Gazi Üniversitesi
Fen Bilimleri Dergisi
PART C: TASARIM VE TEKNOLOJİ

Gazi University
Journal of Science
PART C: DESIGN AND TECHNOLOGY

GU J Sci, Part C, 13(3): 919-931 (2025)

Capacity Balancing and Variation Minimization in Unrelated Parallel Machine Scheduling with Sequence-Dependent Setup Times: A Mathematical Model and Matheuristic Approach

Merhad AY[1*]

[1]*Düzce University, Faculty of Engineering, Department of Industrial Engineering, Duce, Türkiye*

Article Info

Research article
Received: 16/12/2024
Revision: 13/05/2025
Accepted: 20/06/2025

Keywords

Parallel machine scheduling
Matheuristic
Optimization
Capacity balancing

Makale Bilgisi

Araştırma makalesi
Başvuru: 16/12/2024
Düzeltme: 13/05/2025
Kabul: 20/06/2025

Anahtar Kelimeler

Paralel makine çizelgeleme
Matsezgisel
Optimizasyon
Kapasite dengeleme

Graphical/Tabular Abstract (Grafik Özet)

A matheuristic method integrating MILP and genetic algorithm is proposed for scheduling unrelated parallel machines with setup times. The matheuristic algorithm finds near-optimal solutions for large-scale problems within 904–1378 seconds, while the exact model fails to reach optimality within 18,000 seconds. / Hazırlık süreli özdeş olmayan paralel makinelerin çizelgelenmesi için MILP ve genetik algoritmayı entegre eden bir matsezgisel yöntem önerilmektedir. Matsezgisel algoritma, büyük ölçekli problemler için 904–1378 saniye içinde optimal çözüme yakın sonuçlar üretirken, matematiksel model 18.000 saniye içinde optimalliğe ulaşamamaktadır.

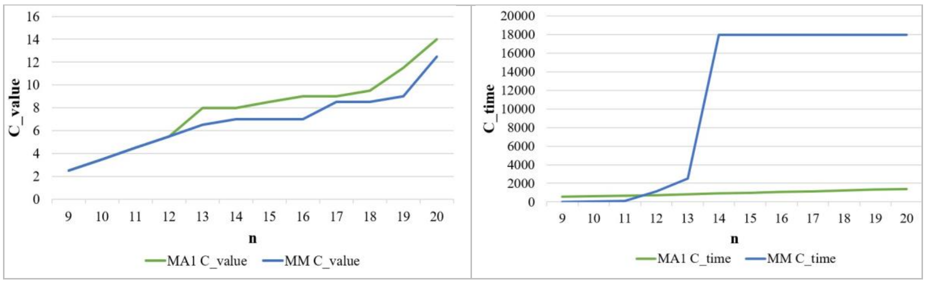

***Figure A**: Comparison of the mathematical model and matheuristic algorithm (solution quality and time for different problem sizes). / **Şekil A:** Matematiksel model ile matsezgisel algoritmanın karşılaştırılması (farklı problem boyutları için çözüm kalitesi ve süresi).*

Highlights (Önemli noktalar)

- *A mathematical model is proposed for capacity balancing and variation minimization. / Kapasite dengeleme ve varyasyon minimizasyonu için bir matematiksel model önerilmektedir.*
- *Two customized genetic algorithm-based mathematical heuristic algorithms are presented. / Genetik algoritma tabanlı iki özelleştirilmiş matsezgisel algoritma sunulmaktadır.*
- *The constrained mutation operator enhances solution feasibility. / Sınırlandırılmış mutasyon operatörü çözüm geçerliliğini artırmaktadır.*
- *Near-optimal solutions are obtained for large-scale problems in short time. / Büyük ölçekli problemlerde kısa sürede optimal çözüme yakın sonuçlar elde edilmektedir.*

***Aim (Amaç):** This study aims to minimize capacity imbalance and variation in job characteristics in unrelated parallel machine scheduling with sequence-dependent setup times. / Bu çalışma, sıra bağımlı hazırlık süreli özdeş olmayan paralel makine çizelgelemesinde kapasite dengesizliğini ve iş özelliklerindeki varyasyonu en aza indirmeyi amaçlamaktadır.*

***Originality (Özgünlük):** The study integrates a MILP model with a problem-specific genetic algorithm-based matheuristic, introducing constrained mutation and customized chromosome structure. / Çalışma, probleme özgü genetik algoritma tabanlı bir matsezgisel yöntem ile MILP modelini entegre ederek sınırlandırılmış mutasyon ve özelleştirilmiş kromozom yapısı sunmaktadır.*

***Results (Bulgular):** While the mathematical model yields optimal results for small problems, matheuristic algorithms provide near-optimal solutions for large problems in significantly shorter time. / Matematiksel model küçük problemler için optimal çözümler üretirken, matsezgisel algoritmalar büyük problemler için çok daha kısa sürede optimal çözüme yakın sonuçlar sunmaktadır.*

***Conclusion (Sonuç):** The proposed hybrid approach shows strong potential in real-world production planning by exhibiting satisfactory performance in terms of accuracy and speed. / Önerilen hibrit yaklaşım, doğruluk ve hız konularında gayet yeterli performans sergileyerek gerçek dünya üretim planlamasında güçlü bir potansiyel göstermektedir.*

*Corresponding author, e-mail: merhaday@duzce.edu.tr DOI: 10.29109/gujsc.1598698

GU J Sci, Part C, 13(3): 919-931 (2025)

Gazi Üniversitesi
Fen Bilimleri Dergisi
PART C: TASARIM VE TEKNOLOJİ

Gazi University
Journal of Science
PART C: DESIGN AND TECHNOLOGY

http://dergipark.gov.tr/gujsc

Capacity Balancing and Variation Minimization in Unrelated Parallel Machine Scheduling with Sequence-Dependent Setup Times: A Mathematical Model and Matheuristic Approach

Merhad AY[1*]

[1]*Düzce University, Faculty of Engineering, Department of Industrial Engineering, Düzce, Türkiye*

Article Info

Research article
Received: 16/12/2024
Revision: 13/05/2025
Accepted: 20/06/2025

Keywords

Parallel machine scheduling
Matheuristic
Optimization
Capacity balancing

Abstract

Scheduling of unrelated parallel machines with sequence-dependent setup times presents significant theoretical and practical challenges due to its combinatorial complexity and frequent occurrence in various production environments. This study addresses a scheduling problem specific to tire manufacturing, focusing on capacity balancing and variation minimization in unrelated parallel machines with sequence-dependent setup times. For small and medium-sized problem sets consisting of 9 to 13 jobs, optimal solutions were obtained using a mathematical model. However, when the number of jobs increased to 14 or more, the solution time exceeded 18000 seconds, and optimality could not be achieved. Therefore, two genetic algorithm-based matheuristic algorithms ($MA1$ and $MA2$) with problem-specific customized chromosome structures are proposed for large-scale problem sets. Additionally, the classical random mutation operator is modified into a constrained random mutation operator tailored to the problem. Experimental results and statistical analyses (p < 0.05) show that the $MA1$ algorithm performs better than $MA2$ in terms of solution quality and find solutions similar to the best feasible solutions produced by the mathematical model within a significantly shorter time frame, averaging between 904 and 1378 seconds for large-scale problems. The study offers notable advantages in terms of both solution time and quality in solving real-world problems. The proposed matheuristic algorithm contributes to the literature through its problem-specific chromosome design, initial population generation method, and constrained random mutation operator.

