ASSESSING THE EFFICIENCY OF A PUBLIC EMERGENCY CARE UNIT THROUGH COMPUTER SIMULATION

Otávio Martins Vasconcelos
NTT Data Brasil
Av. das Nações Unidas, 14401, Chácara Santo Antônio, São Paulo, SP, CEP: 04730-090
otaviomartinsvasconcelos@emeal.nttdata.com

Iuri Pereira Lopes
Petrobras - Petróleo Brasileiro SA
Av. Refinaria Gabriel Passos, 690, Distrito Industrial, Betim, MG, CEP: 32669-205
iurilopes@petrobras.com.br

João Flávio de Freitas Almeida
Department of Industrial Engineering, Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, Belo Horizonte, MG, Brasil. CEP: 31270-901
joao.flavio@dep.ufmg.br

RESUMO

Unidades de Pronto Atendimento (UPAs) devem ser integradas a uma rede hierárquica de saúde para serem eficazes, no entanto, para serem eficientes, estratégias operacionais devem ser adotadas. Propomos um modelo de simulação que avalia filas de pacientes e número de funcionários para definir o nível de eficiência do atendimento clínico e pediátrico em uma UPA. Os resultados sugerem que a unidade pública de saúde é eficiente, com 80% a 97% de utilização dos recursos gargalos, mas há desbalanceamento de recursos. Cenários indicam que o nível de eficácia UPA pode ser aumentada em 24% se os investimentos futuros priorizarem a contratação de um clínico geral, uma enfermeira de triagem e uma secretária. Por outro lado, a redução de qualquer uma destas três funções tem um impacto negativo significativo na eficiência do sistema.

PALAVRAS CHAVE. Cuidados em saúde, Análise de eficiência, Simulação por eventos discretos.

ABSTRACT

While emergency care units must be integrated into a healthcare network to increase their effectiveness, to improve their efficiency, operational strategies must be adopted. We developed a simulation model and evaluated patient queues and staffing levels to assess the efficiency of clinical and pediatric care at a public health unit. Our results suggest that the public health unit is efficient, with 80% to 97% utilization of bottleneck resources, but there are resources imbalances. Scenarios indicate that the UPA effectiveness level can be increased by 24% if future investment prioritizes hiring a general practitioner, a screening nurse, and a secretary in this sequence. Conversely, the reduction of any of these three functions has a significant negative impact on system efficiency.

KEYWORDS. Health care planning, Efficiency analysis, Discrete event simulation.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

1. Introduction

Emergency care units (UPA, from Portuguese acronym) are integrated into a healthcare system, which comprises hierarchical health services operated by professionals, resources, and institutions that fund and regulate health units to improve or maintain health, which is believed to be one of the most crucial aspects of life [Uchimura et al., 2015; Österle and Rothgang, 2021]. Decisions in healthcare are hierarchical, and classified as strategic, tactical, and operational according to a taxonomy [Hulshof et al., 2012]. The process of determining the number of resources, professionals, or equipment in health units to achieve effectiveness and efficiency is a tactical decision, commonly referred to as capacity planning. The presence of queues on such systems increases the complexity of capacity planning [Siciliani and Hurst, 2005], requiring the use of sophisticated methods, such as discrete event simulation (DES) [Law et al., 2007].

This study models a public UPA in Contagem, Minas Gerais (MG), which guidelines for the unit operations in the health care network are described in the National Emergency Care Policy [Ministry of Health, 2024]. According to Fonseca and Ferreira [2009], the efficiency level of health units in the micro-regions of MG is high, and Belo Horizonte micro-region, where the case study UPA is located, is a reference point. However, the general nurses' perception is that UPAs often lack materials, standardization of procedures, human resources and training. Additionally, they reveal the incorrect way the service is used by the population, hindering the work process [Oliveira et al., 2015]. The proposed DES model reinforces the need for standardization of procedures and contribute to its capacity planning by assessing the UPA efficiency, and proposing strategies to improve its effectiveness.

