

**T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**SIRALI AKIŞ ÇİZELGELEME PROBLEMLERİNİN ARI ALGORİTMASIYLA  
ÇÖZÜMÜ**

**MUHAMMED PARLAK**

**YÜKSEK LİSANS TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
ENDÜSTRİ MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN  
DOÇ. DR. ALİ FUAT GÜNERİ**

**İSTANBUL, 2012**

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**SIRALI AKIŞ ÇİZELGELEME PROBLEMLERİNİN ARI ALGORİTMASIYLA  
ÇÖZÜMÜ**

Muhammed PARLAK tarafından hazırlanan tez çalışması 05.06.2012 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Endüstri Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Doç. Dr. Ali Fuat GÜNERİ  
Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Doç. Dr. Ali Fuat GÜNERİ  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Prof. Dr. Hüseyin BAŞLIGİL  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Doç.Dr. İbrahim EMİROĞLU  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

## ÖNSÖZ

---

Çizelgeleme araştırmacılar tarafından çokça incelenen konulardan biridir. Çünkü küreselleşen dünyada rekabet açısından üretimde işlerin veya operasyonların amaca en uygun şekilde sıralamasının belirlenmesi çok önemlidir.

Çizelgeleme konusunda birçok problem yapısı bulunmaktadır. Akış tipi çizelgeleme problemleri ise, bunların en sık karşılaşılan çeşididir. Tez kapsamında, akış tipi çizelgeleme problemlerinin bir alt çeşidi olan sıralı akış tipi çizelgeleme problemleri ele alınmıştır.

Son yıllarda araştırmalarda sık kullanılan sezgisel arama yöntemi olan arı algoritması sıralı akış tipi çizelgeleme problemlerinin çözümüne yeni bir yaklaşım getirmek amacıyla ele alınmıştır.

Bu tezimde bana yardımcı olan başta tez danışmanım Doç. Dr. Ali Fuat Güneri'ye olmak üzere Yurtiçi Yüksek Lisans Burs programı kapsamında tezimi destekleyen Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK)-Bilim İnsanı Destekleme Daire Başkanlığı'na (BİDEB) ve manevi desteklerinden dolayı aileme teşekkürü bir borç bilirim.

Mayıs, 2012

Muhammed PARLAK

## İÇİNDEKİLER

|                                                                                | Sayfa |
|--------------------------------------------------------------------------------|-------|
| ÖNSÖZ.....                                                                     | iii   |
| İÇİNDEKİLER .....                                                              | iii   |
| SİMGE LİSTESİ.....                                                             | v     |
| KISALTMA LİSTESİ.....                                                          | vi    |
| ŞEKİL LİSTESİ.....                                                             | vii   |
| ÇİZELGE LİSTESİ .....                                                          | viii  |
| ÖZET .....                                                                     | ix    |
| ABSTRACT.....                                                                  | x     |
| BÖLÜM 1                                                                        |       |
| GİRİŞ.....                                                                     | 1     |
| 1.1    Literatür Özeti .....                                                   | 1     |
| 1.2    Tezin Amacı .....                                                       | 2     |
| 1.3    Hipotez .....                                                           | 3     |
| BÖLÜM 2                                                                        |       |
| ÜRETİM PLANLAMA VE ÇİZELGELEME .....                                           | 4     |
| 2.1    Çizelgeleme .....                                                       | 4     |
| 2.2    Üretim Çizelgeleme.....                                                 | 5     |
| 2.3    Çizelgeleme Problemlerinin Gösterimi .....                              | 6     |
| 2.4    Çizelgeleme Türleri .....                                               | 10    |
| 2.4.1    İşlerin geliş şekillerine göre sıralama problemleri.....              | 11    |
| 2.4.2    İşlerin görüldüğü tezgâh sayısına göre sıralama problemleri.....      | 11    |
| 2.4.2    İş akışına göre sıralama problemleri.....                             | 12    |
| 2.4.3    Başarı ölçütü .....                                                   | 12    |
| 2.5    Çizelgeleme Problemlerinin Karmaşıklığı.....                            | 13    |
| BÖLÜM 3                                                                        |       |
| AKIŞ TİPİ İŞ ÇİZELGELEME .....                                                 | 17    |
| 3.1    Çizelgelemede Kullanılan Öncelik Kuralları ve Kabulleri.....            | 18    |
| 3.2    Akış Tipi Çizelgeleme Problemlerinin Çözümü ve Sezgisel Yöntemler ..... | 20    |
| BÖLÜM 4                                                                        |       |

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ .....                                   | 22 |
| 4.1 Sıralı Problemlerin Uygulama Temelleri .....                                 | 23 |
| 4.2 Sıralı Akış Tipi Çizelgeleme Problemi Çeşitleri .....                        | 24 |
| 4.4.1 Genel durum problemi .....                                                 | 25 |
| 4.4.2 Özel Durum Problemi .....                                                  | 27 |
| BÖLÜM 5                                                                          |    |
| ARI ALGORİTMASI .....                                                            | 29 |
| 5.1 Giriş .....                                                                  | 29 |
| 5.1 Doğal Ortamında Arılar .....                                                 | 30 |
| 5.3 Arıların besin bulma davranışları .....                                      | 32 |
| 5.4 Arı Algoritması .....                                                        | 39 |
| 5.5 Bir Arı Algoritması Uygulaması .....                                         | 41 |
| BÖLÜM 6                                                                          |    |
| SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNDE ARI ALGORİTMASININ UYGULANMASI ..... | 48 |
| 6.1 Uygulamanın Varsayımları .....                                               | 48 |
| 6.2 Uygulamanın Yapısı ve Oluşturulması .....                                    | 48 |
| 6.3 Uygulanan Arı Algoritmasının Çalışma Mekanizması .....                       | 50 |
| 6.4 Arı Algoritması Parametreleri .....                                          | 53 |
| 6.4 Arı Algoritmasıyla Örnek Bir Uygulama .....                                  | 54 |
| 6.6 Uygulama Problemler Sonuçları .....                                          | 59 |
| 6.7 Uygulama Problemlerinin Değerlendirilmesi.....                               | 63 |
| BÖLÜM 7                                                                          |    |
| SONUÇ VE ÖNERİLER .....                                                          | 66 |
| KAYNAKLAR.....                                                                   | 69 |
| EK-A                                                                             |    |
| BAGİ İÇİN İŞ SIRALAMALARI .....                                                  | 74 |
| EK-B                                                                             |    |
| BASG İÇİN İŞ SIRALAMALARI .....                                                  | 75 |
| ÖZGEÇMİŞ.....                                                                    | 76 |

## SİMGE LİSTESİ

---

|                  |                                                           |
|------------------|-----------------------------------------------------------|
| n                | İş sayısı                                                 |
| m                | Makine sayısı                                             |
| C <sub>max</sub> | İşlerin maksimum tamamlanma zamanı                        |
| n                | Kâşif (çözüm arayan)arı sayısı                            |
| m                | n bölgeden seçilecek bölge sayısı                         |
| e                | Araştırma bölgesi içerisindeki en iyi elit arı sayısı     |
| m-e              | Diğer araştırılacak seçilmiş bölge sayısı                 |
| nep              | Elit arı etrafına gönderilecek arı sayısı                 |
| nsp              | Diğer bölgesi arılarının etrafına gönderilecek arı sayısı |

## KISALTMA LİSTESİ

---

|        |                                                                        |
|--------|------------------------------------------------------------------------|
| FIFO   | FirstIn First Out (İlk Giren İlk Çıkar)                                |
| GA     | GeneticAlgorithm (Genetik Algoritma)                                   |
| LPT    | LongestProcessing Time (Uzun İşlem Süreli)                             |
| NP     | NonPolynomialOptimization (Polynomial Olmayan Optimizasyon)            |
| TA     | Tavlama Benzetimi                                                      |
| SPD    | Smith, Panwalkar, Dudek                                                |
| SPT    | ShortestProcessing Time (Kısa İşlem Süreli)                            |
| BA     | BeeAlgorithm (Arı Algoritması)                                         |
| BAIG   | BeeAlgorithmInsertionGreedy (Arı Algoritmasında yerleştirme Komşuluğu) |
| BASG   | BeeAlgorithmSweepGreedy (Arı Algoritmasında Yer Değiştirme Komşuluğu)  |
| P      | Polynomial Time(Polynomial Optimizasyon Süresi)                        |
| EYÇYDM | En iyi Yerel Çiftli Yer Değiştirme Mekanizması                         |
| EYYM   | En iyi Yerel Yerleştirme Mekanizması                                   |

## ŞEKİL LİSTESİ

|                                                                                         | Sayfa |
|-----------------------------------------------------------------------------------------|-------|
| Şekil 2. 1 Üretim sisteminde bilgi akışı .....                                          | 7     |
| Şekil 2.2 Örnek problemlerin çözüm açısından komplekslikleri.....                       | 16    |
| Şekil 3. 1 Saf akış tipi yerleşim.....                                                  | 17    |
| Şekil 3. 2 Genel akış tipi yerleşim .....                                               | 18    |
| Şekil 4. 1 Örnek SPD algoritmasında birinci adım .....                                  | 25    |
| Şekil 4. 2 Örnek SPD algoritmasında ikinci adım .....                                   | 25    |
| Şekil 4. 3 Örnek SPD algoritmasında üçüncü adım.....                                    | 26    |
| Şekil 4. 4 Örnek SPD algoritmasının çözümü.....                                         | 27    |
| Şekil 5. 1 Optimizasyon problemlerinin sınıflandırılması.....                           | 30    |
| Şekil 5. 2 Arılarda görev dağılımı .....                                                | 32    |
| Şekil 5. 3 Kovandaki dans bölgelerinin temsili gösterimi .....                          | 34    |
| Şekil 5. 4 Bal arılarının yem arama davranışları .....                                  | 35    |
| Şekil 5. 5 Arıların kovanda yaptığı 8 dansı .....                                       | 36    |
| Şekil 5. 6 Arıların yiyecek kaynağının yönünü göstermeleri.....                         | 37    |
| Şekil 5. 7 Arı algoritmasının uygulanacağı fonksiyon.....                               | 43    |
| Şekil 5. 8 Rassal olarak birey üretimi ve uygunluk değerlerinin hesaplanması .....      | 43    |
| Şekil 5. 9 Araştırma bölgesinin belirlenmesi ve elit arıların seçimi .....              | 44    |
| Şekil 5. 10 Komşuluk arama çevresinin belirlenmesi.....                                 | 44    |
| Şekil 5. 11 Araştırma bölgesine izci arıların gönderilmesi.....                         | 45    |
| Şekil 5. 12 Her bir araştırma bölgesindeki en uygun arının seçimi .....                 | 45    |
| Şekil 5. 13 Kalan arıların rassal atanması ve uygunluk değerlerinin hesaplanması.....   | 46    |
| Şekil 5. 14 Yeni popülasyonda araştırma bölgesinin seçimi ve elit arıların seçimi ..... | 46    |
| Şekil 5. 15 Fonksiyonun optimum araştırma bölgesi.....                                  | 47    |



## ÇİZELGE LİSTESİ

|              | Sayfa                                                                 |
|--------------|-----------------------------------------------------------------------|
| Çizelge 4. 1 | 4-iş 4-makine problemi için sıralı matris.....23                      |
| Çizelge 4. 2 | Sıralı problem iş zamanları .....28                                   |
| Çizelge 6. 1 | En iyi yerel çift değiştirme mekanizması.....51                       |
| Çizelge 6. 2 | En iyi yerleştirme mekanizması.....28                                 |
| Çizelge 6. 3 | Uygulanan arı algoritmasının çalışma mekanizması.....53               |
| Çizelge 6. 4 | 7-işlik problemler için belirlenen parametreler.....54                |
| Çizelge 6. 5 | 10-işlik problemler için belirlenen parametreler.....54               |
| Çizelge 6. 6 | 5-iş 5-makine örnek sorusu ..... 55                                   |
| Çizelge 6. 7 | Rassal olarak seçilen iş sıralamaları ve tamamlanma süreleri..... 55  |
| Çizelge 6. 8 | Araştırma bölgelerindeki iş sıralamaları..... 56                      |
| Çizelge 6. 9 | 1.elit işin komşuluk çevresi ve uygunluk değerleri.....56             |
| Çizelge 6.10 | 2.elit işin komşuluk çevresi ve uygunluk değerleri.....57             |
| Çizelge 6.11 | (m-e) komşuluk çevresi ve uygunluk değerleri..... 57                  |
| Çizelge 6.12 | Yeni popülasyondaki iş sıralamaları ve uygunluk değerleri ..... 58    |
| Çizelge 6.13 | Yeni popülasyondaki araştırma bölgeleri..... 58                       |
| Çizelge 6.14 | Genel durum problemlerinin uygulama sonuçları..... 60                 |
| Çizelge 6.15 | Genel problemlerde algoritmaların performans karşılaştırılması.....61 |
| Çizelge 6.16 | Özel durum problemlerinin uygulama sonuçları.....62                   |
| Çizelge 6.17 | Özel problemlerde algoritmaların performans karşılaştırılması .....62 |

## SIRALI AKIŞ ÇİZELGELEME PROBLEMLERİNİN ARI ALGORİTMASIYLA ÇÖZÜMÜ

Muhammed PARLAK

EndüstriAnabilim Dalı

YüksekLisansTezi

TezDanışmanı: Doç. Dr. Ali Fuat GÜNERİ

Çizelgeleme araştırmacılar tarafından çokça incelenen konulardan biridir. Çünkü küreselleşen dünyada rekabet açısından çizelgeleme önemlidir. Tez kapsamında akış tipi çizelgeleme problemlerinin bir alt gelişimi olan sıralı akış tipi çizelgeleme problemleri ele alınmıştır. Bu problem tipini çözmek için arı algoritması geliştirilmeye çalışılmıştır.

Çalışmanın ilk bölümlerinde, çizelgeleme problemleri, akış tipi çizelgeleme problemlerinin yapısından ve arı algoritmasından bahsedilmiştir. Arı algoritmasının sıralı performansını değerlendirmek için yapılan uygulama iki kısım olarak ele alınmıştır.

İlk kısımda, genel durum olarak adlandırılan, işlem zamanları matrisinde en büyük işlem zamanlarının ara makinelerden birinde oluşması durum ele alınmıştır. İkinci kısım problemler ise en büyük işlem zamanlarının ilk ya da son makinede oluşması durumu olan özel durum problemleridir. Yapılan uygulamalarda arı algoritması, Smith Panwalkar Dudek algoritmasından daha iyi sonuçlar vermiştir.

**Anahtar Kelimeler:**Sıralı Akış Tipi Çizelgeleme Problemleri, Arı Algoritması.

## ABSTRACT

---

### ORDERED FLOWSHOP SCHEDULE PROBLEMS SOLVING BY BEE ALGORITHM

Muhammed PARLAK

Department of Industrial Engineering

MSc. Thesis

Advisor: Assoc. Prof. Dr. Ali Fuat GÜNERİ

Scheduling is one of the research topics that scholars have studied widely. Because in the global world, scheduling is very important. In this thesis, the problem of scheduling jobs in a flow-shop environment is considered. Bee algorithm is applied in order to find a new approach to the problem.

In the first parts of the study, the structure of flow-shop problems, ordered flowshop problems and bee algorithm. The application, which is made to determine the performance of bee algorithm, on ordered flowshop problems are reconsidered as two different parts as the nature of the ordered type of flow-shop problems.

In the first part, the general case, which is the maximum processing time for every job occurred in intermediate machines, are reconsidered. In the second kind of problem, the maximum processing time for every job occurred on the first or last machine, which was the special case, are reconsidered. At the applications, the bee algorithm gave better results than Smith, Panwalker, Dudek algorithm.

**Keywords:** Ordered Flow-Shop Problems, Bee Algorithm

---

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

### GİRİŞ

Mevcut bulunan işlerin bir grup kaynağa atanması kısaca olarak tanımlayacağımız çizelgeleme aynı sektörde faaliyet gösteren işletmeler için bulundukları rekabet ortamında bir adım öne geçebilmelerini sağlayacak önemli bir kavramdır. Çizelgeleme problemleri için çok sayıda yöntem geliştirilmiştir. Doğru yöntemi kullanmak yapılmış veya yapılmakta olan işin en iyi çözümünü ortaya koymak için kullanıldığından globalleşen dünya pazarı ve buna bağlı olarak artan rekabet ortamında işletmelere problemlerini çözmede yardımcı olacaktır.

#### 1.1 Literatür Özeti

Çizelgeleme problemleri konusunda birçok çalışma yapılmıştır. Çizelgeleme problemlerinden akış tipi çizelgeleme hakkında yapılan çalışmalar hayli fazladır. Sıralı akış tipi de yapılan çalışmalardan biridir. Bu problem tipini Smith ve arkadaşları tarafından tanımlanmıştır [1], [2]. Bu problem tipi hakkında detaylı bilgi sıralı akış tipi iş çizelgeleme kısmında verilmiştir. Smith ve arkadaşları yaptığı çalışmada genel ve özel durum hallerini tanımlamış ve çözüm yöntemi önermişlerdir. Bu çalışmalarda  $C_{max}$  minimize edilmeye çalışılmıştır. Smith ve Khan birlikte yaptıkları çalışmada bu problem tipi için ortalama akış süresini minimize etmek üzerine çalışmalar yapmıştır [3].

Arı algoritması son yıllarda popüler olmaya başlanmış bir yöntem olmuştur. Arılardan esinlenerek üretilen birçok yöntem bulunmakta çalışma mekanizmaları birbirine benzemektedir. Örneğin, arı algoritması, yapay arı algoritması, arı kolonisi ve kesikli yapay arı algoritması gibi çeşitli isimleri bulunmaktadır. Arı algoritması iş çizelgeleme problemlerinde, proje çizelgelemede, yapay sinir ağları uygulamalarında, gezgin satıcı problemleri gibi birçok alanda kullanılmıştır [4].

Chong ve arkadaşları yaptıkları çalışmada arı kolonisi algoritmasını kullanarak atölye tipi iş çizelgeleme konusunu ele almışlardır. Arı kolonisinin performansı karınca algoritmasına göre iyi sonuçlar elde etmişlerdir. Fakat tabu arama yönteminin sonuçları arı algoritmasına göre daha iyi çıkmıştır [5].

Pham ve arkadaşları arı algoritmasını tek makineli iş problemlerinde erken geç teslim tarihini minimize etmek için kullanmışlardır. Kullandıkları arı algoritmasının literatürde var olan dataları kullanarak diğer çözüm yöntemleriyle kıyaslamışlardır. Elde edilen bulgulara göre arı algoritması hem iyi sonuçlar vermiştir, hem de daha güvenilir sonuçlar vermiştir [6].

Pham ve Koç arı algoritmasını akış tipi iş çizelgeleme problemleri için kullanmıştır. Araştırmalarında arı algoritmasının diğer tekniklere göre daha güvenilir ve daha üstün sonuçlar elde etmişlerdir [7].

Wang ve Pan esnek atölye tipi çizelgeleme problemleri için hibrit akış yapay arı kolonisi önermişlerdir. Çok bilinen kıyaslama problemlerine uygulayarak önerilen algoritmanın etkinliğini ve performansını ölçmüşlerdir [8].

Banharnsakun ve arkadaşları atölye tipi iş çizelgeleme problemlerine best-so-far yapay arı algoritmasını incelemişlerdir. Komşuluk üretme mekanizmasında genetik algoritmadan faydalanmışlardır. Çalışmalarında literatürde bulunan 62 tane uygulama problemleriyle kıyaslayarak çözüm performansını ölçmüşlerdir [9].

## **1.2 Tezin Amacı**

Günümüzde küreselleşen dünyada işletmelerin rekabet ortamında ayakta kalmaları için her alanda en uygun faaliyette bulunmaları gerekir. Bu faaliyetlerin içinde var olan işlerin mevcut kaynaklara doğru atanması gerekir. Bu yüzden çizelgeleme gittikçe önem kazanmaktadır.

Çizelgeleme problemleri araştırmacılar tarafından çokça incelenmiştir. Bu problem tipleri genellikle NP-zor kısmında olduklarından dolayı var olan klasik yöntemlerin yanında sezgisel yöntemler ortaya konulmuştur. Sezgisel yöntemler optimum sonuca yakın değerler ürettiğinden kesin sonucu önermediğinden performansları çok önemlidir. Son yıllarda bu yöntem türlerinden artış ortaya çıkmıştır.

Arı algoritması arıların doğal ortamdaki davranışlarından etkilenecek şekilde ortaya çıkmıştır. İncelenen konularda uygulama performansı çok iyi çıkmıştır. Gittikçe gelişen versiyonları ortaya çıkmıştır.

Bu tez çalışmasının amacı, literatürde var olan sıralı akış tipi iş çizelgeleme probleminin son yıllarda sıkça kullanılan arı algoritmasıyla çözmeyi sağlamaktır. Bu problem tipi için SPD algoritması önerilmiştir. Bu çalışmada arı algoritması kullanılacak ve genetik algoritmasıyla kıyaslanıp performans kalitesi ve etkinliği incelenecektir.

### **1.3 Hipotez**

Bu tez çalışmasında, akış tipi iş çizelgeleme problemlerinin bir alt sınıfı olan sıralı akış tipi iş çizelgeleme probleminin arı algoritması kullanılacaktır. Bununla birlikte sıralı akış çizelgeleme problemi için önerilen SPD algoritması ve daha önce yapılan bir çalışmada bu problem tipi için önerilen genetik algoritmayla önerilen arı algoritması kıyaslanarak çözüm kalitesi ve performansının iyileştirilebilmesi hedeflenecektir.

Bu amaçla başladığım tez çalışmasında ilk olarak (ikinci bölümde) çizelgelemenin ne olduğu, üretimle olan ilişkisi, çizelgeleme problemlerinin yapısı ve komplekslikleri ele alınmıştır. Üçüncü bölümde, akış tipi iş çizelgeleme probleminin yapısı, nasıl olduğu ve literatürde önerilen sezgisel yöntemlerden bahsedilmiştir. Dördüncü bölümde uygulandığı sıralı akış tipi çizelgelemelerinin özellikleri, yapısı, önerilen SPD algoritmasının metodu ve uygulaması anlatılmıştır. Beşinci bölümde, arıların doğal ortamdaki yaşayışları, arı algoritmasının yapısı, nasıl uygulandığı ve örnek bir uygulamadan bahsedilmiştir. Altıncı bölümde örnek bir problem setine arı algoritması uygulanıp SPD algoritması, genetik algoritmayla ve tavlama benzetimi yöntemiyle kıyaslanmıştır. Sonuncu bölümde ise elde edilen bulgulardan ve önerilerden bahsedilmiştir.

### ÜRETİM PLANLAMA VE ÇİZELGELEME

#### 2.1 Çizelgeleme

Çizelgeleme, belirli bir takım görevleri yerine getirmek üzere kaynakların zaman içinde tahsisi olarak tanımlanabilir. Bu tanımda görevler, içinde bulunulan ortama bağlı olarak değişik şekillerde adlandırılabilir.

Çizelgeleme, kıt kaynakların belirli bir zaman boyunca işlere tahsis edilmesiyle ilgilidir. Bu süreç bir veya daha fazla hedefin optimizasyonunu amaçlayan bir karar alma sürecidir (Pinedo [10]). Çizelgeleme, en genel anlamda, bir üretim sisteminde belli bir dönemde (genellikle günlük veya haftalık) yapılacak işlerin sıralarının ve zamanlarının belirlenmesi ve buna uygun olarak gerekli kaynakların (makine, atölye v.b.) tahsis edilmesi şeklinde tanımlanmaktadır (Mutlu [11]).

Çizelgelemede üretim, kaynak ve zaman üç önemli unsurdur. Bu unsurlar dikkate alındığında belirli bir takım işleri yapmak için hangi kaynakların, ne zaman ve nasıl kullanılacaklarının tespit edilmesine çizelgeleme denir (Güldalı [12]). Genel olarak, belirli ürün veya hizmetleri meydana getirmek için gerekli olan faaliyetlerdir. Çizelgeleme, belirli bir takım işleri yapmak için hangi kaynakların, ne zaman ve nasıl kullanılacaklarının tespit edilmesidir.

Örnek olarak, bir okuldaki derslerin çizelgelenmesini ele alınsın. Buradaki iş, öğrencilerin farklı derslere atanmasıdır. Burada bir sınıfta oturabilecek öğrenci sayısı, okuldaki sınıf sayısı, ders verebilecek öğretmen sayısı, okuldaki toplam çalışma saati, ders aralarındaki teneffüs için gereken süre v.b. gibi birçok kısıt söz konusudur.

Görüldüğü gibi çok basit gibi görünen bir okuldaki öğrencilere göre ders çizelgesi hazırlanması işlemi bile, oldukça karmaşık yapıya sahip bir karar verme problemine dönüşebilmektedir. Ayrıca öğrenciler, öğretmenler, ders saatleri, derslik sayısı gibi birçok kaynak kısıtının da göz önüne alınması gerekmektedir (Aladağ [13]).

Çizelgeleme, yaşarken sık sık karşılaştığımız bir olgudur. Çizelgeleme çalışmalarına en sık bir üretim tesisindeki işlerin çizelgelenmesinde, okullarda ders planlarının yapılmasında, havaalanlarındaki uçakların iniş kalkışlarında ve otobüs seferlerinin belirlenmesinde rastlamaktayız.

## **2.2 Üretim Çizelgeleme**

Üretim aşamasında sürekli olarak aşırı stok bulunması, bazı makinelerin sürekli boş kalması, geciken işlerin çokluğu, içinde çok sayıdaki yarı mamul yığınları veya bir kısım tezgâhların çalışırken diğerlerinin boş durması gibi durumlar bir çizelgeleme probleminin göstergesidir. Bununla birlikte üretim sistemi incelendiğinde görülen fazla mesai, geciken işler, düşük tezgâh kullanımı ya da işgücü kullanımı gibi görülen sayısal veriler çizelgeleme probleminin göstergesidir. İşte iyi bir çizelgeme yapılmasıyla bu gibi sorunlar çözülebilir [12],[11].

Bir üretim tesisindeki çizelgeleme faaliyetleri, gelen işlerin, o işlerin yapılması için gerekli kaynaklara (makine, zaman, işgücü v.b.) atanması süreci ile ilgilidir (Aladağ [13]).

Çizelgeleme, üretim planlama sürecinin önemli bir parçasıdır. Üretim sistemleri bilgi akışı sürecinde çizelgeleme fonksiyonun yeri şekil 2.1’de gösterilmiştir. Çizelgeleme işlevi, şekil 2.1’de görüldüğü gibi, sadece atölyeden değil, aynı zamanda orta ve uzun dönemli planlamadan sorumlu üretim planlama işlevinden de etkilenir. Üretim planlama işlevi, kaynak ihtiyaçları, talep tahminleri ve stok seviyelerini göz önünde bulundurarak uzun dönemli kaynak tahsisinin yanı sıra firmanın ürün karışımını da en iyilemeyi amaçlar. Bu yüksek planlama seviyesindeki kararlar çizelgeleme işlevini doğrudan etkileyecektir. Çizelgeleme, en yakın zamandaki üretim amaçlarına ulaşabilmek için uygun makine, iş gücü, malzeme vs. koşulları göz önüne alarak ayrıntılı bir yol ortaya koyar. Aslında üretim çizelgelemenin üretim planlamadan farkı da budur.



