Minimizing Makespan in Open-Shop Operating Room Scheduling Using Variants of Bat Algorithm

Tabark Nameer Abd Ali[1] and Luma S. Hasan[2]
Department of Computer Science, College of Computer Science and Information Technology, University of Al-Qadisiyah, 58001 Al-Diwaniyah, Iraq
{cm.post23.14, luma.hasan }@qu.edu.iq

Keywords: Operating Room Scheduling, Makespan, Bat Algorithm, Operation Room, Surgeries.

Abstract: Operational healthcare management faces growing pressure to develop efficient operating room scheduling systems because these systems require optimal resource management, minimum patient waiting times, and reduced operational expenses. Scheduling problems have traditionally been solved using mixed integer programming (MIP), but these methods have become computationally inefficient when dealing with large complex scheduling scenarios. Exact scheduling methods lose their practical value in hospital operating room planning as the problem complexity rapidly increases according to the number of available operating rooms and scheduled surgeries. The necessity for adaptable intelligent scheduling methods emerges because organisations require methods that maintain excellent solution quality along with swift execution and cost-effective resources. The Bat Algorithm (BA) techniques within swarm intelligence demonstrate successful potential to solve challenging combinatorial issues through natural optimisation methods. This paper uses five Bat Algorithm optimisations to resolve operating room scheduling by reducing makespans. The algorithms examined are the Modified Bat Algorithm (MBA), Chaotic Bat Algorithm (CBA), Discrete Bat Algorithm (DBA), Multi-Objective Bat Algorithm (MOBA), and Binary Bat Algorithm (BBA). For this analysis, two distributions are the Pearson and Fisher distributions. We perform several Operating room scheduling studies to assess the effectiveness and efficiency of these five algorithms on different distributions and compare the results. The findings indicate that all five improved the solution of Operating room scheduling, but the Chaotic Bat Algorithm (CBA) with Fisher's distribution became the best solution for the given scenario.

1 INTRODUCTION

Operating room scheduling is a fundamental optimisation challenge in healthcare systems, where the goal is to allocate a set of surgeries $S = \{s_1, s_2, . . . , s_n\}$ to a set of operating rooms $R = \{r_1, r_2, . . . , r_m\}$ while considering constraints related to surgeon availability, surgery durations, and hospital operational limits. The complexity of this problem arises from the need to minimise key performance indicators such as makespan M while ensuring that each surgery si is scheduled within its feasible time window [T_i start, T_i end]. Stochastic factors, such as variations in surgery durations and unexpected delays, further complicate the optimisation process. Traditional scheduling approaches rely on deterministic models that attempt to minimise an objective function by exhaustively searching the solution space [1]. However, due to the combinatorial nature of the problem, where the number of possible schedules grows exponentially as $O(m_n)$, exact methods such as Mixed Integer Programming (MIP) become computationally impractical for large-scale instances. As a result, metaheuristic algorithms, particularly swarm intelligence techniques, have gained significant attention for their ability to efficiently navigate complex search spaces and provide near-optimal solutions in reasonable time frames. Swarm intelligence-based algorithms, such as the Bat Algorithm (BA), the Bat Algorithm is inspired by the echolocation mechanism of the bats, where the frequency f, velocity v and pulse rate p of all bats vary dynamically in search of improving the quality of their solutions. Adapting their frequency modulation, each bat explores the solution space corresponding to a schedule configuration. Adjusting the intensity of the emitted pulses will perform a local search, or the algorithm will be released for continued global search [2]. The operating room department is a high-cost and high-revenue sector in many hospitals. Hospital administrators continuously seek ways to create schedules that optimise the use of operating rooms while reducing overall expenses. Some hospitals employ an open scheduling method, allowing surgeons to choose any workday for their surgeries, with anaesthetists and nurses coordinating with the surgeons to maximise efficiency [3]. Since the operating room division usually incurs the highest expenses and generates the most income, ensuring effective utilisation of operating rooms is crucial. This study aims to find an effective operating schedule that optimises the use of operating rooms. Due to the complexity of scheduling issues in operating rooms, we propose utilising the Bat Algorithm (BA) to address this problem [4] efficiently.

