**Applying Genetic Algorithm for Patient Appointment Scheduling Optimization**

**Abstract**

The enhancement of patient appointment scheduling represents a significant challenge within healthcare management, with direct implications for patient satisfaction, resource allocation, and overall operational efficiency. This study introduces an innovative methodology that employs Genetic Algorithms (GAs) to improve the scheduling of patient appointments in healthcare facilities. The proposed framework takes into account a variety of constraints, such as appointment length, availability of physicians, and patient preferences, to create an effective scheduling system. By mimicking the processes of natural selection, the GA progressively refines potential solutions, aiming to optimize appointment scheduling while reducing patient wait times and maximizing the use of healthcare resources. The efficacy of the proposed algorithm is assessed through a series of experiments utilizing actual data from a healthcare institution. The results indicate substantial enhancements in scheduling efficiency when compared to conventional approaches, underscoring the potential of GAs as a powerful instrument for tackling intricate scheduling challenges in healthcare environments. The outcomes imply that the adoption of this strategy could improve patient care and operational performance, thereby opening avenues for further exploration and implementation of evolutionary algorithms in healthcare management.

**Keywords:** Appointment System, Patient Unpunctuality, Genetic Algorithm, Patient Waiting Time, Physician Idle Time.

**Highlights**

* By optimizing appointment slots, GAs can minimize patient wait times, leading to improved patient satisfaction.
* Genetic algorithms can effectively allocate resources (such as staff and treatment rooms) based on patient demand patterns, ensuring that healthcare providers are not overburdened or underutilized.
* GAs can help schedule appointments in a way that optimizes patient flow, reducing bottlenecks and ensuring a smooth operation throughout the day. This can lead to shorter turnaround times for procedures and consultations.
* GAs allow for the consideration of multiple objectives simultaneously, such as minimizing patient wait times while maximizing staff efficiency. This holistic approach can lead to better overall performance of the healthcare system.
* Such improvements in patient experience are integral to fostering trust and loyalty, which are vital in the competitive healthcare landscape.
* The application of GAs in healthcare scheduling encourages ongoing research and innovation, leading to the development of new algorithms and techniques that can further improve operational efficiencies.

1. **Introduction**

In the dynamic realm of healthcare, the optimization of patient appointment scheduling has become an essential element in the provision of high-quality care [1]. With healthcare providers grappling with rising patient numbers, constrained resources, and the necessity for individualized care, the urgency for improved scheduling systems is at an all-time high. The optimization of patient appointment scheduling involves a strategic approach to the management and organization of appointments, aimed at maximizing operational efficiency, minimizing wait times, and improving the overall experience for patients [2].

Efficient scheduling plays a crucial role in enhancing the operational effectiveness of healthcare facilities while guaranteeing that patients obtain prompt access to essential services. By utilizing cutting-edge technologies, data analysis, and established best practices, healthcare organizations can reduce scheduling conflicts, optimize resource distribution, and elevate patient satisfaction levels [3]. The optimization process takes into account multiple elements, such as the types of appointments, availability of providers, patient preferences, and trends derived from historical data.

The ongoing digital transformation in healthcare has led to the adoption of advanced scheduling software and algorithms, fundamentally altering conventional appointment systems. This evolution enhances communication between patients and healthcare providers while enabling organizations to respond effectively to varying demands and enhance operational efficiency. Ultimately, the optimization of patient appointment scheduling transcends mere time management; it embodies a patient-centered philosophy that emphasizes accessibility, convenience, and high-quality care within a progressively intricate healthcare landscape [4].

Historically, the process of scheduling appointments has depended on manual methods or rudimentary algorithms that fail to consider the fluid dynamics of healthcare settings. Although these approaches may be adequate for smaller practices or straightforward situations, they often prove inadequate in larger healthcare systems, where the high patient volume and diverse range of services can exceed the capabilities of basic scheduling tools.

Common challenges associated with traditional approaches are as Table1: [5] [6] [7]

**Table1.** Common challenges of appointment scheduling using traditional approaches

|  |  |  |
| --- | --- | --- |
| **Row** | **Challenges** | **Explanations** |
| 1 | Double Bookings | Manual scheduling may result in inaccuracies, such as the assignment of multiple patients to identical time slots, which can create confusion and dissatisfaction for both patients and healthcare providers. |
| 2 | Long Wait Times | Ineffective scheduling practices may lead to bottlenecks, resulting in prolonged wait times for patients awaiting their appointments. This situation can adversely affect both patient satisfaction and the overall outcomes of care. |
| 3 | Underutilization of Resources | Conventional approaches may struggle to adequately align patient demand with the availability of healthcare providers, resulting in instances of both overstaffing and understaffing. This misalignment can lead to resource wastage and diminished operational efficiency. |
| 4 | Inflexibility | Static scheduling systems frequently encounter challenges in responding to unexpected alterations, including cancellations or immediate patient requirements, thereby complicating the ability to address the fluid dynamics inherent in healthcare settings. |
| 5 | Patient Preferences Ignored | Conventional scheduling methods often overlook the specific preferences of patients regarding appointment times, potentially resulting in dissatisfaction and a higher incidence of missed appointments. |
| 6 | Limited Data Utilization | Numerous conventional systems fail to utilize the accessible data regarding patient history, treatment durations, or provider performance, resulting in less than ideal scheduling choices. |
| 7 | Communication Gaps | Manual procedures can lead to insufficient communication between personnel and patients concerning appointment confirmations, reminders, or alterations, thereby heightening the probability of missed appointments. |
| 8 | Administrative Burden | The dependence on manual scheduling methods may result in a heightened administrative burden for personnel, thereby reallocating time and resources that could otherwise be dedicated to patient care. |
| 9 | Inability to Handle Complex Cases | Conventional methods often face challenges when addressing intricate scheduling situations that involve numerous providers or specialties, which can hinder the effective coordination of care. |
| 10 | Lack of Real-Time Updates: | Numerous conventional systems lack the capability to offer real-time insights into appointment availability or modifications, which complicates the dynamic management of schedules for staff. |

There is an increasing acknowledgment of the necessity for more advanced methods of appointment scheduling, which can effectively address the intricacies associated with contemporary healthcare delivery, in light of these constraints.

Healthcare organizations are increasingly adopting optimization techniques that utilize advanced algorithms and computational capabilities to tackle the difficulties related to appointment scheduling. In this context, genetic algorithms (GAs) have surfaced as a notably effective approach, owing to their proficiency in navigating extensive solution spaces and determining optimal or near-optimal scheduling arrangements [8].

Genetic algorithms draw their inspiration from the mechanisms of natural selection and evolutionary theory. They function by emulating the processes of selection, crossover, and mutation, thereby facilitating the evolution of a population of candidate solutions through successive generations. This iterative methodology enables genetic algorithms to traverse intricate optimization landscapes and arrive at solutions that effectively balance various objectives, including the reduction of wait times alongside the enhancement of provider utilization.

The advantages of using genetic algorithms for appointment scheduling optimization are manifold: [9] [10]

1. Adaptability: GAs possess the ability to swiftly respond to fluctuations in patient needs or the availability of healthcare providers, rendering them particularly appropriate for the ever-evolving landscape of healthcare settings.

2. Multi-objective Optimization: Genetic algorithms are capable of addressing multiple objectives concurrently, including patient satisfaction, resource allocation, and operational efficiency, thereby facilitating more comprehensive solutions.

3. Robustness: The exploration of a wide array of potential solutions enables genetic algorithms to avoid the pitfalls of local optima more effectively than conventional optimization techniques.

4. Global Search Capability: Global Search Capability: Genetic Algorithms (GAs) navigate an extensive solution landscape, enabling them to circumvent local optima and identify superior solutions in comparison to conventional optimization techniques. This characteristic is especially advantageous in intricate scheduling situations that involve numerous constraints.

5. Incorporation of Constraints: it allows for the seamless inclusion of various limitations, such as provider specialties, patient preferences, and time constraints, into the optimization framework. This capability ensures that the generated schedules are both practical and achievable.

6. Continuous Improvement: Genetic algorithms exhibit an iterative characteristic that facilitates ongoing enhancement of prior solutions. Through mechanisms such as selection, crossover, and mutation, these algorithms progressively refine appointment schedules, resulting in increasingly optimized outcomes over time.

7. Integration with Other Systems: GAs can be seamlessly incorporated into current healthcare management systems, facilitating a comprehensive scheduling strategy that considers various operational elements.

By leveraging these advantages, genetic algorithms can significantly enhance the efficiency and effectiveness of appointment scheduling in healthcare settings, ultimately leading to improved patient care and operational performance.

1. **Literature review**

In modern healthcare systems, the efficient scheduling of patient appointments is essential for improving operational effectiveness, ensuring high-quality care, and enhancing patient satisfaction [11]. The intricacies of managing patient flow—characterized by diverse patient requirements, the availability of healthcare providers, and the lengths of appointments—often render conventional scheduling approaches inadequate in meeting the growing demands of healthcare settings. As a result, there has been an increase in scholarly investigation into the optimization of patient appointment scheduling through sophisticated computational methods. Among these methods, genetic algorithms (GAs) have gained recognition as a significant approach for tackling the complex issues related to scheduling [12]. This literature review aims to analyze the existing body of research on the use of genetic algorithms in the optimization of patient appointment scheduling, highlighting their methodologies, key findings, and broader implications.

The origins of employing genetic algorithms for optimization challenges can be linked to the foundational research conducted by Holland in 1975, who introduced a heuristic model based on the concepts of natural selection and genetic principles [13]. Genetic algorithms assess a population of possible solutions, progressively refining them towards an optimal outcome through processes such as selection, crossover, and mutation. This evolutionary methodology provides a powerful framework for exploring the extensive solution space associated with patient appointment scheduling. Numerous studies have validated the effectiveness of genetic algorithms in this area, demonstrating their ability to reduce patient waiting times, optimize resource utilization, and improve overall system efficiency.