Daha ayrıntılı ve kısa dönemli olmasıdır. Çizelgeleme, aşağıdaki sorunlarla ilgilenir (Brucker ve Knust [14]) :

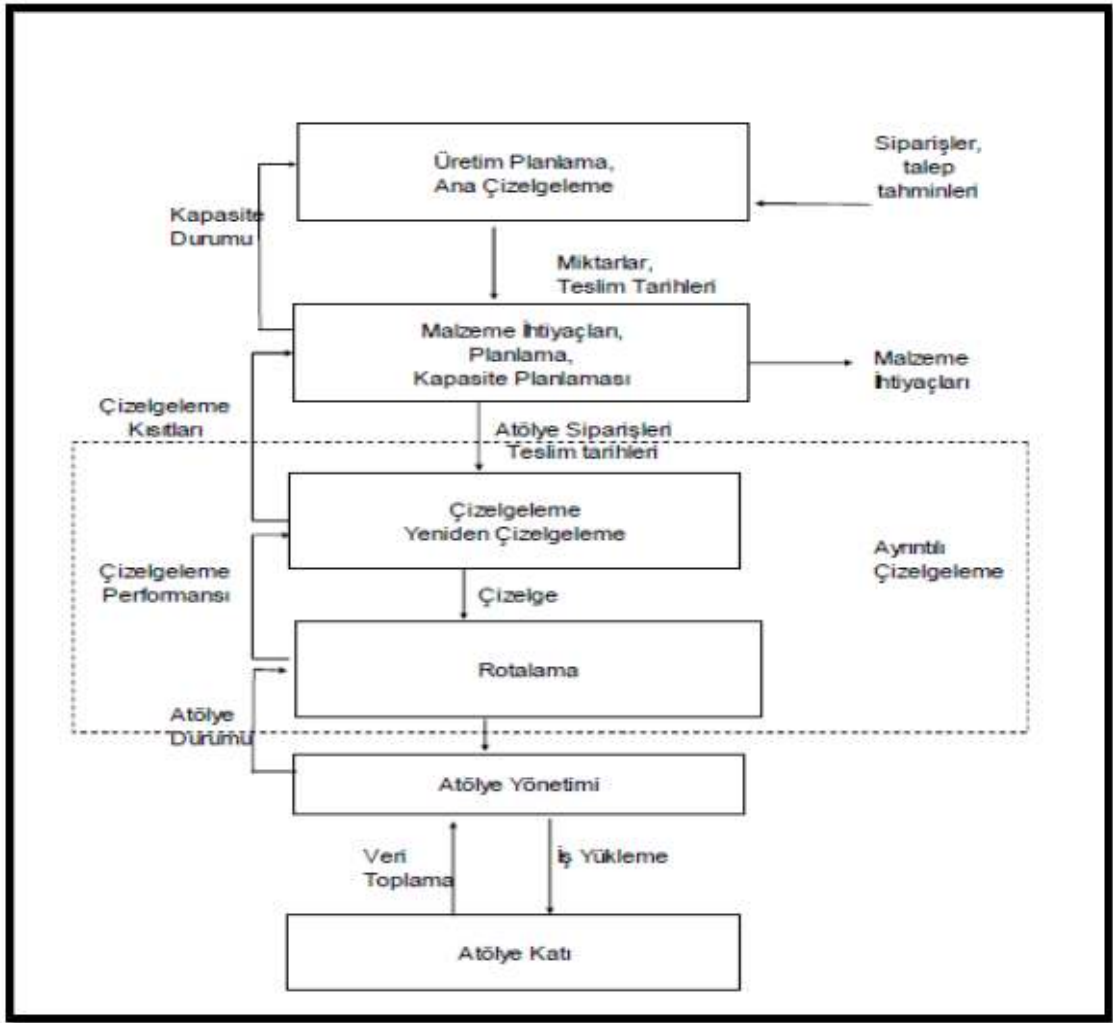
- Hangi iş merkezi hangi işi yapacak?
- Bir iş ne zaman başlayacak ne zaman bitecek?
- İş hangi ekipmanla, kim tarafından yapılacak?
- İşlerin sıralaması ne olacak?

Çizelgeleme problemlerinde iki farklı türden uygunluk kısıtlarıyla karşılaşılmaktadır. Bunlardan ilki makine kapasitelerindeki sınırlamalar, ikincisi ise bazı işlerin işlem sırası üzerindeki teknolojik sınırlamalardır. Çizelgeleme probleminin çözümü, iki gruba ayrılabilen bu kısıtların her hangi bir uygun çözümünü ifade eder. Dolayısıyla çizelgeleme problemini çözme işlemi aşağıdaki sorulara cevap arama işlemini kapsar. Diğer bir deyişle, çizelgeleme problemi atama ve sıralama işlemlerinden oluşur (Kellegöz [15]).

Üretim çizelgeleme, bir veya birkaç performans ölçütünü en iyileyecek şekilde üretim kaynaklarının zaman bazında çeşitli görevlere atanmasını kapsar. Üretim sistemine gelen siparişler öncelikle, teslim tarihlerini de içeren işlere çevrilmelidir. Genellikle bu işlere ait operasyonların belirli bir sırada atölyedeki bazı makineler üzerinde yapılması gerekir. Buna karşın bu operasyonların, örneğin makinelerin meşgul olması gibi nedenlerle bazı durumlarda bekletilmesi gerekebilir. Bunun yanı sıra makine arızaları, işlem zamanlarının beklenenden daha uzun sürmesi gibi beklenmeyen atölye olaylarının da göz önünde bulundurulması gerekir. Yapılması gereken görevlere ilişkin detaylı bir çizelgenin hazırlanması işlemlerin kontrolü ve etkinliğinin sürdürülmesinde faydalı olacaktır.

### 2.3 Çizelgeleme Problemlerinin Gösterimi

Bütün çizelgeleme problemlerinde işler ve makineler sonludur. Makine sayısı  $m$  ve iş sayısı ise  $n$  ile gösterilir. Genellikle  $j$  indisi bir işi,  $i$  indisi ise bir makineyi ifade eder. Eğer bir iş çeşitli işlem adımlarından oluşuyorsa,  $j$  işinin  $i$  makinesinde gördüğü işlemi ifade etmek için  $(i, j)$  ikilisi kullanılır.  $j$  işi ile ilgili diğer parametreler ise şu şekildedir (Pinedo[10]):



Şekil 2.1 Üretim sisteminde bilgi akışı (Pinedo [10])

İşlem Süresi (Processing Time ( $p_{ij}$ )):  $p_{ij}$ ,  $j$  işinin  $i$  makinesindeki işlem süresini belirtir. Eğer işlem süresi makineye göre değişmiyorsa veya  $j$  işi sadece belirli bir makinede işlenebiliyorsa  $i$  indisi atılır.

Salıverme Zamanı (Release Date ( $r_j$ )):  $j$  işinin işlemeye hazır olduğu zamanı belirtir. İşin sisteme geldiği ve en erken işlemeye başlanabileceği zamandır.

Teslim Zamanı (Due Date ( $d_j$ )): İşin tamamlanıp müşteriye teslim edilmesi gereken zamanı belirtir.

Ağırlık (Weight ( $w_j$ )): Genellikle işin öncelik faktörünü belirtmek için kullanılır. İşin diğer işlere göre bağıl önemini belirtir. Bir çizelgeleme problemi genellikle  $\alpha / \beta / \gamma$  üçlü notasyonu ile gösterilir (Pinedo[10]): Bu gösterimde  $\alpha$  alanı makine ortamını belirtir ve yalnız bir değer alabilir.  $\beta$  alanı çizelgeleme kısıtları ve işleme özellikleri ile ilgili

genelbilgilerin verildiği alandır. Bu alanda hiç girdi olmayabilir veya birçok girdi olabilir.  $\gamma$  alanı ise çizelgeleme sonucunda en iyilenecek performans kriterini belirtir. Genellikle tek girdi kullanılır (Pinedo[10]).

Tek Makine (1): Ortamda bir adet makine vardır. Tek makine diğer tüm karmaşık sistemlerin özel bir durumudur.

Özdeş Paralel Makineler ( $P_m$ ): Ortamda birbirine eşdeğer  $m$  adet paralel makine vardır ve işler bu makinelerin herhangi birinde işlenebilir. Eğer  $j$  işinin bazı makinelerde işlenmesine izin verilmiyor sadece  $m$ 'nin  $M_j$  alt kümesine ait makinelerde işlenmesine izin veriliyor ise  $M_j$  de  $\beta$  alanında gösterilir.

Farklı Hızdaki Paralel Makineler ( $Q_m$ ): Farklı hızda  $m$  adet paralel makine söz konusudur.  $i$  makinesinin hızı  $v_i$  ile gösterilir.  $P_{ij}$  zamanında  $j$  işi  $i$  makinesine gelir ve bu makinede  $p_j/v_i$  kadar kalır.

Paralel Farklı Makineler ( $R_m$ ): Burada işlem hızları kendilerine gelen işe göre değişen  $m$  adet paralel makine vardır.  $i$  makinesi  $j$  işin  $v_{ij}$  hızında işleyebilir ve  $j$  işinin  $i$  makinesinde işlem süresi  $p_{ij}/v_{ij}$  kadar olur. Eğer makinenin hızı gelen işlerden bağımsız ise bir önceki farklı hızdaki paralel makine ortamı ile aynı duruma gelir.

Akış Tipi ( $F_m$ ): Birbirine seri  $m$  adet makine vardır. Bütün işler bu makinelerin her birine uğramak zorunda ve aynı işlem rotasını takip etmek zorundadır.

Esnek Akış Tipi ( $FF_c$ ): Akış tipi ve paralel makine ortamının birlikte olduğu durumdur. Seri olarak dizilmiş  $c$  adet aşama vardır. Her aşamada özdeş paralel makineler mevcuttur. Her iş aşama-1, aşama-2, ...aşama- $c$  olmak üzere bütün aşamalardan geçmek zorundadır.

Atölye Tipi ( $J_m$ ): Birbirine seri  $m$  adet makine vardır ve sistemdeki her işin kendine özel rotası vardır. Bütün işler bütün makinelere aynı sırada uğramak zorunda değildir. Eğer işlerin daha önce uğradıkları makineye tekrar uğramaları gerekiyorsa  $\beta$  alanında tekrar faktörü de gösterilmelidir.

Esnek Atölye Tipi ( $FJ_c$ ): Atölye tipinin ve paralel makinelerin beraber olduğu durumun genel ifadesidir. Esnek atölyede  $c$  adet iş merkezi mevcuttur ve her iş merkezinde

paralel makineler vardır. Eğer bir iş daha önce uğradığı iş merkezine daha sonra tekrar uğramak zorundaysa tekrar faktörü  $\beta$  alanında gösterilmelidir.

Açık Atölye (Om): m adet makine vardır ve bütün işler her bir makineye uğramak zorundadır fakat işlerin makinelere uğramaları ile ilgili herhangi bir rotalama kısıtı yoktur. İşler herhangi bir sırada makinelere uğrayabilir.

$\beta$  alanında görünebilecek kısıtlar ve girdiler şöyledir:

Salıverme Zamanı ( $r_j$ ): Bu sembol  $\beta$  alanında belirtilmişse, j işi kendi salıverme zamanından önce başlayamaz demektir. Eğer belirtilmemişse j işi herhangi bir zamanda başlayabilir.

Sıra Bağımlı Hazırlık Süresi ( $s_{jk}$ ): Eğer bu sembol varsa, j işi ile k işi arasındaki geçişlerde sıra bağımlı hazırlık süresi var demektir.  $s_{0k}$ , k işi ilk iş ise onun hazırlık zamanını,  $s_{j0}$  j işi son iş ise j işinden sonraki temizlik zamanını belirtir. Eğer hazırlık zamanları makine türüne göre değişiyorsa bu alanda  $s_{ijk}$  sembolü görünür. Eğer herhangi bir  $s_{jk}$  sembolü bulunmuyorsa sıra bağımlı hazırlık zamanları 0 alınır.

Kesinti (prmp): Eğer makineye bağlanan iş işlem süresi sona erene kadar makinede kalmak zorunda değilse, işleme sırasında durdurulup yerine başka bir iş bağlanabiliyorsa  $\beta$  alanında prmpsembolü gözüktür.

Öncelik Kısıtları (prec): Eğer işler arasında, bir iş başka bir iş tamamlanmadan başlayamaz, şeklinde öncelik kısıtı varsa bu sembol belirtilir.

Makine Kesintileri (brkdwn): Makinelerin sürekli olarak çalışmadığını, kesintiye uğrayabildiğini belirtir.

Permütasyon (prmu): Akış tipi atölyede her makine önüne gelen işleri ilk giren ilk çıkar kuralına göre işliyorsa bu sembol belirtilir.

Bekleme Yok (nwt): açık atölyede işlerin ardışık iki makine arasında beklemesine izin verilmiyorsa bu sembol belirtilir.

Bloklama: Açık atölyede ardışık iki makine arasında yer alabilecek iş yığını sınırlı ise ve iş yığını kapasitesi dolmuşsa, önceki makinede tamamlanan işlerin sonraki makineye salıverilmesine izin verilmez, bu duruma bloklama denir.

Tekrar işleme (recrc): Atölye tipi veya esnek atölye tipi çizelgelemede, bir iş daha önce uğradığı makineye tekrar uğruyorsa, bu sembol belirtilir.

Çizelgeleme probleminde en küçüklenecek performans ölçütü her zaman işlerin tamamlanma zamanlarına bağlı bir fonksiyonu ifade eder. J işinin i makinesindeki tamamlanma zamanı  $C_{ij}$  ve sistemde geçirdiği zaman  $C_j$  notasyonu ile gösterilmekte olup, performans ölçütü aynı zamanda teslim zamanı  $d_j$ 'nin de bir fonksiyonu olabilir. j işinin gecikmesi  $L_j = C_j - d_j$ , j işinin geç bitmesi  $T_j = \max\{L_j, 0\}$  ve j işinin gecikme durum

$U_j = \begin{cases} 1, & C_j < d_j \\ 0, & d_j \end{cases}$  fonksiyonlarıyla hesaplanır. Performans ölçütünü ifade eden  $\gamma$  parametresinin bu fonksiyonlara bağlı alabileceği bazı parametreler aşağıdaki gibidir.

- Maksimum tamamlanma zamanı ( $C_{\max}$ ): Formülasyonu  $C_{\max} = \max(C_1, C_n)$  olup son işin sistemi terk etme zamanını ifade eder. En küçüklenmesi genellikle yüksek makine verimliliğini sağlar.
- Maksimum gecikme ( $L_{\max}$ ): Formülasyonu  $L_{\max} = \max(L_1, L_n)$  olup teslim zamanından sapmaların en büyüğünü ifade eder.
- Akış zamanı ( $\sum C_j$ ).
- Ağırlıklı akış zamanı ( $\sum w_j C_j$ ): Stok taşıma maliyeti gibi çizelgenin neden olduğu maliyetlerin bir göstergesidir.
- Toplam geç bitirme ( $\sum T_j$ ).
- Ağırlıklı geç bitirme ( $\sum w_j T_j$ ).
- Toplam geciken iş ( $\sum U_j$ ).
- Ağırlıklı geciken iş ( $\sum U_j T_j$ ).

## 2.4 Çizelgeleme Türleri

Çizelgeleme problemlerinin çözülmesi için çok sayıda teknik ve yaklaşım geliştirilmiştir. Çözüm için gerekli olan metodun seçimi, çizelgeleme probleminin karışıklığına, modelin yapısına ve amaca göre değişmektedir. Bu yüzden çizelgeleme problemlerinin sınıflandırılması çözüm kolaylığı için gereklidir. Çizelgeleme problemlerine şu ana başlıklar altında sınıflandırılabilir:

- İşlerin geliş şekillerine göre sıralama problemleri
- İşlerin görüldüğü tezgâh sayısına göre sıralama problemleri
- İş akışına göre sıralama problemleri
- Performans ölçütüne göre sıralama problemi

#### 2.4.1 İşlerin geliş şekillerine göre sıralama problemleri

**a)Dinamik problemler:** Bu tip problemlerde işe iş listesi sürekli ve rastgele değişmektedir. İşler düzensiz aralıklarla atölyeye gelir ve herhangi bir zamanda gelebilecek işin özellikleri, sıralamanı sürekli değişmesini gerektirebilir. Bu çizelgeleme problemlerinde genel amaç, tezgâh boş bekleme süresini veya bekleyen iş sayısını en azaltmak, istenen teslim tarihlerini gerçekleştirmek vb. olabilir. Herhangi bir optimizasyon yöntemi yoktur, simülasyon teknikleri ya da öncelik kurallarından en iyi sonucu veren kullanılabilir.

**b)Statik problemler:** Bu tip problemlerde, belirli bir dönem için iş listesi bilinmektedir ve bu liste genellikle değişmez. İşler boş olan bir atölyeye hemen işlenmek üzere düzenli olarak gelirler. Genel amaç, işlerin sırasını bulmaktır. Bu tür problemler için dal sınır algoritması ya da lineer programlar uygun olabilir.

#### 2.4.2 İşlerin görüldüğü tezgâh sayısına göre sıralama problemleri

Tesisteki makine sayısının bir tane veya birden fazla olmasına göre iki tür yapıdan söz edilebilmektedir:

**a)Tek Makineli Atölye:** Tüm makine ortamlarının en basit halidir ve diğer daha karmaşık makine ortamlarının özel bir durumudur.

**b)Çok Makineli Atölye:** Atölyede birden fazla makinenin bulunduğu atölyedir. Bu makineler seri veya paralel olabilir. Paralel makineler aynı işlemi yapan tezgâhlardır, seri makineler ise farklı işlemi yapan tezgâhlardır. Paralel makinelerde gelen işler bu makinelerin herhangi birinde aynı sürede işlenebiliyorsa bu makineler paralel özdeş makinelerdir. Eğer paralel makinelerin işleme hızları birbirinden farklı ise bu makineler farklı hızda paralel makinelerdir. Eğer paralel makinelerin işleme hızları gelen

işinniteliğine göre de değışebiliyorsa, bu makineler paralel farklı makineler olarak adlandırılır.

#### **2.4.2 İş akışına göre sıralama problemleri**

**a)Akış Tipi:** Bu sistemlerde tek bir ürün veya birbirine benzeyenbirkaç ürün vardır. İş merkezleri, ürünün üretimi için gerekli işlemlere göreoluşturulmuştur. Özel amaçlı donanımlar üretim akışına göre sıralanmış olup, oldukçadüzgün ve hızlı bir ürün akışı vardır. İşlerin bütün makinelerdeki işleme sırasibirbirinin aynıdır. Bu düzgün iş akışı, üretim içi stoklara olan ihtiyacı azaltmakta,dolayısıyla çizelgeleme ve planlama çalışmaları büyük ölçüde kolaylaşmaktadır. Daha sonraki bölümlerde geniş olarak açıklanacaktır.

**b)Atölye Tipi** Bu sistemlerde iş merkezleri, bazı donanımlara veya işlemlere göre oluşturulmuştur. Örneğin pres, delme, montaj v.b. işlemlere gidecek olan ürünler, partiler halinde iş merkezlerinden geçerler. İş merkezleri arasındaki akış, herhangi bir sırayı izleyebilir. Bu sistemlerde ürün çeşitliliği çoktur. Donanım genel amaçlı olup donanımın kullanım oranı yüksektir. Bunun yanı sıra, ürün üretim aşamalarından yavaş yavaş geçer ve bundan dolayı üretim içi stok gereksinimleri doğar. Bu stokların kontrolü oldukça zordur. Doğal olarak çizelgeleme maliyetleri yüksektir. Donanım hazırlama maliyetleri de yüksek olup genel amaçlı donanımı tam anlamıyla kullanacak nitelikli iş gücü gereksinimi, personel giderlerini arttırmaktadır. Kısaca atölye tipi üretim sistemlerinde çizelgeleme işlevi zor ve pahalıdır.

**c)Esnek Atölye Tipi:** Klasik atölye tipinde paralel makineler yoktur. Atölyede işlerin paralel makinelerde işlenmesi söz konusu olduğunda esnek atölye tipi çizelgeleme problemi ortaya çıkar.

**d)Açık Atölye:** **Atölyeye:** Gelen işler atölyedeki makinelerde herhangi bir sırada işlenebilir. İşler için bir rotalama kısıtı söz konusu değildir.

#### **2.4.3 Başarı ölçütü**

Organizasyonun tüm amaçlarına ulaşılacaksa, çizelgelerin yönetsel olarak kabul edilebilen performans ölçütlerini yansıtmaları gereklidir. İşlerin sıralanmasında toplam

etkinliğin en iyilenmesi istenir. Ölçütler bu amaçla kullanılır. Bu ölçütlerden bazıları akış süresi, üretim içi stok seviyesi, işlerin tezgâhta bekleme süresi, işlerin ortalama gecikme süresi, geciken iş yüzdesi, tezgâh ve iş gücü kullanım oranlarıdır. Bu başarı ölçütlerinden bazılarının açıklaması şu şekildedir (Pinedo[10]):

-En Büyük Tamamlanma Zamanı (Makespan ( $C_{enb}$ )):Sistemdeki  $n$  adet işin tamamlanması ve sistemi terk etme zamanları  $C_j = (C_1, C_2, C_n)$  ile gösterilecek olursa, sistemi en son terk eden işin tamamlanma zamanı enküçüklenmeye çalışılır. Bu da yüksek makine kullanım oranı demektir.

-Akış Süresi (Flow Time( $F_j$ )):Kriter olarak işlerin sisteme girdiği an ile sistemi terk ettiği an arasındaki zaman farkı olarak tanımlanabilecek işlerin sistemde kalış süresi esas alınır.

-Geç Kalma Süresi (Lateness ( $L_j$ )): İşlerin teslim zamanları ile tamamlanma zamanları arasındaki farkı ifade eder. Geç Kalma Süresi planın teslim tarihinden sapmasını kontrol eder. Eğer pozitifse iş gecikmiş ve teslim zamanında yetiştirilememiş, negatifse iş teslim zamanından önce bitirilmiş demektir $\{L_j = C_j - d_j\}$ .

-Gecikme (Tardiness ( $T_j$ )) : İşler teslim zamanında bitirilememişse, ne kadar süre geciktiğini ifade eder $\{T_j = \max(0, L_j)\}$ .

-Geciken İş Sayısı ( $N_T$ ) : Teslim edilmesi gereken tarihte teslim edilemeyen iş sayısı da başarı ölçütü olabilir. Yapılan planların başarı veya başarısızlığını ölçebilmek için başarı ölçütü olarak yukarıda açıklanan ölçütler veya belirlenecek daha başka ölçütler de kullanılabilir.

## 2.5 Çizelgeleme Problemlerinin Karmaşıklığı

Optimizasyon problemleri, karar değişkenlerinin sürekli ve süreksiz olmasına göre iki kısma ayrılır.Kesikli çözüm uzayına sahip problemler için en iyi çözümü arayan ya da bulan yöntemlere kombinatoryal eniyilemedir.

Kombinatoryal en iyileme problemleri ya P ya da NP problemler olarak adlandırılır. P tipi problemler polinom zaman sınırlı bir algoritma ile etkin zamanda çözülebilmektedir. Eğer problemin çözümü için gereken adım sayısı problem ölçüsünün



bir kuvveti ile sınırlandırılıyorsa bu tür problemler “p problem” (polynomial time) sınıfında olarak kabul edilirler [16]. Örneğin verilen  $n$  basamaklı bir sayının asal olup olmadığını kontrol etmek için çalışma süresi  $n^6$  mertebesinde bir polinomla hesaplanabilen bir algoritma vardır. Dolayısıyla verilen bir sayının asal olup olmadığının araştırılması  $P$  kategorisinde bir problemdir.

$NP$  tipi problemler için polinom zaman sınırlı bir algoritmanın bulunması mümkün görülmemekte ve bu problemler eniyi olarak ancak üstel zamanda çözülebilmektedir. Eğer problem non-deterministik bir çözüme izin veriyor ve çözümü doğrulamak için gereken adımların sayısı problem ölçüsünün bir kuvveti olarak sınırlandırılabilirse bu tür problemler de  $NP$ -problem (nondeterministik polynomialtime) sınıfına girerler. Örnek olarak verilen  $n$  basamaklı bir sayının asal çarpanlarının neler olduğu sorusunu düşünebiliriz. Bu sorunun cevabı için bilinen en iyi algoritmanın çalışma süresi  $n$  sayısına bir polinom cinsinden değil de eksponansiyel fonksiyonlar cinsinden ( $e^n$  misali) bağlıdır, fakat bu problem için eğer bir şekilde cevabı tahmin edebiliyorsak tahminimizin doğruluğunu sınamak için  $n$  sayısına polinom mertebesinde bağımlı bir sürede çalışacak bir algoritma mevcuttur. Dolayısıyla verilen bir  $n$  basamaklı sayının asal çarpanlarının neler olduğu sorusu  $NP$  kategorisindedir [16].

Bir çözüm  $NP$ -problem olarak biliniyorsa, basit polinom zamanlı sağlamaya indirgenebilirdir. Eğer problemin çözüm algoritması diğer herhangi bir  $NP$  problemin çözümü için kullanılacak algoritmaya dönüştürülebiliyorsa bu tür problemler  $NP$  complete olarak tanımlanır (Weisstein [17]). Eğer bir problemi çözen algoritma, başka herhangi bir  $NP$  problemi çözen algoritmaya dönüştürülebiliyorsa, böyle problemlere  $NP$  hard problem denir. Bir problemin  $NP$  olduğunu göstermek  $NP$ -hard olduğunu göstermekten çok daha kolaydır (Weisstein [17]).

Kombinatoriyel optimizasyon problemlerinin büyük çoğunluğu, polinomiye zaman sınırı olmayan problemler sınıfına girerler.  $NP$  problemleri için polinomiye algoritmalar geliştirilememiştir. Bunun yerine  $NP$  kapsamına giren problemlerde, optimum çözüme ulaşmak yerine, en yakın çözümlere ulaşmak tercih edilir. Bu tip problemlerin kesin çözümleri çok zaman aldığından, yerel arama ve stokastik arama yöntemleri ile yaklaşık

özmler elde edilmeye alışılır. Yerel arama yöntemlerinin, yerel optimum noktalarında takılıp kalmasını önlemek için stokastik arama yöntemleri geliştirilmiştir.

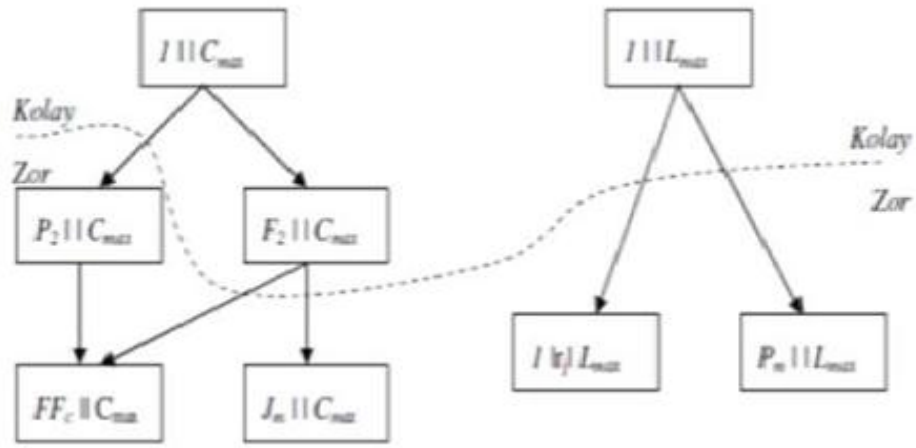
Kombinatoriyel optimizasyon problemlerinin büyük bir kısmı sezgisel yöntemler ile özlebilmektedir. Yani bu tür problemler için optimum değil optimuma yakın özmler elde edilir (Temiz [18]).

