

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

**CAM KESME PROBLEMİNE ÇOK AMAÇLI YAKLAŞIM**

**MUSTAFA ZAHİD GÜRBÜZ**

**DOKTORA TEZİ  
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI  
MATEMATİK MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN  
DOÇ. DR. İBRAHİM EMİROĞLU**

**İSTANBUL, 2015**

T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**CAM KESME PROBLEMİNE ÇOK AMAÇLI YAKLAŞIM**

MUSTAFA ZAHİD GÜRBÜZ tarafından hazırlanan tez çalışması 23.10.2015 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Matematik Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Doç. Dr. İbrahim EMİROĞLU  
Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Doç. Dr. İbrahim EMİROĞLU  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Prof. Dr. Mustafa SİVRİ  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Yrd. Doç. Dr. Ayşe Salman  
Doğuş Üniversitesi

\_\_\_\_\_

Prof. Dr. Fatih Taşçı  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Prof. Dr. İsmail Müfit GİRESUNLU  
İstanbul Üniversitesi

\_\_\_\_\_

Bu alıřma BİDEB 2211 yurtii lisansüstü burs programı kapsamında TÜBİTAK tarafından desteklenmiřtir.

Bu alıřma YTÜ BAP Koordinatorlüğü tarafından 2014-07-03-DOP03 proje numaralı DOP Projesi olarak desteklenmiřtir.

## ÖNSÖZ

---

Cam üreticilerinin daha az maliyetle üretim yaparak fiyatları düşürebilmelerini sağlayabilmek amacıyla ekonomiye katkı sağlamak için yapılmış bir çalışmadır. Temel amaçlar üretimde oluşan fireyi azaltırken cam üzerinde kusurları da azaltarak kaliteyi arttırmaktır. Böylece daha az maliyetle daha kaliteli bir üretim sağlanmış olacaktır. Tez çalışmasında mevcut yöntemler incelenmiş ve yeni bir çözüm yolu öne sürülmüş ve daha sonraki çalışmalara ilham kaynağı olmak hedeflenmiştir.

Bu çalışma süresince her konuda yardımlarını ve desteğini esirgemeyen tez danışmanı Doç.Dr. İbrahim Emiroğlu'na, yorumlarıyla ve fikirleriyle beni motive eden Prof.Dr. Selim Akyokuş'a, bana destek olan Yrd.Doç.Dr. Ayşe Salman'a, değerli çalışma arkadaşlarım Yrd.Doç.Dr. Aysun Güran, Yrd.Doç.Dr. Kıvanç Onan ve diğer tüm mesai arkadaşlarıma yardımlarından dolayı teşekkür ederim. Ayrıca doktora çalışması boyunca sunduğu desteklerinden dolayı TÜBİTAK'a teşekkürlerimi bir borç bilirim.

Bu tezi, varlığını dünyalara değişmeyeceğim, beni hayatım boyunca her zaman destekleyen ve destekleyecek olan ve bu güne kadar iki evlat sahibi olmamızı sağlayarak bana dünyanın en güzel duygusunu iki kez yaşatan sevgili eşim Ayfer Gürbüz'e, ailemizin neşe kaynağı 2 yıllık üyesi Doruk Mete Gürbüz'e, 1 Ekim'de aramıza katılan Kerem Batu Gürbüz'e, beni dünyaya getiren ve bu yaşa gelmemde büyük emeği olan sevgili annem Yüksel Gürbüz ve babam Mehmet Gürbüz'e, kardeşe sahip olmanın güzelliğini anlamamı sağlayan canımdan çok sevdiğim abim Mürre Asım Gürbüz'e ve kardeşim Burcu Gül Gürbüz'e ithaf ediyorum. Varlığınızla beni motive ettiniz. Teşekkür ederim.

Ekim, 2015

Mustafa Zahid Gürbüz

## İÇİNDEKİLER

|                                                    | Sayfa |
|----------------------------------------------------|-------|
| SİMGE LİSTESİ.....                                 | viii  |
| KISALTMA LİSTESİ.....                              | ix    |
| ŞEKİL LİSTESİ.....                                 | x     |
| ÇİZELGE LİSTESİ .....                              | xii   |
| ÖZET .....                                         | xiii  |
| ABSTRACT.....                                      | xv    |
| BÖLÜM 1                                            |       |
| GİRİŞ.....                                         | 1     |
| 1.1. Literatür Özeti .....                         | 1     |
| 1.2. Tezin Amacı .....                             | 2     |
| 1.3. Orjinal Katkı.....                            | 3     |
| BÖLÜM 2                                            |       |
| CAM KESME PROBLEMİ .....                           | 4     |
| 2.1. Problemin Sınıflandırılması .....             | 5     |
| 2.2. Problemin Karmaşıklığı .....                  | 10    |
| 2.3. Problemin Matematik Modeli.....               | 11    |
| 2.4. Çözüm yaklaşımları .....                      | 12    |
| 2.4.1. Deterministik Yöntemler .....               | 12    |
| 2.4.2. Sezgisel Yöntemler.....                     | 12    |
| BÖLÜM 3                                            |       |
| CAM KESME PROBLEMİNDE SEZGİSEL YÖNTEMLER.....      | 14    |
| 3.1. Yerleştirme Yöntemleri .....                  | 14    |
| 3.1.1. Sol Alt Yerleştirme Algoritması (BLF) ..... | 14    |
| 3.1.1.1. Tanım.....                                | 14    |

|           |                                                                                    |    |
|-----------|------------------------------------------------------------------------------------|----|
| 3.1.1.2.  | Akış Şeması.....                                                                   | 15 |
| 3.1.2.    | İlk Uygun Yer Yerleştirme Algoritması (FF).....                                    | 17 |
| 3.1.2.1.  | Tanım.....                                                                         | 17 |
| 3.1.2.2.  | Akış Şeması.....                                                                   | 17 |
| 3.1.3.    | Önerilen Alan Öncelikli Yerleştirme Algoritması (LAFF) .....                       | 18 |
| 3.1.3.1.  | Tanım.....                                                                         | 18 |
| 3.1.3.2.  | Akış Şeması.....                                                                   | 18 |
| 3.1.4.    | Önerilen Sağ Alt – Sol Alt Dengelemeli Yerleştirme Algoritması (BLF-<br>BRF) ..... | 19 |
| 3.1.4.1.  | Tanım.....                                                                         | 19 |
| 3.1.4.2.  | Akış Şeması.....                                                                   | 20 |
| 3.2.      | Tek Amaçlı Evrimsel Yöntemler .....                                                | 21 |
| 3.2.1.    | Genetik Algoritma(GA) .....                                                        | 21 |
| 3.2.1.1.  | Tanım.....                                                                         | 21 |
| 3.2.1.2.  | Akış Şeması.....                                                                   | 22 |
| 3.2.1.3.  | Gen .....                                                                          | 23 |
| 3.2.1.4.  | Koromozom.....                                                                     | 23 |
|           | İkili Kodlama.....                                                                 | 24 |
|           | Reel Kodlama .....                                                                 | 24 |
|           | Permütasyon Kodlama.....                                                           | 25 |
|           | Ağaç Kodlama .....                                                                 | 25 |
| 3.2.1.5.  | Popülasyon.....                                                                    | 26 |
|           | Başlangıç Popülasyonu.....                                                         | 26 |
| 3.2.1.6.  | Uygunluk Fonksiyonu .....                                                          | 27 |
| 3.2.1.7.  | Elitizm.....                                                                       | 27 |
| 3.2.1.8.  | Kromozom seçimi.....                                                               | 28 |
|           | Rulet Tekeri Seçim Algoritması .....                                               | 28 |
| 3.2.1.9.  | Çaprazlama.....                                                                    | 29 |
|           | İkili Kodlamada Çaprazlama .....                                                   | 30 |
|           | Permütasyon Kodlamada Çaprazlama .....                                             | 31 |
| 3.2.1.10. | Mutasyon .....                                                                     | 31 |
|           | İkili Kodlamada Mutasyon .....                                                     | 32 |
|           | Permütasyon Kodlamada Mutasyon .....                                               | 32 |
| 3.2.2.    | Yapay Arı Algoritması(ABC) .....                                                   | 33 |
| 3.2.2.1.  | Gerçek Arıların Davranışları .....                                                 | 33 |
| 3.2.2.2.  | Tanım.....                                                                         | 34 |
| 3.2.2.3.  | Akış Şeması.....                                                                   | 35 |
| 3.2.2.4.  | Başlangıç Durumu .....                                                             | 36 |
| 3.2.2.5.  | İşçi Arıların Kaynağı Geliştirmesi .....                                           | 36 |
| 3.2.2.6.  | Gözcü Arıların Kaynak Seçimi Yapması .....                                         | 36 |
| 3.2.2.7.  | Bir Kaynağın Terkedilmesi.....                                                     | 37 |
| 3.2.2.8.  | Global Ve Yerel Optimum Noktaları.....                                             | 37 |
| 3.3.      | Çok Amaçlı Evrimsel Yöntemler .....                                                | 37 |
| 3.3.1.    | Çok Amaçlı Optimizasyonun Tanımı .....                                             | 37 |
| 3.3.2.    | Pareto Verimliliği .....                                                           | 39 |
| 3.3.3.    | Kuvvet Pareto Evrimsel Algoritma (SPEA, SPEA2) .....                               | 40 |

|                           |                                                              |    |
|---------------------------|--------------------------------------------------------------|----|
| 3.3.4.                    | Pareto Eriřimli Evrim Stratejisi (PAES) .....                | 41 |
| 3.3.5.                    | Bastırılmamış Sıralamalı Genetik Algoritma 2 (NSGA II) ..... | 41 |
| 3.3.6.                    | Önerilen Sağ Sol Dengelemeli Genetik Algoritma (GA_LR) ..... | 43 |
| <b>BÖLÜM 4</b>            |                                                              |    |
| DENEYSEL ÇALIřMALAR ..... |                                                              | 45 |
| 4.1.                      | Deney Ortamı .....                                           | 45 |
| 4.2.                      | Veri Kümeleri .....                                          | 46 |
| 4.2.1.                    | Jacobs veri kümesi .....                                     | 46 |
| 4.2.2.                    | Hopper veri kümesi .....                                     | 46 |
| 4.2.3.                    | ZDF Büyük veri kümesi .....                                  | 47 |
| 4.2.4.                    | Valenzuela Veri kümesi .....                                 | 48 |
| 4.2.5.                    | Sanayi Veri kümesi .....                                     | 48 |
| 4.2.6.                    | Kusur Veri kümesi .....                                      | 49 |
| 4.3.                      | Deney Sonuçları .....                                        | 49 |
| <b>BÖLÜM 5</b>            |                                                              |    |
| SONUÇ VE ÖNERİLER .....   |                                                              | 54 |
| EK-A                      |                                                              |    |
| VERİ KÜMELERİ .....       |                                                              | 57 |
| A-1                       | Kusur Veri Kümesi .....                                      | 57 |
| A-2                       | Sanayi veri kümesi .....                                     | 58 |
| A-3                       | Diğer veri kümeleri .....                                    | 58 |
| KAYNAKLAR .....           |                                                              | 59 |
| ÖZGEÇMİř .....            |                                                              | 63 |

## SİMGE LİSTESİ

---

|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| $S_i$          | Kesilmesi istenen i.parça                                            |
| $\varphi(S_i)$ | Kesilmesi istenen i.parçanın kesim planında bulunması durumu         |
| $w_i$          | Kesilmesi istenen i.parçanın genişliği                               |
| $h_i$          | Kesilmesi istenen i.parçanın yüksekliği                              |
| $k_i$          | Kesilmesi istenen i.parçanın adedi                                   |
| $W$            | Cam bloğunun genişliği                                               |
| $H$            | Cam bloğunun yüksekliği                                              |
| $H'$           | Cam bloğunun kesim planına göre kullanılan yüksekliği                |
| $D$            | Cam bloğunun üzerindeki tüm kusurlar                                 |
| $D_i$          | Cam bloğunun üzerindeki i.kusur                                      |
| $K_n$          | Kesilmesi planlanan cam bloğundaki parçalara denk gelen kusur sayısı |



## KISALTMA LİSTESİ

---

|            |                                                                                                |
|------------|------------------------------------------------------------------------------------------------|
| BLF        | Bottom Left Fill Algorithm, Sol Alt Yerleştirme Algoritması                                    |
| BLF-BRF    | Bottom Left Fill and Right Fill Algorithm, Sağ Alt Sol Alt Dengelemeli Yerleştirme Algoritması |
| FF         | First Fit Decreasing Height Decreasing Width Algorithm, İlk Uygun Yer Algoritması              |
| FFDHDW     | First Fit Decreasing Height Decreasing Width Algorithm, İlk Uygun Yer Algoritması              |
| GA         | Genetic Algorithm, Genetik Algoritma                                                           |
| GA_F       | Genetic Algorithm with Flip Operator, Döndürme Operatörlü Genetik Algoritma                    |
| GA_LR      | Genetic Algorithm With Left Right Balancing, Sağ Sol Dengelemeli Genetik Algoritma             |
| LAFF       | Large Area First Fit                                                                           |
| NSGA II    | Non-dominated Sorting Genetic Algorithm 2, Bastırılmamış Sıralamalı Genetik Algoritma          |
| NSGA II LR | Non-dominated Sorting Genetic Algorithm 2 With Left Right Balancing                            |
| PAES       | Pareto Archived Evolution Strategy, Pareto Elde Etme Stratejisi                                |
| SPEA       | the Strength Pareto Evolutionary Algorithm, Kuvvetli Pareto Evrimsel Algoritması               |
| SPEA2      | the Strength Pareto Evolutionary Algorithm 2, Kuvvetli Pareto Evrimsel Algoritması 2           |

## ŞEKİL LİSTESİ

|                                                                                          | Sayfa |
|------------------------------------------------------------------------------------------|-------|
| Şekil 2. 1 Cam kesme makinesi .....                                                      | 5     |
| Şekil 2. 2 Kombinasyonel problemlerin kavramsal sınıflandırılması .....                  | 6     |
| Şekil 2. 3 Kombinasyonel problemlerin şekilsel sınıflandırılması .....                   | 6     |
| Şekil 2. 4 Yerleştirme şekillerine göre isimlendirme .....                               | 7     |
| Şekil 2. 5 Wascher[9] sınıflandırılması .....                                            | 10    |
| Şekil 3. 1 Sol Alt yerleştirme algoritması örnek kesim planı .....                       | 15    |
| Şekil 3. 2 Sol alt yerleştirme algoritmasının akış şeması.....                           | 16    |
| Şekil 3. 3 İlk uygun yer yerleştirme algoritmasının akış şeması.....                     | 17    |
| Şekil 3. 4 Alan öncelikli yerleştirme algoritmasının akış şeması .....                   | 18    |
| Şekil 3. 5 Sağ Alt - Sol Alt Dengelemeli Yerleştirme algoritması örnek kesim planı ..... | 20    |
| Şekil 3. 6 Sağ Alt - Sol Alt Dengelemeli Yerleştirme algoritması akış şeması.....        | 20    |
| Şekil 3. 7 Genetik algoritma akış şeması .....                                           | 22    |
| Şekil 3. 8 İkili kodlanmış kromozom.....                                                 | 24    |
| Şekil 3. 9 Reel kodlanmış kromozom .....                                                 | 24    |
| Şekil 3. 10 Permütasyon kodlanmış kromozom.....                                          | 25    |
| Şekil 3. 11 Ağaç kodlanmış kromozom .....                                                | 25    |
| Şekil 3. 12 Rulet tekeri algoritmasında 4 kromozomlu bir örnek .....                     | 29    |
| Şekil 3. 13 İkili kodlamada tek noktalı çaprazlama .....                                 | 30    |
| Şekil 3. 14 İkili kodlamada çok noktalı(uniform) çaprazlama .....                        | 31    |
| Şekil 3. 15 Permütasyon kodlamada tek noktalı çaprazlama .....                           | 31    |
| Şekil 3. 16 İkili kodlamada mutasyon.....                                                | 32    |
| Şekil 3. 17 Permütasyon kodlamada mutasyon.....                                          | 32    |
| Şekil 3. 18 Kaynaktan nektar toplamak ile görevli işçi arı.....                          | 33    |
| Şekil 3. 19 Yapay arı algoritması akış şeması .....                                      | 35    |
| Şekil 3. 20 Çok amaçlı optimizasyonun genel şeması .....                                 | 38    |
| Şekil 3. 21 Pareto verimliliğine göre çözümlerin gösterimi .....                         | 39    |
| Şekil 3. 22 NSGA II işlemleri .....                                                      | 42    |
| Şekil 3. 23 NSGA II Yoğunluk mesafesi gösterimi.....                                     | 43    |
| Şekil 3. 24 Sağ sol dengelemeli genetik algoritma örnek yerleşim planı.....              | 44    |
| Şekil 4. 1 Optimizasyon yazılımının arayüzü .....                                        | 45    |
| Şekil 4. 2 Jacobs'ın J1 ve J2 veri kümesinin optimum sonuçlarından birisi .....          | 46    |
| Şekil 4. 3 Sanayi veri kümesinin optimum sonuçlarından birisi.....                       | 48    |
| Şekil 4. 4 Jakops'ın veri kümelerinin sonuçları .....                                    | 50    |
| Şekil 4. 5 Sanayi veri kümesinin sonuçları .....                                         | 51    |
| Şekil 4. 6 Hopper T3a veri kümesinin örnek kesim planı .....                             | 52    |

Şekil 4. 7 Hopper T4a veri kümesinin GA ve NSGA-LR kesim planı karşılaştırması ..... 53

## ÇİZELGE LİSTESİ

---

|                                                                                   | Sayfa |
|-----------------------------------------------------------------------------------|-------|
| Çizelge 2. 1 Kesme ve Yerleştirme problemlerindeki karakteristik özellikler ..... | 8     |
| Çizelge 4. 1 Hopper'ın veri kümelerinin özellikleri .....                         | 47    |
| Çizelge 4. 2 ZDF veri kümelerinin özellikleri .....                               | 47    |
| Çizelge 4. 3 Algoritmalarda kullanılan parametreler .....                         | 49    |
| Çizelge 4. 4 Jakops'ın veri kümelerinin sonuçları .....                           | 49    |
| Çizelge 4. 5 Sanayi veri kümesinin sonuçları.....                                 | 51    |
| Çizelge 4. 6 Hopper veri kümesinin sonuçları .....                                | 52    |

## CAM KESME PROBLEMİNE ÇOK AMAÇLI YAKLAŞIM

Mustafa Zahid GÜRBÜZ

Matematik Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Doç. Dr. İbrahim EMİROĞLU

Cam üreticileri iki gruba ayrılabilir. Bunlardan biricisi, hammadde olarak toprağı kullanarak yüksek ısı altında onu büyük cam bloklarına dönüştüren üreticilerdir. Bir diğeri ise hammadde olarak birinci tip üreticilerin oluşturduğu cam bloklarını kullanarak istenilen ölçülerde kesim yapan üreticilerdir. Bu çalışmada bahsedilen cam üreticileri ikinci tip üreticilerdir.

Cam üreticileri, sanayideki diğeri üreticiler gibi maliyetleri azaltmak için büyük çaba harcamaktadırlar. Üretim sırasında oluşan hammadde maliyeti, işçilik maliyeti, yönetim giderleri, vb maliyetleri mümkün olduğunca azaltmaya çalışarak karlarını yükseltmek ya da satış fiyatlarını düşürmeye çalışmaktadırlar. Bu maliyetlerden biri olan hammadde maliyeti hem hammaddeyi satın alırken hem de kullanırken oluşur. Hammaddeyi tedarik ederken oluşan giderler birim fiyat, sipariş maliyeti, nakliye maliyeti, depolama maliyeti şeklinde özetlenebilir. Hammaddeyi kullanırken ise tüm hammeyi fire vermeden kullanmak çok önemlidir. Ancak üretimin doğası gereği üretim sırasında bir miktar hammadde, kullanılmadan atılmak ya da geri dönüşüme gönderilmek suretiyle fire olarak ayrılmaktadır. Hâlbuki fireyi azaltarak aynı miktarda ürünü daha az hammadde kullanımıyla üretmek hammadde maliyetinin büyük ölçüde azalmasına yardımcı olacaktır. Aynı zamanda daha az sayıda kesim işlemi yapılacağından az eforla daha fazla üretim sağlanarak enerji tasarrufu da sağlanmaktadır.

Cam üretiminde öncelikle büyük cam blokları şeklinde camlar istenilen kalınlıklarda tedarik ettikten sonra daha küçük siparişler bu cam bloklarından kesilerek elde

edilmektedir. İŖte bahsedilen fireler bu kesimler sırasında meydana gelmektedir. Eęer iyi hesaplamadan kesimler yapılırsa oluŖan fireler ok fazla olmaktadır.

Cam retilirken oluŖturulan cam blokları zerinde yer yer kusurlar meydana gelmektedir. Bu kusurlar optik kusur, izik, delik vs Ŗeklinde rneklendilebilir. Eęer kk paraları keserken elimizde bulunan cam bloęunda bu tr kusurlar varsa otomatik olarak kesilen sipariŖlerin bazılarında bu kusurlar bulunacaktır. Bu tr kusurların bulunması camın kalitesini etkiler. Dolayısıyla dŖk kalitede bir rn ya daha ucuza satılacaktır, ya da mŖteri memnuniyetsizlięi oluŖturacaktır. Hatta firmanın itibarına da glge dŖrebilir.

Bu tezde cam bloklarının kesimi sırasında oluŖan fireyi azaltacak uygun kesim planları oluŖturulmaya ynelik optimizasyon teknikleri kullanılmıŖtır. Bu oluŖturulan planlar cam bloęu zerinde hangi paranın hangi sırayla kesilmesi gerektięini planlamaktadır. Dolayısıyla birinci ama, fireyi azaltmak olarak dŖnlebilir. Bunun yanı sıra kesilecek paralarda, cam bloęu zerindeki kusurlara denk gelmeyecek Ŗekilde planlama yapılırsa daha kaliteli bir retim yapılabilir. Bylece ikinci ama, kesim planında paralara denk gelen kusur sayısını mmknse sıfıra indirmek yoksa da en az sayıya indirmek olacaktır.

Bu alıŖmada ncelikle problem tek amalı olarak modellenmiŖtir. Bu tek amalı problem iin nceki alıŖmada kullanılan yntemler incelenmiŖ ve sonuları elde edilmiŖtir. Ardından problemi ok amalı modelleyerek yeni bir zm yntemi nerilmiŖtir. Mevcut yntemlerle karŖılaŖtırılarak nerilen algoritmanın etkinlięi gsterilmiŖtir.

**Anahtar Kelimeler:** ok amalı optimizasyon, cam kesme, kusur optimizasyonu, fire optimizasyonu.

## **A MULTIOBJECTIVE APPROACH TO GLASS CUTTING PROBLEM**

Mustafa Zahid GÜRBÜZ

Department of Mathematical Engineering

Ph.D. Thesis

Advisor: Assoc. Prof. Dr. İbrahim EMİROĞLU

Glass manufacturers can be divided into two major groups. One of these manufacturers is converting to big glass blocks of their raw materials under high temperature using the soil. Another one is cutting the glass block that was formed by first such manufacturer into small pieces with desired sizes. In this study, the second type of manufacturer is considered.

Glass manufacturers, is an intense effort to reduce costs as well as other manufacturers in the industry. They try to reduce the cost of raw materials during the production, labor costs, management costs, and so on to increase their profits or to reduce their selling price. The cost of row materials occurs when buying it and when using it. Raw material supplying cost can be summarized as the unit price, the ordering cost, the shipping cost and the storage costs. Raw material using cost is producing with waste. Wasteless production is very important. But as the nature of the production process, some of the raw material cannot be used efficiently. However, minimum waste results more product with the same amount of raw material. This reduces the cost of raw material significantly. Besides, the cutting operations can be reduced by optimizing waste; as a result it can be an energy efficient process.

After the big glass blocks supplied with desired sizes, the small pieces are cut from that blocks. The mentioned waste occurs in that production phase. If the planning cannot be done efficiently, the waste amount increases.

Some defects can be formed while producing from the soil. These defects can be optical, scratch, hole, branch, etc. If the defects are exists on the glass block, so the small pieces produced by it consists that defects. These defects are effects the quality of the product. So the low quality product means that it can sell cheaper or it can undermine the manufacturer's repetition.

In this study, the cutting plan is determined by using optimization methods. The sequence of cutting process of the small pieces is gathered by these methods. So the first objective is minimizing the waste. In order to increase the quality, number the defects that hits the small pieces must be reduced. So the second objective is minimizing the number of defective products.

In this study, at first, the problem stated as single objective problem. The previous methods are examined and results are saved. After that, the problem is stated as multi objective problem. The Proposed methods get the solution as both single and multi objective perspective. Comparing the previous methods, the efficiency of the proposed method is shown.

**Keywords:** multi objective optimization, glass cutting, defect optimization, waste optimization.



#### 1.1. Literatür Özeti

Cam kesme probleminin temeli mevcut alana sığabilecek en fazla sayıda parçayı kesebilecek planı oluşturmak olarak düşünülebilir. Bu problem literatürde farklı alanlardaki problemlerle benzerlik göstermektedir. Bu tip problemler literatürde kesme ya da paketleme problemleri olarak bilinirler. Çok geniş bir uygulama alanı olduğundan birçok disiplinde de bu probleme yer verilmiştir. Bunlardan birkaçı; yönetim bilimi, mühendislik, matematik, bilgisayar bilimi ve yöneylem araştırmasıdır.

