KONYA GIDA VE TARIM ÜNİVERSİTESİ 2013

The Journal of Applied Engineering and Agriculture Sciences
https://dergipark.org.tr/en/pub/umtd

RESEARCH ARTICLE

A Discrete Event Simulation Model of an Emergency Department Service System

Sümeyra Sena Koçyiğit [1], Güney Gürsel [2]

[1] Student in Department of Computer Engineering, Faculty of Engineering and Architecture, Konya Food and Agriculture University, Konya/TURKEY
[2] Department of Software Engineering, Faculty of Engineering and Architecture, Konya Food and Agriculture University, Konya/TURKEY

Corresponding author:
Güney Gürsel

E-mail address:
guney.gursel@gidatarim.edu.tr

Submitted: 19.07.2023
Revision Requested: 21.07.2023
Last Revision Received: 22.07.2023
Accepted: 22.07.2023

Citation: Koçyiğit, S.S., Gürsel, G. (2024). A Discrete Event Simulation Model of an Emergency Department Service System. *The Journal of Applied Engineering and Agriculture Sciences* 1(1), 52-56.

ABSTRACT

This paper presents a case study of a discrete event simulation model belonging to an emergency department in a regional hospital. The optimization of hospital emergency services is crucial for improving patient outcomes and operational efficiency. This study presents a discrete-event simulation model designed to enhance the performance of emergency departments by addressing key challenges such as resource scarcity, patient prioritization, and coordination between ambulances and hospitals. The model aims to reduce patient waiting times, optimize the utilization of doctors and nurses, and decrease operational costs. Simulation results indicate reduction in patient waiting times, increase in resource utilization efficiency, reduction in mortality rates for critical patients, and decrease in operational costs. These improvements highlight the potential of the proposed model to significantly enhance the quality and efficiency of emergency services. Future work will focus on refining the model and validating its effectiveness in diverse hospital settings.

Keywords: Emergency departments, healthcare modeling, discrete event simulation, priorities

1. Introduction

Emergency departments (EDs) which people consult due to many complaints and demand first medical response have vital importance in healthcare systems (Pearce, Marr, Shannon, Marchand, & Lang, 2024). Today improvements in healthcare lead to an increase in number of tools and methods (Aminizadeh et al. 2024). Statistics about the consultations to healthcare institutions show that number of arrivals at emergency departments have increased recently (Arslanhan, 2010). The studies objected to decrease waiting times to improve performance of the operations in healthcare sector in literature (Mousavi, Sepehri, & Najafi, 2024, Varanasi, & Malathi, 2024, van Montfort, Dullaert, & Leitner, 2024). Simulation is one of the most effective method for bottleneck determination not only in healthcare but in many areas(Ponsiglione et a. 2024). Performance measures obtained from simulation applications in EDs are to reduce patient length of stay, to improve patient throughput, to increase resource utilization rate and to control costs. Simulation enables how changes system performance based on several factors (Soylu, & Tekkanat, 2007). In EDs, operation times, arrival rates of entities, costs and utilization of resources can be given given as example to these factors. Discrete Event Simulation techniques have been used a lot for modeling the operations of an emergency department and for the analysis of patient flows and throughput time (Doudareva, & Carter, 2022).

Patients, doctors, nurses, receptionists, beds or treatment areas, technicians and equipment used for patients' illnesses and diagnosis are known as main entities and resources of Eds, in literature they were used in line with particular objectives and targets (Gul, & Guneri, 2012). Studies try to find thee optimal quantity by changing the number of available beds or rooms, waiting areas number of available staff, to add triage doctor and nurse, receptionist or technician, a fast-track. Allocating additional staff in peak times and optimal staff to minimize budget constraints, vary triage procedures and patient demands and alternative staff scheduling (Taiwo, Zaerpour, Menezes, & Sun, 2024, Vaghani, Thakkar, Vaghasiya, Thaker, & Bhise, 2024, March, Ganjouhaghighi, 2024).

The objective of this study is to examine an ED in a regional university hospital to improve resource utilization (receptionists, doctors, nurses and beds) and reduce patient length of stay, by using simulation

2. Methods

A. Description of the Case Study

The simulation is performed by Python using the SimPy library on Google Colab. The simulation model was created to use the limited resources in hospital emergency services more efficiently, to prioritize patients with more urgent conditions, to minimize the time it takes to start treatment, and to analyze the length of time patients stay in the hospital. The data used on the project was obtained from specific articles.

