DOI: 10.2478/ijasitels-2025-0007

Implementation of Heuristic Algorithms for Simulating Crisis Situations in the Medical System

Pătrăușanu Andrei, Buta Andra-Paraschiva,
Florea Adrian

Abstract

Healthcare systems face significant challenges during crises such as pandemics or mass-casualty events, where resource shortages and patient overflow require rapid, optimized decisions. This paper proposes a simulation-based approach to model and improve hospital resource allocation using a heuristic method—Genetic Algorithms (GA). The aim is to explore how intelligent algorithms can support decision-making under pressure by assigning patients to limited ICU beds and available doctors, considering constraints such as treatment duration, medical priority, and resource availability.

The core method involves evolving allocation strategies over multiple generations, using tournament selection, crossover, mutation, and fitness-based evaluation to optimize both resource usage and patient coverage. The simulation is implemented as a desktop application in C#, with a SQL Server database and an interactive GUI that allows users to run scenarios, configure parameters, and visualize outcomes.

Compared to a First-Come-First-Served (FCFS) baseline, the GA consistently achieves higher efficiency, treating more high-priority patients and reducing resource bottlenecks. The original contribution lies in the dual-resource optimization model and its integration into a flexible, user-friendly tool. Results demonstrate that heuristic-driven simulations can support better planning and training in emergency healthcare environments.

Keywords: genetic algorithm; heuristic optimization; hospital resource allocation; crisis simulation; healthcare systems; emergency planning

1. Introduction

In recent years, healthcare systems around the world have been repeatedly tested by large-scale crises such as pandemics, natural disasters, and other emergency events. These scenarios typically lead to significant resource imbalances, overwhelmed medical staff, and critical delays in patient care. As hospitals operate under intense pressure, the need for intelligent tools that can assist with real-time decision-making becomes increasingly urgent.

Traditional resource allocation methods, such as First-Come-First-Served (FCFS), often fail to deliver efficient or equitable outcomes in such high-demand situations. In contrast, heuristic and metaheuristic algorithms—particularly Genetic Algorithms (GAs) [1] —have shown great promise in solving complex scheduling and optimization problems where multiple constraints must be balanced.

This paper presents a simulation-based approach for improving hospital resource allocation during crisis conditions using a Genetic Algorithm. The proposed system models real-world constraints by assigning patients to limited ICU beds and medical staff, taking into account treatment duration, resource availability, and patient priority.

The goal of this research is to demonstrate how intelligent algorithms can support healthcare professionals in planning and training for crisis events by simulating alternative allocation strategies. A desktop application was developed for this purpose, offering an interactive interface, parameter configuration, and visual comparison between heuristic and traditional methods.

2. Related Work

Numerous studies have addressed the problem of resource allocation in healthcare systems, especially during crisis scenarios such as pandemics or mass-casualty events. Traditional scheduling methods are often insufficient in such contexts due to the high complexity and uncertainty involved. As a result, heuristic and metaheuristic algorithms have emerged as effective tools for optimizing resource distribution under pressure.

In [2] the authors proposed a Genetic Algorithm (GA)-based scheduling model for patient assignment, treating the problem similarly to a job-shop scheduling task. Their results showed improved efficiency and fairness compared to manual allocation methods. The authors of [3] extended this approach by integrating realistic medical workflows, emphasizing better hospital stay management through evolutionary algorithms. Other studies have explored alternative metaheuristics. The work from [4] employed Particle Swarm Optimization (PSO) for inter-hospital medical staff coordination, while [5] proposed a hybrid system combining deep learning with genetic search to improve dynamic healthcare planning. Additionally, [6] adapted GA for emergency nurse scheduling, incorporating both hard and soft constraints to maintain fairness and coverage.

In [7], the authors used the NSGA-II [8] algorithm to optimize the bed allocation in ICU, considering different levels of uncertainty: the amount of emergency patients and their corresponding length of stay, but also the length of stay of the current ICU patients. In [9] the authors present a hybrid approach of a Discrete Event Simulation and a GA (genetic algorithm) in order to minimize the waiting time for rehabilitation by varying the care unit capacity. A similar approach is present in [10], where the main optimization flow is driven by NSGA-II algorithms and the solutions evaluated by the Discrete Event Simulation. The results showed a promising alternative to the bed allocation in Brazil hospitals. In [11], a genetic algorithm approach is followed to obtain a score-based priority mechanism to efficiently use the healthcare resources in hospital units.