One significant advancement in this domain is attributed to the research conducted by NT Huynh in 2018 [14]. This study addresses the optimization of patient scheduling within cardiology departments through the application of a hybrid genetic algorithm (GA). This innovative approach effectively amalgamates various optimization strategies to improve scheduling efficiency. Specifically, Huynh's research introduced a hybrid genetic algorithm that integrates genetic algorithms with simulated annealing, specifically designed to meet the operational requirements of hospital management. The findings indicate that this proposed method outperforms traditional scheduling techniques. An empirical investigation has demonstrated the practical applicability of this approach, highlighting its potential to enhance patient flow and resource management in cardiology environments. Nonetheless, it is crucial to recognize that the implementation of such hybrid models may differ based on the specific operational needs and patient requirements of individual cardiology departments. Consequently, further exploration of recent studies on hybrid algorithms in healthcare may prove advantageous.

The integration of machine learning and optimization in addressing the operational challenges of patient-bed assignment necessitates the development of models that proficiently allocate hospital beds, taking into account factors such as patient requirements, bed availability, and resource limitations. Recent research, particularly a significant study by F. Schäfer [15], illustrates methodologies that leverage machine learning techniques to refine decision-making in bed assignments, thereby enhancing both efficiency and patient outcomes. The study utilized machine learning to improve the precision of forecasting emergency patient admissions, incorporating variables such as weather conditions, temporal factors, and local events. Ultimately, the application of machine learning methods surpassed traditional time series forecasting, achieving an improvement of up to 17% in the accuracy of predicting emergency patient admissions.

A forecasting model grounded in genetic programming (GP) was introduced by Liu, X., et al. [16] to accurately predict daily outpatient visits at a primary healthcare facility. This model employs genetic algorithms to scrutinize and forecast trends derived from historical data, thereby improving the precision of daily visit predictions. The findings indicated that the GP-based approach was more adept at extracting intricate information from a limited training sample, resulting in superior performance in forecasting daily outpatient visits.

Squires, M., et al. [17] introduced an innovative genetic algorithm aimed at optimizing the scheduling of medical treatments, with a particular focus on repetitive Transcranial Magnetic Stimulation (rTMS) sessions to improve operational efficiency within healthcare facilities. This algorithm, designated as LSWT-GA, integrates a survivor selection strategy and utilizes heuristic population methods to bolster scheduling performance. Experimental findings indicate that the LSWT-GA algorithm surpasses its counterparts, achieving the optimal makespan more consistently than the List Scheduling Genetic Algorithm (LS-GA) that employs conventional survivor selection methods, as well as a standard genetic algorithm that utilizes random population initialization (Random-GA).

The research conducted by J Jlassi [18] examines the application of a genetic algorithm to tackle the complexities of patient scheduling within emergency departments, with the objective of alleviating problems related to overcrowding and extended waiting periods. This case study illustrates the algorithm's practical implementation in an actual hospital environment, aimed at improving the efficiency of patient processing. The findings indicate notable enhancements when compared to the First Come First Served (FCFS) approach utilized in the real case study, with reductions in patient waiting times varying from 18.84% to 27.45%.

Apornak, A., et al. [19] employed a genetic algorithm (GA) methodology to enhance human resource management within hospital emergency departments. Their study analyzed a dataset spanning 36 months, which encompassed 108 direct healthcare staff members. To ascertain the key factors influencing human resource optimization, they utilized a fuzzy Delphi model. The primary objectives were to reduce annual direct human resource costs while simultaneously maximizing operational efficiency. After conducting 500 generations, the algorithm successfully determined the optimal number of specialists, general practitioners, and nurses required across three shifts. The proposed strategy improved the fitness function by approximately 36% and facilitated the identification of necessary skills and competencies to address potential demands in emergency services.

The examination of the literature indicates that genetic algorithms (GAs) offer distinct benefits, especially in their capacity to navigate extensive solution spaces and achieve optimal or near-optimal scheduling outcomes [20]. In contrast to traditional algorithms, which often fall into local optima because of their deterministic characteristics, genetic algorithms employ stochastic mechanisms that emulate natural selection. This approach facilitates the exploration of intricate solution landscapes. Through an iterative cycle of selection, crossover, and mutation, these algorithms can dynamically enhance appointment schedules, effectively managing the conflicting requirements of patient needs, physician availability, and clinic resources.

Furthermore, the incorporation of genetic algorithms into the scheduling of patient appointments enables healthcare institutions to more effectively manage fluctuating patient volumes and diverse service needs. This adaptability is essential due to the heterogeneous characteristics of healthcare services and the erratic patterns of patient behavior. The application of genetic algorithms can lead to decreased waiting times for patients, increased utilization of clinicians, and improved service provision, ultimately enhancing operational efficiency while also fostering better patient satisfaction and health outcomes.

The empirical findings detailed in this study reinforce the effectiveness of genetic algorithms within this field. Case studies demonstrate that organizations utilizing genetic algorithms for scheduling have experienced significant enhancements in appointment adherence and a reduction in cancellation rates. These results underscore the potential of genetic algorithms to serve as a revolutionary instrument for managing patient flows and refining scheduling practices in healthcare environments.

Future investigations are essential to enhance these algorithms and tailor them to the unique contexts and constraints present in various healthcare settings. Subsequent research should also examine the potential for integrating machine learning methodologies with genetic algorithms, which could lead to more advanced predictive functionalities in appointment scheduling. Furthermore, the assessment of hybrid models that merge genetic algorithms with alternative optimization strategies may produce more effective solutions, especially in intricate multi-resource environments.

In conclusion, as healthcare systems face the simultaneous challenges of rising patient demand and limited resources, the implementation of advanced computational methods, including genetic algorithms for optimizing appointment scheduling, presents a valuable opportunity. The results of this study highlight both the theoretical importance of utilizing genetic algorithms in this domain and their practical ramifications, which could transform scheduling methodologies and improve operational efficiency in healthcare organizations globally. Ongoing research and utilization of these algorithms will be essential to maintain an efficient, patient-focused healthcare delivery system that adapts to the changing requirements of society.

1. **Methodology**
   1. **Schedule outcomes**

The notations used in this chapter are as follows.

* + 1. **Indices and sets**

|  |  |
| --- | --- |
|  | Patients set |
|  | Index for patients, |
|  | Time slots set |
|  | Index for time slots, *s* |
|  | Physicians set |
|  | Index for physicians, *f* |
| *o* | Index of objective functions, |

* + 1. **Parameters**

|  |  |
| --- | --- |
|  | Capacity for time slot *t* |
|  | Binary parameter equal 1 if patient *i* has preference for physician *d* and equal zero if otherwise |
|  | Scheduled arrival time for patient *i* |
|  | Actual arrival time for patient *i* |
|  | Binary parameter equal 1 if patient *i* is available in time slot *t* and equal zero if otherwise |
|  | Service time for patient *i* |
|  | Weight index for objective function *o* |
|  | Lateness grade of patient *i* which equals to 1 if he/she is early, equals to 2 if he/she is on time, and equals to 3 if otherwise. |
|  | Big number |

* + 1. **Decision variables**

|  |  |
| --- | --- |
|  | Binary assignment variable equal 1 if patient *i* is assigned to time slot *t* and physician *d* and equal zero if otherwise |
|  | Binary assignment variable equal 1 if patient *i* is assigned to physician *d* and equal zero if otherwise |
|  | Total patients assigned to physician *d* in time slot *t* |
|  | Total patients assigned to physician *d* |
|  | Binary assignment variable equal 1 if patient *i* is not assigned at all and equal zero if otherwise |
|  | Total patients assigned to time slot *t* to any physician |
|  | Scheduled service start time for patient *i* |
|  | Actual service start time for patient *i* |
|  | Scheduled service end time for patient *i* |
|  | Actual service end for patient *i* |
|  | Average waiting time for patient *i* |
|  | Average idle time for physician *d* |
|  | Average overtime for physician *d* |
|  | Aggregate objective function |

* + 1. **Mathematical formulation**

The primary goal of the objective function (1) is to reduce the weighted average of excessive patient waiting times, resource idle times, resource overtime, and unscheduled patient appointments.

|  |  |
| --- | --- |
|  | (1) |

Constraint (2) determines the total number of patients assigned to a physician within a time slot. Constraint (3) indicates the total number of patients assigned to each physician in the entire scheduling horizon. Constraint (4) calculates the total number of scheduled patients.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |
|  |  | (3) |
|  |  | (4) |

Furthermore, constraint (5) calculates the total number of assigned patients within a time slot for each physician.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Constraint (6) sets the lower limit for the scheduled start time of a patient equal to the predetermined appointment time for that patient. Constraint (7) indicates that the start time of each scheduled patient should be after the visit of the last patient by the same physician.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |
|  |  | (7) |

Constraint (8) specifies the lower limit for the actual start of service for each patient. Constraint (9) indicates that the actual start time of each patient’s visit should be after the visit of the last patient by the same physician.

|  |  |  |
| --- | --- | --- |
|  |  | (8) |
|  |  | (9) |

Constraint (10) and (11) calculate the scheduled and actual end time of service for each patient, respectively.

|  |  |  |
| --- | --- | --- |
|  |  | (10) |
|  |  | (11) |

Constraint (12) calculates the waiting time for each patient. Also, Constraints (13) and (14) indicate the idle time and overtime for each physician created by each patient.

|  |  |  |
| --- | --- | --- |
|  |  | (12) |
|  |  | (13) |
|  |  | (14) |

Constraints (15) and (16) consider the scheduled and actual service start time equal to the scheduled and actual arrival time for the first patient, respectively.

|  |  |  |
| --- | --- | --- |
|  |  | (15) |
|  |  | (16) |

According to constraint (17) assigning a patient to a time slot is only possible if the patient is available in the time slot. Constraint (18) considers the number of assigned patients to each time slot less than or equal to the upper capacity limit of that slot. Also, constraint (19) ensures the assigning of each patient only to the preferred physician. Constraint (20) specifies which physician belongs to each patient. Constraint (21) prevents the scheduling of two late patients in a row for a physician.

|  |  |  |
| --- | --- | --- |
|  |  | (17) |
|  |  | (18) |
|  |  | (19) |
|  |  | (20) |
|  |  | (21) |

Constraints (22) to (24) are used to guarantee the decision variables’ non-negativity and binary limits.

|  |  |  |
| --- | --- | --- |
|  |  | (22) |
|  |  | (23) |
|  |  | (24) |

* 1. **Definition of chromosome**

In GA, an initial population is created by randomly generating many individual solutions. The size of this population is usually determined based on the specific problem’s characteristics and may encompass hundreds or thousands of potential solutions. The traditional approach involves random generation that effectively covers the entire search space of feasible solutions [21].