DES has been successfully applied on health care systems, such as on the evaluation of admission and discharge policies on Intensive Care Units performance [Hasan et al., 2020], or on evaluating medical testing equipment for improving the quality private hospitals on the diagnostic services [Mira et al., 2023]. However, the DES models are mostly implemented in commercial softwares, and poorly described on researches. Moreover, the literature on process-based generic-purpose discrete event simulation models for healthcare is relatively scarce, with few applications, such as in Corgozinho et al. [2020] and Hassanzadeh et al. [2023]. Thus, we propose a model, and algorithms described on operational level of detail to be implemented as DES in Simpy [Scherfke, 2014; Team Simpy, 2024] for a public health unit capacity planning, contributing to increasing its productivity. SimPy is a Python discrete-event simulation library. The behaviour of patients, equipment and healthcare professionals is modelled with processes. All processes live in an environment. They interact with the environment and with each other via events.

In the following sections we describe this research. In Section 2 we introduce the conceptual model, algorithms and data used in the DES model. Section 3 presents the results, scenarios and discuss practical implementation suitability. Finally, in Section 4, we conclude the study, presenting its limitations, future developments, and research directions.

2. Method and data

The conceptual modeling of UPA operations was conceived by interviewing professionals who work in the system. The model, represented by an Activity Cycle Diagram (ACD), describes the entities' flow, activities, and queues. The process is described as follows: patients arrive at the secretariat for the registry, undergo a screening process, and are classified according to five types of prioritization (1 to 5) according to Manchester protocol [Zachariasse et al., 2017] (emergency, very urgent, urgent, less serious, and mild cases) which sets the maximum waiting time of 0, 10, 50, 120, or 240 minutes to start a consultation.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

The UPA has three general practitians (GPs) and two pediatricians. After the consultation, patients leave the system or experience one of the four types of examination (X-ray, electrocardiogram, urine, and blood test) or take medication. After medication, a patient can leave the system, return to the physician with higher priority care, or undergo tests. A patient must return to the physician with the exam before leaving the system. X-ray and electrocardiogram results are immediate, but blood and urine tests require waiting for the exams' results. During this interval, the patient can undergo further examinations.

The scope of the study is limited to medical care and examinations at the UPA. Therefore, we did not model the emergency system. Furthermore, we adopted some simplifying assumptions, such as arrival rates are constant, patients can perform all exams, each patient performs an exam only once, and after returning to the physician, patients leave the system. The process is illustrated in Figure 1 and further detailed in Algorithms 1–5.

