dHealth 2025
M. Baumgartner et al. (Eds.)

doi:10.3233/SHTI250198

Outpatient Scheduling with Genetic Algorithm: The Power of Mutation Operators

Veronika GOMBÁS[a], Péter Bálint SCSIBRÁN[a], Tibor DULAI[a] and Ágnes VATHY-FOGARASSY[a,1]
[a] *Department of Computer Science and Systems Technology, University of Pannonia, Veszprem, Hungary*

Abstract. Background: Outpatient scheduling is a complex and time-consuming task. To address this challenge, numerous studies have developed various optimization methods, including genetic algorithms. Objectives: This study aims to develop a task-specific genetic algorithm and investigate the effect of different mutation operators on its performance, focusing on minimizing the earliest completion time of scheduled examinations. Methods: Random and two heuristic mutation operators were designed and compared. The effect of these mutation operators and their parameters were evaluated across four fundamentally distinct scheduling scenarios. Results: The exponential mutation operator outperformed all others across all scheduling problems. It achieved an optimal schedule in 100% of runs for the simplest task and in 74.5% of runs for the most complex one. In comparison, the random mutation operator achieved 100% and 1%, while the polynomial operator reached 75.66% and only 0.22%, respectively. Conclusion: The efficiency of the genetic algorithm developed for outpatient scheduling is strongly influenced by the choice of mutation operator. The performance of the algorithm can be greatly enhanced by employing a specialized mutation operator tailored to the objective function.

Keywords. Outpatient scheduling, genetic algorithm, mutation operator

1. Introduction

Optimal scheduling of health examinations and treatments is crucial for resource planning and patient satisfaction. The first article that highlighted the complexity of this problem concluded that patient scheduling could not be centralized in the healthcare sector [1]. Since then, the significant increase in computational capacity and advancements in algorithms have opened the door to numerous possibilities. In recent decades, numerous articles [2] have been published proposing potential solutions to healthcare scheduling problems using both traditional and artificial intelligence methods [3]. For example, Zhou et al. [4] employed mathematical optimization techniques to develop a two-stage stochastic optimization model, which they validated using real-world data. De Queiroz et al. [5] investigated effective coordination strategies for patients arriving at emergency departments by applying the Variable Neighborhood Search method. Li et al. [6] implemented the Q-learning algorithm, while other studies

[1] Corresponding Author: Ágnes Vathy-Fogarassy, Department of Computer Science and Systems Technology, University of Pannonia, Veszprem, Hungary, E-Mail: vathy.agnes@mik.uni-pannon.hu

utilized Markov decision processes [7], or deep learning reinforcement learning [8] to address various healthcare scheduling challenges.

The genetic algorithm [9], as a population-based metaheuristic search method, appears to be a promising approach for effectively solving this problem. For example, in study [10], surgical scheduling is optimized using a genetic algorithm, and its effectiveness is compared to similar algorithms, such as particle swarm optimization. Additionally, a genetic algorithm incorporating the shortest completion time as its objective function was applied to schedule medical treatments in the studies by Squires [11] and Karpagam [12]. In these studies, the proposed genetic algorithm also demonstrated its effectiveness in terms of both quality and time efficiency.

These latest studies also indicate that outpatient scheduling remains an open question, and unfortunately, in everyday practice, patient scheduling is still performed manually by medical support staff. Our main objective is also to develop and optimize a task-specific genetic algorithm for the outpatient scheduling problem to achieve an optimal or near-optimal solution within a reasonable timeframe. The choice of the genetic algorithm as the solution method was justified by the combinatorially explosive search space of the patient examination scheduling problem, where traditional optimization methods often struggle to scale effectively. Genetic algorithms excel at efficiently exploring large and complex search spaces, avoiding local optima traps, and finding near-optimal scheduling solutions within acceptable computational costs. Although numerous articles have generally highlighted the impact of genetic operators of genetic algorithm-based solutions on results [13, 14], no study has yet provided a detailed analysis of their effects within the field of patient scheduling. As part of our research, we thoroughly investigate the impact and significance of genetic operations and genetic algorithm parameters on scheduling effectiveness. In this article, we focus on the influence of the mutation operator and the importance of selecting an appropriate mutation mechanism.

2. Methods

The input to the genetic algorithm consists of the appointment calendars of the physicians and the requested examinations. The state space in which the genetic algorithm searches for the optimal scheduling solution is defined by the physicians' calendars. In these calendars, available time intervals are identified as time slots where examinations can be scheduled.