Defining the chromosome is a crucial part of developing a genetic algorithm. This is because the subsequent search in the solution space using mutation and crossover operators is performed on chromosomes. An inappropriate chromosome definition will lead to the production of unjustifiable or low-quality solutions. As the primary computational axis of scheduling in the current problem is allocating patients to doctors and time blocks, the chromosome will be defined based on the allocation variable. The solution chromosome is considered a three-dimensional matrix. The first dimension represents the patient, the second indicates the time block, and the third represents the doctor’s index. The genes of the chromosome will represent the values of the allocation variable, which are zero and one. Figure 1 shows the general schematic of the chromosome.

A diagram of a cube with numbers and lines

Description automatically generated

**Fig1.** The general structure of the solution chromosome

Each layer of the above chromosome is a two-dimensional matrix that specifies the patients allocated to the doctor related to that layer across different time blocks. For instance, entry (2,2,1) has a value of one, the second patient in the second time block has been allocated to doctor number one. In the figures below, the different layers of the chromosome are shown separately for a better understanding:

A diagram of numbers and symbols

Description automatically generated

**Fig2.** The general form of the chromosome layers

* 1. **Selection process**

During each successive generation, a subset of the current population is chosen to produce the next generation. The selection of individual solutions is carried out using a fitness-based mechanism, where fitter solutions, as evaluated by a fitness function, are generally preferred for selection. There are several selection methods, ranging from methods that rank the fitness of each solution and preferentially select the best ones to other methods that only evaluate a random sample of the population due to potential time constraints [22]. Most selection functions are stochastic and are designed in a way that ensures a small proportion of less-fit solutions are also selected. The purpose of deliberately including less suitable solutions is to maintain population diversity, thereby reducing the risk of premature convergence towards non-optimal solutions. Prominent and extensively studied selection techniques have roulette wheel selection and tournament selection.

* 1. **Generation of a new generation**

After the selection process, the next step is generating a new generation by executing genetic operators, which are (1) Crossover and (2) Mutation [23].

Every newly generated solution is produced from a pair of solutions termed “parents” that have been candidates in the selection phase. Using crossover and mutation mechanisms results in an “offspring” solution, which typically inherits many features from its “parents”. These reproductive strategies based on two parents are inspired by biology, and research indicates that the participation of two “parents” leads to the production of higher-quality chromosomes.

As a result of these processes, the population of the next generation of chromosomes differs from the initial generation. Generally, the average fitness of the new population increases due to the selection of the best individuals or chromosomes from the previous generation. At the same time, a small percentage of suboptimal solutions are included to maintain diversity.

* 1. **Genetic algorithm operator**

After determining the initial population, to produce new solutions and move on to the next generation, it is necessary to extract new solutions from the current ones using the algorithm’s operators. For this purpose, the genetic algorithm employs two operators: mutation and crossover. The implementation of these two operators varies depending on the mathematical model and should be carried out in a way that covers as much of the solution space as possible [24]. Additionally, a mechanism is needed to prevent the generation of infeasible solutions after applying the algorithm’s operators. The following will detail each of these operators.

* + 1. **Crossover operator**



**Fig3.** The mechanism of the Crossover operator

In the figure above, after applying a single-point crossover, the allocation in the first- and second-time blocks for the first offspring is based on the first parent, and the allocation for the remaining time blocks is based on the second parent. Similarly, for the second offspring, the allocation in the first two-time blocks is based on the second parent, and the allocation for the remaining time blocks is based on the first parent. In this way, by inheriting from both parents, two offspring with new solutions have been generated.

* + 1. **Mutation operator**

Unlike the crossover operator, which requires two selected chromosomes as parents for execution, the mutation operator works solely on a single chromosome [25]. The purpose is to diversify the solution space and prevent getting trapped in local optima. Various approaches can be adopted to apply the mutation operator, aiming to introduce diversity into the population and produce new solutions. The chosen approach will vary depending on the physical representation of the chromosome and the problem assumptions. Accordingly, in this study, two approaches fitting the solution space of the problem have been adopted and implemented. Also, the probability of selecting each approach in each operator’s execution is considered equal to the other approach.

Approach 1: The first approach is selecting the patients assigned to a doctor and swapping the designated times between them. Figure 5.4 provides a schematic representation of how this mutation is implemented.



**Fig4.** The implementation the mutation operator (Approach 1)

Approach 2: In the second approach, the mutation operator is based on exchanging the allocation between an assigned patient and an unscheduled one. Figure 5.5 schematically illustrates the implementation of this mutation.



**Fig5.** The implementation of mutation operator (Approach 2)

* 1. **Repair mechanism**

With the application of crossover and mutation operators, there is a possibility that some of the generated solutions might become infeasible. For instance, the crossover operator in the figure illustrates this scenario.



**Fig6.** An example of the repair mechanism

As observed in the figure above, after applying the crossover operator, patient 5 in the first offspring chromosome is simultaneously allocated to two-time blocks, 2 and 4, which is impossible. Also, concerning the mutation operator, if the two swapped patients have different arrival times and one arrives at the hospital late, there might be problems in the new order due to the presence of previous patients with significant delays. This can lead to the generation of infeasible solutions. Therefore, there must be a mechanism in place to rectify or prevent the introduction of such solutions in case any of the above scenarios occur.

To fix it, if there is a problem and it is possible to be solved, a new chromosome is generated. Otherwise, the objective function for that solution is set to positive infinity to exclude it from the solution set.

* 1. **Developing of the Genetic Algorithm**

This section will present the customized development of the GA for our issue domain, building on the fundamental components described in the previous sections. The proposed technique adds custom modifications and extensions to the standard GA structures and mechanisms to address the difficulties of patient-doctor-time block scheduling.

Understanding the iterative nature of GA and the significance of fitness-based selection is essential to creating efficient GA. All generations share the same population size. Every chromosome is assessed, and based on how well it performs the fitness function, it is probabilistically chosen for the following generation, favoring those with a higher fitness. Then, new offspring are created by the reproduction of these selected chromosomes. This evolutionary cycle continues until a predetermined termination condition is met (Wonjae Lee & Hak-Young Kim, 2005). The GA used in this study is presented in the below pseudocode.

**Procedure:** Genetic Algorithm

Get all parameters and save them into par

Determine all genetic algorithm hyper parameters

Generate initial population

**For** it = 1 to Max algorithm iteration

Generate a random integer number from 1 to 3

**If** rand = 1

Use Roulette-Wheel Selection method

**Else if** rand = 2

Use Tournament Selection method

**Else**

Use Rand Selection method

**End if**

**For** k = 1 to nc (number of cross population)

Select two chromosomes based on the chosen method

Apply crossover operator on chromosomes and generate child chromosomes

Update new chromosome features (assign table and cost function)

Save the new generated population

**End for**

**For** k = 1 to nm (number of mutation population)

Select one random chromosome

Apply mutation operator on the selected chromosome

Save the new generated population

**End for**

Merge existing population and newly generated population

Sort the cumulative population based on the cost

Updated best and worst cost

**End for**

**Fig7.** Pseudocode of Genetic Algorithm

1. **Finding and results**
   1. **Sensitivity analysis**

In this part, our objective is to determine the ideal patient-to-physician ratio across varying unpunctuality rates, ranging from 30% to 90% with 10% increments. It has been evaluated by considering three different patient counts (60, 70, 80) and three distinct physician numbers (2, 3, 4). Through sensitivity analysis for each scenario, we aim to identify the near optimal patient-to-physician ratio with the lowest average total cost.

* + 1. **Analysis of the physician idle time**

Upon analyzing the patient-to-physician ratios across various unpunctuality rates (see Appendix A), a distinct pattern emerges. The 80:2 (or 40:1) ratio consistently proves near optimal for four of the seven assessed unpunctuality rates, emphasizing its efficiency in minimizing physician idle time across various scenarios. However, specific unpunctuality rates prompt variations: at 40%, the 70:3 (or 23.3:1) ratio is favored, while an 80% rate sees the 60:3 (or 20:1) ratio as most effective. These shifts suggest that at certain unpunctuality thresholds, a higher physician count relative to patients can be beneficial. Notably, an anomaly arises at the 60% unpunctuality rate, where the 80:4 (or 20:1) ratio is optimal, paralleling the ratio observed at the 80% rate, hinting at a non-linear relationship between unpunctuality and ideal staffing ratios. Figure 8 displays the average idle time of physicians before and after applying the near optimal patient-to-physician ratio at a 70% rate of unpunctuality.