Other similar works, such as [12] use concepts such as operation research [13] to evaluate the responsiveness in healthcare systems under the pressure of a medical crisis. In [14] a hybridization of Genetic Algorithm and extreme learning machine to identify optimal medical events that contribute to mortality of patients is described as a part of a larger prediction mechanism. Works such as [15] exploits the parallelism of individuals evaluation in genetic algorithms in the field of patients distribution. In [16] two strategies such as first come first served and shortest processing time rules are used in conjunction within an heuristic algorithm. Their approach is used to schedule patient treatment depending on the bed allocation in a hospital in Australia. The results concluded a reduction of 8% of waiting time.

A comprehensive study of resource allocation managements in crisis circumstances has been accomplished in [17]. One of the main conclusions of the paper was that the analyzed SRA (Scarce Resource Allocation) protocols resulted in ambiguous lottery system allocation, hence the need for standardization and refinement. The impact of COVID-19 struck many areas of treatment and healthcare. Thesis [18] emphasizes the use of a genetic algorithm with a local search operator for finding optimal configuration of a surrogate of an initial agent -based system used for hospital scheduling areas.

It is also worth mentioning frameworks and tools [19], [20], [21] which prove the effectiveness of implementation and integration of heuristic search in fields such as length of stay, bed allocation scheduling etc.

Despite these contributions, most existing systems either target a single resource type or are limited to theoretical frameworks. In contrast, the present work introduces a dual-resource allocation model (beds and doctors), integrated into a simulation tool with a user-friendly interface that enables practical experimentation and comparative analysis.

3. Methodology

3.1 Problem Formulation

This study addresses the allocation of limited hospital resources—specifically ICU beds and doctors—during crisis scenarios. The goal is to create an optimized allocation plan that reduces waiting time and maximizes the treatment of high-priority patients. The problem is modeled as a constrained multi-objective optimization task, considering both treatment priorities and total resource usage.

3.2 System Architecture

The proposed solution is implemented as a Windows Forms desktop application using C#. It integrates with a SQL Server database to store and retrieve entities such as patients, beds, doctors, and allocations.

The application follows a three-layer architecture:

- Presentation Layer: user interface for simulation control.
- Business Logic Layer: implementation of Genetic Algorithm and evaluation logic.
- Data Access Layer: interaction with the underlying SQL database.

