

**BDT YAZILIMLARINA AİT VERİLERİN, PROFİL  
DELME MAKİNESİ KONTROL PROGRAMINDA  
KULLANILMAK ÜZERE DÖNÜŐTÜRÜLMESİ VE  
İŐLEM SÜRESİNİN OPTİMİZASYONU**

**2012  
YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĐİ**

**Mustafa YILDIZ**

**BDT YAZILIMLARINA AİT VERİLERİN, PROFİL DELME MAKİNESİ  
KONTROL PROGRAMINDA KULLANILMAK ÜZERE  
DÖNÜŞTÜRÜLMESİ VE İŞLEM SÜRESİNİN OPTİMİZASYONU**

**Mustafa YILDIZ**

**Karabük Üniversitesi  
Fen bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalında  
Yüksek Lisans Tezi  
Olarak Hazırlanmıştır**

**KARABÜK  
Eylül 2012**

Mustafa YILDIZ tarafından hazırlanan “BDT YAZILIMLARINA AİT VERİLERİN, PROFİL DELME MAKİNESİ KONTROL PROGRAMINDA KULLANILMAK ÜZERE DÖNÜŞTÜRÜLMESİ VE İŞLEM SÜRESİNİN OPTİMİZASYONU” başlıklı bu tezin Yüksek Lisans tezi olarak uygunluğunu onaylarım.

Yrd. Doç. Dr. İlhami Muharrem ORAK  
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı



Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 07/11/2012

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan: Yrd.Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ)



Üye : Yrd. Doç. Dr. İbrahim ÇAYIROĞLU (KBÜ)

Üye : Yrd. Doç. Dr. İlhami Muharrem ORAK (KBÜ)



.../.../2012

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu bu tez ile Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN  
Fen Bilimleri Enstitüsü Müdürü



*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Mustafa YILDIZ

## **ÖZET**

**Yüksek Lisans Tezi**

### **BDT YAZILIMLARINA AİT VERİLERİN, PROFİL DELME MAKİNESİ KONTROL PROGRAMINDA KULLANILMAK ÜZERE DÖNÜŞTÜRÜLMESİ VE İŞLEM SÜRESİNİN OPTİMİZASYONU**

**Mustafa YILDIZ**

**Karabük Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:**

**Yrd. Doç. Dr. İlhami Muharrem ORAK**

**Eylül 2012, 104 sayfa**

Endüstriyel uygulamalarda CAD uygulamaları kolay tanımlama yapılabilmesi ve hassas işleme yapılması açısından önemli bir alanı içermektedir. Standart CAD programları kullanımı ile sağlanan bu avantajlar özellikle veriye dayalı uygulama geliştirme açısından yeterli bir altyapı sağlayamamaktadır. CAD dosyalarının uygulamada kullanılacak verilerinin alınarak veritabanına entegre edilmesi ile kontrol sistemine gerekli parametrelerin sağlanması ve üretim sonucu elde edilen parametrelerinin analizine imkan sağlanacaktır.

Bu çalışmada örnek bir uygulama alanı olarak Profil Delme ve Kesme Makinesinin CAD çizim programlarından elde edilen verilerle entegrasyonu değerlendirilmiştir. Geliştirilen uygulama ile CAD çizim dosyalarında belirtilen çelik profillerin tipleri, boyları, delinme noktaları ve bu noktaların bulunduğu yüzeyler tespit edilerek bu

verileri veritabanında saklanması sağlanmıştır. Bu verilere dayalı olarak optimum delme sürelerini hesaplanmaktadır. Optimum delme süresi; farklı çaplarda delikleri delmek üzere matkap uçlarının değiştirilme süresi, delme uçlarının çelik profil üzerinde hareket etme süresi gibi parametreler ile belirlenmektedir. Uygulamada üç eksenin beraberce optimize edilmesi sağlanmıştır. Tasarlanan arayüz ile çizimden alınan verilerin izlenmesi sağlanmaktadır. Java programlama dilinde hazırlanan uygulama ile gerekli olan bilgilerin “tarama” yöntemi kullanılarak ayıklanması ve elde edilen verilerin MySQL veritabanı üzerinde saklanması sağlanmıştır.

**Anahtar Sözcükler** : Çelik profil, delme, CAD DXF veri analizi, kontrol sistemi, optimizasyon.

**Bilim Kodu** : 902.1.014

## **ABSTRACT**

**M.Sc. Thesis**

### **CONVERTING CAD DATA TO BE USED USE IN PROFILE DRILLING MACHINE AND OPTIMIZATION OF DRILLING TIME**

**Mustafa YILDIZ**

**Karabük University**

**Graduate School of Natural and Applied Sciences**

**Department of Computer Engineering**

**Thesis Advisor:**

**Assist. Prof. Dr. İlhami Muharrem ORAK**

**September 2012, 104 pages**

CAD applications provide easy definition facilities as well as precise processing features in industrial applications. These features can be achieved with Standard CAD programs. Nevertheless, for introducing applications based on data, these type of programs will not be sufficient. Integrating the all necessary information derived from CAD drawings with a database application will introduce a system to support control system with necessary parameters and gather processing and machine related parameters for further analysis.

In this study Profile Drilling and Cutting Machine has been chosen a sample application to investigate the related data in CAD drawings to be used and developing a database system. With the application introduced in this study profile types, dimensions, positions of drills and the surfaces where drills are to be processed

are identified and stored in database. These information is used by control system for positioning of servo motors. Since the natural structure of the drilling process, optimization will be required to reduce overall drilling time. This optimization will be done taking into consideration replacement time of drilling tools and distances between the drills. The user interface provides visualization of data derived from the drawings. Optimization is done for three axis together. The application developed by using Java language derives all necessary data from drawings by scanning the file. These data are then stored in MySql database.

**Key Words** : Steel profile, drilling, CAD drawing data analysis, control system, optimization.

**Science Code** : 902.1.014



## TEŐEKKÜR

Bu tez alıőmasının planlama, araőtırma ve tamamlanma aőamalarında, her tŸrlŸ desteęini esirgemeyen, sayın hocam Yrd. Do. Dr. İlhami Muharrem ORAK'a teőekkŸrlerimi sunarım.

Ayrıca manevi olarak beni her konuda destekleyen ve alıőmalarımı yŸrŸtebilmem iin imkan sunan sevgili aileme tŸm kalbimle teőekkŸr ederim.

## İÇİNDEKİLER

	<u>Sayfa</u>
KABUL .....	ii
ÖZET .....	iv
ABSTRACT .....	vi
TEŞEKKÜR .....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLER DİZİNİ .....	xi
ÇİZELGELER DİZİNİ .....	xiv
SİMGELER VE KISALTMALAR DİZİNİ .....	xv
BÖLÜM 1 .....	1
GİRİŞ .....	1
BÖLÜM 2 .....	5
MEVCUT SİSTEMİN TANITIMI .....	5
BÖLÜM 3 .....	8
ÇİZİM DOSYASININ TANINMASI .....	8
3.1. DXF DOSYA YAPISI .....	10
3.2. DOSYANIN OKUNMASI .....	14
3.2.1. Çizgi Bilgilerinin İncelenmesi ve Veritabanına Aktarılması İşlemi .....	15
3.2.2. Daire Bilgilerinin İncelenmesi ve Veritabanına Aktarılması İşlemi .....	18
3.3. ELDE EDİLEN BİLGİLERİN ANLAMLANDIRILMASI .....	19
3.3.1. Çizgilerin Anlamlandırılması İşlemi .....	20
3.3.2. Dairelerin Anlamlandırılması İşlemi .....	23
BÖLÜM 4 .....	25
OPTİMİZASYON .....	25
4.1. PROFİL DELME OPTİMİZASYONU YAKLAŞIMI .....	26
4.2. KULLANILABİLECEK MEVCUT OPTİMİZASYON YÖNTEMLERİ .....	29

	<b><u>Sayfa</u></b>
4.2.1. Yöntem 1 – Yakın Komşu Algoritması .....	31
4.2.2. Yöntem 2 – 2-Opt Exchange Algoritması.....	33
4.2.3. Yöntem 3 – Kaydırma Algoritması.....	37
4.3. ÜÇ EKSENLİ OPTİMİZASYON UYGULAMASI.....	39
4.3.1. Maliyetin Hesaplanması.....	41
4.3.2. Optimizasyon İşleminin Gerçekleştirilmesi.....	45
BÖLÜM 5 .....	64
GELİŞTİRİLEN YAZILIMIN TANITIMI.....	64
5.1. PROGRAM HAKKINDA GENEL BİLGİLER .....	64
5.2. PROGRAMIN ÇALIŞTIRILMASI.....	69
BÖLÜM 6 .....	70
SONUÇLAR VE ÖNERİLER.....	70
KAYNAKLAR .....	75
EK AÇIKLAMALAR A. ÖRNEK VERİTABANI TABLO ÇİZELGELERİ.....	78
EK AÇIKLAMALAR B. ÖRNEK ÇİZİM DOSYASI İÇERİKLERİ .....	85
EK AÇIKLAMALAR C. İNCELEME İŞLEMİ SONUCU ELDE EDİLEN DEĞİŞİM GRAFİKLERİ .....	99
ÖZGEÇMİŞ .....	104

## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 2.1.	Mevcut durumda çalışan Profil Delme ve Kesme Makinesi sistemi. ....	5
Şekil 2.2.	Uygulamada kullanılan Profil Delme ve Kesme Makinesi görüntüsü.....	6
Şekil 2.3.	Profil Delme ve Kesme Makinesi yüzey delme işlemi.....	7
Şekil 3.1.	Örnek çizim parametreleri; a) çizgi ve b) daire. ....	10
Şekil 3.2.	Basit bir a) çizgi ve b) daire çizim nesnesine ait Dxf dosya içeriği.....	12
Şekil 3.3.	Çalışma kapsamında incelenmesi planlanan örnek çizim dosyası.....	13
Şekil 3.4.	Dosya okuma işlemine ait genel akış şeması. ....	15
Şekil 3.5.	LINE çizim nesnesine ait bilgilerin okunmasını sağlayan akış şeması. ....	16
Şekil 3.6.	Çizgi nesnelерinin tek tip haline getirilmesi. ....	17
Şekil 3.7.	CIRCLE çizim nesnesine ait bilgilerin okunmasını sağlayan akış şeması. ....	18
Şekil 3.8.	Örnek olarak a) bir çizgiye ait temsili gösterim ve b) kapalı kutu oluşturduğu düşünülen ve değerlendirilebilecek iki ayrı dizilim.....	20
Şekil 3.9.	Geliştirilen uygulamada kullanılan inceleme yöntemi.....	21
Şekil 3.10.	Çizgilerin anlamlandırılması işlemine ait akış diyagramı.....	22
Şekil 3.11.	Dairelerin anlamlandırılması işlemi akış diyagramı. ....	24
Şekil 4.1.	Örnek 4 delikli profil çizimi.....	28
Şekil 4.2.	2-Opt Exchange algoritması için örnek profil görünümü ve delik dizilimi. ....	33
Şekil 4.3.	Kabul edilen başlangıç dizilimini gösteren delme sırası.....	34
Şekil 4.4.	2-Opt Exchange algoritması birinci aşama sonucundaki yeni delme sırası. ....	35
Şekil 4.5.	2-Opt Exchange algoritması ikinci aşama sonucundaki yeni delme sırası. ....	36
Şekil 4.6.	2-Opt Exchange algoritması ile elde edilen optimum dizilim. ....	36
Şekil 4.7.	Kaydırma Algoritması birinci aşama sonucundaki oluşan dizilim. ....	37
Şekil 4.8.	Kaydırma algoritması ikinci aşama sonucunda oluşan dizilim.....	38
Şekil 4.9.	Genel haliyle gerçekleştirilecek olan optimizasyon işlemi basamakları.....	40
Şekil 4.10.	Matkap sınıfı tarafından gerçekleştirilen işlemlere ait akış diyagramı. .	42
Şekil 4.11.	Matkap sınıfı delikMaliyeti() metodu akış diyagramı. ....	43

Şekil 4.12. Optimizasyon işleminde farklı dizilim ve toplam maliyetlerin elde edilmesi. ....	46
Şekil 4.13. 2-Opt Exchange Algoritması Akış Şeması. ....	49
Şekil 4.14. n-Opt Exchange Algoritması Akış Şeması. ....	50
Şekil 4.15. Kaydırma Algoritması Akış Şeması. ....	51
Şekil 4.16. Örnek hesaplamada kullanılacak 3 eksenli çizim. ....	52
Şekil 4.17. Şekil Ek B.9'da verilen 50 Delikli çizim dosyasının üç değişik algoritma ile incelenmesiyle oluşan maliyet akışı. ....	55
Şekil 4.18. Şekil Ek B.9'da verilen 50 delikli çizim dosyasının ikili algoritma kombinasyonları ile incelenmesiyle oluşan akış maliyeti. ....	56
Şekil 4.19. Şekil Ek B.9'da verilen 50 delikli çizim dosyasının üçlü algoritma kombinasyonları ile incelenmesiyle oluşan akış maliyeti. ....	57
Şekil 4.20. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından gerçekleştirilen değişiklik deneme sayıları grafiği. ....	59
Şekil 4.21. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından gerçekleştirilen değişiklik sayıları grafiği. ....	60
Şekil 4.22. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından elde edilen toplam delme zamanı maliyeti grafiği. ....	61
Şekil 5.1. Geliştirilen yazılıma ait ekran görüntüsü. ....	65
Şekil 5.2. Profil Önizleme penceresi. ....	66
Şekil 5.3. Profil Listele penceresi. ....	67
Şekil 5.4. Program menü çubuğu ve içerikleri. ....	68
Şekil 5.5. Ayarlar penceresi. ....	68
Şekil EK B.1. UNP200_8delik.dxf çizim dosyası içeriği. ....	86
Şekil EK B.2. Cizim1_ayniboy_9delik.dxf çizim dosyası içeriği. ....	87
Şekil EK B.3. Cizim2_3farkliboy_9delik.dxf çizim dosyası içeriği. ....	88
Şekil EK B.4. Cizim3_3farkliboy_9delik.dxf çizim dosyası içeriği. ....	89
Şekil EK B.5. 200x100x10DIKDORGEN_10delik.dxf çizim dosyası içeriği. ....	90
Şekil EK B.6. 200x200x10KUTU_12delik.dxf çizim dosyası içeriği. ....	91
Şekil EK B.7. PDKCalisma_13delik.dxf çizim dosyası içeriği. ....	92
Şekil EK B.8. HE200_14delik.dxf çizim dosyası içeriği. ....	93
Şekil EK B.9. PDKCalisma_13delik.dxf çizim dosyasından üretilen 50 delikli çizim. ....	94
Şekil EK B.10. 70Delik.dxf çizim dosyası içeriği. ....	95
Şekil EK B.11. 90Delik.dxf çizim dosyası içeriği. ....	96

	<b><u>Sayfa</u></b>
Şekil EK B.12. 105Delik.dxf çizim dosyası içeriği. ....	97
Şekil EK B.13. 120Delik.dxf çizim dosyası içeriği. ....	98
Şekil EK C.1. 70Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon. ....	100
Şekil EK C.2. 90Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon. ....	101
Şekil EK C.3. 105Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon. ....	102
Şekil EK C.4. 120Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon. ....	103

## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 3.1.	Dxf dosyası çizgi grup kodları ve anlamları.....	11
Çizelge 3.2.	Dxf dosyası daire grup kodları ve anlamları. ....	11
Çizelge 4.1.	Örnek çizim için farklı dizilimdeki toplam maliyet hesaplamaları. ....	29
Çizelge 4.2.	Yakın Komşu algoritması için delikler arası maliyetleri gösterir tablo.....	31
Çizelge 4.3.	Yakın Komşu algoritması birinci aşama. ....	32
Çizelge 4.4.	Yakın Komşu algoritması ikinci aşama.....	32
Çizelge 4.5.	Yakın Komşu algoritması üçüncü aşama. ....	33
Çizelge 4.6.	Delme işlemi dizilimini tutan dizi değişkeni içeriği.....	34
Çizelge 4.7.	Optimizasyon yöntemleri öncelik sıralamasının sonuçlara etkisi. ....	39
Çizelge 4.8.	Örnek çizime ait başlangıç dizilimini tutan optimum[] dizi içeriği. ..	52
Çizelge 4.9.	Örnek çizime ait delik bilgileri.....	53
Çizelge 4.10.	Kullanılan kombinasyonların sonuçlarına göre ağırlık puanları. ....	62
Çizelge 4.11.	Değişik ortalama yöntemleri ile elde edilen sonuçlar. ....	63
Çizelge 6.1.	Değişik durumlara ait deney işlemi sonuçları. ....	71
Çizelge 6.2.	Tercih edilen algoritma kombinasyonlarına ait sonuçlar. ....	73
Çizelge EK A.1.	Örnek çizim dosyasına ait LINE veritabanı tablosu içeriği.....	79
Çizelge EK A.2.	Örnek çizim dosyasına ait CIRCLE veritabanı tablosu içeriği.....	81
Çizelge EK A.3.	Örnek çizim dosyasına ait PROFIL veritabanı tablosu içeriği.....	82
Çizelge EK A.4.	Örnek çizim dosyasına ait DELIK veritabanı tablosu içeriği.....	83
Çizelge EK A.5.	Örnek çizim dosyasına ait OPTIMIZED veritabanı tablosu içeriği.....	84

## SİMGELER VE KISALTMALAR DİZİNİ

### KISALTMALAR

BDT	: Bilgisayar Destekli Tasarım
CAD/CAM	: Computer Aided Design / Computer Aided Manufacturing
Dwg	: Dosya formatı uzantısı (Drawing)
Dxf	: Dosya formatı uzantısı (Drawing Exchange Format)
Par	: Dosya formatı uzantısı (Part)
PLC	: Programmable Logic Control
CNC	: Computer Numerical Control
NC	: Numerical Control
Mhs	: Matkap hareket süresi
Mds	: Matkap ucu delme süresi
Muds	: Matkap ucu değişimi süresi
1,2	: 1 numaralı delik ile 2 numaralı delik arasındaki mesafe



## BÖLÜM 1

### GİRİŞ

Her geçen yıl teknoloji hızla gelişmekte ve teknolojiye olan ihtiyaç yine paralel olarak artmaktadır. Günlük yaşantımızda yer almadığı bir alan neredeyse yok denecek kadar azdır. Aynı şekilde ihtiyaçlarımızı karşıladığımız her türlü araç-gerecin üretim aşamasında da yine teknoloji kullanımı her geçen yıl artmakta ve zorunlu olmaktadır. Üretimde kullanılan her yeni teknolojik araç; üretim miktarı ve üretim kalitesini arttırırken zaman ve hata payını minimuma indirmektedir.

Başka bir açıdan değerlendirilecek olursa, günümüzde kullanımı artan otomasyon araçları, aynı zamanda insan gücünün yerini alırken istihdamın azalmasına da sebep olabilmektedir. Her ne kadar böyle bir görüntü oluşsa da insan faktörü vazgeçilmez bir etkidir. Kalite kontrol aşamasında insan hala büyük bir rol oynar.

Endüstriyel sistemlerde üretim kalitesi ve üretimdeki verimlilik vazgeçilemez faktörlerden ikisidir. Genel olarak amaç, olabilecek en kısa sürede kaliteli ürün elde edebilmektir. Bu amaç uğrunda yeni teknolojik araçlar geliştiren insanlar, her ikisini bir arada sunabilmek adına değişik tasarımlarda bulunmaktadır.

Endüstride imalatta işleme yapan makinelerde Bilgisayar Destekli (CAD/CAM) olmaları oldukça büyük avantajlar sağlamaktadır. Sayısal verilerle çalışmanın getirdiği hassasiyet ve esneklik bu tür ürünlerin kullanımının yaygınlaşmasını sağlamaktadır.

Sumin Wang ve arkadaşları, 1996 yılında yaptıkları çalışmada, CAD/CAM sistemleri ve eş zamanlı mühendislik felsefesi ile ilgili olarak, Bilgisayarla tümleşik imalat sistemlerinde malzeme ve bilgi akışlarının entegrasyonunun önemini vurgulamışlardır [1]. Ayrıca Bilgisayar tabanlı CAD/CAM sisteminin entegrasyonu

için mevcut yazılım, donanım ve ekipmana dayalı seri bir haberleşme arabirimi yöntemi geliştirmişlerdir.

2003 yılında Ismail, N. ve arkadaşları, karmaşık mekanik parçaların modellenmesi ve işlenmesi alanında CAD/CAM sistemlerinin kullanılması üzerinde çalışmışlardır [2].

Johannes, M. S. ve arkadaşları 2006 yılında özel bir atomsal kuvvet mikroskobu kullanarak CAD/CAM tabanlı otomatik nanolitografi sistemi hazırlamışlardır [3]. Bu teknik, sanal bir tasarım ortamında istenen nano desenleri oluşturmak için CAD/CAM yazılımlarını kullanmaktadır.

Oshima, Y. ve Morishita, T. 2011 yılında, mühendislik öğrencileri için 3B (3 Boyut) CAD/CAM ve stereo görme teknolojilerine dayalı bir teknik eğitim programı önermişlerdir [4]. Bu programda, öğrencilerin bir CAD/CAM devre kartı ve görüntü bilgileri işleme sistemi kullanmaya teşvik edilmesini amaçlamışlardır.

2012 yılında ise Nagata, F. ve arkadaşları, bir RV1A endüstriyel robotu için, robot dili kullanmayan CAM sistemi üzerinde çalışmışlardır. Bahsedilen CAM sistemi, herhangi bir robot dili olmaksızın, CAD/CAM ve endüstriyel robotlar arasında esnek bir arayüz olarak çalışır [5].

CAD/CAM sistemleri genelde standart tasarımlarla, sonuca ulaşması istenilen işin daha hızlı ve kaliteli bir şekilde gerçekleştirilmesini sağlamaktadır. Günümüzde modelleme, veri madenciliği alanlarında yapılan çalışmalar sistemlerin veritabanları ile entegre olmasını gerekli kılmaktadır. Ancak bu yapılarla sistemlerin verimliliklerinin anlaşılması, performans iyileştirmelerinin ve tasarım değişikliklerinin gerçekleştirilebilmesi mümkün olabilecektir.

Ayrıca inşaat sektörü de çok gelişmiş, birbirinden farklı ölçek ve tasarımlarda binalar yapılabilmektedir. Beton binaların yanında özellikle depreme dayanıklı olmaları sebebiyle çelik konstrüksiyon binalar ülkemizde yaygınlaşma potansiyeline sahiptir. Bu çalışma; ofis binalar, iş merkezleri, alışveriş merkezleri ve gökdelenler gibi değişik birçok binanın yapımında kullanılan çelik profillerin daha hızlı ve verimli bir

şekilde projeye uygun olarak hazırlanmasına yardımcı olmak amacıyla gerçekleştirilmiştir. Bu projeye konu olan örnek Profil Kesme ve Delme Makinesi, teknik resimlerde bulunan malzeme boy ve delik koordinat bilgilerinin bir personel yardımıyla manuel olarak sisteme aktarılması şeklinde çalışmaktadır. Bu durum, üretilen malzemede muhtemel hatalara ve üretim süresinin uzaması gibi etkilere yol açmaktadır.

Bu çalışmada, birbirinden farklı fakat bir ekseninde kısmen bağımlı olan üç yüzeye sahip profillerin delme işleminin, minimum zamanda tamamlanması için delme sırası optimizasyonu gerçekleştirilecektir. Bu alanda 2005 yılında Simon Sigl ve Helmut A. Mayer, ağaç ve plastik materyal üretiminde tek yüzey kullanarak delme sırası optimizasyonunu 2-Opt Exchange algoritması kullanarak gerçekleştirmişlerdir [6]. Çalışmalarında neden Genetik Algoritmalar yerine bu algoritmayı tercih ettiklerini şu şekilde ifade etmişlerdir;

*“Temel olarak problem, rota değişiminin geçersiz bir rotaya dönüşebilir olmasıdır. Örneğin, 1→2→3 rotası, rastgele değiştirildiğinde ikinci durum olarak 1→3→3 olarak problemin kısıtlamalarını ihlale neden olabilir. Bu sebepten dolayı, permütasyon gibi her zaman geçerli yolları üreten bir mutasyon operatörü kullanmak istiyoruz.”*

2006 yılında Guang Yu Zhu çalışmasında delme işlemi sırası optimizasyonu için parçacık sürü algoritmaları kullanmıştır [7].

Yine parçacık sürü algoritması kullanarak, 2010 yılında Asrul Adam ve arkadaşları, baskı devre üretiminde robotik delme işlemi için bir delme sırası optimizasyonu gerçekleştirmişlerdir [8].

2012 yılında C.Y. Tsai, baskı devrelerde bulunan deliklerin minimum sürede delinmesini sağlamak amacıyla genetik algoritma kullanarak optimum delme dizilimini elde etmeye çalışmıştır [9].

Geliştirilen yazılım, teknik resimlerde bulunan bilgileri okuyup anlamlandırmakta, bu bilgileri üretimin minimum hata ve minimum sürede tamamlanabilmesi amacıyla

delme sırası optimizasyon işlemini gerçekleştirmekte ve bir veritabanı tablosu aracılığıyla Profil Delme ve Kesme Makinesinde işlenmek üzere sisteme aktarmaktadır. Yazılımın sağladığı avantajlar şöyle sıralanabilir;

- Kullanılacak veriler doğrudan teknik resimlerden elde edilerek hatalı bilgi girişleri ortadan kaldırılmaktadır.
- Bilgiyi doğrudan teknik resimlerden hızlıca elde ederek veritabanında saklaması, üretim aşamasının ardından kalite kontrol amacıyla da kullanılabilmesini sağlar.
- Manuel yöntemle girilen delme sırası, gerekenden fazla matkap ucu değişimi ve yola neden olabilmektedir. Yapılan optimizasyon sayesinde değişik dizilimlerin oluşturduğu zaman maliyetleri hesaplanarak en uygun dizilime karar verilmekte ve gereksiz hareketler minimuma indirgenmektedir.
- Ayrıca bu çalışmada, birbirine birer ekseninde bağımlı, birer ekseninde bağımsız olan üç farklı yüzeye ait üç delme kafası için bir optimizasyon uygulaması geliştirilmiştir. Günümüzde kullanılan optimizasyon yöntemlerinden faydalanılarak, farklı bir bakış açısı ile üç farklı yüzeye aynı anda uygulanan bir yöntem geliştirilmiştir.
- Enerji ve zaman kayıplarını minimuma indirgeyerek verimliliği arttırmayı amaçlamaktadır.

## BÖLÜM 2

### MEVCUT SİSTEMİN TANITIMI

Otomasyon, günümüzde hemen hemen her sektörde ihtiyaç duyulan, işlemlerin otomatikleştirilmesini amaçlayan sistemlerdir. Karaçor, M. ve Keleş, K., bu konuda şöyle demektedir [10];

*“Esnek bir üretim sistemini gerçekleştirebileceğiniz, sahada nelerin olup bittiğini bir ekrandan (PC veya HMI) takip edebileceğiniz ve bu verileri kaydedebileceğiniz, stoktaki ürünlerin veya kaynakların planlamasını yapabileceğiniz komple bir mekanizmadır.”*

Uygulamada kullanılan mevcut sistemi Şekil 2.1’de belirtildiği gibi, WinCC Scada – Simotion Kontrol Sistemi ve Profil Delme/Kesme Makinesi olmak üzere üç bölüme incelemek mümkündür. Her bir bileşen, ayrı ayrı büyük görevleri yerine getirir.



Şekil 2.1. Mevcut durumda çalışan Profil Delme ve Kesme Makinesi sistemi.

Temel olarak WinCC, otomatikleştirilmiş süreçlerin izlenmesi için güçlü fonksiyonları ile ölçeklenebilir bir süreç görüntüleme sistemidir [11]. Bir ekran üzerinden işlemlerin durumu hakkında bilgi alınabilmesini sağlar.

Scada (Supervisory Control And Data Aquation – Gözetleyici kontrol ve veri toplama), sahadan veri toplama ve sahadaki aktüatörleri denetlemek amacıyla kullanılan bir sistemdir [10]. Aynı zamanda sahadaki ekipmanların kontrol ve kumanda edilmesine imkan sağlamaktadır.

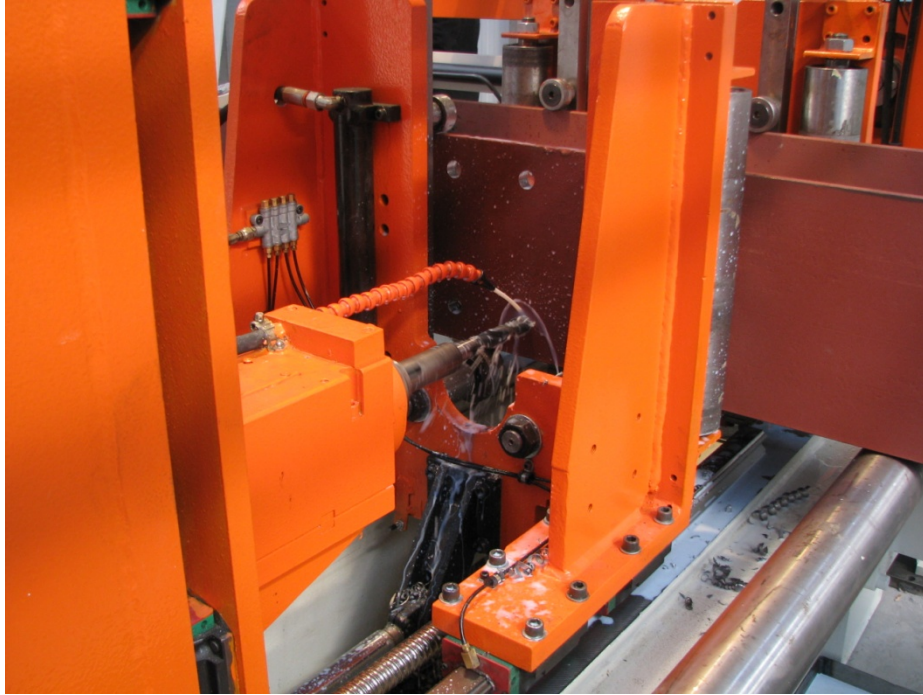
Mevcut sistem, oluşturulan çizim dosyalarında bulunan profillerin manuel yöntemler kullanılarak incelenmesi, yapılacak delme işlemlerine ait koordinat bilgilerinin belirlenmesi ve sisteme aktarılması şeklinde çalışmaktadır. Sistemde reçetelerle sıkça tekrarlanan uygulamalar otomatik olarak seçilebilmektedir. Özellikle birbirinden farklı projelere yönelik uygulama gerçekleştirilmesi durumunda her bir proje için aynı adımların tekrarlanması gerekmektedir. Bu ise hata riskinin ve harcanan zamanın fazla olmasına sebebiyet vermektedir. Sistem girilen boyut, koordinat, yüzey ve yüzeylere ait deliklerin çaplarına bağlı olarak, gerekli durumlarda matkap ucu değişimlerini yine otomatik olarak yapmaktadır.

Delme işlemi, profilin türüne bağlı olarak üç eksende gerçekleştirilebilmektedir. Şekil 2.2 uygulamada olan bir Profil Delme ve Kesme Makinesini göstermektedir. Sistem CNC (Computer Numerical Control) kontrollü olarak çalışmaktadır. Bant üzerinde ilerleyen profil, ilgili koordinatlarda sıkıştırılarak, hareket kontrollü 1 dikey ve 2 yatay olmak üzere toplamda 3 iş mili (matkap) sayesinde delinmektedir. Bu iş millerinin eksen boyunca olan hareketleri eşzamanlı gerçekleşmektedir. Ayrıca profil, gerektiğinde gönyeli olarak kesilebilmektedir.



Şekil 2.2. Uygulamada kullanılan Profil Delme ve Kesme Makinesi görüntüsü.

Sistemdeki bütün hareket kontrolleri Siemens Simotion kontrol ünitesi ile gerçekleştirilmektedir. Malzemenin ilerletilmesi sistemi ve 3 eksenle gerçekleştirilen delme sistemleri servo motor ile sürülmektedir. İlerletme sistemi aynı zamanda malzemenin referans noktasını belirleyerek pozisyonun belirlenmesini sağlamaktadır. Hassas konumla yapılmaya ihtiyacı olan delme işlemlerinde de sıfırlama işlemi ile her bir eksen için bir referans noktası belirlenmektedir. Simotion sistemi alarmların oluşturulması, reçetelerin tutulması, limit sensörlerine göre işlem yapılması gibi diğer destekleyici fonksiyonları da gerçekleştirmektedir. Simotion sistemi temel olarak PLC (Programmable Logic Control) sistemlerinin pek çok özelliğini taşımakla beraber eksen kontrolü uygulamaları için sürücü optimizasyonu gibi bir kısım üstünlükleri de taşımaktadır.



Şekil 2.3. Profil Delme ve Kesme Makinesi yüzey delme işlemi.

Mevcut Profil Delme ve Kesme Makinesi; 1.200 mm genişliğinde ve 600mm yüksekliğinde profilleri işleme kapasitesine sahiptir. Sistem, maksimum 12.000 mm uzunluğundaki U, H, I, L, Kutu tipi profillerin delme ve kesme işlemini yapabilmektedir.

## BÖLÜM 3

### ÇİZİM DOSYASININ TANINMASI

Günümüzde bilgisayar destekli tasarım ve üretimi gerçekleştirmek üzere, Catia, AutoCAD, SolidWorks ya da Solid Edge gibi yazılımlar kullanılmaktadır. Bu yazılımlar, elde edilmek istenen ürün hakkında üretimden önce birçok bilgi sunmak konusunda kolaylıklar sağlamaktadır. Bu yazılımlar, çizimlerde sağladığı kolaylıklar, hassas ölçüm ve birçok işleme özelliği sayesinde tercih edilebilmektedir. Hazırlanan çizim dosyaları disk üzerinde genellikle Dwg, Par, Stp ya da Model gibi dosya uzantılarıyla saklanmaktadır.

2007 yılında Mansour N. Jalid ve Mohammad M. Idrees, bir binanın yapısal çizimine ait Dxf dosyasından yola çıkarak, inşaat öncesi maliyetlerin tespit edilmesi üzerinde çalışmışlardır [12].

2008 yılında Sukimin, Z. ve Haron, H., geometrik varlıklara ait 3 boyutlu katı modellerin özelliklerinin çıkarımı ve dışarıya aktarımı konusunda Dxf dosyalarını kullanmışlardır [13].

2009 yılında Lin, W. Ve arkadaşları, yüzey gözetleme radarında Dxf dosyasının okunabilmesi için 3. şahıs kütüphane kullanmış ve bu sayede havaalanı haritası ve uçak çizimlerinde OpenGL arabirimleri kullanılarak kullanıcıların görüntü etkileşiminde bulunmasını sağlamışlardır [14].

Aynı yıl Jin, Y. ve arkadaşları, Dxf dosyalarından Nurbs eğrileri için gerekli bilgilerin okunması metodu üzerinde durarak, open-CNC ve Cad / Cam yazılımları arasında bütünleştirme çözüm önerilerinde bulunmuşlardır [15].



2010 yılında Pin, W ve Lan-zhi, S. Cad/Cam yazılımları ile CNC sistemi arasında bilgi paylaşımı ve grafik programlama sisteminde kullanılmak üzere NC (Numerical Control) kodlarını üretmek amacıyla Dxf dosyalarını kullanmışlardır [16].

Yine 2010 yılında Cheng, J. ve arkadaşları, Dxf dosyalarında bulunan 3 boyutlu toprak kayması modelleri bilgilerini değerlendirerek arazi yüzeyi ile kayan yüzeyin karşılaştırılması üzerinde çalışmışlardır [17].

2012 yılında Wang, X. ve arkadaşları, endüstriyel robotlar için karmaşık yol planlama talebini karşılamak amacıyla, Dxf dosyalarını kullanan bilgisayar destekli programlama sistemi geliştirmişlerdir [18]. Geliştirilen sistem, Dxf dosyalarında bulunan çizgi, eğri ve daire bilgilerini analiz ederek ayıklamakta ve elde ettiği seyahatleri düzenleyip verimliliği arttırarak doğrudan robotun çalışma dosyasına dönüştürmektedir.

Shang, W. ve arkadaşları ise yine 2012 yılında bir 3D uygulama ile görüntülemek üzere Dxf dosyalarını incelemişlerdir [19]. Daha çok harf ve karakterlerin incelenmesi üzerinde çalışmışlardır.

Yan, L. Ve Xing, Z. aynı yıl Dxf dosyalarının QT platformunda kullanılmak üzere çıkarımı ve yeniden oluşturulması tekniği üzerinde çalışmışlardır [20]. QT platformu, masaüstü, tümleşik ve mobil platformlar için kullanıcı ara yüzü ve uygulamaların geliştirilmesini kolaylaştırmak için tasarlanmış araçlarıyla tam bir geliştirme ortamıdır [21].

