



Beklemez Akış Tipi Çizelgeleme Problemlerinin Analizi ve Hibrit Dağınık Arama Yöntemi ile Çözümü

Ramazan Başar^{1*}, Orhan Engin²

¹Endüstri Mühendisliği Bölümü, Mühendislik Fakültesi, Aksaray Üniversitesi, Aksaray, Türkiye

²Endüstri Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, Konya Teknik Üniversitesi, Konya, Türkiye

Makale Tarihiçesi

Gönderim: 11.05.2021
Kabul: 13.12.2021
Yayın: 10.06.2022

Araştırma Makalesi

Öz – Beklemez Akış Tipi Çizelgeleme (BATÇ), pratik uygulamalarından dolayı kapsamlı bir araştırma alanıdır. BATÇ problemlerinde işler, makinelerde kesintisiz olarak işlem görmek zorundadır. Bir işin tüm makinelerde işleme süresi boyunca, makineler bekleyebilir fakat işler kesintisiz olarak işlenmelidir. Amaç ise makinelerin boşta bekleme süresini en aza indirmektir. BATÇ problemlerinin çoğunluğunda toplam gecikmenin ve maksimum tamamlanma zamanının minimizasyonu olmak üzere, iki performans ölçüsü göz önünde bulundurulur. Literatürde, son yirmi beş yılda BATÇ ile ilgili yapılan çalışmalar analiz edilmiştir. BATÇ problemlerinin çözümü ile ilgili geliştirilen kesin ve yaklaşık çözüm veren yöntemler incelenmiştir. Literatürde 1 ve 2 makineli problemler için optimum çözüm veren matematiksel yöntemler bulunurken, 3 ve daha fazla makineli problemler için standart zamanda optimum çözüm veren bir yöntem bulunmamaktadır. Kabul edilebilir bir süre içerisinde m makine içeren problemlere optimum ya da optimuma yakın çözümler üretebilmek için sezgisel ve meta sezgisel yöntemler geliştirilmektedir. Bu çalışmada, BATÇ problemlerinin çözümü için Hibrit Dağınık Arama (HDA) yöntemi önerilmiştir. Önerilen yöntem, literatürde iyi bilinen kıyaslama problemleri yardımı ile test edilmiştir. Elde edilen sonuçlar, Hibrit Uyarlanabilir Öğrenme Yaklaşım (HUÖY) algoritması ve Hibrit Karınca Kolonileri Optimizasyon (HKKO) algoritması ile kıyaslanmıştır. Amaç fonksiyonu olarak maksimum tamamlanma zamanının minimizasyonu seçilmiştir. Elde edilen çözüm sonuçları, önerilen HDA yönteminin BATÇ problemlerinin çözümünde etkili olduğunu göstermiştir.

Anahtar Kelimeler – Akış tipi, beklemez çizelgeleme, hibrit dağınık arama, tamamlanma zamanı

Analysis of No-wait Flow Shop Scheduling Problems and Solving with Hybrid Scatter Search Method

¹Industrial Engineering Department, Engineering Faculty, Aksaray University, Aksaray, Türkiye

²Industrial Engineering Department, Engineering and Natural Science Faculty, Konya Technical University, Konya, Türkiye

Article History

Received: 11.05.2021
Accepted: 13.12.2021
Published: 10.06.2022

Research Article

Abstract – No-wait flow shop (NWFS) is extensively research area due to its practical applications. In NWFS, jobs are processed in machines without interruption. During the schedule period, machines can wait, but jobs cannot wait. The aim is to minimize the idle time for machines. The majority of NWFS, two performance measures are considered: minimization of total delay and minimization of the makespan. The researches on the NWFS in the last twenty-five years have been analysed from the literature. The methods developed for the solution of the NWFS, which give exact and approximate solutions, have been examined. While there are mathematical methods that give optimum solutions for 1 and 2 machine problems in the literature, there is no method that provides optimum solutions in standard time for problems with 3 or more machines. The difference methods are developed in order to produce optimum or near-optimum solutions to m-machine problems in an acceptable time. A Hybrid Scatter Search Method (HSSM) is proposed for solving the NWFS. The developed HSSM tested with the well-known benchmarking instances in the literature. The results obtained were compared with the Hybrid Adaptive Learning Approach algorithm and the Hybrid Ant Colonies Optimization algorithm. The objective function is makespan minimization. According to solutions, the proposed HSSM is an effective metaheuristic to solve NWFS.

Keywords – Flow shop, no-wait scheduling, hybrid scatter search, makespan

¹ ramazanbasar@aksaray.edu.tr

² orhanengin@yahoo.com

*Sorumlu Yazar / Corresponding Author

1. Giriş

Beklemesiz akış tipi ortamında, sürecin başlangıcından sonuna kadar sistemin herhangi bir aşamasında, işlerin beklemesine izin verilmeksizin kesintisiz olarak yapılması gerekir. Uçuş, tren ve ameliyat çizelgeleme problemleri bunun birkaç örneğidir. Akış tipi ortamında yapılması gereken işler aynı görev listesine sahip ise, işler makinelerde aynı sırada işlem görür. Akış Tipi Çizelgeleme (ATÇ) problemlerinin önemli bir kısmı, beklemesiz kısıtı ile sınıflandırılır. BATÇ problemlerinin çözümü ile ilgili yapılan çalışmalarının çoğunda iki performans ölçüsü göz önünde bulundurulmaktadır. Bunlar, toplam gecikme ve tamamlanma zamanının en küçüklenmesidir. Müşteri isteklerinin yerine getirilmesinin öncelikli olduğu ortamlarda, toplam gecikmenin performans kriteri olarak kullanılması gereklidir. İş bitim tarihleri ile tamamlanma süreleri arasındaki boşluklar artıka toplam gecikme artar.

BATÇ problemlerinde, işler ve 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 tane makine içeren kümeyi,

$j=(1,2,\dots,n)$: n tane makinelerde işlenmesi gereken işin oluşturduğu kümeyi temsilen;

p_{ij} : j . işin i . makinedeki işlem süresini ifade eder.

r_j : j . işin sisteme geliş zamanını ifade eder.

d_j : j . işin müşteriye teslim edileceği tarihi ifade eder.

w_j : j . işin diğer işlere göre ağırlığının yani önceliğini belirten ifadedir.

C_j : j . işin tamamlanma zamanını yani teslim hazır olduğu zamanı ifade eder.

F_j : $C_j - r_j$: j . işin tamamlanma zamanından j . işin hazır olma zamanının çıkarılmasıyla j . iş için akış zamanını yani o işin sistemde geçirdiği zamanı verir.

L_j : $C_j - d_j$: j . işin tamamlanma zamanından j . işin teslim zamanının çıkarılmasıyla j . iş için gecikme zamanı elde edilir.

T_j : $\max(C_j - d_j, 0)$: Tüm işler içerisindeki gecikme zamanı en fazla olanı işin gecikme süresini ifade eder.

BATÇ problemini ifade etmek için $\alpha / \beta / \gamma$ şeklindeki bir gösterim kullanılmaktadır. Bu gösterimde α makinelerin bulunduğu akış tipi ortamını ifade eder. β işlem özellikleri ve beklemesiz olma durumunu ifade eder. γ ile gösterilen alanda ise performans ölçütü yani amaç fonksiyon bilgisini ifade eder. BATÇ problemleri, $F_m / no-wait / C_{max}$ notasyonları ile ifade edilir. Bu notasyonlar aşağıda açıklanmıştır.

F_m : Akış Tipi, işlerin tamamının aynı sırayı takip ettiği ve m adet makineden aynı sırayla geçtiği durumu tanımlar.

$no-wait$: Beklemesiz, işlerin beklemesine izin verilmeksizin sürekli olarak işlem görmek zorunda olduğu durumu tanımlar.

C_{max} : Amaç fonksiyonu, yani performans ölçütümüz bu bölümde ifade edilir.

BATÇ problemlerinin yaygın olarak kullanılan performans ölçüm kriterleri şu şekildedir;

C_{max} : Maksimum tamamlanma zamanı

L_{max} : Maksimum gecikme

TT : Toplam gecikme

Engin ve Güçlü (2018) tamamlanma zamanı kriteri ile BATÇ probleminin matematiksel modelini, denklem 1.1, 1.2 ve 1.3'te tanımlamışlardır:

İşlem süresi: $i(j, k)$ olmak üzere; k , makineleri ve j işleri temsil etmektedir. $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ iş dizilimini temsilen minimum gecikme denklem 1.1'deki gibi olur;

$$mg(\pi_{j-1}, \pi_j) = i(\pi_{j-1}, 1) + \max \left[0, \max_{2 \leq k \leq m} \left\{ \sum_{k=2}^m i(\pi_{j-1}, k) - \sum_{k=1}^{m-1} i(\pi_j, k) \right\} \right] \quad (1.1)$$

Maksimum tamamlanma (MT) zamanı, π dizilim kümesindeki n iş ve m makine için tamamlanma zamanı (M) ile kıyaslayarak denklem 1.2 ve 1.3'te tanımlanmıştır.

$$C_{max}(\pi) = MT(\pi) = \sum_{j=2}^n mg(\pi_{j-1}\pi_j) + \sum_{k=1}^m i(\pi_n, k) \quad (1.2)$$

$$MT(\pi^*) \leq M(\pi_n, m) \quad \forall \pi \in \pi \quad (1.3)$$

Tamamlanma zamanı kriteri ile BATÇ problemi, π dizilim kümesinin tümünde bir π^* diziliminin bulunması şeklinde açıklanmıştır.

