Journal of Engineering Management and Systems Engineering
https://www.acadlore.com/journals/JEMSE

Metaheuristic Optimization for Stochastic Job Scheduling in Parallel Machine Systems with Uncertain Processing and Setup Times

Aleksandar Stanković[1*], Nikola Simić[2], Milica Stanković[3]

[1] Faculty of Mechanical Engineering, University of Niš, 18000 Niš, Serbia
[2] Faculty of Engineering, University of Kragujevac, 34000 Kragujevac, Serbia
[3] Faculty of Economics, University of Niš, 18000 Niš, Serbia

[*] Correspondence: Aleksandar Stanković (aleksandar.stankovic@masfak.ni.ac.rs)

Received: 05-19-2025 **Revised:** 07-10-2025 **Accepted:** 07-22-2025

Citation: A. Stanković, N. Simić, and M. Stanković, "Metaheuristic optimization for stochastic job scheduling in parallel machine systems with uncertai processing and setup times," *J. Eng. Manag. Syst. Eng.*, vol. 4, no. 3, pp. 206–217, 2025. https://doi.org/10.56578/jemse040304.

Abstract: The problem of job scheduling in parallel machine environments, where both processing times and setup times are characterized by stochastic variability, has been investigated with a focus on enhancing the efficiency of resource allocation in complex production systems. Job scheduling, as a critical component of operations research and systems engineering, plays a vital role in the optimization of large-scale, flexible manufacturing and service environments. In this study, a stochastic scheduling model has been formulated to minimize the maximum completion time (denoted as Ct_{max}), under the simultaneous influence of probabilistic job durations and setup times associated with tool preparation. The problem has been addressed using two prominent metaheuristic algorithms: Genetic Algorithm (GA) and Simulated Annealing (SA). These methods were selected due to their demonstrated capacity to navigate large, non-deterministic search spaces efficiently and their adaptability to multi-constraint scheduling problems. A comparative analysis has been conducted by applying both algorithms under identical initial conditions, with algorithmic performance evaluated in terms of solution quality, computational efficiency, and robustness to input variability. The model incorporates key practical considerations, including randomized setup times which are often neglected in conventional deterministic scheduling models, thereby improving its relevance to real-world industrial settings. The formulation of the problem allows for additional constraints and objectives to be flexibly integrated in future research, including resource conflicts, machine eligibility constraints, and energy-aware scheduling. Empirical results suggest that while both algorithms are effective in deriving near-optimal schedules, notable differences exist in convergence behavior and sensitivity to parameter tuning. The findings offer critical insights into the comparative strengths of GA and SA in managing the stochastic nature of parallel machine scheduling problems. By advancing a robust metaheuristic framework that accounts for real-world uncertainties, this study contributes to the ongoing development of intelligent scheduling systems in systems engineering, manufacturing logistics, and automated production planning.

Keywords: Parallel machine scheduling; Stochastic processing times; Stochastic setup times; Metaheuristic optimization; Genetic Algorithm (GA); Simulated Annealing (SA); Resource allocation; Systems engineering

1 Introduction

Efficient production resource planning entails scheduling jobs at appropriate times to maximize productivity and streamline manufacturing operations. The challenge of job scheduling in manufacturing involves multiple stages and focuses on the sequential assignment of tasks to machines, aiming to reduce the value of the objective function while enhancing overall production performance [1]. As a critical component of production system management, production planning establishes key objectives, managerial strategies, and operational methodologies to ensure system effectiveness and alignment with organizational goals. Due to its complexity and dynamic nature, the planning process is typically categorized into short-term, medium-term, and long-term phases, each depending on the specific goals, procedures, and strategic direction of the company [2].

In previous research, many scheduling models assume that all machines and jobs are available for use in real time. However, in practice, full machine availability without certain breakdowns or periods of unavailability is rarely encountered [3]. Machine constraints and availability may result from resource limitations, preventive and corrective maintenance, machine failures, and various other factors affecting machine availability. These constraints reflect the dynamic nature of production planning environments [4]. To assess the current state of production, improve productivity, and identify deficiencies in the manufacturing environment, it is necessary to analyze the entire production process. By simulating the current state of production and conducting a detailed analysis, various deficiencies in the observed manufacturing process can be identified, including critical bottleneck areas, material handling inefficiencies, inadequate machine layout, suboptimal job scheduling, worker delays, material delays, and many other factors affecting production productivity. The market problem is presented in Figure 1:

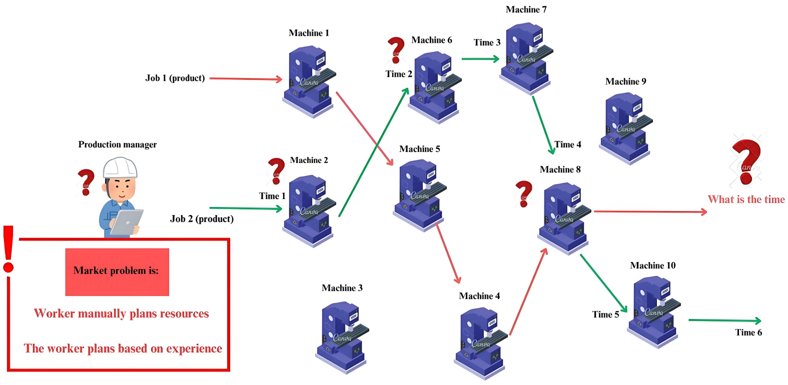

Figure 1. Current state of production scheduling: manual planning based on experience

Following a thorough analysis, an appropriate mathematical model is proposed to optimize the production process and eliminate inefficiencies. Identifying key causes and deficiencies based on research literature, followed by the application of scientifically grounded methods, represents a systematic and scientific approach to solving the problem of job scheduling and planning in manufacturing [5]. Support for operational production planning is elaborated across multiple segments in this research, with the primary goal of improving production conditions and developing new approaches, methods, and techniques for addressing production planning challenges. Planning is a crucial management process that serves as the foundation for a company's economic growth, provided that clear objectives and strategies are in place. Therefore, integrating a well-defined work methodology into the planning process is essential to ensure that all operations follow established rules and to monitor production efficiency, one of the most critical parameters in production planning [6]. The planning of all subprocesses involves strategies and methodologies for allocating all jobs within the manufacturing environment. With advancements in information and advanced technologies, which play a key role in production planning, various methods are used in the field of resource planning and scheduling. The most well-known methods in this context include [7, 8]:

• Exact Methods: These methods are characterized by precisely defined mathematical functions and the determination of optimal solutions depending on the size of the analyzed data. The primary techniques in resource planning and scheduling problems include nonlinear, linear, dynamic, integer, and mixed-integer programming techniques.

