



**Makale / Research Paper**

**Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemlerinin Yapay Arı Koloni Algoritması İle Çözümü**

**İpek KÜPELİ<sup>1</sup>, Ahmet SARUCAN<sup>2\*</sup>, Ahmet Sezer KÜPELİ<sup>3</sup>**

<sup>1</sup>Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Endüstri Mühendisliği ABD. Konya/TÜRKİYE

<sup>2</sup>Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Endüstri Mühendisliği Bölümü.  
Konya/TÜRKİYE

<sup>3</sup>ASC Hidrolik A.Ş. Konya/TÜRKİYE

\*  
[asarucan@ktun.edu.tr](mailto:asarucan@ktun.edu.tr)

**Received/Geliş:** 05.01.2020

**Accepted/Kabul:** 20.03.2020

**Öz:** Bu çalışmada amaç fonksiyonu maksimum tamamlanma süresinin minimizasyonu olan dağıtık permütasyon akış tipi çizelgeleme problemleri ele alınmıştır. Bu problemin klasik akış tipi çizelgeleme probleminden farkı, işlerin birden fazla fabrikaya dağıtılmasıdır. Problemin çözümünde doğadaki arıların besin arama davranışını temel alan yapay arı koloni algoritması kullanılmıştır. Algoritmanın başlangıç çözümleri üretmesinde NEH (Nawaz Ensore Ham) sezgiselinden yararlanılmıştır. Algoritmanın evrelerinde (işçi, gözlemci ve kâşif arı evresi) komşu çözümler için yer değiştirme metodu kullanılmıştır. Bu metotta rastgele seçilen iki işin yerleri değiştirilerek farklı iş sıraları elde edilmiştir. Algoritmanın problem üzerindeki başarısı literatürde iyi bilinen Taillard'ın küçük ve büyük boyutlu test problemleri kullanılarak ölçülmüştür. Algoritma, 14 adet sezgisel ile karşılaştırılmıştır ve en iyi sonuçları sunmuştur.

**Anahtar kelimeler:** Dağıtık permütasyon akış tipi çizelgeleme; NEH sezgiseli; yapay arı kolonisi algoritması

**Solution of the Distributed Permuted Flow Shop Scheduling Problems and Artificial Bee Colony Algorithm**

**Abstract:** In this study, distributed permutation flow type scheduling problems, whose purpose function is minimization of maximum completion time, are discussed. The difference of this problem from the classical flow-type scheduling problem is that jobs are distributed across multiple factories. Artificial bee colony algorithm, based on the foraging behavior of bees in the nature, was used to solve the problem. The NEH (Nawaz Ensore Ham) intuition has been used to produce the initial solutions of the algorithm. In the phases of the algorithm (worker, observer and explorer bee phase), the displacement method is used for neighboring solutions. In this method, different jobs were obtained by changing the positions of two randomly selected jobs. The success of the algorithm on the problem has been measured in the literature using Taillard's small and large test problems. The algorithm has been compared to 14 intuitive and has presented the best results.

**Keywords:** Distributed permutation flow shop scheduling; NEH heuristic; artificial bee colony algorithm

**1. Giriş**

Büyük imalat firmaları birden fazla üretim merkezi işletmektedir. Çizelgeleme problemleri ile ilgili olarak, hangi fabrikanın hangi ürünü ürettiği önemli bir husustur. Makine çizelgeleme problemleri Johnson'un çalışmasından bu yana 60 yıldan uzun bir süredir yoğun bir şekilde incelenmiştir. Genellikle üretim planlama, üretim şirketleri içindeki operasyon yönetimi bölümlerinde operasyonel bir seviyede görünür. Kaynaklar, tipik olarak makineler, sınırlı ve maliyetlidir ve üretim

*Bu makaleye atıf yapmak için*

Küpelî, İ., Sarucan, A., Küpelî, A. S., "Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemlerinin Yapay Arı Koloni Algoritması İle Çözümü" El-Cezeri Fen ve Mühendislik Dergisi 2020, 7(2); 549-562.

*How to cite this article*

Küpelî, İ., Sarucan, A., Küpelî, A. S., "Solution of the Distributed Permuted Flow Shop Scheduling Problems and Artificial Bee Colony Algorithm" El-Cezeri Journal of Science and Engineering, 2020, 7 (2); 549-562.

faaliyetlerinin programlanması karlılığa ve daha yüksek müşteri memnuniyetine katkıda bulunmaktadır [1].

Sezgisel ve metasezgisel algoritmalar, çizelgeleme problemlerinin çözümünde oldukça yaygın olarak kullanılmaktadır. Problemlerin çok boyutlu olması ve hesaplama karmaşıklığı, bu algoritmaların tercih edilmesinde önemli bir etkidir. Metasezgisel algoritmaların birçoğu doğadan ilham alınarak oluşturulmuştur. Bunlardan bazıları; genetik algoritma, karınca kolonisi algoritması, parçacık sürü algoritması, Yapay Arı Kolonisi Algoritması (YAKA), tabu arama vb.'dir.

Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemleri (DPATÇP), güçlü bir mühendislik geçmişi olan bir çizelgeleme problemidir. Geçmiş yıllarda çizelgeleme problemleri üzerinde çalışmalar olmasına karşın, DPATÇP ilgili çalışmalar literatürde yetersizdir ve başlangıç aşamasındadır. Dağıtık çizelgeleme konusunda araştırmalar arttıkça, DPATÇP olan ilgi son zamanlarda giderek dikkat çekmeye başlamıştır [2].

Bu çalışmada akış tipi çizelgeleme problemlerinin farklı bir türü olan DPATÇP ele alınmıştır. Problemin incelenmesinde YAKA'nın kullanılabileceği gösterilmiştir.

## 2. Kaynak Araştırması

Literatürde DPATÇP ile ilgili farklı kısıtları ele alan, farklı yaklaşımların kullanıldığı çalışmalar bulunmaktadır. Bu çalışmalar ve akış tipi çizelgeleme problemlerine çözüm sunan YAKA algoritmasına ilişkin kaynak taraması aşağıdaki gibi sunulmuştur.

Ruiz ve ark. [1] maksimum tamamlanma süresini en aza indirmek için DPATÇP'nin çözümünde başlama, inşa ve imha prosedürlerini içeren yerel bir arama olan tekrarlı açgözlü algoritmayı önermişlerdir.

Ling-Fang ve ark. [2] dağıtık boşluksuz permütasyon akış tipi çizelgeleme probleminin tamamlanma süresi minimizasyonu için iki aşamalı bir memetik algoritma önermişlerdir. Problemin boşluksuz özelliğine göre, ilk popülasyonu üretmek için hibrit bir yöntem, aramayı ve kullanımı dengelemek için de iki arama aşaması tasarlamışlardır. Her aşamada, fabrika görevlerini ve iş sırasını ayarlamak için özel operatör kullanmışlardır. 420 küçük ve 660 büyük ölçekli kıyaslama örneği ile sayısal testler yapılarak etkinliğini test etmişlerdir.

Pan ve ark. [3] DPATÇP'nin toplam akış süresi kriteri ile çözülmesi için ayrık YAKA'yı önermişlerdir. Ekleme operatörünü arama sürecinde komşu çözümler üretmek üzere uyarlamışlardır. İşçi, gözlemci ve kâşif arıları içeren arama aşamalarını probleme göre tasarlamışlardır. DPATÇP'nin toplam akış süresi kriteri ile çözülmesi için önerilen ayrık YAKA'nın etkinliğini, 720 test problemi kullanılarak göstermişlerdir.

