

# **A Multi-Agent LLM Game for Healthcare Decision Support: Combining Mathematical Optimization with AI-Driven Debate for Experiential Learning**

Thavishka Gamage (M14304830)  
gamagetd@mail.uc.edu

*Faculty Advisor: Tommaso Giovannelli, Ph.D.*  
*Department of Mechanical and Materials Engineering*  
*University of Cincinnati*  
*Capstone Project -- Spring 2026*

## **Abstract**

We present an educational game designed to teach core concepts in healthcare operations and decision-making under uncertainty. In the game, players take on the role of hospital administrators who must allocate limited resources (nurses to shifts, or operating room blocks to surgical specialties) when future demand is not known with certainty. The game has five steps. First, the player makes an initial allocation. Second, an optimization solver computes the best allocations under three different assumptions about demand: expected, moderate, and worst-case. Third, three large language model (LLM) agents, each taking one of these assumptions, debate the merits of different strategies. Fourth, the player revises their allocation based on the optimization results and the debate. Fifth, the game evaluates the player's initial and revised allocations by testing each one against multiple demand scenarios and computing performance scores. The system is built in Python with a Flask web server and a browser-based interface. We implement two healthcare scenarios: emergency department nurse scheduling and operating room block allocation. Across three live debate sessions, the agents produced meaningfully different recommendations, with staffing proposals differing by 3 to 7 nurses across perspectives, and the structured debate consistently identified the key trade-off in each scenario.

*Keywords: educational games, healthcare operations, optimization under uncertainty, multi-agent systems, large language models, experiential learning*

## **1. Introduction**

Uncertainty is a basic feature of healthcare operations management. Hospital administrators face multiple sources of uncertainty, including patient demand, surgery duration, length of stay, and staff availability. These uncertain parameters are often hard to pin down because historical data

may be limited or because the underlying patterns shift over time. As a result, the same decision problem can give very different optimal solutions depending on what one assumes about the uncertain parameters. An administrator who assumes demand will follow its historical average may end up with a lean, cost-efficient staffing plan, while one who assumes demand could spike to its worst-case level may end up with a more expensive but safer plan. This trade-off between cost efficiency and safety is one of the defining challenges in healthcare operations.

The presence of multiple conflicting objectives makes these decisions even harder. Limited resources create trade-offs among stakeholders with competing needs. For example, reducing patient wait times may require additional staffing that strains budgets, while controlling costs may increase the risk of staff burnout or delays in patient care. Understanding how different assumptions about uncertainty affect each of these objectives is important for making good choices.

Games and simulations have a long history as tools for teaching operations research concepts. The MIT Beer Distribution Game, developed by Forrester in the 1950s, remains one of the most widely used classroom simulations [1, 4]. It teaches students how information delays and human overreaction produce large cost swings in supply chains. More recently, the Burrito Optimization Game developed by Snyder in collaboration with Gurobi Optimization [2] shows how interactive games can teach mathematical optimization. In that game, players place burrito trucks on a map to maximize profit, then see how their solution compares to the one found by an optimization solver. The game makes the value of optimization concrete: even experienced players struggle to match the solver as the problem grows more complex.

Recent advances in large language models (LLMs) have opened new possibilities for supporting decision-making through structured debate among multiple AI agents. Research on autonomous supply chain management has shown that teams of LLM agents, when given appropriate guardrails and curated data sharing, can manage complex operations effectively [3]. Building on these developments, we present an educational game that combines mathematical optimization with multi-agent LLM debate to teach healthcare operations decision-making under uncertainty.

The rest of this paper is organized as follows. Section 2 reviews related work on educational games and multi-agent LLM systems. Section 3 defines the two healthcare decision problems addressed by the game. Section 4 describes the rules and flow of the game. Section 5 presents the system architecture, including agent design, the debate protocol, and the integration of optimization results. Section 6 reports experimental results from live debate sessions. Section 7 discusses educational applications, and Section 8 concludes with limitations and future work.

## **2. Related Work**

### ***2.1 Educational Games for Operations Research***

The use of games for teaching operations research has a long history. The MIT Beer Distribution Game, introduced by Forrester [4] and studied by Sterman [1], places students in a simulated

supply chain where they must make ordering decisions with limited information. The game shows how rational individual decisions can produce poor system-level outcomes, a phenomenon known as the bullwhip effect. The Burrito Optimization Game [2] takes a different approach: rather than simulating a dynamic system, it presents a static optimization problem (facility location) and challenges players to solve it by hand, then compares their solution to the one found by a mathematical solver. Our game draws on both traditions. Like the Beer Game, it exposes students to the consequences of decision-making under uncertainty. Like the Burrito game, it compares the player's solution to mathematically optimal solutions computed by a solver.