NP problemlerinin özümünde kullanılan metotları şöyle sınıflandırabiliriz:

- Yerel arama metotları
- Yapay sinir ağıları
- Tabu araştırmaları
- Tavlama benzetimi
- Karınca kolonileri
- Yapay bağışıklık sistemleri
- Genetik algoritmalar
- Arı algoritması
- Tepe tırmanma
- Paracık sürü optimizasyonu

izelgeleme problemleri kombinatoriyel eniyileme problemlerine örnektir. Yapılan alışmalar izelgeleme problemleri için geliştirilen bazı algoritmaların polinomiye zamanda özme ulaştırdığını göstermiştir. Bununla birlikte çoğu izelgeleme problemi polinomiye zamanlı algoritmalarla özlmediği için NP-hard problemler olarak tanımlanmaktadır [10],[19]. NP-hard olarak tanımlanan problemlerin matematiksel olarak ispatının yapılması gerekmektedir.

Şekil 2.2’de görldüğü gibi, tamamlanma zamanını en küçklemek amaçlı ve tek makineyi veya iki makineyi akış tipli problemler kolay sınıfına girerken, paralel ikimakineli veya esnek akış tipli problemleri zor sınıfına girmektedir.



Şekil 2.2 Örnek problemlerin çözüm komplekslikleri (Baker[20])

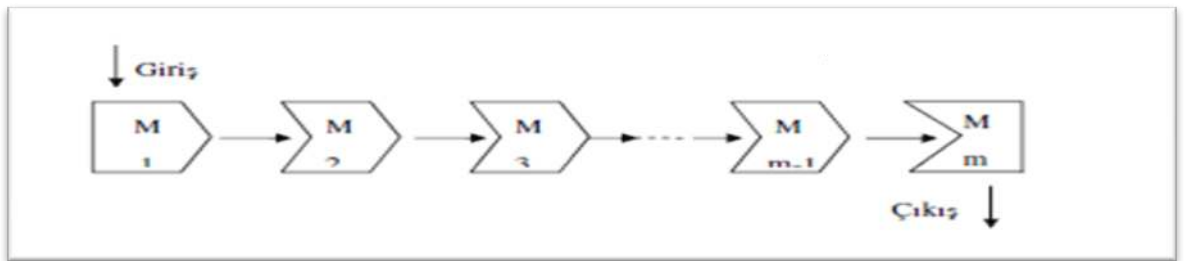
## BÖLÜM 3

### AKIŞ TİPİ İŞ ÇİZELGELEME

Makinelerin birbiri ardına seri olarak sıralandığı yerleşim düzenlerine akış tipi yerleşim denmektedir. Günümüzde birçok üretim tesisinde her bir iş üzerinde bir dizi operasyonun uygulanmasıyla elde edilmektedir. Bu, bütün işlerin aynı sırayla aynı operasyonlardan akmasını gerektirir. Yapılacak işler başlangıç makinesinden başlayarak son buluncaya kadar bir dizi makinde işlem görür ve nihayetinde son makinede işlemlerini tamamlarlar. Bu durumdan doğan akış tipi çizelgeleme kavramı temelde,  $m$  makine ve  $n$  işin bulunduğu bir ortamda, her iş için  $m$  operasyonun bulunması ve bu operasyonların farklı makinelerde ve aynı sıra ile yapılmasıdır.

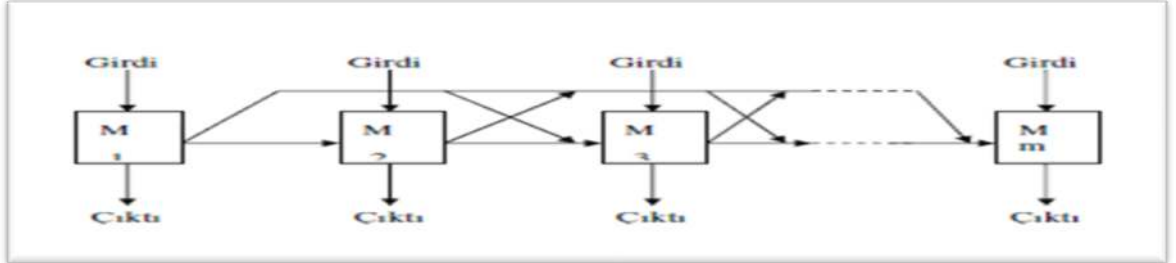
Akış tipi çizelgeleme, birbirinden farklı  $m$  makine ve  $n$  işin bulunduğu, her bir işin  $m$  operasyondan oluştuğu, her bir operasyonun farklı makinelerde yapıldığı ve bütün işlerin operasyonlarının aynı sıra ile yapıldığı çizelgeleme problemleridir [19],[20],[21]. Akış tipi çizelgeleme problemleri üretim aşamasındaki en önemli problemlerden biridir.

Akış tipi yerleşimin çeşitleri bulunmaktadır.  $m$  operasyondan oluşan bir iş ve her bir operasyon için  $m$  adet farklı makine bulunan yerleşim tipi saf akış tipine örnektir.



Şekil 3. 1 Saf akış tipi yerleşim (Baker [20])

Genel akış tipi yerleşiminde bazı işler  $m$  adet operasyona ihtiyaç duyabilirler. İşin ihtiyaç duyduğu operasyonlar birbiri ardına gelen olmayabilirler veya başlangıç ve bitiş operasyonları her bir iş için farklı olabilir. Bu olası şartlara rağmen iş akışları doğrusaldır (Baker [20]).



Şekil 3.2 Genel akış tipi yerleşim (Baker [20])

Bu iki çeşitle beraber akış tipi yerleşiminin özel uygulamaları olan tekrar eden girişli (reentrant) akış tipi, bileşik (compound) akış tipi, sınırlı kuyruklu (finite queue) akış tipi uygulamaları da bulunmaktadır. Tekrar eden girişli akış tipinde işler bazı tezgâhları birden fazla ziyaret etmektedir. Bileşik akış tipinde seri haldeki tezgâhlar yerine birbirine bağlı makine grupları bulunmaktadır. Bunlar seri veya paralel çeşitleri olmasına rağmen genelde paralel tezgâhlardan oluşmaktadırlar. Sınırlı kuyruk akış tipinde ise diğer çeşitlerden farklı olarak birbirine seri bağlı olan makinelerin önünde limitli depo vardır [22],[23].

### 3.1 Çizelgelemede Kullanılan Öncelik Kuralları ve Kabulleri

French'e göre akış tipi çizelgeleme probleminin çözümünde, şu kabuller yapılır (Engin [24]):

1. Bir iş, farklı operasyonların birleşiminden meydana gelmesine rağmen, aynı işin iki operasyonu aynı anda işlenemez.
2. Önceden işler için ayarlama yapılamaz, bir makinede işlem yapılırken diğer işlemin başlayabilmesi için bir öncekinin bitmiş olması gerekir.
3. Her iş  $m$  tane farklı operasyondan oluşur. Bir işin bir makinede ikinci bir defa işlem görmesine izin verilmez.

4. Operasyonların ertelenmesi yapılamaz, her işin süreci tamamlanmalıdır, eksik bırakılıp sonra tamamlanamaz.

5. İşlem zamanları çizelgeden bağımsızdır. Burada iki durum kabul edilir. Birincisi, her bir işlemin hazırlık süresi çizelgeden bağımsızdır. İkincisi, işlerin makineler arasındaki taşıma süreleri ihmal edilir.

6. Süreçler arasında envanter stoklanmasına izin verilebilir; iş, bir sonraki makinede işlenmek üzere bekleyebilir.

7. Her bir makine tipinden sadece bir tane mevcuttur, süreç içerisinde işin makine seçmesine izin verilmez. Bu kabul bazı durumlarda ortadan kaldırılabilir, belirli makineler birden fazla eklenerek darboğaz ortadan kaldırılmaya çalışılabilir.

8. Makinelerin, boş kalma süreleri olabilir.

9. Hiçbir makine bir anda birden fazla operasyonu yerine getiremez.

10. Çizelge periyodu boyunca makineler hazır beklerler. Yani makinelerde arızalar meydana gelmez.

11. Teknolojik kısıtlar önceden bilinmektedir.

12. Olayda rastsallık yoktur. Bu yüzden,

- İş sayısı biliniyor ve sabittir.
- Makine sayısı biliniyor ve sabittir.
- İşlem zamanları biliniyor ve sabittir.
- İhtiyaç duyulacak diğer parametreler baştan beri bilinmektedir.

Makinelerde çizelgelenecek işler, üretilecek çizelgenin amacına ve ulaşılmak istenen esas amaca göre belirli kurallara göre çizelgelenir. Örneğin, amaç eğer işlerin teslim tarihlerini geçirmemek ve zamanında teslim etmek ise teslim zamanı önce olan işler makinelerde önce işlenebilir ya da makinelerin dengeli kullanımı amaçlanıyorsa işler enküçük ve enbüyük işlem zamanlarına göre sırasıyla çizelgelenebilir. Bu kurallarının kullanımı anlaşılmasının ve uygulanmasının kolay olması açısından oldukça avantajlıdır.

- 1.En erken teslim tarihi(EDD): En erken teslim tarihine sahip olan iş ilk önce işlem görür.
- 2.İlk gelen ilk çıkar (FIFO): İlk önce sisteme gelen iş ilk önce işlem görür.
- 3.En kısa işlem zamanı (SPT): En kısa işlem süresine sahip iş önce seçilir.
- 4.En uzun işlem zamanı (LPT): En uzun işlem sahibine olan iş önce seçilir.
- 5.Rastgele: İş seçimi rastgele seçilir.
- 6.En az kalan işlem zamanı: Kalan toplam işlem süresi en az olan iş önce seçilir.
- 7.En uzun kalan işlem zamanı: Kalan toplam işlem süresi en fazla olan iş önce seçilir.
- 8.Kalan en az operasyon sayısı: Kalan toplam operasyon sayısı en az olan iş önce seçilir.
- 9.Kalan en fazla operasyon sayısı: Kalan toplam operasyon sayısı en fazla olan iş önce seçilir.

### **3.2 Akış Tipi Çizelgeleme Problemlerinin Çözümü ve Sezgisel Yöntemler**

Optimal çözüm arayan yöntemlerin (dinamik programlama, tamsayılı programlama, dal-sınır algoritmaları vb.) büyük boyutlu problemler için kullanılması oldukça zordur. Teknolojinin gelişmesiyle işlem kapasitesi yüksek bilgisayarlar bulunsa da büyük boyutlu problemlerin kısa zamanda çözülmesinin zorluğu devam etmektedir. Bunun için büyük hacimli problemler için işlem kapasitesinin artırılması veya çözüm sürecinin basitleştirilmesi gerekir. İşlem kapasitesinin mevcut teknolojiye bağlı olarak gelişmektedir, bununla birlikte istenen seviyenin çok gerisindedir. Çözüm algoritmalarının basitleştirilmesi optimal çözümü bulmaz fakat yerel optimum veya optimuma yakın çözümlere ulaşılmasını sağlar. Bu yüzden sezgisel algoritmalar önem kazanmaktadır ve işlem zamanında kazançlı çıkıp optimale yakın çözümler üreten algoritmalarla karar almada etkinlik sağlanabilmektedir.

Akış tipi çizelgeleme problemlerinde amaç fonksiyonu, toplam akış süresi ağırlığını ve en son işin tamamlanma zamanını minimize etmektir. Akış tipi çizelgelemede, iki makine ve şartlı 3 makine durumu dışında, optimal çözüm üreten kolay bir algoritma bulunmamaktadır (Johnson [25]). Bu algoritma Johnson algoritmasıdır. Akış tipinde optimum çözümü bulmak için tamsayılı programlama ve dal sınır tekniği de

kullanılabilir (Ignall ve Schrage [26]). Fakat genelde iki makinden fazla olduğunda çoğunlukla NP-hard kombinatoriyel problem sınıfına girmektedir [19], [27]. Akış tipi iş çizelgeleme problemlerinde makine sayısı ( $m > 2$ ) olduğu için bu problemlerin çözümü için çeşitli sezgisel yöntemler geliştirilmiştir. Bunların başlıcaları şunlardır:

Palmer'in eğim dizisi (1965), operasyon sürelerinin ilk makinede kısa olanların öne uzun olanların sona yerleştirildiği eğim dizinin tanımlanıp işlerin sıralanmasıdır [28]. Campbell, Dudek, Smith Algoritması (1970),  $m$  makine  $n$  iş problemi  $m-1$  adet  $n$  iş iki makine problemine dönüştürülerek Johnson algoritmasıyla sırasıyla çözülerek en iyi sonucu vereni bulmaya dayanan bir algoritmadır [27]. Gupta Yöntemi (1971), Palmer'e benzer bir eğim dizisinin oluşturulup ancak farklı bir yöntemle çözüldüğü bir dizinin sıralanmasıdır [30]. Dannenbring Yöntemi (1972), Palmer ve Campbell, Dudek, Smith'in geliştirdikleri algoritmaların avantajlarını birleştirerek yeni bir sezgisel yöntem geliştirmişlerdir [31]. Nawaz, Enscoe ve Ham Yöntemi (1983), işlerin tezgâhtaki işlem zamanlarına göre büyükten küçüğe sıralayan bir yöntemdir [32]. Hundal ve Rajgopal Yöntemi (1988), Palmer'in sezgisel yönteminin geliştirilmesiyle elde edilen bir yöntemdir [33]. Widmer ve Hertz Yöntemi (1989), ilk aşamada gezgin satıcı problemindeki gibi analoji ile başlangıç çözümünün bulunup ikinci aşamada tabu arama tekniğiyle iyileştirilen bir yöntemdir [34]. Ho ve Chang Yöntemi (1991) : sezgisel metotlarla belirlenmiş işlerden, birbirini takip eden işler arasındaki boşlukların minimize edildiği bir yöntemdir [35].

Yukarıdaki sezgisel yöntemlerin dışında akış tipi iş çizelgeleme problemlerinin çözümünde tavlama yöntemi [36], [37]; genetik algoritma [38], [39], [40], [41], [42], [43]; tabu arama [44],[45]; bulanık mantık [46],[47]; karınca kolonisi ve parçacık sürü optimizasyonu [48],[49] gibi yöntemlerde çeşitli amaç fonksiyonuna sahip problemler için de kullanılmışlardır.



**SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ**

Araştırmalarda önerilen yöntemlerin performanslarının verim ve etkinliğini ölçmede belirli bir dağılıma uyan rassal sayılardan oluşmuş işlem zamanı matrisleri kullanılır. Fakat herhangi bir dağılım kullanılarak elde edilen rastsal veriler kullanan araştırma problemlerinin pratikte karşılaştığımız problemlerle birebir uyuşmaz, sınırlı bir veritabanı sağlar. Yapılan çalışmalar, teorik problemlerin, işlerin ve tesisin fiziksel karakteristikleriyle orantılı işlem zamanlarına sahip gerçek problemlerle karşılaştırıldıklarında, çözüm açısından kolay olmadığını gözlemlemişlerdir (Körez [50]).

Akış tipi çizelgeleme problemleri içerisinde rassal olmayan bir çeşidi, sıralı işlem zamanlı akış tipi çizelgeleme problemleridir [1].

Sıralı akış problemi, Smith, Panwalkar ve Dudek, tarafından  $n$ -iş,  $m$ -makine probleminin bir alt kategorisi olarak tanımlanmıştır. Bir akış tipi iş çizelgeme probleminin sıralı akış tipi iş çizelgemesi için iki tane özelliğinin olması gerekmektedir [2].

(1) Herhangi bir iş herhangi bir makinede aynı makine üzerindeki ikinci işten daha küçük işlem zamanına sahipse, birinci iş ilgili tüm makinelerde ikinci işten küçük ya da eşit işlem zamanına sahiptir.

(2) Herhangi bir iş, bir makinede  $j$ . küçük işlem zamanına sahipse, diğer tüm işler aynı makinede  $j$ . küçük işlem zamanına sahiptir. Başka bir deyişle, verilen bir iş için minimum işlem zamanına sahip bir makine, diğer tüm işler için de minimum işlem zamanına sahiptir.

Smith ve Khan tarafından tanımlanan iki sıralı problem özelliğini aşağıdaki gibi matematiksel modelle göstermişlerdir [3]:

i. Eğer her  $i$  için,  $P_{it} > P_{kt}$ ,  $k \in N$  ve  $t \in M$ ; ise, Her  $j \in M$  için,  $P_{ij} \geq P_{kj}$  dir.

ii. Eğer her  $i \in N$  ve  $r, j \in M$  için,  $P_{ir} > P_{ij}$  ise,

Her  $k \in N$  için,  $P_{kr} \geq P_{kj}$  dir.

$P_{ij}$   $i$ . işin  $j$ . makinedeki işlem zamanını,  $N$  işin kümesini,  $M$  ise  $m$  makinenin kümesini göstermektedir.

Bunları sağlayan çizelgeye smith sıralı işleme sürelerinin zaman matrisi adını vermiştir. Daha sonraki çalışmalarında bu tablonun eşliğinde basitleştirerek sıralı akış tipi iş çizelgeleme problemi adını vermiştir. Çizelge 4.1 de bu problem tipine uygun bir zaman matrisi örneği bulunmaktadır.

Çizelge 4.1 4-iş 4-makine problemi için sıralı matris

| İŞLER | MAKİNELER |     |    |    |
|-------|-----------|-----|----|----|
|       | A         | B   | C  | D  |
| 1     | 90        | 120 | 80 | 70 |
| 2     | 40        | 54  | 42 | 30 |
| 3     | 74        | 97  | 77 | 58 |
| 4     | 56        | 78  | 59 | 31 |

#### 4.1 Sıralı Problemlerin Uygulama Temelleri

Sıralı problemler uygulamalarına çeşitli endüstri alanlarında rastlanılmaktadır. İşlere göre işlem zamanlarının sıralanması pek çok faktörden kaynaklanır. Bu faktörlerden işlerin birim ya da parça sayısı örnek verilebilir. Eğer her iş için operasyonların yapısı oldukça benzer ancak işler üretim sayılarında büyük farklar içeriyorsa, bütün işler için işlem zamanı birimleri bu farklılığı sıralı bir yapı oluşturarak yansıtır.

İşlem zamanlarını etkileyen bir diğer faktör ise işlerin karmaşıklığıdır. Bir iş her birimde çok basit işlemler gerektirirken başka bir iş zaman harcayıcı işlemler gerektirebilir. Bu durumda, işlerin karmaşıklığı tüm makinelerdeki işlerin işlem zamanları için ciddi bir etki oluşturur.

Benzer şekilde, işlem zamanlarının makineler göre sıralanması makinelerdeki operasyonların farklı yapılarının sonucudur. Eğer bir makinedeki operasyon tüm işler için karmaşıksa, ancak başka bir makinedeki operasyon tüm işler için daha basit ise, ilk makinedeki birim işlem zamanları ikinci makineye oranla daha büyük olacaktır. Bu operasyon karmaşıklığı tüm işleri etkileyeceğinden sıralı işlem zamanı problemlerinin ikinci özgünlüğünün temel sebebidir [2].

#### 4.2 Sıralı Akış Tipi Çizelgeleme Problemi Çeşitleri

Smith, Panwalkar ve Dudek sıralı akış tipi problemleri ele almaya başladıklarında pek çok problem üzerinde çalıştılar. Bu çalışmalarının sonucunda sıralı problemler üzerinde iki önemli karakteristik özellik gözlemlerler. Bunların ilki, eğer sıralı bir problemdeki maksimum işlem zamanı ilk (son) makinede meydana gelmişse, işlerin sıralı işlem zamanlarını azalan (artan) sıra ile ya da literatürde bilinen halleriyle SPT(LPT) kuralıyla sıralamanın daima minimum tamamlanma zamanlı sıralamayı vermesidir [8],[50].

İkincisi ise, eğer sıralı bir problemdeki maksimum işlem zamanı ara makinelerden birinde gerçekleşmiş ise, sıralı işlem zamanları artan, kalan işlerin ise azalan sıralaması şeklinde takip ettiği bir sıralama alt kümesi daima minimum tamamlanma zamanlı sıralamayı vermektedir. Yani optimum sonuç SPT-n-LPT sıralanmasının  $2^{n-1}$  taneli alt kümelerinden biri olacaktır. Bu SPT-n-LPT yapısına piramit yapısı adını vermişlerdir [8],[50].

Panwalker ve Khan sıralı akış tipi iş çizelgeleme problemleri için ortalama akış süresinin minimum değerinin elde edeceği iş sıralanmasının SPT kuralıyla yapıldığında elde edildiğini incelediler [3].

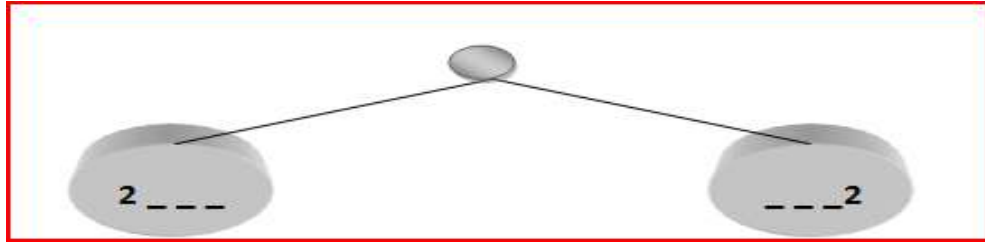
Yapılan çalışmaların sonucuna göre Smith ve diğerleri, sıralı akış tipi iş çizelgeleme için iki temel problem tipini tanımladılar. Bu problem tiplerinin çözümü için iki farklı algoritma önerdiler. Bu problemlerden ilki, genel durum olan tüm işler için sıralı işlem zamanı matrisinde işlem zamanı en büyük olan işin ara makinelerden birinde oluşması durumudur. İkinci problem ise genel durumun özel bir gelişimi olan tüm işler için maksimum işlem zamanlı işin ilk ya da son makinede oluşması durumudur [2].

#### 4.4.1 Genel durum problemi

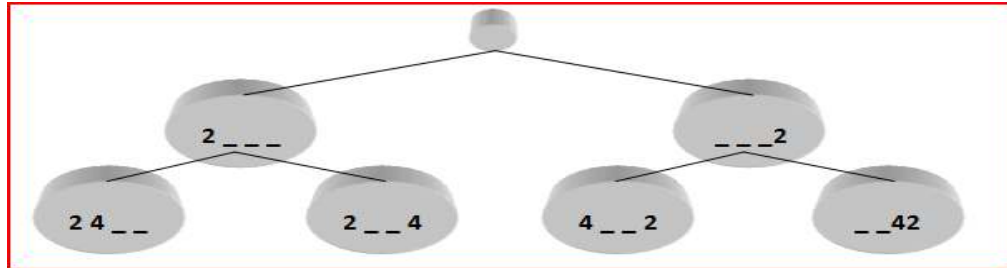
Sıralı akış tipi çizelgeleme problemlerinde, genel durum olarak tanımlanan problem çeşidinde, maksimum işlem zamanı ara makinelerden birinde oluşmaktadır. Bu problem çeşidinin çözümü için önerilen algoritma temelde,  $2^{n-1}$  sıralamanın oluşturulduğu sınırlı deneme yanılma yönteminin bir çeşididir. Önerilen algoritmanın işlem adımlarını çizelge 4.1 de verilerle beraber aşağıda gösterildiği gibi uygulanır [2],[50]:

1)İşleri işlem zamanı matrisindeki ilk makinenin işlem zamanlarına göre artan sıralama ile sıralayın. Eğer işlem zamanlarındaki eşitlik durumu varsa, ikinci ya da üçüncü makinedeki işlem zamanları dikkate alınır.

2) Oluşturulan parçalı sıralama yapısı içinde boş en soldaki pozisyona sıralanmamış en düşük değerli işi yerleştirin. Aynı işi parçalı sıralama yapısı içerisinde boş en sağdaki pozisyona da yerleştirin.

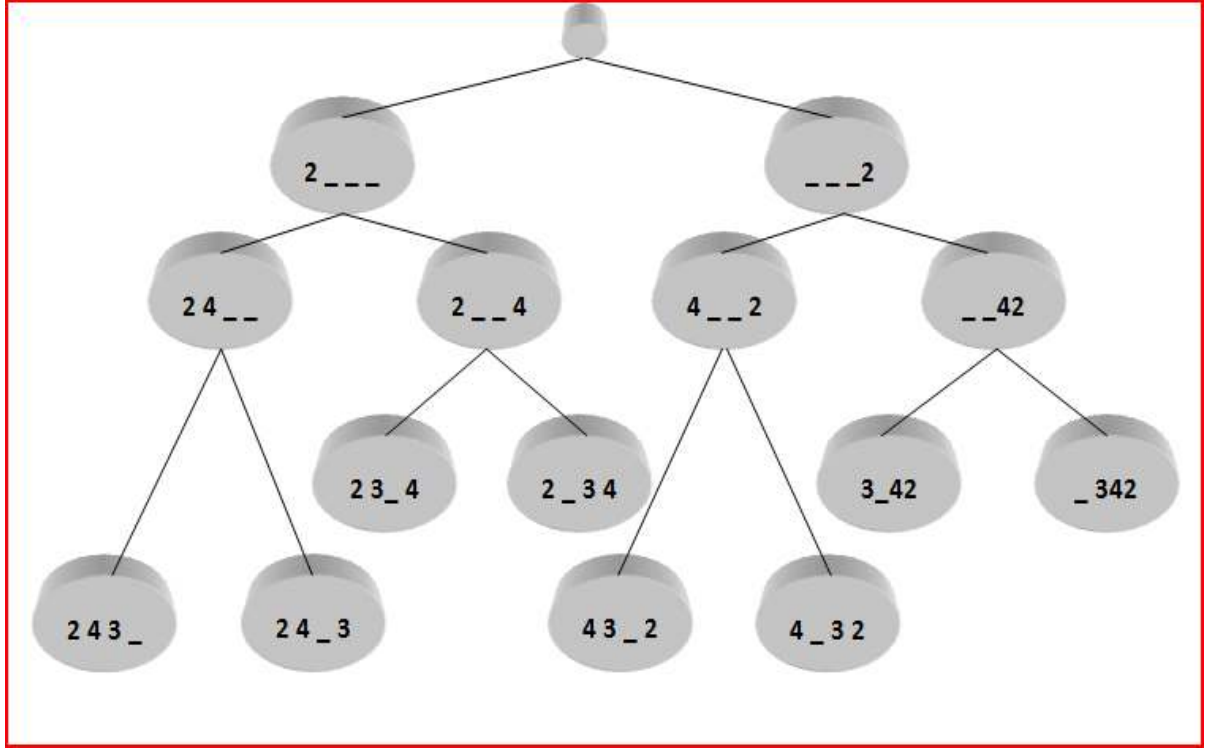


Şekil 4.1 Örnek SPD algoritmasının birinci adımı



Şekil 4.2 Örnek SPD algoritmasının ikinci adımı

3)2.adımdaki her sıralamaya ilk n-1 iş yerleştirilene kadar devam edin. Son olarak en yüksek değerli işi sıralamalardaki boş noktaya yerleştirin.