• Heuristic Methods: These methods do not guarantee finding an optimal solution but efficiently determine sufficiently good solutions in real-time.

• Metaheuristic Methods: The most widely used algorithms in this category for solving resource planning and scheduling problems include GA, Ant Colony Optimization (ACO), SA, Tabu Search (TS), and many others.

• Constraint Programming Techniques: These can be applied to various scheduling problems depending on the optimization objective. Constraint programming with an objective function relates to determining the completion time of the last processed product, denoted as Ct_{max}, which represents the total criterion function value.

• Simulation Methods: These methods have extensive applications, enabling the representation of complex systems in great detail and allowing interactions between components during the simulation process. This capability is one of their key advantages over other methods. Simulation methods are widely used in optimization domains, including resource planning and scheduling. Simulation-based resource planning has significant applications in system planning and control, producing a detailed work plan as the final output.

The problem of job scheduling and planning has broad applications in manufacturing environments. The proposed optimization approach, based on the SA and GA algorithms, provides optimal results. By analyzing graphical optimization results, it is possible to examine job sequencing in detail as well as the total duration of production processes. The following section of the paper presents a literature review on parallel machine scheduling, while the subsequent part is dedicated to a detailed description of the methodology and the application of the proposed mathematical model.

2 Literature Review

This chapter presents a review of relevant literature related to the discussed problem. One of the first researchers to address the issue of resource planning and scheduling using exact methods was Harari and his collaborators in 1957 [9]. Thomalla [10] states that the implementation of automated systems largely depends on the efficient use of resources, where effective algorithms and process planning can significantly increase and ensure a return on investment. A key area of research concerns job scheduling optimization within a given time frame, where each operation is assigned to a machine with potentially varying efficiency and processing time. The objective of this research was to minimize the sum of squared job values, using the Lagrangian relaxation. In the study [11], metaheuristic algorithms such as SA and tabu search were introduced to obtain approximate solutions in various optimization scenarios, focusing particularly on the problem of parallel machine scheduling. The results demonstrate the efficiency of these methods in planning and scheduling processes. Similarly, Thomalla [10] examines job scheduling on identical parallel machines, where a mathematical model incorporating stochastic processing times was developed. The Branch and Bound algorithm were applied to solve this problem, with the primary goal of maximizing service levels for customers while minimizing delivery times.

Heuristic approaches to job planning and scheduling have also been widely explored in the literature. Angell [11] analyzes the scheduling of n jobs on mmm parallel machines to minimize the total processing time (Ct_{max}). The study investigates the problem of parallel machine scheduling with job splitting, assuming that part of a job can be processed simultaneously on two different machines. Furthermore, Liu and Zhang [12] introduce a hybrid GA with modified crossover and mutation operators based on critical schedules. Additionally, Xing and Zhang [13] examine the problem of parallel machine scheduling with multiple resources, providing a more realistic depiction of scheduling processes. The objective of this research is to minimize the objective function Ct_{max} and reduce machine and mold unavailability in terms of maintenance. On the other hand, the study [14] investigates scheduling in real production environments, considering rescheduling and the impact of unforeseen events such as machine failures. The proposed approach is based on an idle-time insertion algorithm, which allows for more flexible job scheduling.

In the study [15], the problem of scheduling n independent jobs with predefined deadlines and release times on multiple parallel machines is analyzed. The study [16] proposes a GA for solving the parallel machine scheduling problem, demonstrating that the proposed methodology is highly effective compared to alternative approaches. Moreover, in the study [17], an SA algorithm is applied to the scheduling of parallel machines with setup times, incorporating a restricted search strategy to reduce computational complexity. The study [18] addresses the scheduling of a set of jobs on two parallel machines to minimize the total time span as an objective function. In contrast, Ying et al. [19] examine the problem of parallel machine scheduling with availability constraints, proposing and detailing a dynamic time-programming algorithm and a fully polynomial-time approximation scheme for a two-machine scheduling problem. The work [20] explores the parallel machine scheduling problem in manufacturing environments, emphasizing the optimization of job completion times to enhance production efficiency.

The research [21] considers the scheduling of parallel computing systems with a shared server, highlighting the complexity of the problem even when the number of machines is fixed. A metaheuristic approach is applied to address this challenge. In the study [22], the authors argue that most studies in resource planning and scheduling within parallel machine models treat machines as singular resources. However, in real-world production environments, additional factors such as mold availability and maintenance significantly impact scheduling. Finally, Cheng et al. [23] analyze the problem of uniform machines with parallel scheduling, where the authors develop a mixed-integer programming model aimed at minimizing time intervals [24].

Based on a detailed analysis of the existing literature, this study utilized the SA and GA metaheuristic algorithms to solve the problem of parallel machine scheduling. The following chapter describes the applied methodology, a mathematical model for the defined problem, and a case study based on the proposed model.

3 Methodology

Metaheuristics represent an advanced optimization technique aimed at finding either the optimal solution or a sufficiently good solution for a given optimization problem. Unlike conventional algorithms and iterative methods, metaheuristics do not guarantee finding a globally optimal solution for the problem at hand. Stochastic optimization is often employed, meaning that the solution found depends on the set of generated random variables. In the field of combinatorial optimization, metaheuristics enable the discovery of optimal solutions with lower computational power requirements compared to other algorithms and methods. As such, they are valuable in solving a variety of optimization problems. Figure 2 presents one of the ways of classifying metaheuristics [25, 26].

3.1 Simulated Annealing (SA)

This algorithm belongs to the group of algorithms that are increasingly used to solve difficult NP problems due to its efficiency in solving them. The basic characteristic and principle of the SA algorithm is that the temperature of the material decreases to the allowable state, corresponding to the lowest temperature, while the material's annealing process involves predefining temperatures at which the material will be exposed for a certain period. Some of the characteristics of this method, as outlined by the study [27], are: (1) the quality of the solution does not depend on the initial solution; however, if the initial solution is not well-defined, the time required to obtain the final solution increases significantly; (2) it can also be negatively affected; (3) it can be applied to solve various discrete and continuous optimization problems. The SA method was first proposed in 1983 by the renowned scientist Kirkpatrick and his collaborators [28]. The application of the SA algorithm in solving complex optimization problems has also inspired the development of other metaheuristic algorithms to solve similar problems, as seen in this study. The pseudo-code of the SA algorithm is presented in Algorithm 1.