Her bir bilgisayar destekli tasarım yazılımı dosya formatı, kendisine ait binary yapı kullanmaktadır. Bu çalışmada hemen her türlü CAD programı tarafından ortak olarak kullanılan, AutoDESK firması tarafından diğer CAD programları ile kararlı veri değişimlerini sağlamak amacıyla geliştirilmiş Dxf dosya formatı tercih edilmiştir. İşlenecek dosya CAD programlarından biri tarafından oluşturulduktan sonra Dxf uzantılı olarak yeniden dönüştürülerek, hazırladığımız yazılım ile işlenebilir hale getirilir. Bu dönüşüm işlemi ile çizim dosyalarında bazı veri kayıpları olabilmektedir. Çalışma kapsamında kullanılan veriler için böyle bir durum söz konusu

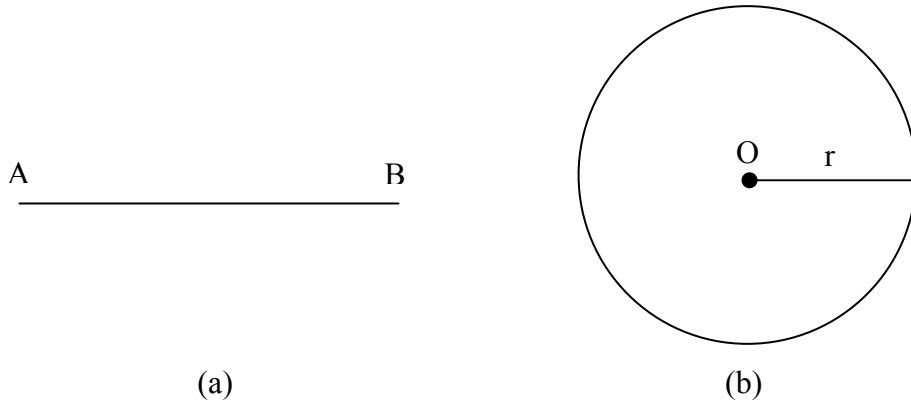
olmadığından güvenle kullanılabilceđi düşünölmektedir. Bir Dxf dosyası basit bir not defteri yardımıyla açılarak içeriđi görüntölenebilir.

Bu bölümde, Dxf dosya türüne dönüştürölmüş bir çizim dosyasının yazılım tarafından okunması ve içerisinden çalışmada kullanılabilcek verilerin ayıklanması işlemleri üzerinde durulmuştur. Genel olarak bu aşamaları iki bölümde incelemek mümkündür;

- Dosyanın okunması ve yazılım tarafından bulunan bütün çizgi ve daire bilgilerinin veritabanı tablolarına aktarılması,
- Veritabanı tablolarına aktarılan bilgilerin, bir sistematik içerisinde incelenmesi ve içlerinde anlamlı olanları kullanılabilir bir yapıda ayrı tablolara tekrar aktarılması, yani bilgilerin anlamlandırılması.

### 3.1. DXF DOSYA YAPISI

DXF dosya formatı, çizim dosyasındaki tüm bilgi ve içeriđin etiketlenmiş veri sunumudur [22]. Oluşturulan etiketlenmiş yapı, tüm içeriđin sayısal değerler ile gruplandırılmış sunumu şeklindedir. Örneđin bütün çizim nesneleri için belirtilecek “Başlangıç X” koordinatı değeri “10” ile ifade edilir ve bu ifade grup kodu olarak isimlendirilir.



Şekil 3.1. Örnek çizim parametreleri; a) çizgi ve b) daire.

Şekil 3.1. çizim dosyalarında bulunan çizgi ve dairelere ait parametrik ifadeleri göstermektedir. Örneğin bir çizgiyi belirtmek için Dxf dosyası içerisinde iki tip koordinat bilgisi bulunmaktadır. Bunlardan birincisi, çizginin başladığı noktaya ait  $x$ ,  $y$  ve  $z$  koordinatlar, ikincisi ise çizginin son bulacağı  $x$ ,  $y$  ve  $z$  koordinat bilgileri. Çizim dosyalarında bulunan çizgilere ait başlangıç ve bitiş noktaları, çizim şekline bağlı olarak değişiklik göstermektedir. Örneğin, Şekil 3.1. göz önüne alındığında eğer çizim işlemi A noktasından başladıysa, başlangıç koordinat bilgileri A noktasını, B noktasından başladıysa başlangıç koordinat bilgileri B noktasını işaret etmektedir. Yani, çizgi ya sağa doğru, ya da sola doğru bir ok işareti belirtecek nitelikte olduğu söylenebilir. Daire çizim nesnesinde ise, daire merkezinde bulunan O noktası, Dxf dosyası içerisinde başlangıç  $x$ ,  $y$  ve  $z$  koordinat bilgileri ile temsil edilir.

Çizelge 3.1. Dxf dosyası çizgi grup kodları ve anlamları.

<b>Grup Kodu</b>	<b>Anlamı</b>
8	Çizginin bulunduğu katman adı
10	A başlangıç noktasının $x$ koordinat bilgisi
20	A başlangıç noktasının $y$ koordinat bilgisi
30	A başlangıç noktasının $Z$ koordinat bilgisi
11	B bitiş noktasının $x$ koordinat bilgisi
21	B bitiş noktasının $y$ koordinat bilgisi
31	B bitiş noktasının $Z$ koordinat bilgisi

Çizelge 3.2. Dxf dosyası daire grup kodları ve anlamları.

<b>Grup Kodu</b>	<b>Anlamı</b>
8	Dairenin bulunduğu katman adı
10	O merkez noktasının $x$ koordinat bilgisi
20	O merkez noktasının $y$ koordinat bilgisi
30	O merkez noktasının $Z$ koordinat bilgisi
40	O merkezli dairenin $r$ yarıçap ölçüsü

Uygulama yazılımı hazırlanırken kullanılan etiket grup kodları ve anlamları Çizelge 3.1 ve Çizelge 3.2’de gösterilmiştir. Çizelge 3.1’de belirtilen koordinat bilgilerinde, Şekil 3.1.’de verilen çizgi için A noktası başlangıç ve B noktası bitiş noktası olarak kabul edilmiştir.

Bu dosya formatı, sadece Grup kodları ve bu grup kodlarına karşılık gelen bilgileri içerdiğinden satırlar kısa fakat sayıca çok fazladır. Basit bir çizim dosyasında 50.000’den fazla satır bulunmaktadır. Bu satırlar arasında çizim bilgileri dışında, ilgili dosyaya ait çizim alanı, sayfa yapısı veya çizim standart parametreleri gibi bilgiler de yer alır. Şekil 3.2, yukarıda belirtilen çizgi ve daire çizim nesnelere ait örnek Dxf dosya içeriğinin yapısını göstermektedir.

0	0
LINE	CIRCLE
5	5
F1C8	F1C0
330	330
F1C5	F1BE
100	100
ACDbEntity	ACDbEntity
8	8
GOVDE	GOVDE
100	6
ACDbLine	Continuous
10	100
436.8189596382217	ACDbCircle
20	10
-25.5	9.999996983868186
30	20
0.0	9.999998773484435
11	30
436.8189596382217	0.0
21	40
42.00000000000001	10.0
31	0
0.0	CIRCLE
0	5
LINE	

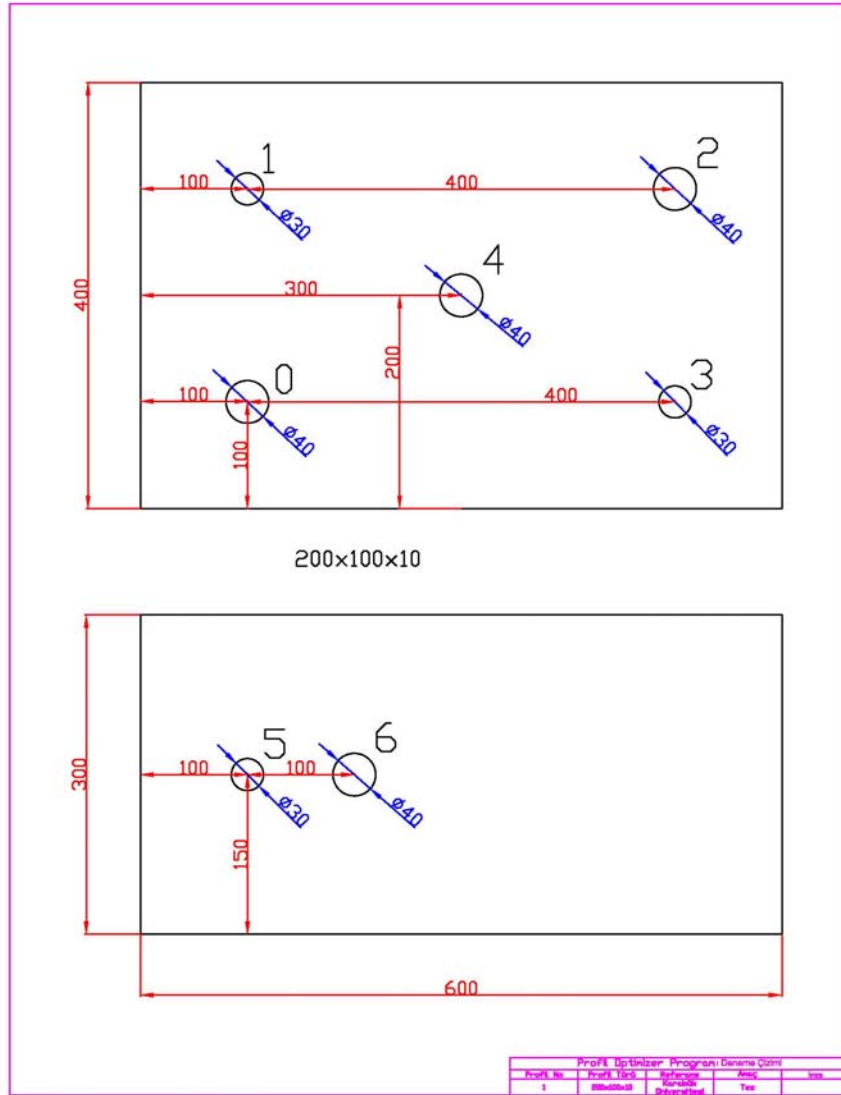
(a) (b)

Şekil 3.2. Basit bir a) çizgi ve b) daire çizim nesnesine ait Dxf dosya içeriği.

Çalışma kapsamında hazırlanacak yazılım tarafından incelenmesi ve optimizasyon işleminin gerçekleştirilmesi planlanan örnek bir çizim dosyası Şekil 3.3’te gösterilmiştir. Şekil üzerinde, çizgiler, daireler, ölçüler ve çerçeve gibi birbirinden farklı ve farklı anlamlar taşıyan çeşitli bilgiler bulunmaktadır. İncelenecek profile ait yüzeylerin dış hat bilgileri çizgilerle, her bir yüzeyde yapılacak delme işlemine ait

delikler de dairelerle tanımlanmıştır. Geri kalan koordinat ve dış çerçeve gibi bilgiler, Dxf dosya organizasyonu içerisinde bulunacağından yazılım tarafında anlamsız sayılacaktır.

Yazılım tarafından anlamlı olduğu belirlenen bu bilgiler, ilk inceleme aşamasında çizgiler ve daireler olmak üzere iki farklı tabloda, koordinat ve boy bilgileri ile birlikte saklanacaktır. Bu aşamada, elde edilen ve tablolara aktarılan daire ve çizgi bilgileri, anlamlı olup olmadıkları ve kullanılabilirlikleri dikkate alınmadan incelenmektedir.



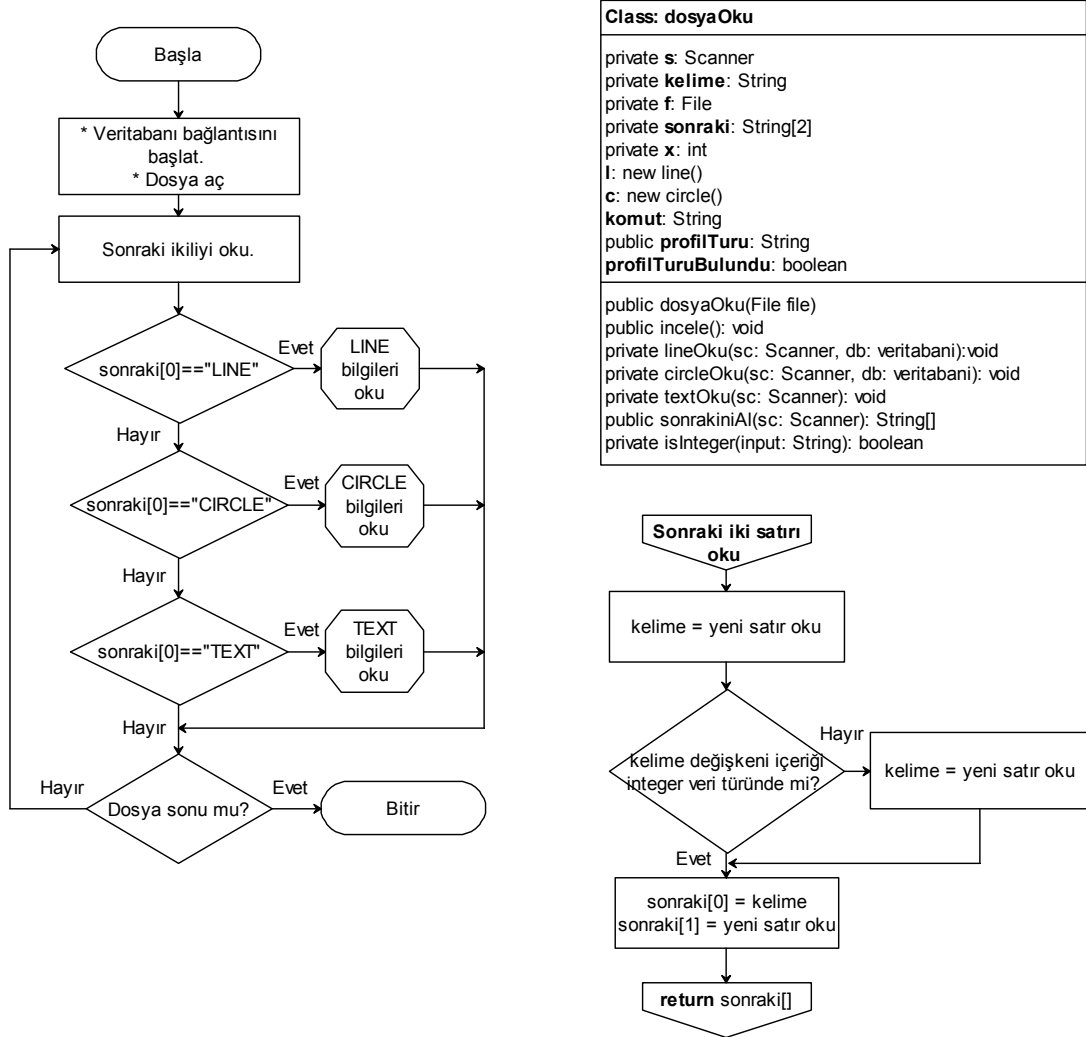
Şekil 3.3. Çalışma kapsamında incelenmesi planlanan örnek çizim dosyası.

Elde edilen bilgilerin kullanılabilir ya da anlamlı olup olmadıklarının değerlendirilmesi işlemi ikinci inceleme aşamasında gerçekleşmektedir. Bu aşama sadece ölçü, numaralandırma veya çerçeve gibi yazılım içerisinde anlamları olmayan bilgilerin temizlenmesi ve kullanılabilmesi varsayılan çizgi ve daire bilgilerinin elde edilmesi içindir.

### **3.2. DOSYANIN OKUNMASI**

Basit bir çizgi ve daireye ait örnek bir Dxf dosya içeriği daha önce Şekil 3.2.'de gösterilmişti. Dosya okuma işlemi gerçekleştirilirken, dosya baştan sona satır satır incelenmektedir. Bu inceleme işleminin gerçekleştirilme aşamaları Şekil 3.4'te verilen akış diyagramı yardımıyla ifade edilebilir. Burada, dosyanın baştan sona okunması ve ilgili bilgilerin veritabanına aktarılması amacıyla kullanılan “dosyaOku” sınıfı, bu sınıfa ait metod ve değişkenler gösterilmiştir.

Dosya incelenirken satırlar, grup kodu ve bu grup koduna karşılık gelen veri olmak üzere ikili gruplar halinde değerlendirilmiştir. Bu ikili grupların birinci elemanı grup kodunu, ikincisi ise bu grup koduna karşılık gelen veriyi temsil etmektedir. İncelenen ikililere ait grup kodu “0” sayısı ise, bir önceki çizim nesnesinin sonlanması ve bir sonrakinin başlaması anlamına gelmektedir. Grup kodu “0”a eşit olduğunda, yeni başlayacak olan çizim nesnesinin ne olduğu bu grup koduna ait verinin kontrol edilmesi ile anlaşılmaktadır. Şekil 3.4. incelendiğinde, “0” grup koduna karşılık gelen üç tip çizim nesnesi beklenmektedir. Bunlar; çizgi nesnesine ait bilgilerin başlayacağı anlamına gelen “LINE”, daire nesnesine ait bilgilerin başlayacağı anlamına gelen “CIRCLE” ve metin nesnesine ait bilgilerin başlayacağı anlamına gelen “TEXT” verileridir. Her bir çizim nesnesi, kendisi ile ilgili bilgilerin incelenerek alınması ve ilgili veritabanı tablosuna aktarılmasını sağlayan ayrı metodlar yardımıyla incelenmektedir. Bu metodlarda yapılacak inceleme işlemi, bir sonraki “0” grup koduna kadar devam eder. “0” grup koduna karşılık gelen “TEXT” verisinin incelenmesi işlemi sadece çizim dosyası içerisinde profil türü bilgisinin tespit edilmesi için gerçekleştirilmektedir.

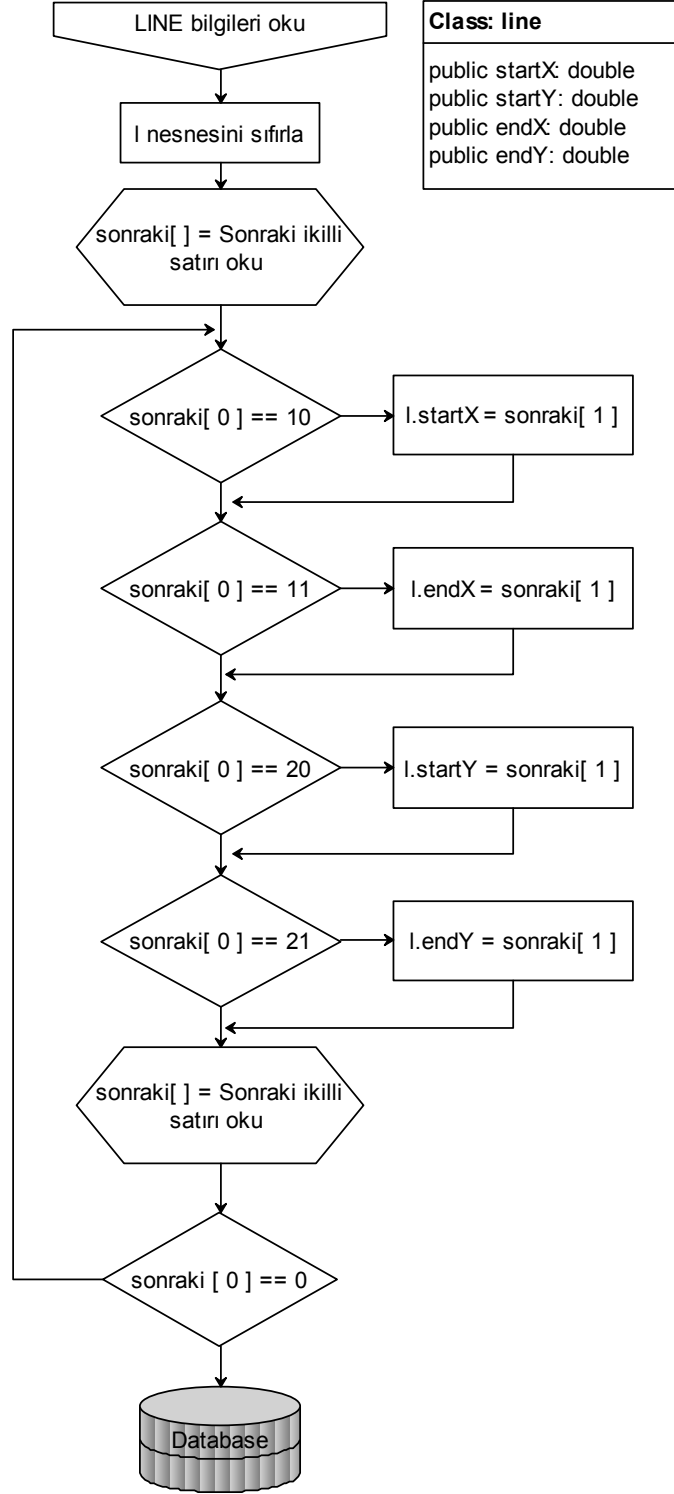


Şekil 3.4. Dosya okuma işlemine ait genel akış şeması.

### 3.2.1. Çizgi Bilgilerinin İncelenmesi ve Veritabanına Aktarılması İşlemi

Dosya okuma işlemi sırasında, “0” grup koduna karşılık gelen veri “LINE” olduğu takdirde, çizgi nesnesine ait özel bilgileri incelemek üzere satırlar yine ikili gruplar halinde incelenmektedir. Yapılan incelemede, bulunan çizgilere ait *startx*, *starty*, *endx* ve *endy* gibi bilgiler aranmakta ve bu bilgilere ait kontrol işlemleri gerçekleştirilmektedir. Daha önce, kullanılacak bu bilgilere ait grup kodları Çizelge 3.1’de verilmişti. Bu grup kodları ile karşılaşıldığında ilgili veri bir değişkene aktarılarak saklanmıştır. Grup kodu “0” olduğu takdirde, çizgi nesnesine ait bütün bilgiler elde edildiği anlaşılır ve bu metod, elde edilen verilerin “LINE” isimli veritabanı tablosuna eklenmesi ile sonlandırılır.

İnceleme işlemi sırasında çizgi nesnesine ait tespit edilen bilgileri daha düzenli tutabilmek amacıyla “line” isimli bir sınıf tanımlanmıştır.



Şekil 3.5. LINE çizim nesnesine ait bilgilerin okunmasını sağlayan akış şeması.



Bu sınıf çizgi nesnelere ait *startx*, *starty*, *endx* ve *endy* gibi bilgileri tutacak şekilde hazırlanmış ve kullanılmak üzere “1” adıyla nesneye dönüştürülmüştür. Gerekli bilgilerin tamamı elde edilene kadar bu nesne içerisinde veritabanı tablosuna aktarılmaya kadar saklanmıştır. Çizgi bilgileri, çizim dosyası içerisinde çizilme sırasına göre yer aldığından dolayı veritabanı tablosuna da yine çizilme sırasıyla aktarılmaktadır.

Bu noktada önemli bir ayrıntı dikkate alınmalıdır. Daha önce de belirtildiği gibi çizgi nesnelere ait başlangıç ve bitiş noktaları, nasıl çizildiğine göre farklılık göstermektedir. Bu durum, anlamlandırma aşamasında çizgilerin oluşturacağı kapalı kutuların incelenmesi için çok detaylı bir inceleme yapılması gerektiği anlamına gelir. Elde edilen çizgi nesnelere ait bilgiler olduğu gibi veritabanı tablosuna aktarılmaktansa, gerçekleştirilecek topolojik bir düzeltme işlemi, tek bir yöntem kullanarak oluşacak kapalı kutuların tespit edilebilmesini sağlar. Bu işlem için küçük koordinatın yatay çizgilerde solda, dikey çizgilerde aşağıda olduğu ve bu noktanın başlangıç noktası olduğu, büyük koordinatın yatay çizgilerde sağda, dikey çizgilerde ise üstte olduğu ve bu noktanın bitiş noktası olduğu varsayılarak veritabanı tablosuna aktarılması gerekir. Bu varsayım ile dört değişik yönde olduğu düşünülen çizgi nesnelere için yapılacak değişim işlemi sonucu Şekil 3.6’da gösterilmiştir.

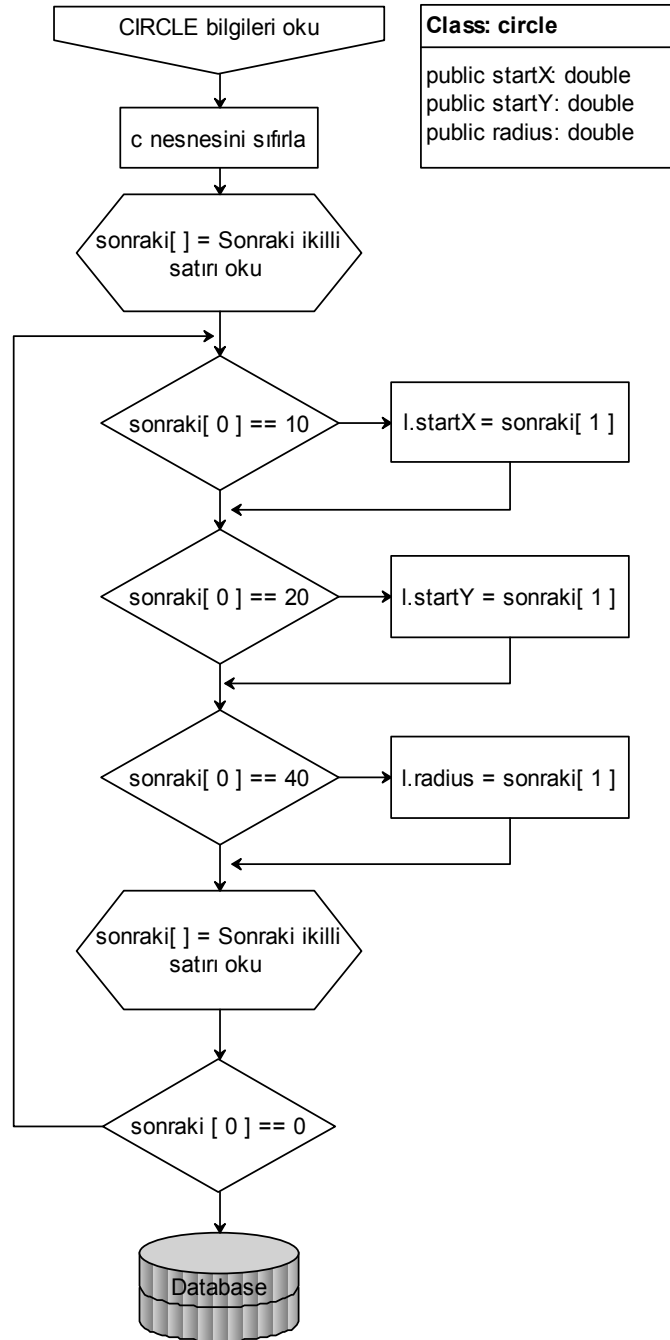


Şekil 3.6. Çizgi nesnelere tek tip haline getirilmesi.

Yapılacak bu değişim işlemi yardımıyla elde edilen inceleme kolaylığı, verilerin anlamlandırılması aşamasında daha detaylı olarak açıklanacaktır.

### 3.2.2. Daire Bilgilerinin İncelenmesi ve Veritabanına Aktarılması İşlemi

Daire bilgilerinin incelenmesi işlemi de çizgi bilgilerinin incelenmesi işlemi ile benzerdir. Farklı olarak burada, bulunan dairelere ait *startx*, *starty* ve *radius* bilgilerinin incelenmesidir. Şekil 3.7, “0” grup koduna karşılık “CIRCLE” verisi ile karşılaşıldığı zaman çalıştırılacak akış şemasını göstermektedir.



Şekil 3.7. CIRCLE çizim nesnesine ait bilgilerin okunmasını sağlayan akış şeması.

Bu aşamada daire nesnesine ait verilerin elde edilmesi için kullanılacak grup kodları daha önce Çizelge 3.2’de verilmişti. Daire nesnesine ait bilgilerin düzenli tutulabilmesi için yine “circle” isimli bir sınıf tanımlanmış ve “c” adıyla nesneye dönüştürülerek gerekli bilgiler bu nesne içerisinde veritabanı tablosuna eklenene kadar saklanmıştır. Daire bilgileri de çizgilerde olduğu gibi, çizim dosyası içerisinde çizilme sırasına göre yer aldığından yine veritabanı tablosuna da aynı sırayla aktarılmaktadır.

### **3.3. ELDE EDİLEN BİLGİLERİN ANLAMLANDIRILMASI**

Anlamlandırma; birinci inceleme işlemi sonucunda elde edilen çizgi ve daire bilgilerinden kullanılabilir olanlarının belirlenmesi ve bu bilgilerin kullanılacağı şekliyle yeni bir veritabanı tablosuna aktarılması olarak tanımlanabilir. Bir bakıma bu işlem, veri yığını içerisinde kullanılabilir olanların belirlenmesi aşamasında veri madenciliği alanının bir bileşeni olarak ta düşünülebilir.

Birinci inceleme işleminin tamamlanmasıyla, çizim dosyası içerisinde bulunan bütün çizgi ve daire bilgileri veritabanı tablolarına aktarılmıştır. Bu aşamada yapılacak inceleme işlemiyle, elde edilen bilgiler arasında kullanılabilir olanların ayıklanması ve anlamlandırılması işlemi gerçekleştirilecektir. Anlamlandırma işlemi sistemli bir şekilde gerçekleştirilmelidir. Bu işlemin gerçekleştirilmesi için izlenecek adımlar şu şekilde özetlenebilir;

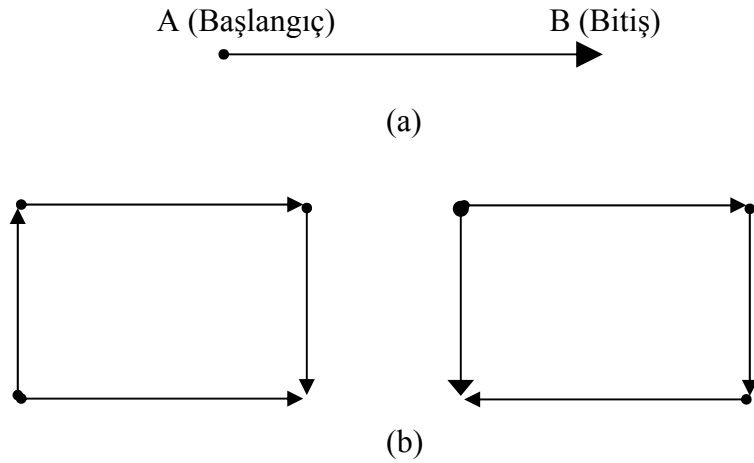
- LINE veritabanı tablosundaki çizgilerin tamamı incelenerek başlangıç ve bitiş noktalarının çakışmasıyla dört adet çizginin kapalı bir kutu oluşturması araştırılır.
- CIRCLE veritabanı tablosu incelenerek, belirlenen bu kapalı kutulara ait koordinat bilgileri arasında kalanların delik olarak isimlendirilmesi.

Böylece, hem profil yüzeylerine ait dış hatları tanımlayan kapalı kutu, hem de bu yüzeylere ait delik bilgileri aynı anda birbirleriyle eşleştiğinde, kullanılabilir veriler, bütün verilerin bulunduğu yığın içerisinde seçilmiş ve anlamlandırılmış olur. Belirtilen bu aşamalar aşağıda detaylı olarak incelenmiştir.

### 3.3.1. Çizgilerin Anlamlandırılması İşlemi

Çizim dosyasında bulunan çizgi nesneleri, profil yüzeylerini belirleyen dış hatların tanımlanmasında kullanılmaktadır. Ayrıca bu dış hatlar bize profilin boyu konusunda da bilgi vermektedir. Bir çizginin bitim noktasında başka bir çizginin başlaması, ilişkisel anlamda birbirlerini önemli kılmaktadır. Yapılacak incelemede, çizgilerin bitim noktalarında başlayan başka çizgilerin olup olmadığı ve bu şekilde devam ederek dört adet çizginin kapalı bir kutu oluşturup oluşturmadıkları kontrol edilecektir. Böylece oluşacak kapalı kutu, çizim dosyasında bulunan profile ait herhangi bir yüzeyi temsil edebilecektir.

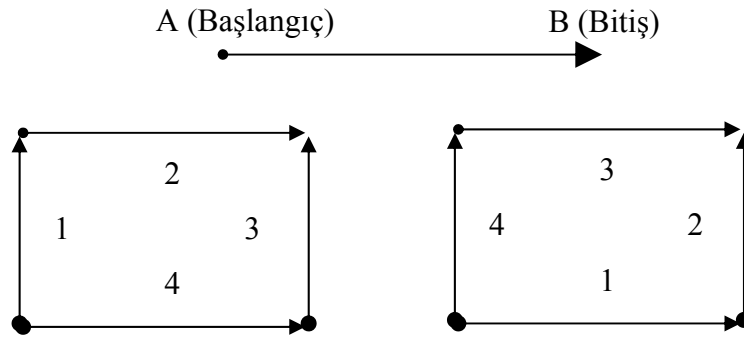
Daha önce de belirtildiği gibi çizgilerde, başlangıç ilk çizmeye başlanan noktayı, bitiş ise çizginin sonlandığı noktayı ifade etmektedir. “LINE” veritabanı tablosuna aktarılma aşamasında başlangıç ve bitiş noktaları topolojik düzeltme işlemi yapılmamış olsaydı, kapalı kutu oluşturan çizgilerin belirlenmesi için Şekil 3.8’de belirtildiği gibi farklı dizilimlerin, her bir çizgi için defalarca taranması gerekecekti.



Şekil 3.8. Örnek olarak a) bir çizgiye ait temsili gösterim ve b) kapalı kutu oluşturduğu düşünülen ve değerlendirilebilecek iki ayrı dizilim.

Bu şekilde yapılacak kontrol işlemi, her bir çizgi için gereksiz kontrol sayısı artışına sebep olurken, yapılacak kontrol işleminin her bir türüne göre kontrol aşamasının detaylarının artmasına ve zorlaşmasına da sebep olacaktır. Daha önce ifade edildiği gibi çizgiler için topolojik düzeltme işlemi gerçekleştirilerek, küçük koordinat

bilgisinin başlangıç ve büyük koordinat bilgisinin bitiş olarak veritabanı tablosunda saklanmasıyla bu olumsuzluklar ortadan kaldırılmıştır. Dolayısıyla bu düzeltme, yapılacak kontrol işleminin tek bir yöntemle gerçekleştirilebilmesini sağlamış olur. Şekil 3.9, çizgi nesnelерinin topolojik düzeltme işlemi sonucunda oluşması beklenen kapalı kutuyu ve geliştirilen yazılımda bu kapalı kutuların tespit edilmesi için izlenen yöntemi temsil etmektedir. Kapalı kutu içerisinde verilen sayılar, ilgili çizginin incelenme ve tespit edilme sırasını ifade etmektedir.

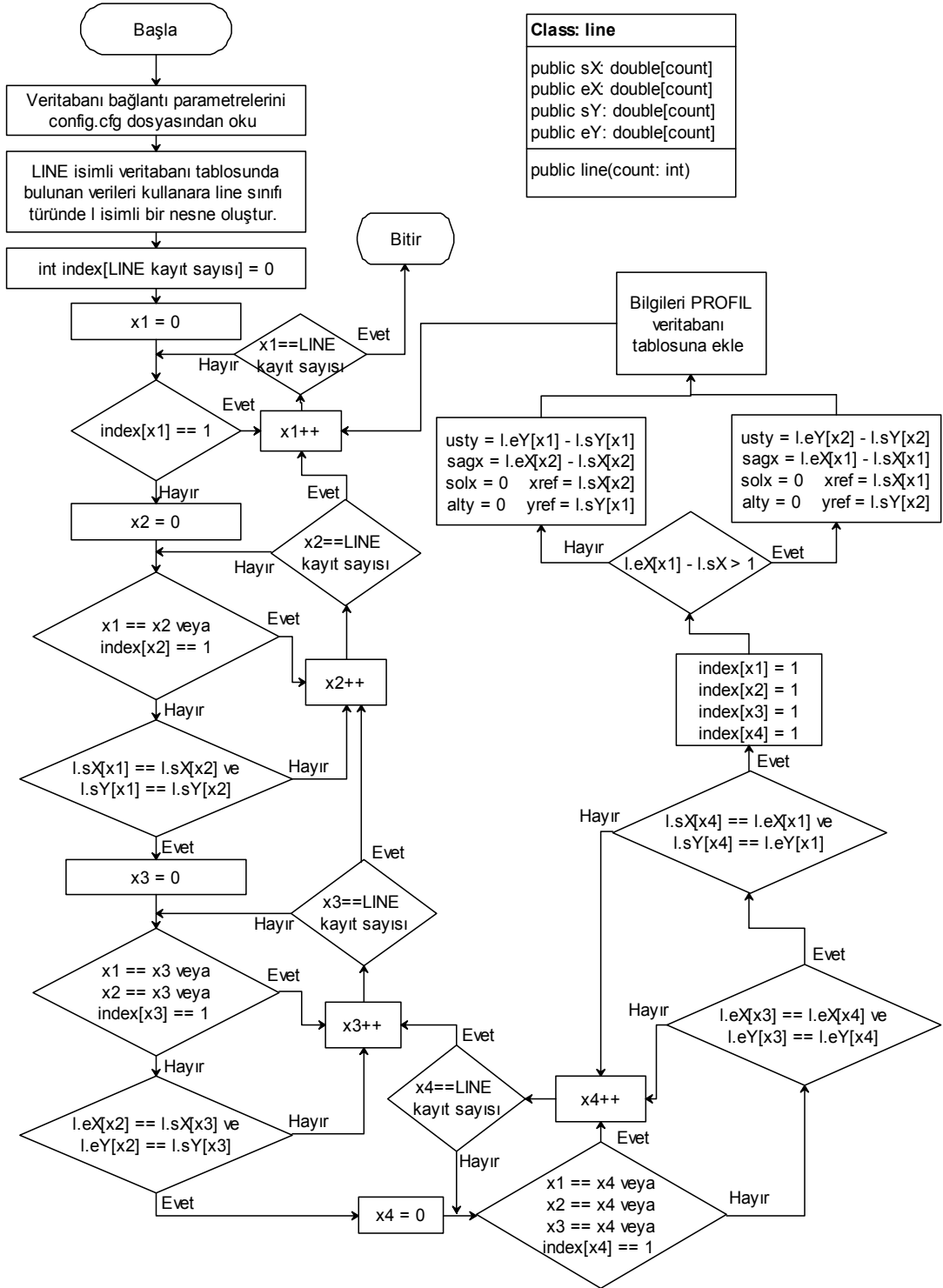


Şekil 3.9. Geliştirilen uygulamada kullanılan inceleme yöntemi.