## ***2.2 Multi-Agent LLM Systems***

Multi-agent frameworks in which multiple LLM instances collaborate or debate have emerged as a useful approach for complex reasoning tasks. Long et al. [3] built an autonomous supply chain testbed around the Beer Distribution Game, testing whether teams of LLM agents could manage supply chains. Their research identified four factors that matter for success: a capable model, guardrails to limit costly errors, curated data shared through a central orchestrator, and careful prompt design. With these techniques, their best systems reduced supply chain costs by up to 67 percent compared to human teams. Our system uses all four of these design principles, adapted to the healthcare decision-making context.

## ***2.3 Optimization Under Uncertainty***

Optimization under uncertainty has been studied using approaches that differ in their assumptions about uncertain parameters [5, 6]. In the simplest case, one assumes a known probability distribution and optimizes for the expected outcome. In the most conservative case, one hedges against worst-case scenarios. Between these extremes are approaches that use a moderate level of caution. Our game does not require players to know these formal frameworks. Instead, it makes this spectrum of attitudes concrete through three AI agents, each of which reasons about the optimization solution computed from its corresponding uncertainty assumption, making the trade-offs between optimism and pessimism visible through debate.

# **3. Problem Definition**

The game implements two healthcare resource allocation problems. In each problem, the player must distribute a limited pool of resources across competing uses, with uncertain demand for each use. We describe the two problems below.

## ***3.1 Emergency Department Nurse Scheduling***

A 30-bed emergency department requires nurses across three 8-hour shifts: morning (7 AM to 3 PM), afternoon (3 PM to 11 PM), and night (11 PM to 7 AM). The decision variables are the number of nurses assigned to each shift. The total nurse pool is 30, and each shift must have between 4 and 15 nurses. Each nurse costs \$500 per shift, giving a daily budget of \$15,000 at full utilization. Each nurse can handle about 5 patients per shift.

Patient demand is uncertain and varies by shift. The game models three demand scenarios representing different levels of caution. Under expected demand, 40 patients arrive in the morning, 55 in the afternoon, and 20 at night. Under moderate demand (75th percentile), the numbers increase to 48, 65, and 26. Under worst-case demand (90th percentile), the numbers are 55, 75, and 32. The afternoon shift has the highest uncertainty, with demand ranging from 35 to 80 patients across all scenarios. The morning shift faces additional uncertainty from flu season, which may increase demand by 20 percent.

Given a staffing allocation, the game evaluates performance using four metrics: average patient wait time (target under 30 minutes), staff workload (target under 8 patients per nurse), daily cost, and coverage risk (the fraction of shifts where demand exceeds nurse capacity).

The optimization solver computes three solutions, one for each demand scenario. Under expected demand, the solver minimizes total staffing cost subject to the constraint that nurse capacity meets demand for each shift, producing a lean allocation of 23 nurses at \$11,500. Under worst-case demand (90th percentile), the solver minimizes the maximum staff workload across shifts, producing a robust allocation of 31 nurses at \$15,500. Under moderate demand (75th percentile), the solver uses a combined objective that balances cost, workload, and wait time, producing an intermediate allocation of 31 nurses at \$15,500 but with lower maximum wait time than the worst-case solution.