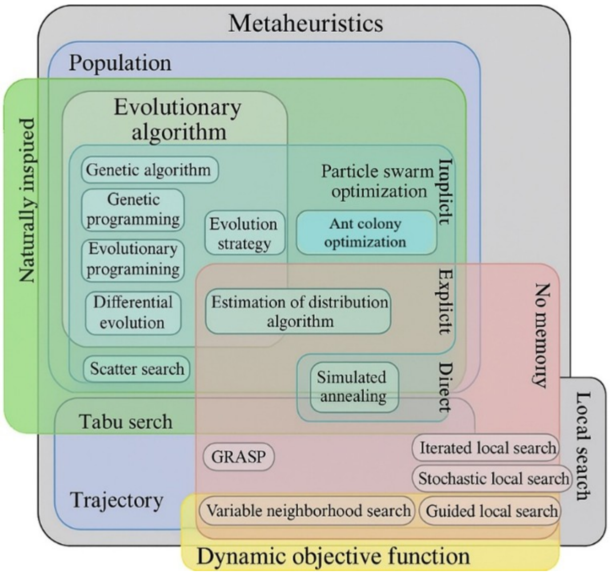

Figure 2. Different ways of classifying metaheuristics

Algorithm 1: Pseudo code of SA

Define the objective function $f(x)$ – e.g., minimizing makespan (Ct_{max}), total tardiness, or cost
Generate an initial solution x (e.g., an initial schedule of operations on machines)
Initialize the starting temperature $T0$ and final temperature Tf
Set the maximum number of iterations N and the cooling factor α ($0 < \alpha < 1$)
Set current solution $x\_curr = x$ and $f\_curr = f(x)$
Set the best solution $x\_best = x\_curr$ and $f\_best = f\_curr$
Initialize iteration counter $n = 0$ and $T = T0$
While $T > Tf$ and $n < N$:
- Generate a new solution x_new in the neighborhood of x_curr
  (e.g., swap two operations, change the machine assignment, etc.)
- Compute $f\_new = f(x\_new)$
- Compute $\Delta f = f\_new - f\_curr$
If $f\_new < f\_curr$:
  - Accept the new solution: $x\_curr = x\_new$, $f\_curr = f\_new$
Else:
  - Compute $p = \exp(-\Delta f/T)$
  - Generate a random number $r \in [0, 1]$
  - **If** $p > r$:
    - Accept the worse solution: $x\_curr = x\_new$, $f\_curr = f\_new$
If $f\_curr < f\_best$:
  - Update the best solution: $x\_best = x\_curr$, $f\_best = f\_curr$
- Decrease the temperature: $T = \alpha * T$
- Increase iteration count: $n = n + 1$
End while
Return x_best and f_best as the optimal schedule and its objective value

3.2 Genetic Algorithm (GA)

GA is a powerful optimization technique commonly applied to complex, nonlinear, or non-differentiable problems. Initially introduced by Holland in the 1970s, the GA simulates natural evolutionary processes, specifically selection, mutation, and crossover, to identify optimal solutions. In this algorithm, potential solutions are represented as individuals within a population, each characterized by a "genetic code" that defines the particular solution. Through successive generations, the algorithm evolves the population toward more optimal solutions by applying principles of natural selection. The GA algorithm is particularly beneficial for exploring large search spaces and mitigating the risk of premature convergence, which could otherwise hinder the discovery of a globally optimal solution. By incorporating mechanisms such as mutation and crossover, the algorithm introduces diversity into the population, enabling it to escape local optima and continue searching for better solutions. The pseudo-code for the GA algorithm is outlined in Algorithm 2.

Algorithm 2: Pseudo code of GA

Define the objective function $f(x)$, where $x = (x_1, x_2, ..., x_n)T$ represents a solution vector
    for resource allocation and scheduling
Encode each solution x into a chromosome (e.g., binary or integer string representing
    job sequences, machine assignments, start times, etc.)
Define the fitness function F such that $F \propto f(x)$ for maximization (e.g., minimizing makespan,
    maximizing resource utilization, or minimizing total delay)
Generate an initial population of chromosomes (solutions)
Initialize crossover probability pc and mutation probability pm
While ($t < Max$ number of generations or stopping criteria not met)
    For each pair of selected parent chromosomes:
        If $rand < pc$, perform crossover to generate offspring
        If $rand < pm$, apply mutation to introduce variability
    Evaluate the fitness of new offspring
    If the fitness of an offspring is better than the parent(s), accept the offspring
    Apply selection strategy (e.g., tournament or roulette-wheel) to choose individuals
        for the next generation based on fitness
    Update the current best solution found so far
    Increment generation counter t
end while
Decode the final solution(s) from chromosomes into real resource plans and schedules
Visualize or interpret results for decision-making (e.g., Gantt chart or schedule table)

4 Mathematical Formulation

This section of the paper presents a model for job scheduling on a parallel set of machines. This type of application can be highly beneficial in a production system during process optimization. In this case, the scheduling is performed on similar types of machines, where the processing of n jobs on a set of machines can be carried out simultaneously [29, 30]. An example of this type of job scheduling on a parallel set of machines is illustrated in Figure 3 [30]. In this context, the formulation of the parallel machine scheduling problem with execution times can be defined as follows. Let $N = \{1, ..., n\}$ be the set of jobs and $M = \{1, ..., m\}$ be the set of machines. The problem consists of scheduling n jobs on m machines, subject to certain rules and constraints during the scheduling process [1, 30].

The following section presents the basic notation of the mathematical model [1, 30]:

m - machines,
n - jobs,
$stohpt_j$ – stochastic processing time,
Ct_{max} – stochastic objective function: maximum productivity with minimum time.

where the variable has the following form:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is processed on machine } i \\ 0 & \text{else} \end{cases}$$

with constraints:

$$\sum_{j=1}^{n} x_{ij} stohpt_j \leq Ct_{\max}, \quad i = 1, \ldots, m \tag{1}$$

$$stohpt_j \leq Ct_{\max}, \quad j = 1, \ldots, n \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, \ldots, n \tag{3}$$

$$x_{ij} \geq 0, \quad i = 1, \ldots, m; \; j = 1, \ldots, n \tag{4}$$

(a) The objective is to minimize the maximum completion time of the schedule commonly referred to as the makespan, denoted as Ct_{max}.