Kesme problemleri, karmaşıklığı yüksek, kombinasyonel problemlerdendir. Problemin algoritmik karmaşıklığı, “NP-Complete” sınıfındadır. Yani, problemin çözümü kabul edilebilir bir zaman aralığında hesaplanamaz. Algoritmanın çalışma süresi parça sayısı arttıkça üstel olarak artar.

Cam kesme uygulamasında yapılan işlem temel olarak dikdörtgensel cam bloğundan istenilen ölçülerde küçük cam parçaları kesmek olarak tanımlanır. Bu kapsamda literatürdeki bazı problemler ile benzerlik göstermektedir. Bu problemler, tekstil firmalarında uygulanan kumaş kesme, karton kutu üreticilerindeki kâğıt kesme, alüminyum levhaların kesilmesi, mobilyacılarda tahtaların kesilmesi şeklinde örnekler çoğaltılabilir. Benzer şekilde konteynıra paket yerleştirme, çelik blokların kesilmesi, gibi örnekler de aynı tip problemin üç boyutlu halleridir. Bunun yanı sıra günümüzde internetin gelişmesiyle birlikte yeni problemler de ortaya atılmıştır. Bunlardan en bilineni internet sayfalarına farklı boyutlardaki dikdörtgensel yapıdaki makalelerin, haberlerin ya da köşe yazılarının uygun bir şekilde yerleştirilmesidir.

Çözüm yaklaşımlarını iki grupta inceleyebiliriz. Birincisi deterministik, ikincisi ise sezgisel(heuristic)dir. Deterministik yöntemler optimum sonucu bulmayı garanti eder. Deterministik yöntemler çoğunlukla lineer programlama yaklaşımı ya da dal-sınır veya dinamik programlama gibi sıralamaya dayalı yöntemlerden oluşur. Bu metodlar Sedgewick [1] tarafından daha ayrıntılı incelenmiştir. Bu problemlerin yüksek karmaşıklığa sahip olanlarında deterministik yöntemler etkisiz kalıyor. Çünkü makul bir zamanda sonuç üretemiyorlar. Bu sebeple sezgisel metotlar öne sürülmüştür. Sezgisel metotlarda hızlı bir şekilde çözüm hesaplarlar fakat deterministik yöntemlerde olduğu gibi optimum sonucu garanti etmezler. “Yeterince iyi” yada “optimuma yakın” şeklinde ifade edilirler. Sezgisel yöntemler genellikle problem spesifiktir. Yani her problemde varolan kısıtlara göre yeniden düzenlenmesi gereklidir. Heuristik metotlar bazen local optimum noktalarına takılırlar. Oysaki deterministik yöntemler global optimum bulma üzerine işlem yaparlar. Sezgisel tekniklere örnek olarak sol-alt algoritması, en uygun yer algoritması, genetik algoritma ve yapay arı algoritması verilebilir.

## **1.2. Tezin Amacı**

Literatürde kesme problemleri üzerine çok çalışılmıştır. Birçok yöntem önerilmiş ve değişik varsayımlarla sonuca gidilmiştir. Ancak bunların optimum olduğunu söylemek zordur. Çünkü optimum sonucu bulmak için tüm ihtimallerin denenmesi gerekmektedir ki bu kez de kesme problemlerinin hacminin büyük olması dolayısıyla kabul edilir bir zamanda sonuca ulaşmak mümkün değildir.

Çok amaçlı evrimsel algoritmaların cam kesme problemi üzerine uygulanarak daha verimli sonuçlar elde etmek amaçlanmaktadır. Bu sayade cam üreticilerinin müşterilerine daha uygun fiyattan ürün sağlayabilmesi amaçlanmaktadır. Aynı zamanda cam bloğundaki firenin en aza indirgenmesi sayesinde geri dönüşüme olan ihtiyaç azaltılmaktadır.

Tez kapsamında, probleme yeni bir boyut kazandırılmıştır. Bu boyut üretimin kalitesidir. Kalite bir üretimde firmalara hem itibar kazandırır hem de rekabet gücü kazandırır. Aynı zamanda ürünlerini daha yüksek kar payıyla pazarlama esnekliği sağlar.

### **1.3. Orjinal Katkı**

Literatürde bu konuda yapılan çalışmalar çoğunlukla tek amaçlı olarak ele alınmıştır. Bu projede çok amaçlı bir model üzerinden problem çözüme kavuşacaktır. Aynı zamanda önerilecek olan ikinci amaç daha önce üzerinde çalışılmamıştır. Bu ikinci amaç, hazırlanan siparişlerdeki kusurların en aza indirgenmesidir. Cam üzerindeki kusurlar siparişin kalitesi için önem arz etmektedir.

Tez çalışmamda, cam üretiminde oluşabilecek kusur tipleri ve sınıfları için sanayi kolunda kabul edilmiş dokümanlardan yararlanılmıştır. Ayrıca üretim kusurları ve sipariş bilgileri gibi veritabanı kayıtları da yine sanayideki işyerinden temin edilmiştir. Ayrıca daha önceki çalışmalarda kullanılan veritabanı kayıtları da kullanılmıştır. Konuyla ilgili yaklaşılacak metotlara temel teşkil edecek olan makaleler, bildiriler, kitaplar, doktora ve yüksek lisans tezleri kullanılmıştır.

Bu projede çok amaçlı elitist evrimsel algoritmalar kullanılarak çözüm elde edilmeye çalışılmıştır. Cam kesme için daha önce kullanılan sol alt algoritması, ilk uygun yer algoritması, genetik algoritma ve benzer algoritmaların uygulanması yapılmıştır. Bu algoritmaların verimliliği elde edilen veriler için çalıştırılıp önerilen yeni algoritma ile karşılaştırılmıştır.

### CAM KESME PROBLEMİ

Cam kesme problemi, cam üreticilerinin her zaman üzerinde durduğu bir problemdir. Şöyleki bir firmanın varoluş temel amacı para kazanmak olduğundan, kar elde edebilecekleri her yol firmalar için büyük önem taşır.

Cam kesme probleminde asıl amaç ortaya çıkan firenin en aza indirgenmesidir. Ancak iyi planlanmamış bir üretim her zaman yüksek fire ile üretimi yapmak anlamına gelir. Bu da firmaların karlılığını etkileyen bir durumdur.

Cam üretimi yapan firmaların ürünleri temel olarak iki kategoriye ayrılır. Birisi, şişe, bardak, vb şekillere sahip olan, düzgün olmayan şekilli cam üretimidir. Diğeri ise pencere, ayna, vb gibi düz cam üretimidir[2].

Cam kesme uygulamasında yapılan işlem temel olarak dikdörtgensel cam bloğundan istenilen ölçülerde küçük cam parçaları kesmek olarak tanımlanır. Bu kapsamda literatürdeki bazı problemler ile benzerlik göstermektedir. Bu problemler, tekstil firmalarında uygulanan kumaş kesme, karton kutu üreticilerindeki kâğıt kesme, alüminyum levhaların kesilmesi, mobilyalarda tahtaların kesilmesi şeklinde örnekler çoğaltılabilir. Benzer şekilde konteynıra paket yerleştirme, çelik blokların kesilmesi, gibi örnekler de aynı tip problemin üç boyutlu halleridir. Bunun yanı sıra günümüzde internetin gelişmesiyle birlikte yeni problemler de ortaya atılmıştır. Bunlardan en bilineni internet sayfalarına farklı boyutlardaki dikdörtgensel yapıdaki makalelerin, haberlerin ya da köşe yazılarının uygun bir şekilde sayfaya yerleştirilmesidir. Kesme ve yerleştirme problemlerinin herkes tarafından kabul görülen varsayımı parçalar kesim planına sokulduğunda plan üzerine yerleştirilen parçalarda üstüste gelme kesinlikle olmamalıdır.

Literatürde bu problem kesme ve yerleştirme problemleri olarak yer bulmaktadır. Bu konuda birçok bilim adamı çalışmıştır. Lodi vd. [3] iki boyutlu paketleme problemleri için (ortogonal kesim için  $90^\circ$ ) rotasyonu ve giyotin kesim gibi kesme çeşitlerini de ele almıştır. Bu konudaki çalışmalar Dyckhoff ve Finke [4], Dowsland and Dowsland [5] ve Dyckhoff vd.[6] tarafından etüt edilmiştir.



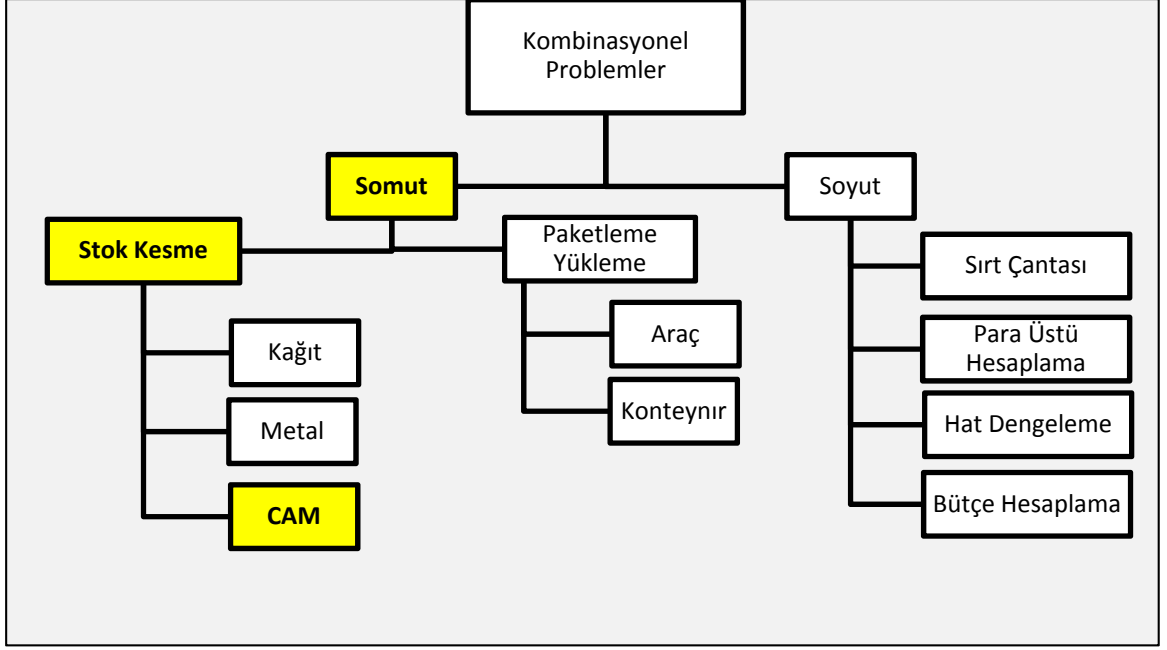
Şekil 2. 1 Cam kesme makinesi

## 2.1. Problemin Sınıflandırılması

Kesme ve yerleştirme probleminin sınıflandırılmasında 1990 yılında Dyckhoff [7] tarafından sunulan sistematik bir sınıflandırma kabul görmüştür. Dyckhoff kesme ve paketleme problemlerini öncelikle iki grup altında ele almaktadır. Üç boyutlu öklit uzayda ifade edilebilen problemler ( $R^n \mid n \leq 3$ ) ve ifade edilemeyen problemler ( $R^n \mid n > 3$ ) olarak ayrılır. Bu soyut problemler zamana bağlı, ağırlıklandırmaya bağlı, finansal değere bağlı şeklinde probleme yeni boyutların kazandırılması ile oluşturulur.

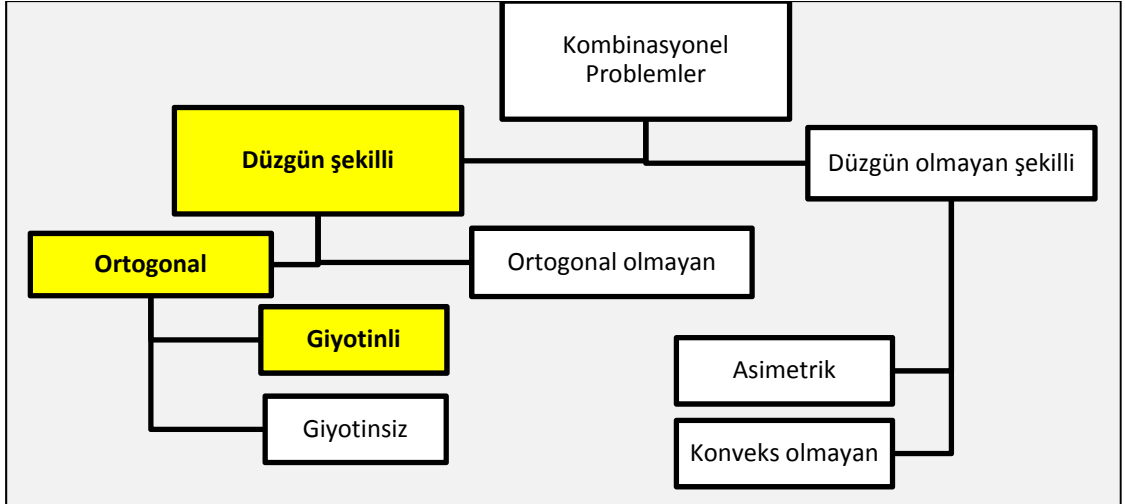
- Somut: Kesme, paketleme ve yükleme problemleri, vs...
- Soyut: sırt çantası problemi, zamanlama(scheduling) problemi, vs..

Cam kesme problemlerinin bu sınıflandırma içindeki yeri Şekil 2.2 de gösterilmiştir.



Şekil 2. 2 Kombinasyonel problemlerin kavramsal sınıflandırılması

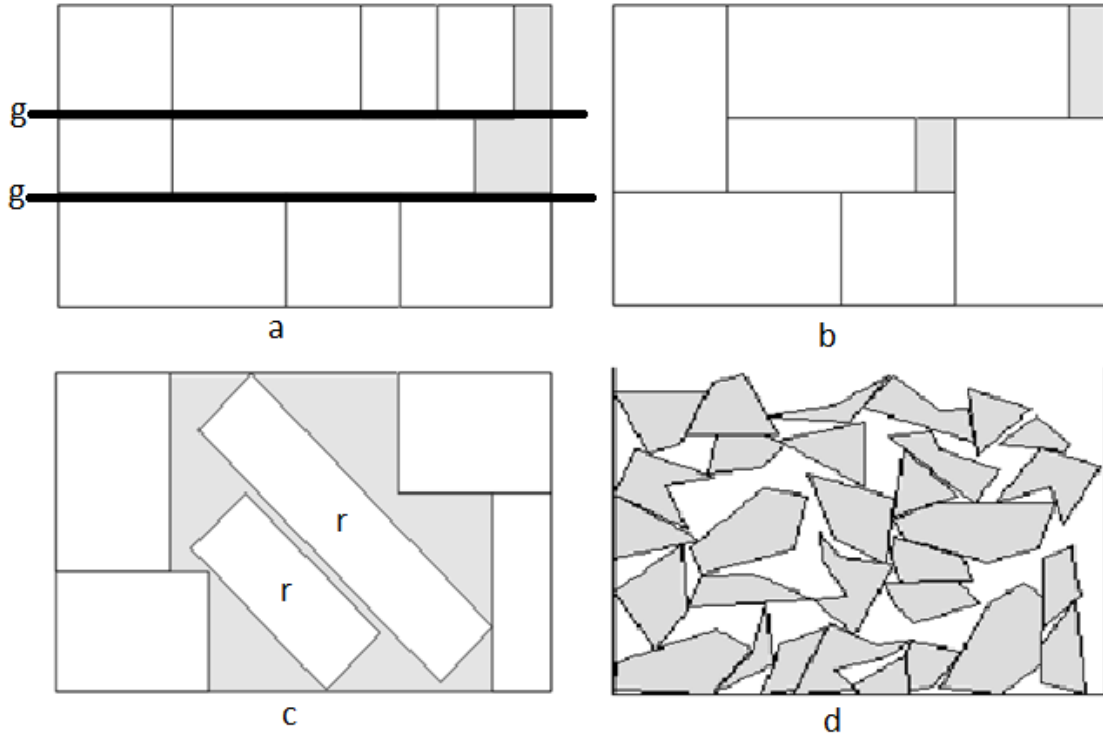
Kesme ve yerleştirme problemleri kesilen cismin şekliyle doğrudan ilişkilidir. Düzgün şekilli (dikdörtgen, yuvarlak) ve düzgün olmayan şekilli (asimetrik, konveks olmayan) cisimler diye iki kısma ayırabiliriz. Düzgün şekilli olanları da ortogonal ve ortogonal olmayan diye ayırabiliriz. Ortogonal olanları da Giyotinli ve giyotinsiz olarak ayırabiliriz. Problemin şeması Şekil 3.4’de gösterilmektedir.



Şekil 2. 3 Kombinasyonel problemlerin şekilsel sınıflandırılması

Bu çalışmada Düzgün şekilli, ortogonal, giyotinli kesme problemi üzerine çalışılmıştır. Düzgün şekilli parçalar, simetrik ölçülere sahip konveks şekillerdir. Dikdörtgen, daire, eşkenar üçgen, vb buna örnek verilebilir. Ortogonal parçalar ise dikdörtgen ya da kare

gibi 4 köşe ve 4 kenara sahip parçalar için kullanılmaktadır[8]. Bu parçaların yerleştirilen zemine paralel ya da dik olacak şekilde yerleştirilmesi gerektiğini ifade eder. Şekil 2.4 de parçaların şekillerine ve yerleştirme şekillerine göre aldığı isimlendirme görülmektedir. Şekilde 2.4'deki a kısmında parçaların giyotinli ve ortogonal kesme, b kısmında ortogonal ve giyotinsiz kesme, c kısmında ortogonal olmayan kesme, d kısmında ise düzgün olmayan şekilli kesme probeminin çözümüne örnekler gösterilmiştir. g ile belirtilen 2 adet giyotin kesim çizgisi görülmektedir. Giyotinli kesimde kesim işlemi bir uçtan başlayarak parçanın sonuna kadar hiç kesintiye uğramadan kesme işlemine devam eder. r ile belirtilen parçalar ise büyük parçanın tabanına dik yada paralel olarak kesilmediği için ortogonal olmayan bir kesim olarak adlandırılmıştır.



Şekil 2. 4 Yerleştirme şekillerine göre isimlendirme

Dyckhoff'un sınıflandırmasında 4 önemli karakteristik özellikten bahsedilmektedir. Bu karakteristik özellikler boyut, atama şekli, büyük objelerin çeşitliliği, küçük objelerin çeşitliliği olarak Çizelge 2.1'de belirtilmiştir.

Çizelge 2. 1 Kesme ve Yerleştirme problemlerindeki karakteristik özellikler

| Karakteristik                 | Sembol | Tanım                                                             |
|-------------------------------|--------|-------------------------------------------------------------------|
| <b>Boyut</b>                  | 1      | 1 boyutlu                                                         |
|                               | 2      | 2 boyutlu                                                         |
|                               | 3      | 3 Boyutlu                                                         |
|                               | N      | N boyutlu, $N>3$                                                  |
| <b>Atama cinsi</b>            | B      | Büyük Objelerin hepsi, Küçük parçaların bir kısmı kullanılabilir. |
|                               | V      | Büyük Objelerin bir kısmı, Küçük parçaların hepsi kullanılabilir. |
| <b>Büyük objelerin çeşidi</b> | O      | Tek Obje                                                          |
|                               | I      | Aynı Ojeden birden fazla                                          |
|                               | D      | Farklı objelerden birden fazla                                    |
| <b>Küçük objelerin çeşidi</b> | F      | Değişik şekillerde az parça olması                                |
|                               | M      | Değişik şekillerde çok parça olması                               |
|                               | R      | Göreceli olarak az değişik şekillerde çok parça olması            |
|                               | C      | Benzer şekillerde olması                                          |

Bu karakteristik özelliklere sahip bazı problem örnekleri aşağıda verilmiştir.

**Sıralama Problemi:** Bazen üretim kapasitesi yetmediğinde bazı ürünlerin üretilirken bazılarının bekletilmesi gerekir. Bu problemde bekletilmesi gereken ürünlerin tespiti için uğraşılır.

**Kutu Yerleştirme Problemi:** Malzemeleri kutulara yerleştirmek, bunu yaparken de en az yüksekliğe sahip olacak şekilde doldurma problemidir.

**Malzeme Kesme Problemi:** Kesme yerleştirme problemleri için genel bir terimdir. Elde bulunana büyük parçalardan istenilen boyutta küçük parçalar kesme ve bunu yaparken de en az fire verecek şekilde kesme problemidir.

**Örtü Problemi:** Kalıpları bir yüzey üzerine dizme problemidir. Genellikle kumaş endüstrisi için uygulanır.



**Yükleme Problemi:** Üç boyutlu mekânlara yine üç boyutlu malzemelerin yerleştirilmesi problemidir. Palet, tır, konteynır yerleřtirmeler bu tipe örnektir.

**Sırt Çantası Problemi:** Hangi nesnelerin alınması gerektiğine karar veren bir problemdir. Nesnelerin değerlerine göre öncelik verilir. Kutu yerleřtirme problemine benzer.

**İç İçe Geçme Problemi:** Düzensiz şekilli nesnelerin paketlenmesi problemidir. Denizaltı inřaasında kullanılır.

**Ortogonal Paketleme Problemi:** Dikdörtgen parçaların tabanları yerleřtirilen yerin tabanına paralel olacak şekilde yerleřtirilmesine dayalı bir problemdir.

**Şablon Yerleřtirme Problemi:** Kesilecek malzemelerin sayısını önemsemeden belirli bir alandan ne kadar çok malzeme şablonu yerleřtirilebileceğini hesaplayan problemdir. Bozuk para üretimi için metal levha üzerinden en fazla sayıda paranın çıkartılabilmesi örneğı verilebilir.

Ayrıca Wa"sscher da dychkoff'un belirttiğı özelliklerde bazı değıřiklikler yapmış ve yeni karakteristik özellikler eklemiřtir[9] . Wascher'e göre bu problemler ařağıdaki 5 karakteristik özelliğe sahiptir.

**Boyut:** (n) Literatürde 1,2 ve 3 boyutlu problemlerle karřılařılmaktadır. Problemdeki (n>3) olan durumlar bu tip problemlerin bařka bir türüdür.

**Atama Şekli:** Burada girdi minimizasyonu ya da çıktı maksimizasyonu olarak iki tür düşünölebilir. Çıktı maksimizasyonunda, büyük parçalar, küçük parçaların tamamını kapsayacak şekilde olmadığı durumda büyük parçanın içerisine daha fazla küçük parça sığdırmaya çalışılmaktadır. Burada bazı parçalar dıřarıda kalacağı için hangisinin kesim planına dâhil edileceğinin tespit edilmesi lazım. Girdi minimizasyonunda ise küçük parçaların tamamı büyük parçaya sığmaktadır. Bu durumda büyük parçadan en az yeri kullanarak kesim planı oluřturmak gerekir.

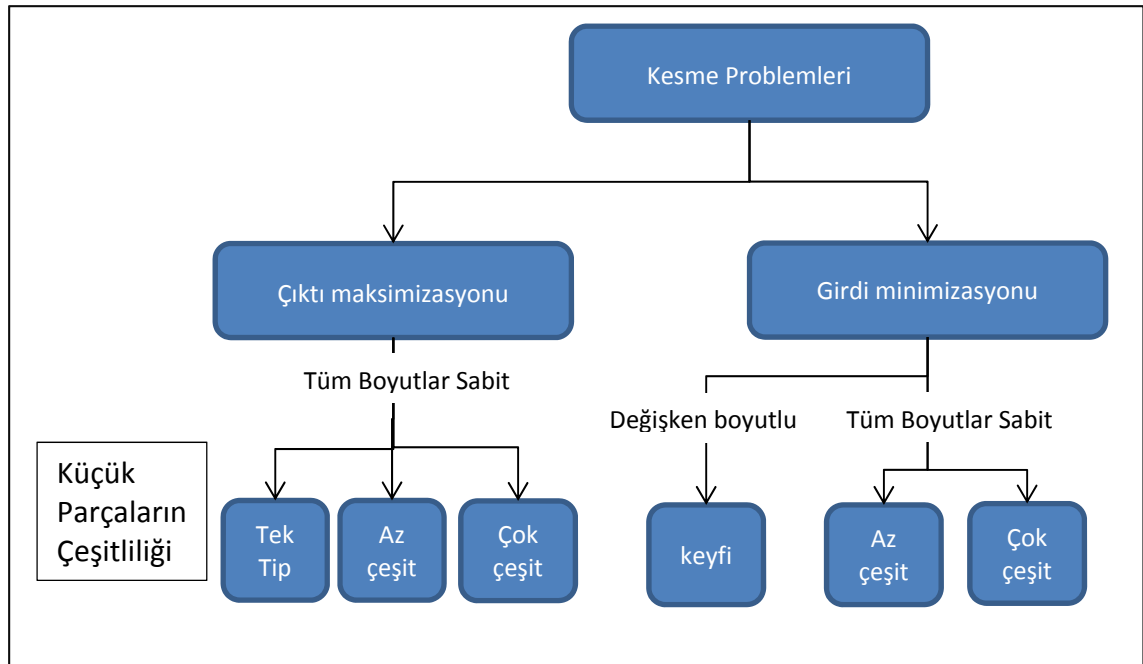
**Küçük parçaların çeřitliliğı:** burada üç çeřitten bahseilmektedir. Birincisi, tek tip parçalardan oluřan kesmedir. Burada parçanın talebinin sonsuz olduğı varsayılır ve cam bloğuna ne kadar çok yerleřtirilebileceğini hesaplayan problemdir. İkincisi az sayıda küçük parça çeřitinin olmasıdır. Burada benzer parçalar gruplanarak az sayıda

grup oluşturulur. Üçüncü çeşit ise çok çeşitli küçük parçaların olmasıdır. Burada gruplama yapıldığında çok grup oluştuğu için her parça bağımsız olduğu varsayılarak işlem yapılır.