A graph of schedule rules

Description automatically generated with medium confidence

**Fig8.** The bar chart of the average physician idle time before and after the near optimal ratio

* + 1. **Analysis of the physician overtime**

In assessing the relationship between unpunctuality rates and the ideal patient-to-physician ratios, specific patterns emerge that provide insights into near optimal staffing strategies (see Appendix B). Predominantly, the 80:4 (or 20:1) patient-to-physician ratio is favored, proving especially effective at unpunctuality rates of 40%, 70%, and 80%. Conversely, the 60:4 (or 15:1) ratio demonstrates its efficacy at 50%, 60%, and notably at the high 90% unpunctuality rate. An outlier in the data is observed at the 30% unpunctuality level, where a 70:4 ratio is near optimal, suggesting that lower unpunctuality rates might require a distinctive staffing approach. These findings underscore the need for dynamic staffing models that can adapt to varying levels of unpunctuality, ensuring patient needs and physician efficiency are met. Figure 9 displays the average overtime of physicians before and after applying the near optimal patient-to-physician ratio at a 70% rate of unpunctuality.

A graph of schedule rules

Description automatically generated with medium confidence

**Fig9.** The bar chart of the average physician overtime before and after the near optimal ratio

* + 1. **Analysis of the patient waiting time**

Across all unpunctuality rates, the 70:4 (or 17.5:1) patient-to-physician ratio is predominantly near optimal (see Appendix C). This suggests that, regardless of the patient punctuality frequency, the most efficient strategy to reduce patient waiting times is to assign one physician for every 17 to 18 patients. Figure 10 displays the average patient waiting time before and after applying the near optimal patient-to-physician ratio at a 70% rate of unpunctuality.

A graph of schedule rules

Description automatically generated with medium confidence

**Fig10.** The bar chart of the average patient waiting time before and after the near optimal ratio

* + 1. **Analysis of the unscheduled patients**

Regardless of variations in punctuality rates ranging from 30% to 90%, a 60:4 (equivalent to 15:1) patient-to-physician ratio is consistently identified as near optimal (see Appendix D). This indicates that an arrangement of 1 physician for every 15 patients is the most effective way to minimize unscheduled patient occurrences, irrespective of the frequency of patient punctuality. Figure 11 displays the average patient waiting time before and after applying the near optimal patient-to-physician ratio at a 70% rate of unpunctuality.

A graph of a number of columns

Description automatically generated with medium confidence

**Fig11.** The bar chart of the average number of unscheduled patients before and after the near optimal ratio

* 1. **Analysis of the near optimal ratio of patients to physicians**

This section will evaluate if the determined ideal patient-to-physician ratios reduce the average cost associated with each scheduling rule when the unpunctuality rate is 70%.

* + 1. **Analysis of the physician idle time**

Table1 shows the analysis of the average total cost associated with the physician's idle time before and after the near optimal ratio of patients to physicians. According to the results, the average total cost decreases significantly after applying the near optimal ratio (P-value=0). This emphasizes how important it is to maximize the patient-to-physician ratio in medical environments. The decrease in expenses not only represents a more effective use of doctors’ time but also suggests significant cost reductions for healthcare organizations.

**Table1.** The impact of the near optimal ratio of patients to physicians on costs (PI scenario)

A table with numbers and a number on it

Description automatically generated

* + 1. **Analysis of the physician overtime**

Table2 illustrates the analysis of the average total cost of physician overtime before and after the ideal patient-to-physician ratio. Implementing the near optimal patient-to-physician ratio of 80:4 led to notable reductions in the average total cost of physician overtime across all scheduling rules. In most cases, costs were nearly halved. This underscores the value of optimizing patient-to-physician ratios in healthcare settings. The reduction in costs not only signifies more efficient utilization of physician time but also indicates potential savings for healthcare institutions. The varying degrees of cost reduction across different scheduling rules emphasize the need for tailored approaches, understanding that while overarching strategies can be effective, individual rules may have specific nuances that need addressing.

**Table2.** The impact of the near optimal ratio of patients to physicians on costs (PO scenario)

A table with numbers and a number

Description automatically generated

* + 1. **Analysis of the patient waiting time**

The analysis of the average total cost related to patient waiting times before and after the ideal patient-to-physician ratio is shown in Table3 The average total cost of patient waiting times across all analyzed scheduling rules was significantly reduced when the best patient-to-physician ratio of 70:4 was used. This emphasizes how crucial it is to manage patient-to-physician ratios in healthcare environments. Such a considerable decrease not only improves patient satisfaction by decreasing wait times but also results in significant financial savings for healthcare organizations. Because different scheduling rules have diverse effects, it is essential to take a nuanced approach and consider the potential for unique dynamics for each rule.

**Table3.** The impact of the near optimal ratio of patients to physicians on costs (WT scenario)

A table with numbers and a number

Description automatically generated

* + 1. **Analysis of the unscheduled patients**

A comparison of the average total cost of unscheduled patients before and after identifying the near optimal ratio of patient-to-physician is shown in Table4. All assessed scheduling rules experienced notable decreases in the average total cost of unscheduled patients when the near optimal patient-to-physician ratio of 60:4 was implemented. This emphasizes how crucial it is to modify the patient-to-physician ratios in healthcare settings to manage interruptions and lower associated expenses efficiently. The significant cost savings across all regulations emphasize the potential for enhanced clinical operations by reducing physician idle periods and the financial advantages for healthcare organizations.

**Table4.** The impact of the near optimal ratio of patients to physicians on costs (NP scenario)

A table with numbers and text

Description automatically generated

* 1. **Managerial Implications**

We propose the following managerial ideas based on our findings.

1. For the PI scenario, managers might consider the 80:2 ratio as a baseline for scheduling. This could serve as a starting point, which can be adjusted based on specific circumstances.

2. For the PO scenario, managers might consider the 80:4 ratios as baseline ratio for scheduling. These can serve as a starting point, and adjustments can be made based on anticipated unpunctuality.

3. For days when extremely high unpunctuality is anticipated, it might be beneficial to adjust the staffing ratio to something closer to 60:3 to manage physician overtime.

4. For the WT scenario, the ratio of 70:4 is the near optimal assignment of patients to physicians, and clinic managers might adopt this as their standard scheduling practice. This would simplify the scheduling process while ensuring minimized patient waiting times.

5. For the NOAPP scenario, 60:4 is the near optimal ratio of physicians to patients through all unpunctuality rates. This ratio offers a reliable framework likely to minimize disruptions caused by unscheduled patients.

6. While the 60:4 ratio manages unscheduled patients effectively; managers should still be prepared with protocols for handling unscheduled patients. This might include setting aside specific time slots for walk-ins or having a quick triage system to assess and prioritize unscheduled patients based on urgency.

7. Given the dynamic nature of patient attendance, it is crucial to monitor unpunctuality rates and adjust scheduling accordingly continuously. Periodic reviews can help in refining the optimal ratios.

8. While the near optimal ratios have broadly proven beneficial, the varied impact across different scheduling rules suggests that some rules might need further tailoring or adjustments to maximize efficiency.

1. **Conclusions**

The sensitivity analysis results yield a comprehensive understanding of the varying patient-to-physician ratios’ impact in diverse scenarios, especially within the range of unpunctuality rates. The consistent effectiveness of the 80:2 ratio in most contexts signifies its potential as a foundational guideline for healthcare scheduling. Nevertheless, unique unpunctuality rates favor varying ratios, underscoring the non-linear relationship between unpunctuality and optimal ratios.

Delving deeper, the notable cost reductions observed in physician overtime and unscheduled patient instances underscore the profound financial advantages of optimal staffing strategies in healthcare. These savings are pivotal for healthcare institutions, often grappling with constrained budgets. Moreover, beyond financial ramifications, the patient experience emerges as a salient theme. By substantially curtailing patient waiting times, healthcare providers can not only bolster operational efficiency but also significantly enhance patient satisfaction. Such improvements in patient experience are integral to fostering trust and loyalty, which are vital in the competitive healthcare landscape.

However, the results also bring to the fore the intrinsic intricacies of healthcare scheduling. While generalized strategies might be broadly effective, the distinct nuances of specific scenarios necessitate more tailored approaches. As observed, the impact of a chosen ratio can be contingent on the scheduling rule deployed, highlighting the need for adaptable and context-aware scheduling solutions.

An intriguing direction for future inquiry is the anomalies discerned, especially at the 60% unpunctuality rate. Deciphering the underpinnings of these anomalies can shed light on deeper intricacies in healthcare scheduling. Additionally, the potential non-linear dynamics between unpunctuality and optimal staffing ratios beckon further exploration, which could unearth novel insights and strategies.

**Declaration of Interest Statement**

**Funding:** this research is not supported by any organization.

**Conflict of interest / Competing interests:** The authors have no conflicts of interest to declare that are relevant to the content of this article.

* **Funding**: This study was not funded by any University.
* **Employment**: Any organization or employment won't gain or loss financially through publication of this manuscript.
* **Financial interests**: The authors declare they have no financial interests

**Ethics approval:** Not applicable

**Consent to participate:** Not applicable

**Consent for publication:** Not applicable

**Availability of data and material:** here are some resources and types of data/materials you might find helpful:

* Academic Journals and Research Papers
* Books
* Online databases
* Government and Healthcare Organization Reports

All of them are mentioned in the manuscript.

**Code availability:** All of the necessary pseudo-codes are mentioned in the manuscript.

**Author's contribution:** The idea, computations and methodology, proofreading and edition belongs to **A**. Introduction and literature review are written by **M**. **H** given some technical recommendation about methodology, computations, etc.

