Society of Production Engineering

ICPES 2025

40th INTERNATIONAL CONFERENCE ON PRODUCTION ENGINEERING - SERBIA 2025

DOI: 10.46793/ICPES25.247B

University of Nis Faculty of Mechanical Engineering

Nis, Serbia, September 18 – 19, 2025

DYNAMIC FLEXIBLE JOB SHOP SCHEDULING PROBLEM BASED ON GENETIC ALGORITHM

Katarina BRENJO[1,*], Aleksandar JOKIĆ[1], Milica PETROVIĆ[1]
Orcid: 0000-0001-8854-8246; **Orcid:** 0000-0002-7417-4244; **Orcid:** 0000-0002-4950-6518;
[1]University of Belgrade – Faculty of Mechanical Engineering, Belgrade, Serbia
*Corresponding author: kbrenjo@mas.bg.ac.rs

Abstract: *The increasing frequency and sophistication of cyber-attacks on manufacturing systems demand that scheduling frameworks evolve to include cybersecurity considerations. The integration of dynamic and cybersecurity-related factors into the flexible job shop scheduling problem modelling is essential to better reflect real-world manufacturing conditions. This paper addresses the flexible job shop scheduling problem in a dynamic manufacturing environment, affected by three unexpected disturbances: the arrival of new jobs into the manufacturing system, job cancellations, and machine tool breakdowns. Particularly, some of these disturbances are caused by cyber-attacks targeting manufacturing systems, increasing risks to production and operational reliability. These disturbances have a significant impact on manufacturing efficiency, affecting delivery deadlines, resource utilization, and overall processing time. In this research paper, a genetic algorithm is applied as a robust artificial intelligence technique suitable for solving NP-hard combinatorial problems such as the dynamic flexible job shop scheduling problem. The algorithm facilitates real-time adjustment through rescheduling mechanisms, aiming to achieve a specified optimization objective – minimizing the total processing time (makespan). The proposed method is implemented in the MATLAB® environment and validated through simulations using relevant benchmark problems. Experimental results demonstrate that the proposed methodology significantly improves adaptability and performance in dynamic manufacturing environments, while maintaining high efficiency despite sudden interruptions. Overall, the proposed approach advances intelligent and adaptive real-time rescheduling in a flexible job shop environment, supporting the Industry 4.0 concept by enhancing the flexibility, efficiency, and performance of intelligent manufacturing systems that can withstand both disturbances and emerging cyber threats.*

Keywords: *dynamic flexible job shop scheduling, genetic algorithm, rescheduling, manufacturing systems, optimization, disturbances, cyber-attack.*

1. INTRODUCTION

In the era of Industry 4.0, manufacturing systems are transforming into intelligent, self-organizing, and data-driven environments that integrate both physical operations and decision-making processes. Within this context, the scheduling of operations plays a critical role in enabling flexibility, efficiency, and responsiveness to dynamic and unpredictable conditions [1]. While these developments improve efficiency and flexibility, they also expose manufacturing environments to a wide range of disturbances, including those caused by cyber-attacks [2]. These disturbances can result in severe consequences, posing significant challenges to traditional scheduling approaches.

The Flexible Job Shop Scheduling Problem (FJSSP) is a well-known NP-hard combinatorial optimization problem that extends the classical Job Shop Scheduling Problem (JSSP) by allowing operations to be performed on one of several alternative machine tools. This flexibility provides a more realistic representation of manufacturing environments, but it also significantly increases computational complexity. When such scheduling problems occur in dynamic environments, where the system is affected by unplanned events, it becomes the Dynamic Flexible Job Shop Scheduling Problem (DFJSSP) [3]. Unlike static models, the DFJSSP better reflects real-world manufacturing conditions by incorporating the need for rescheduling in response to unexpected changes.

This research paper addresses the DFJSSP in the presence of three types of disturbances: (i) the arrival of new jobs [4], (ii) the cancellation of existing jobs [5], and (iii) machine breakdowns [6]. Notably, some of these disturbances represent a direct consequence of cyber-attacks targeting manufacturing systems.

To respond effectively to these challenges, a scheduling approach based on Genetic Algorithms (GA) is proposed [7]. The GA-based approach enables real-time adaptation through dynamic rescheduling, taking into account the current state of the manufacturing system. The algorithm is designed to minimize total processing time (makespan) [8].

Previous studies have demonstrated the effectiveness of GA in addressing key disturbances in manufacturing systems. For instance, [7] developed a modular GA-based framework for dynamic adaptation that supports both regeneration and modification of populations following disruptions, such as machine tool breakdowns and job cancellations, thereby facilitating the effective continuation of the search process. Similarly, the authors of [9] applied GA to dynamic job-shop scheduling with continuous arrival of new jobs, optimizing multiple objective functions and outperforming priority rule-based approaches under both deterministic and stochastic conditions. The research in [10] extended GA applications to real-time rescheduling by considering critical disturbances, such as machine failures, new job arrivals, and job cancellations, thereby enabling the rapid generation of new optimal schedules without reevaluating completed operations. In addition, [11] introduced a novel GA designed to handle machine tool breakdowns, achieving significant reductions in makespan compared to conventional methods by efficiently managing interruptions without deferring operations. Furthermore, [12] proposed an improved GA combined with a rolling scheduling strategy and specialized mutation operators, demonstrating enhanced local search capabilities and objective function performance when handling new job arrivals and machine downtimes in dynamic environments. These contributions highlight the suitability and flexibility of GA-based methods for robust and adaptive scheduling in complex manufacturing environments characterized by high levels of uncertainty and disturbance occurrence.

