JOURNAL OF MODERN MANUFACTURING SYSTEMS AND TECHNOLOGY
E-ISSN: 2636-9575
VOLUME 8, ISSUE 2, SEPT, 2024, 12 – 32
DOI: https://doi.org/10.15282/jmmst.v8i2.10762

RESEARCH ARTICLE

A Whale Optimization Algorithm Approach for Flow Shop Scheduling to Minimize Makespan

MAH Osman 1[1], MFF Ab. Rashid 2[2]*, MAN Mutasim[2]

[1] Faculty of Manufacturing and Mechatronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600, Pekan, Pahang, Malaysia.

[2] Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600, Pekan, Pahang, Malaysia.

ABSTRACT - Flow shop scheduling is crucial in manufacturing and production environments because it directly impacts output and overall production efficiency. It involves processing a set of jobs on multiple machines in a specific order. The objective is to determine the optimal job sequence that minimizes the makespan, which is the total time required to complete all jobs. This study proposes a computerized approach utilizing the Whale Optimization Algorithm (WOA) to solve the flow shop scheduling problem and minimize the makespan. The WOA is a recently developed meta-heuristic algorithm inspired by the bubble-net hunting strategy of humpback whales. The performance of the WOA is evaluated using five benchmark problems with varying numbers of jobs and machines, and the results are compared with those obtained from other algorithms reported in the literature, such as genetic algorithms and heuristic models. The findings demonstrate that the WOA can effectively solve the flow shop scheduling problem and provide improved makespan values, with an average efficiency of 7.33% compared to the other algorithms.

ARTICLE HISTORY

Received	:	25th May 2024
Revised	:	01st July 2024
Accepted	:	09th July 2024
Published	:	30th Sept. 2024

KEYWORDS

Flow shop scheduling
Scheduling optimization
Whale optimization algorithm
Metaheuristics

1.0 INTRODUCTION

In manufacturing and production environments, efficient scheduling of jobs on available machines is crucial for optimizing resource utilization and maximizing productivity. Flow shop scheduling is a widely studied problem in this context, where a set of jobs must be processed on multiple machines in a specific order [1]. The objective is to determine the optimal job sequence that minimizes the makespan, which is the total time required to complete all jobs [2].

Traditionally, flow shop scheduling problems have been addressed using various optimization techniques, including mathematical programming [3], heuristic methods [4], and metaheuristic algorithms [5]. However, as the problem size increases with the number of jobs and machines, the computational complexity escalates, making it challenging to find optimal solutions within a reasonable time frame [6].

In recent years, meta-heuristic algorithms inspired by natural phenomena have gained significant attention for solving complex optimization problems[7]. One such algorithm is the Whale Optimization Algorithm (WOA), which is based on the bubble-net hunting strategy of humpback whales. The WOA has demonstrated promising results in solving various optimization problems and has the potential to be applied to the flow shop scheduling problem[8].

This study aims to develop a computerized approach using the WOA to solve the flow shop scheduling problem and minimize the makespan. The performance of the proposed approach is evaluated using five benchmark problems with varying numbers of jobs and machines, and the results are compared with those obtained from other algorithms reported in the literature, such as genetic algorithms and heuristic models.

1.1 Related Works

Flow shop scheduling has been extensively studied in the literature, and various optimization techniques have been proposed to solve this problem. Traditional methods include mathematical programming techniques, such as branch-and-bound and dynamic programming, which guarantee optimal solutions but are computationally expensive for large problem instances [9].

Heuristic methods, such as the Palmer's Heuristic Model and the Gupta Algorithm, have also been widely used for flow shop scheduling. These methods provide approximate solutions in a reasonable amount of time but do not guarantee optimality [10].

In recent years, metaheuristic algorithms have gained popularity for solving complex optimization problems, including flow shop scheduling. These algorithms are inspired by natural phenomena and can effectively explore the search space while avoiding being trapped in local optima[7].

***CORRESPONDING AUTHOR | M.M.F. Ab. Rashid |** ffaisae@umpsa.edu.my

Genetic algorithms (GAs) are one of the most widely used metaheuristic algorithms for flow shop scheduling. GAs mimic the process of natural selection and evolution, where potential solutions are represented as chromosomes, and genetic operators like crossover and mutation are applied to generate new solutions. Several studies have reported promising results using GAs for flow shop scheduling ([11], [12], [13]).

Another metaheuristic algorithm that has been applied to flow shop scheduling is Particle Swarm Optimization (PSO). PSO is inspired by the social behavior of bird flocking or fish schooling, where particles in the swarm move towards the best-known position based on their own experience and the experience of the entire swarm. Studies by [14], [15]have demonstrated the effectiveness of PSO in solving flow shop scheduling problems.

The Whale Optimization Algorithm (WOA) is a relatively new metaheuristic algorithm proposed by Mirjalili and Lewis in 2016[16]. The WOA is inspired by the bubble-net hunting strategy of humpback whales, where whales create unique spiral-shaped bubbles to encircle and capture their prey. The algorithm mimics this behavior to explore the search space and converge towards the optimal solution [17].

