option*, in which every alpha value equally contributes to final the error. A detailed inspection of the activity reveals that it is instantaneous and starts as soon as the previous activity concludes. As such, looking at Equation 2.1 it can be seen that for that kind of activity, the value of alpha does not influence the final result. Indeed, as the minimum start timestamp in which the activity could have started, *mintime(e)*, is computed as in Equation 2.2, the outcome is always equal to the final timestamp of the previous activity, by construction. Hence, given that the activity at hand is instantaneous, it follows that $mintime(e) = time(e)$. Thus, in this special case, Equation 2.1 becomes:

$$time(e') = \alpha(a) * mintime(e) + (1 - \alpha(a)) * time(e) = time(e) * (\alpha(a) + 1 - \alpha(a)) = time(e) \qquad (4.3)$$

In conclusion, the activity start timestamp is forced to be equal to the end timestamp, whatever alpha value is considered.

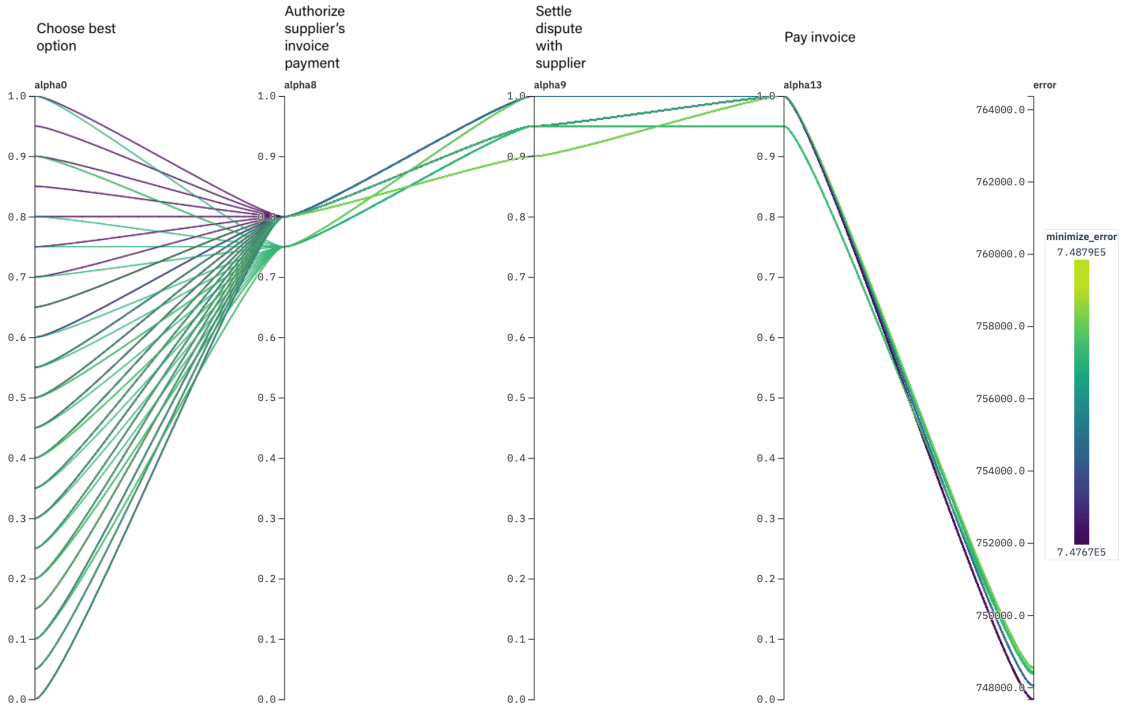


Figure 4.12: Second optimization of α_0, α_8, α_9, and α_{13} with $\delta = 0.05$ keeping fixed all the other parameters to their best values.

The final alpha values obtained are shown in Table 4.6.

For the case of NSGAII, the alpha values trend in objective minimization is described in Figure 4.14 by the parallel coordinate graph, which again has been filtered on the best 20% errors alpha configurations. This second method shows more noise in the identification of the right values. Nonetheless, the color gradient allows us to figure out a unique solution, represented in Table 4.6. Given the high noise in most of the variables, performing a second optimization keeping fixed some of the values does not bring further improvements. Hence, it has been explored an alternative method, shrinking the δ parameter to 0.001, but also this modification did not allow to reduce the indecision in the choice of the best values (Figure 4.14). The reason why this happens is linked, as already mentioned before, to the high complexity of the objective function: its non-convexity and its dependency on a large number of variables makes the algorithm find many alpha configurations able to reduce the target function.

	∂	Choose best option	Settle conditions with suppliers	Create purchase order	Send invoice	Confirm purchase order	Deliver goods services	Approve purchase order for payment	Amend request for quotation	Authorize supplier's invoice payment	Settle dispute with supplier	Release purchase order	Analyze purchase requisition	Send request for quotation to supplier	Pay invoice	Create purchase requisition	Analyze quotation comparison map	Analyze request for quotation	Release supplier's invoice	Create request for quotation	Create quotation comparison map
Local based search	0.05	0.1	0.15	1	0.65	1	0.7	0.9	0.55	0.15	1	0.95	1	0.8	0.95	0	1	0.1	0.85	0.95	0.1
MOGAII	0.05	0.6	0.05	0.9	0.3	0.4	0.75	0.4	0	0.8	0.95	1	1	0.15	1	0	0.9	0.05	1	0.8	0.3
	0.001	0.056	0.013	0.808	0.095	0.764	0.717	0.203	0.998	0.053	0.978	0.793	1	0.025	0.966	0.407	0.94	0.071	0.087	0.829	0.244
NSGAII	0.05	0.4	0	0.5	0.4	0.45	0.75	0.75	1	0	0.2	0.9	0.65	0.35	0.3	0.15	0.75	0.05	0.9	0.75	0
	0.001	0.276	0.341	0.91	0.936	0.368	0.395	0.01	0.661	0.908	0.582	0.924	0.948	0.401	0.158	0.81	0.937	0.043	0.569	0.348	0.107

Table 4.6: Final alpha configuration for each optimization technique purchasing process.

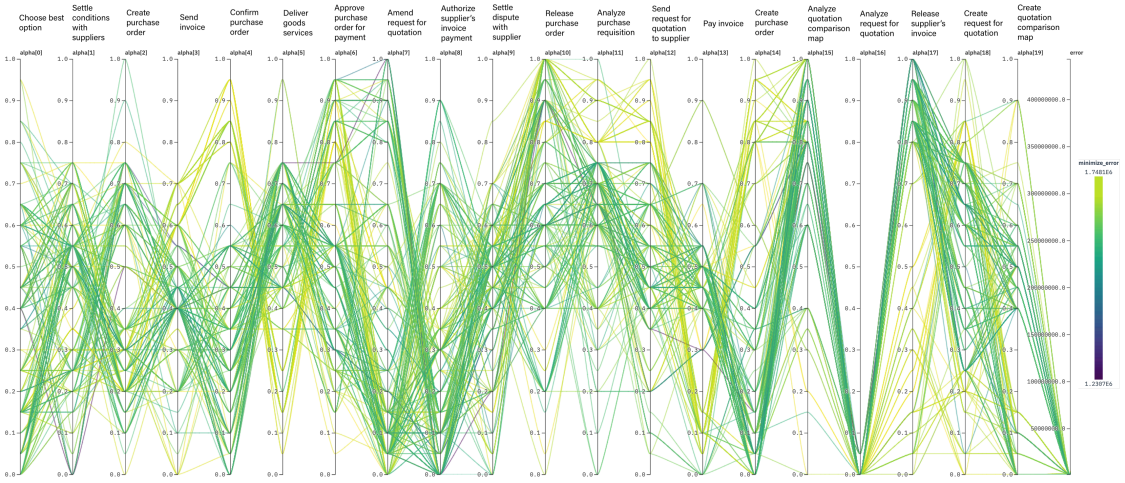


Figure 4.13: Alpha behavior filtered on the 20% best errors for NSGAII optimizer with $\partial = 0.05$ applied on the purchase process.

Figure 4.14: Alpha behavior filtered on the 20% best errors for NSGAII optimizer with $\partial = 0.001$ applied on the purchase process.

To perform a fair comparison with the results obtained by the local-based search approach, it has been con-

sidered the outcomes provided by MOGAII and NSGAII with ∂=0.05. Given the alpha sets in Table 4.6, the event log has been enriched with each of the start timestamps computed from the three approaches leveraging Equation 2.2.

In Figure 4.15 there are drawn three boxplots derived with the same reasoning of the previous section (Equation 4.2), one for each optimization approach. It is possible to appreciate the improvement brought by the genetic algorithms, which confirm the findings of the previous case study. The median error of the local-based search strategy is enhanced by 30% both from MOGAII and NSGAII. Also, the average error is improved by 53% by MOGAII and 62% by NSGAII. Concerning the variation range of the error, MOGAII reduces the standard deviation of 54% with respect to the target algorithm, whereas NSGAII of 39%.

As in the previous case study, Figure 4.15, Figure 4.16, Figure 4.17 do not show the outliers, which again amount to ca. 20% of the elements in all the optimization cases.

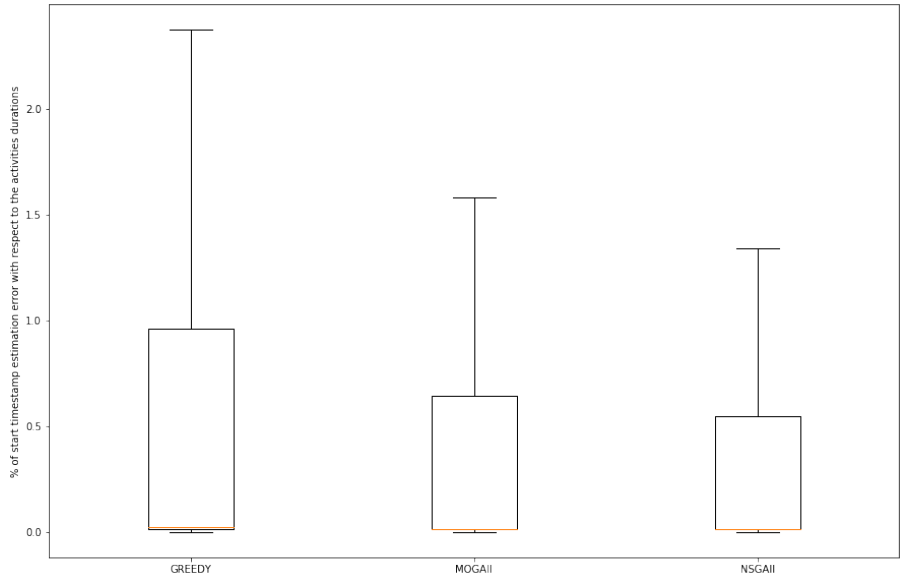


Figure 4.15: Comparison of start timestamp estimation performance of the three optimization algorithms in student recognition process.

Finally, as in the students' credential recognition case study, it has been assessed the performance of the algorithms modifying the granularity step of ∂ to 0.001. It is interesting noting from Table 4.6 that the step modification lets the algorithms converge in points that are far from the extracted ones with ∂=0.05. This is another confirmation of the function's non-convexity.

The graphs in Figure 4.16 and Figure 4.17 show how a larger searching space allows the algorithms to discover lower local minimums, furtherly improving the start timestamps estimation. While the median errors remain unchanged, MOGAII reduces the average error of 70% with respect to the smaller searching space, whereas NSGAII of 58%.

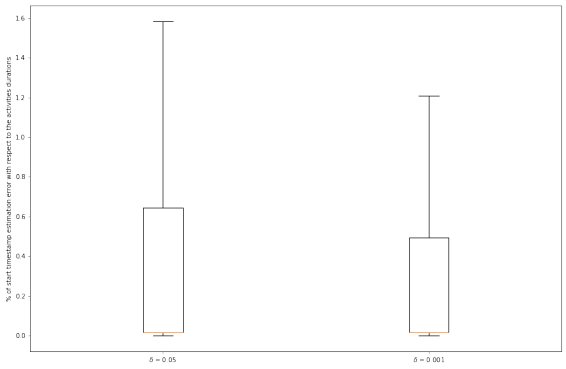


Figure 4.16: Comparison MOGAII with δ=0.05 and MOGAII with δ=0.001.

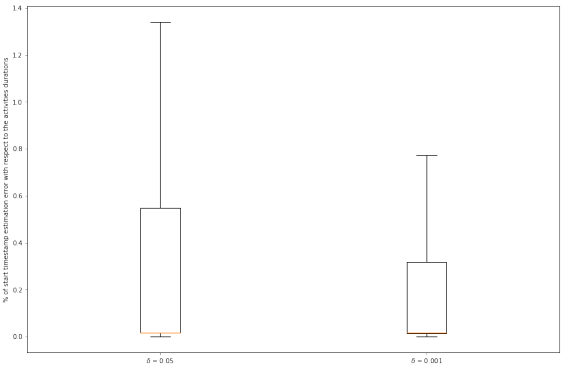


Figure 4.17: Comparison NSGAII with δ=0.05 and NSGAII with δ=0.001.

5 Case Study

In this chapter it is presented a case study based on a Tuscany hospital ED. In the first part, the data at hand and the process characterizing the ED are described. Then, the event log pre-processing, and the process model designed are introduced. Afterward, it is explained how the technique exposed in Chapter 4 can be leveraged to find the missing activities' start timestamps and to create a simulation process. Finally, there are proposed two 'what-if' scenarios aiming to explore some solution frameworks to address the critical problems of healthcare.

5.1 Data and process overview

The case-study event log refers to real-life data collected by the ED of a Tuscany hospital. Starting from January 2017 the ED has adopted a new information system that records every performed activity. The collected data have been anonymized before being used, and made available as a CSV file.

Nine months of process execution data have been gathered, describing the activities performed from 1st January 2017 to 5th September 2017. Each event contained in the dataset provides the case reference (*C*01_*IDUNIVOCOEPISODIO*), the patient reference in an anonymized version (*C*02_*IDUNIVOCOASSISTITO*), the patient's age at the arrival time (ETA_ACCESSO), the date of the access to the ED (*C*05_*ACCESSO*), the reported pathology at triage (C05B_PATOLOGIA_TRIAGE), the diagnosis (*C*05_*DIAGNOSI_PRINCIPALE*), the resignation outcome (*C*07_*ESITO_DIMISSIONE*), the activity executed (*C*100_*EVENTO*), the event timestamp (*C*101_*TIMESTAMP*1), and the activity's attribute (*C*103_*ATTRIBUTO*) which describes additional details about the activity performed. An extract of the event log in table format is shown in Figure 2.1.

The original dataset is composed of 41012 cases, 14 activities, and 511459 events. Each row represents a unique case event, and multiple rows could be linked to the same case and patient. Table 5.1 depicts in detail the activities registered in the dataset and some of the attributes related to those activities (not all the attributes have been

Activity	Attributes
Accesso	118 di altre regioni / Altro Elicottero Ambulanza 118 - con medico / Ambulanza 118 - senza medico Ambulanza privata / Ambulanza Pubblica Autonomo / Eliambulaza 118
Dimissione	Azzurro / Bianco / Giallo Rosso / Verde
PrestazioniPS	03.31, Rachicentesi / 1047, Visita di PS 1235, Sutura cute e sottocute etc.
Triage	Azzurro / Bianco / Giallo Rosso / Verde
Uscita	01 - Attesa Ambulanza / 02-Attesa posto letto 03 - Attesa Parenti / 04 - Attesa posto struttura esterna
Laboratorio Inizio / Laboratorio Fine	Esami-L00086316 etc.
Radiologia Richiesta / Radiologia Accettazione / Radiologia Esecuzione / Radiologia Refertazione	Radiologia Angio / Radiologia ECO Radiologia RMN / Radiologia RX Radiologia TAC
Consulenza Inizio / Consulenza Fine	Anestesia - Ospedale Dermatologia - Ospedale Pediatria - Ospedale etc.
Osservazione	Alta Intesità / Medio bassa intensità Obi / Obi pediatrica ospedale

Table 5.1: Activities and Attributes of the Original Event Log of the ED process.

reported in this table for the sake of brevity). Attributes give additional information about the type of activity performed by the patient. It will be shown their relevance in allowing the distinction of activity types in data pre-processing.

From the information presents in the dataset, it is possible to have an overview of the population age, the frequency of diagnosis, and the urgency codes. The patient population (Figure 5.1) is quite uniformly distributed across the different ages, with picks for infants (between 0 and 4 years), middle-age people (between 40 and 54 years), and elderly people (75-79 years).

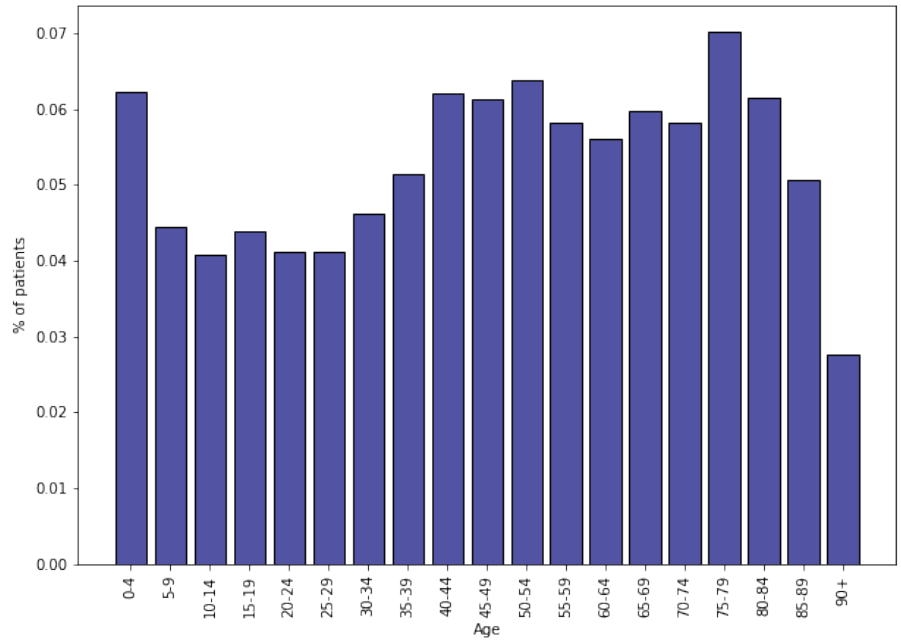


Figure 5.1: Patients' age distribution ED process.

Color	Description	Frequency (%)
Red	Interruption or impairment of one or more vital functions.	2.2
Yellow	Risk of impairment of vital functions. Condition with developmental risk or severe pain.	29.1
Green	Stable condition without evolutionary risk with suffering and relapse to the general condition usually requiring complex services.	53.3
Blue	Stable condition without evolutionary risk that usually does not require simple single-specialist therapeutic services.	14.3
White	Non-urgent problem of minimal clinical relevance.	1.1

Table 5.2: Emergency colors: description and patients' frequency (January-September 2017) in the ED process.

