

**OLAYLARA DAYALI SENARYOLARIN
GENETİK ALGORİTMA İLE
OPTİMİZASYONU**

Bülent ŞİMŞEK

**Doktora Tezi
İşletme Anabilim Dalı
Prof. Dr. Üstün ÖZEN
2014**

Her hakkı saklıdır

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI

Bülent ŞİMŞEK

OLAYLARA DAYALI SENARYOLARIN GENETİK ALGORİTMA
İLE OPTİMİZASYONU

DOKTORA TEZİ

TEZ YÖNETİCİSİ
Prof. Dr. Üstün ÖZEN

ERZURUM-2014



T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLERİ ENSTİTÜSÜ



TEZ BEYAN FORMU

09.05.2014

SOSYAL BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

BİLDİRİM

Atatürk Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğine göre hazırlamış olduğum **OLAYLARA DAYALI SENARYOLARIN GENETİK ALGORİTMA İLE OPTİMİZASYONU** adlı tezin/raporun tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin/raporumun kağıt ve elektronik kopyalarının Atatürk Üniversitesi Sosyal Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım:

Lisansüstü Eğitim-Öğretim yönetmeliğinin ilgili maddeleri uyarınca gereğinin yapılmasını arz ederim.

- ☒ ☐ Tezimin/Raporumun tamamı her yerden erişime açılabilir.
- ☐ ☐ Tezim/Raporum sadece Atatürk 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.

09.05.2014

Bülent ŞİMŞEK



T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



TEZ KABUL TUTANAĞI

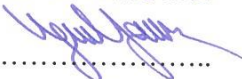
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Prof. Dr. Üstün ÖZEN danışmanlığında, **Bülent ŞİMŞEK** tarafından hazırlanan bu çalışma 09 / 05 / 2014 tarihinde aşağıdaki jüri tarafından. İşletme Anabilim Dalı'nda Doktora Tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Üstün ÖZEN

İmza: 

Jüri Üyesi : Prof. Dr. Uğur YAVUZ

İmza: 

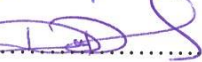
Jüri Üyesi : Doç. Dr. Ahmet BARAN

İmza: 

Jüri Üyesi : Yrd. Doç. Dr. Abdulkadir ÖZDEMİR

İmza: 

Jüri Üyesi : Yrd. Doç. Dr. Deniz DAL

İmza: 

Yukarıdaki imzalar adı geçen öğretim üyelerine aittir. / /

Prof. Dr. Mustafa YILDIRIM
Enstitü Müdürü

İÇİNDEKİLER

ÖZET.....	VI
ABSTRACT	VII
ŞEKİLLER DİZİNİ	VIII
ÇİZELGELER DİZİNİ	X
ÖNSÖZ.....	XII
GİRİŞ	1

BİRİNCİ BÖLÜM LİTERATÜR TARAMASI

1.1. GENETİK ALGORİTMA.....	3
1.2. OLAY YAPILARI İLE İLGİLİ ÇALIŞMALAR	4
1.3. MODELLEME	5

İKİNCİ BÖLÜM SENARYO MODELİ

2.1. GENEL MODEL YAPISI	8
2.2. NESNE YAPISI.....	9
2.2.1. Varlık Öğeleri	10
2.2.1.1. Varlık	10
2.2.1.2. Özellik	11
2.2.1.3. AnaVarlık	11
2.2.1.4. Durum.....	11
2.2.1.5. Nesnesel Varlık.....	13
2.2.1.6. Nesne	15
2.2.1.7. Cisim	15
2.2.1.8. Küme	15
2.2.1.9. Olay	16
2.2.1.10. Bulanık ve Bulanık Grup	20
2.3. MODELLEME TANIMI.....	20
2.3.1. Model	21

2.3.2. Çözüm Motoru.....	21
2.3.3. Kromozom (Senaryo)	25
2.3.3.1. Olay Genleri	27
2.3.4. Populasyon	29
2.3.5. Üreteç.....	36
2.3.6. Çözücü	40
2.4. VARSAYILAN DEĞERLER.....	46

ÜÇÜNCÜ BÖLÜM

SENARYONUN SİMÜLASYONU

3.1. SENARYONUN GENEL İŞLEYİŞİ	49
3.2. OLAYLARIN ÇALIŞMA PRENSİBİ	49
3.2.1. Olay Öncesi Kontroller	49
3.2.2. Olay Şartı	50
3.2.3. Eylem Fonksiyonu	50
3.2.4. Dönem Sonu İşlemleri	51
3.2.4.1. Atama.....	52
3.2.4.2. Etki	53
3.2.4.3. Direnç.....	54
3.2.4.4. Bağlı Durumlar	54
3.2.4.5. Normale Dönüş	55
3.2.4.6. Tahditler	56
3.2.4.7. Tetiklemeler.....	57
3.2.4.8. Son İşlemler	57
3.2.4.9. Puanlama	58

DÖRDÜNCÜ BÖLÜM

GENETİK ALGORİTMANIN UYGULANIŞI

4.1. İLK KROMOZOMLARIN OLUŞTURULMASI.....	59
4.2. ÇAPRAZLAMA.....	59
4.2.1. Puanlama ile Çaprazlama.....	60
4.2.2. Analiz ile Çaprazlama	61

4.3. MUTASYON	63
4.3.1. Çaprazlama İçinde Uygulanan Mutasyon	63
4.3.2. Atıl Genlerin Temizlenmesi	63
4.3.3. Zaman Kaydırması.....	63
4.3.4. Gen İşlemleri.....	64
4.3.5. Akıllı Mutasyon	64
4.3.6. Mutasyon Nesli	65
4.4. KROMOZOM DEĞİŞİMİ.....	65
4.4.1. Üstün Nesil	65
4.4.2. Zayıf Kromozomların Değiştirilmesi.....	65

BEŞİNCİ BÖLÜM ÇÖZÜMLERİN ANALİZİ

5.1. NESİLERDE KROMOZOM PUANLARININ DEĞİŞİMİ	66
5.2. ÇAPRAZLAMA YÖNTEMLERİNİN SONUÇ ÜZERİNDEKİ ETKİSİ	68
5.3. PUAN DEĞERLERİ.....	70
5.4. MUTASYON VE ZAYIF KROMOZOMLARIN DEĞİŞİMİ.....	70

ALTINCI BÖLÜM ÖRNEK SENARYO 1

6.1. MEVCUT DURUM.....	72
6.1.1. Firmalar	72
6.1.2. Ürün Özellikleri.....	73
6.1.3. Müşteri Bilgileri.....	75
6.1.4. Yetkili Bayiler	77
6.1.5. Reklamlar.....	77
6.2. MODEL.....	78
6.2.1. Nesne Yapıları.....	79
6.2.1.1. Firmalar	79
6.2.1.2. Ürünler.....	81
6.2.1.3. Müşteriler	82
6.2.2. Olaylar	84

6.2.2.1. Standart Olay	84
6.2.2.2. Reklamlar	84
6.2.2.3. Ürün Geliştirme	85
6.2.2.4. Yeni Ürün Projelendir.....	85
6.2.2.5. Üretime Başla	85
6.2.2.6. Kampanya	86
6.2.2.7. Prestij	86
6.2.2.8. Promosyon	86
6.2.3. Diğer Firma Hareket Tarzları (Tetiklemeler)	86
6.2.4. Tanınma	87
6.2.4.1. Müşteri Etkileşimi	88
6.2.5. Performans Artışı	92
6.2.6. Üretim.....	92
6.2.7. Tasfiye.....	92
6.2.8. Aylık Müşteri Sayısı	93
6.2.9. Alım Kriterleri.....	93
6.2.10. Maliyet ve Fiyat Güncellemeleri.....	96
6.3. ÇÖZÜM SONUÇLARI	96
6.3.1. Lider Firma Çözümü (Elit)	96
6.3.2. Orta Seviye Firma Çözümü (Anka).....	103
6.3.3. Alt Seviye Firma Çözümü (Çip).....	108
6.3.4. Ürün Hayat Dönemi	111
6.3.4.1. Giriş Dönemi	112
6.3.4.2. Gelişme Dönemi	112
6.3.4.3. Olgunluk Dönemi.....	112
6.3.4.4. Gerileme Dönemi	112
6.3.5. Firmaların Beklenen Davranışları ile Sonuçların Karşılaştırılması	113
6.4. SONUÇ	114

YEDİNCİ BÖLÜM

ÖRNEK SENARYO 2

7.1. FİLTRELER.....	117
7.1.1. Mean Filter (Ortalama Filtresi)	118
7.1.2. Median Filter (Medyan Filtresi)	118
7.1.3. Shrunked Mean (Daraltılmış Ortalama Filtresi) ve Shrunked Median (Daraltılmış Medyan Filtresi).....	118
7.1.4. Shrunked Mean Wide (Daraltılmış Ortalama Filtresi, Geniş) ve Shrunked Median Wide (Daraltılmış Medyan Filtresi, Geniş)	118
7.2. TEST 1	119
7.3. TEST 2	121
7.4. TEST 3	123
7.5. TEST 4	124
 SONUÇ ve ÖNERİLER.....	 127
KAYNAKÇA	129
ÖZGEÇMİŞ.....	132

ÖZET**DOKTORA TEZİ****OLAYLARA DAYALI SENARYOLARIN GENETİK ALGORİTMA İLE
OPTİMİZASYONU****Bülent ŞİMŞEK****Tez Danışmanı: Prof. Dr. Üstün ÖZEN****2014, 132 sayfa****Jüri: Prof. Dr. Üstün ÖZEN****Prof. Dr. Uğur YAVUZ****Doç. Dr. Ahmet BARAN****Yrd. Doç. Dr. Abdülkadir ÖZDEMİR****Yrd. Doç. Dr. Deniz DAL**

Bu tez çalışmasında, stratejik seviyede sosyal bir optimizasyon modeli amaçlanmıştır. Modellenen bir sistemi, verilen olaylar kullanılarak, mevcut durumdan istenen duruma getirebilmek için geçerli ve optimize edilmiş çözüm yolları bulan bir algoritma geliştirilmiştir. Başka bir deyişle, tezin amacı mevcut durum uzayını modelleyerek ve verilen olaylar ile üretilen senaryoları simülasyon ile çözerek uygulayabileceğimiz hareket tarzları içinden bizi en yüksek ihtimalle en iyi sonuca götürecek yolu bulabilmektir.

Algoritmada, yöntemin gereklerine uygun ve daha verimli sonuçlar üretmek amacıyla puanlama, analiz, tam analiz olmak üzere üç adet çaprazlama yöntemi geliştirilmiştir. Modelin yapısına göre, bu üç yöntemden biri seçilebilmektedir.

Çalışmada iki örnek uygulama yapılmıştır; İlki, masaüstü bilgisayar üretimi ve pazarlaması yapan bir firmanın, kârını maksimize etmek üzere yeni ürün projelendirme, ürün geliştirme, reklam, promosyon gibi hareket tarzlarını içeren bir model geliştirilmiş ve firmanın üç yıllık stratejisini belirlemek için geliştirilen optimizasyon modeli kullanılarak senaryolar oluşturulmuştur. Elde edilen sonuçlar, ürün hayat dönemi üzerine olan beklentiler ve kendi içindeki sebep sonuç ilişkileri ile karşılaştırılarak geçerliliği ve doğruluğu incelenmiştir.

İkinci örnek uygulama olarak geliştirilen model yapısı ile önceden tanımlanmış fonksiyonlar kümesinden yapılacak seçimlerle oluşturulacak işlemler dizisinin bir veri kümesine istenen türde değişimleri yapabilecek veya ilk hali bilinen veri kümesinin muhteviyatı bilinmeyen böyle bir işlemler dizisinden geçtikten sonra işlemler dizisinin yapısını bulabilecek bir model denenmiştir. Bu maksatla resim, çizim, manzara gibi farklı özelliklerdeki resimler bazı filtrelerden geçirilerek geliştirilen algoritmadan yapılan filtrelerin yapılış sıraları ile birlikte bulunması istenmiş ve tam sonuç elde edilmiştir.

Anahtar Kelimeler: Optimizasyon, Senaryo, Genetik Algoritma, Olay

ABSTRACT**Ph. D. DISSERTATION****SCENARIOS OPTIMIZATION BASED ON EVENTS WITH GENETIC
ALGHORITMS****Bülent ŞİMŞEK****Advisor: Prof. Dr. Üstün ÖZEN****2014, 132 pages****Jury: Prof. Dr. Üstün ÖZEN****Prof. Dr. Uğur YAVUZ****Assoc. Prof. Dr. Ahmet BARAN****Assist. Prof. Dr. Abdülkadir ÖZDEMİR****Assist. Prof. Dr. Deniz DAL**

In this study, a social optimization model at the strategic level is intended. An algorithm, which finds valid and optimal solutions using given events to improve a modelled system from its current state to the desired state, is developed. In other words, the purpose of the thesis is to model the current system and given events to find the best scenario, which is solved via simulations, with the highest probability to occur.

Three new crossover methods are developed to produce more efficient results according to the requirements of the algorithm; scoring, analysis, and full analysis. In order to improve the performance, considering the structure of the model, one of the three methods can be selected.

Two sample applications were conducted in this thesis. In the first sample application, a company, which manufactures and markets desktop computers, requires producing a three-year strategy using events including developing new products, advertisements, promotions, and others. Using the model developed in this thesis, many scenarios were produced and sorted as solutions. These scenarios were analyzed and verified over the expectations of the product's life-cycle and the cause-effect relations.

The model developed as the second sample application, had a function pool to be built a sequence to modify a data set from the beginning state to a target state or to find out the sequence of functions which had already made this kind of modifications. For this reason, the second model was tested with some multi filtered images with different patterns. The model was asked for and fully succeeded to find the filter sequence where the original and filtered images were given.

Keywords: Optimization, Scenario, Genetic Algorithm, Event

ŞEKİLLER DİZİNİ

Şekil 2.1. Modelin Ana Unsurları	8
Şekil 2.2. Varlık Sınıf Hiyerarşisi	10
Şekil 2.3. Olay Yapısı	16
Şekil 2.4. Modelleme Yapısı	21
Şekil 2.5. Nesil Döngüsü	23
Şekil 2.6. Çözüm Motoru Akış Diyagramı	24
Şekil 3.1. Olay Atama İşlemleri	52
Şekil 3.2. Tahdit Kontrolü	57
Şekil 4.1. Puanlama Yöntemi	60
Şekil 4.2. Analiz Puanlama	61
Şekil 4.3. Tam Analiz Puanlama	62
Şekil 5.1. Firma Modelindeki Kromozom Puanlarının Nesillere Göre Azami ve Asgari Değerleri	66
Şekil 5.2. Diğer Modellere Ait Kromozom Puanlarının Nesillere Göre Azami ve Asgari Değerleri	67
Şekil 5.3. Çözümler Arası İlişkiler	68
Şekil 5.4. Çaprazlama Yöntemine Göre Oluşan Farklar, Firma Modeli	69
Şekil 5.5. Çözümler Arası İlişkiler	71
Şekil 6.1. %10 Tanınan Firma	88
Şekil 6.2. %50 Tanınan Firma	89
Şekil 6.3. %90 Tanınan Firma	89
Şekil 6.4. Müşterilerin Birbiri İle Etkileşimi Sonucu Firmayı Tanıma Değerlerinde Meydana Gelen Değişim Miktarı	91
Şekil 6.5. Elit Bütçe	97
Şekil 6.6. Elit Pazar Payı	98
Şekil 6.7. Elit Aktif Ürün Sayısı	98
Şekil 6.8. Elit Aylık Üretim ve Satış Miktarları	99
Şekil 6.9. Ürün 24, Üretim, Stok ve Satış Miktarları	99
Şekil 6.10. Pazar Paylarının Karşılaştırılması	100
Şekil 6.11. Anka Bütçe	103
Şekil 6.12. Anka Kâr	104

Şekil 6.13. Pazar Payları	104
Şekil 6.14. Pazar Paylarının Karşılaştırılması	105
Şekil 6.15. Çip Bütçe	108
Şekil 6.16. Çip Pazar Payı	109
Şekil 6.17. Ürün Hayat Dönemi.....	111
Şekil 7.1. Test Maksadıyla Kullanılan Resimler	117
Şekil 7.2. 8-Komşu Pikseller	118
Şekil 7.3. Filtrelenen Resimler.....	120
Şekil 7.4. Test 4 Nesil-Puan Grafiği	126

ÇİZELGELER DİZİNİ

Çizelge 2.1. Çözüm Kriteri	46
Çizelge 2.2. Model Kriteri	47
Çizelge 2.3. Varsayılan Değerler	47
Çizelge 2.4. Mutasyon	48
Çizelge 2.5. Kromozom Değişimi	48
Çizelge 5.1. Çaprazlama Yöntemine Göre Oluşan Farklar, Firma Modeli	69
Çizelge 5.2. Çözümler Arası İlişkiler	71
Çizelge 6.1. Firma Bilgileri	73
Çizelge 6.2. Geliştirme Kategorileri	74
Çizelge 6.3. Müşteri Tercihleri ve İhtiyaçları	76
Çizelge 6.4. Reklam Etkileri	78
Çizelge 6.5. Firma Durum Değişkenleri	80
Çizelge 6.6. Ürün Durum Değişkenleri	81
Çizelge 6.7. Ürünün Alım Kriteri Hesabında Kullanılan Ek Yapı	82
Çizelge 6.8. Müşteri Durum Değişkenleri	83
Çizelge 6.9. Firmaların Tanınma Seviyelerine Göre %10 Etkileşim Sonucu Firmanın Artan Tanınma Değerleri	91
Çizelge 6.10. Aylık Müşteri Sayısı	93
Çizelge 6.11. Ürün Tercihinde Puan Katsayıları	94
Çizelge 6.12. Ürün Tercihinde Müşteri Oranlarının Hesaplanması	95
Çizelge 6.13. Başlangıçta Oluşacak Ürün Tercih Oranları	95
Çizelge 6.14. Elit Senaryo	101
Çizelge 6.15. Anka Senaryo	106
Çizelge 6.16. Çip Senaryo	109
Çizelge 7.1. Test Filtre Sırası	119
Çizelge 7.2. Test 1 Parametreler	121
Çizelge 7.3. Test 2 Filtre Sırası	121
Çizelge 7.4. Test 2 Parametreler	122
Çizelge 7.5. Test 2 Sonucu	123
Çizelge 7.6. Test 3 Filtre Sırası	123
Çizelge 7.7. Test 3 Parametreleri	124

Çizelge 7.8. Test 3 En İyi Çözümler.....	124
Çizelge 7.9. Test 4 Filtre Dizisi	125
Çizelge 7.10. Test 4 Parametreleri	125

ÖNSÖZ

Öğrencilik yıllarımda en zevk alarak öğrendiğim konulardan biri optimizasyon yöntemleri olmuştur. O zamanlar aklıma gelen bir soru şöyleydi; Acaba bir bilgisayar programı optimize edilebilir mi? Sonrasında ise fonksiyonlar yerine olayları ve bilgisayarlar yerine insanları yerleştirdiğimizde, “Acaba insanların davranışlarına yön vermek mümkün müdür?” sorusu aklıma kurcaladı. Bilgisayar programlarının optimizasyonu ile kasıt, yapılan gereksiz kısımların atılması ve belli ifadelerin değiştirilmesi gibi optimizasyonlar değil, yöntemin optimize edilmesidir. Bu her konu için ayrı model gerektirir, ihtiyaca göre geliştirilebilir. Doktora tezim için bunun denemeye değer olduğunu düşündüm ve danışman hocamın da onayı ile çalışmalara başladım ve tezimde o yıllarda zihnimi meşgul eden soruları çözebilecek bir algoritma geliştirdim.

Bu çalışmada farklı alanlarda yaptığım örnek uygulamalar arasından ikisini tez içerisinde inceledim. Gerçek dünyadan bir örnek çalışma olması açısından bir alan uzmanının da yardımıyla firma modelini geliştirdik. Bu uygulamada firmanın, reklam, yeni ürün geliştirme, promosyon gibi olayları kullanarak üç yıllık bir süreç için stratejisini belirlemek üzere bir model hazırladık ve sonuçlarını Firmaların gerçekte uygulayacakları davranışlarla karşılaştırdık.

Diğer uygulamada hem yöntemin optimizasyonu, hem de sosyal ve psikolojik optimizasyonu temsilen şöyle bir örnek modelledim: İşin temeline baktığımızda bizim ihtiyacımız, bir veri kümesini değişik işlemlerden geçirerek öncekine benzeyen fakat ihtiyacımıza göre şekillendirilmiş farklı bir veri kümesi elde etmektir. Bu veri kümesini temsilen bir resim kullanmak istedim ve o resim üzerinde değişik filtreler uygulayarak şablon olarak gösterdiğim diğer resmi elde etmeye çalıştım. Amaçlanan sonuca birebir ulaşılabilir olması, psikolojik ve sosyal optimizasyonların bu algoritma ile yapılabileceğini görmemi sağladı.

Çalışmama bilgi ve tecrübeleri ile değerli katkılarda bulunan danışman hocam sayın Prof.Dr. Üstün Özen’e ve desteklerini benden esirgemeyen değerli aileme ve firma modeli konusunda bize yardımlarını esirgemeyen Prof. Dr. Sevtap Ünal’a teşekkürlerimi arz ederim.

GİRİŞ

Günlük yaşamımızda beynimizi çok değişik problemlerle yorarız. Evde, işyerinde, arkadaş çevresinde farklı problemlerle uğraşırız. Bütçemizle ilgili hesap yapmak, şirketin yatırımlarıyla ilgili plan yapmak, tatile giderken memlekete en kısa yoldan gitmek; düşündüğümüz problemlerden bazılarıdır. Bu tür sorunların çözümüyle ilgili birçok sayısal yöntem mevcuttur. Peki ya sayılarla ifade edemeyeceğimiz problemlerimiz? Eğer bilinen bir sistemimiz varsa ve bunu modelleyebiliyorsak bunun nasıl çalışabileceğini görmek veya bu sistemin gelecekteki durumu ne olabileceğini anlamak için bir simülasyon koşturulabilir. Ancak benzetim ile çözülemeyecek ya da çözümü çok zor olan problemler de vardır. Örneğin;

a. Siz bir öğretmensiniz ve sınıfınızın motivasyonu düşük. Ancak öğrencilerinizin neyi motive ettiğini, küstürdüğünü ya da neyin, neyi nasıl etkilediğini, birbirleriyle nasıl etkileşim içinde olduklarını biliyorsunuz. Fakat bu, tüm öğrenciler için farklıdır. Meselâ sınıf içindeki not sıralaması, Ahmet için çok önemliyken, Mehmet için hiçbir önemi yoktur. Bu sınıfı en iyi şekilde nasıl motive edebilirsiniz?

b. Siz bir şirketin pazarlama müdürüsünüz. Ürünlerinizden biri ortada görünür bir sebep yokken, satışı 2 ay içinde %70 düşmüştür. Acaba bu duruma hangi olaylar zinciri sebep olmuş olabilir?

İşte bu tür problemler için neden-sonuç ilişkisine ve nesneler arasındaki ilişkilere ve bu ilişkilerin ihtimallerine dayalı bir model oluşturmak gereklidir. Bu model, “A olayı olursa bunun sonucu ne olur”, “B olayı/durumunun meydana gelebilmesi için hangi şartlar sağlanmalı veya eylemler gerçekleştirilmelidir” gibi sorulara cevap bulabilmelidir.

Bu modelin kurulabilmesi ve çözüm sunabilmesi durumunda sayısal olmayan, psikolojik, toplumsal, soyut veya sosyal problemlerin de bilgisayar destekli olarak çözülebilmesi veya optimize edilmesini mümkün kılan genel bir algoritma olacaktır.

Bu tez çalışmasında, modellenen bir sistemin başlangıç durumundan istenen duruma getirilmesi için gerekli olaylar zincirinin tasarımı ve bu zincirin optimizasyonunu sağlayan bir algoritma geliştirilmiştir. Hesaplamalarda olayların nesneler üzerindeki etkileri kullanılacaktır. Ortamın, nesnelerin ve olayların çeşitli

özellikleri, ihtimalleri ve şartları olabilecektir. Hedef olarak stratejik seviyede ve sosyal problemler düşünülmüştür. Sayısal modellerin çözümü amaçlanmadığından küçük ve yakın değerler ihmal edilmektedir.

Teze başlamadan önce amacım probleme göre şekillenen dinamik bir yapay sinir ağı tasarlamaktı. Algoritmanın uygulanabilir ve kullanılabilir olması için yapay sinir ağları senaryolara ve düğümler olaylara dönüştü. Başta bir senaryolar havuzu oluşturulmakta ve genetik algoritma ile her nesilde yeni senaryolar hazırlanmaktadır. Böylece hedeflenen yapının bir uygulaması gerçekleştirilmiş olmaktadır.

Tasarlanan algoritma, öncelikle verilen parçaları (olay, durum, şart, ihtimal, vb.) birleştirerek senaryolar oluşturmaktadır. Sonra bu senaryolar simülasyonla çözülmekte ve hangi olasılıkla hedefe ne kadar yaklaştığı hesaplanmaktadır. Eğer senaryodaki olaylardan herhangi birinde olasılıkla ilgili bir işlem varsa yeteri kadar tekrar ile senaryonun beklenen sonucu bulunmakta ve genetik algoritma kullanılarak en iyi senaryo bulunmaya çalışılmaktadır.

İlk bölümde bir literatür taraması yapılmıştır. İkinci bölümde önce tüm modelin unsurları hakkında genel bilgi verilmiştir. Bu kısım tüm modeli özetleyerek anlatılacak bölümlerin daha kolay anlaşılması amacıyla hazırlanmıştır. Üçüncü bölümde simülasyonun nasıl koşturulduğu, dördüncü bölümde ise genetik algoritmanın senaryo modeline uygulanışı anlatılmaktadır.

Beşinci bölümde ise yapılan uygulamaların sonuçları incelenmiş, çaprazlama yöntemleri ile özel nesillerin ve mutasyon uygulamalarının sonuç üzerindeki etkileri tartışılmıştır.

Altı ve yedinci bölümlerde yapılan iki örnek uygulama, firma modeli ve filtre modeli, incelenmektedir. Bu bölümlerde her iki modelin tanımı, modellenmesi, sonuçları ve sonuçlarının incelenmesi yapılmıştır.

Geliştirilen sistemin en az intel dual core işlemci, 2 GB Ram Bilgisayar ile Windows 7 işletim sistemi üzerinde çalışacağı varsayılmıştır. Karmaşık modeller için 8 çekirdekli Xeon işlemci kullanılması tavsiye edilir. Geliştirme ortamı olarak C# kullanılmıştır.

BİRİNCİ BÖLÜM

LİTERATÜR TARAMASI

1.1. GENETİK ALGORİTMA

“An Overview of Genetic Algorithms Part 1 & 2” isimli çalışmada Beasley vd. genetik algoritma ve uygulama alanlarını incelemişlerdir. Makalede temel prensiplerin yanında başka tekniklerle de karşılaştırmalar yapılmıştır. (Beasley, 1993)

G. Deng, simülasyon tabanlı optimizasyon uygulamalarında optimizasyon tekniklerini simülasyon analizine entegre etmiştir. Makalesinde iki aşamalı bir yapı önermektedir; (Deng, 2007)

Birince aşamada tüm etki alanı üzerinde küresel bir keşif yapılmıştır. Bu alanda sınıflandırma araçları veya DIRECT (Etki alanının dikdörtgenlere bölünmesi) yöntemleri kullanılabilmektedir.

İkinci aşamada ise; yerel güven bölgesi serbest-türevlerinin bir koleksiyonudur. Bu yöntem istenen doğrulukta çözümler elde etmek için algoritmik verimliliği artırırken uygun örnekleme stratejileri için yol göstererek Bayes teknikleri uygulamaktadır.

“Evolutionary Computation: from Genetic Algorithms to Genetic Programming” isimli çalışmada genetik algoritma ile ilgili temel tekniklerden bahsedilmiş, devamında evrim stratejilerini anlatılmıştır (Abraham vd, 2006).

“Global Optimization Algorithms, Theory and Application” isimli kitapta genetik algoritma ve programlama, makine öğrenimi, tepe tırmanma, benzetilmiş tavlama konularını anlatılmıştır (Weise, 2009: 25).

“Genetik Algoritmalar ve Uygulama Alanları” isimli makalede Genetik algoritma ve uygulama alanları incelenmiştir. Çalışmada ilk olarak genetik algoritma kavramı ve temel teoremi hakkında bilgi verilmiştir. Daha sonra, basit genetik algoritmanın çalışma adımları ve parametre seçimi incelenmiş ve fonksiyon optimizasyonu için bir çözüm örneği verilmiştir. Genetik algoritmanın uygulama alanları, genel ve işletme alanları olmak üzere iki temel gruba ayrılarak incelenmiştir (Emel vd, 2002).

E. Cantu-Paz, incelemesinde paralel genetik algoritma üzerine yapılan en önemli yayınları birleştirilmiş olarak organize etmeyi amaçlamıştır. Çalışmada genetik algoritmayı paralel çalışmayı sağlayan teknikler sınıflandırılmış ve örneklerle gösterilmiştir. (Cantu-Paz, 1998)

“The Parallel Genetic Algorithm As Function Optimizer” isimli çalışmada genetik algoritmanın sürekli fonksiyonlar üzerine optimize edilmesinden bahsedilmiştir. Alt popülasyonlar, yerel iyi çözüm bulmayı hedeflemişlerdir. Genetik algoritmanın belli sayıda tekrardan sonra hâlâ istenen sonuç bulunamamışsa tepe tırmanma yöntemi kullanılmaktadır. Alt popülasyonda bulunan iyi yerel çözüm komşu alt popülasyonlara dağıtılmaktadır.

“Multi-Objective Optimization Using Genetic Algorithms” isimli çalışmada yazarlar, çok amaçlı optimizasyon problemlerini çözmek üzere tasarlanan genetik algoritmaları özetlemişlerdir. Bu algoritmalar, çözüm çeşitliliğini artıran yöntemlerle özel uygunluk fonksiyonları kullanılması açısından geleneksel genetik algoritmalarından ayrılmaktadır. çok amaçlı formülleştirmeler, bir çok karmaşık mühendislik optimizasyon problemleri için gerçekçi bir model olmaktadır. Özelleştirilmiş genetik algoritma bu sorunlara mükemmel çözüm belirlemek için özellikle etkili olduğu ortaya konmuştur (Coit vd, 2005).

1.2. OLAY YAPILARI İLE İLGİLİ ÇALIŞMALAR

“Stochastic Simulation of Event Structures” adlı makalede stokastik işlem cebirini tanımlanmakta ve bir değerlendirme yöntemi üzerine çalışılmaktadır. Stokastik işlem cebirinin markov zincirlerine benzemesinin avantajı olarak analitik ve sayısal performans değerlendirmelerine ait standart tekniklerin kullanılmasına imkân verse de, kullanılan dağılımlar üssel dağılım ile kısıtlanmıştır (Katoen vd., 1996).

Makalede önerilen kısmi düzen ile üssel olmayan dağılımlarında kullanılmasına olanak verilmiştir. Ancak bu durumda markov zincirleri ile olan benzerlik kaybolduğundan, kısmi düzenli modelin analizi için kesikli olay simülasyon tekniklerinin kullanılması önerilmektedir.

“Optimal control of multiple CSPA system based on event-based Q learning” isimli çalışmada yazarlar çoklu taşıyıcılı üretim istasyonu (CSP) sistemi için optimal

kontrol problemi üzerinde çalışmıştır. Bu optimizasyon probleminde amaç, her istasyon için koordineli ileriye dönük kontrol stratejisi seçilerek tüm sistemin kısmi işleme oranının maksimize edilmesidir. Olay tabanlı optimizasyon ve performans potansiyellerinin konseptinin kullanılması fikrine göre, koordineli İleriye dönük kontrol probleminin çözümü için azaltılmış veya ortalama performans kriterleri ile bir olay tabanlı Q-öğrenme algoritması önerilmektedir. Çalışmada bir simülasyon örneği ile önerilen algoritmanın etkinliği gösterilmiştir. Bu simülasyonda elde edilen ve tüm sistemin kısmi işleme oranını gösteren sonuçların, Wolf-PHC algoritmasıyla karşılaştırıldığında, önemli ölçüde yükseldiği görülmüştür (Hao vd., 2009).

Allison'un yaptığı çalışmada doğal olarak oluşan olaylar veya planlanan müdahalelerin etkilerini tahmin etmek için çoklu dalga panel verisi (multi-wave panel data) kullanarak bazı metotlar geliştirmiştir. Bunu, kutuplaştırılmış zaman serileri ve yatay kesit verinin ekonometrik tedavisi ile kesikli zaman serilerinde olayları sentezleyerek gerçekleştirmektedir (Allison, 1994).

Dixon ve Thompson'un yaptıkları çalışmalarında büyük ölçekli planlama ve zamanlama problemleri ile ilgili geçmiş davranışsal araştırma bulgularını dâhil etmişlerdir. Çalışmada ünlü sahne sanatları mekânlarında karşılaşılan sezon aboneliklerinin tekrar alımına yönelik zirve, uç ve yayılma etkilerini birleştiren olaylar ile ilgili stratejiler önerilmektedir. Psikolojik sekans etkilerini maksimize etmeyi hedefleyen olay zamanlama ve yerleştirme probleminin gerçekçi bir matematiksel gösterimi sunulmuştur. Problem, meta-sezgisel ve bir dizi varsayımsal senaryolar kullanılarak çözülmüş, yönetsel bakış açısı ve olayların ardışık etkileri ile olay planlama çabaları nasıl en iyi şekilde birleştirilebileceği üzerine yön sağlamak için test edilmiştir (Dixon vd., 2011).

1.3. MODELLEME

Jameson, çalışmasında sayısal tekniklerin kullanıcı ve öğrenci modellemelerindeki belirsizlik yönetimi tekniklerini üç ana kuramsal çerçevede ele almaktadır; Bayes ağları, Dempster-Shafer teorisi ve bulanık mantık (Jameson, 1996).

Öncelikle nispeten basit kullanıcı problemlerine nasıl uygulanabileceğini göstererek temel kavramları tanıtmaktadır. Sonrasında kuramsal çerçeveden

uyguladıkları tekniklere göre her sistemi karakterize ederek incelemektedir. Son olarak kullanıcı ve öğrenci modellemeleri için bu tekniklerin çeşitli kullanılabilirlik yönlerini ele almaktadır.

Zeigler ve diğerleri kitaplarında; sürekli ve kesikli paradigmaların modelleme ve simülasyon için entegrasyonu üzerine bahsetmişlerdir. İkinci önemli konu ise dağıtık simülasyon ve onun çoklu model bileşenlerindeki çok biçimliliğin birlikte çalışabilirliğine verdiği destektir (Zeigler vd., 2000).

Kitap modelleme ve simülasyon için titiz bir matematiksel temel oluşturmaktadır. Bu uygulamada kullanılan çeşitli simülasyon yaklaşımları, hücrel otomasyon, kaotik sistemler, hiyerarşik blok diyagramlar ve Petri Nets gibi popüler modelleme yöntemleri dahil olmak üzere, entegrasyon için kapsamlı bir çerçeve sağlamaktadırlar. Birleştirici konsept olarak DEVS (Discrete Event System Specification) veri yolu açıklanmış ve nesne tabanlı DEVS modellerinin paralel ve dağıtık ortamlarda nasıl işlem verimli olarak oluşturulabileceğini anlatmıştır.

Geiss çalışmasında matematik ve stokastik modellemenin temelinden bahsettikten sonra kesikli zaman Markov zincirlerini ve Markov zinciri Monte Carlo (MCMC) yöntemleri hakkında geniş bilgi vermiştir. MCMC ile ilgili Gibbs örnekleyicisi ve Metropolis algoritmasının anlatımı örnek uygulamalarla gösterilmiştir (Geiss, 2009).

Hubscher-Younger vd makalelerinde kaynak tahsisi için kesikli olay ve zaman tabanlı modellerin optimizasyon ile entegre edilmesi üzerine çalışmışlardır. Makaleye göre teknik sistemlerinde performans optimizasyonu çok fazla büyütülmemelidir. Hatta küçük iyileştirmeler önemli bir maliyet, kaynak ve zaman tasarrufu neden olabilir. Dinamik sistem optimizasyonuna yapıcı bir yaklaşım, optimizasyon problemini matematiksel anlamda formüle edebilir. Modern sistemlerin karmaşıklığı, özellikle farklı modelleme paradigmaları etkileştiğinde, bu tür biçimselleşmeyi çoğu zaman engellemektedir. Bu tür parazit etkiler ek karmaşıklık getirmektedir. Bu çalışma, çözüm uzayında sayısal optimizasyon yaklaşımının problem uzayının sayısal simülasyonu ile birleştirilmiş olan üretken bir yaklaşım kullanmaktadır. Çoklu paradigma doğasını çözmek için simülasyon, soyut bir icra çerçevesi formunda somutlaştırılabilen bir birleştirici anlamsal etkiye dayanır. Sayısal altyapısının esnekliği nedeniyle, bir çok yapılandırılabilir entegre bir ortam optimizasyon uzmanı için kullanılabilir olmaktadır.

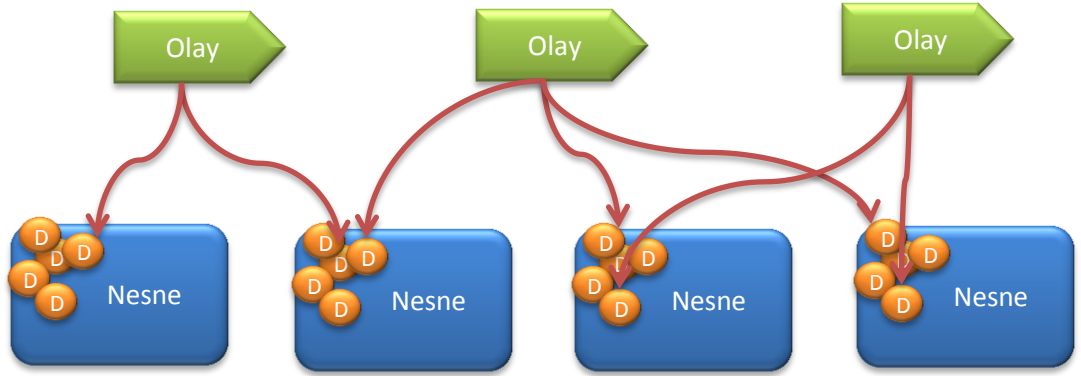
Genel yaklaşım, sürekli zaman, ayrık-olay ve durum geçiş sistemlerini birleştiren bir kaynak tahsisi sorunu ile gösterilmiştir (Hubscher-Younger vd., 2012).

İKİNCİ BÖLÜM

SENARYO MODELİ

2.1. GENEL MODEL YAPISI

Modelde temel olarak, varlıkları temsil eden *Nesneler*, nesnelerin özelliklerini ve bazı değerlerini gösteren *Durumlar* ve nesnelerin durumlarını değiştiren *Olaylar* tanımlanmıştır (Bkz Şekil 0.1. Modelin Ana Unsurları).



Şekil 0.1. Modelin Ana Unsurları

Her bir nesne bir *cisimden* türetilir. Bu şekilde nesneler cisimlerde tanımlanmış durumları da miras alırlar. Aynı zamanda olayların hangi türde nesneleri kendine parametre olarak alacağını ayırt edebiliriz.

Nesnelerin bir araya gelmesiyle *kümeler* oluşturulur. Kümeler, aynı türdeki nesnelerin gruplandırılmasını sağlar. Bir olay için parametre olarak bir küme verildiğinde, bu küme içinden seçilen nesnelere işlem yapılır. Küme yapısında nesneler tek tek tanımlanıp eklenebileceği gibi, belli durumlara sahip nesnelerin oranı veya sayısı da tanımlanabilir. Bu tip tanımlamalar toplumsal problemlerin çözümünde kullanılabilir.

Durumlar ise; ait olduğu varlığın değer veya özelliklerini belirten değişkenlerdir. Sayısal, metin, bulanık değer, dağılım veya özellik gibi değerler alabilirler. Özellik değeri, metinsel bir ifade olup belli bir özelliğin veya yeteneğin varlığını ifade eder.

Durumların kendi içinde direnç, normal değer, etkiler gibi özellikleri vardır. Bunlar durumlar yapısı içinde ayrıca ele alınacaktır.

Olaylar, modelde işlemlerin tanımlandığı yapıdır. Olay tanımı içinde çeşitli özelliklerin tahditleri, olayın gerçekleşme şartı, olayın etkileri, süresi gibi unsurlar yer almaktadır.

Olayların içeriğinde herhangi bir sınırlama olmaması için olayın gerçekleşme şartı, olay esnasında gerçekleştirilen işlemler gibi unsurlar C# kodu olarak eklenmelidir. Bu nedenle de oluşturulan her yeni model için yazılan kodun tekrar derlenmesi gerekmektedir. Asıl kod bir dll (dynamic link library) içinde yer almakta, kullanıcının “*model.cs*” dosyasının içeriğini doldurması gerekmektedir. Bu yapı aynı zamanda simülasyonun çok daha hızlı çalışmasını sağlamaktadır.

Simülasyon, üretilen senaryolar üzerinde koşturulmaktadır. Her bir senaryo bir zaman doğrusundan ve zaman üzerine yayılan olay genlerinden oluşmuştur. Olay genleri; olay ile birlikte olayı kimin gerçekleştirdiği, kime yaptığı ve bundan kimlerin etkilendiğini ile olayın gerçekleşme oranı ve ihtimali gibi bilgileri içermektedir.

Senaryolar modelde *kromozom* adı ile tutulmaktadır. Kromozomlar da *Popülasyon* sınıfının bir ögesi olan nesiller dizisi içinde saklanmaktadır. Çözüm aşamasında tüm işlemler *Çözüm Motoru* tarafından yürütülmektedir. Kromozomlar *Üreteç* tarafından ilk kez üretilmekte, çaprazlanmakta ve mutasyona uğratılmaktadır. Kromozomların faydalı değerini bulan simülasyonlar *Çözücü* tarafından koşturulmaktadır. Çözücüler çok kanallı olarak çalışabilmekte, her bir kanalda ayrı bir çözücü kullanılmaktadır.

