



## A matheuristic algorithm for multi-objective unrelated parallel machine scheduling problem

Tuğba Saraç\*<sup>ID</sup>, Feriştah Özçelik<sup>ID</sup>

Department of Industrial Engineering, Faculty of Engineering and Architecture, Eskişehir Osmangazi University, 26480, Eskişehir, Türkiye

### Highlights:

- Unrelated parallel machine scheduling problem with sequence dependent setup times is considered in a multi-objective manner.
- A matheuristic algorithm is developed for the considered problem.
- Obtained solutions are compared with the results of augmented  $\varepsilon$ -constraint method.

### Keywords:

- Unrelated parallel machine scheduling problem
- Sequence dependent setup times
- Augmented  $\varepsilon$ -constraint method
- Matheuristic algorithm.

### Article Info:

Research Article  
Received:02.02.2021  
Accepted: 27.08.2022

### DOI:

10.17341/gazimmfd.873295

### Correspondence:

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

### Graphical/Tabular Abstract

As shown in Figure A, non-dominated points obtained by the proposed Matheuristic Algorithm (MSA) are close to the lower bound (AS) and better than the points obtained by AEC and upper bound (US). As a result of the tests made, it has been seen that the proposed MSA can solve the problems that the AEC method, which is one of the successful scalarization methods in the literature, cannot solve. In addition, it can obtain more successful solutions in a much shorter time than the AEC method.

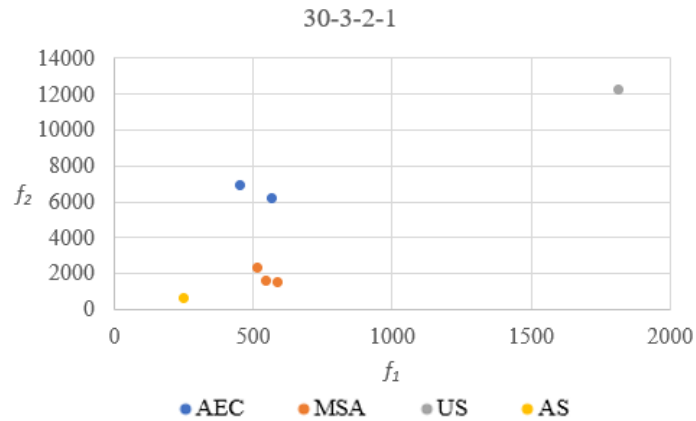


Figure A. Non-dominated Points for 30-3-2-1 Problem

### Purpose:

Machine scheduling problems are one of the basic manufacturing problems. Thus, there are a lot of study in the literature. In most of these studies, the problem is considered as single objective. Although the single objective approach makes it easier to solve problems theoretically, it is often not possible to provide realistic solutions because almost all real-life problems have multi-objective properties. It is aimed to close this gap in the literature by using the multi-objective programming method that emerged as a powerful solution approach.

### Theory and Methods:

The objectives are minimizing the makespan and minimizing the total tardiness. A matheuristic algorithm is developed for solving the considered problem. The performance of the algorithm is compared with the results of the augmented  $\varepsilon$ -constraint method (AEC).

### Results:

The results obtained with the developed algorithm were compared with the solutions of AEC method using randomly generated test problems. Non-dominated points for a sample problem (30-3-2-1) are given in Figure A.

### Conclusion:

With the proposed matheuristic algorithm, both a solution time advantage is obtained and dominant solutions that could not be obtained with the augmented  $\varepsilon$ -constraint method are reached.



## Çok amaçlı ilişkisiz paralel makine çizelgeleme problemi için bir matsezgisel algoritma

Tuğba Saraç\*<sup>ID</sup>, Feriştah Özçelik<sup>ID</sup>

Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, 26480, Eskişehir, Türkiye

### ÖNEÇIKANLAR

- Sıra bağımlı hazırlık sürelerinin olduğu ilişkisiz paralel makine çizelgeleme problemi çok amaçlı olarak ele alınmıştır
- Ele alınan problem için bir matsezgisel algoritma geliştirilmiştir
- Elde edilen sonuçlar genişletilmiş  $\varepsilon$ -kısıt yönteminin sonuçları ile karşılaştırılmıştır

### Makale Bilgileri

Araştırma Makalesi  
Geliş: 02.02.2021  
Kabul: 27.08.2022

### DOI:

10.17341/gazimmfd.873295

### Anahtar Kelimeler:

İlişkisiz paralel makine çizelgeleme problemi, sıra bağımlı hazırlık süreleri, genişletilmiş  $\varepsilon$ -kısıt yöntemi, matsezgisel algoritma

### ÖZ

Makine çizelgeleme problemleri temel üretim problemlerinden birisidir. Bu nedenle literatürde çok sayıda çalışma mevcuttur. Bu çalışmaların önemli bir bölümünde problemin tek amaçlı olarak ele alındığı görülmektedir. Tek amaçlı yaklaşım teorik anlamda problemlerin daha kolay çözülebilmesini sağlasa da gerçek hayat problemlerinin hemen hepsinin çok amaçlı özellik göstermesinden dolayı çoğu zaman gerçekçi çözümler sunamamaktadır. Bu çalışmada, ilişkisiz paralel makine çizelgeleme problemi çok amaçlı olarak ele alınmıştır. Amaçlar son işin tamamlanma zamanının ve toplam gecikmenin enküçüklenmesidir. Ele alınan problemin çözümü için bir matsezgisel algoritma geliştirilmiştir. Geliştirilen algoritma ile elde edilen sonuçlar, genişletilmiş  $\varepsilon$ -kısıt yönteminin sonuçları ile karşılaştırılmıştır. Önerilen matsezgisel algoritma ile hem ciddi bir çözüm süresi avantajı elde edilmiş hem de genişletilmiş  $\varepsilon$ -kısıt yöntemi ile elde edilemeyen baskın çözümlere ulaşılmıştır.

## A matheuristic algorithm for multi-objective unrelated parallel machine scheduling problem

### HIGHLIGHTS

- Unrelated parallel machine scheduling problem with sequence dependent setup times is considered in a multi-objective manner
- A matheuristic algorithm is developed for the considered problem
- Obtained solutions are compared with the results of augmented  $\varepsilon$ -constraint method

### Article Info

Research Article  
Received:02.02.2021  
Accepted: 27.08.2022

### DOI:

10.17341/gazimmfd.873295

### Keywords:

Unrelated parallel machine scheduling problem, sequence dependent setup times, augmented  $\varepsilon$ -constraint method, matheuristic algorithm

### ABSTRACT

Machine scheduling problems are one of the basic manufacturing problems. Thus, there are a lot of study in the literature. In most of these studies, the problem is considered as single objective. Although the single objective approach makes it easier to solve problems theoretically, it is often not possible to provide realistic solutions because almost all real-life problems have multi-objective properties. It is aimed to close this gap in the literature by using the multi-objective programming method that emerged as a powerful solution approach. The objectives are minimizing the makespan and minimizing the total tardiness. A matheuristic algorithm is developed for solving the considered problem. The performance of the algorithm is compared with the results of the augmented  $\varepsilon$ -constraint method. With the proposed matheuristic algorithm, both a solution time advantage is obtained and dominant solutions that could not be obtained with the augmented  $\varepsilon$ -constraint method are reached.

## 1. Giriş (Introduction)

Çizelgeleme problemleri için ilk sistematik çözüm yaklaşımları 1950'lerin ortasında önerilmiştir. O tarihten bu yana literatürde çok sayıda çalışma yapılmıştır. Makine çizelgeleme problemleri ile birçok farklı üretim ortamında karşılaşılmaktadır. Bu nedenle, üretim açısından hala önemli problemlerden birisidir. Çizelgeleme problemleri makine ortamlarına göre tek makine, paralel makine, akış tipi ve atölye tipi çizelgeleme olmak üzere dört ana başlık altında incelenebilir. Paralel makine çizelgeleme problemleri aynı işi yapabilen makinelerin çizelgelemesi problemidir. Son yıllarda her işin her makinadaki işlem sürelerinin farklılık gösterdiği ilişkisiz paralel makine çizelgeleme problemlerine olan ilgi artmıştır. Son on yılda sıralamaya bağlı hazırlık sürelerinin olduğu ilişkisiz paralel makine çizelgeleme alanında yapılmış çalışmalardan erişilebilenler Tablo 1'de verilmiştir. Tablodan, çalışmaların çoğunluğunun tek amaçlı olduğu görülmektedir. Tek amaçlı çalışmalarda en sık ele alınan amaç fonksiyonu  $C_{max}$ 'tır.  $R_m | ST_{sd} | C_{max}$  problem tipini ele alan çalışmalar çözüm yöntemleri açısından incelendiğinde, kesin çözüm yaklaşımlarının [1, 2], sezgisel yaklaşımların [3], metasezgisel yaklaşımların (çok başlangıçlı algoritma [4], karınca kolonisi algoritması [5, 6], genetik algoritma [7], tavlama benzetimi [8-10], yapay bağışıklık sistemi [11], solucan eniyileme algoritması [12], ateşböceği algoritması [13], simbiyotik organizmalar arama [14]), melez çözüm yaklaşımlarının (melez genetik algoritma [15-17], melez yapay arı kolonisi [18], tavlama benzetimi [19]) ve matsezigisel algoritmaların [20] kullanıldığı görülmektedir. Problemik tek amaçlı olarak ele alan diğer çalışmalar incelendiğinde  $C_{max}$  amacını sıra bağımlı hazırlık sürelerinin ( $ST_{sd}$ ) yanı sıra diğer ek kısıtlarla birlikte ele alan [21-24], toplam tamamlanma zamanı [25, 26], ağırlıklı toplam tamamlanma zamanı [27], toplam gecikme [28-31], ağırlıklı toplam gecikme [33-35], ortalama akış zamanı [37], toplam hazırlık maliyeti [38] gibi farklı amaçları ele alanlar mevcuttur.