The creators of WOA have evaluated the algorithm on 29 mathematical optimization problems and 6 structural optimization problems, and the results showed that the WOA performed well in terms of exploration and exploitation capabilities, as well as escaping local minima [16]. According to the obtained results, the WOA ranked first or second in six out of seven test functions for exploitation capability [16]. Additionally, the algorithm outperformed comparison algorithms in 86% of the cases in the exploration test function [18].

The previous researcher also presents a novel empirical analysis of the WOA, focusing on its balance between the exploration and exploitation phases [19]. It employs dimension-wise diversity measurement to evaluate the population's convergence and diversity throughout the optimization process. In addition, the previous researchers also present a multi-strategy mechanism to address WOA's balance in exploration and exploitation [20]. Experimental results show that the WOA outperforms several other algorithms, improving convergence and avoiding local optima on the CEC2017 benchmark suite. However, the application of the WOA to flow shop scheduling problems has not been extensively explored in the literature.

This paper applies the WOA to optimize the makespan in the flow shop scheduling problem, aiming to improve efficiency and reduce completion time. Section 2 provides a detailed explanation of the flow shop scheduling problem, outlining its significance and challenges. Section 3 describes the WOA algorithm, including its key principles and implementation steps. Section 4 presents the results of the optimization process, offering a comprehensive analysis and discussion of the findings. Finally, Section 5 concludes the study by summarizing the key outcomes and suggesting potential areas for future research.

2.0 FLOW SHOP SCHEDULING PROBLEM

The flow shop scheduling problem can be formulated as follows: Consider a set of n jobs, denoted as J = {j1, j2, ..., jn}, and a set of m machines, denoted as M = {M1, M2, ..., Mm}. Each job Ji must be processed on all m machines in the same order, starting from machine M1, then M2, and so on until machine Mm. The example of a flow shop layout for four jobs and three machines is presented in Figure 1.

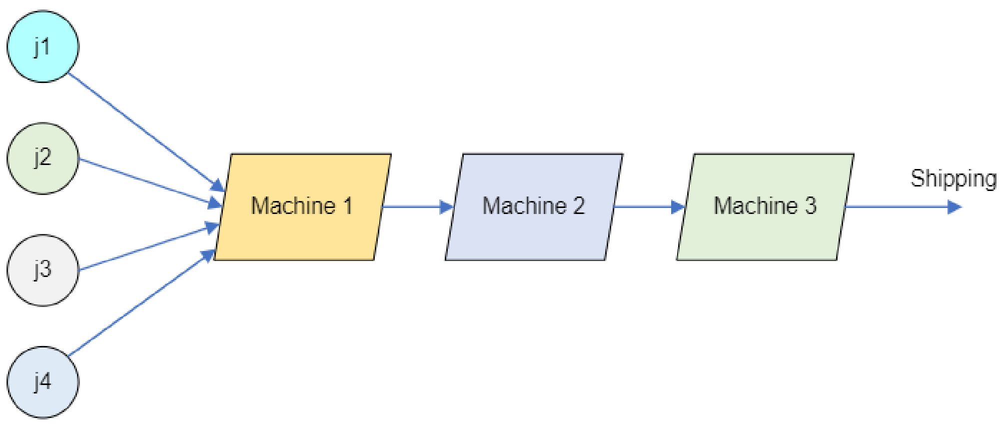

Figure 1: Flow Shop Scheduling Layout (4 jobs and 3 machines)

The objective of the flow shop scheduling problem is to find an optimal job sequence that minimizes the makespan, which is defined as the total time required to complete all jobs on all machines. In other words, the makespan represents the time elapsed from the start of the first job on the first machine until the completion of the last job on the last machine.

The makespan can be calculated using the following equation: Makespan = $\max\{C_{n1}, C_{n2}, ..., C_{nm}\}$

where C_{nk} represents the completion time of the last job on machine M_k. Note that M_k is the k^{th} machine in the stage. The completion time of a job on a machine depends on the processing time of that job on the machine and the completion time of the previous job on the same machine.

The flow shop scheduling problem is subject to several constraints. Firstly, each machine can process only one job at a time, meaning that a machine cannot start processing a new job until it has completed the previous job. Secondly, all jobs must follow the same machine order, ensuring that each job is processed on the machines in the sequence of M_1, M_2, until M_m. Finally, once a job starts processing on a machine, it cannot be interrupted until it is completed on that machine.

The goal is to determine the optimal job sequence that satisfies these constraints while minimizing the makespan. This optimization problem becomes increasingly complex as the number of jobs and machines increases, making it difficult to find the optimal solution using manual or exhaustive methods. Therefore, efficient optimization algorithms, such as the Whale Optimization Algorithm (WOA) proposed in this study, are required to solve the flow shop scheduling problem effectively.

3.0 WHALE OPTIMIZATION ALGORITHM

The proposed methodology for solving the flow shop scheduling problem using the Whale Optimization Algorithm (WOA) involves several key steps, as shown in Figure 2. Firstly, the initialization phase takes place, where the WOA starts by generating a population of random solutions (job sequences) within the search space. The population size and the maximum number of iterations are set as input parameters. These parameters can be adjusted to balance exploration and exploitation capabilities, as well as computational efficiency.