(b) Each job $j \in N$ must be processed exactly once and exclusively on a single machine $k \in M$.

(c) Every job $j \in N$ has a processing time p_j, which varies depending on the machine $k \in M$ to which it is assigned.

(d) The model allows for the initiation of processing newly available jobs that may not have been ready for execution earlier (preemption – $prmp$).

(e) A continuous and safe operational flow is ensured throughout the entire machine line during job execution (nwt – no-wait constraint).

(f) Taking into account the given mathematical formulation and its constraints, the problem can be denoted using the standard scheduling notation as:

$$P_m \mid nwt, prmp, stohpt_j \mid Ct_{\max}$$

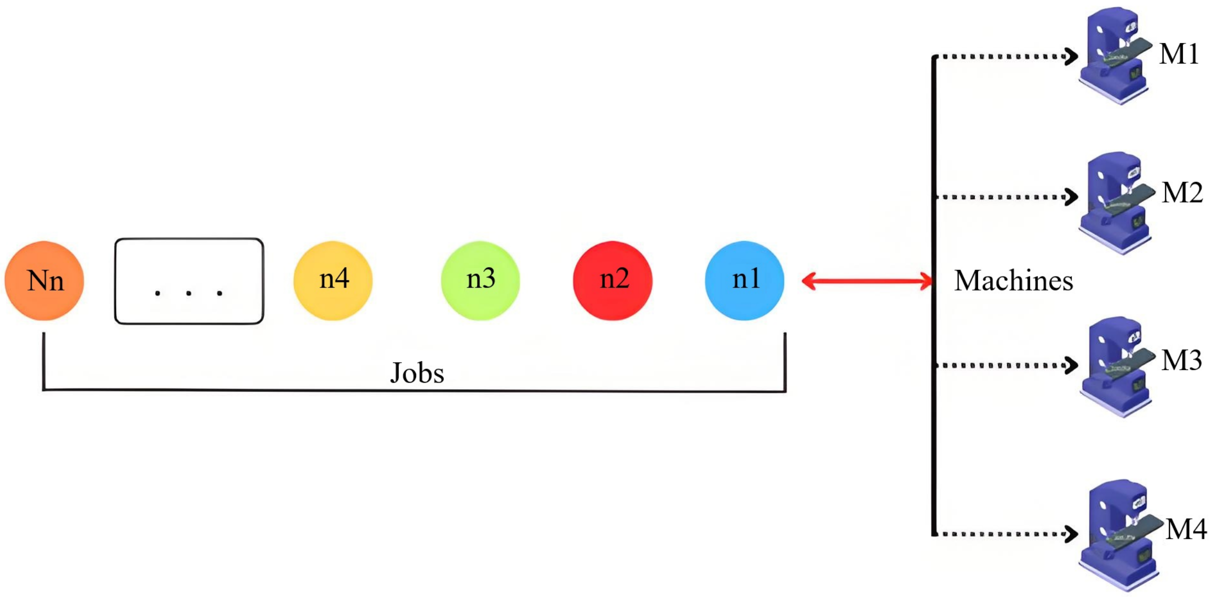

Figure 3. Graphical representation of the model in parallel machine scheduling

This scheduling problem can be understood as the allocation of m parallel machines for processing a set of jobs, under the condition that a stable and uninterrupted workflow is maintained across the machine line. In cases where processing is halted, the system allows for assigning and executing newly available jobs at that time. The optimization objective is to reduce the overall completion time of all jobs, thus achieving maximal scheduling efficiency.

5 Case Study

This section presents the problem of parallel machine scheduling aimed at achieving maximum production efficiency. The SA and GA were employed to optimize job scheduling, in order to ensure the most effective allocation of tasks across a set of parallel machines and thereby maximize productivity. The mathematical formulation of the problem is thoroughly detailed in Chapter 4. The following section provides an overview of the input parameters used in the algorithms. It is important to note that the adopted solutions for each instance are based on the best result obtained for the defined problem. Table 1 shows the job processing times for all machines [30], while Table 2 presents the job setup times on the machines [30].

The following section presents the input parameters used for the optimization processes of the GA and SA, as shown in Table 3.

Table 1. Stochastic processing times in machines M_1 and M_8

N	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
1	29.4	17.2	18.9	25.1	38.3	27	18.6	35.2
2	23.1	52.7	50.4	59.2	55.9	30.3	15.8	17.4
3	25.5	39.6	55.2	10.9	42.1	19.4	23.8	25.6
4	45.3	38.2	36.5	49.1	22.4	12.6	33.7	35.3
5	55.5	56.8	18.3	51.6	12.2	30.4	43.9	25.2
6	48.9	24.1	40	54.2	32.5	40.7	33.3	45.8
7	48.2	27.9	18.4	15.3	42.8	50.1	53.4	27.6
8	23.5	52.6	50.1	59.7	58.4	24.2	45.6	34.5
9	25.3	39.8	35.2	17.6	28.9	34.1	49.8	64.1
10	45.7	38.4	36.9	49.3	25.6	44.8	40.2	44.4