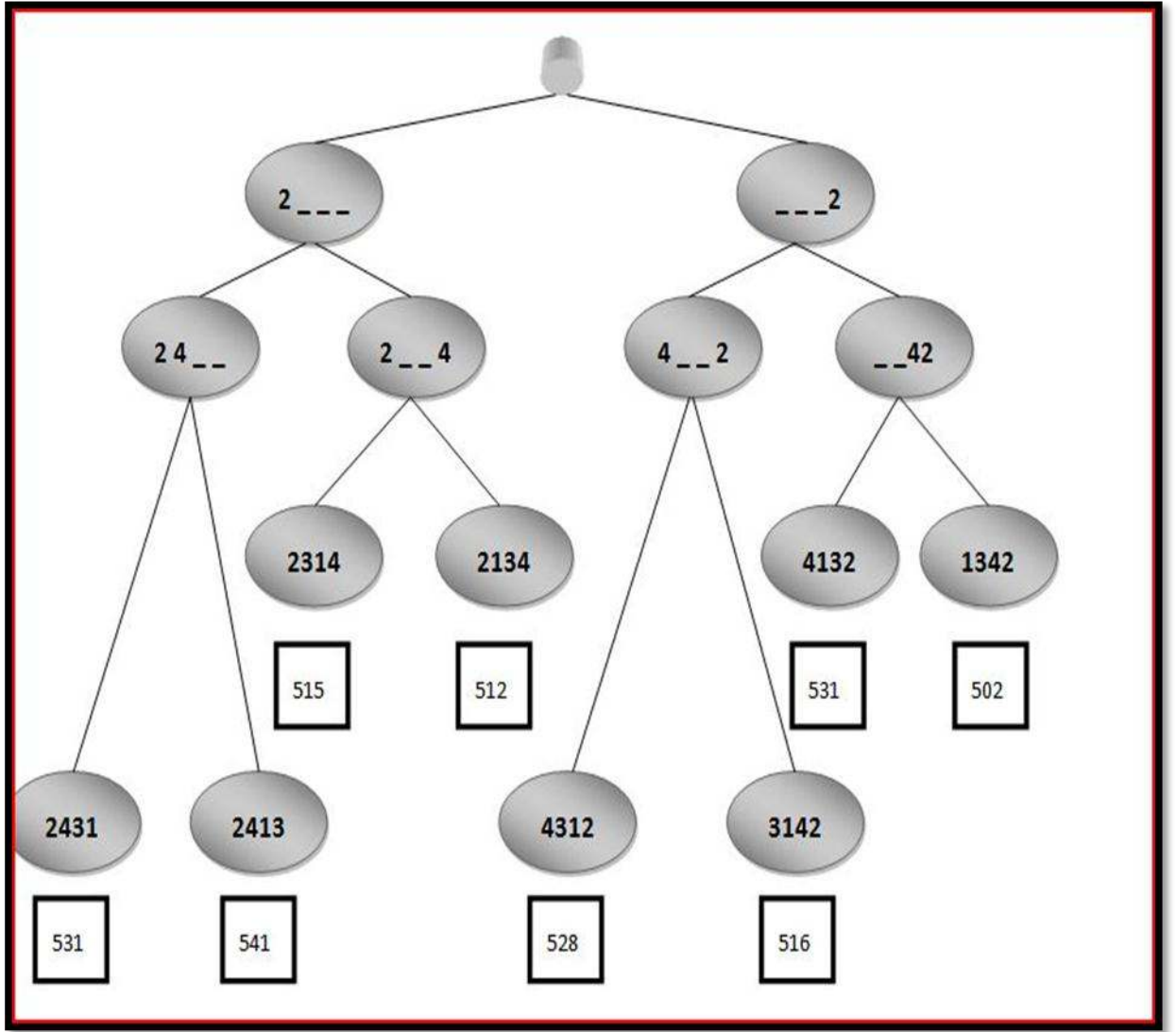


Şekil 4.3 Örnek SPD algoritmasının üçüncü adımı

4) Tamamlanma zamanı için  $2^{n-1}$  sıralamayı değerlendirin. En kısa tamamlanma zamanını veren sıralama en uygundur.

Şekil 4,1 de görüldüğü gibi anlatılan 4 adım uygulandığında 8 tane ( $2^{(4-1)}$ ) tane üretilen çizelgelemelerde en düşük tamamlanma zamanına sahip sıralama 502 birim zamana sahip olan 1-3-4-2 iş sıralanmasıdır. Şekilde gösterilen çözüm yöntemi, incelendiğinde dal-sınır arama yöntemine benzemektedirler.

Ancak çözüm içerisindeki pek çok nokta uygun üst sınır değerleri hesaplanarak elemine edilebilir. Algoritma tarafından oluşturulan tüm sıralamalarında elde edilen karakteristik özellik, en uzun işlem zamanına sahip işin yeri sabitlendiğinde, başlangıç işinden en uzun işlem zamanı olan işin bir solundaki işe kadar önceki tüm işler artan bir sıralama ile yerleştirme şeklindedir. Benzer şekilde, en uzun işlem zamanlı işten sonraki işler de azalan sıralama ile yerleşirler. Bu karakteristiği sağlayan bir sıralama şekil 4,1’de de görüldüğü gibi piramit şeklinde olacaktır. Genel durum problemi için önerilen çözüm yöntemi en az bir optimal çözüm vermektedir [50].



Şekil 4.4 Örnek SPD algoritmasının çözümü

#### 4.4.2 Özel Durum Problemi

Sıralı akış tipi çizelgeleme problemlerinde ikinci problem çeşidi ise genel durumun özel bir gelişimi olan tüm işler için maksimum işlem zamanlı işin ilk ya da son makinede oluşması durumudur. Bu problemin çözümünün gelişimi sırasında Smith,Dudek ve Teuton'un genel varsayımları kullanarak problemin optimizasyon kriteri olarak toplam akış zamanının minimizasyonu ele almıştır. Problemin çözüm adımları aşağıdaki gibidir [1]:

- 1)İşleri herhangi bir makinedeki işlem zamanlarına göre artan sıralama ile sıralayın.
- 2)Eğer en büyük işlem zamanları son makinede gerçekleşmişse, işleri birinci adımda elde edilen sıralamaya göre sıralayın. Eğer ilk makine en büyük işlem

zamanları nasahipse, işleri birinci adımda elde edilen sıralamanın tersine göre sıralayın. Eğer ilk ve son makine eşit işlem zamanlarına sahip ise, iki sıralama da optimum sonuç verecektir.

Yukarıdaki algoritmanın toplam akış zamanı için uygun değer sonuç verdiği Smith, Dudek ve Panwalkar tarafından ispatlanmıştır [2].

Çizelge 4.2 Sıralı problem iş zamanları

| İŞLER | MAKİNE |    |    |    | PARÇA SAYILARI |
|-------|--------|----|----|----|----------------|
|       | A      | B  | C  | D  |                |
| 1     | 28     | 21 | 25 | 30 | 175            |
| 2     | 23     | 17 | 20 | 26 | 150            |
| 3     | 30     | 27 | 29 | 35 | 200            |
| 4     | 18     | 13 | 17 | 23 | 140            |

Çizelge 4,2'deki verileri bu bölümde anlatılan adımları uygulayalım. İlk adıma göre işleri belirli bir makineye göre işlem zamanlarına göre sıralanması gerekir. İşleri ikinci makinedeki işlem zamanlarına göre sıralarsak 4-2-1-3 sıralaması 175 birim zamanla en uygun sıralanmasını vermektedir.

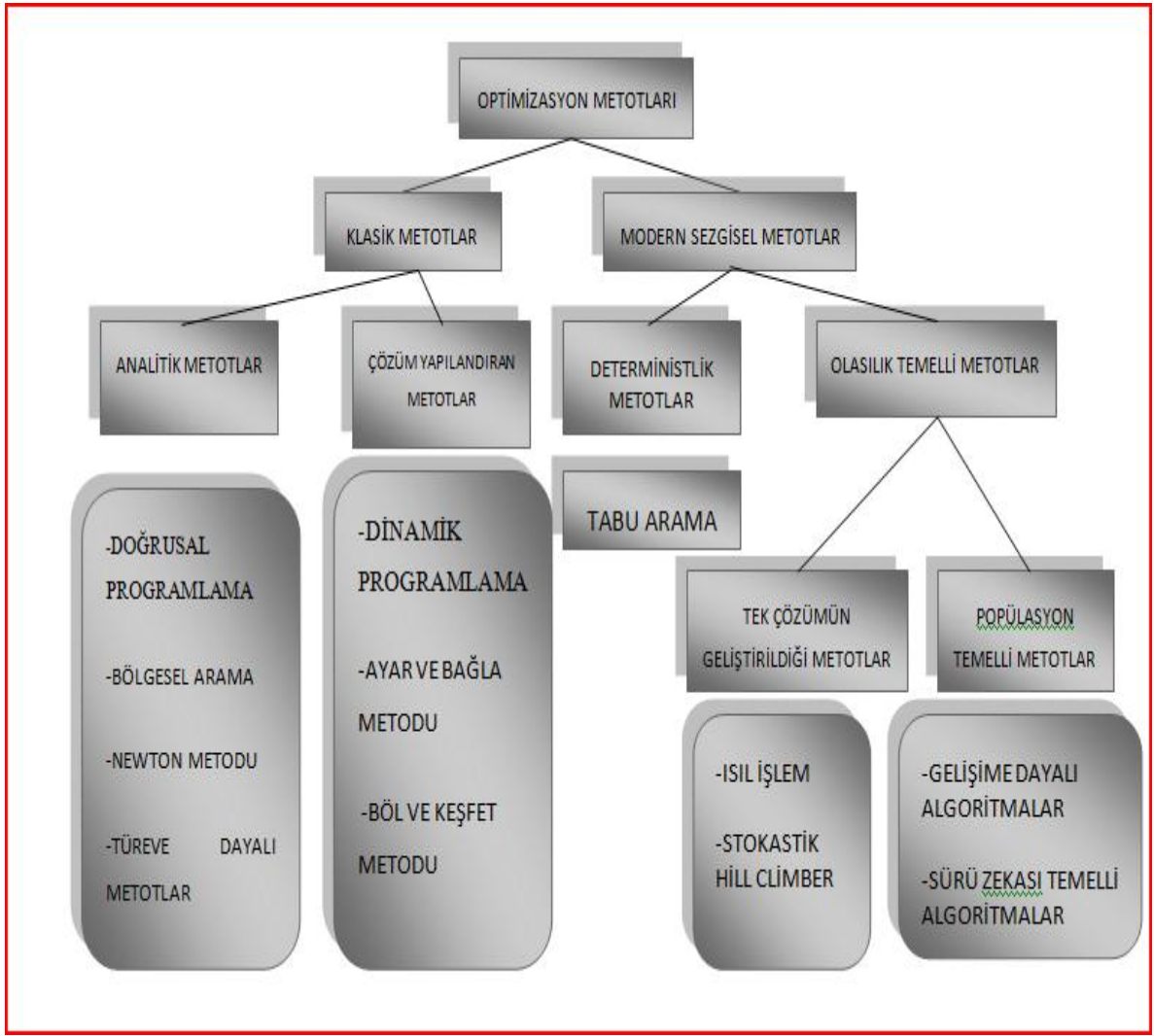
### ARI ALGORİTMASI

#### 5.1 Giriş

Araştırmacılar son yıllarda yaptıkları çalışmalarda doğada var olan sistemlerden çeşitli şekillerden etkilenmişlerdir. Klasik optimizasyon metotları zor problemlerin çözümü konusunda büyük zorluklara neden olmaktadır [51]. Bu yüzden bu NP-hard problemlerin çözümünde uygun sürelerde en iyi sonuca yakın yaklaşık değerler veren algoritmalarla ilgilenmişlerdir. Araştırmacılar çeşitli optimizasyon problemlerini klasik optimizasyon yöntemlerine uyarlamak için oldukça çaba göstermişlerdir. Şekil 5.1’de gösterilen optimizasyon problemlerinin sınıflandırılmasında bu yöntem çeşitlerini görmekteyiz. Önerilen bu sezgisel yöntemler hem etkin hem de daha esnek olup belirli problem gereksinimlerine göre uyarlanabilmektedir.

Bu metotlardan biri de sürü zekâsı temelli algoritmalar. Doğadan esinlenen algoritmaların yeni bir dalı olan sürü zekâsı yaklaşımı, böceklerin içgüdüsel problem çözme becerilerini kullanan etkili meta sezgisel yöntemler geliştirebilmek için böcek davranışlarının modellenmesine odaklanmıştır. Böcekler arasındaki etkileşimin bir sonucu olan kolektif zekânın en önemli parçalarından biri ise bireysel böcekler arasındaki bilgi paylaşımıdır. Bu tür etkileşimli davranışa örnek sürülerden biri işlerini dinamik olarak dağıtabilen ve çevresel bal arıları örnek verilebilir. Bal arıları işlerini dinamik olarak dağıtabilen ve çevresel değişimlere karşı topluluk zekâlarıyla uyarlanabilir cevaplar verebilen bir sürü yapısına sahiptirler (Akay[52]). İleriki kısımlarda arı algoritması hakkında bilgi verilecektir.





Şekil 5.1 Optimizasyon problemlerinin sınıflandırılması (Akay [52])

### 5.1 Doğal Ortamında Arılar

Arılar koloni halinde yaşayan sosyal böceklerdir. Bal arıları, içgüdüsel olarak düzeni bilir ve hayatını kurallarına uygun olarak devam ettirir. Arıların en önemli özelliği koloni içinde hepsinin görevinin belirli olmasıdır ve bir bütün halinde olmalarıdır. Yiyeceklerin depolanması, balın getirilmesi, iletişim ve besin arama gibi işler arıların koloni içindeki görevleridir. Arılar bu görevlerin dışına çıkmazlar. Kovana ait olan her arının görevi bellidir. Bunlar: kraliçe arı, erkek arı ve dişi olan işçi arılardır.

**Kraliçe Arı:** Kolonideki arıların annesi olan kraliçe arı, diğer dişi arılara nazaran en büyüktür ve her kolonide bir tane bulunmaktadır. Kolonide yumurtlama özelliğine sahip kraliçe arı sayesinde üreme meydana gelir, çünkü onun dışındaki diğer dişi arılar

erkek arılarla çiftleşmezler. Bununla birlikte, koloninin bütünlüğünü ve sistemin işleyişini sağlayan maddelerin salgılanmasından sorumludurlar. Çünkü salgılanan bu maddeyle diğer arılar kraliçe arıyı tanır ve bu koku kraliçe arı etrafındaki tüm arı ailesine bulaşır. Bu sayede yabancı kraliçe arı ve kolonisi fark edilir. Böylece koloninin bütünlüğü sağlanmış olur [53], [54], [55], [56].

**Erkek Arılar:** Erkek arılar döllenmemiş yumurtalardan oluşurlar. 6 aydan fazla yaşayamazlar. Erkek arıların kolonideki görevi kraliçe arının döllenmesidir. Dişi arılardan iridirler ancak iğneleri ve besin toplayacak organları yoktur [53], [54], [55], [56].

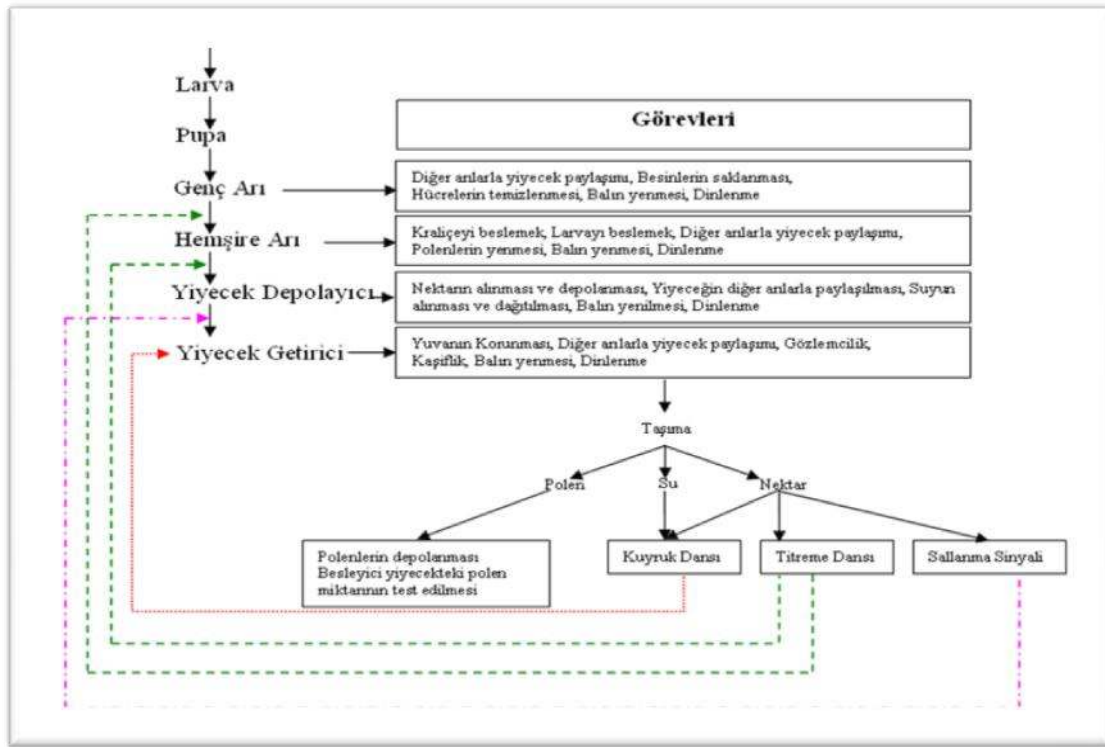
**İşçi Arılar:** işçi arılar döllenmiş yumurtalardan çıkan üreme yeteneği olmayan dişilerdir. Koloninin büyük çoğunluğu oluştururlar. İşçi arılar yiyecek toplanmasından, ölü arıların ve molozların kaldırılmasından, toplanan yiyeceklerin korunmasından, kovanın havalandırılmasından, kovanın güvenliğinden ve diğer arıların beslenmesinden sorumludurlar.

Kendi aralarında iş bölümü yapmak suretiyle çeşitli işleri düzen içinde yürütürler [53], [54], [55], [56].

**Çiftleşme Uçuşu:** Kraliçe arı yuvadan uzaklara çiftleşmek amacıyla uçuş gerçekleştirir. Bu uçuşlar sırasında çiftleşme gerçekleşir. Çiftleşme uçuşu kraliçenin yaptığı bir dans sonrasında başlar. Spermiler kraliçe arının keseciğinde toplanır ve koloninin devamı sağlanır [52], [57].

**Görev Paylaşımı:** Kolonide var olan düzenin sürdürülebilmesi için var olan farklı işler için yeterli bireyin tahsis edilmesi gerekmektedir. Bal arıları kovandaki işler için uzmanlaşmışlardır. Bir arının hayatı bir arının yumurta bırakmasıyla başlamış olur. Yumurtanın yeterince beslendirilmesiyle zamanla larva ve pupaya dönüşür. Larva beslenirken hemşire arılar tarafından larvalara verilen arı sütünün zamanına göre arı ya kraliçe arı olacaktır ya da dişi bir arı olacaktır. Bu devrenin sonunda yumurtadan genç arı ortaya çıkar. Çıkan arıların yetişkin birey olana kadar geçen zaman diliminde her arının beslenme, depolama, bal ve polen elde edilmesi ve dağıtılması, iletişim ve yiyecek arama gibi görevleri bulunmaktadır. Yetişkin arılar yuva çıkışında dururken,

genç arılar ve hemşire arılar beslenme alanında çalışırlar.Şekil 5.2’de arılara ait görev paylaşımı verilmiştir [52], [57].



Şekil 5.2 Arılarda görev dağılımı (Akay [52])

### 5.3 Arıların besin bulma davranışları

Arıların yiyecek arama faaliyetleri araştırmacılar tarafında ilgi çekici bulunan konulardan olmuşlardır. İlk olarak arıların bu faaliyeti Aristo’nun dikkatini çekmiştir. Aristo arıların nektar bulma arayışında görevlendirildiklerini belirtmiştir. Ayrıca arıların yaptığı dans hareketlerini gözlemlemişlerdir.1973 yılına geldiğimizde arı davranışları ve duyuları üzerinde yaptığı çalışmalarla tanınan Avusturya asıllı Karl von Frisch Nobel ödülüne layık görülmüştür. Karl von Frisch arıların mor ötesi ve polarize ışığa karşı hassasiyetlerini ortaya çıkarmış, ayrıca bal arılarının kendilerine özgü bir dans kullanarak kovanda birbirlerine nasıl bilgi aktardıklarını çözmüştür [58].

Bir koloninin başarısı iyi kaynakları eş zamanlı olarak kullanabilme becerisiyle ilişkilidir. Grup zekâsının en olağanüstü örneklerinden birini sergileyen bal arılarındaki amaç en iyiye, en az enerjiyi sarf ederek yani en kısa yoldan ulaşmaktır. Bu süreçte, iyi nektar veya polen kalitesine sahip alanlar daha az enerji harcarak daha fazla arı tarafından

ziyaret edilirken, nispeten düşük kalitedeki kaynaklar daha az sayıda arı tarafından ziyaret edilmektedir. En zengin ve kaliteli nektar içeriğine sahip olan çiçek bölgelerini araştırırken, bölgenin kovana uzaklığı, hava koşulları ve günün hangi vaktinde bulunduğu gibi parametreleri de göz önünde bulundurulmaktadır.

Yapılan araştırmalar bir arı kolonisinin tabiatıta her yönde uzak mesafelere yayılarak verimli nektar ve polen kaynaklarını kullanabilme yeteneğine sahip olduğu yeteneğini göstermektedir. Arıların 9-10 kilometreye kadar varan bir alandaki besinlerin yerlerini kullandıkları gözlenmiştir (Winston [59]).

Tereshko ve Loengarov yiyecek kaynakları, görevli arayıcılar, görevi belli olmayan arayıcılar olmak üzere üç temel öge arılar için yiyecek arama modeli oluşturmuştur (Tereshko ve Loengarov [60] ). Bu öğeleri şöyle açıklanabilir [60], [61] :

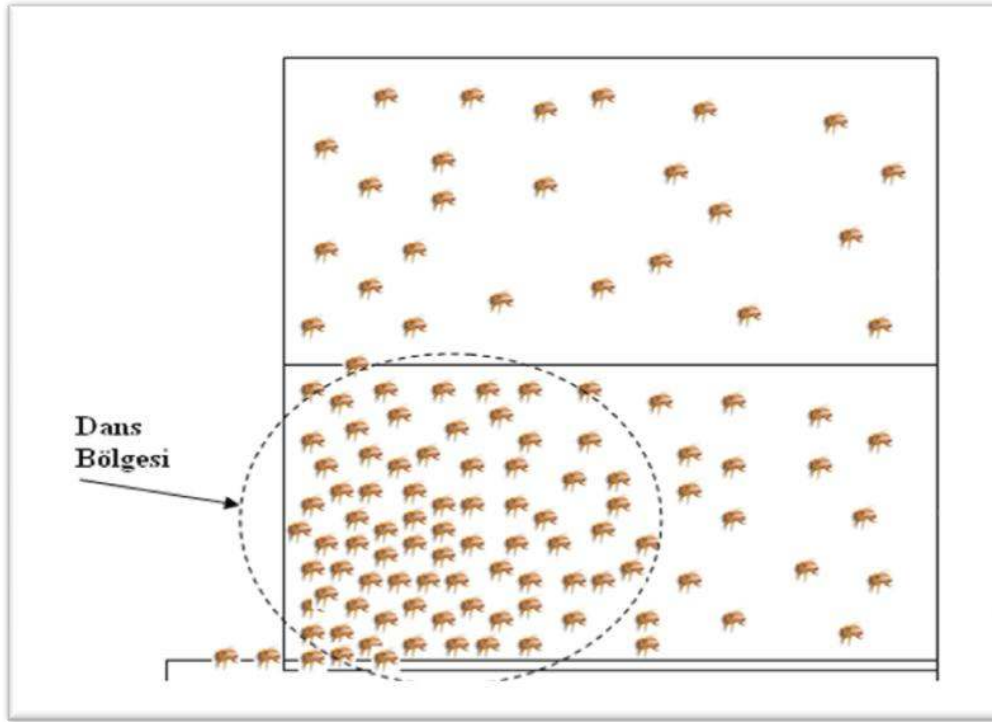
*Yiyecek Kaynakları:* Arıların yiyecek aramak için gittiği kaynaklardır. Kaynağın değeri, kaynağın çeşidi, yuvaya olan uzaklığı, nektar miktarı veya nektarın çıkarılmasının kolaylığı gibi birçok faktöre bağlı olmasına rağmen basitlik açısından sadece kaynağın zenginliği tek ölçüttür.

*Görevi Belirli İşçi Arılar:* Daha önceden belirlenmiş kaynaklardan toplanan yiyeceğin kovana getirilmesi, gittikleri kaynağın konum ve kalite bilgilerini kovadaki diğer arılarla paylaşmaktır işi ile görevli arılardır.

*Görevi Belirli Olmayan İşçi Arılar:* Görevi belli olmayan iki çeşit işçi arı vardır. Bunlar rastgele kaynak arayan kâşif arılar, kovanda bekleyen ve görevli arıları izleyerek bu arılar tarafından gelen bilgiye göre yeni kaynaklara yönelen gözcü arılardır.

Araştırma süreci izci arılarının verimli çiçek bölgelerini araştırması için gönderilmesiyle başlar. İzci arılar rassal olarak bir bölgeden diğerine, bir çiçekten diğerine uçarak, nektar toplarken bölgeleri de kaliteleri açısından analiz eder. Her bir arı araştırma gezisinden döndükten sonra topladıkları nektar ve poleni boşalttıktan sonra ziyaret ettikleri kaynak belirli bir kalite sınırının üzerinde ise *dans bölgesine* giderek geldikleri bölgeleri diğer arılarında seyrettiği bir dans anlatırlar. Arıların dans yapmalarının nedeni sağır olmalarıdır. Bu nedenle birbirleriyle sesli bir iletişim kuramazlar. Şekil 5.3 'de kovanda dans bölgesinin şematik gösterimi bulunmaktadır.

Şekil 5.3’de görüldüğü üzere dans bölgesi bir kovandaki arı yoğunluğunun en fazla olduğu bölgedir. Bu yüzden arıların dans ederken bilgi aktarımında izleyici bulma şanslarının en yüksek olduğu ve kovanda bekleyen arıların da hem kaliteli kaynak bilgisine ulaşabileceği hem de bunlar arasından tercih yapabileceği bir yerdir. Çünkü bu danslar kaynağın kovana olan uzaklığı, doğrultusu, zenginliği gibi bilgiler içermektedir. Araştırmaya çıkacak arılar dans yapan arılar etrafında toplandığından bu bölgelerde arı yoğunluğu artmaktadır.