The genetic algorithm evolves the population over generations to find the optimal solution. The population consists of individuals (chromosomes), each representing a potential scheduling solution. In the genetic algorithm, an individual is defined as a list of scheduled examinations, where each gene corresponds to an allocated time slot for a specific examination, represented as a calendar-reserved slot pair. The genetic algorithm must identify feasible solutions, ensuring that all examinations are scheduled within the available time slots of the corresponding treatment calendars and that no scheduled appointments overlap. The algorithm evaluates each individual using a fitness function, which quantifies the quality of the solution. The formal definition of the fitness function depends on the specific scheduling problem, allowing for optimization based on different scheduling objectives. For example, examinations may be scheduled to finish as early as possible, or the objective may be to minimize the time elapsed between the first and last scheduled examination. In this article, due to space constraints, we focus exclusively on the first scheduling objective and analyze how different mutation operators contribute to achieving this goal. Based on this, the fitness function to be minimized is defined as follows:

$$f(x_i) = \frac{max_{k=1}^{n}\,(t_{x_i}^{k}) - t_{first}}{t_{last} - t_{first}} \tag{1}$$

where n is the number of examinations to be scheduled, t_{first} is the index of the first available slot, and t_{last} is the index of the last available slot in the union of treatment calendars. It is important to highlight that there is a positive linear relationship between the index of the slots and their corresponding actual time points. This means that the later the time point occurs, the higher the index of the time slot representing that time point. Furthermore, $t_{x_i}^{k}$ denotes the index of the time slot in the k-th gene of individual x_i.

The genetic algorithm generates new individuals in each generation through crossover. Crossover can also be implemented in different ways; in this study, a one-point crossover was applied. The newly created individuals may undergo mutation with a given probability. During mutation, a randomly selected examination is rescheduled to a free time slot within the same treatment calendar. In this study, the effectiveness of three fundamentally different mutation operators is examined. The analyzed operators include the well-known random mutation operator, along with two novel heuristic mutation operators. One applies an exponential weighting function, while the other utilizes a polynomial weighting function to select the new time slot. The new time slot for the examination is selected using roulette wheel selection based on the weight values.

The exponential weight function for mutation is defined as follows:

$$w_{exp}(t_j) = \begin{cases} \frac{(t_{last} - t_j + 1)^{k_1}}{(t_{last} - 1)^{k_1}}, & if\ t_i \neq t_j \\ 0, & if\ t_i = t_j \end{cases} \tag{2}$$

The polynomial weight function for mutation is defined as follows:

$$w_{poly}(t_j) = \begin{cases} 1 - \frac{(t_i - t_j + 1)^{k_1}}{{t_i}^{k_1}}, & if\ t_{first} \leq t_j < t_i \\ \frac{(t_{last} - t_j + 1)^{k_2}}{(t_{last} - t_i)^{k_2}}, & if\ t_i < t_j \leq t_{last} \\ 0, & if\ t_i = t_j, \end{cases} \tag{3}$$

where t_j denotes the index of a free time slot of the treatment calendar, and t_i represents the index of the time slot to be mutated. Both weight functions set the index of the time slot to be mutated to 0, thereby forcing the change of individuals, as shown by the characteristics of the weighting functions in Figure 1.

The three mutation operators influence the mutation process differently. The random mutation operator assigns a new time slot randomly, while heuristic mutation operators reschedule based on weighted probabilities. The exponential weighting function favors earlier time slots, whereas the polynomial mutation operator prioritizes slots near the original schedule. We hypothesize that the exponential mutation operator achieves faster and better convergence by promoting the earliest possible completion time in outpatient scheduling.

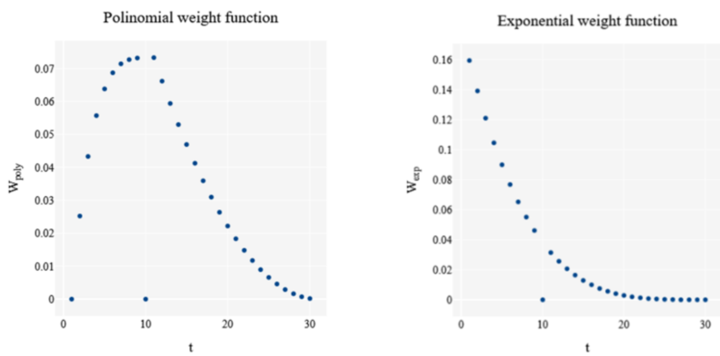

Figure 1.: The characteristics of the polynomial and the exponential weighting functions with parameters k_1=4 and k_2=2.