Birden çok amacı göz önünde bulunduran çalışmalarda ele alınan amaçlar;  $C_{max}$  ve toplam gecikme [39, 40], toplam erkenlik, toplam gecikme ve  $C_{max}$  [41], toplam makine tercihleri ve  $C_{max}$  [42], toplam ağırlıklı akış süresi, toplam ağırlıklı gecikme ve  $C_{max}$  [43], toplam erkenlik ve gecikme süresi [44, 45], toplam ağırlıklı tamamlanma zamanı ve toplam ağırlıklı gecikme [46-49], toplam tamamlanma zamanı ve kaynak miktarı [50], ortalama ağırlıklı akış süresi ve ortalama ağırlıklı gecikme [51], ve kullanılacak makine sayısı ve  $C_{max}$ 'dır [52, 53].

$C_{max}$  amacı, paralel makinelerde son tamamlanan işin tamamlanma zamanını enküçükleyerek aslında işlerin makinalara işlem süresi açısından dengeli dağıtılmasına hizmet etmektedir. Tablo 1'den de görülebileceği gibi çalışmaların pek çoğunda bu amaç fonksiyonu ele alınmıştır. Ancak bu amaç fonksiyonu tek başına ele alındığında bir başka önemli amaç olan toplam gecikmenin enküçüklenmesini garanti edememektedir. Bu nedenle bu çalışmada her iki amaç birlikte ele alınmıştır. Tablo 1'den  $C_{max}$  ve toplam gecikme amaçlarını birlikte ele alan sadece iki çalışma [39, 40] olduğu görülmektedir. Söz konusu bu iki çalışmada da çözüm yöntemi olarak metasezgisel algoritmalar kullanılmıştır. Ele alınan problemin çözümü için ise hem kesin çözüm yöntemlerinin hem de sezgisellerin avantajlarını birleştiren bir matsezigisel algoritma geliştirilmiştir. Çok amaçlı çalışmalardan sadece ikisinde [45, 52] matsezigisel algoritma kullanılmıştır. Saraç ve Tutumlu [52], çalışmalarında işlerin bölünemediği ve bölünebildiği ilişkisiz paralel makine çizelgeleme problemi için iki farklı çok amaçlı matematiksel model önermişlerdir. İşlerin bölünebildiği modelde, işlerin hangi makinalara atanacağı ve sıralarının yanı sıra işlerin kaç bölüneceği ve hangi oranlarda hangi makinelerde işleneceği de belirtilmektedir. Her iki modelin amaçları, son işin tamamlanma zamanının ve kullanılacak makine sayısının enküçüklenmesidir. Çok

amaçlı modeller epsilon kısıt yöntemiyle tek amaçlı yapıya dönüştürülmüştür. Ayrıca işlerin bölünmesine izin verilen matematiksel modelin çözemediği büyük boyutlu problemlerin çözümü için bir matsezigisel algoritma da önerilmiştir. Geliştirilen matsezigisel algoritmada ele alınan problem, üç alt probleme bölünmüş ve çok aşamalı bir çözüm yaklaşımı geliştirilmiştir.

Benzer şekilde, Ekici vd. [45], sıra bağımlı hazırlık sürelerinin dikkate alındığı ve toplam gecikmenin ve erkenliğin enküçüklenmesini amaçladıkları çalışmalarında, ele aldıkları problemin çözümü için ilk aşamada işlerin makinalara atandığı ikinci aşamada ise iş sıralarının belirlendiği çok aşamalı ve yasaklı arama temelli bir matsezigisel algoritma geliştirmişlerdir. Bu çalışmada, Saraç ve Tutumlu [52] ve Ekici vd. [45]'den farklı olarak bütünsel eniyiden uzaklaşma riski yaratabilecek çok aşamalı bir yaklaşım kullanmak yerine problem bütünsel olarak ele alınmıştır. Ayrıca, erişilebilen literatür dikkate alındığında,  $C_{max}$  ve toplam gecikme amaçlarını birlikte ele alan, sıra bağımlı hazırlık süreli, ilişkisiz paralel makine çizelgeleme problemi için ilk defa bu çalışmada matematiksel model tabanlı bir komşu arama algoritması (matsezigisel algoritma) önerilmiştir. Ayrıca, geliştirilen matsezigisel algoritmanın başarısını test etmek için rassal olarak türetilen test problemleri kullanılmıştır. Test problemleri, karşılaştırma için genişletilmiş  $\epsilon$ -kısıt yöntemi ile de çözülmüştür. Genişletilmiş  $\epsilon$ -kısıt yöntemi, zayıf pareto çözümleri önleyebilen bir yaklaşımdır. Bu yöntemde  $\epsilon$ -kısıtlar aylak/artık değişken eklenerek eşitlik kısıtına dönüştürülmektedir. Bu değişkenlerin amaç fonksiyonuna eklenmesiyle tüm amaçların dikkate alınabilmesi sağlandığından pareto çözümlerin elde edilebilmesi mümkün olabilmektedir.

İzleyen bölümde ele alınan problemin tanımı ve matematiksel modeli verilmiştir. Üçüncü bölümde, geliştirilen matsezigisel algoritma ayrıntılı bir şekilde açıklanmıştır. Dördüncü bölümde, elde edilen matsezigisel sonuçlarını karşılaştırmada kullanılacak olan genişletilmiş  $\epsilon$ -kısıt yöntemi sunulmuştur. Beşinci bölümde deneysel sonuçlar ve son bölümde ise sonuç ve öneriler tartışılmıştır.

## 2. Problem Tanımı ve Matematiksel Model (Problem Definition and Mathematical Model)

Ele alınan problemde,  $n$  iş,  $m$  paralel makineden birinde işlem görmelidir.  $j$  işinin her  $l$  makinesindeki işlem süresi ( $p_{jl}$ ) farklıdır. İşlem öncesinde bir hazırlık süresi gerekmektedir ve bu hazırlık süreleri iş sırasına ve işin atandığı makineye göre değişebilmektedir. İşlerin bölünmesine izin verilmemektedir ve işler sıfır zamanında hazırdır. Her makine aynı anda sadece bir iş gerçekleştirebilmektedir. Amaç fonksiyonları son işin tamamlanma zamanının ve toplam gecikmenin enküçüklenmesidir.

Problemin matematiksel modeli (UPM), bu modele ait kümeler, indisler, parametreler, karar değişkenleri, amaç fonksiyonları ve kısıtlar aşağıda verilmiştir;

*Kümeler ve İndisler:*

$N$  : İş kümesi,  $N = \{1, 2, \dots, n\}$   
 $M$  : Makine kümesi,  $M = \{1, 2, \dots, m\}$   
 $i, j \in N$  : iş indisi  
 $k \in N$  : sıra indisi  
 $l \in M$  : makine indisi

*Parametreler:*

$n$  : iş sayısı  
 $m$  : makine sayısı

**Tablo 1.** Sıralamaya bağlı hazırlık sürelerinin olduğu ilişkisiz paralel makine çizelgeleme problemleri (Unrelated parallel machine scheduling problem with sequence dependent setup times)

