



Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü
İşletme Anabilim Dalı Programı

EKONOMİK TEST TASARIMI VE OPTİMİZASYONU İÇİN KISIT PROGRAMLAMA KULLANILMASI

Onur KOYUNCU

Doktora Tezi

Ankara, 2007

EKONOMİK TEST TASARIMI VE OPTİMİZASYONU İÇİN KISIT PROGRAMLAMA KULLANILMASI

Onur KOYUNCU

Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü
İşletme Anabilim Dalı Programı

Doktora Tezi

Ankara, 2007

KABUL VE ONAY

Onur Koyuncu tarafından hazırlanan “Ekonomik Test Tasarımı Ve Optimizasyonu İçin Kısıt Programlama Kullanılması” başlıklı bu çalışma, 27.11.2007 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafında Doktora Tezi olarak kabul edilmiştir.



Prof. Dr. Ülkü ŞİŞİK (Başkan)



Doç. Dr. Ş. Armağan TARIM (Danışman)



Dr. Steven PRESTWICH (İkinci Danışman)



Doç. Dr. Mahmut ARSLAN



Dr. Ayşegül TAŞ

Yukarıda geçen imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. İrfan Çakın
Enstitü Müdürü

BİLDİRİM

Hazırladığım tezin/raporun tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin/raporumun kağıt ve elektronik kopyalarının Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım:

- ✓ Tezimin/Raporumun tamamı her yerden erişime açılabilir.
- ☐ Tezim/Raporum sadece Hacettepe Üniversitesi yerleşkelerinden erişime açılabilir.
- ☐ Tezimin/Raporumun yıl süreyle erişime açılmasını istemiyorum. Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir.



27.11.2007

Onur KOYUNCU

TEŞEKKÜR

Çalışmanın her aşamasında hem yol gösterme, hem değerli fikirleriyle katkıda bulunma, hem de beni şevklendirme konusunda eşsiz katkıları bulunan sevgili meslektaşlarım ve değerli hocalarım olan Ş. Armağan TARIM ve Steven D. PRESTWICH'e,

Altı ay boyunca bana yeni bir iş yeri ve yuva olan ve bu çalışmadaki fikirlerin temellerini oluşturmamı sağlayan ve bağımsız çalışmalarıyla bu çalışmanın şekillenmesinde girdileri bulunan başta James LITTLE, Brahim HNICH, Roberto ROSSI ve Eugene FREUDER olmak üzere tüm Cork Constraint Computation Center ve başta Suresh GOYAL olmak üzere tüm Bell Labs ekibini kapsayan CTVR grubuna,

Tez süreci boyunca her türlü derdimde yanımda olan, bana moral veren, motivasyonumu kaybettiğim zamanlarda tekrar çalışma azmini bulmama yardım eden ailem Uğur ve Nurdan KOYUNCU, Özün ve Volkan YILDIRIM ve Reyya ADVAN ile arkadaşlarım Suphi Taylan TATAR, Can Tekin KUŞHAN, Koray Kaan ÖZDEMİR ve Can Tekgöz GÖKDAL'a,

Fikirleri, eleştirileri ve destekleri için mesai arkadaşlarım Onur Alper KILIÇ, Cem SAYIN, Mehmet Akif AYARLIOĞLU ve Taner SİĞİNDİ'ya,

Akademik hayatı seçmemde çok büyük rol ve emekleri bulunan değerli hocalarım Hasan Işın DENER ve Ülkü ŞİŞİK'e,

Başta oda arkadaşlarım olan David A. BURKE ve Süleyman İÇ olmak üzere tüm mesai arkadaşlarıma ve ismini sayamadığım tüm dostlarıma teşekkür etmeyi bir borç bilirim.

ÖZET

Koyuncu, Onur. Ekonomik Test Tasarımı ve Optimizasyonu İçin Kısıt Programlama Kullanılması, Doktora Tezi, Ankara, 2007.

Bu çalışma, ürün ve bileşenlerinin tek tek ya da bir arada test edilmesi durumunda ortaya çıkan test akışları, test maliyetleri ve ilgili hesapları inceleyerek optimal test ve tamir makineleri seçimi ve optimal test birleşimlerini seçmek suretiyle test stratejileri için bir optimizasyon çerçevesi oluşturmayı amaçlamaktadır. Böylece hangi test ve test ekipmanı birleşimi ile belirli bir kalite düzeyinden ödün vermeden en ekonomik test tasarımı ve stratejisinin uygulanması gerektiğine dair rakamsal sonuçlar elde edilerek test uygulayan işletmeler için iktisadi bir karar destek aracı ortaya konmuş olmaktadır. Çalışma özel olarak elektrik elektronik sektöründe üretilen kart, panel, devre benzeri ürünleri kapsayacak şekilde oluşturulmuştur. Bu amaçla doğrusal programlama (LP), karışık tam sayı programlama (MIP) ve kısıt programlama (CP) teknikleri ile test akışı ve stratejiyi belirleyecek değişkenlerin seçimini yapacak optimizasyon modelleri geliştirilmiş ve açıklanmıştır. Çalışma test probleminin temellerini, hesap prensiplerini ortaya koyan ve hem tek özellikli, hem de çok özellikli test süreçleri için hesap ve optimizasyon prensiplerini belirlemektedir. Tek özellikli test, çok özellikli seri (bağımsız) test ve çok özellikli etkileşimli test süreçleri için geliştirilen modeller test tasarımı alanında daha önce bu çalışmadaki şekliyle ele alınmamış olup, optimizasyon yazılımları aracılığıyla çözülmüş ve elde edilen sonuçların pratik düz mantık yöntemlerine kıyasla daha karlı ve rasyonel çözümler önerdiği saptanmıştır. Literatürde bu spesifik alanla ilgili bir çalışmanın henüz bulunmuyor olması, sonuçların başka çalışmalarla karşılaştırılmasına imkan vermemektedir.

Anahtar Sözcükler : Test akışı, Test tasarımı, Test optimizasyonu, Test stratejisi, Test seçimi, Makine seçimi, Bağımsız test, Etkileşimli test, Kısıt programlama

ABSTRACT

Koyuncu, Onur. Economic Test Design and Application of Constraint Programming for Optimizing Test Design, Ph.D. Dissertation, Ankara, 2007.

This study aims to provide an optimization framework for obtaining optimal testing strategies by presenting the fundamental mathematical structures of test flows, test costs and related calculations which occur when a unit or components of a unit are tested independently or concurrently. Result of this attempt is a group of models which act as decision tools calculating and presenting numerical results helping the companies which apply tests to their products come up with economic solutions on which tests and testing equipment to choose (namely a test strategy) without sacrificing the level of product quality. This study has been put together specifying examples on the electronic products, examples of which can be listed as cards, boards, panels, circuits and related product types. Within this scope methods like linear programming (LP), mixed integer programming (MIP) and constraint programming (CP) have been applied to obtain mathematical equation sets and models which shall help describe the test flows and calculations and which shall decide on the variables of test strategy selection such as what tests and equipments to use. This study provides the basic theory on designing the test process, as well as the mathematical basis for test optimization. Especially the models built for optimizing single attribute test, multiple attribute serial (independent) test and multiple attribute interconnected (concurrent) test processes are the first examples in testing literature with the perception provided on this study. They were solved using numerous optimization softwares and it was found that the resulting strategies were far better than the general rule of thumb selections. Since no work in the literature has been realized on this specific issue so far, a comparison was not possible.

Key Words: Test flow, Test design, Test optimization, Test strategy, Test selection, Equipment selection, Independent testing, Interconnected testing, Constraint Programming

İÇİNDEKİLER

TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER	vi
KISALTMALAR	x
TABLolar	xii
ŞEKİLLER	xiii
GİRİŞ	1

1. BÖLÜM

TEST TASARIMI, TEST STRATEJİSİ, TEST OPTİMİZASYONU VE KULLANILAN TEKNİKLER ÜZERİNE

1. GİRİŞ	9
1.1.ÇALIŞMANIN ÖNEMİ VE GEREKLİLİĞİ	11
1.2.ÇALIŞMADA KULLANILAN SAYISAL TEKNİKLER.....	18
1.2.1.Doğrusal Programlama	19
1.2.2.Karışık Tam Sayı Programlama	20
1.2.3.Kısıt Programlama	21
1.3.ÇALIŞMANIN PLANI	22

2. BÖLÜM

LİTERATÜR ÖZETİ VE DEĞERLENDİRMESİ

2. GİRİŞ	24
2.1.HATA TEMELLİ ÇALIŞMALAR	25
2.1.1.Hata Saptanması	25
2.1.2.Hata Kapsanması	27
2.2.TEST VE TAMİR İÇERİKLİ ÇALIŞMALAR	28

2.2.1.Tamir İçerikli Çalışmalar	29
2.2.2.Test Süresi Temelli Test Planlaması ve Test Çizelgeleme.....	30
2.2.3.Test Planlaması ve Test Stratejisi Seçimi	35
2.2.4.Test Üretimi	40
2.2.5.Muayene – Deney Optimizasyonu	43
2.3.MALİYET TEMELLİ ÇALIŞMALAR	47
2.3.1.Maliyet Temelli Test Stratejisi Karşılaştırmaları	49
2.3.2.Maliyet Modelleri Üzerine Kurulu Test Stratejisi	50
2.3.3.Test Maliyetlerini Etkileyen Faktörler	54
2.3.4.Hata Faktörü İçeren Çalışmalar	59
2.4.EKİPMAN OPTİMİZASYONU	61
2.5.DİĞER KONULAR	62

3. BÖLÜM

TEK ÖZELLİKLİ ÜRÜNLERİN TESTİ

3. GİRİŞ	64
3.1.TEMEL TEST KAVRAMLARI	64
3.1.1.Test Maliyetleri	65
3.1.1.1.Sabit Maliyetler	65
3.1.1.2.Değişken Maliyetler	66
3.1.1.3.Performans Maliyetleri	66
3.1.2.Temel Kavramlar	67
3.1.2.1.Grafik Gösterim Ve Temel Test Akışı	68
3.1.2.2.Girdilerin Özellikleri	69
3.1.2.3.Makine Performans Karakteristikleri	70
3.1.3.Temel Test Akışı Hesaplanması	72
3.2.TEST TASARIMINDA TAMSAYI PROGRAMLAMA KULLANIMI	76
3.2.1.Temel Optimizasyon İçin Karışık Tamsayı Programlama Problemi	77
3.2.2.Çok Ölçütlü Temel Optimizasyon İçin Karışık Tamsayı	

Programlama Problemi	81
3.2.3. Karışık Tamsayı Programlama Yaklaşımı İle Modellemeden	
Kaynaklanan Hata	85
3.3.TEST TASARIMI İÇİN KISIT PROGRAMLAMA KULLANIMI	86
3.3.1.Kısıt Programlama İle Modelleme	88
3.3.2.Örnek Problemler	92

4. BÖLÜM

ÇOK ÖZELLİKLİ ÜRÜNLERİN TESTİ

4.GİRİŞ	100
4.1.ÇOK ÖZELLİKLİ TEST TEMEL KAVRAMLARI	101
4.1.1.Test Kapsaması	102
4.1.2. Ürün Oranı	105
4.1.3.Durum Vektörü Ve Kitle Dağılımı	107
4.2.ÇOK ÖZELLİKLİ BAĞIMSIZ TEST	109
4.3.ÇOK ÖZELLİKLİ ETKİLEŞİMLİ TEST	115
4.4.ETKİLEŞİMLİ TEST İÇİN DOĞRUSAL TAHMİNLEME	128

5. BÖLÜM

DOĞRUSAL TAHMİNLEME TEMELLİ KISIT PROGRAMLAMA

5. GİRİŞ	146
5.1.GENİŞ KAPSAMLI TEST TASARIMI OPTİMİZASYONU	146
5.1.1.Optimizasyon Modeline Dair Hesaplamalar	147
5.1.2.Geniş Kapsamlı Modelin Çözümüne Dair Sorunlar	152
5.2.TAM SENARYO YAKLAŞIMI	154
5.2.1.Tam Senaryo Yaklaşımı ve Karar Değişkenleri	155
5.2.1.1.Problem Girdileri ve İlk Safha Senaryoları	156
5.2.1.2.İkinci Safha Senaryoları	157

5.2.2.Tam Senaryo Yaklaşımı Modeli ve Çözümlemesi	160
5.2.2.1.Amaç Fonksiyonu	160
5.2.2.2.Model Kısıtları	163
5.2.2.3.Sayısal Çözüm ve Örnek Problem	167
 SONUÇ	173
KAYNAKÇA	183
EK 1	206
EK 2	209
ÖZGEÇMİŞ	210

SİMGELER VE KISALTMALAR

Kısaltmalar

AC:	Kabule Dayalı Örneklem (Acceptance Sampling)
AETG:	Otomatik etkin test üreticisi (Automatic Efficient Test Generator)
ASIC:	Uygulama özel entegre çip (Application Specific Integrated Chip)
ATE:	Otomatik Test Ekipmanı (Automatic Test Equipment)
ATPG:	Otomatik test üretimi (Automatic Test Pattern Generation)
BIST:	İçinde-yerleşik-kendi-kendine-test tekniği (Built – In – Self – Test)
CAD:	Bilgisayar destekli tasarım (Computer Aided Design)
CFR:	Sabit hata oranı (Constant Failure Rate)
CLP:	Kısıt Mantık Programlama (Constraint Logic Programming)
CP:	Kısıt Programlama (Constraint Programming)
DES:	Kesikli durum simülasyonu (Discrete Event Simulation)
DFR:	Azalan hata oranı (Decreasing Failure Rate)
DFT:	Test için Tasarım (Design For Test / Testability)
DPMO:	Milyon birim başına hata sayısı (Defect per Million Opportunity)
EA:	Evrimsel algoritmalar (Evolutionary Algorithms)
FRR :	Alandaki hatalı ürün oranı (Field Reject Ratio)
GA:	Genetik Algoritmalar (Genetic Algorithms)
IC:	Entegre çipler (Integrated Chip)
IFR:	Artan hata oranı (Increasing Failure Rate)
IP:	Tam Sayı Programlama (Integer Programming)
LP:	Doğrusal Programlama (Linear Programming)
LSI:	Büyük Ölçekli Entegrasyon (Large Scale Integration)
MIP:	Karışık Tam Sayı Programlama (Mixed Integer Programming)
MILP:	Karışık Tam Sayı Doğrusal Programlama (Mixed Integer Linear Programming)
QC:	Kalite Kontrol (Quality Control)
RCGA:	Gerçek kodlu genetik algoritmalar (Real Coded Genetic Algorithms)

SA:	Tavlama Simülasyonu (<i>Simulated Annealing</i>)
SoC:	Çip üstü sistemler (<i>System – on – Chip</i>)
SPC:	İstatistiksel Süreç Kontrolü (<i>Statistical Process Control</i>)
TAM:	Test Erişim Mekanizması (<i>Test Access Mechanism</i>)
TCA:	Test maliyet analizi (<i>Test Cost Analysis</i>)
TCM:	Teknik maliyet modelleme (<i>Technical Cost Modeling</i>)
VLSI:	Çok Büyük Ölçekli Entegrasyon (<i>Very Large Scale Integration</i>)

Simgeler

I:	Girdi (<i>Input</i>)
O:	Çıktı (<i>Output</i>)
T:	Test / Test Makinesi
P:	Testi geçme durumu (<i>Pass</i>)
F:	Test geçememe durumu (<i>Fail</i>)
R:	Tamir (<i>Repair</i>)
G:	Hatasız ürün (<i>Good</i>)
B:	Hatalı ürün (<i>Bad</i>)
f:	Teste tabi özellik (<i>Feature</i>)
TP:	G'nin P durumu (<i>True Pass</i>)
TF:	B'nin F durumu (<i>True Fail</i>)
FP:	B'nin P durumu (<i>False Pass</i>)
FF:	G'nin F durumu (<i>False Fail</i>)
GG:	G'nin R'den G olarak çıkması
GB:	G'nin R'den B olarak çıkması
BG:	B'nin R'den G olarak çıkması
BB:	B'nin R'den B olarak çıkması
M:	Yeterince büyük sayı (<i>Big M</i>)
y :	Ürün oranı (<i>Yield</i>)

TABLOLAR

Tablo 1. Değişik Test Stratejilerinin Özellikleri	37
Tablo 2. Temel Test Optimizasyon Problemi Girdileri ve Sonuçları	80
Tablo 3. Yeni Optimizasyon Problemi Sonuçları	83
Tablo 4. Alternatif Durum Problemi Çözümü	85
Tablo 5. MIP – CP Karşılaştırmalı Çözümler 1	93
Tablo 6. MIP – CP Karşılaştırmalı Çözümler 2	93
Tablo 7. Test Makinelerinin Kapasite Kullanım Oranları (Problem 1)	97
Tablo 8. Test Makinelerinin Kapasite Kullanım Oranları (Problem 2)	98
Tablo 9. Çok Değişkenli Bağımsız Test Modeli Girdileri	111
Tablo 10. Çok Aşamalı Çözüm	112
Tablo 11. Kısıtlı Çözüm	113
Tablo 12. Modifiye Çözüm	114
Tablo 13. Birikmeli Sınırlı Kusurlu Özellik Olasılıkları	125
Tablo 14. Bağımsız Test Sonucu Olasılıkları	132
Tablo 15. Örnek Senaryolar	133
Tablo 16. Senaryo Dağılımları	134
Tablo 17. FPS Kategorisi Senaryoları	138
Tablo 18. Kategori Temelli Kusursuz Ürünler Kitlesi	138
Tablo 19. Kategori Temelli Kusurlu Ürünler Kitlesi	139
Tablo 20. Hata Oranları	144
Tablo 21. Test Makineleri Özellikleri	156
Tablo 22. İlk Test Safhası Sonunda Durum Vektörleri	157
Tablo 23. İkinci Safha Test Girdi Durum Vektörleri Örnekleri	158
Tablo 24. İkinci Safha Test Sonrası Durum Vektörleri Örnekleri	159
Tablo 25. Birim Maliyet Girdileri	162
Tablo 26. Amaç Fonksiyonu Maliyet Katsayıları	162
Tablo 27. Tam Senaryo Optimizasyon Modeli Çözüm Bilgileri	169
Tablo 28. Tam Senaryo Optimizasyon Modeli Sonuçları	170

ŞEKİLLER

Şekil 1. Test Maliyetleri Şeması	56
Şekil 2. Temel Test Akışı	68
Şekil 3. Temel Test Akışı Karar Değişkenleri	73
Şekil 4. Temel CP Problemi Simülasyonla Kesin Çözüm	96
Şekil 5. Çok Özellikli Bağımsız Test Akışı	102
Şekil 6. Temel Test Akış Şeması	108
Şekil 7. İki Özellikli Seri Test – Grafik Çözüm	115
Şekil 8. A ve B Ürünlerinin C Ürünü Oluşturacak Şekilde Birleşmesi	126
Şekil 9. Doğrusal Yaklaşım	127
Şekil 10. Örnek Problem Akışı	132
Şekil 11. Sonlu Durum Otomatonu	136
Şekil 12. Problem Çözümü	139
Şekil 13. Geniş Kapsamlı Test Optimizasyonu Problemi Ağ Şeması	148

GİRİŞ

“Hikayemiz yaklaşık iki milyon yıl önce başlar. Og isimli bir mağara adamı, iki kayayı birbirlerine gerçekten sertçe vurarak ilkel aletler yapacağı taşları seçmek için ideal taşı belirleyebileceğini fark eder. Kırılmayan taş daha iyi bir alet haline gelecektir. Ve böylece test yapma çağı başlamıştır.

Bu inanılmaz buluşa rağmen test bilimi sonraki iki milyon yıl boyunca çok fazla yeniliğe uğramadı. Yine de nihayetinde teknolojimiz bilgisayarlardan test sürecinde faydalananacak düzeye ulaştı.”

BEDSOLE, J., R. RAINA, A. CROUCH ve M. S. ABADIR. “Very Low Cost Testers: Opportunities and Challenges.” *IEEE Design & Test of Computers*, International Test Conference, September – October 2001: 60.

Gerçekten de yüzyılın dönümünden bu yana tüm teknoloji alanlarında yaşanan gelişme insanlık tarihinin tümünde gözlenen gelişmenin toplamından çok ötede bir düzeye sahne olmaktadır. Bir zamanlar çok ilkel halleri hayal edilebilen cihazlar artık günlük hayatımızın sıradan bir parçasıdır. Leonardo da Vinci’nin tasarladığı, insanların uçmasını sağlayacak mekanizma hala hayata geçmemişse de her gün on binlerce insan binlerce uçakla seyahat etmektedir. Otomobil, telefon, hareketli görüntüler, sesli görüntüler ve nice ev eşyası gibi birkaç yüzyıl önce hayali dahi kurulamayan, yakın geçmişe kadar sayılı zengin insanda bulunan ürün artık sıradan gereksinimler haline gelmiştir. Cep telefonlarının fotoğraf makinesi, video kamerası, MP3 çalıcısı ve oyun oynama imkanları sunduğu düşünülürse, yirmi sene öncesinden günümüze ne kadar inanılmaz bir yol kat edildiği ortadadır. Çok değil 30 sene önce kartlar işlenerek sadece basit analizleri yapabilen oda büyüklüğünde bilgisayarlar bugün neredeyse her eve, okula, işyerine ve resmi kuruluşa girmiş, bilgi paylaşımını hayal edilemez noktalara getiren internetin de yardımı ile hayatımız hayalini bile edemeyeceğimiz bir şekle bürünmüştür.

Bu noktalara gelinirken bilim ve endüstrinin büyük gelişmeler gösterdiği de aşikardır. Ne yazık ki birçok yeniliğin, buluşun ve teknoloji gelişiminin sebebi

kan ve savařlar olmuřtur. Gerçek anlamda otomasyon, mekanik ve iletiřim alanındaki atılımlar savař metodlarının deęiřmesinden kaynaklanmıřtır. B y k meydanlarda orduların g ę s g ę se savařması,  nce tank, u ak, silah, gemi, denizaltı ve bomba ile savařa d n řm ř, b y k g   olabilmenin yolu, bu teknolojileri  nce geliřtirip y ksek miktarlarda  retebilmekten ge er olmuřtur. Telsiz iletiřim, radar, rota belirleme, uydu ve benzeri teknolojiler hep savařta  st n gelme amacıyla geliřtirilmiřtir. G n m zde ise  st n olmanın yolu ticaret ve  retim alanında  nde gelmek, bilgiye sahip olmak ve ekonomiye h kmetmeye varmıřtır. Talep  retebilmek, bireysel t ketimi artırmak ve bunları yaparken maliyetleri en aza  ekecek stratejileri saptamak var olmanın temel kuralı haline gelmiřtir. D nyanın k  k b y k t m  lkeleri bu d ng  i inde yerlerini almaktayken, gerek ulusal gerek uluslararası her piyasada firmalar  etin bir rekabet ortamında m cadele halindedir ve bařarı ancak zorlu řartlarda yařayabilmekle gelmektedir.

Elektronik  r nler en basit mutfak eřyasından en pop ler  r nler olan cep telefonları, bilgisayarlar, otomobiller ve benzeri  r nlerin  alıřmasını saęlar hale gelmiř ve gıda gibi temel bir ihtiyacın  retim ortamında bile elektronik m hendislięi  r n  makineler kullanılır hale gelmiřtir. Artık neredeyse elektronik bir cihazın rol almadıęı bir  retim s reci bulunmamaktadır denilebilir.

 zellikle ikinci d nya savařı sonrası Japonya'nın teknolojide yaptıęı inanılmaz atılım ile  r nlerin  eřitlilięi, fonksiyonları ve  alıřma hızları tahmin edilemeyecek d zeyde iyileřmiř ve yaygın bir tabirle "her řeyin k  ę "  retilir hale gelmiřtir. Oda b y kl ę ndeki bilgisayarlar  nce masa b y kl ę ne, sonra bir kasa ve monit r yeri kapsayacak k  kl ę , sonra diz  st  bilgisayarlara d n řm ř ve nihayet cep telefonu boyutuna inmiřtir. ř phesiz b ylesi bir deęiřim  ok daha hassas  retim s re leri,  ok daha ince ve seri iř ilik,  ok daha karmařık bileřenler ve dolayısı ile potansiyel y ksek maliyetler oluřmasını ka ınılmaz hale getirmektedir.

Bir yandan gelişmekte olan tüketici bilinci ve tüketici hakları, her türlü endüstriyi daha güvenli, daha az zararlı ve daha uzun ömürlü ürünler gerçekleştirmeye itmektedir. Dünyanın küreselleşmesi ile tüm dünyanın büyük tek bir pazar haline gelmesi ise firmalar üzerinde oluşan fiyat rekabeti baskısını artırmış ve firmalar hem üstün kaliteli, hem uzun ömürlü, hem güvenilir, hem de ucuz ürünler üretmeye zorunlu bir hale gelmiştir. Kalite radikal bir faktör olarak görülerek gerek sistem, gerek teknoloji bazında ciddi bir gerekliliğe dönüşmüştür. ISO 9000, MRP, MRP II, ERP, SAP ve Toplam Kalite sistemleri gibi bir dönemin lüks sayılabilecek uygulamaları artık standart ve olmazsa olmaz sistemlere dönüşmüştür. Ürünler için dünyanın her tarafında kullanılabilecek kalite standartları, teste tabi ürün özellikleri, minimum test sonucu değerleri norm belirleyici kuruluşlar tarafından belirlenmeye başlanmıştır. Uluslar arası ticarete bu normlara uymak, uyulduğunu belgelemek artık standart prosedürlere dönüşmüş ve örneğin CE işareti gibi mekanizmalarla sıkı şekilde kontrol edilir hale gelmiştir.

Bu süreç içinde elektronik kartlar, paneller, ana kartlar, devreler, çipler gibi ürünler de inanılmaz bir değişim geçirmiş ve kullanım alanlarının son derece hayati olması nedeniyle yüksek kaliteye sahip olmaları engellenemez bir gereksinim haline gelmiştir. Bu ürünler örneğin uçaklarda, uydularda, sağlık sektöründe, iletişimde ve işyerlerinin artık kalbi haline gelmiş bilgisayar ve makinelerde kullanılmaları nedeniyle, yapacakları olası hatalar, hataların zamanları, ürün ömrü, değişen teknoloji ile uyumlu olacak esneklik gösterebilmeleri, piyasalarını belirlemekte olduğundan, bu ürünlerin kalitelerini taahhüt etmek ciddi maliyetlere yol açar hale gelmiştir.

Bu çalışma üretim alanında faaliyet gösteren firmaların rekabet gücünü artırma amacıyla geliştirilmiş test temelli yaklaşımlara odaklanmaktadır. Çalışmanın mercek altına aldığı rekabet kavramını açmakta fayda vardır. Örgüt yapısı, örgüt iklimi, vizyon, felsefe, hatta siyasi duruş bir firmanın piyasadaki ömrü ve başarısı açısından önemli olabilir. Dağıtım ağı, reklam, tanınırlık ve benzeri pazarlama odaklı stratejileri de rekabet açısından önemlidir. Fakat bu

çalışmanın kapsamında irdelenecek rekabet tamamen ürün kalitesi, ürün başarımı ve maliyetlerin minimize edilmesi odaklıdır.

Açıkça görülmektedir ki kalitenin bir maliyeti vardır. Zaman zaman kalite kontrole yönelik yatırımlar bir firmanın üretim parkı için yapacağı yatırımın yanında önemsiz sayılamayacak maliyetlere ulaşabilmektedir. Testlerin gerçekleştirilmesi ise sektörden sektöre farklılık göstermekle birlikte belli alanlarda üretim maliyetlerine yaklaşabilmektedir. Özellikle elektronik cihazlar, çipler, elektronik kart ve paneller, örneklem alınmadan her parçanın deneye tabi tutulduğu, birçok mekanik bileşenin ve birleşmiş parçaların testlerinin detaylı yapıldığı ürünlere örnek teşkil etmelidir. Aksi mümkün olmakla birlikte birçok firma için oluşacak risk göze alınamayacak boyutlardadır.

Kalite maliyeti gibi kalitesizliğin de bir maliyeti vardır. Kalitesizlikten kasıt, ürünün tedarik zincirinin herhangi bir noktasında başarısızlık göstermesidir. Temel olarak ürün ıskarta ise hammadde ya da bileşenler ziyan olacaktır. Ürünün geldiği noktaya kadar harcanılan emek, işçilik ve makine zamanı, üretim, montaj ve test maliyeti, makine yıpranması, elektrik sarfiyatı kayıp maliyetlerdir ve bu maliyetler tekrar yerine konulamaz. Ürün satılmış ve alana gitmiş ve uygunsuzluk bundan sonra fark edilmiş ise, lojistik, montaj maliyetleri, ürünün hem alana gidişi hem de iadesi esnasında firmaya maliyet olarak dönecektir. Bunun dışında firma itibarının zarar görmesi, olası müşteri kayıpları ve olası tazminatlar da yine ek kalitesizlik maliyetleri olarak gözlenebilecektir.

Tüm bunların ışığında kalitenin sağlanması hem rekabet ve güvenilirlik açısından son derece önemli bir faktör, hem de ciddi bir maliyet kalemi haline gelmektedir. Buradan hareketle yüksek miktarda kalite testi yapan ve bu faaliyet için yüksek maliyetlere katlanan firmaların kalite testlerini gerçekleştirirken bir stratejileri olması gerekliliği açıkça görülmektedir.

Böyle bir strateji geliştirilirken firmanın dikkate alması gereken birçok faktör vardır. Örneğin test sırasında ürünün zarar gördüğü durumlar bulunabilir.

Ampullerin ömrü, otomobillerin “*crash*” testleri, birçok plastik üründe yapılan kopma mukavemeti deneyleri ürünün parçalanmasına ya da kullanılamayacak hale gelmesine neden olmaktadır. Buna benzer her ürünün test edilmediği durumlarda firma teste girecek parça adedini, bu tercihin oluşturacağı güvenilirlik riskini hesaplayarak optimal denek sayısını belirlemelidir. Bu amaçla geliştirilmiş istatistiksel yöntemler ve formüller bulunmaktadır. Bu durum risk ve maliyet dengesine örnek olacak çok basit bir örnektir. Daha az denek, daha düşük maliyete neden olacak fakat aynı zamanda da ürün kalitesi düzeyindeki belirsizliği artıracak, başka bir deyişle firma açısından kalitesizlik riskinin artmasına yol açacaktır. Firma her bir ürünü test edemeyeceğine göre, mutlaka bir optimal nokta bulunmaktadır ve bu da firmanın risk – maliyet tercihinine bağlıdır. Bu ifade zorunluluklar dışında geçerlidir. Örneğin birçok ürün standardında minimum test edilmesi gereken ürün adedi belirlidir ve firma için bir referans oluşturmaktadır, fazlasını yapmak firmanın inisiyatifine kalmıştır.

Bunun yanı sıra firma ürünü etkileyen belirli deneylerin bazılarını yapıp bazılarını yapmamak yoluna gidebilir. Her ne kadar bazı deneyleri atlaması belirli şartlar altında imkansız ise de, yine birçok ürün standardında rastlanılabileceği gibi, ürünün kalite kontrolünde hem periyodik testler hem de periyodik olmayan ya da zorunlu olmayan testler bulunabilmektedir. Bazı testler sadece belgelendirme esnasında bir kereye mahsus yapılmakta, bazı testler ise üretim şartları (makine, hammadde, prosedür, sistem, vb) değişmediği sürece tekrarlanmamaktadır. Firmanın herhangi bir resmi yükümlülüğü olmadığı varsayılır ise firma istediği testi yapıp, istediğini hariç tutabilir. Örneğin sadece iç piyasaya çalışan ve herhangi bir kalite belgesi bulunmayan ve almayı da amaçlamayan bir firma, sadece ürünün piyasada belli bir talep oranını karşılamasını sağlayacak bir kaç hayati deneyi yapma yoluna gidebilir. Bu durum ise yine bir stratejik karar noktasıdır. Hangi testlerin yapılacağı, hangilerinin yapılmaması durumunda üründe oluşacak potansiyel risk ve testlerin hangi sırayla yapılmasının en ekonomik olacağı soruları hep önemli sorulardır. Her tercihin getirileri ve götürüleri bulunmaktadır ve firma yine en risksiz ve en hesaplı noktayı bulmaya çalışacaktır.

Bu çalışma elektronik ürünlerini örnek kabul ederek ve elektronik ve telekomünikasyon ürünleri ile ilgili test literatürünü baz alarak, daha önce incelenmemiş bir konuyu araştırmayı amaçlamaktadır. Belirli bir kalite düzeyinin altına inilmeksizin, hangi test makinelerinin, hangi tamir makinelerinin seçilmesi, hangi testlerin devre dışı bırakılabileceği, hangi testlerin hangi makinelerde yapılabileceği, standart bir test – tamir döngüsünde test akışının nasıl gerçekleşeceği gibi sorulara yanıt aramak bu çalışmanın kapsamını oluşturmaktadır. En önemli faktör ise test stratejisi ya da test tasarımı olarak tanımlanabilecek bu sorular bütünüünün olabilecek en düşük maliyetle gerçekleştirilmesidir. Zira problemin finansal boyutu göz ardı edilecek olur, yani kaynaklara sınır konulmaz ise, en iyi strateji her bir ürünün test edilebilecek her bileşenini, tüm testleri en iyi uygulayacak makinelerle çok sefer test etmek şeklinde olacaktır. Optimizasyon yoluyla kalite – maliyet arasında bir denge noktası bulmak, kaliteden ödün vermeden maliyetleri minimize edecek test tasarımını oluşturmak bu çalışmanın en temel amacıdır.

Bu noktada belirtmek gerekir ki bu çalışma ampirik bir çalışma değildir. Çalışmanın alana katkısının teorik prensip ve hesapları ortaya koymak ve optimizasyon modelleri geliştirerek test tasarım problemini bir deneme – yanılma ya da karşılaştırma temelli uygulamadan ziyade her alanda kullanılabilir optimal kararların verilmesine imkan tanıyan çözümlere kavuşturmak olduğu söylenebilir. Sonuç olarak çalışma teorik temelleri ve genellenebilir modelleri ortaya koymayı amaçlamaktadır. Bu nedenle veri toplanarak alan araştırması eklenmesi yerine, bilgisayar aracılığıyla üretilmiş sentetik verilerin kullanılması tercih edilmiştir. Bu durum geliştirilen modellerin gerçek hayat problemlerinin çözümünde kullanılmasına bir engel oluşturmamaktadır.

Bu kapsamda çalışmada öncelikle bir giriş bölümü oluşturulmuştur. Bu bölüm test tasarımı, test stratejisi belirleme ve test optimizasyonu problemlerinin önemini ve bu tip çalışmaların üzerinde durmanın gerekliliğini açıklamaya yöneliktir. Yine bu bölümde ayrıntılı olarak çalışmanın genel bir planı da

aktarılmaktadır. Daha sonra ayrıntılı bir literatür özeti sunulmakta ve tartışılmaktadır.

Üçüncü bölüm test tasarım probleminin temel kavramları konusunda bilgi sunmakta ve teste tabi tek özelliği olan ürünler için optimal test ve test ekipmanı kararlarının nasıl alınması gerektiğini ortaya koymaktadır. Burada sunulan tek özellikli teste dair optimizasyon modelleri doğrudan uygulanabilir modellerdir ve bu çalışmanın alana ilk katkıları olarak kabul edilebilir.

Dördüncü bölüm çok özellikli ürünlerin test edilmesini konu etmektedir. Bu bölümde öncelikle çok özellikli teste dair temel kavramlar ve hesaplar verilmekte, daha sonra bağımsız test durumu açıklanmaktadır. Bağımsız test için geliştirilen karışık tam sayı programlama ve kısıt programlama modelleri yine bu bölümde ortaya konmaktadır. Bu modeller belirli varsayımlar altında gerçek hayatta da aktif olarak kullanılabilecek modellerdir ve bu çalışmanın alana katkıları içinde gösterilebilirler. Dördüncü bölüm daha sonra etkileşimli test durumunu açıklamakta ve etkileşimli teste dair daha önceden ortaya konmuş kesin çözüm ve yaklaştırma yöntemlerini detaylı olarak tartışmaktadır.

Beşinci bölümde ise zorluğu ve karmaşıklığı yeri geldikçe gösterilecek etkileşimli ve çok özellikli test sürecinde karar aracı olarak kullanılacak optimizasyon modeli ayrıntılı olarak tanımlanmaktadır. Problemin genel bir model olarak ortaya nasıl konulacağı aktarılmakta ve böylesi bir modelin mevcut şartlar altında çözümsüzlüğü irdelenmektedir. Daha sonra tam senaryo yaklaşımı olarak ifade edilen alternatif bir çözüm yöntemi geliştirilerek bu zor problem için optimal sonuçlar veren bir model oluşturulmaktadır. Göz ardı edilebilir hata payları içeren ve bu noktada ancak belirli varsayımlarla kurulabilen bu modelin çözümleri, avantajları ve sınırlılıkları da yine bu bölümde tartışılmaktadır. Bu çözüm yöntemi özellikle kombinatoryal yapıda olup doğrusal olmayan kısıtlar içeren problemler için genel bir çözüm yöntemi olarak da kabul edilebileceğinden bu çalışmanın en önemli bilimsel katkılarından biridir. Yine de

esas amaç olan test tasarımı probleminin modellenmesi ve çözülmesi konusunda elde edilen başarılı sonuçlar bu çalışmayı önemli kılmaktadır.

1. BÖLÜM

TEST TASARIMI, TEST STRATEJİSİ, TEST OPTİMİZASYONU VE KULLANILAN TEKNİKLER ÜZERİNE

1. GİRİŞ

Test ile ilgili araştırmalar son 40 yıl içinde ivme kazanmıştır. Gluss (1959), Firstman ve Gluss (1960) ve Flehinger'in (1965) çalışmalarına bakıldığında bu çabanın aslında ikinci dünya savaşı dönemine denk geldiği görülmektedir. Birçok teknoloji gibi test teknolojisi de savaşlar nedeniyle büyük bir atılıma örnek oluşturmaktadır. Yine de bir dönemin acımasızlığı evrimleşmiş ve insanlığa fayda sağlayacak, hayatı kolaylaştıracak amaçlar uğruna kullanılmaya başlanmıştır.

Test ve test ile ilgili kavramlar ifadesinin açılması ilerleyen aşamalarda verilen bilgilerin netleşmesi için önemlidir. Bu çalışmanın kapsamı itibarıyla test terimi bir ürünün herhangi bir üretim süreci sonrasında uygulanacak, fakat tercihen üretim süreci sonunda gerçekleştirildiği öngörülen kalite kontrol testlerini ifade etmektedir. Ürün kalitesini sağlayacak testlerin bir ya da birden fazla olması söz konusu olabilir. Bu durum ürün üzerindeki teste tabi özelliklerin (*attributes, features*) sayısına bağlıdır. Bu özellikler ürünü oluşturan bileşenler olabilecekleri gibi ürünün özel bir amacı gerçekleştirmeye çalışan aktiviteleri de olabilir. Teste tabi tek özellik olması durumunda test, "Tek Özellikli Test" olarak ifade edilmektedir. Birden fazla özellik söz konusu ise ifade "Çok Özellikli Test" halini almaktadır. Çok özellikli test durumu ise iki alt başlıkta incelenmektedir. Bunların ilki her bir bileşen ya da özelliğin bağımsız olarak test edildiği ve test ya da tamir süreçlerinin birbirlerini etkilemediği "Bağımsız Test" durumudur.

İkincisi ise test ya da tamir sürecinde uygulanan işlemin sadece teste tabi bileşeni değil başka bileşenleri de etkilediği “Etkileşimli Test” durumudur.

Çok sayıda test söz konusu olduğu takdirde bir test süreci söz konusudur. Fakat bu çalışma kapsamı itibarıyla test sürecini, bir kalite kontrol süreci olarak tanımlamamaktadır. Bu bağlamda akla ilk gelen yaklaşım olan İstatistiksel Proses Kontrolü ve benzeri kalite kontrol süreçleri bu çalışma tarafından tanımlanmamakta ve araştırılmamaktadır. Bu çalışma en özet ifadesiyle alternatif testler ve alternatif ekipmanlar arasından uygun test ve test ekipmanını seçerek belirli bir kalite düzeyini sağlayacak en hesaplı test stratejisini saptamayı amaçlamaktadır. Bunun yanı sıra test akışı ve test süreci ile ilgili matematiksel hesapları ve modelleri sunmayı amaçlamaktadır.

Böylece test ile ilgili bazı temel kavramlar ortaya çıkmaktadır. Bunlar öncelikle testin uygulanışını tanımlayan tek ya da çok değişkenli test ile bağımsız ve etkileşimli test kavramlarıdır. İkinci etapta ise test, test ve tamir ekipmanı, bu kavramların ekonomik ve verimli bir şekilde seçilmesi, yani optimizasyonu ve ortaya çıkan sonucu ifade eden test stratejisi kavramları göze çarpmaktadır.

İkinci ve takip eden bölümlerde bu kavramlar netleşecek ve yeni kavramlar bunlara eklenecektir. Örnek olarak, test stratejisinin teste tabi özellikleri hangi oranda içerdiğini gösteren hata kapsanması oranı (*fault coverage*), kaliteli olduğu testler sonucu belirlenen ve bu teşhiste hata payı bulunmayan çıktılarının oranını ifade eden ürün oranı (*yield*) (ki kaliteli ürün oranı anlamında kullanılmaktadır), kaliteli olarak saptanan fakat hata içeren ürünlerin oranını gösteren alandaki ret oranı (*FRR, Field Reject Ratio, Field Return Ratio*), test üretimi, hata yeri saptanması ve makine performans karakteristikleri gibi kavramlar sayılabilir. Bu ifadeler ilerleyen bölümlerde detaylı şekilde inceleneceğinden burada ayrıntılarına girilmeyecektir.

Bu bölümün amacı, çalışmanın ele aldığı problemin ve alanın neden önem taşıdığını ve kullanılacak teknikleri ve çalışmanın planını aktarmaktır.

Çalışmanın önemi ile ilgili alt bölüm kaçınılmaz olarak kendi içinde küçük bir literatür taraması gibi görünecek olsa da alan ile ilgili genel bir literatür taraması kısmı bulunmaktadır. Çözüm yöntemleri bu bölümde irdelenecek ve uygulamaları bulunduran bölümlerde kullanım nedenleri ve elverdiği imkanlar ile sınırlılıkları ayrıntılı olarak incelenecektir. Benzer alanlardaki problemlerde, çözüm teknikleri açısından nasıl yaklaşımlar sergilendiği ise literatür taraması kısmında detaylarıyla incelenecektir. Çalışmanın planı kısmı ise takip eden bölümlerin içeriklerini aktarmayı ve o bölümlerde hangi konuların işleneceğini genel hatlarıyla tanımlamayı amaçlamaktadır. Böylece konu ve konunun önemi ve ele alınışı ile ilgili genel bir fikir oluşturulacak ve çalışma içeriğinin anlaşılması kolaylaştırılmış olacaktır.

1.1. ÇALIŞMANIN ÖNEMİ VE GEREKLİLİĞİ

Test alanında yapılan çalışmalar gerek nedenleri, gerek uygulama yöntemleri ve gerekse sonuçları açısından diğer akademik araştırma alanları gibi akademik ve endüstri için önem taşımaktadır. Bu alanda 60'lı yıllardan bu yana çok sayıda çalışma ortaya konmuş, zaman içinde her yıl tekrarlanan onlarca üst düzey konferans ve sempozyumlar düzenlenir hale gelmiştir. Kalite ve ürün ömrü gibi artık standart olarak aranan şartlar bir yana, özellikle elektrik – elektronik alanında üretilen ürünlerin hem çok pahalı olması, hem de son derece hayati kullanım alanları bulunması ve ek olarak gelişen teknoloji ile birlikte dünyanın en ücra köşelerinde bile kullanım ihtiyacı doğması, olası hataların büyük sorunlara ve maliyetlere yol açmasına neden olmaktadır.

Buradan hareketle son 30 yıl içinde öncelikle ürün kalite düzeyi ve ürün ömrü gibi kavramlar piyasada rekabetin vazgeçilmez koşulları haline gelmiş ve bu da endüstrinin ve işletmelerin bu kavramlar açısından kendilerini güvenilir kılmak ve yaşayabilmek için ayrıntılı çalışmalara girmelerine neden olmuştur. Williams da (1984) mantık testlerinin 1959'da hata modellerinin kullanımı ile başladığını belirtmektedir. Zarar verici test (*Destructive Testing*) söz konusu olmadığı

takdirde yapılabilecek testlerin ve tekrar sayılarının minimum sayıları genellikle standartlar ve şartnamelerle belirlenmiş olup, buradaki şartlar çoğu kez piyasayı tatmin edemeyebilecek sonuçlar verebilmektedir. Testlerin her türlüünü çok kere tekrar etmenin ise üst sınırı olmayabilir. Firmalar her türlü testi onlarca hatta yüzlerce kere uygulayarak mükemmel ürünlere ve %100'lük bir kalite oranına ulaşabilirler. Fakat bunun da en ekonomik yöntem olmadığı ortadadır.

Böylece firmalar için optimizasyon kaçınılmaz hale gelmektedir. Hangi testlerin, hangi makinelerde (ve hatta kaç kez) tekrarlanması gerektiği sorusunun cevabı optimizasyonda yatmaktadır. Yine de o noktaya gelinmeden önce bu problemin hangi yollardan geçerek bu hale geldiği ve hangi nedenlerle önem kazandığı incelenmelidir.

Literatür incelendiğinde 70li yılların ortasında test probleminin hala yeterince tanımlı olmadığı görülmektedir. Finch'in belirttiği üzere (1975), o dönemde LSI (*Large Scale Integration*) için testler hem endüstri hem de kullanıcılar için bilinmez haldeydi ve testlerin nasıl yapılması gerektiği üzerine bir fikir birliği söz konusu değildi ve makul bir maliyetle gerçekleştirilebileceği konusu belirsiz idi. Sistem simülasyonu yapan küçük cihazlardan kapsamlı genel amaçlı test ekipmanlarına kadar yüz binlerce dolarlık cihazlar bulunmaktaydı ve maliyet ile kalite arasındaki denge yeni sorgulanmaya başlamıştı. Bazı hataların asla saptanamayacağı düşünülüyordu (Agrawal ve diğerleri, 1982).

Ürünler giderek karmaşılaşmaktaydı ve artık testin önemi iyice belirginleşmekteydi. Stapper ve diğerleri (1983), üretim sürecinin karmaşılaştığını, entegre devrelerin binlerce bileşeninin çiplerde yapacakları işlere göre gruplandığını, transistörler gibi yüz binlerce bileşenli çiplerin üretildiğini ve tüm bunların doğru çalışmasının üretim esnasında yapılacak kontrollere bağlı olduğunu belirtmektedirler. Aynı fikir Adams ve Hochbaum (1997) tarafından da desteklenmektedir. Adams ve Hochbaum (1997) aradan geçen 14 seneye rağmen test sürecinin hala zaman alıcı ve pahalı bir süreç

olduđuna ve kk dzeltmelerin bile iřletme karını nemli lde etkilediđine iřaret etmektedirler.

Maly ve diđerleri (1986) istatistik ve optimizasyon tekniklerinin sadece birkaç senedir kullanılmaya bařlandđını belirtmekte ve bunun nedeni olarak rnn giderek karmařıklařmasını gstermektedirler. Onlara gre, entegre devre ve VLSI (*Very Large Scale Integration*) iplerinin karmařıklđđı artıp boyutları klmekte ve firmanın karını belirleyecek olan rn oranı bu nedenle dřmekte ve anılan tekniklerin alana girmesini kaınılmaz kılmaktadır.

Bylece rnn karmařıklařması ve boyutlarının klmesi ciddi bir maliyet đesi olarak ortaya çıkmaktadır. Bu fikir sonraları olduka kabul grmř ve test alanındaki hemen her alıřmada vurgulanmıřtır. rnek olarak, Dislis ve diđerleri (1988 ve 1993b), maliyetin inanılmaz artıřının yanı sıra, karmařık dijital sistemlerde aranan kalite ve gvenilirliđin test srecini, rn mrndeki en nemli bileřenlerden biri haline getirdiđini belirtmektedir. Robinson (1989), aynı gerekeyle test planlamasının giderek zor ve pahalı bir iřlem haline geldiđini ve toplam mhendislik maliyetleri iinde dikkat ekecek bir orana sahip olduđunu vurgulamaktadır. Chevalier ve Wein (1997), rnn karmařıklařmasıyla muayene maliyetlerinin, toplam retim maliyetlerinin yarısına denk geldiđini ve test kaynaklarını optimize etmeden rekabet etmenin imkansız olduđuna dikkat ekmektedirler.

Probleme Abadir (1998) ve Biasizzo ve diđerleri (1998) de deđinmekte ve ikinci grup bu karmařıklđđın sistem performansı iin řart olduđunu ve fakat sistemlerin yrtlmesi ve tamirinin ok zorlařtıđını belirtmektedir. řużek ve diđerleri (2000), karmařık sistemlerde hata teřhisi iin harcanan zaman ve paranın hayati hale geldiđini ve olabildiđince dřk tutulması gerektiđini belirtmektedirler. Bedsole ve diđerleri (2001) test maliyetinin giderek arttıđını vurgularken, Maly (2001) tasarım ve test maliyeti – karmařıklđđı kavramlarının giderek bir kabusa dnřtđn sylemektedir. Karmařıklđđın elbette ki artması

gerektiğini, fakat transistör sayısı ve bileşenler arasındaki etkileşim gibi faktörlerin dikkate alınması gerektiğini de eklemektedir (Maly, 2001).

Emmons ve Rabinowitz (2002), Volkerink ve diğerleri (2003), Hashempour ve diğerleri (2003), Ali ve diğerleri (2004), ürün karmaşıklığının maliyetleri ciddi derecede artırdığını belirtmekte ve son iki grup milyonlarca bileşene sahip devrelerin üretiminin çok yakın olduğunu belirtmektedir. Yüz binlerce bileşen için bu denli kaygı söz konusu iken birkaç milyon bileşenli ürünlerin yolda olması, test maliyetlerinin çok daha fazla öneme sahip olacağının açık bir göstergesidir. Özellikle Ali ve diğerleri (2004), bir elektrik sisteminin tüm özelliklerinin 2cm^2 alanında silikona entegre edildiğini belirtirken, bunun test sürecini daha karmaşık hale getirdiğini ve sorunların giderek akutlaştığını önermektedirler. Yakın gelecekte bir transistörün test maliyetinin üretim maliyetini aşacağını söyleyen Ali ve diğerleri, test sürecine ekonomik ve güvenilir bir çözüm bulunmasının şart olduğunu savunmaktadırlar (2004)

Elbette problemin hassaslığını farklı bakış açıları ile tanımlamak da mümkündür. Hawkins ve diğerleri (1989), o dönem için hatasız entegre devrelerin garanti edilebilmesinin mümkün olmadığını ve devrelerin karmaşıklığı, hata anomalileri ve ekonomik nedenlerin bunu mümkün kılmadığını belirtmektedirler. Yüksek kalite ve ürün boyutlarının büyümesinin bunu yalnızca daha kötüye götüreceğini öngören Hawkins ve diğerleri (1989), çözümün tasarım yolundan geçtiğini belirtmekte ve “Test edemeyeceksen üretme” deyişini hatırlatmaktadırlar. Ayrıca ilgilenilen hata kapsamının fiziksel hataları içerdiğini belirtmektedirler. Dear ve diğerleri (1991) de tasarım konusuna odaklanmakta ve çok ilginç bir istatistik vermektedirler. Silikon üreticilerine göre çalıştığı kabul edilen entegre devrelerin yarısı hedef sisteme monte edildiklerinde arız çıkarmaktadırlar ve sektörde ideal hata kapsanma oranının %98’in üstü olarak tanımlanmasına rağmen tasarım kabulünde genellikle %40 - %70 oranında hata kapsanarak ürün tasarımlarının kesinleştirilmeleri söz konusudur (Dear ve diğerleri, 1991). Bu rakamlar gerek

firmaların tutumu, gerekse ürünlerin performansları açısından hayli düşündürücüdür.

Thatcher (1993), endüstride başarı için bir numaralı etkenin kalite olduğunu, müşterilerin en iyiyi beklediğini, istediğini ve en iyiyi bulana dek alternatif ürünleri değerlendireceklerini söylemektedir. Elektronik ürünleri sektörünün yüksek miktarlarda üretim yaptığını ve bu yüzden sabit maliyetlerin giderek önemsizleştiğini öne süren Thatcher, rekabete rağmen ürün fiyatlarının makul olması gerektiğini ve yüksek maliyetin kalite anlamına gelmediğini savunmaktadır (1991). Maxwell ve Aitken (1993) de önemli faktörün ürün kalitesinin giderek daha çok ön plana çıkması olduğuna katılmakta ve bu nedenle hataların olabildiğince erken saptanması ve üreticilerin önce ürünlerinin kalite düzeylerini bilir hale gelmesi, sonra da mevcut kalite düzeyine geliştirmek için çabalamaları gerektiğini söylemektedirler.

Gaitonde ve diğerleri ise (1995) Hawkins ve diğerleri (1989) gibi ürün boyutunun büyümesinin sorunlara yol açacağını düşünmektedir. Talep nedeniyle çip üreticileri tek bir çipe olabildiğince çok fonksiyonellik katmaya çalışmakta ve bu da ürün oranını etkilemektedir (1995). Diğer taraftan Heineken ve diğerleri (1996) entegre bir devrenin başarısının sadece teknoloji düzeyi, fonksiyonellik ya da üstün performansla belirlenemeyeceğini, bunlara üretim maliyetlerinin de eklenmesi gerektiğini savunmaktadırlar.

Yüksek kalite standartları ve milyonda 3.4 hatalı parça oranı gibi değerlerin şart koşulduğunu belirten Ding ve diğerleri (1998), üst düzey bir tesisin bir milyar dolara kurulabileceğini, ATE'lerin (otomatik test ekipmanlarının) her birinin birkaç milyon dolara alınabileceğini, haliyle ekipman seçimi ve ekipmandan faydalanma durumunun maliyetleri ciddi boyutlarda etkileyeceğini ortaya koymaktadırlar. Hatalı kararlar testi geçen bozuk ürünlerin ve hurdaya ayrılan ürünlerin maliyetlerinin de çok yüksek olduğunu belirten Ding ve diğerleri (1998), %100 kontrol, yani her bir ürünün kontrol edilmesini gerekli görmektedirler.

Barbagallo ve diğlerleri (1999), telekomünikasyon alanındaki ürünleri mercek altına almış ve çok hızlı değışen, sürekli yeni özelliklere sahip ürünlerin, bir de üzerine fiyat artırımı söz konusu olmadan satılması talebinin, firmalar üzerinde üretim döngüsünü hızlandırma baskısı oluşturduğunu saptamışlardır. Bu yüzden istenen performans düzeyi, dayanıklılık, bulunabilirlik ve güvenilirlik şartlarını sağlamak için üretim sürecinin her aşamasında sıkı kontrol ve muhtemel hataların olabildiğince erken saptanması şart kabul edilmelidir (Barbagallo ve diğlerleri, 1999). Bu açıdan fikirleri Maxwell ve Aitken (1993) ile uyumludur.

Žužek ve diğlerleri (2000) de yine Barbagallo ve diğlerleri (1999) gibi piyasada rekabet edebilme koşulunun, piyasanın güvenilirlik, bulunabilirlik ve bakım taleplerine cevap vermekten geçtiğini savunmaktadır. Firmalar bu amaçla test edilebilirliğe yönelik tasarım ve etkin test metotları uygulamalıdır. Bu da ancak minimum maliyetle maksimum hatayı saptamaya yönelik testlerin sıralanması gibi 40 seneye yakın bir süredir incelenen bir alan üzerinde daha etkili araştırma yapmakla elde edilecektir (Žužek ve diğlerleri, 2000).

Emmons ve Rabinowitz (2002), ürün karmaşılaşması ve kalite taleplerinin aslında farklı maliyet kaynakları olmadığını ve birbirlerini tetiklediklerini öne sürmektedirler. Buna göre, yüksek kalite ve düşük maliyet gereksinimleri ürünün karmaşılaşmasına ve çok daha fazla potansiyel hata kaynağı oluşturmaya neden olmaktadır. Bu da test miktarını ve otomatik test tercihinin artırmaktadır. Firmalar pahalı cihazlara yatırım yapmakta, muayene kapasitesi, dağıtım ve çizelgelemesi gibi yeni karar değışkenleri oluşmaktadır. Fakat tüm bunları eşzamanlı saptayacak bir çerçeve halen eksiktir (Emmons ve Rabinowitz, 2002). Sermaye perspektifinden maliyete yaklaşan bir diğler örnek çalışma da Chien ve Wu (2003) tarafından sunulmuştur.

Oppermann ve diğlerleri ise (2001, 2003a ve 2003b) esas amacın müşteri memnuniyeti olduğunu, bu nedenle hata oranının sıfıra çekilmesi ya da radikal

düzeyde düşürülmesi gerektiğini savunmakta ve çözüm olarak istatistiksel proses kontrol, örnekleme planları ve teknikleri ile optimizasyonun birleştirilmesini çözüm kabul etmektedirler.

Güncel bir çalışmada ise Goyal ve Mosher (2006), test geliştirme maliyetlerinin, ürün geliştirme maliyetlerinin %30'u kadar olduğunu vurgulamakta ve test sistemlerinin karmaşıklıklarına göre \$100,000 ile \$1,000,000 arasında değiştiğine dikkat çekmektedirler. Fiyat, maliyet ve hata payları konusunda acımasız kabul edilebilecek piyasa baskısı, bu şartlarda test ve ekipman maliyetlerinin kontrol altında tutulmasını zorunlu kılmaktadır. Öyleyse test, hata saptama, tamir tekrarları, test süresi, test ekipmanları, sınırlılıklar ve arz zinciri arasındaki ilişkiler iyi incelenmelidir (Goyal ve Mosher, 2006). Goyal ve Mosher ayrıca test sürecinin modellenmesinin oldukça karmaşık olduğunu belirtmektedirler (2006). Fisher ve diğerleri (2007a) ise, yüksek test ihtiyacının sadece özelliklerin artması ve ürünün karmaşılaşmasından değil, bileşenlerin taşeron firmalarca imal edilmesi ve değişik ürünlerde ortak kullanılabilir yazılımların kullanılması gibi nedenlerden de kaynaklandığını belirtmektedirler. Optimal bir durumda ise artan test maliyetleri firmaya geri dönmekte ve harcanan her dolar, alanda çıkabilecek hatalar cinsinden on dolar tasarruf oluşturmaktadır. Böylece test sürecinin ilgi çekici bir optimizasyon imkanı sağladığı açıkça ortadadır (Fisher ve diğerleri, 2007a). Goyal ve Mosher (2006) ile tutarlı şekilde Fisher ve diğerleri (2007a) de test maliyetleri ile nihai ürünün kalitesi arasındaki ilişkinin modellenmesinin oldukça zor olduğunu belirtmektedirler. Test stratejisi oluşturma probleminin zorluğuna Moore (1994) da dikkat çekmektedir.

Dear ve diğerleri (1991), negatif değer oluşturduğu için test maliyetlerinin testin getirdiği faydalardan daha çok dikkat çektiğini belirtmektedirler. Ayrıca firmanın kaliteye verdiği önemin piyasa tarafından algılanmasının önemine de dikkat çekmektedirler. Bu açılardan bakıldığında yeni test ekipmanı yatırımının yönetim düzeyinde karar verilmesi gereken ciddi bir maliyet kalemi olmasına

rağmen, bazen bu yola gitmek firmanın yüzleşeceği maliyetler açısından daha faydalı olacaktır (Dear ve diğerleri, 1991).

Bu bilgilerin ve bir sonraki bölümde incelenecek çalışmaların ışığında, test süreci, tasarımı, stratejisi ve optimizasyonu, halen firmalar açısından son derece hayati bir maliyet planlama alanıdır. Optimal test stratejileri ve test tasarımı elde edebilmek, haliyle ideal test kombinasyonlarını ve test ekipmanlarını seçebilmek, maliyetlerin aşağı çekilebilmesi ve belirlenmiş bir kalite düzeyinin elde edilmesi için hayati kararlar vermek anlamına gelmektedir. 1950lerin sonundan bu yana araştırmacıların dikkatini çekmiş bu alanda halen incelenmemiş bakir araştırma soruları bulunmaktadır. Bu çalışma, daha önce değişik şekillerde incelenmiş olsa da test akışının saptanması ve buradan gelen bilgilerden faydalanarak test stratejisi ve test tasarımı optimizasyonu problemini ortaya koyarak çözüm araması itibariyle orijinal, zor ve önemli bir problemi mercek altına almaktadır.

Aşağıdaki alt bölümde ise çalışmada kullanılan sayısal tekniklerle ilgili genel bilgiler verecektir.

1.2. ÇALIŞMADA KULLANILAN SAYISAL TEKNİKLER

Bu çalışmada test tasarımı ve optimizasyonu ile ilgili değişik kavramlar üzerinde durulmuş ve bunlarla ilgili matematiksel gösterimler, denklemler, eşitsizlikler ve optimizasyon modelleri kullanılmıştır. Tüm bu matematiksel ifadeler aynı amaçla kullanılmamıştır. İlerleyen bölümlerde verilen formüllerin bir kısmı tanımlama amacı taşımaktadır.