This paper examines an operating room scheduling issue, employing five recently proposed bat optimisation algorithms to solve the OSS to minimise the makespan. These algorithms include the Modified Bat Algorithm (MBA), Chaotic Bat Algorithm (CBA), Discrete Bat Algorithm (DBA), Multi-Objective Bat Algorithm (MOBA), and Binary Bat Algorithm (BBA), along with two distributions: Pearson distribution and Fisher distribution. Various OSS cases are conducted and compared to evaluate the effectiveness and efficiency of the five algorithms with these distributions. The results demonstrate that all five algorithms can significantly improve the solution to open shop scheduling, with the Chaotic Bat Algorithm (CBA) using Fisher's distribution on the artificial dataset appearing to be the most competitive solver for the given cases. The remainder of this paper is structured as follows: Section 2 discusses related works on OSSP, Section 3 presents the problem formulation, Section 4 briefly introduces the five Bat algorithms, Section 5 outlines the proposed model, Section 6 presents the results and discussion, and Section 7 concludes the paper.

2 RELATED WORKS

Lin, Yang Kuei Li, et al. [4] utilised the Artificial Bees Algorithm (ABC) and the MEDD and MLPT algorithms to analyse data generated using the third Pearson distribution for operating times ranging from 40 to 150 minutes. Surgeons were randomly assigned to six operating rooms, and the results demonstrated high efficiency for the ABC. Rahimi et al.. [6] discussed a hybrid simulated cooling algorithm (Hybrid SA) designed to address the no-wait open surgery scheduling problem (NWOSP-SCSP). They developed a Mixed Integer Linear Programming (MILP) model for small cases, which was tested using data from hospitals in the Nova Scotia Health Author [7]. Abdelmaguid and Tamer F proposed a solution for the multi-processor store scheduling problem (DMOSP) in medical maintenance and diagnostics. This was achieved using the NSGA-II and MOGWO algorithms, enhanced with simulated annealing (SA) to improve search effectiveness. Using information on several stores, processing durations, operation priority, and no-interruption restrictions, the model was verified against 30 benchmark scenarios. By modeling the surgical scheduling problem as a flexible job shop scheduling (FJSSP) problem, Xiang et al., [8] used the Ant Colony Optimization (ACO) technique. Data from five distinct test cases with resource constraints, such as operating rooms, nurses, and surgeons, were used. Fei and colleagues [9] investigated the surgical scheduling problem by establishing the weekly operating plan using data from a university hospital in Belgium and a hybrid genetic algorithm (HGA) in conjunction with a vertical generation-based hashing process (CGBH).

3 PROBLEM FORMULATION

Operational healthcare management faces growing pressure to develop efficient operating room scheduling systems because these systems require optimal resource management, minimum patient waiting times, and reduced operational expenses. Scheduling problems have traditionally been solved using mixed integer programming (MIP), but these methods have become computationally inefficient when dealing with large, complex scheduling scenarios. Exact scheduling methods lose their practical value in hospital operating room planning as the problem complexity rapidly increases according to the number of available operating rooms and scheduled surgeries. The necessity for adaptable intelligent scheduling methods emerges because organisations require methods that maintain excellent solution quality along with swift execution and cost-effective resources. The Bat Algorithm (BA) techniques within swarm intelligence demonstrate successful potential to solve challenging combinatorial issues through natural optimisation. Both algorithms receive independent application in optimisation domains, yet researchers have not compared their performance directly for operating room scheduling in the literature.

4 BAT ALGORITHM

The Bat Algorithm (BA) was proposed by Xin-She Yang in 2010. It is a unique optimisation technique that mimics a bat's behaviour as it seeks prey and avoids obstacles during twilight by utilising echolocation similar to microbats (see Fig. 1).

