



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN
BULANIK ORTAMDA DAĞINIK ARAMA YÖNTEMİ İLE ÇÖZÜMÜ**

Ramazan BAŞAR

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

EKİM-2021
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ramazan BAŞAR tarafından hazırlanan “BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN BULANIK ORTAMDA DAĞINIK ARAMA YÖNTEMİ İLE ÇÖZÜMÜ” adlı tez çalışması 22/10/2021 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Orhan ENGİN

.....

Danışman

Prof. Dr. Orhan ENGİN

.....

Üye

Doç. Dr. Ahmet SARUCAN

.....

Üye

Dr. Öğr. Üyesi Serkan KAYA

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ramazan BAŞAR

22/10/2021

ÖZET

YÜKSEK LİSANS TEZİ

BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN BULANIK ORTAMDA DAĞINIK ARAMA YÖNTEMİ İLE ÇÖZÜMÜ

Ramazan BAŞAR

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Orhan ENGİN

2021, 75 Sayfa

**Jüri
Prof. Dr. Orhan ENGİN
Doç. Dr. Ahmet SARUCAN
Dr. Öğr. Üyesi Serkan KAYA**

Çizelgeleme problemleri, günümüzde üzerinde çokça durulan önemli bir konudur. Bu çalışmada, Akış Tipi Çizelgeleme problemi, işlerin beklemesiz olma kısıtı ile ele alınmıştır. Literatürde, 2 makineden fazla olan Beklemesiz Akış Tipi Çizelgeleme (BATÇ) problemlerini matematiksel bir yöntem ile, kabul edilebilir bir sürede çözebilecek bir metot bulunmamaktadır. Bu yüzden Beklemesiz Akış Tipi Çizelgeleme, NP (Polinomiyel Olmayan)-zor problemler sınıfında yer almaktadır. Bu problemlerin çözümü için, optimum ya da optimuma yakın sonuç verecek meta-sezgisel yöntemler geliştirilmektedir. Bu çalışmada ise literatürde birçok probleme uyarlanmış ve oldukça verimli sonuçlar elde edebilen, evrimsel bir arama tekniği olan Dağınık Arama Yöntemi kullanılmıştır. Çalışma sonuçlarının etkinliğini ispatlamak için, literatürde iyi bilinen 192 adet test problemi çözülmüştür. Elde edilen çalışma sonuçları daha önce Beklemesiz Akış Tipi Çizelgeleme problemi için yapılmış olan Genetik Algoritma (Gen-2), Uyarlanabilir Öğrenme Yaklaşım (UÖY) algoritması ve Hibrit Karınca Kolonileri Optimizasyon (HKKO) algoritması ile kıyaslanmıştır. Gerçek hayatta öngörülemeyen sonuçların teslim tarihlerini belirsiz kılmasından dolayı, mevcut problemlerin teslim tarihleri bulanıklaştırılmıştır ve müşteri tatmin indeksleri hesaplanmıştır. Elde edilen tüm sonuçlar göz önüne alındığında, önerilen dağınık arama yönteminin, literatürdeki diğer sezgisel ve meta-sezgisel yöntemler ile rekabet edebilecek, oldukça verimli bir arama tekniği olduğu görülmüştür.

Anahtar Kelimeler: Beklemesiz Akış Tipi Çizelgeleme, Bulanık Teslim Tarihleri, Dağınık Arama Yöntemi, Müşteri Tatmin İndeksi

ABSTRACT

MS THESIS

**SOLUTION OF NO-WAIT FLOW SHOP SCHEDULING PROBLEMS WITH
SCATTER SEARCH METHOD IN FUZZY ENVIRONMENT**

Ramazan BAŞAR

**Konya Technical University
Institute of Graduate Studies
Department of Industrial Engineering**

Advisor: Prof. Dr. Orhan ENGİN

2021, 75 Pages

**Jury
Prof. Dr. Orhan ENGİN
Assoc. Prof. Dr. Ahmet SARUCAN
Asst. Prof. Dr. Serkan KAYA**

Scheduling problems are an important topic that is emphasized a lot today. In this study, the Flow Shop Scheduling problem is handled with the constraint of jobs being no-wait. In the literature, no method can solve the No-Wait Flow Shop Scheduling (NWFSS) problems of more than 2 machines with a mathematical method in an acceptable time. Therefore, No-Wait Flow Shop Scheduling is included in the class of NP (Non-Polynomial)-hard problems. For the solution of these problems, meta-heuristic methods are being developed that will give optimum or near-optimal results. In this study, the Scatter Search Method, this is an evolutionary search technique that has been adapted to many problems in the literature and can yield very efficient results, has been used. To prove the effectiveness of the study results, 192 well-known test problems in the literature were solved. The results of the study were compared with the Genetic Algorithm (Gen-2), Adaptive Learning Approach (ALA) algorithm, and Hybrid Ant Colony Optimization (HACO) algorithm, which were previously made for the No-Wait Flow Shop Scheduling problem. Due to the unpredictable results in real life making the due dates uncertain, the due dates of the existing problems were fuzzed and customer satisfaction indexes were calculated. Considering all the results obtained, it has been seen that the proposed Scatter Search Method is a highly efficient search technique that can compete with other heuristics and meta-heuristics in the literature.

Keywords: No-Wait Flow Shop Scheduling, Fuzzy Due Date, Scatter Search Method, Customer Satisfaction Index

ÖNSÖZ

Yüksek lisans tez çalışmam sırasında bilgi ve tecrübelerinden yararlandığım, desteğini benden esirgemeyen değerli danışman hocam, Prof. Dr. Orhan ENGİN'e ve her zaman maddi ve manevi destekleri ile yanımda olan çok değerli aileme sonsuz teşekkürlerimi sunarım.

Ramazan BAŞAR
KONYA-2021



İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
ŞEKİL LİSTESİ	viii
ÇİZELGE LİSTESİ	ix
SİMGELER VE KISALTMALAR	x
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
2.1. Beklemez Akış Tipi Çizelgeleme Kaynak Araştırması	3
2.2. Dağınık Arama Yöntemi Kaynak Araştırması.....	13
3. MATERYAL VE YÖNTEM	16
3.1. Materyal	16
3.1.1. BATÇ Problemlerinin Notasyonu	16
3.1.2. BATÇ Problemlerinin Gösterimi.....	16
3.1.3. BATÇ Problemlerinin Matematiksel Modeli	17
3.1.4. BATÇ Problemlerinde Teslim Tarihlerinin Bulanıklaştırılması	18
3.1.5. Örnek BATÇ Problemi	20
3.2. Yöntem.....	22
4. BATÇ PROBLEMLERİNİN ÖNERİLEN DAY İLE ÇÖZÜMÜ	27
5. SONUÇLAR VE ÖNERİLER	53
5.1 Sonuçlar	53
5.2 Öneriler	54
KAYNAKLAR	55
ÖZGEÇMİŞ	Hata! Yer işareti tanımlanmamış.

ŞEKİL LİSTESİ

Şekil 1. Teslim tarihlerinin üyelik fonksiyonu	19
Şekil 2. 3-iş x 2-makine problemi tüm kombinasyonların Gantt çizelgesi	20
Şekil 3. 3-iş x 2-makine BATÇ probleminin grafik gösterimi	21
Şekil 4. Dağınık Arama Yöntemi çözüm adımları	23



ÇİZELGE LİSTESİ

Çizelge 1. Kronolojik sıraya göre BATÇ kaynak araştırması	3
Çizelge 2. Kronolojik sıraya göre DAY kaynak araştırması	13
Çizelge 3. 3-iş x 2-makine probleminin hazırlık ve işlem süreleri.....	20
Çizelge 4. 3-iş x 2-makine BATÇ probleminin çözümleri.....	21
Çizelge 5. n=8 için CPLEX sonucuna göre sapma miktarları.....	29
Çizelge 6. n=10 iş için CPLEX sonucuna göre sapma miktarları	30
Çizelge 7. CPLEX sonucuna göre algoritmaların kıyası.....	31
Çizelge 8. 2 makineli problemler için sapma miktarları	32
Çizelge 9. 3 makineli problemler için sapma miktarları	33
Çizelge 10. 5 makineli problemler için sapma miktarları	34
Çizelge 11. 8 makineli problemler için sapma miktarları	35
Çizelge 12. 10 makineli problemler için sapma miktarları	36
Çizelge 13. 15 makineli problemler için sapma miktarları	37
Çizelge 14. 20 makineli problemler için sapma miktarları	38
Çizelge 15. 25 makineli problemler için sapma miktarları	39
Çizelge 16. 192 test probleminde makinelere göre bulunan en iyi C_{max} sayıları	40
Çizelge 17. 192 test probleminde makinelere göre ortalama sapma miktarları	40
Çizelge 18. Bilgisayar ortamı ve maksimum işlem süreleri.....	41
Çizelge 19. 8 iş problemlerinde işlerin akış süreleri	42
Çizelge 20. 10 iş problemlerinde işlerin akış süreleri	43
Çizelge 21. 12 ve 50 iş problemlerinde işlerin akış süreleri.....	44
Çizelge 22. 100 ve 150 iş problemlerinde işlerin akış süreleri.....	45
Çizelge 23. 200 ve 250 iş problemlerinde işlerin akış süreleri.....	46
Çizelge 24. 8 iş problemlerinde ortalama tatmin indeksleri	47
Çizelge 25. 10 iş problemlerinde ortalama tatmin indeksleri.....	48
Çizelge 26. 12 ve 50 iş problemlerinde ortalama tatmin indeksleri	49
Çizelge 27. 100 ve 150 iş problemlerinde ortalama tatmin indeksleri	50
Çizelge 28. 200 ve 250 iş problemlerinde ortalama tatmin indeksleri	51
Çizelge 29. İş sayısına göre ortalama tatmin indeksleri	52

SİMGELER VE KISALTMALAR

Simgeler

C_j	: Tamamlanma Zamanı
C_{max}	: En Büyük Tamamlanma Zamanı
d_j	: j İşinin Teslim Tarihi (Due Date)
\tilde{D}_j	: j İşinin Bulanık Teslim Tarihi
F_j	: Akış Süresi (Flow Time)
j	: İş İndisi
k	: Makine İndisi
L_j	: Geç Kalma Süresi (Lateness)
L_{max}	: Maksimum gecikme
m	: Makine Sayısı
mg	: Minimum Gecikme
n	: İş sayısı
P	: Farklılaştırma Kümesi
p_{ij}	: İşlem Süresi (Processing Time)
R	: Referans Kümesi
r_j	: Serbest Bırakma Zamanı (Release Date)
T_j	: Gecikme (Tardiness)
w_j	: Ağırlık (Weight)
α	: Makine Çevresi
β	: Kısıtlar
γ	: Amaç Fonksiyonu
π	: İş Dizilimi

Kısaltmalar

ATÇ	: Akış Tipi Çizelgeleme
BATÇ	: Beklemesiz Akış Tipi Çizelgeleme
DAY	: Dağınmık Arama Yöntemi
HKKO	: Hibrit Karınca Koloni Optimizasyonu
NEH	: Nawaz, Enscore ve Ham
NP-Zor	: Polinomiyel olmayan zor problemler
UÖY	: Uyarlanabilir Öğrenme Yaklaşımı

1. GİRİŞ

Günümüzde artan küresel rekabet, hizmet ve üretim sektörlerinde yeni ihtiyaçların çıkmasına neden olmuştur. Bu ihtiyaçların başında ürün veya hizmetin müşteriye tam zamanında teslim edilmesi ve gecikmelerin önüne geçilmesi gelmektedir. Bu sorunların üstesinden gelebilmek için yapılması gereken işlerin, teslim tarihleri dikkate alınarak bir uyum içerisinde çizelgelenmesi gerekmektedir. Çizelgeleme problemlerinin önemli bir kısmını Akış Tipi Çizelgeleme (ATÇ) problemleri oluşturmaktadır.

Akış tipi ortamında yapılması gereken işler aynı görev listesine sahiptir, yani işler aynı makinelerde aynı sırada işlem görürler. Bu çalışmada, ATÇ problemlerinden olan, Beklemesiz Akış Tipi Çizelgeleme (BATÇ) ele alınmıştır. BATÇ ortamında, sürecin başlangıcından sonuna kadar sistemin herhangi bir aşamasında, işlerin beklemesine izin verilmeksizin kesintisiz olarak yapılması gerekir. BATÇ problemlerinde, makinelerin beklemesine izin verilebilir fakat makinelerin boşta bekleme sürelerinin minimizasyonu, müşteri tatmini ve maliyetlerin düşürülmesi için gereklidir. Uçuş, tren ve ameliyat çizelgeleme problemleri, BATÇ'nin birkaç örneğidir.

BATÇ üzerine yapılmış olan çalışmalarda toplam gecikme ve maksimum tamamlanma zamanı olmak üzere iki amaç fonksiyonu daha çok dikkate alınmaktadır. Toplam gecikme amaç fonksiyonu, müşteri taleplerinin daha baskın olduğu durumlarda ele alınması gereken performans kriteridir. İşlerin teslim tarihleri ile tamamlanma süreleri arasındaki farkların artması, toplam gecikme amaç fonksiyonunun artmasına neden olur. Toplam gecikme amaç fonksiyonunun artması, müşteri taleplerinin karşılanmasında gecikmelere yol açacağı için istenmeyen bir durumdur. Günümüzün şiddetli küresel rekabeti, müşteri tatmininin teslim tarihinden etkilenmesinden dolayı, toplam gecikme performans ölçümü daha da önemli hale gelmektedir. Öte yandan pek çok kaynak kıt olduğundan ve bu tür kıt kaynakların verimli kullanımı birçok imalat firması için önemli olduğundan, performans ölçümü doğrudan kaynak kullanımı ile ilgilidir. Bu nedenle, her iki performans ölçümü de kritik öneme sahiptir. Bundan dolayı her ikisi de üretim çizelgelemesinde dikkate alınmalıdır.

Gerçek yaşam koşullarında, çeşitli faktörlerin etkisinden dolayı teslim tarihleri belirsizlik gösterebilmektedir. Bu çalışmada, BATÇ problemlerinin teslim tarihleri bulanıklaştırılarak gerçek yaşam koşullarına yakınsanmaya çalışılmıştır. Bulanıklaştırılmış teslim tarihlerine göre müşteri tatmin indeksleri hesaplanmıştır.

Literatürde, BATÇ problemleri için bir veya iki makineli durumlarda, standart bir sürede, optimum sonuç veren, matematiksel yöntemler bulunurken, iki makineden fazla m -makineli problemlerde, standart bir süre içerisinde, optimum sonuç veren, matematiksel bir yöntem bulunmamaktadır. Bu sorunun üstesinden gelebilmek için sezgisel ve meta-sezgisel yöntemlerin kullanımına başvurulmuştur.

Sezgisel ve meta-sezgisel yöntemler, standart bir süre içerisinde, polinomiyele olarak çözülemeyen problemler için optimum veya optimuma çok yakın sonuçlar vermeyi amaçlar. Bu çalışmada ise literatürde birçok probleme uyarlanmış olan Dağınık Arama Yöntemi (DAY), BATÇ problemine uyarlanmıştır.

DAY, ilk olarak Fred W. Glover tarafından, 1970'lerde geliştirilmiş, evrimsel bir arama tekniğidir. DAY, genel olarak Tabu Arama algoritmasının uyarlanabilir hafıza özelliğinin, evrimsel aramaya yerleştirilmiş halidir. DAY, iyi çözümler elde etmek için rastsal veya belli bir stratejik yöntem doğrultusunda oluşturulmuş iyi bireylerin seçilmesi ve bu bireylerin birleştirilerek daha iyi çözümler elde edilmesi esasına dayanır.

Çalışmanın ilerleyen bölümleri aşağıdaki gibi organize edilmiştir. İkinci bölümünde, BATÇ ve DAY ile ilgili literatür araştırması yapılmış ve kronolojik sıraya göre çalışmalar listelenmiştir. Ayrıca literatürde çalışılmış olan BATÇ problemlerinin birçok kritere göre analizi yapılmıştır. Üçüncü bölümde, BATÇ problemlerinin, notasyon ve gösterimi ile ilgili bilgiler verilmiştir. Dördüncü bölümün materyal kısmında, bulanık teslim tarihleri ile ele alınan BATÇ problemleri hakkında bilgi verilirken, metod kısmında ise, DAY hakkında bilgiler verilmiştir. Beşinci bölümde, araştırmanın sonuçları paylaşılmış ve literatürde iyi bilinen üç sezgisel algoritma ile kıyaslama yapılarak sonuçlar tartışılmıştır. Son bölüm olan altıncı bölümde, nihai sonuçlar hakkında bilgi verilmiştir ve gelecek çalışmalar için öneride bulunulmuştur.

2. KAYNAK ARAŞTIRMASI

2.1. Beklemesiz Akış Tipi Çizelgeleme Kaynak Araştırması

Bu bölümde, literatürde BATÇ üzerine yapılmış çalışmalar kronolojik sıraya göre Çizelge 1’de sunulmuştur. Ayrıca BATÇ üzerine yapılmış olan çalışmalar, kullanılan çözüm yöntemi, çalışılan konu ve amaç fonksiyonuna göre analiz edilmiştir.

Çizelge 1. Kronolojik sıraya göre BATÇ kaynak araştırması

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Strusevich (1995)	İşlem sırasında beklemeden iki makine akış tipi çizelgeleme problemi: Kontrol edilebilir makine hızları	$O(n^3)$	Maksimum tamamlanma zamanı
Espinouse ve ark. (1999)	İki makineli, beklemesiz akış atölyesinde, sınırlı makine kullanılabilirliği ile maksimum tamamlanma zamanının azaltılması	Gilmore ve Gomory algoritması	Maksimum tamamlanma zamanı
Kumar ve ark. (2000)	m -makineli, beklemesiz akış istasyonları için parti akışı ve çizelgeleme sezgiselleri	Lot streaming (Parti akışı)	Maksimum tamamlanma zamanı
Moghaddam ve ark. (2008)	Çok amaçlı, beklemesiz akış tipi çizelgeleme problemini bağışıklık algoritmasıyla çözme	Bağışıklık algoritması	Toplam ağırlıklı gecikme
Zhou ve Gu (2009)	Evrım oyunlarına dayalı, bulanık teslim tarihli, beklemesiz akış tipi çizelgeleme problemi üzerine araştırma	Hibrit	Maksimum tamamlanma zamanı
Chihaoui ve ark. (2011)	Müsait olmama kısıtlaması ve farklı serbest bırakma tarihleri altındaki maksimum tamamlanma zamanını en aza indirmek için iki makineli beklemesiz akış tipi çizelgeleme	Dal-sınır algoritması	Maksimum tamamlanma zamanı
Engin ve Günaydın (2011)	Beklemesiz akış tipi çizelgeleme problemlerinde, maksimum tamamlanma zamanını minimuma indirmek için uyarlanabilir öğrenme yaklaşımı	Uyarlanabilir öğrenme yaklaşımı	Maksimum tamamlanma zamanı
Gao ve ark. (2011)	Toplam akış zamanı kriteri ile beklemesiz akış tipi çizelgeleme problemi için ayrık uyum arama algoritması	Ayrık uyum arama algoritması	Toplam akış zamanı
Shafaei ve ark. (2011)	Çok işlemcili, beklemesiz iki aşamalı akış atölyesinde, maksimum tamamlanma zamanı tahmini için uyarlamalı bir nöro bulanık çıkarım sistemi	Uyarlanabilir nöro bulanık çıkarım yöntemi	Maksimum tamamlanma zamanı
Zhao ve Tang (2011)	Kötüleşen işler ve makine kullanılabilirliği kısıtlamaları ile iki makineli beklemesiz akış tipi çizelgeleme üzerine bir not	Matematiksel ispat	Maksimum tamamlanma zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Aydilek ve Allahverdi (2012)	Ortalama maksimum tamamlanma zamanına bağlı, maksimum tamamlanma zamanı ile beklemesiz akış atölyeleri için sezgiseller	Hibrit	Ortalama ve Maksimum tamamlanma zamanı
Czogalla ve Fink (2012)	Beklemesiz akış tipi çizelgeleme problemi için uyum yüzeyi analizi	Uyum yüzeyi analizi	Toplam akış zamanı
Jolai ve ark. (2012)	Sıra bağımlı kurulum süreleriyle beklemesiz esnek bir akış tipi çizelgeleme problemi için yeni bir hibrit meta-sezgisel algoritma	Hibrit	Maksimum tamamlanma zamanı
Muthuswamy ve ark. (2012)	Toplu imal makineleri ile iki makineli, beklemesiz akış atölyesinde, maksimum tamamlanma zamanının azaltılması	Parçacık sürü optimizasyonu	Maksimum tamamlanma zamanı
Nagano ve ark. (2012)	Kurulum zamanları ile beklemesiz akış tipi problemi için yeni bir evrimsel kümeleme araştırması	Küme arama algoritması	Toplam akış zamanı
Samarghandi ve ElMekawy (2012a)	Ayrılabilir kurulum süreleri ve maksimum tamamlanma zamanı kriterleri ile beklemesiz akış tipi çizelgeleme problemi için genetik algoritma ve parçacık sürü optimizasyonu	Hibrit	Maksimum tamamlanma zamanı
Samarghandi ve ElMekawy (2012b)	Beklemesiz akış atölyesi problemini çözmek için meta-sezgisel yaklaşım	Hibrit	Maksimum tamamlanma zamanı
Allahverdi ve Aydilek (2013)	Maksimum tamamlanma zamanına bağlı toplam tamamlanma süresi ile beklemesiz akış atölyeleri için algoritmalar	Hibrit	Maksimum tamamlanma zamanı
Davendra ve ark. (2013)	Maksimum tamamlanma zamanı ile beklemesiz akış tipi çizelgeleme için ayırık kendini düzenleyen göç algoritması	Göç algoritması	Maksimum tamamlanma zamanı
Gao ve ark. (2013)	Toplam akış süresini en aza indirmek amacıyla, beklemesiz akış tipi çizelgeleme problemi için etkili sezgisel	Hibrit	Toplam akış zamanı
Liu ve ark. (2013)	Toplam gecikme kriterleri olan, beklemesiz akış atölyeleri için bazı sezgiseller	Hibrit	Toplam gecikme
Moradinasab ve ark. (2013)	Genetik ve uyarlanabilir yayılımcı rekabet algoritmaları ile beklemesiz iki aşamalı hibrit akış tipi çizelgeleme	Hibrit	Toplam tamamlanma zamanı
Sapkal ve Laha (2013)	Beklemesiz akış tipi çizelgeleme için bir sezgisel	Yeni sezgisel	Toplam akış zamanı
Wang ve Liu (2013)	İki aşamalı, beklemesiz hibrit akış tipi çizelgeleme problemi için genetik bir algoritma	Genetik algoritma	Toplam tamamlanma zamanı
Akhshabi ve ark. (2014)	Toplam akış süresi ile beklemesiz bir akış tipi çizelgeleme problemi için hibrit parçacık sürü optimizasyon algoritması	Hibrit	Toplam akış zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Allahverdi ve Aydilek (2014)	Hazırlık süreleri olan, beklemesiz akış tipinde, maksimum tamamlanma zamanı kısıtlaması ile toplam tamamlanma zamanı	Yerleştirme algoritması, Genetik algoritma, Tavlama benzetimi, Bulut teorisine dayalı tavlama benzetimi, Diferansiyel evrim algoritması	Toplam ve Maksimum tamamlanma zamanı
Asefi ve ark. (2014)	İki amaçlı, beklemesiz esnek akış tipi çizelgeleme problemini çözmek için bir hibrit NSGA-II ve VNS	Hibrit	Ortalama gecikme
Laha ve ark. (2014)	Beklemesiz akış atölyelerinde, toplam akış süresini en aza indirmek için ceza-değiştirme-yerleştirme tabanlı bir algoritma	Vogel sezgiseli	Toplam akış zamanı
Laha ve Sapkal (2014)	m -makinelik, beklemesiz akış tipi çizelgeleme için toplam akış süresini en aza indirgeyen gelişmiş bir sezgisel	Yeni sezgisel	Toplam akış zamanı
Nagano ve ark. (2014)	Sıraya bağlı kurulum süreleriyle, beklemesiz akış atölyesi problemi için evrimsel bir kümeleme araştırması	Hibrit	Toplam akış zamanı
Rabiee ve ark. (2014)	Bağılantısız paralel makinelerle beklemesiz iki aşamalı esnek akış tipi çizelgeleme problemini çözmek için akıllı bir hibrit meta-sezgisel	Hibrit	Maksimum tamamlanma zamanı
Ren ve ark. (2014)	Serbest bırakma tarihleriyle beklemesiz akış tipi çizelgeleme problemi için SPTA tabanlı algoritmaların asimptotik analizi	SPTA tabanlı algoritmalar	Toplam tamamlanma zamanı
Samarghandi ve ElMekkawy (2014)	Beklemesiz akış tipi problemini sıraya bağlı kurulum süreleriyle çözme	Parçacık sürü optimizasyonu	Maksimum tamamlanma zamanı
Shabtay ve ark. (2014)	Beklemesiz kısıtlama ile akış atölyesi için birleşik robot seçimi ve çizelgeleme problemi	En kısa yol yöntemi	Maksimum tamamlanma zamanı ve Robot seçim maliyeti
Araújo ve Nagano (2015)	Sıra bağımlı kurulum zamanı problemi ile beklemesiz akış atölyesi için yeni sezgiseller	Gap ve Quarts sezgiseli	Maksimum Tamamlanma Zamanı ve Toplam Akış Zamanı
Deng ve ark. (2015)	Beklemesiz akış tipi çizelgeleme problemi için etkili bir eş-evrimsel kuantum genetik algoritması	Evrimsel kuantum genetik algoritması	Maksimum tamamlanma zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Ding ve ark. (2015a)	Beklemez akış tipi çizelgeleme problemi için tabu tabanlı, yeniden yapılanma stratejisi ile geliştirilmiş yinelenen, açgözlü algoritma	Hibrit	Maksimum tamamlanma zamanı
Ding ve ark. (2015b)	Beklemez akış atölyelerinde, toplam gecikmenin en aza indirilmesi için hızlandırılmış yöntemler	Hibrit	Toplam gecikme
Nagano ve ark. (2015)	Sıra bağımlı kurulum süreleriyle, beklemez bir akış atölyesinde, toplam akış süresi minimizasyonu için yapıcı bir sezgisel	Quarts sezgiseli	Toplam akış zamanı
Ramezani ve ark. (2015)	Tek tip paralel makineler ve sıraya bağlı kurulum süresi ile beklemez esnek akış tipi: hibrit bir meta-sezgisel yaklaşım	Hibrit	Maksimum tamamlanma zamanı
Samarghandi (2015a)	Sıraya bağlı kurulum süreleri ve sunucu kısıtlamaları olan beklemez bir akış tipi sistem	Genetik algoritma	Maksimum tamamlanma zamanı
Samarghandi (2015b)	Sıraya bağlı kurulum süreleriyle beklemez akış tipi probleminin maksimum tamamlanma zamanı üzerindeki sunucu yan kısıtların etkisinin incelenmesi	Genetik algoritma	Maksimum tamamlanma zamanı
Samarghandi (2015c)	Son tarih kısıtlamaları olan beklemez akış atölyesi problemi için parçacık sürüsü optimizasyonu	Parçacık sürü optimizasyonu	Maksimum Tamamlanma Zamanı
Wang ve ark. (2015)	İki aşamalı, beklemez hibrit akış tipi çizelgeleme için dal-sınır algoritması	Dal-sınır algoritması	Maksimum tamamlanma zamanı
Laha ve Gupta (2016)	Beklemez akış tipinde, maksimum tamamlanma zamanı ve toplam akış süresini minimize etmek için macar cezasına dayanan, yapısal bir algoritma	Hibrit	Maksimum tamamlanma zamanı ve Toplam akış zamanı
Nagano ve Miyata (2016a)	Beklemez akış tipi problemi için yapıcı sezgisel mekanizmaların gözden geçirilmesi ve sınıflandırılması	İnceleme ve sınıflandırma çalışması	
Nagano ve Miyata (2016b)	Beklemez akış tipi çizelgeleme problemi için yüksek kaliteli, bir yapıcı sezgisel çözümü	Yeni sezgisel	Maksimum tamamlanma zamanı
Qi ve ark. (2016)	Toplam akış süresini en aza indirme ile beklemez akış tipi çizelgeleme için hızlı yerel komşu arama algoritması	Yerel komşu arama algoritması	Toplam akış zamanı
Qin ve Zhang (2016)	İki kriterli, beklemez esnek akış atölyesi problemini çözmek için seçkin parçacık sürü optimizasyonu algoritması	Parçacık sürü optimizasyonu	Maksimum tamamlanma zamanı ve Maksimum gecikme