Bu çalışmada, BATÇ problemleri konusunda son yirmi beş yılda yapılan çalışmalar detaylı olarak incelenmiştir. BATÇ problemlerinin çözümü için geliştirilen yöntemler incelenmiştir. BATÇ problemlerinin çözümü için yeni bir hibrit dağınık arama metodu önerilmiş ve literatürde yer alan kıyaslama problemi üzerinde bu yöntem test edilmiştir.

Çalışmanın ikinci bölümünde, BATÇ problemleri ile ilgili literatürde yapılan çalışmalar açıklanmıştır. Üçüncü bölümde, önerilen dağınık arama yöntemi ve prosedürleri verilmiştir. Dördüncü bölümde, HDA yöntemi ile BATÇ probleminin çözümü sunulmuştur. Beşinci bölümde ise BATÇ problemi için önerdiğimiz HDA algoritmasının sonuçları ve değerlendirmesi yapılmıştır.

1.1. Literatür Araştırması

BATÇ problemleri konusunda son yirmi beş yıldaki araştırmalar aşağıda verilmiştir.

Strusevich (1995) işlem sırası beklemez, iki makine ATÇ problemini, kontrol edilebilir makine hızları ile ele almıştır. Amaç olarak, makine hızlarının en uygun değerlerini bulma ve en uygun zaman çizelgesini belirleme olarak tanımlamıştır. Espinouse, Formanowicz ve Penz (1999), sınırlı sayıda makine kullanılabilirliği bulunan, iki makineli BATÇ problemini incelemiştir. Amaç fonksiyonunu, maksimum tamamlanma zamanı (C_{max}) olarak belirlemişlerdir. Kumar, Bagchi ve Sriskandarajah (2000) m-makineli BATÇ için parti akışı ve çizelgeleme sezgisellerini araştırmışlardır. Çalışmanın amacını, çok sayıda akış gerektiren birden fazla ürün ile m-makineli BATÇ' de tamamlanma zamanının en aza indirilmesi olarak belirlemişlerdir. Moghaddam, Vahed ve Mirzaei (2008) ağırlıklı ortalama tamamlanma süresini ve eş zamanlı ağırlıklı ortalama gecikmeyi en aza indirmek için çok amaçlı BATÇ problemini ele almışlardır. Verilen problem için yerel olarak Pareto optimal sınırını aramak amacıyla bağımsızlık algoritmasını önermişlerdir. Chihaoui, Kacem, Hadj-Alouane, Dridi ve Rezg (2011) müsait olmama kısıtlaması ve farklı serbest bırakma tarihleri altındaki tamamlanma zamanını en aza indirmek için iki makineli BATÇ problemini incelemişlerdir. Çözüm için dal-sınır algoritması kullanmışlardır. Engin ve Günaydın (2011) BATÇ problemini, tamamlanma zamanını en aza indirmeye amacı ile ele almışlardır. Problemi çözmek için yapay sinir ağlarının öğrenme yapısına benzeyen iyileştirilmiş uyarlanabilir bir yaklaşımını kullanmışlardır. Elde ettikleri sonuçları, genetik algoritma ile kıyaslayarak, iyileştirilmiş uyarlanabilir öğrenme yaklaşımı algoritmasının, BATÇ problemlerinde daha etkin olduğu vurgulanmıştır. Aydilek ve Allahverdi (2012) m-makineli BATÇ problemine yönelik iki performans ölçütünü göz önünde bulundurmışlardır. Amacı, tamamlanma zamanının belirli bir değerden daha büyük olmayacak şekilde en aza indirgen bir dizi bulunması olarak belirlemişlerdir. Czogalla ve Fink (2012) BATÇ probleminin uyum yüzeyi, yüzey sağlamlığı ve bir çözümün kalitesi ile optimum çözüme olan uzaklık arasındaki korelasyonu incelemişlerdir. Uyum yüzeyinin evrimsel hesaplama ve yerel arama yöntemleriyle aranmaya uygunluğunu araştırmışlardır. Jolai, Rabiee ve Asefi (2012) tarafından sıra bağımlı kurulum süreleriyle, beklemez esnek ATÇ için hibrit meta-sezgisel algoritma geliştirmişlerdir. Amacı, maksimum tamamlanma süresinin en aza indirilmesi olarak belirlemişlerdir. Uyarlanmış yayılımcı rekabet algoritması ve popülasyona dayalı benzetilmiş tavlama birleşimi olan hibrit bir algoritma tasarlamışlardır. Samarghandi ve ElMekkawy (2012) tarafından yapılan çalışmada, ayrılabilir kurulum süreleri ve tamamlanma zamanı kriterleri ile BATÇ problemini çözmek için

genetik algoritma ve parçacık sürü optimizasyonunu kullanmışlardır. Çalışmalarında, her operasyonun ayrı bir kurulum süresi olduğunu ve bir işlemin kurulum süresinin önceki işlemlerden bağımsız olduğunu vurgulamışlardır. [Moradinassab, Shafaei, Rabiee ve Ramezani \(2013\)](#), toplam tamamlanma süresini en aza indirmeyi amaçlayan, kurulum süreleriyle, bekleme iki aşamalı, esnek ATÇ problemini incelemişlerdir. Problemlerini, uyarlanabilir bir yayılımcı rekabet algoritması ve genetik algoritma kullanarak çözmüşlerdir. [Wang ve Liu \(2013\)](#) çalışmalarında, iki aşamalı bekleme hibrit ATÇ için genetik yöntem önermişlerdir. Problemin çözümünde, iki noktalı doğrusal sıra geçişi operatörü ve takas mutasyon operatörü kullanıldığında genetik algoritmanın nispeten daha iyi performans gösterdiğini vurgulamışlardır. [Akhshabi, Moghaddam ve Roodposhti \(2014\)](#), BATÇ problemini çözmek için yerel arama yöntemiyle memetik algoritmanın birleşimine dayanan parçacık sürü optimizasyon algoritmasını önermişlerdir. Amacı, toplam akış süresinin en aza indirilmesi olarak belirlemişlerdir. Ayrıca, önerilen algoritmanın, kendi kendine organize olan rastsal göçmen şeması ile genişletildiğini belirtmişlerdir. Yapılan deneysel çalışmalar ile parçacık sürü optimizasyon tabanlı memetik algoritmanın etkili bir yöntem olduğu ifade etmişlerdir. [Allahverdi ve Aydılek \(2014\)](#) m -makine BATÇ problemini, tamamlama zamanının belirli bir değerden daha fazla olmadığı kısıtlamalarına tabi tutulan toplam tamamlanma süresini en aza indirmek amacıyla incelemişlerdir. Çözüm için yerleştirme algoritması, genetik algoritma, tavlama benzetim algoritması, bulut teorisi tabanlı tavlama benzetim algoritması ve diferansiyel evrim algoritması önermişlerdir. [Asefi, Jolai, Rabiee ve Araghi \(2014\)](#) iki amaçlı, bekleme esnek ATÇ problemi için baskın olmayan sınıflandırma genetik algoritması ve değişken komşu arama yöntemini birleştirerek kullanmışlardır. Bekleme k aşamalı esnek ATÇ problemini, her aşamada aynı m adet makinelerin bulunduğu durumu incelemişlerdir. Mevcut n adet işlerinin planlaması ve böylece n adet işlerinin tamamlanma zamanı ve ortalama gecikmesinin en aza indirgenmesini amaçlamışlardır. [Shabtay, Arviv, Stern, ve Edan \(2014\)](#), ardışık makine çiftleri arasında birden fazla robot aktarma halkası işinde, iki kriterli BATÇ problemini incelemişlerdir. Araştırmada, performans ölçütleri olarak, tamamlanma zamanını ve robot seçim maliyetini almışlardır. İki kriterli en kısa yol problemi kullanarak en uygun çözüme ulaşmaya çalışmışlardır. [Deng, Wei, Su, ve Zhao \(2015\)](#) NP (Non-Deterministic Polynomial Time: deterministik olmayan polinomial zaman problemleri)-zor kombinatoriyel optimizasyon problemi olan BATÇ problemi için rekabetçi bir eş-evrimsel kuantum genetik algoritma önermişlerdir. [Samarghandi \(2015a\)](#) bekleme akış tipi sisteminin, sıraya bağlı kurulum süreleri ve sunucu yan kısıtlamaları ile çizelgelenmesi problemini ele almıştır. Problemin çözümünde, genetik algoritmadan yararlanmıştır. [Samarghandi \(2015b\)](#) bir diğer çalışmada, sıraya bağlı kurulum süreleri ve sunucu kısıtlamaları olan BATÇ problemini analiz etmiştir. Problemi çözmek için iki farklı genetik algoritma geliştirmiştir. [Wang, Liu ve Chu \(2015\)](#) iki aşamalı bekleme esnek ATÇ için genetik algoritma geliştirmişlerdir. İki noktalı doğrusal sıra çaprazlama operatörü ve takas mutasyon operatörü kullanıldığında genetik algoritmanın nispeten daha iyi performans gösterdiğini vurgulamışlardır. [Laha ve Gupta \(2016\)](#) BATÇ'de tamamlanma zamanı ve toplam akış süresini minimize etmek için Macar cezasına dayanan yapısal bir algoritma sunmuşlardır. Başlangıçta klasik atama problemi için kullanılan Macar ceza metodundan türetilen yöntemi, en uygun programı elde etmek için bir yerleştirme tekniği ile daha da geliştirilmiş olan ilk iş dizisini oluşturmak için kullanmışlardır. [Qi vd. \(2016\)](#) ikiden fazla makine için NP-zor olduğu bilinen BATÇ problemindeki toplam akış süresini en aza indirmek için hızlı yerel komşu arama algoritması önermişlerdir. [Bewoor, Prakash ve Sapkal \(2017\)](#) BATÇ problemlerini çözmek için evrimsel hibrit parçacık sürü optimizasyon algoritması önermişlerdir. Önerilen bu yöntemi, işlerin toplam akış süresini en aza indirmek amacıyla kullanmışlardır. [Samarghandi ve Behroozi \(2017\)](#) karmaşık tam sayılı programlama, kısıtlayıcı programlama, numaralandırma algoritması kullanılarak teslim tarihi kısıtlaması ile BATÇ problemine optimum çözüm bulmak için çalışmışlardır. [Shao, Pi ve Shao \(2017\)](#) BATÇ problemini çözmek için genişletilmiş öğretim-öğrenme tabanlı optimizasyon algoritması önermişlerdir. Geliştirdikleri algoritmanın, ön izleme, öğretim, öğrenme ve gözden geçirme olarak dört aşamadan oluştuğu belirtmişlerdir. [Chaudhry, Elbadawi, Usman ve Chugtai \(2018\)](#) genetik algoritmalar kullanarak BATÇ probleminde toplam akış süresinin azaltılması üzerine çalışma yapmışlardır. [Engin ve Güçlü \(2018\)](#) maksimum tamamlama süresini en aza indirme kriteri ile BATÇ için çaprazlama ve mutasyon mekanizmasına dayanan hibrit karınca kolonisi algoritmasını önermişlerdir. Çalışmanın amaç fonksiyonunu, tamamlanma zamanının minimizasyonu olarak belirlemişlerdir. Çalışmalarında, elde ettikleri sonuçları, genetik algoritma ve uyarlanabilir öğrenme yaklaşım algoritması ile kıyaslamışlardır. Önerdikleri hibrit karınca koloni