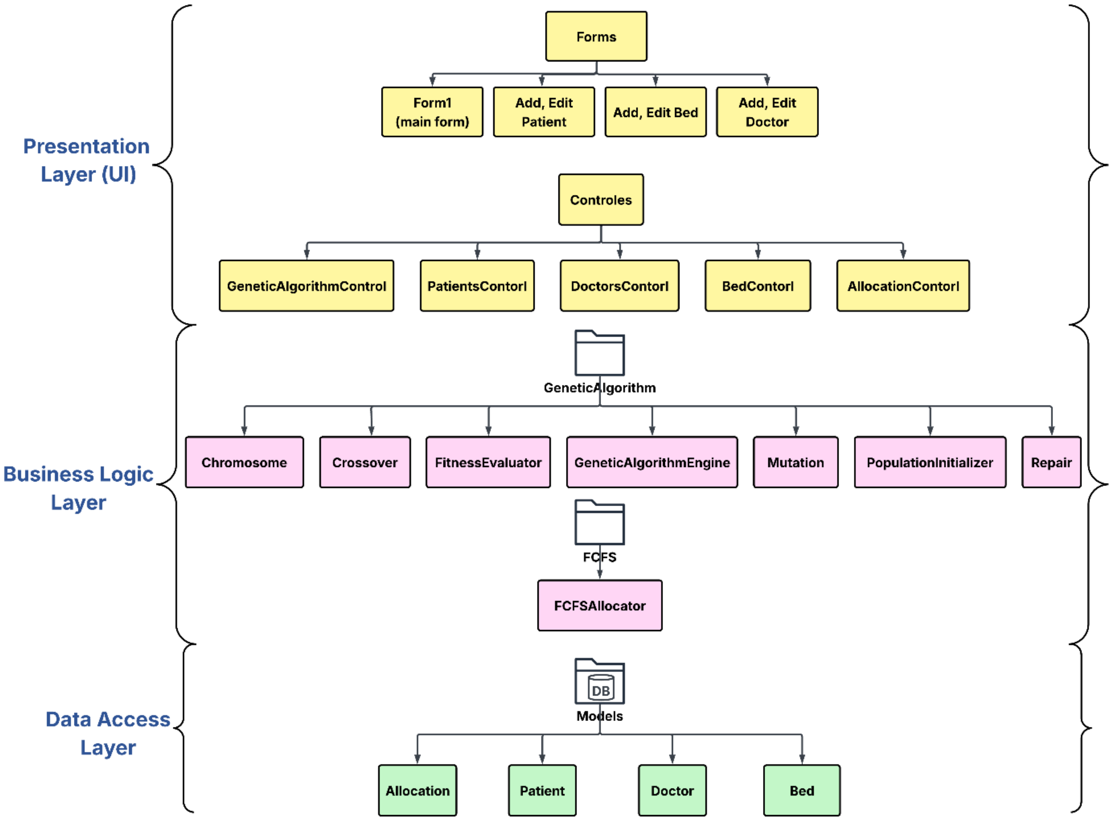

Figure 1 – System Architecture Overview

3.3 Data Representation

Each patient record includes multiple fields: a unique ID, full name, age, time of arrival in the hospital system, treatment priority (a numerical value reflecting urgency), bed duration (estimated time spent occupying a bed), and doctor duration (estimated time required by a doctor for treatment).

Beds and doctors are treated as independent and limited resources. Each bed or doctor can attend to only one patient at a time, and patients are treated in the order in which they are assigned. This reflects real-world constraints in hospital settings where resource conflicts must be avoided.

To ensure simulation accuracy and solution feasibility, a valid allocation must assign every patient exactly once to both a bed and a doctor. Partial assignments or duplicate assignments are considered infeasible and are corrected during the repair phase of the algorithm. The data model enforces these constraints to simulate realistic medical workflows, supporting the calculation of metrics like treated patients, resource utilization, and makespan.