The developed methodology is implemented in the MATLAB® environment, which supports the representation of alternative process plans, the management of dynamic constraints, and the visualization through Gantt charts.

This paper is organized as follows: Section 2 formulates the DFJSSP and defines the associated constraints and objectives. Section 3 presents the proposed GA-based solution methodology. Section 4 provides the experimental evaluation and simulation results discussion. Section 5 concludes the research paper outline. Section 6 provides essential directions and insightful suggestions for future research.

2. DFJSSP FORMULATION

The flexibility of manufacturing processes is a fundamental requirement for efficient and adaptive manufacturing systems, encompassing various dimensions such as machine tool flexibility, tool flexibility, tool orientation flexibility, and process flexibility. This research paper focuses on three key types of flexibility: (i) machine tool flexibility, where a single operation can be performed on multiple alternative machine tools; (ii) process plan flexibility, which refers to the possibility of processing a job in different ways; and (iii) operation sequence flexibility, which represents the ability to change the order of operations during the optimization of manufacturing processes.

According to the mathematical model developed in [13], three alternative manufacturing process plans are generated for each job, based on the criterion of minimizing total production time, taking into account both the processing time of operations and the transportation time between alternative machine tools. The resulting process plans represent one of the key input parameters for the scheduling process, where the final assignment and sequencing of operations are subject to further optimization.

To determine the optimal scheduling plan, this study considers makespan as the objective function. The mathematical formulation of minimizing makespan is defined as:

$$obj = \max(c_{ij}),\left(c_{ij} \in T_d(s_{ij}, c_{ij})\right), \tag{1}$$

where: c_{ij} – the completion time of operation O_{ij}; s_{ij} – the start time of operation O_{ij}; T_d – the set of start times and completion times of all operations of all jobs.

Examples of selected alternative manufacturing process networks for processing four jobs are presented in Figure 1, while Figure 2 illustrates a job-shop scheduling problem.

A set of operations is determined for each job, along with a specified sequence of these operations on machine tools. The processing time for each operation on the corresponding machines is also specified. For instance, to process jobs 1 and 2, three operations are required for each; job 3 requires four operations, while job 4 requires two operations. Based on the information provided by the alternative manufacturing process networks, an initial scheduling plan is generated before any potential disturbance. Subsequently, an optimal rescheduling plan is developed following each of the three mentioned disturbances (Figures 3, 4, and 5).

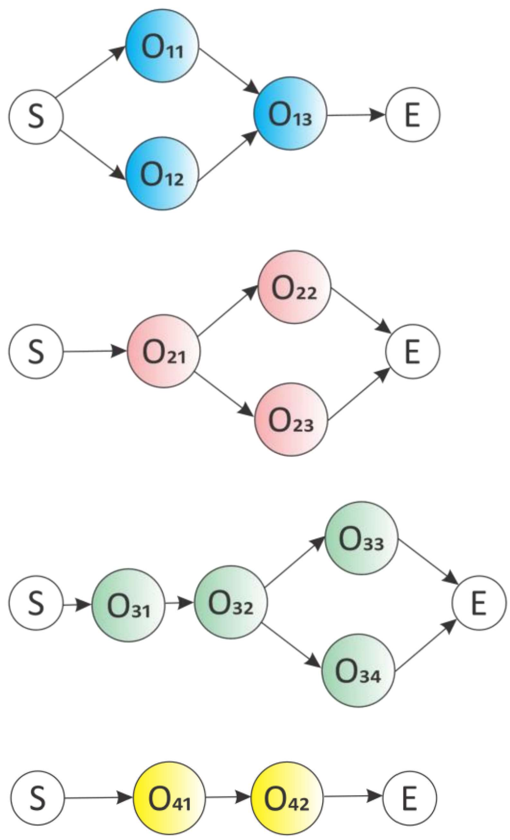

Figure 1. Alternative manufacturing process plans

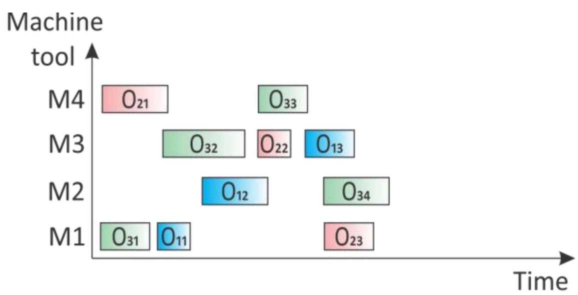

Figure 2. Gantt chart of job shop schedule

The arrival of a new job implies the need to modify the order of existing operations after the arrival time, while also considering operations required for processing the new job (Figure 3). The updated scheduling plan, also referred to as the rescheduling plan, can be utilized to enhance the overall performance of the manufacturing system in processing new jobs while simultaneously meeting the deadlines for all jobs within the system.

