ISSN 2442-6571

Seq2seq encoder–decoder with adaptive metaheuristic for scheduling optimization

Arief Kelik Nugroho [a,1,*], Nurul Hidayat [a,2], Nofiyati [a,3]

[a] Department of Informatics, Faculty of Engineering, Universitas Jenderal Soedirman, Purwokerto, Indonesia
[1] arief.nugroho@unsoed.ac.id; [2] ipung.permadi@unsoed.ac.id; [3] 3nofiyati@unsoed.ac.id
* corresponding author

ARTICLE INFO

Article history
Received September 15, 2025
Revised March 21, 2026
Accepted April 2, 2026
Available online August 31, 2026

Keywords
Job-Shop Scheduling
Seq2Seq
Adaptive Metaheuristic
Makespan Optimization
Scheduling Algorithms

ABSTRACT

The Job Shop Scheduling Problem (JSSP) is a combinatorial optimization problem that is NP-hard and highly complex, particularly in modern manufacturing environments associated with Industry. Conventional metaheuristic methods such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are capable of generating solutions in relatively short time; however, they often face limitations in solution quality due to premature convergence and limited adaptability to dynamic problem conditions. In contrast, deep learning approaches such as Sequence-to-Sequence (Seq2Seq) offer strong representational capabilities for modeling operation sequences, although they still encounter challenges related to training stability and generalization. This study proposes a hybrid approach that integrates a Seq2Seq Encoder–Decoder architecture with an adaptive metaheuristic mechanism to enhance scheduling optimization performance. The Seq2Seq model is utilized to learn underlying patterns in operation sequences, while the adaptive mechanism dynamically adjusts search parameters based on makespan evaluation. The experiments are conducted using datasets from the OR-Library, specifically the 10×10 and 15×15 scenarios, to evaluate the performance and scalability of the proposed method. The experimental results demonstrate that the Seq2Seq + Adaptive Metaheuristic approach consistently produces lower makespan values compared to GA and PSO. For the 10×10 dataset, the proposed method achieves a makespan of 932, outperforming GA (1095) and PSO (1047). Similarly, for the 15×15 dataset, it attains a makespan of 1050, which is better than GA (1250) and PSO (1200). Although the proposed approach requires slightly longer computational time, the improvement in solution quality indicates that it effectively balances exploration and exploitation.

1. Introduction

The Job Shop Scheduling Problem (JSSP) is a classical combinatorial optimization problem categorized as NP-hard, representing one of the most significant challenges in optimization and manufacturing research [1], [2]. JSSP involves determining the optimal sequence of operations for multiple jobs processed on a limited number of machines, aiming to minimize performance indicators such as makespan, delays, and resource utilization efficiency [3]. In the context of Industry, the problem has grown increasingly complex due to dynamic production demands, customer order variability, and system uncertainties [4]. Consequently, intelligent and efficient solutions for JSSP are essential to support highly adaptive and competitive modern manufacturing systems. Traditional approaches such as Mixed Integer Linear Programming (MILP) and Constraint Programming (CP) have proven effective for small-scale instances [1], [5]. However, these methods become computationally infeasible as problem size increases [6], prompting the adoption of metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealings [7], [8], which offer broader solution space exploration and deliver near-optimal solutions in significantly less time. Nevertheless, existing studies indicate that these methods often suffer from slow convergence, susceptibility to local optima, and limited adaptability to dynamic environments [9], [10].

To address these limitations, recent research has turned to machine learning, particularly Deep Learning (DL). The Sequence-to-Sequence (Seq2Seq) architecture, originally developed for natural language processing, has been applied to scheduling problems to automatically model operation sequences [11]. This approach excels at learning complex representations of job and machine sequences while enabling end-to-end decision-making. However, pure deep learning approaches also face significant challenges, such as the need for large datasets, training instability, and limited capacity to generalize to unseen scenarios [7], [12]. Some studies have explored hybridizing deep learning with metaheuristics to overcome their individual shortcomings. For example, integrating GA–ACO and PSO–GA has shown improved solution quality compared to standalone algorithms [13]. Furthermore, methods such as Adaptive Differential Evolution and Memetic Algorithms have been proposed to enhance the exploration–exploitation balance [14]. Despite this progress, many existing studies still rely on static or semi-adaptive parameter settings, with limited automation in parameter adjustment [4], [15]. There is a notable research gap in fully integrating Seq2Seq Encoder–Decoder models with adaptive metaheuristics. Prior studies have mainly focused on deep reinforcement learning [16], [17] or Graph Neural Network integration [18], without fully leveraging Seq2Seq's potential in encoding JSSP operation sequences. In fact, the Seq2Seq model's temporal context awareness could significantly enrich the representation of complex scheduling problems [18], [19].

