

**T.C.
NECMETTİN ERBAKAN ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İÇ MİMARLIK VE ÇEVRE TASARIMI ANABİLİM DALI
İÇ MİMARLIK VE ÇEVRE TASARIMI BİLİM DALI**

**HESAPLAMALI MİMARİ TASARIMIN ANLAMININ
SORGULANMASI VE HESAPLAMALI TASARIM
YÖNTEMLERİNİN İÇ MİMARİ TASARIM SÜRECİNE
ETKİSİNİN İRDELENMESİ**

Alptekin TORUN

YÜKSEK LİSANS TEZİ

**DANIŞMAN
Doç. Dr. Osman PERÇİN**

KONYA - 2023



T.C.
NECMETTİN ERBAKAN ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü



ÖZET

Öğrencinin	Adı Soyadı	Alptekin TORUN		
	Numarası	20813101003		
	Ana Bilim / Bilim Dalı	İç Mimarlık ve Çevre Tasarımı / İç Mimarlık ve Çevre Tasarımı		
	Programı	Tezli Yüksek Lisans	X	
		Doktora		
	Tez Danışmanı	Doç. Dr. Osman PERÇİN		
Tezin Adı	Hesaplamalı Mimari Tasarımın Anlamının Sorgulanması ve Hesaplamalı Tasarım Yöntemlerinin İç Mimari Tasarım Sürecine Etkisinin İrdelenmesi			

Hesaplamalı mimari tasarım her geçen gün gelişen ve güncellenen bir konudur. Bu anlamda konuyla ilgili kavramlar sürekli olarak sorgulanmaktadır. Bu gelişim ve değişimle birlikte konuyla ilişkili bazı kavramlar anlam karışıklığı yaratacak şekilde kullanılabilir. Bu çalışmada bu anlam karışıklıklarını önlemek için kavramların güncel ve kapsamlı bir bakış açısıyla ele alınması amaçlanmıştır. Çalışmada hesaplamalı tasarlarken ortaya koyduğumuz hesaplamalı faaliyetlerden yola çıkarak bilişsel farkındalığımız irdelenmiş ve hesaplamalı mimari tasarımın anlamı bu bilinç üzerinden sorgulanmıştır. Hesaplamalı mimari tasarım yaklaşımları ve yöntemlerini hesaplamalı kılan özellikler ortaya konmaya çalışılmış, bu yaklaşımlar sınıflandırılarak diğer yaklaşımlarla ilişkileri irdelenmiştir. Hesaplamalı tasarım yaklaşımlarının iç mimari tasarım pratiğine etkileri örnekler ile değerlendirilerek hesaplamalı tasarım süreçlerinin geleneksel tasarım süreçlerinden farkları incelenmiştir.

Çalışmada hesaplamalı mimari tasarımın hesaplamalı düşünme ve hesaplamalı yapmanın kesiştiği bir konumda olduğu bulunmuştur. Hesaplamalı mimari tasarım; doğadan ilham alan, veriye dayalı, parametrik, algoritmik, üretken, optimizasyona dayalı, performansa dayalı olarak sınıflandırılmıştır. Hesaplamalı modellerin ve yöntemlerin birbirleriyle birçok yönden kesiştiği ve birbirlerinden beslendiği görülmüştür. Hesaplamalı yaklaşımlar birbirlerinden etkilense de tasarım süreçlerinde odaklandıkları fikirler ve tasarımda kullanılan yöntemler ile birbirlerinden ayrılmıştır. Bu anlamda yöntem ve modelleri, neyi sağladıklarından çok nasıl sağlayabildikleri ile tanımlamanın daha uygun olacağı sonucuna varılmıştır. Hesaplamalı mimari tasarım yöntemlerinin farklı iç mekân tasarımı uygulamalarında yararlanılabileceği görülmüştür. Ancak iç mimarlık disiplini açısından yorumlanan hesaplamalı tasarım araştırmaları nispeten daha azdır ve yeterli değildir. Bu anlamda bu alanda yapılacak çalışmaların konuya farklı bakış açıları getirebileceği düşünülebilir.

Anahtar Kelimeler: Hesaplamalı Mimari Tasarım, Hesaplamalı Düşünme, Hesaplamalı Tasarım Yöntemleri, Dijital Tasarım



T.C.
NECMETTİN ERBAKAN ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü



ABSTRACT

Author's	Name Surname	Alptekin TORUN		
	Student Number	20813101003		
	Department	Interior Architecture and Environmental Design / Interior Architecture and Environmental Design		
	Study Programme	Master's Degree (M.A.)	X	
		Doctoral Degree (Ph.D.)		
	Supervisor	Doç. Dr. Osman PERÇİN		
Title of the Thesis/Dissertation	Questioning The Meaning of Computational Architectural Design and Examining The Effect of Computational Design Methods on The Interior Architectural Design Process			

Computational architectural design is a constantly evolving and updating subject. In this regard, the concepts related to the topic are constantly being questioned. In order to prevent any confusion that may arise from this development and change, this study aims to approach the concepts with a current and comprehensive perspective. By examining our cognitive awareness based on the computational activities we put forward during computational design, the meaning of computational architectural design is questioned through this consciousness. The characteristics that make computational architectural design approaches and methods computational have been identified, and these approaches have been classified to analyze their relationships with other approaches. The effects of computational design approaches on interior architectural design practice have been evaluated with examples, and the differences between computational design processes and traditional design processes have been examined.

In the study, it was found that computational architectural design is in a position where computational thinking and computational doing intersect. Computational architectural design has been classified as nature-inspired, data-driven, parametric, algorithmic, generative, optimization-based, and performance-based. It has been seen that computational models and methods intersect with each other in many ways and feed on each other. Although computational approaches influence each other, they are differentiated from each other in terms of the ideas they focus on and the methods used in design. In this sense, it has been concluded that it is more appropriate to define methods and models based on how they can achieve rather than what they provide. It has been observed that computational architectural design methods can be used in different interior design applications. However, research on computational design interpreted from the perspective of interior design discipline is relatively less and insufficient. In this sense, it can be thought that studies in this field can bring different perspectives to the subject.

Keywords: Computational Architectural Design, Computational Thinking, Computational Design Methods, Digital Design

İÇİNDEKİLER

ÖZET	ii
ABSTRACT.....	iii
İÇİNDEKİLER	iv
ÇİZELGELER LİSTESİ	vii
ŞEKİLLER LİSTESİ	viii
KISALTMALAR	x
GİRİŞ.....	1
1. Araştırma Problemleri.....	2
2. Çalışmanın Amacı.....	2
3. Çalışmanın Önemi	2
4. Yöntem.....	3
1. HESAPLAMA (COMPUTATION) ve HESAPLAMALI DÜŞÜNME (COMPUTATIONAL THINKING)	4
1.1. Hesaplama ve Hesaplamalı Düşünmenin Tarihi.....	4
1.1.1. Geleneksel Hesaplamalı Düşünme Dönemi	5
1.1.2. Yeni Hesaplamalı Düşünme Dönemi	7
1.2. Hesaplamalı Düşünme	8
1.2.1. Hesaplamalı Düşünmenin Bileşenleri.....	9
1.2.1.1. Ayrıştırma	10
1.2.1.2. Soyutlama	10
1.2.1.3. Algoritmik Düşünme	11
1.2.1.4. Genelleme	12
1.2.1.5. Değerlendirme	12
1.2.2. Hesaplamalı Düşünme Araçları ve Uygulamaları	13
1.2.2.1. Modelleme, Simülasyon ve Görselleştirme	13
1.2.2.2. Otomasyon.....	13
1.2.2.3. Kodlama ve Programlama	14
1.2.3. Hesaplamalı Düşünmenin Diğer Düşünme Türleri ve Süreçleri ile İlişkileri	14
1.2.3.1. Hesaplamalı Düşünmenin Problem Çözme Süreciyle İlişkisi	14

1.2.3.2. Hesaplmalı Düşünmenin Mantıksal Düşünmeyle İlişkisi.....	15
1.2.3.3. Hesaplmalı Düşünmenin Eleştirel Düşünmeyle İlişkisi.....	17
1.2.3.4. Hesaplmalı Düşünmenin Analitik Düşünmeyle İlişkisi.....	17
1.2.3.5. Hesaplmalı Düşünmenin Sistemli Düşünmeyle İlişkisi.....	18
1.2.3.6. Hesaplmalı Düşünmenin Matematik ve Bilgisayar Bilimi ile İlişkisi .	18
1.2.3.7. Hesaplmalı Düşünmenin Mühendislik ve Mimari Tasarım Düşüncesi ile İlişkisi.....	20
1.3. Farklı Disiplinleri Kapsayan Bir Kavram Olarak Hesaplmalı Düşünme.....	21
2. HESAPLAMALI MİMARİ TASARIM	23
2.1. Hesaplmalı Mimari Tasarım	23
2.2. Hesaplmalı Mimari Tasarımın Tarihi	25
2.3. Hesaplmalı Tasarımın Boyutları ve Mimari Tasarım Sürecine Yansımaları....	30
2.3.1. Dijital Olmayan Hesaplmalı Tasarım Aracını Kullanmak.....	30
2.3.2. Hesaplmalı Tasarım Aracı Olmayan Hesaplmalı Tasarım Ürünü Kullanmak.....	32
2.3.3. Hesaplmalı Dijital Tasarım Araçları ile Hesaplmalı Mimari Tasarımlar Yapmak	32
2.3.4. Bizim Yerimize Düşünen Makineler	33
2.4. Hesaplmalı Mimari Tasarım Yaklaşımları, Yöntemleri ve Modelleri.....	34
2.4.1. Parametrik Yaklaşım.....	35
2.4.1.1 Parametrik Modeller ve Parametrik Yöntemler.....	36
2.4.1.2. Parametrik Yaklaşımın Hesaplmalı Tasarım ve Diğer Hesaplmalı Yaklaşımlar ile İlişkisi	40
2.4.2. Algoritmik Yaklaşım	42
2.4.2.1. Algoritmik Tasarım Modeli ve Yöntemleri.....	43
2.4.2.2. Algoritmik Tasarım Yaklaşımının Hesaplmalı Tasarım ve Diğer Hesaplmalı Yaklaşımlar ile İlişkisi	46
2.4.3. Üretken Yaklaşım	488
2.4.3.1. Üretken Tasarım Modelleri ve Yöntemleri.....	48
2.4.3.2. Üretken Tasarımın Hesaplmalı Tasarım ve Diğer Hesaplmalı Yaklaşımlar ile İlişkisi	60
2.4.4. Performansa Dayalı Yaklaşım	61
2.4.4.1. Performansa Dayalı Tasarım Modeli ve Yöntemleri.....	62

2.4.4.2. Performansa Dayalı Yaklaşımın Hesaplama Tasarım ve Diğer Hesaplama Yaklaşımları ile İlişkisi	65
2.4.5. Optimizasyona Dayalı Yaklaşım	66
2.4.5.1. Optimizasyona Dayalı Modeller ve Yöntemler	66
2.4.5.2. Optimizasyona Dayalı Yaklaşımın Hesaplama Tasarım ve Diğer Hesaplama Tasarım Yaklaşımları ile İlişkisi	74
2.4.6. Doğadan İlham Alan Yaklaşımlar	76
2.4.6.1. Doğadan İlham Alan Modeller ve Yöntemler	77
2.4.6.2. Doğadan İlham Alan Yaklaşımların Hesaplama Tasarım ve Diğer Hesaplama Tasarım Yaklaşımları ile İlişkisi	80
2.4.7. Veriden Öğrenmeye (Deneyim) Dayalı Yaklaşım	82
2.4.7.1. Makine Öğrenimi Modelleri ve Yöntemleri	83
2.4.7.2. Makine Öğreniminin Hesaplama Tasarım ve Diğer Hesaplama Tasarım Yaklaşımları ile İlişkisi	92
3. İÇ MİMARİ TASARIMDA HESAPLAMALI TASARIM SÜRECİ	94
4. BULGULAR ve YORUMLAR	102
5. SONUÇ ve ÖNERİLER	120
KAYNAKLAR	124

ÇİZELGELER LİSTESİ

Çizelge 1.1. Problem çözme süreci ve hesaplamalı düşünme süreci (Kalelioglu ve diğerleri, 2016).....	15
Çizelge 4.1. Hesaplamalı mimari tasarım yaklaşımlarının odaklandıkları temel noktalar ve bu yaklaşımların öne çıkan özellikleri	103
Çizelge 4.2. Hesaplamalı mimari tasarım yaklaşımlarının iç mimari tasarım sürecine etkileri	107



ŞEKİLLER LİSTESİ

- Şekil 1.1.** Matematik biliminden, bilgisayar bilimine, bilgisayar biliminden hesaplamalı bilimlere, hesaplamalı bilimlerden hesaplamalı düşünmeye ve hesaplamalı alanlara doğru genişleyen ve gelişen sürecin temsili..... 22
- Şekil 2.1.** Hesaplamalı tasarım, hesaplamalı düşünmenin ilişkisi..... 24
- Şekil 2.2.** Parametrik kısıtlamaların gösterimi. 37
- Şekil 2.3.** Zaha Hadid Architects tarafından tasarlanan parametrik yapılar, a'da Serpentine Sackler Galerisi, b'de Roca Londra Galerisi, c'de Stuart Weitzman Falgship Mağazası. 39
- Şekil 2.4.** A'da ofis bölücü örneği, b'de parametrik modelin geometrik formunun esnekliği, c'de boyutsal kısıtlamalar sayesinde birbirlerine eklenecek şekilde tasarlanmış parçalar, d'de alternatif ürün 40
- Şekil 2.5.** Parametrik, dijital ve BIM modellerinin birbirleriyle ilişkilerini gösteren venn şeması. 41
- Şekil 2.6.** Rhino/Grasshopper arayüzü. 45
- Şekil 2.7.** Algoritmik mimari modelin, parametrik, dijital ve BIM modelleri ile ilişkilerini gösteren venn şeması. 47
- Şekil 2.8.** Bir yapının kütle organizasyonu üzerinde yapılan biçim grameri analizi. A'da başlangıç şekli, b'de kurallar, c'de evin kütleli gelişim şeması 51
- Şekil 2.9.** Biçim gramerinin apartman yerleşimi planları üretimi için kullanımı..... 52
- Şekil 2.10.** A'da Hücresel otomat modelinin geçiş kuralları, b'de hücre başlangıç durumu, c'de algoritma tarafından oluşturulan bir model 54
- Şekil 2.11.** A'da fraktal şekillerin oluşturulma adımları ve b'de, oluşturulan fraktal şeklin yemek masası tasarımında kullanım örneği 59
- Şekil 2.12.** Birbirlerine entegre şekilde kullanabilecek programlar örneği. 63
- Şekil 2.13.** A'da mutasyon öncesi ve sonrası, b'de çaprazlama sonucu oluşan yavrular, c'de biyoloji odaklı terminolojiye göre 0 ya da 1 sembolüne gen, genlerden oluşan diziye birey ve popülasyon, d'de kodlama ve kod çözme 70
- Şekil 2.14.** Genetik algoritmanın tasarım süreci aşamaları: formülasyon, yürütme ve keşif aşamaları 70
- Şekil 2.15.** Popülasyon başlatıldıktan sonra, maliyet fonksiyonunun değerlendirilmesi, en iyi bireylerin seçilmesi, çaprazlama ile yeni bireylerin oluşturulması ve bireylerin mutasyonlarla değiştirilmesi olarak devam eden yinelemeli süreç 72
- Şekil 2.16.** A'da sistemin arayüzü, b'de sistem ile oluşturulmuş optimum tasarım sonucu. 73

Şekil 2.17. A'da alternatif çıktılar, b'de sonuç	74
Şekil 2.18. Üretken modeller, optimizasyon algoritmaları ve bina performans simülasyonlarının kesişim kümesi olarak performansa dayalı hesaplamalı tasarım optimizasyonu modeli	75
Şekil 2.19. A'da esinlenen bitkinin yapısı, b'de modelin tasarımında kullanılan prosedür örneği, c'de üç boyutlu model örneği, d'de üç boyutlu yazıcı kullanılarak basılan model	79
Şekil 2.20. Hücresel büyüme simülasyonunun gösterimi.....	80
Şekil 2.21. Yapay sinir ağı modeli	87
Şekil 2.22. A'da veri setindeki örnekler, b'de sentezlenmiş sahneler, c'de birden çok yineleme ile yürütülen bir sahne	90
Şekil 2.23. Stil odaklı mekan sentezi modeli.....	91
Şekil 3.1. Hesaplamalı düşünme sürecini gösteren temsili görsel.....	94
Şekil 3.2. İlham alınan geometrik örüntüler	95
Şekil 3.3. Örnek görsellerdeki tekrar eden ve birbirlerinin yansımaları olan kompozisyonlar.....	96
Şekil 3.4. Uygulanan prosedürün adımları	97
Şekil 3.5. Prosedürün tamamı.....	98
Şekil 3.6. Çizgi kalınlığı ayarı	99
Şekil 3.7. Örüntünün bozulmalar olmadan boyutunun ayarlanması	100
Şekil 3.8. Değişkenlerin değiştirilmesi ile üretilen örneklerden bazıları	101
Şekil 3.9. Seperator örneklerinin kullanımı.....	101
Şekil 4.1. Hesaplamalı tasarım ve sonrasını gösteren sürecin temsili	102

KISALTMALAR

BIM	: Yapı bilgi modelleme
CAD	: Bilgisayar destekli tasarım
CSTA	: Computer Science Teachers Association
ISTE	: International Society for Technology in Education
vb.	: ve benzeri



ÖN SÖZ

Tez çalışmam süresince destek ve anlayışını esirgemeyen değerli danışmanım Doç. Dr. Osman PERÇİN'e, görüş ve önerileri için değerli hocalarıma teşekkürlerimi borç bilirim. Beni destekleyen, yanımda olan aileme ve emeği geçen herkese teşekkür ederim.

Alptekin TORUN

Konya, 2023



GİRİŞ

Hesaplamalı tasarım (computational design), mimariyi de kapsayan çok disiplinli bir alandır. Bu alan mimari tasarım pratiğini etkilemiş, mimari tasarım teorisinin önemli konularından biri haline gelmiştir. 21. yüzyılda dijital teknolojilerin kullanımının artmasıyla birlikte mimari tasarımda hesaplamalı tasarım süreçlerinin kapasitesi genişlemiştir. Bilgisayarlar sadece tasarımı aktarmakla kalmayıp tasarımın bütün aşamalarını yönlendirebilmiştir. Böylece karmaşık geometriler, hesaplamalar kontrol edilmiş veya izlenmiştir. Geometrik yüzeylerin bilgisayar yazılımlarında sayısal veriler ile biçimlendirilebilmesi, tasarımcıları bilinçli ve hesaplamalı bir düşünmeye itmektedir. Hesaplamalı düşünme süreçleriyle birlikte hesaplamalı yöntemler, mimari tasarımcıların düşünme ve tasarlama şekillerini değiştirmiş, üretme kapasitelerini artırmış ve yaptıkları işlemleri hızlandırmıştır. Formun işlevle birlikte biçimlendiği bu süreçte birçok tasarım alternatifi yaratılabilmektedir. Böylece mimari tasarımda özgün tasarımlar ortaya çıkmıştır.

Mimari tasarımda hesaplamalı tasarım yaklaşımları her geçen gün daha çok benimsenmektedir. Bu hızlı gelişim, mimari tasarımda yeni deneyimleri beraberinde getirmektedir. Hesaplamalı tasarım yaklaşımlarındaki gelişmeler, mimari tasarım sürecine bakış açışımızı değiştirmeye devam ettiği gibi ele alınan terimlerin tutarsızca kullanımına da sebep olabilmektedir. Bu durum anlam karışıklığına ya da kavramların anlaşılmasına neden olmaktadır. Bahsedilen bu belirsizlik ve tutarsızlıkları ortadan kaldırmak ve hesaplamalı tasarım süreçlerinin iç mimari tasarım pratiğine etkisini incelemek için aşağıdaki araştırma problemleri üzerinde durulmuştur.

1. Araştırma Problemleri

- Hesaplamalı mimari tasarımı “hesaplamalı” kılan nedir?
- Hesaplamalı mimari tasarımın evrimi nasıldır?
- Hesaplamalı mimari tasarım yaklaşımları nelerdir ve bu yaklaşımları birbirinden ayırt edebilir miyiz?
- Hesaplamalı yaklaşımlar iç mimari tasarım sürecini nasıl etkiler?

2. Çalışmanın Amacı

Öncelikle hesaplamalı tasarım süreçlerini “hesaplamalı” yapan anlamının bağlamını ortaya çıkarmak amaçlanmıştır. Bu sayede hesaplamalı mimari tasarım yaklaşımları ve yöntemlerini konumlandırabilmek için sağlam bir zemin oluşturulmak istenmiştir. Hesaplamalı mimari tasarım yöntemlerinin hesaplamayla ve diğer hesaplamalı yöntemler ile ilişkileri ve farklılıkları incelenmiştir. Bu yöntemlerin iç mekân tasarımı sürecine etkileri değerlendirilmiştir.

3. Çalışmanın Önemi

Hesaplamalı tasarım, sürekli gelişen ve bu anlamda her zaman güncelliğini koruyan bir alandır. Dolayısıyla yeni tartışmalara, sorgulamalara da açıktır. Ancak hesaplamalı modellerin altında yatan fikirlerde hâlâ tutarlılık yoktur (Caetano, Santos ve Leitão, 2020). Eksiklerin giderilmesi için bu çalışmanın yararlı olacağı düşünülmektedir. Öte yandan hesaplamalı düşünmenin, hesaplamalı mimari tasarım yöntemleriyle ilişkilendirilmesi ile konu farklı bir açıdan değerlendirilmiştir.

Hesaplamalı mimari tasarım yöntemlerindeki yeniliklerin ve karmaşıklıkların artmaya devam etmesi, kullanılan terim ve tekniklerin netleştirilmesini gerekli kılar (Fischer ve Herr, 2001). Dolayısıyla yaklaşımların, kavramların ve yöntemlerin yenilenmiş bir bakış açısıyla ele alınması geçmişte olduğu gibi bugün de önemini korumaktadır.

Diğer yandan iç mimarlık anabilim dalı altında hesaplamalı mimari tasarım alanında yapılmış çalışmalar nispeten azdır. Bu anlamda konunun bu disiplinin bakış açısından incelenmesi değerli olabilir.

4. Yöntem

Bu çalışmada literatür taramasıyla elde edilen bilgiler ışığında problemler tespit edildi ve veriler toplandı. Bu anlamda çeşitli tezler, makaleler, kitaplar, web, videolar ve bildirilerden yararlanıldı. Çalışmada, hesaplamalı tasarım teorisyenleri ve bilim adamları tarafından yapılan araştırmalar incelendi. Araştırmanın evrenini hesaplamalı düşünme ve hesaplamalı tasarım alanında yapılmış çalışmalar oluşturmaktadır. Örnekleme ise hesaplamalı düşünmenin ve hesaplamalı mimari tasarımın tarihine, anlamının sorgulanmasına ve bileşenlerine yoğunlaşan çalışmalar ile birlikte iç mekân tasarımı etkileyen hesaplamalı tasarım yöntemlerini kapsayan çalışmalar oluşturur.

Çalışmada alanyazın taraması ile ulaşılan bilgilere göre hesaplamalı mimari tasarımın boyutları, yöntemleri ve modelleri sınıflandırılmıştır. Hesaplamalı mimari tasarım yaklaşımlarının odaklandıkları noktalar ve öne çıkan özellikleri analiz edilmiştir. Alanyazın taraması boyunca olgunlaşan fikirlere göre biriktirilen veriler gözden geçirilerek örnek iç mekân tasarımı uygulamaları seçilmiştir. İç mekân tasarımı için uygulamalarda kullanılan hesaplamalı mimari tasarım yöntemleri, sınıflandırmalara göre değerlendirilmiştir. Hesaplamalı yaklaşımların iç mimari tasarım sürecine etkileri avantajlar ve dezavantajlar açısından incelenmiştir.

BİRİNCİ BÖLÜM

HESAPLAMA (COMPUTATION) ve HESAPLAMALI DÜŞÜNME (COMPUTATIONAL THINKING)

1.1. Hesaplama ve Hesaplamalı Düşünmenin Tarihi

Edward Fox (National Research Council, 2010: 16), hesaplamalı düşünmenin uzun bir tarihsel geçmişe sahip olduğundan bahseder:

“Hesaplamalı düşünce, insan türünde doğuştandır, atalarımız hikâyeler anlatarak gerçeği modelleyip temsil ettiler ve bunu diğer insanlara aktardılar ve bu modelleri keşfetme, bulma ve yaşamı sürdürme için zenginleştirdiler.”

Edward Fox, bu ifadeleriyle aslında hesaplamanın kaçınılmaz bir gerçek olduğunu anlatmak istemiştir. Denning’e (2007) göre hesaplama, evrenseldir ve her yerdedir; biz onu gözlemlemesek de veya onun hakkında düşünmesek bile hesaplama doğada mevcuttur. Bu anlamda hesaplamalı bilgi süreçlerinin yapay olduğu kadar, doğal olarak da gerçekleştiğinin kabul edilmesi hesaplamanın kapsamını genişletir.

Hesaplamalı düşünce terimlerinin tanımı henüz netleştirilememiştir. Aslında ‘hesaplama’ kavramının tam olarak çözülememesinin nedeni, olgunlaşan düşüncelerin sürekli olarak yeni anlayışlara ve mevcut modeller hakkında sorulara yol açmasıdır (Denning, 2010). Çinici’ye (2012: 12,13) göre bu keşfe ve gelişmeye açık alan, “süregelmekte olan düşünme ve üretme enerjisinin kaynağı” gibidir. Bu anlamda hesaplamalı düşünme ve hesaplama kavramlarını anlamak için bu kavramları olgunlaştıran tarihsel süreci kavramak gerekir.

‘Computation’ kavramının ilk kullanımı bilgisayar teknolojilerinin ortaya çıkışından çok daha önceye dayanır. Etimolojik kökenine bakıldığında Latince computare’den geldiği görülen ve 15. yüzyılda ilk defa kullanıldığı düşünülen ‘computation’ kelimesi, içinde hem matematiksel çağrışımlarıyla hem de bir hesaba katma anlamıyla kullanılmıştır (Çinici, 2012: 12,13).

Günümüzün ‘hesaplama’ ve ‘hesaplamalı düşünme’ tanımlarının doğrudan öncülleri olan temel kavramlar 1930’lardan 1990’lara kadar olan dönemde

tartışılmıştır. ‘Hesaplama’ teriminin anlamı, hesaplamanın en eski biçimsel kavramlarının altında yattığı 1930’lu yıllarda, ‘bilgisayar’ terimi ile ortak kullanımda olmuştur, ancak günümüzden farklı çağrışımlarla ifade edilmiştir (Denning, 2010). 1930’lu yıllarda hesaplama, Turing makinelerini veya eşdeğerlerini içermiştir (Denning, 2010). Yani hesaplama, makinenin toplam yapılandırmalarını kapsayan duran yürütme dizileri olarak tanımlanmıştır. Diğer bir deyişle, hesaplamanın geleneksel tanımı, duran algoritmalarla ilişkili olmuştur. 1930’larda önerilen dört hesaplama modeli olan özyinelemeli fonksiyonlar (Gödel, 1934), yeniden yazma kuralları (Post, 1936), lambda hesabı (Church, 1936) ve Turing makinesi (Turing, 1936) hesaplama için referans modeller olmuştur. Hesaplama terimi ise matematiksel fonksiyonları değerlendirmek için mekanik adımlar olarak bilinmiştir. ‘Hesaplamalı düşünme’ kavramı 1980’li yıllara kadar telaffuz edilmese de anlamsal boyutunu karşılamaya çalışan bunun gibi ifadeler sıkça kullanılmıştır. Bu anlamda hesaplamalı düşünmenin anlamını oluşturan temelin 1950’li yıllara dayandığı düşünülür (Denning, 2009a, 2010; Tedre ve Denning, 2016).

Hesaplamalı düşünme teriminin ilk kullanımı 1980’li yıllarda olmuştur (Papert, 1980). Bu yıllarda hesaplamalı düşünmenin eğitim olarak verilmesi tartışılmıştır. Ayrıca o yıllara denk gelen hesaplamalı bilimin doğuşu, hesaplamalı düşünmenin önemli kırılma noktalarından biri olarak görülmüştür. Hesaplamalı bilimin doğuşuyla birlikte hesaplamalı düşünme, bilgisayar bilimi anlayışının doğal bir evrimi olarak belirlemiştir. Diğer bir kırılma noktası, hesaplamalı düşünmenin yeniden gündeme geldiği Jeanette Wing’in (Wing, 2006) makalesinin yayınlanması ile olmuştur. Hesaplamalı düşünme, Wing’in makalesi ile birlikte yeniden tartışmaya açılmış ve yeni bir versiyona bürünmüştür. Denning, bu sebeple hesaplamalı düşünmeyi “geleneksel” ve “yeni” olarak ikiye ayırmıştır (Denning 2017a: 37).

1.1.1. Geleneksel Hesaplamalı Düşünme Dönemi

1950 ve 1960’larda hesaplama için “algoritmik düşünme” ifadesi kullanılmıştır ve algoritmik düşünme, bir dizi makine talimatı tasarlama süreci olarak tanımlanmıştır (Tedre ve Denning, 2016). Alan Perlis, algoritma kavramının zaten başından beri insanlığın bir parçası olduğunu ve bilgisayarların tüm alanlardaki süreçleri

otomatikleştirerek bu süreçleri algoritma haline getirilebileceğini savunmuştur (Katz, 1960). Hesaplamanın tanımının bir dizi makine talimatını içeren algoritmik sürece ve otomasyon ile ilgili bir endişeye bağlanması (Arden, 1980), hesaplamanın makinelere ve bilgisayarlara bağlı bir süreç olarak düşünülebileceğini göstermektedir.

1970 ve 1980'lerde; hesaplama ve hesaplama için önemli kavramlar detaylandırılmış ve bu kavramları birbirlerinden ayıran özellikler irdelenmiştir. Hesaplama ve algoritma kavramının matematik biliminden farkları (Gorn, 1967; Knuth, 1974, 1981, 1985; Tseytin, 1981), matematik biliminin de bilgisayar biliminden farkları incelenmiştir (Dijkstra, 1974; Forsythe, 1968). Edsger W. Dijkstra (1967), algoritmayı bir makineden uyandırılan durum dizisi, hesaplamayı ise yapılan asıl iş olarak ifade etmiştir.

Hesaplamalı düşünme (Computational Thinkking) terimi, ilk olarak Seymour Papert (1980) tarafından dile getirilmeye başlanmıştır (Harel ve Papert, 1991; Papert, 1980, 1996). Ancak, araştırmalarında hesaplamalı düşünceyi net bir şekilde tanımlamamıştır. Papert, çalışmalarında programlama ile problem çözme yeteneklerinin geliştirilebileceğini iddia etmiştir. Papert (1980), 60'lı yıllarda MIT Medya Laboratuvarı'nda bilgisayar tabanlı öğrenme konusundaki ilk deneyleri sayesinde Logo adlı eğitim programlama dilini geliştirmiştir. 1980'lerde, MIT Media Laboratuvarı'nda Harel ve Papert (1991) öğrenmeye dâhil edilmesi gereken yaratıcı süreçleri teşvik etmek için bilgisayarları öğrenme sürecinde kullanmayı önermişlerdir.

1980'lerde geliştirilen Süper Bilgisayarlarla birlikte hesaplamalı bilimler doğmuştur. Bu hareketlilik hesaplamalı düşünce fikrinin gelişmesine büyük bir katkısı olmuştur (Tedre ve Denning, 2016). Bilgi işlemdeki gelişmeler ile birlikte araştırmacılar hesaplamalı düşünme yöntemlerini kendi alanlarında kullanmak için öğrenmiş ve geliştirmiştir.

Hesaplamalı modeller yaratmada elde ettiği başarılarıyla Nobel Fizik Ödülü'nü alan Kenneth G. Wilson, alanında önde gelen birçok bilim adamıyla birlikte hesaplamalı düşünmenin önemine dikkat çekmiş, ayrı bir bilim olarak hesaplamalı bilimin gerekliliğini vurgulamıştır (Denning 2009: 29). O ve diğerleri hesaplamayı

yeni bir bilim dalı, kullandıkları düşünce süreçlerini ise hesaplamalı düşünme olarak tanımlamıştır.

1.1.2. Yeni Hesaplamalı Düşünme Dönemi

Jeannette Wing'in yazdığı makale ile "hesaplamalı düşünce" terimi tekrar gündeme gelmiştir (Wing 2006: 33). Wing (2006), hesaplamalı düşünmeyi, "bilgisayar bilimi kavramlarından yararlanarak problemleri çözmeye, sistemler tasarlama ve insan davranışlarını anlama" yolu olarak tanımlamıştır. Wing (2006), hesaplamalı düşünmenin bilgisayar bilimi alanının genişliğini yansıtan bir dizi zihinsel aracı içerdiğini ancak sadece bilgisayar bilimcilerinin değil herkesin öğrenip kullanabileceği bir beceri seti olduğunu ifade etmiştir. Ona göre hesaplamalı düşünme, uygun bir çözüm üretmek için planlama, soyutlama, uygun parçalara indirgeme sürecini ifade etmektedir.

Wing (2006) tarafından yayınlanan makalenin derinlemesine analizden yoksun olması, hesaplamalı düşünme tanımının belirsiz ve aşırı kapsamlı olması birçok eleştiri almıştır (Barr and Stephenson 2011; Denning 2017a; Hemmendinger 2010; Tedre and Denning 2016). Hesaplamalı düşünmeyi yeni bir kavram olarak düşünmenin yanıltıcılığı, kavramın tarihsel iç görüşünün eksikliği ve hesaplamalı düşüncenin anlamının kapsamıyla ilgili netsizlik, kendisine yoğun eleştirileri beraberinde getirmiştir (Barr and Stephenson 2011; Denning 2017a; Hemmendinger 2010; Tedre and Denning 2016). Araştırmacılar hesaplamalı düşünmenin tanımı konusundaki görüşlerini makalelerde, çalıştaylarda ve komitelerde ifade etmiştir. Yapılan çalışmalarda 'hesaplamalı düşünme' kavramının anlamı tekrardan sorgulanmış, kavramı tanımlayan bileşenler açıklanmış ve bu kavramın nasıl öğretileceği tartışılmıştır.

Wing (2014), hesaplamalı düşünmeyle ilgili tanımını revize ederek, hesaplamalı düşünmenin "problemi formüle etme ve çözümünü/çözümlerini bir bilgisayarın, insanın veya makinenin etkili olarak gerçekleştirebileceği bir şekilde ifade etmesiyle ilgili düşünce süreçleri" olarak tanımlamıştır. Dolayısıyla problemin formülü, hesaplama adımları veya algoritma ile gösterilmiş; bir algoritma

tasarlandığında makinenin hedefi gerçekleştirmesi için modeli uygulayan herhangi bir makineyi kontrol etmenin bir yolu da tasarlanmıştır (Aho, 2012).

Hesaplama düşünme, yardım ettiği disipline ve ihtiyaç duyan kişiye göre farklı anlamlarla ifade edilebilmektedir. Bu bakımdan araştırılan problem, probleme yol gösteren model ve problemin çözümlerini hesaplayacak araç çeşitlilik gösterebilmektedir. Bu durum hesaplama düşünmenin tanımını zorlaştıracaktır. Alfred Aho (2012), bu belirsizlikleri önlemek için hesaplama düşünme ya da hesaplama kavramını, araştırılan problemle eşleşen, iyi tanımlanmış bir hesaplama modeliyle birlikte kullanılmasını önermiştir. Böyle bir durumda hesaplama düşünmenin tanımı kullanılan modele göre şekilleneceği vurgulanmış olmaktadır. İnsan, makine ve bilgisayar gibi bir problem çözücünün veya hedef gerçekleştiricinin, hedefi gerçekleştirebilmesi veya problemi etkili bir şekilde çözebilmesi için uygun hesaplama modelini bulması gerekmektedir. Bu anlamda algoritma kavramı da herhangi bir adım dizisi değil, soyut makineleri ve hesaplama modelini kontrol eden adımlar dizisi olarak algılanmaktadır (Denning, 2017b: 35).

Yeni hesaplama düşünme dönemiyle birlikte hesaplama düşünmeye yüklenen anlam değişmiştir. Geleneksel hesaplama düşünmede programlama becerisi hesaplama düşünmeyi sağlarken, yeni hesaplama düşünmede hesaplama düşünmeyi öğrenmek, programlama becerisini kazanmayı sağlamaktadır (Denning, 2009a, 2017b, 2017a; Denning, Tedre ve Yongpradit, 2017). Dolayısıyla yeni hesaplama düşünme kapsamlı programlama becerisine gerek duymayan herkesin günlük hayatında uygulayabileceği hesaplama adımları olarak görülmüştür (Denning, 2009a, 2017b, 2017a; Denning ve diğerleri, 2017). Geleneksel dönemde programlama deneyimi ve farkındalığı vurgulanırken yeni dönemde programlama deneyimi bir zorunluluk olmamıştır. Dolayısıyla algoritmik düşünmek için problemi zihinde adım adım tanımlayabilmek yeterli olmuştur.

1.2. Hesaplama Düşünme

Hesaplama düşünmeyi tanımlamak için birçok çalışma hazırlanmıştır. Bazı araştırmacılar ise hesaplama düşünmenin kesin bir tanımının gerekli olmadığını iddia

etmiştir (Hu, 2011; Voogt, Fisser, Good, Mishra ve Yadav, 2015). Ancak kesin olan şey, hesaplamalı düşünmenin anlamının ve kapsamının değiştiğidir. Geçmişten bugüne hesaplama ve hesaplamalı düşünmenin kat ettiği yol bunu göstermektedir. Hesaplamalı düşünme yeni dönemiyle birlikte hızlı bir şekilde yaygınlaşmıştır. Ancak hesaplamalı düşünmenin anlamı ve perspektifi hâlâ tartışmaya açıktır. Hesaplamanın ve hesaplamalı düşünmenin, başta bilgisayar bilimi ve matematik bilimi olmak üzere birçok bilim dalından beslendiği ve birçok farklı disipline uygulanabildiği görülmüştür (Anderson, 2016; Denning, 2009b; Wing, 2006, 2008, 2010).

1.2.1. Hesaplamalı Düşünmenin Bileşenleri

Hesaplamalı düşünmeyi anlayabilmenin en iyi yollarından biri olarak onun bileşenlerini algılamak görülmüştür. Hesaplamalı düşünmenin operasyonel tanımı hesaplamalı düşünmenin bileşenlerini algılamaktan geçmektedir. Birçok araştırmacı (Anderson, 2016; Angeli ve diğerleri, 2016; Barr ve Stephenson, 2011; Denner, Werner ve Ortiz, 2012; Guzdial, 2015; Kalelioglu, Gulbahar ve Kukul, 2016; Riley ve Hunt, 2014; Selby, 2013; Shute, Sun ve Asbell-Clarke, 2017; Tang, Yin, Lin, Hadad ve Zhai, 2020; Weintrop ve diğerleri, 2016; Wing, 2006, 2008; Yadav, Mayfield, Zhou, Hambruch ve Korb, 2014), saygın kuruluşlar (Csizmadia ve diğerleri 2015; CSTA ve ISTE 2011; ISTE 2016), çalıştaylar (National Research Council, 2010, 2011) hesaplamalı düşünmenin bileşenlerini tanımlamaya çalışmıştır. Tanımlanan bileşenlerde farklılıklar olsa da çoğunluk tarafından kabul edilen ortak noktaların olduğu da görülmektedir (Selby, 2013; Shute ve diğerleri, 2017). Bahsedilen bileşenler arasında ayrıştırma, soyutlama, algoritmik düşünme, genelleme, değerlendirme, veri analizi, veri toplama, veri temsili, örüntü tanıma, yineleme, özyineleme, otomasyon, sistem tasarımı, hata ayıklama, modelleme, simülasyon, görselleştirme, paralelleştirme, test etme gibi terimler vardır. Ayrıştırma, soyutlama ve algoritmik düşünme çoğunluk tarafından kullanılan ana bileşenlerdir. Genelleştirme ve değerlendirme terimlerinin ise literatürdeki kullanım ve anlam tutarlılığına dayanarak literatürdeki fikir birliğini yansıttığı söylenebilir (Selby ve Woollard, 2014). Bu anlamda ayrıştırma, soyutlama, algoritmik düşünme, genelleştirme ve değerlendirme

terimleri hesaplamalı düşünme sürecine atıf yapan ortak bileşenlerdir. Öte yandan probleme, modele göre bu bileşenlerden daha fazlası da kullanılabilir.

1.2.1.1. Ayırıştırma

Karmaşık problemleri bölmek ve bölünen parçaları da gerektiğinde daha küçük parçalara ayırarak çözmek süreci daha basitleştirecektir. Problemin parçalara bölünüp incelenmesi, problemi daha izlenebilir ve yönetilebilir hâle getireceği için süreç daha verimli gerçekleşecektir (Selby, 2013). Ayırıştırma ile problem parçalara bölünür ve parçalar birbirleriyle ilişkilendirilerek anlamlandırılır. Bu anlamda bu çalışmada ayırıştırmanın kullanıldığı söylenebilir. Araştırma problemi, başlıklara ve alt başlıklara bölünerek ayrı ayrı ve sistematik olarak ele alınmıştır.

1.2.1.2. Soyutlama

Soyutlama, hesaplamanın merkezinde yer alır ve hesaplamalı düşünmenin en önemli bileşenidir (Aho, 2012; Wing, 2008). Soyutlama karmaşık bir sistemin özüne ulaşır. Bir dizi nesnede ortak olan özelliklere odaklanmayı sağlar. Problem, sembolik olarak temsil edilir ve bir nesnenin birçok şeyin yerine geçmesine izin verilir. Böylece problemin çözümü kolaylaşır.

Grover ve Pea'ye (2018) göre soyutlama bilgi gizlemedir ve bu anlamda karmaşıklığı basitleştirmenin bir yoludur. Soyutlama, belli ayrıntıları göz ardı edip daha önemli görülen özelliklere odaklanmayı sağlar. Bir dizi talimat anlamına gelen bir program içinde tanımlanan her prosedür aynı zamanda bir soyutlamadır. Bu anlamda “bir algoritma, girdileri alan, bir dizi adımı yürüten ve istenen bir amacı karşılamak için çıktılar üreten bir sürecin soyutlamasıdır” (Wing, 2014).

Hesaplamalı düşünmenin bir bileşeni olarak soyutlama, daha çok bilgisayar bilimiyle anlamlandırılrsa da kavramın anlaşılması için daha basit örnekler de verilebilir. Örneğin bir iç mekânın geleneksel tarzda olduğunu söylediğinde soyutlama kullanılmış olur. Bu durumda, mekânın üslubu önemli olarak kabul edilirken mekânın hacmi, yapım yılı, mekânın işlevi gibi ayrıntılara yer verilmemiş olur. Başka biri mekânın işlevine ya da başka bir özelliğine önem verebilir ve farklı bir soyutlama

kullanabilir. Soyutlama, birden fazla somut örneğin o şartlar içerisinde işimize yarayan benzer özelliklerini bulmak ve kullanmak ile gerçekleşen bir işlemdir. Birden fazla döneme ait birbirinden farklı tasarımlarda sandalyeler olduğu düşünüldüğünde eğer bu sandalyelerin sadece oturulma özelliği olan işlevi bir isim olarak gösterilirse ya da bir temsil olarak gösterilirse soyutlama yapılmış olur.

1.2.1.3. Algoritmik Düşünme

Algoritmanın en yaygın tanımlarından biri, belirli bir hedefi gerçekleştirmek için izlenmesi gereken sıralanmış adımlar seti olmasıdır. Bir hedefe ulaşmak için ya da bir problemin çözümüne ulaşmak için önce o hedefi izleyen adımları tanımlamak gerekir. Adımların tanımlanması ve adımların birbirleriyle ilişkilerinin kurgulanması, algoritma tasarımını oluşturur. Algoritma genellikle bir insan tarafından tasarlanır. İnsan, yaratıcılığını ve zekâsını kullanarak talimat adımlarını yaratır. Daha sonra geriye sadece talimatları takip etmek kalır. Talimatların takip edilebilmesi için talimat adımlarının açık ve kesin olması gerekir. Talimatlar belirsiz ifadeler içermemelidir ya da yorum gerektirmemelidir.

Algoritmik düşünmek, problemin ya da hedefin adım adım planlanması gerektiğini vurgular. Ancak, algoritmanın temel işlem adımlarının bilgisayar tarafından uygulanabilir derecede kesin olmalıdır. Bu kesinlik derecesi, yeni hesaplamalı düşünme dönemi destekçileri tarafından “basit bir mutfak tarifi” düzeyine kadar indirgense de bir algoritma bilgisayarın onu çalıştırabileceği şekilde yazılmalıdır (Denning, 2017c: 7,8) Algoritma adımları bir insan tarafından anlaşılabilir ancak her işlem adımını bilgisayar algılayamaz. Bu anlamda bir algoritma insan muhakemesine gerek kalmayacak şekilde tasarlanmalıdır. Yani algoritma makine tarafından gerçekleştirilebilir olmalıdır. Her bir dizi talimat adımı ya da ‘insan tarafından yürütülebilen’ her bir tarif, bir algoritma değildir. Ancak bu, insanların bazı hesaplama adımlarını takip edebilmesi veya makine adımlarını planlaması gerekliliğini değiştirmez (Denning ve diğerleri, 2017).