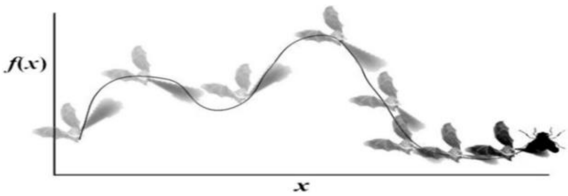

Figure 1: The trajectory of a single bat when it searches for prey and/or avoids difficulties in dusk [10].

This search method is a new metaheuristic swarm intelligence optimisation strategy for global numerical optimisation, and it is inspired by the social behaviour of bats and the phenomenon of echolocation, which is utilised to detect distance. The purpose of its creation was to be applied to global numerical optimisation. The criteria, which are simplified in the [5] approach, can be used to estimate or idealise bat echolocation. All bats use echolocation to improve their sense of location and appear to "know" where they are at all times. Bats employ a range of foraging strategies to find prey, such as flying randomly, flying at a set frequency at a particular location, and using a variable wavelength and loudness A0. They could alter the amplitude r [0, 1] of the pulses they generated and their frequency and wavelength r [0, 1], which vary according to their proximity to their target. The loudness is thought to range from a small constant (positive) Amin to a high A0, though it can vary [11]. This means that each bat is described by its position (x), velocity (v), frequency (f), loudness (A), and emission pulse rate (r). The new position (x_i) and velocities (v_i) can be calculated using the approach for:

$$f_i = f_{min} + (f_{max} - f_{min}) .rand() \quad (5)$$

$$v_i^{t+1} = v_i + (x_i - x^*).f_i \quad (6)$$

$$x_i^{t+1} = x_i + v_i . \quad (7)$$

Where [0, 1] is a random vector produced from a uniform distribution, and β∈ [0, 1] is a constant vector. The best position is determined by comparing the best positions of all bats worldwide. Depending on the domain size in which the issue is addressed, implementation usually allocates frequency f values between 0 and 100. First, an evenly distributed random frequency is assigned to each bat from the range [fmin, fmax]. Once the best solution has been selected from the previously recognised best solutions, a new solution is generated for every bat using the random walk technique.

$$x_{new} = x^* + A^t. \quad (8)$$

The typical PSO functions similarly to fi, which regulates particle velocity and updates bat locations and velocities [12]. This is because fi regulates their speed and distance. To a certain extent, BA combines the traditional PSO with the more focused local search controlled by the sound's volume and pulse rate. It is also important to remember that, as shown in (9), the loudness. And the rate of pulse emission changes as the iterations continue.

$$A_i^{t+1} = \alpha A_i^t ,$$
$$r_i^t = r_i^0 [1-\exp(-\gamma\tau)]. \quad (9)$$

4.1 Bat Algorithm Variants

This section explains the Bat Algorithm (BA) variants and showcases various methods to enhance its efficiency. It is divided into five subsections.

4.1.1 Binary Bat Algorithm (BBA)

The original Bat Algorithm is primarily designed for continuous optimisation. Its operators must be adjusted to specialise in BA for binary optimisation problems. Several researchers have proposed the binary version of BA, namely the binary bat algorithm (BBA). In BBA, artificial bats move in binary searching space by moving from "0" to "1" and from "1" to "0" [5]. Moreover, the feature selection problem has been solved using the Enhanced Binary Bat Algorithm (EBBA). EBBA augments three methods to improve the efficiency of BA in feature selection: the Lévy flight-based mechanism, chaos-based loudness [7], and the procedure to enhance the diversity of the population [13].

4.1.2 Discrete Bat Algorithm (DBA)

Xin-She Yang created the original bat method in 2010 [6], and the Discrete Bat method is a variation of that technique. To find prey, it mimics how bats use echolocation. This algorithm addresses discrete optimisation problems, including routing problems, task scheduling, and shortest path determination. The Discrete Bat Algorithm uses vectors or combinations, like binary encoding or permutations, to express solutions in a discrete space. The basic functions, such as updating locations and velocities, are altered to accommodate the distinct features of the issues it tackles. The algorithm aims to strike a balance between exploring new areas and making use of promising solutions. The main distinction between the discrete and original versions of the BA is that the former uses continuous calculations to update positions, whilst the latter modifies these operations to operate in a discrete search space [10].