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Rabiee ve ark. (2016)	Ortalama gecikmeyi en aza indirmek için bağlantısız paralel makinelerle, gerçeğe uygun, beklemesiz hibrit akış mağazası için biyocoğrafyaya dayalı bir optimizasyon algoritması	Biyocoğrafyaya dayalı optimizasyon	Ortalama gecikme
Samarghandi ve Behroozi (2016)	Teslim tarih kısıtlamalı, beklemesiz akış tipi problemi için bir numaralandırma algoritması	Numaralandırma algoritması	Maksimum tamamlanma zamanı
Ye ve ark. (2016)	Beklemez akış tipi üretimde, maksimum tamamlanma zamanını minimuma indirmek için etkili bir sezgisel yöntem	Ortalama kalkış zamanı sezgiseli	Maksimum tamamlanma zamanı
Ying ve ark. (2016)	Beklemez akış atölyelerinde, toplam akış süresini en aza indirmek için kendinden uyarlamalı yıkma ve yeniden oluşturma algoritması	Uyarlanabilir yıkma ve Yeniden oluşturma algoritması	Toplam akış zamanı
Bewoor ve ark. (2017)	NP-zor beklemez akış tipi çizelgeleme problemlerini çözmek için evrimsel hibrit parçacık sürü optimizasyonu algoritması	Hibrit	Toplam akış zamanı
Jafarzadeh ve ark. (2017)	Düzeltilmiş çok amaçlı, yabancı ot optimizasyonu ve bulanık baskınlık yaklaşımıyla bağlantısız paralel makinelerde, yeniden işleme süresi ile beklemez iki aşamalı esnek akış tipi çizelgeleme problemini çözme	Yabancı ot optimizasyonu	Maksimum tamamlanma zamanı ve Ortalama gecikme
Komaki ve Malakooti (2017)	Dağıtılmış, beklemez akış tipi çizelgeleme probleminin, maksimum tamamlanma zamanını en aza indiren, genel değişken komşu arama algoritması	Değişken komşu arama	Maksimum tamamlanma zamanı
Samarghandi ve Behroozi (2017)	Teslim tarihi kısıtlaması ile beklemez akış tipi probleminin kesin çözümü	Matematiksel model	Maksimum tamamlanma zamanı
Shao ve ark. (2017a)	Beklemez akış tipi çizelgeleme problemini çözmek için genişletilmiş, öğretim-öğrenme tabanlı optimizasyon algoritması	Hibrit	Maksimum tamamlanma zamanı
Shao ve ark. (2017b)	Yinelenen ağgözlü algoritmalarla, dağıtılmış beklemez akış tipi çizelgeleme problemi için maksimum tamamlanma zamanı optimizasyonu	Yinelenen ağgözlü algoritma	Maksimum tamamlanma zamanı
Ye ve ark. (2017a)	Beklemez akış atölyesinde, maksimum tamamlanma zamanını en aza indirmek için geliştirilmiş bir sezgisel	Ortalama boş zaman sezgiseli	Maksimum tamamlanma zamanı
Ye ve ark. (2017b)	Toplam tamamlanma zamanını en aza indirmek amacıyla beklemez akış tipi üretim için etkili ve verimli sezgisel bir yöntem	Mevcut ve gelecekteki boşta kalma sezgiseli	Toplam tamamlanma zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Chaudhry ve ark. (2018)	Genetik algoritma kullanarak beklemesiz akış atölyesinde, toplam akış süresinin azaltılması	Genetik algoritma	Toplam akış zamanı
Engin ve Güçlü (2018)	Beklemez akış tipi çizelgeleme problemlerini çözmek için yeni bir hibrit karınca kolonisi optimizasyon algoritması	Hibrit	Maksimum tamamlanma zamanı
Gao ve ark. (2018)	Öğrenme etkisi, ortak bitiş tarihi ve kontrol edilebilir iş işleme süreleri ile beklemez iki makineli permütasyon akış tipi çizelgeleme problemi	Ortak teslim tarihi atama yöntemi	Toplam erkenlik, Toplam gecikme, Toplam kaynak maliyeti,
Koulamas ve Panwalkar (2018)	Beklemez akış atölyeleri için yeni endeks öncelik kuralları	Öncelik kuralı	Maksimum tamamlanma zamanı
Labıdı ve ark. (2018)	İki makineli, beklemez akış tipi çizelgeleme probleminin, eşit olmayan teslim tarihleri ve uygun olmama kısıtlamalarını hesaplama çalışması	Dal-sınır algoritması	Maksimum tamamlanma zamanı
Li ve ark. (2018)	Sıra bağımlı kurulum süreleri, öğrenme ve unutkanlık etkileri ile beklemez akış atölyeleri için yinelenen ağgözlü sezgiseli	Yinelenen ağgözlü algoritma	Toplam akış zamanı
Riahi ve Kazemi (2018)	Beklemez akış atölyesinin programlanması için yeni bir hibrit karınca kolonisi algoritması	Hibrit	Maksimum tamamlanma zamanı
Zhao (2018)	Beklemez akış tipi çizelgeleme problemi için ayrık bir Su Dalgası Optimizasyonu algoritması	Su dalgası optimizasyonu	Maksimum tamamlanma zamanı
Bewoor ve ark. (2019)	Beklemez akış tipinde, maksimum tamamlanma zamanını minimize etmek için evrimsel hibrit kuş sürüsü optimizasyon algoritması	Hibrit	Maksimum tamamlanma zamanı
Geng ve ark. (2019)	Dışbükey kaynak tahsisi ve öğrenme etkisi ile beklemez bir akış tipi için ortak teslim tarih atama planlaması	$O(n^3)$ ve $O(n \log n)$	Toplam gecikme ve Toplam kaynak tüketim maliyeti
Lin ve Ying (2019)	Sıraya bağlı kurulum süreleri ile beklemez akış tipi üretim hücresinde maksimum tamamlanma zamanı optimizasyonu	Matematiksel model	Maksimum tamamlanma zamanı
Miyata ve ark. (2019a)	Toplam akış süresi minimizasyonu ile m makineli, beklemez akış tipi çizelgeleme problemine önleyici bakım uygulanması: hesaplamalı bir çalışma	Laha ve Sapkal sezgiseli (LSH) ve Framinan, Nagano ve Moccellini sezgiseli (FNMH)	Toplam akış zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Miyata ve ark. (2019b)	Önleyici bakım faaliyetlerini, sıraya bağlı kurulum süreleri ve üretim süresi en aza indirmeli, beklemesiz akış tipi çizelgeleme problemi	Li, Wang ve Wu (LWW) sezgiseli, Hızlı kompozit sezgiseli (FCH)	Maksimum tamamlanma zamanı
Shao ve ark. (2019)	Sıraya bağlı kurulum zamanı ile çok amaçlı dağıtılmış beklemesiz akış tipi çizelgeleme probleminin çözümü için pareto tabanlı dağıtım algoritmasının tahmini	Pareto tabanlı dağıtım algoritması	Maksimum tamamlanma zamanı ve Toplam ağırlıklı gecikme
Sun ve ark. (2019)	Ortak akış ödeneği ve öğrenme etkisi ile beklemesiz akış tipi ortamda, dışbükey kaynak tahsisi planlaması	Matematiksel model	Toplam ağırlıklı erkenlik ve gecikme, Kaynak tüketim maliyeti
Taşgetiren ve ark. (2019)	Enerji açısından verimli beklemesiz akış tipi çizelgeleme problemi için ayırık yapay arı kolonisi algoritması	Arı koloni algoritması	Toplam akış zamanı ve Toplam enerji tüketimi
Urgo (2019)	Kalan iş içeriğinin risk koşullu değerini en aza indirmeye yönelik, beklemesiz bir akış tipi planlamak için dal-sınır yaklaşımı	Dal-sınır algoritması	İş yükü miktarı
Ye ve ark. (2019)	Beklemesiz akış tipi üretim için maksimum ve toplam tamamlanma süreleri arasında takas dengesi	Takas dengeleme sezgiseli	Maksimum tamamlanma zamanı ve Toplam tamamlanma zamanı
Zhao ve ark. (2019a)	Maksimum tamamlanma zamanı amacıyla beklemesiz akış tipi çizelgeleme problemi için bir popülasyon adaptasyon mekanizmasıyla birlikte faktöriyel bazlı bir parçacık sürü optimizasyonu	Parçacık sürü optimizasyonu	Maksimum tamamlanma zamanı
Zhao ve ark. (2019b)	Faktöriyel gösterim ile beklemesiz akış tipi çizelgeleme problemi için uyum yüzeyi analizi	Uyum yüzeyi analizi	Maksimum Tamamlanma Zamanı
Zhao ve ark. (2019c)	Beklemesiz akış tipi çizelgeleme problemi için değişken komşu arama mekanizmasına sahip, karma biyocoğrafya dayalı optimizasyon	Biyocoğrafyaya dayalı optimizasyon	Maksimum tamamlanma zamanı
Zhao ve ark. (2019d)	Beklemesiz akış tipi çizelgeleme problemi için tek dalga mekanizması ile geliştirilmiş su dalgası optimizasyon algoritması	Su dalgası optimizasyonu	Maksimum tamamlanma zamanı

Çizelge 1. (devamı)

Yazar-Yıl	Çalışma Başlığı	Kullanılan Yöntem	Amaç Fonksiyonu
Zhu ve ark. (2019)	Beklemesiz akış tipi çizelgeleme için kuantumdan esinlenen, guguk kuşunun eş evrimsel algoritması	Guguk kuşu arama algoritması	Maksimum tamamlanma zamanı
Allahverdi ve ark. (2020)	Maksimum tamamlanma zamanına bağlı olarak toplam gecikmeyi en aza indirmek için ayrı kurulum sürelerine sahip, beklemesiz akış tipi çizelgeleme problemi	Hibrit	Toplam gecikme
Chen ve ark. (2020)	Beklemesiz ve belirleyici kullanılmayan aralık kısıtlamaları altında iki makineli akış tipi çizelgeleme probleminde, maksimum tamamlanma zamanı minimizasyonu	Gilmore ve Gomory algoritması	Maksimum tamamlanma zamanı
Ferrettia ve Zavanella (2020)	Genel akış tipi çizelgeleme problemi için beklemesiz / engelleme kısıtlamaları ile toplu enerji planlama	Genetik algoritma	Toplam enerji üretim maliyeti
Schaller ve Valente (2020)	Beklemesiz akış tipi ortamda, toplam gecikmeyi en aza indirme	Dal-sınır algoritması ve Yeni sezgisel yöntem	Toplam gecikme
Allahverdi ve ark. (2021)	Maksimum gecikme performans ölçütü ile iki makineli beklemesiz akış hatları için daha iyi bir baskınlık ilişkisi ve sezgiseller	Yapıcı sezgisel yöntemler	Maksimum gecikme
Başar ve Engin (2021)	Bulanık ortamda, hazırlık süreli beklemesiz akış tipi çizelgeleme problemi	Geliştirilmiş dağılık arama	Maksimum tamamlanma zamanı ve Tatmin indeksi
Li ve ark. (2021)	Dağıtılmış heterojen, beklemesiz akış tipi çizelgeleme problemi için ayrık yapay arı koloni algoritması	Arı koloni algoritması	Maksimum tamamlanma zamanı
Keskin ve Engin (2021)	Beklemesiz akış tipi çizelgeleme problemini iki kriterle çözmek için hibrit bir genetik yerel ve küresel arama algoritması	Hibrit	Toplam akış zamanı ve Maksimum tamamlanma zamanı
Koulamas ve Kyparisis (2021)	Reddedilme ile beklemesiz akış tipi çizelgeleme	Dinamik programlama	Maksimum tamamlanma zamanı
Zhu ve ark. (2021)	Beklemesiz akış tipi çizelgeleme için kuantumdan esinlenilmiş guguk kuşu eş evrimsel algoritması	Guguk kuşu arama algoritması	Maksimum tamamlanma zamanı

BATÇ ile ilgili yapılan araştırmalar incelendiğinde, son yıllarda yapılan çalışmalarda önemli bir artış görülmektedir. Kaynak araştırmasındaki çalışmaların dağılımı göz önüne alındığında, son yedi yıl içerisinde yapılan çalışmalar önceki yirmi yıldan daha fazladır. Artan küresel rekabetten dolayı gelecek yıllar içerisinde BATÇ üzerine yapılacak çalışmaların artış göstermesi beklenmektedir.

Kaynak araştırması sonucunda, BATÇ problemlerinin çözümü için en fazla hibrit yöntemlerin kullanıldığı görülmüştür. Öyle ki, BATÇ için yapılmış doksan dört çalışmanın yirmi beş tanesinde hibrit yöntem kullanımına başvurulmuştur. BATÇ ortamlarında, makine sayısının ikiden fazla olması sorunu, polinomial olarak çözülemeyen zor bir problem haline getirmektedir. Bu yüzden, matematiksel modeller standart bir süre içerisinde global optimum sonuca yaklaşmakta yetersiz kalmaktadır. Bu sorunun üstesinden gelebilmek için sezgisel ve meta-sezgisel yöntemlerin kullanımına başvurulmuştur. Literatürde yapılmış olan çalışmalarda, sezgisel ve meta-sezgisel yöntemlerin standart bir işlem süresinde global optimuma yakın sonuçların elde edilebileceği vurgulanmıştır. Hibrit yöntemler tek başına yeterli iyileşme sağlayamayan sezgisel yöntemlerde, birden fazla sezgiselin güçlü yönlerinin birbirine uyarlanarak daha verimli sonuçlar elde edilmesi amacıyla kullanılmaktadır. Bu durum, yapılan çalışmalarda daha fazla hibrit yaklaşımın tercih edilme sebebi olmuştur. BATÇ üzerine yapılmış olan literatür taramasında da hibrit yöntem kullanımının son on yılda oldukça fazla olduğu görülmektedir. Hibrit yaklaşımlar içerisinde en fazla genetik algoritma, parçacık sürü optimizasyonu, tavlama benzetimi ve NEH algoritması kullanımına başvurulmuştur.

Matematiksel çözüm yöntemlerinin NP-zor sınıfta yer alan BATÇ problemlerine karşı yetersiz kalmasından dolayı, çözüm için sezgisel ve meta-sezgisel yöntemlerin kullanımına başvurulmuştur. Genetik Algoritma ise bu yöntemlerin başında gelmektedir. Genetik Algoritma birçok probleme uyarlanabilir olması, çözüm süresini makul seviyelere düşürebilmesi ve optimuma oldukça yakın sonuç vermesinden dolayı kullanımı oldukça yaygın hale gelmiştir. Yapılmış olan kaynak araştırmasında, tek başına en fazla kullanılan çözüm yöntemi Genetik Algoritmadır. Öyle ki, BATÇ için yapılmış doksan dört çalışmanın yedi tanesinde, Genetik Algoritma kullanımına başvurulmuştur.

Dal Sınır yöntemi ve diğer matematiksel çözüm yöntemlerinin kullanımına bakıldığında, üç ve üçten fazla makine içeren problemlerde kullanımının yaygın olmadığı görülmektedir. Bunun sebebi ise BATÇ problemlerinde makine sayısı üç ve üçten fazla olduğu durumlarda NP-zor sınıfına girmektedir. Bundan dolayı Dal Sınır yöntemi ve matematiksel çözüm yöntemleri, polinomial olarak çözülemeyen problemlerde çözüm süresi çok fazla artış göstermekte ve standart bir süre içerisinde optimum sonuç verememektedir. BATÇ kaynak araştırması sonucunda, doksan dört çalışma içerisinde beş adet çalışmada Dal Sınır yöntemi kullanıldığı görülmüştür.

BATÇ problemlerinde Uyum Yüzeyi Analizi ve Karınca Koloni Optimizasyon algoritmasının kullanımına bakıldığı zaman oldukça az çalışma yapıldığı görülmektedir. Öyle ki, incelenen araştırmalar içerisinde sadece iki adet çalışma yapıldığı görülmüştür. Karınca Koloni Optimizasyon algoritmasının, NP-zor problemler için makul çözüm süresinde, optimuma yakın sonuçlar verdiği bilinmektedir. Lakin BATÇ problemlerinin çözümü için incelenen çalışmalar içerisinde 2018 yılına kadar yapılmış çalışma bulunmamaktadır. 2018 yılında ise Engin ve Güçlü, Riahi ve Kazemi tarafından iki tane çalışma yapılmıştır.

Kaynak araştırması sonucunda, BATÇ üzerine yapılmış olan doksan dört çalışma incelenmiştir ve Parçacık Sürü Optimizasyon yönteminin beş adet, Arı Koloni algoritmasının iki adet, Biyocoğrafyaya Dayalı Optimizasyon yönteminin iki adet, Gilmore ve Gomory algoritmasının iki adet çalışmada, çözüm yöntemi olarak kullanıldığı görülmüştür. Diğer matematiksel, sezgisel ve meta-sezgisel yöntemlerin ise birer çalışmada çözüm yöntemi olarak kullanıldığı görülmüştür.

İncelenen çalışmalar içerisinde, dağıtılmış BATÇ problemi son yıllarda çalışılmaya başlanmış güncel konulardan birisidir. Dağıtılmış üretim hatları, modern tedarik zincirlerinde ve üretim sistemlerinde yaygın olarak bulunmaktadır. Bu yüzden, son yıllarda dağıtılmış BATÇ problemleri üzerinde daha fazla durulmaya başlanmıştır. BATÇ problemlerinin gerçek hayatta önemli bir kısmını, esnek BATÇ ortamları oluşturmaktadır. İncelenen çalışmalara göre, esnek BATÇ en sık çalışılmış akış tipi çizelgeleme konularından birisidir. Özellikle son yıllarda daha çok çalışılmaya başlanmış ve gerçek hayatta en çok karşılaşılan problemlerdendir.

Maksimum tamamlanma zamanı performans ölçüsü, BATÇ problemlerinin en önemli kriteridir. Çünkü maksimum tamamlama zamanının minimize edilmesi, işlerin tam zamanında teslimi için büyük önem taşımaktadır. Maksimum tamamlanma zamanı performans kriteri, BATÇ problemlerinde oldukça fazla çalışılmıştır. Yapılan kaynak araştırmasına göre doksan dört çalışma içerisinde kırk sekiz tanesi, maksimum tamamlanma zamanını amaç fonksiyonu olarak ele almıştır. Diğer bir ifade ile incelenen çalışmaların yarısından fazlasını oluşturmaktadır.