Algoritmaları cezbedici kılan şey, tekrar eden hesaplama kalabalığı ile baş etmeyi sağlamasıdır. Benzer problemlerin tekrar tekrar çözülmesi gerektiğinde

algoritma devreye girmelidir. Böylece problemlerin her seferinde yeniden düşünülmesi gerekmez. Yani algoritma bir kez anlaşıldıktan sonra, her yeni problem için sıfırdan çalışılması gerekmez. Bir insan, bir bilgisayar gibi algoritma adımlarını takip edebilir. Ancak, hesaplama yükünün yüksek olduğu problemlerde, diğer bir deyişle tekrar eden görevlerin büyük işlem kalabalığı yarattığı problemlerde, makine ya da bilgisayarı kullanmak daha akıllıcadır (Wing, 2008). Çünkü hesaplama, makineler ya da bilgisayarların hesaplama kapasitesi sayesinde hızlanır. Böylece bir insanın elle çözmesinin çok zaman alacağı problemler daha kısa sürede çözülmüş olur.

Algoritma doğrulandıktan sonra algoritmanın problemi ne kadar hızlı çözebildiğine bakılır. Bu iki ölçütün dışında algoritmanın başarısı; yeni bilgiler çıkarılabilmesi ve insan zekâsının sınırlarını genişletebilmesi (Terzidis, 2006), başkaları tarafından anlaşılabilmesi ve değiştirilebilmesi yeteneği ile ilişkilidir (Jabi, 2013).

1.2.1.4. Genelleme

Genelleme, yeni sorunları ya da hedefleri hızlı bir şekilde çözebilmek için, sorunların ya da hedeflerin önceki yöntemlerinin, çözümlerinin veya süreçlerinin yeni problemlere uygulanabilirliğini sorgulamaktır. Yani genelleme ile belirli bir uygulamadan daha geniş bir uygulanabilirliğe geçmek amaçlanır (Selby, 2013). Bunun için çözümlerin ortak kısımlarını tanıyabilmek ve yeniden kullanabilmek gerekir. Bazı belirli problemleri çözen algoritmalar, bütün benzer problem sınıfını çözmek için uyarlandıktan sonra benzer bir problemle karşılaşıldığında bu genel çözüm uygulanabilir (Csizmadia ve diğerleri, 2015). Bu anlamda benzer bir amaç için daha önceden yazılmış ve doğruluğu kanıtlanmış bir algoritmayı bilmek zaman kazandırabilir (Jabi, 2013).

1.2.1.5. Değerlendirme

Değerlendirme, bir çözümün iyi olduğundan, yani amaca uygun olduğundan emin olma sürecidir. Hesaplamalı düşünme, analiz ve analitik düşünmeyi de içerir, ancak değerlendirme fikri daha büyüktür (Grover ve Pea, 2018). Sorunların çözümleri, istenen sonuca veya hedefe göre birçok açıdan değerlendirilir (Beecher, 2017).

Programlar, modeller, simülasyonlar biçimindeki algoritmalar amaca göre doğruluk ve uygunluk açısından değerlendirilmelidir.

1.2.2. Hesaplamalı Düşünme Araçları ve Uygulamaları

Modelleme, simülasyon, görselleştirme, otomasyon, kodlama ve programlama hesaplamalı düşünmenin uygulanabilmesinde önemli önemli araç ve uygulamalar olabilir.

1.2.2.1. Modelleme, Simülasyon ve Görselleştirme

Modelleme, simülasyon ve görselleştirme terimleri hesaplamalı düşüncemizi somutlaştıran yardımcı araçlardır. Modelleme, simülasyon ve görselleştirme ile hesaplamalı düşüncelerimizin temsilleri manipüle edilebilir. Değişkenlerin değerlerini değiştirerek sonuçlar üzerinde daha etkili kararlar verebiliriz veya sınırlılıkları daha kolay algılayabiliriz. Böylelikle hesaplamalı düşünme ve tasarlama süreci daha hızlı bir şekilde gerçekleşebilir. Modeller, simülasyonlar ve görselleştirmelerin manipülasyonu hesaplamalı düşünme becerilerinin gelişimine katkıda bulunur. Dolayısıyla bu terimler hesaplamalı düşünmeyi kolaylaştıran araçlardır (Wing, 2006), hesaplamalı düşünme için zorunlu bir gereklilik değildir.

1.2.2.2. Otomasyon

Otomasyon, soyutlamaları yorumlamak ve tekrarlayan işlemleri yapabilmek için bir tür bilgisayar veya makineden yararlanmayı ifade eder (Wing, 2008). Otomasyona imkân veren düzeni kurmak hesaplamalı düşünme süreci içerisindedir, ancak otomasyon ile vurgulanmak istenen bu değildir. Otomasyon ile vurgulanmak istenen otomasyonun tekrarlanabilir “otomatik eylem” olmasıdır (Selby ve Woollard, 2014). Bu anlamda otomasyon hesaplamalı düşünme için zorunlu bir gereklilik olmayabilir. Çünkü otomasyon bu anlamıyla genellikle dijital bilgi işlem aracının faaliyetiyle özdeşleştirilir ancak hesaplamalı düşünme bir insan tarafından tekrarlanabilen işlemleri de kapsar.

1.2.2.3. Kodlama ve Programlama

Programlama, soyutlama ve ayrıştırma gibi bilgisayar bilimi kavramlarını kullanarak problemin çözülmesini sağlar, yani bireyi hesaplamalı düşünmeye maruz bırakır (Lye ve Koh, 2014). Programlamanın bir parçası olarak kodlama, gereksinimlerin çevrilmesi ve makine tarafından anlaşılabilir sözdizimine dönüştürülmesini içeren bir hesaplamalı düşünme uygulamasıdır. Grover ve Pea'ye (2013: 2,3) göre “programlama, yalnızca bilgisayar bilimlerinin temel bir becerisi ve hesaplamalı düşünmede yer alan bilişsel görevleri desteklemek için anahtar bir araç değil, aynı zamanda hesaplamalı yeterliliklerin de bir göstergesidir.”

1.2.3. Hesaplamalı Düşünmenin Diğer Düşünme Türleri ve Süreçleri ile İlişkileri

Hesaplamalı düşünme, farklı alanları ilgilendirdiği için birçok düşünce türü ile örtüşür (National Research Council, 2010; Grover ve Pea, 2018; Wing, 2006, 2008, 2010). Bazıları tarafından diğer düşünme türlerinden üstün görülse de bu düşüncenin bir temeli yoktur (Denning, Tedre ve Yongpradit, 2017: 33).

1.2.3.1. Hesaplamalı Düşünmenin Problem Çözme Süreciyle İlişkisi

Literatürdeki hesaplamalı düşünme tanımlarına bakıldığında genel fikrin, hesaplamalı düşünmenin, problemleri anlama ve çözme üzerine olduğudur (Aho, 2012; Barr, 2016; National Research Council, 2010, 2011; CSTA ve ISTE, 2011; Liu ve He, 2014; Wing, 2006, 2008, 2010, 2014). Yapılan hesaplamalı düşünme tanımları genellikle hem hesaplama kavramları hem de problem çözme becerileriyle ilgili unsurlar olarak görülür (Tang ve diğerleri, 2020). “Problem çözme” ifadesi içerisinde belirsizlikler barındırarak kafa karışıklığına neden olabilir (Selby, 2013). Her problem çözme süreci hesaplamalı düşünme olarak görülmeyebilir. Hesaplamalı düşünmenin kavramları, uygulamaları ve yöntemleri vardır. Hesaplamalı düşünme, soyutlama ve ayrıştırma ile problemin tanımlandığı, tercihen verilerin toplanması ve analizi ve örüntü tanınması ile incelendiği; çözümlerin algoritmik süreçler ile planlandığı, test etme, hata ayıklama ve genelleme ile değerlendirildiği; sürecin tercihen otomasyon, modelleme ve simülasyon ile desteklendiği bir problem çözme sürecidir.

Çizelge 1.1. Problem çözme süreci ve hesaplamalı düşünme süreci (Kalelioglu ve diğerleri, 2016).

Problem Çözme Süreci	Hesaplamalı Düşünme Süreci
Problemi tanımlama	Soyutlama, ayrıştırma
Veri toplama ve temsil etme	Veri toplama, veri analizi, kavramsallaştırma, örüntü tanıma, veri temsili
Problem formüle etme, çözümü planlama seçme ve üretme	Algoritmik süreçler, eş zamanlı çalışma, yineleme, genelleme
Çözümü uygulama	Otomasyon, modelleme ve simülasyon
Çözümü değerlendirme, iyileştirme ve geliştirme	Test etme, hata ayıklama, genelleme, veri analizi, değerlendirme

Ayrıca hesaplamalı düşünmeyi sadece “problem çözme” yöntemi olarak ifade etmek uygun olmayabileceği düşünülmüştür. Denning (2017b) ve Nardelli (2019), “problem çözme” ifadesinin dar bir bakış açısı olduğunu iddia etmiştir. Denning’e göre (2017c: 5) “problemler” yerine “endişeler” terimini kullanmak daha uygun olabilir, çünkü hesaplamayı problem olarak görülmeyen görevler için sıklıkla kullanılır. Nardelli (2019), “problem çözme” ifadesi yerine “bir durumu modelleme” ifadesini kullanmıştır. “Bir problemi, çözümünün kişinin belirli bir hedefe ulaşmak istediği bir durumun sadece bir örneği” olduğunu düşünmüştür.

1.2.3.2. Hesaplamalı Düşünmenin Mantıksal Düşünmeyle İlişkisi

Mantıksal düşünmeye anlamını veren “mantık”, doğru ile yanlış arasındaki ayrımı yaparak doğru düşünme biçimlerini gösteren disiplindir. Mantık; felsefe, matematik, bilgisayar bilimi gibi birçok bilimin parçasıdır. Mantıksal düşünmek, mantıksal düşünmeye temel oluşturan mantığı öğrenerek bu mantığa göre doğru sonuçlara ulaşmak veya daha kesin bilgileri elde etmektir (Başarer, 2017; Başarer ve Duman, 2018).

Mantıksal düşünebilen yani problem çözebilen kişilerde bulunması gereken birtakım özellikler; yaratıcılık, kurgulama yeteneği, gözlemlene yeteneği, sayısal yetenekler, pratik beceriler ve iletişim becerileri olarak belirlenmiştir (Temizyürek, 2003: 72-74). Bilgiyi bulma, toplama, sınıflama, sıralama, yorumlama, değerlendirme ve zamanın iyi kullanımı gibi becerilerin gerekliliğinden bahsedilmiştir (Bozdoğan, 2007).

Öte yandan mantıksal düşünme, özellikleri itibariyle hesaplamalı düşünme ile de ilişkilendirilebilir. Mantıksal düşünmenin bazı özellikleri, örneğin zihinsel işlemler yaparak bir problemi çözmek, sayıları etkin kullanmak, durumlar arasındaki ilişkileri ayırt etmek (Bozdoğan, 2007; Demirel, 2003; Karplus, 2003; Kılıç ve Sağlam, 2009; Lawson ve Thompson, 1988; Lazaer, 2000), aynı zamanda hesaplamalı düşünmenin de özellikleridir. Diğer yandan, soyutlama, ayırıştırma, veri toplama, veri analizi, genelleme, test etme, değerlendirme, gibi kavramlar mantıksal düşünme süreçlerinde de kullanılır (Bozdoğan, 2007; Demirel, 2003; Temizyürek, 2003). Hatta bazı araştırmacılar mantıksal düşünmeyi hesaplamalı düşünme konsepti içerisinde değerlendirmiştir (Grover ve Pea, 2018; L'Heureux, Cohen, Boisvert ve Sanghera, 2012; Lockwood ve Mooney, 2017; Voogt ve diğerleri, 2015; Yadav ve diğerleri, 2014).

Hesaplamalı düşünme beceri ölçeği için problem çözme, yaratıcılık, algoritmik düşünme, eleştirel düşünme, iş birlikli öğrenme gibi bileşenler tanımlanmıştır (Korkmaz, Çakir ve Özden, 2017; Yağcı, 2019). Sıralama, tamamlama, hata ayıklama gibi görevlerin olduğu bir hesaplamalı düşünme beceri testi oluşturulmuştur (Román-González, 2015). Mantıksal düşünme becerilerinin tespitinde ise değişkenleri kontrol etme, orantısal düşünme, ilişkisel düşünme, olasılıklı düşünme mantıksal düşünme becerileri ölçek bileşenleri olarak belirlenmiştir (Roadrangka, 1983; Tobin ve Capie, 1981). Bu beceri ölçeği bileşenleri mantıksal düşünmede başarılı olmak için gerekli işlemler olarak görülmüştür (Bozdoğan, 2007; Kılıç ve Sağlam, 2009; Lawson, 1982; Saygıner, 2017; Valanides, 1996). Öte yandan Byeongsu Kim ve diğerleri (Kim, Kim ve Kim, 2014: 443), hesaplamalı düşünmeyi her insanın sahip olduğu bir tür mantıksal düşünme olarak tanımlamış, bireylerin hesaplamalı düşünme becerilerinin mantıksal düşünme beceri testi kullanılarak ölçülebileceğini öne sürmüştür.

Çalışmalarda görüldüğü üzere hesaplamalı düşünürken aynı zamanda mantıksal düşünülür. Ancak mantıksal düşünülmediğinde hesaplamalı düşünmüş olunmayabilir. Mantıksal düşünme, hesaplamalı düşünmeye göre daha olması gerekene yani doğruluğa odaklanır. Algoritmik düşünmenin hesaplamalı düşünme ile özdeşleştiğini ve hesaplamalı düşünme becerilerinin ölçülmesine en yakın

kısımlardan biri olduğu düşünülürdüğünde (Román-González, Pérez-González, Moreno-León ve Robles, 2018: 55), hesaplamalı düşünmenin, mantıksal düşünmeye göre öne çıkmış farklı bir tarafı olduğu görülür. Hesaplamalı düşünme sonucunda bir dijital araç ya da bir insan tarafından çözülebilecek algoritmalar yaratılır. Yani, hesaplama düşünmenin sonucunun bir insan veya dijital araç tarafından uygulanabilir olmasının gerekliliği onu mantıksal düşünme ve matematiksel düşünme gibi diğer alanlardaki geçerli terimlerden ayırır (Selby ve Woollard, 2014: 14).

1.2.3.3. Hesaplamalı Düşünmenin Eleştirel Düşünmeyle İlişkisi

Eleştirel ve mantıksal düşünme birbirlerine benzerdir. İki düşünme türü de mantıksal sorgulamayı kullanır. Eleştirel düşünme, mantıksal düşünme gibi doğru ve makul sonuçlara ulaşmayı amaçlar (Başarer, 2017). Ancak mantıksal düşünmede doğru düşünme kurallarının ve formlarının bilgisi üzerinden hareket edilirken, eleştirel düşünmede aynı zamanda doğru olmayan düşünme biçimlerini de tanımaya, bunların sonuçlarını algılamaya, kanıtlara ve sonuç ürününe ulaşmada kullanılan tüm düşünme işlemlerine odaklanılır (Başarer, 2017; Gürkaynak, Üstel ve Gülgöz, 2003).

Eleştirel düşünürlerin de hesaplamalı düşünürler gibi soyutlama, genelleme, modeller oluşturma gibi yetkinlikleri gereklidir. Ancak eleştirel düşünmenin hesaplamalı düşünme ile benzerlikleri bulunduğu gibi farklılıkları da bulunur. Eleştirel düşünme, beşeri bilimler ve sosyal bilimlerdeki terimlerle tanımlanma ve soyutlanma eğilimindeyken, hesaplamalı düşünme tanımları tipik olarak bilgisayar biliminin kavramsal sözlüğünden yararlanır (Walden, Doyle, Garns ve Hart, 2013).

1.2.3.4. Hesaplamalı Düşünmenin Analitik Düşünmeyle İlişkisi

Analitik düşünme, bir problemi ya da nesnelere ayrıştırarak ele almak, parçaları anlamlandırmak ve sistemin çalışmasının sağlamak amacıyla parçaların birbirleriyle etkileşimini incelemek, parçaları karşılaştırmak, değerlendirmektir (Amer, 2005; L. W. Anderson ve diğerleri, 2000; Sternberg, 2002, 2006).

Analitik düşünmede; hesaplamalı düşünmede de olduğu gibi problemler ayrıştırılarak çözülür. Analitik düşünme, doğru analiz yapmayı gerektirir. Analiz

terimi; soyutlama, ayırıştırma, algoritmik tasarım ve değerlendirme kavramlarını içeren çok geniş bir kavram olarak karşımıza çıkabilmektedir (Selby ve Woollard, 2014: 19). Bu kavram, hesaplamalı düşünmenin temel terimlerini içerir. Bu açıdan analitik düşünme ile hesaplamalı düşünmenin önemli ortak noktalarının olduğu görülür.

1.2.3.5. Hesaplamalı Düşünmenin Sistemli Düşünmeyle İlişkisi

Sistemli düşünme, geniş bir bakış açısı ile problemin farklı bağlantıları arasındaki ilişkilerine ve değişime odaklanan, parçaları bir bütün olarak ele alan bir yaklaşımdır (Senge, 1996; Stave ve Hopper, 2007; Tecim, 2004).

Sistemli düşünme yetisine sahip bir kişi problemin ya da sistemin sınırlarını tanıyabilir, modelleyebilir, modeli test edebilir ve ona göre kararlar alabilir (Shute ve diğerleri, 2017). Bu açıdan düşünüldüğünde sistemli düşünürün bir hesaplamalı düşünür ile ortak noktaları bulunur. Ancak hesaplamalı düşünme sistemli düşünmeden daha fazlasıdır. Dolayısıyla hesaplamalı düşünme, sistemli düşünmeyi kapsar.

1.2.3.6. Hesaplamalı Düşünmenin Matematik ve Bilgisayar Bilimi ile İlişkisi

Matematik, soyutlama ve basitleştirilmiş modellerle problem çözmeyi içerir. Bilgisayar bilimi, matematikçilerin problemin çözümünü otomatikleştirme ve hesaplama yükünü makinelerle yüklemeye çabalarından doğmuştur (National Research Council, 2010; Denning, 2009b). Bilgisayar biliminin tarihi ve temeli, matematiğe ve bilime dayanır. Matematik biliminde problemin cevabı denklemler ile çözülür. Bilgisayar biliminde hedefi gerçekleştirmek üzere hazırlanmış algoritmalar ve bu algoritmaları çözen makineler vardır. Algoritmalar yeni değildir, bilgi çağının başlangıcından önce de kullanılmıştır. Bilgisayar biliminin asıl ayrıcalığı, problemlerin elektronik ve otomatik bir hesaplayıcı olan bilgisayarlarla buluşturmasıdır (Denning ve Tedre, 2021). Böylece problemler güvenilir ve hızlı bir şekilde çözülebilir. Problemi çözmek için alınan talimatları otomatik olarak uygulayan bir bilgi işleyici olması, bilgisayar bilimini matematik biliminden ayrı kılar. Diğer yandan bilgisayarlar, onlara tanınan işlem gücü kadar hesaplama yapabilirken matematik sınırsız işlem temsilleri kapsar (National Research Council, 2010).

Hesaplamaalı düşünme, hem matematiksel düşünme ile hem de bilgisayar bilimi ile yakından ilişkilidir (National Research Council, 2010; Denning, 2009b, 2010; Shute ve diğeri, 2017; Wing, 2008). Hesaplamaalı düşünme, bir problemi çözmeye yaklaşılan genel yollarda matematiksel düşünmeden yardım alır. Gerald Sussman, matematiksel düşünmenin soyut yapılar etrafında döndüğünü, hesaplamaalı düşünmenin ise soyut metodolojiler etrafında döndüğünü ifade etmiştir (National Research Council, 2010). Hesaplamaalı düşünme, matematiksel düşünmeye birçok açıdan benzer olsa da temelde bilgisayar, makine veya insanın fiziksel kısıtlamalarını dikkate alır.

Birçok araştırmacı, hesaplamaalı düşünmeyi, bilgisayar bilimcisi gibi düşünebilmeyi temel alan bir süreç olarak tanımlamıştır (Mannila ve diğeri, 2014; Riley ve Hunt, 2014; Steve Furber, 2012; Wing, 2006, 2008, 2010, 2014). Hesaplamaalı düşünme, belirli esaslarda bilgisayar bilimiyle büyük ölçüde örtüşür. Ancak özellikle yeni hesaplamaalı düşünme dönemiyle birlikte bilgisayar biliminden farklı olarak, problem çözüme yaklaşımlarını geliştirmeye odaklanmıştır.

Hesaplamaalı düşünür, bilgisayar bilimlerinden gelen temel kavramlardan yararlanır ancak bunu sadece bilgisayar bilimcisi gibi düşünerek yapmaz. Hesaplamaalı düşünmenin çevresi ya da bileşenleri yakından incelendiğinde bu kavramların sadece bilgisayar bilimi alanına ait olmadığı görülebilir (Selby 2013; Ubiquity 2007). Sadece bilgisayar bilimcilerine ait olduğu düşünülen temsiller oluşturma, veri analizi, hata ayıklama gibi kavramlar hesaplamaalı düşünme unsurlarının birçok disiplininde zaten paylaşılmıştır. Hesaplama araçlarının etkin bir şekilde kullanımını sağlayan yeni düşünce biçimleri sadece bilgisayar bilimlerinde değil her alanda ortaya çıkmıştır (Barr ve Stephenson, 2011; National Research Council, 2010; Hemmendinger, 2010). Hemmendinger'e göre (2010: 6), insanların bilgisayar bilimcisi gibi düşünmeye yönelmesi yerine kendi meslek disiplinlerine göre düşünmeye yönelmesi gerekir; bu da kendi alanlarındaki problemleri çözmek için hesaplamaı kullanmakla, verimli bir şekilde keşfedilecek soruları bulmak ile olur.

Hesaplamaalı düşünme bileşenlerini düşünce süreçlerine eklenirken bilgisayardan yardım alınması zorunlu değildir. Yani hesaplamaalı düşünür olmak için

bilgisayarlar kullanılmak zorunda değildir (Wing, 2008), bilgisayarlar düşünce süreçlerini geliştirmek, problem çözümlerini bulmak için bir tercihtir. Ancak bu tercihi kullanmamak eksiklik yaratabilir. Hesaplamayı etkili kılan bilgisayardır. Bilgisayarlar düşünce süreçlerinde keşif aracı olarak ve düşünce yürütebilmede bir yardımcı olarak iyi bir işbirlikçidir. Bilgisayarlar ile bilgileri yüksek performanslı grafiklerle görüntüleyerek hesaplamalı düşünmenin gelişimi artırılabilir, bilgisayarın hesaplama kapasitesi ile harcanan zamandan ve emekten tasarruf edilebilir.

1.2.3.7. Hesaplamalı Düşünmenin Mühendislik ve Mimari Tasarım Düşüncesi ile İlişkisi

Hesaplamalı düşünme, “gerçek dünyanın karmaşık kısıtlamaları içinde işleyen büyük karmaşık sistemi tasarlar ve değerlendirirken mühendislik düşüncesinden yardım alır.” (Wing, 2008). Ancak, hesaplamalı düşünürler mühendislik disiplinlerinden farklı olarak fiziksel gerçeklik tarafından kısıtlanmayan sanal dünyalar inşa edebilir (National Research Council, 2010; Wing, 2010). Böylece problem çözümüne hayal güçlerini ve bilgisayarın hesaplama gücünü katabilirler. Yazılımlar ve programlar kullanarak zihinlerinin sınırlarını zorlayabilirler. Temelde, soyutlamalarda matematikten, hesaplamalı düşünmenin gelişimini arttırmada ve çözümleri otomatikleştirmede bilgisayar biliminden, sistemleri gerçek dünyada çalıştırabilmede ise mühendislik düşüncesinden yararlanılır.

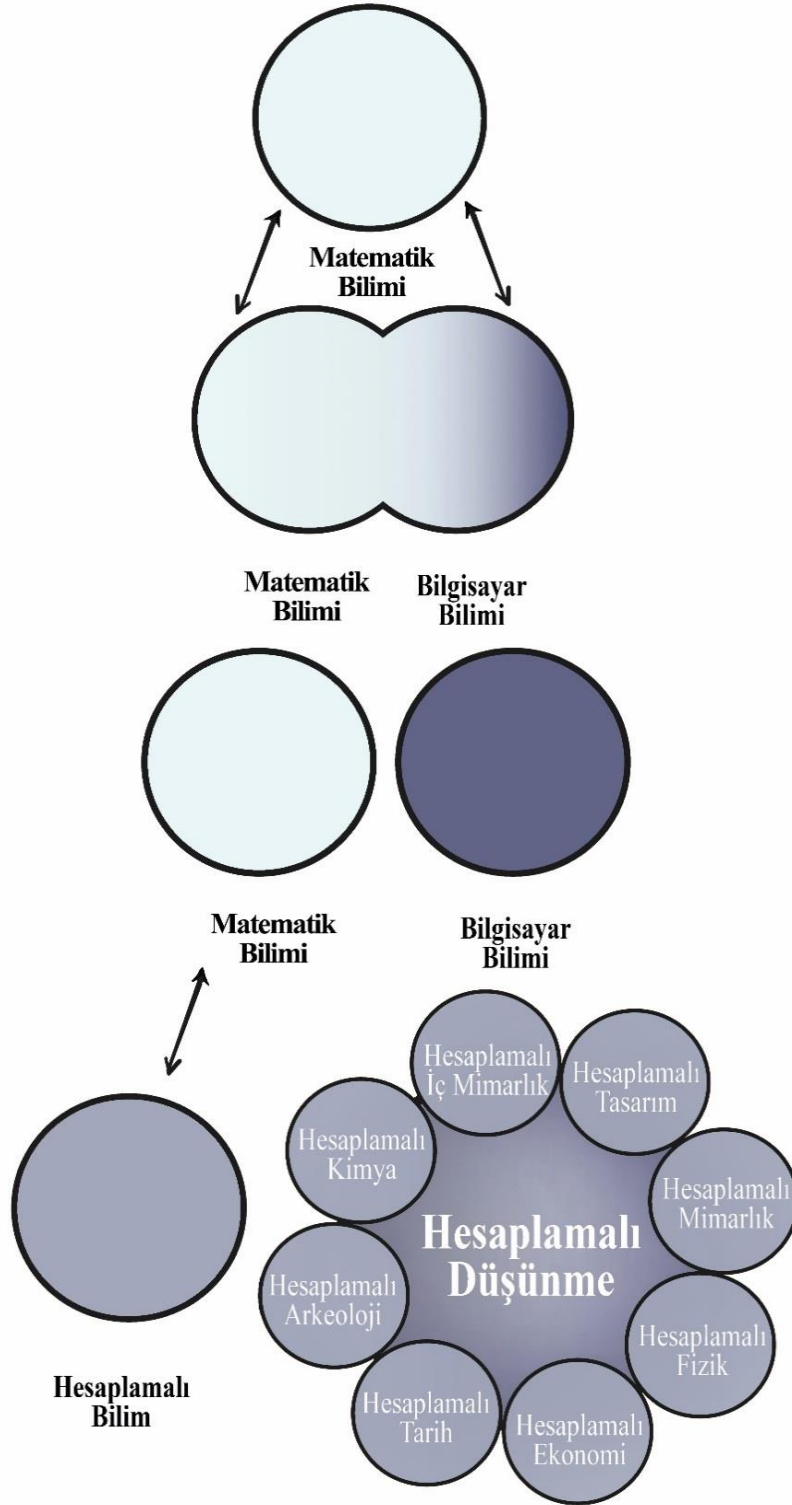
Wing (2006), hesaplamalı düşünmenin bir çözüm tasarlamak için buluşsal akıl yürütmeyi içerdiğini belirtir. Buluşsal düşünülerek de problem adımlar halinde çözülebilir ancak genellikle algoritmik ifadedeki gibi sorunun çözümü garanti edilmez. Tasarım odaklı düşünme, problemlerin pratik, yaratıcı çözümlerine odaklanır. Yaratıcı düşünme, alınan bilginin hayal ve zihin gücünü kullanarak yeniden şekillendirilmesi ve fikirlerin değiştirilmesi sürecidir (Bentley, 1996; Feldhusen, 1995). Bireylerin yaratıcılık seviyeleri ölçüsünde (Runco, 1996), problemlere karşı duyarlılık gösterip çözüm yolları araması, hipotezler geliştirmesi (Torrance, 1968) ve özgün fikirler ortaya koymasındır (Oğuzkan, 1974: 207). Bu bakımdan hesaplamalı düşünme ile ortak yanları bulunur.

Mimari tasarımda hesaplamalı yaklaşımlar, mimari tasarımcıların karmaşık problemlerini, hesaplamalı düşünme becerileriyle ve gelişmiş hesaplama araçlarıyla daha bilinçli bir şekilde çözebilmelerini sağlamıştır. “Mimari bilginin kurgusunun yeniden incelenmesi ve yeniden modellenmesi arayışında hesaplamalı düşünce, tasarımcı ve incelenen obje arasında iletişime ve bilgi aktarımına olanak sağlayan arayüz olarak davranır” (Erdoğan ve Gönenç Sorguç, 2011).

1.3. Farklı Disiplinleri Kapsayan Bir Kavram Olarak Hesaplamalı Düşünme

Hesaplamalı düşünme, hem insanlık hem de makineler tarafından kullanılan farklı disiplinler arasında geçerli düşünme yollarını ve yöntemlerini inceleyen evrensel bir şemsiye terim olarak görülebilir. Bu anlamda, farklı disiplinler arasında bilgi alışverişini de olanaklı kılmıştır. Hesaplamalı düşünme, çoğu bilim ve meslek disiplinlerini çoktan etkilemiştir ve etkilemeye devam edecektir (Beecher, 2017; Denning ve Tedre, 2021; Lee ve Malyn-Smith, 2020; Malyn-Smith ve diğerleri, 2018; Wing, 2014). Bu durum hesaplamalı yöntemlere, teknolojilere yatırım yapılmasının önemini ortaya koymaktadır. Teknolojinin gelişmesi ile hesaplamalı düşünmenin araçları, yöntemleri ve biçimleri değişmiştir. Dolayısıyla kavrama yüklenen anlam da değişebilir. Bu anlamda teknolojik olarak güncel kalmak önemli görülmüştür. Lee ve Andrew McGettrick, karmaşıklığı yönetmek için teknolojik araçlardan yararlanmak gerektiğine, bunun için de teknolojik araçlara hâkim olmak gerektiğine işaret eder (National Research Council, 2010: 11).

Matematik biliminin ihtiyaçlarından bilgisayar bilimi doğmuştur. Bilgisayar biliminden ayrı bir evrimle hesaplamalı bilimler ortaya çıkmıştır. Farklı ihtiyaçlardan hesaplamalı disiplinler evrilmeye devam etmektedir. Bu bölünmeye daha birçok farklı disiplin eklenebilir. Çeşitli disiplinlerde karşılaşılan hesaplamalı teoriye bağlı değişim, günümüzde mimari tasarım alanında da hissedilmiştir. Hesaplamalı düşünmenin birçok yerde kullanılması, doğrudan veya dolaylı olarak mimari tasarımcıları da etkiler.



Şekil 1.1. Matematik biliminden, bilgisayar bilimine, bilgisayar biliminden hesaplamalı bilimlere, hesaplamalı bilimlerden hesaplamalı düşünmeye ve hesaplamalı alanlara doğru genişleyen ve gelişen sürecin temsili.

İKİNCİ BÖLÜM

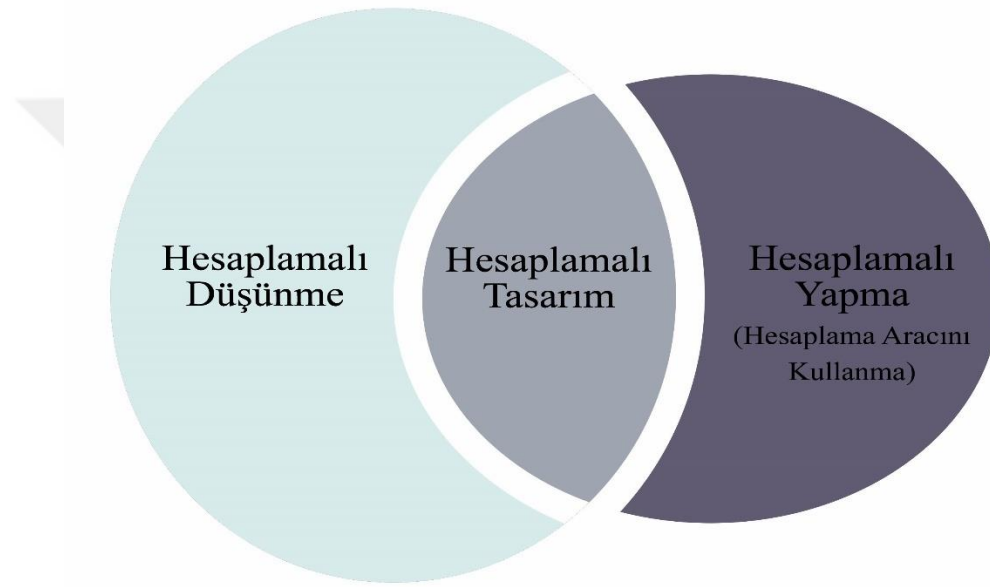
HESAPLAMALI MİMARİ TASARIM

2.1. Hesaplamalı Mimari Tasarım

Tasarım kelimesiyle genelde iki anlam vurgulanır: bir süreç ve bitmiş bir ürün (Lawson, 2005: 3). Hesaplamalı tasarım kavramıyla vurgulan hem bir eylemdir, hem de bir araçtır. Hesaplamalı tasarım, tasarımı ortaya çıkaracak tasarım sürecinin, aracının ya da yönteminin tasarımıdır. Dolayısıyla hesaplamalı tasarım, sadece sonuç ürünü değil aynı zamanda tasarım sürecini tasarlamayı amaçlar (Çolakoğlu ve Yazar, 2007; Harding, 2014; Kolarevic, 2005; Peters, 2013; Rucker, 2006; Sorguç ve Selçuk, 2013; Soltani ve diğerleri, 2020; Terzidis, 2006). Dolayısıyla, alanında uzman bir kişi başkalarının ya da kendisinin hesaplama yapabileceği ve hesaplamanın sonucunda ürün(ler) ortaya çıkarabileceği ya da bir amacını gerçekleştirebileceği bir araç, yöntem ya da prosedürü oluşturursa hesaplamalı tasarım yapmış olur.

Hesaplamalı tasarım yapmak, hesaplamalı düşünebilmeyi gerektirir (Denning, 2017c; Erbaş, 2015; Erdoğan ve Gönenç Sorguç, 2011; Oktan ve Vural, 2021; Weinan, 2020). Hesaplamalı tasarım, hesaplamalı düşünmenin bir alt kümesidir ve yapmak ile düşünmenin bir kesişimi olarak ifade edilebilir (Denning, 2017c: 5). Dolayısıyla bir hesaplamalı tasarımcı olabilmek için hesaplamalı düşünür olmak gerekir (Şekil 2.1.). Ancak, hesaplamalı tasarım araçlarını kullanmak için, hesaplamalı düşünme zorunluluğu yoktur. Örneğin, geleneksel bilgisayar destekli tasarım (CAD) aracını, bir mimari tasarımcının kullanımı için programlayan yazılımcı, bir hesaplamalı düşünürdür, ancak o programı kullanan başka bir tasarımcı hesaplamalı düşünür olmayabilir. CAD aracı, hesaplamalı bir düşünme süreciyle elde edilmiş araçtır, dolayısıyla bir ürün olarak hesaplamalı tasarımdır. Ancak bir hesaplamalı tasarım aracı değildir. Yani bunu kullanan tasarımcı, kendisi ya da başkası için hesaplamalı araç yaratamaz ve genellikle hesaplamalı düşünmez. Çünkü CAD aracının basit kullanımı, bir bilgisayara bilgi girme veya depolama eyleminden başka bir şey değildir. Dijital araçların bu şekilde kullanımı, tasarımcının sadece zihninde tasarladıklarını

dijitalleştirdiği bir faaliyettir. Serbest biçimli basit geometrilerin oluşturulduğu bu mimari modelleme araçları ile üç boyutlu görüntüler üzerine sadece basit manipülasyonlar uygulanabilir. Bu şekilde kullanım, bilgisayarın “elektronik çizim tahtası” gibi kullanımına benzetilmiştir (Oxman, 2006) ve bu “bilgisayarlaşma” (Terzidis, 2006) olarak da bilinmiştir. Kotnik’e (2006, 2010) göre bu düzeydeki dijital tasarım, bilgisayarların sunduğu potansiyelin boşa harcanmasıdır; hesaplamalı düşünmek ve hesaplamalı tasarım yapabilmek için yeterli farkındalıktan uzaktır.



Şekil 2.1. Hesaplamalı tasarım, hesaplamalı düşünmenin ilişkisi

Kaynak: (Denning, 2017c: 5).

Aslında geleneksel CAD araçlarındaki üç boyutlu görüntüler kendi içinde köşeler, yüzeyler, kenarlar gibi hiyerarşik ilişkilere sahip algoritmalarıdır. Bu algoritmalar arka planda formların matematiksel açıklamalarını çalıştırır. Bunların bütünü, yazılımcının bilgisayara tanımladığı numerik hesaplamalarının ürünü olsa da onu kullanan tasarımcı bunlardan habersiz olabilir.

Hesaplama ve tasarım sözcükleri tanımı zor olan ve kesin olmayan kavram olarak görebilir. Hesaplama sözcüğünü tasarım sözcüğü ile ilişkilendirdiğimizde anlamı daha da zorlaşır. Hesaplamalı tasarımdaki ‘hesaplama’, sayısal ve

matematiksel çağrışımlarından fazlasını ifade eder. Mimaride hesaplama ve tasarım kelimeleri bir araya geldiğinde, ayrı ayrı düşünüldüğünden farklı bir anlama ve keşfedilmeye açık bir hâle bürünebilir. Bu ucu açık keşif, mimari tasarımdaki hesaplamalı uygulamalarının tarihi, gelişimi ve potansiyeli algılanabildiği ölçüde daha verimli hâle gelebilir.

2.2. Hesaplamalı Mimari Tasarımın Tarihi

Mimari tasarımcıların çizimlerinde kullandıkları kâğıt, çizim gereçleri, cetvel gibi araçlar, yüzyıllardır fikirlerini organize etmek ve tasarımlarını geliştirmek için kullandıkları yardımcılardır. Tasarımda istenen formu araştırmak ya da sonucu bulmaya yönelik girişimler genellikle geleneksel çizim ile halledilmiştir. Hesaplamalı mimari tasarım, bu tasarım süreçlerine farklı bir anlayış getirmiştir.

Henüz yeterli teknolojinin olmadığı ya da dijital teknolojilere erişimin güç olduğu zamanlarda bazı öncü isimler bugün hesaplamalı mimari tasarım olarak tanımlanan alanda önemli ilk çalışmalara imza atmışlardır. Hesaplamalı üretken modeller olarak tanımlanabilecek ilk çalışmalarda, mekânları betimleyen şekiller, ızgaralar, eksenler ve inşaat çizgileri gibi soyut organizasyon araçları kullanılmıştır (Kolarevic, 1993: 51-53). Bu araçlar yalnızca şekillerin betimlenmesi için değil, aynı zamanda bir çizimin yani modelin davranışını düzenleyerek yeni ürünler üretmiştir (Kolarevic, 1993: 51-53). Bu ilk mimari üretici sistemler Leonardo da Vinci'nin merkezi planlı kiliselerin üretilmesi için hazırladığı üretken sisteme ve hatta Aristoteles'in çalışmalarına kadar uzanır (Mitchell, 1977: 29-35). Durand'in 1821 yılındaki mimari kompozisyonlar yaratan aracı daha yakın bir üretken modelden biridir (Ei-Khaldi, 2007: 12-18). 19. yüzyılın sonlarında mimari tasarımda mimari formlar analog araçlar ile bulunmuştur. Parametrik mimari tasarım uygulamalarının kökenlerini oluşturan bu uygulamalar, dinamik kuvvetlerin kendi kendine optimize edilmiş yeni mimari formları nasıl şekillendirebileceğini göstermiştir.

Gaudi'nin Colonia Güell Şapeli'nin tasarımı için icat ettiği araç (Asılı zincir modeli), mimari formların parametrik ilişkilerinin yapısal kuvvetlere göre incelemesini sağlamıştır (Jos, 1989). Bu modelin daha önceki erken üretken

sistemlerden farkı, parametrik ilişkilere (kısıtlamalara) dayanması ve bu parametrik sonuçların elle değil otomatik olarak hesaplanabilmesi olmuştur. Bu tasarımcılar dijital bir makinenin çalıştırabileceği bir algoritmayı oluşturmamalarına rağmen bir insanın gerçekleştirebileceği şekilde hesaplama adımlarını planlamış; soyutlama, ayrıştırma, genelleme ve değerlendirme ile hesaplamalı düşünce sürecine girmişlerdir.

Desmond Bernal ve etrafındakilerin önderliğinde 1930’lu yıllarda İngiltere’de başlayan avangart hareketi, bilim ve sanat yoluyla açığa çıkan bilginin kendi alanında sıkışıp kalmasının ve dolayısıyla halka ulaşamamasının önüne geçmeyi amaçlayan “üçüncü kültür” adlı bir yapıyı ortaya çıkarmıştır (Rocha, 2004). Farklı disiplinlerdeki kişileri bilimsel araştırma düşüncesi ve ortak hedefler etrafında toplayan bu yapı, mimarinin hesaplamalı dijital teknolojilerle buluşmasına zemin hazırlamıştır. Mimaride hesaplamalı dijital teknolojiler Leslie Martin, Nicholas Negroponte, Peter Eisenman, Herbert Simon gibi öncü isimlerin liderliğinde önemli yol kat etmiştir (Rocha, 2004).

1960 ve 1970’lerdeki yayınların temel konuları; tasarım hesaplamasının teknolojik gelişimi, bu tür teknolojilerin mimari uygulamaları ve yapay zekâ gibi mimari alanla ilişkili çalışmalar olmuştur (Koutamanis, 2005: 631). Mimari tasarımdaki bu ilk hesaplamalı dijital teknolojiler işlevsel bir yaklaşımla ve grafikten yoksun bir şekilde ele alınmıştır. Biçim ve desen oluşturmak için metinler, formüller, diyagramlar ve bilgisayar kodlarından işlevsel ilişkiler yaratılmıştır (Keller, 2005). Bu dönemdeki çalışmaların arka planında, sezgi ile rasyonel düşünce arasındaki bağlantıyı artırmak vardır (Martin, 1967). Bu amaçla bilgilerin analiz ile toplanması, deneylerle güçlendirilmesi ve simülasyonlarla onaylanması planlanmıştır. Bu amaçla 1960’ların sonlarında ve 1970’lerin başlarında dijital olarak yönlendirilen performansa dayalı tasarımdaki ilk dijital araçlar ve optimizasyon teknikleri geliştirilmiştir. Bu araçlar ve yöntemler, tasarımcıların bina performans ölçütlerine ilişkin doğru tahminler elde etmelerini ve tasarımcıya tasarım önerileri hakkında değerlendirici geri bildirim sağlayan “bilgisayar yardımcıları” olarak görülmüştür (Cross ve Maver, 1973: 274). Bilgisayar teknolojilerinin kullanılmaya başladığı ilk yirmi yılda optimizasyon

teknikleri ile de (özellikle mekân planlama) birçok mimari problem çözülmüştür (Gero, 1975).

Bilgisayarın mimari tasarım üzerindeki etkisinin araştırılması ve sorgulanması 1950’li ve 1960’lı yıllara dayanmış olsa da kişisel bilgisayarların yaygınlaşması ve mimari tasarım için dijital araçlara ilgi 1980’lerin ortasında CAD araçlarının ticarileştirilmesiyle başlamıştır (Celani, Vaz ve Pupo, 2013; Koutamanis, 2005; Peters ve Peters, 2013; Weisberg, 2008). Üretici sistemler üzerinde yapılan bilimsel çalışmaların katkısı, Moretti’nin çalışmaları, Sutherland’ın grafik arayüzü bu gelişimde önemli pay sahibidir. 1970’lerde popüler hâle gelen mekân planlama teknikleri, şekil gramerleri ve dikdörtgen düzenlemeler gibi üretici sistemler; farklı yaklaşımları ve hedefleri bir araya getirmiş, böylece bilgisayar destekli mimari tasarımın tanınırlığı artmıştır (Cross, 1977; Koutamanis, 2005).

1980’lere kadar çoğu bilgi işlem metin modunda yapıyorken, 1980’lerden sonra tasarım grafikleri ve kişisel bilgisayarlar kullanılmaya başlanmıştır. CAD sistemleri, 1970’lerin sonlarına kadar mimarlık firmaları tarafından dikkat çekmiyorken, 1980’lerdeki gelişmelerle birlikte 1990’ların başlarında bu firmaların büyük meblağlarda yatırım yaptıkları bir araç haline gelmiştir (Clayton, 2005; Clayton, March, Mitchell ve Stiny, 2005). 1980’lerden itibaren iki ve üç boyutlu dijital modelleme araçlarında patlama yaşanırken, algoritmik tasarım süreçleri üzerine araştırmalarında “prosedürel kısa” girilmiştir (Gaudillière-Jami, 2020: 153).