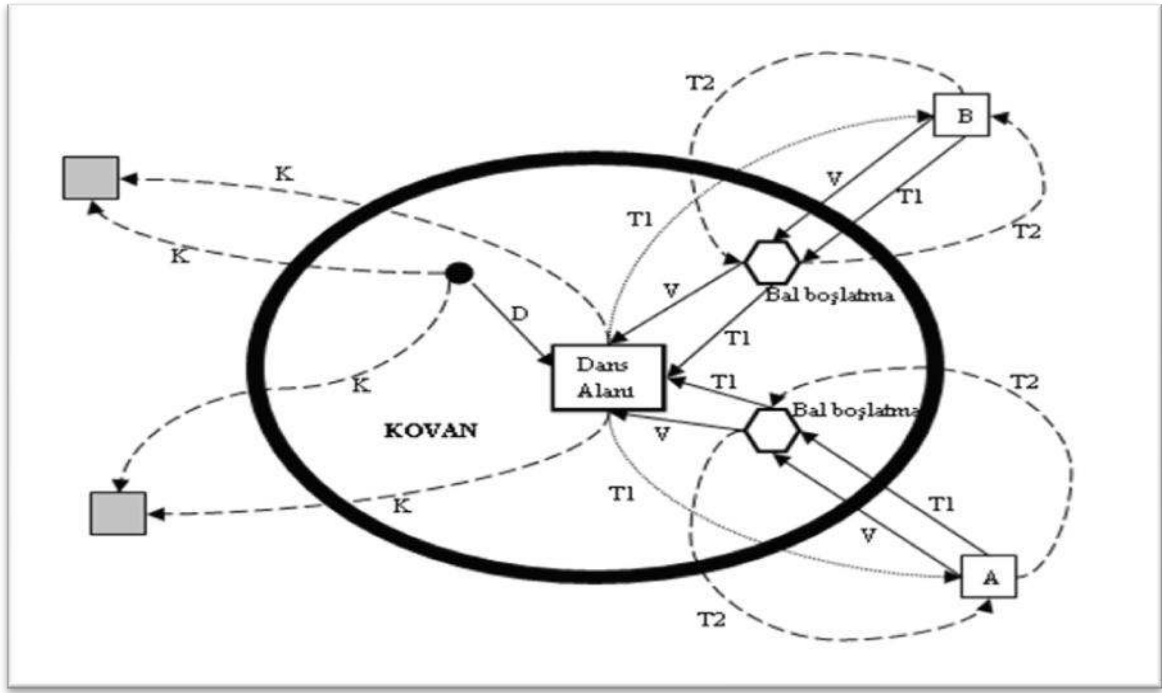


Şekil 5.3 Kovandaki dans bölgesinin temsili gösterimi

Arıların yiyecek arama ve getirme davranışları Şekil 5,4’te gösterilmiştir. A ve B bulunmuş kaynaklar olarak farz edilmiştir. Başlangıçta görevi belli olmayan ve kaynak bilgisinden yoksun arı aramaya başlayacaktır [52], [62], [63],[64].

Bir arı için iki seçenek vardır.

- i) Bir kâşif olabilir ve içgüdüyle veya yüzeysel ipuçlarıyla yuva etrafındaki yiyecekleri kendiliğinden aramaya baslar. (şekildeki K)
- ii) Boş olabilir ve sallanış dansını izler ve yiyecek kaynağını aramaya baslar. (şekildeki D)



Şekil 5.4 Bal arılarının yem arama davranışları [63]

Kaynaklara giden arılar buldukları kaynaklardan nektar getirmeye başlarlar. Böylece bu arılar görevli bir arı haline gelmişlerdir. Nektarı kovana getiren arı için bundan sonra üç seçenek vardır:

- i)Mevcut kaynağı terk edip bağımsız hale gelebilir, dans alanında gözcü arı olabilir. Bu arı V ile gösterilmiştir.
- ii)Aynı kaynağa dönmeden önce dans ederek nektar kaynağının yeri ve miktarı hakkında bilgi paylaşımında bulunabilir. Bu arı T1 ile gösterilmektedir.
- iii)Bilgi paylaşımında bulunmadan kaynaktan nektar getirmeye devam edebilir. Şekilde bu arı T2 ile gösterilmektedir.

Son seçenek bir arı için sosyal davranış olarak görünmese de aslında kaynaktan nektar getiren yeteri kadar arı varsa geçerli bir yoldur.

Kendini organizasyonun temel özellikleri arılarda aşağıdaki adımlara dayanır:

- i) Pozitif Geri besleme: Yiyecek kaynaklarındaki nektar miktarı gözcü arıların onları ziyaret sayısı ile gelişir.

ii) Negatif Geri besleme: Bir yiyecek kaynağının araştırma süreci terk edildiğinde arılar durdurulur.

iii) Dalgalanmalar: Kâşifler, yeni yiyecek kaynaklarının keşfi için rast gele bir arama sürecini tamamlar.

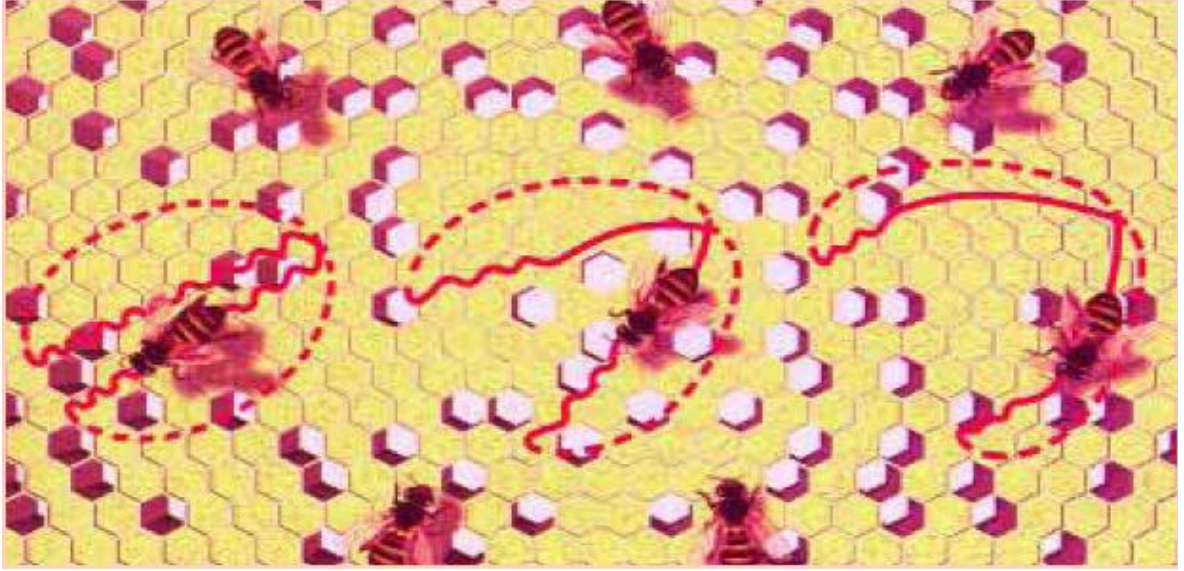
iv) Çoklu Etkileşim: Arılar dans alanında yiyecek kaynağı pozisyonları hakkındaki bilgilerini paylaşırlar [62], [63].

Arı yiyecek kaynağını keşfettikten sonra kovana döner ve diğer arıların dikkatini çekecek şekilde sürekli olarak belli hareketleri tekrarlamaya başlar. Arının bu genel davranışlarından yiyecek kaynağı ile ilgili tüm bilgiler elde edilebilir. Mesela, polen toplamış olan bir arı kovana döndüğünde sadece yükünü arkadaşlarına devredip geri uçarsa *arının faydalandığı kaynak bilinen bir kaynaktır veya verimsizdir* anlamına gelmektedir. Suyun kısıtlı olduğu zamanlarda ise bu dans su kaynağının yerini göstermek için de kullanılır (Yahya [65]).

Arı dansının iki ayrı şekli vardır. Dansın biçimi, yiyecek kaynağının uzaklığına göre değişiklik gösterir. Bunlardan ilki Daire dansı olarak adlandırılan dans en sık rastlanan danstır. Yalnızca işçilere yuvanın yakınlarında 15 metreden daha yakın mesafede bir kaynak olduğunu bildirir. Dans sırasında yakında bir kaynak keşfeden işçi arı ilk önce yuvanın içindeki işçilere nektar verir ve ardından dansa başlar. Diğer arılar daha sonra bu dansa eşlik ederler. Dansçı sürekli küçük daireler çizer. Her 1-2 turdan sonra, bazen de daha sık aralıklarla ters döner. Saniyelerce ya da bir dakika kadar süren bu dans 20 kadar tur olur. Dansın uzun sürmesi ve tur sayısının artması besin kaynağının kovana uzak olduğunu gösterir. Sonra tekrar dansçı ile yuvadaki arılar arasında bir nektar değişimi olur. Dans sona erdiğinde, dans eden arı başka bir besin aramak üzere yuvayı terk eder (Winston [59]).

Arılar yuvadan 15 metre kadar uzaklıktaki besin kaynakları için daire dansını kullanırken, 25-100 metre arasındaki besin kaynakları için de bir geçiş dansı olan sallanma dansını kullanırlar. Bundan başka balarıları yuvadan 100 metreden daha uzak kaynaklar için kaynağın uzaklığını, yönünü ve niteliğini bildiren kuyruk dansı ile iletişim kurarlar. Bu dans aynı zamanda *8 rakamı dansı* olarak da adlandırılır (Frish [52]). Bu dans işçiler adım atarken bir yandan da karınlarını titretirler. Hareketlerinin

karakteristik şekli 8 rakamına çok benzer. Tipik bir kuyruk dansında arı kısa mesafe için dümdüz bir hat üzerinde hareket eder. Vücudunu saniyede yaklaşık olarak 13-15 defa bir yandan diğer yana doğru sallar (Yahya [65]). Şekil 5.4’de arıların yiyecek kaynağının uzaklığı hakkında bilgi vermek için yaptıkları, dalgalı çizgilerle gösterilen 8 dansı görülmektedir.

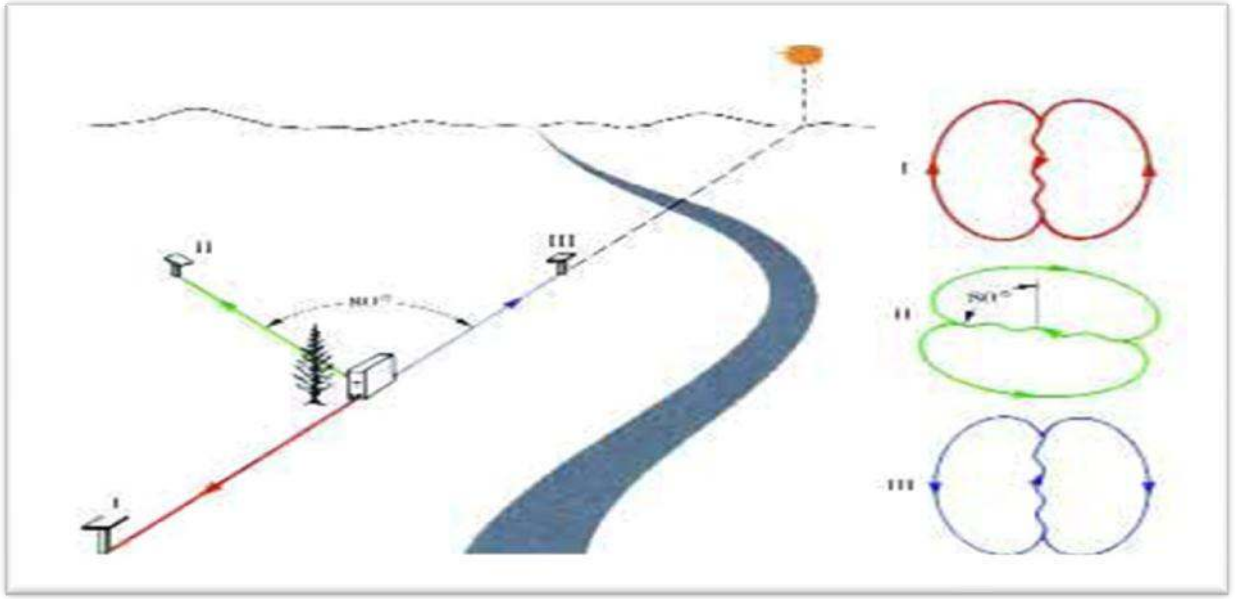


Şekil 5.5 Arıların kovanda yaptığı 8 dansı

Arının düz olarak geçtiği bu yolun, kovanı yukarıdan aşağıya doğru hayali kesen dikmeye yaptığı açı, besin kaynağının güneşe olan açısını verir. Dans ederken yere tam dik gelen üst kısım sembolik olarak güneşi göstermektedir. Eğer arı kovaniyla besin kaynağını ve kovanla güneşin hemen altındaki ufuk çizgisini birleştiren bir çizgi çizilirse, iki çizgi arasında oluşan açının sallanma dansının açısıyla aynı olduğu görülür.

Şekil 5.6 arıların yiyecek kaynağının yönünü nasıl belirlediklerini göstermektedir. Eğer besin kaynağı tam Güneş yönünde veya tam aksi yönde ise dansın orta kısmı yere dik gelecek şekilde olur. Bu I numaralı şekilde gösterilmektedir. Eğer dansın düz olarak verilen doğrultusu, yerçekimi doğrultusu ile 80°C'lik bir açı yapıyorsa bu, yiyecek kaynağının Güneş'in 80°C sağında olduğunu gösterir. Bu II numaralı şekilde gösterilmektedir. Eğer Arı düz yolu yukarı doğru alıyorsa yiyecek kaynağı tam Güneş yönünde, aşağı doğru alıyorsa kaynak Güneş'in tam aksi yönünde demektir. Bu III numaralı şekilde gösterilmektedir.





Şekil 5.6 Arıların danslarıyla yiyecek kaynağının yönünü gösterimi (Yahya [65])

Bu dansı izleyen işçiler besin kaynaklarının yerini tespit edebilirler. Uzaklığı belirten dansın özelliklerinden biri de, her 15 saniyedeki dönüş sayısı ile ölçülen dans temposu ve düz bir hat boyunca yapılan sallanma hareketleri ve vızıldamalardır. Dansın temposu, daha uzaktaki besin kaynakları için yavaşlar, yakındaki besin kaynakları için hızlanır. Yine dansın zamanı, daha uzak mesafedeki kaynaklar için artar.

Kaynağın yönünü bilmek tek başına bir işe yaramaz. Çünkü işçi arıların balözünü toplayabilmeleri için, ne kadar uzağa gitmeleri gerektiğini de bilmeleri gereklidir. Kovana dönen arı, diğer arılara, yine belirli vücut hareketleriyle çiçek polenlerinin bulunduğu uzaklığı da anlatır. Dans boyunca diğer işçiler, tarifi yapan arının etrafında kümelenip her hareketini takip ederler. Ayrıca dansçının titreşen karnına antenleri ile dokunurlar. Çünkü arının havada oluşturduğu kesintili akım besin kaynağının uzaklığını bildirir. Arının gövdesinin alt kısmını sallaması sayesinde hava akımları oluşur. Diğer arılar da antenleri ile bu akımları algılar ve gidecekleri besin kaynağının uzaklığını bu sayede tespit ederler (Düğenci [66]).

Arılara gerekli olan bilgilerden bir tanesi de kaynakta bulunan besinin niteliği ile ilgilidir. Bu bilgiyi de dansı yapan toplayıcı arının üzerine sinen koku sayesinde edinirler (Yahya [65]). Arılar, eğer buldukları kaynak çok zenginse coşkulu bir şekilde dans ederler. Eğer kaynak yakındaysa yuvarlak dans adı verilen danslarını yaparak kaynağın yerini tarif

ederler. Daha uzaktaki kaynaklar içinse 8 şekilli danslarını yaparlar ve buna titreşim hareketlerini de eklerler.

Besin kaynağının başına çok fazla arı toplanması kovanda dans eden arıların sayısı doğrudan bağlantılıdır. Tüm kovan tek bir arının dansıyla harekete geçmez. Önce bir grup arı öncü olarak gider. Bu grup döndüğünde dans ediyorsa daha fazla arı kaynağa doğru yönelir. Buldukları kaynak ne kadar iyi ise, o kadar daha uzun süre dans ederler ve daha fazla takipçi arı toplarlar. Böylece koloninin toplayıcı takımının dikkati sürekli olarak en verimli besin kaynağında olur.

Bu kontrol aslında başlangıçta izci arıların yaptığından farklı değildir. Kaynak hala aynı kalitede ise diğer arılardan bir kısmı da kovana döndüğünde aynı kaynak için dans ederler. Herhangi bir sebepten meydana gelebilecek değişiklikler bu iletişim sistemi sayesinde hemen algılanarak, kaynağa gönderilecek arı miktarı artırılabilir, azaltılabilir veya kaynak tamamen terk edilir.

Bulunan besin kaynağının verimsiz olması arıların yaptığı dansı isteksiz olur ve daha kısa sürer. Bu durum kovandaki diğer arılara da yansır, dansçıların başına toplanan arılar kısa bir süre içinde dağılırlar. Bu durumda yeni bir ekip besin aramak için çıkar (Yahya [65]).

Yukarıda anlatılan sistem sayesinde koloninin ihtiyaç duyduğu nektarı ve poleni hızlı ve verimli bir şekilde toplanır. Çünkü arılar çiçekleri ziyaret ederken aynı zamanda kaynağın yiyecek seviyesini kontrol eder. Böylece kovana geri döndüklerinde kaynak için dans edip etmemeye karar verirler. Eğer yiyecek kaynağı hala yeterince iyi ise her seferinde daha fazla arının ilgisini çekecektir. Böylece koloninin ihtiyaç duyduğu nektar, polen gibi bileşenler en hızlı ve verimli bir şekilde toplanabilecektir.

#### **5.4 Arı Algoritması**

Arı algoritması ilk olarak 2005 yılında pham tarafından ortaya konulmuştur. Algoritma, doğal hayattaki arıların besin toplama aşamasındaki davranışlarından esinlenerek optimizasyon problemleri için optimal sonucu elde etmek için kullanılmaya başlanmıştır [51].

Literatürde arı algoritması, sürekli optimizasyon problemlerinin çözümünde, yapay sinir ağlarının eğitiminde, atölye tipi çizelgeleme gibi kombinatoriyel optimizasyon problemleri için kullanılmıştır.

Her çözüm arayan algortmada olduğu gibi arı algoritmasının uygulanması için başlangıçta tanımlanması ve değerlerinin belirlenmesi gereken parametreler bulunmaktadır.

Bu parametreler:

-Kâşif (çözüm arayan)arı sayısı: **n**

-n bölgeden seçilecek bölge sayısı(etrafında araştırma yapmak üzere gönderilecek iyi sonuçlar elde eden, *uygunluk değeri yüksek* araştırma bölge sayısı): **m**

-Araştırma bölgesi (m) içerisindeki en iyi elit arı sayısı: **e**

-Diğer araştırılacak seçilmiş bölge sayısı: **m-e**

-Elit ( e ) arısı etrafına gönderilecek arı sayısı: **nep**

-Diğer bölgesi arılarının (m-e) etrafına gönderilecek arı sayısı: **nsp**

-Araştırma yapılmak üzere seçilen arının ne kadarlık çevresinde araştırma yapılacağına belirlendiği çevre büyüklüğü: **ngh**

-Durdurma kriteri(iterasyon sayısı, fonksiyonun uygunluk değeri vb.)

Parametreler belirtildikten sonra algoritmanın nasıl işlediğine bakalım. İlk olarak, çözüm uzayında n adet arı rassal olarak yerleştirilir. İkinci adım olarak, problemin amaç fonksiyonuna uygun olarak arıların uygunluk değerleri hesaplanır. Üçüncü adım belirlenen durdurma kriterinin (iterasyon sayısı ya da istenilen çözüme ulaşıp ulaşılmadığı) sağlanıp sağlanılmadığı ölçülür. Eğer durdurma kriteri sağlanıyorsa işleme son verilir; aksi takdirde dördüncü adıma geçilir. Dördüncü adım, uygunluk değerine göre arılar sıralanır. Uygunluk değeri en iyi olandan başlamak üzere, etrafında araştırma yapılacak arılar, araştırma bölge arısı m olarak belirlenir. Araştırma bölgesi belirlenerek komşuluk araştırma mekanizmasının uygulanması içindir. Bu adımda en iyi elit sayısı (e) ve diğer araştırılacak seçilmiş bölge sayısı (m-e) bellidir. Beşinci adımda, araştırma bölgesi arıları etrafına yeni izci arılar gönderilir. Elit arıların etrafına daha

fazla olmak üzere, araştırma bölgesindeki tüm arıların etrafına izci arılar gönderilir. Bu adımda elit arıların yanına daha çok arı gönderilmesinin nedeni *iyi kaynaklar bulmuş olan arıların etrafında daha iyi kaynaklar bulunabilir* yaklaşımından yola çıkarak çözüme yakın alternatiflerin etrafında araştırmayı derinleştirmektir. Bu yerel araştırma da, arı algoritmasının temel kavramlarından bir tanesidir. Altıncı adımda, seçilen komşuluğu araştırılan her bir araştırma bölgelerinin en iyi uygunluk değerine sahip arıları seçilir. Yedinci adımda, yerel araştırma devam ederken, diğer yanda da genel araştırmalar yapmak için başka bölgelerde de çözümler aranır. Bu kısımda ise kalan arılar rassal olarak çözüm uzayında atanmaya atanır. Sonra rassal atanan arılar ve seçilen en iyi arılar tekrar durdurma kriterine göre değerlendirilir. Bu uygulama durdurma kriteri sağlanıncaya kadar devam ettirilir.

### 5.5 Bir Arı Algoritması Uygulaması

Bu bölümde anlatacak uygulama pham kitabında verdiği örnek uygulamadır. Bu uygulamada bir matematiksel fonksiyonda en yüksek değerin bulunması yönünde arı algoritmasının nasıl kullanıldığı anlatılacaktır. Şekil 5.7’de arı algoritmasının uygulanacağı matematiksel fonksiyon verilmiştir.

Problem belirlendikten sonra önce parametrelerin de belirlenmesi gerekmektedir. Keşif arı sayısı 10, Etrafında araştırma yapmak üzere gönderilecek uygunluk değeri yüksek araştırma bölge sayısı 5, elit bölge sayısı 2, elit arısı etrafına gönderilecek arı sayısı 4, diğer bölgesi arılarının etrafına gönderilecek arı sayısı 2 olsun.

Şekil 5.8’de fonksiyon üzerinde rassal olarak 10 nokta seçilmiştir. Daha sonraki adımda uygunluk değeri bulunmuştur. Araştırılacak bölge sayısı 5 olduğundan şekil 5.9 da en iyi 5 değer seçilmiş ve en iyi 2 tanesi elit arı olarak belirtilmiştir. Bu aşamadan sonra daha uygun çözümün bulunması için komşuluk arama bölgelerinin seçilmesi gerekmektedir. Ele alınan problemde araştırma bölgesindeki uygun değerler için belirlenen komşuluk bölgeleri şekil 5.10’da gösterilmiştir.

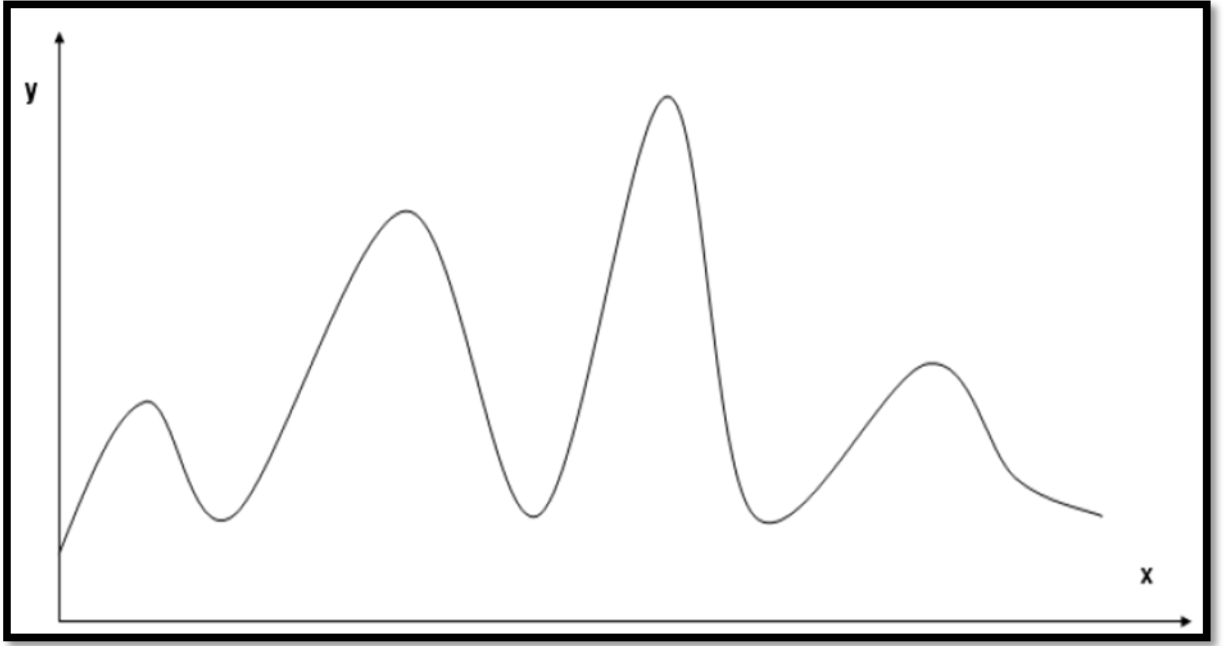
Daha sonraki aşamada araştırma bölgesi arıları etrafına komşuluk arama aralığının sınırlarına göre yeni izci arılar gönderilir. Ele alınan uygulamada elit arıların etrafına 4 tane, diğer bölgesi arılarının (m-e) etrafına gönderilecek arı sayısı 2 tane olarak

belirlenmiştir. Elit arıların yanına daha çok arı gönderilmesinin nedeni iyi kaynaklar bulmuş olan arıların etrafında daha iyi kaynaklar bulunabilir yaklaşımından yola çıkarak çözüme yakın alternatiflerin etrafında araştırmayı derinleştirmektir. Bu uygulama arı algoritmasını diğer arama algoritmalarından ayıran bir özelliktir. Bu adım şekil 5.10'da şematik olarak gösterilmiştir.

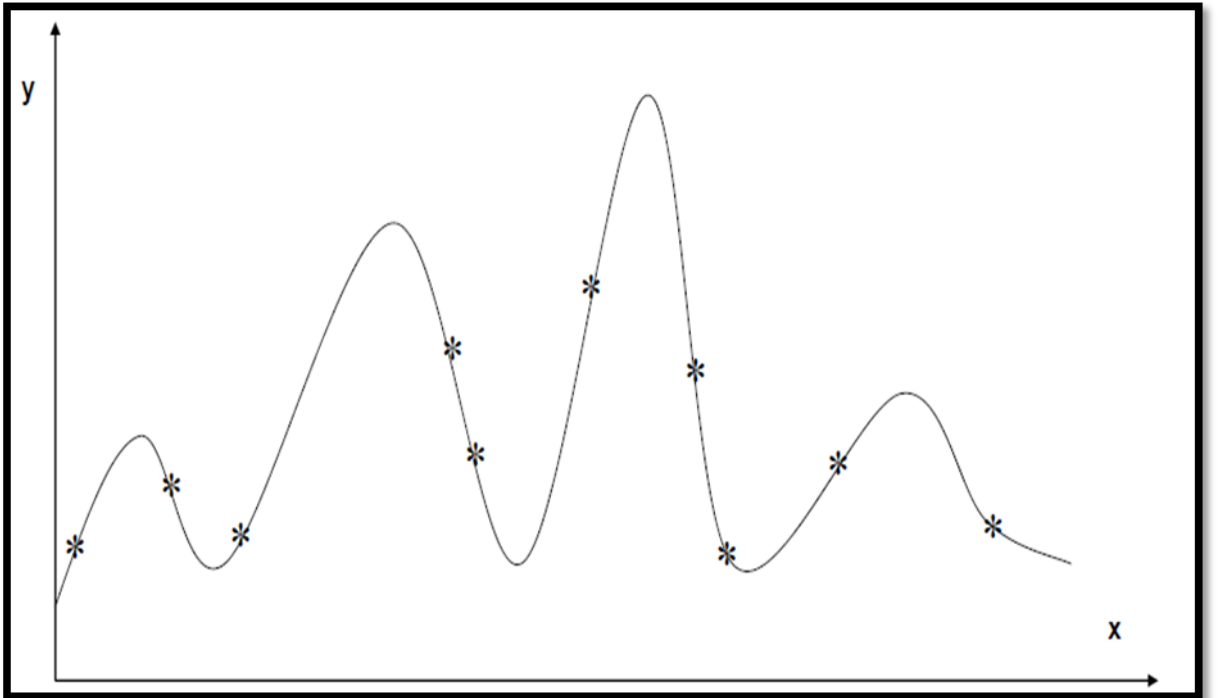
İzci arıların ulaştığı noktaların yeni uygunluk değerleri hesaplanır. Daha sonra her bir araştırma bölgesi içindeki noktanın etrafındaki en uygun izci arı belirlenir. Bu adımdan sonra kalan arıların (10-5) rassal olarak tekrar atanması gerçekleştirilir. Bundaki amaç yerel aramanın dezavantajı olan optimum noktayı göz ardı etmeme durumudur. Diğer araştırılmayan noktalarda daha uygun değere rastgelme olasılığının düşünülmesidir. Bu noktalarda uygunluk değerleri hesaplanır. Bu adımlar sırasıyla şekil 5.11 ve 5.12'de gösterilmiştir.