Table 2. Stochastic setup times in machine M_1 and M_8

M_1	**1**	**2**	**3**	**...**	**8**	**9**	**10**	M_2	**1**	**2**	**3**	**...**	**8**	**9**	**10**
1	5.53	7.12	5.29	...	3.95	2.97	8.12	**1**	7.26	7.06	7.15	...	6.98	1.98	5.06
2	3.04	5.44	8.23	...	1.94	7.09	3.74	**2**	7.02	7.22	3.11	...	5.97	6.05	2.87
3	5.23	8.2	6.26	...	7.94	4.46	2.44	**3**	6.62	2.1	2.13	...	6.97	2.23	2.22
4	3.05	4.11	2.73	...	8.36	4.88	4.91	**4**	2.02	5.06	6.87	...	6.18	6.94	4.95
5	1.69	6.57	2.49	...	8.23	2.52	3.94	**5**	5.84	3.79	5.74	...	5.12	7.76	2.97
6	6.73	4.12	3.85	...	8.09	2.81	2.89	**6**	4.87	5.06	6.92	...	6.05	6.9	7.95
7	2.8	6.89	6.76	...	7.14	5.73	3.02	**7**	2.9	3.95	5.88	...	5.07	1.86	2.01
8	5.22	7.04	7.34	...	4.83	3.91	8.02	**8**	2.11	4.02	4.17	...	3.91	1.95	4.01
9	5.65	4.27	7.14	...	2.95	7.68	6.32	**9**	6.83	3.14	4.07	...	6.97	5.84	5.16
10	3.88	3.37	4.06	...	4.54	2.04	5.12	**10**	2.94	3.18	7.03	...	8.27	7.02	7.06
M_3	**1**	**2**	**3**	**...**	**8**	**9**	**10**	M_4	**1**	**2**	**3**	**...**	**8**	**9**	**10**
1	6.35	5.20	8.15	...	8.40	2.15	3.30	**1**	7.45	7.10	8.20	...	2.35	3.60	8.10
2	4.40	6.80	5.25	...	6.25	3.10	2.95	**2**	8.30	5.60	2.35	...	4.25	7.40	6.20
3	5.10	8.30	5.05	...	5.60	2.45	6.20	**3**	3.75	3.20	2.50	...	8.50	4.10	6.75
4	6.75	7.60	4.45	...	7.35	5.70	8.10	**4**	7.90	5.35	4.80	...	5.30	4.25	7.80
5	8.50	4.10	5.90	...	5.15	3.60	8.50	**5**	3.55	5.75	6.15	...	4.45	3.90	2.15
6	7.25	7.70	5.55	...	5.45	4.25	3.10	**6**	8.10	8.50	5.60	...	3.65	7.25	8.40
7	2.30	2.45	2.60	...	6.80	4.40	5.75	**7**	7.65	7.40	8.00	...	5.10	6.55	4.85
8	6.90	2.15	8.35	...	4.90	2.35	4.60	**8**	4.25	8.35	5.20	...	8.20	7.60	2.30
9	5.05	2.95	2.40	...	3.70	5.25	5.35	**9**	7.30	6.55	5.10	...	2.75	7.15	7.50
10	5.80	3.35	5.10	...	6.15	6.10	3.45	**10**	5.80	4.15	5.45	...	5.45	2.85	7.10
M_5	**1**	**2**	**3**	**...**	**8**	**9**	**10**	M_6	**1**	**2**	**3**	**...**	**8**	**9**	**10**
1	4.50	7.30	5.75	...	7.45	5.30	3.25	**1**	7.35	7.40	7.10	...	4.20	7.10	2.35
2	3.25	5.80	8.10	...	2.60	7.15	4.40	**2**	7.25	7.15	3.55	...	6.15	6.25	3.40
3	6.40	8.25	6.60	...	8.20	4.50	2.10	**3**	7.20	2.30	2.25	...	7.30	2.40	2.15
4	3.15	4.10	3.50	...	8.55	5.35	5.65	**4**	2.15	5.45	7.25	...	6.50	7.05	5.25
5	2.70	7.45	3.20	...	8.10	2.85	4.25	**5**	6.10	4.20	6.30	...	5.35	8.20	3.45
6	7.10	4.35	4.25	...	8.35	3.40	3.75	**6**	5.55	5.25	7.35	...	6.30	7.40	8.10
7	3.60	7.20	7.10	...	7.25	6.30	3.50	**7**	3.40	4.10	6.50	...	5.55	2.60	2.30
8	5.55	7.65	7.35	...	5.50	4.65	8.30	**8**	2.35	4.45	4.20	...	4.25	2.45	4.20
9	6.25	4.10	7.80	...	3.75	8.55	6.60	**9**	7.30	3.25	4.60	...	7.15	6.40	5.05
10	4.45	3.75	4.60	...	4.20	2.70	5.15	**10**	3.15	3.70	7.40	...	8.10	7.35	7.25
M_7	**1**	**2**	**3**	**...**	**8**	**9**	**10**	M_8	**1**	**2**	**3**	**...**	**8**	**9**	**10**
1	6.35	5.20	8.10	...	8.10	2.25	3.40	**1**	7.45	7.35	8.20	...	2.35	3.25	8.10
2	4.15	6.25	5.30	...	6.20	3.15	2.30	**2**	8.10	5.25	2.15	...	4.20	7.30	6.15
3	5.20	8.15	5.25	...	5.35	2.40	6.25	**3**	3.25	3.35	2.10	...	8.25	4.15	6.20
4	6.10	7.05	4.30	...	7.25	5.10	8.30	**4**	7.30	5.15	4.20	...	5.10	4.30	7.25
5	8.30	4.20	5.10	...	5.45	3.25	8.20	**5**	3.30	5.25	6.10	...	4.10	3.20	2.30
6	7.25	7.10	5.20	...	5.35	4.10	3.30	**6**	8.35	8.25	5.30	...	3.20	7.25	8.15
7	2.40	2.60	2.15	...	6.10	4.20	5.15	**7**	7.50	7.35	8.25	...	5.25	6.10	4.20
8	6.05	2.35	8.25	...	4.30	2.50	4.35	**8**	4.20	8.10	5.25	...	8.30	7.35	2.25
9	5.35	2.45	2.10	...	3.15	5.30	5.50	**9**	7.35	6.25	5.15	...	2.45	7.20	7.15
10	5.25	3.15	5.20	...	6.10	6.20	3.40	**10**	8.20	4.30	5.10	...	5.30	2.40	7.20

Table 3. Input parameters used for the optimization processes of the GA and SA

GA Parameters			
$MaxIt$	$nPop$	M_{ij}	$Cross$
500	250	0.02	0.4
SA Parameters			
$MaxIt$	$MaxIt2$	T_{ij}	$alpha$
500	25	100	0.8

In Tables 1 and 2 provide the essential input parameters for the defined mathematical model, forming the foundation for the problem-solving approach. From these tables, it is evident that the examined parallel machine scheduling problem involves $M = 8$ machines and $N = 10$ jobs [30], highlighting a moderately complex scheduling environment. Building on these inputs, the subsequent section presents the results of the conducted research, where the focus is on the performance of the applied optimization algorithms. The graphical representation of job allocation across machines, illustrated in Figures 4, 5, 6 and 7, offers a clear visualization of the generated schedules and their efficiency. The experimental analysis reveals that both the SA and GA yield highly effective solutions for the given scheduling problem. This is validated by the achieved values of the objective function, which confirms the models' capability to optimize job distribution. Moreover, the results underline the importance of carefully selected algorithm parameters, such as temperature and cooling rate in SA, which significantly influence the success and convergence of the optimization process. In the case of the GA, parameters such as population size, crossover rate, and mutation rate play a critical role in guiding the evolutionary process toward optimal or near-optimal solutions. Fine tuning these parameters enhances the algorithm's ability to effectively explore the solution space and avoid premature convergence, ultimately contributing to improved scheduling performance.