Fernandez-Viagas ve ark. [4] toplam akış süresini en aza indirmek için DPATÇP'yi ele almışlardır. Bilgisayar çözümü sürelerinde yüksek kaliteli çözümler elde etmek için on sekiz yapıcı sezgisel ve elde edilen çözümleri daha da geliştirmek için yinelemeli bir geliştirme algoritması önermişlerdir.

Wang ve ark. [5] tamamlanma süresi kriterine göre makine arızası altındaki DPATÇP'leri ele almak için bulanık mantık temelli melez bir dağıtık algoritma (BM-DA) önermişlerdir. Daha umut verici bir arama alanı keşfetmek için BM-DA, dağıtık algoritma tahmin modelini, yeni birey üretmek için genetik algoritmanın çaprazlama ve mutasyon operatörleri ile melezleştirmişlerdir. BM-DA'da, olasılıksal model tarafından üretilen bireylerin oranını dinamik olarak ayarlayarak popülasyon çeşitliliğini korumak için yeni bir bulanık mantık temelli uyarlamalı evrim stratejisini benimsemişlerdir.

Deng ve Wang [6] çok hedefli DPATÇP'yi tamamlanma süresi ve toplam gecikme kriterleri ile çözmek için rekabetçi memetik algoritma önermişlerdir. Önerilen algoritmada iki farklı hedefe karşılık gelen iki popülasyon kullanmışlardır. Her popülasyon için hedefe özgü bazı operatörler ve iki popülasyon arasında özel bir etkileşim mekanizması tasarlamışlardır.

Duan ve ark. [7] hem makine hem de iş sırasına bağlı hazırlık süresi kısıtlamalarını DPATÇP 'de ele almışlardır.

Bargaoui ve ark. [8] DPATÇP maksimum tamamlanma süresini en aza indirmeyi amaçlayan bir meta sezgisel olan kimyasal reaksiyon optimizasyon algoritmasını önermişlerdir. Bu algoritmada, NEH sezgiselini ilk molekül popülasyonunu oluşturmak için uyarlamışlardır. Önerilen algoritmanın performansını doğrulamak için Taillard'ın 720 büyük örneği üzerinde test etmişlerdir.

Li ve ark. [9] her fabrika için farklı taşıma süreleri ve yükleme kapasitesini dikkate alan DPATÇP'yi önermişlerdir. Ayrıca, farklı fabrikalara olan mesafelerin eşit olmadığı, nakliye programının ve yükleme kapasitelerini de dikkate almışlardır. Maksimum tamamlanma süresini en aza indirmek amacıyla problemi çözmek için tavlama benzetimi tabanlı yerel arama kullanmışlardır.

Wang ve ark. [10] DPATÇP'yi çözmek için hibrit ayırık guguk kuşu arama algoritmasını önermişlerdir. Algoritma için, DPATÇP'nin çözümünde parametre ayarının etkisini araştırmışlardır. Ayrıca, 420 küçük ve 720 büyük boyutlu örneğe dayalı sayısal simülasyonlar gerçekleştirmişlerdir.

Naderi ve Ruiz [11] DPATÇP'de yakın zamanda birden fazla fabrikanın işlerini işlemek için uygun olduğu düzenli akış tesis ayarının bir genellemesi olarak önermişlerdir. Problemi, fabrikalara iş atama ve her fabrikaya atanan işlerin zamanlanması şeklinde boyutlandırmışlardır. Tamamlanma süresini optimize etmede bu problem için bir dağılım arama yöntemi önermişlerdir.

Lin ve ark. [12] tüm fabrikalar arasında maksimum tamamlanma süresini en aza indirmek için değiştirilmiş bir tekrarlı açgözlü algoritma önermişlerdir. Önceki yaklaşımlarla karşılaştırıldığında, önerilen algoritma DPATÇP 'nin çözümünde daha basit ancak daha etkili ve verimli olduğunu gözlemlemişlerdir. Önerilen tekrarlı açgözlü algoritmanın performansını doğrulamak için, Taillard'ın test problemlerini kullanmışlardır.

Wang ve ark. [13] DPATÇP'yi çözmek için etkin dağıtım algoritması tahmini önermişlerdir. İlk olarak, en erken tamamlama süreli fabrikası kuralı, uygulanabilir programlar oluşturmak ve program objektif değerini hesaplamak için permütasyona dayalı kodlama için kullanmışlardır. Ardından, çözüm alanının olasılık dağılımını tanımlamak için bir olasılık modeli oluşturmuş ve olasılık modelini üstün bireylerle güncellemek için bir mekanizma sağlamışlardır. Olasılık modelini örnekleyerek, gelecek vaat eden arama bölgesi arasında yeni bireyler oluşturmuşlardır. Ayrıca, yerel sömürüyü geliştirmek için, bazı yerel arama operatörleri problem özelliklerine dayanarak tasarlanmış ve gelecek vaat eden bireyler için kullanmışlardır. Ayrıca, etkin dağıtım algoritmasının parametre ayarlarının etkisini, deneylerin Taguchi tasarım yöntemine dayanarak incelemiş ve uygun bir parametre ayarı önermişlerdir. Son olarak, 420 küçük boyutlu örneğe ve 720 büyük boyutlu örneğe dayalı sayısal simülasyonlar gerçekleştirmişlerdir.

Gao ve ark. [14] DPATÇP'nin çözümü için tabu stratejisinden yararlanan yeni bir tabu arama önermişlerdir. Komşu çözüm oluşturmak için işlerin alt sıralarını değiştiren bir yöntem sunmuşlardır. Gelişmiş bir yerel arama yöntemi önerilmiş ve aynı zamanda tabu arama ile birleştirilmiştir. Algoritmayı test etmek için, Taillard'ın test problemleri kullanmışlardır. Yapılan deneylerde önerilen tabu algoritmasının, diğer sezgisel algoritmaları geride bıraktığı gözlenmiştir.

Gao ve Chen [15] NEH tabanlı bir sezgisel algoritma önermişlerdir. DPATÇP'nin gönderme kuralında, fabrikalara bir seferde bir işi yerleştirmek yerine, bir seferde bir grup iş yerleştirmektedir. Sonuçta, önerilen algoritmanın çözüm kalitesinin mevcut sezgiselden daha iyi olduğunu ortaya koymuşlardır.

Gao ve Chen [16] maksimum tamamlanma süresini en aza indirmek amacıyla genetik algoritma tabanlı bir algoritma önermişlerdir. Önerilen algoritmada, çaprazlama ve mutasyon operatörleri, kısmi iş dizileri kümesinin kullanıldığı DPATÇP çözümleri uygun hale getirmek üzere tasarlanmıştır. Ayrıca, komşu çözümleri keşfetmek için etkili bir yerel arama yöntemi kullanmışlardır.

Liu ve Gao [17] DPATÇP için sezgisel bir çözüm yöntemi olarak elektromanyetizma benzeyen bir mekanizma algoritması uygulanmasını önermişlerdir. Bir DPATÇP'nin ayrık etki alanına uyması için; mesafe, hareket, yük ve kuvvet hesaplamasının yeniden tanımlanması dahil olmak üzere orijinal elektromanyetizma algoritmasında değişiklikler yapmışlardır. Önerilen algoritmanın performansını arttırmak için, kritik fabrika konseptini kullanan bir değişken komşu arama mekanizması geliştirmişlerdir. Naderi'nin Taillard örneklerinden uyarlanmış test problemleri kullanılarak yeni en iyi bilinen çözümler bulmuşlardır.

