



## A mix integer programming model and solution approach to determine the optimum machine number in the unrelated parallel machine scheduling problem

Tuğba Saraç<sup>1\*</sup>, Büşra Tutumlu<sup>2</sup>

<sup>1</sup>Industrial Engineering Department, Eskişehir Osmangazi University, Eskişehir, 26480, Türkiye

<sup>2</sup>Industrial Engineering Department, Kütahya Dumlupınar University, Kütahya, 43100, Türkiye

### Highlights:

- Unrelated parallel machine scheduling problem with sequence dependent setup times and machine eligibility restriction
- A mixed integer mathematical model has been proposed in which machines to be used are also determined
- Local search algorithm and genetic algorithm

### Keywords:

- Unrelated Parallel Machine Scheduling Problem
- Local Search Algorithm
- Genetic Algorithm
- Multi-Objective Programming

### Article Info:

Research Article  
Received: 08.02.2020  
Accepted: 06.06.2021

### DOI:

10.17341/gazimmfd.686683

### Correspondence:

Author: Tuğba Saraç  
e-mail: tsarac@ogu.edu.tr  
phone: +90 530 315 2581

### Graphical/Tabular Abstract

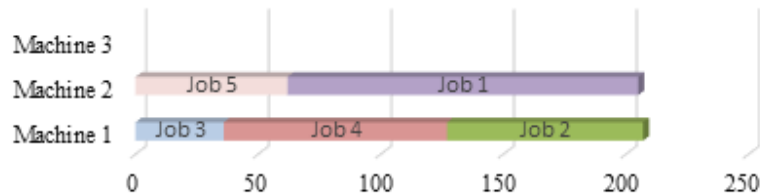


Figure A. Gantt Chart for unrelated parallel machine scheduling problem

**Purpose:** The primary purpose of this study is to present a mathematical model which determines the machines to be used for an unrelated parallel machine scheduling problem with sequence and machine-dependent setup times and machine eligibility restrictions.

### Theory and Methods:

In studies dealing with parallel machine scheduling, it is assumed that all machines will be used. However, for some businesses having special processes, where large furnaces with very intense energy consumption are used during commissioning, it can be very critical to complete jobs using the least number of furnaces. In addition, for many businesses, doing their jobs with fewer machines creates opportunities for non-used machines to be rented to another company or to accept additional jobs as much as the capacity of idle machines. For this reason, in this study, the assumption that all machines will be used has been removed, and a multi-objective mixed-integer mathematical model that will decide both which machines will be used and which jobs will be produced in which order on these machines is proposed for an unrelated parallel machine scheduling problem with sequence and machine-dependent setup times and machine eligibility restrictions. The objectives of the considered problem are minimizing the number of machines to be used and minimizing the makespan. The objective functions of the proposed multi-objective mathematical model are scalarized using the weighted sum method. A Local Search Algorithm (LS) and a Genetic Algorithm (GA) are also offered to solve large-scale problems.

### Results:

In order to show the solution performance of the mathematical model, randomly generated test problems were solved with GAMS / CPLEX. A LS and a GA have been proposed due to the lack of feasible solutions with GAMS / CPLEX for large-size problems. In the large-size problem, when all weight pairs are taken into account, genetic algorithm is more successful than local search algorithm an average of 25.64% in terms of solution quality and 50.31% in terms of time.

### Conclusion:

All studies in the literature that have addressed the parallel machine scheduling problem up to now, it is assumed that all machines will be used. However, by removing this assumption, business can reduce its excess energy consumption and they can rent non-used machines to another company or they can accept additional jobs much as the capacity of idle machines. Thus, in this study, an opportunity has been created for businesses to increase their efficiency and profitability.



## İlişkisiz paralel makine çizelgeleme probleminde eniyi makine sayısının belirlenmesi için bir doğrusal tamsayı matematiksel model ve çözüm yaklaşımları

Tuğba Saraç<sup>1\*</sup>, Büşra Tutumlu<sup>2</sup>

<sup>1</sup>Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, 26480 Eskişehir, Türkiye

<sup>2</sup>Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 43100, Kütahya, Türkiye

### Ö N E Ç İ K A N L A R

- Sıraya bağımlı hazırlık sürelerinin ve makine uygunluklarının dikkate alındığı ilişkisiz paralel makine çizelgeleme problemi
- Kullanılacak makinelerin de belirlendiği bir karma tamsayı matematiksel model
- Yerel arama algoritması ve genetik algoritma

### Makale Bilgileri

Araştırma Makalesi  
Geliş: 08.02.2020  
Kabul: 06.06.2021

### DOI:

10.17341/gazimmfd.686683

### Anahtar Kelimeler:

İlişkisiz paralel makine çizelgeleme problemi, yerel arama algoritması, genetik algoritma, çok amaçlı programlama

### ÖZ

Paralel makine çizelgeleme problemlerini ele alan çalışmalarda genellikle tüm makinelerin kullanılacağı varsayılmaktadır. Ancak devreye alınması sırasında çok yoğun enerji tüketilen büyük fırınların yer aldığı özel süreçlere sahip bazı işletmeler için işlerin en az sayıda fırın kullanılarak tamamlanması çok kritik olabilmektedir. Ayrıca pek çok işletme için de işlerini daha az makine ile gerçekleştirmek, üretimde kullanılmayan makinelerin başka bir işletmeye kiralanabilmesi veya boş kalan makinelerin kapasitesi kadar ek iş kabul edebilmesi fırsatlarını yaratmaktadır. Bu nedenle bu çalışmada, tüm makinelerin kullanılacağı varsayımı kaldırılmıştır, sıra ve makine bağımlı hazırlık sürelerinin ve makine uygunluklarının dikkate alındığı ilişkisiz paralel makine çizelgeleme probleminde hem hangi makinelerin kullanılacağına hem de kullanılacak makinelerde hangi işlerin hangi sırada üretileceğine karar verecek bir matematiksel model önerilmiştir. Ele alınan problemin amaçları, kullanılacak makine sayısının ve son işin tamamlanma zamanının enküçüklenmesidir. Önerilen çok amaçlı matematiksel modelin amaç fonksiyonları, ağırlıklı toplam yöntemi kullanılarak birleştirilmiştir. Matematiksel modelin çözüm performansının gösterilebilmesi için rassal türetilen test problemleri, GAMS/CPLEX ile çözülmüştür. Büyük boyutlu problemlerin çözümünde GAMS/CPLEX ile çözüm elde edilememesi nedeniyle bir yerel arama algoritması ve bir genetik algoritma önerilmiştir. Büyük boyutlu problem için tüm ağırlık çiftleri dikkate alındığında, genetik algoritma, yerel arama algoritmasından çözüm kalitesi açısından ortalama %25,64, süre açısından ise ortalama %50,31 daha başarılı olmuştur.

## A mix integer programming model and solution approach to determine the optimum machine number in the unrelated parallel machine scheduling problem

### H I G H L I G H T S

- Unrelated parallel machine scheduling problem with sequence dependent setup times and machine eligibility restriction
- A mixed integer mathematical model has been proposed in which machines to be used are also determined
- Local search algorithm and genetic algorithm

### Article Info

Research Article  
Received: 08.02.2020  
Accepted: 06.06.2021

### DOI:

10.17341/gazimmfd.686683

### Keywords:

Unrelated parallel machine scheduling problem, local search algorithm, genetic algorithm, multi-objective programming

### ABSTRACT

It is assumed that all machines will be used in studies dealing with parallel machine scheduling problems. However, for some businesses having special processes, where large furnaces with very intense energy consumption are used during commissioning, it can be very critical to complete jobs using the least number of furnaces. In addition, for many businesses, doing their jobs with fewer machines creates opportunities for unused machines to be rented to another company or to accept additional jobs as much as the capacity of idle machines. For this reason, in this study, the assumption that all machines will be used has been removed and a mathematical model has been proposed that will decide both which machines will be used and which jobs will be produced in which order on these machines, for the unrelated parallel machine scheduling problem with sequence and machine dependent setup times and machine eligibility restriction. The objectives of the considered problem are minimizing the number of machines to be used and the completion time of the last job. The objective functions of the proposed multi-objective mathematical model are scalarized using the weighted sum method. In order to show the solution performance of the mathematical model, randomly generated test problems were solved with GAMS / CPLEX. To solve the large problems, a local search algorithm and a genetic algorithm have been proposed due to the lack of feasible solutions with GAMS / CPLEX. In the large-scale problem, when all weight pairs are taken into account, genetic algorithm is more successful than local search algorithm an average of 25.64% in terms of solution quality and 50.31% in terms of time.

## 1. GİRİŞ (INTRODUCTION)

Çizelgeleme, belli bir zaman periyodunda bir veya birden fazla amacı eniyileyecek şekilde kaynakların faaliyetlere atıldığı ve birçok üretim ve hizmet işletmesinde düzenli bir şekilde kullanılan bir karar verme sürecidir [1]. Üretime dayalı işletmelerde çizelgeleme, işlerin makinelerle hangi sırada atanacağı ve hangi kaynaklarla üretileceğinin belirlenmesine yöneliktir [2]. Günümüz rekabet ortamında işletmelerin en önemli problemlerinden birisi, üretim kaynaklarının en verimli şekilde kullanılmasını sağlayan etkin çizelgeler oluşturmaktır. Etkin çizelgeler, siparişlerin gecikme olmadan tamamlanmasına ve fazla mesai maliyetlerinin en aza indirilmesine olanak sağlar. İşletmelerde iş sayısı arttıkça etkin çizelgeler oluşturmak zorlaşmaktadır. Bu nedenle literatürde çizelgeleme problemi üzerine birçok çalışma bulunmaktadır.

Klasik makine çizelgeleme problemlerinde, makine sayısı sabit kabul edilerek işlerin gerçekleştirilmesi için tüm makineler çizelgenmektedir. Ancak seramik sektörü gibi büyük fırınların kullanımına ihtiyaç duyulan bazı sektörlerde, bir fırının devreye alınması çok ciddi enerji tüketimine yol açmaktadır. Bu nedenle bu tip sektörlerde gerekmedikçe bir fırının devreye alınmaması tercih edilir. Ayrıca diğer pek çok işletme için de işlerini daha az makine ile gerçekleştirmek, üretimde kullanılmayan makinelerin başka bir işletmeye kiralanabilmesi veya boş kalan makinelerin kapasitesi kadar ek iş kabul edebilmesi fırsatlarını yaratmaktadır. Bu nedenle bu çalışmada, kullanılacak makine sayısının en küçüklenmesi amaçlanmıştır. Böylelikle işletmelerin karlılığı ve verimliliği artırılabilir.

Bu çalışmada, kullanılacak makinelerin de belirlendiği ilişkisiz paralel makinelerin çizelgenmesi problemi ele alınmıştır. Kullanılacak makine sayısı azaltıldığında doğal olarak son işin tamamlanma zamanı artacaktır. Bu nedenle geliştirilen matematiksel modelin ikinci amacı son işin tamamlanma zamanının en küçüklenmesi olarak belirlenmiştir. Böylece bu iki amacın dengeli bir şekilde ele alınmasıyla kullanılabilir çözümlerin üretilmesi amaçlanmıştır. Ayrıca çalışmanın gerçek hayatta sıklıkla karşılaşılmaması nedeniyle, işlerin sıra ve makine bağımlı hazırlık süreleri ve makine uygunlukları da dikkate alınmıştır. Hazırlık sürelerinin sıraya ve makineye bağımlı olması, bir işin hazırlık süresinin hem kendisinden önceki işe hem de atıldığı makineye göre farklılık göstermesi anlamına gelmektedir. Makine uygunluğu ise her işin her makinede üretilemediğini ifade etmektedir. Bu çalışmada, ele alınan problem için bir karma tamsayılı programlama modeli önerilmiştir. İlişkisiz paralel makine çizelgeleme problemi NP-zordur [3]. Bu nedenle, büyük boyutlu problemlerin çözümlenmesi için bir yerel arama algoritması ve bir genetik algoritma geliştirilmiştir.

