Flow Shop Scheduling Using a Combination of Ant Colony Optimization Algorithm and Tabu Search Algorithm to Minimize Total Tardiness

Hana Merlina Hesti Bestari[1], Pratya Poeri Suryadhini[2], Nopendri[3]
[1,2,3] Industrial Engineering Study Program, Faculty of Industrial Engineering, Telkom University
Jl. Telekomunikasi Terusan Buahbatu, Bandung, Jawa Barat, 40257
Email: hanamerlinahesti@gmail.com, pratya@telkomuniversity.ac.id, nopendri@telkomuniversity.ac.id

ABSTRACT

This paper addresses the problem of production tardiness on five parallel production floors at PT Garmen X, each with an identical machine arrangement. The proposed method combines Ant Colony Optimization (ACO) and Tabu Search (TS) algorithms for flow shop scheduling problems. ACO acts as the primary method for finding the optimal solution. At the same time, the Tabu Search algorithm is applied as a local search to improve the quality of the solution found by ACO. The results show significant performance improvement, with a decrease in total tardiness by 88.09% and a reduction in total makespan by 5.08% compared to the existing method.

Keywords: *Garment, Flow shop, Ant Colony Optimization, Tabu Search, Total Tardiness.*

Introduction

PT Garmen X is a garment manufacturing company utilizing a First Come. First-served scheduling system to manage the production of customer orders. The FCFS method prioritizes tasks based on the order in which customer requests are received, ensuring that the earliest orders are processed in the production schedule [1]. This approach is straightforward but has limitations, especially in complex manufacturing environments. PT Garmen X operates five production floors, all arranged in parallel. This setup means that all five floors work simultaneously, each equipped with identical or similar machinery, allowing them to handle any assigned job. Despite the uniformity in machinery across the production floors, the work assigned to one floor is not transferable to another. In other words, each production floor is responsible for its specific tasks, contributing to the overall production workflow independently. However, this structure presents challenges, particularly related to total tardiness caused by inefficient work sequencing on each production floor. The effectiveness of the scheduling system can be significantly impacted by how healthy tasks are sequenced. Poor sequencing can lead to bottlenecks, increased waiting times, and overall inefficiencies in the production process [2]. As a result, while the FCFS method ensures fairness in order processing, it may not optimize the production workflow, leading to total tardiness and reduced performance in meeting production goals. PT Garmen X experienced a total tardiness of 52% from 25 jobs, totaling 42 days. These issues may result in decreased client satisfaction and decreased recurring business for PT XYZ.

Flow shop scheduling problem is not unique to PT Garmen X and has been prevalent in various industrial sectors, so flow shop production flow has been the focus of research for the past 50 years [3]. The primary objectives of this research typically include minimizing the total makespan, total tardiness, and total idle time within production processes [4]. A variety of methods have been developed to address flow shop scheduling issues. These include branch and bound [5], mixed integer linear programming [6], and the Johnson algorithm [7]. Several metaheuristic techniques have been widely employed to address the flow shop scheduling problem, producing results close to optimal. Metaheuristic methods are particularly advantageous for their ability to handle complex, non-linear problems. Some of the most used metaheuristic techniques include Simulated Annealing (SA) [8], Particle Swarm Optimizations (PSO) [9], Genetic Algorithm (GA) [10], Firefly Algorithm (FA) [11], Tabu Search (TS) [12], dan Ant Colony Optimization (ACO) [13]. However, each metaheuristic method has its advantages and disadvantages. Therefore, many studies try to develop a combination of two/more algorithms to improve algorithm performance in finding optimal solutions.

ACO is a metaheuristic approach modeled after the behavior of ant colonies [14]. It was first developed to solve the Traveling Sales Problem and then used to solve combinatorial optimization problems, such as flow shop and job shop scheduling. Famous examples of using ACO to solve scheduling issues with total tardiness minimization were developed by [15] and [16]. These studies demonstrated ACO's strong computational abilities, particularly its capacity to avoid premature convergence on suboptimal solutions. However, a known limitation of ACO is that it often requires a significant amount of time to arrive at an optimal solution [17]. Therefore, various methods have been proposed to speed up ACO convergence time by applying local search. This combination has previously been used by [18] in minimizing makespan for job shop scheduling, yielding impressive results. The synergy between ACO and local search methods, such as Tabu Search (TS), has proven to be a powerful strategy for achieving more efficient and effective solutions.