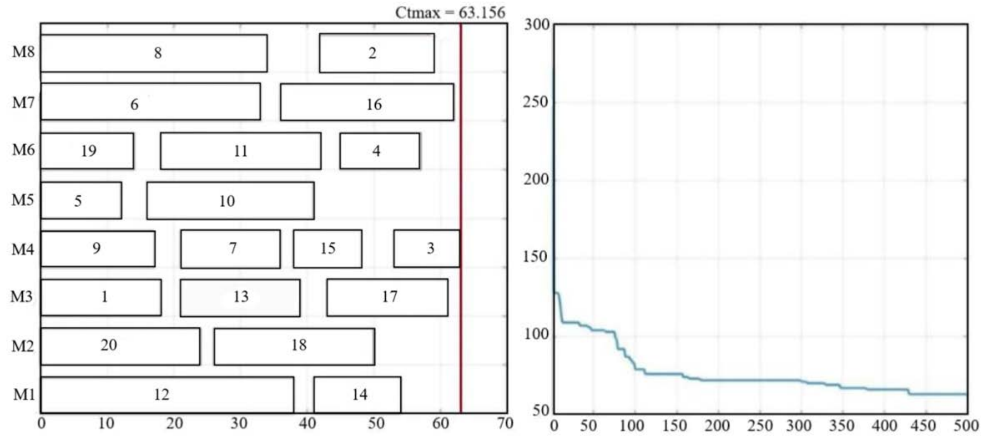

Figure 4. Graphical results based on GA algorithm: problem $N = 20$, $M = 8$

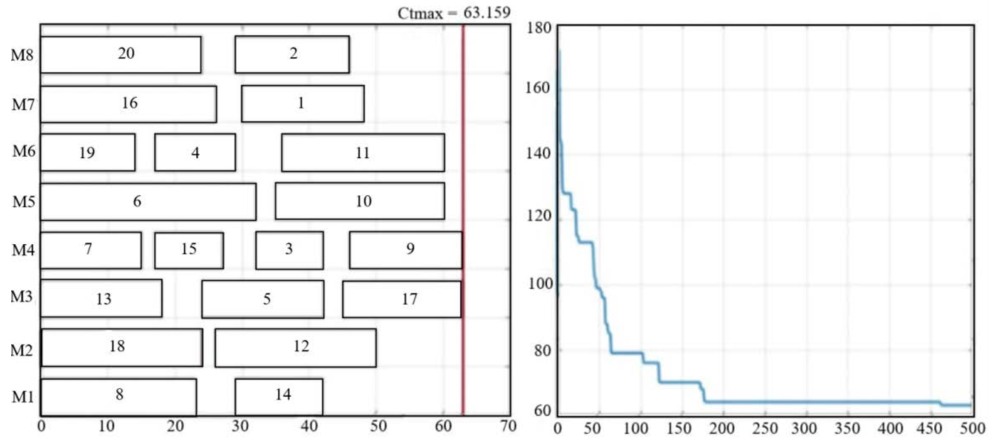

Figure 5. Graphical results based on SA algorithm: problem $N = 20$, $M = 8$

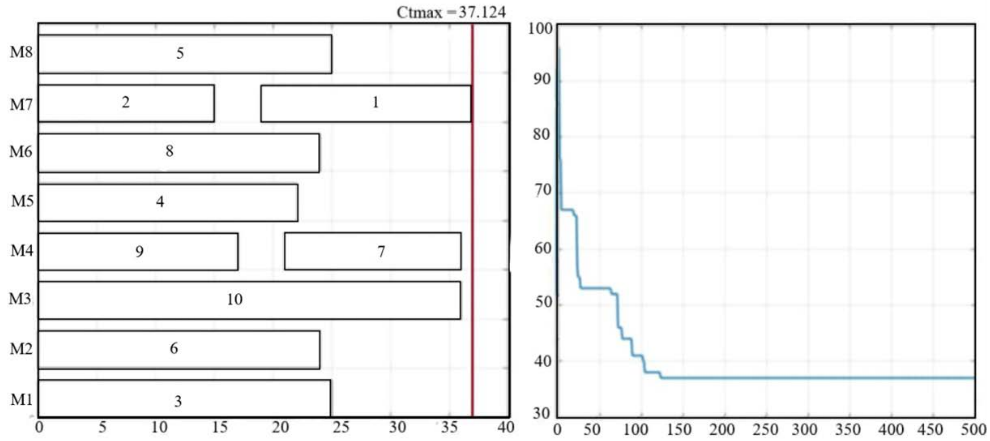

Figure 6. Graphical results based on GA algorithm: problem $N = 10$, $M = 8$

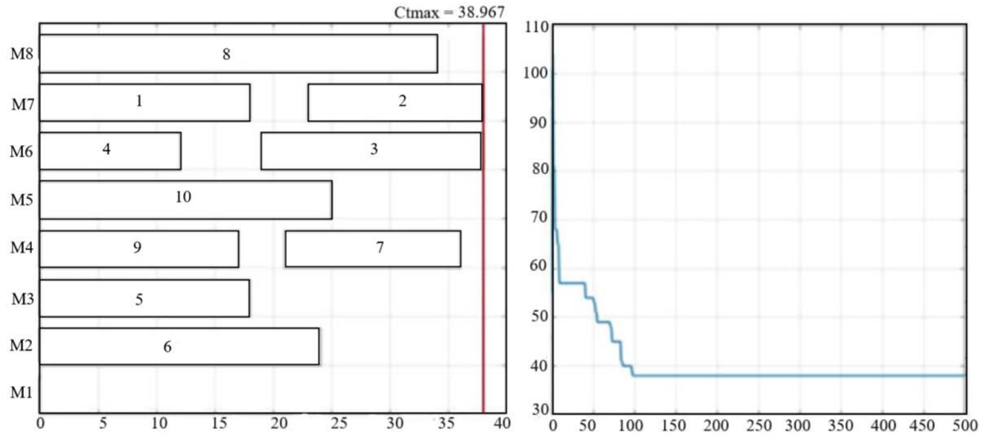

Figure 7. Graphical results based on SA algorithm: problem $N = 10$, $M = 8$

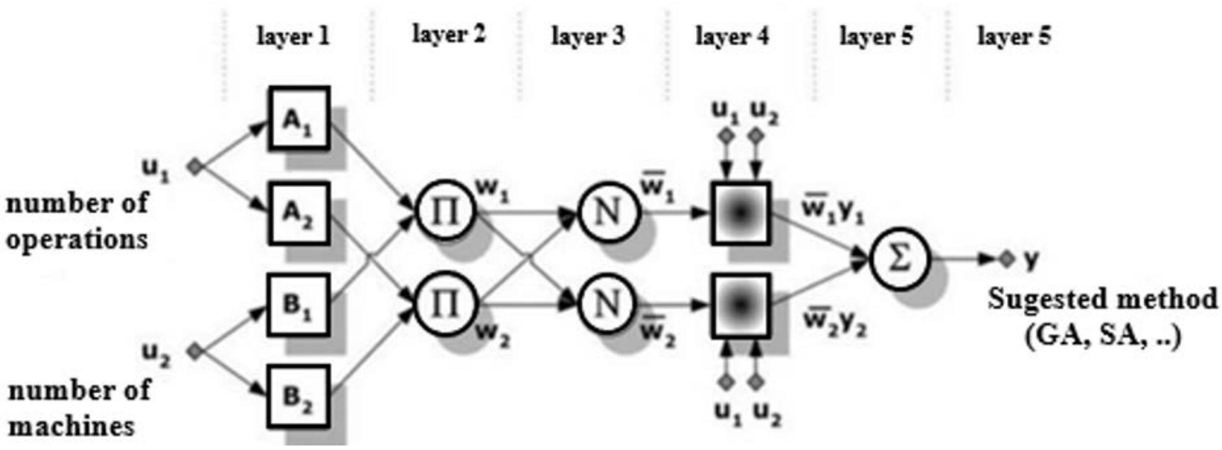

Figure 8. ANFIS structure with two inputs, five layers, and a single output is employed to recommend the most suitable optimization method among the evaluated alternatives

The experimental results demonstrate the effectiveness of both algorithms in solving the given problem. The optimization procedures were applied to multiple problem instances; however, due to the extensive volume of input data, detailed input parameters such as processing times, machines, and job sets are presented for a single representative case, as shown in Tables 1 and 2. A comparative analysis of the GA and SA algorithms confirms the success of both approaches. For the initial dataset, the objective function value for the GA algorithm is Ct_{max}=

63.156, and the SA algorithm function value is Ct_{max}= 63.159. Additionally, the iterative solution illustrated in one of the figures reveals that the GA reached the optimal solution using fewer iterations than SA.

Furthermore, results shown in Figures 5 and 6 indicate that GA produced a schedule with Ct_{max}= 37.124, while SA achieved Ct_{max}= 38.967. In this case, both algorithms required a similar or identical number of iterations to converge. These findings suggest that both GA and SA provide sufficiently robust and effective solutions for the Parallel Machine Scheduling model. Their adaptability and performance make them promising tools for future applications in similar scheduling problems.

In order to avoid this kind of mutual comparison of metaheuristic algorithms, the neuro-fuzzy approach conceptualized on the Adaptive Neuro-Fuzzy Inference System (ANFIS) is proposed, the basis of which is the selection of the optimal algorithm for the observed problem. The application of the ANFIS significantly reduces the time required to select the optimal algorithm for solving specific problems of planning and resource scheduling.

ANFIS combines the strengths of artificial neural networks and fuzzy logic, enabling adaptive learning based on input data and fuzzy rules. In this way, the system automatically recommends the most efficient optimization method under given conditions, thereby accelerating the entire optimization process and increasing its overall effectiveness. The graphic representation of such a system is given in Figure 8 [5].

6 Conclusions