Naderi ve Ruiz [18] permütasyon akış tipi çizelgeleme problemi temel alınarak birden fazla fabrikanın dahil olduğu yeni bir problem ortaya çıkarmışlardır. Bu genelleme kapsamında, her biri seri halinde yerleştirilmiş  $m$  makineli toplam  $F$  özdeş fabrika olduğu varsayılmıştır.  $F$  fabrikaları arasında bir  $n$  kullanılabilir iş kümesi dağıtılmış ve her bir fabrikaya atanan işler için bir işlem dizisi oluşturulmuştur. Optimizasyon kriteri, fabrikalar arasında tamamlanma süresinin veya üretim süresinin minimuma indirilmesi olarak belirtilmiştir. DPATÇP'yi altı farklı alternatif karışık tamsayı doğrusal programlama modeli ile tanımlamışlardır. İki adet fabrika atama kuralı kullanılarak 14 adet sezgiselin ve komşu arama prensibine dayanan algoritmanın performansını değerlendirmişlerdir.

Zhang ve Gu [19] maksimum tamamlanma süresini en aza indirmek amacıyla ara tamponlu akış tipi çizelgeleme problemi için ayrık YAKA önermişlerdir. Ekleme ve yer değiştirme operatörlerinin kombinasyonu, işçi arı safhasında komşu bireylerin üretilmesi için uygulamışlardır. Turnuva seçimi, yerel optimuma düşmekten kaçınmak için kabul edilmişken, optimize edilmiş uç operatör, yerel arama algoritmasını geliştirmek için komşu çözümü araştırmak üzere gözlemci arı aşamasına geçiş yapılmıştır. Boyutu 2 olan turnuva seçimi tekrar uygulanmış ve daha iyi bir çözüm, tekrarlı açgözlü algoritma kullanılarak imhası ve inşası yapılmış ve ardından sonuç en kötüsünün yerini almıştır.

Tasgetiren ve ark. [20] boşluksuz permütasyon akış tipi çizelgeleme problemini toplam gecikme kriteri ile çözmek için ayrık YAKA önermişlerdir. Bu problem, boşta çalışma süresinin bulunmadığı permütasyon akış tipi çizelgeleme probleminin bir çeşididir. Başka bir deyişle, bir makinede ilk işin başlama süresinin boşta kalma kısıtlamasını karşılamak için ertelenmesidir.

Cui ve Gu [21] çalışmalarında vektör temsilini kullanarak, problemin tamamlanma süresini en aza indirmek üzere Geliştirilmiş Ayrık Yapay Arı Kolonisi (GAYAK) algoritmasını önermişlerdir. GAYAK algoritmasında, işçi ve gözlemci arılara yeni çözümler üretmek için yeni bir diferansiyel evrim ve değiştirilmiş değişken komşuluk araması incelemişlerdir. Kâşif arılar, çözümler elde etmek için yıkım ve yapım prosedürleri kullanılmıştır. Simülasyon sonuçları, önerilen GAYAK algoritmasının, aynı kıyaslama durumundaki son teknoloji algoritmalara kıyasla oldukça etkili ve verimli olduğunu göstermişlerdir.

Liu ve Liu [22] permütasyon akış tipi çizelgeleme problemlerinde tamamlanma süresinin minimizasyonu için, hibrit yapay arı koloni algoritmasını önermişlerdir. Hibrit yapay arı kolonide, problemin her çözümü bir gıda kaynağı olarak adlandırılmış ve ayrı bir iş permütasyonu ile temsil edilmiştir. Belirli kalite ve çeşitliliğe sahip ilk popülasyon, NEH sezgiseline dayalı açgözlü rastgele uyarlanmış araştırma prosedüründen elde edilmiştir.

Li ve Yin [23] karma ayrık arı kolonisi algoritması olarak adlandırılan permütasyon akış tipi çizelgeleme problemi için yerel olarak en iyisine ulaşmada, kolay olan tek mutasyon planının kusurlarını telafi etmek için kompozit mutasyon stratejileri içeren ayrık yapay arı koloni algoritması önermişlerdir. Öncelikle, YAKA'nın permütasyon akış tipi çizelgeleme problemine uygun hale getirilmesi için, her bir iş permütasyon bir gıda kaynağı olarak kabul edilmiş ve farklı arılarla yeni bir komşu gıda kaynağı üretmek için farklı işlemler uygulanmıştır. NEH, popülasyonu belirli bir kalite ve çeşitlilikle başlatmak için rastgele başlatma ile birleştirilmiştir. Kompozit mutasyon stratejileri, permütasyon akış tipi çizelgeleme problemini çözmek için önerilmiştir.

Tasgetiren ve ark. [24] en küçük toplam akış zamanını veren permütasyonu bulmak için çeşitli yinelenmiş açgözlü algoritmalarla hibritleştirilmiş ayrı bir YAKA önermişlerdir. Tekrarlı açgözlü algoritma, yerleştirme ve yer değiştirme komşu yapılarına dayanan yerel arama prosedürlerle oluşturulmuştur. Önerilen algoritmaların performansı, Taillard'ın test problemleriyle test edilmiştir.

Karaboga ve Basturk [25], çok değişkenli fonksiyonları optimize etmek için YAKA kullanmışlar ve genetik algoritma, parçacık sürü optimizasyonu ve partikül sürü evrim algoritması tarafından üretilen sonuçlar ile karşılaştırmışlardır. Sonuçlar, YAKA'nın diğer algoritmaları geride bıraktığını göstermiştir.

### 3. Materyal ve Metot

#### 3.1. Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemi

DPATÇP, akış tipi çizelgeleme problemlerinin yeni bir türüdür. Bu problem üzerinde yapılan ilk çalışmalar Naderi ve Ruiz [18] tarafından yapılmıştır. DPATÇP'nin varsayımları şu şekilde tanımlanabilir:

DPATÇP'de  $N$  kümesindeki  $n$  adet iş,  $M$  kümesindeki  $m$  adet makinede işlem görmektedir.  $j$  işinin ( $j \in N$ )  $i$  makinesindeki ( $i \in M$ ) işlem süresi  $P(i, j)$  kadardır.  $n$  adet iş  $F = \{f_1, f_2, \dots, f_k\}$  adet fabrikada işlem görmektedir ( $k \in F$ ). Her  $k$  fabrika aynı makine kümesini içermektedir  $M = \{M_1, M_2, \dots, M_n\}$ . Belli bir fabrikaya bir iş atandığında, başka bir fabrikaya aktarılamaz ve tüm operasyonlar yalnızca o fabrikada yapılabilir.

$\pi^k = [\pi^k(1), \pi^k(2), \dots, \pi^k(n_k)]$  denklemi,  $k$  fabrikasındaki iş kümesini ifade eder ve denklemdeki  $n_k$ ,  $k$  fabrikasına atanan toplam iş sayısını gösterir.  $C_{i,j}$   $j$  işinin  $i$  makinesindeki tamamlanma süresini gösterir. Maksimum tamamlanma süresinin hesaplanması Denklem 1-5 ile sağlanır [13].

$$C_{\pi^k(1),1} = P_{\pi^k(1),1}, k = 1, 2, \dots, F \quad (1)$$

$$C_{\pi^k(j),1} = C_{\pi^k(j-1),1} + P_{\pi^k(j),1}, k = 1, 2, \dots, F; j = 2, 3, \dots, n_k \quad (2)$$

$$C_{\pi^k(1),i} = C_{\pi^k(1),i-1} + P_{\pi^k(1),i}, k = 1, 2, \dots, F; i = 2, 3, \dots, m \quad (3)$$

$$C_{\pi^k(j),i} = \max \{C_{\pi^k(j-1),i}, C_{\pi^k(j),i-1}\} + P_{\pi^k(j),i}, k = 1, 2, \dots, F; j = 2, 3, \dots, n_k; i = 2, 3, \dots, m \quad (4)$$

$$C_{max} = \max C_{n_k, m}, k = 1, 2, \dots, F \quad (5)$$

DPATÇP, belirli bir zamanlama amaç fonksiyonunu en aza indirmek için hem fabrikalara hem de tüm fabrikalardaki işlerin atamalarını belirlemektedir. Bu problemde klasik akış tipi çizelgeleme problemlerindeki varsayımlar benimsenmiştir.