Conventional metaheuristics typically rely on manually defined initial configurations, such as mutation probabilities, exploration rates, or search intensification thresholds. They lack the flexibility to adapt to the dynamic nature of JSSP instances [10], [20] without performance-driven feedback mechanisms, these algorithms risk stagnation and suboptimal outcomes [21]. Although deep learning models offer strong representational capabilities, they often struggle to achieve stable convergence, especially on large and complex datasets [22], [23], [24]. Conversely, traditional metaheuristics demonstrate stability but converge slowly toward high-quality solutions [25]. Hence, there is a pressing need for a hybrid approach that not only combines their strengths but also dynamically adjusts search parameters to enhance stability and convergence quality. Previous research has predominantly evaluated methods using established benchmark datasets like those from the OR-Library [26]. However, relatively few have examined method effectiveness in dynamic environments characterized by random job arrivals, setup times, or system uncertainties [27], [28], [29]. This highlights the urgent need for approaches capable of addressing realistic conditions in smart factories and industry-driven manufacturing systems [30], [31]. This study proposes a novel integration of the Seq2Seq Encoder–Decoder architecture with adaptive metaheuristics to enhance JSSP performance. The Seq2Seq model is employed to deeply learn the sequencing patterns of operations, while adaptive metaheuristics dynamically adjust search parameters based on makespan evaluation [32]. This hybrid framework aims to improve solution quality, accelerate convergence, and enhance model generalization across varying scheduling scenarios. The primary objectives of this research are: (1) to develop a customized Seq2Seq architecture for representing JSSP operation sequences, (2) to design an adaptive metaheuristic mechanism driven by solution performance feedback, and (3) to evaluate the effectiveness of the proposed hybrid framework in comparison to classical and pure machine learning methods [33], [34], [35]. Through these contributions, this study seeks to advance the development of intelligent, adaptive, and efficient scheduling systems for modern manufacturing environments. The integration of Seq2Seq with adaptive metaheuristics shows promise for deployment in large-scale production systems, dynamic scheduling scenarios [36], and real-time decision-making in industry manufacturing settings [37], [38]. Thus, this research not only provides a solid methodological foundation but also expands the research frontier in AI-driven scheduling optimization.

2. Method

2.1. Dataset

In this study, we use benchmark JSSP datasets from the OR-Library, one of the most commonly used sources in job-shop scheduling research. These datasets offer a wide range of problem instances with different sizes and complexity levels, making them suitable for fair and consistent comparisons across algorithms. Among the most frequently used instances are the 10×10 and 15×15 cases, which represent problems with 10 jobs and 10 machines, and 15 jobs and 15 machines, respectively. The 10×10 case includes 10 jobs, each made up of 10 operations that must be processed on specific machines with varying processing times. Due to its moderate complexity, this instance is often used as an initial benchmark to evaluate the performance of new approaches. On the other hand, the 15×15 case is more challenging, involving 15 jobs and 15 machines, with a total of 225 operations. The differences in processing times and machine sequences across jobs make this dataset well-suited for assessing the robustness and scalability of scheduling methods.

2.2. Procedure

Fig.1 illustrates an iterative process comprising several key stages within an adaptive metaheuristic-based optimization system. The procedure initiates with the encoding of input sequences; this involves transforming raw data into a structured format compatible with the predictive framework. Once the encoding is complete, the system proceeds to the decoding phase, where potential solutions are inferred based on the encoded representations. These generated solutions are subsequently evaluated using the makespan metric, which serves as an indicator of scheduling efficiency. The makespan value provides a quantitative measure for assessing the quality of each proposed solution. Following the evaluation, the results are employed to dynamically adjust the parameters of the metaheuristic algorithm. Variables such as mutation probability, exploration rate, and intensification level are recalibrated to enhance the algorithm's capacity to search the solution space effectively and to avoid premature convergence to local optima.

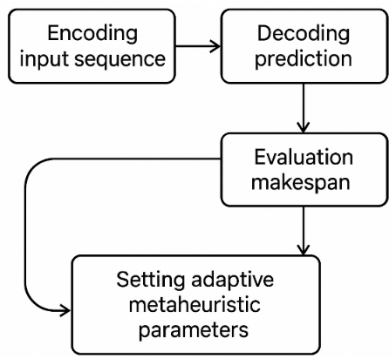

Fig. 1. Flowchart of Seq2Seq Encoder–Decoder integration with adaptive metaheuristic for Job-Shop scheduling.

The process then loops back to the encoding phase, now utilizing the updated parameters. This cycle continues iteratively until the system converges on a solution that meets the predefined optimality criteria. Through this adaptive mechanism, the system incrementally learns from previous outcomes, allowing for continuous refinement and improvement of the solution quality over time.

Fig. 2 illustrates the integration of a Sequence-to-Sequence (Seq2Seq) Encoder–Decoder architecture with the Ant Colony Optimization (ACO) algorithm for solving the Job Shop Scheduling Problem (JSSP). In the bottom-left section, the encoder generates a sequence of hidden states $(h1, h2, \ldots, hT)$, each representing an individual operation within a job scheduling context across machines. These hidden states capture complex sequential dependencies, providing a foundational representation for generating high-quality scheduling solutions. At the center of the diagram, the attention mechanism is depicted using a summation symbol ($\oplus$). This component plays a critical role in selecting and weighting the contribution of each hidden state through attention scores

$(\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,T})$. The resulting aggregated context vector encapsulates relevant information, which is then used by the decoder to update its internal state and generate the next operation in the predicted sequence. This mechanism enables the model to adapt effectively to variations in operational order and scheduling complexity.

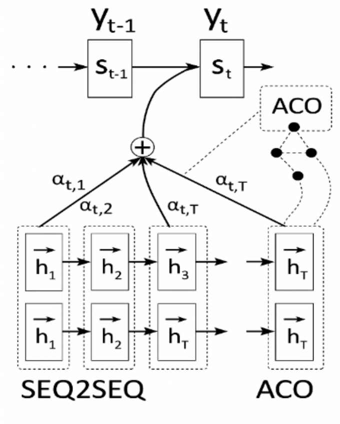

Fig. 2. Integration of Sequence-to-Sequence (Seq2Seq) Architecture with Ant Colony Optimization (ACO) Mechanism.