Yayın	Problem	Çözüm yöntemi
Gedik vd. [1]	$R_m   ST_{sd}   C_{max}$	Kısıt programlama
Fanjul-Peyro vd. [2]	$R_m   ST_{sd}   C_{max}$	Matematiksel model
Coelho vd. [3]	$R_m   ST_{sd}   C_{max}$	LA tabanlı sezgisel
Rosales vd. [4]	$R_m   ST_{sd}   C_{max}$	MSA
Arnout vd. [5]	$R_m   ST_{sd}   C_{max}$	KKA
Arnout vd. [6]	$R_m   ST_{sd}   C_{max}$	KKA
Chang ve Chen [7]	$R_m   ST_{sd}   C_{max}$	GA
Ezugwu [8]	$R_m   ST_{sd}   C_{max}$	TB
Santos vd. [9]	$R_m   ST_{sd}   C_{max}$	TB
Ying vd. [10]	$R_m   ST_{sd}   C_{max}$	TB
Diana vd. [11]	$R_m   ST_{sd}   C_{max}$	YBS
Arnout [12]	$R_m   ST_{sd}   C_{max}$	SEA
Ezugwu ve Akutsah [13]	$R_m   ST_{sd}   C_{max}$	AA
Ezugwu vd. [14]	$R_m   ST_{sd}   C_{max}$	SOA
Abreu ve Prata [15]	$R_m   ST_{sd}   C_{max}$	Melez GA
Vallada ve Ruiz [16]	$R_m   ST_{sd}   C_{max}$	Melez GA
Eroğlu vd. [17]	$R_m   ST_{sd}   C_{max}$	Melez LA ve GA
Lin ve Ying [18]	$R_m   ST_{sd}   C_{max}$	Melez YAK algoritması
Jouhari vd. [19]	$R_m   ST_{sd}   C_{max}$	Melez TB ve sinus kosinüs algoritması
Fleszar vd. [20]	$R_m   ST_{sd}   C_{max}$	Matsezgisel
Afzalirad ve Rezaeian [21]	$R_m   ST_{sd}, r_j, M_j, prec, res   C_{max}$	GA ve YBS
Mir ve Rezaeian [22]	$R_m   ST_{sd}, r_j   \sum C_{max}$	Melez PSO ve GA
Rambod ve Rezaeian [23]	$R_m   ST_{sd}, M_j   C_{max}$	GA ve YAK temelli sezgiseller
Yepes-Borrero vd. [24]	$R_m   ST_{sd}, S_l   C_{max}$	GRASP algoritması
Hsu vd. [25]	$R_m   ST_{sd}   \sum C_j$	Polinom zamanlı alg.
Lee vd. [26]	$R_m   ST_{sd}   \sum C_j$	Kurucu sezgisel
Rauchecker ve Schryen [27]	$R_m   ST_{sd}, M_j   \sum w_j C_j$	Dal-fiyat algoritması
Lee vd. [28]	$R_m   ST_{sd}   \sum T_j$	TA, TB
Lin vd. [29]	$R_m   ST_{sd}   \sum T_j$	İteratif aç gözlü algoritma
Ying ve Lin [30]	$R_m   ST_{sd}   \sum T_j$	YAK
Joo ve Kim [31]	$R_m   ST_{sd}, r_j   \sum T_j$	Sevk etme kurallı GA
Joo ve Kim [32]	$R_m   ST_{sd}, brkdown   \sum T_j$	Sevk etme kurallı GA
Paula vd. [33]	$R_m   ST_{sd}   \sum w_j T_j$	Lagrange gevşetmesi temelli gevşetme ve kesme algoritması
Bektur ve Saraç [34]	$R_m   ST_{sd}, S_l, M_j   \sum w_j T_j$	TA ve TB
Lin ve Hsieh [35]	$R_m   ST_{sd}, r_j   \sum w_j T_j$	Melez sezgisel
Chen [36]	$R_m   ST_{sd}, r_j   \sum w_j U_j$	Sezgisel
Jeong vd. [37]	$R_m   ST_{sd}   ortalama akış zamanı$	Sezgisel
Li ve Milne [38]	$R_m   ST_{sd}   toplam hazırlık maliyeti$	Üç aşamalı bir sezgisel
Canıyılmaz vd. [39]	$R_m   ST_{sd}, M_j   C_{max} + \sum T_j$	GA, YAK
Wang ve Pan [40]	$R_m   ST_{sd}, PM   C_{max}, \sum T_j$	Emperyalist Rekabetçi Algoritma
Kayvanfar ve Teymourian [41]	$R_m   ST_{sd}   \sum (E_j + T_j) + C_{max}$	Zeki su damlacıkları algoritması
Sarıçiçek [42]	$R_m   ST_{sd}, M_j   C_{max}, toplam makine tercihleri$	Çok amaçlı TB
Torabi vd. [43]	$R_m   ST_{sd}, r_j, res   \sum w_j F_j, \sum w_j T_j, \sum C_{max}$	Çok amaçlı PSO
Nogueira vd. [44]	$R_m   ST_{sd}   \sum (E_j + T_j)$	GRASP, LA
Ekici vd. [45]	$R_m   ST_{sd}, r_j   \sum (E_j + T_j)$	TA, matsezgisel
Chyu ve Chang [46]	$R_m   ST_{sd}   \sum w_j C_j, \sum w_j T_j$	Pareto evrimsel algoritma, çok amaçlı TB
Bozorgirad ve Logendran [47]	$R_m   ST_{sd}, r_j, M_j   \sum w_j C_j + w_j T_j$	TA temelli sezgiseller
Mehravaran ve Logendran [48]	$R_m   ST_{sd}, brkdown, r_j   \sum w_j C_{ij} + \sum w_j T_{ij}$	TA temelli sezgisel
Shahvari ve Logendran [49]	$R_m   ST_{sd}, brkdown, r_j, M_j   \sum w_j C_j + \sum w_j T_j$	LA temelli sezgisel
Ruiz ve Romano [50]	$R_m   ST_{sd}   \sum C_j + kaynak miktarı$	Sevk etme kurallarının kullanıldığı sezgiseller
Afzalirad ve Rezaeian [51]	$R_m   ST_{sd}, r_j, M_j, prec   w f_j (C_j - r_j), w t_j T_j$	NSGA- II, çok amaçlı KKA
Saraç ve Tutumlu [52]	$R_m   ST_{sd}, prmp, M_j   C_{max}, makine sayısı$	Matsezgisel
Saraç ve Tutumlu [53]	$R_m   ST_{sd}, M_j   C_{max}, makine sayısı$	GA, LA

GA: Genetik Algoritma, KKA: Karınca Kolonisi Algoritması, YBS: Yapay Bağışıklık Sistemi, MSA: Çok Başlangıçlı Algoritma, TB: Tavlama Benzetimi, YAK: Yapay Arı Kolonisi, YSA: Yapay Sinir Ağları, LA: Yerel Arama, TA: Tabu Arama, SEA: Solucan Eniyileme Algoritması, AA: Ateşböceği algoritması, SOA: Simbiyotik Organizmalar Arama

$p_{jl}$  :  $j$ . işin  $l$ . makinedeki işlem süresi  
 $h_{jl}$  :  $j$ . işin  $l$ . makinede ilk sıraya atanması durumunda hazırlık süresi  
 $s_{ijl}$  :  $i$  işi  $j$  işinden önce  $l$ . makinede çizelgelenmişse, sıra ve makine bağımlı hazırlık süresi  
 $d_j$  :  $j$ . işin teslim zamanı  
 $\beta$  : 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  
 $C_{max}$  : son işin tamamlanma zamanı  
 $T_j$  :  $j$  işinin gecikme süresi  $T_j = enb\{C_j - d_j, 0\}$

(UPM):

**Amaç fonksiyonları:**

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

$$enk f_2 = \sum_j T_j \quad (2)$$

**Kısıtlar:**

$$\sum_j x_{jkl} \leq 1 \quad \forall k, l \quad (3)$$

$$\sum_k \sum_l x_{jkl} = 1 \quad \forall j \quad (4)$$

$$\sum_j x_{jkl} - \sum_i x_{i(k-1)l} \leq 0 \quad \forall k, l \quad k > 1 \quad (5)$$

$$C_j + \beta(1 - x_{j1l}) \geq h_{jl} + p_{jl} \quad \forall j, l \quad (6)$$

$$C_j - \beta(1 - x_{j1l}) \leq h_{jl} + p_{jl} \quad \forall j, l \quad (7)$$

$$C_j + \beta(2 - x_{jkl} - x_{i(k-1)l}) \geq C_i + p_{jl} + s_{ijl} \quad \forall i, j, k, l \quad i \neq j, k > 1 \quad (8)$$

$$C_j - \beta(2 - x_{jkl} - x_{i(k-1)l}) \leq C_i + p_{jl} + s_{ijl} \quad \forall i, j, k, l \quad i \neq j, k > 1 \quad (9)$$

$$C_{max} \geq C_j \quad \forall j \quad (10)$$

$$T_j \geq C_j - d_j \quad \forall j \quad (11)$$

$$x_{jkl} \in \{0,1\} \quad \forall j, k, l \quad (12)$$

$$C_j, T_j \geq 0 \quad \forall j \quad (13)$$

$$C_{max} \geq 0 \quad (14)$$

İlk amaç (Eş. 1) son işin tamamlanma zamanının ve ikinci amaç (Eş. 2) toplam gecikmenin enküçüklenmesidir. Eş. 3 ve Eş. 4'te verilen kısıtlar sırasıyla bir makinanın bir sırasına en fazla bir işin atanabilmesini ve her işin mutlaka bir makinanın bir sırasına atanmasını sağlamaktadır. Eş. 5'te verilen kısıt, işlerin sıra atlamadan atanmasını sağlamaktadır. Eş. 6 ve Eş. 7'de verilen kısıtlar her makinedeki ilk işlerin ve Eş. 8 ve Eş. 9'da verilen kısıtlar ilk sıra haricindeki işlerin tamamlanma zamanını belirlemektedir. Eş. 10'da verilen kısıt, son işin tamamlanma zamanını, Eş. 11'de verilen kısıt ise işlerin gecikme sürelerini hesaplatma kısıtlarıdır. Karar değişkenlerinin işaret kısıtları, Eş. 12, Eş. 13 ve Eş. 14'te verilmiştir.

### 3. Geliştirilen Matsezgisel Algoritma (Developed Matheuristic Algorithm)

Matsezgisel algoritmalar, kesin çözüm yöntemleri ile sezgisel algoritmaların güçlü yönlerini birleştiren çözüm yaklaşımlarıdır. Bu çalışmada ele alınan problemin başarılı çözümlerini daha kısa sürede elde edebilmek için bir matsezgisel algoritma geliştirilmiştir. Geliştirilen algoritma (MSA) iki aşamadan oluşmaktadır. İlk aşamada matematiksel model tabanlı kurucu sezgisel kullanılarak uygun bir çözüm oluşturulmaktadır. Sonraki aşamada ise, eldeki çözüm matematiksel model kullanan bir komşu arama yöntemi ile iyileştirilmektedir. Bu çalışmada kullanılan matematiksel model tabanlı komşu arama yönteminde bir olurlu çözümden başlanır. Bu çözümde rasgele bazı işlerin atamaları kaldırılır diğerleri sabit tutularak kaldırılan işler bir matematiksel model yardımı ile amaç fonksiyonunu iyileştirecek şekilde yeniden atanır. Böylece yeni bir komşu çözüm elde edilmiş olur. Önerilen matsezgisel algoritmanın adımları Tablo 2'de verilmiştir. Bu çalışmada durdurma kriteri; 50 iterasyon, kötü çözümü kabul olasılığı (KO); 0,1 ve karar değişkenlerini serbest bırakma oranının ( $\alpha$ ) başlangıç değeri; 0,2 olarak alınmıştır. Serbest bırakma oranı  $\alpha$ , çözümdeki iyileşmeye bağlı olarak [0,2; 0,5] aralığında kalmak koşulu ile artırılıp azaltılmasına izin verilecek şekilde dinamik olarak tasarlanmıştır. İyileşme olmadığında  $\alpha$  artırılarak daha farklı çözümlerin üretilmesine olanak yaratılmaktadır. İyileşme devam ettiğinde ise  $\alpha$  azaltılarak iyileşmenin elde edildiği çözüme benzer komşuların üretilmesi sağlanmaktadır.