Sıra Bağımlı Hazırlık Süreli Özdeş Olmayan Paralel Makine Çizelgelemede Kapasite Dengeleme ve Varyasyon Minimizasyonu: Matematiksel Model ve Matsezgisel Yaklaşım

Makale Bilgisi

Araştırma makalesi
Başvuru: 16/12/2024
Düzeltme: 13/05/2025
Kabul: 20/06/2025

Anahtar Kelimeler

Paralel makine çizelgeleme
Matsezgisel
Optimizasyon
Kapasite dengeleme

Öz

Sıra bağımlı hazırlık sürelerine sahip özdeş olmayan paralel makinelerin çizelgelenmesi, kombinatoryal karmaşıklığı ve pek çok üretim sürecinde yaygın olarak karşılaşılması nedeniyle hem teorik hem de pratik düzeyde önemli zorluklar barındırmaktadır. Bu çalışmada, sıra bağımlı hazırlık sürelerine sahip özdeş olmayan paralel makineler için lastik imalatında kapasite dengeleme ve varyasyon minimizasyonu özelinde bir çizelgeleme problemi ele alınmaktadır. Probleme yönelik 9 ila 13 işten oluşan küçük ve orta ölçekli problem setlerinde matematiksel model ile optimal çözümler üretilmiş; ancak, iş sayısı 14 ve üzerine çıktığında çözüm süresi 18000 saniyeyi aşmış ve optimum çözümlere ulaşılamamıştır. Bu nedenle, büyük problem setleri için probleme özgü özelleştirilmiş kromozom yapısıyla genetik algoritma tabanlı iki farklı matsezgisel algoritma ($MA1$ ve $MA2$) önerilmektedir. Ayrıca, mutasyon aşamasında klasik rastgele mutasyon operatörü, probleme özgü olarak modifiye edilerek sınırlandırılmış rastgele mutasyon operatörü olarak kullanılmaktadır. Deneysel sonuçlar ve istatistiksel analizler (p<0.05), $MA1$ algoritmasının çözüm kalitesi açısından $MA2$'ye göre daha başarılı olduğunu ve büyük problem setlerinde matematiksel modelin 18000 saniyede bulabildiği en iyi uygun çözümlere benzer çözümleri ortalama 904–1378 saniye gibi oldukça kısa sürelerde bulabildiğini göstermektedir. Çalışma, çözüm süresi ve kalitesi açısından gerçek hayat problemlerinin çözümünde önemli avantajlar sunmaktadır. Önerilen matsezgisel algoritma, probleme özgü olarak tasarlanan kromozom yapısı, başlangıç popülasyonu oluşturma yöntemi ve sınırlandırılmış rastgele mutasyon operatörü ile benzer problemlerin çözümüne yönelik literatüre önemli katkılar sunmaktadır.

*Corresponding author, e-mail: merhaday@duzce.edu.tr
DOI: 10.29109/gujsc.1598698

1. INTRODUCTION (GİRİŞ)

In today's highly dynamic and competitive manufacturing environments, efficient production scheduling is no longer a choice, it is a necessity for survival. Factories producing products in a wide variety of sizes and features, such as tires with varying dimensions and compound structures, must deal with an overwhelming number of constraints, including machine compatibility, sequence-dependent setup times, and resource balancing. Failure to manage these complexities can result in serious inefficiencies, production delays, and rising costs.

Parallel machine scheduling problems, particularly those involving unrelated machines and sequence-dependent setup times, are among the most challenging classes of combinatorial optimization problems. Existing studies in the literature have generally focused on minimizing time-based performance metrics like makespan or tardiness [1-4], leaving a research gap in addressing capacity balancing and variation reduction in job characteristics. For instance, balancing workload across machine groups and minimizing product variation within each machine are critical to ensuring consistent quality and efficient operation. These multifaceted requirements necessitate novel approaches that blend exact optimization with flexible heuristics. To address this, a mixed-integer linear programming (MILP) model is proposed to obtain exact solutions for small and medium sized problems (9-13 jobs), while two genetic algorithm-based matheuristic algorithms are developed to efficiently solve large-scale problems (14-20 jobs) where the mathematical model becomes computationally infeasible.

The proposed methods have been evaluated through real-world case studies, and the results have been analyzed comparatively using statistical methods. This study not only makes a significant contribution to the existing solution approaches for parallel machine scheduling problems but also demonstrates the potential to guide time-critical real-world applications by providing fast and efficient solutions. In this context, the study is expected to make substantial contributions to the literature. The remainder of this paper is structured as follows: Section 2 reviews the relevant literature on parallel machine scheduling problems, emphasizing both mathematical models and matheuristic approaches. Section 3 provides a detailed description of the problem definition and introduces the proposed methods, including the mathematical model and the genetic algorithm-based matheuristic approach. Section 4 presents the experimental results, along with a comparative analysis of the proposed methods using statistical techniques. Finally, Section 5 concludes the paper with key findings and suggestions for future research directions.

2. LITERATURE (LİTERATÜR)

Parallel machine scheduling problems, due to their NP-hard nature, require complex mathematical models for their solutions. These models are often based on mixed-integer linear programming (MILP) or nonlinear programming approaches. Tavakkoli-Moghaddam et al. [5] developed a two-level mathematical model that considers sequence-dependent setup times in unrelated parallel machines, aiming to optimize makespan and tardiness. Akyol and Sarac [6] proposed a mixed integer programming model for the problem of scheduling jobs using shared resources on parallel machines. Safaei et al. [7] designed a multi-objective optimization model for parallel machines, simultaneously optimizing tardiness and completion time. Yepes-Borrero et al. [8] applied mathematical models to multi-objective scheduling problems by incorporating resource constraints. Mathematical models are particularly preferred for small and medium scale problems due to their theoretical accuracy and potential to provide optimal solutions. However, the long solution times of mathematical models in large-scale problems have led to the development of heuristic, metaheuristic or matheuristic algorithms. These methods, while not guaranteeing optimality compared to mathematical models, significantly reduce solution times. Ji et al. [9] applied an adaptive large neighborhood search algorithm to parallel machine scheduling problems, providing solutions for large datasets. Ezugwu [10] achieved efficient results for complex scheduling problems using a firefly algorithm. Haddad et al. [11] combined genetic algorithms with variable neighborhood search methods to effectively explore large solution spaces, offering solutions for large-scale problems.