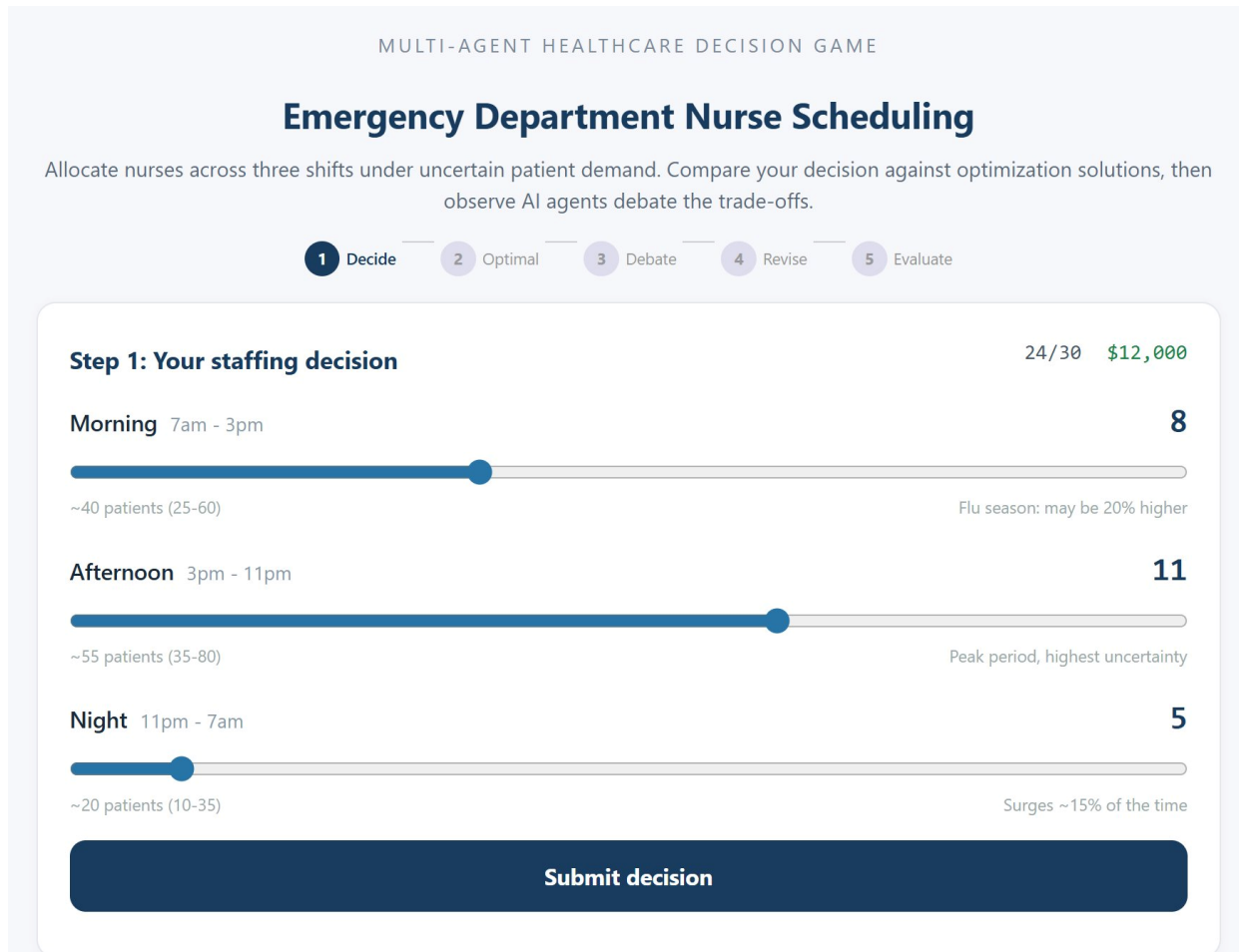
### ***3.2 Operating Room Block Allocation***

A hospital has 35 available operating room blocks per week that must be distributed across three surgical specialties: orthopedics (2 procedures per block, \$8,000 revenue per procedure, 1.5-hour average duration), general surgery (3 procedures per block, \$5,000 revenue, 1.0-hour average), and cardiac surgery (1 procedure per block, \$15,000 revenue, 3.0-hour average with high variance). The decision variables are the number of blocks assigned to each specialty. Total demand is 72 procedures (24 orthopedics, 36 general, 12 cardiac), which exceeds available capacity.

Surgical duration is uncertain. Procedures may take longer than expected, creating overtime penalties: \$2,000 per hour for orthopedics, \$1,500 for general, and \$3,000 for cardiac. Duration uncertainty is modeled through a multiplier applied to the average duration: 1.0x under expected conditions, 1.2x under moderate conditions, and 1.5x under pessimistic conditions. The competing objectives are procedure throughput (the total number of procedures completed per week), net revenue (gross revenue minus overtime penalties), overrun risk (the probability that total surgical time exceeds the allocated block time), and equitable access across specialties (measured as the minimum fraction of demand served across the three specialties, so that no single specialty is disproportionately underserved).

## **4. Game Rules and Flow**

The game is designed as a single-player educational experience. The player's goal is to find a good resource allocation for a given healthcare scenario, where "good" means balancing cost, patient outcomes, and risk. There is no single correct answer. The game is designed to show students that different assumptions about uncertainty lead to different optimal solutions, and that the choice among them depends on the decision-maker's priorities. The game proceeds through five steps, shown in the progress bar at the top of the interface (Figure 1). We illustrate these steps using the Emergency Department Nurse Scheduling scenario described in Section 3.1.



*Figure 1. Step 1 (Decide): The player allocates nurses to three shifts using slider controls. The counter at top right shows total nurses used (24/30) and cost (\$12,000). Below each slider, the expected patient demand is shown (e.g., ~40 patients indicates approximately 40 patients expected in that shift) along with the range of possible demand values in parentheses.*

**Step 1: Decide.** The player is presented with a scenario description, including the available resources, the range of possible demand values, and the key sources of uncertainty. Using slider controls, the player allocates resources (nurses to shifts, or OR blocks to specialties). A counter shows the total resources used and the associated cost in real time. The player submits their initial allocation.

## Emergency Department Nurse Scheduling

Allocate nurses across three shifts under uncertain patient demand. Compare your decision against optimization solutions, then observe AI agents debate the trade-offs.

1 Decide   2 **Optimal**   3 Debate   4 Revise   5 Evaluate

### Step 2: How does your decision compare to the optimization solver?

The solver computed optimal allocations under three different demand assumptions. The same math produces different answers depending on what you assume about tomorrow's patient demand.