**Büyük parçaların çeşitliliği:** Büyük parçalar iki türlü olabilir. Tek bir büyük parça, küçük parçaların hepsi bu büyük parça üzerine yerleştirilir. Büyük parçanın yüksekliği sınırsız kabul edilir. Birden fazla büyük parça probleminde küçük parçaları kesmek için birden fazla büyük parça bulunmaktadır.

**Küçük parçaların şekli:** Küçük parçalar, düzgün şekilli ya da düzgün olmayan şekilli olarak iki kategoride incelenir.

Yukarıda anlatılan sınıflandırma, şekilde özetlenmiştir.



Şekil 2. 5 Wascher[9] sınıflandırılması

## 2.2. Problemin Karmaşıklığı

Kesme ve yerleştirme problemleri az sayıda parça için düşünüldüğünde bütün ihtimallerin sırasıyla denemesi yoluyla kolaylıkla bulunabilir. Ancak parça sayısı arttıkça bu olasılıkların sayısı çok arttığı için bu yolla problemi çözmek imkânsızlaşır.

Kesme problemi bir sıralama problemi olduğundan algoritmik karmaşıklığı permütasyon ile hesaplanır. Büyük O notasyonunda karmaşıklık değeri (2.1) de verilmiştir.

$$O(P(n)) = O(n!) \quad (2.1)$$

Problemin karmaşıklığı NP-Tam sınıfındadır. NP-Tam sınıfı problemin polinomsal zamanda çözülebileceğini ama bu zamanda çözmenin çok zor olacağını belirten bir sınıftır. [10]

### 2.3. Problemin Matematik Modeli

Bu problemi bir tamsayılı doğrusal programlama problemi şeklinde modelleyebiliriz. Bu problemde amaç fireyi minimize etmek olduğu için amaç fonksiyonu da (2.2) denklemindeki gibi olur. Burada  $w_i * h_i$  değerleri kesilen küçük parçaların boyutlarıdır.  $W$  değeri cam bloğunun genişliğidir. Bu genişlik sabittir. Ancak cam bloğunun yüksekliği değişken olduğundan tüm yüksekliğe ( $H$ ) kadar değil de kesilen kısmın bittiği noktadaki yükseklik olan  $H'$  değeri kullanıldı. Neticede, amaç fonksiyonu, kesilen parçaların toplam alanını kullanılan cam bloğunun alanına bölerek elde edilen  $[0,1]$  aralığındaki değerin minimize edilmesidir (denklem 2.2). Bu çalışmada yükseklik sınırı olmadığı için istenen bütün parçalar kesilecek şekilde model kurulmuştur. Bir diğer amaç ise kesilen parçalara denk gelen kusurların en aza indirgenmesidir (denklem 2.3). Bu sebeple  $S_i$  parçalar kümesinin  $i$ . elemanı ile  $D$  kusur kümesinin kesişimindeki tüm elemanların sayısını toplayarak elde edilen sonucu minimize ederek amaca ulaşılır.

#### Amaç fonksiyonu;

$$(1. \text{amaç}) \min \text{ fire} \rightarrow \min \frac{W * H' - \sum_i \varphi(S_i) * w_i * h_i}{W * H'} \quad (2.2)$$

$$(2. \text{amaç}) \min \text{ kusur} \rightarrow \min \sum_i S(D \cap S_i) \quad (2.3)$$

#### Kısıtlar;

Herhangi bir kısıt yoktur. Ancak problemde tüm parçaların yerleştirilmesi istendiği için aşağıdaki kısıtı ekleyebiliriz.

$$\forall \varphi(S_i) = 1, \quad i = 1, 2, \dots, n \quad (2.4)$$

Burada  $S_i$  siparişteki her bir parçayı,  $\varphi$  fonksiyonu ise siparişteki her bir parçanın kullanılmasına göre değer alır. Eğer parça kesim planında varsa 1, yoksa 0 değerini alır. Burada her parçanın 1 olmasını kısıtlara konularak tüm parçaların kesim planına yerleştirmesi hedeflenmiştir.

$$\varphi(S_i) = \begin{cases} 1 & , i. parça kesim planında mevcut \\ 0 & , i. parça kesim planında mevcut değil \end{cases} \quad (2.5)$$

## 2.4. Çözüm yaklaşımları

NP-Tam sınıfındaki problemlerin çözüm yolları iki ana kategoride incelenir. Bunlardan birincisi deterministik yöntemler, diğeri de sezgisel yöntemlerdir.

### 2.4.1. Deterministik Yöntemler

Hesaplamaya dayalı (deterministik) yöntemler çözüm için olası ihtimalleri deneyerek sonuca ulaşmaya çalışırlar. Bu yöntemlerde ortaya çıkan sonuç optimum sonucu verir. Bu kategorideki yöntemler bazı özel arama yöntemleriyle tüm olasılıkları denemek yerine, imkânsız olasılıkları atlayan bir takım hesaplamalar yaparak çalışma süresini azaltmaya çalışırlar.

#### Avantajları:

- Optimum sonucu garanti ederler.
- Local optimum noktalarına takılmazlar.

#### Dezavantajları:

- Çalışma süreleri fazladır. Problemin ölçeği büyüdükçe çözülemez hale gelirler.

Deterministik yöntemler, lineer programlama[11] [12], tam sayılı programlama[13], dinamik programlama[14], dal ve sınır algoritması[13] olarak örneklendirilebilir.

### 2.4.2. Sezgisel Yöntemler

Çalışma sürelerinin fazla olması sebebiyle deterministik çözümler ihtiyaçları karşılayamaz hale geldiğinden bir takım varsayımlar ile etkili çözümler elde edilmeye

alıřılmıştır. Bu yöntemlerde optimum sonuç garanti edilmez. Hatta optimum sonucu bulduğu bilinemez. Ufak sıçramalarla yada komşuluklarda çözüm aranır, eğer komşuluktaki çözüm daha iyi bir çözüm ise o noktadan çözüm bulmaya devam edilir. Bu sebeple bu tip algoritmalar yerel optimum noktalarına takıldığında oradan çıkamayabilir, en iyi çözümü o olarak belirlerler.

Sezgisel çözümler problemin kendi kısıtları haricinde bir takım ek kısıtlar ya da varsayımlar daha eklenerek çözüm üretirler.

**Avantajları:**

- Belirli bir süre içinde çözüm sunarlar.

**Dezavantajları:**

- Optimum çözümü garanti etmezler.
- Yerel optimuma takılabilirler

Her ne kadar dezavantajları olsa da önemli olan hızlı bir şekilde çözüm üretmek ise bir miktar optimumluktan ödün vererek sonuç verirler. Bu sebeple sezgisel yöntemler NP-Tam problemleri için büyük önem taşır.

Sezgisel yöntemler, Sol-Alt [15], Sol-Alt Doldurmalı [16] [17], İlk Uygun Yer [18] [19],genetik algoritma [20] [21], Yapay Arı Algoritması[22], bastırılmayan sıralamalı genetik algoritma [23] örnek olarak verilebilir.

Bu çalışmada sezgisel yöntemler üzerine çalışılmıştır. Bir sonraki bölümde bu sezgisel yöntemler hakkında detaylı bilgi verilmiştir.

### CAM KESME PROBLEMİNDE SEZGİSEL YÖNTEMLER

#### 3.1. Yerleştirme Yöntemleri

Bu tip algoritmalar sezgisel bir yöntem ile bir seferde cam bloğu üzerine küçük parçaları işaretlerler, ve hızlı bir şekilde bir sonuç üretirler. Bu sonuç optimum sonuç olup olmadığı hakkında bilgi verilmez. Ancak yapılan çalışmalar sonucunda optimum olmasa da etkili bir yöntem oldukları ispatlanmış çözümler sunarlar.

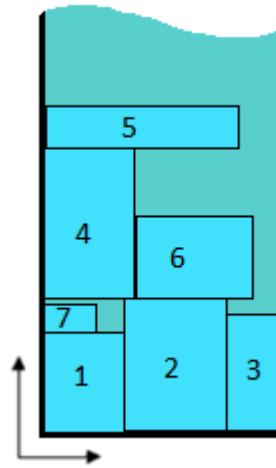
##### 3.1.1. Sol Alt Yerleştirme Algoritması (BLF)

###### 3.1.1.1. Tanım

Sol Alt Yerleştirme Algoritması, parçaları kesmeye sol alt kısımdan başlayacak şekilde kesim planı oluşturur. Eğer blok camda yeterli alan mevcutsa parça camlar her seferinde sol alttan yerleştirilerek kesme işlemi gerçekleştirilir[24]. Bu algorithmada yerleştirilecek parça en son yerleştirilen parçanın yatay hizasına sığıyorsa oraya yerleştirilir, sığmıyorsa üstüne yerleştirilir. Algoritmanın karmaşıklığı  $O(n^2)$ 'dir. Burke vd. [25] bu algorithmaya doldurma özelliği ekleyerek daha etkin bir yerleştirme sunmayı amaçlamıştır. Bu doldurma özelliğine göre Her zaman en son yerleştirilen parçanın değil daha önce yerleştirilen parçaların da yanındaki müsait yerler kontrol edilmektedir. Bu kapsamda algoritmanın çalışma süresi biraz daha artmaktadır. Algoritmanın karmaşıklığı  $O(n^3)$  olmuştur [26].

Şekil 3. 1'de BLF algoritması kullanılarak oluşturulmuş bir kesim planı gösterilmektedir. 1'den 7'ye kadar numaralı parçalar sırasıyla kesilmek istenmektedir. Burada 7

numaralı parça 1 ve 4 ün arasında kalan boşluğa sığıdığı için oraya yerleştirilmiştir. 6 numaralı parça da aynı şekilde boşluğa yerleştirilmiştir.



Şekil 3. 1 Sol Alt yerleştirme algoritması örnek kesim planı

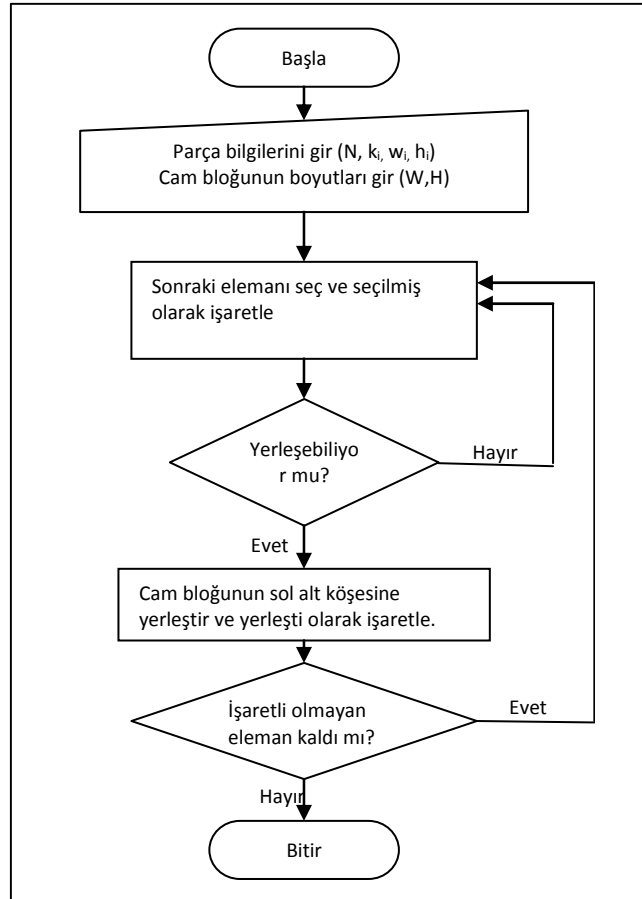
#### 3.1.1.2. Akış Şeması

Algoritma çalışması için gerekli parametreler sağlanır.

- N: Girilecek parça sayısı
- k: her parçadan kaç adet sipariş verildiği
- $w_i$ : i.parçanın genişliği
- $h_i$ : i.parçanın yüksekliği
- W: cam bloğunun genişliği
- H: cam bloğunun yüksekliği

Bu algorithmada parçalar siparişlerin geliş sırasına göre yerleştirilir. Yerleştirme işlemine sol alttan başlanarak yapıldığı için oluşacak fireler doğal olarak sağda ve üstte olacak şekilde bir plan oluşturulur. Kusurlar göz önüne alındığında eğer kusurların yoğunluğu sağ ve üst bölgelerde fazla ise o zaman kusur yönünden daha iyi sonuçlar ortaya çıkar. Kusurların yoğunluğu sol ve alt bölgelerde fazla ise bu kez algoritma sonucu çıkan planı sağ üst köşeden başlayarak kesim işlemi yapılırsa optimuma daha yakın bir sonuç elde edilir. Ancak algoritmanın içerisinde cam bloğunun yönünü değiştirecek herhangi bir operatör bulunmadığı için bu kısımda yetersiz kalmaktadır gibi görülse de cam bloğunu kesim makinesine yerleştirirken uygun düzende (yani kusur yoğunlukları sağ üst bölgelerde fazla olacak şekilde) yerleştirildiğinde algoritmanın içeriğinin değiştirilmesine gerek olmadan daha iyi sonuçlar elde etmek mümkün olur.

Cam bloğunun yüksekliği olan  $H$  değeri algoritmaya parametre olarak verilir. Algoritma çalışırken cam bloğunun sınırları içinde kalacak şekilde parçalara kesim planı oluşturur. Ancak bu tez çalışmasında kullanılan problem için cam yüksekliği sınırsız olarak varsayılarak işlemler yapılmıştır. Bu durumda  $H$  değerini yeterince büyük bir sayı girerek bütün sipariş parçalarının cam bloğundan kesilebileceği öngörülmüştür. Bu sebeple yerleşmeyen parçalar listesi bu algoritmanın sonuçlarından çıkartılmıştır.



Şekil 3. 2 Sol alt yerleştirme algoritmasının akış şeması

BLF Algoritmasında, parçalar verilen sırada sol alt kısımdan başlayarak yerleştirilir. Bu algoritmanın daha verimli olması için parçaların sırası da önemlidir. Parçaları yerleştirmeden önce belirli bir sezgisel yöntemle sıralanırsa o zaman çözümü iyileştirecek yönde ilerlenebilir. Bir sonraki kısımda anlatılan ilk uygun yer algoritması ve alan öncelikli sıralama algoritması, yerleştirmeden önce sıralama yapan bir sezgiseldir.

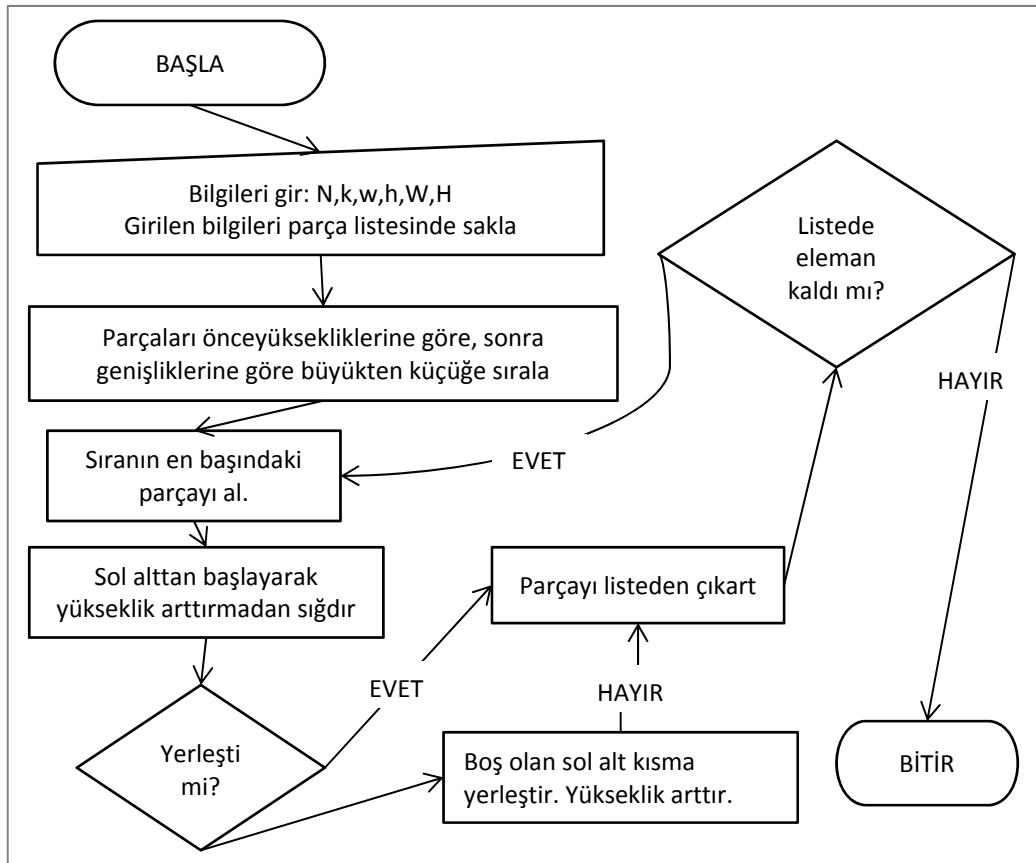


### 3.1.2. İlk Uygun Yer Yerleştirme Algoritması (FF)

#### 3.1.2.1. Tanım

Büyük cam bloğundan kesilecek olan parçaların konumu belirlenirken öncelikle küçük parçaların genişliğe göre, ardından yüksekliklerine göre sıralanması esasına dayanır. Baker [27], Johnson'dan destek alarak algoritmaya yeni bir yön kazandırmıştır. Kesim planında yükseklik arttıran parçaları sona bırakarak daha etkili bir çözüm sunmuştur. Bu algoritma Korf [28] tarafında kesme problemleri için kullanılmıştır. Bir boyutlu problemler için algoritmanın karmaşıklığı Johnson vd. [29] tarafından  $O(n \cdot \log n)$  olarak gösterilmiştir. İki boyutlu kesme problemlerinde kullanılan bu algoritmanın karmaşıklığı ise  $O(n^2 \cdot \log n)$ 'dir.

#### 3.1.2.2. Akış Şeması



Şekil 3. 3 İlk uygun yer yerleştirme algoritmasının akış şeması

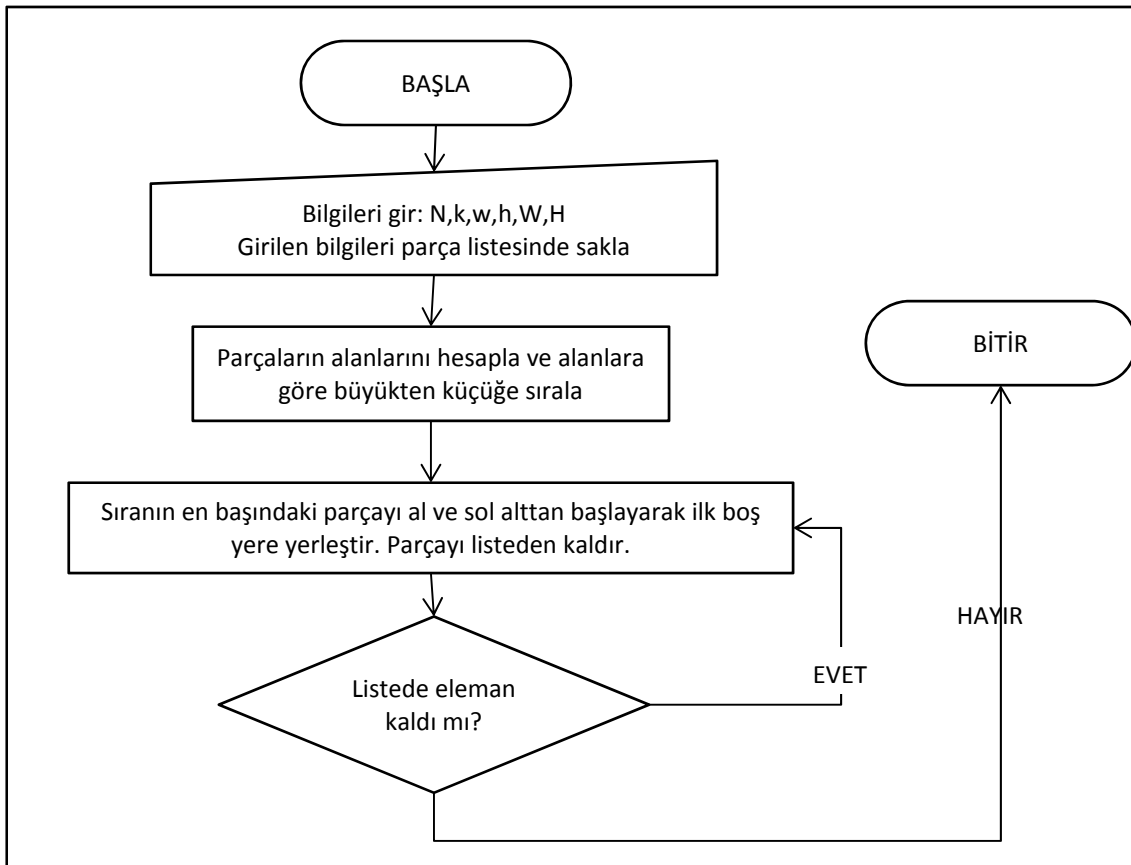
FF algoritması yerleştirme işlemi sırasında parçaları uygun sıraya soktukten sonra sol alt köşeden başlayarak yerleştirir. Burada sıralama değıştirme yoluyla bir sezgisel yöntem kullanılır. Yapılan çalışmalarda daha verimli sonuçlar elde edilmiştir.

### 3.1.3. Önerilen Alan Öncelikli Yerleştirme Algoritması (LAFF)

#### 3.1.3.1. Tanım

Diğer yerleştirme alditimlerinde olduğu gibi LAFF algoritmasında da yerleştirme işlemi sol alt kısımdan başlayarak yapılır. Burada parçaların sırası herbir parçanın sahip olduğu alana göre yapılır. Parçanın kapladığı alan büyük ise o parçaya öncelik verilir [30]. Algoritmanın karmaşıklığı FF ile benzer özellikler gösterir. Karmaşıklık  $O(n^2 \cdot \log n)$ 'dir.

#### 3.1.3.2. Akış Şeması



Şekil 3. 4 Alan öncelikli yerleştirme algoritmasının akış şeması

Bu algoritmada alan olarak büyük olan parçaya öncelik verilerek cam üzerine kesim planı oluşturulmaya yani yerleştirilmeye çalışılır. Bu sebeple ilk yapılacak iş, parçaları alanlarına göre sıralamaktır. Daha sonra yerleştirme işlemi BLF algoritmasında olduğu gibi yapılır.

### 3.1.4. Önerilen Sağ Alt – Sol Alt Dengelemeli Yerleştirme Algoritması (BLF-BRF)

#### 3.1.4.1. Tanım

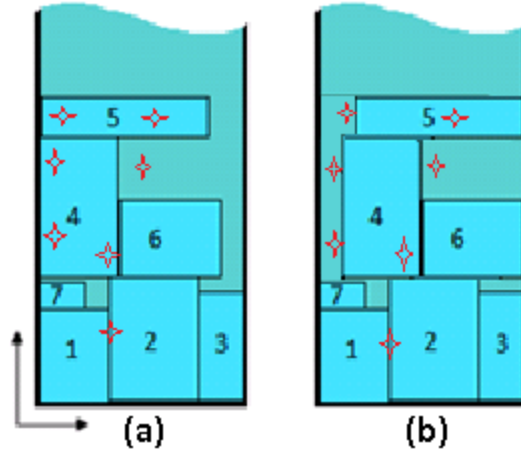
Kusur sayıları da dikkate alarak yerleştirme yaparken, sol alt algoritmasının sonucunda parçalar sol alta dayalı olarak yerleştirildiği için fireler genellikle sağ tarafta oluşur. Kusurların konum itibari ile sol tarafta daha yoğunlukla bulunuyorsa o zaman sol alt bu (BLF) yerleştirme sonucunda parçalara denk gelen kusur sayısı fazla çıkmaktadır. Ancak bu gibi durumlarda fire sağ taraftan değil de sol taraftan verilirse parçaya denk gelen kusur sayısı azalacaktır. Bu nedenle sağ alt – sol alt dengelemeli yerleştirme algoritması (blf-brf) önerilmiştir. Sol Alt yerleştirme (BLF) algoritmasına benzer biçimde çalışır. Bu yerleştirmede kusur sayılarına da dikkat edilerek yerleştirme yapılır. Çok amaçlı çözüm olarak önerilmiştir.

Çözüm planındaki her bir giyotin seviyesinde, sol alta dayalı mı yerleşecek, yoksa sağ alta dayalı mı yerleşecek, buna karar veren bir operatör eklenmiştir.

Diyelim ki n adet giyotin seviyesi oluşsun. Buna g<sub>n</sub> diyelim. Bu durumda g<sub>i</sub> yerleşim planı şu şekilde karar verilir.