Yapılan işlemlerin ardından durdurma kriterine göre uygunluk değerleri incelenir. Amacımız fonksiyonun en yüksek değerini veren noktayı bulmaktır. Bulunan uygunluk değerlerine göre istenen nokta elde edilememiştir. Bu aşamada elimizde var olan oluşmuş yeni popülasyonda tekrar algoritmayı uygulamak gerekmektedir. Bunun için tekrar yeni bir araştırma bölgesi ve elit arı seçimleri yapılır. Şekil 5.13'de bu aşamayı incelediğimizde oluşan yeni araştırma bölgesinde kalan arılardan rassal olarak bulunan noktalardan bir tanesinin de dâhil olduğunu görmekteyiz.

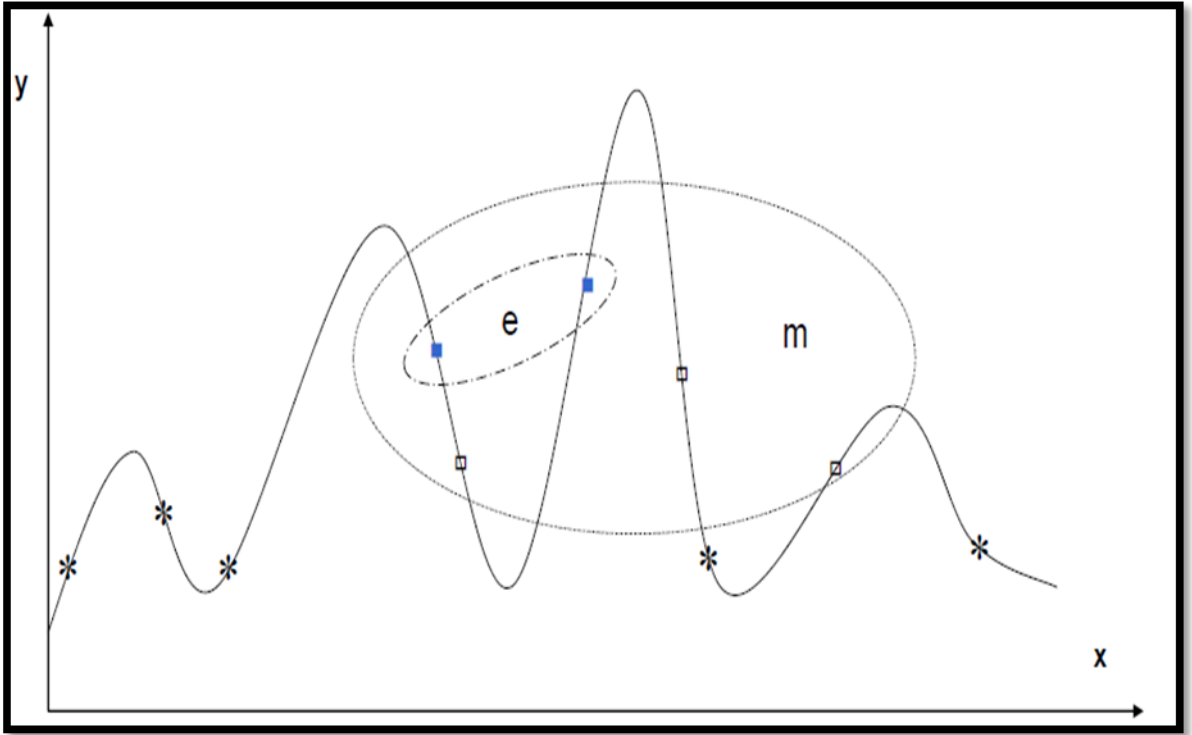
Yukarıdaki aşamadan sonra tekrar algoritmaya uygun bir şekilde durdurma kriteri sağlanıncaya kadar devam ettirilir. İncelenen örnek için şekil 5.14'deki optimum nokta araştırma kümesi örneği gösterilmiştir. Arı algoritması rastgele ve bulunan en iyi çözüm üzerinde beraber ilerlemeyi sağlayarak uygun çözümler üretmektedir. Önemli olan doğru parametre değerlerini ve komşuluk arama bölgesini belirlemektir.



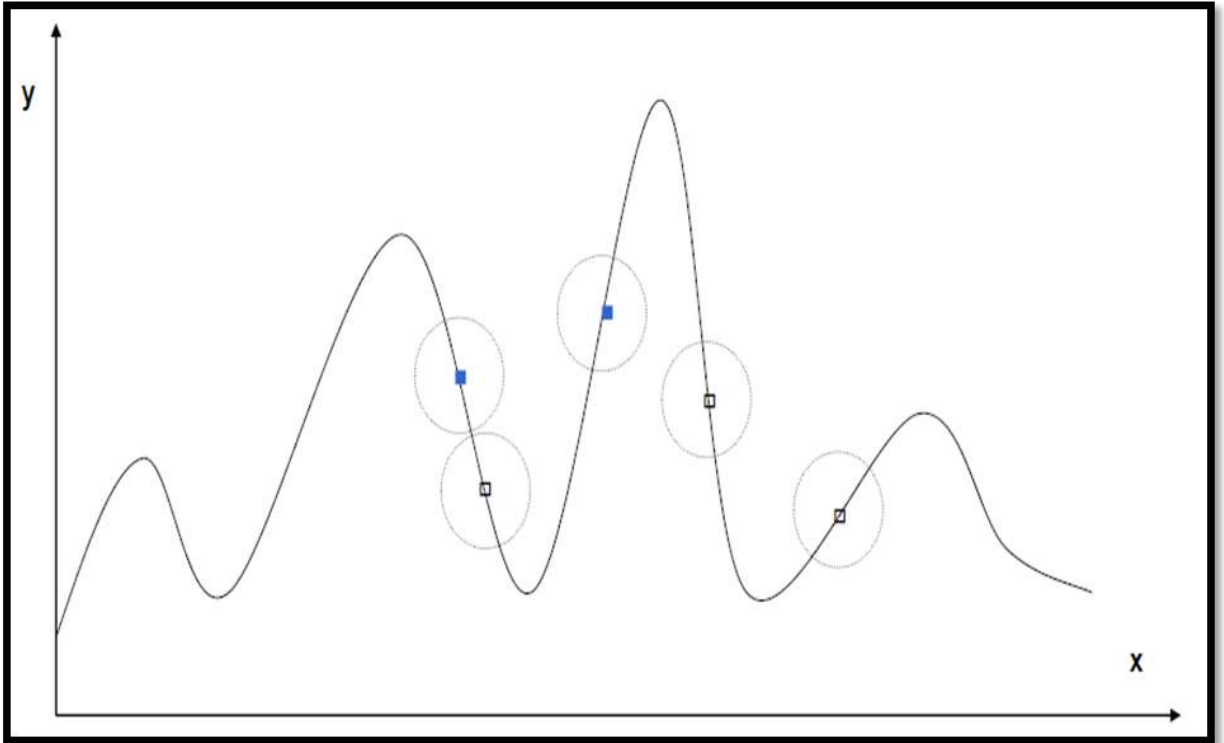
Şekil 5.7 Arı algoritmanın uygulanacağı fonksiyon



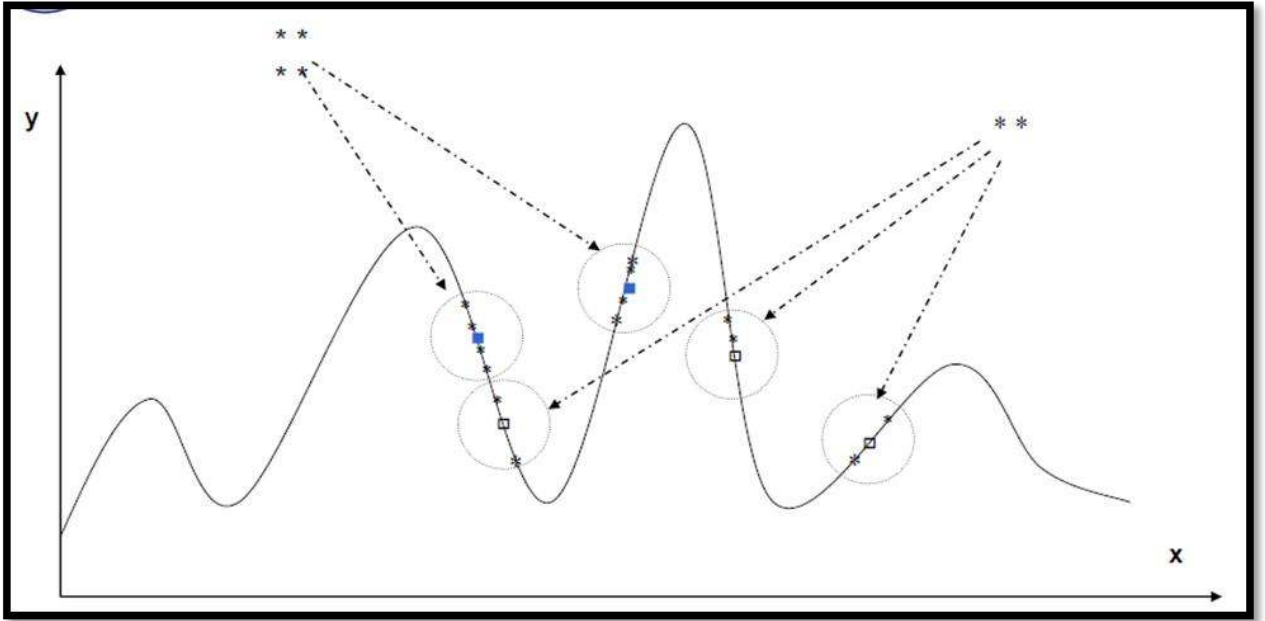
Şekil 5.8 Rassal olarak  $n$  sayıda birey üretimi ve uygunluk değerinin hesaplanması



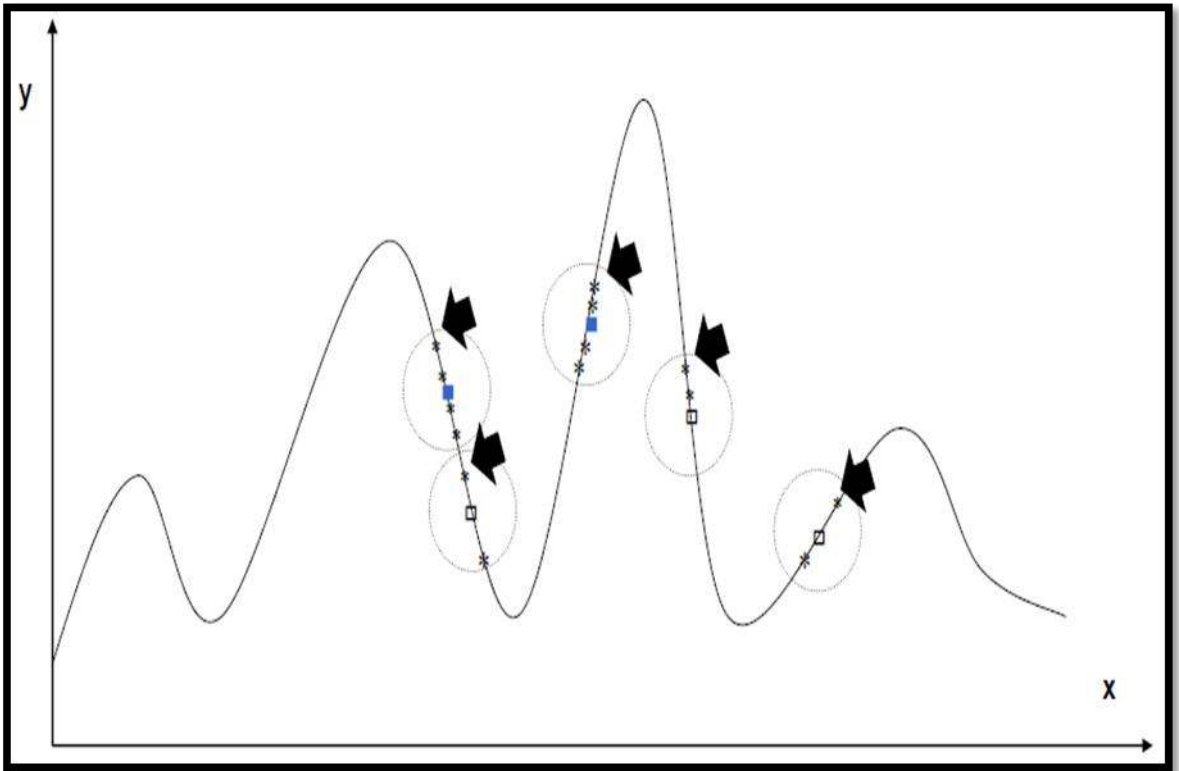
Şekil 5.9 Araştırma bölgesinin belirlenmesi ve elit arıların seçimi



Şekil 5.10 Komşuluk çevresinin aralığının belirlenmesi

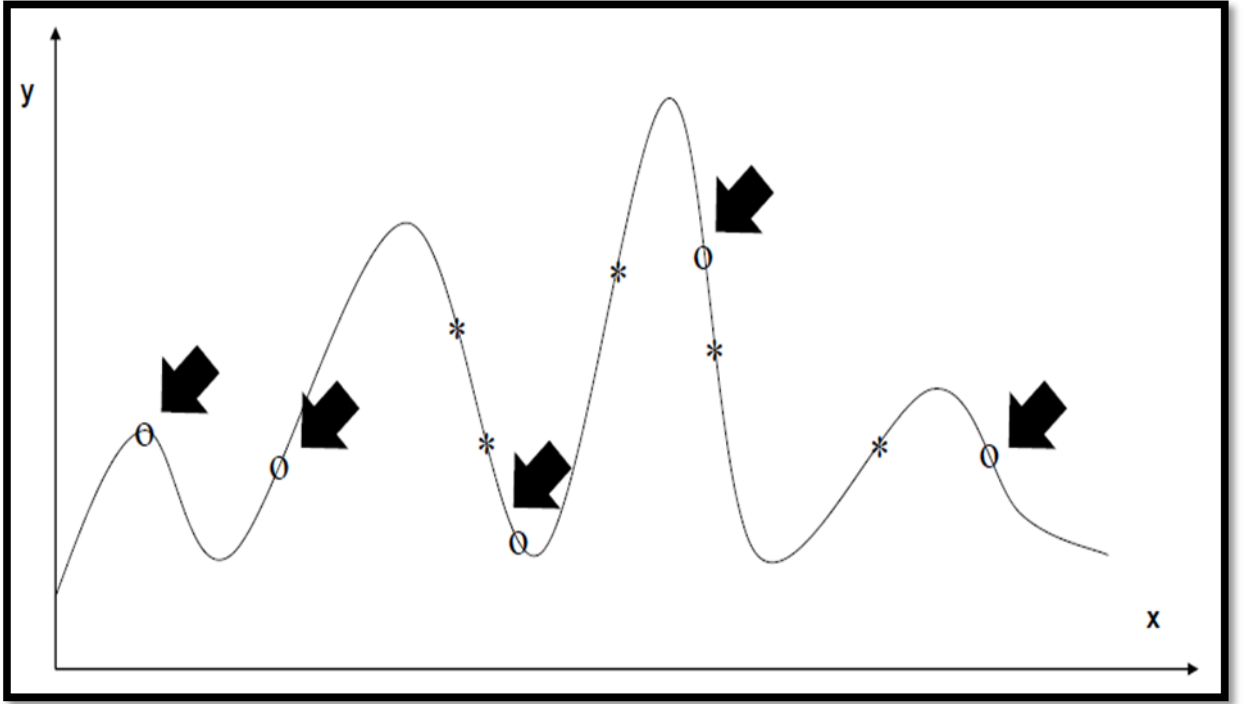


Şekil 5.11 Araştırma bölgesine izci arıların gönderilmesi

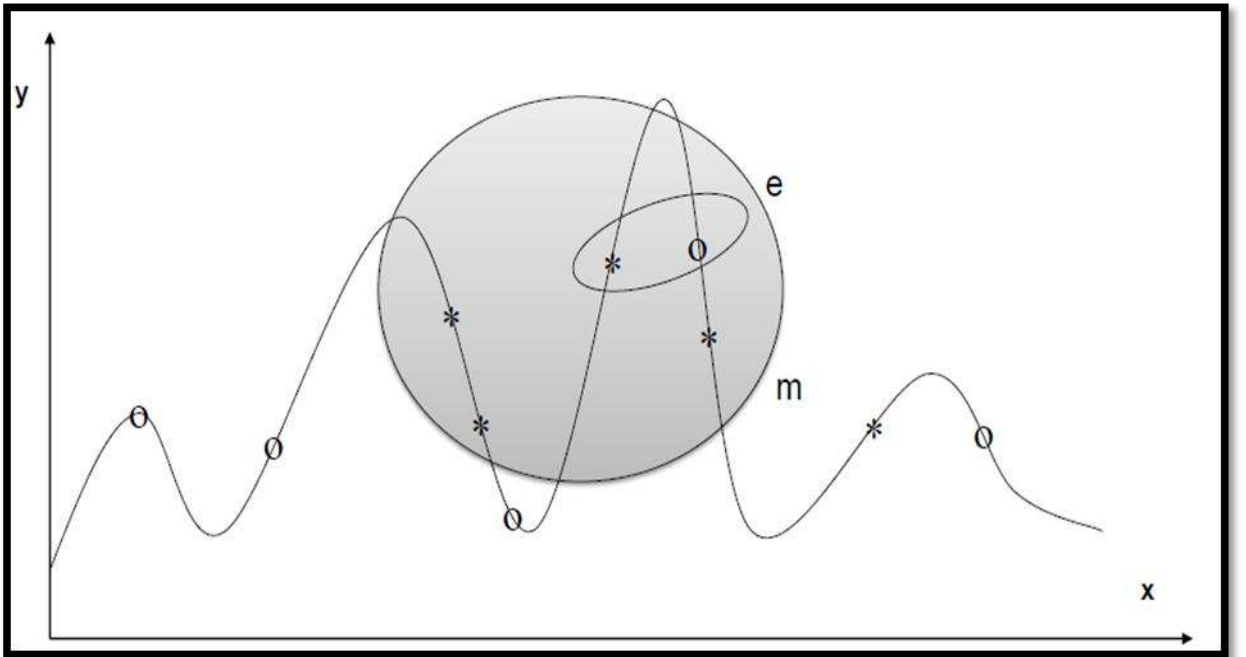


Şekil 5.12 Her bir araştırma bölgesindeki en uygun arıların seçimi

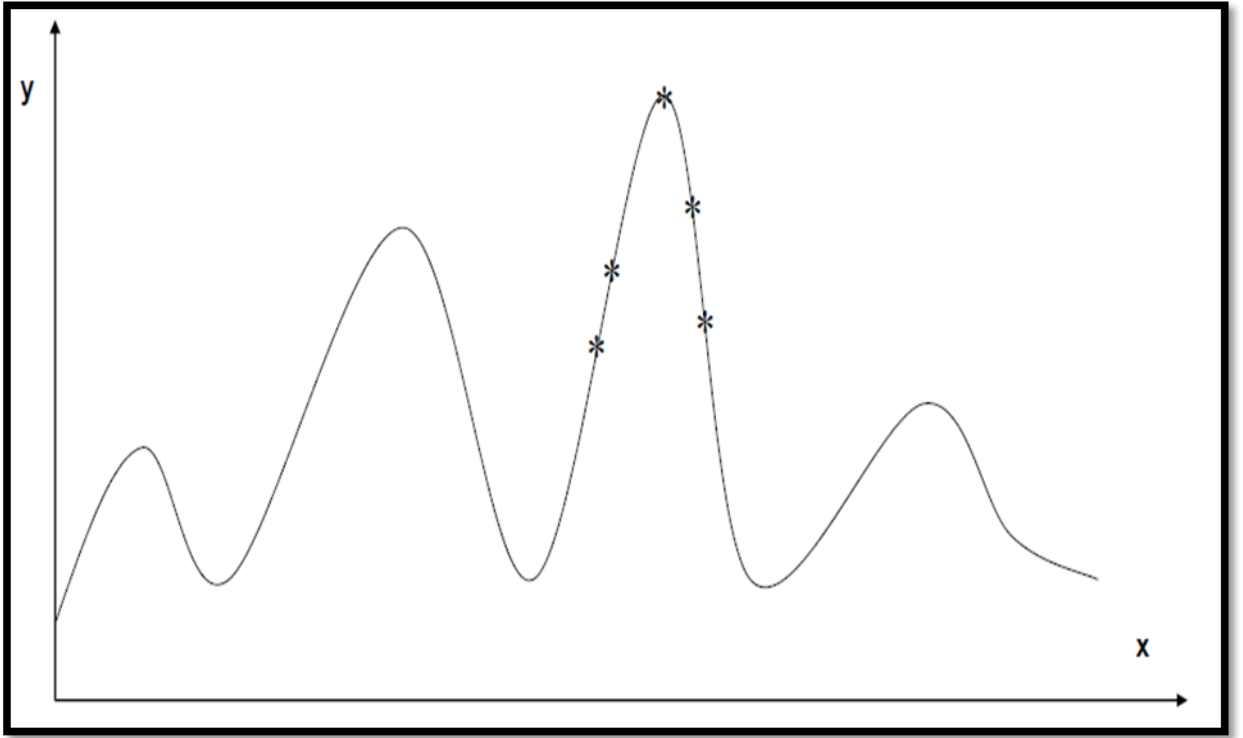




Şekil 5.13 Kalan arıların rassal olarak atanması ve uygunluk değerlerinin hesaplanması



Şekil 5.14 Oluşan yeni popülasyonda araştırma bölgelerinin belirlenmesi ve elit arıların seçimi



Şekil 5.15 Fonksiyonun optimum araştırma bölgesi

---

### SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNDE ARİ ALGORİTMASININ UYGULANMASI

#### 6.1 Uygulamanın Varsayımları

Ele alınan problemler akış tipi iş çizelgeleme problemi oldukları için bu problem tipi için geçerli olan varsayımlar aynen kabul edilmiştir. Baker'e göre akış tipi iş çizelgeleme için genel varsayımlar şunlardır [62]:

1.  $n$  adet çok işlemlili iş kümesi, işlenmek üzere sıfır anında hazırdır.
2. İşlerin hazırlık süreleri sıraya bağlı değildir ve işlem sürelerine dâhildir.
3. İşlem süreleri önceden biliniyordur.
4.  $m$  farklı makine sürekli çalışabilir.
5. İşler kesilemez.

#### 6.2 Uygulamanın Yapısı ve Oluşturulması

Çalışmada sıralı akış tipi çizelgeleme problemlerinin arı algoritmasıyla çözüm performansı incelenmiştir. Ayrıca arı algoritmasında farklı komşuluk yapıları incelenerek en uygun ölçütleri bulunmaya çalışılmıştır.

Çalışmada performans kriteri olarak, en son işin son makinedeki tamamlanma zamanı ( $C_{max}$ ) kullanılmıştır. Uygulamada kullanılan problemler  $n$ -iş,  $m$ -makine olarak  $n/m/P/C_{max}$  gösterimi ile gösterilebilir.

Tamamlanma zamanı aşağıdaki denklem 5'deki gibi hesaplanabilir. Eğer, i. işin j. makinedeki işlem zamanı  $P(i,j)$  ve iş sırası  $(J_1, J_2, \dots, J_n)$  ise,

$$C(J_1,1) = P(J_1,1) \quad (1)$$

$$C(J_i,1) = C(J_{i-1},1) + P(J_i,1) \quad i = 2, \dots, n \quad (2)$$

$$C(J_1,j) = C(J_1,j-1) + P(J_1,j) \quad j = 2, \dots, m \quad (3)$$

$$C(J_i,j) = \max ( C(J_{i-1},j), C(J_i,j-1) ) + P(J_i,j) \quad i = 2, \dots, n ; j = 2, \dots, m \quad (4)$$

$$\text{Toplam tamamlama zamanı } C_{\max} = C(J_n,m) \quad (5)$$

Arı algoritmasının sıralı akış tipi iş çizelgeleme problemlerine uygularken daha önceden yapılmış bir çalışmadan faydalanılacaktır. Bu çalışma Mehmet tufan'ın yaptığı çalışmadır (Körez [50]). Bu çalışmada sıralı akış iş çizelgeleme problemi Smith'in geliştirdiği algoritmayı (SPD) genetik algoritma (GA) ve tavlama benzetimiyle (TA) ile kıyaslamıştır. Yapılan çalışmada genetik algoritmanın çözüm performansı geliştirilen SPD algoritmasına göre oldukça başarılı bir oran elde etmiştir. Bunun aksine geliştirilen tavlama benzetimi performansı düşük çıkmıştır (Körez [50]).

Çalışmada smith ve arkadaşlarının sıralı akış tipi çizelgeleme problemleri için tanımladıkları iki ayrı problem çeşidi ele alınmıştır. Bu problemlerden ilki maksimum işlem zamanının ara makinelerde olduğu genel durum problemleridir. İşlem zamanlarını içeren sıra matrisinde, işler içerisinde en büyük işlem zamanlarının ilk ve son makinede değil ara makinelerden birinde oluşmaktadır. Daha önceden bahsedildiği gibi bu problem tipi için dal-sınır algoritmasına benzer bir çözüm yöntemi önerilmiştir. İkinci problem çeşidi en büyük işlem zamanlarının ilk ya da son makinede olduğu özel problem çeşididir. Smith ve diğerleri uygulama açısından görece daha kolay olan işleri artan ya da azalan biçimde sıralamaya dayalı bir çözüm yöntemi önermişlerdir.

Çalışmada literatürdeki örnekler göz önünde bulundurularak genel ve özel durum problemleri için n-iş, m-makine şeklinde (7x4, 7x5, 7x8, 7x10, 7x15, 10x4, 10x5, 10x8, 10x10, 10x15) 20 farklı iş-makine problemleri oluşturulmuştur. Tüm problem çeşitleri için işlem zamanları sıralı akış tipi çizelgeleme varsayımlarına uygun şekilde [1-25] aralığında üniform olarak oluşturulmuştur. Problemler öncelikle Smith, Panwalkar,

Dudek algoritması (SPD) ile daha sonra genetik algoritma ve tavlama yöntemi ile çözülerek optimum sıralama ve tamamlanma zamanı bulunmuştur.

