Final AI & Simulation Project
A Hospital Simulation

Karen Dianelis Cantero Lopez C411
Hector Miguel Rodriguez Sosa C411
Sebastian Suarez Gomez C411

April 20, 2024

Contents

1 Summary Of the Project **3**

2 Optimization Method Used **3**

3 RAG **3**

4 Database **4**

5 Simulation **5**
- 5.1 Variables of interest . . . 5
- 5.2 Variables that describe the problem . . . 5
- 5.3 Agent-Based Simulation . . . 6
  - 5.3.1 Hospital as the environment . . . 6
  - 5.3.2 Patients as agents . . . 6
- 5.4 Simulation steps . . . 7

6 Results **8**
  - 6.0.1 Simulation 1 . . . 8
  - 6.0.2 Simulation 2. Increasing the number of beds . . . 8
  - 6.0.3 Simulation 3. Increasing the number of initial patients and lambda parameter . . . 9
  - 6.0.4 Simulation 4. Same amount of beds and patients . . . 9
  - 6.0.5 Simulation 5. More beds than patients . . . 9
- 6.1 Interpretation of the results . . . 10
  - 6.1.1 Increasing the number of beds . . . 10
  - 6.1.2 Increasing the number of initial patients and lambda parameter . . . 10
  - 6.1.3 Same amount of bed and patients . . . 11
  - 6.1.4 More beds than patients . . . 11

7 Conclusions **12**

1 Summary Of the Project

This project is a simulation of a hospital that has an initial number of patients and a set amount of ICU (Intensive Care Unit) beds and regular beds. The patients can be in a critical, grave or regular condition. They have a set of symptoms. Using the A* algorithm, the patients are assigned to the beds in a way that prioritizes the most critical patients. The patients' symptoms are treated with a treatment recommended by a RAG. The treatment is a set of medications. The patients can be discharged from the hospital when they are fully recovered from their symptoms (cured) or they feel well enough. Once a patient is discharged, that bed is then made available for another patient. The patients can also die if they are not treated in time.

2 Optimization Method Used

We have used the A* algorithm to assign the patients to the beds. The A* algorithm is a pathfinding algorithm that is used to find the shortest path between two points. In this case, we use it to find the optimal assignment of patients to beds.

The cost of an assignment is $f(x) = g(x)+h(x)$, where $g(x)$ is the cost of assigning a patient to a bed and $h(x)$ is the heuristic cost of the assignment.

We calculate $g(x)$ by taking into consideration the state/condition of the patient, the type of bed, and the age group of the patient.

We take the heuristic cost $h(x)$ as the amount of symptoms the patient has. The lower the amount of symptoms, the lower the cost. This is because the less symptoms a patient has, the less likely they are to die or get worse, and the quicker they can be discharged.

The algorithm works by evaluating the cost of each possible assignment and selecting the one with the lowest cost.

3 RAG

RAG, or Retrieval-Augmented Generation, is a type of model that combines the benefits of both retrieval-based and generative models for natural language processing tasks. The main idea behind RAG is to leverage the vast amount of information available in large text corpora by retrieving relevant documents and conditioning the generation on the retrieved documents. This allows the model to generate more informed and contextually relevant responses.

In this particular implementation, the RAG model is used to recommend medications based on a list of symptoms. The model is initialized with a dataset of medications, each with a description and potential side effects. If the use_persistence configuration is set to True, the model will use a MongoDB database to persist the medication data. The model also uses a SentenceTransformer to create embeddings for each medication description, which are used to compute cosine similarity with the input symptoms. We use MongoDB because of its easiness to use and setup, and because it allows JSON object upserting, which is ideal for the embedded vector.

The query_medications_for_patients method is the main entry point for querying the model. It takes a list of symptoms and returns a list of recommended medications. If the use_llm configuration is set to True, the model will use a Language Model (LLM) to generate a response based on the top documents retrieved from the vector query. The LLM is queried with a context that includes the top documents and the input symptoms, and the response is parsed to extract the medication names. The LLM we use in this latest iteration of the RAG is the Gemini model by Google, via their genai API.

4 Database

The data of the drugs and supplements that was used to build the database was extracted from **https://medlineplus.gov/**. For the extraction, we made a script to scrap the needed information of all the drugs and supplements from the website. In total, we extracted data of around 1700 drugs/supplements.

Our database has 4 important information for each drug.

1. Name of the drug or supplement
2. What is this drug/supplement prescribed for?
3. Side Effects
4. URL to see more information about it

All the information was stored in the Drugs.json and Drugs.csv files for its later use. The script that extracted the data was built in python and for the scrapping of the website we use BeautifulSoup (python library).