1.2.1. Doğrusal Programlama

Genellikle test akışı ve ilintili konularda doğrusal denklem kümeleri kullanılmaktadır ve gerektiği durumlarda, örneğin tek değişkenli test optimizasyonu söz konusu olduğunda, ilgili doğrusal denklem kümesi, problemin yapısına göre indekslenerek ve tanımlayıcı amaç fonksiyonu eklenerek bir doğrusal programlama modeli haline getirilerek optimize edilmekte kullanılmıştır. Tamamen doğrusal değişkenlerden oluşan basit problemler Doğrusal Programlama (LP, *Linear Programming*) kullanılarak çözülmüştür. Doğrusal programlama yöneylem alanında tamamen doğrusal kısıtlar ve sürekli değişkenler söz konusu olduğunda alternatifsiz bir modelleme ve çözüm yöntemidir. Bunun temel sebebi doğrusal modellemenin göreceli kolay olması ve Simplex Algoritması aracılığı ile modelin kolaylıkla çözülebilir olmasıdır.

Genel bir doğrusal programlama modeli, doğrusal olarak ifade edilebilecek bir alan (bu alan örneğin bir politop ya da konveks polihedron şeklinde tanımlanabilecektir) içinde $f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n + b$ şeklinde doğrusal olarak tanımlanabilecek bir fonksiyonun en yüksek ya da en düşük değerini bulmaya çalışmaktadır. Kanonik form olarak isimlendirilen şekliyle bir doğrusal programlama modeli şu yapıdadır:

Maksimize / Minimize et

$$\mathbf{c}^T \mathbf{x}$$

Kısıtlar

$$Ax \leq b$$

Ve

$$x \geq 0$$

Bu ifadede x , yani değişkenler vektörü ile katsayılar vektörü c 'nin transpozu çarpılarak amaç fonksiyonu elde edilmektedir. A matrisi kısıtların katsayılarını

ve b vektörü kısıtların limitlerini belirlemektedir. Hiçbir karar değişkeninin negatif olmayan değerler alması da yine metodun zorunluluğudur ve kısıt seti olarak modele eklenmiştir.

Doğrusal denklem sistemleri Fourier – Motzkin eleme metodu ya da Gaussgil eleme metodu gibi yöntemlerle çözülebilecek olsa da doğrusal programlama metodunu kesin ve en hızlı çözen Simplex Algoritmasının yaratıcısı, yöntemi 1947 yılında sunan George B. Dantzig'dir. Yöntem, o zamandan beri işletme, iktisat ve mühendislik alanlarında etkin olarak kullanılmaktadır. Metodun en yaygın kullanılan çözücüler Lindo, Lingo, Gams ve CPLEX olarak ortaya çıksa da doğrusal programlama problemlerini *Revised Simplex* metodu kullanarak çözebilecek kodlar yazmak da mümkündür.

1.2.2. Karışık Tamsayı Programlama

İlerleyen bölümlerde de görülebileceği gibi genel ifade ile test problemi içinde test makinesi ve makine sayısı gibi değişkenleri içermek durumundadır. Açıkça görülecektir ki bu değişkenler sürekli bir yapıya sahip değildir ve tamsayı değeri alması beklenen karar değişkenleridir. Bu yaklaşımla modellenen problemler Karışık Tamsayı Programlama (MIP, *Mixed Integer Programming*) ailesine dahil olacaklardır. Bu tip problemlerde hem tamsayı, hem de doğrusal karar değişkenleri bir arada bulunmaktadır. Özellikle taşıma, üretim planlama, çizelgeleme problemlerinin modellenmesinde yaygın kullanılan bu metodun çözülmesi doğrusal modeller kadar kolay değildir. Xpress MP ve CPLEX gibi çözücüler, *Branch and Bound* ve *Branch and Cut* başta olmak üzere değişik yöntemlerden yararlanarak çok karmaşık olmayan problemleri kolayca çözebilmektedir.

Yine de LP ve MIP ile modelleme yaklaşımının sınırlılıkları bulunmaktadır. Prensip olarak bu yaklaşımlarda doğrusal olmayan kısıtlar ve mantık kısıtları kolaylıkla kullanılamamaktadır. Bazı temel mantık kısıtları (örneğin iki alternatif

makineden sadece birinin seçilmesi vb) modellenebiliyor olsa da, kısıtlar karmaşıktıkça ve işin içine doğrusal olmayan bir yapı girince bu yaklaşımlarda ısrar etmek, gerçekte optimal olmayan çözümlere yol açacaktır. Arada oluşacak hata payı genellikle görmezden gelinecek bir düzeyde gerçekleşmeyeceğinden doğru modelleme yaklaşımının bulunması hayatidir.

1.2.3. Kısıt Programlama

Bu noktada belirtmek gerekir ki test problemini gerçek hayata yakın modellemek söz konusu olduğunda doğrusal olmayan kısıtlar ve mantık kısıtlarından kaçmanın bilinen bir yolu bulunmamaktadır. Fakat böylesi kısıtlarla modellemeye imkan veren yöntemler de bulunmaktadır. Bu çalışmada tercih edilecek yöntem Kısıt Programlama (CP, *Constraint Programming*) olarak saptanmıştır. Kısıt programlama seçimindeki temel neden, kısıt programlamanın Simplex Algoritması ve MIP problemlerinde kullanılan çözüm metotlarına yapısı itibariyle zaten açık olması ve bunun üstüne gerektiği takdirde doğrusal olmayan kısıtlar ya da mantık kısıtlarını çözebilecek yöntemler belirlenmiş olmasıdır. Zaten bu tekniğin matematiksel programlama ile yakından ilintili olduğu her iki disipline hakim araştırmacılar tarafından da bilinmektedir (Lustig ve Puget, 2001). Kısıt programlama köklerini Mantık Programlama'dan (CLP, *Constraint Logic Programming*) almış bir teknik olup üç tekniğin beraber uygulanmasından oluşmaktadır. Bunlar değişkenlerin alanını ve değişkenlerle ilgili kısıtları tanımlama ve çözüm arama şeklindedir.

Her ne kadar tamamen farklı bir alan gibi doğmuş olsa da, aslında CP de yavaş yavaş yöneylem araştırması ailesinin içinde kendi yerini bulmaktadır. Örneğin Bockmayr ve Kasper, tamsayı programlama (IP, *Integer Programming*) ile CP tekniklerini, Rodošek ve diğerleri (1998) ise MIP ve CP'yi entegre edilmesi konusunu araştırmışlardır. Benzer şekilde Maravelias ve Grossman (2003) çizelgeleme problemlerinin çözümü için hibrid bir MILP – CP modelleme ve çözümü önermektedirler. Bockmayr ve Pisaruk (2003) MIP tekniğinde kullanılan

kesme denklemlerinin elde edilmesi ve / veya modelin çözümsüzlüğünün ispatı için yine kısıt programlama kullanmaktadırlar.

Benzer yöneylem araştırması tekniklerinin kısıt programlama ile çözümü ya da yöneylem ve kısıt programlama yöntemlerinin bir arada kullanılması özellikle yakın geçmişte son derece yaygın bir araştırma alanı haline gelmiştir. Örneğin Caseau ve Laburthe (1995) *Branch and Bound* tekniğinin çizelgeleme problemleri için geliştirilmesinde, Bisdorff ve Laurent (1995) doğrusal programlama problemlerinin çözülmesinde, Little (2000) ve Ottoson (2000) IP yapısındaki kesikli optimizasyon problemlerinin çözülmesinde, Milano (2000) CP ve yöneylem araştırması tekniklerinin entegrasyonunda, Benoist ve diğerleri (2002) ile Tarım ve Miguel (2005) ise Benders Ayırıştırma yönteminin çözülmesi için kısıt programlamadan faydalanmaktadırlar. Özellikle çizelgeleme, planlama ve envanter ilintili konularda kısıt programlamanın kullanımına sayısız örnek bulunması, bu metodun yöneylem araştırması ile entegrasyonu ve bu çalışmadaki modelleme yöntemi olarak seçiminde önemli rol oynamaktadır.

Genel yöneylem araştırması yöntemleri ile modelleme ve çözüm imkanı bulunmayan çok farklı tipteki kısıtlar için modelleme ve çözüm imkanı sunan kısıt programlama, mantık, matematik ve bilgisayar alanlarının bir birleşimi gibi görülebilir ve Lufthansa, Nokia, Renault, Airbus ve Siemens gibi kuruluşlar tarafından, kadro, kaynak, üretim planlama gibi alanlarda aktif olarak kullanılmaktadır.

1.3. ÇALIŞMANIN PLANI

Bu alt bölüm, çalışmanın bu noktadan sonraki seyrini tanımlamayı hedefler. Takip eden bölümde test ile ilintili araştırma alanlarında bugüne kadar yapılmış çalışmalardan en önemli görülen makalelerin incelendiği bir literatür taraması verilecektir. Literatür taraması ile test içerikli araştırmaların tarihsel gelişimi gözlenebilecek olup, araştırma alanlarının temel kolları ve bu çalışmaya gelene

kadar ne tür problemlere çözüm arandığı izlenebilecek ve genel kabul gören araştırma kolları ve konuları hakkında bilgi sahibi olunacaktır.

Bir sonraki bölüm test probleminin temellerini, test akışını, test akışının matematiksel gösterimini ve temel kavramları açıklamaktadır. Aynı bölüm tek özellikli testleri konu alarak bu görevi yerine getirmekte ve daha sonra önce belirli varsayımlar altında karışık tamsayı programlama ve akabinde bu varsayımları gevşeterek kısıt programlama kullanımı ile temel test probleminin nasıl modelleneceğini ve çözüleceğini göstermektedir. Daha sonra elle, MIP ile ve CP ile bulunan çözümlerin karşılaştırılması ve sonuçların irdelenmesi bulunmaktadır.

Dördüncü bölüm ise çok özellikli ürünlerin test edilmesine odaklanmaktadır. Bu tip testler için tanımlanan bağımsız ve etkileşimli test durumları ayrı ayrı ele alınmaktadır. Bağımsız test durumu kısıt programlama ile modellenmekte ve çözülmekte ve sonuçları incelenmektedir. Etkileşimli test ise test tasarımının en zor konusu olara ortaya çıkmaktadır. Bu problem tipi ile ilgili olarak önce Fisher ve diğerleri (2007a ve 2007b) tarafından da ortaya konan kesin hesap yöntemleri incelenmekte, daha sonra Rossi ve diğerleri (2007) tarafından açıklanan yaklaşım ortaya konulmaktadır. Belirli bir hata payı içermesine rağmen küçük problemlerde bile hızla çözümsüzlüğe giden bu zor problem yapısını bir olasılık yaklaşımı ile tahmin eden yaklaşım aktararak, tüm bu bilgilerin ışığında etkileşimli test probleminin girdi ve çıktıların hesaplanması gerçekleştirilmektedir.

Son bölüm ise sonuçları, sonuçların açıklamalarını, araştırmanın içeriğini ve ileri yönelik açık araştırma sorularını ve gelecek çalışmalar için ön görülen sorun ve tavsiyeleri bulundurmaktadır.

2. BÖLÜM

LİTERATÜR ÖZETİ VE DEĞERLENDİRMESİ

2. GİRİŞ

Bu çalışmada ele alındığı şekliyle test, test tasarımı ve test optimizasyonu ile ilgili kavramlar, literatürde oldukça yeni kabul edilebilir. Özellikle bu çalışmada geliştirilen test optimizasyonu yaklaşımı, mevcut çerçevesi ile çok az sayıda akademik araştırmaya konu olmuştur. Yine de hemen her bilimsel araştırma alanında rastlanacağı üzere, bu çalışma da bir sürecin sonucu olarak kabul edilebilecektir. Bu bağlamda bu çalışmaya oluşturulan temelde kullanılan fikirleri irdeleyen çalışmalara ve hatta bu çalışmaların esin aldığı başka çalışmalara rastlamak şüphesiz ki mümkündür.

Çalışmanın bu bölümü, çalışmaya temel kabul edilebilecek ve / veya kullanılan bazı kavramları açıklamakta faydalı olabilecek ve / veya konu ve konunun tarihsel gelişimi hakkında detaylı bilgi verecek bir literatür taramasını içermektedir. Tarama, özellikle telekomünikasyon alanında kullanılan çip ve kartlar, örneğin entegre çipler (IC, *Integrated Chip*), uygulamaya özel entegre çipler (ASIC, *Application Specific Integrated Chip*), bilgisayar ana kartları, telefon ana kartları, çip üstü sistemler (SoC, *System – on – Chip*) gibi, ve sistemler, örneğin büyük ölçekli entegrasyon (LSI, *Large Scale Integration*) ve çok büyük ölçekli entegrasyon (VLSI, *Very Large Scale Integration*) gibi, özetlenebilecek ürün tipi ile ilintili temel test kavramları incelemektedir. Ayrıca bu kavramların akademik çevrelerce ele alınışı, benzer akademik araştırma alanları, kullanılan teknikler ve elde edilen sonuçlarla ilgili bilgi sunmakta ve bu sayede takip eden bölümde açıklanan ve geliştirilen yöntemlerin algılanmasını ve anlamlandırılmasını kolaylaştırmayı amaçlamaktadır.

Test tasarımı ve optimizasyonu konusu elbette yoktan var olmamıştır. Bu alt bölümde incelenen araştırmalardan, konusu itibarıyla ilgili görünse de,

çalışmanın içeriği ile ilgili olmayan ve daha ziyade istatistiksel araştırmaların konusu olan Kalite Kontrol (QC, *Quality Control*), İstatistiksel Süreç Kontrolü (SPC, *Statistical Process Control*), Örneklem (Sampling), Kabule Dayalı Örneklem (AC, *Acceptance Sampling*) gibi alanlara ait çalışmalar hariç tutulmuştur.

2.1. HATA TEMELLİ ÇALIŞMALAR

Sayırsız çalışmalar sunulmuş istatistiksel kalite ve proses kontrolü ve benzeri alanlar mevcut çalışmanın kapsamı içerisinde görülmemiştir. Bu geniş alan bir kenara bırakılırsa, çalışmayı doğrudan ilgilendiren literatürün en eski örneği 1959 tarihli bir çalışmadır. Brian Gluss, bu çalışmada ürünlerde bulunan hataların yakalanmaları durumunu incelemiş ve yepyeni bir alana öncülük etmiştir (1959). Gluss, hata olasılıkları ve bunların dağılımlarını incelemiş ve test süreçlerinde başlangıçtan ürünlerde bulunan hataların ortaya çıkarılmasına kadar geçecek olan sürenin nasıl hesaplanacağına dair matematiksel formüller ortaya koymuştur. Ayrıca bu hataların ortaya çıkma olasılıklarını da ortaya koymuştur. Bu açıdan bakıldığı takdirde, bu çalışmaya temel olan araştırma alanlarının en eskisi hata saptanması (*fault detection*) olarak ortaya çıkmaktadır.

2.1.1. Hata Saptanması

Belirtildiği gibi test optimizasyonu alanına öncülük eden çalışma Gluss (1959) tarafından hazırlanmış ve hata saptanması temelli bir çalışmadır. Uzun yıllar test denince akla gelen en önemli araştırma alanı böylece ortaya çıkmıştır. Takip eden bir çalışmada Gluss, bu kez Firstman ile birlikte benzer bir çalışma daha sunmuştur (1960). Firstman ve Gluss (1960), Gluss'un (1959) çalışmasını bir adım öteye götürmüş ve otomatik hata yerinin saptanmasına yönelik bir program oluşturmaktadırlar. Belirttikleri üzere (Firstman ve Gluss, 1960), o

dönemlerdeki teknoloji ve imkanlar dahilinde bir ürünün (kendi ifadeleri ile sistemin) üzerinde bulunan bir hatanın otomatik bir cihaz kullanılarak saptanması hem çok zor, hem de zaman açısından son derece maliyetli kabul edilmektedir. Günümüz yazılımları ve makinelerinin etkinliği söz konusu olmasa da, Firstman ve Gluss (1960), çalıştıktan sonra değiştirilemeyen fakat çalıştırılmadan önce yeniden programlanıp detay ayar yapılabilen (yani yarı – sabit) bir program geliştirmiş ve bu sürecin temellerini araştırmalarına konu etmişlerdir. Bu çalışmalar açıkça alanlarının ilk örneklerini teşkil etmektedir.

Hataların tespit edilmesi ile ilgili çalışmalar kendi içlerinde birkaç önemli gruba ayrılabilir. Hata tespiti dışında hata yeri belirlenmesi (*fault location*) ve hata kapsamı (*fault coverage*) da önemli ilintili alanlar olarak görülebilecektir. Firstman ve Gluss'un hemen yukarıda belirtilen çalışması hata yeri belirlenmesi alanında yazılmış muhtemel ilk çalışmadır.

Hata yeri saptanması alanında önemli kabul edilecek çalışmalardan biri de Butterworth tarafından yazılmıştır. Butterworth (1972), test modellerinin güvenilirliklerini incelemiş ve hem hatayı yakalayacak uygulanabilir bir test sırası belirlemeye, hem de test süresini (dolayısı ile maliyetini) asgaride tutmaya gayret etmiştir. Çalışma ayrıca bu testin doğrudan ilgilenmediği bir başka alana da referans kabul edilebilir. Bu da Butterworth'un uyguladığı, gerçekçiliği seri çalışan sistemlerde sorgulanabilir olan “n'in k'lısı” (n/k , “*n out of k*”) sistemleridir. Bu sistemlerde varsayım, k veya fazla sayıda bileşeni çalışan n bileşenli bir sistemin çalışacağı, k'dan az sayıda bileşeni çalışan sistemlerin ise hata vereceği şeklindedir. Seth ve Agrawal ise LSI testleri için %100'lük bir hata kapsama oranının genellikle mümkün olmadığını belirtmektedirler (1981).

Butler ve Lieberman (1984) da hata yeri saptanması üzerine çalışmış ve farklı açılımlar sunmuşlardır. Onların çalışması bir ürünün bozulması durumunda hatanın hangi bileşenden kaynaklandığının belirli olmayabileceğini ve böylesi bir durum gerçekleştiği takdirde bu bileşen saptanana dek önceden belirlenmiş bir kontroller serisinin uygulanması gerektiğini söylemektedir. Hata saptanıp ürün

alışır hale getirilmek üzere tamir edilene dek izlenecek kontrol stratejisi maliyetinin yüksek olabileceğini öneren alışma, kontrolde kullanılacak ideal bileşen sırasının pratik olmasa da yapılabilceğini iddia etmektedir. Bu amaçla ürünün yapısının küçük modüller olarak düşünülmesi gerektiği savunulmuş ve sezgisel yöntemler yardımıyla makul ve iktisatlı kontrol stratejileri uygulanabileceğini önermişlerdir (Butler ve Lieberman, 1984).

2.1.2. Hata Kapsanması

Hata kapsanması ise biraz daha farklı bir alandır. Prensip olarak yöneylem araştırmasında karşılaşılan küme kapsama (*set covering*) ve küme paralama (*set partitioning*) problemlerine benzemekte olan bu alan, belirli bir test serisinin, eldeki varsayımsal üründe bulunan teste tabi bileşenlerin (mevcut durumda hataların) hangi oranda kapsandığını yani test edildiğini araştırma konusu kabul etmektedir. Alan doğal olarak testin etkililiğini ölçmeyi ve olası eksikliklerin sebep ve maliyetlerini ve nasıl giderileceğini incelemektedir. Agrawal ve diğerkleri (1982), LSI sistemlerini inceledikleri alışmalarında eksiksiz (hata kapsanması tam olan) bir test serisinin bulunamayacağını savunmaktadırlar ve neden olarak şunları göstermektedirler:

- Mükemmel olmayan hata modelleme (hata, modellenmiş hatalardan biri olmayabilir)
- Tüm özellikler test edilse bile, bazılarının testte aldıkları veriye hassas olmaları
- Test edilebilirlik kaynaklı sınırlılıklar (tasarım nedeniyle test iğnelerinin ipe ulaşamaması)

Agrawal ayrıca Jain ile birlikte istatistiksel hata analizi ve bu kapsamda hata ve hatasızlık simülasyonları hakkında da bilgi vermektedir (1985). Hata kapsanmasının önemine Dear ve diğerkleri (1991) de dikkat çekmektedirler. Dear ve diğerkleri (1991) test için tasarım hakkındaki alışmalarında, genelde

ürün tasarımlarının %40 - % 70 arasında hata kapsanması imkanı verdiklerini fakat piyasada kabul edilebilir bir kalite düzeyi için en az %98 oranında hata kapsanması gerektiğini belirtmektedirler. Bu nedenle testlerde çalışır görünen entegre çiplerin yaklaşık %50'sinin monte edildikleri sistemlerde çalışmadıklarına işaret etmektedirler (Dear ve diğerleri, 1991).

Pomeranz ve Reddy bu alanda yakın zamanlarda yapılmış önemli çalışmalardan birine imza atmışlardır (2003). Pomeranz ve Reddy, depolama zamanı ya da test sürelerinden kaynaklanan şartların bazen ürünün üzerinde gerçekleştirilmesi gereken tüm testlerin yapılamamasına neden olduğunu ve bu durumda olabilecek en fazla sayıda hatanın kapsanarak kısmi bir test gerçekleştirilme zorunluluğu doğduğunu belirtmektedirler. Çalışma böylesi bir durumda maksimal hata kapsanmasının nasıl yapılacağını tanımlamaktadır.

Hata kapsanması ve hata tespiti, test ile ilgili birçok konunun tamamlayıcısı olarak görülebilir. Özellikle bozuklukların dağılımı (üzerinde bulundukları sistem üzerindeki fiziksel dağılımları ve ayrıca hataların olasılık dağılımları) da birçok çalışmada bu kavramlarla birlikte kullanılmış ve beraber analiz edilmiştir. Örnekler için şu çalışmalar incelenebilir: Bose ve Agrawal (2006), Das ve diğerleri (1991 ve 1993), Gaitonde ve Walker (1993), Gaitonde ve diğerleri (1993, 1995 ve 1996), Maxwell ve Aitken (1993), Novak ve diğerleri (1993), Pomeranz ve Reddy (2003), Shi ve Sandborn (2003). Rahatlıkla bu örnekler çoğaltılabilir ve söylenebilir ki test kapsanması çıktı, hatalı ürün oranı ve test tasarımı içerikli makalelerin çoğunda bir yandan göz önünde bulundurulacak faktörlerden bir tanesidir. Bu çalışmada da test kapsanması şeklinde daha sonraki bölümlerde bu kavrama benzer yaklaşımlar irdelenecektir.

2.2. TEST VE TAMİR İÇERİKLİ ÇALIŞMALAR

Şüphesiz ki test tasarımı ve optimizasyonu gibi bir alana gelene kadar incelenmiş konular sadece hata saptanması, kapsanması ve hata yerinin

tespitinden ibaret değildir. Bu çalışmalar şüphesiz test ile ilintili araştırmaların ilk örnekleri olup temellerini hazırlamışlardır. Fakat kısa bir süre sonra hata temelli çalışmalar devam ederken bir yandan da yeni alanlar ortaya çıkmıştır. Zamanla test ve tamir içerikli çalışmalara ilgi artarken yakın geçmişte oldukça spesifik alanlarda çalışmalar yayınlanır hale gelmiştir.

2.2.1. Tamir İçerikli Çalışmalar

Bir diğer temel ve alanının ilk örneği olan çalışma ise Flehinger'e aittir. Bu çalışmada tamir edilebilirlik kavramı ile karşılaşılmaktadır. Flehinger, ürün testinin iki amacı olduğunu ve bunların bozuk olma eğilimi gösteren parçaları ortaya çıkarmak ve ürünün performans özelliklerinin tahmin edilmesini sağlamak olduğunu belirtmektedir (1965).

Çalışma bir Poisson değişkeni olan hata oranının temel kabul edildiği ürünler için hataları “tamir edilebilir” ve “ıskarta – tamir edilemez” olarak sınıflamakta ve optimal test planları önermektedir. Varsayım olarak tamir edilebilir hataların saptanıp yok edilmesini temel amaç kabul ederek test süresinin en aza indirilmesinin optimal test planını sağlayacağı savunulmaktadır (Flehinger, 1965). Hata oranı çözümün bir çıktısı olarak bulunmaktadır ve karar değişkeni değildir. İkinci bir modelde ise hata – ret oranı da modelin içine katılmış ve kısıt olarak kullanılmıştır. Flehinger, bahsi geçen kavramların dağılımını bir fonksiyonda bir araya getirmiş ve matematiksel çözümle (mutlak minimumu bulmak sureti ile) optimal çözümü önermiştir. Flehinger'in bu çalışması (1965) son derece önemlidir zira hem tamir hem de test planı literatürlerine bir geçit görevi görmektedir.

Tamir alanı incelenecek olursa takip eden en eski ilintili çalışma 1976 yılında görülmektedir. Blumenthal ve diğerleri, tamirin her zaman mükemmel olmayabileceğini ve tamir edilen bir ürünün hatasız ilk haline değil, daha eski ve riskli ama çalışan bir ürün haline dönebileceğine değinmektedirler. Bu da

ürünün hayatına etki edecek ve potansiyel olarak beklenenden daha erken arıza oluşmasına neden olabilecektir (Blumenthal ve diğerleri, 1976). Brown ve Proschan (1983), bu fikri genelleştirmiş ve mükemmel ve mükemmel olmayan tamirler için olasılıkların tahmin edilebilirliği varsayımı altında, orijinal hatasız haline çevrilebilecek ürünlerin gerçekleşmeleri arasında geçecek süreyi teorik olarak hesaplamışlardır. Çalışmalarında bolca stokastik süreçlerden ve matematiksel istatistikten faydalanmışlardır (Brown ve Proschan, 1983).

Daha yakın bir çalışmada ise Cui ve diğerleri (2004) bahsedilen bilgileri kullanarak yeni bir karşılaştırma öne sürmüşlerdir. Eğer hem mükemmel tamir ile ürünün orijinal hatasız haline döndürülmesi, hem de az tamir ve çaba ile ürünün sadece çalışır hale getirilmesi imkanı bulunmakta ise, arada bir ödünleşim olanağı bulunmaktadır ve iktisadi açıdan olası fırsatların değerlendirilmesi gerekmektedir (Cui ve diğerleri, 2004).

Fakat belirtmek gerekir ki, burada bahsedilen durum Blumenthal ve diğerleri, 1976 ve Brown ve Proschan, 1983 tarafından ideal ürün ömrü ve arıza beklentisine sahip olmayan bir duruma denktir. Değerlendirmeleri ise şu şekildedir: herhangi bir ürün için artan hata oranı (IFR, *Increasing Failure Rate*) azalan hata oranı (DFR, *Decreasing Failure Rate*) ya da sabit hata oranı (CFR, *Constant Failure Rate*) gerçekleşebilir. İlk durum söz konusu ise iki tamir tipi dengeli kullanılmalıdır. İkinci durum söz konusu ise olası en az tamir yoluna gidilmelidir. Üründe hata oranı sabit ise her iki strateji de eşit beklenen ürün ömrü değerini vermektedir (Cui ve diğerleri, 2004). Buradan görülebilir ki Cui ve diğerleri amaç fonksiyonu olarak ürün ömrünü kullanmakta ve maksimize etmektedirler (2004).

2.2.2. Test Süresi Temelli Test Planlaması ve Test Çizelgeleme

Flehinger'in çalışmasında (1965) irdelenen bir diğer kavram da test planlamasıdır. Bu konunun çok geniş açılımları söz konusudur. Örneğin test

planlaması ile test süresinin ya da maliyetin minimize edildiği çalışmalar, test stratejisi saptama, test çizelgeleme, test üretimi (*test generation*), ve benzeri kavramlar oldukça yakından ilintilidir.

Test süresi yaygın kullanılan bir test optimizasyonu amaç fonksiyonudur. Birçok elektrik – elektronik mühendisliği, yöneylem araştırması ve benzeri teknik çalışmada toplam test süresinin en aza indirilmeye çalışıldığı gözlenebilir. Mevcut çalışmalar arasında Flehinger (1965), Brown ve Proschan (1983) bu yaklaşımları kullanmışlardır. Ayrıca Goel, teknik bir çalışmada özel bir test türü olan “*Gate Count*” ile test süresi arasındaki ilişkiyi incelemiştir (1980). Ebadi ve Ivanov (2001), SoC’lerin test edilebilirliği incelenirken test mekanizmasının ürüne erişim (TAM, *Test Access Mechanism*) özelliğinin mimarisindeki en önemli bileşenin test süresi olduğunu belirtmektedir. Onlara göre, test erişim mimarisi test süresini minimize etmek sureti ile optimize edilmelidir. Bu amaçla Ebadi ve Ivanov genetik algoritmaları kullanmışlardır ve çözdükleri problemi SoC test ve kontrolü sistemleri oluşturmada temel bir problem olarak görmekteydiler.

Pouget ve diğerleri ise tamamen test sürelerinin minimize edilmesine odaklandıkları çalışmalarında (2003), SoC’lerin test edilmesini mercek altına almışlar ve SoC test çözümü oluşturmada en önemli konunun test süresinin minimize edilmesi olduğunu tekrarlamışlardır. Pouget ve diğerleri problemi sezgisel yöntemler kullanarak çözmektedirler. Ayrıca dikkat edilmesi gereken bir nokta da Pouget ve diğerlerinin de belirttikleri gibi (2003) verimli bir test çizelgesi eş zamanlı testler kullanarak test süresini azaltabileceği gerçeğinin avantaj gibi görünmesidir. Fakat bu durum sistem içindeki faaliyeti ve haliyle güç kullanımını artırarak sisteme zarar verebilecektir.

Böylece test süreleri ile test çizelgelemenin nasıl bir arada kullanıldığı da görülmektedir. Test çizelgeleme test literatüründe son derece geniş bir yere sahip bir alandır. Genel öbek büyüklüğü ve çizelgeleme (*Lot Sizing and Scheduling*), üretim çizelgeleme (*Production Scheduling*) ve hat dengeleme

(*Line Balancing*) gibi konularla benzerlik gösteren yaklaşımlar ve teknikler ile de araştırılan test çizelgeleme, özellikle son birkaç on yıl içinde sıklıkla araştırma konusu olmaktadır. Bu bölümde alanın en önemli görülen çalışmalarına yer verilecektir.

Huss ve Gyurcsik (1991), hatalı devrelerin erken saptanması ve ortalama test süresinin minimize edilmesini hedefledikleri çalışmalarında IC'ler için dinamik programlama kullanarak çizelgeleme problemini çözmeyi denemekte ve optimal sonucu elde etmektedirler. Belirttikleri ilginç bir nokta ise standart analog ürünlerde genellikle testler ve toleranslarının genellikle veri kitaplarında önceden saptanmış olduğu ve yapılabilecek tek şeyin çizelgeleme ile testlerin sıralarının değiştirilmesi olduğudur (Huss ve Gyurcsik, 1991).

Çizelgelemenin gerekliliği için farklı bir yaklaşım Chou ve diğerlerinden gelmektedir (1997). Onlara göre IC, dijital kart ya da benzeri bir sistemin testi sırasında tüm testler aynı anda yapılamayacağı için test çizelgeleme gerekmektedir. Fakat değişmeyen bir gerçek, Chou ve diğerlerinin de test süresini en aza indirmeye çalışmasıdır (1997). Çözümlerini ise dinamik programlama ile değil kendi algoritmaları ile arama uzayını küçültme yoluna giderek bulmaktadırlar (Chou ve diğerleri, 1997).

Jiang ve Vinnakota ise Huss ve Gyurcsik ile uyumlu olarak (1991) test çizelgelemesinin amacının hataların testin geç evrelerinden ziyade erken dönemlerinde fark edilmesini sağlamak olması gerektiğini önermektedirler (2001). Literatürde de geniş kabul görmüş bu önerileri (örneğin Dislis ve diğerleri, 1993c) bir yana, çalışmaları ise test alanında teorik bir yeniliğe değil, çizelgeleme yöntemlerinin çözülmesinde polinom zamanlı bir çözüm yöntemi oluşturacak bir sezgisel yöntem geliştirmeye odaklanmıştır (Jiang ve Vinnakota, 2001).

Test zamanını minimize etmeye çalışan bir başka çalışmada ise SoC tasarımlarının test edilmesindeki temel sorunun yüksek veri birikimi oluşması

olduğu belirtilmektedir (Larsson ve diğerleri, 2002). Çalışma bu sorunun üstesinden gelmek için birbirine bağımlı iki problemin incelenmesi gerektiğini belirtmektedir:

- Sade (minimal) bir TAM (Test Erişim Mekanizması) oluşturulması
- Test zamanını minimize eden bir test çizelgesi oluşturulması

Larsson (2002) ve diğerleri de Jiang ve Vinnakota (2001) gibi optimal ve hızlı bir çözüm bulma çabası göstermektedirler. Çalışma Pouget ve diğerleri (2003) gibi eş zamanlı testleri, riskli olduğu için göz ardı etmektedir. Çalışma tamamen teknik test parametrelerinin tasarımına yönelik devam etmekte ve yazarlar kendi geliştirdikleri bir algoritma ile problemi çözmektedirler. Belirtilmelidir ki Pouget ve diğerlerinin çalışması (2003) de test çizelgeleme ile ilgili önemli bir örnektir.

Larsson ve Fujiwara (2003a), yukarıda belirtilen bilgiye paralel bir şekilde, TAM üzerindeki testlerin çizelgelenmesini, benzer makineler üzerinde yapılacak klasik bağımsız görev çizelgeleme problemi ile denk tutmaktadırlar. Bu genel yaklaşımda örtüşen testlere dikkat edildiği takdirde doğrusal zamanda bir optimal çözüm bulunması mümkündür (Larsson ve Fujiwara, 2003a). İkili aynı sene yayımladıkları benzer bir çalışmada (Larsson ve Fujiwara, 2003b) aynı konu ile ilgilenmiş ve genel bir test çözümü geliştirmenin prensiplerini belirtmişlerdir:

- Her teste tabi birim için test seçimi
- İlgili testlere ait kaynakların yerleşim planı
- Birimler ve test kaynaklarını birleştirecek bir erişim sistemi (TAM)
- Testlerin sıraları (Test çizelgesi)

Görüleceği gibi bu çalışmanın konusu hem test sıraları hem de test ve donanım seçimi test çözümlemenin hayati bileşenleridir. TAM tasarımı ise tamamen elektronik mühendislerinin araştırmalarına konu olacak teknik bir araştırma alanıdır.

Kotasek ve diğerkleri ise test süresi ve güç kullanımını minimize etme prensibinden sapmamakla birlikte, içinde – yerleşik – kendi – kendine – test tekniğine (BIST – *Built In Self Test*) odaklanmaktadırlar (2003). Amaçları, BIST'in paralel (ya da eşzamanlı) test yapabilme gücünden faydalanarak test süresinin azaltılmasına imkan verebilmek olup, grafik yöntemler ve genetik algoritmalar ile problemi çözme yoluna gitmişlerdir (Kotasek ve diğerkleri, 2003).

Evrimsel algoritmaların (EA, *Evolutionary Algorithms*) test çizelgelemede kullanımına tek örnek şüphesiz sadece genetik algoritmalar değildir. Örneğin Mergenthaler ve diğerkleri (2003), IC için test çizelgelemede gereksiz fazlalığı (örneğin test tekrarı) azaltmak için tavlama simülasyonu (SA, *Simulated Annealing*) yöntemini kullanmaktadırlar. Amaç fonksiyonu bu çalışmada da test süresinin minimize edilmesi olarak ortaya çıkmıştır. Mergenthaler ve diğerkleri, optimizasyon sonucunda test süresi ve dolayısı ile test maliyetinde önemli kabul edilecek tasarruf sağlamışlardır. Aynı sene yayımlanan bir diğerk makalede ise Zou ve diğerkleri (2003) Larsson ve Fujiwara'nın prensiplerini (2003b) ve Mergenthaler ve diğerklerinin yöntemlerini (2003) doğrular nitelikte bir çalışma sunmuşlardır. Zou ve diğerklerine göre (2003) ürünün piyasaya çıkış süresini azaltmak amacıyla olan SoC üreticileri IC teknolojisi kullanmaya başlamışlar ve böylece SOC testleri önemli bir problem haline gelmiştir. SOC testlerinin temel amacı test süresini azaltmaktır ve üç temel kısımda incelenmektedir. Bunlar:

- Test erişim mekanizması (TAM)
- Ekipman optimizasyonu
- Test çizelgeleme

Çalışmada son iki kısmın optimize edilmesi için test süresini en aza indirme problemi iki boyutlu paketleme (*bin packing*) problemi olarak ele alınmış ve tavlama simülasyonu ile çözülmüştür.

Bu çalışmalar dışında farklı alanlara odaklanan test çizelgeleme çalışmaları da mevcuttur. Örneğin Cota ve diğerleri (2002) test çizelgelemenin yolunun test edilebilir tasarımların yapılmasından geçtiğini öne sürmektedir. Biasizzo ve diğerleri (1998) ardışık teşhis (*sequential diagnosis*) içerikli çalışmalarında teşhis sisteminin, başlangıçtaki durumu (hatalı – hatasız ürün gibi) belirli olmayan bir sistemin durumunu saptamaya yönelik bir prosedürün hata olasılıkları ve maliyetler göz önüne alınarak minimum test maliyeti ile sonuçlandırılmasını konu etmektedirler. Bu amaçla simetrik ve asimetrik testler için test sıralama problemini çözmektedirler (Biasizzo ve diğerleri 1998). Mailharro ise (1998), yapılandırma (*configuration*) problemini kısıt programlama ile çözmektedir. Benzer fakat daha ilintili bir çalışmada ise Little ve diğerleri (2006) termal test için test çizelgeleme problemini yine kısıt programlama kullanarak çözmektedirler. Çalışmada talep, ürünlerin test için içine konulacağı yuvaların kapasitesi ve tipleri (zira belirli ürünler sadece belirli yuvalara konabilmektedir), yuvaların ısı tipleri (yine belirli ürünler sadece belirli ısılarda test edilecektir), ve yapılandırma kuralları kısıtlar olarak ortaya çıkmaktadır (Little ve diğerleri, 2006). Haftalık çizelgeleme amaçlı olarak, en kısa sayıda test ile talebin karşılaştırılması şeklinde görülen amaç fonksiyonu kısıt programlama ile çözülmüş ve optimal çözüm elde edilmiştir.

Bu açımdan bakıldığında test süresi ile ilgili optimizasyon çalışmaları ve test çizelgeleme çalışmaları adeta birbirinin tamamlayıcısı gibi görünmektedirler. Test planlaması, test stratejisi belirleme, test üretimi, maliyet optimizasyonu içerikli alanlarda da durum çok farklı görünmemektedir.

2.2.3. Test Planlaması ve Test Stratejisi Seçimi

Test planlaması ve test stratejisi konusunda Dislis ve ekibi literatüre çok sayıda çalışma sunmaktadırlar. Örneğin Dislis ve diğerleri (1988), daha önce bahsedilen son derece önemli bir kavramın yeniden hatırlanmasına vesile olmaktadır:

“Ekonomik olarak ifade etmek gerekirse, testin çok geç evrelerinde fark edilen hatalar ya da bazı parçaların test edilebilmesinde sorunu olduğu, test kalitesinin düşmesine ya da ürünün yeniden tasarlanması maliyetinin ortaya çıkmasına yol açacaktır. Teste uygun tasarım zorunluluğu ve ucuz test yöntemleri de durumu karmaşıktırmaktadır. Bu noktada ürün tasarımı ve piyasa koşulları da göz önüne alarak ekonomik test süreçleri planlanmalıdır, zira yönetim düzeyinde projeler ekonomik sonuçları ile değerlendirilmektedir.”

DISLIS, C., I. D. DEAR, S. C. LAU, J. R. MILES ve A. P. AMBLER. “Hierarchical Test Strategy Planning Based on Cost Evaluation.” *IEE Colloquium on Computer Aided Test and Diagnosis*, October 1988: 7/1.

Çalışmada, test metotlarını ve stratejilerle ilgili verileri bir hafızada tutan ve test planlamacısının tasarım gereçlerine bir program üzerinden ulaşip ekonomik test tasarımını teknik olarak yapan bir planlama programı sunulmaktadır (Dislis ve diğerleri, 1988). Program tasarım aşamasında farklı test stratejilerinin maliyetlerini karşılaştırmak suretiyle uygun tasarımın yapılmasına yardım etmektedir. Ayrıca teste tabi parçaların ebat, maliyet, tip, konum ve test edilebilirliğini de göz önüne alarak ekipman ihtiyacını saptamaktadır. Test stratejisi maliyetleri hesaplanırken, parça, kart ve sistem bazında tasarım, üretim ve test maliyetleri göz önüne alınmaktadır. Bu maliyetler de kendi içinde makine, işgücü, zaman ve malzeme maliyetleri olarak ayrıştırılmıştır. Test maliyetleri ise test oluşturma ve test uygulama maliyetleri olarak ikiye ayrılmaktadır. Modelin çıktısı ise alternatif test stratejilerinin karşılaştırmasını vermektedir. Görüldüğü gibi Dislis ve diğerleri (1988) hem test stratejisi, hem ürün tasarımı, hem de test maliyetleri ile ilgili bilgi ve karşılaştırma imkanı verecek bir yazılım ortaya koymaktadırlar. Aynı grubun bir başka çalışmasında ise (Dear ve diğerleri, 1989) iyi bir test planlama programında bulunması gereken özellikler şunlardır:

- İyi bir kullanıcı arayüzü
- Bilgisayar destekli tasarım (CAD, *Computer Aided Design*) sistemlerinden tasarım verilerinin alınabilmesi
- Test program tanımı üretilmesi

- “Test için tasarım” dokümanlarının değiştirilebilmesi
- Değişik hiyerarşik kademeler arasında test edilebilirlik karşılaştırması
- Test bilgisi güncellenmesi ve genel bakımının kolay olması
- Gerçek zamanda cevap alınabilmesi

Robinson (1989) ise, herhangi bir stratejinin her durum için ideal olamayacağını savunmakta ve çoklu test stratejisi kullanılmasını önermektedir. Böylece test geliştirilmesi sürecinde ekonomik ve zamansal kısıtlar da göz önünde bulundurularak test yaklaşımlarının uygun bir kombinasyonu seçilebilecektir. Sistem uygun olduğu takdirde Robinson bu yaklaşımın çok daha ucuza geleceğini savunmaktadır. Robinson, seçilecek stratejinin netleşmesi için bir de kural önermektedir:

*“Eğer, Ürün doğru şekilde tasarlanmış,
Ve, Tüm bileşenleri hatasız çalışıyorsa,
Ve, Tüm parçaları doğru şekilde monte edilmişse,
O zaman, Ürün hatasız çalışacaktır.”*

ROBINSON, G. D. “Test Program Development Using Multiple Test Strategies.” *Proceedings of the IEEE Automatic Testing Conference AUTOTESTCON'89*, 1989: 11.

Bu çalışma kapsamında fazla teknik görülmesi ihtimaline rağmen, Robinson'un önerdiği test stratejisi seçimine yardım edecek özellikler aşağıdaki tabloda görülebilecektir.

Tablo 1. Değişik Test Stratejilerinin Özellikleri

Strateji	Test Kalitesi	Teşhis Kalitesi	Test Üretim Maliyeti	Test Cihaz Maliyeti	Uygulama Maliyeti
Bağlantı	Düşük	Yüksek	Düşük	Düşük	Düşük
Dijital Fonksiyonel	Yüksek	Yüksek	Yüksek	Orta	Her yerde
Analog Fonksiyonel	Orta	Düşük	Yüksek	Düşük	Her yerde
Analog Devre İçi	Orta	Yüksek	Düşük	Orta	Orta
Dijital Devre İçi	Orta	Yüksek	Düşük	Orta	Orta
Öykünüm	Yüksek	Düşük	Her yerde	Düşük	Düşük

ROBINSON, G. D. “Test Program Development Using Multiple Test Strategies.” *IEEE*, 1989: 15.

Benzer ve yakın tarihli bir çalışmada Chien ve diğerleri (2007), IC'lerin düzgün test edilmesi için test kapasitesinin, ürün oranının ve döngü süresinin dikkate alınması gerektiğini ve bu faktörler açısından verimli bir test süreci elde edebilmek için alternatif test stratejilerinin karşılaştırılarak doğru yolun bulunması gerektiğini savunmaktadırlar. Pratikte test stratejilerinin tasarım ve seçiminde, maliyet, kalite ve risk değerlendirmelerinin kapalı olarak yapıldığını öne süren Chien ve diğerleri, karar aşamasının detaylı bir analizinin yapılmadığına inanmaktadırlar (2007). Onlar bu ödünleşimleri ve başta belirttikleri faktörleri de göz önüne alan bir taslak hazırlamış testten elde edilecek kar ve müşteri memnuniyetini artırırken test stratejilerinin maliyet ve belirsizliklerini de değerlendiren bir istatistiksel analiz ve karar yapısı oluşturmuşlardır.

Dislis ve diğerleri (1989b), bir başka çalışmalarında test stratejisi ile ilgili yorumlarında optimal test stratejisinin, ürünün tipi, test metodu, firma kaynakları (test makinesi yatırımı), ve test sürecine yatkınlığına bağlı olduğunu ve bunların tümünün de test maliyetlerini etkilemekte olduklarını belirtmektedir. Optimal test stratejisi seçimi ve bundan kaynaklanacak güvenilirlik artışı, tamir ve alanda gerçekleşecek hatalardan kaynaklanacak maliyetleri ciddi düzeyde azaltabilecektir (Dislis ve diğerleri, 1989b).

Dislis ve diğerleri (1993b), "Test Planlayıcısı" (*Test Planner*) isimli bir de program geliştirmişlerdir. Program, ekonomik ve tasarım özellikleri biliniyor ise, ön tanımlı test metotlarını, tasarım içindeki fonksiyonel bölümlere yerleştirmeye yaramaktadır. Sonucun maliyet açısından optimal olması ve kullanıcının belirlediği olası kısıtların (örneğin hata kapsanma oranı ya da test süresi) da ihlal edilmemiş olması programın başarısı olarak kabul edilmektedir (Dislis ve diğerleri 1993b).

Seri sistemleri inceleyen ve maliyet kavramını da test stratejisi seçimine dahil eden bir çalışmada ise Nachlas ve diğerleri (1990), test maliyetlerini değerlendirecek iki maliyet temelli model geliştirmişlerdir. Bu iki model hatasız

ve hatalı test durumları için geliştirilmiştir. Modeller herhangi bir bileşenin hatalı olma ve ürünün testi geçememe olasılıklarının bulunması ve kullanılması temeline dayanmaktadır. Sezgisel yöntemlerle yaptıkları çözümlerin optimal test sıralamalarını verdikleri görülmektedir. Nachlas ve diğerleri, bir test stratejisi seçiminin maliyet odaklı üç sonucu olduğunu söylemektedirler:

- Test sayısına dayanan test maliyeti
- Yanlış karar sonucu hatasız ürünlerin tamir edilme maliyeti
- Yanlış karar sonucu hatalı ürünlerin müşteriye sevk edilme maliyeti

Takip eden bir çalışmada ise Moore (1994) yine aynı konuya odaklanmıştır. Sunulan sistem, test kapsamı ya da ürün oranından feragat etmeden en hesaplı test sürecinin bulunması amacını taşımaktadır. Model aynı zamanda hatalı çıktı oranını da tahmin etmekte ve deneme yanılma yöntemleriyle olası test senaryolarını değerlendirebilmektedir. Moore, test maliyetlerinin ve kalite gereksinimlerinin giderek artmasının bir sonucu olarak, amacın test miktarını, kalite düzeyini düşürmeden azaltmak olduğunu savunmakta ve makul ürünler elde edecek şekilde testlerin %80 azaltılabileceğini önermektedir (1994). Test sürecinin optimizasyonu bile toplam süreç maliyetinde ciddi iktisada imkan tanımaktadır (Moore, 1994).

Test süreci optimizasyonu ile ilgili kayda değer bir çalışma da Barrett ve diğerleri (1999) tarafından yapılmıştır. Çalışma sevk edilmiş ürünlerde rastlanan bozuklukları göz önüne alarak test sürecinin sürekli optimizasyonunu tarif eden bir metot öne sürmektedir. Test sürecinin etkililiğini değerlendiren nümerik teknikler sayesinde belirli hatalar saptanmış ve bu hataların test edilmelerinin daha hassas ve dikkatli gerçekleştirilmesi önerilmiştir (Barrett ve diğerleri, 1999).

Test miktarının azaltılmasını savunan başka çalışmalar da bulunmaktadır. Örneğin Milor ve Sangiovanni – Vincentelli, analog testlerin, kes – ve – dene türü sezgisel yöntemler kullanılarak yapılan zorlu bir test olduğunu ve yüksek

maliyetleri göz önüne alındığında, bu testlerin hataları saptayacak şekilde tasarlanmasının önem kazandığını belirtmektedir (1990). Testlerin sürelerinin uzaması, ekipman bütçesine ayrılan sermaye maliyetinin oldukça yükselmesine neden olmakta, böylece test sürelerinin azaltılmasını gerekli kılmaktadır. Çalışma, yapılacak testleri, parametrik hatası bulunan bileşenleri saptayacak şekilde asgariye indirgemeye çalışan bir algoritma önermektedir. Algoritma, otomatik olarak test kümeleri oluşturarak test süresini azaltmaya çalışmaktadır. Sonuçlar ise hata kapsamında bir azalma oluşturmadan test süresinin ciddi şekilde azaltılabileceğini göstermektedir (Milor ve Sangiovanni – Vincentelli 1990). Bir diğer çalışmada ise Chen ve Lau (1998), bir test sekansındaki testlerin örtüşmesi durumunda sekansın daha küçük bir alt kümesinin de aynı kaliteyi sağlayabileceği ve bu fazla testlerin elimine edilmemesi durumunda gereksiz bir maliyetin oluştuğuna işaret etmektedirler.

2.2.4. Test Üretimi

Tüm bu konularla ilintili ve bu çalışmanın içeriğine prensipte benzeyen bir diğer konu ise test üretimidir (*Test Generation*). Abadir ve Reghbatı'nın (1983) ifadesiyle test üretimi, açık bir test sürecinin en önemli bileşenidir. Test üretiminin temel amacı, teste tabi ürüne uygulandığında ürünün hatalı bileşenlerini duyarlı hale getirerek hatalı bir sinyal üretmesini sağlamak için izlenecek test sıralamasını üretmektir (Abadir ve Reghbatı, 1983). Bu açıdan bakıldığında test çizelgeleme ile test üretimi birbirine son derece yakındır. Aradaki tek fark, test çizelgelemede, belirli bir test kümesinin sıralanması, test üretiminde ise potansiyel testler havuzundan uygun testlerin seçilmesi ve sıralanmasıdır. Başka bir deyişle test üretimi, test çizelgelemeden farklı olarak, test ve hata kapsanmasını da kendi içinde gerçekleştirmektedir. Abadir ve Reghbatı (1983), bu durumla ilgili olarak şöyle söylemektedir:

“Bir test serisi, teste tabi ürünlerdeki hataları yüksek bir yüzdeyle yakalamaktaysa başarılı kabul edilir. Başka bir deyişle, hata kapsama oranı yüksekse başarılı kabul edilir.”

ABADIR, M. S., ve H. K. REGHBATI. “LSI Testing Techniques.” *IEEE Micro*, February 1983: 36.

Yine Abadir ve Reghbatı'ye göre (1983) titiz bir test üretiminin sağlıklı sonuç verebilmesi için şart görülecek üç önemli bileşeni bulunmaktadır:

- İyi bir tanımlayıcı model seçimi: Bu model sistemin tüm faaliyetlerini gerçekçi bir şekilde yansıtabilmelidir.
- Test üretimi sırasında dikkate alınacak hataları kapsayan bir hata modeli: Modelin içerdiği olası hataların oranı maksimize edilirken, modelde kullanılacak testlerin maliyeti minimize edilmelidir.
- Hata modelindeki tüm hataları saptayabilecek testlerin üretilmesi: Bu noktada iki zorluk söz konusudur. İlki hata varsa onu ortaya çıkarmaya zorlayacak bir test serisi bulunmasıdır. Diğeri ise testin hataya hassas olması yani hatanın etkisinin gözlenebilir bir çıktıya dönüşebilmesi (yani saptanabilmesi) şeklindedir.

Abadir ve Reghbatı, takip eden yıllarda IC'ler için iki değişkenli bir test üretimini örnekleyen bir çalışma da ortaya koymaktadırlar (Abadir ve Reghbatı, 1986). Dear ve diğerleri (1991) test üretimi maliyetlerine de değindikleri çalışmalarında test üretiminin tekrarlamayan, bir kereye tabi, fakat kesin hesaplanması çok zor bir maliyet kalemi olduğunu belirtmektedirler. Haliyle açıkça görülmektedir ki sabit maliyet rolü oynayan bu maliyet, üretim miktarı ile ters orantılı olarak önem kazanmaktadır. Dear ve diğerleri (1991), otomatik test üretiminin (ATPG, *Automatic Test Pattern Generation*) genellikle %98 civarında hata kapsanması sağladığını ve elle müdahale edilerek bu oranın artırılabilceğini belirtmektedirler.

Rudnick ve diğerleri (1994) ise Abadir ve Reghbatı'nın (1986) görüşünden ayrılarak, hata odaklı test üretim tekniklerinin hem oldukça karmaşık, hem de

zaman alıcı olduklarını önererek, test üretimi için genetik algoritmalarından faydalanmak gerektiğini savunmaktadırlar. Sıralı devreler için yaptıkları çalışmada, genetik algoritmaları aday test sıralarını evrimleştirmekte kullanmakta ve genetik algoritmalar gibi sezgisel bir yöntem aracılığıyla belirlenimci yöntemlerden daha yüksek hata kapsama oranı ve daha hızlı çözüm zamanına sahip bir model geliştirmektedirler (Rudnick ve diğerleri, 1994).

Rudnick ve diğerleri (1997) test üretiminin geniş bir alanda arama yapısı bulunduğu için genetik algoritmalar açısından ideal bir uygulama alanı olduğunu savundukları başka bir çalışmada ise uygulamanın detaylarından da bahsetmektedirler. Buna göre, test üreticisi bağımsız test vektörleri üreterek işe başlanmaktadır. Hata kapsanmasında bir iyileştirme yapılamayacak noktada ise test sıraları oluşturulmakta ve yine iyileştirme imkanı kalmadığında süreç durmaktadır. Genetik algoritma bir rastsal popülasyonu alarak evrimleştirmekte ve bir hata simülatörü de evrimleşen nesillerin sağlığını kontrol etmektedir. Birkaç nesilden sonra bulunan en iyi test popülasyondan çekilip kaydedilecektir. Yeni bağımsız testler için seçim ölçütü bu en iyi test ile karşılaştırılmaktan geçmektedir. İyileştirme yapılamayacak hale gelinene kadar bağımsız testlerin evrimleşmesine devam edilir ve son nesilden sonra en iyi testler alınır. Yine de Rudnick ve diğerleri (1997), bu çalışmada belirlenimci yöntemler yerine tamamen genetik algoritmaların kullanılması savlarını yumuşatmış ve test edilemeyen hatalar dışında belirlenimci yöntemlerin ve genetik algoritmaların bir arada kullanılmasının yeğ olduğunu savunmuşlardır. Yu ve diğerleri (2002) ise yine genetik algoritmaları kullanarak test üretimi problemini çözmekte ve zaman ve hata kapsama oranı açısından bu yaklaşımın diğerlerine göre üstün olduğunu savunmaktadırlar.

Cohen ve diğerleri ise (1996), test üretiminin aylarca sürebilecek zorlu bir uğraş olduğunu ve çok masraflı olduğunu onaylamaktadırlar. ISO 9000 sistemlerinin bileşenlerde aranacak özellikleri saptadığını, fakat bu bileşenlerin hatasız olmasının da oluşturdukları sistemlerin hatasız çalışacağını garanti edemediğini

savunmaktadırlar. Kombinatoriyal tasarım teorisi (*Combinatorial Design Theory*) temelli geliştirdikleri otomatik etkin test üreticisi (AETG, *Automatic Efficient Test Generator*) ile test üretimini kolaylaştırdıklarını savunmaktadırlar (Cohen ve diğerleri, 1996). Bilineceği üzere kombinatoriyal tasarım, deneysel tasarım (*Experimental Design*) gibi istatistiksel bir deney tasarlama yöntemidir. Ana fikir olarak, bu yaklaşımın klasik testlerden farkı, test sırasında ürünün teste tabi bileşenleri için sadece birer senaryo hazırlamaktan ziyade bileşenlerden ikili, üçlü ve n’li parametreler oluşturacak şekilde senaryolar hazırlamak ve çapraz kontroller ile ürün kalitesini taahhüt etmesindedir. Bir diğer çalışma ise kolay ve yüksek hata kapsama oranına sahip test üretimi yapmanın ancak test edilebilirliği de göz önüne alan ürün tasarımından geçtiğini savunmaktadırlar (Dislis ve diğerleri, 1989a).

Test üretimi probleminin çözümü için farklı yöntemler kullanılması da şüphesiz mümkündür. Örneğin Adams ve Hochbaum (1997) bu problemi bir tur kapsama problemine (*Tour Covering Problem*) benzeterek bir kombinatoriyal optimizasyon problemi olarak çözmektedirler. Sezgisel yöntemler geliştirerek çözdükleri problemin sonuçlarının ise mevcut yöntemlerden daha hızlı ve ekonomik olduklarını belirtmektedirler.

2.2.5. Muayene – Deney Optimizasyonu

Test çizelgeleme ve test üretimi ile benzeşen bir diğer alan da muayene – deney optimizasyonu olarak kabul edilebilir. Basit ve genel ifadelerle test çizelgeleme, testleri sıraya koymayı, test üretimi ise belirli kısıtlar altında uygun bir test planı oluşturmayı içermektedir. Muayene alanındaki çalışmalar ise genellikle muayene noktalarının üretim sürecinin neresine konulması gerektiğini, ideal muayene noktası sayısını, muayene yoğunluğunu, genel bir muayene planı oluşturmayı içermektedir.

Kakade ve diğerleri (2004) ve Penn ve Raviv (2007) muayene ile ilgili çalışmaları iki grupta sınıflandırmayı uygun görmektedirler. Her iki çalışmaya göre de ilk gruptaki çalışmalar bir üretim sistemi dahilinde muayenenin optimal dağıtımıdır. Bu çalışmalar bir muayene birimleri havuzundan hangi birimlerin hemen sonrasına bir muayene birimi konulması gerektiğini bulmaya odaklanmaktadır. İkinci gruptaki çalışmalar ise sistemdeki değişik yerlerde muayeneye tabi ürünlerin hangi oranda seçilmesi gerektiğini yani toplam ürünlerin yüzde kaçının muayene edilmesinin doğru olacağını saptanmasını araştıran çalışmalardır. Kakade ve diğerlerine göre (2004) sayıları çok olmasa da her iki konuyu araştıran çalışmalar da bulunmaktadır.

Van Volsem ve diğerleri (2007) ise tek ve çok aşamalı üretim süreçleri açısından muayenenin ne anlama geldiğini daha net açıklamaktadırlar. Tek aşamalı üretim süreçlerinde muayene, örneklem hacmi (*sample size*), örneklem alma sıklığı (*sampling frequency*) ve kabul sınırlarının (*acceptance limits*) katılığından ibarettir. Böylece muayene planlama problemi sadece bu parametrelerin toplam muayene maliyetini en aza çekecek değerlerini bulmaktan geçecektir. Çok aşamalı üretimde ise bu karar değişkenlerine sistemdeki muayene noktalarının sayısı ve yerleri eklenecektir.

Muayene ile ilgili çalışmaların üretim sırasında kontrol, ara kontrol ve son kontrol gibi algılanabileceği gerçeği düşünülürse, yapılmaya çalışılan işin de test çizelgeleme ve test üretimine benzediği söylenebilir. Her ne kadar muayene durumunda yapılacak kontroller belli olsa ve problem yapısı yer yer önceki çalışmalardan ayrılrsa da, bu çalışmanın içeriği açısından bu alanda yapılmış bilimsel araştırmaların incelenmesi önemli görülmektedir.

Bu bulanık noktayı en güzel netleştiren çalışmalardan biri Chevalier ve Wein tarafından ortaya konmaktadır (1997). Onlara göre devre ve kartların üretiminden gerçekleşecek bazı testlerin montaj aşamasında yapılması gereklidir ve test stratejisi ve muayene planının eş zamanlı optimize edilmesi böylesi durumlarda bir gereklilik halini alabilmektedir. Böylece test stratejisi ve

muayene stratejisinin hem her zaman birbirlerinden ayrılamayabilecekleri, hem de oldukça yakından ilintili oldukları görülmektedir. Yaklaşımları test, tamir ve hatası saptanamayıp sevk edilen hatalı ürünlerin toplam maliyetlerini minimize etmek üzerine kurulmuştur.

Viswanadham ve diğerleri (1996) ise ürün kalitesi ve proses performansının garanti edilebilmesi için muayene noktalarının saptanması gerektiğini öne sürmektedir. Çalışma muayene, üretim, hurda ürün, testi geçmiş hatalı ürün maliyetlerinden oluşan bir toplam maliyeti minimize edecek muayene stratejisini bulmayı amaçlamaktadır. Gerektiği takdirde çıktıların kalite oranı, kullanılabilecek en fazla muayene noktası sayısı ve bir noktada izin verilecek en fazla muayene tekrarı sayısı gibi kısıtların da göz önüne alınması tavsiye edilmektedir. Çalışmada geliştirilen iki modelden biri tavlama simülasyonu, diğeri ise genetik algoritmalarla çözülmektedir. Chen (1997) ise muayene stratejisi oluşturmayı üretimde oluşacak yüksek varyasyonu engellemek amacıyla kullanmaktadır. Monte Carlo simülasyonu ve tavlama simülasyonu kullanarak geliştirdiği yazılımın temel görevini artan kalite maliyetleri ile muayene ve tamir maliyetlerini dengelemek olarak tanımlamaktadır (Chen, 1997).

