ORIGINAL ARTICLE

Modeling Discrete-Event Simulations Using Natural Language Processing: A Healthcare Application

Adnan Miski[1], Asim T. Sharif[2], Ahmed G. Bukhari[3], and Mazen Ismail[2]

[1]*Department of Industrial Engineering, Faculty of Engineering, Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia*
[2]*Department of Medical Education, Faculty of Medicine, King Abdulaziz University, Jeddah, Saudi Arabia*
[3]*Department of Medical Laboratory Sciences, Faculty of Applied Medical Sciences, King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence
Adnan Miski
Department of Industrial Engineering, Faculty of Engineering—Rabigh, King Abdulaziz University, P.O. Box 80205, Jeddah 21589
Kingdom of Saudi Arabia
e-M: agmiski@kau.edu.sa

Submission: 01 Nov. 2024
Accepted: 06 Dec. 2024

Citation
Miski A, Sharif AT, Bukhari AG, and Ismail M. Modeling discrete-event simulations using natural language processing: A healthcare application. JKAU Med Sci 2024; 31(2): 35-47. DOI: 10.4197/Med.31-2.4.

Abstract
Discrete-event simulation (DES) is widely used to model complex healthcare systems; however, manually developing these simulation models often requires extensive effort and specialized expertise. This study explored how natural language processing (NLP) techniques can automate DES model generation from text descriptions and optimize resource allocation in the healthcare domain. We used the GPT-4o large language model to demonstrate that DES models can be automatically generated from natural language prompts with accuracy comparable to traditional simulation software. The GPT-4o model successfully simulated a skin care clinic and a complex medical care facility, producing results aligned with Arena software for metrics such as average queue times and patient throughput. Additionally, GPT-4o determined the optimal resource allocation to minimize costs while satisfying the patient waiting time constraints. The automated generation of simulations shows the potential to combine NLP with DES to accelerate healthcare system modeling and optimization.

Keywords
Discrete-event simulation, Natural language processing, Healthcare simulation automation, Resource allocation optimization, Decision support systems

INTRODUCTION

Healthcare systems are complex and dynamic, requiring sophisticated modeling approaches to analyze and optimize their performance. Discrete-event simulation (DES) has emerged as a powerful tool for simulating healthcare processes, allowing researchers and practitioners to evaluate various scenarios and resource allocation strategies without disrupting real-world operations. However, the development of DES models traditionally requires specialized knowledge and can be time-consuming, limiting accessibility and widespread adoption in healthcare decision-making[1]. Recent advances in artificial intelligence (AI) and natural language processing (NLP) offer promising opportunities for streamlining and automating the DES model generation process. By leveraging language models trained on large volumes of data, there is the potential to create validated simulation models directly from the natural language descriptions of healthcare systems. This approach could significantly reduce the technical barriers to entry into DES modeling and enable the rapid exploration of multiple scenarios.

This study investigates the potential of using NLP to automate the generation of DES models for healthcare applications. We explore two research questions: (i) How can simulation engineers automatically generate DES models for healthcare systems from natural language descriptions? (ii) How can NLP techniques be used to interpret the results of these simulations to optimize resource allocation in complex healthcare systems? This study aims to streamline the development of simulation models and enhance decision-making capabilities through the automated analysis of simulation outputs. By exploring these questions, this study contributes to the growing field of AI applications in healthcare simulations. The findings may have significant implications for improving the accessibility and efficiency of healthcare system modeling, which could lead to better-informed decisions and improved system outcomes.

The article is structured as follows: Section 2 reviews related work and examines previous applications of DES and NLP in healthcare. Section 3 describes the background and methodology, detailing the language models and frameworks used to generate and evaluate simulation models. Section 4 presents the experimental setup and results and compares the performance of GPT-4o with traditional simulation software. Section 5 discusses the findings, focusing on the potential of large language models to automate simulation development and resource allocation. Finally, Section 6 concludes the study by summarizing the key findings and outlining future research directions.

RELATED WORK

DISCRETE-EVENT SIMULATION IN HEALTHCARE

Discrete-event simulation, also known as event-driven or time-to-event simulation, is a powerful technique used to evaluate complex systems and inform decision-making processes. DES is used in several industries, such as manufacturing, logistics, aviation, and healthcare. In healthcare organizations, medical facilities have extensively used DES to enhance operational efficiency, achieve cost savings, and optimize resource allocation[2]. For example, DES has been employed to evaluate the impact of operational policies, such as wait time thresholds, discharge windows, and patient mix combinations, on the behavior and performance of intensive care units (ICUs)[3]. Pradelli et al.[4] used DES to compare the efficacy of parenteral nutrition regimens with and without omega-3 fatty acids, showing that omega-3 supplementation led to cost savings in ICU and non-ICU patients across four countries. Lenin et al.[5] focused on optimizing clinic appointments and staff numbers to reduce patient waiting times using DES. Moreover, Sala et al.[6] utilized DES to analyze the impact of COVID-19 on outpatient healthcare facilities, finding that pandemic-related policies resulted in a significant reduction in the utilization of MRI equipment. Forbus et al.[7] developed a DES model to optimize physician resource allocation in pediatric hospitals, leading to improved operational efficiency and patient care. The simulation effectively modeled patient journeys, enabling the identification of bottlenecks and inefficiencies. The body of research on DES has been expanding due to the ability of this modeling approach to represent complex systems using detailed stochastic factors, which allow for the simulation of variability and uncertainty inherent in real-world processes[2].

NATURAL LANGUAGE PROCESSING-BASED SYSTEMS

NLP involves enabling machines to read, comprehend, and extract meaning from human language. It can be categorized into natural language understanding (NLU), which focuses on comprehending and extracting meaning from text, and natural language generation (NLG), which focuses on text production[8]. Recent advancements in digitalization and the abundance of textual data have enabled significant breakthroughs in NLP, such as summarization, information extraction, question answering, text generation, and sentiment analysis. Furthermore, specialized tasks such as optimizing optical character recognition (OCR) performance for videos have benefited from NLP advancements, leveraging large language models and image super-resolution for code and text extractions[9]. These technological advancements have enhanced people's lives and substantially transformed decision support and expert systems[10].