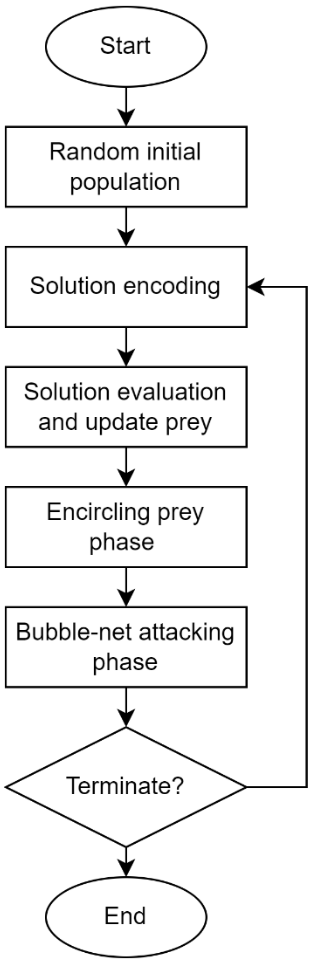

Figure 2: Flow chart of WOA

Next, the problem is encoded by representing each potential solution in the WOA as a job sequence, with each solution encoded as a vector of job indices. For example, a solution [3, 1, 2, 4] represents the job sequence where job 3 is processed first, followed by jobs 1, 2, and 4.

The fitness evaluation step is then carried out for each solution in the population. The makespan is calculated based on the given job sequence and processing times, where the makespan serves as the fitness value for that solution. Lower makespan values represent better solutions, as the objective is to minimize the total time required to complete all jobs.

The core of the methodology lies in the Whale Optimization Algorithm itself, which consists of several key phases inspired by the behavior of humpback whales. The encircling prey phase simulates the behavior of humpback whales encircling their prey. In this phase, the current best solution in the population is considered the "prey," and the other solutions in the population (whales) update their positions to encircle the prey, exploring the search space around the best-known solution.

Additionally, the WOA incorporates a bubble-net attacking phase, which is inspired by the bubble-net hunting strategy of humpback whales. This phase involves a spiral movement pattern, allowing the whales (solutions) to exploit the search space more effectively and potentially escape local optima. The WOA alternates between the encircling and bubble-net attacking phases, effectively balancing exploration and exploitation capabilities.

The algorithm iterates until a specified termination criterion is met, such as a maximum number of iterations or a satisfactory solution quality. During each iteration, the solutions in the population are updated based on the encircling and bubble-net attacking phases, continuously improving the quality of the solutions and converging towards the global optimum.

Finally, once the termination criterion is met, the job sequence corresponding to the best solution found by the WOA is reported as the optimum solution, along with its associated makespan value. This optimum solution represents the job sequence that minimizes the makespan, satisfying the objective of the flow shop scheduling problem.

4.0 COMPUTATIONAL EXPERIMENTS AND RESULTS

The proposed Whale Optimization Algorithm (WOA) approach was applied to five benchmark flow shop scheduling problems with varying numbers of jobs and machines. The performance of the WOA was evaluated by comparing the obtained makespan values with those reported in the literature for other algorithms, such as genetic algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Differential Evolution (DE) algorithms.

The experiment has been conducted using five benchmark flow shop scheduling problems from the following sources:

Table 1: Test Problems for Flow Shop Scheduling

Problem No.	Problem Size	Source
1	5 jobs and 3 machines	[21]
2	5 jobs and 4 machines	[22]
3	8 jobs and 3 machines	[23]
4	10 jobs and 8 machines	[24]
5	10 jobs and 10 machines	[25]

Optimization was conducted with 10 repetition runs, 30 population sizes and 300 iterations. Table 2 presents the results in terms of average fitness, minimum fitness and standard deviation.

Table 2: Computational Experiment Results using Metaheuristic Algorithms (in minutes)

Problem	Indicator	GA	PSO	ACO	DE	WOA
1	Average	31.8	32.8	32.7	30	28.5
	Minimum	28	31	30	27	25
	Std Dev	2.30	1.03	2.06	2.62	2.68

2	Average	74.2	78.8	79.4	78.8	75.4
	Minimum	74	74	74	75	74
	Std Dev	1.87	2.97	3.17	3.05	1.65
3	Average	47.1	45.5	47.3	47.1	43.3
	Minimum	45	43	46	45	41
	Std Dev	1.85	1.90	0.95	1.37	2.21
4	Average	98.3	99.2	102.1	100	93.5
	Minimum	94	92	93	91	88
	Std Dev	2.91	3.19	6.31	4.32	4.62
5	Average	107.4	102.6	110	108.8	96.9
	Minimum	102	96	101	96	91
	Std Dev	4.43	3.72	7.27	8.08	5.02

For Problem 1, the fitness values show a clear distinction among the algorithms. The WOA outperforms all others with a fitness value of 28.5, indicating the most efficient solution. DE follows with a fitness of 30, also demonstrating strong performance. GA, PSO and ACO have slightly higher fitness values of 31.8, 32.8, and 32.7, respectively, suggesting that while they are competitive, they are less effective than WOA and DE for this problem. The best schedule form WOA is shown in Figure 3. The optimum job sequence for this problem is {J4, J1, J3, J2 and J5}.