There are 40 different types of pathologies (e.g., *Limbs Trauma*: 6809 cases, *Other Symptoms or Disorders*: 6110 cases, *Abdominal Pain*: 4247 cases, *Thoracic Pain*: 2227 cases, etc.) and 1418 different diagnoses (e.g., *Inspection Visit*: 1722 cases, *Abdominal Pain*: 1482 cases, *Mild Nonconcussive Head Trauma*: 1194 cases, *Thoracic Pain*: 1132 cases, *Lower limb/hip/knee/ankle/foot contusion*: 1084 cases, *Fever of Unknown Origin*: 994 cases, etc.).

Patients arrive at the hospital in two different ways: 70% get autonomously, whereas 30% by ambulance. When they enter the ED, a triage-nurse registers their arrival. During daylight hours, there are separate triage and re-

sources that accommodate the patients according to their arrival type, while during the night hours just one nurse manages all the entries. Here patients are registered in the ED information system, with their personal data, the symptoms and an emergency color. There are 5 different emergency colors, each representing a degree of urgency. Table 5.2 shows the percentage of arrival cases for each color.

After triage, if necessary, a nurse can take a patient's blood sample. All the subjects are visited by a physician and a nurse. During this activity, the doctor releases the patient's diagnosis and directs the patient to subsequent therapies, tests, or observations. In low-gravity situations, the physician can even decide to immediately release the patient. More frequently, the subject needs deeper assessments. For example, he may need to undergo a radiological examination (*RX*, *TAC*, *Angio*, *RMN*, *ECO*). These tests are performed by radiologist physicians with the assistance of radiologist technicians. Once performed the test, the physicians issue a radiological report. Other patients may be placed under observation, with monitoring aimed at assessing the evolution of the clinical picture and completing the necessary investigations. In some cases, subjects are directed to specific consulting activities that involve medical specialists outside the emergency department. Another activity a patient can be subject to is grouped under the name *PrestazioniPS*: this embraces 83 types of treatments (e.g., application of a cast, skin and subcutaneous suture, application a of bandage, anaesthesia, etc.).

After the treatments, patients can be evaluated again and can be subjected to other analyses. Finally, the subject is discharged. He can leave the hospital in four different ways: he can wait for parents, wait for a bed in the hospital, wait for an ambulance for a transfer, or maybe be directed to another structure.

Since the ED information system does not record information about resources, their activity involvement is not reported directly in the event log, but it is provided at a later stage by the staff of the ED. Table 5.3 reports the resources of the ED, activities they are involved in, and the number per work shift.

Letto	**Medico**	**Infermiere**	**Infermiere Triage**	**Tecnico Radiologo**	**Radiologo**
Osservazione	Visita, PrestazioniPS, Dimissione	Prelievo, Visita, PrestazioniPS	Triage	Radiologia Esecuzione RX, RMN, Angio, TAC	Radiologia Esecuzione ECO, Radiologia Refertazione RX, RMN, Angio, TAC
24h/7 12 resources	00.00 - 8.00 2 resources	7.00 - 14.00 5 resources	7.00 - 14.00 2 resource (1 autonomous, 1 ambulance)	08.00 - 14.00 2 resources	08.00 - 14.00 1 resource
	8.00 -14.00 4 resoruces	14.00 - 22.00 5 resources	14.00 - 22.00 2 resource (1 autonomous, 1 ambulance)	14.00 - 20.00 2 resources	14.00 - 20.00 1 resource
	14.00 - 20.00 4 resources	22.00 - 7.00 4 resources	22.00 - 7.00 1 resource	20.00 - 8.00 2 resources	20.00 - 8.00 1 resource
	20.00 - 24.00 3 resources				

Table 5.3: Number of resources per work shift and activities they are involved in for the ED process.

5.2 Dataset preparation

Like almost all the real-life data, the provided dataset needs a pre-processing procedure in order to let the data be exploitable. One of the challenges of process mining consists in dealing with data quality issues. It often happens that the dataset contains missing information, such as timestamps or attributes. As previously mentioned (Section 2.2), the quality of the data has a significant impact on the results of the simulation, hence it is extremely important to face those issues.

Specifically, the provided data lacks 40206 events' attributes and 19962 patients' references: this missing information cannot be handled, thus we simply decided to accept those null values. One major problem from the process simulation point of view is related to the missing timestamps: specifically, almost all the activities recorded store only the final timestamp; since when dealing with process simulation it is necessary to know the activity duration, a specific technique (described in Chapter 4) has been exploited to estimate the starting timestamp of those activities. As it is shown also in Table 5.1, there are two activities that are split into *Inizio* and *Fine* events, which characterize the starting and ending events of these activities, respectively: *Laboratorio Inizio*, *Laboratorio Fine*, *Consulenza Inizio*, *Consulenza Fine*. For those activities, it is not necessary to estimate the starting timestamp, since both the start and end events are provided. Nonetheless, not all the events related to *Laboratorio* and *Consulenza* have this information. The former activity appears with a missing timestamp in 76 cases; given the low number of cases involved, we decided to simply remove those traces from the log. Concerning the latter activity, there are only 9144 traces showing both the start and end timestamps. All the other cases (9172) contain only the timestamp of *Consulenza Inizio*. Since exploiting the cited technique in Chapter 4 it is not possible to estimate the ending time of an activity while estimating at the same time the starting timestamp of activities that comes after it (there is an intrinsic dependency between those two values), in this case, in order to not throw away the information already given on the starting timestamp of the 9144 cases, the choice has been to estimate the duration distribution of this activity starting from the traces with both the start and complete timestamps information, and to sample from this distribution the duration of the considered activity for each activity with a missing final timestamp. When a sampled duration of *Consulenza* in accordance with the time interval allowed by the next activity's timestamp (the considered activity has to end before the timestamp of the next activity in the control-flow) is found, that duration is accepted and added to the start timestamp in order to find the ending time. If the sampling keeps extracting durations that are not allowed, meaning that the activity would end after the next control-flow activity, after a threshold fixed to 50 iterations the final timestamp of *Consulenza* is set as the mean time between the start and the next activity timestamp. Besides this, the dataset contains 4 traces showing more *Consulenza Fine* events than *Consulenza Inizio*; to deal with this problem we could have leveraged a similar technique as used in precedence, but given the scarce number of traces involved, we simply decided to eliminate them.

Another activity that deserves special attention is *Osservazione*: the timestamp of the related event refers to the starting time, while the end is marked by the timestamp of *Dimissione*. Thus, it has been merely added the end considering the timestamp of the respective case *Dimissione* activity.

Radiology activities are divided into 4 distinct events: *Richiesta*, *Accettazione*, *Esecuzione*, *Refertazione*. ED's experts who provided the dataset claimed that the only relevant events are *Esecuzione* and *Refertazione*. There are 2794 over a total of 21909 traces lacking the timestamps in those activities: we also removed those traces from the

dataset.

Lastly, a final check on the remaining events without timestamps has been carried out: the result showed 212 cases still containing lacking timestamps. The following dictionary shows the detail about the activity and the number of events involved: '*Triage Ambulanza*': 20, '*Visita*': 161, '*Triage Autonomo*': 33. Again, we decided to get rid of those traces.

After all these deletions, the number of remaining cases amounts to 35544, which represents a reduction of ca. 14% of the original event log.

Below there are shown all the pre-processing steps applied to the data, remarking what explained up to now and explaining some additional passages:

- Original log (41012 cases)
- Remove *PrestazioniPS* events that have the same timestamp of *Visita* in the respective case: these events are just a repetition;
- Remove fast-tracks cases about ophthalmic consulting (new log: 38630 cases): those cases are registered by the information system, but they are not handled by the ED;
- Rename *Triage* activity in *Triage Ambulanza* and *Triage Autonomo* accordingly with the corresponding event attribute;
- Specify the names of *Radiologia Esecuzione* and *Radiologia Refertazione* activities with the type showed in the event attribute: TAC, Angio, RX, RMN, ECO;
- Remove *Radiologia Accettazione* and *Radiologia Richiesta*;
- Remove *Uscita* activity;
- Remove *Accesso* activity;
- Remove cases lacking the timestamps in *Radiologia Refertazione* and/or *Radiologia Esecuzione* (new log: 35836 cases);
- Remove cases lacking the timestamps in *Laboratorio Inizio* and/or *Laboratorio Fine* (new log: 35760 cases);
- Add starting timestamp to *Laboratorio Fine* and rename it in *Laboratorio*;
- Remove *Laboratorio Inizio*;
- Add ending timestamp to *Osservazione* (taken from *Dimissione*);
- Remove 4 cases that have more *Consulenza Fine* than *Consulenza Inizio* (new log: 35756 cases);
- Remove cases still containing events without timestamps (new log: 35544);
- Fix *Consulenza*: for the cases that have the same number of *Consulenza Inizio* and *Consulenza Fine* events with all the timestamps available, create the unique activity *Consulenza* with both the starting and ending timestamp;
- For the cases with no timestamps in *Consulenza Fine*, or with more *Consulenza Inizio* than *Consulenza Fine*, sample the value of the ending timestamp and, if it is inside the allowed time interval, accept it, otherwise consider the average timestamp between *Consulenza Inizio* and the next control-flow activity;
- Rename *PrestazioniPS* in *Prelievo* when the event appears between *Triage* and *Visita*.

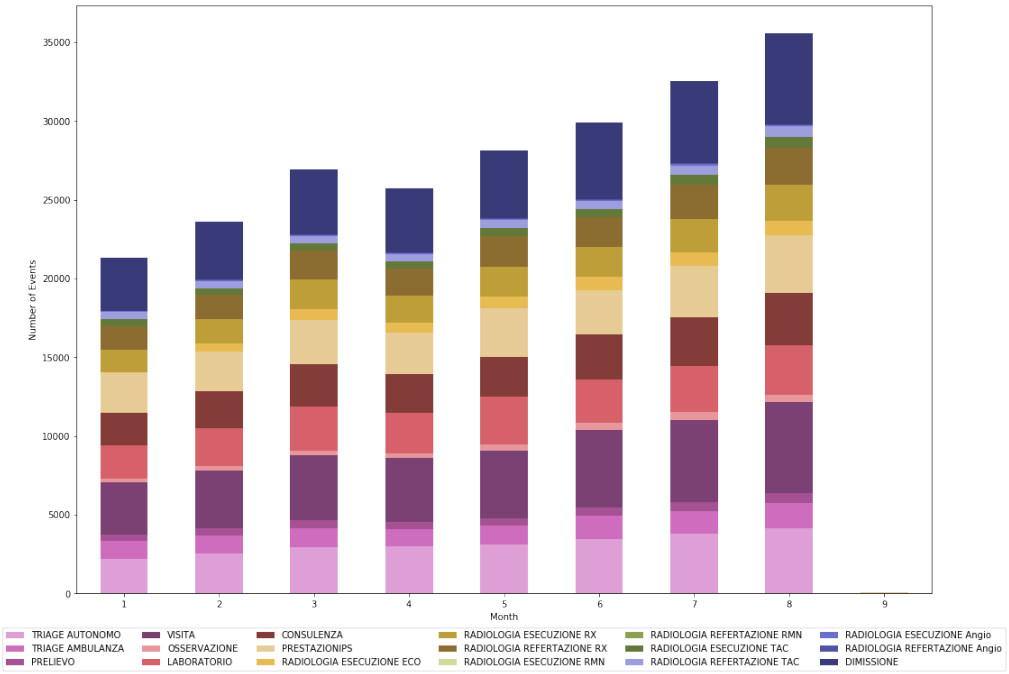


Figure 5.2: Events per month, grouped by activity types ED process.

Finally, we removed the outliers in trace duration. Some traces report events happening some months after the starting trace's time: this represents a system or process error and in order to not let the simulation be affected by such long times, we preferred to discard them. To this aim, we computed the the 99% quantile and removed all the traces with a duration higher than that value. This choice is forced by the fact that we did not want to remove traces with short durations, which domain experts' claimed that are allowed, while we wanted to drop all the cases with a statistical too long case duration. The final dataset is composed of 35359 traces.

Once pre-processed the data, statistics per month can be computed. Figure 5.2 shows the total number of events performed in the ED department per month, coloring them by the type of activities. The graph shows an increasing behavior and this can be explained by looking at the arrival rate per month in Figure 5.3. As it is shown, also the arrival rate grows in summer months. This leads to a pick of events in that period. The reason why there is this increasing behavior in the last months of the dataset may be connected to the fact that the event log at hand refers to a Tuscany hospital, which is a highly populated summer destination. The rise in arrival rate can be linked to the higher number of people present in that period. Unfortunately, we have no available data about the following years to inspect this seasonal behavior hypothesis.

5.3 Process model

After having applied the pre-processing steps, the dataset is now ready to be exploited to draw a model that depicts the control flow. The aim is to find a model that properly describes the patients' paths, starting both from the data

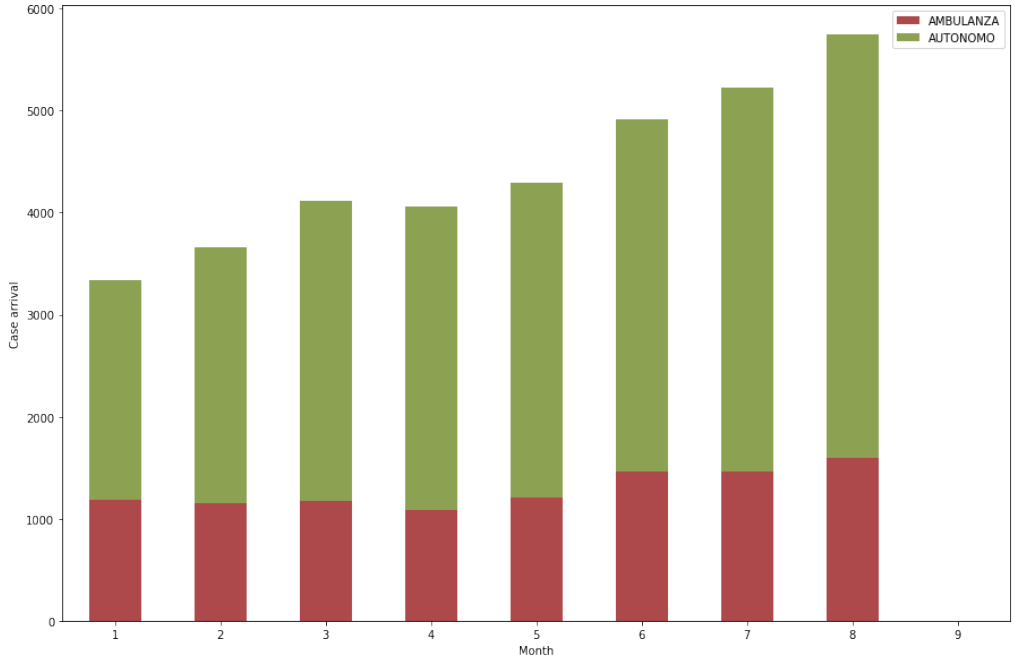


Figure 5.3: Patients arrival rate per month ED process.

and the suggestions received by the domain experts.

Initially, the process activities have been enriched with the event attribute, with the aim to design a model with activities specialized for each pathology sub-process. Unfortunately, this approach reveals several complications in designing the process. Firstly, the resulting process has the characteristic of a *spaghetti process*, that is an unstructured process in which the huge variety of sequences of events affects the trade-off between simplicity and precision discovering the process [20]. Furthermore, since roles are not specialized to each patient's pathology, but they simply carry out specific activities whatever it is the problem of the incoming subject, in this specific model setting the same resource should be shared between the different activities related to each pathology sub-process. Unfortunately, the LANNER L-Sim simulator allows just the creation of a unique queuing system for each activity to which is associated a specific resource, without the possibility of creating shared queues where patients with different pathologies needing the same treatment can queue up together. As a consequence, this prevents the development of a simulated process in which a sorted token arrival management is guaranteed.

An alternative approach is to separate the processes per color. The resulting designs, subdivided by color, are similar to each other and to maintain simplicity in the structuring of the process model, we decided to develop a unique model embracing all the patient types arriving at the ED. Besides this, keeping the distinction of activities per color is again problematic from the simulation point of view: the same queuing problem encountered in the first trial does not ensure the correct management of tokens priorities.

Since we are dealing with an emergency department, there are multiple possible paths a patient can undertake. With respect to the diagnosis revealed during the physician's visit, a subject can be directed to several different examinations. This is highly reflected in the event log, which shows a high number of process variants (7412

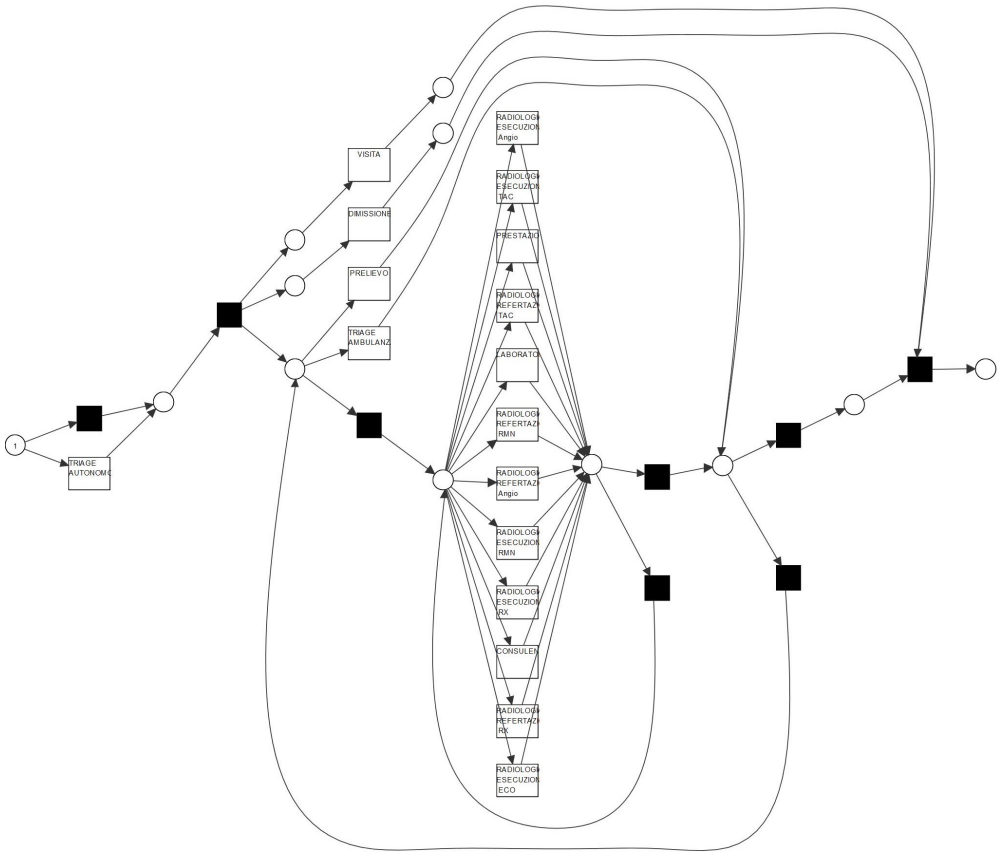


Figure 5.4: Petri net mined with Inductive Miner ED process.

variants). Since building a model that takes into account all the possible paths would be unfeasible, we tried to take advantage both of the information coming from the data and the process knowledge of the experts, who are aware of how the process is conducted.