NLU is used to comprehend meaning that extends beyond the literal interpretation of individual words, encompassing the comprehension of contextual information, sentiment, and communicative intent[11, 12]. Significant results have been obtained using NLU. For instance, Brito et al.[13] leveraged NLU techniques to forecast election results based on social media data, whereas Kastrati et al.[14] employed NLU to analyze student responses and offer tailored feedback. Lin et al.[15] developed a chatbot using natural language understanding to assist construction site managers with efficient information retrieval by accurately predicting the intents and entities from their inquiries. By analyzing the semantic and pragmatic dimensions of language, NLU can enable machines to engage in more naturalistic and meaningful interactions with humans.

In contrast, NLG develops models that produce coherent, readable, and contextually relevant texts. The objective is to enable machines to automatically generate human-like languages based on data or inputs. Extensive scholarly work has been devoted to exploring NLG applications. For example, García-Méndez et al.[16] developed NLG systems capable of generating human-like text with completeness, grammatical correctness, and semantic coherence. Mulla et al.[17] used NLG to automatically generate factual, multiple-sentence, and yes/no-type questions. Lin et al.[18] applied NLG to generate natural language descriptions from structured data sources such as tables and graphs.

Engaging in both NLU and NLG is a fundamental capability of advanced large language models (LLMs). These models are highly adept at comprehending complex language inputs and generating relevant, coherent textual outputs, making them valuable tools for a wide range of applications.

AI SYSTEMS FOR DISCRETE-EVENT SIMULATION AND RESOURCE ALLOCATION IN HEALTHCARE

Developing a DES model can be time-consuming, relying heavily on high-quality data to accurately represent a system's behavior[2]. The healthcare field has recently witnessed a growing trend of incorporating AI techniques into DES models to accelerate the modeling process and mitigate associated expenses. For instance, Gartner et al.[19] linked machine learning results with a DES model to demonstrate that changing staffing patterns can reduce the overestimation of waiting times, potentially enhancing patient satisfaction. Hosseini-Shokouh et al.[20] focused on optimizing service processes in emergency departments using a combination of DES, artificial neural network algorithms, and genetic algorithms to minimize patient waiting times and improve efficiency. Olave-Rojas et al.[21] presented a hybrid emergency medical services simulation model with a machine learning approach for travel speed forecasting. In a recent study, Atalan et al.[22] integrated DES models with machine learning algorithms to estimate the number of patients treated and predict patient waiting times in healthcare institutions based on healthcare resource costs. In another recent study, Ortiz-Barrios et al.[3] integrated AI and DES to support decision-making for ICU capacity management, offering insights for timely interventions and reducing bed waiting times. These studies demonstrate the potential of leveraging AI techniques to enhance the development and accuracy of discrete-event simulation models, particularly in complex and data-intensive domains such as healthcare.

BACKGROUND AND METHODOLOGY

LANGUAGE MODELS

Language models serve as fundamental building blocks in modern NLP systems. The primary function of these models is to predict the sequence of words or codes in relation to the prior context[11]. For a set of training examples (s_1, s_2, ..., s_n), where each example expresses a sequence of words with varying lengths (w_1, w_2, ..., w_n), the predictive output of the language models can be represented as the product of the conditional probabilities, as shown in Equation (1)[23]:

$$p(s) = \prod_{i=1}^{n} p(w_n|w_1, w_2, \ldots, w_{n-1}) \qquad 1$$

LLMs are trained on large datasets to capture intricate relations between words. State-of-the-art LLMs, including generative pre-trained transformer (GPT), bidirectional encoder representations from transformers (BERT), and Claude, have significantly improved the statistical model in Equation (1) and advanced the field of NLP by demonstrating exceptional performance across a range of diverse applications. Researchers have used these LLMs by prompting models to perform tasks, fine-tuning the models according to interest, or using the models to generate code[11]. This study utilizes a transformer-based language model, specifically the GPT architecture, which demonstrates remarkable efficacy in text, code generation, and understanding.

TRANSFORMER

The transformer architecture has emerged as the foundational backbone for training state-of-the-art LLMs, surpassing the capabilities of recurrent and convolutional neural network architectures. Transformer architectures leverage an attention mechanism (Equation 2), comprising self-attention and feed-forward networks, enabling LLMs to predict parallel sequences that handle long sequences with exceptional efficiency[24].

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \qquad 2$$

where the query (Q), key (K), and value (V) matrices are derived from the input sequence, and the dimensionality of the key vectors is represented by d_k. The softmax function is used to normalize the dot products, which are then used to calculate the model's output[24].

GENERATIVE PRE-TRAINED TRANSFORMER (GPT)

Generative pre-trained transformer models, developed by OpenAI in San Francisco California, employ deep learning to understand and generate natural language. The progression of the GPT family of models, from GPT-1 to GPT-4, has demonstrated remarkable advancements in output quality and accuracy[25]. These GPT models have been widely applied in recent academic research to produce a wide range of outputs, such as text, code classification, and images to improve the efficiency in specialized domains[26,27]. This study utilized the latest iteration of the GPT model, GPT-4o, a multimodal language model that integrates and processes diverse data types and addresses complex, multi-step tasks[25]. The GPT-4o language model was used to perform DES and resource allocation, to inform and optimize healthcare decision-making.

EXPERIMENT

Prompt engineering has emerged as a crucial practice for fully leveraging the capabilities of GPT models. This practice is characterized by an iterative process of refining text input or prompts to elicit optimal responses from the language model[28]. This process can be accelerated substantially by adhering to the fundamental principles of prompt engineering. One such principle entails providing the model with comprehensive and detailed contextual information before task initiation, thereby ensuring the relevance and accuracy of the model's responses. Another principle involves breaking down complex tasks into smaller, more manageable components, which can be addressed sequentially.

A high-level framework was developed to address the research questions and facilitate the semi-automation of experiments (see Figure 1). This framework shows the language model's ability to translate textual problem descriptions into functional Python-based DESs, which are then executed on the Replit development platform.