#### Your decision

M:8 · A:11 · N:5 – 24 nurses, \$12,000

M: 5.0 pts/nurse, ~18 min wait | A: 5.0 pts/nurse, ~18 min wait | N: 4.0 pts/nurse, ~6 min wait

#### Cost-minimizing (assumes expected demand)

M:8 · A:11 · N:4 – 23 nurses, \$11,500

\$11,500 cost · 18 min max wait · Understaffed 93% of days in simulation.

#### Safety-maximizing (assumes 90th percentile demand)

M:11 · A:14 · N:6 – 31 nurses, \$15,500

\$15,500 cost · 35 min max wait · Handles worst-case surges. Understaffed 18% of days in simulation.

#### Balanced (assumes 75th percentile demand)

M:10 · A:14 · N:7 – 31 nurses, \$15,500

\$15,500 cost · 16 min max wait · Understaffed 50% of days in simulation. Balances cost and safety.

The \$4,000 gap between cost-minimizing and safety-maximizing is the core lesson: there is no single right answer, only trade-offs.

Next: Run AI agent debate

Back: Change my decision

Figure 2. Step 2 (Optimal): The solver shows three optimal allocations under different demand assumptions. The \$4,000 gap between cost-minimizing and safety-maximizing solutions illustrates the core trade-off.

**Step 2: Compare to optimal.** The game computes three optimal allocations corresponding to three demand assumptions: expected (optimistic), moderate (balanced), and worst-case (pessimistic). The player sees how their allocation compares to each of these solutions. For each solution, the game reports the staffing allocation, total cost, maximum wait time, and understaffed percentage, so that the player can compare all three solutions on the same set of metrics. For example, the cost-minimizing solution (M:8, A:11, N:4) costs \$11,500 with an 18-minute maximum wait but is understaffed 93 percent of the time in simulation, while the safety-maximizing solution (M:11, A:14, N:6) costs \$15,500 with a 35-minute maximum wait and is understaffed only 18 percent of

the time. This step teaches students that the same optimization model gives different solutions depending on input assumptions.



Figure 3. Step 3 (Debate), Round 1: Three agents independently analyze the scenario. The optimistic agent recommends lean staffing (M:8, A:11, N:4), the pessimistic agent hedges against surges (M:12, A:15, N:7), and the mediator proposes a middle ground (M:9, A:14, N:7). The percentage shown on each card (e.g., 85%) is the agent's self-reported confidence in its recommendation.

**Step 3: Observe the AI debate.** Three LLM agents, each assigned to one of the three demand perspectives, independently analyze the scenario and propose a strategy (Round 1). Each agent receives the optimization solution corresponding to its perspective as input. The agents then see each other's proposals and respond with critiques, areas of agreement, and potentially revised recommendations (Round 2, Figure 4). Finally, a mediator agent synthesizes the full debate into a final recommendation (Figure 5) with an explicit trade-off analysis, including what the player would gain or risk by shifting toward the optimistic or pessimistic direction.

**Optimistic Agent**

**Lean Efficiency with Strategic Buffer**

M:8 · A:12 · N:4

I'm making one strategic adjustment to my original optimization result: increasing afternoon staff from 11 to 12 nurses. This single additional nurse (\$500) provides meaningful buffer for the highest-uncertainty shift while maintaining cost discipline. Morning stays at optimized 8 nurses - adequate for expected 40 patients plus reasonable flu season variation. Night stays at minimum 4 nurses since surge probability is only 15% and we can implement surge protocols. This allocation totals 24 nurses at \$12,000 - exactly our budget target.

*I fundamentally disagree with over-preparing for low-probability scenarios. The other agents treat worst-case demand as likely outcomes, leading to systematic over-staffing. Healthcare operates on thin margins - every dollar spent on excessive staffing reduces resources for equipment, training, and other departments. Expected demand patterns exist for a reason, and our optimization shows we can meet service targets efficiently. Flexible staffing through on-call protocols and cross-training provides better surge capacity than permanent over-staffing.*

medium 85%

**Pessimistic Agent**

**Safety-First Surge Protection**

M:10 · A:15 · N:6

Healthcare demands we prepare for adverse scenarios, not optimize for averages. This allocation ensures patient safety even during demand surges: morning staffing handles flu season increases (48 patients = 4.8 per nurse), afternoon staffing at maximum capacity handles worst-case 80 patients (5.3 per nurse), and night coverage protects against the 15% surge probability. The optimization analysis shows that even under 90th percentile demand, we need robust staffing to maintain safe workloads. Better to have unused capacity than overwhelmed staff making errors.

