



**MİMARİ TASARIMIN ÜRETİLMESİNDE GENETİK ALGORİTMA VE
PARAMETRİK TASARIM İLE BİR YÖNTEM ARAYIŞI**

Şule Betül DEMİRKOL

**YÜKSEK LİSANS TEZİ
MİMARLIK ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2019

Şule Betül DEMİRKOL tarafından hazırlanan “MİMARİ TASARIMIN ÜRETİLMESİNDE GENETİK ALGORİTMA VE PARAMETRİK TASARIM İLE BİR YÖNTEM ARAYIŞI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Mimarlık Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Arzu ÖZEN YAVUZ

Mimarlık Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Can Mehmet HERSEK

İç Mimarlık ve Çevre Tasarımı Ana Bilim Dalı, Başkent Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Tayfun YILDIRIM

Mimarlık Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 18/06/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....
Şule Betül DEMİRKOL
18/06/2019

MİMARİ TASARIMIN ÜRETİLMESİNDE GENETİK ALGORİTMA VE PARAMETRİK TASARIM İLE BİR YÖNTEM ARAYIŞI

(Yüksek Lisans Tezi)

Şule Betül DEMİRKOL

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2019

ÖZET

Mimari tasarım süreci oldukça karmaşık bir süreçtir. Daha tasarımın ilk evrelerinde birbirini etkileyen birçok karar verilmeli ve birçok değişken dikkate alınmalıdır. Dolayısıyla doğru kararların verilmesi önemlidir. Ancak yine de verilen kararların tüm olası sonuçlarını öngörmek ve en optimum sonuca ulaşmak çok zordur. Bu nedenle özellikle de bilgi işlemsel tasarım yaklaşımlarının gelişmesi ile birlikte, mimari tasarıma yardımcı olacak birçok yöntem ve teknik araç kullanılmıştır. Kullanılan bu yöntemlerden biri de Darwin'in evrim teorisini ve canlılardaki çoğalma mantığını temel alan Genetik Algoritma yaklaşımıdır. Genetik Algoritma canlılardaki büyüme ve gelişme süreçlerini taklit eden, böylelikle geniş bir çözüm uzayını hızlı bir şekilde tarayarak istenilen optimum sonuca ulaşan bir üretken arama algoritmasıdır. Ayrıca paralel arama yapması sebebiyle gerçek hayat problemleri gibi birçok değişkenin aynı anda optimize edilmesi gereken problemlerde oldukça kullanışlıdır. Bu çalışmada da mimari form üretmede tasarımcıya yardımcı olabilecek, sudoku çözüm mantığından ilham alan üretken bir genetik algoritma modeli hazırlanmıştır. Sudoku bulmacanın analizinden ve üretim kurallarından yola çıkılarak oluşturulan bu modele SuGe modeli adı verilmiştir.

Bilim Kodu : 80107
Anahtar Kelimeler : Genetik Algoritma, Sudoku, Evrim, Mimarlıkta Biçim, Üretken Yaklaşım, Algoritmik Tasarım, Parametrik Tasarım
Sayfa Adedi : 89
Danışman : Doç. Dr. Arzu ÖZEN YAVUZ

SEARCHING FOR A METHOD WITH GENETIC ALGORITHM AND PARAMETRIC
DESIGN IN THE PRODUCTION OF ARCHITECTURAL DESIGN

(M. Sc. Thesis)

Şule Betül DEMİRKOL

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2019

ABSTRACT

The architectural design process is quite complex. In the early stages of the design, many decisions that affect each other must be made and many variables must be considered. Therefore, it is important to make the right decisions. However, it is still very difficult to predict all the possible consequences of the decisions made and to reach the optimum result. Therefore, especially with the development of computational design approaches, many methods and technical tools have been used to assist architectural design. One of these methods is the Genetic Algorithm approach, which is based on Darwin's theory of evolution and the logic of proliferation in living things. Genetic Algorithm is a productive search algorithm that simulates the growth and development processes in living things and thus achieve the desired optimum result via scanning a large solution space rapidly. It is also useful for problems that need to be optimized at the same time as many variables, such as real-life problems thanks to parallel search. In this study, a productive genetic algorithm model is inspired by sudoku solution logic, which can help the designer to produce architectural forms, has been created. This model based on the analysis and production rules of Sudoku puzzles is named the SuGe model.

Science Code : 80107
Key Words : Genetic Algorithm, Sudoku, Evolution, Form in Architecture,
Generative Approach, Algorithmic Design, Parametric Design
Page Number : 89
Supervisor : Assoc. Prof. Dr. Arzu ÖZEN YAVUZ

TEŞEKKÜR

Mimarlık eğitimim boyunca ve tezimi hazırlarken desteğini hiçbir zaman esirgemeyen ve değerli fikirleriyle her zaman önümü açan sevgili hocam, Doç. Dr. Arzu ÖZEN YAVUZ'a çok teşekkür ederim. Bütün hayatım boyunca yanımda olan, desteklerini ve sevgilerini her daim kalbimde hissettiğim annem, babam ve kardeşlerime çok teşekkür ederim. Özellikle akademisyen olması sebebiyle sürekli sorular sorduğum sevgili ablam Neslihan DEMİRKOL'a sabrı ve desteği için çok teşekkür ederim. Ayrıca tezimi hazırladığım süreçte daima yanımda olan, beni dinleyen ve cesaretlendiren sevgili arkadaşlarım Erva ILICAK, Saliha BAYRAK, Kadriye DURMUŞ, Melike DAVARCI, Beyzanur TUHAN, İmran IŞIK, Eda BASKIN, Alperen İKİZ ve Erkan ACAR'a çok teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ŞEKİLLERİN LİSTESİ.....	x
RESİMLERİN LİSTESİ	xiv
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ.....	1
2. GENETİK ALGORİTMA KURAMI VE MİMARİ TASARIMDAKİ YERİ.....	5
2.1. Evrimsel Hesaplama.....	5
2.2. Genetik Algoritmalar.....	6
2.2.1. Genetik algoritma terminolojisi	8
2.2.2. Genetik algoritma işleyişi	10
2.3. Genetik Algoritma Operatörleri.....	13
2.3.1. Parametre kodlama	13
2.3.2. Uygunluk değeri hesaplama ve başlangıç popülasyonunu oluşturma.....	15
2.3.3. Üreme ve seçim.....	16
2.3.4. Çaprazlama operatörü	19
2.3.5. Mutasyon operatörü	22
2.3.6. Durdurma operatörü	23
2.4. Genetik Algoritma Parametreleri	24
2.4.1. Popülasyon büyüklüğü.....	25
2.4.2. Kuşak farkı	26

	Sayfa
2.4.3. Çaprazlama olasılığı.....	26
2.4.4. Mutasyon olasılığı.....	26
2.4.5. Seçim stratejisi	27
2.4.6. Uyum ölçeklemesi.....	27
2.5. Genetik Algoritmanın Avantaj ve Dezavantajları	27
2.6. Genetik Algoritmanın Uygulama Alanları.....	31
2.6.1. Optimizasyon problemleri.....	31
2.6.2. Gezgin satıcı problemleri	32
2.6.3. Araç rotalama problemleri	32
2.6.4. Çizelgeleme problemleri	32
2.6.5. Tesis yerleşimi problemleri.....	33
2.7. Mimari Tasarım Uygulamalarında Genetik Algoritmaların Kullanımı	33
2.7.1. Embriyonik ev	35
2.7.2. Moleküler kurgulu ev	36
2.7.3. Biçimin evrimi projesi.....	37
2.7.4. Form-G modeli.....	41
3. GENETİK ALGORİTMAYA DAYALI BİR MODEL ÖNERİSİ:	
SUGE MODELİ.....	47
3.1. Sudoku.....	47
3.1.1. Sudoku bulmacanın çözümünde genetik algoritma kullanımına dair bir örnek	48
3.2. SuGe Modeli.....	50
3.2.1. Analiz	51
3.2.2. Üretim -I-	53
3.2.3. Üretim -II-	58

	Sayfa
3.2.4. Değerlendirme.....	71
4. SONUÇ	79
KAYNAKLAR	83
ÖZGEÇMİŞ	89

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. SuGe modeli kavramsal şeması	4
Şekil 2.1. İkili kodlu GA ile biyolojik evrim arasındaki benzetim.....	7
Şekil 2.2. Allel, gen ve kromozom.....	9
Şekil 2.3. Gen, kromozom, popülasyonun ilişkisi.....	9
Şekil 2.4. Genotipler ile fenotiplerin eşleşmesi.....	10
Şekil 2.5. Genetik algortima akış diyagramı.....	11
Şekil 2.6. İkili kodlamada ile hazırlanmış kromozom örnekleri.....	14
Şekil 2.7. Gerçek değer kodlama ile hazırlanmış kromozom örnekleri.....	14
Şekil 2.8. Permutasyon kodlama ile hazırlanmış kromozom örnekleri.....	15
Şekil 2.9. Rulet tekerleği örneği.....	17
Şekil 2.10. Rulet tekerleği ve sıralı seçim yöntemlerinin karşılaştırılması.....	18
Şekil 2.11. Basit çaprazlama operatörü.....	20
Şekil 2.12. Tek noktalı çaprazlama.....	21
Şekil 2.13. İki noktalı çaprazlama.....	21
Şekil 2.14. Çok noktalı çaprazlama.....	21
Şekil 2.15. Uniform çaprazlama.....	22
Şekil 2.16. Basit mutasyon operatörü.....	23
Şekil 2.17. Mutasyon yöntemleri.....	23
Şekil 2.18. Biçim grameri ile tasarım örneği.....	37
Şekil 2.19. Modüllerin şekilleri ve ifade ediliş biçimleri.....	38
Şekil 2.20. Temel birleşim kuralları ve varyasyonları.....	38
Şekil 2.21. Biçim grameri ile oluşturulmuş apartman tasarımı örnekleri.....	39
Şekil 2.22. Modüllerin numaralandırılması.....	39

Şekil	Sayfa
Şekil 2.23. Apartman bloğu değerlerinin dizi olarak gösterimi.....	39
Şekil 2.24. Kullanıcı arayüzleri.....	40
Şekil 2.25. En yüksek puanı alan bazı apartman modelleri.....	40
Şekil 2.26. Biçim Evrimi yöntemi sonucu elde edilen ev modeli.....	41
Şekil 2.27. Tasarım sınırlarıyla ilgili bilgilerin girileceği arayüz.....	42
Şekil 2.28. Kullanılacak objelerin seçildiği ve min.-max. hacimlerinin belirlendiği arayüzler.....	42
Şekil 2.29. Ebeveyn seçme ekranı.....	43
Şekil 2.30. Çaprazlama arayüzü.....	43
Şekil 2.31. Mutasyon arayüzü.....	44
Şekil 2.32. Mutasyona uğramış sonuç ürün.....	44
Şekil 2.33. Sonuç ürünün mimari sunumu.....	45
Şekil 3.1. Genetik algoritma modeli matrisi.....	48
Şekil 3.2. Modelin optimal olmayan sonucu.....	49
Şekil 3.3. Sonuç ürünün farklı yerleşimleri.....	49
Şekil 3.4. Sonuç ürün.....	49
Şekil 3.5. SuGe modeli akış şeması.....	50
Şekil 3.6. Bir sudoku bulmaca örneği ve çözümü.....	52
Şekil 3.7. Sudoku bulmacanın genetik algoritmaya göre öğeleri.....	52
Şekil 3.8. Üretim-I adımı ilk popülasyonu.....	53
Şekil 3.9. İki bireyin çaprazlanması örneği.....	54
Şekil 3.10. Üç bireyin çaprazlanması örneği.....	55
Şekil 3.11. Dört bireyin çaprazlanması örneği.....	55
Şekil 3.12. Üretim-I ikinci jenerasyonu.....	56
Şekil 3.13. Üretim-1 mutasyon örnekleri.....	57

Şekil	Sayfa
Şekil 3.14. Farklı Sudoku örnekleri.....	58
Şekil 3.15. Üretim-II’de kullanılacak grid ve rakamların geleceği noktalar.	59
Şekil 3.16. Üretim-II ilk popülasyon.....	60
Şekil 3.17. Üretim-II ilk popülasyonun fenotipinin oluşturulması – adım 1.....	61
Şekil 3.18. Üretim-II ilk popülasyonun fenotipinin oluşturulması – adım 2.....	62
Şekil 3.19. Üretim-II ilk popülasyonun fenotipi.....	63
Şekil 3.20. Çaprazlanacak bireyler.....	64
Şekil 3.21. Üretim-II / 2. Jenerasyon / 1 nolu birey.....	64
Şekil 3.22. Üretim-II / 2. Jenerasyon / 2 nolu birey.....	65
Şekil 3.23. Üretim-II / 2. Jenerasyonun diğer bireyleri.....	65
Şekil 3.24. Üretim-II / fenotip üretme kuralları.....	66
Şekil 3.25. Üretim-II / fenotip üretme algoritması.....	67
Şekil 3.26. Üretim-II / 2.jenerasyon fenotip oluşturma – adım 1.....	68
Şekil 3.27. Üretim-II / 2.jenerasyon fenotip oluşturma – adım 2.....	68
Şekil 3.28. Üretim-II / 2.jenerasyon dolu, yarı dolu ve boş olacak birimlerin belirlenmesi.....	69
Şekil 3.29. Üretim-II / 2.jenerasyon fenotipi.....	70
Şekil 3.30. 5 nolu bireyin mutasyonu.....	71
Şekil 3.31. SuGe modeli kabuk tasarımı.....	72
Şekil 3.32. Kabuk tasarım örneği genotipi.....	73
Şekil 3.33. Kabuk tasarımı fenotip üretimi-adım 1.....	73
Şekil 3.34. Kabuk tasarımı fenotip üretimi- adım 2	74
Şekil 3.35. Kabuk tasarımı örneği.....	74
Şekil 3.36. Yüzey tasarımı aşamaları.....	75
Şekil 3.37. Yüzey tasarımının farklı alanlardaki uygulaması.....	76

Şekil	Sayfa
Şekil 3.38. Kütle tasarımı – A örneği.....	77
Şekil 3.39. Kütle tasarımı – B örneği.....	77
Şekil 3.40. Kütle tasarımı – C örneği.....	77
Şekil 3.41. Kütle tasarımı – D, E, F ve G örnekleri.....	78

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. Kisho Kurukowa Helix Kent Projesi (Tokyo,1961)	34
Resim 2.2. Greg Lynn- Embriyonik Ev.....	35
Resim 2.3. Mohamad Alkhayer ve John M.Johanser - Moleküler Kurgulu Ev.....	36

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

m²

metrekare

vb.

ve benzeri

vs.

vesaire

Kısaltmalar

Açıklamalar

ÇN

Çaprazlama Noktası

DNA

Deoksiribonükleik Asit

GA

Genetik Algoritma

1. GİRİŞ

Yaratıcılık ve biçim arayışının ön planda olduğu tasarım odaklı bölümlerde tasarımcı adaylarının nasıl biçim üreteceği önemlidir. Dolayısıyla tasarım sürecinde oluşturulacak form, mimarın çözmek zorunda olduğu en büyük sorunlardan biridir. Hazırlanacak biçime dair birçok karar tasarımın erken evresinde verilir. Verilecek kararlar tasarım süreci ve oluşacak sonuç ürün için oldukça belirleyicidir. Bu kararların verilebilmesi için tasarımcı tarafından, elde edilmek istenen sonuç ürüne dair birçok araştırma ve analiz yapılmakta ve tasarım süreci buna göre yönetilmektedir. Ancak bu süreç her bir kararın bir diğerini etkilediği uzun ve karmaşık bir süreçtir. Tasarımın en başında verilen bir kararın sonuç ürüne nasıl bir katkı sağlayacağı her zaman öngörülemez. Bütün olası sonuçlara göre alternatifler üretmek ise meşakkatli, maliyetli ve uzun zaman gerektiren bir çözümdür. Ancak bu şekilde bir çalışma yapılmadığında elde edilen sonuç ürün tatmin edici olmazsa, çalışmanın en başına dönülmesi gerekeceğinden bu da aynı derecede vakit kaybettiren, zor ve maliyetli bir uygulama olacaktır. Ayrıca bu süreçte formla birlikte işlev, tarihsel bağlam, üretim teknolojileri, çevresel faktörler, ortak çalışılan diğer disiplinlerin gereksinimleri vb. gibi birçok değişkenin de hesaba katılması gerekmektedir. Böylece süreç tasarımcı için oldukça karmaşık bir hale gelmektedir. Bu nedenle bilgi işlemsel tasarım yaklaşımlarının da gelişmesi ile birlikte tasarımcının, başlangıç evresinde ve süreç boyunca birçok defa çeşitli simülasyonlar ve teknik araçlar kullandığı görülmektedir.

Her ne kadar yapay zekâ ve bilgisayar destekli tasarım yaklaşımları 1970’li yıllarda bilgisayar teknolojilerin gelişmesiyle yaygınlaşmış olsa da, bu yaklaşımların mimari tasarımda kullanımı daha eskilere dayanmaktadır. Bu yöntemlerin kullanılması ile birlikte, mimarlığın biyoloji, genetik, bilişim teknolojileri, matematik ve daha birçok disiplinle bilgi alışverişi güçlenmiştir. Evrimsel tasarım ve alt kollarından biri olan genetik algoritmalar da bu bilgi alışverişinden doğan yaklaşımlardır (Erdoğan ve Sorguç, 2011).

Genetik algoritmalar Darwin’in evrim teorisini temel alan bir arama ve problem çözme yöntemidir. Bu konu üzerine ilk çalışma 1975 yılında Michigan Üniversitesi’nde psikoloji ve bilgisayar bilimi uzmanı olan John Holland tarafından yapılmıştır. Holland evrim sürecinden etkilenmiş ve bu süreci bilgisayar ortamına aktararak mekanik yapıları benzer bir sürece tabii tutarak geliştirmeyi amaçlamıştır. Bu çalışması yayınlandığında bu yöntem

genetik algoritma olarak tanınmıştır (Holland, 1975:2). Bu yöntem bugüne kadar geliştirilerek üretim, tesis yerleşimi, çizelgeleme problemleri, inşaat teknolojileri, optimizasyon, araç rotalama, iş planlama, mimari vb. birçok alanda kullanılmıştır. Yöntemin temel mantığı her seferinde aynı olmakla birlikte, çözülmek istenen probleme ve elde edilmek istenen sonuca göre işlem adımları ve işleyiş şeması kullanıcı tarafından kurgulanır. Eldeki problem için gerekli adımları içerir ve çözüme ulaştıracak doğru soruları sorar. Bu nedenle her genetik algoritma modeli kendine özgüdür.

Bu çalışmada da amaç; genetik algoritma yöntemini kullanarak mimari form üretmeye yarayacak bir model hazırlamaktır. Bugüne kadar genetik algoritmanın mimari alanda kullanımı üzerine birçok çalışma yapılmıştır. Örneğin Çalışır (2015) “Olimpik Havuz Plan Şeması Tasarımında Genetik Algoritmaya Dayalı Bir Model” isimli tezinde mevcut yüzme tesislerinin plan şemalarını inceleyerek, alan, havuz tipi, seyirci sayısı ve diğer bazı özelliklerin kullanıcı tarafından belirlenmesiyle, bu özellikleri içeren optimum plan şemasını oluşturan bir genetik algoritma modeli hazırlamıştır (Çalışır, 2015). Güleç (2014) ise “Mimari Tasarım Alanında Kullanıcı Erişilebilirliğinin Genetik Algoritma İle Optimizasyonu - Ada: Sağlık Kampüsü Uygulaması” isimli çalışmasında sağlık kampüslerindeki kullanıcı erişilebilirliğinin optimizasyonu için bir genetik algoritma modeli hazırlamıştır (Güleç, 2014). Chouchoulas 2003 yılında hazırladığı Biçimin Evrimi çalışmasında ise biçim grameri ve genetik algoritmayı beraber kullanmıştır. Öncelikle temel bir apartman bloğu hazırlamış ve blokların birkaçının bir araya gelme kurallarını biçim grameri ile oluşturmuştur. Ancak bunların uygunluk değerini hesaplayabilmek için kullanıcının bazı kriterlere göre değerlendirmeler yapabileceği bir genetik algoritma modeli kullanmıştır (Güleç, 2014). Ayrıca arazi yerleşimi, konut plan ve cephe tasarımı, sürdürülebilir toplu konut tasarımı, tasarımlarda kullanıcı ihtiyaçlarının optimizasyonu vb. gibi birçok farklı tasarımda genetik algoritma üzerine çalışmalar yapılmıştır.

Ancak diğer yöntemlerden farklı olarak, hazırlanacak bu modelde, kendisi zaten bir optimizasyon problemi olan sudoku bulmaca temel alınmıştır. Amaç sudoku bulmacayı çözümleyerek bir üretim mantığı oluşturmaktır. Böylece sudoku kurallarını ve tasarımcı girdilerini temel alan, elde edilen üretim mantığını kullanarak mimari biçim üretme konusunda tasarımcıya yardımcı olacak bir GA algoritma modeli oluşturulmuştur. Sudoku bulmacayı temel aldığı için bu model SuGe modeli olarak isimlendirilmiştir.

Bu çalışmada öncelikle genetik algoritma hakkında geniş bir literatür taraması yapılmıştır. Genetik algoritmanın mantığı ve işleyişi, hangi alanlarda ne şekillerde kullanıldığı, mimari alandaki kullanımı ve mimari tasarıma katkısı araştırılmıştır. Bu kavramsal araştırma sonucunda yöntemin pratik kullanımını da görmek amacıyla mimaride bu yöntemin kullanıldığı örnekler incelenmiştir. Bütün bu araştırmaların sonucunda ise genetik algoritma kullanılarak mimari biçim üretimi yapacak bir model kurgulanmıştır.

Çalışma 4 temel bölümden oluşmaktadır.

1. bölüm, giriş bölümüdür. Bu bölümde çalışmada da çözmek istenen problem tanıtılacak çalışmanın amacı kapsamı ve yöntemi hakkında bilgi verilecektir.

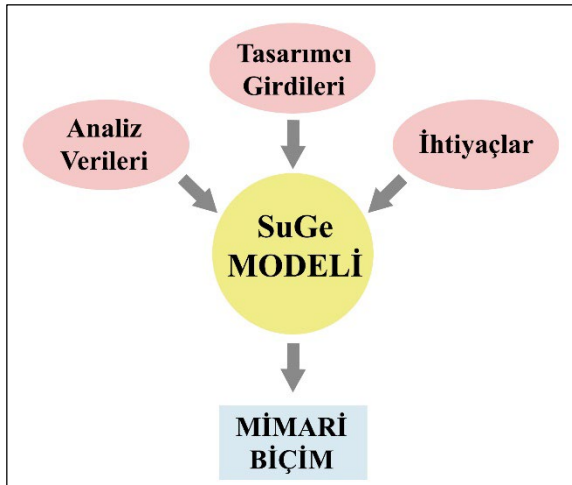
2. bölüm genetik algoritma yaklaşımı hakkında literatür taramasının yapılacağı bölümdür. Bu bölümde genetik algoritmanın tarihsel gelişimi, terminolojisi, işleyiş mantığı vb. temel öğeleri ele alınacaktır. Daha sonra bir algoritma modelinin nasıl yazılacağı, temel operatörleri ve parametreleri detaylı olarak ele alınacaktır. Genetik algoritmanın diğer yöntemlerden farkı, avantajları ve dezavantajları araştırılacaktır. Kullanım alanları ve bu alanlarda ne şekilde kullanıldığı üzerine bir literatür taraması yapılacaktır. En son ise mimari tasarımda genetik algoritma kullanımı üzerine bir araştırma yapılacak ve daha önce kullanıldığı mimari örnekler aracılığıyla pratik kullanımı incelenecektir.

3. bölüm modelin oluşturulacağı bölümdür. Bu bölüm iki aşamada ele alınacaktır.

- Birinci aşama sudoku bulmaca üzerine temel bilgilerin verildiği aşamadır. Bu aşamada sudoku hakkında bir literatür taraması yapılmış ve sudoku bulmacanın bir optimizasyon problemi olarak genetik algoritma yöntemi ile nasıl değerlendirilebileceğine dair bir örnek incelenmiştir.
- İkinci aşamada ise SuGe modelinin oluşturulması üzerine çalışılmıştır. Model aşaması ise kendi içinde dört aşamadan oluşmaktadır;
 - Analiz evresinde, sudoku bulmacanın temel mantığı ve çözüm yöntemi ele alınmıştır. Burada elde edilen veriler modelin kurgulanması için bir temel bilgi oluşturmaktadır ve diğer aşamalar için önem arz etmektedir.

- 1. Üretim evresinde, analiz evresinde elde edilen kurallar temel alınarak bir sudoku bulmacanın genetik algoritma yöntemiyle nasıl oluşturulabileceği adım adım incelenmiştir. Burada amaç GA ile sudoku bulmaca oluşturulurken kullanılacak üretim kurallarının nasıl olduğunu anlamaktır.
- 2. Üretim evresinde, analiz evresinde elde edilen kurallar alınmış ve bunlara tasarımcının girdileriyle birlikte, elde edilmek istenen sonuç ürüne dair veriler eklenerek bu aşamanın temel kısıtları ve kuralları oluşturulmuştur. Bu kurallarla başlanan modele 1. Üretim evresinde elde edilen üretim mantığı uygulanarak mimari biçim üretmeye dayalı SuGe modeli hazırlanmış ve modelin işleyişi tanıtılmıştır.
- Değerlendirme evresinde ise SuGe modeli kullanılarak hangi mimari biçimlerin ne şekilde üretilebileceğine, bunun için modelin nasıl kurgulanması gerektiğine dair örnek çalışmalar yapılmıştır.

SuGe modelinin kavramsal şeması Şekil 1.1’de gösterilmiştir.



Şekil 1.1. SuGe modeli kavramsal şeması

2. GENETİK ALGORİTMA KURAMI VE MİMARİ TASARIMDAKİ YERİ

Evrimsel algoritmaların temeli evrim teorisine dayanır. Evrimsel algoritmalar doğadaki en başarılı ve dikkate değer tasarımların meydana getirilmesi için kullanılan, genel amaçlı bir problem çözme ve optimizasyon yöntemidir. Evrim ve tasarım süreci incelendiğinde birbirlerine oldukça benzedikleri görülmektedir. Bu benzerlikten evrimsel tasarım yöntemi doğmuştur. Temel olarak evrimsel tasarım yöntemi ile tasarım problemlerine çözüm aranırken, evrim sürecine dayalı bir üretken mantık kullanılır (Akbulut, 2009). Genetik algoritmalar da benzer şekilde evrime dayalı algoritmaların bir türüdür.

2.1. Evrimsel Hesaplama

Evrimsel teori; 1840'larda Charles Darwin tarafından canlıların çevre koşullarının etkisiyle nesilden nesile kalıtsal değişimlere uğraması ve farklı özellikler kazanması olarak tanımlanmıştır (Shadmand, 2015). Charles Darwin evrim kuramını öne sürerken, evrimin sebebi olarak, "hayat kavgası dolayısıyla doğal ayıklanma" faktörünü kabul etmiştir. Doğal ayıklanma ve kalıtım, ona göre, yeni nesillerin meydana gelmesinde rol oynayan belli başlı iki faktördür (Öktem, 2010).

Evrimsel hesaplama ise bu doğal seçim faktöründen yola çıkılarak geliştirilmiş bir arama ve eniyileme algoritmasıdır (Aksoy, 2016). Evrimsel hesaplama tekniklerinin kökeni; 1950-1960 yıllarında bazı bilim adamlarının, evrimsel süreçlerin optimizasyonunun mühendislik problemlerinde uygulanabilirliği ile ilgili fikirlerine dayanır (Dianati, Song, ve Treiber, 2003).

Çalışmaları çok az ilgi görmesine ya da hiç ilgi görmemesine rağmen Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), Reed, Toombs ve Baricelli (1967) bu alanda çalışmalar yapmıştır. Doğal genetik çeşitlilikten ve doğal seçimden ilham alan operatörler kullanarak belirli bir sorunun çözüm alternatiflerinden bir popülasyon oluşturmayı hedeflemişlerdir (Mitchell, 1996:3).

Doğal seçim sayesinde çevreye en uyumlu dolayısıyla uygunluk değeri en yüksek olan birey seçilir. Amaç hayatta kalan bireylerden daha iyi özelliklere sahip yeni nesiller

oluşturmaktır. Aynı mantıkla evrimsel algoritmalar kullanılarak işlenen probleme en uygun çözümü bulacak bir arama yapılmaktadır. Bu şekilde nesilden nesile kötü çözümler yok olurken iyi çözümler daha optimize hale gelmektedir (Bentley, 1999:8).

Evrimsel tasarım sistemleri, tasarımcının yerini alması için değil tasarımcıya yardımcı olması için tasarlanmıştır. Ancak geleneksel yöntemlere nazaran tasarımcının rolünü belli ölçüde sınırlamıştır. Tasarımcı problemi ve çözüme dair gerekli kısıtlamaları belirler. Üretilcek çözümlerin temeli olacak bireyleri içeren ilk popülasyonu oluşturur. Süreç sonucunda ise en uygun bireyi, çözümü seçer. Dolayısıyla bu yöntemde tasarımcı süreci yürüten çok kontrol eden kişidir (Akbulut, 2009). Bu yöntemle üretkenliğin, kalitenin ve hızın artırılması ve tasarım maliyetlerinin düşürülmesi hedeflenmiştir (Mitchell, 1996:11).

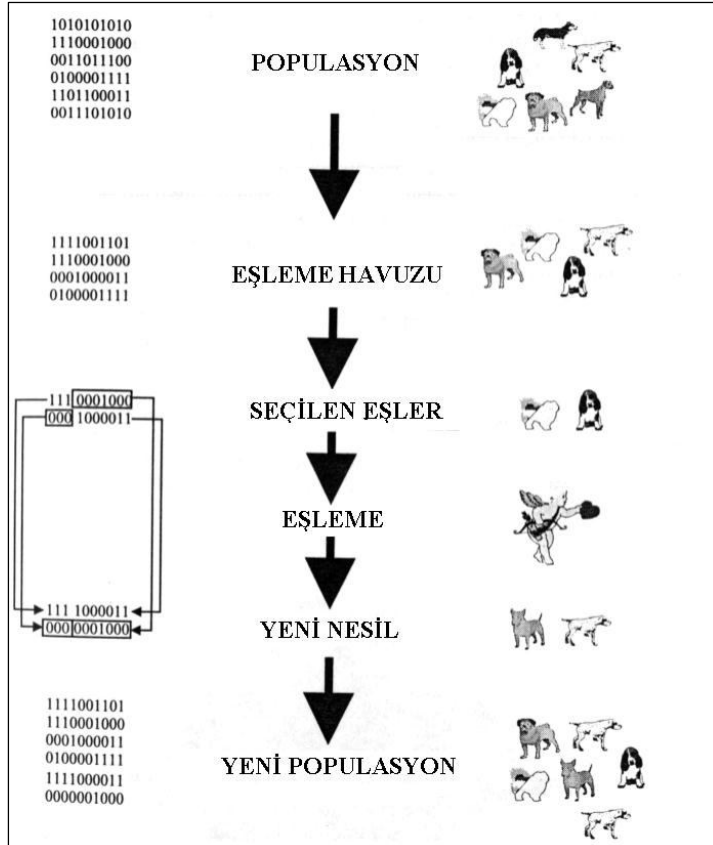
Evrimsel algoritmalar genetik algoritmalar, genetik programlama, evrimsel programlama ve evrimsel strateji gibi alt başlıklara ayrılmıştır (Bentley, 1999:7). Tez kapsamında genetik algoritmalar üzerine çalışılacaktır.

2.2. Genetik Algoritmalar

Genetik algoritmalar, evrimsel mimari yaklaşımının alt kavramlarından biridir. Darwin'in en iyi olan hayatta kalır prensibinden yola çıkarak doğadaki biyolojik süreçleri taklit eder. Çevreye uyum sağlayanın soyunu devam ettirebilmesi, üstün nitelikli ebeveynlerden daha üstün bireylerin elde edilmesi gibi fikirler, karışık bir problemin çözümünde bilinen bazı farklı çözümlerin çaprazlanması ile daha iyi bir çözüm elde edilebileceği fikrini oluşturmuştur ve ilk olarak 1958'de Bremermann tarafından geliştirilmiştir (Vural, 2005).

Bu kavramla ilgili ilk yayını 1967 yılında Bagley yapmıştır (Goldberg, 1989:92). Ancak bu konuda ilk çalışmaları 1975 yılında Michigan Üniversitesi'nde psikoloji ve bilgisayar bilimi uzmanı John Holland yapmıştır. Holland makine öğrenmesi üzerine yaptığı çalışmalarda evrim sürecinden etkilenmiş ve bu süreci bilgisayar ortamına aktarmıştır. Bu şekilde mekanik yapıların evrim sürecini taklit eden bir yöntemle geliştirilebileceği üzerine yaptığı çalışmaların sonucunu yayınladığında bu yöntem Genetik Algoritmalar olarak tanınmıştır (Holland, 1975:2).

Ancak 1989 yılında Holland'ın öğrencisi David Goldberg'in bu konu üzerine yazdığı "Genetic Algorithm in Search Optimization and Machine Learning" kitabı yayınlanana kadar bu yöntemin sağlayacağı yararlar ve pratik kullanımları tam olarak keşfedilememiştir (Dianati, Song ve Treiber, 2003; Goldberg, 1989).



Şekil 2.1. İkili kodlu GA ile biyolojik evrim arasındaki benzetim (Cunkaş, 2006)

Karmaşık ve çözümü birçok etkene bağlı problemlerde en iyi çözümü bulmayı hedefleyen GA'lar tek bir olası çözüm yerine, bütün olası çözümleri içeren bir popülasyon oluşturup en iyi çözüme ulaşmaya çalışırlar (tr.wikipedia.org).

GA bazı varsayımlar üzerine çalışır. Bunlar;

- Bütün bireyler mücadele halindedir ve sadece bazıları bir sonraki nesile aktarılır.
- Nesiller boyunca tekrar eden bir yapının gelecek nesillere aktarılması daha olasıdır.
- Belirlenen başlangıç popülasyonu nesillerin devamlılığı için bir temel niteliğindedir (Akpınar, 2009).

GA'lar aşağıdaki özellikleri taşıyan paralel rastlantısal ve sezgisel arama optimizasyonlarıdır; (Özçakar, 1998)

- Bir problemin çözümü için bir veya daha fazla popülasyon oluşturulabilir.
- Bilinen bazı çözümlerin özelliklerini kullanarak daha iyi bir çözüm üretilebilir.
- Bilinen bir çözümü rastgele değiştirerek yeni bir çözüm elde edilebilir.
- Eniyileme değerleri verilerek popülasyondaki daha uygun bireylerin seçilmesi sağlanabilir.
- Uygun olmayan bireyler popülasyon dışında bırakılabilir.

2.2.1. Genetik algoritma terminolojisi

Genetik algoritmalar hem genetik bilimini hem de bilgisayar bilimini temel alır. Bu nedenle GA terminolojisi iki bilimden de bazı terimler içeren karışık bir terminolojidir. Dolayısıyla GA'nın anlaşılabilmesi için bu terimlerin arasındaki ilişki anlaşılmalıdır (Vural, 2005).

Gen: Gen terimi Danimarkalı bitki bilimci Wilhwlm Johannsen tarafından 1950 yılında üreme hücreleri içerisinde bulunan kalıtsal özellikleri belirleyen birimleri açıklamak için türetilmiştir (Erbaş Korur, 2012).

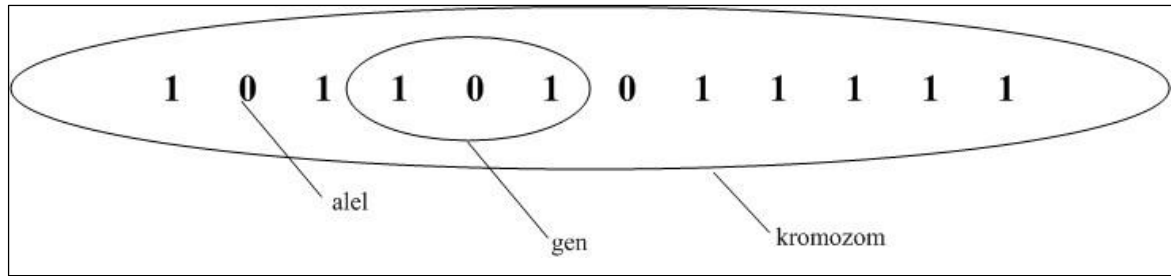
Genetik biliminde kendi başına anlamı olan ve genetik bilgi taşıyan en küçük genetik birim olarak ifade edilir. Kısmi bilgi taşıyan bu küçük yapıların bir araya gelmesiyle tüm bilgileri içeren kromozomlar meydana gelir (Güngör, 2010). GA'da ise gen bit olarak ifade edilir ve çözümün her bir özelliğine karşılık gelir. Bir gen A,B gibi bir karakter olabileceği gibi 0 veya 1 ile ifade edilen bir bit veya bit dizisi olabilir. Tamamen yazılacak algoritmaya bağlı olarak istenilen şekilde ifade edilebilirler (Karasoy ve Ballı, 2016).

Kromozom: Bir ya da birden fazla genin bir araya gelmesiyle oluşurlar ve genetik bilimine göre birleşerek toplumdaki bireyleri oluştururlar (Karasoy ve Ballı, 2016).

GA'da ise genlerin birleşmesiyle oluşmuş bir dizi olarak ifade edilir ve çözüm kümesinin bir üyesi olarak tanımlanabilir (Engin, 2001). Kromozomlar, genetik algoritma yaklaşımında üzerinde durulan en önemli birim olduğu için bilgisayar ortamında iyi ifade edilmeleri gerekmektedir. Kromozomun hangi kısmının ne anlam taşıyacağı, ne tür bilgi içereceği

kullanıcının olaya bakışını değiştirmektedir (Dil, 2015). GA'da genellikle kromozomların birleşmesi ile ilk popülasyon oluşturulur. Dolayısıyla yerine göre bir kromozom bir bireyi ifade edebilir.