The proposed framework comprises seven key elements. First, the process begins with a detailed textual description of the healthcare system including a clear context and task specifications prompt. Second, GPT-4o processes the prompt and applies its deep knowledge of healthcare concepts and advanced programming capabilities. Third, these programming capabilities generate a functionally accurate DES in Python code. Fourth, the generated code is executed in the Replit cloud-based integrated development environment. Fifth, the code produces results that reflect the system's performance and provides valuable insights. Sixth, these outputs are evaluated by a domain expert to ensure the validity and integrity of the findings. Finally, a feedback loop is established, enabling the insights gained from the results and expert assessments to inform refinements of the initial prompt.

SIMULATED HEALTHCARE SYSTEMS

This study examined and simulated two healthcare systems with varying levels of complexity. The first

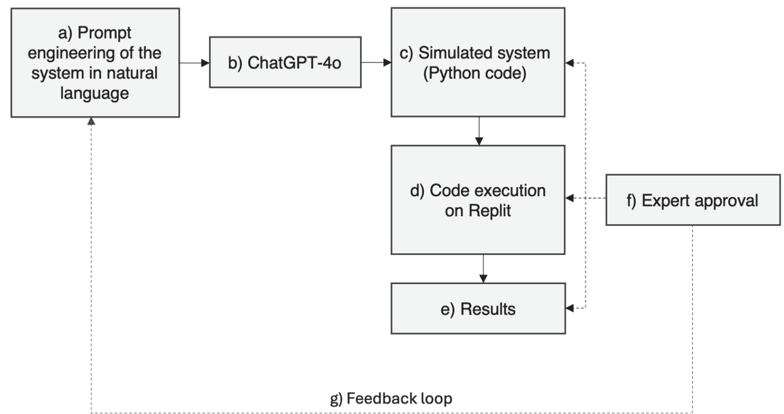

Figure 1. High-level representation of the proposed framework.

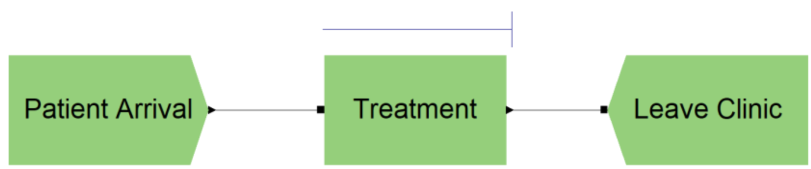

Figure 2. Skin care clinic model in Arena.

system was a skin care clinic, and the second was a medical care facility. Each system was evaluated in terms of two problems. The first problem focused on simulating the system operations, and the second problem analyzed the results from the initial simulation to optimize resource allocation. The GPT model was prompted with the first problem, and its results were validated against the traditional simulation software Arena, which is one of the most popular software for complex systems modeling[2]. Subsequently, the GPT model was prompted with the second allocation problem, and its outputs were compared with OptQuest optimization software for validation.

SKIN CARE CLINIC SYSTEM

The skin care clinic model in Arena is shown in Figure 2 and its description is as follows: Patients arrive at a skin care clinic according to an exponential interarrival time distribution with a mean of 16.2 min, with the first arrival at time 0. At the clinic, patients wait in a single first-in first-out line until one of two doctors is available to see them. The treatment process follows a triangular distribution with a minimum of 23, mode of 25, and maximum of 27. A 14-day simulation is run with 100 replications of 14 hours per day, and the number of patients exiting the system and the average waiting time are calculated.

After obtaining the results from the model, it was prompted with the following: What is the optimal number of doctors in the clinic, given that the hourly rate is $30, and the objective is to minimize the total cost while ensuring that the average waiting time for patients does not exceed 5 minutes?

MEDICAL CARE SYSTEM

The medical care facility model in Arena is shown in Figure 3 and its description is as follows: Patients arrive at a medical care facility according to an exponential interarrival time distribution with a mean of 11 min. Upon arrival, a registration receptionist checks in patients, and this process follows a triangular distribution with a minimum of 5 min, a mode of 10 min, and a maximum of 15 min. After registration, patients wait for one of two doctors at the facility. The checkup time follows a triangular distribution with a minimum of 14 min, a mode of 22 min, and a maximum of 39 min. After the checkup, ten percent of the patients may require lab tests before leaving the facility. The lab test is conducted by one nurse, and it follows a triangular distribution with a minimum of 20 min, a mode of 30 min, and a maximum of 40 min. Thirty percent of the patients may require seeing one of two specialists after the examinations. The specialist consultation follows a triangular distribution with a minimum of 25 min, a

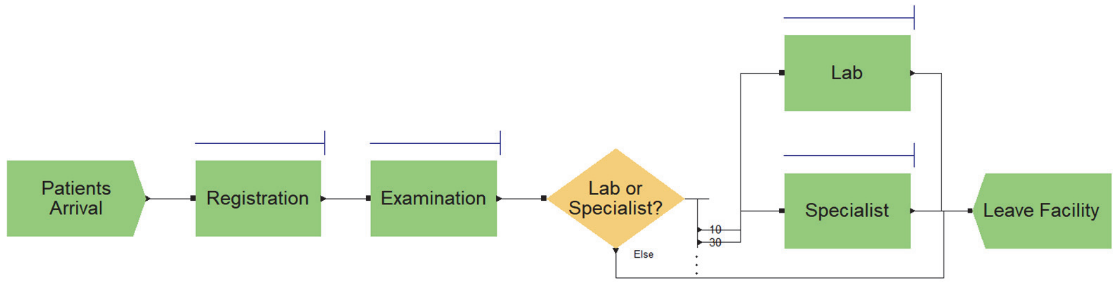

Figure 3. **Medical care facility model in Arena.**

mode of 35 min, and a maximum of 50 min. A discrete-event simulation is run for a duration of 7 days, with 100 replications of 16 hours per day. The number of patients exiting the system and the average time in queue for registration, examination room, lab tests, and consultation are calculated.

After obtaining the results from the model, it was prompted with the following: What is the optimal number of employees in the facility, given that the hourly rate for the receptionist is \$15, the nurse \$25, the doctor \$40, and the specialist \$80? The objective is to minimize the total cost while ensuring that the average waiting time for patients does not exceed 10 minutes for each queue.

RESULTS

The experiments are reproducible; the source code files are publicly accessible in the GitHub repository.