### 3.2. Yapay Arı Koloni Algoritması

Doğadaki bal arılarının besin arama davranışları model alınarak yapılan bu algoritma Karaboga ve Basturk [25] tarafından geliştirilmiştir. Bu algortmada, besin arayan arılar 3 gruba ayrılırlar. Bunlar: işçi arılar, gözlemci arılar ve kâşif arılardır. Koloninin yarısını işçi arılar diğer yarısını da gözlemci arılar oluşturur. İşçi arılar besin kaynağı bulmaya çalışırlar, buldukları besin kaynağını hafızada tutarlar. Gözlemci arılar işçi arılara göre daha iyi besin kaynağı bulma eğilimindedirler. İşçi arıların birkaçı kâşif arılara dönüşerek yeni besin kaynakları bulmaya çalışırlar. Durdurma kriteri sağlanıncaya kadar döngü devam eder. Diğer popülasyon tabanlı algortmalara benzer şekilde, YAKA tekrar eden bir süreçtir.

#### 3.2.1. Başlangıç parametreleri

Temel YAKA'nın üç adet parametresi bulunmaktadır [26]. Bunlar:

- İşçi ve gözlemci arıların toplamına eşit olan besin kaynakların sayısı (BS),
- Besin kaynağının terk edilmesi için limit değeri ve
- İterasyon sayısıdır.

#### 3.2.2. Besin kaynaklarının oluşturulması

$X_{i,j} = \{X_{i,1}, X_{i,2}, \dots, X_{i,n}\}$  Popülasyondaki  $i$ . besin kaynaklarını göstermektedir.

Besin kaynaklarının oluşturulmasında Denklem 6'dan yararlanılır [27].

$$X_{i,j} = X_j^{\min} + r(X_j^{\max} - X_j^{\min}) \quad (6)$$

Burada  $i$  besin kaynağı sayısı ( $i = 1, 2, \dots, BS$ ),  $j$  parametre sayısını ( $j = 1, 2, \dots, n$ ) ifade etmektedir.  $r$  ise  $[0,1]$  aralığında düzgün bir rassal sayıdır.  $X_j^{\max}$  ve  $X_j^{\min}$   $j$  parametresinin alt ve üst değerlerini belirtir.

#### 3.2.3. İşçi arı aşaması

İşçi arılar besin kaynaklarına rastgele olarak gönderilirler. Her bir besin kaynağında yalnız bir işçi arı bulunabilmektedir. İşçi arılar, besin kaynağındaki nektar miktarını yani çözüm değerini hesaplayarak hafızaya alırlar. Bu aşamada, her işçi arı, şu anki  $X_{i,j}$  pozisyonunun yakınında, Denklem 7 kullanarak yeni bir besin kaynağı üretir [23].

$$V_{i,j} = X_{i,j} + \varphi_{i,j}(X_{i,j} - X_{k,j}) \quad (7)$$

Buradaki  $k$ , rasgele seçilen indekstir  $k \neq i$  ve  $j \in \{1, 2, \dots, n\}$  ve  $\varphi_{i,j}$   $[-1,1]$  aralığında düzgün dağılımlı rastgele bir sayıdır. Bulunan yeni besin kaynağı  $V_{i,j}$   $X_{i,j}$  ile kıyaslanır. Eğer  $V_{i,j}$   $X_{i,j}$ 'den daha iyi veya eşit ise,  $V_{i,j}$   $X_{i,j}$ 'nin yerini alacaktır. Aksi takdirde  $X_{i,j}$  korunacaktır.

### 3.2.4. Gözlemci arı aşaması

Gözlemci arılar, işçi arıların buldukları besin kaynaklarının kalitesini değerlendirerek bir  $X_{i,j}$  seçerler. Bu seçim, aşağıdaki Denklem 8 ile hesaplanan olasılık değerine bağlı olarak yapılır.

$$p_i = \frac{u_i}{\sum_{i=1}^{BS} u_i} \quad (8)$$

Buradaki  $u_i$ ,  $i$ . besin kaynağının uygunluk değerini ifade etmektedir.  $p_i$  ise,  $i$  besin kaynağının seçim olasılığını belirtir.  $p_i$  oranı yüksek olan besin kaynaklarının seçilme olasılığı yüksektir [27].

### 3.2.5. Kâşif arı aşaması

Bir besin kaynağı  $X_{i,j}$  limit değeri boyunca daha da iyileştirilemezse, yeni besin kaynakları bulmak için işçi arı kâşif arıya dönüşür. Kâşif arılar rastgele yeni besin kaynaklarını Denklem 9 ile bulurlar [19].

$$X_{i,j} = X_j^{min} + r(X_j^{max} - X_j^{min}) \quad (9)$$

Yeni besin kaynakları bulduktan sonra, YAKA tekrarlanır. Bu tekrarlar, iterasyon sayısı tamamlanincaya kadar yapılır.

Algoritmanın temel adımları aşağıdaki şekilde özetlenebilir [27-30]:

- Başlangıç yiyecek kaynak bölgelerinin üretilmesi
- İşçi arıların yiyecek kaynağı olan bölgeye gönderilmesi
- Seçim için olasılık değerlerinin görevli arıların verdiği bilgiye göre bulunması
- Gözcü arıların bulunan olasılık verilerine göre yiyecek kaynağı bölgesi seçmeleri
- Kaynağı bırakma kriteri: limit ve kâşif arı üretimi

### 3.3. NEH Sezgiseli

Akış tipi çizelgeleme problemlerinin çözümünde veya başlangıç çözümlerinin iyileştirilmesinde NEH sezgiseli oldukça sık kullanılmaktadır. NEH sezgiseli Nawaz, Enscore ve Ham tarafından ortaya çıkarılmıştır. Bu tip problemlerin çözümünde bu sezgisel başarılı sonuçlar ortaya koymaktadır. NEH'in adımları [31]:

*Adım 1:* Her bir işin, toplam iş süresi hesaplanır.

*Adım 2:* İşler toplam iş süresi baz alınarak azalan sırada sıralanır.

*Adım 3:* Sıradaki ilk iki iş seçilir ve tüm alternatif sıraların toplam tamamlanma süreleri hesaplanır. Tamamlanma süresi küçük olan sıra seçilir.

*Adım 4:* 2. adımda belirlenen sıradaki diğer iş seçilir ve bu iş mevcut sıradaki tüm pozisyonlara koyularak toplam tamamlanma süreleri hesaplanır. Tamamlanma süresi küçük olan sıra seçilir.

*Adım 5:* 2. adımda belirlenen sıradaki tüm işler bitinceye kadar 4. adım tekrarlanır.

## 4. Bulgular ve Tartışma

Algoritmaların, Taillard test problemlerindeki çözüm performanslarının kıyaslanmasında Bağlı Sapma Yüzdesi (BSY) değerleri kullanılmıştır. Önerilen algoritmanın performansı için aşağıdaki Denklem 10 kullanılır [18].

$$BSY = \frac{Alg_{\text{çözüm}} - Bei_{\text{çözüm}}}{Bei_{\text{çözüm}}} * 100 \quad (10)$$



Eşitlikte yer alan  $Alg_{\text{çözüm}}$  YAKA ile bulduğumuz optimal çözüm değerlerini ifade eder.  $Bei_{\text{çözüm}}$  ise, karışık tamsayılı doğrusal programlama modelleri ile bulunan, bilinen en iyi sonuçları ifade eder. Bu eşitlik kullanılarak BSY hesaplanır.