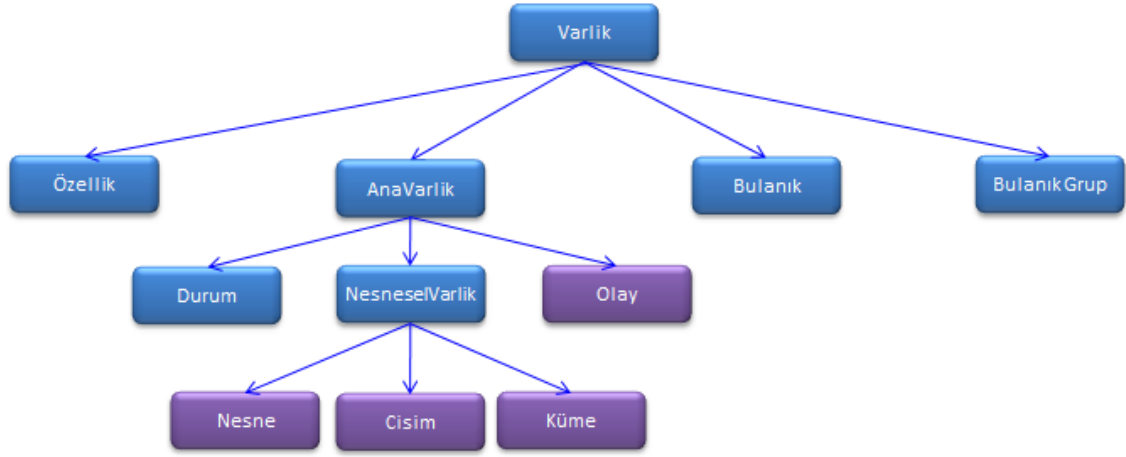
İstenilen başarı oranına veya sınırlanan nesil sayısına ulaşıldığında yeni nesil üretimi durdurularak tüm nesillerdeki en iyi çözümler sıralanmaktadır.

2.2. NESNE YAPISI

Modelde varlık ögeleri ve çözücü ögeleri olmak üzere iki farklı ana sınıf (class) grubu bulunmaktadır. Varlık ögeleri cisim, nesne, olay gibi sınıfları içerirken, çözücü ögeleri çözüm motoru, çözücü, sistem, popülasyon, kromozom gibi sınıfları içermektedir.

2.2.1. Varlık Öğeleri

Varlık öğelerinin tümü *Varlık* sınıfından türetilmiştir. Her çocuk nesne ebeveyninin özelliklerini miras almaktadır. *Varlık*, *AnaVarlık* ve *NesneselVarlık* sınıfları ara sınıflardır ve başka sınıfları oluşturmak için tasarlanmışlardır. *Olay*, *Nesne*, *Cisim* ve *Küme* kendi başlarına kullanılabilen sınıflardır. Diğer sınıflar ise başka sınıfların içinde yer alan veya kullanılan sınıflardır. Sınıf hiyerarşisi aşağıdaki şekilde gösterilmiştir.



Şekil 0.1. Varlık Sınıf Hiyerarşisi

2.2.1.1. Varlık

Tüm varlık öğeleri bu sınıftan türetilmiştir. Bu sınıfta tüm varlıklarda olması gereken bilgi ve özellikler tanımlanmıştır. Bunlar; varlığın ne olduğu, varlığın adı ve varlık numarasıdır.

Varlık oluşturulduğu anda sistem içerisinde bir diziye eklenir. Böylece bir varlık arandığında bu dizi kullanılarak ulaşılabilir. Tüm varlıkların bir kopyası her çözücü için ayrıca oluşturulur. Böylece aynı anda birden çok çözücü çalışırken varlıkların değerleri birbiriyle karışmaz. Varlıktan türetilen tüm sınıflarda ait oldukları çözücü ile bir bağ kurulur ve özgün olarak numaralandırılır.

2.2.1.2. Özellik

Bir varlıkta belli bir özellik tanımlanabilir. Olayın eylem safhasında da tanımlanan özelliğin varlığına göre farklı işlemler yapılabilir. Örneğin ortamın sıcaklığı ile ilgili yapılan bir modelde Soba, Klima ve Kalorifer için “Isıtır” özellikleri, Klima ve Vantilatör için “Soğutur” özellikleri eklenebilir.

Özelliklerin tezatları tanımlanabilir. Örneğin net ve puslu iki tezat özellik olarak tanımlandığında bir görüntü için net özelliği atandığında o nesnede puslu özelliği varsa silinir.

İki varlık arasındaki bağlantılara da özellik atanabilmektedir ki bu tür özellikler ilişki olarak adlandırılacaktır. İlişki tipindeki özelliklerde tezat değerler karşı ilişki için geçerli olur.

Eğer özelliğin “Sabit” değeri “doğru” (true) olarak atanırsa, bu özellik simülasyon esnasında değiştirilemez anlamındadır.

2.2.1.3. AnaVarlık

Varlıklara özelliklerin eklenmesini sağlar. Nesne, cisim, küme, olay ve durumlar bu sınıftan türetilmişlerdir. AnaVarlık’ta tüm özelliklerin saklandığı bir kütüphane bulunmaktadır ve özellikler ile ilgili fonksiyonlar burada tanımlanmıştır.

2.2.1.4. Durum

Durumlar, optimizasyon modelimizdeki ana değişkenlerdir. Ancak normal değişkenlerden farklı olarak, durumlar farklı veri tiplerine sahiptirler ve belirli davranış biçimleri sergilerler. Durumlar, sayılarla ifade edilseler de aslında soyut veya somut bir varlığın iki uç kavram arasındaki yerini temsil etmektedir. Durumlar bu yönüyle bulanık mantığa benzeseler de aşağıda anlatılacak farklı özelliklere sahiptirler.

Bir durum, bir volkanın harekete geçme durumunu temsil edebileceği gibi, bir insanın duygularını da temsil edebilmektedir. Durumlar birbirleriyle ilişkilendirilebilmekte, böylece bir durum değiştiğinde bir başka durum bundan etkilenebilmektedir. Yine durumlar için değişime bir direnç tanımlanabilmekte veya durumun belli bir değere veya başka bir duruma doğru meyletmesi sağlanabilmektedir.

Durumların genel özellikleri aşağıdaki gibi sıralanabilir;

- Her durum bir nesneye aittir.
- Durumlar; Sayısal (Tamsayı veya ondalıklı), Metin, Bulanık, Dağılım, Özellik, Tezat Değer tiplerinde olabilirler.
- Durumların değerleri sınırlandırılabilir.
 - Alt ve üst sınır olarak
 - Başka durumlarla ($A > B$ gibi)
 - Başka durumların toplamlarıyla ($A+B < C+D$ veya $A+B = 5$ gibi)
- Her durum için bir ilk değer atanır.
- Eğer durum amaç fonksiyonunda yer alacaksa, bu durum için bir “hedef değeri” ve “ağırlık değeri” atanır. Hedef değer, bu durumun olması istenen değeridir. Ağırlık değeri o durumun puanın amaç fonksiyonundaki ağırlığını temsil eder.
- Bazı durumlar bir nesnenin başka bir nesneyle olan bir ilişkisinin değerini gösterebilir. Bu durumda, o durum tüm nesneler arasında tanımlanır ve *BağılDurum* olarak adlandırılır. Bağıl durumlar genellikle cisim üzerinde tanımlanarak diğer nesnelerde de tek seferde tanımlanması sağlanır. Örneğin firmaların birbirlerine olan bağımlılıklarını ele aldığımızda $\text{FirmaA.Bağımlılık.FirmaB}$ bir bağıl durumu ifade eder. Ters bağımlılıklar aynı değerde olmak zorunda değildir, ancak bağıl durumlar ile ilgili istenen tahdit tanımlanabilir.

$\text{FirmaA.Bağımlılık.FirmaB} = \text{FirmaB.Bağımlılık.FirmaA}$ veya

$\text{FirmaA.Bağımlılık.FirmaB} + \text{FirmaB.Bağımlılık.FirmaA} = 1$ gibi.

- Bir olay bir durumun değerini
 - Anında değiştirebilir,
 - Derhal veya bir süre sonra etkileyebilir,
 - Belli bir süre boyunca etkileyebilir. Etki süresi ve zamana bağlı etki fonksiyonu kullanıcı tarafından tanımlanabilir.

Etki için örnek vermek gerekirse bir A olayının belli bir insanın morali üzerinde devam eden bir etkisi olabilir.

- Birden fazla olay bir durumu etkiliyorsa bu etkinin nasıl olacağına karar verilebilir.

- Bir durum başka durumlara bağımlı olabilir. Kendi değeri değiştiğinde başka bir durumun değerini de istenen oranda etkileyebilir.

Örneğin; A kişinin morali yükseldikçe doğru karar verme ihtimali artabilir veya aynı şahsın B kişisine olan saygısı arttıkça onun hatalarını görme ihtimali azalabilir.

- Her durum için farklı ihmal değerleri atanabilir. İhmal değeri kadar olan uyumsuzluk ya da fark dikkate alınmaz. Örneğin; ihmal değeri = 0.05 ise 1 ile 1.05 eşittir veya $0.96 > 1$ doğrudur. Kesinlik gerektiren işlemlerde ihmal değeri 0 alınmalıdır.

- Bazen bazı durum değerleri sürekli başka bir değere doğru yaklaşmak istemektedirler.

- Eğer bir durum için normal değer tanımlanmışsa, o durum her zaman biriminde tanımlanan normal değere istenen oran veya değerle yaklaşır. Normal değeri olarak başka bir durum da tanımlanabilir.

Bu özelliğin kullanım alanı olarak şöyle bir örnek verilebilir; A şahsının sinirlilik durumu yüksek bir değere atansa da zamanla düşük olan normal değerine yaklaşacaktır.

- Durumlar için bir direnç tanımlıdır. Bu 0 ile 1 arasında bir değerdir. Değişime karşı olan direnci ifade eder.

Mesela; A şahsı B'ye göre daha çabuk sinirlenir. Bu halde B'nin sinirlilik durumunun direnci A'ninkinden daha yüksek olarak belirlenir.

2.2.1.5. Nesnesel Varlık

Nesne, Cisim ve Kümeleri oluşturmak üzere tasarlanmış bir ara sınıftır. Nesnesel varlıklar durum alabilirler ve olaylarla ilişkilenebilirler. Bu maksatla Nesnesel varlık içerisinde sahip olduğu Durum ve ilişkili olduğu olayları yönetebilmek ve bağ kurabilmek amacıyla her ikisine ait birer liste bulundurur.

Durum ve ilişkisel durumların yönetilmesi işlemleri bu sınıfta yürütülür. Bir nesneye bir durum eklendiğinde sonradan çıkarılması veya değiştirilmesi söz konusu değildir. Çünkü durum, nesnenin zati bir özelliğini tanımlamaktadır, varlığı sonradan değişmez, ama değeri elbette değişebilir.

Yeni bir durum eklendiğinde nesnesel varlık yapısı yeni durumu kendi içeriğindeki listeye ekler. Bundan sonra bu duruma bu liste aracılığıyla ulaşılabılır. Yeni durum eklenirken değeri veya veri tipi de belirtilebilir. Veri tipi belirtilmezse sayısal, ilk değeri belirtilmezse varsayılan değer atanır. Varsayılan değer, sonradan değiştirilmemişse, sayısal değerler için 0, diğer değerler için hiçliktir.

Yeni bir ilişkisel durum eklendiğinde ise biraz daha farklı bir işlem yapılır;

Yapı, öncelikle kendi varlığının (kaynak varlık) ve hedef varlığın hangi tür olduğunu sorgular. Yukarıda da belirttiğim gibi nesnesel varlık kendi başına kullanılamaz. Cisim, nesne veya küme olmak zorundadır. Yapı kendi varlığı ve hedef varlık için birer liste oluşturur. Bu listeye varlık tipine göre nesneler eklenir. Eğer ilgili varlık (kendi varlığı veya hedef varlık) bir nesne ise sadece o nesneyi ekler. Eğer ilgili varlık bir cisimse o cismin tüm üyeleri olan nesneleri o listeye ekler. Son olarak ilgili varlık bir kümeysse bu durumda o kümenin tüm üyelerini o listeye ekler.

Şimdi elimizde iki adet liste var; kaynak nesneler ve hedef listeler. Yapı, birbirinden farklı tüm kaynak listelerden hedef listelere birer durum tanımlar.

DurumEkle(İlişkiselDurumAdı + “.” + HedefNesneAdı);

Bu durumlar kaynak listedeki nesnelere eklenir. Bu ilişkiler daha sonra aşağıdaki şekilde ifade edilecektir;

ANesnesi.İlişkiselDurumAdı.BNesnesi;

C# kodunda ise bir indexer eklenerek aşağıdaki şekilde ifade edilmektedir;

ANesnesi[“İlişkiselDurumAdı.BNesnesi”];

İlişkisel durumların eklenmesinde bu yöntem kullanıcıyı bir çok angarya işlemlerden kurtarmaktadır. FirmaKümesi.IliskiselDurumEkle(“Satış”, MüşteriKümesi); komutu ile kümedeki tüm firma nesnelerine tüm müşterilere ait bir satış ilişkisi tanımlanmış olacaktır.

Gerektiğinde aynı ilişkiye ait tüm durumlara ulaşabilmek için ilişkisel durumlar ilişki adıyla gruplanmışlardır. Eğer belli bir ilişkisel duruma ait tüm durumlara ulaşmak gerekirse ANesnesi.IliskiselDurum(“GrupAdı”); komutu ile ilgili ilişkisel

durumlara ait bir liste oluşturulur. Bu listenin her üyesi ilgili ilişkisel duruma ait durumlardır.

2.2.1.6. Nesne

Modelde her türlü varlıkları temsil edebilen yapıdır. Nesne, nesnesel varlıktan türetildiğinden nesnesel varlığın tüm özelliklerine sahiptir. Her nesne kendisinin tipini belirten bir cisim sınıfına bağlıdır. Bu şekilde olayların ne tür nesneleri fail veya etkilenen olarak kullanılabileceği ayırt edilebilir. Nesneler tanımlandıklarında, türlerini belirten Cisim içinde oluşturulan tüm durum ve özellikleri kendine kopyalar. Bu şekilde aynı türdeki nesnelerde aynı durum ve özellikler bulunur. Ancak bunun yanında istenirse bir nesneye yeni bir durum veya özellik de eklenebilir. Oluşturulan her nesne, özelliklerini taşıdığı Cisim sınıfında bir listeye eklenir. Bu şekilde bir cismin özelliklerine sahip tüm nesneler kolaylıkla bulunabilir.

2.2.1.7. Cisim

Cisim sınıfı, aynı türdeki nesneleri daha kolay yönetmek için oluşturulmuştur. Cisimler, model içinde özne olamazlar. Ancak tüm nesneler, bir cisim temel alınarak oluşturulurlar. Cisim yapısı içinde kendisinden türetilen tüm nesnelere bir bağ vardır. Her oluşturulan nesne, ait olduğu cisme otomatik olarak eklenir.

Cisimler aynı zamanda kümenin özel bir durumu gibi davranabilirler; Bir olay için cisim parametre olarak verildiğinde, tüm elemanları kendinden türeyen birer nesne olan bir küme gibi, nesnelerinden birini parametre olarak verir.

Cisim yapısı içerisinde tanımlanan bir fonksiyon ilişkisel durumları sınırlandırır. Bu fonksiyon hangi cisim veya nesneye ait olduğuna bakmadan verilen isimdeki tüm ilişkisel durumların alt ve üst sınırlarını belirler.

2.2.1.8. Küme

Küme yapısı, bir model içinde bir topluluğu temsil etmek, aynı ya da farklı cisimlerden oluşturulan nesneleri gruplamak için oluşturulmuştur. Olaylarda özne

olarak kullanılabilirler. Bu durumda küme içinden bir nesne seçilerek olay genine gönderilir.

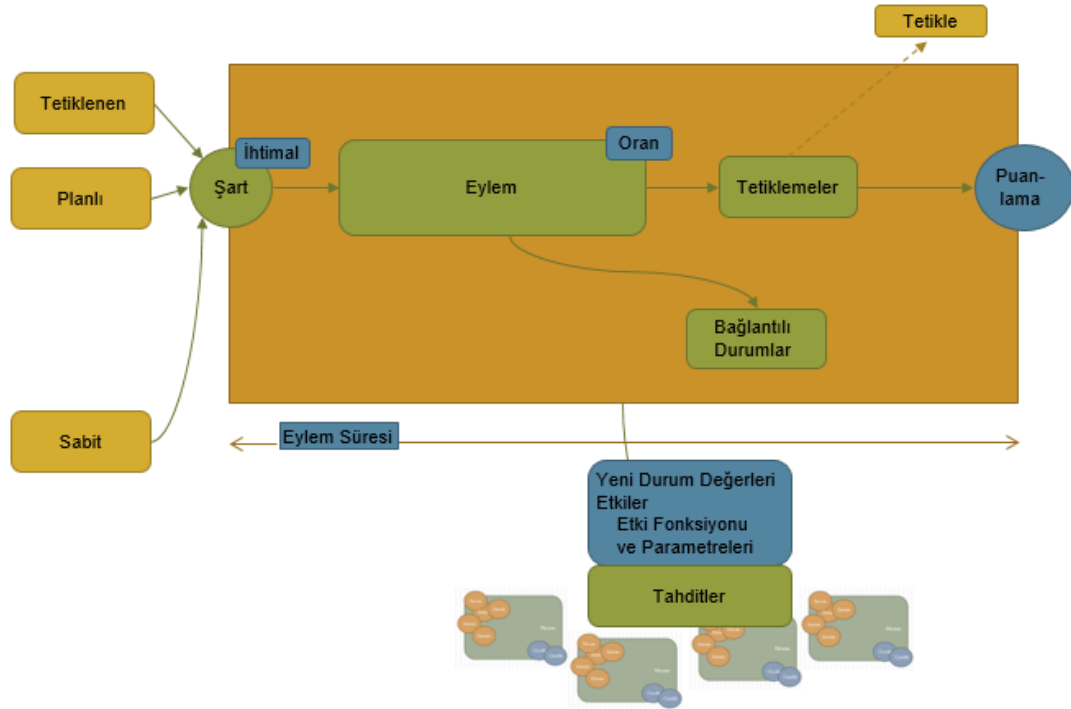
Kümeler, tek tek veya toplu olmak üzere iki farklı şekilde nesneleri temsil edebilirler. Kümede bu iki farklı şekilde temsil edilen nesneler aynı anda bulunabilir.

Tek tek temsil edilen nesneler olduğunda tüm nesneler oluşturularak kümeye eklenir. Kümede her nesneye bir bağ oluşturulur ve o nesne kümenin bir elemanı kabul edilir. Bir nesne birden fazla kümeye üye olabilir.

Kümede toplu olarak temsil edilen nesneler büyük toplulukların modellenmesi için düşünülmüştür. Her nesne tipi için birer nesne oluşturularak o nesneden ne kadar veya hangi oranda bulunduğu bilgisi bulundurulur.

2.2.1.9. Olay

Olaylar, nesnelere ait durumları etkileyen yapılardır. Olaylara ait *şart* ve *eylem* yapıları programlanabilir yapıdadır. Böylece olay içinde yapılacak işlemlerde hiçbir sınırlama bulunamamaktadır. Olay yapısı ile ilgili şema Şekil 0.1. Olay Yapısında gösterilmiştir.



Şekil 0.1. Olay Yapısı

Olaylar senaryo içinde planlanabilir veya sabit yapıdadır. Sabit olaylar her kromozomda belirlenen birim zamanda gerçekleştirilirler. Planlanabilir olaylar ise üretici tarafından oluşturulan senaryo içinde verilen tahditlere göre uygun sayı ve birim zaman aralıklarında yerleştirilir.

Olayların genel özellikleri aşağıdaki gibi sıralanabilir;

- Olayların 3 tür parametresi olabilir.

- **Yapan:** Olayı kimin yaptığı. Bir adet olabilir.

- **Etkilenen:** Olaydan etkilenen nesneler.

- **İlgili:** Olayı gerçekleştiren ya da etkilenen dışında kalan olayla ilgili diğer nesneler.

Örneğin: Ali, Hasan'ı Ahmet ve Veli hakkında uyarıyor.

Ali: Yapan, Hasan: Etkilenen, Ahmet ve Veli: İlgili.

- Olaylar için bir eylem tanımlanır. Eylemler program kodu olarak yazılırlar. Bu şekilde bir olayda hiçbir sınırlama olmadan her türlü işlem tanımlanabilir. Bu olayın hangi parametredeki nesnelerin durumlarını nasıl etkileyeceği belirtilir. Modelde gerçekleştirilen tüm işlemler burada yer alır.

- Şart, olayın gerçekleşmesi için hangi şartların oluşması gerektiği belirtilir. Şart yapısı da program kodu şeklindedir ve bu bize büyük esneklik sağlamaktadır. Şart fonksiyonu bir evet/hayır sonucu döndürür. Evet ise olaya ait eylemin gerçekleşmesi için gerekli şartların uygun olduğu anlaşılır.

Örneğin, A firması bir X projesine başlayacaksa maddi ve personel kaynakları bu proje için yeterli olmalıdır.

- Tetik, önceden belirlenen şartlar oluştuğunda olay kendiliğinden tetiklenir.

Örneğin, A hayvanı kendisine karşı bir tehdit varsa, tehdit ulaşabileceği bir noktada ise ve belli bir güç seviyesinin altındaysa A hayvanı için “Tehdide saldır” olayı tetiklenir. Parametreleri; yapan A hayvanı, etkilenen tehdit, ilgili parametresi ise yoktur.

- Süre; olayın süresi sonunda ilgili durumlar, olayın başındaki değerlerine göre etkilenirler. Olay süresi belli bir aralık olarak da tanımlanabilir. Bu halde olayın

gerçekleşen süresi için verilen aralıkta rastgele sayı üretilir. Olay sonucunda meydana gelen etkiler genellikle bu süre sonunda nesnelere yansıtılır. Ancak kurulan modelin ihtiyacına göre bu etkiler, ilk olayın gerçekleştiği zaman olmak üzere, farklı zamanlarda da nesnelere yansıtılabilirler.

- Etki ağırlığı; bir duruma aynı anda birden fazla olay etki ediyorsa etki ağırlığı hangi olayın etkisinin daha fazla veya az olacağını belirler. Atamalar ile ilgili hesaplamalar, Olayların çalışma prensiplerinde anlatılacaktır.

- Zorunlu özelliği doğru (evet) olan olay, mutlaka çözüm içinde yer alır. Bu tür olayların ne zaman oluşacağı modelde belirtilir ve zorunlu olay, üretilen veya çaprazlanan her senaryoda, belirtilen birim zamanda mutlaka yer alır. Zorunlu olaylar çaprazlamaya dâhil olmazlar, çözücü içerisinde ayrı bir listede tutularak her oluşturulan yeni kromozoma doğrudan eklenirler.

- Olayın planlanabilir özelliği doğru (evet) ise; bu olay çözücü tarafında bir çözüm oluşturulurken kullanılabilir. Tüm olaylar varsayılan olarak planlanabilir olarak kabul edilir. Üreteç bir senaryo üretirken planlanabilir olayları kullanır. Planlanabilir özelliği hayır olan olaylar senaryoya dâhil edilmez. Eğer belli bir zamanda sabit gerçekleşecek bir olay veya sadece tetiklenerek gerçekleşebilecek olaylar planlanamaz olarak işaretlenirler.

- İhtimal; eğer olayın kendisi veya etkileri belli bir olasılık dâhilinde oluşuyorsa, bu durumda bu olay için bir gerçekleşme ihtimali kullanılır. Bu özelliğe 0 ile 1 arasında değer atanır. Senaryo esnasında olay, atanan bu ihtimal oranında çalıştırılır. Çalıştırılmadığı takdirde ihtimalin gerçekleşmemesi nedeniyle çalıştırılmadığına dair bir bilgi kaydedilir.

- Olayların tekrar sayıları sınırlandırılabilir. Eğer bir olay en fazla belli bir sayıda olabiliyorsa, bu olaya ait planlanan ve tetiklenen tüm olay genleri azami sayıdan sonra dikkate alınmazlar ve çalıştırılmama sebepleri azami olay tahdidi olarak kaydedilir. Eğer olayın asgari belli bir sayıda tekrarlanması isteniyorsa üreteç bu olayı en az o miktarda planlar. Ancak yine de gerekli şartların veya gerçekleşme ihtimalinin oluşmaması gibi nedenlerle olay sayısı simülasyon içerisinde asgari sayının altında kalabilir.

- Bir olayın tekrarlanması için geçecek asgari veya azami süre belirlenebilir. Eğer aynı olayın tekrarlanabilmesi için geçmesi gereken bir süre varsa asgari süre olarak atanabilir. Yine bir olayın belli bir süre içinde mutlaka tekrarlanması gerekiyorsa azami süre aralığı ile bu değer belirlenebilir.

- Olayın gerçekleşebileceği en erken veya en geç birim zaman belirlenebilir. Bu durumda olay sadece belirlenen aralıkta gerçekleşir. Üreteç de yeni bir kromozom üretirken bu tahdidi dikkate alarak olayı sadece en erken ve en geç oluşma zamanları arasında planlar. Çaprazlama işlemlerinde ebeveyn kromozomda bulunan olay, çocuk senaryoya kopyalanırken oluşma zamanı değişmediğinden bu tahdit geçerliliğini korur.

- Olaylar genellikle olayın başladığı andaki durumların değerlerini dikkate alırlar ve olay bitiminde durumların değerlerini etkilerler. Ancak gerektiği takdirde durumların değerlerini o anda da değiştirebilirler. Tahditler ile ilgili kontroller o andaki tüm olaylar işlendikten sonra ve zaman ilerlemesi işleminden önce yapılır.

Olaylar, senaryo içerisine olay geni ile yerleştirilir. Olay geni içinde yapan, etkilenen ve ilgili nesne parametreleri üreteç tarafından belirlenerek atanırlar.

Olay içerisinde bulunan fonksiyonlar;

Execute: Simülasyon esnasında olay içinde gerçekleştirilmesi gereken işlemleri yapar. Ancak önce olayın azami tekrar sayısını, tekrar aralığını, eylem şartının gerçekleşme durumunu ve olayın olasılığını kontrol ederek olayın gerçekleşip gerçekleşmeyeceğine karar verir. Olayın gerçekleşmesi durumunda tekrar sayısı ve son tekrar zamanı bilgilerini günceller. Bu fonksiyon olayın gerçekleşmesi ile ilgili herhangi bir veri tutmaz ve bu nedenle hızlı çalışır.

ExecuteEx: Execute fonksiyonu gibi çalışır, ancak bu fonksiyon olayın gerçekleşmemesi durumunda hangi sebeple gerçekleşmediği bilgisini kaydeder.

SetParam: Modelde olayın fail, etkilenen ve ilgili nesne parametrelerinin belirlenmesini sağlar. Bu fonksiyon verilen parametrenin nesne, cisim veya küme olmasını ayırt eder ve gerekli atamayı yapar.

CreateParam: Verilen parametrenin tipine göre olay genine belirli bir nesnesel varlık gönderir. Olay geni için verilen parametre bir nesne ise aynıyle gene gönderir. Eğer bir cisim veya küme ise, öğelerinden belirli tek bir nesneyi gene gönderir.

CreateParamList: Üretcecın çözüm üretmesi esnasında bir gene uygun parametre atayabilmesi için bir olay geninin etkilenen veya ilgili nesnesel varlık parametreleri için uygun nesnelerin listelerini oluşturur. Böylece Üreteç bu listeden seçim yaparak parametre atar.

2.2.1.10. Bulanık ve Bulanık Grup

Eğer bir durumun değeri bulanık olarak tanımlanmışsa, Bulanık ve BulanıkGrup nesneleri kullanılır. Öncelikle bir kavram grubu temsil eden BulanıkGrup tanımlanmaktadır. Ardından ilgili kavramlar ve üyelik fonksiyonları tanımlanmaktadır.

Bulanık içinde aşağıdaki özellikler tanımlanmıştır;

- Üyelik fonksiyonu,
- Varsayılan değer,
- Varsayılan eşik,
- Denklik değeri; Sayısal bir değerın ilgili bulanık kavrama hangi ağırlıkta uyduğunu gösterir.

BulanıkGrup içinde aşağıdaki özellikler tanımlanmıştır;

- Eşik değeri,
- Kendine ait bulanık kavramların listesi,
- Değer fonksiyonu; Sayısal bir değerin en çok hangi kavrama ait olduğunu gösterir,
- Karşılaştırma fonksiyonu; Sayısal bir değerin verilen bulanık değere hangi ağırlıkta temsil ettiğini gösterir.

2.3. MODELLEME TANIMI

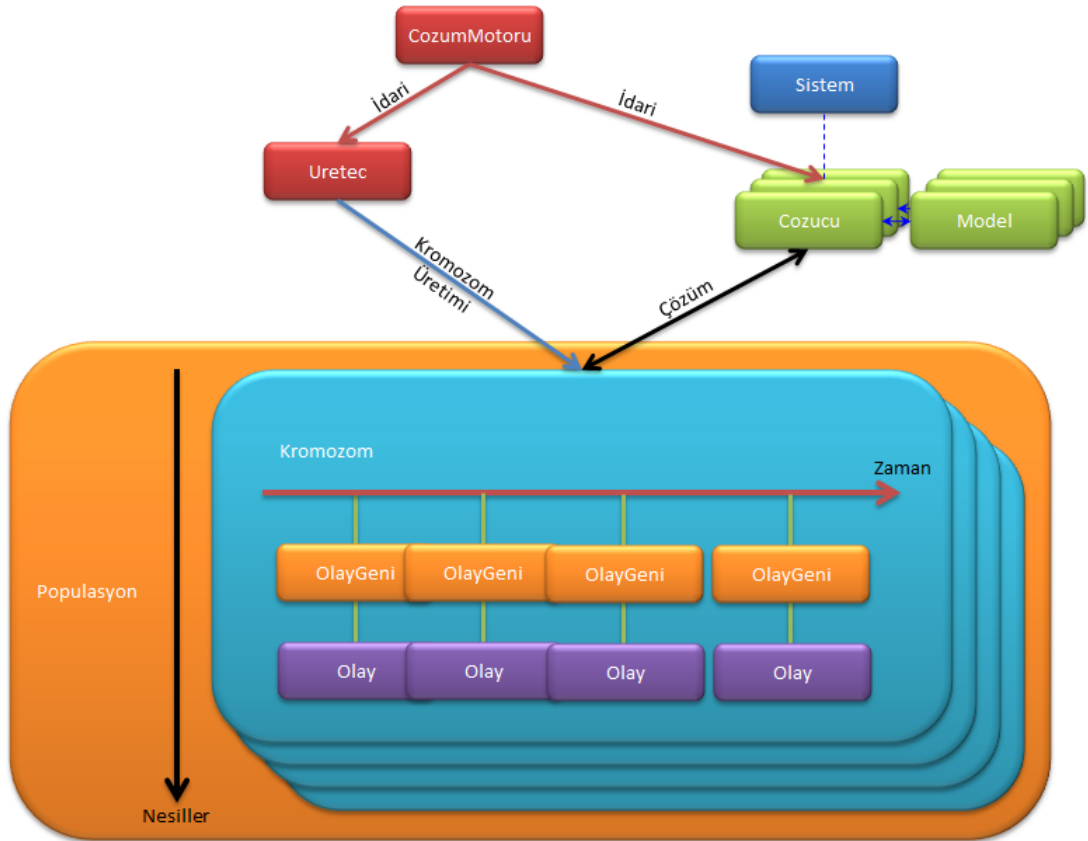
Problemin çözümü için bir dizi nesne tanımlanmıştır. Bu nesnelerden cisim, nesne, olay gibi unsurlar modeli oluşturmaktadır. Üreteç, bu modele göre senaryolar oluşturmakta, farklı çaprazlama ve mutasyon yöntemleriyle yeni nesiller üretmektedir. Çözücü ise üretilen bu senaryoları deneyerek puanlamaktadır. Modellemenin yapısı Şekil 2.4'te gösterilmiştir.

2.3.1. Model

Sistemin amacı, modellenmiş bir problem için senaryolar üretmek, simülasyon mantığı ile deneyerek değerlendirmesini yapmak ve genetik algoritma ile geliştirmek olduğundan; öncelikle problemin modellenmesi gerekmektedir.

Model, yukarda bahsedilen nesne, olay, durum vb. yapılarla kullanıcı tarafından tanımlanmaktadır. Tüm model “Model” isimli bir sınıfta tanımlanmaktadır. Her türlü cisim, nesne, olay ve ilgili durumlar burada tanımlanmalıdır. Daha sonra her “Çözücü” nesnesi tarafından bir model nesnesi oluşturularak senaryonun simülasyonu bu nesne üzerinde koşulacaktır.

Modelleme yapısı aşağıda gösterilmiştir.



Şekil 0.1. Modelleme Yapısı

2.3.2. Çözüm Motoru

Çözüm Motoru, modelde üreteç, popülasyon ve çözücü nesnelerini yöneten yapıdır. Başlangıçta Populasyon nesnesini hazırlar, istenen kanal sayısı kadar çözücü ve her çözücü için bir model nesnesi oluşturur ve çözümün bulunacağı asıl döngüyü başlatır.

Populasyon nesnesi içerisinde yeni nesiller oluşturulur. İlk nesilde tüm kromozomlar yeniden oluşturulur, müteakip nesillerde ise yeni nesil belirlenen çaprazlama yöntemlerinden biriyle çaprazlanarak oluşturulur. Kromozomların oluşturulması ve çaprazlanmaları konuları Üreteç nesnesinde ele alınacaktır.

Mutasyon uygulanıp uygulanmayacağı seçime bağlıdır. Uygulanacaksa bunun kaç nesilde bir, hangi olasılık ve oranda olacağı ve ne tür mutasyonların olabileceği belirlenebilmektedir. Uygulanabilen mutasyonlar; yeni gen eklenmesi, bir genin silinmesi, bir genin olayının, parametrelerinin veya zamanının değiştirilmesidir.

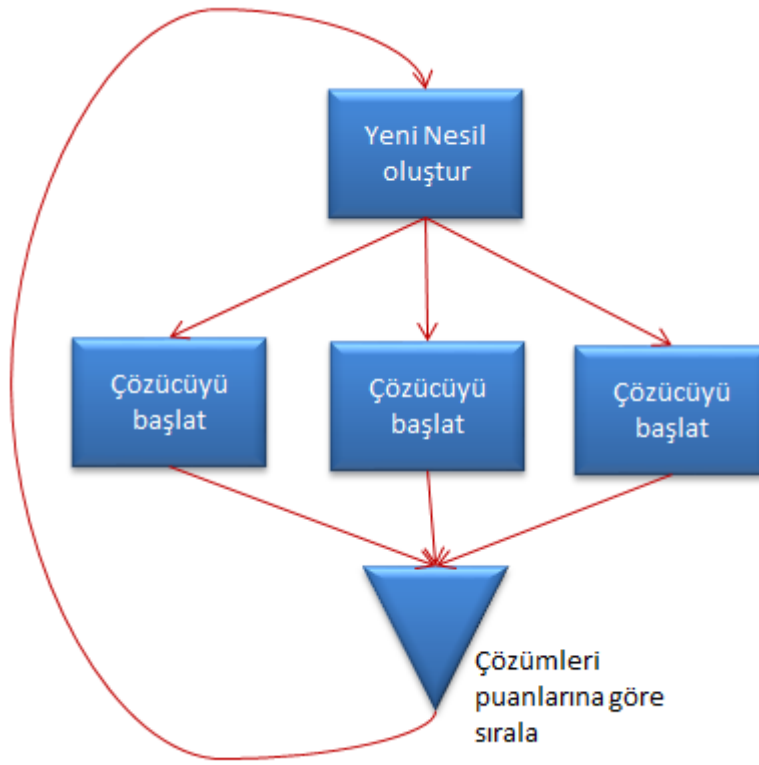
Ayrıca Mutasyon nesli seçildiğinde, son neslin incelenmesinden sonra en iyi kromozomlar mutasyona uğratılarak daha iyi sonuçlar aranmaktadır.

Yeni nesil hazır olduğunda, verilen seçeneklere göre bu nesilde mutasyonun uygulanma durumuna karar verilmekte ve devamında tüm kromozomlar çözücülere gönderilmektedir.

Çözücüler kullanıcının belirlediği sayıda çok kanallı olarak çalışmaktadırlar. Ancak nesil geçişlerinde tüm kanallar birleştirilmekte ve çaprazlama işlemi tek kanalda yapılmaktadır. Nesil-Çözücü ilişkisi aşağıdaki gibidir.

Yeni nesilde kromozom havuzunun oluşturulmasından sonra çözücüler aktif hale geçerler ve sırayla birer kromozom alıp simülasyonlarını koştururlar. Her çözücünün kendine ait bir model nesnesi vardır ve simülasyon her çözücünün kendi model nesnesinde çalıştırılır. Böylece aynı anda çalışan diğer çözücülerin o anki değerleri ile karıştırılmaz.

Simülasyonun her anında mevcut sistemin puanlaması hesaplanarak sonraki çaprazlama işleminde bu değerler dikkate alınır. Eğer en iyi değere simülasyon süresi tamamlanmadan ulaşılıyorsa, incelenen senaryonun süresi bu zamana indirilir.



Şekil 0.1. Nesil Döngüsü

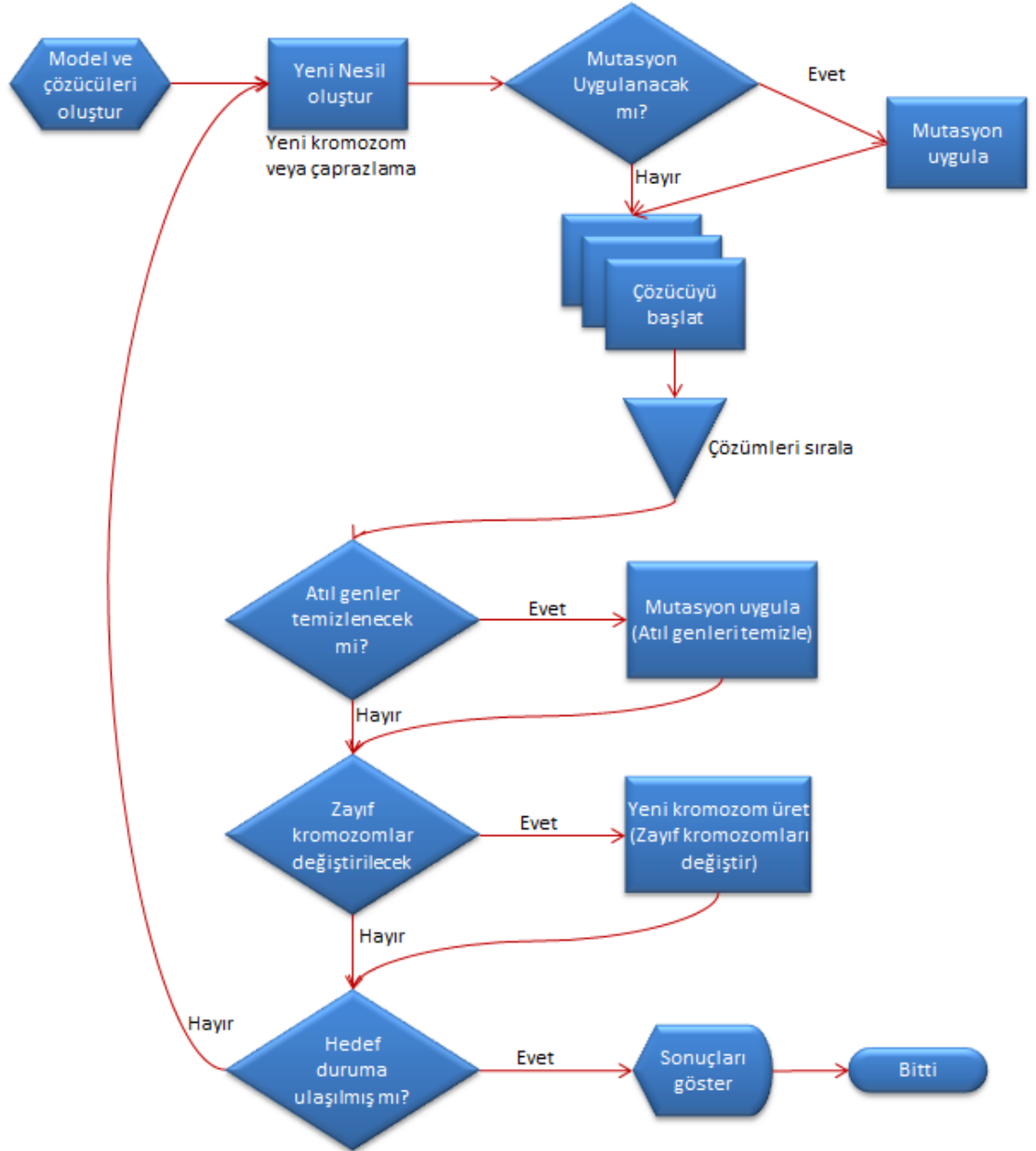
Modelde bazı olayların belli bir ihtimal dâhilinde gerçekleşmesi durumunda Monte-Carlo (Sobol 1974) simülasyonları prensibi gereği simülasyon yeteri kadar tekrarlanarak sonucun ortalama değeri alınır. Simülasyonun kaç kez tekrarlanması gerektiği ile ilgili hesaplama işlemleri Çözücüde açıklanmıştır.

Çözücü, incelediği senaryoyu tamamladığında sonuçlarını kromozom içine kaydeder ve populasyon nesnesinden yeni kromozom ister. Nesildeki kromozomlar bittiğinde populasyon yeni kromozom vermez. Nesildeki tüm kromozomlar tamamlanıp yeni nesil oluşturulduktan sonra populasyon tekrar kromozom dağıtımına devam eder.

Modelde sonuca olumlu-olumsuz herhangi bir etkisi olmayan olay genlerine atıl genler ismi verilmiştir. Bir neslin incelenmesinden sonra populasyon eğer atıl genlerin temizlenmesi isteniyorsa atıl genlerin temizlenmesi mutasyonu uygulanır. Atıl genlerin temizlenmesinin kaç nesilde bir yapılacağı ihtiyaca göre belirlenebilmektedir.

Tezimde incelenen nesil içindeki en kötü sonucu veren önceden belirlenen sayıdaki senaryolar zayıf kromozomlar olarak adlandırılmışlardır. Nesil içinde kromozomlarının ne kadarının zayıf kabul edileceği modeli tasarlayanın

inisiyatifindedir. Çözümde daha çabuk ulaşabilmek ve gen çeşitliliğini artırmak maksadıyla belli nesil aralıklarıyla en zayıf kromozomlar üreteç tarafından oluşturulan yeni kromozomlarla değiştirilmektedir.



Şekil 0.2. Çözüm Motoru Akış Diyagramı

En son neslin incelenmesinden sonra, mutasyon neslinde olduğu gibi, istendiği takdirde bir üstün nesil üretilebilmektedir. Bu nesil tüm nesillerden en iyi kromozomlar seçilerek yeni bir nesil oluşturulmakta ve bu nesilden çaprazlama yöntemiyle yeni bir

nesil türetilmektedir. Böylelikle en iyi kromozomlar çaprazlanarak daha iyi bir nesil üretmek amaçlanmaktadır.

Her neslin sonunda arzulanan hedef değere veya azami nesil sayısına ulaşıp ulaşılmadığı kontrol edilerek optimizasyonun bitirilmesine veya devamına karar verilir.