The upper section of the figure presents the decoder process, which produces predicted outputs (Y_t) based on the previous decoder state $(s_{t,1})$ and the current context from the attention mechanism. This autoregressive process ensures that each output is conditioned on previous predictions, promoting consistency and coherence in the resulting operation sequence. As a result, the decoder dynamically adapts its internal state in response to the evolving structure of the schedule, rather than relying solely on static encoder representations.

On the right-hand side, the integration of the ACO algorithm is visualized, interacting directly with the attention mechanism. ACO optimizes path selection through the hidden state space using pheromone updating, where virtual ants reinforce promising scheduling trajectories while reducing the risk of local optima entrapment. This collaborative framework enables the system to not only learn accurate sequential representations but also guide the search process more globally and adaptively. Overall, the hybridization of Seq2Seq and ACO creates a unified framework that combines deep representation learning with population-based optimization. The Seq2Seq model captures intricate sequence structures within JSSP, while ACO enhances exploration and exploitation strategies. This synergy offers significant advantages in solution quality, convergence speed, and scalability, making the proposed architecture a robust alternative to traditional or single-method scheduling approaches based on either deep learning or metaheuristics.

2.3. Preprocessing

The data is first transformed into sequences of tokenized operations. Augmentation is then performed by applying partial permutations and injecting controlled noise into processing times. The purpose of augmenting benchmark JSSP datasets from the OR-Library is to expand the diversity of available scheduling scenarios, thereby strengthening the generalization capability of the developed model or algorithm. In the context of job-shop scheduling, data augmentation involves modifying processing times, rearranging machine sequences, or adding controlled noise to operation durations. One commonly used augmentation technique is permuting job sequences while preserving the original processing time distribution, effectively shuffling job priorities. This approach is crucial for training algorithms to be robust against variations in job arrival order, particularly in dynamic scheduling environments. Additionally, augmentation can be carried out by introducing slight variations in operation processing times, for example, by adding normally distributed random deviations within a specified range. This strategy generates new problem instances that remain realistic but compel the model to adapt to processing time uncertainty. Such augmentation is also beneficial for creating labeled datasets suitable for machine learning and deep learning approaches. This process ensures that models do not merely memorize patterns in the original benchmark data but also learn to recognize solution structures across a wider range of more complex job-shop scheduling scenarios. The primary objective in solving the Job-Shop Scheduling Problem is to minimize the makespan, which refers to the maximum completion time across all jobs. This goal is crucial because, in many industrial applications, reducing the makespan directly enhances production throughput and improves overall system efficiency.

2.4. Objective function And Parameter

Formally, the objective is expressed as (1)

$$\min C_{max} = max_j C_j \tag{1}$$

The objective function aims to minimize the makespan, defined as the maximum completion time of all jobs in the schedule, where C_{max} denotes the time at which the last job completes processing. Minimizing this value leads to improved system utilization, higher production throughput, and greater overall scheduling efficiency.

The adaptive parameter update mechanism dynamically adjusts metaheuristic parameters such as mutation probability, exploration rate, or intensification factor based on the feedback from solution performance, particularly the makespan. By monitoring the improvement rate or stagnation of the search process, the algorithm modifies these parameters to balance exploration and exploitation. This adaptive adjustment aims to avoid premature convergence, enhance solution quality, and accelerate the convergence speed of the scheduling process in successive iterations.

$$\theta_{t+1} = \theta_t + \alpha(f_{best} - f_{mean}) \tag{2}$$

The fitness function evaluates the quality of a candidate scheduling solution by quantifying its makespan. Typically, the fitness value is defined as inversely proportional to the makespan to align with the objective of minimization. Formally, it can be represented in (3):

$$F(s) = \frac{1}{1+C_{max}(S)} \tag{3}$$

The formula above defines the fitness function for a solution S. A smaller makespan yields a higher fitness value. This formulation guides the algorithm to steer the search process toward solutions with minimal makespan, effectively aligning the optimization direction with the primary objective of reducing overall completion time.

The adaptive update mechanism (5) is defined using explicit mathematical formulations to dynamically adjust parameters such as mutation probability, exploration rate, and intensification factor across iterations.

$$\theta_t + 1 = \theta_t + \alpha.(C_{prev} - C_{current}) \tag{4}$$

To ensure reproducibility and provide a clear overview of the experimental setup, a concise summary of the key parameters used in the proposed framework is presented in Table 1.

Table 1. Key Parameters of Seq2Seq and Adaptive Metaheuristic.

Component	Parameter	Value / Range
Seq2Seq Model	Learning Rate	0.001
	Hidden Size	256
	Batch Size	32
Adaptive Metaheuristic	Mutation Probability	0.1 – 0.5
	Exploration Rate	0.2 – 0.8
	Update Coefficient	0.05 – 0.2

3. Results and Discussion

Fig. 3 illustrates the relationship between Loss and Mutation Probability across training epochs within a metaheuristic- or deep learning-based model. The x-axis represents the number of epochs, ranging from 0 to 50, while the y-axis captures the values of the objective functions, specifically Loss and Mutation Probability. The blue line denotes the Loss values, and the orange line represents the Mutation Probability. The title of the graph clearly indicates that the focus of observation lies in tracking the evolution of both metrics throughout the training process. The Loss curve (blue line) exhibits highly volatile behavior from the beginning to the end of training. Loss values fluctuate sharply between epochs, with certain points nearly reaching zero before surging to magnitudes as high as 1.2 × 10^15. These extreme oscillations suggest instability in the optimization process, potentially stemming from high exploration levels, suboptimal initialization, or the inherent complexity of the objective function that drives highly dynamic solution searches. In contrast, the Mutation Probability curve (orange line) remains nearly constant throughout the iterations. The mutation rate, consistently close to zero, indicates that the mutation mechanism plays a relatively minor or stable role in this training process. This suggests that adaptive behaviors are likely governed by other components, such as exploration probability, intensification mechanisms, or specific update rules, with mutation serving merely as a supplemental operation with limited influence. A comparative analysis reveals that the severe fluctuations in Loss are not mirrored by changes in Mutation Probability. This disparity implies that the primary drivers of Loss instability lie outside the mutation mechanism, likely within the core optimization process, such as ant path selection in ACO or weight updates in neural networks. This observation highlights an opportunity to enhance model performance by dynamically adapting mutation probability to help suppress the pronounced oscillations in Loss.