This research introduces a solution to the flow shop scheduling problem by integrating ACO with Tabu Search (TS) to minimize total tardiness. ACO is employed as the primary algorithm to generate a suitable global solution, while TS is used to refine and improve the quality of the solution iteratively. This hybrid approach prevents the algorithm from becoming trapped in local optima and achieves superior results with faster convergence time. By leveraging the strengths of both ACO and TS, the proposed method aims to provide a robust and efficient solution to complex scheduling challenges.

Research Methods

Problem Statement

Scheduling jobs on specific machines is a core component of production scheduling, crucial for optimizing manufacturing processes and ensuring that operations run smoothly and efficiently. At PT Garmen X, "machine" represents the production floors. The primary aim in this context is to strategically allocate or schedule jobs across these production floors to maximize efficiency and productivity throughout the entire production system. This scheduling task is critical as it directly impacts the efficiency and productivity of the entries production system. In this paper, scheduling is done by considering that production is carried out sequentially on each production floor continuously. Flow shop scheduling at PT Garmen X involves n jobs processed on m production floors, each starting simultaneously. As stated earlier, the goal achieved in this paper is to minimize total tardiness, which can be calculated using several steps:

a. Completion time (C_j): The total time required to complete a job.
   1. Calculate $C_{1j} = p_{ij}$, where p_{ij} = process time for the job i on the production floor j where $(1 \leq i \leq n)$ and $(1 \leq j \leq m)$.
      Where $i, j \in Z$ with Z being the set of positive integers.
   2. Calculate $C_{ij} = C_{i(j-1)} + p_{ij}$
      Calculate $C_{ij} = max\left\{C_{i(j-1)} + p_{ij}\right\}$, for $(1 \leq i \leq n)$ and $(1 \leq j \leq m)$ where $i, j \in Z$ with Z being the set of positive integers.

b. Tardiness (T_i): the condition where work is completed beyond the specified deadline and can be calculated as follows [19]:
   $T_i = max\{0, C_{ij} - d_i\}$, where d_i = due date for job i.

c. The objective of minimized total tardiness can be formulated as follows [19]:

$$min \sum_{i=1}^{n} T_i \tag{1}$$

Ant Colony Optimization (ACO)

ACO is a metaheuristic technique derived from the behavior of ant colonies, which is characterized by a unique form of social cooperation and communication through chemical signals known as pheromones [14]. The ACO algorithm was originally introduced by Marco Dorigo [20] as a novel method for addressing the shortest path problem, such as the Traveling Salesman Problems. The foundational concept of ACO is based on the observation that ants, while searching for food, leave behind a trail of pheromones on the ground. These pheromone trails serve as a guide for other ants, who tend to follow the paths with higher pheromone concentrations, which often correspond to shorter or more efficient routes. This is used as a signal for other ant to follow the route taken by the previous ants. Therefore, ACO assumes that (artificial) ant colonies use previously discovered (artificial) pheromone trails in constructing the best solution that can be iteratively improved. In addition to being based on the value of the pheromone trail function, solution selection will be done incrementally by considering the value of the heuristic information function. Each ant will produce a solution, and the selected solution at each iteration will be stored in the Tabu list. A Tabu list is a memory

list used to store the best solution found so the solution cannot be selected at a particular time. During the solution-building process by the ants, the value of the pheromone trail disappears after the ants choose to include it in the taboo, which is called evaporation. The movement of ants can be seen in Figure 1.