### **Appendix A**

The near optimal ratio of patients to physicians under the PI scenario for different unpunctuality rates.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scheduling Rules | Robustness (Unpunctuality Rate = 30%) | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 18.48 | 65.82 | 96.24 | 31.60 | 65.00 | 148.18 | 17.93 | 35.99 | 79.28 |
| 2ATEND | 4.86 | 113.40 | 148.80 | 44.99 | 89.77 | 156.56 | 27.22 | 73.15 | 124.45 |
| SR-3ATBEG | 11.08 | 95.22 | 177.87 | 21.77 | 130.42 | 171.89 | 22.83 | 40.11 | 151.41 |
| SR-4ATBEG | 31.59 | 138.48 | 192.59 | 27.66 | 127.07 | 135.01 | 16.30 | 82.65 | 109.97 |
| SL-3ATEND | 19.18 | 105.42 | 138.28 | 50.41 | 103.65 | 132.00 | 9.26 | 56.62 | 106.39 |
| SL-4ATEND | 50.26 | 114.44 | 125.30 | 46.55 | 106.50 | 105.13 | 15.02 | 120.94 | 127.95 |
| REPDOME-3 | 43.99 | 125.88 | 160.01 | 28.03 | 104.45 | 109.19 | 17.17 | 74.16 | 126.50 |
| REPDOME-4 | 33.04 | 125.05 | 156.73 | 57.19 | 94.68 | 135.12 | 41.73 | 88.70 | 147.56 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 40%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 27.75 | 126.14 | 142.95 | 32.03 | 92.11 | 109.13 | 28.89 | 67.54 | 108.42 |
| 2ATEND | 24.66 | 131.91 | 131.22 | 39.59 | 99.91 | 144.89 | 7.55 | 17.17 | 91.61 |
| SR-3ATBEG | 33.10 | 97.99 | 168.40 | 33.95 | 103.35 | 113.39 | 10.99 | 49.59 | 143.60 |
| SR-4ATBEG | 84.42 | 133.93 | 142.41 | 36.52 | 108.28 | 96.59 | 25.03 | 86.78 | 134.58 |
| SL-3ATEND | 56.43 | 102.96 | 175.05 | 26.93 | 100.23 | 151.95 | 26.16 | 40.92 | 136.60 |
| SL-4ATEND | 56.28 | 161.42 | 135.13 | 47.02 | 92.18 | 144.26 | 2.16 | 112.33 | 110.15 |
| REPDOME-3 | 108.98 | 127.79 | 185.47 | 70.49 | 72.44 | 142.63 | 35.28 | 68.13 | 108.08 |
| REPDOME-4 | 59.39 | 113.79 | 119.62 | 37.44 | 85.41 | 136.34 | 65.55 | 74.29 | 84.97 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 50%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 36.82 | 135.91 | 182.65 | 18.87 | 76.80 | 140.62 | 27.91 | 34.92 | 147.45 |
| 2ATEND | 16.86 | 121.72 | 138.99 | 10.41 | 87.24 | 116.30 | 26.45 | 38.84 | 85.89 |
| SR-3ATBEG | 17.40 | 104.72 | 177.83 | 30.85 | 57.68 | 133.63 | 24.11 | 71.16 | 72.49 |
| SR-4ATBEG | 69.01 | 140.10 | 161.43 | 32.32 | 121.77 | 176.70 | 13.23 | 97.19 | 116.28 |
| SL-3ATEND | 36.07 | 172.13 | 202.71 | 20.00 | 100.59 | 91.95 | 35.32 | 74.15 | 87.51 |
| SL-4ATEND | 93.82 | 153.04 | 140.38 | 49.09 | 106.79 | 182.05 | 21.42 | 65.14 | 90.68 |
| REPDOME-3 | 69.83 | 137.58 | 180.08 | 70.73 | 131.58 | 163.44 | 13.18 | 82.64 | 85.63 |
| REPDOME-4 | 28.76 | 143.87 | 141.78 | 33.26 | 96.88 | 156.08 | 45.34 | 55.52 | 147.72 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 60%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 48.06 | 152.94 | 171.82 | 13.81 | 75.66 | 86.28 | 20.76 | 25.40 | 94.97 |
| 2ATEND | 44.86 | 106.85 | 137.76 | 12.63 | 101.29 | 141.99 | 13.15 | 17.88 | 118.81 |
| SR-3ATBEG | 49.66 | 105.06 | 140.15 | 20.43 | 145.28 | 107.33 | 9.17 | 39.63 | 108.69 |
| SR-4ATBEG | 39.52 | 131.25 | 177.38 | 36.22 | 92.81 | 141.35 | 21.95 | 82.23 | 114.77 |
| SL-3ATEND | 13.54 | 130.19 | 191.84 | 24.33 | 114.22 | 129.15 | 10.54 | 48.85 | 128.37 |
| SL-4ATEND | 64.13 | 134.81 | 144.24 | 89.26 | 101.83 | 155.39 | 12.79 | 53.47 | 107.90 |
| REPDOME-3 | 93.30 | 109.66 | 196.15 | 59.44 | 71.80 | 153.42 | 25.69 | 83.66 | 126.86 |
| REPDOME-4 | 58.09 | 109.72 | 159.92 | 101.34 | 131.77 | 86.28 | 56.26 | 92.53 | 95.06 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 70%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 12.06 | 116.58 | 156.95 | 5.46 | 64.2 | 126.12 | 13.2 | 39.78 | 119.22 |
| 2ATEND | 37 | 105.72 | 181.25 | 13.8 | 53.7 | 134.11 | 9 | 49.5 | 146 |
| SR-3ATBEG | 22 | 103.86 | 192.32 | 16.8 | 84.06 | 125.43 | 1.8 | 55.74 | 134.22 |
| SR-4ATBEG | 9.42 | 144.18 | 156.93 | 14.4 | 94.32 | 106.49 | 2.28 | 84.3 | 142 |
| SL-3ATEND | 26.28 | 129 | 172.06 | 22.8 | 80.88 | 143.44 | 4.8 | 42.72 | 105.18 |
| SL-4ATEND | 32.4 | 150 | 174.25 | 4.38 | 90 | 146.52 | 10.02 | 73.32 | 151.74 |
| REPDOME-3 | 63.42 | 111.42 | 157.63 | 22.8 | 90 | 108.14 | 33.6 | 79.38 | 126.84 |
| REPDOME-4 | 88.2 | 94.44 | 132.47 | 50.4 | 96 | 148.01 | 39.18 | 70.68 | 109.32 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 80%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 52.78 | 83.38 | 125.99 | 61.00 | 60.62 | 82.93 | 24.34 | 48.66 | 151.13 |
| 2ATEND | 39.56 | 127.68 | 149.52 | 15.12 | 60.50 | 149.15 | 25.60 | 85.45 | 136.53 |
| SR-3ATBEG | 35.73 | 104.84 | 156.45 | 28.67 | 133.48 | 120.03 | 12.13 | 38.19 | 123.15 |
| SR-4ATBEG | 61.51 | 104.45 | 91.77 | 14.30 | 70.33 | 120.37 | 20.83 | 21.12 | 121.92 |
| SL-3ATEND | 49.91 | 88.87 | 141.51 | 57.93 | 77.97 | 132.54 | 9.69 | 41.71 | 118.78 |
| SL-4ATEND | 93.72 | 125.66 | 130.15 | 40.44 | 96.55 | 133.12 | 46.98 | 91.65 | 127.82 |
| REPDOME-3 | 89.08 | 126.18 | 139.14 | 40.39 | 93.81 | 124.13 | 36.85 | 97.06 | 99.75 |
| REPDOME-4 | 91.45 | 84.91 | 121.34 | 65.06 | 96.34 | 82.93 | 66.31 | 82.26 | 121.65 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 90%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 73.54 | 104.34 | 99.48 | 27.62 | 69.32 | 137.04 | 8.47 | 68.48 | 97.96 |
| 2ATEND | 63.39 | 165.19 | 215.42 | 12.03 | 75.66 | 106.15 | 22.70 | 61.50 | 98.37 |
| SR-3ATBEG | 40.69 | 138.13 | 152.72 | 20.20 | 98.07 | 155.55 | 11.33 | 43.02 | 92.48 |
| SR-4ATBEG | 78.46 | 75.25 | 115.23 | 18.58 | 133.71 | 123.45 | 22.31 | 61.60 | 119.72 |
| SL-3ATEND | 65.09 | 124.45 | 108.47 | 32.37 | 70.10 | 159.19 | 5.16 | 69.56 | 105.49 |
| SL-4ATEND | 40.26 | 77.85 | 186.87 | 43.56 | 87.10 | 115.90 | 16.08 | 114.11 | 122.99 |
| REPDOME-3 | 87.02 | 147.65 | 170.21 | 35.67 | 107.65 | 127.30 | 38.34 | 73.22 | 132.26 |
| REPDOME-4 | 84.25 | 126.97 | 156.20 | 65.21 | 117.36 | 137.04 | 28.06 | 79.17 | 135.20 |