5 Simulation

5.1 Variables of interest

In this simulation, we take note of the following variables:

1. Initial number of patients in each state (critical, grave, regular)
2. New patients generated each day (Poisson distribution) and their states
3. Final number of patients in each state (critical, grave, regular)
4. Number of patients that died each day and their respective states
5. Number of patients that were discharged each day and their respective states
6. Number of patients that were cured each day and their respective states
7. Number of patients that got worse each day and their old and new states
8. Number of patients that got better each day and their old and new states
9. Number of patients that stayed the same each day and their respective states
10. Number of patients that were assigned to each type of bed each day

5.2 Variables that describe the problem

The variables that describe this problem are mainly:

1. The number of ICU beds in the hospital
2. The number of regular beds in the hospital
3. The number of initial patients of the simulation
4. The lambda parameter of the Poisson distribution

5.3 Agent-Based Simulation

5.3.1 Hospital as the environment

In this simulation, the environment is represented as a hospital. It can be characterized as follows:

1. It is an accessible environment. The environment is accessible to the patients, who can be assigned to the beds based on their condition.

2. It is an nondeterministic environment. The outcome of the patients' treatment and their bed assignment is not fully predictable.

3. It is a dynamic environment. The state of the environment changes over time as new patients are generated and existing patients are treated and discharged.

4. It is a discrete environment. The simulation is run for a set number of days, and the state of the environment is updated at the end of each day.

5.3.2 Patients as agents

In this simulation, the patients are represented as agents. They can be characterized as follows:

1. They are purely reactive agents. The patients react to their current symptoms, condition, and bed assignment to determine their behavior.

2. They are autonomous agents. The patients have a set of symptoms and a condition (critical, grave, regular) that determine their behavior.

The patients make a decision on how to react based on the following stimuli:

- The symptoms they have
- The treatment recommended by the RAG
- The bed assigned to them
- The condition/state of the patient

The patients can take the following actions:

- Cure a symptom

- Acquire a side effect
- Get better
- Get worse
- Stay the same
- Die
- Be cured

The way the patients react to the stimuli is determined by the rules of the simulation. For example, if a patient receives a treatment for a symptom, there is a high probability that the symptom will be cured, and there is a chance that the patient will acquire one or more side effects of the treatment. The patients' state (critical, grave, regular) also affects their behavior, as patients in critical condition are more likely to die, while patients in regular condition are more likely to get better.

5.4 Simulation steps

The simulation is run for a set number of days (30).

The first step is using the A* method to assign the initial patients to the beds.

The patients are then treated with the treatment (medication) recommended by the RAG. Once the patient receives the medication for a specific symptom, that symptom has a high probability of being cured and there's a chance that the patient acquires one or more of that drug's side effects.

Then there's the "evolution" phase. In this phase the patients are discharged, die, get better or get worse; all depending on the amount of symptoms the patient still has, its status (critical, grave, regular) and the bed assigned.

The last step is to generate new patients which will be added to the patients that survived the day. The amount of new patients is decided by a Poisson distribution. The symptoms are decided randomly from all possible symptoms.

This process is then repeated for the set number of days.

6 Results

We ran the simulation 10 times (because of time constraints and computational power) for each set of parameters, and we have gotten the following mean values:[1]

6.0.1 Simulation 1

We ran this simulation with the following parameters: 5 ICU beds, 10 regular beds, 50 initial patients, and a lambda of 50. We have gotten the following results:

Metric	**Mean**	**Variance**	**Std Deviation**	**Min**	**Max**
Patients discharged	15.63	8.62	2.58	12.63	18.6
Patients dead	32.93	21.63	4.31	28.2	38.33
Patients Cured	0.89	0.60	0.66	0.2	1.66
Patients that got better	9.94	3.60	1.62	8.13	11.86
Patients that got worse	21.16	13.78	3.33	17.2	25.03
Regular Patients Stayed the same	7.60	4.93	1.88	5.5	9.9
Grave patients Stayed the Same	13.22	8.74	2.66	9.9	16.06
Critical Patients Stayed the same	2.88	2.12	1.29	1.5	4.5

6.0.2 Simulation 2. Increasing the number of beds

We ran this simulation with the following parameters: 20 ICU beds, 20 regular beds, 50 initial patients, and a lambda of 50. We have gotten the following results:

Metric	**Mean**	**Variance**	**Std Deviation**	**Min**	**Max**
Patients discharged	20.77	14.45	3.58	15.43	26.06
Patients dead	27.17	31.06	5.33	19.36	34.8
Patients Cured	2.55	2.09	1.38	0.7	4.7
Patients that got better	19.18	11.28	3.17	14.83	24.1
Patients that got worse	25.95	18.35	3.93	20.6	31.83
Regular Patients Stayed the same	8.65	6.21	2.34	5.73	12.3
Grave patients Stayed the Same	13.09	9.43	2.91	9.26	17.66
Critical Patients Stayed the same	3.28	2.65	1.54	1.3	5.86

[1] The variance, standard deviation, min and max values are the mean values of its respective metric per simulation

6.0.3 Simulation 3. Increasing the number of initial patients and lambda parameter

We ran this simulation with the following parameters: 5 ICU beds, 10 regular beds, 100 initial patients, and a lambda of 100. We have gotten the following results:

