Emergent Treatment Behavior in a Multi-agent Hospital Simulation

Authors Redacted

This report discusses the outcomes of a multi-agent simulation that models the treatment of patients by doctors in a simple hospital environment. By modifying the behaviors of both doctor agents and patient agents we can observe results to help discover the best practices for hospitals treating infectious diseases in order to minimize patient deaths. We will conduct several experiments to test the various hypotheses we have developed regarding doctor communication radiuses, optimal patient treatment conditions such as initial health or severity weight, and more. Ultimately, the multi-agent simulation is trying to test the idea of local decisions versus global decisions by doctors in a hospital environment to generate the autonomy necessary for a multi-agent environment.

Commented [LS1]: Future tense? Past tense?

INTRODUCTION

Commented [LS2]: Introduction is poor. Lack of motivation and significance. Also, not enough precision in our overview, overly generic, especially about the important part: agent modeling, agent simulation, experiment objecives.

We developed a hospital simulation program in the Java programming language with the Repast Simphony toolkit. The simulation consists of two types of agents: patients and doctors. Patients arrive at the hospital and require treatment from doctors, while doctors move around the hospital to check up on patients and treat them if necessary. Each doctor ~~values~~ evaluates each patient based on their knowledge of the patient and a value function described later, and makes local decisions based on their evaluations. As we modify various parameters of the system, we ~~hope~~ investigate whether the doctors' decisions ~~will~~ lead to less patient deaths as an emergent behavior.

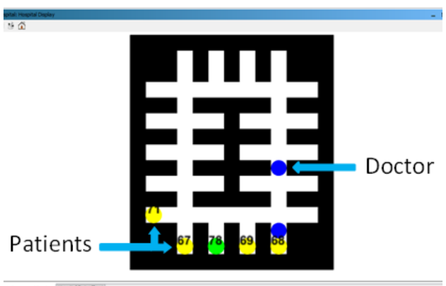
Commented [LS3]: Too hand wavy, confusing.

SIMULATION DESIGN

In this section, we detail the overall design of the hospital simulation system. We first discuss the design of the agent environment and then move on to describe the design of the two simulation agents: doctors and patients. Lastly, we take a look at the parameters of the system and what kind of measurements we use to gauge the performance of the system.

Environment Design

The environment of the hospital simulation is a 13 x 15 grid, comprised of patient rooms, a hallway for doctors to traverse, and walls. The hallway is a rectangle with a one-block width that allows doctors to go through or "pass by" one another. There are no branch hallways off of the main rectangular one, as this removes possible dead ends from the doctors' path. The reasoning behind the hallway being only a one-block width is that it simplified the value algorithm that doctors use to make their movements/decisions, which will be discussed in further detail in the doctor design section. Patient rooms are located on the exterior and interior of the hallway. Patient rooms consist of a 1 x 2 grid space where a patient occupies one grid and doctor(s) occupy the other. Only one patient can occupy a room at a time, but multiple doctors can treat a patient at one time. The rest of the grid space that does not make up the patient rooms or the hallway are considered to be "walls", or places where neither doctors nor patients can reside. The hospital environment controls the flow that patients arrive at the hospital. The environment takes the arrival rate parameter and then applies a normal distribution to that value. For example, an arrival rate of seven ticks makes it so, on average, patients will arrive every seven ticks. However, patients may arrive after one tick or may arrive after twenty ticks. This allows randomness to occur in the arrival time of patients, so that patients aren't always arriving after a set amount of ticks. We chose this approach for arrival rate because we wanted it to model the randomness of a hospital, but also provide the user with some control. Lastly, if an instance occurs where a patient arrives at the hospital and there are no rooms available at that time, the patient is turned away and is not treated or put into any sort of queue. The system then counts it as a "Turned Away" death, which is one of the measurements used to gauge the efficiency and effectiveness of the hospital system (see the measurements section for more information). A diagram of the hospital simulation environment can be viewed below:



Patient Design

The design of the patient agents is rather simple, as they are more or less there to hold values rather than make decisions. Patients arrive at the hospital with a normally distributed starting health and disease severity, which are parameters that can be set prior to running the simulation and act the same way as the arrival rate parameter previously discussed in the environment design section. We felt this accurately modeled the way patients would arrive at a hospital, as some patients are in worse conditions than others. Once a patient arrives at the hospital, they will be placed in either the first available room or in a random room, as controlled by the random room parameter.