Yapılan bu çalışmada bahsedilen çalışmanın verileri alınarak arı algoritması uygulanmıştır. Böylece geliştirilen algoritmanın SPD' ye göre çözüm performansının yanında daha önce bu problem için önerilen diğer algoritmalara olan performansı da incelenmeye çalışılmıştır. Bununla birlikte geliştirilen arı algoritması komşuluk üretme mekanizmasında eniyi yerel çiftli yer değiştirme (swap-greedy) mekanizması (EYÇYDM) ve eniyi yerel yerleştirme (insertion greedy) mekanizması (EYYM) kullanılarak komşuluk üretme mekanizmasına göre geliştirilen algoritmanın performansı detaylı incelenmiştir. Elde edilen çözüm sonuçları ve çözüm performansı ileride anlatılacaktır. Kullanılan veriler diğer çalışmanın sonunda ekte verilmekle beraber bu çalışmanın da sonuna konmuştur.

### 6.3 Uygulanan Arı Algoritmasının Çalışma Mekanizması

Uygulamalarda kullanılan arı algoritmasının şöyle basamaklandırabiliriz:

- Belirlenen izci arı sayısı kadar rastgele iş sıralaması oluşturulur.
- Her oluşturulan iş sıralamalarının toplam tamamlanma zamanı ( $C_{max}$ ) hesaplanır.
- Değeri hesaplanan iş sıralamalarından komşuluk araması için araştırma bölgesindeki olanları belirlenir.
- Araştırma bölgesindeki iş sıralamaları uygunluk değerine göre elit iş sıralamaları ve diğer araştırılacak seçilmiş iş sıralamalarının sayısı belirlenir. Elit iş sıralamalarının diğer seçilmiş bölgelere göre daha çok komşu iş sıralaması üretilir.
- Üretilen iş sıralamalarının  $C_{max}$  değerleri hesaplanır.
- Her bir araştırılma komşuları oluşturulmuş araştırma bölgesindeki iş sıralamalarının en uygunu seçilir.
- Geriye kalan atanmamış arılar sayısı kadar rassal iş sıralaması oluşturulur.
- En iyi iş sıralamaları ve rassal oluşmuş iş sıralamaları incelendiğinde durdurma kriterinin sağlanıp sağlanmadığına bakılır. Kriter sağlanmadıysa tekrartamamlanma zamanları hesaplanır ve tekrar araştırma bölgesi seçimiyle

süreç durdurma kriteri sağlanıncaya kadar uygulanır. Bu adımları anlatılan arı algoritmasının çalışma mekanizmasının özet gösterimi çizelge 6.3 de verilmiştir.

Geliştirilen algorithmada çalışma yönteminde elit arı (e) ve araştırma bölgesindeki geriye kalan arı (m-e) etrafında gönderilecek arı sayısının belirlenmesi komşuluk araştırılacak çevrenin büyüklüğü (ngh) beraber ele alınmıştır.

Arı algoritmasında komşuluk aranacak çevresinin büyüklüğü bu uygulamada geliştirilen algoritma için önemli bir noktadır. Çünkü bu uygulamada iş çizelgeleme problemleri için elit arıların etrafına gönderilecek arı sayısı ve diğer bölgesi arılarının etrafına gönderilecek arı sayısı bu parametreyle ilişkilendirilmiştir. Bu komşuluk arama yapılacağı çevrenin büyüklüğü komşuluk üretme mekanizmasıyla elde edilmiştir. Literatürde çeşitli komşuluk arama mekanizmaları bulunmaktadır [67].

Bu uygulamada komşuluk aramasının çevre büyüklüğünü oluştururken elit arı için iki metot seçilmiş kullanılmıştır. Bunlar EYÇYDM ve EYYM'dir. Bunları kısaca şöyle açıklayabiliriz.

**Eniyi Yerel Çiftli Yer Değiştirme Mekanizması (EYÇYDM):** Bu mekanizmada rassal seçilen bir iş, diğer pozisyonlardaki işlerle yer değiştirerek (n-1) adet yeni çizelge üretilir ve amaç fonksiyonunu eniyileyen çizelge seçilir.

Çizelge 6.1 En iyi yerel çiftli yer değiştirme mekanizması

| İş çizelgesi | Rassal seçilen iş | Seçilen iş | Yeni iş çizelgesi |
|--------------|-------------------|------------|-------------------|
| {1-2-5-3-4}  | {2}               | {1}        | {2-1-5-3-4}       |
|              |                   | {3}        | {1-3-5-2-4}       |
|              |                   | {4}        | {1-4-5-3-2}       |
|              |                   | {5}        | {1-5-2-3-4}       |
|              |                   |            |                   |

**Eniyi Yerel Yerleştirme Mekanizması (EYYM):** Bu mekanizmada rassal seçilen bir iş, diğer tüm pozisyonlara yerleştirilerek (n-1) tane yeni çizelge üretilir ve amaç fonksiyonunu eniyileyen çizelge seçilir.

Çizelge 6.2 En iyi yerel yerleştirme mekanizması

| İş çizelgesi | Rassal seçilen iş | İş pozisyonu | Yeni iş çizelgesi |
|--------------|-------------------|--------------|-------------------|
| {1-2-5-3-4}  | {2}               | {1}          | {2-1-5-3-4}       |
|              |                   | {3}          | {1-5-2-3-4}       |
|              |                   | {4}          | {1-5-3-2-4}       |
|              |                   | {5}          | {1-5-3-4-2}       |

Eğer  $n$  tane iş varsa EYÇYDM komşuluk mekanizmasına göre  $[n*(n-1)]/2$  tane komşu, EYYM komşuluk mekanizmasına göre  $[n-1]^2$  tane komşu üretilir. Uygulama yaptığımız probleme ve komşuluk yapısına göre seçilmiş bölgedeki iş sıralamalarının komşuluk çevresi belirlenir. Elit arı etrafına gönderilecek arı sayısı (nep) bu iki mekanizmayla belirlenmiştir. Yani kullandığımız komşuluk üretme mekanizmasına göre oluşan bölge sayısı kadar izci arı elit arılarının etrafına gönderilir. Örneğin 7 işlik bir problemde komşuluk mekanizması EYÇYDM olduğunda elit arının 21 tane komşusu olduğu için etrafına 21 tane arı gönderilir. Eğer 7 işlik problemde EYYM ile komşuluk üretildiğinde 36 tane komşu için 36 tane arı gönderilir.

Diğer bölgesi arılarının etrafına gönderilecek arı sayısı (nsp) ise mevcut iş sıralamalarındaki iş sıralarındaki işlerin bloklar yöntemiyle ayrılarak yerlerinin değiştirilmesiyle elde edilmiştir. Burada bu yöntemin kullanılmasının nedeni arı algoritmasının temel prensibi olan elit arıların etrafına araştırma bölgesi dışında kalan bölgelere göre daha fazla izci arı gönderilmesinden kaynaklanmaktadır.

Örneğin mevcut çizelgemiz 4-2-1-3 olsun. Bu iş sıralamasının ngh sayısı için komşuluk mekanizmasında 2'den başlayarak  $n-1$  komşuluk bloklarına göre  $S=\{1-3-4-2, 2-1-3-4, 3-4-1-2\}$  iş çizelgeleri elde edilir. İlk iş çizelgesinde 1-3 ve 4-2 iş blokları 2'li blok yapısını göstermektedir. Yerleri değiştirilerek yeni bir sıralama elde edilmiştir. İkinci ve üçüncü iş sıralamalarında 2-1-3-4 ve 4-1-2 3'lü blok yapısını göstermektedir. Bu blokların yerini değiştirilerek yeni iş komşulukları oluşturularak ngh arı sayısı ve araştırma yapacakları komşu aralığı belirlenir.

Durdurma kriteri ise elde edilen en uygun çözümün iş sayısı kadar aynı olmasıdır. Eğer her adımda gelişme var ise yöntemin uygulanmasına izin verilir.

Çizelge 6.3 Uygulanan arı algoritmasının çalışma mekanizması

|                                                                                                                |
|----------------------------------------------------------------------------------------------------------------|
| 1.Rassal iş sıralamalarıyla başlangıç çözüm kümesinin oluşumu.                                                 |
| 2.Başlangıç çözüm kümesindeki iş sıralamalarının tamamlanma sürelerini hesaplanması.                           |
| 3.Durdurma kriteri sağlanmıyorsa // Yeni çözüm kümesinin oluşumu.                                              |
| a)Komşuluk araması için çözümlerin seçimi.                                                                     |
| b)Seçilmiş bölgelere arama yapılması(elit iş sıralamalarında daha fazla araştırma yapılması).                  |
| c)Her bölgede en uygun tamamlanma zamanına sahip iş sıralamasının seçilmesi.                                   |
| d)Geriye kalan iş sıralamalarından rassal seçim yapılması ve her birinin iş tamamlanma süresinin hesaplanması. |
| 4.Son verme.                                                                                                   |

#### 6.4 Arı Algoritması Parametreleri

Uygulama problemlerinde arı algoritmasıyla çözüm aranması için tanımlanması gereken parametreler şunlardır: kâşif arı sayısı (**n**), n bölgeden seçilecek bölge sayısı (**m**), elit arı sayısı (**e**), diğer araştırılacak seçilmiş bölge sayısı (**m-e**), elit arıların etrafına gönderilecek arı sayısı (**nep**), diğer bölgesi arılarının etrafına gönderilecek arı sayısı (**ng**), komşuluk aramasının yapılacağı çevre büyüklüğü ve durdurma kriteridir. Eğer problemdeki iş sayısı N olursa bu parametrelerden **n**, **m**, **e**, **m-e** N sayısı ile ilişkilendirilmiştir. Bu ilişkilendirme ve uygulama problemleri için belirlenen parametreler çizelge 6.3 ve 6.4’de verilmiştir. Diğer parametreler **nep**, **nsp** ve **ng** değerlerinin belirlenmesi önceki kısımda anlatılmıştır.



Çizelge 6.4 7 işlik problemler için belirlenen parametreler

| PARAMETRELER                                                   | DEĞER                  |
|----------------------------------------------------------------|------------------------|
| Kaşif arı sayısı (n)                                           | 50                     |
| Seçilmiş bölge sayısı (m)                                      | N(7)                   |
| Elit arı sayısı(e)                                             | N/2(4)                 |
| Elit arıların etrafına gönderilecek arı sayısı(nep)            | EYÇYDM(21)<br>EYYM(36) |
| Diğer bölgesi arılarının etrafına gönderilecek arı sayısı(nsp) | 12                     |

Çizelge 6.5 10 işlik problemler için belirlenen parametreler

| PARAMETRELER                                                   | DEĞER                  |
|----------------------------------------------------------------|------------------------|
| Kaşif arı sayısı (n)                                           | 50                     |
| Seçilmiş bölge sayısı (m)                                      | N(10)                  |
| Elit arı sayısı(e)                                             | N/2(5)                 |
| Elit arıların etrafına gönderilecek arı sayısı(nep)            | EYÇYDM(45)<br>EYYM(81) |
| Diğer bölgesi arılarının etrafına gönderilecek arı sayısı(nsp) | 24                     |

#### 6.4 Arı Algoritmasıyla Örnek Bir Uygulama

Daha önceki kısımlarda anlatılan geliştirilen arı algoritmasının daha iyi anlaşılması için örnek bir problemin çözümünü inceleyelim. Çizelge 6.5’de gösterildiği gibi sıralı akış problem tipine uygun 5 iş 5 makine problemi bulunmaktadır. Öncelikle bu problemi SPD algoritmasıyla çözüldüğünde en iyi iş sıralaması olarak {5-2-3-1-4} sıralaması ve tamamlanma zamanı 118 olarak elde edilmektedir.

Çizelge 6.6 5-iş 5-makine örnek sorusu

| İş/makine | 1  | 2  | 3  | 4  | 5  |
|-----------|----|----|----|----|----|
| 1         | 19 | 14 | 21 | 16 | 12 |
| 2         | 13 | 12 | 18 | 10 | 8  |
| 3         | 16 | 14 | 11 | 12 | 10 |
| 4         | 12 | 11 | 16 | 9  | 7  |
| 5         | 5  | 3  | 9  | 2  | 4  |

Bu problem için arı algoritmasını uygulamak için gereken parametreleri şunlar olsun :

$n=10$ ,  $m=3$ ,  $e=2$ ,  $nep=EYÇYDM(10)$ ,  $nsp=5$ .

**Adım 1:** Kaşif arı sayısı kadar (10) tane rassal iş sıralaması seçip tamamlanma süresini hesapla.

Çizelge 6.7 Rassal olarak seçilen iş sıralamaları ve tamamlanma zamanları

| İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|-----------------|----------|----------|----------|----------|---------------------|
| <b>3</b>        | <b>4</b> | <b>1</b> | <b>5</b> | <b>2</b> | <b>123</b>          |
| 2               | 4        | 1        | 3        | 5        | 128                 |
| <b>3</b>        | <b>4</b> | <b>1</b> | <b>2</b> | <b>5</b> | <b>121</b>          |
| <b>4</b>        | <b>3</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>125</b>          |
| 4               | 1        | 5        | 2        | 3        | 136                 |
| 2               | 4        | 5        | 3        | 1        | 128                 |
| 3               | 4        | 5        | 1        | 2        | 125                 |
| 4               | 5        | 2        | 3        | 1        | 128                 |
| 3               | 5        | 2        | 4        | 1        | 128                 |
| 5               | 4        | 1        | 2        | 3        | 136                 |

**Adım 2:** Araştırma bölgesine seçilecek iş sıralamalarını belirle. Bunlardan elit ve araştırma bölgesindeki diğer arıları belirt.

Çizelge 6.8 Araştırma bölgesindeki iş sıralamaları

|            | İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|------------|-----------------|----------|----------|----------|----------|---------------------|
| <b>e</b>   | <b>3</b>        | <b>4</b> | <b>1</b> | <b>2</b> | <b>5</b> | <b>121</b>          |
|            | <b>3</b>        | <b>4</b> | <b>1</b> | <b>5</b> | <b>2</b> | <b>123</b>          |
| <i>m-e</i> | <i>4</i>        | <i>3</i> | <i>5</i> | <i>1</i> | <i>2</i> | <i>125</i>          |

**Adım 3:** Seçilen komşuluk üretim mekanizmasına göre elit iş sıralamalarının komşu iş sıralamalarını üretip bu iş sıralamalarının uygunluk değerlerinin hesaplanması. (Elit iş sıralamalarının etrafına gönderilecek izci arı sayısı da üretilen komşu sayısı kadar belirlenir.)

Çizelge 6.9 1.elit işin komşuluk çevresi ve uygunluk değerleri

| İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|-----------------|----------|----------|----------|----------|---------------------|
| <b>3</b>        | <b>4</b> | <b>1</b> | <b>2</b> | <b>5</b> | <b>121</b>          |
| 4               | 3        | 1        | 2        | 5        | 121                 |
| 1               | 4        | 3        | 2        | 5        | 123                 |
| 2               | 4        | 1        | 3        | 5        | 128                 |
| 5               | 4        | 1        | 2        | 3        | 136                 |
| 3               | 1        | 4        | 2        | 5        | 118                 |
| 3               | 2        | 1        | 4        | 5        | 116                 |
| 3               | 5        | 1        | 2        | 4        | 118                 |
| 3               | 4        | 2        | 1        | 5        | 122                 |
| 3               | 4        | 5        | 2        | 1        | 128                 |
| 3               | 4        | 1        | 5        | 2        | 123                 |

Birinci elit arının oluşturduğu araştırma bölgesinde uygunluk değeri en iyi iş sıralaması tamamlanma zamanı 116 olan  $S=\{3-2-1-4-5\}$  sıralamasıdır.

Çizelge 6.10 2.elit işin komşuluk çevresi ve uygunluk değerleri

| İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|-----------------|----------|----------|----------|----------|---------------------|
| <b>3</b>        | <b>4</b> | <b>1</b> | <b>5</b> | <b>2</b> | <b>123</b>          |
| 4               | 3        | 1        | 5        | 2        | 123                 |
| 1               | 4        | 3        | 5        | 2        | 123                 |
| 5               | 4        | 1        | 3        | 2        | 127                 |
| 2               | 4        | 1        | 5        | 3        | 136                 |
| 3               | 1        | 4        | 5        | 2        | 122                 |
| 3               | 5        | 1        | 4        | 2        | 122                 |
| 3               | 2        | 1        | 5        | 4        | 116                 |
| 3               | 4        | 5        | 1        | 2        | 125                 |
| 3               | 4        | 2        | 5        | 1        | 128                 |
| 3               | 4        | 1        | 2        | 5        | 121                 |

İkinci elit arının oluşturduğu araştırma bölgesinde uygunluk değeri en iyi iş sıralaması tamamlanma zamanı 116 olan  $S=\{3-2-1-5-4\}$  sıralamasıdır.

Çizelge 6.11 (m-e) komşuluk çevresi ve uygunluk değerleri

| İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|-----------------|----------|----------|----------|----------|---------------------|
| <b>4</b>        | <b>3</b> | <b>5</b> | <b>1</b> | <b>2</b> | <b>125</b>          |
| 3               | 5        | 4        | 1        | 2        | 125                 |
| 1               | 2        | 3        | 5        | 4        | 125                 |
| 4               | 1        | 2        | 3        | 5        | 117                 |
| 5               | 1        | 2        | 4        | 3        | 128                 |
| 3               | 5        | 1        | 2        | 4        | 136                 |

Araştırma bölgesindeki diğer iş sıralamasında uygunluk değeri en iyi iş sıralaması tamamlanma zamanı 117 olan  $S=\{4-1-2-3-5\}$  sıralamasıdır.

**Adım 4:** Geriye kalan atanmamış arıların (10-3) rassal olarak iş sıralamalarının seçip ve uygunluk değerlerinin hesaplanması. Araştırma bölgesi içindeki en uygun arılarla ve geriye kalan atanmamış arılarla oluşan yeni popülasyonun belirlenmesi.

Çizelge 6.12 Yeni popülasyondaki iş sıralamaları ve tamamlanma süreleri

| İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|-----------------|----------|----------|----------|----------|---------------------|
| <b>3</b>        | <b>2</b> | <b>1</b> | <b>4</b> | <b>5</b> | <b>116</b>          |
| <b>3</b>        | <b>2</b> | <b>1</b> | <b>5</b> | <b>4</b> | <b>116</b>          |
| <b>4</b>        | <b>1</b> | <b>2</b> | <b>3</b> | <b>5</b> | <b>117</b>          |
| 5               | 1        | 2        | 3        | 4        | 122                 |
| 1               | 4        | 3        | 5        | 2        | 123                 |
| 5               | 4        | 3        | 2        | 1        | 128                 |
| 1               | 2        | 3        | 4        | 5        | 119                 |
| 3               | 1        | 4        | 2        | 5        | 118                 |
| 4               | 5        | 2        | 3        | 1        | 128                 |
| 4               | 3        | 2        | 1        | 5        | 122                 |

Adım 5: Oluşan yeni iş çizelgeleme popülasyonu incelenir. Durdurma kriteri sağlanıyorsa algoritma sonlandırılır. Sağlanmıyorsa yeni bir araştırma bölgesi seçilip algoritmaya devam edilir.

Çizelge 6.13 Yeni popülasyondaki araştırma bölgesi

|          | İş sıralamaları |          |          |          |          | Tamamlanma süreleri |
|----------|-----------------|----------|----------|----------|----------|---------------------|
| <b>e</b> | <b>3</b>        | <b>2</b> | <b>1</b> | <b>4</b> | <b>5</b> | <b>116</b>          |
|          | <b>3</b>        | <b>2</b> | <b>1</b> | <b>5</b> | <b>4</b> | <b>116</b>          |
| m-e      | 4               | 1        | 2        | 3        | 5        | 117                 |

Buraya kadar sadece bu problemin bir adımı yapılmıştır. Elde edilen en iyi çizelge tamamlanma zamanı 116 olan {3-2-1-4-5} sıralaması elde edilmiştir. Bu problemi SPD algoritmasıyla yapılmasıyla ise en iyi çizelge 118 birim olan {5-2-3-1-4} sıralaması elde edilmiştir. Bu problemde gösteriyor ki sadece yapılan bir adımda elde edilen çözüm SPD' ye göre daha iyidir.

## 6.6 Uygulama Problemler Sonuçları

Uygulama problemlerinde alınan veriler daha önceden belirtildiği gibi bir çalışmadan alınmıştır (Körez [49]). Bu veriler geliştirilen arı algoritması ile çözülürken elit arı etrafına gönderilecek arı sayısı ve komşuluk çevresi büyüklü için EYÇYDM ve EYYM metotlarından faydalanılmıştır. Bunun için aslında uygulanan mekanizmaya göre iki tip arı algoritması uygulanmıştır. Bunlardan EYÇYDM mekanizmasını kullanan algoritma BASG, EYYM mekanizmasını kullanan algoritma BAIG olarak adlandırılacaktır.

Yapılan çalışmada SPD algoritması Genetik algoritma ve tavlama benzetimi ile kıyaslanmıştır. Bu çalışmada başlangıç popülasyonu olarak 60 kromozom seçilmiştir. Popülasyondaki bireylerin uygunluk değeri  $C_{max}$  en küçükleyecek şekilde belirlenmiştir. Bireylerin seçiminde ise rulet çemberi kullanılmıştır. Genetik operatör olarak da sıraya dayalı çaprazlama seçilmiştir. Mutasyon oranı 0 alınmıştır. Tamamlanma kriteri ise 20 jenerasyon olarak tanımlanmıştır. Genetik algoritmanın performansı da 1200 iterasyonluk bir tavlama benzetimi ile kontrol edilmiştir (Körez [50]).

Bu çalışmada aynı problem verileri kullanılarak arı algoritmasıyla çözülmüştür. Bununla birlikte komşuluk üretim mekanizması yapılarak algoritmanın kendi içinde performansı incelenmeye çalışılmıştır. Ayrıca arı algoritmaları genetik algoritma ve tavlama benzetimi ile kıyaslanarak arı algoritmasının önceden önerilen algoritmalarla olan performansı gözlemlenmiştir.

Genel ve özel durum problem durumları incelenmiştir. Genel durum işler için en büyük işlem zamanlarının ara makinelerden birinde gerçekleşmesi durumudur. Özel durum işlem zamanları matrisinde en büyük işlem zamanları ilk ya da son makinede oluşması durumudur Bu uygulama kapsamında her iki durum problemi için n-iş, m-makine

şeklinde, 7x4, 7x5, 7x8, 7x10, 7x15, 10x4, 10x5, 10x8, 10x10, 10x15 20 farklı iş-makine kombinasyonu ele alınmıştır. İşlem süreleri [1-25] aralığında üniform olarak elde edilmiştir.

Genel durum problemlerinin çözüm sonuçları çizelge 6.14’de verilmiştir. Çizelge incelendiğinde bütün problem kombinasyonlarında önerilen BASG ve BAIG yöntemleriyle arı algoritmasının tüm algoritmalar için en başarılı çözüm yöntemi olduğu görülmektedir. Çizelge 6.15 da ise algoritmaların birbirine olan tamamlanma zamanı baz alınarak çözüm performansı gösterilmiştir. Bulunan tamamlanma zamanlarının birbirine olan oranında çözüm performansı elde edilmeye çalışılmıştır. Uygulama sonucunda BA/SPD’nin ortalama performans değeri 98.61, SPD/GA ortalama performans değeri 99.17 çıkmıştır. Genel durum problemlerinde arı algoritması SPD algoritmasından daha başarılı sonuçlar elde edilmiştir. En kötü performansı da tavlama benzetimi yöntemiyle elde edilmiştir.

Çizelge 6.14 Genel durum problemlerinin uygulama sonuçları

| Problem Çeşidi | SPD               | BASG              | BAIG              | GA                | TA                |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı |
| 7x4            | 145               | 145               | 145               | 149               | 153               |
| 7x5            | 140               | 140               | 140               | 141               | 144               |
| 7x8            | 223               | 223               | 223               | 225               | 234               |
| 7x10           | 225               | 225               | 225               | 226               | 231               |
| 7x15           | 332               | 330               | 330               | 334               | 343               |
| 10x4           | 223               | 216               | 216               | 225               | 231               |
| 10x5           | 231               | 225               | 225               | 231               | 247               |
| 10x8           | 279               | 270               | 270               | 281               | 297               |
| 10x10          | 306               | 293               | 293               | 306               | 337               |
| 10x15          | 361               | 361               | 361               | 366               | 381               |

Çizelge 6.15 Genel problemlerde algoritmalarının performans karşılaştırılması

| PROBLEM | BA/SPD | SPD/GA | SPD/TA | BA/GA | BA/TA |
|---------|--------|--------|--------|-------|-------|
| 7x4     | 100    | 97.32  | 94.77  | 97.32 | 94.77 |
| 7x5     | 100    | 99.29  | 97.22  | 99.29 | 97.22 |
| 7x8     | 100    | 99.11  | 95.3   | 99.11 | 95.3  |
| 7x10    | 100    | 99.56  | 97.4   | 99.56 | 97.4  |
| 7x15    | 99.39  | 99.4   | 96.79  | 99.88 | 96.21 |
| 10x4    | 96.81  | 99.11  | 96.54  | 96    | 93.51 |
| 10x5    | 97.40  | 100    | 93.52  | 97.4  | 91.09 |
| 10x8    | 96.74  | 99.29  | 93.94  | 96.09 | 90.91 |
| 10x10   | 95.75  | 100    | 90.8   | 95.75 | 86.94 |
| 10x15   | 100    | 98.63  | 94.75  | 98    | 94.75 |

Özel durum problemlerinin çözüm sonuçları çizelge 6.16’de verilmiştir. Çizelge incelendiğinde bütün problem kombinasyonlarında önerilen BASG ve BAIG yöntemleriyle arı algoritmasının tüm algoritmalar için en başarılı çözüm yöntemi olduğu görülmektedir. Çizelge 6.17’de ise algoritmaların birbirine olan tamamlanma zamanı çözüm performansı gösterilmiştir. Bulunan tamamlanma zamanlarının birbirine olan oranında çözüm performansı elde edilmeye çalışılmıştır. Uygulama sonucunda BA/SPD ortalama performans değeri 99.39, SPD/GA ortalama performans değeri 98.43 çıkmıştır. Bu da gösteriyor ki, özel durum problemlerinde arı algoritması SPD algoritmasından daha başarılı sonuçlar elde edilmiştir. En kötü performansı da tavlama benzetimi yöntemiyle elde edilmiştir.



Çizelge 6.16 Özel durum problemlerinin uygulama sonuçları

| Problem Çeşidi | SPD               | BASG              | BAIG              | GA                | TA                |
|----------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı | Tamamlanma zamanı |
| 7x4            | 142               | 142               | 142               | 142               | 146               |
| 7x5            | 225               | 225               | 225               | 237               | 236               |
| 7x8            | 236               | 236               | 236               | 243               | 277               |
| 7x10           | 255               | 255               | 255               | 258               | 264               |
| 7x15           | 433               | 429               | 429               | 434               | 483               |
| 10x4           | 220               | 211               | 211               | 230               | 238               |
| 10x5           | 222               | 222               | 222               | 222               | 234               |
| 10x8           | 301               | 301               | 301               | 303               | 321               |
| 10x10          | 356               | 356               | 356               | 358               | 377               |
| 10x15          | 385               | 381               | 381               | 388               | 396               |

Çizelge 6.17 Özel problemlerde algoritmalarının performans karşılaştırılması

| PROBLEM | BA/SPD | SPD/GA | SPD/TA | BA/GA | BA/TA |
|---------|--------|--------|--------|-------|-------|
| 7x4     | 100    | 100    | 97.26  | 100   | 97.26 |
| 7x5     | 100    | 94.94  | 95.33  | 94.93 | 95.33 |
| 7x8     | 100    | 97.12  | 85.19  | 97.11 | 85.19 |
| 7x10    | 100    | 98.84  | 96.59  | 98.83 | 96.59 |
| 7x15    | 99.07  | 99.77  | 89.64  | 98.84 | 88.81 |
| 10x4    | 95.90  | 95.65  | 92.43  | 91.73 | 88.65 |
| 10x5    | 100    | 100    | 94.87  | 100   | 94.87 |
| 10x8    | 100    | 99.34  | 93.76  | 99.33 | 93.76 |
| 10x10   | 100    | 99.44  | 94.42  | 99.44 | 94.42 |
| 10x15   | 98.96  | 99.23  | 97.22  | 98.19 | 96.21 |

## 6.7 Uygulama Problemlerinin Değerlendirilmesi

Sıralı akış tipi iş çizelgelenmesinde arı algoritmasının uygulanmasının performansını incelemek için daha önceden yapılmış bir çalışmanın uygulama problemleri incelenmiş ve kıyaslama yapılmıştır. Bu uygulamalar Smith tarafından tanımlanmış genel ve özel durum problemlerine uygun farklı iş makine kombinasyonunda 20 adet problemidir. Bu problemlerde amaç tamamlanma zamanının minimizasyonudur.