Matsezgisel algoritmada son işin tamamlanma zamanı (Eş. 1) ve toplam gecikme (Eş. 2) amaçlarının birleştirilmesinde ağırlıklı toplam yöntemi kullanılmıştır. Birleştirilmiş amaç fonksiyonu Eş. 15'te verilmiştir. Amaç fonksiyonlarını aynı ölçüğe getirmek amacıyla, amaç fonksiyonları üst sınır değerlerine bölünmüştür.  $C_{max}$  için bir üst sınır değeri olan  $\theta$ , Eş. 16 kullanılarak hesaplanmıştır.  $w_1$  birinci amaç fonksiyonunun,  $w_2$  ise ikinci amaç fonksiyonunun ağırlığıdır.

$$enk z_{MSA} = w_1 \frac{C_{max}}{\theta/(m-1)} + w_2 \frac{\sum_j T_j}{\sum_j d_j} \quad (15)$$

$$\theta = \sum_j enb p_{jl} + \sum_j enb h_{jl} + \sum_{i,l} enb s_{ijl} \quad (16)$$

**Tablo 2.** Matsezgisel algoritmanın adımları (Steps of matheuristic algorithm)

<b>Adım 1:</b> Matematiksel model (UPM-KUR) tabanlı kurucu sezgiseli kullanarak uygun bir başlangıç çözüm bul, mevcut çözüm olarak ata.
<b>Adım 2:</b> Karar değişkenlerinden ( $x_{jkl}$ ) mevcut çözümde '1' değerini almış olanların içinden rassal olarak belirlenen $\alpha$ kadarını serbest bırak (değerlerini 0 yap).
<b>Adım 3:</b> (UPM-İYİLEŞTİR) modelini çöz. (UPM-İYİLEŞTİR) modeli ile elde edilen çözüm mevcut çözümden daha iyi ise mevcut çözümü ve eniyi çözümü güncelle, daha kötü ise KO olasılığı ile kötü çözümü mevcut çözüm olarak kabul et.
<b>Adım 4:</b> Durdurma koşulu sağlandıysa DUR. Sağlanmadı ise Adım 2'ye git.

Başlangıç çözümünün oluşturulmasında kullanılan kurucu sezgiselde kullanılan matematiksel model (UPM-KUR) aşağıda verilmiştir.

(UPM-KUR):

Amaç fonksiyonu:

$$enk_{Z_{KUR}} = w_1 \frac{\tilde{C}_{max}}{\theta/(m-1)} + w_2 \sum_j \sum_k \sum_l x_{jkl} \frac{1}{k} \frac{d_j}{d_{max}} \quad (17)$$

Kısıtlar:

(3)-(5), (12), (14)

$$\tilde{C}_{max} \geq \sum_j \gamma_{jl} x_{j1l} + \sum_i \sum_j \sum_{k>1} \varphi_{jl} x_{jkl} \quad \forall l \quad (18)$$

Kurucu sezgiselde kısa süre içinde atama kısıtlarını sağlayan uygun bir çözüm bulunması hedeflendiğinden (UPM) modeli sadece atama kısıtlarını içerecek şekilde sadeleştirilmiştir. Bu nedenle tamamlanma zamanı ve gecikmeler hesaplanmamıştır. (UPM-KUR) modelinde  $C_{max}$  için yaklaşık bir değer ( $\tilde{C}_{max}$ ) kısıt (18) ile hesaplanmıştır. Burada ilk sıraya atanan işlerin tamamlanma zamanlarını temsilen  $\gamma_{jl}$ , ikinci ve sonraki işlerin tamamlanma zamanlarını temsilen  $\varphi_{jl}$  kullanılmıştır.  $\gamma_{jl}$  ve  $\varphi_{jl}$  sırasıyla Eş. 19 ve Eş. 20 kullanılarak hesaplanmıştır. Amaç fonksiyonunda (Eş. 17)  $C_{max}$  değeri yerine  $\tilde{C}_{max}$  değeri kullanılmıştır. Benzer şekilde gecikme değerleri de hesaplanmadığından teslim zamanı önce olan işlerin ilk sıralara atanmasını sağlayan  $\sum_j \sum_k \sum_l x_{jkl} \frac{1}{k} \frac{d_j}{d_{max}}$  matematiksel ifadesi toplam gecikmenin enküçüklenmesi amacı yerine kullanılmıştır. Toplam gecikmeyi temsil eden amacın normalleştirilmesi için kullanılan  $d_{max}$  değeri Eş. 21 kullanılarak hesaplanmıştır.

$$\gamma_{jl} = p_{jl} + h_{jl} \quad (19)$$

$$\varphi_{jl} = p_{jl} + \text{enb}_i s_{ijl} \quad (20)$$

$$d_{max} = \text{enb}_j d_j \quad (21)$$

Matematiksel model (UPM-KUR) tabanlı kurucu sezgisel algoritmanın adımları Tablo 3’de verilmiştir.

Mevcut çözümün iyileştirilmesinde kullanılan sezgisel ait matematiksel model (UPM-İYİLEŞTİR) aşağıda verilmiştir.

(UPM-İYİLEŞTİR):

Amaç Fonksiyonu:

(15)

Kısıtlar:

(3)-(14)

$$x_{jkl} \geq \alpha x_{jkl} \quad \forall j, k, l \quad (22)$$

Eş. 22’de verilen kısıt, mevcut çözümde ‘1’ değerini almış karar değişkenlerinin değerlerini korumasını sağlamaktadır.

#### 4. Genişletilmiş $\epsilon$ -Kısıt Yöntemi (Augmented $\epsilon$ -Constraint Method)

$f(x)$ , tüm amaçları enküçükleme şeklinde olan çok amaçlı bir matematiksel modelin  $x$  çözümüne ( $x \in X$ ) karşı gelen amaç vektörü iken;  $f(x) \leq f(x^*)$  şartını sağlayan bir  $x$  çözümü bulunamıyor ise,  $x^*$ , etkin çözüm (Pareto-eniyi çözüm),  $f(x^*)$  ise baskın nokta olarak isimlendirilir.  $f(x) < f(x^*)$  şartını sağlayan bir  $x$  çözümü bulunamıyor ise,  $x^*$ , zayıf etkin çözüm,  $f(x^*)$  ise zayıf baskın nokta olarak isimlendirilir. Problemin  $p$  adet amacı var iken, ideal noktası  $y^I = (y_1^I, \dots, y_p^I)$  Eş. 23’de verildiği şekilde hesaplanır.

$$y_k^I = \text{enk}_{x \in X} f_k(x) \quad k = 1, \dots, p \quad (23)$$

Problemin nadir noktası  $y^N = (y_1^N, \dots, y_p^N)$  ise Eş. 24’te verildiği gibi hesaplanır.

$$y_k^N = \text{enb}_{x \in X_E} f_k(x) \quad k = 1, \dots, p \quad (24)$$

Burada  $X_E$  etkin çözüm kümesini göstermektedir.

$\epsilon$ -kısıt yöntemi, çok amaçlı problemlerin amaç fonksiyonlarından birisinin seçilerek tek bir amaç olarak belirlendiği diğer amaç fonksiyonlarının ise  $\epsilon$ -eşitsizlik kısıtları olarak modele eklendiği bir skalerleştirme yöntemidir. Genişletilmiş  $\epsilon$ -kısıt (Augmented  $\epsilon$ -constraint) yöntemi (AEC), klasik  $\epsilon$ -kısıt yönteminin zayıf pareto çözümler üretmesini önlemek amacıyla 2009 yılında Mavrotas [54] tarafından geliştirilmiştir. Bu yöntemin klasik yöntemden iki önemli farkı, ödünleşme tablosunun oluşturulması sırasında sözlüksel

**Tablo 3:** Matematiksel model (UPM-KUR) tabanlı kurucu sezgisel algoritmanın adımları

---

Adım 1: Uygun bir başlangıç çözüm bulunması.  
(UPM-KUR) modelini çöz. Çözüm sonucu elde edilen karar değişkeni değerlerini  $\alpha x_{jkl}$  parametrelerine ata.  
 $\alpha x_{jkl} = x_{jkl}$ ,

Adım 2: Uygun çözümün amaç fonksiyonu değerinin (15) hesaplanması.

Adım 2.1:  $j$ . işin sırasını ( $sira_j$ ) ve ilk sıradaki işin tamamlanma zamanını ( $CC_{j|sira_j=1}$ ) hesapla.

$$sira_j = \sum_k \sum_l k \alpha x_{jkl} \quad \text{ve} \quad CC_{j|sira_j=1} = \sum_l \sum_{k|k=1, \alpha x_{jkl}=1} h_{jl} + p_{jl}$$

Adım 2.2: İkinci ve sonraki sıralarda yer alan işlerin tamamlanma zamanlarını hesapla.  
For  $k=1$  to  $n$   
For  $j=1$  to  $n$   
If  $sira_j = k$  and  $k>1$  then  $CC_j = \sum_i \sum_{l|i \neq j, \alpha x_{jkl}=1, \alpha x_{i(k-1)l}=1} (CC_i + s_{ijl} + p_{jl})$   
Next  $j$   
Next  $k$

Adım 2.3.  $j$ . işin tamamlanma zamanını ( $TT_j$ ) ve amaç fonksiyonu değerini (15) hesapla.  
 $TT_j = \max(CC_j - d_j, 0)$

---

sıralama yaklaşımının kullanılması ve  $\varepsilon$ -kısıtların aylak/artık değişken eklenerek eşitlik kısıtına dönüştürülmesidir. Bu değişkenlerin amaç fonksiyonuna eklenmesiyle tüm amaçların dikkate alınabilmesi sağlandığından pareto çözümlerin elde edilebilmesi mümkün olabilmektedir. Ele alınan problemin genişletilmiş  $\varepsilon$ -kısıt yöntemi ile skalerleştirilmiş modeli (UPM-AEC) ve bu modelde kullanılan ek parametre ve karar değişkenleri aşağıda verilmiştir.