While patients reside in their rooms, their condition can either worsen or improve. If a patient's condition is declining, they lose X health per tick, where X is determined by the severity of their disease. The more severe the disease, the higher the X value is. If a patient's condition is improving, they gain Y health per tick, where Y is again determined by the severity of their disease. Inversely, the more severe the disease, the *lower* the Y value is. The patient's condition can only be switched to improving by being treated by a doctor. In addition, while the patient is improving, there is a chance that their condition can begin to decline again. This is based on the relapse chance parameter. If a patient's overall health reaches zero, they die and are removed from their room, and if their health reaches 100, they are released from the hospital.

Doctor Design

The design of the doctor agent is more complex, as these agents make evaluations based on their environment, and decisions based on these evaluations. The sole purpose of a doctor is to move around the hospital, check up on patients, and treat them if necessary. Upon each tick, every doctor determines an evaluation, V_p, for each patient based on the value algorithm below:

$$V_p = Wd * D + Ws * S + Wh * H + Wt * T$$

where *D is ???, S is ????, H is …., T is ….,* *and the weights Wd, Ws, Wh, and Wt are ???* *and Wd + Ws + Wh + Wt* = 1.0.

Once the doctor determines which patient holds the highest valuation, they will step towards the patient or treat them if they are already adjacent to that patient. This action is deemed their "best move" for that tick. The above value algorithm takes into consideration: the length of time, *T*, in ticks that has passed since the patient's last checkup, the severity of the patient's disease,

Commented [LS5]: How does a patient decide whether its health is declining or improving? Not clear.

Commented [LS4]: Number each figure, and provide caption for each figure, and then refer to each figure by its number in the text. For example, Figure 1.

Commented [LS6]: Not clear how the curve of "decline" or "improvement" works.

Commented [LS7]: Good to have an equation. But each variable should be discussed/explained immediately after it. See my example.

Formatted: Font: Not Italic

Formatted: Font: Italic

S, the patient's health at the time of the checkup, *H*, and the distance from the doctor to the patient, *D*. ~~*Vp* is the value of checking on that certain patient, *D* is the distance to the patient, *S* is the severity of the patient's disease, *H* is the patient's health at their last checkup, and *T* is the time or ticks since the patient was last checked up on~~. *Wd*, *Ws*, *Wh*, and *Wt* are the assigned weights to distance, severity, health and time passed respectively.

When a doctor's "best move" is to treat a patient, they must remain with that patient until that patient's health is improving. The amount of time or ticks that a doctor must wait with a patient is determined by the severity of the patient's disease. Once enough treatment time has ~~been~~ passed, the patient will start improving and the doctor will be free to move again. Also note that once a doctor commits to treating a patient, they must remain with that patient for the specified amount of time. In addition, to keep the doctor's information about each patient relatively up to date, doctors will communicate with other doctors and pass on a patient's checkup information. When a patient is checked on, the doctor will broadcast that patient's basic information to other doctors in a specified radius. This broadcast radius is a parameter that can be set by the user. Overall, we felt like this design accurately represented the actions and responsiveness of a real life doctor.

Commented [LS8]: Exactly how? Equation?

Commented [LS9]: How is this specified and determined?

Emergent Behavior

Based on the design of our hospital environment and agents, we hope to see the doctors working effectively and efficiently to help treat their patients. In other words, we expect that the local decisions made by the doctors, based on their value algorithm, will help minimize the deaths that occur. We believe that the communication that will occur between the doctors will help these agents make the correct decisions to achieve this.