*The fundamental disagreement is about uncertainty management. Other agents optimize for expected or 'realistic' scenarios, while I believe healthcare requires preparation for adverse scenarios. The optimistic agent's 'medium risk' tolerance is inappropriate when patient lives are at stake. The mediator's 'balanced' approach still leaves us vulnerable to predictable surge events. In healthcare, the cost of being wrong about demand is measured in patient harm and staff burnout, not just budget overruns.*

medium 85%

**Mediator Agent**

**Balanced Risk-Adjusted Staffing**

M:10 · A:14 · N:6

After reviewing both perspectives, I'm adjusting my original recommendation to reduce night staffing from 7 to 6 nurses while maintaining strong afternoon coverage. This creates a more cost-conscious approach while preserving essential safety buffers. The 14-nurse afternoon allocation can handle up to 70 patients (5.0 per nurse) comfortably and provides reasonable coverage for higher surges. Morning staffing of 10 handles expected demand plus moderate flu impact. Night reduction to 6 still provides 50% buffer above minimum safety requirements while acknowledging that overnight surges, while serious, are less frequent and can leverage on-call protocols.

*The core disagreement centers on how much buffer is appropriate for low-probability, high-impact events. The Optimistic agent accepts too much surge risk for cost savings, while the Pessimistic agent over-protects against unlikely scenarios. I believe the optimal approach balances 75th percentile demand planning with targeted buffers for the most critical periods, rather than either accepting high risk or over-engineering for extreme scenarios.*

medium 80%

Figure 4. Step 3 (Debate), Round 2: Agents respond to each other's proposals. The italicized text at the bottom of each card shows where the agents disagree.

FINAL RECOMMENDATION

## Risk-Calibrated Staffing

M:9 · A:14 · N:6

This allocation targets the 75th percentile demand scenario while providing meaningful surge protection where it matters most. The 14-nurse afternoon allocation can handle up to 70 patients (5.0 per nurse) comfortably and provides reasonable coverage for higher surges up to 80 patients. Morning staffing of 9 handles expected 40 patients plus moderate flu season impact (4.4 per nurse). Night staffing of 6 provides 50% buffer above minimum requirements while acknowledging overnight surges are less frequent. Total cost of \$14,500 represents a 21% budget premium that buys substantial risk reduction.

### Core trade-off

The player faces a classic healthcare resource allocation dilemma: accepting higher costs and potential inefficiency to protect against low-probability but high-impact demand surges, versus maintaining cost discipline while accepting increased risk during unpredictable peak periods. This is essentially choosing between the cost of unused capacity versus the cost of being overwhelmed.

**Agreement:** All agents agree that the afternoon shift (3pm-11pm) requires the heaviest staffing due to peak demand and highest uncertainty. They also agree that minimum safety requirements (4 nurses per shift) are non-negotiable, that staff workloads above 8 patients per nurse create unacceptable medical error risk, and that some level of surge protection is necessary during flu season. Finally, all recognize this is fundamentally a multi-objective optimization problem where perfect solutions don't exist.

### More optimistic

Save \$2,500 daily (17% cost reduction), meet exact budget target, free resources for other hospital priorities  
Risk: Afternoon surges above 60 patients create dangerous 5.0+ patients per nurse ratios, night shift has zero surge buffer, potential for cascading delays and patient safety issues

### More pessimistic

Maximum protection against 80+ patient afternoon surges, robust flu season coverage, minimal risk of staff burnout or patient deterioration  
Cost: Additional \$1,000 daily (7% increase), may create idle capacity on typical days, reduces budget flexibility for other needs

### Key insight

Healthcare staffing decisions require explicitly choosing your risk tolerance rather than seeking a 'perfect' solution. The mathematical optimization provides a starting point, but the final decision depends on whether you prioritize cost efficiency (accepting surge risk) or safety margins (accepting higher costs). The afternoon shift uncertainty is your critical decision point - how much are you willing to pay to avoid being caught understaffed during your busiest, most unpredictable period?

**Next: Revise your decision**

Figure 5. Final recommendation from the mediator: "Risk-Calibrated Staffing" (M:9, A:14, N:6). The synthesis includes the core trade-off, areas of agreement, and what the player would gain or lose by shifting in either direction.

**Step 4: Revise.** The player now has the chance to revise their allocation, informed by the optimization results and the AI debate. The same slider controls are available, and the player can adjust any or all of their allocations.

## Emergency Department Nurse Scheduling

Allocate nurses across three shifts under uncertain patient demand. Compare your decision against optimization solutions, then observe AI agents debate the trade-offs.

- 1 Decide   2 Optimal   3 Debate   4 Revise   5 Evaluate

STEP 5: HOW DID YOU DO?