Çalışmanın izleyen bölümünde incelenen literatür taraması verilmiştir. Üçüncü bölümde problem tanımlanmış ve bir karma

tamsayılı programlama modeli önerilmiştir. Dördüncü bölümde önerilen yerel arama algoritması ve genetik algoritma açıklanmıştır. Beşinci bölümde sayısal sonuçlar ve son bölümde ise sonuç ve öneriler verilmiştir.

## 2. LİTERATÜR TARAMASI (LITERATURE REVIEW)

Literatürde ilişkisiz paralel makine çizelgeleme problemi ele alan çok sayıda çalışma bulunmaktadır. Hazırlık sürelerinin ele alındığı ilişkisiz paralel makine çizelgeleme problemi ile ilgili son 10 yılda yapılan çalışmalardan erişilebilenler Tablo 1'de verilmiştir.

Sıralamaya bağlı hazırlık sürelerinin yanı sıra makine uygunluk kısıtları da son yıllarda dikkate alınan kısıtlardan birisidir [14, 56, 57]. Tablo 1'den de görülebileceği gibi, çalışmalarda genellikle birden fazla amaç fonksiyonu dikkate alınmış ve çözüm yöntemi olarak sezgisel ya da meta sezgisel yöntemler kullanılmıştır. Tablo 1'de yer alan çok amaçlı çalışmalar incelendiğinde çözüm yöntemi olarak genellikle ağırlıklı toplam yönteminin kullanıldığı göze çarpmaktadır. Ancak Lei vd. [6], Sarıçiçek [7] ve Afralirad ve Rezaeian [18] çalışmalarında baskınlık yöntemini ve Cota vd. [10] Tchebycheff fonksiyonunu kullanmıştır. Dolgui vd. [54] ise ele aldıkları problemi iki alt probleme ayrılarak çözüme ulaştırmışlardır. Bu çalışmada ise, amaç fonksiyonları literatürdeki çalışmaların geneli ile benzer şekilde ağırlıklı toplam yöntemi ile birleştirilmiştir. Ayrıca çözüm yöntemi olarak ta yine literatürde çok tercih edilen GA ve YA algoritmaları kullanılmıştır. Çalışmalar ele alınan amaç fonksiyonları açısından incelendiğinde ise erişilen literatürde daha önce hiç yer almamış olan kullanılacak makine sayısı toplamının ( $\sum y_i$ ) en küçüklenmesi amacı ilk defa ele alınmıştır. Literatürde erişilen çalışmaların tamamında kullanılacak makine sayısı belirli ve sabit kabul edilmiştir. Bu çalışmayı literatürdeki diğer çalışmalardan ayıran en önemli özellik, sadece işlerin hangi makinede hangi sırada üretileceğini belirlemekle sınırlı kalmayıp hangi makinelerin kullanılacağına da karar verilmesidir.

## 3. PROBLEMİN TANIMLANMASI VE MATEMATİKSEL MODEL (PROBLEM DEFINITION AND MATHEMATICAL MODEL)

Ele alınan problemde hangilerinin kullanılacağına karar verilecek  $m$  tane ilişkisiz paralel makinede işlem görecektir  $n$  tane iş bulunmaktadır. İşlerin hazırlık süreleri sıra ve makine bağımlıdır. İşlem süreleri ise makinelerle göre farklılık göstermektedir. Ayrıca her iş her makinede işlenmemektedir. İşlerin bölünmesine izin verilmemektedir ve işler sıfır zamanında hazırır. Her makine aynı anda sadece bir iş gerçekleştirebilmektedir. Amaç fonksiyonlarından birisi son işin tamamlanma zamanının ( $C_{max}$ ) en küçüklenmesidir. Diğerleri ise işlerin işlem göreceği toplam makine sayısının en küçüklenmesidir. Problemin matematiksel modeline ait kümeler, indisler, parametreler, karar değişkenleri, kısıtlar ve amaç fonksiyonları aşağıda verilmiştir;

**Tablo 1.** Sıra ve makine bağımlı hazırlık sürelerinin bulunduğu ilişkisiz paralel makine çizelgeleme problemi ile ilgili yapılan çalışmalar (Literature review for unrelated parallel machine scheduling problem with sequence and machine dependent setup times)

Yıl	Yazar	Amaç	Yöntem
2021	Bu çalışma	$C_{enb}, \sum y_i$	GA, YA
2021	Cota vd.[4]	$C_{enb}, TEC$	$\epsilon$ -Kısıt, SPM
2020	Kim ve Kim [5]	$C_{enb}, PBZ-GBZ$	SY
2020	Lei vd.[6]	$C_{enb}, \sum T_j$	ABC
2020	Sarıççek [7]	$JT, C_{enb}$	TB, TA
2020	Lei vd.[8]	$C_{enb}$	ABC
2020	Yepes-Borrero vd.[9]	$C_{enb}$	SY
2019	Cota vd.[10]	$C_{enb}, TEC$	LNS
2019	Jouhari vd.[11]	$C_{enb}$	TB ve SCA
2019	Ekici vd.[12]	$\sum E_j, \sum T_j$	SA, TA, RKY, SY
2019	Ezugwu[13]	$C_{enb}$	SOA, TB
2019	Bektur ve Saraç[14]	$\sum w_j T_j$	TA, TB
2019	Fanjul-Peyro vd.[15]	$C_{enb}$	MPA
2018	Ezugwu ve Akutsah[16]	$C_{enb}$	FA
2018	Tozzo vd.[17]	$C_{enb}$	GA, VNS
2017	Afzalirad ve Rezaeian[18]	MWFT, MWT	NSGA- II, Çok Amaçlı KKA
2017	Shahvari ve Logendran[19]	$\sum w_j C_j, \sum w_j T_j$	TA
2016	Afzalirad ve Rezaeian[20]	$C_{enb}$	GA, AIS
2016	Mir ve Rezaeian[21]	$\sum C_j$	PSO, GA
2015	Canıyılmaz vd.[22]	$C_{enb}, \sum T_j$	GA, ABC
2015	Diana vd.[23]	$C_{enb}$	AIS
2015	Joo ve Kim[24]	$\sum T_j$	GA
2015	Rosales vd.[25]	$C_{enb}$	YA, MSA Temelli SY
2014	Arnout vd.[26]	$C_{enb}$	KKA
2014	Eroğlu vd.[27]	$C_{enb}$	YA, GA
2014	Kayvanfar ve Teymourian[28]	$\sum (E_j + T_j), C_{enb}$	ZSD
2014	Lee vd.[29]	$\sum C_j$	SY
2014	Li ve Milne[30]	TSC	SY
2014	Lin ve Hsieh[31]	$\sum w_j T_j$	SY
2014	Lin ve Ying[32]	$C_{enb}$	ABC
2014	Nogueira vd.[33]	$\sum E_j, \sum T_j$	Grasp, Path Relinking, İteratif YA
2014	Rambod ve Rezaeian[34]	$C_{enb}$	GA, ABC
2013	Lee vd.[35]	$\sum T_j$	TA, TB
2013	Torabi vd.[36]	$C_{enb}$	PSO
2012	Bozogirad ve Logendran[37]	$\sum w_j C_j, \sum w_j T_j$	TA
2012	Chen[38]	$\sum w_j U_j$	SY
2012	Coelho vd.[39]	$C_{enb}$	YA Tabanlı SY
2012	Fleszar vd.[40]	$C_{enb}$	Matematiksel Modelle Melezlenmiş SY
2012	Joo ve Kim[41]	$\sum T_j$	GA
2012	Wang vd.[42]	$\sum w_j U_j, C_{enb}$	YA
2012	Ying vd.[43]	$C_{enb}$	TB
2012	Ying ve Lin[44]	$\sum T_j$	ABC
2011	Chang ve Chen[45]	$C_{enb}$	GA, TB
2011	Hsu vd.[46]	$\sum C_j$	PZ
2011	Kuo vd.[47]	$\sum C_j$	PZ
2011	Lin vd.[48]	$\sum C_j$	İteratif AA
2011	Mehravaran ve Logendran [49]	$\sum w_j C_j, \sum w_j T_j$	TA
2011	Ruiz ve Romano[50]	$\sum C_j, R$	SY
2011	Vallada ve Ruiz[51]	$C_{enb}$	YA, GA
2010	Arnout vd.[52]	$C_{enb}$	KKA
2010	Chyu ve Chang[53]	$\sum w_j C_j, \sum w_j T_j$	Pareto Evrimsel Yaklaşım, TB
2010	Dolgui vd.[54]	$C_{enb}, T_{enb}$	Karmaşıklık Analizi, AA
2010	Paula vd.[55]	$\sum w_j T_j$	Lagrange Gevşetmesi Temelli RCA

$C_{enb}$ : Son işin tamamlanma zamanının enküçüklenmesi,  $T_{enb}$ : En büyük gecikmenin enküçüklenmesi, MWFT: Ağırlıklandırılmış ortalama akış zamanının enküçüklenmesi, MWT: Ağırlıklandırılmış ortalama gecikmenin enküçüklenmesi,  $\sum C_j$ : Toplam tamamlanma zamanının enküçüklenmesi,  $\sum T_j$ : Toplam geç tamamlanma zamanının enküçüklenmesi,  $\sum E_j$ : Toplam erkenliğin en küçükleme,  $\sum w_j U_j$ : Toplam geciken iş sayısının en küçükleme, TSC: Toplam hazırlık maliyetinin en küçükleme,  $\sum w_j C_j$ : Ağırlıklandırılmış toplam tamamlanma zamanının enküçüklenmesi,  $\sum w_j T_j$ : Ağırlıklandırılmış toplam geç tamamlanma süresinin enküçüklenmesi,  $\sum w_j U_j$ : Ağırlıklandırılmış toplam geciken iş sayısının en küçükleme, R: ikincil kaynak miktarı en küçükleme, TEC: Toplam enerji tüketimin en küçükleme, PBZ-GBZ: Planlanan başlangıç zamanı ile gerçekleşen başlangıç zamanı arasındaki sapmanın enküçükleme, JT: İşlerin tercihlerini enbüyükleme, GA: Genetik algoritma, TB: Tavlama benzetimi, TA: Tabu arama algoritması, YA: Yerel arama algoritması, SOA: Simbiyotik organizmalar arama algoritması, PSO: Parçacık sürüşü optimizasyonu, KKA: Karınca kolonisi algoritması, LNS: Büyük komşuluk arama algoritması, VNS: Değişken komşuluk arama algoritması, SCA: Sine cosine algoritması, RKY: Rasgele kümeleme yaklaşımı, SA: Sıralama algoritması, SY: Sezgisel yöntem, FA: Ateş böceği algoritması, ABC: Yapay arı kolonisi algoritması, RCA: Relax and cut algoritması, AIS: Yapay bağıklık sistemleri, MPA: Matematiksel programlama tabanlı algoritma, MSA: Multi-Start algoritması, ZSD: Zeki Su Damlacıkları Algoritması, PZ: Polinom zamanlı algoritma, AA: Açgözlü algoritma, SPM: Akıllı Havuz Arama Metodu, EST-H: En Erken Tamamlanma Zamanı (EST) tabanlı Sezgisel Algoritma, UTI: İz Yoğunluğu Güncelleme

*Kümeler ve İndisler:*

$N$ : İş kümesi,  $N = \{1, 2, \dots, n\}$

$M$ : Makine kümesi,  $M = \{1, 2, \dots, m\}$

$i, j \in N$  ve belirli bir işi göstermek için kullanılan indislerdir.