4.1.3 Modified Bat Algorithm (MBA)

Although the original Bat Algorithm works well for many optimisation problems, it has drawbacks, including restricted exploration capabilities and the potential to become stuck in local optima. The Modified Bat Algorithm (MBA) is one of the changes suggested to overcome these problems. These changes are intended to improve the algorithm's capacity for exploration and prevent it from being trapped in local optimal solutions [14].

4.1.4 Chaotic Bat Algorithm (CBA)

An improvement on the original Bat Algorithm, the Chaotic Bat Algorithm (CBA), incorporates chaotic ideas to enhance exploration and exploitation capabilities while searching for optimal solutions. This approach efficiently avoids local solutions and increases the probability of achieving global optimal solutions using chaotic systems' deterministic and aperiodic properties [15].

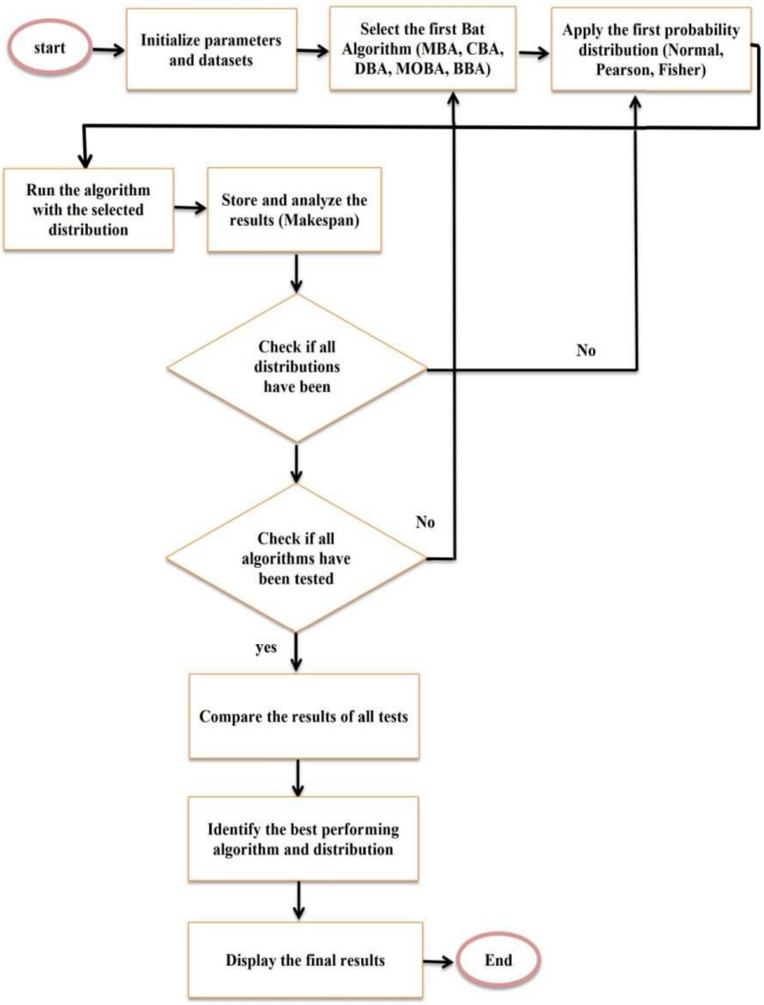

Figure 2: The proposed system.

4.1.5 Multi-Objective Bat Algorithm (MOBA)

Xin-She Yang created the initial bat algorithm in 2010 and expanded it into the Multi-Objective Bat Algorithm (MOBA). MOBA focuses on multi-objective optimisation problems, particularly those involving competing objectives. This approach, which was modified to satisfy the requirements of multi-objective optimisation, mimics bats' echolocation behaviour [16].