Metric	Mean	Variance	Std Deviation	Min	Max
Patients discharged	29.45	27.22	4.99	22.4	37.03
Patients dead	67.43	44.08	6.30	58.46	76.7
Patients Cured	0.88	0.61	0.74	0.1	2.03
Patients that got better	14.56	9.48	2.88	10.5	18.86
Patients that got worse	40.87	30.66	5.13	33.33	48.43
Regular Patients Stayed the same	14.81	12.64	3.36	9.86	19.63
Grave patients Stayed the Same	24.04	16.52	3.88	18.6	29.43
Critical Patients Stayed the same	4.53	3.35	1.75	2.2	7.26

6.0.4 Simulation 4. Same amount of beds and patients

We ran this simulation with the following parameters: 20 ICU beds, 20 regular beds, 40 initial patients, and a lambda of 30. We have gotten the following results:

Metric	Mean	Variance	Std Deviation	Min	Max
Patients discharged	16.22	13.48	3.49	11.2	21.33
Patients dead	12.33	13.43	3.45	7.8	17.63
Patients Cured	2.36	1.91	1.31	0.66	4.43
Patients that got better	15.27	9.68	2.96	10.6	19.23
Patients that got worse	16.52	12.18	3.26	11.96	21.3
Regular Patients Stayed the same	6.37	4.59	2.07	3.36	9.56
Grave patients Stayed the Same	6.14	6.14	2.34	3.03	9.86
Critical Patients Stayed the same	2.31	1.70	1.25	0.8	4.26

6.0.5 Simulation 5. More beds than patients

We ran this simulation with the following parameters: 20 ICU beds, 40 regular beds, 30 initial patients, and a lambda of 15. We have gotten the following results:

Metric	Mean	Variance	Std Deviation	Min	Max
Patients discharged	10.39	8.39	2.71	6.76	14.63
Patients dead	3.80	2.97	1.65	1.5	6.4
Patients Cured	2.11	1.95	1.29	0.4	4.13
Patients that got better	7.68	5.86	2.31	4.76	11.43
Patients that got worse	5.4	4.5	2.03	2.7	8.5
Regular Patients Stayed the same	2.77	2.30	1.43	0.86	4.86
Grave patients Stayed the Same	2.3	1.51	1.18	0.76	4.2
Critical Patients Stayed the same	0.92	0.91	0.89	0.03	2.43

6.1 Interpretation of the results

6.1.1 Increasing the number of beds

In the second simulation, we increased the number of ICU beds and common beds to 20 each. This resulted in a higher number of patients discharged and a lower number of patients dead compared to the baseline simulation. This suggests that increasing the number of beds can help improve patient outcomes by providing more resources for treatment. The number of patients cured also increased in this simulation, indicating that more patients were able to recover from their symptoms. The number of patients that got better also increased, which suggests that the additional beds allowed for more patients to receive treatment and improve their condition. However, the number of patients that got worse also increased, which is likely due to the higher number of patients in the hospital as a result of the increased bed capacity.

6.1.2 Increasing the number of initial patients and lambda parameter

In the third simulation, we increased the number of initial patients and the number of new patients per day to 100 each. This resulted in a higher number of patients discharged and a higher number of patients dead compared to the baseline simulation. This suggests that increasing the number of patients in the hospital can lead to more patients being discharged, but also increases the risk of patient mortality. The number of patients cured decreased in this simulation, indicating that fewer patients were able to recover from their symptoms. The number of patients that got better increased in relation to the baseline simulation, which suggests that more patients were able to improve their condition. However, the number of patients that got worse also increased by double the original amount, which is likely due to the higher number of patients in the hospital.

6.1.3 Same amount of bed and patients

In the fourth simulation, we have roughly the same amount of beds and initial patients with a slightly lower lambda parameter. We can compare these results with those of the second simulation due to the fact that they both have the same amount of beds. Having the same amount of of beds and patients resulted in having more patients discharged than dead, which is an improvement when compared to Simulation 2. This also resulted in having roughly the same amount of patients getting better as patients getting worse; which is also an improvement when compared to Simulation 2. This all suggests that having the same amount of bed and patients can result in better patient care and a more positive evolution of the patient.

6.1.4 More beds than patients

In the fifth simulation, we increased the hospital capacity and decreased both the initial patients and the lambda parameter, in order to achieve a hospital that is not 'overflowing'. This simulation yielded arguably the best results in terms of patient evolution. We can observe that the deaths account for less than a third of all the patients discharged daily. This is possibly due to the fact that having more resources than patients can allow the hospital to provide a more focused service to the patients.

7 Conclusions

In conclusion, the simulation results suggest that increasing the number of beds can help improve patient outcomes by providing more resources for treatment. However, increasing the number of patients in the hospital can also lead to higher patient mortality rates. It is important to strike a balance between bed capacity and patient load to ensure that patients receive the care they need while minimizing the risk of mortality. Further research is needed to explore the optimal bed capacity for different patient populations and conditions. The simulation can be further refined by incorporating additional factors such as patient acuity, treatment protocols, and staffing levels to provide a more comprehensive analysis of hospital operations. Overall, the simulation provides a valuable tool for evaluating different scenarios and identifying areas for improvement in hospital management and patient care.