$k \in N$  ve iş sırasını göstermek için kullanılan indistir.

$l \in M$  ve belirli bir makineyi göstermek için kullanılan indistir.

*Parametreler:*

$p_{jl}$  :  $j$ . işin  $l$ . makedeki işlem süresi

$h_{jl}$  :  $j$ . işin  $l$ . makede ilk sıraya atanması durumunda hazırlık süresi

$s_{ijl}$  :  $i$  işi  $j$  işinden önce  $l$ . makede çizelgelenmiş sıra ve makine bağımlı hazırlık süresi

$b_{jl}$  : Eğer  $j$  işi  $l$ . makede işlem görebiliyorsa 1, diğer durumda 0

$M$  : Yeterince büyük bir pozitif sayı

*Karar Değişkenleri:*

$x_{jkl}$  : eğer  $j$ . iş  $k$ . sırada  $l$ . makineye atandıysa 1; diğer durumda 0

$C_j$  :  $j$ . işinin tamamlanma zamanı

$C_{enb}$  : Son işin tamamlanma zamanı

$y_l$  :  $l$ . makine seçildiyse 1; diğer durumda 0

*Amaç fonksiyonları:*

$$enk f_1 = C_{enb} \quad (1)$$

$$enk f_2 = \sum_l y_l \quad (2)$$

*Kısıtlar:*

$$C_j + M(1 - x_{jkl}) \geq h_{jl} + p_{jl} \quad \forall j, k, l \quad k = 1 \quad (3)$$

$$C_j - M(1 - x_{jkl}) \leq h_{jl} + p_{jl} \quad \forall j, k, l \quad k = 1 \quad (4)$$

$$C_j + M(2 - x_{jkl} - x_{i(k-1)l}) \geq C_i + p_{jl} + s_{ijl} \quad \forall i, j, k, l \quad i \neq j, k > 1 \quad (5)$$

$$C_j - M(2 - x_{jkl} - x_{i(k-1)l}) \leq C_i + p_{jl} + s_{ijl} \quad \forall i, j, k, l \quad i \neq j, k > 1 \quad (6)$$

$$C_{enb} \geq C_j \quad \forall j \quad (7)$$

$$\sum_j x_{jkl} \leq 1 \quad \forall k, l \quad (8)$$

$$\sum_k \sum_l x_{jkl} = 1 \quad \forall j \quad (9)$$

$$\sum_j x_{jkl} - \sum_i x_{i(k-1)l} \leq 0 \quad \forall k, l \quad k > 1 \quad (10)$$

$$b_{jl} \geq x_{jkl} \quad \forall j, k, l \quad (11)$$

$$x_{jkl} \leq y_l \quad \forall j, k, l \quad (12)$$

$$\sum_j \sum_k x_{jkl} \geq y_l \quad \forall l \quad (13)$$

$$x_{jkl} \in \{0,1\} \quad \forall i, k, l \quad (14)$$

$$y_l \in \{0,1\} \quad \forall l \quad (15)$$

$$C_j \geq 0 \quad \forall j \in N \quad (16)$$

$$C_{enb} \geq 0 \quad (17)$$

Eş. 1 ve Eş. 2, amaç fonksiyonlarını göstermektedir. Eş. 1 son işin tamamlanma zamanını ve Eş. 2 işlerin işlem göreceği toplam makine sayısını en küçükmektir. Eş. 3 ve Eş. 4 her makedeki ilk işlerin ve Eş. 5 ve Eş. 6 ilk sıra haricindeki işlerin tamamlanma zamanını belirlemektedir. Eş. 7 son işin tamamlanma zamanını belirlemektedir. Eş. 8 ve Eş. 9 sırasıyla bir makinenin bir sırasına en fazla bir işin atanabilmesini ve her işin mutlaka bir makinenin bir sırasına atanmasını sağlamaktadır. Eş. 10 aynı makineye atanan işlerin sıra atlamadan işlenmesini sağlamaktadır. Eş. 11 işlerin sadece işlem görebileceği makinelere atanmasını sağlamaktadır. Eş. 12 ve Eş. 13 bir makineye iş atanması durumunda, o makinenin seçildiğini gösteren karar değişkeninin değer almasını sağlayan ilişki kısıtlarıdır. Eş. 14, Eş. 15, Eş. 16 ve Eş. 17 karar değişkenlerin işaret kısıtlarıdır.

#### 4. İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN GELİŞTİRİLEN METASEZGİSEL ALGORİTMALAR

(METAHEURISTIC ALGORITHMS DEVELOPED FOR THE UNRELATED PARALLEL MACHINE SCHEDULING PROBLEM)

Ele alınan problem, NP-zor yapıda olması nedeniyle problemin çözümünde metasezgisel algoritmalar kullanılmıştır. Kullanılan algoritmalar; tek nokta temelli algoritmalarından yerel arama algoritması ve popülasyon temelli algoritmalarından da genetik algoritmadır. Bu algoritmaların seçilmesinin nedeni, Tablo 1'den de görülebileceği gibi literatürde daha önce benzer problemlerin çözümünde kullanılmış ([11], [14-16], [18, 19], [21], [27, 28], [33], [35, 36], [39], [45]) ve başarılı çözümler elde etmiş olmalarıdır.

##### 4.1. Algoritmalarda Kullanılan Gösterim Yapısı (Representation Structure Used in Algorithms)

Gösterim yapısı, problemin kısıtları ve özellikleri dikkate alınarak oluşturulmuştur. Oluşturulan yapıda, her işin işlem görebileceği makinelerden sadece birine atanması sağlanmalıdır. Ayrıca işlerin hangi makineye atandığı ve kaçınıcı sırada işlem göreceği gösterilmelidir.

Literatürde makine çizelgeleme problemlerinde genellikle permütasyon gösterimi kullanılmaktadır. Bu çalışmada permütasyon gösterimi yerine, her iş her makede işlenememesinden dolayı yapılması gereken onarım işlemini engelleyecek bir gösterim yapısı tasarlanmıştır. Çünkü onarım işlemi algoritmaların çözüm süresini uzatarak

verimliliği azaltmaktadır. Tasarlanan 2 katmanlı yapı Şekil 1’de verilmiştir. 1. katman işlerin hangi makinelere atandığını, 2. katman ise işlerin sıralanması için gerekli rassal sayıları göstermektedir. Aynı makineye atanan işlerden hangisinin rassal sayısı daha büyükse, o iş daha önce işlenmektedir. Gösterim yapısının boyutu ise iş sayısına göre değişmektedir. Şekil 1’de verilmiş örnekte 5 iş bulunmaktadır. 1, 2 ve 4. işlerin 1. makineye, 3 ve 5. işlerin ise 2. makineye atandığı gösterilmektedir. 1. makinede işlerin sırası; 4-1-2 ve 2. makinede işlerin sırası ise; 5-3 şeklindedir.

	1.iş	2.iş	3.iş	4.iş	5.iş
1.Katman	1	1	2	1	2
2.Katman	0,87	0,65	0,10	0,91	0,53

Şekil 1. Tasarlanan gösterim yapısı (Designed display structure)

#### 4.2. Amaçların Birleştirilmesi (Scalarization of Objective Functions)

Bu çalışmada ele alınan amaçların birleştirilmesi için Ağırlık Toplam Yöntemi kullanılmıştır. Bu yöntemde, birleştirilmiş amaç fonksiyonu ( $z$ ), amaç fonksiyonlarının önem derecelerine karşı gelen ağırlıklar ile çarpılarak toplanması ile elde edilmektedir. İlgili formül Eş. 18’de verilmiştir.

$$enk z = \sum_{r=1}^{\rho} w_r f_r(x) \quad (18)$$

Eş. 18’ de  $w_r$ ,  $r$ . amaç fonksiyonu için belirlenen ağırlık,  $f_r(x)$ ,  $r$ . amaç fonksiyonu ve  $\rho$  amaç fonksiyonu sayısıdır.  $w_r$ , sıfırdan büyük olmalıdır. Ağırlıklı toplam yönteminde eğer amaç fonksiyonlarının alabileceği değerler farklı ölçekte ise, amaç fonksiyonlarının normalleştirilmesi gerekir. Bu çalışmada birinci amaç fonksiyonu, tamamlanma zamanının üst sınırına ( $\bar{ÜS} C_{enb}$ ), ikinci amaç fonksiyonu ise toplam makine sayısına ( $m$ ) bölünerek normalleştirilmiştir.  $\bar{ÜS} C_{enb}$  değerinin hesaplanmasında kullanılan formüller Eş. 19, Eş. 20 ve Eş. 21’de ve normalleştirilmiş amaçların birleştirildiği amaç fonksiyonu ise Eş. 22’de verilmiştir.

$$\hat{p}_j = enb_{il}\{p_{jl}\} \quad \forall j \quad (19)$$

$$\hat{s}_j = enb_{il}\{s_{jl}\} \quad \forall j \quad (20)$$

$$\bar{ÜS} C_{enb} = \frac{1}{m} (\sum_j \hat{p}_j + \sum_j \hat{s}_j + enb_{il}\{h_{jl}\}) \quad (21)$$

$$enk z = w_1 (f_1 / \bar{ÜS} C_{enb}) + w_2 (f_2 / m) \quad (22)$$

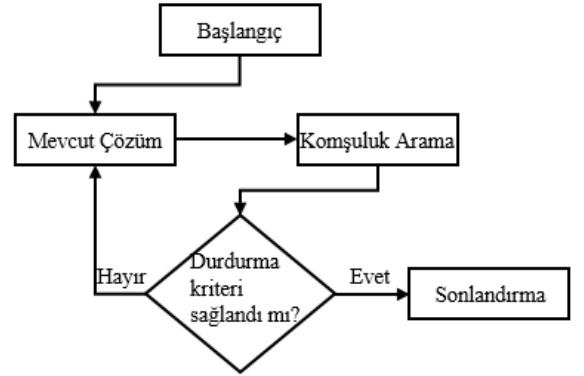
Eş. 19’da her işin makinelerdeki en büyük işlem süresi ( $\hat{p}_j$ ) ve Eş. 20’de her işin en büyük sıra bağımlı hazırlık süresi ( $\hat{s}_j$ ) belirlenmektedir. Eş. 21 ise  $\bar{ÜS} C_{enb}$  değerini hesaplamak için kullanılmaktadır.

#### 4.3. Önerilen Yerel Arama Algoritması (Proposed Local Search Algorithm)

Yerel arama algoritması (YA), en eski ve en basit metasezgisel yöntemlerden birisidir [58]. Hızlı bir şekilde iyi

çözümlere ulaşabilen bir yöntem olduğundan literatürde yaygın olarak kullanılmaktadır. YA, bir başlangıç çözüm ile başlar ve her iterasyonda amaç fonksiyonunu iyileştiren bir komşu çözüm mevcut çözümün yerini alarak süreç devam eder. Bu süreç mevcut çözümde daha fazla bir iyileşme olmayıncaya kadar veya iterasyon sayısına ulaşınca kadar devam eder. Şekil 2’ de YA’nın genel yapısı gösterilmiştir. YA’nın adımları aşağıda verilmiştir;

- Adım 1* : Başlangıç bir çözüm üret ve bu çözümü mevcut çözüm olarak kaydet.  
*Adım 2* : Komşu çözüm üretme operatörlerden birini kullanarak belirli sayıda komşu çözümler üret.  
*Adım 3* : Üretilen komşu çözümlerden en iyi çözümü mevcut çözüm olarak kaydet.  
*Adım 4* : Daha iyi bir çözüm bulunamayınca veya daha önceden belirlenmiş iterasyon sayısına ulaşınca kadar Adım 2 ve Adım 3’ü tekrarla. Eğer daha iyi bir çözüm yoksa veya iterasyon sayısına ulaşıldıysa algoritmayı sonlandır.