Bu çalışmada sıralı akış problemlerinde arı algoritmasının performansını ölçerken algoritma da kullanılan komşuluk üretme mekanizmasının performansı da incelenmiştir. Bunun için arı algoritmasında EYÇYDM ve EYYM metotları her bir problem için ayrı ayrı uygulanıp inceleme yapılmıştır.

Öncelik olarak arı algoritmasında komşuluk üretme mekanizmaları incelendiğinde her iki komşuluk üretme mekanizmasına sahip olan BASG ve BAIG algoritmaların aynı sonuca ulaştıkları gözlemlenmiştir. Her problem tipinde aynı tamamlanma zamanı elde edilmiştir. Bu da gösteriyor ki, arı algoritmasının komşuluk yapısının değişmesi bu akış tipi problem çözümünde sonucu etkileyememiştir. BAIG algoritmasında elit arı etrafına gönderilen arı sayısı BASG fazla olmasına rağmen aynı değerler elde edilmiştir. Elit arının etrafında daha fazla araştırma yapıp iyi sonuç çıkma olasılığının fazla olması mantığına göre gerçekleştirilen komşuluk üretme mekanizma yöntemlerinden aynı sonuçlar elde edilmiştir.

Genel durum problemleri incelendiğinde geliştirilen arı algoritmasının SPD algoritması ve önceki çalışmadan önerilen genetik algoritmaya göre daha başarılı sonuçlar elde edilmiştir. Algoritmalar incelendiğinde en başarısız olanın tavlama benzetimi, en başarılı olanın da arı algoritması olduğu ortaya çıkmıştır. Uygulama problemlerinden arı algoritması 7x4, 7x5, 7x8, 7x10 ve 10x15 problemlerinde SPD algoritmasıyla aynı sonuçlara ulaşmış geriye kalan problemlerde ise daha iyi sonuçlar elde ederek çözümü geliştirmiştir. Çözümler incelendiğinde problem yapısı büyüdükçe arı algoritmasının performansı SPD algoritmasını geride bırakmıştır. Diğer yöntemlerde genetik algoritma yakın sonuçlar elde ederken arı algoritması ise aynı sonuç değerlerini yakalamakla beraber sonuç geliştirilmiştir.

Çözüm performanslarının yüzdelerini incelersek, arı algoritması SPD algoritmasına göre ortalama % 101.42 performans göstermiştir. Bu da uygulama problemlerinde SPD algoritmasının performansı arı algoritmasına göre ortalama % 98.61'de kalmıştır. Genetik algoritma ortalama % 97.8, tavlama benzetimi de ortalama % 93.8 arı algoritmasına göre geride kalmıştır.

Özel durum problemleri incelendiğinde ise geliştirilen arı algoritmasının SPD algoritması ve önceki çalışmadan önerilen genetik algoritmaya göre daha başarılı sonuçlar elde edilmiştir. 10 problemin 7'sinde SPD algoritmasının arı algoritmasıyla aynı çözümlere ulaşmıştır. Fakat arı algoritması diğer 3 problemde sonucu geliştirmiş olmasıyla önerilen diğer problemlere göre daha başarılıdır.

Çözüm performanslarının yüzdelerini incelersek, arı algoritması SPD algoritmasına göre ortalama % 100.6 performans göstererek aynı performansı sağlamıştır. Bununla birlikte Genetik algoritma ortalama % 97.85, tavlama benzetimi de ortalama % 93.1 arı algoritmasına göre performans geride kalmıştır.

Bu bulguların ışığında şunları söylenebilir:

Sıralı akış tipi iş çizelgeleme problemleri için önerilen çözüm yöntemi karşısında arı algoritması iyi sonuçlar elde ederek bu problem tipi için çözüm bulmada kullanılabileceğini göstermiştir.

Küçük boyutu problem uygulamalarında SPD yöntemi arı algoritmasına yakın sonuçlar bulduğundan kullanmak mümkün olabilir. Fakat iş sayısı arttığında SPD yöntemiyle  $2^{n-1}$  tane alternatif inceleneceğinden dolayı daha iyi performans gösteren arı algoritmasının zaman açısından değerlendirilmesi faydalı olacaktır.

Özel durum problemlerinde ise arı algoritması SPD yöntemiyle yakın performans göstermiştir. Bu problemler incelendiğinde küçük boyutlu problemler için uygulanması daha kolay olan SPD kullanmak avantajlı olacaktır. Bununla birlikte problem yapısı büyüdükçe daha iyi çözümler üretebilecek olan arı algoritması tercih edilmesi iyi olacaktır.

Geliştirilen arı algoritmasında elit arı etrafına gönderilecek izci arı sayısı için denenen komşuluk mekanizmaları aynı sonuçları elde etmiştir. Bu da bu problem tipinde önerilen EYÇYDM ile EYYM arasında bir fark olmadığını göstermiştir.

## BÖLÜM 7

---

### SONUÇ VE ÖNERİLER

Bu tez çalışması kapsamında,  $n/m/P/C_{max}$  notasyonu ile gösterilen sıralı akış tipi çizelgeleme problemleri göz önüne alınmıştır. Literatürde akış tipi çizelgeleme problemleri için 2 makine ve  $n$  iş probleminin Johnson algoritması ile optimum çözüm bulunduğu bilinmektedir. Johnson algoritması modifiye edilerek 3 makineli problemler içinde uygulanabilmektedir. Ancak makine sayısının 3 den fazla olduğu durumlarda optimum çözüme ulaşmayı garanti eden bir algoritma henüz geliştirilememiştir. Bu nedenle yapılan çalışmalarda akış tipi çizelgeleme problemlerinin çizelgelenmesinde, sezgisel yöntemler önerilmiştir.

Sıralı akış tipi çizelgeleme problemleri, 1975 yılında Smith Panwalkar ve Dudek tarafından tanımlanmıştır. Bu problem tipi Akış tipi çizelgeleme problemlerinin alt kategorisidir. Genel ve özel olmak üzere iki tipten oluşmaktadır. Çözümü için SPD algoritması kullanılmaktadır. Bu problem tipi için çözüm yöntemi genetik algoritma ve tavlama benzetimi yöntemi ele alınmıştır. Ancak arı algoritmasının kullanıldığı bir çalışma bulunmamaktadır. Literatürdeki bu boşluğu kapatmak için bu problem tipi için arı algoritması yaklaşımı incelenmiştir.

Arı algoritması doğadaki arıların yiyecek bulma arayışından etkilenecek geliştirilmiş bir yöntemdir. Literatürde arı algoritmaların, akış tipi çizelgeleme problemlerinde başarı ile uygulandığını gösteren örnekler bulunmaktadır.

Arı algoritmasının, sıralı akış tipi çizelgeleme problemleri üzerindeki performansını incelemek için önceden yapılmış bir çalışmanın verilerinden faydalanılmıştır. Uygulamada problemlerin Smith Panwalkar ve Dudek tarafından tanımlandığı şekilde,

genel durum problemi ve özel durum problemi olarak ikiye ayrılmış ve uygulama her problem tipi için ayrı gerçekleştirilmiştir. Bunun için 20 örnek problem arı algoritmasıyla çözülmüştür. Bununla birlikte komşuluk üretme mekanizmasında iki farklı yöntem incelenerek performansları ölçülmeye çalışılmıştır.

Öncelikle geliştirilen arı algoritmasında elit arı etrafına gönderilecek izci arı sayısı için denenen komşuluk mekanizmalarının performansları incelendiğinde her iki yöntemle de aynı sonuçları elde etmiştir. Bu da bu problem tipinde önerilen EYÇYDM ile EYYM arasında bir fark olmadığını göstermiştir.

Genel problem durumu incelendiğinde çözüm performanslarının yüzdelerini incelersek, arı algoritması SPD algoritmasına göre ortalama % 101.42 performans göstermiştir. Bu da uygulama problemlerinde SPD algoritmasının performansı arı algoritmasına göre ortalama % 98.61’de kalmıştır. Genetik algoritma ortalama % 97.8, tavlama benzetimi de ortalama % 93.8 arı algoritmasına göre geride kalmıştır. Tüm genel problem durumlarında en kötü olarak SPD’yle çözümler bulunmuş, hatta bazı problemlerde çözüm geliştirilmiştir. Özel durum problemleri incelendiğinde ise geliştirilen arı algoritmasının SPD algoritması ve önceki çalışmadan önerilen genetik algoritmaya göre daha başarılı sonuçlar elde edilmiştir. 10 problemin 7’de SPD algoritmasının arı algoritmasıyla aynı çözümlere ulaşmıştır. Fakat arı algoritması diğer 3 problemde sonucu geliştirmiş olmasıyla önerilen diğer problemlere göre daha başarılıdır.

SPD algoritması genel problemler için uygulanışı basit görünse de öncelikli olarak problem büyüdükçe incelenen alternatif sayısı artacağı ( $2^{n-1}$ ) ve problem yapısı büyüdükçe çözüm performansının düştüğü göz önüne alınırsa arı algoritmasının kullanımı tercih edilmelidir. Özel problemlerde ise SPD algoritmasının kullanımı çok basit ve elde edilen sonucun arı algoritmasına yakın olmasından dolayı yöntem kullanılması iyi olacaktır.

Önerilen arı algoritmasında farklı komşuluk mekanizmalarının kullanımı daha iyi sonuçların elde edilmesine etki edebilir. Ayrıca bu problem tipi için daha büyük problem sekterinin çözülmesi arı algoritmasının performans kalitesini daha çok ortaya koyabilir. Arı algoritmasında daha çabuk çözüme ulaşılabilmesi için farklı algoritmalarla

birleřtirilip hibrit mekanizmalarla daha byk problem setlerinin zmnde veya btn akıř tipi problemleri iin genelleřtirilebilecek bir model elde edilebilir.

## KAYNAKLAR

- 
- [1] Smith, M. L., Panwalkar, S. S. ve Dudek, R. A., (1975). "FlowshopSequencing Problem withOrderedProcessing Time Matrices", Management Science, 21 (5): 544-549.
- [2] Smith, M. L., Panwalkar, S. S., Dudek, R. A., (1976). "FlowshopSequencing Problem withOrderedProcessing Time Matrices: A General Case", Naval ResearchLogisticsQuarterly, 23: 481-486.
- [3] Panwalker, S.S. and Khan, A.W., (1976). "An OrderedFlow-shopSequencing Problem WithMeanCompletion Time Criterion, Int. J. Prod. Res., 14 (5): 631-635.
- [4] Selam, A.A., (2011). Project Scheduling Using BeeColonyOptimization, Doktora Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- [5]Chong, C.S., Low, M.Y.H., Sivakumar, A.I. ve Gay, K.L, (2006). "A BeeColonyOptimizationAlgorithmtoJob Shop Scheduling", Proceedings of the 2006 WinterSimulationConfrence, 2006, Monterey.
- [6] Pham, D.T., Koç, E., Lee, J.Y. ve Phrueksanant, J., (2007). "Using theBeesAlgorithmto Schedule Jobsfor a Machine", Proceedings of the 8<sup>th</sup> International Conference on LaserMethodology CMM and Machine ToolPerformance, 2007, Cardiff.
- [7] Pham, D.T. ve Koç, E. . "FlowshopSequencing Using BeeAlgorithms", The IPROMS InnovativeProductionMachinesandSystems Conference, 2008,Cardiff.
- [8]Pan, J.L.Q, Xie, S. ve Wang, S., (2011). "A HybridArtificialBeeColonyAlgorithmforFlexibleJobSchedulingProblems", Int. J. of Computers, 6 (2) :286-296.
- [9] Banharnsakun, A., Sirinaovakul, B ve Achakul, T., (2012). "Job Shop Schedulingwiththe Best-so-far ABC", Engineering Applications of ArtificialIntelligence, 25(3):583-593.
- [10] Pinedo, M.L., (2012). SchedulingTheory, Algorithms, andSystems, Fourth Edition, Springer, Londra.
- [11] Mutlu, Ö., (1993). Tek Makine Çizelgeleme Probleminde En İyi Teslim Zamanlarının Belirlenmesi, Yüksek Lisans Tezi, Eskişehir Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir.
- [12] Güldalı, A., (1990). Seri İş-akışlı Atmlye Çizelgelemesinde Sezgisel Teknikler, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü.



- [13] Aladağ, A., (2010). Tekrar İşlemeli Esnek Atölye Tipi Çizelgeleme Problemi İçin Yapay Bağışıklık Sistemi İle Bir Çözüm Yaklaşımı, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir.
- [14] Brucker, P., Knust, S., (2007). ComplexScheduling, Springer, Newyork.
- [15] Kellegöz, T., (2006). Toplam Geç Bitirme Zamanının En Küçüklenmesi Performans ÖlçütlüPermütasyon Akış Tipi Çizelgeleme Problemlerinin Çözümünde Genetik Algoritma Yaklaşımı, Yüksek Lisans Tezi, Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü, Kırıkkale.
- [16]Vikipedi Özgür Ansiklopedi, P ile NP Arasındaki İlişki, [http://tr.wikipedia.org/wiki/P ile NP arasındaki ilişki](http://tr.wikipedia.org/wiki/P_ile_NP_arasındaki_iliski), 10 Mart 2012.
- [17] Weisstein E. W.,ComplexityTheory, <http://mathworld.wolfram.com/ComplexityTheory.html>, 21 Mart 2012.
- [18] Temiz, İ. ve Erol, S., (2007). “Bulanık Akış Tipi Çizelgeleme Problemi İçin Çok Amaçlı Genetik Algoritma”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 22 (4):855-862.
- [19] Clifford,R.W., (1980). OrderedFlowshopSequencingWithIn-processConstraints, Doktora Tezi, Texas TechUniversity, Texas.
- [20] Baker, K.R.,(1990). “SchedulingGroups of Jobs in theTwo-Machine Flowshop”, Math Comput. Model, 13: 29–36.
- [21] Hejazi, S.R. ve Saghafian, S., (2005). “Flowshop-schedulingProblemswithMakespanCriterion.AReview”, International Journal of ProductionResearch, 43 (14): 2895-2929.
- [22] Seçme, G., (2006). Akış Tipi Çizelgeleme Problemlerinin Yapay Sinir Ağları ile Modellenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü, Kayseri.
- [23] Akçay, E., (2009). Akış Tipi İş Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemi ile Çok Amaçlı Optimizasyonuna Yönelik Bir Model Önerisi, Doktora Tezi, Kocaeli Üniversitesi, Fen Bilimleri Üniversitesi, Kocaeli.
- [24] Engin, O., (2001). Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Artırılmasında Parametre Optimizasyonu, Doktora Tezi, İTÜ, Fen Bilimleri Enstitüsü, İstanbul.
- [25] Johnson, S.M., (1954). “Optimal Two- and Three StageProductionScheduleswithSetup Times Included”, Naval Res. Log. Quart., 1 (1): 61-68.
- [26] Ignall, E., Schrage, L.E., (1965). “Application of thebranchandboundtechniquetosomeflowshopsschedulingproblems,” Operations Research, 13 (3) :400-412.
- [27] Garg, N., Jain,S. andSwamy,C., (1999), “A RandomizedAlgorithmforFlow Shop Scheduling”, ComputerScience, 1738: 213-218.

- [28] Palmer, D.S., (1965). "Sequencing Jobs Through A Multi-Stage Process In The Minimum Total Time-A Quick Method Of Obtaining A Near Optimum", *Operational Research Quarterly*, 16:101-106.
- [29] Campbell, H.G., Dudek, R.A., Smith, M.L., (1970). "A Heuristic Algorithm For The N-Job M-Machine Sequencing Problem", *Management Science*, 16:630-637.
- [30] Gupta, J.N.D. ve Dudek, R.A., (1971). "An Optimality Criterion For Flowshop Schedules", *AIIE Trans.*, 3: 199-205.
- [31] Dannenbring, D.G., (1970). "An evaluation of flowshop sequencing heuristics", *Management Science*, 23 :1174-1182.
- [32] Nawaz, M., Ensore, Jr.E.E., Ham, I., (1983). "A Heuristic Algorithm For M-Machine, N-Job Flowshop Sequencing Problem", *Omega*, 11:91-98.
- [33] Hundal, T. S. ve Rajgopal, J., (1988). "An Extension Of Palmer's Heuristic For The Flowshop Scheduling Problem", *International Journal Of Production Research*, 26 (6).
- [34] Widmer, M. ve Hertz, A., (1989). "A New Heuristic Method For The Flowshop Sequencing Problem", *European Journal Of Operational Research*, 41:186- 193.
- [35] Ho, J.C. ve Chang, Y.L., (1991). "A New Heuristic For The N-Job, M Machine Flow- Shop Problem", *European Journal Of Operational Research*, 52: 194–202.
- [36] Ogbu, F.A. ve Smith, D.K, (1990). "The Application of the Simulated Annealing Algorithm to the Solution of the n/m/Cmax Flowshop Problem", *European Journal of Operational Research*, 17: 243-253.
- [37] Osman, I., Potts, C., (1989). "Simulated Annealing for Permutation Flow Shop Scheduling", *Omega*, 17: 551-557.
- [38] Chen, C.L., Neppalli, R.V. ve Aljaber, N., (1996). "Genetic algorithms applied to the continuous flowshop problem", *Computers and Industrial Engineering*, 30 (4): 919-929.
- [39] Cleveland, G.A., Smith, S.F., (1989). "Using genetic algorithms to schedule flowshop," *Proceedings of 3rd Conference on Genetic Algorithms*, Publishing, 1989, Virginia.
- [40] Eitler, O., Toklu, B., Atak, M. ve Wilson, J., (2004). "A genetic algorithm for flowshop scheduling problems", *Journal Operations Research Society*, 55: 830-835.
- [41] Murata, T., Ishibuchi, H., Tanaka, H., (1996). "Genetic Algorithms for Flow Shop Scheduling Problems", *Computers and Industrial Engineering*, 30 (4): 1061-1071.
- [42] Ponnambalam, S.G., Aravindan, P. and Chandrasekaran, S., (2001). "Constructive and Improvement Flow Shop Scheduling Heuristics: An Extensive Evaluation", *Prod. Plan. Contr.*, 12:335–344.
- [43] E. Rashidi, M. Jahandar, and M. Zandieh, (2010). "An Improved Hybrid Multi-Objective Parallel Genetic Algorithm for Hybrid Flow Shop

SchedulingwithUnrelatedParallelMachines”, International Journal of AdvancedManufacturingTechnology, 49 :1129-1139.

[44]Grabowski, J. ve Wodecki, M., (2004). “A VeryFast Tabu SearchAlgorithmforthePermutationFlowshop Problem withMakespanCriterion”, Computers&OperationsResearch, 31:1891-1909.

[45]Armentano, V.A. ve Ronconi, D.P., (1999). “Tabu Searchfor Total TardinessMinimization in FlowshopSchedulingProblems”, Computers&OperationsResearch, 26 (3):219-235.

[46]Hong, T.P., Wang, C.L. ve Wang, S.L., (2000). “A HeuristicGupta-BasedFlexibleFlow-Shop SchedulingAlgorithm”, Proceedings of the IEEE International Conference on Systems, 1: 319-322.

[47]Temiz, İ., (2010). “Çok Kriterli Permütasyon Akış Tipi Çizelgeleme Problemi İçin Bir Tavlama Benzetimi Yaklaşımı”, Çankaya UniversityJournal of ScienceandEngineering, 7 (2): 141-153.

[48]Liu, B., Wang, L. ve Jin, Y-H., (2007). “An effectivePSObasedmemeticalgorithmforflowshopscheduling”, IEEE, 37:18-27.

[49]Zobolas, G.I., Tarantilis, C.D. ve Ioannou, G., (2008). “MinimizingMakespan in PermutationFlow Shop SchedulingProblems Using a HybridMetaheuristicAlgorithm”, Computersand Operations Research, 36 (4): 1249-1267.

[50]Körez, M.T., (2005). Sıralı Akış Tipi Çizelgeleme Problemlerinde Genetik Algoritma Uygulaması, Yüksek Lisans Tezi, YTÜ, Fen Bilimleri Enstitüsü, İstanbul.

[51]Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rhim, S. Ve Zaidi, M., (2005). BeeAlgorithm A NovelApproachtoFunctionOptimisation, ManufacturingEngineerCentre, Cardiff University, Cardiff.

[52] Akay, B., (2009). Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi Algoritmasının Performans Analizi, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.

[53]Şifalı Bitkiler, Mucizevi Hayvan Arı, <http://www.sifalibitkiler.us/archives/864>, 10 Mart 2012

[54]Vikipedi Özgür Ansiklopedi, Bal Arısı, [http://tr.wikipedia.org/wiki/Bal\\_arısı](http://tr.wikipedia.org/wiki/Bal_arısı) , 10 Mart 2012.

[55]Türkçe Bilgi, Arı, <http://www.turkcebilgi.com/ansiklopedi/arı> , 10 Mart 2012.

[56]Kargioğlu, A. Balarılarının Şaşırtıcı Özellikleri, [http://www.yaklasansaat.com/dunyamiz/canlilar/arilar\\_sasirtici\\_ozellikleri.asp](http://www.yaklasansaat.com/dunyamiz/canlilar/arilar_sasirtici_ozellikleri.asp), 21 Mart 2012.

[57]Tokmak, M., (2011). Yapay Arı Kolonisi Algoritması İle Ders Çizelgeleme Probleminin Çözümü, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta.

[58]WikipediatheFree Encyclopedia, Karl von Frisch, [http://en.wikipedia.org/wiki/Karl\\_von\\_Frisch](http://en.wikipedia.org/wiki/Karl_von_Frisch), 21 Mart 2012.

- [59]Winston, M.L., (1991). TheBiology of HoneyBee, First Edition, HarverdUniversityPress, USA.
- [60]Tereshko, V., Loengarov, A., (2005). "CollectiveDecisionMaking in Honey-BeeForaging Dynamics", Computing and Information Systems, 9 (3) :1-7.
- [61]Frisch, K.V., (1967). TheDance Language andOrientation of Bees, Harvard UniversityPress, USA.
- [62]Karaboğa, D., (2005). An IdeaBased on HoneyBeeSwarmforNumericalOptimization,Technical Report-tr06, Erciyes University, EngineeringFaculty, ComputerEngineeringDepartment, Kayseri.
- [63]Kıran, M.S., Gündüz, M. ve Şahman, M.A., (2009). "Arı Kolonisi Optimizasyon Algoritması Kullanarak En Kısa Yol Bulma", IV. İletişim Teknolojileri Ulusal Sempozyumu, 2007, İstanbul.
- [64]Tapkan, P., Özbakır, L. ve Baykasoğlu, A., (2010). "Arı Algoritması ve Genelleştirilmiş Atama Problemi: Farklı Komşuluk Yapılarının Karşılaştırılması", Endüstri Mühendisliği Dergisi, 21 (2): 2-13.
- [65]Yahya, H., (2006). Balarısı Mucizesi, İkinci Baskı, Araştırma Yayıncılık, İstanbul.
- [66]Düğenci, M., (2007). Arı Algoritmasının Yapay Sinir Ağı Öğrenmesi İçin Kullanımı ve Atıksu Arıtma Tesis Kontrolü Uygulaması, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- [67]Aarts, E. ve Lenstra, J.K., (1997). LocalSearch in CombinatorialOptimization, Second Edition, John Wiley&Sons Ltd., Newyork.

## BAIG İÇİN İŞ SIRALAMALARI

|                  |       | SIRALAMA     | TAMAMLANMA SÜRESİ |
|------------------|-------|--------------|-------------------|
| GENEL PROBLEMLER | 7x4   | 5761324      | 145               |
|                  | 7x5   | 2731465      | 140               |
|                  | 7x8   | 7314265      | 223               |
|                  | 7x10  | 6257134      | 225               |
|                  | 7x15  | 7164352      | 330               |
|                  | 10x4  | 10814532679  | 216               |
|                  | 10x5  | 95614107823  | 225               |
|                  | 10x8  | 73418610259  | 270               |
|                  | 10x10 | 21047918634  | 293               |
|                  | 10x15 | 598649110273 | 361               |
|                  |       |              |                   |
| ÖZEL PROBLEMLER  | 7x4   | 6753412      | 142               |
|                  | 7x5   | 5723416      | 225               |
|                  | 7x8   | 2431675      | 236               |
|                  | 7x10  | 7534126      | 255               |
|                  | 7x15  | 6742315      | 429               |
|                  | 10x4  | 48310297651  | 211               |
|                  | 10x5  | 21649103857  | 222               |
|                  | 10x8  | 53924106718  | 301               |
|                  | 10x10 | 53421097618  | 356               |
|                  | 10x15 | 58732694101  | 381               |
|                  |       |              |                   |

## BASG İÇİN İŞ SIRALAMALARI

|                  |       | SIRALAMA    | TAMAMLANMA SÜRESİ |
|------------------|-------|-------------|-------------------|
| GENEL PROBLEMLER | 7x4   | 5761324     | 145               |
|                  | 7x5   | 2731465     | 140               |
|                  | 7x8   | 7314265     | 223               |
|                  | 7x10  | 6571324     | 225               |
|                  | 7x15  | 6714352     | 330               |
|                  | 10x4  | 10132547698 | 216               |
|                  | 10x5  | 96541782103 | 225               |
|                  | 10x8  | 79348106215 | 270               |
|                  | 10x10 | 10914237865 | 293               |
|                  | 10x15 | 48965110273 | 361               |
| ÖZEL PROBLEMLER  | 7x4   | 1762435     | 142               |
|                  | 7x5   | 1573246     | 225               |
|                  | 7x8   | 2641375     | 236               |
|                  | 7x10  | 7435126     | 255               |
|                  | 7x15  | 9742315     | 429               |
|                  | 10x4  | 48107936251 | 211               |
|                  | 10x5  | 21810945637 | 222               |
|                  | 10x8  | 53796214108 | 301               |
|                  | 10x10 | 56912107348 | 356               |
|                  | 10x15 | 58763429101 | 381               |

## ÖZGEÇMİŞ

---

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Muhammed PARLAK  
**Doğum Tarihi ve Yeri** : 26.08.1987  
**Yabancı Dili** : İngilizce  
**E-posta** : parlakmp@gmail.com

### ÖĞRENİM DURUMU

| Derece | Alan                  | Okul/Üniversite            | Mezuniyet Yılı |
|--------|-----------------------|----------------------------|----------------|
| Lisans | Endüstri Mühendisliği | Yıldız Teknik Üniversitesi | 2010           |
| Lise   | Fen                   | Haydarpaşa Lisesi          | 2006           |