1980’li yılların başları iki boyutlu çizim ve bugün Yapı Bilgi Modelleme (BIM), olarak bilinen yazılımların öne çıktığı zamanlardır. İki boyutlu çizim çağı (Aish ve Bredella, 2017; Peters ve Peters, 2013: 41,42), iki boyutlu araçlar olan Autodesk’in AutoCAD’i ve BentleySystems’in Microstation aracı ile kendini belli etmiştir. Bu araçlar, geleneksel çizim çalışmalarını dijitale taşımının bir yolu gibidir. Bu araçlar bir hesaplama ürünü olsa da günümüzdeki algoritmik araçlarda yapıldığı gibi hesaplamalı tasarım yapmak için geliştirilmemişti. Yine de tasarımcılar için silme ve çizme işlemlerini tutarlı bir şekilde yaparak, kopyaların oluşturulmasını hızlandırarak ve büyük miktarda bilgiyi depolayarak büyük kolaylıklar sağlamışlardır.

1980'li ve 1990'lı yıllarda mimari tasarımcılar yavaş yavaş üç boyutlu modelleme araçlarını benimsemeye başlamıştır. 1990'larda ilk modern parametrik tasarım araçları geliştirilmiş ve performatif tasarım çalışmaları yapılmıştır (Kolarevic, 2013; Glymph, 2003). Parametrik tasarım ve performansa dayalı tasarım için güçlü ilişkiler kurulmuştur (Oxman, 2008a). 1990'lardan sonra eğri çizgiler (spline), doğrusal olmayan rasyonel temelli eğriler (NURBS), genellikle blob mimarisi olarak tanımlanan izomorfik çoklu yüzeyler ve animasyon mimari form oluşturmada üretken tasarıma yeni bir boyut kazandırmıştır (Kolarevic, 2005a). 1990'ların ortalarında, üç boyutlu NURBS'lerin kullanımıyla daha önce üzerinde çalışılmayan geometrilerle çalışılabilmiştir. Bir dizi olasılığı tanımlayan parametrik fonksiyonlarla temsil edilen sürekli ve eğrisel yüzeyler, daha hassas, kontrollü ve bilinçli tasarım imkânı vermiştir. Tutarlı, sürekli ve dinamik dönüşüm yapabilen tasarım modelleri, geleneksel süreçlerin statik normlarının yerini almıştır. Diğer yandan bu yıllarda geleneksel geometriye karşı deneysel mimariye geniş bir ilgi vardır (Celani ve Veloso, 2015: 37). Peter Eisenman; ölçeklendirme, fraktallar ve üretken diyagramlar gibi bir dizi tasarım tekniğini kullanarak ilk çağdaş üretken tasarım projelerinden bazılarını geliştirmiştir (Agnathidis, 2015: 14,15). Frazer'ın (1995) evrimsel mimari üzerine çalışması ile üretken tasarım kavramı ile biyolojik yapıların gelişiminin arkasındaki biyolojik ilkeleri uygulamaya odaklanan morfogeneze kavramı ile arasındaki ilişkiyi artırmıştır.

1990'ların sonlarında ve 2000'li yılların başlarında hesaplamalı mimari tasarım, birçok bakış açısıyla (morgenetik, biyomimetik, performatif, kinetik, adaptif, algoritmik, vb.) sunulmuş birçok üretken tasarım tekniği (fraktallar, l-sistem, otomatlar vb.) ile aktif olarak tasarım yapılabilen bir seviyeye gelmiştir (Caetano ve Leitão, 2019). 2000'li yıllardan sonra programlama araçlarının kullanımının kolaylaşması, bilgisayar programlama sürecini demokratikleştirmiştir. Mimari tasarımcıların özel ihtiyacını karşılaması için birçok uygulama üretilmiştir. Mimari tasarımcılar, yıllar içinde geliştirilen hesaplama stratejilerinin arkasındaki kavramları öğrenmiştir. Bunları kendi özel ihtiyaç ve ilgi alanlarına hitap eden kendi otomasyon araçlarını oluşturmak için kullanmışlardır. Sürdürülebilir mekânlar yaratmak için mekânların performansını optimize etmeye yönelik tasarım süreçleri geliştirmişlerdir. Simülasyon ve optimizasyona dayalı tasarım metodolojisi ile bilgisayarların belirli

koşullar altında sayısız deneme ve hatayı çalıştırmasına izin vererek en iyi çalışan çözümler bulmaya çalışmışlardır.

2010'lu yıllardan sonra simülasyon ve optimizasyon araçlarının erişilebilirliği artmıştır. Bilgisayarların katlanarak artan gücü ve kapasitesi, mimarların ve diğer profesyonellerin bir bina performans simülasyonu ve optimizasyon tekniğini tasarım uygulamalarına dahil etmelerini sağlamıştır (Attia, Hamdy, O'Brien ve Carlucci, 2013; Evins, 2013; Hamdy, Nguyen ve Hensen, 2016; Shi, Tian, Chen, Si ve Jin, 2016; Wortmann ve Nannicini, 2016). Optimizasyon algoritmasının seçimi, çözülmesi gereken probleme, istenen çözümün kalitesine, zaman sınırına, bilgi işlem kaynağına ve uygulayan kişinin uzmanlığına göre değişmiştir (Yang, 2010). Kullanıcı ihtiyaçlarına ve çözülmesi gereken spesifik probleme bağlı olarak en verimli optimizasyon tekniklerini araştırmıştır. Artık çok daha fazla mimari tasarımcı, tasarım sürecinde en verimli ve en hızlı sonucu bulmak için birçok seçeneği araştırabilmekte, test edebilmekte, değerlendirebilmekte, tekrarlayan görevlerle saatlerce ya da haftalarca uğraşmak yerine bu görevlerin otomasyonu ve optimizasyonu ile ilgilenebilmektedir. Mimari tasarımcılar, bir bilgisayarın gerçekleştirilmesi için bir dizi görevi hesaplamalı düşünerek formüle edebilmekte, formları bir dizi performansa yönelik simülasyona tabi tutup ve en uygun sonucun optimize edilmesini sağlayabilmektedir. Böylece, farklı türlerdeki modeller ve karmaşık sistemler birden çok soyutlama düzeyinde tanımlanabilmektedir (Toussi, 2020).

Gelişmiş yeni yapay zekâ yaklaşımları, daha akıllı algoritmaları ortaya çıkarmıştır (Carpo, 2017). Veri depolama alanlarının büyümesi, büyük veri kümelerinin sağlanabilmesi, hesaplama süresini kısaltan donanımlar ile birleşmiştir. 2010'lardan sonraki veri bolluğu sayesinde makine öğrenimi ve yapay zekâ teknolojileriyle birlikte mimari tasarımda daha verimli otomasyon gerçekleşmiştir (Carpo, 2017). Hesaplamalı tasarımdaki son gelişmeler ile birlikte hesaplamalı tasarım yöntemlerinin, yaklaşımlarının ve araçlarının entegre bir şekilde kullanımı ve böylece tasarım sürecinden olabildiğince verim alınabilmesi amaçlanmıştır.

2.3. Hesaplamalı Tasarımın Boyutları ve Mimari Tasarım Sürecine Yansımaları

2.3.1. Dijital Olmayan Hesaplamalı Tasarım Aracını Kullanmak

Hesaplamalı mimari tasarım dijital olarak yapılmak zorunda değildir. 19. yüzyılın sonlarında kullanılan Gaudi'nin asılı zincirlerden oluşan modeli (Jos, 1989), Frei Otto'nun analog hesaplamaya dayanan minimal yüzey deneyleri (Otto ve Rasch, 1996) ilk parametrik hesaplamalı tasarımlardandır. Elle kurgulanan şekil gramerleri (Özbek, 2004) ya da çeşitli fiziksel formlar (Oktan ve Vural, 2021; Varinlioglu, Halici ve Alacam, 2016) ile de hesaplamalı tasarım yapılabilir. Ancak fiziksel bir ortamda, yazılım ortamındaki karmaşıklık düzeyine ulaşmak zor olabilir.

Mimari tasarımcıların 20. yüzyılın ortalarına kadar, gelişmiş bilgisayar programları yoktur. Bunun yerine, şimdi parametrik olarak kabul edilen formları üretmek için bir tür “doğal hesaplama” kullanmışlardır. Gaudi ve Otto'nun mimari tasarımda biçim bulmanın öncüleri olduğu söylenebilir. Onlar herhangi bir bilgisayar aracı olmadan parametrik tasarım yapan ilk mimari tasarımcılardan bazılarıdır.

Gaudi'nin modeli hesaplamalı ve parametrik tasarımın analog olarak icraa edilmesine iyi bir örnek olabilir. Gaudi, Colònia Güell şapelinin karmaşık tonozlu tavanları ve kemerlerini oluşturmak için ağırlıklarla yüklü zincirleri kullanarak şapelin baş aşağı bir modelini yaratmıştır (Burry, 2007). Bu şekilde, “Hooke yasası” nedeniyle zincirler her iki ucundan asılmış, statik yükü bağlantılar arasında eşit olarak dağıtarak katener eğrisine dönüşmüştür. Böylece bir dizi parametreye sahip bir parametrik model oluşmuştur. Oluşturulan modelin zincirinin uzunluğu, bağlantı noktasının konumu, ağırlığı gibi bir dizi bağımsız parametresi ve yer çekimi kuvvetini kullanan parametreleri oluşturulmuştur (Davis, 2013a). Gaudi, bu parametrik modelin parametrelerini değiştirerek yer çekimi kuvveti aracılığıyla kemerin şeklini oluşturmuş ve ona bağlı diğer kemerler de bu değişiklikten etkilenmiştir. Bu durumda Gaudi, Colonia Güell şapelinin diğer versiyonlarını üretirken ortaya çıkan yapının saf sıkıştırımda olacağından emin olmuştur. Yükleri manipüle ederken de mimari formu

inceleyebilmek için modelin altına bir ayna yerleştirmiştir. Gaudi, bu şekilde parametre değerlerini değiştirerek deneme-yanılma yoluyla sonucu optimize etmiştir.

Algoritmik düşünme, her aşamada ortaya çıkan durum hakkında net düşüncelerle ve bütünü temsil eden küçük parçaları çözerek, doğal bir sonuca varmak için adım adım kuralları tanımlar. Gaudi, kemerlerini tasarlamak için, bir tahtaya kat planını çizmiş ve ardından sütunların başlamasını istediği yerden üzerlerinde ağırlık bulunan dizeleri asmış, uygun gördüğü şekilde daha fazla dize eklemiş ve bu, baş aşağı bir kuvvet modelini yaratmıştır (Kaznatcheev, 2017). Gaudi'nin modelinin parametrelerindeki ayarlamaları, istediği kemerleri ve sütunları sağlamıştır. Bu açıdan düşünüldüğünde uyguladığı yöntemin adımları birbirinden bağımsız değildir ve bu nedenle daha az algoritmik olmayabilir (Kaznatcheev, 2017). Gaudi'nin modeli, son prototipi ortaya çıkarmak için denklemleri ve fonksiyonları değiştirmenin sonuçlarını hesaplayan dijital parametrik otomatik stratejilere eşdeğerdir.

Durand'in yarattığı üretken sistem bir başka analog hesaplamalı tasarım örneğidir (Durand, 1813; akt. Madrazo, 1994). Durand, yarattığı üretken sistemde, tarihi ve işlevsel yapıların mevcut planlarını sadeleştirerek, plandaki yapı unsurlarını basit geometrik formlar olarak ifade etmiştir (Durand, 1813; akt. Madrazo, 1994). Bir yapı üretmek için yapının sadeleştirilmiş unsurlarından mantıklı mimari kompozisyonlar yaratmak için altı adımlı bir süreç planlamıştır (Durand, 1813; akt. Madrazo, 1994). İlk adımda kompozisyonların ana eksenlerini yerleştirmiş, ikinci adımda yeni bir ikincil eksen ızgarası yaratarak birincil olanları tamamlamış, üçüncü adımda akslar boyunca duvarlar döşemiş, dördüncü adımda duvarlarla sınırlanan alanlara kolonlar yerleştirmiş ve son olarak beşinci adımda duvarlar, revaklar, merdivenler ve diğer mimari öğeler plan görünümünde çizmiştir. Son olarak ise yükseklik ve kesit plandan oluşturulmuştur.

Durand, bu yapıları unsurlarına ayırarak ayrıştırma, bu unsurları basit geometrik formlar olarak sadeleştirerek soyutlama yapmıştır. Kompozisyonlar oluşturan süreci adım adım planlayarak da algoritmik düşünmüştür. Durand'in üretken sisteminde mimarlığın temel ilkelerini örnekler, yapı unsurlarını belli kurallara ya da belli bir stile göre birleştirmez. Üretken sistemi diğer mimari yapı kompozisyonları

için genellenebilir. Planlar, yükseklikler ve kentsel formların oluşturulması için sistematik olarak kullanılabilir (Durand, 1813; akt. Madrazo, 1994).

2.3.2. Hesaplamalı Tasarım Aracı Olmayan Hesaplamalı Tasarım Ürünü Kullanmak

Geleneksel CAD ve Yapı Bilgi Modelleme (BIM) araçları, araçların yazılımını yapan yazılımcı açısından hesaplamalı bir tasarım ürünüdür. Ancak kullanıcısının hesaplamalı tasarım yapmasını genellikle sağlamaz.

Mimari tasarımcı, tasarım faaliyetini doğrudan kâğıt, maket gibi araçlarla ya da bir makine ile etkileşime girerek yapar. Tasarımcı, kâğıt veya maket gibi araçlar ile etkileşime girerken, kara kutusunda düşünür. Bu bilişsel süreçte örtük bilgi kullanılır. Aynı bilişsel sürece, bilgisayar destekli tasarım (CAD) araçlarıyla (Tedeschi, 2014: 16) modelleme yaparken de girilebilir. Çünkü CAD, geleneksel çizime benzer şekilde, yani eskiz faaliyetinin dijital ortamdaki çevirisi gibi kullanılır. Yani tasarımcı zihninde hâlihazırda kavramsallaştırdıklarını direk olarak bilgisayar sistemine girdiği sunduğu veya depoladığı yerdir. Dolayısıyla CAD uygulamalarında sadece tasarım nesnesinin nihai sonucu görüntülenir ve sürecin geçmişi örtülü tutulur.

Bilgisayar destekli tasarımların ilki olan Sketchpad (Sutherland, 1963), BIM araçları, NURBS tabanlı modelleme araçlarının her biri hesaplamalı tasarımın ürünüdür. Ancak her biri ile hesaplamalı tasarım yapılamayabilir. Geleneksel iki ve üç boyutlu CAD ve BIM araçları kullanımları gereği genellikle hesaplamalı tasarım sürecine girilmez. AutoCAD gibi iki boyutlu araçlar, Sketchup gibi üç boyutlu CAD araçları ya da ArchiCAD gibi BIM araçları analog bir tekniğin dijital bir uygulaması gibidir. CAD ya da BIM yazılımları süreç odaklı değildir, sonuç odaklıdır.

2.3.3. Hesaplamalı Dijital Tasarım Araçları ile Hesaplamalı Mimari Tasarımlar Yapmak

Hesaplamalı dijital tasarım aracında hesaplamalı bir mimari tasarım üretilebilir. Kullanılan bu hesaplamalı tasarım artık aynı zamanda hesaplamalı tasarım yapmayı sağlayan bir araçtır. Buna, dijital bir hesaplamalı tasarım aracının içinde yeni

bir araç yaratmak ya da tasarım sonucuna ulaştıracak süreci dijital bir araçta tasarlamak da denebilir. Terzidis'e (2004, 2006, 2007) göre ancak bu şekilde şaşırtıcı sonuçlara ulaşabiliriz. Ona göre bu yaklaşım, bilgisayarın hesaplama gücünden en yüksek derecede yararlanabilmeyi amaç edinen ve kavrayışın ötesinde yöntemler ve araçlar kullanmayı ifade eder.

Mimari tasarımcılar için sunulmuş birçok dijital hesaplamalı tasarım aracı vardır. Bir mimari tasarımcı, genelde bir yazılım tasarlamaz, ancak bir amatör programcı (Woodbury, 2010: 9) gibi ya da bir yarı-programcı (Gane, 2004: 42) gibi hareket edebilir. Yarı-programcı kelimesi, yeni hesaplamalı düşünürleri tanımlamak için iyi bir kelime olabilir. Bu hesaplamalı mimari tasarım araçları, sunduğu kısa yollar ve elverişli arayüzü ile yazılım bilgisi olmayan bir mimari tasarımcının dâhi kullanabileceği şekilde programlanmıştır ve algoritmik düşünebilmeyi mümkün kılmıştır. Ayrıca daha zorlayıcı durumlar da olabilir. Örneğin metinsel komut dosyaları hesaplamalı tasarım araçlarının arkasındaki gerekli mantıksal yaklaşıma meydan okur (Burry, 2011). Bu, Kotnik'in (2006, 2010) de vurguladığı gibi hesaplama süreçlerine ilişkin farkındalığın en yüksek olduğu durumdur. Dolayısıyla tasarımcı, dijital tasarım sürecinin içerisine hesaplamalı tasarım yöntemlerini de sokarak teknolojiye sahip olmanın yanında teknolojiye hâkim olunabileceğini gösterir. Böylece elde edeceği tasarımın gerçek değerler ve veriler ile hazırlandığını doğrulayabilir.

2.3.4. Bizim Yerimize Düşünen Makineler

Makine öğrenimi bir yapay zekâ uygulamasıdır. Yapay zekâ, yapay bir işletim sisteminin akıllı canlılara özgü davranışları gerçekleştirebilme ve insana özgü bilişsel özellikleri kullanabilme yeteneği sağlar (Charniak ve McDermott, 1985; Nabyev, 2012). İnsan beyni doğuştan öğrenme yeteneğine sahiptir ancak bilgisayarların öğrenmesi için gerekli koşullar belirlenmelidir. Yapay zekâ (makine öğrenimi algoritmaları), verilerden anlam çıkarır, bu verileri daha geniş temsillere dönüştürmek için soyutlar, soyutlanmış verileri yeni bağlamlara uyarlamak için geneller ve öğrenilen bilginin potansiyelini ölçmek için değerlendirir (Lantz, 2019: 10-17). Bu anlamda, makine için gerekli koşullar yaratıldığında, tıpkı bir insan gibi hesaplamalı düşünebilir.

Hesaplama tasarımı süreçleri; problemin yorumlanmasına, çözüm üretmek için stratejilerin seçimlerine, tasarım prosedürlerine ve parametreleştirmeye dayanır. Tekrarlayan işleri manuel olarak yapmak yerine otomatikleşmeden yararlanır. Makine öğrenimi ise bu otomatikleştirmenin otomatikleştirilmesini sağlar (Raschka, 2016). Böylece tasarımcı, hesaplama adımlarını veya algoritmaları bir makinenin çalıştırabileceği şekilde planlamak (programlama) için uğraşmak yerine bir makinenin kendi sürecini geliştirmesi için zemin hazırlar. Bu süreçte bir makinenin yanında bir insan da hesaplama düşünmüş olur. Verilerle düşünebilmek ya da veriler üzerinde gerçekleştirilen bu işlemi anlamak, hesaplama düşünmeyi gerektirir (Mike, Ragonis, Rosenberg-Kima ve Hazzan, 2022).

2.4. Hesaplama Mimari Tasarım Yaklaşımları, Yöntemleri ve Modelleri

Bu bölümde hesaplama mimari tasarımı incelemek için yaklaşım, yöntem ve model kavramları kullanılmıştır. Bu çalışmada yaklaşım, hesaplama mimari tasarıma yaklaşım biçimini belirlemek açısından kullanılmıştır. Yöntem, bir yaklaşımın uygulaması olarak ifade edilmiştir. Bir yaklaşım altında birkaç yöntem olabilir. Model ise yöntemin uygulanmasının sonucu olarak ifade edilmiştir. Dolayısıyla model, tasarıma ulaştırılan sürecin kendisidir. Bu çalışmada yaklaşım, daha çok 'ne' sorusuna cevap verirken yöntem ve model 'nasıl' sorusuna cevap vermiştir.

Çalışmada hesaplama mimari tasarım yaklaşımları sınıflandırılmıştır: parametrik, algoritmik, üretken, optimizasyona dayalı, performansa dayalı, doğadan ilham alan, veriye ve deneyime dayalı. Bu yaklaşımlar altındaki hesaplama yöntemleri, onları hesaplama yapan nitelikleri açısından değerlendirilmiş ve diğer hesaplama yaklaşımları ile ilişkileri ortaya çıkarılmaya çalışılmıştır.

Günümüzde karmaşık formlar ve örüntüler üretebilmenin birçok yolu vardır. Geleneksel tasarım süreciyle mümkün olmayan birçok tasarım oluşturulabilir. Bu anlamda hesaplama mimari tasarım yaklaşımları ve yöntemleri, tasarımcıların bunlarla neyi sağladıkları ya da üretebildikleri ile değil, bunu nasıl gerçekleştirebildikleriyle benzersiz olduğu sonucuna varılmıştır. Çalışmada bu yöntem ve modeller, iç mimari tasarım sürecine etkileri ile birlikte ele alınmıştır.

Hesaplama yöntemleri birbirlerinden farklı bileşenler ve işlemler ile ayırt edilebilseler de hesaplama yaklaşımları arasında keskin ayrımların olmadığı görülmüştür. Bu anlamda bir yöntem, teknik veya araç birçok yaklaşımdan beslenebilir. Bu durum tanımları bulanıklaştırmıştır. Bu anlamda hesaplama yaklaşımları birbirlerinden keskin bir şekilde ayırmak yerine bu yaklaşımların odaklandığı fikirlere yoğunlaşmak daha uygun görülebilir. Öte yandan kalıplaşmış isimler de anlamların bulanıklığını artırabilir. Nitekim, parametrik tasarımın sadece parametreler kullanıldığı için parametrik olmadığı ya da algoritmik tasarımın, sadece algoritmalarla ilişkili olduğu için algoritmik olmadığı görülmektedir.

2.4.1. Parametrik Yaklaşım

“Parametrik” teriminin tasarımcılar tarafından ilk olarak ne zaman kullanıldığıyla ilgili tartışmalar olsa da terimin matematikteki karşılığının çok daha önceye dayandığı görülür. 19. yüzyılın başlarında, Dana (1837) ve Leslie (1821), üç boyutlu modelleri tanımlamak için parametrik denklemlerden yararlanırken parametrenin matematiksel anlamını kullanmıştır (Davis, 2013a: 18-22). Dana, “Kristal Figürlerin Çizimi Üzerine” adlı makalesinde bir dizi kristal çizimin genel adımlarını ve çeşitli parametrelerin çeşitli kristallerin çizimini etkilediği denklemleri aktarmıştır. Bu denklemlerle açık fonksiyonlar yoluyla parametrelerle ilişkili sonuçları bulmayı açıklamıştır (Davis, 2013a: 18-22).

Mimari tasarımda ‘parametrik’ teriminin anlamı, matematikteki anlamından fazlasıdır: Parametrik tasarımda, bir mimari problemi çözmek için yaratılan parametrik modelde parametre değerlerini değiştirerek modelin diğer değişkenlerini otomatik olarak etkilemek amaçlanır. Mimari tasarım amacıyla kullanılan ilk parametrik araçlardan günümüze kadar ortak bir amaç benimsenmiştir: O da kısıtlamalar yoluyla bir tasarımın parametreleri arasındaki ilişkilerin incelenmesini sağlamaktır (Bukhari, 2011: 8; Davis, 2013; Ei-Khaldi, 2007; Gane, 2004; Hudson, 2010; Jabi, 2013; Kalay, 1989; Kolarevic, 2013; Oxman, 2017; Monedero, 2000; Moretti, 1971: 207; Szalapaj, 2001; Woodbury, 2010). Schumacher’in sunduğu “parametrisizm” (2008) gibi bazı görüşler vurguyu farklı taraflara çekebilse de

parametrik tasarım bu kullanılma biçimiyle ayırt edici bir özellik kazanmıştır (Davis, 2013b).

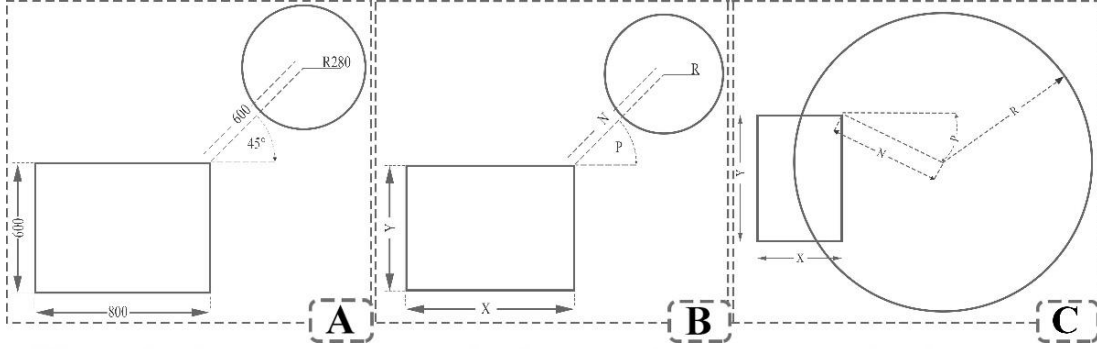
2.4.1.1 Parametrik Modeller ve Parametrik Yöntemler

Parametrik modelin üç temel bileşeni olduğu görülür: parametreler (değişkenler), parametrik denklem ve kısıtlamalar. Parametreler etkileşimli olarak değiştirilebiliyorsa değişken olarak tanımlanırlar. Değişkenler, bir tasarımı bir değerler topluluğu olarak ifade edebilmek ve tasarımı oluşturacak verileri daha sonra kullanabilmek için değerleri ya da verileri tutan bir kapsayıcıdır (Woodbury, 2010: 50,51). Tasarım değişkenleri en uygun çözümün bulunması sürecinde değiştirilebilen değerleri temsil etmektedir. Değişken olarak belirlenen kriterlerin değerlerinin düşürülüp artırılması ile geometri üzerinde değişiklik yapılır. Matematiksel, geometrik, topolojik, temsili, çevresel, yapısal, malzemeye yönelik ve insani faktörler olmak üzere birçok parametre (değişken) tanımlanmıştır (Jabi, 2013: 196,197).

Parametrik denklem, mimari tasarımcıların tasarımın belirli örneklerini oluşturulabileceği ve gerektiğinde değiştirilebileceği bir dizi ilkedir. Parametrik denklemler, birbirine bağlı geometrileri ve mimari formu oluşturan nesnelere arasındaki ilişkilerin oluşturulmasını sağlar (J. Burry ve Burry, 2010). Parametre değerlerinin değiştirilmesi, denklemlerin mimari tasarım formları arasındaki ilişkileri yapılandırmasıyla sonuçlanır. Böylece ilişkisel geometri yani “karşılıklı olarak bağlantılı bileşen geometrisi” tanımlanır (Burry, 1999).

Kısıtlama modellemeye temel bir değişiklik getirir: Bir tasarımın parçaları, koordineli bir şekilde ilişki kurar ve birlikte değişir. Kısıtlamalar, oluşturulacak parametrik modelin sınırlılığını belirleyen önemli faktörlerdir ve geometrik elemanların onları tanımlayan değişken ya da değişkenler üzerinde bir bağımlılık oluşturur (Gane ve Haymaker, 2007: 2,3). Burry ve Murray (1997), parametrik tasarımdaki kısıtlamaların işlevini en sade şekliyle anlatmıştır. Aşağıdaki şekilde (Şekil 2.2.’deki A), parametreleri olan ancak kısıtlamaları olmayan bir şekil örneği verilmiştir. İki nesneden oluşan bu nesne grubu, kısıtlamaları olmayan belli sabit ilişkiler barındırmaktadır (Şekil 2.2.’deki A). Bu açık tasarımın boyutlandırılmış

parametre değerlerinden herhangi biri değiştirildiğinde nesnelere birbirlerini kısıtlayacak şekilde ilişkilendirilmedikleri için işlem yalnızca bir dizi silme ve yeniden çizim yoluyla yapılabilir.



Şekil 2.2. Parametrik kısıtlamaların gösterimi.

Kaynak: (Burry ve Murray, 1997)

Şekil 2.2.'deki B'de ise parametrik bir yaklaşım örneği görülmektedir. Nesnelerin parametreleri arasında kısıtlamaları olan bu şeklin parametre değerleri değiştirildiğinde şekil kısıtlamalara göre güncellenebilir. Örneğin x, y ve r parametrelerinin değerleri değiştirildiğinde B'den C'ye doğru objelerin aralarındaki kısıtlamalara göre bir değişim olabilir. Dolayısıyla şekil B'nin şekil A'dan farkı, şekil B'nin ilişkisel geometri özelliğine sahip olması yani kısıtlamalarının olmasıdır. Örneğin, belirli bir genişlik oranı olarak bir pencere açıklığı, bir mimari cephenin parametre değerlerini etkileyecek şekilde kısıtlanabilir. Böylece, pencere bileşiminin parametre değerlerinin değiştirilmesi, kısıtlamaların kombinasyonlarına göre bütün mimari cepheyi yeniden yapılandırabilir. Bu durumda modeldeki pencerenin parametre değerlerini değiştirmek istediğimizde diğer ilgili parametreleri değiştirmek gerekmez, birbirleriyle ilişkilendirilen parametreler parametrik modeli eş zamanlı olarak değiştirir.

Parametrik tasarım modelinin parametrik olma özelliği, onun kapsamını geniş kılmıştır. Bu durum, onun anlamını ifade etmede karışıklığa neden olabilir. Bir tasarım kendi içinde zaten birçok parametreye sahiptir. Bu anlamda zaten neredeyse tüm tasarımlar parametrelere sahiptir (Aish ve Woodbury, 2005: 152; Hudson, 2010: 18-22; Jason Gerber, 2007: 54). Ancak parametresi olan her tasarım 'parametrik tasarım' değildir. 'Parametrik tasarım' bir süreçtir. Bu anlamda parametreler sadece bir nesneyi

yaratmak için kullanılmaz, tasarım araştırmasındaki birer araç haline gelir (Davis, 2013a; Woodbury, 2010). Bir modelin parametrik olabilmesi için kısıtlamalara bağlı birçok parametresinin olması gerekir. Yapılan tanımlamalara göre eğer bu parametrelerden birinin değeri değiştirildiğinde kısıtlamalara bağlı olarak diğer parametreler etkileniyorsa ve bu şekilde birçok alternatif oluşturabiliyorsa parametrik tasarım yapılmış olur. Bu anlamda parametrik formlar sadece parametrelerin varlığından ya da parametre değerlerinin hassas ayarından dolayı değil, aynı zamanda parametrik ilişkilere veya kısıtlamalara göre türetilbildikleri için değerlidir. Parametrelerin sonuçlarıyla olan bu ilişki parametrik tasarımı özgün kılar (Davis, 2013b: 24; Dino, 2012: 210). Dolayısıyla parametriklik, parametreler ve geometrik model arasındaki bağlantıyı vurgulamasıyla potansiyel olarak bir dizi model türünü de dışlayabilir (Davis, 2013b: 31). Örneğin hücresele otomat ve evrimsel tasarım yöntemlerinde bu açık bağlantı eksiktir. Bu yöntemlerde, değişkenler üzerinde yapılan ayarların sonuçları üzerindeki etkisi sezilemez.

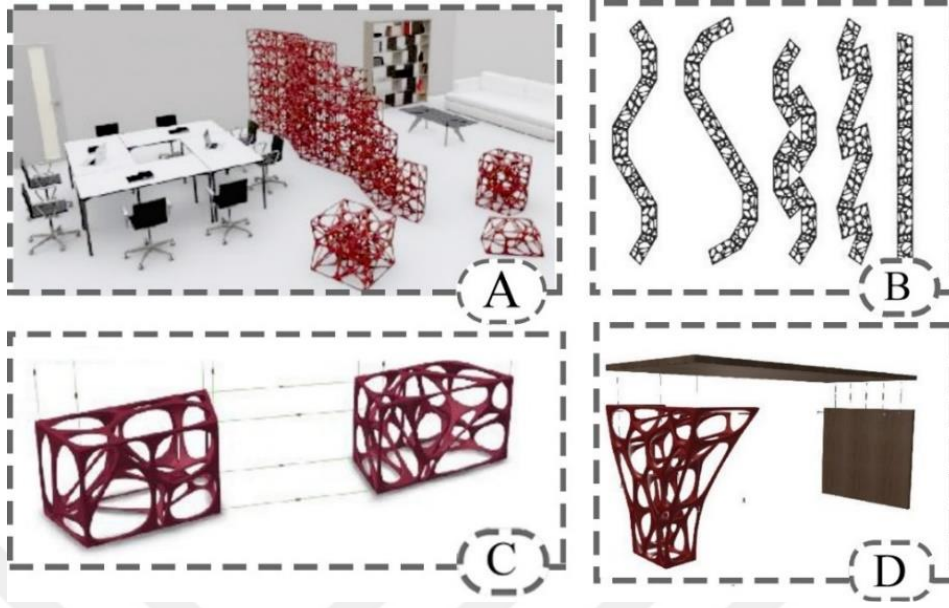
İç mekân tasarımında parametrik ilişkilerden yararlanılabilir. Örneğin “Zaha Hadid Architects” tarafından tasarlanan Serpentine Sackler Gallery, Roca London Gallery, Stuart Weitzman Flagship Store eserlerindeki strüktür tasarımlarının iç mekânın kimliğini oluşturan estetik unsurlar olarak kullanıldığını görülür (Şekil 2.3.). Bu eserler genellikle parametrik modelleyiciler ile tasarlanır. Yapının yapısal performansı ve estetik formunun formülü parametrik modelin içinde gizlidir. Parametrik kısıtlamalara sahip bu tasarımlar bir noktadan başlayarak tüm biçimlerin birbirleriyle ilişki kurmasını sağlar. Bu ilişkiler, yapıların kabuklarından iç mekân unsurlarına kadar yansıyan bir bütünlük görüntüsünün oluşmasını sağlamıştır. Yapıların bütününe yansıyan akışkan bir form görüntüsü oluşmuştur.



Şekil 2.3. Zaha Hadid Architects tarafından tasarlanan parametrik yapılar, a'da Serpentine Sackler Galerisi, b'de Roca Londra Galerisi, c'de Stuart Weitzman Flagship Mağazası.

Kaynak: (Sanal 1, 2022).

Parametrik tasarım araçlarında mobilya tasarım süreci yaratıcı bir şekilde gerçekleştirilebilir (Hamad ve Husein, 2020). Manavis ve diğerleri (2021), birlikte kullanılması amacıyla bütünsel bir yaklaşımla bir dizi benzersiz geometrik formu üretmek için bir kavramsal çerçeve önermiştir. Bu amaçla parametrik modeldeki algoritmik prosedüre dayalı olarak üç boyutlu baskı ile üretilmesi için kutular gibi basit ve doğal formlara odaklanan bir dizi oda/ofis bölücü modellemiştir (Şekil 2.4.'te a). Uygun kodu oluşturmak için Grasshopper'ın kullanıldığı parametrik modele bir dizi geometrik parametre ve kısıtlama tanımlanmıştır. Böylece kullanıcı bunları belirli bir değer aralığını değiştirerek birçok tasarım kombinasyonu üretebilmiştir. Parametrik modeldeki parametrik ilişkiler sayesinde tasarlanacak bölücü istenen açıya ve yöne göre konumlandırılırken ya da bölücünün toplam geometrisiyle ilgili değişiklik yapılırken, geometrik form da otomatik olarak değiştirilebilmiştir (Şekil 2.4.'te b). Aynı dış boyutlarda ve uygun montajları ile alternatif tasarımlar üretilmiştir (Şekil 2.4.'te c). Ayrıca benzer hesaplama koduyla çok sayıda bileşen sınırlı bir sürede ve yaygın olarak kabul edilen üç boyutlu baskı üretim süreçleri izlenerek üretilmiştir (Şekil 2.4.'te d).



Şekil 2.4. A’da ofis bölücü örneği, b’de parametrik modelin geometrik formunun esnekliği, c’de boyutsal kısıtlamalar sayesinde birbirlerine eklemenecek şekilde tasarlanmış parçalar, d’de alternatif ürün

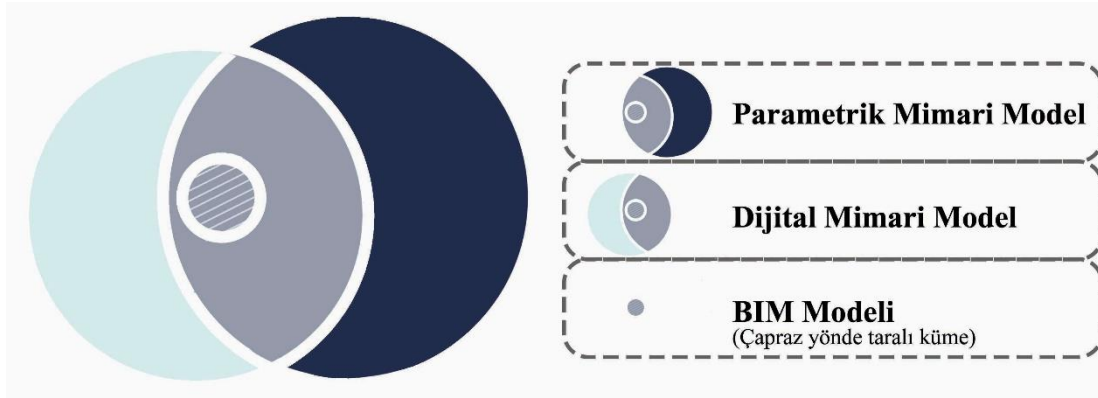
Kaynak: (Manavis ve diğerleri, 2021).

2.4.1.2. Parametrik Yaklaşımın Hesaplamalı Tasarım ve Diğer Hesaplamalı Yaklaşımlar ile İlişkisi

Parametrik modeldeki parametre değerleri, elle hassas olarak değiştirilerek tasarım şekli aranır. CAD programında bir prizmanın uzunluk, genişlik gibi parametre değerleri elle değiştirilerek olası değişiklikler araştırabilir ya da bir BIM aracında sembolik parametreler değiştirilerek de yeni varyasyonlar keşfedilebilir. Bu eylemler için hesaplamalı düşünmek gerekemeyebilir. Ancak, tasarımcının elinde daha fazlasını yapabilmesi için imkânlar vardır. Parametrelerin ince ayarı, tanıdık bir model oluşturmak amacıyla değil, tasarımcıyı tasarım çözüm alanının yeni ve beklenmedik bölgelerine götürebilecek düzeyde olduğunda dijital araçların gücünden daha verimli bir şekilde yararlanılmış olur (Hudson, 2010; Jabi, Soe, Theobald, Aish ve Lannon, 2017; Oxman, 2006, 2017; B. Peters ve Peters, 2013; Terzidis, 2004, 2006). Çeşitli parametrelerden ve kısıtlamalardan yola çıkan bir prosedür, algoritma ya da komut dosyası yaratıldığında, diğer bir deyişle, parametrik sistemleri “orta ile yüksek karmaşıklığa sahip” şekilde (Jabi ve diğerleri, 2017: 5) kullanıldığında, tasarım

sürecinde algoritmalarla tam anlamıyla yararlanılmış olur. Bu süreçte hesaplamalı düşünmenin bilincine varıldığında ise hesaplamalı tasarlanmış olur ya da diğer bir deyişle hesaplamalı tasarım yapılmış olur.

Mimaride parametriklik uzun yıllar kullanılmıştır. Yirminci yüzyılın başındaki Antoni Gaudí'nin asılı zincir modelinden 1950'li yıllardaki Frei Otto'nun fiziksel parametrik modellerine, Moretti'nin 1960'lı yıllardaki parametrik mimari çalışmalarından Sketchpad aracına, 1990'lardaki geçmişe dayalı parametrik yazılımlardan BIM araçlarına, metinsel programlama araçlarından görsel komut dosyalarına kadar çeşitli yazılım ortamlarında parametrik modeller oluşturulabilir. Şekil 2.5.'teki venn şemasında da görüldüğü üzere dijital yöntemler parametrik yöntemlerle sınırlı değildir ve parametrik mimari modeller dijital olmak zorunda değildir. Gaudi, Otto gibi mimari tasarımcılar analog parametrik araçları kullandılar. BIM araçlarında, parametrik ilişkilerin nesnelere kendi içinde saklı olduğu dijital modeller oluşturulabilir (Humppi ve Österlund, 2016). BIM araçları, dijital ve parametrik mimari model kümelerinin kesişiminde bulunur (Şekil 2.5.). Günümüzde dijital modeller olarak geçmişe dayalı parametrik modelleme araçları, metinsel modelleme araçları ve özellikle Grasshopper, Dynamo gibi görsel komut oluşturma araçları kullanılmaktadır.



Şekil 2.5. Parametrik, dijital ve BIM modellerinin birbirleriyle ilişkilerini gösteren venn şeması.

Parametrik tasarım modelleri dijital tasarım araçlarının görünmez matematiksel tanımıyla oluşturulabileceği gibi Grasshopper, Dynamo ve Generative Components gibi komut oluşturma araçlarında da oluşturulabilir. Geometrinin

algoritmik tanımı bu komut oluşturma araçlarında açıktır. Bir parametrik sistemin temel bileşeni, bu araçlardaki şema veya grafik olarak adlandırılan algoritmanın kendisidir; dolayısıyla komut dosyası arabirimleri parametrik modeller oluşturmaya yatkındır (Dino, 2012). Bu araçlarla komut dosyasını yaratıldığında algoritmik bir yaklaşımdan yararlanılmış olur ve dolayısıyla hesaplamalı tasarım yapılmış olur. Ancak asıl parametrik tasarım işi; komut dosyasını yarattıktan sonra başlar (Chebiyyam, 2022). Parametrik tasarımın tam olarak konumu, bu komut dosyasındaki parametreler (değişkenler) ile oynandığı ve istenen sonucun aramaya koyulduğu yerdir. Tasarım araştırması için araç (komut dosyası) yarattıktan sonraki aşamada parametrelerle oynanabilir ve sonuç optimize edebilir. Ancak bu iş yükü hesaplamalı düşünmeyi gerektirmez. Hesaplamalı tasarım prosedürün yaratıldığı, yani algoritmik tasarımın yapıldığı yerdir.

Parametrik/algoritmik model, aynı zamanda üretken bir potansiyel barındırır. Parametrik ilişkilerin olası sonuçlarını elle bulmak yerine tüm olasılıkları hesaplayan üretken bir yaklaşım benimsenebilir. Ayrıca alternatifleri üreten üretken sistem (Bölüm 2.4.3.) bir simülasyon aracıyla değerlendirilebilir (Bölüm 2.4.4.) ve değerlendirmelere göre sonuç optimize edilebilir (Bölüm 2.4.5.). Tüm bu süreç makine öğrenimi yöntemleriyle de hızlandırılabilir (Bölüm 2.4.7.) Bu durumda algoritmik, parametrik, üretken yaklaşımların yanında performansa, optimizasyona ve veri öğrenmeye dayalı bir yaklaşım benimsenmiş olur. Bu anlamda parametrik modeller ve araçlar anahtar bir rol üstlenir (Oxman, 2017b). Dolayısıyla parametrik model; tasarım araştırması sırasında parametrelerin çeşitliliği yoluyla tasarım keşif uzayını genişletme potansiyelleri nedeniyle üretken, hesaplama adımlarını açık olarak sunabildiği için algoritmik, formun parametrik kontrolünün tasarım ortamlarının araştırılmasına izin verdiği için analitik ve en iyi formu bulmaya odaklı bir süreç olduğu için optimizasyona yönelik bir yöntem olarak hareket edebilir (Dino, 2012).

2.4.2. Algoritmik Yaklaşım

Tasarım düşüncesi, örtük ya da açık bilgidен oluşur (Woodbury ve Burrow, 2006; Oxman, 2006). Açık tasarım uzayı ziyaret edilen ve kaydı tutulan mevcut bir uzayken, örtük tasarım uzayı, olabilecek tüm olası tasarım çözümünü kapsayan

uzaydır. Hem açık hem de örtük uzaydaki bilgi, deneyim yoluyla elde edilebilir ancak açık bilgide örtülü bilginin aksine bilgi biçimleri resmileştirilir (Collins, 2010). Açık bilgi, bilişsel süreçler ile iyi formüle edilmiş ve tüm aşamaları kaydedilmiş ve hatta bu bilgilerle tekrar etkileşime izin verebilen bilgi temsillerini ifade eder. Bir tasarım sürecinin algoritmik olması için bu açık bilgi alanında oluşturulması gerekir (Kotnik, 2006, 2010). Böylece tasarım prosedürü izlenebilir, algoritma ile sonucu arasında kesin ilişkiler kurulabilir ve tasarım süreci daha kontrol edilebilir hâle getirilebilir (Caetano ve diğerleri, 2020: 294-297).