Viswanadham ve diğerlerinin (1996) kullandığı “bir muayene noktasında izin verilecek en fazla muayene sayısı” genelde dikkate alınmamış ilginç bir noktadır. Duffua ve Al – Najjar (1997) de bu noktaya dikkat çekmekte ve bu varsayımı yumuşatmaktadırlar. Önceki çalışmayı bir adım ileri götüren Duffua ve Al – Najjar (1997), ideal muayene tekrar sayısının teste tabi özellikte gözlenen hata oranları ve muayene maliyetinin bir fonksiyonu olduğunu ileri sürmektedirler. Muayene maliyetleri, hatalı kabul ve hatalı ret maliyetlerinin toplamından oluşan amaç fonksiyonlarını minimize etmek için de bir algoritma geliştirmişlerdir (Duffuaa ve Al – Najjar, 1997). Fakat çalışmaları bir ürün üzerindeki teste tabi tüm bileşenleri değil sadece kritik bileşenleri kapsamaktadır. Burada kritik bileşenden kast edilen, hatalı olmaları felakete,

ciddi tehlikeye ya da çok yüksek maliyetlere neden olan bileşenler şeklindedir (Duffuaa ve Al – Najjar, 1997).

Hatalı muayene olasılığı varken hataya müsait sistemlerin optimal muayene stratejileri Herer ve Raz tarafından incelenmiştir (1997). Chelbi ve Ait-Kadi (1999) tarafından yapılan bir çalışmada ise amaç cihazların rastsal olarak bozulma sıklığı ve sistemin atıl kalma süresini en aza çekmektir. Bu amaçla sistemin muayene edilme sıklıklarının belirleneceği bir muayene stratejisi bulunmaya çalışılmaktadır. Benzer fakat daha ilintili ve kapsamlı bir çalışmada Raz ve diğerleri (2000), maliyeti minimumda tutmak için, rastsal olarak arıza yapan bir makinede üretilen ürünlerin hangi bileşenlerinin hangi sırada test edilmesi gerektiğini incelemektedirler. Bunu yaparken önem verdikleri üç nokta, maliyeti minimumda tutmak, her bir ürünün kalitesinin bilinmesi ve testi geçen her ürünün kesin olarak hatasız olduğunun bilinmesi şeklindedir. Buldukları optimal muayene stratejisinin aynı zamanda üretimde kullanılacak öbek büyüklüğünü optimize etmek amacıyla da kullanılmasının mümkün olduğu da gösterilmektedir (Raz ve diğerleri, 2000).

Yine öbek büyüklüğü ve muayene stratejisini bir arada irdeleyen başka çalışmalar da bulunmaktadır. Örneğin Yeh ve Chen (2006) ürünlerin minimal tamir garantisiyle satıldığı ve gittikçe kötüye giden sistemlerde üretildiği durumlar için öbek büyüklüğü ve muayene stratejisini bir arada optimize etmişlerdir. Yeh ve Chen (2006) geliştirdikleri bir algoritma ile birim zamandaki maliyeti minimize eden bir optimal öbek büyüklüğü ve stratejinin bulunduğunu göstermektedirler. Van Volsen ve diğerleri ise çok aşamalı bir üretim sisteminde muayene yerleri, sayıları ve örneklem hacmi, örneklem alma sıklığı ve kabul limitlerini bir arada optimize ettikleri bir çalışma sunmuşlardır (2007). Bu çalışmada evrimsel algoritmaların içine yerleştirdikleri bir kesikli durum simülasyonu (DES, *Discrete Event Simulation*) ile optimal çözüme ulaşmışlardır.

Emmons ve Rabinowitz (2002) kalite kontrol faaliyetlerinin kısa ve uzun dönem olarak ayrı kararlar gerektireceğini belirtmektedirler. Bir firmada bulunan

muayene faaliyetlerinin ayrı birimlere bölüştürülmesi ya da muayene çizelgeleme kısa dönem kararlara örnek teşkil ederken, toplam muayene oranı ve muayene kapasitesi gibi kararlar uzun dönem kararlara örnek teşkil edecektir (Emmons ve Rabinowitz, 2002). Onların bilimsel katkısı tüm kapasitelerde sistem performansını tahmin etmeye müsaade eden bir muayene dağıtım ve atama metodu önermeleridir. Böylece daha gerçekçi problemlerin çözümü için bir teorik altyapı ortaya koymaktadırlar.

Çok aşamalı üretim için muayene planlaması ise Kakade ve diğerleri (2004) tarafından incelenmiştir. Kakade ve diğerleri (2004) muayene miktarını etkileyen faktörleri de tanımlamaktadırlar. Onlara göre muayene miktarı, ürün fiyatı, ürünün maliyeti, müşteri beklentilerinin karşılanamamasından kaynaklanacak ceza maliyetleri, tamir maliyetleri ve bozuk ürün oranına bağlıdır. Çalışma, çıktı ürün oranı ve muayene isabetliliği arasındaki ilişkiyi incelemektedir ve tavlama simülasyonu ile çözüm önerilmektedir. Sonuçları ise pratik olarak göz kararı yapılmış çözümlerden çok daha iyi sonuç vermektedir. Belirttikleri üzere küçük problemlerde tavlama simülasyonu optimale yakın stratejiler sunmakta, büyük problemlerde ise optimal çözüm bilinemediği için karşılaştırma imkanı bulunmamaktadır.

Muayene içerikli çalışmalardan da görüleceği gibi, bu alandaki çalışmalar çoğu test çizelgeleme ve test üretimi çalışmalarının aksine, amaç fonksiyonu olarak test süresinin minimize edilmesini değil doğrudan maliyetin minimize edilmesini kullanmaktadırlar. Test literatüründe çok sayıda maliyeti tanımlamaya ve / veya analiz etmeye ve / veya optimize etmeye yönelik çalışma da bulunmaktadır.

2.3. MALİYET TEMELLİ ÇALIŞMALAR

Test maliyeti ile ilgili en önemli erken dönem çalışmalarından biri yine Dislis ve ekibine aittir (Dislis ve diğerleri, 1989a). Çalışmada belirtildiği üzere entegre devrelerin boyut olarak büyümeye ve yapı olarak karmaşılaşmaya başlaması,

test maliyetlerinin toplam ürün maliyeti içinde çok önemli bir bileşeni haline getirmektedir. Dislis ve diğerleri, bu maliyetleri kaliteden ödün vermeden en aza çekmenin yolunun üretim başlamadan teste yönelik ürün tasarımından geçtiğini ve ürünün maliyetini belirleyebilmek için üründen bağımsız olarak ürün tasarımı, üretim ve test maliyetlerini etkileyen faktörlerin incelenmesi gerektiğini savunmaktadırlar (1989). Geliştirdikleri test stratejisi denetleyicisi ise test metodu incelemesini ürün hayat döngüsünde gerçekleşecek tüm bu maliyetlere dayanarak yapmaktadır.

Dislis ve diğerlerinin (1989a) kullandıkları maliyet modelleri incelenecek olursa, test ile ilgili maliyetlerin iki grupta incelendiği görülecektir. Bunlar ürün tipi ve bileşenlerine dayanan test üretimi maliyetleri ve test uygulama maliyetleridir. Test uygulama maliyetleri otomatik test cihazlarının (ATE, *Automatic Test Equipment*) çalışma maliyetleri, test programlama ve uygulama zamanları, işgücü, testte kullanılan parti büyüklüğü gibi uygulamalardan kaynaklanmaktadır. Genel bir ifadeyle çalışmada sabit ve değişken maliyetlerin kullanıldığı açıktır. Çalışma test stratejilerinin bu maliyet modellerini kullanarak test stratejilerini karşılaştırması ile tamamlanmaktadır (Dislis ve diğerleri, 1989a). Brunel Üniversitesi bünyesinde test alanında araştırmalar yapan ekip daha sonraki yıllarda bu çalışmayı geliştirmiş ve sadece maliyeti yeterli görmeyerek kalite düzeyi ve tamir ile ilgili faktörleri (tamir tekrarı sayısı ve tamir kalitesi gibi) de modellerine dahil ederek daha kapsamlı bir test denetleme modeli ortaya koymuşlardır (Dislis ve diğerleri, 1997).

Bu noktada test maliyeti ile ilgili çalışmalara bir parantez açarak, test stratejilerinin karşılaştırılması ile ilgili çalışmalara bir göz atmak doğru olacaktır. Zira Dislis, Dear ve Ambler'ın baskın olduğu test ile ilgili faktörleri araştıran Brunel Üniversitesi'ndeki kalabalık bir akademisyen grubu gibi, test stratejilerinin karşılaştırılması konusunu detaylı inceleyen bir akademisyen grubu daha bulunmaktadır. Volkerink, Khoche ve Rivoir'in başını çektiği bu grup test ekonomisi üzerine eğilmektedirler.

2.3.1. Maliyet Temelli Test Stratejisi Karşılaştırmaları

Dislis ve diğerlerinin entegre devrelerin büyümesi, hızlanması ve karmaşılaşmasından kaynaklanan kalite ve maliyet kaygılarına tamamen katıldıkları görülen Volkerink ve diğerleri (2002, 2003), endüstrinin halen test için harcanacak paranın nerede ve ne zaman kullanılması gerektiği konusunda çelişkide olduğunu öne sürmektedirler (Volkerink ve diğerleri, 2001, 2002 ve 2003). Volkerink ve diğerleri (2001) bu soruya cevap arayan bir ekonomik modelleme yaklaşımının sıfırdan yeni modeller kurmak yerine mevcut ekonomik model ve ilintili verileri kullanma yönünde kaldıraç görevi görebileceğini önermektedir. Böyle bir modelin ürün tasarımından piyasaya sürüme kadar tüm ciddi maliyet kalemlerini içermesi, kullanıcı dostu bir ara yüze sahip olması, detaylı modelleme ve olası içerik genişlemelerine müsaade etmesi, mevcut piyasa ve teknoloji verilerini kullanabilmesi, parametre tahmini ve trend değişimlerini öngörebilmesi gerekmektedir (Volkerink ve diğerleri, 2001). Bu konuda yazarların ortaya koydukları yöntemin prensipleri ve teknik detayları çalışmada belirtilmektedir.

Takip eden çalışmalarında Volkerink ve diğerleri (2002), test maliyetlerini detaylı olarak incelemektedirler. Fakat sadece test ekipmanı üzerinde geçen zamana odaklanan maliyet tanımları aynı zamanda test plakası ve problemleri gibi testin fiziksel teknik içeriğine odaklanmaktadır. Başka bir çalışmada modern test tekniklerini içeren bir test portföyü sunan Volkerink ve diğerleri (2003), bu tekniklerin optimal düzenlenmesinin uygulama alanı ve teknolojik gelişime dayandığını ve saptanması için teknik maliyet modelleme yönteminin (*Technical Cost Modeling*) son derece önemli bir bileşen olduğunu savunmaktadırlar.

Literatürde test stratejilerinin karşılaştırılması, oldukça yaygın işlenmiş bir konudur. Özellikle paralel test ile ucuz test ekipmanı kullanımı arasında seçim yapma üzerine oldukça fazla çalışma bulunmaktadır. Rivoir (2003 ve 2004), ve Zang ve Ito (2006), Dislis ve ekibinin çalışmalarına ek olarak örnek olarak gösterilebilir. Rivoir, konu test maliyetlerine geldiğinde en yaygın iki yöntemin

eski teknolojili ve düşük maliyetli test ekipmanı kullanmak ve paralel test yoluna gitmek olduğunu söylemektedir (Rivoir, 2004). Paralel testin iki tipi bulunmaktadır. Bunlardan ilki aynı anda çok sayıda ürünün test edilmesini ifade eden Çok-Yerli Test (*Multi – Site Testing*) olup, ikincisi ise bir ürünün üzerindeki teste tabi birden fazla özelliğin aynı anda test edilmesini ifade eden Eşzamanlı Test (*Concurrent Testing*) olarak tanımlanabilir. Rivoir'e göre (2004), endüstri Çok – Yerli Test stratejisinin ekipman maliyetini, test süresini ve test tekrarını artıracığını öne sürmekte ve ucuz ekipman stratejisine yönelmektedir. Rivoir doğru olduğunu söylediği tüm bu kaygılara rağmen, sadece ekipman maliyetini değil, test maliyetini artıran tüm bileşenlerin maliyetlerini de ciddi ölçüde azalttığı için paralel test stratejisinin tercih edilmesi gerektiğini söylemektedir (Rivoir, 2003 ve 2004). Hesaplamalar için Rivoir sermaye maliyeti, test faaliyet maliyeti, prob (sonda cihazı) maliyeti ve kaçış maliyeti toplamından oluşan aşağıdaki maliyet formülünü kullanmaktadır:

$$C = C_{CAP} + C_{OP} + C_{PC} + C_{PKG}$$

Düşük maliyetli ATE ile paralel testin maliyet açısından karşılaştırıldığı bir çalışma da (Zang ve Ito, 2006) literatürde bulunan elektrik – elektronik mühendisliği içerikli teknik test karşılaştırmalarına sadece bir örnek olarak not düşülebilir. Bu çalışmanın içeriği ile ilintili olmadığı için bahsedilen alanda mevcut çalışmalara girilmeyecektir.

2.3.2. Maliyet Modelleri Üzerine Kurulu Test Stratejisi

Test maliyetlerine geri dönüldüğü zaman yine Dislis ve diğerleri (1993a ve 1993c) tarafından yapılmış iki çalışma ile karşılaşılacaktır. Ekonomik analiz ve piyasa analizlerinin her türlü firma tarafından aktif olarak kullanıldığını ve bunun test ile ilgili kararlar için de geçerli olması gerektiğini öneren Dislis ve diğerleri (1993a) esas sorunun test kararlarının gerçek maliyetlerini tahmin etmenin her zaman kolay olmadığını belirtmektedirler.

Dislis ve diğerleri (1993c), karmaşık VLSI sistemlerinin testinde kalite, zaman, maliyet, test cihazlarının elverişliliği ve tasarım tercihinin önemli faktörler olduğunu ve bu yüzden test strateji alternatiflerinin ürün geliştirilme süreci bitmeden değerlendirilmesi gerektiğini savunmaktadırlar. Bu da hem maliyet hem de kalite tahmini yapabilecek bir model gerektirmektedir. Modelin test cihazları, test ve tamir döngüleri, işgücü, malzeme, test üretimi, kalite düzeyi (hata kapsama oranı), zaman kullanımı ve benzeri faktörlerin maliyetlerini hesaplayabilmesi verimli bir ürün ömrü (tasarım, üretim, test ve piyasaya sunum) için hayatidir. Fakat Dislis ve diğerleri (1993c), verilerde ve girdilerde görülmesi beklenebilecek belirsizliğin modelin vereceği sonuçları etkileyeceğini ve bunun önemli bir sorun olabileceğini öne sürmektedirler. Dick, Dislis, Ambler ve Trischler (1994) yine test maliyet modellerine ilişkin bir başka çalışmalarında test maliyet değişkenleri üzerine bir hassasiyet analizi yapmış ve birim üretim maliyeti, üretim hacmi, hata kapsamı gibi parametreleri içeren beş değişkenin tahmin doğruluğunu doğrudan etkileyen en önemli parametreler olduğunu bulmuşlardır.

Pynn (1997) ise başka bir noktaya dikkat çekmektedir. Birçok iş fiyatlandırma formülünde dikkate alınan maliyetler, genellikle parça, iş gücü, ekipman gibi somut maliyet kalemleriyle tanımlanmaktadır. Oysa devre, elektronik panel, kart gibi ürünler söz konusu olduğunda, maliyetin hatalı tanımı iş tahmini maliyeti tahminleri ile gerçek üretim maliyetleri arasında fark oluşmasına neden olmaktadır. Örneğin test maliyetleri düşük hatasız ürün oranlarına sahip bir üretim sisteminde normalden çok daha yüksek olacaktır ve maliyet tahminleri yapılırken bu tip hassas noktalar özenli bir çalışma gerektirmektedir. Genel geçer test maliyeti modelleri maliyetleri ekipman, uygulama ve kaçış (hatalı ürün) maliyetleri çerçevesinde incelemekte ve örneğin test süresini sadece bir ürünün testte geçirdiği süre olarak kabul edip, hatalı ürünlerin tekrarlı testlerini göz ardı edebilmektedir. Pynn böylesi hataların önüne geçebilmek için normalden çok daha fazla değişken içeren bir test maliyet analizi (TCA, *Test Cost Analysis*) modeli geliştirmiştir (1997).

Scheffler ve diğerleri (1998) maliyet modellemenin amaçlarını ürünün nihai maliyetinin tahmin edilmesi, maliyet yapısının anlaşılması ve hesaplı ideal bir üretim çözümü önermesi olarak özetlemektedirler. Bunun için de malzeme ve parça maliyetleri, işgücü (tasarım, üretim ve test üretimi) maliyetleri, ekipman maliyetleri ve amortismanları ve çıktı ile testi ilgilendiren faktörlerin bilinmesi gerektiğini söylemektedirler ve en önemli amaç olarak da ideal çözümün bulunmasını yani optimizasyonu göstermektedirler. Yöntem olarak da iki alternatif bulunduğunu ve bunların ya parametreleri değiştirerek çözümü incelemekten ya da alternatif senaryoları karşılaştırarak sonuçlara bakmaktan geçtiğini savunmaktadırlar. Maliyeti ise üç başlık altında incelemektedirler:

- Tekrarlamayan maliyetler (TM): Ekipman ve tasarım maliyetleri
- Doğrudan maliyetler (DM): Parçalar, malzemeler, işgücü, test ve tamir maliyetleri
- Kaçış maliyetleri (KM): Çıktı oranı ve hata kapsamı oranına bağlı hurda ve ret maliyetleri

Bu açılımla Scheffler ve diğerleri (1998), nihai maliyet ve nihai kaliteyi formül olarak aşağıdaki gibi ifade etmektedirler. Formüller göstermektedir ki aşırı test kaliteyi artırmakta fakat çok yüksek birim maliyetlere de yol açmaktadır.

$$Nihai_Maliyet = \frac{\sum DM_{Birim} + \sum KM + \sum TM}{Sevk_edilen_ürün_sayısı}$$

$$Nihai_Kalite = \frac{Hatasız_sevk_edilen_ürün_sayısı}{Sevk_edilen_ürün_sayısı}$$

Chen ve diğerleri (1999b) ise tasarım döngüsünün kısalabilmesi, bütçenin aşılmaması ve maliyetlerin azaltılabilmesi için doğru test maliyet tahminlerini temel alan bir tasarım yöntemi belirlenmesi gerektiğini ve bunun için de test maliyetlerini tahmin edecek bir model gerektiğini söylemektedirler. Asıl önemli

olanın henüz tasarım aşamasındayken bu tahminlerin yapılması olduğunu öneren Chen ve diğerleri, böyle erken bir aşamada tahmin edilmesi mümkün olabilecek basit parametrelerle (test süreleri ve test maliyet oranları) yola çıkmaktadırlar. ASIC üretimi konu edilen model tahmin yapmak için hata kapsama oranı, hassas bölgelerin (ürün üzerinde hataya müsait kısımlar, kritik alanlar olarak da kullanılmaktadır) oranları ve çıktı oranları gibi faktörleri de içermektedir (Chen ve diğerleri, 1999b). Chen ve diğerlerinin (1999b) öne sürdüğü bu fikir yeni değildir. Örneğin Pattipati ve Alexandiridis (1990), bir hatanın ürünün çip bileşeni henüz monte edilmeden tespit edilmesi \$ 0.30 tutacaksa, monte edilip kart haline geldiğinde aynı hatanın tespit maliyetinin \$ 3 olacağını belirtmektedirler. Sistem düzeyinde bu maliyet \$ 30 haline gelecek ve müşteriye yollandıktan sonra oluşacak bir hatada ise maliyet \$ 300'e çıkacaktır (Pattipati ve Alexandiridis, 1990).

Test alanındaki çalışmalarda test maliyeti modelleri ile sıklıkla karşılaşılmaktadır. Bir diğer çalışmada Evans (1999), test maliyetini azaltma ve çıktı verimini artırma amaçlı bir maliyet modeli kurgulamaktadır. Model birim test maliyetini oluşturan tüm öğeleri (örneğin, test süresi, çıktı oranı, ekipman verimi, ATE amortismanı, işgücü, kapasite vb maliyetler) içermekte ve optimizasyon amacı dışında bu öğelerin karşılaştırmalarını modelin yapısından gelen karmaşıklıkla elimine ederek yapabilmektedir. Model ayrıca elle ya da genel kullanılır tablo tabanlı programlarla (örneğin Excel) senaryo karşılaştırmalarına imkan vererek, CPU (*Cost per Unit*, Birim başına maliyet), CPSUTIL (*Cost per Utilized Second*, Kullanılan saniye başına maliyet) ve UPM (*Good Units per Month Tested*, Aylık test edilen hatasız ürün sayısı) gibi endüstride ortak kullanılan değerleri de hesaplayabilmektedir. Bunu ise başa baş noktası, çıktı, atıl süre ve maliyet analizleriyle gerçekleştirmektedir. Ferrario ve diğerleri (2003) de benzer bir çözümü kablosuz telefon çipleri için aramaktadırlar.

2.3.3. Test Maliyetlerini Etkileyen Faktörler

Oppermann ve diğerleri (2001 ve 2003b) istatistiksel proses kontrolü (SPC) ile harmanladıkları kalite maliyet modelini proses kontrolü ve muayene stratejisi saptama (örneğin SPC ile %100 muayene arasında seçim yapma) gibi problemlere çözüm önerme amacıyla geliştirmişlerdir. Oppermann ve diğerleri (2001 ve 2003b) çalışmalarında teknolojik sürecin hemen arkasında muayene süreci bulunan ve bulunmayan durumlar için maliyet formülleri önermektedirler. Teknolojik bir süreç (üretim süreci) T ile tanımlanırsa peşi sıra muayene istasyonu bulunmayan T süreci için hesaplanacak maliyet k şöyle gerçekleşecektir:

$$k = kT + p * kF$$

Muayene bulunması durumunda ise oluşacak maliyet şöyle olacaktır:

$$k = kT + kp + p * kR$$

Bu formüllerde kT birim başına üretim maliyeti, p hata oranı, kF hatalı ürün maliyeti, kp muayene maliyeti ve kR tamir maliyeti anlamına gelmektedir. İlk formül, muayene olmamasından dolayı, sadece üretim ve beklenen hatalı ürün için hata sonrası oluşacak maliyetleri, ikinci formül ise üretim, muayene ve beklenen hatalı ürün için tamir maliyetlerini içermektedir. Oppermann ve diğerleri (2003a) kalite maliyetlerinin optimizasyonu için sadece iki alternatif strateji söz konusu ise maliyet hesabı yoluyla karşılaştırma, ikiden fazla alternatif strateji arasında optimal olanı seçmek için dinamik programlama önermektedirler. Bu maliyet hesaplarının Rivoir (2003) tarafından kullanılan maliyet hesaplarına oldukça benzediği görülebilir.

Test maliyetlerini tanımlayan Kapur ve diğerleri (2001) ise birçok çalışmada görüldüğü gibi sınıflandırmayı tasarım, test ve sermaye maliyetleri şeklinde yapmaktadırlar. Düşük maliyetli test için gerekli öğeleri inceleyen Kapur ve

diğerleri, tahmin edilebilir test, kalite oranı çok yüksek olması beklenmeyen parçaların testi için eski teknoloji ekipman kullanımı ve etkin test ile test maliyetlerinin ciddi oranda düşürülebileceğini önermektedir.

Williams ve Ambler (2002), yine test maliyet modelleri üzerine çalışmış ve test sürecine giren sistemler için bir test maliyet modeli önermişlerdir. Maliyet modeli firmalara üretim ve geliştirme süreçlerini de göz önüne alacak şekilde test süreçlerini ürün risklerine uygun şekilde düzeltme imkanı sunmaktadır. Çünkü test süreci de işletmelerdeki diğer süreçler gibi maliyet yönünden etkili olmalıdır. Bu durum basitçe iki şekilde sonuçlanabilir:

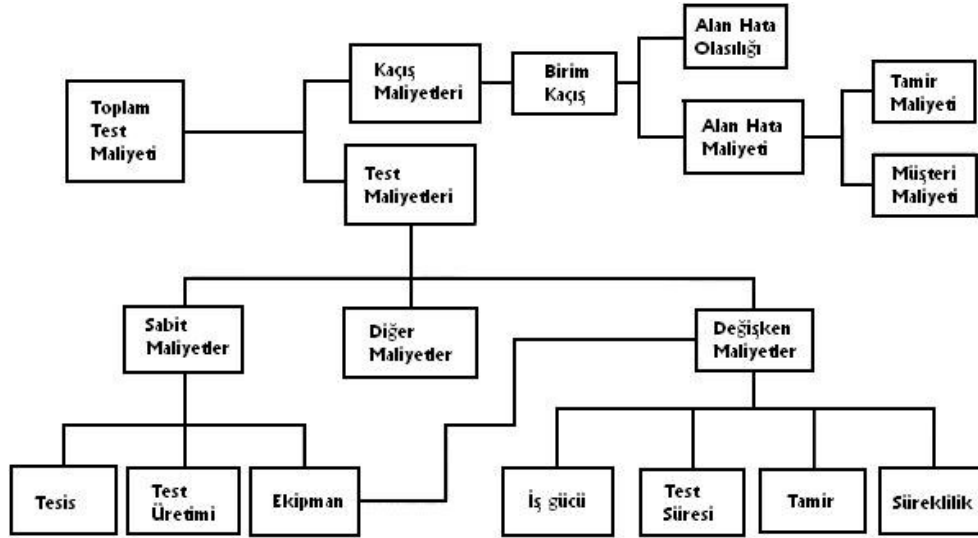
- Firma çok az test yaparak piyasada kötü kaliteli ürün üretiliyor damgasını yiyebilir
- Firma çok fazla test yaparak ürünün piyasaya ulaşma süresini geciktirerek fırsat maliyetlerine ve yüksek maliyetli tesislere katlanmaktadır

Williams ve Ambler'in (2002) önerdikleri maliyet hesabı ise Oppermann ve diğerlerinin (2001) önerdikleri ilk formülle aynıdır:

$$C_{OMT} = C_{MT} + C_{TES}$$

Yani genel test maliyeti C_{OMT} (*Cost of Overall Manufacturing Tests*), test maliyeti C_{MT} (*Cost of Manufacturing Tests*) ve kaçış maliyetleri yani saptanamayan hatalı ürün maliyetleri C_{TES} (*Cost of Test Escapes*) toplamına eşittir. Williams ve Ambler test maliyetinin şema gösterimini de sunmaktadırlar (Şekil X).

Şekil 1. Test Maliyetleri Şeması



WILLIAMS, D., ve A. P. AMBLER. "System Manufacturing Test Cost Model." *Proceedings of the IEEE ITC International Test Conference*, 2002: 482 – 490.

Williams ve Ambler'e (2002) ek olarak Ambler'in (2001) verdiği test maliyeti sınıflandırması da test sürecinde karşılaşılan maliyetlerin algılanmasında son derece önemli rol oynayacaktır. Ambler, test maliyetinin önemsenmemesinin imkansız olduğunu ve maliyetin detaylı analizinin katkısı yadsınamayacak bir kar kapısı olduğunu belirtmektedir. Piyasaya çıkış zamanı, kalite algısı, piyasa ve ürün ömrü gibi faktörlerin önemine de değinen Ambler (2001), yine de ekonomik analizin tek başına her soruna çare olamasa da karar verme sürecinde önemli bir yardımcı rolü oynadığına dikkat çekmektedir. Bu açılımla tanımladığı maliyet faktörleri şu şekildedir:

- Tasarım ve Test:

- Tasarım, Test için Tasarım (DFT, *Design For Test*) ve test süresi
- Bütçe
- Piyasa dengeleri
- Kalite ve güvenilirlik
- Hedef çıktı
- Eğitim
- Paketleme teknolojisi
- Ürün tasarımı ve silikon maliyetleri

- Bilgisayar ve tasarım (CAD, *Computer Aided Design*) maliyetleri
- Sermaye yatırımı
- Test İçin Tasarım (DFT) ve Otomatik Test Ekipmanı (ATE)
 - Satın alma, amortisman ve bakım
 - Operatör ve programlama maliyetleri
 - Hazırlama (*Setup*) ve test uygulama zamanı
 - İşgücü fiyatları
 - Hata kapsamı
 - Program hazırlama, test ve teşhis süreleri
 - Üretim hacmi ve çıktı oranı
 - Ürün yelpazesi
- Alan Hizmet Maliyetleri (Satış Sonrası Maliyetler)
 - Tamir yerleri
 - Lojistik destek
 - Alanda tamir maliyetleri
 - Atıl ve sistem arızası zamanlarının maliyetleri
 - Tekrar yoklamaların azaltılması
 - Bakım için tasarım

Yukarıdaki maliyet listesi, mevcut çalışmalar içinde verilmiş en kapsamlı maliyet kalemlerini içermektedir. Literatürdeki maliyet modelleri ve maliyet odaklı çalışmalar genellikle bu maliyetlerin bir alt kümesini kapsayacak şekilde geliştirilmişlerdir.

Maliyet modelleri üzerine çalışan Reinosa ve Michel de ikisi burada incelenecek çalışmalarla (Reinosa ve Michel, 2003 ve Michel ve Reinosa, 2004) bu alana katkıda bulunmuşlardır. NEMI (*National Electronics Manufacturing Initiative*, Milli Elektronik Ürünler Üretim Teşebbüsü) kapsamında yürütülen Test Stratejisi Projesi'nde (*Test Strategy Project*) araştırma yapan ekip, üreticilerin belirli bir test stratejisi seçiminin finansal etkilerine ilişkin ölçümleri yapabilmesi, finansal referans oluşturmak, test stratejileri arasındaki farkları görebilmek ve sevk

sonrası oluşacak hataları ve garanti unsurlarını gözleyebilmek gibi amaçlara hizmet etmek üzere hazırlanmış bir test maliyet modelini betimlemektedirler (Reinosa ve Michel,2003).

Model, dokuz aşamalı olacak şekilde iki test stratejisinin karşılaştırmasını, çıktı oranı ve milyon birim başına hata sayısı (DPMO, *Defect per Million Opportunity*) temelli performans karşılaştırması, piyasaya sürüm zamanı (*Time – to – Market*) ile ilgili tasarrufları değerlendirmektedir. Modelin maliyet ilintili bazı hesaplarına dair formüller şu şekildedir (Reinosa ve Michel, 2003 ve NEMI, 2003):

- Çıktı oranı: Çıktıyı olasılık yoğunluğu fonksiyonunun altında toleranslar arasında kalan alan oranı olarak tanımlayan çalışma, Poisson dağılımından gelen çıktılar öngörmektedir.
- DPMO: Kullanılma amacı bazı firmaların sadece süreç sonu hata oranına odaklanmalarıdır.
- Test kapsama oranı: Test etkililiği ile test erişimi çarpımı olarak hesaplanmıştır.

$$T_c = T_e * T_a$$

- Hatalı ürün sayısı: Test kapsamı ile hata sayısı çarpımı olarak hesaplanmıştır.

$$D_f = T_c * D$$

- Kaçış sayısı: Hata sayısı ile testte kapsanmayan kısımların oranı çarpımı olarak hesaplanmıştır.

$$D_o = D_b * (1 - T_c)$$

- Test etkililiği: Teste giren hatalı ürünlerden test sonu yakalanan kısmın oranı olarak hesaplanmıştır.

$$T_e = \frac{D_i - D_o}{D_i}$$

Michel ve Reinosa (2004) ise modelin sınırlılıklarına da değinmektedir. Buna göre, modelin kapsadığı ürün tipleri endüstrideki tüm ürün tiplerini kapsamamaktadır. Ayrıca model her test ile ilgili test kapsamını çok aşamalı

test gibi kabul ederek üst üste binecek şekilde hesaplamakta, bu durum da birden fazla testin birbirini tamamlama amacıyla bir arada kullanılması durumunda hatalı test kapsama oranına yol açmaktadır. Örneğin ilk test %60'lık hata grubunu, ikinci test ise kalan %40'lık hata grubunu tamamen kapsarken, model her testi ayrı değerlendirdiği için %100'lük bir kapsama oranını saptayamamaktadır. Son olarak model mükemmel test durumunu varsaymakta ve hatalı kabul ve hatalı ret gibi durumları dikkate alamamaktadır.

Literatürde bu çalışmadakine benzer test ve tamir durumunu bir arada ele alan çok fazla sayıda çalışma bulunmamaktadır. Detaylarına sonraki bölümlerde ayrıntılı olarak girilen Goyal ve Mosher (2006), Rossi ve diğerleri (2006), Fisher ve diğerleri (2007a ve 2007b) içeri olarak bu çalışmaya en yakın araştırma alanına sahiptirler. Zaten bu çalışmaların içerikleri, teorik kısmın ifade edilmesi sırasında ayrıntılı olarak ele alınacaktır. Bu çalışmalar hariç, bir miktar yayının yine de incelenmesi doğru olacaktır.

2.3.4. Hata Faktörü İçeren Çalışmalar

Ding ve diğerleri (1998) karar hatalarının gerçekleştiği bir test durumunu ele almaktadırlar. Sonraki bölümlerde açıklanacak olan test sonucunun ürünün özelliğini yansıtmadığı durumlar geçerli iken tekrarlamalı testi, iki test stratejisinin karşılaştırılmasını, test ekipmanının tutarlılık ve kapasiteleri, girdilerin kalitesi ve çıktılardan beklenen kalite düzeyinin bunlara olan etkisini incelemektedirler. Tekrarlamalı testte hatalı ürünlerin tekrarlı testi daha ekonomik bir yöntemdir. Ding ve diğerlerinin verdikleri örneğe bakılırsa, %95'i kaliteli bir girdi kümesi %90 olasılıkla doğru ve kalitesiz ürünler %1 olasılıkla hatalı test edilecek olsa, $(0.90) \cdot (95)$ ve $(0.01) \cdot (5)$ toplamından yaklaşık 86 ürün testi geçecek ve 14 ürün hatalı oldukları şüphesiyle ayrılacaktır. Bu durumda 86 yerine 14 ürünün test edilmesi ekonomik strateji olacaktır. Ding ve diğerleri (1998) gerçekten bu yaklaşımın ekonomik strateji olup olmadığını kontrol etmekte ve bunun için belirli bir çıktı oranını matematiksel programlama

modeline şans kısıtı olarak katmaktadırlar. Modelleri hurda oranı ve test maliyetlerini makine kısıtları altında minimize etmeye çalışmaktadır. Modelin sonuçları aynı zamanda optimal strateji ve test tekrarı sayısını da ortaya koymaktadır. Bulguları ise, kabul edilen ürünlerin tekrarlı testinin ancak test isabetliliği zayıfken, girdi kalitesi kötüyken ya da çıktıların çok düşük bir hata oranına sahip olması gerekliken doğru strateji olabileceğini, aksi takdirde hatalı ürünlerin test edilmesinin tercih edilmesi gerektiğini ortaya koymaktadır (Ding ve diğerleri, 1998).

Trichy ve diğerleri (2001) ise bir test – teşhis – tamir ağı oluşturmaya çalışmaktadırlar. Onlara göre elektronik ürün tasarım temelindeki en büyük tasarruf kapısı maliyettir. Yazarlar bir teknik maliyet modelleme (TCM, *Technical Cost Modeling*) yaklaşımı kullanmaktadırlar. Bu açıdan çalışma, önceki alt bölümlerde incelenen maliyet modelleri literatürünün de bir parçasıdır. TCM, bileşenlerin ve bileşenler ya da ilgili teknoloji ve üretim sistemleri değiştiğinde maliyetin nasıl etkileneceğini incelemekte ve bunun için en detaylı kalemlere kadar maliyeti ayrıştırarak bir nevi hassasiyet analizi yapmaktadır. Model, tamir ve testleri, hata kapsamasını ve test ile tamirde oluşacak karar hatalarını da içeren bir eklenti ile olası potansiyelleri saptayabilecek bir analiz aracı haline getirilmektedir (Trichy ve diğerleri, 2001).

Shi ve Sandborn (2003) ise aynı problemi ele alarak ağ üzerindeki test – teşhis – tamir noktalarının yer ve tiplerini optimize etmeye çalışmaktadırlar. Optimizasyon imkanı veren parametrelerden en önemlileri hata kapsama ile test maliyeti, teşhis başarı oranı ile teşhis maliyeti ve tamir başarı oranı ile tamir maliyeti olup bu ikililer arasında tasarruf imkanı oluşmaktadır (Shi ve Sandborn, 2003). Yazarlar hedef olarak da hata kapsama oranı, test birimlerinin konumu, tamir işleminin ne zaman yapılması ve kaç tamir girişiminde bulunulması gibi kararlar belirlemişlerdir. Çözüm için bekleme sırası araması (WSS, *Waiting Sequence Search*) isimli yeni bir arama algoritması ile gerçek kodlu genetik algoritmaları (RCGA, *Real Coded Genetic Algorithms*) bir arada kullanmaktadırlar. Alfredsson ise (1997), yedek parça envanterleri ile ilgili bir

destek sistemini optimize etmeye çalıştığı araştırmasında yedek parça optimizasyonu, tamir düzeyi ve test ekipman seçim problemini eş zamanlı çözmeye çalışmaktadır. Burada test ve tamir noktalarını bulmaya çalışan Alfredsson problemi bir makine seçim optimizasyonuna çevirmektedir. Böylece bu çalışma ile ilintili bir diğer alan daha ortaya çıkmaktadır: ekipman optimizasyonu.

2.4. EKİPMAN OPTİMİZASYONU

Temel prensip olarak makinelerin kapasitesi ve performansı ile fiyatları arasında bir denge noktası bulunacağından bu noktada da bir optimizasyon imkanı söz konusudur. Çalışmaya özel olarak, test makinelerinin hatalı ve hatasız ürünleri doğru kararlar ile ayırabilmesi, tamir makinelerinin ise hatalı ürünleri hatasız ürünler halinde çıkarırken hatasız ürünleri bozmaması, aranan performans karakteristikleridir. Fakat literatürde bulunan ekipman optimizasyon çalışmaları Alfredsson (1997) örneğinde de görüldüğü gibi farklı tercihler ortaya koyabilir.

Jin ve diğerleri (2003) piyasa taleplerinin kalite ve güvenilirlik düzeyini artırdığını ve bunun da ekipman seçimine etki ettiğini belirtmekte ve belirli bir kalite düzeyine değişik ekipman bileşenleriyle girilebileceğinden, ekipman seçiminin doğal bir optimizasyon problemi olduğunu belirtmektedirler. Sistemin arızada olma durumunu minimize eden Jin ve diğerleri (2003) optimal bir ekipman kümesi bulmaya çalışmaktadırlar.

Oysa Camerana ve diğerleri (2004), çok çiftlikli bir sistem için traktör ve benzeri ekipmanı optimize etmeye çalışmaktadırlar. Uygulama açısından tamamen farklı bir alan olsa da prensip olarak oldukça ilintili sayılabilecek bu çalışma maliyeti minimize ederken, belirli makinelerin belirli tarlalarda çalışabilmesini (örneğin test kapsamı gibi), çiftliklerde gerekli çalışma süresi, makineler için minimum çalışma süreleri gibi yine genel optimizasyon kısıtlarından

faydalanmaktadır. Çözüm yöntemi ise karışık tamsayı doğrusal programlama (MILP, *Mixed Integer Linear Programming*) olarak belirlenmiştir.

2.5. DİĞER KONULAR

Çalışmanın bu noktasında test ile ilgili literatür tamamlanmış gibidir. Bir miktar çalışmaya ilerleyen bölümlerde yerleri geldikçe atıf yapılacaktır. Fakat test optimizasyonu gibi yeni bir alanın hangi alt çalışma alanlarını içerdiği, nasıl problemlerle ilgilendiği, nasıl bir evrim geçirdiği, nasıl tekniklerle problemlere çözüm aradığı gibi sorular yukarıdaki bilgilerde cevap bulabilecektir. Şüphesizdir ki bu alanda çok daha fazla çalışma bulunmaktadır. Örneğin tasarım, test için ve test edilebilirlik için tasarım, çok fazla çalışmanın verildiği bir diğer test ilintili alandır. Khoche ve diğerleri (2001), Ali ve diğerleri (2004) ve yukarıda incelenen çalışmaların bazıları bu alana örnek olmaktadır.

Çıktıların durumu ile ilgili apayrı bir literatür alanı söz konusudur. İleride görüleceği üzere bu çalışmada ürün oranı teste tabi tüm özellikleri hatasız olup testi geçmiş ürünlerin, testi geçmiş toplam ürünler içindeki yüzdesi ile ifade edilmektedir. Fakat konu ile ilgilenen araştırmacılar için Cox ve diğerleri (1983), Stapper ve diğerleri (1983), Ferris – Prabhu (1985), Maly ve diğerleri (1986), Stapper (1986), Das ve diğerleri (1991), Gaitonde ve Walker (1993), Stapper (1993), Tegethoff ve Chen (1994), Gaitonde ve diğerleri (1995), Stapper ve Rosner (1995), Heineken ve diğerleri (1996), Pleskacz ve diğerleri (1997), Chen ve diğerleri (1999a) ve Pleskacz (1999) gibi çalışmalar ayrıntılı ürün oranı tahminleme yöntemleri sunmaktadır. Ürün oranının tamamlayıcısı görevi gören alandaki hatalı ürün oranı (FRR) tahminlemeleri için de Seth ve Agrawal (1981), Williams ve Brown (1981), Sousa ve Texeira (1991), Das ve diğerleri (1993) ve Gaitonde ve diğerleri (1993) faydalı bilgiler sunmaktadır.

Muayenenin farklı bir yerde gerçekleştirilmesi (*offline inspection*) yine birçok çalışmaya konu olmaktadır (örneğin Barbagallo ve diğerleri, 1999, Finkelshtein

ve diğ erleri, 2005 ve Wang, 2007). Sepulveda ve Nachlas (1997) ve Govindaraju (2005) testin istatistiksel yanı ile ilgilenmiş, Agrawal (1978) ise rastsal test stratejisinin ne zaman tercih edilmesi gerektiğini incelemiştir. Willams (1986), Seth ve diğ erleri (1990) ve Thatcher (1993) test edilebilirlik kavramını inceleyen pek çok çalışmaya örnek teşkil ederken, test mühendislerinin eğitimleri ile ilgili çalışmalar bile yazılmıştır (Bertrand ve diğ erleri, 2003). Tüm bunlar göreceli olarak bu çalışmayla ilintili sayılabilecekken, Iyengar ve diğ erleri (2003), West (2003) ve Bose ve Agrawal (2006) gibi çalışmalar, test ve ilintili alanların tamamen elektrik – elektronik mühendisliği temelli teknik tarafına örnek teşkil etmekte ve bu çalışma ile tamamen ilgisiz olan fakat her sene binlerce çalışmanın yayınlandığı apayrı bir test araştırması alanını temsil etmektedir.

Görüldüğü gibi test optimizasyonu ve ilintili alanlar oldukça farklı disiplinlere uzanmaktadır. Bu bölümde, sadece takip eden bölümlerde karşılaşı lacak teorik ve pratik araştırmaların anlamlandırılmasında faydalı olacağı düşünölen alanlardaki çalışmalara yer verilmiştir. Takip eden bölümle birlikte çalışmanın esas amacı olan test stratejisi, test optimizasyonu, makine seçimi, test kapsaması, kalite seviyesi (ürün oranı) gibi kavramların detaylı incelenmesi ve bir arada matematiksel modellenme amacıyla kullanılmaları söz konusudur.

3. BÖLÜM

TEK ÖZELLİKLİ ÜRÜNLERİN TESTİ

3. GİRİŞ

Bu bölümde test kavramının bileşenleri incelenecektir. Bunlar test maliyetleri, test ile ilgili temel kavramlar, test akışının dinamikleri olarak sıralanabilir. Literatür bölümünde de belirtildiği üzere test maliyetleri, ödünleştirmeler yoluyla optimizasyon imkanı vereceğinden dolayı, bu çalışma için yüksek öneme sahiptir. Test ile ilgili temel bilgiler, test akışının yöneylem amacıyla matematiksel modellenmesi, temel kavramların ifade edilmesi, kesin hesaplar ve optimizasyon amaçları ayrıntılı olarak tanımlanacak ve ilgili örnek modeller çözülecek ve irdelenecektir.

Tüm bunlar yapılırken, örnek olması açısından aksi belirtilen yerler hariç ürünün tek özelliği olduğu ve test ile tamir makinelerinin de tek tip olduğu varsayılmaktadır. Böylece olabilecek en basit test düzeneği ile karşılaşılmaktadır. Basit test düzeneği kullanılarak temel kavramlar ve tanımlar aktarılacak ve daha karmaşık konular için gerekli hazırlık yapılmış olacaktır.

3.1. TEMEL TEST KAVRAMLARI

Temel test kavramlarının inceleneceği bu alt bölümde, test maliyetleri, test döngüsü ve akışı, test ve test tasarımı ile ilgili kavram ve terimler, test tasarımı ile ilgili kesin hesaplar, test tasarımında kullanılabilecek optimizasyon yöntemleri ve modelleri, bu modeller ile ilgili açıklamalar ve örnekler irdelenecektir.

3.1.1. Test Maliyetleri

Test maliyetleri deęişik kalemlerde incelenebilir. Bir önceki bölümde, dięer çalışmalar bir yana, Ambler (2001) ve Williams ve Ambler (2002) tarafından ayrıntılı maliyet kalemleri ve maliyetleri etkileyen faktörler anılmaktadır. Yine de maliyet incelenmesinde kullanılan genel kurallar çerçevesinde bakıldığında maliyetleri kabaca sabit ve deęişken maliyetler olarak iki başlık altında incelemek mümkündür.

3.1.1.1. Sabit Maliyetler

Sabit maliyetler belirli bir ürünün tasarımından piyasaya sunulmasına kadar gerçekleşen süreçte sadece bir kez karşılaşılan ve tekrar etmeyen maliyetler olarak tanımlanabilir. Açıkça görülmektedir ki üretim hacmi büyüdükçe bu maliyetler toplam maliyet içinde daha küçük bir yüzdeye sahip olacak, yani göreceli olarak önemleri azalacaktır.

Sabit maliyetler dahilinde ilk karşılaşılan öge makine yatırımı olacaktır. Üç önemli sabit maliyet nedeni, test, tamir ve hem test hem tamir yapabilen birleşik makinelerdir. Makine maliyetini etkileyen birçok faktör vardır. Makinelerin seçimi, sektör, firma büyüklüğü, üretim kapasitesi ve test içeriğine baęlı olarak deęişebilmekte olup, birkaç bin dolardan yüz binlerce dolara kadar deęişebilecek tasarruflara gebedir. Makinelerin yanı sıra test cihazları için geliştirilen yazılımlar da sabit maliyet sınıfına girebilecek olup, son derece önemli ve yüksek bir maliyet kalemi olarak deęer kazanmaktadır.

3.1.1.2. Değişken Maliyetler

Değişken maliyetler ise sabit maliyetlerden farklı olarak üretimin tüm süreçleri boyunca bazen faaliyete bağlı, bazen de faaliyetten bağımsız olarak oluşan maliyetlerdir. Bu maliyet tipi de kendi içinde farklı gruplara ayrılmaktadır. Bunların arasında işgücü, test ve tamir işlemlerinin aldığı süre, test ve tamir makinelerinin performansları ve kapasitelerinden kaynaklanacak maliyetler ve çıktılarından kaynaklanan maliyetler ve diğer maliyetler bulunmaktadır. Yıllık toplam olarak ele alındığı zaman, değişken maliyetler de sabit maliyetler ile başa baş noktasına birkaç sene içinde ulaşabilecektir.

İşgücü maliyetleri test operatörü, tamir operatörü, hata çözücü ve benzeri çalışanları kapsamaktadır. Diğer maliyetlerin içinde de, tamir yapılacak yerin test yapılan yer ile aynı olmaması durumunda karşılaşılabilecek paketleme, taşıma, tekrar paketleme ve geri taşıma, olası detaylı hata analizi ve benzeri beklenmeyen durumlar bulunabilecektir. Süre, çıktı ve makine performanslarından kaynaklanacak maliyetler ve bu kalemlerle ilgili tanım ve açıklamalar daha ayrıntılı bir inceleme gerektirmektedir.

3.1.1.3. Performans Maliyetleri

Test ve tamir makinelerinin performanslarının (yani verimlilik ve isabetli sonuç verebilme başarımlarının) sebep olacağı maliyetler bu çalışmada performans maliyetleri olarak tanımlanmıştır. Test ve tamir cihazlarının az ya da çok hatalı çalışmaları durumunda performans maliyetleri oluşmaktadır. Örneğin bir ürün testi geçemeyip tamir makinesine aktarıldığı zaman, üzerinde bulunan kusur gerçekte düzeltilememişse, aynı testten tekrar başarısız olacağı beklenebilir. Böyle bir durumda da ürün test sisteminde bir turu firmaya hiçbir katma değer sağlamadan atmış ve maliyetleri artırmış olacaktır. Bu durum tamir makinesi performansının yol açacağı maliyetlere bir örnektir.

Test makinesi performansında görülecek olası hatalar ise firmanın son kullanım için sevk edeceği bitmiş ürünlerdeki hata oranlarını etkileyecektir. Zira test makinesinin onay verdiği fakat gerçekte kusurlu olan bir ürün, en kötü ihtimalle son kullanılacağı yerde fark edilecektir. Böylesi durumlarda ürünün değiştirilmesi, yerinde tamiri, firmaya geri yollanarak firmada tamiri, lojistik ve ulaşım masrafları, firmanın piyasadaki imajının zarar görmesi nedeniyle oluşabilecek satış kayıpları gibi maliyetler oluşabilecektir.

Varsayımsal olarak Amerika'da faaliyet gösteren bir firmanın Çin'e sevk ettiği bir ürün, alan montajından sonra hata verdiği takdirde, bu durumun tazmini hem maddi hem de firma imajı açısından Amerikan firması için oldukça pahalıya mal olacaktır. Örneğin böyle bir durumda sadece hatalı ürünün TnT Lojistik kullanılarak Chicago'ya geri dönmesi ve tekrar Pekin'e yollanması 3,600 USD gibi bir maliyete sahiptir. Firma, Çin'e senede 100,000 adet ürün satıyor ve bunların %5'i hatalı çıkıyor ise, yüksek gibi görünen %95'lik başarılı ürün oranından doğan hatalı ürünlerin lojistik maliyeti 18,000,000 USD tutmaktadır. Bu masrafa tamirat ve olası müşteri tazminatları eklenecek olursa hem test, hem de test başarımı, hem de test tasarımının önemi açıkça görülmektedir. Bu örnek hem cihaz performansları, hem de kusurlu ürünün nakledilmesi durumunun ortak bir örneğidir. Makine performanslarının detaylarına bir sonraki alt bölümde ayrıntılı olarak değinilecektir.

Önceki bölümden de hatırlanacağı gibi test süreleri ve çıktı oranları da ciddi maliyet kalemleri olarak tanımlanmaktadır.

3.1.2. Temel Kavramlar

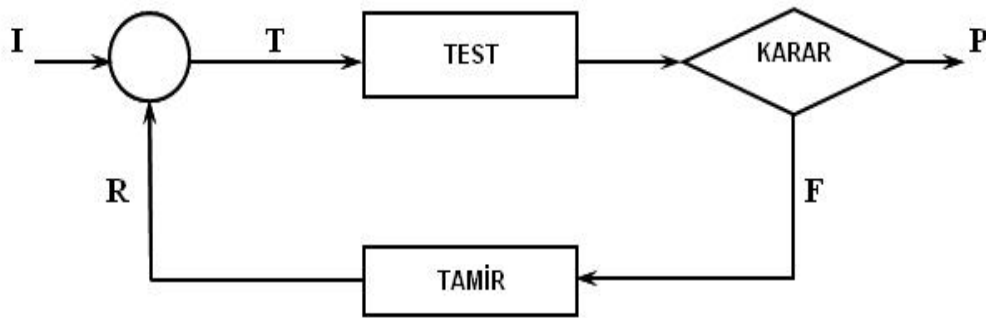
Bu alt bölüm test tasarımı ve test optimizasyonunda bu çalışma kapsamında kullanılacak temel kavramların açıklanmasını ve takip eden alt bölümlerde

açıklanacak hesaplar ve geliştirilecek modellerin anlaşılmasını kolaylaştırmayı amaçlamaktadır.

3.1.2.1. Grafik Gösterim Ve Temel Test Akışı

Test tasarımı ve hesaplamalarında kullanılan temel kavramların kolayca açıklanabilmesi için olası en basit test akışı örnek olarak kullanılacaktır. Aşağıda basit test akışı grafik olarak görülmekte ve grafiğin açıklaması bulunmaktadır:

Şekil 2. Temel Test Akışı



Şekil 2’de de görüldüğü gibi girdi ürünler (I, *Input*) ilk düğüme gelmektedir. İlk düğüm bir envanter noktası gibi kabul edilebilir. Daha sonra test makinesine giden ürünler (T, *Test*) ikiye ayrılmaktadır. Testi geçen ürünler (P, *Pass*) yoluna devam etmekte, geçemeyen ürünler (F, *Fail*) ise tamir makinesine gitmektedir. Tamir gören ürünler (R, *Repair*) tekrar ilk düğüme giderek envantere eklenmekte ve yeniden test edilmektedir. Bu döngü, test stratejisine bağlı olarak belirli sayıda ya da belirli bir çıktı oranına ulaşılan dek tekrarlanabilir. Böyle bir test döngüsünde akış miktarlarını belirleyen, girdi ürünlerin özellikleri ve test ile tamir makinelerinin özellikleridir. Burada tanımlanan test akışı, bu çalışma ile ilintili az sayıdaki akademik araştırmada ortak kullanıldığı görülen bir şekli göstermektedir. Örneğin Rossi ve diğerleri

(2006), Goyal ve Mosher Jr (2006), Fisher ve diğerleri (2007a ve 2007b) çalışmalarında yukarıdaki şekilde gösterilen test akışına çok yakın ya da aynı akışı kullanmaktadırlar.

3.1.2.2. Girdilerin Özellikleri

Girdilerin özellikleri iki temel değişkenle incelenebilir. Bunlar girdilerin test edilecek özelliklerinin sayısı ve her özellik için girdi kitlesindeki başarılı üretim oranı olarak görülmektedir. Girdilerin test edilecek özellikleri f (*feature*) ile gösterilecektir. Basit bir ifadeyle f , nihai ürünün test edilecek özelliklerinin adedini temsil etmektedir. Bu bileşenler bağımsız ya da etkileşimli olarak test edilebilir. Bağımsız test durumunda her özellik tek başına test edilmekte ve test ile tamir faaliyetleri sadece ilgili özelliği etkilemektedir. Etkileşimli test durumunda ise birden fazla özellik bir arada test edilebilmekte ve yapılan test, ilgili özellik dışındaki bazı özellikleri de etkileyebilmektedir. Örneğin bir parçanın mekanik tamiri sırasında başka bir devre hasar görebilmektedir.

Girdi kitlesindeki başarılı üretim oranı ise teste giren ürünün teste tabi özelliğinin düzgün çalışıyor olma olasılığına denk gelmektedir. Bu tez çalışmasında, diğer birçok çalışmada da (örneğin Flehinger, 1965, Butterworth, 1972, Brown ve Proschan, 1983, Gaitonde ve diğerleri, 1993 vs.) olduğu üzere teste girecek ürünlerin test edilecek özelliklerinin düzgün çalışıyor olma olasılıkları konusunda bilgi sahibi olunduğu varsayımı vardır. Diğer taraftan, problemin ele alınışına bağlı olarak, oran ya da olasılık kavramları toplam ürün sayısı ile çarpılarak, oran yerine her özellik için, özelliğin düzgün çalıştığı ürün sayısı da kullanılabilir. Gerçek hayatta çok sayıda ürünün test edildiği göz önüne alındığında her iki durum da ayrı olarak ya da bir arada kullanılabilir. Aralarındaki fark sadece Bernoulli Deneyi ile Binom Deneyi arasındaki farka benzeyecektir. Bu çalışmada, bu alandaki diğer çalışmaların genelinde de olduğu gibi, düzgün çalışan özelliklerin sayısı ya da olasılıkları için G (*Good*), bozuk özelliklerin sayısı ya da olasılıkları için ise B (*Bad*) işaretçileri kullanılacaktır.

3.1.2.3. Makine Performans Karakteristikleri

Test ve tamir makinelerinin performansları da test akışını belirleyecek etkenlerden biridir. Çalışmanın varsayımlarından bir diğeri de bu makine performanslarının karar verici tarafından biliniyor olduğudur. Bu tip varsayımlar literatürde de sıklıkla görülmektedir. Hatta benzer çalışmalarda kullanılmak üzere test ve tamir parametrelerinin tahmin edilmesi başlı başına bir araştırma alanı olarak göze çarpmaktadır. Gianfagna (1978), Volkerink ve diğerleri (2002b), West (2003), Wilson ve diğerleri (2006), Wilson ve Goyal (2007), Wilson ve diğerleri (2007) test verileri, verilerin işlenebilmesi ve verilerden parametre tahminleri yapılabilmesi içerikli çalışmalar ortaya koymaktadırlar. Test makinelerinin hata yapabileceği varsayımını barındıran ve doğru karar verebilme olasılıklarının bilindiği varsayılan çalışmalara örnek olarak Jain ve Agrawal (1985), Nachlas ve diğerleri (1990), Gaitonde ve diğerleri (1995), Ding ve diğerleri (1998), Trichy ve diğerleri (2001), Goyal ve Mosher (2006) ve Wang (2007) gösterilebilir. Tamir makineleri için ise Trichy ve diğerleri (2001) ve Cui ve diğerleri (2004) bu tip varsayımlarda bulunmaktadır. Çalışmanın bu aşamasında ise basit ve bağımsız test döngüleri için performans özellikleri tanımlanacaktır. Etkileşimli test durumları ise ileride ilgili bölümlerde anlatılacaktır.

Gerçek hayatta da rastlanılacağı üzere, her makinenin farklı güvenilirlik ve isabetlilik (*accuracy*) özellikleri vardır. Bir makinenin, sıradan bir işi doğru şekilde yapabileceği gibi düşük değerlerde de olsa hatalı yapma olasılığı bulunmaktadır. Böylesi bir durum, makinenin üretildiği teknolojinin düzeyi, makinenin yaşı, çalışma ortamının fiziksel koşulları ve benzer sebeplere bağlı olarak gerçekleşebilir. Bir test makinesinin bir ürünü test etmesi söz konusu olduğunda oluşan deneyin sonucu ile istatistiksel bir testin sonucu birbirine son derece benzemektedir.

Örneğin makine düzgün çalışan bir ürünü testten geçirir ya da hatalı bir ürünü yakalayarak tamire yollar ise doğru bir iş yapmış demektir. Oysa hatasız bir

ürün testi geçemez ve tamire yollanırsa, hatalı reddetme durumu yani istatistiksel çıkarımda birinci tip hata olarak tanımlanan durum gerçekleşir. Benzer şekilde hatalı bir ürün testi geçer ise, bu kez de hatalı kabul durumu gerçekleşir ve istatistiksel çıkarımdaki ikinci tip hata oluşmuş olur. Bu senaryoların olasılıklarının bilindiği varsayımının tekrar edilmesi bu noktada hayatidir. Gösterim olarak, test makinelerine ait bu olasılıklar α (alfa) sembolü ya da İngilizce kısaltmaları ile verilecektir. Bahsedilen olasılıkların tanımları şunlardır:

α_{TP} : “TP - *True Pass*”, hatasız ürünün testi geçme olasılığı (doğru karar),

α_{FF} : “FF – *False Fail*”, hatasız ürünün testi geçememe olasılığı (I. Tip Hata),

α_{TF} : “TF - *True Fail*”, hatalı ürünün testi geçememe olasılığı (doğru karar),

α_{FP} : “FP – *False Pass*”, hatalı ürünün testi geçme olasılığı (II. Tip Hata).

Bu kavramlar literatürde de sıklıkla araştırmalara dahil edilmişlerdir. Bu çalışmalardan ilk akla gelenler, Nachlas ve diğerleri (1990), Duffuaa ve Al-Najjar (1997), Trichy ve diğerleri (2001), NEMI (2003), Reinosa ve Michel (2003), Rossi ve diğerleri (2006), Goyal ve Mosher (2006), Chien ve diğerleri (2007), Fisher ve diğerleri (2007a ve 2007b), Lee ve diğerleri (2007) tarafından ortaya konmuş çalışmalardır.