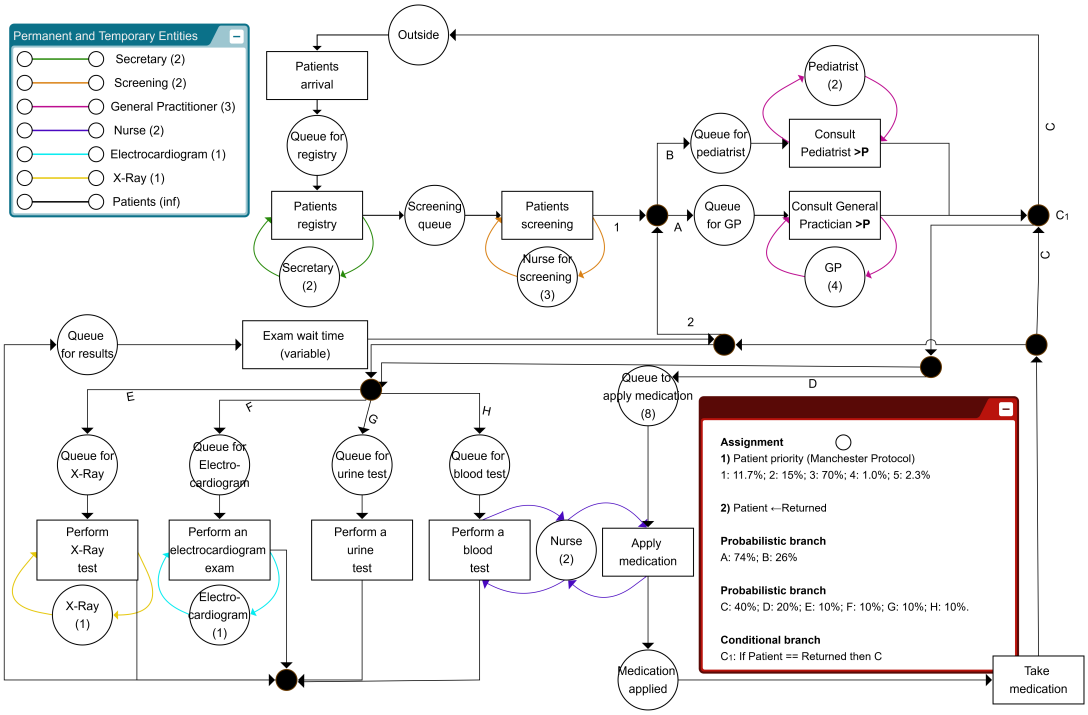


Figure 1: Activity Cycle Diagram of the UPA unit

Algorithm 1 starts the simulation. The powerful statistical tool creates a set of replications to produce robust metrics of the UPA system. Users, such as healthcare managers, or UPA decision-makers, can set the simulation time, warm-up, and the number of replications in the Star Simulation function, that runs the simulation. This function creates a simulation object, that calls a Run Replications method.

The Simulation class, defined by the Algorithm 2, runs the replications, computes output statistics with replications data, and produces charts. Each iteration of Run Replications function creates an object of the Replication class, described in Algorithm 5, which starts the process through the Create Arrivals' method. Algorithms 3 and 4 are Entity and Resources classes that create Patients objects and the resources that patient entities perform activities.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

Algorithm 1 Run Simulation

```
function Start Simulation(simulation_time, warmup_time, N)
    duration ← simulation_time                          ▷ User set simulation time
    warmup ← warmup_time                                ▷ User set warm-up time
    replications ← N                                    ▷ User sets number of replications
    Simulation ←Simulation(replications)                ▷ Creates object. Class: Algorithm 2
    Simulation.Run Replications(duration, warmup)       ▷ Algorithm: 2. Method: 11
end function
```

Algorithm 2 Simulation Class (number of replications)

```
replications ← number of replications

function Compute Statistics(.)
    Average time in system
    Average service time (resource)
    Average number in the system
    Average number in queue (resource)
    Utilization(resource)
end function

function Run Replications(duration, warm-up)
    for i in replications do
        Replication(duration, warm-up).Create Arrivals()    ▷ Alg.: 5. Method: 5
    end for
    Compute Statistics() ▷ Create statistics with replications data. Calls method: 3
    Generate Charts()                                   ▷ Generates charts for analysis
end function
```

Algorithm 3 Entity Class(Entities in the System, *time*)

```
name ← Entity += 1                                      ▷ Creates Name ID for each entity
in_queue ← 0                                            ▷ Saves time which entity enters a queue
exit_queue ← 0                                          ▷ Saves time that entity leaves a queue
in_system ← time                                        ▷ Saves time which entity enters the system
exit_system ← 0                                         ▷ Saves time that entity leaves the system
system_time ← 0                                         ▷ Saves the time the entity spent in the system
current_process ← Card registration                     ▷ Saves the Entity's process
list_resources                                          ▷ Saves resources used by entity in a given time
```

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

Algorithm 4 Resources Class()

```
ResourceData ← List if resources of Table 3, capacity and if is priority.

function CREATES RESOURCES(ResourceData)
    for j in ResourceData do
        resourceName ← resource[0]
        resourceCapacity ← resource[1]
        priority ← resource[2]
    end for
end function
```