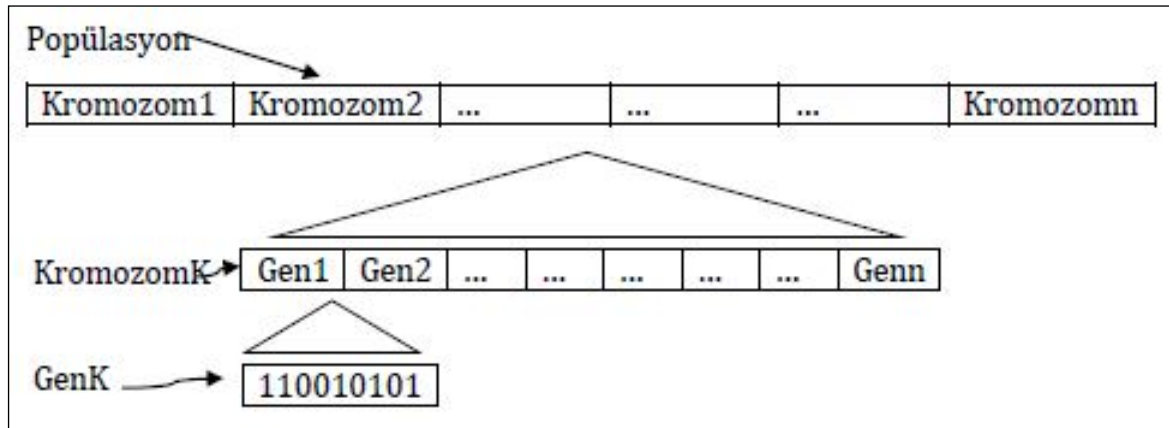
Allel: Her bir genin alabileceği değerleri ifade eder. GA'da ise her bir özelliğin değeridir (Vural, 2005).



Şekil 2.2. Allel, gen ve kromozom (Çakar, 2009)

Lokus: Her bir genin kromozom üzerindeki yerini ifade eder. GA'da ise her bir özelliğin pozisyonu olarak değerlendirilir (Çalışır, 2015).

Popülasyon: Kromozomlar veya bireyler topluluğudur. GA'da işlenen problemin alternatif çözümler kümesi olarak ifade edilebilir (Güngör, 2010). Popülasyonda kaç birey olacağı ile ilgili bir kısıtlama bulunmamaktadır. Ulaşılmak istenen çözüme göre bir sayı belirlenmelidir. Ancak popülasyondaki birey sayısı en başta belirlenir ve işlemin sonuna kadar sabit tutulur. Birey sayısı arttıkça çözüme ulaşma süresi azalır (Çalışır, 2015).

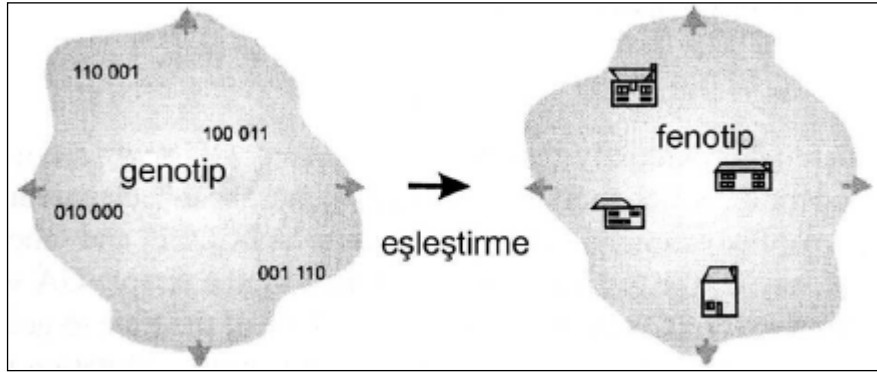


Şekil 2.3. Gen, kromozom, popülasyonun ilişkisi (Dil, 2015)

Genotip: Bir insanda bulunan genler o kişinin genetik özelliklerini yani genotipini belirler (Güngör, 2010). Genotip genetik biliminde kromozomların birleşmesiyle meydana gelmektedir ancak GA’da bireyler genellikle tek kromozomdan oluştuğu için kromozomun içerdiği bilgiler yani bireyler aynı zamanda genotip olarak tanımlanabilir (Çalışır, 2015). Genetik değişimler genotip seviyesinde gerçekleşir (Badem, 2007).

Fenotip: Genotipin dışavurumudur. Genetik biliminde saç rengi, göz rengi vb. özelliklerin bir gen olarak kodlanması genotip iken bunların bireyde mavi göz sarı saç vb. şekilde görünür olmuş haline ise fenotip denilir (Çalışır, 2015). Yani genotipin somutlaştırılmış hali olan bir görüntüdür ve genotipin sonucu ancak fenotip boyutunda yorumlanabilir (Badem, 2007).

GA’da ise aynı mantıkla deşifre edilmiş çözüm olarak tanımlanabilir (Vural, 2005). Bir ev örneği dikkate alındığında pencere, kapı, çatı vs. öğelerin yerleri fenotipte ifade edilirken bunların kodlama ile verilen karşılığı genotipleri ifade eder (Bentley, 1999:9).

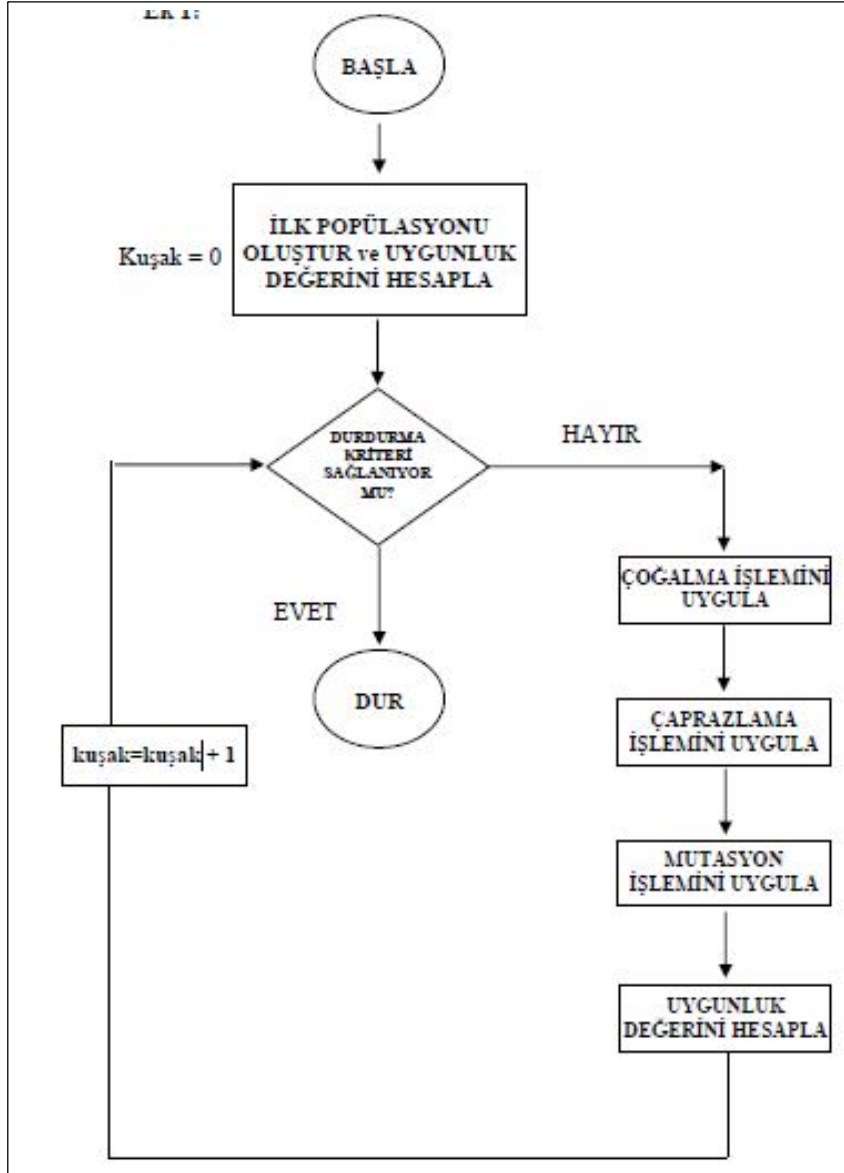


Şekil 2.4. Genotipler ile fenotiplerin eşleşmesi (Bentley, 1999:9)

2.2.2. Genetik algoritmanın işleyişi

Genetik algoritmalar çözülmek istenen probleme göre tasarlanırlar. Bu nedenle tek bir standart algorithmadan bahsetmek mümkün değildir. Ama temel işlem sırası birçok algorithmada aynı şekildedir (Erdoğan, 2007). Bütün algoritmalarda amaç her adımda bir önceki nesilden daha yüksek uygunluk değerine sahip bireyler üretmektir (Vural, 2005).

Şekil 2.5 de GA'nın akış diyagramı gösterilmiştir. Genel işleyiş mantığı ise altı adımda açıklanmıştır (Gökay Emel ve Taşkın, 2009:31).



Şekil 2.5. Genetik algoritma akış diyagramı (Gökay Emel ve Taşkın, 2009:32)

1.Adım: Arama uzayında tüm çözümler kodlanır ve bunlardan bir başlangıç popülasyonu oluşturulur. Biyolojiyi temel aldığından bu gruba popülasyon (toplum), her bir çözüme birey ya da yerine göre kromozom adı verilir (Çakar, 2009). Popülasyon büyüklüğü için belirli sabit bir sayı yoktur. Ancak çeşitliliği olabildiğince yansıtılmalıdır. Bu nedenle popülasyon büyüklüğü bir düzine olabileceği gibi birkaç bin de olabilir. Bu değer genellikle çözmek istenen problemin çözüm parametrelerine göre belirlenir (Gökay Emel ve Taşkın, 2009:72; İşçi ve Korukoğlu, 2003).

2. Adım: Popülasyondaki çözümlerin ne kadar iyi olduğu hesaplanır. Bulunan değere uygunluk değeri denir. Optimum çözümü bulmaya yardımcı önemli bir adımdır. Her bir kromozomun şifresini çözerek problemin çözümünde gerekli parametrelere değerler verilir. Bu değerlere göre ise her kromozom için bir uygunluk değeri belirlenmiş olur (Çakar, 2009). Bu değer her GA modelinde kendine özgüdür ve çoğunlukla bu adımın verimli ve hassas olması algoritmanın başarısında büyük rol oynar (İşçi ve Korukoğlu, 2003).

3. Adım: Bireyler eşlenerek bir grup birey bir sonraki aşama için seçim yöntemleri ile çoğaltılır. Çeşitli seçim yöntemleri vardır (Gökay Emel ve Taşkın, 2009:31; İşçi ve Korukoğlu, 2003).

4. Adım: Yeni bireylerin uyum değeri hesaplanır ve bu bireyler çaprazlama ve mutasyon işlemlerine tabi tutularak yeni popülasyonlar elde edilir (Gökay Emel ve Taşkın, 2009:31).

5. Adım: Durdurma kriterini sağlayana kadar bu işlemler tekrar edilir.

6. Adım: Durdurma kriteri sağlandığında işlem durdurulur ve amaca en uygun çözüm seçilir (Gökay Emel ve Taşkın, 2009:32).

Anlatılan işleyiş adımlarında genetik algoritma en az üç aşamada olasılığa dayalı adımlar içermektedir. Bu adımlar: (Çakar, 2009)

- Başlangıç popülasyonunun oluşturulması
- Genetik operasyonlar için popülasyondan bireyler seçilmesi
- Çaprazlama veya mutasyon gibi genetik operatörleri uygulamak için seçilen birey üzerinde bir nokta belirlenmesidir.

Bu işlemlerin yetersiz kaldığı noktalarda algoritmada çözüme özel değişiklikler yapılabilir. Bir önceki adıma geri dönülebilir. Algoritma daha erken ya da daha geç bitirilebilir. Bu değişiklikler problemin cinsine, optimum çözüm fikrinin karşılığına ve kullanıcın beklentilerine bağlı olarak, yine kullanıcı kontrolünde gerçekleştirilmelidir (Erdoğan, 2007).

2.3. Genetik Algoritma Operatörleri

Genetik Algoritmanın işleyişi sırasında yeni nesiller oluşturulurken uygulanan işlemlere genetik operatörler denir. Bu operatörler sayesinde çözüm uzayı genişletilir ve daha uygun çözümlere ulaşılması sağlanır (Akbulut, 2009). Bunlar; parametre kodlama, uygunluk değeri hesaplama, başlangıç popülasyonu oluşturma, seçim, çaprazlama, mutasyon ve durdurma operatörleridir.

2.3.1. Parametre kodlama

Gerçek hayatta genetik işlemler genler seviyesinde gerçekleşir. Aynı şekilde genetik algoritma da, araştırma uzayındaki değerler üzerinden değil, bu değerlerin bazı kodlanmış karşılıkları (diziler) üzerinden çalışır (Vural, 2005).

Yani GA genotip ile fenotip arasındaki ilişkiye benzer şekilde kodlar ve bunların karşılığı olan çözümler üzerinde işlem yapar (Engin, 2001). Kodların çözümler şeklinde tanımlanmasına şifreleme, çözümlerin kodlara dönüştürülmesine ise şifre çözme denir. Şifreleme için sayı veya harfler kullanılabilir ancak genellikle bireyler 0 ve 1 ler den oluşan diziler şeklinde şifrelenmektedir (Vural, 2005). GA'da kromozomların kodlanma biçimi çok önemlidir. Bu aşamada üç notaya çok dikkat edilir; (Cheng, Gen ve Tsujimurab, 1996)

1. Kromozomun uygun olup olmadığı
2. Kromozomun belirlenen şartları sağlayıp sağlamadığı
3. Kromozomun kodlama haritasında tek olması

Kodlama teknikleri ise çözülmek istenen probleme göre farklılık gösterir (docplayer.biz.tr). Son dönemlerde birçok farklı kodlama biçimi geliştirilmiştir (Cheng, Gen ve Tsujimurab, 1996). En sık kullanılanları ikili kodlama, gerçek değer kodlama ve permutasyon kodlamadır (Karasoy ve Ballı, 2016).

İkili Kodlama; her kromozom 1 ve 0'lardan oluşan bir dizi olacak şekilde yapılan kodlamadır (docplayer.biz.tr).

Kromozom1	1101100100110110
Kromozom2	1101111000011110

Şekil 2.6. İkili kodlamada ile hazırlanmış kromozom örnekleri (docplayer.biz.tr)

Ancak tam sayı uzayında komşu olan bazı değerler bit uzayında komşu değildir. Örneğin 15 ve 16 tam sayı uzayında ardışık sayılarken bit uzayında 01111 ve 10000 olarak kodlanırlar. Bu nedenle ikili kodlama kullanıldığında kromozomlar ilk oluşturulurken ya da operatörler sonucu uygunluk değerinin hesaplanması gerektiğinde, bit değerlerinin tam sayı karşılıklarına çevrilerek kullanılması gerekir. Bu da sürecin uzamasına ve iş yükünün artmasına sebep olacaktır (Vural, 2005). Bu yöntemde sayılar adından da anlaşılacağı gibi kodlama işlemine girmez, oldukları gibi kullanılırlar.

Gerçek Değer Kodlama; gerçel sayılar gibi kompleks sayıların yer aldığı problemlerde kullanılır (Karasoy ve Ballı, 2016). Bu yöntemde sayılar olduğu gibi kullanılır kodlama işlemine dâhil edilmez (Erdoğan, 2007). İkili kodlamanın dezavantajlarını ortadan kaldıran bir yöntemdir. Bu yöntemle problemin sadece ana değişkenleri genlere aktarılır. Diğer değişkenler ana değişkenlere bağlı olarak hesaplanabildiğinden genlere aktarılmamıştır (Vural, 2005).

Bu şekilde gereksiz işlemler bilgisayar hafızasından çıkarılarak çözüm uzayının büyümesi engellenmiştir (Stockton, Quinn ve Khalil, 2004).

Kromozom1	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom2	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom3	(back), (back), (right), (forward), (left)

Şekil 2.7. Gerçek değer kodlama ile hazırlanmış kromozom örnekleri (docplayer.biz.tr)

Permutasyon Kodlama; her kromozom, ilgili karakterin sıralamadaki pozisyonunu belirten sayılardan oluşan bir dizi ile ifade edilir (docplayer.biz.tr).

Örneğin çizelgeleme problemlerinde süreçlerin bağıl sırası önemli olduğundan permutasyon kodlama kullanılmaktadır (Vural, 2005). Bu kodlama türü pozisyonların önemli olduğu Gezgin Satıcı problemlerinde ve iş sıralama problemlerinde de kullanılmaktadır (Karasoy ve Ballı, 2016).

Kromozom1	1 5 3 2 6 4 7 9 8
Kromozom2	8 5 6 7 2 3 1 4 9

Şekil 2.8. Permutasyon kodlama ile hazırlanmış kromozom örnekleri (docplayer.biz.tr)

Bazı başka kodlama yöntemleri aşağıdaki gibidir; (Cheng, Gen ve Tsujimurab, 1999)

- a) Operasyona dayalı kodlama
- b) İşe dayalı kodlama
- c) Liste önceliğine dayalı kodlama
- d) İş çifti ilişkisine dayalı kodlama
- e) Öncelik kuralına dayalı kodlama
- f) Tamamlanma zamanına dayalı kodlama
- g) Makineye iş yükleme grafiğine göre kodlama
- h) Rassal anahtarlı kodlama
- ı) Karışık kodlama

2.3.2. Uygunluk değeri hesaplama ve başlangıç popülasyonunu oluşturma

Uygunluk Fonksiyonu: Nesiller değiştikçe daha iyi bir sonuç elde edilebilmesi için rastgele bireyler seçmek yerine daha sağlıklı ve kaliteli çözümler seçilebilir. Bunun için çözümün barındırdığı her bir gen için, istenilen çözüme uygunluğuna göre bir değer verilir (Güngör, 2010). Toplamda bireyin çözüm kalitesini gösteren bu değere uygunluk fonksiyonu denir. Sonuç ürünün iyi olup olmadığına ve gelecek nesile aktarılıp aktarılmayacağına uygunluk değerine bakılarak karar verilir (Badem, 2007; Akpınar, 2009).

Maksimizasyon problemlerinde dizinin uygunluk fonksiyonu amaç fonksiyonuna eşittir. Bir çözümün uygunluk değeri ne kadar yüksekse çoğalma ve bir sonraki nesile aktarılma olasılığı o kadar artacaktır (Yeniay, 2001). Ancak minimizasyon problemlerinde amaç en küçük değere ulaşmak olduğundan amaç fonksiyonu ile uygunluk fonksiyonu ters orantılıdır (Gökay Emel ve Taşkın, 2009:37).

Kromozomlardan oluşan alternatif çözüm kümesine ise popülasyon denir. İlk popülasyon algoritmanın temelini oluşturur. Genellikle ilk popülasyon rastgele oluşturulur. Ancak

uygunluk deęerleri hesaplanıp buna g re seimlerde yapılmaktadır. Bu  ekilde sonuca gelindięinde istenilene en uygun   z m  bulmak daha olasıdır. Pop lasyonun m mk n olduęu kadar  ok  eřitli   z mleri temsil etmek iin yeterli arama alanına yayılması gerekir. Pop lasyon b y kl ę  bu noktada  nemli bir etkindir. Ka bireyden oluřacaęına   z lmek istenen probleme baęlı olarak iřlemin bařında karar verilir ve bu sayı s re boyunca sabit tutulur (Yeniay, 2001).

2.3.3.  reme ve seim

Seim operat r , bir  nceki pop lasyondan hangi bireylerin sonraki pop lasyona aktarılacaęını belirlemeye yarayan operat rd r (Erdoęan, 2007). Doęal evrimsel s rete sadece en iyiler hayatta kalır ve sadece en iyi  zellikler bir sonraki nesile aktarılır. GA da benzer bir seim mantıęı kullanır (G ng r, 2010).

Pop lasyondaki her bir  zellięe bir bařlangı deęerleri verilir ve bu deęerlere g re uygunluk deęeri hesaplanır (Erdoęan, 2007). Bu uygunluk deęerine g re seim yapılır. Uygunluk deęeri y ksek olanın bir sonraki nesile aktarılması daha olasıdır. Bu operat r Darwin'in doęal seim operat r n n yapay bir modelidir (Goldberg, 1989:10).

Seme kriterlerinde uyumluluk esastır. Bu  ekilde daha sonra  aprazlama ve mutasyonlar sonucu daha uyumlu bireyler ortaya  ıkacaktır. B t n bireylerde uyumluluk g zetilebileceęi gibi rastgele seimde yapılabilir (Kurt ve Semetay, 2001). Bu iřlemler sonucu seilemeyen bireyler yok olur (Yurtu ve  aęa, 2006).

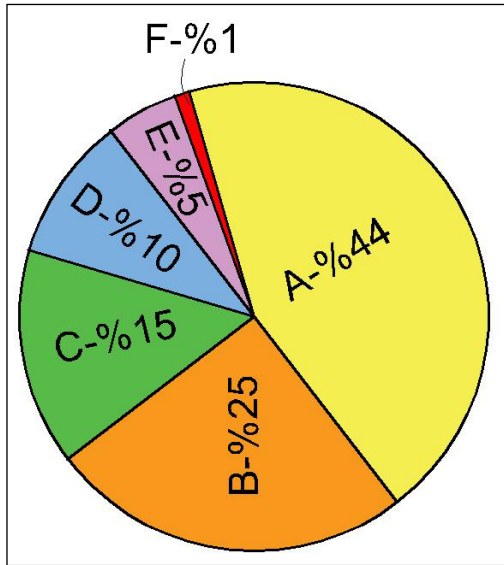
 reme ise, seilmiř bireyleri bir eřleme havuzuna kopyalama iřleminde ve havuzda bireyleri  iftler halinde gruplara ayırma iřleminde oluřur (Engin ve Fıęlalı, 2002).

Seim iin  eřitli y ntemler bulunmaktadır. Bunlardan bazıları rastgele seim, rulet tekerleęi seimi, turnuva seimi, sıralama seimi, sabit durum seimi ve elitizm seimidir (G ng r, 2010; Horat, 2014).

Rastgele Seim: Pop lasyondaki bireylerin uygunluk deęerini hesaba katmadan, herhangi bir kıstasa baęlı kalmadan rastgele yapılan seim y ntemidir (G ng r, 2010).

Rulet Tekerleği Seçim Yöntemi: Amaç uygunluk değeri yüksek olan bireylerin ebeveyn seçilebilme ihtimalini arttırmaktır (Horat, 2014). Popülasyondaki birey çeşitleri bir rulet üzerine yerleştirilir. Rulet üzerinde kaplayacakları alan uygunluk değerleri ile doğru orantılı olacak şekilde belirlenir (Akpınar, 2009). Dolayısıyla uygunluk değeri en yüksek olan birey rulette en büyük payı alacaktır. Seçim işlemi için bir adet bilye rulet üzerine fırlatılır ve durduğu aralıktaki birey seçilir (Güngör, 2010). Bu yöntemde her bireyin seçilme olasılığı kendi uygunluk değerinin toplam uygunluk değerine oranı kadardır (Karasoy ve Ballı, 2016). Bu şekilde yeni jenerasyonların ebeveyni olacak bireylerin uygunluk değeri yüksek olan bireylerden seçilme olasılığı artacaktır. Ancak uygunluk değeri düşük olan bireylerin de seçilme ihtimali vardır (Kevran, 2009).

Basit bir örnekle anlatacak olursak uygunluk değerleri sırasıyla 44, 25, 15, 10, 5 ve 1 olan, A,B,C,D,E ve F olarak isimlendirdiğimiz 6 bireyden oluşan bir popülasyon için hazırlanan rulet tekerleği Şekil 2.9.'daki gibi olacaktır.



Şekil 2.9. Rulet tekerleği örneği

Bu örnek üzerinden algoritma adımları şöyle açıklanabilir:

[Toplam] Popülasyondaki tüm bireylerin uygunluk fonksiyonlarını topla – toplam 100.

[Seçim] $(0,100)$ aralığında rastgele bir sayı seç - 58.

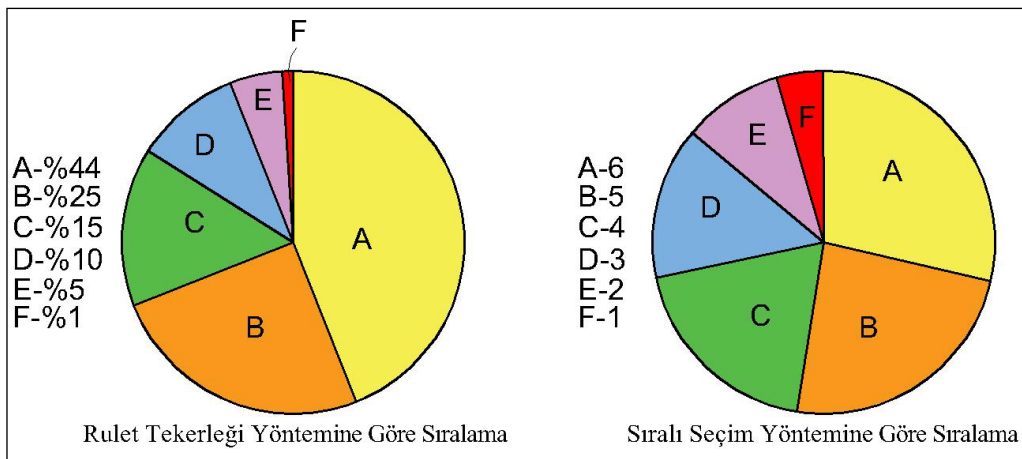
[Döngü] Popülasyon üzerinden gidip 0'dan başlayarak bireylerin uygunluk değerlerini topla – toplamın belirlenen sayıdan (58) büyük olduğu noktada dur. (A+B=67 olduğundan B

kromozomunda durulur.) Bulunulan yerdeki bireyi seç (B kromozomu seçilir) (Güngör, 2010). Dolayısıyla seçilme ihtimali en fazla olan birey A, en az olan birey ise F'dir.

Sıralı Seçim: Rulet tekerleği yöntemine benzer bir yöntemdir. Ancak rulet tekerleği yöntemindeki dezavantajlara çözüm üretmiştir. Rulet tekerleği yönteminde eğer bir çözümün uygunluk değeri örneğin %85 ise diğer çözümlerin seçilme ihtimali çok azalacaktır. Sıralı seçimde ise bu olumsuzluğu engellemek için popülasyondaki n kadar birey uygunluk fonksiyonuna göre sıraya dizilir. Uygunluk fonksiyonu en düşük olan bireye 1, bir sonrakine 2 numara verecek şekilde numaralandırılır. En uygun birey n sayısını alır (Güngör, 2010; Kurt ve Semetay, 2001). Bireyler verilen bu değerler dikkate alınarak rulet tekerleği üzerine yerleştirilir ve rulet tekerleği yöntemiyle aynı şekilde seçim yapılır.

Bu yöntemde iyi olan bireyin seçilme ihtimali yine en fazladır ancak uygunluk değerleri arasındaki büyük farklar ortadan kaldırılarak, uygunluk değeri nispeten düşük olan bireylere de seçilme şansı tanınır (docplayer.biz.tr).

Örneğin Şekil 2.9'da A,B,C,D,E ve F bireylerini uygunluk değerlerine göre sıralamıştık. Aynı bireylere sıralı seçim yöntemiyle sıralayalım.



Şekil 2.10. Rulet tekerleği ve sıralı seçim yöntemlerinin karşılaştırılması

Şekil 2.10'da görüldüğü üzere iki yöntemde de seçilme ihtimali en yüksek olan birey uygunluk değeri en yüksek olan A bireyi, seçilme ihtimali en düşük olan birey ise uygunluk değeri en düşük olan F bireyidir. Ancak A dışındaki bireylerin seçilme ihtimali sıralı seçim yönteminde rulet tekerleği yöntemine göre daha fazladır.

Sabit Durum Seçimi: Bu yöntemin amacı kromozomların çoğunluğunun yeni nesile aktarılmasını sağlamaktır (Güngör, 2010). İki adımdan oluşur. Öncelikle popülasyondaki uygunluk değeri yüksek olan bireyler seçilerek bu bireylerden yeni nesil çocuk bireyler oluşturulur. İkinci evrede popülasyondaki düşük uygunluk değerine sahip bireyler çıkarılarak yerlerine yeni oluşturulan bireyler konulur. Bu şekilde popülasyon büyüklüğü değişmez ancak popülasyondaki uygunluk değeri yüksek olan bireyler korunurken, diğer bireylerin de uygunluk değeri yükselmiş olur (Gökay Emel ve Taşkın, 2009:48).

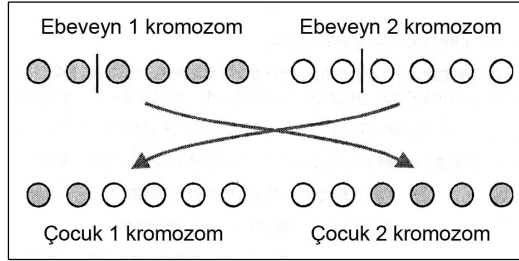
Turnuva Seçimi: Bu yöntemde öncelikle popülasyondan iki adet birey seçilir ve bu iki bireyin uygunluk değeri karşılaştırılır. Uygunluk değeri yüksek olan birey seçilirken diğer birey elenir. Popülasyondaki bütün bireyler karşılaştırılmaya tabii tutulana kadar işlem devam ettirilir. Ancak bu yöntemin bazı dezavantajları vardır. İkili seçim sırasında uygunluk değeri nispeten yüksek iki birey karşı karşıya geldiğinde bunlardan biri elenecektir. Yine aynı şekilde uygunluk değeri nispeten düşük iki birey karşı karşıya geldiğinde bunlardan biri yeni popülasyona dâhil edilecektir. Dolayısıyla popülasyondaki kaliteli bazı çözümler yok olabilir ya da çok düşük kalitede bazı çözümler yeni popülasyona dâhil olabilmektedir (Gökay Emel ve Taşkın, 2009:47).

Elitizm Yöntemi: Bu yöntemde amaç popülasyonun en uygun bireylerini kaybetmemektir. Bu nedenle popülasyondaki uygunluk fonksiyonu en yüksek olan bireyler olduğu gibi bir sonraki nesile aktarılırlar. Diğer bireyler arasında ise uygunluk değerleri dikkate alınacak bir yöntemle seçim yapılır ve yeni popülasyonun bireyleri tamamlanır. Bu yöntemi avantajı bir örnekleme hatası veya genetik operatörün kullanılması sonucu uygunluk değeri yüksek bireylerin kaybolmasının önüne geçmesidir. İlk defa 1975 yılında De Jong tarafından önerilmiştir (Coşkun, 2006).

2.3.4. Çaprazlama operatörü

En önemli operatörlerden biridir. Mevcut gen havuzunun potansiyelini arttırmak için iki farklı kromozomun genetik bilgilerinden yeni bir nesil oluşturulmasını sağlar (docplayer.biz.tr). Farklı bireylerin genetik parçalarını kombine eder (Vural, 2005). Böylece zaten uygunluk değeri yüksek olan bireylerden uygunluk değeri daha yüksek bireyler elde edilebilir. İki amacı vardır. Birincisi çözüm uzayını aramak, ikincisi bu aramayı ebeveynlerin mevcut genetik bilgilerini koruyarak yapmaktır (Gökay Emel ve Taşkın,

2009:49). Böylece oluşan çocuk birey ebeveynlerinin bazı özelliklerini alacaktır ve ebeveynlerinden farklı bir genotipe sahip olacaktır (Güngör, 2010).



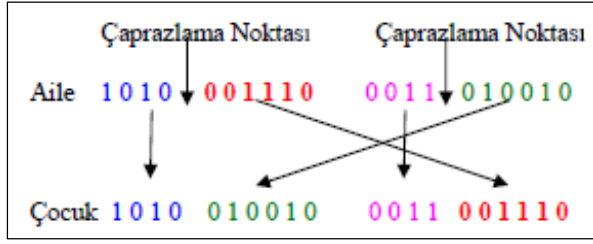
Şekil 2.11. Basit çaprazlama operatörü (Bentley, 1999:10)

Çaprazlama operatörleri çaprazlanacak ebeveyn sayısına göre üç temel sınıfa ayrılabilir (Çakar, 2009).

- Yeni birey tek ebeveyninden üretiliyorsa aseksüel,
- Yeni birey iki ebeveyninden üretiliyorsa seksüel,
- Yeni birey ikiden fazla ebeveyninden üretiliyor ise çoklu-birleşim (muti-recombination) adını alır.

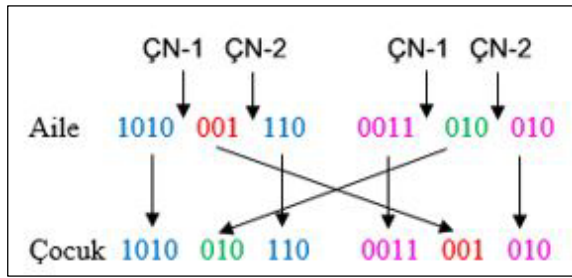
Literatürde farklı problemlere uygun farklı çaprazlama yöntemleri mevcuttur. Bunlardan bazıları tek noktalı çaprazlama, iki noktalı çaprazlama çok noktalı çaprazlama ve uniform çaprazlamadır.

Tek Noktalı Çaprazlama: GA'da en çok kullanılan çaprazlama yöntemidir. Bazı kaynaklarda basit çaprazlama olarak da geçer. Çaprazlanacak bireylerin dizi uzunluklarının eşit olması gerekir. Diziler üzerinde rastgele bir çaprazlama noktası belirlenir ve bu noktadan sonra gelen genler karşılıklı olarak yer değiştirilir (Vural, 2005). Çaprazlama noktasının doğru seçilmesi önemlidir. ÇN'na göre çok uygun bireyler ya da uygunluk değeri çok düşük bireyler elde edilebilir (Çakar, 2009). Uzun bit zincirine sahip diziler için çok kullanışlı değildir. Çünkü bu yöntemle fazla çeşitlilikte kombinasyonlar oluşturmak mümkün değildir (Erdoğan, 2007).



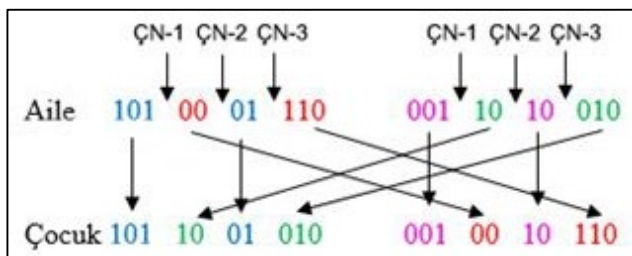
Şekil 2.12. Tek noktalı çaprazlama (İşçi ve Korukoğlu, 2003)

İki ve Çok Noktalı Çaprazlama: İki noktalı çaprazlamada iki çaprazlama noktası belirlenir ve bu noktalar arasındaki genlerin yeri değiştirilir (Güngör, 2010). Tek noktalı çaprazlamada dizinin ya baş ya da son bölümü çocuk bireye aktarılır. İkisi birden aktarılamaz. Ancak dizinin hem başında hem sonunda iyi genetik bilgi içeren bölümler varsa tek noktalı çaprazlama bütün iyi genleri alamayacaktır. Bu tarz durumlarda çift noktalı çaprazlama daha iyi sonuç verecektir (Çakar, 2009).



Şekil 2.13. İki noktalı çaprazlama

Çok noktalı çaprazlama ise bu yöntemin geliştirilmiş halidir. Birden fazla çaprazlama noktası belirlenir ve bu noktalar arasında kalan bölgeler karşılıklı olarak yer değiştirilir (Güngör, 2010). Çaprazlama noktası sayısı tek olduğunda değişim olasılığı asimetrik biçimde bit pozisyonuna bağlı olarak değişir. Birçok durumda simetri nedeni ile küçük çift sayılı çaprazlama nokta sayısı belirlenmesi önerilmektedir (Çakar, 2009).



Şekil 2.14. Çok noktalı çaprazlama

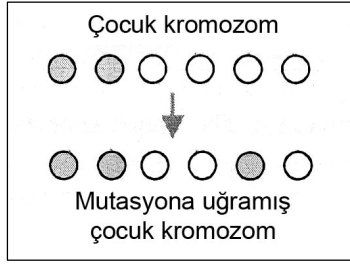
Uniform Çaprazlama: Genetik algoritmanın gücünü arttırmak için geliştirilmiştir. Öncelikle çaprazlanacak bireylerle aynı dizi uzunluğuna sahip, rastlantısal bir çaprazlama maskesi oluşturulur. Bu çaprazlama maskesinde 1 yazan gen yerine aile 1'den, 0 yazan gen yerine ise Aile 2'den aynı sıradaki gen alınarak Çocuk 1 oluşturulur. Daha sonra çaprazlama maskesinde 1 yazan geni 0, 0 yazan gen 1 olarak değiştirilerek çaprazlama maskesinin tersi oluşturulur. Çocuk 2'yi oluşturmak için, Çocuk 1 oluşturulurken yapılan işlemin aynısı bu defa çaprazlama maskesinin tersi kullanılarak yapılır (Bolat, Erol ve İmrak, 2004).

Aile 1	0	0	1	1	1	1
Aile 2	1	1	1	1	0	0
Maske	0	1	0	1	0	1
Maskenin Tersi	1	0	1	0	1	0
Çocuk 1	0	1	1	1	1	0
Çocuk2	1	0	1	1	0	1

Şekil 2.15. Uniform çaprazlama (Vural, 2005)

2.3.5. Mutasyon operatörü

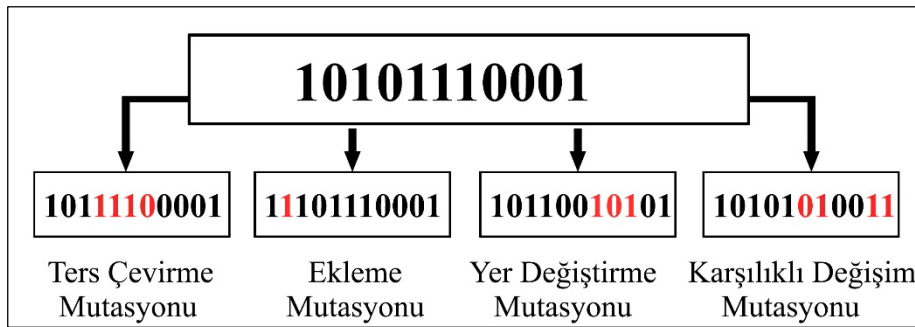
Çaprazlama ile mevcut genler korunarak popülasyon içinde yeni bireyler oluşturulur. Bu nedenle çeşitlilik şansı azdır. Yani mevcut genler gerekli olan tüm bilgiyi içermezse tatmin edici bir çözüme ulaşılamaz ve genler bir süre sonra birbirini tekrar etmeye başlayabilir (İşçi ve Korukoğlu, 2003). Bu nedenle popülasyonda yeni kromozomlar üretmek ve çeşitlilik sağlamak için mutasyon operatörü kullanılır. Algoritmanın tekrara düştüğü, tıkanıp durduğu noktalarda ya da çözüme en uygun birey oluşmadığında mutasyon operatörü ile kromozom dizilerinde bazı değişiklikler yapılabilir (Gökay Emel ve Taşkın, 2009:57). Ancak genler belli bir mutasyon oranına göre mutasyona uğratılırlar. Bu değer genellikle düşük bir değerdir. Çünkü yüksek bir mutasyon değeri belirlenirse elde edilen uyum değeri yüksek bireyler kaybedilebilir ve bunun sonucunda algoritma en uygun çözümden uzaklaşır (Erdoğan, 2007).