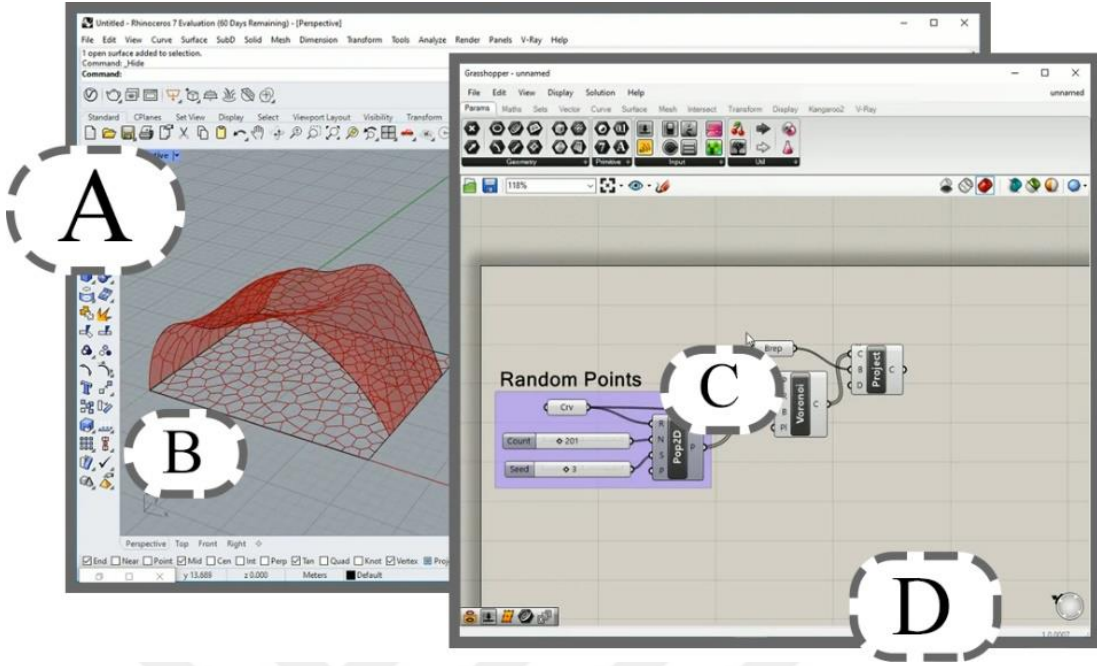
Terzidis (2006: 41), bu dijital teknolojilerde, algoritmaların potansiyellerinin farkına varıp, en üst düzeyde yararlanmayı vurgular. Algoritmiklik, Terzidis'in (2004, 2006, 2007) dikkat çektiği hesaplamanın potansiyeliyle birlikte tasarım prosedürünün açık, izlenilebilir ve kontrol edilebilir olmasıyla ilişkili olduğu düşünülebilir (Caetano ve diğerleri, 2020). Algoritmik tasarım prosedürleri genellikle karmaşık problemlerin çözülmesi için yaratılır, ancak daha basit prosedürler onun algoritmik yapıda olmadığını göstermez.

2.4.2.1. Algoritmik Tasarım Modeli ve Yöntemleri

Algoritmik tasarım, CAD programlarına eklenen script dilleri vasıtasıyla Autolisp, Rhinoscript, Mayascript, Maxscript, Designscript gibi araçlarda kodlamayla yapılmıştır. Diğer yandan biçim gramerinin prosedür adımlarının açık bir şekilde ifade edilebildiği tasarım alanları ile algoritmik yaklaşım vurgulanmıştır. Ancak mimari tasarımcılar yaygın olarak görsel komut oluşturmaya dayalı araçlar olan Grasshopper, Dynamo ve GenerativeComponents'i kullanarak parametrik/algoritmik modeller oluşturduğu görülür. Tasarımcılar parametrik/algoritmik modelleme araçları vasıtasıyla algoritmik sistemler üretir ve parametrik tasarım yaparlar. Parametrik/algoritmik model, tasarım öğelerini etkileyen iç ve dış değişkenler arasındaki ilişkileri, karmaşık sistemlerin tasarımını bilgilendirmek için kullanılacak parametrik denklemlere çevirir. Bu denklem takımları ile artırılmış hesaplama kontrolünü sağlamak için algoritmadan yararlanır. Koşullar, döngüler ve parametreler arasındaki ilişkileri belirleyen parametrik denklemler algoritmanın çoklu adımlarının parçasıdır.

Generative Components, Grasshopper, Dynamo nispeten benzerdir (Chaszar ve Joyce, 2016). Modelin oluřturmasını sađlayan řemaları kullanırlar (řekil 2.6.). Parametrik řemalar, parametrik bir model iindeki iliřkileri sađlar. řemalar, birbirlerine bađlanarak parametrik modeldeki iliřkileri kurar. řemalar yalnızca ileri beslemelidir, dolayısıyla dngsel bađımlılıkları yoktur. Ynlendirilmiř ve dngsel olmayan bu grafikler, iřlemler arasındaki veri akıřını tanımlar (Davis, Burry ve Burry, 2011). Girdi verisini deđiřtirmek, grafik boyunca akan veriyi deđiřtirir.

Rhinoceros-Grasshopper, iki pencerede ynetilir (řekil 2.6.'da A ve D). Ekranların birinde řemalar vasıtasıyla modelleme yapılır. Bu ekranda tasarım prosedr iřlenir. Diđer ekranda, yaratılan prosedre eř zamanlı olarak gncellenen grsel grntlenmektedir. Rhinoceros penceresinde gsterilen tasarım nesnesinin grsel grnts parametrik tanımla eřzamanlı ve etkileřimli olarak bu pencerede řekillenir. Rhino/Grasshopper'daki, tanımlar (grsel programlama) bir dizi kural ve talimattan oluřur. Bu programlar ađlar ile birbirine bađlanan dđmlerden oluřur. İki dđm tr vardır: verileri statik ve dinamik olarak depolayan parametreler ve bu parametrelerden veya diđer bileřenlerden beslendikleri veriler zerinde eylemler gerekleřtiren ve bu eylemleri genellikle yeni bir veri kmesiyle sonulandıran bileřenler (Bazafkan, 2018: 20). Grasshopper'da bir dizi algoritmik kurallara bađlı dđmler (řemalar), sol taraftaki parametrelerden giriř verilerini alıp sađ taraftaki ıkıř verilerine bađlar. Veri aktarımı bu řekilde ađlar ve dđmler boyunca devam eder (řekil 2.6.). řekil 2.6.'da Rhinoceros 3d modelleme ortamı A'yı ve Rhinoceros ile paralel olarak alıřan bir pencere olan Grasshopper editr D'yi gstermektedir. B grseli, C ile gsterilen, grsel algoritma ile oluřturulmuřtur. C'deki grsel birbirlerine bađlı řemalardan oluřan tanımını (grsel programı) gstermektedir.



Şekil 2.6. Rhino/Grasshopper arayüzü.

Grasshopper programındaki şemalar ile oluşturulan algoritmik kurgu, bir iç mekânın donatılarının veya mobilyalarının tasarımını temsil edebilir (Buna, Badiu ve Eles, 2015; Felek, 2022). Grasshopper gibi parametrik/algoritmik tasarım araçlarında kurgulanan süreç, oluşturulan ilk modeldeki değişkenlerin ilişkilerinin kurulmasını ve başa dönmeden nihai ürünün versiyonlarının ortaya çıkarılmasını sağlar. Bu algoritmik yapı sayesinde tasarımda yapılan değişikliklerde tasarımcının süreci baştan planlaması gerekmez. Böylece üç boyutlu modellerin farklı versiyonlarını hızlı bir şekilde oluşturulabilir. Örneğin, daha hızlı montaj tasarımını gerçekleştirmek için farklı parametrik tasarım araçları ve veri girişi için Excel dosyalarında yapılan tasarım tabloları kullanılarak raf ve dolap tasarımı yapılmıştır (Buna, Badiu ve Eles, 2015; Felek, 2022). Üç boyutlu modellerin farklı versiyonlarını hızlı bir şekilde oluşturmak için tasarım tablosuna boyutlar, malzemeler, açılar gibi giriş verileri girilmiştir. Tasarlanan hesaplamalı model, müşterilerin değişen talepleri doğrultusunda Excel’de oluşturulan tablodaki değişkenlerin ayarlanmasını ve buna göre tasarımın yapılandırılmasını sağlamıştır.

2.4.2.2. Algoritmik Tasarım Yaklaşımının Hesaplamalı Tasarım ve Diğer Hesaplamalı Yaklaşımlar ile İlişkisi

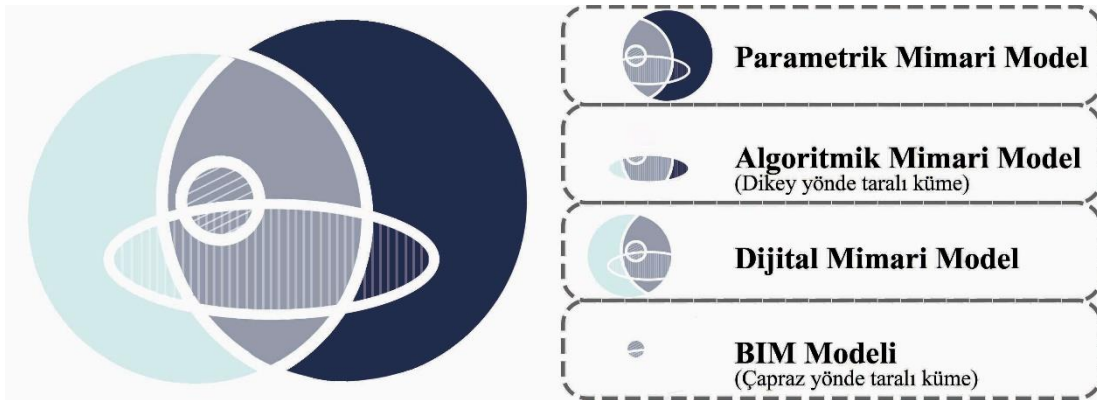
Algoritmik tasarımda, hesaplama prosedürü adım adım kodlanır, matematiksel ve işlevsel bir algoritma inşa edilir. Bu nedenle, hesaplamalı tasarımın tam anlamıyla karşılık bulduğu yerdir. Bu nedenle algoritmik tasarım, algoritmik düşünmeyi gerektirir. Hesaplama fonksiyonunun algoritmik bir tanımı; optimal bir çözümün hesaplanması olasılığından, tasarımın karmaşıklığından ve performansa uygunluğundan ziyade incelenen mimari elemanlar arasındaki geometrik ilişkiyi kontrol etme yeteneğinde yatmaktadır. Bu nedenle komut oluşturma algoritmik bir tasarım olarak görülürken hücresele otomat, evrimsel tasarım yöntemleri vb. yöntemler, tasarım sürecindeki girdilerinin belirsizliği nedeniyle henüz bütünüyle algoritmik düşünce fikriyle örtüşmemektedir (Çıltık, 2008: 40).

Komut oluşturma araçlarını kullananların sahip olması gereken alt beceriler arasında veri akışını kavramak, ayrıştırma, kodlarla isimlendirmek, soyutlama, algoritmik ve matematiksel düşünme bulunur (Woodbury, 2010 : 24-34). Bu anlamda hesaplamalı tasarım süreci, soyutlamalar arasında ilişkilerin olduğu çoklu otomatik algoritmaların kodlanmış ilişkisini içerir (Soleimani, 2019). Bu kodlar, formun geometrik durumuna atıfta bulunan bir birim olan “form oluşturma’nın resmi dijital alt katmanıdır.” (Lorenzo-Eiroa ve Sprecher, 2013). Tasarım sürecinde bu soyutlamalara yani parametrelere farklı öncelikler verilebilir.

Tasarımcı, parametrik tasarımın algoritmik boyutunda, diğer bir deyişle algoritmik tasarım sürecinde, tasarım ürünüyle ilgili tüm verileri yazılıma kodladığı için ekranda sürecin tüm aşamalarını görebilir. Dolayısıyla tasarım süreci açık tasarım uzayında ilerletilir. Tasarımcı bu durumda kara kutusunda düşündüklerini, veriler ve bilgiler aracılığıyla komutlara yansıtarak belirgin hâle getirebilir. Veriler ve bilgiler sistematik bir şekilde, parametrelere ve algoritmalara tanımlanır. Tasarım nesnesinin değiştirilebilir parametreler aracılığıyla oluşum geçmişiyle bağlantılı olduğu tasarım geliştirme sürecinin (grafik) bir geçmişi tutar ve tasarımcıların keşfederken gerçek zamanlı geri bildirim erişmesine olanak tanır. Parametrik/algoritmik tasarım modelleri, belirli sınırlar içinde değiştirilebilen bir dizi parametreye dayalı bir

tasarımın oluşturulmasını tamamen otomatikleştirir. Komut dosyası yazılarak hem kara kutu modelleme yazılımının sınırlamalarından kurtulunur hem de daha hızlı yineleme için üretkenlik artırılır (Burry, 2011). Mimari tasarımcıların yarattığı bu algoritmik ve parametrik modeller performans simülasyonları, optimizasyon algoritmaları ve makine öğrenimi yöntemleri ile de desteklenebilir.

Dijital parametrik/algoritmik modellerin dışında algoritmik modellerde oluşturulabilir. Algoritmik mimari modelin diğer modeller ile kesişimleri venn şemasında gösterilmiştir (Şekil 2.7.). Algoritmik model, parametrik yaklaşımla fiziksel model (analog) olarak da oluşturulabilir (Oktan ve Vural, 2021). Bu durumda dijital olmayan algoritmik ve parametrik bir model kullanılmış olur. Bir başka durum, bir BIM modelinin algoritmik yaklaşımda kullanımınıdır. Bu durum için algoritmik BIM aracı Dynamo kullanılır. Venn şemasının dışında kalan durumlar da vardır. Bir mimari tasarım süreci, analog olarak hesaplama adımlarıyla planlanabilirken, bu hesaplama adımları ile parametrik ilişkiler kurulmayabilir. Diğer yandan, CAD araçlarında izlenebilir basit komutlar oluşturularak sıralı ve kurallı biçimde tasarımlar yapılabilir (Öksüz, 2020). Bu durumda parametrik ilişkilerden yararlanılmayabilir. Dolayısıyla bu durumda sadece dijital araçta hesaplama adımları planlanmış olur.



Şekil 2.7. Algoritmik mimari modelin, parametrik, dijital ve BIM modelleri ile ilişkilerini gösteren venn şeması.

2.4.3. Üretken Yaklaşım

Mimari tasarımda üretkenlik, en arınmış anlamıyla birden fazla seçeneği üretme olasılığıyla ilgilidir. Üretken yaklaşım çıkarılabilecek ürünlerin potansiyeline odaklanır (Aghaei-Meibodi, 2016: 56). Bu potansiyel birçok üretici ile oluşturulabilir. Tasarımcı, üretken tasarım yaparken malzeme veya ürünle doğrudan ya da uygulamalı olarak etkileşime girmek yerine yarattığı üretici bir sistemden yardım alır (Fischer ve Herr, 2001). Tasarımcı üreticiyi yaratır, üretici tasarımları üretir. Bu anlamda üretkenliğe anlamını “üretici” verir. Üretken tasarım, bir anlamda tasarım üreten üreticiyi tasarlamak olduğu için üretkenliğin anlamı üreticiye göre değişir. Buradaki ‘üretici’ tasarımcının tasarım yapması için aracı bir rol üstlendiği kurallar veya algoritmalar olabilir (Singh ve Gu, 2012).

2.4.3.1. Üretken Tasarım Modelleri ve Yöntemleri

Mimari tasarımda üretken sistem belli bir mantığa, kurala ve amaca dayalı olarak bilgisayar programıyla, geometrik dönüşümlerle, bir diyagramla ya da diğer prosedürlerle yaratılabilir. Kural tabanlı modeller (biçim grameri, fraktallar ve 1-sistemleri vb.) birlikte ajan tabanlı modeller, hücresel otomalar, parametrik/algoritmik sistemler, optimizasyon yöntemleri (Bölüm 2.4.5.), yapay sinir ağları (Bölüm 2.4.7.) gibi birçok araç veya yöntem üretici rol üstlenebilir. Bu üretken sistemler tam otomatik süreçten adım adım kullanıcı kontrollü sürece kadar farklı derecelerde otomasyona sahiptir. Uygulama, dijital araçlarla gerçekleştirilmek zorunda olmasa da büyük miktarda veri üretmenin ve otomasyonun sağlanması açısından bilgisayarlardan yararlanmak uygun görülür. Bu yüzden üretken sistemler, her yaklaşımda yöntemlerde olduğu gibi daha çok dijital ile ilişkilendirilmiştir. Örneğin optimizasyon algoritmaları, rastgele bir tahmin veya bilinen bir çözümden yeni çözümler arayabilir ya da üretebilir. Bu anlamda özellikle genetik algoritmalar optimizasyon görevleriyle birlikte üretken tasarım yöntemlerinin bir parçası olarak mimari tasarımda yaygın olarak kullanılır (Bölüm 2.4.5.).

Mimari tasarımda amacın belli kurallara göre formüle edildiği birçok üretken yöntem bulunur. Mimari tasarımcıların en çok kullandığı kural tabanlı üretken

yöntemlerden biri biçim (şekil) gramerleridir. Biçim gramerleri, biçimlerin kurallar aracılığıyla dönüştürülmesi ve yeni biçimlerin oluşturulmasını amaçlayan hesaplamalı bir yaklaşımdır (Oxman, 2006; Oxman, 2008). Biçim gramerleri, tıpkı dildeki gibi bir takım sembollerin birbirleriyle ilişkilerini yöneten kurallara göre anlam kazanır (Stiny ve Gips, 1978). Bu kurallar, özgün bir biçim grameri dili üretir. Bu model dört bileşenden oluşur: başlangıç şekli, biçimler, biçim kuralları ve semboller dizisi (Stiny, 1980). Biçim grameri kuralları tariflenmiş bir başlangıç şekli ile başlar. Bir başlangıç şeklinin konumu, yönelimi ve ölçüsü kural setlerine uyulması ile birlikte diğer adımlarda ortaya çıkacak diğer biçimlerin konumu, yönelimi ve ölçüsünü belirler. Biçimler, tasarım tanımları olarak temsil edilir ve bir algoritma ile ifade edilebilir (Çolakoglu, 2002; Knight, 1999; Stiny, 1980). Biçimler, nokta ve çizgi gibi tek boyutlu elemanlardan, iki boyutlu çeşitli yüzeylerden ve çeşitli üç boyutlu kütlelerden oluşabilir, standart ya da parametrik olarak tanımlanabilir. Standart biçim gramerinde şeklin noktalarını bağlayan çizgiler ve nesnelere arasındaki ilişkiler değişmezken, parametrik biçim gramerinde aynı şeklin köşe noktalarının koordinatları, çizgilerin boyutları, açıları değiştirilebilir (Knight, 1994). Parametre değerlerindeki değişiklikler aynı anda bütün kompozisyona yansır.

Biçim gramerlerinin kuralları, oluşturulan tasarımların aynı zamanda açıklamalarıdır. Biçimler, bu kurallar doğrultusunda mimari tasarımı ifade etmek için dönüştürülebilir ve değiştirilebilir. Nokta, çizgi yüzey ve kütlelerden oluşan gramerler üzerinde toplama, çıkarma, kesiştirme gibi “boolean” işlemlerinin yanı sıra döndürme, yansıtma gibi Öklid dönüşümleri ve parametrik değişiklikler uygulanabilir (Mitchell, 1986; Stiny, 1984; Tan, 1989). Tanımlanmış kurallar ve biçim seti ile birlikte istenen adımda istenilen kural uygulanabilir, böylece birçok alternatif tasarım elde edilebilir. Ancak bu kurallar sabit değildir yani değiştirilebilir, geliştirilebilir, yapılandırılabilir ve eşzamanlı olarak sonuçlara yansıtılabilir. Biçim gramerleriyle yapılan çalışmalar, farklı yollarla ve farklı kurallarla da oluşturulabilir (Torus, 2008).

Biçim gramerleri formal ve informal olarak ikiye ayrılırlar. Formal biçim gramerleri, bilgisayara biçim üretme yönteminin tanımlandığı algoritmik bir tasarım süreciyle ilişkilidir. İnformal biçim gramerleri ise tasarımın biçim üretme adımlarında

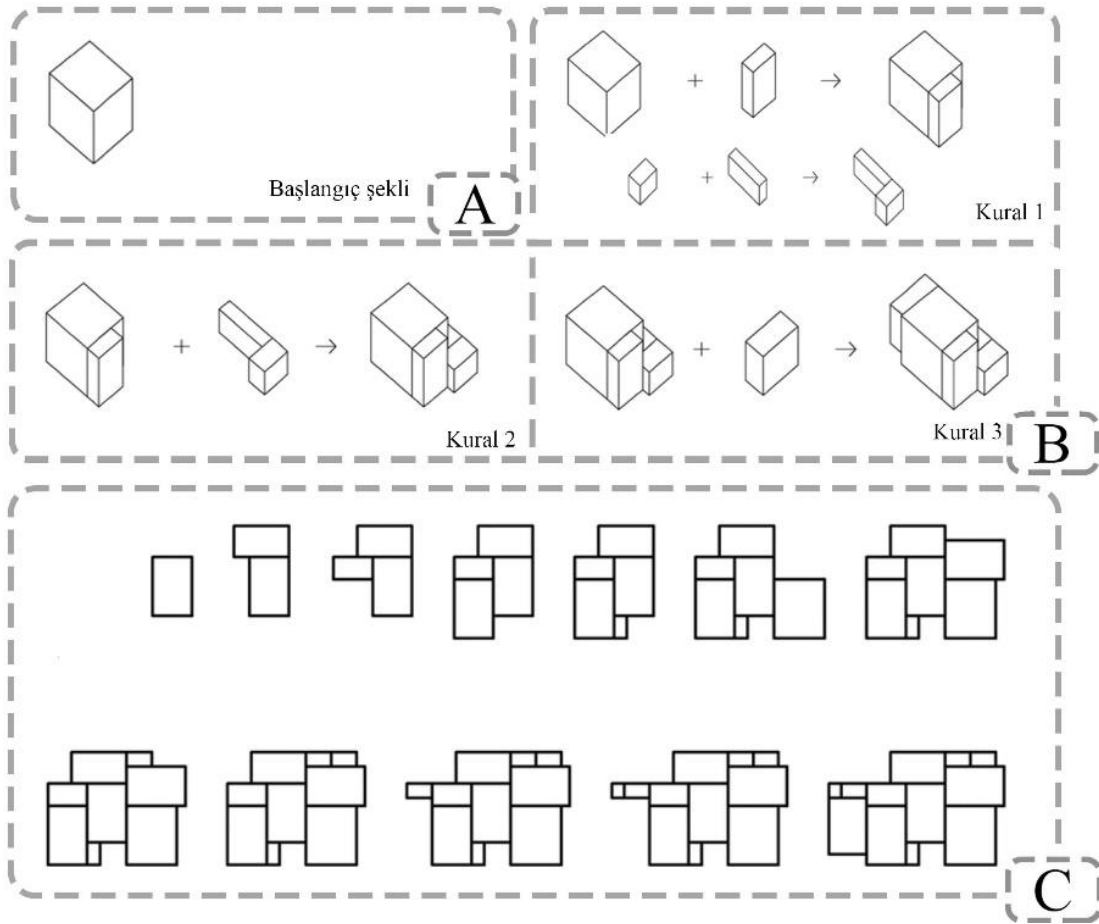
uygulanan kuralların şekillerle ifade edilmesidir. Kurallar bilgisayarın dilinde sembollerle ifade edilir. Bir şekil dilbilgisi, A ve B'nin şekiller $A \rightarrow B$ biçiminde gösterilen kurallardan oluşur (Stiny, 1977). Kural, tasarımda bir "A" ile sembolize edilen şeklin, "B" ile sembolize edilen şekil ile değiştirilebileceğini belirtir. "Etiketler" olarak bilinen semboller ise tanımlanın son işlemin hangi şeklin üzerinde olacağını ve gerçekleşecek işlemin hangi şekilde yapılacağını belirtir.

Biçim grameriyle tasarım süreçleri, formülasyon, yürütme ve keşif aşamaları olarak aşamalara ayrılabilir. Formülasyon aşamasında tasarımcı, tasarım amacına göre biçimler arasındaki ilişkileri formüle eder. Başlangıç biçim, biçim kuralları tasarım amacına göre formüle edilir. Tasarım süreci açık ve esnekler. Kurallar ve süreç açık olduğu için tasarım sürecindeki değişiklikler, daha algılanabilirler. Tasarımda yeniden mühendislik veya geri izleme gerektiğinde tasarımcı sürecine müdahale edebilir. İstenen adımda yeni kurallar tanımlanabilir. Yürütme aşamasında başlangıç şekli kural setlerine göre form değiştirerek diğer şekilleri oluşturur. Biçim grameri kuralları yinelemeli ve algoritmik bir yapıya sahip olduğu için yeni biçimlerin üretimini sağlar (Aksoy, 2001; Stiny, 1980; Stiny ve Gips, 1972). Keşif aşamasında tasarımcı, türetilen formlardan tasarım kurgusuna en uygun olanı seçebilir. Kural veya tekrar sayısının durumuna göre karmaşık ürünler elde edilebilir.

Biçim gramerleriyle soyut formlar üretilir. Tasarım, tasarımcının soyut formları yarattığı kurallara ve ilişkilendirdiği amaca göre anlam kazanır. Biçim gramerlerinde biçimlerin karşılığı olan anlamlar tanımlama şemaları ile ifade edilir. Dolayısıyla şekiller, ayrıca fonksiyonu olan elemanlardır (Özbek, 2004: 11). Bu anlamda iç mimar, biçim grameri ile yapacağı çalışmalarda yaratacağı kuralları, biçimlerle oluşturacağı ilişkileri planlanan tasarım amacıyla ilişkilendirebilir. Mimari tasarımcının türettiği formları amacına uygun bir mekâna veya herhangi bir iç mekân donatısına dönüştürebilmesi yeteneklerine ve seçimlerine bağlıdır. Bu anlamda mimari planlar oluşturmak için mimari öğeler belirlenen biçim grameri kurallarına göre manipüle edilir. Konutla ilgili işlevsel gereksinimler, odaların bağlam içindeki şekli ve yerleşimiyle ilgili tercihler odanın boyutuna, şekline, yönüne ve birbirlerine göre komşuluk ilişkilerine yansıtılır. Diğer yandan bir yapının mimari üslubu analiz edilip

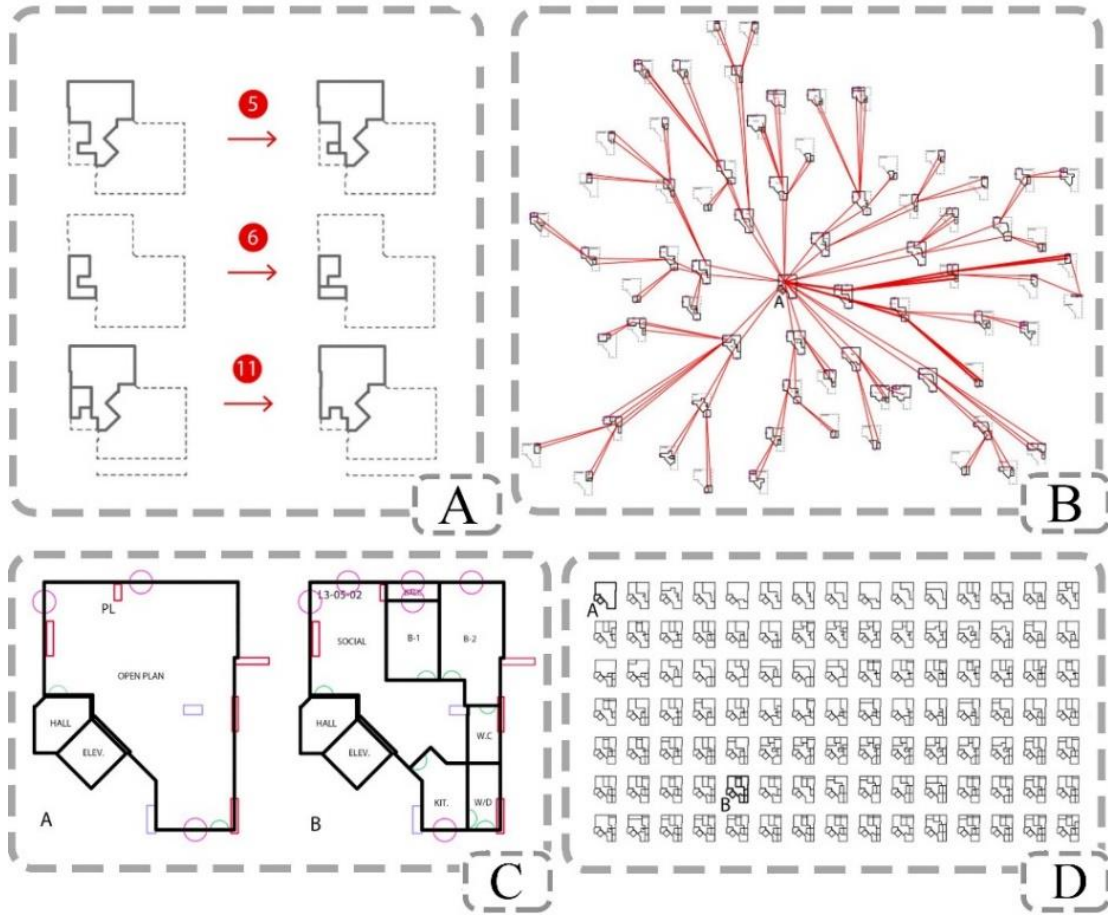
aynı tipolojide iç mekân kurguları inşa edilebilir (Eilouti ve Hamamieh Al Shaar, 2012; Güzelci, 2012; Torus, 2011) ya da sadece tasarım ihtiyaçlarına göre kurallar oluşturulabilir (Veloso, Celani ve Scheeren, 2018).

Şekil 2.8.'de biçim grameri, apartman yerleşimi planları üretiminde kullanılmıştır (Veloso ve diğerleri, 2018). A'da biçim gramerine dayalı olarak uzamsal alt bölümlenme ve birleştirme kuralları tanımlanmıştır. B'de kural setlerine göre oluşturulan şekil eylem grafiği, C'de kurallar tarafından oluşturulabilecek açık ve özel kat planı olmak üzere iki plan örneği, D'de oluşturulabilecek planlar gösterilmiştir.



Şekil 2.8. Bir yapının kütle organizasyonu üzerinde yapılan biçim grameri analizi. A'da başlangıç şekli, b'de kurallar, c'de evin kütleli gelişim şeması

Kaynak: (Saban, 2018)



Şekil 2.9. Biçim gramerinin apartman yerleşimi planları üretimi için kullanımı

Kaynak: (Veloşo ve diğerleri, 2018).

Bir diğer üretken sistem otomatlarıdır. Otomatlar, çevrelerinden kendilerine bilgi girdisini işler ve özellikleri, bu girdilere tepkilerini yöneten kurallara göre değiştirir. Hücresel otomatlar ve ajan tabanlı modeller, otomat araçlarının iki yaygın sınıfını oluşturur (Crooks ve Heppenstall, 2012).

Hücresel otomat, bir dizi basit matematiksel kurala göre şekillenen çeşitli hücreler topluluğudur (Ilachinski, 2001; Von Neumann, 1951; Wolfram, 1984). Hücresel otomat, ızgara ve hücrelerden oluşur. Hücreler, tasarım oluşturmak için belirlenen ızgaralarda, belirli kural setine göre iki ya da üç boyutlu olarak tanımlanır (Krawczyk, 2002). Izzaradaki hücrelerin durumu boş veya dolu olarak karakterize edilerek mevcut uzayı gereksinimlere göre tanımlayabilir. Hücreler, komşu hücrelerinin durumuna göre yönlendirilir. Bir sonraki hücredeki değerin belirlenmesi,

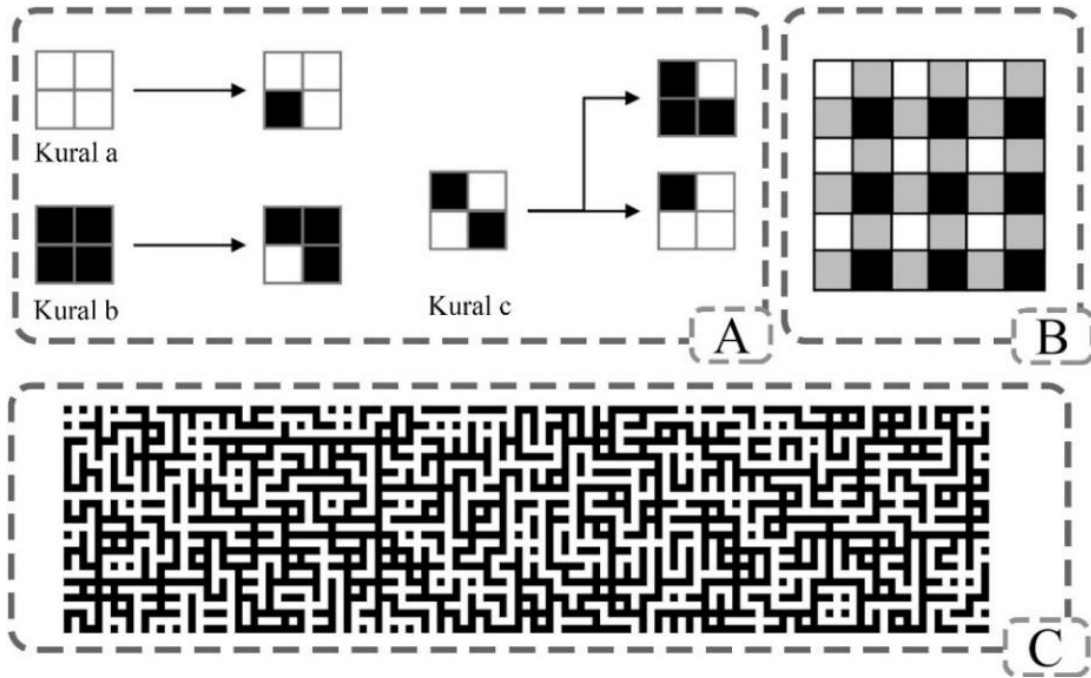
tanımlanan komşuluktaki hücrelerin değerlerine dayalı olarak gerçekleşir. İlgili kurallar, hücre durumlarını değiştirmek için belirlenir. Üretken süreç komşu hücrelerin durumları tarafından yönlendirildiğinden bağlama karşı duyarlıdır. Her hücre kendi bağlamına göre hareket eder ve bu etkileşim ile bir sistem yaratılır.

Hücresel otomatlar, karmaşık formları oluşturabilmek ve kısıtlamalarla başa çıkabilmek için hızlı bir şekilde çok sayıda çözüm üretilmiştir (Herr ve Ford, 2016; Khalili Araghi ve Stouffs, 2015). Matematiğin açıklamakta güçlük çektiği karmaşık sistemler, hücresel otomat ile türetilebileceği görülür. Hücreler zaman içinde yinelemeli olarak gelişerek tahmin edilmesi zor olan karmaşık modellere bürünür (Sarkar, 2000; Wolfram, 2002).

Hücresel otomat, mimari tasarımcılara esin kaynağı oluşturulabilecek formlar yaratabilir (Anzalone ve Clarke, 2003; Rucker, 2006; Silver, 2006). Hücresel otomatlar, mimari tasarım için genellikle karmaşık biçim ifadelerinin geliştirilmesinde, yapı formlarının temsillerinin oluşturulmasında kullanıldı. Hücresel otomatların kurallarındaki değişimlerin yanında hücrelerin şekilleri, komşuları ve durumlarındaki değişimler, mimari tasarımlara uyarlandı (Herr ve Ford, 2015, 2016). Hücresel otomatın form oluşturma yeteneğine rağmen, mimaride nispeten daha az uygulaması vardır. Mimari tasarım amacıyla kullanılan hücresel otomat uygulamaları, keşfedilebilecek potansiyel çözümler sağlasa da (Ford, 2013; Herr ve Ford, 2015, 2016; Herr ve Kvan, 2007; Khalili Araghi ve Stouffs, 2015), form oluşturma sürecinde geleneksel yöntemlere göre daha iyi bir seçenek olduğu söylenemeyebilir (Ford, 2013: 81).

Hücresel otomat, iç mekânda dekoratif motifler oluşturmada ya da mimari süslemelerin, desenlerin ve örüntülerin oluşturulması için kullanılabilir. Sabetfard ve Nadimi (2020), bir mimari süsleme türü olan Kare Kufi'nin üretilmesi için hücresel otomat modelini kullanmıştır. Öncelikle kalıp oluşturma kurallarını tanımlayabilmek için kare Kufi yazıların açık ve belirgin bir tanımı ve bunlarla ilgili kavramları içeren yazılı ve görsel kaynaklar gözden geçirmiştir. Kare Kufi desenlerinin ızgara tabanlı özü, onları hücresel otomat modellemesi için uygun bir örnek oluşturmuştur. Çalışmada Margolus komşuluğunu kullanan bir hücresel otomat algoritması

kullanılarak karmaşık tipte kalıplar üretmiştir. Desenler orijinal kare Kufi yazılarına görsel benzerlik elde etmeyi amaçlayan pikselleriyle modellenmiştir. Geliştirilen ilkelere dayalı olarak, bazı modeller oluşturulmuştur. Tüm desenler, her desende çizgiler yalnızca bir birim genişlikte olan kare bir ızgara üzerinde oluşturulmuştur. Her hücrenin durumu komşuları tarafından belirlenmiştir. Hücreler birbirleriyle etkileşime girerek, gerçek sanat eserlerini en iyi tanımlayan dinamik bir duruma dönüşmüştür. Basit kurallar uygulayarak karmaşık morfolojiler üretmiştir. Yöntem, basit kurallar uygulayarak doğru ve görsel olarak da kabul edilebilir örüntüler üretebilmiştir.



Şekil 2.10. A'da Hüresel otomat modelinin geçiş kuralları, b'de hücre başlangıç durumu, c'de algoritma tarafından oluşturulan bir model

Kaynak: (Sabetfard ve Nadimi, 2020).

Bir diğer otomat, ajan tabanlı modellerin, hüresel otomattan farklılıkları bulunur. Ajan tabanlı modellerdeki ajanlar, konumunu sabitleyebilmesinin dışında hücrelerin aksine konumunu hareket ettirerek çevrelerinde özgürce dolaşabilir ve hücrelerin aksine birden fazla öz niteliğe sahip olabilirler (Crooks ve Heppenstall, 2012; Klügl ve Bazzan, 2012). Ajanlar, gerçek dünya ile eşlemek üzere, doğal ya da yapay her türlü özerk varlığı temsil edebilir. Öğrenme ve adaptasyon dâhil olmak üzere karmaşık davranışları sergileyebilirler (Bonabeau, 2002; Gilbert ve Terna, 2000:

11,12). Ajan tabanlı modeller, performanslarını en üst düzeye çıkarabilecek davranışları keşfederek öğrenen ajanlara hitap eden pekiştirmeli öğrenme ile ilişkilendirilir (Bölüm 2.4.7.).

Ajan tabanlı modellerde, ajanların etkileşimleri, çevrenin konfigürasyonuna ve kurallara göre değişebilse de bazı ortak özellikleri değişmez (Crooks ve Heppenstall, 2012; Klügl ve Bazzan, 2012). Ajan tabanlı modellerin bileşenleri, ajanlar, çevre ve kurallardır. Ajan tabanlı model, bir model veya simülasyon ortamında yer alan etkileşimli ajanlardan oluşur. Ajan tabanlı modellemedeki ajanların iki temel özelliği, özerklik ve sosyal yetenekleridir. Bir ajan, tasarım hedeflerini karşılamak için özerk eylem yeteneğine sahiptir; kullanıcı müdahalesi olmaksızın talimatları yerine getirebilir ve kararlar alabilir (Hayes, 1999; Jennings ve Sycara, 1998). Ajan çevreyle etkileşim halindedir, etki alanındaki değişiklikleri algılar ve bunlara tepki verir (Genesereth ve Ketchpel, 1994). Ajanlar, bir dizi hedefi veya görevi gerçekleştirmek için önceden tanımlanmış parametreler aracılığıyla karmaşık dinamik çevrelerde özerk olarak algılar, hareket eder ve diğer ajanlar ile etkileşime girer. Ajanlar bu etkileşime dayalı olarak zaman içinde değişir. Çevrede yaşayan ve zaman içinde değişebilen bir dizi özelliğe sahip olan öğeleri ve belirli davranışları tahmin etmek için ya da tasarımların performanslarını değerlendirmek için kullanılabilir. Ajanlar, ulaşılması gereken hedefe yönelik olarak çevreleriyle ve diğer ajanlarla etkileşimli bir şekilde hareket ederek diğer varlıklar, engeller veya gerekli hedefler hakkında bir farkındalık sağlarlar.

Bu anlamda ajan tabanlı modelin formülasyon aşamasında ajanların doğrulanması, modelin belirli bir bağlama göre ince ayarlarının yapılması ve onaylanması ile ilgili hususlar gerçekleştirilir. Tasarımcı, hedefinin gerçekleştirmek için ajanların başlangıç durumunu, yerlerini, tepki verici davranışlarını ve çevrenin özelliklerini kodlar. Daha sonra ortamdaki ajanların zaman içinde etkileşimi (simülasyon) yürütülür. Ajanların çevre tarafından koşullandırılan eylemleri, üretken modelin yürütülmesiyle gerçek dünya hakkında toplu davranışlar veya oluşumlar canlandırılabilir. Böylece tasarımcı, ajanlar arasındaki etkileşimlerden gelecek hakkında varsayımlarda bulunulabilir.

Ajan tabanlı modeller ya da simülasyonlar, insan hareketlerinin önemli zamansal dinamiklerini ortaya çıkararak kullanıcılar için uygun iç mekân düzenlemelerinin tasarlanmasında yardımcı olabilir. Simüle edilmiş ajanlar, analizi yeniden üretimi veya tahmini yapılacak bir fenomeni veya senaryoyu üretmek için kullanılabilir (Klügl ve Bazzan, 2012). Örneğin, Scheutz ve Mayer (2016), ofis çalışanlarının daha verimli iletişim kurabildikleri, yaratıcı olabildikleri ve üretken bir performans gösterebildikleri bir ofis tasarlamak için büyük veri analiz yöntemleriyle birlikte ajan tabanlı modellerin simülasyonlarını önermiştir. Büyük veri analizinin ofis iç mekânının tasarımını bilgilendirmek için ajan tabanlı modellerin potansiyelini artırabileceği görülmüştür. Büyük veri analizi, çalışanların veriminin artırılması için birçok konuda gerekli bilgiler, uygun bir ofis tasarımı için belirleyici rol oynayabileceği düşünülmektedir. Bu bilgiler, mevcut alanın boyutlarını, çeşitli amaçlar için ayrılmış alanların yüzdesini, iç organizasyonu, çalışanların ofis içi hareketlerini (masa paylaşımı, tele-çalışma, çalışanların yoğun dolaşım yolları) kapsar. Ajan tabanlı simülasyonlar ise gelecekteki olası çalışan davranışları ve olası mimari kurulumların gözlemlenmesi için deneysel bir ortam olarak kullanılır. Çalışanları temsil eden, belirli bir alana yayılmış çeşitli konumlara sahip ajanlar kullanılarak gelecekteki çalışanların davranışları, ofis alanları içindeki yörüngeleri ve alanların işlevsel rollerindeki değişikliklerin etkileri etmen tabanlı modelleme ve simülasyonlar ile ortaya çıkarılır.

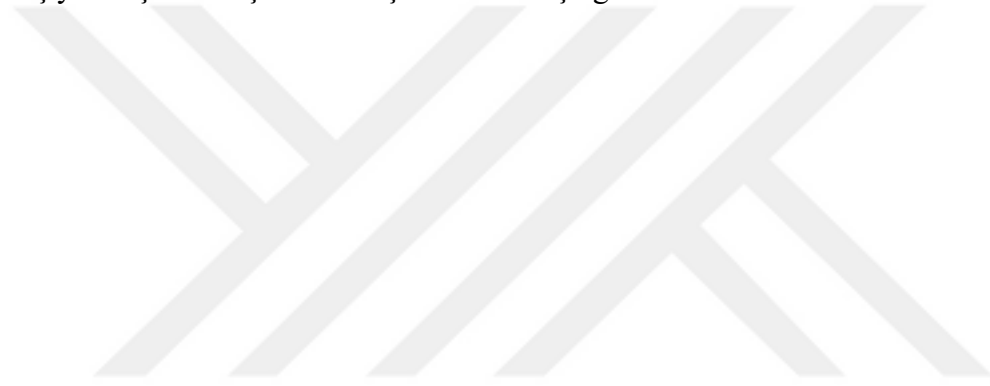
Blecha, Mikulecky, Tucnik ve Matyska (2017), akıllı ev ortamlarının modellenmesi için, bir dairenin akıllı iç ortamını temsil eden ajan tabanlı modellerde senaryolar doğrulamıştır. Diğer yandan Fouad ve El Antably (2022), ajan tabanlı modelleme simülasyonlarını kullanarak COVID-19 yayılımının iç mekân tasarımını etkileyebilecek ipuçları edinmiştir. Çalışmada mekânsal davranış ve hastalık bulaşma modellerini modellenip, çeşitli parametreler manipüle edilerek yapılan simülasyonlarda alternatif mekânsal davranışların hastalık yayılımını nasıl etkileyeceğiyle ilgili fikirler edinmişlerdir. Okul düzenlemeleri ve öğrencilerin mekânsal davranışları, ajanların mekânsal davranışlarının ana itici güçleri olmuştur. Öğrencileri ve öğretmenleri temsil eden ajanlar, hastalık yayılımını etkileyen hareket ve aktivite seviyesi parametreleri olarak soyutlamıştır.