Algorithm 5 is the Replication class. Replication objects create resources, and patients arrivals. The CREATE ARRIVALS function creates Patients objects, from the Entity class and calls PROCESS ENTITY method. This method receives a patient, and its route, allocates it in a queue, and checks its service priority before occupying the resource, described in function REQUEST RESOURCE, which allocates the required resource for performing the service, perform function WAIT OPERATION TIME, counts queuing time and number of patients, and compute Resources Usage. At the end of the activity, the resource is released in the method RELEASES RESOURCE. Following, a method COMPUTE PATIENT CYCLE save patients' and resources data, such as time and number of patients in queue, time and number of patients in process, which is used to produce statistics at the end of the simulation. Therefore, the PROCESS ENTITY method requests the required types of resources listed in Table 3, process operations which times are described in Table 2, release resources, and DECIDE NEXT PROCESS of patients, according to the conceptual model described in the ACD 1, in a discrete event schema. The method DECIDE NEXT PROCESS receives the patient, and its current process and decides the patient's flow based on method SEARCH NEXT PROCESS, that evaluates its current process, a condition or a probabilistic decision, according to ACD in Figure 1. Being a conditional or probabilistic decision, the probability distributions are evaluated in Appendix A, Table 4.

In the case of returning to GPs or pediatricians, the patient has its priority increased. If the decision is to wait for the exam results, it is necessary to process the waiting time. The PROCESS ENTITY method checks whether the following process is the system output, closing the general patient cycle and computing its statistics, or calling the following process.

2.1. Data

Data on intervals between arrivals, were obtained indirectly by evaluating indicators of four months of operation at the UPA in 2023. Records indicate 11,7% of patients have priority 1, 15% (2), 70% (3), 0,01% (4), and 3,3% (5). However, due to the lack of process digitalization, our data sample is limited, and some estimates were based on UPA's health care professionals. Exams, such as electrocardiogram, X-ray, urine, and blood have uniform distribution. Due to the absence of numerous records on registration, screening, GP and pediatrician consultation, and medication application, experts estimated data using a triangular distribution, as described in Appendix A, Table 2. The number of the resources, and the processes in which they are allocated, is described in Appendix A, Table 3. The historical flow of patients sets processes probabilistic branches, presented in Appendix A, Table 4.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

Algorithm 5 Replication Class(replication time, warm-up time)

```
1: resources ← Resources.CREATE RESOURCES()        ▷ Algorithm: 4. Calls method: 3
2: duration ← replication time                                    ▷ Algorithm: 1
3: warmup ← warm-up time                                          ▷ Algorithm: 1
4:
5: function CREATE ARRIVALS(.)
6:     time ← 0
7:     while time < duration do
8:         Patient ← Entity(entities)          ▷ Create Patient object. Class: Algorithm 3
9:         Registers Patient in the System                ▷ Records arrival for statistics
10:        PROCESSES ENTITY(Patient, "Card registration")                ▷ Calls method: 53
11:    end while
12: end function
13:
14: function REQUEST RESOURCE(process)
15:    ResourcesRequested ← List of resources of Table 3
16:    return ResourcesRequested[process] ▷ Returns requested resource of each process
17: end function
18:
19: function WAIT OPERATION TIME(process)
20:    ProcessTimes ← Probability distributions of Table 2
21:    return ProcessTimes[process]          ▷ Returns distribution time of each process
22: end function
23:
24: function RELEASE RESOURCES(process)
25:    ResourcesReleased ← List of resources of Table 3
26:    return ResourcesReleased[process]     ▷ Returns released resource of each process
27: end function
28:
29: function COMPUTE PATIENT CYCLE(Patient, process)
30:    ProcessData ←                          ▷ Data saved in each queue-process cycle
31:    (Patient: name,
32:    process: process,
33:    in_queue: in_queue,
34:    exit_queue: exit_queue,
35:    queue_time: exit_queue - in_queue,
36:    enter_process: exit_queue,
37:    exit_process: exit_process,
38:    processing_time: exit_process - enter_process)
39: end function
40:
41: function SEARCH NEXT PROCESS(process)
42:    NextProcess ← A Process/Decision according to ACD in Figure 1 and Table 4.
43:    return NextProcess[process]                    ▷ Returns next process
44: end function
45:
46: function DECIDE NEXT PROCESS(Patient, process)
47:     NextProcess(Patient) ← SEARCH NEXT PROCESS(process)      ▷ Calls method: 41
48:    if NextProcess(Patient) == Clinical then  ▷ If patient must return to the clinic
49:        PatientPriority + +                          ▷ Increase patient priority
50:    end if
51:    return NextProcess(Patient)
52: end function
```