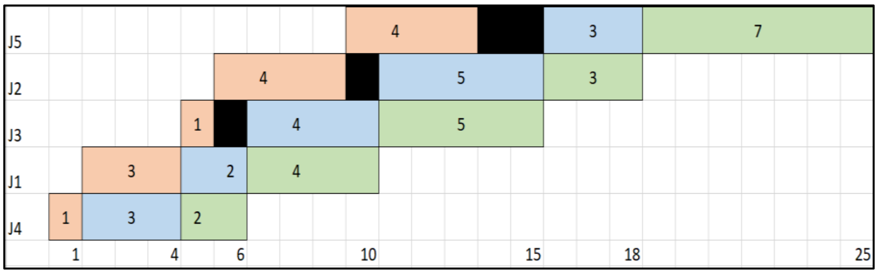

Figure 3: Best Schedule for Problem 1

In Problem 2, GA shows its strength with the lowest fitness value of 74.2, followed by WOA. The DE also performs well with a fitness value of 78.8, indicating it is effective but not as optimal as GA. PSO and ACO have similar fitness values of 78.8 and 79.4, respectively, with ACO being slightly less effective. This problem highlights GA's efficiency and the comparable performance of WOA, DE, and PSO. The best production schedule from GA is shown in Figure 4, whereas the optimum job processing sequence is {J2, J1, J4, J5 and J3}.

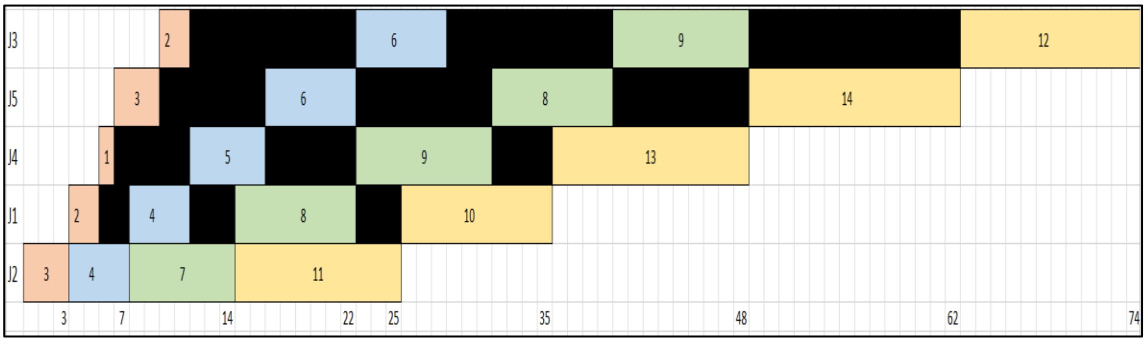

Figure 4: Best Schedule for Problem 2

Problem 3 presents a different scenario where WOA leads with a fitness value of 43.3, showing its consistent superiority. PSO follows closely with a fitness value of 45.5, indicating good performance. Both GA and DE achieve the same fitness value of 47.1, suggesting that they are similarly effective for this problem. ACO is slightly less optimal with a fitness value of 47.3. This problem demonstrates WOA's continued effectiveness and the solid performance of PSO, GA, and DE. The optimum job processing sequence for this problem 3 is {J8, J2, J7, J3, J6, J5, J4 and J1}. The obtained schedule for Problem 3 is depicted in Figure 5.