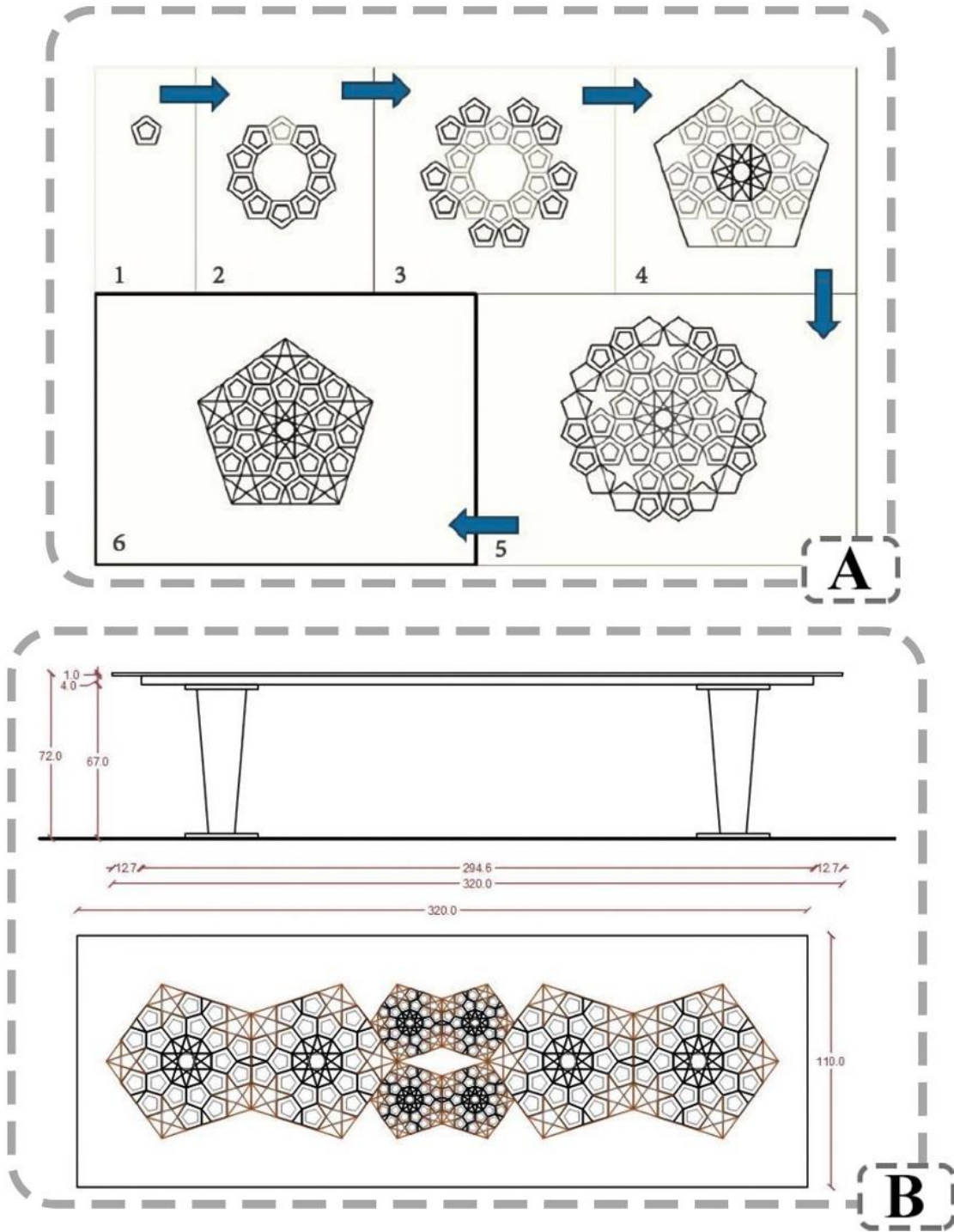
Bir diğerk üretken sistem fraktallardır. Düzensiz veya parçalanmış anlamına gelen fraktal (fraktus), Öklid geometrisiyle çözülemeyen durumlar için daha kapsamlı açıklamalar getirir (Mandelbrot, 1982). Öklid geometrisinden farklı olarak özel bir şekilde ya da ölçekte değildir, doğadaki formlara uygulanabilir ve bir algoritma ile tanımlanabilir (Sarıyıldız, 1991). Asıl olarak matematik disiplinine ait olan fraktalın mimarlıktaki anlamı, matematiksel anlamından biraz farklıdır.

Fraktal geometri iki anahtar bileşenden oluşur: birbirini takip eden aynı şekil ve kalıplar olan kendine benzerlik ve bu kalıp ve şekillerin ayrıntı düzeyini veren fraktal boyut (J. Burry ve Burry, 2010). Fraktal boyut, fraktal geometrinin ne kadar düzensiz ve karmaşık olduğuyula ilgili sayısal bir ölçüt oluştururken; özyineleme, başlangıçtaki geometriye benzer olan çeşitli ölçeklerde nesnelere yeniden üretir (Ostwald ve Vaughan, 2016). Böylece fraktal geometri, özyineleme yoluyla büyüme ya da daha küçük birimlerinde bölünmeler yoluyla yapılar oluşturur (J. Burry ve Burry, 2010). Fraktal geometride, başlangıçtaki temel biçim, üretim kurallarına göre yinelenerek bu başlangıç şekline benzer örüntüler ve şekiller oluşturulur (Çağdaş, 1994). Bu anlamda objelerin birbiriyle uyumlu olduğu varyasyonlar üretilebilir. Fraktal geometride, temel bir başlangıç formundan ve aynı algoritmadan varyasyonlar üretilebildiği için üretilen formlar arasında uyum ve bütünlük duygusu oluşur. Bu uyum, mekân tasarımında varyasyonlar üretmek için kullanılabilir. Mimari yapılar da fraktalda olduğu gibi büyük ölçekten küçük ölçeklere kadar kendine benzer biçimlerden oluşabilir (Bovill, 1996). Bu anlamda mimari organizasyonlar için fraktal kurgular geliştirilebilir, fraktal geometri ile mekânı oluşturan öğeler veya işlevler ilişkilendirebilir (Ediz, 2003; Turhan, 2018). Fraktallar, doğadaki detayların sürekliliğini, karmaşıklığını, çeşitliliğini aynı oranda mimari yapıya yansıtma imkânı tanıyabilir (Ediz, 2003).

Fraktallar son birkaç on yıldır mimarideki ulaşım ağları, sıhhi tesisat hatları, bina planları ve kotları, parseller, bloklar, mahalleler gibi karmaşık altyapıları analiz etmede faydalı olmasının yanında verimli ve sağlam yapısal formlar tasarlamak için bir yöntem olarak geliştirildi (Driscoll, 2019: 44-50). Fraktallar mimari tasarımda bir tema, gelişim ve çeşitlilik yaratmak, kentsel ortamları ve binaları analiz etmek ve

onları yaratmak için kullanıldı (Driscoll, 2019). Estetik çekiciliği ve doğal nesnelere betimlemedeki ustalığı nedeniyle günümüzde, iç tasarımda ve mobilya tasarımında da ilham verici bir araç olarak kullanılmıştır (İbrahim, Kamel ve Khamis, 2021; Mahmoud, 2019). Doğadan ilham alan fraktal şekli kullanan bir mobilya örneği için tasarım süreci adımları planlanmıştır (İbrahim ve diğerleri, 2021). Şekil 2.11.'deki mobilya tasarımı için ilk olarak orta kısmı boş olan bir beşgen oluşturulur. Şeklin yinelenen, dönen kopyaları bir merkez noktası etrafında oluşturulur. Pentagonun daha fazla kopyası eklenir. Dış çizgileri uzatarak daha büyük bir beşgen oluşturulur. Şekli tamamlamak için kenarlara daha fazla beşgen eklenir ve büyük beşgenin iç açılarında beş yıldız şekli oluşturulacak şekilde fazla çizgiler kaldırılır.





Şekil 2.11. A’da fraktal şekillerin oluşturulma adımları ve b’de, oluşturulan fraktal şeklin yemek masası tasarımında kullanım örneği

Kaynak: (İbrahim ve diğerleri, 2021: 83).

Doğadaki formları üretici algoritmalara göre tanımlayan bir diğer önemli araç L-sistemleridir. L sistemleri Aristid Lindenmayer (1968) tarafından bitki taksonomilerini tanımlamak için geliştirilmiştir. L-sistemleri, genel olarak belirli fraktalları tanımlamak için de kullanılabilir. L-sistemler de fraktallar gibi tekrarlayan desenler ve doğal organik formlar oluşturmada başarılıdır. L-sistemleri ve fraktallar, diğer kural tabanlı modellerde olduğu gibi belirli kuralları olan içsel sınırlamalar vasıtasıyla belirli bağlamlar ve amaçlar için formüle edilirler (Ei-Khaldi, 2007). L-sisteminin amacı, sistemin ana konseptini oluşturan yeniden yazma sistemlerini kullanarak basit bir başlangıç nesnesinin parçalarını art arda değiştirerek karmaşık nesnelere tanımlamaktır (Chan ve Chiu, 2000; Prusinkiewicz ve Hanan, 1989; Prusinkiewicz ve Lindenmayer, 1996).

2.4.3.2. Üretken Tasarımın Hesaplamalı Tasarım ve Diğer Hesaplamalı Yaklaşımlar ile İlişkisi

Üretici sistem tasarımcıya tasarımlarını yaratması için bir araç olarak hizmet eder. Bu aracı yaratırken çoğu zaman hesaplamalı düşünülür dolayısıyla hesaplamalı tasarlanır. Hesaplamalı tasarım nihayetinde potansiyel üretimlerin olması için kendi aracımızı yaratmayla ilgilidir. Dolayısıyla üretken sistemler (üreticiler) hesaplamalı tasarımın sonucudur.

Temel anlamıyla üretkenlik başlı başına kapsayıcıdır. Dolayısıyla üretken mimari tasarım, tasarımcıların mimari tasarım problemlerine uygun çözümler üretmek için üretken sistemlerden yararlandığı ve böylece daha büyük çözüm alanlarını keşfederek referans çevrelerini genişlettikleri bir şemsiye terim olarak kabul edilir (Azadi ve Nourian, 2021; Dino, 2012; Mukkavaara ve Sandberg, 2020; Singh ve Gu, 2012; Veloso ve Krishnamurti, 2021).

Üreticinin seçiminde önemli kriterler; kullanıcıların üretici ile nasıl etkileşime girdiği, tasarım probleminin niteliği, çözüm uzayının keşfinin optimizasyon ihtiyaçları ile nasıl dengelendiğidir (Mukkavaara, 2021: 15-18). Üretici, tasarım üzerinde kontrolün daha yüksek olduğu parametrik ve algoritmik tasarım yetenekleri üzerine kurulu olabileceği gibi, örtük tasarım alanını kullanan hücresel otomat veya evrimsel

tasarım gibi tasarım prosedürünün izlenebilirliği mümkün olmayan yöntemler üzerine de kurulu olabilir. Parametrik modellerde tasarım alanı tanımlanıp sınırlanırken, kural odaklı üretken modellerde tasarım alanı uygun kurallar ve ön koşullar ile belirsiz olarak genişletilebilir (Veloso ve Krishnamurti, 2021). Diğer yandan verilerle eğitilen makine öğrenimi modelleri ile veriler ile iletişime geçebiliriz (Bölüm 2.4.7.).

Üretkenlik, yinelemelerin otomatikleştirilmesiyle verimli hâle getirilir. Belirli bir şekle ulaşmayı sağlayan hesaplamalı tasarım aracı (örneğin parametrik/algoritmik model), değerlendirme sonuçlarıyla şekli belirleyecek analiz araçları ve modeldeki işlemi kullanıcının yerine gerçekleştirerek potansiyel tüm alternatifleri üreten optimizasyon araçları (örneğin genetik algoritmalar kullanan optimizasyon araçları) ile birlikte çalışabilir. Optimizasyon araçları, yinelemeli tasarım tekniklerini kullanarak tasarımları iyileştirebildikleri gibi fikir üretmek ve çözüm alanını keşfetmek için de kullanılabilirler (Bölüm 2.4.5.). Böylece birçok tasarım alternatifi üretilerek modelin üretkenliği öne çıkarılabilir.

2.4.4. Performansa Dayalı Yaklaşım

“Performans” kavramı, 1940’larda ve 1950’lerde başlayan ve geniş sonuçları olan bir dizi entelektüel çabadan evrilmiş çok kapsamlı bir kavramdır (Hensel, 2010; Kolarevic, 2005). Performans modelleri, diğer paradigmatik sınıflardan farklı bir yöntem kümesi oluşturmaz (Oxman, 2006). Dolayısıyla geniş bir yelpazedeki tasarım uygulamalarını tanımlayan genel bir terimdir.

Mimari tasarımda performans, yapının performansının vurgulandığı bir ilkeler bütünü ya da bir tasarım felsefesi olarak düşünülebilir (Hensel, 2010; Shi, 2010). Performans, bir tasarım çözümünün öngörülen davranışının arzu edilirliliğinin bir ölçüsüdür (Kalay, 1999). Mimari yapının performansı, “binanın insan ihtiyaçlarını ve çevresel faktörlerle ilişkili olarak mimari gereksinimlerini yani bağlamı karşılama kapasitesidir” (Turrin, 2014: 54). Bu anlamda performans, mimari tasarım için yol gösterici bir tasarım ilkesidir (Kolarevic, 2003). Performansa dayalı bir hesaplamalı tasarım modeli de istenen bir performans tarafından yönlendiren süreç tasarımıdır. Bu anlamda mimari form, bir tasarım nesnesinin istenen performansına veya davranışına göre oluşturulur.

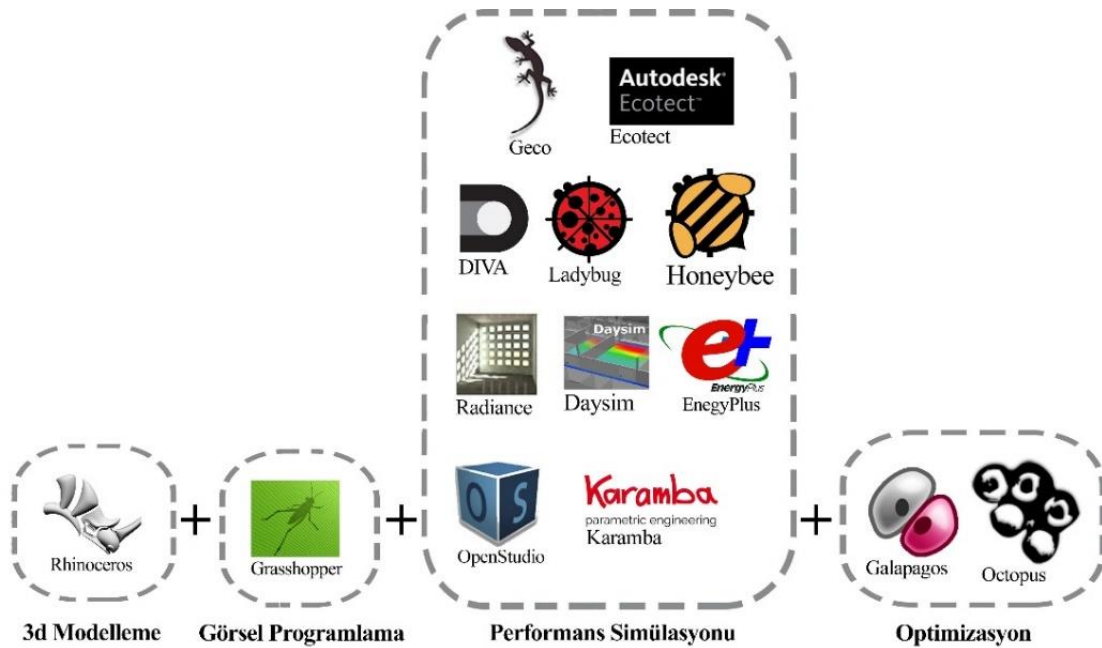
2.4.4.1. Performansa Dayalı Tasarım Modeli ve Yöntemleri

Performansa dayalı modeller genel olarak simülasyondan yararlanır. Simülasyon, önceden belirlenmiş çevresel koşullar altında bir dijital modelin performansını tahmin etme yöntemidir. Simülasyon araçları, yapının davranışını veya tepkisini simüle ederek beklenen performans kriterlerini tahmin eden ve gözlemlenebilir çıktıları performans göstergelerinin uygun nicelikleriyle eşleştirilmesini amaçlayan araçlardır (Becker, 2008; Malkawi ve Augenbroe, 2004). Bu araçlar, tasarımcının yaptığı değişikliklerin sistem performansı üzerindeki etkisini gözlemleyebilmesini sağlar. Simülasyon yazılımları, çevresel, finansal, mekânsal, sosyal, kültürel, ekolojik veya teknolojik verilerin değerlendirilebilmesini sağlar (Leatherbarrow, 2005). Bütün bu veriler, belirli bir bağlamdaki kuvvetleri ifade eder. Mimaride performansa dayalı veya performatif tasarım, bu bağlam içerisindeki çeşitli verilerle birbirini aynı anda ve sürekli olarak etkileyen birbirleriyle ilişkili yapılar ağı oluşturmayı ifade eder. Performansa yönelik tasarımın paradigmatik çekiciliği, tam da bu ilişkili anlamların çokluğunda yatmaktadır ve sıklıkla çatışabilen bu anlamları, performans hedefleri için bir şekilde uzlaştırmak, performansa dayalı tasarımdaki temel zorluklardan bazılarıdır (Kolarevic, 2004: 45, 2005).

Mimari tasarımcıların genellikle erken tasarım aşamalarında kullandıkları bina simülasyonların istatistiksel modeller, optimizasyona dayalı yöntemler gibi birçok farklı yöntemle kullanılmıştır (Østergård, Jensen ve Maagaard, 2016). Ayrıca performans, hesaplamalı modelin kendisinde de saklı olabilir. Örneğin Gaudi, asılı zincir modelinde yapısal performansı incelerken, bu modelde performans ölçütlerini modelin kendisinde saklı olan Hooke yasasından almıştır. Ancak günümüzde genel olarak performansa dayalı mimari sistemlerin üretken veya parametrik bir model, bina performansı simülasyon motoru ve optimizasyon araçlarıyla kullanıldığı görülür. Üretken veya parametrik bir model, bir dizi prosedür kuralından veya algoritmik süreçten çok sayıda tasarım alternatifi üretebilir. Bina performansı simülasyon motoru her bir tasarım alternatifinin uygunluğunu değerlendirir ve optimizasyon yöntemi hedeflere daha yakın olan tasarım alternatiflerini yinelemeli bir şekilde arar.

Günümüzde mimari tasarımcılar performans modellerini daha çok Rhino ve Revit gibi modelleme araçları, Grasshopper, Dynamo gibi komut oluşturma araçları, çeşitli bina performans ölçüm araçları ve optimizasyon araçlarıyla entegre olarak kullandığı görülür (Şekil 2.12.). CAD sistemi belirli parametreler ve boyutlar ile geometrik ilişkiye dayalı bir model çizme yollarını kolaylaştırır. Parametrik analiz, geniş alternatif popülasyonlarının analizini kolaylaştırmak için performansa dayalı tasarım yöntemlerini bu üç boyutlu modelleme aracı ve parametrik modelleme aracı ile birleştirir.

Mimari tasarıma yön verecek çevresel ve iklimsel veriler, tasarım süreci boyunca hesaplanabilir değişkenler olarak kullanılabilir. Grasshopper'da ve birçok eklentisi sayesinde doğal aydınlatma, yapay aydınlatma, termal konfor verileri, havalandırma ve akustik gibi yapı fiziğini etkileyen verilere göre optimum tasarımlar yapılabilir (Anderson, 2014, Bazafkan, 2018, Eltaweel ve Su, 2017). Bu manada estetikten ödün vermeden sürdürülebilir ve ekolojik tasarımlar üretilebilir. Tüm bu eklentiler birbirleriyle etkileşime girebilir, buna bağlı olarak birden çok amaç için daha doğru ve optimum sonuçlar alınabilir.



Şekil 2.12. Birbirlerine entegre şekilde kullanabilecek programlar örneği.

Bir yapının tasarım katmanlarının ve sürdürülebilirlik ölçütlerinin anlaşılıp optimize edilmesi için bütünsel bir bakış açısı gerekir. Bu anlamda performansa dayalı hesaplamalı tasarım optimizasyonunun (Bölüm 2.4.5.) önemi ortaya çıkar.

İç mimarlar optimize edilmiş sürdürülebilir çözümlerin bulunması için iç bileşenleri koordine ederek performansa dayalı bina tasarımı için diğer disiplinlerle birlikte işbirlikçi bir rol üstlenebilir (Gökmeral, 2014: 175-193). Bu anlamda, iç mekân konforunu etkileyen havalandırma, aydınlatma, enerji kullanımı, malzeme performansı gibi konularda iç mimarın kararlarıyla beraber amaca yönelik performans dayalı modeller oluşturulabilir. Bina tasarımında, iç mimarların iç mekân konforunu etkilemeye yönelik kararları bütünlük bir tasarım için değerlidir (Zimmerman, 2006).

İç mekânda doğal aydınlatmanın verimli kullanımı önemli performans problemlerinden biridir. Yapay aydınlatma kullanımında yani birincil enerji tüketiminde önemli bir azalma sağlamak için gün ışığından optimum şekilde yararlanmak gerekir. İç mekân gün ışığı performansı için coğrafi konum, yerel iklim, gün ışığı mevcudiyeti, bina geometrisi, pencere özellikleri, pencere-duvar oranı ve yönü gibi bir dizi parametreden etkilenir (Yu ve Su, 2015). Ayrıca gün ışığının kullanımı, bütünsel bir bakış açısıyla düşünüldüğünde yapının çevreyle ilişkisi, cephelerin tasarımı ve yapının mühendisliği ile alakalı olduğu gibi; iç mekan tasarımı odağında düşündüğümüzde iç mekândaki formlar, işlevler, yönlendirmeler ve malzemeler ile doğrudan ilişkilidir. Tavan yüksekliği, pencere ışık geçirgenliği, dış gölge uzunluğu, pencere genişliği gibi parametreler, güneş kazanımları, iletilen kazançlar ve kayıplar üzerinde büyük bir etkiye sahiptir (Futrell, Ozelkan ve Brentrup, 2015). Performansa dayalı hesaplamalı tasarım optimizasyonu modelinde, bu parametrelerin kontrol edilebilmesi, değiştirilebilmesi ve uygun sonuçların bulunması sağlanır. İç mekân gün ışığı performansı incelemeleri; gün ışığı katsayısı, gün ışığı özerkliği ve benzeri gibi gün ışığı mevcudiyeti metriklerini kullanan yazılım simülasyonu ve saha ölçümü yoluyla yapılmıştır (Futrell, Ozelkan ve Brentrup, 2015).

Performans kriterleri birbirleriyle ilişkilidir. Tasarımın erken aşamalarında gün ışığı ve enerji tasarrufu performansını, termal konforu ve görsel konforu iyileştirebilir. Diğer yandan konforun çeşitli yönleri aynı anda maksimize edilemez (da Graça,

Kowaltowski ve Petreche, 2007). Gün ışığını ve termal performansı iyileştirmek için bir parametreyi bir yönde değiştirmek, diğer amaç ile bir değiş tokuşla sonuçlanır, ancak uzlaşmacı çözümler bulunabilir.

Optimum gün ışığı ve doğal ısı sağlamak için binanın doğru gölgeleme cihazlarını ve yönlendirmesini ayarlama stratejileri dahil edilmiştir (Eltaweel ve Su, 2017). Parlamayı önlemek, yararlı gün ışığını filtrelemek için panjur gölgeleme kullanılmıştır. Hareketli panjur tasarımı, hem ısıtma ve soğutma için kullanılan enerjisinin azaltılması hem de gerekli gün ışığının verimli kullanımı için önemlidir. Bu anlamda birçok çalışmada, gün ışığının verimli kullanımı, termal konfor, görsel konfor ve enerji tüketimi açısından panjur, pencere, cephe, ışıklık, gölgelik, fotovoltaiik parametreleri optimize edilmiştir (Eltaweel ve Su, 2017). Panjur tasarımı için, pencere-duvar oranı, pencere genişliği, duvar alanı, perde panjuru sayısı, perde derinlik oranı ve perde eğim açısı, panjur aralığı ve elek derinliği gibi birçok faktör göz önünde bulundurulmuştur (Wagdy ve Fathy, 2015).

2.4.4.2. Performansa Dayalı Yaklaşımın Hesaplamalı Tasarım ve Diğer Hesaplamalı Yaklaşımlar ile İlişkisi

Performans simülasyonlarındaki çok sayıda tasarım parametresi ve aralarındaki ilişkilerin karmaşıklığı hesaplamalı tasarımı mecbur kılmıştır (Ercan ve Elias-Ozkan, 2015). Performansa dayalı tasarımda, belirlenen performans hedefine ulaşılması için bir hesaplama modelinden yararlanılır. Parametrik ve üretken modeller, çeşitli optimizasyon algoritmaları ve makine öğrenimi teknikleri optimize edilmiş bir performans sonucuna ulaşmak için entegre olarak görevler üstlendiği görülmektedir.

Performans kavramı, simülasyon ve optimizasyon ile doğrudan ilişkilidir. En iyi performans hedefine ulaşmak için en iyi performans gösteren optimum sonuç aranır. Bağlamdaki koşullara uyumluluğu ölçmek için o bağlamın belirli kuvvetlerini yansıtan simülasyon oluşturulur ve böylece çalıştırılan simülasyon, form üretimi için belirleyici faktör haline gelir (Oxman, 2008c). Form, bir tasarım nesnesinin istenen performansına veya davranışına göre optimizasyon algoritmaları ile optimize edilir. Bu nedenle mimari hesaplamalı tasarım modellerinde de bir aday tasarımı optimize

etmek için genellikle bina performans simülasyonu ve optimizasyon teknikleri ile birlikte kullanıldığı görülür.

2.4.5. Optimizasyona Dayalı Yaklaşım

Latince “en iyi” anlamına gelen “optimus” kelimesinden türetilen optimizasyon terimi, mimaride mimarlığın ölçülebilir özelliklerine ve neyin optimize edildiğine karar vermeye odaklanır (Canestrino, 2021). Optimizasyon işlemi mevcut alternatif çözümler içerisinde amaca uygun en iyi ya da en iyileri bulma işidir. Mimari tasarımda optimizasyon, mevcut kaynaklardan en iyi şekilde yararlanmak için matematik, bilgisayar bilimi, yapısal tasarım, mühendislik, simülasyon ve birçok hesaplamalı tasarım yöntemini içeren disiplinler arası bir konudur. Mimari tasarımda optimizasyonu mühendislikteki ve diğer disiplinlerdeki kullanımından ayıran temel fark ise erken tasarım aşamalarında tasarımların geometrik varyasyonlar ile optimize edilmesidir (Ekici, Çubukçuoğlu, Turrin ve Sarıyıldız, 2019). Mimari tasarım hesaplanan optimum, genellikle son ürün değil, tasarım araştırması için başlangıç noktası olarak kullanılır (Bradner, Iorio ve Davis, 2014; Wortmann, Cichocka ve Waibel, 2022).

Optimizasyon, alternatiflerden en verimlisine yaklaşmaya yardımcı olur (Radford ve Gero, 1980). “En iyi çözüme yaklaşmak” daha doğru bir ifade olabilir, çünkü optimizasyon algoritmaları genellikle en iyi çözüme götürmek yerine, tasarımcılara çözüm alanı hakkında fikir verir (Stouffs ve Rafiq, 2015). Optimizasyon algoritmaları büyük bir tasarım alanının otomatik olarak araştırılmasıyla ya da alternatif çözümleri bulmakla ilgilenir (Attia ve diğerleri, 2013). Dolayısıyla süreci otomatikleştirerek insan beyninin bilişsel sınırlamalarını aşacak boyuttaki bilgiyi değerlendirmeyi sağlar. Sonucunda yakınsanan (iyileştirilmiş) çözümler yaratılır.

2.4.5.1. Optimizasyona Dayalı Modeller ve Yöntemler

Mimari tasarımda optimizasyon yöntemleri evrim geçirmiştir. Bu evrimin sonucunda Gaudi, Otto ve Isler gibi öncülerin fiziksel form bulma modellerden matematiksel denklemlere, matematiksel olarak hesaplamının sürekliliğinden hesaplamının ayrıklaştırılmasına ve türev yöntemlerinden türevsiz kara kutu modeline

geçilmiştir (Canestrino, 2021). Optimizasyon sürecinde fiziksel form bulma modellerinin hem nihai formun hem de başlangıç parametrelerinin optimumunu aynı anda garanti edememesi, yüksek düzeyde kesinlik sağlayan matematiksel optimizasyon süreçlerinin otomatikleştirildiği dijital yöntemlere dayalı optimizasyona yol açmıştır (Canestrino, 2021). Daha karmaşık olan çok amaçlı optimizasyon talebi ise simülasyon süreçlerine dayalı sayısal veya yinelemeli yöntemlerle optimal tasarımı aramaya itmiştir. Metasezgisel yöntemlerin egemen olduğu mimari tasarımdaki güncel çalışmalarda evrimsel optimizasyon süreçlerinde olduğu gibi büyük miktarda veri verimli bir şekilde işlenerek türetilmiştir (Wortmann ve Nannicini, 2017). Metasezgisel algoritmalar ile optimizasyon modelindeki değişken ve kısıtlamaların fazla olduğu dolayısıyla çözüm uzayının büyük olduğu zor ve karmaşık problemlere kısa sürede çözümler üretilmiştir (Asadi ve Geem, 2015).

Optimizasyon algoritmaları için tasarım değişkenleri, amaç fonksiyonu ve kısıtlamalara göre farklı sınıflandırmalar yapılabilir (Sahab, Toropov ve Gandomi, 2013; Yang ve Koziel, 2011). Kısıtlamalar, fonksiyonlar ve değişkenlerle sistem davranışını tanımlar. Hiçbir amacın olmadığı durumda, yalnızca kısıtlamalar olabilir. Mimari tasarımla ilgili performans kriterleri bazen amaçlardan çok kısıtlamalar olarak ifade edilebilir. Böyle bir problemde herhangi bir uygun çözüm optimal bir çözümdür.

Optimum çözümler amaç fonksiyonuna göre yaratılır. Optimizasyon problemlerinde maksimum ya da minimumunu bulmak için amaç fonksiyonu minimize veya maksimize edilmeye çalışılır (Haupt, 2003). Bir mekânın enerji kullanımı, bir mekânın inşaat maliyeti gibi unsurlar minimize edilebilir ya da bir kâr, sonuç, başarı puanı gibi kriterler maksimize edilebilir. Dolayısıyla amaç fonksiyonu; iyileştirilmiş bir çözümün bulunmasını sağlayan, bir tasarımın performansının değerlendirilmesini sağlayan fonksiyondur. Optimizasyon problemleri, amaç fonksiyonlarının sayısına bağlı olarak tek amaçlı veya çok amaçlı olarak sınıflandırılabilir. Tek amaçlı algoritmalar, tek bir amaç fonksiyonunu optimize etmeyi amaçlarken çok amaçlı algoritmalar aynı anda birden fazla amaç fonksiyonu dikkate alır. Tek amaçlı problemlerde, minimize veya maksimize edilecek sadece bir uygunluk fonksiyonu olduğu için optimizasyon sürecinin sonucu nihai tasarım kararı olarak

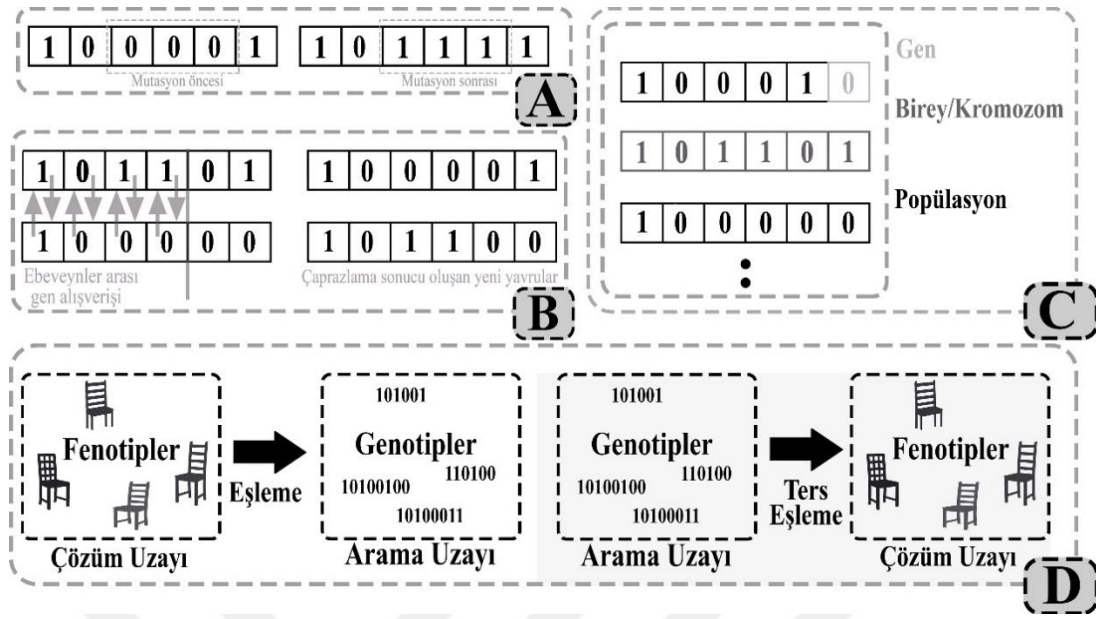
kullanılabilir. Öte yandan çok amaçlı optimizasyon problemleri, tek amaçlı optimizasyon problemlerine göre daha zor problemlerdir. Özellikle birbiriyle çelişen amaçların olması durumunda ayrılan bütçe ve zaman daha da önemli hâle gelebilmektedir.

Tasarımcılar birbiriyle rekabet eden birçok amacı bir arada çözmeye çalışır (Keough ve Benjamin, 2010). Mekânın tasarımında, aynı anda birden fazla karmaşık hedefi göz önünde bulundurmak gerekebilir. Bu hedefler, birbirini çeşitli yönlerden etkileyen değişken ve genellikle kısıtlamaları içerir. Mimari tasarımcılar, mekânın enerji kullanımı, maliyetleri, yapısal tasarımı, konforu vb. gibi birden çok performans hedefinin nasıl optimize edileceğiyle ilgilenir. Bu gibi birden fazla amacın olduğu durumlarda, çok amaçlı optimizasyon kullanılır. Mimari tasarımların doğası bu tür problemlere eğilimlidir (Cichocka, Browne ve Rodriguez, 2017: 392; Ekici ve diğerleri, 2019: 367; Wortmann, Cichocka ve Waibel, 2022). Örneğin, mekânın gün ışığını en üst düzeye çıkarılırken, parlama en aza indirilebilir (Wortmann, Costa, Nannicini ve Schroepfer, 2015). Bunun gibi çelişen hedefleri dengelemek için çok kriterli optimizasyon yaklaşımları tanıtılır. Bununla birlikte, çok amaçlı optimizasyon, mimari tasarımda devam eden bir araştırma alanını oluşturur.

Bir optimizasyon süreci genellikle amaç fonksiyonunun birçok kez, çoğu zaman binlerce ve hatta milyonlarca konfigürasyonun değerlendirilmesini içerir (Yang ve Koziel, 2011). Parametrelerin kademeli olarak değiştirildiği deneme yanılma süreci ile manuel aramalar zaman alıcı ve etkisizdir. Uygun optimizasyon yöntemlerinin kullanılması, çok karmaşık ve zaman alıcı simülasyonlar gerekli olduğunda bile verimli tasarım keşfine izin vererek hedefe ulaşmada önemli bir rol oynadığı görülür. Birçok optimizasyon algoritması bu süreci verimli hale getirmek için icat edilmiştir. Algoritmanın performansına, kullanılan problemin tipine ve matematiksel tanımına bağlı olarak değişen kendilerine özgü terminolojileriyle birlikte birçok optimizasyon yöntemi bulunur (Machairas ve diğerleri, 2014). Mimari tasarımcılar daha dar bir çerçevedeki optimizasyon yöntemlerine yoğunlaştıkları görülür. Mimari tasarımda daha çok meta-sezgisel algoritmalar tercih edilir.

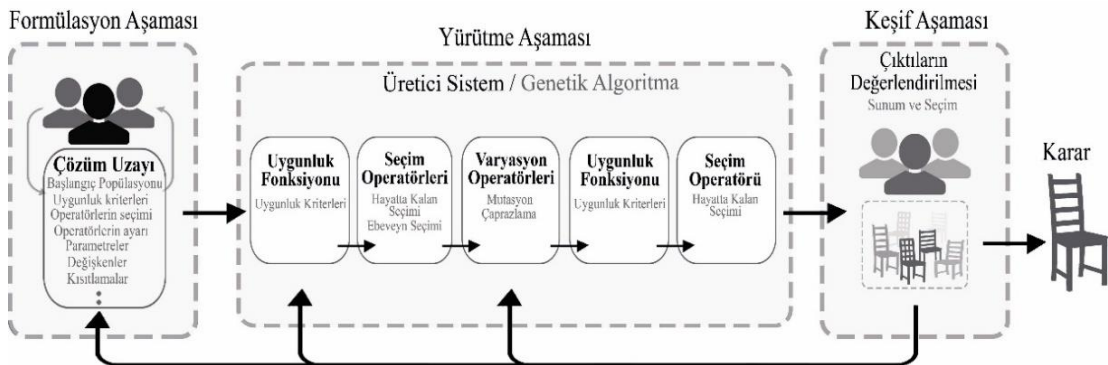
Mimari tasarımda yaygın olarak kullanılan metasezgisel yöntemlerden biri olan genetik algoritmalarıdır. Genetik algoritmalar, Darwin'in (1859) evrim teorisinden esinlenir. Genetik algoritmaları açıklayan kavramlar evrim teorisi ile ilişkilendirilir. Evrim sürecindeki bu kavramlar, optimizasyon problemlerini çözmeye bağlantılı olarak tasarım sürecindeki bazı kavramlara karşılık gelir. Çevre kavramı amaç fonksiyonu olarak, birey kavramı aday çözümler olarak, uygunluk kavramı nitelik olarak, gen kavramı değişken olarak, alel kavramı değer olarak ve popülasyon kavramı çözüm kümesi olarak tanımlanabilir (Eiben ve Smith, 2015: 14,17; Kramer, 2017). Darwin (1859), sunduğu evrim teorisinde biyolojik çeşitliliği esas alır ve doğal seçilimi vurgular. Doğal seçim, çevre koşullarına en iyi adapte olan bireyleri seçmeyi ifade eder. Bireyler, popülasyon içerisinde bulunur (Şekil 2.13.'de c). Bireyler çevreyle ne kadar uyumluysa diğer bir deyişle uygunluk değerini ne kadar karşılıyorsa o kadar üreme ve çoğalma şansını yakalar. Aksi halde bireyler yavru olmadan ölür. Dolayısıyla popülasyondaki bir dizi birey diğer bireylerle kıyaslanarak başarıları ölçülür.

Genetik algoritmalar, olası çözümleri bulmak için gerçek dünyanın bazı yönlerinin bilgisayar (üretici araç) tarafından manipüle edilebilecek şekilde soyutlanmalarını (temsilleri) kullanır. Bunun için orijinal problem bağlamından evrimin gerçekleştiği problem çözme alanına veri aktarılmalıdır. Diğer bir deyişle fenotip uzayı ile genotip uzayı arasında bağ kurulmalıdır. Genotipleri içeren genotip uzayı, arama uzayı olarak ve fenotipleri içeren fenotip uzayı çözüm uzayı olarak ifade edilebilir (Bentley, 1999: 8-12). Fenotip uzayı, olası çözümleri oluşturan nesnelere oluşur, dolayısıyla orijinal problem bağlamını temsil eder. Genotip uzayı, tüm evrimsel araştırmanın gerçekleştiği yerdir, dolayısıyla evrimin gerçekleştiği problem çözme alanıdır (Şekil 2.13.'de d). Her bireyin bir genotip temsili ve bir fenotip temsili vardır. Genotip gösterim, tasarımın bir modelini oluşturmak için kullanılacak bilgileri kodlayan genetik yapısını temsil ederken, fenotip gösterim, bir tasarımın gözlenebilir özelliklerini gösteren gerçek tasarım modelidir.



Şekil 2.13. A'da mutasyon öncesi ve sonrası, b'de çaprazlama sonucu oluşan yavrular, c'de biyoloji odaklı terminolojiye göre 0 ya da 1 sembolüne gen, genlerden oluşan diziyeye birey ve popülasyon, d'de kodlama ve kod çözme

Uygun olmayan herhangi bir seçim, algoritmanın yakınsamasını zorlaştırarak anlamsız sonuçlar üretilmemesi için formülasyon aşaması önem arz etmektedir. Amaç fonksiyonu, varyasyon ve seçim operatörleri, kısıtlamalar, popülasyonun büyüklüğü, çaprazlama ve mutasyon oranları gibi çeşitli parametrelerin seçimi ve formülasyonu çözümlerin ayrıntısını, çeşitliliğini, verimini ve sonuçlara ulaşılma süresini etkilemektedir (Grefenstette, 1986; Yang, 2021).



Şekil 2.14. Genetik algoritmanın tasarım süreci aşamaları: formülasyon, yürütme ve keşif aşamaları

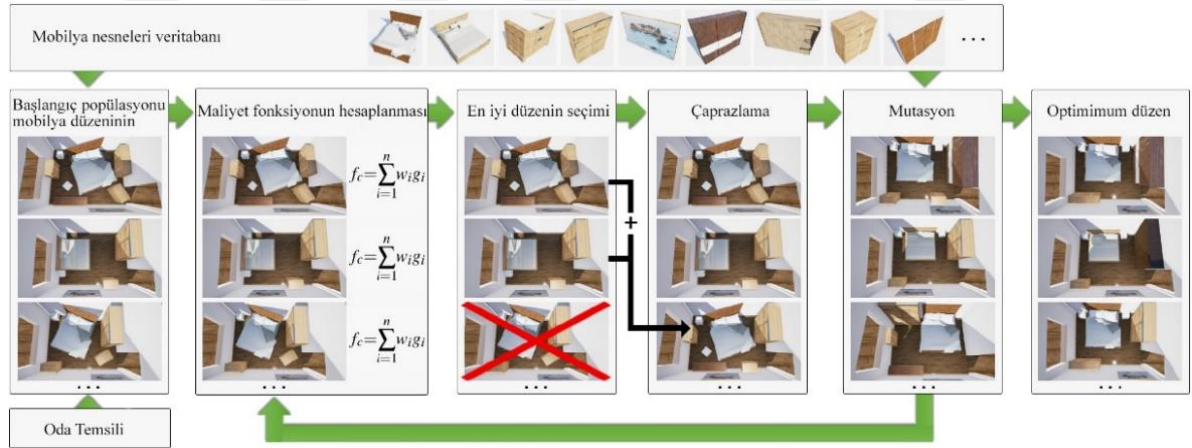
İlk olarak başlangıç popülasyonu rasgele bir parametre kümesi ile üretilir. Bir dizi çözüm seçilir (genellikle rastgele) ve bu, ilk popülasyonu oluşturur. Bireylerin her birinin uygunluğu hesaplanır. En uygun bireylerin daha yüksek seçim olasılığına sahip olduğu bir seçim gerçekleşir. Çözüm kümesinin alternatifinin artması için bir grup birey (ebeveyn) rastgele ancak uygunluk değeri ile orantılı bir olasılıkla seçilir. Varyasyon operatörleri olan çaprazlama ve mutasyon kullanılarak yeni alternatif çözümler (yavru) yaratılır. Mutasyon, bir çözüm adayına uygulanır ve değiştirilmiş yeni bir çözüm adayıyla sonuçlanır. Mutasyonda da hangi parçaların değiştirileceği ve bunların yerine hangi parçaların seçileceği rastlantısal olarak yapılır (Şekil 2.13.'de a). Çaprazlama operatörü, iki yavru oluşturmak için iki kromozomun alt dizilerini değiştirir (Şekil 2.13.'de B). Seçilmiş aday çözümlerin özelliklerini rastgele bir şekilde yeni bir çözümde birleştirir (Syswerda, 1993). Eski bireylerin bir kısmı, yenilerinin girmesine izin vermek için popülasyondan çıkarılabilir. Yeni bireylerin uygunlukları hesaplanır ve popülasyona (çözüm kümesine) dâhil edilir. Eğer yineleme eşiğine ya da yeterince yavruya ulaşılmazsa çaprazlama ve mutasyon işlemlerine geri dönülür ve aynı işlem sırası takip edilir, ancak yeterince yavruya ulaşıldığı halde çözüm onaylanmazsa ya da maksimum sayıda çözüme ulaşılmazsa çaprazlama ve mutasyondan bir önceki aşamadan yani yürütme aşamasının ilk adımından devam eder (Bentley, 1999). Popülasyondaki en iyi bireyler bir sonraki nesli üretmek için seçilir veya popülasyondaki en kötü bireyler yenisiyle değiştirilir. Böylece popülasyonun en uygun bireyine yaklaşılmaya çalışılır (Mitchell, 2009).