Relying only on human opinions may not reflect what actually happens in reality. Thus a hybrid approach, data-driven and human opinion-driven, allows simplifying the discovery of the process and, at the same time, follows the outcomes of the data.

In order to exploit process discovery techniques, it is necessary to transform the ED dataset into an event log format. This format allows for the definition of traces that contain sorted events. Events are characterized by the name of the activity performed, the timestamp, and attributes. Since resources' information is received from domain experts and is not recorded by the information system, it is not included in the event log.

Once transformed the table format of the dataset into an event log, the first step is to mine a model in order to understand which is the general structure of the problem at hand. We exploited the Prom plug-in *Mine Petri-net with inductive visual miner*. The result is shown in Figure 5.4.

The model shows an evident issue: most of the activities are in parallel. This is due to the variability of the process paths present in the dataset. Since the process tasks are often performed in a different order, the algorithm

tends to place the activities in parallel. Clearly, this situation is completely unrealistic because a patient can not be involved in two or more activities simultaneously. Given the failure of the process discovery approach, we opted to draw the model from scratch, following the variants emerging from the dataset and the advice of the experts, who explained to us the entire process as described in section 5.1. Figure 5.5 shows the final structure of the BPMN model designed in Cardanit.

Exploiting the Python library PM4Py, we computed the fitness and the precision based on alignments, generalization, and simplicity of the model. The fitness outcome (95%) certifies the goodness of the model in reproducing most of the possible paths reported in the event log, while precision (71%) shows that the model accounts for a fraction of behaviors which are not seen in the event log. Generalization amounts to 98%, meaning that the resulting model will be able to reproduce future behavior of the process. Finally, the value of model simplicity is 65% which is a high value, considering the complexity of the process.

The choice of creating two different activities for *Triage* comes from a simpler configuration of the resources in process simulation. Since there are distinct nurses designated for the type of patient's access, it is convenient to keep this activity separated between *Autonomo* and *Ambulanza*. After the registration at triage, a patient may undergo a blood sampling (this possibility is represented by the *XOR-gateway prelievo*). The token (representing a patient in the ED) passes then to the activity *Visita*.

After the visit, the model shows firstly a XOR-gateway (*XOR-gateway dimissione immediata*) which takes into account that some patients can be directly resigned, and, after it, an AND-gateway. The activities following the latter gateway are performed in parallel. In order to keep a high alignment with the event log, we decided to structure the model drawing activities' free paths and/or loops in order not to force each token to perform all the parallel activities.

The first activity that appears in parallel with the others is *Osservazione*. About 10% of the patients are monitored in the ED and meanwhile they can be subjected to visits or other therapies. The remaining percentage does not need a bed for observation, hence the *XOR-gateway free path osservazione* before that activity and the free branch *no_osservazione* allow to skip it.

In parallel to *Osservazione* there is *Laboratorio*. This activity is executed outside the emergency department by resources not involved in the studied process, and it does not include the presence of the patient. Again, this activity can also not occur, thus a free path (*no_laboratorio*) allows to pass over it. However, when it does take place, it can also be repeated more than once in a unique trace. This behavior is allowed by the model by a loop (denominated *loop_laboratorio*).

The lower part of the model shows the last group of activities that can be carried out in parallel with *Osservazione* and *Laboratorio*. All the patients that are not resigned suddenly after the visit are subject to at least one of these tests/therapies. Indeed, the model does not include the possibility to skip them through a free branch. The *XOR-gateway activities* implies that those tasks can not be executed in parallel with each other, but they can be subsequently repeated. This behavior is structured through the *loop altre attivita* edge.

When all the parallel activities are completed, the token reaches the last activities, where finally it is fired and the process concludes.

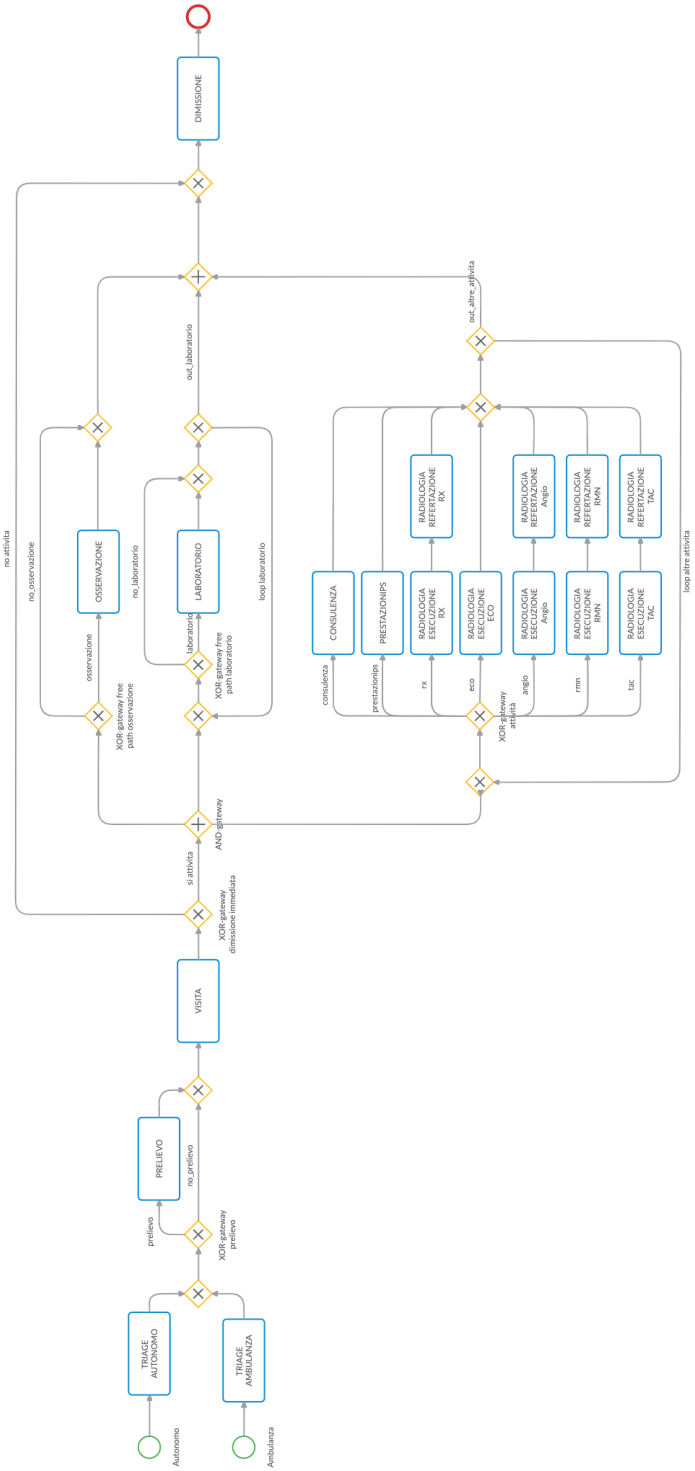


Figure 5.5: BPMN model for the ED process.

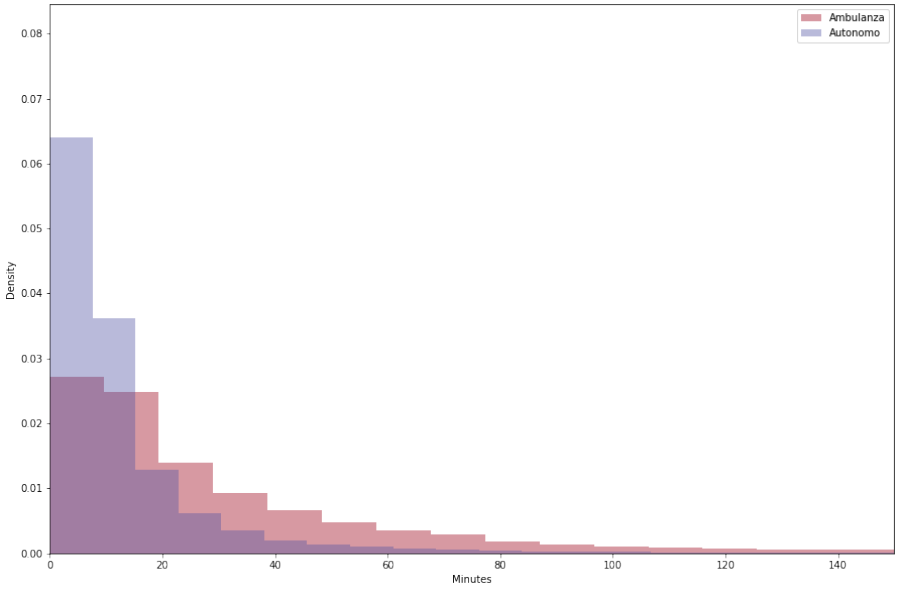


Figure 5.6: Inter-trigger timer distributions ED process.

5.4 Simulation parameters

In order to create a simulation process, it is necessary to define the simulation parameters. Firstly, it is possible to retrieve from the event log the starting date of the events: 1st January 2017. The duration of the simulation is set to a high number of days (in our case 10000) to ensure the entire development of the process. In particular, the trigger count parameter is set to 35359, i.e., the number of patients received by the hospital during the time period considered, and the simulation life span is determined just by the time necessary to conclude all the life-cycle processes of the created tokens. Since the BPMN model is structured with two starting events, two different trigger count parameters are defined: one for the ambulance (10322 tokens) and one for the autonomous arrival cases (25037).

The two inter-trigger timer parameters (one for each arrival method), which define the tokens' arrival time, are computed cycling on all the traces of the log and considering the time difference between the first event of two consecutive cases belonging to the same arrival class. The two resulting lists of tokens' arrival times are then fit to a series of distributions, and the most suitable ones are adopted to define the parameters. Figure 5.6 shows the distribution of the two inter-trigger timers. Autonomous arrival patients are characterized by an average inter-trigger timer of 14 minutes, whereas ambulance arrival patients of 34 minutes.

The choice not to introduce a calendar to the trigger count to model the increasing arrival behavior in the summer months shown in Figure 5.3 comes from the objective of the research, which stands in developing a model which resembles the general process of the ED. Since the available data span from January to September 2017, we are not able to understand if the growth is a due to a seasonal behavior or to other factors. Hence, we decided to set a unique trigger count parameter for the total duration of the process.

The next step is to define the number of resources, their calendars, and the activity they work on as described

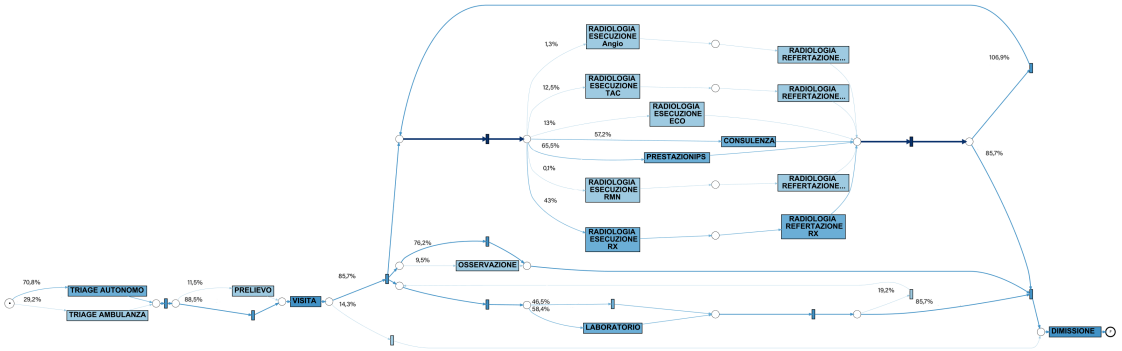


Figure 5.7: Branches probability XOR-gateways ED process.

in Table 5.3 in Section 5.1.

Afterward, the branches' probabilities for each XOR-gateway are declared. The Prom plug-in *Multi-perspective process explorer* has been exploited to discover them. Exploiting the Prom plug-in *Convert BPMN to Petri net (control-flow)* we convert the BPMN into a Petri net, which is the model syntax accepted by the Prom plug-in. Then, once given in input the resulting model, the tool returns the number of times an activity is performed along with the percentage for each branch of the split, computed as the ratio of the frequency of the activity and the total number of traces. Figure 5.7 shows the discovered probabilities.

The last step consists in defining the activities' duration parameters, namely the probability distribution for each activity. Since the event log does not contain the initial timestamps of several activities, it is necessary to estimate them before being able to find the correct durations. To this aim, we exploited the technique presented in Chapter 4. Its development for this case study is explained in detail in Section 5.5.

5.5 Case study's start timestamps estimation

Given the optimal results achieved by the genetic algorithm MOGAII with ∂=0.001, we decided to leverage it to perform the optimization of the error function shown in Equation 4.1 and estimate the start timestamp in the ED case study. To run the pipeline presented in Section 4.2, we structured the workflow in modeFRONTIER shown in Figure 5.8.

Besides the central CPython node which contains the Python classes needed for the creation of the BPSim document and the computation of the error, it receives in input another script which is used to activate the LANNER L-sim simulator (LSim), the event log, the BPMN model, a parameter which defines the type of optimization algorithm (funcion_error) to use and the vector of alpha. The output returns the error computed.

The optimization is run on a dedicated machine equipped with Intel Xeon E5-2667 v4 3.20GHz and 16 GB of RAM and lasts ca. 2 days, with a running time mostly influenced by the creation of the simulation process. MOGAII algorithm explored 4753 different combinations of the alpha set, which means it has developed the same number of simulations with which computing the error using Equation 4.1. It has to be noticed that, as in the previous case studies, the optimization should be run by considering a set of different random seeds for

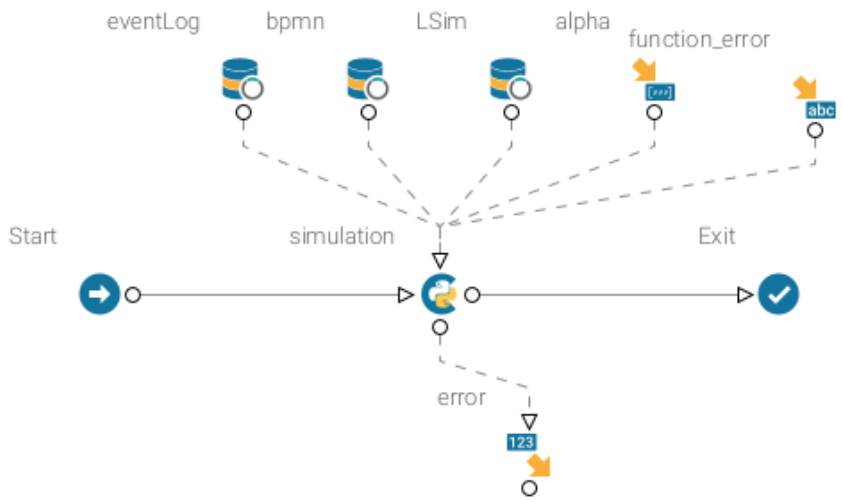


Figure 5.8: modeFRONTIER workflow for ED Process.

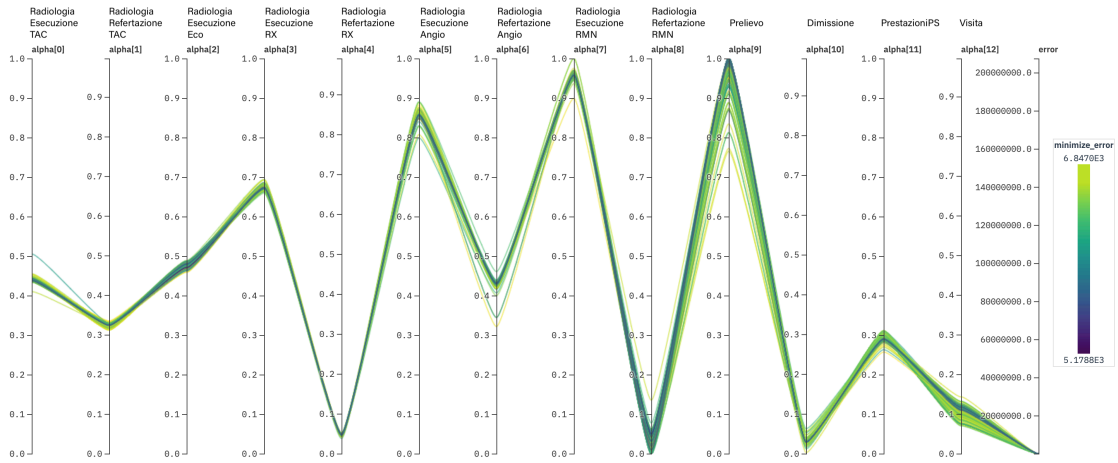


Figure 5.9: Alpha behavior filtered on the 20% best errors for MOGAII optimizer with $\partial = 0.001$ applied on the ED process.

each simulation process created given an alpha configuration thus reducing the deterministic setting given by the single run. However, this approach would exponentially increase the running time, hence, for the sake of time, we decided to run just a single simulation for each alpha combination.

The behavior of the optimization is shown in Figure 5.9, which depicts the 20% best errors alpha's trend. It is remarkable the accordance with which the best error values determine the value of alpha, which allows concluding that the algorithm reached a (sub)optimal point. The color gradient shows how, as the values of each alpha tend to the optimal one, the error decrease. The final alpha configuration for each activity is shown in Table 5.4.

Radiologia Esecuzione TAC	Radiologia Refertazione TAC	Radiologia Esecuzione ECO	Radiologia Esecuzione RX	Radiologia Refertazione RX	Radiologia Esecuzione Angio	Radiologia Refertazione Angio	Radiologia Esecuzione RMN	Radiologia Refertazione RMN	Prelievo	Dimissione	PrestazioniPS	Visita
0.438	0.324	0.471	0.673	0.048	0.856	0.43	0.957	0.032	1	0.03	0.289	0.117

Table 5.4: Alpha values for the computation of the start timestamps of the ED process.

5.6 Case study simulation process

Once enriched the original log with the best-estimated start timestamps, created the BPSim document with the process parameters, and built the process BPMN model we are able to develop an accurate simulation process. In this section, it is assessed the quality of the results in order to validate the simulated process. To counteract the deterministic outcomes given by the single simulation run, we launched the simulation 20 times, each with a different random seed. Every run simulates the same number of traces of the original event log. Each simulation has the same activity duration and XOR-gateway distributions from which the simulator extracts a sample and defines the different parameter settings. Finally, we compute some statistics about the outcomes obtained. Their correctness has been evaluated both by comparing the simulation models with the original log and by the domain experts' review.

Table 5.5 shows the results about the activities' frequencies obtained with the branches' probabilities considered in Figure 5.7. The statistics represent the average, the median, the first and the third quantile, and a confidence interval of 90% on the values computed on the 20 simulations.