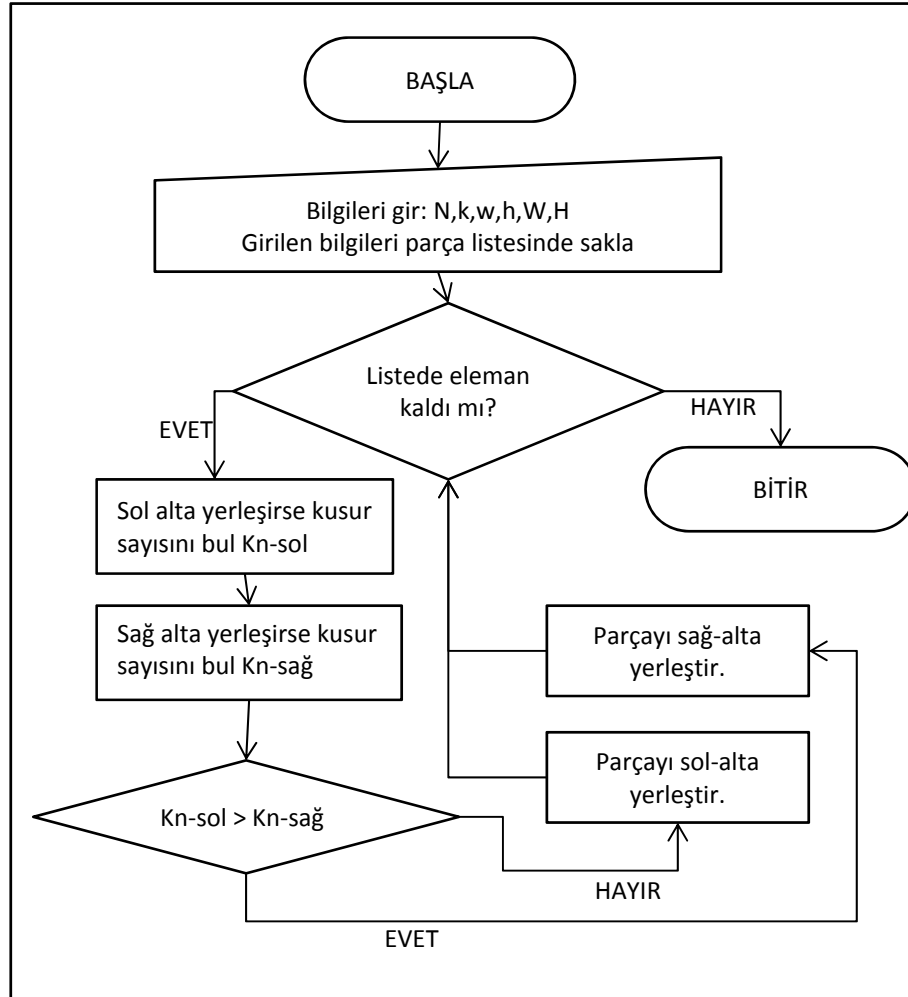
$$g_i = \begin{cases} \text{BLF yerleşimi uygula, } i. \text{ seviyede kusur yoğunluğu solda ise} \\ \text{BRF yerleşimi uygula, } i. \text{ seviyede kusur yoğunluğu sağda ise} \end{cases}_{i=1,2,\dots,n}$$

Sağ Alt(BRF) algoritması Sol alt (BLF) algoritması ile karakteristik olarak aynıdır. Tek farkı parçaları sola dayalı değil de sağa dayalı olarak yerleştirmesidir. Burada asıl farkı yaratan uygulama karar verme mekanizmasıdır.



Şekil 3. 5 Sağ Alt - Sol Alt Dengelemeli Yerleştirme algoritması örnek kesim planı

#### 3.1.4.2. Akış Şeması



Şekil 3. 6 Sağ Alt - Sol Alt Dengelemeli Yerleştirme algoritması akış şeması

Yukarıdaki Şekil 3. 5'e bakıldığında a'da bulunan çözümde kusurlar sol tarafta daha fazla bulunmaktadır. Parçalara denk gelen kusur sayısı 6'dır. b'de is 4, 5 ve 6 numaralı parçaların sağ tarafa kaydırılmasıyla 3 adet kusur artık parçalara denk gelmemektedir. Böylece kusur sayısında bir iyileşme görülmektedir.

### **3.2. Tek Amaçlı Evrimsel Yöntemler**

Evrimsel algoritmalar yerleştirme algoritmaları gibi tek seferde çözüm üretmezler. Bulduğu çözümü daha iyiye götürecek şekilde adım adım ilerler[31],. Belirli bir süre sonunda algoritma sonlanır. En son bulunan çözüm en iyi çözümdür.

#### **3.2.1. Genetik Algoritma(GA)**

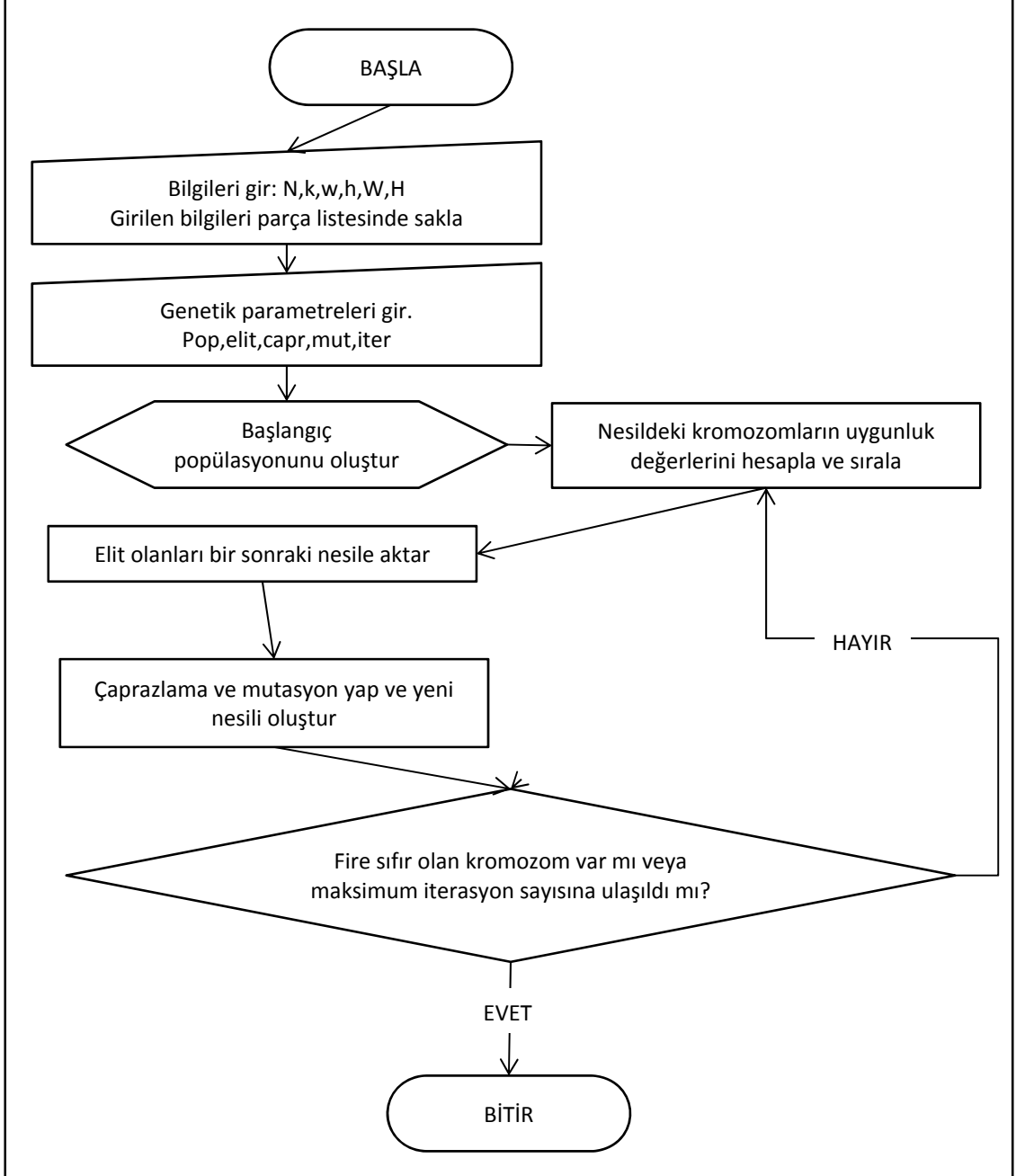
##### **3.2.1.1. Tanım**

Bu algoritma, ilk defa Holland [32] [33] [34]tarafından 1970'lerde bahsedilmiş, ardından De Jong [35] genetik operatörlerin analizi konusunda çalışmalar yapmıştır. Ardından Goldberg [36] tarafından genetik algoritmanın kural tabanlı makine öğrenmesi teknikleri üzerinde çalıştığı gösterilmiştir. Daha sonra yine Goldberg [37] genetik algoritmanın optimizasyon problemleri üzerinde kullanılmasını temel alan bir kitap yazarak bilime çok önemli bir katkıda bulunmuş ve sonraki bilim adamlarının önüne ışık tutmuştur. Ono ve Watanabe [38] genetik algoritmayı kesme problemleri üzerinde çalıştırmışlardır. Ono ve Watanabe [38] iki tür kesme problemi ile ilgilenmiştir. Bunlardan birincisi boruların kesilmesi, ikincisi ise iki boyutlu kesme probleminde kesilecek parçalar için kullanılan büyük parçanın en az şekilde kullanılmasıydı.

Genetik algoritma, canlıların kalıtım ile çeşitlenmesi fikrinden yola çıkarak oluşturulmuştur. Bu çeşitlenme işlemiiki bireyin kromozomlarının çaprazlanması ile oluşur. Cam kesme probleminde kromozomlar birer çözümü ifade eder. Bu kromozomlar çaprazlanarak yeni birey oluşur, bu birey bazen mutasyona uğrar ve bambaşka bir birey haline gelir. Her bir bireyin kalitesi o bireyin bir sonraki nesilde yaşamasına etki eder. Kaliteli bireyler yaşar, zayıflar ölür. Kaliteli bireylerin çaprazlanması sonucu daha kaliteli bireyler ortaya çıkabilir fikri, genetik algoritmanın ana temasıdır.

Algoritmanın akış şeması sonraki kısımda anlatılmaktadır. Ayrıca genetik algoritmanın çalışması için bir takım parametrelere ihtiyaç vardır. Bu parametreler de bir sonraki kısımda anlatılmaktadır.

### 3.2.1.2. Akış Şeması



Şekil 3. 7 Genetik algoritma akış şeması

Algoritmanın akışı şeması Şekil 3. 7’de verilmiştir. Algoritmanın çalışması için başlangıç popülasyonuna ihtiyacı vardır. Bu başlangıç popülasyonu bir sonraki neslin

oluşturulması için temel teşkil eder. Bir sonraki nesile bazı kaliteli bireyler hiçbir işleme uğramadan direk geçerler. Bazıları ise çaprazlama ve mutasyona uğrayarak bir sonraki nesile yeni birer bireyler kazandırmaya çalışırlar. Bu yeni bireylerden iyi olanlar seçilir, kötü olanlar yok edilir. Bu işlem nesiller boyu devam eder ve belirli bir zaman sonra popülasyonda sadece iyi bireyler kalmış olur.

### **3.2.1.3. Gen**

Genetik algoritmada her bir değişken gen olarak adlandırılır. Genler bilinmeyenlerdir. Genetik algoritmada genlere rastgele değerler atanır ve çözüme olan etkisi incelenir. Genlerin değeri bir kez atandıktan sonra bu değerler belirli kriterlere göre değiştirilerek sonuç iyileştirilmeye çalışılır.

Genler dört biçimde gösterilebilir. Bunlar; ikili (binary), reel, sıralama ya da permütasyon, ağaç (tree) şeklindedir.

**Genlerin İkili kodlanması:** Burada her bir genin 0 yada 1 değerini alır.

**Genlerin reel kodlanması:** Burada her bir genin 0,1 arasında rastgele ondalıklı bir değer alır veya değişik bir yöntem olarak alt limit, üst limit arasında ondalıklı ya da tam sayılı bir değer alır.

**Genlerin permütasyon kodlanması:** Burada her bir genin kendisine ait özel bir numara alır. bu numara o geni diğerlerinden ayır eden bir anahtar görevi görür.

**Genlerin ağaç kodlanması:** Burada her bir genin bir düğüm olarak gösterilir.

### **3.2.1.4. Kromozom**

Problemin her bir tekil çözümü bir bireyi temsil eder. Buna kromozom da denir. Genetik algoritmada kromozomların en iyisi çözüm olarak kabul edilir. Kromozomlar genlerden oluşur. Genlerin kodlama tipine göre kromozomların da kodlama tipi belirlenir. Kromozom genlerin bir araya gelmesiyle oluşturulur. Problemin karmaşıklığına göre genlerin sayısı değişir.

### İkili Kodlama

Burada herbir gen 0 veya 1 değerinden birisini alır. Genler bir araya gelerek bir kromozomu oluşturur. Burada kullanılacak genlerin sayısı  $n$  ve problemdeki bağımsız değişken sayısı  $x$  olmak üzere,

$$n = \log_2 x \quad (3.1)$$

şeklinde belirlenir. Örneğin 4 değişkenli bir problemi kodlamak için 2 adet gen yeterlidir. 00,01,10,11 şeklinde kodlanara 4 değişkeni de içerebilir. 24 genden oluşan bir kromozom yapısı Şekil 3. 8’da gösterilmiştir.

|            |                          |
|------------|--------------------------|
| Kromozom A | 110011010111010001001110 |
| Kromozom B | 101011110000010100110011 |

Şekil 3. 8 İkili kodlanmış kromozom

Bu kodlama şekli genellikle atama, sipariş önceliklendirme ya da sırt çantası problemi gibi problemlerde kullanılır. Sırt çantası problemi ele alınacak olursa, sırt çantasına koyulabilecek eşyaların her birinin bir değeri var ve sırt çantasının da kapasitesi var. Tüm eşyalar çantaya sığmadığında hangi ürünün çantaya koyulması gerektiği eşyaların değerine göre belirleniyor. Bu durumda eğer  $i$ . eşya çantaya koyulacaksa 1 koyulmayacaksa 0 değerini alıyor. Tüm eşyalar için düşünüldüğünde bir kromozom yapısı oluşuyor. Şekil 3. 8’da A koromozomu eşya 1 ve 2 çantaya koyulacağını, 3 ve 4 çantaya koyulmayacağını ifade etmektedir. Diğerleri de benzer şekilde yorumlanır.

### Reel Kodlama

Burada genler rastgele reel değerler alırlar. 0,1 aralığında veya alt-üst limit çerçevesinde belirlenebilir. Bu değerler biraraya gelerek reel değerli kromozomları oluşturular. Aşağıdaki Şekil 3. 9’da 5 gene sahip reel değerli bir kromozom görülmektedir.

|            |        |        |        |        |        |
|------------|--------|--------|--------|--------|--------|
| Kromozom A | 0,8965 | 0,5596 | 0,2145 | 0,3214 | 0,6545 |
| Kromozom B | Kuzey  | Güney  | Güney  | Doğu   | Batı   |

Şekil 3. 9 Reel kodlanmış kromozom

Bu kromozom yapısında genler sayı değerleri alabildiği gibi nominal değerler de alabilir. Şekil 3. 9’de B kromozomu bir robotun hedefe ulaşmak için kullandığı ilerleme yönlerini



belirtsin. Bu durumda robotun önce kuzeye sonra 2 kez güneye sonra doğuya sonra batıya gideceğini ifade etmektedir.

### Permütasyon Kodlama

Permütasyon kodlamada genleri ifade eden birer numara yada harf kullanılabilir. Burada kromozomu oluşturan genler bir sıra ifade ederler.

|            |   |   |   |   |   |
|------------|---|---|---|---|---|
| Kromozom A | 5 | 2 | 4 | 3 | 1 |
| Kromozom B | A | C | D | B | E |

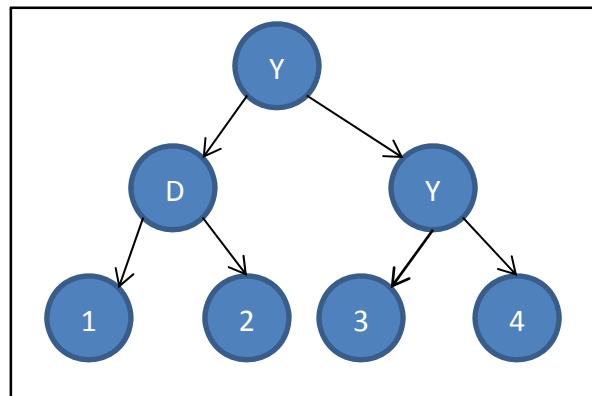
Şekil 3. 10 Permütasyon kodlanmış kromozom

Cam kesme probleminde olduğu gibi sıranın önemli olduğu problemlerde kullanılır. Şekil 3. 10'da A kromozomunda ifade edilen 5 adet cam parçası siparişinin kesilme sırası belirtilmiştir. A kromozomundaki örneğe göre önce 5 nolu parçanın sonra 2 nolu parça ardından 4,3,1 nolu parça sırayla cam bloğundan kesilecektir. B kromozomu için de gezgin satıcı problemi örnek verilebilir. Gezgin satıcı probleminde 5 adet şehire hangi sırayla gidilirse en az yol katedileceğini hesaplar. Bu örnekte önce A şehrine sonra oradan C şehrine ardından D, B ve E şehirlerine sırayla gidileceğini ifade eder.

Bu tez çalışmasında permütasyon kodlama yapısı kullanılmıştır.

### Ağaç Kodlama

Bu yapıda hiyerarşik bir düzende kodlama yapılır. Bu kodlamada düğümler genleri ifade eder.



Şekil 3. 11 Ağaç kodlanmış kromozom

Cam kesme problemi ağaç kodlaması ile gösterilmek istenirse Şekil 3. 11'deki gibi bir yapıda olabilir. Burada 4 tane parça kesilmek isteniyor. 1 ve 2 numaralı parça birbirine dikey olarak birleştirileceğini ifade ediyor. 3 ve 4 numaralı parça kendi aralarında birbirine dikey olarak birleşeceğini ifade ediyor. Sonra bu iki grup parça yani 1,2 ile 3,4 grubu parçalar birbirine yatay olarak birleştirilmesi gerekmektedir.

#### **3.2.1.5. Popülasyon**

Kromozomlar problem için çözüm kümesini ifade etmektedir. Genetik algoritma, birden fazla kromozom üzerinde çalışarak aralarından en iyilerini seçen bir evrimsel algoritmadır. Genetik algoritmanın her bir iterasyonunda belirli sayıda kromozom üzerinde çalışılır. İşte bu kromozomların bir arada bulunduğu topluluğa popülasyon denir. Popülasyondaki bireyler üzerine çaprazlama ve mutasyon gibi genetik operatörler kullanılarak yeni bireyler oluşturulur. Bu yeni bireyler bir sonraki popülasyonu oluşturmak için kullanılır.

Popülasyonda kaç adet bireyin bulunacağı algoritmanın performansını etkiler. Bu sayı algoritma başlamadan önce belirlenmesi gerekir. Bu değere popülasyon sayısı denir. Popülasyon sayısı çok büyük seçildiğinde (ör:100) algoritma daha yavaş çalışır. Küçük seçildiğinde (ör:5) algoritma hızlanır fakat bireylerin çeşitliliği azalır ve çözümler çok farklılaşmaz. Dolayısıyla yerel optimum noktasına takılma olasılığı artar. Bu durumda ortalama bir değer seçmek (ör:25) hem algoritmanın hızını çok etkilemez hem de bireysel çeşitliliği sağlar.

#### **Başlangıç Popülasyonu**

Algoritmanın başlaması için popülasyon sayısı kadar bireyin oluşturulması gerekir. Genetik algortmada herşey rastgele oluşturulur. Dolayısıyla bu ilk popülasyon da rastgele oluşturulur ve çözümün her iterasyonda biraz daha iyileşmesi beklenir. Ancak başlangıç popülasyonu stratejisi uygulayarak algoritmaya biraz daha hız vermek mümkündür. Bilinen bazı sezgisel algoritmaların çözümleri başlangıç popülasyonuna eklenerek bu sağlanabilir.

Bu tez çalışmasında BLF algoritmasının, FF algoritmasının ve LAFF algoritmasının sonuçları başlangıç popülasyonuna 3 birey olarak eklenmiştir. Diğer bireyler rastgele oluşturulmuştur. Böylece bu üç algoritmanın ürettiği sonuçlardan her zaman eşit yada daha iyi bir çözüm bulmak garanti altına alındı.

#### **3.2.1.6. Uygunluk Fonksiyonu**

Popülasyondaki her birey kalitesine göre değer alır. Bu değere uygunluk değeri denir ve çözüm kümesinin değerini verir. Uygunluk değeri bir fonksiyon ile elde edilir. Bu fonksiyona da uygunluk fonksiyonu denir.

NP-Tam karmaşıklığına sahip problemlerde optimum çözüm bilinmediği için bu fonksiyonu belirlemek de oldukça güçtür. Bu nedenle bu fonksiyon genellikle varsayımlar ile oluşturulur.

Cam kesme probleminde amaç fireyi minimize etmek olduğu için fire değerini veren fonksiyon uygunluk fonksiyonu olarak belirlenmiştir. Uygunluk fonksiyonu aşağıdaki şekilde hesaplanarak uygunluk değeri bulunur.

$$f = \frac{\sum_i w_i * h_i}{W * H'} \quad (3.2)$$

Denklem 3.2'ye göre kesim planına yerleştirilen parçaların toplam alanının cam bloğundaki kullanılan alana bölümü ile elde edilir. Cam kesme probleminde genişlik (W) sabit fakat yükseklik sınırsız olarak çözüm elde edilmiştir. Bu nedenle kullanılan alan H yerine  $H'$  olarak belirtilmiştir.

#### **3.2.1.7. Elitizm**

Popülasyondaki bazı kaliteli bireyleri kaybetmemek için bir sonraki popülasyona hiçbir işleme maruz kalmadan direk olarak gönderilmesi gerekir. Bu iyi çözümün elde tutulması gerektiği için yapılır. Çünkü bireyler çaprazlama ve mutasyona uğradıklarında daha iyi bir sonuç vermeyi garanti etmezler. Tüm işlemlerden sonra hep daha kötü sonuçlar çıkar ise bu kez de bir önceki popülasyonda bulunan iyi çözümü kaybetmemek gerekir.

Elitizmde kaç adet bireyin bir sonraki popölasyona geçmesi gerektiğini belirten değere elitizm değeri denir. Bu değeri yüzde olarak da verilebilir veya sayı olarak da verilebilir.

#### **3.2.1.8. Kromozom seçimi**

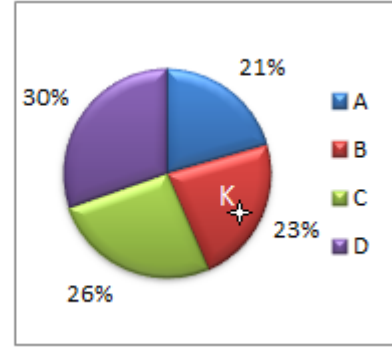
Genetik algoritmanın operatörleri olan çaprazlama ve mutasyon yapmak için popölasyondan bireylerin seçilmesi gerekir. Seçim işlemi GA için çok önemlidir. Çünkü bu seçim işlemi sonraki aşamalarda elde edilecek bireyin daha kaliteli olmasında etkili olacaktır. Seçim işlemi sırasında kaliteli bireyler arasından seçilen bireyler sayesinde daha kaliteli bireyler elde edilebilir varsayımı sebebiyle uygunluk değeri yüksek olan bireylerden seçmeye çalışılır. Ancak sadece bu bireyleri kullanmak yerel optimuma takılmanın olasılığını artırır. Bu nedenle kaliteli bireylerden seçmek yerine o bireylerden seçilme olasılığını arttırarak ve diğer bireylere de seçilme şansı verecek bir algoritma geliştirmek gereklidir.

Baker [39] doğrusal aşamalı bir seçim algoritması geliştirmiştir. Daha sonra bu algoritmanın etkinliğini arttıracak çalışma yapmıştır [40]. Alam vd. [41], yinelemeli iki aşamalı seçim algoritmasını geliştirmişlerdir. Onal ve Beldek [42] ise bu yinelemeli iki aşamalı seçim algoritmasını daha etkin hale getirmişlerdir. Ayrıca Miller ve Goldberg [43] turnuva seçim algoritmasının etkinliğini göstermişlerdir. Başka bir seçim algoritması da rulet tekeri algoritmasıdır [44]. Genetik algoritmalarda en sık kullanılan yöntem budur. Uygulamadaki kolaylığı ve etkinliği sayesinde tercih edilmektedir. Bu tez çalışmasında da rulet tekeri seçim algoritmasından yararlanılmıştır.

#### **Rulet Tekerı Seçim Algoritması**

Bu algoritmada uygunluk değeri yüksek olan bireylerin seçilme şansı daha yüksektir. Bireyler uygunluk değerine göre bir rulet tekeri üzerine yerleştirilirler. Uygunluk değeri iyi olan teker üzerinde daha fazla alan kaplayacak şekilde normalize edilir. Daha sonra bu teker üzerinden rastgele bir nokta seçilir ve seçilen noktaya karşılık gelen birey seçilmiş olur.

| Kromozomlar | Uygunluk Değeri | Normalize Değer |
|-------------|-----------------|-----------------|
| A           | 250             | 21%             |
| B           | 280             | 23%             |
| C           | 320             | 27%             |
| D           | 370             | 31%             |



Şekil 3. 12 Rulet tekeri algoritmasında 4 kromozomlu bir örnek

Şekil 3. 12' de 4 tane kromozomun uygunluk değerleri verilmiştir ve bu örnek maksimizasyon problemi olarak ele alınmıştır. Öncelikle bu değerleri normalize ederek yüzdelik dilimleri tespit edilir. Bu yüzdelik dilimlere göre rulet tekeri üzerine yerleştirilir. Eğer bu bir maksimizasyon problemi ise o zaman bu normalize değerler aynen kullanılır. Eğer bu bir minimizasyon problemi ise o zaman uygunluk değeri en çok olan kromozom, teker üzerinde daha fazla alan kaplaması gerekiyor. Bu nedenle normalize değerleri ters çevirerek yani kromozomların normalize değeri sırasıyla A:31, B:27, C:23 ve D:21 olarak rulet tekeri üzerine yerleştirilir. Daha sonra rulet tekeri üzerine herhangi bir nokta rastgele seçilir. Diyelim ki K noktası seçilmiş olsun. O zaman K değerine karşılık gelen kromozom B kromozomudur. B kromozomu seçilmiş eleman olarak kullanılır.