Çözüm Motorunun akış şeması Şekil 2.6'da gösterilmiştir. Anlatılan işlemler ile ilgili ayrıntılar dâhil oldukları nesneler ile birlikte anlatılacaktır.

2.3.3. Kromozom (Senaryo)

Her bir kromozom bir senaryoyu temsil etmektedir. Bir senaryoda zamana yayılmış olaylar, olaylara ait parametreler ve simülasyon sonucunda elde edilen istatistikler bulunmaktadır. Tüm simülasyon kromozom ve ait olduğu model üzerinde oynanmaktadır.

Senaryoyu oluşturan her bir olay, kromozom içerisinde bir *olay geni* ile temsil edilmektedir. Yeni oluşturulan bir senaryoda zaman doğrusuna dizilmiş planlı veya sabit olay genleri bulunmaktadır. Sabit olay genleri, her senaryoda mutlaka bulunması gereken olayları temsil etmektedir. Planlı olay genleri ise kromozomları üreten *üreteç* tarafından, planlanabilir olaylar arasından çeşitli tahditler göz önüne alınarak oluşturulan olay genleridir. Üçüncü olarak tetiklenen olay genleri bulunmaktadır. Tetiklenen olay genleri, modelin kendisinde bulunmayıp, çeşitli durumların oluşmasıyla tetiklenen olaylardır ve simülasyonun koşulması esnasında gerekli şartların yerine gelmesi sonucunda üretilerek senaryoya dâhil olurlar. Diğer olay genleri gibi tetiklenen olaylar da kromozomun zaman tablosuna işlenir ve istatistikleri tutulur.

Kromozomun zaman tablosunda her bir anında birden fazla olay bulunabilir. Bu durumda olaylar aynı anda aynı durum değerleriyle başlatılır. Eğer bir olayın sonucu aynı nesnenin aynı özelliğini aynı anda etkiliyorsa bu durumda olayların etkilerinin birbirine bağımlı veya bağımsız olmasına göre ayrı ayrı işlenmekte veya etkinin ağırlıklı ortalaması alınmaktadır. Olayın durum değerine anlık etki etmesi gerekiyorsa, aynı zaman birimindeki sonra gelen olay güncellenen değere göre işlem görür. Daha farklı etkiler için 2 veya 3 aşamalı durumlar tanımlanabilir. Durum hesaplamaları ileride ayrıntılı olarak ele alınacaktır.

Her bir kromozom üzerinde simülasyon istatistikleri tutulmaktadır. Bu istatistiki değerler aşağıdaki bilgileri kapsamaktadır;

- Kromozomun genel puanı,
- Azami puana ulaşılan simülasyon anı,
- Tekrar (iterasyon) sayısı,
- Her bir tekrar ile ilgili sonuçlar; Simülasyonda rastsal değerler olduğundan simülasyon verilen güvenilirlik derecesine göre yeteri kadar tekrar edilir.
- Kromozomun safahatı; Kromozomun hangi kromozomlardan türetildiği, uğradığı mutasyonlar gibi bilgiler burada yer alır.
- Her bir olay için olay genlerinde;
 - Olayın gerçekleşip gerçekleşmediği,
 - Olayın hangi oranda gerçekleştiği,
 - Olayın oluşmasının belli bir ihtimal dâhilinde oluşma şartının olması durumunda bu ihtimal değeri için hangi değer hesaplandığı ve ihtimalin gerçekleşip gerçekleşmediği,
 - Olayın meydana gelebilmesi için bir koşul varsa, bu koşulun oluşup oluşmadığı,
 - Çeşitli sebeplerle olayın oluşmama nedenleri: olayın meydana gelme zamanı ile ilgili tahditler; olayın azami oluşma sayısı, olayın asgari oluşma aralığı, vb.
 - Olayın atıl olup olmadığı,

Tüm bu bilgilere simülasyon sonunda ulaşılabilir.

Yeni bir kromozom oluşturulduğunda, kromozom sırayla şu işlemleri yapar;

- Kendini Popülasyona, dolayısıyla geçerli nesle ekler.
- Popülasyondan sıradaki kromozom numarasını alır. Bu numara tüm çözücülerde tektir ve kromozomun ayırt edilmesini sağlar.
- Puan, Süre, İterasyon sayısı gibi değerlerine sıfır değerini atar.
- Senaryoda olay genlerini saklamak üzere kullanılan zaman doğrusunu oluşturur ve sabit olayları çözücülerde tutulan taslaklarından kendine kopyalar.

2.3.3.1. Olay Genleri

Olay genleri, kromozomu oluşturan parçalardır. Olaylar, kromozoma olay genleri ile yerleştirilirler. Olay geninde yukarıda bahsedilen istatistiksel değerler ile birlikte şu bilgiler yer almaktadır;

- Olay,
- Fail; olayı kimin yaptığı veya tetiklediği bilgisidir. Sadece bir adet nesne olabilir.
- Etkilenen; olaydan etkilenen nesneleri temsil eder. Burada birden çok nesne yer alabilir.
- Param; nesne olabileceği gibi, cisim veya küme de olabilir. Buraya parametre olarak birden çok nesnesel varlık eklenebilmektedir. Olaydan dolaylı olarak etkilenen veya olayla ilgili nesneleri etkileyen nesne bilgilerinin burada yer alması düşünülmüştür. Örneğin; A şahsı B'ye C hakkında bir bilgi veriyorsa B'nin C hakkındaki bilgisi bundan etkilenecektir. Veya A bir konuda C'yi kendine örnek alıyorsa olayda C'nin bazı durum değerleri kullanılacaktır. Bu tür durumlarda C nesnesi Param olarak olay genine eklenmektedir.
- Olay geninin tetiklenme şekli; Sabit, planlı veya tetiklenen olabilir.
- Olayın meydana gelme zamanı.
- Olayın oluşma ihtimali.
- Olayın varsayılan oranı. Olay tam etkileriyle veya kısmen meydana gelmiş olabilir. Eğer olay tam gerçekleşmemişse veya etkileri belli bir duruma göre sınırlanıyorsa *oran* değeri kullanılarak olayın etkilerinin ne ölçüde olacağı sisteme aktarılabilir.

Olay geni aslında olay için parametrik değerleri ve sonuçlarını tutmaktadır. Olay geni içindeki değerler, olayın eylemi tarafından kullanılabilir. Böylece modeli hazırlayan personel, olaya karışan nesneleri ve diğer parametreleri kullanabilir.

Bir olay genini oluşturmak için kullanılan iki farklı tipte kurucu fonksiyon tasarlanmıştır; İlki tamamen yeni bir olay geni oluşturmak için kullanılmaktadır. Bu fonksiyon birer Nesne, Olay ve zaman parametresi almaktadır. Olay zorunlu parametre olup Nesne ve zaman verilmeyebilir. Olayın faili önemli değil veya bir taraf değilse veya olay ortam, sistem gibi genel unsurlar tarafından tetikleniyorsa Nesne boş

bırakılabilir. Zaman verilmemişse değeri “0” olarak alınır ve daha sonra değiştirilebilir. Algoritma, olay genini bu şekilde oluşturup zaman değerini olayı atadığı zamana göre sonradan verebilir.

Olay geni oluşturulduğunda sırasıyla aşağıdaki işlemler yapılır;

- Sistem tarafından verilen son kimlik numarasını alır ve değerini bir artırır. Bu şekilde tüm çözümler içinde her olay genine farklı bir kimlik numarası verilerek ayırt edilmesi sağlanır.

- Temsil ettiği olay ile ilişkilendirilir.

- Olay geninin oran değeri atanır. Bu değer varsayılan olarak “1”dir.

- Olayın oluşma ihtimali değeri belirlenir. Olaya ait oluşma ihtimali değeri sabitse, olduğu gibi alınır. Bu değer bir aralık olarak verilmişse bu olay geni için belirtilen aralıkta bir rastsal bir değer, olayın oluşma ihtimal değeri olarak belirlenir.

- Eğer bir nesne verilmişse, bu nesne olayı gerçekleştiren, yani “Fail” olarak atanır.

- Olay geninin vuku bulacağı zaman değeri atanır.

- Olay geninin tetiklenme tipi “standart” olarak belirlenir. Eğer olay geni Sabit veya tetiklenen ise değeri sonradan değiştirilir. Sistemin sağlıklı işleyişi açısından tüm değerlere ilk oluşumlarında bir varsayılan değer atanır.

- Olay geninin temsil ettiği olayın simülasyon sonucunda oluştuğu, olaya ait şartın gerçekleştiği, olayın gerçekleşme ihtimalinin oluştuğu bilgilerine başlangıçta “hayır” değeri atanır. Bu durumlar gerçekleştiğinde değer “evet” olarak değiştirilir. Etkilenen ve ilgili nesneler dizisi boş olarak başlangıç durumuna getirilir.

Diğer tip fonksiyonlar ise kopyalama fonksiyonlarıdır. Bu fonksiyonlar kendi çözücüsündeki veya başka bir çözücündeki olay genini kopyalayarak çoğaltırlar.

Bu fonksiyonlar çalıştıklarında olay genini oluştururken referans verilen olay geninin değerlerini kendine kopyalar. Ancak olay geninin temsil ettiği olayın simülasyon sonucunda oluştuğu, olaya ait şartın gerçekleştiği, olayın gerçekleşme ihtimalinin oluştuğu bilgilerine “hayır” değeri atanır

Birinci tip kurucu fonksiyondan farklı olarak etkilenen ve ilgili nesne bilgileri de doldurulur. Bu iki dizi için öncelikle bir bellek ayrılır ve nesneler sırasıyla kopyalanır. Kopyalanan olay geni farklı bir çözücünde ise Fail, etkilenen ve ilgili nesneler isimleri

kullanılarak kendi çözücüsünün kütüphanesindeki kopyası bulunarak ilişkilendirilir. Farklı çözücüler üzerindeki hiçbir nesne, olay veya durum değeri ortak kullanılmaz. Her çözücüde ayrıca bir kopya oluşturulur. Çünkü bir nesne ortak kullanıldığında, değerleri o anda çalışan başka bir çözücünün değerleri ile karışması kaçınılmaz olur.

2.3.4. Populasyon

Populasyon, esas itibariyle tüm nesillerin tutulduğu sınıftır. Modelde ayrıca nesiller ile ilgili işlemleri yönetir.

Populasyon, üreteç sınıfından nesil mevcudu kadar kromozom talep eder ve bunlarla ilk nesli oluşturur. Çözücülerden gelen kromozom taleplerine cevap verir ve sırayla birer kromozom gönderir. İşlenen kromozomları, istatistikleriyle birlikte ilgili nesilde tutar. Nesildeki tüm kromozomların incelenmesinden sonra sonuçlarına göre kromozomları sıralar. En iyi kromozom hedef kriterleri tutturmuşsa veya azami nesil sayısı oluşmuşsa sisteme optimizasyon işleminin son bulması gerektiğini bildirir.

Atıl genlerin temizlenmesi istenmişse senaryo puanına, senaryo sonuna kadar hiçbir etkisi olmayan veya hiç gerçekleşemeyen olaylar senaryodan çıkarılır. Böylece etkisiz olaylar senaryodan çıkarılmış olur.

Zayıf kromozomların değiştirilmesi istenmişse, belirlenen oran kadar o nesildeki en kötü sonucu alan kromozomlar yerine üreteçten yeni kromozom istenir. Bu şekilde başarısız genler yerine farklı genlerin denenmesi amaçlanmıştır.

Optimizasyon devam edecekse, nesil ağırlıklı rastgele seçim metodu (Çunkaş 2006: 19) ile ikişer kromozomdan çaprazlanarak yeni bir nesil oluşturulmaktadır. Çaprazlama konusu Üreteç içinde ele alınacaktır.

Populasyonun yapısı şu şekildedir;

Tüm nesillerin tutulduğu ve üyeleri Kromozom olan iki boyutlu bir “Nesiller” dizisi vardır. İlk boyut her bir nesil için, diğeri ise her nesildeki kromozomlar içindir. Ayrıca erişimi kısaltmak için kullanılan, simülasyonun gerçekleşmesi esnasında geçerli nesli tutan bir “Nesil” dizisi mevcuttur.

Toplam nesil sayısını, üstün ve mutasyon nesilleri ile birlikte toplam nesil sayısını, hedefe ulaşmak için belirlenen puan için kabul edilebilir sapma oranını,

populasyon sayısını, geçerli kromozom numarasını gösteren sayısal değişkenler mevcuttur. Ayrıca populasyonun kendi içinde kullandığı, dışardan erişilemeyen problemin çözülme durumu, özet rapor, sıradaki nesil, en son mutasyon yapılan nesil gibi bazı değerleri vardır.

Populasyon içinde bulunan fonksiyonlar ve işlevleri aşağıdaki gibidir;

YeniNesil: Yeni bir nesil oluşturur.

- Eğer geçerli nesil -1 ise “İlkNesliOlustur” fonksiyonu çağrılır.

- Geçerli nesil -1’den büyükse Sıradaki nesil değişkenine bakılır;

- 1 için “YeniNesilOlustur” fonksiyonu,

- 2 için “MutasyonNesli” fonksiyonu,

- 3 için “UstunNesil” fonksiyonu çalıştırılır.

- Sistem içindeki atıl hafıza boşaltılır. Bu işlemin her ne kadar işletim sistemi tarafından yapılması beklense de simülasyon esnasında her bir kromozom için ayrı ayrı yapılan tanımlama ve senaryo sonunda kromozomun kapatılmasıyla boşalan hafıza yeterince güncel temizlenememektedir ve program tarafından nesil aralarında temizlenmesi gerekmektedir.

-“Mutasyon Uygulanacak” bilgisi Sistem ayarlarında mutasyonun uygulanıp uygulanmayacağı ve mutasyon uygulanan son nesilden beri mutasyon aralığı kadar nesil geçip geçmediği kontrol edilerek belirlenir.

- Aynı işlem Atıl genlerin temizlenmesi ve zayıf kromozomların değiştirilmesi bilgileri için de benzer şekilde uygulanır.

MutasyonNesli: Mutasyonların uygulandığı nesil işlemleridir.

Populasyona yeni bir nesil ekleyerek, önceden tüm nesiller içindeki en iyi kromozomların kopyalanarak oluşturulduğu bir nesli kullanarak, buradaki her bir kromozomu “MutasyonUygula” fonksiyonu ile mutasyona uğratarak yeni nesle ekler.

UstunNesil: Populasyona yeni bir nesil ekleyerek, önceden tüm nesiller içindeki en iyi kromozomların kopyalanarak oluşturulduğu bir nesli “Caprazla” fonksiyonunu kullanarak, buradaki kromozomları çaprazlayarak yeni bir nesil oluşturur.

İlkNesliOlustur: İlk nesli oluşturan fonksiyondur.

- Geçerli nesil değerini bir artırır.
- Populasyonun “Nesil” dizisini oluşturur. Bu aslında nesil dizisini içinde tutan bir nesne yapısıdır. Bu nesnenin oluşturulması sağlanır.
- Oluşturulan Nesil dizisi Nesiller dizisine eklenir.
- Populasyon sayısı kadar kromozom üretilerek içini doldurması için Üretcin “Rastgele Çözüm Oluştur” fonksiyonuna gönderilir. Her kromozom ilk oluşturulduğunda kendi otomatik Nesil dizisine eklediğinden oluşturulan kromozom nesillere kendiliğinden eklenmektedir.

YeniNesilOlustur: Sonraki nesli oluşturur.

- Geçerli nesil değerini bir artırır.
- Eski nesil adıyla bir dizi oluşturur ve nesil dizisini kopyalar.
- Nesil dizisinde yeni boş bir nesil oluşturur.
- Nesil dizisini Nesillere ekler.
- Eski nesli çaprazlayarak geçerli nesli oluşturur.

Çaprazlama:

Puanlama Bölümü

- En düşük puan değerini tam puan olan “1” yapar.
- Uzunluğu populasyon sayısı olan bir “Kromozom Puanları” dizisi oluşturur.
- Toplam Puan değerini “0” yapar.
- En düşük puan, değerini eski nesilde kendi içinde karşılaştırarak kendinden daha düşük bulduğu değeri alır.
- Kromozom puanı dizisine kendi karşılığı olan kromozomun puan değerinin en düşük puandan farkını atayarak bu değeri Toplam puana ekler.
- Kümülatif toplam değerine “0” atanır.
- Her kromozom puanı toplam puana bölünerek kümülatif puana eklenir ve kümülatif puanın o anki değeri kromozom puanının değerine atanır.

Buraya kadar tüm kromozom puanları en düşük olanın değeri kadar düşürülerek toplam puan içindeki oranı bulunmuş ve kümülatif olarak değerlendirilmiştir. Bundan sonra 0 ile 1 arasında verilecek bir rastsal sayı bizi bir kromozoma götürecektir ve bu kromozomun seçilme ihtimali puanı oranında olacaktır. Her kromozomun değerinin en düşük olanın değeri kadar azaltılması ile düşük puanlı kromozomların seçilme ihtimali azaltılmış, dolayısıyla yüksek puanlı kromozomların seçilme ihtimali artırılmıştır. Bunun nedeni daha yüksek puanlı bir kromozomdan çaprazlanacak kromozomların da daha yüksek puanlı olmalarının daha yüksek bir olasılığa sahip olduğu varsayımdır.

Çaprazlama Bölümü

- Seçilecek kromozomu tespit etmek için bir “d” değişkeni üretilir.
- Çaprazlanacak iki kromozomun sıra numaralarının tutmak üzere 2 uzunluğunda bir “a” dizisi oluşturulur.
- Yeni neslin her kromozomu oluşturulana kadar aşağıdaki işlemler tekrarlanır;
- İki kez tekrarlanan bir döngü başlatılır.
- d’ye 0 ile 1 arasında rastgele bir değer atanır.
- Bir “i” değeri oluşturulur ve -1 değeri atanır.
- i değeri döngü değeri olarak kullanılarak Kromozom puanları içinde d’nin karşılık geldiği kromozom sıra numarası bulunur. i değeri döngü içinde bu sıra numarasını almış olur.
- a dizisinin döngü değerine göre kromozomun sıra numarası (i) atanır. Böylece aynı kod iki defa yazılmayarak çaprazlanacak iki kromozom da aynı şekilde belirlenir.
- Eğer ikinci döngüdeyse ve seçilen iki kromozom aynı ise döngü değeri düşürülerek ikinci döngü tekrarlanır. Böylece kromozomun kendisiyle çaprazlanması engellenmiş olur.
- Eski nesil içindeki kromozomlar sıra numaraları ile bulunarak birer kromozom değişkenine atanırlar ve Üretcin “Çaprazla” fonksiyonuna gönderilerek çaprazlanan yeni kromozomu nesile ekler.
- Kromozomun işlem özetine çaprazlandığı ebeveyn kromozomlar not edilir.

Mutasyon: Nesildeki mutasyon görece kromozomları belirler ve mutasyon görmelerini sağlar.

MutasyonAtılGen: AtılGenTemizleIslem fonksiyonunu tetikler.

AtılGenTemizleIslem: Atıl genlerin temizlenmesini sağlayan fonksiyondur.

Sırasıyla aşağıdaki işlemler gerçekleştirilir;

- Son atıl genin temizlendiği nesli geçerli nesil olarak atar.
- Geçerli nesil içindeki tüm kromozomlara aşağıdaki işlemleri uygular;
 - Sıradaki kromozomu alır.
 - “İşlendi” değişkeni oluşturulur ve hayır değeri atanır.
 - Kromozomun tüm zamanları için aşağıdaki işlemler tekrarlanır;
 - Her bir zamandaki tüm olay genleri için aşağıdaki işlemler tekrarlanır;
 - Geçerli zamanda sıradaki olay geni alınır.
 - Bu olay geninin puanlama esnasında doldurulan “Gen Atıl mı?” değeri kontrol edilerek eğer atılsa aşağıdaki işlemler uygulanır;
 - kromozomun işlem özetine Olayın adı ve hangi zaman biriminde bulunduğu bilgileri ile birlikte silindiği not edilir.
 - Olay geni kromozomdan çıkarılır.
 - İşlendi bilgisi evet olarak atanır.
 - İşlendi değeri evetse kromozomun işlem özetine “Atıl genler silindi” bilgisi eklenir.

MutasyonIslemSayisi: Sadece Populasyon içinde kullanılan ve izin verilen farklı mutasyon çeşidi sayısını veren fonksiyondur.

MutasyonUygula: İstenilen kromozom için mutasyonun uygulandığı fonksiyondur.

- İşlem çeşidi sayısını “MutasyonIslemSayisi” fonksiyonu ile öğrenir.

- İşlem çeşidi sayısı 0'dan büyükse sistemdeki tüm olayları bir dizide toplayarak kromozomdaki her zaman birimi ve her zaman birimindeki her olay geni için aşağıdaki işlemleri gerçekleştirir;

- Sistem ayarlarında belirtilen mutasyon oranına göre olay genine mutasyon uygulanacağına rastsal olarak karar verilir.

- Eğer uygulanacaksa;

- Sıfır ile İşlem çeşidi aralığında bir sayı rastsal olarak seçilir. Seçilen bu sayı, seçilebilen mutasyon çeşitleri arasında bu olay genine uygulanacak mutasyon işleminin sıra numarasını gösterir.

- Seçilen Mutasyon türüne göre;

- Gen Sil: Seçili olay geni silinir.

- Gen Değiştir, Parametre: Olay genine Olayın "CreateParam" fonksiyonu kullanılarak oluşturulan Nesne referansları Üretcin "ParamAta" fonksiyonu ile atanırlar.

- Gen Değiştir, Zaman: olay geni bulunduğu zamandan çıkarılarak, rastsal olarak belirlenen başka bir zaman birimine eklenir.

- Gen Değiştir, Olay: Olay geni silinerek yerine rastsal olarak yeni bir olay atanır ve "Gen Değiştir, Parametre"de anlatıldığı şekilde parametreleri belirlenir.

- Gen Ekle: Planlanabilir olaylar içerisinde bir olay seçilerek yeni bir olay geni oluşturulur ve "Gen Değiştir, Parametre"de anlatıldığı şekilde parametreleri belirlenir. Müteakiben genin ekleneceği zaman belirlenerek buraya eklenir ve tetiklenme şekli "planlı" olarak atanır.

- Yapılan tüm mutasyon işlemlerinde kromozomun işlem özetine not alınır.

- Eğer mutasyon için belirlenen seçenekler içerisinde zaman kaydırması varsa, 1 ile 10 arasında rastsal olarak belirlenen kadar zaman birimindeki tüm olaylar zamanın sonuna kaydırılır ve tüm senaryodaki zaman bilgileri güncellenir.

- Eğer akıllı mutasyon yapılmasına izin verilmişse aşağıdaki işlemler uygulanır;

- Akıllı mutasyonda yapılan işlem; mutasyon oranı kadar en az faydalı olay genlerinin, sisteme daha çok faydası olan olay genleri ile değiştirilmesidir.

- Bu maksatla kromozom içindeki tüm zaman birimlerinin kendilerine ait puanları ve ait oldukları zaman birimi bir listeye eklenerek burada sıralanırlar.

- Mutasyon oranı kadar en düşük puanlı olay genleri, aynı miktardaki en yüksek puanlı olay genlerinin kopyaları ile değiştirilirler.

ZayıfKromozomUygula: Nesildeki en zayıf kromozomları, yeni üretilen kromozomlarla değiştirir.

- Zayıf kromozomun yapıldığı son nesli günceller.

- Değiştirilecek zayıf kromozom sayısını sisteme girilen seçeneklere göre belirler.

- Tüm kromozomları puanlarına göre ayrı bir listede sıralar.

- En düşük puanlı değiştirilecek zayıf kromozom sayısı kadar kromozom için aşağıdaki işlemler yapılır;

- Kromozomun işlem özetine yeniden oluşturulduğu bilgisi eklenir.

- Kromozomun durumu “Bekliyor” olarak güncellenir. Bu şekilde kromozomun simülasyonu yapılabilir. Çözücü sadece durumu bekliyor olarak görünen kromozomların simülasyonlarını koşturur.

- Üreticinin “RastgeleCozumOlustur” fonksiyonu ile yeni bir kromozom üretilir.

- Sistem aracılığı ile geçerli çözücüye ulaşılarak kromozomun oynanması ve böylece istatistiki değerlerinin oluşması sağlanır.

HedefDurumu: Çözümün hedefe ulaşp ulaşmadığını kontrol eder.

- “SonucAl” fonksiyonunu çağırır.

- Nesildeki her kromozomun puanını kontrol ederek neslin en yüksek puanını bulur.

- Optimizasyonun durması gerektiğinde kullanılacak nesil sayısı ve nesildeki en iyi kromozomun izin verilen sapma kadar hedefe yaklaşma durumu kontrol edilerek belirlenir.

- Eğer optimizasyonun durması gerektiğine karar verilmişse sıradaki nesil tipi bir artırılır. Sistem ayarlarında Mutasyon nesli veya üstün nesil çalıştırılması yoksa sıradaki nesil tipi tekrar artırılarak bu işlemlerin atlanması sağlanır.

KromozomAl: Çözücünün sıradaki kromozomu alması için kullanılır.

Fonksiyon, başlangıçta tüm diğer işlem kanallarına karşı kendini kilitleyerek aynı anda bir kanalda çalışmasını teminat altına alır. Bu şekilde aynı anda tek kromozomun dağıtımını yapılarak kromozomların sağlıklı dağıtılması sağlanır. Başka bir kanal bu fonksiyonu kullanmak istediğinde fonksiyonun işini bitirmesini bekler. Fonksiyon işlem sonunda kendini çözerek başka bir kanalın kendisini kullanmasına izin verir.

Fonksiyon, nesildeki popülasyon sayısı aşılmadıkça sıradaki kromozomu gönderir. Popülasyon sayısına ulaşılmışsa geriye bir şey döndürmeyerek talep eden çözücünün işlemin bittiğini anlamasını sağlar.

SonucAl: HedefDurumu'nda kullanılır. En iyi sonuçları ayrı bir listeye sıralı olarak toplar.

InitSonuc: Çözüm Motorunda başlangıçta kullanılır. En iyi kaç çözümün tutulması gerektiğine karar verir ve ilgili alanların oluşturulmasını sağlar.

Sonuc: Çözüm Motorunda en sonda kullanılır. En iyi sonuçların puan bilgilerini kaydeder.

Kaydet: Modelin çözümü için verilen tüm seçenekleri bir dosyaya kaydeder.

Yukle: Dosyaya kaydedilen çözüm seçeneklerini yükler ve geçerli ayarlar olarak kabul eder.

2.3.5. Üreteç

Üreteç, kromozomların ilk defa üretilmelerini ve çaprazlamalarını sağlayan sınıftır. Kromozomların baştan üretilmesi sırasında verilen kısıtlara göre rastgele bir senaryo üretir. Bunun için de planlanabilir olayları sırayla ele alarak kaç tekrar üreteceğine karar verir ve uygun parametreler atayarak izin verilen zaman aralığına yerleştirir.

Çaprazlama içinse 5 farklı yöntem seçilebilmektedir;

Tek Noktadan Bölünme: Bu yöntemde iki senaryo tek noktadan ikiye bölünerek parçalarını birbirleriyle değiştirirler. Bölünme yeri rastgele seçilerek iki senaryoda da aynı yerden bölünürler.

Grup Olarak Bölünme: Bu yöntemde her iki senaryoda eşit anlarda 2 ilâ 10 arasında zaman birimi uzunluğunda gruplandırmalar oluşturulur. Çaprazlanan kromozoma birinci senaryodaki tek gruplar, ikinci senaryodaki çift gruplar sırayla atanır.

Puanlama: Bu çaprazlama yönteminde kromozomdaki genler yani senaryodaki olaylar incelenerek iki senaryonun belirli bir anında faydalı olan olaylar yeni kromozoma kopyalanmaktadır. Bu metodun uygulanabilmesi için senaryonun, simülasyon esnasında her birim zaman sonunda ulaştığı puan değeri kaydedilmektedir. Böylece iki senaryodan hangi birim zamandaki olayların oluşumlarının yeni çaprazlanan senaryoya kopyalayacağına karar verilirken, senaryolardaki bir önceki birim zaman puanı ile olan farkı alınarak hesaplanan anlık puanlar karşılaştırılarak yüksek olanı seçilir ve bu puan, kabul edilebilir puan eşiğinin üzerindeyse çocuk kromozoma kopyalanır.

Aslında bu çaprazlama yönteminin içinde bir mutasyon oluşumu da vardır. Kabul edilebilir puan eşiği, kullanıcı tarafından ayarlanabilmektedir. Tezde varsayılan olarak -0,02 verilmiştir. Bu eşik değeri ile belli bir puanın altındaki genlerin çocuk kromozomda olmamaları sağlanmaktadır. Böylece sonuca zarar veren olaylar senaryodan çıkarılmakta ve sonuca daha çabuk ulaşılabilir. Kabul edilebilir puan eşiğinin negatif bir değer olması daha uygundur. Çünkü bazen bir olay sistemi amaçlanan sonuçtan uzaklaştırabilmekte, ancak bu işlem daha sonra sonuca daha çok yaklaşılmaya sebebiyet verebilmektedir. Kabul edilebilir puan eşiği ile ne kadarlık puan düşüşüne izin verilebileceği belirlenmektedir.

Puanlama için kullanılan “Caprazla2” fonksiyonu analiz ve tam analiz yöntemlerini gerçekleştiren “Caprazla3” ve “Caprazla4” fonksiyonlarıyla arasında sadece küçük farklar vardır. Temel fark puanlama şeklindedir.

Caprazla2 fonksiyonun C# kodu aşağıda gösterilmiştir. Gruplanarak kapalı gösterilen “Tahditleri sağla” bölümü ise tüm çaprazlama fonksiyonlarında standart olup çaprazlanan kromozomun verilen tahditlere uygunluğunu kontrol eder ve Çocuk kromozomu sonuç olarak döndürerek “while (true)” sonsuz döngüsünü kırar. Deneme sayısı sistemde tanımlanan azami deneme sayısını aşması durumunda tahditler esnetilir. Bu değer varsayılan olarak 50 deneme şeklinde ayarlanmıştır.

```

public static Kromozom Caprazla2(Kromozom a, Kromozom b) // Puanlama
{
    Kromozom Cocuk = new Kromozom();
    Kromozom[] x = new Kromozom[2];
    x[0] = a;
    x[1] = b;
    int dene = 0;
    while (true)
    {
        int i = 0, y;
        if (Sistem.Mutasyon) Cocuk.AnlikPuanlar = new Collection<double>();

        // genlerini kopyala
        while (++i < Sistem.Sure)
        {
            y = a.AnlikPuanlar[i] > b.AnlikPuanlar[i] ? 0 : 1;
            if (x[y].AnlikPuanlar[i] > Sistem.KabulEdilebilirPuanEsigi) // bir e
                foreach (OlayGeni g in x[y].zaman[i - 1])
                {
                    if (g.tetik == Tetik.Planli)
                    {
                        Cocuk.zaman[i - 1].Add(g);
                    }
                }
            if (Sistem.Mutasyon) Cocuk.AnlikPuanlar.Add(x[y].AnlikPuanlar[i]);
        }
        Tahditleri sagla
    }
}

```

Analiz: Seçimli çaprazlama metodunun daha ayrıntılı sürümüdür. Bu yöntemde her bir olayın simülasyon sonuna kadar olan etkileri hesaplanmaktadır. Bu maksatla bir önceki zaman birimine ait puan değeri kaydedilir. İçinde bulunulan zaman birimine ait olaylar gerçekleştirilir. Bundan sonra simülasyon sonuna kadar yeni planlanabilir olay eklenmez. Sadece sabit ve tetiklenen olaylar çalıştırılır. Simülasyon sonundaki puan değeri ile bir önceki zaman biriminin puan farkı bu zaman biriminin puanı olur.

Bu yöntem ile her olay grubunun sisteme olan gerçek etkisi bulunmaktadır, çünkü tetiklemeler ve etki değişkenleri de dikkate alınır hale gelmişlerdir.

Tam Analiz: Bu metotsa analiz yönteminin daha ayrıntılı bir halidir. Bu yöntemde tüm zaman birimindeki olaylar senaryodan tek tek çıkarılarak denenirler. Böylece çıkarılan olayın tüm sistem içindeki değeri görülür. Burada senaryonun olay varken aldığı puandan olay çıkarılarak test edildiği puan hesaplanır ve olayın sisteme olan gerçek katkısı bulunur.

Analiz ve tam analiz yöntemleri simülasyonların çok uzamasına neden olmaktadır. Bu nedenle modellerin hazırlanmasında seçimli çaprazlama metodu tercih edilmektedir. Olayların birbiriyle etkileşimi olmadığı durumlarda da bu yöntem tercih edilmelidir. Olaylar başka olayları tetikleyebiliyorsa analiz yöntemi, olaylar kendinden sonra gerçekleşen olayları herhangi bir şekilde etkileyebiliyorlarsa tam analiz yöntemi kullanılmalıdır.

ParamAta: Bir olay genine gerekli parametrelerin atanmasını sağlar. Fail nesne bu fonksiyona hazır olarak verilir. Etkilenen ve ilgili nesne parametreleri olayın CreateParamList fonksiyonu kullanılarak oluşturulur. Parametre olabilecek etkilenen ve ilgili nesne listesinden fail çıkarılır.

Önce asgari ve azami etkilenen parametre sayısı aralığında etkilenen nesne tek olarak seçilir. Müteakiben parametre olabilecek ilgili nesne listesinden etkilenen nesne olarak seçilen nesneler çıkarılır. Böylece fail, etkilenen ve ilgili nesneler tek olarak seçilmiş olur. İlgili nesneler parametresi için de seçilecek nesne sayısı belirlenerek tek olarak seçilirler.

RastgeleCozumOlustur: Bu fonksiyon, rasgele bir kromozom oluşturur. Bunun için sistemde kayıtlı tüm olayları sırayla ele alarak eğer olay planlanabilir olarak tanımlanmışsa kendisine ait “OlayEkle” fonksiyonuyla belirlenen sayı kadar bu olayı kromozoma ekler.

OlayEkle: Yeni oluşturulan bir kromozoma bir olay ekler. Rasgele oluşturulan kromozomlar için kullanıldığından bir olay geni oluşturur. Parametrelerini ParamAta ve olaya ait CreateParam fonksiyonlarını kullanarak atar. Olayın meydana gelebileceği ilk ve son birim zaman aralığında bir zaman belirler ve olay genini bu zamana ekler.

Uygunluk: Çaprazlama fonksiyonlarında tahditlerin sağlandığının kontrol edilmesi için kullanılır. Bir kromozom bu fonksiyona kontrol için gönderildiğinde; Sistemde tanımlı tüm olayları tek tek ele alarak aşağıdaki kontrolleri yapar;

Olayların toplamı ve her zaman birimindeki tekrarları sayılır.

Asgari olay sayısı kontrol edilir. Kontrol edilen olayın tekrar sayısı asgari tekrar sayısının altında ise; bu sayıya ulaşılan kadar bu olaydan kromozoma eklenir. Böylece

her olay için asgari olay tekrar sayısı tahdidi karşılanmış olur, ancak kromozom bu şartın yerine gelmesi için mutasyona uğramak zorundadır.

Azami olay sayısı kontrolünde de benzer işlemler yapılır. Asgari olay sayısı kontrolü işleminden farklı olarak incelenen olay için bir silme oranı belirlenir ve olaylar bu silme oranı kullanılarak rasgele silinir.

Olayın tekrar edebilmesi için geçmesi gereken asgari süre olarak tanımlanan asgari tekrar aralığı karşılanmadığında toplam olay sayısı asgari olay sayısını karşılıyorsa zamanından önce tekrarlanan olay silinir ve olay sayısı güncellenir. Eğer toplam olay sayısı yeterli değilse sonuç kromozom uygun değil olarak döner ve başka bir kromozomun üretilmesi istenir.

Azami olay aralığı karşılanmadığında fonksiyon doğrudan kromozom uygun değil olarak cevap verir.

2.3.6. Çözücü

Çözücü, modelin oluşturulduğu ve kromozomların simülasyon yöntemiyle puanlandığı nesne yapısıdır. Çözücü içerisinde modelin tüm unsurları ile birlikte bir kopyası oluşturulmaktadır. Her çözücü koşturacağı her simülasyon boyunca kendi model kopyasını kullanır. Bu şekilde aynı anda çalışan çözücülerin değerleri birbiriyle karıştırılmamış olur.

Çözücü popülasyon nesnesinden bir kromozom talep eder. Popülasyon sıradaki işlenmemiş kromozomu çözücüye gönderir. Eğer çözülecek kromozom kalmamışsa, yani nesil tamamlanmak üzere ise çözücü bekleme moduna geçer ve çözüm motoru tarafından tekrar aktifleştirilmeyi bekler.

Çözücü ilk olarak senaryoyu başlangıç durumuna getirir. Bu esnada kullanacağı fonksiyonları belirler. Eğer Analiz veya tam analiz çaprazlama yöntemlerinden biri seçilmişse senaryoların oynatılmasında farklı bir fonksiyon kullanılacaktır. Bu fonksiyon farklı istatistikler almanın dışında, senaryoyu her seferinde kaldığı yerden planlanmış olaylar olmadan simülasyonun sonuna kadar oynatıp puan değerlerini almaktadır. Bir başka seçim ise atıl olayların elenmesi durumudur. Eğer içinde bulunulan nesilde atıl olaylar elenecekse, yine buna yönelik ek istatistikler alacak farklı

bir fonksiyon çalışacaktır. Yine alınacak genel istatistiklere göre olayların içinde eylemi gerçekleştiren fonksiyon değiştirilebilmektedir.

Olaylar içerisinde rastsal sayıların olmasından dolayı aynı senaryo iki kez çalıştırıldığında farklı sonuçlar alınabilmektedir. Bu nedenle senaryonun belli bir doğrulukta olabilmesi için senaryonun yeteri kadar tekrarlanması ve sonuçlarının ortalamasının alınması gerekmektedir. Senaryonun kaç kez tekrarlanması gerektiği aşağıdaki şekilde hesaplanmaktadır.

Asgari iterasyon (tekrar) sayısı kadar simülasyon tekrarlandığında çözücü elde ettiği değerlere göre koşulması gereken tekrar sayısını hesaplar. Sonra eksik kalan sayı kadar simülasyon tekrar edilir. t tablosunda $\alpha = 0,025$ alınmıştır. Tekrar sayısının aşırı artmaması için bir üst sınır belirlenebilir. İterasyon hesabı şu adımlarla yapılmaktadır:

İterasyon Hesabı İçin Kullanılan değişkenler;

n : oynanan iterasyon sayısı

n^* : olması gereken iterasyon sayısı (Arzulanan tekrar sayısı)

h : verilen tekrar sayısında sonuçların toplandığı yarı aralık.

h^* : arzulanan yarı aralık.

S : varyans

$\bar{Y}$: Puan ortalaması

abh : Arzulanan Bağlı Hassasiyet (Kullanıcı tarafından belirlenir)

1. Asgari iterasyon sayısı (n) kadar simülasyonu tekrarla.

2. Arzulanan tekrar sayısını (n^*) hesapla;

$$h = \frac{t_n \times s}{\sqrt{n}}$$

$$h^* = \bar{Y} \times abh$$

$$n^* = n \times \left(\frac{h}{h^*} \right)^2 + 1$$

3. Eğer uygulanan tekrar sayısı arzulanan tekrar sayısından azsa

- a. n^* - n kez simülasyonu tekrarla
- b. Adım 2'ye dön.

Bazen n^* değerine ulaşıp tekrar hesaplandığında bu değer biraz daha ileri kaydığı görülebilir. Tekrar yeni n^* değerine ulaşılan kadar simülasyon tekrarlanır ve n^* değeri tekrar kontrol edilir. Genellikle bu döngü değer 3-4 kez tekrarlanmaktadır. Aşırı artma ihtimaline karşı Azami iterasyon sayısı belirlenmiştir, bu değere ulaşıldığında iterasyonlar bitirilir.

Çözüm sonunda elde edilen istatistiki değerler kromozom içerisine kaydedilerek saklanır.

Çözücü yapısı ve fonksiyonları;

Çözücü içerisinde 11 adet liste veya kütüphane bulunmaktadır. Kütüphane yapısı liste benzeri bir yapı olup elemanlarına isimleri ile ulaşılabilir.

Varlık listesi; sistemde dâhil olan bütün varlık ve varlıktan türemiş yapılar için burada bir referans tutulur. Çözücü, modelde adı geçen bir nesne, olay, durum, vb. varlıkları bu liste ile tanır.

Olaylar, Cisimler, Kümeler, Nesneler, Özellikler, Durumlar; varlıklar listesinde olmalarına rağmen işlemlerin daha hızlı yapılabilmesi ve gruplama ihtiyaçlarından dolayı ayrıca kendilerine ait listeleri tutulmaktadır.

NesneKoku; Nesnesel varlıkları (Nesne, Cisim, Küme) ayrıca tutmaktadır. Bu yapı üretcin olay genlerine etkilenen ve ilgili nesnesel varlık parametrelerini atama yaparken nesnesel varlık adaylarını belirlemek için kullanılmaktadır.

Tahditler; çözücünün simülasyon esnasında her zaman ilerlemesinde tahditleri sağlaması gerekmektedir. Tanımlanan tüm tahditler burada tutularak kontrolleri sırayla yapılmaktadır.

Taslak; her kromozomda yer alacak sabit olay genlerini tutan, bir boyutu zaman olan, iki-boyutlu bir listedir.

Tetikler; Simülasyonda belirli şartların oluşması durumunda ilgili olayın tetiklenmesini sağlarlar. Bir class yapısı içerisinde tanımlanan Tetikler listesi, tetik

ekleme fonksiyonları ile güçlendirilmiştir. İki fonksiyon da aynı işleve sahip olup, tek farkları birisinde tetiklemenin istenen bir zaman aralığında kısıtlanabilmesidir.

Çözücü içerisinde üzerinde simülasyonun gerçekleştirildiği zaman doğrusu üzerinde olay genlerini saklayan yapı mevcuttur. Ayrıca puan bilgilerinin saklandığı “AnlıkPuanlar”, iterasyonlara ait puan bilgilerinin saklandığı “PuanTablosu” ve hedef kriterlerin bulunduğu “KriterDurum” listeleri mevcuttur.

Çözücüye ait fonksiyonlar aşağıda anlatılmıştır;

BasaAl; Bu fonksiyon modeldeki tüm durum, puanlama, istatistiki bilgi ve tetiklenme verilerini ilk durumlarına getirir. Yeni bir kromozom geldiğinde, kromozom için simülasyonun tekrar

n_star; n^* değerini hesaplayan fonksiyondur. Bkz. İterasyon hesabı için Kullanılan değişkenler.

Tetikle; Planda yer almayan bir olayın istenen gecikme, tekrar sayısı ve tekrar aralığı ile tetiklenmesi sağlanır.