The number of tokens processed by each activity in the simulated process resembles quite precisely the real one. Besides *Radiologia Esecuzione/Refertazione RMN*, which displays only ca. 30 events and does not influence the total performance of the simulation, the activity showing the highest difference is *Laboratorio*, with a frequency that on average differs by 6% from the original one. The average distance between the original and the simulated frequencies' activities is 2.1%.

Another piece of information validating the simulation is the activity durations. Unfortunately, we can not compare the results with the original log, indeed it is not possible to figure out the real values without knowing the starting timestamp. Nonetheless, a verification of the quality of the outcomes has been requested to domain experts, who confirmed the goodness. Figure 5.10 shows the activity durations boxplots. In this picture, we removed from the visualization the outliers which squeeze the visualization in order to have a clear view of the results. It can be identified the average time in green and the median in orange. It should be noticed that the average activity's duration in some of the activities is highly influenced by outliers. This can be claimed by studying their distance with the median value, which depicts the central outcome of the series.

Since the activities' duration is inferred from the original log enriched with the initial activities' timestamps, their statistics allow an understanding of the quality of the estimation on the timestamps of the start events done with the optimal alpha set. The majority of the activities are characterized by median durations in accordance with reality, as confirmed by domain experts. The exceptions are represented by:

- *Radiologia Esecuzione RMN* which shows both a high mean and a high median compared to the expected one. The reason why this happens is linked to the low frequency of the activity. Since there are few samples, it becomes complex for the algorithm to build the right distribution for the duration of the activity and the result does not reflect reality.

	Real Log	Simulated Logs					
Activity		**Avg**	**Median**	**Q1**	**Q3**	**CI_min**	**CI_max**
Triage Ambulanza	10318	10318	10318	10318	10318	10318	10318
Triage Autonomo	25041	25041	25041	25041	25041	25041	25041
Prelievo	4070	3975.8	3975	3934	4010.5	3958.99	3992.61
Visita	35359	35359	35359	35359	35359	35359	35359
Osservazione	2880	2881.1	2883	2842.75	2911.75	2866.96	2895.22
Laboratorio	21872	20699	20670	20563.75	20733.25	20633.80	20704.20
Radiologia Esecuzione ECO	5154	4980.3	4981	4952.25	5018.25	4966.59	4994.01
Radiologia Esecuzione TAC	4126	4057.95	4059	4029	4075	4044.01	4071.89
Radiologia Refertazione TAC	4128	4057.95	4059	4029	4075	4044.01	4071.89
Radiologia Esecuzione Angio	445	444.65	444	439.75	453.5	440.81	448.49
Radiologia Refertazione Angio	445	444.65	444	439.75	453.5	440.81	448.49
Radiologia Esecuzione RX	14854	14436.75	14447	14338	14449.25	144003.53	14469.97
Radiologia Refertazione RX	14889	14436.75	14447	14338	14449.25	144003.53	14469.97
Radiologia Esecuzione RMN	30	27.8	28	21	32.25	25.70	29.90
Radiologia Refertazione RMN	30	27.8	28	21	32.25	25.70	29.90
PrestazioniPS	23365	23105.05	23082	22957	23267	23042.10	23168.00
Consulenza	21308	20792.95	20816	20727.25	20914.25	20746.17	20839.73
Dimissione	35359	35359	35359	35359	35359	35359	35359

Table 5.5: Comparison activities frequencies original log vs simulated log ED process.

- *Visita and Prelievo* seems to have swapped values. In particular, domain experts affirm that the *Visita* should last ca. 30-40 minutes, while the *Prelievo* ca. 5 minutes. This error can be connected to the fact that these activities are sequential and the algorithm is not able to correctly balance their durations. Furthermore, *Prelievo* is performed in 71.5% of the cases to high emergency patients' (Red and Yellow patients), with which urgency is essential. Domain experts claimed that in those cases, resources report the data in the information system after having completed all the necessary steps for the patient, and this behavior can insert some noise in its correctness.

The high variance appearing in *Consulenza* can be explained considering that it groups several types of this activity. A consultancy can be carried out for different reasons and accordingly to that, its duration can change.

Similar reasoning is valid for *Osservazione*. A patient remains monitored for a median time of 8 hours, but it happens to have fast releases subjects or cases which need to be supervised longer. Subjects in observation for some days highly affect the average duration which has a value equal to 16 hours.

On the other hand, the variability of *Radiologia Esecuzione Angio* is an intrinsic characteristic of the exam itself. Depending on the region of the body in which angio radiology is carried out, the duration can vary.

One of the most important factors influencing the quality of an ED is the patients' waiting time because it is an indicator of the congestion of the structure. It often happens that subjects have to attend long queues before being served.

Given that the original log does not allow us to explore this aspect, we exploited the simulations to understand its behavior. We collected the average total amount of hours of waiting time per month (Figure 5.11) between the 20 simulations and the average and median time waited by patients' (Figure 5.12). The first graph is shown to confirm the stability of the simulation. As it can be noticed, there are no increasing queues as the total number of patients arriving at the ED grows. Indeed, the total waiting time per month is rather steady throughout the

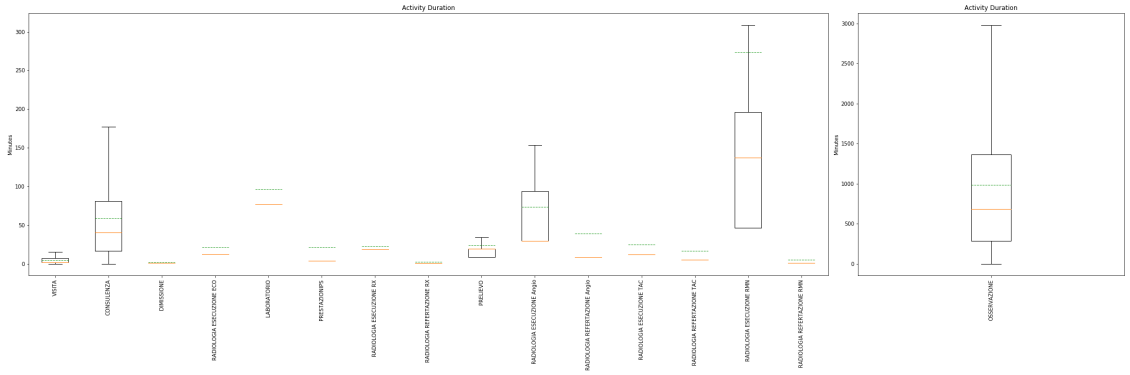


Figure 5.10: Activities duration boxplots ED process simulation.

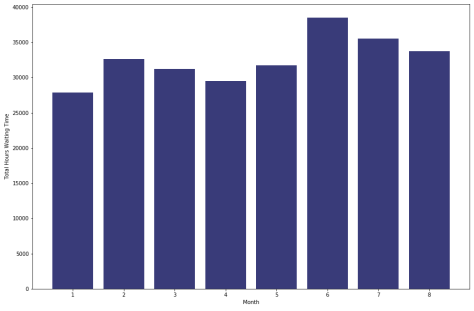


Figure 5.11: Total waiting time hours per month ED process simulation.

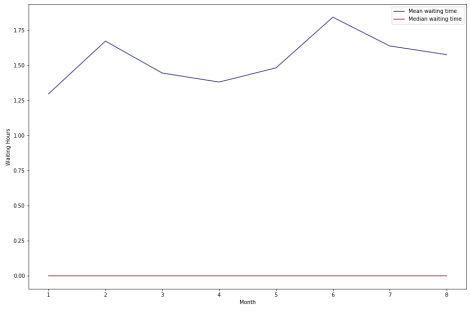


Figure 5.12: Average and median waiting hours per month ED process simulation.

duration of the simulation. From the second graph, we can see that the waiting time has an average of 1 hour and 30 minutes a median value of less than a minute.

We report the same graphs with a weekly x-axis (Figures 5.13, 5.14). The objective of these charts is to illustrate the variability of the waiting times inside the same month. This fluctuation is caused by the kind of activities performed and the number of arrival patients a week.

A statistic that is of fundamental importance for the efficient management of the entire ED is related to the detailed waiting times per activity. The boxplots in Figure 5.15 depict the outcomes of the simulations and give the opportunity to understand which tasks represent the process bottlenecks. The majority of the activities have both median and mean waiting times of a few minutes. The problems emerge when a token is directed toward radiological exams. If on the one hand the execution of the exam, performed by a radiologist technician, does not require long queuing, on the other hand, the drafting of the final report involving a radiologist, represents the main process bottleneck. The reason why the process shows this behavior can be attributed to the number of resources allocated to each activity. As shown in Table 5.3, in each task shift the number of technicians is twice the number of radiologists. Thus, even if as shown in Figure 5.10, the radiological exams last more than the reporting activity, since they are carried out by more resources, they require less waiting time. One way to tackle this issue

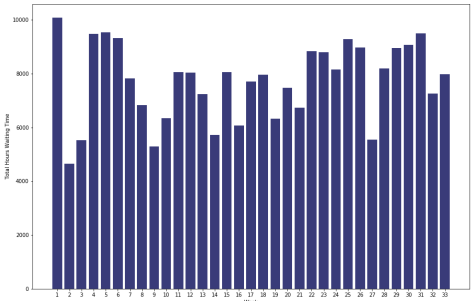


Figure 5.13: Total waiting time hours per month ED process simulation.

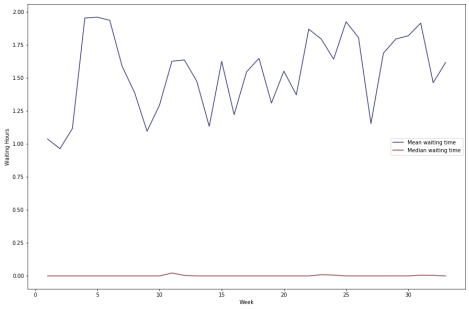


Figure 5.14: Average and median waiting hours per week ED process simulation.

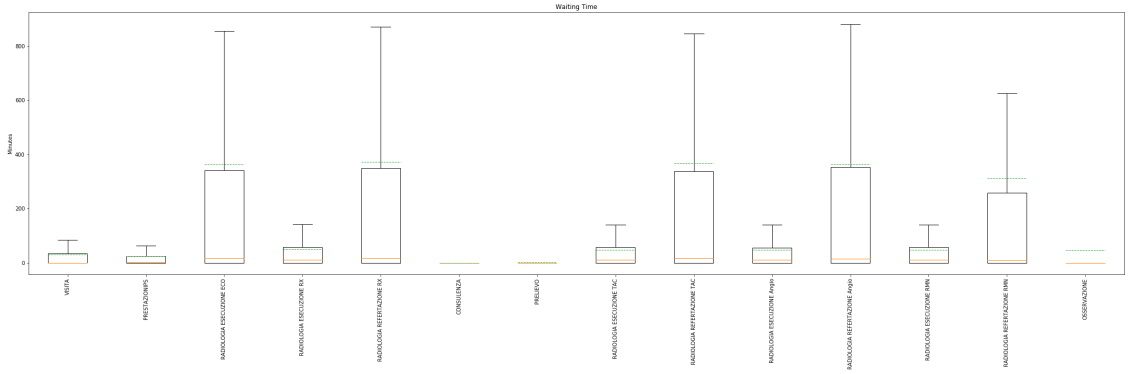


Figure 5.15: Activities waiting duration boxplots ED process simulation.

can be to simply increase the number of radiologists.

Given the waiting times and the activities' durations, another outcome that can be validated on the original log is the case duration. Figure 5.16 and 5.17 reproduce a comparison between the 20 simulations and the original log. The latter graph shows a logarithmic scale with a focus on the pick of the distributions. The large variety of trace durations observed in the graph is caused by the different treatments that patients need.

The median case duration in the original log is 149 minutes, whereas in the simulated ones is 186 minutes. In simulated logs, cases last 20% longer than in reality, which is an acceptable result given the initial conditions of the event log.

All the performance indicators we described in this section demonstrate the high accuracy of the simulation processes developed. Given the valid results achieved, it has been possible to develop various 'what-if' scenarios reflecting plausible process alternatives.

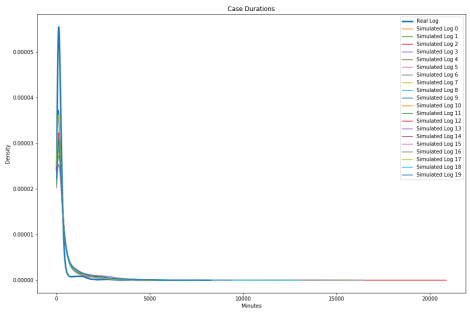


Figure 5.16: Original log vs simulated logs case duration densities ED process.

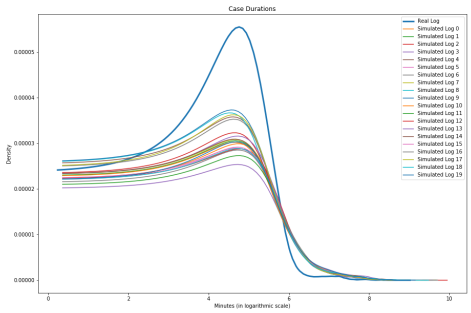


Figure 5.17: Original log vs simulated logs case duration densities with logarithmic x-axis ED process.

5.7 What-if scenarios

In this section, we explore two different 'what-if' scenarios frameworks to investigate how different organizational settings can impact reality. The main aim is to propose some approaches to counteract the emerging problems in healthcare. As it will be shown, some assumptions are made to build the different scenarios which represent only a few examples of the possibilities that process simulation offers to healthcare managers to optimize their processes. All the models presented in this section can be easily customized according to the healthcare managers' requirements.

One of the key performance indicators of the efficiency of a hospital is the case duration, i.e., the amount of time lasting from the admission to the resignation of a patient. As a consequence, hospital managers are required to take strategic decisions aiming to decrease as much as possible the patients' permanence inside the ED structures.

For this reason, both studies have as objective the research of possible solutions to reduce the case durations of the hospital's emergency department. All the reasoning is based on the assumption that, given similar conditions (activity durations, inter-trigger timer, number of patients, etc.), the most important effects are due to modification of waiting times. Indeed, it will be shown how specific modifications to the ED settings would drastically produce benefits to the patients' waiting times and as a consequence to the case durations.

One of the possibilities to reach the declared objective is to develop fast tracks that allow certain patient categories to be re-directed to reserved departments. Inspecting the age distribution in Figure 5.1, it appears that the ED receives a high quantity of pediatric subjects: more than 10% of the total cases. For this reason, it is proposed a scenario simulation in which a pediatric preferential pathway is introduced in the ED, in order to analyze how the process would react to the patients' reduction.

The second scenario proposes a multi-variable optimization aimed to examine how costs and waiting times can vary accordingly to different settings of resources. Since this problem has a set of possible solutions, it is difficult to find the right trade-off without the opinion of decision-makers. Therefore, two different resource configurations will be inspected: the first one aims to cut the total costs, while the second has as objective the reduction of the case duration, through the downsizing of the waiting times. Another assumption made on this 'what-if' scenario is that the objective is to reduce as much as possible the total waiting times, without weighting the impact this

would have on the single activities. This strong hypothesis is due to the lack of information about the degree of importance in respecting specific waiting times for each activity. However, it will be shown how this problem can be faced by proposing alternative plausible solutions. In addition, it will be introduced an approach in which it is set a threshold limit to the average waiting time of each activity, leading to discarding non-feasible solutions. This technique is developed to show another methodology with which the optimization can be performed. Also this technique can be effortlessly customized with desired boundary conditions.

5.7.1 Pediatric fast track

In order to build a simulation scenario, a new process model has to be developed. Figure 5.18 shows how the previous case study model has been modified to this aim. The pediatric cases, which can be identified with the relative event attribute, begin their process through a specific start event accordingly to their arrival type: *Ambulanza Pediatria* or *Autonomo Pediatria*.

Moreover, a new XOR-gateway is used to direct these subjects to another department. The new pediatric patients routing directs the subjects outside the ED and re-accepts them just before the resignation.

Those patients who are not directed to the preferential pathway proceed along the *no pediatria* branch and they follow the canonical process.

As the process changed, also the simulation parameters have to be re-computed. Four different inter-trigger timers and trigger counters have to be found, one for each start event. The event log is characterized by 4243 pediatric cases, of which 3880 arriving autonomously and 363 by ambulance. Concerning the other arrival cases, we can simply subtract from the previous case study start trigger counts the pediatric cases, obtaining: 21161 autonomous and 9955 ambulance arrivals.

The distributions of the inter-trigger timers are computed by separately considering the pediatric and non-pediatric cases. The technique exploited to find the arrival distributions follows the same approach used in section 5.4: the timestamps of each *Triage* activity is collected for each distinct group and the difference between successive events is computed. Since the non-pediatric cases group all the ED patients but children, the frequency with which they arrive is higher than the pediatric ones. On average, a *normal* patient arrives every 35 minutes by ambulance and 16 minutes autonomously. In pediatric cases, every 16 hours and 90 minutes, respectively.

In order to let the added gateway route the pediatric case to the reserved department, it is necessary to mark those tokens coming from the pediatric access in such a way that a property added to the gateway allows to recognize and address them to the right path. Our choice has been to sign the pediatric tokens with a property parameter equal to 1, while all the others as -1. As the token arrives at the gateway, if it is labeled as 1 it takes the branch *pediatria*, whereas all the others go through the other activities.

All the remaining simulation parameters have not been changed with respect to the case study ones.