### 3.2.1.9. Çaprazlama

Genetik algorithmada yeni popülasyonu oluşturmak ve yeni bir çözüm elde etmek için kullanılan iki operatörden birisi çaprazlama diğeri ise mutasyondur. Çaprazlama yapmak için öncelikle popülasyondan iki adet kromozom seçilir. Bu kromozomlar üzerine çarpazlama işlemi uygulanır.

Çaprazlama her seçilen kromozoma uygulanmaz. Bazen seçilen kromozomları çaprazlanmadan işleme devam edilir. Bu sayede mevcut popülasyonun çok farklılaşmaması sağlanır. Bu çaprazlamaya girmeyen kromozomlar murasyon operatörüne tabi tutulmak üzere bir sonraki aşamaya geçirilir. Bir kromozom çiftinin çaprazlamaya uğrayıp uymaması çaprazlama olasılığı denen bir değere bağlıdır. Bu değer 0-100 aralığında seçilmiş bir değerdir. 0 değeri çiftin çaprazlamaya kesin olarak

uğramayacağını, 1 değeri ise mutlaka çaprazlama yapılması gerektiğini ifade eder. Çaprazlama, genetik algoritmanın yeni bireyler üretmek için kullandığı önemli bir operatördür. Bu nedenle çaprazlama olasılığını düşük yapmak genetik algoritmanın performansını düşürür. Çünkü seçilen çiftlerin büyük çoğunluğu çaprazlamaya uğramaz ise popülasyona yeni bireyler katılamaz ve aynı popülasyonla bir sonraki iterasyona geçilmek zorunda kalınır.

Çaprazlama yapmak için bazı yöntemler var. Tek noktalı çaprazlama, kromozom üzerinde bir nokta belirlenir. O noktanın solunda kalanlar 1.kromozomdan, sağında kalanlar 2.kromozomdan alınarak yeni bir kromozom oluşturulur. Aynı şekilde bu kez aynı noktanın solunda kalanlar 2.kromozomdan, sağında kalanlar da 1.kromozomdan alınarak yeni bir kromozom daha oluşturulur. Böylece 2 ebeveyn kromozomdan 2 adet yavru kromozom oluşturulmuş olur. Uniform Çaprazlama tekniğinde ise benzer durum söz konusudur. Burada çaprazlama için tek bir nokta seçilmez birden fazla nokta seçilir. Her noktanın bulunduğu yere kadar 1.kromozomdan başlayarak sırasıyla ebeveynlerden genler alınır.

### İkili Kodlamada Çaprazlama

| Ebeveyn    | Dizilim             |   | Yavru       | Dizilim             |
|------------|---------------------|---|-------------|---------------------|
| Kromozom A | 1 1 1 0 0 0 1 1 0 1 |   | Kromozom A' | 1 1 1 0 1 1 0 0 0 1 |
| Kromozom B | 1 0 1 0 1 1 0 0 0 1 |   | Kromozom B' | 1 0 1 0 0 0 1 1 0 1 |
|            |                     | K |             |                     |

Şekil 3. 13 İkili kodlamada tek noktalı çaprazlama

Tek noktalı çaprazlamada Şekil 3. 13'deki gibi bir K noktası seçilir. Kromozom A' ismi verilmiş olan yavru kromozomu oluşturmak için K noktasının solunda kalan 1110 değeri A kromozomundan alınır. K noktasının sağında kalan kısım 110001 değeri de b kromozomundan alınır. Böylece ilk yavru birey oluşturulmuş olur. Benzer şekilde B' kromozomu da önce B kromozomunda 1010 değeri alınır, sonra A kromozomundan 001101 değeri alınır. Böylece ikinci yavru birey de oluşturulmuş olur.

| Ebeveyn    | Dizilim             |          | Yavru       | Dizilim             |
|------------|---------------------|----------|-------------|---------------------|
| Kromozom A | 1 1 1 0 0 0 1 1 0 1 |          | Kromozom A' | 1 1 1 0 0 0 1 0 0 1 |
| Kromozom B | 1 0 1 0 1 1 0 0 0 1 |          | Kromozom B' | 1 0 1 0 1 1 0 1 0 1 |
|            |                     | K1 K2 K3 |             |                     |

### Şekil 3. 14 İkili kodlamada çok noktalı(uniform) çaprazlama

Çok noktalı çaprazlamada birden fazla  $K_i$  noktası seçilir. Kaç noktadan çaprazlama yapılacağı yine rastgele olarak seçilir. Şekil 3. 14’de görüldüğü gibi her  $K$  değerine gelindiğinde parçası alınan kromozom değiştirilir. Bütün genler dolana kadar bu işleme devam edilir.

### Permütasyon Kodlamada Çaprazlama

| Ebeveyn    | Dizilim             | Yavru       | Dizilim             |
|------------|---------------------|-------------|---------------------|
| Kromozom A | 1 5 4 3 2 6 9 8 7 0 | Kromozom A' | 1 5 4 3 8 2 6 7 0 9 |
| Kromozom B | 8 2 5 4 6 7 3 1 0 9 | Kromozom B' | 8 2 5 4 1 3 6 9 2 0 |

K

### Şekil 3. 15 Permütasyon kodlamada tek noktalı çaprazlama

Burada bir  $K$  noktası seçilir ve kromozomlar o  $K$  noktasının ayırdığı bölge üzerinden yapılır. İkili kodlamada olduğu gibi direk sağdakiler ve soldakiler şeklinde bir çaprazlama söz konusu olamaz. Çünkü eğer A kromozomundan soldakiler, B kromozomundan sağdakiler alınmış olsaydı o zaman numaralar tekrar etmiş olacaktı. Hâlbuki permütasyon kodlamada her numaranın sadece bir kez bulunması gerekiyor. Bu nedenle permütasyon çaprazlamada farklı bir yöntem uygulanmıştır. Öncelikle  $K$  noktasının solunda kalan kısım A kromozomundan aynen alınır. Sağında kalan kısım ise B kromozomundan alınmadan önce, A kromozomundan alınan genler B kromozomunda işaretlenir. İşaretlenmemiş genler soldan başlayarak yeni bireye eklenir. Böylece yeni bireyin bütün genleri birbirinden farklı bir yapıya kavuşur. İkinci yeni bireyi oluşturmak için de birkaç yöntem mevcuttur. Birinci yöntem;  $K$  noktasına kadar B kromozomundan alınır. Alınan genler A kromozomunda işaretlenir ve işaretsiz olan genler sırayla yeni bireye aktarılır. İkinci yöntem; ilk bireyi oluşturduktan sonra ikinci birey ilkinin tam tersi alınarak yapılır. Yani ilk kromozom xyz şeklinde ise ikinci kromozom zyx şekline kabul edilir.

#### 3.2.1.10. Mutasyon

Bir sonraki popülasyona aktarılmak üzere hazırlanan kromozomlar bazen rastgele olarak şekil değiştirirler. Bu düzensiz şekil değiştirmeye mutasyon denir. Mutasyon, algoritmanın yerel optimum noktasına takılmasını engeller. Çaprazlama sonucu elde

edilen bireyler belirli bir olasılık değeriyle mutasyona uğrarlar. Bu olasılık değerine mutasyon olasılığı denir. Mutasyon olasılığı küçük bir değerdir. Böylelikle her birey mutasyona uğratılmaz.

Çaprazlama işlemine maruz kalan bireyler bazen aynı çözüm üzerinde hareketsiz kalabilirler. Mutasyon sonucu tamamen farklı bir çözümün popülasyona girmesi sağlanır. Böylece popülasyondaki çeşitlilik artar ve daha iyi çözümler bulma ihtimali artar.

### İkili Kodlamada Mutasyon

|                     |   |   |   |   |   |   |   |   |   |   |
|---------------------|---|---|---|---|---|---|---|---|---|---|
| <b>Kromozom A'</b>  | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| <b>Kromozom A''</b> | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

Şekil 3. 16 İkili kodlamada mutasyon

Rastgele seçilen bir genin direkt olarak değeri değiştirilir. Şekil 3. 16'de A' kromozomundaki dördüncü gen seçilmiş, 0 olan değeri 1 olarak değiştirilmiş ve A'' kromozomu oluşmuştur. Bir sonraki popülasyona bu mutasyona uğramış olan A'' kromozomu taşınmaktadır.

### Permütasyon Kodlamada Mutasyon

|                    |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---|---|---|---|---|---|---|---|---|
| <b>Kromozom A'</b> | 1 | 5 | 4 | 3 | 8 | 2 | 6 | 7 | 0 | 9 |
| <b>Kromozom A'</b> | 1 | 5 | 4 | 7 | 8 | 2 | 6 | 3 | 0 | 9 |

Şekil 3. 17 Permütasyon kodlamada mutasyon

Burada yine ikili kodlamadan farklı olarak; herhangi bir gen direkt olarak değeri değiştirilmek suretiyle mutasyon yapılamamaktadır. Çünkü değeri değiştirilen genin aynı kromozomdaki diğer genlerle çakışma ihtimali doğmaktadır. Bu nedenle permütasyon kodlama gen yerine gen çifti seçilir. Ve seçilen çiftler birbiri ile yer değiştirir. Bu sayede çakışma önlenmiş olur. Şekil 3. 17'deki örnekte dördüncü gen mutasyona uğratılmış. Bunun için de sekizinci gen de onun çifti olarak seçilmiş ve bu iki gen yer değiştirmiştir.



### 3.2.2. Yapay Arı Algoritması(ABC)

#### 3.2.2.1. Gerçek Arıların Davranışları

Gerçek bir arı kolonisinde yapılacak işler için arılar arasında bir iş bölümü vardır. Bu iş bölümünü yaparken herhangi bir merkezi otorite olmadığından tamamen kendiliğinden bir bölümlenme olmaktadır. Teresko [45]'nin yaptığı araştırmalar sonucunda önerdiği minimum yiyecek arama modelinin üç temel bileşeni vardır. Bunlar yiyecek kaynakları, görevi belirli işçiler ve görevi belirsiz işçiler. Bu modele göre yapılacak iki eylem vardır. Birincisi yiyecek kaynağını tespit etme, ikicisi ise belirli bir kaynaktan vazgeçmedir.



Şekil 3. 18 Kaynaktan nektar toplamak ile görevli işçi arı

Yiyecek kaynakları, arılar için bal üretmek için yada polen vs taşımak için bulduğu çiçek gibi besin kaynaklarıdır. Bir yiyecek kaynağının değeri birçok faktöre bağlı olarak düşünülebilir. Mesela kaynaktaki besin miktarı, yuvaya uzaklığı, besinin elde edilmesindeki kolaylık gibi...

Görevi belirli işçi arılar belirlenen kaynaktaki besinleri kovana getirmekle sorumludur. Aynı zamanda kaynağın değeri ile ilgili bilgileri de kovanda bekleyen diğer arılarla paylaşırlar. Bu sayede diğer işçiler de kaynağa doğru giderler veya kaynaktan vazgeçerler. Görevi belirsiz arılar iki çeşittir. Yeni kaynaklar bulmak için gezinen kaşif arılar ve kaşif arıların getirdiği kaynak hakkındaki bilgiyi alıp gerekirse yeni kaynağa doğru yönelen gözcü arılar. Kaşif arılar koloninin yaklaşık %5'ini oluştururlar.

Kovanda bilgi paylaşımı dans alanında yapılır. Bu dans alanında kaynaktan gelen arı dans ederken kovanda bekleyen arılar o arıya dokunarak kaynak hakkında koku tat gibi değerleri kullanarak kaynağın değeri ile ilgili bilgileri alırlar. Kovanda değişik alanlarda dans ederek kaynak hakkında daha fazla arının haberdar olmasını sağlarlar. Taşınan

bilgiye göre dans deđiřtiđi iin sadece koku ve tat ile deđil, dansın hareketlerine gre de kaynak bilgisi deđerlendirilir. Daire dansı kaynađın 50-100 metre apında bir alanda ulduđunu belirtir. Kuyruk dansı 8 biiminde yapılan danstır. Bu dans kaynađın 100 metre ila 10 km arasında bir alanda bulunduđunu syler. Titreme dansı, arıların dzensiz ve yavař olarak bacaklarını ileri geri hareket ettirmesiyle yapılır. Bu dansa kaynađın ok fazla nektar tařıdıđını ve hemen bal yapma ařaması gemek istediđini belirtir[46].

Kovandaki arılar kaynađın yn bilgisini aldıktan sonra gneřden yararlanarak kaynađı bulurlar. Arıları gzleri gneřin ynnn kapalı havalarda dahi algılayacak biimde yaratılmıřtır. Yapılan dansın tekrarlanma sıklıđı kaynađın uzaklıđını belirtir. Az tekrarlanma kaynađın uzak olduđunu gsterir. Aynı zamanda yapılan hareketler kaynakla gneř arasında 45 derecelik bir aı oluřturarak yapıldı iin yn hakkında da bilgi vermiř olurlar.

Arılar kaynađa ulařırken kaynađın uzaklıđına ve tařıdıkları yke gre ykseklik ayarı yaparak kendi enerjilerini verimli bir řekilde kullanmaktadırlar.

#### **3.2.2.2. Tanım**

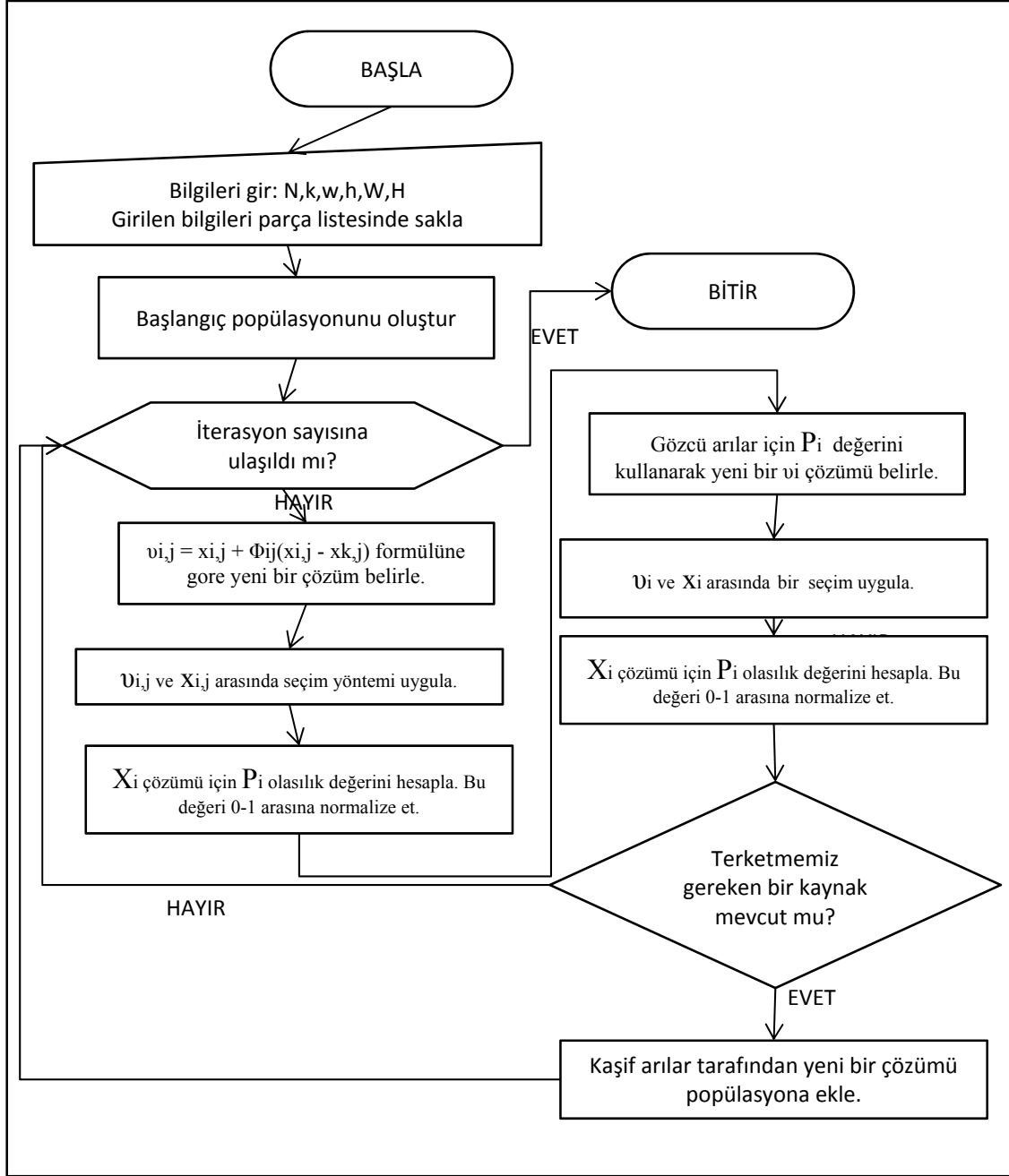
Dođada bu řekilde hareket eden arıların davranıřlarından yola ıkarak Karabođa ve Akay [22], yapay arı algoritmasını geliřtirmiřtir. Bu algoritma ile nmerik fonksiyonların optimizasyonu zerinde kaydadeđer sonular elde etmiřtir.

Bu algoritmaya gre zm poplasyonu oluřturulduktan sonra iyi kaynakların seimi iin arıların hareketlerinden faydalanılarak bir seim algoritması oluřturulmuřtur.

ABC algoritması sr zekasına dayalı bir algoritmadır. Gerek arıların yiyecek arama dngs iyi bir řekilde simle edilmiřtir. Kařif arılar sayesinde yerel optimum noktasına takılmadan global optimum zmler elde edilebilmektedir. Bu algoritma nmerik problemler iin geliřtirilmiřtir. Ancak bu tez alıřmasında ayırık problemler iin de kullanılabildiđi gsterilmiřtir.

### 3.2.2.3. Akış Şeması

Yapay arı algoritmasında popülasyon oluşturulduktan sonra her çözüm komşu çözümlerden birisiyle karşılaştırılır. Ardından yeni çözümün kalitesine bakılarak mevcut çözüm terkedilir veya elde tutulur. Eğer bir kaynak terkedilmiş ise kaşif arılar tamamen yeni bir çözümü popülasyona eklerler. Bir sonraki iterasyon da aynı şekilde devam eder. Sonuçta en iyi çözümler bir yerde saklanır.



Şekil 3. 19 Yapay arı algoritması akış şeması

#### 3.2.2.4. Başlangıç Durumu

Yapay arı algoritmasında başlangıç popülasyonu rastgele oluşturulur. Bu popülasyondaki çözümlerin her birinin değeri hesaplandıktan sonra bir de ek olarak geliştirilememe değeri 0 (sıfır) olarak tanımlanır. Bu değer kaynağı geliştirmek için yapılan her işlemde eğer geliştirme yapılamadıysa artacak şekilde kullanılır. Belirli bir eşiği geçen geliştirilememe değerine sahip çözümler için tekrar geliştirilmeye çalışılmaz. Artık bu çözüm için kaşif arı göreve başlar ve daha iyi bir çözüm aramaya başlar.

#### 3.2.2.5. İşçi Arıların Kaynağı Geliştirmesi

İşçi arıların kaynağı daha iyi hale getirmesi için kaynağın komşuluğundaki değerleri de gezmesi gerekiyor. Kaynağa çok yakın başka kaynakların varlığında eğer daha iyi bir kaynak mevcutsa o zaman mevcut kaynak bırakılıp bir sonraki turda yeni kaynağa hareket edilir. Aynı zamanda bu yeni kaynağın geliştirilememe değeri sıfırlanır. Eğer komşuluğunda daha iyi bir kaynak bulunamadıysa geliştirilememe değeri artar ve eşik değerine ulaşması beklenir. Eşik değerine ulaşılan kaynağa artık işçi arı gönderilmez. Onun yerine bir kaşif arı görevlendirilerek o kaynağın daha iyisinin bulunması beklenir.

#### 3.2.2.6. Gözcü Arıların Kaynak Seçimi Yapması

Her iterasyonda hangi kaynaklara gidileceğine gözcü arılar karar verir. Bu arılar dans alanına gelen işçi arılardan aldıkları bilgilerle ve kaşif arıların yeni buldukları kaynaklarla ilgili bilgileri değerlendirerek bir seçim uygular. Bu çalışmada rulet tekerleği seçim algoritması uygulanmıştır.

Rulet tekerleğinde her bir kaynağın değeri tüm kaynakların toplam değerine oranlanan kaynak seçilme olasılığı olan  $P_i$  değeri belirlenir.

$$P_i = \frac{uygunluk_i}{\sum_i uygunluk_i} \quad (3.3)$$

Bu denklemde  $uygunluk_i$  değeri i. kaynağın değerini belirtir. Burada görüleceği gibi bir kaynağın uygunluk değeri arttıkça o kaynağı seçecek gözcü arıların sayısı da artacaktır. Bu özelliğe pozitif geribesleme özelliği denir.

Burada seçim yapan gözcü arılar işçi arılar gibi kaynağa gidip kaynak hakkında ve komşuları hakkında bilgi sahibi olurlar. Kovanda kalan işçi arılar da artık birer gözcü arı olurlar ve yeni gelen bilgiler ışığında hangi kaynağa gideceklerini seçerler.

#### **3.2.2.7. Bir Kaynağın Terkedilmesi**

Bir kaynağın terkedilmesi o kaynaktan artık faydalanamamaya göre karar verilir. Bu algorithmada bir kaynağa gidildiğinde komşuluğunda daha iyi bir çözüm bulunamaması geliştirilememe değerini arttırmaktadır. Bu değer belirli bir eşiği geçtikten sonra artık oraya işçi arı gönderilmez. Bu eşik değerine limit denir. Orada görevli işçi arı da artık kaşif görevi görür ve rastgele yeni kaynak aramaya çıkar. Algorithmada her iterasyonda sadece bir tane işçi arı kaşif arı olarak görevlendirilebilir.

#### **3.2.2.8. Global Ve Yerel Optimum Noktaları**

ABC algoritması kullandığı seçim yöntemleri ile hem yerel optimum noktalarını bulmaya çalışır. İşçi arıların ve gözcü arıların kaynağın komşuluklarını araştırmasında yerel optimum noktalarına, terkedilen kaynakların kaşif arılar tarafından yeni kaynaklar araştırılmasında da global optimum noktalarına erişim sağlanır. Bu yönüyle ABC algoritması yerel optimuma takılmadan arama uzayını geniş tutabilmektedir.

### **3.3. Çok Amaçlı Evrimsel Yöntemler**

#### **3.3.1. Çok Amaçlı Optimizasyonun Tanımı**

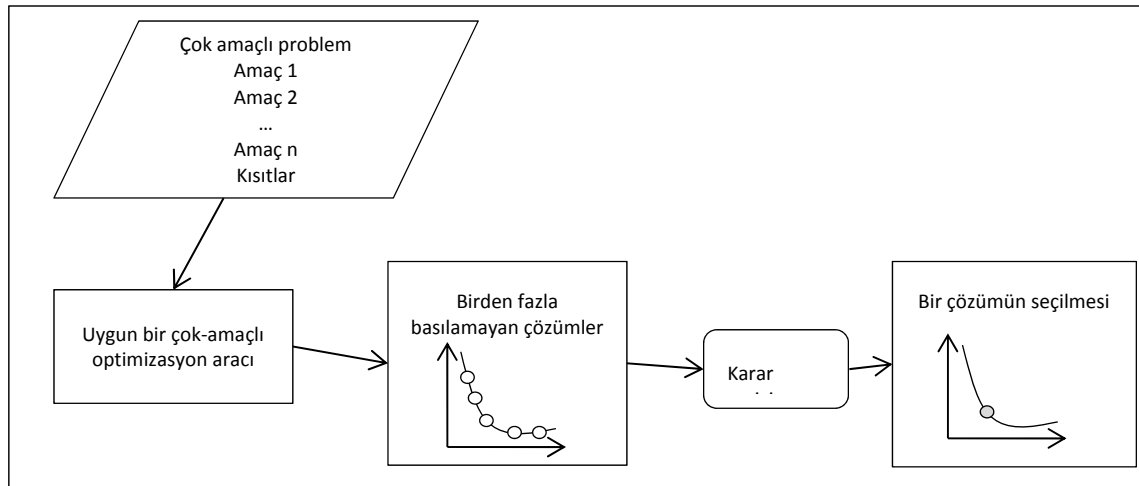
Amaç yada amaçlara yönelik, belirli kriterleri sağlayan çözüm yada çözümler kümesine uygun çözümler kümesi adı verilir [47]. Tek amaçlı optimizasyonda belirli bir amacı en iyi yapan çözüm kümesi bulunur. Ancak çok amaçlı optimizasyonda birden fazla amaç olduğu için çözümlerden bazıları bir diğerine üstünlük sağlayabilir. Bir amaca göre iyi olan çözüm diğer bir amaca göre kötü bir çözüm olabiliyor. Örnek vermek gerekirse birinci amaç kaliteyi arttırmak, ikinci amaç maliyeti azaltmak olarak düşünüldüğünde hem kaliteli hem de düşük maliyetli çözüm bazen bulunamayabilir. Hatta kalitenin çok iyi olduğu durumda maliyetin en yüksek olduğu durum ortaya çıkabilir.