Yapılacak inceleme işleminde, Şekil 3.9’da verilen her iki dizilim göz önüne alındığında ilgili sıralamadaki çizgilerin başlangıç ve bitiş noktalarının birleşme sırası aynı olmaktadır. Diğer bir ifadeyle, 1 numaralı çizginin bitiş noktası 2 numaralı çizginin başlangıç noktasına, 2 numaralı çizginin bitiş noktası 3 numaralı çizginin bitiş noktasına, 3 numaralı çizginin başlangıç noktası 4 numaralı çizginin bitiş noktasına ve son olarak oluşacak kutunun kapanmasını sağlayacak şekilde 4 numaralı çizginin başlangıç noktası 1 numaralı çizginin başlangıç noktası ile birleşmektedir. Verilen her iki örnek te aynı şekilde tanımlanabilmektedir. Bu ise tek bir kontrol yordamının, oluşacak kapalı kutuları tanımlamak için yeterli olacağı anlamına gelmektedir. Bu şekilde her bir çizgiyi, çok sayıda karşılaştırma işlemine tabi tutmak yerine, bir sistematik oluşturmak ve tek bir karşılaştırma işlemi ile sonuca ulaşmak yazılımın verimliliği açısından daha doğru olacaktır.

Çizgilerin incelenmesi ve anlamlandırılması işleminin gerçekleştirilmesini sağlayan akış diyagramı Şekil 3.10’da gösterilmiştir. Her bir çizgiye ait bilgileri saklamak ve gerektiğinde kullanabilmek amacıyla “line” isimi bir sınıf oluşturulmuş, incelenecek

her bir çizgiye ait bilgiler bu sınıfın nesneye dönüştürülmesinin ardından veritabanı tablosundan bu nesnede yer alan dizilere aktarılmıştır. Böylece yapılacak inceleme işlemi sırasında, veritabanı veri akışı yoğunluğunu önlenmiş ve sistemin yoğunluğu sorunu giderilmiştir.



Şekil 3.10. Çizgilerin anlamlandırılması işlemine ait akış diyagramı.

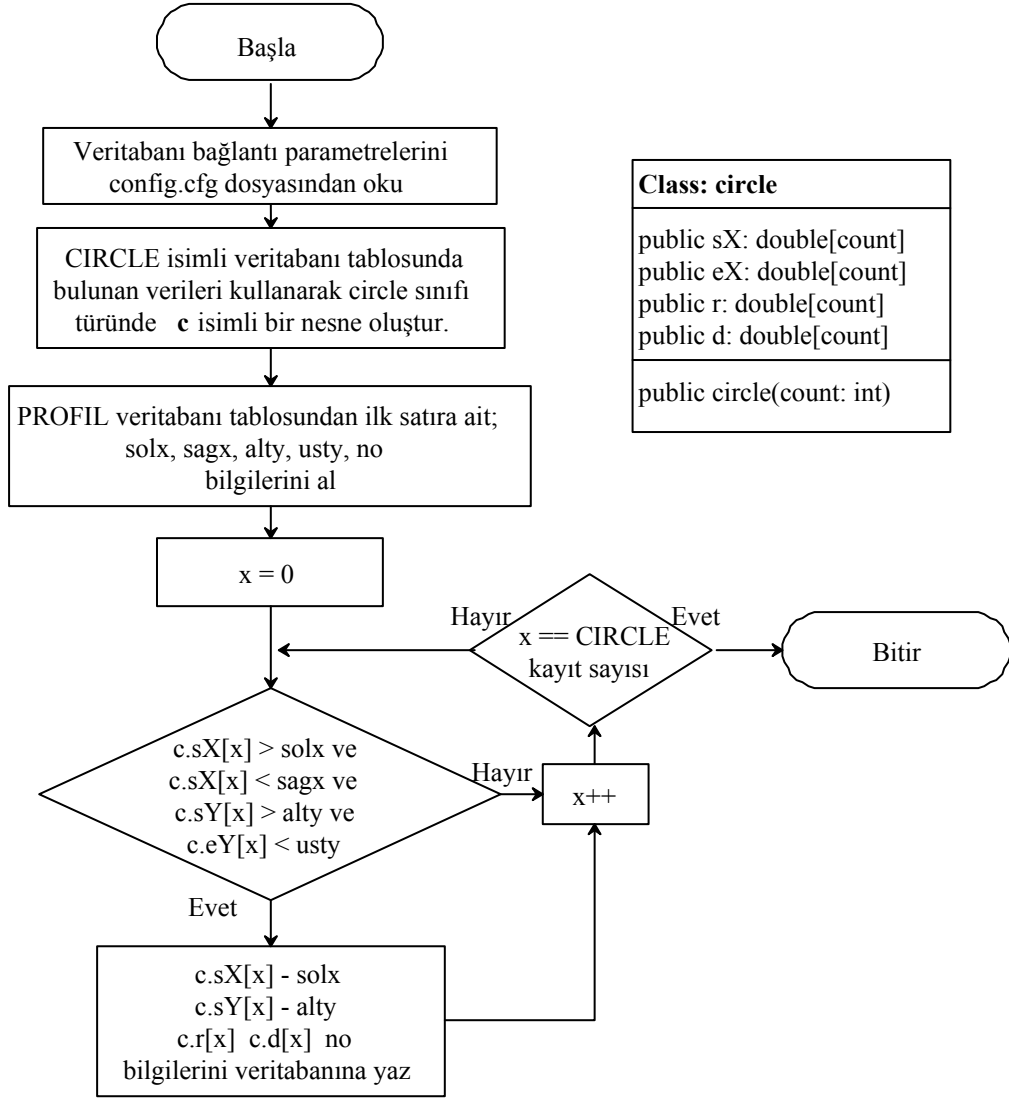
Elde edilen bu kapalı kutular, her bir çizgi için *startx*, *starty*, *endx*, *endy* gibi bilgilerinin saklanması yerine, bütünlük arz edecek şekilde *alty*, *usty*, *solx*, *sagx* biçimine dönüştürülerek “PROFIL” isimli veritabanı tablosuna kaydedilmiştir. Ayrıca daha sonra ölçekleme işlemlerinin daha verimli olması açısından, *solx* ve *alty* bilgileri sıfırlanarak, *x* referans değeri için *xref* ve *y* referans değeri için *yref* şeklinde iki alan daha tanımlanmış, *usty* ve *sagx* bilgileri de bu duruma uygun olarak yeniden değerlendirilmiştir. Ayrıca bulunan her kapalı kutu, profile ait bir yüzey anlamı taşımamaktadır. Bu ancak kapalı kutuya ait daire bilgisi mevcut ise söylenebilir.

### 3.3.2. Dairelerin Anlamlandırılması İşlemi

Çizim dosyasında bulunan çizgiler, kutu olarak tanımlandıktan sonra, bu kapalı kutu bilgileri kullanılarak “CIRCLE” veritabanı tablosunda bulunan daire bilgilerinin de gerekli olanlarının belirlenmesi ve delik olarak isimlendirilerek farklı bir veritabanı tablosuna aktarılması gerekmektedir.

Şekil 3.11, daire bilgilerinin anlamlandırılarak delik bilgisine dönüştürülmesi işlemine ait akış diyagramını göstermektedir. İnceleme işleminde yine koordinat bilgileri, veritabanı işlem yoğunluğunu arttırmamak amacıyla, “circle” isimli bir sınıf tanımlanarak bu sınıfın nesneye dönüştürülmesinin ardından ilgili dizilere aktarılmıştır.

İnceleme işlemi, her bir deliğe ait koordinat bilgileri, daha önce bulunan profillere ait *alty*, *usty*, *solx* ve *sagx* koordinatlarının arasında olup olmadığı karşılaştırılarak gerçekleştirilmektedir. Eğer bu şart sağlanırsa, ilgili daire koordinat bilgileri, hangi profil yüzeyine ait bir delik olduğunu anlamak amacıyla şartın sağlandığı profil numarası da eklenerek “DELİK” isimli veritabanı tablosuna aktarılmaktadır. Böylece daire bilgileri içerisinde anlamlı olanlar ayıklanmış, dolayısıyla daha önceden bulunan her bir dairenin delik olarak isimlendirilmek zorunda olmadığı anlaşılmıştır. Bu inceleme işleminin ardından sadece şartı sağlayan daireler delik olarak isimlendirilir.



```

Class: circle
public sX: double[count]
public eX: double[count]
public r: double[count]
public d: double[count]

public circle(count: int)
  
```

Şekil 3.11. Dairelerin anlamlandırılması işlemi akış diyagramı.



## BÖLÜM 4

### OPTİMİZASYON

Optimizasyon kelimesi Türk Dil Kurumu'na göre anlam olarak; “en uygun duruma getirme” demektir [23]. En uygun ifadesi ise uygulamanın çeşidine göre farklılık göstermektedir. Bazı uygulamalarda en yüksek, bazı uygulamalarda en düşük anlamına gelirken bazı uygulamalarda da ortada, ideal bir değer olarak ifade edilebilmektedir.

Literatürde üzerinde en fazla çalışılan optimizasyon problemi olarak “Seyahat Eden Satıcı” uygulaması ile karşılaşılmaktadır. Bu problemde; birbirinden farklı şehirleri gezmek isteyen bir satıcının, her şehre bir kez uğramak şartıyla turu en kısa sürede hangi yoldan tamamlayacağı sorusuna cevap aranmaktadır. Bu problemin çözümünde değişik yöntemler geliştirilmiştir. Çözüm önerisi olarak geliştirilen yöntemlerden sezgisel yolla sonuca ulaşmaya çalışan Nearest Neighbor (Yakın Komşu) Algoritması ve 2-Opt Exchange Algoritması ile ilgili olarak web sayfaları üzerinden çalışma mantıkları çevrimiçi olarak örneklerle izlenebilir [24, 25].

Eryavuz, M. ve Gencer, C., 2001 yılında merkezi bir depodan coğrafi olarak dağılmış çeşitli talep noktalarına dağıtım veya toplama rotalarının, araç filosunun kat ettiği toplam mesafeyi minimize etmek amacıyla Araç Rotalama Problemi (ARP - VRP) üzerinde çalışmışlardır [26]. Kullandığı tasarruf algoritmalarını takiben, rotaları daha iyileştirmek amacıyla 2-opt algoritması ve or-opt algoritmalarını kullanmıştır. Burada, farklı depolara erişim hakkı olan, çok sayıda farklı araç bulunmaktadır. Fakat gerçekleştireceğimiz çalışmada  $x$  eseninde bulunan kısmi bağımlılığa benzer, araçların hareketlerini tanımlayacak herhangi bir kısmi bağımlılık durumu bulunmamaktadır.

Tođan, V. ve Dalođlu, A., 2006 yılında yaptıkları alıřmalarında, üç boyutlu kafes sistemlerinin řekil ve boyut optimizasyonu iřleminde Genetik Algoritmaları kullanmıřlardır [27]. Burada üç boyut kavramı, kullandıkları kafesin uzunluk, geniřlik ve derinlik boyutlarına iřaret etmektedir.

Uđur, A., 2008 yılında üç boyutlu kúbik bir geometrik nesne yüzeylerinde bulunan noktaların, Genetik Algoritmalar kullanılarak minimum zamanda tamamlanmasını sađlamak amacıyla yol planlaması üzerinde alıřmıřtır [28]. alıřmasında, Öklid ifadelerinden faydalanarak noktalar arasında minimum yolu belirlemek üzere dik üçgendeki hipotenüsü kullanmıřtır. Noktaların deđerlendirilmesi amacıyla, bu noktaların kúp üzerinde birbirleri arasındaki yolculuđu temsil etmek üzere tek bir yüzey gibi deđerlendirmiřtir.

Jackson, J. vd., 2010 yılında araç rotalama problemi üzerinde alıřmıřlardır [29]. alıřmalarında tabu search ve 2-Opt Exchange algoritmalarının kombinasyonunu kullanmıřlardır. Bu alıřmada da araçların bařlangı noktalarına dönmeleri öngörölmüřtür

#### **4.1. PROFİL DELME OPTİMİZASYONU YAKLAřIMI**

Profil Delme ve Kesme Makinesi delme iřlemini, girilen parametrelere uygun olarak otomatik gerekleřtirmektedir. Kullanıcı tarafından girilen bu parametreler belirli bir sistematik olmaksızın verildiđinden dolayı, gerekenden fazla matkap ucu deđiřimi nedeniyle istenmeyen zaman kayıplarına yol aabilmektedir. Optimizasyonun uygulanmasında ama bu zaman kayıplarını minimuma indirmek ve en optimal sürede delme iřleminin gerekleřtirileceđi dizilimi elde etmektir. Maliyet hesaplama iřleminde;

- X Maliyet:  $X$  ekseninde gerekli koordinata ulařmak için gereken süre
- Y Maliyet: İlgili yüzeyde  $Y$  ekseninde gerekli koordinata ulařmak için gereken süre
- C Maliyet: İlgili yüzeyde gereken matkap ucu deđiřim için gereken süre
- D Maliyet: İlgili yüzeyde delme iřlemi için gereken süre

parametreleri belirleyici faktörlerdir. Farklı yüzeyler, birbirlerine göre farklı maliyetleri aynı zaman aralığında karşılayabilmektedir. Örneğin; bir yüzey  $Y$  Maliyetini karşılarken, diğer bir yüzey  $D$  Maliyetini karşılayabilir. Yüzeyler  $x$  eksen, dolayısıyla  $X$  Maliyeti dışında birbirlerinden bağımsızdır.  $X$  ekseninde oluşacak her hareket her üç yüzeyi de etkilemektedir. Bu parametreler dikkate alınarak en uygun zaman maliyetinin bulunması hedeflenmektedir.

Geliştirilecek yazılım kapsamında kullanılması gereken yöntem, bütün noktalardan minimum maliyetle geçilecek olması nedeniyle kısmen seyahat eden satıcı problemine benzerlik göstermektedir. Ancak satıcının başladığı şehre dönme noktasında farklılık vardır. Profil delme işlemi gerçekleştirilirken, son delme işlemi yapılan noktadan ilk delme işlemi gerçekleştirilen noktaya dönmek gibi bir zorunluluk bulunmadığından, delme işlemi yapılan ilk nokta ile son nokta birbirinden uzak olabilir.

Gerçekleştirilecek optimizasyon işleminde amaç toplam delme zamanını minimum süreye indirmektir. Toplam delme süresi hesaplanırken; matkap ucu değişimi ve delme kafasının delikler arasında hareket etme süresi zaman üzerine etki eden en belirleyici parametreler olduğundan bu iki değer dikkate alınmaktadır. Bir delikten diğerine geçerek delme işlemi gerçekleştirildiğinde geçen zaman maliyeti  $C(i, j)$  ise bu maliyet hesabı genel haliyle Denklem 4.1'deki gibi gösterilebilir.

$$\begin{aligned} C(i, j) &= x\text{Maliyet} + y\text{Maliyet} + c\text{Maliyet} + d\text{Maliyet} \\ &= |i_x, j_x| * Mhs + |i_y, j_y| * Mhs + Muds_{i \rightarrow j} + Mds \end{aligned} \quad (4.1)$$

Burada;

$|i_x, j_x| * Mhs$ :  $x$  ekseninde  $i$  deliğinden  $j$  deliğine olan hareket etme süresi

$|i_y, j_y| * Mhs$ :  $y$  ekseninde  $i$  deliğinden  $j$  deliğine olan hareket etme süresi

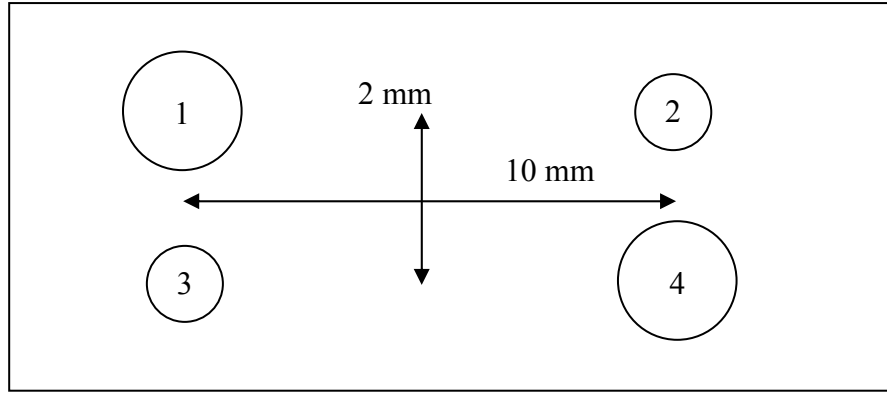
$Muds_{i \rightarrow j}$  :  $i$  deliğinden  $j$  deliğine geçerken matkap ucu değişim süresi

$Mds$  : Matkap ucunun deliği delme süresi

Eğer delik çapları aynı ise,  $C(i, j)$  maliyeti hesaplanırken  $Muds$  değeri dikkate alınmaz. Bu durumda maliyet;

$$C(i, j) = |i_x, j_x| * Mhs + |i_y, j_y| * Mhs + Mds \quad (4.2)$$

olacaktır. Optimizasyon işlemi bu noktada önem taşımaktadır. Minimum süreyi sağlamak için matkap ucu değişikliğinin yapılması ve sonra  $j$  deliğinin delinmesi sağlanabilirken, bazı durumlarda önce aynı ölçüdeki başka bir delik delindikten sonra matkap ucu değiştirilip  $j$  deliğine geçilebilir.



Şekil 4.1. Örnek 4 delikli profil çizimi.

Basit olarak Şekil 4.1’de gösterildiği gibi yerleştirilmiş 2 büyük ve 2 küçük olmak üzere 4 adet delikten oluşan bir profil yüzeyi örneğini düşünürsek, değişik delme dizilimlerine ait delme süreleri Çizelge 4.1’de gösterildiği gibi hesaplanabilir.

Yapılan hesaplamalarda görüldüğü gibi,  $Mhs = 1 \text{ sn/mm}$  ve  $Mds = 12 \text{ sn}$  olarak hesaplandığında optimum delme süresi 38 sn olan  $1 \rightarrow 4 \rightarrow 2 \rightarrow 3$  dizilimi kabul edilirken,  $Mhs = 2 \text{ sn/mm}$  ve  $Mds = 12 \text{ sn}$  olarak hesaplandığında optimum delme süresi 56 sn olan  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$  dizilimi kabul edilmektedir. Bu parametrelerin durumuna bağlı olarak dizilim de değişebilmektedir.

Çizelge 4.1. Örnek çizim için farklı dizilimdeki toplam maliyet hesaplamaları.

Mhs (sn/mm)	Muds (sn)	Delme Sırası	Toplam Maliyet (sn)
1	12	1→2→3→4	56
		1→2→4→3	56
		1→3→2→4	40
		1→3→4→2	50
		<b>1→4→2→3</b>	<b>38</b>
		1→4→3→2	46
2	12	1→2→3→4	88
		1→2→4→3	80
		<b>1→3→2→4</b>	<b>56</b>
		1→3→4→2	64
		1→4→2→3	64
		1→4→3→2	80

Hesaplama işlemini gerçekleştirirken; Muds matkap ucunun değiştirilme süresini saniye cinsinden belirtir. Mhs ise matkap ucunun 1mm mesafeyi kat etme süresi olarak sn/mm cinsinden belirlenmiştir.

#### 4.2. KULLANILABİLECEK MEVCUT OPTİMİZASYON YÖNTEMLERİ

Optimizasyon için ilk olarak “Brute Force” yöntemiyle muhtemel bütün çözüm dizilimleri tek tek oluşturulmuş, oluşturulan bu dizilimlere ait maliyetler hesaplanarak en kısa olanı seçilmeye çalışılmıştır. Bu yöntemde,  $n$  adet delik için  $n!$  adet varyasyon bulunmaktadır. Bu da delik sayısının artması ile sonuca ulaşmanın ne kadar zor olabileceğini göstermiştir.

Daha sonra, her yüzey kendi içerisinde mevcut yöntemlerden bazıları kullanılarak optimizasyona tabi tutulmuş, ardından her yüzeyde elde edilen optimum süreye ait dizilimler birbirine kaynaştırılarak, profile ait üç yüzey için sonuç dizilimi elde edilmeye çalışılmıştır. Bu yaklaşım hızlı ve basit çözüm üretmesine karşılık tam anlamıyla en kısa sürenin elde edilemeyeceği görülmüştür. Daha açık bir ifadeyle, yüzeylere ait dizilimlerin birleştirilmesiyle elde edilen çözüm diziliminde, a yüzeyi

için matkap ucu deęişimi gerçekteşmeden b yüzeyinde delme işlemleri gerçekteşirilmesi yerine, bazı durumlarda a yüzeyinde matkap ucu deęiştirilerek dięer delme işleminin yapılması ve daha sonra b yüzeyindeki işleme devam edilmesi daha mantıklı bir sonuç verebileceęi fark edilmiştir.

Yapılan bu denemelerin ardından, her yüzey sanki tek bir yüzeymiş gibi bütün olarak deęerlendirilmesi ve yapılacak optimizasyon işleminin, her yüzeye aynı anda ve bir kerede uygulanmasının daha kesin sonuç vereceęi anlaşılmıştır. Bu şekilde yapılacak hesaplamalarda, ayrı yüzeylerde oluşacak maliyetlerin ortak deęerlendirilmesi karmaşaya neden olabilmektedir. Çünkü her yüzey için  $x$  koordinatları ortak iken,  $y$  koordinatları birbirinden bağımsız ve delme ucu kalınlıkları birbirinden farklıdır. Mevcut yöntemler arasında, bu karmaşaya çözüm üretebilen bir algoritma ile karşılaşılma ve özel bir yöntem geliştirilme ihtiyacı görülmüştür. Geliştirilecek uygulamada, yüzeylerin  $x$  koordinatında kısmen bağımlı, dięer parametrelerde birbirinden bağımsız olduęu dikkate alınarak yeni bir yöntem geliştirilmiştir. Geliştirilen yöntem ile Seyahat Eden Satıcı problemi arasındaki benzerlik daha önce de belirtilmiştir.

Seyahat Eden Satıcı Problemi ile ilgili olarak çok farklı çözüm yöntemleri mevcuttur. Bunlar arasından çalışmamızda kullanılabilir üç yöntem seçilmiş ve tek yüzeyde uygulamaları geliştirilerek denemeler yapılmıştır.

Bu yöntemlerden birincisi, satır ve sütunların kesişimlerinde, ilgili satır ve sütun numaralarına ait delikler arası maliyetlerin gösterildięi iki boyutlu dizi ve bu iki boyutlu dizideki maliyetler üzerinden minimum tamamlanma süresine ait dizilime ulaşmaya çalışan Yakın Komşu Algoritmasıdır.

İkincisi ise tek boyutlu bir dizi içerisinde tutulan delme sıralamasında, ikili olarak deęişimler yardımıyla oluşan yeni dizilimlerin maliyetlerini karşılaştırarak en uygun dizilime ulaşmaya çalışan 2-Opt Exchange algoritmasıdır.

Üçüncü ve son yöntem ise, 2-Opt Exchange algoritması ile benzerlik gösteren, fakat karşılıklı olarak yer değiştirmek yerine dizilimdeki her bir indis içerisinde tutulan delik numaralarının sırayla kaydırılmasını tanımlayan Kaydırma algoritmasıdır.

Her bir uygulama ile ilgili olarak gerçekleştirilmek istenen üç yüzeydeki optimizasyon uygulamasına ait yapılan çalışmalar ve algoritmaların çalışma mantığı aşağıda ayrıntılı olarak açıklanmıştır.

#### 4.2.1. Yöntem 1 – Yakın Komşu Algoritması

Bu algoritma kullanılarak yapılacak hesaplamalara örnek olması amacıyla Şekil 4.1’de verilen örnek çizimde yer alan delikler ve bu delikler arası maliyetler kullanılmıştır. Bu maliyetler tek tek hesaplanarak Çizelge 4.2’de belirtilen şekilde 2x2 bir diziye aktarılmıştır.

Çizelge 4.2. Yakın Komşu algoritması için delikler arası maliyetleri gösterir tablo.

	1	2	3	4
1	-	22	14	12
2	22	-	12	14
3	14	12	-	22
4	12	14	22	-

$$C(1,2) = Mhs * |1,2| + Mds(1,2) = 1sn/mm * 10 mm + 12 sn = 22 sn$$

$$C(1,3) = Mhs * |1,3| + Mds(1,3) = 1sn/mm * 2 mm + 12 sn = 14 sn$$

$$C(1,4) = Mhs * |1,4| + Mds(1,4) = 1sn/mm * (10 + 2) mm + 0 sn = 12 sn$$

$$C(2,3) = Mhs * |2,3| + Mds(2,3) = 1sn/mm * (10 + 2) mm + 0 sn = 12 sn$$

$$C(2,4) = Mhs * |2,4| + Mds(2,4) = 1sn/mm * 2 mm + 12 sn = 14 sn$$

$$C(3,4) = Mhs * |3,4| + Mds(3,4) = 1sn/mm * 10 mm + 12 sn = 22 sn$$

Genel olarak bu algoritma en düşük maliyete ulaşmak üzere, daima en yakında bulunan bir sonraki komşuyu referans alır. Bir sonraki dizilim değeri, daha önce ziyaret edilmeyen en kısa mesafeli rota elemanıdır. 4 delikli örnek üzerinde bu algoritmanın işleyişi aşağıda açıklanmıştır.

Birinci aşamada, Çizelge 4.3'te gösterildiği gibi başlangıç deliği olan 1 numaralı delik için, başlangıç satırı 1 kabul edilir ve bu satırdaki en kısa süreye ait sütun, sonraki dizilim değeri olarak alınır.

Çizelge 4.3. Yakın Komşu algoritması birinci aşama.

	1	2	3	4
1	-	22	14	12
2	22	-	12	14
3	14	12	-	22
4	12	14	22	-

İkinci aşamada incelenecek satır olarak, bir önceki aşamada sıradaki dizilim değeri olarak bulunan sütun numarası (4) kabul edilir. Bu satırda bulunan en kısa süreye ait sütun Çizelge 4.4'te gösterildiği gibi belirlenir.

Çizelge 4.4. Yakın Komşu algoritması ikinci aşama.

	1	2	3	4
1	-	22	14	12
2	22	-	12	14
3	14	12	-	22
4	12	14	22	-

Burada en kısa süre 12 sn olan 1. Sütun olmasına karşılık, sütun numarasının işaret ettiği bu delik daha önce kullanıldığından dolayı 2 numaralı sütun en kısa süre olarak kabul edilmiştir. İşleme 2 numaralı satırdan devam edilmektedir.

Çizelge 4.5'te görüldüğü üzere, bu satırda da son olarak daha önce kullanılmamış olan 3 numaralı sütun, sıradaki dizilim değeri olarak kabul edilmiş ve dizilim tamamlanmıştır. Sonuç olarak bulunan optimum delme sırasını belirten dizilim 1→4→2→3 olarak bulunmuş ve toplam maliyet  $12+14+12=38$  sn olarak hesaplanmıştır.

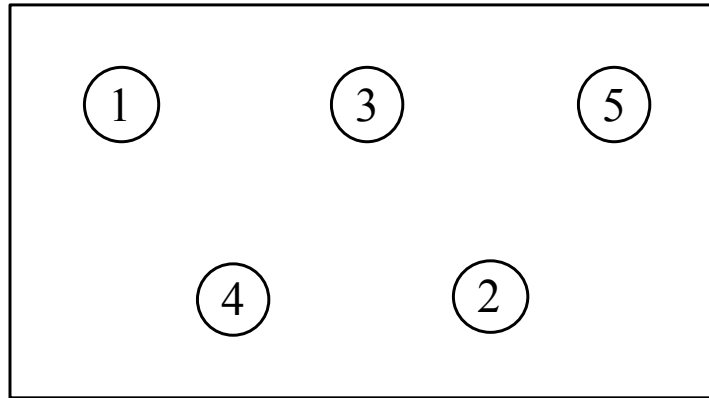


Çizelge 4.5. Yakın Komşu algoritması üçüncü aşama.

	1	2	3	4
1	-	22	14	12
2	22	-	12	14
3	14	12	-	22
4	12	14	22	-

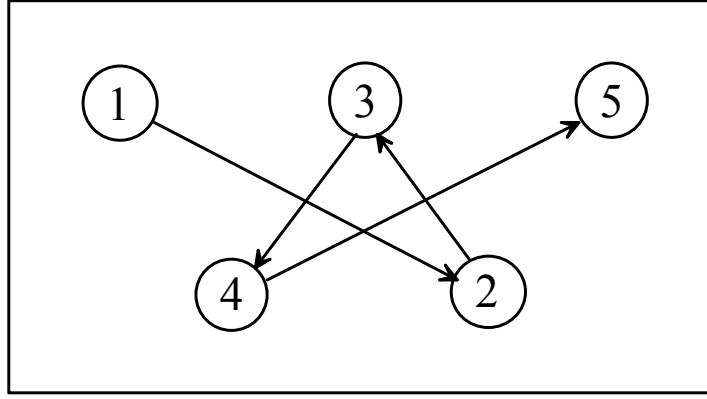
#### 4.2.2. Yöntem 2 – 2-Opt Exchange Algoritması

Bu yöntemde delme sırası diziliminde yapılan değişiklikler sonucunda oluşan maliyetlerin tekrar tekrar hesaplanması, yapılan hesaplamalar ile elde edilen yeni maliyetin daha önceden hesaplanarlardan daha düşük olduğu durumlarda yeni çözüm olarak kabul edilmesi şeklinde işlemektedir. Daha fazla maliyetin düşürülmesini sağlayacak değişiklik yapılmadığında inceleme işlemi sonlandırılmakta ve son bulunan dizilim optimum dizilim olarak kabul edilmektedir.



Şekil 4.2. 2-Opt Exchange algoritması için örnek profil görünümü ve delik dizilimi.

Şekil 4.2’de gösterildiği gibi bir profil yüzeyi örnek olarak kabul edelim. Başlangıç dizilimi için veritabanı tablosuna eklenme sırası olarak kabul edilen 1→2→3→4→5 delme dizilimi, Şekil 4.3’te belirtildiği gibi gösterilecektir. Bu gösterimde, gereksiz gidiş-gelişten doğan aşırı maliyetler belirlenecek, eğer bu rotayla ilgili olarak yapılacak bir değişim toplam maliyeti düşürürse değişiklik kabul edilecektir.



Şekil 4.3. Kabul edilen başlangıç dizilimini gösteren delme sırası.

Delme sırasına ait dizilim, Çizelge 4.6’da gösterildiği gibi *optimum* isimli tek boyutlu bir dizi içerisinde tutulmaktadır. Bu dizilimde, sırasıyla ikili olarak indis numaralarına karşılık gelen delik numaraları yer değiştirilerek oluşan yeni toplam maliyet hesaplanmaktadır.

Çizelge 4.6. Delme işlemi dizilimini tutan dizi değişkeni içeriği.

İndis No	0	1	2	3	4
Delik No	1	2	3	4	5

Aşağıda 1 numaralı indiste tutulan delik numarası ile 2 numaralı indiste tutulan delik numarasının değişimi sonucu oluşan yeni dizilime örnek gösterilmiştir.

1→~~2~~→~~3~~→4→5 (mevcut dizilim) | 1→3→2→4→5 (yeni dizilim)

Yapılan bu değişikliğin ardından maliyet azaldığı takdirde, değişikliğin yapıldığının yazılım tarafından anlaşılması ve kontrol döngüsünü tekrar çalıştırması amacıyla bir değişken değeri set edilir. Yeni oluşan dizilim ise sonuç dizilimi olarak kabul edilir.

Eğer maliyet azalmadıysa, yapılan değişiklik geri alınarak, sıradaki kontrol işlemi gerçekleştirmek üzere 2 numaralı indiste tutulan delik numarası ile 4 numaralı indiste tutulan delik numarası için değişim işlemi gerçekleştirilerek oluşan maliyet kontrol edilir.

1→2→3→4→5 (mevcut dizilim) | 1→4→3→2→5 (yeni dizilim)

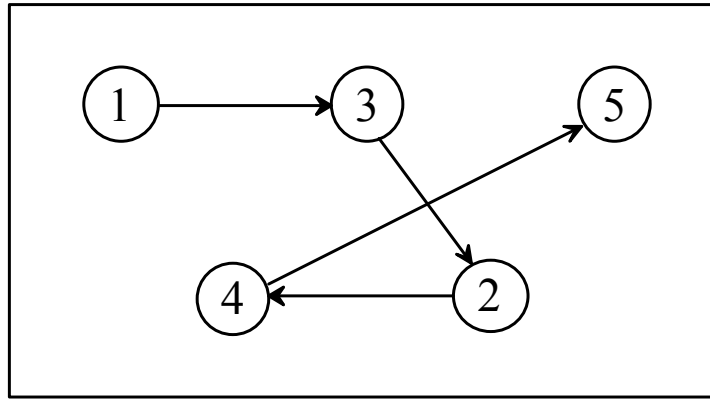
Benzer şekilde, yeni dizilimindeki toplam maliyet azaldıysa değişiklik kabul edilerek kontrolün tekrar çalışabilmesi için kontrol değişkeni set edilir.

Örnek üzerinde işlemler uygulanarak dizilimin nasıl gerçekleştirildiğini görmek üzere, işlem basamakları aşağıdaki gibi gerçekleşecektir. Burada maliyet hesabı için delik çapları aynı kabul edilmiş, sadece hareket mesafeleri dikkate alınmıştır.

1. işlem basamağı:

1→2→3→4→5 (mevcut dizilim) | 1→3→2→4→5 (yeni dizilimi)

Oluşacak yeni maliyet, daha önceki maliyete göre düşük olacağından yeni dizilim kabul edilir ve yeni değişikliklerin kontrol edilmesine devam edilir. Yeni delme işlemi sırası, Şekil 4.4'te gösterilmiştir.



Şekil 4.4. 2-Opt Exchange algoritması birinci aşama sonucundaki yeni delme sırası.

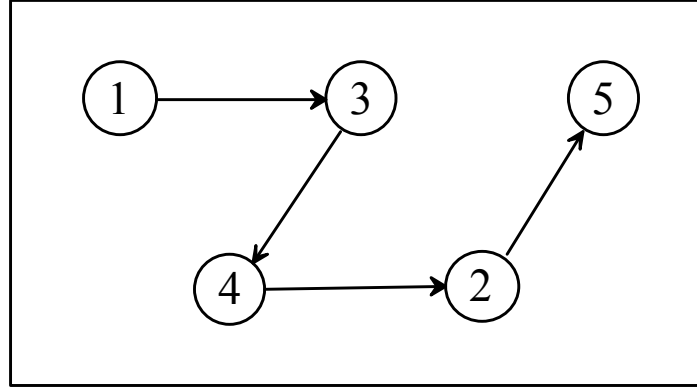
2. İşlem basamağı: Yeni dizilim üzerinden tekrar değerlendirmeye devam edilir.

1→3→2→4→5 (mevcut dizilim) | 1→2→3→4→5 (yeni dizilim)

1→3→2→4→5 (mevcut dizilim) | 1→4→2→3→5 (yeni dizilim)

1→3→2→4→5 (mevcut dizilim) | 1→3→4→2→5 (yeni dizilim)

Oluşan son dizilimde, toplam maliyet azalacak ve yeni dizilim olarak kabul edilecektir. Değişiklik yapıldığından dolayı set edilen kontrol değişkeni sebebiyle, aynı kontroller tekrar çalıştırılacaktır. Oluşan yeni dizilim Şekil 4.5'te gösterilmiştir.

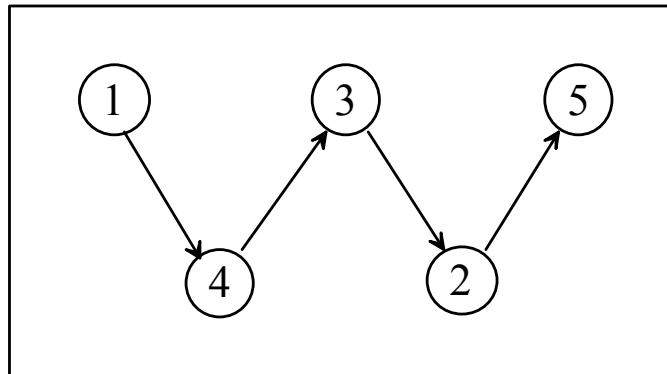


Şekil 4.5. 2-Opt Exchange algoritması ikinci aşama sonucundaki yeni delme sırası.