Genetic algorithms are widely used metaheuristic methods in parallel machine scheduling problems, generating solutions inspired by the fundamental principles of biological evolution. Vallada and Ruiz [12] demonstrated the effectiveness of genetic algorithms in unrelated parallel machines with sequence-dependent setup times. Ozcelik and Sarac [13] used genetic algorithm to minimize the makespan by taking into account the unavailable time periods in parallel machine scheduling problems. Zeidi and Hosseini [14] combined genetic algorithms with simulated annealing to develop an approach aimed at minimizing total tardiness costs. Kim and Kim [15] achieved solutions by integrating genetic algorithms with sequence-dependent setup times. Antunes et al. [16] compared the performance of genetic algorithms with other metaheuristic methods for unrelated parallel machines and concluded that genetic algorithms outperform others, especially in large-scale problems. Due to their ability to explore vast solution spaces and their rapid performance in large-scale problems, genetic algorithms are extensively employed in the literature.

Matheuristic algorithms, which combine mathematical models and genetic algorithms, have the potential to balance solution quality and computational efficiency. These hybrid approaches integrate the precision advantage of mathematical models with the flexibility and speed of genetic algorithms. For instance, Chang et al. [17] applied hybrid approaches in complex manufacturing processes such as surface-mount technology to derive diverse solutions. The approach proposed in this study offers a novel framework that integrates mathematical model with genetic algorithms. The proposed matheuristic algorithms featuring problem-specific chromosome structure, a constrained mutation operator, and alternative crossover strategies are designed to deliver high quality solutions for large-scale problems within practical time limits. This integrated methodology presents both theoretical novelty and practical applicability for complex scheduling problems.

3. PROBLEM DEFINITON and METHODOLOGY (PROBLEM TANIMI VE METODOLOJİ)

The study presents a solution approach inspired by the machine scheduling problem encountered in a tire manufacturing facility. The facility operates with two distinct machine groups: one group consisting of four machines ($V1$, *V2*, *V3*, *V4*) and another group with two machines (*T1*, *T2*). The tire production process is characterized by various constraints and parameters, which contribute to the complexity of the scheduling problem.

The processing time for each job varies across machines, resulting in $n \times m$ distinct processing time combinations in a system with n jobs and m machines. These processing times are represented by the P matrix in this study. Additionally, every machine is not capable of performing every job. To address this, a compatibility matrix (Y) is defined, where binary values (0-1) indicate machine-job compatibility: $Y_{ij} = 1$, denotes that job i can be processed on machine j, while Y_{ij} =0 indicates that it cannot be processed.

The characteristics of the jobs are defined by parameters such as aspect ratio (TG), rim size (JG), and mixture group (KG). These parameters represent the unique requirements of each job and vary across different jobs. Independently of the machines, jobs have sequence-dependent setup times, which are represented by the S matrix. For example, in a system with n jobs, there are $n \times n$ distinct setup time values. Additionally, the production quantity for each job is defined as an integer in the U matrix, where all values are strictly greater than zero. The problem also includes the following constraints:

- Jobs cannot be split.
- Each job must be processed on a single machine.
- A machine cannot process more than one job simultaneously.
- Jobs must be executed consecutively without skipping any in the sequence, ensuring no gaps in the sequence.
- Each machine must operate within its daily working time limit and cannot exceed this constraint.

The objective function of the problem aims to minimize the difference in capacity utilization rates between machine groups V and T. Additionally, it seeks to minimize the variations in TG, JG, and KG values among the jobs assigned to each machine. This approach strives to achieve balanced capacity utilization among machine groups and reduce variability in the assigned jobs. The objective function incorporates four distinct goals, each assigned weighted coefficients based on their importance. This weighting ensures that higher priority objectives are given more significance in the solution process and allows decision variables with different units to be effectively incorporated into the same objective function

3.1. Mathematical Model (Matematiksel Model)

To address the problem, a mixed-integer linear programming (MILP) model has been proposed. In the literature, various mathematical models for unrelated parallel machines with sequence-dependent setup times focus on different objective functions and constraints, such as minimizing total tardiness [3, 14], optimizing constrained resource usage [5], employing compatibility matrices [18], and minimizing makespan [19, 20]. Unlike previous studies, the proposed mathematical model (MM)

introduces a multi-objective structure that emphasizes minimizing the difference in capacity utilization rates among machine groups and reducing parameter variability (TG, JG, and KG) within each machine. Furthermore, by simultaneously incorporating job characteristics, capacity utilization constraints, and compatibility matrix constraints, the proposed model distinguishes itself from existing approaches in the literature.

Indices

i, b: jobs

j : machines

k: sequences

Parameters

f: Maximum job sequence

v: Number of machines in group V

t: Number of machines in group T

OT: Maximum operating time of machines

M: Big number

S_{ib}: Setup time of job b done after job i

P_{ij}: Processing time of job i on machine j

U_i: Amount of production job i

Y_{ij}: Matrix indicating whether job i is compatible machine j

TG_i: Aspect ratio of job i

JG_i: Rim size of job i

KG_i: Mixture type of job i

W_1: Weight of the importance level of difference in capacity utilization rates

W_2: Weight of the importance level of differences in aspect ratios

W_3: Weight of the importance level of differences in rim sizes

W_4: Weight of the importance level of differences in mixture types

Decision variables

X_{ijk}: $\begin{cases} 1, & \text{if job } i \text{ assigned to machine } j \text{ in sequence } k \\ 0, & \text{otherwise} \end{cases}$

Z_{ibjk}:
$\begin{cases} 1, & \text{if job } b \text{ is assigned to machine } j \text{ in sequence } k \text{ following job } i \\ 0, & \text{otherwise} \end{cases}$

C_{jk}: completion time of the job in sequence k on machine j

D_j: Makespan of machine j

ZZ_j: The capacity utilization rate of machine j

V: The capacity utilization rate of machine group V

T: The capacity utilization rate of machine group T

TG_max_j: Maximum aspect ratio on machine j

TG_min_j: Minimum aspect ratio on machine j

TG_dif_j: Difference between TG_max_j and TG_min_j on machine j

JG_max_j: Maximum rim size on machine j

JG_min_j: Minimum rim size on machine j

JG_dif_j: Difference between JG_max_j and JG_min_j on machine j

KG_max_j: Maximum mixture type value on machine j

KG_min_j: Minimum mixture type value on machine j

KG_dif_j: Difference between KG_max_j and KG_min_j on machine j

Formulation

$$minimize \quad W_1 * |V - T| + W_2 * \sum_j TG\_dif_j + W_3 * \sum_j JG\_dif_j + W_4 * \sum_j KG\_dif_j \tag{1}$$

subject to:

$$D_j \geq C_{jk} \qquad \forall\, j, k \tag{2}$$

$$C_{j0} \geq X_{ij0} * P_{ij} * U_i \qquad \forall\, i, j \tag{3}$$

$$Z_{ibjk} \leq (X_{ijk} + X_{bjk+1})/2 \qquad \forall\, i, b, j, k : k \neq f \tag{4}$$

$$Z_{ibjk} \geq (X_{ijk} + X_{bjk+1}) - 1 \qquad \forall\, i, b, j, k : k \neq f \tag{5}$$

$$C_{jk+1} \geq C_{jk} + P_{bj} * U_b * Z_{ibjk} + S_{ib} * Z_{ibjk} \qquad \forall\, i, b, j, k : k \neq f \tag{6}$$