YOUR INITIAL DECISION (M:8 A:11 N:5)

Expected Mean demand		75th %ile Above average		90th %ile Worst case	
Morning	5.0 pts/n · ~18 min	Morning	6.0 pts/n · ~45 min <span style="color:red">▲</span>	Morning	6.9 pts/n · ~58 min <span style="color:red">▲</span>
Afternoon	5.0 pts/n · ~18 min	Afternoon	5.9 pts/n · ~44 min <span style="color:red">▲</span>	Afternoon	6.8 pts/n · ~57 min <span style="color:red">▲</span>
Night	4.0 pts/n · ~6 min	Night	5.2 pts/n · ~33 min <span style="color:red">▲</span>	Night	6.4 pts/n · ~51 min <span style="color:red">▲</span>

YOUR REVISED DECISION (M:11 A:10 N:9)

Expected Mean demand		75th %ile Above average		90th %ile Worst case	
Morning	3.6 pts/n · ~2 min	Morning	4.4 pts/n · ~10 min	Morning	5.0 pts/n · ~18 min
Afternoon	5.5 pts/n · ~38 min <span style="color:red">▲</span>	Afternoon	6.5 pts/n · ~53 min <span style="color:red">▲</span>	Afternoon	7.5 pts/n · ~68 min <span style="color:red">▲</span>
Night	2.2 pts/n · ~0 min	Night	2.9 pts/n · ~0 min	Night	3.6 pts/n · ~1 min

AI RECOMMENDATION (M:9 A:14 N:6)

Expected Mean demand		75th %ile Above average		90th %ile Worst case	
Morning	4.4 pts/n · ~11 min	Morning	5.3 pts/n · ~35 min <span style="color:red">▲</span>	Morning	6.1 pts/n · ~47 min <span style="color:red">▲</span>
Afternoon	3.9 pts/n · ~5 min	Afternoon	4.6 pts/n · ~14 min	Afternoon	5.4 pts/n · ~35 min <span style="color:red">▲</span>
Night	3.3 pts/n · ~0 min	Night	4.3 pts/n · ~10 min	Night	5.3 pts/n · ~35 min <span style="color:red">▲</span>

Play again

Figure 6. Step 5 (Evaluate): Stress test grids compare the player's initial decision, revised decision, and AI recommendation under expected, 75th percentile, and 90th percentile demand. In each cell, "pts/n" denotes patients per nurse (staff workload) and "min" denotes the estimated wait time in minutes. Warning triangles highlight shifts where demand exceeds capacity.

**Step 5: Evaluate.** The game evaluates the player's initial and revised allocations by computing performance scores. A stress test grid (Figure 6) shows how both allocations perform under each of the three demand scenarios, highlighting shifts where demand would exceed capacity. The AI recommendation is also shown for comparison. This allows the player to see the consequences of their decision under different possible outcomes. The player can then play again with different initial allocations or switch to a different scenario.

## 5. System Design and Architecture

The system has five components: a game engine managing scenarios and player state, an optimization engine that solves the mathematical formulations underlying the problems described in Section 3 from multiple perspectives, a multi-agent debate system with three LLM agents, an orchestrator controlling data flow and enforcing guardrails, and a web-based interface for player interaction.

### ***5.1 Agent Persona Design***

The multi-agent debate system has three agents, each implemented as a separate call to the Anthropic Claude API (model: claude-sonnet-4-20250514) with a distinct system prompt defining its reasoning perspective.

**Optimistic agent.** This agent receives the cost-minimizing optimization solution (computed under expected demand) and argues for lean, cost-efficient allocations. It focuses on expected demand values, points out the cost of over-preparation, and accepts moderate risk in exchange for resource savings.

**Pessimistic agent.** This agent receives the safety-maximizing optimization solution (computed under worst-case demand) and argues for robust allocations that hedge against adverse scenarios. It focuses on the upper tail of demand distributions and points out the consequences of being unprepared.

**Mediator agent.** This agent receives the balanced optimization solution (computed under moderate demand). In Round 1, it proposes its own intermediate strategy. In the synthesis phase, it reviews the complete debate transcript from both rounds and produces a final recommendation. The synthesis is organized around the core trade-off, areas of agreement among all three agents, the recommended allocation with rationale, and an explicit comparison showing what the player would gain or lose by shifting toward the optimistic or pessimistic direction.

### ***5.2 Structured Debate Protocol***

The debate follows a two-round protocol plus a synthesis phase, based on the multi-agent debate approach described in [3]. In Round 1, each agent independently analyzes the scenario and proposes an allocation, producing a structured response with a strategy name, numerical allocation, rationale, risk assessment, and confidence level. In Round 2, each agent receives the other two agents' Round 1 proposals and responds with areas of agreement, points of disagreement, and a potentially revised recommendation. In the synthesis phase, the mediator agent receives the complete debate history and produces the final recommendation.