Sonlandırma koşulu, evrimsel döngünün ne zaman sona ereceğini ifade eder. Sonlandırma koşulu; uygunluk değerlendirme sayısı belirli bir sınıra ulaştığında, değerlendirme süresi ve maliyeti gibi kısıtlamalar tanımlandığında, uygunluk fonksiyonunun iyileştirmelerinin ilerlemesi önemli ölçüde azaldığı durumlarda veya popülasyon çeşitliliği belirli bir eşiğin altına düştüğünde sağlanabilir (Eiben ve Smith, 2015: 34; Kramer, 2017: 17). Sonlandırmadan sonra en iyi genotipin kodunun çözülmesiyle iyi bir fenotip çözümü elde edilebilir.

Yürütme işlemi tamamlandıktan sonra tasarımcı, birçok çözüm arasından seçimler yapabilir. Tatmin edici sonuçlar çıkmazsa döngü devam eder. Tekrardan

uygunluk değerlerine seçimler ve mutasyon ve çaprazlama ile alternatifler artırılabilir. Formülasyon aşamasına geri dönüp, parametreler, kısıtlamalar, değiştirilebilir. Uygun sonuç bulunduğu süreç tanımlanır.

Genetik algoritmalar, mekan tasarımında iç mekan tasarımında yardımcı olabilir. Kán ve Kaufmann (2017), iç mekân mobilyalarının düzenlerini ve uyumlu renk dağılımlarını optimize etmek için genetik algoritma kullanmıştır (Şekil 2.15.). Bu çalışmada genetik algoritmanın amacı, belirli bir oda için her bir iç tasarımı; ergonomi, estetik ve işlevsellik açısından değerlendiren bir maliyet fonksiyonunu minimuma indirmektir. Optimizasyonda maliyet fonksiyonunu oluşturmak için iç mekân tasarımında kullanılan yönergeleri temsil eden matematiksel ifadeler oluşturulmuştur. İç mekân düzeniyle ilgili yönergeler, mobilyaların açıklıkları, kendi aralarındaki mesafeleri, grup olarak ilişkileri, yönlendirmeleri, hizaları ve dağılımlarıyla ilişkilendirilmiştir. Minimize edilecek maliyet fonksiyonu daha sonra bu ifadelerin ağırlıklı toplamı olarak tanımlanmıştır. Ayrıca uyumlu renk konfigürasyonu ve tutarlı malzeme türleri elde etmek için malzeme ataması optimize edilmiştir.

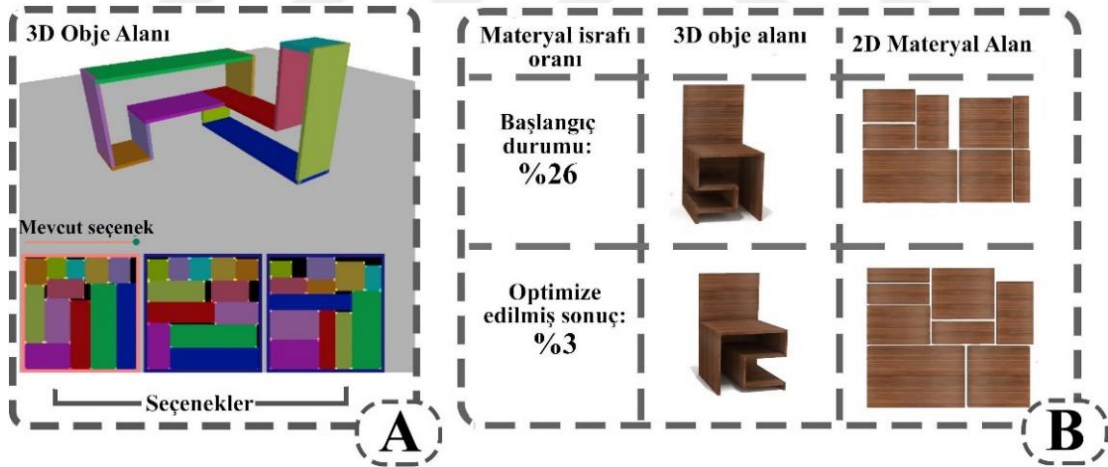


Şekil 2.15. Popülasyon başlatıldıktan sonra, maliyet fonksiyonunun değerlendirilmesi, en iyi bireylerin seçilmesi, çaprazlama ile yeni bireylerin oluşturulması ve bireylerin mutasyonlarla değiştirilmesi olarak devam eden yinelenmeli süreç

Kaynak: (Kán ve Kaufmann, 2017).

İç mimari tasarımda etkin malzeme ve enerji kullanımı için optimizasyon algoritmalarından yararlanılabilir. Koo, Hergel, Lefebvre ve Mitra (2017), atıkları en aza indiren ve daha etkin bir malzeme kullanımı ile sonuçlanan mobilyalar tasarlamak

için kullanıcı tarafından belirlenen kısıtlamalara göre malzeme kullanımını dinamik olarak analiz eden hesaplamalı bir sistem önermiştir. Bu sistemde tasarımcı, geleneksel tasarımda olduğu gibi tasarlanan tasarımı direk alıp onu üretmek yerine tasarımı, mevcut materyalden en iyi şekilde yararlanacağı biçimde oluşturmuştur. Lazer kesimde malzeme tasarrufu sağlayan parametrelerin belirlenmesi ve tasarımın formunun oluşturulması aynı anda etkileşimli bir şekilde gerçekleşmiştir. Kullanıcı belirlediği hacme uyacak şekilde formu oluştururken sistem ahşap parçalarını malzeme israfını önleyecek şekilde otomatik olarak iki boyutlu düzlemde ayarlamıştır (Şekil 2.16.'da b). Sistem, dinamik olarak iki boyutlu düzlemde malzeme alanı düzenlerini analiz etmiş ve belirtilen tasarım kısıtlamalarına göre üç boyutlu olarak tasarım alternatifleri önermiştir (Şekil 2.16'da a). Malzeme israfını azaltmak için optimize edilmiş alternatifler, algoritma tarafından izlenen bir tasarım keşif yolunu temsil etmiştir. Algoritma, mevcut iki boyutlu yerleşime göre hangi parçanın değiştirileceğini ve malzeme alanı yerleşimine göre parçaların nasıl uyarlanacağını belirlemiştir.

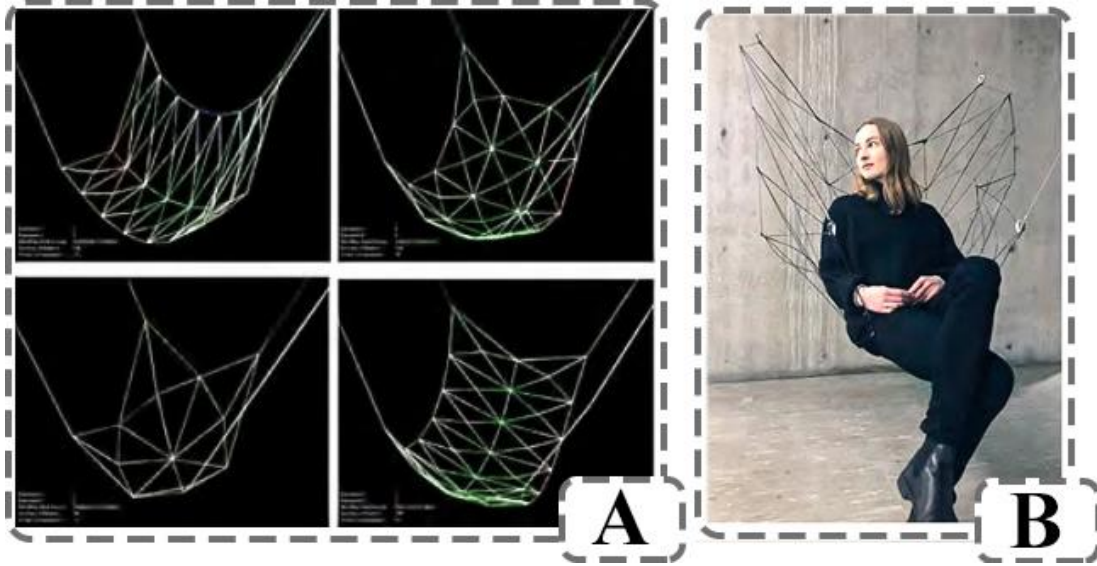


Şekil 2.16. A'da sistemin arayüzü, b'de sistem ile oluşturulmuş optimum tasarım sonucu.

Kaynak: (Koo ve diğerleri, 2017).

Mobilya tasarımı için geleneksel tasarım sürecinde ulaşılamayan yapısal performans bilgileri, optimizasyon yöntemleriyle sağlanabilir. Tasarım sürecinde malzeme bilgisi, optimum verim sağlayan mobilya tasarımları için önemlidir. Estrada, Wyller ve Dahy (2019), malzemenin mühendislik özelliklerinin hesaplamalı tasarımı

bilgilendiren ana rol olduđu bir alıřma sunmuřtur. Karbon fiber kompozitlerin ekme mukavemetinin baskın niteliđi, hafif bir asılı sandalye tasarımına ilham vermiřtir. Malzemenin baskın niteliđi maksimuma ıkarılmıř, malzeme kullanımını minimumda tutmak amacıyla entegre bir hesaplamalı tasarım gerekleřtirmiřtir. Gaudi'nin asılı modellerinden ilham alındıđı bu alıřmada, yerekimi tabanlı form bulmayı sađlamak iin Karamba kullanılmıř, yapısal optimizasyon iin ise Galapagos kullanılmıřtır. İnsan yk altında sandalyenin deformasyonu simle edilmiř ve simlasyon sonuları, sıkıřtırma altındaki gelerin yzdesinin daha az olduđu ađ konfigrasyonunu bulmak iin kullanılmıřtır (řekil 2.17.).



řekil 2.17. A'da alternatif ıktılar, b'de sonu

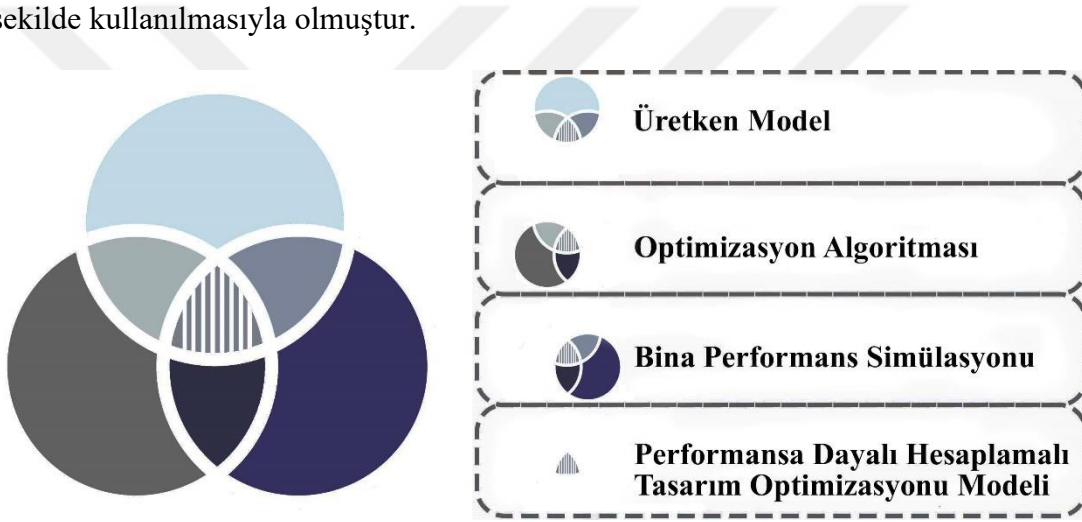
Kaynak: (Estrada ve diđerleri, 2019).

2.4.5.2. Optimizasyona Dayalı Yaklařımın Hesaplamalı Tasarım ve Diđer Hesaplamalı Tasarım Yaklařımları ile İliřkisi

Mimari tasarımda optimizasyon rutini, üretken sistemlerden (parametrik, algoritmik, kural tabanlı, ajan tabanlı, veriye dayalı vb.) iyileřtirilmiř varyasyonlar üretilmesini sađlar. Bir bina performans optimize etmenin en az dört adımı sunulmuřtur (Machairas ve diđerleri, 2014: 107) tasarım deđiřkenlerinin ve ilgili kısıtlamalarının tanımlanması, bir bina performans simlasyon aracının seimi ve bir bina modelinin oluřturulması, uygun bir ama fonksiyonunun seimi, uygun bir

optimizasyon algoritmasının seçimi. Öncelikle sorunun formüle edilmesi ile süreç başlar. Tasarımcı çözümlerin performansının ve sonuçların kalitesinin önemli ölçüde artırılabilmesi için sorunu formüle eder.

Mimari tasarımların optimizasyonu, sürdürülebilir bina tasarımları amaçlayan performans odaklı mimari tasarım açısından özel bir öneme sahiptir. “Performansa dayalı hesaplamalı tasarım optimizasyonu”, hesaplamalı tasarım araç/yöntemleri, performans simülasyonları ve optimizasyon teknikleri birleşiminden doğmuştur (Touloupaki ve Theodosiou, 2017). Bu genellikle parametrik/algorithmik tasarım aracı, bina performans simülasyonları ve evrimsel optimizasyon tekniğinin entegre bir şekilde kullanılmasıyla olmuştur.



Şekil 2.18. Üretken modeller, optimizasyon algoritmaları ve bina performans simülasyonlarının keşim kümesi olarak performansa dayalı hesaplamalı tasarım optimizasyonu modeli

Kaynak: (Touloupaki ve Theodosiou, 2017).

Parametrik araçlarda performans optimizasyonu çalışmaları, binanın ve mekânın performansını artırır. Rhino/Grasshopper’da parametreleştirilmiş başlangıç tasarımlarını oluşturur. Grasshopper, kullanıcıların çeşitli tasarım parametreleri arasında mantıksal ilişkiler tanımlamasına olanak tanır. Geometrik modeli tanımlayan geometrik veriler, gerekli olan malzeme özelliklerini belirleyen fiziksel veriler, çevresel parametreler ve optimizasyon sürecinin kontrolünü sağlayan algoritma verileri seçilir. Bu girdi verileri performans simülasyon programına taşınabilir. Alınan verilere dayalı olarak simülasyon programında performans analizi yapılabilir.

Simülasyon sonuçları optimum hedefe ulaşana kadar bu işlem döngüsü devam eder. Bu işlem tasarımcı tarafından manuel olarak yapılabileceği gibi bir optimizasyon algoritması tarafından otomatik olarak da yönlendirilebilir. Grasshopper ile entegre olarak çalışan Galapagos gibi eklentiler, optimizasyon görevleri ve üretken tasarım için giriş parametrelerini değiştirerek uygun sonuçların bulunmasına yardımcı olur.

2.4.6. Doğadan İlham Alan Yaklaşımlar

Doğa, minimum kaynakları kullanarak maksimum performans veren etkili ve verimli malzeme ve formlar sunmasının yanında bunları tasarlamak için yöntemler, araçlar, mekanizmalar ve sistemlerin yaratılmasına rehberlik eder. Mimari tasarımcılar en başından beri doğada var olan geometrik formlardan ilham almış ve doğal sistemleri algoritmik olarak modellemişlerdir (Rocker, 2006). İlk örnekler, biyolojik canlıların kendini yenileme yeteneklerini ilham alan hücresel otomat, bitkilerin dallanmasını ve gelişim sürecini modellemek amacıyla geliştirilen l-sistemleri ve yinelemeli örüntüler oluşturan fraktallardır (Bölüm 2.4.3.). Diğer yandan günümüzde mimari tasarımda yaygın olarak kullanılan biyoloji tabanlı optimizasyon yöntemleri; sinir sisteminden ilham alan yapay sinir ağları (Bölüm 2.4.7.), doğal seçim ilkesinden ilham alan evrimsel algoritmalar (Bölüm 2.4.5.) veya sürü davranışlarından ilham alan sürü zekâsı tabanlı algoritmalarıdır. Diğer bölümlerde daha detaylı olarak işlenen bu modellerin dışında, literatürde biyomimetik tasarım, morfojenetik tasarım, malzemeye dayalı tasarım gibi doğadan ilham alan ve öğrenen tasarım yaklaşımlar kullanılmıştır.

“Biyomimesis (Biomimicry, bios-hayat ve mimesis-taklit etmek)”, kavramı bir bilim dalı olarak ilk kez 1997 yılında tanımlanmıştır (Benyus, 1997). Benyus, mimari tasarım problemlerini de kapsayan birçok problemin doğa modellerinden ilham alınarak çözülebileceğini vurgular. Doğa gözlenir ve doğanın zekâsından dersler çıkarılır. Doğal ilkelerin değerlendirilmesi, bunların hesaplama mantığının anlaşılması doğanın zekâsının mimari tasarımlar için bir problem çözme kaynağı olarak kullanılmasını sağlar. Bu anlamda biyomimetik mimarinin amacı sadece mekânı şekillendirmek ve ölçmek değildir. Biyomimetik, daha çok doğadan işlevsel bir ders alma ile ilgilidir. Biyomimetik tasarımda, doğanın biçimsel özelliklerinin ötesinde

daha derin, daha duyarlı bir doğa anlayışı geliştirme amacı vardır (El-Zeiny, 2012). Doğadaki organizma, cansız varlık ya da bir olgudan dersler çıkarılabilir.

Mimari tasarımda kullanılan ve doğanın ilkelerinden öğrenen bir diğer yaklaşım, doğal morfogenezdır. Bu yaklaşıma göre mimari formlar ile organizmalar arasında benzeşim kurulabilir. Bu anlamda organizmaların çevreye göre değişen ve adapte olarak gelişen dinamik süreci mimari form yaratımı için ilham kaynağı olur (Leach, 2009; Menges, 2007; Roudavski, 2009). Morfogenezde form, malzeme ve işlev birbirinden ayrı düşünülemez (Menges, 2007). Malzeme, formun özünü oluşturur ve işlevini belirler (N. Oxman, 2010). Doğadan ilham alan form tasarımı süreçlerinde malzeme özellikleri, malzemenin davranışı ve malzemenin çevreleri arasındaki etkileşimi önem kazanır. Doğal yapıların morfolojik yapısı ve malzeme sistemleri, bir tasarımın hesaplamalı olarak türetilmesine ve detaylandırılmasına izin verir (Menges, 2012).

Biyomimetik doğayı daha çok bir “inovasyon” kaynağı olarak ele alırken, morfogenez, “doğadaki madde ve enerji döngülerinin doğal biçimleri oluşturma süreçlerine” odaklanır (Beyaztaş, 2012). Diğer yandan doğadaki morfogenez süreci ile derinden ilişkili olan evrimsel algoritmalar bir taklitten de öte biyolojideki üretken sistemlerin prensiplerinin mimariye aktarılmasıdır.

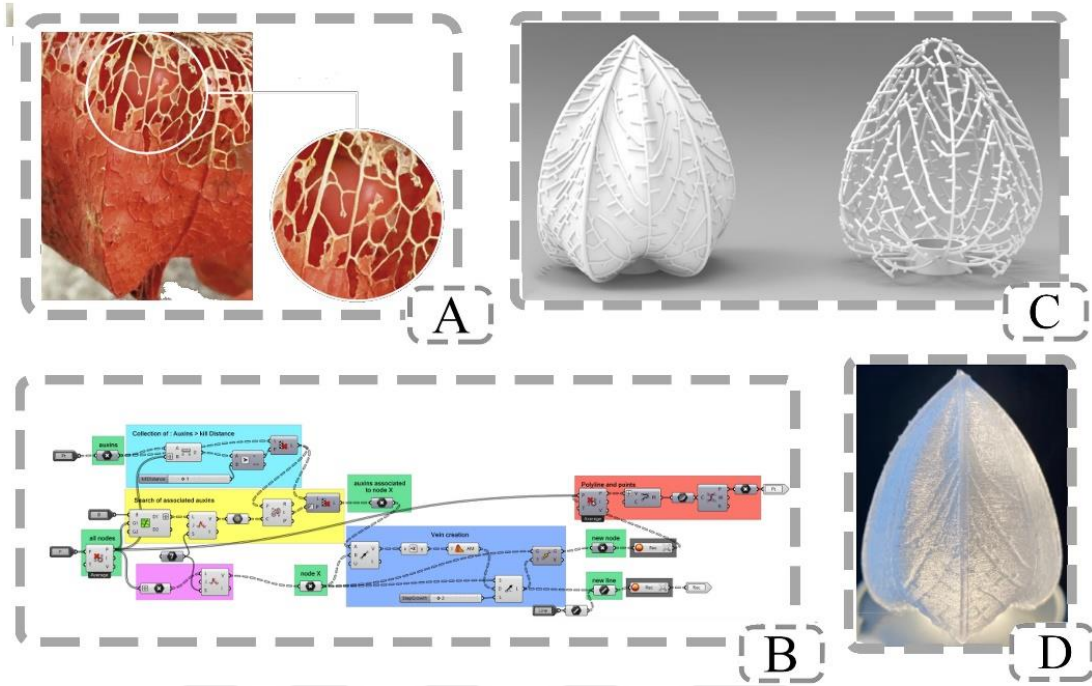
2.4.6.1. Doğadan İlham Alan Modeller ve Yöntemler

Doğa, tarihsel dönemlerden beri tasarım sürecinde referans olarak kullanılmıştır (Özülkü, 2010). Doğadaki biçim, oluşumlar ve olgular araştırılmış ve doğadaki bu tasarım mantığı mimari tasarım amaçları için kullanılmıştır. Biyomimetik ilkeler hesaplamalı modeller ile birleştirilirken, doğal morfogenezde hesaplamalı üretken modeller için rehber olmuştur.

Biyomimetik yaklaşım çerçevesinde doğal sistemlerin araştırılması ve soyutlanmasıyla, hesaplamalı mimari tasarım modelleri oluşturulabilir (Ahmar, Fioravanti ve Hanafi, 2013; Sorguç ve Selçuk, 2013b; Yazıcı, 2020). Hesaplama kullanılarak doğal ilkelerin mantığının mimari tasarım sürecine nasıl dönüştürüldüğü önemli bir konudur. Doğal sistemlerin karmaşık doğası araştırılarak, sistemin

parametrelerinin hesaplama modelini çalıştırabileceği şekilde soyutlanması gerekir. Bu çerçevede doğal sistemlerin parametreleri, kuralları ve matematiksel özellikleri anlaşılır ve soyutlanarak yeniden oluşturulur. Doğal sistemi ya da modeli modelleme sürecinde, modelin etki alanını tanımlamak için seçilen parametreler referans sistemini, yani hesaplamalı modeli oluşturur. Doğal sistemin parametreleri, bu sistemin kurallarıyla birlikte referans modele aktarılır. Sistemin kuralları, referans modelde algoritmalara yansır.

Dolayısıyla hesaplamalı tasarımda doğadan ilham alan yaklaşımlar, biyoloji, teknoloji ve mimari tasarım üçgeninde, yenilikçi hesaplamalı yaklaşımlar üretmiştir. Karmaşık problemlere çözümler sunmak amacıyla doğadan ilham alan hesaplamalı teknikler, doğadaki ve biyolojideki belirli özellikleri ve mekanizmaları taklit etmek için tasarlanmıştır. Bu mantığı kullanan yaklaşımlardan biri olan biyomimikri, daha çok fonksiyonel problemleri çözmeyi amaçlayan bir strateji olarak bilinir. Ancak, doğanın karmaşıklığı, sembolik anlamı açısından da taklit edilebilir. Bu duruma örnek olarak Graziosi ve diğerleri (2021), “Çin Feneri” olarak bilinen *Physalis Alkekengi* bitkisinin yapısından ve şeklinden (Şekil 2.19.’da a) esinlenerek bir aydınlatma tasarımı yapmıştır. Yaprakların iki boyutlu gelişimini ve damarlanma modelini resmileştirmek için literatürde açıklanan hesaplamalı bir modelden yola çıkmış ve bir lambanın geometrisini sunmak için Grasshopper’da bir algoritma yaratmıştır (Şekil 2.19.’da b). Biyo-ilhamlı fonksiyonel, estetik ve sembolik ürün tasarlamak için parametrik/algoritmik modelden yararlanmıştır. Algoritmik model, çözümün fizibilitesini garanti etmiş, karmaşıklığı kontrol etmiş ve parametre değişiklikleri ile benzersiz birçok ürünün oluşturulmasına imkân vermiştir.



Şekil 2.19. A'da esinlenilen bitkinin yapısı, b'de modelin tasarımında kullanılan prosedür örneği, c'de üç boyutlu model örneği, d'de üç boyutlu yazıcı kullanılarak basılan model

Kaynak: (Graziosi ve diğerleri, 2021).

Biomimetik yaklaşıma dayanan hesaplamalı modellerle birlikte doğal morfogenez kavramıyla ilişkilendirilen hesaplamalı modeller de literatürde çokça kullanılmıştır. Daha önce bahsedilen form bulmaya yönelik kullanılabilen üretken sistemler veya algoritmalar, doğal morfogenez süreci ile ilişkilendirilmiştir.

Morfogenez terimi, performansa dayalı süreçlerin dijital form bulma modelleriyle entegrasyonu ile de öne çıkarılabilir (Oxman, 2009: 1030, 1031). Adriaenssens, Block, Veenendaal ve Williams'a (2014: 3) göre optimizasyon sürecinde yaratılan şekiller kümesinden bir keşif yapmaya çalışırken giderek form geliştirmeye meyilli olduğumuzda hesaplamalı morfogenez terimiyle karşılaşılabılır. Bu anlamda morfogenez, formun üretilmesinde kuvvetlerin rolüyle ilgilenen hem doğal hem de yapay tüm ilgili vakaların etrafında odaklandığı merkezi kavramlar bütünü olarak görülür (Oxman, 2009, 2011). Dijital morfogenez, hesaplama yoluyla doğadaki karmaşıklığın gelişimini simüle eder. Doğanın ilkelerini şekil tanımlama

algoritmalarına çevirerek, doğadaki karmaşıklığı açıklar ve ortaya çıkarır. Böylece, doğal morfojenetik süreçlerde farklı üretken sistemlerin işlevi belirginleşir.

Klemmt ve Sugihara (2020), hüresel büyüme simülasyonlarını, geometrik oluşumu kontrol edebilen biyo-ilhamlı bir form oluşturma mekanizması olarak kullanmayı amaçlamıştır. Tasarladıkları algoritma sayesinde, hücreler çeşitli kuvvetlere tepki vererek, belli kurallara göre bölünerek ve belli konumlara eklenerek geometrik birikimler oluşturmuştur (Şekil 2.20.'de a). Bu birikimler lobi tavanında ve resepsiyon masasında dekorasyon ögesi olarak kullanılmıştır (Şekil 2.20'de b). Şekil 2.28.'deki A'da hücre büyüme simülasyonu ile oluşturulabilecek çeşitli topolojiler, B'de lobinin tavanını kaplayan ve resepsiyon masasının önünü ve üstünü oluşturan hücre büyüme simülasyonu ile geliştirilmiş şekiller görülmektedir (Klemmt ve Sugihara, 2020). Çalışmada karmaşıklık düzeyleri nedeniyle daha önce sorgulanamayan doğal fenomenlerin ve biyolojik sistemlerin ilkeleri, dijital süreçlerle yeniden üretilerek çıkarılmıştır. Doğal kalıplar yeniden yorumlanarak iç mekan tasarımında farklı bir bağlama göre yeniden uyarlanabilmektedir.



Şekil 2.20. Hüresel büyüme simülasyonunun gösterimi

Kaynak: (Klemmt ve Sugihara, 2020).

2.4.6.2. Doğadan İlham Alan Yaklaşımların Hesaplamalı Tasarım ve Diğer Hesaplamalı Tasarım Yaklaşımları ile İlişkisi

Form, malzeme ve yapının entegrasyonunu destekleyen bir dizi hesaplama stratejisi geliştirilmiş ve önerilmiştir (N. Oxman, 2010). Doğal morfogenezin ya da

biyomimetik yaklaşımının temel ilkelerinden ilham alan hesaplamalı morfogenetik tasarımlar ya da hesaplamalı biyomimetik tasarımlar yaratılabilir. Şekilden ziyade malzemenin davranışına odaklanan malzeme tabanlı hesaplamalı modeller kullanılabilir (N. Oxman, 2010; Yazıcı, 2013).

Mimaride özellikle morfogenez, dijital üretken yöntemleri içeren kapsayıcı bir anlayışa bürünerek hesaplamalı tasarım süreçleriyle ilişkilendirilmiştir (Kolarevic, 2005c). Mimarideki dijital morfogenez, doğadaki morfogenez süreçleriyle büyük ölçüde benzer bir ilişki kursa da gerçek büyüme veya adaptasyon mekanizmalarını benimsemek zorunda olmadığı bir anlayış içerisinde değerlendirilmiştir (Roudavski, 2009). Bu üretken modeller biçimsel karmaşıklığı, belirsizliği ve sürekli olarak kendini yeniden oluşturuşu ile tasarımcıları dijital morfogenetik potansiyelin bir “düzenleyicisi” haline getirmiştir (Kolarevic, 2005c). Bu anlamda özellikle morfogenez sürecin altında yatan ilkeler, hesaplamalı tasarımın anlaşılması için kavramsal bir çerçeve sağlamıştır (M. Hensel, Menges ve Weinstock, 2006; Menges, 2012).

Hesaplamalı tasarımda doğadan ilham alan yaklaşımlar ile diğer hesaplamalı yaklaşımlar kesişir. Doğa, üretkendir, performatiftir, optimumu arar ve algoritmik olarak soyutlanabilir. Doğa, genellikle optimizasyon mantığıyla çalışır (Leach, 2009). Bu, Gaudi ve Otto gibi mimari tasarımcıların doğadaki yapısal organizasyonlardan ilham alan form bulma deneylerinde görülmektedir. Doğayı gözlemleyen ve yorumlayan Gaudi ve Otto gibi öncüler, hesaplamalı fiziksel modelleri doğal sistemlerin davranışını anlamak için kullanmıştır. Evrimsel gelişim döngüsü boyunca optimize edilen doğal yapılar, geçmişte olduğu gibi günümüzde de mimari tasarımcılar için büyük bir bilgi kaynağı oluşturur (Symeonidou, 2019). Ancak günümüzde, doğanın salt taklidi olarak basitleştirilmiş modeller yerine üretici dinamikleri oluşturan büyüme, dönüşüm ve adaptasyon süreçleri kullanılır (Rossi, 2019). Bu anlamda doğadaki karmaşıklığın gelişimini simüle etmeyi ifade eden dijital morfogenez süreçleri, çeşitli algoritmalar ve üretici sistemlerle ilişkilendirilir.

Doğa kanıtlanabilir bir şekilde sürdürülebilirdir. Bu anlamda doğanın ilkelerinin öğrenilmesi, minimum kaynakların kullanılarak maksimum performansa

sahip kalıcı çözümlerin bulunmasını sağlar. Janine Benyus (1997), daha sürdürülebilir bir gelecek sağlamak için doğayı taklit etme ihtiyacını biyomimetik kavramı ile tartışmıştır. Dolayısıyla hesaplamalı tasarım sürecinde biyomimetik ilkelerin kullanılması, yüksek etkinlik ve performansa sahip sistemler sunar. Doğada her zaman var olan işlevsel entegrasyon, performans kapasitesi ile sonuçlanır (Menges, 2012). Doğadaki canlı ya da cansız varlıklar doğanın ilkelerine uyum sağlarken biçim kazanırlar. Bu doğal biçimler analiz edilip hesaplamalı teknolojiler ile yeniden üretilmiştir. Bu anlamda çeşitli performans simülasyonları da doğanın bir taklididir. Doğa olaylarının ve davranışlarının taklidi, doğadan öğrenbilme potansiyelini artırır.

2.4.7. Veriden Öğrenmeye (Deneyim) Dayalı Yaklaşım

Dijital gelişim ve dönüşüm, günümüzdeki veri bolluğunu verimli bir şekilde değerlendirmeyi cezbettirmiştir. Bu dönüşüm, verilerden önemli kalıpları ve eğilimleri ortaya çıkarmayı ya da diğer bir deyişle “verilerin ne söylediğine kulak verip verilerden öğrenmeyi” gerekli kılmıştır (Hastie, Tibshirani ve Friedman, 2009).

Verilerden öğrenim, istatistik ile mümkün olabilmektedir. İstatistik, veriden öğrenmeye ve deneyime dayalı yaklaşımda örnek veri üzerinden genellenebilir yapılar ortaya çıkarmak için matematiksel çerçeveyi sağlarken; bilgisayar bilimi, bu öğrenme yaklaşımlarının optimizasyonu ve performansı ile ilgilenir (Alpaydın, 2020). Makine öğrenimi, bu ortak amaçlar ile kesişir ve bilgisayarların mevcut verilerdeki istatistiksel olarak anlamlı kalıpların sistematik keşfi yoluyla veriye dayalı model oluşturmasıyla ilgilidir (Alpaydın, 2020). Makine öğrenimi, açıkça programlamaya başvurmak yerine belirli bir görevi gerçekleştirmek için verilere göre tahminler yapan algoritmaları kapsar (Samuel, 1959).

Veri öğrenmeye dayalı yaklaşımda, tasarımcının denetimiyle ya da denetimi olmadan otomatik olarak verileri öğrenen ve elde ettiği deneyime göre gelişen bilgisayar programları oluşturulur (Mitchell, 1997: xv). Makine öğrenimi, verilerden örüntüler çıkarabilen ve daha sonra bu örüntülerden gelecekteki verileri tahmin eden, orijinal verilere benzer bir davranışı taklit eden veya belirsizlik altında diğer karar verme türlerini gerçekleştirmek için kullanan bir dizi algoritmayı ifade eder (Murphy,

2012). Makine öğrenimi algoritmaları kendisine verilen verileri ayrıştırır, verilerden öğrenir, öğrendikçe daha iyi hâle gelir ve öğrendiklerini benzer durumlar için kullanır. Bunu, çözümleri analiz ederek ve mevcut verilerdeki istatistiksel olarak anlamlı ortak kalıpları öğrenerek yapar.

Deneyim, toplanan verilerin analizi ile olur (Mohri, Rostamizadeh ve Talwalkar, 2018). Veriler, alanında uzman biri tarafından dijital olarak etiketlenen veya çevre ile etkileşim yoluyla elde edilen diğer bilgi türleri şeklinde olabilir. Verilerden deneyim kazandırılan bilgisayarlar, mimari tasarım problemlerinin çözümü için tahminler yapmak ve gelecekte olabilecek senaryoları tahmin etmek için kullanılabilir (Kvochick, 2020). Makine öğrenimi modelleri, mimari tasarım çözümleri yaratmak için örnek verileri, geçmiş deneyimleri kullanarak belirli kalıpları veya düzenlilikleri tespit eder. Süreci anlamak veya bu kalıpları tahmin etmek için olası tüm durumların ya da sürecin tamamının tanımlanması gerekmez. Öğrenme modeli, sunulan veriler ve parametrelerden öğrenerek (denetimli öğrenme) veya kendi kendini eğiterek (denetimsiz öğrenme, derin öğrenme) çözüm tespitleri yapar.

2.4.7.1. Makine Öğrenimi Modelleri ve Yöntemleri

Makine öğrenmesinin gerçekleştirilebilmesi için üç ana bileşen belirlenebilir: veri setleri (dataset), özellikler (features) ve algoritmalar (Şapcı ve Pektaş, 2021). Bu ana bileşenlerle birlikte tipik bir makine öğrenimi süreci birkaç adımdan oluşur: Veriler toplanır, veriler hazırlanır, model seçilir, verilerle model eğitilir, değerlendirmeler yapılır ve model optimize edilir (Raschka ve Mirjalili, 2017: 11-13; Lantz, 2019:18).

Veri setleri, geçmiş deneyimlerin örneklerini içerir ve bu örnekler tahminler veya kararlar vermek için kullanılır. Veri setleri sayılardan, resimlerden, metinlerden ve diğer birçok veriden oluşabilir. Bu verilerin niteliğinin ve niceliğinin artması sonuçların doğruluğunu ve güvenilirliğini artıracak için toplanan veriler, yeterli kalitede, yeterli miktarda ve çalıştırılabilir şekilde işlenmiş olmalıdır. Algoritmayı doğru şekilde eğitmek için yeterli miktarda verinin ne kadar olduğu hakkında ideal bir tarif olmasa da ihtiyaç duyulan veri genellikle kabaca tahmin edilebilir (Sydorenko,

2021). Özellikle derin öğrenme sistemleri gibi yüksek derecede işlemci gerektiren sistemler, standart makine öğrenimi sistemlerine göre daha fazla veri gerektireceği görülür. Bu veri setleri, üç türe ayrılabilir (Raschka ve Mirjalili, 2017: 190,191): eğitim veri seti, doğrulama veri seti ve test veri seti. Modeli eğitmek için eğitim veri seti, modelin eğitiminin ne kadar iyi gittiğini test etmek için test veri seti ve modelin performansını tekrar tekrar değerlendirmek için bir doğrulama seti kullanılır (Raschka ve Mirjalili, 2017: 190,191). Model veri setleri ile eğitildikten sonra, modelin iyi olup olmadığı görülmeye çalışılır. Eğitilmiş modelinin performansını değerlendirmek için veriler test edilir. Bunun için daha önce ayrılan veri seti kullanılır. Eğitilen makine, modelin bilmediği girdileri içeren değerlendirme veri setine göre test edilir ve halihazırdaki eğitilmiş modelin doğruluğu ölçülür.

Özellikler, makine öğreniminde yapılacak işlemin hangi kriterlere ve etkenlere göre yapılacağını belirleyen, sayısal veya sembolik olarak temsillerdir (Nilsson, 2005). Özellikler, mimari tasarımcının nesne hakkında ne öğrenmeye veya temsil etmeye çalıştığına bağlı olarak değişebilir. Mimari projenin genel görünümü, malzemelerin boyutları, sayısal özellikleri makine öğrenimi yöntemlerinde kullanılmak amacıyla bilgi hâline getirilebilir (Kvochick, 2020). Tasarımcı, mevcut özellikler kümesindeki hedef değişkenden kritik veya etkili değişkeni belirlenmesi gerekir. Ham veriden anlamlı özellikler çıkarmak için mimari tasarım problemine özgü yönergelerden yararlanılabilir. Bu yönergeler bağlama özgü tasarım kriterlerini içerir.

Eğitim seti örneklerinden matematiksel bir model oluşturan bir algoritma kullanılır. Farklı problem görevlerini çözmek için birçok farklı makine öğrenme algoritması geliştirilmiştir. Belirli bir problem ifadesinin gereksinimlerine göre en uygun algoritma seçilir. Genel olarak, makine öğrenimi algoritmalarının öğrenme süreci, algoritmaları beslemek için kullanılan verilere bağlı olarak denetimli, denetimsiz veya pekiştirmeli öğrenme yaklaşımı ile gerçekleştirilir (Maini ve Sabri, 2017). Veri türünün etiketli veya etiketsiz olmasına bağlı olarak modelin eğitimi, denetimli ve denetimsiz öğrenme olarak değişir. Etiketleme, gelecekteki veri işleme için denetimli öğrenme modeline bir öğrenme temeli sağladığı için denetimli öğrenmede bir dizi girdi ölçüsüne dayalı olarak bir sonuç ölçüsünün değeri tahmin

edilebilir (Burkov, 2019). Denetimsiz öğrenmede, sonuç ölçütü yoktur ve amaç, bir dizi girdi ölçütü arasındaki ilişkileri ve kalıpları tanımlamaktır (Hastie ve diğerleri, 2009).

Denetimli öğrenme modelleri, kullanıcının modele müdahalesinin en fazla olduğu ve bu yüzden adının “denetimli” olduğu makine öğrenimi modelleridir (Grossfeld ve Grieve, 2021; Middleton, 2021). Denetimli öğrenme modeliyle çalışırken modelin doğru kararı ondan öğrenebilmesi için önceden etiketlenmiş verilere ihtiyaç duyulur. Denetimli öğrenme modellerinde algoritma, kullanıcı tarafından etiketlenmiş bir veri kümesi üzerinden eğitilir. Dolayısıyla model eğitilirken doğru cevap önceden bilinir. Uzman kişi, bilgisayarı verilerle besleyerek ve doğru cevapları gösterir. Bir bilgisayara eğitim verileri ve nasıl yanıt verileceğini öğretmek için bir model verilir. Yeterli eğitimden sonra, algoritma daha önce görülmemiş girdi verilerine karşılık gelen doğru sonucu tahmin edebilir.

Genel olarak denetimli öğrenme; sınıflandırma ve regresyon olmak üzere iki önemli alt alandan oluşmuştur. Sınıflandırma modelleri nesnelere bilinen sınıflara ayırmaya izin verirken, regresyon analizi, hedef değişkenlerin sürekli sonuçlarını tahmin etmek içindir (Raschka ve Mirjalili, 2017: 3,4). Sınıflandırma, sınırlı sayıda seçenek içinde bir nesnenin türünü veya sınıfını tahmin etme görevidir. Sınıflandırma işlemi tüm nesnelere tanıtılması ve kategorize edilmesi ile başlar, program tarafından öğrenilen işlev, istenilen doğruluk seviyesine ulaşıncaya kadar kontrol ve düzenleme ile devam eder (Şapcı ve Pektaş, 2021: 74). Böylece bilgisayara sınıflandırma yapma öğretilmiş olur. Örneğin bir iç mekân mobilyalarının stillerini denetimli öğrenme algoritmasına tanıtmak için etiketlendiğinde, algoritma bir müddetten sonra bu mobilyaların stillerinin görüntüleri arasındaki ilişkiyi öğrenebilir ve daha sonra etiketleri olmayan yeni görüntüleri kendisi sınıflandırabilir.

Denetimsiz öğrenmede etiketlenmemiş veriler kullanılır. Bu modeldeki verilerde aranılan şey bilinmese de bilgi kümelerinden çıkarımlar yapılabilir (Hastie ve diğerleri, 2009). Denetimsiz öğrenmede, bilgisayarın uygun gördüğü şekilde modelin kalıplarını tanımlamasına izin verilir. Diğer bir deyişle algoritmanın kendi kendine öğrenmesi sağlanır. Sistem, bir öğretmen olmadan verilerdeki gizli bilgileri

bulmaya çalışır. Denetimsiz öğrenmenin yaygın bir kullanımı kümelemedir. Kümeleme, manuel müdahale olmadan bilgisayarın verileri ortak temalar ya da benzer nesnelere halinde gruplandırıldığı ve tanımlanan katmanlar halinde düzenlediği görevdir (Alpaydın, 2020).

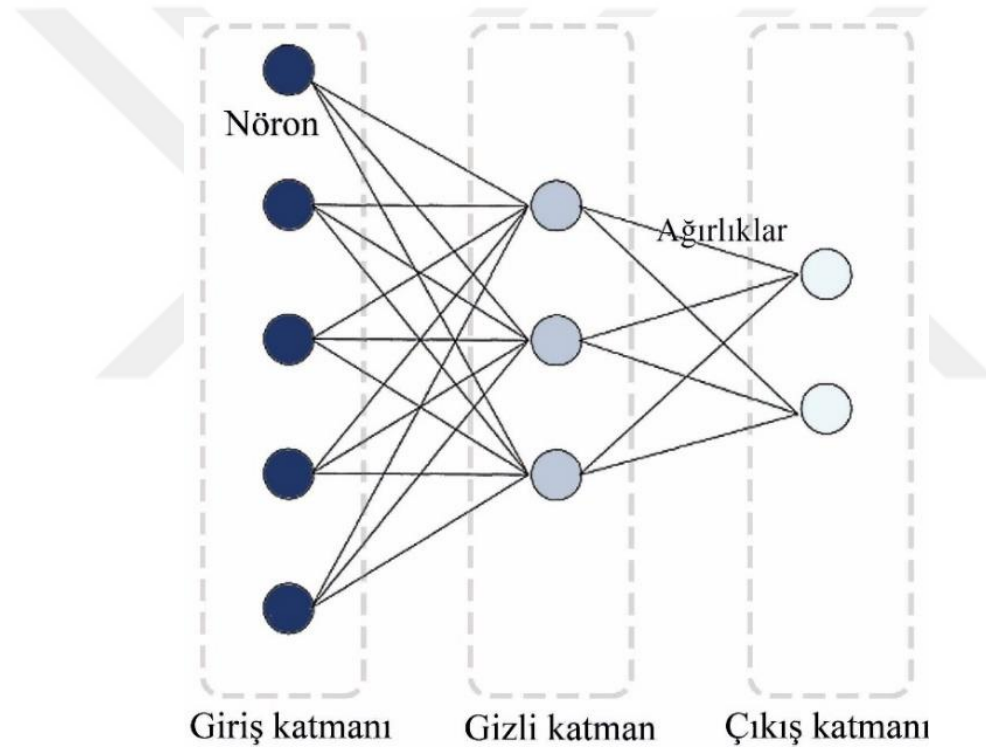
Pekiştirmeli öğrenme ise bir modelin kendi eylemlerinden geri bildirimini alarak öğrenmesini sağlayan bir deneme yanılma yaklaşımıdır (Maini ve Sabri, 2017: 81-83). Bu yaklaşımda bilgisayarlar doğru kararlar aldığında ödüllendirilir ve başarısız kararlar aldığında cezalandırılır. Böylece onun kalıpları öğrenmesine ve bilinmeyen veriler üzerinde daha doğru kararlar vermesine olanak tanınır.