Şekil 2.16. Basit mutasyon operatörü (Bentley, 1999:10)

Çeşitli mutasyon yöntemleri vardır. Bunlar ters çevirme, ekleme, yer değiştirme ve karşılıklı değişim mutasyonlarıdır.

Ters Çevirme mutasyonunda kromozom dizisinin bir bölümü seçilerek kendi içinde ters çevrilir. Dizide geri kalan genler aynı şekilde korunur. Ekleme mutasyonunda rastgele bir gen seçilir ve kromozom dizisi içerisinde rastgele bir gen kaldırılarak yerine seçilen gen konulur. Yer değiştirme mutasyonunda kromozom dizisi içerisinde bir gen ya da gen grubu seçilerek dizi içerisindeki yeri değiştirilir (Gökay Emel ve Taşkın, 2009:58). Karşılıklı değişim mutasyonunda kromozom dizisi içerisinde seçilen iki adet gen ya da gen grubu yer değiştirilir (Vural, 2005). Şekil 2.17’de mutasyon yöntemlerine dair örnekler gösterilmiştir. Bu yöntemler kullanılırken mutasyona uğratılan genler kırmızı renkle belirtilmiştir.



Şekil 2.17. Mutasyon yöntemleri

2.3.6. Durdurma operatörü

Genetik algoritma modeli bir döngü halindedir. Bütün operatörler durdurulana kadar nesiller boyunca devam eder. Bir durdurma kriteri konulmazsa döngü sonsuza kadar devam edecektir (Gökay Emel ve Taşkın, 2009:63). Bir durdurma operatörü eklendiğinde her nesil sonrası durdurma kriterinin sağlanıp sağlanmadığı kontrol edilir ve durdurma kriterinin

sağlandığı noktada algoritma durur. Çeşitli durdurma kriterleri vardır. Önceden belirlenen kuşak sayısına ulaşılması, istenen uygunluk fonksiyonu değerinin sağlanması veya algoritmanın belirlenen süre kadar devam etmesi hâlinde döngü durdurulabilir (Gökay Emel ve Taşkın, 2002).

Hesaplama Zamanı Kriteri: Bu yöntemde önceden bir nesil sayısı ya da hesaplama zamanı belirlenir ve bu hedefe ulaşıldığında algoritma durur. Ancak bu yöntem her zaman çok sağlıklı bir seçim olmayabilir. Eğer algoritma gerekenden önce durdurulursa ileriki evrelerde oluşabilecek daha sağlıklı çözümler göz ardı edilmiş olur. Tam tersi algoritma gereğinden geç durdurulursa bu da hesaplama zamanını arttıracaktır (Gökay Emel ve Taşkın, 2009:63).

Optimizasyon Hedefi Kriteri: Bu yöntemde önceden ulaşılmak istenen uygunluk değeri belirlenir. Maksimizasyon problemlerinde uygunluk değeri bu değer üstünde bir çözüm üretildiğinde, minimizasyon problemlerinde ise uygunluk değeri bu değer altında olan bir çözüm üretildiğinde algoritma durur (Vural, 2005).

Minimum İyileşme Kriteri: Bu yöntemde nesiller boyunca elde edilen çözümlerin uygunluk değerleri arasındaki farka bakılır. GA mantığı gereği nesiller arasında çözümlerin iyileşme oranı önce hızlı bir artış gösterecek ve bu artış zamanla yavaşlayacaktır. Nesiller arasındaki iyileşme oranı giderek azalıp sifıra yaklaştıkça artık iyileşme beklenemeyeceği öngörülerek algoritma durdurulur (İşlier, 2001).

Durdurma kriteri belirlerken sadece çözümün kalitesi dikkate alınmaz. Bununla birlikte çözüme ulaşmak için harcanacak zaman da önemlidir. Bu nedenle algoritmanın hangi noktada durdurulacağına karar verirken bu iki kriter beraber değerlendirilmeli ev en kısa sürede en makul sonuca ulaşmak hedeflenmelidir (İşlier, 2001).

2.4. Genetik Algoritma Parametreleri

Algoritma işlemeye başlamadan önce belirlenmesi gereken bazı parametreler vardır. Bu parametreleri doğru bir şekilde belirlemek algoritmanın kaliteli olması, sağlıklı bir şekilde çalışması ve başarıya ulaşması için önemlidir. Bu parametrelere kontrol parametreleri de denir (Horat, 2014; Güngör, 2010). Belirlenen parametrelerin değişimi çözüm kümesini değiştirmekte, çözüm süresini ve birey seçimini etkilemektedir (Güleç, 2014).

Parametreler birbirlerini doğrusal olmayan biçimde etkilediği için aynı anda optimize edilemezler. Tüm algoritmalarda değerlendirilecek optimal parametreleri bulabilmek için birçok araştırma yapılmış ancak bu konuda bir uzlaşmaya varılamamıştır (Mitchell, 1996:131). Her problemin çözümü için GA modeli özelleştiğinden parametrelerde aynı şekilde probleme göre farklılaşmaktadır. Bu nedenle bu başlıkta GA için genel olarak uygulanabilecek bazı parametreler incelenmiştir. Bunlar popülasyon büyüklüğü, kuşak aralığı, çaprazlama olasılığı, mutasyon olasılığı, seçim stratejisi ve uyum ölçeklemesi olarak sayılabilir (Gökay Emel ve Taşkın, 2002).

2.4.1. Popülasyon büyüklüğü

Popülasyon büyüklüğü genetik algoritmanın tüm performans ve verimliliğini etkiler. Bu nedenle önemli bir faktördür. Genelde az sayıda birey içeren popülasyonlarda genetik algoritmalar etkin bir sonuç vermemektedir. Çünkü popülasyon yeterli sayıda örnek çözüm içermediğinden algoritma çözüm uzayının ancak bir kısmını tarayabilir ve bu yüzden çok fazla seçenek arama uzayı dışında kalır. Ancak çok sayıda birey içeren popülasyonlarda GA çözüm uzayının daha geniş bir alanını taradığından daha fazla seçenek ile daha çeşitli çözümler üretebilir. Böylece algoritmanın henüz optimum değere ulaşmamış bir çözümle erken sonlanmasının önüne geçilmiş olur. Ancak büyük popülasyon kullanmanın dezavantajı ise daha fazla kromozoma değerlendirme yapılacağından çözüme ulaşma süresinin artmasıdır (Gökay Emel ve Taşkın, 2009:72-73).

Bazı araştırmacılar en uygun popülasyon büyüklüğünü bulabilmek için araştırmalar yapmışlardır. 1989 yılında Schaffer ve arkadaşları fonksiyon optimizasyonu için yaptıkları çalışmalar sonucunda 20-30 aralığında bir popülasyon büyüklüğünün uygun olduğu çıkarımına varmışlardır. Genel olarak kısıt içeren problemlerde ise dizi uzunluğu ile popülasyon büyüklüğünün yakın değerler olması gerektiği ortaya konmuştur (Gökay Emel ve Taşkın, 2009:72). Odetayo ise 1993'te GAPOLE adı verilen GA-tabanlı program ile yaptığı araştırma sonucunda 300'ün optimum bir popülasyon büyüklüğü olduğunu belirlemiştir (Vural, 2005).

2.4.2. Kuşak farkı

Bir popülasyonda bireyler rastgele ya da uygunluk değerleri dikkate alınarak seçilebilirler. Bütün bireyler rastgele seçildiğinde optimum çözümden uzaklaşılabilir. Tam tersine tamamı uygunluk değerine göre seçilirse bölgesel en iyi çözümlerin uygunluk değerleri birbirine yaklaşabilir. Bu tarz sorunları önlemek için her kuşakta bazı bireyleri rastgele, bazı bireyleri de uygunluk değerlerine göre seçmek gereklidir (Güngör, 2010).

Algoritmada bir sonraki nesil için kaç adet yeni kromozomun seçildiğini gösteren parametreye ise kuşak farkı denir. Bu oranın %100 olması kromozomların tamamının yer değiştirdiği anlamına gelmektedir (Güngör, 2010).

2.4.3. Çaprazlama olasılığı

Çaprazlamada amaç mevcut iyi kromozomların özelliklerini birleştirerek daha iyi kromozomlar oluşturmaktır (Gökay Emel ve Taşkın, 2002). Çaprazlamanın hangi sıklıkla yapılacağını belirleyen değer ise çaprazlama olasılığıdır. Eğer hiç çaprazlama yapılmazsa bu değer %0 olur. Dolayısıyla yeni bireyler eski bireylerin aynısı olacaktır. Tam tersine bütün bireyler çaprazlama yolu ile elde edilirse bu oran %100 olacaktır (Kurt ve Semetay, 2001). Çaprazlama oranı ise genelde 0.6 ile 1 arasında bir sayıdır. Daha küçük oranlarda çeşitlilik azalırken daha büyük oranlarda en iyi çözüm gözden kaçırılabilir. Diğer yandan iyi kromozomların bozulma olasılığı da artmaktadır (Horat, 2014; Gökay Emel ve Taşkın, 2002).

2.4.4. Mutasyon olasılığı

Kromozomların ne sıklıkla mutasyona uğrayacağını gösteren parametredir. Amaç popülasyonun genetik çeşitliliğini korumaktır. Ayrıca kromozomların birbirine benzediği ve GA'nın sıkıştığı noktalarda çözüm olarak mutasyon kullanılır (Horat, 2014; Güleç, 2014). Eğer bu değer %0 olursa bireyler hiç değişmeden kalır. Eğer bu değer %100 olursa popülasyon içindeki bireyler tamamen değişir (Kurt ve Semetay, 2001).

Mutasyon, çaprazlamaya göre daha az uygulanan bir operatördür. Genellikle değeri 0.5' den küçük bir sayıdır (Akpınar, 2009). Eğer bu değer artarsa algoritmanın kararlılığı bozulabilir

ve rastlantısal bir aramaya dönüşebilir. Ancak bu durumda kayıp genetik parçaları tekrar bulmakta olasıdır (Gökay Emel ve Taşkın, 2009:79).

2.4.5. Seçim stratejisi

Nesiller çeşitli yöntemlerle çoğaltılırlar. Bu yöntemlerden biri olan kuşaksal stratejide, popülasyondaki bütün kromozomlar tamamen yavrularla yer değiştirilir ancak bu durumda uygunluk değeri en yüksek olan birey de yok olacağından bu stratejiyi elitizm ile birlikte kullanmak faydalıdır. Elitizm yönteminde popülasyondaki uygunluk fonksiyonu en yüksek olan bireyler olduğu gibi bir sonraki nesile aktarılırlar. Bu şekilde en uygun bireyler korunmuş olur.

Bir diğer yöntem olan denge durumu stratejisinde ise her kuşakta sadece birkaç kromozom yenilenir. Genellikle en kötü kromozomlar çıkarılarak, yeni kromozomlar onların yerine yerleştirilirler (Gökay Emel ve Taşkın, 2009:80; Gökay Emel ve Taşkın, 2002). Bu stratejilerin en önemli noktası bireyleri mümkün olduğunca rastgele seçip çaprazlamaktır. Sonuca ne kadar yavaş ilerlenirse, o kadar iyi sonuçlar alınır (Güleç, 2014).

2.4.6. Uyum ölçeklemesi

Fonksiyon ölçeklemesi olarak da isimlendirilir. Birçok ölçekleme yöntemi mevcuttur. Bunlardan en çok kullanılanları doğrusal ölçekleme, üstsel ölçekleme ve sigma kısaltmasıdır (Yeniay, 2001). Genetik Algoritmanın etkin bir şekilde işleyebilmesi için probleme en uygun ölçekleme yönteminin seçilmesi oldukça önemlidir (Gökay Emel ve Taşkın, 2009:80).

2.5. Genetik Algoritmanın Avantaj ve Dezavantajları

Genetik algoritmanın diğer optimizasyon yöntemlerinden bazı farkları vardır. Temel farklar şunlardır;

1. GA'lar parametre değerleri ile değil bunların kodlanmış versiyonları ile çalışır. Çözüm üretebilmek için parametrelerin kodlanması gerekir (Goldberg, 1989:7).

2. Diğer yöntemlerden farklı olduğu en önemli konu paralel olmasıdır. Klasik optimizasyon yöntemleri seri özelliktedir. Yani tek noktadan başlar, sırayla çözümleri inceler ve adım adım en iyi çözümü bulmaya çalışır. Ancak sonuç ürün optimum değilse baştan başlanması gerekmektedir. Ancak GA'lar tek bir çözümle değil çözümler kümesi ile çalışır ve tek seferde çok fazla yönde araştırma yapabilir. Eğer bir yol sonuca ulaşmazsa elenir ve algoritma başka bir yoldan aramaya devam eder. Bu şekilde algoritmanın yerel en iyi çözümlere takılma ihtimali azalır ve global optimuma ulaşması kolaylaşır (Vural, 2005). Ancak bazı durumlarda yine de yerel en iyi çözüme daha hızlı ulaşılsa algoritma bu noktada sonuçlanabilir. Bunu engellemek için çeşitlilik artırılabilir ya da uygunluk fonksiyonu her adımda değiştirilebilir. Ancak yine de GA'da çözüm kümesi iyi çözümlerle birlikte kötü çözümleri de içerdiğinden en iyi çözüme ulaşma garantisi yoktur (Akpınar, 2009).

3. Değerlendirme işleminde türev değerini değil uygunluk fonksiyonunu kullanır. Böylece sürekli ya da kesikli fonksiyonlar kullanılması algoritmanın yapısını etkilemez. Ayrıca yardımcı bir bilgi kullanılmasına da ihtiyaç kalmaz (Söke, 2003).

4. Geleneksel yöntemler sürekli optimizasyonlara deterministik dönüşüm uygularken, genetik algoritmalar olasılık kurallarını uygular (Çakar, 2009).

Bunlarla birlikte genetik algoritmanın diğer optimizasyon yöntemlerinin başarısız kaldığı bazı durumlarda çok çapıcı sonuçlar üretebilmesini sağlayan bazı avantajları vardır.

- Paralel çalışma özelliği sayesinde çok geniş bir arama uzayını tarayabilirler. Bu nedenle makul bir sürede araştırılamayacak geniş bir alternatif çözüm kümesine sahip, doğrusal olmayan problemler için oldukça uygundur. Gerçek hayatta problemler genellikle doğrusal değildir (Vural, 2005).
- Yine paralel çalışma özelliği sayesinde etkin bir arama yaparak en iyi çözümlere en hızlı şekilde ulaşır (Karasoy ve Ballı, 2016).
- Sürekli olmayan, zamanla değişen ve birçok yerel optimum noktası olan problemler için çok uygundur. Çünkü çok yönlü çalıştığından yerel optimumlara takılma ihtimali daha azdır (Ağalday, 2018).

- Gerçek hayatta problemler tek bir parametreye bağlı değildir. Bazen bazılarında ödün vererek bir diğer parametrenin optimize edilmesi gerekir. Bu şekilde birden fazla parametreye bağlı problemlerde GA sayesinde bir çözümde bir parametre optimize edilirken başka bir çözümde başka bir parametre optimize edilebilir (Vural, 2005). Bu şekilde sadece mutlak en iyiyi değil, parametrelere bağlı en iyilerin listesini çıkarabilir. Bundan sonra kendi amacına en uygun çözümü seçmek kullanıcı inisiyatifine kalır (Şeker T. , 2008).
- Farklı yöntemlerle birleştirilerek hibrid olarak kullanılabilirler (Çakar, 2009; Kaya, 2012).
- Genetik algoritmalar parametrelerle değil bunların kodlanmış versiyonları ile ilgilenir. Çözümleri rastgele değiştirip uygunluk fonksiyonlarına göre bu değişimi değerlendirir (Goldberg, 1989:7). Olasılıklara göre çalıştığından ne kadar iyi sonuç alınacağı önceden kestirilemez (Karasoy ve Ballı, 2016).
- Paralel bilgisayarlarda çalışmaya uygundur (Çakar, 2009; Kaya, 2012).
- Kendi kuralları ile çalıştığından problemin kurallarını bilmesi gerekmez. Genetik sayı sistemine göre üretilen sayılarla çalışır. Bunlar deney verileri veya analitik fonksiyonlar olabilir (Çakar, 2009; Şeker T. , 2008).

Ancak GA güçlü ve etkili bir problem çözme yöntemi olmasına rağmen her probleme uygulanamazlar. Bütün bu avantajların yanında bazı kısıtlamaları ve dezavantajları da mevcuttur. Bunlar bazıları aşağıda açıklanmıştır.

- En önemli konulardan biri kodlama dilidir. Kodlama dili olası değişiklikleri tolere edebilecek bir dil olmalıdır. Aksi takdirde hatalar ve anlamsızlıklar ortaya çıkacaktır. Bu nedenle kodlama dili güçlü, etkili ve karmaşıktır (Vural, 2005).
- Uygunluk fonksiyonunun doğru tanımlanması çok önemlidir. Aksi takdirde algoritma gerçek problemi görmezden gelip başka bir sorunun çözümünü araştırabilir.

- Diğer GA parametrelerinin doğru belirlenmesi de çok önemlidir. Örneğin popülasyon büyüklüğü yeterli olmazsa doğru çözüm için yeterli arama yapılamayabilir. Ya da çaprazlama, mutasyon gibi diğer operatörlerin oranları uygun değilse veya seçim stratejisi doğru değilse optimum çözümden uzaklaşılır (Akpınar, 2009).
- Başka bir problem ise prematüre yakınsamadır. Prematüre yakınsama bireyler arasındaki farklılığı çok erken ve hızlı bir şekilde yok ederek genetik yığılma problemine yol açar. Yani eğer popülasyondaki bir bireyin uygunluk değeri diğerlerinden çok fazla yüksekse devamlı o birey seçileceğinden kuşaklar değiştikçe popülasyonun çeşitliliği azalır ve yerel bir optimuma doğru ilerler. Popülasyon büyüklüğünün az olmaması, uygun seçim stratejisi kullanmak vs. yöntemlerle bu olayın önüne geçilebilir (Vural, 2005).
- Kesin sonuç gerektiren problemlere uygun değildir. Çünkü sınırlı bir sürede mükemmel çözüme ulaşma garantisi yoktur (Kaya, 2012).
- Diğer optimizasyon yöntemleri belirgin yöntemlerdir. Ancak Genetik Algoritmalarda olasılık kuralları kullanıldığından aynı genetik algoritma modeli aynı şartlar altında tekrar çalıştırıldığında aynı sonucu vermez (Şeker T. , 2008).
- Teorik alt yapıları basit olup problem uzayının özelliklerinden istenilen kadar yararlanamazlar (Ağalday, 2018).
- Rastgeleliği sebebiyle özellikle diğer optimizasyon yöntemlerinin kolaylıkla çözebileceği daha basit yapıları problemler için çözümü bulması diğer yöntemlerden uzun süreceği için, kullanılması mantıklı değildir (Ağalday, 2018).
- Bulunan çözümün en iyi olup olmadığı kontrol edilemez. Bu nedenle en iyi çözümün ne olacağının hiçbir şekilde bilinemediği ya da tahmin edilemediği problemlerde kullanmak daha mantıklı olacaktır (Şeker T. , 2008).
- Yerel optimumları atlayabilir. Bu nedenle yerel optimumlara ulaşmak istendiğinde kullanılacak en iyi yöntem değildir (Kaya, 2012).

2.6. Genetik Algoritmanın Uygulama Alanları

Genetik algoritmalar yapısı itibariyle karmaşık ve büyük bir çözüm uzayını taramayı gerektiren problemlerde, kısa sürelerde başarılı sonuçlar elde etmektedir. Özellikle 1985 yılından sonra birçok farklı alanda birçok farklı şekilde kullanılmıştır (Özçakar, 1998). Bunlardan bazıları; (Cunkaş, 2006)

- ✓ Optimizasyon Problemleri
- ✓ Gezgin Satıcı Problemi
- ✓ Araç Rotalama Problemi
- ✓ Çizelgeleme Problemi
- ✓ Tesis Yerleşim Problemi
- ✓ Hücresel Üretim Problemi
- ✓ Atama Problemi
- ✓ Makine öğrenmesi
- ✓ Montaj Hattı Dengeleme Problemi
- ✓ Baskı Devre Kartlarında İşlem Sırası Belirleme Problemi
- ✓ Sistem Güvenilirliği Problemi
- ✓ Görüntü işleme
- ✓ Sosyal sistemler

2.6.1. Optimizasyon problemleri

Optimizasyon problemleri genellikle tek bir değerin optimizasyonu şeklinde çalışılır ve bu problemler diğer klasik yöntemlerle çözülebilirler. Ancak birden fazla değerin bulunduğu problemleri tek kriterli hale getirmek zordur (Goldberg, 1989:7). Böyle çok amaçlı problemlerde GA başarıyla kullanılır (Engin, 2001).

“Genetik Algoritma İle Ankara İlinde Mobese (Kent Güvenlik Yönetim Sistemi) Yerleştirme Optimizasyonu” tezi GA ile optimizasyon problemlerinin çözümüne dair bir örnektir (Horat, 2014).

2.6.2. Gezgin satıcı problemleri

Bu problemlerde amaç kat edilen yolu minimize edecek bir rota çizmektir. En çok kullanılan yöntemlerden biridir. Birçok problem bu probleme benzer bir mantıkla çözülür. Bu problemlerde gidilecek şehir sayısı ile algoritmanın çözüm bulma zamanı arasında üstel bir ilişki vardır. Yani gidilecek şehir sayısı n ise incelenecek rota sayısı $(n-1)!$ kadardır. Bu mantıkla gidilecek şehir sayısını 1 arttırdığımızda $(n+1)$ incelemek rota sayısı n kat artacaktır $(n!)$. Bir algoritmanın yaygın performans ölçütünün hesaplama zamanı olduğu dikkate alındığında bu problemlerin genetik algoritma ile çözülmesinin önemi ortaya çıkmaktadır (Gökay Emel ve Taşkın, 2009:98).

Bu yöntemle “Google Maps ve Genetik Algoritmalarla GSP Çözümü İçin Öneri” (Karasoy ve Ballı, 2016) tezi örnek verilebilir.

2.6.3. Araç rotalama problemleri

Kısaca bir depodan ayrılan ve farklı müşterilere malzeme götüren araçların rotalanması problemleridir. Her aracın kapasitesi ve gideceği yol bellidir. Bu şartlar altında araçların en düşük maliyetle müşterilere ulaşması amaçlanır. Daha karmaşık bir problem olan pencereli araç rotalamada zaman faktörü de devreye girer (Horat, 2014).

“Zaman Pencereli Araç Rotalama Problemi’nin Genetik Algoritma İle Modellenmesi” (Dursun, 2009) tezi ve “Araç Rotalama Problemleri Ve Zaman Pencereli Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı” (Şeker Ş. , 2007) tezi bu probleme bir örnek olarak verilebilir.

2.6.4. Çizelgeleme problemleri

Farklı işlem süreleri ve belli teslim tarihleri olan işlerin çizelgelenmesi üzerine bir yöntemdir. İlk defa 1985 yılında Davis tarafından uygulanmıştır (Gökay Emel ve Taşkın, 2009:96). 1987’de Liepins ve arkadaşları, belirli teslim tarihleri ve işlem süreleri olan işlerin çizelgelenmesi problemini araştırmışlardır. Genetik Algoritmalar çizelgeleme problemlerinde optimuma yakın çözüme, diğer çözüm yöntemlerinden çok daha kısa bir sürede ulaşmışlardır.

“Hemşire Çizelgeleme Problemlerinin Genetik Algoritma İle Çözümü” (Çivril, 2009) ve “Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu” (Engin, 2001) tezleri bu yönteme örnektir.

2.6.5. Tesis yerleşimi problemleri

Araç gereçleri veya başka gerekli kaynakları belli kıstaslara göre optimum performans sağlayacak şekilde yerleştirme problemleridir. Bu noktada her parçanın yerleşiminde özel bir takım ihtiyaçlar olabilmektedir. Bununla birlikte maliyette dikkate alınmalıdır. Hızlı ve doğru kararlar vermek önemlidir. Yerleşimle ilgili yanlış verilecek bir karar üretim aşamasında ek maliyete ve zaman kaybına sebep olabilir. Görüldüğü üzere bu problem birçok değişkenin birbirini etkilediği ve aynı anda hepsine karar verilmesi gereken bir problemdir. Bu nedenle genetik algoritma kullanımına oldukça uygundur (Horat, 2014).

“Mekanik Otopark Tesislerinin Yer Seçimi Ve Boyutlandırılması İçin Genetik Algoritma Tabanlı Bir Yaklaşım” (Karasay, 2016) tezi bu kullanıma örnektir.

2.7. Mimari Tasarım Uygulamalarında Genetik Algoritmaların Kullanımı

Mimarlık ve genetik ilişkisinin temeli 1960’lı yıllara dayanmaktadır. İlk örneklerinden biri 1961 yılında Kisho Kurukowa’nın strüktürünü DNA sarmalı şeklinde tasarladığı Helix Kent projesidir. Öncelikle şehirleri büyüyen gelişen ve kendi kendine yenilenen sistemler olarak gören metabolist akım ortaya çıkmıştır. Daha sonra bu fikri Genetik Mimarlık akımı takip etmiştir. (www.academia.edu) Bu akım adaptasyon yeteneği olan, kendi kendine üreyebilen, gelişen, evrimleşen ve hatta ölen mimari öğeler üretmeyi amaçlamıştır. (gulden.kokturk.com) Mekânların bir organizma mantığında tasarlanması, dolayısıyla yapıların canlılık özelliği taşıması üzerine düşünceler gelişmiştir (www.academia.edu).



Resim 2.1. Kisho Kurukowa - Helix Kent Projesi (Tokyo,1961)

Bütün bu düşünceler dikkate alınarak mimarın sonuç ürünü değil bir mühendisin DNA üzerine çalışmalarında olduğu gibi bu ürüne ulaşacak süreci düşünmesi ve bu ürünü oluşturabilecek yazılımları tasarlaması beklenmektedir (Erbaş Korur ve Dülgeroğlu Yüksel, 2013). Bahsedilen bu yazılımlar Darwin'in evrim teorisini temel alan, ilk defa 1970 yılında John Holland tarafından hakkında çalışmalar yapılan genetik algoritma yöntemini kullanan yazılımlardır (gulden.kokturk.com).

Genetik algoritmalar canlıların oluşum sürecini taklit eder. Bu yöntemde genlerin işlevlerini algoritmalar ve kodlar tanımlar. Bu şekilde farklı operatörler yardımıyla çoğalma gerçekleşir. Böylelikle aynı aileden olan ancak birbirinden farklı olan biçimler elde edilir. İstenene en uygun form elde edilene kadar bu genetik süreç devam ettirilir (Frazer, 1995:58).

John Frazer'e göre (1995) evrimsel mimarinin amacı yapılı çevrede, doğal çevrenin karakteristik özelliği olan simbiyotik davranışı ve metabolik dengeyi sağlamaktır. Evrimsel mimarlık tanımını ise doğadaki adaptasyonu bazı yardımcı araçlar kullanarak bilgisayar ortamına aktarma ve bu şekilde yeni formlar üretmek olarak açıklamaktadır (Frazer, 1995:9).

Kolarevic'e göre ise (2003) evrimsel tasarımın arkasındaki anahtar kavram genetik algoritmalarıdır. GA'lar çaprazlama, mutasyon vb. genetik süreçleri takip eder ve kromozom yapısına benzer bir dizi yapısına sahiptir. Çeşitli parametreler bu yapıya kodlanır ve bu parametrelere bağlı uygunluk değerleri hesaplanır. Buna göre yapılan seçimlerle daha sağlıklı kuşaklar elde edilir. Asıl önemli olan ortaya çıkacak biçimden ziyade sistem mantığının ve sürecin doğru kurgulanmasıdır (Kolareviç, 2003:37).

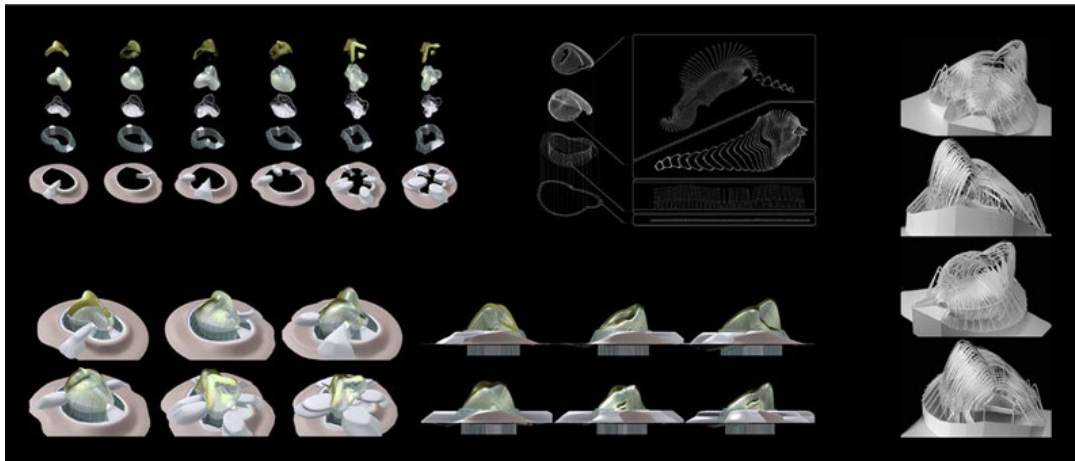
2.7.1. Embriyonik ev

Bu alanda yapılan çalışmalardan biri mimar Greg Lynn'e ait olan embriyonik evler projesidir. Gerekli yazılımı 1998 yılında Lynn ve öğrencileri Kolombiya'da geliştirmiştir. Bu projede öncelikle mimarlar kullanıcılara anket düzenleyerek bir evde ihtiyaç duydukları alanlarla ilgili birtakım sorular sormuşlar ve verilen cevapları analiz ederek bir form oluşturmuşlardır. Yani bu çalışmada mimar formu üreten değil, formun üretildiği süreci denetleyen bir görev almıştır (Livesey, Parlac, Taron ve Langjing, 2011).

Bütün bu çalışmanın sonucunda altı adet temel form üretilmiştir. Bunlar genetik süreçlere tabii tutularak binlerce farklı form üretilebilecektir. Bu formların hepsi aynı alüminyum omurgaya aynı kiriş ve panellere sahip olmasına rağmen karakter ve özellikleri farklıdır. Bu şekilde isteyen kullanıcı program sayesinde kendi evinde değişiklikler yaparak, istediği şekle getirebilecektir (Erbaş Korur, 2012).

Greg Lynn bu çalışmada;

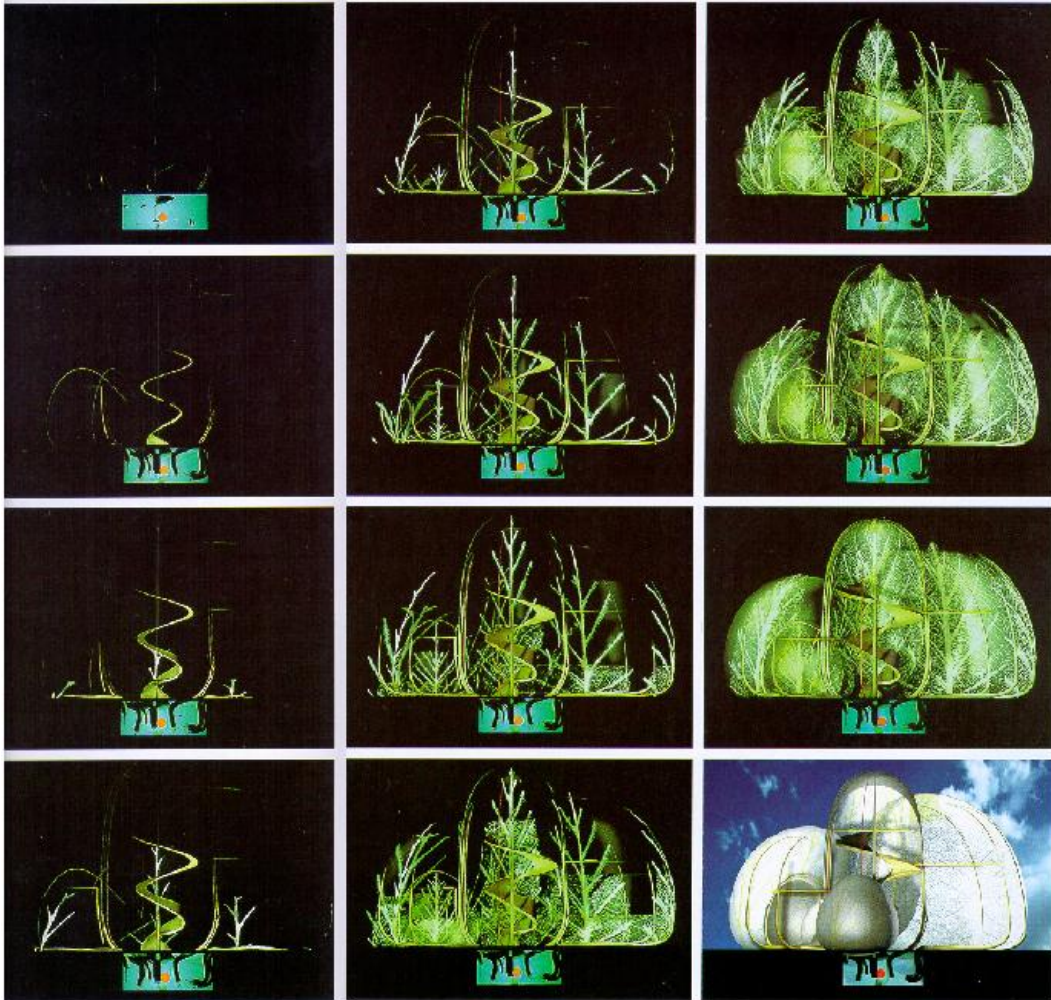
- Modernist yaklaşımın ötesinde sonsuz sayıda tekrarlamalar ile üretilebilecek organik esnek, genetik bir ev prototipi üzerine düşünmeyi
- Genel bir marka ürün kavramının yerine, kullanıcının isteklerine göre şekillenebilecek tasarımlar oluşturmayı
- Standart olmayan mimari formların üretimi için mevcut otomatik üretim teknolojilerinin yeteneklerini geliştirmeyi amaçlamıştır. (www.docam.ca)



Resim 2.2. Greg Lynn - Embriyonik Ev

2.7.2. Moleküler kurgulu ev

Mohamad Alkhayer ve John M.Johanser tarafından 2000 yılında tasarlanmıştır ancak tasarım 2200 yılı içindir. Bu çalışmaya göre bir mimar tarafından tasarım hazırlanır ve bu tasarım moleküler olarak kodlanır. Daha sonra bu kod seçilen bölgeye saksıya tohum ekme mantığı ile yerleştirilir. Tasarım yazılan genetik algoritma mantığında bir bitkinin büyümesini taklit ederek oluşmaya başlar. Öncelikle yapının temeli görevini görecektir kökler gelişir. Daha sonra sırasıyla üst yapıyı oluşturacak elemanlar, iç dış kaburgalar, kafes sistemi, merdivenler, dış ve iç duvarlar, katlar, açıklıklar ve en son mekanik sistem oluşur. Bu süreç dokuz günde tamamlanacak şekilde tasarlanmıştır. Oluşan yapı değişen koşulları ve kullanıcı ihtiyacını dikkate alarak kendini yenileyebilmektedir. Hatta ileriki yıllarda binanın kendini yıkması ve bina büyüme operasyonlarının gelecekteki binalar için geri dönüştürülmesi hedeflenmiştir (Johansen, 2002:132,133).

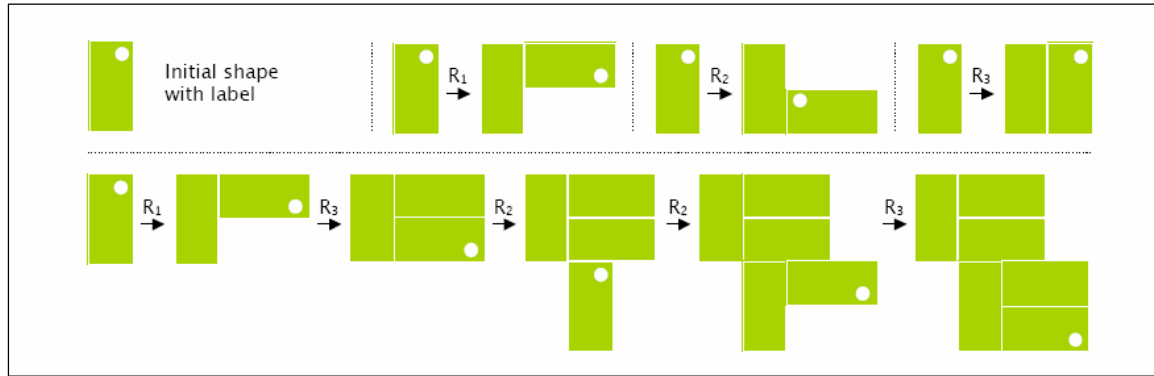


Resim 2.3. Mohamad Alkhayer ve John M.Johanser - Moleküler Kurgulu Ev (www.ona.vg)

2.7.3. Biçimin evrimi projesi

Biçim grameri ve genetik algoritmaların beraber kullanıldığı, Orestes Chouchoulas tarafından 2003 yılında hazırlanan “Biçimin Evrimi” olarak isimlendirilebilecek çalışmadır. Genetik algoritmalar etkili arama algoritmalarıdır ve gerekli işlevsel şartları sağlamaya yardımcı olur. Biçim grameri ise estetik ve yapısal şartları sağlamamıza yarar. Bu iki yöntem birleştirildiğinde hem işlevsel gereksinimleri karşılayabilen, hem de görsel açıdan tasarımcıyı tatmin edebilecek tasarımlar geliştirmek fikri bu tasarımın temelini oluşturmuştur. Bu yöntemde öncelikle tasarımcı tarafından bir biçim grameri oluşturulur. Daha sonra biçim grameri yardımıyla oluşturulan biçimler genetik algoritma ile işlevsel performanslarına göre optimize edilirler (Chouchoulas, 2003).