Akış zamanı bir ürünün ortaya çıkış sürecindeki dönüşümün toplam zamanıdır. Bir veya birçok sürecin içinden geçen bir ürünün yolculuk süresi akış zamanı olarak nitelendirilir. BATÇ probleminin en önemli sorunlarından birisi olan toplam akış zamanı minimizasyonu, son on yılda üzerinde çokça durulmaya başlanmıştır. İncelenen çalışmalar içerisinde on altı adet çalışma, toplam akış zamanı amaç fonksiyonu ile ele

alınmıştır. Maksimum tamamlanma zamanı amaç fonksiyonundan sonra en çok çalışılmış olan amaç fonksiyonunun, toplam akış zamanı olduğu görülmüştür.

Bir ürünün zamanında teslimi, müşteri memnuniyeti ve küresel anlamda diğer firmalar ile rekabet açısından çok önemlidir. İş bitim tarihleri ile tamamlanma süreleri arasındaki boşluklar artıkça toplam gecikme artar. Bunun için BATÇ problemlerinde gecikmenin minimizasyonu gereklidir. Kaynak araştırması sonucunda ise doksan dört çalışma içerisinde dokuz tanesinin toplam gecikme kriterini amaç fonksiyonu olarak ele aldığı görülmüştür.

İki ya da daha fazla amaç fonksiyonunun eş zamanlı olarak optimize edildiği çok kriterli BATÇ problemleri, gerçek yaşam koşullarına uygunluk açısından önem arz etmektedir. İncelenen çalışmalar içerisinde on beş adet çalışmanın çok kriterli olarak çalışıldığı görülmüştür. Çok kriterli BATÇ çalışmalarının son dokuz yıl içerisinde çalışılmış olması dikkat çekmektedir ve gelecek yıllar içerisinde çok kriterli BATÇ çalışmalarında artış olması muhtemeldir.

Toplam tamamlanma zamanı beş adet, ortalama gecikme üç adet, maksimum gecikme iki adet çalışmada amaç fonksiyonu olarak ele alınmıştır.

Yapılmış olan kaynak araştırmasında, çalışmaların çoğunluğu teorik olarak çalışılmıştır. Çalışmalarda en çok kullanılan kıyas problemleri ise Taillard ve Reeves'in test problemleridir.

2.2. Dağılık Arama Yöntemi Kaynak Araştırması

Bu bölümde, DAY üzerine yapılmış olan çalışmalar, kronolojik sıraya göre Çizelge 2'de sunulmuştur. Çizelge 2'de ki, araştırılan problem bilgisi, DAY'nin hangi problemler için çözüm yöntemi olarak kullanıldığını belirtmektedir.

Çizelge 2. Kronolojik sıraya göre DAY kaynak araştırması

Yazar-Yıl	Çalışma Başlığı	Araştırılan Problem
Nowicki ve Smutnicki (2006)	Akış tipi çizelgelemede dağılık aramaya bir takım bakış açıları	Akış tipi çizelgeleme
Oktay ve Engin (2006)	Endüstriyel problemlerin çözümünde dağılık arama yöntemi: literatür araştırması	Literatür araştırması
Sagarna ve Lozano (2006)	Dağıtım algoritmalarının tahmini ile yazılım testi, karşılaştırma ve iş birliğinde dağılık arama	Dağılık arama ve dağıtım algoritmalarının yazılım testi

Çizelge 2. (devamı)

Yazar-Yıl	Çalışma Başlığı	Araştırılan Problem
Bożejko ve Wodecki (2007)	Akış tipi çizelgeleme problemi için paralel dağılık arama algoritması	Akış tipi çizelgeleme
Haq ve ark. (2007)	Genel akış tipi çizelgeleme problemi için bir dağılık arama yaklaşımı	Akış tipi çizelgeleme
Rahimi-Vahed ve ark. (2008)	İki kriterli beklemesiz akış tipi çizelgeleme problemi için çok amaçlı bir dağılık arama	Beklemesiz akış tipi çizelgeleme
Rao ve Lakshmi (2008)	Kombinatoriyel optimizasyon için çok amaçlı dağılık arama algoritması	Kombinatoriyel optimizasyon problemleri
Saravanan ve ark. (2008)	Permütasyon akış tipi çizelgeleme problemleri için dağılık arama yönteminin performans değerlendirmesi	Permütasyon akış tipi çizelgeleme
Sonuç (2008)	İki boyutlu HP modelinde protein katlanma problemlerine dağılık arama yaklaşımı	Protein katlanma problemi
Engin ve ark. (2009)	Çok amaçlı bulanık permütasyon akış tipi çizelgeleme problemi için bir dağılık arama yöntemi: gerçek bir yaşam uygulaması	Permütasyon akış tipi çizelgeleme
Mansour ve ark. (2009)	Sınav çizelgeleme için dağılık arama tekniği	Sınav çizelgeleme
Tang ve ark. (2010)	Yükleme maliyeti ile araç rotalama problemini çözmek için bir dağılık arama algoritması	Araç rotalama problemi
Kaya ve ark. (2011)	Erken ve geç cezalı permütasyon akış tipi çizelgeleme problemlerinin dağılık arama meta sezgiseli ile çözümü	Permütasyon akış tipi çizelgeleme
Nasiri ve Kianfar (2011)	Kısmi atölye çizelgeleme problemi için hibrit bir dağılık arama	Kısmi atölye çizelgeleme problemi
Sadiq ve Mohammed (2012)	Atölye tipi çizelgeleme için iyileştirilmiş dağılık arama	Atölye tipi çizelgeleme
Engin ve ark. (2013)	Kullanılabilirlik kısıtlamaları ile bulanık atölye çizelgeleme problemi için bir dağılık arama yöntemi	Kullanılabilirlik kısıtlı atölye tipi çizelgeleme problemi
Pendharkar (2013)	Kömür üretim planlaması için bulanık amaçların etkileşim tabanlı dağılık arama ile çok kriterli optimizasyonu	Kömür üretim planlama problemi
Benavides ve ark. (2014)	Heterojen işçilerle akış tipi çizelgeleme	Akış tipi çizelgeleme problemi
Naderi ve Ruiz (2014)	Dağıtılmış permütasyon akış tipi çizelgeleme problemi için bir dağılık arama algoritması	Dağıtılmış permütasyon akış tipi çizelgeleme problemi
Gonzales ve ark. (2015)	Esnek atölye tipi çizelgeleme problemi için yeniden yol bağlama ile dağılık arama	Esnek atölye tipi çizelgeleme
Guo ve Tang (2015)	Sıra bağımlı hazırlık süreleri ile tek makine toplam ağırlıklı gecikme çizelgeleme problemi için geliştirilmiş bir dağılık arama algoritması	Tek makine çizelgeleme problemi
Çiçekli ve Bozkurt (2016)	Permütasyon akış tipi çizelgeleme probleminin dağılık arama ile optimizasyonu	Permütasyon akış tipi çizelgeleme
Külahlı (2016)	Esnek atölye tipi çizelgeleme problemleri için bir dağılık arama metodu	Esnek atölye tipi çizelgeleme problemleri
Riahi ve ark. (2017)	Karışık engellemeli akış tipi çizelgeleme için dağılık arama	Akış tipi çizelgeleme problemi

Çizelge 2. (devamı)

Yazar-Yıl	Çalışma Başlığı	Araştırılan Problem
Yang ve ark. (2017)	Toplam gecikmeyi en aza indirmek için dağıtılmış akış tipi çizelgelemede dağınık arama	Dağıtılmış akış tipi çizelgeleme problemi
Alsaidi ve ark. (2020)	NP-zor problemleri çözmek için meerkat klan algoritmasına dayalı geliştirilmiş dağınık arama algoritması	Gezgin satıcı problemi ve Esnek iş çizelgeleme problemi
Franca-Vieira ve ark. (2020)	Dağınık arama meta-sezgiseli ile bir koniğin ters tasarımı	Fotonik yapıların optimizasyonu
Tan ve ark. (2020)	Optimal ve enerji tasarruflu çelik üretimi-sürekli döküm için hibrit dağınık arama algoritması	Çelik üretiminde sürekli döküm çizelgeleme problemi
Başar ve Engin (2021)	Bulanık ortamda hazırlık süreli beklemesiz akış tipi çizelgeleme problemi	Beklemesiz akış tipi çizelgeleme

Çizelge 2’de özetlenmiş olan DAY kaynak araştırmasında, DAY’nin çizelgeleme problemleri için çokça kullanıldığı görülmektedir. İncelenmiş olan çalışmaların çoğunluğunda çizelgeleme problemleri için DAY’nin etkin sonuçlar verdiği vurgulanmıştır. DAY’nin etkinliğini artırmak için literatürde yer alan diğer sezgisel ve meta-sezgisel yöntemler ile birlikte kullanımının standart halinden daha yoğun olması dikkat çekmektedir. Yani DAY’nin diğer sezgisel yöntemler ile birlikte hibrit bir şekilde ya da standart dağınık arama prosedürünün farklı teknikler kullanılarak iyileştirilmiş şekilde kullanımının daha etkin sonuçlar verdiği yapılmış olan çalışmaların çoğunluğunda vurgulanmıştır.

3. MATERYAL VE YÖNTEM

3.1. Materyal

Bu çalışmada, materyal, BATÇ problemleridir. BATÇ problemlerinin notasyonu, gösterimi, matematiksel modeli aşağıda sunulmuştur.

3.1.1. BATÇ Problemlerinin Notasyonu

BATÇ problemlerinin tümünde, işler ve işlerin işlenmesi için makineler kümesi bulunmaktadır. Problemleri çözerken, kümeler ve kümedeki elemanların çizelgeleme problemi cinsinden ifade edilmesi gerekir. BATÇ problemi için gereken notasyon ve tanımlar aşağıda sunulmuştur.

$i=(1,2,\dots,m)$: m adet makine içeren makineler kümesini, $j=(1,2,\dots,n)$: n adet makinelerde işlem görmesi gereken işler kümesini temsilen;

- p_{ij} : j işinin i makinesindeki işlem süresini,
- r_j : j işinin sisteme geliş zamanını,
- d_j : j işinin teslim tarihini,
- w_j : j işinin diğer işlere göre ağırlığını diğer bir ifade ile önceliğini,
- C_j : j işinin tamamlanma zamanını yani teslim hazır olduğu zamanı,
- F_j : $C_j - r_j$: j işi için akış süresini yani o işin sistemde geçirdiği zamanı,
- L_j : $C_j - d_j$: j işi için gecikme süresini,
- T_j : $\max(C_j - d_j, 0)$: Tüm işler içerisinde ki en fazla gecikme süresini ifade eder.

3.1.2. BATÇ Problemlerinin Gösterimi

Çizelgeleme problemlerinde ortam koşulları, işlem özellikleri ve amaç fonksiyon bilgisi $\alpha | \beta | \gamma$ şeklinde bir gösterim ile ifade edilir. Maksimum tamamlanma zamanı amaç fonksiyonu ile BATÇ problemini ifade etmek için $F_m | no-wait | C_{max}$ notasyonları kullanılır. Bu notasyonların tanımı aşağıda sunulmuştur.

F_m : Akış tipinde üretimin olduğu ortamı yani m adet makinede işlerin tamamının aynı sırayı takip ettiği durumu ifade eder.

no-wait: Beklemesiz olma durumunu yani, işlerin kesinti olmaksızın işlem görmek zorunda olduğu durumu ifade eder.

C_{max} : Amaç fonksiyonu olan maksimum tamamlanma zamanının minimize edilmesi gerektiği durumu tanımlar.

BATÇ problemleri için yaygın olarak kullanılan amaç fonksiyonları şu şekildedir;

- C_{max} : Maksimum tamamlanma zamanı
- L_{max} : Maksimum gecikme
- TT : Toplam gecikme

3.1.3. BATÇ Problemlerinin Matematiksel Modeli

Akış tipi ortamında, yapılması gereken işler, aynı görev listesine sahip ise, yani işler aynı makinelerde aynı sırada işlem görüyor ise, bu ortam akış tipi olarak adlandırılır. ATÇ problemlerinin bir kısmı, beklemesiz kısıtı ile sınıflandırılır. Beklemesiz olarak adlandırılan akış tipi ortamında işlerin beklemesine izin verilmeksizin sürekli olarak işlem görmek zorundadır.

Maksimum tamamlanma zamanı amaç fonksiyonu ile BATÇ probleminin matematiksel modeli, denklem (1, 2, 3) ile aşağıdaki şekilde tanımlanmıştır (Engin ve Güçlü 2018).

$i(j, k)$ işlem süresini, $k=(1, 2, \dots, m)$ makineleri, $j=(1, 2, \dots, n)$ işleri ve $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ iş dizilimini temsil etmek üzere, minimum gecikme denklem (1) ile bulunur;

$$mg(\pi_{j-1}, \pi_j) = i(\pi_{j-1}, 1) + \max \left[0, \max_{2 \leq k \leq m} \left\{ \sum_{k=2}^m i(\pi_{j-1}, k) - \sum_{k=1}^{m-1} i(\pi_j, k) \right\} \right] \quad (1)$$

MT , maksimum tamamlanma zamanı ve M , tamamlanma zamanı olmak üzere π dizilim kümesindeki n iş ve m makine için C_{max} , denklem (2) ve (3)'de aşağıdaki gibi tanımlanmıştır.

$$C_{max}(\pi) = MT(\pi) = \sum_{j=2}^n mg(\pi_{j-1}\pi_j) + \sum_{k=1}^m i(\pi_n, k) \quad (2)$$

$$MT(\pi^*) \leq M(\pi_n, m) \quad \forall \pi \in \pi \quad (3)$$

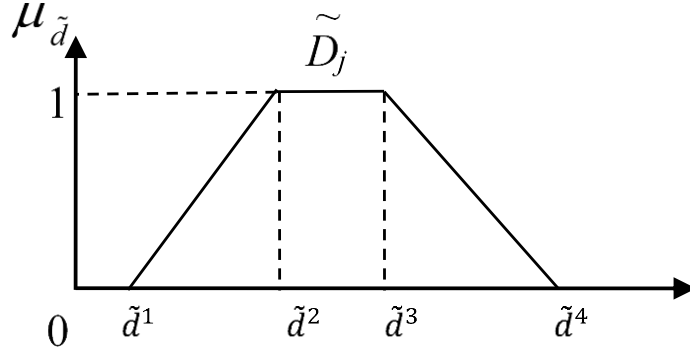
Maksimum tamamlanma zamanı amaç fonksiyonu ile BATÇ problemi, π dizilim kümesinin tamamında en uygun π^* diziliminin bulunmasıdır.

3.1.4. BATÇ Problemlerinde Teslim Tarihlerinin Bulanıklaştırılması

Çizelgeleme problemleri üzerine yapılan çalışmaların çoğunluğunda, problemin deterministik olduğu varsayılmaktadır. Yani belirsizlik durumu dikkate alınmaz, kesin bilgilerin elde olduğu varsayılarak çözüme ulaşılr. Ancak gerçek hayatta çizelgeleme problemleri, beklenmeyen ortam koşullarından etkilenir. Bundan dolayı çizelgeleme problemlerinin çözümünde belirsizlik durumları dikkate alınmalıdır. Aksi takdirde elde edilen çözüm, uygulama aşamasında beklenmeyen durumlardan dolayı optimum çözümden sapma gösterecektir. Bunun önüne geçmek için Prof. Lotfi A. Zadeh tarafından 1965 yılında geliştirilmiş olan bulanık küme teorisi kullanılmaya başlanmıştır. Bu çalışmada da bulanık ortam koşulları dikkate alınmaktadır.

Kökçam ve Engin (2010) tarafından yapılan çalışmada, Bulanık küme teorisi şu şekilde tanımlanmaktadır; “Bulanık küme teorisi, tam olarak tanımlanması zor olan sistemlerin modellenmesinde kullanılmaktadır. Dolayısıyla belirsizlikle dolu bir problemin en uygun çözümünü bulmak için belirsizlikle uğraşabilecek çözüm tekniklerine ihtiyaç vardır. Böyle bir problemin çözümünde günümüzde kullanılabilecek en iyi yöntemlerden biri bulanık küme teorisi yöntemidir.”

Bu çalışmada, işlerin teslim tarihleri, yamuk bulanık sayı olarak tanımlanmıştır. Zhou ve Gu (2009) tarafından \tilde{D}_j ile gösterilen bulanık teslim tarihi Şekil 1’de $(\tilde{d}^1, \tilde{d}^2, \tilde{d}^3, \tilde{d}^4)$ olarak gösterilmiştir.



Şekil 1. Teslim tarihlerinin üyelik fonksiyonu

İşler için tatmin indeksi denklem (4)'de tanımlanmıştır. Bu çalışmada her bir iş, müşteri siparişini temsil etmektedir. İş tatmini aynı zamanda müşteri tatmini olarak tanımlanmıştır.

$$\tilde{\mu}_{\tilde{d}_j} = \begin{cases} 0 & C_j \leq \tilde{d}^1 \\ \frac{C_j - \tilde{d}^1}{\tilde{d}^2 - \tilde{d}^1} & \tilde{d}^1 < C_j < \tilde{d}^2 \\ 1 & \tilde{d}^2 \leq C_j \leq \tilde{d}^3 \\ \frac{\tilde{d}^4 - C_j}{\tilde{d}^4 - \tilde{d}^3} & \tilde{d}^3 < C_j < \tilde{d}^4 \\ 0 & \tilde{d}^4 \leq C_j \end{cases} \quad (4)$$

Denklem (4)'de görüldüğü üzere, eğer j işinin teslim tarihi \tilde{d}^1 'den küçük ise alacağı tatmin indeksi 0'dır. j işinin teslim tarihi \tilde{d}^1 ile \tilde{d}^2 arasında ise tatmin indeksini bulmak için j işinin tamamlanma zamanından \tilde{d}^1 değeri çıkarılarak, $\tilde{d}^2 - \tilde{d}^1$ 'e bölünür. Eğer j işinin teslim tarihi \tilde{d}^2 ile \tilde{d}^3 arasında ise tatmin indeksi 1 değerini alır. Eğer j işinin teslim tarihi \tilde{d}^3 ile \tilde{d}^4 arasında ise tatmin indeksini bulmak için \tilde{d}^4 'den j işinin tamamlanma zamanı çıkarılarak, $\tilde{d}^4 - \tilde{d}^3$ 'e bölünür. Son olarak, eğer j işinin teslim tarihi \tilde{d}^4 değerinden büyük ise alacağı tatmin indeks değeri 0 olur.

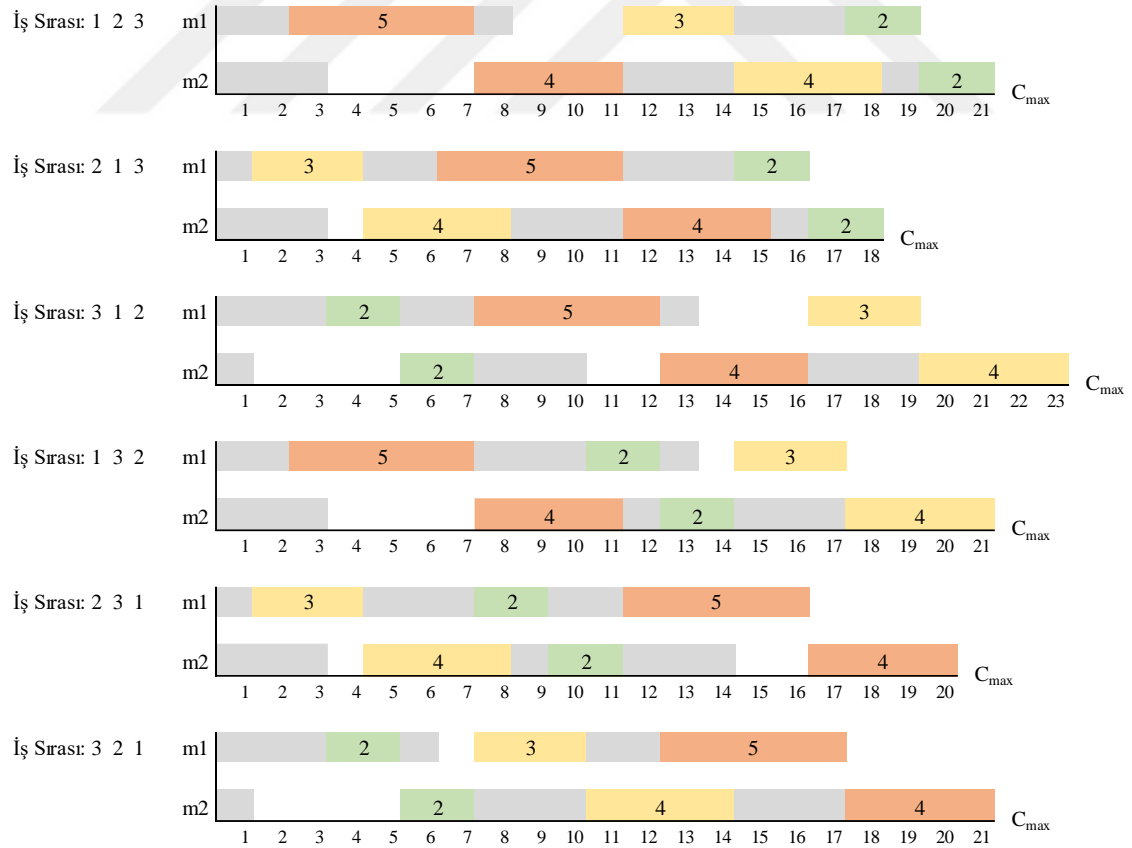
3.1.5. Örnek BATÇ Problemi

Bu bölümde, BATÇ problemlerinin daha iyi anlaşılması için Shyu ve ark. (2004) tarafından oluşturulmuş olan 3-iş x 2-makine problemi ele alınmıştır. Probleme ait işlem ve hazırlık süreleri Çizelge 3'te verilmiştir.

Çizelge 3. 3-iş x 2-makine probleminin hazırlık ve işlem süreleri

İşler	1	2	3
P_{i1}	5	3	2
P_{i2}	4	4	2
S_{i1}	2	1	3
S_{i2}	3	3	1

Çizelge 3'te verilmiş olan 3-iş x 2-makine probleminde işler 3! tane sıralamaya girebilmektedir. Bu durumda bu problem için $3 * 2 * 1 = 6$ farklı çizelgenin oluşması söz konusudur. Tüm sıralamalara ait Gantt gösterimleri Şekil 2'de verilmiştir.