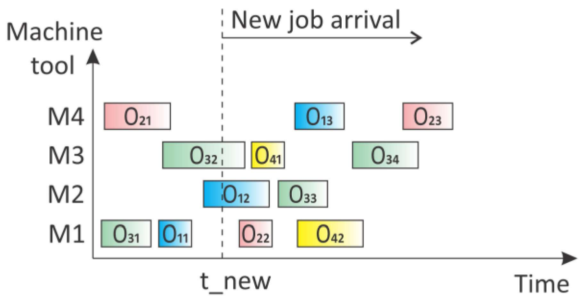

Figure 3. Rescheduling due to the arrival of a new job 4 in the system

Job cancellation represents a disturbance that requires the termination of processing for a specific job. Therefore, after the cancellation, rescheduling is performed for the remaining jobs, excluding all remaining scheduled operations of the canceled job (Figure 4).

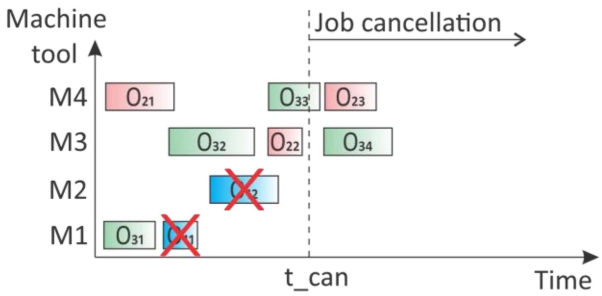

Figure 4. Rescheduling due to the cancellation of job 1

A machine tool breakdown results in the rescheduling of all ongoing operations that were not completed at the moment of failure and were assigned to the failed machine tool. Modeling approaches for handling such disturbances typically assume that all incomplete operations are reassigned to alternative machine tools suitable for continuing the processing of the corresponding job (Figure 5).

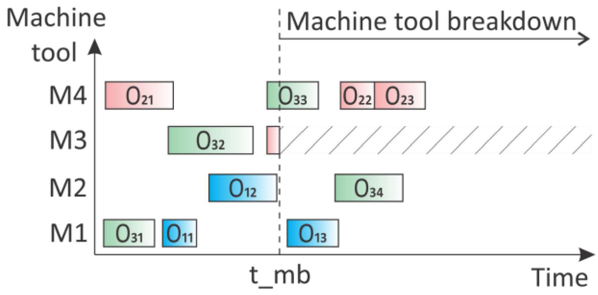

Figure 5. Rescheduling due to the breakdown of machine tool M3

The following assumptions are taken into account during the rescheduling processes illustrated in Figures 3, 4, and 5:

- The time required for rescheduling is considered to have no substantial impact, and processing of all jobs on all machine tools resumes immediately after the rescheduling is completed.
- If an operation was in progress at the moment a disturbance occurred, it continues on the same machine tool; in the case of a machine breakdown, the operation is reassigned to an alternative machine tool on which it can be processed.
- At any given moment, each machine tool can process only one operation of a single job.
- Jobs are available for processing starting from time $t_0 = 0$ in the initial scheduling plan, and from time $t = r_i$ after a disturbance occurs, where r_i represents the release time – the earliest moment when the next operation of job i can initiate.
- Different operations of the same job cannot be processed simultaneously.
- Once an operation on a machine tool is completed, the job is immediately transferred to the machine where the next operation is scheduled, considering also the transportation time between machines.
- The setup time of the machine tool, as well as other production resources for the subsequent operation is not taken into account in DFJSSP.
- The duration of the machine tool breakdown is unknown. Therefore, all operations that were scheduled to be processed on the failed machine tool must be reassigned to alternative machines during rescheduling.

2.1 The mathematical model of the DFJSSP

The mathematical model of the dynamic flexible job shop scheduling problem, presented below, is based on the research outlined in studies [14, 15].

List of symbols:
N – number of jobs $i = 1, \ldots, N$;
M – number of machine tools, $m = 1, \ldots, M$;
O_{ijk}^{m} – the k-th operation of the j-th alternative manufacturing process of job i executed on machine tool m;
r_i – release time for job i is the earliest time when the next operation of job i can start after the disturbance occurs;
r_m – release time for machine m is the earliest time when the next operation can start on machine tool m after the disturbance occurs;
t_{new} – new job arrival time;
t_{can} – job cancellation time;
t_{mb} – machine tool breakdown time;
t_d – time of disturbance occurrence (in general);
s_{ijk}^{m} – start time of the operation O_{ijk}^{m};
c_{ijk}^{m} – completion time of the operation O_{ijk}^{m};
z_{ijk}^{m} – a binary variable that takes the value 1 if the machine tool assigned to perform O_{ijk}^{m} remains unchanged, and 0 otherwise.