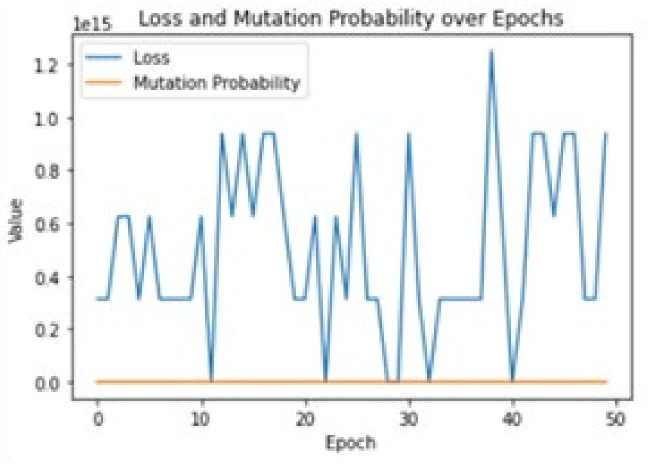

Fig. 3. Loss And Mutation Probability

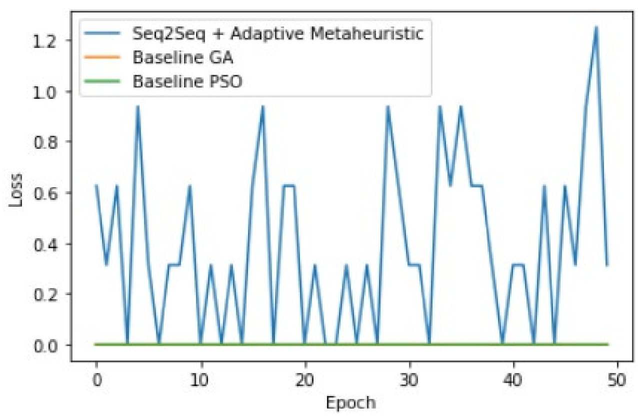

Fig. 4. Loss Trend across Epochs for Seq2Seq + Adaptive Metaheuristic, GA, and PSO.

The graph presents a comparative analysis of Loss values over 50 epochs for three different optimization strategies: Seq2Seq combined with Adaptive Metaheuristic (blue line), Baseline Genetic Algorithm (GA, orange line), and Baseline Particle Swarm Optimization (PSO, green line). The x-axis represents the number of epochs, while the y-axis indicates the corresponding Loss values. This visualization aims to evaluate the effectiveness of the proposed method relative to baseline algorithms in the context of optimization tasks such as the Job Shop Scheduling Problem (JSSP). The blue curve, representing the Seq2Seq + Adaptive Metaheuristic approach, exhibits considerable fluctuations in Loss across epochs. These pronounced variations reflect a highly dynamic search process, characterized by continuous exploration and exploitation. Such behavior indicates that the hybrid approach actively adjusts its search strategy in response to evolving conditions, resulting in a more adaptive and responsive optimization trajectory. In contrast, the orange (Baseline GA) and green (Baseline PSO) curves remain nearly flat and close to zero throughout the training process. This suggests that both baseline algorithms maintain a relatively stable performance but appear less responsive in reducing or adapting the Loss. The lack of significant variation may indicate premature convergence where the algorithm settles into sub-optimal solutions early and fails to improve further over time.

This comparison highlights a key advantage of the Seq2Seq + Adaptive Metaheuristic method: its ability to explore a broader solution space. While the higher fluctuation in Loss might be perceived as instability, it also reflects the model's capacity to escape local optima. Meanwhile, the stability observed in GA and PSO comes at the cost of reduced adaptability, suggesting a potential failure to explore alternative, better-performing regions of the solution space. Overall, the graph underscores the dynamic nature of the proposed hybrid framework compared to traditional baselines. The adaptive strategy embedded within the model enables a more balanced trade-off between exploration and exploitation. Although further refinement is needed to enhance convergence stability, the results suggest that the proposed method holds greater potential for outperforming baseline algorithms in solving complex optimization problems like the JSSP.

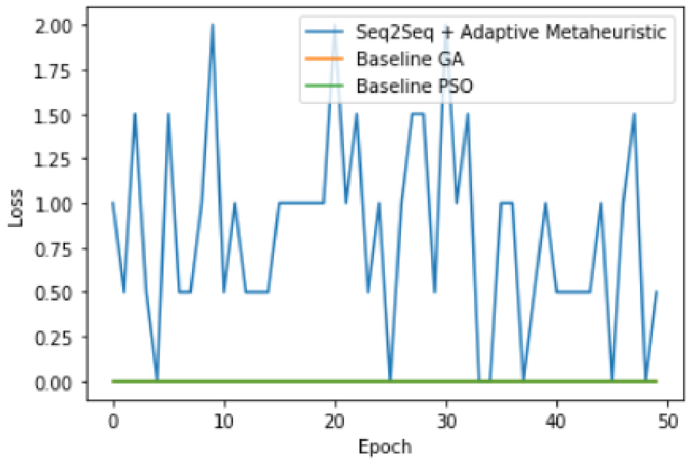

Fig. 5. Loss Comparison over Epochs between Seq2Seq + Adaptive Metaheuristic, GA, and PSO.