### 3. İşlem Basamağı:

$1 \rightarrow \underline{3} \rightarrow \underline{4} \rightarrow 2 \rightarrow 5$  (mevcut dizilim) |  $1 \rightarrow \underline{4} \rightarrow \underline{3} \rightarrow 2 \rightarrow 5$  (yeni dizilim)

Yapılan bu değişiklik ile oluşabilecek minimum maliyet elde edilmiş olacak, fakat kontrol değişkeni daha önce yapılan değişim işlemi sonucunda set edildiğinden dolayı her bir indis için değişim kontrol işlemi tekrar yapılacaktır. Yapılacak kontrolde maliyette azalmaya neden olacak bir değişiklik olmayacağından dolayı sonuç dizilimi bulunmuş sayılarak değerlendirme sona erecektir. Kabul edilen minimum delme maliyetine ait  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$  dizilimi, Şekil 18'de gösterilmiştir.



Şekil 4.6. 2-Opt Exchange algoritması ile elde edilen optimum dizilim.

### 4.2.3. Yöntem 3 – Kaydırma Algoritması

Bu yöntem 2-Opt Exchange algoritması ile benzerlik göstermesine karşılık iki delik numarasının karşılıklı değişimi yerine sadece bir indis içerisinde bulunan delik numarasının yeri değiştirilmektedir. Yapılan değişiklik işlemleri aşağıda belirtildiği şekilde gerçekleştirilmektedir.

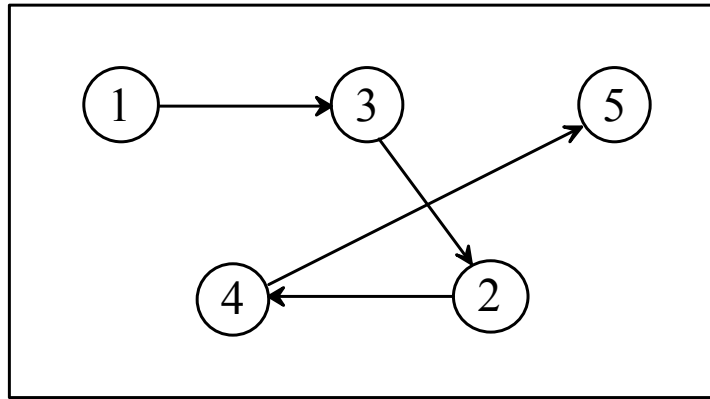
1→**2**→3→4→5 (mevcut dizilim) |      2→1→3→4→5 (yeni dizilim)

Yapılan bu değişiklik sonucunda oluşan dizilime ait toplam maliyet daha önce kabul edilen dizilime ait toplam maliyetten daha düşük ise yapılan değişiklik kabul edilir. Aksi takdirde değişiklik reddedilerek yeni bir değişiklik işlemi ile devam edilir. Şekil 4.3'te belirtilen örnek çizime ait başlangıç dizilimi, 2-Opt Exchange Algoritmasında olduğu gibi kabul edilerek optimizasyon işlemini örnek üzerinde açıklamaya çalışalım.

#### 1. işlem basamağı:

1→**2**→3→4→5 (mevcut dizilim) |      2→1→3→4→5 (yeni dizilim)

1→2→**3**→4→5 (mevcut dizilim) |      1→3→2→4→5 (yeni dizilim)



Şekil 4.7. Kaydırma Algoritması birinci aşama sonucundaki oluşan dizilim.

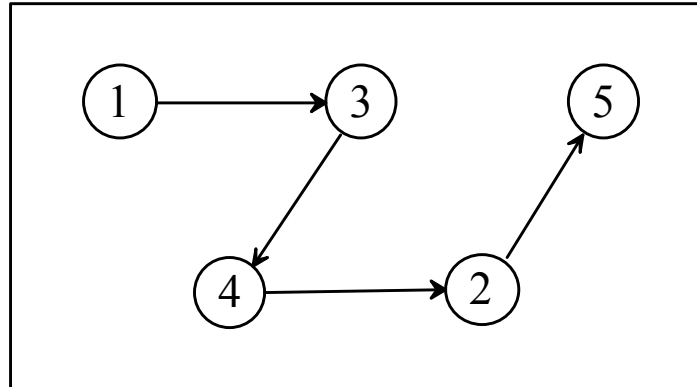
Oluşan son dizilim ile hesaplanan toplam maliyet, daha önce kabul edilen toplam maliyetten daha düşük olacağı için yeni dizilim olarak kabul edilir. Değişikliğin

gerçekleştiğini ve kontrol işleminin tekrar çalıştırılması gerektiğini belirtmek amacıyla yine bir kontrol değişkeni set edilir. Oluşan yeni dizilim Şekil 4.7’de gösterilmiştir. Yeni dizilim üzerinden kontrol işlemleri tekrar başlatılarak yeni dizilim aranır.

2. işlem basamağı:

$1 \rightarrow \underline{3} \rightarrow 2 \rightarrow 4 \rightarrow 5$ (mevcut dizilim)		$\underline{3} \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ (yeni dizilim)
$1 \rightarrow 3 \rightarrow \underline{2} \rightarrow 4 \rightarrow 5$ (mevcut dizilim)		$1 \rightarrow \underline{2} \rightarrow 3 \rightarrow 4 \rightarrow 5$ (yeni dizilim)
$1 \rightarrow 3 \rightarrow 2 \rightarrow \underline{4} \rightarrow 5$ (mevcut dizilim)		$1 \rightarrow 3 \rightarrow \underline{4} \rightarrow 2 \rightarrow 5$ (yeni dizilim)

Yapılan kontroller neticesinde elde edilen son dizilim daha düşük bir toplam maliyet sunmaktadır. Dolayısıyla yeni dizilim olarak bu kabul edilir ve yapılan değişiklik neticesinde oluşan dizilim Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Kaydırma algoritması ikinci aşama sonucunda oluşan dizilim.

3. işlem basamağı:

$1 \rightarrow \underline{3} \rightarrow 4 \rightarrow 2 \rightarrow 5$ (mevcut dizilim)		$\underline{3} \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 5$ (yeni dizilim)
$1 \rightarrow 3 \rightarrow \underline{4} \rightarrow 2 \rightarrow 5$ (mevcut dizilim)		$1 \rightarrow \underline{4} \rightarrow 3 \rightarrow 2 \rightarrow 5$ (yeni dizilim)

Yapılan bu değişiklik ile amaçlanan en düşük maliyete sahip, daha önce tespit edilen ve Şekil 4.6’da gösterilen dizilim elde edilmiş olur.

### 4.3. ÜÇ EKSENLİ OPTİMİZASYON UYGULAMASI

Çalışma kapsamında bulunmak istenen optimum süreyi hesaplamak amacıyla, gerçekleştirilen denemeler sonucunda 2-Opt Exchange ve Kaydırma Algoritmaları, üç eksenle gerçekleştirilecek minimum maliyet hesaplamalarına uygun olduğu görülmüş ve bu algoritmaların kombinasyonu geliştirilen uygulamaya adapte edilmiştir. İnceleme işlemi, daha düşük maliyetli yeni bir dizilim elde edilebildiği sürece tekrarlanmaktadır.

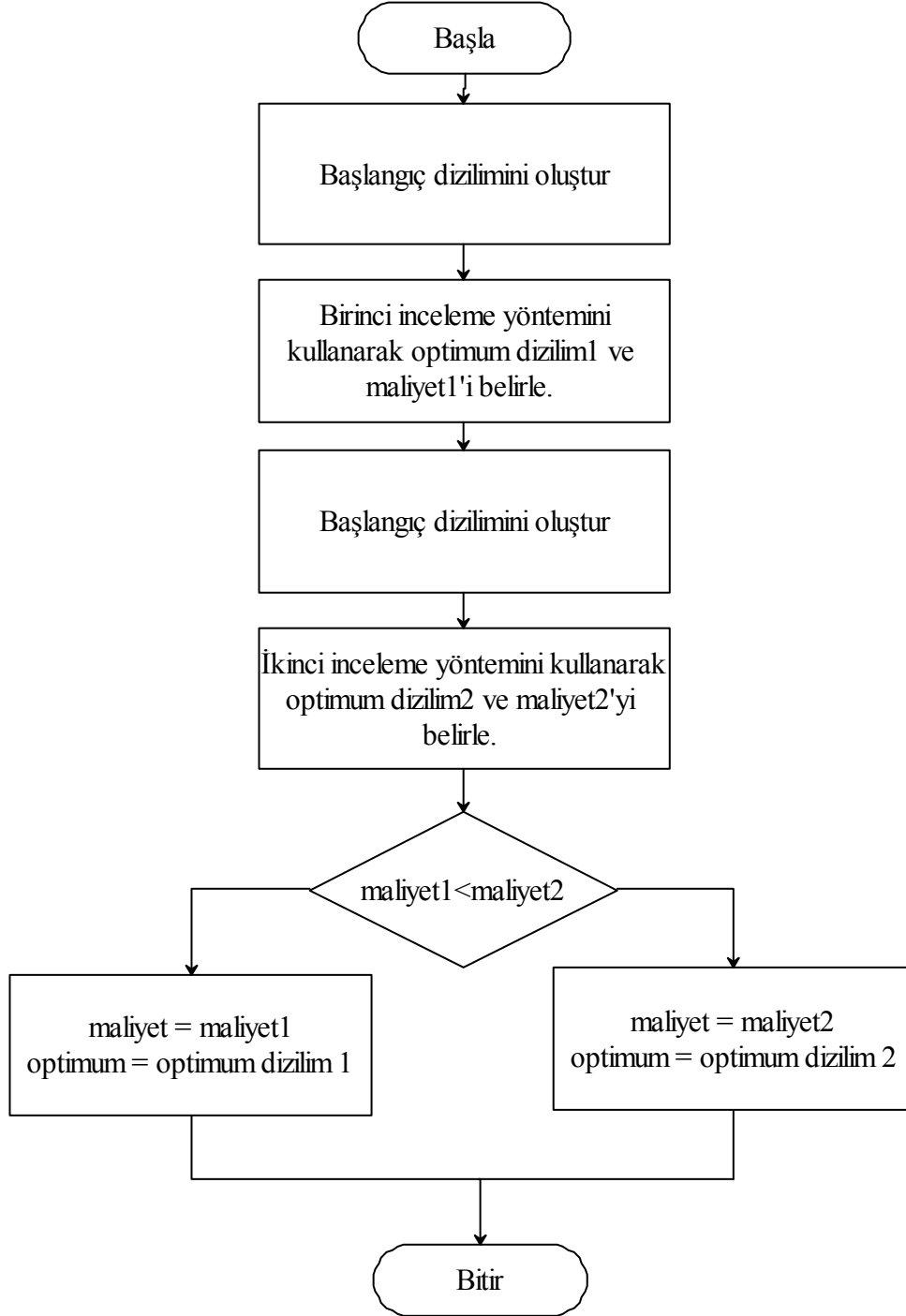
Algoritmaların kombinasyonu kullanılırken ilk uygulanacak olan algoritmanın hangisi olduğu, bazı durumlarda sonucu etkilemektedir. Örneğin önce Kaydırma algoritması ve daha sonra 2-Opt Exchange algoritması kullanıldığında elde edilen delme işlemi dizilimi ve toplam maliyet, önce 2-Opt Exchange algoritması ve daha sonra Kaydırma algoritması kullanılarak elde edilen delme işlemi dizilimi ve toplam maliyet ile farklılık gösterebilmektedir. Dolayısıyla her iki yöntem farklı toplam maliyet ve delme işlemi dizilimlerinde son bulabilmektedir. Bu durumun ortadan kaldırılabilmesi için yazılım tarafından her iki şekilde de inceleme yapılmalı ve daha düşük toplam maliyete ait dizilim optimum olarak kabul edilmelidir.

Şekil Ek B.9’da verilen 50 delikli çizim örneği üzerinde, bahsedilen algoritmaların değişik kombinasyonlarda kullanımlarına ait sonuçlar Çizelge 4.7’de gösterilmiştir.

Çizelge 4.7. Optimizasyon yöntemleri öncelik sıralamasının sonuçlara etkisi.

2-Opt Exchange Algoritması	n-Opt Exchange Algoritması	Kaydırma Algoritması	Deneme Sayısı	Değişiklik Sayısı	Toplam Maliyet (sn)
1	-	-	7105	38	1166,921
-	1	-	37904	45	725,572
1	2	-	29634	53	885,622
2	1	-	15405	47	726,275
-	-	1	3675	35	919,281
1	2	3	61816	75	736,83
2	3	1	42834	63	749,76

Çizelge 4.7’de, algoritmalara ait sütunlarda belirtilen sayılar, ilgili algoritmanın kullanım önceliğini gösteren sıra numaralarıdır. Görüldüğü üzere, her bir algoritma ile elde edilen sonuç ile, bu algoritmaların değişik sıralamalarla verdiği sonuçlar büyük oranda farklılık göstermektedir.



Şekil 4.9. Genel haliyle gerçekleştirilecek olan optimizasyon işlemi basamakları.



Çalışma kapsamında 2-Opt Exchange algoritması geliştirilerek, karşılıklı birden fazla elemanın aynı anda değiştirilebilmesini sağlayan değişik bir yöntem daha kullanılmış ve bu yöntem n-Opt Exchange olarak ifade edilmiştir. Burada  $n$  ifadesi sabit belirli bir sayıyı göstermemekte, alabileceği değerler 2 ile toplam delik sayısının yarısı arasındadır. Bu yöntemin kullanımı daha sonraki bölümlerde detaylı olarak açıklanmıştır. Genel olarak delme işlemi dizilimi üzerine yazılım tarafından gerçekleştirilen optimizasyon işleminde, Şekil 4.9’da gösterilen işlem basamakları kullanılmaktadır.

Belirtilen işlem basamakları iki temel işlem adımından oluşmaktadır. Birincisi optimizasyon işlemi için yeni dizilimlerin elde edilmesi, ikincisi ise elde edilen dizilimlerin oluşturduğu toplam maliyetin hesaplanmasıdır. Her iki işlem için iki ayrı sınıf oluşturulmuş ve bu iki sınıfın birbiriyle etkileşimi ile optimizasyon işlemi gerçekleştirilmiştir.

#### **4.3.1. Maliyetin Hesaplanması**

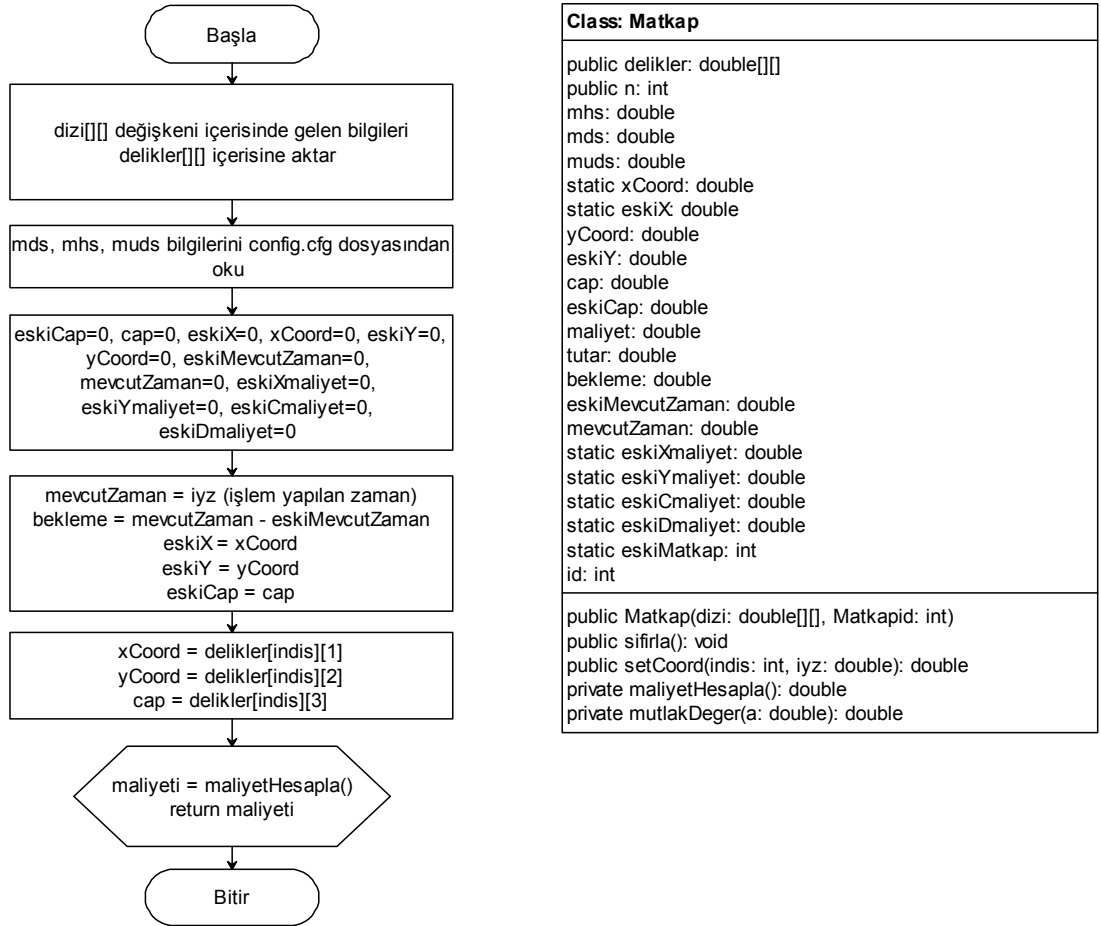
Optimizasyon işleminde, matkap uçlarının her birini temsil etmek amacıyla bir “Matkap” sınıfı oluşturulmuş ve bu matkap sınıfı kullanılarak her yüzeyde delme işlemini gerçekleştirecek matkap ucunu temsil etmek üzere ayrı ayrı  $m_1$ ,  $m_2$  ve  $m_3$  isimlerinde toplam 3 adet Matkap nesnesi tanımlanmıştır. Bu nesnelerin her birinde ilgili yüzeyde gerçekleştirilen delme işlemleri için  $x$  ve  $y$  koordinat bilgileri, bu koordinat bilgilerinin bir önceki delme işlemine ait eski değerleri, mevcut ve eski matkap ucu kalınlığı bilgisi, ilgili yüzeyde bulunan deliklerin koordinat ve matkap ucu kalınlığı bilgileri gibi bilgiler tutulmaktadır. Ayrıca, her bir matkap nesnesi için bir önce yapılan delme işlemine ait eksen boyunca gerçekleşen hareket, delme ve değişim gibi maliyetleri tutan değişkenler static olarak tanımlanmıştır.

Optimizasyon işleminin en kritik noktası,  $x$  koordinatında birliktelik sağlamak olduğundan, yine bu bilgiler de “static” olarak tanımlanmıştır. Bu şekilde, tanımlanan her bir matkap nesnesi için  $x$  koordinatı bilgisi değiştiğinde, aynı sınıftan üretilen diğer matkap nesneleri için de  $x$  koordinatı otomatik olarak değiştirilmiş olacaktır.

Matkap sınıfı tanımlanırken;

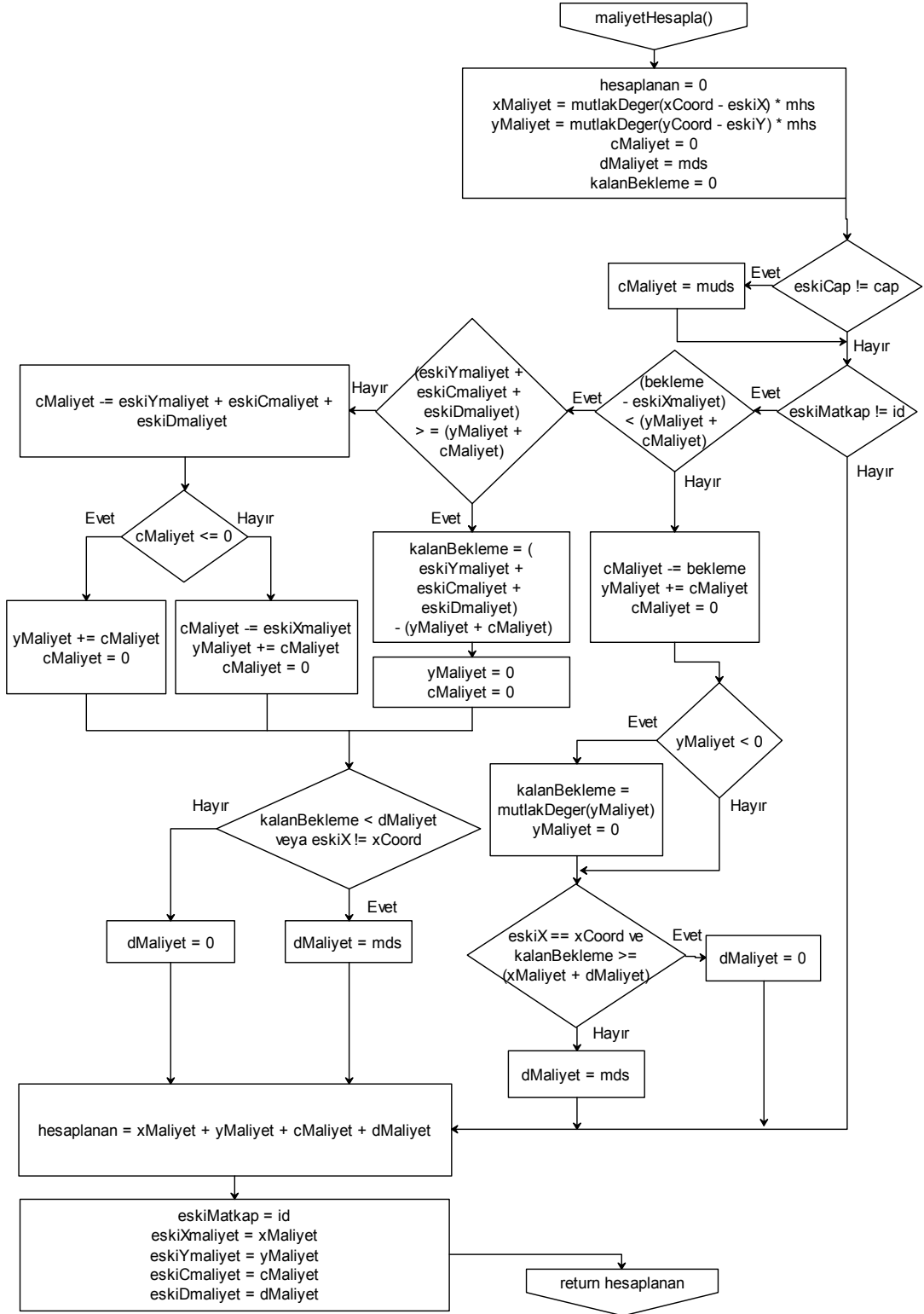
- Matkap(double dizi[ ][ ], String ad) - Constructor
- Sifirla( )
- setCoordinate(int a, double mevcutSure)
- delikMaliyeti( )
- getLength( )
- mutlakDeger(double a)

gibi metodlar kullanılmıştır. Şekil 4.10'da, tanımlanan matkap nesnesinde yer alan metod ve parametreleri gösteren akış diyagramı verilmiştir.



Şekil 4.10. Matkap sınıfı tarafından gerçekleştirilen işlemlere ait akış diyagramı.

Matkap sınıfı ile tanımlanan metodlar, kullanım amaçları ve kullanım şekilleri aşağıda açıklanmıştır.



Şekil 4.11. Matkap sınıfı `delikMaliyeti()` metodu akış diyagramı.

Matkap(double dizi[ ] [ ], int Matkapid) metodu, sınıfın yapıcı metodudur. Nesneye dönüştürülme aşamasında çalıştırılmak istenen komut satırlarını tutar. Burada, gelen dizi değişkeni iki boyutlu olup, birinci boyut ilgili deliklerin indis numarasını, ikinci boyut ise bu deliklerin x, y koordinatları, delik çapı gibi parametrik değerlerini tutar. id değişkeni ise, maliyetler hesaplanırken bir önce iş yapan matkap yüzeyinin hangisi olduğunu tanımlamak amacıyla kullanılmıştır.

sifirla( ) metodu, matkabın işlem yapılmamış sayılan başlangıç noktasına geri döndürmek amacıyla kullanılır. Bu metoda özellikle, değişik varyasyon hesaplamalarının yapıldığı aşamada ihtiyaç duyulmaktadır. Metod, çağırıldığı noktaya herhangi bir değer döndürmez.

delikMaliyet( ) metodu, optimizasyon işleminin önemli noktalarından biridir. Bu metod yardımıyla, daha önce işlem yapılan delik ile mevcut delik arasında geçen zaman maliyeti, mevcut deliğin yer aldığı yüzeye ait matkap nesnesi tarafından, yüzeyler arası eş zamanlı maliyetler dikkate alınarak hesaplanır. Yapılan hesaplama işlemi, Şekil 4.11’de gösterilmiştir. Metod, çağırıldığı noktaya double veri türünde delik maliyet değerini döndürür.

setCoordinate(int indis, double iyz) metodunda gelen *indis* bilgisi, ilgili matkap nesnesi içerisinde tutulan, bu yüzeye ait deliklerden istenen indis numarasındaki deliği temsil eder. Dolayısıyla, belirtilen indisli delik, işlem yapılmak istenen delik olarak tanımlanmaktadır. *iyz* ise, maliyet hesabı için, ilgili matkap nesnesinin en son işlem yaptığı andan şu ana kadar geçen süreyi belirtir. Bu süre, matkap ucu değişimi için yeterli olduğu durumlarda, matkap ucu değişim süresinin maliyet olarak hesaba katılmaması olarak yorumlanır. Delinmesi gereken delik bu methodla belirlendikten ve koordinat bilgileri tanımlandıktan sonra *delikMaliyet()* metodu buradan çağırılır. Metod, çağırıldığı noktaya herhangi bir değer döndürmez.

getLength( ) metodu, bulunduğu matkap nesnesinde ilgili yüzeye ait delik sayısının öğrenilmesi amacıyla oluşturulmuştur.

mutlakDeger(double a) metodu ise, parametre olarak gelen double veri türündeki a değişkeninin tuttuğu değeri daima pozitif olarak geri döndürmektedir.

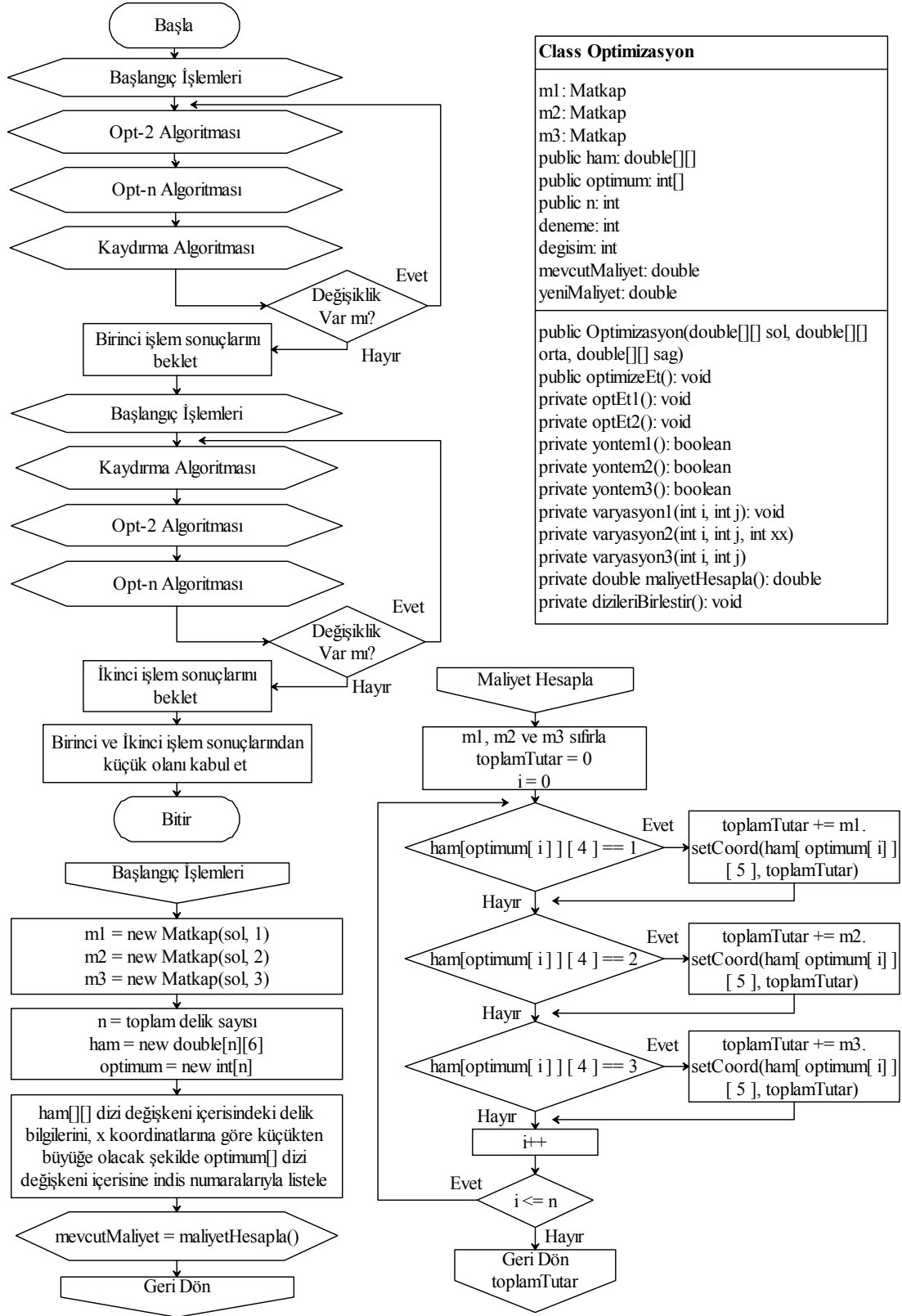
#### 4.3.2. Optimizasyon İşleminin Gerçekleştirilmesi

Optimizasyon işleminin gerçekleştirilmesi amacıyla kullanılacak metodlar optimizasyon sınıfı içerisinde tanımlanmış ve aşağıda listelenmiştir. Bu sınıfın kullanım amacı, farklı dizilimlerin elde edilmesi ve elde edilen dizilimlere ait maliyetler hesaplanarak minimum maliyeti sağlayan dizilime ulaşmaktır.

- Optimizasyon(double[] sol, double[] orta, double[] sag) –Constructor
- dizileriBirlestir()
- optimizeEt()
- optEt1(), optEt2()
- varyasyon1(), varyasyon2(), varyasyon3()
- sureHesapla()

Optimizasyon(double[] sol, double[] orta, double[] sag) metodu, yapıcı metod olup sınıfın nesneye dönüştürülmesi aşamasında çalıştırılması istenen komut satırlarının bulunduğu metoddur.

Burada, her bir yüzey için matkap sınıfı bir nesneye dönüştürülür ve bu yüzeylere ait delik bilgileri ilgili nesnelere aktarılır. Ayrıca sonuç dizilimini temsil edecek olan *optimum[]* dizi değişkeni, toplam delik adedi boyutunda oluşturulur ve *dizileriBirlestir()* metodu çağırılarak, *x* koordinatına göre küçükten büyüğe sıralanarak başlangıç delme dizilimi oluşturulur. Yapıcı metod olmasından dolayı, çağırıldığı noktaya herhangi bir değer döndürmez. Sınıfa ait genel algoritma Şekil 4.12’de gösterilmiştir.



Şekil 4.12. Optimizasyon işleminde farklı dizilim ve toplam maliyetlerin elde edilmesi.

dizileriBirlestir() metodunda amaç, bir başlangıç dizilimi oluşturmaktır. Bu işlem, bütün delikler dikkate alınarak, deliklerin ortak hareketin sağlandığı  $x$  koordinat bilgilerine göre küçükten büyüğe sıralanması şeklinde olmaktadır. Bu şekilde bir yol izlenmesinin nedeni, delme işlemine başlangıç olarak ilk  $x$  koordinatının kullanılmasının öngörülmesinden dolayıdır. Optimizasyon işlemi sırasında, bir başka delik ile delme işlemine başlanması daha kısa sonuç vereceği hesaplanır ise bu öngörü ortadan kalkacak ve başlangıç olarak kabul edilen sonuç diziliminde yerini alacaktır. Metod, çağırıldığı noktaya herhangi bir değer döndürmez.

optimizeEt() metodu, optimizasyon işleminin kendisidir. Optimizasyon işlemine başlangıçta yapılacak ilk değişiklik işlemi, daha sonraki değişiklikleri ve dolayısıyla elde edilecek sonucu etkileyecektir. Burada iki farklı sıralamada optimizasyon algoritmaları uygulanmakta, en ideal sonucun elde edildiği yönteme ait değerler optimum delme dizilimi ve maliyeti olarak kabul edilmektedir. Daha açık bir ifadeyle, optEt1() metodu ve optEt2() metodu ile aşağıda açıklanacak yöntem neticesinde elde edilen en ideal sonucu kabul edilmesi için yapılacak kontroller bu metoda yer alır.

optEt1() ve optEt2() metodları, boolean türünde “tekrar” isimli değişken tarafından kontrol edilerek bir döngü yardımıyla sürekli olarak değişik dizilimler denemekte, yapılan her değişiklik sayesinde *tekrar* değişkenine “true” değeri atanarak döngünün tekrarlama sağlanmaktadır. Bu döngü içerisinde yer iki metod farklı önceliklerde ilgili algoritmaları sırayla tekrar tekrar çağırır.

optEt1() metodunda öncelikle 2-Opt Exchange algoritması sayesinde, karşılıklı olarak dizi içerisindeki elemanlar birerli olarak birbirleriyle yer değiştirir. Yapılan değişiklikler neticesinde daha düşük maliyete sahip yeni dizilimler elde edilemiyorsa, daha sonra 2-Opt Exchange algoritmasına benzer olan n-Opt Exchange algoritması çalıştırılır. Bu yöntemde, karşılıklı olarak birer elemandan daha fazla ve dizi eleman sayısının yarısından daha az olmak koşuluyla, yer değiştirecek eleman sayısı artırılarak bütün değişiklik denemeleri gerçekleştirilir. Örneğin;  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$  dizilimi için, ilk yapılan karşılıklı ikişer elemanın yer değiştirme işlemi sonucu elde edilen yeni dizilim  $3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 5$  şeklinde olmaktadır. Bu

şekilde yapılacak n-Opt Exchange deęişiklikleri sonucunda daha düşük maliyete sahip bir dizilim elde edilmezse, son olarak Kaydırma algoritması ile, her bir dizi elemanını için, sadece tek bir elemanın yer deęiştirmesi sağlanarak daha düşük maliyete sahip dizilim aranır. Yapılan bu kontroller sonucunda, her üç yöntemle de daha düşük maliyete sahip bir deęişiklik gerçekleşmedięi taktirde algoritmalar sonlandırılarak elde edilen dizilim ve bu dizilime ait maliyet daha sonra karşılaştırılmak üzere saklanır.