#### 4.1. Küçük Boyutlu Test Problemleri

Küçük boyutlu test problemleri 420 adet örnekten oluşmaktadır. Test problemleri: fabrika sayısı  $F = \{2,3,4\}$ , iş sayısı  $n = \{4,6,8,10,12,14,16\}$  ve makine sayısı  $m = \{2,3,4,5\}$  olmak üzere 84 kombinasyondan oluşmaktadır. Her kombinasyonun 5 örneği, toplamda 420 örnek oluşmaktadır. YAKA'nın performansı, 14 adet sezgisel algoritma [18] ile karşılaştırılmıştır. YAKA'nın BSY değerleri Çizelge 1'de verilmiştir. BSY değerleri  $F_{xn}$  olarak gruplandırılmıştır ve her satırda 20 adet veri bulunmaktadır [29].

Tablo 1: Sezgisel Algoritmaların BSY Değerleri (Küçük Boyutlu)

Örnek $F_{xn}$	SPT1	SPT2	LPT1	LPT2	Johnson1	Johnson2	CDS1	CDS2	Palmer1	Palmer2	NEH1	NEH2	VND(a)	VND(b)	YAKA
2x4	12.95	10.02	19.31	17.80	6.13	3.76	2.69	0.72	7.75	5.22	2.61	0.15	<b>0.00</b>	0.15	0.83
2x6	13.30	10.38	29.53	28.37	10.34	8.08	7.56	4.60	7.57	4.83	4.42	1.44	1.26	1.44	<b>0.26</b>
2x8	17.31	16.16	35.43	33.09	9.75	7.98	12.33	10.33	9.97	8.64	4.43	2.53	2.10	2.52	<b>0.48</b>
2x10	21.03	17.49	36.93	37.12	8.38	7.52	9.72	8.38	10.18	9.25	4.28	3.27	3.22	2.89	<b>0.96</b>
2x12	20.40	17.23	39.36	37.42	10.14	8.29	11.81	10.49	10.77	10.32	6.95	4.13	3.38	3.90	<b>1.97</b>
2x14	17.73	17.34	40.00	38.36	10.27	9.21	6.59	5.24	10.57	9.24	6.05	3.16	2.70	2.82	<b>2.60</b>
2x16	17.11	16.95	42.42	41.63	12.33	10.99	10.53	8.26	10.97	9.00	6.66	3.84	<b>2.92</b>	3.30	3.67
3x4	8.87	5.36	6.42	5.41	5.31	1.99	5.01	2.55	4.82	1.50	0.43	0.43	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
3x6	21.24	11.93	27.63	25.26	6.91	5.28	8.50	6.99	9.40	3.40	4.04	1.39	<b>0.68</b>	1.39	1.10
3x8	17.71	17.93	26.87	25.14	10.14	8.30	9.17	8.67	11.84	6.98	5.08	2.43	1.86	2.43	<b>1.46</b>
3x10	24.48	16.23	37.98	35.43	12.67	10.07	13.56	10.81	13.97	11.67	8.42	3.79	2.59	3.63	<b>1.90</b>
3x12	25.92	19.55	42.12	40.14	14.90	10.12	14.66	11.00	14.39	11.36	7.66	5.08	3.66	5.08	<b>2.90</b>
3x14	23.44	19.95	41.00	38.56	14.62	14.04	16.45	13.47	17.97	14.05	10.54	4.90	4.48	4.46	<b>4.26</b>
3x16	25.31	22.83	41.71	39.39	14.39	10.41	15.55	11.60	15.68	11.06	8.18	3.98	<b>3.50</b>	3.81	4.31
4x4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	<b>0.00</b>
4x6	13.34	11.31	14.52	12.52	7.65	5.62	4.99	3.39	6.02	2.98	3.10	0.47	<b>0.00</b>	<b>0.00</b>	0.16
4x8	15.21	13.89	16.86	16.44	9.20	5.56	9.23	8.61	10.17	7.04	4.19	<b>0.77</b>	<b>0.77</b>	<b>0.77</b>	2.23
4x10	22.65	16.84	34.47	30.64	13.19	8.93	13.90	9.30	16.21	7.23	7.07	2.30	<b>1.57</b>	2.22	2.36
4x12	25.51	20.54	38.72	34.06	13.61	8.68	18.13	14.49	15.49	12.39	8.58	4.97	4.23	4.68	<b>3.95</b>
4x14	26.61	22.31	42.89	39.79	13.50	8.51	15.47	11.23	16.53	12.25	10.94	4.54	<b>4.25</b>	4.46	5.15
4x16	28.70	24.09	44.01	40.67	19.12	13.96	17.62	13.69	19.34	15.84	9.79	5.59	<b>5.08</b>	5.55	5.76
<b>Ortalama</b>	<b>18.99</b>	<b>15.63</b>	<b>31.34</b>	<b>29.39</b>	<b>10.60</b>	<b>7.97</b>	<b>10.64</b>	<b>8.28</b>	<b>11.41</b>	<b>8.30</b>	<b>5.88</b>	<b>2.82</b>	<b>2.30</b>	<b>2.64</b>	<b>2.20</b>

YAKA diğer sezgisel yöntemlere göre oldukça başarılı sonuçlar vermektedir. Çizelge 1'deki verilere bakıldığında, 2 fabrikalı 6,8,10,12,14 işlerde, 3 fabrikalı 4,8,10,12,14 işlerde ve 4 fabrikalı 4,12 işlerde YAKA, VND (a) algoritmasına göre çok düşük sonuçlar elde etmiştir. 4 fabrika 4 işte diğer algoritmalar ile aynı sonuçları bulmuştur. Geri kalan  $F_{xn}$  örneklerde VND (a) algoritmasının sonuçlara yakın değerler bulmuştur. Ortalamaya bakıldığında YAKA'nın BSY değeri %2,20'dir ve BSY değerlerini büyükten küçüğe sıraladığımızda 1. sırada yer almaktadır.

#### 4.2. Büyük Boyutlu Test Problemleri

Büyük boyutlu test problemleri 720 adet örnekten oluşmaktadır. Test problemleri: fabrika sayısı  $F = \{2,3,4,5,6,7\}$ , iş sayısı  $n = \{20,50,100,200,500\}$  ve makine sayısından  $m = \{5,10,20\}$  oluşmaktadır.

Tablo 2: Sezgisel Algoritmaların BSY Değerleri (Büyük Boyutlu)

Örnek F	SPT1	SPT2	LPT1	LPT2	Johnson1	Johnson2	CDS1	CDS2	Palmer1	Palmer2	NEH1	NEH2	VND(a)	VND(b)	YAKA
2	18.71	17.71	33.70	32.36	12.68	11.58	9.29	8.52	10.44	9.34	2.92	1.21	0.10	0.32	<b>0.08</b>
3	19.95	17.58	33.05	31.57	13.40	11.18	10.83	8.92	11.72	9.43	3.42	1.15	0.10	0.35	<b>-0.04</b>
4	20.07	17.23	33.30	30.42	13.48	10.67	11.19	8.54	11.95	8.90	4.21	1.11	0.06	0.46	<b>0.05</b>
5	20.04	16.73	32.89	29.57	13.18	10.24	11.29	8.07	12.24	8.76	4.28	0.92	0.11	0.46	<b>0.10</b>
6	20.32	16.07	32.58	28.55	13.57	9.94	11.50	7.83	12.70	8.45	4.73	0.95	0.11	0.51	<b>0.10</b>
7	21.04	15.42	32.02	27.14	13.61	9.71	11.49	7.31	12.53	8.21	4.86	0.81	0.10	0.45	<b>0.09</b>
<b>Ortalama</b>	<b>20.02</b>	<b>16.79</b>	<b>32.92</b>	<b>29.93</b>	<b>13.32</b>	<b>10.55</b>	<b>10.93</b>	<b>8.20</b>	<b>11.93</b>	<b>8.85</b>	<b>4.07</b>	<b>1.03</b>	<b>0.10</b>	<b>0.43</b>	<b>0.06</b>



YAKA'nın performansı, 14 adet sezgisel algoritma [18] ile karşılaştırılmıştır. YAKA'nın BSY değerleri Çizelge 2'de verilmiştir. BSY değerleri  $F$  olarak gruplandırılmıştır ve her satırda 120 adet veri bulunmaktadır [29].