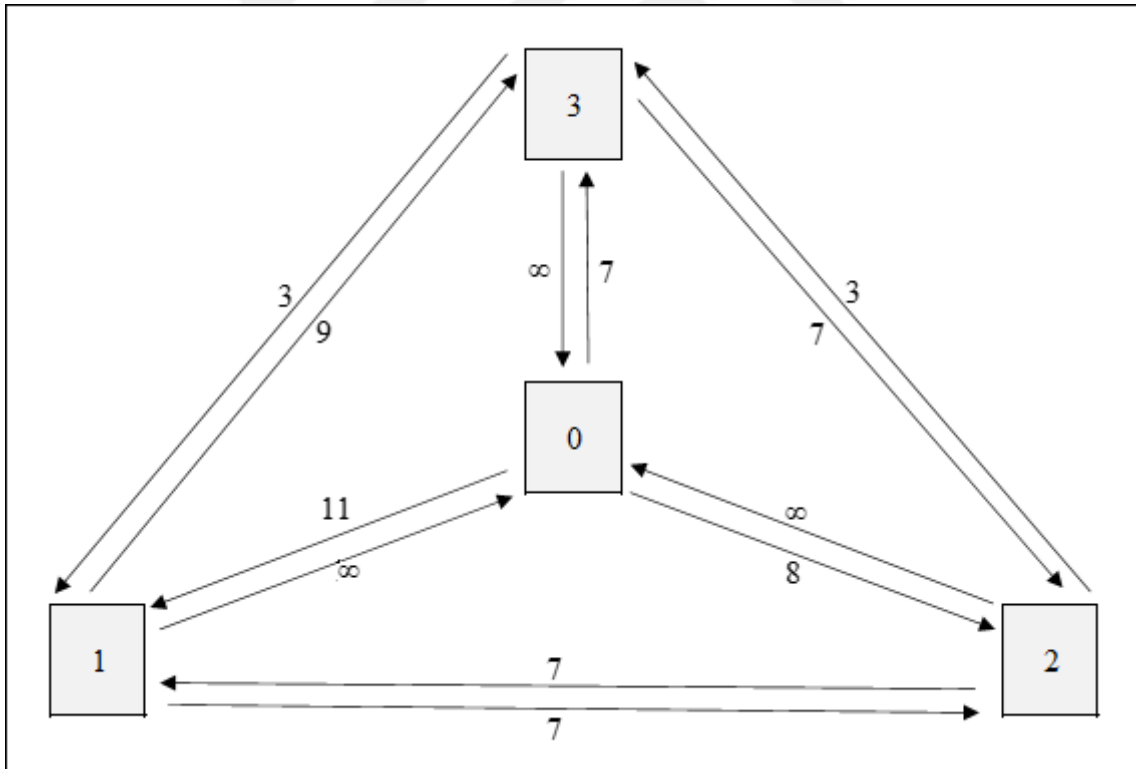
Şekil 2. 3-iş x 2-makine problemi tüm kombinasyonların Gantt çizelgesi

Shyu ve ark. (2004) tarafından oluşturulmuş olan 3-iş x 2-makine BATÇ problemine ait tüm çözümler Çizelge 4’te verilmiştir.

Çizelge 4. 3-iş x 2-makine BATÇ probleminin çözümleri

İş sırası	C_{max}
1-2-3	21
2-1-3	18
3-1-2	23
1-3-2	21
2-3-1	20
3-2-1	21

Maksimum tamamlanma zamanının en küçük olmasını sağlayan iş sırası 2-1-3 şeklinde olmuş ve C_{max} değeri 18 olarak bulunmuştur. Probleme ait grafik gösterimi ise Şekil 3’te verilmiştir.



Şekil 3. 3-iş x 2-makine BATÇ probleminin grafik gösterimi

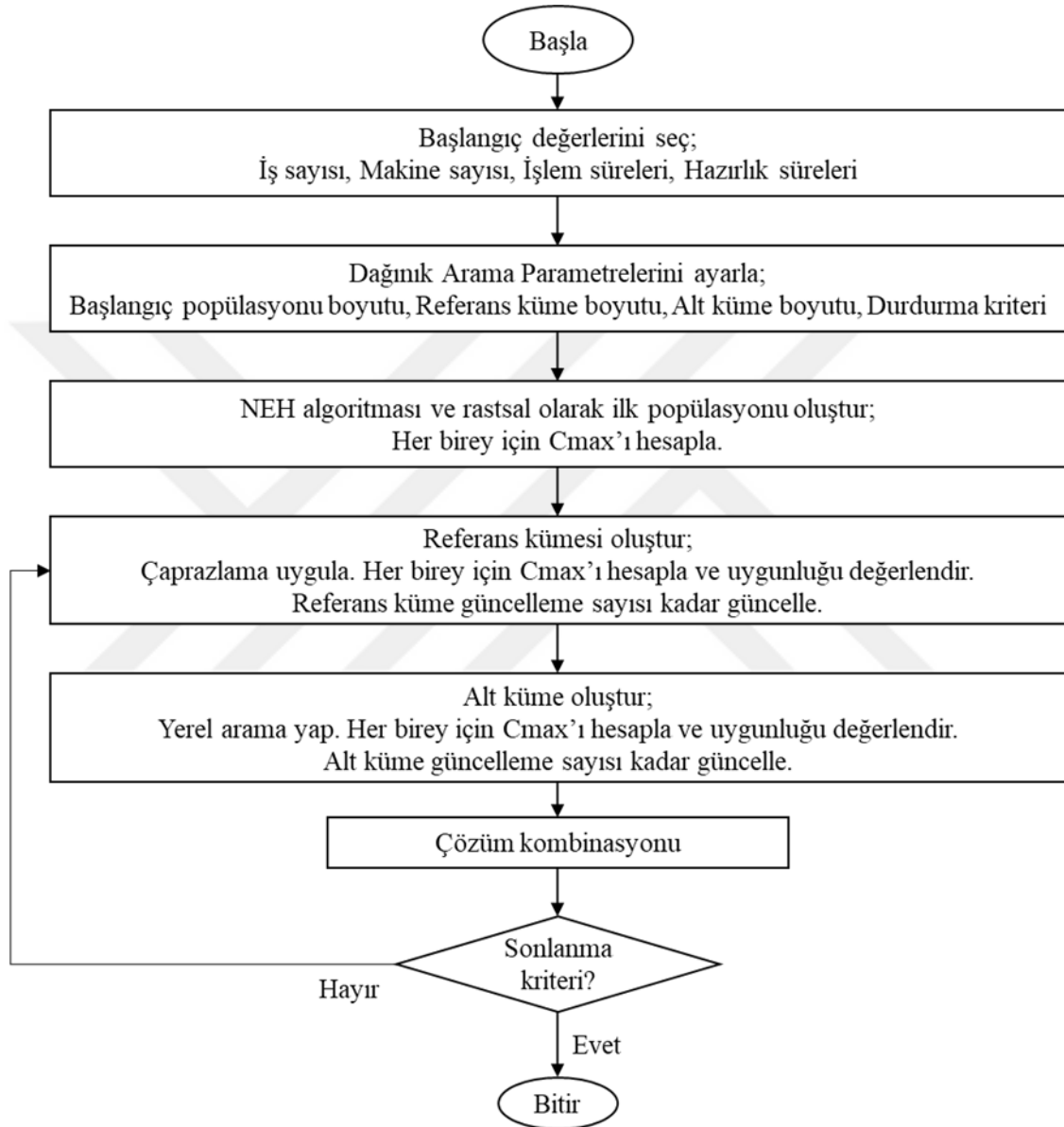
3.2. Yöntem

Bu çalışmada, BATÇ problemlerinin çözümü için DAY kullanılmıştır. DAY, 1970'lerde Fred Glover tarafından ortaya atılmıştır. Fakat 1990'dan sonra ilgi görmeye başlamıştır. DAY, çözüm kalitesi açısından iyi bireylerin birleştirilmesi ve oluşan yeni nesiller içerisinde en iyilerinin seçilmesine dayanan evrimsel bir meta-sezgisel algoritmadır. DAY, günümüze kadar polinomial olarak çözülemeyen birçok zor probleme uyarlanmış oldukça etkili bir optimizasyon tekniğidir. DAY'nin çeşitli iyileştirme teknikleri ile geliştirilmiş ve farklı sezgisel yöntemler ile birleştirilerek hibrit bir yaklaşım olarak kullanılmış versiyonları literatürde bulunmaktadır.

DAY, 6 işlem ve 3 sonlanma kriteri olmak üzere 9 adımdan oluşur (Oktay ve Engin 2006). Bu adımlar aşağıda özetlenmiştir;

1. *Başlangıç popülasyonu oluşturma*; rastsal ve çeşitli teknikler kullanılarak ilk popülasyon oluşturulur.
2. *Referans kümesi oluşturma*; ilk popülasyondan amaç fonksiyonu açısından en iyi bireylerin seçilmesi aşamasıdır.
3. *Alt küme oluşturma*; referans kümesine seçilmiş olan kaliteli bireylerin alt kümelerinin oluşturulduğu aşamadır.
4. *Kombinasyon uygulama*; alt kümede yer alan bireyler çeşitli teknikler kullanılarak birleştirilir ve yeni nesiller ortaya çıkarılmış olur.
5. *İyileştirme*; kombinasyon uygulamasından sonra ortaya çıkan yeni nesillerin hem uygun bir çözüm hem de kaliteli bir çözüm olması için iyileştirildiği aşamadır.
6. *Referans küme güncelleme*; iyileştirilmiş olan yeni nesillerden en iyi çözüm verenler referans kümesindeki bireylerden daha kaliteli ise referans kümesindeki bireylerin yerini yüksek kaliteli yeni nesiller alır.
7. *Yeni referans kümesi oluşturma kriteri*; altıncı aşama ile yedinci aşama tek bir adım olarak düşünülebilir. Çünkü bu aşamada yeni referans kümesi seçilerek eski referans kümesi güncellenmiş olur. Üçüncü aşamadan altıncı aşamaya kadar süreç uygulanmaya devam edilir.
8. *Yeni popülasyon oluşturma kriteri*; başlangıç popülasyonunun ne zaman güncellenmesi gerektiğini belirten kriterdir. İkinci aşamadan yedinci aşamaya kadar süreç uygulanmaya devam edilir.
9. *Sonlanma kriteri*; sürecin ne zaman bitirilmesi gerektiğini ifade eder.

Lourenço ve ark. (2000) tarafından, DAY'nin referans kümesine ait parametre özellikleri, r_1 : en iyi çözüm sayısı ve r_2 : farklı çözüm sayısı olmak üzere, $R = r_1 + r_2$ şeklinde verilmiştir. Şekil 4'te, BATÇ problemleri için bu çalışmada uyarlanmış olan, DAY'ne ait çözüm adımları, verilmiştir.



Şekil 4. Dağınk Arama Yöntemi çözüm adımları

DAY genellikle Tabu Aramanın uyarlanabilir bellek fikrini evrimsel arama kavramına yerleştirir. DAY bir dizi çözüm üzerinde çalışan, evrimsel bir algoritmadır. DAY, bireyleri stratejik olarak birleştirerek ve yüksek kaliteli nesiller elde etmek için nesillerin yerel komşularını arayarak yeni çözümler oluşturur. Birleştirilecek olan çözümler, problemin özelliklerine göre önceden belirlenmiş bir kuralla seçilir. Bu adım,

diğer sezgisellerde genellikle rastsal olarak gerçekleşir. Ayrıca oluşturulan çözümler, sürecin geri kalanına katılmaya uygun olmasa bile tüm olası çözümlere yerel arama uygulanır. Bu fikrin sonucu olarak sadece elde edilen çözümlerin değil aynı zamanda çözümlerin yakın komşularının da arama süreci için uygun olduğu ortaya çıkmaktadır. Bu durumda, daha geniş bir çözüm uzayı içerisinde arama yapabiliyor ve yoğunluğu daha fazla kontrol edebiliyoruz. Çözüm kombinasyonu metodolojisi, elde edilen yeni nesiller içerisinde, ayrı olarak bulunan yüksek kaliteli çözümleri yakalamaya çalışır. Tüm bunlara ek olarak, DAY çeşitlendirmeyi diğer sezgisel yöntemlerden farklı bir şekilde ele alır. Tüm arama süreci için kabul edilen bir çeşitlendirme ölçüsünden yararlanır ve her bir çözümün, çeşitlilik değerlerini aynı anda mevcut diğer çözümlere göre hesaplar. Böylece sürecin geri kalanı için kullanılacak en çeşitli çözüm kümesini seçmek için arama sürecini mümkün kılar. DAY'nin proje çizelgeleme, kaynak kullanılabilirlik maliyetleri ile çizelgeleme, araç rotalama problemi, yazılım testi, depo yerleşim problemi, yapay sinir ağları, kümeleme problemi, p-medyan problem ve alt küme seçim problemi gibi alanlarda uygulamaları bulunmaktadır. Dağınık arama sürecine ait işlem adımları aşağıda verilmiştir.

Adım 0. Girdiler:

- İş sayısı;*
- Makine sayısı;*
- İşlem süreleri;*
- Hazırlık süreleri;*
- Başlangıç popülasyon büyüklüğü;*
- Referans kümesi büyüklüğü;*
- Alt küme büyüklüğü;*
- Sonlanma kriteri;*

Adım 1. NEH algoritması ve rastsal olarak başlangıç popülasyonunu oluştur.

Adım 2. Başlangıç popülasyonunu yerel arama ile iyileştir.

Adım 3. Referans kümesini seç.

Adım 4. Alt küme oluştur.

Adım 5. Çözüm kombinasyonu uygula.

Adım 5.1. Çözümleri değerlendir.

Adım 5.2. Popülasyonu güncelle.

Adım 6. Referans kümesi güncelleme sayısı kadar 3'den 5.2'ye kadar tekrar et.

Adım 7. Popülasyon kümesi güncelleme sayısı kadar 2'den 6'ya kadar tekrar et.

Adım 8. Sonlanma kriteri sağlanıncaya kadar 1'den 7'ye kadar tekrar et.

Adım 9. Çıktılar:

Elde edilen minimum C_{max} değeri.

Elde edilen minimum C_{max} değerine ait iş sırası.

DAY'nin temel olarak beş ana bileşeni bulunmaktadır ve bunlar aşağıdaki şekildedir: Çeşitlendirme yöntemi, İyileştirme yöntemi, Referans kümesi güncelleme, Alt küme oluşturma ve Çözüm birleştirme yöntemidir.

Çeşitlendirme yöntemi: Arama sürecini başlatmak için kullanılan bir yöntemdir. DAY'nin ana fikri, eski nesillerin en uygun özelliklerinin, yeni nesillere aktarılması için çeşitli çözümlerin birleştirilmesi olduğundan, ilk popülasyon kümesi çeşitli çözümlerden oluşmalıdır. Ayrıca çözümler, mümkün olduğunca kaliteli olmalıdır. Arama süreci problemin özelliklerine bağlı olarak değişebilecek olsa da ya rastsal bir seçim ile ya da sezgisel bir başlatma yöntemi ile oluşturulmuş tek bir çözüm ile başlar. Daha sonra, çeşitlendirme operatörleri kullanılarak, yeni ve farklı çözümler üretilir. Öte yandan çeşitlendirme operatörü kullanmak zorunlu değildir. Problemin yapısına bağlı olarak, tüm başlangıç çözümleri birbirinden bağımsız olarak üretilebilir. Bu durumda çözümlerin, çeşitlilik şartını sağladıklarından emin olunmalıdır.

İyileştirme yöntemi: Yerel aramanın tanımlandığı süreçtir. İyileştirme yönteminin kullanılması zorunlu olmasa da çoğu problem için uygulanması şiddetle tavsiye edilir. Bu yöntem, işlem sırasında yoğunlaştırmayı yönetir ve yeni çözümlerin komşularının aranması içinde kullanılır. Çeşitlendirme, başlangıç popülasyonunun çözüm kalitesini artırmak için de kullanılabilir veya çözüm birleştirme ile oluşturulan çözümlerin komşularının da araştırılması için kullanılabilir.

Referans kümesi güncelleme yöntemi: Referans kümesindeki elemanların seçildiği süreçtir. Referans kümesi elemanlarının seçimi genellikle basittir; popülasyondan en yüksek kaliteli ve çeşitlilik içeren çözümlerin seçilmesi ile elde edilir. Popülasyon, algoritmanın bir güncelleme aşamasında ele alınan tüm çözümlerin kümesidir. Başlangıçta popülasyonu, çeşitlendirme ile oluşturulan ilk çözümler kümesidir ve daha sonra iyileştirme yöntemi ile geliştirilebilir. Arama sürecinin sonraki adımlarında, popülasyon, ana çözümlerin birleştirilmesi ile oluşturulan tüm yeni nesillerin kümesidir. Dağılık arama süreci, referans kümesi üzerinde çalışır ve referans kümesi, R büyüklüğünde bir dizi yüksek kaliteli çözümden oluşur. Referans kümesi güncelleme yöntemi, alt küme güncelleme yöntemi ve çözüm birleştirme yöntemi referans kümesi üzerinde çalışır. Bu nedenle referans kümesi, DAY'nin özü niteliğindedir. Referans kümesi, amaç fonksiyon değeri açısından yüksek kaliteli çözümler içeren r_1 büyüklüğünde bir kümeden ve çeşitlendirme ölçüsü açısından farklı çözümler içeren r_2 büyüklüğünde iki ana kümenin birleşiminden oluşur. Referans kümesinde r_2 büyüklüğündeki küme elemanlarının seçim aşamasındaki temel fikir, r_1

büyükliğindeki referans kümesi için önceden seçilmiş çözümlere göre çözümlerin çeşitliliğinin hesaplanmasıdır. En yüksek çeşitlilik değerine sahip olan çözümler r_2 büyüklüğündeki kümeye seçilir.

Alt küme oluşturma yöntemi: Bu yöntem, ana çözümlerin birleştirilerek yeni çözümlerin oluşturulduğu ve oluşturulan çözümlerin bir küme içerisinde tutularak, amaç fonksiyon değeri açısından karşılaştırılmasının yapıldığı bir aşamadan oluşmaktadır.

Çözüm birleştirme yöntemi: Mevcut çözümler kullanılarak yeni çözümler üretmek için mevcut çözümlerin çeşitlilik içeren ve amaç fonksiyon değeri açısından yüksek kaliteli parçalarının birleştirildiği aşamadır. Çözüm birleştirme süreci, alt küme oluşturma yöntemi ile eş zamanlı olarak hareket etmelidir. Ayrıca çözüm birleştirme yöntemi, çözümlerin iyi özelliklerini geliştirecek ve bu özelliklerin yeni nesillere etkili bir şekilde aktarabilecek şekilde kurgulanmalıdır.

DAY'nde mümkün olan en az hesaplama çabası ile global optimuma ulaşma şansını artırmak için algoritmanın, amaç fonksiyon değeri açısından yüksek kaliteli çözümler ile başlatılması çok önemlidir. Referans kümesinde ise popülasyondan hem yüksek kaliteli çözümlerin hem de çeşitlilik sağlayacak olan çözümlerin seçilmesi, alt kümede çeşitli ve yüksek kaliteli yeni nesillerin seçilmesine olanak sağlayacaktır. Yapılan bu çalışmada, popülasyondan en yüksek kalitede iki çözümün ve çeşitlilik açısından en farklı bir çözümün seçilimi ile birleştirilecek olan çözüm grupları oluşturulmuştur. Son aşama olan çözüm birleştirme yönteminde ise, ortaya çıkan yeni çözümler, yeni bir popülasyon olarak tutulur ve iyileştirilmiş popülasyondaki çözümler ile karşılaştırması yapılarak en iyi çözümler, en kötü çözümlerin yerini alırlar. Durdurma koşulu sağlanana kadar, arama süreci ve iyileştirmeler sonucu yapılan güncellemeler devam eder. Durdurma koşulu, belirlenen bir yinleme sayısı veya yüksek kaliteli çözümlerin gelişmesinde yeni bir değişikliğin gözlenmemesini sağlayan bir şartın olması ile belirlenir.

4. BATÇ PROBLEMLERİNİN ÖNERİLEN DAY İLE ÇÖZÜMÜ

Bu bölümde, BATÇ problemine uyarlanmış olan DAY'nin etkinliğini ispatlamak için Engin ve Günaydın (2011) tarafından geliştirilen kıyaslama problemleri kullanılmıştır. Çözülen problem setindeki iş sayıları $n=8, 10, 12, 50, 100, 150, 200, 250$ olmak üzere 8 farklı türdedir. Makine sayıları ise $m=2, 3, 5, 8, 10, 15, 20, 25$ olmak üzere 8 farklı makine sayısından oluşmaktadır. Problemlerin işlem süreleri ise a, b ve c grubu olmak üzere 3 farklı işlem süresi aralığına sahiptir. a grubu problemlerin işlem süreleri 1 ile 10 arasında rastsal değerler almakta ve üniform dağılıma sahip bir yapıdadır. b grubu problemlerin işlem süreleri 1 ile 50 arasında rastsal değerler almakta ve üniform dağılıma sahip bir yapıdadır. c grubu problemlerin işlem süreleri ise 1 ile 100 arasında rastsal değerler almakta ve üniform dağılıma sahip bir yapıdadır. Problemlerin hazırlık süreleri ise 1 ile 10 arasında, üniform dağılıma uygun bir yapıda, rastsal üretilmiş verilerden oluşmaktadır. Toplamda 8 farklı iş sayısı, 8 farklı makine sayısı ve 3 farklı işlem süresinden oluşan $8 \times 8 \times 3 = 192$ farklı problem seti çözülmüştür.

DAY ile elde edilmiş olan maksimum tamamlanma zamanı amaç fonksiyonunun çözüm sonuçları, Genetik Algoritma (Gen-2), Uyarlanabilir Öğrenme Yaklaşım (UÖY) algoritması ve Hibrit Karınca Koloni Optimizasyon (HKKO) algoritması ile kıyaslanmıştır. Gen-2 algoritması Aldowaisan ve Allahverdi (2003) tarafından, UÖY algoritması Engin ve Günaydın (2011) tarafından ve HKKO algoritması ise Engin ve Güçlü (2018) tarafından önerilmiştir.