SKIN CARE CLINIC SYSTEM RESULTS

The LLM was prompted with a clear English description of the skin care clinic system. It then generated a DES model in Python with a clear explanation of the code and automatically visualized the average waiting time in the queue per replication (see Figure 4). The simulation model initiated the process by importing essential Python libraries required for random number generation and data visualization. The relevant parameters and data structures were then defined. Subsequently, the model correctly implemented the logic underlying DES. Specifically, it initiated patient arrivals at time zero and randomly generated treatment times. Next, the simulation was run with exponential patient arrival times. Finally, the model defined and executed replication steps, appending the average time in queues and the total number of patients exiting the system to a list. Notably, to ensure the validity of the results, the initial prompt should explicitly specify the usage of the SimPy library.

A comparison of the results in Table 1 shows that the simulations generated by the large language model GPT-4o yielded outputs comparable to those produced by the established industrial simulation software Arena. The average time spent in the queue is a crucial metric for assessing the efficiency of simulated healthcare systems. The GPT-4o model reported an average time in queue of 18.31 min, while the Arena software reported 18.06 min. These results suggest that the two models exhibit comparable performance, with Arena demonstrating a marginally lower average time in queue. This minor discrepancy may be attributable to differences in the respective systems' underlying algorithms or optimization techniques. Another key performance indicator is the total number of patients who successfully exited the system, which represents the throughput and capacity of the modeled healthcare system; both GPT-4o and Arena yielded comparable results.

After executing the initial simulation problem prompt and recording the results, the LLM was provided with the resource allocation problem prompt to determine the optimal number of doctors required to reduce patient waiting times and minimize the total operational costs (see Figure 5). Notably, the language model generated visualizations of the average patient waiting time, even though this metric was not specified in the original prompt. These visualizations covered the baseline simulation scenario, as shown in Figure 4, and the optimized levels, as shown in Figure 5. This automated plotting capability may reflect the prevalence of DES samples with associated data visualizations available on platforms such as GitHub[8].

Table 2 presents a comparative performance evaluation of the GPT-4o model and the OptQuest

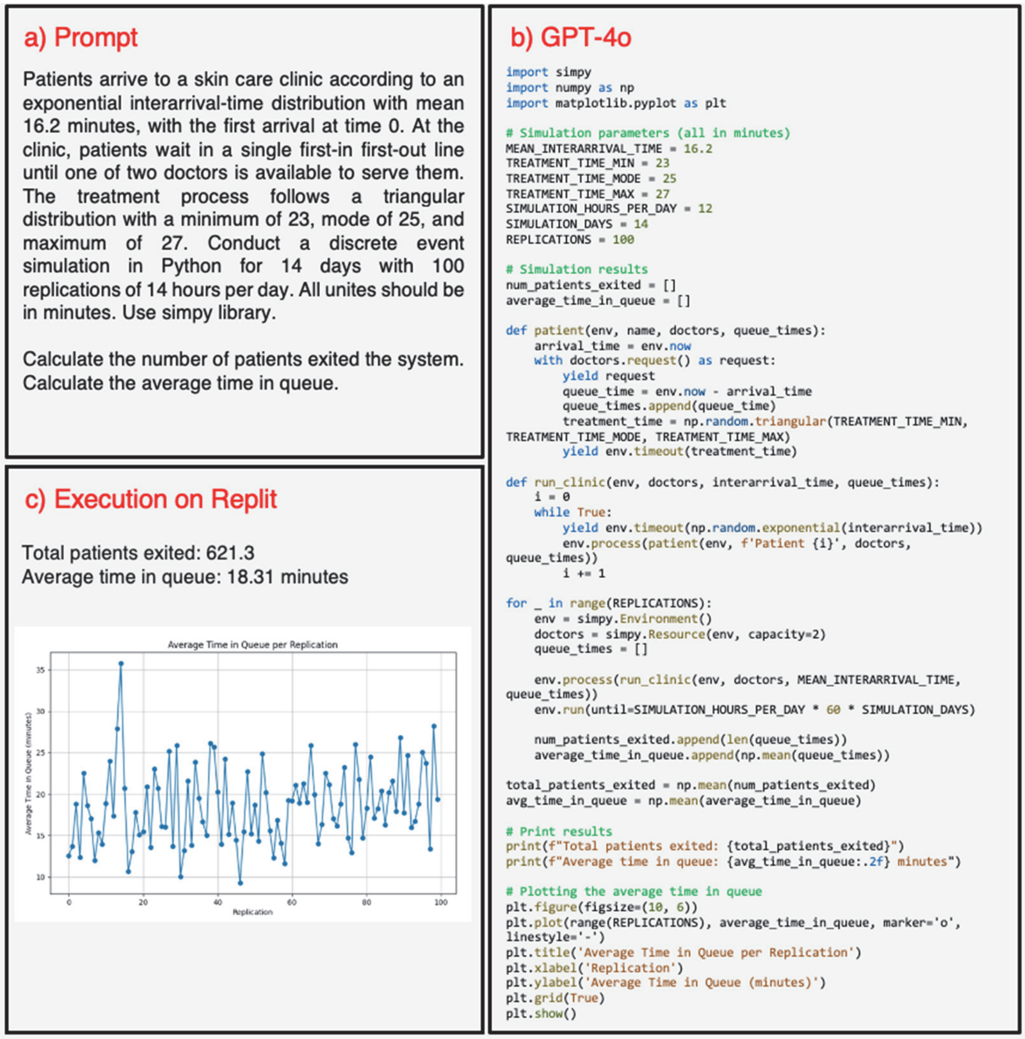

Figure 4. **Skin care system prompt and outputs.**

Table 1. **Comparing GPT-4o and Arena simulation results of the skin care clinic system.**

	GPT-4o	Arena Software
Avg. time in queue (min.)	18.31	18.06
Total patients exited the system	621.3	618

software within Arena, focusing on the average queue time and total cost for the optimal number of doctors. The results show that the average time in queue is comparable between the two approaches, suggesting similar effectiveness in modeling patient flow. While the costs associated with the GPT-4o model were slightly higher than those of Arena, the difference was insignificant, indicating that the GPT-4o model was able to achieve comparable resource optimization. Importantly, both OptQuest and GPT-4o correctly identified three doctors as the optimal staffing level, demonstrating the effectiveness of the automated simulation approach in determining the most efficient resource allocation for the healthcare system.