algoritmasının, BATÇ problemleri için daha etkin sonuçlar verdiğini vurgulanmışlardır. Gao, Liu, Wang ve Lu (2018) öğrenme etkisi, ortak bitiş tarihi ve kontrol edilebilir işleme süreleri ile beklenmeyen iki makineli permütasyon ATÇ problemini çalışmışlardır. Ortak bitiş tarihi atama yöntemi kullanılarak problemin, önerilen model kapsamında polinomiyel olarak çözülebilir olduğunu göstermişlerdir. Labıdı, Koolı, Ladharı, Gharbı ve Suryahatmaja (2018) iki makineli BATÇ probleminin, eşit olmayan teslim tarihleri ve uygun olmama kısıtlamalarını hesaplamak için bir çalışma yapmışlardır. Problem için bir matematiksel formülasyon önermişlerdir. Geliştirdikleri modelde, dal-sınır algoritmasına yeni eklemeler yaparak standart dal-sınır algoritmalarından daha iyi performans elde ettiklerini vurgulamışlardır. Bewoor, Chandraprakash ve Sapkal (2019) BATÇ problemlerinde, tamamlanma zamanını minimize etmek için evrimsel hibrit kuş sürüsü optimizasyon algoritmasını önermişlerdir. Algoritmada evrimsel arama, NEH ve tavlama benzetimi algoritmasını kullanmışlardır. Cheng, Ying, Li ve Hsieh (2019) sıraya bağlı kurulum süreleriyle karışık BATÇ problemini, tamamlanma zamanının en aza indirilmesi amacı ile ele almışlardır. Tamamlanma zamanı kriterine göre küçük problemleri çözmek için karma tamsayılı doğrusal programlama modelini sunulmuştur. Orta ve büyük boyutlu problemleri çözmek için ikili yinelenen ağgözlü bir meta-sezgisel önermişlerdir. Lin ve Ying (2019) sıraya bağlı kurulum süreleri ile beklenmeyen akış tipi üretim hücresinde tamamlanma zamanı optimizasyonunu incelemişlerdir. Sun, Geng, Wang ve Liu (2019) öğrenme etkisi ve dışbükey kaynağa bağlı işlem süreleri ile iki makineli BATÇ problemini incelemişlerdir. Amacı, kaynak tüketim maliyetini ve toplam akış süresinin minimize edilmesi olarak belirlemişlerdir. Tasgetiren, Yüksel, Gao, Pan ve Li (2019) beklenmeyen permütasyon ATÇ problemini incelemişlerdir. Çalışmanın amacını, toplam akış süresinin ve toplam enerji tüketiminin minimizasyonu olarak belirlemişlerdir. Bunun için iki amaçlı karışık tamsayı programlama modeli geliştirmişlerdir. Zhao vd. (2019) BATÇ probleminde tamamlanma zamanını minimize etmek için tek bir dalga mekanizmasına sahip, tek bir su dalgası optimizasyonu olarak adlandırılan su dalgası optimizasyon algoritmasını önermişlerdir. Zhao vd. (2019a) faktöriyel gösterim ile BATÇ problemi için uyum yüzeyi analizini yapmışlardır. Çalışmanın amacını tamamlanma zamanının minimizasyonu olarak belirlemişlerdir. Zhu vd. (2019) BATÇ problemini çözmek için kuantumdan esinlenilmiş, guguklu eş arama algoritmasını önermişlerdir. Performans kriterini, maksimum tamamlanma zamanının minimize edilmesi olarak belirtmişlerdir. Algoritmanın üç aşamadan oluştuğunu belirttikten bu aşamaların, kuantum temsili, guguk kuşu arama diferansiyel evrimi ve yerel komşu arama olduğunu ifade etmişlerdir. Yapılan hesaplamalar sonucunda, önerdikleri algoritmanın küçük ve orta boyutlu BATÇ problemlerinde etkin sonuçlar verdiğini vurgulamışlardır. Allahverdi, Aydılek ve Aydılek (2020) tamamlanma zamanına bağlı olarak toplam gecikmeyi en aza indirmek için ayrı kurulum sürelerine sahip, BATÇ problemini incelemişlerdir. Blok tavlama benzetimi adı verdikleri bir algoritma ile yinelenen arama algoritması birleştirilerek PA algoritması elde etmişlerdir. Chen, Li ve Wang (2020) beklenmeyen ve kullanılmayan periyod kısıtlamaları ile iki makineli ATÇ için tamamlanma zamanı minimize etmişlerdir. Problem çözümü için Gilmore ve Gomory Algoritması ile Johnson algoritmasını kullanmışlar ve algoritmaların etkinliği hakkında sayısal sonuçlar vermişlerdir. Ferretti ve Zavanella (2020), genel akış tipi problemi için beklenmeyen ve engelleme kısıtlamaları ile toplam enerji maliyetini en aza indiren bir çizelgeleme modeli önermişlerdir. Ara stokları olmayan iki makineli bir akış atölye sistemini dikkate almış ve çözüm için genetik algoritmayı kullanmışlardır. Pourhejazy, Lin, Cheng, Ying ve Lin (2020) BATÇ problemini, serbest bırakma zamanları ile optimize etmek için geliştirilmiş ışın arama algoritmasını tasarlamışlardır. Tamamlanma zamanını en aza indirmek için karışık tamsayı doğrusal programlama formülasyonunu önermişlerdir. Yerel arama tabanlı ışın arama algoritması geliştirmişler ve geliştirilen yöntemlerin etkinliği sayısal analizler ile değerlendirmişlerdir. Schaller ve Valente (2020) BATÇ ortamında toplam gecikmenin en aza indirilmesi için kesin bir dal-sınır algoritması geliştirmişlerdir. İki farklı sezgisel yöntemi probleme uyarlamışlar ve iki sezgisel arasında kıyaslama yapmışlardır. Problemin yapısına dayalı olarak, hızlandırma yöntemlerine sahip eklemeli arama iyileştirme prosedürünü önermişler ve makul bir sürede çok daha iyileştirilmiş çözümler bulunduğunu belirtmişlerdir. Zhao, He ve Wang (2020) BATÇ probleminin hem tamamlanma zamanı hem de toplam enerjiyi en aza indirme kriterleriyle enerji açısından verimli çizelgelemesini ele almak için TS-CEA adı verilen probleme özgü bilgiye sahip iki aşamalı bir işbirlikçi evrimsel algoritma önermişlerdir. Zhao vd. (2020a) büyük ölçekli BATÇ problemlerini, tamamlanma zamanının minimizasyonu kriteri ile çözmek için yapboz oyunundan esinlenen bir sezgisel algoritmayı kullanmışlardır. Büyük ölçekli karşılaştırma örneklerindeki deneysel sonuçların, algoritmanın