To evaluate the effectiveness of the three mutation operators, four test scenarios were defined, each representing a different level of scheduling complexity. *Scenario 1* involves scheduling two healthcare examinations within calendars that have a maximum length of 8 weeks. *Scenario 2* requires scheduling three examinations within calendars that span 6 weeks. *Scenario 3* involves scheduling four examinations within calendars that extend up to 10 weeks. *Scenario 4* is the most complex, requiring the scheduling of five examinations within calendars, with the longest extending up to 16 weeks.

The genetic algorithm was executed multiple times for each scenario to measure the impact of mutation operators and their parameters. The parameters of the genetic algorithm that were not examined in this study were set empirically as follows: *population size = 300*, *number of generations = 200*, *elitism percentage* = 30, *mutation rate* = 0.3. The impact of the k_1 and k_2 parameters was analyzed using the GridSearch [15] method within the ranges $k_1 \in [1, \ldots, 6]$ and $k_2 \in [1, \ldots, 6]$. Each mutation function was executed 100 times for every parameter combination in each scenario, starting from the same randomly initialized population. The results of these 100 runs were then evaluated by selecting the best individual from each run as the final solution.

3. Results

The impact of mutation operators and the effect of their associated parameters were investigated based on the difference between the fitness values and the manually determined optimal fitness values for each scenario. Figure 2 presents boxplot diagrams illustrating the distribution of deviations in Scenario 1, the genetic algorithm found the optimal patient scheduling solution in almost every case when using all three mutation operators, except for one instance when polynomial weighting was applied. This scenario represents a simple scheduling task where the heuristic applied during mutation had minimal impact on the results. For more complex scheduling tasks (Scenarios 2–4), the beneficial effect of the heuristic mutation operator with exponential weighting is clearly evident. With this mutation operator, the genetic algorithm found the optimal solution in a significant majority of the 100 runs, producing a non-optimal solution only in a few cases. This was not observed for the polynomial and random mutation operators. The results indicate that random mutation exhibits greater variability; however, in some instances, the algorithm still managed to find the optimal solution. Furthermore, it is evident that polynomial weighting deteriorated the efficiency of the genetic algorithm rather than improving it.

Table 1 shows the ratio of optimal results obtained by the genetic algorithm for different scenarios and mutation operators. As seen, when applying the exponential weighting function, the genetic algorithm reached the optimum in 100% or nearly 100% of the runs for the first three scenarios. Even for the most complex task, it successfully provided an optimal scheduling solution in 74.5% of cases. However, when using the other two mutation functions, the success rate decreased significantly with increasing task complexity. These results demonstrate that the exponential mutation operator enhances the efficiency of the task-specific genetic algorithm when optimizing for the earliest completion time.

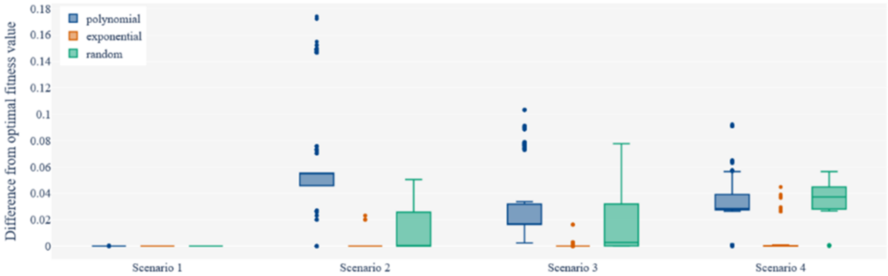

Figure 2. Distributions of fitness value deviations from the optimal value for different mutation operators, grouped by scenario.

Table 1. Percentage of optimal results obtained by the genetic algorithm using various mutation operators in each scenario.

Scenario	Exponential	Polynomial	Random
Scenario 1	100.00 %	75.66%	100.00 %
Scenario 2	99.00 %	0.75 %	53.00 %
Scenario 3	90.00 %	0.00 %	16.00 %
Scenario 4	74.50 %	0.22%	1.00 %

The appropriate value of the k_1 parameter in the exponential weighting function can enhance the efficiency of the genetic algorithm. Figure 3 illustrates the relationship between the deviation of the genetic algorithm's averaged best results from the optimal solution and the value of k_1. Each scenario shows a clear correlation between the parameter value and the average deviation. When $k_1 = 4$, the genetic algorithm reaches the optimum or comes very close to it.