Parameters

Commented [LS10]: Would be better to also include a table showing each parameter's range of values, and default values, etc., and for the rest of the paper to refer back to this table, making it easier for your readers.

Below is a list of parameters that are included in our simulation that allow the user to easily manipulate many aspects of the agents and their environment:

1. *Number of Patients at Start*
   This value sets the number of patients that start off in rooms when the simulation begins.
2. *Number of Doctors*
   This value sets the number of doctors that are employed by the hospital, i.e., the number of doctor agents.
3. *Patient Arrival Rate*
   This sets the rate that patients arrive at the hospital. This value is based on a normal distribution, so the parameter value that is set by the user ends up being the average arrival time of patients. For example, if the arrival rate is set to seven, then a patient could arrive after four units of measure or after nine units of measure, but the average will be seven in the end. The units of time are in Repast ticks.

4. *Maximum Number of Patients to Arrive*
   This sets the maximum number of patients that will arrive at the hospital and is part of the simulation's termination. Once the maximum number of patients has arrived, the simulation will not spawn anymore patients and will end when all the active patients are either released from the hospital or die.
5. *Random Rooms*
   This Boolean value specifies how patients will be added to rooms. If it is set to true, patients will be randomly placed in rooms throughout the hospital. If it is set to false, patients will be grouped up close to each other
6. *Patient Arrival Health*
   This value sets the starting health that patients arrive at the hospital with. As with *Patient Arrival Rate*, this value is based on a normal distribution, so the

parameter value set by the user ends up being the average starting health out of all patients.

7. *Patient Disease Severity*
This value sets the disease severity that patients arrive at the hospital with. As with *Patient Arrival Rate*, this value is also based on a normal distribution, so the parameter value set by the user ends up being the average disease severity out of all patients. This value can only be integers from one to five, with one being the least severe and five being the most severe.
8. *Relapse Chance*
This sets the probability that on a tick, if a patient's condition is improving, that they will relapse and their health will start to decline again.
9. *Doctor Communication Radius*
This value sets the broadcast radius that doctors will be able to communicate with other doctors when they send out a patient's information after a checkup. The radius is in Repast grid units. For example, if the radius is set to three, then when a doctor broadcasts a patient's information, it will only be picked up and recorded by other doctors if those doctors are within three grid blocks.
10. *Patient Value Weights (Wd, Ws, Wh, Wt)*
These values set the respective weights that are applied to a doctor's evaluation of each individual patient that they use to make their next "best move". These weights are again, distance to the patient, the patient's disease severity, the patient's health and the time in ticks since the patient's last known checkup. Using these weights, the user is able to manipulate the preferences of doctors to test various scenarios. For example, one could change these weights to test to see if more lives are saved if doctors tend to treat patients with closer to them or further away.

Measurements

The measurements that we chose to observe are ones that provide us with the most accurate insight into the effectiveness and efficiency of the hospital simulation system. The following are descriptions of these observed values:

1. *Patients Saved*
This value describes the total number of patients that have been saved by doctors at a given tick. This value is increased whenever a patient's health reaches 100 and they are released from the hospital.
2. *Hospital Deaths*
This value describes the total number of patients that have died while they were in the hospital at a given tick. This value is increased whenever a patient's health reaches zero and they are removed from the hospital.
3. *Turned Away Deaths*
This value describes the total number of patients that were not admitted into the hospital. This value increases when a patient arrives at the hospital and there are no available rooms for that patient to reside in. We counted this as a separate measurement, because it could provide useful insight into the efficiency of patient treatment/turnaround.