en güncel sezgisel yöntemlerden daha üstün olduğunu vurgulamışlardır. Allahverdi, Aydilek, Aydilek ve Allahverdi (2021) maksimum gecikme performans ölçütü ile iki makineli BATÇ problemi için daha iyi bir baskınlık ilişkisi olan sezgisel yöntemler geliştirmişlerdir. Önerdikleri sezgisel yöntemlerin etkinliğini kanıtlamak için hesaplamalı deneyler yapmışlardır. Başar ve Engin (2021) BATÇ probleminde, teslim tarihlerini bulanıklaştırarak dağınık arama yöntemi ile müşteri tatmin indekslerini hesaplamışlardır. Performans kriteri olarak maksimum tamamlanma zamanının minimizasyonu ve müşteri tatmin indeksinin maksimizasyonunu seçmişlerdir. Keskin ve Engin (2021) BATÇ problemini iki kriterle çözmek için hibrit bir genetik yerel ve küresel arama algoritması önermişlerdir. Çalışmada amaç fonksiyonu olarak maksimum tamamlanma zamanı ve toplam akış zamanı minimizasyonunu ele almışlardır. Koulamas ve Kyparisis (2021) BATÇ problemini, işlerin toplam ret edilme maliyeti ile ele almışlardır. Çalışmada, problemin çözümü için üçüncü dereceden polinom zamanlı bir dinamik programlama algoritması sunmuşlardır. Ayrıca çalışmalarında, işlerin toplam tamamlama süresi ve toplam reddetme maliyetini en aza indirmek için üçüncü dereceden bir polinom zamanlı dinamik programlama algoritması sunmuşlardır. Li, Li ve Gao (2021) heterojen dağıtılmış BATÇ probleminin çözümü için ayrık yapay arı koloni algoritmasını kullanmışlardır. İşlem süresini azaltmak için elde edilen komşuluğun değerlendirilmesini hızlandıracak yeni bir yöntem önermişler ve yapılan hesaplamalar sonucunda ayrık yapay arı koloni algoritmasının, en yüksek kalitede çözümler elde ettiğini vurgulamışlardır. Zhu, Luo ve Li (2021) BATÇ probleminin çözümü için kuantumdan esinlenilmiş guguk kuşunun eş evrimsel algoritmasını kullanmışlardır. Performans kriteri olarak tamamlanma zamanını minimize etmeye çalışmışlardır. Kuantum çözüm oluşturma, kuantum popülasyon evrimi ve gelişmiş bir yerel arama yöntemini önermişlerdir.

BATÇ problemleri ile ilgili literatürde, son yirmi beş yılda yapılan ve yukarıda verilen çalışmalar analiz edildiğinde, 1995-2000 yılları arasında üç ve 2001-2011 yılları arasında da dört çalışmanın yapıldığı görülmektedir. Oysa 2012-2021 yılları arasında elli üç çalışmanın olduğu belirlenmiştir. Bu çalışmaların on iki tanesi de son iki yılda yapılan araştırmalardır. BATÇ problemleri ile ilgili literatürde son yıllarda yoğun çalışmalar yapıldığı görülmektedir.

BATÇ probleminin çözümünde kullanılan kesin ve meta-sezgisel yöntemlere göre literatürde yapılan bazı çalışmalar ayrıca Tablo 1- 5'te verilmiştir.

Tablo 1

BATÇ problemlerinin çözümü için dal-sınır yöntemini kullanan çalışmalar

Yıl	Yazar	Çalışma
2011	<u>Chihaoui vd.</u>	Müsait olmama kısıtlaması ve farklı serbest bırakma tarihleri altındaki tamamlanma zamanını en aza indirmek için iki makineli BATÇ
2015	<u>Wang vd.</u>	İki aşamalı hibrit BATÇ için dal-sınır algoritması
2018	<u>Labıdı vd.</u>	İki makineli BATÇ probleminin, eşit olmayan teslim tarihleri ve uygun olmama kısıtlamalarını hesaplama çalışması
2020	<u>Schaller ve Valente</u>	Beklemez bir akış tipi ortamda toplam erkenliği ve gecikmeyi en aza indirme

Tablo 1'de görüldüğü gibi Dal-sınır yönteminin incelenen çalışmalar içerisinde kullanımının yaygın olmadığı görülmektedir. Bunun sebebi ise BATÇ problemlerinin makine sayısı 3 ve 3' ten fazla olduğu durumlarda NP-zor sınıfında yer almasıdır. Bundan dolayı dal-sınır yöntemi polinomiyele olarak çözülemeyen problemlerde, çözüm süresi çok fazla artış göstermekte ve optimum sonuç verememektedir.

Tablo 2

BATÇ problemlerinin çözümü için Genetik Algoritma kullanan çalışmalar

Yıl	Yazarlar	Çalışma
2000	<u>Kumar vd.</u>	<i>m</i> -makinelik beklemesiz akış istasyonları için parti akışı ve çizelgeleme sezgiselleri
2012	<u>Samarghandi ve ElMekkawy</u>	Ayrılabilir kurulum süreleri ve tamamlanma zamanı kriterleri ile beklemesiz ATÇ problemi için genetik algoritma ve parçacık sürüsü optimizasyonu
2013	<u>Moradinasab vd.</u>	Genetik ve uyarlanabilir yayılımcı rekabet algoritmaları ile beklemesiz iki aşamalı hibrit ATÇ
2013	<u>Wang ve Liu</u>	İki aşamalı beklemesiz hibrit ATÇ için genetik algoritma
2015	<u>Deng vd.</u>	Beklemez ATÇ için etkili eş-evrimsel kuantum genetik algoritması
2015b	<u>Samarghandi</u>	Sıraya bağlı kurulum süreleriyle beklemez akış tipi probleminin tamamlanma zamanı üzerindeki sunucu yan kısıtların etkisinin incelenmesi
2015a	<u>Samarghandi</u>	Sıraya bağlı kurulum süreleri ve sunucu kısıtlamaları olan beklemez bir akış tipi sistem
2018	<u>Chaudhry vd.</u>	Genetik algoritmalar kullanarak beklemez akış atölyesinde toplam akış süresinin azaltılması
2020	<u>Ferrettia ve Zavanella</u>	Genel ATÇ problemi için beklemez / engelleme kısıtlamaları ile toplu enerji planlama
2021	<u>Keskin ve Engin</u>	BATÇ problemini iki kriterle çözmek için hibrit bir genetik yerel ve küresel arama algoritması

Genetik algoritma, birçok probleme uyarlanabilir olması, çözüm süresini makul seviyelere düşürebilmesi ve optimuma oldukça yakın sonuç vermesinden dolayı kullanımı oldukça yaygın hale gelmiştir. Tablo 2’de görüldüğü üzere BATÇ problemlerinde de genetik algoritmaların kullanımının ağırlıklı olduğu açıktır.

Tablo 3

BATÇ problemlerinin çözümü için uyum yüzeyi analizi kullanan çalışmalar

Yıl	Yazar	Çalışma
2012	<u>Czogalla ve Fink</u>	BATÇ problemi için uyum yüzeyi analizi
2019a	<u>Zhao vd.</u>	Faktöriyel gösterim ile BATÇ problemi için uyum yüzeyi analizi

Tablo 3’te BATÇ problemlerinde uyum yüzeyi analizi kullanımına bakıldığı zaman oldukça az çalışma yapıldığı görülmektedir.