2.2 New job arrival and job cancellation

From the moment the disturbance occurs, it is necessary to reschedule the interrupted operations. Operations that were completed prior to the occurrence of the disturbance, as well as those that were performed at that moment, are excluded from the rescheduling process. In the case of the arrival of a new job (Figure 3), the completed operations O_{11}, O_{21} and O_{31} ($c_{ijk}^{m} \leq t_{new}$), as well as the operations in progress at the time O_{12} and O_{32} ($s_{ijk}^{m} < t_{new} < c_{ijk}^{m}$), are excluded from the rescheduling process. In the case of job cancellation (Figure 4), the following operations are not considered during rescheduling: O_{11}, O_{12}, O_{21}, O_{22}, O_{31}, and O_{32} ($c_{ijk}^{m} \leq t_{can}$), as well as operation O_{33} ($s_{ijk}^{m} < t_{can} < c_{ijk}^{m}$), which was performing at the time of cancellation. The remaining operations are included in the rescheduling ($s_{ijk}^{m} > t_d$, where $t_d = t_{new}$ in the case of a new job arrival, and $t_d = t_{can}$ in the case of job cancellation).

The release time $r_i$is calculated based on the general equation (2):

$$r_i = c_{ijk}^{m} \times z_{ijk}^{m} + t_d \times \left(1 - z_{ijk}^{m}\right). \tag{2}$$

In the case where an operation of a job was in progress at the time t_d, and the machine tool on which the operation was being performed remains unchanged (i.e., $z_{ijk}^{m} = 1$), the equation (2) takes the following form, given by the equation (3):

$$\begin{aligned} r_i &= c_{ijk}^{m} = \\ &= \left\{ s_{ijk}^{m} + t_{ijk}^{m} \,\middle|\, s_{ijk}^{m} < t_d < c_{ijk}^{m} \right\}. \end{aligned} \tag{3}$$

On the other hand, if the operation was completed prior to the occurrence of the disturbance, assuming $z_{ijk}^{m} = 0$ (indicating that it is unknown whether the machine tool on which the next operation will be performed will change), the earliest possible start time r_i of the next operation of the job after t_d is calculated according to equation (4):

$$\begin{aligned} r_i &= t_d = \\ &= \max\left(\left\{ c_{ijk}^{m} \,\middle|\, c_{ijk}^{m} < t_d \right\}, t_d \right) \end{aligned} \tag{4}$$

It is necessary to determine the state of the machine tools at the moment t_d, that is, whether an operation is being performed on the machine at that moment or is the machine available. The earliest possible start time of the next operation O_{ijk}^{m} on machine m after the disturbance occur is calculated using the following expression (5):

$$r_m = c_{ijk}^{m} \times z_{ijk}^{m} + t_d \times \left(1 - z_{ijk}^{m}\right). \tag{5}$$

At the moment the disturbance occurred, if the operation was being performed on machine tool m, it is assumed that $z_{ijk}^{m} = 1$, which indicates that the operation continues on the same machine tool, and equation (5) reduces to expression (6):

$$\begin{aligned} r_m &= c_{ijk}^{m} = \\ &= \left\{ s_{ijk}^{m} + t_{ijk}^{m} \,\middle|\, s_{ijk}^{m} < t_d < c_{ijk}^{m} \right\}. \end{aligned} \tag{6}$$

If the operation O_{ijk}^{m} was completed prior to the arrival of the new job in the system, it is assumed that $z_{ijk}^{m} = 0$ (indicating that it is unknown whether the next operation of the job i will be performed on the same machine tool), and equation (5) reduces to expression (7):

$$r_m = t_d = \\ = \max\left(\left\{c_{ijk}^{m} \left| c_{ijk}^{m} < t_d \right.\right\}, t_d\right). \quad (7)$$

2.3 Machine breakdown

In the case of a machine tool breakdown (Figure 5), the completed operations O_{11}, O_{12}, O_{21}, O_{31}, and O_{32}, are excluded from rescheduling, as well as operation O_{33}, which was in progress on a machine unaffected by the failure. On the other hand, operation O_{22}, which was being processed on the failed machine, must be reassigned to an alternative machine tool. The earliest possible start time of the next operation of job i after the occurrence of the disturbance can be calculated using the equation (8):

$$r_i = c_{ijk}^{m} \times z_{ijk}^{m} + t_{mb} \times \left(1 - z_{ijk}^{m}\right), \\ m \neq r; \quad (8)$$

r represents the machine tool that has failed.

If $m \neq r$, two possible cases arise. The first case occurs when operation O_{ijk}^{m} was performing at the moment of the failure of another machine tool, in which case $z_{ijk}^{m} = 1$, i.e., the operation continues to be performed on machine m, and the equation (8) reduces to the equation (9):

$$r_i = c_{ijk}^{m} = \\ = \left\{s_{ijk}^{m} + t_{ijk}^{m} \left| s_{ijk}^{m} < t_{mb} < c_{ijk}^{m} \right.\right\}. \quad (9)$$

The second case applies when operation O_{ijk}^{m} was completed before the failure of the other machine tool, in which case $z_{ijk}^{m} = 0$, since it is unknown whether the next operation of job i will be performed on the same machine m as the previous one, as presented in the equation (10):

$$r_i = t_{mb} = \\ = \max\left(\left\{c_{ijk}^{m} \left| c_{ijk}^{m} < t_{mb} \right.\right\}, t_{mb}\right), i \in N. \quad (10)$$