*Parametreler:*

$\varepsilon$  : Toplam gecikme amacına ait  $\varepsilon$ -kısıtın sağ taraf sabiti.  
 $\mu$  : Çok küçük pozitif sayı (Bu çalışmada  $10^{-6}$  olarak alınmıştır).  
 $f_1^I, f_2^I$  : İlk ve ikinci amaçların ideal değerleri  
 $f_1^N, f_2^N$  : İlk ve ikinci amaçların nadir değerleri

*Karar Değişkeni:*

$a$  : Toplam gecikme amacına ait  $\varepsilon$ -kısıta eklenen aylak değişken.

(UPM-AEC):

*Amaç fonksiyonu:*

$$f_{AEC} = \text{enk}(C_{\max} - \frac{\mu a}{f_2^N - f_2^I}) \quad (25)$$

*Kısıtlar:*

$$(3) - (14)$$

$$\sum_j T_j + a = \varepsilon \quad (26)$$

$$a \geq 0 \quad (27)$$

Ele alınan problemin genişletilmiş  $\varepsilon$ -kısıt yöntemi ile çözülebilmesi için izlenen adımlar Tablo 4'de verilmiştir.

## 5. Deneysel Sonuçlar (Computational Results)

Geliştirilen matsezigisel algoritma ve kıyaslamada kullanılacak AEC ile, öncelikle oyuncak problem daha sonra da rassal olarak türetilen test problemleri çözülmüş ve elde edilen sonuçlar ayrı alt başlıklarda tartışılmıştır. Tüm testler Intel (R) Core (TM) i7- 5700HQ CPU@2.70 GH işlemcisi, 8 GB belleği ve Windows 10 işletim sistemine sahip bir bilgisayarda 7200 saniye süre limiti ile yapılmıştır. Hem MSA hem de AEC, GAMS 24.1.3 ile kodlanmış ve çözücü olarak Cplex kullanılmıştır.

### 5.1. Test Problemlerinin Türetilmesi (Generation of Test Problems)

Test problemlerinde kullanılan teslim tarihi dışındaki parametrelerin değerleri veya türetilmelerinde kullanılan dağılımlar Tablo 5'de

verilmiştir. Teslim tarihi ( $d_j$ ) ise Logendran vd. [55] önerdiğine benzer şekilde aşağıdaki gibi türetilmiştir. Burada KDD kesikli düzgün dağılıma karşı gelmektedir.  $b_1$  ve  $b_2$  teslim zamanlarının sıklığını kontrol etmekte kullanılan parametrelerdir.  $b_2$  değeri küçüldükçe teslim zamanı değeri de sıklaşmaktadır.

$$d_j \sim KDD \left[ \frac{\sum_l \sum_i p_{jl}}{m} \times \left( 1 - b_1 - \frac{b_2}{2} \right), \frac{\sum_l \sum_i p_{jl}}{m} \times \left( 1 - b_1 + \frac{b_2}{2} \right) \right] \quad (28)$$

**Tablo 5.** Parametrelerin değerleri/dağılımları (Parameter values/distributions)

Parametre	Dağılım/ Değer
$n$	10, 20, 30, 40 ve 50
$m$	2 ve 3
$p_{jl}$	$U(1,100)$
$h_{jl}$	$U(1,100)$
$s_{ijl}$	$U(1,100)$
$b_1, b_2$	$b_1 = 0,8$ ve $b_2 = 0,4$ ve $0,8$

Test problemleri, sırasıyla iş-makina sayısı, teslim zamanı sıklık parametresi ( $b_1 = 0,8$  ve  $b_2 = 0,4$  ise 1,  $b_1 = 0,8$  ve  $b_2 = 0,8$  ise 2) ve örnek numarası dikkate alınarak kodlanmıştır. Örneğin, 10-3-1-1, 3 makinede işlem görecektir 10 işin olduğu ve teslim zamanlarının sıkı olarak türetildiği problemin ilk örneğini göstermektedir. Her küçük boyutlu ( $n = 10$ ) problemlerden 3'er adet, büyük boyutlu problemlerden ( $n \geq 20$ ) ise birer adet örnek türetilmiştir. Toplamda ( $1 \times 2 \times 2 \times 3$ ) 12 adet küçük problem ve ( $4 \times 2 \times 2 \times 1$ ) 16 adet büyük problem türetilmiştir.

### 5.2. Örnek Problem (Sample problem)

Beş iş, iki paralel makinadan oluşan örnek problemin işlem süreleri ( $p_{jl}$ ), ilk sıradaki işin hazırlık süreleri ( $h_{jl}$ ) ve teslim zamanları ( $d_j$ ) Tablo 6'da ve sıra bağımlı hazırlık süreleri ( $s_{ijl}$ ) ise Tablo 7'de verilmiştir.

**Tablo 6.** Örnek Problemin  $p_{jl}, h_{jl}, d_j$  Parametre Değerleri (Parameter Values of Sample Problem)

$j$	$d_j$	$p_{jl}$		$h_{jl}$	
		$l=1$	$l=2$	$l=1$	$l=2$
1	4	70	86	69	66
2	29	73	53	63	5
3	49	58	2	97	10
4	15	16	77	73	78
5	65	53	37	89	22

**Tablo 7.** Örnek Problemin  $s_{ijl}$  Parametre Değerleri (Parameter Values of Sample Problem)

$i$	$j$	$l=1$					$l=2$				
		1	2	3	4	5	1	2	3	4	5
1		0	66	40	16	4	0	20	33	23	52
2		50	0	84	98	51	77	0	14	19	18
3		93	29	0	2	94	34	36	0	99	39
4		40	13	9	0	5	77	54	52	0	20
5		26	7	22	64	0	42	22	76	90	0

**Tablo 4:** Genişletilmiş  $\varepsilon$ -kısıt yönteminin adımları (Steps of Augmented  $\varepsilon$ -constraint)

<i>Adım 1:</i> (UPM) modeli için ideal $f^I = (f_1^I, f_2^I)$ ve nadir $f^N = (f_1^N, f_2^N)$ noktaları hesapla. (Ödönleşme tablosunun sözlüksel sıralama yaklaşımı ile oluşturulması iki amaçlı problemler için ideal ve nadir noktaların hesaplanması ile aynı anlama gelmektedir.)
<i>Adım 2:</i> F baskın noktalar kümesi olmak üzere, $(f_1^I, f_2^N)$ noktasını kümenin ilk elemanı olarak ata. Ve $\varepsilon$ 'a başlangıç değerini ata. $\varepsilon = f_2^N$ .
<i>Adım 3:</i> (UPM-AEC) modelini güncel $\varepsilon$ değeri için çöz. Elde edilen çözümü $(f_1^*, f_2^*)$ , F kümesine ekle. $\varepsilon = \varepsilon - \Delta$ formülünü kullanarak $\varepsilon$ değerini güncelle. (Burada $\Delta = (f_2^N - f_2^I)/50$ olarak alınmıştır.)
<i>Adım 4:</i> Eğer $\varepsilon \geq f_2^I$ ise Adım 3'e git. Değilse DUR.

Örnek problemin ideal noktasını bulabilmek için (UPM) modeli sırasıyla  $f_1$  ve  $f_2$  amaç fonksiyonları için tek amaçlı olarak çözülmüş ve  $y^1 = (171, 400)$  noktası elde edilmiştir. Nadir noktayı elde edebilmek için ise (UPM) modeli ilk olarak  $f_1$  amacı ve  $f_2 \leq 400$  ek kısıtı ile çözülmüş ve 194 amaç fonksiyonu değerine ulaşılmıştır. İkinci olarak ise model  $f_2$  amacı ve  $f_1 \leq 171$  ek kısıtı ile çözülmüş ve 430 amaç fonksiyonu değerine ulaşılmıştır. Böylece nadir nokta  $y^N = (194, 430)$  olarak elde edilmiştir. Örnek problem MSA ve AEC yöntemleri ile sırasıyla 51 farklı ağırlık seti ve 51 farklı  $\varepsilon$  değeri kullanılarak çözülmüştür. Her iki yöntemle de bir saniyenin altında bir sürede (171, 430) ve (194, 400) olmak üzere iki farklı baskın noktaya erişilmiştir. Elde edilen baskın çözümler beklendiği üzere ideal nokta ile nadir noktanın sınırladığı alanın içinde yer almaktadır.

(171, 430) baskın noktasına karşı gelen etkin çözüme ait Gantt şeması Şekil 1'de, (194, 400) baskın noktasına karşı gelen etkin çözüme ait Gantt şeması ise Şekil 2'de verilmiştir.

### 5.3. Test Sonuçları (Test Results)

Türetilen tüm test problemleri MSA ve AEC yöntemleri ile sırasıyla  $q \in \{1, 2, \dots, 51\}$  farklı ağırlık seti ve  $\varepsilon$  değeri kullanılarak çözülmüştür. Ağırlık setleri  $w_1^1 = 0$  ve  $w_2^1 = 50$  olmak üzere  $q > 1$  için Eş. 29 ve Eş. 30'da verilen formüller kullanılarak hesaplanmıştır. Her bir problem için  $\varepsilon_1 = f_1^N$  olarak alınmış ve  $k > 1$ , için  $\varepsilon$  değerleri Eş. 31'de verilen formül kullanılarak hesaplanmıştır.