MEDICAL CARE SYSTEM RESULTS

The LLM generated lengthy Python code in response to the medical care problem prompt, akin to the previous simulation problem. The generated code, which is available in the GitHub repository, includes automated data visualization (see Figure 6). However, the produced code exhibited variable name conflicts. Despite this, the model was prompted again to identify and resolve the naming conflicts.

a) Prompt

What is the optimal number of doctors in the clinic, given that the hourly rate is $30, and the objective is to minimize the total cost while ensuring that the average waiting time for patients does not exceed 5 minutes.

b) GPT-4o Output

```
DOCTOR_HOURLY_RATE = 30

def calculate_total_cost(num_doctors):
    average_time_in_queue, total_patients_exited,
average_time_in_queue_list = run_simulation(num_doctors)
    cost_of_doctors = num_doctors * DOCTOR_HOURLY_RATE *
SIMULATION_HOURS_PER_DAY * SIMULATION_DAYS
    cost_of_waiting_time = average_time_in_queue *
total_patients_exited * WAITING_TIME_COST_RATE
    total_cost = cost_of_doctors + cost_of_waiting_time
    return total_cost, average_time_in_queue, total_patients_exited,
average_time_in_queue_list

# Find the optimal number of doctors
doctor_range = range(1, 11)  # Testing from 1 to 10 doctors
results = []

for num_doctors in doctor_range:
    total_cost, average_time_in_queue, total_patients_exited,
average_time_in_queue_list = calculate_total_cost(num_doctors)
    results.append((num_doctors, total_cost, average_time_in_queue,
total_patients_exited, average_time_in_queue_list))

# Extracting the optimal number of doctors
optimal_doctors = None
min_cost = float('inf')
for num_doctors, total_cost, average_time_in_queue,
total_patients_exited, average_time_in_queue_list in results:
    if average_time_in_queue <= 5 and total_cost < min_cost:
        min_cost = total_cost
        optimal_doctors = num_doctors

# Print results
for num_doctors, total_cost, average_time_in_queue,
total_patients_exited, average_time_in_queue_list in results:
    print(f"Doctors: {num_doctors}, Total Cost: ${total_cost:.2f},
Average Queue Time: {average_time_in_queue:.2f} minutes, Patients
Exited: {total_patients_exited:.2f}")

print(f"\nOptimal number of doctors: {optimal_doctors}")

# Plotting the average time in queue per replication for the optimal
number of doctors
for num_doctors, total_cost, average_time_in_queue,
total_patients_exited, average_time_in_queue_list in results:
    if num_doctors == optimal_doctors:
        plt.figure(figsize=(10, 6))
        plt.plot(range(REPLICATIONS), average_time_in_queue_list,
marker='o', linestyle='-')
        plt.title(f'Average Time in Queue per Replication for
{num_doctors} Doctors')
        plt.xlabel('Replication')
        plt.ylabel('Average Time in Queue (minutes)')
        plt.grid(True)
        plt.show()
```

c) Execution on Replit

Doctors: 1, Total Cost: $5040.00, Average Queue Time: 1761.01 minutes, Patients Exited: 402.53
Doctors: 2, Total Cost: $10080.00, Average Queue Time: 17.98 minutes, Patients Exited: 615.43
Doctors: 3, Total Cost: $15120.00, Average Queue Time: 2.38 minutes, Patients Exited: 621.40
Doctors: 4, Total Cost: $20160.00, Average Queue Time: 0.49 minutes, Patients Exited: 623.63
Doctors: 5, Total Cost: $25200.00, Average Queue Time: 0.11 minutes, Patients Exited: 618.55

Optimal number of doctors: 3

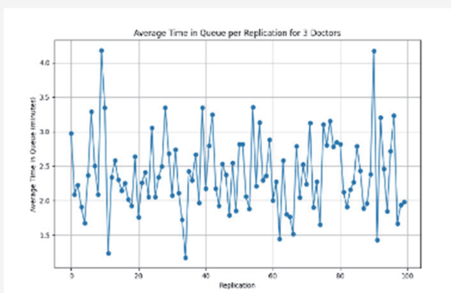

Figure 5. Skin care system resource allocation prompt and outputs.

Table 2. Comparison of GPT-4o and OptQuest optimization results of the skin care clinic system.

	Control (Number of Doctors)	GPT-4o	Arena Software (OptQuest)
Avg. time in queue (min.)	1	1761.01	1794
	2	17.98	18.065
	3	2.38	2.343
Objective function (minimize total cost)	1	$5040.00	$5038.91
	2	$10,080.00	$10,063.58
	3	$15,120.00	$15,104.13

Table 3 shows that while the GPT-4o model generally exhibited shorter average waiting times, GPT-4o and Arena demonstrated comparable performance in terms of overall patient throughput. These findings suggest that the GPT-4o language model is a viable alternative for handling complex simulation problems in healthcare systems. In fact, the GPT-4o model was able to achieve similar levels of patient throughput as the established Arena simulation software, indicating its potential to automate and streamline the simulation development process for healthcare applications.

Following the execution and analysis of the initial simulation scenario, the LLM was presented with a resource allocation prompt. The goal was to leverage the model to determine the optimal staffing