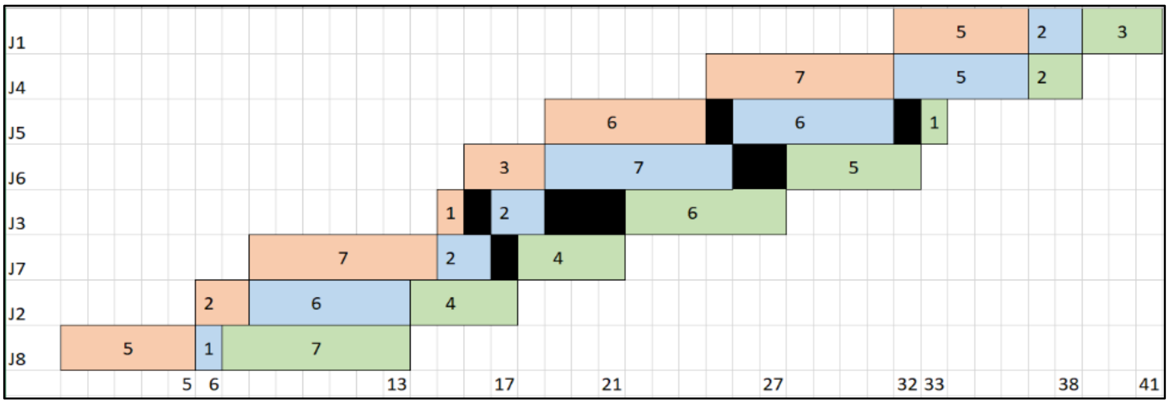

Figure 5: Best Schedule for Problem 3

For Problem 4, WOA maintains its lead with a fitness value of 93.5, indicating it produces the best solution (Figure 6). This fitness value comes from the job sequence of {J9, J6, J10, J8, J4, J5, J3, J7, J1 and J2}. GA and PSO follow with higher fitness values of 98.3 and 99.2, respectively, showing they are effective but less optimal than WOA. DE and ACO have fitness values of 100 and 102.1, respectively, with ACO again being the least effective. This problem further establishes WOA's dominance and the competitiveness of GA and PSO.

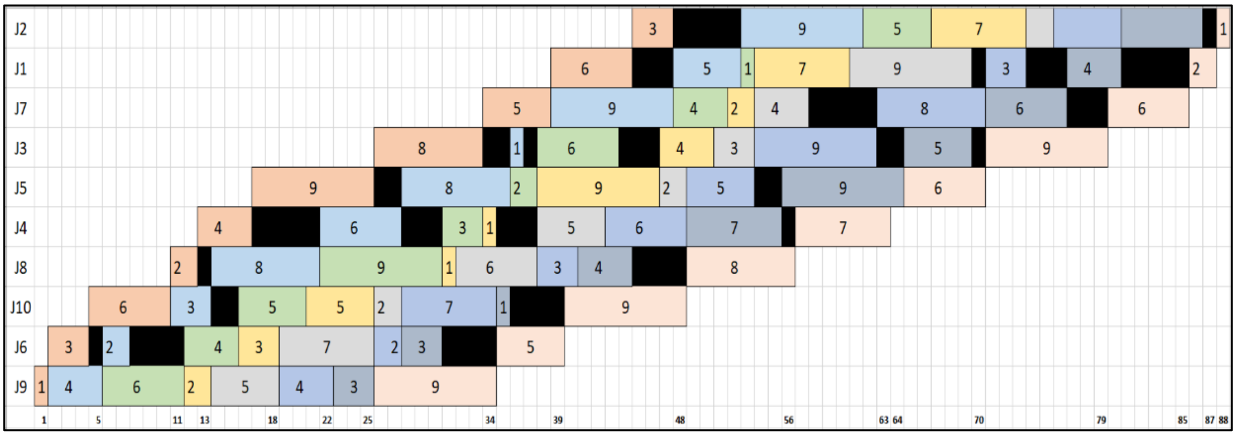

Figure 6: Best Schedule for Problem 4

In Problem 5, WOA is again the most effective algorithm with a fitness value of 96.9 as in Figure 6. PSO and DE follow with fitness values of 102.6 and 108.8, respectively, indicating good but less optimal performance. GA and ACO have higher fitness values of 107.4 and 110, respectively, with ACO being the least effective. This problem continues the trend of WOA's superiority and the relative effectiveness of PSO and DE. The optimum job sequence for Problem 5 is { J3, J1, J2, J6, J5, J10, J9, J8, J7, and J4}.

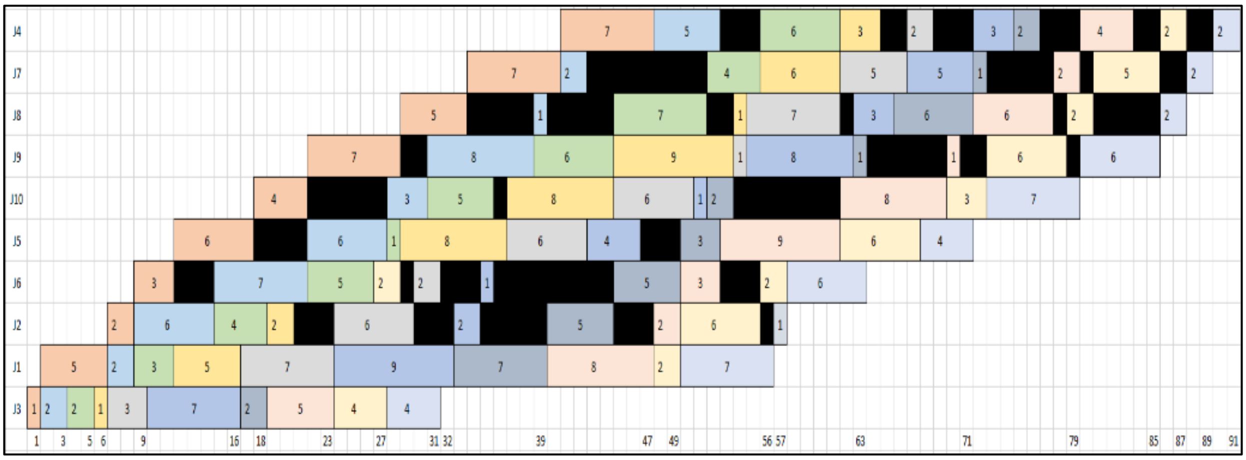

Figure 7: Best Schedule for Problem 5

Across all problems, the WOA consistently delivers the best performance with the lowest average fitness values, demonstrating its superior capability in finding optimal solutions. PSO and DE also perform well, showing strong and consistent results across most problems. GA and ACO, while useful, tend to be less effective compared to WOA, PSO, and DE, with ACO often having the highest fitness values and thus the least effective solutions in this comparison.