Fig. 5 compares the Loss values across 50 training epochs for three different methods: the proposed Seq2Seq combined with an Adaptive Metaheuristic (blue line), the Baseline Genetic Algorithm (GA, orange line), and the Baseline Particle Swarm Optimization (PSO, green line). The x-axis represents the number of training epochs, while the y-axis shows the corresponding Loss values. The primary objective of this figure is to assess the optimization performance of the proposed model relative to the two baseline algorithms. The blue curve, representing the Seq2Seq + Adaptive Metaheuristic approach, demonstrates substantial fluctuations in Loss, ranging from near-zero to nearly 2.0, with an intense up-and-down pattern throughout the training iterations. This behavior indicates an active exploration of the solution space, wherein the algorithm persistently evaluates diverse configurations in pursuit of optimal solutions. Although this pattern may appear unstable, such volatility often reflects the model's capability to escape local optima and explore alternative regions of the search space. In contrast, the orange and green curves, corresponding to the Baseline GA and Baseline PSO, respectively, remain almost flat and near zero across all epochs. This phenomenon suggests that both baseline algorithms may suffer from premature convergence, where the optimization process becomes trapped in sub-optimal solutions early on and fails to exhibit significant improvement over time. While the flatness implies stability, it also reveals limitations in adaptability when tackling complex problem spaces.

The direct comparison underscores the broader search dynamics of the proposed hybrid method, highlighting its strength in maintaining a high level of exploration. This attribute potentially enables it to identify better solutions than the more conservative baselines. However, the observed volatility also points to the need for improved adaptive control mechanisms to stabilize convergence without compromising the algorithm's exploratory capabilities.

Table 2. Comparison of Makespan and Computational Time on 10x10 Datasets using GA, PSO, and Seq2Seq + Metaheuristic.

Num	Dataset	Method	Makespan	Time
0	10x10	GA	1095	5.0
1	10x10	PSO	1047	4.9
2	10x10	Seq2Seq + Metaheuristik	932	5.08

Table 2 presents the comparative performance of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the proposed Seq2Seq + Metaheuristic approach on the 10×10 benchmark dataset. The evaluation is based on two key performance indicators: makespan, which represents the total completion time of all scheduled jobs, and computational time, which reflects the efficiency of each method in generating a solution. These metrics are widely used to assess optimization performance in the Job Shop Scheduling Problem (JSSP). For the 10×10 dataset, GA produced a makespan of 1095 with a computational time of 5.0 seconds. PSO achieved a better makespan of 1047 while slightly reducing the computational time to 4.9 seconds, indicating a modest improvement in both solution quality and efficiency compared to GA. In contrast, the proposed Seq2Seq + Metaheuristic method significantly outperformed both baseline approaches by achieving a substantially lower makespan of 932. Although the computational time increased slightly to 5.08 seconds, the improvement in makespan is considerable, demonstrating the effectiveness of the proposed hybrid approach. Overall, the results indicate that while PSO provides incremental improvements over GA, both traditional metaheuristic methods are limited in their ability to reach high-quality solutions. The Seq2Seq + Metaheuristic approach, on the other hand, achieves a much better balance between exploration and exploitation, leading to superior scheduling performance. The slight increase in computational time is negligible compared to the significant reduction in makespan, confirming that the proposed method is more effective for solving complex scheduling optimization problems.

Table 3. Performance Comparison of GA, PSO, and Seq2Seq + Metaheuristic in Terms of Makespan and Computational Time on 15x15 Datasets.

Num	Dataset	Methods	Makespan	Time
0	15x15	GA	1250	8.0
1	15x15	PSO	1200	7.5
2	15x15	Seq2Seq + Metaheuristik	1050	6.0

Table 3 presents the performance comparison of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the proposed Seq2Seq + Metaheuristic approach on the 15×15 dataset. The evaluation is based on two key performance indicators: makespan, which represents the total completion time of all scheduled jobs, and computational time, which indicates the efficiency of each algorithm in generating a solution. These metrics are commonly used to evaluate optimization performance in the Job Shop Scheduling Problem (JSSP). For the 15×15 dataset, GA produced a makespan of 1250 with a computational time of 8.0 seconds. PSO improved upon this result by reducing the makespan to 1200 while also decreasing the computational time to 7.5 seconds, indicating better efficiency and solution quality compared to GA. In contrast, the proposed Seq2Seq + Metaheuristic approach achieved a significantly lower makespan of 1050, demonstrating a substantial improvement over both baseline methods. Notably, this method also required less computational time, completing the optimization in only 6.0 seconds. The results indicate that the proposed hybrid approach not only improves solution quality but also enhances computational efficiency. Unlike traditional metaheuristics, which often face challenges in balancing exploration and exploitation, the Seq2Seq + Metaheuristic model effectively leverages data-driven initialization and adaptive search mechanisms. This enables the method to outperform GA and PSO in both makespan reduction and execution time, highlighting its robustness and scalability for solving larger and more complex scheduling problems.