$$\sum_j \sum_k X_{ijk} = 1 \qquad \forall i \tag{7}$$

$$\sum_k X_{ijk} \leq Y_{ij} \qquad \forall\, i, j \tag{8}$$

$$\sum_i X_{ijk} \leq 1 \qquad \forall\, j, k \tag{9}$$

$$\sum_b X_{bjk+1} \leq \sum_i X_{ijk} \qquad \forall\, j, k : k \neq f \tag{10}$$

$$D_j \leq \text{OT} \qquad \forall\, j \tag{11}$$

$$ZZ_j = D_j / OT \qquad \forall\, j \tag{12}$$

$$V = \sum_j ZZ_j / v \qquad \forall\, j : j = \{1, 2, \dots, v\} \tag{13}$$

$$T = \sum_j ZZ_j / t \qquad \forall\, j : j = \{1, 2, \dots, t\} \tag{14}$$

$$TG\_max_j \geq TG_i * X_{ijk} \qquad \forall\, i, j, k \tag{15}$$

$$TG\_min_j \leq TG_i * X_{ijk} + M * (1 - X_{ijk}) \qquad \forall\, i, j, k \tag{16}$$

$$TG\_min_j \geq 1 \qquad \forall\, j \tag{17}$$

$$TG\_dif_j = TG\_max_j - TG\_min_j \qquad \forall\, j \tag{18}$$

$$JG\_max_j \geq JG_i * X_{ijk} \qquad \forall\, i, j, k \tag{19}$$

$$JG\_min_j \leq JG_i * X_{ijk} + M * (1 - X_{ijk}) \qquad \forall\, i, j, k \tag{20}$$

$$JG\_min_j \geq 1 \qquad \forall\, j \tag{21}$$

$$JG\_dif_j = JG\_max_j - JG\_min_j \qquad \forall\, j \tag{22}$$

$$KG\_max_j \geq KG_i * X_{ijk} \qquad \forall\, i, j, k \tag{23}$$

$$KG\_min_j \leq KG_i * X_{ijk} + M * (1 - X_{ijk}) \qquad \forall\, i, j, k \tag{24}$$

$$KG\_min_j \geq 1 \qquad \forall\, j \tag{25}$$

$$KG\_dif_j = KG\_max_j - KG\_min_j \quad \forall j \tag{26}$$

$$X_{ijk} \in \{0,1\} \quad \forall i,j,k \tag{27}$$

$$C_{jk} \geq 0 \quad \forall j,k \tag{28}$$

$$Z_{ibjk} \in \{0,1\} \quad \forall i,b,j,k \tag{29}$$

$$D_j, ZZ_j, V, T \geq 0 \quad \forall j \tag{30}$$

$$TG\_max_j, TG\_min_j, TG\_dif_j \geq 0 \text{ and integer} \quad \forall j \tag{31}$$

$$JG\_max_j, JG\_min_j, JG\_dif_j \geq 0 \text{ and integer} \quad \forall j \tag{32}$$

$$KG\_max_j, KG\_min_j, KG\_dif_j \geq 0 \text{ and integer} \quad \forall j \tag{33}$$

The objective function of the model is defined by Equation 1, aiming to minimize the absolute difference in capacity utilization between machine groups $|V - T|$ and the variability in aspect ratio (TG), rim size (JG), and mixture group (KG) on each machine using weighted importance coefficients. Equation 2 determines the completion time of the last job on each machine, while Equation 3 defines the completion time of the first job. To arrange job sequences, Equations 4 and 5 use the Z variable to identify whether a specific job follows another, assigning a value of 1 or 0 accordingly. Equation 6 calculates the completion time of a job by considering the sequence-dependent setup time from the previous job. Equation 7 ensures that each job is assigned to only one machine and in a specific order, while Equation 8 enforces that jobs are assigned only to machines capable of performing them, based on the compatibility matrix. To regulate sequencing, Equation 9 restricts each position on a machine to one job, and Equation 10 prohibits gaps in the sequence of jobs on a machine. Equation 11 imposes a constraint to prevent machines from exceeding their daily operating time. Capacity utilization rates are computed for each machine using Equation 12, and for the V and T machine groups using Equations 13 and 14, respectively. The maximum and minimum values of TG, JG, and KG parameters, along with their differences, are defined and minimized through Equations 15-26. Finally, Equations 27-33 specify sign constraints, establishing the permissible value ranges and types of the decision variables.

The developed mathematical model provides effective solutions for small-scale problems; however, its practical applicability is limited for large-scale problems due to the exponential increase in solution times. This limitation arises because the problem belongs to the NP-hard class, where solution times grow exponentially with problem size. To address this, heuristic, metaheuristic, or matheuristic algorithms are commonly employed to achieve near-optimal solutions within shorter timeframes. In this context, a genetic algorithm-based matheuristic approach has been developed to meet critical time-bound requirements, such as the efficient scheduling of daily production plans.

3.2. Matheuristic Algorithm (Matsezgisel Algoritma)

The matheuristic algorithm is designed to provide faster and more efficient solutions by integrating heuristic algorithms with mathematical models [21]. The proposed algorithm in this study is a genetic algorithm-based matheuristic approach. Its fitness function partially utilizes the modified version of the proposed mathematical model (MM). The matheuristic algorithm generates new chromosomes by maintaining the solution representation and applying feasible initial population, crossover, and mutation operations. In this study, two different crossover operators are employed, resulting in two distinct matheuristic algorithms ($MA1$ and $MA2$). The sole difference between $MA1$ and $MA2$ is the crossover operator used; all other steps remain identical. These hybrid methods combine the speed and flexibility of genetic algorithms with the mathematical precision of linear models, offering a robust approach to solving complex problems such as unrelated parallel machines with sequence-dependent setup times. The algorithm's flowchart is illustrated in Figure 1.

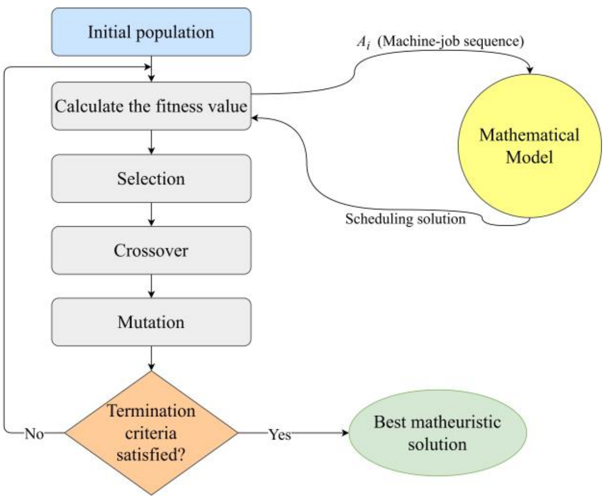

Figure 1. Flowchart of the genetic algorithm-based matheuristic approach (Genetik algoritma tabanlı matsezgisel yaklaşımın akış şeması)

In the matheuristic algorithm, an initial population [22] is generated by creating a problem-specific number of chromosomes, corresponding to the predefined population size. An example chromosome is illustrated in Figure 2. The chromosome structure developed specifically for the problem, represents a one-dimensional machine-job assignment sequence A_i, which is sent to the mathematical model. In the chromosome, genes sequentially represent jobs, while the values of the genes denote the machines to which these jobs are assigned. This customized representation replaces the two-dimensional compatibility matrix Y_{ij} used in the mathematical model (MM) with a one-dimensional compatibility sequence A_i. When generating chromosomes for the initial population, gene values (machine assignments) are randomly assigned based on the compatibility matrix Y_{ij}. For instance, if the compatible machines for the first job are given by Y_{1j}= [0, 1, 1, 0, 1], the corresponding set of suitable machines is {2, 3, 5}. Machine numbers are assigned to genes by randomly selecting values from this set.