Particular attention has been reserved for the gateway probabilities study. In the previous section, we claimed that it is not allowed by the limitations of the simulator on the queuing system to distinguish the activities according to some patient attributes (e.g., color, diagnosis, pathology). For this reason, we decided to structure the simulation model also without characterizing different activities for the different patient types. This is reflected also in the configuration of the tokens, which represent patients without specializing the different categories. This approach turned out to be problematic in the 'what-if' scenarios realization. Since activities and tokens are patient types agnostic, it is not possible to understand *a-posteriori* what percentage of subjects belonging to a certain cate-

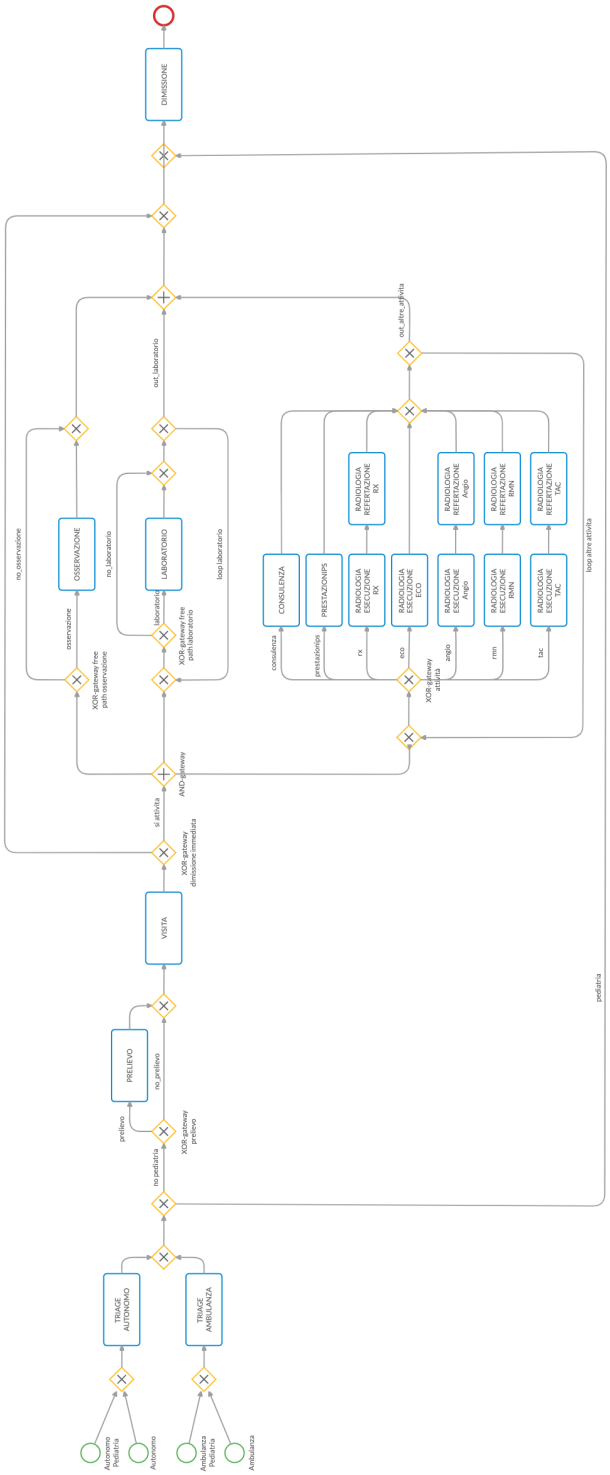


Figure 5.18: BPMN model for what-if scenario with pediatric fast track.

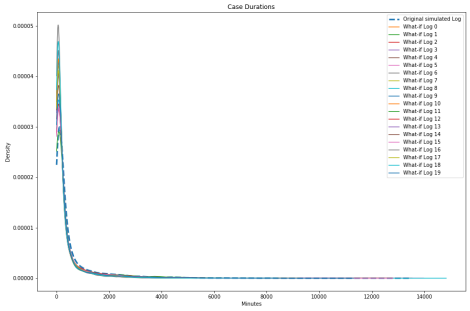


Figure 5.19: Case duration densities *normal settings* simulation vs what-if scenario.

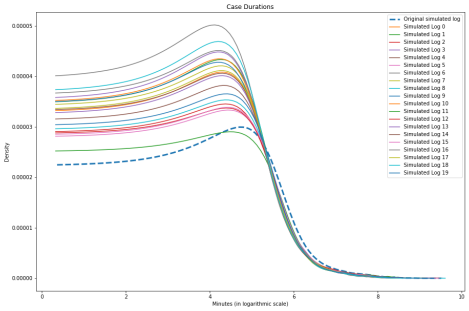


Figure 5.20: Case duration densities *normal settings* simulation vs what-if scenario with logarithmic x-axis.

gory performs a given activity. Given this, we decided to keep the previous case study XOR-gateways distributions (Figure 5.7), so that the reduction in the number of incoming patients is then reflected also in the activities frequencies.

The simulation parameters, the new process model, and the same event log enriched with the initial start timestamps computed with the alpha set in Table 5.4 allow the creation of the BPSim document, which in turn is given in input to the LANNER L-Sim simulator to create the simulated process.

The simulation is again launched changing the seed 20 times in order to counteract the deterministic settings given by the single run.

In the remainder of this subsection, we present a comparison of the outcomes between the *normal setting* simulation process and the 'what-if' scenario, inspecting whether the structural change may provide benefits in terms of case duration.

The graph in Figure 5.19 shows a comparison between the average case duration density of the original simulation and the densities of the 'what-if' scenario. In particular, the dashed line, representing the normal simulated process, has been obtained as an average of the 20 simulations performed in the previous section. From the higher picks, which can be better appreciated in the logarithmic scale of Figure 5.20, it is evident how the densities of shorter case durations are higher for the considered scenario. This allows an understanding of the effectiveness of the inserted fast track. Precisely, it emerges a median case duration reduction of 35%, passing from 3 hours and 6 minutes in the original case to 2 hours in the new one.

To have a more detailed view of the improvement, Figure 5.21 shows the comparison of the different percentiles values between the original scenario and this new one. In order to discover these results, the case durations characterizing each quantile have been calculated on the original simulation, and the percentiles for each scenario computed (Table 5.6). Since the yellow line is always higher than the blue one, it can be concluded that the new scenario allows for a case duration reductions. The first percentile (Q 0.1) groups 9% more traces in the new scenario, meaning that 19% traces have a case duration shorter than 20 minutes. Similar reasoning can be done for the other results. As a general trend, the plot confirms the result of shortened case duration already discovered in the precedent graphs.

Since the case duration corresponds with the sum of the performed activity durations and their respective

Quantile	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Case durations	20 m	1 h 10 m	1h 34 m	2 h 14 m	3 h 6 m	4 h 30 m	6 h 40 m	11 h 9 m	1 d	8 d 13 h

Table 5.6: Case duration quantiles.

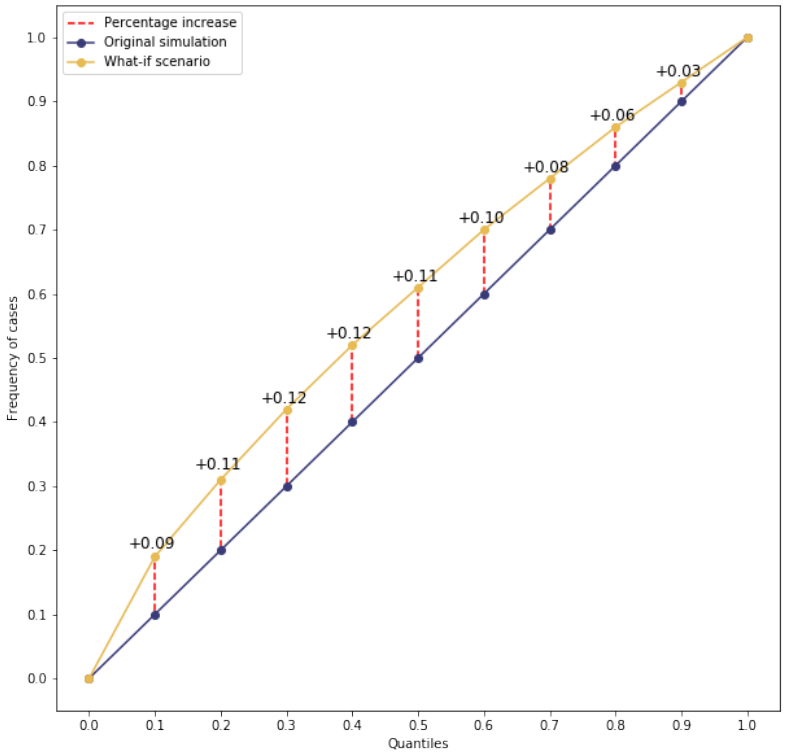


Figure 5.21: Comparison case duration quantiles original simulation vs what-if scenario pediatric fast track.

waiting times, having considered the same activity duration as the original simulation, the diminishment obtained is caused principally by the waiting time. In the following, it is inspected the activities' waiting times in order to understand their different behaviors and how they impact the case durations. Figure 5.22 shows how the total waiting time per month drops compared to the actual ED configuration. An average decrease of 30% of the total original waiting times per month is experienced in the new ED configuration. As a consequence, also the average waiting time per month drops remarkably (Figure 5.23).

Even if the non-characterization of tokens and activities by patient types does not allow us to carry out a precise analysis of the real impact of the pediatric cases removal on each activity, we present in Figure 5.24 a comparison of how activities' waiting times shortened. This reduction, given our limitations, depicts how a change in the quantity of arrival tokens similar to the number of pediatric cases observed in the event log affects the waiting durations of each activity. The steepest decrease is observed in the radiology activities, which however still remain the bottleneck of the entire process.

Another important information is the effect of the new configuration on resources. Figure 5.25 shows a comparison between the original and the new setting of the ED for all the roles. We grouped the statistics about the

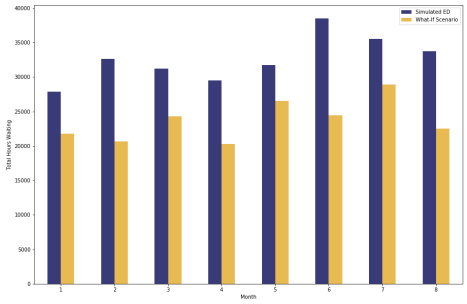


Figure 5.22: Activities total waiting times per month ED *normal settings* simulation vs what-if scenario.

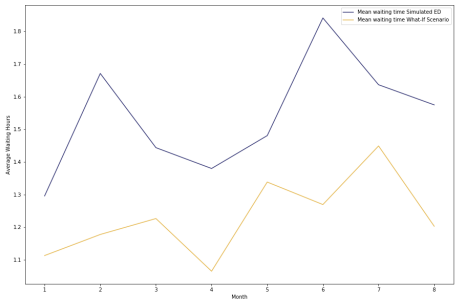


Figure 5.23: Average and median activities waiting times per month ED *normal settings* simulation vs what-if scenario.

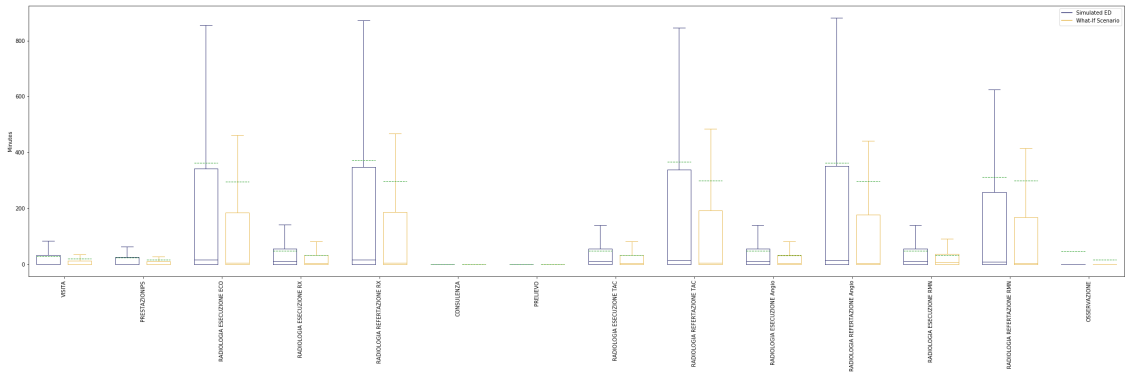


Figure 5.24: Comparison activities waiting times ED *normal settings* simulation vs what-if scenario.

different business hours in unique resources, which express the average activity time and the average inactivity time. The pie charts depict a general decreasing activity time. *Infermiere Triage* is the only resource not impacted by the modified ED structure because it is involved just in the activities *Triage Ambulanza* and *Triage Autonomo*, which are the only activities still performed by all the subjects.

Besides the beds exploited in observation activity, the resources receiving the greatest benefit from the ED renovation are those working in the radiology tasks. Indeed, their average degree of utilization decreased of 6%.

The outcomes of the 'what-if' scenario proved the effectiveness of opening a pediatric fast track in decreasing the waiting times, and as a consequence the case durations. The resources appear less overloaded and patients are more efficiently served.

However, opening a new department to receive just the pediatric case may require not only large investments in terms of structures, but also new dedicated resources and the return on the investment can take a long time. For this reason, in the following section, there are explored some short-term and cost savings solutions.

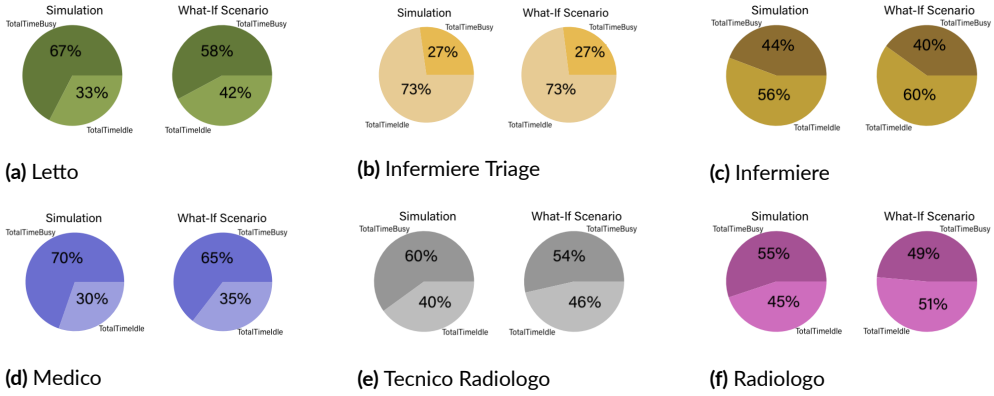


Figure 5.25: Comparison degree of utilization resources Simulation vs What-If scenario.

5.7.2 Costs and waiting times optimization

While one of the most critical problems to be faced in EDs is the case duration, there are other performance indicators hospital managers have to take into account when taking strategic decisions. In particular, they not only are required to offer efficient services but also to stay within their established budgets. For this reason, the objective of this second 'what-if' scenario is to research if there exist alternative resource settings which allow achieving a similar case durations cutting the total costs, or maintaining the same costs and drecrease the case durations. Since all the parameters, but the resources configuration, exploited for the developed 'what-if' scenario are left unchanged, the difference in case durations is predominantly impacted by the waiting times. Given this, all the reasonings aimed to reduce the case duration will be conducted on understanding the effects of the waiting times on it.

It will be shown how the results of the optimization highly depend on the choice of the objective function considered. The problem has been structured with the aim to show a general approach with which healthcare managers can leverage process simulation to carry out process optimizations. As already mentioned, the structure of the problem can be customized to find out insights into different aspects. Accordingly to the target of the research, the objective functions and/or their dependencies can be changed and obtain the desired results.

In order to explore several resource arrangements and find the best ones, it has been exploited a multi-variable optimization algorithm, MOGAII, aiming to minimize both total costs (ε_1) and total waiting times (ε_2) as functions of the number of resources. Equation 5.1 shows the expressions of the objective functions, where $w(i)$ is the total waiting time of activity i, G is the total number of days of the simulation, $h(m)$ is the daily work shift of resource m, and $c(m)$ is his hourly cost. The former explicitly depends on the quantity of resources and their cost per hour, the latter expresses the total activities waiting time, which depends just implicitly on resources.

$$\begin{cases} \varepsilon_1 = G \sum_{m=0}^{n} h(m) * c(m) \\ \varepsilon_2 = \sum_{k=0}^{i} w(i) \end{cases} . \tag{5.1}$$

Table 5.7 lists the roles, their hourly costs and the number of resources involved in each work shift. The hourly

cost has been obtained by considering an average position salary derived from Indeed.com*. The last column describes the intervals of the number of resources, defined considering their original configuration. These parameters can be modified according to the availability of resources and budgets of the emergency department.

The algorithm begins by extracting a random population of 120 resource combinations from established ranges (see column 'Range of resources' in Table 5.7), defining the optimization starting points. For each of the 120 resource combinations, the process model is simulated and the total waiting time and cost is computed. Afterwards, the 120 resource combinations are altered, with the constraint that each can be modified at most 40 times. This means that at most $120 \times 40 = 4800$ configurations are tested. Finally, it creates a frontier containing all non-dominated solutions. Those optimal points reveal the best explored combinations in terms of costs and waiting times. In particular, while exploring the available points, the algorithm finds several settings characterized by same waiting times or same costs and selects as best just those configurations which carry out the minimization of both.

The necessary components to run the process simulation are the process model and the simulation parameters, which are the same as the case study's ones (Figure 5.5, section 5.4), except for the number of resources that change at each optimization step. Let's notice that the optimization has been run without changing the random seed, which would have allowed to increase the stochasticity of the simulations. This choice is due to the time needed to run the entire optimization: the total run without changing the seed last ca. 1 day on a dedicated machine equipped with Intel Xeon E5-2667 v4 3.20GHz and 16 GB of RAM. If for each resource configuration explored, twenty different simulations scenarios would had been run, the time needed to find out the final result would explode. Keeping fixed the random seed, exactly the same simulation parameters (except for the number of resources) are used, and this would introduce a small bias error. Nevertheless, the objective of the optimization is to find a general idea of the best resource configurations. Once found the solutions and chosen the one of interest, in order to compare the results with the original simulation, twenty simulations with appropriate random seeds have been carried out, with the aim of introducing process stochasticity.

As in the start timestamp estimation optimization approach in Section 4.2, a modeFRONTIER workflow has been structured (Figure 5.26). Its skeleton is similar to the previous ones. As inputs it takes the process model, one combination of resources (*alpha_resources*), and the script needed to enable the LANNER L-Sim simulator. The CPython node contains the Python classes that allow the creation the process simulation and errors computation, which is the output of the workflow. The BPSimpy Python library enables the automatic BPSim document creation used to produce the simulation process.

Although the optimization is limited to exploring 4800 points, it stops after 3924 iterations because it is not able to furtherly improve the found solutions. The 70 non-dominated solutions are represented in Figure 5.27.

As a general trend, the total waiting time decreases only when increasing the total cost. However, it is interesting noting how those variables vary according to the type of resources involved. The graph shows points in which the total number of resources is higher than other ones, but the total waiting time is larger. This is due to particular role configurations in which there are involved resources that have less impact in decogestioning the ED.

An example is depicted by the two circled points. Their resource configuration is expressed in Table 5.8. It can be seen how the total waiting time of the second resource setting is lower even if the number of people involved

*`https://it.indeed.com/`
Indeed is an employment worldwide website.

Role	Hourly cost	Work shift hours	Current number of resources	Range of resources
Letto	30€	All day	12	[8, 16]
Medico	50€	8 - 14	4	[2, 6]
		14 - 20	4	[2, 6]
		20 -24	1	[1, 4]
		00 - 8	2	[1, 5]
Infermiere	20€	7 - 14	5	[2, 8]
		14 - 22	5	[2, 8]
		22 - 7	5	[2, 6]
Infermiere triage Ambulanza	20€	7 - 14	1	[1, 3]
		14 - 22	1	[1, 3]
Infermiere triage Autonomo	20€	7 - 14	1	[1, 3]
		14 - 22	1	[1, 3]
Infermiere triage	20€	22 - 7	1	[1, 3]
Tecnico Radiologo	30€	8 - 14	2	[1, 5]
		14 - 20	2	[1, 5]
		20 - 8	2	[1, 5]
Radiologo	50€	8 - 14	1	[1, 5]
		14 - 20	1	[1, 5]
		20 - 8	1	[1, 5]

Table 5.7: Resources cost, current and range number of resources per role work shift.

is inferior. This is due, in this specific example, to the higher beds quantity, which allows a reduction in queue of activity *osservazione*.