If $m = r$ and $s_{ijk}^{m} < t_{mb} < c_{ijk}^{m}$, this indicates that the machine tool on which the operation O_{ijk}^{m} was in progress has failed. The interrupted operation is transferred to a different machine tool for further processing, in which case $z_{ijk}^{m} = 0$, i.e., O_{ijk}^{m} will continue to be processed on alternative machine m, and equation (8) is replaced by equation (11):

$$r_i = t_{mb} \quad (11)$$

Following the previous evaluation of the time when processing of job i can be resumed – r_i, the time at which the machine tool m can resume processing the job – r_m, is calculated according to expression (12):

$$r_m = c_{ijk}^{m} \times z_{ijk}^{m} + t_{mb} \times \left(1 - z_{ijk}^{m}\right), m \neq r. \quad (12)$$

3. GENETIC ALGORITHM-BASED SOLUTION METHODOLOGY

A biologically inspired optimization method based on Genetic Algorithms (GA) is applied to obtain an optimal scheduling plan in accordance with the defined mathematical model. Each individual (chromosome) in the population consists of a primary substring, representing the operation sequence, and a secondary substring, encoding selected alternative manufacturing process plans. The initial population is generated based on the number of jobs and operations, while the fitness function is evaluated using the mathematical model for optimization defined in equation (1).

The algorithm then iteratively performs the core steps of selection (via roulette wheel), crossover, and mutation, applying operators to both substrings to generate new solutions. This process allows the algorithm to explore a wide solution space, maintain diversity, and converge toward optimal or near-optimal schedules. A detailed implementation procedure of the GA is provided in [16].

When a disturbance occurs, the GA generates an optimal rescheduling plan by repeating the same evolutionary steps, incorporating new jobs, excluding cancelled ones, or considering machine tool breakdowns. In this process, the initial population is generated to reflect the number of jobs remaining to be processed and the current state of the manufacturing system.

4. EXPERIMENTAL VERIFICATION AND DISCUSSION

To verify the mathematical models for the Dynamic Flexible Job Shop Scheduling Problem (DFJSSP), considering three types of disturbances, three experiments were conducted using 24 problems of varying complexity, which encompassed 18 benchmark jobs, as comprehensively presented in [17]. The networks of alternative manufacturing processes for all jobs included in the experiments were also adopted from this reference.

A genetic algorithm was used for both scheduling and rescheduling optimization, with the objective function makespan. The following genetic algorithm parameters were adopted for the initial scheduling phase: a population size of 120, a maximum number of generations set to 100, a crossover probability of pc = 0.6, and a mutation probability of p_m = 0.2. For the rescheduling phase, the parameters were adjusted to a population size of 100, with a maximum number of generations set to 80, while keeping the crossover and mutation probabilities unchanged. The proposed method and the corresponding experiments are implemented and tested in the MATLAB® environment. Experiment 1 addresses the scheduling problem labeled as Problem 21, focusing on the arrival of three new jobs at 30s. Experiment 2 involves scheduling for Problem 23 and analyzing the case of three jobs with cancellations occurring at 50s. Experiment 3 corresponds to Problem 22 and analyzes the breakdown of two machine tools that occurs at 40s.

The scheduling problems differ in terms of the number of jobs and the number of operations. For each job, three alternative process plans were generated and used during the scheduling optimization.

4.1 Arrival of new jobs

The scheduling of selected manufacturing processes for the initial set of jobs (2-3-5-6-7-9-10-11-13-14-16-18 – Problem 21) proceed without disturbance until the arrival of new jobs: 12, 15, and 17 into the manufacturing system at time $t_{new} = 30s$, at which point the rescheduling of the remaining operations is performed. In the first phase, prior to the arrival of the new jobs, a genetic algorithm generates the initial scheduling plan for the twelve jobs. After the arrival of the three new jobs, a new primary substring is formed, containing all unfinished operations of the initial jobs, as well as all operations of the new jobs, along with a new secondary substring that, compared to the initial one, also includes genes with information on the selected alternative manufacturing processes for the new jobs.

If the machine tool on which the first operation of a new job should begin is available at time $t_{new} = 30s$, processing starts immediately. Otherwise, the operation waits until the current processing is completed. For all operations that are yet to start, the earliest possible start time is the sum of the arrival time of the new jobs and the transportation time from the previous machine tool (if the previous and following operations are performed on different machines; otherwise, transportation time is disregarded). Operations that are in progress continue until completion. Afterward, according to the selected manufacturing process, the next operation either immediately starts on the same machine tool or the job is transferred to a different machine. In this way, the second phase of the algorithm performs rescheduling based on the new situation.

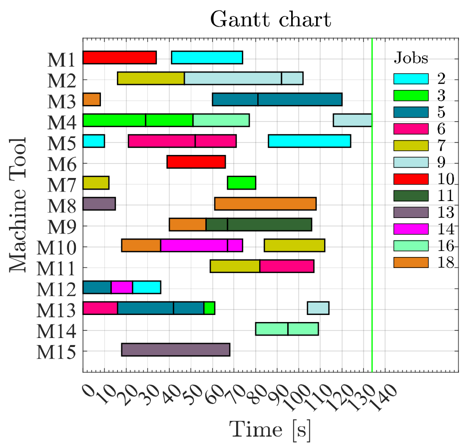

Figure 6. Problem 21 initial scheduling (*makespan* = 134s)

Figure 6 illustrates the Gantt chart representing the initial, optimal scheduling plan before the arrival of the new jobs, with a total processing time (makespan) of 134 seconds for all 12 jobs.