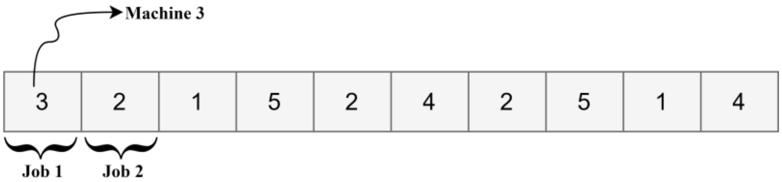

Figure 2. Chromosome structure (solution representation) (Kromozom yapısı (çözüm gösterimi))

To adapt the mathematical model to the chromosome structure used in the matheuristic algorithm, the parameter Y_{ij} is transformed into A_i, and one of the model's constraints, Equation 8, is replaced with the Equation 34. This new equation ensures that the machine-job assignments in the chromosomes generated by the matheuristic algorithm remain consistent with the compatibility constraints. The newly introduced parameter A_i in the mathematical model helps reduce the solution space, enabling the algorithm to achieve near-optimal solutions in a shorter time frame.

$$\sum_j \sum_k j * X_{ijk} = A_i \qquad \forall i \qquad (34)$$

After generating the initial population and calculating fitness values in the matheuristic algorithm, the process continues with selection, crossover, and mutation stages. The selection phase determines which chromosomes will proceed to the next generation. In this study, the roulette wheel selection method is employed. This method calculates the selection probabilities of chromosomes based on their fitness values [23]. Among the candidate chromosomes, the one with the highest fitness value has the greatest probability of being selected. While chromosomes with higher fitness values have a greater probability of being selected, chromosomes with lower fitness values also retain a chance of selection, ensuring genetic diversity within the population. This approach allows the evolutionary process to avoid local optima and enables a broader exploration of the solution space. Figure 3 illustrates an example of the selection probabilities in a population of three chromosomes using the roulette wheel method. Fitness value reflects the quality of a chromosome's solution to the problem, while probability represents its relative advantage within the population. For instance, Chromosome 3, having the highest fitness value, has a 60% probability of being selected, whereas Chromosome 1, with the lowest fitness value, has a 10% probability. This structure promotes the selection of high-quality chromosomes while preserving diversity within the population, supporting the principles of robust evolutionary search.

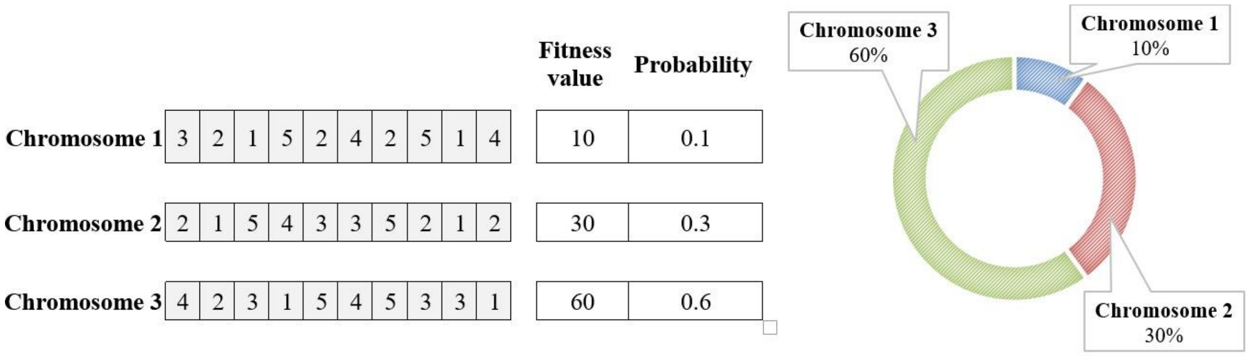

Figure 3. Roulette wheel selection method (Rulet tekerleği seçim yöntemi)

The crossover phase is a critical process where two new children are generated through the exchange of genes between two selected chromosomes. In this study, two different crossover operators, specifically designed to align with the solution representation, are utilized. The first operator is implemented in the $MA1$ algorithm, and second operator is applied in the $MA2$ algorithm to evaluate performance using an alternative crossover strategy. The effectiveness of both operators is analyzed through comparative evaluations presented in the experimental results section, where the performance differences between the algorithms are assessed using statistical methods.

In the $MA1$ algorithm, a uniform crossover operator [24] is employed. This operator performs crossover by exchanging genes between two chromosomes based on a predefined probability value. In this study, the probability value is set to 0.5. If a randomly generated number between 0 and 1 exceeds 0.5, the genes are exchanged. Figure 4 provides an example of the uniform crossover method. In the machine-job assignment representation used in this study, genes (jobs) are exchanged in the same order, ensuring no conflicts occur. However, if operators that alter the order of genes are used, there is a risk of assigning incompatible jobs to machines due to the crossover of genes in different positions. To mitigate such risks, carefully selected operators have been utilized in this study, ensuring the accuracy and validity of the solution process are preserved.

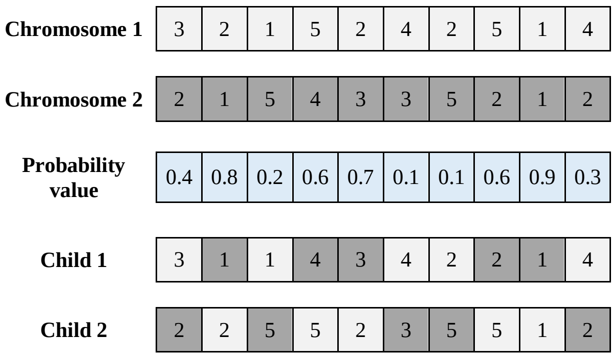

Figure 4. Uniform crossover operator (Tekdüze çaprazlama operatörü)

In the $MA2$ algorithm, a two-point crossover operator [25] is utilized. Figure 5 presents an example of the operator. In the method, two random points are selected on the chromosomes, and a segment is exchanged between the two chromosomes, resulting in the creation of two new child. The gene order is preserved in this operator, and no alterations are made to the sequence of genes. Consequently, there are no conflicts or inconsistencies in the fitness evaluation performed through the mathematical model. This ensures the integrity and validity of the solution process.