Büyük boyutlu test problemlerinde fabrika, makine ve iş sayıları küçük boyutlu test problemlerine göre fazla olduğundan dolayı problemin çözümünün iyileştirilmesinde NEH sezgiseli uygulanmıştır. NEH sezgiseli rastgele besin kaynakları oluşturma evresinde kullanılmıştır.

Çizelge 2'de gösterildiği gibi YAKA diğer sezgisel yöntemlere göre oldukça başarılı sonuçlar vermektedir. Çizelge 2'deki ortalama değerler temel alınarak, BSY değerlerini büyükten küçüğe sıraladığımızda YAKA 1. sırada yer almaktadır ve BSY değeri %0,06'dır.

Tüm fabrikalarda YAKA, diğer algoritmalarından daha iyi sonuçlar elde etmiştir. 3. fabrikada YAKA "-0,04" değerini bulmuştur. Diğer algoritmalar, bu fabrikada eksi değere ulaşamamıştır. Bu durum, karışık tamsayı doğrusal programlama modelleri ile bulunan sonuçlardan daha iyi sonuçlar bulunduğunu göstermektedir.

Tablo 3. Ta001 Örneğinin Sonuçları

Örnek	Fabrika	İş	Makine	Bilinen en iyi çözüm	YAKA
	F	N	m		
Ta0001_3.txt	3	20	5	598	580

3 fabrika 20 iş x 5 makineli bir örnekte en iyi çözüm değeri "598" bulunmuştur [31]. YAKA ile bu örneği çözdüğümüzde "580" değeri bulunmuştur (Çizelge 3). Her bir fabrikada bulunan  $C_{max}$  değerleri ve Genel  $C_{max}$  değeri Şekil 1'de gösterilmiştir.

Şekil 1'e göre Ta001 örneğinin YAKA ile çözümünde fabrikalara atanan iş sıraları  $C_{max}$  değeri aşağıdaki gibidir. Şekil 2-3-4'de fabrikaların Gannt şemaları verilmiştir.

İş Sayısı: 20  
Makine Sayısı: 5  
Fabrika Sayısı: 3  
Hedef Değer: 582  
İterasyon Sayısı: 1000  
Limit Sayısı: 2

İşçi An Sayısı: 5  
Gözetimci An Sayısı: 5

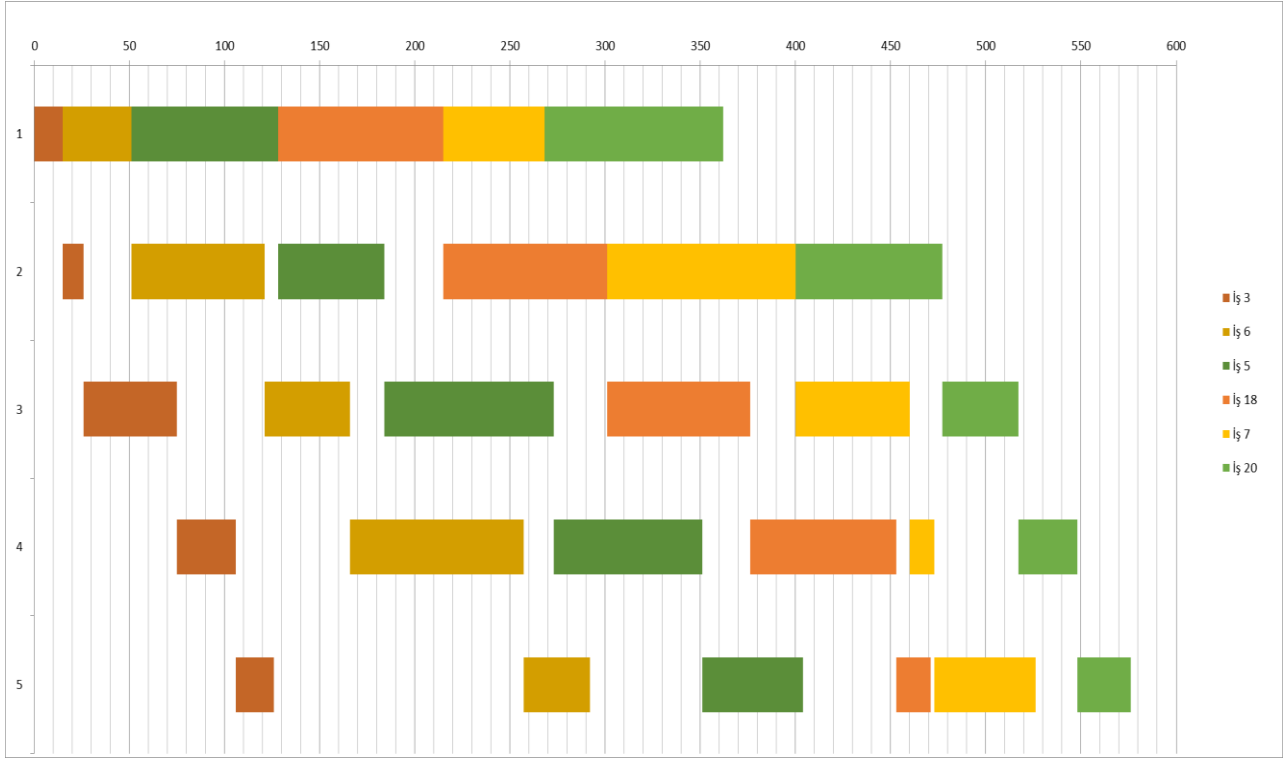
SONUÇ

Genel Cmax = 580  
1. Fabrika Sırası / Cmax = 3-6-5-18-7-20 / 576  
2. Fabrika Sırası / Cmax = 9-14-16-11-2-13-12 / 578  
3. Fabrika Sırası / Cmax = 17-15-1-4-19-8-10 / 580