Şekil 2. Yerel arama algoritmasının genel yapısı  
(General structure of local search algorithm)

*Başlangıç çözüm oluşturma*: İşlerin işlem görebildiği makineler atanmasını sağlamak için her işin atanabileceği makineler kümesi oluşturulur. Bu kümelerden her iş için rasgele makine seçimleri yapılarak başlangıç çözümün birinci katmanı ve rassal sayılar türetilerek de ikinci katmanı oluşturulmaktadır. Örneğin; 5 iş ve 3 makine olsun. İşlerin atanabileceği makine kümeleri sırasıyla;  $J_1 = \{1,2\}$ ,  $J_2 = \{1,2,3\}$ ,  $J_3 = \{2,3\}$ ,  $J_4 = \{1,2,3\}$ ,  $J_5 = \{1,2,3\}$ ’dür. Şekil 3’de bir başlangıç çözüm için örnek verilmiştir. Toplam makine sayısı üç olmasına rağmen bu başlangıç çözümde sadece iki makine kullanıldığı görülmektedir. 1 ve 4. iş 1. makineye ve 2, 3 ve 5. iş ise 2. makineye atandığı ve 1. makinede işlerin sırası 4-1 ve 2. makinede işlerin sırası ise 3-5-2 şeklinde olduğu gösterilmiştir.

1	2	2	1	2
0,17	0,55	0,70	0,21	0,58

Şekil 3. Başlangıç çözüm (Initial solution)

**Komşu çözümler üretme:** Literatürde ele alınan problemde kullanılan sezgisel algoritmalar komşu çözümler üretmek için araya ekleme, ikili yer değiştirme ve ters çevirme gibi operatörler kullanılmaktadır. Fakat bu çalışmada bu operatörler, sadece birinci amaç fonksiyonunun değerinde değişiklik sağlayıp, ikinci amaç fonksiyonunun değerinde hiçbir değişiklik sağlanamamasından dolayı kullanılmamıştır. Ayrıca bu operatörlerin kullanması, her iş her makinede işlem göremediğinden onarım işlemi gerektirmektedir. Bu nedenle komşu çözümlerin üretilmesi için rasgele bir iş belirlenip atandığı makine, o işin atanabileceği makine kümesinden rasgele seçilip değiştirilmektedir. Aynı zamanda seçilen işin ikinci katmanındaki rassal sayı yerine de yeniden farklı bir rassal sayı atanmaktadır. Şekil 4’de komşu üretme işlemi için bir örnek verilmiştir.

seçilen iş ↓				
1	2	2	1	2
0,17	0,55	0,70	0,21	0,58
mevcut çözüm				
1	2	2	1	3
0,17	0,55	0,7	0,21	0,42
komşu çözüm				

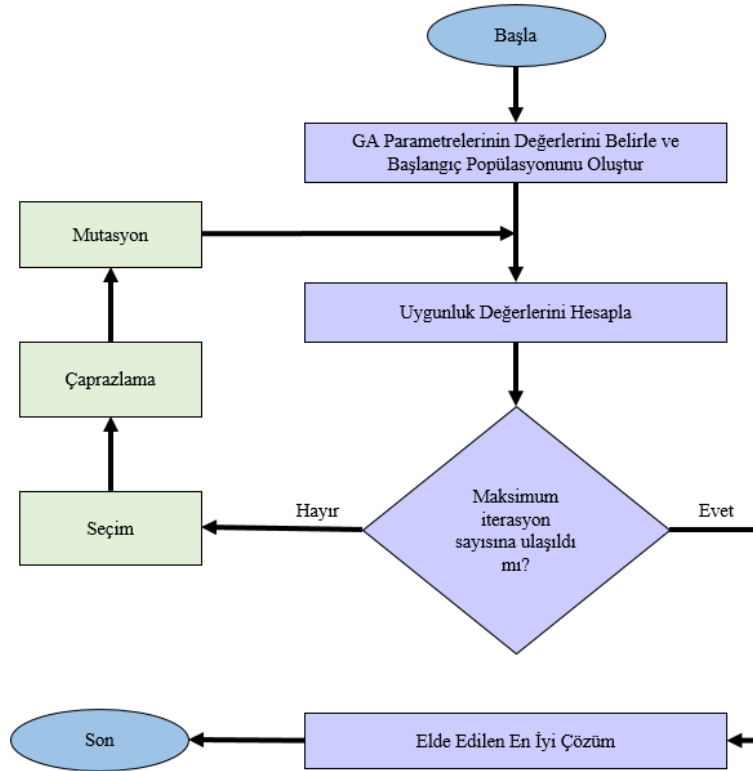
Şekil 4. Komşu üretme işlemi (Neighbor generation process)

#### 4.4. Önerilen Genetik Algoritma (Proposed Genetic Algorithm)

Genetik algoritma (GA), 1970’li yıllarda John Holland tarafından ortaya atılmıştır. Genetik algoritmaları, doğadaki evrime dayanan güçlü ve etkili araştırma algoritmaları olarak tanımlanmıştır [59]. GA ile zor problemlerin çözümünde daha hızlı ve eniyi çözüme yakın çözümler elde edilmektedir. Tüm uzayı taramadıklarından dolayı etkin arama yaparak geleneksel yöntemlere göre çok daha kısa bir sürede çözüme ulaşırlar [60]. GA, popülasyon temelli algoritmalarından biri olduğu için aynı anda birden fazla çözümü değerlendirmektedir. Böylelikle aynı anda uzayda birden fazla noktayı değerlendirip yerel en iyi noktaya takılmamaktadır. Şekil 5’ de GA’nın genel yapısı gösterilmiştir ve adımları aşağıda verilmiştir.

- Adım 1 :** GA parametre değerlerini belirle ve başlangıç popülasyonunu oluştur.
- Adım 2 :** Bireylerin uygunluk değerlerini hesapla.
- Adım 3 :** Bir sonraki nesle iyi bireylerin geçme şansını artırmak için seçim işlemini gerçekleştir.
- Adım 4 :** Bireyler arasında gen alışverişinin sağlanması için çaprazlama işlemini gerçekleştir.
- Adım 5 :** Çeşitliliği artırmak için mutasyon işlemini gerçekleştir.
- Adım 6 :** Enbüyük iterasyon sayısına (nesil sayısına) ulaşıldıysa DUR, ulaşılmadıysa Adım 2’ye DÖN.

Parametrelerin değerlerinin belirlenmesi: GA’nın performansını etkileyen en önemli adımlarından biri



Şekil 5. Genetik algoritmanın genel yapısı (General structure of genetic algorithm)

parametre değerlerine karar verilmesidir. GA parametreleri: popülasyon büyüklüğü, çaprazlama oranı, mutasyon olasılığı, elitizm sayısı ve iterasyon sayısıdır.

**Başlangıç popülasyonunun oluşturulması:** Yerel arama algoritmasının başlangıç çözümünde bahsedilen durumlar bu algortimada da geçerlidir. Ancak yerel aramada bir çözüm üretirken, GA' da popülasyon büyüklüğü kadar çözüm üretilmektedir. Şekil 6'da GA başlangıç popülasyonu için bir örnek gösterilmiştir.

Birey 1	2	2	3	3	2
	0,51	0,12	0,77	0,24	0,29
Birey 2	1	2	2	1	1
	0,25	0,97	0,17	0,14	0,61
.			.		
.			.		
.			.		

Şekil 6. GA Başlangıç popülasyonu (GA initial population)

**Seçim işlemi:** Popülasyondaki bireylere seçim yöntemlerinden birisi uygulanarak seçilen bireylerin gelecek nesille aktarılması işlemidir. Rassal seçim, rulet çarkı, turnuva yöntemi, BEST T% vb. seçim yöntemleri bulunmaktadır. Bu çalışmada, literatürde yaygın olarak kullanılan İkili turnuva yöntemi kullanılmıştır. İkili turnuva yönteminde, rasgele iki birey seçilir ve seçilen bireylerin hangisinin uygunluk değeri daha iyiyse o birey bir sonraki nesille aktarılır. Popülasyon büyüklüğü kadar bu işlem tekrarlanır. Böylelikle hem iyi bireyler bir sonraki nesille aktarılmakta ve hem de popülasyon büyüklüğü değişmemektedir.

**Çaprazlama işlemi:** Popülasyondaki bireylerin birbirleri ile gen aktarımı yapılarak gelecek nesilde daha iyi bireyler elde edilebilmektedir. Bu işlemde farklı çözümler arasında bilgi

değişimi yoluyla arama uzayının araştırılmamış bölgelerine ulaşım sağlanır [61]. Popülasyonda, '*çaprazlama oranı × popülasyon büyüklüğü*' kadar bireye çaprazlama işlemi uygulanır. Problem türüne göre geliştirilen birçok çaprazlama operatörü vardır. Bu çalışmada tek noktalı çaprazlama operatörü kullanılmıştır. Bu operatörde, rasgele bir nokta belirlenir ve seçilen iki birey bu noktadan çaprazlanır. Şekil 7'de çaprazlama işleminin nasıl yapıldığı gösterilmiştir.

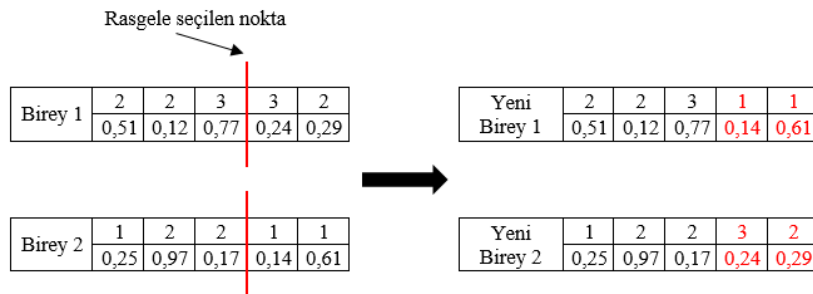
**Mutasyon işlemi:** Popülasyondaki '*toplam gen sayısı × mutasyon olasılığı*' kadar genin değerinin değiştirilmesi işlemidir. Bu işlemin uygulanma amacı, seçim ve çaprazlama işlemlerinden sonra kaybedilen bilginin tekrar elde edilmesidir. Mutasyona uğratabilmek için farklı mutasyon yöntemleri kullanılabilir. Bu çalışmada ise mutasyona uğrayacak gen sayısı kadar rasgele konumlar belirlenir. Seçilen konumdaki işin atandığı makine, atanabileceği bir makine ile değiştirilerek gerçekleştirilir. Aynı zamanda o işin ikinci katmanındaki rassal sayısı yerine de yeniden farklı bir rassal sayı atanmaktadır. Aslında yerel aramada komşu üretmede kullanılan yöntem uygulanmaktadır. Şekil 8'de mutasyon işleminin nasıl yapıldığı gösterilmiştir.