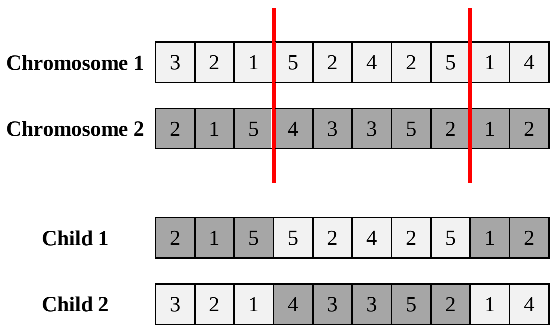

Figure 5. Two-point crossover operator (İki noktalı çaprazlama operatörü)

The next phase of the algorithm, mutation, aims to enhance genetic diversity within the population and prevent convergence to local optima by introducing changes to genes with a low probability. Various operators representing different gene alteration strategies, such as swap, shift, and inversion, are commonly used in this stage [26]. However, the chromosome structure employed in the matheuristic algorithm is not compatible with methods that alter gene order. Another category of mutation operators, including random, uniform, and creep mutations, assigns random or bounded values to selected genes instead of altering their order [27]. In this study, the random mutation operator has been modified to align with the chromosome structure. In its standard form, random mutation replaces genes with values randomly selected within predefined lower and upper bounds. However, this operator is not directly compatible with the chromosome structure used in the study, as the gene values cannot take any value within the specified range. To address the incompatibility, the mutation operator has been constrained and adapted to the compatibility matrix Y_{ij}, similar to the approach used for generating the initial population. With this modification, the values assigned to the mutated genes are randomly selected from the feasible values determined by the compatibility matrix, ensuring the validity and integrity of the chromosome structure throughout the mutation process.

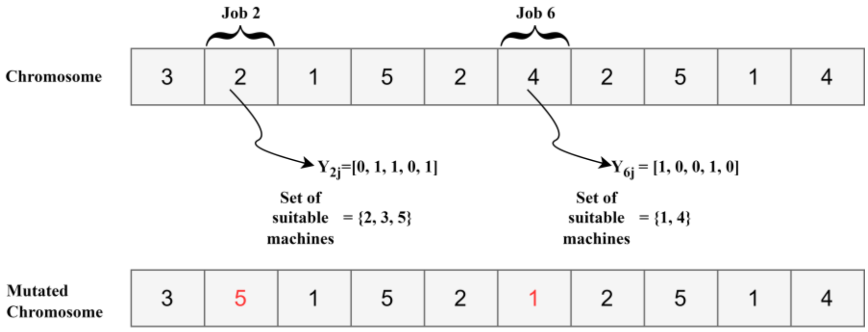

Figure 6. Constrained random mutation (Sınırlandırılmış rastgele mutasyon)

Figure 6 presents an example application of the constrained random mutation operator. For instance, in a chromosome with 10 genes, the 2nd and 6th genes are selected for mutation. To determine the possible values for these genes, compatibility arrays Y_{2j} and Y_{6j} first generated from the compatibility matrix. Arrays are then used to derive the sets of suitable machines for each selected gene. During the mutation process, a value is randomly assigned to each selected gene from its respective set of suitable machines. This method ensures that genes are replaced with random but constrained values, allowing the algorithm to produce valid and meaningful results within the solution space.

Finally, the execution of the matheuristic algorithm terminates based on termination criteria such as the target fitness value or the maximum number of iterations. These criteria ensure that the algorithm either achieves a predefined performance threshold or completes the specified maximum number of steps. Thus, the algorithm operates efficiently in terms of computational time while striving to achieve the desired solution quality.

4. EXPERIMENTAL RESULTS (DENEYSEL SONUÇLAR)

In the experimental studies, various problem sets (PS) were derived based on a real-world daily machine scheduling problem from a tire manufacturing company. While preparing the problem sets, several constraints, such as time, quantities, and intervals observed in the company, were taken into account. The parameters for the problem sets, which were created based on different job sizes (n), are summarized in Table 1. Each problem set also includes a setup time matrix (S), a sequence representing production quantities (U), a machine-job compatibility matrix (P), an array of aspect ratios (TG), an array of rim widths (JG), and a mixture group (KG) sequence.

Table 1. Parameters of the problem sets (Problem setlerinin parametreleri)

	n	v	t	TG_min, TG_max	JG_min, JG_max	KG_min, KG_max
PS_1	9	4	2	25, 55	15, 17	1, 2
PS_2	10	4	2	25, 55	15, 17	1, 2
PS_3	11	4	2	25, 55	15, 17	1, 2
PS_4	12	4	2	25, 55	15, 17	1, 2
PS_5	13	4	2	25, 55	15, 17	1, 2
PS_7	14	4	2	25, 55	15, 17	1, 2
PS_8	15	4	2	25, 55	15, 17	1, 2
PS_9	16	4	2	25, 55	15, 17	1, 2
PS_{10}	17	4	2	25, 55	15, 17	1, 2
PS_{11}	18	4	2	25, 55	15, 17	1, 2
PS_{12}	19	4	2	25, 55	15, 17	1, 2
PS_{13}	20	4	2	25, 55	15, 17	1, 2

The proposed matheuristic algorithms ($MA1$, $MA2$) and the mathematical model (MM) were implemented using the Python programming language and executed on a computer equipped with an Intel Core i5 2.40 GHz processor and 8 GB of RAM. Initially, the mathematical model was tested on 13 different problem sets. All tests were limited to a maximum runtime of 18000 seconds (5 hours) and repeated multiple times. The objective function values (C_{value}) and average solution times (C_{time}) obtained from these tests are presented in Table 2.

The test results indicate that the time required to obtain optimal solutions increases exponentially with the number of jobs. When the test duration reached the 18000 seconds, the process was automatically terminated. Consequently, optimal* solutions were achieved for problem sets with 9 to 13 jobs, while optimal* solutions could not be obtained for problem sets with 14 to 20 jobs. Furthermore, the fact that even longer durations than 18000 seconds are required for daily scheduling applications highlights the practical limitations of the mathematical model. This underscores the necessity of employing alternative methods for daily operational planning, where quick and effective solutions are critical.

Table 2. Test results for the mathematical model (Matematiksel model için test sonuçları)

	n	MM - C_{value}	MM - C_{time}
PS_1	9	2.5*	8.27
PS_2	10	3.5*	28.42
PS_3	11	4.5*	122.78
PS_4	12	5.5*	1113.71
PS_5	13	6.5*	2525.50
PS_7	14	7.0	18000
PS_8	15	7.0	18000
PS_9	16	7.0	18000
PS_{10}	17	8.5	18000
PS_{11}	18	8.5	18000
PS_{12}	19	9.0	18000
PS_{13}	20	12.5	18000

As an alternative to the mathematical model and to achieve rapid results in daily scheduling, the proposed matheuristic algorithm aims to provide near-optimal solutions within a short time frame. The matheuristic algorithm was configured with fixed parameters for all tests, which include a population size of 45, 30 iterations, a crossover rate of 0.9, and a mutation rate of 0.01. The runtime of the matheuristic algorithm was limited to 18000 seconds and algorithm was executed multiple times, as with the mathematical model. During the testing process, results were obtained using two different crossover operators (uniform and two-point). The performance of these operators was evaluated in terms of average solution value and runtime, and the detailed results are presented in Table 3. The test results demonstrate that the matheuristic algorithm offers significant advantages over the mathematical model, particularly in time-constrained applications such as daily scheduling, by giving fast and effective solutions.