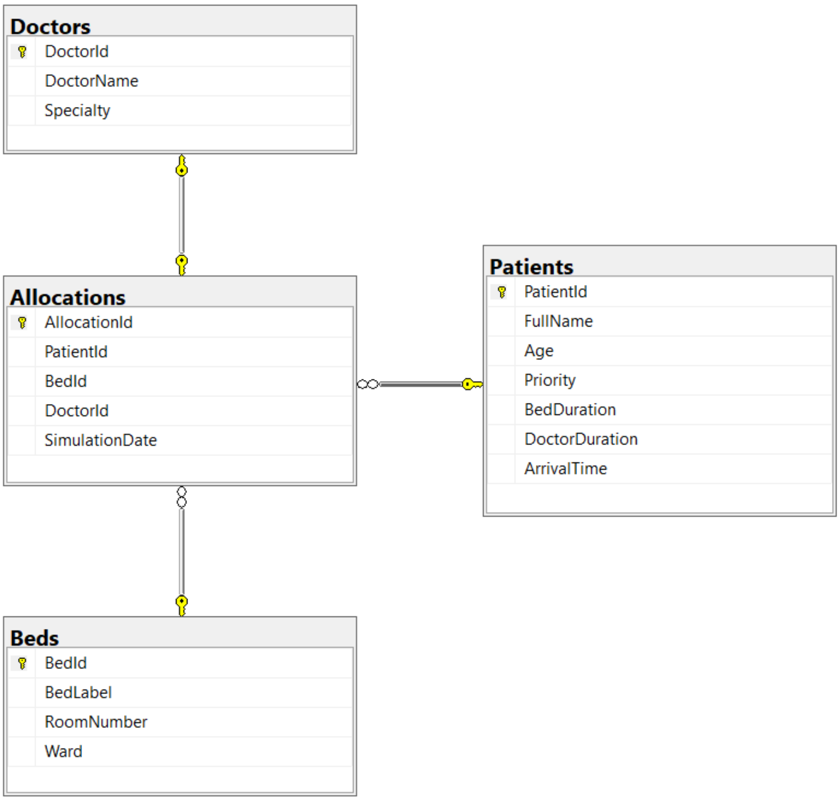

Figure 2 – Database Entity Relationship Diagram

3.4 Chromosome Representation

Each potential solution, or chromosome, encodes a full allocation plan of patients to two types of hospital resources: ICU beds and doctors. The chromosome has a dual-segment structure:

- BedSegments: a list of patient IDs assigned to each bed.
- DoctorSegments: a list of patient IDs assigned to each doctor.

This representation ensures that each patient receives both required resources, and that resource usage and treatment times can be calculated accurately.

At runtime, a chromosome might look like this:

- BedSegments[0] = [2, 5, 7] → Bed 1 is assigned to Patients 2, 5, and 7
- DoctorSegments[0] = [5, 2] → Doctor 1 is treating Patients 5 and 2

This mapping allows the algorithm to simulate how patients are distributed across limited resources and how much time each resource will be occupied.

3.5 Fitness Function

The fitness function evaluates each chromosome based on:

- The number of patients successfully treated (i.e., assigned both a bed and a doctor),
- The total priority of treated patients (higher-priority patients improve the score),
- The overall makespan (total time resources are occupied), which the algorithm aims to minimize.

The goal is to maximize treated priority while minimizing resource overload. The final fitness is normalized between 0 and 1.

$$\text{fitness} = \frac{1}{\text{makespan}} * \text{total\_priority}$$

$$normalized\_fitness \frac{fitness}{max\_possible\_fitness}$$

Where:

- makespan = the maximum time any resource (bed or doctor) is occupied
- total_priority = sum of priorities for all patients successfully treated
- max_possible_fitness = a normalization constant calculated from the ideal scenario

3.6 Genetic Operators

The algorithm follows the classical GA steps:

1. **Initialization**: A population of chromosomes is randomly generated by distributing patient IDs across resources without duplication.
2. **Selection**: Tournament selection is used to choose parent chromosomes based on fitness.
3. **Crossover**: A modified Partially-Mapped Crossover (PMX) combines two parents to produce offspring, applied separately to both segments.
4. **Mutation**: A uniform swap mutation randomly exchanges two patients in a segment, with a configurable mutation rate .
5. **Repair**: A repair function ensures all patients are assigned exactly once to each resource.
6. **Elitism**: The best solution of each generation is carried over to preserve high-quality solutions.

3.7 Parameters

The performance of the genetic algorithm depends significantly on the configuration of its parameters. These parameters influence convergence speed, solution quality, and exploration capability. The following parameters are used in this study:

- Population Size: Determines the number of chromosomes per generation. A higher value increases diversity but also computation time. In the simulations, values between 20 and 200 were tested.
- Number of Generations: Sets how many iterations the algorithm will perform. Higher values allow more time for convergence.
- Mutation Rate: Defines the probability of mutation in each chromosome.
- Tournament Size: Refers to the number of chromosomes selected at random for each tournament.

- Elitism: Ensures the best chromosome from each generation is preserved in the next. This prevents loss of high-quality solutions.
- Crossover Method: PMX was used due to its ability to preserve relative order and segment structure.

Parameter tuning was done manually by testing various combinations and observing the impact on solution quality.

Parameter	Min	Max	Default
Population Size	20	200	50
Number of Generations	50	500	100
Mutation Rate (%)	5%	40%	20%
Tournament Size	2	20	5

Table 1 – Genetic Algorithm Parameters

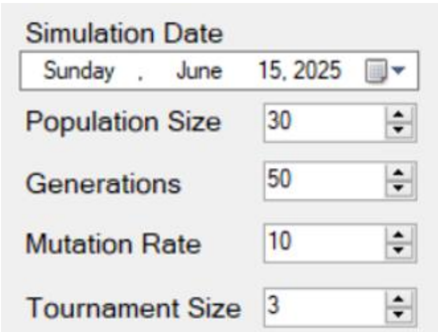

Figure 3 – Genetic Algorithm Parameters Panel

3.8 Simulation Workflow

Before running a simulation, the application loads data from the database (patients, doctors, beds) and user configures parameters such as population size, number of generations, mutation rate, and tournament size.