Tablo 4

BATÇ problemlerinin çözümü için karınca kolonisi algoritmasını kullanan çalışmalar

Yıl	Yazar	Çalışma
2018	<u>Engin ve Güçlü</u>	BATÇ problemleri için hibrit karınca kolonisi optimizasyon yöntemi

Tablo 4’de görüldüğü üzere Karınca kolonisi algoritmasının, NP-zor problemler için makul çözüm süresinde optimuma yakın sonuçlar verdiği bilinmektedir. 2018 yılında ise Engin ve Güçlü, çalışma yapmıştır. BATÇ problemlerinin çözümünde sezgisel yöntem kullanımı arttıkça, hibrit çözüm yöntemlerinin kullanımında da artış olmuştur. Tablo 5’te ise BATÇ problemleri için hibrit yöntem kullanılmış çalışmalara yer verilmiştir.

Tablo 5

BATÇ problemlerinin çözümü için hibrit yöntem kullanan çalışmalar

Yıl	Yazar	Çalışma
2012	<u>Jolai vd.</u>	Sıra bağımlı kurulum süreleriyle beklemesiz esnek bir ATÇ için hibrit meta-sezgisel yöntem
2013	<u>Wang ve MingLiu</u>	İki aşamalı beklemesiz hibrit ATÇ için genetik algoritma
2013	<u>Moradinasab vd.</u>	Genetik ve uyarlanabilir yayılımcı rekabet algoritmaları ile beklemesiz iki aşamalı hibrit ATÇ
2014	<u>Asefi vd.</u>	İki hedefli, beklemesiz esnek ATÇ problemini çözmek için hibrit NS-GA-II ve VNS
2014	<u>Akhshabi vd.</u>	Toplam akış süresi ile beklemesiz bir ATÇ için hibrit parçacık sürüsü optimizasyon algoritması
2015	<u>Wang vd.</u>	İki aşamalı beklemesiz hibrit ATÇ için dal-sınır yöntemi
2017	<u>Shao vd.</u>	Beklemesiz ATÇ problemi için genişletilmiş öğretim-öğrenme tabanlı optimizasyon algoritması
2017	<u>Bewoor vd.</u>	NP-zor BATÇ problemlerini çözmek için evrimsel hibrit parçacık sürü optimizasyonu algoritması
2018	<u>Engin ve Güçlü</u>	BATÇ problemlerini çözmek için hibrit karınca kolonisi optimizasyonu
2019	<u>Bewoor vd.</u>	Beklemesiz akış tipinde tamamlanma zamanını minimize etmek için evrimsel hibrit kuş sürüsü optimizasyon algoritması
2019	<u>Tasgetiren vd.</u>	Enerji açısından verimli BATÇ için ayrık yapay arı kolonisi
2020	<u>Allahverdi vd.</u>	Tamamlanma zamanına bağlı olarak toplam gecikmeyi en aza indirmek için ayrı kurulum sürelerine sahip, BATÇ problemi
2021	<u>Allahverdi vd.</u>	Maksimum gecikme performans ölçütü ile iki makineli beklemesiz akış hatları için daha iyi bir baskınlık ilişkisi ve sezgiseller
2021	<u>Başar ve Engin</u>	Bulanık ortamda hazırlık süreli BATÇ problemi
2021	<u>Zhu vd.</u>	BATÇ için kuantumdan esinlenilmiş guguk kuşu eş evrimsel algoritması

Tablo 5’te BATÇ üzerine yapılan çalışmalar incelendiğinde çözüm metotlarından hibrit yöntemlerin en fazla kullanıldığı görülmektedir. Üç ve üçten fazla makine içeren çizelgeleme ortamlarında sorun önemli derecede NP-zor olduğundan matematiksel modeller makul süre içerisinde optimum sonuç verememektedir. Bunun için daha kısa süre içerisinde kabul edilebilir bir hata payı ile optimum sonuca yakın sonuç elde etmek amacıyla yapılan çalışmalarda sezgisel ve meta-sezgisel yöntemlerin kullanımına başvurulmuştur. Yapılan çalışmalar bu kadarı ile sınırlı kalmamış hem CPU süresinin hem de optimum sonuca yaklaşma hata payının azaltılması için farklı matematiksel, sezgisel ve meta-sezgisel çözüm yöntemleri birbirleri arasında birleştirilerek kullanılmıştır.

Literatürde, son yirmi beş yılda yapılan çalışmalar analiz edildiğinde aşağıdaki bulgulara ulaşılmıştır.

BATÇ problemlerinin çözümünde, performans kriteri olarak çoğunlukla (%76) maksimum tamamlanma zamanı (C_{max}) minimizasyonu seçilmiştir.

BATÇ problemlerinin çözümünde, çok amaçlı performans kriterleri için yapılan çalışmalar az (%12) sayıdadır.

BATÇ problemlerinin çözümünde, kesin sonuç veren yöntemler ile ilgili çok az (%9) çalışma yapılmıştır. BATÇ problemlerinin çözümünde, ağırlıklı olarak meta-sezgisel yöntemler kullanılmıştır. Bu yöntemlerin içinde en çok (%22) kullanılanı Genetik Algoritmalarıdır.

BATÇ problemlerinin çözümünde, son yıllarda hibrit yöntemler çokça (%19) tercih edilmektedir.

2. Materyal ve Yöntem

Dağınık arama yöntemi ilk olarak Fred Glover tarafından 1970'lerde ortaya atılmıştır. Günümüze gelinceye kadar üzerinde önemli geliştirmeler yapılmıştır. İlk literatüre girdiği günden bu yana birçok NP-zor probleme uyarlanan dağınık arama meta-sezgiseli, oldukça etkili bir optimizasyon tekniğidir. Dağınık arama yönteminin temelinde üretilen çözümleri belleğinde tutma ve en iyi çözümlerin kombinasyonları ile daha iyi çözümler elde etmeye dayanır. Günümüzde farklı iyileştirme teknikleri ile birlikte kullanılan yöntemin, oldukça verimli versiyonları literatürde bulunmaktadır.

Dağınık arama yöntemi, 9 adımdan oluşur (Oktay vd. 2006). Bu adımlar aşağıda verilmiştir.

1. *Başlangıç popülasyonu oluşturma*; Başlangıç popülasyonu, tüm çözüm kümesine yayılmış şekilde ve iyi bireylerden oluşmalıdır. Bu aşamada çeşitli başlangıç oluşturma teknikleri kullanılabilir. Başlangıç popülasyonu oluşturulduktan sonra yine çeşitli teknikler kullanılarak popülasyon iyileştirilir.
2. *Referans küme oluşturma*; Referans kümesi iyileştirilmiş olan ve iyi bireyleri içeren popülasyondan en iyi bireylerin seçilmesi ile oluşturulur. Program sonlanıncaya kadar referans kümesi güncellemeye devam edilir.
3. *Alt küme oluşturma*; Referans kümesindeki bireylerden iyi alt kümelerin seçilmesi ile oluşturulur.
4. *Kombinasyon uygulama*; Alt küme elemanlarının birbiri ile birleştirilmesi aşamasıdır ve yeni bireyler ortaya çıkar.
5. *İyileştirme*; Yeni bireylerin çeşitli teknikler kullanılarak iyileştirilmesine dayanan aşamadır.
6. *Referans küme güncelleme*; İyileştirilmiş olan yeni bireyler ile referans kümesindeki bireyler bu aşamada kıyaslanır ve yeni bireylerin daha iyi olması durumunda referans kümesi güncellenir.
7. *Yeni referans kümesi oluşturma kriteri*; Yeni bir referans kümesine ihtiyaç duyuluncaya kadar üçten altıya kadar adımların tekrarlanması gerektiğini belirten kriterdir.
8. *Yeni popülasyon oluşturma kriteri*; Yeni bir popülasyona ihtiyaç duyuluncaya kadar ikiden yediye kadar adımların tekrarlanması gerektiğini belirten kriterdir.
9. *Sonlanma kriteri*; Programın ne zaman sonlanması gerektiğini içeren kriterdir. Program sonlanıncaya kadar 1'den 8'e kadar adımlar tekrar edilir.

Çizelgeleme problemlerine çözüm üretmek için kullanılmış dağınık arama yöntemi ile ilgili literatürde yapılmış olan çalışmalar aşağıda özetlenmiştir.