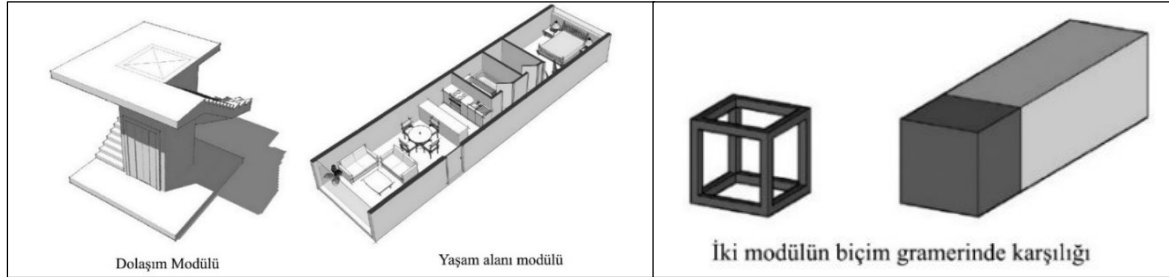
Buradaki birleştirici nokta biçim grameri ile üretilen biçimin kodunun genetik algorithmada genotip olarak kabul edilmesidir. Bu sayede biçim kodu tüm evrim süreçlerinde GA’nın kodlama dilinde geçerli hale gelir (Chouchoulas, 2003). Biçim kodu hangi ilk biçimle başladığını ve hangi kuralların hangi sırayla uygulanarak sonuç biçime ulaşıldığını gösteren zincirdir (Chouchoulas ve Day, 2007).



Şekil 2.18. Biçim grameri ile tasarım örneği (Badem, 2007)

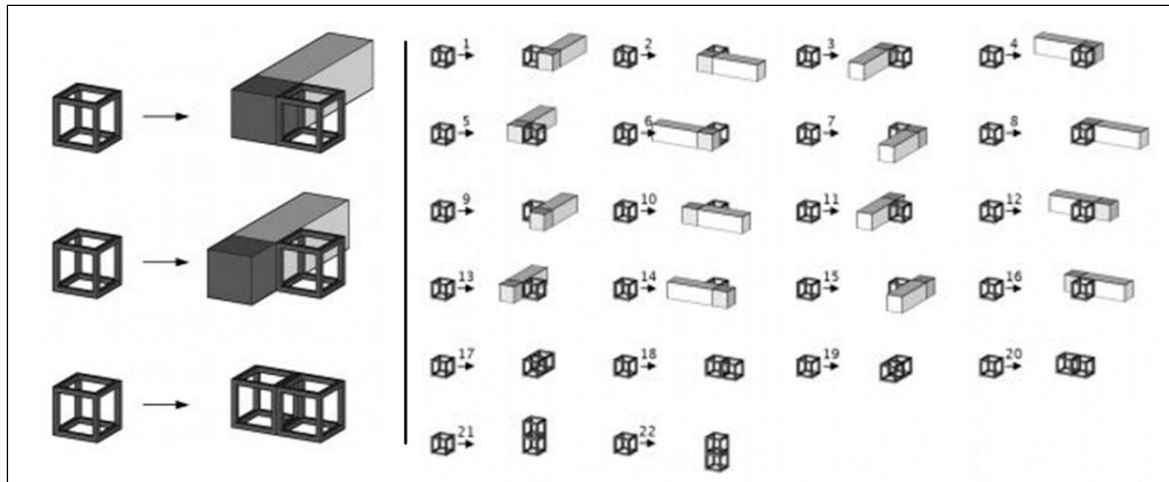
Chouchoulas bu yöntemi test edebilmek için bir apartman bloğu örneğini ele almıştır. Basit olması açısından yaşam alanı ve dolaşım birimi olmak üzere iki adet biçim belirlemiştir. Bir gereklilik olarak sirkülasyon alanlarının bitişik olması ve zemin katta bir giriş olması kuralları dikkate alınmıştır. Dolaşım modülü yatay ve düşey sirkülasyonu aynı anda sağlayabilecek 4x4x4 m boyutlarında bir küp, yaşam alanı ise 16x4x4 m boyutlarında bir apartman bloğu olarak tasarlanmıştır. Şekil 2.19’de yaşam alanı modülünün muhtemel bir

yerleşimi gösterilmiştir. Buna göre giriş kapısı duvar boyunca belli bir alanda yer değiştirebilecek şekilde planlanmıştır. Yaşam alanı bloğunun gri kısmı oturma odasını göstermektedir (Chouchoulas ve Day, 2007).



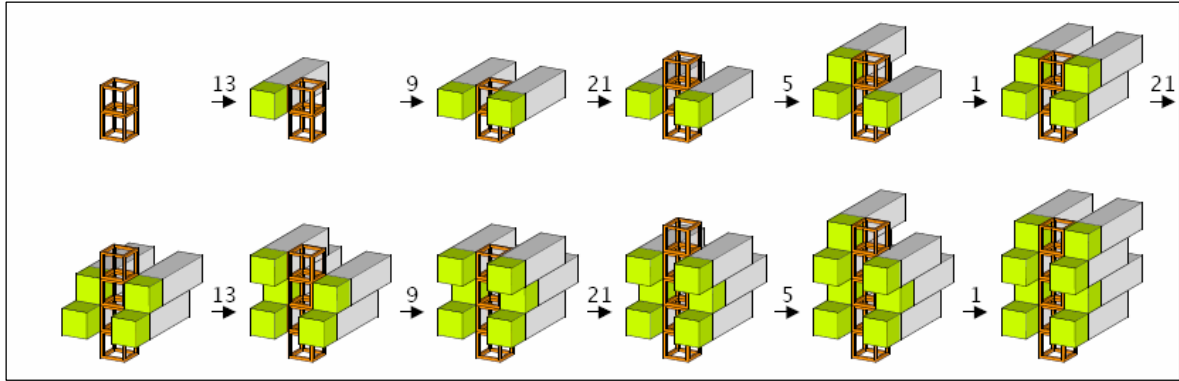
Şekil 2.19. Modüllerin şekilleri ve ifade ediliş biçimleri (Chouchoulas ve Day, 2007)

Bu adımlardan sonra modüllerin birleşimi için 3 temel kural belirlenmiştir. İlk iki kural dolaşım ve konut modüllerinin nasıl birleşeceğini gösteren kurallardır. 3. kural ise dolaşım modüllerinin birbiriyle birleşmesini göstermektedir. Daha sonra bu üç temel kuralın yatay ve düşey yöndeki varyasyonlarından 22 adet kural türetilmiştir. (Şekil 2.20) Bunun dışında ek kural olarak sadece zemin katta yaşam alanı modülü olmaması ile ilgili bir kural eklenmiştir (Chouchoulas ve Day, 2007).



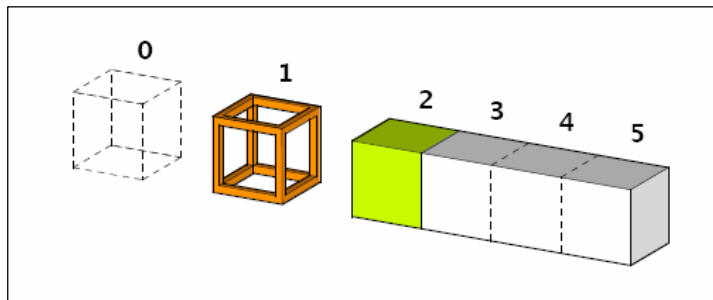
Şekil 2.20. Temel birleşim kuralları ve varyasyonları (Chouchoulas ve Day, 2007)

Bütün bunlardan sonra belirlenen birleşim kurallarına göre basit apartman bloğu oluşturulmaya başlanmıştır (Chouchoulas ve Day, 2007). (Şekil 2.21)

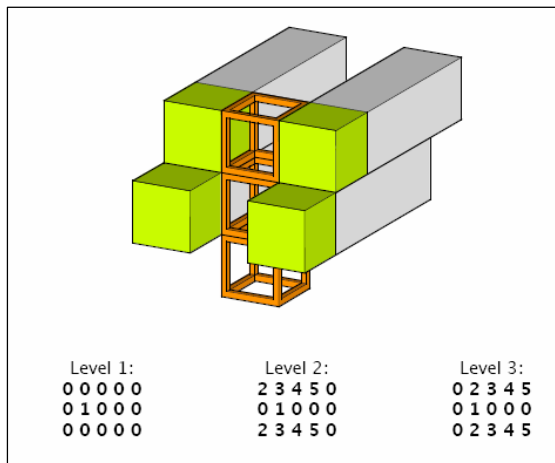


Şekil 2.21. Biçim grameri ile oluşturulmuş apartman tasarımı örnekleri (Badem, 2007)

Buraya kadar olan kısımda biçim grameri kullanılmıştır. Bundan sonraki adım ise Genetik Algoritmaların devreye girdiği kısımdır. Bu adımda öncelikle biçim kodlarla ifade edilmelidir. Bunun için boşluğa 0, dolaşım modülüne 1 ve yaşam modülünün her bir birimine 2,3,4,5 değerleri verilerek biçimler bir kod matrisi haline dönüştürülürler (Chouchoulas ve Day, 2007). (Şekil 2.22) (Şekil 2.23)



Şekil 2.22. Modüllerin numaralandırılması (Badem, 2007)



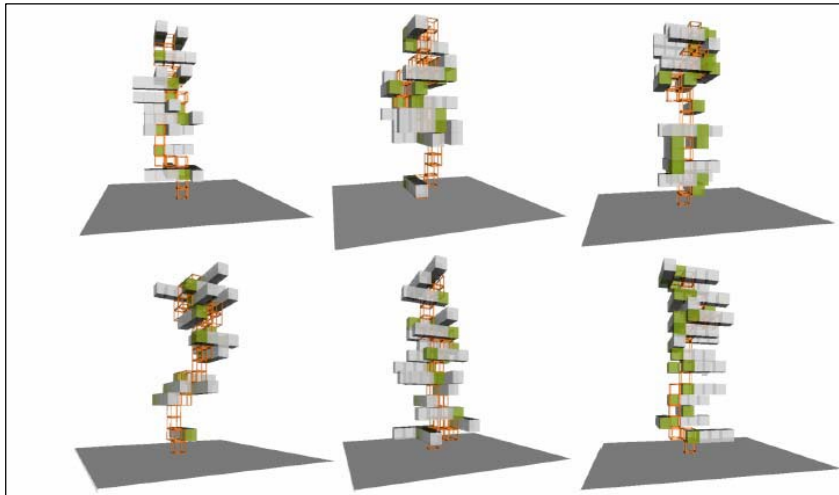
Şekil 2.23. Apartman bloğu değerlerinin dizi olarak gösterimi (Badem, 2007)

Bundan sonra klasik genetik algoritma çalışmaya başlar. Seçim yapabilmek için oluşturulan bu blokların bir uygunluk değeri olması gerekmektedir. Burada uygunluk değeri oluşturulan bloğun kullanıcı isteklerini ne kadar karşılayabildiği ile ilgilidir. Öncelikle iki adet arayüz oluşturulur. Birincisi kullanıcının biçimi görüntüleyebileceği, döndürebileceği ve ölçekleyebileceği bir arayüzdür İkincisi ise uygunluk değeri için gerekli kriterlerin seçileceği arayüzdür. Bu arayüzde apartman bloğunun karakteristik özellikleri sıralanır ve kullanıcının yükseklik, görünüş, balkon vb. diğer ölçütlere bir puan vermesi istenir (Badem, 2007).

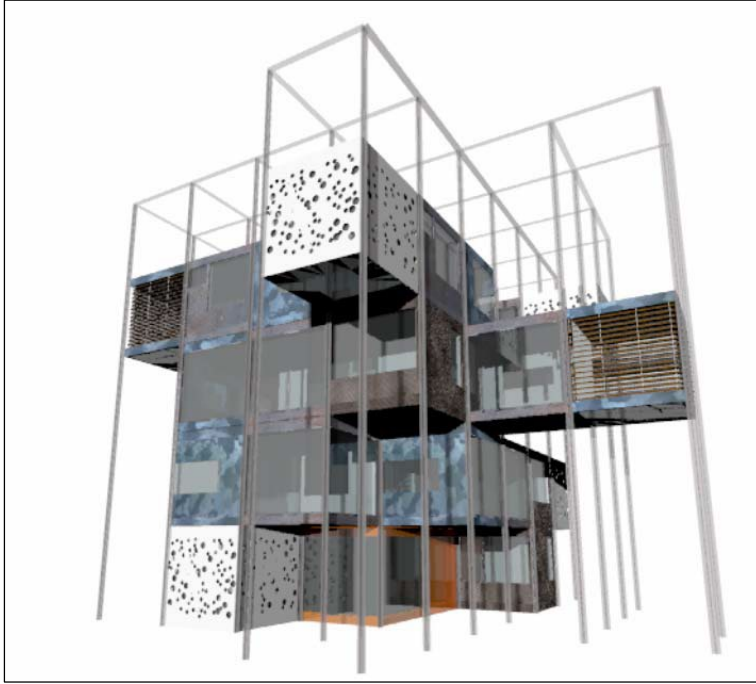


Şekil 2.24. Kullanıcı arayüzleri (Badem, 2007)

Son adım uygunluk değerinin karşılaştırılması ve sonuç ürünün seçilmesi adıımıdır. Popülasyondaki bireyler rastgele ikiye ikiye seçilir ve kıyaslanır. Uygunluk değeri düşük olan elenir. Bu yöntem turnuva seçim yöntemidir. Bu kıyaslama işlemi popülasyondaki tüm bireyler bitene kadar uygulanır. En çok puan alan ürün sonuç ürün olacaktır (Badem, 2007).



Şekil 2.25. En yüksek puanı alan bazı apartman modelleri (Badem, 2007)



Şekil 2.26. Biçim evrimi yöntemi sonucu elde edilen ev modeli (Badem, 2007)

2.7.3. Form-G modeli

Bu model Yazgı Badem tarafından 2007 yılında yazdığı yüksek lisans tezinde hazırlanmıştır. Tasarımın ilk evrelerinde birbirini etkileyecek ve değiştirecek birçok karar verilmesi gerekmektedir. Dahası bu kararların sonuçlarını bu evrede görmek mümkün olamayacağı için verilecek en doğru kararları tespit etmek çok zordur. Tasarımcının bütün olası alternatifleri denemesi de mümkün değildir. Bu nedenle erken tasarım evresinde tasarımcının karar vermesine ve form üretmesine yardımcı olacak, bilgisayar teknolojilerini kullanan ve genetik algoritma prensibine göre işleyen, bu sayede birçok alternatifi istenen kriterlere göre hızlı bir şekilde analiz edecek bir bilgisayar modeli oluşturmak amaçlanmıştır (Badem, 2007).

Form G modelinde amaç bütün işi bir bilgisayara yaptırmak değildir. Tasarımcının rolü de büyüktür. Bu modeli oluşturacak, gerekli parametreleri ve kısıtlamaları belirleyecek, uygunluk değerinin kıstaslarını seçecek ve sürecin kontrolünü sağlayacak kişi tasarımcıdır. Öncelikle modelin arayüzü tasarlanmıştır. Buna göre tasarımcı ilk olarak arazi sınırlarını belirlemek için arazinin kaç köşe noktasından oluşacağını ve bu noktaların koordinatlarını sisteme girer. Bununla birlikte tasarımın oluşacağı üç boyutlu sınırın belirlenebilmesi için bir yükseklik değeri de girilmelidir (Badem, 2007). (Şekil 2.27)

The first screenshot, titled "Boundary Determination", shows a window with a label "Number of corners of the site" and a dropdown menu set to "4". A "NEXT" button is located to the right. The second screenshot, titled "Boundary Coordinates", shows a window with two columns of input fields. The left column is labeled "Site coordinates" and contains fields for x1 (0), x2 (100), x3 (100), and x4 (0). The right column is labeled "y" and contains fields for y1 (0), y2 (0), y3 (100), and y4 (100). To the right of these fields is a "Height Constraint" section with a "Max Height" field set to 60. "Back" and "NEXT" buttons are at the bottom.

Şekil 2.27. Tasarım sınırlarıyla ilgili bilgilerin girileceği arayüz (Badem, 2007)

Bundan sonra tasarımcının karşısına kullanılacak üç boyutlu objelerin belirleneceği bir ekran çıkar. Daha sonra bu objelerin minimum ve maksimum hacimlerinin belirleneceği başka bir ekran çıkacaktır. Badem bu model için üç adet dörtgen prizma bir adet silindir seçmiştir.

The first screenshot, titled "Object Selection", shows a window with two lists. The left list contains "Box" and "Cylinder". The right list contains "Box1", "Box2", "Box3", and "Cylinder1". There are "Add" and "Remove" buttons between the lists, and a "GO" button at the bottom. The second screenshot, titled "Object Volumes", shows a window with four rows of input fields. The first row is for "Box1" with "Min.Volume" 2 and "Max.Volume" 10. The second row is for "Box2" with "Min.Volume" 60 and "Max.Volumue" 25. The third row is for "Box3" with "Min.Volume" 12 and "Max.Volume" 45. The fourth row is for "Cylinder1" with "Min.Volume" 8 and "Max.Volume" 25. "Back" and "GENERATE" buttons are at the bottom.

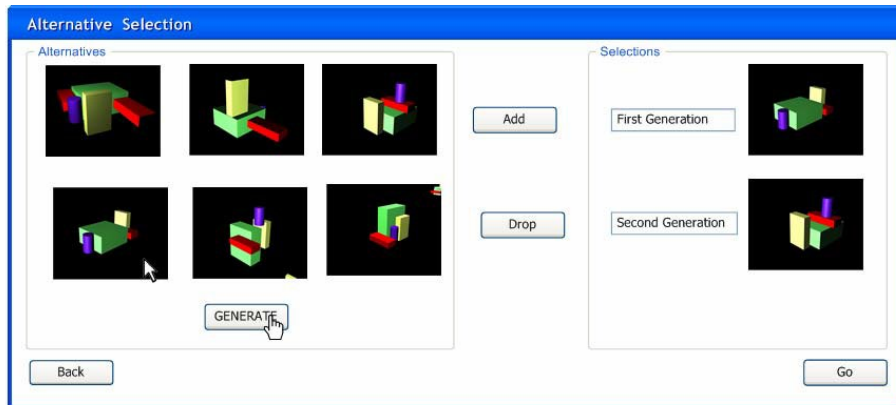
Şekil 2.28. Kullanılacak objelerin seçildiği ve min.-max. hacimlerinin belirlendiği arayüzler (Badem, 2007)

Daha sonra uygunluk değerleri belirlenmiş ve buna göre sistemin seçim yapması sağlanmıştır. Uygunluk değerleri ise şu kriterlere göre belirlenmiştir;

- Tasarım başta belirlenen arazi sınırlarına ve yükseklik şartına uymalıdır.
- Objeler komşu olmalıdır ancak birbiriyle kesişme ya da çakışma yaşamamalıdır.

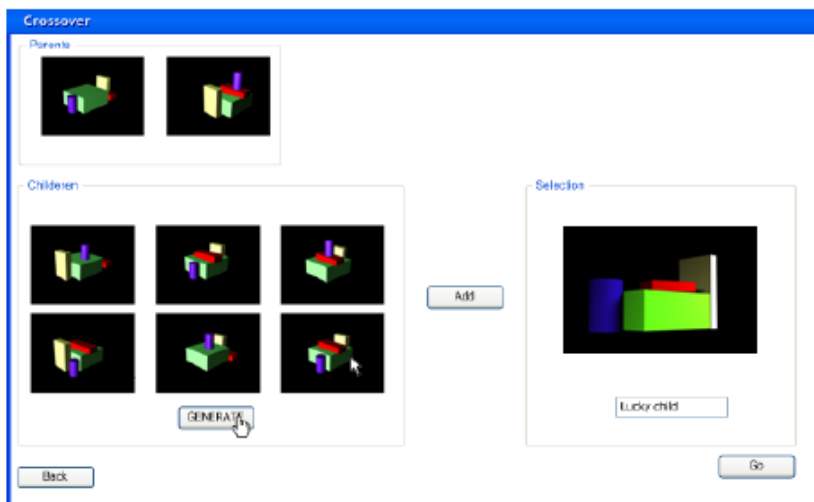
- Minimum cephe ve maksimum hacim kriterini sağlayan adaylara öncelik verilecektir.

Bundan sonraki seçim aşaması tasarımcıya bırakılmıştır. Tasarımcı her seferinde altı adet alternatifi görecektir isterse bir sonraki altılıya geçebilecektir. Bunlar içinden iki tanesini ebeveyn olarak seçip diğer kuşağa gönderecektir. İstedikini seçip başka daha iyi bir alternatif gördüğünde önceki seçiminden vazgeçebilmektedir. Böylece sadece mekanik olmayan tasarımcı etkileşimli bir sistem oluşturulmuştur (Badem, 2007).



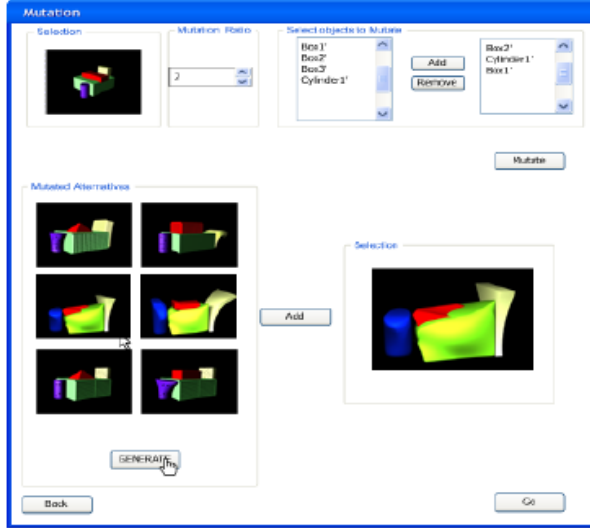
Şekil 2.29. Ebeveyn seçme ekranı (Badem, 2007)

Bundan sonra seçilen ebeveynlerde bulunan objelerin boyut ve konum bilgileri çaprazlanarak yeni nesil üretilmiştir. Aynı mantıkla tek seferde altı adet alternatifi görülebileceği ve bunların içinden seçim yapılabilecek bir arayüz oluşturulmuştur (Badem, 2007).



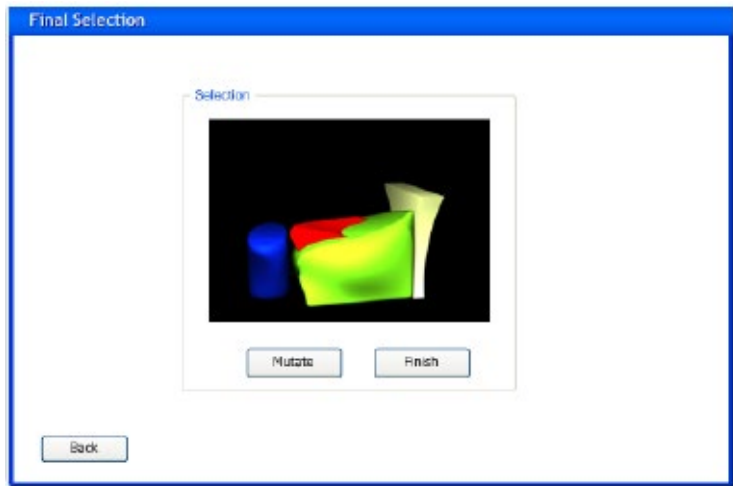
Şekil 2.30. Çaprazlama arayüzü (Badem, 2007)

Daha sonra çocuk birey seçilir ve bu bireye mutasyon uygulanır. Mutasyon için tasarlanan arayüzde iki bölüm vardır. Birinci bölümde mutasyon oranı ve mutasyona uğrayacak objeler seçilir. İkinci bölümde ise seçilen bireyin mutasyona uğramış farklı alternatifleri görülür ve tasarımcı tarafından seçim yapılır.

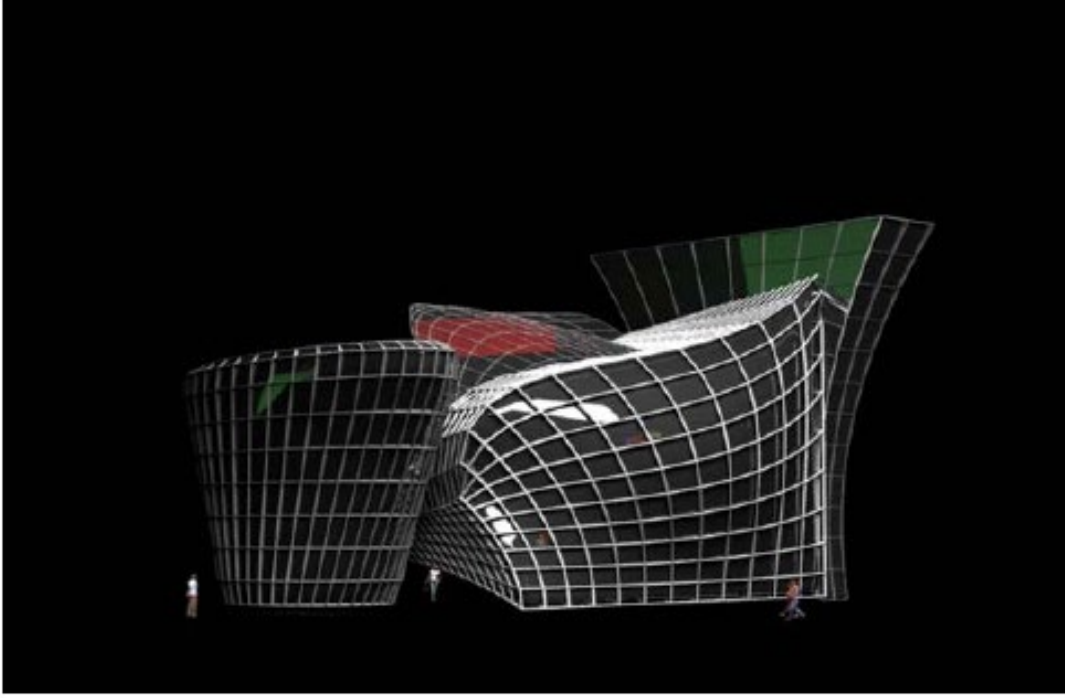


Şekil 2.31. Mutasyon arayüzü (Badem, 2007)

İstenilen tasarım seçildiğinde artık modelin son aşamasına gelinmiştir. Bu ekranda sonuç ürün görülür ve iki seçenek vardır. Tasarımcı sonuç üründen memnunsa GA modelini sonuçlandırabilir veya daha uygun hale getirmek istiyorsa mutasyonlara devam edebilir. Mutasyona devam etmeyi tercih ederse bir önceki arayüz olan mutasyon ekranına geri dönüp aynı mantıkla elinde sonuç ürünü tekrar tekrar mutasyona uğratabilir (Badem, 2007).



Şekil 2.32. Mutasyona uğramış sonuç ürün (Badem, 2007)



Şekil 2.33. Sonuç ürünün mimari sunumu (Badem, 2007)

3. GENETİK ALGORİTMAYA DAYALI BİR MODEL ÖNERİSİ: SUGE MODELİ

Son dönemde üretken yaklaşımlar mimari tasarımlarda sıklıkla kullanılmaya başlanmıştır. Bu üretken yaklaşımlardan birisi de genetik algoritmadır. GA canlılardaki üreme, çoğalma süreçlerini ve evrim sürecini taklit ederek problemlere çözüm bulmayı amaçlamaktadır. Bu şekilde en uygun çözümü daha hızlı bir şekilde ve birçok alternatif çözümü inceleyerek bulabilmektedir.

Bu çalışmada da sudoku bulmacanın kurallarını temel alan mimari biçimler üretmek konusunda tasarımcıya yardımcı olacak bir genetik algoritma modeli geliştirmek amaçlanmıştır. Bunun için öncelikle genetik algoritma mantığı kullanılarak sudoku bulmaca analiz edilmiştir. Bu analizler sonucunda elde edilen veriler, tasarımcının eklediği verilerle birleştirilerek oluşturulacak modelin temel girdileri elde edilmiştir. Bu girdiler ile mimari form üretmeye yarayacak genetik algoritma modeli hazırlanmıştır. Son olarak da modelin nasıl kullanılacağı ve ne tarz formlar üretebileceği üzerine çalışmalar yapılmıştır.

3.1. Sudoku

Sudoku, standart biçimde oluşturulmuş 9×9 boyutlarında bir diyagram üzerinde çözülen, ilk kez 1984 yılında Japonya'da bulmaca olarak yayınlanan ve buradan bütün dünyaya yayıldığı düşünülen, mantığı ve olasılıkları kullanan, eğitilmiş veya eğitimsiz herkesin çözmekten zevk alacağı bir sayı bulmaca oyunudur (Cinan, 2010; Şişman, 2017; www.bilgiustam.com) Japonca "Sayılar tek olmalı" anlamına gelen "Suuji wa dokishin ni kagiru" kelimelerinin kısaltması olan Sudoku, günümüzde Asya'dan, Avrupa ve Kuzey Amerika'ya da yayılan oldukça popüler bir oyundur (www.sudokuoyuna.net).

Sudoku bulmaca soldan sağa ve yukarıdan aşağıya 9'ar karelik bölümlerden oluşur. Bu 81 kare ayrıca $3 \times 3 = 9$ kareden oluşan dokuz bölüme daha ayrılmıştır. Her biri tek bir çözüme sahip olan bulmacada önceden bazı kareler doldurulmuştur. Amaç bu karelerden yola çıkarak bulmacanın geri kalanında her satır, her sütun ve her 9 karelik bölümde 1'den 9'a kadar olan sayıları birer kere kullanarak bütün kareleri doldurmaktır.

Sudoku oyununun amacı üç boyutlu düşünebilme yeteneğini geliştirmektir. Her sudoku oyununun tek bir çözümü olduğu için tahmin edilerek çözmek neredeyse imkânsızdır (sudoku.yazarokur.com). Son yıllarda araştırmacılar, sudoku bulmacayı çözmek için grafik teorisi, yapay zekâ ve genetik algoritma gibi çeşitli yöntemleri uygulamaya başlamışlardır (Geem, 2007).

50 ye yakın sudoku çeşidi bulunmaktadır ve hala yeni türleri üretilmektedir. Bunlardan bazıları; klasik sudoku, bölgesel sudoku, ardışık sudoku, ardışksız sudoku, multi sudoku, köşegen sudoku, toplamli sudoku, zincir Sudoku'dur (sudoku.matematiktutkusu.com).

3.1.1. Sudoku bulmacanın çözümünde genetik algoritma kullanımına dair bir örnek

Hakan Satman Genetik Algoritmalar (2016) kitabında sudoku bulmacayı bir optimizasyon problemi olarak tanımlamıştır. “Tabloya a-b arasındaki tam sayılar nasıl yerleştirilmelidir ki; satır, sütun ve köşegen elemanlarının toplamı c olsun? “cümlesinden yola çıkarak bir genetik algoritma modeli hazırlamıştır. Burada bahsedilen c değerini bulmak için $(1+2+3+...+8+9)/3$ formülünü kullanmıştır. Bu bir bölgesi 3x3 kareden oluşan sudoku için hazırlanmış bir formüldür. Eğer her bir bölümü 4x4 kareden oluşan bir sudoku hazırlanmak istenirse 1’den 16’ya kadar olan sayılar toplanarak 4’e bölünmelidir. Sudoku cinsine göre formül revize edilir.

Bu GA modeli için Şekil 3.1’deki matris hazırlamıştır. Bu matris sudoku bulmacanın 3x3 kareden oluşan bir bölgesini ifade etmektedir. X_{ae} ifadesinde a satır numarasını e ise sütun numarasını göstermektedir. Yani X_{12} ifadesi 1. Satır 2. Sütundaki x değerini ifade etmektedir (Satman, 2016:120).

				Toplam
	x_{11}	x_{12}	x_{13}	15
	x_{21}	x_{22}	x_{23}	15
	x_{31}	x_{32}	x_{33}	15
Toplam	15	15	15	

Şekil 3.1. Genetik algoritma modeli matrisi

Genetik algoritma modeli için oluşturulan kurallar şu şekildedir;

- Her satır, her sütun ve köşegendeki rakamların toplamı 15'e eşit olmalıdır.
- X değerleri 1-9 arasında bir rakam olmalıdır.
- Her rakam sadece bir kere kullanılmalıdır.

Bu kurallar sonucunda genetik algoritmaya dayalı bir bilgisayar modeli oluşturulmuştur. Bunun için hazırlanan bilgisayar kodlarına her satırın toplamının, her sütunun toplamının ve köşegenlerin toplamının 15'ten mutlak farkını içeren değişkenler eklenmiştir. Amaç tüm değişkenlerin 0 olduğu çözümleri bulmaktır. Topluluk büyüklüğü 200, nesil sayısı 100 olarak belirlenmiştir ve program çalıştırılmıştır.

Şekil 3.2'de örnek bir çözüm gösterilmiştir. Ancak bu çözümde bütün satır, sütun ve köşegenlerin toplamı 15 değildir. Bu nedenle bu çözüm optimal bir çözüm değildir (Satman, 2016:121).

	[,1]	[,2]	[,3]
[1,]	6	2	9
[2,]	8	1	5
[3,]	4	9	3

Şekil 3.2. Modelin optimal olmayan sonucu

Sonuç olarak bu şartları sağlayan 3 adet optimum çözüm bulunmuştur. Ancak bunlar aynı çözümün öğelerinin yer değiştirilmesi ile elde edildiğinden aslında tek çözüm vardır (Satman, 2016:122).

	[,1]	[,2]	[,3]
[1,]	2	7	6
[2,]	9	5	1
[3,]	4	3	8

	[,1]	[,2]	[,3]
[1,]	8	3	4
[2,]	1	5	9
[3,]	6	7	2

	[,1]	[,2]	[,3]
[1,]	4	3	8
[2,]	9	5	1
[3,]	2	7	6

Şekil 3.3. Sonuç ürünün farklı yerleşimleri

				Toplam
	2	7	6	15
	9	5	1	15
	4	3	8	15
Toplam	15	15	15	

Şekil 3.4. Sonuç ürün

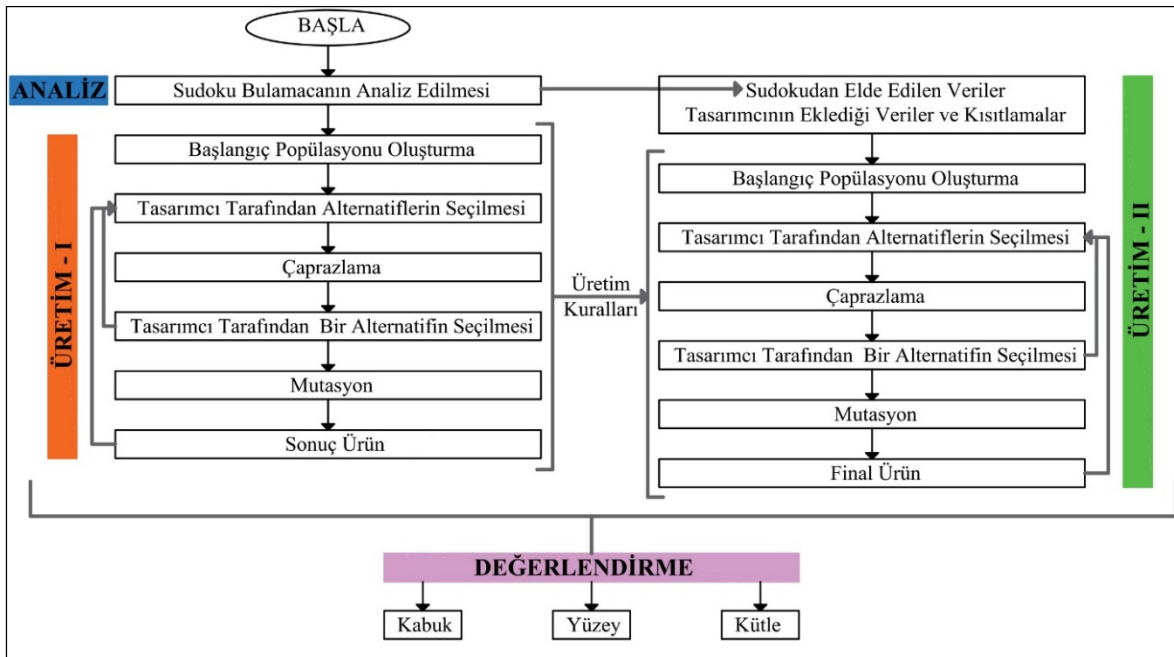
Bu şekilde sudoku probleminin genetik algoritmayla çözümüne dair alternatif bir senaryo kurgulanmıştır. Kurallar ve değişkenler farklılaştırılarak, sudoku probleminin çözümüne dair daha farklı GA modelleri de hazırlanabilir (Satman, 2016:122).

3.2. SuGe Modeli

Mimarlık ve matematik ilişkisi çok eskilere dayanan bir ilişkidir. Mimaride matematikten esinlenmek piramitlerde ve eski çağ tapınaklarında çokça rastlanılan bir üsluptur (Kopar, 2013). Bununla birlikte mimari bir öge oluşturulurken ve form geliştirilirken birçok matematiksel hesaplamayla birlikte simetri, denge vb. birçok öğeler kullanılır.

Bu çalışmada ise amaç mimarlık ve matematiğin ara kesitinde bir form üretme yöntemi geliştirmektir. Bunun için sudoku bulmacanın temel kuralları dikkate alınarak bir evrimsel tasarım süreci yürütülecek ve bir genetik algoritma modeli oluşturulacaktır.

Oluşturulacak model ve işlem adımları aşağıdaki gibidir;



Şekil 3.5. SuGe modeli akış şeması

Temel olarak evrimsel tasarım süreci, üç aşamadan oluşur; tanımlama, üretim ve değerlendirme. Tanımlama aşamasında problem kısıtları, değişkenler kümesi ve başlangıç popülasyonu sayısal olarak oluşturulur. Üretim aşamasında belirlenen kriterler dâhilinde

başlangıç popülasyonunun bireyleri üzerinde genetik operatörler kullanılarak tasarım süreci yürütülür. Değerlendirme aşamasında ise oluşturulan yeni çözümlerden uygun olanlar seçilir (Akbulut, 2009).

Bu mantıkla oluşturulan SuGe modeli de birkaç evreden oluşmaktadır. Birinci evre analiz evresidir. Bu evrede sudoku bulmacanın kuralları ve işleyişi incelenerek, oluşturulacak modelin girdileri elde dılecektir. Bir sonraki evre Üretim-1 evresidir. Bu evrede klasik sudokunun genetik operatörlerle nasıl üretilebileceği çözümlenecektir. Bu aşama ise oluşturulacak modelin temelini oluşturan evredir. Sonra Üretim-2 evresinde bu noktaya kadarki çalışmalardan yola çıkılarak form üretmede kullanılacak genetik algoritma modeli geliştirilecektir. Son aşama ise değerlendirme aşamasıdır. Bu aşamada ise oluşturulan modelin mimaride nasıl kullanılabileceği ve ne tarz formlar üretilebileceği konusunda değerlendirmeler yapılacaktır.