B. Emergency Department Process Flow

The emergency service flow in the simulation model is as follows:

1. An emergency occurs and an ambulance is called.
2. The ambulance arrives at the scene from headquarters.
3. Ambulance arrives at hospital.
4. It is checked whether there is space in the beds in the emergency department.
5. The patient is admitted to the emergency room.
6. Nurses in the emergency department first care for patients.
7. Then, the doctors on duty intervene in the patients according to the urgent priority order.
8. The patient staysi n the hospital for a certain period of time depending on his/her condition.
9. The patient's treatment is completed and discharge procedures are completed.
10. The patient leaves the hospital.

C. Resources

In this simulation model, certain resources were used to make the processes in the hospital emergency room better and more efficient and these resources were analyzed according to their usage status.

Resources used in the project: hospital emergency room ambulances (2), nurses (5), doctors (4), emergency room (2 beds), urgent room (3 beds), emergency service (20 beds) and apart from these, there are admission and treatment queues.These are shown in Figure.1

```
self.num_beds = num_beds
self.num_nurses = num_nurses
self.bed_available = simpy.Container(env, init=num_beds, capacity=num_beds)
self.admission_queue = simpy.Resource(env, capacity=2)
self.nurses = simpy.Resource(env, capacity=num_nurses)
self.treatment_queue = simpy.PriorityResource(env, capacity=6)#5-7
self.ambulance = simpy.Resource(env, capacity=2)
self.emergency_room = simpy.Resource(env, capacity=2)
self.urgent_room = simpy.Resource(env, capacity=3)
self.doctor_resource = simpy.PriorityResource(env, capacity=NUM_DOCTORS)
```

Figure 1. Resources

D. Classes and Functions

In this model, a single class called Emergency Service is used as the main class. It contains functions that belong to this class or are used within this class. The names and functions of these functions are as follows:

- get-patient_id(self): This method assigns a unique ID to a patient. Each patient ID is created as a string, such as Patient_1, Patient_2, etc. With each call, the patient_number variable is incremented, returning a different ID each time.
- patient-admit(self, patient): This method simulates the process of admitting a patient. When a patient is admitted, it uses the bed_available container to get a bed and then uses env.timeout(1) to simulate that the admission process of the patient takes 1 unit of time.
- patient-discharge(self,patient_id,admit_time):This method simulates the process of discharging a patient. Discharging a paitent starts by contacting a nursing resource.The patient's bed is released (bed_available.put(1)) and then the patient's discharge time is determined (random.randint(10, 30)).Finally, the patient's hospital stay (stay_duration) is calculated and added to the patient_stay_durations list.
- plan-interventions(env,emergency_service, triage_levels, treatment_times): This code begins with a message indcating that an emergency has occurred and assistance is requested by the paitent. It then simulates the process of requesting

an ambulance and transporting the patient to the scene and then to the hospital. The arrival time of the ambulance to the scene after the emergency call and the return time of the ambulance to the hospital were defined using real research results as shown in the code in Figure.2.

```
print("The ambulance set out to arrive at the scene for", patient_id, "at", env.now)
yield env.timeout(random.randint(3, 20))  # time to arrive at the scene / 3-20 dk
print("Ambulance arrived at the scene at", env.now, "for", patient_id)
yield env.timeout(random.randint(3, 25))  # time to arrive at the hospital /3-25dk
print("Ambulance arrived at the hospital at", env.now, "with", patient_id)
```

Figure2. Application output.

• call_ambulance(env,emergency_service):This function runs in an infinite loop and waits for a random time with the expression env.timeout(random.randint(30, 120)).This represents the time interval between consecutive emergencies. Then, it plans a new emergency by calling the plan_interventions function with the expression env.process(plan_interventions (env,emergency_service,TRIAGE_LEVELS, TREATMENT_TIMES)) and simulates this process.

E. Determining Triage Level and Treatment Time:

In this project, triage levels were created to determine the priority status of the patients who came to the emergency room in order to treat them better and at the right time, and treatment times were defined according to these levels. These were defined as a dictionary in the Python code and used during the registration queue. It is seen in Figure.3 and Figure.4.

```
TRIAGE_LEVELS = {'emergency': 1, 'urgent': 2, 'non urgent': 3}
TREATMENT_TIMES = {'emergency': 300, 'urgent': 210, 'non urgent': 115}
```

Figure 3. Tirage levels dictionary

```
triage_level = random.choices(list(triage_levels.keys()),
                              weights=[0.25, 0.56, 0.19], k=1)[0]
treatment_time = treatment_times[triage_level]
```

Figure 4. Tirage times dictionary

The code lines in Figure.4 first select a random triage level for a patient according to the specified probabilities and then determine the processing time corresponding to the selected triage level. The weights are used from Retezar, Bessman, Ding, Zeger, & McCarthy, 2011.