## 5. SAYISAL SONUÇLAR (NUMERICAL RESULTS)

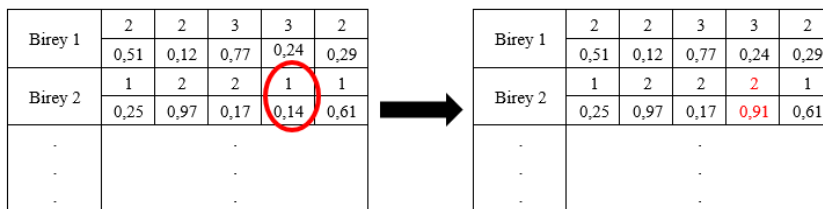
Bu bölümde deneysel çalışmalar iki alt başlıkta sunulmuştur. Öncelikle küçük boyutlu bir örnek probleme ait tüm parametre değerleri verilmiş ve önerilen çözüm yöntemleri ile elde edilen sonuçlar tartışılmıştır. Daha sonra farklı boyutlarda rassal test problemleri türetilmiş ve elde edilen sonuçlar yorumlanmıştır.

### 5.1. Örnek Problem (Sample Problem)

5 iş ve 3 makinenin olduğu küçük boyutlu bir örnek problem türetilmiştir. Örnek problemin işlem süreleri ( $p_{ji}$ ), ilk sıraya



Şekil 7. GA'da çaprazlama işlemi (Crossover process in GA)



Şekil 8. GA'da mutasyon işlemi (Mutation process in GA)



atandığındaki hazırlık süreleri ( $h_{jl}$ ) ve makine uygunluk parametre değerleri ( $b_{jl}$ ) Tablo 2’de ve sıra ve makine bağımlı hazırlık süreleri ( $s_{ijl}$ ) ise Tablo 3’de verilmiştir.

**Tablo 2.** Örnek problemin  $p_{jl}$ ,  $h_{jl}$ ,  $b_{jl}$  parametre değerleri (Parameter values of  $p_{jl}$ ,  $h_{jl}$ ,  $b_{jl}$  of the example problem)

$j$	$l$	$p_{jl}$	$h_{jl}$	$b_{jl}$
1	1	88	30	0
1	2	95	28	1
1	3	30	38	0
2	1	47	88	1
2	2	63	3	1
2	3	32	98	1
3	1	6	30	1
3	2	60	50	1
3	3	61	79	0
4	1	45	96	1
4	2	88	99	1
4	3	74	17	1
5	1	76	63	0
5	2	17	45	1
5	3	54	85	0

Örnek problem, birinci ve ikinci amaç fonksiyonlarına ait ağırlıkların ( $w_1, w_2$ ); (1; 0), (0,95; 0,05), (0,9; 0,1), ..., (0; 1) olmak üzere 21 farklı değeri için GAMS/CPLEX çözücüsü, GA ve YA ile çözülmüştür. Elde edilen çözümler Tablo 4’de verilmiştir.

Tablo 4’den görülebileceği gibi, örnek problem farklı ağırlık değerleri ile çözdürüldüğünde her ikisi de enküçükleme biçiminde olan amaç fonksiyonlarından birisinin değeri artarken diğer amaç fonksiyonunun değerinin azaldığı veya tam tersi bir amaç fonksiyonunun değeri azalırken diğer amaç fonksiyonunun değerinin arttığı görülmektedir. Bu durum amaç fonksiyonlarının birbirini ile ödünleşik olduğunu göstermektedir. Ayrıca çok amaçlı problemlerde problemin ideal ve nadir noktalarının farklı olması amaçların ödünleşik olduğunun bir diğer göstergesidir. Bu örnek problemin  $C_{enb}$  amacının ideal ve nadir değerleri sırasıyla 196 ve 398 ve makine sayısı amacının ise ideal ve nadir değerleri ise sırasıyla 1 ve 3’dür. İdeal nokta (196, 1) nadir noktaya (398,

3) eşit olmadığından amaçların ödünleşik olduğu açık bir şekilde görülmektedir.

Amaç fonksiyonlarının ağırlıkları  $w_1=0,55$  ve  $w_2=0,45$  olarak alındığında GAMS yazılımı, GA ve YA algoritmalarıyla aynı çözüm elde edilmiştir. Bu çözümde, birinci ve ikinci makineler kullanılmak üzere seçilmiş ve son işin tamamlanma zamanı 207 olarak gerçekleşmiştir. İlgili çözümünün Gantt Şeması Şekil 9’da verilmiştir.

### 5.2. Test Sonuçları (Test Results)

Önerilen matematiksel modelin, yerel arama ve genetik algoritma yöntemlerin başarıları rassal türetilen problemler kullanılarak test edilmiştir. Test problemlerinde; işlem süresi ( $p_{jl}$ ), ilk işin hazırlık süresi ( $h_{jl}$ ), sıra ve makine bağımlı hazırlık süresi ( $s_{ijl}$ ) parametre değerleri, literatürdeki diğer çalışmalarda olduğu gibi, [1,100] aralığında düzgün dağılıma uygun olarak türetilmiştir. İşlerin atanabileceği makineleri gösteren  $b_{jl}$  parametresi  $j$ . iş  $l$ . makinede işlem görüyorsa 1 değerini almaktadır. Bu parametrenin değerleri %75 oranında 1, %25 oranında ise 0 olacak şekilde rassal türetilmiştir. Excel VBA kullanılarak 4 farklı test problemi türetilmiştir. Tablo 5’de türetilen test problemlerinin iş ve makine sayıları verilmiştir.

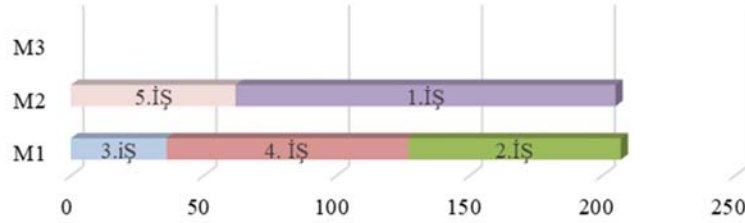
Türetilen test problemleri önerilen matematiksel modelin kodlandığı GAMS programında CPLEX çözücüsü ile çözüm süresi 10800 saniye ile sınırlandırılarak çözülmüştür. Ayrıca test problemleri yerel arama ve genetik algoritmanın kodlandığı PYTHON 3.7 programında çözüm süresi 3600 saniye ile sınırlandırılmış ve 5 kez çözülmüştür. Tablo 6 ve Tablo 7’de sırasıyla önerilen yerel arama algoritması ve genetik algoritmada kullanılan parametre değerleri verilmiştir. Genetik algoritma ve yerel arama algoritmasının en uygun iterasyon sayılarının belirlenebilmesi için her test problemi öncelikle 10000 iterasyon sayısı ile çözülmüş ve kaçınıcı iterasyonda başarılı bir çözüme yakınsandığı tespit edilmiştir. Parametre değerleri belirlenirken literatürde sıklıkla kullanılan değerler dikkate alınmıştır. Problemin amaç fonksiyonunda (Eş. 22) yer alan ağırlıklar ( $w_1, w_2$ ) için (1; 0), (0,95; 0,05), (0,9; 0,1), ..., (0; 1) olmak üzere 21 farklı değer kullanılmıştır.

**Tablo 3.** Örnek problemin  $s_{ijl}$  parametre değerleri (Parameter values of  $s_{ijl}$  of the example problem)

$i$	$j$	$l$	$s_{ijl}$	$i$	$j$	$l$	$s_{ijl}$	$i$	$j$	$l$	$s_{ijl}$	$i$	$j$	$l$	$s_{ijl}$	$i$	$j$	$l$	$s_{ijl}$
1	2	1	61	2	1	1	81	3	1	1	39	4	1	1	24	5	1	1	42
1	2	2	45	2	1	2	96	3	1	2	27	4	1	2	91	5	1	2	48
1	2	3	61	2	1	3	13	3	1	3	91	4	1	3	5	5	1	3	82
1	3	1	20	2	3	1	16	3	2	1	29	4	2	1	33	5	2	1	96
1	3	2	48	2	3	2	6	3	2	2	34	4	2	2	86	5	2	2	31
1	3	3	51	2	3	3	22	3	2	3	78	4	2	3	98	5	2	3	26
1	4	1	32	2	4	1	78	3	4	1	46	4	3	1	52	5	3	1	29
1	4	2	3	2	4	2	28	3	4	2	70	4	3	2	86	5	3	2	41
1	4	3	63	2	4	3	61	3	4	3	74	4	3	3	56	5	3	3	18
1	5	1	88	2	5	1	58	3	5	1	44	4	5	1	57	5	4	1	5
1	5	2	56	2	5	2	1	3	5	2	38	4	5	2	100	5	4	2	95
1	5	3	66	2	5	3	41	3	5	3	22	4	5	3	10	5	4	3	46

**Tablo 4.** Örnek problem için elde edilen çözümler (Obtained solutions for the sample problem)

Ağırlıklar		GAMS/CPLEX			YA			GA		
$w_1$	$w_2$	$f_1$	$f_2$	$z$	$f_1$	$f_2$	$z$	$f_1$	$f_2$	$z$
1	0	196	3	0,61	196	3	0,61	196	3	0,61
0,95	0,05	196	3	0,63	196	3	0,63	196	3	0,63
0,90	0,10	207	2	0,65	207	2	0,65	207	2	0,65
0,85	0,15	207	2	0,65	207	2	0,65	207	2	0,65
0,80	0,20	207	2	0,65	207	2	0,65	207	2	0,65
0,75	0,25	207	2	0,65	207	2	0,65	207	2	0,65
0,70	0,30	207	2	0,65	207	2	0,66	207	2	0,66
0,65	0,35	207	2	0,66	207	2	0,66	207	2	0,66
0,60	0,40	207	2	0,66	207	2	0,66	207	2	0,66
0,55	0,45	207	2	0,66	207	2	0,66	207	2	0,66
0,50	0,50	207	2	0,66	285	2	0,78	207	2	0,66
0,45	0,55	207	2	0,66	227	2	0,69	207	2	0,66
0,40	0,60	207	2	0,66	207	2	0,66	207	2	0,66
0,35	0,65	398	1	0,65	207	2	0,66	207	2	0,66
0,30	0,70	398	1	0,61	207	2	0,66	398	1	0,61
0,25	0,75	398	1	0,56	207	2	0,66	398	1	0,56
0,20	0,80	398	1	0,52	285	2	0,71	398	1	0,52
0,15	0,85	398	1	0,47	207	2	0,66	398	1	0,47
0,10	0,90	398	1	0,42	207	2	0,66	398	1	0,42
0,05	0,95	398	1	0,38	207	2	0,67	398	1	0,38
0	1	531	1	0,33	565	1	0,33	606	1	0,33

**Küçük Boyutlu Örnek Problem için Gantt Şeması****Şekil 9.**  $w_1=0,55$  ve  $w_2=0,45$  için elde edilen çözümünün Gantt Şeması  
(Gantt Chart of Obtained Solution for  $w_1 = 0.55$  and  $w_2 = 0.45$ )**Tablo 5.** Test problemleri (Test problems)

Test Problemi	İş Sayısı	Makine Sayısı
TP-1	10	6
TP-2	20	10
TP-3	40	20
TP-4	100	40

**Tablo 6.** YA parametre değerleri

(Parameter values of local search algorithm)

Parametre	Değeri
Komşu Sayısı	200
İterasyon Sayısı	
TP-1	200
TP-2	800
TP-3	1000
TP-4	2000