5 PROPOSED MODEL

This proposed system (Fig. 2) uses five types of Bat Algorithms for solving Operating room scheduling using these five variants of BA (Bat Algorithm): the Modified (BA), Chaotic (BA), Discrete (BA), Multi-Objective (BA), and Binary (BA) with Pearson's, Fisher distribution.

6 RESULT AND DISCUSSION

The system was implemented using Python 3.12 on a machine with the following specifications: Intel Core i7-4600U 2.10 GHz with Turbo Boost up to 2.70GHz, 8.00GB RAM, 64-bit x64 architecture.

6.1 Variants of Bat Algorithm with Pearson Distribution

In parameters (number of surgeries = 10, number of rooms = 3, number of surgeons = 5, iterations = 100, population = 20, location = 90), with used the Equation BMS (Balanced Makespan) :

$$BMS = Max\ (T) - \frac{1}{N}\sum_{i=1}^{N} T_i\ , \quad (10)$$

where $T = (T_1, T_2, \ldots, T_N)$ is the vector of processing times for each room, N is the number of rooms, $Max\ (T)$ represents the maximum processing time among all rooms, $\sum_{i=1}^{N} T_i$ is the total processing time across all rooms.

The Modified Bat Algorithm improves upon the standard Bat Algorithm by refining the position update strategy and incorporating adaptive parameters for better convergence. In this case, the MBA achieved a makespan of 178.2, which is reasonable. The Chaotic Bat Algorithm employs chaotic maps to increase randomness in position updates, helping to escape local optima and enhance diversity in the search space. However, in this case, the CBA resulted in a higher makespan of 176.0 compared to the Multi-Objective Bat Algorithm (MBA), suggesting that excessive randomness may lead to less efficient solutions. With a makespan of 184.5, the Discrete Bat Algorithm showed the highest value in the Pearson distribution. This suggests that the performance and instability of solutions vary greatly. Its intermittent nature is the primary cause of this instability, making the search less fluid than other algorithms. This feature makes it more difficult for the algorithm to continue improving, even if it might be useful for applications that need rigorous selection. Performance stability might be enhanced by optimising update techniques, which would lessen the detrimental consequences of excessive volatility. The Multi-Objective Bat Algorithm (MOBA) is a well-known algorithm for solving multi-objective problems because it balances exploration and exploitation. Its makespan of 171.4, less than that of the MBA and CBA, suggests that it is more stable when dealing with multi-criteria optimisation problems. Even if it can yield excellent results, improving the search approach could further lower variance. As a result, it works well when balancing several goals is necessary, but it might not be the ideal choice for issues with a single answer. With a makespan of 167.1, the Binary Bat Algorithm (BBA) yielded the lowest value in the Pearson distribution. This reflects excellent stability in performance compared to the other algorithms. The BBA is based on a binary representation, which allows it to search in a structured manner, effectively identifying optimal solutions. This organised approach reduces solution fluctuations by concentrating the search in specific areas, enhancing efficiency in problems with clear constraints (Table 1 and Table 2). Resulting in personal distributions for various bat algorithms.