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

```
53: function PROCESSES ENTITY(Patient, process)
54:     Enter_Queue(Patient)                    ▷ Put Patient in queue and save entry time
55:     if Patient Has Priority then            ▷ Checks if Patient has priority
56:         REQUEST RESOURCE(process)           ▷ Calls method: 14 for resource with priority
57:     else
58:         REQUEST RESOURCE(process)           ▷ Calls method: 14
59:     end if
60:     Exit queue and compute time (Patient)   ▷ Remove Patient from queue
61:     WAIT OPERATION TIME(process)            ▷ Calls method: 19
62:     RELEASES RESOURCE(Patient)              ▷ Calls method: 24
63:     Computes Resources Usage(resources)
64:     COMPUTE PATIENT CYCLE(Patient, process) ▷ Calls method: 29
65:     NextProcess ← DECIDE NEXT PROCESS(Patient, process)  ▷ Calls method: 46
66:     if NextProcess == outside then
67:         COMPUTE PATIENT CYCLE(Patient, process)  ▷ Calls method: 29
68:     else
69:         PROCESS ENTITY(Patient, NextProcess)     ▷ Calls method: 53
70:     end if
71: end function
```

The model verification consisted of checking computational results of cycles and flows of each patient. Although several combinations of flows are possible, all flows occurred according to the rule in the UPA and in Figure 1. In the validation phase, we compared the UPA's sample of records patients queue times with such metrics on simulation. The values in Table 1 were presented to the UPA team, who validated the results. Emergency patients, which is color red in Manchester protocol, are directly taken to UPA beds, which is out of scope of our conceptual model, while mild case patients, which color is blue in Manchester protocol, are less numerous, some patients give up staying in queue, leaving the system after long waiting times, and they were not tracked on this health unit. On running the simulation model, we visually set the warm-up period for five days for average queue time of very urgent patient (priority 1), as illustrated in Figure 2.

We model the UPA system in which replications are independently identically distributed (*iid*). Therefore, the subsequent study stage is sizing replications to increase the robustness of the metric (average queue of urgent patients). After five replications of the UPA's clinical and pediatric care, we observe an average of 20,35 ± 7,5 IC 95% in the number of urgent patients in the queue.

For reaching a desired precision $P(\bar{x}-h \leq \mu \leq \bar{x}+h) = 1-\alpha$ with CI 95%, the simulation is sized to n^* replications, estimated in $n^* = \lceil n(h/h^*)^2 \rceil$, where $h = t_{n-1,1-\alpha/2}s/\sqrt{n}$, $\bar{x}$ is the metric's sample average, h the precision, i.e., half-width of confidence interval, μ the population mean, the number of replications n, the sample standard deviation s, and $t_{n-1,1-\alpha/2}$ as the t-student distribution. The average queue time of urgent patients is 20,35 minutes. We set a desired precision (h^*) of 15% of the mean, i.e. 3 min for urgent patients, therefore $P(17 \leq \mu \leq 23) = 0,95$, consequently, the validated simulation experiment was sized to $n^* = 55$ replications.

3. Results and discussion

After analysing the current key performance indicators and the results of the simulations, we proposed the scenarios that would improve the effectiveness and efficiency of

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br



Figure 2: Warm-up period for resources' use

Patient queue time	Target (max.)	Real average ($\bar{x}$)	Simulation $\bar{x}$, CI(95%)
Very urgent (Orange)	10 min	22 min	21,87 [19,74; 23,99] min
Urgent (Yellow)	50 min	24 min	22,12 [19,96; 24,28] min
Less serious (Green)	120 min	68,5 min	56,01 [49,96; 62,06] min
Mild cases (Blue)	240 min	Not available	112,75 [105,22, 120,26] min

Table 1: Efficiency target of protocol, real, and simulation metrics of patients queue time.

the UPA with the least increase in resources, in line with the Ministry of Health's policy. The values presented in Table 1 validate the results, and are supplemented to better describe the UPA's current operations. The simulation reveals a highly efficient public health unit. However, the target of maximum waiting time of very urgent patients are not reached, and the most used resources, i.e., GP, nurse, and secretary represents a bottleneck for the overall UPA system. Therefore, future investments in increasing resource capacity should prioritize such resources. Simulating the UPA system considering the removal of one GP, a screening nurse, or a secretary would yield an unstable system with average arrival rates higher than server rates. In practice, such a strategy could be tragic.