The algorithm runs for a predefined number of generations, during which fitness scores are tracked and visualized. Upon completion, the best solution is stored in the database and can be reviewed using the interface. The FCFS strategy is also applied as a baseline for performance comparison.

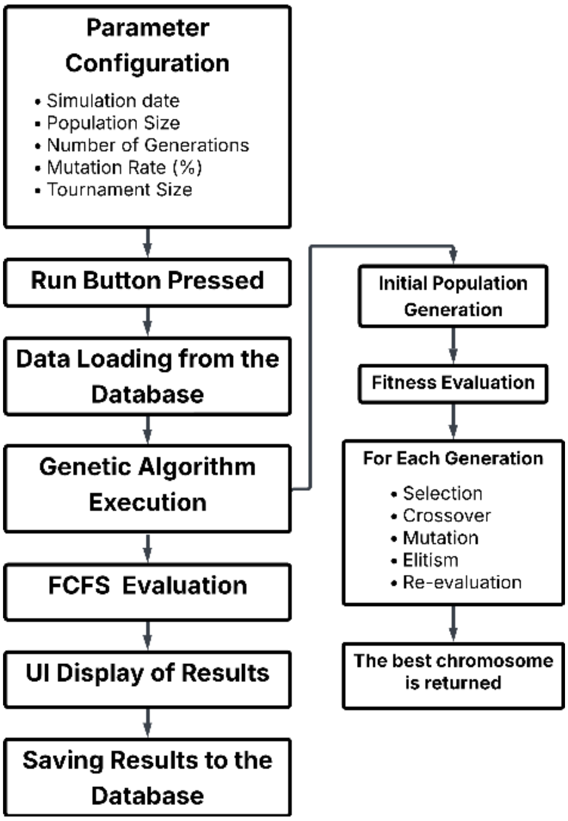

Figure 4 – Execution Flow

3.9 Baseline for Comparison – FCFS

To measure the effectiveness of the genetic algorithm, the application also includes a First-Come-First-Served (FCFS) baseline. This simple method assigns patients to beds and doctors in arrival order and does not perform optimization. Its fitness score is used as a reference point in simulation results.

The table below summarizes the main differences between the Genetic Algorithm and the First-Come-First-Served approach implemented in the application.

Criterion	**Genetic Algorithm (GA)**	**First-Come-First-Served (FCFS)**
Optimization goal	Minimize makespan, maximize treated priority	Assign based on arrival time only
Resource awareness	Balances usage across doctors and beds	Does not optimize resource distribution
Adaptability	Adapts to patient severity and resource duration	Fixed and rigid

Solution quality	Evolves better results over generations	Typically lower efficiency
Conflicts or duplicates	Avoided using repair function	No conflict resolution
Performance indicator	Fitness score (0 to 1), plotted per generation	One-time result, used as a benchmark

Tabel 2 – Comparative summary between GA and FCFS

4. Results and discussions

The effectiveness of the proposed genetic algorithm (GA) was evaluated by comparing it with a baseline First-Come-First-Served (FCFS) approach under various simulation scenarios. The results highlight improvements in patient prioritization, resource utilization, and overall system efficiency.

4.1 Experimental Setup

The dataset used for simulation includes a full week of data, with 40 patients per day, and a predefined number of 7 beds and 5 doctors. Each patient has attributes like arrival time, priority, bed usage duration, and doctor usage duration—these durations are calculated automatically based on a formula that considers patient priority and some randomness. All values are synthetic and were generated to reflect realistic variation in hospital demands.

$$\text{BedDuration} = \text{Priority} \times 30 + \text{random}(0, 30)$$

$$\text{DoctorDuration} = \text{Priority} \times 10 + \text{random}(0, 30)$$

The number of available resources (beds and doctors) can be modified at any time by the user, by adding or removing records directly from the application's interface. This allows running simulations under different stress levels, such as low capacity or patient surges.