The analysis of minimum fitness results across five distinct optimization problems highlights WOA as consistently outperforming other optimization techniques. WOA consistently yielded the lowest minimum fitness scores across the different problem instances, suggesting its effectiveness in finding optimal solutions. This consistency underscores WOA's robustness and reliability in tackling various optimization challenges.

In terms of standard deviation, PSO tends to show the most consistent performance with lower standard deviations in most problems, indicating its reliability in producing stable optimization results. GA also demonstrates good consistency across several problems. ACO shows highly consistent results for some problems but significantly higher variability in others. DE and WOA exhibit more variability in their outcomes, suggesting that while they may produce good results, their performance can be less predictable. This analysis highlights the importance of considering both the quality of optimization results and the consistency of algorithm performance when selecting an optimization method.

Figure 8 shows the average convergence plot for the problem. This study's results highlight the WOA's potential as a promising metaheuristic algorithm for solving flow shop scheduling problems. By balancing exploration and exploitation capabilities effectively, the WOA can navigate the search space and converge towards optimal or near-optimal solutions, outperforming traditional heuristic methods and improving makespan values.

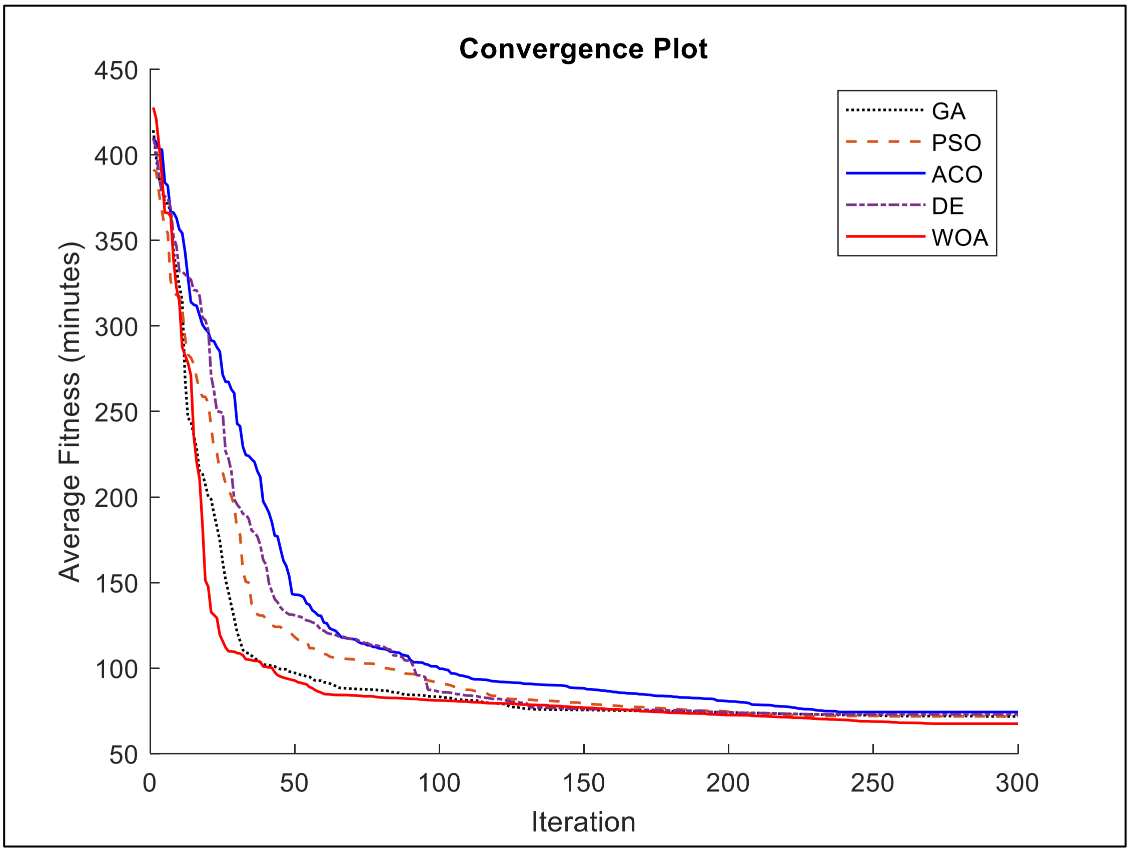

Figure 8: Convergence Plot of Flow Shop Scheduling Optimization

The proposed Whale Optimization Algorithm (WOA) approach demonstrated promising results in solving the flow shop scheduling problem across the five benchmark problems evaluated. Compared to other algorithms, the WOA consistently provided improved makespan values, indicating its effectiveness in minimizing the total time required to complete all jobs.