In an iterative process of simulation, analysis of results and hypothesis generation, we describe four additional scenarios. The current situation is described as "As Is", and additional scenarios are described as "To Be". On **Scenario To Be 1** the number of secretaries is reduced from two to one, and registry time is reduced 50% due to investments in process redesign. In this scenario, the average queue time is reduced 1,47 minutes, from 8,34 to 6,87. **Scenario To Be 2** equipment capacity is doubled, presenting 2 X-rays, 2 MRIs, 4 electrocardiograms, and 16 nurses' assistance. Average resource utilization is reduced, however, average high-priority patient time is slightly increased. **Scenario To Be 3** increases capacity from two secretary and nurse screenings to three professionals. Results show a considerable reduction on results average utilization, however, GP utilization is not reduced. Finally, **Scenario To Be 4** considers an additional investment on another GP. Resources average utilization and queue time are presented in Figures 3 and 4.

Discrete event simulation models are often implemented by commercial software. Despite the advantages of a graphical interface and robust analysis, the costs of such applica-

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

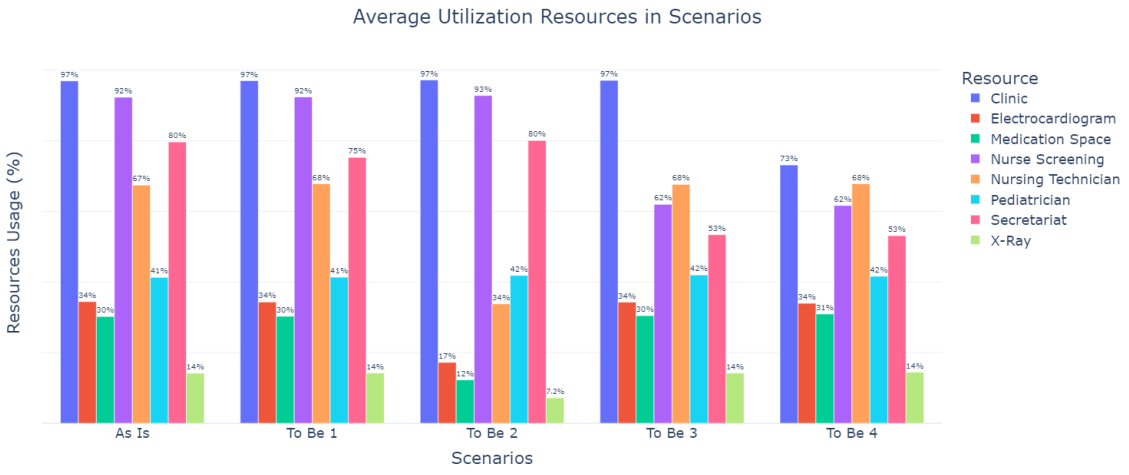


Figure 3: Average resources utilization in scenarios.