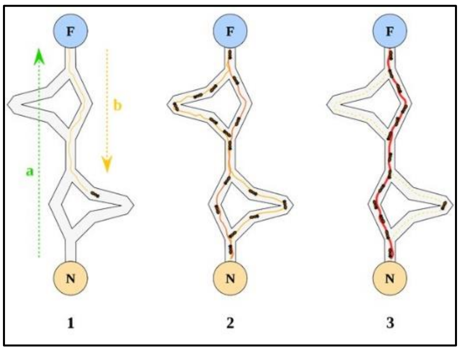

Figure 1. (1) Ants finding food, (2) Ant colony looking for the shortest distance, (3) Ant colony gets the shortest distance
Reference: [21]

Ant Colony Optimization Parameters
ACO has several parameters used to perform the initiation [22]:

a. Number of cycles (NC_{max}), refers to the number of iterations executed throughout the solution search process. The ants require this number of cycles to find a solution.
b. Number of ants (m), the number of artificial entities used to form various combinations of solutions at each cycle/iteration.
c. Pheromone trail (τ_{ij}), the chemical substance left by each ant during the journey, as information for the next ant to pass the same path.
d. Heuristic information (η_{ij}), is the visibility of the job selected in each phase based on the mathematical function value.
e. The relative significance of the pheromone trail (α), is the weight assigned to the pheromone trail parameter. Hence, the solution obtained follows the ant's history in the previous movement. The value of the parameter $\alpha \geq 0$.
f. The relative importance of heuristics information (β), is the weight assigned to the heuristic information parameter such that the resulting solution is usually based on a mathematical function. Parameter value $\beta \geq 0$.
g. Evaporation coefficient (ρ), is a measure of the evaporations coefficient of the pheromone trail dissipating over time, causing the evaporation of pheromones, which prevents all ants from following identical paths? The parameter value of the evaporation coefficient is $0 \leq \rho \leq 1$. The probability of ants performing the exploitation process at each stage (q_0) $(0 \leq q_0 \leq 1)$.

Ant Path Selection Rules

During path selection in schedule formation, ant k on job i will have job j with the following rules [4]:

$$j = \begin{cases} \underset{u \in S_k(i)}{\arg\max}\left\{[\tau_{iu}]^{\alpha}[\eta_{iu}]^{\beta}\right\}, & if\ q \leq q_0 \\ J, & otherwise \end{cases} \tag{2}$$

Where, τ_{iu} is the number of pheromone trail on edge (i,u), whit the calculation of η_{iu} as follow:

$$\eta_{ij} = \frac{1}{\max\,(t^* + p_i,\ d_i)} \tag{3}$$

Equation (3) calculates heuristic information using the Modified Due Date method with a value of $t^* = 0$ for this case because it is a type of static scheduling [23]. $S_k(i)$ is the set of jobs fefeasible for ant k to select on the job i. inside the for ant k to selecton the job i inside $tabu_k$. Path selection is determined by comparing the q_0 with the q value. The value of q represents a value obtained arbitrarily from probability uniform in [0,1],. In contrast value of q_0 is a parameter that has been initialized at the beginning of the algorithm to determine whether the ants choose a path based on the exploitation or exploration process. J is a random variable chosen based on the cumulative probability distributions, i.e.

where to move from job i ant k chooses job j with a cumulative probability value greater than the value of q. The value of J is obtained based on the following probability distribution:

$$P_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij}^{\alpha})\,(\eta_{ij}^{\beta})}{\sum_{u \in S_k(i)} (\tau_{iu}^{\alpha})(\,\eta_{iu}^{\beta})}, & If\ j \in S_k(i) \\ 0\,, & otherwise \end{cases} \tag{4}$$

Based on these two rules, if the value $q \leq q_0$ will perform exploitation using Equation (2); otherwise, the exploration process will be carried out using Equation (4).

Pheromone Update

After forming a solution schedule, the ants will perform a pheromone update to reduce the pheromone trail so that it is possible to explore the solution space more widely. The following is the local pheromone update rule performed by ants [15]:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \rho.\tau_0 \tag{5}$$

Where, $\tau_0 = \frac{1}{p_i}$ is initial pheromone.

Furthermore, once all ants have completed their schedules, a global pheromone update is conducted to reinforce the most successful pheromone trail created by the ants. The global pheromone update is updated with the following equation:

$$\tau_{ij} = (1-\rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \tag{6}$$

Where,

$$\Delta\tau_{ij}(t) = \begin{cases} \dfrac{1}{T}, & for\ each\ i,j\ of\ the\ best\ solution \\ 0, & otherwise \end{cases} \tag{7}$$

In Equation (6), ρ $(0 < \rho < 1)$ and T are the total tardiness as the best possible value of the objective function for the best schedule.