EXPERIMENTAL SETUP

We ran several experiments to determine the effects of various parameters on our environment and to discover any coherent behaviors that may emerge as a result of our agent decisions. These experiments were also designed to help find a set of optimal parameters that effectively minimize the number of patient deaths occurring within the hospital. In total, there were six questions and six hypotheses that we wanted to test which resulted in six different experiments. During the first five experiments we decided to change only one parameter at a time because

Commented [LS11]: Lack of details on the relationships among these four weights.

we wanted to see what effect each individual parameter had on the number of patients being saved. In each experiment, we ran the simulation ten times for each tested parameter value with the same ten random seeds. The ~~default and~~ experimental and default parameters were ~~as follows~~shown in Tables 1 and 2, respectively.~~:~~

Table 1: Experimental Parameters

Parameter	Value 1	Value 2	Value 3	Value 4	Value 5
Broadcast Radius	1	4	7	10	13
Health Weight	3	1	0	-1	-3
Severity Weight	3	1	0	-1	-3
Number Doctors	1	2	3	4	5
Randomize Rooms	True	False	-	-	-

Table 2: Fixed Parameters

Parameter	Default Value
Patients at Start	5
Arrival Rate	7
Max Patients	500
Starting Health	70
Disease Severity	3
Relapse Chance	0.05
Randomize Rooms	False
Number of Doctors	2
Broadcast Radius	3
Distance Weight	-3
Severity Weight	3
Health Weight	0.5
Time Weight	2.5

The default parameters above were chosen because they consistently resulted in about three hundred patient deaths, which is close to half the patients arriving at the hospital. This allowed us to better observe the effects of changing parameters in our experiments.

Question 1: *Is it better to place all patients in the same area of the hospital or to place them in rooms all throughout the hospital?*

Hypothesis 1: Placing patients in the same area will lead to more effective treatment due to providers having less distance to walk to each patient. Since distance is negatively weighted in the doctors' patient valuation function, doctors will be more likely to treat all patients, and the distance from doctors to patients will be less.

Experiment 1: We ran the simulation ten times with the randomize rooms parameter set to true, and ten times again with the parameter set to false.

Question 2: *Does an increase in the doctors' broadcast radius lead to more effective treatment?*

Hypothesis 2: Greater communication by doctors will lead to less patient deaths, as there is a better chance that doctors can share knowledge when they check up on a patient. As a result, doctors will be able to check up on patients less frequently because they can benefit from other doctors checking up as well. In addition, doctors will have more up to date information about multiple patients, which will allow them to better evaluate which patient needs the most help next.

Experiment 2: For this experiment we tried several values for the doctors' broadcast radius. We started it at one and incremented it by three until it had reached thirteen. That is, we ran the simulation with broadcast radius values of one, four, seven, ten, and thirteen. The simulation was run ten times for each of these values using the same ten random seeds for each value.

Question 3: *Should doctors prioritize protecting patients who are still relatively*

Commented [LS12]: How did you arrive at these default values?

Commented [LS13]: But, there could be an infinite number of configurations that would result in 300 deaths. How did you come up with this particular configuration for your default?

healthy or saving patients who are near death?

Hypothesis 3: Doctors should focus on saving patients who are near death, as the healthier patients can survive for a longer period of time while the doctors are treating others. This will allow the doctors' treatment to be used more effectively, as they can treat those who need it most and then return to the healthier patients later.

Experiment 3: We ran the simulation with five different values for the health weight. These values were three, one, zero, negative one, and negative three. Again, the experiment was run ten times for each value. Because of the way the valuation function works, a positive health weight means that patients with higher health are considered more valuable to treat than those with lower health. Negative weights mean the doctors consider the least healthy patients to be more valuable for treatment. A health weight of zero means the doctors are indifferent to the patients' health values.

Question 4: *Should doctors prioritize patients based on the severity of their disease?*

Hypothesis 4: Doctors should focus on treating patients with the most severe diseases, as these patients' health declines rapidly. However, doctors should not weight severity *too* heavily, as patients that have been treated recently are much less likely to need treatment.

Experiment 4: This experiment was very similar to experiment three. We ran the simulation ten times for each of five different values of the severity weight. Again, we used the values three, one, zero, negative one, and negative three. Due to the nature of the valuation function, a positive severity weight means the doctors value patients with a higher severity (a more severe disease) higher than those with a lower severity. Negative weights mean patients with less severe diseases are valued higher, and a weight of zero means the doctors are indifferent to the patients' disease severity.