### ***5.3 Optimization Engine***

Rather than asking LLM agents to figure out optimal solutions on their own, the system solves each optimization problem programmatically and provides the relevant solution to each agent as input. The optimization engine uses the scipy optimization library, specifically the SLSQP (Sequential Least Squares Programming) algorithm from `scipy.optimize.minimize`, to solve each problem subject to constraints on shift staffing bounds and total pool size. The optimistic agent

receives the cost-minimizing solution, the pessimistic agent receives the safety-maximizing solution, and the mediator receives the balanced solution. Each agent then reasons about its optimization result, adding qualitative judgment that the solver cannot provide, such as how the recommendation would affect different stakeholders in practice. This design separates computation (which solvers do well) from qualitative reasoning (which LLMs do well).

#### 5.4 Guardrails

We implement guardrails that constrain agent outputs. Allocation recommendations are clamped to minimum and maximum bounds per shift or specialty. Total allocations are checked against the available resource pool. Budget constraints flag recommendations that exceed the daily budget. Output format validation makes sure agents produce responses that can be parsed. When a guardrail is triggered, a warning is added to the agent's output and the recommendation is adjusted to the nearest valid value.

#### 5.5 Orchestrator

The orchestrator manages the debate lifecycle: it sets up agents with their system prompts and optimization solutions, runs each debate round, enforces guardrails on all outputs, and assembles the final synthesis. It controls what information each agent receives at each stage, following the data curation principle from [3]. In Round 1, each agent sees only its own optimization solution and the scenario description. In Round 2, each agent also sees the other two agents' proposals.

#### 5.6 Web Interface

The game is served through a Flask server on localhost. Players interact through a browser-based interface with slider controls for resource allocation, real-time cost and pool tracking, scenario information with demand ranges and uncertainty notes, a five-step progress indicator (Decide, Optimal, Debate, Revise, Evaluate), a live debate display showing all three agents' reasoning across both rounds, the mediator's final synthesis with trade-off analysis, and a stress test grid showing how the player's decision and the AI recommendation perform under each demand scenario.

## 6. Experimental Results

### 6.1 Agent Debate Analysis

We ran the multi-agent debate three separate times on the nurse scheduling scenario, each producing a complete two-round debate with synthesis. Table 1 summarizes the final recommendations across all three runs.

**Table 1.** Final synthesis recommendations across three debate sessions.

Run	Strategy	M	A	N	Total	Cost
1	Afternoon-Focused Risk Mgmt	9	14	5	28	\$14,000
2	Strategic Risk-Adjusted	9	13	5	27	\$13,500

3	Targeted Surge Protection	10	13	6	29	\$14,500
---	---------------------------	----	----	---	----	----------

Several patterns come out of the debate data. The optimistic agent consistently recommended the cost-minimized solution (M=8, A=11, N=4, totaling 23 nurses at \$11,500), which matches the optimization result under expected demand. The pessimistic agent proposed much higher staffing (30 to 34 total nurses), especially for the afternoon shift, where it consistently allocated the maximum of 15 nurses. The mediator produced an intermediate recommendation in all three runs.

The debate produced real position changes. In all three runs, the optimistic agent increased its afternoon allocation by one nurse (from 11 to 12) after seeing the other agents' proposals, acknowledging the peak-period risk. In Run 3, the pessimistic agent triggered the budget guardrail with a \$17,000 proposal and then revised its recommendation downward in Round 2. These adjustments show that the structured debate protocol produces adaptive reasoning rather than rigid adherence to initial positions.

All three synthesis recommendations fell within a narrow but distinct range: 27 to 29 total nurses at \$13,500 to \$14,500. The afternoon shift consistently received the highest allocation (13 to 14 nurses), reflecting the agents' agreement that the afternoon peak is the most important source of uncertainty.

## 6.2 Optimization Results

Table 2 presents the optimization results for the nurse scheduling scenario from three demand perspectives.

**Table 2.** Optimization results under three demand perspectives.

Perspective	M	A	N	Total	Cost	Max Wait
Cost-minimizing (expected)	8	11	4	23	\$11,500	18.0 min
Safety-first (90th %ile)	11	14	6	31	\$15,500	35.4 min
Balanced (75th %ile)	10	14	7	31	\$15,500	15.6 min

The \$4,000 gap between the cost-minimizing and safety-first solutions is a 35 percent cost increase. This gap is what motivates the multi-agent debate: each agent must explain why its underlying demand assumption makes more sense for the decision at hand. The game presents all three solutions side by side so that the player can see the concrete trade-offs before watching the debate.

## 7. Educational Applications

### 7.1 Learning Objectives

The game is designed to support the following learning objectives. First, students learn that the same optimization model produces different optimal solutions depending on what one assumes

about the uncertain parameters. This is the central lesson of the optimization comparison step. Second, students observe how structured debate among agents with different perspectives can bring out trade-offs that a single analysis would miss. Third, students gain experience evaluating decisions under multiple scenarios through the stress test step, where they see how their allocation performs under expected, moderate, and worst-case demand. Fourth, students develop intuition for the relationship between mathematical optimization and qualitative judgment, as the agents ground their debate in optimization results while adding practical reasoning about risk and stakeholder impact.