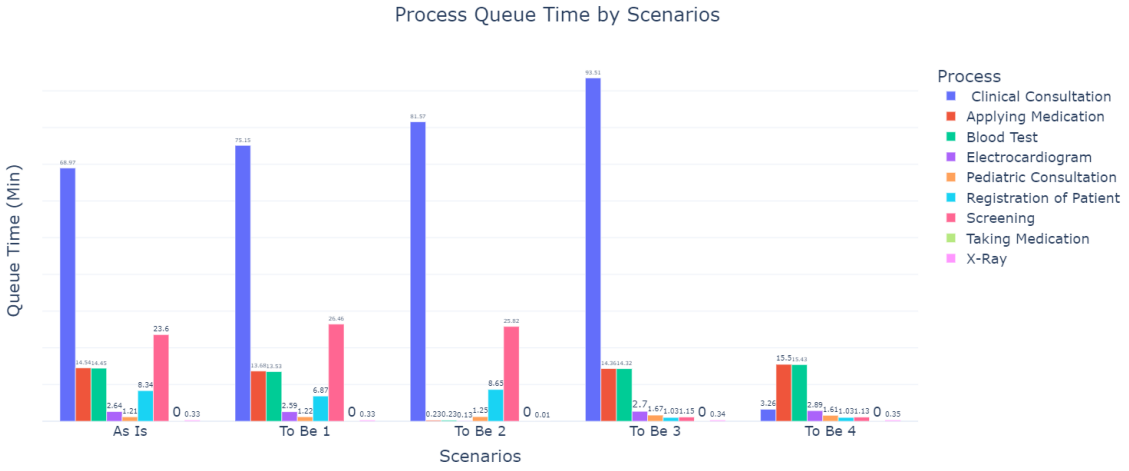


Figure 4: Average queue time in scenarios.

tions can be an obstacle for many organizations. Furthermore, the literature on open-source simulation models for healthcare is scarce. Thus, developing simulators in generic language for operational planning of the public health unit is strategic, fills a gap in the operational research literature for health, does not add costs to the system, and contributes to increasing productivity.

4. Conclusion

In this paper, we proposed a discrete-event simulation model to assess the clinical and pediatric services efficiency of a public UPA. The model enabled the analysis of the current practice and different configurations of patient and resource flows in the health unit, grouped into optimistic and pessimistic scenarios. The results suggest that the public UPA is efficient. Even though, the overall system utilization can be further improved in 24% if future investment prioritizes, hiring a GP, a screening nurse, and a secretary.

We provided a detailed model of the UPA health unit, however, the study has some limitations. Data quality can be improved by digitizing data collections in operations and

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

queues, which would imply changing distributions, the required numbers of replications, and improving the quality of outcome statistics. Another issue is the scope of the study. It was limited to clinical and pediatric care activities and did not include the flow of emergency patients. Consequently, this is an avenue for future work

Alternative strategies, such as increasing working hours, or referring to other units and granting vouchers for patients who reach a waiting time limit, were adopted in Spain, and in Portugal, respectivelly, for mitgating excessive queues of elective surgery [Siciliani and Hurst, 2005]. Although these are temporary strategies, these scenarios can also be simulated in future studies.

Bibliography

Corgozinho, M. C. B., Ferreira, V. A. S., and de Freitas Almeida, J. F. (2020). Scaling hospital laboratory queues via discrete event simulation using simpy. In *Operations Management for Social Good: 2018 POMS International Conference in Rio*, p. 119–128. Springer.

Fonseca, P. C. and Ferreira, M. A. M. (2009). Investigação dos níveis de eficiência na utilização de recursos no setor de saúde: uma análise das microrregiões de Minas Gerais. *Saúde e Sociedade*, 18(2):199–213.

Hasan, I., Bahalkeh, E., and Yih, Y. (2020). Evaluating intensive care unit admission and discharge policies using a discrete event simulation model. *Simulation*, 96(6):501–518.

Hassanzadeh, H., Boyle, J., Khanna, S., Biki, B., Syed, F., Sweeney, L., and Borkwood, E. (2023). A discrete event simulation for improving operating theatre efficiency. *The International Journal of Health Planning and Management*, 38(2):360–379.

Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., and Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS. *Health systems*, 1(2):129–175.

Law, A. M., Kelton, W. D., and Kelton, W. D. (2007). *Simulation modeling and analysis*, volume 3. Mcgraw-hill New York.

Ministry of Health (2024). Portaria 1,601/2011 of the ministry of health. https://bvsms.saude.gov.br/bvs/saudelegis/gm/2011/prt1601_07_07_2011_rep.html. Online; accessed April 2024.