Question 5: *What is the relationship between adding additional doctors and patient deaths?*

Hypothesis 5: More doctors will lead to less patient deaths, but the benefit of an additional doctor will decrease with every extra doctor added. This is because there are more doctors available to treat the patients, but they cannot treat them all and as such there will still be deaths.

Experiment 5: We ran the simulation using one, two, three, four, and five doctors. Again, we ran the simulation ten times for each value.

Question 6: *Will a combination of the best parameter values from each experiment above minimize total patient deaths most effectively?*

Hypothesis 6: Combining optimal parameters from earlier experiments should help the hospital be more effective, but the combination won't result in the most optimal parameter combination. The experiments above all hold the other parameters constant at a default value, and we believe that there is a combination which involves changing multiple parameters which could lead to a better solution than what our initial experiments suggest.

Experiment 6: The Repast Simphony batch simulation tool helped us generate and run every possible combination of the five experimental parameters that we tested (1,250 total combinations). Then we were able to process that data to gather results and make comparisons between all of those combinations and the optimal combination suggested by our first five experiments.

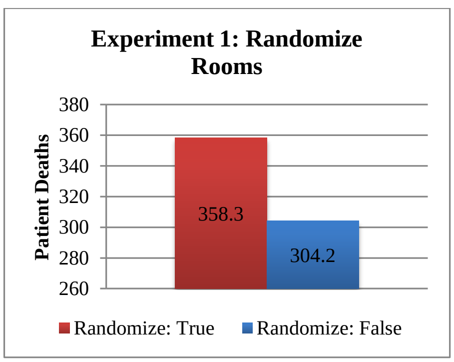
Commented [LS14]: How do you capture this in a doctor's reasoning. Strange.

Commented [LS15]: Good.

RESULTS

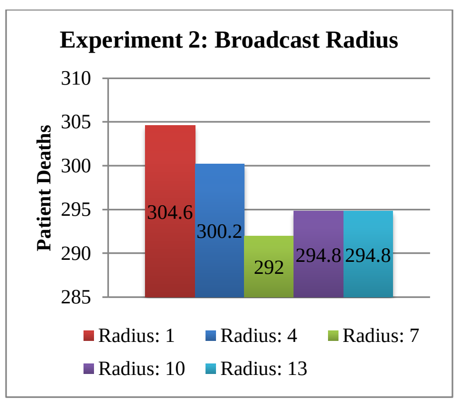
Experiment 1: Random Rooms

After averaging the results of each run, we found that placing patients into rooms near each other (randomize rooms set to false) resulted in a 15% decrease in patient deaths. We believe this decrease to have occurred because clustering patients closer together reduces the amount of time that doctors spend on travel, when they could be spending it on treating/checking up on patients. In the end our hypothesis was confirmed, in that placing patients closer together would be more beneficial than sporadically placing them about the hospital.



Experiment 2: Broadcast Radius

We found that the best value for the broadcast radius was seven, but this only resulted in a 4% decrease in patient deaths from the worst alternate value. Based on our experiments, patient deaths were highest with a broadcast radius of one, decreased with a radius of four, minimized with a radius of seven, and then increased by less than 1% when going from radius seven to radiuses of 10 and 13. We believe these outcomes occurred because the more you increase the broadcast radius, the more likely other doctors will receive the broadcasted information. They will then be able to make more accurate valuations of patients and be able to treat those that need it the most. We speculate that the increase in patient deaths with radiuses 10 and 13 was negligible due to the randomized nature of our experiments and that, in general, a higher broadcast radius would prove to be more beneficial--global knowledge vs. local knowledge. Overall our hypothesis was confirmed; however, only a 4% decrease in deaths was much smaller than we had predicted. This in part could be accredited to how we coded the doctor agents, because with them knowing the same information after receiving a broadcast, they would essentially evaluate the same patients with similar values and thus deem heading to the same patient as their “best move”. Then they would waste time by both going to treat the same patient, when one of their time could be better spent treating another lesser valued patient.