Buradan açık olarak görülmektedir ki kusursuz bir ürün ya doğru bir karar ile kabul edilecek ya da yanlış bir karar ile ret edilecektir ve üçüncü bir alternatif senaryo bulunmamaktadır. Benzer durum hatalı ürünler için de geçerlidir. Hatalı bir ürün ya doğru bir karar ile ret edilecek ya da yanlış bir karar ile kabul edilecektir. Bu durumda herhangi bir test makinesi için olasılık değerleri kullanılan şu formüller geçerlidir:

$$\alpha_{TP} + \alpha_{FF} = 1$$

$$\alpha_{TF} + \alpha_{FP} = 1$$

Test makineleri ile ilgili yukarıdaki performans formüllerine denk benzer formüllerin de tamir makineleri için oluşturulabileceği açıktır. Aynı tür hatalı ya da başarılı sonuçlar tamir makinelerinden de gözlenebilir. Örneğin tamir makinesine bozuk olarak gelen bir ürün, tamir sonrası hatasız olarak çıkabilirse doğru işlem yapılmış demektir. Aynı ürün bozuk olarak çıkarsa hata söz konusu olacaktır. Benzer şekilde hatalı test sonucu tamir birimine gelen sağlam bir ürün, tamirden yine sağlam olarak çıkarsa doğru, bozuk olarak çıkarsa hatalı bir işlem yapılmış olacaktır. Tamir makineleri için oluşan değişkenler β sembolü ile gösterilmekte olup açıklamaları aşağıdaki gibidir:

β_{GG} : “GG – *Good to Good*”, sağlam ürünün tamirden sağlam çıkma olasılığı,

β_{GB} : “GB – *Good to Bad*”, sağlam ürünün tamirden bozuk çıkma olasılığı,

β_{BG} : “BG – *Bad to Good*”, bozuk ürünün tamirden sağlam çıkma olasılığı,

β_{BB} : “BB – *Bad to Bad*”, bozuk ürünün tamirden bozuk çıkma olasılığı,

Test makinelerinde görülen eşitliğin bir benzeri olan tamir makineleri denklemleri ise şu şekilde gerçekleşecektir:

$$\beta_{GG} + \beta_{GB} = 1$$

$$\beta_{BB} + \beta_{BG} = 1$$

Bu kavramlara Rossi ve diğerleri (2006), Fisher ve diğerleri (2007a ve 2007b) de değinmektedirler. Böylece temel test akışını etkileyen tüm parametreler tanımlanmış durumdadır. Yukarıda tanımlanan değişkenler ve kavramlarla örnek bir test akışı problemi çözümü aşağıdadır.

3.1.3. Temel Test Akışı Hesaplanması

Bu alt bölümde temel test akışı örnek üzerinden açıklanarak sistem netleştirilecektir. Test edilecek ürünün tek özelliği bulunmaktadır ve ürünlerin tamamı bozuktur. (GI = 0 ve BI = 100). Test ve tamir makinelerinin özellikleri ,

$\alpha_{TP} = 1.0$, $\alpha_{FP} = 0.1$, $\beta_{GG} = 1.0$, ve $\beta_{BG} = 0.5$ şeklindedir. Bu durumda akışın matematiksel hesabı şöyle olacaktır:

$$GO = \alpha_{TP} * G \quad \text{Sağlam çıktıların denklemi}$$

$$BO = \alpha_{FP} * B \quad \text{Bozuk çıktıların denklemi}$$

$$GR = \alpha_{FF} * G \quad \text{Tamire giden sağlam ürünlerin denklemi}$$

$$BR = \alpha_{TF} * B \quad \text{Tamire giden bozuk ürünlerin denklemi}$$

$$GS = \beta_{BG} * BR + \beta_{GG} * GR \quad \text{Tamirden çıkan sağlam ürünlerin denklemi}$$

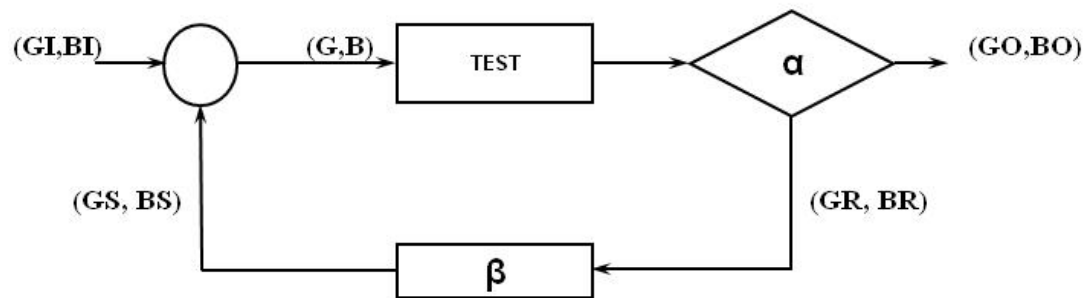
$$BS = \beta_{BB} * BR + \beta_{GB} * GR \quad \text{Tamirden çıkan bozuk ürünlerin denklemi}$$

$$G = GS + GI \quad \text{Teste giren toplam sağlam ürünlerin denklemi}$$

$$B = BS + BI \quad \text{Teste giren toplam bozuk ürünlerin denklemi}$$

Bu ifadeler aynı zamanda test akışı parametrelerini hesaplamakta kullanılacak matematiksel bir modelin, bir doğrusal denklemler kümesinin bileşenleri olarak da görülebilecektir. Matematiksel modelin grafik gösterimi aşağıdaki gibidir:

Şekil 3. Temel Test Akışı Karar Değişkenleri



Yukarıda verilen denklem seti hata payı $\epsilon < 0.0001$ sağlayarak tekrarlı şekilde çözülürse bulunan sonuç aşağıdaki gibi olacaktır:

$$GO = 81.8182$$

$$BO = 18.1818$$

$$G = 81.8182$$

$$B = 18.1818$$

$$GR = 0.0000$$

$$BR = 163.6364$$

$$GS = 81.8182$$

$$BS = 81.8182$$

Sonuçlar adet belirtse de 100 adet ürün girerek ve sürekli değişkenler kullanılarak aslında olasılıklar ifade edilmeye çalışılmıştır. Sonuçlara bakıldığı takdirde hiçbir sağlam ürünün tamire gitmediği ($GR=0$) görülmektedir. Bu sonuç, sağlam ürünlerin testi geçme olasılığı $\alpha_{TP} = 1.0$ olmasından kaynaklanmaktadır. Problemde verilen test ve tamir makineleri kullanılarak, tümü bozuk ürünler, test ve tamir döngüsüne sokulduğunda, çıkan ürünlerin sağlam olma olasılığı %81.82, bozuk olma olasılıkları ise %18.18 olarak görülmektedir.

Aynı makine ile yarısı sağlam yarısı bozuk ürünler ($GI = 50$, $BI = 50$) test edilirse olasılıklar %90.91 ve %9.09 olarak değişecektir. Bu deneyde her yüz ürün için yaklaşık 182 test ve yaklaşık 82 tamir işlemi gerçekleştirilmektedir. Alternatif bir sistemde, tamamı bozuk ürünler, farklı test ve tamir makinelerinde işleme girerlerse ($\alpha_{TP} = 0.85$, $\beta_{bg} = 0.85$), kusursuz çıktı olasılığı %88.44 ve kusurlu ürün olasılığı %11.56 olacaktır. Böyle bir deneyde her 100 ürün için 220 test, 120 tamir işlemi gerçekleştirilmektedir. Aradaki toplam 76 işlemin temel sebebi, ilk sistemde kusursuz ürünlerin testi geçememe gibi bir durumlarının olmaması olarak görülmektedir. Bu da mükemmel bir test makinesinin, kötü bir tamir makinesiyle bile birlikte olsa daha hızlı çalıştığını ve daha başarılı sonuçlar verdiğini göstermektedir.

Yukarıda verilen örnek problem tamamen matematiksel denklemlerin çözümünden oluşmaktadır. Bu problem başka bir stratejide testten çıktıktan sonra yeniden aynı teste sokulabilirdi. Sadece girdi kitlesinin sağlam ürün oranı

değişerek, $GI = 81.8$ ve $BI = 18.2$ olacaktır. Bu girdilerle aynı denklem setinin çözümü ise kusursuz çıktıların %96.7 oranına yükseleceğini göstermektedir. Bu örnek, aynı parti ürünün maliyet etkeni göze alınmamak şartıyla aynı test işlemine ikinci kez sokulmasının faydalı olabileceğini göstermektedir.

Dahası, aşağıda verilen denklem seti, t indeksi tekrar edilecek test sayısını gösterecek şekilde, her t için hesaplanır ise, tekrarlamalı olarak aynı problemi herhangi bir t değeri için çözecektir. Sadece tek bir özelliğin test edildiği akılda tutulursa, aşağıdaki denklem setinin aynı zamanda her farklı t değeri için farklı α ve β değerlerine izin verdiği görülmektedir. Böylece, aynı özelliğin art arda n farklı makinelerde, tekrarlamalı olarak teste girmesi durumunda elde edilecek kusursuz çıktı olasılığının değerini bulmak için kullanılabilir.

$$GS_t = \beta_{BG,t} * BR_t + \beta_{GG,t} * GR_t$$

$$BS_t = \beta_{BB,t} * BR_t + \beta_{GB,t} * GR_t$$

$$G_t = GS_t + GI_t$$

$$B_t = BS_t + BI_t$$

$$GR_t = \alpha_{FF,t} * G_t$$

$$BR_t = \alpha_{TF,t} * B_t$$

$$GO_t = \alpha_{TP,t} * G_t$$

$$BO_t = \alpha_{FP,t} * B_t, \forall t \in (1..n)$$

Çalışmanın temel konusu olan yöneylem araştırması perspektifine geçmeden önce bu temel kavramların görülmesi ve temel matematiksel hesapların incelenmesi önemlidir. Mevcut aşamada artık optimizasyon imkanı bulunmaktadır. Yukarıdaki örneklerde farklı makinelerin farklı sonuçlar verdiği ortadadır. Performansları ile ilintili olarak makinelerin fiyatlarının da değişeceği göz önüne alınırsa, son ürünlerin kalitesi ile makine yatırımı arasında bir ödünleşim fırsatı gözlenmektedir. Bu alt bölümde temel test akışı ile ilgili bir optimizasyon problemi kullanılarak, böyle bir problemin nasıl modelleneceği ve

nasıl sonuç vereceği incelenecektir. Aynı zamanda test tasarımı konusunda optimizasyon kullanımına da bir giriş yapılmış olacaktır.

3.2. TEST TASARIMINDA KARIŞIK TAMSAYI PROGRAMLAMA KULLANIMI

Test tasarımında da farklı alanlarda olduğu gibi sayısal bir teknikle analiz, karşılaştırma, tahmin, istatistiksel hesaplamalar, matematiksel kesin hesaplamalar ve yöneylem teknikleri aracılığıyla optimizasyon yapmak mümkündür. Literatürdeki birçok çalışmada bunların örnekleri vardır. Örneğin Oppermann ve diğerleri (2001 ve 2003a) istatistiksel proses kontrol tekniklerini, Little ve diğerleri (2006) ve Mailharro (1998) kısıt programlama tekniklerini, Yokoyama ve diğerleri (2001) ve Camerana ve diğerleri (2004) karışık tamsayı programlama (MILP, *Mixed Linear Integer Programming*) tekniklerini, Huss ve Gyurcsik (1991) ve Chou ve diğerleri (1997) dinamik programlama tekniklerini, Mergenthaler ve diğerleri (2003), Zou ve diğerleri (2003) ve Viswanadham ve diğerleri (1996) tavlama simülasyonu tekniklerini, Ebadi ve Ivanov (2001), Kotasek ve diğerleri (2003), Rudnick ve diğerleri (1994 ve 1997) ve Yu ve diğerleri (2002) ise genetik algoritmaları kullanmaktadırlar. Görüldüğü gibi çözüm yöntemi anlamında çok sayıda alternatif teknik bulunmaktadır ve çözüm tercihi problem yapısı ve tekniğin çözüm gücü ve hızı ile ilgilidir.

Sürekli değişkenlerden oluşan doğrusal sistemlerin çözümü için alternatifsiz yöntem gibi görünen Simplex Algoritması ve doğrusal programlama, sistemin içine tamsayı değişkenler de eklendiği zaman yerini genellikle karışık tamsayı programlama ya da karışık tamsayı doğrusal programlama yöntemlerine bırakırlar. Bu çalışmada kullanılacak iki çözüm tekniğinden biri de karışık tamsayı programlamadır. Diğer yöntem ise kısıt programlamadır.

3.2.1. Temel Optimizasyon İçin Karışık Tamsayı Programlama Problemi

Bu alt bölümde daha önce açıklanan ve örneklenen temel test akışı problemi artık bir optimizasyon modeli haline getirilecektir. Prensip ve hesapları yukarıda verilen model açıklanacak, akış şemaları, optimizasyon modeli ve çözümü verilecek ve irdelenecektir.

Firma, tek özelliği test edilecek ürünü için iki test makinesi arasında seçim yapacaktır. Kullanılacak tamir makinesi belirlidir. Tamir makinesi kusurlu ürünleri %90 olasılıkla düzeltebilmekte ve kusursuz ürünleri de %10 olasılıkla bozmaktadır. Alternatif tamir makinelerinin bir tanesi, kusurlu ürünleri de kusursuz ürünleri de %70 olasılıkla, diğeri ise her iki işlemi %90 olasılıkla doğru olarak saptayabilmektedir. İlk optimizasyon probleminde çıktılar için en alt kusursuz ürün seviyesi ve makine kapasiteleri göz ardı edilecektir. Böylece test makinelerinin fiyatları, değişken test ve tamir makinesi işletim maliyetleri ve kusurlu ürün çıkması durumunda oluşacak maliyet bilgileri ile oynayarak ideal test makinesi seçiminin nasıl değişeceği örneklenebilecektir. Bu problemin akış şeması Şekil 3'te görüldüğü gibi olacaktır. Kesin hesabı artık matematiksel bir denklem kümesi değil, bir yöneylem modeli haline gelecektir. Bu modeli inceleyince amaç fonksiyonu aşağıdaki fonksiyonun minimize edilmesini gerektirmektedir:

$$VcR * (GS[1] + BS[1]) + Fc * BI[1] + \sum_i VcT_i * (GT_i[1] + BT_i[1]) + \sum_i FcT_i * use[t_i, 1]$$

Amaç fonksiyonu, tamir makinesinin değişken maliyetleri (VcR) ve testi geçen kusurlu ürünlerin maliyetleri (Fc) ile T1 ve T2 test makinelerinin değişken maliyetleri (VcT), sabit maliyetleri (FcT) toplamından oluşan fonksiyonu en aza indirmeye çalışmaktadır. Tamir maliyeti, tamir edilen ürünlerle, kusurlu ürünler toplamı kaçış maliyetiyle çarpılmıştır. Değişken test maliyetleri teste giren ürünlere ve sabit ekipman maliyeti ise kullanılan makine sayısı ile çarpılmıştır. Bu model ayrıca Pynn'in (1997) parmak bastığı, test tekrarının modellere dahil

edilmemesinden kaynaklanan maliyet tahmini hatasına da düşmeyecek bir yapıdadır. Modelin kısıtları ise şu şekildedir:

$$GI[0] = 50$$

$$BI[0] = 50$$

İlk iki kısıt girdi kitlesinin hata dağılımının 50 kusurlu 50 kusursuz üründen oluştuğunu göstermektedir.

$$GI[t] = \alpha_{TP,T1}[t] * GT_1[t] + \alpha_{TP,T2}[t] * GT_2[t]$$

$$GS[t] + GI[t-1] = GT_1[t] + GT_2[t]$$

$$GR[t] = \alpha_{FF,T1}[t] * GT_1[t] + \alpha_{FF,T2}[t] * GT_2[t]$$

$$GS[t] = \beta_{GG} * GR[t] + \beta_{BG} * BR[t]$$

$$BI[t] = \alpha_{FP,T1}[t] * BT_1[t] + \alpha_{FP,T2}[t] * BT_2[t]$$

$$BS[t] + BI[t-1] = BT_1[t] + BT_2[t]$$

$$BR[t] = \alpha_{TF,T1}[t] * BT_1[t] + \alpha_{TF,T2}[t] * BT_2[t]$$

$$BS[t] = \beta_{GB} * GR[t] + \beta_{BB} * BR[t]$$

Bu noktada karşılaşılan ve dörder kısıttan oluşan iki kısıt seti, sırasıyla kusursuz (G) ve kusurlu (B) ürünlerin akışlarını göstermektedir. Girdiler (GI ya da BI kısıtları), iki test makinesine girecek, girdi ve tamir edilenlerin toplamı, test edilenlerin toplamına eşit olacak (ilk GS ve BS kısıtları), testten geçemeyenler tamire gidecek (GR ve BR kısıtları) ve tamire giren ürünler işlem görecektir (ikinci GS ve BS kısıtları). Mevcut kısıtlar, temel test akışı hesabı kısıtlarına oldukça benzemektedir. Bunlardan sonraki kısıtlar ise optimizasyon açısından gerekli kısıtlardır.

$$GI[t] + BI[t] = GI[t-1] + BI[t-1]$$

$$GT_i[t] + BT_i[t] \leq \text{cap}_{T_i} * \text{use}[T_i, t] , \forall i \in (1, 2)$$

$$\sum_i \text{use}[T_i, t] = 1$$

Son grubun ilk kısıtı, girdi ve çıktılarının sayısını eşitlemektedir. Böylece sistem içinde hiçbir ürün kaybolmamaktadır. Hemen sonraki kısıt, iki test makinesinin kapasitelerini sınırlandırmakta ve mevcut kapasitelerine dayanarak kaç test makinesi kullanılması gerektiğini saptamaktadır. Elimizdeki problemde makine kapasiteleri çok yüksek tutulmuş olduğundan, bu kısıt herhangi bir sonuç değişikliği sağlamamakta ve “use” değişkeni, yani kaç makine kullanılacağı değişkeni doğal olarak 0-1 değişken haline gelmektedir. Son kısıt da sistemi iki test makinesinden birini seçmeye zorlamaktadır. Zira hem T1 hem de T2 makinesi için “use” değişkeni aynı anda 1 değeri alamamaktadır.

Yukarıdaki model t indisine (test sayısı) bağlı olarak her kısıt için çoğalacaktır. Bu özelliği, modelin 2 ya da daha fazla test için de kullanılabilir olmasını sağlamaktadır. Aynı zamanda bu özellik sayesinde testin her tekrarı için (her t için), farklı bir α ya da β katsayısı kullanılma fırsatı bulunmaktadır. Bu özellik de belirli bir öğrenme eğrisinin ve olası daha gerçekçi senaryoların modellenbilmesine olanak sağlamaktadır.

Yukarıdaki kısıtlardan oluşan bir optimizasyon modeli temel test akışına dayanarak test makinesi tercihi yapabilmektedir. Benzer bir tercihin tamir makineleri için de yapılabileceğini görmek çok zor değildir. Problem girdileri ile oynanarak bulunan değişik sonuçlar Tablo 2’de sergilenmektedir.

Tablo 2. Temel Test Optimizasyon Problemi Girdileri ve Sonuçları

Değişkenler	Senaryo 1	Senaryo 2	Senaryo 3	Senaryo 4a	Senaryo 4b
$\alpha_{TP,T1}$	0.7	0.7	0.7	0.7	0.7
$\alpha_{TF,T1}$	0.7	0.7	0.7	0.7	0.7
$\alpha_{TP,T2}$	0.9	0.9	0.9	0.9	0.9
$\alpha_{TF,T2}$	0.9	0.9	0.9	0.9	0.9
β_{BG}	0.9	0.9	0.9	0.9	0.9
β_{GG}	0.1	0.1	0.1	0.1	0.1
VcT1	100	100	100	100	100
VcT2	100	200	100	100	100
VcR	100	100	100	100	100
Fc- Hatalı Ürün Maliyeti	1,000	1,000	1,000	8,400	4,600
FcT1	100,000	100,000	100,000	100,000	100,000
FcT2	100,000	100,000	200,000	200,000	200,000
Girdi Miktarı (G-B)	50 – 50	50 – 50	50 – 50	50 – 50	0 – 100
Z – Amaç Fonksiyonu	127,804	142,424	142,424	269,317	283,000
Makine Seçimi	T2	T1	T1	T2	T2
Test Edilen	104.9-56.1	118.2-57.6	118.2-57.6	104.9-56.1	98.8-111
Tamir Edilen	10.5-50.5	35.5-40.3	35.5-40.3	10.5-50.5	9.9-99.9
Ürün Oranı	%94.39	%82.73	%82.73	%94.39	%88.9

Tablo 2, temel problem hakkında önemli denge noktalarını göstermektedir. İlk senaryoya bakıldığında, makine fiyatları (FcT) ve değişken maliyetler (VcT) her iki makine için de aynı olduğu durumda, hem doğru kabul hem de doğru ret oranları yüksek olan ikinci makine tercih edilmiştir. Fakat ikinci senaryoda görüldüğü gibi, daha yüksek performansa sahip ikinci makine, değişken faaliyet maliyeti iki katına çıktığında artık ekonomik seçim olmaktan uzaklaşmaktadır. Daha düşük performansa sahip ilk makine amaç fonksiyonu yani toplam maliyeti yükseltecek şekilde seçilmekte, bu esnada kusursuz ürün oranı %94.39'dan %82.73'e düşmektedir.

Şüphesiz seçimi etkileyen birçok faktör bulunmaktadır. Üçüncü senaryoda değişken maliyetler eşit tutulmuş, bu sefer ikinci makinenin fiyatı yani sabit maliyeti iki katına çıkartılmıştır. Bu fiyat farkı, değişken maliyetlerdeki ve çıktılardaki kusurlu ürün masrafı farkını dengeleyememiş ve tekrar ilk makine tercih edilmiştir. İlk makinenin girdileri aynı olduğu ve sadece ikinci makinenin masrafının artması sonucu ekonomik seçim olmaktan çıkması nedeniyle toplam maliyet değişmemiştir.

Dördüncü senaryonun ilk kısmı, diğer girdilerin seçimi nasıl etkileyebileceğini göstermektedir. Testi geçen hatalı ürün başına oluşan maliyet 1000 birimden 8400'e çıktığı zaman, ikinci makine, iki kat yüksek fiyata (100,000 birim fark) sahip olmasına rağmen, ilk makinenin yol açtığı yüksek hatalı çıktı oranı (yaklaşık %17.3) nedeniyle en ekonomik çözüm haline gelmektedir. Burada toplam maliyetin yaklaşık iki katına çıktığı da gözlenmektedir. Senaryonun ikinci kısmında ise 4600 birimlik kusurlu nihai ürün maliyeti, yaklaşık bir sınır olmaktadır. 4500 ile 4600 değerleri arasında makine tercihi değişmektedir.

Bu senaryoda girdiler değiştirilmiş ve sisteme giren tüm ürünlerin bozuk olduğu varsayılmıştır. Böyle bir durumda yüksek maliyetine rağmen ikinci makine seçilmiş ve 283,000 birim toplam maliyet ve %88.9 başarımlık oranı yakalanmıştır. Kusurlu ürün başına maliyet 4500 olduğu takdirde ise 280,530 birim toplam maliyet ve %66.8 başarımlık oranı söz konusu olmaktadır. Böyle bir durumda bile yaklaşık %1'lik maliyet farkına katlanarak başarımlık oranını %22.1 artırmak firma için daha arzu edilesi bir yol olarak görünmektedir.

3.2.2. Çok Ölçütlü Temel Optimizasyon İçin Karışık Tamsayı Problemi

Bir önceki alt bölümdeki örnekte incelenen optimizasyon problemi, dikkatli bakıldığı takdirde, ayrı olarak çözülmüş iki basit test akışı probleminin karşılaştırması gibi görünmektedir. Problemi biraz daha karmaşıktırmak

amacıyla aşağıdaki kısıtların eklenmesi durumunda model, makine seçerken çıktı kalitesi ve makine sayısı belirlenmesi gibi ölçütleri de dikkate alacaktır.

$$\sum_i use[i, t] \leq nm$$

$$GI[nbt] \geq (GI[0] + BI[0]) * y$$

İlk kısıt toplam kullanılan makine sayısı için bir üst sınır oluşturmaktadır. Elimizdeki örnekte izin verilen makine sayısı çok yüksek tutulduğu için bu kısıt bir önem arz etmemektedir. Aynı zamanda bu kısıtın işe yarar hale gelmesi için, test makinesi kapasitelerinin de çözümü etkileyecek kadar yüksek tutulmaması gerekmektedir. Böylece “use” değişkenleri artık (0–1) değişken gibi davranıp hangi makinenin seçileceğini belirtmek yerine, hangi makineden kaçar tane kullanılması gerektiğini gösterebilecektir. Sonraki kısıt ise testten çıkan ürünlerin içindeki kusursuz ürün oranını belirli bir oranın üzerinde tutmak üzere yazılmıştır.

Burada karşılaşılan y değişkeni, arzu edilen kusursuz çıktı oranını göstermektedir ve önceki bölümlerde de görülen ürün oranı (*yield*) ile eş anlamlı olarak kullanılmaktadır. Başka bir deyişle, yukarıda bahsedilen testi geçen kusurlu çıktı oranının tam tersidir. Bu kısıt özellikle önemlidir zira artık optimizasyon probleminin çözümünde daha düşük kaliteli fakat daha ucuz yöntemin seçilmesi gibi bir lükse yer olmayacaktır. Şüphesiz arzu edilen kalite düzeyi elle ayarlanarak bu imkan sağlanabilir. Yer olmayacak demekten kasıt, belirlenen minimum kalite düzeyinin altına her ne kadar daha hesaplı alternatifler kullanılabilecek olsa da inilmeyecektir.

Böylece orijinal modelden ($\sum_i use[T_i, t] = 1$) kısıtı atılarak problem çözülebilir. Bu

durumda her iki makineden de değişik sayılarda kullanıldığı çözümlere ulaşmak mümkündür. Yine de tek bir makinenin tercih edilmesi ve tüm makinelerin aynı tip olması istendiği takdirde ise modelin bu özelliği zorlaması için aşağıdaki kısıtların eklenmesi gerekecektir.

$$\text{use}[T_i, t] \leq nT_i * M, \forall i \in (1, 2)$$

$$\sum_i nT_i = 1$$

Bu kısıtlar yaygın kullanılan tamsayı programlama kısıtları olup, M yeterince büyük bir sayıyı göstermektedir. Böylece T_1 ya da T_2 makineleri kullanılacak olursa sadece 0 ve 1 değerleri alabilen nT_1 ve nT_2 değişkenleri 1 değeri almaya zorlanmaktadır. Üçüncü kısıt ise nT_1 ve nT_2 değişkenlerinden yalnızca birinin 1 değeri almasını zorlayarak T_1 ve T_2 test makinelerine ait “use” değişkenlerinden birini 0 değeri almak durumunda bırakmaktadır. Böylece sadece tek tip makine seçimi söz konusu olacaktır. Tüm yeni kısıtlar kullanılarak problem tekrar çözüldüğünde bulunan sonuçların bir kısmı Tablo 3’te görülmektedir. $\alpha_{TP,T1} = 0.85$, $\alpha_{TF,T1} = 0.95$, $\alpha_{TP,T2} = 0.95$, $\alpha_{TF,T2} = 0.85$

Tablo 3. Yeni Optimizasyon Problemi Sonuçları

Değişkenler	Senaryo 1	Senaryo 2	Senaryo 3	Senaryo 4	Senaryo 5	Senaryo 6
$\alpha_{TP,T1}$	0.85	0.85	0.85	0.85	0.85	0.85
$\alpha_{TF,T1}$	0.95	0.95	0.95	0.95	0.95	0.95
$\alpha_{TP,T2}$	0.90	0.90	0.90	0.90	0.90	0.90
$\alpha_{TF,T2}$	0.90	0.90	0.90	0.90	0.90	0.90
VcT_1	100	100	100	100	100	100
VcT_2	100	100	100	100	100	100
FcT_1	100,000	100,000	100,000	100,000	100,000	100,000
FcT_2	100,000	100,000	100,000	99,800	99,800	99,800
$CapT_1$	50	50	50	50	50	50
$CapT_2$	50	100	100	50	50	50
Y	%90	%90	%95	%90	%90	%90
Girdi Miktarı	50 – 50	50 – 50	50 – 50	50 – 50	100 – 0	0 – 100
Z	427,142	227,804	427,142	427,004	311,960	540,292
Makine	T1	T2	T1	T2	T2	T1
Adedi	4	2	4	4	3	5
Ürün Oranı	%97.14	%94.39	%97.14	%94.39	%99.87	%94.38

Tablo 3 de göstermektedir ki, fiyat ve kapasite eşitliği durumunda model ilk makineyi seçmektedir. Yani ilk makine daha yüksek performansa sahiptir. Fakat ikinci senaryoda görülmektedir ki, ilk makinenin kapasitesi iki katına artırıldığında, model ürün kalitesinden %3 civarında ödün vererek maliyeti yarıya indirmekte ve ikinci makineyi tercih etmektedir. Firmalar maliyeti şüphesiz ki önemsemek durumundadır.

Fakat pek çok sektörde rastlanabileceği üzere maliyet – kalite ödünleşimi için belirli sınırlar bulunabilmektedir. Böylesi bir durumda firma nihai ürünlerin kusursuz olma olasılığı için katı yaptırımlar uygulayacaktır. Üçüncü senaryoda da görüldüğü üzere, ikinci senaryo üzerine %95'lik sağlam çıktı oranı da eklendiğinde, maliyet tekrar iki katına çıkmakta ve firma daha düşük kapasiteli (yani göreceli olarak diğer makinenin iki katı pahalı) olmasına rağmen çıktı kalitesini tutturabilmek adına tekrar ilk makineye dönmektedir.

Dördüncü senaryo, kapasite ve değişken maliyet dengesi durumunda yaklaşık makine fiyatına bağlı tercihin hangi noktada oluştuğunu göstermektedir. Gerçek değer 99834.5 birimlik bir bedelde gerçekleşmektedir. T_2 makinesinin bu bedel altında satılması durumunda, tercih hep T_2 makinesi olacaktır. Fakat bu diğer şartların aynı kalmasına bağlıdır. Beşinci senaryoda görüldüğü gibi teste giren tüm ürünler kusursuz ise, T_2 makinesinden 4 yerine 3 adet kullanılarak, 311,960 gibi en düşük maliyetle %99.87 verimle en yüksek kusursuz ürün elde edilmektedir. Burada %0.13'lük kusursuz ürün kaybı, test ve tamir makinelerinin mükemmel olmamasından kaynaklanmaktadır. Son senaryo ise göstermektedir ki, teste giren tüm ürünlerin kötü olması durumunda makine performansı ön plana çıkmakta ve T_1 makinesi avantajlı olmadığı bilinen fiyat ve kapasitesine rağmen tercih nedeni olmaktadır. Bu makineden 5 adet kullanılmak suretiyle nihai ürün için saptanan %90'lık başarımlar oranı ancak sağlanabilmekte ve bu esnada gözlenen en yüksek maliyet olan 540,292 ve gözlenen en düşük başarımlar oranı olan %94.38 değerleri elde edilmektedir.

3.2.3. Karışık Tamsayı Programlama Yaklaşımı İle Modellemeden Kaynaklanan Hata

Bir önceki alt bölümde incelenen örnek, kabul ettiği varsayımları dolayısı ile karışık tamsayı programlama kullanılarak çözülebilecek problemlere bir örnektir. Fakat her problem için aynı durum geçerli olmayabilir. Matematiksel olarak incelendiğinde tek bir varsayımın kaldırılması, karışık tamsayı programlama temelli modellerin bu problemleri çözmesini hatalı bir yöntem haline getirmektedir.

Örnek durum olarak makinelerden yalnızca birinin tercih edilmesi durumunun geçerli olmadığı kabul edilsin. Bu problemin modellenmesi için bir önceki alt bölümde belirtilen $(\sum_i use[T_i, t] = 1)$ kısıtına ek olarak, takip eden ve modele sonradan eklenen son üç kısıt da modelden çıkarılmalıdır. Böylece model artık tek bir makine yerine her iki makineden de uygun olduğu kadar seçme şansına sahip olacaktır.

Yukarıda çözülen optimizasyon problemini, aynı makine performans parametreleri (alfa ve beta değerleri), aynı değişken maliyetler (100), aynı sabit maliyetler (100,000), aynı kapasiteler (50), aynı girdi dağılımı (GI = 50 ve BI = 50) ve %90 minimum başarımla oranı ile yeniden tanımlayıp ismine de alternatif durum problemi dediği varsayalım. Alternatif durum problemi çözüldüğünde bulunan sonuçlar aşağıdaki tabloda görülebilir.

Tablo 4. Alternatif Durum Problemi Çözümü

Optimal amaç fonksiyonu değeri : 425811.6883	
use[T ₁ ,1] = 2	use[T ₂ ,1] = 2
GI[1] = 97.1753	BI[1] = 2.8247
GT ₁ [1] = 8.4416	BT ₁ [1] = 56.4935
GT ₂ [1] = 100.0000	BT ₂ [1] = 0.0000
GR[1] = 11.2662	BR[1] = 53.6688
GS[1] = 58.4416	BS[1] = 6.4935

Sonuçlara bakıldığında modelin her iki makineden de kullanmayı tercih ettiği görülmektedir. Sonuçların en ilgi çekici yanı, herhangi bir makine tek başına kullanıldığında elde edilen en yüksek sağlam çıktı oranının daha da üstünde başarımla elde edilmesi olarak görülmektedir. Bu bir sinerji göstergesinden ziyade, temel modelin ancak belirli varsayımlarla çalışabildiğine işaret etmektedir. Zira sinerji bir yana, model, elindeki bilgileri istismar etmektedir. Açık göstergesi ise her iki test makinesinde test edilen miktarlardır. Bu değerler incelendiğinde görülmektedir ki ilk makine kusurluları yakalama konusunda ikincisinden daha başarılıdır ve sonuçlardan da görülmektedir ki neredeyse tüm kötü ürünler bu makinede test edilmiştir. Aynı şekilde kusursuz ürünleri test etmede başarılı olan ikinci makine ise kusursuz ürünlerin tümünü test etmiş, ilk makinede kusursuz hiçbir ürün test edilmemiştir. Bu da göstermektedir ki, model formülasyon olarak doğru görünse de pratikte eksiklikleri bulunmaktadır.

3.3. TEST TASARIMI İÇİN KISIT PROGRAMLAMA KULLANIMI

Önceki alt bölümde, problem açıklamada karışık tamsayı modelleme yaklaşımının eksik kaldığı ve o yaklaşımla modellemenin yöntemin çözüm yapısı nedeniyle eldeki bilgileri kullanmak suretiyle çözümleme esnasında taraflı hareket ettiği ve ortaya çıkan sonuçların da makul olmaktan uzak, yanlış değerler olduğu gösterilmişti.

Sonuçların makul olamamasının temel sebebi, optimizasyon algoritmalarının karı maksimize etmek ya da zararı minimize etmek amacıyla, değişkenlerin alacakları değerleri çözümün uygunluğunu bozmayacak kadar öteleme eğiliminde olmasıdır. Belirli problemlerde sorun oluşturmayacak bu durum, test makinesi seçiminde ciddi yanlışlıklara yol açmaktadır. Özet olarak kusursuz ürünler kendilerinin testi geçme olasılıklarının daha yüksek olduğu makineye, kusurlu ürünler ise kendilerinin testi geçememe olasılıklarının daha yüksek olduğu makineye aktarılmaktadır. Hangi ürünün kusurlu, hangisinin ise kusursuz

olduğunun bilinmediği gerçeğine rağmen, böylesi bir yaklaşımla kurulan model, çözüm algoritmasının adeta böyle bir bilgiye sahipmiş gibi davranmasına ve bilgiyi istismar etmesine yol açmaktadır.

Fakat bu durum gerçekçilikten hayli uzaktır. Karışık tamsayı programlama modeli adeta test makinelerinin başında durup hangi ürünün iyi olduğunu anlayıp, uygun makineye yönlendiren bir kahin gibi davranmaktadır. Böyle bir bilgiye sahip olunması durumunda ise test yapmaya gerek olmadığı ortadadır. Öyle ise modellemede karşılaşılan bu yaklaşım sorunu çözümlmelidir ki tek özellik için test tasarımı rahatlıkla yapılabilir.

Tam bu noktada ortaya çıkan iyi ve kötü ürünlerin akışının uygun makinelere yapılması sorunu, bir başka ifadeyle rastsallığın ortadan kalkması sorunudur. Bu sorunun çözülmesi için, örneğin istatistik biliminin örnekleme dalında sistematik örnekleme (*systematic sampling*), rastsal sayılar tablosu kullanımı ya da orantılı dağıtım (*proportional allocation*) gibi yöntemler kullanıldığı bilinen bir gerçektir. Rastsallık prensibi, denk olasılıklar öngördüğü için, herhangi bir ürünün herhangi bir makineye gitmesi olasılığının tüm ürünler ve tüm makineler arasında eşit olması gerektiğini söylemektedir. Eşit olasılıkların sağlanması durumunda da beklenen durum, ürünlerin girdi dağılımının (yani iyi ve kötü olma olasılıklarının, GI ve BI dengelerinin) tasarımda kullanılacak her makineye girecek ürünler için aşağı yukarı eşit olmasını gerektirmektedir.

Başka bir deyişle, girdilerin kusur oranı test için seçilecek makinelerin gerçekleştirdiği testin ilk seferi için her makineye gelen ürün için aynı kalacaktır. Test sonrası değişen oranlar tekrar test öncesi birbiri ile birleşecek ve test tekrarında tüm makineler arasında eşit kalacaktır. Gerçek hayatta elbette ki eşitlik söz konusu olmayabilecektir. Fakat uzun vadede, çok fazla ürün için test yapıldığında gelinecek denge noktasındaki beklenen değer, her makineye giren ürünlerin ortalama kusur oranının birbirine eşit olacağıdır.

Böylece karşılaşılan durum, yani makinelere dağıtılan ürünlerde karşılaşılabilecek kusur oranlarının aynı olması durumu, istatistiksel örnekleme konusunda da karşılaşılan orantılı dağıtım (*proportional allocation*) durumu ile aynıdır.

Aşağıdaki alt bölüm yukarıda bahsedilen sorunların kısıt programlama ile çözülecek bir modelleme yaklaşımına gidilerek nasıl giderilebileceğini açıklamaktadır. Bir sonraki alt bölüm ise kısıt programlama modelleri ile karışık tamsayı programlama modellerini karşılaştırmakta ve örnekler aracılığı ile değerlendirmektedir.

3.3.1. Kısıt Programlama İle Modelleme

Akışları orantılı dağıtım yoluyla düzenlemek için modele eklenmesi gereken yeni kısıtlar bulunmaktadır. Bu kısıtların temel prensibi şu şekildedir: Eğer bir test makinesi, optimal test tasarımı içinde yer alacaksa, bu makineye herhangi bir zamanda girecek olan ürünlerin sağlam – bozuk ürün dağılımı, bu makineye ürün yollayan stok noktasındaki ürün dağılımının aynısı olmalıdır. Daha önceki alt bölümlerde incelenen problem üzerinden örnekleme yapılarak bu kısıtın matematiksel ifadesi aşağıdaki gibidir. Hatırlanacağı gibi $use[i,t]$ değişkeni, “t” sıralı testte “i” test makinesinden kaç adet kullanılacağını, $GT_1[t]$ değişkeni de “t” sıralı test esnasında T_1 makinesinde test edilen kusursuz ürünlerin adedini (örnek problemlerde hep yüz üzerinden adedini dolayısıyla oranını göstermiştir) göstermektedir.

$$use[T_i,t] > 0 \Rightarrow \frac{(GI[t-1]-\varepsilon)}{(GI[t-1]+BI[t-1])} \leq \frac{GT_i[t]}{(GT_i[t]+BT_i[t])} \leq \frac{(GI[t-1]+\varepsilon)}{(GI[t-1]+BI[t-1])}, \quad \forall i \in (1,2)$$

Yukarıda görülen kısıt T_1 ve T_2 makineleri için, bu makinelerin kullanılması durumunda test ettikleri ürünlerin içinde bulunan kusursuz ürün oranı için hem bir alt sınır hem de bir üst sınır oluşturmaktadır. Kısıtlarda görülen ε terimi, görmezden gelinebilecek bir hata payının yarısını ifade etmektedir. Böylece

herhangi bir test makinesine giden ürünler, toplam 100 adet ürün girildiği düşünülürse, baştaki girdi oranının $\%2\epsilon$ yakınlığında dağılacaktır. Kusurlu olmayan ürünler için kısıtlar yazıldığı zaman, kusurlu ürünler için bu kısıtları tekrar yazmak gereksizdir zira basit matematik de göstermektedir ki bir grup orantılı dağıtıldıktan sonra diğer grup da doğrudan orantılı dağılacaktır. Yine de bu kısıtlar bir değişime daha tabi tutulmalıdır. Zira kısıt programlama tarafından tümü tamsayı cinsinden kabul edilecek ifadeler yüzünden, her biri yüzde olasılığı gösteren bu kısıtların sıfıra yuvarlanması olasılığı baş gösterecektir. Bu durumda her iki kısıt da $(0 \leq 0 \leq 0)$ formunu alacak ve herhangi bir sorun görülmeyecektir.

Bahsedilen problemde kaçınmanın en iyi yolu kısıtları 100 ile çarpmaktır. Böylece ifadeler sıfıra yuvarlanabilecek $0.42 \leq 0.45 \leq 0.48$ örneği yerine $42 \leq 45 \leq 48$ değerleri alacak ve esas amacına ulaşacaktır. Araştırmacı istediği takdirde 100 yerine daha büyük rakamlar kullanabilecektir. Örneğin 1,000 ya da 10,000 kullanıldığı takdirde binler ve on binler basamaklarına kadar zorlama oluşturulabilecektir. Örneğin girdi kitlesinin $[50,50]$ dağıldığı ve $\epsilon = 5$ olarak atandığı varsayılırsa elde edilecek kısıtlar her i için şu şekilde olacaktır:

$$\text{use}[T_i, t] > 0 \Rightarrow 100 * \frac{(GI[t-1]-\epsilon)}{(GI[t-1]+BI[t-1])} \leq 100 * \frac{GT_i[t]}{(GT_i[t]+BT_i[t])} \leq 100 * \frac{(GI[t-1]+\epsilon)}{(GI[t-1]+BI[t-1])}$$

Böylece kısıtların sayısal ifadesi ve tam değerleri aşağıdaki gibi olacaktır.

$$\text{use}[T_i, t] > 0 \Rightarrow 100 * \frac{(500-5)}{(50+50)} \leq 100 * \frac{GT_i[t]}{(GT_i[t]+BT_i[t])} \leq 100 * \frac{(500+5)}{(50+50)}$$

$$\text{use}[T_i, t] > 0 \Rightarrow 45 \leq 100 * \frac{GT_i[t]}{(GT_i[t]+BT_i[t])} \leq 55$$

Modelleme açısından tek fark hemen yukarıda verilmiş olan son kısıttır. Fakat bu kısıta bakıldığında hemen fark edilmektedir ki birbirinin farklı makineler için tekrarı olacak her iki kısıt da, hem bir mantık operatörü (“ise” işareti), hem de doğrusal olmayan kısıtlar içermektedir (bölme işlemleri). Yukarıdaki örnekte 45

ve 55 değerleri sadece örnek oluşturmaktadır. Bu değerler normalde karar değişkenidir ve probleme bağlıdır. Kısıtların bu son haline buradan itibaren “tamsayılaştırılmış kısıtlar” denecektir.

Kısıtlarda rastlanan mantık operatörü “ise” ($\Rightarrow$ işareti) bir şart durumunu göstermektedir. Kendisini takip eden kısıt, ancak bu işareten önceki şart doğru olursa devreye girecektir. Bu da, ürünlerin her bir makineye belirli bir kusur oranına göre dengeli dağıtılması prensibinin ancak o makine optimal çözüm içinde seçilen bir makine ise devreye girmesini sağlamaktadır.

Akışları dengeleyen kısıtlar ise basit olarak sağlam ve bozuk ürünlerin test edilme oranlarının makineler arasında dengeli olmasını zorlamakta olup, sadece o makine seçilmiş ise devreye girmektedir. Orantıyı, test makinesinin hemen önündeki stok noktasında oluşan orana göre belirleyen bu kısıtlar, modelin ve sonuçların gerçekçi olmasını sağlamak için yazılmışlardır.

Karışık tamsayı programlama teknikleri yapıları itibarıyla çok kısıtla ifade edilebilen bazı mantık önermelerini çözebilmekte, fakat doğrusal olmayan kısıtları ise kesinlikle çözememektedirler. Yukarıda incelenen problemin yapısı itibarı ile doğrusal olmayan kısıtların kullanılması ise hayati bir önem taşımaktadır. Böylece açıkça görülmektedir ki içerikleri belli, matematiksel ifadeleri yazılmış bu kısıtlar, karışık tamsayı programlama ile modellenip çözülen herhangi bir yöntemin içine eklenemeyecektir. İşte tam bu noktada kısıt programlama teknikleri devreye girmektedir. Kısıt programlama belirli şartlar altında hem mantık önermelerini, hem de doğrusal olmayan kısıtları çözebilecek bir modelleme yaklaşımına fırsat tanımakta ve bu kısıtlara sahip modelleri çözebilen yöntemleri içeren çözüm algoritmalarına sahip bir optimizasyon alanıdır.

Yukarıdaki kısıtlar hemen hiçbir değişikliğe uğramadan kısıt programlama kısıtları olarak yazılarak, bir önceki alt bölümde verilen karışık tamsayı programlama tabanlı optimizasyon modeline eklenebilecektir. Burada dikkat

edilmesi gereken, kısıt programlamanın sadece kesikli tanım kümeleri kapsamında çalışabildiği gerçeğidir. Bu nedenle özellikle doğrusal olmayan kısıtlarda ve mantık önermeleri içeren kısıtlarda yer alan tüm değişkenlerin tamsayı değişkenler haline getirilmesi hayati önem arz etmektedir. Haliyle GI, BI, GT₁, BT₁, GT₂ ve BT₂ değişkenlerinin tamsayı değişkene çevrilmesi ve tanım kümelerinin makul bir aralıkta tanımlanması gereklidir. Mevcut problemde bu değerler (0 – 200) olarak kullanılmıştır. Problemin tanımlandığı kısıt programlama modeli aşağıda görülmektedir.

Minimize

$$VcR * (GS[1] + BS[1]) + Fc * BI[1] + \sum_i VcT_i * (GT_i[1] + BT_i[1]) + \sum_i FcT_i * use[t_i, 1]$$

Subject to

$$GI[0] = 80$$

$$BI[0] = 20$$

$$GI[t] = \alpha_{TP,T1}[t] * GT_1[t] + \alpha_{TP,T2}[t] * GT_2[t]$$

$$GS[t] + GI[t-1] = GT_1[t] + GT_2[t]$$

$$GR[t] = \alpha_{FF,T1}[t] * GT_1[t] + \alpha_{FF,T2}[t] * GT_2[t]$$

$$GS[t] = \beta_{GG} * GR[t] + \beta_{BG} * BR[t]$$

$$BI[t] = \alpha_{FP,T1}[t] * BT_1[t] + \alpha_{FP,T2}[t] * BT_2[t]$$

$$BS[t] + BI[t-1] = BT_1[t] + BT_2[t]$$

$$BR[t] = \alpha_{TF,T1}[t] * BT_1[t] + \alpha_{TF,T2}[t] * BT_2[t]$$

$$BS[t] = \beta_{GB} * GR[t] + \beta_{BB} * BR[t]$$

$$GI[t] + BI[t] = GI[t-1] + BI[t-1]$$

$$GT_i[t] + BT_i[t] \leq cap_{Ti} * use[T_i, t], \forall i \in (1, 2)$$

$$\sum_i use[i, t] \leq nm$$

$$GI[nbt] \geq (GI[0] + BI[0]) * y$$

$$use[T_i, t] > 0 \Rightarrow 100 * \frac{(GI[t-1]-3)}{(GI[t-1] + BI[t-1])} \leq 100 * \frac{GT_i[t]}{(GT_i[t] + BT_i[t])} \leq 100 * \frac{(GI[t-1]+3)}{(GI[t-1] + BI[t-1])}$$

$$\forall i \in (1, 2)$$

Burada önemli birkaç nokta bulunmaktadır. Öncelikle bu ebatla bir problem için ϵ terimi 3'ten küçük değerler aldığı takdirde problem çözümsüz kalmaktadır. Pratikte 100 yerine 10,000 ile çarpım kullanılmıştır. Böylece problemin kritik doğrusal olmayan kısıtları bir nebze olsun rahatlatılmış ve arama rutinlerine esneklik sağlanmıştır. Katsayıların 10,000 olması nedeni ile ϵ terimi de 300 değeri almıştır.

Ayrıca en önemli noktalardan biri de tamsayılaştırılmış kısıtlara sahip problemin çözülebilmesi için makine performansları ve girdi kitlesinin dağılımı arasında da bir ilişki bulunmaktadır. Örneğin mevcut problemdeki makineler kullanıldığı takdirde girdiler içindeki kusursuz ürün oranına ait alt sınır %79 olarak hesaplanmıştır.

Bir sonraki alt bölüm örnek problemleri içermektedir. Bu alt bölümde yeni eklenen akış orantılama kısıtlarının hem ilk hem de son hali kullanılarak karşılaştırma yapılacaktır.

3.3.2. Örnek Problemler

Problemin, karışık tamsayı programlama modeli ile karşılaştırmalı sonuçları aşağıdaki tabloda görülmektedir. İlk tablo akış orantılama kısıtlarının ilk halinin kullanıldığı kısıt programlama modelini içermektedir ve girdileri önceki bölümde kullanılan karışık tamsayı programlama modeli ile aynıdır. İkinci tabloda ise tamsayılaştırılmış akış orantılama kısıtları kullanılmaktadır. Bu problemde girdi kitle dağılımı [80,20] olarak değiştirilmiş ve $\epsilon = 3$ olarak kullanılmıştır. Girdi kitle dağılımı karışık tamsayı programlama modelinde de değiştirilmiştir.

Tablo 5. MIP – CP Karşılaştırmalı Çözümler 1

MIP	CP	Simülasyon
Z = 425811.6883	Z = 427000.0000	
Use[T ₁ ,1] = 2 use[T ₂ ,1] = 2	use[T ₁ ,1] = 3 use[T ₂ ,1] = 1	use[T ₁ ,1] = 3 use[T ₂ ,1] = 1
GI[0] = 50.0000 BI[0] = 50.0000	GI[0] = 50 BI[0] = 50	GI[0] = 50 BI[0] = 50
GI[1] = 97.1753 BI[1] = 2.8247	GI[1] = 97 BI[1] = 3	GI[1] = 96.3 BI[1] = 3.7
GT ₁ [1] = 8.4416 BT ₁ [1] = 56.4935	GT ₁ [1] = 94 BT ₁ [1] = 54	GT ₁ [1] = 83.9 BT ₁ [1] = 42.7
GT ₂ [1] = 100.0000 BT ₂ [1] = 0.0000	GT ₂ [1] = 19 BT ₂ [1] = 3	GT ₂ [1] = 27.5 BT ₂ [1] = 14.6
GR[1] = 11.2662 BR[1] = 53.6688	GR[1] = 16.0000 BR[1] = 54.0000	GR[1] = 15.1 BR[1] = 53.6
GS[1] = 58.4416 BS[1] = 6.4935	GS[1] = 63.0000 BS[1] = 7.0000	GS[1] = 61.8 BS[1] = 6.9

Tablo 6. MIP – CP Karşılaştırmalı Çözümler 2

MIP	CP	Simülasyon
Z = 317922.0779	Z = 488000.0000	
use[T ₁ ,1] = 1 use[T ₂ ,1] = 2	use[T ₁ ,1] = 1 use[T ₂ ,1] = 2	use[T ₁ ,1] = 1 use[T ₂ ,1] = 2
GI[0] = 80.0000 BI[0] = 20.0000	GI[0] = 80 BI[0] = 20	GI[0] = 80 BI[0] = 20
GI[1] = 98.8312 BI[1] = 1.1688	GI[1] = 98 BI[1] = 2	GI[1] = 97.97 BI[1] = 2.03
GT ₁ [1] = 10.3896 BT ₁ [1] = 23.3766	GT ₁ [1] = 30.8 BT ₁ [1] = 6.8	GT ₁ [1] = 36.3 BT ₁ [1] = 7.6
GT ₂ [1] = 100.0000 BT ₂ [1] = 0.0000	GT ₂ [1] = 79.8 BT ₂ [1] = 16.6	GT ₂ [1] = 73.5 BT ₂ [1] = 15.5
GR[1] = 11.5584 BR[1] = 22.2078	GR[1] = 12.60000 BR[1] = 21.40000	GR[1] = 12.74 BR[1] = 21.22
GS[1] = 30.3896 BS[1] = 3.3766	GS[1] = 30.60000 BS[1] = 3.40000	GS[1] = 30.3 BS[1] = 3.3

Tablo 5'te bulunan çözümlere bakıldığında ilk görülen, kısıt programlama modelinin daha sıkı şartlar koyuyor olmasının etkisiyle masrafların biraz artmış olmasıdır. Fakat %2.8'lik bir toplam fiyat artışı çok yüksek olmamakla birlikte, kısıt programlama modelinin kurulma ve çözülme imkanının olmadığı durumlarda karışık tamsayı modelinin de optimal olmasa da fikir verecek çözümler üretebileceğini göstermektedir.

Gerçek problem çözümünde ise maliyetin %53.5 oranında arttığı görülmektedir. Bu sonuç gerçeği yansıtan ciddi bir model kullanılmadığı takdirde gerçeği yansıtmayan ve uygulanması mümkün olmayan bir planın ne kadar cazip görünebileceğini ve fakat gerçeği yansıtan hesaplar kullanılmaması takdirde oluşabilecek zararı ortaya koymaktadır.

Makine seçimlerine bakıldığında, ilk tabloda kısıt programlama modelinin optimal çözümünde T_1 makinesinden 3 adet ve T_2 makinesinden 1 adet bulunmasıdır. Oysa Tablo 6'da bulunan gerçekçi çözümün makine seçimi karışık tamsayı programlama modelinin optimal çözümü ile aynıdır. Girdi kitlesi dağılımlarındaki radikal fark, tamir ve tekrar test değişkenlerini ciddi olarak etkilediğinden bu değerlerin yorumlanması fazla bir önem arz etmemektedir.

Kısıt programlama modelinden asıl beklenen konuda ise değerler gerçeğe oldukça yaklaşmaktadır. Kusurlu ve kusursuz ürünlerin akışı söz konusu iken, ilk tabloda, karışık tamsayı yaklaşımıyla modellemeden kaynaklanan hatayla yanlış sonuçlar gözlenen T_2 makinesinin, artık sadece iyi ürünlerin tümü yerine, her iki tip ürün için de çalıştığı dikkat çekmektedir. T_1 makinesinde istismar edilebilecek kusurlu ürünlerin ağırlıklı olarak test edilmesi yerine dengeli bir şekilde kusursuz ürünlerin ağırlıklı test edildikleri görülmektedir.

Kısıt programlama modelinde oluşan şartlardan dolayı tamire gönderilen ürün miktarı her iki tipte de hafifçe artmış, çıktılarının kalitesi konusunda ise iki model yaklaşık aynı sonuçları vermiştir. Doğrusal olmayan yapıdan kaynaklanan tamsayı değişkenlerden dolayı %3'e yuvarlanan kusurlu ürün oranı, tamsayı programlama modelindeki sürekli değişken durumunda %2.83 olarak görülmektedir. Aradaki %0.17 fark ise yuvarlamadan oluşmaktadır. Bu açıdan bakıldığı zaman yumuşak akış orantısı kısıtları kullanan kısıt programlama modelinin çok daha etkili olduğu açıkça ortadadır.

İkinci tablo ise farklı bir durumu sergilemektedir. Ağır ve tamsayılaştırılmış akış orantılama kısıtları içeren kısıt programlama modeli, girdi dağılımının çok

yakınında oranlarla çalışan makineler seçmiştir. Her iki makinede de hem kusurlu hem kusursuz ürünler test edilmektedir. Zaten formülü itibari ile bu problemin tam rastsalılık sağlaması beklenmekte ve neredeyse tamamen girdi ile orantılı malzeme akışı elde edilebilmesi gerekmektedir.

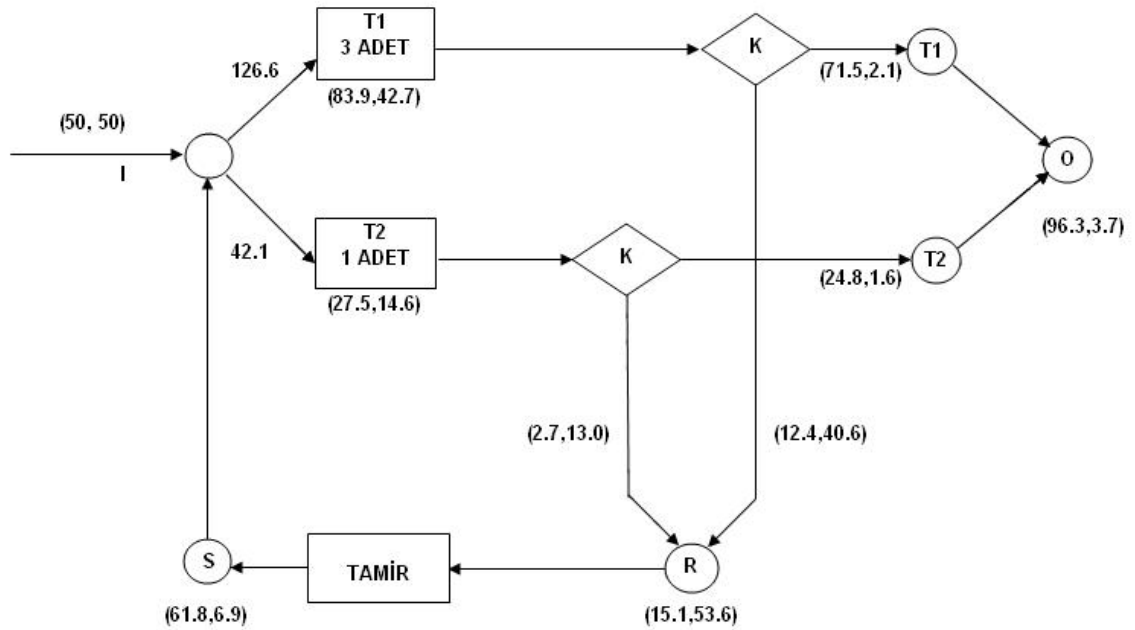
Bu ölçütler aşağıda tartışılacaktır. Tamir sürecindeki değerler hem kısıt hem de tamsayı programlama modellerinde birbirlerinin neredeyse aynısı olarak oluşmaktadır. Kusursuz ürün oranı ise %98 civarında gerçekleşmektedir.

İlk kısıt programlama modeli, ikincisinden sonra gereksiz gibi görünse de aslında bu hatalı bir kanı olacaktır. Bunun nedeni ilk problemin simülasyon çözümü ile bilgisayar çözümü farklarının değerlendirilmesi ile anlaşılabacaktır.

Araştırmacı, sonuçlara bakarak yine de bazı belirsiz noktaları sorgulayabilecektir. Örneğin ilk makinede test edilen ürünler için kusurlular toplamın yaklaşık üçte birini oluştururken, ikinci makinede kusurlu ürünlerin ciddi miktarda azınlık kalması gene bir istismar durumu olup olmadığı sorusunu akla getirmektedir. Sonuçları gösteren tablolarda ise, bu problemin makinelere eşit oranda toplam ürün dağıtımı gözetilmek sureti ile simülasyon çözümleri de gösterilmektedir.

Şekil 4, ilk kısıt probleminin simülasyon çözümünü göstererek tabloda bulunan değerlerin hangi noktalarda görüldüğü konusunda referans olmaktadır. Grafik, akış kapsamında bulunan tüm düğüm ve oklar üzerinde oluşan ürün trafiğini göstermektedir. Yukarıdan aşağı doğru her turda gerçekleşen kusursuz ve kusurlu ürünlerin toplam miktarları parantezler içinde verilmekte ve o noktada oluşan toplam ürün trafiğini göstermektedir.

Şekil 4. Temel CP Problemi Simülasyonla Kesin Çözüm



İlk sonuç tablosuna bakılırsa, tamire giren değerlere bakılırsa simülasyon çözümde yaklaşık (15.1,53.6) değer tamire girdiği fark edilecektir. Bilgisayar tarafından belirlenen kısıt programlama çözümünde bu değer (16,54), karışık tamsayı çözümünde ise (11,54) olarak bulunmuştur. Tamirden çıkan ürünler incelenirse, simülasyon çözümde yaklaşık (62,7), kısıt programlama çözümünde (63,7) ve karışık tamsayı çözümünde (58.5,6.5) bulunmuştur. Kısıt programlamada yüksek değerlerin oluşma sebebi daha önce de belirtildiği gibi, bu modelin göreceli daha sıkı şartlara sahip olmasıdır. İkinci sonuç tablosunda ise her üç sonucun birbirine çok yakın tamir akışlarına sahip olduğu açıktır. Özellikle kısıt programlama ve simülasyon çözüm arasındaki fark neredeyse yok denecek kadar azdır.

Kusursuz ve kusurlu ürünlerin makinelerle nasıl dağıldığına bakılırsa, ağır akış orantılarına sahip ikinci modelde kullanılan makinelerin, ciddi bir şekilde %81.9 ve %82.7 kusursuz ürün oranları ile çalışmakta oldukları görülecektir. Bu sonuç neredeyse mükemmel rastsallığa işaret etmektedir. İlk kısıt programlama çözümünde sorunlu görünen nokta, iki makinede test edilen kusursuz ürün oranlarındaki farklılıktır. Bahsedilen durum sadece ilk problemin

çözümü için geçerlidir. Karışık tamsayı çözümünde sırasıyla %12.9 ve %100 çıkan kusursuz ürün test edilme oranlarına rastlanmaktadır.

Kısıt programlama çözümünde ilk test makinesinde 94 kusursuz, 54 kusurlu toplam 148 ürün test etmektedir. Karşılaşılan oran ise %63.5 kusursuz ürün şeklindedir. İkinci makinede ise bu oran (19,3) değerlerinden dolayı %86'ya yükselmiştir. Oysa simülasyon çözüme bakıldığında ilk makinede (83.9, 42.7), ikinci makinede ise (27.5, 14.6) ürün test edildiği görülmektedir. Bu iki makinede test edilen kusursuz ürün oranı sırasıyla %66.3 ve %65.3 olarak görülmektedir. Bu iki değer hem birbirlerine, hem de kısıt programlamada kullanılan 3 adet T_1 makinesine oldukça yakındır. O zaman kısıt programlamada kullanılan T_2 makinesinde neden bu kadar farklı bir değer ortaya çıkmıştır? Model iyi ürünleri bu makineye göndererek bilgileri karışık tamsayı programlama modeli gibi istismar mı etmektedir?