DAY algoritması, Microsoft Office Professional Plus 2019 paketi içerisinde ki Excel ortamında Visual Basic dilinde yazılmış ve Intel Core i5 64 bit 2.5 GHz 4 GB RAM'e sahip bir bilgisayarda çözülmüştür. DAY'nin parametreleri belirlenirken, her makine ve iş grubundan seçilen problemler üzerinde tam faktöriyel olarak parametre tasarımı yapılmıştır. Parametre optimizasyonunda kullanılan parametre değerleri aşağıda verilmiştir;

Başlangıç popülasyonu büyüklüğü	: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Alt küme büyüklüğü	: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Referans kümesi büyüklüğü	: 3, 4, 5
Popülasyon güncelleme sayısı	: 50, 60, 70, 80, 90
Referans kümesi güncelleme sayısı	: 50, 100, 150, 200, 250

Tam faktöriyel olarak yapılan parametre optimizasyonu sonucunda, problemlerin çözümü için en iyi sonuç veren DAY'ne ait parametre değerleri aşağıda verilmiştir;

Başlangıç popülasyonu büyüklüğü	: 10
Alt küme büyüklüğü	: 30
Referans kümesi büyüklüğü	: 3
Popülasyon güncelleme sayısı	: 80
Referans kümesi güncelleme sayısı	: 200

8 ve 10 iş grubuna ait toplam 48 adet problem, Engin ve Güçlü (2018) tarafından CPLEX ile matematiksel model olarak çözülmüştür. Gen-2, UÖY, HKKO ve önerilen DAY, CPLEX ile elde edilmiş olan optimum sonuçlara göre kıyaslanmıştır. Optimum sonuca göre sapmalar aşağıda verilen denklem (5) ile hesaplanmıştır. Sonuçlar, Çizelge 5 ve Çizelge 6'da verilmiştir. Çizelge 7'de ise 8 iş ve 10 işe göre ortalama sapmalar ve optimum sonuca ulaşma sayıları verilerek algoritmaların karşılaştırılması yapılmıştır.

$$\text{Optimum Sonuçtan \%Sapma} = \frac{\text{Karşılaştırılan } C_{max} - \text{Optimum } C_{max}}{\text{Optimum } C_{max}} \times 100 \quad (5)$$

Önerilen DAY'nin etkinliğini ispatlamak için 192 test problemi üzerinde Gen-2, UÖY ve HKKO algoritmaları ile kıyaslama yapılmıştır. Kıyaslama yapılırken dört sezgisel algoritma içerisinde en iyi C_{max} değerini bulan algoritmaya göre %Sapma değeri denklem (6)'ya göre hesaplanmıştır. Sonuçlar makine sayılarına göre gruplanmış biçimde Çizelge 8-15'te verilmiştir. Çizelge 16'da 192 problem için en iyi C_{max} değerini bulma sayısına göre dört algoritmanın karşılaştırılması yapılmıştır. Çizelge 17'de ise makine sayılarına göre ortalama sapmalar hesaplanarak dört algoritmanın karşılaştırılması yapılmıştır.

$$\text{En İyi Sonuçtan \%Sapma} = \frac{\text{Karşılaştırılan } C_{max} - \text{En İyi } C_{max}}{\text{En İyi } C_{max}} \times 100 \quad (6)$$

Çizelge 5. n=8 için CPLEX sonucuna göre sapma miktarları

Problem	C_{max}					%Sapma			
	Gen-2	UÖY	HKKO	DAY	CPLEX	Gen-2	UÖY	HKKO	DAY
8x2-a	93	93	93	93	93	0,0000	0,0000	0,0000	0,0000
8x2-b	255	255	255	255	255	0,0000	0,0000	0,0000	0,0000
8x2-c	465	468	468	465	465	0,0000	0,6452	0,6452	0,0000
8x3-a	92	92	92	92	92	0,0000	0,0000	0,0000	0,0000
8x3-b	290	304	302	290	290	0,0000	4,8276	4,1379	0,0000
8x3-c	565	562	560	558	558	1,2545	0,7168	0,3584	0,0000
8x5-a	127	126	126	126	126	0,7937	0,0000	0,0000	0,0000
8x5-b	396	398	398	396	396	0,0000	0,5051	0,5051	0,0000
8x5-c	752	744	744	744	744	1,0753	0,0000	0,0000	0,0000
8x8-a	132	132	132	132	132	0,0000	0,0000	0,0000	0,0000
8x8-b	518	518	518	518	518	0,0000	0,0000	0,0000	0,0000
8x8-c	990	990	990	990	990	0,0000	0,0000	0,0000	0,0000
8x10-a	153	152	152	151	151	1,3245	0,6623	0,6623	0,0000
8x10-b	602	602	602	602	602	0,0000	0,0000	0,0000	0,0000
8x10-c	1162	1169	1162	1162	1162	0,0000	0,6024	0,0000	0,0000
8x15-a	193	193	193	193	193	0,0000	0,0000	0,0000	0,0000
8x15-b	777	777	775	773	773	0,5175	0,5175	0,2587	0,0000
8x15-c	1487	1487	1487	1487	1487	0,0000	0,0000	0,0000	0,0000
8x20-a	217	216	216	216	216	0,4630	0,0000	0,0000	0,0000
8x20-b	892	892	892	887	887	0,5637	0,5637	0,5637	0,0000
8x20-c	1742	1742	1742	1742	1742	0,0000	0,0000	0,0000	0,0000
8x25-a	249	249	249	249	249	0,0000	0,0000	0,0000	0,0000
8x25-b	1078	1069	1062	1050	1050	2,6667	1,8095	1,1429	0,0000
8x25-c	2102	2051	2051	2051	2051	2,4866	0,0000	0,0000	0,0000

Çizelge 5'te 8 iş problemleri için Gen-2, UÖY, HKKO ve DAY algoritmalarının elde ettiği C_{max} değerleri ile birlikte CPLEX ile elde edilmiş optimum C_{max} değerleri verilmiştir. CPLEX ile elde edilen C_{max} değerine göre algoritmaların %Sapmaları hesaplanmış ve Çizelge 5'te verilmiştir. Çizelge 5'te görüldüğü üzere, DAY problemlerin tamamında optimum sonucu bularak, diğer algoritmalarından daha iyi sonuç vermiştir. Çizelge 7'de ise algoritmaların ortalama %Sapmaları ile birlikte optimum sonuca ulaşma sayıları verilmiştir.

Çizelge 6. n=10 iş için CPLEX sonucuna göre sapma miktarları

Problem	C_{max}					%Sapma			
	Gen-2	UÖY	HKKO	DAY	CPLEX	GEN-2	UÖY	HKKO	DAY
10x2-a	110	110	110	110	110	0,0000	0,0000	0,0000	0,0000
10x2-b	338	344	342	337	337	0,2967	2,0772	1,4837	0,0000
10x2-c	593	584	583	581	581	2,0654	0,5164	0,3442	0,0000
10x3-a	119	120	118	117	117	1,7094	2,5641	0,8547	0,0000
10x3-b	379	367	362	362	362	4,6961	1,3812	0,0000	0,0000
10x3-c	693	688	672	667	667	3,8981	3,1484	0,7496	0,0000
10x5-a	142	140	138	138	138	2,8986	1,4493	0,0000	0,0000
10x5-b	494	513	501	462	462	6,9264	11,0390	8,4416	0,0000
10x5-c	912	904	894	873	873	4,4674	3,5510	2,4055	0,0000
10x8-a	171	171	170	169	169	1,1834	1,1834	0,5917	0,0000
10x8-b	618	623	614	571	571	8,2312	9,1068	7,5306	0,0000
10x8-c	1180	1180	1166	1076	1076	9,6654	9,6654	8,3643	0,0000
10x10-a	182	184	178	176	176	3,4091	4,5455	1,1364	0,0000
10x10-b	654	679	656	612	612	6,8627	10,9477	7,1895	0,0000
10x10-c	1290	1228	1192	1159	1159	11,3028	5,9534	2,8473	0,0000
10x15-a	223	221	215	216	215	3,7209	2,7907	0,0000	0,4651
10x15-b	887	879	856	849	849	4,4759	3,5336	0,8245	0,0000
10x15-c	1683	1707	1658	1632	1632	3,1250	4,5956	1,5931	0,0000
10x20-a	264	262	260	257	257	2,7237	1,9455	1,1673	0,0000
10x20-b	1090	1081	1062	1043	1043	4,5062	3,6433	1,8217	0,0000
10x20-c	2115	2097	2062	2025	2025	4,4444	3,5556	1,8272	0,0000
10x25-a	285	284	278	275	275	3,6364	3,2727	1,0909	0,0000
10x25-b	1250	1248	1215	1195	1195	4,6025	4,4351	1,6736	0,0000
10x25-c	2428	2389	2342	2346	2342	3,6721	2,0068	0,0000	0,1708

Çizelge 6'da, 10 iş problemleri için Gen-2, UÖY, HKKO ve DAY algoritmalarının elde ettiği C_{max} değerleri ile birlikte CPLEX ile elde edilmiş optimum C_{max} değerleri verilmiştir. CPLEX ile elde edilen C_{max} değerine göre algoritmaların %Sapmaları hesaplanmış ve Çizelge 6'da verilmiştir. Çizelge 6'da görüldüğü üzere, DAY 0,0265 ortalama %Sapma ile diğer algoritmalarından daha iyi sonuç vermiştir. Çizelge 7'de ise algoritmaların ortalama %Sapmaları ile birlikte optimum sonuca ulaşma sayıları verilmiştir.

Çizelge 7. CPLEX sonucuna göre algoritmaların kıyası

Algoritmalar	Ortalama %Sapma		Optimum Sonuç Sayısı
	8 iş	10 iş	
Gen-2	0,4644	4,2717	16
UÖY	0,4521	4,0378	16
HKKO	0,3448	2,1641	21
DAY	0,0000	0,0265	46

Çizelge 7’de görüldüğü üzere 8 iş ve 10 iş grubu toplam 48 problem için optimum sonuçtan ortalama %Sapma değeri en düşük çıkan algoritma önerilen DAY’dır. 8 iş grubu problemlerde, Gen-2 algoritması %0,4644, UÖY algoritması %0,4521, HKKO algoritması %0,3448 ve DAY algoritması %0,0000 optimum sonuçtan sapma göstermiştir. DAY algoritması 24 adet 8 iş probleminin tamamında optimum sonuca ulaşmayı başarmıştır. 10 iş grubu problemlerde ise Gen-2 algoritması %4,2717, UÖY algoritması %4,0378, HKKO algoritması %2,1641 ve DAY algoritması %0,0265 optimum sonuçtan sapma göstermiştir. DAY algoritması 24 adet 10 iş grubu problemde 22 tanesi için optimum sonuca ulaşmayı başarmıştır. Gen-2, UÖY, HKKO ve DAY algoritmaları toplam 48 adet problemde sırasıyla 16, 16, 21 ve 46 tane optimum sonuca ulaşmıştır. Sonuçlardan görüleceği üzere DAY diğer yöntemlerden oldukça verimli sonuçlar vermiştir.

Çizelge 8’den Çizelge 15’e kadar 192 adet BATÇ test probleminin Gen-2, UÖY, HKKO ve DAY algoritmaları ile elde edilen C_{max} değerlerinin bilinen en iyi C_{max} değerinden %Sapma miktarları makine sayılarına göre gruplandırılmış biçimde verilmiştir. %Sapma hesaplamaları 7, 8, 9 ve 10. denklemler ile yapılmıştır. Çizelgelerde DAY’nin işlem süreleri ise saniye cinsinden verilmiştir.

$$Gen - 2 \%Sapma = \frac{Gen - 2 C_{max} - En\ iyi\ C_{max}}{En\ iyi\ C_{max}} \times 100 \quad (7)$$

$$UÖY \%Sapma = \frac{UÖY\ C_{max} - En\ iyi\ C_{max}}{En\ iyi\ C_{max}} \times 100 \quad (8)$$

$$HKKO \%Sapma = \frac{HKKO\ C_{max} - En\ iyi\ C_{max}}{En\ iyi\ C_{max}} \times 100 \quad (9)$$

$$DAY \%Sapma = \frac{DAY\ C_{max} - En\ iyi\ C_{max}}{En\ iyi\ C_{max}} \times 100 \quad (10)$$

Çizelge 8. 2 makineli problemler için sapma miktarları

Problem	%Sapma				DAY İşlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x2-a	0,0000	0,0000	0,0000	0,0000	1
8x2-b	0,0000	0,0000	0,0000	0,0000	1
8x2-c	0,0000	0,6452	0,6452	0,0000	1
10x2-a	0,0000	0,0000	0,0000	0,0000	1
10x2-b	0,2967	2,0772	1,4837	0,0000	1
10x2-c	2,0654	0,5164	0,3442	0,0000	1
12x2-a	0,0000	0,0000	0,0000	0,0000	1
12x2-b	0,0000	0,0000	0,0000	0,0000	1
12x2-c	2,5245	0,0000	0,0000	0,0000	1
50x2-a	6,2847	5,7301	5,7301	0,0000	2
50x2-b	11,0622	13,0735	11,0622	0,0000	2
50x2-c	12,5848	11,9064	11,9064	0,0000	2
100x2-a	9,5797	8,5044	8,5044	0,0000	5
100x2-b	18,5382	18,9701	18,5382	0,0000	5
100x2-c	20,7609	21,7572	21,2319	0,0000	5
150x2-a	9,5146	9,1909	9,1909	0,0000	8
150x2-b	20,0641	20,4764	20,1328	0,0000	8
150x2-c	21,7294	21,5549	21,5549	0,0000	9
200x2-a	10,9981	10,4651	10,4651	0,0000	12
200x2-b	20,2845	20,8703	20,4519	0,0000	13
200x2-c	22,5688	21,7867	21,7867	0,0000	12
250x2-a	9,6738	10,2737	10,2737	0,0000	18
250x2-b	21,6850	21,1626	21,1626	0,0000	18
250x2-c	23,4762	22,1056	22,1056	0,0000	19

Çizelge 8’de 2 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde 24 problem içerisinde, Gen-2 algoritması 6 tane, UÖY algoritması 6 tane, HKKO algoritması 6 tane ve DAY algoritması 24 tane, en iyi C_{max} değerini bulmuştur. DAY’ne ait en yüksek işlem süresi ise 250x2-c probleminde 19 saniye olmuştur.

Çizelge 9. 3 makineli problemler için sapma miktarları

Problem	%Sapma				DAY İşlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x3-a	0,0000	0,0000	0,0000	0,0000	1
8x3-b	0,0000	4,8276	4,1379	0,0000	1
8x3-c	1,2545	0,7168	0,3584	0,0000	1
10x3-a	1,7094	2,5641	0,8547	0,0000	1
10x3-b	4,6961	1,3812	0,0000	0,0000	1
10x3-c	3,8981	3,1484	0,7496	0,0000	1
12x3-a	0,0000	0,0000	0,0000	0,0000	1
12x3-b	0,0000	0,0000	0,0000	0,0000	1
12x3-c	2,1302	5,0888	2,1302	0,0000	1
50x3-a	9,0580	14,1304	9,0580	0,0000	3
50x3-b	23,3679	16,0464	16,0464	0,1830	3
50x3-c	25,7914	18,6938	18,6938	0,0000	3
100x3-a	11,1517	14,8080	11,1517	0,0000	6
100x3-b	27,8621	22,2011	22,2011	0,0000	7
100x3-c	30,8293	28,6954	28,6954	0,0000	7
150x3-a	14,9853	13,6283	13,6283	0,0000	10
150x3-b	28,1919	27,4300	27,4300	0,0000	10
150x3-c	31,5260	29,9221	29,9221	0,0000	11
200x3-a	16,7659	16,8575	16,8575	0,0000	17
200x3-b	26,1141	27,4221	26,1141	0,0000	17
200x3-c	32,2969	31,2412	32,0642	0,0000	18
250x3-a	16,2250	17,2414	16,2250	0,0000	24
250x3-b	28,9397	28,0320	28,0320	0,0000	25
250x3-c	32,6435	30,2445	30,2445	0,0000	26

Çizelge 9’da, 3 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 4 tane, UÖY algoritması 3 tane, HKKO algoritması 4 tane ve DAY algoritması 23 tane, en iyi C_{max} değerini bulmuştur. DAY’ne ait en yüksek işlem süresi ise 250x3-c probleminde 26 saniye olmuştur.

Çizelge 10. 5 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x5-a	0,7937	0,0000	0,0000	0,0000	1
8x5-b	0,0000	0,5051	0,5051	0,0000	1
8x5-c	1,0753	0,0000	0,0000	0,0000	1
10x5-a	2,8986	1,4493	0,0000	0,0000	1
10x5-b	6,9264	11,0390	8,4416	0,0000	1
10x5-c	4,4674	3,5510	2,4055	0,0000	1
12x5-a	4,2424	6,6667	4,2424	0,0000	2
12x5-b	7,7206	4,4118	4,4118	0,0000	2
12x5-c	5,5394	7,0943	5,7337	0,0000	2
50x5-a	16,3430	14,2395	14,2395	0,0000	4
50x5-b	26,7259	25,1046	25,1046	0,0000	5
50x5-c	22,6039	28,1440	22,6039	0,2216	5
100x5-a	20,9571	16,5017	16,5017	0,0000	9
100x5-b	34,4441	27,4860	27,4860	0,0000	9
100x5-c	37,5235	28,1933	28,1933	0,0000	9
150x5-a	17,2376	20,6630	17,2376	0,0000	15
150x5-b	34,8969	30,0995	30,0995	0,0000	15
150x5-c	36,5602	31,4307	31,4307	0,0000	16
200x5-a	20,3956	14,6271	14,6271	0,0000	24
200x5-b	33,8380	21,7530	21,7530	0,0000	24
200x5-c	26,9606	37,6594	26,9606	0,0000	25
250x5-a	22,3592	15,5615	15,5615	0,0000	36
250x5-b	35,9442	25,3433	25,3433	0,0000	37
250x5-c	39,9977	29,9433	29,9433	0,0000	37

Çizelge 10'da, 5 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 1 tane, UÖY algoritması 2 tane, HKKO algoritması 3 tane ve DAY algoritması 23 tane, en iyi C_{max} değerini bulmuştur. DAY'ne ait en yüksek işlem süresi ise 250x5-b ve 250x5-c probleminde 37 saniye olmuştur.

Çizelge 11. 8 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x8-a	0,0000	0,0000	0,0000	0,0000	2
8x8-b	0,0000	0,0000	0,0000	0,0000	2
8x8-c	0,0000	0,0000	0,0000	0,0000	2
10x8-a	1,1834	1,1834	0,5917	0,0000	2
10x8-b	8,2312	9,1068	7,5306	0,0000	2
10x8-c	9,6654	9,6654	8,3643	0,0000	2
12x8-a	5,7292	2,0833	2,0833	0,0000	2
12x8-b	7,5110	3,3873	3,3873	0,0000	2
12x8-c	10,1866	0,4666	0,4666	0,0778	2
50x8-a	18,4862	10,9170	10,9170	0,0000	6
50x8-b	29,8253	21,0927	21,0927	0,0000	6
50x8-c	32,5389	26,1916	26,1916	0,0000	6
100x8-a	23,3506	14,7517	14,7517	0,0000	13
100x8-b	29,4829	36,2212	29,4829	0,0000	13
100x8-c	37,2046	26,3270	26,3270	0,0000	14
150x8-a	21,9840	13,8584	13,8584	0,0000	24
150x8-b	34,5352	24,8441	24,8441	0,0000	24
150x8-c	27,2469	37,5162	27,2469	0,0000	23
200x8-a	23,7683	17,5630	17,5630	0,0000	37
200x8-b	38,2425	30,7270	30,7270	0,0000	36
200x8-c	41,7336	34,8511	34,8511	0,0000	38
250x8-a	25,8953	21,0897	21,0897	0,0000	55
250x8-b	41,2547	34,5318	34,5318	0,0000	56
250x8-c	43,6246	33,7438	33,7438	0,0000	57

Çizelge 11’de, 8 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 3 tane, UÖY algoritması 3 tane, HKKO algoritması 3 tane ve DAY algoritması 23 tane, en iyi C_{max} değerini bulmuştur. DAY’ne ait en yüksek işlem süresi ise 250x8-c probleminde 57 saniye olmuştur.

Çizelge 12. 10 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x10-a	1,3245	0,6623	0,6623	0,0000	2
8x10-b	0,0000	0,0000	0,0000	0,0000	2
8x10-c	0,0000	0,6024	0,0000	0,0000	2
10x10-a	3,4091	4,5455	1,1364	0,0000	2
10x10-b	6,8627	10,9477	7,1895	0,0000	2
10x10-c	11,3028	5,9534	2,8473	0,0000	2
12x10-a	9,7436	3,5897	3,5897	0,0000	2
12x10-b	4,8257	8,9812	4,8257	0,0000	2
12x10-c	9,5305	3,2936	3,2936	0,0000	2
50x10-a	18,9041	11,3699	11,3699	0,0000	7
50x10-b	30,3325	20,9246	20,9246	0,0000	7
50x10-c	32,5611	25,0107	25,0107	0,0000	7
100x10-a	22,1140	12,5869	12,5869	0,0000	15
100x10-b	34,5292	34,9855	23,6002	0,0000	17
100x10-c	28,9209	38,6940	28,9209	0,0000	17
150x10-a	24,0697	15,6854	15,6854	0,0000	28
150x10-b	37,2691	24,7798	24,7798	0,0000	28
150x10-c	39,6386	28,7880	28,7880	0,0000	29
200x10-a	14,3270	25,6225	14,3270	0,0000	43
200x10-b	40,4764	27,0314	27,0314	0,0000	45
200x10-c	29,0920	43,5402	29,0920	0,0000	45
250x10-a	27,3536	14,4087	14,4087	0,0000	67
250x10-b	44,4269	29,5333	29,5333	0,0000	69
250x10-c	47,9497	32,7810	32,7810	0,0000	70

Çizelge 12’de, 10 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 2 tane, UÖY algoritması 1 tane, HKKO algoritması 2 tane ve DAY algoritması 24 tane, en iyi C_{max} değerini bulmuştur. DAY’ne ait en yüksek işlem süresi ise 250x10-c probleminde 70 saniye olmuştur.

Çizelge 13. 15 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x15-a	0,0000	0,0000	0,0000	0,0000	2
8x15-b	0,5175	0,5175	0,2587	0,0000	2
8x15-c	0,0000	0,0000	0,0000	0,0000	2
10x15-a	3,7209	2,7907	0,0000	0,4651	2
10x15-b	4,4759	3,5336	0,8245	0,0000	2
10x15-c	3,1250	4,5956	1,5931	0,0000	2
12x15-a	9,5041	4,1322	4,1322	0,0000	3
12x15-b	11,8662	6,0410	6,0410	0,0000	3
12x15-c	12,0089	4,9253	4,9253	0,0000	3
50x15-a	10,4265	18,0095	10,4265	0,0000	10
50x15-b	26,6622	19,0110	19,0110	0,0000	11
50x15-c	19,2729	27,8309	19,2729	0,0000	11
100x15-a	24,7007	16,3831	16,3831	0,0000	22
100x15-b	26,5403	37,4954	26,5403	0,0000	23
100x15-c	38,4717	26,8630	26,8630	0,0000	23
150x15-a	26,7067	13,1816	13,1816	0,0000	40
150x15-b	43,3706	23,6270	23,6270	0,0000	41
150x15-c	42,9520	20,7312	20,7312	0,0000	39
200x15-a	28,7125	14,8831	14,8831	0,0000	65
200x15-b	44,8124	31,9939	31,9939	0,0000	64
200x15-c	29,5680	48,6898	29,5680	0,0000	65
250x15-a	31,6308	13,3014	13,3014	0,0000	96
250x15-b	50,0775	25,9067	25,9067	0,0000	97
250x15-c	30,9434	49,4479	30,9434	0,0000	97

Çizelge 13'te, 15 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 2 tane, UÖY algoritması 2 tane, HKKO algoritması 3 tane ve DAY algoritması 23 tane, en iyi C_{max} değerini bulmuştur. DAY'ne ait en yüksek işlem süresi ise 250x15-b ve 250x15-c probleminde 97 saniye olmuştur.

Çizelge 14. 20 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x20-a	0,4630	0,0000	0,0000	0,0000	2
8x20-b	0,5637	0,5637	0,5637	0,0000	2
8x20-c	0,0000	0,0000	0,0000	0,0000	2
10x20-a	2,7237	1,9455	1,1673	0,0000	3
10x20-b	4,5062	3,6433	1,8217	0,0000	3
10x20-c	4,4444	3,5556	1,8272	0,0000	3
12x20-a	1,4388	6,4748	1,4388	0,0000	3
12x20-b	5,4545	5,4545	5,4545	0,0000	3
12x20-c	7,9825	1,4912	1,4912	0,0439	3
50x20-a	18,9250	11,0862	11,0862	0,0000	14
50x20-b	26,2845	17,9685	17,9685	0,0000	14
50x20-c	21,7141	29,9736	21,7141	0,0000	15
100x20-a	24,1983	13,7609	13,7609	0,0000	30
100x20-b	37,8502	22,6059	22,6059	0,0000	31
100x20-c	41,5625	24,0500	24,0500	0,0000	31
150x20-a	27,8244	14,4511	14,4511	0,0000	52
150x20-b	31,0439	44,3840	31,0439	0,0000	52
150x20-c	45,9464	31,1714	31,1714	0,0000	51
200x20-a	17,2181	29,8713	17,2181	0,0000	83
200x20-b	49,7235	28,8034	28,8034	0,0000	83
200x20-c	30,8434	51,2237	30,8434	0,0000	84
250x20-a	33,7397	20,9754	20,9754	0,0000	124
250x20-b	51,3801	31,3017	31,3017	0,0000	125
250x20-c	52,4118	32,6228	32,6228	0,0000	126

Çizelge 14’te, 20 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 1 tane, UÖY algoritması 2 tane, HKKO algoritması 1 tane ve DAY algoritması 23 tane, en iyi C_{max} değerini bulmuştur. DAY’ne ait en yüksek işlem süresi ise 250x20-c probleminde 126 saniye olmuştur.