Experiment 3: Health Weight

After running this experiment, we observed that negative three was the worst value for the health weight, resulting in 364.5 hospital deaths on average. As a refresher, at a lower

Commented [LS17]: I disagree. First, this process lacked rigor. Should not simply dismiss it as ‘negligible”. Statistically, is it significant?

I could think of at least one reason why more information could be worse: swarming effect. All doctors would go to the same patient, instead of spreading themselves out …

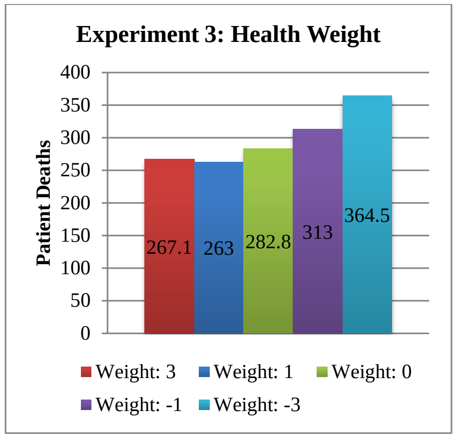
I strongly disagree with both the process and the results with this one.

Commented [LS16]: Would be better to also show the average time spent on travelling by each doctor, in each configuration. Key.

Commented [LS18]: e

Commented [LS19]: YES, this is exactly what I had in mind in the above. Since you have this, how could you dismiss those as “negligible”? Poor POJI.

weighted value, doctors value treating patients with lower health and vice versa. So, the outcome of a negative three weight contradicted our hypothesis because we believed that doctors should prioritize patients near death, when in actuality this ended up causing the most amount of patients to die. We also noted from the experiment that the amount of deaths decreased as the health weight was increased, but only until a weight of one. At this value, only 263 patients died (a 28% decrease), which made it the most optimal value. Increasing the weight from one to three resulted in 267.1 deaths on average, which was still very close to the most optimal weight, but still increased the death count slightly. We believe these results could have possibly occurred due to patients dying before doctors could reach them. In other words, since doctors are ~~valuing~~valuing the treatment of patients with lower health at a higher level, they will tend to move towards and treat patients who are about to die. Which could mean that doctors end up losing time when a patient dies. For example, say a doctor is on their way to treating a patient and then that patient dies before the doctor reaches them. The doctor now has to reevaluate and figure out which patient they should now go visit, which could be on the other side of the hospital, essentially making the doctor spend more time traveling around than treating patients. As a result, doctors should tend to prioritize the protection of healthier patients.