Öte yandan gerçek hayat problemlerinin birçoğu çok amaçlı bir yapıya sahiptir [48]. Maliyet minimizasyonu, performans maksimizasyonu, kalite maksimizasyonu, fire minimizasyonu gibi amaçların 2 veya daha fazlası genellikle bir arada kullanılırlar.

Çok amaçlı problemleri çözmek daha zordur. Dolayısıyla çözüm yöntemlerinin çalışması tek amaçlı yöntemlere göre daha uzun sürer.

Burada çıkan sonuçlar genellikle birden fazladır. Çünkü birbirine üstünlük sağlayamayan çözümler birden fazla olabilir. Eğer tüm olası çözümler içerisinde, tüm amaçlarda diğerlerinden daha iyi bir çözüm bulunmuş ise o zaman o çözüm direk işleme alınabilir. Ancak amaçlardan birisinde diğer çözümlerden iyi, fakat diğer amaçlarda diğer tüm çözümlerden iyi değilse o zaman burada en iyi çözümden söz edilemez. Sonuç olarak çözümler kümesi elde edilir. Bu çözümler kümesinin içerisinde bir tanesinin seçilip işleme alınması gerekmektedir.

Çelişen amaçlara göre oluşturulan çözümler kümesinin içerisinde bir tanesinin seçilmesi işini karar verici tarafı üstlenir. Bu karar vericinin seçim için kullandığı bilgiye üst düzey bilgi denir. Ve genellikle bu karar organizasyonun yöneticileri tarafından verilir.

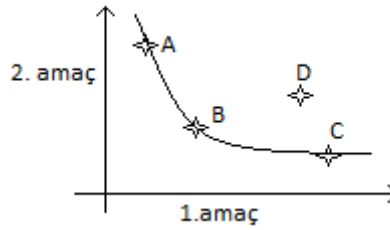


Şekil 3. 20 Çok amaçlı optimizasyonun genel şeması

Çok amaçlı problemlerin çözümü için tek amaçlı meta-sezgisel yöntemlerden yararlanılır. Bunlardan en popüler olanı genetik algoritmadır. Bir çok araştırmacı çok amaçlı problemi çözmek için genetik algorithmadan yararlanmışlardır [48][49][50]. Genetik algoritma çeşitli değişikliklerle çok amaçlı hale çevrilir.

Marler ve Arora [50], çözümler kümesi üzerinde seçim işlemini kolaylaştırmak için karar vericiye bir takım bilgiler sağlar. Bu bilgiler her bir çözümün sahip olduğu değer arasındaki farkları ortaya koyar. Bu farklara göre önceden belirlenmiş tercihlere göre çözümler arasından bir tanesinin seçilmesi sağlanır. Burada amaçlar arasında ağırlıklandırma, hedef programlama, özel kısıtlar ekleme şeklinde teknikleri kullanılabilir. Bu tür işlemlere sonsal eklemeler(posteriori articulation) denir.

### 3.3.2. Pareto Verimliliği



Şekil 3. 21 Pareto verimliliğine göre çözümlerin gösterimi

Çok amaçlı problemlerde birden fazla amacın çeliştiği durumlarda bir ama diğerine üstünlük sağlayamıyorsa o zaman elde edilen çözümlerden tüm amaçlara göre basılamayan çözümler pareto optimal yada pareto verimli çözüm olarak ele alınır [47].

İki amaçlı problem ele alındığında, bir çözümün basılamayan çözüm olabilmesi için birinci amaç fonksiyonuna göre diğer çözümlerden daha iyi olması yada 1 amaç fonksiyonuna göre daha iyi olmadığı çözümlerde ikinci amaç fonksiyonuna göre daha iyi olması gerekiyor. Eğer herhangi bir çözüm ele alınan çözümü her iki amaç fonksiyonunda birden geçiyorsa yani daha iyi bir çözüm sunuyorsa çözüm basılmış demektir[47]. Şekil 3. 21'deki örnekte bir minimizasyon problemi olduğu varsayalım. A noktası birinci amaç fonksiyonuna göre bakıldığında B, C ve D noktasından daha iyi sonuç vermiştir, ikinci amaç fonksiyonuna göre bakıldığında ise B,C ve D den daha kötü sonuç vermiştir. Burada A noktasını her iki amaç fonksiyonunda birden geçen herhangi bir çözüm yoktur. Çünkü birinci amaç fonksiyonuna göre tüm çözümlerden iyidir. Dolayısıyla A noktası pareto-optimal çözümlerden birisidir. C noktası birinci amaç fonksiyonuna göre A,B ve D çözümlerinden daha kötü ancak ikinci amaç fonksiyonuna göre en iyi çözüm olduğu için aynen A noktası gibi C'de pareto-optimal çözümdür. B noktası iki amaç fonksiyonunda da en iyi çözümü sunmamaktadır. Buna rağmen her iki

amaç fonksiyonunda da B noktasını geçen bir çözüm bulunmadığından o da pareto-optimal noktadır. Şöyle ki B noktası A noktasına göre 2.amaç fonksiyonuna göre daha iyi olduğundan A noktası tarafından basılamaz, C noktasına göre 1.amaç fonksiyonunda daha iyi olduğundan C noktası tarafından da basılamaz, D noktasına göre de 1.amaç fonksiyonunda iyidir. O yüzden tüm noktalar denendiğinde B noktasına baskın gelen bir nokta bulunmamaktadır. D noktası birinci amaç fonksiyonuna göre C noktasından daha iyi, ikinci amaç fonksiyonuna göre A noktasından daha iyidir. A ve C noktaları D noktasını basamaz. D noktası birinci amaca göre B noktasından daha kötü, ikinci amaca göre yine B noktasından daha kötüdür. Bu nedenle D noktası B noktası tarafından basılmıştır. Dolayısıyla D noktası pareto-optimal değildir.

Bu şekilde tüm çözüm kümelerinin pareto verimliliği tespit edilir. Pareto-optimal noktaların oluşturduğu kümeye pareto-ön kümesi denir [47]. Karar verici bu pareto-ön kümesinin elemanlarından birini seçer.

### **3.3.3. Kuvvet Pareto Evrimsel Algoritma (SPEA, SPEA2)**

Zitzler ve Thiele [51]'in SPEA algoritmasında basılamayan çözümler elitist olarak bir sonraki nesle aktarılır. Diğer çözümler harici popülasyon olarak saklanır. Eğer popülasyon bu elitist, basılamayan çözümlerle doldurulamadıysa geri kalan elemanlar harici popülasyondan seçilir. Ek olarak deterministik bir yöntemle sınıflandırma yaparak basılamayan çözümlerde çeşitlilik sağlanmıştır. Bu algoritma yazara göre sırt çantası problemlerinde iyi sonuçlar vermektedir.

Zitzler vd. [52], SPEA2 adında daha geliştirilmiş bir algoritma tanımlamışlardır. SPEA'ya göre 3 kısımda geliştirme yapılmış ve daha iyi sonuçlar elde edilmiştir. Her bireyin kaç bireyden baskın geldiği ve kaç birey tarafında basıldığını tutan bir uygunluk şeması eklendi. En yakın komşuluk yoğunluk tahmin tekniği sayesinde arama işleminde öngörücü rehber özelliği kazandı. Sınır çözümlerini bulmayı iyileştirecek arşiv kırpma özelliği eklendi.



### **3.3.4. Pareto Eriřimli Evrim Stratejisi (PAES)**

Knowles ve Corne [53] daha basit bir algoritma öne sürmüşler (PAES). Burada 1 birey mutasyona uğrayarak bir çocuk oluşturur, bu çocuk ile birey karşılaştırılır. Eğer çocuk bireyden baskın ise o zaman çocuk bir sonraki popülasyona aktarılır. Eğer birey daha baskın ise o zaman mutasyonla yeni bir çocuk elde edilir. Eğer birey ve çocuk birbirine baskın değil ise, önceki basılamayan çözümler ile karşılaştırılır. Önceki çözümler arşivinde daha baskın bir çözüm varsa o çözüm bir sonraki nesle aktarılır. Eğer çocuk arşivdeki herhangi bir bireyi basabiliyorsa o zaman çocuk da arşive eklenir. Arşivdeki basılan çözüm atılır. Eğer çocuk arşivdeki hiçbir bireyi basamıyorsa o zaman çocuğa en yakın çözümlerin mesafesi (crowding distance) kabul aralığında ise o zaman çocuk arşive eklenir.

### **3.3.5. Bastırılmamış Sıralamalı Genetik Algoritma 2 (NSGA II)**

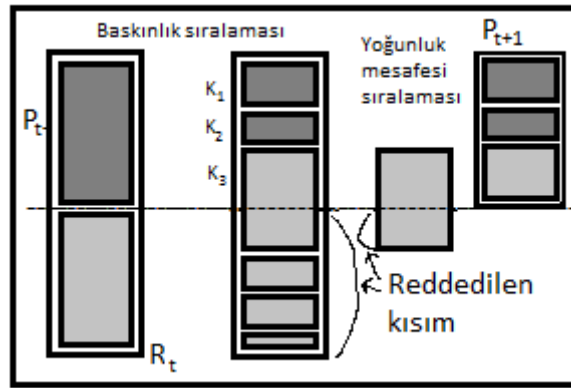
Bu bölümde anlatılacak olan NSGA II algoritması, pareto-öncül elemanları bulma konusunda en popüler çok amaçlı optimizasyon algoritmaları arasındadır. Son tekniklere göre yapılan algoritmaların hemen hepsi elitist yaklaşımlar üzerine kurulmuştur[47]. Bu yaklaşımda, bir sonraki nesile aktarılacak olan bireylerin seçimi ile ilgilenilmektedir. Tek amaçlı genetik algoritmada uygunluk değeri en iyi olan bireyler elit bireyler olarak seçilmekteydi. Ancak çok amaçlı problemde bir amaca göre en iyi olan diğer bir amaca göre en iyi olmayabilir. Bu durumda böyle bir sıralama yapmak mümkün değildir. Bu amaçla elitist yaklaşımlar ortaya çıkmıştır.

Deb [47] tarafından yayınlanan NSGA II algoritması şu üç özellik üzerine kurulmuştur:

- Elitist yaklaşıma sahip
- Çeşitliliği sağlayacak mekanizmaya sahip
- Basılamayan çözümler prensibine dayalı

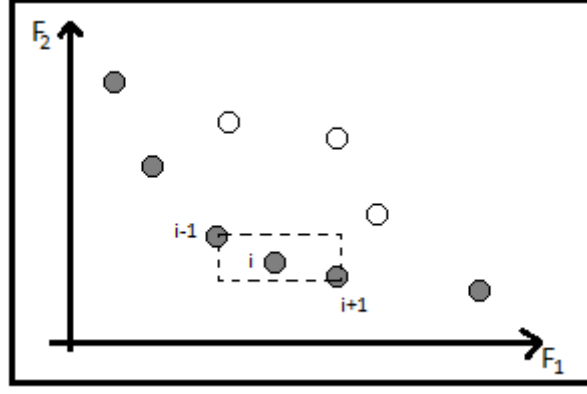
NSGA II standart genetik algoritma operatörleri ile çalışır. Yeni bireyler çaprazlama ve mutasyon ile oluşturulur.

Şekil 3. 22’de NSGA II’de bir sonraki popülasyonu oluşturmak için yapılan seçim işlemi gösterilmektedir. Başlangıçtaki popülasyon sayısı  $N$  olmak üzere, ebeveyn ve çocuklar bir arada toplanarak  $2N$  bireye sahip yeni bir popülasyon ( $R_t$ ) oluşturulur.  $R_t$  kümesinin bireyleri basılamayan çözümler kümeleri olarak sınıflandırılır ( $K_i, i = 1, 2, \dots, n$ ). Birinci basılamayan çözümler kümesi ( $K_1$ ) bir sonraki popülasyona ( $P_{t+1}$ ) aktarılır. Sonra ikinci basılamayan küme ( $K_2$ ) aktarılır ve bu şekilde  $P_{t+1}$  kümesi  $N$  elemanlı olana kadar devam eder.  $P_{t+1}$ ’in kapasitesi dolduktan sonra geriye kalan elemanlar atılır. Ve bir sonraki iterasyona geçilir. Basılamayan kümenin elemanları bir sonraki popülasyona aktarıldığında popülasyon kapasitesini aşıyorsa o zaman kümenin tamamı aktarılmaz. Bu kümenin elemanları yoğunluk mesafesi (crowding distance) yöntemine göre sıralanır ve mesafesi çok olandan başlayarak  $P_{t+1}$  doldurulur.



Şekil 3. 22 NSGA II işlemleri

Yoğunluk mesafesi hesaplaması Şekil 3. 23’de gösterilmektedir. Her birey için bu hesaplama yapılır.  $i$ . birey kendisinden bir önceki ve bir sonraki birey arasında kalacak şekilde bir küboid çizilir. İki boyutlu bir grafikte bu bir dikdörtgendir. Bu küboidin çevresi hesaplanarak yoğunluk mesafesi bulunur. Bu mesafenin kısa olması bireyin önceki ve sonraki bireye yakın olduğu anlamına gelir.



Şekil 3. 23 NSGA II Yoğunluk mesafesi gösterimi

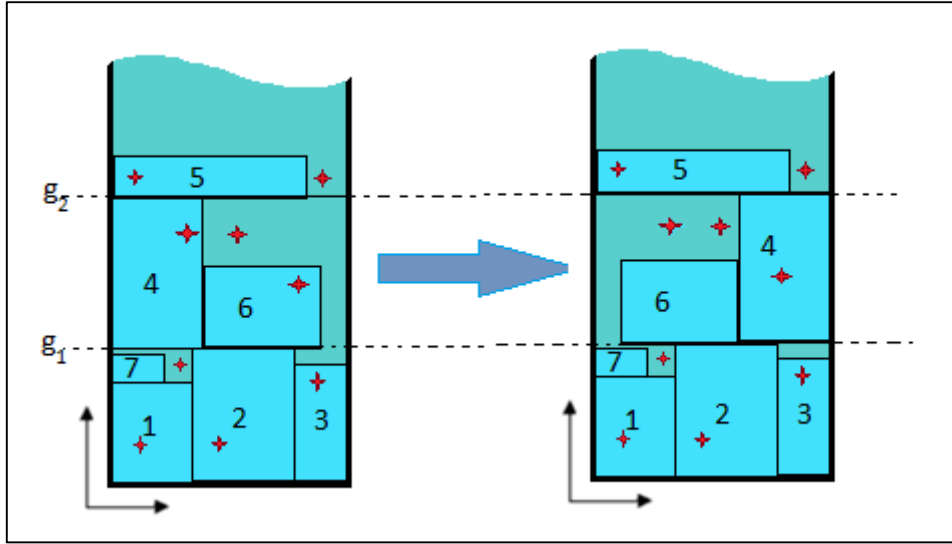
Deb [47]'e göre yoğunluk mesafeleri yakın olan bireyler birbirine benzer çözümlerdir ve çeşitliliği sağlamak için bir sonraki popülasyona birbirine uzak olan bireylerin geçmesi gerekmektedir. Bu sebeple yoğunluk mesafelerine göre sıralanan bireyler büyükten küçüğe doğru bir sonraki popülasyona aktarılır. Ta ki popülasyon kapasitesi dolana kadar.

### 3.3.6. Önerilen Sağ Sol Dengelemeli Genetik Algoritma (GA\_LR)

Bu tez çalışması için önerilen bir algoritmadır. Tek amaçlı genetik algoritma ile çalışma prensipleri benzerdir. Kalite konusunda iyileştirme yapacak bir sezgisel eklenerek çok amaçlı bir hal almıştır.

Bu çalışmada birinci amaç fireyi azaltmak, ikinci amaç ise kaliteyi arttırmak için cam üzerindeki kusurları azaltmaktır. Kusurları da azaltabilmek için genetik algoritmanın yerleşiminde bir düzenlemeye gidilmiştir.

Şekil 3. 24'de sağ sol dengelemeli genetik algoritma örnek yerleşim planı gösterilmektedir. Genetik algoritma uygunluk değerini hesaplamak ve yerleşim planı oluşturmak için sol alt yerleştirme tekniğini kullanmaktadır. Bu tekniğe göre genetik algoritmanın sunduğu çözümler gen sırasına göre sol alt kısımdan başlayarak yerleştirilir. Alt kısma yerleştirilemeyen parçalar bir üst seviyeye çıkarak, giyotin hizası olan  $g_1$  çizgisini belirler. Tabandan  $g_1$ 'e kadar olan kısım birinci giyotin seviyesini gösterir. Bu algoritmada her bir giyotin seviyesinde parçaların sol alt mı yoksa sağ alt mı yerleştirilirse daha uygun olacağını tespit eden bir uygulama yapılmıştır.



Şekil 3. 24 Sağ sol dengelemeli genetik algoritma örnek yerleşim planı

Şekil 3. 24’de  $g_2$  seviyesinde yani  $g_1$  ve  $g_2$  arasında kalan kısım sol alt kısma dayalı olarak yerleştirildiğinde mevcut olan 3 kusurdan iki tanesine denk geliyor iken sağ alt kısma dayalı olarak yerleştirildiğinde bir tane kusura denk gelmektedir. O halde yerleşim planını 2. seviyede sağ alt dayalı olarak yapmak daha iyi sonuçlar vermektedir. Bu sayede genetik algoritma fireyi azaltma yönünde çalışırken, sağ sol dengeleme algoritması kusurları da azaltmaya çalışmaktadır.

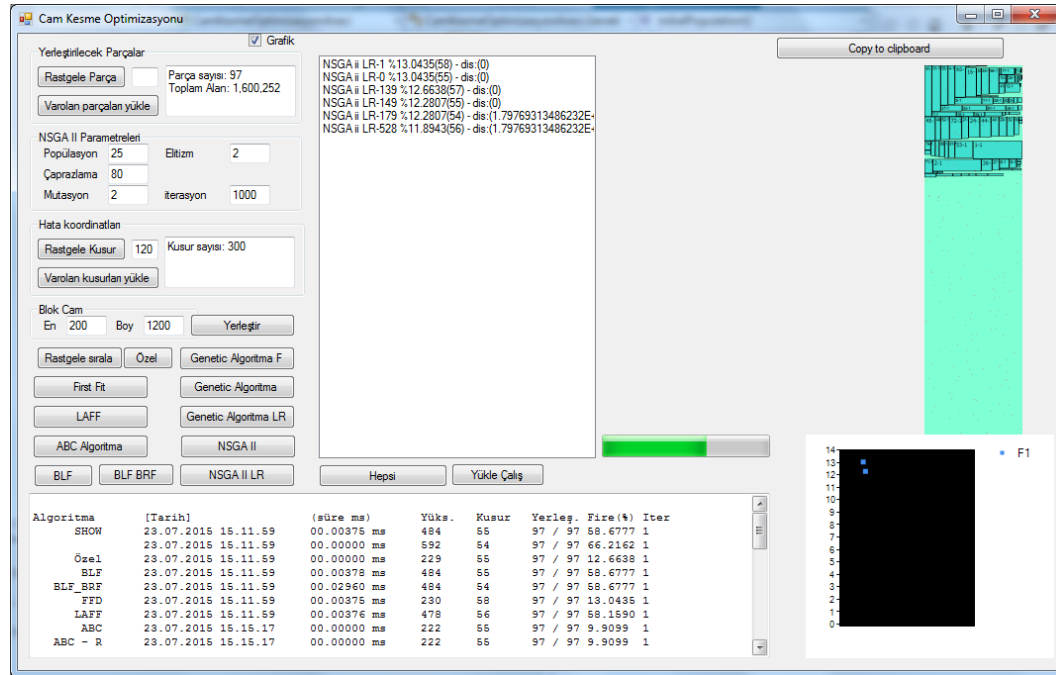
### DENEYSEL ÇALIŞMALAR

#### 4.1. Deney Ortamı

Deney için kullanılan yazılım hazırlanırken Microsoft Visual Studio.Net 2012 ortamı kullanılmıştır. C# dili ile kodlama yapılmıştır. Deneyler aşağıdaki özelliklere sahip bir dizüstü bilgisayar ile yapılmıştır:

- 6 GB Ram
- Intel i7 3517U 1.90 GHz 4 çekirdekli işlemci
- 128 GB SSD harddisk
- 2 GB paylaşımlı ekran kartı

Deney için hazırlanan yazılımın ekran görünümü Şekil 4. 1’de verilmiştir.



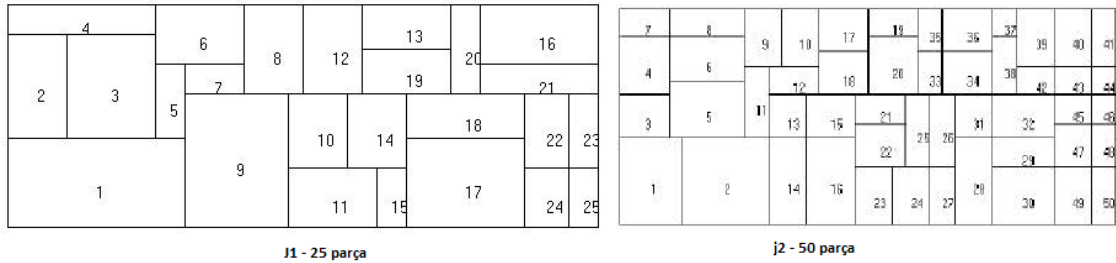
Şekil 4. 1 Optimizasyon yazılımının arayüzü

## 4.2. Veri Kümeleri

Litetürde yer alan veri kümeleri ve sanayiden alınan veriler kullanılmıştır. Aynı zamanda çeşitliliği arttırmak için uygun şekilde rastgele oluşturulan veriler de kullanılmıştır. Literatürdeki veri kümeleri genellikle sıfır fire ile yerleşebilen küçük parçaların cam bloğuna yerleştirilmesi yada kesilmesini öngören planlardır. Ancak bu tez çalışmasında giyotinle kesme makinesinin sadece yatay yönlü bir giyotin kullanması sonucu her zaman sıfır fire bulmak mümkün olmamaktadır. Bu kısıt kullanılan kesme makinesinin yapısından kaynaklandığı için bu şekilde ele alındı.

### 4.2.1. Jacobs veri kümesi

Jacobs [15] 1996 yılında yaptığı çalışmalarda kendi veri kümesini kullanmıştır. Genişliği 40cm olan cam bloğu üzerinde sıfır fire ile yerleşebilen 25 adet küçük cam parçasıyla oluşturduğu kümeye J1 adını vermiştir. Aynı şekilde genişliği 40cm olan cam bloğu üzerinde sıfır fire ile yerleşebilen 50 adet küçük cam parçasıyla oluşturduğu kümeye de J2 demiştir.



Şekil 4. 2 Jacobs'ın J1 ve J2 veri kümesinin optimum sonuçlarından birisi

### 4.2.2. Hopper veri kümesi

Hopper çalışmalarında 1999 [54] ve 2000[55] yıllarında yayınladığı çalışmalarında kullandığı veri kümesini kullanmıştır. Bu da aynı şekilde sıfır fire ile 200 cm cam bloğuna yerleşebilen parçalardan oluşmaktadır. Hopper veri kümesi çok fazla veri içerdiğinden yapılan çalışmaların büyük çoğunluğuna referans olmuştur. Aşağıdaki tabloda veri kümelerinin özellikleri açıklanmıştır.

Çizelge 4. 1 Hopper'ın veri kümelerinin özellikleri

| İsim                    | Parça Sayısı | Optimum cam bloğu |
|-------------------------|--------------|-------------------|
| T1a, T1b, T1c, T1d, T1e | 17           | 200x200           |
| T2a, T2b, T2c, T2d, T2e | 25           | 200x200           |
| T3a, T3b, T3c, T3d, T3e | 29           | 200x200           |
| T4a, T4b, T4c, T4d, T4e | 49           | 200x200           |
| T5a, T5b, T5c, T5d, T5e | 72           | 200x200           |
| T6a, T6b, T6c, T6d, T6e | 97           | 200x200           |
| T7a, T7b, T7c, T7d, T7e | 199          | 200x200           |

#### 4.2.3. ZDF Büyük veri kümesi

Zhang vd [56] 2013 de yayınladığı iki arama sezgisel algoritması isimli çalışmasında varolan veri kümelerinin kapasitesini yeterli bulmayıp daha fazla parça içeren veri kümeleri oluşturmuştur. Ancak bilgisayarda hesaplama süresi fazla olduğu için evrimsel algoritmalar biraz ağır kalmaktadır. Bu nedenle Zhang evrimsel olmayan bir sezgisel kullanarak çözüme gitmiştir.