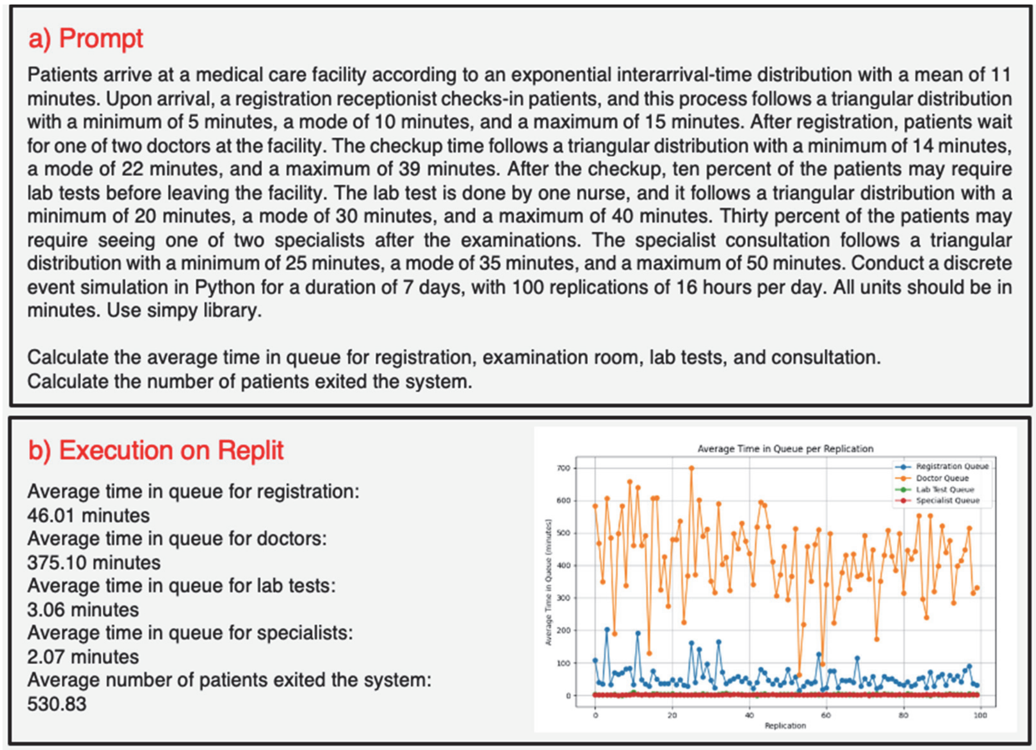

Figure 6. **Medical care system prompt and outputs.**

Table 3. **Comparing GPT-4o and Arena simulation results of the medical care system.**

	GPT-4o	Arena Software
Avg. time in queue for registration (min.)	46.01	46.47
Avg. time in queue for doctors (min.)	375.10	388.02
Avg. time in queue for lab tests (min.)	3.06	2.97
Avg. time in queue for specialists (min.)	2.07	1.755
Total patients exited the system	530.83	531.00

levels of receptionists, doctors, nurses, and specialists required to minimize patient waiting times and overall operational costs (see Figure 7).

The comparative analysis presented in Table 4 shows that the outputs generated by the LLM differed from those generated by OptQuest software. This discrepancy may be attributed to the use of divergent algorithms to calculate the average waiting time in the queue. Nonetheless, the language model successfully determined the optimal allocation of resources, which satisfies the specified objective function and constraints.

In summary, the findings suggest that GPT-4o can execute simulations with varying complexities. The model showed an understanding of the fundamental principles underlying DES, as well as the domain-specific context. Additionally, the LLM determined the optimal allocation of resources and automatically generated appropriate visualizations for the given problems. This outcome indicates that providing the model with only an English-language description of the simulation and resource allocation problem could save simulation engineers significant time and reduce the costs of acquiring traditional high-end simulation software. These results highlight the substantial potential of leveraging LLMs in decision-making processes within the healthcare system and other industries.

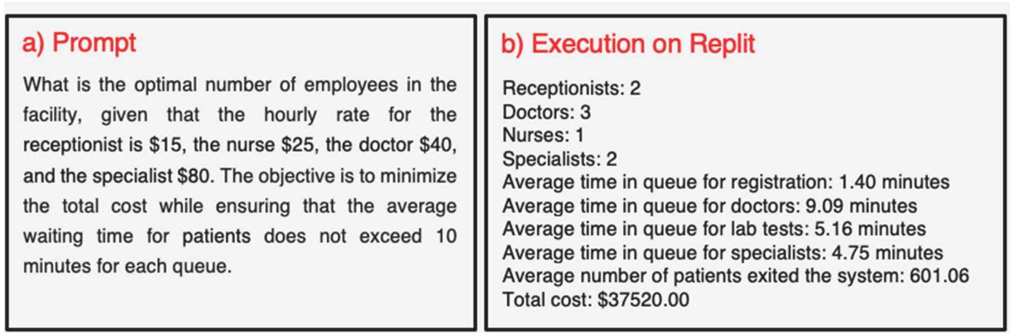

Figure 7. Medical care system resource allocation prompt and outputs.

Table 4. Comparison of GPT-4o and OptQuest optimization results of the medical care system.

	GPT-4o Optimal Value	Arena Software (OptQuest)
Receptionists	2	2
Doctors	3	3
Nurses	1	2
Specialists	2	2
Total Cost	**$37,520**	**$40,296.64**

DISCUSSION

The results of this study highlight the promising capabilities of GPT-4o in automating and optimizing the generation of DES models for healthcare applications, thereby offering improved efficiency compared to traditional simulation software.

LLMS FOR DISCRETE-EVENT SIMULATION

GPT-4o has demonstrated impressive proficiency in programming with Python, coupled with a comprehensive grasp of simulation and healthcare-specific terminology and concepts. Furthermore, this advanced model exhibits the capacity to generate functionally accurate DESs of healthcare systems based solely on textual descriptions provided in English. Earlier GPT models were limited in handling simple simulation problems due to technical limitations, such as the number of parameters used during training, and the context window size[8,30]. In contrast, GPT-4o can generate simulations for systems with varying degrees of complexity.

Although GPT-4o is a groundbreaking LLM in our study, recent advances have significantly expanded the capabilities of such language models. Meta-AI recently launched Llama, continuing with enhanced efficiency and scalability, designed to meet the demands of complex research applications better[30]. Google introduced the Gemini model, which integrates advanced multimodal capabilities, combining text, images, and other data types, further pushing the boundaries of what language models can achieve[31]. Additionally, Anthropic researchers in San Francisco, California have made considerable progress with Claude, emphasizing improvements in reasoning and language comprehension capabilities, contributing to more reliable and interpretable outputs[30]. These advancements indicate that the magnitude and complexity of simulations generated by the framework proposed in this study could continue to expand, facilitating increasingly sophisticated applications in the future.

Automation of DESs, as highlighted in our research, can reduce simulation time, resources, and costs, and minimize human error during modeling. Conventional simulation software, such as Arena, Simio, or AnyLogic requires meticulous design, comprehensive calibration, and thorough validation procedures. While these fundamental steps are essential, they can be resource-intensive and prone to human error, particularly when modeling complex systems. In contrast, our approach, which uses the capabilities of the GPT-4o model and the Replit development environment, provides a more efficient and streamlined process that can minimize human error. Furthermore, the adaptability afforded by leveraging prompt engineering techniques enables expeditious iterations and modifications, thereby transcending the intrinsic rigidity typically associated with conventional simulation software.