Table 3. Test results for the matheuristic algorithms (Matsezgisel algoritmalar için test sonuçları)

	n	MA1 - C_{value}	MA2 - C_{value}	MA1 - C_{time}	MA2 - C_{time}
PS_1	9	2.5*	2.5*	571.17	578.91
PS_2	10	3.5*	4	633.00	583.96
PS_3	11	4.5*	5	668.84	663.47
PS_4	12	5.5*	6	734.65	731.02
PS_5	13	8	8	836.13	881.04
PS_7	14	8	9.5	907.54	915.18
PS_8	15	8.5	10.5	995.01	986.60
PS_9	16	9	11.5	1084.32	1092.16

PS_{10}	17	9	12	1142.56	1102.91
PS_{11}	18	9.5	12.5	1242.16	1257.00
PS_{12}	19	11.5	15	1353.82	1325.42
PS_{13}	20	14	15.5	1378.21	1399.33

$MA1$ represents the matheuristic algorithm utilizing the uniform crossover operator, while $MA2$ represents the matheuristic algorithm employing the two-point crossover operator. Both methods were statistically compared in terms of the obtained solution values (C_{value}) and solution times (C_{time}). The statistical test results, conducted using the Wilcoxon signed ranks test method at a significance level of $p < 0.05$, are presented in Table 4.

Table 4. Wilcoxon signed ranks test results (Wilcoxon işaretli sıralar testi sonuçları)

	Pairs	Z	p**-value**
Pair 1	MA1 - C_{value} & MA2 - C_{value}	-2.814	0.0049
Pair 2	MA1 - C_{time} & MA2 - C_{time}	-0.0784	0.9375
Pair 3	MA1 - C_{value} &MM - C_{value}	-2.536	0.0112
Pair 4	MA2 - C_{value} & MM - C_{value}	-2.941	0.0033

The statistical test results indicate that a p-value less than 0.05 signifies a significant difference in the solution values obtained by the $MA1$ and $MA2$ algorithms. This finding demonstrates that the $MA1$ algorithm outperforms $MA2$ in terms of solution quality. However, when comparing solution times, the p-value greater than 0.05 indicates no statistically significant difference between the two algorithms' runtimes. These results suggest that the $MA1$ algorithm achieves lower solution values within the same solution time as $MA2$. Consequently, the $MA1$ algorithm emerges as a more effective solution method.

When comparing the results of the $MA1$ algorithm with the results of the mathematical model (MM), it is evident that the $MA1$ algorithm can produce near-optimal solutions in significantly shorter solution time. The comparative graphs presented in Figure 7 clearly support this observation. An analysis of the graph in Figure 7.a reveals that the matheuristic algorithm generates solution values close to those of the mathematical model, demonstrating its effectiveness as a viable alternative in terms of solution quality.

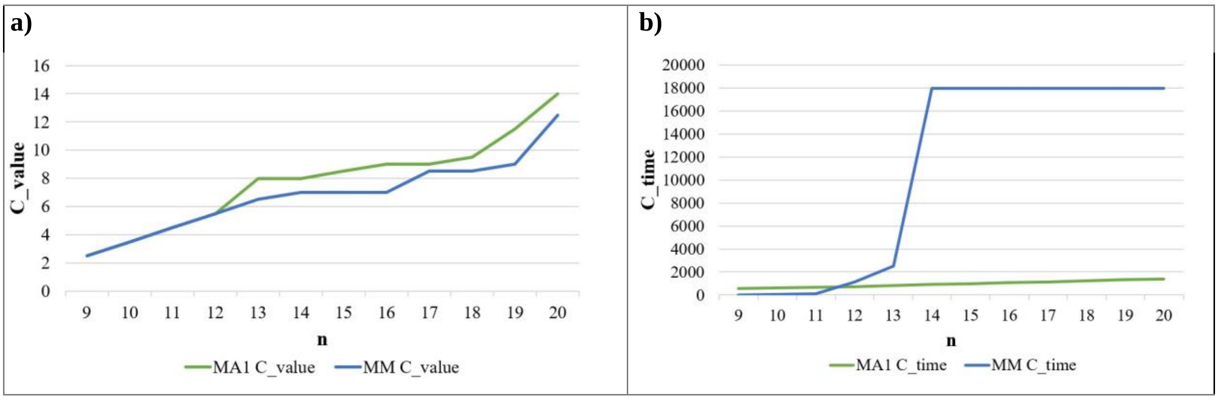

Figure 7. a) Comparison of solution values: $MM - MA1$, b) Comparison of solution times: $MM - MA1$ (a) Çözüm değerlerinin karşılaştırılması: $MM - MA1$, b) Çözüm sürelerinin karşılaştırılması: $MM - MA1$)

Figure 7.b compares the solution times of the two methods. Analyzing the graph reveals that for larger problem sets, where the number of jobs increases significantly, the solution time for the mathematical model (MM) escalates dramatically. In contrast, the $MA1$ algorithm consistently achieves results within shorter solution time.

The experimental results demonstrate that the proposed matheuristic algorithms, particularly $MA1$, produce high quality solutions in significantly shorter times compared to the mathematical model, especially for large-scale problem instances. These findings align with previous studies [12-17] in the literature that emphasize the scalability and efficiency of genetic algorithm-based methods in unrelated parallel machine scheduling problems with sequence dependent setup times. Unlike many prior works that focus primarily on minimizing makespan, the proposed approach incorporates multiple real-world constraints, such as capacity balancing and job diversity, making it more applicable to industrial environments. Furthermore, the use of a constrained random mutation operator and problem specific chromosome structure adds a layer of problem awareness that enhances search efficiency which are not commonly observed in standard genetic algorithm implementations.

5. CONLUSION (SONUÇ)

This study addresses a scheduling problem for unrelated parallel machines with sequence-dependent setup times and develops both a mathematical model and matheuristic algorithms to solve it. The mathematical model provided optimal solutions for small and medium sized problems but demonstrated practical limitations due to its exponentially increasing solution time for large-scale problems. To overcome this issue, genetic algorithm-based matheuristic methods were proposed and tested with two different crossover operators.

Experimental results and statistical analyses revealed that the proposed matheuristic algorithms, particularly for large-scale problems, could produce near-optimal solutions in significantly shorter solution times compared to the mathematical model. The $MA1$ algorithm outperformed the $MA2$ algorithm in terms of solution quality and demonstrated an ability to achieve results close to those of the mathematical model. However, no statistically significant difference was observed between the two matheuristic algorithms in terms of solution times. The proposed matheuristic algorithm produced results that are either identical or very close to the exact solutions obtained by the mathematical model for small and medium sized problems. For large scale problems, due to the time constraint (18000 seconds), the solution process of the mathematical model ended without proving optimality. Therefore, the solutions obtained by the mathematical model in large-scale problems were the best feasible solutions that were obtained within the specified time. The matheuristic algorithm, on the other hand, found solutions close to these best feasible solutions in large-scale problems in short periods of time, such as 904-1378 seconds. Therefore, the developed matheuristic approach is not only theoretically reliable but also practically suitable for generating fast and effective solutions in real-world production environments, particularly for daily scheduling tasks.