When analyzing the individual problem instances, it is evident that the WOA's performance varied depending on the complexity of the data set and the number of jobs and machines involved. For simpler problems with fewer jobs and machines, such as the second problem instance with 5 jobs and 4 machines, the WOA could not outperform the genetic algorithm. However, as the problem size increased, with more jobs and machines, the WOA demonstrated its capability to find better makespan values than the heuristic models employed in the reference sources.

Remarkably, the WOA achieved significant improvements in the makespan for the larger problem instances, such as the fourth problem with 10 jobs and 8 machines, and the fifth problem with 10 jobs and 10 machines. This highlights the WOA's ability to effectively navigate the search space and converge towards optimal or near-optimal solutions, even in complex scenarios with larger numbers of jobs and machines.

Overall, the results of this study demonstrate the potential of the WOA as a promising metaheuristic algorithm for solving flow shop scheduling problems. By effectively balancing exploration and exploitation capabilities, the WOA can navigate the search space and converge towards optimal or near-optimal solutions, outperforming traditional heuristic methods and providing improved makespan values. The average efficiency improvement of 5.559% across all five benchmark problems further solidifies the WOA's applicability and potential for practical implementation in manufacturing and production environments.

5.0 CONCLUSIONS

This study investigated the application of the Whale Optimization Algorithm (WOA), a recently developed metaheuristic algorithm, to the flow shop scheduling problem. The objective was to determine the optimal job sequence that minimizes the makespan, which is the total time required to complete all jobs on all machines. The performance of the proposed WOA approach was evaluated using five benchmark problems with varying numbers of jobs and machines, and the results were compared with those obtained from other algorithms.

The findings demonstrate that the WOA is an effective optimization technique for solving flow shop scheduling problems. Across the benchmark problems, the WOA provided improved makespan values compared to existing algorithms for four out of the five problem instances. The average efficiency improvement achieved by the WOA was 7.33%, indicating its potential to enhance productivity and resource utilization in manufacturing and production environments.

Furthermore, the WOA exhibited excellent convergence behavior, particularly for larger problem instances with more jobs and machines. Despite the increased complexity, the algorithm converged to optimal or near-optimal solutions within a reasonable number of iterations, showcasing its efficiency and robustness.

While the results of this study are encouraging, there is still room for further research and improvements. Future work could explore the integration of the WOA with other optimization techniques or the development of hybrid approaches to enhance its performance further. Additionally, the application of the WOA to other variants of the flow shop scheduling problem, such as permutation flow shops or flexible flow shops, could be investigated.

Overall, this study contributes to the field of production scheduling by introducing a novel metaheuristic approach based on the WOA for solving the flow shop scheduling problem. The findings provide valuable insights into the algorithm's effectiveness and potential for practical implementation in manufacturing and production environments, where efficient scheduling is crucial for maximizing productivity and minimizing costs.

ACKNOWLEDGEMENT

The authors would like to acknowledge Universiti Malaysia Pahang Al-Sultan Abdullah for funding this research under the UMP Grant RDU223017.

7.0 REFERENCES

[1] M. Khatami, A. Salehipour, and T. C. E. Cheng, “Flow-shop scheduling with exact delays to minimize makespan,” *Comput Ind Eng*, vol. 183, p. 109456, 2023, doi: https://doi.org/10.1016/j.cie.2023.109456.

[2] X. Chen, Q. Miao, B. M. T. Lin, M. Sterna, and J. Blazewicz, “Two-machine flow shop scheduling with a common due date to maximize total early work,” *Eur J Oper Res*, vol. 300, no. 2, pp. 504–511, 2022, doi: https://doi.org/10.1016/j.ejor.2021.07.055.

[3] C. H. Lim and S. K. Moon, “A Two-Phase Iterative Mathematical Programming-Based Heuristic for a Flexible Job Shop Scheduling Problem with Transportation,” *Applied Sciences*, vol. 13, no. 8. 2023. doi: 10.3390/app13085215.

[4] W. Shao, Z. Shao, and D. Pi, “Effective constructive heuristics for distributed no-wait flexible flow shop scheduling problem,” *Comput Oper Res*, vol. 136, p. 105482, 2021, doi: https://doi.org/10.1016/j.cor.2021.105482.

[5] Ö. Tosun, M. K. Marichelvam, and N. Tosun, “A literature review on hybrid flow shop scheduling,” *International Journal of Advanced Operations Management*, vol. 12, no. 2, pp. 156–194, Jan. 2020, doi: 10.1504/IJAOM.2020.108263.

[6] J. K. Lenstra, V. A. Strusevich, and M. Vlach, “A historical note on the complexity of scheduling problems,” *Operations Research Letters*, vol. 51, no. 1, pp. 1–2, 2023, doi: https://doi.org/10.1016/j.orl.2022.11.006.