Makine öğreniminin sonraki evrimlerini sinir ağları ve derin öğrenme modelleri oluşturur (Burkov, 2019). Bir yapay sinir ağı, işleme elemanlarından (nöronlar) ve bunlar arasındaki bağlantıların yanı sıra eğitim ve geri çağırma algoritmalarından oluşan, insan beyninde yer alan ve karmaşık öğrenme ve yorumlama yeteneğine sahip biyolojik sinir hücresinden ilham alan bir makine öğrenimi paradigmasıdır (Kasabov, 1996). Yapay sinir ağları ve derin öğrenme, standart makine öğrenmesi algoritmalarındaki tek bir matematiksel model yerine sinirsel ağ yapılarını taklit eden diyagramlardaki hesaplamalarla çalışır. Yapay sinir ağları, insan beyninde bulunan birbirine bağlı bu nöron ağlarını taklit eder. Böylece yapay sinir ağı algoritmaları; basit bir girdiden basit bir çıktıyı hesaplayabilir, öğrenebilir, hafızada tutabilir ve veriler arasındaki ilişkileri kurabilir (Kasabov, 1996).

Yapay sinir ağları katman ya da katmanlardan oluşur. Her bir katmanda nöron(lar) bulunur ve bu nöronlar kendinden bir önceki ve bir sonraki katmandaki nöronlarla bağlantılıdır (Şekil 2.21.). Bu katmanlar verinin sisteme girdiği yer olan giriş katmanı, bilginin işlendiği yer olan gizli katman ve sistemin veriler üzerinde ne yapacağına karar verdiği yer olan çıkış katmanıdır. Yapay sinir ağı katmanlarında, bir sonraki katmanın birimlerine bağlı, nöron adı verilen bir dizi küçük birim (matematiksel fonksiyon) bulunur. Girdiler (veriler), giriş katmanındaki nöronlara gelir. Nöronlardaki veriler, dönüştürülerek bir sonraki katmana aktarılır. Bağlantılar bir katmandaki bir nöronu aynı katmandaki veya başka bir katmandaki başka bir nörona bağlar. Nöronları birbirine bağlayan her bağlantı, ağırlık adı verilen bir sayı ile

ilişkilendirilir (Şekil 2.21.). Ağırlıklar, girdiler ve çıktılar arasındaki ilişkileri ayarlar ve girdi değerlerini aktarır. Her nöron bir önceki katmandan gelen girdileri kabul eder, girdileri ağırlıklar ile ayarlar, toplar ve ileriye iletir. En iyi değerlerin bulunması için ağırlıkların güncellenmeleri sağlanarak öğrenme sağlanır.

Bir sinir ağının tahminleri, ileriye ve geriye yayılım ile gerçekleşir. İleriye yayılım, bilginin nöronlardan işlenerek ileriye taşınmasını sağlarken; geri yayılım, ağın tahmin edilen çıktısını gerçek çıktıya mümkün olduğunca yakın hâle getirmek için ağırlıkların güncellenmesini sağlar (Raschka ve Mirjalili, 2017). Bir optimizasyon algoritması da kayıpları en aza indirmek için sinir ağının parametrelerini değiştirir.



Şekil 2.21. Yapay sinir ağı modeli

Kaynak: (Wang, 2003).

Yapay sinir ağı modelinde, giriş katmanı, gizli katman ve çıkış katmanındaki nöronlar bulunur. Oklar, farklı nöronlar arasındaki bağlantıları temsil eder. Yapay sinir ağları birçok gizli katmana sahip olarak çok daha karmaşık hâle gelebilir. Bir giriş katmanı, bir çıkış katmanı ile birlikte birden fazla gizli katmanlardan oluşan yapay sinir ağına derin sinir ağı denir ve bu, derin öğrenmenin temelini oluşturur (Wang,

2003). Tipik bir sinir ağı, iki ila üç katmana sahip olabilirken, derin öğrenme ağı yüzlerce katmana sahip olabilir (Lecun, Bengio ve Hinton, 2015). Bir sinir ağı modeli içindeki gizli katmanların sayısı arttıkça hesaplama ve problem çözme yetenekleri artar (Lecun, Bengio ve Hinton, 2015). Derin öğrenmedeki bu çok katmanlı sinir ağı ve nöronlar arasında bağlantılı yollar mevcuttur.

Bir derin öğrenme sistemi, görevlerin tamamlaması için yüksek verimlilik ve performans sağlarken; bir sinir ağı veya standart bir makine öğrenimi sistemi, derin öğrenme sistemine kıyasla düşük verimlilikle görevlerini gerçekleştirir (Middleton, 2021; Wolfewicz, 2021). Standart makine öğrenimi algoritmaları genellikle ön işlemeden geçen yapılandırılmış (etiketli) veriler kullanılırken, derin öğrenme algoritmalarında genellikle yapılandırılmış verilere gerek duyulmaz (Lecun, Bengio ve Hinton, 2015). Derin öğrenme modelleri, standart makine öğrenimi algoritmalarının aksine ham girdiden daha yüksek seviyedeki özellikleri aşamalı olarak birden çok katman kullanarak çıkarır ve verinin yapısına göre hangi parametrelere ne ağırlık verileceğini kendisi keşfeder. Geleneksel makine öğrenimi teknikleri, bir uzman tarafından özellik çıkarımına ihtiyaç duyarken, derin öğrenme modeli, yüksek boyutlu ham verilerden çeşitli özellikleri bağımsız olarak öğrenebilir (Indolia, Goswami, Mishra ve Asopa, 2018). Giriş katmanından çıkış katmanına kadar uzanan her katman gitgide öğrenerek ve daha karmaşık bir soyutlamaya ulaşarak özellikleri tanımlar (Lecun ve diğerleri, 2015; Rusk, 2015). Bu süreç, daha az insan müdahalesi ile daha büyük hacimli veriler işlenebileceğinden verimlidir ancak sistemlerinin kurulması daha fazla zaman ve maliyet gerektirir (Goyal, 2019; Middleton, 2021; Wolfewicz, 2021).

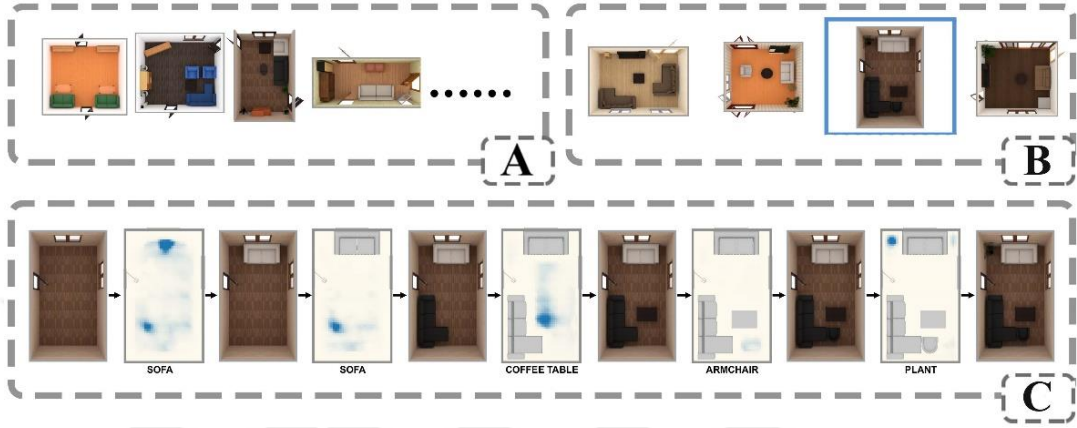
Mimari tasarım pratiği, dijital olarak makine öğrenimi modelleri tarafından manipüle edilebilir. Mimari çizimlerdeki belli özelliklere sahip pikseller ya da görüntüler özel bir bilgi tanımı oluşturabilir. Makine öğrenimi algoritması, bir mimari çizimi tanıması için binlerce veriyle eğitildikten sonra aynı tip çizimlerin ortak noktalarını yakalar. Örneğin algoritma, birçok verinin etiketlenmesinden sonra bir sandalyenin mobilya olduğunu kavrayabilir, insan müdahalesi olmaksızın yeni verileri alıp sıralayabilir. Bu veriler, yeni tasarımlar üretmek için kullanılabilir.

Evrişimsel sinir ağları (CNNs) ve çekişmeli üretici ağlar (GANs) gibi üretken derin öğrenme algoritmaları, mimari tasarım hedeflerinde üretken çözümler sunar (Foster, 2019). Üretken amaçlar için kullanılan makine öğrenimi modeli veri kümesinden öğrenir ve bu deneyimi yeni örnekler üretmek için kullanır (Goodfellow, Bengio ve Courville, 2016). Veri setindekilerle benzer özelliklere sahip yeni tasarım alternatiflerini üretebilir. Bu algoritmalar iç mimari tasarımda özellikle mobilya düzeni ve seçimi için kullanılmıştır (Zhang, Zhang, Liang ve Hall, 2019).

İç mekân tasarımında mobilya ve aksesuarların seçimi, yerleşimi, malzemeleri ve uyumu önemli yönlerden bazılarıdır. Tasarımcı, üç boyutlu sahnedeki nesnelerin her birini birer birer elle konumlandırmak veya nesnelerin uyumluluğunu test etmek yerine makine öğrenimi yaklaşımlarıyla süreci otomatikleştirebilir (Emil Racec, Stefania Budulan, 2016). Üç boyutlu sahne sentezi, bir dizi nesneyi seçerek ve bunları makul bir şekilde dönüştürerek iç mekân düzenlemelerinin yapılması sağlanır (Zhang ve diğerleri, 2019). Çalışmalarda bunun için mevcut sahneler anlaşılıp ardından sahne sentezlendiği bir süreç takip edilmiştir. İlk olarak herhangi bir girdi formatı (metin, RGB-D görüntüleri, kullanıcı etkileşimleri, ilk üç boyutlu sahne modeli, veri kümesi vb.) kabul edilmiş, ardından girdi modele göre yeniden biçimlendirilmiş (ayrıştırılır, dönüştürülür ve uyarlanır) ve son olarak optimize edilmiştir (Zhang ve diğerleri, 2019). Nesnelerin seçilmesi, konumlandırılması, nesnelere arası açık alanlar, nesnelerin erişilebilirliği ve yönlendirilmesini kapsayan işlevsel kısıtlamalar ile birlikte nesnelerin uyumu ve görsel etkisi gibi estetik için kısıtlamalar kullanılmıştır (Zhang ve diğerleri, 2019).

Wang, Savva, Chang ve Ritchie (2018), istenen oda tipi ve girdi olarak duvar yapısı verilen bir odaya nesnelere seçmek ve yerleştirmek için CNN tabanlı model kullanmıştır. Evrişimli sinir ağı modeli, büyük veri kümesiyle eğitilmiştir (Şekil 2.22.'de a). Evrişimli katmanlar, yukarıdan aşağıya görünümü çok kanallı oda temsillerinin görüntülerinden desenler ve nesnelere arasındaki ilişkileri, yerleştirme önceliklerini yakalamıştır. Model, ilk önce bir girdi modelinin yukarıdan aşağıya bir görünümünü hesaplamış, sonra bir nesne ekleyip eklememe olasılığını hesaplamıştır. Ardından olası sahne konumu ve nesne kategorilerinin olasılık dağılımını oluşturarak

hangi tür nesneyi ve nereye yerleştireceğinin tahminini yapmıştır. Bu süreç, her seferinde, her bir nesneyi yinelemeli olarak ekleyerek devam etmiştir (Şekil 2.22.'de c). Böylece bir odanın planlaması sıfırdan yinelemeli olarak oluşturulmuştur (Şekil 2.22.'de b).



Şekil 2.22. A'da veri setindeki örnekler, b'de sentezlenmiş sahneler, c'de birden çok yineleme ile yürütülen bir sahne

Kaynak: (Wang ve diğerleri, 2018).

He ve diğerleri (2020), belirli işlemlere sahip birden fazla odada, stil uyumu yakalayan iç mekân nesnelere seçmek bir iç mekân sahne sentezi yaklaşımı önermiştir. Sistem, seçim için iki düzeyli öneri vermiştir: kategori düzeyi ve örnek düzeyi (Şekil 2.23.'de B). Kategori düzeyi, belirli oda tipleri için nesne kategorileriyle ilişkilendirilirken, örnek düzeyi, belirli kategorideki örneklerin görünüm çeşitliliğiyle ilişkilendirilmiştir.

Belirli oda tiplerine göre stil uyumlu iç mekân oda içerikleri önermek için bir denetimli makine öğrenimi modeli benimsenmiştir. Veri kümesinden çeşitli oda tiplerindeki nesne kategorilerinin bilgileri alınarak model eğitilmiştir. Böylece model, nesne kategorileri ile oda tipleri arasındaki gizli ilişkiyi ortaya çıkarabilmiştir. Model, bu durumda her oda tipi için nesne kategorisi oluşturma olasılığını verebilmiştir. Her oda bir veya daha fazla oda tipiyle etiketlendiğinde, bir kat planı verildiğinde sırasıyla her kategori için olasılık toplanmıştır.

Bir konu modeline (oda tipi) göre nesne kategorilerinin oluşma olasılığı dağılımı hesaplandıktan sonra odalar arasındaki ilişkiler dikkate alınarak işlev çeşitliliği ve stil uyumluluğu göz önünde bulundurulmuş her kategoriden nesnelere örneklenmiştir. Her oda için, kullanıcı tarafından sağlanan stil şablonuna dayalı beş renkli bir stil paleti hesaplanmıştır. Alanın boyutu, işlev çeşitliliği ve stil uyumu ile ilgili çoklu kısıtlamalara tabi olan bir optimizasyon yöntemiyle nesne örnekleri seçilmiştir. Böylece bir dizi öneri listelerinden tüm ev için hem boyut uyumluluğunu hem de stil uyumluluğunu en iyi karşılayan örneklerin bir kombinasyonu bulunmuştur.

Aşağıda stil odaklı mekan sentezi örneği verilmiştir (He ve diğerleri, 2020). Buna göre, belirli işlevlerde ve stil uyumlu odalar için girdiler (a) verilmiştir. Yöntemin girdisi, her odaya bir veya daha fazla etiket atanmış bir kat planı ve stil şablonu için örnek bir görüntüdür. Belirlenen kategorideki nesne örnekleri seçildikten sonra örnekler, boyut (alan işgal eden nesnelere), grup (belirli nesnelere kendi aralarındaki ilişkileri), stil kısıtlamalarına göre optimize edilmiştir (b). Her oda için referans stil ile benzer paranoma şablonuna göre iç mekân sentezi tamamlanmıştır (c).



Şekil 2.23. Stil odaklı mekan sentezi modeli.

Kaynak: (He ve diğerleri, 2020).

Birden fazla odada stil uyumluluğu elde etmek için model geliştirilmiştir. Konut düzeyinde stil uyumluluğu için odalar arasındaki nesne bağlantıları göz önünde bulundurulmuştur. Böylece, her odayı ayrı ayrı ele almak yerine, odalar arasındaki ilişkileri dikkate alarak nesne örnekleri seçilebilmiştir. Oda içi stil uyumluluğunu geliştirmek için nesne grubu kavramı tanıtılmıştır. Grup kısıtlaması, gruplardaki örnek parçalar arasındaki malzeme uyumluluğunu dikkate almıştır. Örnek önerilerini

kısıtlamak için odanın büyüklüğü ve kullanıcı faaliyetleri göz önünde bulundurularak sınırlı sayıda mobilya kullanılabilmesi için boyut (kullanılan alan) kısıtlaması uygulanmıştır.

2.4.7.2. Makine Öğreniminin Hesaplamalı Tasarım ve Diğer Hesaplamalı Tasarım Yaklaşımları ile İlişkisi

Veriden öğrenmeye dayalı yaklaşımda, bir işlemi çözmek için herhangi bir algoritmayı açıkça kodlamak yerine algoritmanın modeli ve çıktısı öğrenmesi için onu makine verileri ile beslemek amaçlanmıştır. Böylece makinenin probleme nasıl yaklaşacağı öğretilmiştir. Algoritmayı kodlayarak doğrudan kesin sonuçlara gitmek yerine genelleme ve tahmine dayalı yaklaşıma gidilmiştir. Veriler ile düşünmek hesaplamalı düşünmeyi ve yüksek derecede hesaplamalı bir işlemi gerektirdiği görülür.

Makine öğrenimi, diğer hesaplamalı yaklaşımlarla ve yöntemlerle entegre kullanılarak daha verimli tasarım süreçleri sağlamıştır. Örneğin, makine öğrenimi yöntemleri, parametrik/algoritmik modelleme aracı ile entegre olarak kullanılabilir (Vukorep ve Kotov, 2021).

Makine öğrenimi modelleri, tasarım alanını daha geniş bölgelerini daha hızlı araştırarak önemli kalıplar çıkarma becerisine sahip olduğu için, parametrik modellerin yüksek boyutlu tasarım çıktılarını daha etkili bir şekilde anlamak ve verilerden tahminlerle karmaşık simülasyon ile baş etmek için önemli avantajlar sağladığı görülmüştür (Tamke, Nicholas ve Zwierzycki, 2018). Makine öğrenimi, parametrik tasarım komut dosyasının sonunda bulunan nihai çıktıyı tahmin ederek, tasarım modelindeki karmaşık tasarım süreçlerini güncellerken veya değiştirirken harcanan zamanı kısaltmıştır (Sebestyen, 2021). Maliyetli değerlendirmelerin sayısını azaltarak ve gereksiz değişkenleri hariç tutulmasını sağlayarak performans analizi ve optimizasyon sürecini hızlandırır (Pereira, Belém ve Leitão, 2020). Makine öğrenimi yaklaşımıyla oluşturulan vekil modeller performans optimizasyon problemlerinde geniş tasarım alanlarının yaklaşık tahminleriyle daha hızlı ve verimli keşfedilmesini sağlar (Kozziel ve diğerleri, 2011; Wortmann ve diğerleri, 2015).

Diğer yandan özellikle derin öğrenme modelleri üretken bir potansiyel gösterir. Yeni bir ürün üretmek için görüntülerden özellik öğrenebilir. Üreticinin bir parçası olarak bir sinir ağı kullanmanın birçok tekniği vardır (Veloso ve Krishnamurti, 2021: 47-53). Bu anlamda veriden öğrenme, hesaplamalı tasarımın bir parçası olarak rol üstlenir.



3. İÇ MİMARİ TASARIMDA HESAPLAMALI TASARIM SÜRECİ

Hesaplama tasarımı süreci ve düşünce yapısı, iç mimari tasarıma yeni bir perspektif ve yaratıcı fırsatlar sunarak, daha optimize edilmiş ve kullanıcı odaklı iç mekanlar yaratmayı hedefler. Hesaplama düşünme, geleneksel iç mimari tasarım süreçlerine analitik ve yenilikçi bir bakış açısı getirerek tasarımın verimliliğini ve işlevselliğini artırır.

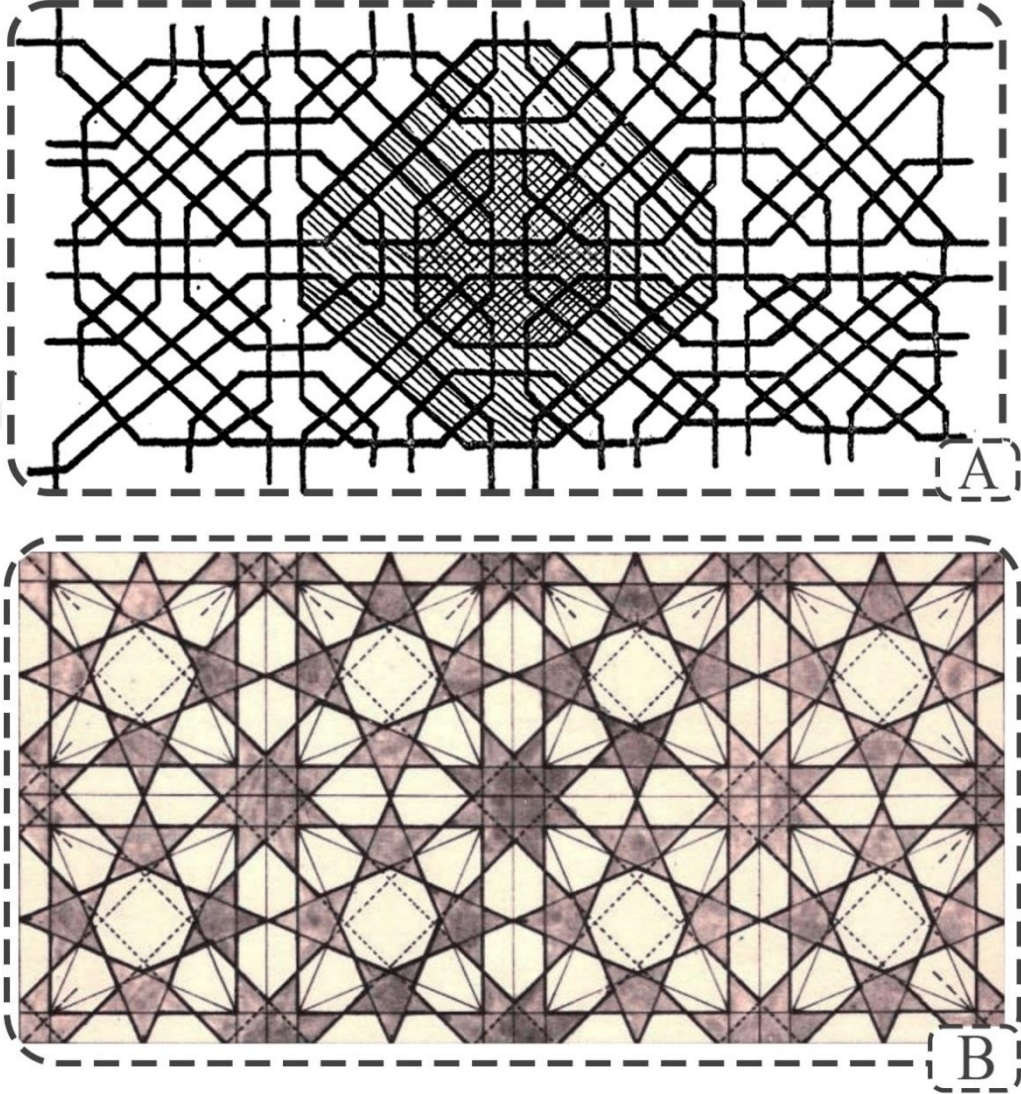
İç mimari proje örneği üzerinden hesaplama düşünmenin nasıl uygulanabileceği ve nasıl sonuçlar elde edilebileceği incelenebilir. Bir iç mimar, bir iç mekân tasarım görüntüsüne ulaşmak için bir parametrik/algortmik tasarım aracı olan Grasshopper'da tasarım sürecini planlarken hesaplama düşünür (problemi, formüle eder). Yani, tasarım sürecini (formüle edilmiş hedefi), parametrik tasarım modeline göre (hesaplama modele göre) parametrik/algortmik tasarım yazılımının çalıştığı bilgisayarın gerçekleştirilmesi için planlar. İç mimar, bu süreçte, hedefini adım adım planlamak için algortmik düşünür. Farkında olarak ya da olmayarak ayrıştırma, soyutlama, genelleme ve değerlendirmeyi sürece dâhil eder. Bunu yaparken de modellemeden, simülasyondan, görselleştirmeden, kodlamadan ve otomasyondan yararlanabilir.



Şekil 3.1. Hesaplama düşünme sürecini gösteren temsili görsel

Bir örüntü veya fonksiyon analiz edildikten sonra bir iç mekân donatısının tasarımı için hesaplama adımları olarak kurgulanabilir. Örneğin, Anadolu Selçuklu sanatında kullanılan geometrik motifler analiz edildikten sonra benzer motiflerin

tasarlaması için algoritma adımları olarak yeniden planlanabilir. Bu algoritma ile alternatif motiflere sahip seperatörler tasarlanabilir (Rhino Grasshopper, 2019).

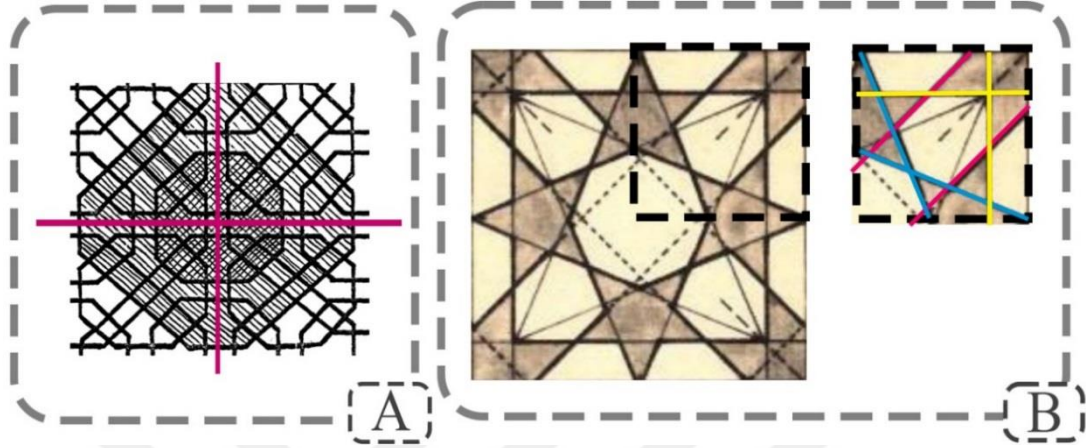


Şekil 3.2. İlham alınan geometrik örüntüler

Kaynak: A'da (Şimşir, 1990), b'de (Akar ve Keskiner, 1978).

Anadolu Selçuklu sanatında kullanılan geometrik motifler, çizgilerin kırılma açılarının eşkenar dörtgen, altı köşeli yıldız ve kare gibi geometrik desenleri oluşturduğu zengin motiflerdir (Şimşir, 1990). Örnek geometrik desenlerin tekrar ettiği ve bu desenlerdeki tekrar eden en küçük parçaların simetrik olarak birbirini tamamladığı görülür (Şekil 3.2.). Bu örüntülerin tekrar eden en küçük parçaları tespit

edilir. Oluşturulacak model örnek görsellerde (Şekil 3.3.) olduğu gibi çizgilerine kadar ayrıştırılır. Oluşturulan modelde, bu desenlerde olduğu gibi bu en küçük parçaların yansımaları alınarak bir bütün oluşturulur (Rhino Grasshopper, 2019).



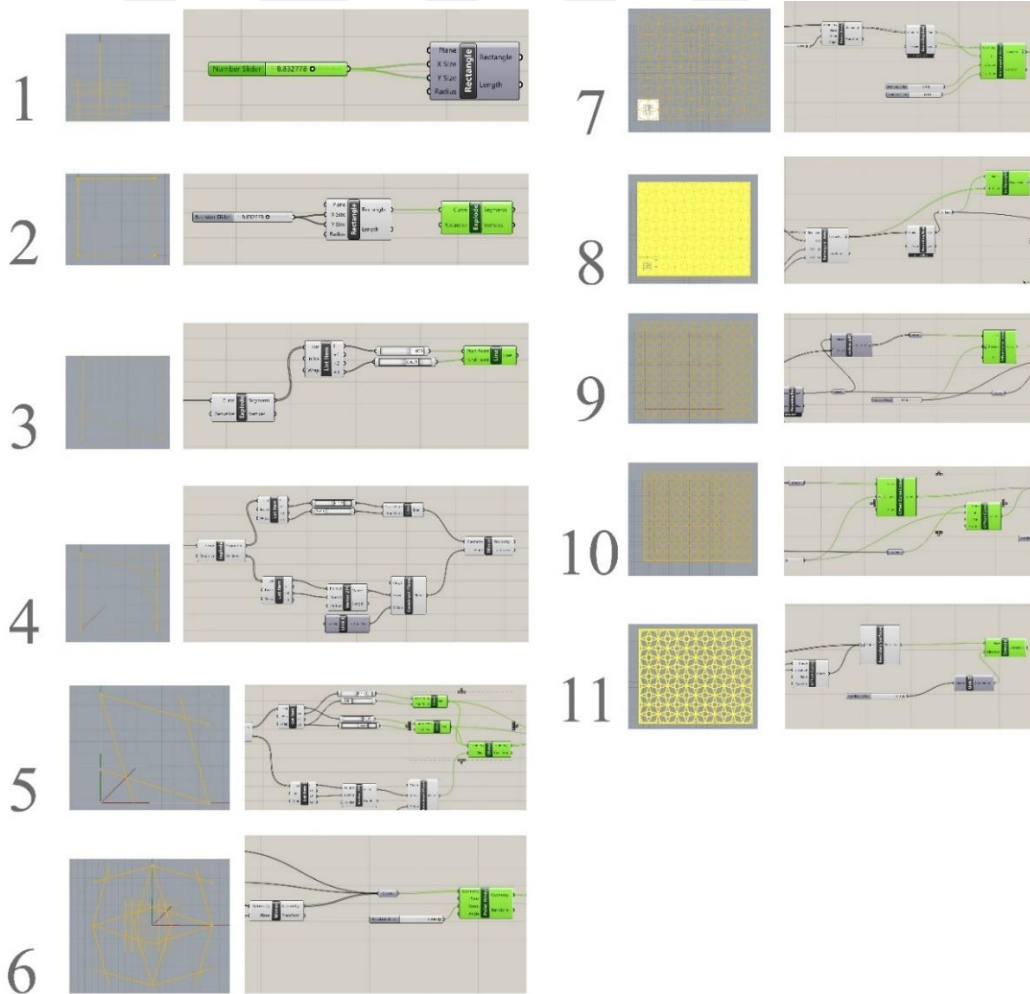
Şekil 3.3. Örnek görsellerdeki tekrar eden ve birbirlerinin yansımaları olan kompozisyonlar

Genelleme, belirli çözümlerin ortak yanlarını yakalayıp daha geniş bir uygulanabilirlik için bu ortak yanları kullanabilmeyi sağlar. Daha önce kullanılan belirli çözümler ve algoritmalar benzer amaçlara ya da hedeflere yönelik kullanılması süreci hızlandırabilir. Bu anlamda daha önce karşılaşılan benzer bir çözümün bazı alt parçaları (Rhino Grasshopper, 2019), benzer başka bir problemin çözüm süreci içinde kullanılabilir. Böylece yeni çizgi kombinasyonlarından yeni motifler üreten ve bu motiflerden farklı seperatör tasarımları çıkartan bir prosedür yaratılır. Problem ayrıştırılarak ve gereksiz ayrıntılardan arındırılarak algoritma yaratılır. Bu algoritma, tasarım sürecinin bir soyutlaması olarak görülebilir (Şekil 3.4.).

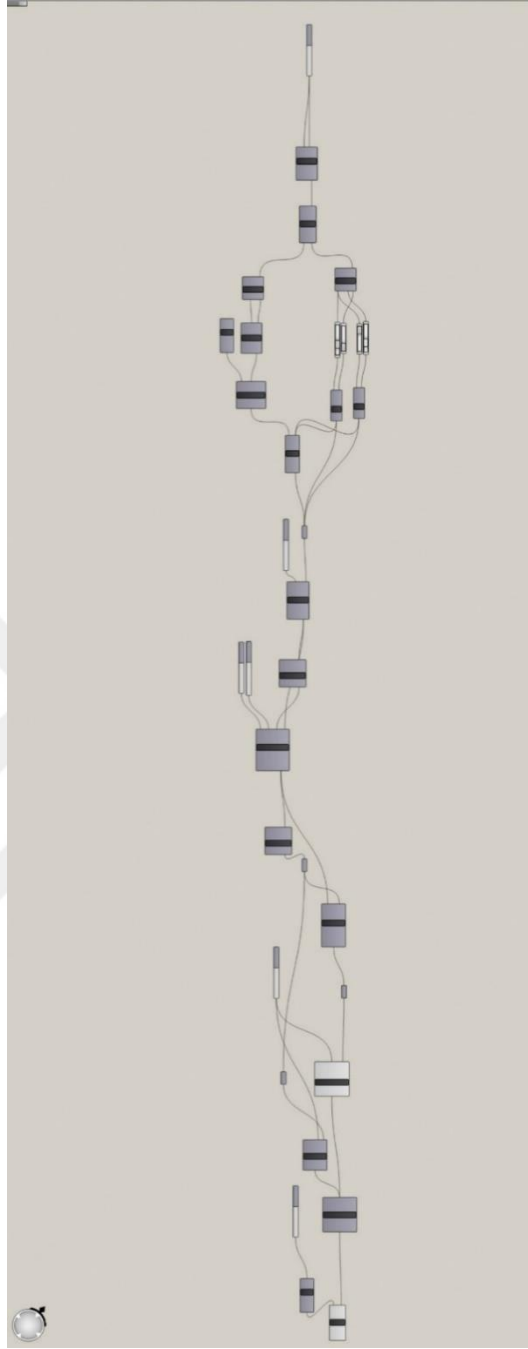
Grasshopper, kullanıcılarının işlem adımlarını sıralı ve doğrusal bir şekilde düşünmesini zorunlu kılmaz (Öksüz, 2017). Ancak takip edilen işlemler problemin ele alınış biçimine ve takip edilen iş akışına göre planlanabilir. Bu anlamda bu işlemler, bileşenler ya da bileşen grupları hesaplama adımları olarak gösterilebilir (Şekil 3.5.).

Hesaplama adımları tanımlanır. İlk olarak örnek desende tekrar eden en küçük parçanın sınırlarını belirlemek için “rectangle” bileşeni kullanılır ve oluşturulan karenin x ve y koordinatlarını kontrol edebilen kaydırıcı (number slider) oluşturulur. İkinci adımda karenin kenarları üzerinde belirlenecek noktalar ile oluşturulması

planlanan çizgilerin açılarının kontrol edilebilmesi amaçlanır. Bunun için öncelikle “explode” bileşeni ile dikdörtgen eğrilere ayrılır. Üçüncü adımda “List item” ile belirlenen iki kenar üzerinde “point on curve” bileşeni ile nokta konumları belirlenir. Bu noktalar “line” bileşeni ile birleştirilerek çizgi oluşturulur. Dördüncü ve beşinci adımda bu çizginin z ekseninde yansımalarının oluşturulması için mirror bileşeni kullanılır. Benzer işlemler, simetrisi alınması planlanan diğer çizgi için de tekrarlanır. Yansıtılmış çizgi grupları “curve” bileşeni ile bir araya toplanır. Altıncı adımda oluşturulan parçanın “Polar array” bileşeni ile XY eksenlerinde yansımalarının oluşturulması sağlanır. Yedinci adımda son oluşturulan grubu x ve y eksenlerinde kopyalayarak devam ettirmek için bir kaydırıcı ile birlikte “rectangular array” kullanılır. Diğer adımlar, elde edilen çizgilere kalınlık verilmesi ve oluşan şeklin yükseltilmesi için yapılan işlemlerdir.



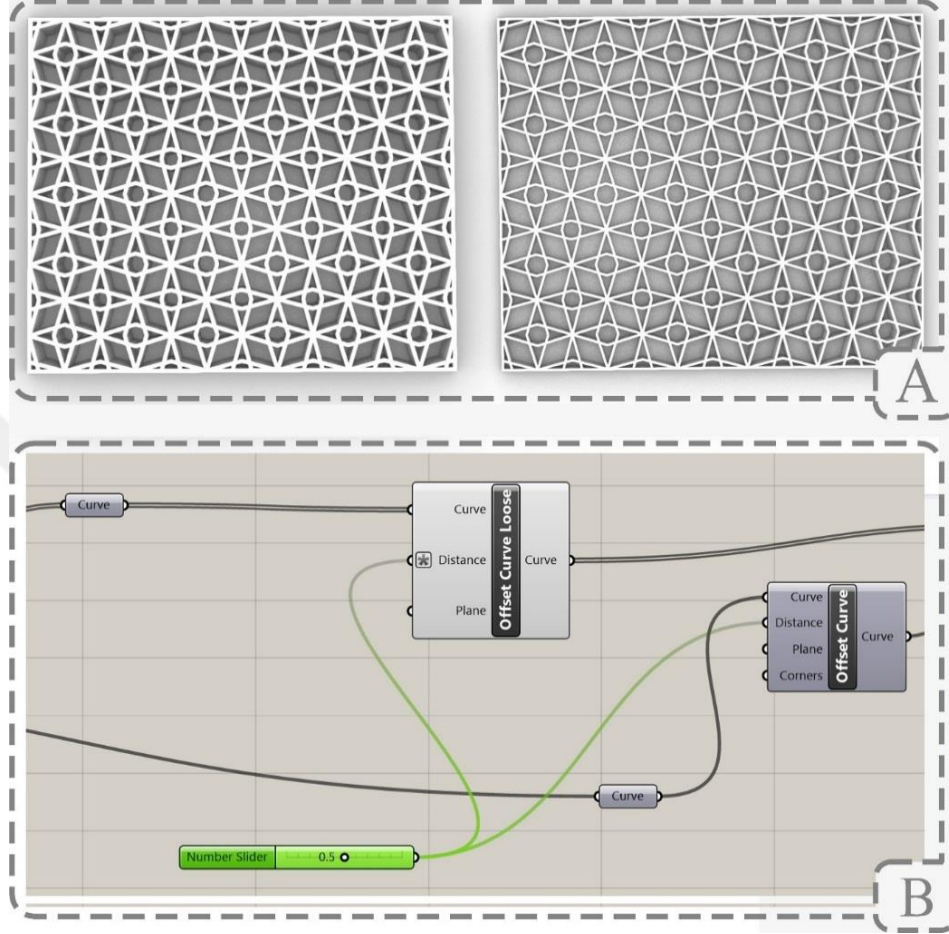
Şekil 3.4. Uygulanan prosedürün adımları



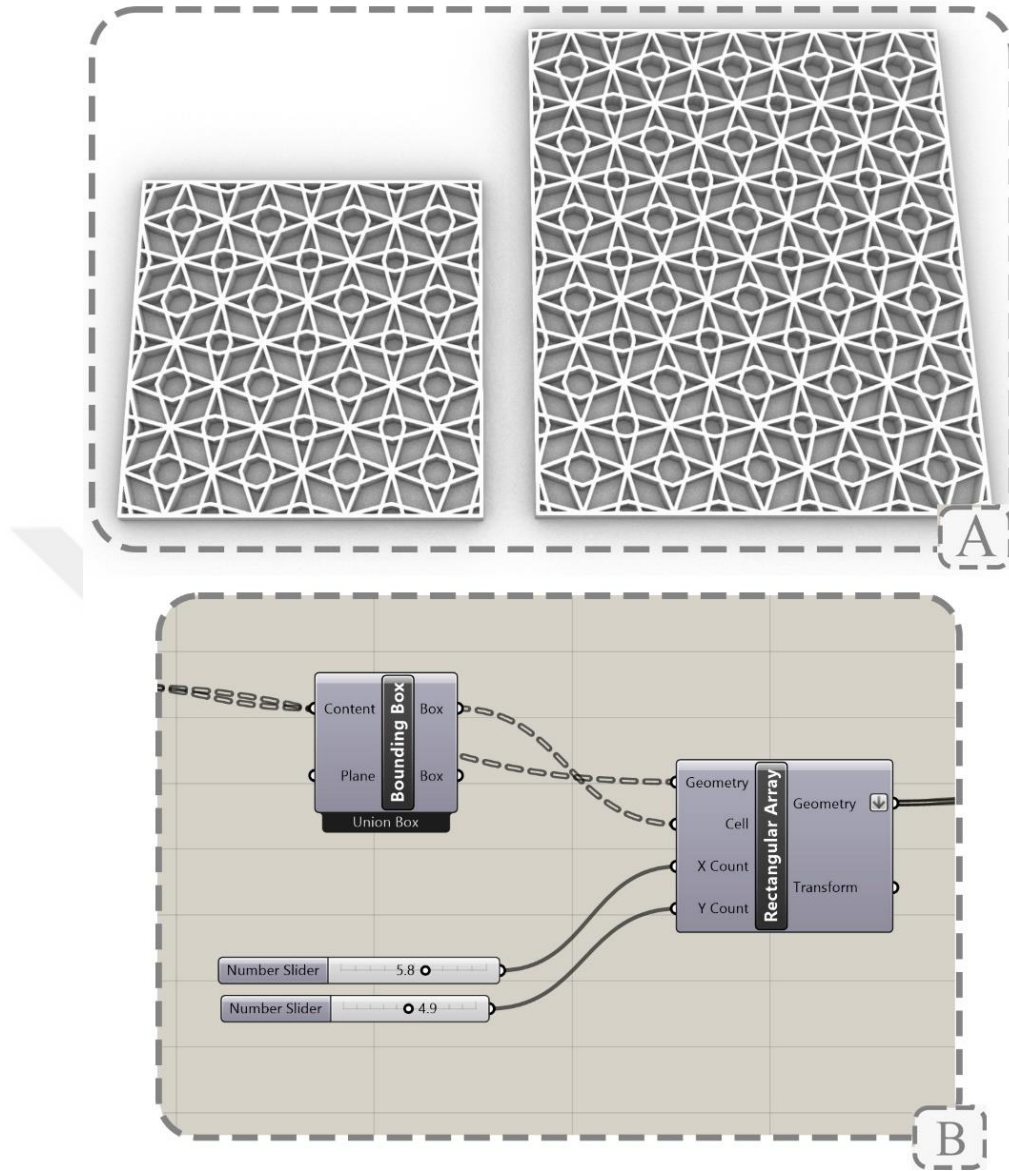
Şekil 3.5. Prosedürün tamamı

Modelin değişkenlerinin değiştirilmesi ile seperatörün şekillenmesini sağlayacak prosedür yaratılabilir. Modelin boyutunun ayarlanması, motifin deseninin değiştirilmesi, modeldeki desenin çizgi kalınlığının ayarlanması ve modelin yüksekliğinin belirlenmesi sağlanabilir. Böylece algoritmik ve parametrik kurgu sayesinde model motifte bozulmalar olmadan istenilen ölçülere göre optimize

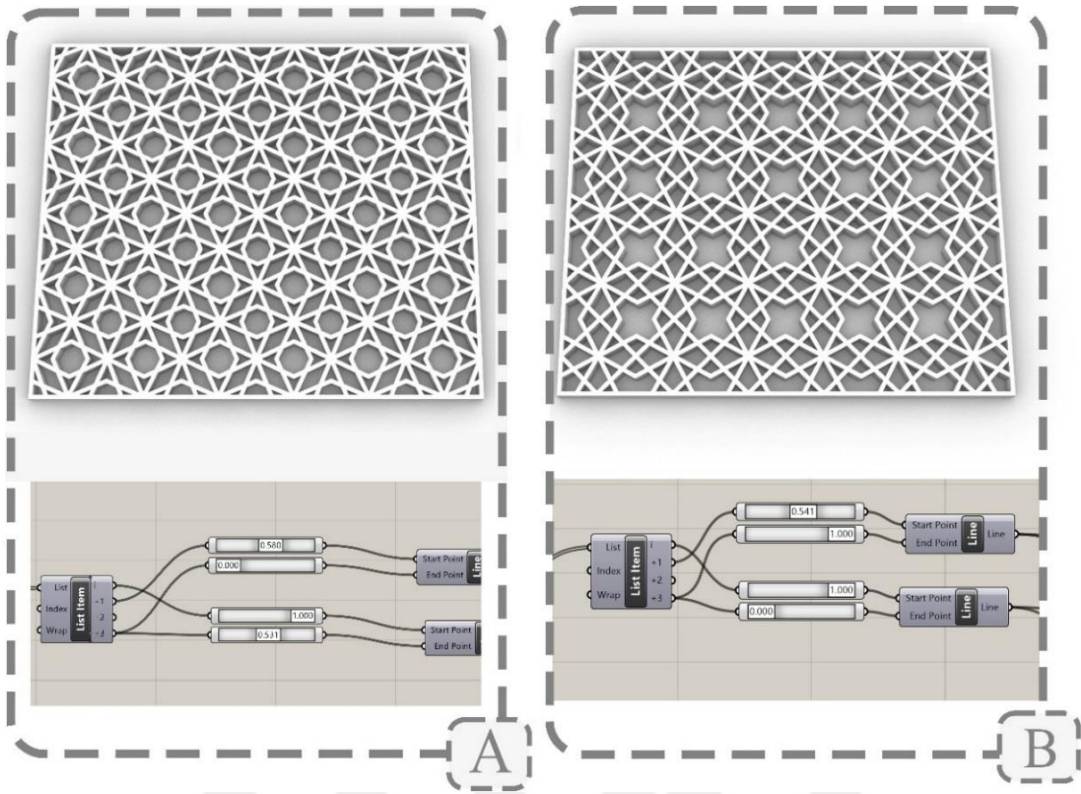
edilebilir hale gelir. Diğer yandan deęişkenlerin deęiştirilmesi ile alternatif motifler üretilebilir.