Tabu Search (TS)

Fred Glover introduced Tabu Search (TS) in 1986 as an innovative metaheuristic technique. This method enhances local heuristic search processes by directing them to explore the solution space beyond the confines of traditional local optimization. Tabu Search is widely regarded as one of the most effective strategies for tackling combinatorial optimization challenges. It was specifically developed to address the shortcomings of local search methods, which tend to become trapped in suboptimal solutions. By employing Tabu Search, a more thorough examination of the solution space is possible, increasing the likelihood of identifying best solutions. This method uses adaptive memory to avoid revisiting previously explored solutions, enabling a more effective search and reducing the likelihood of getting trapped in suboptimal solutions [24]. Insertions and exchanges are frequently performed to generate new neighborhoods [25]. TS can be applied to various issues, from computer scheduling and channel balancing to cluster analysis and space planning [26]. This paper uses TS as a local search to improves the solutions formed by ACO. There are several rules in local search [27], namely:

1. Swapping is performed by constructing random numbers i and j to indicate the position of i and j. The swap process replaces the job in position i with the job in position j.
2. Insertion is done by building random numbers, a process that is almost the same as swapping, but the difference is that the job in position i is move to position j.
3. Block insertion is done by constructing numbers i, j, and k. And then adding k *job*s, starting from job i and ending at position j.

In this paper, TS as a local search, performs the swapping process by randomly swapping jobs from one production floor to another.

Ant Colony Optimization – Tabu Search (ACO-TS)

In this research, the combination of the ACO and TS algorithms is called ACO-TS. This merger minimizes the convergence time and improves the standard of the generated solution. The search for the best scheduling solution to minimize total tardiness can be done through the following steps.

Step 0. Input the company's scheduling dataset. Go to Step 1.

Step 1. Initialize the parameters and set the parameter values $NC_{max}, m, \alpha, \beta, \rho, q_0$. Go to Step 2.

Step 2. Set the values of τ_{ij} with the values of 1 and η_{ij} according to Equation (3), $\epsilon\ i, j$.

Step 3. Initialize the Tabu Search requirements in the form of Tabu tenure and Tabu list size.

Step 4. Perform ant generation. The generated solution is k. Go to Step 5.

Step 5. Start the iteration by making a stochastic decision to select the next ant path. The value of q is randomly selected with uniform probability in $[0, 1]$. Go to Step 6.
Step 6. Does the value of $q \leq q_0$? If yes, go to Step 7; if not, go to Step 8.
Step 7. Path selection is performed by exploitation using Equation (2). Go to Step 9.
Step 8. Use Equation (4) to select a path during the exploration process. Go to the Step 9.
Step 9. Perform local pheromone update according to Equation (5). Continue to the Step 10.
Step 10. Calculate the fitness value according to Equation (1). Go to Step 11.
Step 11. Find the best solution from S' in the tabu list with the smallest total tardiness criteria.
Step 12. Whether $S < S'$ is the process of making the best solution decision. S is the newly found best solution, while S' is the best in the tabu list. If yes, go to Step 14; if not, go to Step 13.
Step 13. Since $S > S'$, another alternative solution is searched to avoid getting stuck in a cycle or local solution already explored. Go to Step 9.
Step 14. Update the tabu list and add the newly found best solution (S); because $S < S'$, then the solution (S') with the highest total tardiness value will be removed, go to Step 15.
Step 15. If the present quantity of ants(k) matches the specified number of ants (m), go to Step 16. If not, do $k = k + 1$ again until if the present quantity of ants(k) matches the specified number (m) at the initialization stage, so go to Step 5.
Step 16. According to Equation (6), perform a global pheromone update from a selection of local solutions. Then, go to Step 17.
Step 17. Does the value of $NC = NC_{max}$? If not, then an additional cycle $NC = NC + 1$. Go to step 5. But, if yes, then the process is stopped and go to Step 18.
Step 18. Print the best solution.