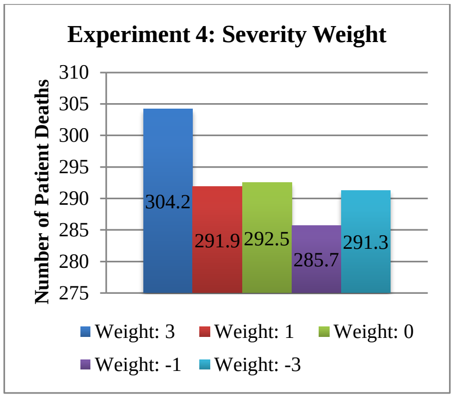


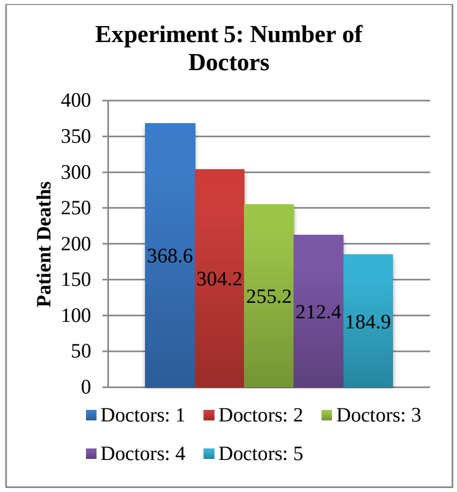
Experiment 4: Severity Weight

Through this experiment, we discovered that a severity weight of three was the worst value, with it resulting in 304.2 patient deaths on average. Patient deaths decreased as the severity weight was lowered, in that doctors should prioritize treating less severe patients. This trend continued until a severity weight of negative one was reached, which resulted in 285.7 patient deaths on average (a 6% decrease). Decreasing the weight from negative one to negative three resulted in 291.3 deaths on average, so again a very small increase in deaths. In the end, this experiment shows that doctors should slightly deprioritize patients with more severe diseases, which ends up contradicting our hypothesis, as we believed doctors should treat patients with more severe diseases. Overall, the outcome of this experiment was very similar to that of experiment three. This experiment's results lined up with the results found in experiment three, in that doctors should work to protect the healthier patients. However, the change in deaths that resulted from changing the severity weight was significantly smaller than the result of changing the health weight, deeming a patient's health to be more important when it comes to evaluating who the doctor should treat.

Commented [LS20]: You could find this number!! Tally up the number of patients who died without a doctor visit!

Now, it is just speculation and inconclusive as a result.





Experiment 5: Number of Doctors

This experiment was pretty straightforward and as we anticipated, using only one doctor resulted in the most patient deaths (368.6 on average), while using five doctors resulted in the least deaths (184.9 on average). It should be pretty obvious that by adding more doctors, more patients will be saved because there are more doctors to treat the patients. However, we did notice a diminishing return with the amount of patients saved when adding another doctor. Two doctors resulted in an 18% decrease in deaths, while adding a third resulted in a 16% decrease, adding a fourth resulted in a 17% decrease, and finally adding a fifth resulted in a 13% decrease in deaths. Overall this experiment confirmed our hypothesis that adding more doctors would result in less patient deaths, but also that the benefit of an additional doctor starts to decrease as the number of doctors increases.

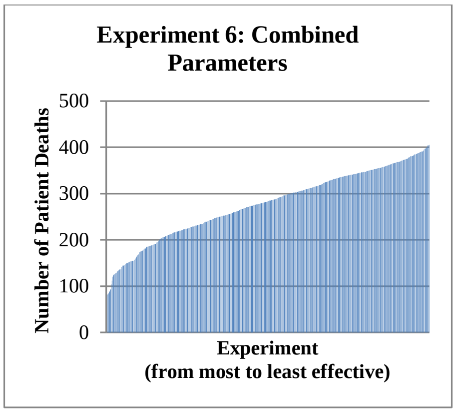
Experiment 6: Combined Parameters

Of the 1,250 different combinations that we tested in this experiment our optimal parameter solution was the 32nd most effective possible combination. The combination caused a 26.5% mortality rate (132.4 deaths). The most effective combination found in the experiment caused only 79.4 deaths which is only a 15.9% mortality rate. As you can see in the graph above there is a wide disparity of effectiveness between all the different possible combinations. Overall, these results confirmed our hypothesis that there was indeed a more optimal combination than what the results of our first five experiments suggested. However, even though our suggested combination wasn't the most effective it was still interesting to see that it fell in the top 2.6% of all the possible combinations in terms of effectiveness. Another very interesting outcome of this experiment was that the experiments which performed more effectively than our suggested combination still used some of our suggested values and/or some of the second or third best suggested parameter values.

Commented [LS21]: So many more POJI could have been done with this, to look at in general which parameter values are better, and which combinations of parameter values are better.

Commented [LS22]: Which one? Your default? I am very confused here.

Commented [LS23]: Where? Results not shown. Thus not convincing.



FUTURE WORK

While developing the simulation and conducting experiments we thought about a variety of future experiments and research that could be interesting. We also encountered a number of points in our simulation development that could have been improved upon. This section will discuss some of those ideas and improvements.