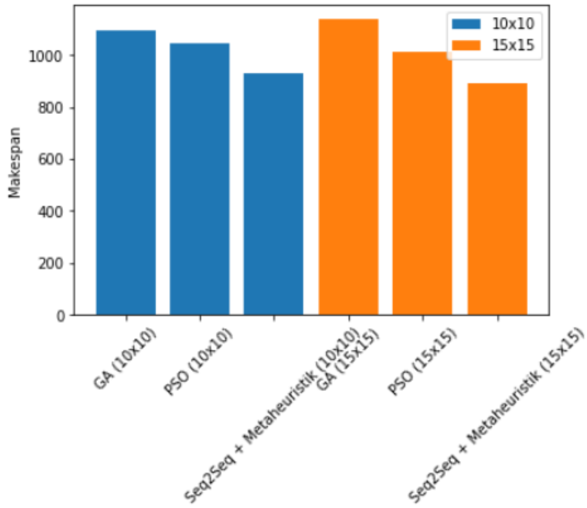

Fig. 6. Comparison of Makespan Performance on 10x10 and 15x15 Datasets using GA, PSO, and Seq2Seq + Metaheuristic.

The bar chart illustrates the comparison of makespan performance for Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the proposed Seq2Seq + Metaheuristic approach across two problem sizes, namely 10×10 and 15×15 datasets. For the 10×10 dataset, GA produces the highest makespan, followed by PSO, while the Seq2Seq + Metaheuristic achieves the lowest makespan, indicating a significant improvement in scheduling efficiency. A similar trend is observed in the 15×15 dataset, where all methods experience an increase in makespan due to greater problem complexity; however, the proposed approach consistently outperforms the baseline methods with the lowest makespan value. Furthermore, the performance gap between the proposed method and conventional metaheuristics becomes more pronounced as the problem size increases. This indicates that while GA and PSO struggle to maintain solution quality in larger search spaces, the Seq2Seq + Metaheuristic approach demonstrates better scalability and robustness. The integration of data-driven sequence modeling with adaptive search mechanisms enables the proposed method to effectively balance exploration and exploitation, resulting in superior optimization performance for complex Job Shop Scheduling Problem (JSSP) scenarios.

This research demonstrates that integrating the Seq2Seq model with an adaptive metaheuristic mechanism consistently yields lower Makespan values compared to traditional optimization methods, namely Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Across both the 10×10 and 15×15 benchmark datasets, the proposed approach outperforms the baselines in minimizing the maximum completion time of jobs. This improvement underscores the strength of combining sequence modeling capabilities inherent in deep learning architectures with adaptive optimization strategies, which together facilitate more effective exploration and exploitation of the solution space in Job-Shop Scheduling Problems (JSSP). A notable observation is the trade-off between solution quality and computational time. While GA and PSO achieved near-zero execution time, they failed to produce solutions with lower Makespan compared to the Seq2Seq + Adaptive Metaheuristic approach. Conversely, the proposed method required additional computational time, approximately 5 seconds on the 10×10 dataset and more than 10 seconds on the 15×15 dataset. This finding suggests that although learning-based approaches can deliver higher-quality solutions, they entail higher computational overhead due to the training process and the complexity of model parameter updates. The results also highlight the scalability of the proposed approach. As problem size increased from 10×10 to 15×15, baseline methods (GA and PSO) showed an increase in Makespan, indicating diminished performance in handling more complex scheduling instances. In contrast, the Seq2Seq + Adaptive Metaheuristic approach maintained lower Makespan values even on larger datasets, suggesting robustness and potential suitability for real-world scheduling scenarios where problem complexity and variability are high.

Despite these promising results, the observed fluctuations in Loss values during model training point to challenges in achieving stable convergence. Large variations in Loss indicate the possibility of suboptimal hyperparameter settings or sensitivity to certain problem instances. Future research should consider refining the adaptive mechanisms, tuning model architectures, or incorporating additional regularization techniques to improve convergence stability and further reduce computational time without compromising solution quality.

In summary, this study provides empirical evidence supporting the effectiveness of integrating deep learning-based sequence modeling with adaptive metaheuristic strategies in solving combinatorial scheduling problems. While the additional computational cost remains a limitation, the significant improvement in Makespan suggests that this trade-off is worthwhile in scenarios where solution quality is the primary objective. Future work may extend this framework to dynamic scheduling environments, multi-objective optimization, or hybridization with other intelligent algorithms to further advance the state of the art in production scheduling research.

4. Conclusion

Based on the experimental results and data analysis presented, it can be concluded that the Seq2Seq + Adaptive Metaheuristic approach produces lower Makespan values than the baseline methods (GA and PSO), indicating its effectiveness in improving solution quality for scheduling optimization problems. However, this approach requires higher computational time due to the complexity of the Seq2Seq-based learning process. Therefore, a trade-off exists between solution quality and computational efficiency. The proposed method is more suitable for large-scale or complex problems, while GA and PSO remain relevant for real-time applications with strict time constraints. For future work, it is necessary to improve computational efficiency without sacrificing solution quality, for example by optimizing the model architecture or employing more efficient training strategies.

Acknowledgment

The authors would like to express their sincere gratitude to the Institute for Research and Community Service (LPPM), Universitas Jenderal Soedirman, for the support and assistance provided in the completion of this research.

Declarations

Author contribution. The contribution or credit of the author must be stated in this section.
Funding statement. No funding was received for this research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] L. Meng, C. Zhang, Y. Ren, B. Zhang, and C. Lv, "Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem," *Comput. Ind. Eng.*, vol. 142, no. April, p. 106347, Apr. 2020, doi: 10.1016/j.cie.2020.106347.