KromozomAl; İşlenecek bir kromozomu alarak simülasyona hazır hale getirir. Sırasıyla;

- Kromozomun durumunu “İşleniyor” olarak günceller.
- Bir zaman doğrusu oluşturur ve üzerine kromozomdaki tüm olay genlerini kopyalar. Tüm işlemler çözücü üzerindeki veri üzerinde yapılarak, kromozomun yapısının aslına sadık kalması sağlanır.

Yukarıda bahsedilen PuanTablosu, AnlıkPuanlar ve KriterDurum listeleri boş olarak oluşturulur. Kriter bilgileri modelin incelenmesi ile, puan bilgileri ise simülasyon koşulması esnasında doldurulacaktır.

KromozomAl4; Bu KromozomAl fonksiyonunun özel bir durumudur. Tam analiz çözüm yönteminde kromozom baştan, incelenen zaman olmaksızın denenmekteydi. Bu kromozom alma fonksiyonu diğerinden farklı olarak incelenen zamanı çözüm zaman doğrusuna kopyalamamakta ve KriterDurum listesini doldurmaktadır.

KromozomSonucuKaydet; Elde edilen istatistiki bilgileri, olay genlerinin atıl olma durumlarını kromozom üzerine kaydeder.

PuanKaydet; İterasyon sonunda elde edilen puan bilgilerini kromozomun işlem özetine kaydeder. En yüksek puanın kaydedildiği zamanı dikkate alarak bu zamandan sonraki olayları plandan çıkararak gösterir ve kriter olarak belirlenen durum değişkenlerinin ulaştıkları değerleri kaydeder.

DurumAl ve DurumDeğişikliği; Olayların atıl olup olmadıklarını anlamak için kullanılırlar. DurumAl, tüm durumların değerlerini “ÖncekiDurum” olarak kaydeder, DurumDeğişikliği ise olayın çalıştırılmasından sonra durum değerlerini önceki durumları ile karşılaştırarak fark arar.

ExecuteOlaylar; Verilen zaman doğrusu üzerinde, içinde bulunan geçerli zamanın olduğu noktadaki tüm olay genlerini simülasyonda çalıştırır. ExecuteOlaylarAtilmi fonksiyonu ise atıl olay genleri araştırıldığında kullanılır. Ek olarak DurumAl ve DurumDeğişikliği fonksiyonlarını kullanır. Çözücü hangi fonksiyonu kullanacağına iterasyon başında karar verir.

CozumHazirla; Taslak olaylar hazırlanır ve Çaprazlama metoduna göre kromozomu oynayacak uygun fonksiyon belirlenir.

Adımla; Simülasyon fonksiyonları içerisinde en önemli fonksiyonlardan biridir. Her birim zaman için ayrı ayrı çalıştırılarak mevcut birim zamandaki tüm olayların çalıştırılarak durum değerlerinin güncellenmesini sağlar. Sırasıyla aşağıdaki işlemler gerçekleştirilir;

- Olayların çalıştırılmasını sağlayan ilgili fonksiyonu çağırır.
- Her durum için DurumTamamla işlemini gerçekleştirir. Bu konu “Dönem Sonu İşlemlerinde” ayrıntılı olarak anlatılmıştır.
- Durum tamamlama işlemlerinde toplu yapılması gereken işlemler yapılır. (bkz. Dönem Sonu İşlemleri 6 ilâ 9. alt maddeler)

ResetTetik; Tüm kayıtlı tetikleri başlangıç konumuna getirir.

IterasyonOyna; Bir kromozom için baştan sona gerekli iterasyon sayısı kadar bir simülasyonu oynatır.

Fonksiyon içerisinde gerçekleşen işlemler sırasıyla şöyledir;

- Boş bir puan tablosu oluşturulur.
 - Kullanılacak olay işleme fonksiyonu belirlenir.
 - İstenen iterasyon sayısına ulaşınca kadar aşağıdaki işlemler tekrarlanır;
- Aşağıda ismi geçen fonksiyonlar bu bölümde açıklanmıştır.

- ResetTetik çalıştırılır.
- BasaAl çalıştırılır.
- Tüm birim zamanlar için Adımla çalıştırılır.
- PuanKaydet çalıştırılır.
- n_star fonksiyonu ile gerekli iterasyon sayısı hesaplanır ve azami iterasyon sayısı da göz önünde bulundurularak istenen iterasyon sayısı artmışsa güncellenir.
- Tamamla çalıştırılır.

KromozomOyna; Bu fonksiyon, kromozomun sistemden alınmasından simülasyonların oynanıp kaydedilmesine kadar olan tüm işlemleri gerçekleştirir. Kullanılan çözüm yöntemine göre üç farklı metodu vardır. Sistem uygun olan fonksiyonu seçerek çalıştıracaktır. Aralarındaki fark, uygulanan çözüm yöntemi gereği kullanılması gereken ek işlemlerdir.

KanalBaslat; Yeni bir işlem kanalının oluşturularak başlatılmasını sağlar.

KanalIslem; Popülasyondan kromozom geldiği sürece bu kromozomları alır (Populasyon.KromozomAl) ve işler (KromozomOyna). Populasyondan yeni kromozom gelmemesi neslin bittiğini gösterir. Bu durumda fonksiyon, yani kanal sona erer. Yeni nesilde yeni kanalla işlemler devam eder.

GenelPuan; Kromozom puanının hesaplar.

Puan ağırlığı sıfırın üzerinde olan tüm durumlar için aşağıdaki formül kullanılır.

$$Puan = \frac{\sum \left(Durum.Puan.Ağırlık \times \frac{1 - |Durum.HedefDeğer - Durum.Değer|}{Durum.Değer.Max - Durum.Değer.Min} \right)}{\sum Durum.Puan.Ağırlık}$$

ModelHazırla; Tüm durum değerlerinin ilk değerlerini hazırlar.

Tamamla; Kromozom puan bilgilerinin kromozom üzerinde kaydı yapılır.

2.4. VARSAYILAN DEĞERLER

Sistem içinde bazı varsayılan değerler tutulmaktadır. Bu değerler, sistem içinde kullanılmakta ve kullanıcı için kolaylık olması için tanımlanmıştır. İhtiyaç halinde bu değerler değiştirilebilir.

Çizelge 2.1. Çözüm Kriteri

Kriter	Varsayılan Değeri	Açıklama
Çaprazlama Fonk No	2	Çaprazlama Yöntemi
Kanal Sayısı	16	Aynı anda çalışacak çözücü sayısı
Simülasyon Suresi	1000	Birim Zaman
Nesil Sayısı	50	Üstün ve mutasyon nesilleri hariç
Popülasyon Sayısı	100	Her nesildeki kromozom sayısı
Arzulanan Bağlı Hassasiyet	0,025	Gerekli iterasyon sayısının hesaplanmasında kullanılır. Sonucun arzulanan aralığının ortalamadan hangi oranda uzaklaşabileceği değer.
Hedef Sapma	0%	Kabul edilebilir sapma
Kabul Edilebilir Puan Eşiği	-0,02	Anlık puanı bu değerden düşük olan olaylar çaprazlamada yok edilir.
Min Iterasyon	10	Uygulanacak asgari iterasyon sayısı
Max Iterasyon	100	Gerekli iterasyon sayısı daha büyük olsa dahi uygulanacak azami iterasyon sayısı

Çizelge 2.2. Model Kriteri

Varsayılan Değer	Değeri	Açıklama
Azami Olay Sayısı	100	Aynı olay en fazla kaç kez tekrarlanabilir
Azami Toplam Olay Sayısı	500	
Azami Bağlı Değişim Oranı	10000%	Bağlı durum değişkenlerinin birbirini azami etkileme oranı
Asgari Olay Aralık Süresi	0	Aynı olayın tekrarlayabilmesi için gereken asgari süre
Azami Olay Aralık Süresi	1000	Aynı olayın asgari tekrarlanması gereken aralık
Asgari Olay Suresi	1	Birim zaman
Azami Olay Suresi	10	Birim zaman
Azami Etkilenen Parametre Sayısı	10	Etkilenen nesne olarak eklenebilecek azami nesne sayısı
Azami İlgili Parametre Sayısı	10	İlgili nesne olarak eklenebilecek azami nesne sayısı
En İyi Sonuç	10	Listelenecek en iyi çözüm sayısı. Eğer kromozom değişimi olacaksa bu en az popülasyon sayısı olur

Model kriterleri, bir olay veya durum değişkeni tanımlandığında ilgili özelliklerine varsayılan olarak yukarıdaki değerler verilir, gerektiğinde her olay veya durum için ayrı ayrı değiştirilebilir.

Çizelge 2.3. Varsayılan Değerler

Varsayılan Değer	Değeri	Açıklama
Varsayılan değer	0	Yeni bir durum değişkeni eklendiğinde varsayılan bir değer atanır
Varsayılan min	0	Alt Sınır
Varsayılan max	100000	Üst Sınır

Durum değişkenleri tanımlandığında bu değer ve sınırlamalar ile oluşturulur, her durum için ayrı ayrı değiştirilebilir.

Çizelge 2.4. Mutasyon

Varsayılan Değer	Değeri	Açıklama
Mutasyon Nesli	Evet	Çözümün tamamlanmasından sonra en iyi kromozomların mutasyona uğratarak yeni bir nesil oluşturulması
Atıl Genleri Temizle	Evet	Senaryoya hiçbir etkisi olmayan olayların silinmesi
Atıl Genleri Temizle Nesil Aralığı	17	
Mutasyon	Evet	Mutasyonun uygulanıp uygulanmayacağı
Mutasyon Aralığı	5	Mutasyon işleminin kaç nesilde bir yapılacağı
Mutasyon Olasılığı	0,75%	
Mutasyon Oranı	0,85%	
Zaman Kaydırması	Evet	Uygulanabilecek mutasyon çeşitleri
Gen Ekle	Evet	
Gen Sil	Evet	
Gen Değiştir	Evet	
Gen Değiştir Olay	Evet	
Gen Değiştir Parametre	Evet	
Gen Değiştir Zaman	Evet	

Çizelge 2.5. Kromozom Değişimi

Varsayılan Değer	Değeri	Açıklama
Üstün Nesil	Evet	Çözümün tamamlanmasından sonra en iyi kromozomların çaprazlanarak yeni bir nesil oluşturulması
Zayıf Kromozom	Evet	Belirlenen yüzde kadar en zayıf kromozomlar yenileri ile değiştirilir.
Değiştirilecek Popülasyon Oranı	10%	
Zayıf Kromozom Nesil Aralığı	10	

ÜÇÜNCÜ BÖLÜM

SENARYONUN SİMÜLASYONU

3.1. SENARYONUN GENEL İŞLEYİŞİ

Senaryonun başlangıcında zaman değişkeni sıfırlanır. Tüm nesneler oluşturulur ve durum değişkenleri başlangıç durumuna getirilir. Sabit olaylar ve planlanmış olaylar zaman doğrusunda yerlerini alırlar. İstatistik bilgileri sıfırlanır.

Bundan sonra zaman değişkeni birer birer artırılarak gösterdiği anda bulunan olaylar çalıştırılır. Her zaman birimi sonunda tüm durumlar ile ilgili işlemler tamamlanır. Tahdit, etki gibi unsurlar durum değerlerine uygulanır.

Tetiklenmeler için belli kontrol aralıkları verilebildiğinden kontrol zamanı geldiğinde belirli şartların oluşup oluşmadığı kontrol edilir ve gerekiyorsa önceden tanımlanmış olaylar tetiklenir.

Zaman değişkeni simülasyon süresine eriştiğinde sistem durur ve toplanan istatistikler kaydedilir.

3.2. OLAYLARIN ÇALIŞMA PRENSİBİ

Olayın aslı eylem fonksiyonunun çalıştırılmasıdır. Ancak eylemin olabilmesi için bir takım şartların gerçekleşmesi gerekmektedir. Olay çalıştırıldığında tüm bu şartlar kontrol edilerek eylem gerçekleştirilir. Eylemin içeriği kullanıcı tarafından yazıldığından olaylar ile ilgili herhangi bir kısıtlama bulunmamaktadır.

3.2.1. Olay Öncesi Kontroller

İlk olarak olayın gerçekleşmesi için bir ihtimal değeri atanmışsa bir rastgele sayı üretilerek olayın gerçekleşme ihtimali ile karşılaştırılır ve sonuç kaydedilir.

Olayın azami tekrar edilebileceği sayı ve olayın bir önceki oluşumu ile arasındaki zaman aralığının olayın oluşabilmesi için asgari zaman aralığını karşılayıp karşılamadığı kontrol ve kaydedilir.

Eğer bu kıstaslar yerine gelmişse varsa kullanıcı tarafından yazılan olay şartı kontrol edilir. Bu şart da gerçekleştiği takdirde eylem gerçekleştirilir. Eylem gerçekleştirilemediği takdirde nedeni kaydedilir.

3.2.2. Olay Şartı

Olayın gerçekleşme şartı kullanıcı tarafından tanımlanmak istendiğinde sonucu olumlu ve olumsuz dönen bir fonksiyon yazılır. Örneğin bir firmanın bir faaliyeti yapabilmesi için öncelikle kaynağının yeterli olup olmadığı kontrol edilmek isteniyorsa aşağıdaki gibi bir kod yazılabilir.

```
bool SartKaynakYeterlimi(OlayGeni Param) // Faaliyet1
{
    return Param.Param[0]["Maliyet"].Deger < Param.Fail["Kaynak"].Deger;
}
```

Burada olaygeni (Param) olarak gelen parametre değerleri kullanılır. Yukarıdaki örnekte ilgili nesne (Param.Param) olarak gelen faaliyetin maliyet durum değişkeni ile Fail olarak gelen firmanın kaynak durum değişkeni karşılaştırılarak bir sonuç döndürülür.

Birden fazla faaliyet varsa

```
bool SartKaynakYeterlimi(OlayGeni Param) // Faaliyet1
{
    double toplam = 0;
    foreach (NesneselVarlik nv in Param.Param)
    {
        toplam += nv["Maliyet"].Deger;
    }
    return toplam < Param.Fail["Kaynak"].Deger;
}
```

Yukarıdaki kod önceki örneğin benzeridir. Sadece birden fazla faaliyetin (Param.Param) geldiği kabul edilmiştir. Bu yüzden önce maliyetler toplanıp sonra karşılaştırılmıştır.

3.2.3. Eylem Fonksiyonu

Kullanıcı tarafından tanımlanan bir fonksiyondur. Nesne durum değişkenlerini etkileyen işlemler burada yazılır. Örneğin bir firmanın kar ve pazar payını artıran bir faaliyet aşağıdaki gibi tanımlanabilir.

```

void Eylem15(OlayGeni Param) // Faaliyet1
{
    int sure = Param.olay.sure;
    Param.oran = 1;
    // Fail.PazarPayi %0.04 oranında artır
    Param.Fail["PazarPayi"].AddDegerOranFark(0.04, Param.oran, sure);

    // Fail.Kar 50 değerinde artır
    Param.Fail["Kar"].AddDegerFark(50, Param.oran, sure);

    // Etkilenen.PazarPayi %0.04 oranında azalt
    foreach (var etki in Param.Meful) // Meful = etkilenen
    {
        etki["PazarPayi"].AddDegerOranFark(-0.04, Param.oran, sure);
    }
    // Tahditler içinde "PazarPayi toplamı %100 olacaktır" şeklinde bir tahdit
    olduğundan değişen Pazar payları sonunda toplamı %100 olacak şekilde oranlar
    dengelenecektir.
}

```

Şart fonksiyonunda olduğu gibi eylem fonksiyonu da parametre olarak bir olaygeni almaktadır. (Olaygeni için bkz. 1.3.3.1 Olaygenleri). Yukarıdaki örnekte eylem fonksiyonunun temsil ettiği faaliyeti gerçekleştiren firmanın pazar payı artmakta, bu faaliyetten etkilenen diğer firmaların pazar payları ise aynı oranda azalmaktadır. Birden çok etkilenen firma olması durumunda toplam pazar payı %100'ün altında bir değer olacaktır. Ancak öncesinde yapılan bir tahdit tanımı ile tüm firmaların pazar payı toplamlarının 1 olması gerektiği belirtildiğinden, sistem bu tür durumları kontrol edecek ve tahditlerin gerçekleşmesi için düzeltme yapacaktır. Bu konu ileride daha ayrıntılı olarak ele alınacaktır.

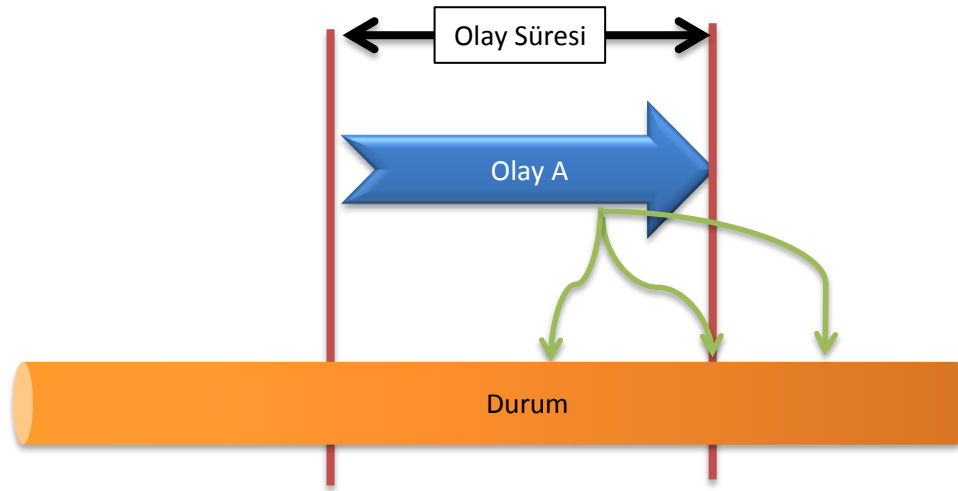
3.2.4. Dönem Sonu İşlemleri

Zaman değişkeninin her ilerletilmesinde durum değişkenleri ile ilgili bazı kontrol ve işlemler yapılmaktadır. Bu işlemler aşağıda sırasıyla listelenmiştir.

1. Atama
2. Etki
3. Direnç
4. Bağlı Durumlar
5. Normale Dönüş
6. Tahditler
7. Yeni Tetiklemeler
8. Puanlama

3.2.4.1. Atama

Durumlara atama işlemi doğrudan yapılabileceği gibi zamanlı değerler ile de yapılabilir. Zamanlı değer, bir değer bir duruma hangi zamanda atanacağını tutan bir yapıdır. Bir olayın eylem fonksiyonu içinde bir durum değişkeni değiştiriliyorsa bu işlemin duruma ne zaman etki edeceği belirtilmelidir. Bu olayın süresi kadar zaman birimi sonra olabileceği gibi olayın devam ettiği bir an da olabilir (Bkz. Şekil 3.1).



Şekil 3.1. Olay Atama İşlemleri

Eylem fonksiyonu içinde farklı işlem türleriyle atama yapılabilmektedir. Aşağıda zamanlı değerler için yapılabilen atama çeşitleri gösterilmiştir.

Durum = Değer

Durum = Durum + Değer

Durum = Durum * Değer

Durum = Durum * (1 + %Değer)

Durum1 → Durum2 (%Değer) (Durum1, Durum2'ye %değer oranında yaklaşıyor)

Aynı anda birden fazla olay aynı duruma bir atama yapmaya çalışıyor olabilir. Bu tezde durum değerleri sayı olarak düşünülmemiştir. Sayısal yapılacak işlemlerde durum yerine sayı tanımlanabilir. Durumlar daha çok bir duygunun durumu veya bir varlığın iki uç nokta arasındaki durumu düşünülmüştür. Bu yapı bu yönüyle bulanık mantık ile

bir benzerlik gösterse de durum yapısının farklı parametreleri vardır (tahditler, direnç, vb.). Bu nedenlerle aynı anda etki eden olayların sonuçlarının ağırlıklı ortalaması alınmaktadır. Örneğin A olayı Ahmet beyi sinirlendiriyor, B olayı ise sakinleştiriyorsa ve bu iki olayın etkileri aynı anda ortaya çıkmışsa, etkilerin ağırlığına göre ortalaması alınabilir. Eğer iki olayın farklı bir etkileşimi oluyorsa bunlar olay içinde ele alınabilir.

Sonuç olarak olay içinde yapılan işlemlerde durumun olay başlangıcındaki değeri esas alınır; atama işlemi olay sonunda, olayın devamında veya olayın tamamlanmasından belli bir süre sonra duruma dâhil edilir; aynı anda duruma dâhil edilecek değerlerin ortalaması alınır.

3.2.4.2. Etki

Bazı durumlarda bir olayın etkisi belli bir süre devam eder. Bu etki sabit olabileceği gibi zamana bağlı değişken de olabilir. Bu etkinin değeri ve ne kadar devam edeceği kullanıcı tarafından belirlenir.

Etki özelliğinin kullanılabilmesi için durumlara örnek olarak bir topluluğun düşüncesinin belli bir yöne kayma durumu veya bir yaralanmadan kaynaklanan ağrı durumu verilebilir.

Etkinin zamana bağlı değişimi için bir etki fonksiyonu tanımlanır. Bu fonksiyonu parametre olarak etkinin ömrü, etkinin değeri ve olaydan sonra geçen süreyi almaktadır. Aşağıda bu fonksiyon için bir örnek verilmiştir. Burada etkinin ömrü zamanla azalmaktadır. Aynı fonksiyon farklı etkiler için kullanılabilir.

```
double EtkifonkA(double eOmr, double value, double t)
{
    return value * (eOmr - t) / eOmr;
}
```

Etki değeri atamalardan sonra bulunan ortalama yeni değer üzerine uygulanır. Sonuç ise direnç ile birlikte hesaplanır. Etki ömrü dolan etkiler, etki listesinden çıkarılarak simülasyonun devamında dikkate alınmaz.

3.2.4.3. Direnç

Direnç, bazı durumların değişime karşı bir dirence sahip olduklarında kullanılır. Örneğin insanların duyguları için bir direnç düşünebiliriz. Kimi insan başkalarına göre daha kolay sinirlenir. Eğer sizin bir yapıya sahipse, modelde o insan için sinirlenme durum değişkenine daha yüksek bir direnç değeri verilebilir. Bu durumda aynı etkiyle farklı kişilerde farklı tepkiler gözlemlenebilir. Somut bir örnek vermek gerekirse, metallerin farklı çekme dirençleri vardır. Aynı kuvvet uygulansa da farklı oranlarda deforme olurlar.

Atama, etki ve direnç aşağıdaki şekilde durum değişkenine eklenmektedir;

$$Yeni\ Değer = \left[\left(\left(n: Atama\ Sayısı \begin{cases} n > 0 \rightarrow \sum \frac{Atama\ Değeri}{n} \\ n = 0 \rightarrow Eski\ Değer \end{cases} \right) + \sum etki \right) \times (1 - Direnç) \right] + [Eski\ Değer \times Direnç]$$

Eğer yeni atama yapılmışsa, yapılan atamaların ortalaması, atama yapılmamışsa eski değer alınır. İçinde bulunulan zaman birimi içinde oluşan etkiler toplanarak hesaplanan yeni değere eklenir. Direnç 0 ile 1 arasında bir değerdir. Direnç 1'se o durumun değeri değişmez, direnç 0'sa her zaman hesaplanan yeni değer, durumun yeni değeri olur. Diğer hallerde ise yukarıdaki gibi hesaplanır.

3.2.4.4. Bağlı Durumlar

Değeri değiştiğinde, başka bir durumun değerini de etkileyen durum değişkenleridir. Kullanılabileceği yerlere örnek vermek gerekirse; İnsan ne kadar mutlu ise yaptığı işte o kadar dikkatli ve başarılıdır önermesini doğru kabul edersek dikkat ve başarılı olma ihtimali durum değişkenlerini mutluluk durum değişkeninin bağlı durumudur diyebiliriz. Bu durum değişkenleri doğru orantılı olarak değişecektir. Eğer bağıllık çift yönlü ise bu tanımın her iki yönde de yapılması gerekir.

Durum değişkeninin değeri değiştiğinde bağlı durumun değerinin bundan ne kadar etkileneceğini belirlemek için bir bağıllık oranı tanımlanmıştır. Bağlı durum, bu oran kadar etkilenebilir. Eğer bağıllık ters orantılı ise bu değere negatif bir sayı verilebilir. Bağıllık oranının azami değeri model kriterlerinde sınırlanabilir. Varsayılan sınırlama olarak 100 kat belirlenmiştir.

Bağıl durumlar limitli ve limitsiz olmak üzere iki çeşittirler. Limitsiz bağlı durumlarda bağlı durum değişkeni doğrudan oran kadar etkilenir. Sonraki zaman birimine değere eklenen sayı olarak geçer.

Limitli bağlı durumlarda ise etkileyen ve etkilenen durum değişkenlerinin üst ve alt değerleri dikkate alınır ve buna göre değişim gerçekleşir. Bu işlem aşağıdaki formülde gösterilmiştir.

$$Atanan\ Değişim = (YeniDeğer - EskiDeğer) \times \frac{max_{BağılDurum} - min_{BağılDurum}}{max_{Durum} - min_{Durum}}$$

Atanan değişim değeri, bağıllık oranı ile birlikte bir sonraki zaman biriminde işlenmek üzere sisteme kaydedilir.

3.2.4.5. Normale Dönüş

Bazen bir durum değişkeni, başka bir değer veya duruma meyleder. Burada durum değişkeninin normal şartlarda olması gereken bir değeri olduğu kabul edilir ve durum değişkeni sürekli bu değere ulaşmak ister.

Örneğin insanın ruh halini modellemek istersek; motivasyon, mutluluk, sinirlenme, moral, vb. değerleri durum değişkeni olarak düşünürsek, bu değerlerin normal şartlarda bulundukları değerleri vardır. Bu değerler kişiden kişiye göre değişebilir. Bir takım etkilere maruz kaldıktan sonra bu değerler değişse de bir süre sonra tekrar normal konumlarına gelirler. Eğer kalıcı değişiklik oluyorsa veya normal değeri değişkense, bu değer için başka bir durum değişkeni gösterilebilir.

Durum değişkeninin her zaman biriminde normal değerlerine ne kadar yaklaşacağını belirleyen bir değer vardır. Bu değer oran veya sayısal olabilir. Normale yaklaşma değeri oransal olduğunda aşağıdaki şekilde hesaplanır;

YD: Yeni Değer

ED: Eski Değer

ND: Normal Değeri

NY: Normale Yaklaşma Oranı / Değeri

$$YD = (ND \times NY) + (ED \times (1 - NY))$$

Normale yaklaşma değeri sayısal ise;

$$YD = YD \pm NY$$

Eğer durum değişkeninin değeri ile normal değer arasındaki fark normale yaklaşma değerinden azsa yeni değer normal değeri olur.

3.2.4.6. Tahditler

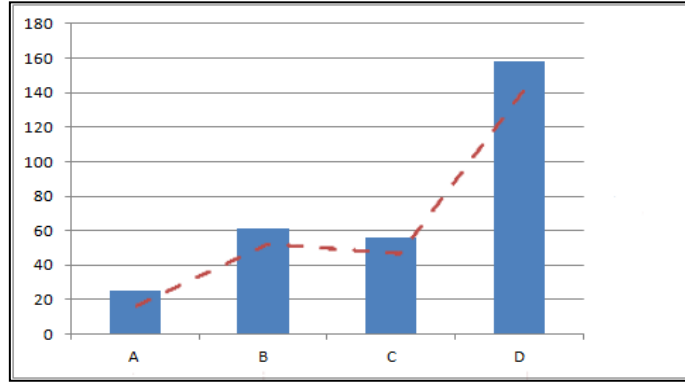
Durum değişkenlerinin değerleri için azami ve asgari değerler belirlenebildiği gibi birden çok durum değişkeni arasında bir tahdit ilişkisi kurulabilmektedir.

Şimdiye kadar anlatılan dönem sonu işlemleri her durum değişkeni için tek tek ele alınmaktaydı. Bundan sonraki dönem sonu işlemleri toplu olarak ele alınmaktadır.

Tahdit yapısı $\sum sol \stackrel{\text{def}}{=} \sum sağ$ şeklindedir. Sol ve sağ tarafa sayı veya durum değişkeni eklenebilir. Karşılaştırma operatörü olarak $=, \neq, <, >, \leq$ veya $\geq$ atanabilir. Karşılaştırma esnasında durum değişkenlerinin ihmal değerleri dikkate alınır. Yani ihmal değeri 0,05 ise $1 > 1,04$ karşılaştırmasının sonucu doğru döner.

Tahdit kontrolü bu tezde korunması gereken bir dengeyi ifade etmektedir. Kontrol sonucunda eğer eşitlik dışına çıkmışsa veya eşitsizliğin yönü bozulmuşsa tahdidin karşılanması için bir düzeltme yapılır.

Örneğin $A + B + C + D \leq 240$ tahdidi varsa ve toplamı 300 ediyorsa bu dört durum değişkeninin değerleri azaltılır. Denklemin her iki tarafında da durum değişkenleri olabilir. Durum değişkenlerinin farklı direnç değerleri varsa düzeltme esnasında dikkate alınır. Eğer her iki tarafta da direnci 1'den farklı durum değişkenleri mevcutsa eşitliğin veya eşitsizliğin her iki tarafının hangi değere göre düzeltileceği aşağıdaki şekilde hesaplanmıştır.



Şekil 3.2. Tahdit Kontrolü

$$Denge\ Noktası = \frac{\sum SolDeğer \times \sum SolDirenç + \sum SağDeğer \times \sum SağDirenç}{\sum SolDirenç + \sum SağDirenç}$$

Denklemin her iki yanında da durum değerleri kendi içinde direnç değerlerine göre değişirler.

3.2.4.7. Tetiklemeler

Tetiklemeler tetikleme fonksiyonları içinde yapılmaktadır. Bu fonksiyon, kullanıcı tarafından yazılır ve içinde istenen şartlar kontrol edilerek istenen olay tetiklenir. Bu olay istatistiklerde tetiklenen olay olarak görülür.

Dönem sonu işlemlerinde ise bu fonksiyonun çalıştırılması gerçekleştirilir. Yine bu fonksiyonun çağrılması iki şarta bağlıdır: Birinci şart, bu tetikleme için çalıştırılacak en erken zaman ve en geç zamanın belirlenebilmesidir. Belirlenen bu zaman aralığı dışında bu tetikleme yapılmaz. Diğer şart ise tetikleme için bir kontrol periyodunun belirlenebilmesidir. Tetikleme sadece bu periyotlarda yapılır. Eğer her an kontrol edilmesi isteniyorsa kontrol periyodu 1 zaman birimi olarak verilebilir.

3.2.4.8. Son İşlemler

Model içinde her zaman sonunda hesaplanması istenen bazı işlemler olabilir. Bu tür işlemler burada bir eylem tanımı içinde yapılabilmektedir.

Örneğin firmaların her birim zaman sonundaki toplam kârı görülmek isteniyorsa burada bu hesaplanabilir.

3.2.4.9. Puanlama

Puanlama sadece amaç fonksiyonu içinde yer alan durum değişkenleri için yapılır. Amaç fonksiyonuna durum değişkenleri bir ağırlık değeri ile birlikte eklenir. Böylece amaç fonksiyonu $ax_1 + bx_2 + cx_3 + \dots$ şeklinde olur. Her dönem sonunda anlık puanlar hesaplanır. En yüksek anlık puan değeri senaryonun puanı olarak kabul edilir. Çünkü senaryo bu puan değerinin olduğu zamanda kesilebilir. Bu nedenle tüm senaryoların en yüksek anlık puan değerinde son bulduğu kabul edilir.

Puanlama sistemi Ağırlık tabanlı genetik algoritmanın (Weight-based genetic algorithm) (Ghosh, 2004) puanlama sistemini andırsa da hedef değere yönelik ve sisteme olan katkısını gösterecek şekilde düzenlenmiştir.

Puanlama aşağıdaki formülde görüldüğü şekilde yapılmaktadır. Burada her durum değişkeninin kendi mümkün alanı içinde hedef değere olan yakınlığı anlık puan olarak alınmıştır.

$$Puan_t = \frac{\sum_i \left(Ağırlık_i \times \left(1 - \left| \frac{Değer_i - HedefDeğer_i}{Max_i - Min_i} \right| \right) \right)}{\sum Ağırlık_i} - Puan_{t-1}$$

Puanlar kaydedilirken bir önceki anın puanından çıkarılarak kaydedilir. Böylece bu zaman biriminin sisteme olan katkısı veya zararının puan değeri kaydedilmiş olur.

Herhangi bir t anında hiçbir olay olmasa da dönem sonu işlemleri mutlaka uygulanır. Bu esnada önceden kalan etkiler, normale dönüş işlemleri, vb. olabilir. Dolayısı ile boş olan bir t anında dahi anlık puan değeri değişebilir.

DÖRDÜNCÜ BÖLÜM

GENETİK ALGORİTMANIN UYGULANIŞI

Şimdiye kadar sınıf bazında anlatılan işlemler aşağıda bir bütün olarak ele alınarak tasarlanan genetik algoritmanın işleyişi sırasıyla anlatılacaktır.

4.1. İLK KROMOZOMLARIN OLUŞTURULMASI

Yeni kromozomlar planlanabilir olaylardan uygun sayıda oluşturularak rastgele üretilmektedirler. Planlanabilir olayın asgari ve azami tekrar sayıları göz önüne alınarak kaç kez tekrar edileceği rastgele olarak belirlenir ve olaylar teker teker eklenir.

Her bir olay senaryoda bir olay geni ile temsil edilir. Olay geni oluşturulur ve ilgili olay bu olay genine bağlanır. Müteakiben olay geninin parametreleri belirlenir. Eğer olay bir fail alabiliyorsa olayın mümkün failleri içinden bir fail seçilir.

Etkilenen ve ilgili olan nesneler birden fazla olabilmektedir. Aynı zamanda bu parametrelerin azami sayıları da sınırlandırılmıştır. Öncelikler kaç etkilenen nesnesel varlık ve kaç ilgili nesnesel varlık olacağı belirlenir ve iki liste oluşturulur. Bu listelerin birinde ilgili olayın etkilenen olarak kabul edebileceği nesnesel varlıklar, diğerinde ilgili olarak kabul edebileceği nesnesel varlıklar vardır. Bu listelerde eğer fail varsa listeden çıkarılır. Olay genine parametre olarak etkilenen ve ilgili nesnelere bu iki listeden belirlenen sayıda tekrar olmayacak şekilde seçim yapılır. Önce etkilenen nesnesel varlıklar seçilir ve seçilen nesneler ilgili listesinden çıkarılır. Bu şekilde bir nesne fail, etkilenen ve ilgili parametrelerinde yalnız bir kez olabilir.

Son olarak olay geninin zamanı belirlenir. Bu o olay için izin verilen ilk ve son zamanlar arasında rastgele seçilir ve senaryonun zaman doğrusuna eklenir. Olayın gerçekleşme zamanlarının aralık kısıtlamaları senaryonun çalışması sırasında ele alındığından burada dikkate alınmaz.

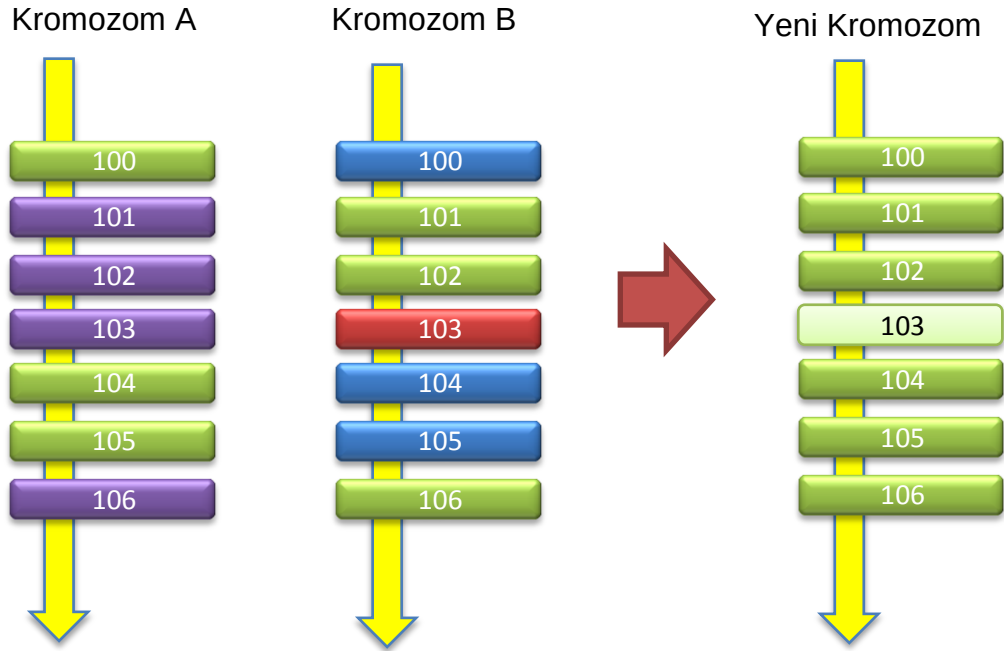
4.2. ÇAPRAZLAMA

Çaprazlama için Tek noktadan bölünme, Grup olarak bölünme, Puanlama, Analiz ve tam analiz olmak üzere beş farklı yöntem kullanılmış, ancak bunlar içerisinde

puanlama yöntemi bir modelin geliştirilmesinde, tam analiz yöntemi en iyi sonucun bulunmasında tercih edilmiştir. Burada puan karşılaştırması çaprazlama ve analiz yöntemleri açıklanacaktır. Diğer çaprazlama yöntemleri için 0. Üreteç başlığına bakılabilir.

4.2.1. Puanlama ile Çaprazlama

Puanlama ile yapılan çaprazlamada simülasyon esnasında senaryonun her zaman sonunda kromozomun anlık puan değeri hesaplanmaktadır. Anlık puanlar, o zaman biriminde gerçekleşen olayların hedef değerlere ne kadar yaklaştığını göstermektedir. Çaprazlanacak iki kromozomun aynı zamana denk gelen olaylar grubunun anlık puan değerleri karşılaştırılır, büyük olan alınır. Eğer yüksek olan puan değeri kabul edilebilir puan eşliğinin üzerindeyse yeni kromozoma kaydedilir.



Şekil 4.1. Puanlama Yöntemi

Şekil 4.1’de puanlama ile çaprazlama için bir örnek gösterilmiştir. Şekilde anlık puan değeri daha yüksek olan olay grupları yeşil olarak gösterilmiştir. Seçilen olaylar yeni kromozoma grup olarak kopyalanırlar. Yani o zaman biriminde kaç olay oluşmuşsa anlık puan değeri bu olayların tamamına aittir ve birlikte işlem görürler.

Eğer yüksek olan anlık puan değeri, kabul edilebilir puan eşiğini geçememişse, örnekte kırmızı olarak gösterilmiştir, çaprazlanan yeni kromozomda bir mutasyon uygulanarak bu zamana olay kopyalanmaz. Şekilde görüldüğü gibi burada boş bir olay grubu oluşturulur.

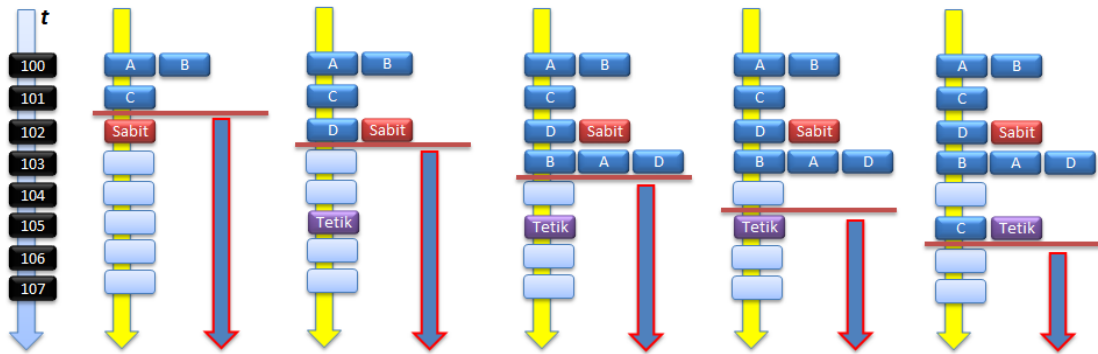
Tezde puanlama ile çaprazlama yöntemi sonuç ve işlem süresi açılarından model yapısının ispatında tercih edilmiştir.

4.2.2. Analiz ile Çaprazlama

Analiz yönteminde çaprazlama işlemi, puanlama ile çaprazlama yönteminin aynısıdır. Fark puanlamanın yapılaş şeklinde ortaya çıkar. Bu yöntemde her olay grubunun gerçek puan değerinin hesaplanması için simülasyon her t anı için kısmen tekrar koşturulur.

Anlık puanlar hesaplanırken aşağıdaki adımlar takip edilmektedir;

1. Sabit olaylar ve t anına kadar olaylar koşulacak senaryoya kopyalanır.
2. Senaryo simülasyon sonuna kadar oynatılır. Bu esnada senaryoya tetiklenen olaylar eklenebilir. Senaryo gereken iterasyon sayısı kadar tekrarlanır.
3. Senaryo sonunda elde edilen puan t anının anlık puanı olarak kaydedilir.
4. Senaryo sonuna gelinmişse işlem bitirilir, aksi halde t anı bir ilerletilir ve Adım 1'e dönülür.



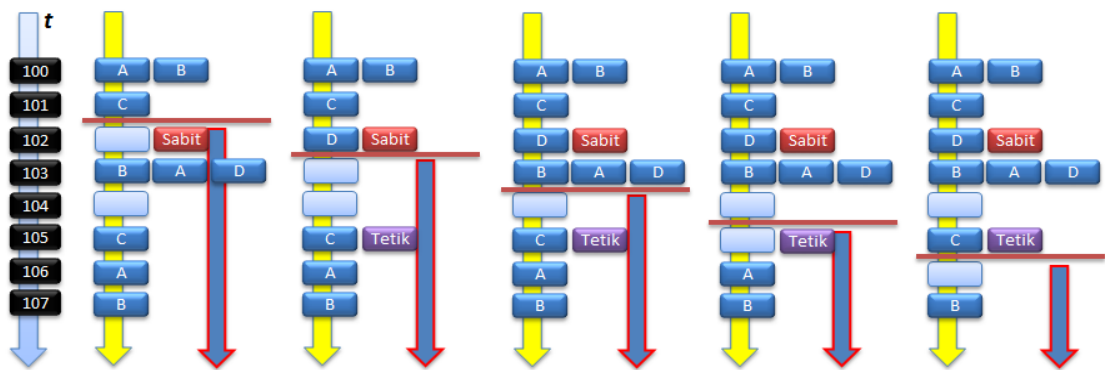
Şekil 4.2. Analiz Puanlama

Şekil 4.2’de gösterilen örnekte planlanabilir olaylar mavi, sabit olaylar kırmızı ve tetiklenen olaylar mor gösterilmiştir. $t = 105$ ’te tetiklenen olayının $t = 102$ ’deki D olayının etkisiyle tetiklendiği kabul edilmiştir.