Bu soruların cevabı olumsuz olacaktır. Nedenini görebilmek için ise her iki çözüm yönteminde makinelerin kullanılan kapasitelerine bakmak gereklidir. Kullanılan kapasiteler ile bilgiler aşağıdaki tabloda gösterilmektedir.

Tablo 7. Test Makinelerinin Kapasite Kullanım Oranları (Problem 1)

Çözüm	Birim Kapasite	Toplam Kapasite	Toplam Kullanılan	Kapasite Kullanımı
T_1 MIP	50	100	64.94	%64.9
T_2 MIP	50	100	100.0	%100
T_1 CP	50	150	148	%98.7
T_2 CP	50	50	22	%44
T_1 EL	50	150	126.6	%84.4
T_2 EL	50	50	42.1	%84.2

Tablo 8. Test Makinelerinin Kapasite Kullanım Oranları (Problem 2)

Çözüm	Birim Kapasite	Toplam Kapasite	Toplam Kullanılan	Kapasite Kullanımı
T ₁ MIP	50	50	33.77	%67.5
T ₂ MIP	50	100	100.0	%100
T ₁ CP	50	50	37.6	%75.2
T ₂ CP	50	100	96.4	%96.4
T ₁ EL	50	50	43.9	%87.8
T ₂ EL	50	100	89.0	%89

Öncelikle belirtilmelidir ki, simülasyonla yapılan kesin hesaplamalarda herhangi bir optimizasyon kaygısı güdülmeden, kısıt programlama çözümünde elde edilen makine tercihleri kullanılmış olup, hem makine kapasiteleri, hem de kusurlu – kusursuz ürün açısından orantılı bir şekilde ürün dağıtılmıştır. Böyle bir durumda her iki makine tipinde yer alan makinelere denk görevler düşmesi gerekmektedir. Simülasyonla yapılan hesaplarda görülen kusurlu – kusursuz oranında (ilk problem için %66.3 ve %65.3 ve ikinci problem için %82.5 ve %82.6) ve yukarıdaki tabloda görülen hesaplanmış kapasite kullanım oranlarında (ilk problem için %84.4 ve %84.2 ve ikinci problem için %87.8 ve %89) rastlanan yakınlığa bakılarak da bu durum teyit edilebilecektir. Simülasyonla çözüm haricinde böyle bir eşit kapasite kullanımı söz konusu değildir. Ancak ağır kısıtlı ikinci çözüm tablosunda görülen kısıt programlama çözümü makul sayılabilecek düzeyler göstermektedir (iki makine için sırasıyla %75.2 ve %96.4).

Bu durumun nedeni açıktır. Simülasyonla hesaplamada optimizasyon kaygısı güdülmemesi ve diğer iki yöntemde durumun tam tersi olması. Fakat bu nedenledir ki, karışık tamsayı programlama modeli elindeki bilgileri istismar etmektedir. Mevcut şartlar altında kısıt programlama çözümüne bakılacak olursa, ilk problemde sorun gibi görünen ikinci makinedeki %86'lık oran açıklanabilecektir. Görüldüğü gibi iki makinede yapılan toplam 170 test faaliyetinin 148'i T₁ tipindeki 3 makinede gerçekleştirilmiştir. T₂ makinesi ise kalan son 22 test işlemini yapmaktadır. Bu durumda model zaten öncelikli

olarak T_1 makinelerini tercih etmekte ve %98.7 kapasite ile çalıştırmaktadır. Diğer makinede ise sadece kalan işleri yapmaktadır. Esas makineler olan T_1 'lerde ise kusursuz ürün test edilme oranı, en orantılı ve dengeli çözüm olan Simülasyonla çözümden sadece %2.8 sapma göstermektedir. İkinci kısıt programlama modelinde ise böyle bir durum söz konusu değildir. Zira test makinelerine giden ürünlerin içinde rastlanacak kusurlu kusursuz oranının korunmasını zorlayan ağır kısıt, makine verimliliğinin göz ardı edilerek rastsallığın ön planda tutulmasına odaklanmaktadır.

Gerçek hayat ile bağlantısı kurulacak olur ise her iki kısıt programlama modeli de durumsal olarak gerçekçidir. Makine tiplerinden birinin diğerine göre daha tercih edilir olduğu bilgisi mevcut ise, rasyonel bir firma şüphesiz o makinenin verimini artırmaya yönelecektir. Böyle bir bilgi ya da tercihin bulunmaması ve makinelerin üretimi (mevcut durumda test sürecini) amaca uygun şekilde karşılaması önemli tutulduğunda ise ikinci model neredeyse mutlak rastsallığa müsaade eden yapısı ile gerçeği düzgün yansıtan bir karar aracı haline gelecektir. Bütün bu bilgiler ışığına açıkça görülmektedir ki kısıt programlama modelleri tarafsız çalışan, elindeki bilgileri istismar etmeyen sonuçlar vermektedir. Simülasyonla çözülemeyecek daha büyük problemler söz konusu olduğunda da güvenilir bir şekilde makine seçimlerini yapabilecektir.

4. BÖLÜM

ÇOK ÖZELLİKLİ ÜRÜNLERİN TESTİ

4. GİRİŞ

Bir önceki bölümde test edilecek tek özelliği olan ürünlerin test dinamikleri, akışları, kesin hesapları ve nasıl optimize edileceği incelenmişti. Bu bölümde ise teste tabi birden fazla özelliği olan ürünlerin test edilmesi söz konusudur. Çok özellikli ürünlerin testi, varsayımlara bağlı olarak son derece karmaşık ve çözülmesi çok zor bir problem haline gelebilmektedir. Hesapların bu karmaşıklığı, özellik sayısı belirli bir değerin (7) üzerine çıktığında problemi çözümsüz kılmakta ve bu karmaşık yapının optimize edilerek, en iyi test stratejisini önerecek tasarımın bulunması da aynı nedenle imkansız hale gelmektedir.

Takip edecek alt bölümlerde böyle durumlar söz konusu olduğunda geçerli olan kesin hesaplar, yaklaşımlar, tahmin yöntemleri ve optimizasyon modelleri ayrıntılı olarak işlenecek, örneklenecek ve yorumlanacaktır.

Çok özellikli test tasarımı temel olarak iki ana gruba ayrılmaktadır. Sınıflandırma bağımsız test ve bağımlı test şeklinde oluşmaktadır. Bağımsız test durumunda ürünlerin test edilecek birden fazla özellikleri bağımsız olarak (ayrı ayrı) test edilecek ve bu testlerin sonuçları hep beraber nihai ürünün kalitesini belirleyecektir. Bu işlem esnasında değişik özelliklerin testleri bağımsız olarak yapıldığı için herhangi bir özelliğin testi diğerlerini etkilemeyecek ve sadece ilgili teste tabi özelliği etkileyecektir.

Bağımsız testlerin dinamikleri, hesaplamaları ve optimizasyonları, sınıflandırmanın göreceli olarak kolay olan kısmıdır. Bağımlı test söz konusu olduğunda ise hesaplar daha da karmaşılaşmaktadır. Zira böylesi bir durumda

herhangi bir özellik test edilirken başka bir özelliğin değerlerine de etki edebilmektedir.

Sözelimi bir devreye mekanik bir test uygulanırken devrenin bileşenlerini fiziksel olarak bir arada tutan kaynaklar zarar görebilmektedir. Aynı anda birkaç özelliğin test edilebildiği ve birkaç özelliğin de etki gördüğü düşünülürse, oluşan şartların karmaşıklığı daha rahat anlaşılabilir.

Aşağıda çok değişkenli test incelenmeye başlanacaktır. İlk olarak çok özellikli testin temel kavramları açıklanacak, daha sonra bağımsız test tasarımı, ardından bağımlı test kesin hesapları, bağımlı test olasılıklı yaklaşım yöntemi ve bağımlı test optimizasyonu bölümleri gelecektir.

4.1. ÇOK ÖZELLİKLİ TEST TEMEL KAVRAMLARI

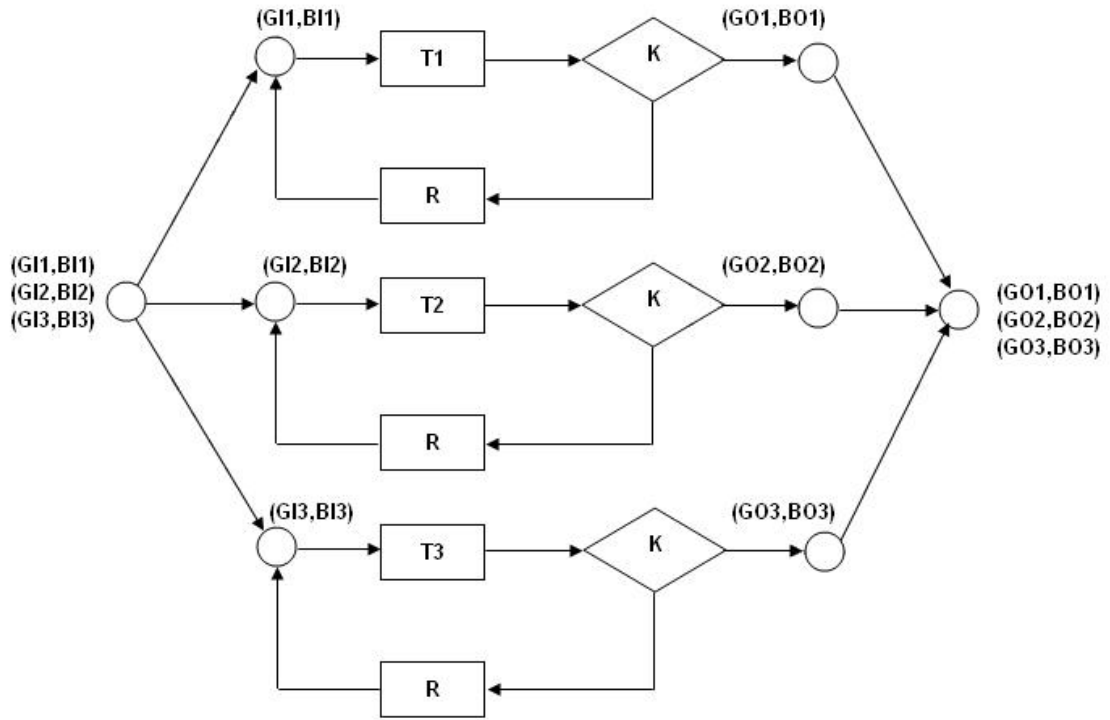
Çok özellikli test ile ilgili temel test kavramları, daha önce açıklanmış olan tek özellikli testin temel kavramlarının hepsini içermektedir. Doğal olarak test sürecinin temelleri olan daha önce verilmiş terimler ve hesaplamalardaki temel mantık, daha karmaşık boyutta olan çok özellikli test için de geçerli olacaktır.

Çok özellikli testin tek özellikli testten ayrılan ve ona ek kabul edilebilecek son derece önemli bileşenleri bulunmaktadır. Bu bileşenlerin tanımlanması için örnek durum olarak bağımsız çok özellikli bir test problemi kullanılacaktır. Bağımsız çok özellikli test problemi, hem görünüş hem de hesaplama olarak birden fazla tek özellikli test probleminin bir araya gelmiş hali gibidir.

Normal olarak bu testleri birbirleri ile kaynaştıran ek kısıtlar gerekmektedir. Bu kısıtlar haricinde ise tek özellikli test problemine oldukça benzemektedir.

Örnek bir bağımsız test akışı, aşağıdaki şekilde görülebilir.

Şekil 5. Çok Özellikli Bağımsız Test Akışı



Şekilden de görüleceği gibi, teste giren ürünün test edilecek üç özelliği bulunmaktadır. Bu üç özellik, sırasıyla T1, T2 ve T3 testlerine girmektedir. Her test diğerinden bağımsız olarak test edilmektedir. Her üç testin çıktıları ise nihai ürünü meydana getirmektedir. Yukarıdaki şekil kullanılarak çok özellikli testin etkileşim içermeyen en önemli üç temel kavramı açıklanacaktır.

4.1.1. Test Kapsaması

Çok özellikli ürün testinde test tasarımı yapılırken, tek özellikli testlerde olduğu gibi iki makine arasında seçim yapmanın ötesinde, başka özelliklere de bakılmalıdır. Test kapsamı, bu özelliklerin başında gelmektedir ve test makinelerinin özellikleri ile doğrudan ilişkilidir. Daha önce literatür bölümünde hata kapsamı kavramı detaylı şekilde incelenmişti. Test kapsamı ile hata kapsamı arasında problemi ele alış biçiminden kaynaklanan çok küçük farklar bulunmaktadır.

Hata kapsama oranında ürün üzerinde bulunan potansiyel hataların, seçilecek testler tarafından hangi oranda kapsandığına bakılmaktaydı. Bu çalışmada ise hata kapsama oranının değişik testler tarafından sağlandığı kabul edilmektedir. Tüm testlerin yapılacağı durumlarda (örneğin bağımsız testlerde) testleri gerçekleştirecek test ve tamir ekipmanının seçimi, tüm testlerin yapılmama şansı bulunan durumlarda (örneğin bağımlı test) ise hedeflenen ürün oranını sağlayacak şekilde testlerin ve testleri yapacak test ekipmanının seçilerek hata kapsama oranının kendiliğinden oluşması öngörülmektedir. Haliyle kullanılan test kapsamı terimi aslında hata kapsamı terimini de yapısı itibarıyla içermektedir. İlgili bir örnek aşağıda bulunabilecektir.

Çalışmanın varsayımlarından biri olarak test makinelerinin her türlü fonksiyonel testlerin sadece birini, birden fazlasını, ya da hepsini yapabiliyor olma olasılığı bulunmaktadır. Örneğin, yukarıda akış şeması verilen ürünün testinde kullanılma ihtimali bulunan iki makineden ilki, sadece üçüncü özelliği test ediyor olsun. Diğer bir makine de hem ilk hem de ikinci özelliği test ediyor olsun. Her iki makinenin de seçilmesi durumunda tüm özellikler test edilmiş olacaktır.

Test kapsamı işte bu test edilen özelliklerin kapsama düzeyini ifade etmede kullanılan ölçü birimidir. Mevcut örnekte test kapsamı %100 olarak tanımlanacak, yani “tam test kapsamı” söz konusu olacaktır. Oysa ikinci makine sadece ilk özelliği test ediyor olsa ve kullanılacak başka bir makine bulunmaması söz konusu olsa, ikinci özellik test edilmemiş olacaktır. Bu durumda test kapsamı %66.6 seviyesinde kalacaktır. İşte bu noktada çalışmanın hata kapsama oranı ile ilgili varsayımı devreye girmektedir. Test kapsama oranı %100 olan bir ürüne dair hata kapsama oranı da %100 kabul edilmektedir.

Verilen %66.6 örneği farazidir ve teste tabi her özelliğin eşit önemde olduğu varsayımına dayanmaktadır. Oysa iletişim sektöründe kullanılan bir panel kartı üzerinde test edilebilecek yüzlerce özellik bulunmaktadır. Şüphesizdir ki bu testlerin her biri aynı düzeyde önemli değildir. Burada önemli olan etkenler, test

edilecek özellikte rastlanan üretim kalitesi düzeyi ve test edilecek özelliğin bitmiş üründe aldığı rolün önemidir. Nasıl bir otomobilin motoru ya da göstergesi ya da vites sistemi ile bagajında bulunan bir vidanın kalitesi ürünün kalitesine aynı önemde yansımamaktaysa aynı durum birçok ürün için geçerlidir. Aslına bakılırsa, firmalar önemsiz ve yüksek yüzdelerle kusursuz parçaları test etmek için zaman harcamayabilir ve bunları test kapsamının dışında tutabilirler. Zaten literatürde de bir çok çalışma kaliteden ödün vermeden hata kapsama oranını düşürerek maliyetleri azaltma yolunu önermektedirler (Örneğin Chen ve Lau, 1998 ve özellikle de Pomeranz ve Reddy, 2003).

Böylece test kapsamı için bir alt sınır oluşturulabilir. Örneğin, “Tüm test edilecek özelliklerin %99’u kapsanacak” gibi bir prensip söz konusu olabilir. Fakat bunlar pratik hayatın problemleridir ve firmaların seçimine dayanmaktadır. Bu çalışmada varsayılan temel prensip bağımsız testler için test kapsamının %100 olması gerektiğidir. Sadece bağımlı testlerde çok yüksek kalitede bir ürün bileşeni söz konusu olduğunda, model bu özelliğin test edilmemesine izin verebilecektir.

Test kapsamı pratikte değilse de teoride %100’den yüksek bir değer de alabilir. Örneğin üç özellikli yukarıdaki ürünü test etmek için kullanılabilecek alternatif makinelerin test kapsama özelliklerinin şöyle olduğu kabul edilsin: İlk makine sadece birinci özelliği, ikinci makine sadece ikinci özelliği, üçüncü makine sadece üçüncü özelliği, dördüncü makine hem birinci hem ikinci özelliği, beşinci makine hem ikinci hem üçüncü özelliği test edebiliyor. Dördüncü ve beşinci makinelerin hem yüksek performans hem de ucuz maliyetli olduğu da varsayalım. Bu durumda daha pahalı ve düşük performanslı ilk üç makine kullanılarak %100 kapsama elde edileceğine, yüksek performanslı ve ucuz son iki makinenin tercih edilmesi daha ekonomik ve doğru bir tercih olacaktır.

Fakat bu sefer de ikinci özellik iki kez test edilmiş olacaktır. Bu durum, ürün kalitesi ve çıktılarda oluşacak olası hatalardan kaynaklanacak maliyetler

açısından son derece olumlu olmakla beraber, ikinci özelliğin iki kez test edilmesi nedeniyle değişken test maliyetlerini artırmaktadır.

Literatürde alt problemin yapısındaki modeller iki başlık altında incelenmektedir. Bunlardan ilki küme bölümlleme (*set partitioning*), ikincisi ise küme kapsama (*set covering*) problemleri olarak isimlendirilir. Küme bölümlleme problemlerinde amaç bir kümeyi hiçbir parçası açıkta kalmayacak ya da hiçbir parçası birden fazla tekrarlanmayacak alt kümelerle ayırmaktır. Yukarıdaki problemlerden örnek verilecekse $\{1,2,3\}$, $\{3,4\}$ ve $\{1,5\}$ şeklinde makine seçimleri üç özellikten her birini birer kez kapsayacak seçilmeye uygun makine seçim stratejileridir. Bu üç stratejiden en hesaplı olanı küme bölümlleme probleminin optimal çözümü olacaktır. Aynı problem küme kapsama problemi olarak ele alınsa idi her üç özelliği de en az bir kez kapsayacak her türlü strateji uygun bir nokta olacak ve bunların en hesaplısı da optimal çözüm kabul edilecekti.

Örneğin $\{1,2,3\}$, $\{1,2,3,4\}$, $\{1,2,3,4,5\}$, $\{1,2,5\}$, $\{2,3,4\}$ ve $\{4,5\}$ gibi stratejilerden her biri optimal strateji olmaya aday olacaktı. Bir varsayım olarak bu tez çalışması, gerektiği durumlarda eldeki problemi küme kapsama problemi olarak ele almaktadır. Ürünlerin testi geçmelerine rağmen tekrar test edilmesine izin verilmesi sadece kaliteyi artırmak amacıyla ya da başka özelliklerin belirli bir düzeyde test edilmesini sağlama mecburiyetinden dolayı gerçekleşebilir. Böyle bir durum için Ding ve diğerlerinin (1998) çalışması örnek teşkil edecektir.

4.1.2. Ürün Oranı

Ürün oranı aslında kusursuz ürün oranı (*yield*) anlamına gelmektedir. Tam kastedilen ise tüm testleri geçen nihai ürünlerin içindeki kusursuz ürün oranını göstermektedir. Bu oran daha önceki alt bölümlerde incelenen tek özellikli test optimizasyonu problemlerinde karşılaşılan “y” değişkenine tekabül etmektedir. Fakat çok özellikli testte kullanılan “y” (tanım amacıyla y_c olarak gösterilecektir)

ile tek özellikli testte kullanılan “y” (tanım amacıyla y_t olarak gösterilecektir) arasında ciddi farklılıklar bulunmaktadır.

Tanım gereği y_t , tek bir testin bir ya da daha fazla kez tekrarlı uygulamalarından oluşan kusursuz ürün oranı iken, y_{ϕ} çok daha farklı bir anlama sahiptir. Çok özellikli testin her bir test edilen özelliği kendi başlarına birer y_t olarak görülebilir. Her testin sonucunun, o ürün bileşeninin kusursuz olma olasılığı olarak kabul edilmesi durumunda, nihai ürünün kusursuz olma olasılığı, y_{ϕ} değeri, farklı y_t değerlerinden oluşan bir birleşik olasılık şeklinde tanımlanabilecektir. Çünkü açıkça görülebilir ki nihai ürünün hata yapma olasılığı, içindeki herhangi bir bileşenin hata yapma olasılığına bağlıdır. Bu durumda ürünün çalışma olasılığı, her bir bileşeninin çalışıyor olmasına dayanmaktadır. Böylece çok özellikli test için kusursuz nihai ürün oranı şu şekilde yazılabilir:

$$y_{\phi} = \prod_i y_i, \quad \forall i \in [1..t]$$

Yukarıdaki formül gereği, çok özellikli testte nihai ürünün başarı oranını belirli bir düzeyin yukarısında tutmak için izlenebilecek iki yol bulunmaktadır. Bunlardan ilki üstteki denklemi modelin içine yerleştirmek ve y_{ϕ} değerine bir kısıt eklemek. Örneğin, $\{ y_{\phi} > 0.95 \}$ denklemi nihai ürün başarı oranını %95’in üzerinde olmaya zorlayacaktır. Diğer yol ise, y_{ϕ} ’nin alması istenen değere göre, her bir bileşeni için kısıt koymaktır. Böylece üç özellikli bir ürün için, örneğin, her i için $y_i \geq 0.98$ gibi bir kısıt eklenirse, elde edilecek kusursuz çıktı oranının olası en düşük değeri aşağıdaki gibi olacaktır:

$$y_{\phi} = \prod_i y_i = 0.98 * 0.98 * 0.98 = \%94.12$$

Bu yol izlenirse optimizasyonun doğru yapılması için üstteki denkleme de gerek kalmayacaktır. Denklem sadece kusursuz çıktı oranını göstermek için hesaplatılacaktır. Hangi yolun seçileceği araştırmacının tercihine kalmıştır. Bu tercihi etkileyecek etkenler, her bir bileşenin önemi, olası çok kritik bileşenlerde

ulaşılmak istenen yüksek başarımlar oranları, hatası kolay fark edilen ve başarımlar oranı çok kritik olmayan basit bileşenler ve optimizasyon modelinde fazladan bir doğrusal olmayan kısıt kullanımı ile başa çıkılabilme durumu olacaktır. Son etken özellikle önemlidir çünkü ($y_c = \prod_i y_i$) kısıtı doğrusal olmayan bir kısıttır.

4.1.3. Durum Vektörü ve Kitle Dağılımı

Durum vektörü, hesap detaylarına daha sonra değinilecek olan bir kavramdır. Basit bir şekilde, herhangi bir ürünün test döngüsü içindeki herhangi bir andaki durumunu tanımlamaktadır. Fisher ve diğerleri (2007a), durum vektörü konusunda ayrıntılı tanımlar vermektedirler. Onların tanımıyla durum vektörü aşağıdaki gibi irdelenmektedir:

Ürünün teste tabi k özelliği bulunduğu ve bu özelliklerin f_j ile gösterildiği varsayılırsa, k özellikten her birinin durumu, f_j , iyi ya da kötü (kusursuz ya da kusurlu, sağlam ya da bozuk) olarak tanımlanabilir. Bir nihai ürünün problemsiz çalışması ise içindeki k bileşenin her birinin iyi durumda olmasına bağlıdır. Herhangi bir i durumunda ürünün durumunu gösteren G_i vektörü, k tekrarlı $g_{i,j}$ değerlerinden oluşmaktadır. Her bir $g_{i,j}$ değeri, i durumunda f_j özelliği iyi ise 1, kötü ise 0 değeri almaktadır. Böylece durum vektörü şöyle gösterilebilir:

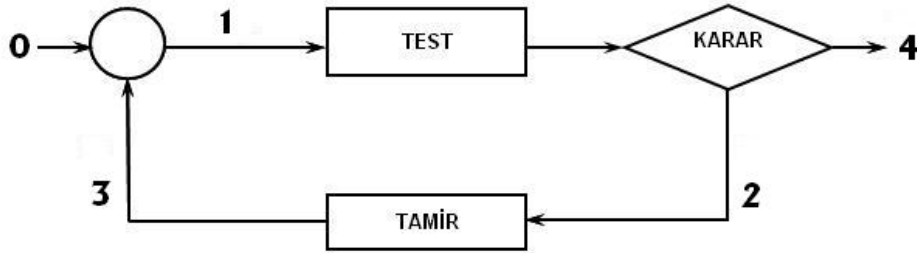
$$G_i = [g_{i,1}, g_{i,2}, \dots, g_{i,k}]$$

Örneğin, beş özellikli ve ikinci ve dördüncü özelliği bozuk olan bir ürünün üçüncü durumdaki durum vektörü $G_3 = [1, 0, 1, 0, 1]$ şeklinde değer alacaktır.

Buradan da görülmektedir ki her özellik iki değer alabileceğinden, k özellikli bir ürünün herhangi bir zamandaki durum vektörü, $m = 2^k$ alternatif değer alabilecektir (Rossi ve diğerleri, 2006 ve Fisher ve diğerleri, 2007a). Önemli olan, test akışı esnasında herhangi bir anda bu değişik vektöre sahip ürünlerden kaç adet olacağıdır.

Şekil 6 bazı kavramların açıklanmasında yardımcı olması amacıyla kullanılacaktır.

Şekil 6. Temel Test Akış Şeması



Yeniden Fisher ve diğerlerinin (2007a) çalışmasındaki terminoloji kullanılacak olursa, N_i^s değişkeni s oku üzerinde bulunan m tane G_i alternatifinden i sıra numaralı durum vektörüne sahip ürünlerin sayısını ifade etmektedir. Böylece her bir N_i^s değişkeni, durum kitlesi ya da kitle olarak isimlendirilmektedir. Kitle dağılımı ise m tane N_i^s değişkeninden oluşan bir vektör olarak tanımlanacaktır. Matematiksel gösterimi ise şöyledir:

$$P_s = [N_1^s, N_2^s, \dots, N_m^s]$$

Bu ifade en basit anlatımla, her bir durum birleşiminden kaçar adet bulunduğunu göstermektedir. Örneğin sayısı 100 olan girdilerin tek özelliklerinin bulunduğu ve 80 tanesinin kusursuz olduğu bir durumda gerçekleşecek değerler şöyle olacaktır:

$$k = 1 \quad f_i = [f_1]$$

$$g_{1,1} = 1 \quad g_{2,1} = 0$$

$$G_1 = [1] \quad G_2 = [0]$$

$$N_1^0 = 80 \quad N_2^0 = 20$$

$$P^0 = [80, 20]$$

İki özellikli bir ürünün testinde ise bu değerler varsayımsal rakamlarla şu hali alacaktır:

$$\begin{aligned}
 k &= 2 & f_i &= [f_1, f_2] \\
 g_{1,1} &= 1 & g_{1,1} &= 1 & g_{2,1} &= 1 & g_{2,2} &= 0 \\
 g_{3,1} &= 0 & g_{3,2} &= 1 & g_{4,1} &= 0 & g_{4,2} &= 0 \\
 G_1 &= [1,1] & G_2 &= [1,0] & G_3 &= [0,1] & G_4 &= [0,0] \\
 N_1^0 &= 75 & N_2^0 &= 15 & N_3^0 &= 8 & N_4^0 &= 2 \\
 P^0 &= [75,15,8,2]
 \end{aligned}$$

Bu değerlerin anlamı, teste giren ürünlerin 75 tanesi kusursuz, 15 tanesinin ikinci özelliği kusurlu, 8 tanesinin ilk özelliği kusurlu ve 2 tanesinin de her iki özelliği kusurludur.

4.2. ÇOK ÖZELLİKLİ BAĞIMSIZ TEST

Burada incelenecek olan çok özellikli bağımsız test, bir önceki bölümde incelenen tek özellikli test ile ilgili kısmın biraz daha karmaşılaştırılmış halidir. Çok özellikli bağımsız test yerine “Çok Özellikli Seri Test” ifadesi de bu tip testleri açıklama amacıyla kullanılabilir. Anlatılmak istenen farklı testlerin birbirlerini etkilemeden teste girmeleridir. Buradaki problemin yapısı Butterworth (1972) tarafından tanımlanan problemi de anımsatmaktadır.

Şekil 5'ten de hatırlanacağı gibi, bağımsız test durumunda test edilecek özelliklerin her biri, tek özellikli test gibi davranmaktadır. Fakat bu testlerin kusursuz çıktıları, birleşik olasılık mantığıyla nihai ürünün kusursuz olma olasılığını oluşturmaktadır. Dolayısı ile tek testlerin her birine ayrı olarak ya da, ürün oranı üzerine bir verim kısıtı eklenerek bağımsız testlerin test tasarımı ve nihai ürünün kalitesi üzerine problem modellemesi yapma fırsatı bulunabilmektedir.

Bu bölümde yapılacak modellemelere altyapı oluşturacak optimizasyon modeli, tek özellikli test kısmında incelenen kısıt programlama modelidir. Kısıt programlama modeli bu bölümde her bir farklı özellik için çoğaltılacak ve gerekli yeni kısıtlar eklenerek çok özellikli bağımsız test problemine çözüm aranacaktır. Bahsedilen model aşağıdaki şekildedir.

Minimize

$$VcR * (GS[1] + BS[1]) + Fc * BI[1] + \sum_i VcT_i * (GT_i[1] + BT_i[1]) + \sum_i FcT_i * use[t_i, 1]$$

Subject to

$$GI[0] = 80$$

$$BI[0] = 20$$

$$GI[t] = \alpha_{TP,T1}[t] * GT_1[t] + \alpha_{TP,T2}[t] * GT_2[t]$$

$$GS[t] + GI[t-1] = GT_1[t] + GT_2[t]$$

$$GR[t] = \alpha_{FF,T1}[t] * GT_1[t] + \alpha_{FF,T2}[t] * GT_2[t]$$

$$GS[t] = \beta_{GG} * GR[t] + \beta_{BG} * BR[t]$$

$$BI[t] = \alpha_{FP,T1}[t] * BT_1[t] + \alpha_{FP,T2}[t] * BT_2[t]$$

$$BS[t] + BI[t-1] = BT_1[t] + BT_2[t]$$

$$BR[t] = \alpha_{TF,T1}[t] * BT_1[t] + \alpha_{TF,T2}[t] * BT_2[t]$$

$$BS[t] = \beta_{GB} * GR[t] + \beta_{BB} * BR[t]$$

$$GI[t] + BI[t] = GI[t-1] + BI[t-1]$$

$$GT_i[t] + BT_i[t] \leq cap_{Ti} * use[T_i, t], \forall i \in (1, 2)$$

$$\sum_i use[i, t] \leq nm$$

$$GI[nbt] \geq (GI[0] + BI[0]) * y$$

$$use[T_i, t] > 0 \Rightarrow 100 * \frac{(GI[t-1]-3)}{(GI[t-1] + BI[t-1])} \leq 100 * \frac{GT_i[t]}{(GT_i[t] + BT_i[t])} \leq 100 * \frac{(GI[t-1]+3)}{(GI[t-1] + BI[t-1])}$$

$$\forall i \in (1, 2)$$

Bu problemin ürün girdi kitlesi, artık iyi ve kötü ürün sayısını (ya da toplamı) 100 ise aynı zamanda da olasılığı gösteren bir vektör değil, özellik sayısı ile

ilintili bir matris olacaktır. Başka bir deyişle hem iyi hem kötü ürünler ayrı birer vektör olacaktır. Makine sayısı ve ürün oranı kısıtları hariç tüm kısıtlar her test için çoğaltılacak ve bu testlerin çıktıları ürün oranının birleşik olasılık formülü ile birbirlerine bağlanacaktır. Alternatif yol olarak da birleşik olasılık yerine her bir testin çıktısı ayrı olarak belirli bir düzeyin üstünde olmaya zorlanacaktır.

Probleme her makinenin, birden fazla testi yapabiliyorsa, o testteki performans bilgileri (α ve β değerleri) girilmeli, makine herhangi bir özelliğin testini yapamıyorsa o testteki performansı ile ilgili test kapsamı durumu dikkate alınmalı ve yeni kısıtlar eklenmelidir. Öncelikle bir örnek üzerinden iki özellikli bir ürünün seri testi incelenecektir. Mevcut örnekte iki alternatif test makinesi her iki özelliği de test edebilecek niteliklere sahiptir. Problemin girdilerinin yapısı aşağıdaki tabloda olup örnek bir problem için farazi değerler girilmiştir. İndekslerde gözlenen son rakamlar, değerin 2 özellik testinden hangisi için geçerli olduğunu göstermektedir. Her iki özellik için de girdilerin yarısı kusursuzdur.

Tablo 9. Çok Değişkenli Bağımsız Test Modeli Girdileri

Hatasız Kabul Olasılığı $\alpha_{tp,t1,1} = 0.75$ $\alpha_{tp,t2,1} = 0.90$ $\alpha_{tp,t1,2} = 0.85$ $\alpha_{tp,t2,2} = 0.90$	Hatasız Ret Olasılığı $\alpha_{ff,t1,1} = 0.75$ $\alpha_{ff,t2,1} = 0.90$ $\alpha_{ff,t1,2} = 0.95$ $\alpha_{ff,t2,2} = 0.90$	Sabit Maliyetler $F_{ct1} = 200000$ $F_{ct2} = 100000$
Hatalı Ret Olasılığı $\alpha_{ff,t1,1} = 0.25$ $\alpha_{ff,t2,1} = 0.10$ $\alpha_{ff,t1,2} = 0.15$ $\alpha_{ff,t2,2} = 0.10$	Tamir Parametreleri $\beta_{bb} = 0.2$ $\beta_{bg} = 0.8$ $\beta_{gb} = 0.2$ $\beta_{gg} = 0.8$	Test Kapasiteleri $Cap_{t11} = 50$ $Cap_{t21} = 50$ $Cap_{t12} = 50$ $Cap_{t22} = 50$
Hatalı Kabul Olasılığı $\alpha_{fp,t1,1} = 0.25$ $\alpha_{fp,t2,1} = 0.10$ $\alpha_{fp,t1,2} = 0.05$ $\alpha_{fp,t2,2} = 0.10$	Değişken Maliyetler $V_{ct1,1} = 200$ $V_{ct2,1} = 100$ $V_{cs,1} = 300$ $V_{ct1,2} = 100$ $V_{ct2,2} = 100$ $V_{cs,2} = 100$	Kaçış maliyeti, Ürün Oranı ve En Fazla Makine Sayısı $F_c = 100$ $y = 0.80$ $nm = 50$

Problemin modellenmesinde ise yukarıda verilen ve her test için indekslenerek çoğaltılan kısıtlara ek olarak sadece çıktı ile ilgili kısıtlar gelmektedir. Bu kısıtlar

hem birleşik olasılık ile ürün oranını hesaplamakta, hem de oranı önceden belirlenmiş bir değerin (bu problem için örneğin, $y = 0.80$) üstünde olmaya zorlamaktadır.

$$GO = \prod_i GI_{1*i}$$

$$\frac{GO}{10000} \geq y$$

Yukarıda verilen problem çözüldüğünde bulunan çözüm aşağıdaki gibidir:

Tablo 10. Çok Aşamalı Çözüm

Solution with Objective Value: 1474735.0000
Solution with Objective Value: 1474648.0000
Solution with Objective Value: 1173822.0000
Solution with Objective Value: 1173735.0000
Optimal Solution with Objective Value: 1173735.0000
use[t1,1] = 2
use[t2,1] = 2
use[t1,2] = 1
use[t2,2] = 3
GO = 8265

Çözümünden de görülmektedir ki ilk testte her iki makineden ikişer tane, ikinci testte ise ilk makineden bir, ikinci makineden üç tane kullanılmaktadır. Kaliteli çıktı oranı %82.65'te kalmıştır. Çözücü, ilk etapta 1,474,735 birimlik toplam maliyete sahip bir uygun çözüm elde etmiş ve ancak altıncı aşamada en iyi sonucu elde edebilmiştir. En iyi çözüme ulaşıldığında 1,173,735 birimlik toplam maliyet oluşmuştur. Bu maliyet tek özellikli test probleminin kısıt programlama ile çözüldüğünde bulunan 427,000 birimlik maliyetin 2.5 katından biraz fazladır. Yapılan denemelerde görülmüştür ki mevcut makineler ve ürün özellikleri ile

ürün kalitesinin %89'a kadar çıkarılma olanağı bulunmaktadır. Böyle bir durumda çözüm aşağıdaki gibidir:

Tablo 11. Kısıtlı Çözüm

Optimal Solution with Objective Value: 1479072.0000
use1[t1,1] = 2
use1[t2,1] = 2
use2[t1,1] = 3
use2[t2,1] = 2
GO = 8928

Modele çıktı oranı üzerinden yükleme yapıldığı takdirde, kaliteli ürün oranının %89.28'e kadar artırılabilirdiği görülmektedir. Böyle bir durumda bir önceki çözümden farklı olarak toplamda 8 yerine 9 makine kullanılmıştır. Seçilen makine düzeni ilk test için aynı kalırken, ikinci testte ciddi farklılıklar göstermektedir. Buradan anlaşılmaktadır ki, toplam test performansı için kritik olan test özelliği, ikinci özellik yani ikinci testtir. Elde edilen yaklaşık %6.63'lük kalite artırımı, maliyet cinsinden yaklaşık %26'lık bir ek masraf ile karşılanabilmektedir. Ayrıca not edilmelidir ki, bu çıktı oranı, mevcut makinelerin herhangi birleşimi ile elde edilebilecek en iyi orandır. Başka bir deyişle $y = 0.90$ gibi bir değer kısıt olarak belirlendiği takdirde, çözücü sadece optimal çözüme değil, herhangi bir kullanılabilir çözüme bile ulaşamamaktadır.

Oysa makineler farklı olursa değişik sonuçlar elde edilebilecektir. Genel bir kural olarak girdilerdeki iyileştirmeler ya amaç fonksiyonuna maliyet tasarrufu olarak yansımali ya da artırılması tercih edilecek değişkenler üzerinde olumlu etkilerde bulunmalıdır. Test problemi örneğinden yola çıkılacak olursa, en iyi örnek kaçınılmaz olarak kusursuz çıktı oranının artması şeklinde görülmektedir. Bu duruma basit bir örnek aşağıda verilmektedir.

Örnek olarak bir önceki problemin girdilerinde sadece T_1 test makinesinin ilk test için bozuk ürünlerde gösterdiği performansını sadece %10 değiştirilecektir. Böylece $\alpha_{tf,t1,1}$ değişkeni 0.75 yerine 0.85, ve $\alpha_{fp,t1,1}$ değişkeni 0.25 yerine 0.15 olacaktır. Bu minik değişimin sonuçlara yansımaları aşağıdaki şekilde olacaktır.

Tablo 12. Modifiye Çözüm

Optimal Solution with Objective Value: 1263872.0000
 $use1[t1,1] = 1$
 $use1[t2,1] = 3$
 $use2[t1,1] = 3$
 $use2[t2,1] = 1$
 $GO = 8928$

Sonuçtan da görüleceği gibi kusursuz çıktı oranı değişmemiştir. Fakat makine seçimlerinde radikal bir değişiklik ortaya çıkmış ve farklı makine seçimi ile maliyetler yaklaşık %14.6 oranında azaltılmıştır.

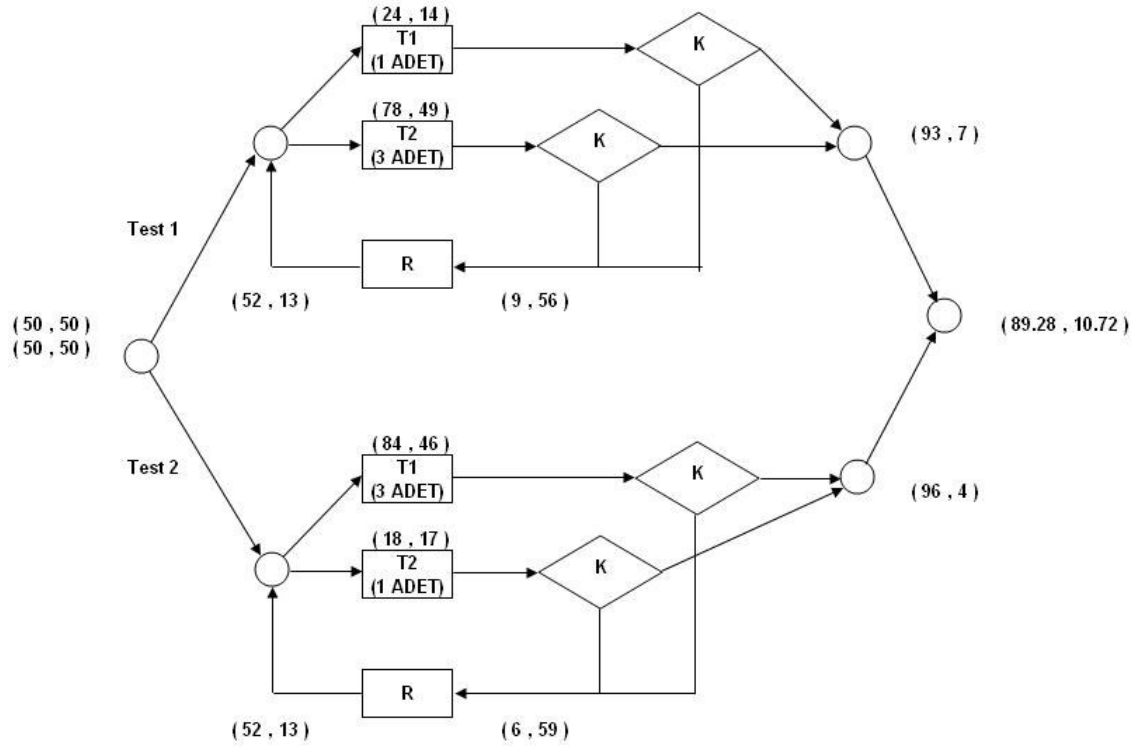
Bu problemin önemli noktaları grafik gösterim ile de sunulacaktır. Son örnekte aktarılan, 1263872 maliyetli ve %89.28'lik kusursuz çıktı oranına sahip çözümün sadece toplam değerlerinin verildiği grafik çözüm, aşağıdaki şekilde incelenebilir.

Çözümlerde Şekil 7'de de görüleceği üzere, ağır tamsayılaştırılmış kısıtlar yerine hafif akış orantılama kısıtları kullanılmasına rağmen 4 test makinesi tipinde de çok yakın kusurlu kusursuz ürün oranı dağılımı dikkat çekmektedir. Bu değerler sırasıyla (%63.15, %61.42, %64.62, %51.43)¹ olarak hesaplanmaktadır. Son makinede görülen %10 civarı fark da önceki bölümde anlatıldığı şekilde modelin makine verimliliğini de dikkate almasından kaynaklanmakta ve son makinenin sadece kalan ürünleri test ediyor olmasına

¹ Örneğin şekilde ilk makinede görülen değerlerden $24 / (24+14) = 0.6315$ değeri kolayca hesaplanabilir.

dayanmaktadır. Bu şartlar test tasarımının rastsal bir yapıda test sürecini işlettiği rahatlıkla söylenebilecektir.

Şekil 7. İki Özellikli Seri Test – Grafik Çözüm



Aynı problemde tamsayılaştırılmış ağır akış oranı kısıtları kullanıldığı takdirde, yani makine verimini göz ardı ederek belirli bir hata payı içerecek şekilde tam rastsal ürün akışı tercih edilir ise durum biraz daha farklı olacaktır. Böylesi bir durumda problemin çözüm süreci çok karmaşıklaşmakta ve hafif akış kısıtları kullanılırken saniyenin yüzdeleri ile ifade edilebilecek sürede çözülen problem, tamsayı kısıtlar altında optimal sonuca ulaşmakta oldukça zorlanmakta, çoğu kez süreci tamamlayamamaktadır.

Problem son örnekte verilen girdilerin aynısı ile çözdürüldüğünde ILOG OPL Studio programı 0.5 saniye içinde (genellikle 0.29 – 0.43 saniye arası) uygun bir çözüm bulmakta fakat daha sonra optimal sonuca ulaşmak için aramaya devam ettiğinde uzunca bir süre mevcuttan daha iyi bir amaç fonksiyonu değerine

ulaşamamaktadır. Programın problemi çözerken bulduğu ilk uygun çözümde maliyet olarak 384,562 birim ve makine seçimi ilk test için bir adet T_1 ve üç adet T_2 gözlenmektedir.

Bu noktada çözümü hızlandırmak amacıyla yaygın kullanılan bir yöntem olarak kesmeler kullanılmaya başlanmıştır. Basit olarak kesmeler, amaç fonksiyonunu bir kısıt olarak modelin içine eklemek ve yine modelin içine amaç fonksiyonunu istenen değerlerden kesebilecek kısıtlar atamaktır. Modelin öncelikle amaç fonksiyonu Z değişkenine değiştirilmeli, daha sonra Z değeri model içinde amaç fonksiyonuna atanmalı ve A değeri istenen kesme olacak şekilde, aşağıdaki kısıtlar eklenirse bahsedilen işlem tamamlanmış olacaktır.

$$Z = \sum_i \left(VcR * (GS[1,t] + BS[1,t]) + Fc * BI[1] + \sum_i VcT_{i,t} * (GT_i[1,t] + BT_i[1,t]) + \sum_i FcT_i * use[t_i,1,t] \right)$$

$$Z \leq A$$

İlk bulunan sonucun amaç fonksiyonundan daha düşük bir değer A değişkenine atanırsa, bu bir kesme kısıtı olacak ve program artık bulduğu bir önceki sonuç ya da daha kötü sonuçları dikkate almadan daha iyi bir çözüm aramaya zorlanacaktır. Örneğin ilk uygun sonuçtan sonra $A = 350,000$ olarak saptanarak model yeniden çalıştırılırsa amaç fonksiyonu 296,340 olarak bulunacak ve kusursuz çıktı oranı da %94.87 düzeyine gelecektir. OPL programı bu sonuca yaklaşık 4.55 – 4.65 saniye arasında ulaşmakta ve sonrasında gelen arama süreci uzun ve hareketsiz bir bekleme olmaktadır.

Bu süreç istenen problem için tatmin edici bir sonuç elde edilene ya da çözümsüz kalınana dek tekrarlanabilir. İncelenen örnek problemde çözümsüz kalma sınırı, amaç fonksiyonunun yaklaşık olarak 274,916 değerinin altında kalan değerlerdir. Problemin optimal çözümü açısından bakıldığında bu değer [274,916 – 296,340] kapalı kümesi içinde bulunmaktadır. Zira 296,340 değeri optimal olabileceği gibi yukarıda belirtilen aralık içinde bulunan herhangi bir değer de optimal amaç fonksiyonu değeri olabilecektir. Buradaki problem

kesmeler kullanarak daha iyi bir optimal çözüm olup olmadığını kontrol etmede geçecek sürenin kaybedilme ya da kazanılma tercihindedir. Örneğin yukarıdaki problemde optimal amaç fonksiyonu aralığı bellidir ve aralığın sağ ucunda gerçekten uygun ve optimal çözüm adayı bir değer bulunmaktadır. Aralığın geri kalanı ve sol ucunda bulunan değerlerin çözüm içerip içermediği ispatı ile vakit kaybedilmek istenmediği varsayıldığına göre, araştırmacı ya da firma 296,340 maliyetli çözümü kullanmaya kalkarsa yapabileceği maksimum hata %7.7 yüksek maliyetle faaliyet yapmak şeklinde gerçekleşecektir. Belirtmelidir ki en fazla %7.7 yüksek maliyet ifadesi de kesinlik içermemekte, sadece alınacak risk düzeyinin varsayımsal izdüşümü görevi görmektedir.

Özetle bu alt bölümde çok özellikli testlerin bağımsız olarak yapılması durumunda ortaya çıkacak dinamikleri, hesaplanan akış oranlarının oluşturduğu rastsallığın ihlali durumunu, getirilebilecek hafif ya da ağır kısıtlar aracılığıyla bu ihlalin aşılması, kapasite kullanımı yüksek makinelerde rastsallık ya da tüm sistem içinde belirli bir hata payı içerisinde rastsallık zorlanması, ağır kısıtlar durumunda yaklaşık optimal sonuçların elde edilmesi için kesme kısıtları ve oluşacak farkın tanımı ile birlikte bağımsız seri testlerin incelenmesi tamamlanmaktadır. Aşağıdaki diğer alt bölümler çok özellikli testin etkileşimli durumunu incelemeyi amaçlamaktadır.

4.3. ÇOK ÖZELLİKLİ ETKİLEŞİMLİ TEST

Birden fazla test edilecek özelliği bulunan ürünler söz konusu olduğunda test süreçleri her zaman birbirinden bağımsız olamayabilecektir. Özellikle fiziksel temas, zorlama, bir sistemin test edilmesi, fonksiyon test edilmesi gibi durumlarda test işlemi, test edilen özellik dışında başka özellikleri de etkileyebilecektir. Böyle bir duruma Chen ve Lau (1998) değinmekte ve belirli test şartları sağlanırken bazı testlerin hem aynı özelliği test ederek gereksiz tekrar yaratabileceğini hem de bu testler sırasında ürün özelliklerinin birbirlerinden etkilenebileceklerini belirtmektedirler.

Bir kartın testinden örnek vermek gerekirse, kartın üzerinde bulunan lehimler, çipler ve bağlantılar aynı anda test edilecektir. Test makinesi her özelliğin testi için hata payı taşıyan olumlu ya da olumsuz bir karar verecektir. Fakat verilen karar ve ürünün gerçek durumu arasındaki ilişki, tek özellikli test ya da bağımsız çok özellikli test gibi belirli sınırlar içinde değildir. Örneğin, tek özellikli testte, testi geçmiş ve gerçekten kusursuz bir ürünün olasılığı belirlidir. Aynı mantıkla çok özellikli bağımsız test durumunda da her bir özellik için belirlidir. Böylece birleşik olasılık formülü ile hatasız çalışabilecek minimum ürün oranı öngörülebilir.

Bu fenomeni durum vektörleri açıklayabilmektedir. Tek özellikli testlerde, ürünün durumunu iyi ve kötü ürünler için tek boyutlu birer vektör yani iki sayı göstermektedir. Bu değerlerin girdiden çıktıya dönüşme sürecinde ise her bir sayı test ve tamir makinelerinin özellikleri doğrultusunda değişikliğe uğramaktadır. Çok özellikli bağımsız test durumunda ise durum vektörleri hem iyi hem de kötü ürünler için özellik sayısı kadar eleman içeren birer vektör haline gelmektedirler. Bağımsız testin yapısı gereği bu vektörlerin aynı özelliğe denk gelen elemanları birbirlerinden bağımsız olarak değişmekte ve test süreci bittikten sonra birleşik olasılık formülü ile bir araya gelmektedirler. Yani çok boyutlu bir işlem ya da etkileşim söz konusu değildir. Fakat çok özellikli etkileşimli test söz konusu olduğu zaman en az bir test makinesi, test ettiği ürünün birden fazla özelliğini değiştireceği için durum vektörünün elemanları ayrı olarak değil, aynı anda değişmek durumunda kalacaklardır. Bir bakıma, bu noktaya kadar çok ciddi bir öneme sahip olmayan durum vektörleri artık gereklidirler ve hayati rol oynamaktadırlar. Zira tek özelliğin değişimi söz konusu olduğunda girdilerin kitle dağılımı sadece kusurlu – kusursuz ($G_I - B_I$) şeklinde olacak, durum vektörleri de, örneğin $[73] - [17]$ şeklinde olacaktır. Her ne kadar bu iki değer iki durumu gösterse de tek değişken bulunmasından dolayı karmaşık işlemler gerekmemektedir. Zaten ürün ya iyi ya da kötüdür.

Fakat birden fazla özellik söz konusu ise bazı özellikler kusursuz, bazıları da kusurlu iken farklı ürünler testi değişik miktarlarda geçebilecektir. Temel kavramlar kısmında da anlatıldığı üzere bir ürünün her bir özelliği iyi ya da kötü durumda olabilir. Böylece test sürecinin her safhasında teste tabi ürünlerin içinde bulunabilecekleri $m = 2^k$ farklı durum, haliyle m adet farklı durum vektörü bulunmaktadır. Problemde tek özellik söz konusu iken $m = 2^1 = 2$ olmakta ve hesaplar kolaylıkla yapılabilen iken üç özellikli bir ürün için $m = 2^3 = 8$ durum, yedi özellikli bir ürün için ise $m = 2^7 = 128$ durum söz konusudur. Problem girdi gereksinimlerine bakıldığında bu durum zorluklar yaratmaktadır. Her bir özelliğin sağlam olma olasılığı bilinebilse bile birkaç özellik sorunsuz iken spesifik herhangi bir özelliğin kusurlu olma olasılığı ve benzeri bilgileri elde edebilmek oldukça zorlayıcı olacaktır. Ayrıca sadece test akışının ve her aşamada ürünlerin tek seferde ya da tüm test boyunca oluşturdukları dağılımı hesaplamak özellik sayısı arttıkça oldukça zor ve karmaşık hale gelmektedir. Eric Fisher ve diğerleri (2007a) bu hesaplamaları gündeme getirmiş ve duruma açıklık kazandırmışlardır.

Fisher ve diğerlerinin çalışması (2007a) bu alanda yapılmış ilk çalışma olmasa da durum vektörleri ve dolayısı ile test akış hesaplamalarına, bu çalışmada değinildiği anlamıyla, ilk açıklık getiren çalışmalardan biridir.

Durum vektörlerinin, test akışının her aşamasında hesaplanması gerektiği daha önce ifade edilmişti. Fisher ve diğerleri (2007a) test akışı probleminin üç girdisini tanımlamışlardır. Bunlar test makinesi ve tamir makinesi performans parametreleri ile girdi kitle dağılımından ibarettir.

Girdi kitle dağılımı, k özelliğe sahip bir ürün için $m = 2^k$ adet olması muhtemel durumun her biri için girdi kitlesinde bulunan eleman sayısından oluşan bir vektördür. Örneğin, “1” kusursuz, “0” da kusurlu özelliği gösterecek şekilde, dört özellikli bir ürün için karşılaşılabilecek durumlar şunlardır: $\{[1,1,1,1], [1,1,1,0], [1,1,0,1], \dots, [0,0,1,0], [0,0,0,1], [0,0,0,0]\}$. Yani 1 ve 0 değerlerinin her türlü dördü kombinasyonu, girdilerin alabileceği her alternatif durumu tanımlamakta

ve bu yüzden, binom açılımı nedeniyle, 2^k durum alternatifi bulunmaktadır. Her bir durum vektörü bir G_i , ve girdi kitlesinde her bir durum vektöründe bulunan eleman sayısı bir N_i^0 , ve girdi kitle dağılımı da N_i^0 değerlerinden oluşan bir P^0 vektörü olarak tanımlanacaktır. Hesabı basitleştirme amacıyla yazarlar, her bir durumun olasılık ya da sıklığını vermek yerine, her bir özelliğin kusursuz olma olasılığını kullanarak durumların olasılık ya da sıklıklarını hesaplama yöntemini önermektedirler.

Buna göre herhangi bir durumda bulunan girdi ürün sayısı aşağıdaki gibi olacak ve olasılık değeri ise bu rakamın kitle boyutuyla (F_p) çarpılmamış halinden elde edilecektir.

$$N_i^0 = F_p * \prod_{g_{i,j}=1} p_j * \prod_{g_{i,j}=0} (1 - p_j)$$

Bu duruma bir örnek vermek gerekirse: Bir ürünün 100 adet girdisi ve üç özelliği olduğu ve G_3 durumunda sadece üçüncü özelliğinin kusurlu olduğu varsayalım. İlk özelliğin kusursuz ürün olasılığı %90, ikinci ürünün %85 ve üçüncü ürünün %95 ise, G_3 durumunun girdi kitlesinde görülme sıklığı şöyle bulunacaktır:

$$N_3^0 = 100 * [0.90 * 0.85 * (1 - 0.95)] = 3.825$$

Bu durum olasılık cinsinden ise %3.825 olarak gösterilecek ve sıklık, tamsayı değişkenler kullanılıyorsa program ya da araştırmacı tarafından tamsayıya yuvarlanacaktır.

Test akışının hesaplanabilmesi için girdi ürün kitle dağılımı dışında iki önemli problem girdisi daha bulunmaktadır. Bunlar test ve tamir makinelerinin performans parametreleridir. Fisher ve diğerleri (2007a) makine performans parametrelerini de pratikte bu çalışmada verildiği şekilde, fakat ifade olarak biraz farklı tanımlamışlardır. Örneğin akış şemasında (Şekil 6) bir test noktasından testi geçerek çıkan ürünleri (Durum Vektörü ve Kitle Dağılımı

kısımındaki şekildeki 4 numaralı ok) 1 ve geçemeyen ürünleri (Durum Vektörü ve Kitle Dağılımı kısmındaki şekildeki 2 numaralı ok) 2 numaralı varsayımsal oklara göndermektedirler. Test makinesi performans parametrelerini (α 'ları), $\alpha_i^j(S)$ olarak tanımlamaktadırlar ve bu parametreler için isabetlilik parametreleri ifadesini kullanmaktadırlar.

Belirtilen parametre G_i durumundaki ürünün, S test makinesinde test edilir ise j okundan çıkma olasılığını göstermektedir. Örneğin, bu çalışmada kullanılan gösterim ile $\alpha_{TP, T1}$ parametresi $\alpha_i^1(T1)$ şeklinde gösterilmektedir. Çalışmanın bu noktaya kadar olan kısmında durum vektörleri aktif rol oynamadıklarından dolayı performans gösterimleri kapsamında kendilerine ilk kez yer bulmakta gibi görünseler de aslında kapalı bir ifade ile durum vektörleri önceden de tanımlanmıştır. TP, yani testi geçme kararının doğru olması durumu, teste giren ürünün durumunun kusursuz olduğunu belirtmektedir. Başka bir deyişle tek özellik temelinde özelliğin "1" olduğu durum için TP (doğru geçiş) ve FF (hatalı kalma) kararları kullanılmakta ve durum vektörü hakkında fikir vermektedir.

Durum olasılıklarına benzer bir hesap mantığı test makinesi parametreleri için de geçerlidir. Problem girdisi olarak her bir test makinesi S_u için, eğer S_u makinesi a özelliğini test ediyor ise $q_{u,a}$ yani a özelliği kusursuz olan ürünün testi geçme olasılığı ve $q'_{u,a}$ yani a özelliği kusurlu ürünlerin testi geçme olasılıkları tanımlanmalıdır. Bu değerler basitçe TP ve FP değerlerine denk gelmektedirler. TF ve FF değerleri de bu iki değerden basitçe hesaplanabilecektir. S_u test makinesi a özelliğini test etmiyor ise, $q_{u,a} = 0$ değeri verilerek bu durum matematiksel hesaba yansıtılabilir. Mevcut şartlar altında herhangi bir çok özellikli ürünün testi geçme olasılığı şöyle tanımlanacaktır:

$$\alpha_i^1(S_u) = \left(\prod_{q_{u,a} > 0, g_{i,a} = 1} q_{u,a} \right) * \left(\prod_{q_{u,a} > 0, g_{i,a} = 0} q'_{u,a} \right)$$

Örneğin, yukarıdaki örnekte incelenen ürünün özelliklerini sırayla, kusursuz iseler 0.9, 0.9 ve 0.8 ile geçiren, kusurlu iseler 0.05, 0.05, ve 0.1 ile geçiren bir

T_2 test makinesinin kullanıldığı varsayılın. Ürünlerden G_3 durumunda bulunanların testi geçecek kısmının oranı (ya da testi geçme olasılığı) şöyle bulunabilir:

$$\alpha_3^1(T_2) = 0.9 * 0.9 * 0.1 = 0.081$$

Böylece tüm ürünler içinde G_3 durumunda olup testi geçecek bozuk ürün olasılığı $(0.081) * (0.03825) = 0.0031$ ve toplamda testi geçecek ürün sayısı da 0.31 adet olacaktır. Başka bir deyişle, T_2 makinesi kullanılırsa üçüncü özelliği bozuk bir ürünün testi geçme olasılığı %0.31 olacaktır. Tüm özellikleri kusursuz bir ürün için oluşacak α değeri, $(0.9)*(0.9)*(0.8) = 0.648$ olacak ve kitle içinde $(0.9)*(0.85)*(0.95) = 0.727$ oranında bulunan bu kusursuz ürünlerin testi geçen kısmı, toplam teste giren ürünlerin $(0.727)*(0.648) = 0.471$ 'lik kısmına denk gelecektir. Yani toplam ürünlerin %47.1'i kusursuz ürün olarak testi geçmiş olacaktır.