Şekil 1: Ta001 Örneğinin YAKA ile Çözümünden Elde Edilen Sonuçlar

1. Fabrikadaki iş sırası 3-6-5-18-7-20

$C_{max} = 576$



Şekil 2. 1. Fabrikanın Gantt Şeması

2. Fabrikadaki iş sırası 9-14-16-11-2-13-12

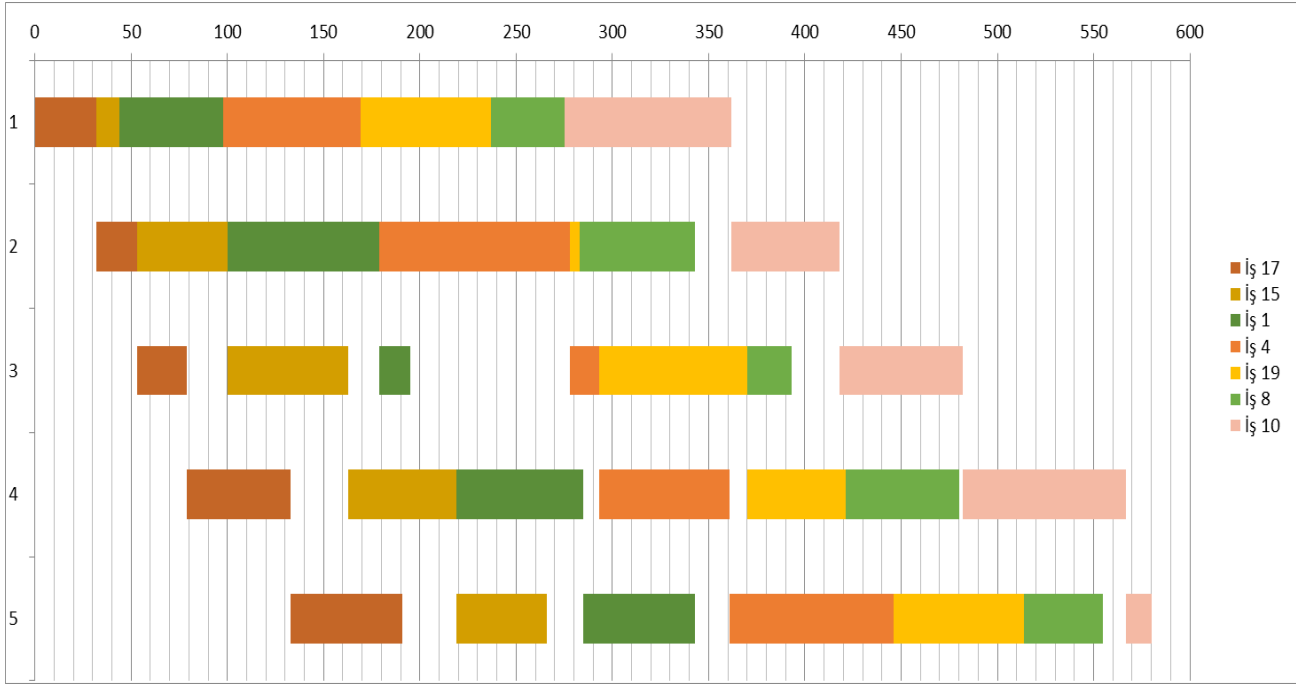
$C_{max} = 578$



Şekil 3. 2. Fabrikanın Gantt Şeması

## 3. Fabrikadaki iş sırası 17-15-1-4-19-8-10

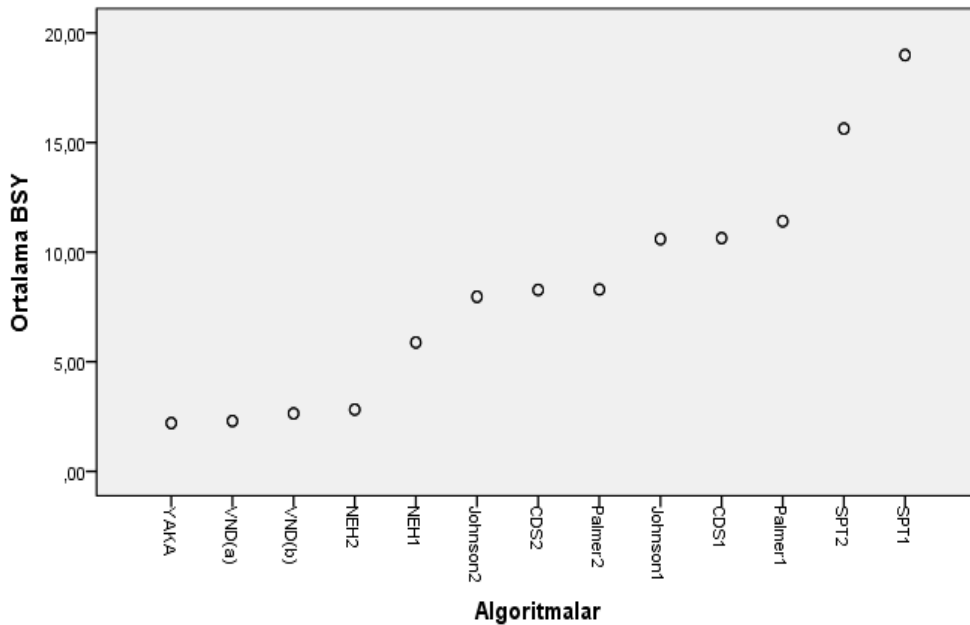
$$C_{max} = 580$$



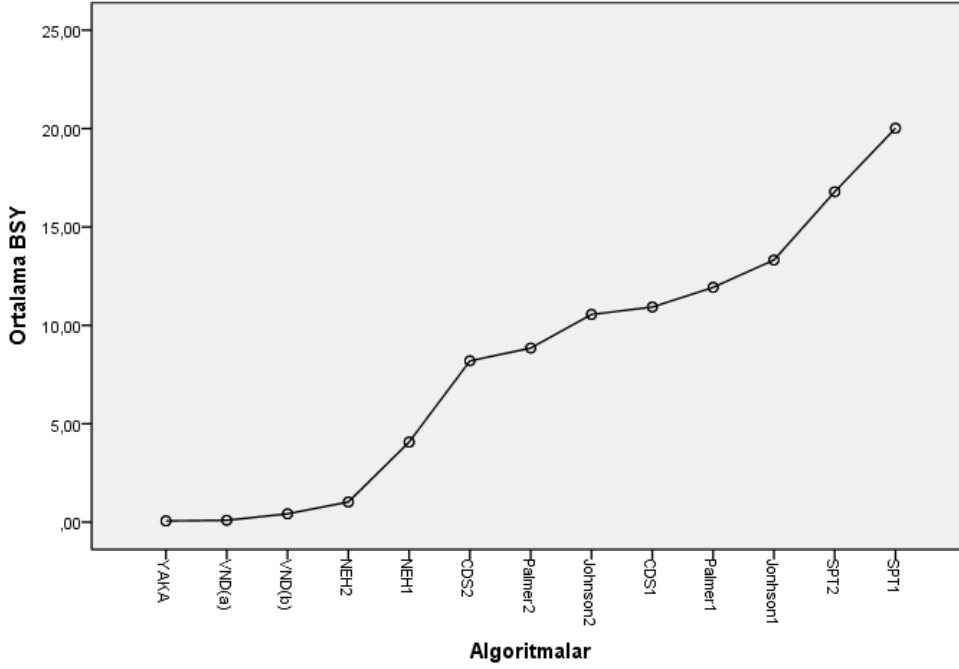
Şekil 4. 3. Fabrikanın Gantt Şeması

Genel  $C_{max}$  değeri ise en büyük  $C_{max}$  değeri seçilerek “580” olarak belirlenir. Eşitlik (10) kullanılarak Ta001 örneği YAKA ile çözüldüğünde BSY değeri “-3,01” elde edilmiştir.

Çizelge 1 ve 2’deki BSY değerlerinin arasındaki farkın istatistiksel olarak anlamlı olup olmadığını kontrolü için ANOVA One Way testi kullanılmıştır.  $13 \times 420 = 5460$  adet veri kullanılarak Şekil 5 ve  $13 \times 720 = 9360$  adet veri kullanılarak Şekil 6 oluşturulmuştur.



Şekil 5. Sezgisel Algoritmaların %99 Güven Aralığındaki BSY Değerleri (Küçük Boyutlu)



Şekil 6. Sezgisel Algoritmaların %99 Güven Aralığındaki BSY Değerleri (Büyük Boyutlu)

LPT1 ve LPT2 algoritmalarının BSY değerleri diğer algoritmalara göre çok yüksek olduğu için grafiklere dâhil edilmemiştir. Burada parametrik ANOVA'nın her üç hipotezi (normallik, eş varyanslılığı ve bağımsızlığı) kontrol edilmektedir. Yanıt değişkeni BSY'dir ve kontrol edilen faktörler  $n$ ,  $m$  ve  $F$ 'dir. BSY yanıt değişkeninde istatistiksel olarak anlamlı farklılıklar oluşmaktadır. %99 güven aralığında anlamlı fark aralıklarını göstermektedir.