Şekilde görüldüğü gibi incelenen anın sonrasında sadece sabit veya tetiklenen olaylar bulunabilir. Bu şekilde Etki, normale dönüş, bağlı işlemler, tetiklenmeler, vs. işlenmiş olacağından t anındaki olayların senaryo sonundaki gerçek etkileri görülmüş olacaktır. Puanlama ile çaprazlama yönteminde bazı olayların etkileri sonraki olayların puanına karışabilmekteydi.

Değişik senaryolarla yapılan denemelerde Analiz yönteminin bir eksiği tespit edilmiştir. Bir olayın sonraki faaliyetlere dolaylı bir etkisi varsa bu görülemiyordu. Örneğin şirketiniz bir yatırım yapıyor. Bu yatırımın bir maliyeti vardır, ancak bundan sonra yapılacak üretimlerde maliyet belli bir yüzde azalmaktadır. Bu eylemin faydası daha sonraki olaylarda ortaya çıktığından bu analiz yönteminde anlaşılamayacaktır. Bu nedenle bir genin gerçek etkisini anlamak için;

1. Senaryo bir kez tüm olayları ile koşularak senaryo puanı alınır.
2. Sonra her birim zaman için o birim zamanda bulunan olay olmadan çalıştırılır.
3. t anının anlık puanı olarak her an için hesaplanan puanın senaryo puanından olan farkı alınır.



Şekil 4.3. Tam Analiz Puanlama

Çaprazlama için en uygun puanlama tam analiz yöntemidir. Ancak daha hızlı sonuca ulaşmak maksadıyla, problemin türüne ve olayların özelliğine göre puanlama

veya analiz yöntemleri de kullanılabilir. Eğer modelde rassal sayı hiç kullanılmamışsa bu durumda çözüm sürecinde tek iterasyon kullanmak uygun olan seçimdir.

4.3. MUTASYON

Modelde sekiz farklı mutasyon uygulanmıştır.

4.3.1. Çaprazlama İçinde Uygulanan Mutasyon

Puanlama ve analiz çaprazlama yöntemlerinde kabul edilebilir puan eşiğinin altında olan olay gruplarının silinmesi olan mutasyondur.

4.3.2. Atıl Genlerin Temizlenmesi

Atıl genleri temizlenmesi, her hangi bir durum değişkenini etkilemeyen olayların silinmesi işlemidir. Atıl genlerin hangi aralıkla temizleneceği belirlenebilmektedir. Eğer içinde bulunulan nesilde bu mutasyon uygulanacaksa, her birim zamanın ilerlemesinden önce tüm durum değişkenlerinin bir kopyası kaydedilir. Bir sonraki birim zamanda durum değişkenlerinin değerleri ve etki ile daha sonra gelecek değişiklikleri kontrol edilerek senaryonun bu anındaki olayların sistemde herhangi bir değişiklik yapıp yapmadığı kontrol edilir. Eğer durum değişkenlerinde hiçbir değişiklik olmamışsa bu olay grubu atıl olarak işaretlenir.

Böyle bir durumun oluşması için bu andaki olayların gerekli şartların oluşmadığı için meydana gelememesi, olayın eylem fonksiyonu içinde durum değişkenlerine herhangi bir müdahale yapılmaması veya kendi değerlerinin atanması ile olabilir.

4.3.3. Zaman Kaydırması

İzin verilen mutasyonlar içinde seçildiği takdirde senaryoda zaman öne doğru kaydırılır. Bu mutasyonda senaryonun en fazla ilk 10 kadar birim zamanı olayları ile birlikte senaryonun sonuna taşınır. Müteakiben olay genlerinin zaman bilgileri güncellenir.

4.3.4. Gen İşlemleri

Mutasyon işlemlerinin kaç nesilde tekrarlanacağı mutasyon aralığı ile belirlenebilmektedir. Mutasyonun uygulanacağı nesilde mutasyon olasılığı kadar kromozoma mutasyon oranı kadar mutasyon uygulanır.

Uygulanabilecek mutasyonlar ise yeni bir olay eklenmesi, var olan bir olayın silinmesi veya bir olayın parametrelerinin değiştirilmesi olabilir.

Yeni bir olayın eklenirken gerekli parametreler yeni bir kromozomun oluşturulmasında olduğu gibi oluşturulur.

Gen içinde değişiklik ise üç şekilde olabiliyor;

1. Olayı değiştirilmesi, dolayısıyla burada olay parametreleri de değiştirilmektedir.
2. Olay parametrelerinin değiştirilmesi
3. Olayın başka bir zamana kaydırılması işlemleridir.

4.3.5. Akıllı Mutasyon

Rastgele mutasyonların istenen verimi sağlamaması nedeniyle akıllı bir mutasyona ihtiyaç duyulmuştur.

Akıllı mutasyon seçildiğinde, çaprazlama sonucunda senaryodan kopyalanan genlerin anlık puanları da mutasyon işlemi sonunda silinmek üzere yeni kromozoma kopyalanmaktadır. Böylece yeni kromozomda hangi genlerin iyi veya kötü olduğu bilinebilmektedir.

Genler anlık puanlarına göre sıralanırlar ve mutasyon oranı kadar en kötü genin yerine en iyi genler kopyalanır. Bu şekilde az etkili olayların yerine sonuca daha çok yaklaşan olaylar planlanmış olur. Yapılan incelemelerde akıllı mutasyon daha başarılı sonuçlar vermiştir.

4.3.6. Mutasyon Nesli

Mutasyon nesli, optimizasyon işlemi sonunda tüm nesillerdeki en iyi kromozomlarla sadece mutasyon geçirmiş yeni bir nesil oluşturulmasıdır. Burada bulunan çözümlerde mutasyonla daha iyisi aranmaktadır. Seçimlidir. Yapılan denemelerde en iyi sonuçlar arasına az sayıda mutasyon nesline ait kromozom girmiştir.

4.4. KROMOZOM DEĞİŞİMİ

Sonuca daha hızlı yaklaşmak veya daha doğru bir sonuç elde edebilmek için kromozomların değiştirilmesidir.

4.4.1. Üstün Nesil

Eğer üstün nesil seçilmişse, tüm nesillerdeki en iyi kromozomlar çaprazlanarak yeni bir nesil oluşturulur. Yapılan denemeler sonucunda en iyi 100 çözüm içinde ortalama 34'ü üstün nesle aittir.

4.4.2. Zayıf Kromozomların Değiştirilmesi

Zayıf kromozomların değiştirilmesiyle o nesildeki kötü kromozomların yerine yeni kromozomların üretilerek eklenmesi, böylece genetik algoritma içinde daha geniş bir arama yapılabilmesi ve kötü kromozomların soyunun devam etmemesi amaçlanmıştır.

Zayıf kromozomların kaç nesilde bir değiştirileceği ve neslin ne kadarı zayıf olarak kabul edileceği seçilebilmektedir.

Yapılan denemeler sonucunda en kötü kromozomların yenileriyle değiştirilmesi ile en iyi sonuç daha erken bulunsa da daha iyi bir sonuç bulunmasını veya aynı değerde bir sonuç bulunmasını garanti edememektedir.

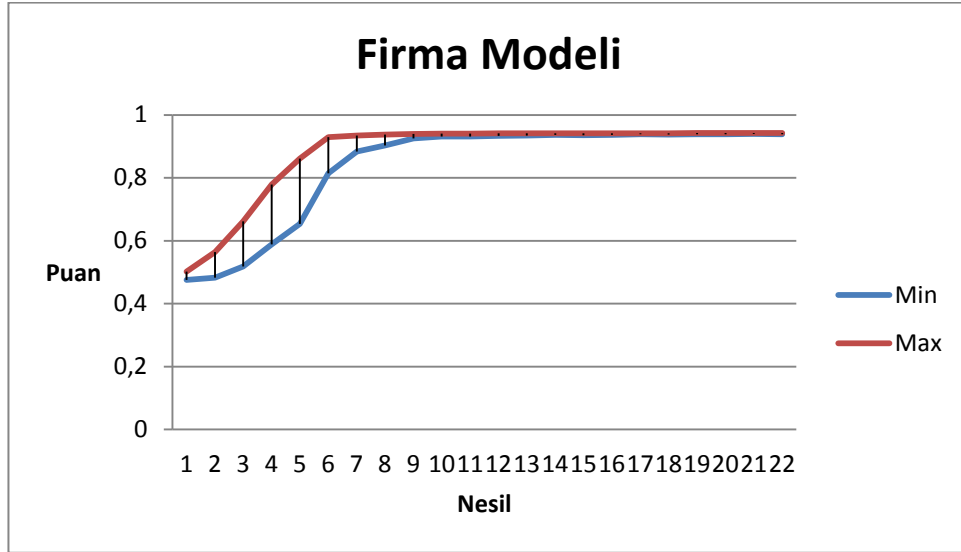
BEŞİNCİ BÖLÜM

ÇÖZÜMLERİN ANALİZİ

5.1. NESİLERDE KROMOZOM PUANLARININ DEĞİŞİMİ

Hazırlanan modelin çözümlenmesi esnasında kullanıcıya gösterilmek üzere çeşitli veriler kaydedilmektedir. Çözüm esnasında hangi kromozomun alındığı, kaçınıcı iterasyonun çözüldüğü, kromozomun hangi puanı aldığı gibi bilgiler raporlanmaktadır. İşlem sonunda en iyi kromozomlar puanları ile birlikte sıralanmaktadır.

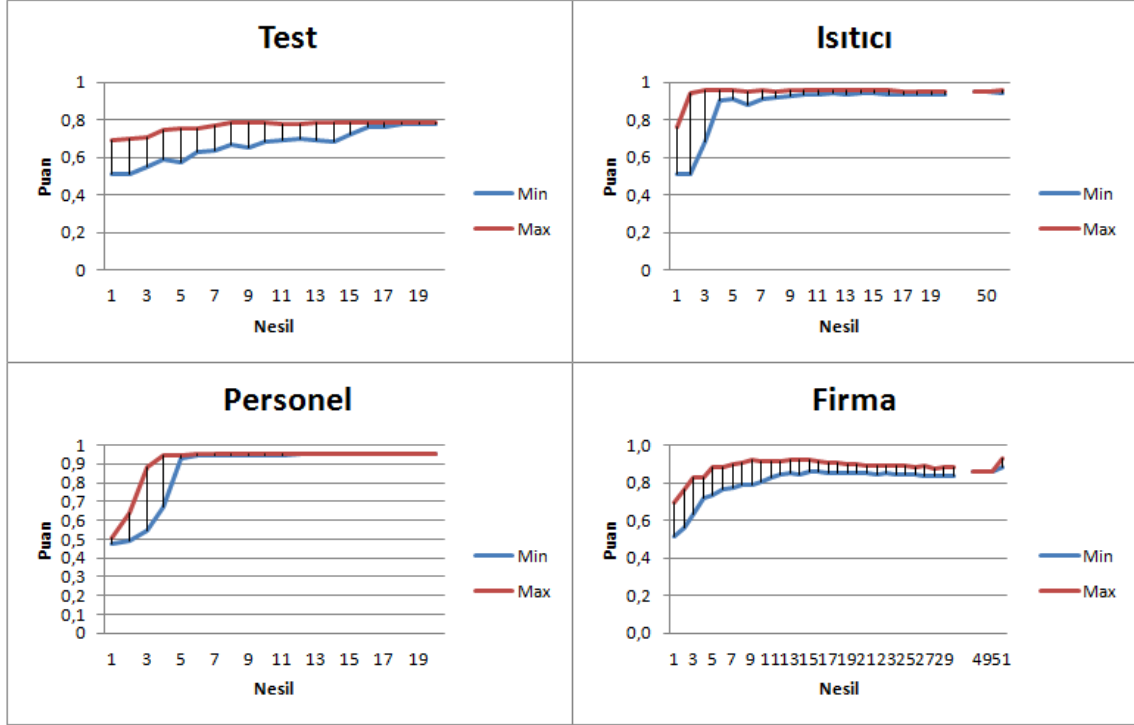
Sonuçları incelediğimizde kromozom puanlarının nesiller ilerledikçe yükseldiğini görmekteyiz. Bir firma modelinde ilk nesilde 0,47 – 0,50 aralığında olan kromozom puanları 3. nesilde 0,51–0,66 ve 5. nesilde 0,65 - 0,86 oluyor. Kromozomların nesillerdeki asgari ve azami puanlarını grafiğe döktüğümüz zaman aşağıdaki çizelge oluşmaktadır.



Şekil 5.1. Firma modelindeki kromozom puanlarının nesillere göre azami ve asgari değerleri

Bu örnekte dokuzuncu nesilden itibaren puanlarda gittikçe daha küçük değişimler olmakta ve bir dengeye ulaşmaktadır. Puanlama ile çaprazlama yöntemi kullanıldığında daha erken nesillerde iyi sonuçlar alınabilmektedir. Analiz yöntemi ile yapılan çaprazlamalarda sonuçların daha erken dengeye ulaşması beklenirken 15 ilâ 20.

nesillerde denge oluşmaktadır. Şekil 5.2’de gösterilen farklı model çözümlerden puanlama analiz yöntemiyle çaprazlanmıştır. Şekil 5.2’de gösterilen grafiklerde 51. nesil üstün nesli göstermektedir.



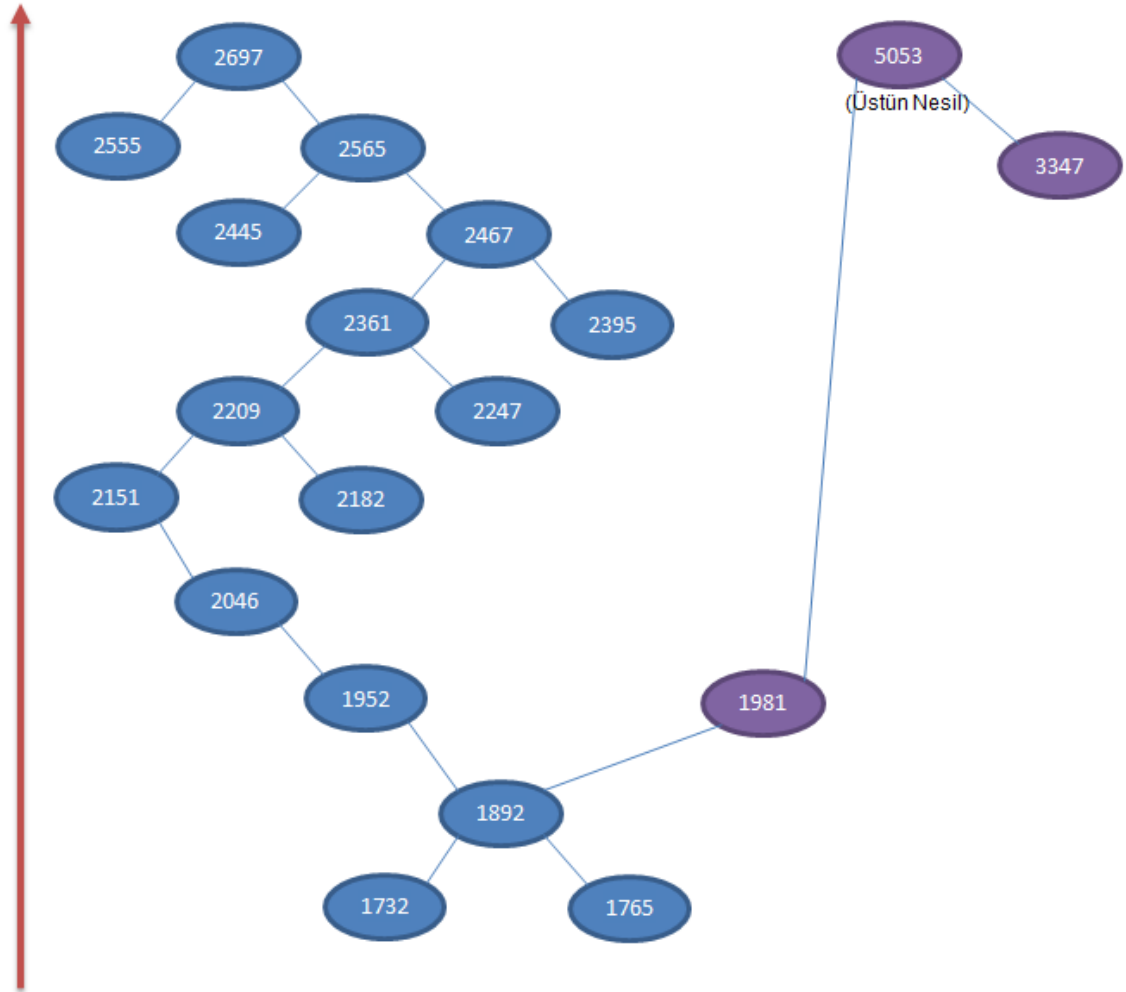
Şekil 5.2. Diğer modellere ait kromozom puanlarının nesillere göre azami ve asgari değerleri

Grafiklerde görüldüğü üzere 5 ilâ 10. nesillerden sonra puan durumları bir dengeye ulaşmakta ve bundan sonraki değişimler küçük oranda olmaktadır.

Puan değerlerinin dengeye gelmesinin sebeplerini araştırdığımızda senaryoların birbirlerine benzedikleri göze çarpmaktadır. Benzerliklerin sebebini incelediğimizde ise kromozomların genellikle 8 ilâ 10. nesillere kadar geliştiği, bundan sonra kromozomların yapılarının benzeşmeye başladığı tespit edilmiştir. Bu seviyeden sonra kromozomlar küçük oranlarda değişmekte veya gelişmektedirler.

Benzer kromozomların kökenleri incelendiğinde genellikle aynı kökenden geldikleri görülmektedir. Firma örneğinde en iyi çözüm olan kromozom 5053 (üstün nesil) ile yedinci iyi çözüm olan kromozom 2697'nin (26. nesil) kökleri incelendiğinde kromozom 1892'de kesiştikleri görülmektedir (Bkz. Şekil 5.3). Kromozom 5053, 2697 ve 1892'nin olay dizilimi incelendiğinde %97 oranında aynı olduğu görülmektedir. Bazı

olayların diğer iki senaryonun ataları olan 1892’de olmamasına rağmen 5053 ve 2697’de ortak olduğu gözlemlenmiştir.

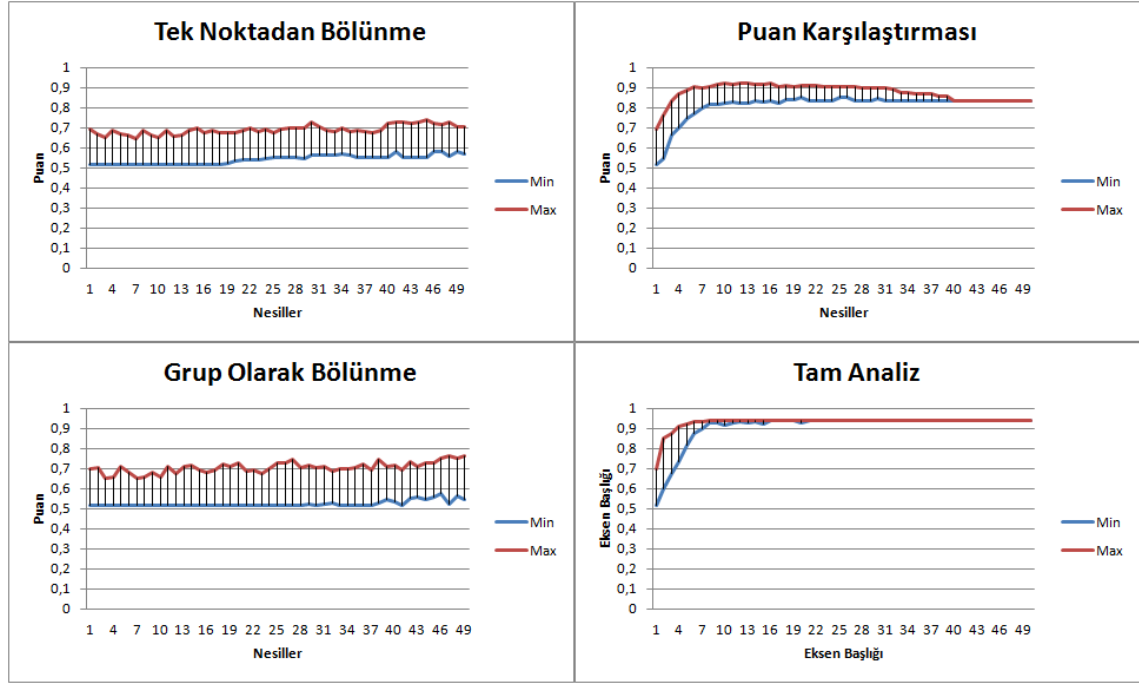


Şekil 5.3. Çözümler Arası İlişkiler

5.2. ÇAPRAZLAMA YÖNTEMLERİNİN SONUÇ ÜZERİNDEKİ ETKİSİ

Farklı çaprazlama yöntemlerinin sonuçlarını incelediğimizde tek noktadan bölünme ve grup olarak bölünme yöntemlerinin en iyi sonucu bulmada yeterli olamadıkları görülmektedir. Puanlama ve analiz yöntemlerinde yüksek puanlı genler seçilerek yeni kromozom çaprazlandığından ve kromozom içinde aynı sıradaki genlerin ikisi de kötü olduğunda her iki genin de ret edilerek olaysız bir gen eklenmesiyle puan-nesil grafiğinde yukarı doğru bir hareket gözlemlenmektedir.

En iyi sonuca Tam analiz yöntemiyle ulaşılırken uygun en hızlı sonuca ise puanlama yöntemiyle ulaşılmaktadır. Şekil 5.4'te çaprazlama yöntemine göre oluşan puan grafikleri ve hesaplama süreleri görülmektedir. Firma modeli için %92,37 başarılı bir çözüm 7 dakika 53 saniyede bulunmuştur.



Şekil 5.4. Çaprazlama Yöntemine Göre Oluşan Farklar, Firma Modeli

Çizelge 5.1. Çaprazlama Yöntemine Göre Oluşan Farklar, Firma Modeli

Yöntem	Tek Noktadan Bölünme	Grup olarak Bölünme	Puanlama	Tam Analiz
Elde Edilen En Yüksek Puan (1 üzerinden)	0,7387	0,7607	0,9236	0,9400
İşlem Süresi (sa:dk:sn)	00:05:21	00:05:38	00:07:53	01:18:12

5.3. PUAN DEĞERLERİ

Sonuçlarda genellikle %100'lük bir başarı beklenmemektedir. Çünkü istenen sonuç örneğimizde pazar payının ve firma kârının artması ise pazar payı olarak %100 ve firma karı olarak da yüksek bir değer verilmiştir. Modelde başka firmalar da olduğundan pazar payının %100 olması pek muhtemel değildir. Ancak asıl istenen hedef gerçekleşmektedir.

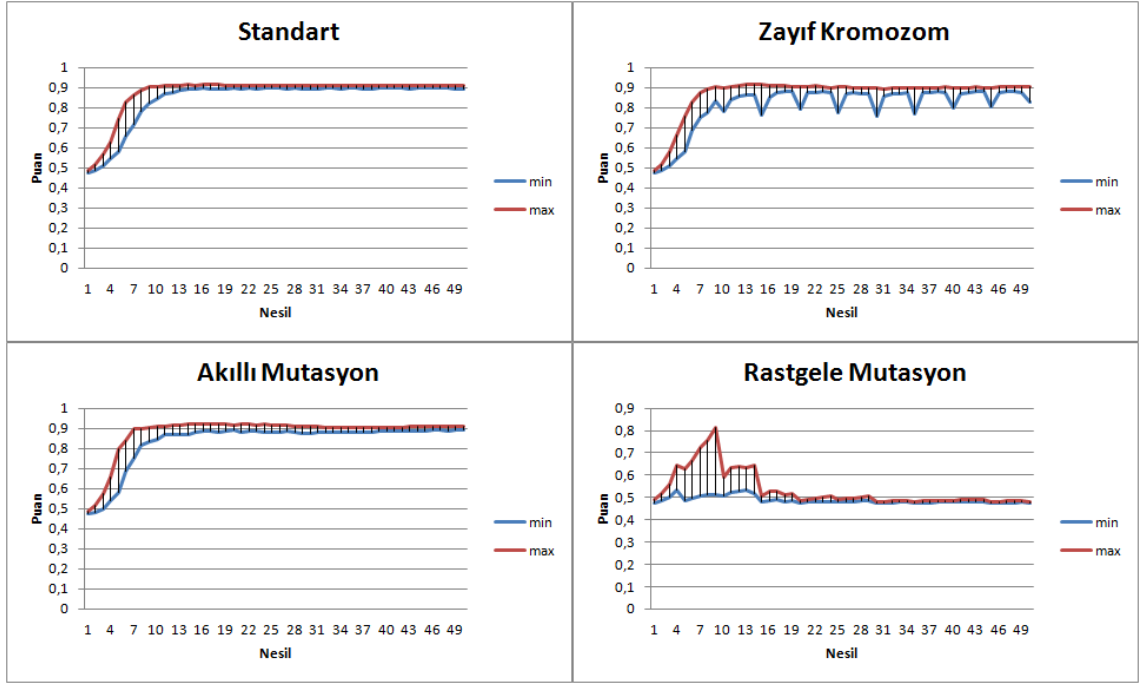
Model bir maksimizasyon problemi ise hedef değer olarak büyük bir değer alınmaktadır. Ayrıca kromozom puanını kriter olarak alınan durumların tanımlı aralığı da etkilemektedir. Hedef puan ve durumun maksimum değeri değiştirilerek farklı puan değerleri alınabilse de genellikle aynı sonuçlar bulunmaktadır. Tek etkisi, çaprazlama için kromozomun seçilme olasılığında bir farklılık olabilir.

5.4. MUTASYON VE ZAYIF KROMOZOMLARIN DEĞİŞİMİ

Nesiller ilerledikçe kromozomların puan seviyeleri bir dengeye ulaşmaktaydı. Bu durumun kırılması için yeni genlere ihtiyaç vardır. Ancak zayıf kromozomların değiştirilmesi veya rastgele yapılan mutasyonların yeterli faydası olmadığından akıllı bir mutasyon şekline ihtiyaç duyulmuştur. Bu maksatla geliştirilen akıllı mutasyon, en zayıf mutasyon oranı kadar geni en faydalı olanlarla değiştirmektedir.

Yapılan denemelerde (5 nesilde bir, %25 mutasyon olasılığı ve %10 mutasyon oranı ile) akıllı mutasyon ile %1 civarında daha olumlu sonuçlara ulaşılmıştır. (Bulunan oranlar modele göre değişmektedir.)

Puanlama yöntemi ile çaprazlanan Firma modelinde %25 mutasyon oranı ve %10 mutasyon olasılığı kullanıldığında Şekil 5.5'teki grafikte gösterilen sonuçlar elde edilmiştir. Kromozom değişimi ve mutasyon 5 nesilde bir olarak ayarlanmıştır.



Şekil 5.5. Çözümler Arası İlişkiler

Çizelge 5.2. Çözümler Arası İlişkiler

Mutasyonların Karşılaştırılması	Standart	Zayıf Kromozom	Rastgele Mutasyon	Akıllı Mutasyon
Elde Edilen En Yüksek Puan	0,9134	0,9168	0,8108	0,9207

Görüldüğü üzere bu model için üretilen rastgele kromozomlar genelde 0,5 puan etrafında toplanmaktadır. (Optimize edilmemiş tüm modellerde beklenen sonuçtur.) 5 nesilde bir eklenen zayıf kromozomlar, min çizgisini periyodik olarak aşağı çekmekte, puanlama yönteminin etkisiyle sonraki nesillerde min çizgisi tekrar yükselmektedir. En iyi sonuca zayıf kromozomlarının değişmesi, küçük oranda fayda sağlamaktadır.

Rastgele mutasyon ise 5 nesilde bir kötü kromozomlar ile birlikte iyi kromozomları da değiştirdiğinden nesiller ilerledikçe 0,5 puan seviyesinde dengelenmiştir.

Akıllı mutasyon ise doğrudan kötü genleri iyi olanlarla değiştirdiğinden nispeten daha büyük bir fayda sağlamıştır.

ALTINCI BÖLÜM

ÖRNEK SENARYO 1

Türkiye'deki 19,1 milyon hanenin yüzde 51,7'sinde yani yaklaşık 9,9 milyon hanede bilgisayar bulunuyor. Tüm hanelerin yüzde 34,6'sında masaüstü, 20,6'sında dizüstü bilgisayar bulunuyor (ekonomi.haberturk.com). Dolayısıyla Türkiye'de yaklaşık 6,6 milyon masaüstü bilgisayar kullanıcısı bulunuyor. Ancak 3 milyon kullanıcının sonraki alımlarda masaüstü bilgisayar alacağı tahmin ediliyor. Bilgisayar kullanıcı sayısı yıllık %1,3 artsa da, kullanıcıların dizüstü ve tablet bilgisayar tercihleri nedeni ile masaüstü bilgisayar satışları 2013 yılında %10,1 düşmüştür. Ancak masaüstü bilgisayar satışlarında önümüzdeki yıllar için ortalama %3,8 düşüş öngörülmektedir (www.webrazzi.com).

Bir bilgisayar firması Intel tabanlı masaüstü bilgisayar pazarlamaktadır. Bu firma bilgisayar satışındaki 3 yıllık stratejisini belirlemek istemektedir. Bu maksatla bir optimizasyon modelinin kurulmasına ihtiyaç duyulmaktadır.

Problem aynı model üzerinde lider firma, orta seviye firma ve alt seviye firma olmak üzere üç farklı rakip firma düzeyinde çözülecektir. Firmaların amacı gelecek 2 yıl için kârının maksimize edilmesidir.

Örnek olayda adı geçen firma isimleri alfabetik sıra oluşturacak şekilde oluşturulmuştur. Adı geçen firmalardan biri ile aynı isimde ticari bir firma varsa tesadüfidir ve kendisi ile hiçbir ilgisi yoktur.

6.1. MEVCUT DURUM

6.1.1. Firmalar

Bu sektörde toplam sekiz adet firma bulunduğu farz edilmektedir.

Firmaların genel durumları Çizelge 6.1'deki gibidir. Tüm firmalar başlangıçta, düşük konfigürasyonlu modeller için %10, yüksek konfigürasyonlu ürünler için %20 kar marjı uygulamaktadır. Firmalar gerektiğinde kampanya uygulayarak kâr marjını %10 ilâ %20 kadar daha düşürebilir veya prestij göstergesi olarak kâr paylarını %10 ilâ

%20 kadar artırabilirler. Ancak kâr marjı, tasfiye edilen ürünler hariç, %1 ilâ %50 aralığındadır. Firmalar genellikle %1 pazar payı kaybında reklam vermekte ve kampanya uygulamakta, %5 pazar payı artışında itibar göstergesi olarak kar paylarını artırmaktadırlar. Lider Firma, 3 ayda bir mevcut ürünlerinde iyileştirme yaparken, diğer firmalar 3 ilâ 6 ayda bir iyileştirme yapmaktadırlar. Lider firma yılda 2 kez yeni ürün (alt, orta ve üst sınıftan birer adet) geliştirmekte, diğer firmalar ise lider firmanın yeni ürün geliştirmesini müteakip 1 ilâ 3 ay içinde yeni ürün geliştirmektedirler. Ancak Gigabit firması yılda bir kez alt ürün, Hardware firması ise yılda bir kez üst ürün geliştirmektedir.

Çizelge 6.1. Firma Bilgileri

Firma / Durumlar	Anka	Bilgi	Çip	Dream	Elit	Future	Gigabit	Hardware
Pazar Payı	11,30%	6,78%	3,39%	13,56%	33,90%	19,77%	2,82%	8,47%
Tanınma Oranı	25%	15%	8%	30%	75%	44%	63%	19%
Mevcut Ürün Sayısı	2	1	1	3	7	4	1	2
Bütçe (milyon TL)	98	65	32	104	292	198	32	82
Aylık Sabit maliyet	370.000	210.000	130.000	450.000	930.000	610.000	130.000	290.000
Aylık Üretim kapasitesi	20.000	10.000	5.000	25.000	55.000	35.000	5.000	15.000

6.1.2. Ürün Özellikleri

Piyasada bulunan sekiz firma masaüstü bilgisayar üretmekte ve pazarlamaktadır. Tüm firmalar stoklarında bir önceki ay satılan miktarın %10 fazlası olacak şekilde üretim yapmaktadırlar.

Bugün bir ofiste kullanılmak üzere standart özellik ve performansa sahip bir bilgisayarın performans puanı 100 olarak kabul edilmiştir. Müşteri ihtiyaçları ve teknoloji bilgisayar performans ve yeteneklerinde yıllık %10-15 civarı artış getirmektedir. Ürün maliyetleri performans puanı oranında düştüğünden, ilerleyen yıllara rağmen aynı kategoride bulunan ürünler yaklaşık aynı maliyete sahip olmaktadır. Yani bugünkü 100 performans puanlı bir ürün ile bir yıl sonraki 112 performans puanlı bir ürün yaklaşık aynı maliyettedir. Firmalar satış fiyatlarını genellikle üç ayda bir güncellemektedirler. Ancak kar marjı %5'in altına düşmediyse zam yapmamaktadır. Azalan maliyetlerde fiyatı belirlenen kar marjına göre düşürmektedir.

Gigabit firması sadece düşük konfigürasyonlu ürün ve Hardware firması sadece yüksek konfigürasyonlu ürün üretmektedirler.

Yeni bir ürün üretilmeden önce projelendirilmektedir. Düşük, orta ve yüksek konfigürasyonlu ürünlerin projelendirilmesi sırasıyla 300.000 TL, 500.000 TL ve 600.000 TL tutmaktadır. Projelenmiş bir ürünün üretime başlanması ise 750 adet birim ürün maliyeti tutmaktadır. Yeni bir ürünün projelendirilmesi bütçenin yarısını aşmadığı durumlarda yapılmaktadır.

Mevcut ürünlerin gelişen performans ihtiyacına cevap verebilmeleri için bazı özellikleri geliştirilerek aynı isimle satılmaya devam edilmektedir. Ürünlerin geliştirilmesi üç kategoride yapılmaktadır ve tüm ürünlere uygulanabilmektedir. Aynı ürün tekrar geliştirilebilir. Firma ürüne yapılacak geliştirme için bir kategoriye karar verebilmektedir ancak artan maliyet ve performans tabloda gösterildiği gibi değişebilmektedir. Ürün fiyatı bir sonraki fiyat güncellemesine kadar değiştirilmemektedir.

Çizelge 6.2. Geliştirme Kategorileri

Geliştirme seviyesi	Performans artışı	Maliyet Artışı (TL)
A	+1 - 3	+15 ~ 25
B	+2 - 4	+25 ~ 35
C	+3 - 6	+35 ~ 50

Üretilen modellerin ömürleri normal dağılıma göre genellikle 18 ilâ 24 ay arasında değişmektedir. Bu süre sonunda ürün üretimden kaldırılır. Eğer bir ürünün satış miktarı aylık 1000 adedin altında veya toplam üretiminin %5'i altında kaldığında da ürünü sonlandırabilmektedir. Ancak ürünün ilk 6 ayında ürünün yeni tanınması nedeniyle tasfiye yapmamaktadır. Firma, aktif ürün sayısı için altında ise ürünlerinin ömrünü uzatmaktadır. Tasfiye edilen ürünler %15'e kadar zararla elden çıkarılmaktadırlar.

6.1.3. Müşteri Bilgileri

Önümüzdeki yıllarda masaüstü bilgisayar piyasası için Türkiye’de 3 milyon potansiyel alıcı olduğu tahmin edilmektedir. Yapılan istatistiklere göre Türkiye’de 2013 yılının son çeyreğinde 207 bin adet masaüstü bilgisayar satılmıştır. Masaüstü bilgisayar satışının yıllık %3,8 azalacağı öngörülmektedir.

Kullanıcıların performans ihtiyacı, piyasa hakkındaki bilgisi, alım tercihi gibi bilgileri gösteren çizelge aşağıdadır.

Çizelge 6.3. Müşteri Tercihleri ve İhtiyaçları

Müşteri						
Piyasa hakkındaki bilgisi	Alım Tercihi	Oranı	Maddi İmkânı (TL)	Yenileme periyodu (ay)	Performans ihtiyacı (Yıllık %10-15 artış gösterir)	Seçim Kriteri Seçilen firmalar arasından ilgili kriterin değeri oranına göre rastgele seçim yapılır
Bilgili	Performans	10%	1750 - 3500	18 - 36	140 – 250	En yüksek performanslı 5 ürün arasından seçer.
	Fiyat/Performans	25%	700 - 2800	24 - 48	80 - 200	En yüksek 5 fiyat/performans oranına sahip ürün arasından seçer.
	Fiyat	20%	500 - 1750	36 - 60	80 - 150	En düşük fiyatlı 8 ürün arasından seçer. Firmayı 1 birim fazla tanıyorsa 20TL fazla ödeyebilir. 1 birim fazla performans için 10TL fazla ödeyebilir.
Az Bilgili	Fiyat	20%	500 - 2100	36 - 72	80 - 160	En düşük fiyatlı 8 ürün arasından seçer. Firmayı 1 birim fazla tanıyorsa 20TL fazla ödeyebilir
	Bilinen Marka	25%	500 - 1750	30 - 84	80 - 140	En çok tanıdığı 5 firma arasından seçim yapar.

Tabloda gösterilen maddi imkân ve performans ihtiyacı düzgün dağılmıştır. Yenileme periyodu ise normal dağılımdır.

Müşterilerin %25'i yalnızca promosyon uygulanan bayilerden alım yapmaktadır.

Müşterilerin alım yaparken markayı tanıma bilgisi de alım tercihleri arasında yer almaktadır. Markanın tanınması sadece olumlu olarak kabul edilmiştir. Markanın olumsuz tanınması bu örneğin dışında bırakılmıştır. Müşterinin markayı tanıması; reklamlara, ürünün elinde bulunmasına ve başka müşterilerle olan etkileşimi sonucu değişmektedir. Yapılan reklamların müşteri üzerinde farklı etkileri bulunmaktadır (Bu etkiler reklamlar ile beraber ele alınacaktır). Müşterinin alım yaptığı markayı artık tanıdığı kabul edilmiştir. Müşterinin marka ile hiçbir etkileşimi olmadığında marka yıllık %20 ~ 40 kadar unutulmaktadır.

6.1.4. Yetkili Bayiler

Müşterilerin %25'i fiyat avantajı nedeniyle sadece yetkili bayilerden alım yapmaktadır. Firmalar bayiler ile bir yıllık promosyon anlaşması yaparlar ve bu bayilere %20 daha düşük kar marjı ile satış yaparlar. (Kar marjı %10 ise bu bayilere %8 kar ile satış yapılır.) Bayiler ile promosyon anlaşması yapan firmalar, bu bayiler dışında yaygın dağıtım ile de satış yaparlar.

Genel olarak firmaların %30'u promosyon anlaşması yapar. Lider firmanın promosyon anlaşması yapma oranı ise %50'dir. Dream firması hiçbir zaman promosyon anlaşması yapmazken Bilgi firması ise promosyon anlaşması her yıl yenilemektedir.

6.1.5. Reklamlar

Firmalar televizyon, dergi, internet, outdoor veya duyurum yoluyla reklam verebilmektedirler. Üç ayın toplamı için reklama ayrılan bütçe %1 ile sınırlandırılmıştır.

Reklamlarda elde edilen fayda ve maliyetler tabloda gösterildiği gibidir. Reklamın, markanın tanınmasına olan etkisi olarak daha önce ulaşılammış müşterilerin ne kadarına daha ulaşıldığını ifade etmektedir. Bu şekilde aynı reklamın zaten tanıyan müşterilerdeki etkisi ile tanımayan müşterilerdeki etkisi farklı olmaktadır.

Çizelge 6.4. Reklam Etkileri

Reklam Türü	Markanın Tanınmasına Olan Etkisi	Maliyet (TL)	Açıklamalar
Reklam (TV)	%10	718.000	1 Ay için, 8 sn süreli 4 kanalda 50'şer kez gösterim için
Reklam (Dergi)	Bilinçli müşteri gruplarında %20 Diğerlerinde %3	79.000	Aylık dergi
Duyurum	Bilinçli müşteri gruplarında %10 Diğer %4	150.000	
Reklam (internet)	Bilinçli müşteri gruplarında %10 Diğer %4	44.000	Aynı reklamın 3 ay boyunca azalarak da olsa etkisi devam eder
Reklam (Outdoor)	%8 ~ 10	644.280	1 aylık

6.2. MODEL

Firma modelinin tanımlanmasında aşağıda bahsedilen unsurlar kullanılmıştır.

Modelde Firma, Ürün, Müşteri, Dağıtıcı, Genel olmak üzere beş farklı cisim (nesne türü) tanımlanmıştır; Modelde kullanılacak tüm nesneler bu cisimlerden türetilcektir. Geliştirilen sistem, bu cisimlerden türetilen nesnelerin tüm değerlerinin verilen şartlara uymasını sağlayacak, nesneler arasındaki etkileşimleri yönetecek ve gereken istatistiklerin tutulmasını sağlayacaktır.

Firma nesneleri modelde belirtilen sekiz adet firmayı temsil edecektir. Ürün nesneleri ise bu firmaların üretilip pazarladıkları ürünleri temsil edeceklerdir. Modelde açıklanan ve beş temel gruba ayrılan müşteri kitlesi Müşteri nesneleri ile temsil edilecektir. Genel cisminden türeyecek bir Ortam isimli nesne bulunacaktır. Bu nesnenin amacı ise sisteme ait genel değerlerin tutulması ve yönetilmesidir.

Tüm cisim yapılarına başlangıçta gerekli durum değişkenleri eklenmiştir. Böylece türeyen tüm nesne yapıları, tanımlamaları esnasında ait oldukları cisim yapılarının durum değişkenlerini miras alacaklardır.

Modelin Tanım, Oluşturma ve Fonksiyon Tanımları bölümleri bulunmaktadır. Tanım bölümünde, tüm modelin kullanabilmesi gereken cisim, nesne, olay gibi yapılar ile kullanıcının farklı amaçlarla oluşturabileceği yapılar tanımlanır.

Oluşturma bölümünde ise tanımlı yapıları cisim ve nesnelerin kendileri oluşturulur ve istenen durum değişkenleri burada eklenir. Durum değişkenlerine ait her türlü tahdit, bağlantı, ilk değer, hedef değer ve diğer tüm özellikler burada tanımlanır. Olay tanımları yine bu bölümde yapılır ve fonksiyon tanımları bölümünde tanımlanan olayların şart ve eylem fonksiyonları burada ilişkilendirilir ve olayların planlanabilme, tahdit, parametre yapıları burada sisteme tanımlanır. Parametre yapıları ile hangi olayın hangi türde fail, etkilenen ve ilgili cisim, küme veya nesneleri kabul ettiği bilgisi sisteme verilir. Her çözücü için modelin bir kopyası oluşturulur ve bu bölümdeki tanımlamalar her çözücü için bir kez yapılır.