Çizelge 15. 25 makineli problemler için sapma miktarları

Problem	%Sapma				DAY işlem Süresi (s)
	Gen-2	UÖY	HKKO	DAY	
8x25-a	0,0000	0,0000	0,0000	0,0000	2
8x25-b	2,6667	1,8095	1,1429	0,0000	2
8x25-c	2,4866	0,0000	0,0000	0,0000	3
10x25-a	3,6364	3,2727	1,0909	0,0000	3
10x25-b	4,6025	4,4351	1,6736	0,0000	4
10x25-c	3,6721	2,0068	0,0000	0,1708	4
12x25-a	0,6369	0,0000	0,0000	0,0000	4
12x25-b	0,3817	0,0000	0,0000	1,4504	4
12x25-c	1,5582	2,5711	1,5582	0,0000	5
50x25-a	20,0824	14,7271	14,7271	0,0000	16
50x25-b	25,4061	15,2064	15,2064	0,0000	17
50x25-c	26,9563	13,7427	13,7427	0,0000	17
100x25-a	17,8591	27,8536	17,8591	0,0000	35
100x25-b	41,2235	26,8257	26,8257	0,0000	36
100x25-c	43,2874	25,9833	25,9833	0,0000	39
150x25-a	29,9361	15,3817	15,3817	0,0000	64
150x25-b	47,3286	30,3247	30,3247	0,0000	66
150x25-c	25,9496	49,6645	25,9496	0,0000	66
200x25-a	33,3721	19,8374	19,8374	0,0000	102
200x25-b	50,1044	26,2807	26,2807	0,0000	103
200x25-c	51,9039	27,5451	27,5451	0,0000	103
250x25-a	35,1193	16,5653	16,5653	0,0000	153
250x25-b	36,2135	54,5520	36,2135	0,0000	154
250x25-c	56,1558	30,5122	30,5122	0,0000	155

Çizelge 15'te, 25 makine problemleri için kıyaslaması yapılan algoritmaların en iyi C_{max} değerinden %Sapma miktarları verilmiştir. Çizelge incelendiğinde, 24 problem içerisinde, Gen-2 algoritması 1 tane, UÖY algoritması 4 tane, HKKO algoritması 5 tane ve DAY algoritması 22 tane, en iyi C_{max} değerini bulmuştur. DAY'ne ait en yüksek işlem süresi ise 250x25-c probleminde 155 saniye olmuştur.

Çizelge 16. 192 test probleminde makinelere göre bulunan en iyi C_{max} sayıları

Makine Sayısı	Bulunan en iyi C_{max} sayısı			
	Gen-2	UÖY	HKKO	DAY
2	6	6	6	24
3	4	3	4	23
5	1	2	3	23
8	3	3	3	23
10	2	1	2	24
15	2	2	3	23
20	1	2	2	23
25	1	4	5	22
Toplam	20	23	28	185

Çizelge 16’da Gen-2, UÖY, HKKO ve DAY’nin makine sayılarına göre gruplandırılmış biçimde buldukları toplam en iyi C_{max} sayıları verilmiştir. Sonuçlar incelendiğinde 192 adet test problemi içerisinde, Gen-2 algoritması 20 tane, UÖY algoritması 23 tane, HKKO algoritması 28 tane ve önerilen DAY algoritması ise 185 tane en iyi C_{max} değerini bulmayı başarmıştır.

Çizelge 17. 192 test probleminde makinelere göre ortalama sapma miktarları

Makine Sayısı	Ortalama %Sapma			
	Gen-2	UÖY	HKKO	DAY
2	10,1538	10,0444	9,8571	0,0000
3	15,3932	14,7634	13,9415	0,0076
5	19,1855	16,7278	15,5344	0,0092
8	21,3200	17,0883	16,2351	0,0032
10	21,6235	17,6799	15,0993	0,0000
15	21,6695	17,2455	14,1837	0,0194
20	22,4268	17,8074	15,1409	0,0018
25	23,3558	17,0457	14,5175	0,0675
Ortalama	19,3910	16,0503	14,3137	0,0136

Çizelge 17’de ise BATÇ problemleri üzerinde DAY ile kıyaslaması yapılan algoritmaların, 192 adet test problemi için makine sayılarına göre gruplandırılmış biçimde, bilinen en iyi C_{max} değerinden ortalama %Sapma değerleri verilmiştir. Gen-2 algoritması, ortalama %19,3910, UÖY algoritması, ortalama %16,0503 ve HKKO algoritması, ortalama %14,3137 sapma göstermiş iken önerilen DAY algoritması, ortalama %0,0136 ile diğer algoritmalarından önemli derecede düşük bir sapma göstermiştir. Çizelge 18’de ise BATÇ problemlerini çözmek için DAY ile kıyaslaması

yapılan algoritmaların çalıştırıldığı bilgisayar ortamı özellikleri ve maksimum işlem süreleri verilmiştir.

Çizelge 18. Bilgisayar ortamı ve maksimum işlem süreleri

Algoritma	Bilgisayar özellikleri	Maksimum işlem süresi (s)
HKKO	AMD Turion 64 2 GHz-1GB RAM	110
UÖY	PC Pentium IV-3.0 GHz-1 GB RAM	110
DAY	Intel Core i5-2.5 GHz-4 GB RAM	155
Gen-2	PC Pentium IV-3.0 GHz-1 GB RAM	156

Çizelge 18’de görüldüğü üzere, 192 adet BATÇ test probleminde algoritmaların maksimum işlem süreleri verilmiştir. En küçük işlem süresi 110 saniye ile Engin ve Güçlü (2018) tarafından önerilen HKKO algoritmasına ve Engin ve Günaydın (2011) tarafından önerilen UÖY algoritmasına aittir. İkinci en küçük işlem süresi ise 155 saniye ile bu çalışmada önerilen DAY algoritmasına aittir. Aldowaisan ve Allahverdi (2003) tarafından önerilen Gen-2 algoritması ise 156 saniye ile en büyük işlem süresini veren algoritma olmuştur.

Çizelge 24 ve Çizelge 25’te 8 ve 10 iş grubu problemler için bulanık teslim tarihlerine göre, bulunan iş sırası, her işe ait tatmin indeksi ve ortalama tatmin indeksi verilmiştir. Çizelge 26-28’de ise 12, 50, 100, 150, 200 ve 250 iş grubu problemler için bulanık teslim tarihlerine göre, ortalama tatmin indeksi verilmiştir.

Teslim tarihleri bulanıklaştırılırken $P(I-\tau-R/2)$ ve $P(I-\tau+R/2)$ aralığı dikkate alınmıştır. τ ve R teslim tarihi sıklığı ve teslim tarihi aralığını ifade etmektedir (Yüksel ve ark. 2020). \tilde{d}^1 , \tilde{d}^2 , \tilde{d}^3 ve \tilde{d}^4 değerleri belirlenirken sırası ile 0,1 0,5 0,9 ve 1,3 kat sayıları kullanılmış ve aşağıdaki 11, 12, 13 ve 14. denklemler ile hesaplama yapılmıştır (Fisher 1976).

$$\tilde{d}^1 = 0,1 \times dj \quad (11)$$

$$\tilde{d}^2 = 0,5 \times dj \quad (12)$$

$$\tilde{d}^3 = 0,9 \times dj \quad (13)$$

$$\tilde{d}^4 = 1,3 \times dj \quad (14)$$

Çizelge 21-23'te DAY ile çözülen 12, 50, 100, 150, 200 ve 250 iş grubu toplam 144 problem için tüm işlerin toplam ve ortalama akış süreleri, Çizelge 19-20'de ise 8 ve 10 iş problemleri için iş sırasına göre ayrıntılı akış süreleri verilmiştir.

Çizelge 19. 8 iş problemlerinde işlerin akış süreleri

Problem										Toplam	Ortalama
8x2-a	π :	i2	i7	i6	i8	i4	i5	i3	i1		
	F_j :	15	30	42	57	68	80	85	93	470	58,750
8x2-b	π :	i2	i5	i7	i4	i1	i6	i8	i3		
	F_j :	39	64	107	129	141	211	252	255	1198	149,750
8x2-c	π :	i6	i8	i5	i7	i4	i1	i2	i3		
	F_j :	145	216	252	328	370	387	460	465	2623	327,875
8x3-a	π :	i5	i6	i8	i4	i2	i7	i3	i1		
	F_j :	11	25	28	42	49	63	73	86	377	47,125
8x3-b	π :	i1	i3	i2	i8	i5	i6	i4	i7		
	F_j :	20	46	86	141	166	224	240	281	1204	150,500
8x3-c	π :	i1	i3	i2	i8	i4	i7	i5	i6		
	F_j :	68	156	223	304	379	408	525	558	2621	327,625
8x5-a	π :	i7	i3	i5	i8	i4	i1	i6	i2		
	F_j :	29	49	65	77	92	107	117	126	662	82,750
8x5-b	π :	i1	i3	i5	i8	i6	i4	i7	i2		
	F_j :	111	171	231	265	294	353	381	396	2202	275,250
8x5-c	π :	i1	i3	i5	i8	i6	i4	i7	i2		
	F_j :	215	334	445	506	551	658	717	744	4170	521,250
8x8-a	π :	i2	i1	i6	i3	i5	i7	i4	i8		
	F_j :	35	55	65	91	99	106	121	132	704	88,000
8x8-b	π :	i2	i7	i6	i4	i1	i3	i5	i8		
	F_j :	160	210	266	303	364	461	499	518	2781	347,625
8x8-c	π :	i2	i7	i4	i1	i3	i5	i6	i8		
	F_j :	307	403	489	611	796	866	916	990	5378	672,250
8x10-a	π :	i2	i4	i6	i7	i3	i5	i1	i8		
	F_j :	38	59	77	95	115	132	139	151	806	100,750
8x10-b	π :	i2	i7	i3	i5	i1	i4	i6	i8		
	F_j :	178	290	378	435	444	527	587	602	3441	430,125
8x10-c	π :	i2	i7	i6	i3	i5	i1	i4	i8		
	F_j :	354	569	668	859	969	983	1147	1162	6711	838,875
8x15-a	π :	i2	i1	i6	i4	i7	i3	i5	i8		
	F_j :	71	88	103	117	143	167	182	193	1064	133,000
8x15-b	π :	i2	i1	i4	i6	i7	i3	i5	i8		
	F_j :	320	387	434	501	627	680	734	773	4456	557,000
8x15-c	π :	i2	i1	i4	i6	i7	i3	i5	i8		
	F_j :	627	754	834	957	1204	1310	1403	1487	8576	1072,000
8x20-a	π :	i2	i6	i7	i3	i1	i4	i5	i8		
	F_j :	97	109	143	161	177	189	206	216	1298	162,250
8x20-b	π :	i2	i1	i7	i3	i4	i6	i5	i8		
	F_j :	456	526	634	691	727	787	854	887	5562	695,250
8x20-c	π :	i2	i4	i7	i3	i1	i6	i5	i8		
	F_j :	900	961	1275	1396	1513	1556	1681	1742	11024	1378,000
8x25-a	π :	i2	i7	i3	i1	i6	i5	i4	i8		
	F_j :	126	149	168	192	209	218	234	249	1545	193,125
8x25-b	π :	i2	i7	i3	i1	i6	i5	i4	i8		
	F_j :	630	700	773	859	908	932	983	1050	6835	854,375
8x25-c	π :	i2	i7	i3	i1	i6	i5	i4	i8		
	F_j :	1263	1385	1525	1702	1792	1833	1926	2051	13477	1684,625

Çizelge 20. 10 iş problemlerinde işlerin akış süreleri

Problem											Toplam	Ortalama	
10x2-a	π :	i9	i6	i5	i4	i10	i8	i7	i3	i1	i2		
	F_j :	16	28	40	51	60	73	85	93	99	110	655	65,5
10x2-b	π :	i2	i9	i5	i10	i4	i8	i3	i6	i7	i1		
	F_j :	22	63	103	147	208	246	266	313	328	337	2033	203,3
10x2-c	π :	i9	i10	i4	i8	i6	i5	i3	i7	i2	i1		
	F_j :	83	158	265	328	400	476	511	546	575	581	3923	392,3
10x3-a	π :	i1	i4	i9	i5	i6	i3	i7	i10	i8	i2		
	F_j :	16	29	46	61	67	74	87	103	113	117	713	71,3
10x3-b	π :	i2	i1	i9	i5	i10	i4	i8	i7	i6	i3		
	F_j :	28	81	126	169	216	249	282	328	345	362	2186	218,6
10x3-c	π :	i2	i1	i9	i5	i10	i4	i8	i7	i6	i3		
	F_j :	49	146	235	318	411	471	530	612	634	667	4073	407,3
10x5-a	π :	i3	i7	i5	i10	i4	i2	i1	i6	i9	i8		
	F_j :	27	42	57	63	75	93	101	119	128	138	843	84,3
10x5-b	π :	i2	i6	i7	i5	i10	i8	i3	i1	i9	i4		
	F_j :	114	153	206	257	283	314	366	392	438	462	2985	298,5
10x5-c	π :	i2	i1	i9	i8	i3	i6	i7	i5	i10	i4		
	F_j :	218	251	334	385	489	581	681	781	831	873	5424	542,4
10x8-a	π :	i2	i8	i3	i6	i1	i9	i7	i5	i10	i4		
	F_j :	30	47	58	81	97	119	132	141	158	169	1032	103,2
10x8-b	π :	i2	i8	i3	i6	i1	i9	i7	i5	i10	i4		
	F_j :	140	229	279	363	398	421	477	495	546	571	3919	391,9
10x8-c	π :	i2	i6	i1	i9	i7	i5	i10	i4	i8	i3		
	F_j :	264	438	526	698	757	801	904	932	1027	1076	7423	742,3
10x10-a	π :	i2	i6	i5	i1	i9	i7	i10	i4	i8	i3		
	F_j :	43	61	77	97	115	132	145	151	165	176	1162	116,2
10x10-b	π :	i2	i6	i1	i9	i7	i10	i5	i4	i8	i3		
	F_j :	192	261	346	410	462	484	546	551	600	612	4464	446,4
10x10-c	π :	i2	i6	i1	i9	i7	i10	i5	i4	i8	i3		
	F_j :	370	493	658	791	892	930	1042	1050	1147	1159	8532	853,2
10x15-a	π :	i6	i1	i9	i7	i5	i10	i2	i4	i3	i8		
	F_j :	66	93	111	132	145	158	174	187	203	216	1485	148,5
10x15-b	π :	i3	i8	i4	i2	i1	i9	i7	i5	i10	i6		
	F_j :	285	331	408	469	574	64	724	772	807	849	5283	528,3
10x15-c	π :	i2	i1	i9	i7	i5	i10	i6	i4	i8	i3		
	F_j :	595	810	950	1098	1188	1248	1322	1448	1524	1632	11815	1181,5
10x20-a	π :	i8	i3	i2	i4	i6	i1	i7	i9	i5	i10		
	F_j :	92	110	131	145	166	179	212	232	248	257	1772	177,2
10x20-b	π :	i2	i4	i8	i3	i6	i1	i9	i7	i5	i10		
	F_j :	447	482	547	599	695	739	861	940	1024	1043	7377	737,7
10x20-c	π :	i2	i4	i8	i3	i6	i1	i7	i9	i5	i10		
	F_j :	887	945	1071	1174	1358	1436	1727	1860	1993	2025	14476	1447,6
10x25-a	π :	i8	i4	i3	i2	i1	i9	i7	i6	i5	i10		
	F_j :	115	125	151	162	188	207	236	247	263	275	1969	196,9
10x25-b	π :	i3	i4	i8	i2	i6	i7	i1	i9	i5	i10		
	F_j :	544	572	678	737	828	1003	1038	1110	1156	1195	8861	886,1
10x25-c	π :	i8	i3	i4	i2	i1	i9	i7	i6	i5	i10		
	F_j :	1119	1227	1289	1473	1680	1831	2090	2160	2282	2346	17497	1749,7

Çizelge 21. 12 ve 50 iş problemlerinde işlerin akış süreleri

Problem	Toplam Akış Süresi	Ortalama Akış Süresi	Problem	Toplam Akış Süresi	Ortalama Akış Süresi
12x2-a	952	79,33	50x2-a	15163	303,26
12x2-b	2712	226,00	50x2-b	41851	837,02
12x2-c	4817	401,42	50x2-c	74002	1480,04
12x3-a	986	82,17	50x3-a	14406	288,12
12x3-b	3422	285,17	50x3-b	43518	870,36
12x3-c	6262	521,83	50x3-c	77855	1557,1
12x5-a	1206	100,50	50x5-a	16288	325,76
12x5-b	4336	361,33	50x5-b	51504	1030,08
12x5-c	8290	690,83	50x5-c	100176	2003,52
12x8-a	1504	125,33	50x8-a	18369	367,38
12x8-b	5505	458,75	50x8-b	62012	1240,24
12x8-c	10830	902,50	50x8-c	117134	2342,68
12x10-a	1441	120,08	50x10-a	19765	395,3
12x10-b	6066	505,50	50x10-b	68062	1361,24
12x10-c	11903	991,92	50x10-c	136445	2728,9
12x15-a	1915	159,58	50x15-a	23616	472,32
12x15-b	7940	661,67	50x15-b	83442	1668,84
12x15-c	15561	1296,75	50x15-c	167873	3357,46
12x20-a	2301	191,75	50x20-a	25211	504,22
12x20-b	9893	824,42	50x20-b	97690	1953,8
12x20-c	20134	1677,83	50x20-c	191259	3825,18
12x25-a	2627	218,92	50x25-a	28052	561,04
12x25-b	11975	997,92	50x25-b	111179	2223,58
12x25-c	23197	1933,08	50x25-c	211916	4238,32

Çizelge 22. 100 ve 150 iş problemlerinde işlerin akış süreleri

Problem	Toplam Akış Süresi	Ortalama Akış Süresi	Problem	Toplam Akış Süresi	Ortalama Akış Süresi
100x2-a	53406	534,06	150x2-a	123357	822,38
100x2-b	166432	1664,32	150x2-b	366477	2443,18
100x2-c	313218	3132,18	150x2-c	666690	4444,60
100x3-a	58533	585,33	150x3-a	132437	882,91
100x3-b	175712	1757,12	150x3-b	394549	2630,33
100x3-c	307029	3070,29	150x3-c	734838	4898,92
100x5-a	64498	644,98	150x5-a	143557	957,05
100x5-b	201827	2018,27	150x5-b	447157	2981,05
100x5-c	372588	3725,88	150x5-c	858687	5724,58
100x8-a	71726	717,26	150x8-a	156441	1042,94
100x8-b	237971	2379,71	150x8-b	520544	3470,29
100x8-c	450158	4501,58	150x8-c	983054	6553,69
100x10-a	75127	751,27	150x10-a	167414	1116,09
100x10-b	258303	2583,03	150x10-b	570776	3805,17
100x10-c	473546	4735,46	150x10-c	1088893	7259,29
100x15-a	84008	840,08	150x15-a	184142	1227,61
100x15-b	296640	2966,4	150x15-b	647048	4313,65
100x15-c	573028	5730,28	150x15-c	1242092	8280,61
100x20-a	92084	920,84	150x20-a	197307	1315,38
100x20-b	338811	3388,11	150x20-b	706448	4709,65
100x20-c	656847	6568,47	150x20-c	1381629	9210,86
100x25-a	99729	997,29	150x25-a	211729	1411,53
100x25-b	375488	3754,88	150x25-b	783221	5221,47
100x25-c	721945	7219,45	150x25-c	1506502	10043,35

Çizelge 23. 200 ve 250 iş problemlerinde işlerin akış süreleri

Problem	Toplam Akış Süresi	Ortalama Akış Süresi	Problem	Toplam Akış Süresi	Ortalama Akış Süresi
200x2-a	219698	1098,49	250x2-a	359228	1436,91
200x2-b	680487	3402,44	250x2-b	1058202	4232,81
200x2-c	1253529	6267,65	250x2-c	1928045	7712,18
200x3-a	230896	1154,48	250x3-a	360103	1440,41
200x3-b	725582	3627,91	250x3-b	1119024	4476,01
200x3-c	1309650	6548,25	250x3-c	2073511	8294,04
200x5-a	254237	1271,19	250x5-a	394038	1576,15
200x5-b	819599	4098,00	250x5-b	1240056	4960,22
200x5-c	1538338	7691,69	250x5-c	2334841	9339,36
200x8-a	279414	1397,07	250x8-a	426073	1704,29
200x8-b	906664	4533,32	250x8-b	1414946	5659,78
200x8-c	1725571	8627,86	250x8-c	2659297	10637,19
200x10-a	288126	1440,63	250x10-a	452285	1809,14
200x10-b	984192	4920,96	250x10-b	1506805	6027,22
200x10-c	1853714	9268,57	250x10-c	2839438	11357,75
200x15-a	318263	1591,32	250x15-a	487981	1951,92
200x15-b	1107737	5538,69	250x15-b	1691062	6764,25
200x15-c	2107182	10535,91	250x15-c	3265497	13061,99
200x20-a	341463	1707,32	250x20-a	522227	2088,91
200x20-b	1206900	6034,50	250x20-b	1875401	7501,60
200x20-c	2328517	11642,59	250x20-c	3653295	14613,18
200x25-a	361960	1809,80	250x25-a	555988	2223,95
200x25-b	1333355	6666,78	250x25-b	2034336	8137,34
200x25-c	2606553	13032,77	250x25-c	3916994	15667,98