Performance Parameters

After obtaining the proposed scheduling results from the ACO-TS algorithm, the suggested approach's effectiveness is measured to minimize total tardiness. This measurement is carried out with two parameters, namely Efficiency Index (EI) and Relative Error (RE), which are calculated in the following equation [28]:

$$EI = \frac{T_{company's\ actual\ method}}{T_{proposed\ method}} \tag{8}$$

If the $EI\ value > 1$, the proposed method is better than the company's actual method in reducing total tardiness.

$$RE = \frac{T_{company's\ actual\ method} - T_{proposed\ method}}{T_{company's\ actual\ method}} \times 100\% \tag{9}$$

Results and Discussion

This paper evaluates the application of the ACO-TS method on data sourced from PT Garmen X. The data indicates that 25 jobs must be scheduled across 5 production floors. The 18 stages outlined above determine the optimal scheduling solution utilizing MATLAB software. The computations are performed on a computer equipped with an AMD Ryzen 7 4700U processor, Radeon Graphics, a 2.00 GHz CPU, and 16 GB of RAM. As a combination algorithm, the ACO-TS algorithm has a challenge in implementing the best solution: setting the correct parameters to achieve optimal performance. This tuning process often requires extensive experimentation and repeated iterations, which can be time-consuming and resource-consuming. In this paper, several scenarios are made for each parameter obtained from previous research, producing good results in achieving the objectives. These parameter scenarios will form several parameter combinations that are then assessed for their performance in achieving the smallest total tardiness.

Table 1. Scenario for 6 ACO Parameters

Number of Cycles (NC_{max})	Number of Ant (m)	α	β	Evaporation Coefficient (ρ)	q_0
5.000	50	1	1	0,5	0,1
10.000	100	10	10	0,99	0,5 0,9

Based on Table 1, 96 different parameter combinations will be tested to determine which yields the minimum total tardiness for this scheduling problem. Each combination is tested 30 times to ensure reliable results and assess the variability of outcomes. The Tabu Search (TS) component uses two key parameters: Tabu Tenure and Tabu List size. Tabu Tenure is set at 5, determining how long a solution is prohibited from revisiting. At the same time, the Tabu List size matches the number of jobs to be scheduled to prevent premature convergence on suboptimal solutions. These parameters were selected through empirical trials to optimize the Tabu Search's performance. The average total tardiness for each combination is shown in Table 2, providing insights into the most effective settings for improving scheduling efficiency and productivity. By evaluating these combinations, the study identifies the optimal configuration for enhanced scheduling performance. This version maintains the essential information while being more concise.

Table 2. The Minimum (Best) Total Tardiness Value From 30 Trials on Each Parameter Combination

Combination	Average Makespan (Days)	Average Total Tardiness (Days)	Average Computation Time (Seconds)
1	56	24	30,8844
2	60	17	55,6093
3	57	13	60,8185
4	58	20	55,3381
5	56	18	54,8716
…	…	…	…
80	58	14	167,6304
81	56	5	110,7002
82	60	13	291,8960
83	58	14	168,4175
84	59	16	168,7565
85	58	14	161,3477
86	56	5	147,4991
87	61	13	166,7031
…	…	…	…
92	55	16	206,2337
93	58	9	203,4163
94	57	5	226,3750
95	59	14	344,7472
96	57	11	440,5782

Table 2 shows that out of 96 parameter combinations, three-parameter combinations produce the smallest total tardiness value, namely the 81st parameter combination, 86th parameter combination, and 94th parameter combination, with a total tardiness of 5 days. In addition to the smallest total tardiness value, parameter selection is also based on One-Way ANOVA analysis to evaluate the effect of each parameter on total tardiness minimization. Based on the One-Way ANOVA analysis, the parameters that have a significant impact on total tardiness are the number of cycles (NC_{max}) and the number of ants (m) because the $p-value\ < 0.05$ is obtained. The larger the number of cycles, the more the number of iterations, the more effective it is in minimizing tardiness because a more significant number of iterations allows the algorithm to explore better and exploit the solution space, make incremental improvements, and direct the search towards the optimal solution. Also, the higher the number of ants, the more pheromone trails will be updated based on the solutions found. This helps create a more accurate and informative pheromone trail, leading subsequent ants to choose a more optimal path. However, too many cycles and the number of ants used will increase the complexity of the calculation. n the problem at PT Garmen X, the parameters $\alpha, \beta, \rho, dan\ q_0$ have no significant effect because the value of each $p-value > 0.05$. However, there is a difference in the average value of the total delay generated by each parameter.