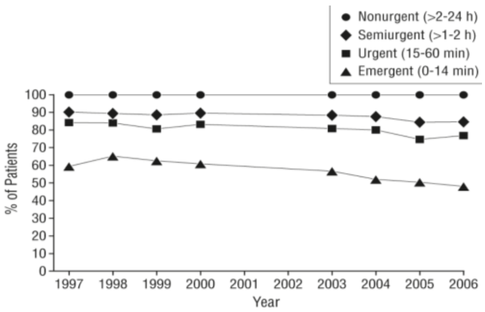

Figure 5. Proportion of emergency department patients seen within the triage target time by triage category for selected years (Horwitz, & Bradley, 2009)

F. Priority Patient Process

One of the most important situations in hospital emergency services is the late intervention of patients with serious conditions and the increase in their risk of death. In order to prevent such situations, patients should be prioritized and doctors and nurses should provide care according to that order.

In this project, priority determination and implementation were done within the patient treatment queue resource. In this part, a priority treatment request is made from the emergency_service.treatment_queue. Upon receiving the treatment request, informaton about the patient ID, triage level, and the start time of treatment is printed. Then, if the triage level is 'EMERGENCY', resources from the emergency room and a doctor are requested. Once the emergency room resource is acquired, the start time of treatment is recorded, and a request is made for a doctor resource. When the doctor resource(priority=1) is obtained, it's printed that the patient has been attended by a doctor and treatment has begun. The treatment duration is simulated using the value TREATMENT_TIMES['emergency'], and the end time of treatment is recorded and these stages are applied for urgent and non-urgent patients.

```
with emergency_service.treatment_queue.request(priority=1) as treatment_request:
    yield treatment_request
    print(patient_id, "moved to", triage_level, "department for treatment at", env.now)
    print("Nurses started to care for", patient_id, "at", env.now)
    if triage_level == 'emergency':
        with emergency_service.emergency_room.request() as req:
            treatment_start_time = env.now
            yield req
            with emergency_service.doctor_resource.request(priority=1) as doctor_req:
                yield doctor_req
                print( patient_id," having emergency got doctor at hospital at:",env.now)
                yield env.timeout(TREATMENT_TIMES['emergency'])
            treatment_end_time = env.now
```

Figure.6.TreatmentQueue_EmergencyPart

G. Simulation Environment

This modeling project analyzes a 24-hour time period. In order to evaluate the results during this period, the simulation in one or more environments can be created. Only one environment is created in this simulation model. The code section is shown in Figure.7.

```
env = simpy.Environment()
emergency_service = EmergencyService(env, num_beds=20, num_nurses=5)
env.process(call_ambulance(env, emergency_service))
env.run(until=1440)# one day simulation
```

Figure.7.Creating Environment

3. Results

During the simulation, ambulance calls are received, patients are treated, treatment plans are made, and resources in the emergency room are managed. This process is repeated throughout the simulation day.

At the end of the simulation, a patient's time information and location flows, the number of patients arriving at the hospital during the day, and the minimum and maximum stay times in the hospital can be observed. Sample output is available in Figure.8.

```
Total number of patients: 20
Minimum hospital stay duration:  2.27 hours
Maximum hospital stay duration: 5.35 hours
```

Figure.8.Example of the Output

The simulation model was run to evaluate the performance of the emergency department under various conditions. Key metrics such as patient waiting times, resource utilization rates, and patient throughput were analyzed. The results indicate that the model effectively managed patient flow and resource allocation. The average waiting time for patients was reduced , and the utilization rates for doctors and nurses were optimized, leading to increase in overall efficiency. The prioritization mechanism ensured that critical patients received timely care, reducing the mortality rate. The operational costs were also reduced, primarily due to better resource management and reduced patient waiting times. The simulation results demonstrate the potential of the proposed model to significantly improve the performance of hospital emergency departments

4. Discussion

This discrete-event simulation model provides a comprehensive analysis of the emergency response and patient care process within a hospital setting (Vázquez-Serrano, Peimbert-García, & Cárdenas-Barrón, 2021). The simulation begins with an ambulance being dispatched and ends with the patient being discharged after receiving necessary medical treatment. Through this model, It has been effectively mapped the critical stages of emergency care, highlighting the key points of intervention and the flow of activities that ensure timely and efficient patient management.