## **7.2 Course Integration**

We envision three modes of use. In a lecture demonstration, the instructor runs the demo script (which requires no API key) to walk through a scenario, showing how different assumptions lead to different optimal solutions and how the agents debate the trade-offs. In an interactive session, students play the game in the browser, make their own staffing decisions, observe the AI debate, and compare their scores across rounds. In an assignment, students analyze the optimization results and debate transcripts from multiple runs and write a report on the trade-offs in their assigned scenario.

## **8. Discussion and Limitations**

Several limitations should be noted. The optimization models are simplified versions of real healthcare operations. Actual nurse scheduling involves shift preferences, skill mix requirements, and multi-day planning horizons not captured in our single-day model. The agent debate outputs are not deterministic: the same scenario produces different debates on different runs. While this variability is educationally interesting (it shows students that there is no single correct analysis), it makes formal evaluation harder. The system requires access to the Anthropic Claude API for the live debate, which introduces cost (about \$0.01 to \$0.03 per debate) and latency. A demo mode with pre-scripted agent responses is available for settings where API access is not practical.

Despite these limitations, the system shows the educational value of combining optimization with multi-agent debate. The agents consistently identify the key trade-offs in each scenario, produce diverse recommendations that span the cost-safety spectrum, and generate synthesized insights that are coherent and relevant to the learning objectives.

## **9. Conclusion**

We have presented an educational game that combines mathematical optimization with multi-agent LLM debate to teach healthcare operations decision-making under uncertainty. The game walks players through a structured five-step process: making an initial decision, comparing it to optimization results, observing an AI debate, revising the decision, and evaluating the outcome. The system implements three agent personas representing different attitudes toward uncertainty, a structured debate protocol with guardrails and data curation, and an optimization engine that

provides each agent with the solution corresponding to its perspective. Experimental evaluation across three live debate sessions shows that the system produces diverse and educationally valuable recommendations that consistently identify the key trade-offs. Future work will extend the system with additional healthcare scenarios, conduct formal user studies measuring learning outcomes, and adapt the system for real-time competitive sessions.

## References

- [1] J. D. Sterman, "Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment," *Management Science*, vol. 35, no. 3, pp. 321-339, 1989.
- [2] L. V. Snyder, "The Burrito Optimization Game," *OR/MS Today*, INFORMS, 2022. Available: <https://doi.org/10.1287/orms.2022.04.14>
- [3] C. Long, D. Simchi-Levi, A. P. Calmon, and F. P. Calmon, "When supply chains become autonomous," Working Paper, Harvard University and MIT, 2025.
- [4] J. W. Forrester, *Industrial Dynamics*. Cambridge, MA: MIT Press, 1961.
- [5] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton, NJ: Princeton University Press, 2009.
- [6] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595-612, 2010.
- [7] Anthropic, "Claude API Documentation," <https://docs.anthropic.com>, 2025.
- [8] T. Giovannelli, "A multi-agent LLM game for healthcare decision support," Project Description, University of Cincinnati, 2026.

## Appendix A: Installation and Usage Guide

### A.1 Prerequisites

The system requires Python 3.10 or higher and pip. An Anthropic API key is required only for the live AI debate; all other features (optimization, evaluation, demo) work without an API key.

### A.2 Installation

Download and extract the project files, then install dependencies:

```
unzip healthcare_game.zip && cd healthcare_game && pip install anthropic flask
```

### A.3 Running the Demo (No API Key Required)

The demo runs optimization, a simulated agent debate, scoring, and evaluation:

```
python demo.py
```

### A.4 Running the Live AI Debate

Set the API key and run the command-line interface:

```
$env:ANTHROPIC_API_KEY = "sk-ant-your-key" # PowerShell
export ANTHROPIC_API_KEY="sk-ant-your-key" # Mac/Linux
python main.py
```

### A.5 Running the Web Game

Start the Flask server and open the game in a browser:

```
python server.py # Open http://localhost:5000
```

### A.6 Running Benchmarks

```
python evaluation/benchmark.py # Full benchmark
python evaluation/benchmark.py --quick # Fast version
python evaluation/benchmark.py --export r.json # Export
```

### A.7 Project Structure

```
healthcare_game/
  demo.py          # Demo (no API key)
  main.py          # CLI with live Claude debate
  server.py        # Flask web server
  agents/          # Agent modules
  optimization/    # scipy solvers
  scenarios/       # Game scenarios
  orchestrator/    # Debate flow
  evaluation/      # Benchmarking
```

### A.8 API Endpoints

When running `server.py`, the following REST API endpoints are available: GET `/api/health` (health check), GET `/api/scenarios` (list scenarios), GET `/api/optimize/<id>` (optimization results from three perspectives), POST `/api/evaluate/<id>` (evaluate a player's decision), and POST `/api/debate/<id>` (run the full multi-agent debate).