6.2 Variants of Bat Algorithm with Fisher Distribution

In parameters (No. surgeries = 2172, No. rooms = 62, No.surgeons = 334, iterations = 700, population = 200).MOBA performed best in Makespan, with the lowest value among all algorithms. It also achieved the best balance (Balanced Makespan = 831.58), indicating a more equitable distribution of work across operating rooms. MOBA's success can be explained by the fact that it considers multiple objectives simultaneously, allowing for a good balance between minimising overall time and achieving balanced distribution. CBA performed worst regarding Makespan, with the highest value (4178.26 minutes), but Balanced Makespan was (969.57). BBA a high Makespan (4146.85 minutes). Balanced Makespan was 938.16, which is significantly unbalanced. This is likely due to BBA relying on binary updates, making it less flexible in adapting to process distribution requirements. DBA performed slightly better than BBA but still poorly compared to MOBA and MBA. The balanced makespan was 936.75, roughly equal to BBA's, indicating inefficient process distribution. This result suggests that DBA could not find a highly efficient schedule, perhaps due to the intermittent updates, which may have limited the search for optimal solutions. In parameters (No. surgeries = 10, No. rooms = 3, No.surgeons = 5 , iterations = 100, population = 20) with.(MBA) MBA performed very well, achieving the second-best Makespan after CBA. The distribution was fairly balanced, with Balanced Makespan = 17.11, which is good compared to other algorithms but not the best. Therefore, we conclude that the MBA was efficient but not the best in balance compared to the CBA.MOBA's performance was close to an MBA's but had a slightly higher Makespan. Balanced Makespan was 17.55 minutes, close to MBA but slightly worse. Therefore, we conclude that MOBA did not outperform MBA but achieved similar results. It had the worst Makespan of all the algorithms (350.7 minutes), meaning it did not find an efficient solution like the other algorithms. The balanced makespan was 25.3 minutes, larger than all the other algorithms, indicating that the work distribution was not as good. Therefore, we conclude that BBA performed poorly compared to the other algorithms in Makespan and Balance. The best makespan among all algorithms (334.5 minutes), meaning that CBA found the optimal solution in terms of execution time. The best distribution of work across rooms, with a Balanced Makespan = 9.2 minutes, is lower than all other values, indicating a very balanced distribution. Therefore, we conclude that CBA outperformed all other algorithms regarding execution time and balanced distribution, making it the best choice. DBA achieved an average result with Makespan = 347.4 minutes, which is better than BBA but worse than MBA, MOBA, and CBA. Balanced Makespan was 22.11 minutes, which is not bad but not the best. Therefore, we conclude that DBA was better than BBA but not the best overall. In general, CBA is the best choice because it has achieved the lowest makespan and the best distribution of work. If there are limitations to using CBA, MBA or MOBA are the next best choices. BBA is not a good choice for this problem because it achieved the worst results.

Table 1: Various BA with pearson distributions.

No.Surgeries	No. R	No.Surgeons	iter	Pop.	Shape_param	Scale_param	Various BA.	Distribution	Mk.
10	3	5	100	20	3	90	MBA	Pearson	178.2
10	3	5	100	20	3	90	CBA	Pearson	176.0
10	3	5	100	20	3	90	DBA	Pearson	184.5
10	3	5	100	20	3	90	MOBA	Pearson	171.4
10	3	5	100	20	3	90	BBA	Pearson	167.1

Table 2: Variants of the BA with fisher distributions.

No. Surgeries	No. R	No. Surgeons	iter	Pop.	Dfn-Dfd	Scale	Various BA.	Distribution	Mk.	MK.MBS
2172	62	334	700	200	20,45	90	MBA	Fisher	4073.9	865.2
2172	62	334	700	200	20,45	90	CBA	Fisher	4178.2	969.5
2172	62	334	700	200	20,45	90	DBA	Fisher	4145.4	936.7
2172	62	334	700	200	20,45	90	MOBA	Fisher	4040.2	831.5
2172	62	334	700	200	20,45	90	BBA	Fisher	4146.8	938.1
10	3	5	100	20	15,60	90	MBA	Fisher	342.4	17.1
10	3	5	100	20	15,60	90	BBA	Fisher	350.7	25.3
10	3	5	100	20	15,60	90	CBA	Fisher	334.5	9.2
10	3	5	100	20	15,60	90	MOBA	Fisher	342.8	17.5
10	3	5	100	20	15,60	90	DBA	Fisher	347.4	22.1

7 CONCLUSIONS

Reducing costs while maintaining high-quality services is a priority for many healthcare facilities. The operating room is often one of the most expensive departments in a hospital, so hospital managers are constantly seeking effective ways to optimise their Use to save on operational costs. This The research employs five newly proposed Bat optimisation algorithms to address operating room scheduling and reducing the manufacturing period. The algorithms include the MBA, DBA, Multi-MOBA, BBA, and CBA, which were tested with two types of distributions: Pearson distribution and Fisher distribution. The results indicated that the CBA achieved the best performance when applied to an artificial dataset using Fisher distribution. Future research will focus on optimizing Bat algorithm variants, addressing constraints, and comprehensively exploring more realistic goals. The findings indicate that all five improved the solution of Operating room scheduling, but the CBA with Fisher's distribution became the best solution for the given scenario.