Before each simulation, the user selects the parameters that will control the behavior of the genetic algorithm.

This flexibility allows users to experiment with different optimization strategies and evaluate how the algorithm responds.

4.2 Parameter Variation Impact

To analyze how GA parameters affect performance, various test scenarios were executed by adjusting population size, mutation rate, tournament size, and number of generations. All configurations were compared to the static baseline of FCFS (67.73% fitness).

Population Size	Generations	Mutation Rate (%)	Tournament Size	Fitness Max (%)	Fitness Avg (%)	FCFS
30	50	10	3	79.89	78.16	67.73
30	100	10	3	76.81	76.2	67.73
50	50	20	4	79.89	78.53	67.73
50	100	20	4	77.65	76.88	67.73
100	50	30	5	81.32	79	67.73
100	100	30	5	81.32	80.72	67.73

Table 4 – Comparison of Fitness Scores under Different GA Settings

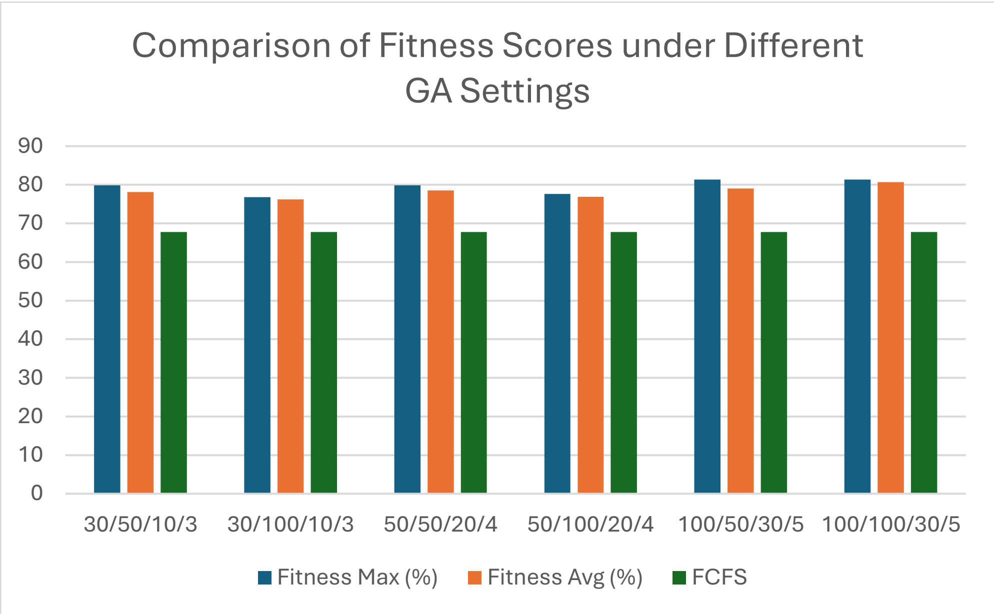

Figure 5 - Fitness Trends for Various Configurations

Table 4 and Figure 5 show that:

- A population of 100 and mutation rate of 30% achieved the highest max fitness (81.32%).
- More generations do not always improve results; best average fitness occurred around 90 generations.
- Moderate tournament size (3–5) provided a balance between selection pressure and diversity.
- The genetic algorithm consistently outperformed FCFS across all tested configurations.

4.3 Tournament Size Impact

Population Size	Generations	Mutation Rate (%)	Tournament Size	Fitness Max (%)	Fitness Avg (%)	FCFS
50	50	20	3	81.45	78.51	67.73
50	50	20	5	78.01	77.07	67.73
50	50	20	7	77.53	76.12	67.73
50	50	20	9	77.17	76.45	67.73

Table 5. Fitness Scores for Different Tournament Sizes

Table 5 analyze the effect of varying tournament size while keeping other parameters constant. A tournament size of 3 resulted in the highest max fitness (81.45%). Larger sizes slightly reduced performance, possibly due to early convergence.