Fonksiyon tanımları bölümünde ise olaylara ait şart ve eylem fonksiyonlarından başka tetikleme, etki fonksiyonları, bulanık mantık kabul fonksiyonları ve kullanıcının gerekli gördüğü diğer fonksiyonlar tanımlanır.

Ayrıca sistemde tanımlı özel iki olay bulunmaktadır. İlki, her kromozom için kromozomun simülasyonun ve her iterasyonunun başlangıcında Modeli başlangıç durumuna getiren hazırlık olayı tanımlanmıştır. Aslında tüm durum değişkenleri sistem tarafından ilk değerlerine getirilmektedir. Ancak kullanıcı tarafından tanımlanan başka yapı ve listeler varsa, bunların da ilk değerine alınmasına imkân verilmiştir. Diğer özel olay ise, her birim zaman ilerlemesinde yapılması gereken işlemlerin sisteme girilmesine olanak sağlar. Bu olaya ait eylem fonksiyonu da bu bölümde tanımlanmakta ve oluşturma bölümünde hangi fonksiyonun özel olay eylem fonksiyonları olduğu sisteme gösterilir.

6.2.1. Nesne Yapıları

6.2.1.1. Firmalar

Firmalar çizelgede gösterilen durum değişkenleri ile modellenmiştir. Ayrıca firmanın sonraki ürün geliştirme zamanı, sonraki ürün projelendirme zamanı ve sonraki

promosyon zamanı tutulmaktadır. Firmalara ait durum değişkenlerine model başlangıcında Çizelge 6.1’de belirtilen değerler kullanılmıştır.

Çizelge 6.5. Firma Durum Değişkenleri

Durum Değişkeni	Açıklama
PazarPayı	Firmanın Pazar payı
Kar	Firmanın toplam kârı
Tanınma	Firmanın müşteriler tarafından tanınma oranı. Yalnız başlangıçta kullanılmakta, bu değer ile müşterilerin firmayı tanıma oranları bağlantılı durum olarak oluşturulmaktadır. Devamında da bu değer kullanılmakta, tanınma durumu güncellenmemektedir.
ÜrünSayısı	Firmanın aktif ve pasif toplam ürün sayısı
AktifÜrünSayısı	Sadece aktif ürün sayısı
Maliyet	Firmanın simülasyon başlangıcından beri yaptığı harcama toplamı
Bütçe	Firmanın elinde bulunan nakit miktar
AltKarMarjı	Firmanın yeni üreteceği ürüne fiyat belirlerken kullanacağı en düşük kâr oranı
ÜstKarMarjı	Firmanın yeni üreteceği ürüne fiyat belirlerken kullanacağı en yüksek kâr oranı
AylıkSabitMaliyet	Firmanın aylık sabit maliyeti
ÖncekiPazarPayı	Firmanın pazar payının %1 azalması veya %5 artmasını kontrol edebilmesi için önceki pazar payını tutan durum değişkeni. Bu değer ilgili orandaki değişim olmadıkça güncellenmez.
ÜretimKapasite	Firmanın toplam üretim kapasitesi
AylıkÜretim	Firma ürünlerin aylık üretim toplamı ile hesaplanan aylık üretim miktarı
DağıtıcıVar	Firmanın promosyon anlaşması olup olmadığı
DağıtıcıSonu	Firmanın promosyon anlaşması varsa ne zaman sona erdiği
FirmaNo	Firma numarası
AylıkSatış	Firma ürünlerin aylık satış toplamı ile hesaplanan aylık satış miktarı

6.2.1.2. Ürünler

Ürün nesneleri önceden tanımlanarak bir ürün havuzu oluşturulmakta ve yeni ürün üretecek firmalar için ürün havuzundan bir ürün tahsis edilerek ürün istenen özelliklere göre yapılandırılarak simülasyona dâhil olur. Ürün nesnesi için tanımlanan durum değişkenleri Çizelge 6.6’da gösterilmiştir.

Çizelge 6.6. Ürün Durum Değişkenleri

Durum Değişkeni	Açıklama
SabitMaliyet	Ürünün projelendirme maliyeti + üretime başlama maliyeti olarak hesaplanmaktadır.
BirimMaliyet	Bir adet ürünün üretilmesi için gerekli maliyet $BirimMaliyet = \frac{Performans^{\frac{5}{4}}}{3} \times MaliyetÇarpanı$ olarak hesaplanmaktadır.
ToplamMaliyet	Ürünün sabit maliyetinin tahmini ürün ömrü boyunca yapılacak üretimler, firmanın aylık sabit maliyeti ve ürünün birim maliyeti dikkate alınarak yapılan maliyet hesaplaması. Aylık en az 5000 adet üretileceği varsayılmıştır. Toplam maliyet bir ürün içindir. Ürünün Birim Başına düşen toplam maliyeti $Maliyet = BirimMaliyet + \frac{SabitMaliyet}{ÜrünÖmrü \times ÜrünStokMiktarı} + \frac{FirmaAylıkSabitMaliyet}{FirmaAylıkÜretim}$
Fiyat	Ürünün toplam maliyeti Firmanın kar marjına göre hesaplanan ürün fiyatıdır.
Performans	Ürünün işlemci, hafıza, disk, ekran kartı gibi tüm özelliklerinin dâhil edildiği bir puanlama sistemi olduğu kabul edilmiştir. Günümüz için temel seviye 100, ihtiyaç duyulan aralık 80 ilâ 200 olduğu varsayılmıştır. Performans ihtiyacı aylık artmaktadır.
Seviye	Genel olarak alt, orta ve üst seviye olmak üzere ürün performansına göre üç grup halinde kategorize edilmektedir.
BaşlangıçTarihi	Ürünün ilk üretim tarihi
TasfiyeTarihi	Ürün için planlanan tasfiye tarihi. Gerekli görüldüğünde bu tarih ötelenebilmektedir.
Ömür	Ürünün tahmini ömrü
Durumu	Ürünün hangi durumda olduğunu belirtir. Aşağıdaki değerlerle ifade edilmiştir. 1: Proje, 2: Aktif, 3: Tasfiye Sürecinde, 4: Tasfiye
ToplamKar	Üründen elde edilen toplam kâr
OncekiAySatış	Ürünün bir önceki ayki satışı. Sonraki ay yapılacak üretim miktarı için kullanılmaktadır.
AylıkSatış	Ürünün o ayki satışı
ToplamSatış	Ürünün toplam satış miktarı
StokMiktarı	Ürünün stok miktarı
FirmaNo	Ürünün ait olduğu firmanın numarası
UrunNo	Ürüne verilen sıra numarası
ÜretilenMiktar	Ürünün aylık üretim miktarı

Müşterilerin ürün alımlarını simüle etmek maksadıyla 1000 adet müşteri grubu oluşturulmuştur (bkz. 6.2.1.3 6.2.1.3. Müşteriler). Her müşteri grubunun farklı alım gücü ve tercihi olması sebebiyle ürünler farklı kriterlerle müşterinin alım listesine dâhil olmakta ve sıralanmaktadır. Her zaman ilerlemesinde tüm ürünlerin 1000 farklı müşteri grubuna göre sıralanabilmesi için hesaplamaları yapmak üzere şu şekilde bir yapı geliştirilmiştir.

Çizelge 6.7. Ürünün Alım Kriteri Hesabında Kullanılan Ek Yapı

Üye Değişkeni	Açıklama
Urun	Ürün Nesnesi
fpo	Fiyat / performans oranı ($\frac{Fiyat}{Performans^{3/2}}$)
Perf	Performans değeri
Fiyat	Ürünün Fiyatı
pPerf	Sıradaki müşteri grubuna göre hesaplanan performans puanı
pFiyat	Sıradaki müşteri grubuna göre hesaplanan fiyat puanı
pfpo	Sıradaki müşteri grubuna göre hesaplanan fiyat / performans oranı puanı
pTanınma	Sıradaki müşteri grubuna göre hesaplanan firmanın tanınması puanı
FirmaNo	Firma referans numarası

Ürünün müşteri kriterlerine göre puanlanmasında öncelikle fiyat, performans, fiyat-performans oranı değerlerinin azami ve asgari değerleri bulunarak tüm değerler 0-100 arasına getirilmektedir. Fiyat ve fiyat-performans oranı için düşük değer yüksek puan alacak şekilde hesaplanmaktadır (Bkz. 6.2.5 Alım Kriterleri).

6.2.1.3. Müşteriler

Beş farklı müşteri tipi için beş müşteri nesnesi oluşturulmuştur. Gerekli istatistiki değerler bu nesneler üzerinden tutulmaktadır.

Çizelge 6.8. Müşteri Durum Değişkenleri

Durum Değişkeni	Açıklama
Sayı	Müşteri grubundaki kişi sayısı
MaddiAlt	Müşteri grubundaki kişilerin ürün alımı için tahsis ettikleri asgari para miktarı
MaddiÜst	Müşteri grubundaki kişilerin ürün alımı için tahsis ettikleri azami para miktarı
PeriodAlt	Müşteri grubundaki kişilerin asgari ürün yenileme periyodu
PeriodÜst	Müşteri grubundaki kişilerin azami ürün yenileme periyodu
PerformansAlt	Müşteri grubundaki kişilerin asgari performans ihtiyacı
PerformansÜst	Müşteri grubundaki kişilerin azami performans ihtiyacı
Tanıma.<Firma>	Müşterinin firmayı tanıması. Burada bağlı durum kullanılmıştır. Yani sistem her müşteri-firma ikilisi için ayrı bir tanıma durum değişkeni oluşturmuştur.
TipNo	Müşteri tip numarası

Yukarıdaki yapı ile birlikte müşterilerin ürün alımlarını daha gerçekçi oluşturabilmek için 1000 adet müşteri grubu oluşturulmuştur. Müşterilerin alım periyodu ve modelde belirtilen aylık müşteri sayısı (Bkz. 6.2.4

6.2.8. Aylık Müşteri Sayısı) dikkate alınarak başlangıçta 68-69 kişilik müşteri grupları oluşturulmuştur. Her müşteri grubu için spesifik olarak maddi imkan, performans ihtiyacı ve firmayı tanıma oranları belirlenmiştir. Ayrıca tüm grupların dörtte biri yalnızca promosyonlu ürün alacak şekilde tercih yapmaktadır.

6.2.2. Olaylar

Modelde standart olay, reklamlar, ürün geliştirme, yeni ürün projelendir, üretime başla, kampanya, prestij ve promosyon olmak üzere 8 farklı olay tanımlanmaktadır.

6.2.2.1. Standart Olay

Standart olay, her birim zaman ilerlemesinde yapılması gereken işlem ve kontrolleri yönetmektedir. Bu olay kromozoma dâhil değildir. Firma ve ürüne ait aylık üretim, satış gibi durum değerleri burada sıfırlanarak sonraki birim zamanda doldurulması sağlanır.

Ayrıca firmanın dağıtıcı ile promosyon anlaşmasının sonlanması, ürünün tasfiye veya ömrünün uzatılması işlemleri, sonraki birim zamanda yapılacak üretim miktarının hesaplanması, ürünlerin müşterilere göre puanlanması ve müşterinin ürün alım işlemleri, müşteri sayı ve performans ihtiyaçlarının güncellenmesi, firmaya ait pazar payının, maliyet ve kâr miktarlarının hesaplanması, ürünün satış miktarlarının kaydedilmesi ve diğer firmalarının hareket tarzlarının belirlenmesi (tetikleme işlemleri) bu bölümde yönetilmektedir. Birim zaman olarak ay alınan modelimizde buradaki işlemler aylık olarak tekrarlanan kontrol ve işlemlerdir.

6.2.2.2. Reklamlar

Modelde beş adet reklam olayı tanımlanmıştır (Bkz. 6.1.5 6.1.5. Reklamlar). Her bir reklam olayının şartı ilgili reklamın maliyetinin bütçenin %1'i geçmemesidir. Bu şart gerçekleştiği takdirde planlı olan reklam olayı gerçekleştirilir. Her bir reklam olayında müşteri tipine ve reklam türüne göre firmanın tanınması artırılır, reklamın maliyeti kadar firma bütçesinden tahsil edilir.

İnternet reklamlarında reklamın azalarak devam eden etkisi olduğu tanımlandığından burada modelin “etki” özelliği kullanılmıştır. Etki fonksiyonu olarak $Etki\ değeri = Değer \times \frac{Etki\ Ömrü-t}{Etki\ Ömrü}$ kullanılmıştır. Etki ömrü internet reklamları için 3 ay, tanınma oranına etkisi ise müşteri tipine göre %10 veya %4 olarak verilmiştir. Diğer reklamlarda tanınma değeri 6.2.2 Tanınma’da belirtildiği gibi ve 6.1.5 6.1.5. Reklamlar’da gösterilen oranlar dâhilinde değiştirilmektedir.

6.2.2.3. Ürün Geliştirme

Üç seviye ürün geliştirme olayı mevcuttur. Olay şartı, firma bütçesinin 100.000 TL üzerinde olması ve ürünün üretime başlayalı 6 ayı geçmesidir.

Seviye 1 ürün geliştirme olayı için ürün performansı 1 ilâ 3 artırılıp ürün maliyeti 15 ilâ 25 TL artırılmaktadır. Ürünün fiyatı değiştirilmemektedir, çünkü zaten ürünün birim maliyeti zamanla düşmektedir. Seviye 2 için 2 ilâ 3 birim performans artışı için maliyet 25 ilâ 35 TL, seviye 3 için 3 ilâ 6 birim performans artışı için maliyet 35 ilâ 50 TL artırılmaktadır.

6.2.2.4. Yeni Ürün Projelendir

Ürün projelendirme olayları üç adettir; Alt, orta ve üst seviye ürün projelendirme. Olay şartları alt, orta ve üst seviye ürünler için sırasıyla firma bütçesinin 6 milyon, 9 milyon ve 12 milyon TL üzerinde olmasıdır. Projelendirme maliyetleri ise model tanımında verildiği gibidir.

Yeni ürün projelendirme olayında önce ürün havuzundan yeni bir ürün alınarak firma ile ilişkilendirilir ve istenen seviyeye göre ve zamanın performans kriterlerine göre bir performans değeri atanır. Ürünün başlangıç tarihi atanır. Sabit maliyetine projelendirme maliyeti eklenir. 18 ilâ 24 ay arasında bir ömür biçilir. Ürünün birim maliyeti hesaplanır. İlgili maliyetler firmanın bütçesinden düşülür. Firmanın ürün sayısı güncellenir. Son olarak ürün için üretime başlama olayı tetiklenir.

6.2.2.5. Üretime Başla

Üretime başlama olayı planlanmayıp yeni ürünün projelendirilmesi ile tetiklenmektedir. Bu olay için bir şart fonksiyonu tanımlanmamıştır.

Olayın eyleminde, ürünün maliyetinin hesaplanması, firma bütçesinden 750 ürünün toplam maliyeti kadarlık bir miktar üretime başlama maliyeti olarak düşülmektedir. Ürünün durumu aktif olarak güncellenmektedir. Firmanın kar marjına göre ürünün fiyatı hesaplanmaktadır.

$$\begin{aligned} \text{Ürün Fiyatı} = & \text{Ürün}_{\text{Toplam Maliyet}} \\ & \times \left(1 \right. \\ & + \left(\text{Firma}_{\text{AltKarMarjı}} + (\text{Firma}_{\text{ÜstKarMarjı}} - \text{Firma}_{\text{AltKarMarjı}}) \right. \\ & \left. \left. \times \frac{\text{Ürün}_{\text{Performans}} - \text{Ortam}_{\text{TemelPerformans}}}{1,5 \times \text{Ortam}_{\text{TemelPerformans}}} \right) \right) \end{aligned}$$

6.2.2.6. Kampanya

Firmanın kampanya uygulaması olayında firmanın alt ve üst kâr marjları %10 ~ %20 düşürülür.

6.2.2.7. Prestij

Firma prestij göstergesi olarak alt ve üst kâr marjları %10 ~ %20 yükseltir.

6.2.2.8. Promosyon

Firmanın “DağıtıcıVar” durum değişkenini 1 olarak günceller ve “DağıtıcıSonu” durum değişkenini 12 ay sonrasına günceller. Olay şartı ise mevcut bir dağıtıcının olmamasıdır.

6.2.3. Diğer Firma Hareket Tarzları (Tetiklemeler)

En yüksek pazar payına sahip firma lider firma olarak kabul edilmiştir. Diğer firmalar genel olarak lider firmayı taklit etme eğilimindedirler. Tetiklenen firmalar için de olay şartı kontrol edilerek eylemin gerçekleşip gerçekleştirilmeyeceğine karar verilir.

Örneğin ilgili firmanın bütçesi yeterli değilse ürünü projelendiremez. Pazar payının %5 artması durumunda firma prestij olayını tetiklemektedir.

Önceki pazar payının %1 düşmesi durumunda ucuzdan pahalıya doğru tüm reklamlar firma için tetiklenir. Olaylar bütçe durumuna göre gerçekleştirilir. Ayrıca kampanya olayı tetiklenir ve önceki pazar payı güncellenir.

Lider firma 3 ayda bir, diğer firmalar 3 ilâ 6 ayda bir tüm ürünlerini geliştirir. Ürün geliştirme seviyesi rassal olarak belirlenir. Tetiklemeden sonra, sonraki ürün geliştirme zamanı belirlenerek tekrar tetiklenmesi sağlanır.

Lider firma yılda 2 kez (6 ayda bir) alt, orta ve üst sınıf ürün projelendirir. Lider firmanın ürün projelendirmesi ile diğer firmalar da 1~3 ay gecikmeyle her üç sınıf ürün projelendirirler. Yalnız Gigabit yılda bir kez sadece alt sınıf ürün geliştirir, Hardware ise yılda bir kez sadece üst sınıf ürün geliştirir.

Ürünün üretime başlaması, asıl firmamızda olduğu gibi, ürünün projelendirilmesi ile tetiklenir.

Sonraki promosyon tarihi gelen firma için; Önce, sonraki promosyon tarihi 12 ay sonrasına güncellenir. Dream firması bayi ile promosyon anlaşması yapmadığından tetikleme yapılmaz. Bilgi firması düzenli anlaşma yaptığı için doğrudan tetikleme yapılır. Lider firma için %50 ihtimalle, diğer firmalar için ise %30 ihtimalle promosyon olayı için tetikleme yapılır.

6.2.4. Tanınma

Modelde tanınma, müşterinin firmayı tanıma oranı olarak kabul edilmekte ve müşterinin ürün alım kriterlerinde kullanılmaktadır ve standart olay içerisinde yönetilmektedir.

Tanınma oranının güncellenmesinde artış ve azalış değerleri bire bir atanmamaktadır. Sebebi ise örneğin bir reklam olayı ile müşteri grubunun A firmasını tanınması 10 birim artırılacaksa, firmayı zaten tanıyan müşterilerin varlığından dolayı bu değer 10 birim artırılmaz. Artış değeri, tanımayan müşterilerin azaldığı oran olarak hesaplanır;

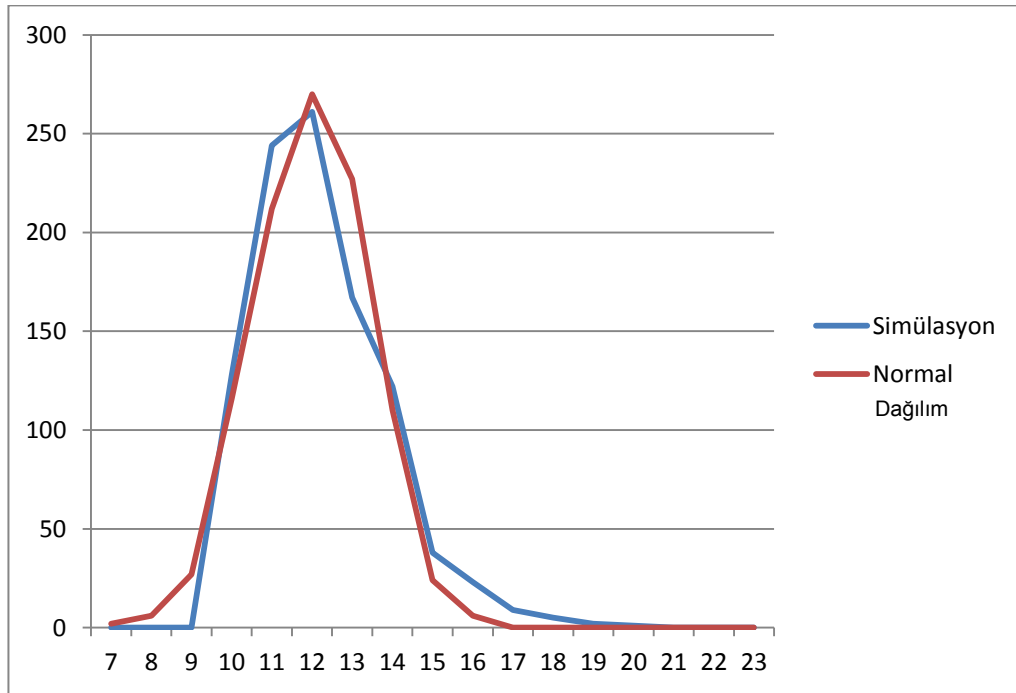
i: Meydana gelen reklam olayı anı için

$$Tanınma_{i+1} = 1 - (1 - Tanınma_i) \times (1 - DeğişimOranı_i)$$

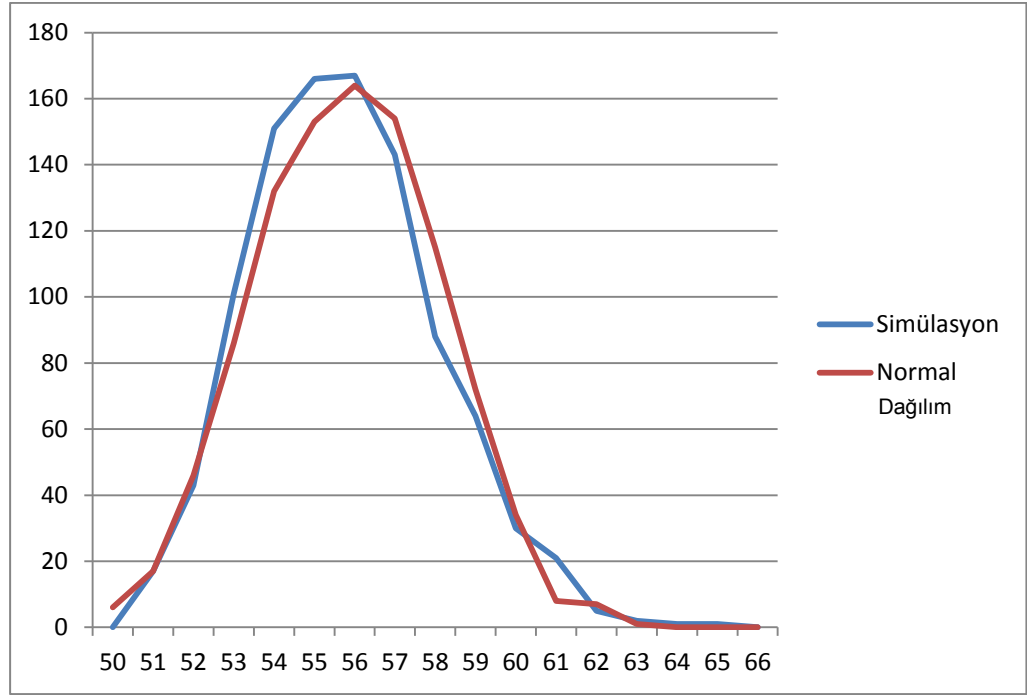
6.2.4.1. Müşteri Etkileşimi

Her müşteri %10 ihtimalle başka herhangi bir müşteri ile etkileşimde bulunur. Etkileşimde Her iki müşteri de birbirinin tanıdığı firmaları tanır. Grup olarak modellenen müşterilerin firmaları tanımlarındaki etkileşimi modellemek için müşteriler tek tek modellenerek işlem simüle edilmiştir. Müşteri gruplarının kendi içinde etkileşimde bulunduğu kabul edilmiş, farklı müşteri gruplarının etkileşimi ihmal edilmiştir. Elde edilen sonuçlar aşağıdaki gibidir;

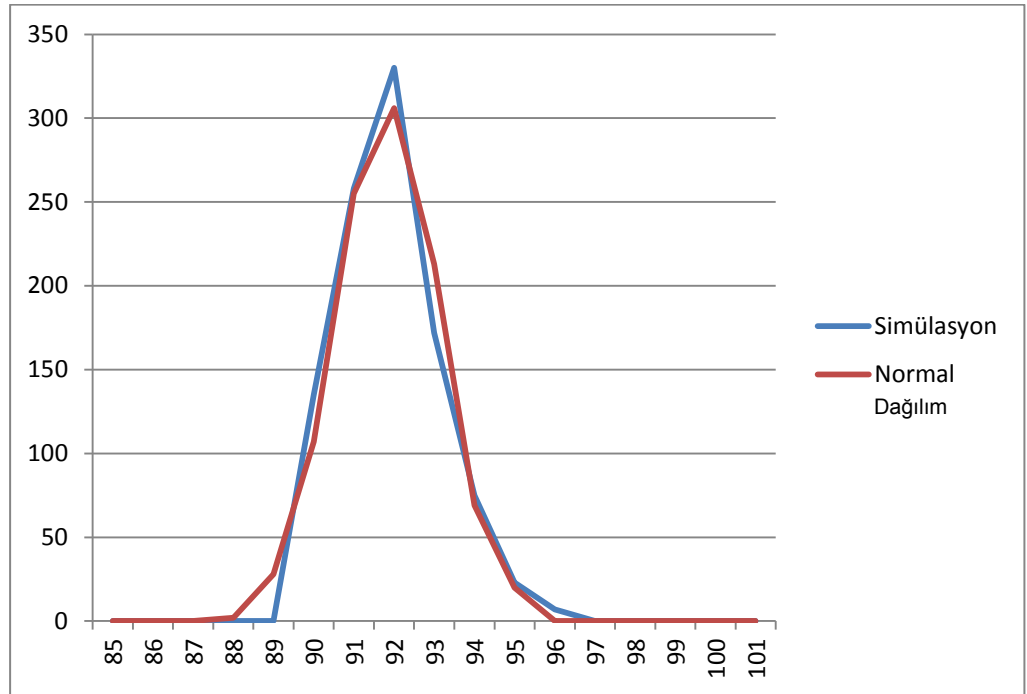
1000 örnek ile yapılan simülasyonda sırasıyla %10, %50 ve %90 tanınan bir firmanın bir etkileşimden sonra firmaların tanınması aşağıdaki şekillerdeki gibi dağılmaktadır;



Şekil 6.1. %10 Tanınan Firma



Şekil 6.2. %50 Tanınan Firma



Şekil 6.3. %90 Tanınan Firma

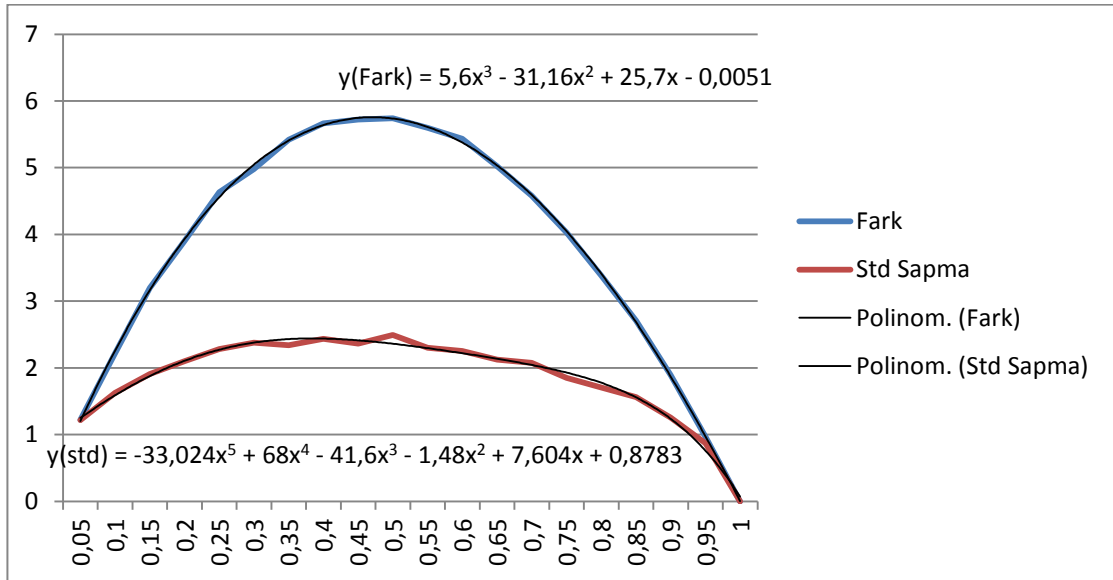
Şekillerde simülasyondan elde edilen sonuçlar ile örneklemin ortalaması ve standart sapması parametre alınarak oluşturulan normal dağılımlar ile karşılaştırılmıştır. %50 için dağılım 3 parametrelili Weibull dağılımına kaysa da genel olarak normal dağılım firmanın tanınmasında oluşan değışimi karşılamaktadır.

Diğer oranlar için aynı simülasyon yapıldığında Çizelge 6.9'daki sonuçlar elde edilmiştir.

Bu çizelgeyi grafik haline getirdiğimizde ortalamada meydana gelen değışimi ve standart sapmayı formüle edebiliriz. Şekil 6-4'de simülasyon sonucu elde edilen değerler ile çizilen grafik ve formüle edilen polinom gösterilmektedir. Formülde önceki tanınma değeri “x” olarak gösterilmiştir.

Çizelge 6.9. Firmaların Tanınma Seviyelerine Göre %10 Etkileşim Sonucu Firmanın Artan Tanınma Değerleri

Önceki Değer	Yeni Değer		Fark	
	Ortalama	Std. Sapma	Değer	Oran
0	0	0	0	0,00%
5	6,222	1,216847	1,222	24,44%
10	12,214	1,625486	2,214	22,14%
15	18,199	1,906148	3,199	21,33%
20	23,899	2,099714	3,899	19,50%
25	29,631	2,280096	4,631	18,52%
30	34,977	2,379595	4,977	16,59%
35	40,417	2,339041	5,417	15,48%
40	45,663	2,435864	5,663	14,16%
45	50,724	2,364281	5,724	12,72%
50	55,739	2,490959	5,739	11,48%
55	60,595	2,298907	5,595	10,17%
60	65,435	2,254279	5,435	9,06%
65	70,021	2,12569	5,021	7,72%
70	74,579	2,076959	4,579	6,54%
75	79,03	1,850703	4,03	5,37%
80	83,387	1,70799	3,387	4,23%
85	87,707	1,563058	2,707	3,18%
90	91,891	1,255834	1,891	2,10%
95	95,968	0,873485	0,968	1,02%
100	100	0	0	0,00%



Şekil 6.4. Müşterilerin Birbiri İle Etkileşimi Sonucu Firmayı Tanıma Değerlerinde Meydana Gelen Değişim Miktarı

6.2.5. Performans Artışı

Aylık %0,5 ilâ %1,5 artış olduğu takdirde yıllık %10 ilâ %15 artış sağlanmaktadır. Yapılan simülasyonlarda yıllık ortalama artış %13 olarak görülmüştür. Performans artışı yeni üretilen ürünlerin performansında ve müşterilerin performans ihtiyacında standart olay içerisinde dikkate alınmaktadır.

6.2.6. Üretim

“Durumu” durum değişkeni aktif olan ürünler bu işlem için ele alınırlar. Böylelikle projelendirme aşamasındaki, tasfiye sürecinde olan ve tasfiye edilmiş ürünler üretilmezler. Önceki ay yapılan satışın %10 fazlası stokta olacak şekilde üretim miktarı hesaplanır. Eğer üretilecek miktar 1000’in altında ise ürünün ilk 6 ayında olmasına bakılır. İlk 6 ayındaki ürünler için firmanın üretim kapasitesi ve aktif ürün sayısı dikkate alınarak üretim miktarı tekrar hesaplanır. Firma bütçesinin üretim miktarını karşılayıp karşılamayacağı kontrol edilir. Yetersiz kalması durumunda üretilecek miktar azaltılır.

Üretilecek ürün miktarının kesinleşmesinden sonra ürünün “ÜretilenMiktar” durum değişkeni güncellenir. Üretim maliyeti hesaplanarak firma bütçesinden düşülür. Firmanın aylık üretim miktarı artırılır ve ürünün stok miktarı güncellenir.

6.2.7. Tasfiye

Ürünün tasfiye işlemleri durumu aktif olan ürünlere uygulanır. Ürünün tasfiye tarihi gelen veya önceki ayki satışı 1000’in altında olan ve 6 ayını doldurmuş olan ürünler tasfiye sürecine alınır. Ancak firmanın aktif ürün sayısı 3 veya altında ise ürünün tasfiye tarihi gelmesine rağmen ürün tasfiye sürecine alınmayarak ömrü uzatılır.

Tasfiye sürecine alınan ürün artık üretilmez ve fiyatında her ay %5’lik bir indirim yapılır. Ancak bu indirim hiçbir zaman azami indirim miktarını geçemez. Modelde en düşük %15 zararla ürünün satılabileceği kabul edilmiştir.

Tasfiye sürecindeki ürünün önceki ayki satışının veya stok miktarının 50’nin altına düşmesiyle ürün satıştan kaldırılır.

6.2.8. Aylık Müşteri Sayısı

Modelde her müşteri grubunun aylık yapacağı toplam alım sayısını modelleyebilmek için aşağıdaki çizelge hesaplanmıştır. Türkiye’de 2013 yılının son çeyreğinde bilgisayar satışlarının 207 bin (www.genelhaber.org) olduğu dikkate alınır 3 aylık toplamı 206.139 olan modelin gerçek verilerle örtüştüğü görülmektedir.

Müşteri sayısının yıllık %3,8 azalması için standart olayda aylık % 3,5 müşteri gruplarından rastgele azaltılmaktadır.

Çizelge 6.10. Aylık Müşteri Sayısı

Müşteri							
Piyasa hakkındaki bilgisi	Tercihi	Oran (%)	Müşteri Sayısı	Yenileme periyodu (ay)			Aylık Müşteri Sayısı
				Asgari	Azami	Ortalaması	
Bilgili	Performans	10	300.000	18	36	27	11.111
	Fiyat/Performans	25	750.000	24	48	36	20.833
	Fiyat	20	600.000	36	60	48	12.500
Bilgisiz	Fiyat	20	600.000	36	72	54	11.111
	Bilinen Marka	25	750.000	30	84	57	13.158
Toplam		100	3.000.000				68.713

6.2.9. Alım Kriterleri

Müşterinin alım tercihlerinin modellenmesi için aylık alım yapan müşteri sayısını temsil etmek üzere 1000 adet müşteri grubu tanımlanmıştır. Müşteri gruplarının ¼’ü alım için bayi promosyonu olan dağıtımı, geri kalanı yaygın dağıtımı tercih etmektedir. Her müşteri tipinin oranına ve yenileme periyoduna göre aylık alım yapan müşteri sayısı hesaplanmıştır. Buna göre her müşteri grubundaki birey sayısı eşit olacak şekilde, aylık müşteri sayısı dikkate alınarak performans ihtiyacı ve maddi imkânı müşteri tipine göre verilen aralıkta rastgele üretilen değerlerle oluşturulmuştur. Yuvarlamalardan dolayı gruplar 69 kişi olarak belirlenmiştir. Aylık müşteri sayısında meydana gelen % 4,1’lik fazlalık ihmal edilmiştir.

Müşterinin alım tercihinin belirlenebilmesi için ürünler, müşteri tiplerine göre farklı kriterlerle puanlanarak sıralanmaktadır. Her ürünün puanı fiyat, performans, fiyat/performans oranı ve firmanın tanınması puanlanarak, farklı müşteri tipleri için

farklı katsayılarla çapılarak toplamı hesaplanmıştır. Kriterlere göre hesaplanan puanlar aşağıdaki gibidir;

$$Fiyat Puanı = 100 - 100 \times \frac{Fiyat - Fiyat_{min}}{Fiyat_{max} - Fiyat_{min}}$$

$$Performans Puanı = 100 \times \frac{Performans - Performans_{min}}{Performans_{max} - Performans_{min}}$$

f/p : Fiyat/Performans Oranı

$$f/p = \frac{Fiyat}{Performans^{\frac{3}{2}}} \times 100$$

$$f/p Puanı = 100 - 100 \times \frac{f/p - f/p_{min}}{f/p_{max} - f/p_{min}}$$

Tanınma puanı, müşterinin firmayı tanınması kadardır. Tüm kriter puanları 0 ilâ 100 arasında olacak şekilde tasarlanmıştır. Müşteri tiplerine göre her kriter puanı için kullanılacak katsayı Çizelge 6.11'deki gibidir.

Çizelge 6.11. Ürün Tercihinde Puan Katsayıları

Müşteri			Seçim Kriteri				
Piyasa hakkındaki bilgisi	Tercihi	Oran %	Tercih dağılımı	Performans	Fiyat/Performans	Fiyat	Tanınma
Bilgili	Performans	10	İlk 5 ürün	5	2	1	1
	Fiyat/Performans	25	İlk 5 ürün	4	5	1	1
	Fiyat	20	İlk 8 ürün	3	4	3	2
Bilgisiz	Fiyat	20	İlk 8 ürün	1	3	4	2
	Bilinen Marka	25	İlk 5 ürün	1	3	2	5

Her müşteri grubunda birden fazla birey bulunduğu ve aynı gruptaki bireylerin de farklı tercihte bulunma ihtimalleri olduğundan grup içinde farklı tercihlerin görülmesi gerekmektedir. Müşteri tiplerinin ürün seçim kriterlerinde belirtilen ilk beş

veya sekiz ürün arasından seçer ifadelerinin karşılanması ve sıralamada daha önde olan ürünün daha çok seçilmesi maksatlarıyla her ürün sıralamadan kendinden sonra gelenden %50 fazla satılır kabul edilmiştir. Buna göre ilk beş ve ilk sekiz ürün arasından seçim yapacak müşteri sayıları Çizelge 6.12'deki şekilde hesaplanmaktadır.

Çizelge 6.12. Ürün Tercihinde Müşteri Oranlarının Hesaplanması

Sıralama	Sayı	Oran	Sıralama	Sayı	Oran
1	506	38,39%	1	1709	34,69%
2	338	25,59%	2	1139	23,12%
3	225	17,06%	3	759	15,42%
4	150	11,37%	4	506	10,28%
5	100	7,58%	5	338	6,85%
Toplam	1319		6	225	4,57%
			7	150	3,05%
			8	100	2,03%
			Toplam	4926	

Çizelgede sıralamanın sonundaki ürün 100 müşteri alım için tercih edecek kabul edilip bir önündeki her üründe alım tercihi %50 fazla kabul edilmiştir. Hesaplanan oran ise sıralamadaki ürünün alım tercihinin oluşan toplama göre oranıdır. Müşteri gruplarında bu toplam başta 69 olacak ve zamanla azalacaktır.

Buna göre başlangıçta oluşacak değerler aşağıdaki gibidir.

Çizelge 6.13. Başlangıçta Oluşacak Ürün Tercih Oranları

Sıralama	Sayı	Oran	Sıralama	Sayı	Oran
1	26	38,39%	1	24	34,69%
2	18	25,59%	2	16	23,12%
3	12	17,06%	3	11	15,42%
4	8	11,37%	4	7	10,28%
5	5	7,58%	5	5	6,85%
Toplam	69		6	3	4,57%
			7	2	3,05%
			8	1	2,03%
			Toplam	69	

Tüm tedavüldeki ve müşterinin alım gücü dâhilindeki ürünler müşterinin alım tercihi için puanlanır ve sıralamaya tabi tutulurlar. Müşterinin performans ihtiyacını karşılamayan ürünler sadece performans puanı ile sıralanırlar. Böylece bu ürünler

sıralamanın sonunda olurlar ve başka uygun ürün olmadığı takdirde müşterinin en üst performansa sahip yetersiz ürünü seçmesine imkân verilir. Burada müşterinin alım için uygun ürün olmadığında ihtiyacına en yakın ürünü seçeceği varsayılmıştır. Eğer yetersiz stok nedeni ile müşterinin tercih ettiği ürün yoksa kendine göre sıralanan ürünlerden sırayla mevcut stok miktarı yeterli ise alım yapar.

Alım işleminde ürünün stok miktarı, aylık satış miktarı, toplam satış miktarı, üründen elde edilen toplam kâr durum değişkenleri güncellenir. Satıştan elde edilen gelir firmanın bütçesine eklenir. Tüm alım işlemleri standart olay içerisinde yer alır.

6.2.10. Maliyet ve Fiyat Güncellemeleri

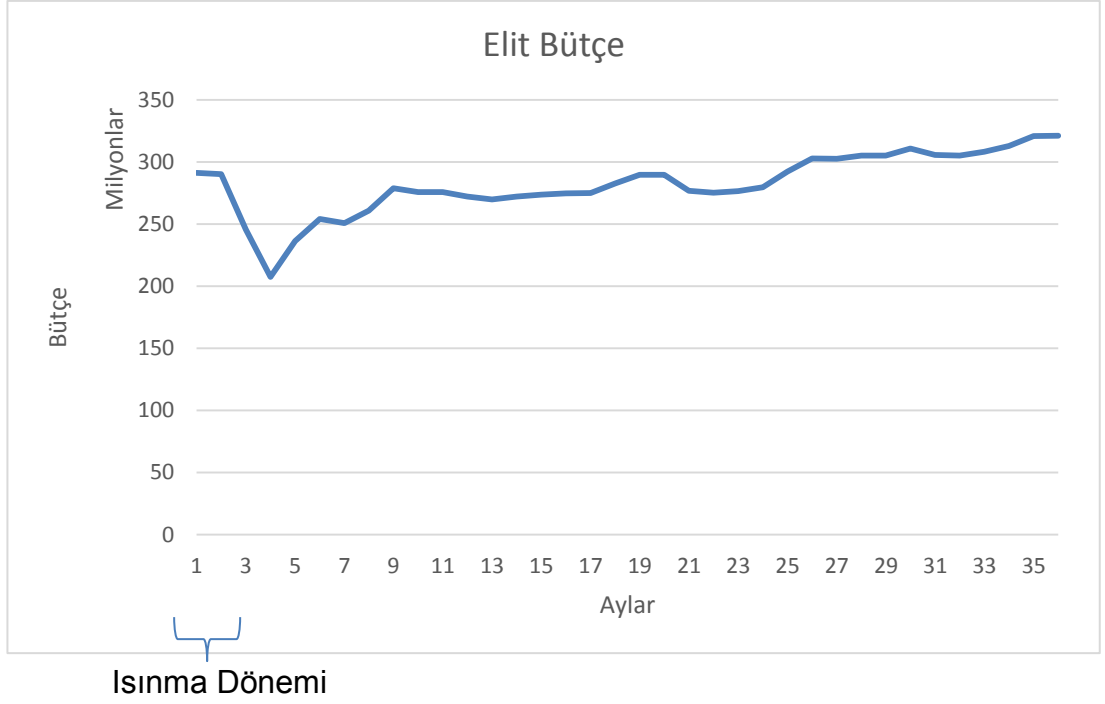
Performans artışı ile birlikte maliyetler düştüğünden 3 ayda bir %3,5 azaltılan “Maliyet Çarpanı” durum değişkeni kullanılmıştır. Tüm firmalar da ürünlerini aynı periyotta maliyet ve fiyat güncellemesi yaparlar.