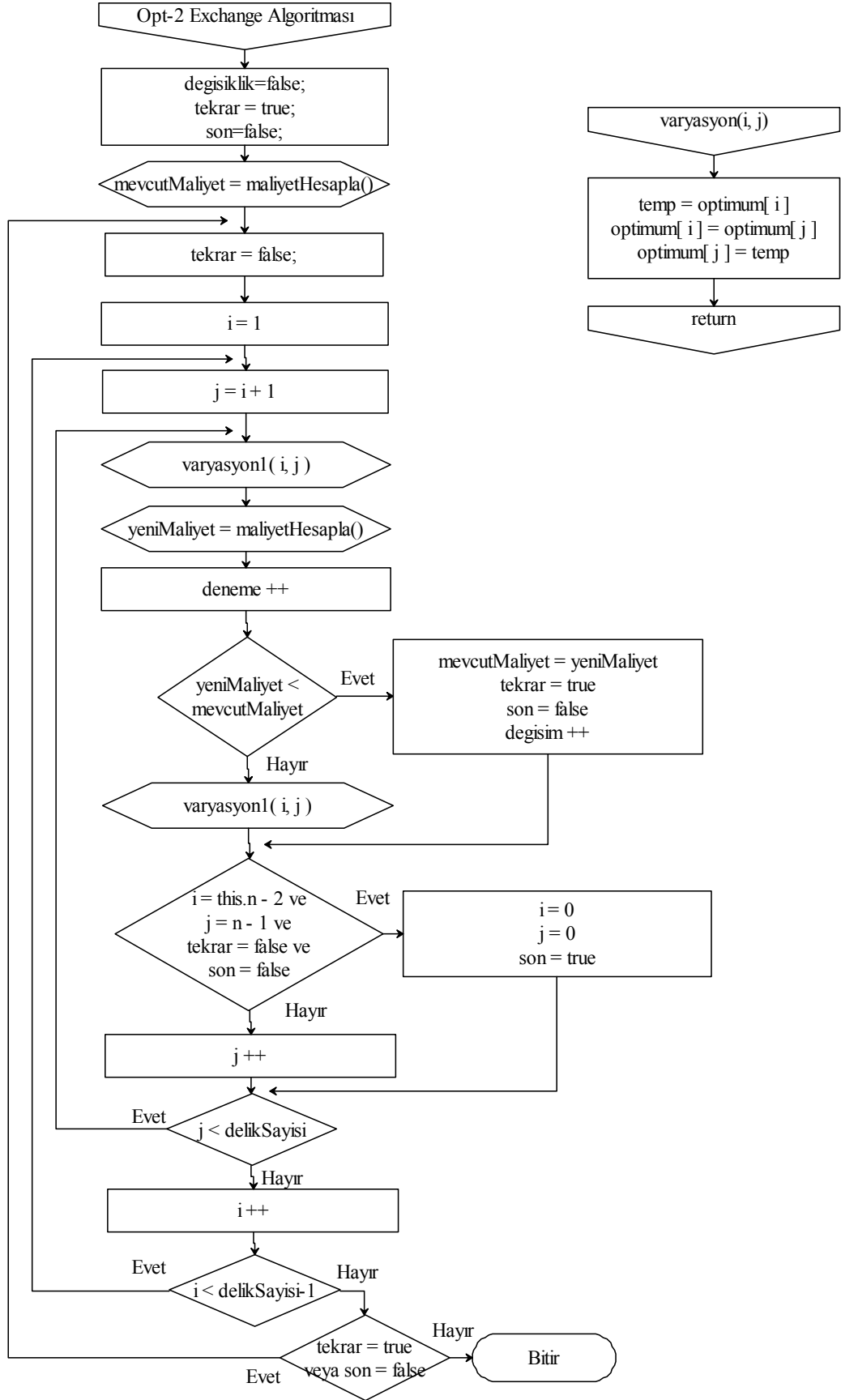
optEt2() metodunda ise, yukarıdaki işlemlerin aynısı, önce Kaydırma algoritması, daha sonra 2-Opt Exchange ve son olarak n-Opt Exchange algoritmaları sırasına uygun olarak başlangıç dizilimi ile tekrarlanır.

Son işlem basamaęı, hangi yöntem yardımıyla daha düşük maliyetle delme işleminin gerçekleştirileceęine karar vermektir. Daha düşük maliyete sahip delme dizilimi optimum dizilim olarak kabul edilir. Metod, çağırıldığı noktaya herhangi bir deęer döndürmez.

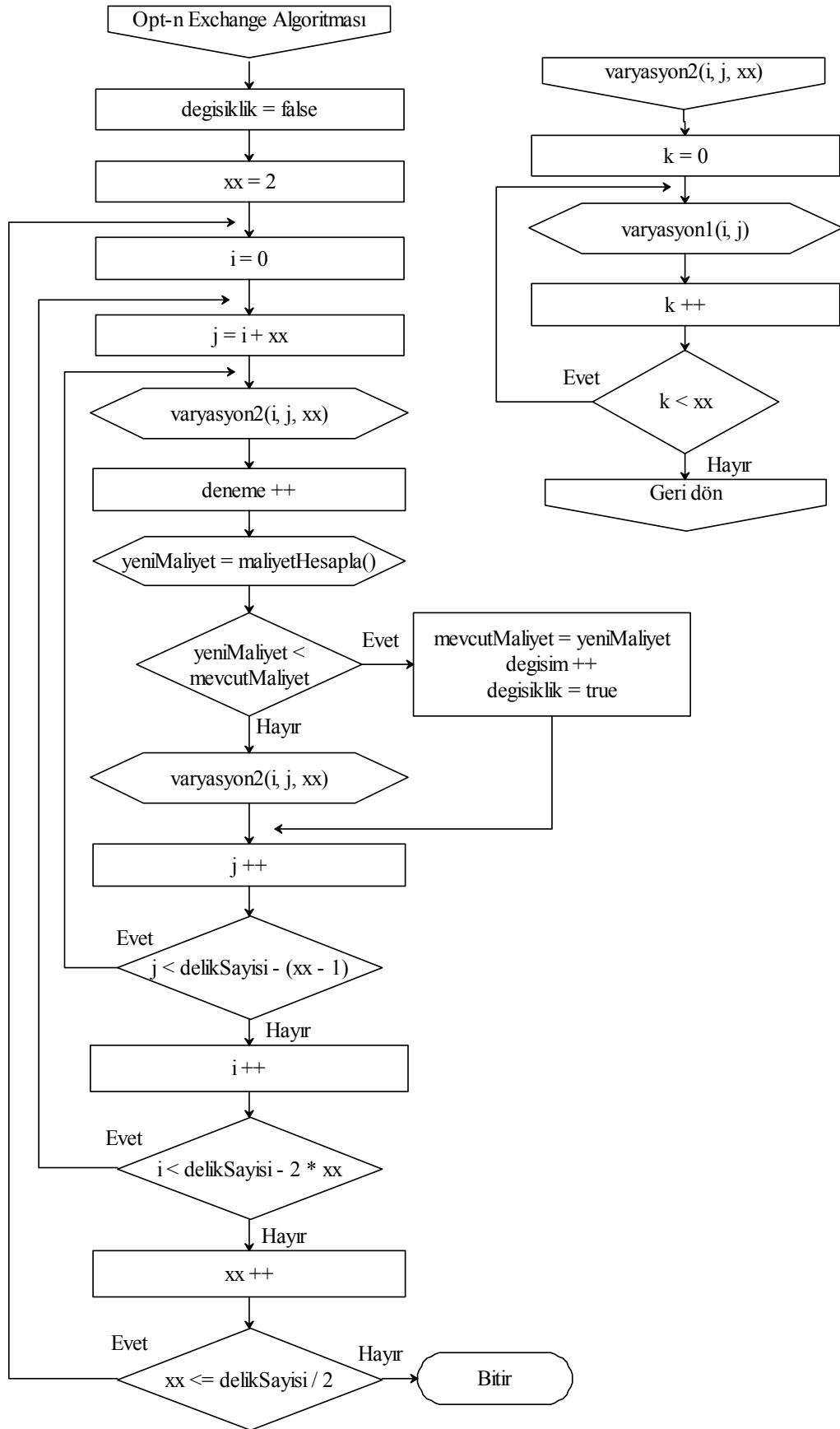
varyasyon1, varyasyon2() ve varyasyon3() metodları, sırasıyla yukarıda belirtilen 2-Opt Exchange, n-Opt Exchange ve Kaydırma algoritmalarıyla yapılacak bir sonraki aşama deęişikliklerin uygulanmasını sağlamaktadır. Bu işlemleri gerçekleştirmek üzere, her metod gerçekleştireceęi deęişiklik ile ilgili olarak bazı parametreleri ister. Bu metodlar, çağırıldıkları noktaya herhangi bir deęer döndürmezler.

Aşaęıda; Şekil 4.13 2-Opt Exchange Algoritması, Şekil 4.14 n-Opt Exchange Algoritması ve Şekil 4.15 Kaydırma Algoritmasına ait akış diyagramları gösterilmiştir.

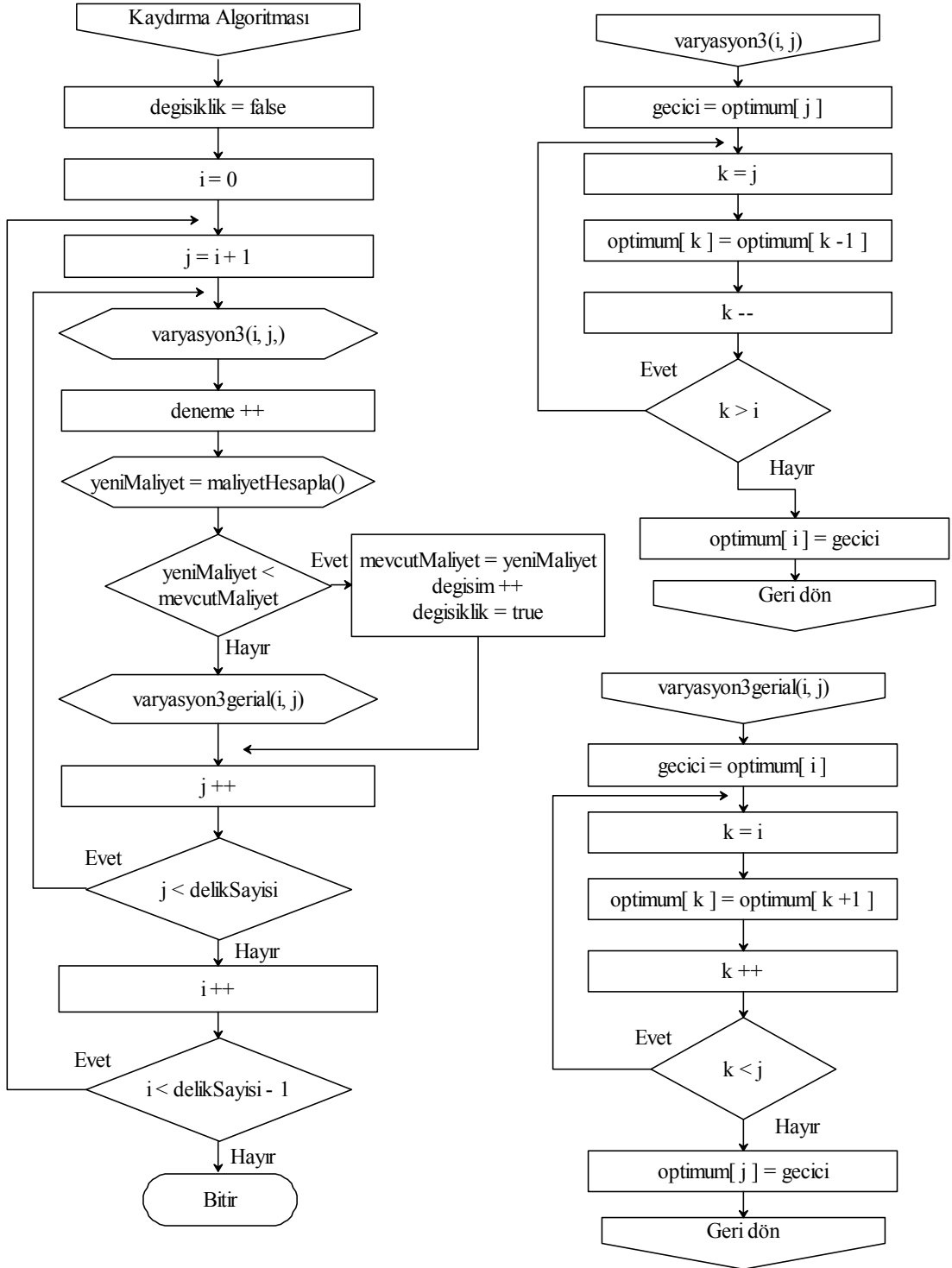




Şekil 4.13. 2-Opt Exchange algoritması akış şeması.



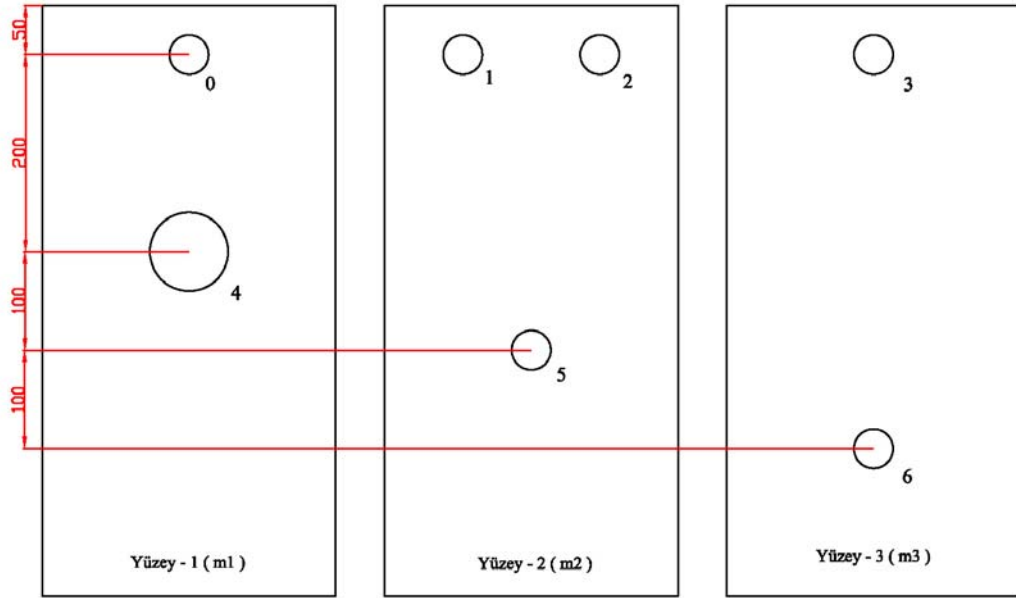
Şekil 4.14. n-Opt Exchange algoritması akış şeması.



Şekil 4.15. Kaydırma algoritması akış şeması.

maliyetHesapla() metodu, çözüm olarak öngörülen yeni dizilimin toplam maliyetini hesaplamak amacıyla kullanılır. Dizilimdeki her bir delik sırayla incelenerek, ilgili yüzeye ait matkap nesnesinin koordinat bilgileri güncellenir ve bulunan maliyetler birbirine eklenerek toplam maliyet değerine ulaşılır. Metod, çağırıldığı noktaya ilgili

dizilimin toplam maliyet değerini double veri türünde döndürür.  $x$  ekseninde artan sırayla rastgele düzenlenmiş delik numaralarının belirtildiği ve toplam maliyet hesaplamasının nasıl yapıldığını açıklamak amacıyla kullanılacak örnek bir çizim dosyası Şekil 4.16’da verilmiştir.



Şekil 4.16. Örnek hesaplamada kullanılacak 3 eksenli çizim.

Burada başlangıç dizilimi olarak deliklere ait numaralar kabul edildiğinde başlangıç delme dizilimini tutacak olan tek boyutlu *optimum[]* dizi değişkeni içeriği Çizelge 4.8’deki gibi olur.

Çizelge 4.8. Örnek çizime ait başlangıç dizilimini tutan *optimum[]* dizi içeriği.

İndis No	0	1	2	3	4	5	6
Delik No	0	1	2	3	4	5	6

Başlangıç diziliminde ifade edilen delik numarası, veritabanı tablosunda tutulan delik bilgilerine ait *ID* numarasını temsil etmektedir. Her bir deliğe ait koordinat, delik çapı ve yüzey bilgileri yine ID numaraları kullanılarak iki boyutlu farklı bir dizi içerisinde tutulmaktadır. Bu bilgilerin yer aldığı dizi içeriği Çizelge 4.9’da gösterilmiştir.

Çizelge 4.9. Örnek çizime ait delik bilgileri.

ID	xCoord	yCoord	Cap	matkapNo	matkapSiraNo
0	50	100	10	1	1
1	50	50	10	2	1
2	50	250	10	2	2
3	50	100	10	3	1
4	250	100	20	1	2
5	350	150	10	2	3
6	550	100	10	3	2

Delme işleminin optimum tamamlanma süresine ait dizilimi tutan *optimum[]* dizi değişkeni içerisinde tutulan sayılar, Çizelge 4.9’da belirtilen *ID* sütunu ile eşleştirilerek delme işlemine ait maliyet hesaplanır.

Mevcut delme işlemi sırasına göre ilk delinmesi gereken delik, çözüm diziliminin ilk elemanı olan ve Çizelge 4.9’da 0 *ID* numarasına sahip deliktir. Burada öncelikle, *matkapNO* sütunundan, ilgili delik bilgisinin hangi yüzey ve matkap tarafından delineceği belirlenir. Bu örnekte, 1 numaralı matkap delme işlemini gerçekleştirir. Dolayısıyla, *m1* matkap nesnesine ait *setCoord()* metodu, ilgili deliğin bu matkapta kaçınıcı delik olduğu ve daha önce yerine getirilen delme işlemlerinin toplam maliyeti gönderilerek çalıştırılır. Bu metod, daha önce yapılan delme işlemlerinde kullanılan *xCoord*, *yCoord*, *cap* gibi bilgileri de hesaba katarak bir maliyet belirler ve oluşan maliyeti çağırıldığı noktaya geri döndürür. Geri döndürülen bu değer, daha önce oluşan ve toplam maliyet değerini tutan değişkenin değerine eklenir.

Daha sonra, *optimum[]* dizi değişkeninde 1 numaralı indiste yer alan ikinci delik için maliyet hesabına geçilir. *ID* değeri 1 olan bu deliğin Çizelge 4.9’dan faydalanarak, 2. matkap olan *m2* matkap nesnesine ait birinci delik olduğu anlaşılır. Dolayısıyla *m2* matkap nesnesine ait *setCoord()* metodu, ilgili deliğin bu matkapta kaçınıcı delik olduğu ve daha önce yerine getirilen delme işlemlerinin toplam maliyet bilgileri gönderilerek çalıştırılır. Burada da bir maliyet değeri belirlenerek oluşan maliyet değeri metodun çağırıldığı noktaya döndürülür.

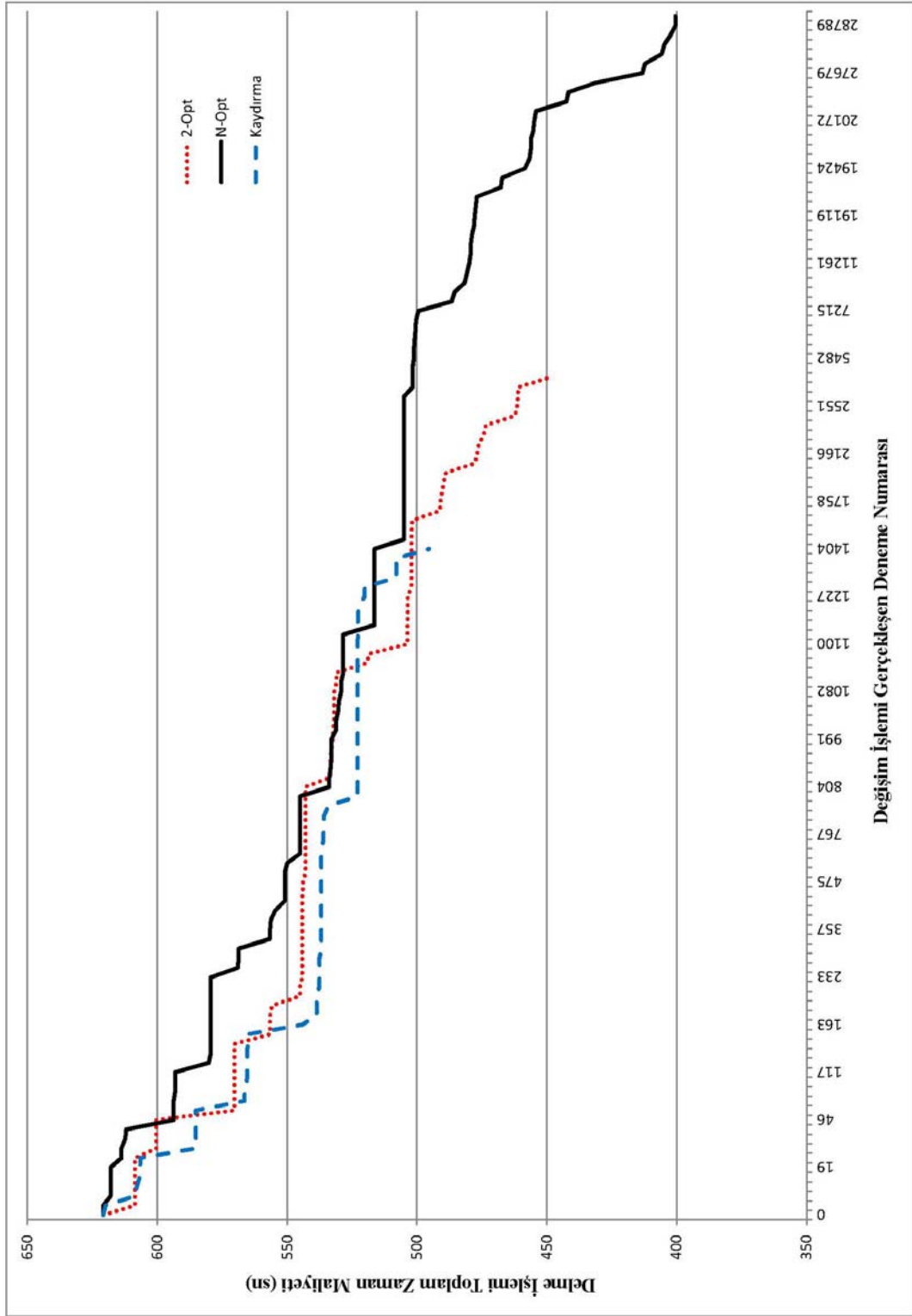
Delme işlemi, yukarıda açıklanan işlem basamaklarına benzer şekilde, *optimum*[ ] dizi değişkeni içerisinde tutulan delme sırasına uygun olarak tamamlanana kadar devam eder. Böylece gerçekleşecek delme işleminin toplam maliyeti elde edilmiş olur.

Geliştirilen algoritmaları, elde edilen bu veriler ışığında grafikler yardımıyla detaylı olarak örneklerle açıklayalım. Şekil EK B.9'da verilen üç yüzeyde toplam 50 delik bulunduran çizim dosyasının bahsedilen her bir algoritma ile tek tek incelenmesi sonucu oluşan maliyet değişimleri Şekil 4.17'de gösterilmiştir. Çizim dosyalarına bağlı olarak her algoritma farklı noktalarda tıkanmakta, dolayısıyla farklı algoritmalar farklı çizim dosyaları için ideal sonuca daha yakın sonuç vermektedir. Elde edilen bu sonuçlar, algoritmaların kombinasyonu kullanıldığında daha ideal noktalara ulaştırılabilmektedir. Bu durum Şekil 4.18 ve Şekil 4.19'da gösterilmiştir.

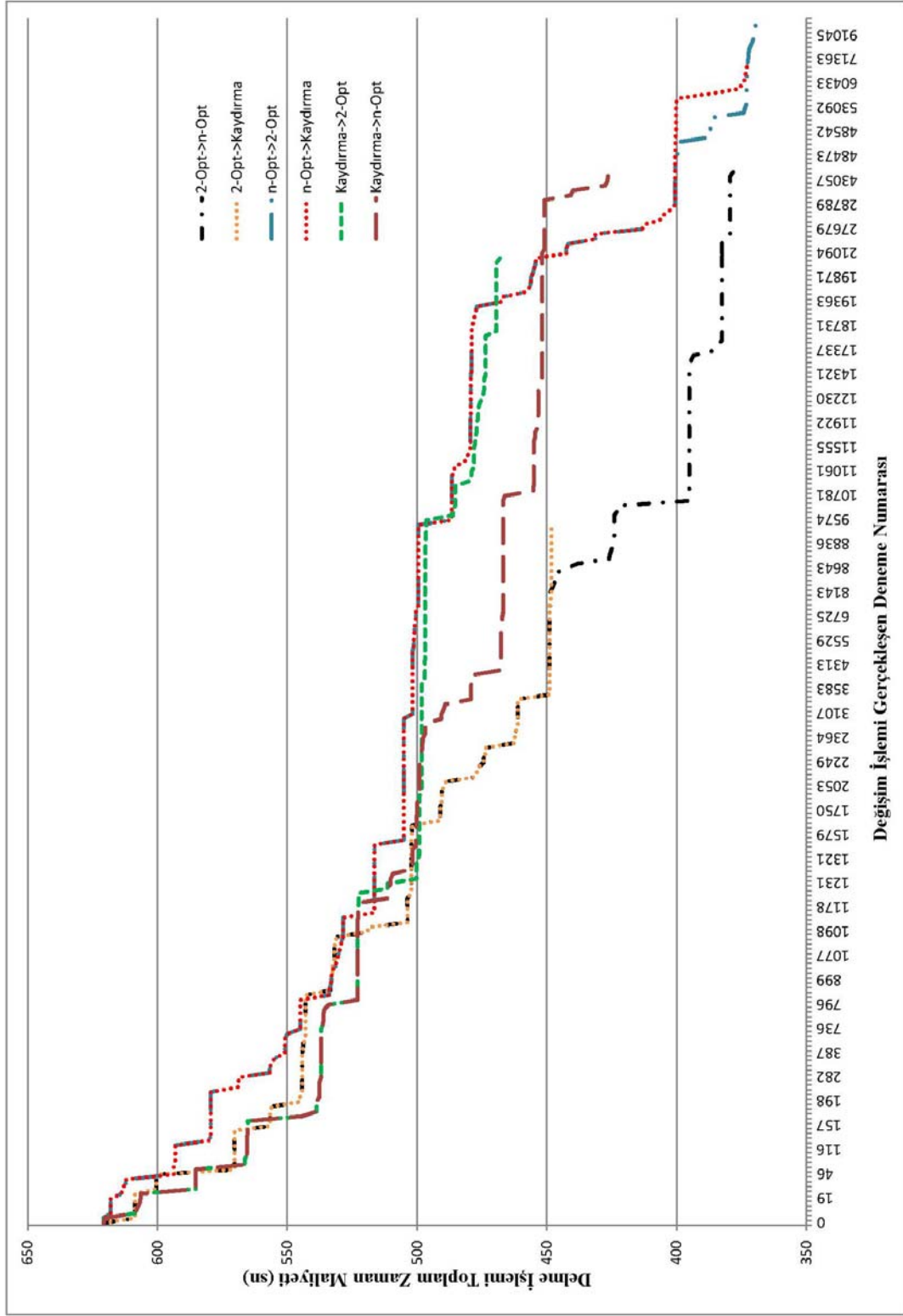
Şekil EK B.9'da verilen 50 delikli çizim dosyası bahsedilen üç algoritmanın değişik ikili kombinasyonları kullanılarak incelenmiş ve yine farklı tıkanma noktalarında optimizasyon işlemi sonlanmıştır. Kullanılan altı değişik optimizasyon algoritması kombinasyonu ve bu kombinasyonlar kullanılarak yapılan inceleme işleminde oluşan maliyet değişimleri Şekil 4.18'de gösterilmiştir.

Şekil 4.19 ise yine 50 delikli çizim dosyasının, bu algoritmaların üçlü kombinasyonları kullanılarak yapılan inceleme işlemi sonucunda oluşan maliyet değişimlerini göstermektedir. Elde edilen bu veriler ışığında, oluşturulacak kombinasyonlarda kullanılacak ilk algoritma, optimizasyon işleminin akışını doğrudan etkilemektedir. Ancak, her zaman aynı kombinasyonu kullanmak en ideal dizilime ulaşmak için yeterli değildir. Çizim dosyası içeriğine bağlı olarak kullanılması gereken kombinasyon farklılık göstermektedir. Bu sebepten dolayı, geliştirilen yazılım içerisinde değişik kombinasyonlar yardımıyla çizim dosyası incelenmekte ve bu kombinasyonlardan elde edilen zaman maliyetleri değerlendirilerek en düşük maliyeti sağlayan kombinasyonun ürettiği delme işlemi dizilimi kullanıcıya aktarılmaktadır. EK Açıklamalar B, üç yüzeyde sırasıyla toplam 70 – 90 – 105 - 120 delik bulunan çizim dosyaları ve EK Açıklamalar C'de ise bu

çizim dosyaları için en iyi sonucu veren üç değişik kombinasyona ait maliyet akışlarını gösteren maliyet akış grafikleri verilmiştir.

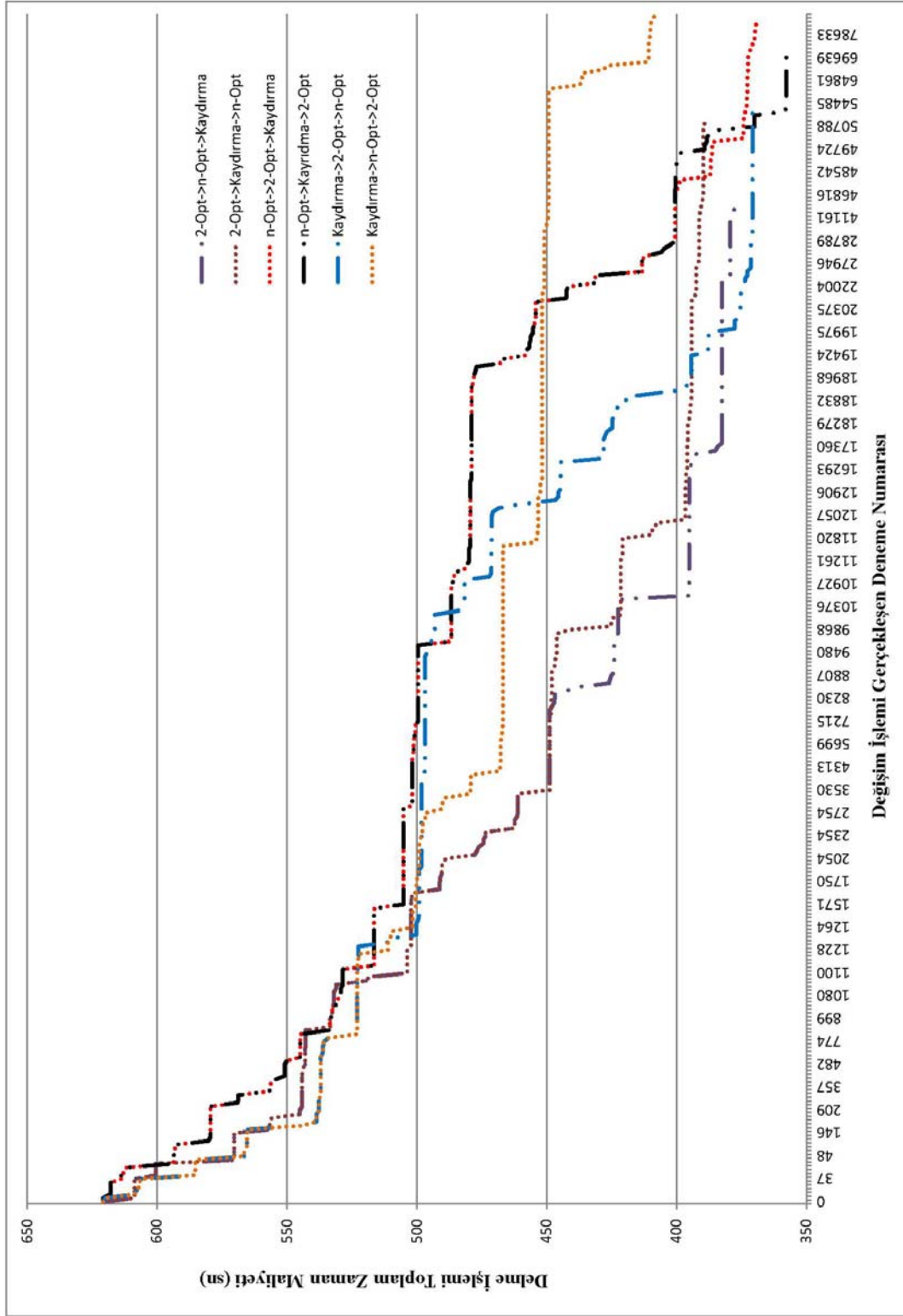


Şekil 4.17. Şekil Ek B.9’da verilen 50 Delikli çizim dosyasının üç değişik algoritma ile incelenmesiyle oluşan maliyet akışı.



Şekil 4.18. Şekil Ek B.9'da verilen 50 delikli çizim dosyasının ikili algoritma kombinasyonları ile incelenmesiyle oluşan akış maliyeti.

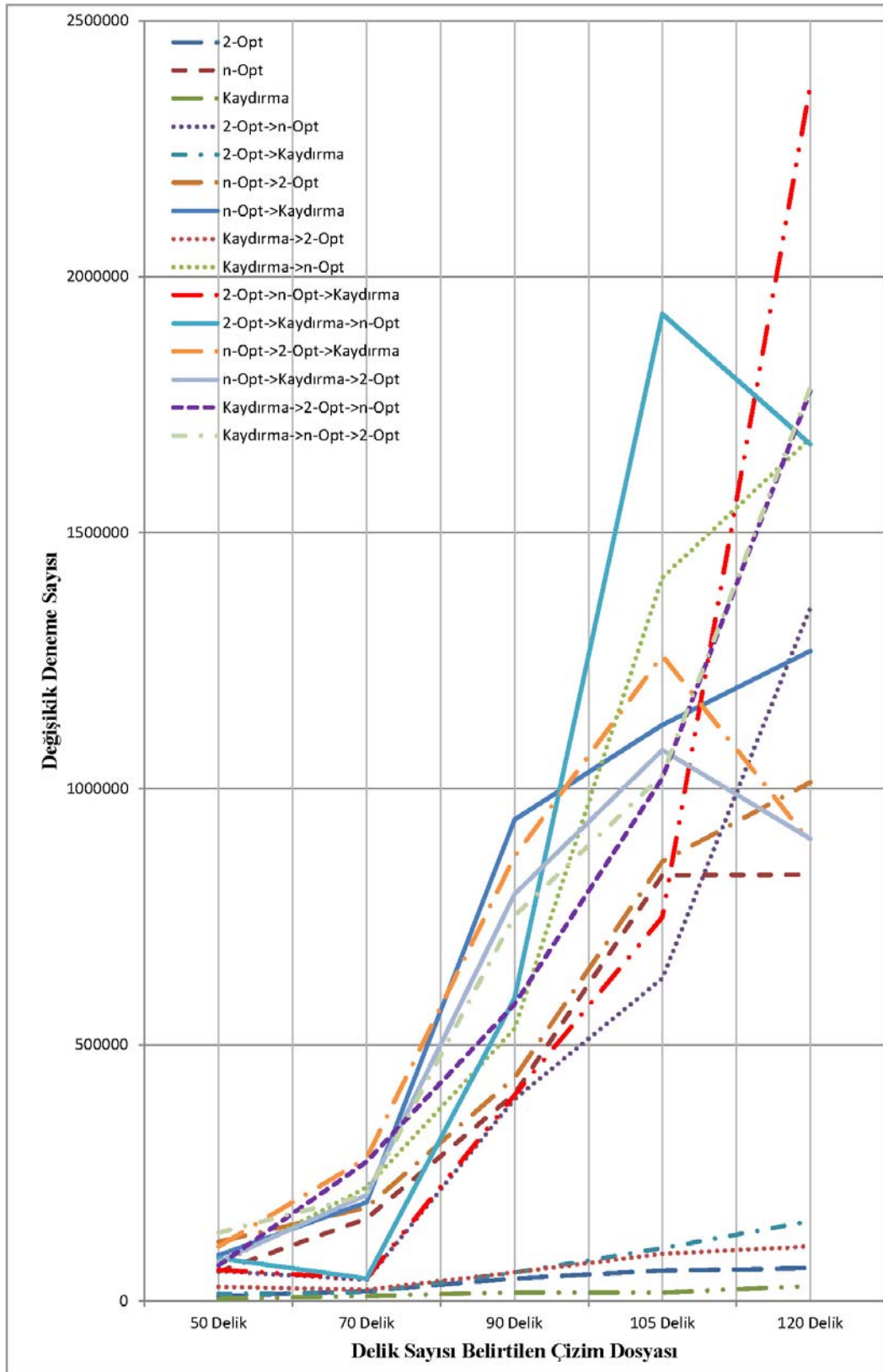




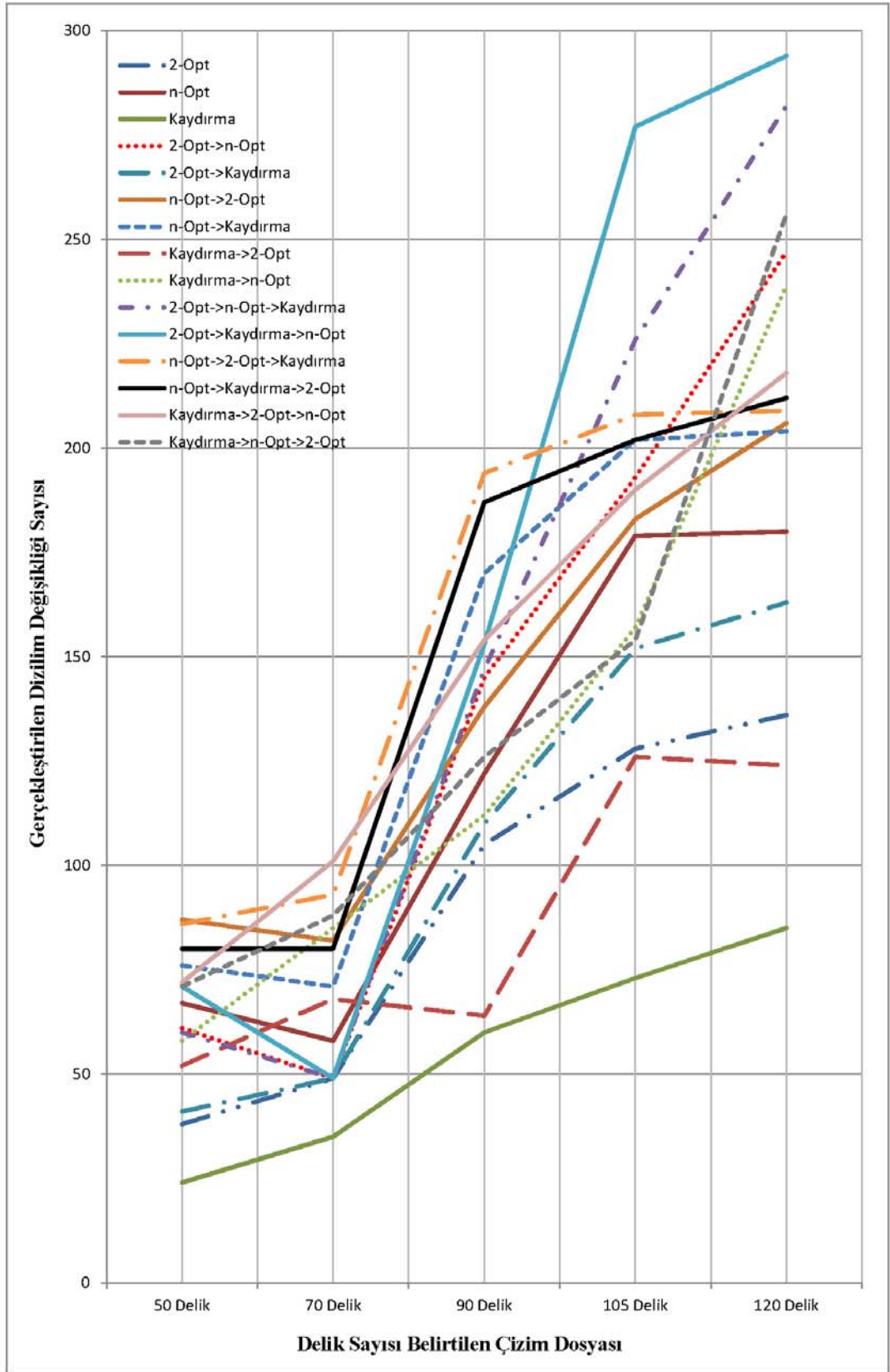
Şekil 4.19. Şekil Ek B.9'da verilen 50 delikli çizim dosyasının üçlü algoritma kombinasyonları ile incelenmesiyle oluşan akış maliyeti.