An essential information emerging from the graph is that the original resource configuration is dominated by several other solutions (highlighted by the shaded portion), meaning that it is not an optimal one in terms of the objective functions considered. Hence, it is possible to find a resource setting that allows achieving better total costs and/or total waiting times performance, thus reducing case durations.

In the remainder of this subsection, we study two different resources configurations: the first one represents an alternative which allows achieving the same performance in terms of total waiting times and case durations as the original one, but diminishes total costs, whereas the second one allows maintaining the same costs, but drastically reduce the total waiting times and case durations.

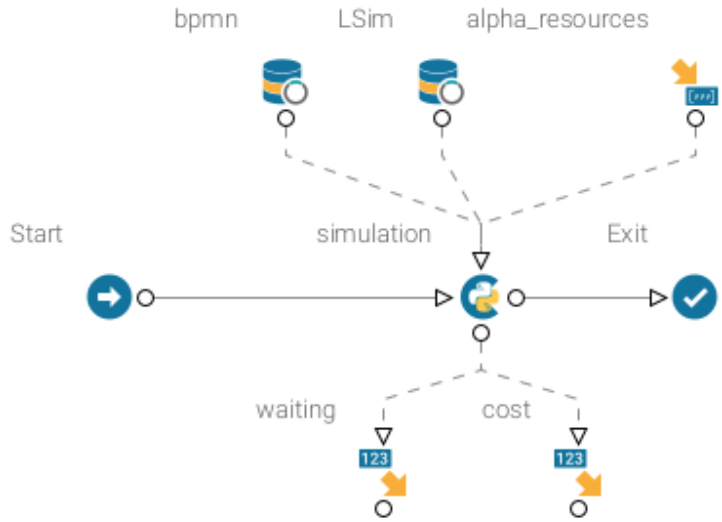


Figure 5.26: modeFRONTIER workflow for cost-waiting time optimization.

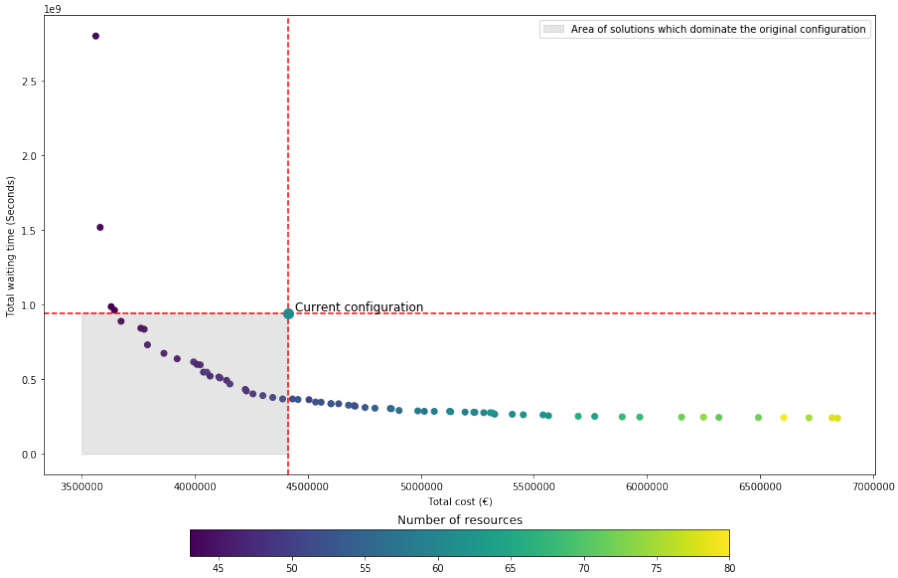


Figure 5.27: Non-dominated solutions waiting time vs cost optimization.

5.7.2.1 Similar waiting time resources configuration

In this scenario it is proposed a solution to cut the actual total costs of the ED, maintaining the same case durations as the actual ones. The graph in Figure 5.27 shows that by navigating along the dashed red horizontal line, it is possible to identify configurations characterized by the same total waiting time but different costs compared to the original configuration. The intersection with the function drawn by the non-dominated solutions identifies

Letto	Medico turno 1	Medico turno 2	Medico turno 3	Medico turno 4	Infermiere turno 1	Infermiere turno 2	Infermiere turno 3	Infermiere triage ambulanza turno 1	Infermiere triage autonomo turno 1	Infermiere triage ambulanza turno 2	Infermiere triage autonomo turno 2	Infermiere triage turno 3	Tecnico Radiologia turno 1	Tecnico Radiologia turno 2	Tecnico Radiologia turno 3	Radiologo turno 1	Radiologo turno 2	Radiologo turno 3	Total Number resources	Total cost	Total waiting time (sec)
13	5	4	3	5	6	5	6	1	1	1	2	3	5	5	5	2	5	3	80	6.6e6€	2.42e8
16	5	4	3	5	6	4	6	1	1	1	1	1	3	5	4	2	4	3	75	6.71e6€	2.41e8

Table 5.8: Comparison resources configurations.

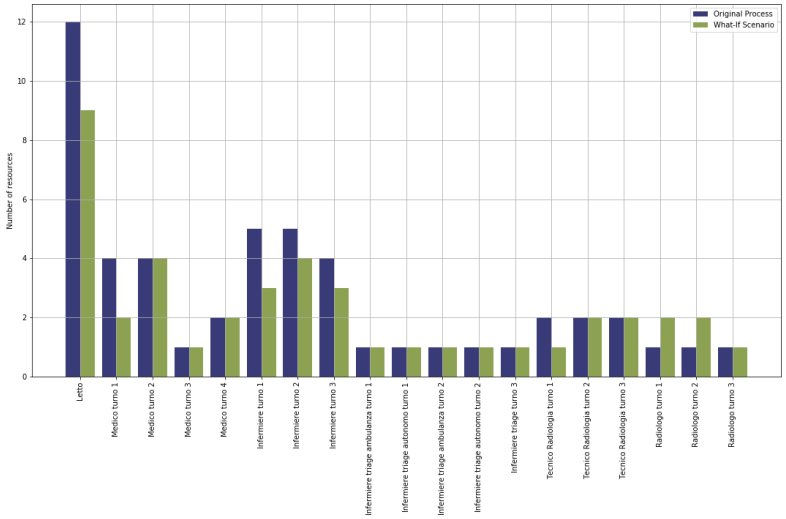


Figure 5.28: Comparison what-if scenario vs original number of resources ED process.

the best resource setting, which minimizes the total costs for the same waiting times.

The objective of this section is to understand if the non-dominated resource configuration identified can represent a valid alternative to the actual one. To this aim, since the optimization realized just 70 possible optimal configurations and no one of these has exactly the same waiting time as the original one, the nearest point to the intersection of the function with the horizontal straight line with a slightly lower total waiting time will be selected.

Given the configuration represented in Figure 5.28, a collection of 20 simulations is created in order to carry out statistics on them. In fact, to contrast the deterministic outcomes given by the single simulation run, a different random seed has been set for each simulation.

To confirm the case durations similarity between the original simulation and the 'what-if' scenario, there are proposed two graphs (Figure 5.29, Figure 5.30) comparing the case duration densities in both the cases. In particular, for the original setting, it has been considered the average case duration density, whereas for the 'what-if' scenario a representation of all the 20 outcomes has been depicted. The overlapping original and new densities testifies to the similarity between the durations. The small deviations between the 'what-if' case durations densities are due to the randomness of the process.

The graphs in Figure 5.31 and Figure 5.32 show a comparison of the total and average waiting time per month of the original configuration and the new one. As expected, they have similar behavior. Nonetheless, they do not perfectly replicate. This is due to the stochastic behavior of the simulations. However, considering even these small fluctuations, the total average waiting time remains at 1 hour and 30 minutes.

While, as it has been shown, the waiting times and case durations are similar, the new configuration allows

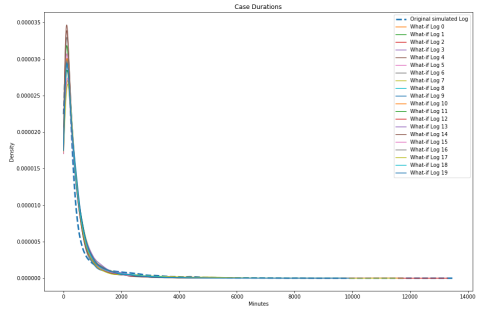


Figure 5.29: Case duration densities *normal settings* simulation vs what-if scenario with x-axis in logarithmic scale.

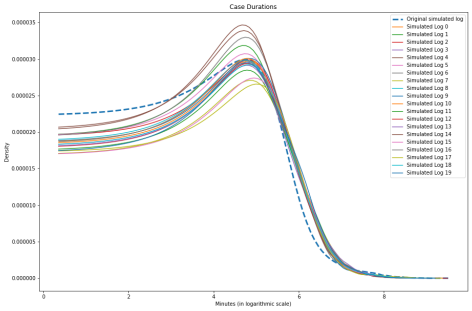


Figure 5.30: Case duration densities *normal settings* simulation vs what-if scenario with logarithmic x-axis.

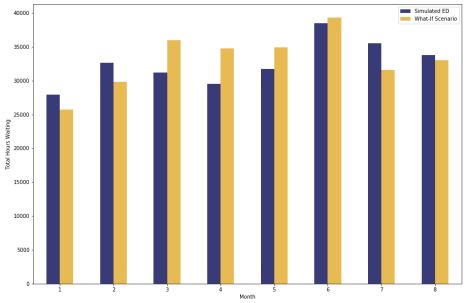


Figure 5.31: Comparison activities total waiting times per month ED *normal resource setting* simulation vs *new resource setting* what-if scenario.

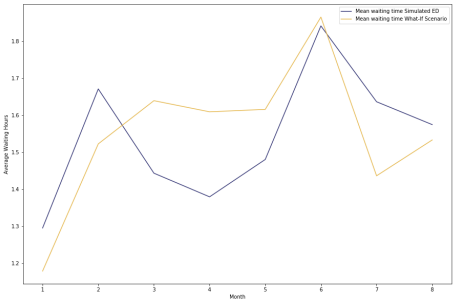


Figure 5.32: Comparison average and median activities waiting times per month ED *normal resource setting* simulation vs *new resource setting* what-if scenario.

cutting more than 16% of the total costs, passing from a daily cost of 18080€ to 15120€. This is made possible by a different resource setting in which it is increased the number of resources for the activities creating the bottlenecks of the process and reducing those in excess.

In section 5.6 it has been highlighted how the activities related to the radiological reports constitute the main problem of the process, creating long queues and slowing down the entire patients' paths. At the same time, there are activities in which there are lower waiting times or no queues at all. This behavior suggests that there are resource overflows in certain activities, while in others there is the necessity to increase the capacity. Resources re-organization is of fundamental importance because it allows cutting several activities costs in which there is no necessity and investing in others, which reveals scarce productivity.

One of the ideas proposed by the considered setting is to diminish the number of nurses in all work shifts. The predominant change happens in the first work shift, in which the simulation suggests changing the workforce from 5 to 3 people. In the other two work shifts, just one nurse seems to be in excess in the new configuration with respect to the original one. As for the nurse, also the physicians need a reorganization in the first turn, passing from 4 to 2. What emerges from this analysis and looking at Figure 5.33 is that this modification does not intensely

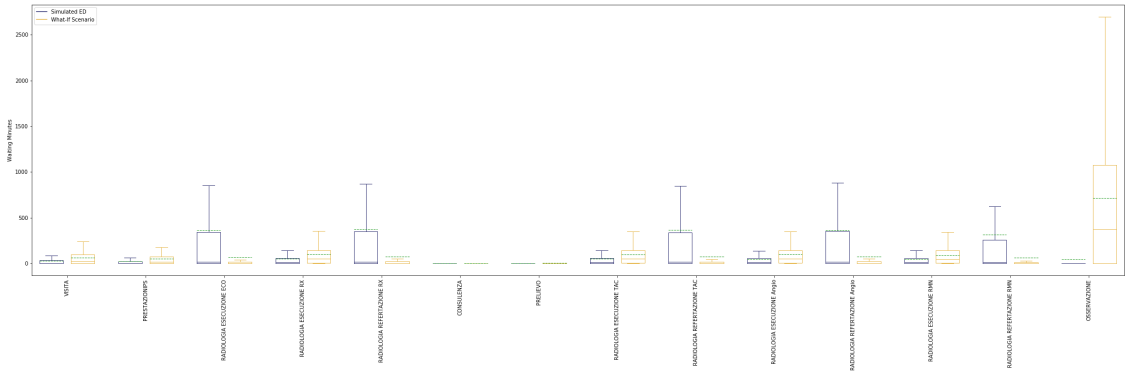


Figure 5.33: Comparison activities waiting times ED *normal resource setting* simulation vs *new resource setting* What-if Scenario.

influence the waiting times in the activities in which they are involved. In fact, *Visita*, *PrestazioniPS*, and *Prelievo* still do not suffer of long queuing. At the same time, this reduction accounts for a drastic hourly saving, amounting to 130€ (ca. 6% of the actual cost).

The slight rise in the values of the radiological tests execution is due to the downsizing of radiologist technicians in the first work shift: differently from the original one, in which two people are involved, the new configuration requires just one of them.

The bottleneck of the process in this new scenario becomes the activity *Osservazione*, in which now just 9 beds are available. The majority of the total waiting time is now shifted to this activity, with a mean of 900 minutes, i.e. 15 hours. It is clear that a situation like this is not acceptable in an ED, but this problem arises because of the reduced frequency with which this activity is performed with respect to the others. In particular, the activities related to the radiology exams are more frequent than *Osservazione* and the optimization focuses on reducing their waiting times by adding resources such as radiologists. Indeed, if the number of radiologists would not change, the total time required for the tokens to perform such activities would overtake the time required by the observation activity, increasing the total waiting time of the process. Thus, in terms of absolute cost and waiting time it is more convenient to change the number of radiologists from one to two and reduce the number of beds causing higher waiting times for it, than the opposite.

This behavior is due to the strong assumption cited before: the fact that all the activities have the same weight in the computation of the total waiting time. An alternative approach in which each activity is weighted by the average of its frequency would produce balanced waiting time reductions.

Additionally, it can be set in the optimization a limit to each activity waiting time, so that it becomes prohibited accepting solutions not feasible. This approach requires the knowledge of the maximum waiting time admissible for each activity which is an information that only domain experts are able to provide. In our technique, missing this types of details, we chose to just minimize the total waiting time. Nonetheless, we will show in Subsection 5.7.3 how this method can be leveraged, assuming a possible threshold on the average waiting time for each activity.

Despite this imprecision, this scenario gives an idea of what would happen re-arranging the workforce. The new configuration is able to reach the same case durations reducing abundantly the total costs, which was the objective of the research. The problem caused by the long waiting time for activity *Osservazione* can be solved

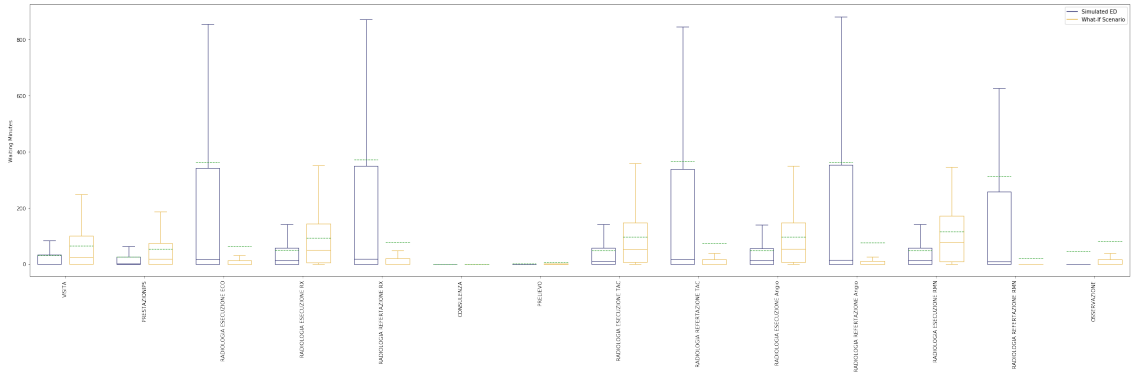


Figure 5.34: Comparison activities waiting times ED *normal resource setting* simulation vs *new resource setting* What-if Scenario with number of beds increased.

by just increasing the number of beds. Since it is a parallel activity with the radiological ones, still remaining the highest queuing activities after *Osservazione*, it would not affect their development. Adopting this correction, the resource configuration would represent a valid alternative to the actual one. An example is reported in Figure 5.34 where the only modification to the 'what-if' scenario resource configuration affects the number of beds, passing from 9 to 11. This increases the total costs of 9% with respect to the solution considered in this 'what-if' scenario, nonetheless it represents a valid alternative to the actual resource configuration since it reduces the real costs by more than 8%, with an expense per day amounting to 16560€.

Finally, it can be carried out an analysis of the degree of utilization of each resource, comparing the original and the new scenario workload (Figure 5.35). As a consequence of the reduction of the number of resources, the time in which a bed appears involved in an activity remarkably increases, passing from an average of 67% of its total available time to 90%. A similar trend characterizes the nurses, whose average time busy grows by 16%. Since both in the first and second work shifts another figure has been introduced, radiologists free up their workload by 4 percentage points. On the other hand, the number of triage nurses has not changed, thus their engagement remains the same. Concerning the remaining roles, the small modifications in the number of physicians and radiologist technicians do not generate changes in their workload. This result can also be influenced by the randomness of the simulations, in which activities may appear with different frequencies bringing slight fluctuations to the final outcomes.

5.7.2.2 Similar cost resources configuration

The idea of this second 'what-if' scenario is to suggest another plausible solution characterized by the same costs as the original one. We start from the assumption that the current allocated budget for the ED is an acceptable and sustainable one and we try to find a resource configuration that would increase the efficiency.

Given the same amount of costs, this section demonstrates how it is possible to reorganize the resources in such a way that the case durations drop with respect to the original ones.

As in the previous section, it is assessed the feasibility of the proposal delving into the stability of the new simulation, the activity waiting times, and the resource utilization.

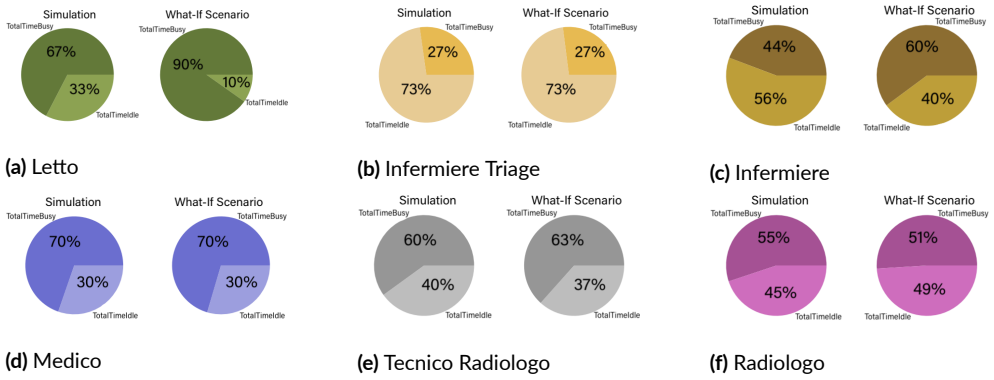


Figure 5.35: Comparison degree of utilization resources *normal resource setting* simulation vs *new resource setting* what-if scenario.