**Tablo 7.** GA parametre değerleri

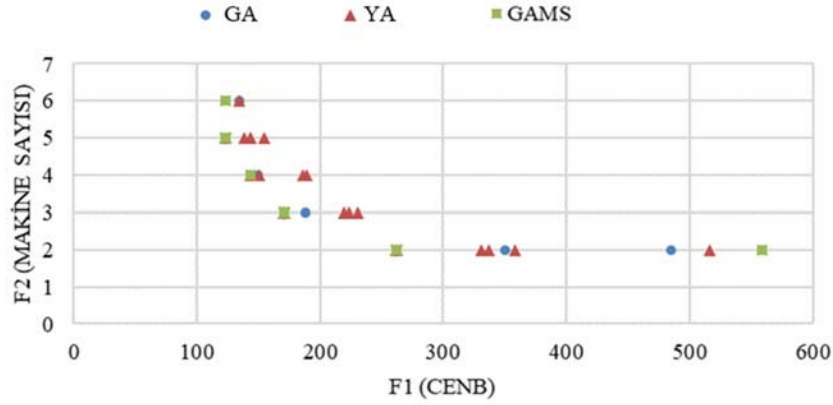
(Parameter values of genetic algorithm)

Parametre	Değeri
Popülasyon Büyüklüğü	200
Çaprazlama Oranı	0.80
Mutasyon Oranı	0.01
Elitizm Sayısı	1
İterasyon Sayısı	
TP-1	200
TP-2	800
TP-3	1000
TP-4	2000

Tablo 8, Tablo 9, Tablo 10 ve Tablo 11'de sırasıyla TP-1, TP-2, TP-3 ve TP-4 problemlerinin önerilen tüm yöntemlerle elde edilen çözümleri, Şekil 10, Şekil 11, Şekil 12 ve Şekil 13'de de elde edilen  $f_1$  ve  $f_2$  amaç fonksiyonu değerleri grafik

**Tablo 8.** TP-1 için elde edilen çözümler (Obtained solutions for TP-1)

Ağırlıklar		GAMS/CPLEX				YA				GA			
$w_1$	$w_2$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$
1	0	123	6	0,38	10,56	135	6	0,42	16,66	135	6	0,42	8,45
0,95	0,05	123	5	0,40	28,98	144	5	0,46	17,00	123	5	0,40	8,51
0,90	0,10	123	5	0,43	34,00	123	5	0,43	17,22	123	5	0,43	7,55
0,85	0,15	123	5	0,45	103,00	139	5	0,49	21,72	123	5	0,45	7,86
0,80	0,20	123	5	0,47	124,20	144	5	0,52	17,86	123	5	0,47	9,04
0,75	0,25	123	5	0,49	189,48	155	5	0,57	17,27	123	5	0,49	8,25
0,70	0,30	144	4	0,51	397,34	144	4	0,51	23,66	144	4	0,51	8,02
0,65	0,35	171	3	0,52	454,89	186	4	0,61	17,19	171	3	0,52	7,80
0,60	0,40	171	3	0,52	335,70	189	4	0,62	17,73	171	3	0,52	8,20
0,55	0,45	171	3	0,52	457,70	171	3	0,52	18,53	150	4	0,55	7,73
0,50	0,50	171	3	0,51	514,19	231	3	0,61	23,14	188	3	0,54	8,30
0,45	0,55	171	3	0,51	512,00	151	4	0,58	17,73	171	3	0,51	8,54
0,40	0,60	171	3	0,51	431,34	171	3	0,51	17,64	171	3	0,51	8,44
0,35	0,65	262	2	0,50	559,63	220	3	0,56	18,28	171	3	0,51	8,49
0,30	0,70	262	2	0,48	548,97	224	3	0,56	17,88	262	2	0,48	8,97
0,25	0,75	262	2	0,45	293,00	262	2	0,45	18,71	262	2	0,45	8,78
0,20	0,80	262	2	0,43	251,14	331	2	0,47	18,27	262	2	0,43	9,59
0,15	0,85	262	2	0,40	399,94	358	2	0,45	18,52	262	2	0,40	8,75
0,10	0,90	262	2	0,38	260,91	337	2	0,40	18,62	262	2	0,38	9,40
0,05	0,95	262	2	0,36	385,39	262	2	0,36	18,00	350	2	0,37	9,23
0	1	559	2	0,33	1,17	516	2	0,33	17,90	485	2	0,33	9,44

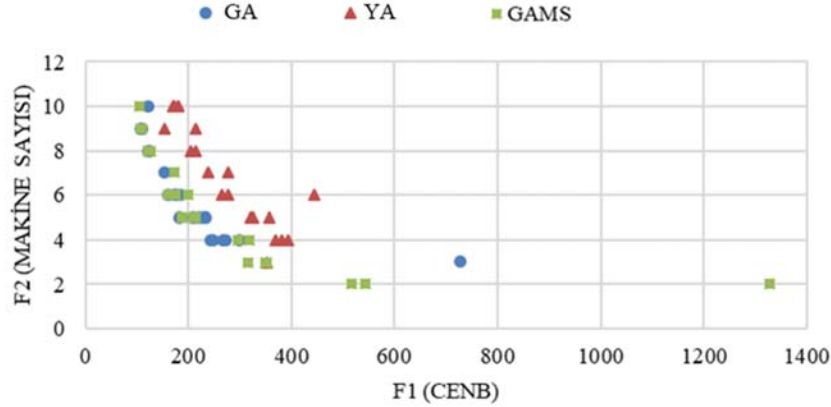
**Şekil 10.** TP-1 çözümlerin grafiği (Graphic of TP-1 Solutions)

üzerinde sunulmuştur. İlgili tablolar, dört bölümden oluşmaktadır. Tablonun en solda yer alan ilk bölümünde ağırlıklar, son üç bölümünde ise sırasıyla GAMS/CPLEX, YA ve GA ile elde edilen  $f_1$  ve  $f_2$  amaç fonksiyonlarının değerleri, skalerleştirilmiş amaç fonksiyonu değeri ve saniye cinsinden çözüm süreleri verilmiştir. YA ve GA için çözüm değerleri 5 tekrar sonucunda bulunan en başarılı değerlerdir. İlgili tablolarda her bir ağırlık seti için, elde edilen skalerleştirilmiş amaç fonksiyonu değerlerinden ( $z$ ) en

başarılısı koyu yazılmıştır. Tablo 8'den de görülebileceği gibi, GAMS/CPLEX tüm ağırlıklar için en başarılı  $z$  değerlerine ulaşmıştır. GA, 21 çözümün 16'sında, YA ise sadece 7'sinde başarılı  $z$  değerleri elde edebilmiştir. Çözüm süreleri açısından ise GA'nın çözümleri 10 sn.'nin altında kalırken, GAMS/CPLEX ise 500 sn.'yi aşan çözüm sürelerine ulaşmıştır. Şekil 10'dan da görülebileceği gibi, GA ve GAMS/CPLEX ile elde edilen çözümler YA'nın çözümlerine baskındır.

**Tablo 9.** TP-2 için elde edilen çözümler (Obtained solutions for TP-2)

Ağırlıklar		GAMS/CPLEX				YA				GA			
$w_1$	$w_2$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$
1	0	104	10	0,26	10801,13	170	10	0,43	395,57	123	10	0,31	63,65
0,95	0,05	108	9	0,31	10800,27	180	10	0,48	390,43	108	9	0,31	64,93
0,90	0,10	110	9	0,34	10800,28	170	10	0,49	380,41	110	9	0,34	66,24
0,85	0,15	128	8	0,40	10801,13	213	9	0,59	389,42	153	7	0,43	67,04
0,80	0,20	123	8	0,41	10800,27	181	10	0,57	389,20	125	8	0,41	77,16
0,75	0,25	123	8	0,43	10801,25	154	9	0,52	455,36	123	8	0,43	64,19
0,70	0,30	176	6	0,49	10800,30	172	10	0,61	379,33	176	6	0,49	67,39
0,65	0,35	200	6	0,54	10800,28	205	8	0,62	408,63	186	6	0,52	72,87
0,60	0,40	174	7	0,54	10800,30	213	8	0,64	393,91	183	5	0,48	82,87
0,55	0,45	162	6	0,50	10800,30	238	7	0,65	396,55	162	6	0,50	70,48
0,50	0,50	188	5	0,49	10800,16	276	7	0,70	389,60	173	6	0,52	73,74
0,45	0,55	209	5	0,51	10800,03	266	6	0,63	406,98	209	5	0,51	76,23
0,40	0,60	296	4	0,54	10800,30	320	5	0,62	454,62	230	5	0,53	74,69
0,35	0,65	213	5	0,51	10800,28	276	6	0,63	393,46	233	5	0,53	72,14
0,30	0,70	318	4	0,52	10800,31	325	5	0,60	452,38	273	4	0,49	71,70
0,25	0,75	351	3	0,45	10800,31	357	5	0,60	383,00	221	5	0,51	78,30
0,20	0,80	315	3	0,40	10800,28	351	3	0,45	455,20	242	4	0,44	83,85
0,15	0,85	349	3	0,39	10800,30	369	4	0,48	402,60	267	4	0,44	72,02
0,10	0,90	544	2	0,32	10803,25	381	4	0,46	375,57	298	4	0,43	78,32
0,05	0,95	516	2	0,26	10810,80	393	4	0,43	380,53	247	4	0,41	76,74
0	1	1327	2	0,20	891,92	445	6	0,60	504,81	728	3	0,30	89,58

**Şekil 11.** TP-2 çözümlerin grafiği (Graphic of TP-2 solutions)

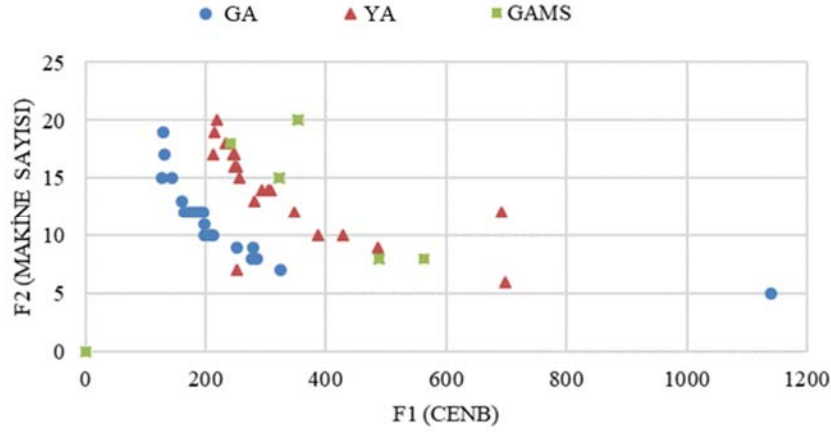
Tablo 9'dan görülebileceği gibi, TP-2 problemi için GAMS/CPLEX son ağırlık seti dışında süre limiti (10800sn.) nedeniyle durmuştur. GAMS/CPLEX, 21 çözümün 17'sinde, GA 11'inde başarılı z değerleri elde edebilmiştir. YA ise hiçbir ağırlık seti için başarılı z değerlerine ulaşamamıştır. YA'nın çözüm süreleri 500 sn. ve GA'nın çözüm süreleri ise 90 sn. ve altında kalmıştır. Şekil 11'den de görülebileceği gibi, GA ve GAMS/CPLEX ile elde edilen

çözümler YA ile elde edilen çözümlerin hemen hepsine baskındır. GA hem çözüm başarısı hem de çözüm süresi yönü ile öne çıkmıştır.