4.4 Mutation Rate Impact

Population Size	Generations	Mutation Rate (%)	Tournament Size	Fitness Max (%)	Fitness Avg (%)	FCFS
50	50	5	3	80.14	78.24	67.73
50	50	10	3	79.63	78.18	67.73
50	50	15	3	79.51	77.7	67.73
50	50	20	3	79.38	76.94	67.73

Table 6 – Fitness Scores for Different Mutation Rates

Table 6 highlight the impact of mutation rates from 5% to 20%. The best fitness was achieved with a 5% rate. Excessive mutation disrupted convergence. A small rate improved exploration without destabilizing solution quality.

4.5 Generations Impact

Population Size	Generations	Mutation Rate (%)	Tournament Size	Fitness Max (%)	Fitness Avg (%)	FCFS
50	50	15	3	79.51	77.7	67.73
50	90	15	3	81.32	80.08	67.73
50	120	15	3	79	78.37	67.73
50	170	15	3	79.25	78.56	67.73

Table 7 – Fitness Evolution with Increasing Generations

As shown in Table 7 increasing generations from 50 to 90 yielded noticeable gains. Beyond that, improvement plateaued or declined slightly. This indicates diminishing returns in prolonged evolution.

4.6 Population Size Impact

Population Size	Generations	Mutation Rate (%)	Tournament Size	Fitness Max (%)	Fitness Avg (%)	FCFS
30	50	15	3	76.22	75.21	67.73
50	50	15	3	79.51	77.7	67.73
70	50	15	3	80.4	77.45	67.73
100	50	15	3	81.32	74.57	67.73

Table 8 – Fitness Evolution under Different Population Sizes

Table 8 summarize the impact of varying population size. A population of 70 offered the best balance, while 100 led to higher maximum fitness but lower average, indicating over-diversity.

In all experiments, GA results were significantly superior to those of FCFS, demonstrating the robustness and adaptability of evolutionary optimization for crisis-driven hospital resource allocation.

4.7 Observations and Insights

The simulation results highlight several important observations:

• The Genetic Algorithm consistently outperforms FCFS in terms of fitness, which reflects better allocation efficiency and treatment success.

• The algorithm adapts well to different resource constraints, even when the number of beds or doctors is limited.

• High-priority patients are more likely to be assigned when using GA compared to FCFS.

• The graphical interface and live fitness feedback allow quick interpretation of results, even for non-technical users.

These findings support the conclusion that evolutionary algorithms like GA can significantly improve hospital planning and decision-making, especially in scenarios with high demand and limited capacity.

5. Conclusions

This study proposed and implemented a simulation framework designed to support decision-making in hospital crisis situations through heuristic resource allocation. By combining a custom-built software system with a genetic algorithm-based optimization strategy, the project successfully demonstrated how intelligent heuristics can enhance operational efficiency in constrained healthcare environments.

The developed Windows Forms application, written in C#, provides a modular interface for loading patient data, configuring simulations, visualizing outcomes, and comparing heuristic optimization against a baseline First-Come-First-Served (FCFS) strategy. The integration with a SQL Server database ensures realistic and reusable scenarios, making the tool suitable for both academic research and operational testing.

Experimental results confirmed that the Genetic Algorithm (GA) significantly outperforms the FCFS method in terms of treated patient priority, resource utilization, and overall system efficiency. The flexible design of the GA engine, including adjustable parameters such as population size, number of generations, mutation rate, and tournament size, allowed extensive exploration of the solution space. The analysis also highlighted how parameter tuning can influence convergence and fitness scores.

Overall, the system proved to be both functional and extensible, with clear potential for real-world impact in crisis management planning.

References

[1] M. Mitchell, "An Introduction to Genetic Algorithms," *MIT Press,* 1998.

[2] N. Jayanthi, S. Baskar and S. Kannan, "Patient Scheduling System for Medical Treatment Using Genetic Algorithm," *International Journal of Computer Applications,* vol. 37, no. 11, pp. 21-28, 2012.

[3] M. Kühn, M. Spies and D. Schwarzbach, "A Multi-objective Optimization Approach Based on Deterministic and Metaheuristic Techniques to Resource Management in Health Crisis Scenarios Under Uncertainty," *Expert Systems with Applications,* vol. 197, no. 116702, 2022.