Burada YAKA'da bulduğumuz sonuçların istatistiksel olarak diğer algoritmaların sonuçları ile kıyasladığımızda Çizelge 1 ve 2'deki sonuçları destekler nitelikte olduğu görülmektedir.

## 5. Sonuç ve Öneriler

Klasik akış tipi çizelgeleme problemlerinden farklı olarak, işler birden fazla fabrikada her fabrikada aynı makine kümesinde işlem görmektedirler. Diğer varsayımları akış tipi çizelgeleme varsayımları ile aynıdır. Bu durum çizelgelemeyi akış tipine göre oldukça zorlaştırmaktadır. Bu nedenle problemin çözümünde birçok araştırmacı farklı çözüm yöntemlerinden ve algoritmalarından yararlanmışlardır. Doğadaki bal arılarının nektar arama yeteneğinden ilham alınarak oluşturulan arı koloni algoritması akış tipi çizelgeleme problemlerinde yaygın olarak kullanılan güçlü bir algoritmadır. Algoritma optimal sonuca yakın sonuçlar bulmada oldukça başarılıdır.

Bu nedenle çalışmamızda DPATÇP çözümü için YAKA seçilmiştir. Amaç fonksiyonumuz tamamlanma süresinin minimizasyonudur ve DPATÇP çözümü için YAKA kullanılmıştır. YAKA'nın problem çözümündeki etkinliği Taillard'ın test problemleri ile kıyaslanmıştır. Karşılaştırılan diğer sezgisel yöntemlere göre en iyi çözüm üreten algoritma YAKA olmuştur.

Bu çalışmaların devamında ileride yapılacak olan araştırmalarda alternatif olarak işlem süreleri bulanık sayılardan seçilebilir. Daha sonra elde edilecek bulgular, bu ve benzer araştırma sonuçlarıyla kıyaslanabilecektir.

## Kaynaklar

- [1]. Ruiz, R., Q.-K. Pan, and B. Naderi, Iterated Greedy methods for the distributed permutation flowshop scheduling problem, Omega, 2019, 83: 213-222.

- [2]. Ling-Fang, C., W. Ling, and W. Jing-jing, A Two-Stage Memetic Algorithm for Distributed No-Idle Permutation Flowshop Scheduling Problem, in 2018 37th Chinese Control Conference (CCC), 2018, 2278-2283.
- [3]. Pan, J.-Q., W.-Q. Zou, and J.-H. Duan, A Discrete Artificial Bee Colony for Distributed Permutation Flowshop Scheduling Problem with Total Flow Time Minimization, in 2018 37th Chinese Control Conference (CCC), 2018: China, 8379-8383.
- [4]. Fernandez-Viagas, V., P. Perez-Gonzalez, and J.M. Framinan, The distributed permutation flow shop to minimise the total flowtime. *Computers & Industrial Engineering*, 2018, 118: 464-477.
- [5]. Wang, K., Y. Huang, and H. Qin, A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown, *Journal of the Operational Research Society*, 2017. 67(1): 68-82.
- [6]. Deng, J. and L. Wang, A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm and Evolutionary Computation*, 2017, 32: 121-131.
- [7]. Duan, W., et al., EDA based probabilistic Memetic Algorithm for distributed blocking permutation flowshop scheduling with sequence dependent setup time, in 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, IEEE. 992-999.
- [8]. Bargaoui, H., O. Belkahla Driss, and K. Ghédira, A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion. *Computers & Industrial Engineering*, 2017, 111: 239-250.
- [9]. Li, Z., et al., The distributed permutation flowshop scheduling problem with different transport timetables and loading capacities, in 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, IEEE. 2433-2437.
- [10]. Wang, J., L. Wang, and J. Shen, A hybrid discrete cuckoo search for distributed permutation flowshop scheduling problem, in 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, 2240-2246.
- [11]. Naderi, B. and R. Ruiz, A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, 2014, 239(2): 323-334.
- [12]. Lin, S.-W., K.-C. Ying, and C.-Y. Huang, Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm, *International Journal of Production Research*, 2013, 51(16): 5029-5038.
- [13]. Wang, S.-y., et al., An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, 2013, 145(1): 387-396.
- [14]. Gao, J., R. Chen, and W. Deng, An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research*, 2013. 51(3): 641-651.
- [15]. Gao, J. and R. Chen, An NEH-based heuristic algorithm for distributed permutation flowshop scheduling problems. *Scientific Research and Essays*, 2011, 6: 3094-3100.
- [16]. Gao, J. and R. Chen, A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Computational Intelligence Systems*, 2011, 4(4): 497-508.
- [17]. Liu, H. and L. Gao, A Discrete Electromagnetism-Like Mechanism Algorithm for Solving Distributed Permutation Flowshop Scheduling Problem, in 2010 International Conference on Manufacturing Automation, 2010, 156-163.
- [18]. Naderi, B. and R. Ruiz, The distributed permutation flowshop scheduling problem. *Computers & Operations Research*, 2010, 37(4): 754-768.
- [19]. Zhang, S.-j. and X.-s. Gu, An effective discrete artificial bee colony algorithm for flow shop scheduling problem with intermediate buffers, *Journal of Central South University*, 2015, 22(9): 3471-3484.

- [20]. Tasgetiren, M.F., et al., A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion, *Applied Mathematical Modelling*, 2013, 37(10-11): 6758-6779.
- [21]. Cui, Z. and X. Gu. An Improved Discrete Artificial Bee Colony Algorithm for Hybrid Flow Shop Problems. in *Intelligent Computing for Sustainable Energy and Environment*. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.
- [22]. Liu, Y.-F. and S.-Y. Liu, A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. *Applied Soft Computing*, 2013, 13(3): 1459-1463.
- [23]. Li, X. and M. Yin, A discrete artificial bee colony algorithm with composite mutation strategies for permutation flow shop scheduling problem, *Scientia Iranica*, 2012, 19(6): 1921-1935.
- [24]. Tasgetiren, M.F., et al., A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences*, 2011, 181(16): 3459-3475.
- [25]. Karaboga, D. and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 2007, 39(3): 459-471.
- [26]. Tosun, Ö., Yapay Arı Kolonisi Algoritması ve Permütasyon Akış Tipi Çizelgeleme Problemine Uygulanması, *Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü*, 2012: 1-111.
- [27]. Pan, Q.-K., et al., A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, 2011, 181(12): 2455-2468.
- [28]. Akgüngör A.P., Yılmaz Ö., Korkmaz E., Doğan E., "Meta-Sezgisel Yöntemlerle Sabit Zamanlı Sinyalize Kavşaklar için Optimum Devre Süresi Modeli", *El-Cezerî Fen ve Mühendislik Dergisi*, 2019, 6(2): 259-269.
- [29]. Arseven İ., Dağıtık Permütasyonlu Akış Tipi Çizelgeleme Problemlerinin Yapay Arı Koloni Algoritması ile Çözümü, *Yüksek Lisans, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü*, 2019.
- [30]. Arseven İ., Sarucan A., Dağıtık Permütasyonlu Akış Tipi Çizelgeleme Problemlerinin Çözümü İçin Bir Yapay Arı Koloni Algoritması, *6. Uluslararası GAP Mühendislik Kongresi, Şanlıurfa*, 2018, 172-175.
- [31]. Kalczynski, P. and J. Kamburowski, On the NEH heuristic for minimizing the makespan in permutation flow shops, *Omega*, 2007, 35(1): 53-60.