Tamir makinesi parametreleri de Fisher ve diğerlerinin (2007a) çalışmasında $\beta_{i,j}(R)$ şeklinde gösterilmektedir. β değerleri bu çalışmada tamir makinesinin ürünleri değiştirme performansını göstermekte idi. Ürünler olduğu gibi kalabiliyor (iyiyken iyi, GG, veya kötüyken kötü, BB) ya da tamir olabiliyor (BG) ya da tamir olmak yerine bozuk hale gelebiliyor (GB) idi. Fisher ve diğerlerinin β değeri ise, etkililik parametreleri şeklinde ifade edilmektedir ve R tamir makinesine gelen G_i durum vektörüne sahip ürünlerin G_j durum vektörü şeklinde makineden çıkmaları olasılıklarını yansıtmaktadır. Böylece her bir spesifik tamir makinesi R_u ve her özellik f_a için, $r_{u,a}$ parametresi, R_u makinesinin a özelliğini kusursuzdan kusurluya çevirme olasılığını ve $s_{u,a}$ parametresi ise, aynı özelliği kusurludan kusursuza çevirme olasılığını göstermektedir. Diğer olası senaryolar ise bu iki durumdan türetilir ve böylece k özellikli bir ürün için tamir makinesi hakkında 4^k parametre girdisi yerine 2^k parametre girdisi yeterli olacaktır.

Verilen bu tarifler ışığında yeni β için kesin hesap aşağıdaki gibi olacaktır.

$$\beta_{i,j}(R_u) = \left(\prod_{A_{1,0}} r_{u,v} \right) * \left(\prod_{A_{1,1}} (1 - r_{u,v}) \right) * \left(\prod_{A_{0,1}} s_{u,v} \right) * \left(\prod_{A_{0,0}} (1 - r_{u,v}) \right)$$

Yukarıdaki formülde, $A_{a,b} = \{v : 1 \leq v \leq k, g_{i,v} = a, g_{j,v} = b\}$ olup, a ve b $\{0,1\}$ tamsayı değerleri içinde tanımlıdır. Örneğin, iki özellikli bir ürünün G_i durumunda ilk özelliği, G_j durumunda da ikinci özelliği kusursuzdur. R1 makinesi, ilk özelliği iyiden kötüye 0.1 olasılıkla çevirmekte ve ikinci özelliği kötüden iyiye 0.9 olasılıkla çevirmektedir. Bu durumda bu makineye G_i durumunda giren ürünlerin G_j durumuna çevrilme olasılıkları $0.9 * 0.1 = 0.09$ olarak hesaplanacaktır.

Görüldüğü gibi artık test ve tamir süreçlerinde bağımsız bir hareket görülmemektedir. Dar anlamda her özelliğin bağımsız olduğu varsayımı Fisher ve diğerleri (2007a) tarafından da yapılmıştır, fakat ürün makro bir bütün olarak kabul edilirse, durum vektörü değişimi bağımsız ve doğrusal bir sürecin değil, etkileşimli ve doğrusal olmayan bir sürecin sonucu olarak gerçekleşmektedir.

Böylece görülmektedir ki, özellik sayısı k arttıkça, problem kısa sürede başa çıkılamayacak kadar büyümektedir. Üç özellik mevcutken sadece 8 durum vektörü bulunmakta, fakat özellik sayısı sekize çıktığı anda 256 ayrı durum vektörü oluşmaktadır. Böyle bir durumda sembolik ve hesabı kolaylaştıran 100 girdi ile çalışmak da artık uygulanabilir olmayacak ve bazı olasılıklar (örneğin her özelliğin kusurlu olması durumu) hesaplanamayacak kadar küçük değerler almaya başlayacaklardır.

Bu durumun bir benzeri literatürde de görülmektedir. Lee ve Unnikrishnan (1998), muayene faaliyetleri için bu çalışmada yapılan araştırmayı andıracak şekilde muayene yeri ve ekipmanı seçimini belirleyecek bir muayene stratejisi seçiminin ilgi çekici bir kalite planlama problemi olduğunu belirtmektedirler. Muayene noktaları yerine test üretimi konulduğu takdirde bu sav net bir şekilde görülecektir. O çalışma da eldeki problemin zorluğuna dikkat çekmektedir (Lee

ve Unnikrishnan, 1997). Problem boyutu büyüdükçe ortaya çıkan seçeneklerin ve çözüm için gereken bilgisayar hafıza kapasitesinin çözümü imkansız kılacak derecede büyüdüğünü bu çalışma da onaylamaktadır.

Fisher ve diğerleri (2007a) bu üstel artan durum vektörleri sorununu çözmek ve vektör sayısını kontrol altına alabilmek için öncelikle kusurlu özellik sayısını kısıtlama yoluna gitmektedirler. Özellik sayısının k olduğu durumda $0 \leq b \leq k$ şeklinde bir b sayısı seçilerek sadece b ve daha az sayıda kusurlu özelliği olan ürünlerin dikkate alınmasını öneren bu yaklaşım, b sayısından daha fazla bozuk özellikleri olan bir ürünün mutlaka yakalanacağı varsayımına dayandırılmaktadır. Bu varsayıma göre b sayısı öyle seçilmelidir ki, firmanın elinde bu sayının üstünde bozuk özellik ihtiva eden ürünleri saptayıp ıskartaya çıkarabilecek kapasitede test makineleri bulunsun. Anlatılan özellikte bir makine bulunduğu takdirde denge durumunda sistemde hiçbir zaman b sayısından fazla kusurlu özelliğe sahip ürün bulunmayacaktır.

Böylece sadece b ya da daha az özellikleri kusurlu olan durum vektörleri dikkate alınacaktır. Dikkate alınacak bu vektörler “geçerli durumlar” (*valid states*) olarak adlandırılmıştır ve sayıları m' ile gösterilmektedir ve formülü aşağıda verilmiştir.

$$m' = \sum_{i \in b} \binom{k}{i}$$

Test noktasında k özellikten her birinin p olasılıkla iyi, $q = (1 - p)$ olasılıkla kötü olduğu varsayılır ise tam olarak b tane kusurlu özellik olma olasılığı basit binom olasılığı ile gösterilebilir:

$$P(b) = \left(\frac{k!}{b!(k-b)!} \right) * (p^{k-b}) * (q^b)$$

Sonuçta b ya da daha az sayıda kusurlu ürün olasılığı ise aşağıdaki gibi olacaktır.

$$T(b) = \sum_{i \in b} P(i)$$

Bu formüllerden hareket ederek 30 özellikli bir ürün ve her bir özelliğin değişik kusursuz olma olasılıkları için oluşan birikmeli olasılıklar Tablo 13'teki gibidir. Tabloda görülmekte olan $p = 0.99$ sütunu Fisher ve diğerlerinin çalışmasından (2007a) alıntıdır.

Tablo 13. Birikmeli Sınırlı Kusurlu Özellik Olasılıkları

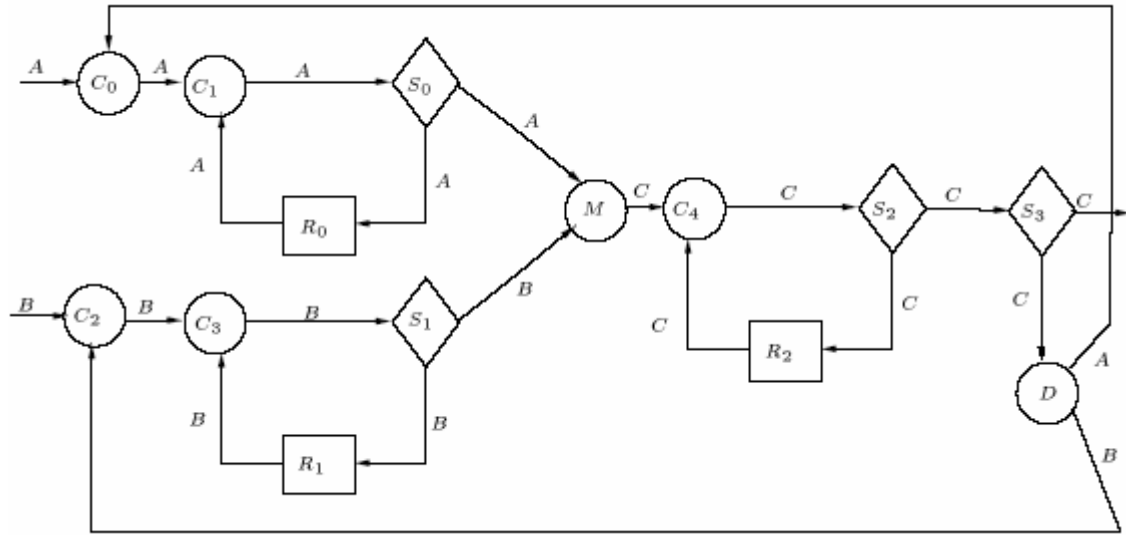
	p = 0.99	p = 0.95	p = 0.90		p = 0.99	p = 0.95	p = 0.90
P(0)	0.739	0.210	0.042	-	-	-	-
P(1)	0.224	0.339	0.141	T(1)	0.963	0.549	0.183
P(2)	0.033	0.259	0.227	T(2)	0.996	0.808	0.410
P(3)	0.003	0.127	0.236	T(3)	0.999	0.935	0.646
P(4)	-	0.045	0.177	T(4)	-	0.980	0.823
P(5)	-	0.012	0.102	T(5)	-	0.992	0.925

Tablo 13 göstermektedir ki, eğer her özellik %99 olasılıkla kusursuz olacaksa, 30 özellikli bir ürün gibi büyük bir problem sadece iki özellik üzerinden teste tabi tutulabilecektir. Çünkü bu durumda 2 yada da daha az özelliğin hatalı olma olasılığı %99.6 olacak ve ıskartaya çıkması gereken ürünler (yani üç ya da daha fazla kusurlu özelliği olan ürünler) $(1-0.966) = \%0.4$ olacaktır. Fakat tabloda da görüleceği gibi kusursuz olma olasılıkları %95 olursa, aynı seviyeye ulaşabilmek için yaklaşık 4 özelliğin test edilmesi gerekmektedir. %90 kusursuz ürün olasılığında ise %7.5 ıskarta payı ancak 5 özellik test edildiğinde gerçekleşebilmektedir. Bu değer bile %99 kusursuz özellik olasılığına ait tek özelliğin test edilmesi durumunun, ıskarta bakımından çok altında kalmakta ve fakat durum vektörleri açısından bakılığında ilk grup $2^1 = 2$ durum vektörü, ikinci grup ise $2^5 = 32$ durum vektörüne sahiptir. Bu da göstermektedir ki Fisher ve

diğerlerinin (2007a) bu yaklaşımını kullanmak için ürün özelliklerinin kusursuz olma olasılıklarının oldukça yüksek olması gerekmektedir.

Fisher ve diğerleri (2007a) aynı zamanda karmaşık test tasarımları ile ilgili bir adım daha atmış ve fakat bu tasarımların doğrudan matematiksel ifade edilmesi oldukça karmaşık ve doğrusal olmayan fonksiyonlara yol açtığı için, bu tasarımları doğrudan incelemekten geri durmuşlardır. Örnek bir test akışı aşağıdaki şekilde görülmektedir.

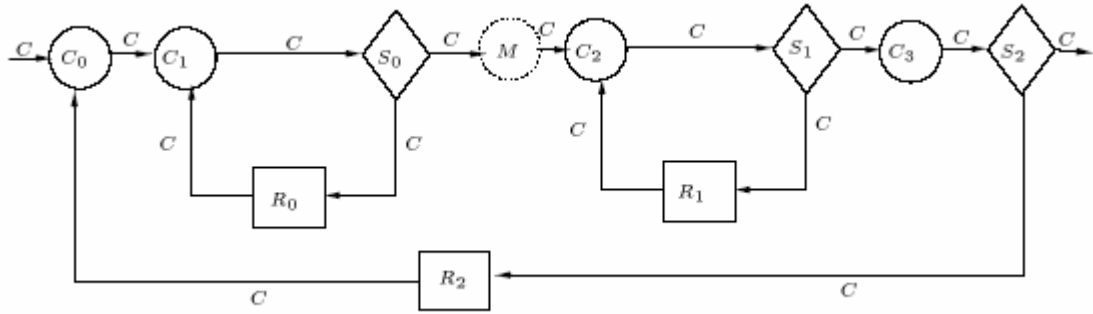
Şekil 8. A ve B Ürünlerinin C Ürünü Oluşturacak Şekilde Birleşmesi



FISHER, E., S. FORTUNE, M. GLADSTEIN, S. GOYAL, W. B. LYONS, J. MOSHER ve G. WILFONG. "Economic Modeling of Global Test Strategy II: Software System and Examples." *Bell Labs Technical Journal*, 12, 1, May 2007b: 175 – 186.

Şekilden de görüleceği gibi A ve B özellikleri ayrı olarak teste girmekte ve testi geçen çıktıları C₄ düğümünde birleşmekte ve C özelliğinin test akışını başlatmaktadır. Burada karşılaşılan M ve D düğümleri gibi çoklayıcı ("multiplexer") ve azlayıcı ("demultiplexer") düğümlerin yol açtığı doğrusal olmayan denklemler nedeniyle, A ve B özellikleri sanki C'nin bileşenleriymiş gibi ele alınmakta ve sistem C cinsinden kabul edilerek doğrusallaştırılmaya çalışılmıştır. Bu da eldeki zor problemi kolaylaştırmaya yönelik olarak Fisher ve diğerlerinin (2007a) getirdiği ikinci varsayımdır. Bu girişim sonucunda test akışı Şekil 9'da gösterildiği gibi olmaktadır.

Şekil 9. Doğrusal Yaklaşım



FISHER, E., S. FORTUNE, M. GLADSTEIN, S. GOYAL, W. B. LYONS, J. MOSHER ve G. WILFONG. "Economic Modeling of Global Test Strategy II: Software System and Examples." *Bell Labs Technical Journal*, 12, 1, May 2007b: 175 – 186.

Doğrusal yaklaşımda ürün C cinsinden kabul edilerek sistemde dolaşmakta ve A ve B özellikleri bu çalışmanın önceki alt bölümlerinde dinamikleri verilmiş olan bağımsız seri testler olarak tanımlanmıştır. Bu iki seri testten sonra C özelliği ise çıktılara uygulanan bir bağımsız test gibi görülmektedir.

Fisher ve diğerlerinin çalışması (2007a) problemi ve test akışını düzgün bir şekilde ortaya koymakta, fakat hesaplama kısmı, kesin olmasına rağmen büyük sayıda özellik için çözülemez olarak görünmektedir. Optimal bir test tasarımı açısından bakıldığında ise bu çalışma hiçbir şekilde konuya değinmemektedir. Testlerin ön tanımlı olduğu senaryolar üzerinden test akışının kesin hesaplarını vermektedir.

Her ne kadar Fisher ve diğerlerinin (2007a) önerdiği ikinci doğrusal gevşetme modeli önceki alt bölümlerde matematiksel yapısı ve optimizasyon yaklaşımı belirtilmiş olan seri test yaklaşımına benzese de böyle bir doğrusallaştırma olmadığı takdirde problem çok karmaşık bir yapıya sahiptir. Bu denli çabuk büyüyen bir probleme ait optimizasyon yöntemi geliştirmek hemen hemen imkansız gibi görünmektedir. Özellik sayısının yediyi geçtiği durumda kesin hesabı bile yapılamayan bir problemde bir de maliyet minimizasyonu eşliğinde test makinesi ve tamir makinesi seçimi yoluyla test tasarımı yapılması çok küçük özellikler sayısı için bile modellenemez bir duruma yol açacaktır.

Nihayetinde anlaşılmaktadır ki bu probleme ait durum vektörlerinin üstel çoğalmaktan kurtarılması problemin optimize edilmesi için bir ön şart kabul edilmelidir. Bir sonraki alt bölüm bu problemin doğrusal zamanda yaklaşık çözümünün nasıl olacağını aktaracaktır.

4.4. ETKİLEŞİMLİ TEST İÇİN DOĞRUSAL TAHMİNLEME

Bir önceki alt bölümde etkileşimli ve çok özellikli test probleminin kesin hesapları (*exact calculation*) incelenmiştir. Çalışmalar göstermektedir ki çok özellikli testler, eğer bağımsız test olarak yapılmıyorsa, problemin hem girdisi hem de karar değişkenlerinden olan durum vektörlerinin sayısı, ürün özelliğine bağlı üstel artış göstermektedir. Eklenen her özellik için ikiye katlanan durum vektörü sayısı, yedi özellikten sonra hesaplanamaz hale gelmektedir.

Gerçek hayatta karşılaşılan problemlere bakılırsa, test edilecek özellik sayısı en basit ürünlerde bile onları geçebilmekte, hele telekomünikasyon donanımları sektörü gibi karmaşık kartların üretildiği sektörlerde ise birkaç yüze kadar çıkabilmektedir. Bu durumda kesin değerlere yaklaştırma yapmak ve belirli bir hata payını göze alma pahasına bu hesapları doğrusal zamanda hesaplanabilecek hale getirmek dışında çare kalmamaktadır. Dislis ve diğerleri (1993b) böyle durumlar için önerilerini şöyle belirtmektedirler:

“En iyi çözümü bulmak için tüm alanı aramaya yönelik çözüm metotları sadece küçük problemler için uygun olabilir ve bu metotlar gerçekçi boyutlu problemlerde hızla kullanılamaz hale gelebilirler. O nedenle bu metotlara alternatifler üretilmelidir.”

DISLIS, C., J. DICK ve A. P. AMBLER. “Algorithms for Optimised Test Strategy Selection.” *Proceedings of IEEE ITC International Test Conference*, 1993b: 384.

Literatür incelendiği zaman ideal yöntemin yaklaştırma olduğu ve bu yaklaştırmanın zaten yapıldığı görülür. Rossi ve diğerleri (2006), Fisher ve

diğerlerinin (2007a) aksine, doğrusal olmayan denklemler kullanmaktan çekinmemiş ve olasılık prensiplerinden de faydalanarak test sürecinde değişen durum vektörlerini doğrusal zamanda yaklaşık olarak çözebilecek bir matematiksel yaklaşım getirmişlerdir.

Net bir ifadeyle, bir önceki alt bölümde anlatılan çalışmada ürünlerdeki test edilecek özellik sayısına bağlı üstel bir artış görülmekte iken, Rossi ve diğerleri (2006), doğrusal sayıda artan denklem sayısıyla problemi küçük bir hata payını göze alarak çözülebilir kılmışlardır. Aşağıda bu çalışmanın detayları incelenecektir. Zira bu çalışmada önerilen model takip eden bölümde incelenecek çok özellikli testlerin optimizasyonu problemi içinde aktif olarak kullanılacaktır. O noktada test tasarımını optimize ederken durum vektörlerinin hesaplanması için Rossi ve diğerlerinin (2006) geliştirdiği yaklaştırma yöntemi kullanılarak ideal test stratejisi saptanacak ve böylelikle bir karar verme gereci geliştirilecektir.

Rossi ve diğerlerinin çalışması (2006), test tasarımının belirlenmiş olduğu bir durumda etkileşimli akışların hesaplanması ve sistemin kaldırabileceği kapasitenin belirlenmesini amaçlamaktadır. Kapasite perspektifi bu çalışma için hayati öneme sahip olmasa da, akışların lineer zamanda hesaplanabiliyor olması, hem bu çalışmanın önereceği test tasarımının kurgulanması, hem de büyük problemlere uygulanabilir hale getirilmesi için son derece kritik bir rol oynamaktadır.

Rossi ve diğerleri (2006), durum vektörlerinin sayısının üstel artması sorununa karşı senaryo azaltma teknikleri önermektedirler. Bu yaklaşımın makul sonuçlar vermesi için makinelerin oldukça başarılı olması varsayımını ortaya koymuşlar ve Fisher ve diğerlerinin çalışmasında (2007a) görülebilecek yukarıda açıklanan “b” adet kötü özellikten fazlasının saptanıp ıskartaya çıkarılabilir olduğu mantığını da kullanmaktadırlar. Ayrıca bu kullanımdan oluşan eksik kısımlar dışındaki senaryoların genellenmesinin de bazı hata paylarına yol açacağı konusuna dikkat çekmektedirler.

Rossi ve diğerleri (2006), kesin hesap yöntemlerini tekrar incelemiş ve alternatif bir gösterim de sunmuşlardır. Bir önceki alt bölümden hatırlanacağı gibi durum vektörlerinin gösterimleri örneğin, $G_i = [1, 0, 1, \dots, 0, 1]$, şeklinde olmaktadır.

Bu yeni çalışmada aynı vektör $I_{101\dots 01}$ şeklinde gösterilmekte ve bu değişken aynı zamanda bu vektöre sahip ürünlerin oranını da göstermektedir. Bu durumda $I_{..}$ durum vektörüne denk gelen $P_{..}$ değişkeni testi geçme olasılığı ve $F_{..}$ değişkeni testte başarısız olma olasılığı kabul edilirse, önceki alt bölümde verilen testi geçme formülü,

$$\alpha_i^1(S_u) = \left(\prod_{q_{u,a} > 0, g_{i,a} = 1} q_{u,a} \right) * \left(\prod_{q_{u,a} > 0, g_{i,a} = 0} q'_{u,a} \right)$$

$P_{11} = TP[1] * TP[2] * I_{11}$ formülüne eşit hale gelmektedir. Yani I_{11} oranında görülen her iki özelliği de kusursuz ürünlerin testi geçme oranı da her iki kusursuz özelliğin testi doğru şekilde geçmesi ile oluşacak birleşik olasılığa eşittir. Bir üstte görülen formül de zaten aynı hesabı önermektedir.

Aynı mantık kullanılarak kusursuz bir ürünün testi geçememe olasılığı, $\{ F_{11} = (1 - TP[1] * TP[2]) * I_{11} \}$ yani her iki kusursuz özelliğin en az birinin testi geçememesi olasılığı ile kusursuz ürün oranının çarpımından elde edilecektir. Bu şekilde tüm olası denklemler elde edilebilir.

Genel bir prensip olarak testi geçme ($P_{..}$) olasılıkları için P 'nin indeksindeki her i 'nci 1 değeri, yani kusursuz özellik için $TP[i]$ ve her j 'nci 0 değeri, yani kusurlu özellik için $FP[j]$ terimleri çarpılır. Yani $q_{u,a}$ ve $q'_{u,a}$ değişkenleri yerine TP ve FP değişkenleri gelmiştir. Tamir süreci ise yine malzeme özelliklerinin değişimine yol açan ve fakat karar noktası olmadığı için lineer sayıda denklem ile hesaplanabilen bir süreç olarak tanımlanmaktadır. Örneğin tamirden çıkan kusursuz ürünlerin hesaplanması için β terimlerinin indeksleri kullanılarak ve

parantez içine de ilgili özelliğin I değişkenindeki indeks sırası yerleştirilerek formül şöyle yazılabilecektir:

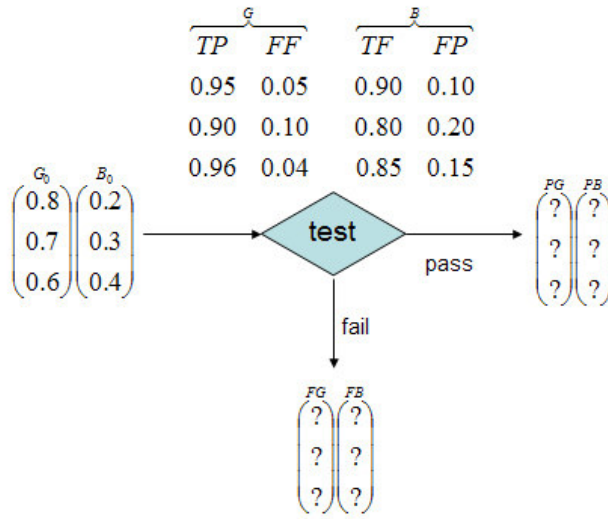
$$R_{11} = (GG[1]*GG[2])*I_{11} + (BG[1]*GG[2])*I_{01} + (GG[1]*BG[2])*I_{10} + (BG[1]*BG[2])*I_{00}$$

Böylece kusursuz özelliklerin tamirden bozulmadan çıkması ve kusurlu özelliklerin tamirden düzelerek çıkması olasılığı, tamirden çıkan toplam kusursuz ürün olasılığını vermiş olacaktır. Bu formül de bir önceki alt bölümde verilen genel β formülü ile aynıdır. Sadece gösterim farkı bulunmaktadır. Artık bilindiği üzere, tüm bu hesaplamalar kesin ve doğrudur. Fakat özellik sayısı, durum vektörleri sayısını üstel artırmaktadır. Rossi ve diğerleri (2006), fonksiyonel yapısı doğrusal olmasa da bu üstel denklem sayısını zamansal olarak özellik sayısı ile doğru orantılı olacak şekilde azaltacak bir yaklaşım önermektedirler. Bu amaçla üç temel soru sormaktadırlar:

- Test edilen parçanın testi geçme / kalma olasılığı nedir?
- Testi geçen ürünlerin her bir parçasının, teste tabi her özelliğinin, ortalama iyi / kötü olma olasılıkları (PG ve PB vektörleri) nedir?
- Testi geçemeyen ürünlerin her bir parçasının, teste tabi her özelliğinin, ortalama iyi / kötü olma olasılıkları (FG ve FB vektörleri) nedir?

Örnek bir test problemi aşağıdaki şekilde görüldüğü gibi olsun. Problem Rossi ve diğerleri (2006) tarafından sunulmaktadır.

Şekil 10. Örnek Problem Akışı



ROSSI, R., S. A. TARIM, B. HNICI ve S. PRESTWICH. "Multi-Component Testing in Telecommunications." *Proceedings of the European Conference on Operational Research EURO XXI*, July 2006: 4.

Problemde teste tabi 3 özellik, bu özelliklerin girdi dağılımları ve makine karakteristikleri görülmektedir. Test sonucu olasılıkları bağımsız olarak kolaylıkla hesaplanabilecektir. Örneğin ilk özellik açısından bakılırsa hesaplar aşağıdadır ve tüm hesaplar takip eden tabloda görülmektedir.

$$TP = 0.8 \cdot 0.95 = 0.76$$

$$FP = 0.2 \cdot 0.10 = 0.02$$

$$FF = 0.8 \cdot 0.05 = 0.04$$

$$TF = 0.2 \cdot 0.90 = 0.18$$

Tablo 14. Bağımsız Test Sonucu Olasılıkları

Özellik	TP	FF	TF	FP
1	0.760	0.040	0.180	0.020
2	0.630	0.070	0.240	0.060
3	0.576	0.024	0.340	0.060

Bu değerlerin hesaplanması basit olsa da, problemin çözümü olmamakla birlikte gereklidir. Çünkü her bir özelliğin içinde bulunabileceği dört alternatif durum

mevcuttur ve bu alternatif durumların olasılıkları artık bellidir. Etkileşimli bir üründen bahsedildiği takdirde problem şu hali almaktadır: Test sonucunda biriken nihai ürünlerin durumları, bileşenlerinin yani özelliklerinin durumlarına bağlıdır. Her özellik dört durumdan birinde olabileceğine göre, basit permütasyon sonucu görülebilir ki k özellikli nihai ürünün içinde bulunabileceği 4^k alternatif durum bulunmaktadır. Bu durum zaten bir önceki alt bölümde de görülmekteydi ve testi geçen ürünler ve geçemeyen ürünler için ayrı olarak 2^k durum söz konusu idi.

Rossi ve diğerleri (2006), bu 64 durumun her birine “senaryo” ismini vermektedirler. Bu senaryoların örnek teşkil edecek ilk on tanesi aşağıdaki tabloda görülebilmektedir. Gerekli görüldüğü takdirde tüm senaryolar için hesaplanmış değerle EK 1’de bulunabilecektir.

Tablo 15. Örnek Senaryolar

Senaryo	Özellik 1	Özellik 2	Özellik 3	SP_i	Karar
1	FF - 0.04	FF - 0.07	FF - 0.024	0,0000672	FF
2	FF - 0.04	FF - 0.07	TF - 0.340	0,0009520	TF
3	FF - 0.04	FF - 0.07	FP - 0.060	0,0001680	TF
4	FF - 0.04	FF - 0.07	TP - 0.576	0,0016128	FF
5	FF - 0.04	TF - 0.024	FF - 0.024	0,0000230	TF
6	FF - 0.04	TF - 0.024	TF - 0.340	0,0003264	TF
7	FF - 0.04	TF - 0.024	FP - 0.060	0,0000576	TF
8	FF - 0.04	TF - 0.024	TP - 0.576	0,0005530	TF
9	FF - 0.04	TP - 0.630	FF - 0.024	0,0006048	FF
10	FF - 0.04	TP - 0.630	TF - 0.340	0,0085680	TF

Her üç özelliğin durumu bir senaryo oluşturmaktadır. Fakat ürünlerin durumu nihayetinde bir karar mekanizması da gerektirmektedir. Bu kararın test sonucu değil, gerçekteki nihai durumun yansıması olması gerekmektedir. Zira bir ürünün iki özelliği kusursuz ve biri kusurlu iken her üç özellik de testi geçerse, testin kararı “P” yani ürünün testi geçmesi olacakken, burada bir “hatalı kabul”

durumu söz konusudur. Ürün aslında kusurlu özelliği nedeniyle kusurludur ve verilen “P” kararı aslında hatalı bir “P”, yani “FP” durumuna işaret etmektedir.

Karar prensipleri özetlenecek olursa beş ayrı durumla karşılaşılmaktadır. Bu durumlardan ilk grup ikinci, dördüncü ve beşinci grupla ortak birer ölçüt bulundurmaktadır fakat temel bir prensip olduğu için ayrı bir grup olarak tanımlanmıştır. Ek 1’de tümü görülebilecek durumların bir sınıflandırması şu şekilde yapılabilir:

- Tüm özellikler aynı ise gerçekteki durum o özellik (4 adet),
- En az bir TF bir özellik varsa gerçekteki durum TF (36 adet),
- En az birer FF ve FP beraber ise gerçekteki durum TF (13 adet),
- Tüm özellikler TP ve FP ise gerçekteki durum FP (6 adet),
- Tüm özellikler TP ve FF ise gerçekteki durum FF (6 adet),

Senaryolar tablosunda gözlenen spesifik durumlar gruplanabilir yapıdadır. Örneğin hatalı retler ya da hatalı kabuller, hangi özellik bileşimine ait olurlarsa olsunlar, test süreci sonunda aynı grupta yer alacaklardır. Benzer mantık kullanılarak tüm karar durumları gruplanarak senaryolar özet bir tabloda bir araya getirilebilir. Böylesi bir genelleme özetlenecek olursa, senaryoların rastlanan sıklıkları ve toplam gerçekleşme olasılıkları aşağıdaki tabloda görüldüğü gibi olacaktır. Burada yazarlar (Rossi ve diğerleri, 2006), biraraya getirilmiş her gruba da bir senaryo kategorisi tanımlı getirmektedirler ve FP kararını FPS – Hatalı Kabul Senaryosu (*False Pass Scenario*) olarak tanımlamışlardır.

Tablo 16. Senaryo Dağılımları

Senaryo	Adet	Toplam Olasılık
TP	1	0.2757888
FF	7	0.0602112
FP	7	0.0665064
TF	49	0.5974936

Tablodan görülen gerçek, yukarıdaki şekilde verilen dağılımdan bir ürün teste sokulduğu takdirde %28 olasılıkla testi doğru şekilde geçecek, %6 olasılıkla hatalı şekilde ret edilecek, %7 olasılıkla hatalı şekilde testi geçecek ya da %60 olasılıkla doğru bir kararla ret edilecektir.

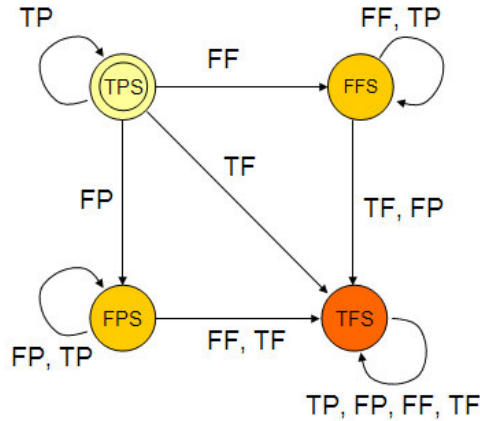
Ayrıca açıkça görülmektedir ki sadece TP ve FF kategorilerine giren ürünler tamamen kusursuz ürünlerdir. Aynı mantıkla, FP ve TF kategorilerindeki ürünler ise en az bir kusurlu bileşen içermektedir. Bu mantıkla kusursuz ürünler için gerçekleşecek akış oranı %34 ve kusurlu ürün oran %66 olarak hesaplanabilir (Rossi ve diğerleri, 2006).

Rossi ve diğerleri (2006), yukarıda açıklanan kategori kararı prensiplerini bir sonlu durum otomatonu olarak tanımlamaktadırlar. Sonlu durum otomatonu her okun bir olasılığı temsil ettiği bir Binom zincirini temsil eden ve durumlar arası geçiş prensiplerini gösteren bir grafiktir.

Otomaton ve grafik çözüm kullanımı test literatüründe de karşılaşılabilen bir durumdur. Örneğin Aho ve diğerleri (1991), test üretimi içerikli çalışmalarında otomaton ve benzeri grafik teorisiyle ifade edilebilen yöntemleri Kırsal Çinli Postacı Turu (*Rural Chinese Postman Tour*) olarak tanımlayıp çözmektedirler. Pattipati ve Alexandridis (1990), Pattipati ve Dontamsetty (1992), Novak ve diğerleri (1993), Dreschler ve diğerleri (1997) tasarım ve test çizelgeleme ve benzeri problemlerin çözümleri için yine grafik temelli VE / VEYA (*AND / OR*) grafiklerinden faydalanan çözümler ortaya koymaktadırlar.

Aşağıdaki şekil ise Rossi ve diğerleri (2006) tarafından kullanılan ve bu çalışmada incelenen probleme ilişkin otomatonu göstermektedir.

Şekil 11. Sonlu Durum Otomatonu



ROSSI, R., S. A. TARIM, B. HNICI ve S. PRESTWICH. "Multi-Component Testing in Telecommunications." *Proceedings of the European Conference on Operational Research EURO XXI*, July 2006: 5.

Şekil 11, yukarıda açıklanan karar prensiplerini grafik olarak göstermektedir. Örneğin sol üst köşede de görüldüğü gibi, ürünün TPS yani doğru kabul durumunda olabilmesi için her özelliğinde TP durumunun gözlenmesi yani bir TP döngüsü oluşması gerekmektedir.

Problemin doğası gereği sadece tek bir TPS senaryosu gerçekleşebilir. Zira teste tabi k özellikli bir ürünün her özelliği 4 farklı durumdan birinde olabilir, fakat sadece tek bir grupta her bileşen kusursuz olacaktır ve TPS senaryosu gözlenecektir.

FPS ve FFS senaryoları ise aynı ve belirli sayıda senaryo çıkarmak durumundadır. Bu durumun izahı basittir. Zira her özelliği kusursuz ürünler söz konusu ise, k özelliğin her biri TP ya da FF olmak durumundadır.

Bu durumda her özellik için 2 alternatif bulunmakta ve kusursuz ürünler toplam 2^k farklı senaryoya sahip olmaktadır. Bu 2^k senaryonun sadece bir tanesi TPS kategorisinde bulunduğu göre $(2^k - 1)$ tanesi de FFS kategorisinde bulunmak durumundadır. Bu senaryolar değişik TP ve FF kombinasyonlarını ve tamamen FF'ten oluşan tek senaryoyu içermektedir.

Benzer bir mantık FPS senaryoları için de geçerlidir. Bir testi geçen ürünler düşünülürse k özelliğin her biri TP ya da FP durumunda olabilir. Bu durumda k özellik için 2^k durum söz konusu olabilecektir.

Testi geçen ürünlerden yalnızca tek bir TPS kategorisine uygun senaryo olduğu hatırlanır ise, FPS kategorisine girecek senaryo sayısı $(2^k - 1)$ olacaktır.

Geriye kalan TFS senaryolarının sayısını hesaplamak da sıradandır. Toplam senaryo sayısının 4^k olduğu bilindiğine göre, TFS kategorisine kalan senaryo sayısı $\{ 4^k - 2*(2^k - 1) - 1 \}$ olacaktır. Bu değer de $(2^{2k} - 2*2^k + 1) = (2^k - 1)^2$ değerine yani hem FPS hem de FFS senaryo sayılarının karesine eşittir.

Artık şeklin de yardımıyla bir senaryonun, içinde yer alan özelliklerin oklarının takip edilmesi suretiyle sınıflandırmak ve oklardaki olasılıkları çarparak senaryonun gerçekleşme olasılığını bulmak sıradan bir durumdur.

Rossi ve diğerlerinin (2006) çalışması bu noktadan sonra şöyle devam etmektedir. Sınıflandırma işlemi de yapıldıktan sonra ürünlerin hangi senaryo kategorilerinin üyesi oldukları bellidir. Fakat bu durum, örneğin hatalı kabul ya da doğru ret edilen ürünlerin tamamen kötü bileşenlerden oluştukları anlamına gelmemektedir.

Testi doğru bir kararla geçemeyen bir ürünün bile içinde kusursuz olan bazı özellikleri bulunabilir. Prensipte kusurlu ürünlerin içindeki kusursuz özellikler ve kusursuz ürünlerin içindeki kusurlu bileşenler en azından olasılık bazında dahi olsa ayrıştırılmalıdırlar.

Bu durumun bir incelemesi yapılacak olursa, örnek olarak FPS kategorisindeki senaryolar incelenecektir. Toplu senaryo tablosunda FP kararına sebep veren yedi senaryo ve içlerindeki kusursuz parçalar aşağıdaki tabloda görülmektedir.

Tablo 17. FPS Kategorisi Senaryoları

Senaryo	Özellik 1	Özellik 2	Özellik 3	Olasılık	Karar
43	TP – 0.76	TP – 0.630	-	0.0287280	FP
47	TP – 0.76	-	-	0.0027360	FP
48	TP – 0.76	-	TP – 0.576	0.0262656	FP
59	-	TP – 0.630	-	0.0007560	FP
60	-	TP – 0.630	TP – 0.576	0.0072576	FP
63	-	-	-	0.0000720	FP
64	-	-	TP – 0.576	0.0006912	FP

Herhangi bir senaryo kategorisi kapsamında herhangi bir özellik için kusursuz ürün bulunan satırların olasılıkları toplanır ise, o senaryo kategorisinde ilgili özellik için kusursuz ürün oranları elde edilmiş olur. Bu işlem kusursuz ürünler için ve kusurlu ürünler için her kategoride uygulanabilir.

Örneğin FPS kategorisinde, üç özellik için elde edilen kusursuz bileşenlerin, toplam ürün kitlesinin içindeki oranları sırasıyla (%5.773, %3.674, %3.421)² olarak bulunacaktır.

Bu işlemler her kategori ve hem kusurlu hem kusursuz ürünler için gerçekleştirilirse aşağıdaki sonuçlar bulunacaktır.

Tablo 18. Kategori Temelli Kusursuz Ürünler Kitlesi

Kategori	Bileşen 1	Bileşen 2	Bileşen 3
FP	0.05773	0.03674	0.03421
TP	0.27579	0.27579	0.27579
PG	0.33352	0.31253	0.31000
FF	0.06021	0.06021	0.06021
TF	0.40627	0.32726	0.22979
FG	0.46648	0.38747	0.29000
Toplam	0.8	0.7	0.6

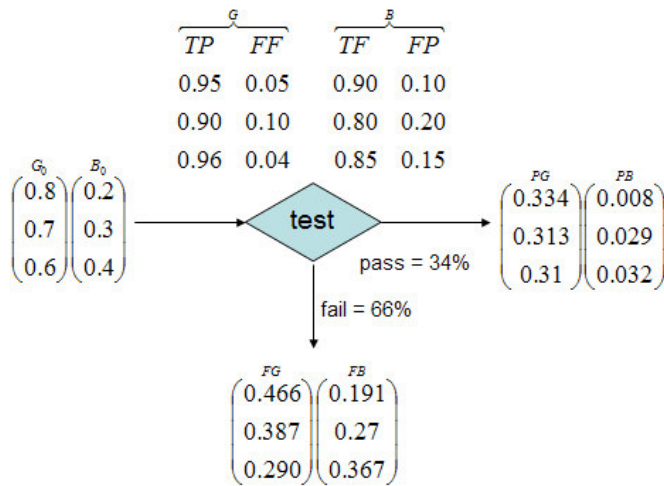
² Örneğin ilk özellik için Tablo 17’den Özellik 1 sütununda TP gözlenen satırlara ait olasılıklar alınarak $0.0287280 + 0.0027360 + 0.0262656 = 0.05773$ değeri hesaplanabilir.

Tablo 19. Kategori Temelli Kusurlu Ürünler Kitlesi

Kategori	Bileşen 1	Bileşen 2	Bileşen 3
FP	0.00878	0.02976	0.03229
TP	0	0	0
PB	0.00878	0.02976	0.03229
FF	0	0	0
TF	0.19122	0.27204	0.36771
FB	0.2	0.3	0.4
Toplam	0.8	0.7	0.6

Tablodaki değerler kullanılarak artık ilk problem çözüme kavuşturulabilecektir. Bulunan değerler durum vektörlerindeki yerlerine konacak olursa aşağıdaki şekil elde edilecektir.

Şekil 12. Problem Çözümü



ROSSI, R., S. A. TARIM, B. HNICH ve S. PRESTWICH. "Multi-Component Testing in Telecommunications." *Proceedings of the European Conference on Operational Research EURO XXI*, July 2006: 6.

Buraya kadar yazarlar tam tarama yaparak her türlü senaryo kombinasyonunu ve çıktıları içindeki iyi ve kötü ürün oranlarını etkileşimli olarak saptamışlardır. Fakat bu işlemler durum vektörlerinin doldurulmasını sağlamasına rağmen pratik işlemlerden ve Fisher ve diğerlerinin çalışmalarındaki (2007a)

hesaplamaları daha açık, kesin ve pratik yoldan yapmaktan öteye gitmemektedir. Bu yüzden Rossi ve diğerleri (2006), durum vektörü değerleri yerine tam olasılık değerlerini yazarak vektör formülleri oluşturmuşlardır.

Örneğin testi geçen ürünlerin içinde bulunan kusursuz birinci özellik olasılığı (ya da oranı):

$$PG[1] = (TP[1]*TP[2]*TP[3]) + (TP[1]*FP[2]*TP[3]) + (TP[1]*TP[2]*FP[3]) + (TP[1]*FP[2]*FP[3])$$

şeklinde bulunabilecektir. Bu formül yeniden düzenlenir ise;

$$PG[1] = TP[1] * (TP[2]*TP[3] + FP[2]*TP[3] + TP[2]*FP[3] + FP[2]*FP[3])$$

$$PG[1] = TP[1] * ((TP[2] + FP[2]) + (TP[3] + FP[3]))$$

elde edilecek ve formülün genel hali aşağıdaki gibi olacaktır.

$$PG[x] = TP[x] * \prod_{i \neq x} (TP[i] + FP[i])$$

Yani x özelliğinin doğru kabul edilme olasılığı kendisinin testi doğru olarak geçmesi ve diğer iki özelliğin testinin ise her türlü doğru ya da yanlış kabul sonucu vermesine bağlıdır. Aynı özellik kusurlu olsa idi, bu kez de ilgili x özelliğinin olasılığı, özelliğin kusurlu iken testi geçmesi ve diğer özelliklerin de şu ya da bu şekilde testi geçmelerine bağlı olacaktı. Bu durumda da testi hatalı geçme olasılığı aşağıdaki gibi gerçekleşecekti.

$$PB[x] = FP[x] * \prod_{i \neq x} (TP[i] + FP[i])$$

Bu formüllerin sağ kolunda gözlenen hiçbir terim herhangi bir test probleminde karar değişkeni olmayacak, bilakis önceden tanımlanmış parametreler olacaktır.

Yani önerilen yaklaşım durum vektörlerinin elemanlarını kolayca dolduracak bir çerçeve önermektedir.

Testin geçilmesi durumunda çıktılarıyla ilgili formüller yukarıdaki gibidir. Ürünün testi geçememesi durumunda ise aynı mantık geçerli olmakta, sadece formüller biraz daha kabarılaşmaktadır. Çünkü testi geçme durumu, ancak tüm kusursuz parçaların bir arada saptanması ya da kusursuz parçaların saptanması yanında kusurlu bazı parçaların yanlışlıkla testi geçmesi durumunda gerçekleşmektedir. Oysa herhangi bir kusurlu parçanın testi geçememesi durumunda ürün de doğrudan testi geçememe pozisyonunda kalmaktadır.

Kusursuz bileşenlerin hatalı ret edilmesi durumu ile ilgili karar prensibi yukarıda tanımlanmıştı. FF kararına denk gelen bu durumun oluşması için tek gerekli şart, ürünü oluşturan teste tabi özelliklerin TP ve FF sonuçlarından oluşmasıdır. Tüm özelliklerin FF yani hatalı kabul edilmesi dışında geçerli tek kural budur.

Bu durumda kusursuz bir ürünün reddedilme olasılığı FF ve tüm FF – TP kombinasyonlarını içerecektir. Ayrıca Rossi ve diğerlerinin (2006) tanımına göre, olasılık çevrimi de kullanılarak, kusursuz özellik oranı, özelliğin FF olması dışında, özelliğin TP olması ve aynı zamanda diğer özelliklerden en az birinin testi geçememesi demektir.

Böylece bu kategoriye giren ürünlerde rastlanacak her hangi bir kusursuz özelliğin tüm ürünler içindeki oranı da yukarıdakiler ile benzer mantıkla aşağıdaki formülle hesaplanabilir.

$$FG[x] = FF[x] + TP[x] * (1 - \prod_{i \neq x} (TP[i] + FP[i]))$$

Formül, özelliğin hatalı ret olasılığını, doğru kabul ve diğer en az bir özelliğin testi geçememe olasılığı toplamı olarak göstermektedir. En az bir özelliğin testi geçememe olasılığı ise, tüm özelliklerin doğru ya da yanlış testi geçme

olasılığının bütünleyicisidir. Zira formülün son kısmı da bu kavramı ifade etmektedir.

Aynı mantıkla testi geçemeyen ürünler içinde bulunan herhangi bir özelliğin kusurlu olan miktarının oranı (ya da kusurlu olma olasılığı) da basitçe hesaplanabilir. Özelliğin ya testi doğru bir kararla geçememiş olması, ya da yanlış bir kararla geçmesine rağmen başka bir özellik testi geçemediği için ret grubuna dahil edilmiş olması gereklidir. Bu durumda formül şöyle olacaktır:

$$FB[x] = TF[x] + FP[x] * (1 - \prod_{i \neq x} (TP[i] + FP[i]))$$

Tamir süreci ise karmaşık hesap gerektirmediği için ve bir karar noktası olmadığı için sadece kendisine gelen ürünleri değiştirmektedir ve herhangi bir özel hesap gerektirmemektedir. Rossi ve diğerlerinin (2006) de belirttiği üzere, bu çalışmada şu ana kadar tanımlanan ve formüllenen tamir süreci aynı şekilde geçerlidir. Onların gösterimiyle tamirde çıkan kusursuz ve kusurlu ürünlerin oranı aşağıdaki gibidir.

$$RG[x] = (FF[x] * GG[x]) + (TF[x] * BG[x])$$

$$RB[x] = (FF[x] * GB[x]) + (TF[x] * BB[x])$$

Rossi ve diğerleri (2006) ayrıca, bir araya gelip birleşen akışların nasıl birleştirileceği ve normalleştirileceğini de açıklamışlardır. Birleşme kısmı sıradandır ve birleşecek akışlarda bulunan her özelliğin iyi ve kötü bileşenleri birbirine eklenir. Normalleştirme ise önceki bölümlerde açıklanan kısıt programlama modelinde bulunan akış orantılama kısıtlarındaki mantığa benzemektedir. Örneğin birleşen akışların kusursuz özellikleri toplandıktan sonra, birleşmiş toplam akışa bölünecektir. Aynı işlem her özellik ve her özellik durumu için yapılmalıdır. Örneğin:

Birleşecek iki akış n özelliğe sahip $(g_0[1], \dots, g_0[n])$ ve $(b_0[1], \dots, b_0[n])$ durum vektörleri ile $(g_1[1], \dots, g_1[n])$ ve $(b_1[1], \dots, b_1[n])$ durum vektörleri olacak ise akış birleşimleri ve normalleştirilmiş akış birleşimleri aşağıdaki gibi olacaktır.

$$g''[i] = g_0[i] + g_1[i]$$

$$b''[i] = b_0[i] + b_1[i]$$

$$norm(g''[i]) = \frac{g_0[i] + g_1[i]}{g_0[i] + b_0[i] + g_1[i] + b_1[i]}$$

$$norm(b''[i]) = \frac{b_0[i] + b_1[i]}{g_0[i] + b_0[i] + g_1[i] + b_1[i]}$$

Ortaya konan dinamiklerle Rossi ve diğerlerinin önerdikleri test akış hesaplama modeli ise şu şekildedir (2006) :

$$FLG1 = (GR1 + G1) / (GR1 + G1 + BR1 + B1);$$

$$FLG2 = (GR2 + G2) / (GR2 + G2 + BR2 + B2);$$

$$FLB1 = (BR1 + B1) / (GR1 + G1 + BR1 + B1);$$

$$FLB2 = (BR2 + B2) / (GR2 + G2 + BR2 + B2);$$

$$PG1 = FLG1 * TP1 * (FLG2 * TP2 + FLB2 * FP2) * (GR1 + G1 + BR1 + B1);$$

$$PG2 = FLG2 * TP2 * (FLG1 * TP1 + FLB1 * FP1) * (GR2 + G2 + BR2 + B2);$$

$$PB1 = FLB1 * FP1 * (FLG2 * TP2 + FLB2 * FP2) * (GR1 + G1 + BR1 + B1);$$

$$PB2 = FLB2 * FP2 * (FLG1 * TP1 + FLB1 * FP1) * (GR2 + G2 + BR2 + B2);$$

$$FG1 = (FLG1 * FF1 + FLG1 * TP1 * (1 - (FLG2 * TP2 + FLB2 * FP2))) * (GR1 + G1 + BR1 + B1);$$

$$FG2 = (FLG2 * FF2 + FLG2 * TP2 * (1 - (FLG1 * TP1 + FLB1 * FP1))) * (GR2 + G2 + BR2 + B2);$$

$$FB1 = (FLB1 * TF1 + FLB1 * FP1 * (1 - (FLG2 * TP2 + FLB2 * FP2))) * (GR1 + G1 + BR1 + B1);$$

$$FB2 = (FLB2 * TF2 + FLB2 * FP2 * (1 - (FLG1 * TP1 + FLB1 * FP1))) * (GR2 + G2 + BR2 + B2);$$

$$GR1 = (FG1 * GG1 + FB1 * BG1);$$

$$GR2 = (FG2 * GG2 + FB2 * BG2);$$

$$BR1 = (FG1 * GB1 + FB1 * BB1);$$

$$BR2 = (FG2 * GB2 + FB2 * BB2);$$

$$PG1_0; PG2_0; PB1_0; PB2_0; FG1_0; FG2_0;$$

$$FB1_0; FB2_0; GR1_0; GR2_0; BR1_0; BR2_0;$$

Yukarıdaki iki özellikli bir ürün için yazılmış örnekte görüldüğü gibi yazarlar sadece durum vektörlerinin elemanlarını hesaplamakta ve araya da bir tamir dönüşümü koymak sureti ile her noktada oluşacak durum vektörlerini elde edebilmektedirler. Doğrusal olmayan kısıtlar bulundurmasına karşın

hesaplanması çok daha kolay olan bu yöntemle, uygun çözücü programlar bulunduğu takdirde çok fazla özelliğe sahip ürünler için gereken hesaplar da yapılabilecektir.

Yöntem, kesin hesap olmadığı ve yaklaştırma yapıldığı için hata payları taşımaktadır, fakat hata paylarının oldukça düşük olduğu ve kabul edilebilir sınırlar içinde bulunduğu söylenebilecektir. Özellikle büyük problemleri çözülebilir hale getirmesi sayesinde pratik anlamda son derece kullanışlı ve güvenilir bir model önerilmiştir. Yine de akılda tutulmalıdır ki yazarlar önerdikleri modelin kullanılabilir olması için makinelerin yüksek performans göstermeleri gerektiğini belirtmektedirler.

Rossi ve diğerlerinin çalışmasında (2006) sergilenen hata grafikleri incelendiğinde de tercihen makinelerin yüksek performanslı olması gerektiği, ama ideal durum için genellikle hem girdi kitlesinin kalitesinin, hem de makine performanslarının yüksek olmasının gerekliliği görülecektir. Bu şartlar oluşturulduğunda en düşük hata payları elde edilmektedir. Rossi ve diğerleri (2006) tarafından hesaplanan hata oranları şu şekildedir.

Tablo 20. Hata Oranları

Özellik	PG	PB	FG	PB	RG	RB
3						
Ortalama	0.06	5.07	1.66	0.37	1.29	1.29
Varyans	0.00	0.33	0.05	0.00	0.02	0.02
5						
Ortalama	0.03	6.11	1.84	0.70	1.63	1.63
Varyans	0.00	0.52	0.05	0.01	0.04	0.04
7						
Ortalama	0.02	6.51	1.90	0.91	1.77	1.77
Varyans	0.00	0.64	0.05	0.02	0.04	0.04

Tablo 20 göstermektedir ki, hata oranları özellik sayısı ile doğru orantılı olarak artmaktadır. Tüm hataların azalarak artıyor olması varsayımsal olarak yöntemin

daha çok özellik durumunda daha başarılı olacağı fikrini vermekte olsa da karşılaştırma yapacak kesin hesap bulunmaması, bu saptamanın sadece regresyon temelli bir çıkarım olmasından öteye gitmemesine neden olmaktadır.

Testi geçme durumlarında kötü ürünlerde ortalama %6 hata payı gözlenmektedir ki, bu ölçümler mevcut en yüksek hata grubunu oluşturmaktadır. Testi geçen ürünler içindeki kusursuz ürünlerde hata payı %0.06 ve daha az çıkmıştır ki bu gerçekten de çok düşük bir hata oranına işaret etmektedir. Testi geçemeyen ürünlerde ise kusursuz ürünlerin oranlarında %1.9 ve altı hata gözlenmekte, kusursuz ürünlerdeki hatalar da %1'in altında kalarak güven vermektedir. Tamir kısmında ise hem kusurlu hem kusursuz ürünler için eşit hatalar söz konusudur ve %2'nin altında kalan bu hatalar özellik sayısı ile doğru orantılı artmaktadır.

Sonuç itibarıyla Rossi ve diğerlerinin (2006) çalışması, doğrusal olmamakla birlikte önceki bölümlerde verilen karmaşık optimizasyon problemlerinin bir adım öteye götürülmesi için çok uygun bir altyapı hazırlamaktadır. Bir sonraki alt bölümde bu yaklaşım altyapı olarak kullanılacak ve bu yolla sıfırdan ekonomik, verimli, yüksek çıktılı ve etkin bir test tasarımının nasıl kurgulanacağı gösterilecektir. Bu tez çalışmasının bilimsel katkısı da o noktada ortaya çıkmaktadır. Özetle çok özellikli test optimizasyonu problemine dair uygulanacak testlerin, test ekipmanlarının ve bu ekipmanlara yollanacak ürün miktarlarının optimal bir şekilde belirlenmesine imkan verecek bir model takip eden bölümde sunulacaktır.

5. BÖLÜM

DOĞRUSAL TAHMİNLEME TEMELLİ KISIT PROGRAMLAMA

5. GİRİŞ

Daha önceki bölümler, test tasarımı probleminin karmaşık yapısını, kullanılacak değişkenlerin nasıl hızla çoğaldığını ve problemi çözümsüzlüğe götürdüğünü açıkça göstermektedir. Mevcut literatür içinde problemi çözülür kılmaya çalışan ve bu amaçla yaklaştırma yöntemleri kullanan tek çalışma Rossi ve diğerlerine (2006) ait olan çalışmadır. Bu çalışmada geliştirilecek çok özellikli ve etkileşimli test tasarımı modeli de Rossi ve diğerlerinin (2006) modelinden faydalanmaktadır.

Aşağıdaki alt bölümlerde önce geliştirilmesi planlanan model ile ilgili bilgiler verilecek, daha sonra ise kurgulanan model ve çözüm yöntemi detaylı olarak incelenecektir. Problemi tanımlayacak ideal modellemenin tanımı ile çözümsüzlüğü irdelenecek ve problemi çözülür kılmak için geliştirilen yaklaşım anlatılarak örnekli çözümler verilecektir.

5.1. GENİŞ KAPSAMLI TEST TASARIMI OPTİMİZASYONU

Geniş kapsamlı test tasarımı modeli, eş zamanlı iki ağ akışı modeli ile yaklaştırma yönteminin (Rossi ve diğerleri, 2006) bir arada kullanılmasını öngörecektir. En basit ifadeyle öncelikle test noktalarına aktarılabilecek ürün miktarlarının akışı ilk ağ modelini oluşturmaktadır. İkinci ağ akışı ise aktarılan ürünlere ait olan ve her test noktasında değişime uğrayan durum vektörlerindeki değişimi yansıtmaktadır. Test noktalarında ise Rossi ve

diğerlerinin (2006) yaklaşımı kullanılarak teste giren ürünlerde görülecek durum vektörleri değişimlerinin hesaplanması söz konusu olacaktır.

Probleme yaklaşım açısından belirli varsayımları belirtmek önemlidir. Öncelikle model üç farklı test makinesi arasından seçim yapacaktır. Teste tabi özellik sayısı iki olarak saptanmıştır. Toplam test süreci iki test safhasından ibarettir. Her iki test sürecinde de en az bir testin seçilmesi ön şart olarak belirlenmiştir. Her iki safhada da test edilmeyen bir ürün çıktı olarak tanımlanmamakta, yani ürünün ilk test safhasından sonra doğrudan çıktı kısmına gitmesine, ya da ilk test safhasını atlayarak teste ikinci safhadan başlamasına müsaade edilmemektedir. Ayrıca ilk test safhasında belirli bir makinede test edilen ürün, ikinci safhada aynı makineye tekrar gidememektedir. Her test makinesinin yanında bir de tamir istasyonu bulunmakta ve tamir istasyonlarının test makinesinden bağımsız aynı kalite karakteristiklerine sahip oldukları kabul edilmektedir. Herhangi bir test noktasında yapılacak test miktarı için bir test tekrarı kısıtı konmamış olup, ideal çıktı denge noktasına kadar tekrarlı test yapılacağı varsayılmıştır.

5.1.1. Optimizasyon Modeline Dair Hesaplamalar

Yukarıda bahsedilen ağ akışları Şekil 13'te görülmektedir. Öncelikle ürün miktarı akışı irdelenecektir. Teste giren ürünlerin miktarı A değişkeni ile gösterilmektedir. Test sürecinin başlama noktası hayali bir "0" numaralı makine ile tanımlanmaktadır ve bu makineden diğer üç makineye olası ürün aktarımları söz konusudur. $A_{i,j,k}$ okları, "i" makinesinden "j" makinesine "k" sayılı test sürecinde aktarılan ürünü temsil etmektedir. Haliyle örneğin başlangıç noktasından ikinci makineye gönderilen ürün miktarı, başlangıç için "0" değerli test safhası kullanıldığından A_{020} değişkeni ile ifade edilmektedir.

Şekil 13'teki modele istinaden dağıtılan ürün miktarının tanımlanması için kullanılacak ilk akış dengeleme kısıtı şu şekilde olacaktır:

$$A = \sum_j A_{0,j,0}$$

Aynı miktarda ürün, test sonrası birinci ve üçüncü makineye dağıtılacaktır. Bu durumda örneğin A_{020} miktarı, ikinci makineden ilk test safhası sonunda birinci ve üçüncü makinelere gönderilen miktarı temsil eden A_{211} ve A_{231} değişkenleri ile gösterilmektedir. Bu durumda yeni oluşan akış kısıtlarının genel yapısı da aşağıdaki gibidir.

$$A_{0,j,0} = \sum_i A_{i,j,1} , i \neq j$$

İkinci test safhasına giren ürünler her makinenin başında birleşmektedirler. Bu durumda oluşan birleşme B değişkeni ile gösterilmektedir ve denklemleri şu şekildedir:

$$B_{j,2} = \sum_i A_{i,j,1}$$

Son test safhasından geçen ürünler ise artık çıktı noktasına gitmektedirler. Örneğin ilk test makinesi için artık $B_{1,2}$ miktarı doğrudan çıktı kısmına gitmektedir ve $A_{1,4,2}$ değişkenine eşittir. Bu değişkende gösterilen hayali 4 numaralı makine ise çıktı noktasını temsil etmektedir. Böylece çıktı miktarını tutarlı hale getirecek denklemler şu şekilde oluşacaktır:

$$A = \sum_j B_{j,2} = \sum_i A_{i,4,2}$$

Ürün akış ağına dair temel prensipler bu şekilde ortaya konulabilir. Haliyle ilk ağ modeli bu kısıtlarla tamamlanmaktadır. Durum vektörleri ile tanımlanan ürün kalitesi ağı ise biraz daha karmaşık denklemlerle ifade edilmek durumundadır.

Teste giren ürün kitlesinin kalite karakteristikleri iki satır halinde gösterilmektedir. İlk satır ilk özelliği, ikinci satır ise ikinci özelliği göstermektedir. Her satırdaki ilk değer kaliteli özellik olasılığını, ikinci değer ise bozuk özellik olasılığını göstermektedir. Örnekle tanımlanacak olursa G_{100} değeri teste giren ürünlerin teste tabi ilk özelliğinin kusursuz olma olasılığı, B_{100} değeri ise bu özelliğin kusurlu olma olasılığını göstermektedir. Aynı şekilde G_{200} ve B_{200} değerleri benzer durumları ikinci özellik açısından tanımlamaktadır.