3.2.1. Analiz

Çalışmanın ilk basamağıdır. Amaç klasik sudokunun mantığını ve temel kurallarını analiz etmektir. Bu temel sudoku kuralları Üretim-I ve Üretim-II aşamalarında ilk popülasyonu oluşturmak için kullanılacaktır.

Klasik bir sudoku bulmaca $9 \times 9 = 81$ kareden oluşan ve yine kendi içinde $3 \times 3 = 9$ 'ar karelik bölümlere ayrılmış bir grid üzerinde oynanır. Amaç değerleri verilen hücreleri kullanarak boş hücrelere gelecek sayıları bulmaktır. Kuralları şu şekildedir;

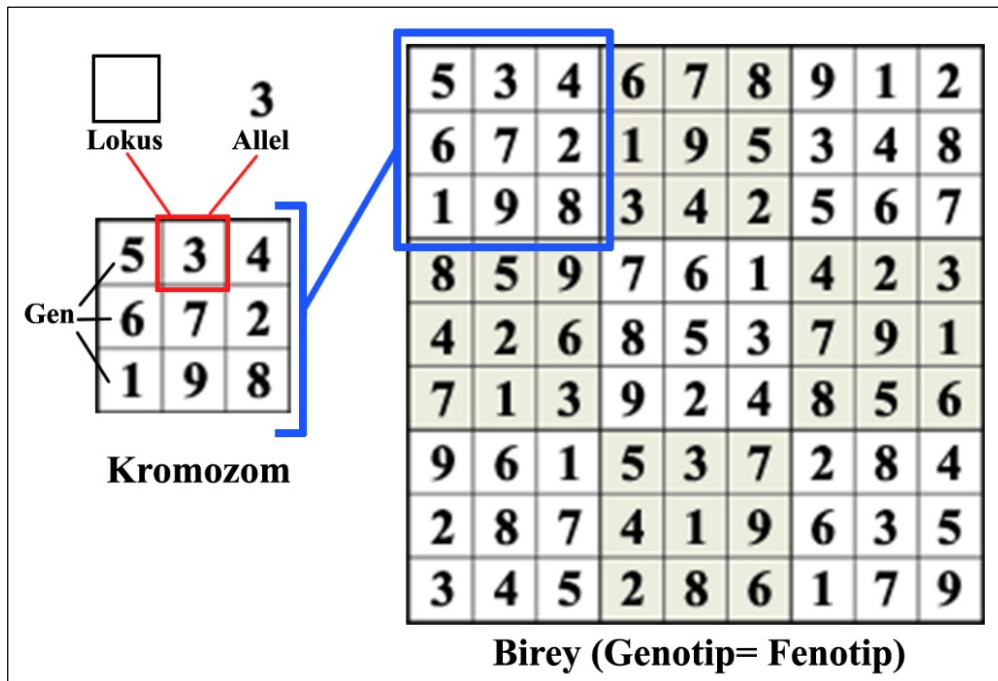
- 1'den 9'a kadar olan rakamlar kullanılmalıdır.
- Her rakamdan eşit sayıda (9'ar adet) kullanılmalıdır
- Her satıra her biri bir kere kullanılacak şekilde 1'den 9'a kadar rakamlar yerleştirilmelidir.
- Her sütuna her biri bir kere kullanılacak şekilde 1'den 9'a kadar rakamlar yerleştirilmelidir.
- Her 3×3 birimlik bölgeye her biri bir kere kullanılacak şekilde 1'den 9'a kadar rakamlar yerleştirilmelidir. (Şekil 3.6.)

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Şekil 3.6. Bir sudoku bulmaca örneği ve çözümü (eodev.com)

Sudokuyu genetik algoritma mantığında incelersek her rakamı bir gen ve dokuz kareden oluşan her bir bölgeyi ise kromozom olarak kabul edebiliriz. Dolayısıyla her bir kare bir genin yerini ifade ettiğinden lokustur ve her bir genin değeri ise alleldir. Dokuz adet kromozomun kurallar çerçevesinde birleşmesi ile birey oluşur. Yani her sudoku, problemin bir çözüm alternatifidir ve popülasyondaki bir bireyi ifade eder. Aynı zamanda her bir sudoku bir genotiptir. Sudokuda genlere verilen bir karşılık görüntü olmadığından genotip ve fenotip bu problemde aynı olacaktır. (Şekil 3.7.)



Şekil 3.7. Sudoku bulmacanın genetik algoritmaya göre öğeleri

3.2.2. Üretim - I

Bu bölümde sudoku bulmacanın ve türevlerinin genetik algoritma ile nasıl oluşturulabileceği araştırılacaktır. Burada elde dillecek üretim kuralları ikinci üretim aşaması için referans olacaktır.

Popülasyon oluşturma

İlk adım alternatif çözümleri içeren bir popülasyon oluşturma adımıdır. Bunun için klasik sudoku mantığında $9 \times 9 = 81$ kareden oluşan sudoku altlığı çalışma alanı olarak, klasik sudoku kuralları ise algoritmanın kuralları olarak kabul edilmiştir. Süreç boyunca sabit olacak popülasyon büyüklüğü ise dokuz olarak belirlenmiştir. Bu şartlar ile ilk popülasyon rastlantısal bir şekilde elde edilmiştir.

<table><tr><td>4</td><td>9</td><td>6</td><td>5</td><td>8</td><td>3</td><td>1</td><td>7</td><td>2</td></tr><tr><td>2</td><td>5</td><td>1</td><td>6</td><td>7</td><td>9</td><td>8</td><td>3</td><td>4</td></tr><tr><td>3</td><td>7</td><td>8</td><td>1</td><td>2</td><td>4</td><td>6</td><td>9</td><td>5</td></tr><tr><td>6</td><td>2</td><td>5</td><td>4</td><td>3</td><td>1</td><td>7</td><td>8</td><td>9</td></tr><tr><td>9</td><td>3</td><td>4</td><td>7</td><td>6</td><td>8</td><td>2</td><td>5</td><td>1</td></tr><tr><td>1</td><td>8</td><td>7</td><td>2</td><td>9</td><td>5</td><td>3</td><td>4</td><td>6</td></tr><tr><td>7</td><td>6</td><td>9</td><td>3</td><td>5</td><td>2</td><td>4</td><td>1</td><td>8</td></tr><tr><td>8</td><td>1</td><td>3</td><td>9</td><td>4</td><td>6</td><td>5</td><td>2</td><td>7</td></tr><tr><td>5</td><td>4</td><td>2</td><td>8</td><td>1</td><td>7</td><td>9</td><td>6</td><td>3</td></tr></table> BİREY -1-	4	9	6	5	8	3	1	7	2	2	5	1	6	7	9	8	3	4	3	7	8	1	2	4	6	9	5	6	2	5	4	3	1	7	8	9	9	3	4	7	6	8	2	5	1	1	8	7	2	9	5	3	4	6	7	6	9	3	5	2	4	1	8	8	1	3	9	4	6	5	2	7	5	4	2	8	1	7	9	6	3	<table><tr><td>1</td><td>8</td><td>5</td><td>6</td><td>2</td><td>3</td><td>7</td><td>9</td><td>4</td></tr><tr><td>4</td><td>2</td><td>7</td><td>9</td><td>5</td><td>8</td><td>3</td><td>1</td><td>6</td></tr><tr><td>6</td><td>9</td><td>3</td><td>4</td><td>1</td><td>7</td><td>2</td><td>8</td><td>5</td></tr><tr><td>5</td><td>4</td><td>8</td><td>3</td><td>7</td><td>2</td><td>9</td><td>6</td><td>1</td></tr><tr><td>7</td><td>6</td><td>9</td><td>1</td><td>4</td><td>5</td><td>8</td><td>2</td><td>3</td></tr><tr><td>2</td><td>3</td><td>1</td><td>8</td><td>6</td><td>9</td><td>5</td><td>4</td><td>7</td></tr><tr><td>9</td><td>5</td><td>6</td><td>7</td><td>8</td><td>4</td><td>1</td><td>3</td><td>2</td></tr><tr><td>8</td><td>1</td><td>2</td><td>5</td><td>3</td><td>6</td><td>4</td><td>7</td><td>9</td></tr><tr><td>3</td><td>7</td><td>4</td><td>2</td><td>9</td><td>1</td><td>6</td><td>5</td><td>8</td></tr></table> BİREY -2-	1	8	5	6	2	3	7	9	4	4	2	7	9	5	8	3	1	6	6	9	3	4	1	7	2	8	5	5	4	8	3	7	2	9	6	1	7	6	9	1	4	5	8	2	3	2	3	1	8	6	9	5	4	7	9	5	6	7	8	4	1	3	2	8	1	2	5	3	6	4	7	9	3	7	4	2	9	1	6	5	8	<table><tr><td>1</td><td>3</td><td>2</td><td>8</td><td>7</td><td>9</td><td>4</td><td>5</td><td>6</td></tr><tr><td>4</td><td>7</td><td>9</td><td>3</td><td>5</td><td>6</td><td>8</td><td>2</td><td>1</td></tr><tr><td>6</td><td>5</td><td>8</td><td>1</td><td>4</td><td>2</td><td>7</td><td>3</td><td>9</td></tr><tr><td>5</td><td>8</td><td>1</td><td>7</td><td>9</td><td>3</td><td>6</td><td>4</td><td>2</td></tr><tr><td>3</td><td>4</td><td>7</td><td>2</td><td>6</td><td>5</td><td>1</td><td>9</td><td>8</td></tr><tr><td>9</td><td>2</td><td>6</td><td>4</td><td>8</td><td>1</td><td>5</td><td>7</td><td>3</td></tr><tr><td>7</td><td>9</td><td>5</td><td>6</td><td>3</td><td>8</td><td>2</td><td>1</td><td>4</td></tr><tr><td>2</td><td>6</td><td>4</td><td>9</td><td>1</td><td>7</td><td>3</td><td>8</td><td>5</td></tr><tr><td>8</td><td>1</td><td>3</td><td>5</td><td>2</td><td>4</td><td>9</td><td>6</td><td>7</td></tr></table> BİREY -3-	1	3	2	8	7	9	4	5	6	4	7	9	3	5	6	8	2	1	6	5	8	1	4	2	7	3	9	5	8	1	7	9	3	6	4	2	3	4	7	2	6	5	1	9	8	9	2	6	4	8	1	5	7	3	7	9	5	6	3	8	2	1	4	2	6	4	9	1	7	3	8	5	8	1	3	5	2	4	9	6	7
4	9	6	5	8	3	1	7	2																																																																																																																																																																																																																																													
2	5	1	6	7	9	8	3	4																																																																																																																																																																																																																																													
3	7	8	1	2	4	6	9	5																																																																																																																																																																																																																																													
6	2	5	4	3	1	7	8	9																																																																																																																																																																																																																																													
9	3	4	7	6	8	2	5	1																																																																																																																																																																																																																																													
1	8	7	2	9	5	3	4	6																																																																																																																																																																																																																																													
7	6	9	3	5	2	4	1	8																																																																																																																																																																																																																																													
8	1	3	9	4	6	5	2	7																																																																																																																																																																																																																																													
5	4	2	8	1	7	9	6	3																																																																																																																																																																																																																																													
1	8	5	6	2	3	7	9	4																																																																																																																																																																																																																																													
4	2	7	9	5	8	3	1	6																																																																																																																																																																																																																																													
6	9	3	4	1	7	2	8	5																																																																																																																																																																																																																																													
5	4	8	3	7	2	9	6	1																																																																																																																																																																																																																																													
7	6	9	1	4	5	8	2	3																																																																																																																																																																																																																																													
2	3	1	8	6	9	5	4	7																																																																																																																																																																																																																																													
9	5	6	7	8	4	1	3	2																																																																																																																																																																																																																																													
8	1	2	5	3	6	4	7	9																																																																																																																																																																																																																																													
3	7	4	2	9	1	6	5	8																																																																																																																																																																																																																																													
1	3	2	8	7	9	4	5	6																																																																																																																																																																																																																																													
4	7	9	3	5	6	8	2	1																																																																																																																																																																																																																																													
6	5	8	1	4	2	7	3	9																																																																																																																																																																																																																																													
5	8	1	7	9	3	6	4	2																																																																																																																																																																																																																																													
3	4	7	2	6	5	1	9	8																																																																																																																																																																																																																																													
9	2	6	4	8	1	5	7	3																																																																																																																																																																																																																																													
7	9	5	6	3	8	2	1	4																																																																																																																																																																																																																																													
2	6	4	9	1	7	3	8	5																																																																																																																																																																																																																																													
8	1	3	5	2	4	9	6	7																																																																																																																																																																																																																																													
<table><tr><td>7</td><td>8</td><td>9</td><td>1</td><td>4</td><td>6</td><td>5</td><td>3</td><td>2</td></tr><tr><td>2</td><td>5</td><td>1</td><td>9</td><td>3</td><td>8</td><td>4</td><td>6</td><td>7</td></tr><tr><td>3</td><td>4</td><td>6</td><td>7</td><td>5</td><td>2</td><td>8</td><td>1</td><td>9</td></tr><tr><td>4</td><td>1</td><td>8</td><td>3</td><td>6</td><td>9</td><td>2</td><td>7</td><td>5</td></tr><tr><td>5</td><td>2</td><td>7</td><td>8</td><td>1</td><td>4</td><td>6</td><td>9</td><td>3</td></tr><tr><td>9</td><td>6</td><td>3</td><td>2</td><td>7</td><td>5</td><td>1</td><td>4</td><td>8</td></tr><tr><td>8</td><td>7</td><td>4</td><td>6</td><td>2</td><td>3</td><td>9</td><td>5</td><td>1</td></tr><tr><td>6</td><td>3</td><td>2</td><td>5</td><td>9</td><td>1</td><td>7</td><td>8</td><td>4</td></tr><tr><td>1</td><td>9</td><td>5</td><td>4</td><td>8</td><td>7</td><td>3</td><td>2</td><td>6</td></tr></table> BİREY -4-	7	8	9	1	4	6	5	3	2	2	5	1	9	3	8	4	6	7	3	4	6	7	5	2	8	1	9	4	1	8	3	6	9	2	7	5	5	2	7	8	1	4	6	9	3	9	6	3	2	7	5	1	4	8	8	7	4	6	2	3	9	5	1	6	3	2	5	9	1	7	8	4	1	9	5	4	8	7	3	2	6	<table><tr><td>2</td><td>4</td><td>8</td><td>6</td><td>1</td><td>3</td><td>7</td><td>9</td><td>5</td></tr><tr><td>5</td><td>3</td><td>1</td><td>8</td><td>7</td><td>9</td><td>2</td><td>6</td><td>4</td></tr><tr><td>6</td><td>9</td><td>7</td><td>4</td><td>5</td><td>2</td><td>8</td><td>1</td><td>3</td></tr><tr><td>4</td><td>2</td><td>3</td><td>1</td><td>6</td><td>8</td><td>5</td><td>7</td><td>9</td></tr><tr><td>7</td><td>5</td><td>6</td><td>9</td><td>3</td><td>4</td><td>1</td><td>8</td><td>2</td></tr><tr><td>8</td><td>1</td><td>9</td><td>5</td><td>2</td><td>7</td><td>4</td><td>3</td><td>6</td></tr><tr><td>3</td><td>6</td><td>2</td><td>7</td><td>8</td><td>5</td><td>9</td><td>4</td><td>1</td></tr><tr><td>9</td><td>8</td><td>5</td><td>3</td><td>4</td><td>1</td><td>6</td><td>2</td><td>7</td></tr><tr><td>1</td><td>7</td><td>4</td><td>2</td><td>9</td><td>6</td><td>3</td><td>5</td><td>8</td></tr></table> BİREY -5-	2	4	8	6	1	3	7	9	5	5	3	1	8	7	9	2	6	4	6	9	7	4	5	2	8	1	3	4	2	3	1	6	8	5	7	9	7	5	6	9	3	4	1	8	2	8	1	9	5	2	7	4	3	6	3	6	2	7	8	5	9	4	1	9	8	5	3	4	1	6	2	7	1	7	4	2	9	6	3	5	8	<table><tr><td>5</td><td>9</td><td>1</td><td>8</td><td>7</td><td>4</td><td>6</td><td>2</td><td>3</td></tr><tr><td>4</td><td>7</td><td>8</td><td>6</td><td>3</td><td>2</td><td>5</td><td>9</td><td>1</td></tr><tr><td>6</td><td>2</td><td>3</td><td>1</td><td>9</td><td>5</td><td>4</td><td>8</td><td>7</td></tr><tr><td>1</td><td>8</td><td>5</td><td>7</td><td>6</td><td>9</td><td>3</td><td>4</td><td>2</td></tr><tr><td>2</td><td>6</td><td>9</td><td>4</td><td>8</td><td>3</td><td>7</td><td>1</td><td>5</td></tr><tr><td>7</td><td>3</td><td>4</td><td>2</td><td>5</td><td>1</td><td>8</td><td>6</td><td>9</td></tr><tr><td>9</td><td>4</td><td>6</td><td>3</td><td>1</td><td>7</td><td>2</td><td>5</td><td>8</td></tr><tr><td>3</td><td>5</td><td>2</td><td>9</td><td>4</td><td>8</td><td>1</td><td>7</td><td>6</td></tr><tr><td>8</td><td>1</td><td>7</td><td>5</td><td>2</td><td>6</td><td>9</td><td>3</td><td>4</td></tr></table> BİREY -6-	5	9	1	8	7	4	6	2	3	4	7	8	6	3	2	5	9	1	6	2	3	1	9	5	4	8	7	1	8	5	7	6	9	3	4	2	2	6	9	4	8	3	7	1	5	7	3	4	2	5	1	8	6	9	9	4	6	3	1	7	2	5	8	3	5	2	9	4	8	1	7	6	8	1	7	5	2	6	9	3	4
7	8	9	1	4	6	5	3	2																																																																																																																																																																																																																																													
2	5	1	9	3	8	4	6	7																																																																																																																																																																																																																																													
3	4	6	7	5	2	8	1	9																																																																																																																																																																																																																																													
4	1	8	3	6	9	2	7	5																																																																																																																																																																																																																																													
5	2	7	8	1	4	6	9	3																																																																																																																																																																																																																																													
9	6	3	2	7	5	1	4	8																																																																																																																																																																																																																																													
8	7	4	6	2	3	9	5	1																																																																																																																																																																																																																																													
6	3	2	5	9	1	7	8	4																																																																																																																																																																																																																																													
1	9	5	4	8	7	3	2	6																																																																																																																																																																																																																																													
2	4	8	6	1	3	7	9	5																																																																																																																																																																																																																																													
5	3	1	8	7	9	2	6	4																																																																																																																																																																																																																																													
6	9	7	4	5	2	8	1	3																																																																																																																																																																																																																																													
4	2	3	1	6	8	5	7	9																																																																																																																																																																																																																																													
7	5	6	9	3	4	1	8	2																																																																																																																																																																																																																																													
8	1	9	5	2	7	4	3	6																																																																																																																																																																																																																																													
3	6	2	7	8	5	9	4	1																																																																																																																																																																																																																																													
9	8	5	3	4	1	6	2	7																																																																																																																																																																																																																																													
1	7	4	2	9	6	3	5	8																																																																																																																																																																																																																																													
5	9	1	8	7	4	6	2	3																																																																																																																																																																																																																																													
4	7	8	6	3	2	5	9	1																																																																																																																																																																																																																																													
6	2	3	1	9	5	4	8	7																																																																																																																																																																																																																																													
1	8	5	7	6	9	3	4	2																																																																																																																																																																																																																																													
2	6	9	4	8	3	7	1	5																																																																																																																																																																																																																																													
7	3	4	2	5	1	8	6	9																																																																																																																																																																																																																																													
9	4	6	3	1	7	2	5	8																																																																																																																																																																																																																																													
3	5	2	9	4	8	1	7	6																																																																																																																																																																																																																																													
8	1	7	5	2	6	9	3	4																																																																																																																																																																																																																																													
<table><tr><td>2</td><td>8</td><td>3</td><td>7</td><td>4</td><td>1</td><td>9</td><td>5</td><td>6</td></tr><tr><td>7</td><td>4</td><td>5</td><td>6</td><td>3</td><td>9</td><td>8</td><td>1</td><td>2</td></tr><tr><td>6</td><td>9</td><td>1</td><td>8</td><td>5</td><td>2</td><td>3</td><td>7</td><td>4</td></tr><tr><td>5</td><td>1</td><td>8</td><td>2</td><td>7</td><td>3</td><td>4</td><td>6</td><td>9</td></tr><tr><td>3</td><td>6</td><td>7</td><td>4</td><td>9</td><td>8</td><td>1</td><td>2</td><td>5</td></tr><tr><td>9</td><td>2</td><td>4</td><td>5</td><td>1</td><td>6</td><td>7</td><td>8</td><td>3</td></tr><tr><td>1</td><td>5</td><td>9</td><td>3</td><td>6</td><td>7</td><td>2</td><td>4</td><td>8</td></tr><tr><td>8</td><td>7</td><td>6</td><td>9</td><td>2</td><td>4</td><td>5</td><td>3</td><td>1</td></tr><tr><td>4</td><td>3</td><td>2</td><td>1</td><td>8</td><td>5</td><td>6</td><td>9</td><td>7</td></tr></table> BİREY -7-	2	8	3	7	4	1	9	5	6	7	4	5	6	3	9	8	1	2	6	9	1	8	5	2	3	7	4	5	1	8	2	7	3	4	6	9	3	6	7	4	9	8	1	2	5	9	2	4	5	1	6	7	8	3	1	5	9	3	6	7	2	4	8	8	7	6	9	2	4	5	3	1	4	3	2	1	8	5	6	9	7	<table><tr><td>4</td><td>1</td><td>8</td><td>7</td><td>6</td><td>9</td><td>3</td><td>5</td><td>2</td></tr><tr><td>7</td><td>5</td><td>2</td><td>8</td><td>1</td><td>3</td><td>9</td><td>4</td><td>6</td></tr><tr><td>6</td><td>3</td><td>9</td><td>5</td><td>4</td><td>2</td><td>8</td><td>1</td><td>7</td></tr><tr><td>2</td><td>8</td><td>4</td><td>3</td><td>7</td><td>6</td><td>5</td><td>9</td><td>1</td></tr><tr><td>9</td><td>6</td><td>3</td><td>2</td><td>5</td><td>1</td><td>4</td><td>7</td><td>8</td></tr><tr><td>1</td><td>7</td><td>5</td><td>9</td><td>8</td><td>4</td><td>6</td><td>2</td><td>3</td></tr><tr><td>3</td><td>9</td><td>6</td><td>4</td><td>2</td><td>7</td><td>1</td><td>8</td><td>5</td></tr><tr><td>5</td><td>4</td><td>7</td><td>1</td><td>3</td><td>8</td><td>2</td><td>6</td><td>9</td></tr><tr><td>8</td><td>2</td><td>1</td><td>6</td><td>9</td><td>5</td><td>7</td><td>3</td><td>4</td></tr></table> BİREY -8-	4	1	8	7	6	9	3	5	2	7	5	2	8	1	3	9	4	6	6	3	9	5	4	2	8	1	7	2	8	4	3	7	6	5	9	1	9	6	3	2	5	1	4	7	8	1	7	5	9	8	4	6	2	3	3	9	6	4	2	7	1	8	5	5	4	7	1	3	8	2	6	9	8	2	1	6	9	5	7	3	4	<table><tr><td>9</td><td>1</td><td>6</td><td>7</td><td>3</td><td>5</td><td>4</td><td>2</td><td>8</td></tr><tr><td>5</td><td>8</td><td>3</td><td>6</td><td>2</td><td>4</td><td>7</td><td>9</td><td>1</td></tr><tr><td>7</td><td>2</td><td>4</td><td>8</td><td>1</td><td>9</td><td>5</td><td>3</td><td>6</td></tr><tr><td>4</td><td>7</td><td>1</td><td>2</td><td>6</td><td>3</td><td>9</td><td>8</td><td>5</td></tr><tr><td>3</td><td>9</td><td>8</td><td>4</td><td>5</td><td>1</td><td>2</td><td>6</td><td>7</td></tr><tr><td>2</td><td>6</td><td>5</td><td>9</td><td>7</td><td>8</td><td>1</td><td>4</td><td>3</td></tr><tr><td>8</td><td>5</td><td>2</td><td>1</td><td>4</td><td>6</td><td>3</td><td>7</td><td>9</td></tr><tr><td>1</td><td>4</td><td>9</td><td>3</td><td>8</td><td>7</td><td>6</td><td>5</td><td>2</td></tr><tr><td>6</td><td>3</td><td>7</td><td>5</td><td>9</td><td>2</td><td>8</td><td>1</td><td>4</td></tr></table> BİREY -9-	9	1	6	7	3	5	4	2	8	5	8	3	6	2	4	7	9	1	7	2	4	8	1	9	5	3	6	4	7	1	2	6	3	9	8	5	3	9	8	4	5	1	2	6	7	2	6	5	9	7	8	1	4	3	8	5	2	1	4	6	3	7	9	1	4	9	3	8	7	6	5	2	6	3	7	5	9	2	8	1	4
2	8	3	7	4	1	9	5	6																																																																																																																																																																																																																																													
7	4	5	6	3	9	8	1	2																																																																																																																																																																																																																																													
6	9	1	8	5	2	3	7	4																																																																																																																																																																																																																																													
5	1	8	2	7	3	4	6	9																																																																																																																																																																																																																																													
3	6	7	4	9	8	1	2	5																																																																																																																																																																																																																																													
9	2	4	5	1	6	7	8	3																																																																																																																																																																																																																																													
1	5	9	3	6	7	2	4	8																																																																																																																																																																																																																																													
8	7	6	9	2	4	5	3	1																																																																																																																																																																																																																																													
4	3	2	1	8	5	6	9	7																																																																																																																																																																																																																																													
4	1	8	7	6	9	3	5	2																																																																																																																																																																																																																																													
7	5	2	8	1	3	9	4	6																																																																																																																																																																																																																																													
6	3	9	5	4	2	8	1	7																																																																																																																																																																																																																																													
2	8	4	3	7	6	5	9	1																																																																																																																																																																																																																																													
9	6	3	2	5	1	4	7	8																																																																																																																																																																																																																																													
1	7	5	9	8	4	6	2	3																																																																																																																																																																																																																																													
3	9	6	4	2	7	1	8	5																																																																																																																																																																																																																																													
5	4	7	1	3	8	2	6	9																																																																																																																																																																																																																																													
8	2	1	6	9	5	7	3	4																																																																																																																																																																																																																																													
9	1	6	7	3	5	4	2	8																																																																																																																																																																																																																																													
5	8	3	6	2	4	7	9	1																																																																																																																																																																																																																																													
7	2	4	8	1	9	5	3	6																																																																																																																																																																																																																																													
4	7	1	2	6	3	9	8	5																																																																																																																																																																																																																																													
3	9	8	4	5	1	2	6	7																																																																																																																																																																																																																																													
2	6	5	9	7	8	1	4	3																																																																																																																																																																																																																																													
8	5	2	1	4	6	3	7	9																																																																																																																																																																																																																																													
1	4	9	3	8	7	6	5	2																																																																																																																																																																																																																																													
6	3	7	5	9	2	8	1	4																																																																																																																																																																																																																																													

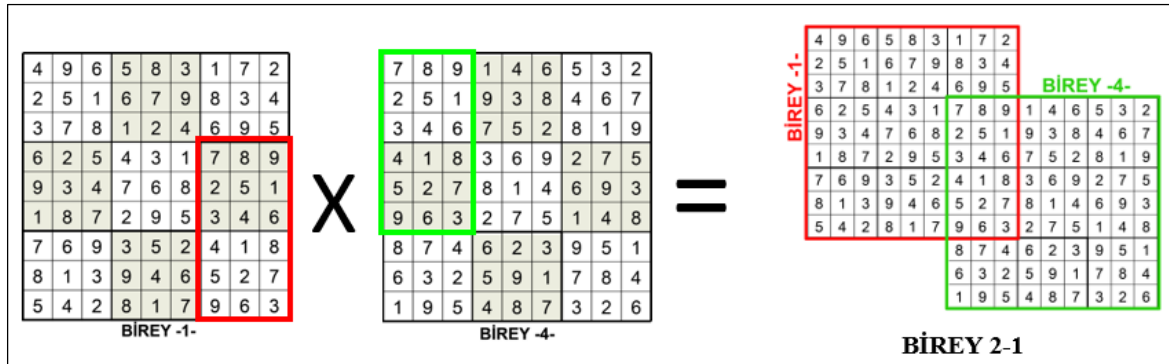
Şekil 3.8. Üretim-I adımı ilk popülasyonu

Oluşturulan ilk popülasyondaki her bir üye bir sudoku örneğidir. Yani tek adımda klasik sudokular elde edilmiştir. GA yapısı gereği paraleldir. Yani aynı anda birçok yönde arama yapar. Kullanıcı istediği adımda algoritmayı durdurabilir. Bir ya da birkaç adım geriye gidebilir. En başa dönüp başka bir ilk popülasyon oluşturup aramaya devam edebilir. Dolayısıyla eğer istenilen çözüm oluştuysa bu noktada kullanıcı algoritmayı durdurabilir. Ancak bu çalışmada diğer sudoku türevlerini ve üretim adımlarını anlayabilmek için bu algoritmanın işlem adımlarına devam edilecektir.

Seçim ve çaprazlama

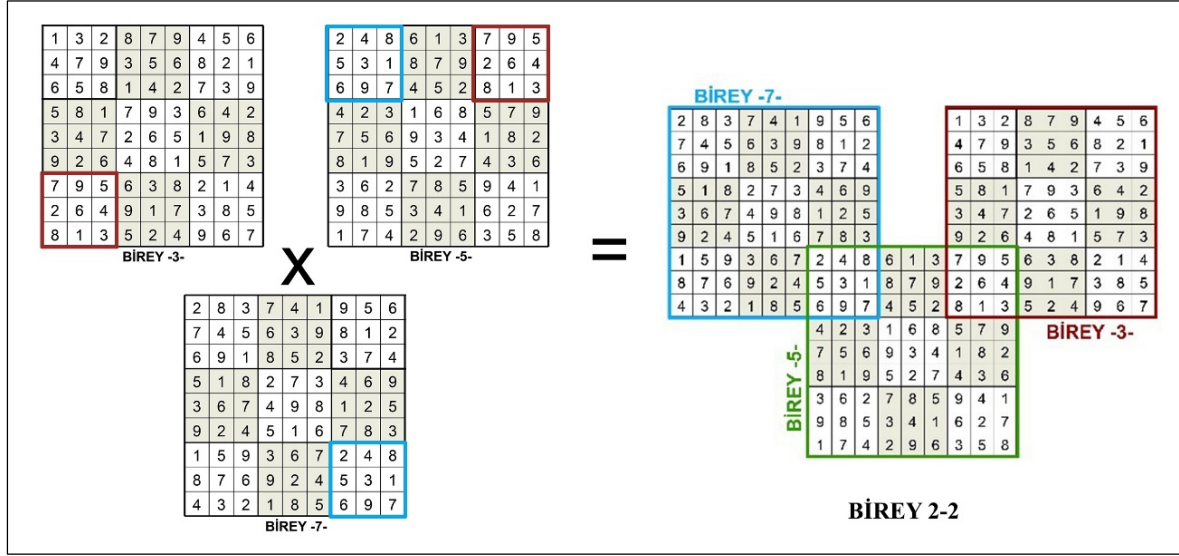
Sıradaki adım ebeveyn görevi göreceк çözümlerin seçilip çaprazlanması adıımıdır. Farklı çaprazlama biçimlerini görmek amacıyla bu çalışmada, iki birey seçip çaprazlamak yerine farklı sayıda ve çeşitlilikte birey seçilip çaprazlanmıştır. Çaprazlama işlemi yapılırken seçilen ebeveynlerde aynı olan kromozomlar çakıştırılmıştır.

Örneğin 1 ve 4 nolu bireyin ortak 2 adet geni bulunmaktadır. Bu iki bireyin ortak olan genleri çakıştırılarak ikinci jenerasyon, 2-1 nolu birey elde edilmiştir.



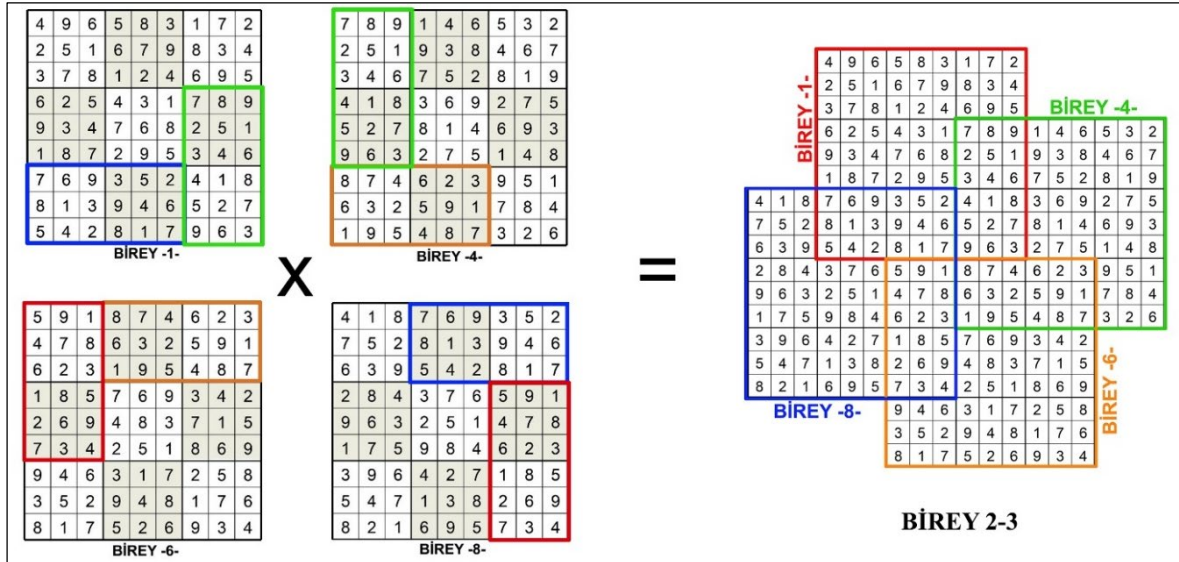
Şekil 3.9. İki bireyin çaprazlanması örneği

3 bireyin çaprazlandığı bir örnek olarak 3,5 ve 7 nolu bireyler çaprazlanmıştır. Bu örnekte 3 ve 5 nolu bireyin 1 adet, 5 ve 7 nolu bireyin 1 adet ortak geni vardır. Bu genler birleştirilerek 2-2 nolu birey oluşturulmuştur.



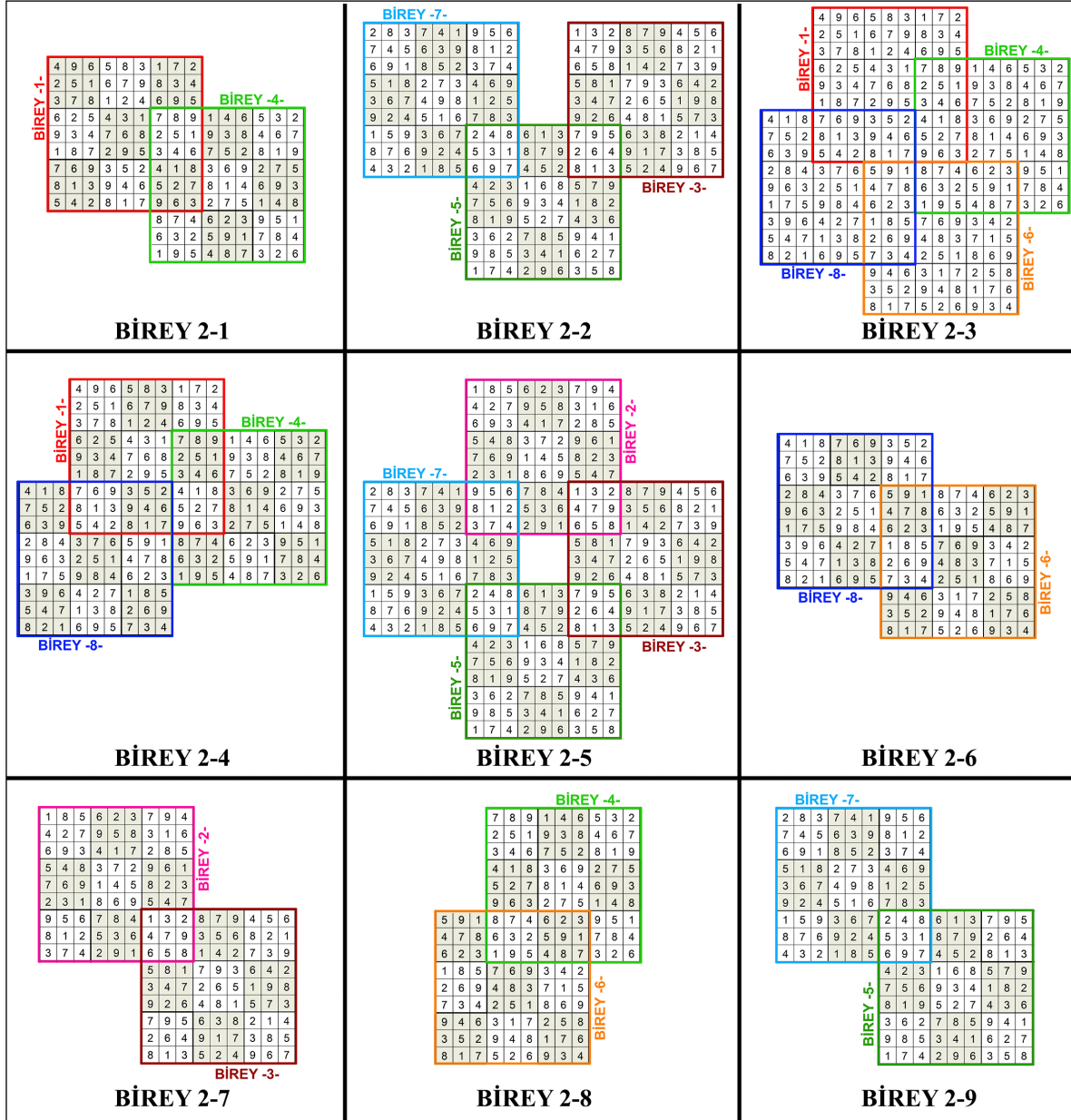
Şekil 3.10. Üç bireyin çaprazlanması örneği

4 bireyin çaprazlandığı örnek içinse 1,4,6 ve 8 nolu bireyler çaprazlanmıştır. Bu örnekte 1 ve 4 nolu bireyin 2 adet, 4 ve 6 nolu bireyin 2 adet, 6 ve 8 nolu bireyin 2 adet ve 1 ve 8 nolu bireyin 2 adet ortak geni vardır. Bu ortak genler birleştirilerek 2-3 nolu birey oluşturulmuştur.