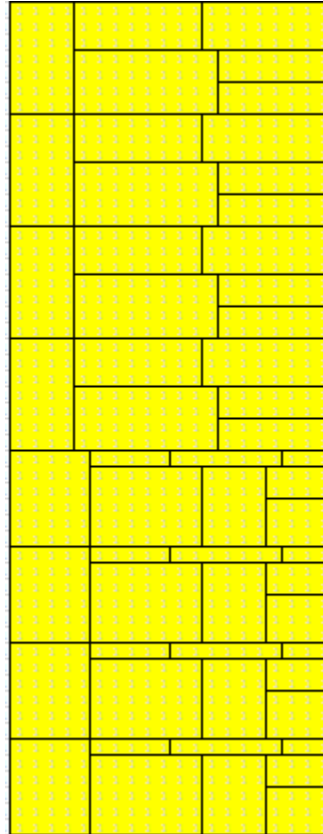
Çizelge 4. 2 ZDF veri kümelerinin özellikleri

| İsim | Parça Sayısı | Optimum cam bloğu |
|------|--------------|-------------------|
| Z1   | 580          | 100               |
| Z2   | 660          | 100               |
| Z3   | 740          | 100               |
| Z4   | 820          | 100               |
| Z5   | 900          | 100               |
| Z6   | 1532         | 3000              |
| Z7   | 2432         | 3000              |
| Z8   | 2532         | 3000              |
| Z9   | 5032         | 3000              |
| Z10  | 5064         | 3000              |
| Z11  | 7564         | 3000              |
| Z12  | 10064        | 3000              |
| Z13  | 15096        | 3000              |
| Z14  | 25032        | 3000              |
| Z15  | 50032        | 3000              |
| Z16  | 75032        | 3000              |

#### 4.2.4. Valenzuela Veri kümesi

Valenzuela ve Wang [57] tarafından firesiz olarak istenilen ölçülerde ve istenilen parça sayısına sahip veri kümesi oluşturmak için bir yöntem geliştirildi. Bu yöntemde en-boy oranı dengeli veya alan ölçüleri dengeli olacak şekilde küçük parça üretimi sağlanabilir. Bu özellikler de çalışmalarında yer almıştır. Bu çalışmada rastgele ölçülerde herhangi bir kurala bağlı olmadan üretilen verilere “path” ismi verildi. Örneğin 50 parça içeren bir kümeye path50 adı verildi. En boy ölçüleri orantılı olacak şekilde üretilen verilere de “nice” ismi verildi. Aynı örnekte 50 parçadan oluşan veri kümesine nice50 denildi. Bu şekilde istenilen veri kümesi rahatlıkla oluşturularak yapılan çalışmalara büyük kolaylık getirildi. Ancak bu yöntemde tek dezavantaj hazır bir veri kümesi olmadığı için yapılan çalışmalardaki veriler sıfır fireli olmasına karşın aynı veri kümesi üretilmeyeceğinden çalışmalarda kullanılan yöntemlerin karşılaştırılması biraz zayıf kalmaktadır.

#### 4.2.5. Sanayi Veri kümesi



Şekil 4. 3 Sanayi veri kümesinin optimum sonuçlarından birisi



Cam sektörünün önde gelen temsilcilerinde birisi ile görüşülerek örnek bir sipariş kümesi tedarik edildi. Bu kümede farklı ölçülerde 56 adet küçük cam siparişi bulunmaktadır. Hesaplanması istenilen cam bloğunun genişliği ise 200 cm'dir. Yapılan incelemelerde bu siparişlerin sıfır fire ile üretilebildiği tespit edilmiştir. Dolayısıyla bu da sıfır fireli bir veri kümesi olmuştur. Bu küme "San" olarak adlandırılacaktır.

#### 4.2.6. Kusur Veri kümesi

Bu veriler literatürde henüz yer almamaktadır. Cam bloğu üzerindeki kusurları bu tez çalışmasının orijinal ürünü olduğundan veri kümesi sanayi temsilcisi ile görüşülüp oluşturulmuştur. Alınan bilgiler eşiğinde cam üzerindeki kusurlar rastsal olarak istenilen sayıda hazırlanıp deneyler o rastsal kusurlar üzerinden yapılmıştır. Bu küme  $K_n$  olarak adlandırılacaktır.

#### 4.3. Deney Sonuçları

Deneylerde kullanılan parametreler ve hangi algoritmalarda kullanıldığı özet olarak şu şekildedir.

Çizelge 4. 3 Algoritmalarda kullanılan parametreler

| Parametre             | Değer     | Algoritma                      |
|-----------------------|-----------|--------------------------------|
| Popülasyon Sayısı     | 25 adet   | GA, ABC, NSGA                  |
| Elitizm sayısı        | 2 adet    | GA                             |
| Çaprazlama Yüzdesi    | %80       | GA, ABC, NSGA                  |
| Mutasyon Yüzdesi      | %2        | GA, ABC, NSGA                  |
| İterasyon Sayısı      | 10000 kez | GA, ABC, NSGA                  |
| Blok Cam En           | 200 cm    | FFD, BLF, LAFF , GA, ABC, NSGA |
| Blok Cam Boy (mm)     | 1200 cm   | FFD, BLF, LAFF , GA, ABC, NSGA |
| Blok Cam Kusur Sayısı | 300 adet  | FFD, BLF, LAFF , GA, ABC, NSGA |

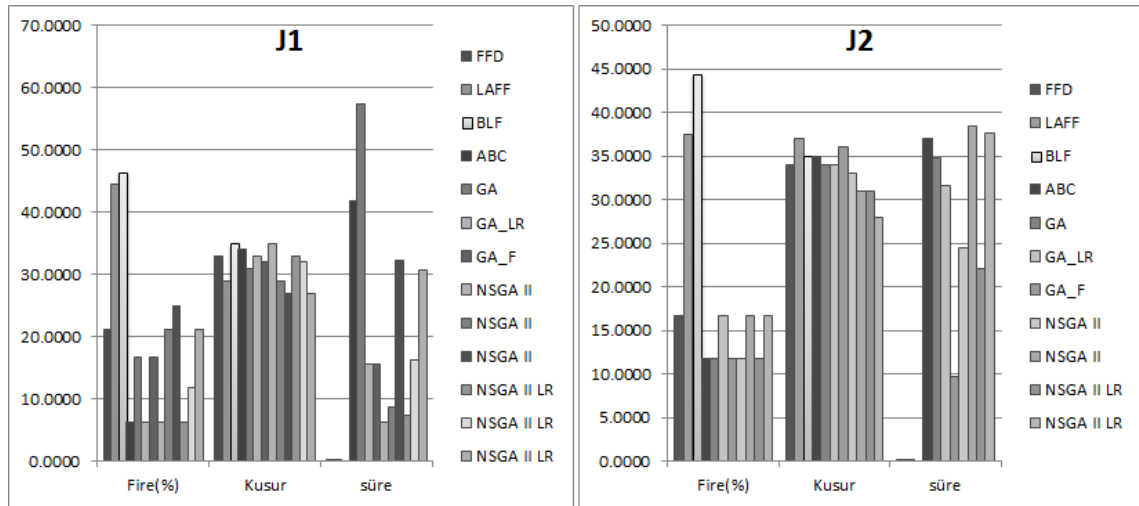
Bu parametreler ışığında çıkan sonuçlar aşağıda belirtilmiştir.

Çizelge 4. 4 Jakops'ın veri kümelerinin sonuçları (j1 ve j2)

| j1      | FFD     | LAFF    | BLF     | ABC      | GA         | GA_LR    | GA_F     |
|---------|---------|---------|---------|----------|------------|----------|----------|
| Fire(%) | 21.0526 | 44.4444 | 46.4286 | 6.2500   | 16.6667    | 6.2500   | 16.6667  |
| Kusur   | 33      | 29      | 35      | 34       | 31         | 33       | 32       |
| süre    | 0.03122 | 0.02473 | 0.00059 | 41.89704 | 57.46676   | 15.5581  | 15.63384 |
|         | NSGA II |         |         |          | NSGA II LR |          |          |
| Fire(%) | 6.2500  | 21.0526 | 25.0000 |          | 6.2500     | 11.7647  | 21.0526  |
| Kusur   | 35      | 29      | 27      |          | 33         | 32       | 27       |
| süre    | 6.27532 | 8.59587 | 32.245  |          | 7.35156    | 16.35331 | 30.74355 |

| J2      | FFD      | LAFF    | BLF        | ABC      | GA       | GA_LR    | GA_F    |
|---------|----------|---------|------------|----------|----------|----------|---------|
| Fire(%) | 16.6667  | 37.5000 | 44.4444    | 11.7647  | 11.7647  | 16.6667  | 11.7647 |
| Kusur   | 34       | 37      | 35         | 35       | 34       | 34       | 36      |
| süre    | 0.00079  | 0.0009  | 0.00072    | 37.09048 | 34.72578 | 31.67813 | 9.72556 |
|         | NSGA II  |         | NSGA II LR |          |          |          |         |
| Fire(%) | 11.7647  | 16.6667 | 11.7647    | 16.6667  |          |          |         |
| Kusur   | 33       | 31      | 31         | 28       |          |          |         |
| süre    | 24.49113 | 38.4436 | 22.04389   | 37.59187 |          |          |         |

Jakops'ın veri kümelerinin bakıldığında sezgisel algoritmalar kusur değerlerinde iyi sonuçlar elde edebilmelerine karşı fire konusunda çok başarısız sonuçlar vermiştir. Genetik algoritma (GA) her iki amaç değerinde de sezgisellere göre daha iyi sonuç vermiştir. Önerilen sağ sol dengelemeli genetik algoritma (GA\_LR) ise fire bakımından genetik algortmadan çok daha iyi sonuç elde edebilmesine karşın kusur oranını bir miktar arttırmıştır. Parçaları döndürebilme özelliği eklenen GA\_F algoritması da GA ile benzer sonuçlar üretmiştir. Yapay arı algoritması (ABC) fire olarak güzel sonuçlar elde etmesine karşın kusur bakımından diğerlerine göre daha iyi sonuç elde edememiştir. Çok amaçlı bir algoritma olan NSGA II algoritması da önerilen GA\_LR ile benzer ancak kusur olarak biraz kötü sonuç elde etmiştir. Önerilen sağ sol dengelemeli NSGA II algoritmasında (NSGA II LR) da en iyi sonuçlar elde edilmiştir.



Şekil 4. 4 Jakops'ın veri kümelerinin sonuçları

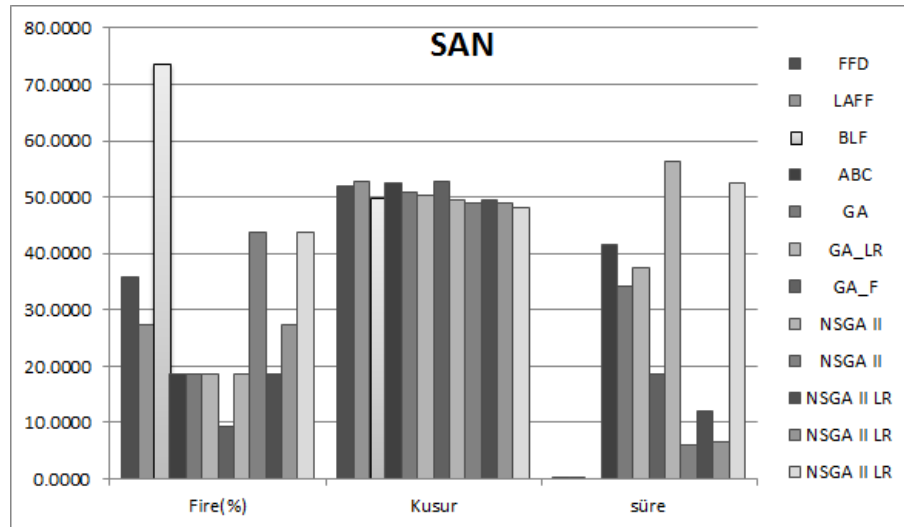
Çok amaçlı algoritmalarda tek bir iyi çözüm yerine birden fazla çözümler sunulmuştur. Bunun sebebi birinci amaç fonksiyonuna göre daha iyi olan bir çözüm ikinci amaca göre

daha kötü olabilir. Bu iki çözüm arasında karar verme üst düzey bilgi gerektirir. Bunu bir karar vericinin yapması uygun görülmektedir.

Çizelge 4. 5 Sanayi veri kümesinin sonuçları

| SAN     | FFD      | LAFF    | BLF        | ABC     | GA       | GA_LR  | GA_F   |
|---------|----------|---------|------------|---------|----------|--------|--------|
| Fire(%) | 7.1429   | 5.4545  | 14.7541    | 3.7037  | 3.7037   | 3.7037 | 1.8868 |
| Kusur   | 130      | 132     | 125        | 131     | 127      | 126    | 132    |
| süre    | 0.00259  | 0.00261 | 0.00273    | 0       | 0        | 0      | 0      |
|         | NSGA II  |         | NSGA II LR |         |          |        |        |
| Fire(%) | 3.7037   | 8.7719  | 3.7037     | 5.4545  | 8.7719   |        |        |
| Kusur   | 124      | 122     | 124        | 122     | 120      |        |        |
| süre    | 56.31868 | 5.97399 | 12.03528   | 6.51373 | 52.51298 |        |        |

Bu verikümesinde de benzer sonuçlar elde edilmiştir. Önerilen GA\_LR algoritması genetik algoritma ile aynı fire oranını bulmasına karşın asıl amaçlanan kusur oranını azaltma konusunda da başarılı olmuştur. Aynı şekilde Önerilen NSGA II LR algoritması da NSGA II algoritmasına göre daha faydalı sonuçlar üretmeyi başarmıştır.



Şekil 4. 5 Sanayi veri kümesinin sonuçları

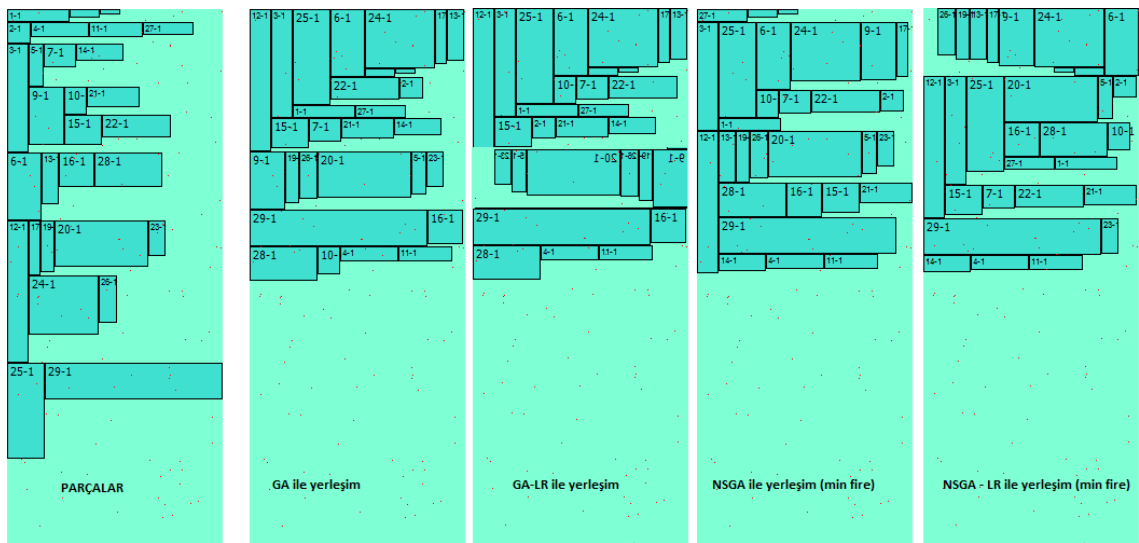
Hopper veri kümelerinin sonuçları da benzer özellikler göstermektedir. Çizelge 4.6 da da tüm veri kümelerinin sonuçları gösterilmektedir.

Çizelge 4.6'dan da anlaşılacağı gibi önerilen algoritmalar kusur azaltma yönünde daha iyi sonuçlar vermektedir. Şekil 4.6'da hopper verisetindeki T3a parçalarının kesim planı gösterilmiştir. Bilindiği üzere NSGA algoritmasında fire ve kusur minimizasyonu yapıldığı için birden fazla kesim planı oluşmaktadır. Şekil 4.6'da bu planlardan sadece

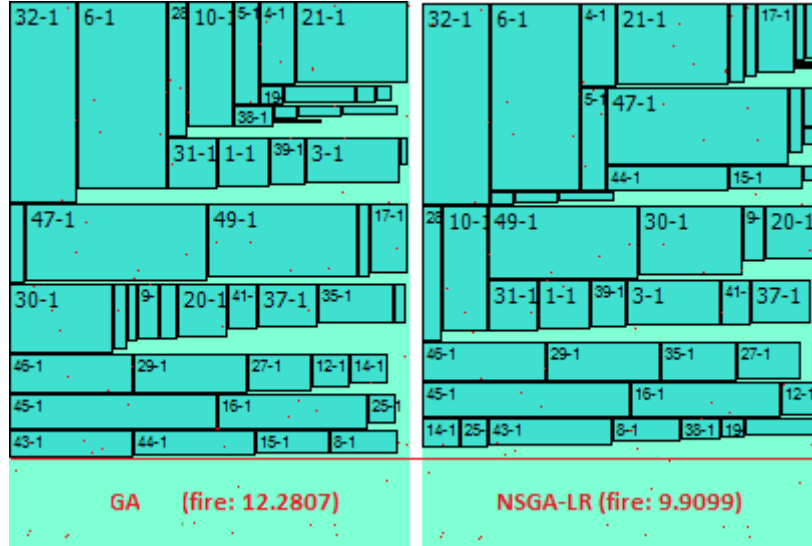
fireyi minimize eden çözüm gösterilmiştir. Şekilden de anlaşıldığı üzere NSGA ile elde edilen çözümler fire bakımından GA ile elde edilen çözümlerden biraz daha iyi sonuç vermiştir. GS-LR ve NSGA-LR çözümlerine bakıldığında bası parçaları sağdan sola doğru yerleştirilmiştir. Bunun sebebi; o bölgede kusurların yoğunluğu sol tarafta olduğu için sağ taraftan yapılan yerleşimde kusur bakımından daha iyi sonuç elde etmektir.

Çizelge 4. 6 Hopper veri kümesinin sonuçları

|     |         | GA      | GA_LR   | NSGA II  | NSGA II | NSGA II | NSGA II  | NSGA II | NSGA II  |
|-----|---------|---------|---------|----------|---------|---------|----------|---------|----------|
|     |         |         |         |          |         |         | LR       | LR      | LR       |
| T1a | Fire(%) | 16.6667 | 15.9664 | 15.9664  | 15.9664 | -       | 15.9664  | 17.3554 | 21.5686  |
|     | Kusur   | 57      | 55      | 57       | 53      |         | 53       | 51      | 45       |
|     | süre    | 38.2568 | 41.7377 | 6.4443   | 21.4325 |         | 3.6791   | 38.1256 | 55.2423  |
| T2a | Fire(%) | 14.1631 | 11.1111 | 14.1631  | -       | -       | 11.1111  | 14.1631 | 17.3554  |
|     | Kusur   | 52      | 50      | 51       |         |         | 50       | 48      | 46       |
|     | süre    | 7.6586  | 30.0783 | 31.7915  |         |         | 25.8583  | 18.7915 | 30.0394  |
| T3a | Fire(%) | 15.6118 | 14.1631 | 14.1631  | 18.3673 | -       | 14.1631  | 15.6118 | 18.3673  |
|     | Kusur   | 56      | 54      | 53       | 51      |         | 53       | 51      | 48       |
|     | süre    | 24.5326 | 31.6103 | 15.9483  | 32.1254 |         | 15.1883  | 22.8744 | 44.9112  |
| T4a | Fire(%) | 11.8943 | 12.2807 | 11.5044  | 12.2807 | 14.5299 | 9.9099   | 14.1631 | -        |
|     | Kusur   | 58      | 54      | 52       | 51      | 49      | 53       | 49      |          |
|     | süre    | 11.2556 | 16.5988 | 4.7415   | 16.5988 | 24.8444 | 33.3540  | 32.5186 |          |
| T5a | Fire(%) | 12.6658 | 10.7143 | 10.7143  | 13.4199 | -       | 10.3139  | 12.2807 | 16.3180  |
|     | Kusur   | 57      | 55      | 53       | 51      |         | 51       | 51      | 49       |
|     | süre    | 19.2565 | 29.3536 | 57.3773  | 22.5712 |         | 62.9556  | 56.1658 | 56.9614  |
| T6a | Fire(%) | 13.0435 | 12.6658 | 10.7143  | 11.1111 | -       | 9.9099   | 13.0435 | -        |
|     | Kusur   | 56      | 55      | 52       | 51      |         | 53       | 50      |          |
|     | süre    | 32.7745 | 42.9888 | 72.3790  | 66.5655 |         | 45.3658  | 34.1146 |          |
| T7a | Fire(%) | 2.3438  | 1.6716  | 1.1858   | 2.3438  | -       | 1.1858   | 2.2483  | 4.8665   |
|     | Kusur   | 254     | 253     | 253      | 252     |         | 252      | 250     | 232      |
|     | süre    | 59.5116 | 71.5300 | 112.8522 | 84.3563 |         | 106.8522 | 91.9589 | 125.5552 |



Şekil 4. 6 Hopper T3a veri kümesinin örnek kesim planı



Şekil 4. 7 Hopper T4a veri kümesinin GA ve NSGA-LR kesim planı karşılaştırması

Şekil 4.7 de görüldüğü gibi, NSGA-LR ile bulunan çözümde daha az fire verilerek kesim planı oluşturuldu. Aynı zamanda NSGA-LR nin planına bakıldığında kusur bakımından GA dan daha iyi sonuç verdiğini Çizelge 4.6'dan görebiliriz.

Valezuela veri kümesi sonuçlarında da bu tezi destekleyecek şekilde çözümler elde edilmiştir.

Ayrıca ZDF büyük veri kümelerinin çalışma süreleri çok fazla sürmektedir. Bu veri kümesinde evrimsel algoritmalar yerine önerilen BLF-BRF algoritmasının kullanılmasının daha uygun olacaktır. Buna ek olarak sanayi temsilcisi ile görüşüldüğünde gerçek hayat uygulamalarında bu tarz büyük veriler üzerinde çalışmanın günümüzde gerekli olmadığı savunulmuştur. Bu nedenle bu veri kümesinin problemin kapsamı dışında kaldığı kabul edilmiştir.

### SONUÇ VE ÖNERİLER

Bu tez çalışmasında cam kesme problemi incelenmiştir. Bu problem literatürde stok kesme ve paketleme problemleri arasında yer almaktadır. Dikdörtgensel düz parçalar giyotinli kesme yöntemiyle kesilecek şekilde çalışmalar yapılmıştır.

Cam kesme problemi NP\_Tam sınıfında bir problem olduğundan çözüm yöntemleri olarak deterministik yöntemler ile sonuç elde edilememiştir. Deterministik çözümlerden elde edilecek optimum çözümlerin bulunması teoride mümkündür ancak kabul edilebilir bir sürede çözüm elde edilememektedir. Bu nedenle değişik sezgisel algoritmalar kullanılmak kaçınılmazdır. Sezgisel algoritmalar bir takım varsayımlarda bulunarak çözümü iyileştirme yoluna giderler ancak buldukları çözümden daha iyi bir çözüm olup olmadığı konusunda bir fikir vermezler. Yeteri kadar iyi bir çözüm bulunduğu anda algoritma son bulur ve bu çözüm olarak kabul edilir.

Stok kesme sırasında oluşan atık parçalar üretime herhangi bir katkısı olmadığı gibi hammaddenin de verimli kullanılmamasına yol açmaktadır. Dolayısıyla kesme problemlerindeki temel amaç fire miktarını en aza indirmektir. Bu sayede hammadde maliyetini düşürerek firma kazanç elde etmeyi hedeflemiştir.

Buna ek olarak firmaların göz önüne alması gereken başka unsurlar da vardır. Kalite bunların başında gelir. Kaliteli bir ürün firma için itibar anlamına da gelmektedir. Kaliteyi etkileyen en önemli unsur cam üzerinde bulunan kusurlu kısımların sayısıdır. Bazı durumlarda tamamen kusursuz üretim zorunludur. Ancak bazı durumlarda tamamen kusursuz olmasına gerek duyulmamaktadır. Bir miktar kusur müşteriler tarafından kabul edilebilmektedir. Tabi ki bu kalite düşüşü ürünün fiyatını da etkilemektedir. Bu nedenle üretici firma her zaman kaliteyi en üst düzeyde tutmak

istemektedir. Bu nedenle cam kesme problemine kusurları en aza indirecek yeni bir amaç fonksiyonu tanımlanmıştır. Blok camdaki kusurların önceden tespiti sağlanmış ve cam kesim planında o kusurlara denk gelecek sipariş sayısını azaltmak hedeflenmiştir. Bu kusurlu kısımlar da fire olarak dışarı atılması amaçlanmıştır.

Cam kesme işlemi sırasında firelerin oluşması neredeyse kaçınılmaz olmaktadır. Bu nedenle kusurlar firelere denk gelecek şekilde plan yapılarak aynı fire ile daha kaliteli ürünler çıkarılmıştır. Bu kusur enküçüklemesi sayesinde üretici firmanın ürün kalitesi artmıştır ve aynı zamanda piyasadaki saygınlığı artmıştır.

Bu yeni amaç fonksiyonu ile beraber problem çok amaçlı optimizasyon problemi halini almıştır. Çok amaçlı problemlerde tek bir sonuç elde edilememektedir. Birinci amaca göre iyi olan bir çözüm diğer amaçlara göre kötü olabilir. Çok amaçlı problemlerde çözümler kümesi elde edilir. Buna da “pareto front” çözümler denir. Pareto çözümler birbirine karşı baskın olmayan çözümlerin kümesidir. Bu çözümlerin elde edilmesi ile birlikte planlama yapan kişinin bunlardan kendisine uygun olanı seçmesi beklenir. Bu seçim işlemi de çok kriterli karar verme probleminin konusudur. Bu tarz karar verme problemlerinde en temel yaklaşım üst düzey bilgi ve tecrübe gerektiren uzman görüşü ile sağlanmaktadır. Uzman, elindeki seçeneklerden yola çıkarak başka bir takım yönetimsel bilgileri de kullanarak bir kesim planına karar verir.