Bu değişkenleri genel olarak $G_{a,j,k}$ ve $B_{a,j,k}$ olarak ifade etmek mümkündür. İndeksler sırası ile özelliği, test makinesini ve test safhasını göstermektedir. Akışı tanımlayabilmek amacıyla bu durum vektörleri her test öncesi ve sonrasında sırasıyla $GI_{a,j,k}$ ve $BI_{a,j,k}$ değişkenleri ile $GO_{a,j,k}$ ve $BO_{a,j,k}$ değişkenlerine dönüştürülecektir. İlk grup teste giren ürünlerin durum vektörlerini, ikinci grup ise testte değişime uğradıktan sonra durum vektörlerinin aldığı son hali ifade etmektedir.

Açıkça görüleceği gibi ilk test safhasının durum vektörleri, “0” noktasındaki durum vektörü ile eşittirler. $GI_{a,j,k}$ ve $BI_{a,j,k}$ vektörlerinin, $GO_{a,j,k}$ ve $BO_{a,j,k}$ vektörlerine dönüşümünde ise Rossi ve diğerlerinin önerdiği yaklaştırma metodu kullanılmaktadır. Bu kullanımda $GI_{a,j,k}$ ve $BI_{a,j,k}$ vektörleri, Rossi ve diğerleri tarafından tanımlanan $G0 - B0$ ve $G1 - B1$ değişkenlerine, $GO_{a,j,k}$ ve $BO_{a,j,k}$ vektörleri ise $PG0 - PG1$ ve $PB0 - PB1$ değişkenlerine tekabül etmektedir.

İkinci test safhasına girmeden önce yukarıda açıklandığı şekilde ürünler testten çıktıkları makinelerden farklı makinelere dağılmaktadırlar. Böyle bir durumda ikinci safhaya girecek ürünler, test makinelerinin başında yeni kitleler oluşturmaktadırlar. Yani durum vektörleri yeniden değişmektedir.

Bu sorun basit olasılık hesapları kullanılarak aşılabilecektir. Örneğin ikinci safhada ilk makinenin girişinde oluşacak ilk özelliğe ait değer, bu noktaya ikinci

ve üçüncü makinelerden aktarılan ürünlere bağlı olacaktır. Olasılık teorisine göre oluşacak yeni durum vektörü değeri ise iki makineden gelen ürünlerin miktarlarına göre ağırlıklandırılmış ortalaması olacaktır. Sayısal bir örnekle açıklamak gerekirse, ilk test makinesine ikinci safhada ikinci makineden 40, üçüncü makineden 60 ürün gelmiş olsun. İkinci makineden gelen ürünlerin ilk özelliği %90, üçüncü makineden gelen ürünlerin ilk özelliği de %80 olasılıkla kusursuz çalışıyor olsun. Bu durumda ilk test makinesinde teste girecek ürünlerin ilk özelliğinin kusursuz çalışma olasılığı şöyle hesaplanacaktır:

$$GI_{111} = \frac{40 * 0.90 + 60 * 0.80}{40 + 60} = 0.84$$

Bu formül genellenecek olursa test girişlerinde oluşacak kitlelere ait durum vektörleri şöyle hesaplanabilir:

$$GI_{a,i,k} = \frac{\sum_i (A_{i,j,k} * GO_{a,i,k})}{\sum_i A_{i,j,k}}$$

$$BI_{a,i,k} = \frac{\sum_i (A_{i,j,k} * BO_{a,i,k})}{\sum_i A_{i,j,k}}$$

Girdilere ait yeni durum vektörleri hesaplandıktan sonra yeniden yaklaştırma yöntemi uygulanarak çıktılar belirlenecektir. Artık ürünler çıktıların biriktiği hayali dördüncü makineye aktarılacak ve aynı ağırlıklı ortalama hesabı bir kez de burada kullanılacaktır. Zira üç makineden gelen ürünler artık çıktı kitlesine ait olan durum vektörlerini oluşturacak ve bu yüzden ağırlıklı ortalama gerekecektir.

Geriye tek kalan, durum vektörlerindeki $GO_{a,j,k}$ olasılıklarını birbirleriyle çarparak ürün oranını belirlemek ve bir kısıt kullanarak ürün oranının istenen bir değerin altına inmemesini zorunlu tutmaktır. Akış içinde oluşacak makine seçimine ait sabit maliyetler, test esnasında oluşacak değişken maliyetler ve alana sevk

edilen bozuk ürünlere ait ceza maliyetlerinden oluşan amaç fonksiyonu da yazıldığında model artık test sürecini tam olarak yansıtacak hale gelmektedir.

Fakat bu modelin kurulması, problemin tanımlanması açısından pozitif bir adım olsa ve görünüş olarak estetik bir matematiksel yaklaşım öneriyor gibi görünse de, problemin çözümü açısından yine handikaplı bir durum ortaya koymaktadır. Bu sorunlar aşağıdaki alt bölümde tartışılmaktadır.

5.1.2. Geniş Kapsamlı Modelin Çözümüne Dair Sorunlar

Geniş kapsamlı modelin hazırlanması, uygulama ve çalışmanın hazırlanması sürecinde zahmetli bir çalışmaya neden olmuştur. Fakat böylesi bir modelin çözümlenerek test tasarımı probleminin sonuçlandırılması ne yazık ki mümkün olmamıştır. Modelleme kısmında aşılın sorunlar, yeni sorunlar olarak çözümleme aşamasında ciddi engeller haline gelmiştir.

Bu noktada bu tip problemleri çözümleyen yazılımlara değinmek gerekir. Daha önceki alt bölümlerde isimleri zikredilen yazılımlar içinde özellikle iki tanesi son derece gelişmiş ve üst düzey modelleme ve çözümleme yazılımları olarak ortaya çıkmaktadır. Bunlar ILOG firması tarafından geliştirilmiş OPL (*Optimization Programming Language*) programı ile Bell laboratuvarları tarafından geliştirilmiş AMPL (*A Mathematical Programming Language*) programlarıdır.

OPL programı CPLEX ve Solver kütüphanelerini kullanmak sureti ile özellikle MIP ve CP türü problemlerin çözümünde son derece etkili bir yazılımdır. Fakat kısıt programlama tekniğinin yapısı itibariyle mantıksal gerektirmeler ve doğrusal olmayan denklemler model içinde kullanılacak ise, bu denklemlerde kullanılan tüm değişkenlerin tamsayı değişkenler olmasını gerekli kılmaktadır.

Bu çalışma kapsamında yapılan araştırmalarda Rossi ve diğerlerinin (2006) kullandığı yaklaştırma yönteminin OPL programı kullanarak çözülmesinin mümkün olmadığı görülmüştür. Bu duruma yol açan birkaç temel neden bulunmaktadır. Öncelikle yaklaştırma metodunun neredeyse tamamı doğrusal olmayan kısıtlardan oluşmaktadır. Bir diğer neden, değişkenler için tanımlanan tamsayı alanların (*integer domain*) küçük olması durumunda (örneğin 0 – 1 aralığında değer alan bir olasılığın 0 – 1,000 alanına yansıtılması), tamsayı çözüm bulmak için yapılması gereken yuvarlama işlemi, tam sonuç verecek hassasiyete sahip değildir. Tamsayı alanı 10,000'e çıkarıldığında ise tamsayı hesaplarını dengeleyecek katsayılar programın müsaade ettiği tamsayı sınırlarının dışına çıkmaktadır.

Bu durumda ürün miktarına bağlı ağ modelini saliselerle ölçülecek hızda çözen OPL programının, geniş kapsamlı modelin çözümü için kullanılması imkansız hale gelmektedir.

Görülmüştür ki yaklaştırma metodu, herhangi bir tamsayı değişken çevrimine gerek bırakmadan AMPL programı ile çok hızlı bir şekilde çözülebilmektedir. Web tabanlı ücretsiz AMPL sürümü, doğrusal olmayan denklem kümelerini çözmek için Minos çözücüsü kullanarak gradyan tabanlı bir çözüm yaklaşımı ile yine birkaç saniye bile tutmayan bir sürede sonuç vermektedir. Fakat Minos çözücüsü de kullandığı gradyan temelli çözüm yöntemi nedeniyle tamsayı değişkenleri sürekli değişken gibi kabul etmektedir. Bu durum ise ürün miktarı akışı hesabı ve test makinesi seçimini uygulanabilir olmayan değerler alacak şekilde sonuç vermektedir. Aynı nedenle oluşturulan geniş kapsamlı modelin çözülmesi için AMPL programı da kullanılamamaktadır.

Aslında kendi alanlarında dünyanın en başarılı programları ile çözülemeyen test tasarımı probleminin karmaşık yapısı, sırf bu şekilde ortaya konabilir. Problemin çözülebilmesi için Yerel Arama (*Local Search*) temelli bir çözüm yaklaşımı kullanılması önerilebilir. Zira paket programlar çözüm aramasına her zaman doğru bir değişkenle başlamayabilirler ve bu yüzden çözümleme süresi başa

çıkılamayacak ve tolerans gösterilmeyecek düzeye ulaşabilir. Oysa yerel arama yöntemleriyle değişkenlere değer atama süreci yönlendirilebilir ve çözümsüz dallarda uzun süre arama yapmak yerine bu dallar bir an önce kesilerek çözüm süresi ciddi şekilde azaltılabilir. Fakat bu çalışmanın öngördüğü çözüm yöntemleri arasında yerel arama bulunmaması, bu yaklaşımı ancak ileride konuyu araştıracak bilim adamları için bir tavsiye olarak not düşmeye yarayacaktır.

Yaşanan tüm bu imkansızlıklara rağmen yukarıda genel akışı anlatılan geniş kapsamlı test probleminin çözümü için bir yöntem geliştirilmiştir. Bir sonraki alt bölüm geliştirilen yaklaşımı açıklayacak ve örnekleyecektir.

5.2. TAM SENARYO YAKLAŞIMI

Kombinatoryal yapıdaki problemlerin çözümünde zaman zaman kullanılan bir yöntem de tam numaralandırma (*full enumeration*) ya da başka bir isimle kaba kuvvet (*brute force*) yöntemidir. Bu yöntemin temel prensibi olasılıkların tümünü dikkate alarak karar vermeye dayanmasıdır. Bir karar ağacındaki her düğümü tek tek denemek buna en iyi örnektir. Yerel arama yöntemlerinin ortak amacı ise bu metodun kullanımının önüne geçmektir. Bu açıdan bakıldığında mevcut problemi çözmek için bir önceki alt bölümde öne sürülen yerel arama kullanımı tavsiyesi de mantıklı bir temele oturmaktadır.

Fakat yerel arama yöntemlerinin kullanılması şu ya da bu nedenle mümkün olmadığı zaman tam numaralandırma yöntemi kaçınılmaz bir alternatif olarak ortaya çıkmaktadır. Mevcut problem bu açıdan incelendiğinde girdi ürünlerinin sayısına dayalı kombinatoryal bir yapı görülebilecektir. Örneğin 100 girdi ürün söz konusu olsa, bu ürünlerin üç makineye paylaştırılması kombinatoryal bir yapı sergilemekte ve ürün miktarına dayalı olarak üstel artan alternatif karar sayısı oluşmaktadır. Testin ikinci safhasında ise üç makineye dağılmış bu yüz ürün, her bir makine sonrası kendi içlerinde ikiye ayrılarak diğer iki makineye

tekrar dağılacak olup ortaya çıkacak karar noktalarının sayısı başa çıkılamayacak bir değere ulaşacaktır.

Fakat küçük miktarda belirlenen girdi sayısı, hem tam numaralandırma yöntemi ile çözülecek bir yapı sergileyecek, hem de doğru modellendiği takdirde oluşacak olasılıklar hakkında genellemeye müsait fikirler verecektir. Çalışmanın bu kısmı böylesi bir yapıyı açıklamayı, problemi bu yaklaşımla çözümlemeyi ve örneklerle yaklaşımın anlaşılır kılınmasını amaçlamaktadır. Olası test kombinasyonlarının her biri birer senaryo olarak isimlendirilecek olup, tam numaralandırma metodu da böylece uygulamada “tam senaryo yaklaşımı” halini alacaktır. Burada kastedilen, alternatif senaryoların tümünün tek tek belirlenmesi ve optimizasyon modelinin bu senaryolar üzerine kurulmasıdır.

5.2.1. Tam Senaryo Yaklaşımı ve Karar Değişkenleri

Tam senaryo yaklaşımında kullanılacak varsayımlar, geniş kapsamlı model ile aynıdır. Bu açıdan bakıldığında ürünler ilk olarak birinci safha testine tabidir ve üç alternatif makinede test işlemi göreceklidir. Miktarlarına optimizasyon modeli karar verecek olsa da, girdi vektörleri teste giren ürünlerin durum vektörleri ile aynı olacağından potansiyel sonuçlar kolaylıkla hesaplanabilecektir. Tam senaryo yaklaşımının amacı yaklaştırma yöntemini optimizasyon modelinin içinden çıkararak problemi OPL programı ile çözülebilir hale getirmektir. Bu amaçla olası tüm senaryolar belirlenecek ve AMPL programı ayrı olarak kullanılarak tüm durum vektörü dönüşümleri saptanacaktır. Daha sonra bulgular genel bir optimizasyon probleminin modellenmesinde kullanılacaktır.

Probleme dair teori bir örnek yardımıyla açıklanacağından, ilk etapta girdilerin kalitesi, test makinelerinin özellikleri ve ilk test safhası sonucu oluşacak durum vektörlerinin tanımlanması faydalı olacaktır.

5.2.1.1. Problem Girdileri ve İlk Safha Senaryoları

Teste tabi ürünlerin en başta gözlenen durum vektörleri %90 oranında kusursuz özellikler öngörmektedir. İki özellik varsayıldığı hatırlanacak olursa girdi durum vektörleri şu şekilde oluşmaktadır:

$$\begin{bmatrix} G0 & B0 \\ G1 & B1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}$$

Test makinelerinin özellikleri ise aşağıdaki tabloda verilmektedir. Görüldüğü gibi ikinci makine sadece tek özellik testi yapan bir makinedir ve ilk özelliği test etmediği için bu makineye giren bir ürünün durum vektöründe ilk ürüne ait değerler herhangi bir değişikliğe uğramamaktadır.

Tablo 21. Test Makineleri Özellikleri

Makine	1				2				3			
Teşhis	TP	FF	TF	FP	TP	FF	TF	FP	TP	FF	TF	FP
Özellik 1	0.9	0.1	0.9	0.1	-	-	-	-	0.95	0.05	0.85	0.15
Özellik 2	0.9	0.1	0.9	0.1	0.95	0.05	0.85	0.05	0.85	0.15	0.95	0.05

Yukarıda belirtilen test akış şeması da (Şekil 13) göz önüne alındığında problemin modellenmesi için gerekli tüm girdiler elde edilmiş olacaktır. Fakat türev girdiler kabul edilebilecek bir diğer girdi seti daha bulunmaktadır. Tam senaryo yaklaşımının temel amacı olan yaklaştırma yöntemini ayrı olarak hesaplama gerekliliğinden dolayı birinci ve üçüncü makineye giren olası ürünlerin durum vektörü değişimleri yaklaştırma yöntemi ile hesaplanarak, ikinci makineye giren ürünlerin durum vektörü değişimi ise önceki bölümlerde anlatılan tek özellikli teste dair hesaplamalar kullanılarak saptanacaktır. Böylece optimizasyon modelinin yapısı olabildiğince basitleştirilmeye çalışılacaktır.

Yukarıda matris halinde verilen girdiler ilk test safhası sonunda, eğer ilk safhada ilgili makinelerin kullanımına karar verilirse, aşağıda görülecek durum vektörlerine (matrislere) dönüşeceklerdir.

Tablo 22. İlk Test Safhası Sonunda Durum Vektörleri

Makine	1	2	3
PG0	0.987805	0.9	0.982759
PG1	0.987805	0.9828	0.993506

Tabloda sadece kusursuz ürün olasılıkları verilmektedir. Herhangi bir makineye ait kusurlu ürünlere ait durum vektörü elemanları basitçe aşağıdaki formülle bulunabilir:

$$PBi,m = 1 - PGi,m, \forall m \in \{1,2,3\}$$

Burada durumu aktarma amacıyla gelişigüzel eklenen m indeksi, makineleri temsil etmektedir. Böylece ilk safha senaryoları tamamlanmaktadır. İkinci safha senaryoları aşağıdaki alt bölümde görülmektedir.

5.2.1.2. İkinci Safha Senaryoları

Şekil 13'te görülen akış şemasından açıkça görülmektedir ki, ilk test safhasından çıkacak ürünler değişik kombinasyonlarla ikinci safha test sürecine girdi haline geleceklerdir. Çalışma bu noktada 5 adet üründen oluşan bir girdi kitlesi olduğunu kabul edecek ve örnekler buna dayanarak hazırlanacaktır. Unutulmamalıdır ki ilk safha sonunda yukarıdaki tabloda verilen tüm durum vektörleri oluşmak zorunda değildir. Bu tamamen hangi makineye kaç ürün aktarıldığı ile ilgilidir. En azından şu söylenebilir ki, tablodaki makinelerden en az birinde ortaya çıkan durum vektörleri gerçekleşecek ve ikinci safhadaki test akışı en azından bu vektör üzerinden yürüyecektir.

Girdi kitlesindeki 5 adet ürün, potansiyel olarak topluca herhangi bir makineye gönderilebilir, ya da değişik makinelere dağıtılabilir. İkinci safhada oluşacak senaryolar tamamen bu kalıba uygun olarak hazırlanmalıdır. İkinci safhadaki herhangi bir test makinesine hiç ürün gelmeyebilir, ya da en fazla 5 adet ürün gelebilir. Bu gelen ürünler de kendisi haricindeki makinelerden aktarılmış olacaktırlar. Potansiyel olarak 0 ile 5 arasındaki ürün sayısı, diğer iki makineden her türlü kombinasyonla gelebilecektir. Hiç ürün gelmemesi durumu, matematiksel model oluşturulurken ilgilenilecek bir nokta olduğundan şu anda dikkate alınmayacaktır. Buna göre herhangi bir makineye diğer iki makineden gelen ürünlere ait senaryolar aşağıdaki gibi oluşacaktır. Senaryolar “sce” kümesi ile tanımlanacak olursa:

$$sce = \{(1-0),(0-1),(2-0),(1-1),(0-2),(3-0),(2-1),(1-2),(0-3),(4-0),(3-1),(2-2),(1-3), \\ (0-4),(5-0),(4-1),(3-2),(2-3),(1-4), (0-5)\}$$

Görülmektedir ki her makinede 20 farklı senaryo gözlenebilecektir. Her test makinesine girdi olacak durum vektörleri ise önceki bölümlerde verilen $Gl_{a,j,k}$ ve $Bl_{a,j,k}$ vektörlerine dair hesaplamalarla bulunabilecektir. Burada senaryoyu betimleyen (x-y) değerleri $A_{i,j,k}$ oklarını temsil etmekte, yani ortalama alınırken kullanılacak ağırlıklandırmaları göstermektedir. Her makineye ait ikişer örnek aşağıda görülmekte olup, tüm girdi değerleri EK 2’de bulunmaktadır.

Tablo 23. İkinci Safha Test Girdi Durum Vektörleri Örnekleri

Makine	1		2		3	
Sce	(2-1)	(2-3)	(2-1)	(2-3)	(2-1)	(2-3)
$GO_{1,*,1}$	0.9 (*2)		0.987805 (*1)		0.987805 (*1)	
$GO_{2,*,1}$	0.9828 (*2)		0.987805 (*1)		0.987805 (*1)	
$GO_{1,*,1}$	0.982759 (*3)		0.982759 (*3)		0.9 (*2)	
$GO_{2,*,1}$	0.993506 (*3)		0.993506 (*3)		0.9828 (*2)	
$Gl_{1,1,2}$	0,9276	0,94968	0,986133	0,9848	0,929267	0,95268
$Gl_{2,1,2}$	0,986367	0,98922	0,9897	0,99122	0,984467	0,9858

Tabloda her makinenin girdi aldığı makineler farklı olduğu için $GO_{a,j,k}$ değerlerinde “j” indeksi yerine “*” değeri konmuş, tablo içinde ise “*” indeksinin aldığı değer gösterilmiştir. Tablo basitçe ağırlıklı ortalamalar kullanarak herhangi bir senaryo için oluşacak test girdisine ait durum vektörlerini göstermektedir. Tüm makinelerde oluşması muhtemel tüm senaryoları için (toplam 60 adet) girdi durum vektörleri hesaplanmalıdır.

Bir sonraki aşamada ise her senaryoya ait durum vektörlerinin, senaryonun öngördüğü test sonrası alacakları değerler yine AMPL ile optimizasyon modeline dahil edilmeden hesaplanabilecektir. Tablo 22’de örneklenen senaryolar için oluşan test sonrası durum vektörleri aşağıdaki tabloda verilmektedir. Görüldüğü gibi tablo yeni değişkenler kullanmaktadır. İkinci safha girdilerinin Rossi ve diğerlerine (2006) ait yaklaştırma modeli kullanılarak testten geçirilmesi, artık nihai çıktıları oluşturacak çıktı kitlesine ait durum vektörlerinin hesaplanmasında kullanılacak değerleri ortaya koymaktadır. $OC_{a,m,s}$ değişkenleri ise bu değerleri göstermektedir. İndeks olarak kullanılan “a” ifadesi özelliği, “m” ifadesi makineyi ve “s” ifadesi ise senaryoyu göstermektedir. Haliyle örneğin ikinci test makinesinden çıktıyı oluşturmak üzere gelecek ürünlerin ilk özelliği, on yedinci senaryo kullanılarak test sürecinden geçmişse, bu $OC_{1,2,17}$ olarak ifade edilecektir. Bu kez “*” indeksi, ilgili makineyi göstermektedir. Tüm senaryolara ait test sonrası durum vektörleri EK 2’de bulunabilecektir.

Tablo 24. İkinci Safha Test Sonrası Durum Vektörleri Örnekleri

Makine	1		2		3	
sce	(2-1)	(2-3)	(2-1)	(2-3)	(2-1)	(2-3)
$OC_{1,*,s}$	0,990441	0,992565	0,986133	0,984800	0,988656	0,989841
$OC_{2,*,s}$	0,995730	0,996123	0,997451	0,9977	0,997642	0,997774

Bu noktada tekrar belirtilmelidir ki tüm bu hesaplanan senaryolar potansiyel senaryolardır ve gerçekleşmeleri ancak bir optimizasyon modeli çözümünde seçilmeleri durumunda oluşacaktır. $OC_{a,m,s}$ kümesinden seçilecek senaryolara ilişkin durum vektörleri yeniden $GO_{a,m}$ değişkenleri ile tanımlanacaktır (Bu

değişkenlere ilerleyen alt bölümlerde değinilecektir). Bir diğer hayati nokta da şudur: ikinci safha senaryoları topluca hesaplandıktan sonra artık ilk safha senaryolarının optimizasyon modeline eklenmesine gerek kalmamaktadır. Zira ikinci safha senaryoları zaten ilk safha senaryolarını hem aktarılan ürün miktarı, hem de durum vektörlerinin değişimi açılarından kapsamaktadır. Tek unutulmaması gereken maliyet tabanlı bir amaç fonksiyonu yazarken kullanılacak senaryo maliyetleri belirlenirken ilk safha senaryolarını da hesaba katmaktır.

Bu hesaplarla birlikte tam senaryo yaklaşımı ile optimize edilecek test stratejisine ait tüm kısıt girdileri tamamlanmaktadır. Bir sonraki alt bölüm optimizasyon modelinin oluşturulmasını, matematiksel gösterimini ve örnekli çözümünü sunmaktadır.

5.2.2. Tam Senaryo Yaklaşımı Modeli ve Çözümlemesi

Belirtildiği gibi tam senaryo yaklaşımı için gerekli her türlü bilgi artık elde edilmiştir. Bu alt bölümde oluşturulacak optimizasyon modeli burada yeni olarak sunulacak bir takım ek kısımlar da içerecektir fakat genel olarak yukarıda kullanılan matematiksel ifadeler ve hesaplara dayanmaktadır. Çalışmanın bu kısmı üç alt bölümden oluşmaktadır. Öncelikle amaç fonksiyonu yazılacak, daha sonra model betimlenecek ve son olarak da uygulamalı örneklerle geliştirilen model incelenecektir.

5.2.2.1. Amaç Fonksiyonu

Geliştirilecek modelin amaç fonksiyonu temel olarak iki maliyet kaleminin toplamından oluşmaktadır. Bu maliyet kalemlerinden biri alana aktarılan hatalı ürünü (*field reject ratio*) cezalandırmakta, diğeri ise herhangi bir senaryonun

seiminden kaynaklanan maliyeti betimlemektedir. Burada senaryodan kasıt artık ikinci safha testinde seilen bir senaryodur.

alıřmanın nceki alt blmlerinde de belirtildiėi gibi kaliteli rn oranı, ya da rn oranı, “y” deėiřkeni ile tanımlanmakta ve nihai rnler iinde kusursuz alıřacak rn miktarını gstermekteydi. Modelin yapısı itibariyle y deėiřkeni 0 ile 10,000 arasında bir deėer alabilmektedir. Haliyle $y = 9921$ gibi bir ifade, aslında kaliteli rn oranının %99.21 olduėunu sylemektedir. Bu durumda $(10,000 - y)$ ile hatalı rn miktarı olasılıėı dolaylı olarak elde edilecektir. Her %0.01’lik hata iin 1 birim ceza maliyeti, problemin zldė maliyet leėi aısından uygun grldė iin $(10,000 - y)$ deėeri, ama fonksiyonunun ilk bileřenini oluřturmaktadır.

Senaryolara ait maliyetin hesaplanması ise biraz daha karmařık bir yapı sergilemektedir. Temel olarak bir senaryo seildiėi takdirde hem ilk, hem de ikinci safhada kullanılacak makineler saptanmıř olmaktadır.

Bylece ncelikle kullanılacak makinelerin sabit maliyetleri dikkate alınacaktır. rneėin ikinci makine iin seilen senaryo (1-0) ya da (3-0) řeklindeyse, ilk safhada ilk makinenin, ikinci safhada ise ikinci makinenin seildiėi aıktır ve ilgili makinelerin sabit maliyetleri devreye girecektir. Makineler iin kapasite kısıtlaması ngrlmediėi iin tek bir makine seilmiř sayılacaktır. Oysa seilen senaryo (1-2) ya da (3-1) gibi her iki diėer makineden rn aktarımı ngryorsa, bu durumda  makinenin de sabit maliyetleri eklenmelidir.

Daha sonra ise her iki test safhasında oluřan deėiřken maliyetler toplam maliyete eklenmelidir. rneėin ikinci makinede (1-2) senaryosunun seilmesi durumunda nce ilk makinenin ilk safhada yaptıėı 1 rne ait testin deėiřken maliyeti, yanına nc makinenin ilk safhada yaptıėı 2 rne ait testin deėiřken maliyeti ve son olarak da ikinci safhada ikinci makinenin 3 rne uyguladıėı testin deėiřken maliyeti toplam maliyete eklenmelidir.

Örnek problemde öngörülen sabit ve değişken maliyetler ile her üç makine ve her senaryo için oluşan toplam senaryo maliyetleri aşağıdaki tablolarda görülecektir.

Tablo 25. Birim Maliyet Girdileri

Makine	1	2	3
Sabit Maliyetler	100	70	90
Değişken Maliyetler	3	2	1

Tablo 26. Amaç Fonksiyonu Maliyet Katsayıları

sce	M1	M2	M3	sce	M1	M2	M3
(1-0)	175	175	194	(3-1)	279	278	275
(0-1)	194	163	163	(2-2)	278	276	274
(2-0)	180	180	198	(1-3)	277	274	273
(1-1)	269	268	267	(0-4)	206	172	172
(0-2)	198	166	166	(5-0)	195	195	210
(3-0)	185	185	202	(4-1)	284	283	279
(2-1)	274	273	271	(3-2)	283	281	278
(1-2)	273	271	270	(2-3)	282	279	277
(0-3)	202	169	169	(1-4)	281	277	276
(4-0)	190	190	206	(0-5)	210	175	175

Burada gösterilen senaryo maliyetleri ancak ilgili senaryo optimal çözüm içinde bulunuyor ise, yani seçilmişse gerçekleşecektir. Senaryo maliyetleri matematiksel modelde $CSC_{m,s}$ değişkeniyle gösterilecektir. Bu ifade “m” makinesinde, “s” senaryosu seçildiği takdirde ortaya çıkacak maliyeti tanımlamaktadır. Herhangi bir makinede, herhangi bir senaryonun seçimi ise $t_{m,s}$ karar değişkeni ile gösterilmektedir. Bu değişken sadece tamsayı 0 ve 1 değerleri almaktadır. Senaryo seçilirse değeri 1, aksi takdirde 0 olacaktır. Bu durumda amaç fonksiyonu Z şöyle oluşmaktadır:

$$Z = (10,000 - y) + \sum_{m,s} (CSC_{m,s} * t_{m,s})$$

Yukarıda açıklanan prensipler ve hesap yöntemleri kullanılarak amaç fonksiyonu bu şekilde elde edilmiş olacaktır. Takip eden alt bölümde modelin kısıtları incelenecektir.

5.2.2.2. Model Kısıtları

Amaç fonksiyonu saptandıktan sonra sıra modelin kısıtlarına gelmektedir. Kısıtlar birkaç başlık altında incelenebilecektir. Öncelikle ikinci test safhasında herhangi bir test makinesi için sadece bir senaryonun gerçekleşmesi gerekmektedir. Tam senaryo yaklaşımı kullanılmasa idi, modelden beklenen test makinesinin ve bu makineye gelen ürün aktarım miktarlarının belirlenmesi olacaktı. Bu durumda senaryolardan sadece biri gerçekleşmiş olacaktı. Belirlenen durumu sağlayacak kısıt aşağıdaki gibidir:

$$\sum_{m,s} t_{m,s} = 1, \quad \forall m \in \{1, 2, 3\}, s \in \{1, 2, \dots, 20\}$$

Daha sonra ikinci test safhasındaki her test makinesinden seçilip uygulanan bir senaryo sonucu çıkan ürünlerin test sonrası durum vektörleri hesaplanmalıdır. Önceden belirtildiği üzere tüm senaryoların çıktıları $OC_{a,m,s}$ değişkenleri ile hesaplanmış ve $GO_{a,m}$ değişkenlerine dönüşmeleri ise ilgili senaryoların seçilmesine bağlıdır. $GO_{a,m}$ değişkenlerinde görülen “a” indeksi teste tabi özelliğe, “m” indeksi ise ürünün test işlemi gördüğü makineye aittir. Dönüşüm yine $t_{m,s}$ karar değişkenleri üzerinden yapılacaktır. Buna göre nihai ürünü oluşturacak durum vektörleri aşağıdaki formülle hesaplanacaktır.

$$GO_{a,m} = \sum_s (OC_{a,m,s} * t_{m,s}) \quad \forall a \in \{1, 2\}, m \in \{1, 2, 3\}$$

Daha sonra kısıt programlama kullanımını mümkün kılmak amacıyla sürekli yapı gösteren bu değişkenler tamsayı değişkenlere çevrilmelidir. Bu amaçla $GOI_{a,m}$ değişkenleri kullanılacaktır. Değişim şu şekilde gerçekleşecektir:

$$10,000 * GO_{a,m} - \varepsilon \leq GOI_{a,m} \leq 10,000 * GO_{a,m} + \varepsilon \quad \forall a \in \{1, 2\}, m \in \{1, 2, 3\}$$

Bu çalışmada hata payı olarak “1” alınmıştır. Böylece oluşacak en büyük hata on binde yarım olacaktır. Başka bir deyişle örneğin %98.23 olan bir olasılık en kötü ihtimalle %98.22 ya da %98.24 değerlerinden birinden hatalı yuvarlanmış olacaktır. Kısıt programlama modelinin çalışabilmesi için bu kadar küçük bir hata payının göze alınması bir mecburiyet olarak ortaya çıkmaktadır.

Bir sonraki aşamada senaryolara ait ürün akış miktarlarının bulunması gereklidir. Çünkü bu miktarlar nihai ürünün durum vektörü hesaplanırken kullanılacak ağırlıkları oluşturmaktadır. Akış miktarlarının hesaplanmasında senaryoların sistematik yapısı ile birlikte senaryo seçimini belirleyen karar değişkenleri kullanılacaktır.

Fark edileceği üzere bir test makinesine kaç adet ürün sevk edilmişse, o değer bir fazlası kadar senaryo alternatifi bulunmaktadır. Örneğin tek ürün aktarımı halinde iki senaryo $\{(1-0) \text{ ve } (0-1)\}$, iki ürün aktarımı halinde üç senaryo $\{(2-0), (1-1) \text{ ve } (0-2)\}$ alternatifi gözlenmektedir.

Tablo 23 incelenecek olursa bu durumun her beş farklı ürün miktarı için de geçerli olduğu görülür. Senaryo seçim karar değişkenleri $t_{m,s}$ ise sadece 0 ve 1 değerleri almaktadır. Bir senaryo seçilmediği takdirde zaten aktarılan ürün miktarı $A_{m,s}$ de sıfır değeri alacağından senaryo alternatiflerinin sistematik yapısı ve karar değişkenleri bir arada kullanılarak aktarılan ürün miktarları belirlenebilir:

$$A_{m,s} = t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{1, 2\}$$

$$A_{m,s} = 2 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{3, 4, 5\}$$

$$A_{m,s} = 3 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{6, \dots, 9\}$$

$$A_{m,s} = 4 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{10, \dots, 14\}$$

$$A_{m,s} = 5 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{15, \dots, 20\}$$

Bu formüller kullanılarak doğru ürün akış miktarları kolayca hesaplanmış olacaktır. Bunlara ek olarak teste giren tüm ürünlerin testten geçiyor olmalarını garanti etmek adına tüm ürünlerin dağıtımı da bir kısıtla garanti altına alınmalıdır. Bu kısıt aşağıdaki gibi olacaktır.

$$\sum_{m,s} A_{m,s} = A$$

Mevcut problem için A, değeri toplam girdi miktarı olan beşe eşittir. Son grup kısıtlara geçmeden önce belirtilmelidir ki kısıt programlama kullanımı ile problemi çözebilmek için aktarılan ürün miktarları da tamsayı değişken ($AI_{m,s}$) olarak tanımlanmalıdır. Bu kısıt oldukça sıradandır:

$$A_{m,s} = AI_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{1, 2, \dots, 20\}$$

Geriye sadece nihai ürünlerin durum vektörlerinin hesaplanması, bu değerlerin kullanılması ile ürün oranı y'nin saptanması ve y için bir alt sınır kısıtı belirlemek kalmaktadır. Nihai ürüne ait durum vektörleri aslında $GI_{a,j,k}$ değerleri gibi ağırlıklı ortalama ile hesaplanacaktır. Bu değerler AOQ_a değişkenleri ile gösterilmekte ve teste tabi her a özelliği için oluşan ortalama çıktı kalitesini (*Average Output Quality*) betimlemektedirler. AOQ_a değişkenlerinin hesabı için modele eklenen kısıtlar şu şekildedir:

$$AOQ_a = \sum_{m,s} (AI_{m,s} * GOI_{m,s}) \quad , \quad \forall a \in \{1, 2\}$$

Formülden kolayca görülmektedir ki $AI_{m,s}$ ve $GOI_{a,m}$ gibi karar değişkenlerinin çarpımı söz konusudur ve bu da açıkça doğrusal olmayan bir denklemdir. Çözümleme için OPL programı kullanılacağı ve kısıt programlama yaklaşımı uygulanacağı için bu denklemlerin tamsayı haline getirilmesi gerekmektedir. Böylece yukarıda gerçekleştirilen tamsayı çevrimi kısıtlarının nedeni de açıklanmaktadır.

Son olarak ürün oranı y hesaplanacak ve kısıtlanacaktır. Önceden belirtildiği gibi bir ürünün hatasız çalışma olasılığı, tüm bileşenlerinin kusursuz olma olasılıkları çarpımına eşitti. Böyle bir değer ancak AOQ_1 ve AOQ_2 değişkenlerinin çarpımı yoluyla hesaplanabilir. Bu değerler 10,000 üzerinden olasılıkları gösterdiği için öncelikle bir ara değişken yt kullanılacak ve sonra bu yt değişkeni 10,000'e bölünerek y hesaplanacaktır. Son işlem sırasında yuvarlamalardan oluşacak hataları önlemek için gene bir hata payı bırakılacak ve son olarak y değişkenine bir alt sınır tanımlanarak modelin matematiksel gösterimi tamamlanacaktır.

Bu kısıtlar sırasıyla aşağıdaki toplu modelin son üç kısıtını oluşturmaktadır. Bu çalışmada hemen yukarıdaki ikinci kısıt için hata payı yine 1 olarak alınmış ve y için belirlenen alt sınır olan y_0 değişkeni için de %99 ve üstü değişik değerler kullanılmıştır.

Tüm bu bilgilerin ışığında test tasarımı problemi için oluşturulan tam senaryo yaklaşımlı optimizasyon modeli bir araya getirilebilir. Model artık bir kısıt çözümleme programı ile çözülebilecek hale gelmiştir. Model bu çalışmada OPL programı ile çözülmektedir ve aşağıdaki matematiksel yapıya sahiptir.

Minimize

$$Z = (10,000 - y) + \sum_{m,s} (CSC_{m,s} * t_{m,s})$$

Subject to

$$\sum_{m,s} t_{m,s} = 1, \quad \forall m \in \{1, 2, 3\}, s \in \{1, 2, \dots, 20\}$$

$$GO_{a,m} = \sum_s (OC_{a,m,s} * t_{m,s}) \quad \forall a \in \{1, 2\}, m \in \{1, 2, 3\}$$

$$10,000 * GO_{a,m} - \varepsilon \leq GOI_{a,m} \leq 10,000 * GO_{a,m} + \varepsilon \quad \forall a \in \{1, 2\}, m \in \{1, 2, 3\}$$

$$A_{m,s} = t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{1, 2\}$$

$$A_{m,s} = 2 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{3, 4, 5\}$$

$$A_{m,s} = 3 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{6, \dots, 9\}$$

$$A_{m,s} = 4 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{10, \dots, 14\}$$

$$A_{m,s} = 5 * t_{m,s} \quad , \quad \forall m \in \{1, 2, 3\}, s \in \{15, \dots, 20\}$$

$$\sum_{m,s} A_{m,s} = A$$

$$AOQ_a = \sum_{m,s} (AI_{m,s} * GOI_{m,s}) \quad , \quad \forall a \in \{1, 2\}$$

$$yt = \prod_a AOQ_a$$

$$\left(\frac{yt}{10,000}\right) - \varepsilon \leq y \leq \left(\frac{yt}{10,000}\right) + \varepsilon$$

$$y \geq y_0$$

Takip eden alt bölümde yukarıdaki model OPL programı aracılığıyla çözülecek ve sonuçlar yorumlanacaktır.

5.2.2.3. Sayısal Çözüm ve Örnek Problem

Bu alt bölümde önceden verilen örnekler kapsamında verilen girdiler ve bir önceki alt bölümde verilen model kullanılarak problemi yansıtan sayısal bir

örnek incelenecektir. Girdiler ve model OPL programına girilecek olursa kullanılan optimizasyon kodu şu şekildedir:

```

int sce = 20;
range s 1..sce;
int mac = 3;
range m 1..mac;
int att = 2;
range a 1..att;

float OC[a,m,s] = [[[0.987807, 0.995776, 0.987807, 0.991767, 0.995776,
0.987807, 0.990441, 0.993098, 0.995776, 0.987807, 0.989780, 0.991767,
0.993765, 0.995776, 0.987807, 0.989384, 0.990971, 0.992565, 0.994166,
0.995776] , [0.9878, 0.9828, 0.9878, 0.9853, 0.9828, 0.9878, 0.98613,
0.984467, 0.9828, 0.9878, 0.98655, 0.9853, 0.98405, 0.9828, 0.9878, 0.9868,
0.9858, 0.9848, 0.9838, 0.9828], [0.9946, 0.993974,0.9946, 0.988656,
0.993974, 0.9946, 0.988656, 0.990632, 0.993974, 0.9946, 0.985702, 0.988656,
0.991622, 0.993974, 0.9946, 0.985112, 0.987473, 0.989841, 0.992216,
0.993974]],[[0.995257, 0.996739, 0.995257, 0.995974, 0.996739, 0.995257,
0.995730, 0.996224, 0.996739, 0.995257, 0.995610, 0.995974, 0.99635,
0.996739, 0.995257, 0.995539, 0.995827, 0.996123, 0.996427,
0.996739],[0.99714, 0.998073, 0.99714, 0.997606, 0.998073, 0.99714,
0.997451, 0.997762, 0.998073, 0.99714, 0.997373, 0.997606, 0.99784,
0.998073, 0.99714, 0.997326, 0.997513, 0.9977, 0.997877,
0.998073],[0.997981, 0.998257, 0.997981, 0.997724, 0.998257, 0.997981,
0.997642, 0.997808, 0.998257, 0.997981, 0.997601, 0.997724, 0.997851,
0.998257, 0.997981, 0.995777, 0.997675, 0.997774, 0.997876, 0.998257]]];

float CSc[m,s] = [[175, 194, 180, 269, 198, 185, 274, 273, 202, 190, 279, 278,
277, 206, 195, 284, 283, 282, 281, 210], [175, 163, 180, 268, 166, 185, 273,
271, 169, 190, 278, 276, 274, 172, 195, 283, 281, 279, 277, 175], [194, 163,
198, 267, 166, 202, 271, 270, 169, 206, 275, 274, 273, 172, 210, 279, 278, 277,
276, 175]];

var int t[m,s] in 0..1;
var float+ GO[a,m];
var int GOI[a,m] in 0..10000;
var float+ A[m,s];
var int AI[m,s] in 0..5;
var int AOQ[a] in 0..10000;
var int yt in 0..maxint;
var int y in 0..10001;

minimize
10000 - 1*y + sum(m in 1..mac, s in 1..sce) CSc[m,s]*t[m,s]

```

```

subject to{
forall(a in 1..att, m in 1..mac) GO[a,m] = sum(s in 1..sce)(OC[a,m,s]*t[m,s]);
forall(m in 1..mac) sum(s in 1..sce)t[m,s] = 1;
forall(m in 1..mac, s in 1..2) A[m,s] = t[m,s];
forall(m in 1..mac, s in 3..5) A[m,s] = 2*t[m,s];
forall(m in 1..mac, s in 6..9) A[m,s] = 3*t[m,s];
forall(m in 1..mac, s in 10..14) A[m,s] = 4*t[m,s];
forall(m in 1..mac, s in 15..20) A[m,s] = 5*t[m,s];
sum(m in 1..mac, s in 1..sce) A[m,s]=5;
forall(a in 1..att, m in 1..mac) 10000*GO[a,m] -1 <= GOI[a,m] <=
10000*GO[a,m] +1;
forall(s in 1..sce, m in 1..mac) AI[m,s] = A[m,s];
forall(a in 1..att) AOQ[a]*5 = sum(m in 1..mac, s in 1..sce) AI[m,s]*GOI[a,m];
yt = AOQ[1]*AOQ[2];
yt/10000 - 1 <= y <= yt/10000 + 1;
y >= 9910;};

```

Yukarıda verilen kod bire bir OPL programı tarafından okunacak koddur. Yani model bu şekilde programa girildiği takdirde burada incelenecek sonuçların tam olarak aynıları elde edilecektir. Bu problemin çözümü ile ilgili bilgiler ve sonuçlar aşağıdaki tablolarda görülmektedir.

Tablo 27. Tam Senaryo Optimizasyon Modeli Çözüm Bilgileri

Kısıt Sayısı	141
Değişken Sayısı	196
Çözücü Hafızası	672088
Hatalı Çözüm Noktaları	1246
Seçim Noktaları	1249
Çözüm Süresi	0.85 sn

Bu bilgiler çözücünün tuttuğu bir nevi istatistiki bilgilerdir. Optimizasyon problemlerinde özellikle çözüm süresi son derece önemli bir faktördür. Çünkü karmaşık problemler potansiyel olarak uzun çözüm süreleri sergilemektedirler ve birçok araştırmacı için çözüm sürelerini kısaltacak etkin algoritmalar oluşturmak başlı başına bir araştırma dalıdır. Bu manada ilk etapta çözüm alınamayan bir problem, bu çalışmada tam senaryo yaklaşımı ile 1 saniyenin altında bir çözüm süresi ile çözülmektedir. Bu nokta problem açısından son

derece önemli bir bilimsel katkıdır. Aşağıdaki tabloda ise problem sonuçları ve değişkenlerin aldıkları değerler görülecektir.

Tablo 28. Tam Senaryo Optimizasyon Modeli Sonuçları

Çözüm Sırasında Elde Edilen Amaç Fonksiyonu Değerleri	Solution with Objective Value: 660.0000 Solution with Objective Value: 659.0000 Solution with Objective Value: 630.0000 Solution with Objective Value: 629.0000 Optimal Solution Objective Value: 629.0000	
Senaryo Seçimleri	$t[1,5] = 1$ $t[2,1] = 1$ $t[3,5] = 1$	
Test Çıktılarının Durum Vektörleri	GO[1,1] = 0.9957 GO[1,2] = 0.9878 GO[1,3] = 0.9939 GO[2,1] = 0.9967 GO[2,2] = 0.9971 GO[2,3] = 0.9982	GOL[1,1] = 9958 GOL[1,2] = 9879 GOL[1,3] = 9940 GOL[2,1] = 9967 GOL[2,2] = 9972 GOL[2,3] = 9982
Aktarılan Miktarlar	A[1,5] = 2.0000 A[2,1] = 1.0000 A[3,5] = 2.0000	AI[1,5] = 2 AI[2,1] = 1 AI[3,5] = 2
Nihai Ürün Durum Vektörleri	AOQ[1] = 9935 AOQ[2] = 9974	
Ürün Oranı	yt = 99091690 y = 9910	

Mevcut problemin çözümü göstermektedir ki, birinci makinede beşinci senaryo yani (0-2) şeklinde bir aktarım, ikinci makinede ilk senaryo yani (1-0) şeklinde bir aktarım, üçüncü makinede ise yine beşinci senaryo yani (0-2) şeklinde bir aktarım söz konusudur. Yani ikinci safhada her üç test makinesi de kullanılmış, ilk safhada ise sadece ilk ve ikinci makineler kullanılmıştır. Çıktıların kalitesi %99.1 şeklindedir ve toplam maliyet 629 birimdir.

Model biraz daha serbest bırakılır, örneğin y için sınır %99, ya da %98 yada %95'e indirilirse sadece ikinci makinede ikinci stratejiye geçiş görülmekte ve çıktı oranı %99'un altına inmemektedir. Burada amaç fonksiyonundaki hatalı ürün maliyetlerinin rol oynadığı öngörülebilir. Zira bu kalem amaç fonksiyonundan çıkarıldığında, %95 ve %98 ürün oranları sırayla istenirse aynı sonuçlarla ilk makinede birinci, ikinci makinede dokuzuncu ve üçüncü makinede ikinci senaryo seçilmekte, yani tamamen farklı bir strateji uygulanmaktadır. Ürün oranı %98.35 olarak görülmekte ve maliyet 507 birime düşmektedir.

İlk girdiler kullanılarak ürün oranı zorlandığı takdirde elde edilen en iyi değer %99.12 şeklindedir. Böylesi bir durumda ilk makinede dokuzuncu, ikinci ve üçüncü makinelerde ise ilk senaryo seçilmekte ve toplam maliyet 659 birime çıkmaktadır. Makine özellikleri, maliyet değerleri ile oynayarak modeldeki değişim gözlenebilir. Fakat normalde rastlanacağı üzere modelin belirli yaklaşımlar göstereceği açık olduğundan buna gerek görülmemiştir. Örneğin %99.12'den daha yüksek ürün oranı elde etmek için makine özelliklerini daha iyi hale getirmek ya da girdi ürünlerin kalitesini artırmak şarttır. Bu değer aşılamaması maliyetlerden bağımsız olup tamamen teknoloji şartlarından kaynaklanmaktadır. Ürün oranı üstüne zorlanan alt sınır ise böyle yüksek değerlerde olduğunda maliyet kalemleriyle oynamak sadece maliyeti değiştirecek ve stratejiye etki etmeyecektir. Maliyet yönünden etkiyi net görebilmek için ürün oranı üzerindeki kısıt olabildiğince gevşek tutulmalıdır.

Yine de yapılan deneyler ve elde edilen sonuçlar modelin sağlıklı bir şekilde çalıştığını ve eksiksiz olarak sonuçları saptanmış senaryoları kullanmak sureti ile bir test stratejisi belirlenmesi ve test tasarımı oluşturulmasına imkan verdiğini göstermektedir. Burada geliştirilen yaklaşımlar ve modeller, test tasarımı probleminde yaklaşımları açısından bilim dünyasında ilk kez ele alınmaktadır. Çalışmanın mevcut hali ile birçok sınırlılığa maruz kalınması söz konusu ise de, geliştirilen model ve yöntem, hem konuyu derinlemesine inceleme, hem temel prensipleri ortaya koyma, hem de küçük çaplı da olsa gerçeğe neredeyse bire bir yansıtan bir test probleminin matematiksel modellemesini ortaya koyarak optimale yakın bir şekilde çözmesi açısından önemlidir.

Bulunan sonuçların gerçek hayattaki eşdeğer bir problemle karşılaştırıldığında optimal olarak değer kazanmayacağı belirtilmelidir. Zira test noktalarında çalıştırılan Rossi ve diğerlerine (2006) ait algoritma bir kesin hesap değil on bin üzerinden tek basamaklarla ifade edilen hata payları öngören bir yaklaşımdır. Problemin tamsayı değişkenlere çevrim esnasında içerdiği yine on binde beş ya da daha küçük hata payları da bu çalışmada yaklaştırma metodu sonuçlarına

etki etmektedir. Fakat elde edilen model son derece hızlı ve verimli çalışmaktadır.

Endüstride büyük ölçekli ve kombinatorial yapıda optimizasyon problemleri söz konusu olduğunda birçok ar – ge çalışanı optimal çözümü bulmaya çalışmadan sadece uygulanabilir bir çözüm arayışına girmektedirler. Zira optimal çözümü bulma çabası uzun saatler alacak bir modelleme ve çözümleme sürecini beraberinde getirebilecektir. Oysa endüstriyel faaliyetler beklemeyecektir. Örneğin günde yüz binlerce tır seferi organize eden bir lojistik firması için saatler değil dakikalar bile önemlidir ve sıklıkla optimal maliyet bir kenara bırakılarak günün kaybedilmemesi için sadece uygulanabilir bir plan aranması söz konusu olmaktadır. Bu çalışmada geliştirilen model ise veri girişi basit bir yazılımla oldukça kolaylaştırılabilir ve çözümlenmesi bir saniye bile almayan ve sonuçları optimal bir çözümden sadece on bin üzerinden tek haneli rakamlarla ifade edilebilecek çözümler bulma imkanı sunmaktadır. Bu açıdan bakıldığında ortaya konan modelleme ve çözüm yaklaşımı son derece başarılı kabul edilebilecektir.

SONUÇ

İnsanoğlunun hükmetme güdüsü ve gücü elinde tutma arzusu nedeniyle süre gelen savaşlar, yine insanoğlunun zekası, araştırmacı kimliğini destekleyen merakı ile birleşince, dünyadaki insanoğlu hükümdarlığının oran olarak çok küçük bir kısmını kapsayan yakın tarihlerden itibaren çok büyük teknolojik gelişmeler yaşandı. Neyse ki sular durulduktan sonra insanoğlu imza attığı büyük sıçramaları bu kez hayatını kolaylaştırmak amacıyla kullanmaya karar verdi. Bu durum her ne kadar bu kez savaşın boyutunu ekonomik ve teknolojik bir platforma taşımış olsa da en azından dünya nüfusunun büyük bir kısmı için hayat kolaylaştı, yeni istihdam kapıları açıldı ve refah ve rahatlık umudu doğdu. Her ne kadar gelir dağılımı ve eşit fırsat konuları hala muğlak olsa da bunları tartışmanın yeri burası değildir.

Savaşta üstünlük için geliştirilmiş birçok teknolojinin günlük hayata girmesi ile önce göreceli rahat olan üretim sektörü, zamanla insanoğlunun sorgulayıcı ve rasyonel – çıkarıcı doğası ve teknolojik ürünlerin insan hayatı, kaynakları ve doğasını doğrudan etkilemesi nedeniyle ciddi bir fiyat ve kalite rekabeti ortamına sürüklendi. Bunun doğal bir uzantısı olarak ürün kalitesi, ürün ömrü, ürünlerde rastlanan hata olasılıkları ve fiyat, piyasada var olmak isteyen firmalar için kaçınılmaz stratejik parametreler haline geldiler.

Özellikle elektrik – elektronik ürünler piyasasında bu baskı çok yakından hissedildi. Klasik kalite kontrol yaklaşımları bir süre sonra yetersiz kalmaya başladı. Hata payları çok düşük iken buna bir de üretim ve kalite maliyetlerinin artarak eklenmesi ile fiyat rekabeti konusunda zorlanan büyük firmalar, kaçınılmaz olarak faal araştırma yoluna gitmek durumunda kaldılar. Böylece kurulan büyük organizasyonlar ve laboratuvarlar, üniversitelerin de araştırmaya faal katılımı ile ciddi bir araştırma sürecinin başlamasına ve artarak devamına ön ayak oldular.

Ürün testleri düşük hata paylarıyla çalışması gereken firmalar için giderek daha ciddi bir maliyet kaynağı haline geldi. Teknolojik gelişmelerle birlikte ürünlerin giderek karmaşılaşması ve fiziksel anlamda küçülmesi, fiziksel birçok testin uygulanmasında zorluklara neden oldu. Bu zorluklar daha karmaşık ve hassas test makinelerinin tasarlanmasına yol açarken maliyetlerin giderek yükselmesi firmaları yine hayati stratejik kararlar almak zorunda bıraktı. Dikkatle incelendiği zaman piyasa talepleri, ürünle ilgili oluşan şartlar ve rekabet ortamı, test stratejisi ve maliyetlerinin önemini tetiklerken, test süreci ile ilgili araştırmalara verilen önemin de giderek artmasını sağladı.

Bütün bu olaylar özellikle son 30 – 40 sene içinde bir çığ gibi büyüyen bir araştırma alanına hayat verdi. Özellikle mühendisler ve büyük üretici firmalar bu alanın teknik gereksinimlerini incelemekte ve hem teknolojiyi ötelemeye çalışırken hem de ilerleyen teknolojinin yol açtığı yeni problemlere çözüm aramaktadırlar.

Bu süreç içinde matematiksel yöntemlerin ve yöneylem araştırması tekniklerinin de zaman zaman kullanıldığı gözlenebilecektir. Bu alanda alınacak stratejik kararlar, bunları güdüleyen faktörler ve yönetsel bir bakış açısı da bazı çalışmalarda gözlenebilecektir. Fakat nadiren gözlenebilen bir durum, bu iki yaklaşımın bir arada gözetilerek strateji belirlenmesidir. Hem mühendislik perspektifi ile fiilen testlerin performansına yönelik kararları, hem de işletmecilik perspektifi ile yönetsel optimal kararların bir arada bulunacağı stratejik bir karar mekanizmasıdır. Böylesi girişimler bulunmakta ise de mevcut olanların genellikle çok karmaşık ve çözümsüzlüğe hızla giden bir problem yapısı ile karşılaşması söz konusudur. Haliyle bu alanda halen araştırmaya müsait çok fazla problem bulunmaktadır.

Bu çalışmanın amacı test süreçleri ile ilgili temel bilgileri ortaya koymak, henüz çok az incelenmiş test süreci dinamiklerine ışık tutmaktır. Bu alt bölümde hedeflenen amaçlara ne ölçüde ulaşıldığı ve geliştirilen metotların, elde edilen

bulguların ve halen araştırmaya açık olan noktaların bir değerlendirilmesi ve tartışması yapılacaktır.

Çalışma öncelikle işletme, endüstri mühendisliği ve yöneylem alanında çalışan bir bilim adamına, test tasarımı, test planlama, test stratejisi belirleme ve genel anlamda teste dair kararları verme üzerine yapılan geçmiş çalışmaları ve test araştırması alanının tarihsel gelişimini gözler önüne seren etkili bir literatür taraması içermektedir. Böylelikle hata saptanması ile 1960'lı yılların başında dikkat çekerek günümüze kadar geçen zamanda araştırma problemlerinin hangi alanlara kaydığını ve hangi ölçüde detaylandığını gözlemek mümkündür. Başka bir deyişle test araştırmasının evrim süreci gözler önüne serilmektedir.

Daha sonra ise giriş kısmında ve ilk ünitelerde verilen bilgilerin literatür taraması ile yeterince desteklendiği düşünüldüğünden doğrudan test süreci ile ilgili teknik bilgilere geçilmiştir. Burada teknik bilgilerden kastedilen şüphesiz bir elektrik – elektronik mühendisinin ya da bir test teknisyeninin yorumlayacağı türden bilgiler değildir. Yani makine mühendisliği ya da ürün detayları ile ilgili bilgiler bu çalışmanın kapsamında değildir. Burada kastedilen teknik bilgiler, test sürecinin bir matematiksel süreç olarak ifade edilmesi ve bu matematiksel ifadelerle dayalı bir karar verme metodunun belirlenmesini amaçlayan akış şemaları, ağ yapıları, optimizasyon modelleri ve denklem kümeleridir. Test sürecinin matematiksel ifade edilmesini amaçlayan çok ciddi çalışmalar bulunmaktadır. Fakat bu çalışmalar içinde burada amaçlanan yaklaşım ve ifade tarzına sahip olanların sayısı çok azdır.

Çalışma öncelikle basit test döngüsüne açıklamakta ve doğrusal denklemler seti şeklinde ifade etmektedir. Böyle bir tanıımı vermekteki amaç tek özelliği teste tabi ürünler söz konusu olduğunda olası test ve tamir makinelerinin seçim stratejisini belirleyecek bir optimizasyon modeli oluşturabilmektir. Takip eden alt bölümlerde böyle bir model geliştirilmiş ve örneklenmiştir. Anılan modelleme yaklaşımı test araştırması alanında bir ilktir. Burada birkaç noktanın üstünde durmak gereklidir. Ürünlerin giderek karmaşıklaştığı telekomünikasyon

sektöründe böyle bir modelin gerekliliği kendi başına bir soru işareti olup, uygulama alanı bulma ihtimali oldukça düşüktür. Fakat yaklaşım kendi içinde tutarlıdır ve matematiksel anlamda test sürecinin temel karar mekanizmasının belirlenmesinde önemli rol oynamaktadır. Ayrıca tasarım karmaşıklığı daha az olan ürünler için kullanılması son derece uygun ve faydalı olacak bir modeldir. Dikkat edileceği üzere model sadece test makineleri seçimi yapmaktadır. Eşzamanlı olarak alternatif tamir makineleri arasından da seçim yapacak bir model oluşturmak da mümkündür. Böyle bir problem ileride sürecelecek çalışmalarda ele alınacaktır.

Takip eden bölümde teste tabi birden fazla özelliği bulunan ürünlerin maruz kalacağı test süreçleri incelenmektedir. Bu testler bağımsız test ve etkileşimli test olarak iki başlık altında incelenmiştir. Bağımsız test durumu, teste tabi özelliklerin test sürecinden ayrı ayrı test edilmesini ve bu süreçte diğer özelliklerde herhangi bir değişim gözlenmemesini öngörmektedir. Böyle bir problem için üretilmiş olan model gerçek hayatta ciddi uygulama alanları bulabilecektir. Modelin yapısı birden fazla tek özellikli test sürecinin bir arada çalışması şeklinde olduğundan tek özellikli test hesaplarının önemi de bir kez daha vurgulanmalıdır. Geliştirilen optimizasyon modeli yine literatürde bir ilktir ve hem bilimsel, hem de uygulama açısından önem arz etmektedir. Bu yaklaşımla ilgili de sınırlılıklar ve tavsiyeler söz konusudur. Model basit görünmesine karşın alana aktarılacak nihai ürünler üzerine bir başarı oranı kısıtı içermektedir. Böyle bir kısıt bileşik olasılık hesabı içeren yapısı nedeniyle doğrusal olmayan bir denklemdir. Küçük problemlerde bile doğrusal olmayan kısıtlar, problemi çözümsüzlüğe götürebilir. Örneğin bu çalışmada model optimal sonucu hızla elde edemediği için amaç fonksiyonu üzerine kesme kısıtları kullanılmıştır. Belirli bir optimallik boşluğu (*optimality gap*) yakalandıktan sonra da çözümü iyileştirme çabası sonlandırılmıştır. Anılan modelin optimal olarak çözülmesi yine ileriye dönük açık araştırma problemlerinden bir diğeridir.

Etkileşimli çok özellikli test süreci ise bu tarz problemlerde karşılaşılabilecek en karmaşık ve zor problem tipidir. Hem modellenmesi hem de çözümlenmesi

oldukça zahmetli olan bu problemlerle çok az bilim adamı uğraşmış, hepsi de problemin sadece küçük bir bölümünü çözmeye odaklanmışlardır. Problemin karmaşıklığı ve zorluğu, doğrudan bu problemi incelemeyen bazı çalışmalarda da önemle belirtilmektedir. Problemin bazı parçalarını inceleyen çalışmalar, bu çalışmanın gerçekleştirilmesine de ışık tutmuş ve o çalışmaların bazıları aktif olarak bu çalışma kapsamında da kullanılmıştır. Etkileşimli test sürecinin detaylarının açıklandığı dördüncü bölüme ait alt bölümlerde bu çalışmalara ve ortaya koydukları bulgulara değinilmektedir.