Agent Design: Patients

The relatively simple and static patient agents that we designed worked perfectly for the experiments we completed. However, we think that making more intelligent patients could be an exciting route for future experiments. One exciting possibility for intelligent patients is that they could have the ability to move around the hospital rather than only sit in their own room. This possibility could provide some intriguing results when combined with some of the other future experiments we suggest below such as the infection of other agents in the hospital. Essentially, this additional action for agents more closely reflects the real world because in a real hospital patients aren't always forced to stay in their room. We also believe that enabling patient motion would allow us to see an emergent behavior where more hospital agents are getting infected if those agents are designed with that capability.

Commented [LS24]: In a hospital, would it allow patients with infectious disease to move around? Realistic?

Agent Design: Doctors

The doctors in our experiment were designed in such a way that multiple doctors would all preference treating the same patient. Ultimately, this leads to an emergent behavior within our simulation where doctors would follow each other around the hospital treating patients in the same area rather than spreading out throughout the hospital to treat more patients at once. In future experiments we thought it would be interesting to setup the simulation so that doctors could somehow know whether or not the patient they prefer is already preferred. However, this also lends itself to more global communication within the hospital which may not be preferred for the multi-agent environment we're trying to setup.

Commented [LS25]: ?

Commented [LS26]: This is what I mentioned earlier, causing harm to the overall scheme.

Commented [LS27]: This might be irrelevant since if a doctor A receives a broadcast from another doctor D about a patient P, then A knows already that D must be there with P.

For future experiments we also thought that adding the ability for doctors to get infected would be a good idea. First of all, this possibility helps the model better reflect the real world, and it also could lead to some interesting scenarios. For example, what happens when all of the doctors get infected? Would it be beneficial for doctors to preference the treatment of other doctors before other infected patients?

Environment Design

The environment design we used in our experiments was extremely static because there was no way to alter how the hospital was laid out. In the future it would definitely be compelling to work with a more dynamic hospital environment. Specifically, we think that it would be interesting to experiment with a few different approaches of creating a dynamic hospital. The easiest way to make our hospital dynamic would be by adding the ability to add or subtract rooms within the simulation. The second interesting approach to a dynamic hospital would be adding the concept of wings to the hospital. Wings would be exciting future research because they could help limit communication between doctors, and they could potentially help bring forth the possibility of specialized doctor agents that can only work in a certain wing.

CONCLUSION

The hospital simulation we created may not be one of the most advanced/complex settings for modeling a multi-agent system, but it's still beneficial to look at and has the main characteristics of a multi-agent system. Our simulation had well-defined agents (patients and doctors) that made local autonomous decisions in an effort to achieve their own goals. Out of the doctors' goals to save as many patients as they could, a coherent behavior emerged where doctors worked together in an effective and efficient manner to minimize the ~~amount~~ number of deaths that occurred at the hospital.

With our experiments, we observed the effects that changing the doctors' treatment preferences, their manner of communication and patient room assignment would have on the amount of patients saved. The experiments with randomizing room assignments, increasing doctor communication range and increasing the amount of doctors held very straightforward results. The experiments that proved to be far more interesting were the ones dealing with changing the weights in the doctor's value algorithm for patient's health and disease severity. We believed that doctors who gave higher values to patients with more severe diseases and less health would cause there to be a smaller amount of deaths. Our reasoning behind this was that the healthier and less severe patients would be able to survive longer without being treated, whereas near death patients would need doctors to get to them faster. In the end, however, giving a higher value to more severe patients and patients closer to death, end up causing the most amount of deaths.

Overall, we felt that our simulation was able to accurately portray that of a real life hospital and provide a means of determining the best scenario to achieve the least amount of deaths. Improvements could still be made to our system, particularly in the realm of patients interaction/movement and doctors overall communication. That being said, our system was still able to demonstrate local autonomy arriving at a global emergent behavior.

Commented [LS28]: Vague term. Look at the simulation?

Commented [LS29]: Hmm …