[7] B. Toaza and D. Esztergár-Kiss, “A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems,” *Appl Soft Comput*, vol. 148, p. 110908, 2023, doi: https://doi.org/10.1016/j.asoc.2023.110908.

[8] M. Abd Elaziz, S. Lu, and S. He, “A multi-leader whale optimization algorithm for global optimization and image segmentation,” *Expert Syst Appl*, vol. 175, p. 114841, 2021, doi: https://doi.org/10.1016/j.eswa.2021.114841.

[9] J. S. Neufeld, S. Schulz, and U. Buscher, “A systematic review of multi-objective hybrid flow shop scheduling,” *Eur J Oper Res*, vol. 309, no. 1, pp. 1–23, 2023, doi: https://doi.org/10.1016/j.ejor.2022.08.009.

[10] M. Nageswara Rao, K. Prakash Babu, K. K. Dama, S. K. Malyala, and P. K. Chaurasiya, “Flexible Manufacturing System Simultaneous Scheduling Through Palmer Heuristic Algorithm BT - Technology Innovation in Mechanical Engineering: Select Proceedings of TIME 2021,” P. K. Chaurasiya, A. Singh, T. N. Verma, and U. Rajak, Eds., Singapore: Springer Nature Singapore, 2022, pp. 1013–1021. doi: 10.1007/978-981-16-7909-4_94.

[11] S. Schulz, “A genetic algorithm to solve the hybrid flow shop scheduling problem with subcontracting options and energy cost consideration,” *Advances in Intelligent Systems and Computing*, vol. 854, pp. 263–273, 2019, doi: 10.1007/978-3-319-99993-7_23.

[12] D. Li, X. Meng, Q. Liang, and J. Zhao, “A heuristic-search genetic algorithm for multi-stage hybrid flow shop scheduling with single processing machines and batch processing machines,” *J Intell Manuf*, vol. 26, no. 5, pp. 873–890, Feb. 2014, doi: 10.1007/s10845-014-0874-y.

[13] J. Yan, J. Wen, and L. Li, “Genetic algorithm based optimization for energy-aware hybrid flow shop scheduling,” in *Proceedings of the 2014 International Conference on Artificial Intelligence, ICAI 2014 - WORLDCOMP 2014*, 2014, pp. 358–364. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84953357421&partnerID=40&md5=85a27447b9e56d9082071b3ea538214c

[14] M. Eddaly, B. Jarboui, and P. Siarry, “Combinatorial Particle Swarm Optimization for solving Blocking Flowshop Scheduling Problem,” *J Comput Des Eng*, 2016, doi: 10.1016/j.jcde.2016.05.001.

[15] C.-J. Liao, Chao-Tang Tseng, and P. Luarn, “A discrete version of particle swarm optimization for flowshop scheduling problems,” *Comput Oper Res*, vol. 34, no. 10, pp. 3099–3111, 2007, Accessed: Jan. 10, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054805003643

[16] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[17] M. S. Uzer and O. Inan, “Application of improved hybrid whale optimization algorithm to optimization problems,” *Neural Comput Appl*, vol. 35, no. 17, pp. 12433–12451, 2023, doi: 10.1007/s00521-023-08370-x.

[18] M. H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili, “A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations,” *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4113–4159, 2023, doi: 10.1007/s11831-023-09928-7.

[19] A. M. Ahmed *et al.*, “Balancing exploration and exploitation phases in whale optimization algorithm: an insightful and empirical analysis,” in *Handbook of Whale Optimization Algorithm*, S. B. T.-H. of W. O. A. Mirjalili, Ed., Academic Press, 2024, pp. 149–156. doi: https://doi.org/10.1016/B978-0-32-395365-8.00017-8.

[20] M. Li, X. Yu, B. Fu, and X. Wang, “A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems,” *Neural Comput Appl*, 2023, doi: 10.1007/s00521-023-08287-5.

[21] I. A. Chaudhry and A. M. Khan, “Minimizing makespan for a no-wait flowshop using genetic algorithm,” *Sadhana*, vol. 37, pp. 695–707, 2012.

[22] I. A. Chaudhry, I. A. Q. Elbadawi, M. Usman, and M. Tajammal Chugtai, “Minimising total flowtime in a no-wait flow shop (NWFS) using genetic algorithms,” *Ingenier{\'\i}a e Investigación*, vol. 38, no. 3, pp. 68–79, 2018.

[23] A. K. Agarwal and N. Kumar, “Flow Shop Scheduling Problem For 8-Jobs, 3-Machines With Make Span Criterion,” *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 7, no. 11, pp. 1757–1762, 2012.

[24] A. K. Agarwal and R. Garg, “Flow Shop Scheduling Problem For 10-Jobs, 8-Machines With Make Span Criterion,” *International Journal of Innovative Research and Development*, pp. 389–403, 2013.

[25] A. K. Agarwal and R. Garg, “Flowshop Scheduling Problem for 10-Jobs, 10-Machines By Heuristics Models Using Makespan Criterion,” *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 2, no. 1, pp. 1058–2319, 2013.