### **Appendix B**

The near optimal ratio of patients to physicians under the PO scenario for different unpunctuality rates.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scheduling Rules | Robustness (Unpunctuality Rate = 30%) | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 33.99 | 28.00 | 18.00 | 46.25 | 23.17 | 20.13 | 53.96 | 32.67 | 19.00 |
| 2ATEND | 25.44 | 29.34 | 12.30 | 36.25 | 20.33 | 15.25 | 50.11 | 31.67 | 21.00 |
| SR-3ATBEG | 22.04 | 27.33 | 14.38 | 40.25 | 19.67 | 18.13 | 64.17 | 30.67 | 27.75 |
| SR-4ATBEG | 42.96 | 24.00 | 19.38 | 34.50 | 38.67 | 21.00 | 54.65 | 36.33 | 15.50 |
| SL-3ATEND | 26.09 | 28.00 | 19.13 | 32.50 | 27.17 | 19.38 | 46.38 | 26.33 | 19.75 |
| SL-4ATEND | 50.29 | 22.83 | 17.50 | 43.75 | 37.00 | 18.00 | 39.76 | 27.83 | 22.38 |
| REPDOME-3 | 35.50 | 19.67 | 9.50 | 30.00 | 26.33 | 17.75 | 65.34 | 24.50 | 18.75 |
| REPDOME-4 | 36.86 | 22.83 | 20.75 | 43.75 | 27.83 | 14.38 | 75.11 | 25.50 | 20.25 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 40%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 53.62 | 25.17 | 21.63 | 49.25 | 20.67 | 17.13 | 62.95 | 27.83 | 31.25 |
| 2ATEND | 46.08 | 31.33 | 13.38 | 47.50 | 27.83 | 18.38 | 48.27 | 21.67 | 28.00 |
| SR-3ATBEG | 43.80 | 26.00 | 24.38 | 45.00 | 37.17 | 29.50 | 52.00 | 30.83 | 18.75 |
| SR-4ATBEG | 68.21 | 22.50 | 12.50 | 59.50 | 30.17 | 23.13 | 49.51 | 32.83 | 24.38 |
| SL-3ATEND | 69.96 | 19.00 | 25.50 | 48.75 | 30.17 | 25.50 | 76.58 | 29.00 | 27.50 |
| SL-4ATEND | 67.39 | 29.67 | 21.75 | 33.25 | 23.67 | 29.13 | 81.58 | 32.67 | 20.75 |
| REPDOME-3 | 85.99 | 23.50 | 12.88 | 50.75 | 27.67 | 25.13 | 70.39 | 35.17 | 26.75 |
| REPDOME-4 | 65.44 | 11.83 | 19.50 | 49.25 | 32.33 | 14.75 | 79.02 | 34.33 | 30.63 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 50%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 77.66 | 22.33 | 21.75 | 49.25 | 34.83 | 23.00 | 57.45 | 33.83 | 31.50 |
| 2ATEND | 48.18 | 21.33 | 22.13 | 59.25 | 36.83 | 29.13 | 61.48 | 46.50 | 26.88 |
| SR-3ATBEG | 46.20 | 35.83 | 17.88 | 35.25 | 39.00 | 28.00 | 68.55 | 30.67 | 32.25 |
| SR-4ATBEG | 60.01 | 25.17 | 19.13 | 43.00 | 36.67 | 28.00 | 43.11 | 40.00 | 35.13 |
| SL-3ATEND | 55.29 | 18.17 | 13.50 | 43.75 | 40.67 | 20.00 | 75.41 | 42.83 | 31.00 |
| SL-4ATEND | 95.41 | 17.50 | 18.00 | 52.50 | 27.83 | 22.12 | 61.96 | 36.67 | 25.88 |
| REPDOME-3 | 80.16 | 30.67 | 17.25 | 52.75 | 37.67 | 31.13 | 63.84 | 38.00 | 26.63 |
| REPDOME-4 | 58.63 | 32.17 | 14.63 | 34.00 | 25.17 | 18.88 | 94.67 | 44.17 | 35.25 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 60%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 79.28 | 21.33 | 25.50 | 60.25 | 34.00 | 29.50 | 87.63 | 40.33 | 22.88 |
| 2ATEND | 58.93 | 28.00 | 23.38 | 53.00 | 31.83 | 32.75 | 53.82 | 33.83 | 30.50 |
| SR-3ATBEG | 81.08 | 34.00 | 19.63 | 39.00 | 35.33 | 27.88 | 56.09 | 36.33 | 35.13 |
| SR-4ATBEG | 70.01 | 31.50 | 19.25 | 43.75 | 38.67 | 26.00 | 84.22 | 52.50 | 28.63 |
| SL-3ATEND | 44.02 | 31.00 | 29.88 | 42.00 | 25.17 | 28.63 | 73.02 | 46.17 | 25.88 |
| SL-4ATEND | 76.32 | 36.33 | 24.38 | 55.00 | 41.00 | 29.25 | 65.64 | 33.17 | 27.88 |
| REPDOME-3 | 97.90 | 33.33 | 24.50 | 45.75 | 27.17 | 36.75 | 70.59 | 27.33 | 27.75 |
| REPDOME-4 | 58.09 | 23.33 | 29.38 | 56.00 | 37.17 | 23.25 | 91.88 | 60.50 | 34.25 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 70%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 49.53 | 32.31 | 30.88 | 49.50 | 33.15 | 37.74 | 69.60 | 48.99 | 39.36 |
| 2ATEND | 59.90 | 30.00 | 31.38 | 47.40 | 46.50 | 24.13 | 81.00 | 46.50 | 26.76 |
| SR-3ATBEG | 62.00 | 33.00 | 29.88 | 58.50 | 33.90 | 30.88 | 63.63 | 45.00 | 32.13 |
| SR-4ATBEG | 73.71 | 24.15 | 22.63 | 63.30 | 45.81 | 21.38 | 80.64 | 32.31 | 34.00 |
| SL-3ATEND | 62.64 | 25.65 | 14.75 | 47.40 | 47.49 | 25.50 | 75.40 | 53.67 | 29.13 |
| SL-4ATEND | 50.70 | 37.14 | 26.00 | 54.90 | 30.30 | 24.13 | 62.01 | 38.01 | 30.00 |
| REPDOME-3 | 82.71 | 46.50 | 29.00 | 58.50 | 33.90 | 33.75 | 80.55 | 41.64 | 36.30 |
| REPDOME-4 | 91.35 | 39.81 | 16.75 | 37.20 | 45.48 | 28.50 | 85.09 | 28.50 | 27.75 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 80%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 88.89 | 35.50 | 23.25 | 68.75 | 28.17 | 31.13 | 80.92 | 43.33 | 35.25 |
| 2ATEND | 75.78 | 26.50 | 28.00 | 58.75 | 56.67 | 26.63 | 83.05 | 44.33 | 31.88 |
| SR-3ATBEG | 71.11 | 28.33 | 30.38 | 61.00 | 35.00 | 29.12 | 77.57 | 39.67 | 41.12 |
| SR-4ATBEG | 84.76 | 32.17 | 12.75 | 52.75 | 27.17 | 34.63 | 95.67 | 42.17 | 29.50 |
| SL-3ATEND | 80.21 | 31.83 | 22.50 | 58.50 | 47.67 | 24.75 | 79.10 | 44.33 | 30.12 |
| SL-4ATEND | 80.36 | 33.33 | 29.25 | 65.75 | 37.83 | 26.25 | 96.74 | 53.83 | 33.75 |
| REPDOME-3 | 85.29 | 41.67 | 26.63 | 55.00 | 38.00 | 35.25 | 96.67 | 37.67 | 35.13 |
| REPDOME-4 | 101.97 | 31.83 | 32.37 | 61.75 | 38.33 | 46.75 | 106.15 | 37.00 | 41.63 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 90%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 107.52 | 30.00 | 29.75 | 71.75 | 39.17 | 35.88 | 86.49 | 68.50 | 39.00 |
| 2ATEND | 96.95 | 39.33 | 33.00 | 80.50 | 32.67 | 31.63 | 75.35 | 56.50 | 34.38 |
| SR-3ATBEG | 74.85 | 32.17 | 27.88 | 56.50 | 39.67 | 43.75 | 110.17 | 52.33 | 31.50 |
| SR-4ATBEG | 100.98 | 34.83 | 31.25 | 54.75 | 47.17 | 31.75 | 71.90 | 52.67 | 42.13 |
| SL-3ATEND | 101.80 | 40.67 | 35.63 | 65.75 | 45.17 | 32.25 | 69.83 | 59.50 | 38.38 |
| SL-4ATEND | 67.88 | 38.92 | 21.25 | 64.50 | 54.00 | 31.50 | 81.79 | 39.50 | 45.88 |
| REPDOME-3 | 95.51 | 38.67 | 33.00 | 71.50 | 59.67 | 27.00 | 107.17 | 56.50 | 31.50 |
| REPDOME-4 | 91.12 | 27.83 | 24.38 | 58.00 | 46.33 | 35.50 | 79.28 | 45.83 | 43.75 |