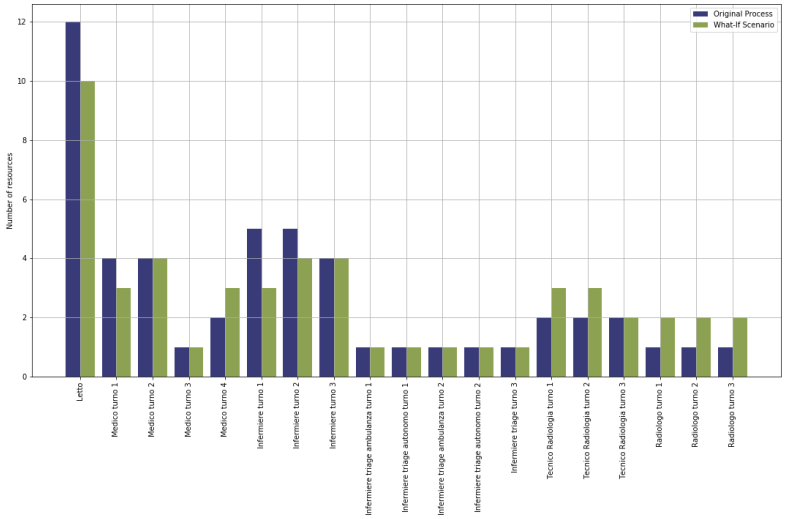


Figure 5.36: Comparison what-if scenario vs original number of resources ED process.

The new configuration of resources is obtained by considering the intersection of the non-dominated solutions with the vertical dashed line, which depicts settings with the same costs. Similarly to the previous case, there is not a perfect matching between the extracted solutions by the optimization algorithm and the straight line. Thus the nearest one with a slightly smaller cost is selected, obtaining the resource arrangement shown in Figure 5.36.

The resource setting depicted in the graph has a daily cost of 18060€, a similar value to the actual one. Nevertheless, it will be shown how the average case duration almost halves. To remove the deterministic behavior of the optimization given by the single simulation run, the usual approach of considering 20 different random seeds and carrying out the same number of simulations is exploited. From the results obtained, different statistics have been created.

Firstly, it is inspected the general stability of the simulation considering the total waiting time for each month (Figure 5.37). The graphs shows also a comparison with the actual ones, and it evidently appears the benefit

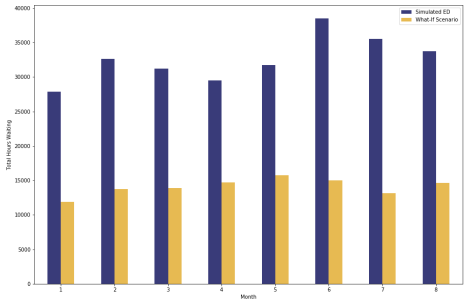


Figure 5.37: Comparison activities total waiting times per month ED *normal resource setting* simulation vs *new resource setting* what-if scenario.

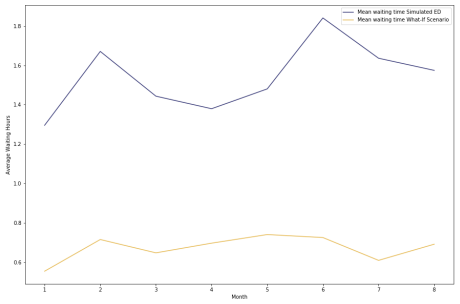


Figure 5.38: Comparison average and median activities waiting times per month ED *normal resource setting* simulation vs *new resource setting* what-if scenario.

Quantile	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Case durations	20 m	1 h 10 m	1h 34 m	2 h 14 m	3 h 6 m	4 h 30 m	6 h 40 m	11 h 9 m	1 d	8 d 13 h

Table 5.9: Case duration quantiles.

produced by the new configuration. In Figure 5.38, the yellow line depicting the average waiting time with the new resource setting, is placed in contrast with the old one. It is visible a consistent different behavior representing an average drop of 55% of the original results.

Given this significant result, it is interesting exploring what are the consequences on the case durations. As the activity duration parameters exploited for the simulation of the considered 'what-if' scenario are the same as the original simulation, the average case life-cycle is influenced just by the modification of waiting times. Figure 5.39 shows the comparison of these quantities and how they changed in this scenario, whereas Figure 5.40 has a logarithmic x-axis to offer a zoom on the pick of the densities. In both the figures, in order to keep a clean visualization, the original simulation average density has been considered for the comparison.

The density of shorter case duration is considerably higher in this new scenario than in the previous one. To have a more detailed view of the effect, Figure 5.41 shows the comparison of the different percentiles between the original simulation configuration and the new one. The high distance between the yellow and the blue line indicates an increase in the number of shorter case durations in the new scenario. Thus, it can be confirmed that thanks to the drop in waiting times, subjects are served with more efficiency and cases are solved faster. The new median case duration amounts to 55% of the original one. Nevertheless, it has to be pointed out that, even if case life cycles are shorter, the time needed to process the entire number of subjects remains ca. 240 days, as in the original case. This is due to the inter-trigger timer which regulates the arrival rate of the patients.

As in the previous 'what-if' scenario, it is possible to inspect how every single activity waiting time modifies as a result of the variations made.

The reorganization of resources has a considerable effect on the total waiting time and in the remainder of this subsection it is examined which resource modification has the greater consequences on the process behavior.

The most significant impact is caused by the near-duplication of resources involved in radiological exams. In the original configuration the average waiting time, also influenced by high picks, amounts in case of reports to 370

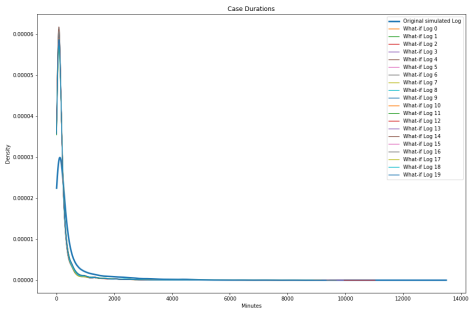


Figure 5.39: Case durations *normal resource setting* simulation vs *new resource setting* what-if scenario.

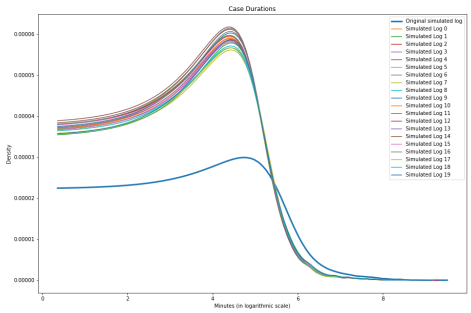


Figure 5.40: Case durations *normal resource setting* simulation vs *new resource setting* what-if scenario with x-axis in logarithmic scale.

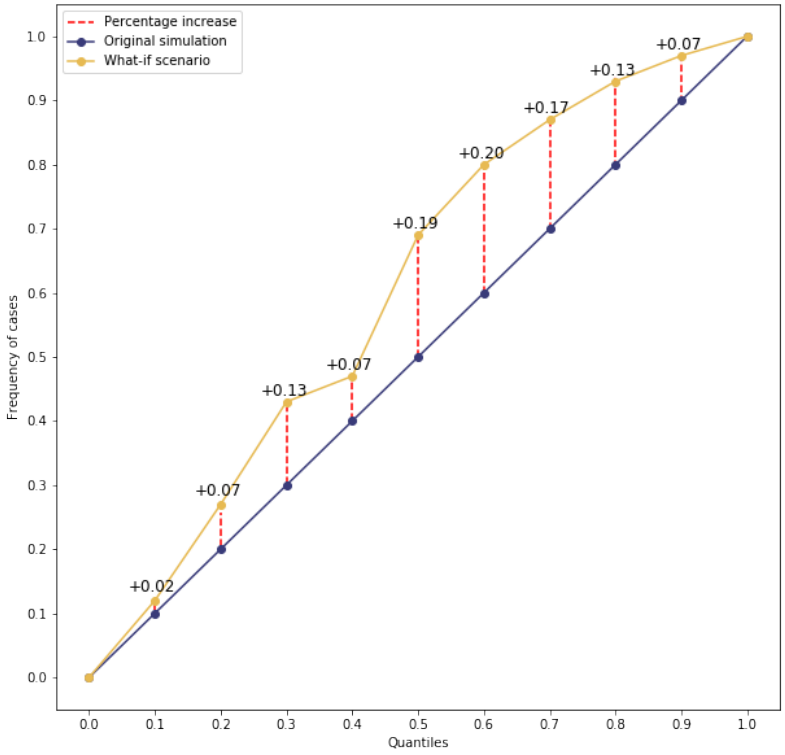


Figure 5.41: Comparison case duration quantiles between *normal resource setting* simulation vs *new resource setting* what-if scenario.

minutes (ca. 6 hours) and in case of execution to 114 minutes (ca. 2 hours). In this new setting, the availability of new resources lets decrease the average of both reporting and execution waiting times to 14 minutes. Concerning the statistics about the median, it results in a diminishment for both the activities from 16 minutes for reporting

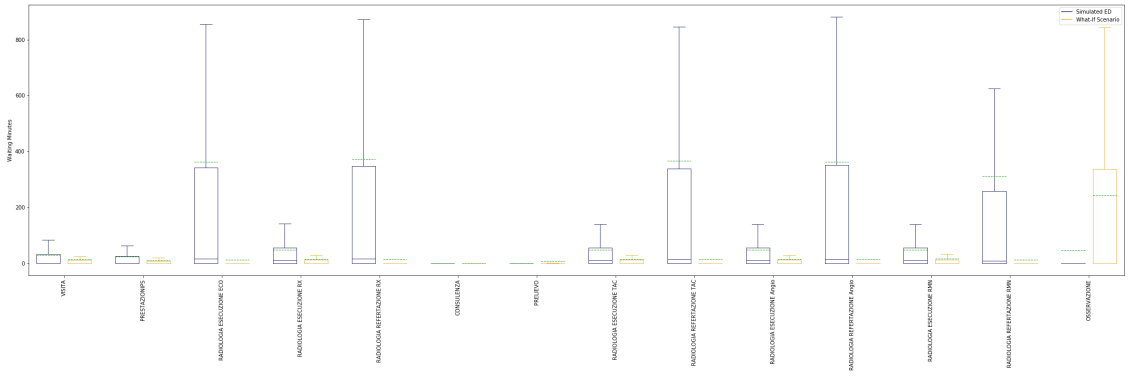


Figure 5.42: Comparison activities waiting times ED *normal resource setting* simulation vs *new resource setting* What-if Scenario.

and 12 minutes for radiological exams execution to null median waiting times in the new scenario, meaning that the majority of subjects do not queue for these activities.

On the other hand, a similar behavior found in *Osservazione* to the previous 'what-if' scenario is experienced: even if the median waiting time is zero, meaning that the majority of the patients do not have to wait before being served, the average amounts to 243 minutes. Again, this value is highly influenced by outliers, but it suggests that the approach already explained, where the activities waiting time are limited, would allow the discovery of more applicable solutions.

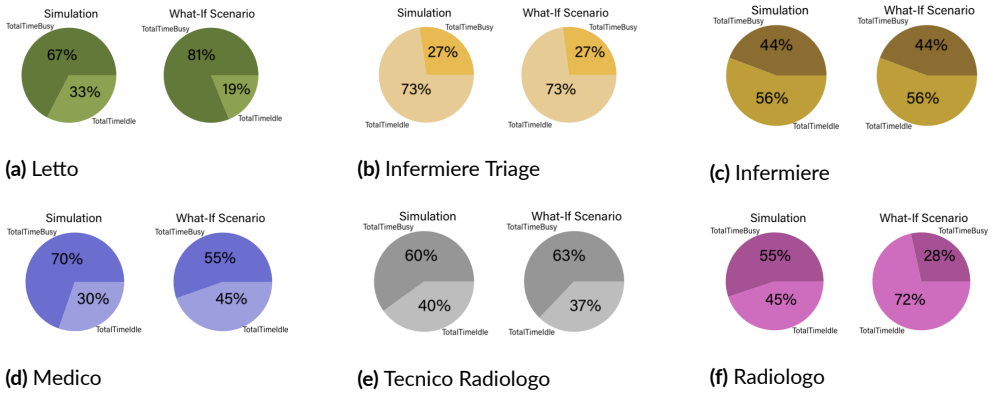
Nonetheless, as in the previous case, this study gives an idea of how, while maintaining the same costs, it is possible to find a resource configuration that can drastically impact the activities' waiting times and the case durations.

A final examination of the resources' degree of utilization in Figure 5.43 shows how the new configuration adopted would redistribute the workload. The reduction in the number of beds lets them increase their occupation by 14%. Another figure highly impacted is the radiologist: the high increase in this type of resource allows to halve the workload. In the new configuration, a physician is removed from the first work shift and added to the fourth one. This modification allows for a reduction of 15% their occupation. All the other resources remain on average with a similar amount of work as in the original configuration.

5.7.3 Costs and waiting times optimization with boundary conditions

In this sub-section, it is briefly introduced a similar optimization approach to the previous one, which includes a boundary condition on the average activities waiting time. The objective of this research is to show an additional possibility offered by the simulation tool and the optimization technique. In particular, this approach represents a method to force the optimization to find out domain-acceptable solutions.

When performing an optimization it is possible to declare some conditions that must be satisfied by the outcomes. If they are not followed, the solution is marked as unfeasible and is discarded.



(a) Letto **(b)** Infermiere Triage **(c)** Infermiere

(d) Medico **(e)** Tecnico Radiologo **(f)** Radiologo

Figure 5.43: Comparison degree of utilization resources *normal resource setting* simulation vs *new resource setting* what-if scenario.

The choice of the boundary condition can be modified depending on the type of target. In our case study, given the lack of detailed objectives, we decided to set a strict threshold of 3 hours on the average waiting time of each activity, so that all the solutions presenting higher mean waiting times are discarded. Thus, the new optimization problem is rewritten as shown in Equation 5.2, where $\overline{w}(i)$ expresses the average waiting time of each activity and 10800 seconds correspond to the threshold of 3 hours.

$$\begin{cases} \varepsilon_1 = G \sum_{m=0}^{n} h(m) * c(m) \\ \varepsilon_2 = \sum_{k=0}^{i} w(i) \\ \overline{w}(i) \leq 10800 \end{cases} . \tag{5.2}$$

The strategy of adding a boundary to the feasible space represents a possible solution to the problem arisen in the previous optimization, where the minimization of total waiting times resulted in a decrease in some activities' waiting times and a high increase in others.

To run the optimization, it has been exploited the modeFRONTIER workflow in Figure 5.26. The environment allows the definition of boundaries on each variable of the optimization problem, hence to each activity waiting time has been set a limit of acceptance of 10800 seconds. The optimization carried out by MOGAII algorithm explored 2185 resource configurations, of which 271 were unfeasible. The non-dominated solutions amounted to 33.

Figure 5.44 presents a comparison of the non-dominated solutions of the first objective optimization (Equation 5.1) (without boundary conditions) and of the second one (Equation 5.2).

The boundary conditions lets increase the level of the non-dominated solutions frontier. The shaded solutions found with the previous optimization are discarded because they show at least one activity with an average waiting time higher than three hours. The new optimization is indeed able to find only two configurations dominating the original one, which, given the considered threshold, are not able to drastically reduce either the total costs or the total waiting time.

It is interesting noting how, after a certain cost value, the solutions of the two optimizations belong to the same frontier, suggesting that when the budget increases the target is reached automatically, without needing to discard solutions with a threshold.

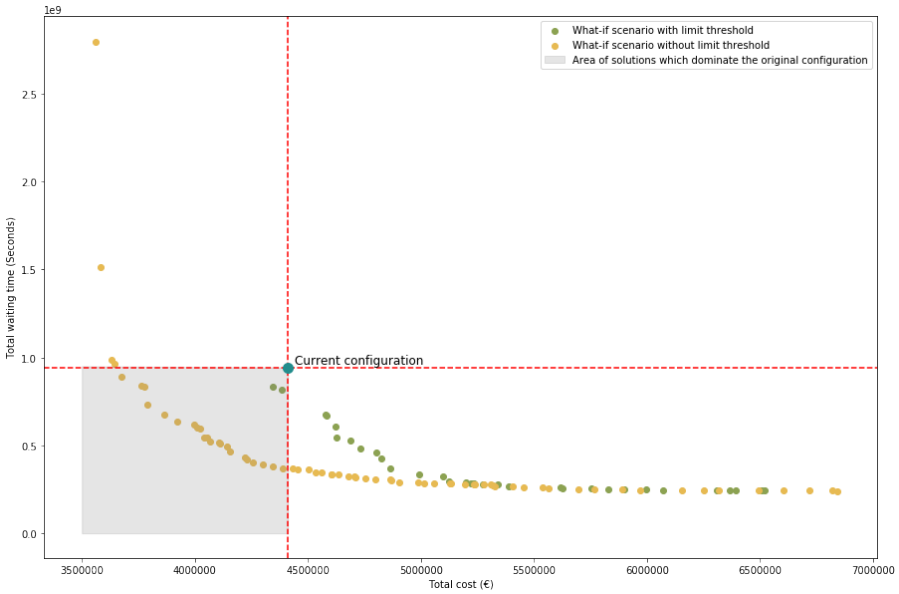


Figure 5.44: Comparison non-dominated solutions of optimization without boundary conditions and with boundary conditions on the average activities' waiting time.

In the remainder, we analyze one of the resources configuration discovered that dominates the original one to understand its benefits in terms of costs, case durations, and waiting times.

The new resource configuration considered is shown in Figure 5.45; it allows a cost reduction of 1.5%, passing from a daily cost of €18080 to €17880, and a total waiting time decrease of 9%.

The small reduction in waiting times leads to a decrease also in the case durations, as shown by the densities comparison in Figure 5.46, Figure 5.47 and by the quantiles graph in Figure 5.48.

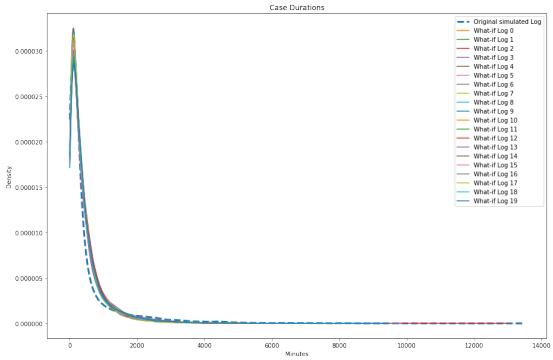


Figure 5.46: Case duration densities between *normal resource setting* simulation vs what-if scenario with threshold.