Bu çalışmada gerçekleştirilen optimizasyon işlemlerinde sonlandırma durumu olarak, daha düşük maliyeti sağlayacak yeni bir dizilim değişikliği gerçekleştirilemeyecek olan tıkanma noktasıdır. Bahsedilen algoritmaların her biri, yalnız başına kullanıldığında farklı noktalarda tıkanmakta ve farklı delme dizilimi/maliyeti sonuçları üretmektedir. Ayrıca, değişik çizim dosyaları üzerinde yapılan hesaplamalarda, her zaman aynı algoritma ya da aynı algoritmalar kombinasyonu minimum sonucu vermemektedir. Erken tıkanma noktalarını aşmak ve daha düşük delme maliyetleri elde edebilmek amacıyla, bu algoritmalar farklı kombinasyonları şeklinde kullanılmıştır. Gerçekleştirilen yazılım tarafından bu kombinasyonlardan elde edilen sonuçlara göre, maliyete dayalı karar işlemi de otomatik olarak gerçekleştirilir. Değişik kombinasyonlar tarafından elde edilen maliyet değerleri karşılaştırılarak, elde edilen en düşük maliyete ait delme işlemi sıralaması, optimum dizilim olarak kabul edilir ve kullanıcıya bu dizilim ve maliyet iletilir. Daha önce Çizelge 4.7’de, kullanılacak algoritmaların öncelik sıralamasının, elde edilecek sonuçları büyük oranda etkilediği, algoritmaların değişik kullanım sıralamalarına ait örnek sonuçlarla ifade edilmişti.

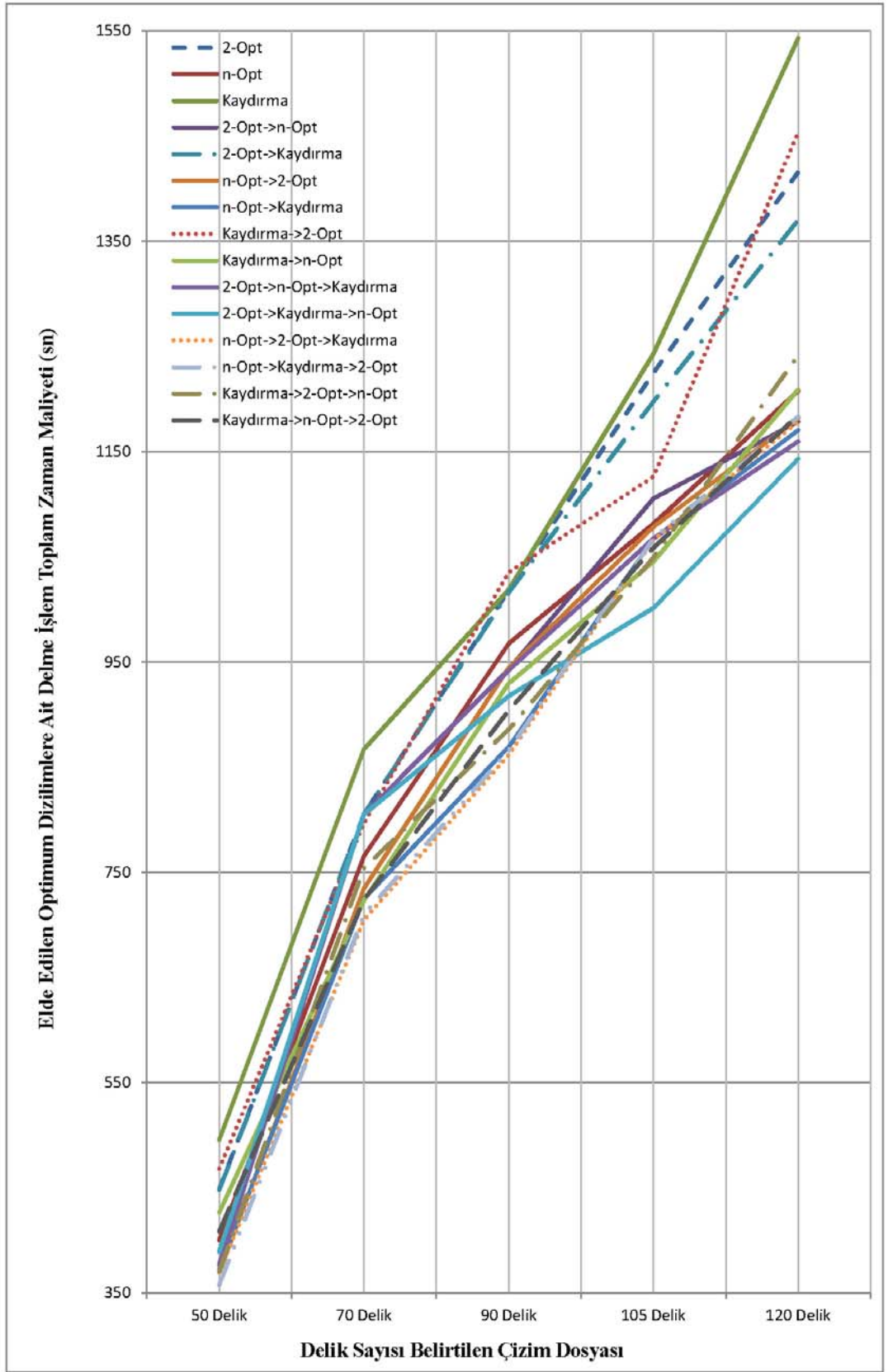
Değişik algoritma kombinasyonları kullanılarak 50 delik içeren çizim dosyası optimizasyon işlemine tabi tutularak sonuçlar elde edilmiş ve bu sonuçlara ait maliyet değişimleri yakınsama grafikleri Şekil 4.17, Şekil 4.18 ve Şekil 4.19’da verilmiştir. Bu grafiklere ek olarak 70, 90, 105 ve 120 delik içeren çizim dosyaları ve bu çizim dosyalarına ait en ideal sonucu üreten üç adet algoritmaya ait maliyet değişim grafikleri Ek Açıklamalar C’de verilmiştir. Yapılan bu değerlendirmelerde kullanılan tüm algoritma kombinasyonlarına ait dizilim değişikliği deneme sayıları Şekil 4.20’de, bu deneme işlemleri sonucunda daha düşük toplam delme süresi maliyeti sağlayan değişiklik sayıları Şekil 4.21’de ve bu algoritmalar ile elde edilen toplam delme süresi maliyetleri Şekil 4.22’te gösterilmiştir. Yazılım içerisinde, x ekseninde hareket parametresi 0,02 sn/mm, y ekseninde hareket parametresi 0,01 sn/mm, delme işlemi süresi 12 sn ve matkap ucu değişikliği süresi 30 sn olarak kabul edilmiş ve bu değerler yardımıyla verilen maliyet değerleri hesaplanmıştır.



Şekil 4.20. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından gerçekleştirilen değişiklik deneme sayıları grafiği.



Şekil 4.21. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından gerçekleştirilen değişiklik sayıları grafiği.



Şekil 4.22. Değişik çizim dosyalarına göre algoritma kombinasyonları tarafından elde edilen toplam delme zamanı maliyeti grafiği.

Hızlı ve doğru sonuca ulaşmak amacıyla Şekil 4.20 ve Şekil 4.22’te elde edilen veriler bir arada değerlendirilmelidir. Delme işleminin tamamlanacağı toplam delme süresi maliyetinin düşük olduğu sonuca, daha az deneme sayısı ile ulaşan kombinasyonun belirlenmesi yazılımı gereksiz iş yükünden kurtararak daha hızlı ve doğru sonuca ulaşmayı sağlayacaktır. Bu amaçla her bir çizim dosyasından elde edilen veriler öncelikle toplam delme süresi maliyetine göre küçükten büyüğe doğru sıralanmış ve en düşük maliyetten başlayarak sırasıyla 100, 90, 80, 70 ve 60 ağırlık puanları ilgili kombinasyonlara atanmıştır. Diğer kombinasyonlara, hesaplama işlemini sorunsuz tamamlayabilmek adına 1 ağırlık puanı atanmıştır. Daha sonra, seçilen ilk beş kombinasyon, kendi içerisinde yapmış olduğu değişiklik deneme sayılarına göre yeniden küçükten büyüğe doğru sıralanmış ve aynı mantıkla ağırlık puanlamaları atanmıştır. Bu puanlama sonucu Çizelge 4.10’de gösterilmiştir.

Çizelge 4.10. Kullanılan kombinasyonların sonuçlarına göre ağırlık puanları.

Kombinasyon	Puanlamalar									
	50 Delik		70 Delik		90 Delik		105 Delik		120 Delik	
	Deneme	Maliyet	Deneme	Maliyet	Deneme	Maliyet	Deneme	Maliyet	Deneme	Maliyet
2-Opt	1	1	1	1	1	1	1	1	1	1
n-Opt	1	1	1	1	1	1	1	1	1	1
Kaydırma	1	1	1	1	1	1	1	1	1	1
2-Opt->n-Opt	1	1	1	1	1	1	1	1	80	70
2-Opt->Kaydırma	1	1	1	1	1	1	1	1	1	1
n-Opt->2-Opt	60	80	1	1	1	1	1	1	1	1
n-Opt->Kaydırma	80	60	100	60	60	80	1	1	90	80
Kaydırma->2-Opt	1	1	1	1	1	1	1	1	1	1
Kaydırma->n-Opt	1	1	70	80	1	1	70	90	1	1
2-Opt->n-Opt->Kaydırma	1	1	1	1	1	1	1	1	60	90
2-Opt->Kaydırma->n-Opt	1	1	1	1	1	1	60	100	70	100
n-Opt->2-Opt->Kaydırma	70	90	60	100	70	100	80	60	100	60
n-Opt->Kaydırma->2-Opt	90	100	90	90	80	90	1	1	1	1
Kaydırma->2-Opt->n-Opt	100	70	1	1	100	70	100	80	1	1
Kaydırma->n-Opt->2-Opt	1	1	80	70	90	60	90	70	1	1

Elde edilen ağırlık puanları, kullanılan kombinasyonların değişik çizim dosyalarından elde ettiği verileri tek bir değerle temsil etmek amacıyla, Aritmetik, Geometrik ve Harmonik ortalama işlemine tabi tutulmuş ve elde edilen sonuçlar Çizelge 4.11’te gösterilmiştir. Burada *n-Opt->2-Opt->Kaydırma* algoritma kombinasyonunun genel olarak en ideal sonucu ürettiği söylenebilir. Daha sonra *n-Opt->Kaydırma* algoritma kombinasyonunun, ikinci sırada genel olarak ideal sonucu ürettiği tabloda bulunan veriler yorumlanarak ifade edilebilir. Bu değerler; az sayıda dizilim değişikliği denemesi gerçekleştirerek minimum maliyete yaklaşan algoritma kombinasyonları hakkında fikir edinmemize yardımcı olmuş ve gerçekleştirilecek optimizasyon işleminde *n-Opt->Kaydırma* ve *n-Opt->2-Opt->Kaydırma* yöntemleri kullanılarak çizim dosyalarının değerlendirilmesinin yeterli olacağı sonucuna ulaşmamızı sağlamıştır.

Çizelge 4.11. Değişik ortalama yöntemleri ile elde edilen sonuçlar.

<b>Kombinasyon</b>	<b>Aritmetik</b>	<b>Geometrik</b>	<b>Harmonik</b>
2-Opt	1	1	1
n-Opt	1	1	1
Kaydırma	1	1	1
2-Opt->n-Opt	15,80	2,37	1,25
2-Opt->Kaydırma	1	1	1
n-Opt->2-Opt	14,80	2,33	1,25
n-Opt->Kaydırma	61,20	31,61	4,74
Kaydırma->2-Opt	1	1	1
Kaydırma->n-Opt	31,60	5,69	1,65
2-Opt->n-Opt->Kaydırma	15,80	2,36	1,25
2-Opt->Kaydırma->n-Opt	33,60	5,79	1,65
n-Opt->2-Opt->Kaydırma	79,00	77,30	75,65
n-Opt->Kaydırma->2-Opt	54,40	14,86	2,46
Kaydırma->2-Opt->n-Opt	52,40	14,43	2,46
Kaydırma->n-Opt->2-Opt	46,40	13,43	2,45

## BÖLÜM 5

### GELİŞTİRİLEN YAZILIMIN TANITIMI

Yazılım ihtiyaçlarının karşılanmasında çok değişik programlama dili seçeneği mevcuttur. Her birinin kendisine göre avantajları ve tercih etmek için değişik üstünlükleri bulunmaktadır. Geliştirilen yazılım için Java programlama dili tercih edilmiştir. Java programlama dili, platform bağımsız bir dil olması nedeniyle, her türlü işletim sisteminde sorunsuz olarak çalışabilmekte, ayrıca performans açısından da istenen seviyeyi sunmaktadır.

Verilerin düzenli bir şekilde saklanabilmesi amacıyla veritabanı programına ihtiyaç duyulmuş ve Java programlama dili ile sorunsuz çalışan açık kaynak kodlu MySQL veritabanı programı tercih edilmiştir. Pratik kurulumu ve kolay kullanımı nedeniyle bütünlük bir çözüm olan WampServer programı bünyesinde yer alan MySQL veritabanı yazılımı kullanılmıştır.

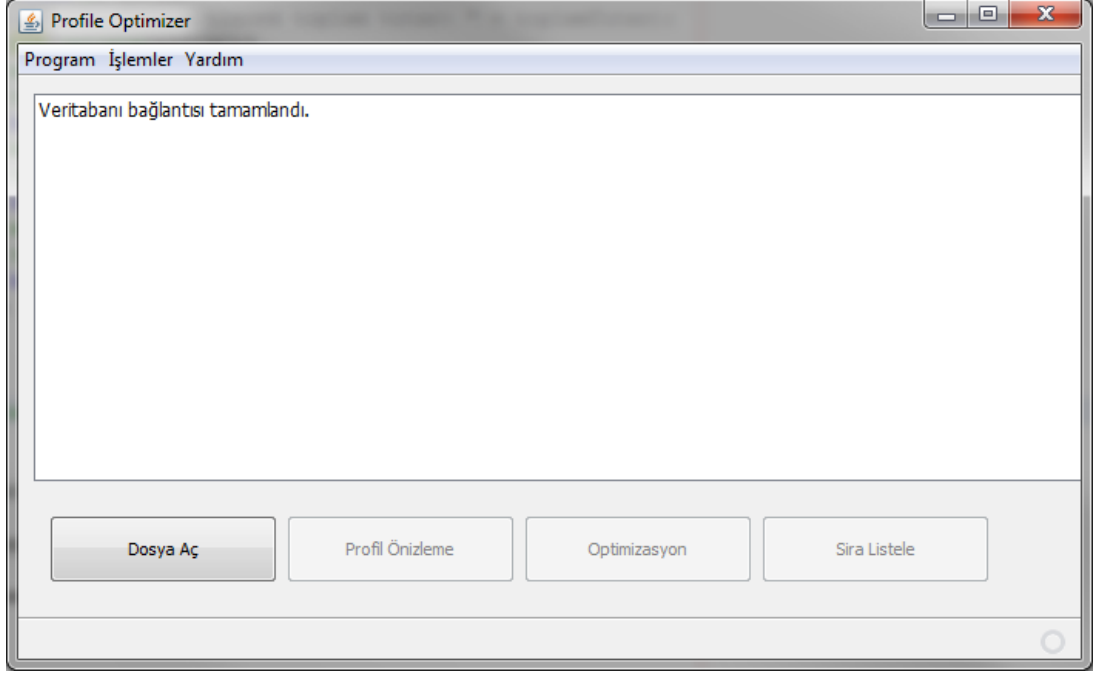
#### 5.1. PROGRAM HAKKINDA GENEL BİLGİLER

Yazılımda, basit ve öz bir yerleşim oluşturulmuş ve kullanımı kolay bir yapı halinde düzenlenmiştir. Şekil 5.1’de gösterildiği gibi menü çubuğu, işlem log kayıt bölümü, kontrol düğmeleri ve durum çubuğundan oluşan bir arayüz tasarlanmıştır. Ekranda görüntülenen düğmelerin kullanım amacı şöyle açıklanabilir;

Dosya Aç: İncelenmek istenen dosyayı bulmak üzere bir göz at penceresi açar. Dosyanın açılmasını takiben inceleme işlemi otomatik olarak başlar ve gerekli çizgi ve daire bilgileri ilgili veritabanı tablolarına aktarılır. Bu aşamada bulunan çizgi ve daire bilgileri anlamsızdır. Yine bu işlemi takiben anlamlandırma aşaması otomatik olarak devam eder ve bulunan çizgi ve daire bilgileri; profil dış kenar çizgileri ve bu



izgiler arasında kalan daireleri delik olarak adlandırılarak ilgili tablolara aktarılır. Tm iřlemlerin tamamlanmasının ardından bu anlamlı bilgiler profil yzeyleri olarak Profil nizleme penceresinde grntlenir.



řekil 5.1. Geliřtirilen yazılıma ait ekran grnts.

Profil nizleme: Dosya A dęmesinin son adımı olarak aılan Profil nizleme penceresini yeniden amak amacıyla kullanılır. Bu pencerede, her bir yzey farklı bir zim olarak ekrana yansıtılır. Yansıtılan bu zimlerde, yzeye ait delik bilgileri, deliklerin veritabanı tablosundaki sıra bilgileri ve her bir profil yzeyinin lleri gsterilmektedir. Bu pencere yardımıyla yanlış tanımlanan yzey bilgileri dzeltilebilirken, profil tr de tanımlanabilmektedir. řekil 5.2’de bu pencereye ait bir grnt gsterilmektedir.

Optimizasyon: Profil yzey bilgilerinin kesinleřmesinin ardından optimizasyon iřleminin bařlatılmasını saęlar. Yapılan optimizasyon iřlemi sonucunda ulařılan dizilimin toplam delinme sresi ve bu srenin elde edildięi toplam deęiřiklik adedi bilgileri log ekranında grntlenir.



Şekil 5.2. Profil önizleme penceresi.

Sıra Listele: Şekil 5.3'te gösterildiği gibi, elde edilen sonuç diziliminin ve profile ait parametrik değerlerin görüntülendiği bir pencere açar. Bu pencerede, delme sırası kolayca takip edilebilir. Ayrıca bulunan sonuç dizilimi ve gerekli diğer bilgiler, daha önceden seçilen ölçekleme yöntemi kullanılarak “optimized” isimli veritabanı tablosuna aktarılabilir.

**Kullanılan Ölçekleme Yöntemi:** Mutlak (Absolute)

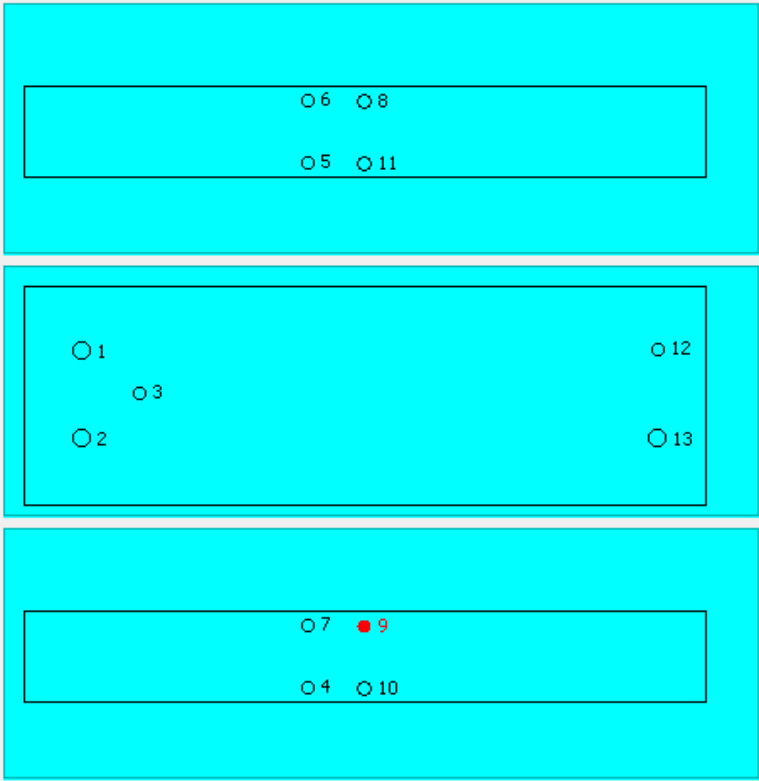
Sıra No	Veritabanı ID	X Koordinatı	Y Koordinatı	Delik Çapı	Matkap No
1	1	80.0	213.0	25.0	2
2	2	80.0	93.0	25.0	2
3	3	160.0	153.0	18.0	2
4	9	392.0	20.0	18.0	3
5	9	392.0	20.0	18.0	1
6	6	392.0	105.0	18.0	1
7	6	392.0	105.0	18.0	3
8	7	467.0	105.0	20.0	1
9	7	467.0	105.0	20.0	3
10	8	467.0	20.0	20.0	3
11	8	467.0	20.0	20.0	1
12	4	870.0	213.0	18.0	2
13	5	870.0	93.0	25.0	2

Veritabanına Ekle

Kapat

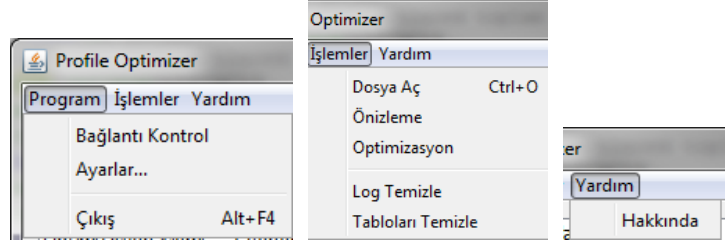
**Bulunan Profil Türü:**  
???

**Profil Boyu:**  
935.0 mm



Şekil 5.3. Profil listele penceresi.

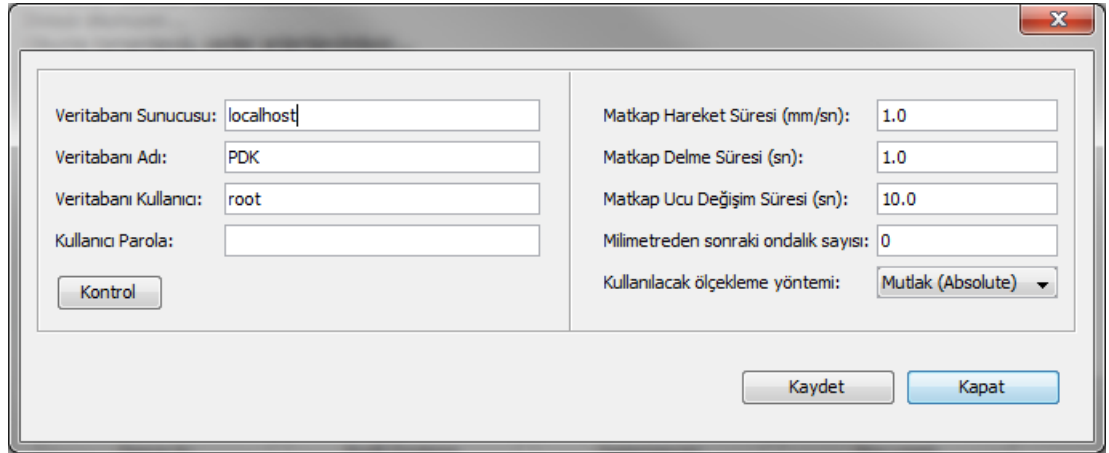
Program penceresinde bulunan menü çubuğu ve içerikleri Şekil 5.4'te gösterilmiştir. Buradan, yukarıda tanımlanan işlemlerin dışında aşağıda belirtilen işlemler de yapılabilmektedir.



Şekil 5.4. Program menü çubuğu ve içerikleri.

Program-Bağlantı Kontrol: Bu seçenek yardımıyla veritabanı ile program arasındaki bağlantı, mevcut parametrik bilgiler kullanılarak test edilir. Eğer bağlantıda bir sorun olursa, bir uyarı yardımıyla kullanıcıya bildirilir.

Program-Ayarlar: Şekil 5.5'te gösterildiği gibi bir pencere görüntüler. Bu pencere yardımıyla, programın işleyişi ile ilgili parametre değerleri belirlenebilir. Ayrıca kontrol isimli düğme yardımıyla, tanımlanan veritabanı bilgileri ile bağlantı kontrolü gerçekleştirilebilir.



Şekil 5.5. Ayarlar penceresi.

Program-Çıkış: Program penceresinin kapatılarak, programın sonlandırılmasını sağlar.

İşlemler-Log Temizle: Program ana penceresinde bulunan bilgilendirme bölümündeki açıklamaların temizlenmesini sağlar.

İşlemler-Tabloları Temizle: Kullanılacak veri tabanı tablolarında daha önceden eklenmiş verilerin silinmesini sağlar. Bu işlem, her yeni dosya incelemesinde otomatik olarak çalışmaktadır.

Yardım-Hakkında: Program hakkında bilgilerin yer aldığı bölümdür.

## **5.2. PROGRAMIN ÇALIŞTIRILMASI**

Program çalıştırılmadan önce, MySQL veritabanı programının çalışır ve kullanıma hazır durumda olması gerekmektedir. Ayrıca çalışmaya başlamadan önce, Dosya menüsünden Ayarlar seçeneği yardımıyla Şekil 5.5'te belirtilen veritabanı bağlantı bilgileri ve hesaplamalarda kullanılacak olan Matkap Hareket Süresi (sn/mm), Matkap Delme Süresi(sn), Matkap Ucu Değişim Süresi (sn) parametrik değerleri sisteme girilmelidir. Ardından program kullanıma hazırdır.

İşlemleri gerçekleştirmek üzere ana pencerenin alt bölümünde yer alan düğmeler sırasıyla takip edilmelidir. Öncelikle “Dosya Aç” düğmesi yardımıyla dosya bulunur ve inceleme-anlamlandırma işlemleri başlatılır. “Profil Önizleme” penceresi otomatik olarak görüntülenir ve program tarafından algılanan yüzey ve profil bilgileri kullanıcı tarafından kontrol edilir. Görüntülenen bilgilerde yapılması gereken düzeltmeler varsa bu düzeltmeler yapıldıktan sonra güncelle düğmesi yardımıyla yapılan değişiklikler geçerli hale getirilir ve açık olan “Profil Önizleme” penceresi kapatılır. Bu işlemin ardından, Optimizasyon düğmesi ile optimizasyon işlemi gerçekleştirilir.

Tüm bu işlemlerin tamamlanmasıyla “Sıra Listele” düğmesine basılarak yazılım tarafından belirlenen delme sırası incelenebilir. Açılan pencerede tablo üzerinde bulunan satırlara fare sol tuşu yardımıyla tıklanıldığında, alt bölümde yer alan önizleme çizimlerinde ilgili delikler gözlemlenebilir. Bulunan dizilimin ve bu dizilimde bulunan gerekli bilgilerin kalıcı hale getirilerek veritabanı tablosuna aktarılması amacıyla “Veritabanına Ekle” düğmesi kullanılır ve işlemler tamamlanmış olur.

## BÖLÜM 6

### SONUÇLAR VE ÖNERİLER

Yapılan bu çalışma ile standart bir çizim programından alınan imalat resimlerinin analizi yapılarak, işlem açısından kullanılacak delikler ve çaplarının doğru bir şekilde algılanması sağlanmıştır. Bu verilerin veritabanında saklanması ile gerek SCADA sisteminde gösterilmesi, gerekse otomasyon sisteminin kontrol amaçlı kullanması için bir imkan sağlanmıştır. Bu yaklaşımla manuel veri girişlerinden kaynaklanan hataların önlenmesi sağlanmıştır.

Sistemin sağladığı diğer bir avantaj ise profil delme aşamasının gerçekleştirilmesine yönelik hızlı bir optimizasyon işleminin yapıyor olmasıdır. Sistem birbirinden  $y$  ve  $z$  eksenlerinde bağımsız,  $x$  ekseninde ise bağımlı hareket eden bir yapıda olduğundan 3 eksenin beraberce optimize edilmesi zorunluluğu bulunmaktaydı.

Geliştirilen yazılım veri bazlı bir işlem yapılmanın yanında oluşturulan veritabanında işleme ait tüm verilerin saklanıp değerlendirilmesine imkan verecek şekilde bir yapının oluşturulmasına fırsat tanımaktadır. Bu ise maliyet, verimlilik, üretim sonuçlarının değerlendirilmesi gibi önemli yaklaşımlar için önem arz etmektedir.

Çalışma kapsamında geliştirilen yazılımın doğruluğunu test etmek amacıyla, sekiz farklı çizim dosyası üzerinde, her bir çizim dosyası için bir, iki ve üç yüzey kullanılarak hesaplamalar yapılmıştır. Yapılan bu hesaplamalara ait veriler Çizelge 6.1’de verilmiştir.

Çizelge 6.1’de bulunan veriler iki grup halinde oluşturulmuştur. Bunlardan birincisi, gerçekleştirilen yazılım tarafından hızlı bir şekilde elde edilen ve yaklaşık sonucu gösteren veriler, ikincisi ise mevcut tüm varyasyonların denenmesi sonucunda, uzun zaman harcanarak yapılan hesaplamalara ait kesin sonucu gösterdiği varsayılan

Çizelge 6.1. Değişik durumlara ait deney işlemleri sonuçları.

Deney No	Dosya Adı	Yüzey 1 Delik Sayısı	Yüzey 2 Delik Sayısı	Yüzey 3 Delik Sayısı	Toplam Delik Sayısı	Deneme Sayısı	Değişiklik Sayısı	Bulunan Maliyet (sn)	En İyi Maliyet (sn)	Toplam Çözüm Sayısı	Mevcut Çözüm No	Maliyet Farkı (sn)	Maliyet Farkı/İdeal Maliyet Oranı	Güvenilirlik
1	UNP200_8delik.dxf	0	6	0	6	30	2	52,98	52,98	720	1	0	0,00%	100,00%
		2	6	0	8	68	2	64,00	52,00	40.320	343	12	23,08%	76,92%
		2	6	2	10	127	4	52,00	52,00	3.628.800	1	0	0,00%	100,00%
2	cizim1_ayniboy_9delik.dxf	0	5	0	5	18	0	130,00	130,00	120	1	0	0,00%	100,00%
		4	5	0	9	95	1	134,00	134,00	362.880	1	0	0,00%	100,00%
		4	5	4	13	264	4	134,00	134,00	6.227.020.800	1	0	0,00%	100,00%
3	cizim2_3farkliboy_9delik.dxf	0	5	0	5	18	1	290,00	290,00	120	1	0	0,00%	100,00%
		4	5	0	9	218	3	272,00	272,00	362.880	1	0	0,00%	100,00%
		4	5	4	13	672	6	272,00	272,00	6.227.020.800	1	0	0,00%	100,00%
4	cizim3_3farkliboy_9delik.dxf	0	5	0	5	18	1	290,00	290,00	120	1	0	0,00%	100,00%
		4	5	0	9	159	4	294,00	284,00	362.880	10.611	10	3,52%	96,48%
		4	5	4	13	330	11	306,00	284,00	6.227.020.800		22	7,75%	92,25%
5	200x100x10DIKDORT GEN_10delik.dxf	0	6	0	6	30	2	53,00	53,00	720	1	0	0,00%	100,00%
		4	6	0	10	127	4	63,50	51,50	3.628.800		12	23,30%	76,70%
		4	6	4	14	571	8	51,50	51,50	87.178.291.200	1	0	0,00%	100,00%
6	200x200x10KUTU_12delik.dxf	0	6	0	6	30	2	53,00	53,00	720	1	0	0,00%	100,00%
		6	6	0	12	211	6	53,00	53,00	479.001.600	1	0	0,00%	100,00%
		6	6	6	18	789	11	65,00	---	6.402.373.705.728E+3		Sonuç Yok	???	???
7	PDKCalisma_13delik.dxf	0	7	0	7	160	10	362,69	362,69	5.040	1	0	0,00%	100,00%
		6	7	0	13	330	14	404,94	377,19	6.227.020.800		28	7,36%	92,64%
		6	7	6	19	2586	28	394,19	---	121.645.100.408.832E+3		Sonuç Yok	???	???
8	HE200_14delik.dxf	0	6	0	6	30	2	52,98	52,98	720	1	0	0,00%	100,00%
		8	6	0	14	493	12	60,00	60,00	87.178.291.200	1	0	0,00%	100,00%
		8	6	8	22	2943	24	69,00	---	112.400.072.777.761E+7		Sonuç Yok	???	???

verilerdir. Bu verileri elde etmek amacıyla kullanılan çizim dosyalarına ait görüntüler Ek Açıklamalar B’de verilmiştir.

Mevcut tüm varyasyonlara ait maliyet verileri, bu varyasyonlara ait delme sırası dizilimleri, geliştirilen yazılım tarafından maliyet hesaplama işleminde kullanılan algoritmalar tarafından hesaplanarak elde edilmiştir.

Elde edilen bu veriler ışığında görülmektedir ki, geliştirilen yazılım tarafından elde edilen optimum dizilimlere ait maliyetler, en düşük olan maliyetlere oldukça yakın sonuç vermektedir. Sekiz farklı çizim dosyası üzerinde yapılan yirmi dört farklı hesaplama sonucunda; genel olarak elde edilen yanılma payı %0 iken, 200x100x10DIKDORTGEN\_10delik.dxf çizim dosyasında iki yüzey için yapılan denemede elde edilen en yüksek yanılma payı %23,30 olarak bulunmuştur.

Geliştirilen yazılımın en büyük avantaj hızlı sonuç vermesidir.Çizelge 6.1’de verilen örneklerden 14 delik için tüm varyasyonların hesaplanması yaklaşık 24 saat civarında bir zaman almıştır. 14 delik üzerinde bulunan çizim örnekleri için tüm varyasyonların hesaplanması işlemi, varyasyon sayısının faktöriyel olarak yükselmesi nedeniyle hesaplanamamıştır. Şekil Ek B.9’da, Şekil Ek B.7’de verilen çizim dosyasına değişik koordinat ve çaplarda delikler eklenerek üç yüzey için toplam 50 adet delik sayısının elde edildiği örnek çizim dosyasını göstermektedir. Bu çizim dosyası, hazırlanan yazılım tarafından, Intel Q9400 Core 2 Quad 2,66 GHz (6Mb Cache) 1333Mhz FSB, 4Gb Ram bellek ve Microsoft Windows 7 (64 Bit) işletim sistemli bilgisayar tarafından incelendiğinde, 61816 adet değişiklik denemesi içerisinde 75 adet değişim gerçekleştirilerek optimum dizilime 260 milisaniyede ulaşılmıştır. Çok hızlı sonuç vermesi, yazılımın avantajlarından biri olarak sayılabilir. Dezavantajları arasında ise, kesin sonuca ulaşamama ihtimali gösterilebilir. Çizelge 6.1’de verilen örnekler göz önüne alındığında, delik sayısının artmasıyla hata payına sahip sonuç elde etme ihtimalinin azaldığı görülmüştür. Algoritmaların uygulandığı farklı 24 örneğin 3 adetinde hesaplama işlemine ait bütün varyasyonlar gerçekleştirilememiştir, 5 adet hesaplama işleminde ise en kısa süreye ulaşamamıştır. Buradan 21 örnekten 16 adet net doğru sonuç, yaklaşık %76,19 güvenilir sonuç alındığı anlamına gelebilir. Çizelge 6.1’de verilen “Güvenilirlik”



sütunundaki 21 adet değerin ortalaması alındığında ise %96,90 güvenilir sonuç verdiği de gözlemlenmektedir.