6.3. ÇÖZÜM SONUÇLARI

Model üç farklı seviyede firma için tekrar çözülmüştür. Her üç firmanın bütçe miktarı dikkate alındığında ortalama %50’lik bir artış olduğu görülmüştür. Bununla birlikte üretimi devam eden ve stokta bulunan ürünler mevcuttur. Bütçedeki artış hesaplanırken 3. aydaki bütçe miktarı esas alınmıştır. Çünkü firmanın mevcut ürünleri model başlangıcında projelendirilip üretilmeye başlandığından bu bölüm simülasyonun ısıtılma süreci olarak kabul edilmiştir. Isıtma süreci ihmal edilse dahi bütçelerin ortalama artış oranı %27,7 olmaktadır.

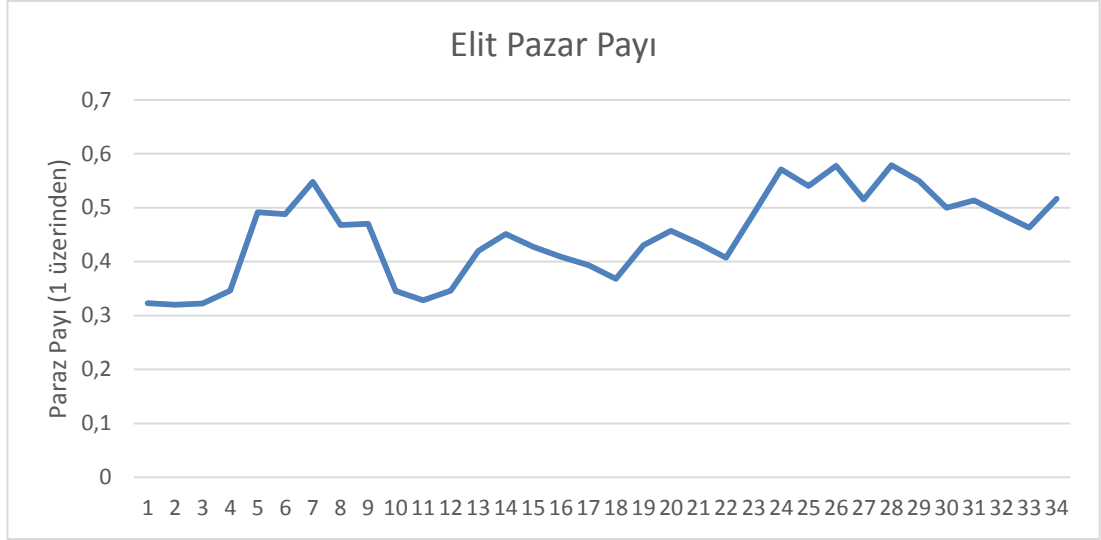
6.3.1. Lider Firma Çözümü (Elit)

Lider firma durumunda olan Elit firması adına yapılan çözümde, firma 298 milyon TL ile simülasyona başlamış, sabit olayların çalışması sonucu bütçesi, 207,5 milyon TL’ye düşmüş ve simülasyona göre 3 yılın sonunda 321,1 milyon TL’lik bir bütçe elde etmiştir. Buna göre %54,75’lik bir büyüme gerçekleştirmiştir.

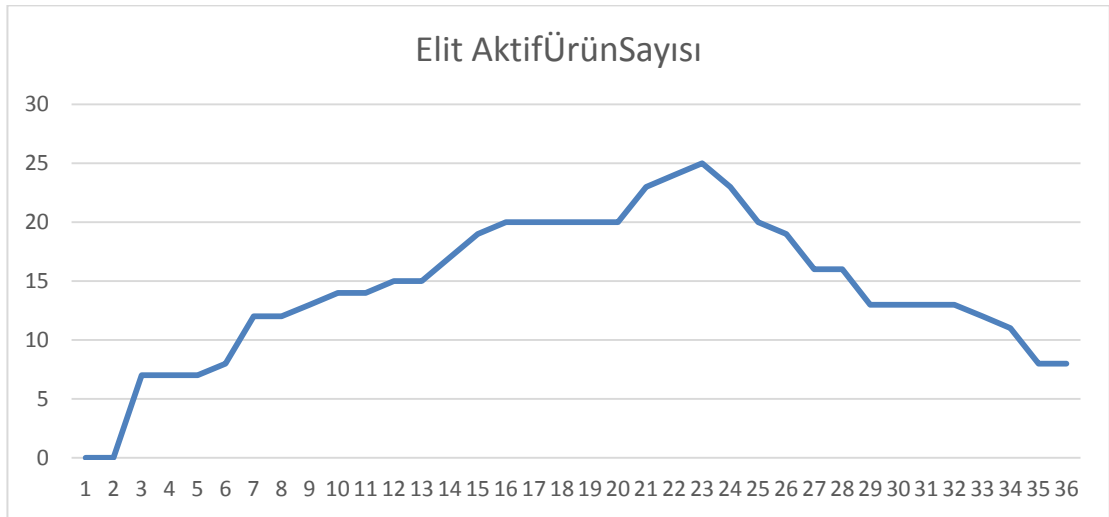


Şekil 6.5. Elit Bütçe

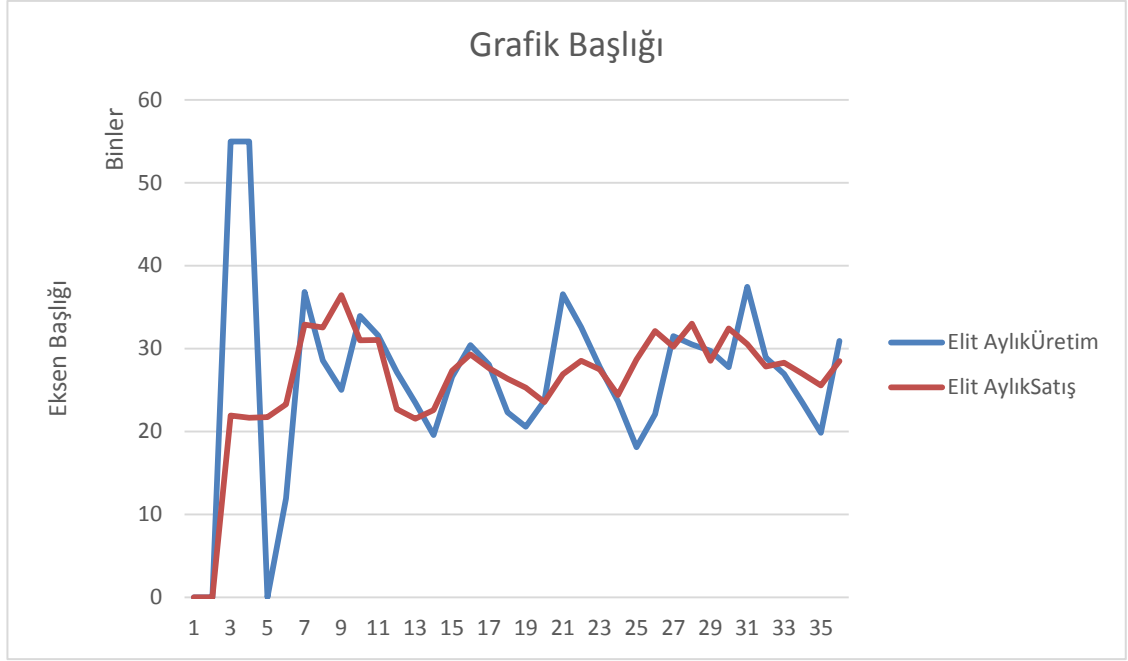
Elit firmasının pazar payını dikkate aldığımızda %59,88 artışla bütçede gösterilen verileri onaylar niteliktedir. Aktif ürün sayısını (firmanın ilgili ayda üretim ve satışını yaptığı ürün sayısı) incelediğimizde ise 23. ayda azami miktarı olan 25 adet ürüne çıktığı ve bundan sonra düşüşe geçtiği görülmüştür. Kriterler içerisinde 36. ayın sonunda azami bütçe miktarına ulaşılacak istenmesi nedeni ile algoritma son yılda yapılacak yeni ürün geliştirmelerini gereksiz görmüştür. Bu nedenle iki yıllık bir stratejinin belirlenmesi için 36 aylık bir çözüm istenmiştir.



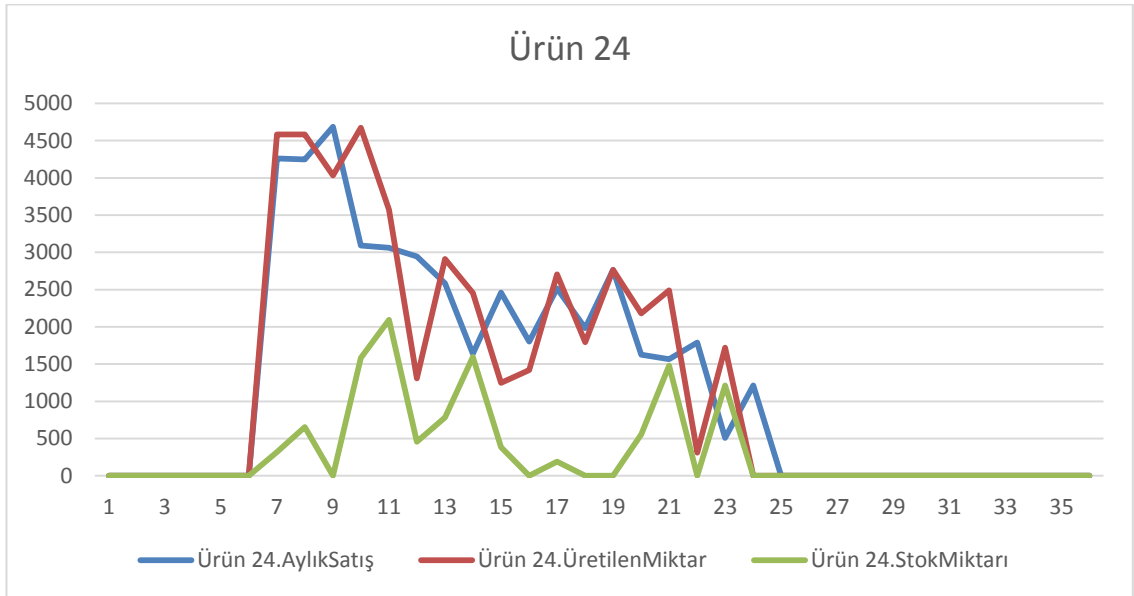
Şekil 6.6. Elit Pazar Payı



Şekil 6.7. Elit Aktif Ürün Sayısı



Şekil 6.8. Elit Aylık Üretim ve Satış Miktarları

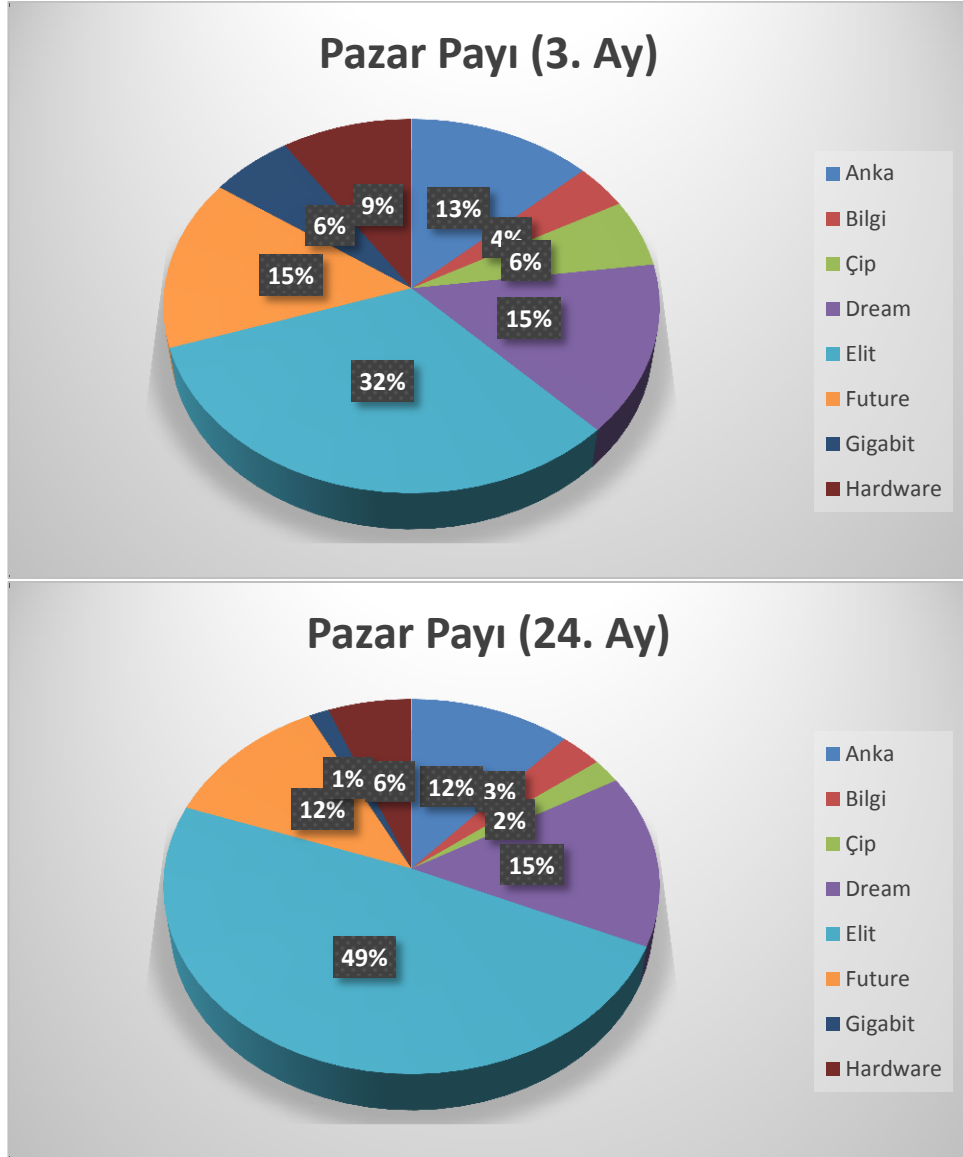


Şekil 6.9. Ürün 24, Üretim, Stok ve Satış Miktarları

Şekilde altıncı ayda üretime başlayan ve 24. ayda tasfiye edilen Elit firmasına ait ürünlerden biri olan “Ürün 24”e ait üretim, stok ve satış miktarları görülmektedir.

Elit firmasının aylık üretim ve satış grafiğine baktığımızda ikinci yıl içerisinde bir dengeye ulaştığı görülmektedir. Üretim miktarının ilk 6 ayda dalgalı olmasının sebebi

ise başta sıfır olan ürün miktarından sonra ilk eklenen ürünlerin önceki ayki satış miktarı olmadığından üretim kapasitesini dolduracak şekilde planlanmasıdır. Başta tam kapasiteyle yapılan üretim sonucu beşinci ayda yeni ürüne ihtiyaç duyulmamış ve yedinci aydan sonra olması gereken seviyeye gelmektedir.



Şekil 6.10. Pazar Paylarının Karşılaştırılması

Pazar paylarını karşılaştırdığımızda Elit firması 24. aya kadar pazar payını önemli ölçüde artırmasına rağmen diğer firmaların pazar paylarının düştüğü veya ancak korunduğu görülmüştür.

Elit için algoritma ile bulunan en iyi senaryo aşağıdaki çizelgedeki gibidir. Elit firmasına ilk ayda sabit olarak eklenen 7 adet ürünün projelendirilmesi ve projeli ürünlerin üretime başlama olayları çizelgede yer almamaktadır.

Çizelge 6.14. Elit Senaryo

Aylar	Olaylar				
0	Reklam TV	Reklam Internet			
1	Reklam Dergi	Reklam Duyurum	Reklam Internet	Kampanya	Promosyon
2	Reklam TV	Reklam Dergi	Reklam Duyurum	Kampanya	Promosyon
3	Reklam Dergi	Reklam Dergi	Reklam Outdoor		
4	Reklam Internet	Reklam Internet	Ürün Projelendir Üst		
5	Reklam TV	Ürün Projelendir Alt	Ürün Projelendir Üst	Ürün Projelendir Üst	Ürün Projelendir Üst
6	Reklam Dergi				
7	Reklam Dergi	Ürün Projelendir Orta	Kampanya	Kampanya	
8	Reklam Dergi	Ürün Projelendir Alt			
9	Reklam Dergi	Prestij			
10	Reklam Outdoor	Reklam Outdoor	Ürün Projelendir Alt	Prestij	
11	Reklam TV	Reklam Dergi	Reklam Duyurum		
12	Ürün Projelendir Orta	Ürün Projelendir Üst	Kampanya		
13	Reklam Internet	Ürün Projelendir Üst	Ürün Projelendir Üst		
14	Reklam Outdoor	Ürün Projelendir Orta			
15	Reklam Dergi	Reklam Dergi	Promosyon		
16					
17	Reklam Internet				
18	Kampanya				
19	Reklam Outdoor	Ürün Projelendir Alt	Ürün Projelendir Alt	Ürün Projelendir Orta	Prestij
20	Reklam Outdoor	Reklam Outdoor	Reklam Outdoor	Ürün Projelendir Orta	
21	Reklam Internet	Reklam Internet	Ürün Projelendir Orta	Kampanya	
22	Reklam Internet	Reklam Internet	Reklam Internet		
23					

Çizelge 6.14. (Devam)

24	Reklam Dergi	Reklam Outdoor	Ürün Projelendir Alt		
25					
26	Kampanya				
27	Reklam TV	Reklam TV	Reklam Duyurum	Reklam Internet	Reklam Outdoor
28	Promosyon				
29	Kampanya	Prestij			
30	Reklam Dergi	Reklam Duyurum	Reklam Internet	Reklam Internet	Prestij
31	Reklam Outdoor	Prestij			
32	Reklam Dergi				
33	Reklam Duyurum	Reklam Internet	Reklam Internet		
34	Reklam Internet				
35					

Senaryoda dikkat çeken özelliklerden bazıları şunlardır;

Elit yedinci aya kadar 4 kez kampanya yaparak fiyat indiriminde bulunmuş ve pazar payını önemli ölçüde artırmıştır. Dokuz ve onuncu aylarda ise prestij göstergesi olarak fiyat artırımına gitmiştir. Son fiyat indirimi 21. ayda olup bundan sonra iki kez fiyat artırımına gitmiştir.

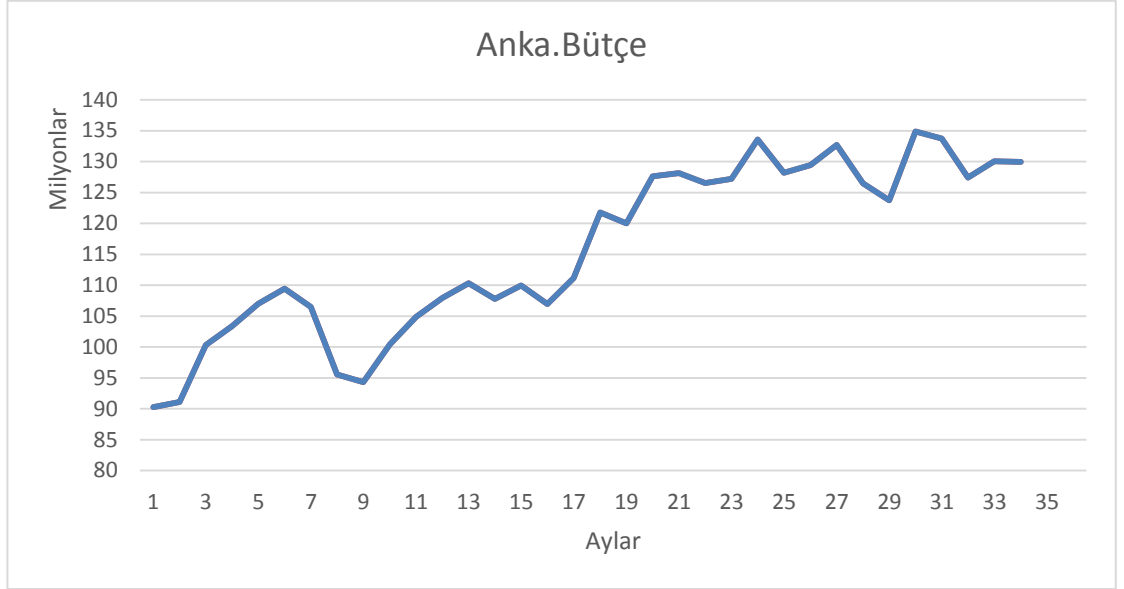
Reklam faaliyetlerinin başlangıçta daha yoğun olmak üzere tüm senaryo boyunca devam ettiği gözlemlenmiştir.

Yeni ürünlerin projelendirilmeleri ikinci yıla kadar devam etmiş, bütçenin 36. ayda yüksek olması için en son 24. aydan sonra yeni ürün projelendirilmemiştir. Yeni ürün seviyeleri karşılaştırıldığında toplamda sayılar eşit veya yakın olsa da, başlangıçta üst seviye ürün ağırlıktayken, 14. ay itibariyle yalnızca orta ve alt seviye ürün projelendirilmiştir.

Promosyon anlaşması düzenli olarak yapılmıştır. Sadece iki aylık promosyon anlaşması bulunmayan dönem mevcuttur.

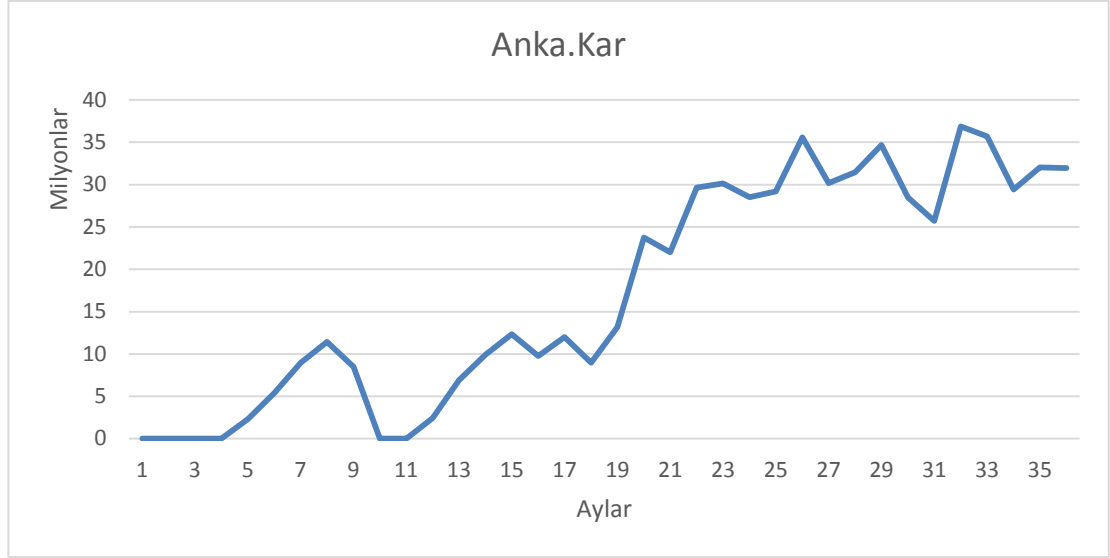
6.3.2. Orta Seviye Firma Çözümü (Anka)

Anka, model başlangıcında 98 milyon TL bütçeye sahipken, ısınma süresince bütçesi 90,3 milyon TL'ye düşmüş, en iyi çözümün simülasyonu sonunda %43,96'lık büyüme ile 130 milyon TL'lik bir bütçeye ulaşmıştır.



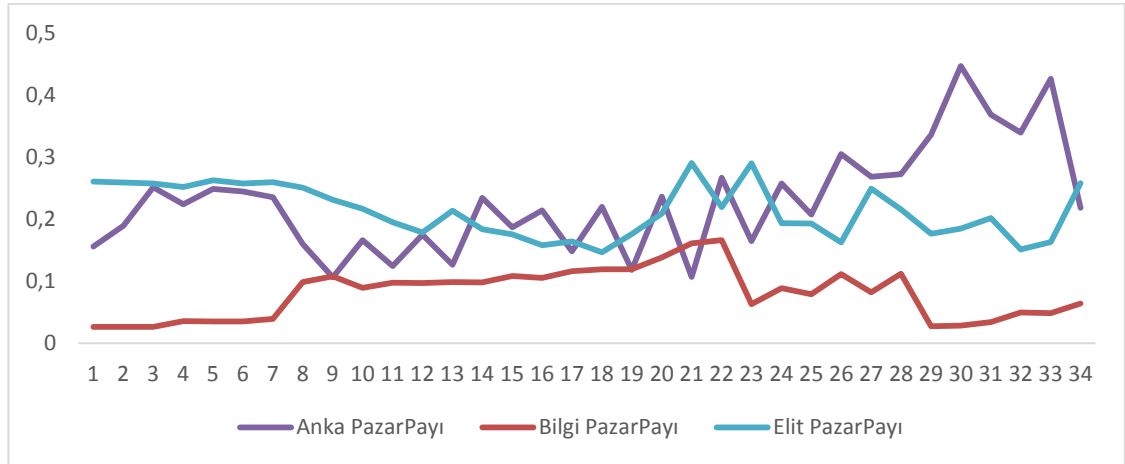
Şekil 6.11. Anka Bütçe

Anka firmasının bütçe ve kâr grafikleri **Şekil** ve **Şekil** 'de gösterilmiştir. Modelde belirtilen amaç kâr iken, kârın bütçenin ilk değerinden olan farkı olarak tanımlanması nedeniyle aynı grafik ve değerler ortaya çıkmaktadır. Bu nedenle modelde kâr yerine bütçe değeri tercih edilmiştir.

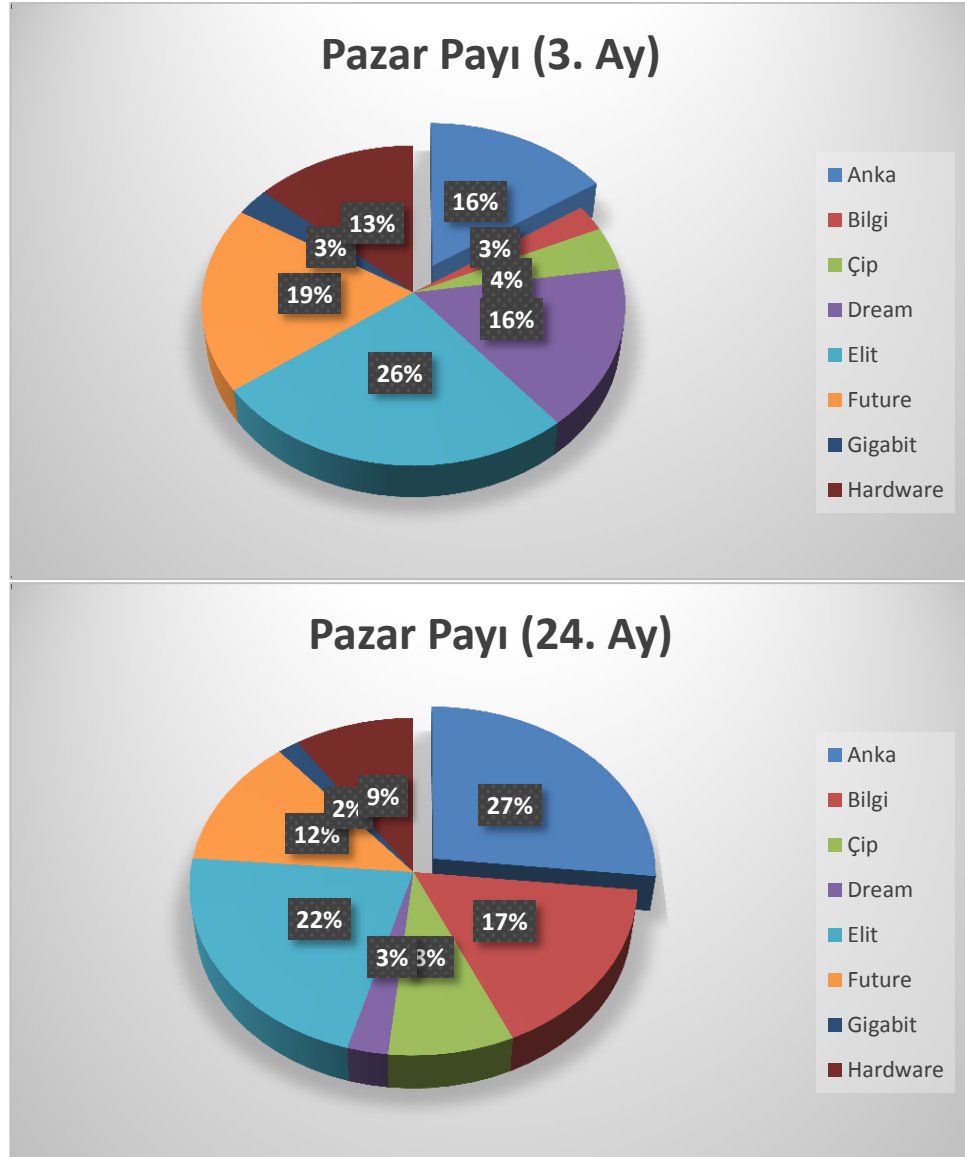


Şekil 6.12. Anka Kâr

Firmaların pazar paylarını karşılaştırıldığında iki yılın sonunda Anka'nın %16'dan %27'ye çıktığı görülmektedir. **Şekil** 'te Bilgi firmasının pazar payındaki artış dikkat çekmektedir. Firmanın ürünleri incelendiğinde sekizinci ayda üretimine başladığı üst seviye bilgisayarın yüksek tercih oranı nedeniyle bu başarıyı elde ettiği, ancak ürünün tasfiyesi ile birlikte bu başarısının kaybolduğu görülmektedir. (Bkz. **Şekil**).



Şekil 6.13. Pazar Payları



Şekil 6.14. Pazar Paylarının Karşılaştırılması

Anka için bulunan en iyi senaryo aşağıdaki çizelgedeki gibidir. Anka firmasına ilk ayda sabit olarak eklenen 2 adet ürünün projelendirilmesi ve projeli ürünlerin üretime başlama olayları çizelgede yer almamaktadır. Elit firmasından farklı olarak Anka'nın ilk iki yıl sadece 2+6 adet ürün projelendirdiği görülüyor. Sekiz adet üründen dört adedinin tüm üretiminin aynı ay içinde satıldığı ve firmaya yüksek kâr getirdiği tespit edilmiştir. Ancak 24. aydan sonra artan ürün sayısı ve üretime rağmen yeni ürün projelendirme maliyetleri nedeni ile kâr oranı azalmıştır.

Çizelge 6.15. Anka Senaryo

Aylar	Olaylar					
0	Reklam TV	Reklam Dergi	Promosyon			
1	Reklam Duyurum					
2	Reklam TV	Reklam TV	Reklam TV	Ürün Geliştir 2	Ürün Projelendir Üst	Promosyon
3	Reklam Dergi	Ürün Projelendir Üst				
4	Reklam Duyurum					
5	Reklam TV	Reklam TV	Reklam TV	Reklam Dergi	Ürün Projelendir Alt	
6	Reklam TV					
7	Reklam TV	Reklam Dergi	Reklam Duyurum			
8	Reklam TV	Reklam TV	Reklam TV	Reklam Duyurum	Reklam Duyurum	Reklam Duyurum
9	Reklam TV	Reklam Dergi	Kampanya			
10						
11	Reklam Dergi	Reklam Dergi				
12	Reklam Duyurum	Reklam Duyurum	Kampanya			
13	Reklam Dergi	Reklam Duyurum	Reklam Internet	Reklam Outdoor	Reklam Outdoor	
14	Reklam TV	Reklam TV	Reklam Dergi	Ürün Projelendir Alt	Ürün Projelendir Orta	
15	Reklam Internet	Kampanya	Prestij			
16	Reklam TV	Reklam Dergi	Reklam Outdoor			
17	Reklam Duyurum	Promosyon				
18	Reklam Dergi	Reklam Dergi				
19						
20	Reklam Dergi					
21	Reklam Dergi	Reklam Duyurum				
22	Reklam Dergi	Reklam Internet	Reklam Internet	Reklam Internet	Ürün Projelendir Orta	Kampanya
23	Reklam Dergi	Reklam Duyurum	Reklam Duyurum	Reklam Internet	Reklam Internet	

Çizelge 6.15. (Devam)

24	Reklam TV	Reklam Dergi	Ürün Projelendir Alt	Ürün Projelendir Orta	Kampanya	
25	Reklam TV	Reklam Outdoor				
26	Reklam Dergi	Reklam Dergi	Reklam Internet	Reklam Internet	Ürün Projelendir Alt	
27	Reklam TV					
28	Reklam TV	Reklam Duyurum	Ürün Projelendir Üst			
29	Reklam Duyurum	Ürün Projelendir Orta	Ürün Projelendir Üst	Ürün Projelendir Üst		
30	Reklam Dergi	Reklam Duyurum				
31	Reklam Dergi	Reklam Duyurum	Ürün Projelendir Orta			
32	Reklam TV	Reklam Duyurum	Reklam Internet			
33	Reklam TV	Reklam Duyurum	Ürün Projelendir Orta			
34	Reklam TV	Reklam Internet	Kampanya			
35						

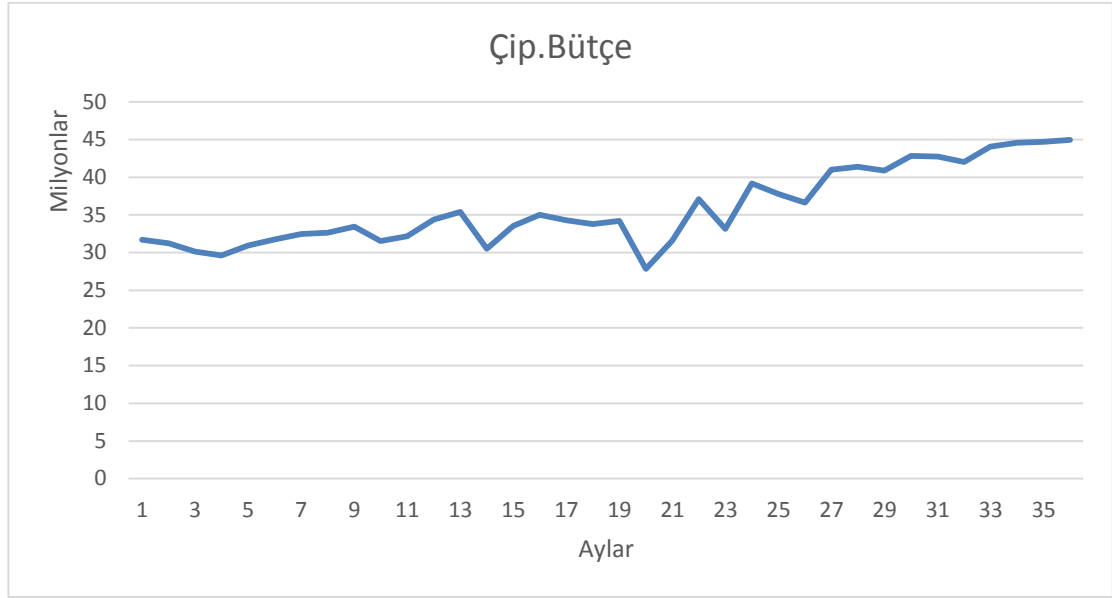
İkinci yıldan sonra ürün projelendirme sayısının artmasına neden olarak diğer firmaların 21. ayda aynı anda yeni ürün çıkarmaları nedeniyle Anka'nın da 24 ilâ 29. aylar arası 7 adet yeni ürün projelendirme ihtiyacı duyması olarak değerlendirilmiştir.

Anka firmasının çözümünde bulunan 79 adet reklam olayının %56'sı ilk 18 ayda gerçekleşmiştir. Promosyon anlaşması ise 3 ay hariç 29. aya kadar devam etmektedir.

Ürün geliştirme faaliyeti yalnızca bir kez planlanan senaryoda 6 kez indirim ve bir kez fiyat artırımı mevcuttur. Ancak 15. ayda bir kampanya ve prestij olayı bir arada bulunduğundan birbirlerinin etkisini neredeyse yok etmektedirler. Sonuç küçük bir indirim olarak gerçekleşmektedir. Olaylar münferit olarak değil, her birim zamandaki olayların toplu etkisi olarak planlandığından ve yeni kromozoma topluca aktarıldığından birbirlerinin zıddı olay iki olayın aynı gende bulunmaları mümkün olmuştur. Sonuçların incelenmesi aşamasında bu iki olay senaryodan çıkarılabilirler.

6.3.3. Alt Seviye Firma Çözümü (Çip)

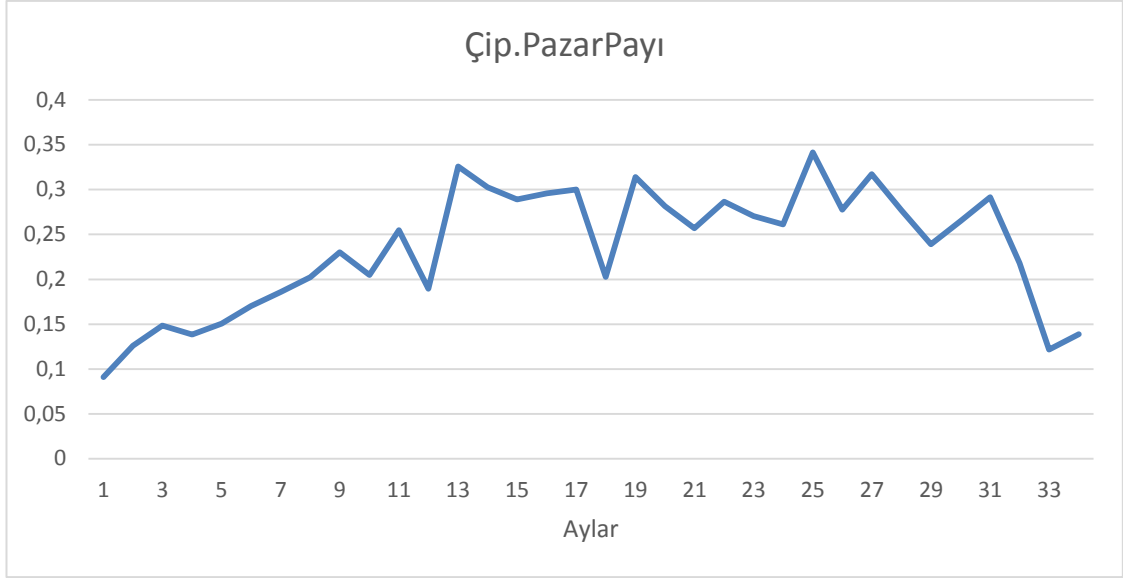
Çip, model başlangıcında 32 milyon TL bütçeye sahipken, ısınma süresince bütçesi 29,6 milyon TL'ye düşmüş, en iyi çözümün simülasyonu sonunda %40,48'lik büyüme ile 45 milyon TL'lik bir bütçeye ulaşmıştır.



Şekil 6.15. Çip Bütçe

Pazar payı ise 25. aya kadar yükselmiş, müteakibinde ürün projelendirilmemesi nedeni ile pazar payı düşmüştür. Çip'in senaryosunda başlangıç ürünü ile birlikte 11 ürününü 10'u ilk 24 ayda, sonuncusu 25. ayda projelendirilmektedir. Senaryoda 6 adet üst sınıf, 4 adet alt sınıf ürün projelendirilirken orta sınıf ürün projelendirilmemiştir. Çip firmasına ait ürünlerde stokta bekleyen ürün sayısının diğer firma çözümlerine göre daha az olduğu dikkat çekmiştir.

Ürün geliştirme faaliyetleri genel olarak hiçbir çözümde tercih edilmemiştir. Buna neden olarak modelde gösterilen performans artışındaki getirinin maliyetteki artışı karşılamaması olarak değerlendirilmektedir.



Şekil 6.16. Çip Pazar Payı

Alt seviye firma olan Çip için yüksek bütçeli TV reklamına senaryoda hiç yer verilmezken diğer reklam sayıları da diğer firmaların çözümlerine göre daha geride kalmaktadır.

Promosyona diğer firma çözümlerinde olduğu gibi önem verilse de 7 aylık boşluk ile diğerlerinin gerisinde kalmaktadır.

Çip firması için bulunan en iyi senaryoda fiyat artırımını hiç görülmezken, yaklaşık 6 ayda bir indirim uygulanmaktadır. 19 ilâ 21. aylarda ise üç ay arka arkaya indirim yapılmıştır.