Bu tez çalışmasında kesme planını oluştururken sol alt yerleştirme algoritması temel alınmıştır. Bu algoritma, yerleştirilecek olan parça en sol attan başlayarak ilk uygun yere yerleştirilmektedir. Bu yerleştirme tipine göre eğer kusurlu bölge sol tarafta bulunuyorsa kesilecek parça kusura denk gelmektedir. Önerilen algoritma sayesinde kusurlar göz önüne alınarak sol ya da sağ alttaki ilk uygun yere yerleştirme işlemi yapılarak mümkün olduğunca kusurlardan kaçış sağlanmıştır.

İkinci olarak pareto çözümleri bulmak için NSGA II algoritması temel alınmıştır. Bu algoritma, hem fireyi azaltmaya çalışırken hem de kusurları azaltmaya çalışmaktadır. Buna ek olarak sağ sol dengelemeli bir çözüm önerisi ile beraber yukarıdaki benzer bir mantıkla kusurlarda daha da iyileştirilme sağlanmıştır.

Sonuç olarak, bu tez kapsamında sunulan çözüm önerileri sayısal örneklerle çalışılmış ve etkinliği bir önceki bölümde açıkça gösterilmiştir. Amaçlanan sonuçlara çözüm

olarak yeni önerilen algoritmaların kullanılabilir olduğu anlaşılmış ve bu çalışmanın hedefine ulaşılmıştır. Ayrıca bu tez kapsamında işbirliği içinde bulunan sanayi temsilcisi tarafından da etkinliği kabul görmüştür.

Bu çalışmayı geliştirmek isteyen diğer araştırmacılar için öneriler şu yöndedir.

- Diğer çok amaçlı evrimsel algoritmalar denenebilir.
- Kesilecek parçalarda döndürme özelliği eklenerek firesi ve kusuru daha az üretim yapılmaya çalışılabilir.
- BLF-BRF algoritması biraz daha geliştirilerek kusur minimizasyonunda daha iyi hale getirilebilir.
- Siparişlerin teslim süresi gibi yeni bir amaç fonksiyonu eklenebilir.
- Çıkan pareto sonuçları arasında karar verme sağlayacak yeni bir takım özellikler ve kısıtlamalar getirilebilir.

Bu tez kapsamında yapılan çalışmaların, bu konuda çalışması bulunan diğer araştırmacılar için bir ilham kaynağı olması ve katkılarının bulunması temennimizdir.



## VERİ KÜMELERİ

## A-1 Kusur Veri Kümesi

300 adet kusur 200x1200 birimlik alana dağılmış vaziyette.

| #  | X   | Y    | #  | X   | Y    | #   | X   | Y    | #   | X   | Y    | #   | X   | Y    | #   | X   | Y    |
|----|-----|------|----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 1  | 92  | 260  | 51 | 66  | 855  | 101 | 140 | 68   | 151 | 68  | 278  | 201 | 16  | 804  | 251 | 198 | 860  |
| 2  | 192 | 939  | 52 | 106 | 1146 | 102 | 193 | 123  | 152 | 164 | 638  | 202 | 162 | 261  | 252 | 156 | 932  |
| 3  | 167 | 981  | 53 | 25  | 164  | 103 | 94  | 752  | 153 | 155 | 872  | 203 | 191 | 802  | 253 | 107 | 222  |
| 4  | 139 | 1073 | 54 | 169 | 623  | 104 | 160 | 440  | 154 | 127 | 848  | 204 | 29  | 1042 | 254 | 130 | 1001 |
| 5  | 157 | 648  | 55 | 96  | 501  | 105 | 125 | 21   | 155 | 151 | 339  | 205 | 95  | 557  | 255 | 165 | 306  |
| 6  | 198 | 57   | 56 | 82  | 403  | 106 | 75  | 961  | 156 | 52  | 444  | 206 | 153 | 469  | 256 | 168 | 636  |
| 7  | 139 | 207  | 57 | 105 | 679  | 107 | 50  | 429  | 157 | 162 | 2    | 207 | 43  | 934  | 257 | 152 | 685  |
| 8  | 51  | 453  | 58 | 31  | 789  | 108 | 51  | 715  | 158 | 49  | 194  | 208 | 149 | 496  | 258 | 69  | 1048 |
| 9  | 61  | 182  | 59 | 73  | 746  | 109 | 20  | 691  | 159 | 39  | 307  | 209 | 180 | 307  | 259 | 120 | 943  |
| 10 | 109 | 900  | 60 | 1   | 1068 | 110 | 59  | 201  | 160 | 66  | 639  | 210 | 83  | 527  | 260 | 190 | 378  |
| 11 | 27  | 347  | 61 | 38  | 110  | 111 | 91  | 617  | 161 | 95  | 797  | 211 | 188 | 239  | 261 | 3   | 917  |
| 12 | 1   | 923  | 62 | 197 | 177  | 112 | 140 | 453  | 162 | 12  | 925  | 212 | 184 | 934  | 262 | 111 | 1185 |
| 13 | 135 | 1106 | 63 | 103 | 213  | 113 | 162 | 52   | 163 | 66  | 1068 | 213 | 104 | 213  | 263 | 110 | 720  |
| 14 | 26  | 938  | 64 | 86  | 576  | 114 | 126 | 1047 | 164 | 17  | 753  | 214 | 1   | 955  | 264 | 162 | 710  |
| 15 | 159 | 133  | 65 | 57  | 351  | 115 | 133 | 1179 | 165 | 155 | 606  | 215 | 162 | 427  | 265 | 149 | 1155 |
| 16 | 135 | 535  | 66 | 132 | 1083 | 116 | 151 | 986  | 166 | 158 | 1165 | 216 | 133 | 498  | 266 | 5   | 10   |
| 17 | 50  | 284  | 67 | 120 | 120  | 117 | 127 | 745  | 167 | 108 | 878  | 217 | 198 | 221  | 267 | 64  | 134  |
| 18 | 120 | 1172 | 68 | 42  | 49   | 118 | 77  | 769  | 168 | 191 | 1138 | 218 | 67  | 566  | 268 | 28  | 855  |
| 19 | 38  | 1192 | 69 | 100 | 708  | 119 | 29  | 187  | 169 | 3   | 234  | 219 | 84  | 1074 | 269 | 2   | 1167 |
| 20 | 58  | 1166 | 70 | 70  | 630  | 120 | 37  | 313  | 170 | 135 | 701  | 220 | 144 | 199  | 270 | 148 | 1180 |
| 21 | 137 | 949  | 71 | 26  | 1001 | 121 | 159 | 822  | 171 | 18  | 445  | 221 | 183 | 583  | 271 | 167 | 596  |
| 22 | 106 | 162  | 72 | 45  | 65   | 122 | 119 | 851  | 172 | 150 | 342  | 222 | 31  | 836  | 272 | 163 | 266  |
| 23 | 21  | 154  | 73 | 3   | 518  | 123 | 120 | 614  | 173 | 188 | 1094 | 223 | 102 | 466  | 273 | 123 | 228  |
| 24 | 33  | 125  | 74 | 96  | 381  | 124 | 173 | 951  | 174 | 150 | 1029 | 224 | 150 | 261  | 274 | 127 | 748  |
| 25 | 47  | 376  | 75 | 44  | 755  | 125 | 29  | 266  | 175 | 103 | 120  | 225 | 9   | 635  | 275 | 119 | 758  |
| 26 | 90  | 841  | 76 | 81  | 81   | 126 | 135 | 30   | 176 | 134 | 875  | 226 | 37  | 219  | 276 | 193 | 766  |
| 27 | 18  | 652  | 77 | 170 | 458  | 127 | 149 | 154  | 177 | 124 | 1097 | 227 | 16  | 420  | 277 | 17  | 701  |
| 28 | 153 | 204  | 78 | 75  | 286  | 128 | 10  | 220  | 178 | 114 | 815  | 228 | 142 | 1061 | 278 | 28  | 267  |
| 29 | 41  | 225  | 79 | 146 | 459  | 129 | 72  | 705  | 179 | 33  | 59   | 229 | 151 | 34   | 279 | 32  | 949  |
| 30 | 22  | 1095 | 80 | 88  | 1142 | 130 | 59  | 511  | 180 | 40  | 1060 | 230 | 140 | 932  | 280 | 45  | 1125 |
| 31 | 137 | 1095 | 81 | 142 | 433  | 131 | 200 | 1110 | 181 | 128 | 10   | 231 | 132 | 418  | 281 | 75  | 66   |
| 32 | 20  | 805  | 82 | 46  | 223  | 132 | 140 | 773  | 182 | 187 | 299  | 232 | 12  | 142  | 282 | 17  | 1035 |
| 33 | 173 | 652  | 83 | 152 | 610  | 133 | 52  | 664  | 183 | 99  | 350  | 233 | 160 | 419  | 283 | 45  | 996  |
| 34 | 160 | 546  | 84 | 147 | 509  | 134 | 33  | 758  | 184 | 166 | 95   | 234 | 80  | 39   | 284 | 41  | 775  |
| 35 | 115 | 339  | 85 | 70  | 141  | 135 | 195 | 737  | 185 | 10  | 357  | 235 | 88  | 324  | 285 | 86  | 763  |
| 36 | 36  | 375  | 86 | 32  | 934  | 136 | 90  | 1134 | 186 | 29  | 3    | 236 | 199 | 2    | 286 | 173 | 753  |
| 37 | 172 | 688  | 87 | 54  | 580  | 137 | 78  | 45   | 187 | 117 | 899  | 237 | 107 | 300  | 287 | 9   | 1161 |
| 38 | 13  | 417  | 88 | 151 | 427  | 138 | 75  | 925  | 188 | 88  | 126  | 238 | 105 | 544  | 288 | 96  | 627  |
| 39 | 152 | 1048 | 89 | 173 | 1115 | 139 | 185 | 1187 | 189 | 183 | 934  | 239 | 9   | 266  | 289 | 190 | 402  |
| 40 | 133 | 653  | 90 | 26  | 99   | 140 | 156 | 353  | 190 | 167 | 505  | 240 | 54  | 154  | 290 | 74  | 431  |

| #  | X   | Y    | #   | X   | Y    | #   | X   | Y    | #   | X   | Y    | #   | X   | Y    | #   | X   | Y    |
|----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 41 | 151 | 33   | 91  | 140 | 134  | 141 | 36  | 1004 | 191 | 28  | 273  | 241 | 86  | 946  | 291 | 194 | 220  |
| 42 | 109 | 576  | 92  | 52  | 789  | 142 | 146 | 533  | 192 | 22  | 810  | 242 | 150 | 332  | 292 | 200 | 137  |
| 43 | 133 | 1170 | 93  | 178 | 709  | 143 | 62  | 876  | 193 | 167 | 68   | 243 | 33  | 580  | 293 | 200 | 469  |
| 44 | 71  | 848  | 94  | 10  | 264  | 144 | 15  | 605  | 194 | 71  | 754  | 244 | 66  | 766  | 294 | 185 | 887  |
| 45 | 94  | 513  | 95  | 146 | 1170 | 145 | 114 | 242  | 195 | 32  | 327  | 245 | 95  | 544  | 295 | 195 | 1103 |
| 46 | 154 | 6    | 96  | 62  | 138  | 146 | 56  | 671  | 196 | 156 | 1161 | 246 | 144 | 471  | 296 | 111 | 265  |
| 47 | 177 | 456  | 97  | 30  | 479  | 147 | 172 | 913  | 197 | 200 | 947  | 247 | 39  | 1133 | 297 | 149 | 964  |
| 48 | 57  | 927  | 98  | 164 | 928  | 148 | 52  | 1043 | 198 | 189 | 100  | 248 | 101 | 677  | 298 | 121 | 937  |
| 49 | 138 | 1031 | 99  | 120 | 50   | 149 | 197 | 1140 | 199 | 88  | 600  | 249 | 85  | 43   | 299 | 82  | 1143 |
| 50 | 54  | 676  | 100 | 59  | 611  | 150 | 89  | 55   | 200 | 158 | 776  | 250 | 180 | 190  | 300 | 200 | 943  |

## A-2 Sanayi veri kümesi

56 adet parça

| #  | En | Boy | Adet |
|----|----|-----|------|
| 1  | 50 | 60  | 4    |
| 2  | 70 | 50  | 4    |
| 3  | 40 | 50  | 4    |
| 4  | 40 | 30  | 4    |
| 5  | 40 | 20  | 4    |
| 6  | 50 | 10  | 4    |
| 7  | 70 | 10  | 4    |
| 8  | 30 | 10  | 4    |
| 9  | 40 | 70  | 4    |
| 10 | 90 | 40  | 4    |
| 11 | 70 | 20  | 8    |
| 12 | 80 | 30  | 8    |

## A-3 Diğer veri kümeleri

Diğer veri setleri için ilgili referanslara bakınız.

## KAYNAKLAR

---

- [1] Sedgewick, R., (1992). "Algorithms in C++", Addison-Wesley Professional, 595-606.
- [2] Tooley, F., (1985). "Handbook of Glass Manufacture", Glass Industry, 3rd edition.
- [3] Lodi, A., Martello, S. ve Vigo, D., (1999). "Heuristic and metaheuristic approaches for a class of two-dimensional bin Packing problems", *INFORMS Journal on Computing* 11: 345–357.
- [4] Dyckhoff, H. ve Finke, U., (1992). "Cutting and Packing in Production and Distribution", Physica Verlag, Heidelberg.
- [5] Dowsland, K.A. ve Dowsland, W.B., (1992). "Packing problems", *European Journal of Operational Research* 56:2–14
- [6] Dyckhoff, H., Scheithauer, G. ve Terno, J., (1997). "Cutting and packing (C&P), in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.)", *Annotated Bibliographies in Combinatorial Optimization*, Wiley, Chichester, 393–413.
- [7] Dyckhoff, H., (1990). "Typology of cutting and packing problems.", *European Journal of Operational Research* 44:145-159.
- [8] Hinxman, A.I., (1980). "The Trim Loss and Assortment Problems.", *European Journal of Operational Research*, 5: 8-18.
- [9] Wascher, G., Haußner, H. ve Schumann, H., (2007). "An improved typology of cutting and packing problems", *European Journal of Operational Research* 183:1109–1130
- [10] Garey, M.R. ve Johnson, D.S., (1979). "Computers and Intractability: A Guide to the Theory of Np-Completeness", W. H. Freeman & Co., New York, NY, USA.
- [11] Gilmore, P.C. ve Gomory, R.E., (1961). "A linear programming approach to the cutting stock problem (Part 1)", *Operations Research*, 9: 849-859.
- [12] Gilmore, P.C., Gomory, R.E., (1963). "A linear programming approach to the cutting stock problem (Part 2)", *Operations Research*, 11:863-888.
- [13] Land, A.H. ve Doig, A.G., (1960). "An automatic method of solving discrete programming problems", *Econometrica* 28 (3):497-520.

- [14] Sedgewick, R., (1992). "Algorithms in C++", Addison-Wesley Professional; 2nd Edition edition, 595-606.
- [15] Jakobs, S., (1996). "On genetic algorithms for the packing of polygons.", European Journal of Operations Research 88:165-181.
- [16] Bonabeau, E., Dorigo, M. ve Theraulaz, G., (1999). "Swarm Intelligence: From Natural to Artificial Systems". New York, NY: Oxford University Press.
- [17] Tereshko, V., (2000). "Reaction-diffusion model of a honeybee colony's foraging behaviour, M. Schoenauer, et al, Eds., Parallel Problem Solving from Nature VI", Lecture Notes in Computer Science, 1917: 807-816.
- [18] Johnson, D.S., Demers, A., Ullman, J. D., Garey, M.R. ve Graham, R.L., (1974). "Worst-Case Performance Bounds For Simple One-Dimensional Packing Algorithms", Siam J. Comput. 3: 299-326.
- [19] Korf, R.E., (2002). "A New Algorithm for Optimal Bin Packing", AAAI-02 Proceedings, 731-736.
- [20] Holland, J.H., (1975). "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, Mi?
- [21] Goldberg, D.E., (1989). "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley Publishing Company, Inc.
- [22] Karaboga D. ve Basturk B., (2007). "A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, 39(3): 459-471.
- [23] Deb, K., Pratap, A., Agarwal, S. ve Meyarivan, T., (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II" IEEE Trans Evol Comput, 6(2): 182–197
- [24] Jakobs, S., (1996). "On genetic algorithms for the packing of polygons.", European Journal of Operations Research 88: 165-181,.
- [25] Burke, E., Hellier, R., Kendall, G. ve Whitwell, G., (2006). "A New Bottom-Left-Fill Heuristic Algorithm for the Two-Dimensional Irregular Packing Problem", Operations Research 54 (3): 587–601,
- [26] Chazelle, B., (1983). "The Bottom-Left Bin-Packing Heuristic:An Efficient Implementation", IEEE Transactions On Computers,32(8):697-707.
- [27] Baker, B.S., (1985). "A New Proof for the First-Fit Decreasing Bin Packing Algorithm," Journal of Algorithms, 6:49-70.
- [28] Korf, R.E., (2002). "A New Algorithm for Optimal Bin Packing", AAAI-02 Proceedings, 731-736.
- [29] Johnson D.S., Demers A., Ullman J. D., Garey M.R. ve Graham R.L., (1974). "Worst-Case Performance Bounds For Simple One-Dimensional Packing Algorithms", Siam J. Comput. 3(4):299-325.

- [30] Gürbüz, M.Z., Akyokuş, S., Emiroğlu, İ. ve Güran, A., (2009). "An efficient Algorithm for 3D Rectangular Box Packing", *Applied Automatic Systems (AAS)*. 131-134
- [31] Fogel, D. B., (1995). "Evolutionary Computation: Toward a New Philosophy of Machine Intelligence", IEEE Press, New York.
- [32] Holland, J.H., (1962). "Outline for a logical theory of adaptive systems," *J.Assoc. Comput. Mach.*, 3: 297–314.
- [33] Holland, J.H., (1975). "Adaptation in Natural and Artificial Systems". Ann Arbor, MI: Univ. of Michigan Press.
- [34] Holland, J.H. ve Reitman, J.S., (1978). "Cognitive systems based on adaptive algorithms," in *Pattern-Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic.
- [35] De Jong, K.A., (1975). "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Diss. Abstr. Int. 36(10), 5140B, University Microfilms no. 76-9381.
- [36] Goldberg, D.E., (1985). "Genetic algorithms and rule learning in dynamic system control," in *Proc. 1st Int. Conf. on Genetic Algorithms and Their Applications*. Hillsdale, NJ: Lawrence Erlbaum, 8–15.
- [37] Goldberg, D.E., (1989). "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley Publishing Company, Inc
- [38] Ono T. ve Watanabe G., (1997). "Genetic Algorithms for Optimal Cutting", *Evolutionary Algorithms in Engineering Applications*, 515-530
- [39] Baker, J.E., (1985). "Adaptive selection methods for genetic algorithms. In J. J. Grefenstette, editor.", *Proceedings of an International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Hillsdale, NJ, 101-111.
- [40] Baker, J.E., (1987). "Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette, editor." *Proceedings of the 2nd International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Hillsdale, NJ, 14-21
- [41] Alam, M.S., Islam, M.M., Yao, X. ve Murase, K., (2011). "Recurring two-stage evolutionary programming: a novel approach for numerical optimization", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 41:1352-1365.
- [42] Onal, T. ve Beldek, U., (2013). "Kromozom Seçim Metriğinin Yinelemeli İki-Aşamalı Evrimsel Programlamada Performansa Katkısı", *Çankaya University Journal of Science and Engineering* Volume 10(1): 79-90
- [43] Miller, B.L. ve Goldberg, D.E., (1995). "Genetic Algorithms, Tournament Selection, and the Effects of Noise", *Complex Systems* 9:193—212
- [44] Goldberg, D.E., (1989). "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, Reading Massachusetts.

- [45] Tereshko, V., (2000). "Reaction-Diffusion Model of a Honeybee Colony's Foraging Behaviour", PPSN VI Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, 807-816
- [46] Hamdan, K., How Do Bees Make Honey, [https://www.countryrubes.com/images/How\\_do\\_bees\\_make\\_honey\\_update\\_9\\_09.pdf](https://www.countryrubes.com/images/How_do_bees_make_honey_update_9_09.pdf), 21 Haziran 2015.
- [47] Deb, K. (2001). "Multi-objective optimization using evolutionary algorithms (1st ed.)", Wiltshire: John Wiley & Sons, Ltd.
- [48] Konak, A., Coit, D. W., ve Smith, A. E., (2006). "Multi-objective optimization using genetic algorithms: a tutorial. Reliability Engineering & System Safety", 91(9):992-1007.
- [49] Deb, K., (2008). "Introduction to Evolutionary Multiobjective Optimization. In J. Branke, & e. al. (Eds.)", Multiobjective Optimization 1:59-96
- [50] Marler, R., ve Arora, J., (2004). "Survey of multi-objective optimization methods for engineering", Structural and Multidisciplinary Optimization, 26(6): 369-395.
- [51] Zitzler, E. ve Thiele, L., (1999). "An evolutionary algorithm for multiobjective optimization: The strength Pareto approach.", Technical report 43, Computer engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.
- [52] Zitzler, E., Laumanns, M., ve Thiele, L., (2001). "SPEA2: Improving Strength Pareto Evolutionary Algorithm.", Zurich: TIK Swiss Federal Institute of Technology Zurich. TIK-Report 103
- [53] Knowles, J. D. ve Corne, D.W., (1999). "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization.", In Congress on Evolutionary Computation (CEC 99), Volume 1: 98 – 105. IEEE Press.
- [54] Hopper E. ve Turton B. C. H., (1999). "A genetic algorithms for a 2D industrial packing problem", Computers in Engineering 37: 375-378.
- [55] Hopper E. ve Turton B. C. H., (2000). "An Empirical Investigation of Meta-heuristic and Heuristic Algorithms for a 2D Packing Problem.", European Journal of Operational Research 128(1): 34-57.
- [56] Zhang, D., Wei, L., Leung, S.C.H., ve Chen, Q., (2013). "A Binary Search Heuristic Algorithm Based on Randomized Local Search for the Rectangular Strip-Packing Problem". INFORMS J. on Computing 25, 2: 332-345
- [57] Valenzuela, C. L. ve Wang, P. Y., (2001). "Data set generation for rectangular placement problems.", European Journal of Operational Research, 134(2):150-163.

## ÖZGEÇMİŞ

---

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Mustafa Zahid GÜRBÜZ  
**Doğum Tarihi ve Yeri** : 05/08/1982 Kahramanmaraş  
**Yabancı Dili** : İngilizce  
**E-posta** : mzahidgurbuz@hotmail.com

### ÖĞRENİM DURUMU

| Derece        | Alan                   | Okul/Üniversite            | Mezuniyet Yılı |
|---------------|------------------------|----------------------------|----------------|
| Yüksek Lisans | Matematik Mühendisliği | Yıldız Teknik Üniversitesi | 2007           |
| Lisans        | Matematik Mühendisliği | Yıldız Teknik Üniversitesi | 2005           |
| Lise          | Sayısal                | Özel Tercüman Lisesi       | 2000           |

### İŞ TECRÜBESİ

| Yıl  | Firma/Kurum        | Görevi              |
|------|--------------------|---------------------|
| 2014 | Doğuş Üniversitesi | Öğretim Görevlisi   |
| 2006 | Doğuş Üniversitesi | Araştırma Görevlisi |
| 2005 | TET Yazılım        | Yazılım Uzmanı      |
| 2003 | ISD Bilişim        | Yazılımcı           |

## **YAYINLARI**

### **Makale**

1. Aysun Güran, Nilgün Güler Bayazıt, M.Zahid Gürbüz "Efficient feature integration with Wikipedia based semantic feature extraction for Turkish text summarization", Turkish Journal of Electrical Engineering and Computer Science (SCI-E), 2013, Vol. 21 Issue 5, p1411-1425. 15p

### **Bildiri**

1. M.Zahid Gürbüz, İbrahim Emiroğlu, Kıvanç Onan, "A Multi-Objective Optimization Approach To Glass Cutting Problem", CIE44 & IMSS'14 Proceedings, 14-16 October 2014, Istanbul / Turkey, Pages: 1830-1835
2. M.Zahid Gürbüz, İbrahim Emiroğlu, Selim Akyokuş, "An improved 2D glass cutting solution with genetic algorithms", INISTA 2011.
3. Dilara Torunoğlu, Erhan Çakırman, Murat Can Ganiz, Selim Akyokuş, M.Zahid Gürbüz "Analysis of preprocessing methods on classification of Turkish texts ", INISTA 2011.
4. M.Zahid Gürbüz, Selim Akyokuş, İbrahim Emiroğlu, Aysun Güran, "An efficient Algorithm for 3D Rectangular Box Packing", AAS 2009.
5. Aysun Güran, Selim Akyokuş, Nilgün Güler Beyazıt, M.Zahid Gürbüz, "Turkish Text Categorization Using N-Gram Words", INISTA 2009, International Symposium on INnovations in Intelligent SysTems and Applications.

### **Proje**

1. AB Projesi Extreme Future Factory projesi (2013)
2. Sanayi Bakanlığı Teknogirişim Sermayesi Desteği Projesi (2013)
3. BAPK Projesi YTÜ Bap Proje No: 2014-07-03-DOP03 (2014)