Çizelge 24. 8 iş problemlerinde ortalama tatmin indeksleri

Problem										Ortalama
8x2-a	İş Sırası	i2	i7	i6	i8	i4	i5	i3	i1	
	Tatmin İndeksi	1,0000	0,7500	0,8081	0,7931	0,7500	0,6526	0,6898	0,5776	0,7526
8x2-b	İş Sırası	i2	i5	i7	i4	i1	i6	i8	i3	
	Tatmin İndeksi	0,6842	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,9605
8x2-c	İş Sırası	i6	i8	i5	i7	i4	i1	i2	i3	
	Tatmin İndeksi	0,6791	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,9599
8x3-a	İş Sırası	i5	i6	i8	i4	i2	i7	i3	i1	
	Tatmin İndeksi	0,0000	0,1250	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,7656
8x3-b	İş Sırası	i1	i3	i2	i8	i5	i6	i4	i7	
	Tatmin İndeksi	1,0000	1,0000	1,0000	1,0000	0,9157	1,0000	0,9083	0,9654	0,9737
8x3-c	İş Sırası	i1	i3	i2	i8	i4	i7	i5	i6	
	Tatmin İndeksi	0,5938	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,9599	0,9442
8x5-a	İş Sırası	i7	i3	i5	i8	i4	i1	i6	i2	
	Tatmin İndeksi	0,2292	0,9815	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,9013
8x5-b	İş Sırası	i1	i3	i5	i8	i6	i4	i7	i2	
	Tatmin İndeksi	0,6565	1,0000	1,0000	0,8616	0,7708	0,7738	0,7259	0,6884	0,8096
8x5-c	İş Sırası	i1	i3	i5	i8	i6	i4	i7	i2	
	Tatmin İndeksi	0,7265	1,0000	1,0000	0,8104	0,7073	0,7053	0,6697	0,6340	0,7816
8x8-a	İş Sırası	i2	i1	i6	i3	i5	i7	i4	i8	
	Tatmin İndeksi	0,3333	1,0000	1,0000	1,0000	0,9632	0,8496	0,8304	0,7813	0,8447
8x8-b	İş Sırası	i2	i7	i6	i4	i1	i3	i5	i8	
	Tatmin İndeksi	0,6010	1,0000	1,0000	0,8109	0,7679	0,7522	0,6404	0,5569	0,7662
8x8-c	İş Sırası	i2	i7	i4	i1	i3	i5	i6	i8	
	Tatmin İndeksi	0,6745	1,0000	0,9334	0,8458	0,7913	0,6469	0,5676	0,5217	0,7477
8x10-a	İş Sırası	i2	i4	i6	i7	i3	i5	i1	i8	
	Tatmin İndeksi	0,5357	1,0000	1,0000	0,9806	0,8864	0,8010	0,7344	0,7009	0,8299
8x10-b	İş Sırası	i2	i7	i3	i5	i1	i4	i6	i8	
	Tatmin İndeksi	0,7216	1,0000	1,0000	0,7740	0,6366	0,6231	0,5721	0,4999	0,7284
8x10-c	İş Sırası	i2	i7	i6	i3	i5	i1	i4	i8	
	Tatmin İndeksi	0,6996	1,0000	0,9277	0,8063	0,6556	0,5544	0,5534	0,4744	0,7089
8x15-a	İş Sırası	i2	i1	i6	i4	i7	i3	i5	i8	
	Tatmin İndeksi	0,3871	1,0000	1,0000	0,8333	0,7542	0,7032	0,6284	0,5833	0,7362
8x15-b	İş Sırası	i2	i1	i4	i6	i7	i3	i5	i8	
	Tatmin İndeksi	0,7022	1,0000	0,8482	0,6723	0,6282	0,5213	0,4515	0,4132	0,6546
8x15-c	İş Sırası	i2	i1	i4	i6	i7	i3	i5	i8	
	Tatmin İndeksi	0,7460	1,0000	0,8062	0,6319	0,5941	0,4928	0,4201	0,3883	0,6349
8x20-a	İş Sırası	i2	i6	i7	i3	i1	i4	i5	i8	
	Tatmin İndeksi	0,5253	1,0000	0,9743	0,7588	0,6494	0,5605	0,5074	0,4671	0,6803
8x20-b	İş Sırası	i2	i1	i7	i3	i4	i6	i5	i8	
	Tatmin İndeksi	0,7167	1,0000	0,8327	0,6103	0,4900	0,4231	0,3757	0,3387	0,5984
8x20-c	İş Sırası	i2	i4	i7	i3	i1	i6	i5	i8	
	Tatmin İndeksi	0,7444	1,0000	0,8518	0,6256	0,5201	0,4154	0,3658	0,3286	0,6065
8x25-a	İş Sırası	i2	i7	i3	i1	i6	i5	i4	i8	
	Tatmin İndeksi	0,6890	1,0000	0,8269	0,6839	0,5639	0,4643	0,4201	0,3911	0,6299
8x25-b	İş Sırası	i2	i7	i3	i1	i6	i5	i4	i8	
	Tatmin İndeksi	0,7460	1,0000	0,7295	0,5782	0,4506	0,3549	0,3093	0,2881	0,5571
8x25-c	İş Sırası	i2	i7	i3	i1	i6	i5	i4	i8	
	Tatmin İndeksi	0,7421	1,0000	0,7149	0,5698	0,4409	0,3445	0,2976	0,2758	0,5482

Çizelge 25. 10 iş problemlerinde ortalama tatmin indeksleri

Problem												Ortalama
10x2-a	İş Sırası	i9	i6	i5	i4	i10	i8	i7	i3	i1	i2	
	Tatmin İndeksi	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,838
10x2-b	İş Sırası	i2	i9	i5	i10	i4	i8	i3	i6	i7	i1	
	Tatmin İndeksi	0,355	0,581	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
10x2-c	İş Sırası	i9	i10	i4	i8	i6	i5	i3	i7	i2	i1	
	Tatmin İndeksi	0,688	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
10x3-a	İş Sırası	i1	i4	i9	i5	i6	i3	i7	i10	i8	i2	
	Tatmin İndeksi	0,173	1,000	0,903	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
10x3-b	İş Sırası	i2	i1	i9	i5	i10	i4	i8	i7	i6	i3	
	Tatmin İndeksi	0,333	0,895	1,000	1,000	1,000	1,000	0,955	1,000	0,948	0,923	
10x3-c	İş Sırası	i2	i1	i9	i5	i10	i4	i8	i7	i6	i3	
	Tatmin İndeksi	0,587	1,000	1,000	1,000	1,000	0,952	0,886	0,919	0,855	0,841	
10x5-a	İş Sırası	i3	i7	i5	i10	i4	i2	i1	i6	i9	i8	
	Tatmin İndeksi	0,438	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	
10x5-b	İş Sırası	i2	i6	i7	i5	i10	i8	i3	i1	i9	i4	
	Tatmin İndeksi	0,635	1,000	1,000	0,871	0,722	0,674	0,697	0,666	0,671	0,628	
10x5-c	İş Sırası	i2	i1	i9	i8	i3	i6	i7	i5	i10	i4	
	Tatmin İndeksi	0,727	1,000	1,000	0,817	0,826	0,782	0,749	0,698	0,627	0,582	
10x8-a	İş Sırası	i2	i8	i3	i6	i1	i9	i7	i5	i10	i4	
	Tatmin İndeksi	0,365	1,000	1,000	1,000	1,000	1,000	0,959	0,866	0,844	0,814	
10x8-b	İş Sırası	i2	i8	i3	i6	i1	i9	i7	i5	i10	i4	
	Tatmin İndeksi	0,599	1,000	1,000	0,876	0,699	0,577	0,544	0,486	0,483	0,461	
10x8-c	İş Sırası	i2	i6	i1	i9	i7	i5	i10	i4	i8	i3	
	Tatmin İndeksi	0,712	1,000	0,971	0,837	0,654	0,538	0,504	0,443	0,440	0,421	
10x10-a	İş Sırası	i2	i6	i5	i1	i9	i7	i10	i4	i8	i3	
	Tatmin İndeksi	0,264	1,000	1,000	1,000	0,886	0,804	0,735	0,671	0,665	0,659	
10x10-b	İş Sırası	i2	i6	i1	i9	i7	i10	i5	i4	i8	i3	
	Tatmin İndeksi	0,669	1,000	0,995	0,771	0,633	0,514	0,480	0,413	0,407	0,378	
10x10-c	İş Sırası	i2	i6	i1	i9	i7	i10	i5	i4	i8	i3	
	Tatmin İndeksi	0,743	1,000	0,940	0,737	0,603	0,484	0,447	0,382	0,379	0,346	
10x15-a	İş Sırası	i6	i1	i9	i7	i5	i10	i2	i4	i3	i8	
	Tatmin İndeksi	0,672	1,000	0,989	0,815	0,677	0,598	0,580	0,542	0,532	0,512	
10x15-b	İş Sırası	i3	i8	i4	i2	i1	i9	i7	i5	i10	i6	
	Tatmin İndeksi	0,696	1,000	0,881	0,728	0,658	0,000	0,491	0,424	0,373	0,348	
10x15-c	İş Sırası	i2	i1	i9	i7	i5	i10	i6	i4	i8	i3	
	Tatmin İndeksi	0,733	1,000	0,826	0,640	0,508	0,418	0,374	0,358	0,333	0,325	
10x20-a	İş Sırası	i8	i3	i2	i4	i6	i1	i7	i9	i5	i10	
	Tatmin İndeksi	0,576	1,000	1,000	0,789	0,683	0,580	0,562	0,509	0,461	0,412	
10x20-b	İş Sırası	i2	i4	i8	i3	i6	i1	i9	i7	i5	i10	
	Tatmin İndeksi	0,727	1,000	0,780	0,612	0,534	0,438	0,413	0,364	0,334	0,285	
10x20-c	İş Sırası	i2	i4	i8	i3	i6	i1	i7	i9	i5	i10	
	Tatmin İndeksi	0,739	1,000	0,761	0,596	0,517	0,420	0,413	0,358	0,319	0,269	
10x25-a	İş Sırası	i8	i4	i3	i2	i1	i9	i7	i6	i5	i10	
	Tatmin İndeksi	0,683	1,000	0,880	0,673	0,585	0,495	0,452	0,393	0,355	0,322	
10x25-b	İş Sırası	i3	i4	i8	i2	i6	i7	i1	i9	i5	i10	
	Tatmin İndeksi	0,732	1,000	0,787	0,598	0,495	0,463	0,377	0,328	0,281	0,246	
10x25-c	İş Sırası	i8	i3	i4	i2	i1	i9	i7	i6	i5	i10	
	Tatmin İndeksi	0,737	1,000	0,736	0,597	0,502	0,412	0,372	0,312	0,274	0,237	

Çizelge 26. 12 ve 50 iş problemlerinde ortalama tatmin indeksleri

Problem	Ortalama Tatmin İndeksi	Problem	Ortalama Tatmin İndeksi
12x2-a	0,6333	50x2-a	0,7181
12x2-b	0,8958	50x2-b	0,9838
12x2-c	0,9100	50x2-c	0,9940
12x3-a	0,9643	50x3-a	0,9588
12x3-b	0,9520	50x3-b	0,9389
12x3-c	0,9211	50x3-c	0,8620
12x5-a	0,9420	50x5-a	0,9867
12x5-b	0,7442	50x5-b	0,6160
12x5-c	0,7261	50x5-c	0,5574
12x8-a	0,8248	50x8-a	0,6876
12x8-b	0,6496	50x8-b	0,7519
12x8-c	0,5964	50x8-c	0,7107
12x10-a	0,7598	50x10-a	0,5699
12x10-b	0,6591	50x10-b	0,5257
12x10-c	0,6092	50x10-c	0,5202
12x15-a	0,6366	50x15-a	0,6145
12x15-b	0,5376	50x15-b	0,5124
12x15-c	0,5306	50x15-c	0,5114
12x20-a	0,5702	50x20-a	0,5105
12x20-b	0,6052	50x20-b	0,5387
12x20-c	0,5060	50x20-c	0,5400
12x25-a	0,5604	50x25-a	0,5442
12x25-b	0,5886	50x25-b	0,5041
12x25-c	0,5701	50x25-c	0,5339

Çizelge 24'te, 8 iş grubu problemlerde, en düşük tatmin indeksi 0,5482 ile 8x25-c problemine ait iken en yüksek tatmin indeksi 0,9737 ile 8x3-b problemine aittir. Çizelge 25'te ise 10 iş grubu problemlerde en düşük tatmin indeksi 0,5177 ile 10x25-c problemine ait iken en yüksek tatmin indeksi 0,9838 ile 10x2-a problemine aittir. Çizelge 26'da 12 iş grubu problemlerde en düşük tatmin indeksi 0,5060 ile 12x20-c problemine ait iken en yüksek tatmin indeksi 0,9643 ile 12x3-a problemine aittir. Çizelge 26'da 50 iş grubu problemlerde en düşük tatmin indeksi 0,5041 ile 50x25-b problemine ait iken en yüksek tatmin indeksi 0,9940 ile 50x2-c problemine aittir.

Çizelge 27. 100 ve 150 iş problemlerinde ortalama tatmin indeksleri

Problem	Ortalama Tatmin İndeksi	Problem	Ortalama Tatmin İndeksi
100x2-a	0,7600	150x2-a	0,7674
100x2-b	0,9927	150x2-b	0,9969
100x2-c	0,9952	150x2-c	0,9937
100x3-a	0,9641	150x3-a	0,9865
100x3-b	0,8764	150x3-b	0,8643
100x3-c	0,8050	150x3-c	0,7792
100x5-a	0,9494	150x5-a	0,9031
100x5-b	0,5544	150x5-b	0,5476
100x5-c	0,5099	150x5-c	0,5458
100x8-a	0,6056	150x8-a	0,6197
100x8-b	0,6702	150x8-b	0,5307
100x8-c	0,6205	150x8-c	0,5874
100x10-a	0,6710	150x10-a	0,6224
100x10-b	0,5289	150x10-b	0,5347
100x10-c	0,5338	150x10-c	0,5532
100x15-a	0,6128	150x15-a	0,5716
100x15-b	0,5222	150x15-b	0,5967
100x15-c	0,5241	150x15-c	0,5556
100x20-a	0,5170	150x20-a	0,5279
100x20-b	0,5198	150x20-b	0,5115
100x20-c	0,5532	150x20-c	0,5755
100x25-a	0,5346	150x25-a	0,5330
100x25-b	0,5365	150x25-b	0,5942
100x25-c	0,5189	150x25-c	0,5050

Çizelge 27’de 100 iş grubu problemler incelendiğinde, en düşük tatmin indeksi 0,5099 ile 100x5-c problemine ait iken en yüksek tatmin indeksi 0,9952 ile 100x2-c problemine aittir. Çizelge 27’de 150 iş grubu problemler incelendiğinde, en düşük tatmin indeksi 0,5050 ile 150x25-c problemine ait iken en yüksek tatmin indeksi 0,9969 ile 150x2-b problemine aittir.

Çizelge 28. 200 ve 250 iş problemlerinde ortalama tatmin indeksleri

Problem	Ortalama Tatmin İndeksi	Problem	Ortalama Tatmin İndeksi
200x2-a	0,7178	250x2-a	0,7679
200x2-b	0,9960	250x2-b	0,9984
200x2-c	0,9985	250x2-c	0,9950
200x3-a	0,9946	250x3-a	0,9902
200x3-b	0,8533	250x3-b	0,8385
200x3-c	0,7726	250x3-c	0,7562
200x5-a	0,9253	250x5-a	0,8848
200x5-b	0,5105	250x5-b	0,5074
200x5-c	0,6132	250x5-c	0,5106
200x8-a	0,5841	250x8-a	0,5413
200x8-b	0,5385	250x8-b	0,5090
200x8-c	0,5482	250x8-c	0,5185
200x10-a	0,5761	250x10-a	0,5103
200x10-b	0,5638	250x10-b	0,5195
200x10-c	0,5121	250x10-c	0,5100
200x15-a	0,5391	250x15-a	0,5092
200x15-b	0,5476	250x15-b	0,5188
200x15-c	0,5475	250x15-c	0,5237
200x20-a	0,5019	250x20-a	0,5076
200x20-b	0,5587	250x20-b	0,5044
200x20-c	0,5093	250x20-c	0,5073
200x25-a	0,5385	250x25-a	0,5248
200x25-b	0,5952	250x25-b	0,5212
200x25-c	0,5579	250x25-c	0,5358

Çizelge 28’de 200 iş grubu problemler incelendiğinde en düşük tatmin indeksi 0,5019 ile 200x20-a problemine ait iken en yüksek tatmin indeksi 0,9985 ile 200x2-c problemine aittir. Çizelge 28’de 250 iş grubu problemler incelendiğinde en düşük tatmin indeksi 0,5044 ile 250x20-b problemine ait iken en yüksek tatmin indeksi 0,9984 ile 250x2-b problemine aittir.

Ortaya çıkan sonuçlar incelendiğinde, bir işin tatmin indeksinin kendinden sonra gelen işlerin tatmin indekslerini etkilediği görülmektedir. Kötü çıkan bir tatmin indeksi sonucunda çok yüksek bir oran ile sonraki işlerde tatmin sonuçları daha da kötüye gitme eğilimindedir. Çizelge 29’da iş sayısına göre tüm problemlerin ortalama tatmin indeksi sonuçları verilmiştir.

Çizelge 29. İş sayısına göre ortalama tatmin indeksleri

İş Sayısı	Ortalama Tatmin İndeksi
8	0,7550
10	0,7264
12	0,7035
50	0,6746
100	0,6615
150	0,6585
200	0,6500
250	0,6254
Ortalama	0,6819

Çizelge 29’da görüldüğü üzere 8 iş grubu problemlerin ortalama tatmin indeksi 0,7550 çıkmış iken 250 iş grubu problemlerin ortalama tatmin indeksi ise 0,6254 çıkmıştır. İş sayısının 8’den 250’ye çıkması sonucunda, ortalama tatmin indeksinde %20,72’lik bir kötüleşmeye neden olduğu görülmüştür.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

BATÇ problemleri, polinomial olarak çözülemeyen problemler arasında yer almaktadır. Günlük yaşamda, geniş bir kullanım alanı bulunan BATÇ problemleri için kabul edilebilir bir sürede, optimum çözümü bulabilecek bir matematiksel yöntem bulunmamaktadır. Bu nedenlerden dolayı, BATÇ problemlerinin, kabul edilebilir bir sürede, optimum veya optimum çözüme yakın sonuçlar bulabilmek için sezgisel yöntemlerin kullanımına başvurulmuştur. Bu çalışmada ise evrimsel bir yaklaşım olan Dağınık Arama meta-sezgisel yöntemi, BATÇ problemine uyarlanmıştır. Gerçek yaşam koşullarında, teslim tarihlerinin belirsiz olmasından dolayı, bu çalışmada teslim tarihlerinin bulanıklaştırılması yapılarak gerçek yaşam koşullarına uygun bir çalışma yapılmıştır.

BATÇ problemleri üzerinde DAY'nin etkinliğini ispatlamak için, literatürde iyi bilinen 192 adet test problemi, maksimum tamamlanma zamanı amaç fonksiyonuna göre çözülmüş ve elde edilen sonuçlar BATÇ problemi için yapılmış olan üç sezgisel yöntemin sonuçları ile kıyaslanmıştır. Kıyaslaması yapılan algoritmaların tüm problemler için ortalama %Sapma değeri dikkate alındığında, Gen-2, %19,3910, UÖY, %16,0503, HKKO, %14,3137 ve bu çalışmada önerilen DAY algoritması, %0,0136 sapma göstermiştir. Elde edilen sonuçlara göre DAY'nin kıyaslaması yapılan diğer yöntemlerden oldukça verimli sonuçlar verdiği görülmüştür.

Yapılan çalışmada, ele alınan toplam 192 problem için, DAY ile bulunan en iyi iş sırasına göre işlerin; akış süreleri, toplam akış süreleri ve ortalama akış süreleri hesaplanmıştır. Teslim tarihlerinin bulanıklaştırılarak DAY ile çözümü sunucunda 8, 10, 12, 50, 100, 150, 200 ve 250 iş grubu problemlerinin ortalama tatmin indeksi, sırası ile 0,7550, 0,7264, 0,7035, 0,6746, 0,6615, 0,6585, 0,6500 ve 0,6254 olarak hesaplanmıştır. 8 iş grubu problemlerinin ortalama tatmin indeksine göre; 10, 12, 50, 100, 150, 200 ve 250 iş grubu problemlerinin ortalama tatmin indeksinde sırası ile %3,94, %7,33, %11,92, %14,14, %14,67, %16,16 ve %20,72'lik bir kötüleşme gözlemlenmiştir. 192 problemin tamamında ise ortalama tatmin indeksi 0,6819 olarak hesaplanmıştır.

Maksimum işlem süresi dikkate alınarak algoritmaların kıyası yapıldığında ise 110 saniye ile en küçük işlem süresi HKKO ve UÖY'na aittir. En büyük işlem süresi ise

156 saniye ile Gen-2 algoritmasına aittir. DAY yöntemine ait maksimum işlem süresi ise 155 saniye olmuştur ve diğer yöntemlerden önemli bir farklılığın olmadığı görülmüştür. DAY en iyi işlem süresini veren HKKO algoritmasından ortalama %14,3 UÖY algoritmasından ise %16,0 daha iyi sonuç vermektedir.

5.2 Öneriler

Gelecekte yapılacak olan çalışmalarda DAY'nin, çok amaçlı BATÇ problemleri için amaç fonksiyonlarının eş zamanlı optimizasyonunda kullanılabileceği önerilmektedir. BATÇ problemlerinin DAY ile çözümünde, ilk popülasyonun üretilmesi için farklı başlangıç çözümü oluşturma teknikleri kullanılarak daha kaliteli bir çözüm ile algoritmanın başlatılması, işlem süresinin azalmasına katkı sağlayacaktır. DAY'nin çözüm iyileştirme aşamasında ise farklı yapıcı sezgisel yöntem kullanımı, amaç fonksiyonunun global optimum çözüme yaklaşmasına katkı sağlayacaktır. DAY ile BATÇ'nin, gerçek yaşam problemler üzerinde çalışılması, pratikte nasıl sonuçlar vereceği hakkında önemli bir bilgi sahibi olunması açısından literatüre bir katkı sağlayacağı düşünülmektedir. Ayrıca DAY'nin literatürde etkinliği ispatlanmış diğer meta-sezgisel algoritmalar ile birleştirilerek hibrit bir yöntem olarak kullanılması, çözüm kalitesinde olumlu sonuçlar ortaya çıkaracaktır.

KAYNAKLAR

- Akhshabi, M., Moghaddam, R. T. & Roodposhti, F. R., 2014, A hybrid particle swarm optimization algorithm for a no-wait flow shop scheduling problem with the total flow time, *Int J Adv Manuf Technol*, 70, 1181-1188.
- Aldowaisan, T. & Allahverdi, A., 2003, New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*, 30(8), 1219-1231.
- Allahverdi A. & Aydilek H., 2013, Algorithms for no-wait flowshops with total completion time subject to makespan, *Int J Adv Manuf Technol*, 68, 2237-2251.
- Allahverdi A. & Aydilek H., 2014, Total completion time with makespan constraint in no-wait flowshops with setup times, *European Journal of Operational Research*, 238, 724-734.
- Allahverdi, A., Aydilek, H. & Aydilek, A., 2020, No-wait flowshop scheduling problem with separate setup times to minimize total tardiness subject to makespan, *Applied Mathematics and Computation*, 365, 124688.
- Allahverdi, M., Aydilek, H., Aydilek, A. & Allahverdi, A., 2021, A better dominance relation and heuristics for two-machine no-wait flowshops with maximum lateness performance measure, *Journal of Industrial & Management Optimization*, 17(4), 1973-1991.
- Alsaidi, S. A. A. A., Muhsen, D. K. & Ali, S. M., 2020, Improved scatter search algorithm based on meerkat clan algorithm to solve NP-hard problems, *Periodicals of Engineering and Natural Sciences (PEN)*, 8(3), 1555-1565.
- Araújo, D. C. & Nagano, M. S., 2015, New heuristics for the no-wait flowshop with sequence-dependent setup times problem, *Brazilian Journal of Operations & Production Management*, 12(2), 180-194.
- Asefi, H., Jolai, F., Rabiee, M. & Araghi, M. E. T., 2014, A hybrid NSGA-II and VNS for solving a bi-objective no-wait flexible flowshop scheduling problem, *Int J Adv Manuf Technol*, 75, 1017-1033.
- Aydilek, H. & Allahverdi, A., 2012, Heuristics for no-wait flowshops with makespan subject to mean completion time, *Applied Mathematics and Computation*, 219, 351-359.
- Başar, R. & Engin, O., 2021, A No-Wait Flow Shop Scheduling Problem with Setup Time in Fuzzy Environment. *In International Conference on Intelligent and Fuzzy Systems, Springer, INFUS, Cham*, 607-614.
- Benavides, A. J., Ritt, M. & Miralles, C., 2014, Flow shop scheduling with heterogeneous workers, *European Journal of Operational Research*, 237(2), 713-720.