After the arrival of jobs 12, 15, and 17 at time $t_{new} = 30s$ and the following rescheduling, a new rescheduled plan is generated (Figure 7). It is observed that in this case, the makespan was 149s, indicating an increase in the total processing time due to the dynamic disturbance. Although the completion time is longer, the proposed approach successfully maintains the stability and functionality of the manufacturing system despite the disturbance caused by the arrival of new jobs.

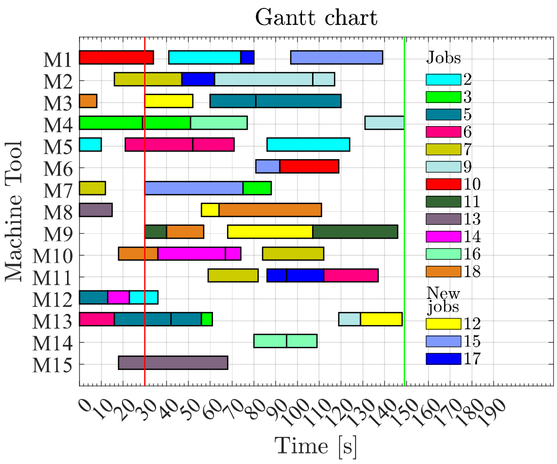

Figure 7. Rescheduling after arrival of new jobs 12, 15, and 17 at $t_{new} = 30s$ (*makespan* = 149s)

4.2 Job cancellations

The scheduling of selected manufacturing processes for the initial set of jobs (1-4-5-6-7-8-9-11-12-13-14-15-16-17-18 – Problem 23) proceed without disturbance until the cancellation of certain jobs: 1, 7, and 17 in the manufacturing system at time $t_{can} = 50s$, at which point the rescheduling of the remaining operations is performed. In the first phase, before the cancellation event, a genetic algorithm generates an initial scheduling plan with selected alternative manufacturing processes for all fifteen jobs. After the cancellation of the three jobs, a new primary substring is generated containing all unfinished operations of the jobs remaining in the system, excluding all remaining operations of the cancelled jobs, as well as a new secondary substring which, compared to the initial one, excludes as many elements (genes) as there are jobs that stop processing from the cancellation time onward.

Operations that have been completed before time $t_{can} = 50s$ remain fixed and are not subject to rescheduling. Operations that are in progress at the time of cancellation continue until completion if they belong to jobs that were not cancelled, while those belonging to cancelled jobs are interrupted immediately at that moment t_{can}. All operations of the cancelled jobs, whether in progress or not yet started, are removed from the rescheduling plan. For all remaining operations that have not yet begun, the earliest possible start time is set to the cancellation time t_{can} plus transportation time from the previous machine (if the machine tool is different). The second phase of the algorithm performs rescheduling based on this updated set of twelve jobs.

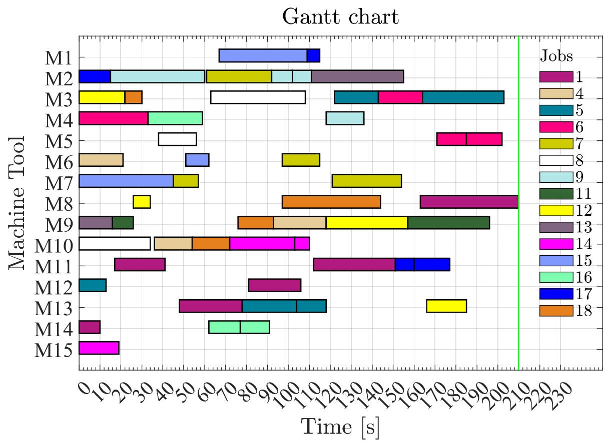

Figure 8. Problem 23 initial scheduling (*makespan* = 210s)

Figure 8 shows the Gantt chart representing the initial, optimal scheduling plan before the cancellation of jobs, with a total processing time (makespan) of 210s for all fifteen jobs. After the cancellation of jobs 1, 7, and 17 at time $t_{can} = 50s$ and the following rescheduling, a new rescheduled plan is formed (Figure 9).

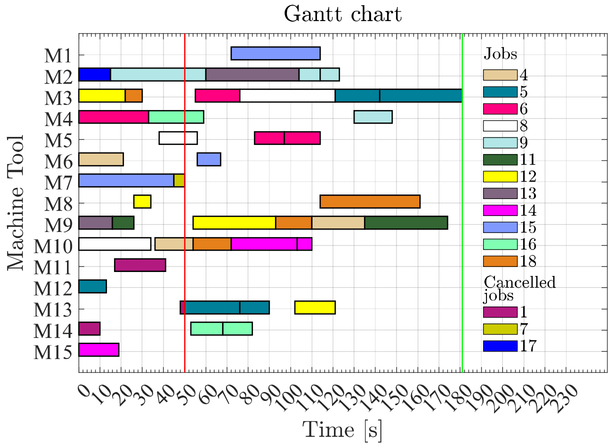

Figure 9. Rescheduling after cancellation of jobs 1, 7, and 17 at $t_{can} = 50s$ (*makespan* = 181s)

It is observed that in this case, the makespan was reduced to 181s, reflecting the removal of processing due to job cancellations. The proposed approach effectively adapts to dynamic changes and maintains operational efficiency within the manufacturing system, despite disturbances caused by job cancellations.