Çizelge 6.16. Çip Senaryo

Aylar	Olaylar				
0	Reklam Dergi	Reklam Dergi	Reklam İnternet	Ürün Projelendir Üst	Kampanya
1	Reklam Outdoor				
2	Reklam Duyurum	Ürün Projelendir Üst	Kampanya		
3	Promosyon				
4	Reklam Dergi				
5	Reklam Dergi	Reklam Duyurum	Reklam İnternet		

Çizelge 6.16. (Devam)

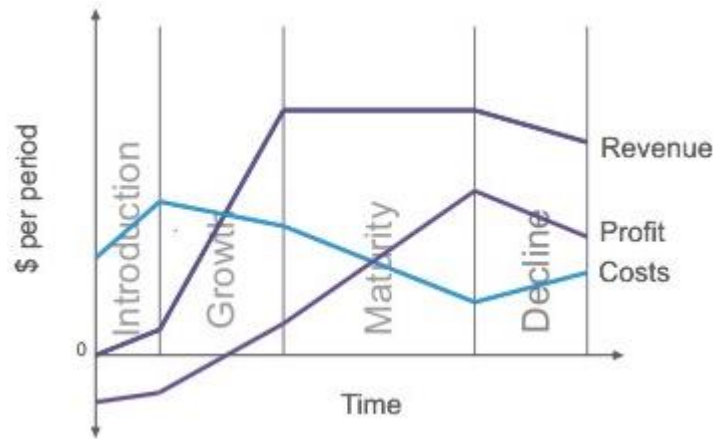
6	Reklam Duyurum	Ürün Projelendir Alt	Kampanya		
7					
8	Ürün Projelendir Üst				
9	Reklam Duyurum				
10					
11	Reklam Internet	Reklam Internet	Ürün Projelendir Alt		
12	Reklam Dergi	Reklam Dergi	Kampanya		
13	Reklam Dergi	Ürün Projelendir Alt	Ürün Projelendir Üst		
14	Ürün Projelendir Üst				
15	Reklam Dergi	Reklam Duyurum			
16	Reklam Dergi	Promosyon			
17	Reklam Internet				
18	Reklam Dergi	Reklam Duyurum			
19	Reklam Dergi	Ürün Projelendir Alt	Kampanya		
20	Reklam Duyurum	Kampanya			
21	Reklam Dergi	Reklam Dergi	Reklam Internet		
22					
23	Reklam Dergi				
24	Ürün Projelendir Üst				
25					
26	Reklam Dergi				
27					
28	Reklam Duyurum	Reklam Internet			
29	Reklam Duyurum	Reklam Internet			
30	Reklam Dergi				
31	Reklam Internet				
32	Promosyon				
33	Reklam Dergi	Reklam Duyurum	Reklam Internet		
34	Reklam Dergi				

6.3.4. Ürün Hayat Dönemi

Ürün hayat devreleri bir ürünün yaşam eğrisinin anlaşılmasını kolaylaştıran dört evreden oluşur (<http://www.nku.edu>, <http://www.marketingteacher.com>, <http://www.mindtools.com>, <http://www.quickmba.com>);

- Giriş dönemi (Introduction)
- Gelişme dönemi (Growth)
- Olgunluk dönemi (Maturity)
- Gerileme dönemi (Decline)

Bazı kaynaklarda öncesinde ürünün geliştirilmesi (Development) ve sonrasında ölümünü de (Withdrawal) içeren grafikler gösterilse de genellikle dört ana evre olarak ele alınmaktadır.



Şekil 6.17. Ürün Hayat Dönemi (<http://www.mindtools.com/>)

Ürün ömrü ve evreleri için kesin bir süre yoktur, ürüne göre değişebilir. Bazı ürünlerde bu birkaç ay olabileceği gibi başka bir üründe yıllar olabilir.

Bu grafik üzerinde ürün hareket ettiğinde, rekabet, dağıtım, fiyatlandırma, tutundurma ve pazar bilgisiyle ilgili stratejilerin periyodik olarak değerlendirilmesi ve muhtemelen değiştirilmesi gereklidir. Ürün yaşam eğrisini kullanmakla pazarlama yöneticileri, kârlı olabilecek ürünleri daha iyi hale getirerek ürünün varlığını sürdürmesini sağlayabilir, kârsız ürünleri ise sonlandırabilirler.

6.3.4.1. Giriş Dönemi

Yaşam eğrisinin giriş aşaması, ürünün pazar alanında ilk kez yer aldığı, başlangıçta satışların çok az ve karlılığın negatif olduğu aşamadır. Bu aşamada müşteri henüz ürünü tanımaz. Dolayısıyla, ürünün kârdan çok tanıtılması önemlidir.

Başlangıçta fiyatlandırma ile ilgili iki strateji mevcuttur. Eğer rakip sayısı azsa önce yüksek fiyat verilip zamanla fiyatlar azaltılır. Ancak fiyatlar müşteri için çok yüksek bulunursa ürünün hayat dönemi başlamadan sona erer. İkincisi, başlangıçta düşük fiyatlandırılıp zamanla yükseltmektir. Bu yöntem rakip sayısının çok olması durumunda kullanılır. Bu yöntemin dezavantajı ise kârın yetersiz kalabilmesidir.

6.3.4.2. Gelişme Dönemi

Bazı ürünler giriş aşamasından öteye gidemezler. Bu dönemde malın rakiplerden daha iyi olduğunu ispatlamak için çok fazla reklam harcaması yapılır. Gelişme dönemi süresince, satışlar hızlı bir şekilde artar, mal giderek aşağı gelir gruplarınca da satın alınmaya başlanır ve kârlılık en üst düzeye ulaşır. Bunun ardında da kârda gerileme başlar. Bu, pazarın büyümesi anlamındadır. Ürünün pazarda benimsenmeye başlanmasından sonra birçok rakip işletme pazara girmeye çalışır. Gelişme aşaması ürünün devamlılığı açısından oldukça kritik bir aşamadır, çünkü bu gelişme süresince ürünün başarısına karşı görülen rekabetçi tepkiler ürünün yaşam umudunu etkiler.

6.3.4.3. Olgunluk Dönemi

Olgunluk dönemi süresince satış artışı azalan oranda devam eder, devamında durağanlaşır ve zamanla azalmaya başlar. Bu aşamada ürün tüm gelir grupları tarafından satın alınır ve genellikle ürünün en uzun süren evresidir. Olgunluk aşaması, pazarda pek çok rakip markanın bulunmasından ötürü yoğun rekabetle karakterize edilir. Rakip işletmeler, kendi ürünlerindeki farklılıklar üzerinde dururlar.

6.3.4.4. Gerileme Dönemi

Ürünün yaşam eğrisinin son aşaması olan gerileme döneminde ürün satışları ve kârlılığı, sağlanan teknolojik ilerlemeler, toplumsal eğilimlerde görülen değişimler vb.

nedenlerle artan hızda azalmaya devam eder. Bu durumda işletme, kârlılığını olumsuz yönde etkileyen malları ya doğrudan mal karmasından çıkarır ya da ürüne yapılan harcamaları kısıtlar, dağıtım ağını daraltır. Aslında bu dönemde, sektörel olarak tüm satışlar azalır. Satışlar artan oranda azalabildiği gibi, birden bire de kesilebilir. Rakip işletmelerin büyük bir kısmı gelişme aşamasında pazardan çekildiği için doğrudan rekabetle karşılaşmaz.

6.3.5. Firmaların Beklenen Davranışları ile Sonuçların Karşılaştırılması

Genel olarak masaüstü bilgisayarı ele alındığında mevcut durumda olgunluk döneminden gerileme dönemine bir geçiş aşamasında olduğu söylenebilir.

Reklam faaliyetleri firmanın tanınma oranını artırmakta, böylelikle müşterilerin ürün alımlarında tercih sebebi olmaktadır. Ürünleri tercih edilen firmanın pazar payı ve dolayısı ile kârı artacağından olumlu sonuç beklentisi olan olaylardır. Ancak tanınmanın etkisinin daha fazla olması simülasyon boyunca daha yüksek değerlere sahip olması ile mümkündür ki reklam olaylarının senaryonun başında olmaları bunu sağlayacaktır. Her üç firma sonucu incelendiğinde reklam olaylarının dağılımının senaryonun başından sonuna doğru azalarak devam ettiği gözlemlenmektedir. Düşük bütçeli Çip firması için yüksek bütçeli reklam olayları kullanılmamıştır.

Senaryolar ürün sayısı açısından incelendiğinde daha çok ürün projelendirmenin daha çok tercih edilme sebebi olacağı düşünülebilir. Bu maksatla hedef kriteri olarak bütçenin yanında ürün sayısının çokluğu da eklenerek ürün sayısının firma kârına olan etkileri incelenmek istenmiştir. Yapılan çözüm sonucunda projelendirme maliyetleri nedeniyle bütçenin daha da düştüğü, üretim kapasitesinin kısıtlanması ile de üretim ve dolayısıyla satışların beklendiği gibi artmadığı görülmüştür. Bu nedenle bulunan çözümlerdeki ürün projelendirme sayıları uygun bulunmuştur.

Firmalar için yapılan üç yıllık optimizasyonlarda kriter olarak bütçenin alınması nedeni ile çözümlerde üçüncü yılda yeni ürün projelendirme tercih edilmemiştir. Çünkü belirli bir maliyetle yapılacak projelendirme sonucunda üçüncü yıl için ürünün yapılacak satışlarla kendini amorti etme şansı düşüktür. Çözümlerde son aylarda planlanan az sayıdaki yeni ürün projelendirme olayları firmanın beklenen davranışlarına uygun düşmektedir.

Diğer firmaların hareket tarzlarına göre incelenen firmanın da bazı önlemler alması beklenmektedir. Anka'nın çözümünde diğer firmalar 21. ayda aynı anda yeni ürün projelendirmişler, müteakibinde senaryo sonuna yaklaşılmış olmasına rağmen 24 ilâ 29. aylar arasında 7 adet yeni ürün projelendirilmiş, ortalama 6 olan aktif ürün sayısını bu dönemde 11'e çıkarmıştır. Oysa diğer firma çözümlerinde aktif ürün sayısı 24. aya kadar artmakta, devamında ise düşüşe geçmektedir.

Firma çözümlerinde aktif ürün sayıları karşılaştırıldığında böyle bir kısıt konmamasına rağmen firmanın bütçesi ve üretim kapasitesinin izin verdiği oranda gerçekleşmektedir. Böylelikle daha büyük firmaların çözümünde daha çok yeni ürün projelendirme planlanmaktadır.

Firma büyüdükçe prestij göstergesi olarak daha yüksek fiyatlı ürün satılmaktadır. Firma çözümleri karşılaştırıldığında lider firmanın dört kez fiyat artırımına gittiği ve ürünlerinin daha pahalıya sattığı gözlemlenmiştir. Orta seviye firma olan Anka bir kez fiyat artırımını yaparken, alt seviye firma olan Çip ise hiç fiyat artırımını yapmamıştır.

Lider firma ilk yedi ayda indirim yapmış ve pazar payını artırdıktan sonra fiyatlarını tekrar yükseltmiştir. Firma küçüldükçe indirim olayının planlanması giderek rutinleşmektedir.

Promosyon faaliyeti, müşteri kitlesini artırdığında, daha düşük kâr ile de olsa tüm firmalarca tercih edilmiştir. Firma büyüdükçe promosyonun yenilenmesi daha düzenli hale gelse de aralarında büyük farklar bulunmamaktadır.

6.4. SONUÇ

Tüm firma çözümleri incelendiğinde senaryoların, modellenen çerçevede gerçekçi çözümler olduğu, firmaların beklenen hareket tarzları ile uyduğu değerlendirilmektedir.

Müşterilerin modelde gruplanarak daha bireysel tercih ve etkileşimlerin modellenmesi, senaryo çözümlerini daha gerçekçi bir sonuca götürmüştür. Böyle bir modelin standart bir modelleme aracı ile yapılması mümkün görülmemektedir. Çözümlerden birinin simülasyon dillerinden biri ile modellenmesi mümkün olsa da modelleri üretilip karşılaştırmasını yapacak bir araca rastlanmamıştır.

Model içerisinde stokastik unsurların olması nedeni ile elde edilen sonuçların puanlarının ortalaması alınarak değerlendirilmektedir. En iyi çözümün sonuçları istendiğinde ise yeni bir iterasyonun sonucu verilmektedir. Risk değerlendirilmesi yapılması gerekiyorsa sonuçlar arasında verilen iterasyon puanlarının da değerlendirilmesi yapılmalıdır. Alt sınır olarak belirlenen puanın altında istenen orandan daha fazla iterasyon sonucu varsa sonraki iyi çözüm tercih edilebilir. Model sonucunda tüm kromozomların sonuç puanları gösterilmekte ve en iyi çözümler sıralanmaktadır.

YEDİNCİ BÖLÜM

ÖRNEK SENARYO 2

İkinci örnek senaryo olarak bir Resim İşleme Modeli kurgulanmıştır. Bu problemde 6 farklı tipte resim önceden çeşitli filtrelerden geçirilerek değiştirilmiştir. Kurulan modelin en yakın filtreleme senaryosunu bulması amaçlanmaktadır. Yani orijinal resme hangi filtreler hangi sırayla uygulanırsa değiştirilen resme ulaşılmaktadır.

Seçilen resimler;



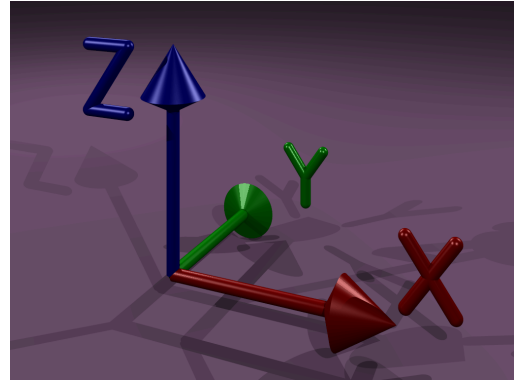
Basit Çizim



Karikatür



Manzara



3 Boyutlu Çizim



Cihaz



İnsan

Şekil 7.1. Test Maksadıyla Kullanılan Resimler

6 farklı resim seçilmesinin nedeni modelin farklı karmaşıklıkta resimlere uygulanan filtreleri çözümleyip çözemediğinin görülmesidir. Amaç, filtrelenmekte olan resmin hedef resme ne kadar benzediği olup her resim için eşit ağırlıktadır. Buradaki 6 resme aynı filtre uygulanmıştır. Çözüm sürecinde de her resme aynı anda aynı filtre uygulanacaktır.

Yapılan testlerde Intel i7 920 işlemcili ve 6GB RAM olan bir bilgisayar kullanılmıştır.

7.1. FİLTRELER

Tüm filtreleme işlemleri HSV (Hue – Saturation – Value) (Renk – Doygunluk – Parlaklık) renk paleti ile yapılmaktadır. Bu nedenle tüm resimler filtrele işleminden önce RGB (Kırmızı – Yeşil – Mavi) renk paletinden HSV'ye dönüştürülmekte, tüm filtre işlemlerinin tamamlanmasını müteakip tekrar RGB formatına dönüştürülerek resim kaydedilmektedir. Filtrelemeler esnasında renk ve doygunluk değerleri aynı bırakılarak sadece parlaklık değeri değiştirilmektedir.

Aşağıda, örnek uygulamada kullanılan altı basit filtre algoritması açıklanmıştır.

7.1.1. Mean Filter (Ortalama Filtresi)

1	2	3
4	5	6
7	8	9

Şekil 7.2. 8-Komşu pikseller

Bu filtreleme işleminde tüm piksellerin filtrelenmiş değeri, her 8-komşu piksel ile kendisinin ortalaması alınarak bulunur. Şekilde sarı piksele ait sekiz komşu beyaz piksel görülmektedir.

7.1.2. Median Filter (Medyan Filtresi)

Filtrelenmiş piksel değeri 8-komşu piksel ile kendisinin medyanı alınarak bulunur.

7.1.3. Shrunked Mean (Daraltılmış Ortalama Filtresi) ve Shrunked Median (Daraltılmış Medyan Filtresi)

Bu iki filtre de tıpkı ortalama ve medyan filtreleri gibi çalışır. Farkı ise; Bu filtreler bir parametre değeri alır ve kontrol ettiği komşu pikselin parlaklık değeri bu parametre değerine küçük eşitse ortalamaya veya medyan hesaplamasına dâhil eder. Dolayısı ile bu filtrede sadece birbirine yakın renkler birbirine daha da yaklaşır (Şimşek 2005).

7.1.4. Shrunked Mean Wide (Daraltılmış Ortalama Filtresi, Geniş) ve Shrunked Median Wide (Daraltılmış Medyan Filtresi, Geniş)

Bu filtre de bir öncekine ek olarak bir kapsam parametresi vardır. Bu parametre komşuluğun genişliğini belirlemektedir. Kapsam 1 ise 8-komşu pikseller hesaplamaya dâhil olur, kapsam 2 ise hesaplanan piksele 2 uzaklıktaki pikseller de dâhil olur. Burada ele alınacak piksel sayısı $(2 \times \text{kapsam} + 1)^2$ olarak hesaplanabilir. (Şimşek 2005)

7.2. TEST 1

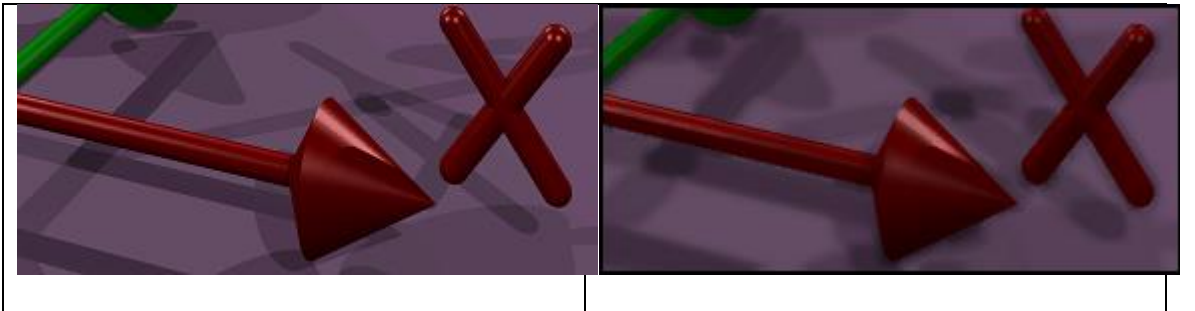
Testte altı adet resim bulunmaktadır. Bu resimlere Çizelge 7-1'deki filtreler sırasıyla uygulanmıştır. Şekil 7.3. **Filtrelenen Resimler** orijinal resimler ve filtrelenen resimler birlikte görülmektedir. Orijinal resimler ve referans olmak üzere filtrelenen resimler test modeline verilmektedir. Test uygulaması verilen resimleri filtrelenen resimlere dönüştürmek üzere filtre havuzunda tanımlı filtreleri uygulayarak karşılaştırmaktadır. Bu test için senaryoların karşılığı olan filtre dizileri içerisinde referans olarak verilen resimlere en yakın sonucu veren filtre dizisi en iyi çözüm olarak alınmaktadır.

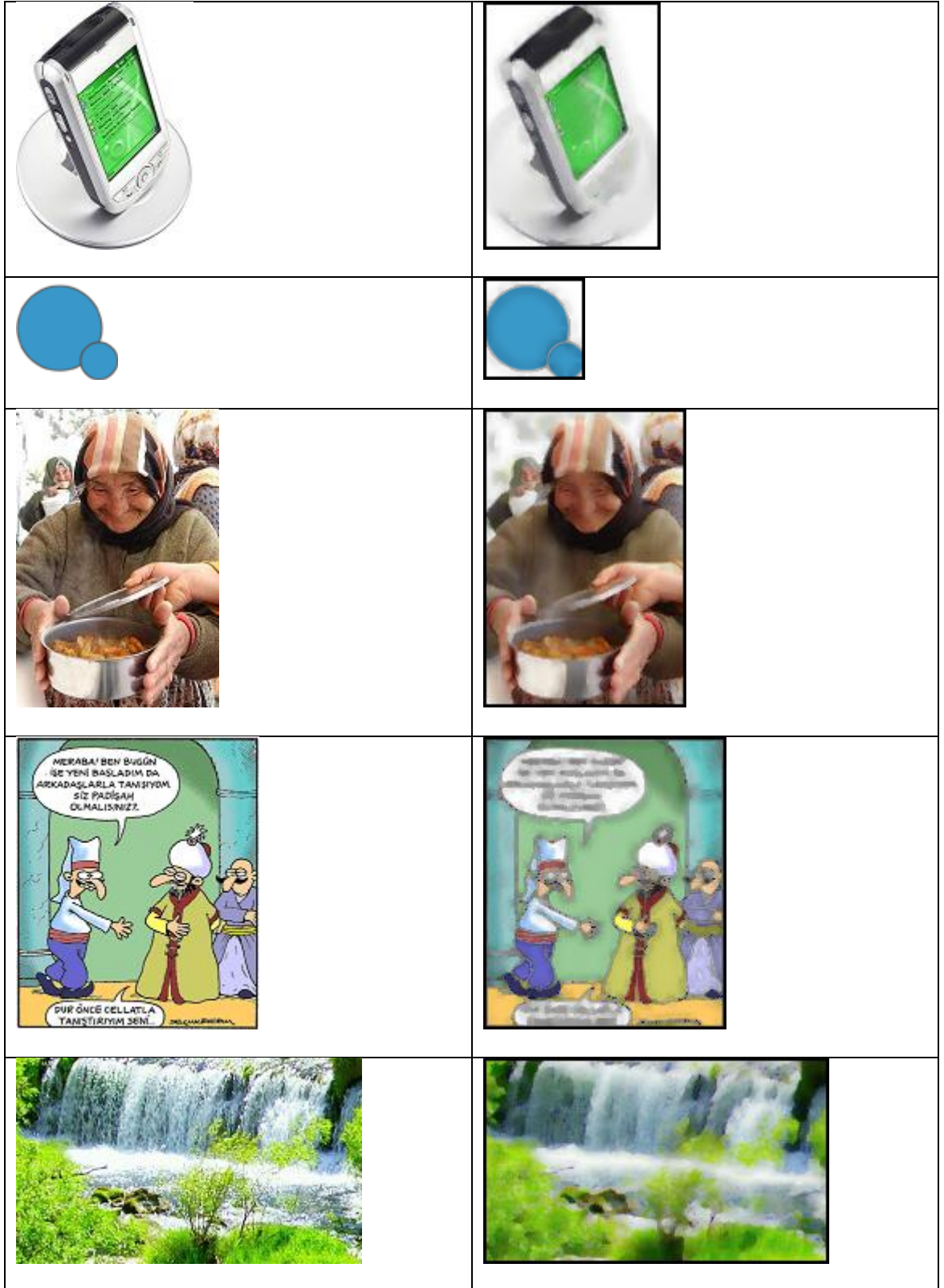
Test için aşağıdaki filtreler sırasıyla uygulanmıştır;

Çizelge 7.1. Test Filtre Sırası

İşlem sırası		Tekrar sayısı
1	Mean	1
2	Median	1
3	Shrunked Mean	1
4	Shrunked Median	1
5	Shrunked Mean Wide	1
6	Shrunked Median Wide	1
Parametreler		
Eşik Değeri		20
Sınır		2

Bu filtreleme sonucunda üretilen resimler ile resimlerin asıllarının karşılaştırması aşağıdaki gibidir;





Şekil 7.3. Filtrelenen Resimler

Çözüm için modele aşağıdaki parametreler verilmiştir;

Çizelge 7.2. Test 1 Parametreler

Parametreler	
Çaprazlama Yöntemi	Tam Analiz
Kanal Sayısı	8
Azami Nesil Sayısı	20
Popülasyon Sayısı	100
Azami Olay Sayısı	2
Üstün Nesil	Evet
Zayıf Kromozomları Değiştir	8 Nesil, %10
Mutasyon	Hayır

Bu model için çözüm süresi 28 dakika 53 saniye olarak gerçekleşmiştir. Bu süre içinde yaklaşık 450.000 filtreleme işlemi yapıldı. En uygun çözüme ikinci nesilde ulaşıldı. En iyi çözümde test olarak yapılan filtre dizisi doğru olarak bulunmuştur.

7.3. TEST 2

İkinci testimizde test 1'in üzerine filtre tekrarları artırılarak işlem karmaşılaştırılmıştır. Test 2 için aşağıdaki filtreler gösterilen tekrar sayısı kadar sırasıyla uygulanmıştır;

Çizelge 7.3. Test 2 Filtre Sırası

İşlem sırası		Tekrar sayısı
1	Median	3
2	Shrunked Median	1
3	Shrunked Mean Wide	1
4	Shrunked Median Wide	2
5	Shrunked Mean	2
6	Shrunked Median	2
Parametreler		
Eşik Değeri		20
Sınır		2

Çözüm için modele aşağıdaki parametreler verilmiştir;

Çizelge 7.4. Test 2 Parametreler

Parametreler	
Çaprazlama Yöntemi	Tam Analiz
Kanal Sayısı	8
Azami Nesil Sayısı	10
Popülasyon Sayısı	150
Azami Olay Sayısı	3
Üstün Nesil	Evet
Zayıf Kromozomları Değiştir	8 Nesil, %10
Mutasyon	Hayır

Bu model için çözüm süresi 3 saat 47 dakika 6 saniye olarak gerçekleşmiştir. En uygun çözüme sekizinci nesilde ulaşıldı. Bu testte uygulanan filtre dizisine yakın bir çözüm bulunmuştur.

En iyi 12 çözüm aşağıdaki senaryoyu bulmuştur;

- | | |
|----|--------------------|
| 1 | Median |
| 2 | Median |
| 3 | Median |
| 4 | ShrunkedMedian |
| 5 | ShrunkedMeanWide |
| 6 | ShrunkedMedianWide |
| 7 | ShrunkedMedian |
| 8 | ShrunkedMedianWide |
| 9 | ShrunkedMean |
| 10 | ShrunkedMean |
| 11 | ShrunkedMean |
| 12 | ShrunkedMedian |

Çizelge 7.5. Test 2 Sonucu

	Cihaz	Basit Çizim	Karikatür	Manzara	3 Boyutlu Çizim	İnsan
Orijinal resimle filtre arasındaki renk farkı	29,8716	37,8618	32,1388	28,5675	9,6890	26,1073
Hesaplanan renk farkı	0,6174	0,1190	0,5597	1,1534	0,2833	1,03276

Renk farkı hesaplanırken toplam renk farkı resmin boyutuna oranlanmıştır. Burada hesaplanan ortalama fark %2,10 olarak hesaplanmıştır. İlk altı adım bire bir aynı iken yedinci adımda ShrunkedMedianWide yerine ShrunkedMedian filtresi kullanılmıştır. Sonraki filtreler incelendiğinde filtre sayıları aynıdır, yalnız ShrunkedMean bir kez fazla kullanılmıştır. Sıralamada ise iki filtrenin yeri değişmiştir. Hesaplanan fark dikkate alınırsa test 2 %97.9 başarı göstermiştir.

7.4. TEST 3

Üçüncü Test için ilk iki test ile aynı koşullarda tüm filtreler yerine Çizelge 7-6'da gösterilen üç filtre sırasıyla uygulanmıştır;

Çizelge 7.6. Test 3 Filtre Sırası

İşlem sırası		Tekrar sayısı
1	Shrunked Mean Wide	4
2	ShrunkedMedian	2
3	Median	3
Parametreler		
Eşik Değeri		20
Sınır		2

Çözüm için modele aşağıdaki parametreler verilmiş ve İşlem 2 sa18 dk 30sn sürmüştür.

Çizelge 7.7. Test 3 Parametreleri

Parametreler	
Çaprazlama Yöntemi	Puanlama
Kanal Sayısı	8
Azami Nesil Sayısı	10
Popülasyon Sayısı	200
Azami Olay Sayısı	4
Üstün Nesil	Evet
Zayıf Kromozomları Değiştir	8 Nesil, %10
Mutasyon	Hayır

Çizelge 7.8. Test 3 En İyi Çözümler

Uygulanan Filtre	Kromozom No		
	2116	1518	2173
Shrunked Mean Wide	ShrunkedMeanWide	ShrunkedMeanWide	ShrunkedMeanWide
Shrunked Mean Wide	ShrunkedMeanWide	ShrunkedMeanWide	ShrunkedMeanWide
Shrunked Mean Wide	ShrunkedMeanWide	ShrunkedMeanWide	ShrunkedMeanWide
Shrunked Mean Wide	ShrunkedMeanWide	ShrunkedMeanWide	ShrunkedMeanWide
ShrunkedMedian	ShrunkedMedian	Median	ShrunkedMean
ShrunkedMedian	ShrunkedMean	ShrunkedMedianWide	ShrunkedMedian
Median	Median	Median	Median
Median	Median	Median	Median
Median	ShrunkedMedian		ShrunkedMedian
	Median		Median

Burada yakın sonuçlar bulunmuş, ancak birebir sonuç bulunamamıştır. Nesil sayısını artırarak aynı problemi tekrar çözdüğümüzde tam sonuç elde edilmektedir. Yapılan karşılaştırmada en iyi çözümde filtreler ve sıralamalar aynı olup fazladan ShrunkedMedian filtresi bulunmaktadır.

7.5. TEST 4

Bu testte Test 3 farklı parametrelerle tekrar çözülmüştür.

Çizelge 7.9. Test 4 Filtre Dizisi

İşlem sırası		Tekrar sayısı
1	Shrunked Mean Wide	4
2	ShrunkedMedian	2
3	Median	3
Parametreler		
Eşik Değeri		20
Sınır		2

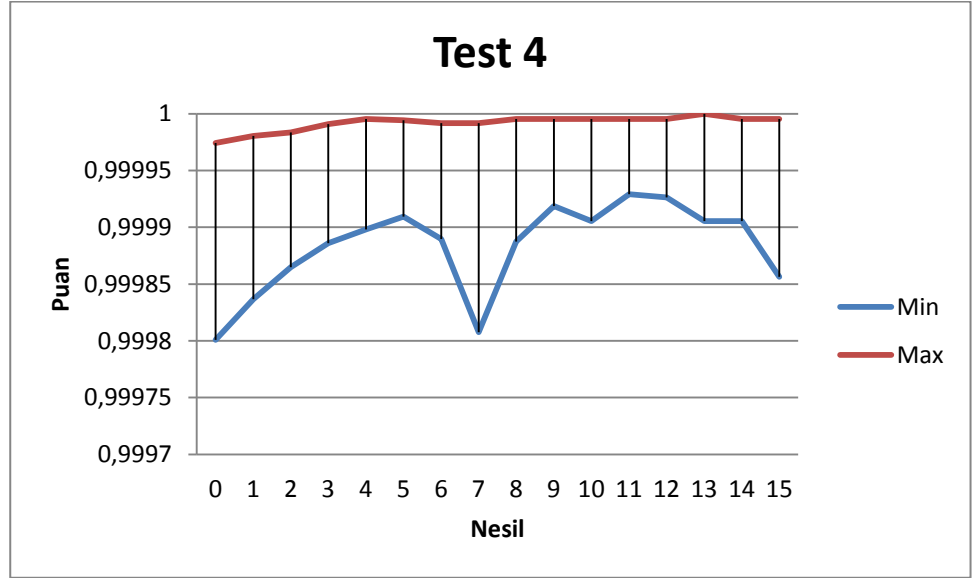
Çözüm için modele Çizelge 7.10'daki parametreler verilmiştir. Test 4'de test 3 ile aynı filtre dizisi kullanılmış, çözüm farklı parametrelerle yapılmıştır. Çizelgede ayrıca parametrelerin karşılaştırılması da görülmektedir.

Çizelge 7.10. Test 4 Parametreleri

Parametreler	Test 3	Test 4
Çaprazlama Yöntemi	Puanlama	Puanlama
Kanal Sayısı	8	10
Azami Nesil Sayısı	10	16
Popülasyon Sayısı	200	180
Azami Olay Sayısı	4	4
Üstün Nesil	Evet	Evet
Zayıf Kromozomları Değiştir	8 Nesil, %10	8 Nesil, %12
Mutasyon	Hayır	Hayır

İşlem 3 saat 30 dakika 56 saniye sürmüştür. Bu çözümde tam sonuç elde edilmektedir. Test 4, Nesil-Puan Grafiğinden en uygun çözümün 13. nesilde (Kromozom 2458) bulunduğu görülüyor. Grafikte gösterilmeyen üstün neslin tamamında Kromozom 2458 ebeveyn olarak alınarak birebir çözüm bulunmuştur. Grafikte min sınırın 7 ve 15. nesillerde düşme sebebi ise burada %12 zayıf kromozomların değiştirilmesidir.

Bu örneklerde çözüm sürecinin uzun olma sebebi; her bir olay olarak bir filtreleme işleminin alınması ve her bir tek işlemin içerisinde resmin boyutuna göre her bir piksel için elli ilâ yüz yirmi civarı işlem yapılmasıdır.



Şekil 7.4. Test 4 Nesil-Puan Grafiği

SONUÇ ve ÖNERİLER

Olaylara dayalı senaryoların genetik algoritma ile optimizasyonu yöntemi ile her türlü optimizasyon problemi çözülebilmektedir. Standart problemlerin çözümü bu yöntem ile daha uzun sürebilmektedir. Ancak bu yöntemin amacı kolaylıkla herhangi bir optimizasyon yöntemi için modellenemeyen problemlerin burada çözülebilmesidir.

Tezin başlangıcında amaçlanan sosyal hayatın modellenmesi ve verilen bir amacın optimizasyonu bu yöntem ile mümkündür ancak modelleme işlemi bir başka tezin konusu olabilecek ölçüde detaylı bir çalışma gerektirmektedir. İlk örnek çalışmada yapılan firma örneğinin sonuçları incelendiğinde hedeflenen süre için alternatif senaryolar ile birlikte hareket tarzlarının belirlenebildiği, tekrarlanan iterasyonlar ile farklı olası durumların önceden görülebileceği değerlendirilmiştir. Böylelikle uygulamanın gerçek hayat problemlerinde strateji ve hareket tarzlarının belirlenmesinde kullanılabileceği ispatlanmıştır.

Diğer örnek çalışmada ise fiziksel, ekonomik, vb. problemlerin yanında büyük bir veri kümesini temsilen bir resim ve işlem veya olayları temsilen fonksiyonlar kullanılarak bir iş süreci veya yazılımsal algoritmaların dahi optimize edilebileceği gösterilmiştir.

Burada geliştirilen modelin bir güçlü yanı; değişken olarak kullanılan durum değerlerindeki normale dönüş, direnç, etki gibi özelliklerin varlığıdır. Bu özelliklerle sürekliliği olan bazı enerjiler modelde kolaylıkla tanımlanabilmektedir. Psikolojik durumları simüle etmek için tasarlanan bu yapılar “Level Set” gibi resim işlemede kullanılan bazı algoritmalarındaki enerjileri veya ilgili nesnenin sıcaklığının oda sıcaklığına olan meyli gibi konuları rahatlıkla temsil edebilmektedirler.

Olaylara dayalı senaryoların genetik algoritma ile optimizasyonu içinde kullanılacak parametreler modele göre değişiklik gösterebilir. Örneğin olaylar birbirinden bağımsız ve sonuçları başka olayları etkilemiyorsa çaprazlama için tam analiz yerine puanlama yöntemini kullanmak aynı sonucu daha hızlı getirecektir. Ancak olaylar veya sonuçları arasında bir bağlantı varsa tam analiz yöntemini kullanmak optimum sonucu verecektir.

Farklı uygulama alanları için oluşturulan yapılar, daha sonra benzer uygulamalar için de kullanılabilir. Bu maksatla benzer alanlarda yapılacak çalışmaların bir araya getirilerek kütüphaneler oluşturulması, gelecek modellemelerin daha kolay ve kısa sürede yapılabilmesi açısından büyük fayda sağlayacaktır.

Kullanım kolaylığı ve modellemenin daha hızlı yapılabilmesi için kütüphane kullanımına ek olarak; model için tanımlamaların yapılabileceği bir ara yüzün geliştirilmesi ve kullanıcı tanımlı olarak eklenen şart, tetik ve eylem fonksiyonlarının okunabilirliğinin artması ve kodlama süresinin kısalması için bir ara dilin geliştirilmesi faydalı olacaktır.

Küçük modellerde, modelin doğruluğunun test edilmesi ve sonuçların değerlendirmesi kolay olsa da model büyüdükçe bu konuda başka araçların geliştirilmesi ihtiyacı belirmiştir. Bu maksatla geliştirilecek araçta durum değerlerinin ve diğer istatistiki verilerin bir veri tabanında saklanabilmesi, çeşitli sorgu ve grafiklerin sunulabilmesi gerekmektedir.

Stokastik unsurların olması durumunda model, Monte Carlo simülasyonları gereğince çözülmektedir. Ancak veri kümesinin büyük ve çözüm sayısının çok fazla olması nedenleri ile inceleme aşamasında yeni bir iterasyonun sonuçları kullanıcıya gösterilmektedir. Çözümün incelenmesi aşamasında birden çok iterasyonun sonuçlarının birlikte incelenmesine imkân verilmelidir.

Bununla birlikte puanlamaya ek olarak bir risk faktörünün eklenmesi ve iterasyonlar içerisinde kritik durum değişkenleri belirtilen değerlere ulaşan simülasyon sonucuna sahip çözümler elenebilmelidir.

KAYNAKÇA

- Abraham, A., Nedjah, N., Macedo, Mourelle L. (2006). *Evolutionary Computation: from Genetic Algorithms to Genetic Programming*. Studies in Computational Intelligence (SCI) 13.
- Allison, P.D. (1994). "Using Panel Data to Estimate the Effects of Events". *Sociological Methods & Research*. 23(2).
- Beasley, D., Bull, D.R., Martin, R.R. (1993). *An Overview of Genetic Algorithms Part 1 & 2*.
- Cantu-Paz, E. (1998). *A Survey of Parallel Genetic Algorithms*. University of Illinois at Urban-Champaign.
- Cao, X. (2007). *Stochastic Learning And Optimization: A Sensitivity-Based Approach*, Springer.
- Chair, J.T.O., Belytschko, T., Fish J., Hughes, J.R.T., Johnson, C., Keyes, D., Laub, A., Petzold, L., Srolovitz, D., Yip, S. (2006). "Simulation-Based Engineering Science, Revolutionizing Engineering Science through Simulation", *Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science*
- Coit, D.W., Konak, A., Smith, A.E. (2005). *Multi-Objective Optimization Using Genetic Algorithms: A Tutorial*
- Cox, E., (1994). *The Fuzzy Systems Handbook*, New York: AP Professional.
- Çunkaş, M., (2006). *Genetik Algoritmalar ve Uygulamaları Ders Notları*. Konya: Selçuk Üniversitesi Teknik Eğitim Fakültesi.
- Deng, G., (2007). *Simulation-Based Optimization*. University of Wisconsin.
- Dixon, M.J., Thompson, G.M. (ty) *Optimal Event Sequencing: Sequence Effects in Bundling and Scheduling Events*. Cornell University.
- Emel, G.G., Taşkın, Ç. (2002). "Genetik Algoritmalar ve Uygulama Alanları", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, XXI(1).
- Geiss, C. (2009). "Stochastic Modeling", Erişim Tarihi 05.01.2013, <http://users.jyu.fi/~geiss/scripts/stochastic-models.pdf>
- GenelHaber.org, 12 Ağustos 2013 tarihli haber, Erişim Tarihi: 5 Aralık 2013, <http://www.genelhaber.org/bilgisayar-satislarini-gecti/>.

- Ghosh, A., Satchidananda Dehuri (Nisan 2004). "Evolutionary Algorithms for Multi-Criterion Optimization: A Survey, *International Journal of Computing & Information Sciences*, 2(I), 38-54.
- Habertürk, 26 Ekim 2012 tarihli teknoloji haberi, <http://ekonomi.haberturk.com/teknoloji/haber/788533-iste-turkiyenin-internet-karnesi>, Erişim Tarihi: 5 Aralık 2013.
- Hao, T., Haifeng W., Lei Z., Jianghong H. (Aralık 2009). "Optimal control of multiple CSPA system based on event-based Q learning", *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai.
- Hubscher-Younger T., Mosterman, P.J., deLand, S., Orqueda, O., Eastman, D. (2012). "Integrating Discrete-Event And Time-Based Models With Optimization For Resource Allocation", *Proceedings of the 2012 Winter Simulation Conference*.
- Jameson, A. (1996). "Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues" [Elektronik Sürüm], *User Modeling and User-Adapted Interaction*, 5. Erişim tarihi: 01 Mayıs 2010, <http://www.dfki.de/~jameson/pdf/umuai96.jameson.pdf>
- Konar, A. (2000). *Behavioral and Cognitive Modeling of the Human Brain*, CRC Press.
- Katoen, J.P. Brinksma, E., Latella, D., Langerak, R. (1996). *Stochastic Simulation of Event Structures*, Erişim Tarihi: 28 Nisan 2010, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.44.4380&rep=rep1&type=pdf>
- Marketing Teacher (2014). *The Product Life Cycle (PLC)*. Erişim Tarih: 31.03.2014. <http://www.marketingteacher.com/lesson-store/lesson-plc.html>
- Michie, D., Spiegelhalter D. J., Taylor C.C. (1994). *Machine Learning, Neural and Statistical Classification*. Prentice Hall.
- Mind, Tools (ty). *The Product Life Cycle*, Erişim Tarih: 31.03.2014. http://www.mindtools.com/pages/article/newSTR_80.htm
- Mitsuo, Gen - Lin, Lin – Runwei, Cheng (2004). "Bicriteria Network Optimization Problem using Priority-based Genetic Algorithm", *IEEJ Transactions on Electronics Information and Systems*.
- Mühlenbein, H., Schomisch, M., Born. J. (1991). *The Parallel Genetic Algorithm As Function Optimizer*, Elsevier.

- Nikolova. E., Karger. D.R. (2008). "Route Planning under Uncertainty: The Canadian Traveller Problem" [Belirsizlikte Yol Bulma: Kanadalı Yolcu Problemi], *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 969-974.
- Nilsson. H. (2005). "Dynamic Optimization for Functional Reactive Programming using Generalized Algebraic Data Types", *ICFP '05 Talin, Estonya*. Erişim Tarihi: 28 Nisan 2010, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.7810&rep=rep1&type=pdf>
- Northern Kentucky University, Product Life Cycle, Erişim Tarihi, 31.03.2014. http://www.nku.edu/~issues/internet_marketing/newwebpage1.html
- Quick. MBA, *The Product Life Cycle*, Erişim (31.03.2014). <http://www.quickmba.com/marketing/product/lifecycle/>
- Sobol. I.M. (1974). *The Monte Carlo Method*, The University of Chicago Press.
- Şimşek. B. (2005). "Road Extraction From High Resolution Satellite Imagery", *A Master Project Report Submitted to the Graduate School of Informatics of the Middle East Technical University*.
- Webrazzi (2013). 3 Aralık 2013 tarihli haber, <http://www.webrazzi.com/2013/12/03/pc-satislari-beklenenden-hizli-dusuyor-ama-kullanim-orani-azalmiyor/>, Erişim Tarihi: 5 Aralık 2013.
- Weise, T. (2009). *Global Optimization Algorithms, Theory and Application*.
- Zadeh, L. A. (Nisan 1988). *Fuzzy Logic* [Elektronik Sürüm], 1988 *IEEE*, 83-93.
- Zeigler, B.P., Praehofer H., Kim T.G. (2000). *Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2nd Edition. Academic Press.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Bülent Şimşek
Doğum Yeri ve Tarihi	Almanya - 1973
Eğitim Durumu	
Lisans Öğrenimi	Kara Harp Okulu
Y. Lisans Öğrenimi	ODTÜ
Bildiği Yabancı Diller	İngilizce, Almanca
Bilimsel Faaliyetleri	
İş Deneyimi	
Stajlar	
Projeler	Lojistik Bilgi Sistemleri
Çalıştığı Kurumlar	Türk Silahlı Kuvvetleri
İletişim	
E-Posta Adresi	Simsek.bulent@gmail.com
Tarih	