- Bewoor, L. A., Chandraprakash, V. & Sapkal, S. U., 2019, Evolutionary hybrid particle swarm optimization algorithm to minimize makespan to schedule a flow shop with no wait, *Journal of Engineering Science and Technology*, 14(2), 609-628.
- Bewoor, L. A., Prakash, V. C. & Sapkal, S. U., 2017, Evolutionary hybrid particle swarm optimization algorithm for solving Np-hard no-wait flow shop scheduling problems, *Algorithms*, 10(121), 17 pages.
- Bożejko, W. & Wodecki, M., 2007, Parallel scatter search algorithm for the flow shop sequencing problem, *International Conference on Parallel Processing and Applied Mathematics*, Springer, Berlin, Heidelberg, 180-188.
- Chaudhry, I.A., Elbadawi, I.A.Q., Usman, M., Chughtai, M.T., 2018, Minimizing total flowtime in a no-wait flow shop (NWFS) using genetic algorithms, *Ingeniería E Investigación*, 38(3), 68-79.
- Chen, K., Li, D. & Wang, X., 2020, Makespan minimization in two-machine flow-shop scheduling under no-wait and deterministic unavailable interval constraints, *Journal of Systems Science and Systems Engineering*, 29(4), 400-411.
- Chihaoui, F.B., Kacem, I., Hadj-Alouane, A.B., Dridi, N. & Rezg, N., 2011, No-wait scheduling of a two-machine flow-shop to minimize the makespan under non-availability constraints and different release dates, *International Journal of Production Research*, 49(21), 6273-6286.
- Czogalla, J., Fink, A., 2012, Fitness landscape analysis for the no-wait flow-shop scheduling problem, *J Heuristics*, 18, 25-51.
- Çiçekli, U. G. & Bozkurt, S., 2016, Permütasyon akış tipi çizelgeleme probleminin dağınık arama ile optimizasyonu, *Ege Academic Review*, 16.
- Davendra, D., Zelinka, I., Senkerik, R. & Jasek, R., 2013, Discrete self-organizing migrating algorithm for flow shop scheduling with no wait makespan, *Mathematical and Computer Modelling*, 57, 100-110.
- Deng, G., Wei, M., Su, Q. & Zhao, M., 2015, An effective co-evolutionary quantum genetic algorithm for the no-wait flow shop scheduling problem, *Advances in Mechanical Engineering*, 7(12), 1-10.
- Ding, J.Y., Song, S., Gupta, J. N., Zhang, R., Chiong, R. & Wu, C., 2015a, An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem, *Applied Soft Computing*, 30, 604-613.
- Ding, J.Y., Song, S., Zhang, R., Gupta, J. N. & Wu, C., 2015b, Accelerated methods for total tardiness minimization in no-wait flow shops, *International Journal of Production Research*, 53(4), 1002-1018.
- Engin, O. & Güçlü, A., 2018, A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems, *Applied Soft Computing*, 72, 166-176

- Engin, O. & Günaydın, C., 2011, An adaptive learning approach for no-wait flowshop scheduling problems to minimize makespan, *International Journal of Computational Intelligence Systems*, 4(4), 521-529.
- Engin, O., Kahraman C. & Yilmaz, M. K., 2009, A scatter search method for multiobjective fuzzy permutation flow shop scheduling problem: a real-world application, *U.K. Chakraborty (Ed.): Comput. Intel. in Flow Shop and Job Shop Sched.*, SCI 230, 169-189.
- Engin, O., Yılmaz, M. K., Baysal, M. E. & Sarucan, A., 2013, Solving Fuzzy Job Shop Scheduling Problems with Availability Constraints Using a Scatter Search Method, *J. of Multi-Valued Logic and Soft Computing*, 21, 317-334.
- Espinouse, M.L., Formanowicz, P. & Penz, B., 1999, Minimizing the makespan in the two-machine no-wait flow- shop with limited machine availability, *Computers & Industrial Engineering*, 37, 497-500.
- Ferretti, I. & Zavanella, L. E., 2020, Batch energy scheduling problem with no-wait/blocking constraints for the general flow-shop problem, *Procedia Manufacturing*, 42, 273-280.
- Fisher, M. L., 1976, A dual algorithm for the one-machine scheduling problem, *Mathematical programming*, 11(1), 229-251.
- Franca-Vieira, L., Rodriguez-Esquerre, V. F., Sisnando, A. D. & Rubio-Mercedes, C. E. 2020, Inverse design of a taper by scatter search metaheuristic, *IEEE Photonics Journal*, 12(4), 1-9.
- Gao, F., Liu, M., Wang, J. J. & Lu, Y. Y., 2018, No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times, *International Journal of Production Research*, 56(6), 2361-2369.
- Gao, K., Pan, Q. & Li, J., 2011, Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion, *Int J Adv Manuf Technol*, 56, 683-692.
- Gao, K., Pan, Q., Suganthan, P. N. & Li, J., 2013, Effective heuristics for the no-wait flow shop scheduling problem with total flow time minimization, *Int J Adv Manuf Technol*, 66, 1563-1572.
- Geng, X. N., Wang, J. B. & Bai, D., 2019, Common due date assignment scheduling for a no-wait flowshop with convex resource allocation and learning effect, *Engineering Optimization*, 51(8), 1301-1323.
- González, A. M., Vela, C. L. & Varela, R., 2015, Scatter search with path relinking for the flexible job shop scheduling problem, *European Journal of Operational Research*, 245, 35-45.
- Guo, Q. & Tang, L., 2015, An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times, *Applied Soft Computing*, 29, 184-195.

- Haq, A. N., Saravanan, M., Vivekraj, A. R. & Prasad, T., 2007, A scatter search approach for general flowshop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, 31(7-8), 731-736.
- Jafarzadeh, H., Moradinasab, N. & Gerami, A., 2017, Solving no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines and rework time by the adjusted discrete multi objective invasive weed optimization and fuzzy dominance approach, *Journal of Industrial Engineering and Management JIEM*, 10(5), 887-918.
- Jolai, F., Rabiee M. & Asefi, H., 2012, A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times, *International Journal of Production Research*, 50(24), 7447-7466.
- Kaya, S., Akgöbek Ö. & Engin, O., 2011, Erken ve geç cezalı permütasyon akış tipi çizelgeleme problemlerinin dağınık arama meta sezgiseli ile çözümü, *NWSA e-Journal of New World Science Academy, Engineering Sciences*, 6(1), 415- 420.
- Keskin, K. & Engin, O., 2021, A hybrid genetic local and global search algorithm for solving no-wait flow shop problem with bi criteria, *SN Appl Sci*, 3(6), 1-15.
- Komaki, M. & Malakooti, B., 2017, General variable neighborhood search algorithm to minimize makespan of the distributed no-wait flow shop scheduling problem, *Prod. Eng. Res. Devel.*, 11, 315-329.
- Koulamas, C. & Kyparisis, G. J., 2021, The no-wait flow shop with rejection, *International Journal of Production Research*, 59(6), 1852-1859.
- Koulamas, C. & Panwalkar, S.S., 2018, New index priority rules for no-wait flow shops, *Computers & Industrial Engineering*, 115, 647-652.
- Kökçam, A. H. & Engin O., 2010, Solving the fuzzy project scheduling problems with meta-heuristic methods, *Sigma Journal of Engineering and Natural Sciences*, 28, 86-101.
- Kumar, S., Bagchi, T. P. & Sriskandarajah, C., 2000, Lot streaming and scheduling heuristics for m-machine no-wait flowshops, *Computers & Industrial Engineering*, 38, 149-172.
- Külahlı, S., 2016, Esnek atölye tipi çizelgeleme problemleri için bir dağınık arama metodu, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.
- Labıdı, M., Koolı, A., Ladharı, T., Gharbı, A. & Suryahatmaja, U. S., 2018, A computational study of the two-machine no-wait flow shop scheduling problem subject to unequal release dates and non-availability constraints, *IEEE Access*, 6, 16294-16304.
- Laha, D. & Gupta, J. N., 2016, A Hungarian penalty-based construction algorithm to minimize makespan and total flow time in no-wait flow shops, *Computers & Industrial Engineering*, 98, 373-383.

- Laha, D., Gupta, J. N. & Sapkal, S. U., 2014, A penalty-shift-insertion-based algorithm to minimize total flow time in no-wait flow shops, *Journal of the Operational Research Society*, 65, 1611-1624.
- Laha, D. & Sapkal, S. U., 2014, An improved heuristic to minimize total flow time for scheduling in the m-machine no-wait flow shop, *Computers & Industrial Engineering*, 67, 36-43.
- Li, H., Li, X., & Gao, L., 2021, A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem, *Applied Soft Computing*, 100, 106946.
- Li, X., Yang, Z., Ruiz, R., Chen, T. & Sui, S., 2018, An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects, *Information Sciences*, 453, 408-425.
- Lin, S. W. & Ying, K. C., 2019, Makespan optimization in a no-wait flowline manufacturing cell with sequence-dependent family setup times, *Computers & Industrial Engineering*, 128, 1-7.
- Liu, G., Song, S. & Wu, C., 2013, Some heuristics for no-wait flow shops with total tardiness criterion, *Computers & Operations Research*, 40, 521-525.
- Lourenço, H., Marti, R. & Laguna, M., 2000, Assigning proctors to exams with scatter search. in computing tools for modeling, *Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J. L. González-Velarde, Eds. Kluwer Academic Publishers, Boston, MA, 215-227.
- Mansour, N., Isahakian, V. & Ghalayini, I., 2009, Scatter search technique for exam timetabling, *Applied Intelligence*, 34, 2, 299-310.
- Miyata, H. G., Nagano, M. S. & Gupta, J. N., 2019a, Some heuristics for no-wait flow shops with total tardiness criterion, *Engineering Optimization*, 51(4), 680-698.
- Miyata, H. H., Nagano, M. S. & Gupta, J. N., 2019b, Integrating preventive maintenance activities to the no-wait flow shop scheduling problem with dependent-sequence setup times and makespan minimization, *Computers & Industrial Engineering*, 135, 79-104.
- Moghaddam, R. T., Vahed, A. R. R. & Mirzaei, A. H., 2008, Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm, *Int J Adv Manuf Technol*, 36, 969-981.
- Moradinasab, N., Shafaei, R., Rabiee, M. & Ramezani, P., 2013, No-wait two stage hybrid flow shop scheduling with genetic and adaptive imperialist competitive algorithms, *Journal of Experimental & Theoretical Artificial Intelligence*, 25(2), 207-225.
- Muthuswamy, S., Gallego, M. C. V., Maya, J. & Santiago, M. R., 2012, Minimizing makespan in a two-machine no-wait flow shop with batch processing machines, *Int J Adv Manuf Technol*, 63, 281-290.

- Naderi, B. & Ruiz, R., 2014, A scatter search algorithm for the distributed permutation flowshop scheduling problem, *European Journal of Operational Research*, 239(2), 323-334.
- Nagano, M. S. & Miyata, H. H., 2016a, Review and classification of constructive heuristics mechanisms for no-wait flow shop problem, *Int J Adv Manuf Technol*, 86, 2161-2174.
- Nagano, M. S. & Miyata, H. H., 2016b, A high-quality solution constructive heuristic for no-wait flow shop scheduling problem, *Industrial Engineering & Management Systems*, 15(3), 206-214.
- Nagano, M. S., Miyata, H. H. & Araújo, D. C., 2015, A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times, *Journal of Manufacturing Systems*, 36, 224-230.
- Nagano, M. S., Silva, A. A. & Lorena, L. A. N., 2012, A new evolutionary clustering search for a no-wait flow shop problem with set-up times, *Engineering Applications of Artificial Intelligence*, 25, 1114-1120.
- Nagano, M. S., Silva, A. A. & Lorena, L. A. N., 2014, An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times, *Expert Systems with Applications*, 41, 3628-3633.
- Nasiri, M. & Kianfar, F., 2011, A hybrid scatter search for the partial job shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, 52, 9-12, 1031-1038.
- Nowicki, E. & Smutnicki, C., 2006, Some aspects of scatter search in the flow-shop problem, *European Journal of Operational Research*, 169(2), 654-666.
- Oktaý, S. & Engin, O., 2006, Scatter search method for solving industrial problems: literature survey, *Sigma Journal of Engineering and Natural Sciences*, 3, 144-155.
- Pendharkar, P. C., 2013, Scatter search based interactive multi-criteria optimization of fuzzy objectives for coal production planning, *Engineering Applications of Artificial Intelligence*, 26, 1503-1511.
- Qi, X., Wang, H., Zhu, H., Zhang, J., Chen, F. & Yang, J., 2016, Fast local neighborhood search algorithm for the no-wait flow shop scheduling with total flow time minimization, *International Journal of Production Research*, 54, 16, 4957-4972.
- Qin, Y. & Zhang, Y., 2016, Elite particle swarm optimization algorithm for solving the bi-criteria no-wait flexible flow shop problem, *International Journal of Grid and Distributed Computing*, 9(12), 267-282.
- Rabiee, M., Jolai, F., Asefi, H., Fattahi, P. & Lim, S., 2016, A biogeography-based optimization algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimize mean tardiness, *International Journal of Computer Integrated Manufacturing*, 29(9), 1007-1024.

- Rabiee, M., Rad, R. S., Mazinani, M. & Shafaei, R., 2014, An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines, *Int J Adv Manuf Technol*, 71, 1229-1245.
- Rahimi-Vahed, A. R., Javadi, B., Rabbani, M. & Tavakkoli-Moghaddam, R., 2008, A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem, *Engineering Optimization*, 40(4), 331-346.
- Ramezani, P., Rabiee, M. & Jolai, F., 2015, No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: a hybrid meta-heuristic approach, *J Intell Manuf*, 26, 731-744.
- Rao, A. R. M. & Lakshmi, K., 2008, Multi-objective scatter search algorithm for combinatorial optimization, *Advanced Computing and Communications 16th International Conference on, ADCOM 2008*, 303-308.
- Ren, T., Zhang, C., Lin, L., Guo, M. & Xie, X., 2014, Asymptotic analysis of SPTA-based algorithms for no-wait flow shop scheduling problem with release dates, *Hindawi Publishing Corporation the Scientific World Journal*, 2014, 7 pages.
- Riahi, V. & Kazemi, M., 2018, A new hybrid ant colony algorithm for scheduling of no-wait flowshop, *Oper Res Int J*, 18, 55-74.
- Riahi, V., Khorramizadeh, M., Newton, M. H. & Sattar, A., 2017, Scatter search for mixed blocking flowshop scheduling, *Expert Systems with Applications*, 79, 20-32.
- Sadiq, A.T. & Mohammed, K. S., 2012, Improved scatter search for job shop scheduling problem, *International Journal of Research and Reviews in Soft and Intelligent Computing*, 2, 1, 104-107.
- Sagarna, R. & Lozano, J.A., 2006, Scatter Search in software testing, comparison and collaboration with Estimation of Distribution Algorithms, *European Journal of Operational Research*, 169(2), 392-412.
- Samarghandi, H., 2015a, A no-wait flow shop system with sequence dependent setup times and server constraints, *IFAC-PapersOnLine*, 48(3), 1604-1609.
- Samarghandi, H., 2015b, Studying the effect of server side-constraints on the makespan of the no-wait flow shop problem with sequence dependent setup times, *International Journal of Production Research*, 53(9), 2652-2673.
- Samarghandi, H., 2015c, A particle swarm optimization for the no-wait flow shop problem with due date constraints, *International Journal of Production Research*, 53(9), 2853-2870.
- Samarghandi, H. & Behroozi, M., 2016, An enumeration algorithm for the no-wait flow shop problem with due date constraints, *IFAC*, 49(12), 1803-1808.

- Samarghandi, H. & Behroozi, M., 2017, On the exact solution of the no-wait flow shop problem with due date constraints, *Computers and Operations Research*, 81, 141-159.
- Samarghandi, H. & ElMekkawy, T. Y., 2012a, A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion, *Int J Adv Manuf Technol*, 61, 1101-1114.
- Samarghandi, H. & ElMekkawy, T. Y., 2012b, A meta-heuristic approach for solving the no-wait flow-shop problem, *International Journal of Production Research*, 50(24), 7313-7326.
- Samarghandi, H. & ElMekkawy, T. Y., 2014, Solving the no-wait flow-shop problem with sequence-dependent set-up times, *International Journal of Computer Integrated Manufacturing*, 27(3), 213-228.
- Sapkal, S.U. & Laha, D., 2013, A heuristic for no-wait flow shop scheduling, *Int J Adv Manuf Technol*, 68, 1327-1338.
- Saravanan, M., Haq, A. N., Vivekraj, A. R. & Prasad, T., 2008, Performance evaluation of the scatter search method for permutation flowshop sequencing problems, *The International Journal of Advanced Manufacturing Technology*, 37(11-12), 1200-1208.
- Schaller, J. & Valente, J. M., 2020, Minimizing total earliness and tardiness in a no wait flow shop, *International Journal of Production Economics*, 224, 107542.
- Shabtay, D., Arviv, K., Stern, H. & Edan, Y., 2014, A combined robot selection and scheduling problem for flow-shops with no-wait restrictions, *Omega*, 43, 96-107.
- Shafaei, R., Rabiee, M. & Mirzaeyan, M., 2011, An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop, *International Journal of Computer Integrated Manufacturing*, 24(10), 888-899.
- Shao, W., Pi, D. & Shao, Z., 2017a, An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem, *Applied Soft Computing*, 61, 193-210.
- Shao, W., Pi, D., Shao, Z., 2017b, Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms, *Knowledge-Based Systems*, 137, 163-181.
- Shao, W., Pi, D. & Shao, Z., 2019, A pareto-based estimation of distribution algorithm for solving multi objective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time, *IEEE Transactions on Automation Science and Engineering*, 16(3), 1344-1359.
- Shyu S. J., Lin B. M. T. & Yin P. Y., 2004, Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time, *Computers & Industrial Engineering*, 47(2-3), 181-193.

- Sonuç, S.B., 2008, İki boyutlu HP-model' inde protein katlanma problemine dağınık arama yaklaşımı, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Koç Üniversitesi.
- Strusevich, V. A., 1995, Two machine flow shop scheduling problem with no wait in process: Controllable machine speeds, *Discrete Applied Mathematics*, 59, 75-86.
- Sun, X., Geng, X. N., Wang, J. B. & Liu, F., 2019, Convex resource allocation scheduling in the no-wait flowshop with common flow allowance and learning effect, *International Journal of Production Research*, 57(6), 1873-1891.
- Tan, Y., Zhou, M., Zhang, Y., Guo, X., Qi, L. & Wang, Y., 2020, Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting, *IEEE Transactions on Automation Science and Engineering*, 17(4), 1814-1828.
- Tang, J., Zhang, J. & Pan, Z., 2010, A scatter search algorithm for solving vehicle routing problem with loading cost. *Expert Systems with Applications*, 37(6), 4073-4083.
- Taşgetiren, M. F., Yüksel, D., Gao, L., Pan, Q. K. & Li, P., 2019, A discrete artificial bee colony algorithm for the energy-efficient no-wait flowshop scheduling problem, *Procedia Manufacturing*, 39, 1223-1231.
- Urgo, M., 2019, A branch-and-bound approach to schedule a no-wait flow shop to minimize the CVaR of the residual work content, *Computers & Industrial Engineering*, 129, 67-75.
- Wang, S., Liu, M., 2013, A genetic algorithm for two-stage no-wait hybrid flow shop Scheduling problem, *Computers & Operations Research*, 40, 1064-1075.
- Wang, S., Liu, M., Chu, C., 2015, A genetic algorithm for two-stage no-wait hybrid flow shop Scheduling problem, *International Journal of Production Research*, 53(4), 1143-1167.
- Yang, Y., Li, P., Wang, S., Liu, B. & Luo, Y., 2017, Scatter search for distributed assembly flowshop scheduling to minimize total tardiness, *2017 IEEE Congress on evolutionary computation (CEC)*, 861-868.
- Ye, H., Li, W. & Abedini, A., 2017a, An improved heuristic for no-wait flow shop to minimize makespan, *Journal of Manufacturing Systems*, 44, 273-279.
- Ye, H., Li, W., Abedini, A. & Nault, B., 2017b, An effective and efficient heuristic for no-wait flow shop production to minimize total completion time, *Computers & Industrial Engineering*, 108, 57-69.
- Ye, H., Li, W. & Miao, E., 2016, An effective heuristic for no-wait flow shop production to minimize makespan, *Journal of Manufacturing Systems*, 40, 2-7.
- Ye, H., Li, W. & Nault, B. R., 2019, Trade-off balancing between maximum and total completion times for no-wait flow shop production, *International Journal of Production Research*, 3235-3251.

- Ying, K. C., Lin, S. W. & Wu, W. J., 2016, Self-adaptive ruin-and-recreate algorithm for minimizing total flow time in no-wait flowshops, *Computers & Industrial Engineering*, 101, 167-176.
- Yüksel, D., Taşgetiren, M. F., Kandiller, L. & Gao, L. 2020, An energy-efficient bi-objective no-wait permutation flowshop scheduling problem to minimize total tardiness and total energy consumption. *Computers & Industrial Engineering*, 145, 106431.
- Zhao, C., & Tang, H., 2011, A note on two-machine no-wait flow shop scheduling with deteriorating jobs and machine availability constraints, *Optimization Letters*, 5(1), 183-190.
- Zhao, F., Liu, H., Zhang, Y., Ma, W. & Zhang, C., 2018, A discrete water wave optimization algorithm for no-wait flow shop scheduling problem, *Expert Systems with Applications*, 91, 347-363.
- Zhao, F., Qin, S., Yang, G., Ma, W., Zhang, C. & Song, H., 2019b, A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective, *Expert Systems with Applications*, 126, 41-53.
- Zhao, F., Qin, S., Zhang, Y., Ma, W., Zhang, C. & Song, H., 2019c, A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem, *Expert Systems with Applications*, 126, 321-339.
- Zhao, F., Xue, F., Yang, G., Ma, W., Zhang, C. & Song, H., 2019a, A fitness landscape analysis for the no-wait flow shop scheduling problem with factorial representation, *IEEE Access*, 7, 21032-21047.
- Zhao, F., Zhang, L., Liu, H., Zhang, Y., Ma, W., Zhang, C. & Song, H., 2019d, An improved water wave optimization algorithm with the single wave mechanism for the no-wait flow-shop scheduling problem, *Engineering Optimization*, 51(10), 1727-1742.
- Zhou, Y. & Gu, X., 2009, Research on no-wait flow shop scheduling problem with fuzzy due date based on evolution games, *2019-2nd IEEE International Conference on Computer Science and Information Technology*, 495-499.
- Zhu, H., Luo, N. & Li, X., 2021, A quantum-inspired cuckoo co-evolutionary algorithm for no-wait flow shop scheduling, *IET Collaborative Intelligent Manufacturing*
- Zhu, H., Qi, X., Chen, F., He, X., Chen, L. & Zhang, Z., 2019, Quantum-inspired cuckoo co-search algorithm for no-wait flow shop scheduling, *Applied Intelligence*, 49, 791-803.