4.3 Machine tool breakdowns

The scheduling of the selected manufacturing processes for the initial set of jobs (2-3-4-5-6-8-9-10-11-12-13-14-16-17-18 – Problem 22) proceed without disturbance until the breakdown of machine tools M6 and M11 at time $t_{mb} = 40s$,at which point the rescheduling of the remaining operations is performed. In the first phase, prior to the breakdown event, a genetic algorithm generates an initial scheduling plan that selects alternative manufacturing processes for all fifteen jobs. After the machine tool breakdowns, all operations assigned to the broken machine that are still in progress are interrupted. Operations that were completed before t_{mb} remain fixed and are excluded from rescheduling. Operations that were in progress on the broken machine tools at t_{mb} are halted and rescheduled from the beginning, considering alternative manufacturing processes that exclude the broken machines. Operations that were in progress on machine tools that remain functional at the moment the disturbance occurs, continue processing according to the initial plan until completion.

A new primary substring is generated, containing the unfinished operations of all jobs, including those interrupted due to the breakdown, with updated processing routes that exclude the failed machine tools. A new secondary substring is also generated, including the updated selection of alternative manufacturing processes for the affected jobs. For all remaining operations yet to start, the earliest possible start time is set to the machine breakdown time t_{mb} plus the transportation time from the previous machine if the operation is not performed on the same machine tool. The second phase of the algorithm performs rescheduling based on this updated set of operations and manufacturing routes, adapting to the constraints imposed by the machine breakdown.

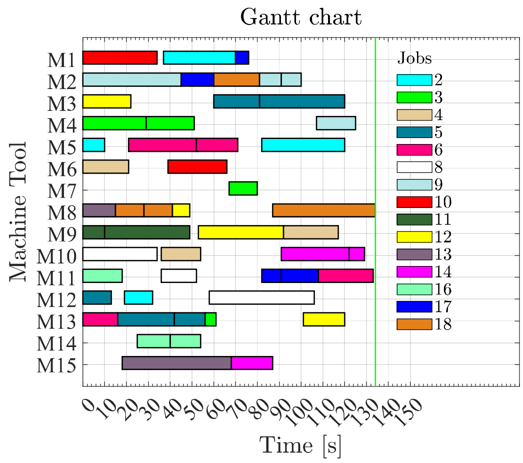

Figure 10. Problem 22 initial scheduling (*makespan* = 134s)

After the breakdowns at $t_{mb} = 40s$ and the following rescheduling, a new rescheduling plan is formed and shown in Figure 11.

Figure 10 shows the Gantt chart representing the initial, optimal scheduling plan before the machine tool breakdowns, with a total processing time (makespan) of 134s for all fifteen jobs.

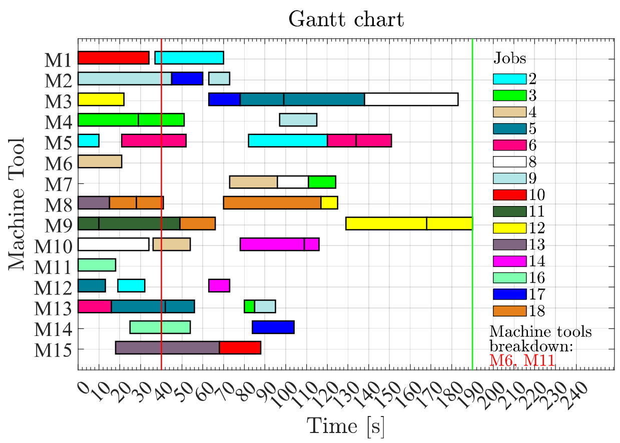

Figure 11. Rescheduling after machine tools M6 and M11 breakdown at $t_{mb} = 40s$ (*makespan* = 190s)

The results show that the makespan increased to 190s, indicating an extension of the total processing time due to the disturbance caused by the machine breakdowns. Despite the increase in the makespan, the proposed approach effectively maintains the stability and functionality of the manufacturing system by dynamically adapting the scheduling to the changed conditions.

5. CONCLUSION

This paper has presented a comprehensive approach to the Dynamic Flexible Job Shop Scheduling Problem (DFJSSP) under the influence of three types of disturbances: the arrival of new jobs, job cancellations, and machine tool breakdowns. Importantly, these disturbances are considered not only as typical dynamic events but also as potential consequences of cyber-attacks targeting manufacturing systems, which highlights the growing cybersecurity risks within modern production environments.

A genetic algorithm-based methodology was developed and implemented in the MATLAB® environment, enabling effective real-time rescheduling that minimizes total processing time (makespan). The experimental evaluation was performed on benchmark problems, demonstrating that the proposed approach maintains high scheduling efficiency and adaptability despite the presence of dynamic and cyber-induced disturbances. The results validate that the methodology can successfully manage and reduce the negative impact of such disturbances on production performance, thus supporting the Industry 4.0 concept of intelligent, flexible, and secure manufacturing systems.