Çizelge 6.2. Tercih edilen algoritma kombinasyonlarına ait sonuçlar.

Çizim Dosyası	Veri Türü	Başlangıç Maliyeti (Saniye)	n-Opt->Kaydırma	n-Opt->2-Opt->Kaydırma	Kabul Edilen Sonuç
50 Delikli Çizim Dosyası	Değişim Deneme Sayısı	620,772	88.959	105.540	<b>105.540</b>
	Toplam Delme Maliyeti (Saniye)		372,644	369,304	<b>369,304</b>
	İşlem Süresi (Milisaniye)		427		
70 Delikli Çizim Dosyası	Değişim Deneme Sayısı	1036,416	192.017	28.688	<b>28.688</b>
	Toplam Delme Maliyeti (Saniye)		724,637	704,587	<b>704,587</b>
	İşlem Süresi (Milisaniye)		1.430		
90 Delikli Çizim Dosyası	Değişim Deneme Sayısı	1443,99	939.316	867.247	<b>867.247</b>
	Toplam Delme Maliyeti (Saniye)		869,339	861,752	<b>861,752</b>
	İşlem Süresi (Milisaniye)		6.467		
105 Delikli Çizim Dosyası	Değişim Deneme Sayısı	1831,858	1.124.304	1.259.791	<b>1.259.791</b>
	Toplam Delme Maliyeti (Saniye)		1067,05	1066,049	<b>1066,049</b>
	İşlem Süresi (Milisaniye)		9.979		
120 Delikli Çizim Dosyası	Değişim Deneme Sayısı	1952,016	1.268.739	895.211	<b>1.268.739</b>
	Toplam Delme Maliyeti (Saniye)		1170,28	1178,694	<b>1170,28</b>
	İşlem Süresi (Milisaniye)		10.789		

Çalışma kapsamında kullanılan üç adet algoritmanın tıkanma noktalarını aşmak ve daha düşük maliyetlere ulaşabilmek üzere bu üç algoritmanın değişik kombinasyonları ile değişik hesaplamalar yapılmış ve Çizelge 4.11’de ulaşılan sonuçlara göre en ideal deneme ve maliyet parametrelerine göre iki adet

kombinasyon belirlenmiştir. Böylece daha hızlı bir şekilde, ideale yakın sonuçlar elde edilebilmektedir. Seçilen algoritma kombinasyonları ile yapılan hesaplama sonuçları ve optimizasyon işlemine tabi tutulmadan önce kabul edilen başlangıç durumuna ait delme işlemi maliyetleri Çizelge 6.2’de verilmiştir. Görüldüğü üzere gerçekleştirilen optimizasyon işlemi ile öncelikle verilerin bir personel yardımıyla sisteme aktarılmasıyla geçen süre dikkate alınmadığında sadece delme işlemi için %40 civarında maliyet düşümü sağlanmıştır. Aynı zamanda belirtilen sonuçlara ulaşmak üzere örnek olarak 120 delik içeren çizim dosyası için her iki algoritma kombinasyonu ile inceleme işleminin tamamlanması ve bu iki kombinasyon arasında elde edilen toplam delme işlemi maliyetine dayalı karar verilerek kullanıcıya sonuçların aktarılması işlemi yaklaşık 11 sn sürmüştür. Gerçekleştirilen bu optimizasyon işlemi için harcanan 11 sn süreye karşılık delme işleminin tamamlanması için gereken süreden yaklaşık 782 sn kazanılmış olmaktadır.

Daha sonraki aşamalarda, kullanılan optimizasyon algoritmalarına ek olarak, Genetik Algoritmalar yardımıyla mevcut maliyet hesaplama yöntemlerinden de faydalanarak yeni algoritmalar ya da algoritma kombinasyonları denenebilir. Elde edilen veriler ile kıyaslanarak geliştirilen algoritmaların verimliliğini arttırmaya yönelik çalışmalar gerçekleştirilebilir.

Ayrıca bu çalışmada incelenen çizim dosyaları, tek bir düzlemde oluşturulan üç yüzeye ait bilgiler içermektedir. Bu çalışma kapsamında ağırlıklı olarak optimizasyon işlemine yönelik algoritmalar üzerinde çalışmalar gerçekleştirilmiştir. Çalışma kapsamında yüzey verileri algılanırken yüzeyleri seçme ve değiştirme işlemi kullanıcıya bırakılmıştır. Üç boyutlu profillere ait çizim dosyaları, teknik resim kuralları da dikkate alınarak yüzeylerin tam olarak algılanmasına yönelik geliştirmeler yapılabilir.

Geliştirilen yazılım tarafından kullanılan optimizasyon yönteminin, daha sonraki aşamalarda, birbiri ile kısmen bağımlı birden çok yüzey kullanılarak gerçekleştirilecek çalışmalara zemin oluşturacağı düşünülmektedir. Bu çalışma daha çok delme işlemi üzerinde yoğunlaşmış olmasına karşılık, farklı işlemlerin kullanıldığı değişik üretim alanlarına da adapte edilebilir.

## KAYNAKLAR

1. Wang, S., Zhang, H., Ma, F. and Wang, H., “Concurrent engineering-based CAD/CAM technology and application”, *Industrial Technology (ICIT'96), Proceedings of The IEEE International Conference*, Shanghai, 320-323 (1996).
2. Ismail, N., Lim, K. S. and Tan, C. F., “Modelling and machining of complex mechanical parts using CAD/CAM systems”, *Research and Development (SCORED 2003), Proceedings, Student Conference*, 183-187 (2003).
3. Johannes, M. S., Kuniholm, J. F., Cole D. G. and Clark, R. L., “Automated CAD/CAM - based nanolithography using a custom atomic force microscope”, *Automation Science And Engineering, IEEE Transactions*, 3 (3): 236-239 (2006).
4. Oshima, Y. and Morishita, T., “Development of educational material for manufacturing engineering using stereo vision and 3D CAD/CAM”, *Ubiquitous Robots And Ambient Intelligence (URAI) 2011 8th International Conference*, Incheon, 409-413 (2011).
5. Nagata, F., Yoshitake, S., Otsuka, A., Watanabe, K. and Habib, M. K., “CAM system without using robot language for an industrial robot RV1A”, *Industrial Electronics (ISIE), 2012 IEEE International Symposium*, Hangzhou, 1529-1534 (2012).
6. Sigl, S. and Mayer, H.A., “Hybrid evolutionary approaches to CNC drill route optimization”, *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference*, Vienna, 1: 905-910 (2005).
7. Zhu, G. Y., “Drilling path optimization based on swarm intelligent algorithm”, *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, Kunming, 193-196 (2006).
8. Adam, A., Abidin, A. F. Z., Ibrahim, Z., Husain, A. R., Yusof, Z. M and Ibrahim, I., “A particle swarm optimization approach to robotic drill route optimization”, *Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, Kota Kinabalu, Malaysia, 60 – 64 (2010).

9. Tsai, C. Y., Liu, C. H. and Wang, Y. C., “Application of genetic algorithm on IC substrate drilling path optimization”, *International Conference on Advanced Mechatronic Systems*, Tokyo, Japan, 441 – 446 (2012).
10. Karaçor, M. ve Keleş, K. “Otomasyon sistemlerinin bileşenleri”, *Endüstri ve Otomasyon*, 136: 16-21, (2008).
11. İnternet: Siemens “SIMATIC WinCC – Process visualization with Plant Intelligence” [http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure\\_simatic-wincc\\_en.pdf](http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-wincc_en.pdf) (2012).
12. Jadid, M. N., Idrees, M. M., “Cost estimation of structural skeleton using an interactive automation algorithm: a conceptual approach”, *Automation in Construction*, 16: 797 – 805 (2007).
13. Sukimin, Z. and Haron, H., “Geometric entities information for feature extraction of solid model based on DXF file”, *Information Technology (2008), ITSIM (2008), International Symposium*, Kuala Lumpur, 3: 1 – 5 (2008).
14. Lin, W., Nan, L. and Xiaohao, X., “The research of display system for surface surveillance radar”, *4th International conference on Computer Science & Education*, Nanning, 679 – 682 (2009).
15. Jin, Y., Liu, Y. Wei, J. and Guo, X., “The research on NURBS curve extraction method of new type numerical control machine”, *9th International Hybrid Intelligent Systems Conference*, Shenyang, 2: 13 – 17 (2009).
16. Pin, W. and Lan-zhi, S., “Design and realization of NC graphic programming system”, *Software Engineering (WCSE), 2010 Second World Congress*, Wuhan, 1: 26 – 29 (2010).
17. Cheng, J., Yan, E., Zhang, T. and Hu, X., “Development of 3d landslide dynamic modeling based on DSAF integration system”, *Bio-Inspired computing: Theories and Applications(BIC-TA), 2010 IEEE 5th International Conference*, Changsha, 1245 – 1250 (2010).
18. Wang, X., Liu, D, Tao, Y. and Cui, Yuxia, “An optimized path planning method for off-line programming of an industrial robot”, *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, Hangzhou, 3: 57 – 60 (2012).
19. Shang, W., Zhong, J. and Yan, Q., “Analysis of DXF file with an application to 3D graphic display”, *International conference on Information and Automation (ICIA)*, Shenyang, China, 611 – 615 (2012).

20. Yan, L. and Xing, Z., “Extraction and Reconstruction Technique of AutoCAD Graphic on Qt Platform”, *World Automation congress (WAC)*, 1 – 5 (2012).
21. İnternet: Digia, “Product–Digia Plc” <http://qt.digia.com/product/> (2012).
22. İnternet: AutoDesk “AutoCAD 2012 - DXF Reference” [http://images.autodesk.com/adsk/files/autocad\\_2012\\_pdf\\_dxf-reference\\_enu.pdf](http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf) (2012).
23. İnternet: Türk Dil Kurumu “Optimizasyon” [http://www.tdk.gov.tr/index.php?option=com\\_gts&arama=gts&guid=TDK.GTS.508f1a75960479.53312221&kelime=optimizasyon](http://www.tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.508f1a75960479.53312221&kelime=optimizasyon) (2012).
24. İnternet: Upatras “The T.S.P. Example” <http://students.ceid.upatras.gr/~papagel/project/tspprobl.htm> (2012).
25. İnternet: “2-Opt Exchange Algorithm” <http://www-e.uni-magdeburg.de/mertens/TSP/node3.html> (2012).
26. Eryavuz, M. ve Gencer, C., “Araç rotalama problemine ait bir uygulama”, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 6 (1): 139-155 (2001).
27. Toğan, V. ve Daloğlu, A., “Genetik algoritma ile üç boyutlu kafes sistemlerin şekil ve boyut optimizasyonu”, *İMO Teknik Dergisi*, 251: 3809-3825 (2006).
28. Uğur, A., “Path planning on a cuboid using genetic algorithms”, *Fifth Internation Conference on Machine Learning and Cybernetics*, 178: 3275-3287 (2008).
29. Jackson, J. Girard, A., Rasmussen, S. and Schumacher, C., “A combined tabu search and 2-opt heuristic for multiple vehicle routing”, *American Control Conference Marriott Waterfront*, Baltimore MD, 3842-3847 (2010).

**EK AÇIKLAMALAR A.**

**ÖRNEK VERİTABANI TABLO ÇİZELGELERİ**

Çizelge EK A.1. Örnek çizim dosyasına ait LINE veritabanı tablosu içeriği.

<b>ID</b>	<b>STARTX</b>	<b>STARTY</b>	<b>ENDX</b>	<b>ENDY</b>	<b>BOYX</b>	<b>BOYY</b>
1	61.94	81.09	166.94	81.09	105	0
2	199.94	73.09	271.94	73.09	72	0
3	199.94	65.09	271.94	65.09	72	0
4	199.94	57.09	199.94	81.09	0	24
5	223.94	57.09	223.94	81.09	0	24
6	247.94	57.09	247.94	81.09	0	24
7	100.05	57.09	100.05	81.04	0	23.95
8	124.05	57.09	124.05	81.04	0	23.95
9	100.05	73.09	124.05	73.09	24	0
10	166.94	81.09	271.94	81.09	105	0
11	61.94	354.09	271.94	354.09	210	0
12	1	0	1	1	0	1
13	0	1	1	1	1	0
14	0	0	0	1	0	1
15	20.06	17.87	20.06	82.13	0	64.26
16	14.06	0	64.06	0	50	0
17	64.06	0	64.06	2.35	0	2.35
18	14.06	100	64.06	100	50	0
19	27.88	6.83	59.92	9.39	32.04	2.56
20	14.06	100	64.06	100	50	0
21	64.06	97.65	64.06	100	0	2.35
22	27.88	90.61	59.92	93.17	32.04	2.56
23	-920002.24	3914294.05	-919877.24	3914294.05	125	0
24	-919877.24	3913994.05	-919877.24	3914000.23	0	6.18
25	-920002.24	3913994.05	-919877.24	3913994.05	125	0
26	-920002.24	3913994.05	-920002.24	3914000.23	0	6.18
27	-919945.14	3914023.27	-919945.14	3914264.84	0	241.57
28	-919934.34	3914023.27	-919934.34	3914264.84	0	241.57
29	-920002.24	3913994.05	-919970.99	3913994.05	31.25	0
30	-919925.04	3914006.66	-919882.84	3914012.57	42.2	5.91
31	-919996.64	3914006.66	-919954.44	3914012.57	42.2	5.91
32	-919877.24	3914287.88	-919877.24	3914294.05	0	6.17
33	-920002.24	3914294.05	-919877.24	3914294.05	125	0
34	-920002.24	3914287.88	-920002.24	3914294.05	0	6.17
35	-919996.64	3914275.54	-919954.44	3914281.45	42.2	5.91
36	-920002.24	3914294.05	-919970.99	3914294.05	31.25	0
37	-919925.04	3914275.54	-919882.84	3914281.45	42.2	5.91
38	-919762.3	3913994.05	-919762.3	3914294.05	0	300
39	-919762.3	3913994.05	-918827.3	3913994.05	935	0
40	-919762.3	3914294.05	-918827.3	3914294.05	935	0
41	-918827.3	3913994.05	-918827.3	3914294.05	0	300
42	-919762.3	3914287.88	-918827.3	3914287.88	935	0
43	-919762.3	3914275.54	-918827.3	3914275.54	935	0
44	-919762.3	3914000.4	-918827.3	3914000.4	935	0
45	-919762.3	3914012.74	-918827.3	3914012.74	935	0
46	-919762.3	3913725.05	-918827.3	3913725.05	935	0

Çizelge EK A.1. (devam ediyor)

<b>47</b>	-918827.3	3913600.05	-918827.3	3913725.05	0	125
<b>48</b>	-919762.3	3913600.05	-918827.3	3913600.05	935	0
<b>49</b>	-919762.3	3913600.05	-919762.3	3913725.05	0	125
<b>50</b>	-920171.9	3913376.62	-920171.9	3913443.67	0	67.05
<b>51</b>	-920192.41	3913357.99	-920190.53	3913357.99	1.88	0
<b>52</b>	-920192.41	3913357.99	-920190.53	3913357.99	1.88	0
<b>53</b>	-920192.41	3913357.99	-920192.41	3913359.87	0	1.88
<b>54</b>	-920192.41	3913357.99	-920192.41	3913359.87	0	1.88
<b>55</b>	-918621.9	3913695.2	-918612.51	3913695.2	9.39	0
<b>56</b>	-918621.9	3913695.2	-918612.51	3913695.2	9.39	0



Çizelge EK A.2. Örnek çizim dosyasına ait CIRCLE veritabanı tablosu içeriği.

<b>ID</b>	<b>STARTX</b>	<b>STARTY</b>	<b>RADIUS</b>	<b>DIAMETER</b>
1	-919682.3	3914207.14	12.5	25
2	-919682.3	3914087.14	12.5	25
3	-919602.3	3914147.14	9	18
4	-918892.3	3914207.14	9	18
5	-918892.3	3914087.14	12.5	25
6	-919369.8	3913705.05	9	18
7	-919294.8	3913705.05	10	20
8	-919294.8	3913620.05	10	20
9	-919369.8	3913620.05	9	18
10	-919522.3	3914207.14	12.5	25
11	-919522.3	3914087.14	12.5	25
12	-919219.8	3913705.05	10	20
13	-919219.8	3913620.05	10	20

Çizelge EK A.3. Örnek çizim dosyasına ait PROFIL veritabanı tablosu içeriği.

<i><b>YUZEYNO</b></i>	<i><b>USTY</b></i>	<i><b>ALTY</b></i>	<i><b>SOLX</b></i>	<i><b>SAGX</b></i>	<i><b>XREF</b></i>	<i><b>YREF</b></i>
<b>1</b>	300	0	0	935	-919762.3	3913994.05
<b>2</b>	125	0	0	935	-919762.3	3913600.05

Çizelge EK A.4. Örnek çizim dosyasına ait DELİK veritabanı tablosu içeriği.

<b>ID</b>	<b>STARTX</b>	<b>STARTY</b>	<b>RADIUS</b>	<b>DIAMETER</b>	<b>YUZEYNO</b>
1	80	213.09	12.5	25	1
2	80	93.09	12.5	25	1
3	160	153.09	9	18	1
4	870	213.09	9	18	1
5	870	93.09	12.5	25	1
6	240	213.09	12.5	25	1
7	240	93.09	12.5	25	1
8	392.5	105	9	18	2
9	467.5	105	10	20	2
10	467.5	20	10	20	2
11	392.5	20	9	18	2
12	542.5	105	10	20	2
13	542.5	20	10	20	2

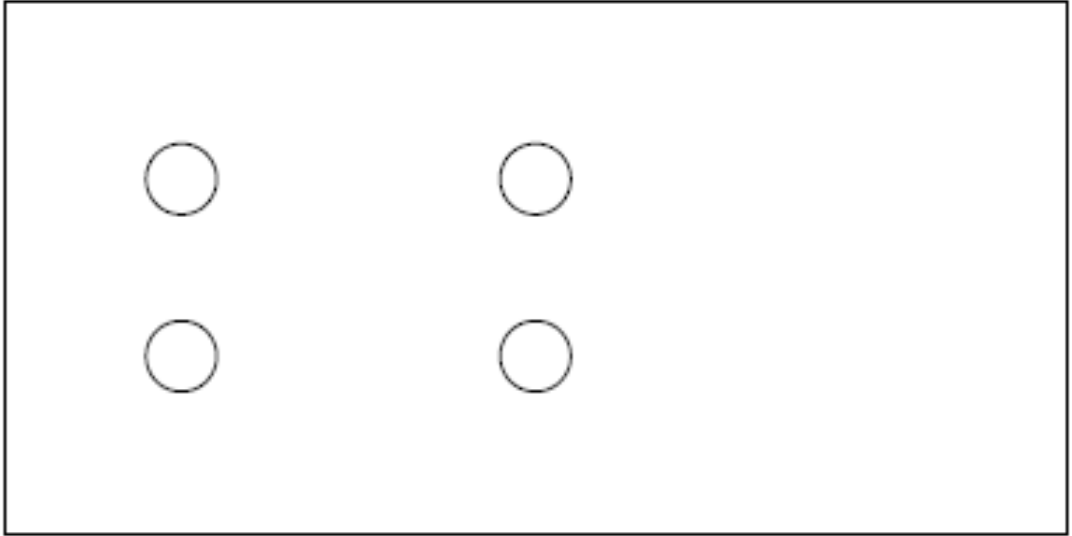
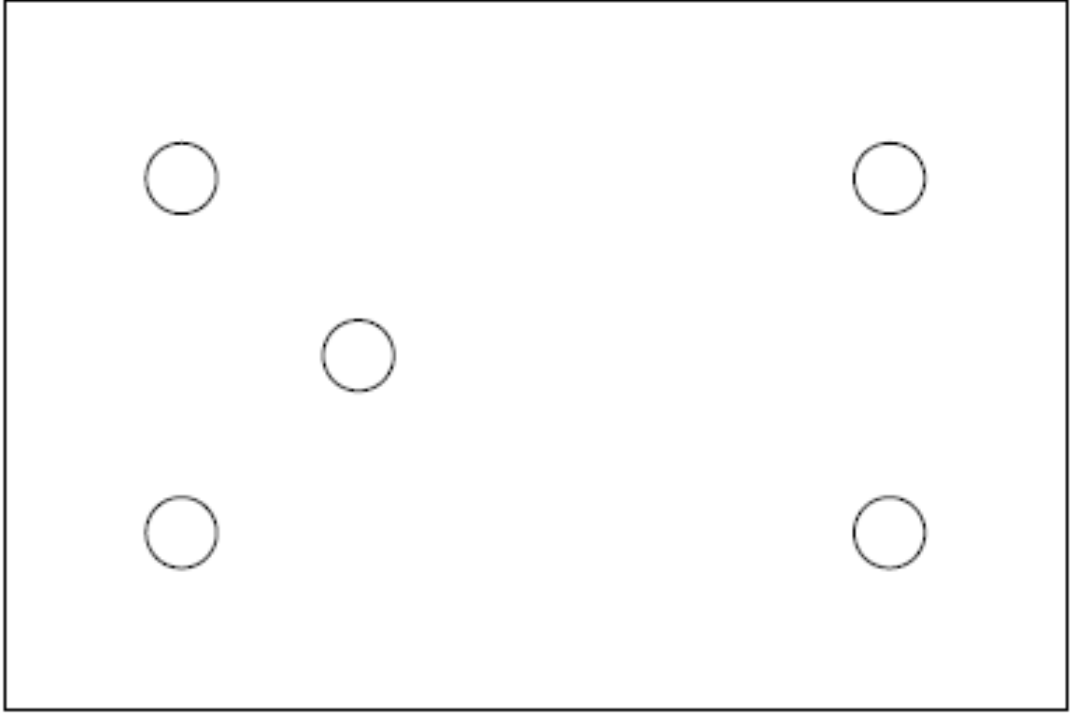
Çizelge EK A.5. Örnek çizim dosyasına ait OPTIMIZED veritabanı tablosu içeriği

	<i>ID</i>	<i>XCOORD</i>	<i>YCOORD</i>	<i>RADIUS</i>	<i>DIAMETER</i>	<i>YUZEYNO</i>
<b>GÖRECELİ (RELATIVE) ADRESLEME</b>	1	160	153.09	9	18	2
	2	-80	-60	12.5	25	2
	3	0	120	12.5	25	2
	4	160	0	12.5	25	2
	5	0	-120	12.5	25	2
	6	152.5	-73.09	9	18	3
	7	0	0	9	18	1
	8	0	85	9	18	1
	9	0	0	9	18	3
	10	75	0	10	20	1
	11	0	0	10	20	3
	12	0	-85	10	20	3
	13	0	0	10	20	1
	14	75	0	10	20	3
	15	0	0	10	20	1
	16	0	85	10	20	1
	17	0	0	10	20	3
	18	327.5	108.09	9	18	2
	19	0	-120	12.5	25	2
<b>MUTLAK (ABSOLUTE) ADRESLEME</b>	<i>ID</i>	<i>XCOORD</i>	<i>YCOORD</i>	<i>RADIUS</i>	<i>DIAMETER</i>	<i>YUZEYNO</i>
	1	160	153.09	9	18	2
	2	80	93.09	12.5	25	2
	3	80	213.09	12.5	25	2
	4	240	213.09	12.5	25	2
	5	240	93.09	12.5	25	2
	6	392.5	20	9	18	3
	7	392.5	20	9	18	1
	8	392.5	105	9	18	1
	9	392.5	105	9	18	3
	10	467.5	105	10	20	1
	11	467.5	105	10	20	3
	12	467.5	20	10	20	3
	13	467.5	20	10	20	1
	14	542.5	20	10	20	3
	15	542.5	20	10	20	1
	16	542.5	105	10	20	1
	17	542.5	105	10	20	3
	18	870	213.09	9	18	2
19	870	93.09	12.5	25	2	

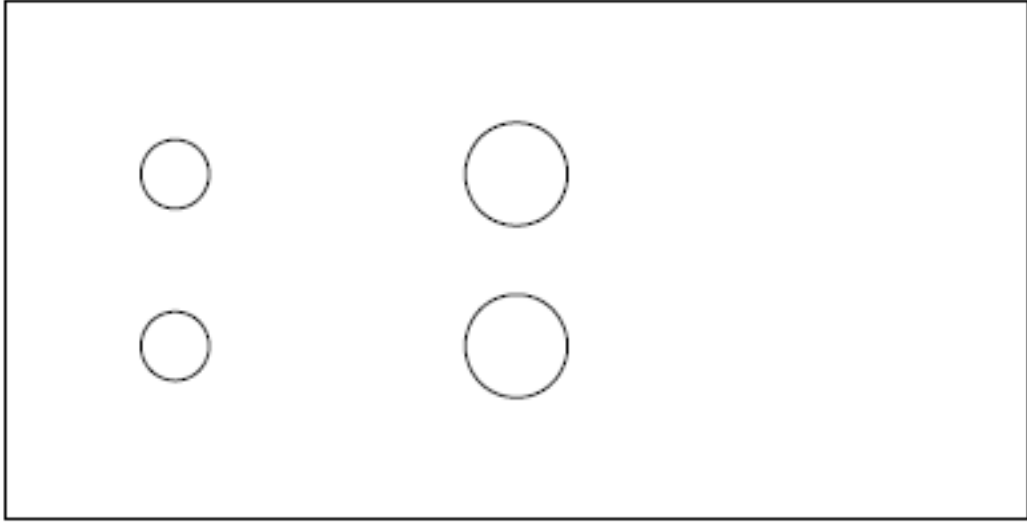
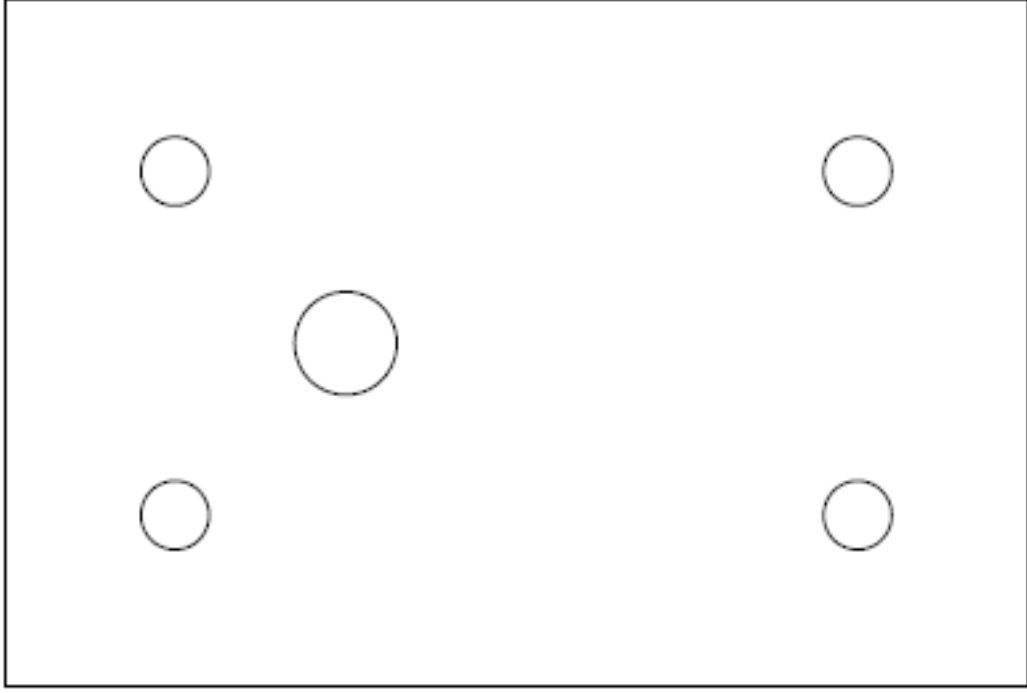
**EK AÇIKLAMALAR B.**

**ÖRNEK ÇİZİM DOSYASI İÇERİKLERİ**



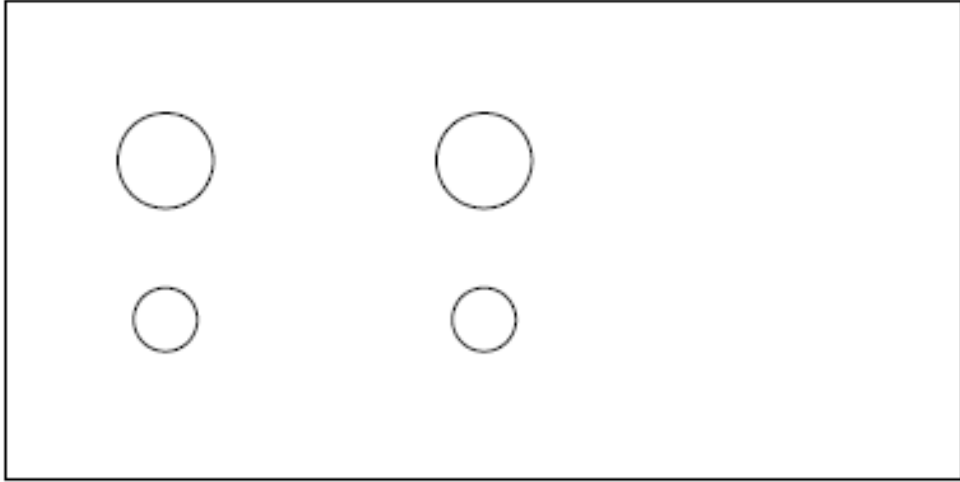
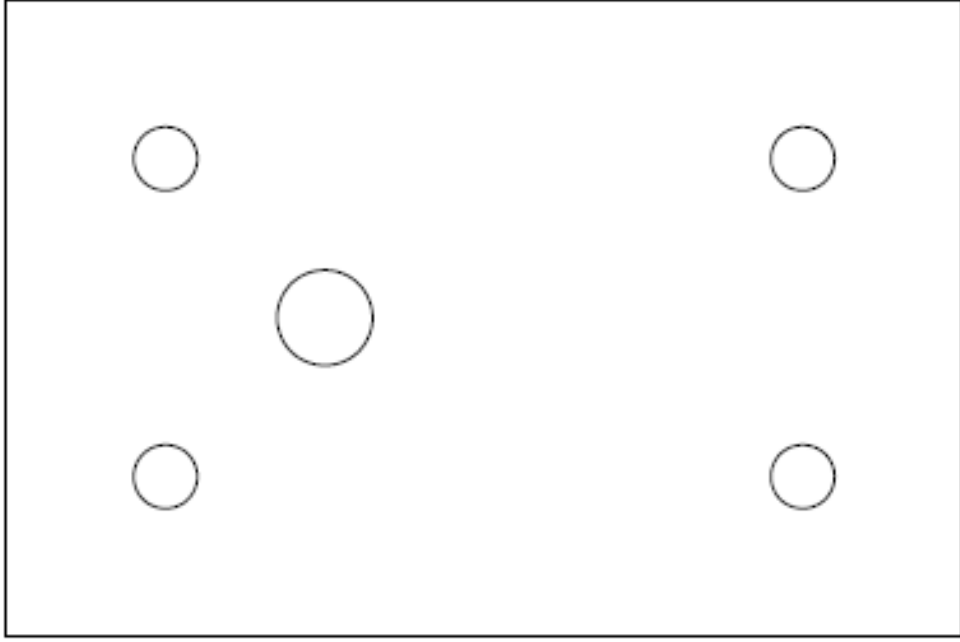


Şekil EK B.2. Cizim1\_ayniboy\_9delik.dxf çizim dosyası içeriği.

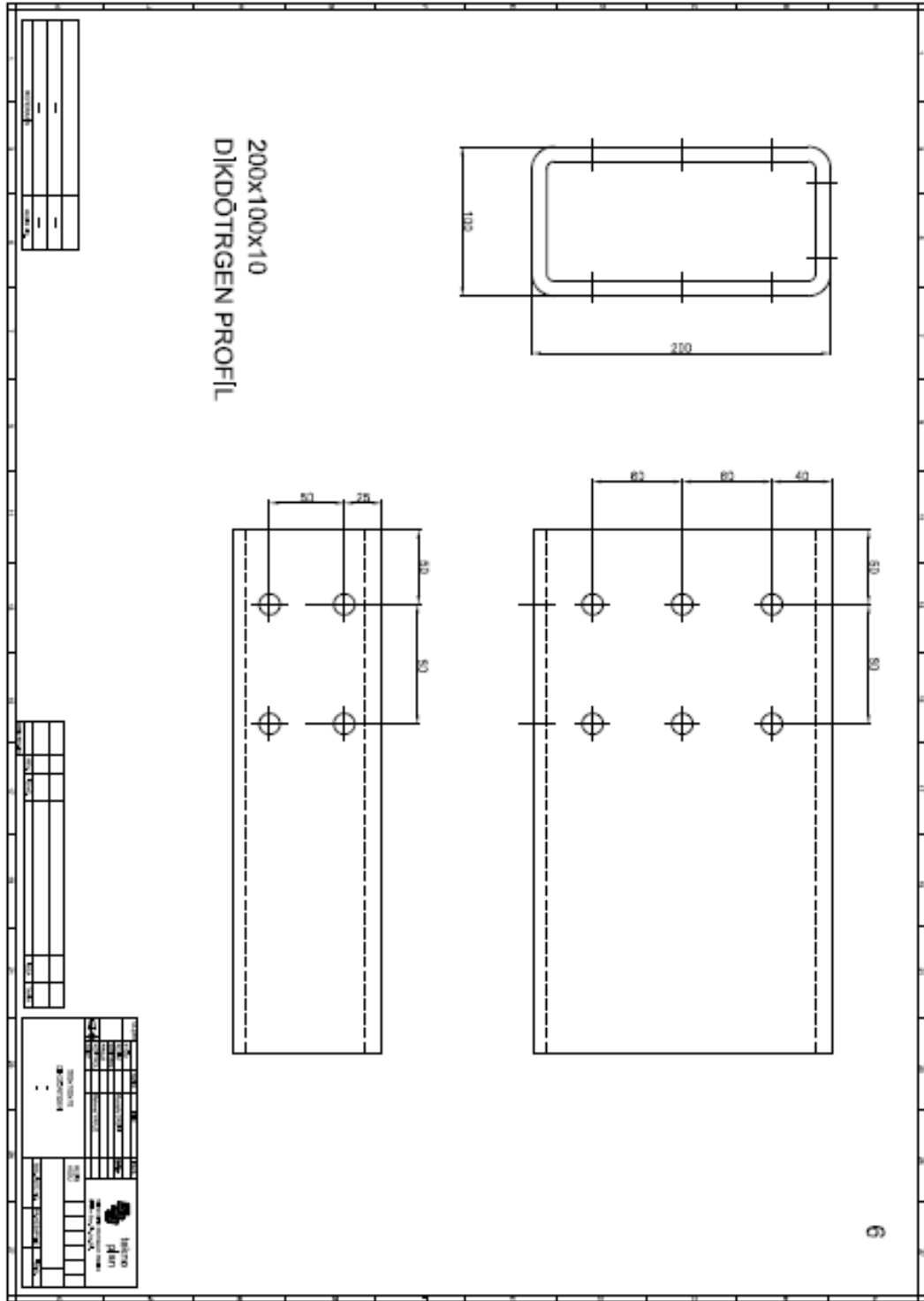


Şekil EK B.3. Cizim2\_3farkliboy\_9delik.dxf çizim dosyası içeriği.