Şekil 3.11. Dört bireyin çaprazlanması örneği

Örneklerde gösterildiği şekilde 2, 3 ve 4 adet ebeveyn kullanılarak başka çaprazlamalar da yapılmıştır. Böylece dokuz bireyden oluşan 2. Jenerasyon elde edilmiştir. Bu ikinci jenerasyon aşağıdadır.



Şekil 3.12. Üretim-I ikinci jenerasyonu

Çaprazlama sonucu elde edilen 2. Jenerasyonun her bir bireyi bir multi sudoku örneğidir.

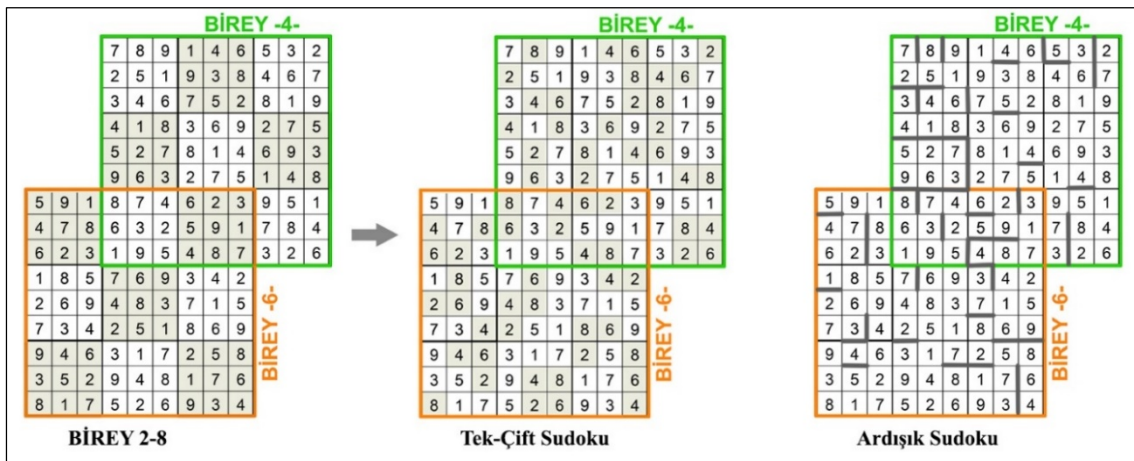
Multi sudoku 2, 3, 4, veya 5 sudokunun içiçe bir arada bulunduğu sudoku çeşididir. Aynı sudoku kuralları ile çözülürler ancak bütün sudokuların çözümü birbirine bağlıdır. Her bir sudoku kendi içinde değerlendirilir fakat multi sudokunun bir parçasını çözebilmek için diğer parçadaki rakamlardan faydalanmak gereklidir. (sudoku.matematiktutkusu.com)

Seçim ve mutasyon

Son aşama mutasyon aşamasıdır. Bu aşamada elde edilen çözümlerde optimuma yaklaştırmak amacıyla, bazı değişiklikler yapılır. Farklı mutasyon çeşitleri vardır. İstenirse belli mutasyon oranları tanımlanarak tüm popülasyon mutasyona uğratılabileceği gibi mevcut haliyle zaten optimuma yakın olan bir çözüm seçilerek kullanıcı kontrolünde sadece bu bireye mutasyon da uygulanabilir. Mutasyon operatörünün bir diğer görevi çeşitlilik sağlamaktır. Algoritmada elde edilen sonuç ürünlerin hepsi birbirine benzemeye başladıysa ya da elde edilen çözümler optimum çözümden hala uzaksa mutasyon operatörü ile mutasyona uğratılan bireyler bir ya da birkaç kuşak geriye götürülüp o noktadan sürece dahil edilebilirler. Böylece süreç tekrar edildiğinde daha farklı çözümler elde edileceğinden çeşitlilik sağlanır ve en iyi çözüme daha çok yaklaşılabılır.

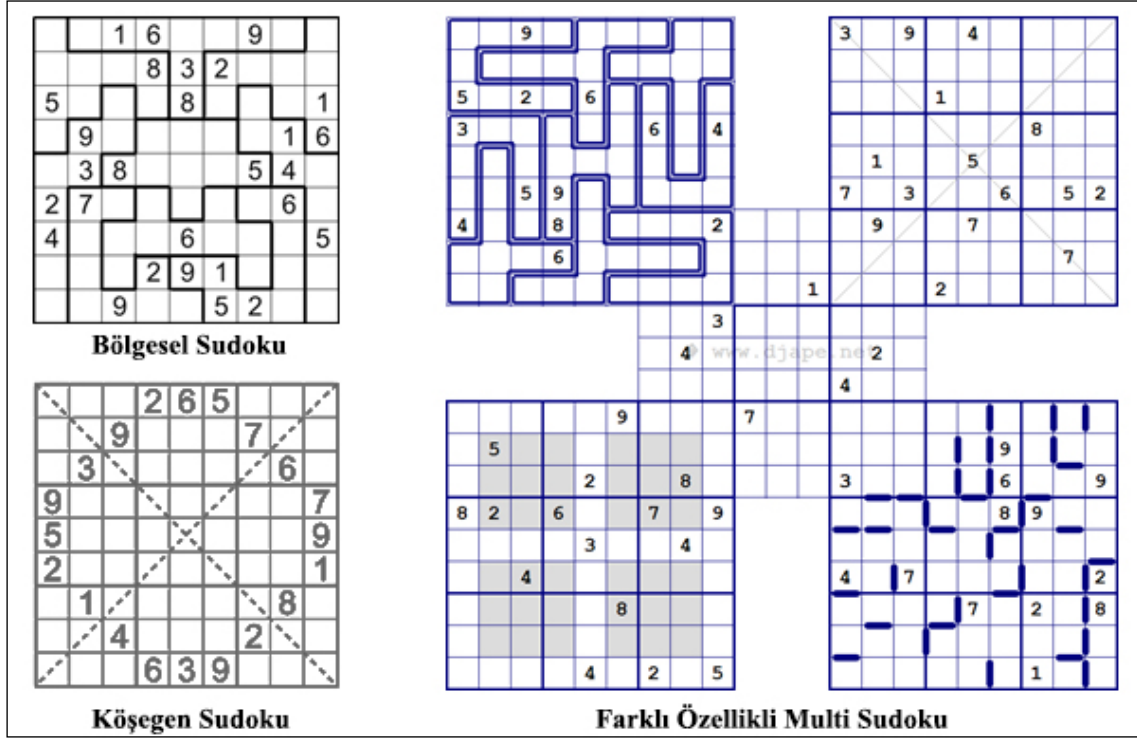
Sudokuda mutasyonu tanımlamak için yine sudoku çeşitleri dikkate alınmıştır. Çalışmada kullanılan sudoku altılığında her bir $3 \times 3 = 9$ br'den oluşan bölgeleri ayrıştırabilmek amacı ile gri ve beyaz renkler kullanılmıştır. Bu nedenle elde edilen tüm sudokular bu şekildedir. Ancak bunun yerine çift rakamların bulunduğu hücreleri gri renk, tek rakamların bulunduğu hücreleri beyaz renk yapacak şekilde bir mutasyon uygulayabiliriz. Bu şekilde elde ettiğimiz yeni sudokunun çeşidi tek-çift sudokudur.

Başka bir mutasyon çeşidi olarak ardışık sayıların arasında kalan çizgiyi koyu renkle ifade edecek değişiklik yapabiliriz. Bu şekilde elde edeceğimiz sudoku çeşidi ise ardışık sudokudur.



Şekil 3.13. Üretim-1 mutasyon örnekleri

Başka mutasyon örnekleri olarak tanımlı olan 3x3 lük bölgeleri deforme ederek bölgesel sudoku, köşegen kuralı ekleyerek köşegen sudoku ya da bunların birden fazlasını tek sudoku üzerinde kullanarak farklı özellikli multi sudokular elde edilebilir.



Şekil 3.14. Farklı sudoku örnekleri (www.djape.net, sudoku.matematiktutkusu.com)

3.2.3. Üretim - II

Bu bölüme kadar sudoku bulmacanın analizi yapılmış, klasik sudoku bulmaca ve türevlerinin genetik algoritma modeli ile nasıl oluşturulacağını ve üretim basamaklarının neler olduğu araştırılmıştır. Bu bölümde bütün bu çalışmalardan yola çıkılarak bir sudoku bulmacanın mantığıyla nasıl mimari biçim üretilebileceği üzerine bir model tanıtılacaktır.

Popülasyon oluşturma

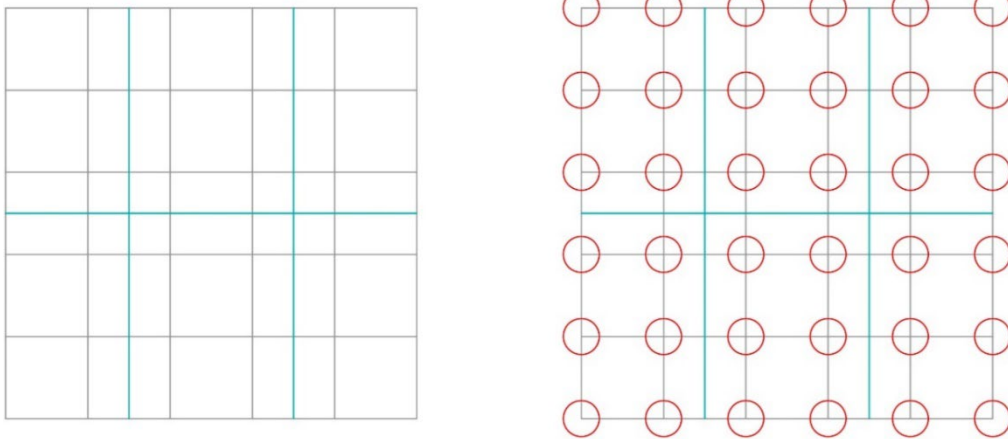
Bu adımda öncelikli olarak ilk üretim evresindeki kurallar ve kısıtlamalar dikkate alınacaktır. Ancak burada tasarımcı faktörü önemlidir. Bu kurallar tasarımcı tarafından elde edilmek istenen biçim dikkate alınarak değiştirilip geliştirilecektir. Ayrıca sudoku bulmacada bulunmayan, tasarımcının eklediği bir takım kısıtlamalar ve kurallar da bulunmaktadır.

Tasarımcı tarafından belirlenen kurallar;

1. Kural: Sudokuda $9 \times 9 = 81$ karelik bir grid kullanılırken üreteceğimiz bu modelde $5 \times 5 = 25$ kareden oluşan ve kendi içinde 6 bölüme ayrılmış bir grid temel altlık olarak kullanılacaktır.
2. Kural: Rakamlar kutucuklara değil kesişim noktalarına yerleştirilecektir. (Şekil 3.15)
3. Kural: 1'den 6'ya kadar olan rakamlar kullanılacaktır.

Bu kurallarla birlikte sudokunun temel kuralları da dikkate alınacaktır ancak grid altı noktalı olduğundan 1'den 9'a kadar olan rakamlar değil 1'den 6'ya kadar olan rakamlar kullanılacaktır. Dolayısıyla diğer kurallar şu şekildedir;

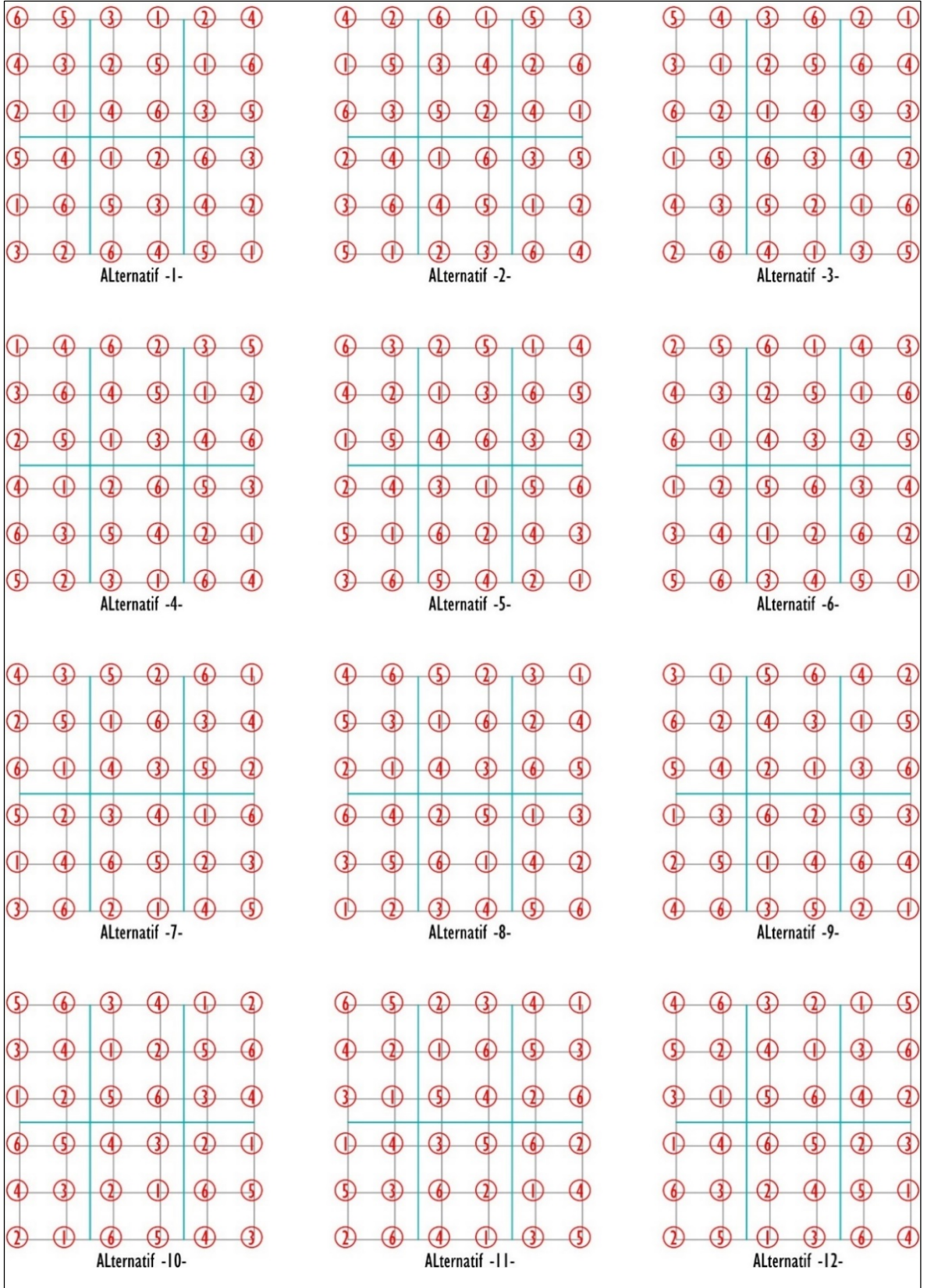
4. Kural: Her rakamdan eşit sayıda (6'şar adet) kullanılmalıdır.
5. Kural: Her satıra her biri bir kere kullanılacak şekilde 1'den 6'a kadar rakamlar yerleştirilmelidir.
6. Kural: Her sütuna her biri bir kere kullanılacak şekilde 1'den 6'a kadar rakamlar yerleştirilmelidir.
7. Kural: Her bölgeye her biri bir kere kullanılacak şekilde 1'den 6'a kadar rakamlar yerleştirilmelidir.



Şekil 3.15. Üretim-II'de kullanılacak grid ve rakamların geleceği noktalar

Üretim-I'deki mantıkla bu adımda da her rakam bir gen, genlerin yerleştirildiği noktalar lokus, genlerin değeri allel olarak kabul edilmiştir. Dolayısıyla altı adet gen içeren işaretlenmiş her bölge bir kromozomdur ve altı kromozomun birleşimi de popülasyondaki her bir çözümü yani bireyleri ifade eder. Bu bireyler aynı zamanda genotiptir.

Bu kurallar ile on iki adet alternatiftan oluşan ilk popülasyon rastgele bir şekilde oluşturulur.

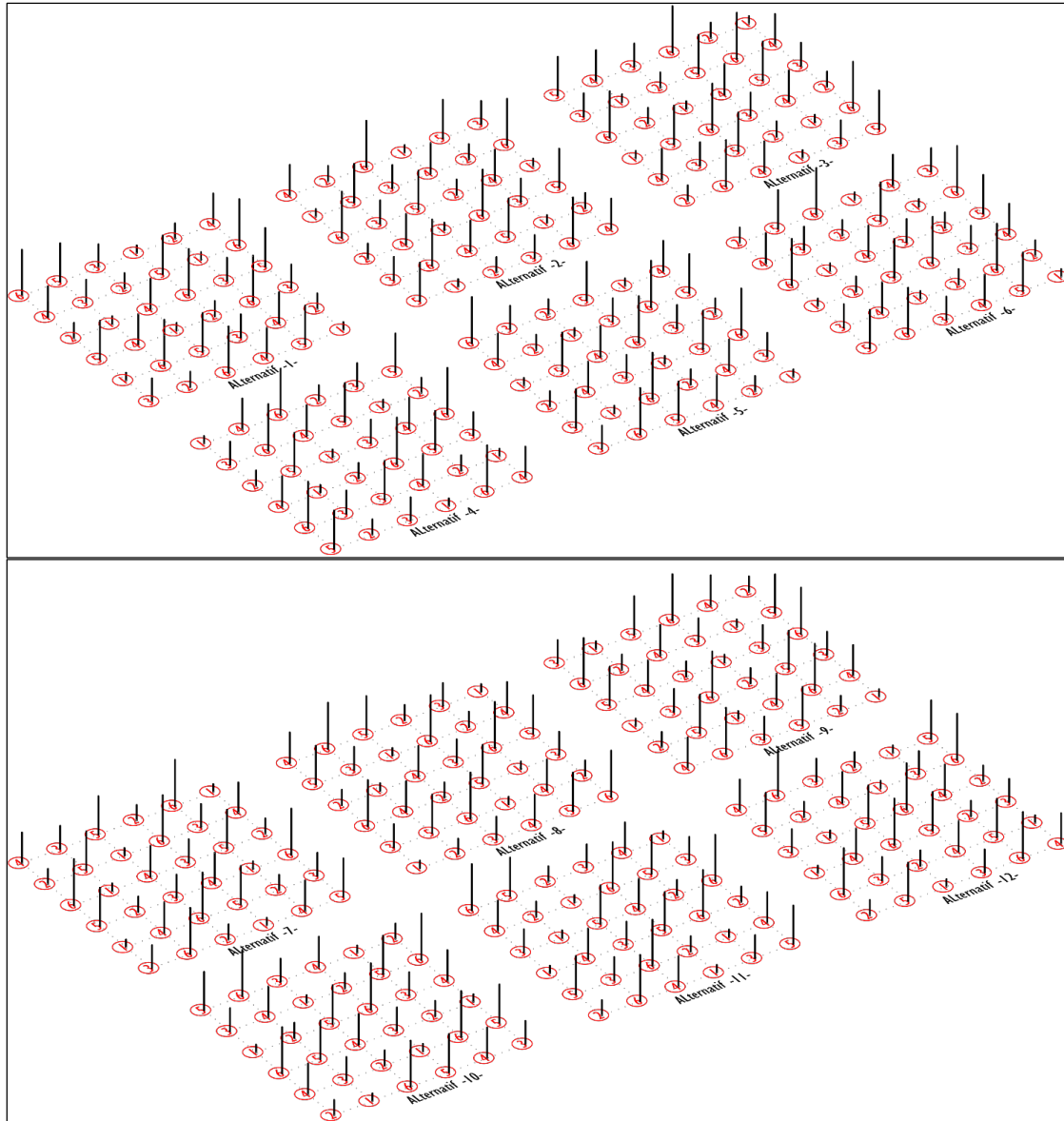


Şekil 3.16. Üretim-II ilk popülasyon

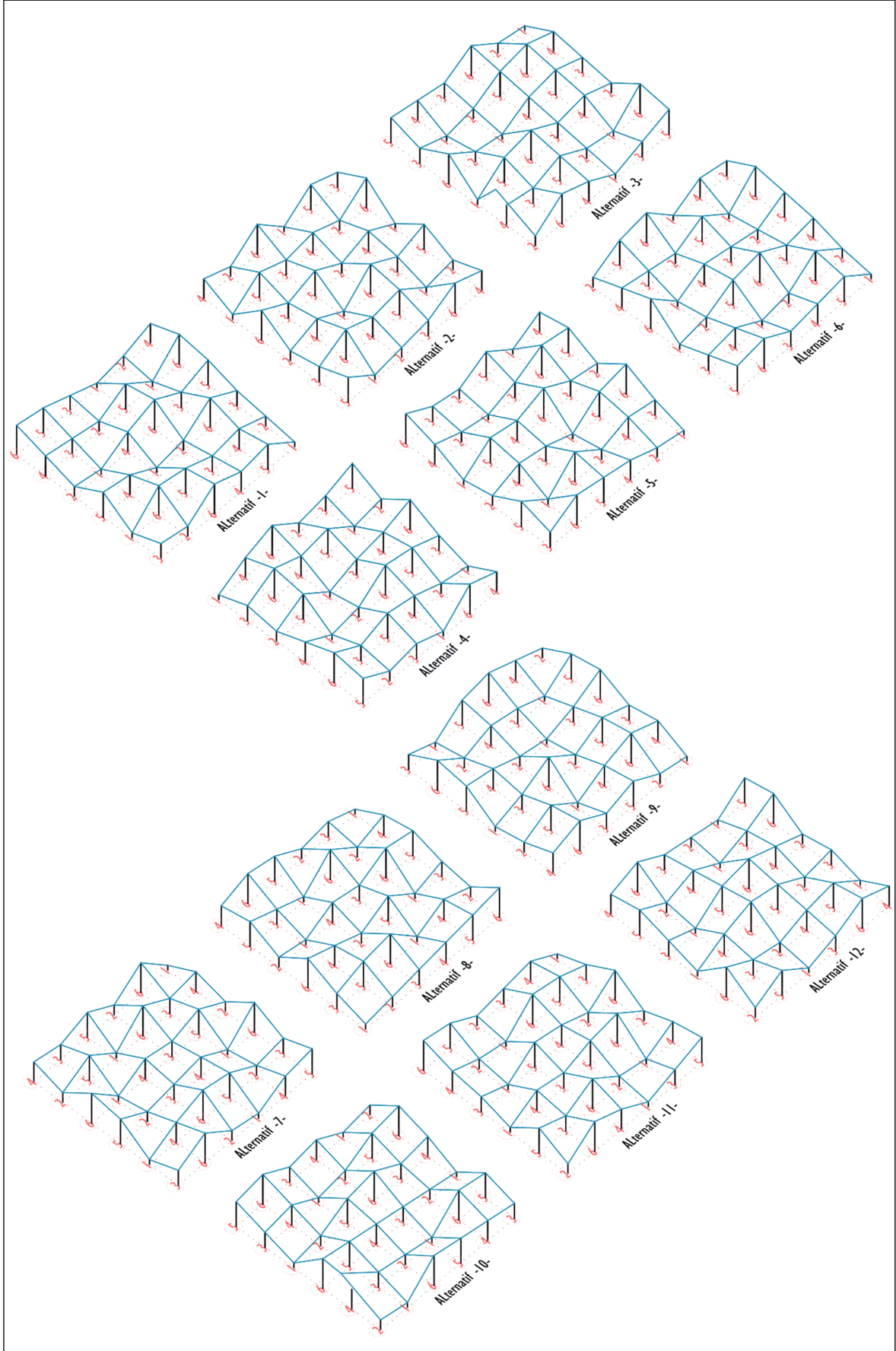
Üretim I evresinde genotip ve fenotip aynı idi. Ancak üretim-II evresinde çözümlerin mimari biçimlere dönüşebilmesi için genoipleri karşılayan fenotipler olmalıdır. Bu fenotipleri elde etmek için üç adımdan oluşan bir yöntem belirlenmiştir.

1. Adım: Her noktaya, üstünde yazan sayı değeri kadar br boyutunda z yönüne doğru bir çizgi çizilecektir.
2. Adım: Çizilen bu dikey çizgilerin üst noktası birleştirilecektir.
3. Adım: Düşey çizgilerin birleşiminden oluşan düzlemin araları doldurulacaktır.

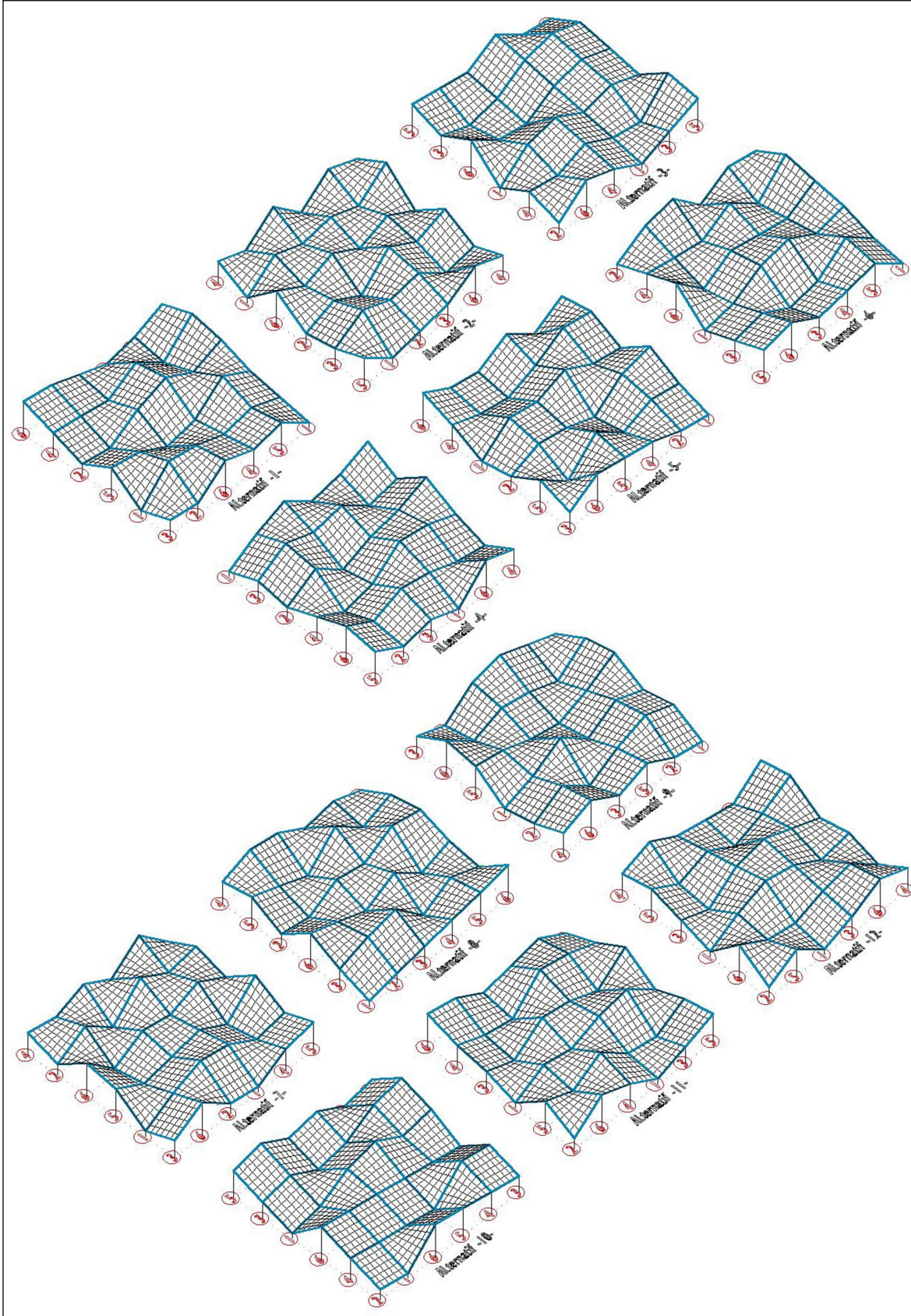
Bu şekilde ilk popülasyonun fenotipi oluşturulmuştur.



Şekil 3.17. Üretim-II ilk popülasyonun fenotipinin oluşturulması – adım 1



Şekil 3.18. Üretim-II ilk popülasyonun fenotipinin oluşturulması – adım 2

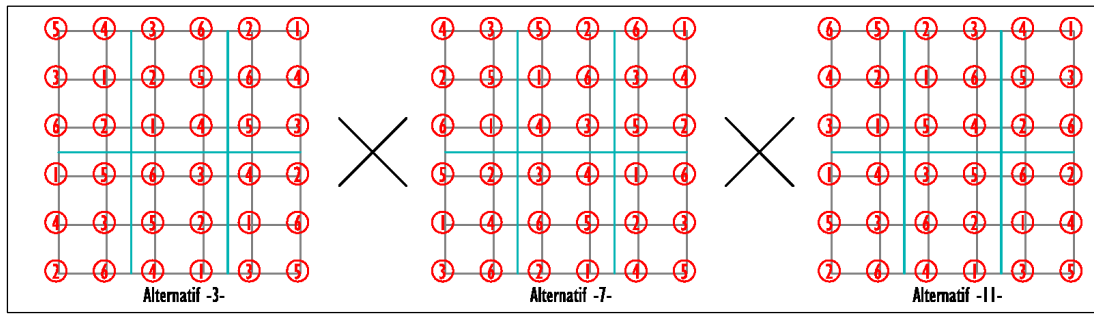


Şekil 3.19. Üretim-II ilk popülasyonun fenotipi

Seim ve aprazlama

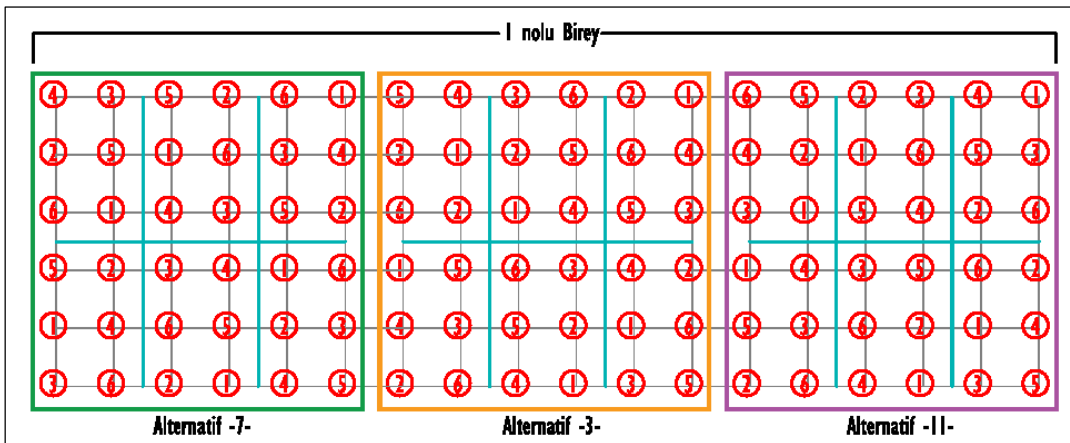
Bu adımda yine Üretim-I’de olduėu gibi kromozomlar akıřtırılarak aprazlamalar yapılmıřtır. Ancak SuGe Modelinde bir en iyileme mantıėı kurulmamıřtır. Bu nedenle aprazlanacak bireyler ve bu bireylerin aprazlamaya dahil edilecek kromozomları rastgele seilmektedir. Yani rastgele seilen bir bireyin rastgele seilen bazı kromozomları kaldırılıp yerine aprazlandıėı diėer bireyin ya da bireylerin kromozomları yerleřtirilmiřtir.

Farklı bir arada olma biimlerini gsterebilmek amacı ile üç bireyin ebeveyn seilmiřtir. Bunlar 3,7 ve 11 nolu alternatiflerdir. Yeni nesil bunlardan herhangi ikisinin aprazlanmasıyla oluřan seksüel bireylerden ve üçünün aprazlanması iře oluřan oklu birleřim bireylerden oluřturulmuřtur.

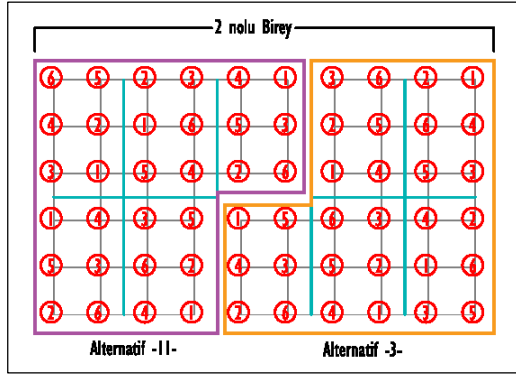


řekil 3.20. aprazlanacak bireyler

Bir alternatif olarak alternatiflerde hibir deėiřiklik yapılmadan sadece yanyana geliřleri ile yeni bir birey oluřturulmuřtur. Bu birey 18 kromozomlu bir oklu birleřim örneėidir.

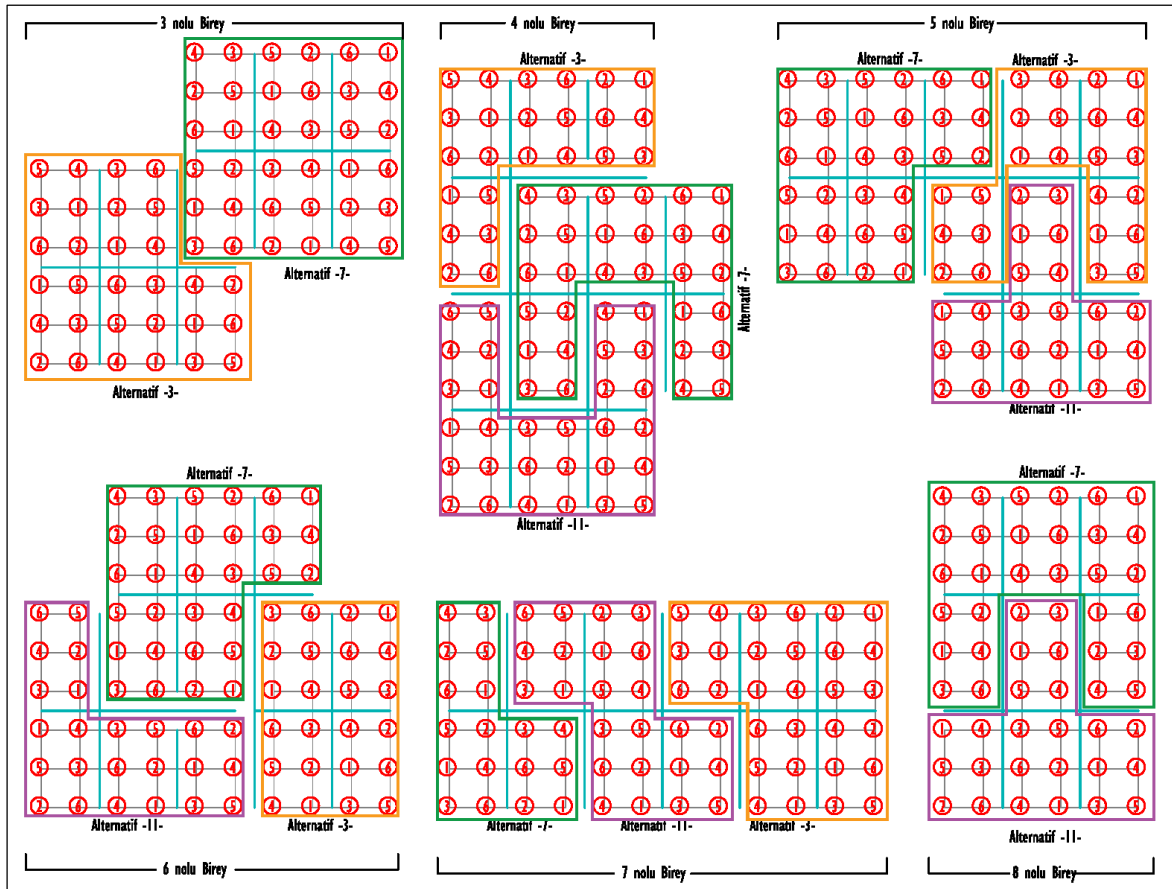


řekil 3.21. Üretim-II / 2. jenerasyon / 1 nolu birey



Şekil 3.22. Üretim-II / 2. jenerasyon / 2 nolu birey

Bir diğer çarazlama çeşidi kromozom değişimidir. Örnek olarak Alternatif -3- ve Alternatif -11- çaprazlanmıştır. Bu örnekte her bireyin bir kromozomu yerine diğer bireyin bir kromozomu yerleştirilmiştir. Bu şekilde 10 adet kromozomdan oluşan, seksüel bir birey elde edilmiştir. Bu şekilde çaprazlamalarla oluşturulan 2. Jenerasyon diğer üyeleri aşağıdaki gibidir. (Şekil 3.23)



Şekil 3.23. Üretim-II / 2. jenerasyonun diğer bireyleri

2. Jenerasyonun genotipi Şekil 3.21, Şekil 3.22 ve Şekil 3.23’de gösterilmiştir. Fenotip oluşturulurken daha önce belirlenen ve üç adımdan oluşan yöntemin ilk iki adımı aynı şekilde uygulanacaktır. Bunlar;

1. Adım: Her noktaya, üstünde yazan sayı değeri kadar br boyutunda z yönüne doğru bir çizgi çizilecektir.
2. Adım: Çizilen bu dikey çizgilerin üst noktası birleştirilecektir.

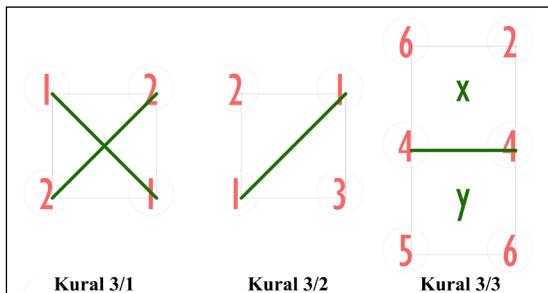
Ancak çaprazlamalar sonucu alternatiflerin bir araya gelmesi söz konusudur. Bu bir araya gelişlerden dolayı rakamların yerleşimi ilk popülasyon oluşturulurken belirlenen kurallardan farklılık göstermektedir. Örneğin ilk kurallara göre her satır ve sütunda her rakam tek bir kere kullanılırken özellikle birleşimler sonucu satır ve sütunlara 6’dan fazla sayı yerleştirildiğinden bir rakam birden fazla kullanılmaktadır. Bu değişimlerin fenotipe de yansması gerektiği düşüncesinden ve fenotipte daha çeşitli formlar elde etme isteğinden, düşey çizgilerin birleşiminden oluşan düzlemin kapatılması ile ilgili olan 3. adımda bazı değişiklikler yapılmıştır ve alanların kapatılması bazı şartlara bağlanmıştır.