The problem of resource planning and job scheduling represents one of the most complex challenges within the domain of combinatorial optimization. Efficient scheduling in a production system plays a crucial role in overall manufacturing performance, directly influencing productivity, throughput, and resource utilization. One of the primary objectives of the presented model based on Parallel Machine Scheduling is to maximize productivity by minimizing the total processing time of all jobs.

The proposed optimization approach aims to achieve high machine utilization while minimizing the overall completion time. The results obtained confirm the effectiveness of applying metaheuristic algorithms such as GA and SA in solving the parallel machine scheduling problem. These models enable improved resource utilization and a reduction in stochastic makespan, making them suitable for application in real-world production environments.

The contribution of this research lies in the formulation of a practical and adaptable optimization model that integrates mathematical modeling with scheduling strategies, thereby facilitating more informed and data-driven decision-making. Furthermore, the flexibility of the model allows it to be extended or integrated with additional constraints and objectives depending on the specific needs of different industries.

Future research directions include the application of machine learning techniques for predicting optimal parameters of metaheuristic algorithms, as well as forecasting objective function values prior to executing the optimization process. In addition, advanced planning is expected to increasingly rely on artificial intelligence tools to develop models capable of real-time prediction and adaptation of key scheduling parameters. One such promising direction is reinforcement learning, which has the potential to dynamically guide decision-making in complex and uncertain production environments. Therefore, the integration of the presented model with AI-based predictive and adaptive systems represents both a natural extension of this research and a step toward intelligent, future-ready manufacturing systems.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] A. Stanković, G. Petrović, R. Turudija, D. Marković, and Ž. Ćojbašić, "Hybrid machine learning approach for parallel machine scheduling under uncertainty," *Expert Syst. Appl.*, vol. 279, p. 127427, 2025. https://doi.org/10.1016/j.eswa.2025.127427

[2] A. Stanković and G. Petrović, "Priority decision rules with a fuzzy MCDM approach for solving flexible job shop problem: A real case study of optimizing manufacturing," *Acta Polytech. Hung.*, vol. 22, no. 1, 2025. https://doi.org/10.12700/APH.22.1.2025.1.8

[3] Z. M. Rakićević, "Podrška operativnom planiranju proizvodnje i upravljanju resursa u malim i srednjim preduzecima," Ph.D. dissertation, Univerzitet u Beogradu, Fakultet Organizacionalih nauka, 2018.