In particular, three separate experiments were carried out to evaluate the rescheduling capabilities of the proposed solution under different types of disturbances. The first experiment analyzed the arrival of three new jobs into the system, requiring the integration of their operations into the existing schedule. The second experiment examined the cancellation of three jobs, focusing on the adaptation of the schedule by removing unfinished operations and redistributing available resources. The third experiment addressed the breakdown of two machine tools, requiring the reassignment of operations that were either ongoing or planned for the failed machines to alternative machines. Across all three cases, the genetic algorithm successfully adjusted the scheduling plans, demonstrating resilience, efficiency, and robustness in dynamically changing environments that could originate from or be aggravated by coordinated cyber-attacks. This research opens numerous paths for further investigation to improve the robustness and applicability of DFJSS in cybersecurity-aware manufacturing environments. First, future studies could incorporate the explicit modeling of machine tool downtime, including the waiting time until the failure is resolved and the machine becomes available again. This would provide a more realistic representation of manufacturing disturbances, enabling more precise rescheduling strategies.

Moreover, the simultaneous occurrence of multiple disturbances, possibly caused or aggravated by coordinated cyber-attacks, within the same manufacturing system poses complex rescheduling challenges. Future research could focus on developing integrated rescheduling frameworks that consider all such disturbances together, improving the system's resilience to sequential disturbances.

In terms of optimization approaches, it is valuable to explore and compare alternative metaheuristic algorithms (e.g., Ant Colony Optimization, Particle Swarm Optimization, or hybrid methods) for dynamic scheduling problems, to evaluate their relative performance, convergence behavior, and adaptability under dynamic and cybersecurity-related disturbances.

ACKNOWLEDGEMENT

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 17801, Cybersecurity of Motion Control Systems in Industry 4.0 - MCSecurity, as well as by the Ministry of Science, Technological Development, and Innovations of the Serbian Government under Contract No. 451-03-137/2025-03/200105.

REFERENCES

[1] D. A., Rossit, F., Tohmé, and M. Frutos, "Industry 4.0: smart scheduling", *International Journal of Production Research*, vol. 57, pp. 3802-3813, 2019.

[2] A. B. Avila, A. Zarreh, C. Saygin, H. D. Wan, and Y. Lee, "Evaluation of Dynamic Scheduling Policies against Cyber-attacks on an Open-Shop Manufacturing System using Simulation", *International Journal of Multidisciplinary and Current Research*, vol. 9, 2021.

[3] X. N. Shen, and X. Yao, "Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems", *Information Sciences*, vol. 298, pp. 198-224, 2015.

[4] K.B. Ali, S. Bechikh, A. Louati, H. Louati, and E. Kariri, "Dynamic Job Shop Scheduling Problem with New Job Arrivals using Hybrid Genetic Algorithm", *IEEE Access*, vol. 12, pp. 85338-85354, 2024.

[5] N. Zhu, G. Gong, D. Lu, D. Huang, N. Peng, and H. Qi, "An effective reformative memetic algorithm for distributed flexible job-shop scheduling problem with order cancellation", *Expert Systems with Applications*, vol. 237, p. 121205, 2024.

[6] N. Al-Hinai, and T. Y. Elmekkawy, "Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm", *International Journal of Production Economics*, vol. 132, pp. 279–291, 2011.

[7] A. Madureira, "A Genetic Approach for Dynamic Job-Shop Scheduling Problems", *MIC'2001 - 4th Metaheuristics International Conference*, no. March. pp. 41–46, 2001.

[8] N. Kundakcı, and O. Kulak, "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem", *Computers* and *Industrial Engineering,* vol. 96, pp. 31–51, June. 2016.

[9] Y. M. Wang, H. L. Yin, and K. Da Qin, "A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions", *The International Journal of Advanced Manufacturing Technology*, vol. 68, no. 5–8, pp. 1317–1326, 2013.

[10] S. Lin, E. Goodman, and W. Punch, "A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problem", *ICGA,* pp. 481–488, 1997.

[11] A. K. Jain and H. A. Elmaraghy, "Production scheduling/rescheduling in flexible manufacturing", *International Journal of Production Research*, vol. 35, no. 1, pp. 281–309, 1997.

[12] L. Yin, L. Gao, X. Li, and H. Xia, "An improved genetic algorithm with rolling window technology for dynamic integrated process planning and scheduling problem", *IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 414–419, 2017.

[13] M. Petrović, N. Vuković, M. Mitić, and Z. Miljković, "Integration of process planning and scheduling using chaotic particle swarm optimization algorithm", *Expert Systems with Applications*, vol. 64, pp. 569-588, 2016.

[14] S. Lv, and L. Qiao, "Process planning and scheduling integration with optimal rescheduling strategies", *International Journal of Computer Integrated Manufacturing*, vol. 27, no. 7, pp. 638–655, 2014.

[15] H. Xia, X. Li, and L. Gao, "A hybrid genetic algorithm with variable neighborhood search

for dynamic integrated process planning and scheduling", *Computers & Industrial Engineering*, vol. 102, pp. 99–112, 2016.

[16] Miljković, K., Petrović, M., Babić, B., Dynamic integrated process planning and scheduling based on genetic algorithms, Technical solution, 2021.

[17] M. M. Petrović, "Design of intelligent manufacturing systems by using artificial intelligence", Ph.D. dissertation, University of Belgrade-Faculty of Mechanical Engineering, Belgrade, Serbia, 2016.