Although AI-powered simulation development offers significant advantages, users should recognize that these methods are not intended to replace the expertise of seasoned simulation modelers. As highlighted in this study, minor discrepancies may exist between the outputs of the GPT-4o model and those of established industrial simulation software. In addition, LLMs can generate code with bugs that require an engineer's intervention to rectify and produce valid results. Therefore, human expertise remains crucial for ensuring the accuracy and reliability of the simulation outputs.

AI-POWERED RESOURCE OPTIMIZATION FOR ENHANCED DECISION-MAKING PROCESSES

This study demonstrated that GPT-4o can determine the optimal allocation of resources to enhance simulated systems. Furthermore, the language model automatically generated and presented several viable simulation scenarios. By leveraging the capabilities of GPT-4o, the LLM can provide decision-makers with executable Python code that presents these scenarios and identifies the optimal resource allocation, thereby enabling more informed decision-making processes. Additionally, prompt engineering techniques can significantly improve the productivity of the decision-making process by enabling rapid modifications to text prompts, thus facilitating the exploration of various scenarios and their associated outcomes. These findings suggest that AI-driven resource allocation has substantial potential for improving the decision-making process.

CONCLUSION

Integrating NLP with DES has shown potential for automating the development and optimization of healthcare system models. In response to the first research question, this study successfully demonstrated that LLMs, such as GPT-4o, can generate DES models from natural language descriptions, thereby reducing the time and expertise required to model complex healthcare processes. The ability to automatically translate descriptive text into executable simulation codes allows for the rapid creation of models suitable for varying levels of complexity, making DES more accessible to healthcare professionals without specialized simulation knowledge. The study results also highlight integrating traditional simulation software with LLMs to expedite the simulation process.

For the second research question, applying NLP techniques in interpreting simulation results has proven to be effective in optimizing resource allocation. By analyzing the output of DES models, language models can generate actionable insights, such as identifying optimal staffing levels, to improve the efficiency of healthcare operations. This research demonstrates the potential of AI-driven systems to enhance decision-making processes in healthcare, providing a scalable and flexible approach to managing complex systems.

Future work can explore using Sora, an innovative framework for text-to-image and text-to-animation generation developed by OpenAI, to animate simulation models based on natural-language descriptions[32]. This emerging research area could provide a visual representation of simulation processes, further enhancing accessibility for non-technical users. Additionally, leveraging GPT models to create digital twins of healthcare facilities could allow for continuous monitoring, simulation, and optimization, thereby advancing the integration of AI in dynamic and real-time decision-making environments. Furthermore, this study highlights how traditional simulation software can integrate LLMs to expedite the simulation process. Combining the strengths of conventional simulation tools with AI-powered language models creates possibilities for faster, more efficient simulations in complex domains such as healthcare, reducing the need for manual intervention, and enabling quicker resolutions in decision-making. These future directions promise to expand the capabilities of current simulation systems, offering more sophisticated solutions to complex healthcare challenges.

CONFLICTS OF INTEREST

The authors have no conflicts of interest to declare. All co-authors have seen and agreed with the contents of the manuscript. There are no financial interests to disclose. We certify that the submission is an original work and is not under review at any other publication.

DISCLOSURE

The authors did not receive any form of commercial support, either in the form of compensation or financial assistance, for this case report. The authors have no financial interest in any of the products, devices, or drugs mentioned in this article.

ETHICAL APPROVAL

This study did not involve human participants, animal subjects, or any sensitive data requiring ethical approval. Since the research was limited to a comparative analysis of software tools, an ethical review was not applicable

DATA AVAILABILITY STATEMENT

The experiments were conducted in a reproducible manner and the source code is available in the GitHub repository: https://github.com/AGMiski/Modeling-Discrete-Event-Simulations-Using-Natural-Language-Processing-A-Healthcare-Application.

DISCLAIMER/PUBLISHER'S NOTE

REFERENCES CITED

[1] Wallrath R, Franke MB. Integrating MILP, Discrete-Event Simulation, and Data-Driven Models for Distributed Flow Shop Scheduling Using Benders Cuts. Processes 2024, 12, 1772. https://doi.org/10.3390/pr12081772.

[2] Vázquez-Serrano JI, Peimbert-García RE, Cárdenas-Barrón LE. Discrete-Event Simulation Modeling in Healthcare: A Comprehensive Review. Int. J. Environ. Res. Public Health 2021, 18, 12262. https://doi.org/10.3390/ijerph182212262.

[3] Ortiz-Barrios M, Arias-Fonseca S, Ishizaka A, Barbati M, Avendaño-Collante B, Navarro-Jiménez E. Artificial Intelligence and Discrete-Event Simulation for Capacity Management of Intensive Care Units during the Covid-19 Pandemic: A Case Study. J. Bus. Res. 2023, 160, 113806. https://doi.org/10.1016/j.jbusres.2023.113806.

[4] Pradelli L, Eandi M, Povero M, Mayer K, Muscaritoli M, Heller AR, Fries-Schaffner E. Cost-Effectiveness of Omega-3 Fatty Acid Supplements in Parenteral Nutrition Therapy in Hospitals: A Discrete-event simulation Model. Clin. Nutr. 2014, 33, 785–792. https://doi.org/10.1016/j.clnu.2013.11.016.

[5] Lenin RB, Lowery CL, Hitt WC, Manning NA, Lowery P, Eswaran H. Optimizing Appointment Template and Number of Staff of an OB/GYN Clinic—Micro and Macro Simulation Analyses. BMC Health Serv. Res. 2015, 15, 1–17. https://doi.org/10.1186/s12913-015-1007-9.

[6] Sala F, Quarto M, D'urso G. Simulation Study of the Impact of COVID-19 Policies on the Efficiency of a Smart Clinic MRI Service. Healthcare 2022, 10, 619. https://doi.org/10.3390/healthcare10040619.

[7] Forbus JJ, Berleant D. Using Discrete-Event Simulation to Balance Staff Allocation and Patient Flow between Clinic and Surgery. Modelling 2023, 4, 567–584. https://doi.org/10.3390/modelling4040032.