[4] A. Yusuf and K. Saffu, "Planning and performance of small and medium enterprise operators in a country in transition," *J. Small Bus. Manage.*, vol. 43, no. 4, pp. 480–497, 2005. https://doi.org/10.1111/j.1540-627X.2005.00148.x

[5] A. Stanković, G. Petrović, D. Marković, and Ž. Ćojbašić, “Solving flexible job shop scheduling problem with transportation time based on neuro-fuzzy suggested metaheuristics,” *Acta Polytech. Hung.*, vol. 19, no. 4, pp. 209–227, 2022.

[6] X. Liu, F. Chu, F. Zheng, C. Chu, and M. Liu, “Parallel machine scheduling with stochastic release times and processing times,” *Int. J. Prod. Res.*, vol. 59, no. 20, pp. 6327–6346, 2021. https://doi.org/10.1080/00207543.2020.1812752

[7] A. Stanković, G. Petrović, Ž. Ćojbašić, and D. Marković, “An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem,” *Oper. Res. Eng. Sci.*, vol. 3, no. 3, pp. 13–28, 2020. https://doi.org/10.31181/oresta20303013s

[8] N. Simić, M. Stefanović, M. Milenkov, A. Stanković, and D. L. Vukić, “Model for resource allocation in the public sector based on strategic importance and quality,” *Teh. Vjesn.*, vol. 32, no. 2, pp. 628–636, 2025. https://doi.org/10.17559/TV-20240805001902

[9] F. Harary and I. C. Ross, “A procedure for clique detection using the group matrix,” *Sociometry*, vol. 20, no. 3, pp. 205–215, 1957. https://doi.org/10.2307/2785673

[10] C. S. Thomalla, “Optimization of a steam boiler job shop scheduling with alternative process plans,” *Int. J. Prod. Econ.*, vol. 74, no. 1-3, pp. 125–134, 2001. https://doi.org/10.1016/S0925-5273(01)00119-0

[11] C. A. Angell, “Formation of glasses from liquids and biopolymers,” *Science*, vol. 267, no. 5206, pp. 1924–1935, 1995. https://doi.org/10.1126/science.267.5206.1924

[12] L. Liu and C. Zhang, “A branch and bound algorithm for the robust parallel machine scheduling problem with sequence-dependent setup time,” *arXiv preprint*, 2017. https://doi.org/10.48550/arXiv.1712.00756

[13] W. Xing and J. Zhang, “Parallel machine scheduling with splitting jobs,” *Discrete Appl. Math.*, vol. 103, no. 1-3, pp. 259–269, 2000. https://doi.org/10.1016/S0166-218X(00)00176-1

[14] R. Qing-dao-er ji and Y. Wang, “A new hybrid genetic algorithm for job shop scheduling problem,” *Comput. Oper. Res.*, vol. 39, no. 10, pp. 2291–2299, 2012. https://doi.org/10.1016/j.cor.2011.12.005

[15] S. Wang and M. Liu, “Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning,” *J. Manuf. Syst.*, vol. 37, no. 1, pp. 182–192, 2015. https://doi.org/10.1016/j.jmsy.2015.07.002

[16] Y. H. Kim and R. S. Kim, “Insertion of new idle time for unrelated parallel machine scheduling with job splitting and machine breakdowns,” *Comput. Ind. Eng.*, vol. 147, p. 106630, 2020. https://doi.org/10.1016/j.cie.2020.106630

[17] C. A. Glass, C. N. Potts, and P. Shade, “Unrelated parallel machine scheduling using local search,” *Math. Comput. Model.*, vol. 20, no. 2, pp. 41–52, 1994. https://doi.org/10.1016/0895-7177(94)90205-4

[18] E. Vallada and R. Ruiz, “A genetic algorithm for the unrelated parallel machine scheduling problem with sequence-dependent setup times,” *Eur. J. Oper. Res.*, vol. 211, no. 3, pp. 612–622, 2011.

[19] K. C. Ying, Z. J. Lee, and S. W. Lin, “Makespan minimization for scheduling unrelated parallel machines with setup times,” *J. Intell. Manuf.*, vol. 23, pp. 1795–1803, 2012. https://doi.org/10.1007/s10845-010-0483-3

[20] K. Hasani, S. A. Kravchenko, and F. Werner, “Simulated annealing and genetic algorithms for the two-machine scheduling problem with a single server,” *Comput. Oper. Res.*, vol. 39, no. 10, pp. 3778–3792, 2014.

[21] C. Miao and J. Zou, “Parallel-machine scheduling with time-dependent and machine availability constraints,” *Math. Probl. Eng.*, vol. 2015, p. 956158, 2015. https://doi.org/10.1155/2015/956158

[22] M. N. Haddad, L. P. Cota, M. J. Souza, and N. Maculan, “Solving the unrelated parallel machine scheduling problem with setup times by efficient algorithms based on iterated local search,” in *16th International Conference, ICEIS 2014*, 2014. https://doi.org/10.1007/978-3-319-22348-3_8

[23] T. C. E. Cheng, S. A. Kravchenko, and B. M. Lin, “Preemptive parallel-machine scheduling with a common server to minimize makespan,” *Nav. Res. Logist.*, vol. 64, no. 5, pp. 388–398, 2017. https://doi.org/10.1002/nav.21762

[24] E. B. Edis, C. Oğuz, and I. Ozkarahan, “Parallel machine scheduling with additional resources: Notation, classification, models, and solution methods,” *Eur. J. Oper. Res.*, vol. 230, no. 3, pp. 449–463, 2013. https://doi.org/10.1016/j.ejor.2013.02.042

[25] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003. https://doi.org/10.1145/937503.937505

[26] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization,” *Nat. Comput.*, vol. 8, no. 2, pp. 239–287, 2009. https://doi.org/10.1007/s11047-008-9098-4

[27] M. Madić, “Mathematical modeling and optimization of laser cutting process using artificial intelligence methods,” Ph.D. dissertation, Faculty of Mechanical Engineering in Niš, University of Niš, 2013.

[28] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no.

4598, pp. 671–680, 1983. https://doi.org/10.1126/science.220.4598.671
[29] A. Stanković, D. Marković, G. Petrović, and Ž. Ćojbašić, "Simulated annealing and particle swarm optimization for the vehicle routing problem and communal waste collection in urban areas," in *14th International Conference on Accomplishments in Mechanical and Industrial Engineering, DEMI 2019*, 2019, pp. 497–505.
[30] A. Stanković, G. Petrović, D. Marković, and S. Marković, "Solving the unrelated parallel machine scheduling problem with setup times," in *Proceedings of the Eighth International Conference Transport and Logistics*, 2021.