Tablo 10'dan görülebileceği gibi, GAMS/CPLEX süre limiti içinde TP-3 problemine hiçbir ağırlık seti için çözüm üretememiştir. GA, 21 çözümün tamamında başarılı z değerleri elde edebilmiştir. GA, YA'dan daha kısa sürede

**Tablo 10.** TP-3 için elde edilen çözümler (Obtained solutions for TP-3)

Ağırlıklar		GAMS/CPLEX				YA				GA			
$w_1$	$w_2$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$
1	0	-	-	-	-	218	20	0,55	1487,13	130	19	0,33	182,48
0,95	0,05	-	-	-	-	233	18	0,60	1188,74	127	15	0,34	178,72
0,90	0,10	-	-	-	-	213	17	0,57	1970,29	132	17	0,38	171,41
0,85	0,15	-	-	-	-	246	17	0,66	1328,04	145	15	0,42	180,66
0,80	0,20	-	-	-	-	247	16	0,66	1544,72	165	12	0,45	1166,95
0,75	0,25	-	-	-	-	248	17	0,68	1541,99	190	12	0,51	201,63
0,70	0,30	-	-	-	-	214	19	0,66	1210,73	160	13	0,48	183,80
0,65	0,35	-	-	-	-	251	16	0,69	1065,77	198	11	0,52	183,52
0,60	0,40	-	-	-	-	247	17	0,71	1370,46	182	12	0,51	182,25
0,55	0,45	-	-	-	-	280	13	0,68	1406,44	174	12	0,51	181,85
0,50	0,50	-	-	-	-	257	15	0,70	1431,93	196	12	0,55	174,86
0,45	0,55	-	-	-	-	304	14	0,73	1537,86	197	10	0,50	179,98
0,40	0,60	-	-	-	-	307	14	0,73	1438,31	209	10	0,54	177,62
0,35	0,65	-	-	-	-	294	14	0,71	1333,83	213	10	0,51	194,72
0,30	0,70	-	-	-	-	348	12	0,68	1536,36	208	10	0,51	198,11
0,25	0,75	-	-	-	-	386	10	0,62	1599,94	279	9	0,51	188,54
0,20	0,80	-	-	-	-	428	10	0,62	1479,80	252	9	0,49	186,80
0,15	0,85	-	-	-	-	487	9	0,57	1451,31	286	8	0,45	211,79
0,10	0,90	-	-	-	-	697	6	0,45	1552,07	277	8	0,43	206,66
0,05	0,95	-	-	-	-	252	7	0,40	1476,43	325	7	0,37	206,51
0	1	-	-	-	-	692	12	0,60	1120,26	1139	5	0,25	225,46

**Şekil 12.** TP-3 çözümlerin grafiği (Graphic of TP-3 solutions)

çözümü ulaşmıştır ve Şekil 12'den de görülebileceği gibi, GA'nın çözümleri, YA'nın çözümlerine baskındır.

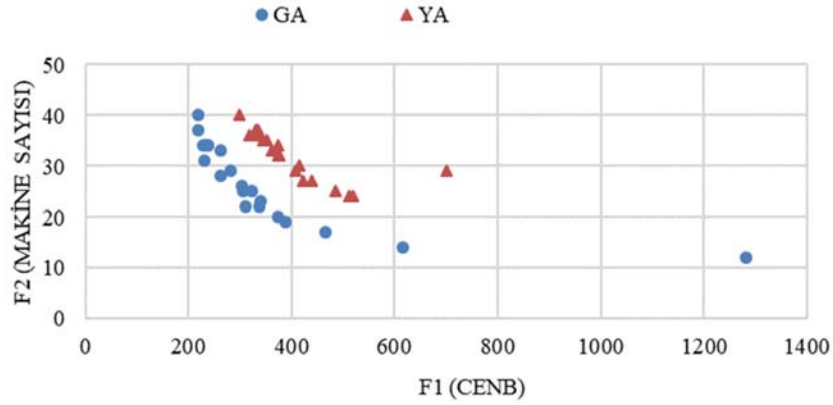
Tablo 11'den görülebileceği gibi, GAMS/CPLEX süre limiti içinde TP-4 problemine hiçbir ağırlık çifti için çözüm üretememiştir. YA ise tüm çözümlerde süre limiti olan 3600 sn.'yi aştığı için durmuştur. GA, 21 çözümün tamamında en

başarılı  $z$  değerlerini elde etmiştir. GA, YA'dan daha kısa sürede çözüme ulaşmıştır. Ayrıca, Şekil 13'den de görülebileceği gibi, GA'nın çözümleri, YA'nın çözümlerine baskındır.

GA'nın YA'dan ne kadar başarılı olduğunu sayısal olarak ifade edebilmek için bu iki algoritma çözüm kalitesi ve süre

**Tablo 11.** TP-4 için elde edilen çözümler (Obtained solutions for TP-4)

Ağırlıklar		GAMS/CPLEX				YA				GA			
$w_1$	$w_2$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$	$f_1$	$f_2$	$z$	$t(s)$
1	0	-	-	-	-	298	40	0,60	3600,25	218	40	0,44	1428,51
0,95	0,05	-	-	-	-	362	33	0,73	3600,08	233	34	0,49	1910,51
0,90	0,10	-	-	-	-	332	37	0,69	3600,08	220	37	0,49	1389,28
0,85	0,15	-	-	-	-	374	34	0,77	3600,07	236	34	0,53	1717,18
0,80	0,20	-	-	-	-	337	36	0,72	3600,05	238	34	0,55	1761,79
0,75	0,25	-	-	-	-	330	36	0,72	3600,01	231	31	0,54	1778,57
0,70	0,30	-	-	-	-	336	37	0,75	3600,10	228	34	0,58	1755,49
0,65	0,35	-	-	-	-	331	37	0,76	3600,08	262	33	0,63	1476,00
0,60	0,40	-	-	-	-	344	35	0,76	3600,01	283	29	0,63	2371,06
0,55	0,45	-	-	-	-	319	36	0,76	3600,15	262	28	0,60	1816,84
0,50	0,50	-	-	-	-	351	35	0,79	3600,11	306	25	0,62	1775,93
0,45	0,55	-	-	-	-	373	32	0,78	3600,09	322	25	0,63	1815,22
0,40	0,60	-	-	-	-	376	32	0,78	3600,10	304	26	0,63	1762,80
0,35	0,65	-	-	-	-	407	29	0,76	3600,04	340	23	0,61	2165,93
0,30	0,70	-	-	-	-	414	30	0,77	3600,10	311	22	0,59	1910,98
0,25	0,75	-	-	-	-	423	27	0,72	3600,01	337	22	0,58	1739,58
0,20	0,80	-	-	-	-	512	24	0,69	3600,08	375	20	0,55	1907,04
0,15	0,85	-	-	-	-	438	27	0,71	3600,10	388	19	0,52	1936,00
0,10	0,90	-	-	-	-	484	25	0,66	3600,01	467	17	0,48	1517,05
0,05	0,95	-	-	-	-	520	24	0,62	3600,12	617	14	0,39	1576,04
0	1	-	-	-	-	700	29	0,72	3600,07	1282	12	0,30	2056,79

**Şekil 13.** TP-4 çözümlerin grafiği (Graphic of TP-4 solutions)

açısından kıyaslanmıştır. Kıyaslamada GAMS/CPLEX'in çözemediği TP-4 probleminin çözümleri dikkate alınmıştır. Her farklı ağırlık çifti için elde edilen çözüm için GA'nın başarı yüzdesi Eş. 23-24'deki formüller kullanılarak hesaplanmıştır. Hesaplanan değerler Tablo 12'de verilmiştir.

$$\text{Çözüm Başarısı}_{(w_1, w_2)} = \frac{z_{YA} - z_{GA}}{z_{YA}} * 100 \quad (23)$$

$$\text{Süre Başarısı}_{(w_1, w_2)} = \frac{t_{YA} - t_{GA}}{t_{YA}} * 100 \quad (24)$$

Tablo 12'den de görülebileceği gibi, GA, YA'dan çözüm kalitesi açısından ortalama %25,64, süre açısından ise ortalama %50,31 daha başarılı olmuştur.

**Tablo 12.** TP-4 probleminde GA'nın YA'dan başarısının sayısal olarak belirlenmesi  
(Numerically determining the success of GA from YA in TP-4 problem)

$w_1$	$w_2$	Çözüm	Süre
		Başarısı $(w_1, w_2)$ %	Başarısı $(w_1, w_2)$ %
1	0	26,67	60,32
0,95	0,05	32,88	46,93
0,90	0,10	28,99	61,41
0,85	0,15	31,17	52,30
0,80	0,20	23,61	51,06
0,75	0,25	25,00	50,60
0,70	0,30	22,67	51,24
0,65	0,35	17,11	59,00
0,60	0,40	17,11	34,14
0,55	0,45	21,05	49,53
0,50	0,50	21,52	50,67
0,45	0,55	19,23	49,58
0,40	0,60	19,23	51,03
0,35	0,65	19,74	39,84
0,30	0,70	23,38	46,92
0,25	0,75	19,44	51,68
0,20	0,80	20,29	47,03
0,15	0,85	26,76	46,22
0,10	0,90	27,27	57,86
0,05	0,95	37,10	56,22
0	1	58,33	42,87

## 6. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu çalışmada, sıra ve makine bağımlı hazırlık sürelerinin ve makine uygunluk kısıtlarının olduğu ilişkisiz paralel makine çizelgeleme probleminde eniyi makine sayısını belirleyen çok amaçlı bir matematiksel model önerilmiştir. Ele alınan amaç fonksiyonları kullanılacak makine sayısının ve son işin tamamlanma zamanının enküçülenmesidir. Amaçların birleştirilmesinde ağırlıklı toplam yöntemi kullanılmıştır.

Literatürde yer alan makine sayısının sabit olduğu çalışmalardan farklı olarak bu çalışmada, kullanılacak makinelerin belirleniyor olması, işletmelere enerji tüketimlerini azaltma, kullanılmayan makinelerin başka işletmelere kiralanabilmesi veya yeni siparişlerin kabul edilebilmesi gibi çok önemli avantajlar sağlamaktadır. Böylelikle üretimlerinde paralel makineler kullanan işletmelere verimliliklerini ve karlılıklarını artırma fırsatı sunulmaktadır.

Önerilen matematiksel modelin performansı rassal türetilen test problemleri kullanılarak gösterilmiştir. Matematiksel model ile çözülemeyecek, büyük boyutlu problemlerin çözümü için bir YA ve bir GA geliştirilmiştir. Türetilen test problemleri, önerilen matematiksel model, YA ve GA ile çözülerek elde edilen çözümler karşılaştırılmıştır. Büyük boyutlu problemler için önerilen matematiksel model ve GAMS/CPLEX ile belirlenen süre limiti içerisinde çözüm elde edilememiştir. Geliştirilen GA ve YA ile elde edilen çözümler karşılaştırıldığında, GA'nın YA'ya göre çözüm

kalitesi açısından ortalama %25,64 ve süre açısından ise ortalama %50,31 daha başarılı olduğu görülmüştür.

Gelecek çalışmalarda, ele alınan amaçlar farklı amaç birleştirme yöntemleri kullanılarak birleştirilebilir. Ayrıca tek amaçlı problemlerin çözümü için önerilmiş klasik YA ve GA yerine çok amaçlı metasezgisel algoritmalar uygulanabilir.