Nowicki ve Smutnicki (2006) akış tipi probleminde dağınık arama yönteminin etkinliğini incelemişlerdir. Çözüm için tabu arama yöntemi ve dağınık arama yönteminin entegrasyonu kullanmışlardır. Problem boyutu artmasına rağmen algoritmanın iyi özelliklerinin ölçeklenebilir olarak kaldığını vurgulamışlardır. Bożejko ve Wodecki (2007) ATÇ problemini, işlerin tamamlanma süreleri toplamının en aza indirilmesi kriteri ile ele almışlardır. Problemi çözmek için dağınık arama yöntemine dayalı paralel algoritma kullanmışlardır. Elde edilen sonuçları diğer çözüm yöntemleri ile kıyaslamış ve paralel hesaplamalar sonucunda süper lineer hızlanma gözlemlendiğini belirtmişlerdir. Haq, Saravanan, Vivekraj ve Prasad (2007) genel bir ATÇ problemi için dağınık arama yöntemini kullanmışlardır. Dağınık arama yönteminin Tabu arama yaklaşımı ile kıyaslama yapmışlar ve dağınık aramanın önemli derecede daha iyi çözüm verdiği açıklamışlardır. Saravanan, Haq, Vivekraj ve Prasad (2008) permütasyon ATÇ problemleri için dağınık arama yönteminin performans değerlendirmesini yapmışlardır. Amaç fonksiyonunu, işlerin tamamlanma zamanının minimizeasyonu olarak belirlemişlerdir. Engin, Kahraman ve Yılmaz (2009) çok amaçlı permütasyon, bulanık ATÇ problemini çözmek için bir dağınık arama metodu önermişlerdir. Ortalama gecikme ve geciken iş sayısını minimize etmeyi amaçlamışlardır. Kaya, Akgöbek ve Engin (2011) erken ve geç cezalı, ortak teslim tarihli ATÇ problemleri için dağınık arama yöntemini kullanmışlardır. Engin, Yılmaz, Baysal ve Sarucan (2013) kullanılabilirlik kısıtlı atölye tipi problemler için dağınık arama

yöntemini önermişlerdir. Naderi ve Ruiz (2014) dağıtılmış permütasyon ATÇ probleminin çözümü için bir dağınık arama kullanmışlardır. Amaç fonksiyonunu tamamlanma zamanı minimizasyonu olarak belirlemişlerdir. Çiçekli ve Bozkurt (2016) tarafından yapılmış olan çalışmada, akış tipi problemlerinin dağınık arama yöntemi ile çözümü yapılmıştır. Araştırmada, Reeves tarafından oluşturulmuş rec41 akış tipi problemler çözülmüştür. Riahi, Khorramizadeh, Newton ve Sattar (2017) karışık engellemeli, ATÇ problemini ele almışlardır. Çalışmada amaç kriteri olarak tamamlanma zamanını belirlemişlerdir. Problemi çözmek için dağınık aramanın başlangıç aşamasında, NEH sezgiselinin değiştirilmiş bir yöntemini kullanmışlardır. Yang, Li, Wang, Liu ve Luo (2017) dağıtılmış permütasyon ATÇ probleminde, toplam gecikmeyi minimize etmek için dağınık arama algoritmasını önermişlerdir. Problemi boşa kalma ve teslim tarihleri dahil olmak üzere gerçek uygulama kısıtlamaları dikkate alarak incelemişlerdir. Çalışmada dağınık arama tabanlı memetik algoritmayı önermişlerdir. Algoritmanın başlangıç çözümü ve alt küme oluşturma mekanizması için birleşik sezgisel tarama kullanmışlardır. İyileştirme mekanizması için ise üç aşamalı tavlama benzetimi yöntemini probleme uyarlamışlardır. Alsaïdi, Muhsen ve Ali (2020) NP-zor problemleri çözmek için Meerkat Klan algoritmasına dayalı değiştirilmiş dağınık arama algoritmasını önermişlerdir. Önerilen algoritmayı gezgin satıcı ve esnek iş çizelgeleme problemine uygulamışlardır. Çalışmalarında yapılan hesaplamalı deneyler sonucunda, önerdikleri değiştirilmiş dağınık arama algoritmasının, Meerkat Klan algoritmasından ve dağınık arama algoritmasından daha iyi olduğunu vurgulamışlardır. Tan vd. (2020) pota tahsisini dikkate alarak bir çelik üretimi-sürekli döküm çizelgeleme problemini incelemişlerdir. Çalışmalarında en hızlı üretimi ve en az enerji tüketimini sağlamak için, maksimum tamamlanma zamanını, boşa kalma cezalarını ve bekleme süresiyle ilgili enerji tüketimi cezalarını en aza indirmeyi amaçlayan karma bir tamsayı matematiksel programlama modelini sunmuşlardır. Problemi çözmek için, dağınık arama ve karma tamsayı programlamanın kombinasyonuna dayalı iki aşamalı bir yaklaşım geliştirmişlerdir.

3. Bulgular ve Tartışma

BATÇ probleminin çözümü için hibrit bir dağınık arama algoritması önerilmiştir. Algoritma, Visual Basic dilinde yazılmıştır. BATÇ problemleri, Windows Intel Core i5 64 bit işlemcili, 2.5 GHz ve 4 GB RAM'i olan bir bilgisayarda test edilmiştir. Önerilen algoritmanın adımları Şekil 1'de verilmiştir.

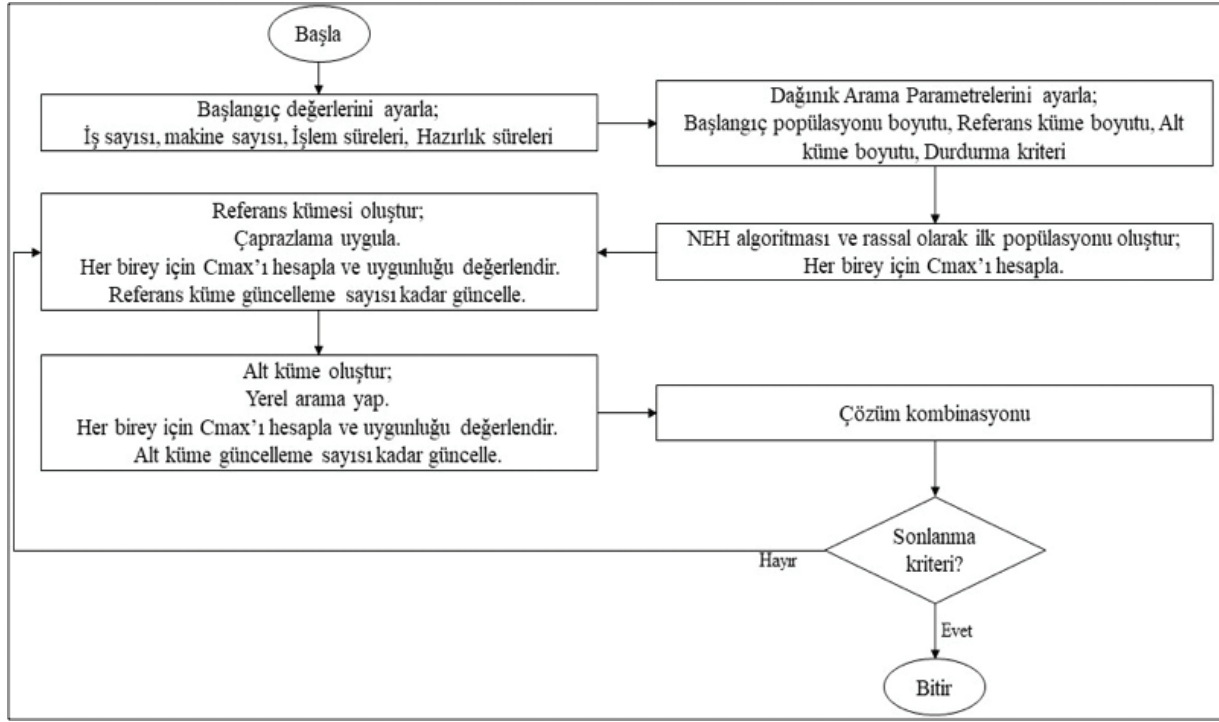
HDA algoritmasının parametreleri, tam faktöriyel deney tasarımı yardımı ile belirlenmiştir. Parametre optimizasyonu yapılırken kullanılan parametre değerleri aşağıda verilmiştir;

Başlangıç popülasyon boyutu: 5, 10, 15, 20, 25, 30
 Alt küme boyutu: 10, 20, 30, 40, 50, 60
 Alt küme güncelleme sayısı: 5, 10, 20, 30, 40, 50
 Referans küme boyutu: 3, 4, 5
 Referans küme güncelleme sayısı: 5, 10, 20, 30, 40
 İterasyon sayısı: 10, 20, 30, 40, 50, 60

Tam faktöriyel deney tasarımı sonucunda, BATÇ problemlerin çözümünde kullanılan en iyi parametre değerleri aşağıdaki gibi belirlenmiştir.