The findings from this simulation model can be instrumental in identifying areas for improvement in emergency response and hospital operations. By addressing the delays and inefficiencies highlighted in the model, healthcare providers can enhance patient care and operational efficiency. Future research could expand this model by incorporating additional variables such as patient demographics, seasonal variations in emergency cases, and more detailed healthcare provider workflows. Moreover, implementing real-time data integration could further refine the model, making it a valuable tool for dynamic decision-making in emergency healthcare management.

In conclusion, this discrete-event simulation model offers a detailed framework for understanding and optimizing the emergency care process. The study results are similar to literature (Zhang 2018). By applying the insights gained from this study, hospitals can improve their response times, patient care quality, and overall operational efficiency, ultimately leading to better patient outcomes and enhanced healthcare delivery.

Acknowledgments

This study is the project report of CENG 4205 Modelling and Simulation Lecture of 4th grade student Sümeyra Sena Koçyiğit supervised by Güney Gürsel.

References

Aminizadeh, S., Heidari, A., Dehghan, M., Toumaj, S., Rezaei, M., Navimipour, N. J., ... & Unal, M. (2024). Opportunities and challenges of artificial intelligence and distributed systems to improve the quality of healthcare service. *Artificial Intelligence in Medicine*, 149, 102779

Arslanhan, S. (2010). Does an increase in the examination number represent a significant increase in access?. Economic Policy Research Foundation of Turkey (EPRFT) Policy Topic. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. *Oxford: Clarendon*, 1892, pp.68–73.

Doudareva, E., & Carter, M. (2022). Discrete event simulation for emergency department modelling: A systematic review of validation methods. *Operations Research for Health Care*, 33, 100340.

Ganjouhaghighi, N. (2024). Improving Emergency Department Efficiency: A Study of Physician Scheduling Strategies to Reduce Patient Wait Times.

Gul, M., & Guneri, A. F. (2012). A computer simulation model to reduce patient length of stay and to improve resource utilization rate in an emergency department service system. *International Journal of Industrial Engineering*, 19(5), 221-231

Horwitz, L. I., & Bradley, E. H. (2009). Percentage of US emergency department patients seen within the recommended triage time: 1997 to 2006. *Archives of Internal Medicine*, *169*(20), 1857-1865.

Mousavi, S. R., Sepehri, M. M., & Najafi, S. E. (2024). Analysis of Importance-Performance of Scenarios for Improving Patient length of stay in the Operating Room. *Trends in Health Informatics*, 1(1), 1-8.

Pearce, S., Marr, E., Shannon, T., Marchand, T., & Lang, E. (2024). Overcrowding in emergency departments: an overview of reviews describing global solutions and their outcomes. *Internal and Emergency Medicine*, 19(2), 483-491.

Ponsiglione, A. M., Zaffino, P., Ricciardi, C., Di Laura, D., Spadea, M. F., De Tommasi, G., ... & Amato, F. (2024). Combining simulation models and machine learning in healthcare management: strategies and applications. Progress in Biomedical Engineering, 6(2), 022001.

Retezar, R., Bessman, E., Ding, R., Zeger, S. L., & McCarthy, M. L. (2011). The effect of triage diagnostic standing orders on emergency department treatment time. *Annals of emergency medicine*, 57(2), 89-99.

Soylu, S., & Tekkanat, A. (2007). Interactions amongst kernel properties and expansion volume in various popcorn genotypes. *Journal of Food Engineering*, 80(1), 336-341.

Taiwo, E. S., Zaerpour, F., Menezes, M. B., & Sun, Z. (2024). A complexity-based measure for emergency department crowding. *International Journal of Operations & Production Management*, 44(4), 768-789

Vaghani, K., Thakkar, V., Vaghasiya, S., Thaker, J., & Bhise, A. (2024, March). Implementation of Queuing Theory in Emergency Departments. In 2024 *IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation* (IATMSI) (Vol. 2, pp. 1-6). IEEE.

van Montfort, L., Dullaert, W., & Leitner, M. (2024). Task-splitting in home healthcare routing and scheduling. arXiv preprint arXiv:2406.19288.

Varanasi, S., & Malathi, K. (2024). Self-improved COOT optimization-based LSTM for patient waiting time prediction. *Multimedia Tools and Applications*, 83(13), 39315-39333.

Vázquez-Serrano, J. I., Peimbert-García, R. E., & Cárdenas-Barrón, L. E. (2021). Discrete-event simulation modeling in healthcare: a comprehensive review. *International journal of environmental research and public health*, 18(22), 12262.

Zhang, X. (2018). Application of discrete event simulation in health care: a systematic review. *BMC health services research*, 18, 1-11.