The results of the One-Way ANOVA analysis show that the 81st parameter combination has the best performance in minimizing total tardiness; the average value of total tardiness evidence this generated the smallest compared to other parameter combinations, which is 17,70 days. Therefore, a suitable parameter combination for this problem is a combination with a parameter composition of $(NC_{max}) = 10.000$, m = 100, $\alpha = 1$, $\beta = 10$, $\rho = 0{,}5$, and $q_0 = 0{,}9$. The parameter combination was chosen because it provides optimal performance, namely providing the lowest total tardiness value, and has good enough consistency because it has the lowest average total tardiness. Based on the results of running the ACO-TS algorithm using this combination, the best (minimum) total tardiness value is five days. This best value is used as a scheduling proposal for PT Garmen X with a Gantt chart display that can be seen in Figure 2.

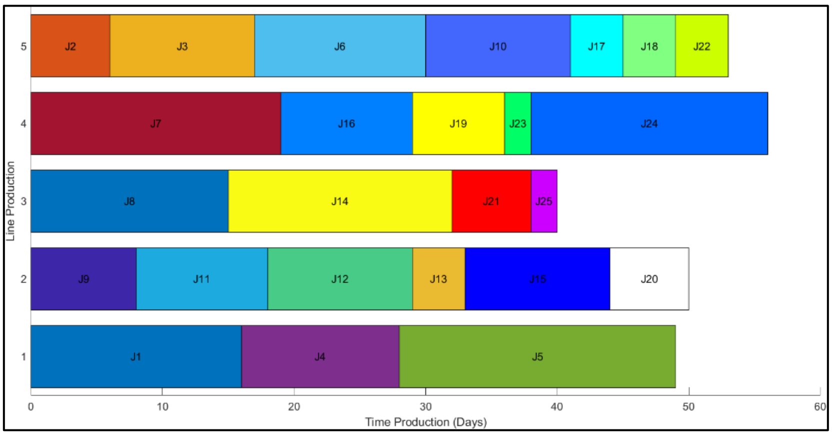

Figure 2. The proposed Gantt chart was generated from the 81st parameter combination for the five production floors at PT Garmen X

Figure 2 illustrates the sequence and duration of jobs on each production line to minimize total tardiness. Each color represents a different job. Colored blocks indicate that the job can be completed before the deadline, while white blocks indicate late jobs. In Figure 2, there is a late job, job J20, which has a total tardiness of 5 days out of 25 jobs for 56 days. This result was obtained with a computation time of 110.7002 seconds, equivalent to 1.85 minutes. The proposed scheduling conditions, which combine the Ant Colony Optimization and Tabu Search (ACO-TS) methods, result in a smaller total tardiness compared to the FCFS method currently used by PT Garment X. However, to ensure that the proposed scheduling method can efficiently and effectively achieve production goals, a performance test is conducted using two parameters: Efficiency Index (EI) and Relative Error (RE). The first step involves calculating the Efficiency Index (EI) from the total tardiness results using Equation (8).

$$EI = \frac{42}{5} = 8{,}4$$

The EI value of 8.4 $>$ 1 is obtained; this shows that the proposed scheduling is better than the method currently used by PT Garmen X in minimizing total tardiness. Furthermore, the Relative Error (RE) calculation with Equation (9) is as follows.

$$RE = \frac{42-5}{42} \times 100\%$$

$$RE = 88{,}09\%$$

The relative error (RE) value obtained shows a decrease in total tardiness of 88.09% using the ACO-TS algorithm. These results clearly show that the combination of the ACO-TS method can find a better flow shop scheduling solution for the problem faced by PT Garmen X and provide a short computation time. However, this computation time depends on the complexity of the problem. The more complex the problem and the larger the size of the job to be scheduled, the more memory usage and long computation time can be involved. This paper’s computation time is relatively small because it only schedules 25 jobs for five production floors.