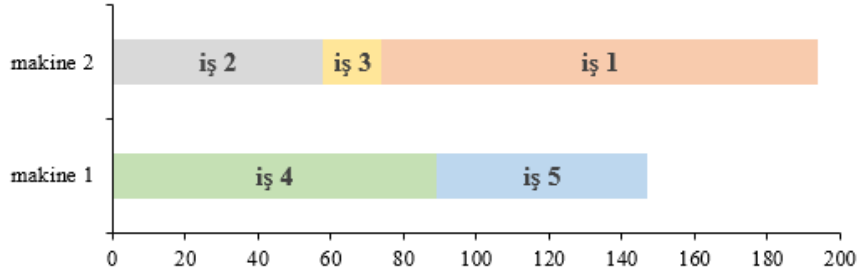
$$w_1^q = w_1^{q-1} + 1 \quad (29)$$

$$w_2^q = w_2^{q-1} - 1 \quad (30)$$

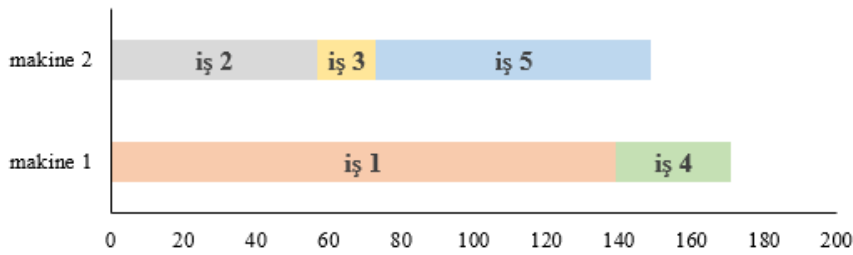
$$\varepsilon_q = \varepsilon_{q-1} - \frac{(f_1^N - f_1^1)}{50} \quad (31)$$

#### 5.3.1. Küçük boyutlu problemler ( $n = 10$ ) (Small sized problems ( $n = 10$ ))

Küçük boyutlu problemlerin ideal ve nadir noktaları Eş. 23 ve Eş. 24'te verildiği gibi GAMS Cplex çözücüsü kullanılarak hesaplanmış ve elde edilen değerler Tablo 8'de verilmiştir. Küçük boyutlu problemler MSA ve AEC ile çözülmüş ve iki makinanın olduğu test problemleri için elde edilen sonuçlar Tablo 9'da, üç makinanın olduğu problemlerin sonuçları ise Tablo 10'da sunulmuştur. Tablo 9 ve 10'un ilk sütununda test problemi numarası, ikinci ve üçüncü sütunlarında AEC yöntemi ile elde edilen baskın noktaların birinci ve ikinci amaç fonksiyonu değerleri, son olarak dördüncü ve beşinci sütunlarında ise MSA ile elde edilen baskın noktaların birinci ve ikinci amaç fonksiyonu değerleri verilmiştir. Tablolarda diğer yöntem tarafından bulunamamış baskın noktalar koyu yazılmıştır. Elde edilen sonuçlar incelendiğinde, AEC yönteminin 10-2-1-1, 10-3-1-3 ve 10-3-2-1 problemleri için MSA'nın bulamadığı birer baskın noktaya ulaştığı görülmektedir. MSA ise AEC yönteminin bulamadığı toplamda 11 baskın çözüme ulaşmıştır.



Şekil 1. (171, 430) baskın noktasına karşı gelen etkin çözüme ait Gantt şeması  
(Gantt schema of the efficient solution for non-dominated point (171,430))



Şekil 2. (194, 400) baskın noktasına karşı gelen etkin çözüme ait Gantt şeması  
(Gantt schema of the efficient solution for non-dominated point (194,400))

Tablo 8. Küçük Boyutlu Test Problemlerinin İdeal ve Nadir Noktaları  
(Ideal and Nadir Points of the Small-Sized Test Problems)

problem	ideal nokta	nadir nokta	problem	ideal nokta	nadir nokta
10-2-1-1	(388, 764)	(422, 932)	10-3-1-1	(145, 72)	(183, 116)
10-2-1-2	(307, 657)	(316, 737)	10-3-1-2	(211, 559)	(214, 586)
10-2-1-3	(218, 372)	(219, 454)	10-3-1-3	(183, 247)	(207, 437)
10-2-2-1	(322, 997)	(376, 1178)	10-3-2-1	(173, 225)	(231, 389)
10-2-2-2	(248, 493)	(263, 545)	10-3-2-2	(179, 471)	(203, 489)
10-2-2-3	(302, 747)	(315, 803)	10-3-2-3	(204, 430)	(273, 551)



**Tablo 9.** Küçük Boyutlu Problemler için Test Sonuçları ( $m=2$ )  
(Test Results for Small-sized problems ( $m=2$ ))

problem	$f_1^{AEC}$	$f_2^{AEC}$	$f_1^{MSA}$	$f_2^{MSA}$
10-2-1-1	388	932	388	932
	390	837	390	837
	-	-	415	822
	417	803	-	-
	422	764	422	764
10-2-1-2	451	769	451	769
	307	737	307	737
10-2-1-3	310	669	310	669
	316	657	316	657
	218	454	218	454
10-2-2-1	219	372	219	372
	322	1178	322	1178
10-2-2-2	333	1124	333	1124
	335	1029	335	1029
	355	1004	355	1004
	376	997	376	997
	248	545	248	545
10-2-2-3	263	493	263	493
	302	803	302	803
10-2-2-1	315	747	315	747

Küçük boyutlu problemler, AEC ve MSA ile 51 farklı  $\varepsilon$  değeri ve ağırlık seti ile çözüldüğünden her bir problem için bu çözümlerin ortalama süresi hesaplanmış ve Tablo 11'de verilmiştir. Yöntemler çözüm süreleri açısından değerlendirildiğinde MSA'nın AEC yöntemine küçük problemlerde bile avantaj sağladığı görülmektedir.

### 5.3.2. Büyük boyutlu problemler ( $n \geq 20$ ) (Large sized problems ( $n \geq 20$ ))

Büyük boyutlu problemlerde ideal ve nadir noktaların bulunması için kullanılan matematiksel modeller ile eniyi çözümler elde edilemediğinden, ideal ve nadir noktalar yerine amaç fonksiyonlarının alt ve üst sınır değerleri kullanılmıştır.  $f_1$  amacının sınır değerlerinin hesaplanmasında Saraç vd. [56]'in önerdiğine benzer bir yapı benimsenmiştir.  $f_1$  alt sınır ( $f_1^{AS}$ ) formülü Eş. 32'de, üst sınır ( $f_1^{US}$ ) formülü ise Eş. 33'te verilmiştir.  $f_2$  amacı için alt sınır değeri ( $f_2^{AS}$ ) ise her işin ilk hazırlık süresi ve işlem süresi toplamının enküçük olduğu  $l^*$  makinasına ( $l^* = \underset{l}{\operatorname{argmin}}(h_{jl} + p_{jl})$ ) ilk sırada atanması durumunda

oluşacak gecikmelerin toplamı alınarak hesaplanmıştır.  $f_2$  amacının üst sınırını hesaplamak için öncelikle her iş enbüyük işlem süresine sahip olduğu  $l^{**}$  makinasına ( $l^{**} = \underset{l}{\operatorname{argmax}}(p_{jl})$ ) atanmıştır.

Sonrasında her makinaya atanan işlerin sıralarını belirlemek üzere işler, teslim zamanları azalan şekilde sıralanmıştır. Ve bu atama ve sıralar için toplam gecikme hesaplanmıştır.

$$f_1^{AS} = 1,05 \frac{\sum_j m_{in} p_{jl}}{m} \quad (32)$$

$$f_1^{US} = 1,05 \frac{\sum_j (\max_{i,l} s_{ijl} + \max_{i,l} p_{jl})}{m} \quad (33)$$

Büyük boyutlu problemler MSA ve AEC ile çözülmüş ve elde edilen sonuçlar Tablo 12'de, ortalama çözüm süreleri ise Tablo 13'de sunulmuştur.

Tablo 12'de her iki yöntemle elde edilen amaç fonksiyonu değerleri, kendi içlerinde karşılaştırılmış ve her yöntemin baskın çözümleri ayrı ayrı sunulmuştur. Tablodan da görülebileceği gibi 30-2-2-1, 40-2-2-1, 50-2-2-1 ve 50-3-1-1 problemlerine AEC yöntemi ile uygun çözüm bulunması mümkün olamamıştır. MSA ise tüm problemlere çözüm bulmuştur. Her iki yöntemle bulunan noktaların birbiri ile karşılaştırılabilmesi için problem bazında çizilen grafikler iki makine ( $m=2$ ) problemleri için Şekil 3'de ve üç makine ( $m=3$ ) problemleri için de Şekil 4'de verilmiştir. Tüm grafiklerde alt sınırları gösteren noktalar için sarı, üst sınırları gösteren noktalar için gri, AEC ile elde edilen noktalar için mavi ve MSA ile elde edilen noktalar için ise turuncu renk kullanılmıştır.