Future studies may focus on enhancing the performance of the proposed matheuristic algorithms through adaptive parameter tuning, integration with other metaheuristic strategies, or the integration of local search components to further improve solution quality. In addition, alternative objective formulations and operator designs may be explored to better accommodate highly dynamic or large-scale problems.

ACKNOWLEDGMENTS (TEŞEKKÜR)

I would like to thank the Brisa Company and Mustafa Tacettin for providing case study and data that significantly contributed to the development of this study. / Bu çalışmanın geliştirilmesinde, vaka çalışması ve veri seti sağlayarak önemli katkıda bulunan Brisa firmasına ve Mustafa Tacettin’e teşekkür ederim.

DECLARATION OF ETHICAL STANDARDS (ETİK STANDARTLARIN BEYANI)

The author of this article declares that the materials and methods they use in their work do not require ethical committee approval and/or legal-specific permission.

Bu makalenin yazarı çalışmalarında kullandıkları materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel bir izin gerektirmediğini beyan ederler.

AUTHORS’ CONTRIBUTIONS (YAZARLARIN KATKILARI)

***Merhad AY*:** He developed method, conducted the experiments, analyzed the results and performed the writing process.

Yöntem geliştirmiş, deneyleri yapmış, sonuçlarını analiz etmiş ve makalenin yazım işlemini gerçekleştirmiştir.

CONFLICT OF INTEREST (ÇIKAR ÇATIŞMASI)

There is no conflict of interest in this study.

Bu çalışmada herhangi bir çıkar çatışması yoktur.

REFERENCES (KAYNAKLAR)

[1] Lin, Y.-K., Pfund, M., & Fowler, J. (2011). Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. Computers & Operations Research, 38(6), 901-916.

[2] Su, L. (2009). Minimizing earliness and tardiness subject to total completion time in an

identical parallel machine system. Computers & Operations Research, 36(2), 461-471.

[3] Kongsri, P., & Buddhakulsomsiri, J. (2020). A Mixed Integer Programming Model for Unrelated Parallel Machine Scheduling Problem with Sequence Dependent Setup Time to Minimize Makespan and Total Tardiness. 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 605-609.

[4] Diana, R. O. M., Souza, S. R., & Wanner, E. (2021). A robust multi-response VNS-aiNet approach for solving scheduling problems under unrelated parallel machines environments. Expert Systems with Applications, 182, 115140.

[5] Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, M., Izadi, M., & Sassani, F. (2009). Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. Computers & Operations Research, 36(11), 3224-3230.

[6] Akyol, E., & Saraç, T. (2017). Paralel makina çizelgeleme problemi için bir karma tamsayılı programlama modeli: ortak kaynak kullanımı. Gazi University Journal of Science Part C: Design and Technology, 5(3), 109-126.

[7] Safaei, S., Naderi, R., Sohrabi, A., & Hatami, A. (2015). Scheduling of unrelated parallel machines using two multi-objective genetic algorithms with sequence-dependent setup times. International Journal of Advanced Design and Manufacturing Technology, 8, 11-22.

[8] Yepes-Borrero, J. C., Perea, F., Ruiz, R., & Villa, F. (2020). Bi-objective parallel machine scheduling with additional resources during setups. European Journal of Operational Research, 292(2), 443-455.

[9] Ji, B., Zhang, S., Yu, S. S., & Zhang, D. (2023). An enhanced adaptive large neighborhood search for unrelated parallel machine scheduling with sequence dependent setup times. IEEE Access.

[10] Ezugwu, A. (2021). Advanced discrete firefly algorithm with adaptive mutation-based neighborhood search for scheduling unrelated parallel machines with sequence-dependent setup times. International Journal of Intelligent Systems, 37(3), 4612-4653.

[11] Haddad, M. N., Coelho, I. M., Souza, M., Ochi, L., Santos, H., & Martins, A. X. (2012). GARP: A new genetic algorithm for the unrelated parallel machine scheduling problem with setup times. 31st International Conference of the Chilean Computer Science Society, 152-160.

[12] Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. European Journal of Operational Research, 211(3), 612-622.

[13] Özçelik, F., & Saraç, T. (2022). Kullanılamayan Zaman Dilimlerinin ve Sıra Bağımlı Hazırlık Sürelerinin Olduğu Paralel Makina Çizelgeleme Problemi. Gazi University Journal of Science Part C: Design and Technology, 10(3), 588-600.

[14] Zeidi, J. R., & Mohammad Hosseini, S. (2015). Scheduling unrelated parallel machines with sequence-dependent setup times. The International Journal of Advanced Manufacturing Technology, 81(1-4), 1487-1496.

[15] Kim, J., & Kim, H.-J. (2020). Parallel machine scheduling with multiple processing alternatives and sequence-dependent setup times. International Journal of Production Research, 59(18), 5438-5453.

[16] Antunes, A., Matos, M. A., Rocha, A. M., Costa, L., & Varela, L. (2022). A statistical comparison of metaheuristics for unrelated parallel machine scheduling problems with setup times. Mathematics.

[17] Chang, Y.-C., Chang, K.-H., & Zheng, C.-P. (2022). Application of a non-dominated sorting genetic algorithm to solve a bi-objective scheduling problem regarding printed circuit boards. Mathematics.

[18] Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. Journal of Industrial and Management Optimization, 15(3), 845-864.

[19] Bozorgirad, M. A., & Logendran, R. (2012). Sequence-dependent group scheduling problem on unrelated-parallel machines. International Journal of Production Economics, 140(2), 844-853.

[20] Fanjul-Peyro, L., Ruiz, R., & Pérez, Á. (2019). Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. Computers & Operations Research, 102, 199-213.

[21] Thiruvady, D., Blum, C., & Ernst, A. T. (2020). Solution merging in matheuristics for resource constrained job scheduling. Algorithms, 13(10), 256.

[22] Maaranen, H., Miettinen, K., & Penttinen, A. (2007). On initial populations of a genetic

algorithm for continuous optimization problems. Journal of Global Optimization, 37(3), 405-436.
[23] Cerf, R. (2015). The quasispecies regime for the simple genetic algorithm with roulette wheel selection. Advances in Applied Probability, 49(4), 903-926.
[24] Semenkin, E., & Semenkina, M. (2012). Self-configuring genetic algorithm with modified uniform crossover operator. Lecture Notes in Computer Science, 7431, 414-421.
[25] Zainuddin, F. A., Abd Samad, M. F., & Tunggal, D. (2020). A review of crossover methods and problem representation of genetic algorithm in recent engineering applications. International Journal of Advanced Science and Technology, 29(6s), 759-769.
[26] Soni, N., & Kumar, T. (2014). Study of various mutation operators in genetic algorithms. International Journal of Computer Science and Information Technologies, 5(3), 4519-4521.
[27] Rajakumar, B. R. (2013). Static and adaptive mutation techniques for genetic algorithm: a systematic comparative analysis. International Journal of Computational Science and Engineering, 8(2), 180-193.