[8] Jackson I, Jesus Saenz M, Ivanov D. From Natural Language to Simulations: Applying AI to Automate Simulation Modelling of Logistics Systems. Int. J. Prod. Res. 2024, 62, 1434–1457. https://doi.org/10.1080/00207543.2023.2276811.

[9] Alahmadi MD, Alshangiti M. Optimizing OCR Performance for Programming Videos: The Role of Image Super-Resolution and Large Language Models. Mathematics 2024, 12, 1036. https://doi.org/10.3390/math12071036.

[10] Otter DW, Medina JR, Kalita JK. A Survey of the Usages of Deep Learning for Natural Language Processing. IEEE Trans. Neural Netw. Learn. Syst. 2021, 32, 604–624. https://doi.org/10.1109/TNNLS.2020.2979670.

[11] Min B, Ross H, Sulem E, Veyseh APB, Nguyen TH, Sainz O, Agirre E, Heintz I, Roth D. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. ACM Comput. Surv. 2023, 56, 1–40. https://doi.org/10.1145/3605943.

[12] Hu L, Liu Z, Zhao Z, Hou L, Nie L, Li J. A Survey of Knowledge Enhanced Pre-Trained Language Models. IEEE Trans. Knowl. Data Eng. 2024, 36, 1413–1430. https://doi.org/10.1109/TKDE.2023.3310002.

[13] Brito KDS, Filho RLCS, Adeodato PJL. A Systematic Review of Predicting Elections Based on Social Media Data: Research Challenges and Future Directions. IEEE Trans. Comput. Soc. Syst. 2021, 8, 819–843. https://doi.org/10.1109/TCSS.2021.3063660.

[14] Kastrati Z, Dalipi F, Imran AS, Nuci KP, Wani MA. Sentiment Analysis of Students' Feedback with Nlp and Deep Learning: A Systematic Mapping Study. Appl. Sci. 2021, 11, 3986. https://doi.org/10.3390/app11093986.

[15] Lin WY. Prototyping a Chatbot for Site Managers Using Building Information Modeling (BIM) and Natural Language Understanding (NLU) Techniques. Sensors 2023, 23, 2942. https://doi.org/10.3390/s23062942.

[16] García-Méndez S, Fernández-Gavilanes M, Costa-Montenegro E, Juncal-Martínez J, Javier González-Castaño F. A Library for Automatic Natural Language Generation of Spanish Texts. Expert Syst. Appl. 2019, 120, 372–386. https://doi.org/10.1016/j.eswa.2018.11.036.

[17] Mulla N, Gharpure P. Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications. Prog. Artif. Intell. 2023, 12, 1–32.

[18] Lin Y, Ruan T, Liu J, Wang H. A Survey on Neural Data-to-Text Generation. IEEE Trans. Knowl. Data Eng. 2024, 36, 1431–1449. https://doi.org/10.1109/TKDE.2023.3304385.

[19] Gartner D, Padman R. Machine Learning for Healthcare Behavioural OR: Addressing Waiting Time Perceptions in Emergency Care. J. Oper. Res. Soc. 2020, 71, 1087–1101. https://doi.org/10.1080/01605682.2019.1571005.

[20] Hosseini-Shokouh SM, Mohammadi K, Yaghoubi M. Optimization of Service Process in Emergency Department Using Discrete-event simulation and Machine Learning Algorithm. Arch. Acad. Emerg. Med. 2022, 10, e44. https://doi.org/10.22037/aaem.v10i1.1545.

[21] Olave-Rojas D, Nickel S. Modeling a Pre-Hospital Emergency Medical Service Using Hybrid Simulation and a Machine Learning Approach. Simul. Model. Pract. Theory 2021, 109, 102302. https://doi.org/10.1016/j.simpat.2021.102302.

[22] Atalan A, Şahin H, Atalan YA. Integration of Machine Learning Algorithms and Discrete-Event Simulation for the Cost of Healthcare Resources. Healthcare 2022, 10, 1920. https://doi.org/10.3390/healthcare10101920.

[23] Bengio Y, Ducharme R, Vincent P, Jauvin C, Ca, JU, Kandola J, Hofmann T, Poggio T, Shawe-Taylor J. A Neural Probabilistic Language Model. J. Mach. Learn. Res. 2003, 1, 1137–1155. https://doi.org/10.1162/153244303322533223.

[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention Is All You Need. Adv Neural Inf Process Syst 2017.

[25] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, Almeida D, Altenschmidt J, Altman S, et al. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774 2023.

[26] Alahmadi MD, Alshangiti M, Alsubhi J. SCC-GPT: Source Code Classification Based on Generative Pre-Trained Transformers. Mathematics 2024, 12, 2128. https://doi.org/10.3390/math12132128.

[27] Yenduri G, Ramalingam M, Selvi GC, Supriya Y, Srivastava G, Maddikunta PKR, Raj GD, Jhaveri RH, Prabadevi B, Wang W, et al. GPT (Generative Pre-Trained Transformer)—A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. IEEE Access 2024, 12, 54608–54649. https://doi.org/10.1109/ACCESS.2024.3389497.

[28] Lo, LS. The CLEAR Path: A Framework for Enhancing Information Literacy through Prompt Engineering. J. Acad. Librariansh. 2023, 49, 102720. https://doi.org/10.1016/j.acalib.2023.102720.

[29] Lee M. A Mathematical Investigation of Hallucination and Creativity in GPT Models. Mathematics 2023, 11, 2320. https://doi.org/10.3390/math11102320.

[30] Zhao Q, Li J, Liu J, Kang Z, Zhou Z. Is Word Order Considered by Foundation Models? A Comparative Task-Oriented Analysis. Expert Syst. Appl. 2024, 241, 122700. https://doi.org/10.1016/j.eswa.2023.122700.

[31] Lindemann NF. Chatbots, Search Engines, and the Sealing of Knowledges. AI Soc. 2024. https://doi.org/10.1007/s00146-024-01944-w.

[32] Brooks T, Hellsten J, Aittala M, Wang T-C, Aila T, Lehtinen J, Liu M-Y, Efros AA, Karras T. Generating Long Videos of Dynamic Scenes. Adv Neural Inf Process Syst 2022, 35, 31769--31781