Mira, J., Ferreira, D. C., Caldas, P., and Varela, M. (2023). Medical testing services' quality enhancement with discrete-event simulation.

Oliveira, S. N. d., Ramos, B. J., Piazza, M., Prado, M. L. d., Reibnitz, K. S., and Souza, A. C. (2015). Emergency care units (UPA) 24h: the nurses' perception. *Texto & Contexto-Enfermagem*, 24:238–244.

Scherfke, S. (2014). Discrete-event simulation with simpy. *OFFIS–Institute for Information Technologie, USA*.

Siciliani, L. and Hurst, J. (2005). Tackling excessive waiting times for elective surgery: a comparative analysis of policies in 12 OECD countries. *Health policy*, 72(2):201–215.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

Team Simpy (2024). Simpy - discrete event simulation for python. https://simpy.readthedocs.io. Online; accessed May 2024.

Uchimura, L. Y. T., Viana, A. L. d., Silva, H. P. d., and Ibañez, N. (2015). Unidades de Pronto Atendimento (UPAs): características da gestão às redes de atenção no Paraná. *Saúde em Debate*, 39:972–983.

Zachariasse, J. M., Seiger, N., Rood, P. P., Alves, C. F., Freitas, P., Smit, F. J., Roukema, G. R., and Moll, H. A. (2017). Validity of the manchester triage system in emergency care: A prospective observational study. *PloS one*, 12(2):e0170811.

Österle, A. and Rothgang, H. (2021). Health Care Systems. In *The Oxford Handbook of the Welfare State*. Oxford University Press. ISBN 9780198828389. URL https://doi.org/10.1093/oxfordhb/9780198828389.013.29.

A. Appendix

Case study data on process time distributions, number of resources and corresponding process, and processes probabilistic branches.

Process	Distribution	Time (minutes)
Card registration	Triangular	4, 14, 8
Screening	Triangular	6, 14, 11
Clinical consultations	Triangular	10, 20, 15
Pediatric consultations	Triangular	8, 20, 15
Apply medication	Triangular	10, 60, 40
Take medication	Triangular	5, 40, 15
Blood test	Uniform	2.5, 3.5
Urine examination	Uniform	1.5, 2.5
X-ray	Uniform	4.5, 5.5
Electrocardiogram	Uniform	11, 13
Blood test with internal analysis	Uniform	115, 125
Blood test with external analysis	Uniform	220, 260
Urine examination analysis	Uniform	55, 65

Table 2: Process times and distributions

Resource	Quantity	Process
Secretary	2	Card registration
Screening Nurse	2	Screening
GP	3	Clinic consultation
Pediatrician	2	Pediatric consultation
X-ray	1	X-ray examination
Electrocardiogram	1	Electrocardiogram examination
Nurse assistant	2	Apply medication
Medication area	8	Take medication and Apply medication

Table 3: Number of resources and corresponding process. Some activities don't have resource.

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br

Process	Following Process	Probability
Screening	Clinical	74%
	Pediatrician	26%
Clinic consultation	Exit system	40%
	Apply medication	20%
	X-ray	10%
	Electrocardiogram	10%
	Urine	10%
	Blood test	10%
Pediatric consultation	Exit system	40%
	Apply medication	20%
	X-ray	10%
	Electrocardiogram	10%
	Urine	10%
	Blood test	10%
Medication	Exit system	40%
	Return to physician	20%
	X-ray	10%
	Electrocardiogram	10%
	Urine	10%
	Blood test	10%
Urine test	Return to physician	70%
	X-ray	10%
	Electrocardiogram	10%
	Blood test	10%
Blood test	Return to physician	70%
	X-ray	10%
	Electrocardiogram	10%
	Urine examination	10%
X-ray examination	Return to physician	70%
	Blood test	10%
	Electrocardiogram	10%
	Urine examination	10%
Electrocardiogram examination	Return to physician	70%
	Blood test	10%
	X-ray	10%
	Urine examination	10%

Table 4: Processes probabilistic branches

DOI: 10.59254/sbpo-2024-193930
https://proceedings.science/p/193930?lang=pt-br