Başlangıç popülasyon boyutu: 15
 Alt küme boyutu: 30
 Alt küme güncelleme sayısı: 10
 Referans küme boyutu: 3
 Referans küme güncelleme sayısı: 10
 İterasyon sayısı: 40

HDA algoritmasının standart dağınık arama algoritmasından farklılıkları, başlangıç popülasyonu ve alt küme oluşturma sürecinden kaynaklanmaktadır. Başlangıç popülasyonu oluşturulurken NEH algoritması ve rastsal yöntem kullanılmıştır. Alt küme oluşturulurken ise yerel arama yöntemi kullanılmıştır. BATÇ probleminin çözümü ile ilgili bir örnek verilmiştir. Rastal olarak oluşturulmuş 3 iş 3 makineden oluşan bir BATÇ probleminin işlem ve hazırlık süreleri Tablo 6'da verilmiştir. Örnek problem, önerilen HDA algoritması ile çözümlenerek minimum tamamlanma süresi 40 olarak hesaplanmıştır. Bu problemin Gantt



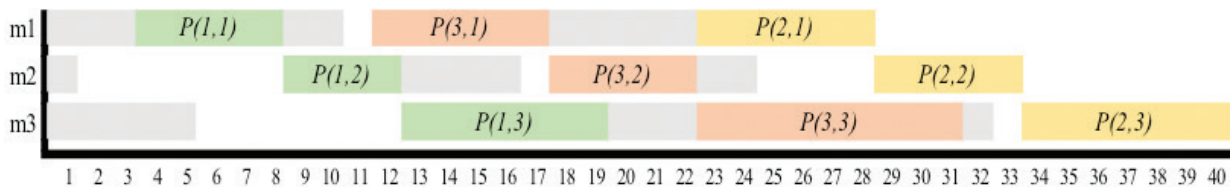
Şekil 1. Hibrit Dağılık Arama algoritmasının çözüm adımları

şeması Şekil 2’de verilmiştir. P_{ij} i işinin j makinesindeki işlem süresini, S_{ij} ise i işinin j makinesindeki hazırlık süresini ifade etmektedir.

Tablo 6

3 iş 3 makine problemi işlem ve hazırlık süreleri

İşler	j_1	j_2	j_3
P_{i1}	5	6	6
P_{i2}	4	5	5
P_{i3}	7	7	9
S_{i1}	3	5	2
S_{i2}	1	2	4
S_{i3}	5	1	3



Şekil 2. BATÇ 3 iş 3 makine problemi 1-3-2 iş sırasına ait Gantt şeması

Şekil 2’de görüldüğü üzere işlerin 1-3-2 iş sırasında 1. iş 19 birim zamanda 3. iş 31 birim zamanda 2. iş 40 birim zamanda tamamlanmaktadır. Tüm işlerin maksimum tamamlanma zamanı olan C_{max} değeri ise son makineden son işin bitiş zamanı olan 40’a eşittir.

Literatürde Engin ve Günaydın (2011) tarafından oluşturulan BATÇ kıyaslama problemleri, önerilen Hibrit Dağılık Arama algoritması ile çözülmüştür. Problemler n -iş x m -makine olmak üzere her grupta a , b ve c tipi örnekler yer almaktadır. Elde edilen sonuçlar, Engin ve Günaydın (2011) tarafından önerilen Hibrit Uyarlanabilir Öğrenme Yaklaşım (HUÖY) algoritması ve Engin ve Günaydın (2018) tarafından önerilen Hibrit Karınca Kolonileri Optimizasyon (HKKO) algoritması ile kıyaslanmıştır. Ayrıca kıyaslama problemleri, CPLEX (Doğrusal, karma tamsayı v.d. programlama için matematiksel bir çözüm aracı) Optimizer ile matematiksel model olarak çözülmüş optimum sonuçlara göre değerlendirilmiştir. Kıyaslama problemlerinden b grubunda yer alan 16 tanesi için önerilen HDA algoritması, HKKO algoritması ve HUÖY algoritması ile elde edilen max-tamamlanma zamanı (C_{max}) değerlerinin, CPLEX ile elde edilmiş optimum $-C_{max}$ değerlerinden yüzde sapmaları denklem 3.1 yardımı ile hesaplanmıştır ve sonuçlar Tablo 7'de verilmiştir. Hazırlık süreleri [1, 10] ve işlem süreleri [1, 50] arasında uniform dağılıma uygun olarak türetilen bu problemler b grubu olarak isimlendirilmektedir.

$$Sapma (\%) = \frac{(Hibrit\ Dağılık\ Arama\ Cmax) - (Optimum\ Cmax)}{Optimum\ Cmax} \times 100 \quad (3.1)$$

Tablo 7

Sezgisel yöntemlerin karşılaştırma sonuçları

Problem $n \times m - b$	Cmax				%Sapma		
	HKKO	HUÖY	HDA	CPLEX	HKKO	HUÖY	HDA
8x2	255	255	255	255	0.0000	0.0000	0.0000
8x3	302	304	290	290	4.1379	4.8276	0.0000
8x5	398	398	396	396	0.5051	0.5051	0.0000
8x8	518	518	518	518	0.0000	0.0000	0.0000
8x10	602	602	602	602	0.0000	0.0000	0.0000
8x15	775	777	773	773	0.2587	0.5174	0.0000
8x20	892	892	887	887	0.5637	0.5637	0.0000
8x25	1062	1069	1050	1050	1.1429	1.8095	0.0000
10x2	342	344	337	337	1.4837	2.0772	0.0000
10x3	362	367	362	362	0.0000	1.3812	0.0000
10x5	501	513	462	462	8.4416	11.039	0.0000
10x8	614	623	571	571	7.5306	9.1068	0.0000
10x10	656	679	612	612	7.1895	10.947	0.0000
10x15	856	879	849	849	0.8245	3.5336	0.0000
10x20	1062	1081	1043	1043	1.8217	3.6433	0.0000
10x25	1215	1248	1195	1195	1.6736	4.4351	0.0000

Tablo 7'de görüldüğü üzere HDA yöntemi ile 16 problemin tamamında optimum çözüm bulunmuştur. 16 test problemi için HKKO algoritması ortalama %2.223 HUÖY algoritması %3.3992 ve önerilen HDA algoritması da %0.0000 optimum sonuçtan sapma değerine ulaşmıştır. Algoritmaların maksimum CPU zamanları; HKKO algoritması için 110 saniye, HUÖY algoritması için 110 saniye ve önerilen HDA algoritması için 155 saniye olarak ölçülmüştür.

4. Sonuçlar

Bu çalışmada, BATÇ problemlerinin literatür analizi yapılmış ve bu tür problemleri çözmek için Hibrit Dağınık Arama yöntemi önerilmiştir. Hibrit Dağınık Arama metodunun etkinliğini kanıtlamak için literatürde yer alan kıyaslama problemleri çözümlenerek elde edilen sonuçlar, [Engin ve Günaydın \(2011\)](#) tarafından önerilen Hibrit Uyarlanabilir Öğrenme Yaklaşım algoritması ve [Engin ve Güçlü \(2018\)](#) tarafından önerilen Hibrit Karınca Kolonileri Optimizasyon algoritması ile karşılaştırılmıştır. HDA algoritması 16 problemin tamamında optimum sonuç vermiştir. HUÖY algoritması 16 problemde 3 tanesinde optimum sonuç vermiştir. HKKO algoritması da 16 problemde 4 tanesinde optimum sonuca ulaşmıştır. Önerilen HDA algoritması, BATÇ problemlerinin çözümünde başarılı sonuçlar vermektedir.

Gelecekte yapılacak olan çalışmalarda, HDA algoritmasının, çok amaçlı BATÇ problemlerinin çözümünde kullanılabileceği önerilmektedir. BATÇ problemleri üzerine bundan sonra yapılacak çalışmalarda, dağınık arama algoritması için farklı başlangıç çözümleri ve iyileştirme metodları kullanılarak geliştirilmesi, çözüm kalitesinin artmasına ve işlem sürelerinin azaltılmasına katkı sağlayacağı öngörülmektedir.

Yazar Katkıları

Ramazan Başar: Metodoloji, Yazılım, Veri analizi, Yazma-orijinal taslak, Görselleştirme, Araştırma.
Orhan Engin: Denetim, Metodoloji, Veri analizi, Yazma, Doğrulama, İnceleme, Kavramsallaştırma.

Çıkar Çatışması

Yazarlar çıkar çatışması bildirmemişlerdir.