Araştırma süreci içerisinde bu problemin gerçekten zor bir problem olduğu bilfiil gözlenmiştir. İlk modelleme çabaları esnasında karşılaşılan en ciddi zorluk, karmaşık yapıdaki etkileşimli test sürecinin tek bir genel model olarak tanımlanamamasında karşılaşılmıştır. Aslında problem tek bir genel model olarak ifade edilmiştir. İlgili bölümlerde de açıklandığı üzere bu genel model eşzamanlı çalışan bir ürün akış ağı, ürün karakteristik ağı ve test noktalarında çalışan bir yaklaştırma modülünden oluşmaktadır. Fakat yaklaştırma modülünü seri bir şekilde çözen AMPL programı, diğer iki ağda kullanılan ve test ekipmanı seçimin ve amaç fonksiyonunun tanımlanmasında hayati rol oynayan tamsayı değişkenleri çözememektedir. Diğer iki ağı seri bir şekilde çözen ve tamsayı değişkenlerde herhangi bir sorun yaratmayan OPL programı ile de yaklaştırma metodu sonuçlandırılmamaktadır. Böylece ilk olarak yapılması tasarlanan genel modeli açıkça ifade edilmiş değişkenlerle kurma yaklaşımı modellenebilir, fakat en iyi paket programlara dahi çözülemez bir probleme dönüşmüştür.

Bu noktada ileriye dönük yeni bir araştırma problemi önerisi söz konusudur. Yazılacak olası kodlarla bu iki programın etkileşimli çalıştırılması araştırmaya değer bir noktadır. Böyle bir yaklaşım sonuçlandırılabilirse aynı zamanda tamsayı ağ modelleri ile sürekli değer alan değişkenlerden oluşan doğrusal olmayan modeller bir arada çözülür hale gelecektir. Başka bir deyişle olasılık hesabı içeren optimizasyon modelleri oluşturulabilecek ve genel optimizasyon araştırmalarına büyük bir katkı sağlanacaktır. Mevcut probleme özel bir diğer açık araştırma problemi ise yerel arama metotlarının kullanılmasıdır. Böylelikle

arama uzayına başlanırken değişkenlerin seçimi ve verimsiz karar dallarından bir an önce kurtulmak mümkün hale gelebilecektir. Özellikle gerçek kodlu yerel arama modellerinin değişik hesap yaklaşımlarını bir arada kullanarak problemi çözülür kılma potansiyeli olduğu düşünülebilir. Bu da ileri yönelik araştırma tavsiyeleri arasında sayılabilecektir.

İlk yaklaşım sonuç vermediği için bu çalışma kapsamında alternatif bir çözüm yöntemi oluşturulmuştur. Tam senaryo yaklaşımı olarak tanımlanan bu çözüm yöntemi için kaba kuvvet arama yöntemlerinden ilham alınmıştır. Öncelikle probleme girdi olarak kabul edilecek ürün miktarı azaltılarak olası tüm test kombinasyonları birer senaryo haline getirilmiştir. Daha sonra bu senaryoların maliyetleri belirlenerek seçilecek senaryolara dayalı bir amaç fonksiyonu oluşturulmuştur. Tam senaryo yaklaşımı bu senaryolar arasından seçim yapan bir optimizasyon modeli olarak tanımlanarak çözülmüştür. Böylece ilk yaklaşımda planlanan tek tek test makinelerini ve bu makinelere dağılacak ürün miktarlarını belirleme sistemine eşdeğer bir çıktı elde edilmiştir.

Seçilen senaryolar değişik test safhalarında hangi makinelerin kullanılacağını ve bu makinelerde hangi miktarlarda ürün test edileceğini saptamaktadırlar. Aradaki modelleme açısından fark, modeli açıkça ifade edilmiş değişkenlerle kurmak yerine, bu değişkenlerin alacağı farklı değerlere dayanan senaryoların kullanımıdır.

Sonuçlar açısından ise iki durum söz konusudur. İlki problem çıktılarının her iki yaklaşımda da aynı olmalarıdır. Yani ulaşılmak istenen çözümlerden herhangi bir feragat durumu söz konusu değildir. İkinci durum ise ilk yaklaşımla çözülemeyen problemin, tam senaryo yaklaşımı ile bir saniyenin altında çözülebilir hale getirilmesidir. Bu açıdan tam senaryo yaklaşımı kullanılarak öncelikle test tasarımı problemlerinin nasıl çözüleceği ve ne sonuçlar vereceği noktası aydınlatılmıştır. Böyle bir modelleme sistemi ve elde edilen sonuçlar yine test araştırması alanında ilk kez bu çalışmada elde edilmektedir. Ayrıca çözüm yaklaşımı açısından benzer problemlerle karşılaşılması durumunda

kullanılabilecek bir yaklaşım önerilmiş olup, bu da yöneylem araştırmasına bir katkı olarak görülebilir.

Burada akla takılabilecek spesifik iki soru bulunmaktadır. Bunlardan ilki tüm alternatiflerin senaryolaştırılması ile ilintilidir. Teste tabi özellik sayısı arttığı takdirde oluşacak senaryoların da ciddi biçimde artması söz konusudur. Bu bir handicap gibi görülmemelidir. Zira olası kombinasyonların (senaryoların) oluşturulması ve sonuçlandırılması basit yazılımlarla kolaylıkla halledilebilecektir. Bu değerlerin tek tek elle hesaplanması gerekmemektedir. Zaten senaryo sonuçları elde edildikten sonra oluşturulan model ne kadar fazla senaryo içerirse içersin, senaryo sonuçlarını ihtiva eden tam senaryo optimizasyon modeli kısıtlar doğrusal yapıdadır. Unutulmamalıdır ki 10 değişkenli doğrusal olmayan kısıtlar bir problemi çözümsüzlüğe götürebilecekken, binlerce değişkenli doğrusal kısıtlardan oluşan bir problem yine saniyeler içinde çözülebilecektir. Bu açıdan bakıldığında büyük ölçekli bir problemde sadece senaryoların oluşturulması ve sonuçlandırılması konusu belirli bir miktar zahmet verecektir fakat model bir kez oluşturulduğunda çözümlenmesi konusunda bir sorun yaşanacağı düşünülmemektedir.

Diğer bir potansiyel soru ise girdi miktarı örneğin 500 üründen 5 ürüne düşürüldüğü takdirde modelin gerçeği ne kadar yansıttığı üzerine sorulabilir. Böyle bir problem de yine bir handicap oluşturmamaktadır. Bunun gerekçesi son derece basittir. Varsayımsal bir problemde bir senaryo kapsamında bir test makinesi olduğu düşünölsün. Bu makinenin de ikinci safha bir makine olduğu ve yine dördüncü bölümde aktarıldığı gibi iki makineden beslendiğı varsayölsün. Hedef makineye 100 adet ürün girdiğı takdirde bir uçta (100-0) senaryosu, diğer uçta ise (0-100) senaryosu bulunacaktır. Yani en uç ihtimallerde ürünler hedef makineye ya tamamen ilk kaynak makineden, ya da ikinci kaynak makineden gelecektir. İlk senaryo gerçekleşirse ikinci teste girecek ürünlerin durum vektörleri tamamen ilk kaynak makinede görölen durum vektörleri ile aynı, ikinci senaryo gerçekleşir ise de tamamen ikinci kaynak makinede görölen durum vektörleri ile aynı olacaktır. Şimdi teste giren ürün sayısının beş olduğu ve

kaynak makinelerinde bir önceki durumla aynı durum vektörleri bulunduğu varsayılın. Bu durumda ise uç senaryolar (5-0) ve (0-5) şeklinde gerçekleşecektir. Her iki varsayımsal problemde de test makinesi aynı kaynak makineden tüm girdilerini aldığı takdirde rastlanacak durum vektörleri aynı olacaktır. Ek 2'ye bakıldığı takdirde teste giren miktar sabitken uç senaryolar arasında doğrusala yakın bir girdi vektör değişimi gözlenecektir. Bu tamamen ağırlıklı ortalamalar kullanıldığından böyledir. Teste giren ürün sayısının 5 ya da 100 olması arasındaki tek fark da aralarında oluşacak kombinasyon sayısının farklı olmasıdır. Yine Ek 2'deki durum vektörleri incelenirse 5 üründe bile olasılıklar arasındaki farkın çok küçük olduğu gözlenecektir. Haliyle durum vektörlerinin küçük yuvarlamalara tabi olması dışında fazla bir fark olduğu söylenemeyecektir. Bu yüzden problem girdi miktarını azaltmanın genel çözüme, hatta belki büyük girdi miktarlı bir çözüme karşı fazla bir zayıflığı olduğu düşünülmemektedir. Aradaki olası fark ise yine ileri dönük bir araştırma problemi olarak tanımlanabilir.

Tam senaryo yöntemi ile ilgili de bir sınırlılığın belirtilmesinde fayda vardır. Daha önce de belirtildiği üzere yaklaştırma metodu bir kesin hesap metodu değildir. Hata payları on binde tek haneli rakamlarla ifade edilebilecek olsa da bir hata payı içermektedir. Yaklaştırma metodu üzerine bir de tam senaryo optimizasyon modelinde sürekli değişkenleri tamsayı değişkenlere çevirmekte kullanılan hata payları eklenmektedir. Burada öngörülen en yüksek hata oranı yirmi binde bir olacaksa da tüm bu hata payları akılda tutulmalıdır. Araştırmacı ya da kullanıcı için bu payları önemsiz kabul edildiği takdirde çözüm optimal çözüm olarak kabul edilebilir. Fakat bu çalışmada önemsiz kabul edilecek kadar küçük bu hata paylarının olası gerçek hayat uygulamalarında sorun yaşatacağı yorumlanırsa bu durumda hata terimleri ile oynanması önerilir. Yaklaştırma yöntemi için neler yapılacağı kısmına değinmenin yeri burası olmamakla birlikte tam senaryo yöntemi için bu çalışmada on binde bir kabul edilen hata paylarının el yordamıyla daraltılması önerilebilir.

Tam senaryo yöntemiyle ilgili geleceğe yönelik açık araştırma problemleri de bulunmaktadır. Örneğin bu çalışma birbirleri ile aynı özelliklere sahip tamir makinelerinin otomatik olarak test makinelerinin yanında bulunduklarını varsaymaktadır. Tamir makinelerinin de alternatiflerinin bulunması ve test makineleri gibi ve test makineleri ile eşzamanlı seçilebilmeleri başlı başına zorlu bir araştırma konusudur. Benzer şekilde test makinelerinden bağımsız tamir üniteleri bulunması ve seçimi de yine zorlu bir araştırma konusu olacaktır.

Bir diğer ileri yönelik araştırma problemi ise ürün akışı varsayımı üzerinedir. Bu çalışmada ilk safhada herhangi bir makinede teste giren ürünlerin bir sonraki safhada yine aynı makinede teste girmesine izin verilmemiştir. Bunun temel nedeni kullanılan yöntemler ve prensiplerin girdi kitlesi kaliteli ve teşhis ve tamir başarımı yüksek makinelerde daha iyi sonuçlar vereceğinin düşünülmesidir. Oysa ciddi derecede kötü girdiler ve ortalama makine performans karakteristikleri söz konusu olduğu takdirde bir testten çıkan ürün kitlesinin yeniden aynı teste girmesi makul bir strateji olabilir. Buradan hareketle böylesi bir kısıtın gevşetilerek problemin yeniden ele alınması söz konusu olabilir.

Tam senaryo yaklaşımında göz ardı edilen bir diğer nokta da makine kapasiteleridir. Ürün sayısı zaten düşük tutulmuşken bir de makine kapasitesi eklenmesi gereksiz görülmüştür. Böylece her test safhasında seçilecek makine sayısı birden fazla olmamaktadır. Gerçek hayat uygulamalarında ise durum farklı olabilir. O takdirde makine kapasitelerini de dikkate alacak şekilde modelleme yoluna gitmek de gelecek araştırmalar için bir fikir kabul edilmektedir.

Tüm bu değerlendirmeler ve öneriler dışında test araştırması kapsamında geliştirilen modellerin kullanımında önemle dikkate alınması gereken bazı ek hususlar bulunmaktadır. Bu hususların en önemlisi verilerle ilgilidir. Açıkça görüleceği gibi bu çalışmada varsayımsal sentetik veriler kullanılmıştır. Gerçek hayat uygulaması söz konusu olduğunda ise modelin hatalı sonuçlar ve yanlış yönlendirmeler ile verimsiz stratejiler seçilmesine neden olma potansiyeli,

tamamen veri kalitesine bağılıdır. Maliyet kalemleriyle ilgili verilerde ciddi problemler yaşanacağı düşünülmemektedir. En azından uygulayıcı kendisi için önemli maliyet kalemlerini tanımlayacak ve uygulayacak ise kendisi için tatmin edici sonuçlar elde edecektir. Mesela makine yatırımı ile ilgili maliyetler gerçek hayat problemlerinde kolaylıkla ve kesin değerlerle tahmin edilebilir. Fakat değişken maliyetler kaleminin içine hangi faktörlerin konacağı önemli bir problemdir. Bu çalışmada betimlenen maliyet kalemleri ve referans gösterilen çalışmalarda tanımlanan maliyet kalemleri dikkatle incelenmelidir. Bunun dışında öngörülecek maliyet değerlerinin gerçeği yansıtabilmesi de çok önemli bir konudur. Bu noktalarda yapılacak yanlış öngörüler ya da maliyet yapıları, seçilecek stratejiyi radikal şekilde etkileyecektir.

Verilerin güvenilirliği ile ilgili bir diğer önemli nokta da makinelere ait performans parametresi tahminleri için söz konusudur. Başka çalışmalarda olduğu gibi bu çalışmada da makine performans parametrelerinin kesin olarak bilindiği varsayılmaktadır. Burada sunulan metotların kullanımından verim alabilmek için bu değerlerin gerçekçi tahmin edilmesi bir ön şarttır. Aksi takdirde önerilecek strateji yine ciddi hata payları içerebilir. Benzer bir durum girdi ürünlerinin durum vektörleri için de geçerlidir. Tahmin edilmesi göreceli olarak daha kolay olsa da, bu parametrelerin hatalı değerler olarak modele girdi oluşturması yine sonuçların güvenilirliğini düşürecektir.

Akla gelen hemen hemen tüm değerlendirmeler, sınırlamalar ve ileri dönük araştırma problemleri bunlardan ibarettir. Bu çalışmanın hem bilim hem de endüstri dünyasına faydalı olması ve test tasarımı temelli problemler için araştırmacılara fikir vermesi ve ışık tutması umulmaktadır. Teknolojideki gelişim de zaman gibi durmamakta ve kendi yolunu çizmektedir. Ürünlerin karmaşılaşması, test stratejileri seçimi, kalite ve ürün ömrü gibi faktörler her an daha da önemli hale gelmektedir. Bu yüzden test araştırması da gün geçtikçe ciddi ve mecburi bir araştırma alanı haline gelmektedir. Umulan odur ki bilim ve teknoloji insanlığın huzur ve refahına hizmet etsin ve teknolojik gelişmeler ve akademik çalışmalar insanoğlunun hüsrânını değil mutluluğunu sağlasın.

KAYNAKÇA

- ABADIR, M. "Economics Modeling of Multichip Modules Testing Strategies." *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B*, 21, 4, November 1998: 360 – 370.
- ABADIR, M. S., ve H. K. REGHBATI. "LSI Testing Techniques." *IEEE Micro*, February 1983: 34 – 51.
- ABADIR, M. S., ve H. K. REGHBATI. "Functional Test Generation for Digital Circuits Described Using Binary Decision Diagrams." *IEEE Transactions on Computers*, C-35, 4, April 1986: 375 – 379.
- ADAMS, J. B. ve D. S. HOCHBAUM. "A New and Fast Approach to Very Large Scale Integrated Sequential Circuit Test Generation." *Operations Research*, 45, 6, November – December 1997: 842- 856.
- AGRAWAL, V. D. "When to Use Random Testing." *IEEE Transactins on Computers*, C-27, 11, November 1978: 1054 – 1055.
- AGRAWAL, V. D., S. C. SETH ve P. AGRAWAL. "Fault Coverage Requirements in Production Testing of LSI Circuits." *IEEE Journal of Solid State Circuits*, SC – 17, 1, February 1982: 57 – 61.
- AHO, A. V., A. T. DAHBURA, D. LEE ve M.Ü. UYAR. "An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours." *IEEE Transactions on Communications*, 39, 11, November 1991: 1604 – 1615.

- ALFREDSSON, P. "Optimization of Multi – Echelon Repairable Item Inventory Systems With Simultaneous Location of Repair Facilities." *European Journal of Operational Research*, 99, 1997: 584 – 595.
- ALI, L., R. SIDEK, I. ARIS, R. WAGIRAN ve M. A. M. ALI. "Design of a SOC for Low Cost IC Testing." *Proceedings of the IEEE ICSE2004*, 2004: 374 – 378.
- AMBLER, A. "The Cost of Test." *Proceedings of the IEEE Automatic Test Conference AUTOTESCON*, 2001: 515 – 523.
- AMBLER, T. ve M. ABADIR. "Design and Test Economics – An Extra Dimension" *IEEE Design and Test of Computers*, July – September 1997: 15 – 16.
- BARBAGALLO, S., ve M. L. BODONI, A. BENSO, S. CHIUSANO, P. PRINETTO. "Testing Embedded Memories in Telecommunications Systems." *IEEE Communications Magazine*, June 1999: 84 – 89.
- BARRETT, N., S. MARTIN ve C. DISLIS. "Test Process Optimization: Closing the Gap in the Defect Spectrum." *Proceedings of the IEEE ITC International Test Conference*, 1999: 124 – 129.
- BEDSOLE, J., R. RAINA, A. CROUCH ve M. S. ABADIR. "Very Low Cost Testers: Opportunities and Challenges." *IEEE Design & Test of Computers*, International Test Conference, September – October 2001: 60 – 69.
- BENOIST, T., ve E. GAUDIN, B. ROTTEMBOURG. "Constraint Programming Contribution to Benders Decomposition: A Case Study." *Proceedings of CP'02*, 2002: 603 – 617.

- BERTRAND, Y., M. – L. FLOTTES, L. BALADO, J. FIGUERAS, A. BIASIZZO, F. NOVAK, S. DI CARLO, P. PRINETTO, N. PRICOBÌ, H-J. WUNDERLICH ve J-P. VAN DER HEYDEN. "Test Engineering Education in Europe: The EuNICE – Test Project." *Proceedings of the 2003 IEEE International Conference on Microelectronic Systems Education (MSE'03)*, 2003.
- BIASIZZO, A., A. ŽUŽEK ve F. NOVAK. "Sequential Diagnosis with Asymmetrical Tests." *The Computer Journal*, 41, 3, 1998: 163 – 170.
- BISDORFF, R. ve S. LAURENT. "Industrial Linear Optimization Problems Solved by Constraint Logic Programming." *European Journal of Operations Research*, 84, 1995: 82 – 95.
- BLAIR, D. W. "Technical and Economic Requirements of Integrated SOC Testing." *Proceedings of the IEEE / CPMT / SEMI International Electronics Manufacturing Technology Symposium*, 2003: 215 – 219.
- BOCKMAYR, A. ve T. KASPER. "A Unifying Framework for Integer and Finite Domain Constraint Programming." *Research Report for MPI*, MPI I 97 2 008, August 1997.
- BOCKMAYR, A. ve N. PISARUK. "Detecting Infeasibility and Generating Cuts for MIP using CP." *Proceedings of the CPAIOR'03*, 2003.
- BOSE, S. ve V. D. AGRAWAL. "Fault Coverage Estimation for Non – Random Functional Input Sequences." *Proceedings of the IEEE ITC International Test Conference*, 2006: 1 – 10.
- BROWN, M. ve F. PROSCHAN. "Imperfect Repair." *Journal of Applied Probability*, 20, 1983: 851 – 859.

- BUTLER, D. A., ve G. J. LIEBERMAN. "Inspection Policies for Fault Location." *Operations Research*, 32, 3, Reliability and Maintainability, May – June 1984: 566 – 574.
- BUTTERWORTH, R. "Some Reliability Fault – Testing Models." *Operations Research*, 20, 2, March – April 1972: 335 – 343.
- CAMERANA, E. A., C. GRACIA ve J. M. CABRERA SIXTO. "A Mixed Integer Linear Programming Machinery Selection Model for Multifarm Systems." *Biosystems Engineering*, 87, 2, 2004: 145 – 154.
- CASEAU, Y. ve F. LABURTHE. "Improving Branch and Bound for Jobshop Scheduling with Constraint Propagation." *Combinatorics and Computer Science*, 1995: 129 – 149.
- CHELBI, A., ve D. AIT-KADI. "An Optimal Inspection Strategy for Randomly Failing Equipment." *Reliability Engineering and Systems Safety*, 63, 1999: 127 – 131.
- CHEN, T. J. "Quantitative Selection of Inspection Plans for Variation Risk Management." Yüksek Lisans Tezi, Berkeley, USA: University of California, 1997.
- CHEN, T. Y., ve M. F. LAU. "A New Heuristic for the Test Suite Reduction." *Information and Software Technology*, 40, 1998: 347 – 354.
- CHEN, T., V.-K. KIM ve M. TEGETHOFF. "IC Yield Estimation at Early Stages of the Design Cycle." *Microelectronics Journal*, 30, 1999a: 725 – 732.

- CHEN, T., V.-K. KIM ve M. TEGETHOFF. "IC Manufacturing Test Cost Estimation at Early Stages of the Design Cycle." *Microelectronics Journal*, 30, 1999b: 733 – 738.
- CHIEN, C. F., H. J. WANG ve M. WANG. "A UNISON Framework for Analyzing Alternative Strategies of IC Final Testing for Enhancing Overall Operational Effectiveness." *International Journal of Production Economics*, 107, 2007: 20 – 30.
- CHIEN, C., ve J. WU. "Analyzing repair decisions in the site imbalance problem of semiconductor test machines." *IEEE Transactions on Semiconductor Manufacturing*, 16, 4, 2003: 704 – 711.
- CHEVALIER, P. B., ve L. M. WEIN. "Inspection for Circuit Board Assembly." *Management Science*, 43, 9, September 1997: 1198 – 1213.
- CHOU, R. M., K. K. SALUJA ve V. D. AGRAWAL. "Scheduling Tests for VLSI Systems Under Power Constraints." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5, 2, June 1997: 175 – 185.
- COHEN, D. M., S. R. DALAL, J. PARELIUS ve G. C. PATTON. "The Combinatorial Design Approach to Automatic Test Generation." *IEEE Software*, September, 1996: 83 – 88.
- COTA, E., L. CARRO, A. ORAILOGLU ve M. LUBASZEVSKI. "Test Plan and Design Space Exploration in a Core – Based Environment." *Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE.02)*, 2002.

- COX, P., P. YANG ve P. CHATTERJEE. "Statistical Modeling for Efficient Parametric Yield Estimation of MOS VLSI Circuits." *Proceedings of IEEE International Electron Devices Meeting IEDM 83*, 1983: 242 – 245.
- CUI, L., W. KUO, H. T. LOH ve M. XIE. "Optimal Allocation of Minimal & Perfect Repairs Under Resource Constraints." *IEEE Transactions on Reliability*, 53, 2, June 2004: 193 – 199.
- DAS, D. V., S. C. SETH ve V. D. AGRAWAL. "Estimating the Quality of Manufactured Digital Sequential Circuits." *Proceedings of the IEEE ITC International Test Conference*, 1991: 210 – 217.
- DAS, D. V., S. C. SETH ve V. D. AGRAWAL. "Accurate Computation of Field Reject Ratio Based on Fault Latency." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1, 4, December 1993: 537 – 545.
- DEAR, I. D., C. DISLIS, S. C. LAU, J. R. MILES ve A. P. AMBLER. "Hierarchical Testability Measurement and Design for Test Selection by Cost Prediction." *Proceedings of the 1st IEEE European Test Conference*, 1989: 50 – 57.
- DEAR, I. D., C. DISLIS, A. P. AMBLER ve J. DICK. "Economic Effects in Design and Test." *IEEE Design and Test of Computers*, December, 1991: 64 – 77.
- DESHAYES, H. "Cost of Test Reduction." *Proceedings of IEEE ITC International Test Conference*, 1998: 265 – 270.
- DICK, J. H., E. TRISCHLER, C. DISLIS ve A. P. AMBLER. "Sensitivity Analysis in Economic Based Test Strategy Planning." *Journal of Electronic Testing: Theory and Applications (JETTA)*, 5, May 1994: 239 – 252.

- DING, J., B. S. GREENBERG ve H. MATSUO. "Repetitive Testing Strategies When the Testing Process Is Imperfect." *Management Science*, 44, 10, October 1998: 1367 -1378.
- DISLIS, C., A. P. AMBLER, I. D. DEAR ve J. H. DICK. "Economics in Test an Design." *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors ICCD'93*, 1993a: 384 – 387.
- DISLIS, C., I. D. DEAR, J. R. MILES, S. C. LAU ve A. P. AMBLER. "Cost Analysis of Test Method Environments." *Proceedings of International Test Conference*, 1989a: 875 – 883.
- DISLIS, C., I. D. DEAR, S. C. LAU, J. R. MILES ve A. P. AMBLER. "Hierarchical Test Strategy Planning Based on Cost Evaluation." *IEE Colloquium on Computer Aided Test and Diagnosis*, October 1988: 7/1 – 7/8.
- DISLIS, C., I. D. DEAR, S. C. LAU ve A. P. AMBLER. "Knowledge Based Test Strategy Planning." *IEE Colloquium on Algorithmic and Knowledge Based CAD for VLSI*, November 1989b: 2/1 – 2/5.
- DISLIS, C., J. DICK ve A. P. AMBLER. "Algorithms for Optimised Test Strategy Selection." *Proceedings of IEEE ITC International Test Conference*, 1993b: 383 – 391.
- DISLIS, C., J. H. DICK, I. D. DEAR, I. N. AZU ve A. P. AMBLER. "Economic Modeling for the Determination of Test Strategies for Complex VLSI Boards." *Proceedings of International Test Conference*, 1993c: 210 – 217.

- DISLIS, C., A. F. ALANI ve I. P. JALOWIECKI. "A Framework for the Management of MCM Test Strategies." *Proceedings of IEEE International Conference of Multichip Modules*, 1997: 272 – 277.
- DONOVAN, J., ve E. MURPHY. "Total System Reliability: Integrated Model for Growth and Test Termination." *Quality and Reliability Engineering International*, 21, 2005: 329 – 344.
- DRECHSLER, R., W. KUNZ, D. STOFFEL ve A. ŽUŽEK. "Decision Diagrams and AND / OR Graphs for Design Automation Problems." *Proceedings of the IEEE International Conference on Information, Communications and Signal Processing ICICS'97*, September, 1997: 246 – 250.
- DUFFUAA, S. O. ve H. J. AL – NAJJAR. "A General Inspection Plan for Critical Multicharacteristic Components." *International Journal of Production Research*, 35, 10, 1997: 2723 – 2736.
- EBADI, Z. S. ve A. IVANOV. "Design of an Optimal Test Access Architecture Using a Genetic Algorithm." *Proceedings of the 10th IEEE Asian Test Symposium*, 2001: 205 – 210.
- EMMONS, H. ve G. RABINOWITZ. "Inspection Allocation for Multistage Deteriorating Production Systems." *IIE Transactions*, 34, 2002: 1031 – 1041.
- EVANS, A. C. "Applications of Semiconductor Test Economics, and Multisite Testing to Lower Cost of Test." *Proceedings of the IEEE ITC International Test Conference*, 1999: 113 – 123.

FERRARIO, J., R. WOLF, S. MOSS, ve M. SLAMANI. "A Low – Cost Test Solution for Wireless Phone RFICs." *IEEE Communications Magazine*, September, 2003: 82 – 88.

FERRIS – PRABHU, A. V. "Role of Defect Size Distribution in Yield Modeling." *IEEE Transactions on Electron Devices*, ED – 32, 9, September 1985: 1727 – 1736.

FINCH, T. "LSI Test Strategies: Present and Future." *Proceedings of the IEEE Solid – State Circuits Conference ISSCC*, February 1975.

FINKELSHTEIN, A., Y. T. HERER, T. RAZ ve I. BEN – GAL. "Economic Optimization of Off – Line Inspection in a Process Subject to Failure and Recovery." *IIE Transactions*, 37, 2005: 995 – 1009.

FIRSTMAN, S. I., ve B. GLUSS. "Otimum Search Routines for Automatic Fault Location." *Operations Research*, 8, 4, July – August 1960: 512 – 523.

FISHER, E., S. FORTUNE, M. GLADSTEIN, S. GOYAL, W. B. LYONS, J. MOSHER ve G. WILFONG. "Economic Modeling of Global Test Strategy I: Mathematical Models." *Bell Labs Technical Journal*, 12, 1, May 2007a: 161 – 173.

FISHER, E., S. FORTUNE, M. GLADSTEIN, S. GOYAL, W. B. LYONS, J. MOSHER ve G. WILFONG. "Economic Modeling of Global Test Strategy II: Software System and Examples." *Bell Labs Technical Journal*, 12, 1, May 2007b: 175 – 186.

FLEHINGER, B. J. "Product Test Planning for Repairable Systems." *Technometrics*, 7, 4, November 1965: 485 – 494.

- GAITONDE, D. D., ve D. M. H. WALKER. "Test Quality and Yield Analysis Using the DEFAM Defect to Fault Mapper." *Digest of Technical Papers of the IEEE/ACM International Conference of Computer – Aided Design ICCAD'93*, 1993: 202 – 205.
- GAITONDE, D. D., D. M. H. WALKER ve W. MALY. "Accurate Yield Estimation of Circuits with Redundancy." *Proceedings of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, 1995: 155 – 163.
- GAITONDE, D. D., J. KHARE, D. M. H. WALKER ve W. MALY. "Estimation of Reject Ratio in Testing of Combinatorial Circuits." *Proceedings of the 11th Annual IEEE VLSI Test Symposium*, 1993: 319 – 325.
- GAITONDE, D. D., W. MALY ve D. M. H. WALKER. "Fault Probability Prediction for Array Based Designs." *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1996: 30 – 38.
- GARG, V., D. J. STOGNER, C. ULMER, D. E. SCHIMMEL, C. DISLIS, S. YALAMANCHILI ve D. S. WILLS. "Early Analysis of Cost / Performance Trade – Offs in MCM Systems." *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B*, 20, 3, August 1997: 308 – 319.
- GIANFAGNA, M. A. "A Unified Approach to Test Data Analysis." *Proceedings of the 15th Conference of Design Automation*, June 1978: 117 – 124.
- GLUSS, B. "An Optimum Policy for Detecting a Fault in a Complex System." *Operations Research*, 7, 4, July – August 1959: 468 – 477.

- GOEL, P. "Test Generation Costs Analysis and Projections." *Proceedings of the 17th Conference of Design Automation*, June 1980: 77 – 84.
- GOVINDARAJU, K. "Design of Minimum Average Total Inspection Sampling Plans." *Communications in Statistics – Simulation and Computation*, 34, 2005: 485 – 493.
- GOYAL, S., ve J. H. MOSHER. "An Improved Test Process Model for Cost Reduction." *Bell Labs Technical Journal*, 11 (1), 2006: 173 – 190.
- HASHEMPOUR, H., F. J. MEYER, F. LOMBARDI ve F. KARIMI. "Hybrid Multisite Testing at Manufacturing." *Proceedings of the IEEE ITC International Test Conference*, 2003: 927 – 936.
- HAWKINS, C. F., H. T. NAGLE, R. R. FRITZEMEIER ve J. R. GUTH. "The VLSI Circuit Test Problem – A Tutorial." *IEEE Transactions on Industrial Electronics*, 36, 2, May 1989: 111 – 116.
- HEINEKEN, H. T., J. KHARE ve W. MALY. "Yield Loss Forecasting in the Early Phases of the VLSI Design Process." *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1996: 27 – 30.
- HERER, Y. T. ve T. RAZ. "Further Results in the Optimal Policy for Imperfect Inspection in Discrete Time." *Production Planning and Control*, 8, 4, 1997: 377 – 384.
- HUSS, S. D., ve R. S. GYURCSIK. "Optimal Ordering of Analog Integrated Circuit Tests to Minimize Test Time." *Proceedings of the 28th ACM / IEEE Design Automation Conference*, 1991: 494 – 499.

- IYENGAR, V., K. CHAKRABARTY, M. D. KRASNIEWSKI ve G. N. KUMAR. "Design and Optimization of Multi – Level TAM Architectures for Hierarchical SOC's." *Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)*, 2003.
- JAIN, S. K. ve V. D. AGRAWAL. "Statistical Fault Analysis." *IEEE Design & Test*, 1985: 38 – 44.
- JERVAN, G., Z. PENG ve R. UBAR. "Test Cost Minimisation for Hybrid BIST." *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'2000)*, 2000: 283 – 291.
- JIANG, W. ve B. VINNAKOTA. "Defect – Oriented Test Scheduling." *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, 9, 3, June 2001: 427 – 438.
- JIN, S. H., Y.-K. YEO, I. MOON, Y. CHUNG ve I.-W. KIM. "Equipment Selection for the System Unavailability of Jacketed Reactors with Discrete Cost Data." *Journal of Loss Prevention in the Process Industries*, 16, 2003: 443 – 448.
- KAKADE, V., J. F. VALENZUELA ve J. S. SMITH. "An Optimization Model for Selective Inspection in Serial Manufacturing Systems." *International Journal of Production Research*, 42, 18, September 2004: 3891 – 3910.
- KAPUR, R., R. CHANDRAMOULI, ve T. W. WILLIAMS. "Strategies for Low – Cost Test." *IEEE Design and Test of Computers*, November – December 2001, 47 – 54.
- KHOICHE, A., R. KAPUR, D. ARMSTRONG, T. W. WILLIAMS, M. TEGETHOFF ve J. RIVOIR. "A New Methodology for Improved Tester Utilization." *Proceedings of ITC International Test Conference*, 2001: 916 – 923.

- KOTÁSEK, Z., D. MIKA ve J. STRNADEL. "Test Scheduling for Embedded Systems." *Proceedings of the IEEE Euromicro Symposium on Digital System Design (DSD'03)*, 2003.
- LARSSON, E. ve H. FUJIWARA. "Optimal System – on – Chip Test Scheduling." *Proceedings of the 12th Asian Test Symposium (ATS'03)*, 2003a.
- LARSSON, E. ve H. FUJIWARA. "Test Resource Partitioning and Optimization for SOC Designs." *Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)*, 2003b.
- LARSSON, E., K. ARVIDSSON, H. FUJIWARA ve Z. PENG. "Integrated Test Scheduling, Test Parallelization and TAM Design." *Proceedings of the 11th Asian Test Symposium (ATS'02)*, 2002.
- LEE, D., I. PARK, E. VOLKERINK ve J REARICK. "Empirical Validation of Yield Recovery Using Idle Cycle Insertion." *IEEE Design & Test of Computers*, July – August, 2007: 362 – 372.
- LEE, J. ve S. UNNIKRISHNAN. "Planning Quality Inspection Operations in Multistage Manufacturing Systems with Inspection Errors." *International Journal of Production Research*, 36, 1, 1998: 141 – 155.
- LITTLE, J. "Integer Programming, Constraint Logic Programming And Their Collaboration In Solving Discrete Optimisation Problems." Doktora Tezi, Uxbridge, UK: Brunel University, 2000.

- LITTLE, J., S. GOYAL, P. CREED, S. BERRY ve D. COKELY. "Thermal Test Scheduling Using Constraint Programming." *Proceedings of the INCOM*, May 2006.
- LUSTIG, I. J. ve J. F. POUGET. "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming." *INTERFACES*, 31, 6, November – December 2001: 29 – 53.
- MAILHARRO, D. "A Classification and Constraint – Based Framework for Configuration." *Artificial Intelligence for Engineering and Design, Analysis and Manufacturing*, 12, 1998: 383 – 397.
- MALY, W. "The Design and Test Cost Problem" *IEEE Design and Test of Computers*, 2001.
- MALY, W., A. J. STROJWAS ve S. W. DIRECTOR. "VLSI Yield Prediction and Estimation: A Unified Framework." *IEEE Transactions on Computer Aided Design*, CAD- 5, 1, January 1986: 114 – 130.
- MARAVELIAS, C. T. ve I. E. GROSSMAN. "A Hybrid MILP / CP Decomposition Approach for the Scheduling of Batch Plants." *Proceedings of the CPAIOR'03*, 2003.
- MARTIN, S., R. BLACK, C. DISLIS ve D. FARREN. "The Evolution of a System Test Process." *Proceedings of the IEEE ITC International Test Conference*, 1999: 680 – 688.
- MAXWELL P. C. ve R. C. AITKEN. "Test Sets and Reject Rates: All Fault Coverages Are Not Created Equal." *IEEE Design and Test of Computers*, March 1993: 42 – 51.

- MERGENTHALER, W., B. MAUERSBERG, J. FELLER, L. J. STUEHLER, W. T. O'GRADY Jr. ve J. S. LEDFORD. "Application of the Simulated Annealing Local Search Technique to Problems of Redundancy Elimination in Functional and Parametric Tests of Integrated Circuits." *Mathematics and Computers in Simulation*, 62, 2003: 443 – 451.
- MICHEL, C., ve R. D. REINOSA. "Test Strategy Cost Model Innovations." *Proceedings of the IEEE ITC International Test Conference*, 2004: 382 – 394.
- MILANO, M. "Integration of CP and OR for Combinatorial Optimization." *Tutorial PACLP2000*, April 2000.
- MILOR, L. ve A. SANGIOVANNI-VINCENTELLI. "Optimal Test Set Designs for Analog Circuits." *Digest of Technical Papers of the IEEE International Conference on Computer – Aided Design ICCAD'90*, 1990: 294 – 297.
- MOORE, T. J. "A Test Process Optimization and Cost Modeling Tool." *Proceedings of the IEEE ITC International Test Conference*, 1994: 103 – 110.
- NACHLAS, J. A., S. R. LONEY ve B. A. BINNEY. "Diagnostic – Strategy Selection for Series Systems." *IEEE Transactions on Reliability*, 39, 3, August, 1990: 273 – 280.
- NEMI Board Essembly Technical Group Test Strategy Project. "Maufacturing Test Strategy Cost Model User's Guide." March, 2003.
- NOVAK, F., A. BIASIZZO ve M. ŽELE. "Sequential Fault Diagnosis in System Maintenance." *Proceedings of the IEEE Computers in Design, Manufacturing and Production CompEuro'93*, 1993: 119 – 124.

- OPPERMANN, M., W. SAUER ve W. WOHLRABE. "Optimization of Inspection Strategies by Use of Quality Cost Models and SPC." *24th International Spring Seminar on Electronics Technology (IEEE ISSE 2001)*, May 2001: 293 – 297.
- OPPERMANN, M., W. SAUER ve W. WOHLRABE. "Optimization of Quality Costs." *Robotics and Computer Integrated Manufacturing*, 19, 2003a: 135 – 140.
- OPPERMANN, M., W. SAUER, W. WOHLRABE ve T. ZERNA. "New Quality Cost Models to Optimize Inspection Strategies." *IEEE Transactions on Electronics Packaging Manufacturing*, 26, 4, October 2003b: 328 – 337.
- OTTOSSON, G. "Integration of Constraint Programming and Integer Programming for Combinatorial Optimization." Doktora Tezi, Uppsala, SWEDEN: Uppsala University, 2000.
- PATTIPATI, K. R. ve M. DONTAMSETTY. "On a Generalized Test Sequencing Problem." *IEEE Transactions on Systems, Manufacturing and Cybernetics*, 22, 2, March / April 1992: 392 – 396.
- PATTIPATI, K. R. ve M. G. ALEXANDRIDIS. "Application of Heuristic Search and Information Theory to Sequential Fault Analysis." *IEEE Transactions on Systems, Manufacturing and Cybernetics*, 20, 4, July / August 1990: 872 – 887.
- PENN, M. ve T. RAVIV. "Optimizing the Quality Control Station Configuration." *Naval Research Logistics*, 54, 2007: 301 – 314.

- PLESKACZ, W. A. "Yield Estimation of VLSI Circuits with Downscaled Layouts." *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems DFT'99*, November 1999: 55 – 60.
- PLESKACZ, W. A., W. MALY ve H. T. HEINEKEN. "Detection of Yield Trends." *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1997: 62 – 68.
- POMERANZ, I., ve S. M. REDDY. "On Maximizing the Fault Coverage for a Given Test Length Limit in a Synchronous Sequential Circuit." *Proceedings of the 21st IEEE VLSI Test Symposium (VTS' 03)*, 2003.
- POUGET, J., E. LARSSON ve Z. PENG. "SOC Test Time Minimization Under Multiple Constraints." *Proceedings of the 12th Asian Test Symposium (ATS'03)*, 2003.
- PYNN, C. T. "Analyzing Manufacturing Costs." *IEEE Design and Test of Computers*, July – September 1997: 36 – 40.
- RAZ, T., Y. T. HERER ve A. GROSFELD – NIR. "Economic Optimization of Off – Line Inspection." *IIE Transactions*, 32, 2000: 205 – 217.
- REINOSA, R. D., ve C. MICHEL. "Manufacturing Test Cost Model." *Proceedings of the IEEE ITC International Test Conference*, 2003.
- RIVOIR, J. "Lowering Cost of Test: Parallel Test or Low Cost ATE?" *Proceedings of the 12th IEEE Asian Test Symposium (ATS'03)*, 2003.

- RIVOIR, J. "Parallel Test Reduces Cost of Test More Effectively Than Just a Cheap Tester." *Proceedings of the IEEE / SEMI International Electronics Manufacturing Technology Symposium*, 2004.
- ROBINSON, G. D. "Test Program Development Using Multiple Test Strategies." *Proceedings of the IEEE Automatic Testing Conference AUTOTESTCON'89*, 1989: 9 – 18.
- RODOŠEK, R., M. G. WALLACE ve M. T. HAJIAN. "A New Approach to Integrating Mixed Integer Programming and Constraint Logic Programming." *Baltzer Journals*, November, 1998.
- ROSSI, R., S. A. TARIM, B. HNICH ve S. PRESTWICH. "Multi-Component Testing in Telecommunications." *Proceedings of the European Conference on Operational Research EURO XXI*, July 2006.
- RUDNICK, E. M., ve J. H. PATEL. "Efficient Techniques for Dynamic Test Sequence Compaction." *IEEE Transactions on Computers*, 48, 3, March 1999.
- RUDNICK, E. M., J. H. PATEL, G. S. GREENSTEIN ve T. M. NIERMANN. "Sequential Circuit Test Generation in a Genetic Algorithm Framework." *Proceedings of the 31st ACM / IEEE Design Automation Conference*, 1994: 698 – 704.
- RUDNICK, E. M., J. H. PATEL, G. S. GREENSTEIN ve T. M. NIERMANN. "A Genetic Algorithm Framework for Test Generation." *IEEE Transactions On Computer-Aided Design of Integrated Circuits And Systems*, 16, 9, September 1997: 1034 – 1044.

- SCHEFFLER, M., D. AMMANN, A. THIEL, C. HABIGER ve G. TRÖSTER. "Modeling and Optimizing the Costs of Electronic Systems." *IEEE Design and Test of Computers*, July – September 1998: 20 – 26.
- SETH, S. C. ve V. D. AGRAWAL. "Forecasting Rejects Rate of Tested LSI Chips." *IEEE Electron Device Letters*, EDL – 2, 11, November, 1981: 286 – 287.
- SETH, S. C., V. D. AGRAWAL ve H. FARHAT. "A Statistica Theory of Digital Circuit Testability." *IEEE Transactions on Computers*, 39, 4, April 1990: 582 – 586.
- SEPÚLVEDA, A., ve J. A. NACHLAS. "A Simulation Approach to Multivariate Quality Control." *21st International Conference on Computers and Industrial Engineering*, 33, 1-2, 1997: 113 – 116.
- SHI, Z. ve P. SANDBORN. "Optimization of Test / Diagnosis / Repair Location(s) and Characteristics in Electronic System Assembly Using Real – Coded Genetic Algorithms." *Proceedings of the IEEE ITC International Test Conference*, 2003: 937 – 946.
- SOUSA, J. J. T. ve J. P. TEXEIRA. "Defect Level Estimation for Digital ICs." *Proceedings of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, 1992: 32 – 41.
- STAPPER, C. H. "Yield Statistics for Large Area ICs." *IEEE International Solid State Circuits Conference*, 1986: 168 – 169.
- STAPPER, C. H. "Improved Yield Models for Fault – Tolerant Memory Chips." *IEEE Transactions on Computers*, 42, 7, July 1993: 872 – 881.

- STAPPER, C. H., F. M. ARMSTRONG ve K. SAJI. "Integrated Circuit Yield Statistics." *Proceedings of the IEEE*, 71, 4, April 1983: 453 – 470.
- STAPPER, C. H, ve R. J. ROSNER. "Integrated Circuit Yield Management and Yield Analysis: Development and Implementation." *IEEE Transactions on Semiconductor Manufacturing*, 8, 2, May 1995: 95 – 102.
- TARIM, S.A. ve I. MIGUEL. "A Hybrid Benders' Decomposition Method for Solving Stochastic Constraint Programs with Linear Recourse." *Proceedings of the CSCLP 2005: Joint Annual Workshop on ERCIM/CoLogNet on Constraint Solving and Constraint Logic Programming*, 2005: 148 – 162.
- TEGETHOFF, M. M. V., ve T. W. CHEN. "Defects, Fault Coverage, Yield and Cost, in Board Manufacturing." *Proceedings of the IEEE International Test Conference*, 1994: 539 – 547.
- THATCHER, C. W. "Design – For – Testability Economics." *Proceedings of the IEEE ITC International Test Conference*, 1993.
- TRICHY, T., P. SANDBORN, R. RAGHAVAN ve S. SAHASRABUDHE. "A New Test/Diagnosis/Rework Model for Use in Technical Cost Modeling of Electronic Systems Assembly." *Proceedings of the IEEE ITC International Test Conference*, 2001: 1108 – 1117.
- VAN VOLSEM, S., W. DUALLERT ve H. VAN LANDEGHEM. "An Evolutionary Algorithm and Discrete Event Simulation for Optimizing Inspection Strategies for Multi – Stage Processes." *European Journal of Operational Research*, 179, 2007: 621 – 633.

- VISWANADHAM, N., S. M. SHARMA ve M. TANEJA. "Inspection Allocation in Manufacturing Systems Using Stochastic Search Techniques." *IEEE Transactions on Systems, Manufacturing and Cybernetics – Part A: Systems and Humans*, 26, 2, March 1996: 220 – 230.
- VOLKERINK, E. H., A. KHOICHE, L. A. KAMAS, J. RIVOIR ve H. G. KERKHOFF. "Tackling Test Trade – Offs from Design, Manufacturing to Market Using Economic Modeling." *Proceedings of the IEEE ITC International Test Conference*, 2001: 1098 – 1107.
- VOLKERINK, E. H., A. KHOICHE, J. RIVOIR ve K. D. HILLIGES. "Test Economics for Multi-Site Test with Modern Cost Reduction Techniques." *Proceedings of the 20th IEEE VLSI Test Symposium (VTS'02)*, 2002a.
- VOLKERINK, E. H., A. KHOICHE, J. RIVOIR ve K. D. HILLIGES. "Modern Test Techniques: Tradeoffs, Synergies, and Scalable Benefits." *Journal of Electronics Testing: Theory and Applications*, 19, 2003: 125 – 135.
- VOLKERINK, E. H., A. KHOICHE ve S. MITRA. "Packet – Based Input Test Data Compression Techniques." *Proceedings of the IEEE ITC International Test Conference*, 2002b: 154 – 163.
- WANG, C.-H. "Economic Off-Line Quality Control Strategy with Two Types of Inspection Errors." *European Journal of Operational Research*, 179, 2007: 132 – 147.
- WEST, B. G. "Simultaneous Bidirectional Test Data Flow for a Low – Cost Wafer Test Strategy." *Proceedings of the IEEE ITC International Test Conference*, 2003: 947 – 951.

- WILLIAMS, D., ve A. P. AMBLER. "System Manufacturing Test Cost Model." *Proceedings of the IEEE ITC International Test Conference*, 2002: 482 – 490.
- WILLIAMS, T. W. "VLSI Testing." *IEEE Computer*, October 1984, 126 – 136.
- WILLIAMS, T. W. ve N. C. BROWN. "Defect Level as a Function of Test Coverage." *IEEE Transactions on Computers*, C – 30, 12, December 1981, 987 – 988.
- WILLIAMS, T. W., ve K. P. PARKER. "Design for Testability – A Survey." *Proceedings of the IEEE*, 71, 1, January 1983, 98 – 112.
- WILSON, S., ve B. FLOOD, S. GOYAL, J. MOSHER: BERGIN, J. O'BRIEN ve R. KENNEDY. "Parameter Estimation for a Model With Both Imperfect Test and Repair." *Proceedings of 25th IEEE VLSI Symposium (VTS'07)*, 2007.
- WILSON, S., ve S. GOYAL. "Bayesian Estimation for an Imperfect Test and Repair Model." *12th IEEE European Test Symposium*, May 2007.
- WILSON, S., T. JOYCE ve E. J. LISAY. "Reliability Prediction from Censored Field Return Data." *European Safety and Reliability Conference*, September 2006.
- YEH, R. H. ve T. – H. CHEN. "Optimal Lot Size and Inspection Policy for Products Sold with Warranty." *European Journal of Operational Research*, 174, 2006: 766 – 776.
- YOKOYAMA, R., Y. HASEGAWA ve K. ITO. "A MILP Decomposition Approach to Large Scale Optimisation in Structural Design of Energy Supply Systems." *Energy Conversion and Management*, 43, 2002: 771 – 790.

- YU, X., A. FIN, F. FUMMI ve E. M. RUDNICK. "A Genetic Testing Framework for Digital Integrated Circuits." *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02)*, 2002.
- ZANG, G. ve H. ITO. "Concurrent Core Testing for SOC Using Merged Test Set and Scan Tree." *IEICE Transactions Information & Systems*, 89-D, 3, March 2006: 1157 – 1164.
- ZOU, W., S. M. REDDY, I. POMERANZ ve Y. HUANG. "SOC Test Scheduling Using Simulated Annealing." *Proceedings of 21th IEEE VLSI Symposium (VTS'03)*, 2003.
- ŽUŽEK, A., A. BIASIZZO ve F. NOVAK. "Sequential Diagnosis Tool." *Microprocessors and Microsystems*, 24, 2000: 191 – 197.

EK 1

Tüm Senaryolar

Senaryo	Özellik 1	Özellik 2	Özellik 3	SP _i	Karar
1	FF - 0.04	FF - 0.07	FF - 0.024	0,0000672	FF
2	FF - 0.04	FF - 0.07	TF - 0.340	0,0009520	TF
3	FF - 0.04	FF - 0.07	FP - 0.060	0,0001680	TF
4	FF - 0.04	FF - 0.07	TP - 0.576	0,0016128	FF
5	FF - 0.04	TF - 0.024	FF - 0.024	0,0000230	TF
6	FF - 0.04	TF - 0.024	TF - 0.340	0,0003264	TF
7	FF - 0.04	TF - 0.024	FP - 0.060	0,0000576	TF
8	FF - 0.04	TF - 0.024	TP - 0.576	0,0005530	TF
9	FF - 0.04	TP - 0.630	FF - 0.024	0,0006048	FF
10	FF - 0.04	TP - 0.630	TF - 0.340	0,0085680	TF
11	FF - 0.04	TP - 0.630	FP - 0.060	0,0015120	TF
12	FF - 0.04	TP - 0.630	TP - 0.576	0,0145152	FF
13	FF - 0.04	FP - 0.060	FF - 0.024	0,0000576	TF
14	FF - 0.04	FP - 0.060	TF - 0.340	0,0008160	TF
15	FF - 0.04	FP - 0.060	FP - 0.060	0,0001440	TF
16	FF - 0.04	FP - 0.060	TP - 0.576	0,0013824	TF
17	TF - 0.18	FF - 0.07	FF - 0.024	0,0003024	TF
18	TF - 0.18	FF - 0.07	TF - 0.340	0,0042840	TF
19	TF - 0.18	FF - 0.07	FP - 0.060	0,0007560	TF
20	TF - 0.18	FF - 0.07	TP - 0.576	0,0072576	TF
21	TF - 0.18	TF - 0.024	FF - 0.024	0,0001037	TF
22	TF - 0.18	TF - 0.024	TF - 0.340	0,0014688	TF
23	TF - 0.18	TF - 0.024	FP - 0.060	0,0002592	TF
24	TF - 0.18	TF - 0.024	TP - 0.576	0,0024883	TF
25	TF - 0.18	TP - 0.630	FF - 0.024	0,0027216	TF

26	TF - 0.18	TP - 0.630	TF - 0.340	0,0385560	TF
27	TF - 0.18	TP - 0.630	FP - 0.060	0,0068040	TF
28	TF - 0.18	TP - 0.630	TP - 0.576	0,0605318	TF
29	TF - 0.18	FP - 0.060	FF - 0.024	0,0002592	TF
30	TF - 0.18	FP - 0.060	TF - 0.340	0,0036720	TF
31	TF - 0.18	FP - 0.060	FP - 0.060	0,0006480	TF
32	TF - 0.18	FP - 0.060	TP - 0.576	0,0062208	TF
33	TP - 0.76	FF - 0.07	FF - 0.024	0,0012768	FF
34	TP - 0.76	FF - 0.07	TF - 0.340	0,0180880	TF
35	TP - 0.76	FF - 0.07	FP - 0.060	0,0031920	TF
36	TP - 0.76	FF - 0.07	TP - 0.576	0,0306432	FF
37	TP - 0.76	TF - 0.024	FF - 0.024	0,0004378	TF
38	TP - 0.76	TF - 0.024	TF - 0.340	0,0062016	TF
39	TP - 0.76	TF - 0.024	FP - 0.060	0,0010944	TF
40	TP - 0.76	TF - 0.024	TP - 0.576	0,0105062	TF
41	TP - 0.76	TP - 0.630	FF - 0.024	0,0114912	FF
42	TP - 0.76	TP - 0.630	TF - 0.340	0,1627920	TF
43	TP - 0.76	TP - 0.630	FP - 0.060	0,0287280	FP
44	TP - 0.76	TP - 0.630	TP - 0.576	0,2757888	TP
45	TP - 0.76	FP - 0.060	FF - 0.024	0,0010944	TF
46	TP - 0.76	FP - 0.060	TF - 0.340	0,0155040	TF
47	TP - 0.76	FP - 0.060	FP - 0.060	0,0027360	FP
48	TP - 0.76	FP - 0.060	TP - 0.576	0,0262656	FP
49	FP - 0.02	FF - 0.07	FF - 0.024	0,0000336	TF
50	FP - 0.02	FF - 0.07	TF - 0.340	0,0004760	TF
51	FP - 0.02	FF - 0.07	FP - 0.060	0,0000840	TF
52	FP - 0.02	FF - 0.07	TP - 0.576	0,0008064	TF
53	FP - 0.02	TF - 0.024	FF - 0.024	0,0000115	TF
54	FP - 0.02	TF - 0.024	TF - 0.340	0,0001632	TF

55	FP - 0.02	TF - 0.024	FP - 0.060	0,0000288	TF
56	FP - 0.02	TF - 0.024	TP - 0.576	0,0002765	TF
57	FP - 0.02	TP - 0.630	FF - 0.024	0,0003024	TF
58	FP - 0.02	TP - 0.630	TF - 0.340	0,0042840	TF
59	FP - 0.02	TP - 0.630	FP - 0.060	0,0007560	FP
60	FP - 0.02	TP - 0.630	TP - 0.576	0,0072576	FP
61	FP - 0.02	FP - 0.060	FF - 0.024	0,0000288	TF
62	FP - 0.02	FP - 0.060	TF - 0.340	0,0004080	TF
63	FP - 0.02	FP - 0.060	FP - 0.060	0,0000720	FP
64	FP - 0.02	FP - 0.060	TP - 0.576	0,0006912	FP

EK 2

Tüm Senaryolar İçin Girdi ve Çıktı Durum Vektörü Hesapları

sce	att	Makine 1		Makine 2		Makine 3	
		Girdi	Çıktı	Girdi	Çıktı	Girdi	Çıktı
(1-0)	G1	0,900000	0,987807	0,987800	0,987800	0,987800	0,994600
	G2	0,982800	0,995257	0,987800	0,997140	0,987800	0,997981
(0-1)	G1	0,982800	0,995776	0,982800	0,982800	0,900000	0,993974
	G2	0,993500	0,996739	0,993500	0,998073	0,982800	0,998257
(2-0)	G1	0,900000	0,987807	0,987800	0,987800	0,987800	0,994600
	G2	0,982800	0,995257	0,987800	0,997140	0,987800	0,997981
(1-1)	G1	0,941400	0,991767	0,985300	0,985300	0,943900	0,988656
	G2	0,988150	0,995974	0,990650	0,997606	0,985300	0,997724
(0-2)	G1	0,982800	0,995776	0,982800	0,982800	0,900000	0,993974
	G2	0,993500	0,996739	0,993500	0,998073	0,982800	0,998257
(3-0)	G1	0,900000	0,987805	0,987800	0,987800	0,987800	0,994600
	G2	0,982800	0,995257	0,987800	0,997140	0,987800	0,997981
(2-1)	G1	0,927600	0,990441	0,986133	0,986133	0,929267	0,988656
	G2	0,986367	0,995730	0,989700	0,997451	0,984467	0,997642
(1-2)	G1	0,955200	0,993098	0,984467	0,984467	0,958533	0,990632
	G2	0,989933	0,996224	0,991600	0,997762	0,986133	0,997808
(0-3)	G1	0,982800	0,995776	0,982800	0,982800	0,900000	0,993974
	G2	0,993500	0,996739	0,993500	0,998073	0,982800	0,998257
(4-0)	G1	0,900000	0,987805	0,987800	0,987800	0,987800	0,994600
	G2	0,982800	0,995257	0,987800	0,997140	0,987800	0,997981
(3-1)	G1	0,920700	0,989780	0,986550	0,986550	0,921950	0,985702
	G2	0,985475	0,995610	0,989225	0,997373	0,984050	0,997601
(2-2)	G1	0,941400	0,991767	0,985300	0,985300	0,943900	0,988656
	G2	0,988150	0,995974	0,990650	0,997606	0,985300	0,997724
(1-3)	G1	0,962100	0,993765	0,984050	0,984050	0,965850	0,991622
	G2	0,990825	0,996350	0,992075	0,997840	0,986550	0,997851
(0-4)	G1	0,982800	0,995776	0,982800	0,982800	0,900000	0,993974
	G2	0,993500	0,996739	0,993500	0,998073	0,982800	0,998257
(5-0)	G1	0,900000	0,987805	0,987800	0,987800	0,987800	0,994600
	G2	0,982800	0,995257	0,987800	0,997140	0,987800	0,997981
(4-1)	G1	0,916560	0,989384	0,986800	0,986800	0,917560	0,985112
	G2	0,984940	0,995539	0,988940	0,997326	0,983800	0,995777
(3-2)	G1	0,933120	0,990971	0,985800	0,985800	0,935120	0,987473
	G2	0,987080	0,995827	0,990080	0,997513	0,984800	0,997675
(2-3)	G1	0,949680	0,992565	0,984800	0,984800	0,952680	0,989841
	G2	0,989220	0,996123	0,991220	0,997700	0,985800	0,997774
(1-4)	G1	0,966240	0,994166	0,983800	0,983800	0,970240	0,992216
	G2	0,991360	0,996427	0,992360	0,997877	0,986800	0,997876
(0-5)	G1	0,982800	0,995776	0,982800	0,982800	0,900000	0,993974
	G2	0,993500	0,996739	0,993500	0,998073	0,982800	0,998257

ÖZGEÇMİŞ

Kişisel bilgiler

Adı Soyadı : Onur KOYUNCU
Doğum Yeri ve Tarihi : Ankara, 31.05.1976

Eğitim Durumu

Lisans Öğrenimi : Orta Doğu Teknik Üniversitesi, İstatistik, 1998
Yüksek Lisans Öğrenimi : Hacettepe Üniversitesi, İşletme, 2002
Bildiği Yabancı Diller : İngilizce (Çok iyi), Almanca (Orta)
Bilimsel Faaliyetleri : İstatistiksel Analiz, Fabrika Yerleşim Planlaması, Test Tasarımı ve Optimizasyonu, Envanter Optimizasyonu, Yöneylem Araştırması, Kısıt Programlama

İş Deneyimi

Stajlar : Bulunmamaktadır.
Projeler : Reorganizasyon "Hacettepe Üniversitesi", Ankara, 2002.
Reorganizasyon "Diyanet İşleri Başkanlığı", Ankara, 2003.
Proje: "Cork University Hastanesi Envanter Optimizasyonu", İrlanda, 2004 – 2005.
Proje: "TAC SCM, Foreseer", İrlanda, 2005.
Proje: "CTVR, Test Set Optimisation", İrlanda, 2004 – 2005.
Çalıştığı Kurumlar : Prokal AŞ, Ankara, Kalite Yöneticisi, 1998-2000.
H. Ü. İİBF İşletme Bölümü, Araştırma Görevlisi, 2000-halen.
UCC, Cork Constraint Computation Center, İrlanda, 2005.

İletişim

E-posta adresi : okoyuncu@hacettepe.edu.tr
onur.koyuncu@gmail.com

Tarih : 27 Kasım 2007