**Tablo 10.** Küçük Boyutlu Problemler için Test Sonuçları ( $m=3$ )  
(Test Results for Small-sized problems ( $m=3$ ))

problem	$f_1^{AEC}$	$f_2^{AEC}$	$f_1^{MSA}$	$f_2^{MSA}$
10-3-1-1	145	116	145	116
	183	72	183	72
10-3-1-2	211	586	211	586
	214	559	214	559
10-3-1-3	183	437	183	437
	188	430	188	430
	194	399	194	389
	-	-	198	385
	202	312	-	-
10-3-2-1	-	-	205	311
	207	247	207	247
	173	389	173	389
	179	376	-	-
	188	297	188	297
	-	-	194	285
	-	-	196	275
-	-	209	247	
10-3-2-2	231	225	231	225
	179	489	179	489
	181	486	181	486
10-3-2-3	203	471	203	471
	204	551	204	551
10-3-2-1	-	-	205	483
	-	-	211	465
	-	-	213	446
	-	-	215	441
	-	-	220	433
	-	-	273	430
	273	430	273	430

**Tablo 11.** Küçük Boyutlu Problemler için Ortalama Çözüm Süreleri (Average CPU Times for Small-sized problems )

<i>problem</i>	$\bar{t}_{AEC}$	$\bar{t}_{MSA}$	<i>problem</i>	$\bar{t}_{AEC}$	$\bar{t}_{MSA}$
10-2-1-1	4594	555	10-3-1-1	244	230
10-2-1-2	3633	393	10-3-1-2	4029	293
10-2-1-3	794	387	10-3-1-3	1089	255
10-2-2-1	3899	405	10-3-2-1	1619	293
10-2-2-2	1649	393	10-3-2-2	3471	478
10-2-2-3	3958	547	10-3-2-3	2235	383

Şekil 3 ve 4'te verilen grafikler incelendiğinde hem iki makine hem de üç makine 20 boyutlu problemlerinde AEC ve MSA yöntemleri ile elde edilen noktalar birbirine yakın iken problem boyutu büyüdükçe MSA ile alt sınır değerlerine çok daha yakın ve AEC ile elde edilen noktalara baskın çözümler elde edildiği görülmektedir. Sonuç olarak önerilen MSA literatürde kabul görmüş amaç birleştirme yöntemlerinden birisi olan AEC yönteminin çözemediği problemleri çözebilmiş, çözebildikleri için ise çok daha kısa sürelerde daha

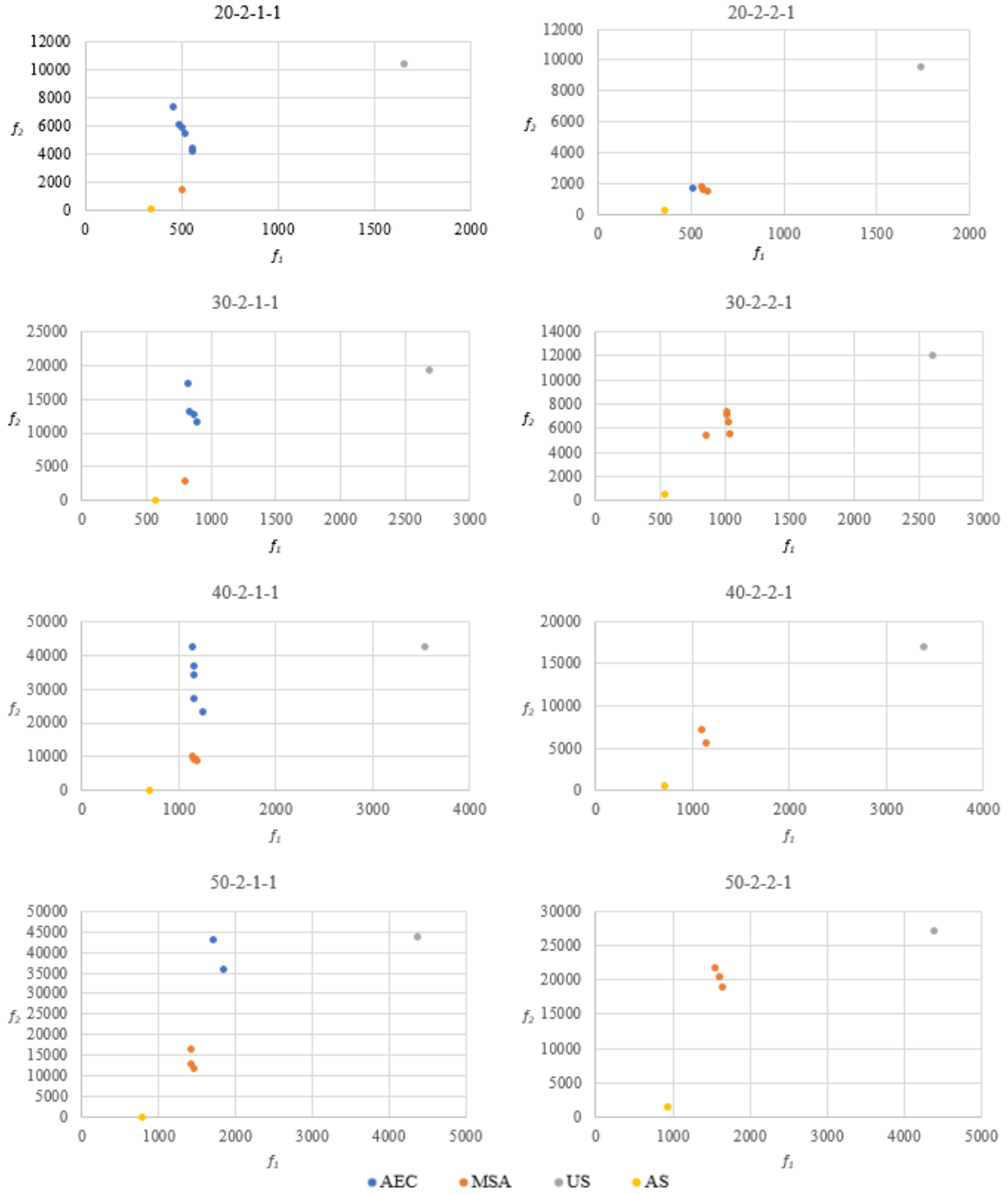
başarılı çözümler elde edebilmiştir. Büyük boyutlu problemler için ortalama çözüm süreleri Tablo 10'da verilmiştir. Yöntemler çözüm süreleri açısından değerlendirildiğinde MSA'nın AEC yöntemine avantaj sağladığı görülmektedir.

**Tablo 13.** Büyük Boyutlu Problemler için Ortalama Çözüm Süreleri (Average CPU Times for Large-sized problems )

<i>problem</i>	$\bar{t}_{AEC}$	$\bar{t}_{MSA}$	<i>problem</i>	$\bar{t}_{AEC}$	$\bar{t}_{MSA}$
20-2-1-1	7200	1268	20-3-1-1	7200	1236
20-2-2-1	7200	1449	20-3-2-1	7200	1531
30-2-1-1	7200	984	30-3-1-1	7200	1680
30-2-2-1	7200	1664	30-3-2-1	7200	1567
40-2-1-1	7200	1549	40-3-1-1	7200	1725
40-2-2-1	7200	1393	40-3-2-1	7200	1708
50-2-1-1	7200	1385	50-3-1-1	7200	1285
50-2-2-1	7200	1531	50-3-2-1	7200	1154

**Tablo 12.** Büyük Boyutlu Problemler için Test Sonuçları (Test Results for Large-sized problems )

<i>problem</i>	$f_1^{AEC}$	$f_2^{AEC}$	$f_1^{MSA}$	$f_2^{MSA}$	<i>problem</i>	$f_1^{AEC}$	$f_2^{AEC}$	$f_1^{MSA}$	$f_2^{MSA}$		
20-2-1-1	454	7335,80	499	1533	20-3-1-1	347	1389,00	366	1216		
	488	6108,92				375	1360,00			372	1060
	503	5904,44				375	987				
	516	5495,48				379	946				
	554	4473,08				397	915				
	555	4268,60				401	842				
20-2-2-1	513	1727,00	559	1822	20-3-2-1	298	588,00	300	493		
			562	1583		340	488,00	318	362		
			590	1490		336	217				
30-2-1-1	824	17355,70	795	2829	30-3-1-1	488	10482,90	508	1602		
			832	13122,24		494	7563,30	529	1164		
			864	12737,38		520	6833,40	561	1156		
			892	11582,80		513	2314				
30-2-2-1			1003	7448	30-3-2-1	453	6920,96	513	2314		
			1006	7179		568	6220,52	543	1643		
			1014	6555				589	1510		
			1030	5587		40-3-1-1	1015	11076,00	730	419	
			843	5458		1135	10632,96	803	299		
40-2-1-1	1137	42662,00	1140	10119	40-3-2-1	786	16083,36	745	2688		
			1148	9431		787	13468,80	767	2618		
			1161	9173		886	13141,98	800	2384		
			1183	8796		927	11834,70	817	2196		
			1238	23184,68		983	11181,06	833	1679		
40-2-2-1			1086	7321				850	1587		
			1126	5771				50-3-1-1		875	3505
50-2-1-1	1706	42982,62	1416	16525				905	2907		
			1422	12854				936	2477		
			1454	11655				1002	2458		
50-2-2-1			1534	21936	50-3-2-1	1456	22462,00	849	3500		
			1598	20657		1461	18714,00	1007	3497		
			1638	19016		1586	15904,00	1146	2758		