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, *International Journal of Advanced Manufacturing Technology*, 70, 1181-1188. <https://link.springer.com/content/pdf/10.1007/s00170-013-5351-9.pdf>
- 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. <https://doi.org/10.1016/j.ejor.2014.04.031>
- 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. <https://doi.org/10.1016/j.amc.2019.124688>
- 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. <http://dx.doi.org/10.3934/jimo.2020054>
- 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. <http://dx.doi.org/10.21533/pen.v8i3.1563.g642>
- 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, *International Journal of Advanced Manufacturing Technology*, 75, 1017-1033. <https://link.springer.com/content/pdf/10.1007/s00170-014-6177-9.pdf>
- Aydilek, H. & Allahverdi, A. (2012). Heuristics for no-wait flowshops with makespan subject to mean completion time, *Applied Mathematics and Computation*, 219, 351-359. <https://doi.org/10.1016/j.amc.2012.06.024>
- 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, Cham* 607-614. https://link.springer.com/chapter/10.1007/978-3-030-85626-7_71
- 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. https://jestec.taylors.edu.my/Vol%2014%20issue%202%20April%202019/14_2_8.pdf

- 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), 1-17. <https://doi.org/10.3390/a10040121>
- 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. https://link.springer.com/chapter/10.1007/978-3-540-68111-3_20
- Chaudhry, I.A., Elbadawi, I.A.Q., Usman, M. & Chughtai, M.T. (2018). Minimising total flowtime in a no-wait flow shop (nwfs) using genetic algorithms, *Ingeniería E Investigación*, 38(3), 68-79. <https://doi.org/10.15446/ing.investig.v38n3.75281>
- 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. <https://link.springer.com/article/10.1007/s11518-020-5456-2>
- Cheng, C. Y., Ying, K. C., Li, S. F. & Hsieh, Y. C. (2019). Minimizing makespan in mixed no-wait flowshops with sequence-dependent setup times, *Computers & Industrial Engineering*, 130, 338-347. <https://doi.org/10.1016/j.cie.2019.02.041>
- Chihaoui, F.B., Kacem, I., Hadj-Alouane, A.B., Dridi, N. & Rezg, N. (2011). No-wait scheduling of a two-machine flow-shop to minimise the makespan under non-availability constraints and different release dates, *International Journal of Production Research*, 49(21), 6273-6286. <https://doi.org/10.1080/00207543.2010.531775>
- Czogalla, J. & Fink, A. (2012). Fitness landscape analysis for the no-wait flow-shop scheduling problem, *J Heuristics*, 18: 25-51. <https://link.springer.com/content/pdf/10.1007/s10732-010-9155-x.pdf>
- Çiçekli, U. G. & Bozkurt, S, (2016). Permütasyon akış tipi çizelgeleme probleminin dağınık arama ile optimizasyonu, *Ege Academic Review*, 16, 31-40. Doi: 10.21121/eab.2018OZEL24421
- 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. <https://doi.org/10.1177/1687814015622900>
- 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. <https://doi.org/10.1016/j.asoc.2018.08.002>
- 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. <https://doi.org/10.1080/18756891.2011.9727810>
- 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.): Computer Intelligence in Flow Shop and Job Shop Scheduling*, SCI 230, 169-189. https://link.springer.com/chapter/10.1007/978-3-642-02836-6_6
- 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, *Journal of Multiple-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. [https://doi.org/10.1016/S0360-8352\(99\)00127-8](https://doi.org/10.1016/S0360-8352(99)00127-8)
- 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. <https://doi.org/10.1016/j.promfg.2020.02.097>
- Gao, F., Liu, M., Wang, J.J. & Lu, Y.Y. (2018). *International Journal of Production Research*, 56(6), 2361-2369. <https://doi.org/10.1080/00207543.2017.1371353>
- 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. <https://link.springer.com/content/pdf/10.1007/s00170-005-0244-1.pdf>

- 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. <https://doi.org/10.1080/00207543.2011.653012>
- Kaya, S., Akgöbek Ö. & Engin, O. (2011). Scatter search metaheuristic for solving the flow shop scheduling with earliness and tardiness penalties, *NWSA e-Journal of New World Science Academy, Engineering Sciences*, 6(1), 415- 420. <https://dergipark.org.tr/tr/download/article-file/186280>
- Keskin, K. & Engin, O. (2021). A hybrid genetic local and global search algorithm for solving no-wait flow shop problem with bi criteria, *SN Applied Sciences*, 3, 628: 1–15. <https://doi.org/10.1007/s42452-021-04615-3>
- Koulamas, C. & Kyparisis, G. J. (2021). The no-wait flow shop with rejection, *International Journal of Production Research*, 59(6), 1852-1859. <https://doi.org/10.1080/00207543.2020.1727042>
- 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. [https://doi.org/10.1016/S0360-8352\(00\)00035-8](https://doi.org/10.1016/S0360-8352(00)00035-8)
- Labidi, M., Kooli, A., Ladhari, T., Gharbi, A. & Suryahatmaja, U.S. (2018). *IEEE Access*, 6, 16294-16304. <https://ieeexplore.ieee.org/abstract/document/8315022>
- Laha, D. & Gupta, J.N.D. (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. <https://doi.org/10.1016/j.cie.2016.06.003>
- 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. <https://doi.org/10.1016/j.asoc.2020.106946>
- 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. <https://doi.org/10.1016/j.cie.2018.12.025>
- 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, *International Journal of Advanced Manufacturing Technology*, 36, 969-981. <https://link.springer.com/article/10.1007/s00170-006-0906-7>
- 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. <https://doi.org/10.1080/0952813X.2012.682752>
- 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. <https://doi.org/10.1016/j.ejor.2014.05.024>
- Nowicki, E. & Smutnicki, C. (2006). Some aspects of scatter search in the flow-shop problem, *European Journal of Operational Research*, 169(2), 654-666. <https://doi.org/10.1016/j.ejor.2004.08.021>
- Oktay, S. & Engin, O. (2006). Endüstriyel Problemlerin Çözümünde Dağılık Arama Yöntemi: Literatür Araştırması, *Sigma: Mühendislik ve Fen Bilimleri Dergisi*, 24(3), 144-155. <https://kutuphane.dogus.edu.tr/mvt/pdf.php>
- Pourhejazy, P., Lin, S. W., Cheng, C. Y., Ying, K. C. & Lin, P. Y. (2020). Improved beam search for optimizing no-wait flowshops with release times, *IEEE Access*, 8, 148100-148124. <https://en.x-mol.com/paper/article/1296865714041921536>
- 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. <https://doi.org/10.1080/00207543.2016.1150615>
- Riahi, V., Khorramzadeh, M., Newton, M. H. & Sattar, A. (2017). Scatter search for mixed blocking flowshop scheduling, *Expert Systems with Applications*, 79, 20-32. <https://doi.org/10.1016/j.eswa.2017.02.027>
- Samarghandi, H. (2015a). A no-wait flow shop system with sequence dependent setup times and server constraints, *IFAC-PapersOnLine*, 48(3), 1604-1609. <https://doi.org/10.1016/j.ifacol.2015.06.315>
- 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. <https://doi.org/10.1080/00207543.2014.974846>

- 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. <https://doi.org/10.1016/j.cor.2016.12.013>
- Samarghandi, H. & ElMekkawy, T.Y. (2012). A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion, *International Journal of Advanced Manufacturing Technology*, 61, 1101-1114. <https://link.springer.com/content/pdf/10.1007/s00170-011-3766-8.pdf>
- 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. <https://link.springer.com/content/pdf/10.1007/s00170-007-1053-5.pdf>
- Schaller, J. & Valente, J. M. (2020). Minimizing total earliness and tardiness in a no-wait flow shop, *International Journal of Production Economics*, 224, 107542. <https://doi.org/10.1016/j.ijpe.2019.107542>
- 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. <https://doi.org/10.1016/j.omega.2013.07.001>
- Shao, W., Pi, D. & Shao, Z. (2017). An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem, *Applied Soft Computing*, 61, 193-210. <https://doi.org/10.1016/j.asoc.2017.08.020>
- Strusevich, V.A. (1995). Two machine flow shop scheduling problem with no wait in process: Controllable machine speeds, *Discrete Applied Mathematics*, 59, 75-86. [https://doi.org/10.1016/0166-218X\(93\)E0153-P](https://doi.org/10.1016/0166-218X(93)E0153-P)
- 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. <https://doi.org/10.1080/00207543.2018.1510559>
- 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. <https://ieeexplore.ieee.org/document/9078850>
- Tasgetiren, 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. <https://doi.org/10.1016/j.promfg.2020.01.347>
- Wang, S. & Liu, M. (2013). A genetic algorithm for two-stage no-wait hybrid flow shop Scheduling problem, *Computers & Operations Research*, 40, 1064-1075. doi.org/10.1016/j.cor.2012.10.015
- 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. <https://doi.org/10.1016/j.cor.2012.10.015>
- 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. <https://ieeexplore.ieee.org/document/7969399>
- Zhao, F., He, X. & Wang, L. (2020). A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem, *IEEE Transactions on Cybernetics*. <https://pubmed.ncbi.nlm.nih.gov/33095728/>
- Zhao, F., He, X., Zhang, Y., Lei, W., Ma, W., Zhang, C. & Song, H. (2020a). A jigsaw puzzle inspired algorithm for solving large-scale no-wait flow shop scheduling problems, *Applied Intelligence*, 50(1), 87-100. <https://link.springer.com/article/10.1007/s10489-019-01497-2>
- Zhao, F., Zhang, L., Liu, H., Zhang, Y., Ma, W., Zhang, C. & Song, H. (2019). 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. doi.org/10.1080/0305215X.2018.1542693
- 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. <https://ieeexplore.ieee.org/document/8630830>
- Zhu, H., Luo, N. & Li, X. (2021). A quantum-inspired cuckoo co-evolutionary algorithm for no-wait flow shop scheduling, *IET Collaborative Intelligent Manufacturing*. <https://doi.org/10.1049/cim2.12002>
- 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. <https://link.springer.com/article/10.1007/s10489-018-1285-0>