## KAYNAKLAR (REFERENCES)

1. Pinedo, M., Scheduling: Theory, Algorithms and Systems, Springer Science Business Media, New York, 665, 2008.
2. Akyol, E., Saraç, T., Paralel makine çizelgeleme problemi için bir karma tamsayı programlama modeli: ortak kaynak kullanımı, Gazi Üniversitesi Fen Bilimler Dergisi, 5 (3), 109-126, 2017.
3. Pei, Z., Wan, M., Wang, Z., A new approximation algorithm for unrelated parallel machine scheduling with release dates, Annals of Operations Research, 285, 397-425, 2019.
4. Cota, L.P., Coelho, V.N., Guimaraes, F. G., Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times, International Transactions in Operational Research, 28 (2), 996-1017, 2021.
5. Kim, Y.H., Kim, R.S., Insertion of new idle time for unrelated parallel machine scheduling with job splitting and machine breakdowns, Computers & Industrial Engineering, 147, 2020.
6. Lei, D., Yuan, Y., Cai, J., An improved artificial bee colony for multi-objective distributed unrelatedparallel machine scheduling, International Journal of Production Research, 2020.
7. Sarıçipek, İ., Multi-objective scheduling by maximizing machine preferences for unrelated parallel machines, Sigma Journal of Engineering and Natural Sciences-Sigma Muhendislik ve Fen Bilimleri Dergisi, 38 (1), 405-420, 2020.
8. Lei, D.M., Liu, M.Y., An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance, Computers & Industrial Engineering, 141, 2020.
9. Yepes-Borrero, J.C., Villa, F., Perea, F., GRASP Algorithm for the unrelated parallel machine scheduling problem with setup times and additional resources, Expert Systems With Applications, 141, 2020.
10. Cota, L.P., Guimaraes, F.G., Ribeiro, R.G., Meneghini, I.R., de Oliveira, F.D., Souza, M.J. F., Siarry, P., An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem, Swarm and Evolutionary Computation, 51, 2019.
11. Jouhari, H., Lei, D.M., Al-qaness, M.A.A., Abd Elaziz, M., Ewees, A.A., Farouk, O., Sine-Cosine algorithm to enhance simulated annealing for unrelated parallel machine scheduling with setup times, Mathematics, 7 (11), 2019.

12. Ekici, A., Elyasi, M., Ozener, O.O., Sarıkaya, M. B., An application of unrelated parallel machine scheduling with sequence-dependent setups at vestel electronics, *Computers & Operations Research*, 111, 130-140, 2019.
13. Ezugwu, A.E., Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times, *Knowledge-Based Systems*, 172, 15-32, 2019.
14. Bektur, G., Saraç, T., 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, *Computers & Operations Research*, 103, 46-63, 2019.
15. Fanjul-Peyro, L., Ruiz, R., Perea, F., Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times, *Computers & Operations Research*, 101, 173-182, 2019.
16. Ezugwu, A.E., Akutdah, F., An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times, *IEEE Access*, 6, 54459-54478, 2018.
17. Tozzo, E., Cotrim, S.L., Galdamez, E.V.C., Leal, G.C.L., A genetic algorithm and variable neighborhood search for the unrelated parallel machine scheduling problem with sequence dependent setup time, *Acta Scientiarum-Technology*, 40, 2018.
18. Afzalirad, M., Rezaeian, J., A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA II and MOACO approaches, *Applied Soft Computing*, 50, 109-123, 2017.
19. Shahvari, O., Logendran, R., An enhanced tabu search algorithm to minimize a bi- criteria objective in batching and scheduling problems on unrelated parallel machines with desired lower bounds on batch sizes, *Computers & Operations Research*, 77, 154-176, 2017.
20. Afzalirad, M., Rezaeian, J., Resource constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions, *Computers & Industrial Engineering*, 98, 40-52, 2016.
21. Mir, M.S.S., Rezaeian, J., A robust hybrid approach based on particle swarm optimization and genetic algorithm to minimize the total machine load on unrelated parallel machines, *Applied Soft Computing*, 41, 488-504, 2016.
22. Canıylmaz, E., Benli, B., İlkay, M.S., An artificial bee colony algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence dependent setup times and due dates, *International Journal of Advanced Manufacturing Technology*, 77, 2105-2115, 2015.
23. Diana, R.O.M., Filho, M.F., Souza, S.R., Victor, J.F., An immune- inspired algorithm for an unrelated parallel machines' scheduling problem with sequence and machine dependent setup- times for makespan minimisation, *Neurocomputing*, 163, 94-105, 2015.
24. Joo, C.M., Kim, B.S., Hybrid genetic algorithms with dispatching rules for unrelated machine scheduling with setup time and production availability, *Computers & Industrial Engineering*, 85, 102-109, 2015.
25. Rosales, O.A., Bello, F.A., Alvarez, A., Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine dependent setup times, *International Journal of Advanced Manufacturing Technology*, 76, 1705-1718, 2015.
26. Arnout, J.P., Rabadi, G., Musa, R., A two stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence dependent setup times, *Journal of Intelligent Manufacturing*, 21, 693-701, 2010.
27. Eroğlu, D.Y., Özmutlu, H.C., Özmutlu, S., Genetic algorithm with local search for the unrelated parallel machine scheduling problem with sequence dependent set- up times, *International Journal Of Production Research*, 52, 19, 5841-5856, 2014.
28. Kayvanfar, V., Teymourian, E., Hybrid intelligent water drops algorithm to unrelated parallel machines scheduling problem: a just-in-time approach, *International Journal of Production Research*, 52 (19), 5857-5879, 2014.
29. Lee, C.H., Liao, C.J., Chao, C.W., Unrelated parallel machine scheduling with dedicated machines and common deadline, *Computers & Industrial Engineering*, 74, 161-168, 2014.
30. Li, Q., Milne, R.J., A production scheduling problem with sequence dependent changeover cost, *International Journal of Production Research*, 52, 13, 4093-4102, 2014.
31. Lin, Y.K., Hsieh, F.Y., Unrelated parallel machine scheduling with setup times and ready times, *International Journal of Production Research*, 52, 4, 1200-1214, 2014.
32. Lin, S.W., Ying, K.C., ABC- Based manufacturing scheduling for unrelated parallel machines with machine dependent and job sequence dependent setup times, *Computers & Operations Research*, 51, 172-181, 2014.
33. Nogueira, J.P., Arroyo, J.E.C., Villadiego, H.M.M., Gonçalves, L.B., Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties, *Electronic Notes in Theoretical Computer Science*, 302, 53-72, 2014.
34. Rambod, M., Rezaeian, J., Robust meta- heuristic implementation for unrelated parallel machines scheduling problem with rework processes and machine eligibility restrictions, *Computers & Industrial Engineering* 77, 15-28, 2014.
35. Lee, J.H., Yu, J.M., Lee, D.H., A tabu search algorithm for unrelated parallel machine scheduling with sequence and machine dependent setups: minimizing total tardiness, *International Journal of Advanced Manufacturing Technology*, 69, 2081-2089, 2013.
36. Torabi, S.A., Sahebjamnia, N., Mansouri, S.A., Bajestani, M.A., A particle swarm optimization for a fuzzy multi- objective unrelated parallel machine scheduling problem, *Applied Soft Computing*, 13, 4750-4762, 2013.



37. Bozorgirad, M.A., Logendran, R., Sequence dependent group scheduling problem on unrelated parallel machines, *Expert Systems with Applications*, 39, 9021-9030, 2012.
38. Chen, C.L., Iterated hybrid metaheuristic algorithms for unrelated parallel machines problem with unequal ready times and sequence dependent setup times, *International Journal of Advanced Manufacturing Technology*, 60, 693-705, 2012.
39. Coelho, I.M., Haddad, M.N., Ochi, L.S., Farias, R., Souza, M.J.F., A hybrid CPU- GPU local search heuristic for the unrelated parallel machine scheduling problem, 2012 Third Workshop on Applications for Multi-Core Architecture, IEEE, 19-23, 2012.
40. Fleszar, K., Charalambous, C., Hindi, K.S., A variable neighbourhood descent heuristic for the problem of makespan minimization on unrelated parallel machines with setup times, *Journal of Intelligent Manufacturing*, 23, 1949-1958, 2012.
41. Joo, C.M., Kim, B.S., Genetic algorithm with an effective dispatching method for unrelated parallel machine scheduling with sequence dependent and machine dependent setup times, *IE Interfaces*, 25, 3, 357-364, 2012.
42. Wang, I., Wang, Y. C., Chen, C.W., Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics, *Flexible Services and Manufacturing Journal*, 25, 343-366, 2012.
43. Ying, K.C., Lee, Z.J., Lin, S.W., Makespan minimization for scheduling unrelated parallel machines with setup times, *Journal of Intelligent Manufacturing* 23, 1795-1803, 2012.
44. Ying, K.C., Lin, S.W., Unrelated parallel machine scheduling with sequence and machine dependent setup times and due date constraints, *International Journal of Innovative Computing Information and Control*, 8, (5A), 3279-3297, 2012.
45. Chang, P.C., Chen, S.H., Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times, *Applied Soft Computing*, 11, 1263-1274, 2011.
46. Hsu, C.J., Kuo, W.H., Yang, D.L., Unrelated parallel machine scheduling with past sequence dependent setup times and learning effects, *Applied Mathematical Modelling*, 35, 1492-1496, 2011.
47. Kuo, W.H., Hsu, C.J., Yang, D.L., Some unrelated parallel machine scheduling problems with past sequence dependent setup time and learning effects, *Computers & Industrial Engineering*, 61, 179-183, 2011.
48. Lin, S.W., Lu, C.C., Ying, K.C., Minimization of total tardiness on unrelated parallel machines with sequence and machine dependent setup times under due date constraints, *International Journal of Advanced Manufacturing Technology*, 53, 353-361, 2011.
49. Mehravaran, Y., Logendran, R., Bicriteria supply chain scheduling on unrelated parallel machines, *Journal of the Chinese Institute of Industrial Engineers*, 28 (2), 91-101, 2011.
50. Ruiz, R., Andres- Romano, C., Scheduling unrelated parallel machines with resource- assignable sequence-dependent setup times, *International Journal of Advanced Manufacturing Technology*, 57, 777-794, 2011.
51. Vallada, E., Ruben, R., A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times, *European Journal of Operational Research*, 211, 612-622, 2011.
52. Arnout, J.P., Rabadi, G., Musa, R., A two stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence dependent setup times, *Journal of Intelligent Manufacturing*, 21, 693-701, 2010.
53. Chyu, C.C., Chang, W.S., A pareto evolutionary algorithm approach to bi- objective unrelated parallel machine scheduling problems, *International Journal of Advanced Manufacturing Technology*, 49, 697-708, 2010.
54. Dolgui, A., Ereemeev, A.V., Kovalyov, M.Y., Kuznetsou, P.M., Multi-product lot sizing and scheduling on unrelated parallel machines, *II E Transactions*, 42, 514-524, 2010.
55. Paula, M.R., Mateus, G.R., Ravetti, M.G., A non-delayed relax and cut algorithm for scheduling problems with parallel machines, due dates and sequence dependent setup times, *Computers & Operations Research*, 37, 938-949, 2010.
56. Bektur G., Saraç T., Two parallel injection machine scheduling under crane constraint, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31 (4), 903-911, 2016.
57. Furugi A., A tabu search algorithm for the unrelated parallel machine scheduling problem with machine availability constraint and sequence-dependent setup time, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36 (3), 1539-1550, 2021.
58. Talbi, E.G., *Metaheuristics from design to implementation*, John Wiley and Sons, 2009.
59. Çalışkan, F., Yüksel, H., Dayık, M., Genetik algoritmaların tasarım sürecinde kullanılması, *SDU Teknik Bilimler Dergisi*, 6 (2), 21-27, 2016.
60. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, USA, 1989.
61. Deb, K., *Genetic algorithms for function optimization*, *Genetic Algorithms and Soft Computing*, 3-19, Eds. Herrera, F. & Verdegay, J L., Physica - Verlag, Heidelberg, 1996.