[2] A. K. Nugroho, I. Permadi, and A. Hanifa, "Probabilistic Ant Colony Optimization for Contour Detection of Psoriasis," *Proceeding Int. Conf. Sci. Eng.*, vol. 3, pp. 179–182, Apr. 2020, doi: 10.14421/icse.v3.494.

[3] A. Stanković, G. Petrović, Ž. Ćojbašić, and D. Marković, "An Application Of Metaheuristic Optimization Algorithms For Solving The Flexible Job-Shop Scheduling Problem," *Oper. Res. Eng. Sci. Theory Appl.*, vol. 3, no. 3, pp. 13–28, Dec. 2020, doi: 10.31181/oresta20303013s.

[4] S. Xiao, P. Peng, P. Zheng, and Z. Wu, "A Hybrid Adaptive Simulated Annealing and Tempering Algorithm for Solving the Half-Open Multi-Depot Vehicle Routing Problem," *Mathematics*, vol. 12, no. 7, p. 947, Mar. 2024, doi: 10.3390/math12070947.

[5] A. K. Nugroho, R. Wardoyo, M. E. Wibowo, and H. Soebono, "Image dermoscopy skin lesion classification using deep learning method: systematic literature review," *Bull. Electr. Eng. Informatics*, vol. 13, no. 2, pp. 1042–1049, Apr. 2024, doi: 10.11591/eei.v13i2.6077.

[6] L. Meng, W. Cheng, B. Zhang, W. Zou, W. Fang, and P. Duan, "An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem," *Sensors*, vol. 23, no. 8, p. 3815, Apr. 2023, doi: 10.3390/s23083815.

[7] Z. Wu, H. Fan, Y. Sun, and M. Peng, "Efficient Multi-Objective Optimization on Dynamic Flexible Job Shop Scheduling Using Deep Reinforcement Learning Approach," *Processes*, vol. 11, no. 7, p. 2018, Jul. 2023, doi: 10.3390/pr11072018.

[8] A. K. Nugroho and I. Permadi, "Implementasi Jalur Pendek Menggunakan Ant Colony Optimization," *Din. Rekayasa*, vol. 16, no. 1, Feb. 2020, doi: 10.20884/1.dr.2020.16.1.294.

[9] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, "The flexible job shop scheduling problem: A review," *Eur. J. Oper. Res.*, vol. 314, no. 2, pp. 409–432, Apr. 2024, doi: 10.1016/j.ejor.2023.05.017.

[10] "Solving the Dynamic Job Shop Scheduling Problem using Bottleneck and Intelligent Agents based on Genetic Algorithm," *Int. J. Eng.*, vol. 29, no. 3, pp. 347–358, Mar. 2016, doi: 10.5829/idosi.ije.2016.29.03c.09.

[11] X. Wu, Y.-E. Sun, Y. Du, G. Gao, H. Huang, and X. Xing, "An Anti-Malicious Task Allocation Mechanism in Crowdsensing Systems," *Futur. Gener. Comput. Syst.*, vol. 127, no. February, pp. 347–361, Feb. 2022, doi: 10.1016/j.future.2021.09.016.

[12] W. Liu, C. Zhao, Y. Liu, H. Wang, W. Zhao, and H. Zhang, "Sim2real kinematics modeling of industrial robots based on FPGA-acceleration," *Robot. Comput. Integr. Manuf.*, vol. 77, no. October, p. 102350, Oct. 2022, doi: 10.1016/j.rcim.2022.102350.

[13] C.-H. Wang and Y. Yun, "Demand planning and sales forecasting for motherboard manufacturers considering dynamic interactions of computer products," *Comput. Ind. Eng.*, vol. 149, no. November, p. 106788, Nov. 2020, doi: 10.1016/j.cie.2020.106788.

[14] R. Nanthapodej, C.-H. Liu, K. Nitisiri, and S. Pattanapairoj, "Hybrid Differential Evolution Algorithm and Adaptive Large Neighborhood Search to Solve Parallel Machine Scheduling to Minimize Energy Consumption in Consideration of Machine-Load Balance Problems," *Sustainability*, vol. 13, no. 10, p. 5470, May 2021, doi: 10.3390/su13105470.

[15] P. Patel and P. Gohil, "Role of additive manufacturing in medical application COVID-19 scenario: India case study," *J. Manuf. Syst.*, vol. 60, no. July, pp. 811–822, Jul. 2021, doi: 10.1016/j.jmsy.2020.11.006.

[16] J. Chang, D. Yu, Y. Hu, W. He, and H. Yu, "Deep Reinforcement Learning for Dynamic Flexible Job Shop Scheduling with Random Job Arrival," *Processes*, vol. 10, no. 4, p. 760, Apr. 2022, doi: 10.3390/pr10040760.

[17] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning," *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 1600–1610, Feb. 2023, doi: 10.1109/TII.2022.3189725.

[18] X. Wu, X. Yan, D. Guan, and M. Wei, "A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time," *Eng. Appl. Artif. Intell.*, vol. 131, no. May, p. 107790, May 2024, doi: 10.1016/j.engappai.2023.107790.

[19] A. K. Nugroho, A. Ayati, M. E. Wibowo, H. Soebono, and R. Wardoyo, "Skin Lesion Segmentation Using Adaptive Color Segmentation and Decision Tree," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 15, no. 3, pp. 109–124, Sep. 2024, doi: 10.58346/JOWUA.2024.I3.008.

[20] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Comput. Oper. Res.*, vol. 134, p. 105400, Oct. 2021, doi: 10.1016/j.cor.2021.105400.