Conclusion

This paper introduces a hybrid method combining ACO and TS to address the flow shop scheduling issue and efficiently reduce overall tardiness. In this approach, the ACO algorithm is the primary component, tasked with finding a global solution by exploring the broader solution space. Simultaneously, Tabu Search acts as a local search technique, refining and enhancing the solutions generated by ACO. The synergy between ACO and TS allows the combined method to accelerate the convergence process, improving the overall quality of the solutions obtained. When applied to data from PT Garmen X, this ACO-TS combination demonstrates significant improvements over the company's existing scheduling methods. The test results reveal a dramatic decrease in total tardiness by 88.09% and a reduction in the total makespan by 5.08%. These improvements highlight the method's effectiveness in optimizing job schedules and enhancing production efficiency. Moreover, the ACO-TS method achieves these results with a relatively short computation time, making it a practical solution for industrial applications.

In future research, it is expected that ACO-TS can be applied to scheduling scenarios involving more machines or more complex operations and utilized for scheduling problems with multiple objectives, such as optimizing total tardiness while considering production costs or resource usage. In addition, in the future, this ACO-TS algorithm is expected to be able to be used by adding start time variations on each production floor to increase flexibility and adaptability. By incorporating different start time scenarios, this approach can be refined to better meet the dynamic needs of modern manufacturing environments, such as in the automotive or electronics and service industries.

References

[1] S. N. Chapman, *The fundamentals of production planning and control*. Pearson/Prentice Hall, 2006.

[2] D. M. Utama, L. R. Ardiansyah, and A. K. Garside, “Penjadwalan Flow shop Untuk Meminimasi Total Tardiness Menggunakan Algoritma Cross Entropy–Algoritma Genetika,” *Jurnal Optimasi Sistem Industri*, vol. 18, no. 2, pp. 133–141, Oct. 2019, doi: 10.25077/josi.v18.n2.p133-141.2019.

[3] J. N. D. Gupta and E. F. Stafford, “Flowshop scheduling research after five decades,” *Eur J Oper Res*, vol. 169, no. 3, pp. 699–711, Mar. 2006, doi: 10.1016/j.ejor.2005.02.001.

[4] B. Yagmahan and M. M. Yenisey, “A multi-objective ant colony system algorithm for flow shop scheduling problem,” *Expert Syst Appl*, vol. 37, no. 2, pp. 1361–1368, Mar. 2010, doi: 10.1016/j.eswa.2009.06.105.

[5] J. Schaller and J. Valente, “Branch-and-bound algorithms for minimizing total earliness and tardiness in a two-machine permutation flow shop with unforced idle allowed,” *Comput Oper Res*, vol. 109, pp. 1–11, Sep. 2019, doi: 10.1016/j.cor.2019.04.017.

[6] J. Bautista-Valhondo and R. Alfaro-Pozo, “Mixed integer linear programming models for Flow Shop Scheduling with a demand plan of job types,” *Cent Eur J Oper Res*, vol. 28, no. 1, pp. 5–23, Mar. 2020, doi: 10.1007/s10100-018-0553-8.

[7] H. Allaoui and A. H. Artiba, “Johnson’s algorithm: A key to solve optimally or approximately flow shop scheduling problems with unavailability periods,” *Int J Prod Econ*, vol. 121, no. 1, pp. 81–87, Sep. 2009, doi: 10.1016/j.ijpe.2008.09.018.

[8] M. K. Hajji, O. Hamlaoui, and H. Hadda, “A simulated annealing metaheuristic approach to hybrid flow shop scheduling problem,” *Advances in Industrial and Manufacturing Engineering*, vol. 9, Nov. 2024, doi: 10.1016/j.aime.2024.100144.

[9] I. H. Kuo *et al.*, “An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model,” *Expert Syst Appl*, vol. 36, no. 3 PART 2, pp. 7027–7032, 2009, doi: 10.1016/j.eswa.2008.08.054.

[10] L. Wang, L. Zhang, and D. Z. Zheng, “An effective hybrid genetic algorithm for flow shop scheduling with limited buffers,” *Comput Oper Res*, vol. 33, no. 10, pp. 2960–2971, Oct. 2006, doi: 10.1016/j.cor.2005.02.028.

[11] M. K. Marichelvam, T. Prabaharan, and M. Geetha, “Firefly algorithm for flow shop optimization,” *Studies in Computational Intelligence*, vol. 585, pp. 225–243, 2015, doi: 10.1007/978-3-319-13826-8_12.