REFERENCES

[1] J. Dick, J. M. Schumann, B. Nuseibeh, L. Athens, J. Zobel, and A. M. Qazi, Scheduling: Theory, Algorithms, and Systems, 2022.

[2] K. R. Baker and D. Trietsch, Principles of Sequencing and Scheduling, Hoboken, NJ: John Wiley & Sons, 2018.

[3] M. M. Ahmadian, M. Khatami, A. Salehipour, and T. Cheng, "Four decades of research on the open-shop scheduling problem to minimise the makespan," European Journal of Operational Research, vol. 295, no. 2, pp. 399–426, 2021.

[4] Y. K. Lin and M. Y. Li, "Solving operating room scheduling problem using artificial bee colony algorithm," Healthcare, vol. 9, no. 2, 2021, doi: 10.3390/healthcare9020152.

[5] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74, 2010.

[6] A. Rahimi, S. M. Hejazi, M. Zandieh, and M. Mirmozaffari, "A novel hybrid simulated annealing for no-wait open-shop surgical case scheduling problems," Applied System Innovation, vol. 6, no. 1, pp. 1–21, 2023, doi: 10.3390/asi6010015.

[7] T. F. Abdelmaguid, "Bi-objective dynamic multiprocessor open shop scheduling for maintenance and healthcare diagnostics," Expert Systems with Applications, vol. 186, May 2021, doi: 10.1016/j.eswa.2021.115777.

[8] W. Xiang, J. Yin, and G. Lim, "An ant colony optimisation approach for solving an operating room surgery scheduling problem," Computers and Industrial Engineering, vol. 85, pp. 335–345, 2015, doi: 10.1016/j.cie.2015.04.010.

[9] H. Fei, N. Meskens, and C. Chu, "A planning and scheduling problem for an operating theatre using an open scheduling strategy," Computers and Industrial Engineering, vol. 58, no. 2, pp. 221–230, 2010, doi: 10.1016/j.cie.2009.02.012.

[10] M. Shehab, et al., "A comprehensive review of bat inspired algorithm: Variants, applications, and hybridization," Archives of Computational Methods in Engineering, vol. 30, no. 2, Springer Netherlands, 2023, doi: 10.1007/s11831-022-09817-5.

[11] W. A. Almahdi, H. A. Lafta, and Y. H. Ali, "Intelligent task scheduling using bat and harmony optimization," Iraqi Journal of Science, vol. 64, no. 8, pp. 4187–4197, 2023, doi: 10.24996/ijs.2023.64.8.38.

[12] Y. K. Lin and Y. Y. Chou, "A hybrid genetic algorithm for operating room scheduling," Health Care Management Science, vol. 23, no. 2, pp. 249–263, 2020, doi: 10.1007/s10729-019-09481-5.

[13] J. Feng, H. Kuang, and L. Zhang, "EBBA: An enhanced binary bat algorithm integrated with chaos theory and Lévy flight for feature selection," Future Internet, vol. 14, no. 6, pp. 1–16, 2022, doi: 10.3390/fi14060178.

[14] S. U. Umar, T. A. Rashid, A. M. Ahmed, B. A. Hassan, and M. R. Baker, "Modified bat algorithm: A newly proposed approach for solving complex and real-world problems," Soft Computing, vol. 28, no. 13–14, pp. 7983–7998, 2024, doi: 10.1007/s00500-024-09761-5.

[15] A. H. Gandomi and X. S. Yang, "Chaotic bat algorithm," Journal of Computational Science, vol. 5, no. 2, pp. 224–232, 2014, doi: 10.1016/j.jocs.2013.10.002.

[16] X. S. Yang, "Bat algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp. 267–274, 2011, doi: 10.1504/IJBIC.2011.042259.