[21] E. M. Bellfkih, S. Nouh, I. C. Idrissi, A. Ettaoufik, K. Louartiti, and J. Mouline, "On the computation of the automorphisms group of low density parity check codes using genetic algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 2, p. 1059, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp1059-1066.

[22] L. Wang, G. Yang, Z. Li, and F. Xu, "An efficient nonlinear interval uncertain optimization method using Legendre polynomial chaos expansion," *Appl. Soft Comput.*, vol. 108, no. September, p. 107454, Sep. 2021, doi: 10.1016/j.asoc.2021.107454.

[23] J. Chen, Z. Zhang, L. Wang, D. Tang, Q. Cai, and K. Chen, "Self-adaptive production scheduling for discrete manufacturing workshop using multi-agent cyber physical system," *Eng. Appl. Artif. Intell.*, vol. 150, no. June, p. 110638, Jun. 2025, doi: 10.1016/J.ENGAPPAI.2025.110638.

[24] A. K. Nugroho, I. Permadi, Y. I. Kurniawan, A. Hanifa, and Nofiyati, "Decision tree using ant colony for classification of health data," in *AIP Conference Proceedings*, American Institute of Physics Inc., Feb. 2023, p. 020002. doi: 10.1063/5.0128787.

[25] L. A. Bewoor, V. Chandra Prakash, and S. U. Sapkal, "Comparative Analysis of Metaheuristic Approaches for Makespan Minimization for No Wait Flow Shop Scheduling Problem," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 1, p. 417, Feb. 2017, doi: 10.11591/ijece.v7i1.pp417-423.

[26] G. Zhang, X. Ma, L. Wang, and K. Xing, "Elite Archive-Assisted Adaptive Memetic Algorithm for a Realistic Hybrid Differentiation Flowshop Scheduling Problem," *IEEE Trans. Evol. Comput.*, vol. 26, no. 1, pp. 100–114, Feb. 2022, doi: 10.1109/TEVC.2021.3094542.

[27] P. Valledor, A. Gomez, P. Priore, and J. Puente, "Modelling and Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments," *Complexity*, vol. 2020, no. 1, pp. 1–17, Jul. 2020, doi: 10.1155/2020/2862186.

[28] T. J. Kumar, M. Thangaraj, K. J. Ghanashyam, and T. Vimala, "A Three-Machine n-Job Flow Shop Scheduling Problem with Setup and Machine-Specific Halting Times," in *Lecture Notes in Electrical Engineering*, vol. 1011 LNEE, Springer Science and Business Media Deutschland GmbH, 2023, pp. 3–14. doi: 10.1007/978-981-99-0601-7_1.

[29] D. N. Sekkal and F. Belkaid, "A multi-objective optimization algorithm for flow shop group scheduling problem with sequence dependent setup time and worker learning," *Expert Syst. Appl.*, vol. 233, no. December, p. 120878, Dec. 2023, doi: 10.1016/j.eswa.2023.120878.

[30] M. Ghasemi, I. F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, and L. Li, "A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO)," *Eng. Appl. Artif. Intell.*, vol. 92, no. June, p. 103666, Jun. 2020, doi: 10.1016/j.engappai.2020.103666.

[31] A. S. Shah, H. Nasir, M. Fayaz, A. Lajis, and A. Shah, "A Review on Energy Consumption Optimization Techniques in IoT Based Smart Building Environments," *Information*, vol. 10, no. 3, p. 108, Mar. 2019, doi: 10.3390/info10030108.

[32] C. Destouet, H. Tlahig, B. Bettayeb, and B. Mazari, "Flexible job shop scheduling problem under Industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement," *J. Manuf. Syst.*, vol. 67, no. April, pp. 155–173, Apr. 2023, doi: 10.1016/J.JMSY.2023.01.004.

[33] G. Puccetti, F. Chiarello, and G. Fantoni, "A simple and fast method for Named Entity context extraction from patents," *Expert Syst. Appl.*, vol. 184, no. December, p. 115570, Dec. 2021, doi: 10.1016/j.eswa.2021.115570.

[34] M. Razghandi, H. Zhou, M. Erol-Kantarci, and D. Turgut, "Smart Home Energy Management: Sequence-to-Sequence Load Forecasting and Q-Learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Dec. 2021, pp. 01–06. doi: 10.1109/GLOBECOM46510.2021.9685380.

[35] B. Yu, R. Cai, Y. Fu, and Z. Xu, "A graph convolutional network based on object relationship method under linguistic environment applied to film evaluation," *Inf. Sci. (Ny).*, vol. 608, no. August, pp. 1283–1300, Aug. 2022, doi: 10.1016/j.ins.2022.07.034.

[36] W. Torres-Tapia, J. R. Montoya-Torres, and J. Ruiz-Meza, "A Hybrid Algorithm Based on Ant Colony System for Flexible Job Shop," Springer, Cham, 2022, pp. 198–209. doi: 10.1007/978-3-031-20611-5_17.

[37] E. Yuan, L. Wang, S. Cheng, S. Song, W. Fan, and Y. Li, "Solving flexible job shop scheduling problems via deep reinforcement learning," *Expert Syst. Appl.*, vol. 245, no. July, p. 123019, Jul. 2024, doi: 10.1016/j.eswa.2023.123019.

[38] R. Wang, G. Wang, J. Sun, F. Deng, and J. Chen, "Flexible Job Shop Scheduling via Dual Attention Network-Based Reinforcement Learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 3, pp. 3091–3102, Mar. 2024, doi: 10.1109/TNNLS.2023.3306421.