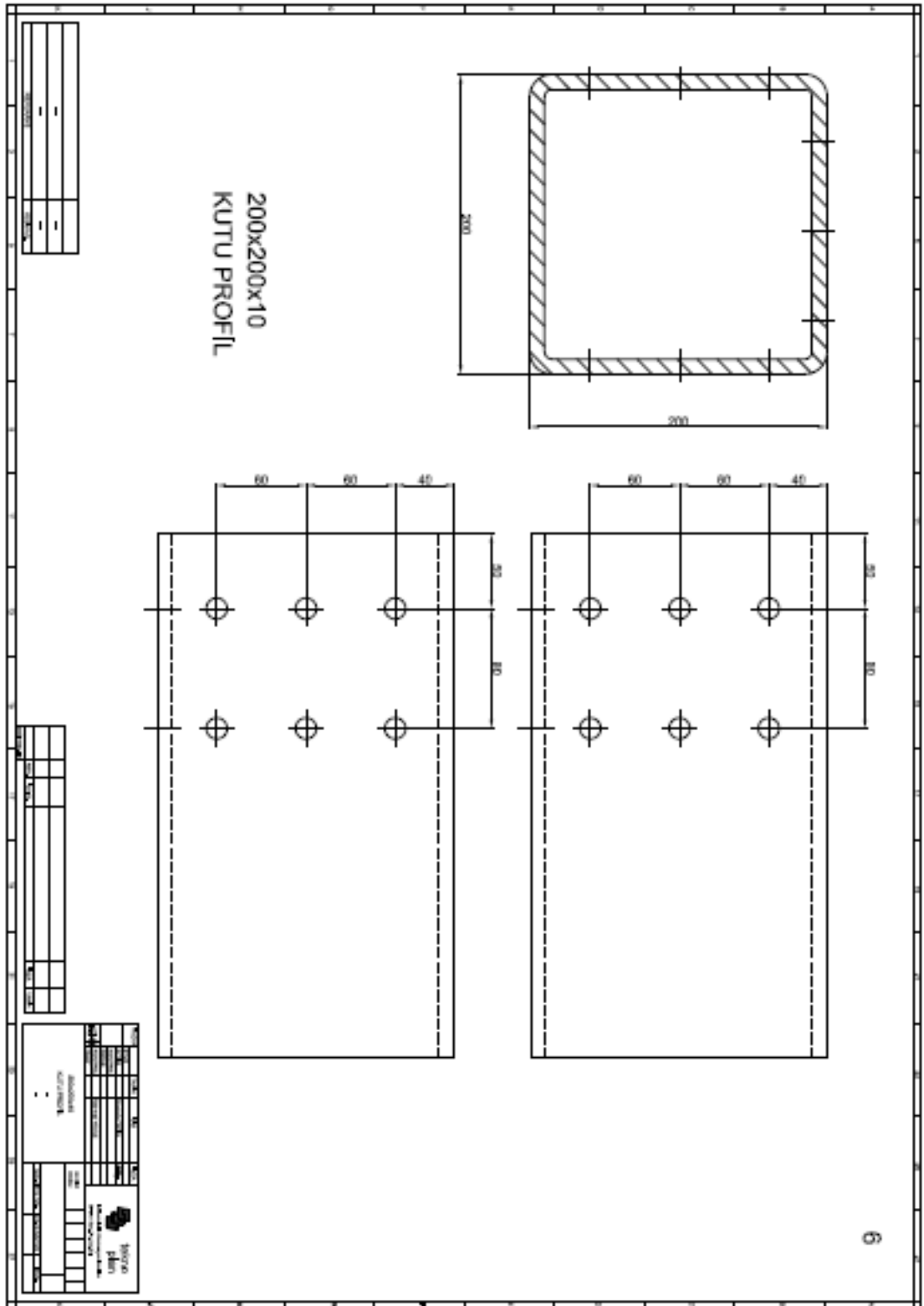


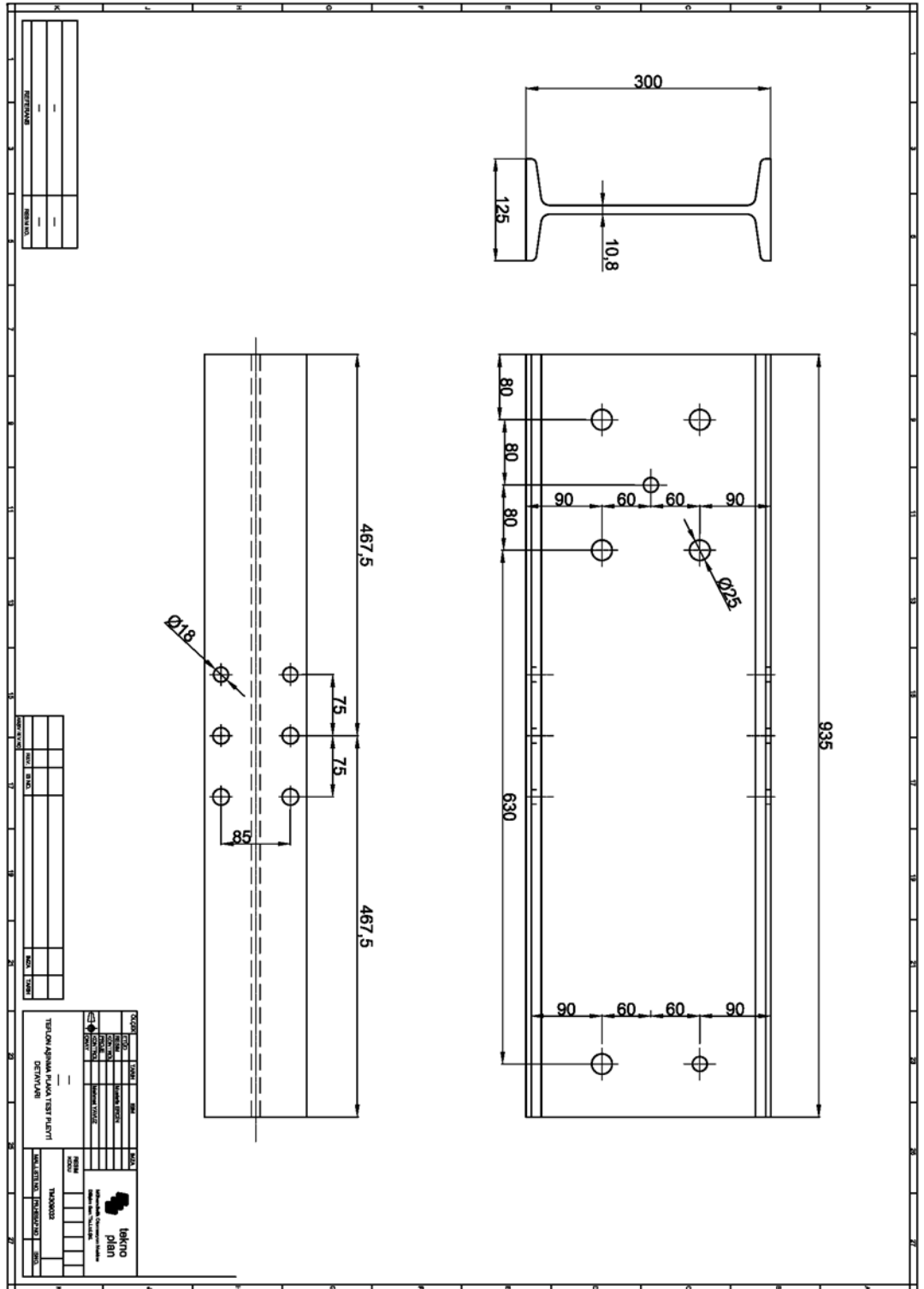


Şekil EK B.4. Cizim3\_3farkliboy\_9delik.dxf çizim dosyası içeriği.

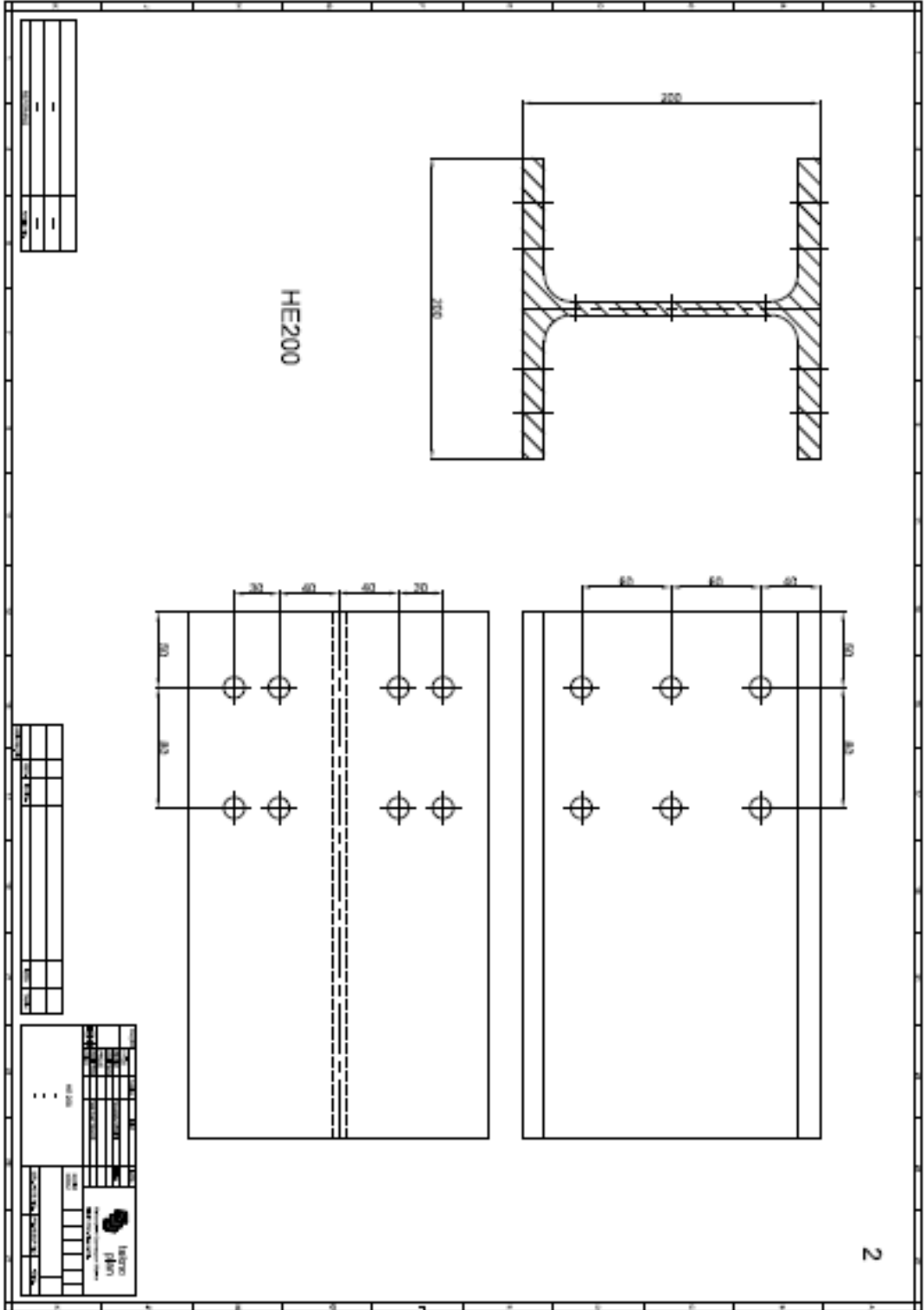


Şekil EK B.5. 200x100x10DIKDORGEN\_10delik.dxf çizim dosyası içeriği.

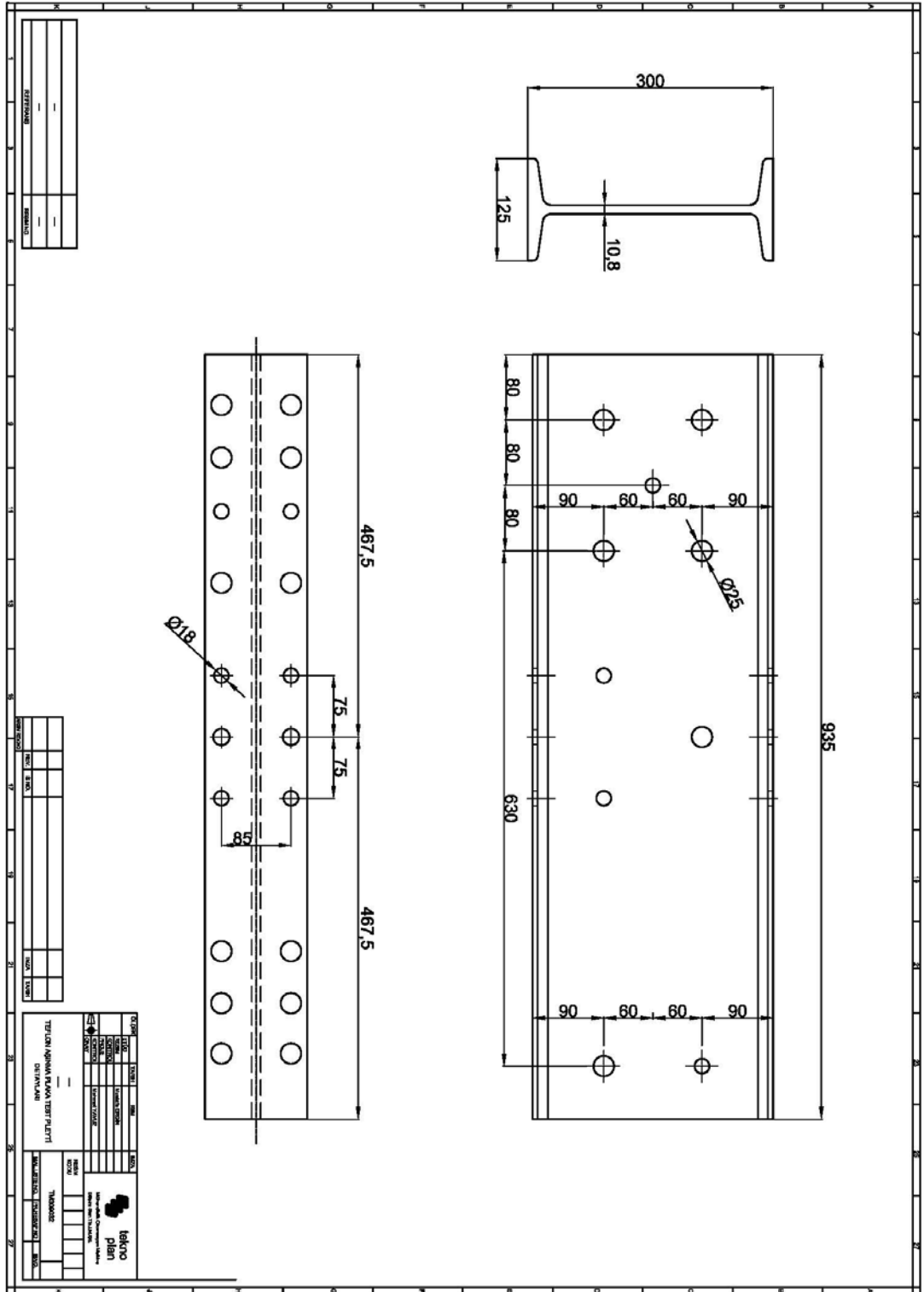




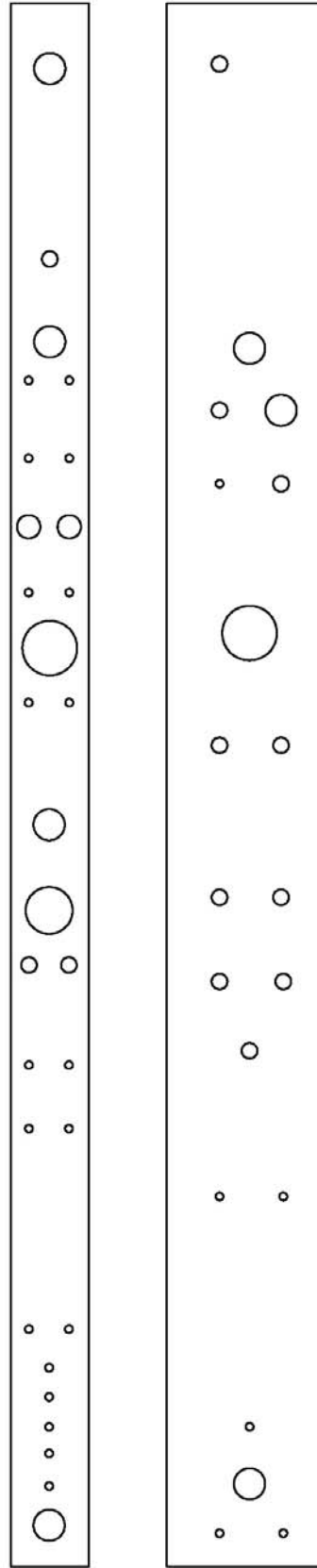
Şekil EK B.7. PDKCalisma\_13delik.dxf çizim dosyası içeriği.



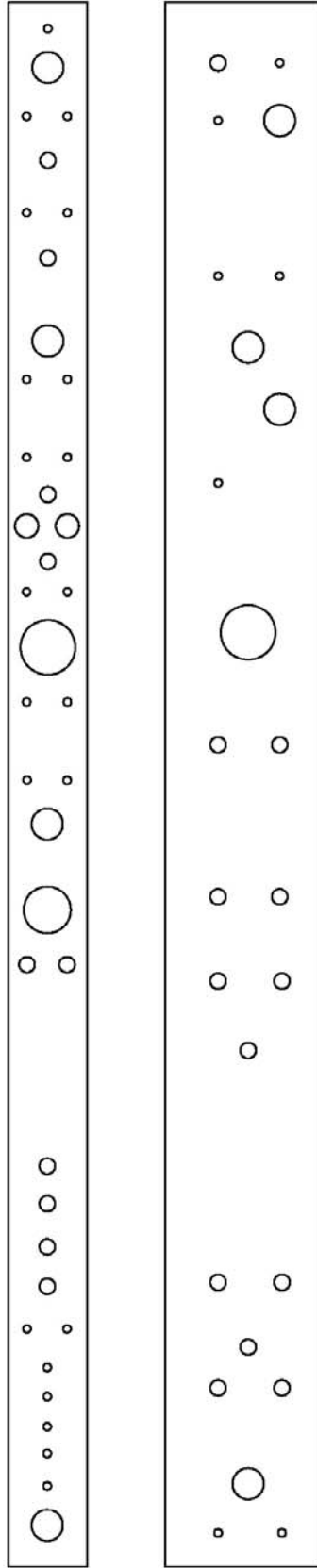
Şekil EK B.8. HE200\_14delik.dxf çizim dosyası içeriği.



Şekil EK B.9. PDKCalisma\_13delik.dxf çizim dosyasından üretilen 50 delikli çizim.

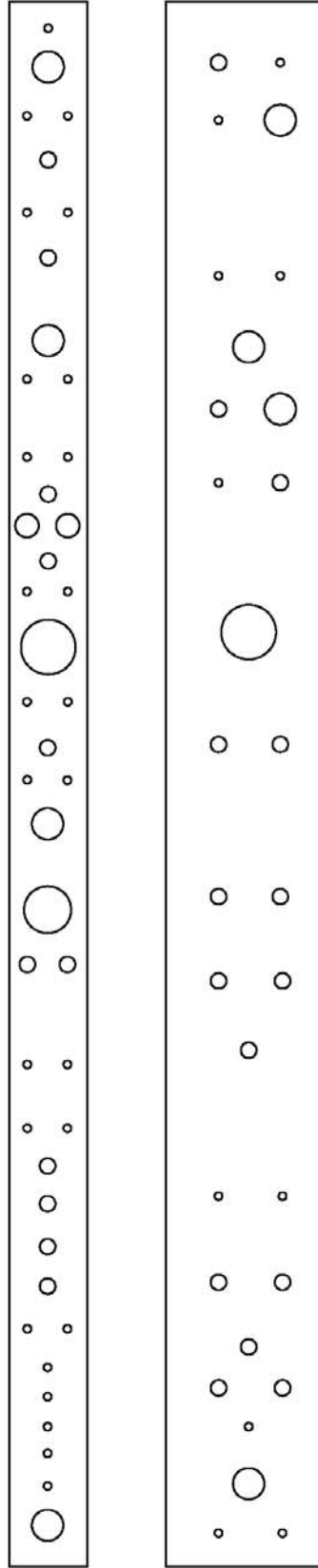


Şekil EK B.10. 70Delik.dxf çizim dosyası içeriği.

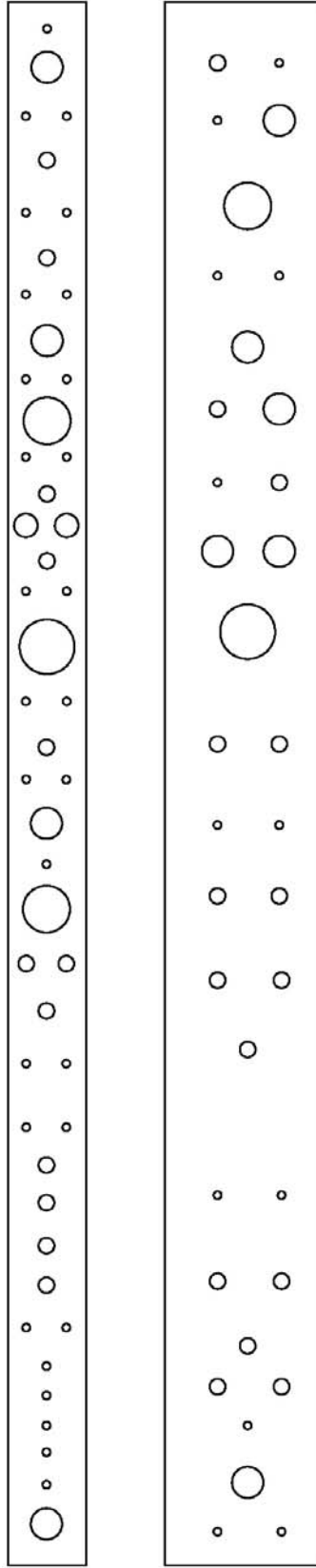


Şekil EK B.11. 90Delik.dxf çizim dosyası içeriği.





Şekil EK B.12. 105Delik.dxf çizim dosyası içeriği.

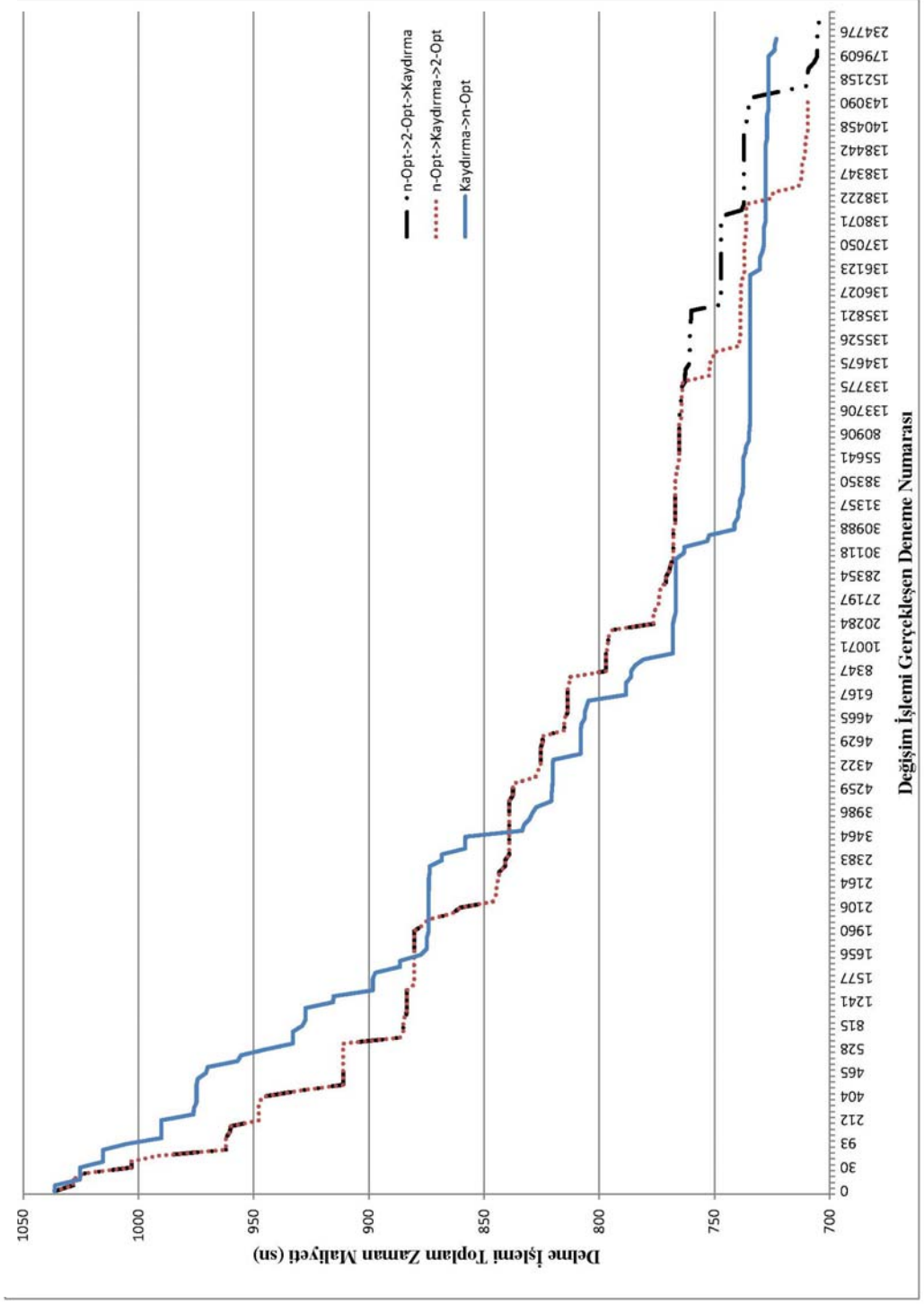


Şekil EK B.13. 120Delik.dxf çizim dosyası içeriği.

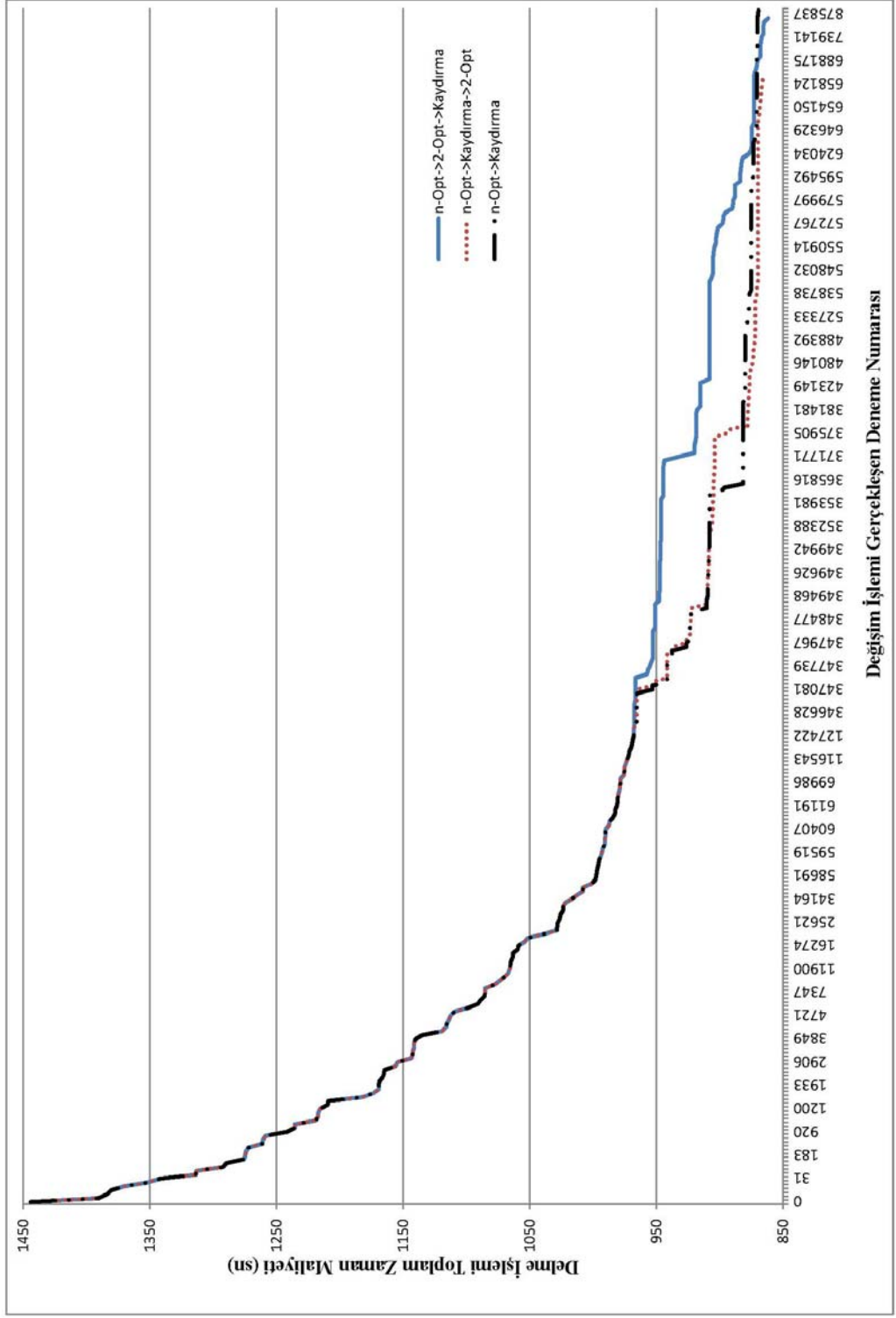
**EK AÇIKLAMALAR C.**

**İNCELEME İŞLEMİ SONUCU ELDE EDİLEN DEĞİŞİM GRAFİKLERİ**

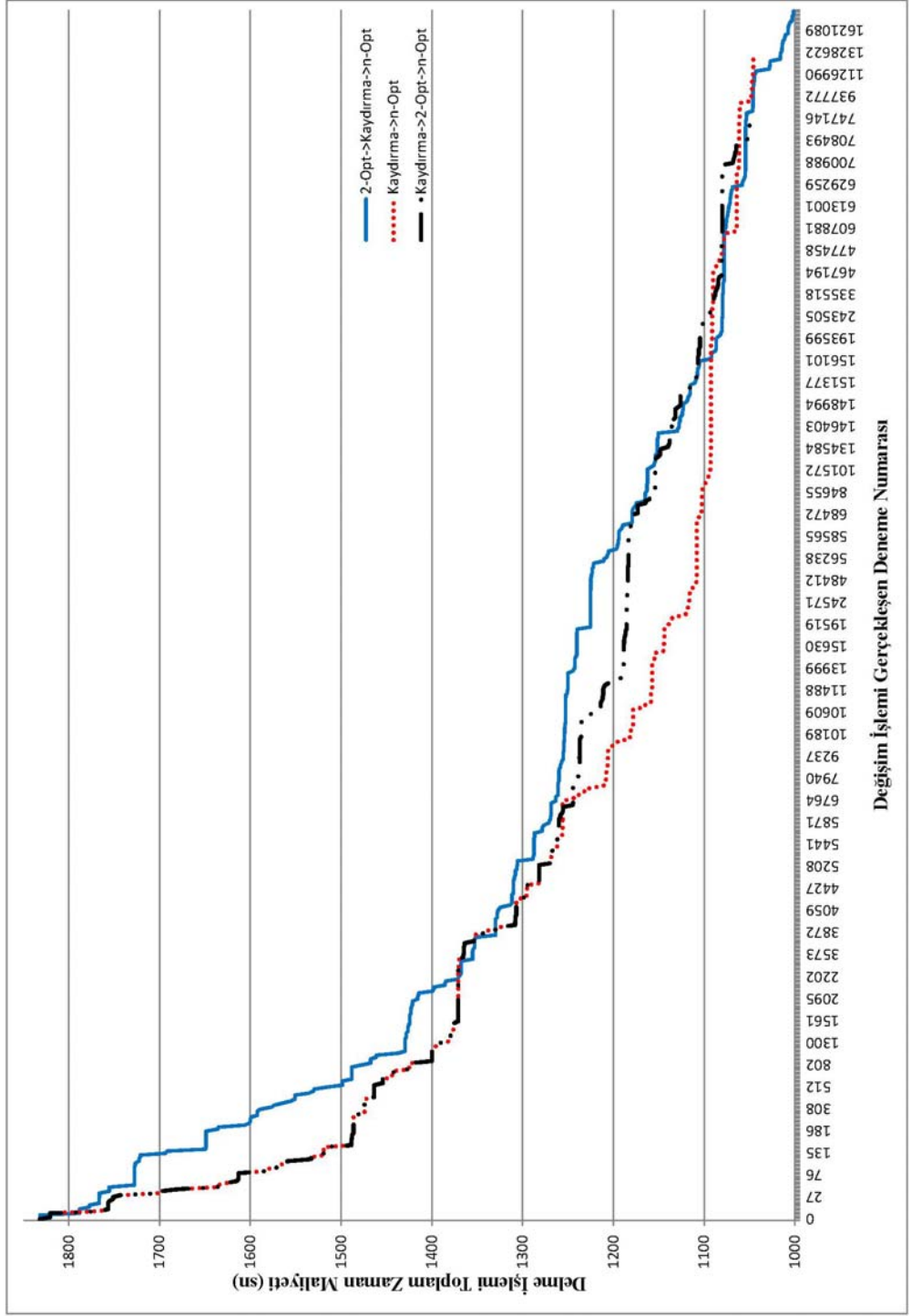
Şekil EK C.1. 70Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon.



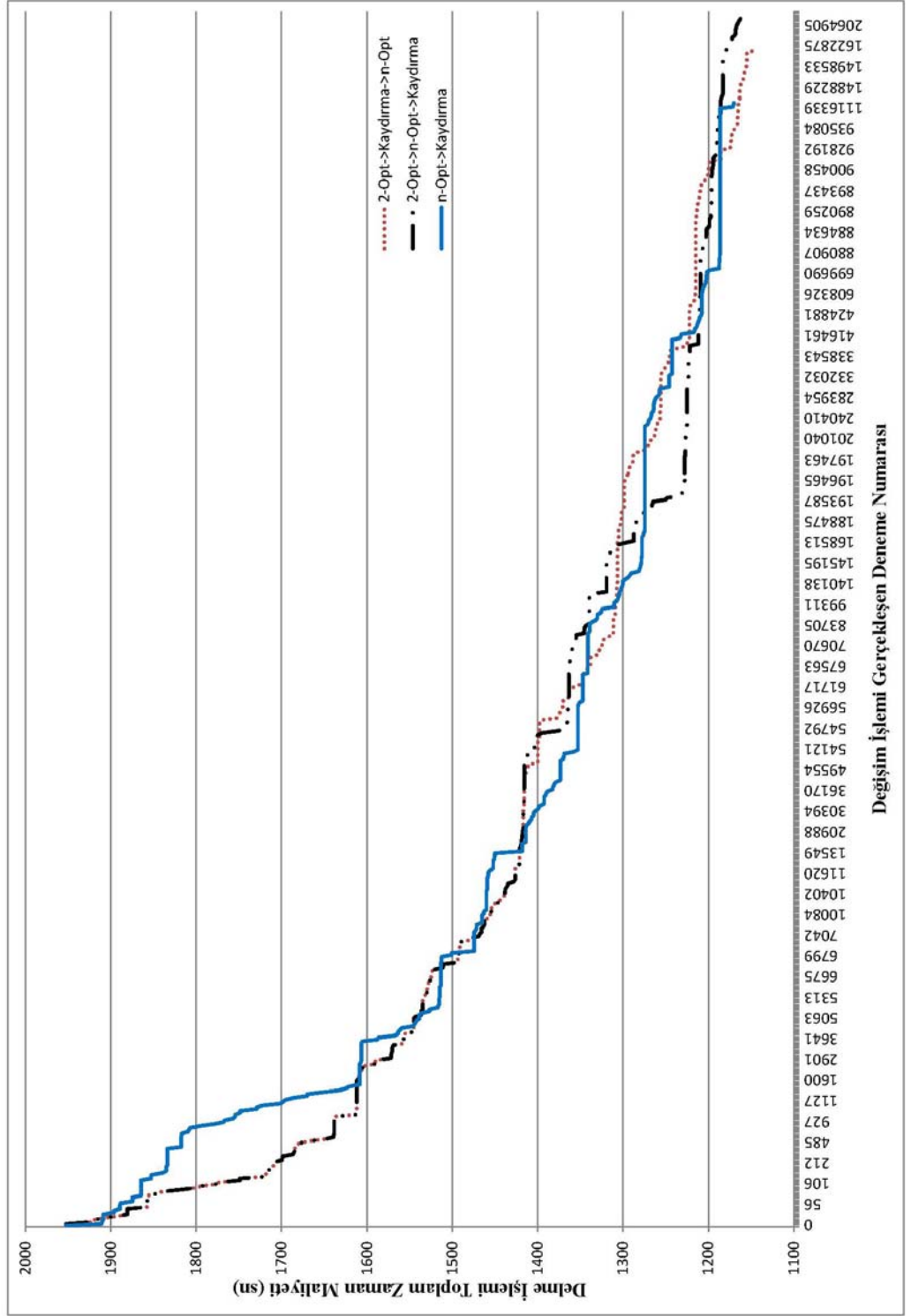
Şekil EK C.2. 90Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon.



Şekil EK C.3. 105Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon.



Şekil EK C.4. 120Delik.dxf çizim dosyası için en iyi sonucu veren üç kombinasyon.



## ÖZGEÇMİŞ

Mustafa YILDIZ 1980’de Manisa’da doğdu; ilköğrenimini sırasıyla Diyarbakır ve İstanbul’da, orta öğrenimini İstanbul’da tamamladı. Gelibolu Lisesi’nde Fen Alanından mezun olduktan sonra 1997 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik Öğretmenliği Bölümü’ne girdi. 2002 yılında “iyi” derece ile mezun olduktan sonra Gelibolu Ticaret Meslek ve Anadolu Ticaret Meslek Lisesi Bilgisayar Bölümünde Öğretmen olarak göreve başladı ve Araç Anadolu Sağlık Meslek Lisesi’nde görevine devam etmektedir. Halen; 2009 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda başlamış olduğu yüksek lisans programını sürdürmektedir.

### ADRES BİLGİLERİ

Adres: Araç Anadolu Sağlık Meslek Lisesi.  
Yeşil Mahalle No: 36 37800  
Araç / KASTAMONU

Tel / Faks: (366) 362 15 51

E-posta: wiper.gazi@gmail.com