[12] P. N. Nhu and T. N. N. Thi, “Application of tabu search, TS, to solve a flow shop scheduling problem with changeover times in operations: A case study,” *BOHR International Journal of Operations Management Research and Practices*, vol. 3, no. 1, pp. 1–7, 2022, doi: 10.54646/bijomrp.2024.22.

[13] T. Stützle, “An Ant Approach to the Flow Shop Problem,” *Fifth European Congress on Intelligent Techniques & Soft Computing (EUFIT’98)*, vol. 3, pp. 1560–1564, 1998, [Online]. Available: http://iridia.ulb.ac.be/dorigo/ACO/ACO.html.

[14] Karjono, Moedhiono, and D. Kurniawan, “Ant Colony Optimization,” *Jurnal TICOM*, vol. 4, no. 3, 2016.

[15] A. Bauer, R. F. Hartl, C. Strauss, and B. Bullnheimer, “Minimizing Total Tardiness on a Single Machine Using Ant Colony Optimization,” 2014. [Online]. Available: https://www.researchgate.net/publication/260423657

[16] M. den Besten, T. Stützle, and M. Dorigo, “Ant Colony Optimization for the Total Weighted Tardiness Problem.,” 2000, [Online]. Available: https://www.researchgate.net/publication/220702108

[17] O. I. R. Farisi, B. Setiyono, and R. I. Danandjojo, “A Hybrid Firefly Algorithm-Ant Colony Optimization for Traveling Salesman Problem,” *Jurnal Buana Informatika*, vol. 7, no. 1, pp. 55–64, 2015.

[18] K. L. Huang and C. J. Liao, “Ant colony optimization combined with taboo search for the job shop scheduling problem,” *Comput Oper Res*, vol. 35, no. 4, pp. 1030–1046, Apr. 2008, doi: 10.1016/j.cor.2006.07.003.

[19] K. R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*, Second Edition. John Wiley & Sons, Inc., 2019.

[20] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant System: An Autocatalytic Optimizing Process,” 1991.

[21] M. D. Toksari, “A hybrid algorithm of Ant Colony Optimization (ACO) and Iterated Local Search (ILS) for estimating electricity domestic consumption: Case of Turkey,” *International Journal of Electrical Power and Energy Systems*, vol. 78, pp. 776–782, Jun. 2016, doi: 10.1016/j.ijepes.2015.12.032.

[22] Liliani and A. Alfian, “Usulan Penjadwalan Produksi Dengan Algoritma Ant Colony (Studi Kasus PT Shima Prima Utama Palembang),” *Simposium Nasional RAPI XIII*, 2014.

[23] J. C. Nyirenda, “Relationship Between The Modified Due Date Rule and The Heuristic of Wilkerson and Irwin,” *Journal of Computer Science and Engineering*, vol. 17, no. 1, pp. 101–111, 2014, Accessed: Jul. 13, 2024. [Online]. Available: https://doi.org/10.5784/17-0-192

[24] C. Venkateswarlu, “A Metaheuristic Tabu Search Optimization Algorithm: Applications to Chemical and Environmental Processes,” *IntechOpen*, 2022, Accessed: Mar. 22, 2024. [Online]. Available: http://dx.doi.org/10.5772/intechopen.98240

[25] Z. Zhao, X. Chen, Y. An, Y. Li, and K. Gao, “A property-based hybrid genetic algorithm and tabu search for solving order acceptance and scheduling problem with trapezoidal penalty membership function,” *Expert Syst Appl*, vol. 218, May 2023, doi: 10.1016/j.eswa.2023.119598.

[26] H. Pirim, B. Eksioglu, and E. Bayraktar, “Tabu Search: A Comparative Study,” 2008.

[27] I. H. Sahputra, T. Octavia, and A. S. Chandra, “Tabu Search Sebagai Local Search pada Algoritma Ant Colony untuk Penjadwalan Flow Shop,” *Jurnal Teknik Industri*, vol. 11, no. 2, pp. 188–194, 2009.

[28] S. Mardiani, “Penjadwalan Produksi Dengan Metode Tabu Search Menggunakan Software VBA Macro Excel di PT Citra Abadi Sejati,” *Scientifict Journal of Industrial Engineering*, vol. 2, no. 2, 2021, Accessed: Aug. 05, 2024. [Online]. Available: https://doi.org/10.30998/.v2i2.4179