### **Appendix C**

The near optimal ratio of patients to physicians under the WT scenario for different unpunctuality rates.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scheduling Rules | Robustness (Unpunctuality Rate = 30%) | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 4.50 | 6.78 | 7.62 | 6.66 | 1.99 | 7.46 | 6.51 | 7.28 | 8.35 |
| 2ATEND | 7.50 | 7.08 | 4.86 | 4.87 | 5.48 | 7.24 | 6.77 | 6.93 | 8.50 |
| SR-3ATBEG | 6.99 | 6.92 | 8.68 | 5.15 | 7.74 | 8.91 | 6.18 | 8.53 | 8.89 |
| SR-4ATBEG | 6.91 | 8.61 | 6.87 | 7.83 | 9.34 | 7.29 | 8.01 | 7.67 | 8.19 |
| SL-3ATEND | 6.07 | 5.95 | 5.83 | 7.33 | 9.01 | 8.15 | 6.64 | 6.41 | 7.13 |
| SL-4ATEND | 6.72 | 10.23 | 5.10 | 6.36 | 8.28 | 6.94 | 7.36 | 7.69 | 8.54 |
| REPDOME-3 | 12.23 | 9.06 | 5.25 | 10.17 | 10.75 | 8.52 | 10.48 | 9.38 | 10.64 |
| REPDOME-4 | 9.10 | 7.97 | 7.87 | 14.02 | 8.91 | 7.38 | 13.74 | 10.94 | 10.36 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 40%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 6.08 | 6.79 | 6.28 | 5.42 | 7.72 | 6.99 | 4.37 | 5.67 | 7.89 |
| 2ATEND | 6.05 | 3.80 | 9.02 | 6.05 | 6.89 | 6.58 | 7.06 | 6.97 | 6.58 |
| SR-3ATBEG | 5.79 | 6.32 | 6.26 | 6.53 | 6.97 | 8.19 | 6.67 | 5.90 | 9.71 |
| SR-4ATBEG | 6.83 | 8.80 | 7.03 | 6.69 | 7.58 | 8.78 | 7.57 | 8.17 | 8.66 |
| SL-3ATEND | 6.61 | 7.38 | 8.49 | 6.90 | 7.48 | 7.92 | 6.19 | 5.85 | 8.63 |
| SL-4ATEND | 5.91 | 6.57 | 5.20 | 6.31 | 8.80 | 7.72 | 7.20 | 6.80 | 7.88 |
| REPDOME-3 | 10.90 | 9.10 | 8.64 | 11.47 | 11.49 | 7.83 | 7.35 | 8.75 | 8.02 |
| REPDOME-4 | 12.03 | 9.31 | 6.71 | 12.23 | 9.49 | 8.41 | 13.83 | 9.78 | 7.56 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 50%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 6.37 | 5.38 | 4.68 | 5.90 | 5.89 | 6.07 | 5.05 | 7.03 | 6.33 |
| 2ATEND | 5.41 | 6.30 | 6.92 | 5.94 | 7.31 | 6.47 | 5.47 | 6.77 | 8.81 |
| SR-3ATBEG | 6.73 | 6.29 | 8.41 | 8.23 | 6.93 | 7.94 | 7.06 | 6.92 | 7.77 |
| SR-4ATBEG | 6.78 | 9.83 | 7.87 | 5.08 | 7.42 | 7.60 | 8.28 | 6.82 | 8.75 |
| SL-3ATEND | 6.04 | 7.75 | 5.10 | 5.16 | 8.11 | 6.22 | 7.77 | 7.13 | 6.38 |
| SL-4ATEND | 8.45 | 8.57 | 7.62 | 7.49 | 8.50 | 8.83 | 6.27 | 8.79 | 8.41 |
| REPDOME-3 | 11.69 | 10.23 | 7.86 | 10.38 | 11.04 | 8.28 | 8.26 | 10.05 | 8.23 |
| REPDOME-4 | 10.89 | 8.67 | 6.10 | 12.11 | 9.31 | 8.79 | 12.11 | 8.99 | 8.85 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 60%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 7.46 | 7.27 | 8.08 | 5.78 | 7.33 | 6.67 | 5.43 | 6.45 | 5.84 |
| 2ATEND | 5.67 | 5.00 | 7.98 | 6.26 | 8.52 | 5.21 | 7.05 | 7.73 | 7.70 |
| SR-3ATBEG | 8.31 | 8.11 | 5.31 | 5.38 | 7.01 | 6.96 | 4.58 | 6.53 | 7.99 |
| SR-4ATBEG | 5.62 | 9.84 | 7.00 | 8.33 | 8.08 | 7.14 | 6.06 | 7.29 | 8.20 |
| SL-3ATEND | 6.97 | 8.03 | 8.10 | 5.99 | 7.79 | 7.95 | 5.58 | 7.10 | 8.13 |
| SL-4ATEND | 8.93 | 7.90 | 9.02 | 6.58 | 6.44 | 7.41 | 6.09 | 8.93 | 7.41 |
| REPDOME-3 | 10.20 | 7.47 | 6.60 | 13.54 | 10.44 | 7.92 | 8.36 | 10.38 | 7.71 |
| REPDOME-4 | 11.21 | 7.57 | 6.26 | 12.54 | 10.85 | 7.83 | 13.03 | 9.90 | 9.38 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 70%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 4.20 | 6.18 | 5.07 | 2.70 | 6.60 | 7.68 | 2.52 | 7.20 | 6.78 |
| 2ATEND | 3.00 | 5.58 | 7.16 | 1.68 | 7.08 | 8.58 | 1.98 | 7.56 | 6.30 |
| SR-3ATBEG | 2.28 | 8.28 | 7.09 | 2.70 | 5.64 | 5.87 | 2.28 | 8.22 | 7.80 |
| SR-4ATBEG | 2.52 | 10.02 | 6.33 | 3.12 | 8.58 | 7.57 | 3.12 | 7.92 | 10.20 |
| SL-3ATEND | 4.56 | 8.82 | 9.39 | 3.96 | 6.36 | 6.93 | 1.56 | 7.92 | 7.62 |
| SL-4ATEND | 3.96 | 6.36 | 7.70 | 4.38 | 7.32 | 6.00 | 1.86 | 2.85 | 8.00 |
| REPDOME-3 | 8.34 | 7.44 | 6.68 | 6.06 | 9.48 | 7.26 | 6.18 | 9.30 | 8.40 |
| REPDOME-4 | 8.00 | 5.76 | 7.22 | 7.56 | 8.82 | 8.83 | 6.30 | 9.72 | 8.70 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 80%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 6.34 | 5.61 | 8.85 | 4.13 | 8.00 | 6.77 | 5.95 | 5.77 | 9.01 |
| 2ATEND | 4.06 | 6.40 | 5.79 | 6.30 | 8.07 | 5.54 | 6.66 | 4.82 | 6.82 |
| SR-3ATBEG | 5.66 | 7.69 | 8.21 | 5.27 | 7.39 | 6.36 | 4.79 | 6.41 | 8.10 |
| SR-4ATBEG | 5.81 | 6.97 | 8.82 | 7.14 | 10.15 | 7.80 | 6.18 | 8.52 | 8.97 |
| SL-3ATEND | 6.31 | 6.98 | 7.17 | 4.57 | 6.23 | 8.09 | 5.58 | 6.05 | 7.78 |
| SL-4ATEND | 7.40 | 6.16 | 7.44 | 4.92 | 8.08 | 6.22 | 6.20 | 7.89 | 8.78 |
| REPDOME-3 | 12.26 | 9.41 | 7.47 | 10.36 | 8.50 | 9.36 | 9.55 | 8.77 | 7.17 |
| REPDOME-4 | 10.97 | 7.72 | 7.30 | 11.15 | 8.93 | 4.99 | 9.69 | 11.38 | 7.23 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 90%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 5.05 | 6.73 | 7.25 | 4.43 | 7.38 | 6.13 | 6.18 | 5.83 | 8.29 |
| 2ATEND | 6.05 | 8.65 | 8.61 | 5.69 | 7.74 | 7.20 | 5.82 | 5.06 | 6.44 |
| SR-3ATBEG | 4.20 | 8.78 | 6.20 | 6.81 | 7.58 | 7.57 | 7.54 | 6.77 | 3.15 |
| SR-4ATBEG | 8.44 | 7.18 | 7.82 | 6.43 | 9.50 | 7.91 | 7.30 | 7.34 | 9.16 |
| SL-3ATEND | 7.07 | 6.89 | 5.52 | 6.61 | 4.44 | 6.82 | 7.68 | 6.57 | 7.94 |
| SL-4ATEND | 6.17 | 8.97 | 10.46 | 7.25 | 6.63 | 6.60 | 7.40 | 8.64 | 9.04 |
| REPDOME-3 | 10.31 | 8.37 | 5.56 | 9.95 | 7.27 | 7.29 | 8.66 | 10.82 | 9.11 |
| REPDOME-4 | 12.59 | 9.26 | 7.47 | 11.25 | 8.61 | 6.98 | 11.61 | 11.55 | 8.89 |