3/1- Eğer bir karede çapraz köşelerde karşılıklı olarak aynı rakam varsa o kare fenotip oluşturulurken boş bırakılır.

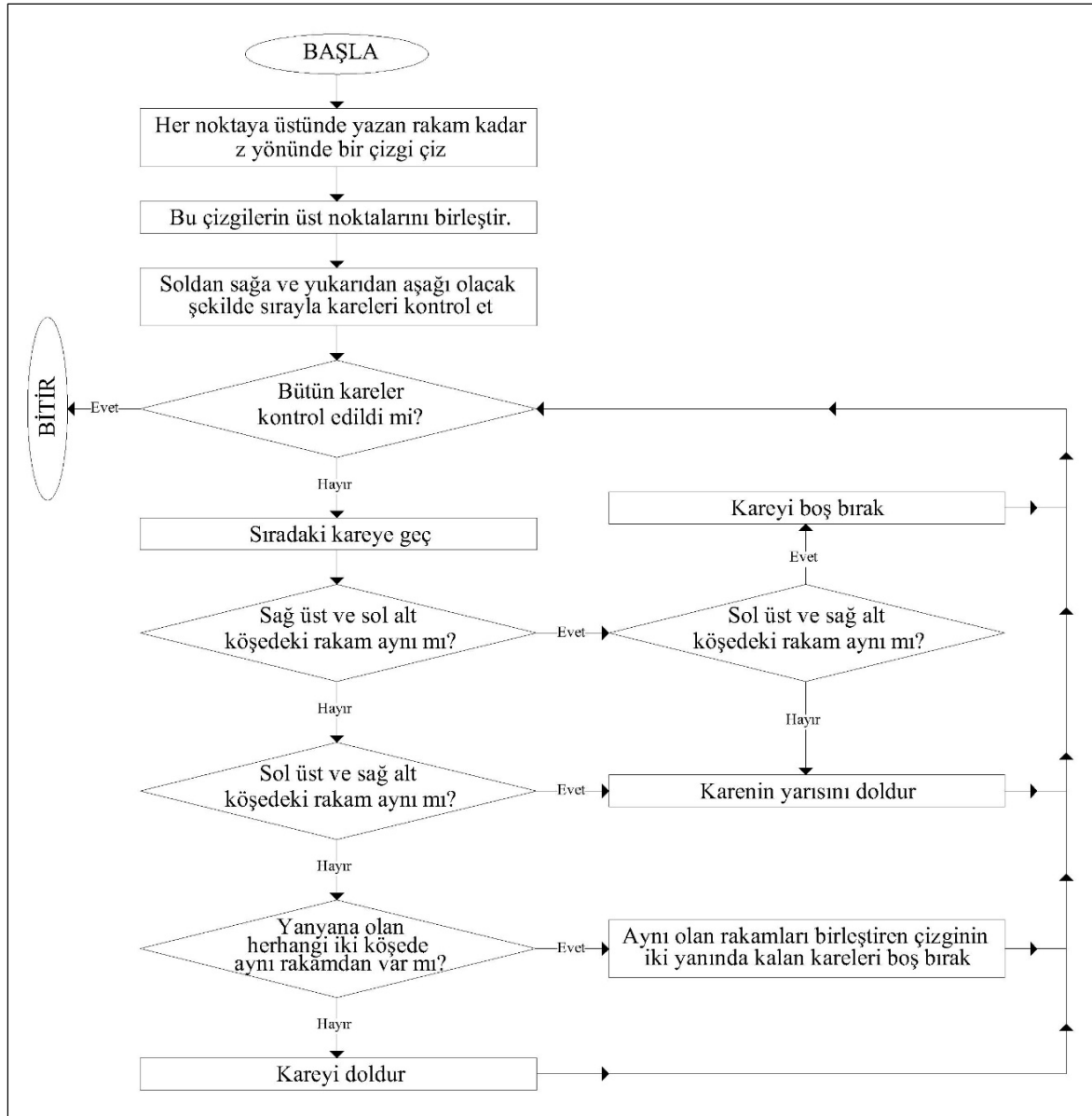
3/2- Eğer bir karede sadece tek bir çapraz köşede karşılıklı olarak aynı rakam varsa o rakamları birleştirilecek bir çizgi eklenerek kare iki bölüme ayrılır ve fenotip üç boyutlu hale getirilirken bu bölümlerden biri dolu biri bir boş bırakılır.

3/3- Eğer bir karede aynı rakam yan yana geliyorsa fenotipte o rakamları birleştiren çizginin her iki yanında kalan kareler boş bırakılır. Şekil 3.24’de 4 rakamları yan yana geldiğinden x ve y ile ifade edilen kareler modelde boş bırakılacaktır.

3/4- Bu üç kural dışında kalan bütün kareler modelde dolu karelerdir.

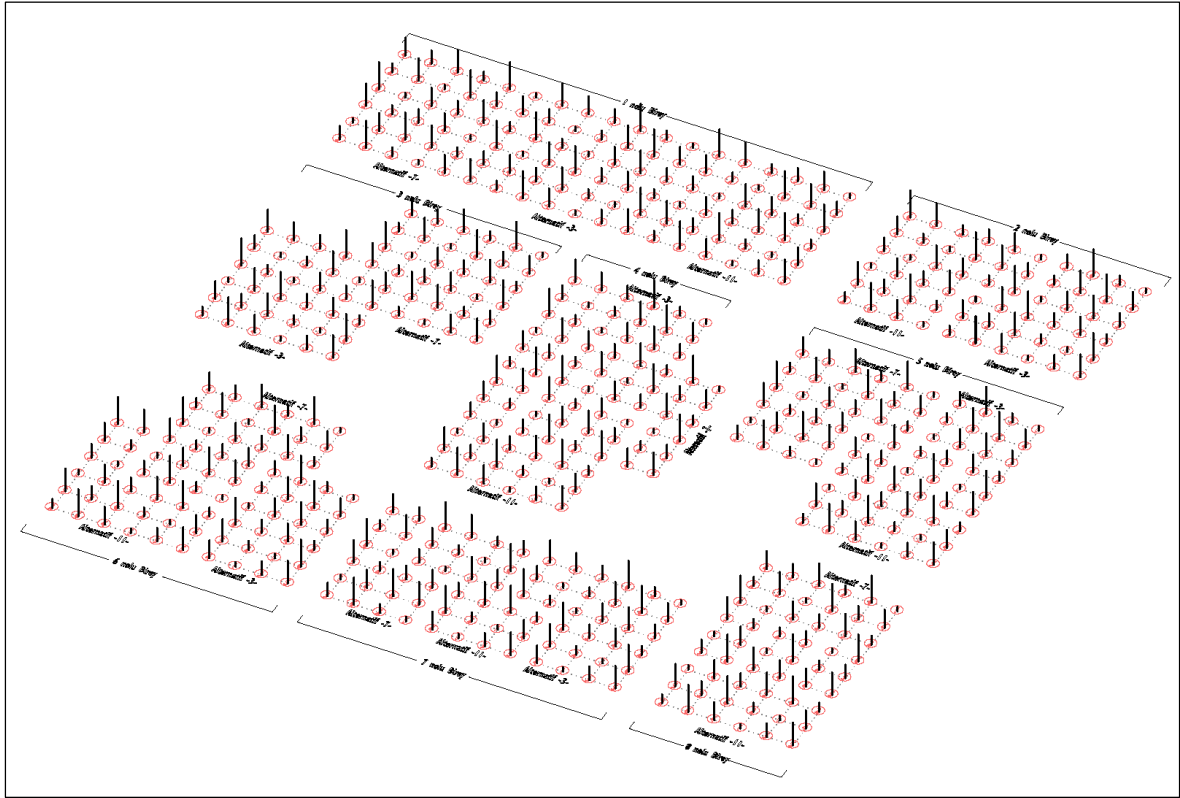


Şekil 3.24. Üretim-II / fenotip üretme kuralları

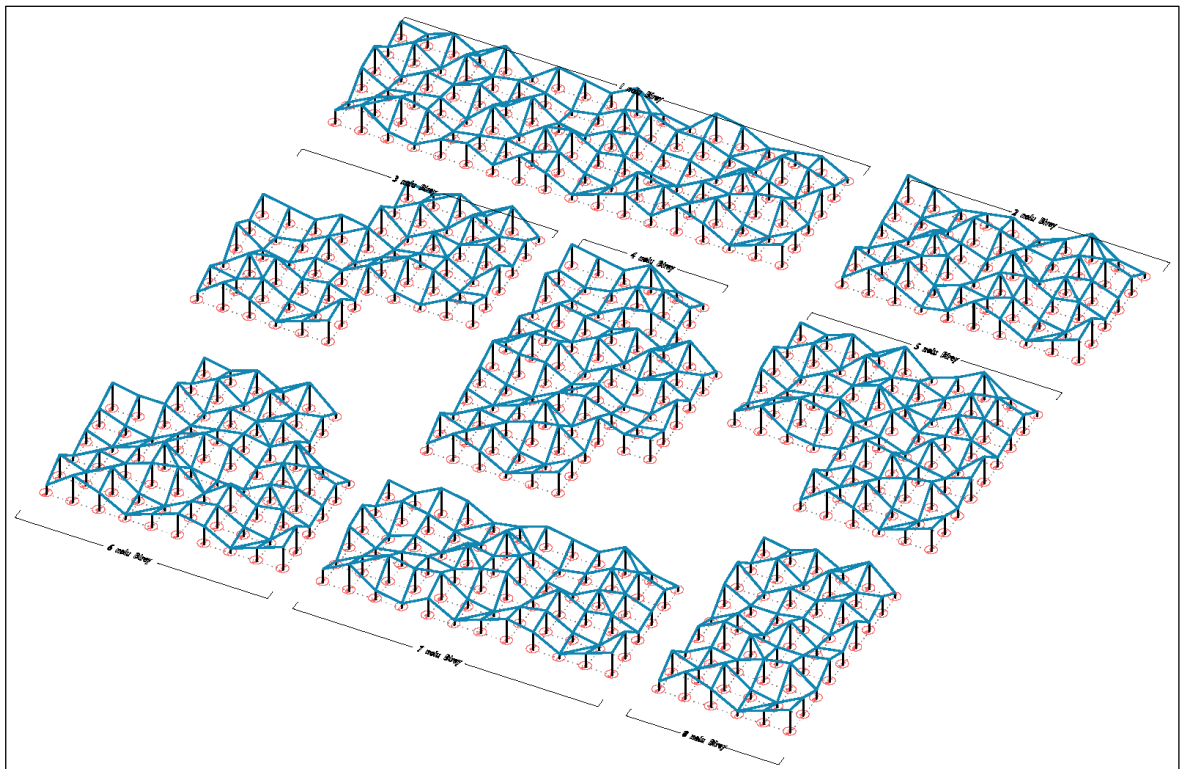


Şekil 3.25. Üretim-II / fenotip üretme algoritması

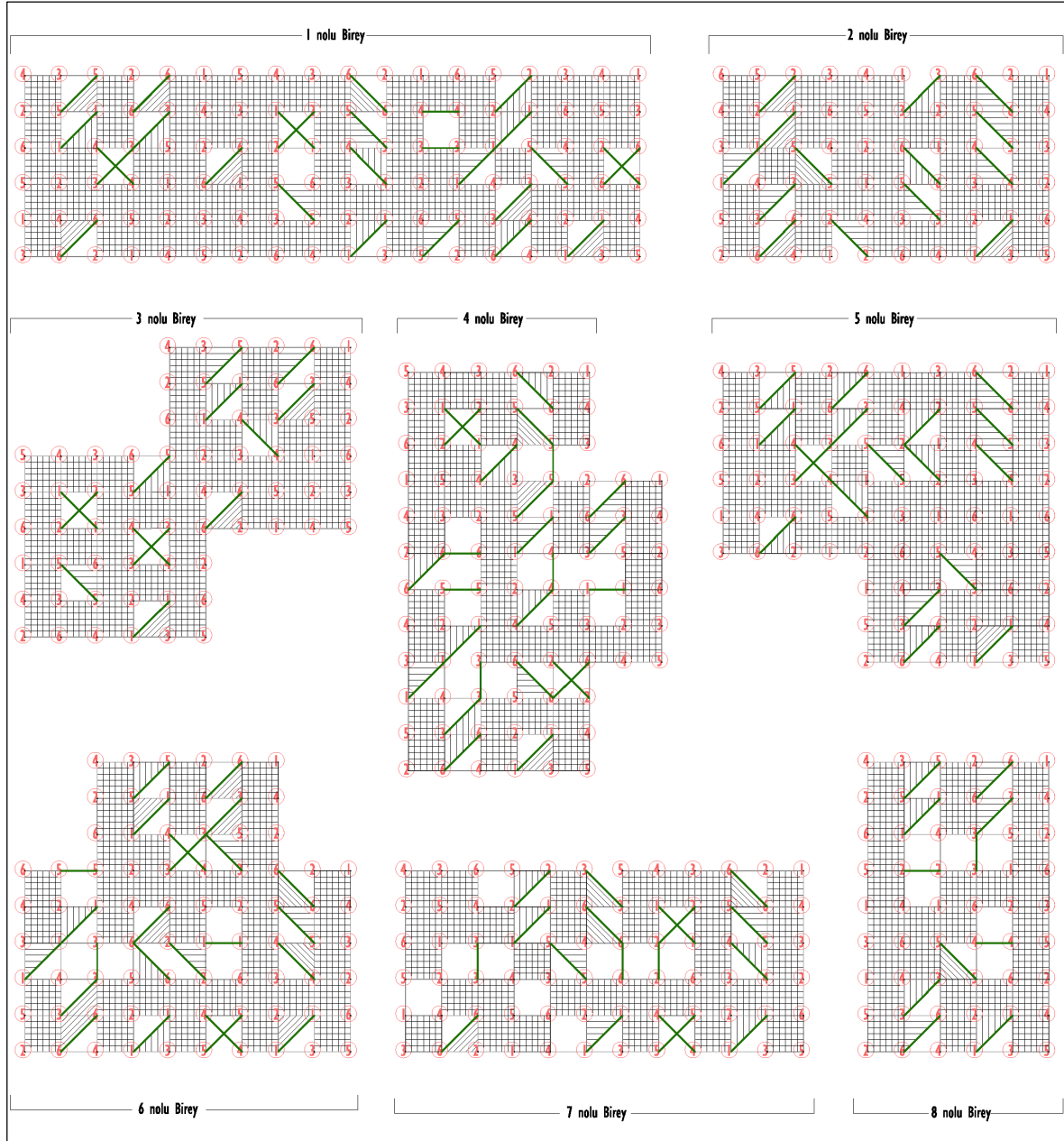
Oluşturulan bu algoritma takip edilerek 2. jenerasyonun fenotipi oluşturulmuş ve fenotipte dolu, yarı dolu ve boş olacak kareleri belirlenmiştir.



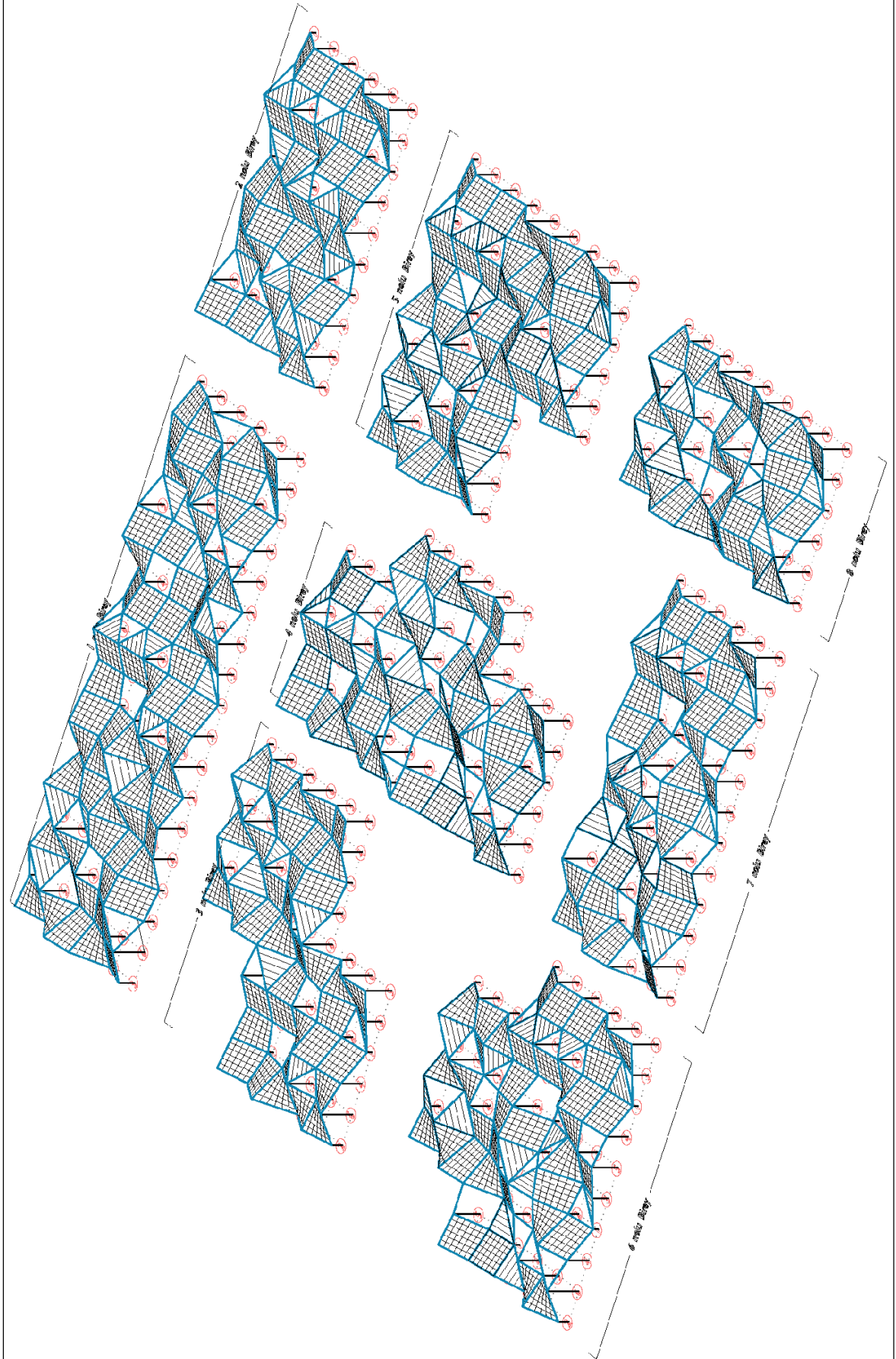
Şekil 3.26. Üretim-II / 2.jenerasyon fenotip oluşturma – Adım 1



Şekil 3.27. Üretim-II / 2.jenerasyon fenotip oluşturma – Adım 2



Şekil 3.28. Üretim-II / 2.jenerasyon dolu, yarı dolu ve boş olacak birimlerin belirlenmesi

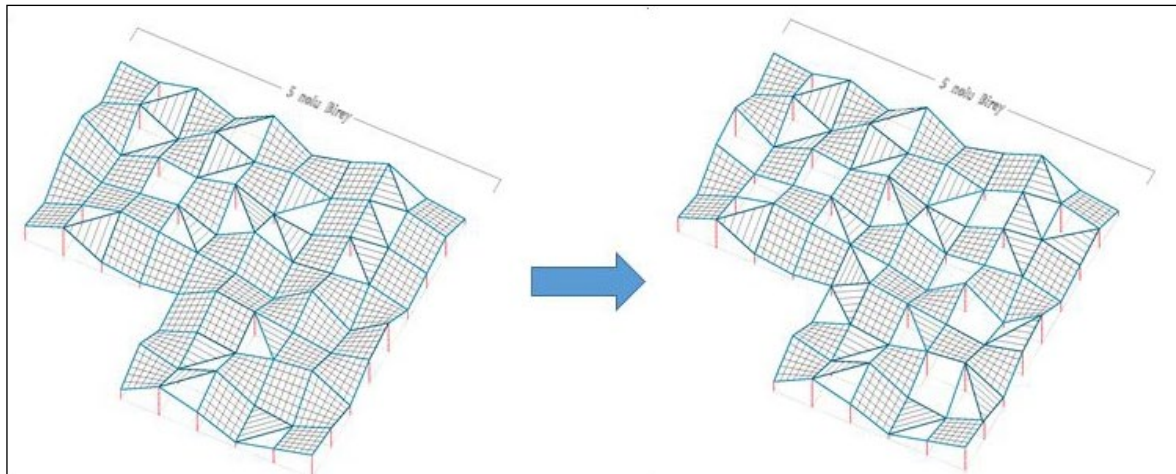


Şekil 3.29. Üretim-II / 2.jenerasyon fenotipi

Seim ve mutasyon

Bu adım artık popölasyondan bir ürünün seilip mutasyona uğratılacağı evredir. Mutasyon çözümleri geliřtirmek, çeřitlendirmek ve istenilen çözüme daha uygun hale getirmek için kullanılır. Bunun için belli bir mutasyon oranı belirlenerek bütün bireylere mutasyon uygulanıp, bunlar arasından bir seim yapılabileceğı gibi bir birey seilip sadece o bireye istenilen yöntem ve řekillerde mutasyon da uygulanabilir.

SuGe modelinde 5 nolu birey seilmiştir ve řeffaflığı üzerinden bir mutasyon uygulanmıştır. 5 nolu bireyde çoğunlukla dolu kareler bulunmaktadır. Biraz daha řeffaf olmasını sağlamak için rastgele kareler seilip bazıları boşaltılırken, bazıları ise yarı dolu hale getirilmiştir.



řekil 3.30. 5 nolu bireyin mutasyonu

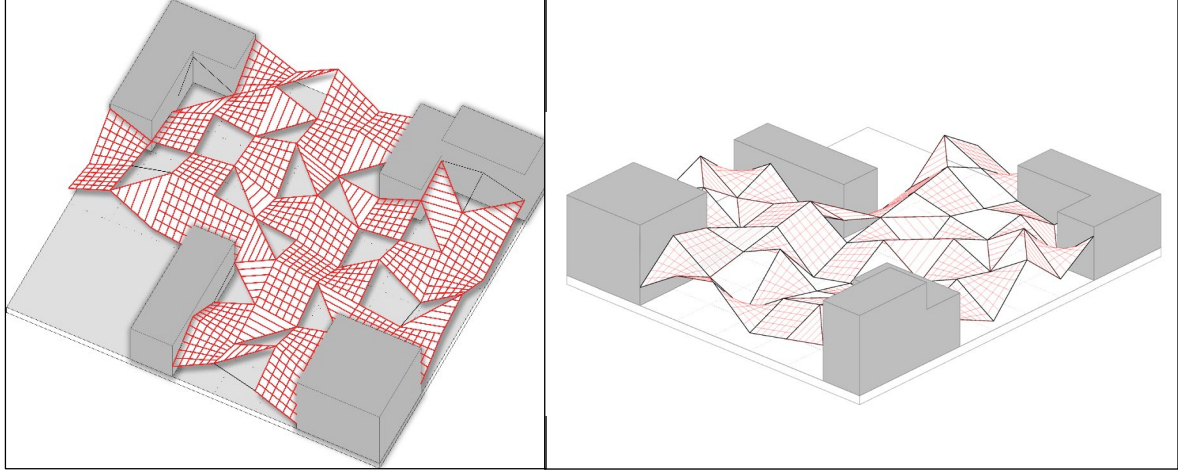
Bu işlem uygulanırken nasıl bir sonuç ürün elde edilmek istendiğı önemlidir. Buna yönelik olarak modele yeni kareler eklenebilir, modelin oranı değıřtirilebilir, bazı kareler birleřtirilebilir ya da bir modelin yüksekliğı ile ilgili değıřiklikler gibi başka mutasyonlar da uygulanabilir.

3.2.4. Değerlendirme

Bu bölüme kadar olan çalışmada Sudoku bulmacadan yola çıkarak biçim üretmede tasarımcıya yardımcı olacak bir model olan SuGe modelinin oluřturulma adımları açıklanmıştır. Bu bölümde ise bu modelle üretilecek biçimlerin mimari tasarımlarda ne řekilde kullanılabileceğı araştırılacaktır. Bu değıřlendirmeler üç bařlık altında toplanmıştır.

Kabuk tasarımı

Bu tasarımlar mimari de bir üt örtü ya da kabuk olacak şekilde kullanılabilirler. Örneğin Üretim-II evresinde üretilen 5 nolu birey bir bina kompleksinin arasında kalan meydanı örten bir kabuk olarak kullanılabilir.

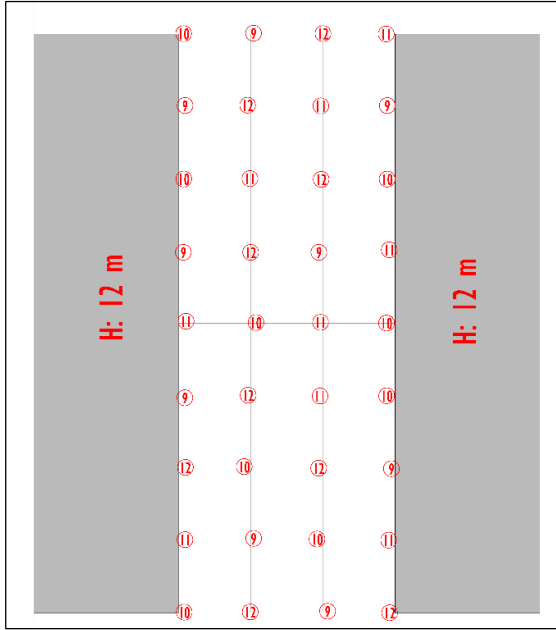


Şekil 3.31. SuGe modeli kabuk tasarımı

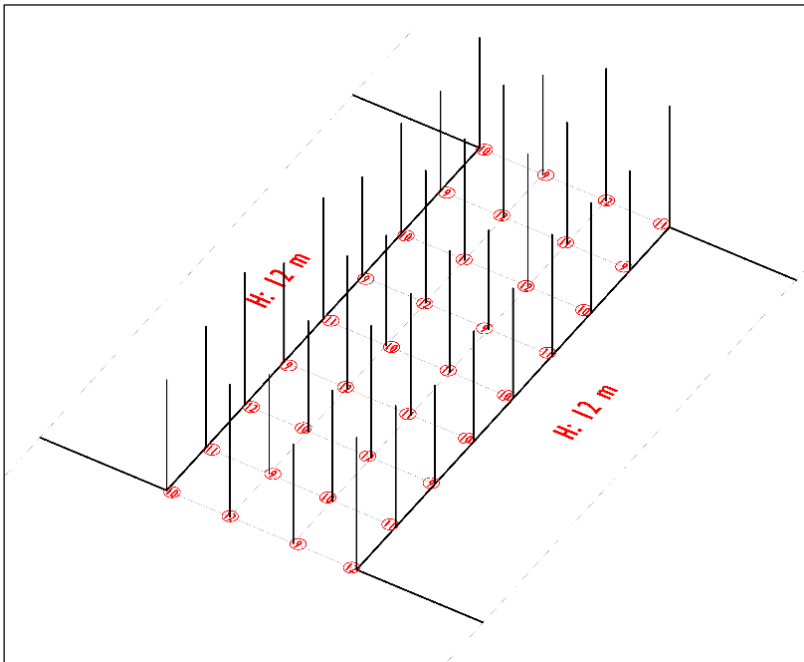
SuGe modeli kurgulanırken elde edilmek istenen sonuç ürün dikkate alınarak tasarlanmalıdır. Bu şekilde başlangıçta modelin uygulanacağı alan, kullanılacak rakamlar ve gerekli diğer kısıtlamalar ilk popülasyon evresinde belirlenir. Böylece süreç boyunca elde edilen çözümler, çözülmek istenen probleme yönelik özellikler barındıracaktır. Dolayısıyla elde edilen sonuç ürün de istenilen ihtiyaçları karşılayacaktır.

Örneğin Şekil 3.32’de görülen, yüksekliği 12 m olan iki binanın arasına bir kabuk tasarlamak için SuGe modeli hazırlanacak olsa, modelin kurgulanacağı başlangıç alanı olarak bu iki binanın arasındaki alan dikkate alınmalıdır. Bu örnek modeli görebilmek amacıyla, iki bina arasındaki alan $3 \times 8 = 24$ kareden oluşan bir gridal sisteme ayrılmıştır. Kabuğun yaklaşık olarak binaların üst kısmına yakın bir yerde olması istendiğinden bu modelde 9,10,11 ve 12 rakamları kullanılmıştır. Bu rakamlar her biri eşit sayıda kullanılacak şekilde rastgele olarak bu gridal sistem üzerine yerleştirilmiştir. Daha sonra SuGe modelindeki üretim adımları takip edilebileceği gibi ilk elde edilen genotipte kullanılabilir. Bunun seçimi, elde edilmek istenen ürünle ve tasarımcının istekleri ile bağlantılıdır. Genotip belirlendikten sonraki adım fenotip oluşturma algoritması kullanılarak önce her kesişim noktasına üstünde yazan rakam

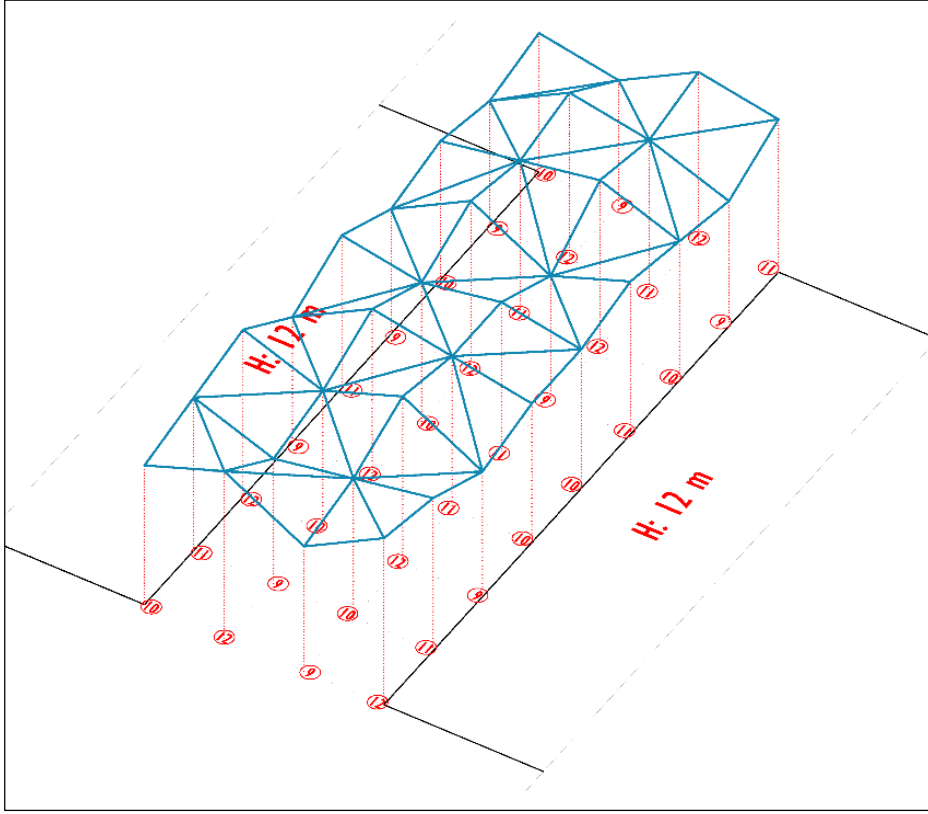
değerinde dikey bir çizgi çizilmiştir. (Şekil 3.33) Bu çizgilerin üst noktaları birleştirilmiştir. Ancak 3. Adımda değişiklik yapılarak bütün karelere yazan rakamlardan bağımsız olarak bir köşegen eklenmiştir. (Şekil 3.34) Daha sonra her hücrenin yarısı dolu yarısı boş olacak şekilde model üç boyutlu hale getirilmiştir ve böylece iki binayı birbirine bağlayan bir kabuk elde edilmiştir. (Şekil 3.35)



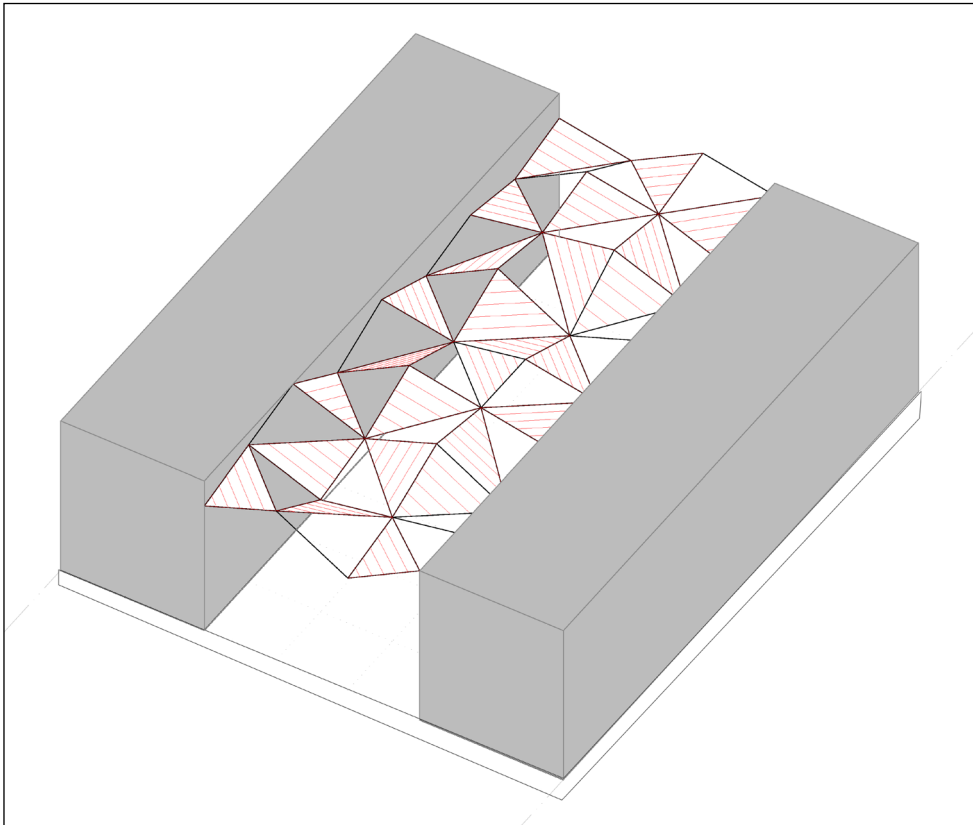
Şekil 3.32. Kabuk tasarım örneği genotipi



Şekil 3.33. Kabuk tasarımı fenotip üretimi-adım 1



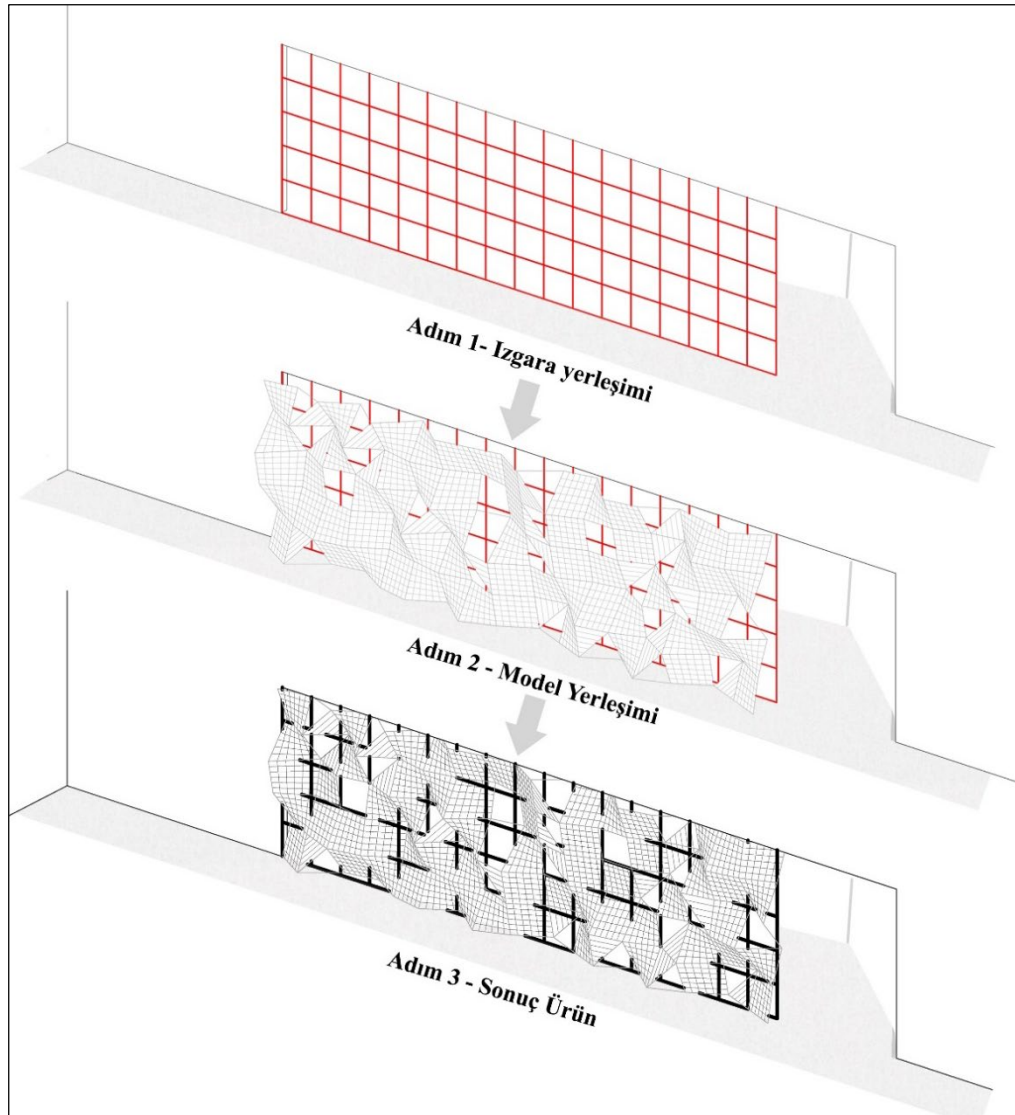
Şekil 3.34. Kabuk tasarımı fenotip üretimi- adım 2



Şekil 3.35. Kabuk tasarımı örneği

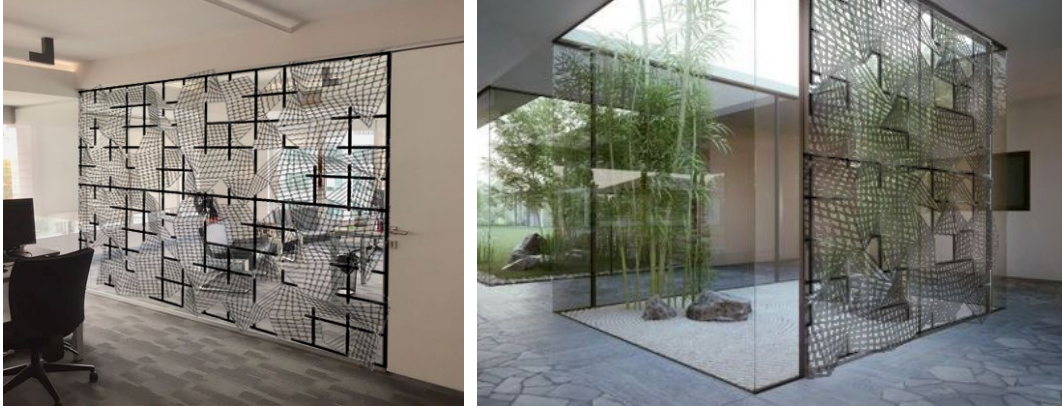
Yüzey tasarımı

SuGe modeli ile yüzeyler de tasarlanabilir. Örnek olarak bir seperatör tasarlanmıştır. Bu tasarım için son popülasyona ait 1 nolu birey kullanılmıştır. Öncelikle bu bireye ait genotipi oluşturmakta kullanılan, alan sınırlarını belirten ızgara korunmuştur. Bu ızgara seperatörün yapılacağı bölümde kemik aksı oluşturacak şekilde yerleştirilmiştir. Daha sonra 1 nolu birey bu ızgaraya yerleştirilmiştir. Ancak bu durumda seperatör tek bir yöne doğru gelişmektedir. Model oluşturulurken 1’den 6’ya kadar rakamlar kullanıldığından modelin z yönündeki alanı 0-6 aralığındadır. Seperatörün ayrılan iki bölüme doğru da gelişmesini sağlamak için model ızgaranın üzerinde yarısı bir bölümde, diğer yarısı öbür bölümde kalacak şekilde 3 br. kaydırılmıştır. (Şekil 3.36)



Şekil 3.36. Yüzey tasarımı aşamaları

Bu çalışmada hazır olan model bir seperatör olacak şekilde yeniden düzenlenmiştir. Ancak modele başlarken bir yüzey tasarlama fikriyle başlanıp, buna göre model özelleştirilirse, kısıtlamalar ve kurallar buna göre belirlenirse çok daha farklı ve uygun tasarımlar elde edilecektir.