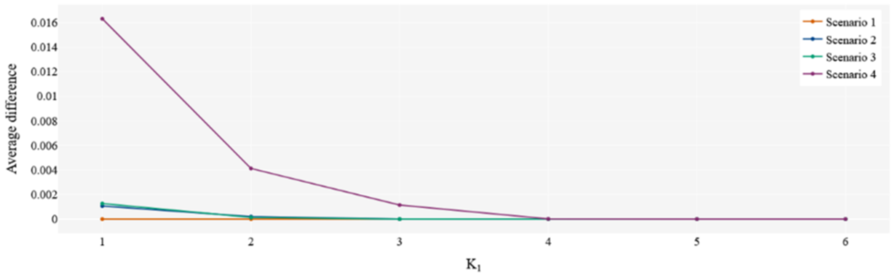

Figure 3. Effect of the k_1 parameter of the exponential mutation operator on the average results.

4. Discussion

The efficiency of the task-specific genetic algorithm for the outpatient scheduling problem was enhanced by the exponential mutation operator, producing optimal or near-optimal solutions. The algorithm achieved nearly 100% optimal results across all examined scenarios using an exponential mutation operator, while the random and polynomial mutation operators performed significantly worse. These findings highlight the importance of selecting appropriate mutation operators in scheduling. Although the exponential mutation operator proved highly effective for minimizing the earliest completion time, further investigation is needed to assess the performance of the polynomial mutation operator for the makespan objective function.

Acknowledgement

This work was supported by the 2020-1.1.2-PIACI-KFI-2020-00045 ('Collaborative Hospital Information Platform,' Development of a process-oriented medical system to support inpatient specialized care) project with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund.

References

[1] G.H. Robinson, P. Wing, L.E. Davis, Computer simulation of hospital patient scheduling systems, *Health Services Research*, **3**(2) (1968) 130-141.

[2] I. Rahimi, A.H. Gandomi, A Comprehensive Review and Analysis of Operating Room and Surgery Scheduling. *Arch Computat Methods Eng* **28**, 1667–1688 (2021).

[3] D.R.T. Knight, C.A. Aakre, C.V. Anstine, B. Munipalli, P. Biazar, G. Mitri, J.R. Valery, T. Brigham, S.K. Niazi, A.I. Perlman, J.D. Halamka, A.M.A. Dabrh, Artificial intelligence for patient scheduling in the real-world health care setting: A metanarrative review, *Health Policy and Technology*, **12** (4) (2023), 100824.

[4] L. Zhou, N. Geng, Z. Jiang, S. Jiang, Integrated Multiresource Capacity Planning and Multitype Patient Scheduling. *INFORMS Journal on Computing* **34**(1) (2021), 129-149.

[5] T.A. De Querioz, M. Iori, A. Kramer, Y.H. Kuo, Dynamic scheduling of patients in emergency departments. *European Journal of Operational Research*, **310**(1) (2023), 100-116.

[6] Li, Yafei, et al. Optimal scheduling in cloud healthcare system using Q-learning algorithm, *Complex & Intelligent Systems*, **8**(6) (2022) 4603-4618.

[7] H. Xu, Y. Fang, C.A. Chou, N. Fard, L. Luo, A reinforcement learning-based optimal control approach for managing an elective surgery backlog after pandemic disruption. *Health Care Management Science*, **26**(3) (2023), 430-446.

[8] J. Zou, Y. Jin. W. Liu, Outpatient scheduling problem in smart hospital with two-agent deep reinforcement learning algorithm, *Discover Computing*, **27** (2024), 41.

[9] S. Sivanandam, S. Deepa, *Introduction to Genetic Algorithms*. Springer, Berlin, Heidelberg, 2008.

[10] M. Eshghali, D. Kannan, N. Salmanzadeh-Meydani, A.M. Esmaieeli Sikaroudi, Machine learning based integrated scheduling and rescheduling for elective and emergency patients in the operating theatre, *Annals of Operations Research*, **332**(1) (2023), 989-1012.

[11] M. Squires, X. Tao, S. Elangovan, R. Gururajan, X. Zhou, U.R. Acharya, A novel genetic algorithm based system for the scheduling of medical treatments, *Expert Systems with Applications*, **195** (2022), 116464.

[12] M. Karpagam, M. Kanipriya, K. Suresh, J. Briskilal, Patient Scheduling System for Medical Treatment Using Genetic Algorithm, *Journal of Population Therapeutics and Clinical Pharmacology*, **30**(8) (2023), 268-273.

[13] K. Deep, M. Thakur, A new mutation operator for real coded genetic algorithms, *Applied Mathematics and Computation*, **193**(1) (2007) 211-230.

[14] I. De Falco, A. Della Cioppa, E. Tarantino, Mutation-based genetic algorithm: performance evaluation, *Applied Soft Computing*, **1**(4) (2002), 285-299.

[15] P. Liashchynskyi, P. Liashchynskyi, Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS, *arXiv preprint arXiv,* **1912** (2019), 06059.