### **Appendix D**

The near optimal ratio of patients to physicians under the NOAPP scenario for different unpunctuality rates.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scheduling Rules | Robustness (Unpunctuality Rate = 30%) | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 26 | 11 | 1 | 30 | 14 | 3 | 34 | 25 | 12 |
| 2ATEND | 23 | 8 | 2 | 35 | 17 | 6 | 34 | 22 | 11 |
| SR-3ATBEG | 22 | 7 | 0 | 32 | 11 | 2 | 38 | 21 | 6 |
| SR-4ATBEG | 20 | 3 | 3 | 28 | 8 | 2 | 32 | 20 | 6 |
| SL-3ATEND | 24 | 9 | 3 | 30 | 14 | 7 | 35 | 25 | 10 |
| SL-4ATEND | 19 | 5 | 3 | 31 | 11 | 3 | 31 | 19 | 9 |
| REPDOME-3 | 3 | 4 | 2 | 14 | 2 | 1 | 25 | 7 | 2 |
| REPDOME-4 | 9 | 4 | 1 | 5 | 7 | 2 | 13 | 7 | 1 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 40%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 26 | 10 | 6 | 30 | 17 | 7 | 43 | 25 | 10 |
| 2ATEND | 23 | 12 | 1 | 28 | 16 | 6 | 34 | 26 | 14 |
| SR-3ATBEG | 23 | 7 | 4 | 30 | 16 | 4 | 36 | 22 | 7 |
| SR-4ATBEG | 20 | 4 | 2 | 25 | 9 | 1 | 32 | 16 | 3 |
| SL-3ATEND | 21 | 8 | 3 | 28 | 14 | 4 | 36 | 23 | 7 |
| SL-4ATEND | 20 | 9 | 2 | 31 | 9 | 2 | 36 | 19 | 7 |
| REPDOME-3 | 10 | 2 | 2 | 12 | 2 | 2 | 28 | 9 | 1 |
| REPDOME-4 | 5 | 2 | 3 | 6 | 6 | 0 | 10 | 7 | 8 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 50%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 25 | 16 | 2 | 30 | 19 | 9 | 38 | 23 | 15 |
| 2ATEND | 26 | 13 | 6 | 31 | 15 | 6 | 38 | 23 | 11 |
| SR-3ATBEG | 22 | 9 | 1 | 28 | 18 | 5 | 33 | 21 | 10 |
| SR-4ATBEG | 22 | 4 | 0 | 33 | 10 | 4 | 30 | 19 | 8 |
| SL-3ATEND | 23 | 9 | 5 | 33 | 11 | 8 | 32 | 22 | 9 |
| SL-4ATEND | 16 | 6 | 4 | 25 | 9 | 2 | 35 | 14 | 5 |
| REPDOME-3 | 5 | 5 | 3 | 15 | 4 | 5 | 30 | 6 | 3 |
| REPDOME-4 | 7 | 1 | 0 | 9 | 5 | 4 | 14 | 8 | 3 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 60%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 23 | 9 | 4 | 29 | 16 | 4 | 42 | 30 | 14 |
| 2ATEND | 24 | 13 | 2 | 31 | 19 | 9 | 33 | 24 | 14 |
| SR-3ATBEG | 18 | 6 | 2 | 31 | 19 | 5 | 40 | 20 | 9 |
| SR-4ATBEG | 23 | 0 | 0 | 29 | 6 | 0 | 39 | 19 | 8 |
| SL-3ATEND | 25 | 7 | 2 | 30 | 16 | 5 | 36 | 22 | 12 |
| SL-4ATEND | 17 | 6 | 2 | 29 | 13 | 4 | 38 | 13 | 2 |
| REPDOME-3 | 9 | 5 | 6 | 7 | 2 | 3 | 25 | 9 | 5 |
| REPDOME-4 | 5 | 5 | 0 | 10 | 3 | 2 | 14 | 10 | 2 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 70%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 17 | 13 | 5 | 20 | 18 | 5 | 29 | 26 | 11 |
| 2ATEND | 17 | 14 | 3 | 22 | 23 | 6 | 28 | 24 | 15 |
| SR-3ATBEG | 20 | 6 | 2 | 20 | 16 | 3 | 27 | 19 | 8 |
| SR-4ATBEG | 18 | 4 | 1 | 19 | 9 | 5 | 28 | 18 | 2 |
| SL-3ATEND | 14 | 7 | 1 | 17 | 16 | 7 | 29 | 23 | 8 |
| SL-4ATEND | 14 | 9 | 3 | 20 | 9 | 8 | 27 | 20 | 7 |
| REPDOME-3 | 13 | 6 | 0 | 11 | 8 | 2 | 19 | 7 | 3 |
| REPDOME-4 | 6 | 6 | 2 | 10 | 6 | 2 | 19 | 7 | 3 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 80%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 17 | 13 | 5 | 20 | 18 | 5 | 29 | 26 | 11 |
| 2ATEND | 17 | 14 | 3 | 22 | 23 | 6 | 28 | 24 | 15 |
| SR-3ATBEG | 20 | 6 | 2 | 20 | 16 | 3 | 27 | 19 | 8 |
| SR-4ATBEG | 18 | 4 | 1 | 19 | 9 | 5 | 28 | 18 | 2 |
| SL-3ATEND | 14 | 7 | 1 | 17 | 16 | 7 | 29 | 23 | 8 |
| SL-4ATEND | 14 | 9 | 3 | 20 | 9 | 8 | 27 | 20 | 7 |
| REPDOME-3 | 13 | 6 | 0 | 11 | 8 | 2 | 19 | 7 | 3 |
| REPDOME-4 | 6 | 6 | 2 | 10 | 6 | 2 | 19 | 7 | 3 |
| Scheduling Rules | **Robustness (Unpunctuality Rate = 90%)** | | | | | | | | |
| **60:2** | **60:3** | **60:4** | **70:2** | **70:3** | **70:4** | **80:2** | **80:3** | **80:4** |
| 2ATBEG | 25 | 8 | 4 | 31 | 18 | 4 | 36 | 24 | 12 |
| 2ATEND | 23 | 8 | 3 | 31 | 19 | 3 | 37 | 25 | 13 |
| SR-3ATBEG | 23 | 5 | 6 | 28 | 14 | 2 | 32 | 25 | 7 |
| SR-4ATBEG | 17 | 4 | 0 | 28 | 9 | 1 | 34 | 18 | 7 |
| SL-3ATEND | 23 | 6 | 1 | 31 | 21 | 10 | 32 | 23 | 10 |
| SL-4ATEND | 22 | 5 | 0 | 28 | 17 | 5 | 33 | 15 | 6 |
| REPDOME-3 | 8 | 4 | 3 | 13 | 9 | 8 | 28 | 7 | 5 |
| REPDOME-4 | 5 | 2 | 1 | 12 | 7 | 6 | 16 | 5 | 2 |

**References**

[1] Aminizadeh, S., Heidari, A., Dehghan, M., Toumaj, S., Rezaei, M., Navimipour, N. J., ... & Unal, M. (2024). Opportunities and challenges of artificial intelligence and distributed systems to improve the quality of healthcare service. Artificial Intelligence in Medicine, 149, 102779.

<https://doi.org/10.1016/j.artmed.2024.102779>

[2] Niu, T., Lei, B., Guo, L., Fang, S., Li, Q., Gao, B., ... & Gao, K. (2023). A Review of Optimization Studies for System Appointment Scheduling. Axioms, 13(1), 16. <https://doi.org/10.3390/axioms13010016>

[3] Javaid, M., Haleem, A., & Singh, R. P. (2024). Health informatics to enhance the healthcare industry's culture: An extensive analysis of its features, contributions, applications and limitations. Informatics and Health. <https://doi.org/10.1016/j.infoh.2024.05.001>

[4] Schmitt, M. (2024). Digital Health And Improvement Of Healthcare Access (Doctoral dissertation, Purdue University Graduate School).

[5] Kuiper, A., de Mast, J., & Mandjes, M. (2021). The problem of appointment scheduling in outpatient clinics: A multiple case study of clinical practice. Omega, 98, 102122. <https://doi.org/10.1016/j.omega.2019.102122>

[6] Seyedi, P., Eshghi, K., & Carter, M. W. (2024). A paradigm shift in appointment Scheduling: Introducing a decentralized integrated Online booking system. Expert Systems with Applications, 257, 124836. <https://doi.org/10.1016/j.eswa.2024.124836>

[7] Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. European Journal of Operational Research, 258(1), 3-34. <https://doi.org/10.1016/j.ejor.2016.06.064>

[8] Ala, A., Alsaadi, F. E., Ahmadi, M., & Mirjalili, S. (2021). Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II. Scientific Reports, 11(1), 19816. <https://doi.org/10.1038/s41598-021-98851-7>

[9] Ala, A., Torkayesh, S. E., Torkayesh, A. E., & Iranizad, A. (2020). A hybrid genetic algorithm for appointment scheduling in a health examination system. International Journal of Value Chain Management, 11(4), 293-310. <https://doi.org/10.1504/IJVCM.2020.111075>

[10] Rivera, G., Cisneros, L., Sánchez-Solís, P., Rangel-Valdez, N., & Rodas-Osollo, J. (2020). Genetic algorithm for scheduling optimization considering heterogeneous containers: A real-world case study. Axioms, 9(1), 27. <https://doi.org/10.3390/axioms9010027>

[11] Boppana, V. R. (2022). Impact Of Dynamics CRM Integration On Healthcare Operational Efficiency. Available at SSRN 5004925. <https://dx.doi.org/10.2139/ssrn.5004925>

[12] Lee, C. K. H. (2018). A review of applications of genetic algorithms in operations management. Engineering Applications of Artificial Intelligence, 76, 1-12. <https://doi.org/10.1016/j.engappai.2018.08.011>

[13] Albadr, M. A., Tiun, S., Ayob, M., & Al-Dhief, F. (2020). Genetic algorithm based on natural selection theory for optimization problems. Symmetry, 12(11), 1758. <https://doi.org/10.3390/sym12111758>

[14] Huynh, N. T., Huang, Y. C., & Chien, C. F. (2018). A hybrid genetic algorithm with 2D encoding for the scheduling of rehabilitation patients. Computers & Industrial Engineering, 125, 221-231. <https://doi.org/10.1016/j.cie.2018.08.030>

[15] Schäfer F, Walther M, Grimm DG, Hübner A. Combining machine learning and optimization for the operational patient-bed assignment problem. Health Care Management Science. 2023 Dec;26(4):785-806. <https://doi.org/10.1007/s10729-023-09652-5>

[16] Liu, X., Gu, F., Bai, Z., Huang, Q., & Ma, G. (2022). Forecasting of daily outpatient visits based on genetic programming. Iranian Journal of Public Health, 51(6), 1313. <https://doi.org/10.18502/ijph.v51i6.9676>

[17] Squires, M., Tao, X., Elangovan, S., Gururajan, R., Zhou, X., & Acharya, U. R. (2022). A novel genetic algorithm based system for the scheduling of medical treatments. Expert Systems with Applications, 195, 116464. <https://doi.org/10.1016/j.eswa.2021.116464>

[18] Jlassi, J., Rekik, I., Elloumi, S., & Chabchoub, H. (2023). Genetic Algorithm for Patients Scheduling in Emergency Department: A Case Study. International Journal of Supply and Operations Management, 10(4), 439-455. <https://dx.doi.org/10.22034/ijsom.2023.109945.2766>

[19] Apornak, A. (2021). Human resources allocation in the hospital emergency department during COVID-19 pandemic. International Journal of Healthcare Management, 14(1), 264-270. <https://doi.org/10.1080/20479700.2020.1861173>

[20] Isa, F. M., Ariffin, W. N. M., Jusoh, M. S., & Putri, E. P. (2024). A Review of Genetic Algorithm: Operations and Applications. Journal of Advanced Research in Applied Sciences and Engineering Technology, 40(1), 1-34. <https://doi.org/10.37934/araset.40.1.134>

[21] Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. Reliability engineering & system safety, 91(9), 992-1007. <https://doi.org/10.1016/j.ress.2005.11.018>

[22] Shukla, A., Pandey, H. M., & Mehrotra, D. (2015, February). Comparative review of selection techniques in genetic algorithm. In 2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE) (pp. 515-519). IEEE. <https://doi.org/10.1109/ABLAZE.2015.7154916>

[23] Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 380-384). IEEE. <https://doi.org/10.1109/COMITCon.2019.8862255>

[24] Mirjalili, S., & Mirjalili, S. (2019). Genetic algorithm. Evolutionary algorithms and neural networks: theory and applications, 43-55.

[25] Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A., & Leong, K. Y. (2017). Crossover and mutation operators of genetic algorithms. International journal of machine learning and computing, 7(1), 9-12. <http://eprints.uthm.edu.my/id/eprint/3688>