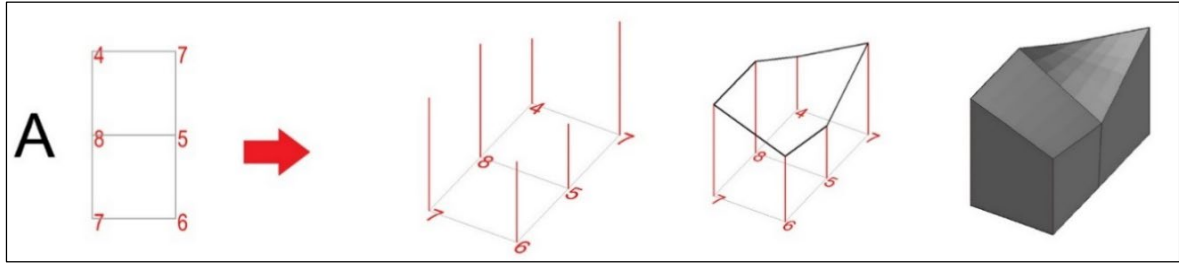
Şekil 3.37. Yüzey tasarımının farklı alanlardaki uygulaması

Kütle tasarımı

Çalışmanın bu noktasına kadar model hep tek ızgara ve düzlem üzerinde tanımlanmıştır. Bu başlık altında bu modelin mantığından yola çıkılarak bir kütle oluşturulup oluşturulamayacağı araştırılmıştır. Bunun için yine kendine özel bir algoritma yazılmalıdır. Öncelikle kütlenin oturacağı alan belirlenmiş ve bu alan modelin sınırları olarak kabul edilmiştir. Daha sonra kütlenin yüksekliğini de ifade edebilmesi için 4, 5, 6, 7 ve 8 rakamlarının kullanılmasına karar verilmiştir. Örneğin A ile ifade edilen tasarımdan yola çıkarak ifade edecek olursak; (Şekil 3.38)

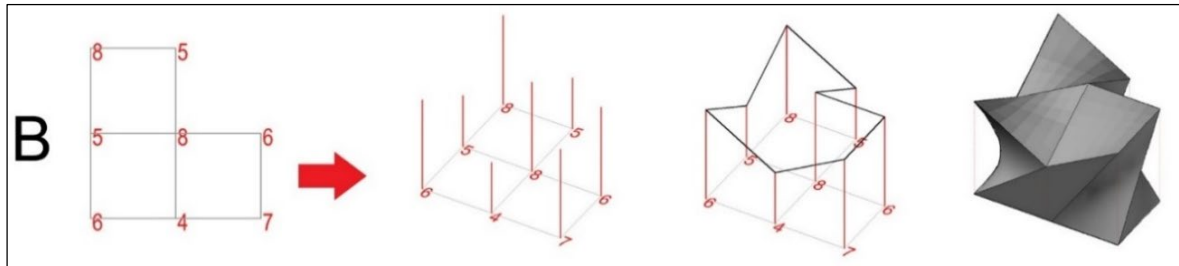
1. 2 adet hücreden oluşan taban alanı oluşturulmuştur.
2. Taban alanı üzerinde bulunan 6 adet noktaya belirlenen rakamlar rastgele yerleştirilmiştir.
3. Her noktaya üzerinde yazan değer boyutunda z yönüne doğru bir çizgi çekilmiştir.
4. Bu çizgilerin üst noktaları birleştirilerek tavan görevi görecektir olan düzlem oluşturulmuştur.
5. Tavan düzlemi ile taban düzlemini oluşturan çokgenler arasında kalan bölge kapatılmıştır.

A örneğinde taban ve tavan düzlemleri oluşturulduktan sonra, bu düzlemler doğrudan birleştirilmiştir.



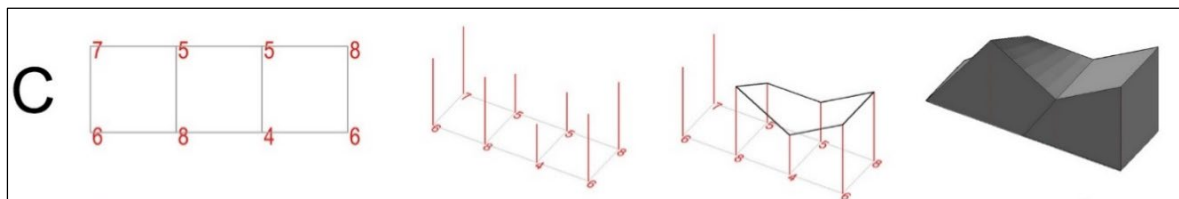
Şekil 3.38. Kütle tasarımı – A örneği

B örneğinde ise üç hücreden oluşan bir taban alanı belirlenmiş ve A örneğindeki adımlar takip edilmiştir. Taban ve tavan düzlemi oluşturuluna kadar olan adımlar aynen tekrar edilmiştir. Ancak bu sefer z yönünde çizilen çizgiler kütlenin kenarları olarak kullanılmamıştır. Bunun yerine tabandaki bir nokta, tavanda kendi doğrultusunda olmayan başka bir nokta ile birleştirilerek kütlenin kenarları oluşturulmuştur. (Şekil 3.39) Bu sayede durağan formlar yerine daha esnek ve değişken formlar üretilebilecektir. Bununla birlikte aynı sınırlar ve aynı rakamlar kullanılsa bile farklı biçimsel formlar elde edilebilecektir.



Şekil 3.39. Kütle tasarımı – B örneği

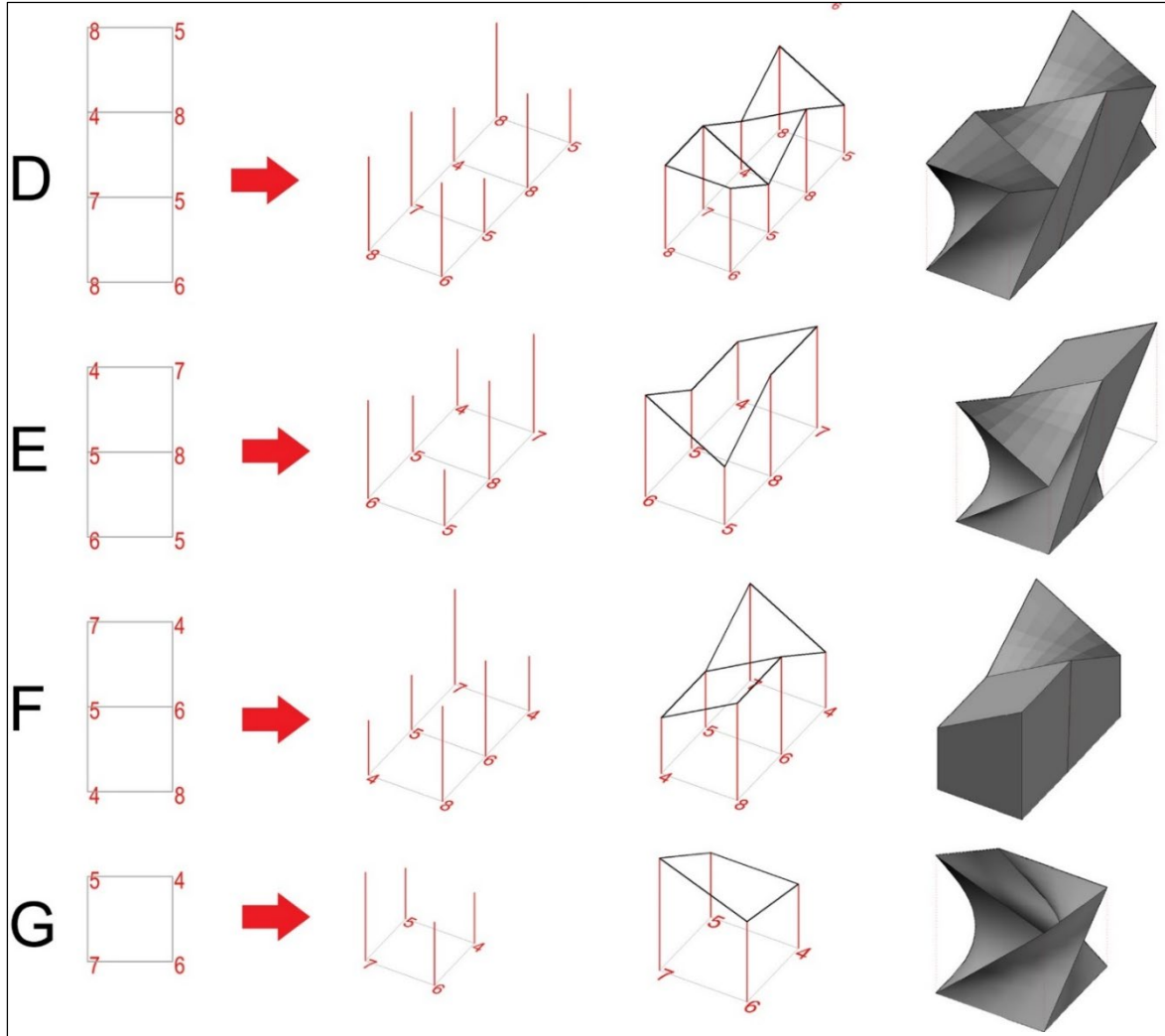
C örneği ise yine üç hücreden oluşan bir taban alanına sahiptir ancak A ve B örneklerindeki iki ayrı yöntemin beraber kullanılması ile oluşturulmuştur. Bu nedenle tavan düzlemi 2 adet hücreden oluşmaktadır. Dolayısıyla taban ve tavan düzlemlerinin iz düşümü aynı değildir. (Şekil 3.40)



Şekil 3.40. Kütle tasarımı – C örneği

Bu yöntem mimari bir yapının veya elemanın formunu oluşturmak için kullanılabileceği gibi bir dekor ürününün ya da bir eşyanın formunu oluşturmak için de kullanılabilir. Önemli olan modelin ne için hazırlandığına en başta karar vermek, istenen çözüme ulaşmak için gerekli olan kural ve kısıtlamaları doğru belirlemek ve probleme en uygun cevabı verecek özgün bir model oluşturmaktır. Böylece sapmalar olmadan doğru ve optimal çözüme ulaşılabilir. Bununla birlikte sürpriz formlar elde etmede oldukça başarılı bir modeldir.

Şekil 3.41’de anlatılan üç yöntemle ve bunların bir arada kullanılması ile oluşturulabilecek bazı kütle çalışmaları gösterilmiştir.



Şekil 3.41. Kütle tasarımı – D, E, F ve G örnekleri

4. SONUÇ

Genetik algoritmalar eldeki problemi doğadaki evrim mantığıyla ele alırlar. Doğadaki ve canlılardaki evrimsel süreci taklit eden operatörleri kullanarak mevcut çözümler içinde arama yaparlar. Geçmişten günümüze doğayı taklit eden birçok tasarımlar yapılmıştır. Ancak Darwin'in evrim kuramı oldukça yeni yaklaşımların doğmasını sağlamıştır. Bu evrimsel bilgi işleme teknikleri sayesinde geleneksel yöntemlerle çözümü zor olan karmaşık problemler hızlı ve etkili bir şekilde çözülebilmektedir. Bu problemler genellikle çözüm alanı çok geniş olan ve bütün alternatif çözümlerin incelenmesinin mümkün olmadığı problemlerdir. GA'lar her zaman en iyi çözümü bulmayı vadetmese de, bu tarz problemlerde kısa sürelerde makul sonuçlar elde etmektedir.

Tasarım problemleri de doğası gereği karmaşık, disiplinler arası ve yaratıcılık gerektiren problemlerdir. Her ne kadar son dönemlerde bilgisayar teknolojilerinin gelişmesi ile birlikte mimari tasarımların hazırlanması ve sunumu kolaylaşmış olsa bile, bir tasarım fikrinin oluşumu ve gelişimi tasarımcının beyninde gerçekleşen bir muammadır. Ayrıca oluşturulan bir tasarım ne kadar iyi olursa olsun, o problem için en iyi çözüm olup olmadığını bilmek mümkün değildir. Çünkü oluşturulabilecek sonsuz tasarım alternatifinin hepsini bulmak ve uygunluğunu kontrol etmek mümkün değildir.

Bu çalışmada bütün bu problemler dikkate alınarak tasarımcıya tasarımın ilk evrelerinde yardımcı olacak SuGe modelini geliştirmek amaçlanmıştır. Bu model oluşturulurken zaten var olan sudoku bulmacanın üretim mantığı dikkate alınmıştır. Genetik Algoritma mantığı sayesinde tasarımcı tasarıma başlarken sonuç olarak elde etmek istediği mimari tasarımın gereklerini en baştan dikkate alabilecektir. Sonuç ürünün gerekleri ve getirdiği kısıtlamalar sayesinde çözüm uzayı daraltılabilecek ve istenen sonuca en uygun çözümlerle süreç yürütülecektir. Bu sayede sonuç ürüne gelindiğinde kullanışsız, estetik olamayan, işlevsel ya da biçimsel sıkıntıları olan sürpriz tasarımlarla karşılaşma riski ortadan kalkacaktır. Ayrıca GA'nın paralel çalışma mantığı sayesinde ulaşılan çözüm tasarımcıyı tatmin etmese bile, kolaylıkla gerideki bir adıma dönerek veya paralel başka bir aramaya yönelerek istenilen sonuca ulaşılan kadar süreç tekrarlanabilecektir.

Ayrıca SuGe modelinde formlar sudoku bulmacanın üretim adımlarını tekrarlayan bir mantıkla üretilmektedir. Bu sayede tasarımcıya beklenmedik ve yenilikçi formlar üretme konusunda yardımcı olmaktadır.

Bu modelde bir diğer etken kullanıcı faktörüdür. GA mantığı ve sudoku kurallarına ek olarak kullanıcının istek ve kısıtlamalarını da dikkate alan bir model oluşturulmuştur. Her ne kadar bu tarz üretken yaklaşımlar ve diğer destekleyici öğeler tasarım için büyük kolaylık olsa da tasarımcı etkisi olmadığında, tasarım eylemi bir makinanın kaydedilmiş adımları tekrarlamasından farklı olmayacaktır. Bu şekilde ise farklı ve yaratıcı tasarımlar beklemek mümkün değildir. Kullanıcı –ki bu çalışmada kullanıcı tasarımcıdır- SuGe modelinde oldukça etkin bir rol oynamaktadır. Daha süreç başlatılırken tasarımcı girdileri dikkate alınır ve bütün süreç tasarımcı kontrolünde devam eder. Sürecin ne kadar devam edeceği, hangi adımda durdurulacağı, mutasyon faktörü, sonuç ürünün seçimi vb. gibi birçok adım tamamen kullanıcı geri bildirimleriyle yönetilmektedir. Dolayısıyla tasarımcı burada hem tasarımın ana fikrini oluşturan, hem de süreci yöneten kişidir.

Dolayısıyla SuGe modeli sudoku kurallarını, tasarımcının beynindeki bir kara kutuda işleyen ve çoğunlukla bir muamma olan tasarım sürecini ve doğanın döngüsünü bir araya getiren mimarlık, matematik, geometri ve biyoloji gibi birçok bilimi içinde barındıran, farklı, yenilikçi, beklenmedik mimari formlar üretmeyi amaçlayan bir genetik algoritma modelidir. Bu çalışmada öncelikle genetik algoritma mantığı tanıtılmış ve mimaride kullanımları üzerine örnekler incelenmiştir. Daha sonra evrim süreci ile bir sudoku bulmacanın nasıl üretilebileceği ele alınmış ve bir kurallar zinciri oluşturulmuştur. Sonuçta ise bütün bu çalışmalardan yola çıkılarak tasarımcıya form üretmede yardımcı olacak olan SuGe modelinin nasıl oluşturulacağı, çalışma mantığı, mimari tasarımlarda kullanımı ve tasarım sürecine katkısı ele alınmıştır.

Ancak bu çalışmada öncelikli olan genetik algoritmanın mimari tasarımlarda nasıl kullanılabileceğini anlamak ve örnek bir modelin nasıl oluşturulacağını ve nasıl işleyeceğini göstermektir. Bu nedenle modelin başından değerlendirme aşamasına kadar modelin kurgusu temel bileşenler üzerinden anlatılmıştır. Tek bir problem üzerinden bir çalışma yapmak yerine genel bir soruya cevap aranmış ve bazı adımlarda alternatif eylemleri ve sonuçları görmek amacıyla, farklı kararlar verilerek model birkaç yönde geliştirilmiştir. Böylece ileride bu model kullanılarak bir biçim oluşturulmak istendiğinde, tasarımcının

modelin sağladığı imkânları daha iyi anlayabilmesi ve tam olarak elde etmek istediği biçime göre uyarlayabileceği bilgiye sahip olması amaçlanmıştır.

Dolayısıyla elde edilen SuGe modeli probleme özel kurgulanarak mimari tasarımın birçok alanında farklı şekillerde kullanılabilir. Bu sayede mimari tasarım ve görselleştirme süreçleri hızlanacak ve kolaylaşacaktır. Örneğin arkeolojik sit alanları ve öğren yerlerinde kullanılacak örtülerin tasarımı bu model ile yapılabilir. Geniş açıklıklı mekanların üst örtüsü yine SuGe modelinin mantığı kullanılarak tasarlanabilir. Ayrıca bu örtülerin taşıyıcı sistemi tasarlanırken, temel altlık olarak belirlenen grid baz alınabilir. Bu sayede inşa edilebilirlik üzerine de bir optimizasyon sağlanması mümkündür. Bununla birlikte model kent ölçeğinde ele alınarak, büyük ölçekli kentsel tasarımlar yapılırken kullanılabilir. Temel grid sistemimiz kentin gridal sistemi olarak ele alınabilir ve gridal dokuya sahip bir kent parçası bu model sayesinde kurgulanabilir. Böylece en başta verilen kararlar tasarıma uygun, daha kontrollü ve sonuç odaklı olacağından en uygun tasarımlara daha hızlı bir şekilde ulaşılabilecektir.

Bununla birlikte ileride modelin bilgisayar ortamına aktarılması ve zaman zaman başka yöntemlerle hibrid olarak kullanılması da öngörülmektedir. Böylece daha hızlı ve etkin arama yapılabilecektir. Ayrıca bazı diğer sistemlerle entegre edilerek arazi sınırları, topoğrafya, iklim rüzgar vb. faktörleri de dikkate alan bir model haline getirilebilir. Hatta diğer disiplinlerle ilişkisi üzerine bir düzenleme de eklenerek tasarımların gerçek hayatta uygulanabilirliğine katkı sağlanabilir. Zaten genetik algoritma mantığı itibari ile hibrid kullanıma oldukça uygundur.

Ayrıca tasarımların biçimsel çözümleri ile birlikte işlevsel çözümleri de çok önemlidir. Model geliştirilerek biçim ve işlevin beraber değerlendirildiği bir hale getirilebilir. Böylece biçimsel çözümlerle birlikte işlevsel kaygıları da göz ardı etmeyen daha geniş kapsamlı ve daha kullanışlı bir model haline gelmesi de öngörülmektedir.

Bütün bunlarla birlikte modelin parametrik tasarım mantığı sayesinde tasarım eğitiminde kullanılabileceği de öngörülmektedir. Yeni tasarımcı adaylarının biçim üretme mantığını anlaması, tasarıma dair verdiği kararların sonuçlarını görebilmesi, tasarım sürecinin her adımının birbirini nasıl etkilediğini kavrayabilmesi için oldukça uygun bir yöntemdir. Bu model sayesinde değişen her bir kararın biçime nasıl yansıdığını, her bir parametrenin mimari formu ve tasarım sürecini nasıl etkilediğini, tasarımcı ve tasarım eylemi arasındaki

ilişkiyi neden sonuç bağlamında kolaylıkla anlamlandırabileceklerdir. Ayrıca model sayesinde hızlı bir şekilde deneysel formlar üretebilecek, benzer parametrelere bağlı birçok alternatif biçimi bir arada görebilecek ve dolayısıyla birçok karşılaştırma yaparak hangi formun neden daha iyi olduğuna karar verebileceklerdir. Böylece çoğunlukla ilk defa tasarım eğitimi almaya başlayan ve bundan kaynaklı olarak tasarım eylemini tam olarak anlamlandırmaları zor olan bu öğrenciler için temel tasarım eğitiminde yaratıcı mimari tasarımı kavramaları kolaylaşacaktır.

KAYNAKLAR

- Ağalday, M. F. (2018). *Genetik Algoritma İle Merkezi Sınavlarda Tek ve Çok Boyutlu Yakınlığa Göre En İyi Oturum Planının Oluşturulması*, Yüksek Lisans Tezi, Fatih Sultan Mehmet Vakıf Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü, İstanbul, 41-42.
- Akpınar, F. (2009). *Yerleştirme Rotalama Problemi İçin Bir Genetik Algoritma*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 15-22.
- Aksoy, Y. (2016). *Sürdürülebilir Toplu Konut Yerleşmesi Tasarımı İçin Pareto Genetik Algoritmaya Dayalı Bir Model Önerisi: SSPM*, Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 11.
- Badem, Y. (2007). *Genetik Algoritmaların Yaratıcı Mimari Tasarımda Kullanımı*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 114-127.
- Bentley, P. (1999). *An Introduction to Evolutionary Design by Computers*. San Francisco: Morgan Kaufmann Publishers, 7-10.
- Bolat, B., Erol, K. O. ve İmrak, C. E. (2004). Genetic Algorithms in Engineering Applications and the Function of Operators. *Journal of Engineering and Natural Sciences Mühendislik ve Fen Bilimleri Dergisi*, 4, 264-271.
- Cheng, R., Gen, M. ve Tsujimura, Y. (1996). A Tutorial Survey Of Job-shop Scheduling Problems Using Genetic Algorithms—I. Representation. *Computers & Industrial Engineering*, 30(4), 983-997.
- Cheng, R., Gen, M. ve Tsujimura, Y. (1999). A Tutorial Survey Of Job-shop Scheduling Problems Using Genetic Algorithms, Part II: Hybrid Genetic Search Strategies. *Computers & Industrial Engineering*, 36(2), 343-364.
- Chouchoulas, O. ve Day, A. (2007). Design Exploration Using a Shape Grammar With a Genetic Algorithm. *Open House International*, 32(2), 26-35.
- Cinan, S. (2010). Sudoku Bulmacasında Zihinsel Adımlar: Hariç Tutma ve Dahil Tutma Taktikleri. *Türk Psikoloji Dergisi*, 25(66), 42-56.
- Coşkun, A. (2006). *Genetik Algoritma Kullanılarak Kimyasal Maddelerin Deriden Geçiş Katsayılarının ve Molekül Yapılarının Bulunması*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 39.
- Çakar, K. (2009). *Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu*, Doktora Tezi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü, Erzurum, 43-71.
- Çalışır, K. (2015). *Olimpik Havuz Plan Şeması Tasarımında Genetik Algoritmaya Dayalı Bir Model*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 6-7.
- Çivril, H. (2009). *Hemşire Çizelgeleme Problemlerinin Genetik Algoritma İle Çözümü*, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta.

- Dil, Y. (2015). *Yazılım Projelerinin Optimizasyon Problemi Olarak İncelenmesi ve Genetik Algoritma İle Çözümü*, Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı, İstanbul, 11-12.
- Dursun, P. (2009). *Zaman Pencere Arayış Problemi'nin Genetik Algoritma İle Modellenmesi*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Engin, O. (2001). *Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu*, Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 17-18.
- Engin, O. ve Fırlı, A. (2002). Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu. *İstanbul Teknik Üniversitesi Dergisi*, 1(1), 1-6.
- Erbaş Korur, Z. N. (2012). *Genetik Mimarlık Kavramının Günümüz Mimarlık Anlayışları İçindeki Yeri*, Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 51-53, 61.
- Erbaş Korur, Z. N. ve Dülgeroğlu Yüksel, Y. (2013). Genetik Biliminin Mimarlık Disiplinleri Üzerindeki Etkisi. *Beykent University Journal Of Science And Engineering*, 6(1), 11-25.
- Erdoğan, E. ve Sorguç, A. (2011). Hesaplamalı Modeller Aracılığıyla Mimari ve Doğal Biçim Türetim İlkelerini İlişkilendirmek. *METU Journal of the Faculty of Architecture*, 269-281.
- Erdoğan, Y. S. (2007). *Genetik Algoritmalar Kullanılarak Sonlu Eleman Güncellemesi Yöntemiyle Hasar Tespiti Ve Parametre Belirlenmesi*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 12-17.
- Frazer, J. (1995). *An Evolutionary Architecture*. London: Architectural Association, 9,58.
- Geem, Z. W. (2007). Harmony Search Algorithm for Solving Sudoku. *KES: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, s. 371-378.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Alabama: Addison Wesley Publishing Company, 7,10,92.
- Gökay Emel, G. ve Taşkın, Ç. (2002). Genetik Algoritmalar ve Uygulama Alanları. *Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152.
- Gökay Emel, G. ve Taşkın, Ç. (2009). *Sayısal Yöntemlerle Genetik Algoritmalar*. Bursa: Alfa Aktüel Yayınları, 31-96.
- Güleç, D. (2014). *Mimari Tasarım Alanında Kullanıcı Erişilebilirliğinin Genetik Algoritma İle Optimizasyonu - Ada: Sağlık Kampüsü Uygulaması*, Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 40-41.
- Güngör, Ö. (2010). *Genetik Algoritmaya Dayalı Kitleli Bireyselleştirme Amaçlı Konut Tasarım Modeli*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 59-72.

- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 2.
- Horat, B. (2014). *Genetik Algoritma İle Ankara İlinde Mobese (Kent Güvenlik Yönetim Sistemi) Yerleştirme Optimizasyonu*, Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, Ankara, 38-49.
- İnternet: Akbulut, D. (2009). Evrimsel Tasarım Yöntemi ve Yaratıcılığın Süreç İçerisindeki Yeri. URL: <http://dergipark.ulakbim.gov.tr/sanatvetasarim/article/view/5000137099/5000126069>, Son Erişim Tarihi: 13.01.2019.
- İnternet: Akyol Altun, T. D. ve Köktürk, G. Ütopyadan Gerçeğe Doğa Bilimlerinin Mimarlığa Etkileri Robotik, Nanoteknoloji ve Genetik URL: <http://gulden.kokturk.com/isparta.pdf>, Son Erişim Tarihi: 05.03.2019.
- İnternet: Cevre, U., Özkan, B. ve Uğur, A. Gezgin Satıcı Probleminin Genetik Algoritmalarla Eniyilemesi ve Etkileşimli Olarak İnternet Üzerinde Görselleştirilmesi. URL: <https://docplayer.biz.tr/2986350-Gezgin-satici-probleminin-genetik-algoritmalarla-eniyilemesi-ve-etkilesimli-olarak-internet-uzerinde-gorsellestirilmesi.html>, Son Erişim Tarihi: 19.4.2019.
- İnternet: Chouchoulas, O. (2003). Shape Evolution URL: <http://fufurasu.org/research/aid02abstract.pdf>, Son Erişim Tarihi: 05.05.2019.
- İnternet: Cunkaş, M. (2006). Genetik Algoritmalar ve Uygulamaları. URL: <https://docplayer.biz.tr/14536665-Yrd-doc-dr-mehmet-cunkas.html>, Son Erişim Tarihi: 14.12.2018.
- İnternet: Dianati, M., Song, I. ve Treiber, M. (2003). An Introduction to Genetic Algorithms and Evolution Strategies. URL: <http://pami.uwaterloo.ca/~sd625/Students/msm/gaes.pdf>, Son Erişim Tarihi: 21.03.2019.
- İnternet: Genetik Algoritmalar. URL: http://tr.wikipedia.org/wiki/Genetik_algoritmalar, Son Erişim Tarihi: 19.11.2018.
- İnternet: Greg Lynn's Embryological House: case study in the preservation of digital architecture. URL: <http://www.docam.ca/conservation/embryologicalhouse/GL3ArchSig.html>, Son Erişim Tarihi: 04.11.2019.
- İnternet: Kartoğlu, G. Dijital Ortamda Mimari Tasarım Kapsam Ve Olanakları. URL: https://www.academia.edu/7341001/Seminer_Makale_Hesaplama%C4%B1_Tasar%C4%B1m <http://www.ona.vg/vision/growthhouse.html>, Son Erişim Tarihi: 15.02.2019.
- İnternet: Kopar, A. (2013). Mimarlık ve Matematik. URL: <http://www.mimarimedya.com/mimarlik-ve-matematik/>, Son Erişim Tarihi: 26.04.2019.
- İnternet: Livesey, G., Parlac, V., Taron, J. M. ve Langjing, X. (2011). The Embryological House, Greg Lynn. URL:

- https://xulangjing.files.wordpress.com/2011/10/fs_form_langjing_xu.pdf, Son Eriřim Tarihi: 05.05.2019.
- İnternet: Livesey, G., Parlac, V., Taron, J. M. ve Langjing, X. (2011). The Embriyological House, Greg Lynn. URL: https://xulangjing.files.wordpress.com/2011/10/fs_form_langjing_xu.pdf, Son Eriřim Tarihi: 05.05.2019.
- İnternet: Multi Sudoku. URL: <https://sudoku.matematiktutkusu.com/sudoku-cesitleri/6multi-sudoku.html>, Son Eriřim Tarihi: 16.03.2019.
- İnternet: Multi Sudoku. URL: <http://www.djape.net/multi-sudoku/>, Son Eriřim Tarihi: 22.03.2019.
- İnternet: Nedir? URL: <http://www.sudokuoyuna.net/nedir.html>, Son Eriřim Tarihi: 05.02.2019.
- İnternet: Project for a Helix City URL: <http://www.kisho.co.jp/page/200.html>, Son Eriřim Tarihi: 22.04.2019.
- İnternet: Sudoku Çeřitleri. URL: <https://sudoku.matematiktutkusu.com/sudoku-cesitleri/>, Son Eriřim Tarihi: 29.03.2019.
- İnternet: Sudoku Nasıl Oynanır. URL: <https://sudoku.yazarokur.com/nasil-oynanir>, Son Eriřim Tarihi: 18.03.2019.
- İnternet: Sudoku Nedir? URL: <https://sudoku.matematiktutkusu.com/31-sudoku-nedir.html>, Son Eriřim Tarihi: 2019-04-05.
- İnternet: Sudoku Nedir? URL: <https://www.bilgiustam.com/sudoku-nedir/>, Son Eriřim Tarihi: 11.03.2019.
- İnternet: Sudoku Örneęi. URL: <https://eodev.com/gorev/12054645>, Son Eriřim Tarihi: 20.05.2019.
- İnternet: řiřman, M. (2017). *Sudoku nedir? Sudoku nasıl çözölür?*. URL: <http://www.milliyet.com.tr/Sudoku-nedir--Sudoku-nasil-cozulur--molatik-302/>, Son Eriřim Tarihi: 03.03.2019.
- İnternet: The Molecular House. URL: <http://www.ona.vg/vision/growthhouse.html>. Son Eriřim Tarihi: 14.04.2019.
- İnternet: Yalçın, N. *Genetik Algortimalar*. URL: <https://docplayer.biz.tr/7339332-Genetikalgortimalar-aras-gor-nesibe-yalcin-bilecik-universitesi.html>, Son Eriřim Tarihi: 05.02.2019.
- İřçi, Ö. ve Korukoęlu, S. (2003). Genetik Algoritma Yaklařımı ve Yöneylem Arařtırmasında Bir Uygulama. *Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Faköltesi Dergisi*, 10(2), 191-208.
- İřlier, A. A. (2001). Üretim Hücrelerinin Bir Genetik Algortima Kullanılark Oluřturulması. *Anadolu Üniversitesi Bilim Ve Teknoloji Dergisi*, 2(1), 137-157.

- Johansen, J. M. (2002). *Nanoarchitecture: A New Species of Architecture*. New York: Princeton Architectural Press, 132-133.
- Karaday, T. (2016). *Mekanik Otopark Tesislerinin Yer Seçimi Ve Boyutlandırılması İçin Genetik Algoritma Tabanlı Bir Yaklaşım*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Karaday, O. ve Ballı, S. (2016). Google Maps ve Genetik Algoritmalarla GSP Çözümü İçin Öneri. *XVIII. Akademik Bilişim Konferansı*.
- Kaya, İ. (2012). *Genetik Algoritmaların Optimal Güzergah Belirlenmesine Uygulanması*, Yüksek Lisans Tezi, Haliç Üniversitesi Fen bilimleri Enstitüsü, İstanbul, 31.
- Kevran, M. M. (2009). *Çoklu Sensör Konumlandırma Problemlerinin Genetik Algoritmalar ile Çözülmesi*, Hava Harp Okulu Komutanlığı Havacılık ve Uzay Teknolojileri Enstitüsü, İstanbul, 25.
- Kolareviç, B. (2003). *Architecture in the Digital Age: Design and Manufacturing*. New York: Spon Press, 37.
- Kurt, M. ve Semetay, C. (2001). Genetik Algoritmalar ve Uygulama Alanları. *Mühendis ve Makina Dergisi*.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. London: The MIT Press, 3,11,131.
- Öktem, Ü. (2010). Olgu, Kuram, Darwin Öncesi Evrim Kuramları ve Darwin'in Evrim Kuramı. *Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Antropoloji Dergisi*, (23), 021-039.
- Özçakar, N. (1998). Genetik Algoritmalar. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 27(1), 69-82.
- Satman, M. H. (2016). *Genetik Algoritmalar*. İstanbul: Türkmen Kitabevi, 120-122.
- Shadmand, S. (2015). *Biçim Oluşturmada Doğadan Yararlanılarak Üretken Bir Sistemin Denenmesi*, Yüksek Lisans Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 56.
- Söke, A. (2003). *Genetik Algoritma Ve Benzetilmiş Tavlama İle İki Boyutlu Giyotinsiz Kesme Problemlerine Olasılıksal Yaklaşım*, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, 34.
- Stockton, D., Quinn, L., ve Khalil, R. (2004). Use of genetic algorithms in operations management Part 1: Applications. *Proceedings of the Institution of Mechanical Engineers*, 218(3), 315-327.
- Şeker, Ş. (2007). *Araç Rotalama Problemleri Ve Zaman Pencere Stokastik Araç Rotalama Problemlerine Genetik Algoritma Yaklaşımı*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Şeker, T. (2008). *Düzlemsel Çelik Çerçevelerin Genetik Algoritma İle Optimizasyonu*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 41.

- Vural, M. (2005). *Genetik Algoritma Yöntemi İle Toplu Üretim Planlama*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 7-31.
- Yeniay, Ö. (2001). An Overview Of Genetic Algorithms. *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2(1), 37-49.
- Yurtçu, Ş. ve İçağa, Y. (2006). Evrimsel Algoritmaların İnşaat Mühendisliği Sistemlerinde Kullanımı. *Yapı Teknolojileri Elektronik Dergisi*, 1, 51-59.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : DEMİRKOL, Şule Betül
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 18.11.1991, Malatya
 Medeni hali : Bekâr
 Telefon : 0 (535) 442 52 44
 e-mail : sulebetul44@gmail.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Mimarlık	Devam Ediyor
Lisans	Gazi Üniversitesi / Mimarlık	2015
Lise	Malatya Lisesi	2009

İş Deneyimi

Yıl	Yer	Görev
2015-2019	Vega Mimarlık Müh. Turz. San. Tic. Ltd. Şti	Mimar
2014-2015	Aymaz Mimarlık Müh. İnş. Ltd. Şti	Mimar

Yabancı Dil

İngilizce

Yayınlar

Demirkol, Ş. B. ve Özen Yavuz, A. (2018). Genetik Algoritma Yaklaşımının Mimaride Kullanımı ve Örneklerle Genotip Oluşturma, *II. Uluslararası Multidisipliner Çalışmaları Kongresi*

Hobiler

Yüzmek, Kitap okumak, Puzzle yapmak



GAZİ GELECEKTİR..