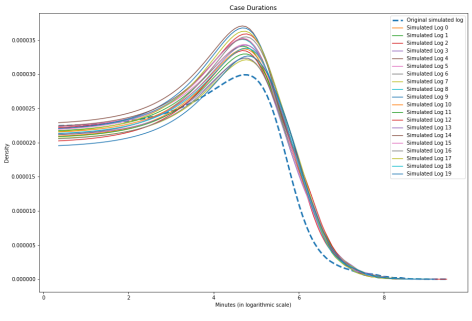


Figure 5.47: Case duration densities between *normal resource setting* vs what-if scenario with threshold with x-axis in logarithmic scale.

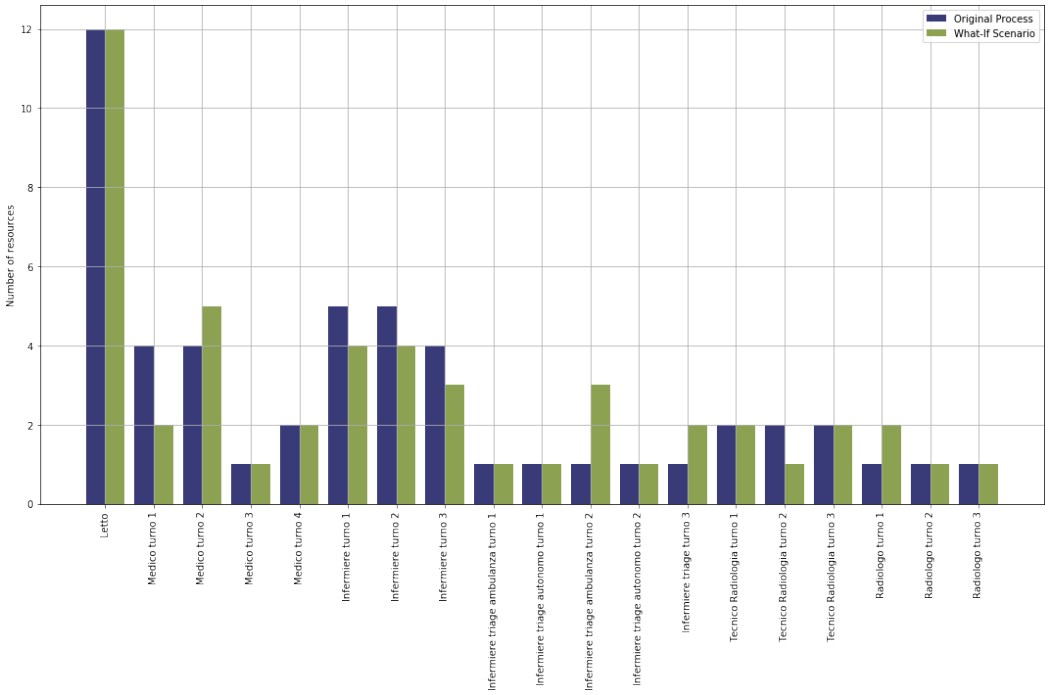


Figure 5.45: Comparison what-if scenario vs original number of resources ED process.

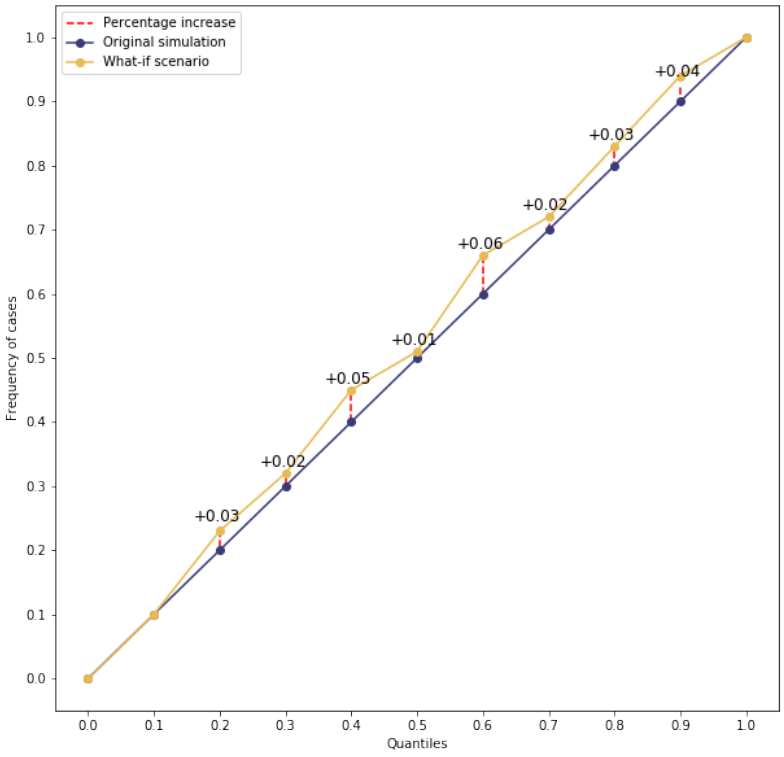


Figure 5.48: Comparison case duration quantiles original simulation vs what-if scenario with waiting threshold fast track.

Quantile	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Case durations	20 m	1 h 10 m	1h 34 m	2 h 14 m	3 h 6 m	4 h 30 m	6 h 40 m	11 h 9 m	1 d	8 d 13 h

Table 5.10: Case duration quantiles.

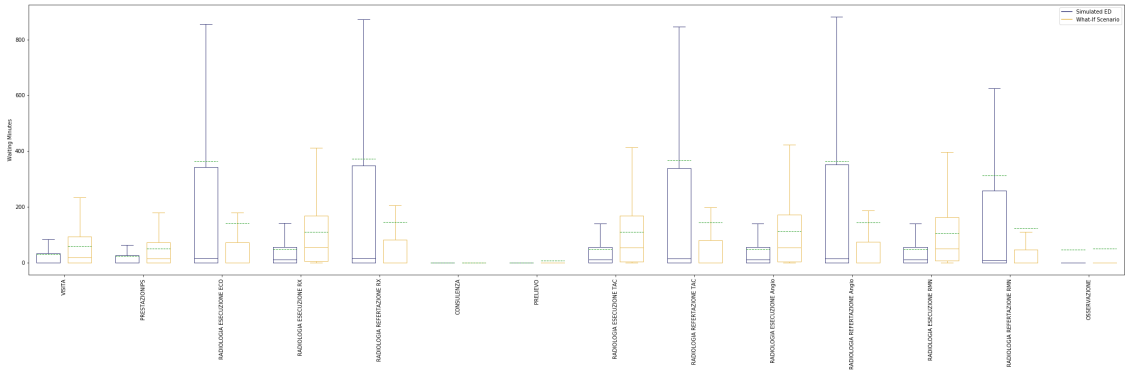


Figure 5.49: Comparison activities waiting times ED *normal resource setting* simulation vs *new resource setting* with waiting threshold what-if scenario.

The impact of the new resources configuration on each activity waiting time is presented in Figure 5.49. Each activity has an average waiting time below the 3 hours threshold, as forced by the optimization boundaries. It is interesting analyzing how the waiting times are re-distributed among each activity. The bottlenecks of the process are solved thanks to the slight increase of the radiologist physicians. On the other hand, the reduction of one radiologist technician lets increases the waiting time for all the radiological exams execution. The same consideration is valid for the reduction in the number of nurses, which causes an increase in the waiting in *Visita*, *PrestazioiPS*, and *Prelievo*.

Nonetheless, given these modifications, as already mentioned, the total waiting time decreases, as shown also in Figure 5.50 and Figure 5.51 where it is considered a comparison between the original simulation and the outcome of the optimization solution: particularly, the average waiting time per month is reduced from 1 hour and 30 minutes to 1 hour and 12 minutes.

To conclude, the new optimization method carried out a feasible resource configuration which would represent an alternative to the original one, allowing for a reduction of costs and waiting time.

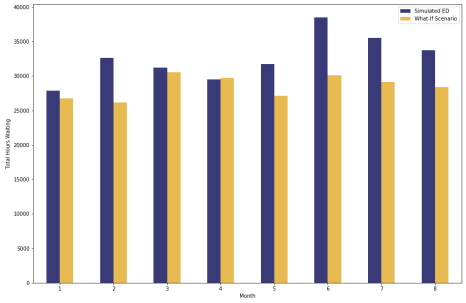


Figure 5.50: Comparison activities total waiting times per month ED *normal resource setting* simulation vs *new resource setting* with waiting threshold what-if scenario.

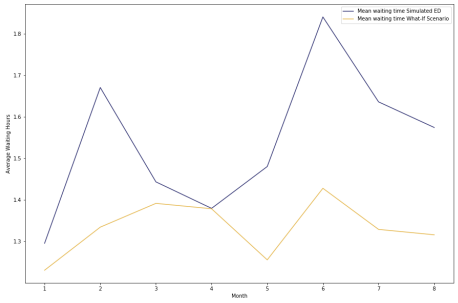


Figure 5.51: Comparison average and median activities waiting times per month ED *normal resource setting* simulation vs *new resource setting* with waiting threshold what-if scenario.

6

Conclusion And Future Works

This thesis work presents a case study on a Tuscany ED aiming to describe how process simulation can support healthcare managers in strategic decision-making. Despite the data quality issues of the provided event log and the assumptions needed to be taken in order to deal with the limitations of the simulation software, the found results show the effectiveness of process simulation in discovering plausible alternatives to address emerging healthcare issues.

The starting point for the creation of the process simulation is represented by the event log originating from the hospital information system. As typically happens, the available data misses the activity start timestamps which is an essential detail for defining the simulation parameter about the distribution of the activity durations. As a consequence, the main effort required for producing the simulation model consists in their accurate estimation. For this reason, the first section of the thesis leverages and enhances a technique that derives an estimation of the start event timestamps through the optimization of a parametric function. We contribute to enhancing this method by introducing a genetic optimization approach to increase the exploration capacity in the solution searching space and augment the convergence of the optimization. To assess the quality of the improvement, we consider two case studies with complete event logs, remove their activity start timestamps, and compare the resulting estimations found by the original local-based search method and new algorithm strategies with the original ones, proving the genetic optimization effectiveness. Given the successful result obtained, the same technique allows the enrichment of the ED case study.

The second part of this research focuses on developing an accurate simulation process, delving into all the steps to extract the necessary process information. Despite the simulation software limitations and the initial problems of the event log, the accuracy of the process developed, which is assessed both by carrying out statistics comparisons with the original data and by domain experts, allows the creation of 'what-if' scenarios aiming to explore solutions to the principal healthcare issues, such as congestion, limited budgets and the scarce number of resources. The first scenario suggests the introduction of a pediatric fast track which is able to reduce the median case durations by 35% with respect to the original ED structure. In order to face the scarce budget and the high waiting times

of the ED, the second 'what-if' scenario proposes a multi-objective optimization problem aimed to find the bests resource configurations which minimize both the total costs and the total waiting times. The outcomes reveal optimal solutions able to improve the original configuration, both in terms of costs and total waiting times. For this reason, two configurations have been studied: one configuration cuts 16% of the total costs while maintaining a similar total waiting time and case duration, whereas the second one provides a reduction of the case durations and waiting time by 55% allocating the same budget as in the original configuration.

We acknowledge the presence of limitations due both to the simulation software and the lack of healthcare managers' prefixed objectives which force us to make some possibly unrealistic assumptions for scenario development. However, it is explained during the development of the simulations how once obtained an accurate simulated process, it is possible to easily customize the research target and/or the dependencies of the optimizations to reach the desired result. Nonetheless, the outcomes still represent an important starting point for future research and should stimulate the consideration of process mining as an impacting technique to drive healthcare-managers decisions.

To conclude, we present an overview of the possible future development of this work aiming to overtake the limitations and possibly incorrect assumptions from a domain-based point of view taken in this research. We split the next steps into two parts: one related to the start timestamp estimation technique and one to the process simulation framework.

- **Start timestamp estimation improvement**
  - The introduction of a weighted error function in Equation 4.1 allows for an improvement in the final estimation of start timestamp accuracy for activities with a simulated waiting time distribution highly different, in terms of absolute time, from the waiting time distribution computed enriching the event log with the estimated start timestamps during the various runs of the optimization, but penalizes those activities with the same distributions' integral difference, but with a lower distance in terms of absolute time. In the next step, we aim to introduce an error function that allows a good estimation of start timestamps for all the activities.
  - The accuracy of the start timestamps estimations might be improved by reformulating the objective function Equation 4.1 as a multi-objective optimization problem.
  - Given the stochastic behavior of genetic algorithms, to furtherly validate the improvement brought by this technique, we aim to perform an additional study on their convergence rate in dependence on the alpha step ∂ by running a batch of simulations and by studying their outcomes statistics.

- **Process simulation improvement**
  - One of the limitations given by the simulation software is its inability to apply token priorities and sharing queues between activities. Hospital patients are associated with a color which specifies their urgency degree. This information is essential for the ED process because the most injured patients (marked as red or yellow patients) need rapid treatments and they have preferential pathways. At the same time, they require the same resources as the other patients, thus they contribute to the congestion of the ED. In the following step, we aim to realize a simulation process that considers this priority information by leveraging a simulation software without those limitations. This may require looking for alternative process simulation software solutions. This would allow the development of 'what-if' scenarios specifying different boundary waiting times for each patient type.

– Another limitation of the simulation software prevents specifying distinct arrival distributions per time slot. Typically, EDs experience lower congestion during night hours. For this reason, in a future step we aim to find a simulation software that allows configuring inter-trigger timers according to the day hours.

– The lacking of decision-makers guidelines and some limitations of the simulation software forced us to make some assumptions in developing the pediatric fast track 'what-if' scenario. This prevented us from the distinction between *normal* and pediatric patients when defining the XOR-gateways probabilities. In particular, we kept the same gateways probabilities and the same BPMN model developed for the process simulation. In future development, we will establish in advance the type of 'what-if' scenario we want to develop, thus being able to define conditional probabilities at each XOR-gateway according to the type of patient considered. This modification will allow the specification of the gateway probabilities for each specific patient type in 'what-if' scenarios.

References

[1] M. Draghi and D. Franco, "Documento di economia e finanza 2022." Ministero dell'economia e delle finanze, pp. 42–48.

[2] L. Aboueljinane, E. Sahin, and Z. Jemai, "A review on simulation models applied to emergency medical service operations," *Computers I& Industrial Engineering*, vol. 66, no. 4, pp. 734–750, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835213003100

[3] N. Martin, *Data Quality in Process Mining*. Cham: Springer International Publishing, 2021, pp. 53–79. [Online]. Available: https://doi.org/10.1007/978-3-030-53993-1_5

[4] W. van der Aalst et Al., "Process mining manifesto," pp. 169–194, 2011.

[5] Xes, extendible event stream. [Online]. Available: https://xes-standard.org/

[6] W. van der Aalst, *Process Mining Data Science in Action*. Springer, 2016.

[7] *Information technology — Object Management Group Business Process Model and Notation*, Technologies de l'information — Modèle de procédé d'affaire et notation de l'OMG Std., 2013.

[8] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer, 2012.

[9] C. Fracca, A. Bianconi, F. Meneghello, M. De Leoni, F. Asnicar, and A. Turco, "Bpsimpy: A python library for wfmc-standard process-simulation specifications," *BPM 2021 Demos and Resources track*, 05 2021.

[10] "Bpsim 2013: Workflow management coalition: Bpsim–business process simulation specification."

[11] G. van Hulzen, N. Martin, B. Depaire, and G. Souverijns, "Supporting capacity management decisions in healthcare using data-driven process simulation," *Journal of Biomedical Informatics*, vol. 129, p. 104060, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046422000764

[12] C. Fracca, M. De Leoni, F. Asnicar, and A. Turco, "Estimating activity start timestamps in the presence of waiting times via process simulation," *Springer*, 2022.

[13] A. Rozinat, R. Mans, M. Song, and W. van der Aalst, "Discovering simulation models," *Information Systems*, vol. 34, no. 3, pp. 305–327, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437908000690

[14] "Discrete modeling and simulation of business processes using event logs," *Procedia Computer Science*, vol. 29, pp. 322–331, 2014, 2014 International Conference on Computational Science.

[15] C. Poloni and V. Pediroda, *GA coupled with computationally expensive simulations: tools to improve efficiency*. Wiley, 1997, pp. 267–288. [Online]. Available: http://hdl.handle.net/11368/2545751

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] modefrontier. [Online]. Available: https://engineering.esteco.com/modefrontier/

[18] Volta. [Online]. Available: https://volta-release.esteco.com/

[19] N. Melão and M. Pidd, "Use of business process simulation: A survey of practitioners," *Journal of the Operational Research Society*, vol. 54, no. 1, pp. 2–10, 2003. [Online]. Available: https://doi.org/10.1057/palgrave.jors.2601477

[20] D. Duma and R. Aringhieri, "An ad hoc process mining approach to discover patient paths of an emergency department," *Flexible Services and Manufacturing Journal*, vol. 32, pp. 6–34, 2018.

[21] H. M. Marin-Castro and E. Tello-Leal, "Event log preprocessing for process mining: A review," *Applied Sciences*, vol. 11, no. 22, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10556

[22] J. Nakatumba, "Resource-aware business process management : analysis and support," Ph.D. dissertation, Mathematics and Computer Science, 2013.

[23] V. Denisov, D. Fahland, and W. M. P. van der Aalst, "Repairing event logs with missing events to support performance analysis of systems with shared resources," in *Application and Theory of Petri Nets and Concurrency*, R. Janicki, N. Sidorova, and T. Chatain, Eds. Cham: Springer International Publishing, 2020, pp. 239–259.

[24] A. Rogge-Solti, R. S. Mans, W. M. P. van der Aalst, and M. Weske, "Repairing event logs using timed process models," in *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, Y. T. Demey and H. Panetto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 705–708.

[25] M. Camargo, M. Dumas, and O. González-Rojas, "Learning accurate business process simulation models from event logs via automated process discovery and deep learning," in *Advanced Information Systems Engineering*, X. Franch, G. Poels, F. Gailly, and M. Snoeck, Eds. Cham: Springer International Publishing, 2022, pp. 55–71.

[26] S. J. J. Leemans and Prom, "Inductive visual miner manual," 2017.

[27] S. Leemans, D. Fahland, and W. van der Aaslt, "Discovering block-structured process models from event logs - a constructive approach," *Springer*, 2013.

[28] F. Mannhardt, M. de Leoni, and H. A. Reijers, "The multi-perspective process explorer," in *BPM*, 2015.

Acknowledgments

I would like to express my gratitude to my supervisor Dr. Massimiliano De Leoni, who gave me the opportunity to work on this challenging project and assisted me with valuable advice throughout.

A special thank also to Dr. Francesca Meneghello for her kindness and helpfulness in following and helping me in the development of the project. I would like to extend my sincere thanks to Dr. Fabio Asnicar and Dr. Alessandro Turco as representatives of ESTECO for their special care and willingness in assisting every stage of the project.

Finally, I would like to express my gratitude to Dr. Davide Aloini, Dr. Alessandro Stefanini, and Dr. Elisabetta Benevento from the University of Pisa, who provided us with the process data and gave deep insights into the study.