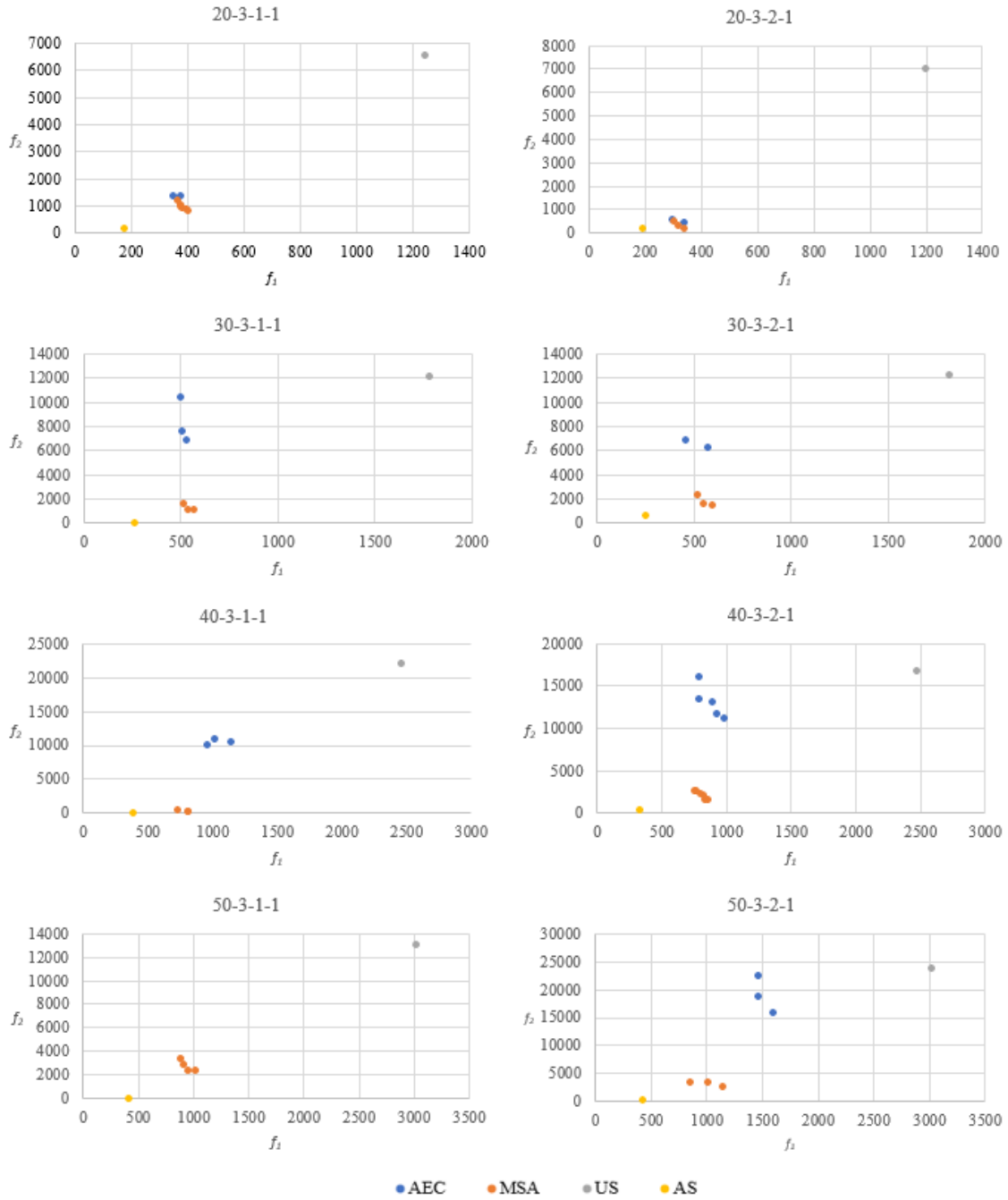


Şekil 3. Büyük Boyutlu Problemler için Baskın Noktalar ( $m=2$ ) (Non-dominated Points for Large-sized Problems ( $m=2$ ))

## 6. Sonuçlar (Conclusion)

Paralel makine çizelgeleme problemleri daha karmaşık makine ortamlarının çizelgenmesinde temel teşkil ettiği için oldukça önemli bir problemdir. Bu problem için geliştirilecek yaklaşımlar daha karmaşık problemlere uyarlanabilmektedir. Bu çalışmada sıra bağımlı hazırlık sürelerinin olduğu ilişkisiz paralel makine çizelgeleme problemi için son işin tamamlanma zamanının ve toplam gecikmenin enküçüklenmesi amaçlanmıştır. Ele alınan problemin çözümü için bir

mat sezgisel algoritma geliştirilmiştir. Geliştirilen algoritma ile elde edilen sonuçlar, genişletilmiş  $\varepsilon$ -kısıt yöntemi ile elde edilen çözümler ile karşılaştırılmıştır. Rastal olarak türetilen problemler ile yapılan testler sonucunda önerilen MSA'nın literatürde yer alan başarılı amaç birleştirme yöntemlerinden birisi olan AEC yönteminin çözemediği problemleri çözebildiği, çözebildikleri için ise çok daha kısa sürelerde daha başarılı çözümler elde edebildiği görülmüştür. Ele alınan problem, gelecek çalışmalarda sanayide karşılaşılan farklı kısıtlar dahil edilerek genişletilebilir.



Şekil 4. Büyük Boyutlu Problemler için Baskın Noktalar ( $m=3$ ) (Non-dominated Points for Large-sized Problems ( $m=3$ ))

#### Kaynaklar (References)

- Gedik R., Kalathia D., Egilmez G., et al., A constraint programming approach for solving unrelated parallel machine scheduling problem, *Computers & Industrial Engineering*, 121, 139-149, 2018.
- 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.
- 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.
- Rosales O.A., Bello F.A., Alvarez A., Efficient metaheuristic algorithm and reformulations for the unrelated parallel machine scheduling problem with sequence and machine dependent setup times, *International Journal of Advanced Manufacturing Technology*, 76, 1705-1718, 2015.
- Arnout J. P., Rabadi G., Musa R., Two stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines-part II: enhancements and experimentations, *Journal of Intelligent Manufacturing*, 25, 43-53, 2014.
- 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.
- 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.

8. Ezugwu A.E., Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times, *Knowledge-Based Systems*, 172, 15-32, 2019.
9. Santos H.G., Toffolo T.A.M., Silva, C.L.T. F., Berghes G.V., Analysis of stochastic local search methods for the unrelated parallel machine scheduling problem, *International Transactions In Operational Research*, 26 (2), 707-724, 2019.
10. 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.
11. 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.
12. Arnaut J.P., A worm optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times, *Annals of Operations Research*, 285 (1-2), 273-293, 2020.
13. Ezugwu A.E., Akutsah F., An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times, *IEEE Access*, 6, 54459- 54478, 2018.
14. Ezugwu A.E., Adeleke O.J., Viriri S., Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times, *PLOS ONE*, 13(7) Article Number: e0200030, 2018.
15. Abreu L.R., Prata B.A., A Hybrid Genetic Algorithm for solving the Unrelated Parallel Machine Scheduling problem with Sequence Dependent Setup Times, *IEEE Latin America Transactions*, 16 (6), 1715-1722, 2018.
16. Vallada E., Ruiz 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.
17. 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.
18. 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.
19. Jouhari H., Lei D., Al-qaness M.A.A. et al., Sine-Cosine Algorithm to Enhance Simulated Annealing for Unrelated Parallel Machine Scheduling with Setup Times, *Mathematics*, 7 (11), 1120, 2019.
20. 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.
21. 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.
22. 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.
23. Rambod M., Rezaeian J., Robust meta- heuristic implementation for unrelated parallel machines scheduling problem with rework processes and machine eligibility restrictions, 77, 15- 28, 2014.
24. Yepes-Borrero J.C., Villa F., Perea F., et al. GRASP algorithm for the unrelated parallel machine scheduling problem with setup times and additional resources, *Expert Systems with Applications*, 141, Article Number: 112959, 2020.
25. Hsu C.J., Cheng T.C.E., Yang D.L., Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time, *Information Sciences*, 181, 4799- 4803, 2011.
26. 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.
27. Rauchecker G., Schryen G., Using high performance computing for unrelated parallel machine scheduling with sequence-dependent setup times: Development and computational evaluation of a parallel branch-and-price algorithm, *Computers & Operations Research*, 104, 338-357, 2019.
28. 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.
29. 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.
30. 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.
31. 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.
32. 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.
33. 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.
34. Bektur G., Sarac 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.
35. 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.
36. 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.
37. Jeong B., Kim S.W., Lee Y.J., An assembly scheduler for TFT LCD manufacturing, *Computers & Industrial engineering*, 41, 37- 58, 2001.
38. Li Q., Milne R.J., A production scheduling problem with sequence dependent changeover cost, *International Journal of Production Research*, 52 (13), 4093-4102, 2014.
39. Canyılmaz E., Benli B., Ilkay 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.
40. Wang M., Pan G., A Novel Imperialist Competitive Algorithm with Multi-Elite Individuals Guidance for Multi-Object Unrelated Parallel Machine Scheduling Problem, *IEEE Access*, 7, 121223-121235, 2019.
41. 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.
42. Saricicek I., Multi-Objective scheduling by maximizing machine preferences for unrelated parallel machines, *Sigma Journal of Engineering and Natural Sciences*, 38 (1), 405-420, 2020.
43. 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.
44. 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.
45. Ekici A., Elyasi M., Ozener O.O., et al., An application of unrelated parallel machine scheduling with sequence-dependent setups at Vestel Electronics, *Computers & Operations Research*, 111, 130-140, 2019.
46. 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.
47. Bozorgirad, M.A., Logendran R., Sequence dependent group scheduling problem on unrelated parallel machines, *Expert Systems with Applications*, 39, 9021- 9030, 2012.
48. 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.

49. 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.
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. 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.
52. Saraç T., Tutumlu B., A bi-objective mathematical model for an unrelated parallel machine scheduling problem with job-splitting, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37 (4), 2293-2308, 2022.
53. Saraç T., Tutumlu B., A mix integer programming model and solution approaches to determine the optimum machine number in the unrelated parallel machine scheduling problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37 (1), 329-345, 2022.
54. Mavrotas G., Effective implementation of the  $\varepsilon$ -constraint method in Multi-Objective Mathematical Programming problems, *Applied Mathematics and Computation*, 213, 455–465, 2009.
55. Logendran R., McDonell B., Smucker B., Scheduling unrelated parallel machines with sequence-dependent setups, *Computers & Operations Research*, 34 (11), 3420-3438, 2007.
56. Saraç T., Sipahioglu A., Akyol Ozer E., A two-stage solution approach for plastic injection machines scheduling problem, *Journal of Industrial and Management Optimization*, 17 (3), 1289-1314, 2021.