[4] Y. Chen and Y. Lu, "Medical Staff Scheduling Based on Particle Swarm Optimization under Inter-Hospital Cooperation Mode," *International Journal of Environmental Research and Public Health,* vol. 19, no. 10, 2022.

[5] G. Girimurugan, G. Vinoth and A. Muthukumaran, "A novel framework of dynamic healthcare planning using genetic algorithm and deep learning techniques," *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, p. 4235–4246, 2021.

[6] A. Heiniger, M. Reiss and S. Rachuba, "Workforce scheduling in hospital emergency departments using a genetic algorithm," *Health Care Management Science,* vol. 25, p. 132–147, 2022.

[7] F. Wan, J. Fondrevelle and T. e. a. Wang, "Two-stage multi-objective optimization for ICU bed allocation under multiple sources of uncertainty," *Sci Rep,* vol. 18925, no. 13, 2023.

[8] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, April 2002.

[9] C. Yan, N. McClure, S. P. Dukelow, B. Mann and J. Round, "Optimal Planning of Health Services through Genetic Algorithm and Discrete Event Simulation: A Proposed Model and Its Application to Stroke Rehabilitation Care," *MDM policy & practice,* vol. 7(2), no. 23814683221134098, 2022.

[10] B. Oliveira, J. Vasconcelos, J. Almeida and L. Pinto, "A Simulation-Optimisation approach for hospital beds allocation," *International Journal of Medical Informatics,* vol. 141, no. 104174, 2020.

[11] W. Feng, Z. Lou, N. Kong and H. Wan, "A multiobjective stochastic genetic algorithm for the pareto-optimal prioritization scheme design of real-time," *Operations Research for Health Care,* vol. 15, pp. 32-42, 2017.

[12] S. Farahi and K. Salimifard, "A simulation–optimization approach for measuring emergency department resilience in times of crisis,," *Operations Research for Health Care,* vol. 31, no. 100326, 2021.

[13] M. Nacht, "Operations Research," *International Encyclopedia of the Social & Behavioral Sciences,* pp. 10873-10876, 2001.

[14] G. Krishnan and S. Sowmya Kamath, "A novel GA-ELM model for patient-specific mortality prediction over large-scale lab event data," *Applied Soft Computing,* vol. 80, pp. 525-533, 2019.

[15] X. Pang, Y. Ge, K. Wang, A. Traina and H. Wang, "Patient assignement optimization in cloud healthcare systems: a distributed genetic algorithm," *Health Inf Sci Syst,* vol. 11, no. 30, 2023.

[16] W. Allihaibi, M. Cholette, M. Masoud, J. Burke and A. Karim, "A heuristic approach for scheduling patient treatment in an emergency department based on bed blocking," *International Journal of Industrial Engineering Computations,* vol. 11, pp. 1-20, 2020.

[17] J. S. Sohmer, S. Fridman, D. Peters and e. al, "Winning the Lottery: A Simulation Study Comparing Scarce Resource Allocation Protocols in Crisis Scenarios," *Cureus,* vol. 17, no. 1, 05 January 2025.

[18] T. Ogundare, "Surrogate assisted by genetic algorithm for simulation of a hospital-resource allocation: lockdown policy analysis for COVID-19 pandemic case," *MSc dissertation, University of Witwatersrand,* 2022.

[19] L. Luo, J. Li, X. Xu, W. Shen and L. Xiao, "A Data-Driven Hybrid Three-Stage Framework for Hospital Bed Allocation: A Case Study in a Large Tertiary Hospital in China," *Computational and mathematical methods in medicine,* no. 7370231, 2019.

[20] J. Jiang, D. Sheng, X. Chen, Q. Tian, F. Li and P. Yang, "Data-driven collaborative healthcare resource allocation in pandemics," *Transportation Research Part E: Logistics and Transportation Review,* vol. 192, no. 103828, 2024.

[21] S. Ridler, A. Mason and A. Raith, "A simulation and optimisation package for emergency medical services," *European Journal of Operational Research,* vol. 298, no. 3, pp. 1101-1113, 2022.