Şekil 3.6. Çizgi kalınlığı ayarı



Şekil 3.7. Örüntünün bozulmalar olmadan boyutunun ayarlanması



Şekil 3.8. Değişkenlerin değiştirilmesi ile üretilen örneklerden bazıları



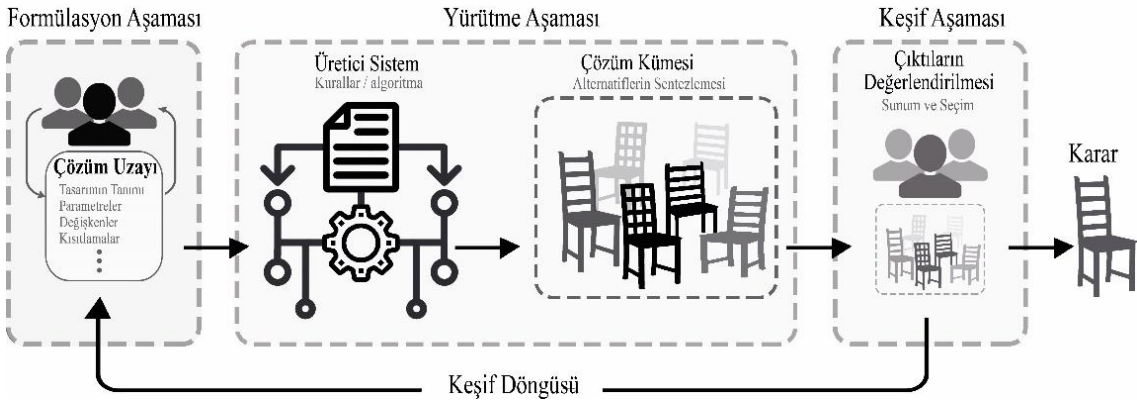
Şekil 3.9. Seperatör örneklerinin kullanımı

4. BULGULAR ve YORUMLAR

Hesaplamalı mimari tasarım; Leonardo da Vinci'nin üretici sistemlerinden Gaudi'nin fiziksel modellerine, ilk dijital parametrik-algoritmik tasarım araçlarından makine öğrenimine kadar çok yol kat etti. Bu çalışmaların temelinde bir hesaplamalı düşünme faaliyeti yatmaktadır.

Her hesaplamalı yöntemin kendine has terminolojisi, bileşenleri ve teknikleri vardır. Ancak hesaplamalı tasarım süreçleri genel olarak üç aşamaya ayrılabilir: formülasyon, yürütme ve keşif aşamaları. Bu aşamalar, çoğu araştırmacının üretken tasarım sürecinin aşamaları için belirledikleriyle kesişir (Bohnacker, Groß, Laub ve Lazzeroni, 2009; Cagan, Campbell, Finger ve Tomiyama, 2005: 171; Dino, 2012; Farzaneh, 2017: 191; McKnight, 2017: 178; Mukkavaara ve Sandberg, 2020: 4; Reyling, 2019: 19; Veloso ve Krishnamurti, 2021: 29).

Hesaplamalı düşünme, genel anlamda formülasyon aşamasının olduğu yerdir. Dolayısıyla bu yaklaşımları 'hesaplamalı' kılan, bu modellerin (araçlar) formüle edildiği aşamadır. Tasarımcı, formülasyon aşamasında hesaplamalı düşünerek genellikle model, araç ya da prosedürü formüle eder. Formülasyon aşamasında, tasarım amacı ve prosedürü bir makinenin, insanının ya da bilgisayarın yürütebileceği şekilde soyutlanır. Formüle edilen bu yardımcılardan amaca uygunluğu denetlendikten sonra problemi yürütmesi sağlanır ve ardından çözümler keşfedilir (Şekil 4.1.).



Şekil 4.1. Hesaplamalı tasarım ve sonrasında gösteren sürecin temsili

Hesaplama mimari tasarım yaklaşımları, tasarım sürecine farklı bakış açıları getirmektedir. Her bir yaklaşım, tasarım sürecinde birçok fırsat sunarken, belirli farklılıkları da beraberinde getirir. Aşağıda bu farklı hesaplama mimari tasarım yaklaşımları karşılaştırılarak, her birinin odaklandığı temel nokta ve her birinin birbirlerine göre öne çıkan özellikleri incelenmiştir (Çizelge 4.2.).

Çizelge 4.1. Hesaplama mimari tasarım yaklaşımlarının odaklandıkları temel noktalar ve bu yaklaşımların öne çıkan özellikleri

Yaklaşım ve Yaklaşımın Odaklandığı Temel Nokta	Öne Çıkan Özellikleri
<p>Parametrik Yaklaşım</p> <p>Kısıtlamalar yoluyla parametrik ilişkilerin incelenmesi</p>	<ul style="list-style-type: none"> • Geometrik ve akışkan formların tasarım süreçleriyle ilişkilendirilmeye meyillidir. Topolojik tasarımlar ile daha çok ilişkilendirilebilir. • Tasarım süreci açık olarak yürütülmek zorunda değildir. • Parametrik tasarım yapılırken hesaplama düşünülmesi zorunlu değildir. Dolayısıyla parametrik tasarım, hesaplama tasarımla doğrudan ilişkili değildir. • Kısıtlama, parametre ve değişken kavramları temel bileşenleridir.
<p>Algoritmik Tasarım</p> <p>Tasarım sürecinin açık bir alanda adım adım planlanması</p>	<ul style="list-style-type: none"> • Geometrik ve akışkan formların tasarım süreçleriyle ilişkilendirilmeye meyillidir. Topolojik tasarımlar ile daha çok ilişkilendirilir. • Tasarım sürecinin tamamını kontrol edilebilir. • Programlama ve kodlama kavramı ile sıklıkla ilişkilendirilir. • Hesaplama düşünme bilincinin tamamen açık olduğu bir yaklaşımdır.
<p>Üretken Yaklaşım</p> <p>Tasarım sürecini yönetecek üreticinin tasarlanması</p>	<ul style="list-style-type: none"> • Sadece parametreler ve kısıtlamalar ile ilişkilendirilmez; şekil grameri, hücresel otomat gibi kural tabanlı yöntemlerden makine öğrenimi modellerine kadar üretken bir hesaplama model oluşturmak mümkündür. • Hesaplama mimari tasarım için diğer yaklaşımları kapsayan bir şemsiye terim olarak kullanılabilir. • Üretici sistemin alternatif çıktılar üretip çeşitlilik sağlamasına odaklanılır.

<p>Performansa Dayalı Yaklaşım</p> <p>Belirli performans hedeflerinin karşılanması</p>	<ul style="list-style-type: none"> • Performansa dayalı tasarım, simülasyonların performans analizinde kullanımıyla ilişkilendirilmeye meyillidir. • Performansa dayalı hesaplamalı tasarım, tasarım sürecinde performansı optimize etmeyi amaçlar. Diğer hesaplamalı yaklaşımlar, performans odaklılığı bu kadar belirgin bir şekilde vurgulamayabilir veya performansı optimize etme amacı gütmeyebilir. • Performansa dayalı hesaplamalı tasarımda, tasarım kararları nesnel verilere dayanılarak alınır. Daha objektif ve doğru tasarım seçimlerine ulaşılır. • Performansa dayalı hesaplamalı tasarım, genellikle çok disiplinli optimizasyon yöntemlerini kullanır. Bu yaklaşım, tasarım sürecinde farklı disiplinler arasındaki etkileşimi ve bağımlılıkları ele alır.
<p>Optimizasyona Yönelik Yaklaşım</p> <p>Tasarım seçenekleri arasında en iyisi ya da en iyilerinin araştırılması</p>	<ul style="list-style-type: none"> • Optimizasyona dayalı hesaplamalı tasarım, birden fazla tasarım parametresini aynı anda optimize etme yeteneği ile öne çıkar. Bu, tasarımda birden çok değişkenin aynı anda dikkate alınmasını ve optimize edilmesini sağlar. Diğer hesaplamalı yaklaşımlar genellikle tek bir değişken üzerinde yoğunlaşır. • Optimizasyona dayalı hesaplamalı tasarım, genellikle yinelemeli bir süreci içerir. Tasarım parametreleri ve hedef fonksiyonları üzerinde tekrarlayan hesaplamalar ve analizler yapılır. Bu iteratif süreç, tasarımın iyileştirilmesine ve en uygun çözüme yaklaşılmaya olanak sağlar. Diğer hesaplamalı yaklaşımlar daha doğrusal veya basamaklı bir süreç izleyebilir. • Birçok optimizasyon algoritması bulunur. Amaca uygun en uygun optimizasyon algoritmasını bulmak ayrı bir zorluktur. • Optimizasyona dayalı hesaplamalı tasarım, tasarım problemini matematiksel bir model olarak ifade eder. Tasarım parametreleri, hedef fonksiyonları ve kısıtlar matematiksel ifadelerle temsil edilir. Bu matematiksel modelleme, optimizasyon algoritmalarının kullanılmasını sağlar. Diğer hesaplamalı yaklaşımların bazıları daha veri odaklı veya deney tabanlı olabilir. • Optimizasyona dayalı hesaplamalı tasarım, karar verme analizi için kapsamlı bir çerçeve sunar. Tasarım sürecindeki farklı seçenekleri

	<p>değerlendirmek, riskleri analiz etmek ve alternatif çözümleri karşılaştırmak için optimize edilmiş modeller kullanılır.</p>
<p>Doğadan İlham Alan Yaklaşım</p> <p>Doğadaki formları, süreçleri ve sistemlerden ilham alma</p>	<ul style="list-style-type: none"> • Diğer hesaplamalı tasarım yaklaşımlarından farklı olarak, biyolojik ilham alan yaklaşım doğal evrimsel süreçlerden öğrenilmiş özellikleri ve adaptasyon yeteneklerini kullanır. • Bu yaklaşım, geleneksel optimizasyon yöntemlerinin sınırlarını aşarak, tasarımın daha özgün ve ilham verici olmasını sağlayabilir. • Doğadan ilham alan hesaplamalı tasarım, değişen çevresel koşullara uyum sağlama yeteneğiyle öne çıkar. Biyolojik sistemlerde olduğu gibi, bu yaklaşım tasarımın esnekliğini ve adaptasyon yeteneğini artırmayı amaçlar. • Doğadan ilham alan hesaplamalı tasarım, farklı disiplinlerin birleşimini gerektirir. Biyolojiden, matematikten, bilgisayar biliminden ve mühendislikten alınan bilgilerin entegrasyonunu gerektirir. • Doğadan ilham alan hesaplamalı tasarım, değişen çevresel koşullara uyum sağlama yeteneğiyle öne çıkar. Biyolojik sistemlerde olduğu gibi, bu yaklaşım tasarımın esnekliğini ve adaptasyon yeteneğini artırmayı amaçlar. Doğal sistemlerde olduğu gibi, tasarım süreci sırasında değişkenlikler ve belirsizliklerle başa çıkabilme kabiliyeti öne çıkar. • Doğadan ilham alan hesaplamalı tasarım, doğa ile uyumlu tasarımların geliştirilmesini teşvik edeceği için doğal olarak sürdürülebilirlik tasarımı öncelikle olur.
<p>Veriye Dayalı Tasarım</p> <p>Tasarım sürecindeki kararları verilere dayalı</p>	<ul style="list-style-type: none"> • Makine öğrenimi tabanlı yaklaşım, karmaşık ve çok boyutlu veri setlerini analiz etme ve bunlardan anlamlı bilgiler çıkarma yeteneğine sahiptir. Bu, tasarım sürecinde daha geniş bir bakış açısı sunar ve karmaşık ilişkileri tespit edebilir. Diğer

analizler ve makine öğrenimi kullanarak almak	<p>hesaplama tasarımı yaklaşımlarından daha esnek ve adaptif olabilir.</p> <ul style="list-style-type: none"> • Makine öğrenimi, geçmiş verileri kullanarak gelecekteki sonuçları tahmin etme yeteneğine sahiptir. Tasarım sürecinde bu, gelecekteki performansı tahmin etmek ve tasarım kararlarını bu tahminlere göre optimize etmek için kullanılabilir. Diğer hesaplama tasarımı yaklaşımlarında genellikle bu kadar tahmine dayalı bir optimizasyon bulunmaz. • Veriye dayalı hesaplama tasarımı, otomatik öğrenme yeteneklerini kullanarak tasarımın sürekli olarak güncellenmesini ve uyarlanmasını sağlar. Makine öğrenimi modelleri, yeni veri akışlarıyla beslendikçe ve geri bildirim alındıkça güncellenebilir ve iyileştirilebilir. Bu şekilde, tasarım sürekli olarak optimize edilebilir ve yeni bilgilere dayalı olarak geliştirilebilir. • Bu yaklaşım, tasarım sürecinde veri analizi ve modelleme ile örüntüleri ve ilişkileri ortaya çıkarmayı hedefler. Diğer hesaplama tasarımı yaklaşımları genellikle daha deterministik ve önceden belirlenmiş kurallara dayalıdır.
-----------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Her hesaplama yönteminin kendine göre avantajı olsa da bir tasarım hedefini birden çok yöntem gerçekleştirebilir. Günümüzdeki hesaplama tasarımı modelleri, geleneksel tasarım sürecinde ulaşılamayan karmaşıklık düzeyinde birçok tasarım alternatifini üretebilir. Tasarımcılar, amaçları için mümkün olan en hızlı ve en verimli tasarım modelini bulmaya çalışırlar. Bu anlamda yöntem ve modelleri neyi sağladıklarından çok bunu nasıl sağlayabildikleri ile tanımlamak daha uygundur.

Hesaplama yöntemleri, kullanılan farklı bileşenler ve işlemlerle birbirlerinden farklılaşsalar da aralarında keskin bir ayrım olmadığı gözlenmiştir. Bir yöntem, teknik veya araç, birçok yaklaşımdan beslenebilir. Günümüzde hesaplama araçları birçok yaklaşımı bir arada içermektedir ve farklı hesaplama yaklaşımları içeren birçok araç entegre şekilde kullanılabilir. Bu nedenle hesaplama yaklaşımları birbirlerinden keskin bir şekilde ayırmak yerine, odaklandıkları fikirler ve öne çıkan özellikleri üzerinden değerlendirmek daha uygundur.

İç mimari tasarım sürecinde birçok hesaplamalı yaklaşımdan yararlanılabilir: Parametrik, algoritmik, üretken, doğadan ilham alan, optimizasyona dayalı ve veriden öğrenmeye dayalı yaklaşımlar. Bu yaklaşımlar altındaki birden çok yöntem ve model, avantajlarıyla birlikte ulaşılmak istenen hedefe göre iç mimari tasarım süreçlerinde kullanılır. Bu modeller ve yöntemler, birbirleriyle birçok yönden kesişen yaklaşımlardan beslenir. Yaklaşımlar birbirlerinden etkilense de tasarım süreçlerinde odaklandıkları fikirler ve kullanılan yöntemler ile birbirlerinden ayrılırlar. Aşağıda hesaplamalı mimari tasarım yaklaşımlarının iç mimari tasarım sürecine etkileri değerlendirilmiştir (Çizelge 4.3).

Çizelge 4.2. Hesaplamalı mimari tasarım yaklaşımlarının iç mimari tasarım sürecine etkileri

Yaklaşım	Avantaj	Dezavantaj
Parametrik Yaklaşım	<ul style="list-style-type: none"> • Esneklik ve Özelleştirme: Değişkenlerin değiştirilmesiyle yeni sonuçlar elde edilir. Bu iç mekan tasarımında esneklik sağlar ve iç mekanın özelleştirmesini mümkün kılar. Örneğin, bir ofis projesinde, değişkenlerle oynayarak mobilya ve bölme sistemleri gibi iç mekan unsurları kişiselleştirilebilir. Parametrelerin değiştirilmesiyle, farklı konfigürasyonlar ve ölçekler elde edilebilir, böylece mekanın işlevselliğine ve kullanıcı ihtiyaçlarına daha iyi uyum sağlanır. • Verimli Planlama ve Alan Kullanımı: Parametrik modelde değişkenler ile etkileşime girmek, iç mekanların planlanması ve alan kullanımının optimize edilmesi konusunda büyük bir avantajlar sunar. Örneğin, bir konut projesinde, parametrik model 	<ul style="list-style-type: none"> • Karmaşıklık: Parametrik modelin karmaşıklığı, iç mekan tasarımında farklı bileşenlerin ve unsurların uyumlu bir şekilde bir araya getirilmesini zorlaştırabilir. • İnsan Odaklı Tasarımın Azalması: Parametrik modelin kullanımı, bazen estetik kaygılar kaygılar gibi ilgili kullanıcı ihtiyaçları, ergonomi ve kullanılabilirlik gibi insan odaklı tasarım unsurlarının geri planda kalmasına neden olabilir. • Uyumsuzluk Sorunları: Parametrik modelin karmaşıklığı, iç mekan tasarımında farklı bileşenlerin ve unsurların uyumlu bir şekilde bir araya getirilmesini zorlaştırabilir.

	<p>kullanarak iç mekanın en verimli şekilde düzenlenmesi hedeflenebilir. Parametreler aracılığıyla mobilya yerleşimi, dolaşım akışı ve depolama alanı gibi unsurlar optimize edilerek, kullanılabilir yaşam alanı artırılabilir ve gereksiz boşluklar en aza indirilebilir.</p> <p>• Estetik ve Biçimsel Çeşitlilik:</p> <p>Parametrik kısıtlamalardan oluşan model, iç mekanlarda estetik ve biçimsel çeşitlilik sağlamada etkili bir araçtır.</p> <p>• Performans Optimizasyonu:</p> <p>Parametreler arasındaki ilişkilerin hesaplanmasıyla iç mekanlarda performans optimizasyonu sağlanabilir. Parametreler arasındaki ilişkilerin hesaplanmasıyla enerji verimliliği artırılabilir, akustik performans iyileştirilebilir ve kullanıcı deneyimi optimize edilebilir.</p> <p>• Hızlı İterasyon ve Analiz:</p> <p>Parametrik model, iç mekan tasarımında hızlı iterasyon ve analiz imkanı sunar. Tasarım parametreleri kolayca değiştirilebilir ve bunun sonucunda farklı tasarım seçenekleri hızlıca oluşturulabilir. Bu, tasarım ekibinin farklı alternatifleri değerlendirmesini, görselleştirmesini ve analiz etmesini sağlar.</p>	<p>• Sınırlı Kullanıcı Görüşü:</p> <p>Parametrik modelin tasarımı, genellikle tasarımın optimize edilmesi veya belirli hedeflere yönelik optimize edilmesiyle ilgilenirken, kullanıcıların fikirleri, tercihleri ve geri bildirimleri bazen yeterince dikkate alınmayabilir.</p> <p>• Teknoloji ve Bilgi Gereksinimi:</p> <p>Parametrik model, genellikle ileri düzeyde bilgisayar destekli tasarım modelleme yazılımlarının kullanılmasını gerektirebilir. Bu, tasarım ekibinin ileri düzeyde teknoloji bilgisine sahip olmasını ve bu yazılımları etkin bir şekilde kullanmasını gerektirebilir. Ayrıca, parametrik modelin tasarımı konusunda uzmanlaşmış kişilerin ekibe dahil edilmesi ve tasarım sürecinin karmaşıklığına uygun bir altyapının oluşturulması gerekebilir.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Algoritmik Yaklaşım	<p>• İzlenebilirlik:</p> <p>Algoritmik tasarım yaklaşımında tasarım süreci adım adım planlanır ve bu adımların izlenebilirliği sağlanır. Tasarımcılar, tasarımın her aşamasını belirli bir sıra ve mantıkla ilerleyerek takip edebilirler. Bu, tasarım sürecinin daha yapılandırılmış ve öngörülebilir olmasını sağlar. İç mimari tasarımda, örneğin bir mekanın planlanması, alanın kullanımına, hareket akışına ve işlevselliğe yönelik adımlarla izlenebilir hale getirilebilir. Örneğin bir ofis tasarımında, algoritmik tasarım yaklaşımıyla tasarımcılar, çalışanların hareket akışını optimize etmek için oturma düzenini, yürüyüş yollarını ve servis alanlarını adım adım planlayabilirler.</p> <p>• Kontrol Edilebilirlik:</p> <p>Algoritmik tasarım, tasarım sürecinin tamamen kontrol edilebilir olmasını sağlar. Tasarımcılar, tasarım kararlarını ve parametreleri istedikleri gibi ayarlayabilirler. Bu, tasarımcıların istedikleri sonuca ulaşmak için farklı seçenekleri test etmelerine ve denemelerine olanak tanır. Örneğin bir ofis iç mekan tasarımında, algoritmik tasarım yaklaşımıyla tasarımcılar, çalışanların işbirliğini artırmak için oturma düzenini ve bölme duvarlarını farklı şekillerde kontrol edebilirler. Bu, farklı senaryoları test ederek en uygun çözümü bulmalarına yardım eder.</p>	<p>• Formülasyonun Zorluğu:</p> <p>Karmaşık iç mekan tasarımı problemlerinde algoritmanın doğru şekilde formüle edilmesi zorlaşabilir. Verilerin doğru şekilde analiz edilmemesi veya hatalı parametrelerin kullanılması sonucunda yanlış sonuçlar üretebilir.</p> <p>Teknoloji ve Bilgi Gereksinimi:</p> <p>Matematiksel algoritmalara ve programlamaya hakim olmayı gerektirir. Tasarım sürecinde algoritmaların optimize edilmesi ve doğru sonuçların elde edilmesi zaman alıcı olabilir. Karmaşık algoritma ve veri kümeleri gerektirebilir. Görsel programlama araçları ile yapılabilmesine rağmen, bazı durumlarda daha gelişmiş programlama bilgisine ihtiyaç duyulabilir.</p> <p>• Yaratıcılığın sınırlanması:</p> <p>Tamamen algoritmik bir süreç, tasarımcının yaratıcılığını ve özgünlüğünü sınırlayabilir. Görsel programlama araçlarının sınırlı yetenekleri, karmaşık tasarımların ifadesini kısıtlayabilir.</p>
----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>• Yeniden Kullanılabilirlik:</p> <p>Açık bilgiler, geçmiş tasarım projelerinden elde edilen bilgilerin yeniden kullanılmasını kolaylaştırır. Tasarımcılar, daha önceki projelerden elde edilen başarıları veya deneyimleri kullanarak yeni projelerde daha etkili tasarımlar yapabilirler. Bu da tasarım sürecinin tekrarlanabilirliğini artırır.</p> <p>• Bilinçli Tasarım:</p> <p>Algoritmik tasarım yaklaşımı, tasarım sürecinde hesaplamalı düşünme bilincinin tamamen açık olduğu bir yaklaşımdır. Dolayısıyla tasarımcılar, tasarım sürecini yüksek bir bilinç düzeyle ele alır. matematiksel ve işlevsel algoritmaları kullanarak tasarım problemlerini çözerler.</p> <p>• Karmaşık Tasarım Problemlerinin Anlaşılabilirliği:</p> <p>Tasarım iterasyonları ve optimizasyonu kolaylaştırır. Karmaşık tasarım problemlerini analiz etme ve optimize edilmiş çözümler üretme yeteneği sağlar.</p>	
<p>Üretken Yaklaşım</p>	<p>• Tasarımın Modülerliği ve Tekrar Kullanılabilirliği</p> <p>Şekil gramerleri, hücresel otomat iç mekan bileşenlerinin modüler bir şekilde tasarlanmasını sağlayarak, bu bileşenlerin farklı projelerde tekrar kullanılabilmesini kolaylaştırır. Bu da tasarım sürecini hızlandırır ve maliyetleri düşürür. Her bir bileşen, bağımsız olarak</p>	<p>• Zaman Alıcılık:</p> <p>Şekil gramerleri, tasarım diline ve projenin gereksinimlerine özgü olarak oluşturulmalıdır. Bu, zaman ve kaynak gerektirebilir. Ayrıca, büyük ve karmaşık projeler için şekil gramerleri oluşturmak ve yönetmek daha zor olabilir.</p>

	<p>değiştirilebilir ve genişletilebilir. Örneğin, bir ofis iç mekanında, modüler hücreler olarak tasarlanmış çalışma alanları veya toplantı odaları kullanılabilir. Bu modüler tasarım, mekanın ihtiyaçlara göre kolayca yeniden yapılandırılmasını sağlar. Fraktallar, tasarımın farklı ölçeklerde uygulanabilmesini ve tekrar edilebilmesini sağlar. Bu, tasarımın genişletilebilirliğini artırır ve farklı iç mekan projelerinde kullanılabilirliğini sağlar.</p> <p>• Çevreyle Etkileşimde Bulunan Adaptif Sistemler:</p> <p>Ajan tabanlı tasarımda ajanlar, iç mekanın çevresiyle etkileşime geçebilir, verileri analiz edebilir ve öğrenebilir. Bu, iç mekanın kullanıcılarının ihtiyaçlarına daha iyi uyum sağlayan adaptif ve öğrenen sistemlerin tasarımını mümkün kılar.</p> <p>• Dağıtılmış Sistemlerde Paralleştirme ve Ölçeklenebilirlik:</p> <p>Ajan tabanlı tasarım, iç mekanın paralel olarak çalışmasını ve büyüklüğüne ve kullanıcı sayısına göre ölçeklenebilir olmasını sağlar. Bu, yüksek performans, hızlı yanıt süreleri ve büyük ölçekli projeler için uygunluk sağlar.</p> <p>• Estetik Açından Çekici ve Sıradışı Tasarımlar:</p> <p>Fraktal geometri ve hücresel otomat gibi modeller doğadaki karmaşık ve tekrarlı desenlerin matematiksel bir modelini sağlar. Bu, iç mekan tasarımında</p>	<p>• Karmaşıklık:</p> <p>Büyük ve karmaşık bir iç mekan tasarımında, şekil gramerlerinin anlaşılması ve yönetimi zorlaşabilir. Bu da tasarım sürecini karmaşıktırabilir ve hatalara neden olabilir.</p> <p>• Tanımlama Problemi:</p> <p>Modellerin hatalı veya eksik tanımlanması, otomatik olarak üretilen kodun hatalı olmasına ve tasarımda sorunlara yol açmasına neden olabilir.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>benzersiz, estetik açıdan çekici ve sıradışı tasarımların oluşturulmasına olanak tanır.</p> <p>• Açık ve Yapılandırılmış bir Tasarım Dili:</p> <p>Şekil gramerleri, iç mimaride bir tasarım dilinin oluşturulmasına yardımcı olur. Bu dil, tasarım sürecinde yer alan tüm paydaşlar arasında daha iyi iletişim sağlar ve ortak bir anlayış oluşturulmasına katkıda bulunur. Örneğin, belirli semboller veya semboller kombinasyonları, belirli mekan özelliklerini ifade etmek için kullanılabilir. Bu şekilde, tasarımcılar, müşteriler ve diğer paydaşlar arasında bilgi ve fikir alışverişi daha verimli hale gelir.</p>	
<p>Performansa Dayalı Yaklaşım</p>	<p>• Enerji Verimliliği:</p> <p>Optimal enerji verimliliği: Performansa dayalı hesaplamalı tasarım yaklaşımı, iç mekanların enerji tüketimini optimize etme imkanı sağlar. Örneğin, bir ofis binasının enerji tüketimini analiz ederek ısıtma, soğutma ve aydınlatma sistemlerini optimize etmek, enerji tasarrufu elde etmeyi sağlar.</p> <p>• İyi Termal Konfor ve İç Hava Kalitesi:</p> <p>Performansa dayalı tasarım, iç mekanların termal konforunu ve iç hava kalitesini iyileştirme potansiyeli sunar. Örneğin, bir konutun termal konforunu optimize etmek için ısıtma, soğutma ve havalandırma sistemleri analiz edilerek ideal sıcaklık ve havalandırma düzenlemeleri yapılabilir.</p>	<p>• Karmaşık Analiz ve Modelleme:</p> <p>Performansa dayalı tasarım yaklaşımı, karmaşık analizler ve modellemeler gerektirebilir. Bu, daha fazla bilgi ve uzmanlık gerektirebilir.</p> <p>• Yüksek Maliyetler:</p> <p>Performansa dayalı tasarımın uygulanması, ek maliyetler gerektirebilir. Özellikle analiz araçlarının kullanımı, uzman danışmanlık hizmetleri veya özel yazılımların gerekliliği maliyetleri artırabilir.</p> <p>• Süreç Karmaşıklığı ve Zaman Alıcılığı</p> <p>Performansa dayalı tasarım süreci, daha karmaşık ve zaman alıcı olabilir. Analizler, optimizasyon</p>

	<p>• Performans Tabanlı Karar Verme ve Optimizasyon:</p> <p>Performansa dayalı tasarım, iç mekan tasarımında performans tabanlı karar verme ve optimizasyonu destekler. Örneğin, bir otel tasarımının enerji tüketimi, ısı yalıtımı, akustik performans, havalandırma vb. faktörler analiz edilerek en iyi tasarım kararları verilebilir.</p> <p>• Akustik Performansın İyileştirilmesi:</p> <p>Bu yaklaşım, iç mekanların akustik performansını optimize etme imkanı sunar. Örneğin, bir sinema salonunun ses yalıtımı, yankı kontrolü ve ses düzenlemesi analiz edilerek optimum ses deneyimi sağlanabilir.</p> <p>• Sürdürülebilirlik Hedeflerine Uygun Tasarım:</p> <p>Performansa dayalı tasarım, iç mekandaki ışıklandırmayı ve doğal aydınlatmayı optimize etme olanağı sağlar. Örneğin, bir restoranın aydınlatma tasarımı, doğal ışık kullanımı ve yapay aydınlatma sistemlerinin analiz edilerek ideal ışıklandırma düzenlemeleri yapılabilir.</p> <p>• İyi Havalandırma ve Konforlu Çalışma Ortamı:</p> <p>Bu yaklaşım, iç mekamlarda iyi havalandırma ve konforlu çalışma ortamlarının sağlanmasına yardımcı olur. Örneğin, bir ofis alanının havalandırma sistemi analiz edilerek hava dağıtımı, hava kalitesi ve konfor faktörleri optimize edilebilir.</p>	<p>süreçleri ve tekrarlayan iterasyonlar tasarım sürecini uzatabilir.</p> <p>• Teknik Bilgi ve Uzmanlık Gerektirmesi:</p> <p>Performansa dayalı tasarımın uygulanması, teknik bilgi ve uzmanlık gerektirebilir. Analiz araçlarının ve yöntemlerinin doğru bir şekilde kullanılması ve yorumlanması için uzmanlığa ihtiyaç vardır.</p> <p>• İnsan Faktörünü Tam Olarak Yansıtamaması:</p> <p>Performansa dayalı tasarım, bazen insan faktörünü tam olarak yansıtmayabilir. Örneğin, tasarım kararları sadece performans hedeflerini dikkate alırken, kullanıcıların tercihlerini veya estetik değerleri tam olarak göz önünde bulundurmayabilir.</p> <p>• Farklı Kullanıcı Gereksinimleri:</p> <p>Performansa dayalı tasarım, farklı kullanıcı gereksinimlerini tam olarak karşılamayabilir. İnsanların konfor tercihleri ve ihtiyaçları, tek bir performans kriteriyle sınırlı olmayabilir.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>• Işıklandırma ve Doğal Aydınlatmanın Optimize Edilmesi:</p> <p>Performansa dayalı tasarım, iç mekanların sürdürülebilirlik hedeflerine uygun olmasını sağlama imkanı sunar. Örneğin, enerji verimliliği, atık yönetimi ve doğal kaynak kullanımı gibi faktörler analiz edilerek çevre dostu tasarımlar yapılabilir.</p> <p>İyi Havalandırma ve Konforlu Çalışma Ortamı:</p> <p>Bu yaklaşım, iç mekanlarda iyi havalandırma ve konforlu çalışma ortamlarının sağlanmasına yardımcı olur. Örneğin, bir ofis alanının havalandırma sistemi analiz edilerek hava dağıtımı, hava kalitesi ve konfor faktörleri optimize edilebilir.</p>	
<p>Optimizasyona Dayalı Yaklaşım</p>	<p>• Optimize Edilmiş Alan Kullanımı:</p> <p>Optimizasyon algoritmaları, iç mekan tasarımında mevcut alanın en etkili şekilde kullanılmasını sağlayabilir. Bu, kullanılabilir alanın maksimize edilmesini ve gereksiz boşlukların önlenmesini sağlar.</p> <p>• Enerji Verimliliği:</p> <p>Optimizasyon algoritmaları, iç mekan tasarımında enerji verimliliğini artırmaya yardımcı olabilir. Işıklandırma, ısıtma-soğutma sistemleri ve malzeme seçimi gibi faktörler optimize edilerek enerji tasarrufu sağlanabilir.</p>	<p>• Karmaşık Algoritmalar:</p> <p>Optimizasyon algoritmaları genellikle karmaşık matematiksel hesaplamalar gerektirebilir. Bu nedenle, bu yaklaşımın uygulanması zaman ve kaynak gerektirebilir.</p> <p>• Tasarımcı Yaratıcılığının Sınırlamaları</p> <p>Optimizasyon odaklı tasarımlar, bazen tasarımcının yaratıcılığını veya estetik tercihlerini kısıtlayabilir. Bu, tasarımın kişisel ve özgün bir ifadeye sahip olma potansiyelini azaltabilir.</p>

	<p>• Proaktif ve Öngörülü Tasarım Yaklaşımı:</p> <p>Optimizasyon, iç mimari tasarımda proaktif ve öngörülü bir yaklaşımı teşvik eder. Örneğin, bir ofis tasarımında, çalışanların hareketliliği göz önünde bulundurularak odaların ve koridorların optimize edilmesiyle verimli bir çalışma ortamı oluşturulabilir.</p> <p>• Hızlı ve Doğru Karar Verme Süreci:</p> <p>Optimizasyon algoritmaları, iç mimari tasarımda hızlı ve doğru karar verme sürecini destekler. Örneğin, bir restoran tasarımında, oturma düzeninin optimize edilmesiyle müşteri akışı ve hizmet hızı iyileştirilebilir.</p> <p>• Performans ve Kalite İyileştirmesi:</p> <p>Optimizasyon, iç mekan tasarımında performans ve kaliteyi artırmak için kullanılabilir. Örneğin, bir spor salonu tasarımında, akustik performansın optimize edilmesiyle gürültü kontrolü sağlanabilir ve sporcuların performansı desteklenebilir.</p> <p>• Sürdürülebilirlik Hedeflerinin Desteklenmesi:</p> <p>Optimizasyon, iç mimari tasarımda sürdürülebilirlik hedeflerine ulaşmada yardımcı olabilir. Örneğin, bir ofis binası tasarımında enerji tüketimi, atık yönetimi ve malzeme seçimi gibi faktörler optimize edilerek çevresel etkiler azaltılabilir.</p>	<p>• Optimizasyon Sonuçlarının Gerçek Dünyaya Uyarlanmasının Zorluğu:</p> <p>Elde edilen optimizasyon sonuçlarının gerçek dünyadaki mekanlara uyarlanması ve uygulanması bazen zorluklar yaratabilir.</p> <p>• Yüksek Maliyet:</p> <p>Optimizasyon problemleri, yüksek hesaplama gücü gerektirebilir: Enerji verimliliği optimizasyonu karmaşık hesaplamalar gerektirebilir ve bu da yüksek hesaplama gücü ihtiyacını beraberinde getirebilir. Karmaşık algoritmaların gerektirdiği zaman ve kaynak kullanımı, tasarım sürecinin uzamasına ve maliyetlerin artmasına neden olabilir.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Doğadan İlham Alan Yaklaşım</p>	<p>• Esneklik ve Dönüştürülebilirlik:</p> <p>Biyomimetik, doğadaki organizmaların esnekliğini ve dönüştürülebilirliği taklit edilerek iç mekan tasarımlarında esneklik ve dönüştürülebilirlik sağlanabilir. Doğadaki optimizasyon prensiplerini kullanarak iç mekanlarda optimize edilmiş yapılar oluşturabilir.</p> <p>• Enerji Verimliliği:</p> <p>Doğadaki enerji verimli süreçlerden ilham alarak iç mekan tasarımlarında enerji tasarrufunu teşvik edilebilir. Yalıtım, güneş enerjisi kullanımı, pasif soğutma ve ısıtma gibi doğal stratejilerle iç mekanlarda enerji tüketimini azaltabilir. Örneğin, termit yuvalarından ilham alarak pasif soğutma ve ısıtma sistemleri geliştirilebilir. Karıncaların sıcaklık ve nem düzenlemesi gibi prensipler kullanılarak iç mekan iklimi optimize edilebilir.</p> <p>• Adaptasyon ve İyileştirme:</p> <p>Doğadaki adaptasyon ve iyileştirme mekanizmaları iç mekanlarda uygulayarak kullanıcı deneyimini geliştirebilir. Doğadaki işlevsel ve akıllı sistemlerin taklit edilmesiyle iç mekanlarda işlevsel ve akıllı çözümler geliştirilebilir.</p> <p>• İnsan Sağlığı ve Refah:</p> <p>Biyomimetik, insan sağlığı ve refahını iyileştiren iç mekan tasarımlarının geliştirilmesine yardımcı olabilir. Doğadan ilham alan tasarımlar, iç mekan tasarımlarında sürdürülebilirlik prensiplerini teşvik edebilir. Geri</p>	<p>• Teknik Zorluklar:</p> <p>Biyomimetik tasarım, doğal süreçleri taklit etmeyi gerektirebilir ve bu da teknik zorluklar ve karmaşıklıklar ortaya çıkarabilir. Örneğin, doğal hareket mekanizmalarının yapay olarak oluşturulması teknik beceri ve bilgi gerektirebilir.</p> <p>• Maliyet ve Bakım:</p> <p>Doğal malzemelerin ve sistemlerin maliyeti ve bakım gereksinimleri biyomimetik tasarımları maliyetli hale getirebilir. Örneğin, bazı biyolojik malzemelerin maliyeti yüksek olabilir ve bakımı zor olabilir. Morfogenetik tasarım, karmaşık hesaplamalar ve simülasyonlar gerektirebilir. Tasarım süreci, detaylı analizler ve modelleme aşamaları içerebilir, bu da tasarım sürecini uzatabilir.</p> <p>• Sınırlı Uygulanabilirlik:</p> <p>Biyomimetik tasarım, doğadaki özelliklerin iç mekanlara doğrudan uygulanmasının zor olabileceği durumları içerir. Örneğin, birçok biyolojik özellik, iç mekanlarda direkt olarak uygulanması mümkün olmayabilir.</p> <p>• Doğal Kaynak Kısıtlamaları:</p> <p>Biyomimetik tasarımın uygulanmasında doğal kaynakların sınırlı olması</p>
-------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>dönüştürülebilir malzemelerin kullanımı, su tasarrufu, yenilenebilir enerji kaynakları ve çevresel etkileri azaltan stratejiler gibi uygulamalarla sürdürülebilir iç mekanlar sağlar. Biyomimesis yaklaşımı, doğadaki kaynak tasarrufu stratejilerini iç mekan tasarımlarında kullanarak sürdürülebilirlik hedeflerine katkı sağlayabilir.</p>	<p>tasarımda kısıtlamalara neden olabilir. Örneğin, nadir bulunan bir bitkinin özelliklerinin iç mekân tasarımında kullanılması sınırlı olabilir.</p>
<p>Veriden Öğrenmeye Dayalı Yaklaşım</p>	<p>• Otomatik Öneriler: Makine öğrenimi, tasarım sürecinde otomatik olarak önerilerde bulunabilir. Örneğin, iç mekan tasarımında renk paleti, mobilya seçimi veya düzenleme gibi alanlarda öneriler sunabilir.</p> <p>• Kişiselleştirilmiş Tasarım Makine öğrenimi, kullanıcı tercihlerini analiz ederek kişiselleştirilmiş iç mekan tasarımları sunabilir. Kullanıcıların zevklerini, tercihlerini ve ihtiyaçlarını göz önünde</p>	<p>• Veri Eksikliği: Makine öğrenimi, iç mekan tasarımında kullanılacak yeterli miktarda uygun veriye ihtiyaç duyar. Bu veri eksikliği, doğru ve güvenilir sonuçlar elde etmeyi zorlaştırabilir.</p> <p>• Öznellik ve Estetik Kaybı: Makine öğrenimi, tasarım sürecinde bazı öznellikleri ve estetik değerleri göz ardı edebilir. İnsanların sanatsal</p>

	<p>bulundurarak daha özelleştirilmiş bir deneyim sağlayabilir. Mekanın işlevselliğini artırmak için mobilya yerleşimi, depolama alanları ve dolaşım yolları gibi alanları optimize edebilir.</p> <p>• İyileştirilmiş Enerji Verimliliği:</p> <p>Makine öğrenimi, enerji tüketimini optimize etmek için iç mekan tasarımlarını analiz edebilir. Aydınlatma, ısıtma ve soğutma sistemleri gibi alanlarda enerji verimliliğini artıracak öneriler sunabilir.</p> <p>• Hızlı ve Verimli Tasarım</p> <p>Makine öğrenimi, tasarım sürecini hızlandırabilir ve verimliliği artırabilir. İnsan tasarımcıların manuel olarak yapması gereken birçok tekrarlayan görevi otomatikleştirerek daha hızlı sonuçlar elde edilebilir. Veri analizi ve örüntü tanıma sayesinde, benzersiz tasarım fikirleri ve kombinasyonlar keşfedilebilir, tasarım süreci hızlandırılabilir ve çeşitlilik sağlanabilir.</p> <p>• Hata Azaltma:</p> <p>Makine öğrenimi, iç mekan tasarımında hataları azaltabilir. Örneğin, ölçümler ve matematiksel hesaplamalar gibi hassas işlemlerde insan hatalarını en aza indirgeyebilir.</p> <p>• İnsanlarla İşbirliği</p> <p>Makine öğrenimi, tasarım sürecinde insanlarla işbirliğini destekleyebilir. İnsan tasarımcılarla birlikte çalışarak onlara yardımcı olabilir, tasarım</p>	<p>anlayışını tam olarak yansıtmakta zorlanabilir ve özgün tasarımların yerine daha benzer ve genel tasarımlar sunabilir.</p> <p>• Ölçekleme ve Maliyet:</p> <p>Makine öğrenimi modellerinin ölçeklendirilmesi ve gereken altyapı ve kaynaklar için maliyetler önemli bir dezavantaj olabilir. Büyük veri setleri, yüksek işlem gücü ve veri depolama gereksinimleri, projelerin maliyetini artırabilir.</p> <p>• Güvenlik ve Gizlilik:</p> <p>Makine öğrenimi, iç mekan tasarımında kullanılan verilerin gizliliği ve güvenliği konularında endişelere yol açabilir. Özellikle kişisel tercihler ve kullanıcı verileri gibi hassas bilgilerin korunması önemlidir.</p> <p>• İnsan Dokunuşu Eksikliği:</p> <p>Makine öğrenimi, tasarımda insan dokunuşunu ve yaratıcılığı tam olarak yerine getiremez. İnsan tasarımcıların estetik, deneyim ve duygusal zeka gibi alanlarda sağladığı katkıyı tam anlamıyla yerine koymak zor olabilir.</p> <p>• Eğitim ve Yetenek Gereksinimi:</p> <p>Makine öğrenimi, iç mekan tasarımında teknik zorluklar ortaya çıkarabilir. Veri</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	sürecini geliştirebilir ve yeni fikirler sunabilir.	<p>toplama ve işleme, algoritma geliştirme ve model optimizasyonu gibi teknik becerileri gerektirebilir.</p> <p>• Veri Yanlılığı: Makine öğrenimi, iç mekan tasarımında kullanılan verilerde yanlılık oluşabilir. Eğitim verileri seçimindeki eksiklikler veya önyargılar, makine öğrenimi modellerinin objektif sonuçlar üretmesini engelleyebilir.</p>
--	-----------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SONUÇ ve ÖNERİLER

Hesaplamanın günümüze kadarki evrimi hedeflerimizi ve problemlerimizi çözmeye alışkanlıklarımızı değiştirdiğini göstermektedir. Hesaplama düşünme, birçok disiplini etkiledi ve hesaplama teoriye bağlı olarak geliştirdi. Yardım edeceği disipline ve ihtiyaç duyan kişiye göre problemle eşleşen hesaplama modelleri kullanıldı. 1936'da Alan Turing tarafından icat edilen Turing modeli, bilgisayar bilimi için bir devrimdi. Turing insanların makul olarak hesaplanabilen her şeyi hesaplayabiliyordu ancak aynı zamanda günlük hesaplamaları kolayca temsil edemeyecek kadar ilkel. Günümüzde artık her disiplin kendi hesaplama makinelerini, modellerini tasarlayabiliyor. Hesaplama ve dolayısıyla hesaplama düşünme tasarım sürecini de derinden etkiliyor. Şimdi her alanda, hesaplama tasarımcılar bir modeli programlar, yeni bir model tasarlar ya da her ikisini birden yapar. Hesaplama mimari tasarımcı, çözümleri bulmak için kendi aracını yaratır. Bunun için, eski hesaplama düşünürler gibi programlama yapmak zorunda değildir. Artık hesaplama tasarımcıların (düşünürler) birçok imkânı vardır.

Hesaplama tasarlarkenki bilişsel faaliyetlerimiz, hesaplama tasarımının anlamının anlaşılması için değerlidir. Hesaplama düşünme, birçok düşünme biçiminden beslenir: mantıksal, sistemli, analitik, eleştirel düşünme biçimleri ve tasarım düşüncesi. Hesaplama düşünme, bu düşünme biçimleriyle benzer özellikler gösterse de daha fazlasını ifade eder. Diğer düşünme süreçlerinde olmayan bileşenler ve yardım ettiği disipline göre değişen hesaplama model, hesaplama düşünmeyi farklı kılar. Diğer yandan hesaplama düşünme, sürece odaklanır. Düşünce süreci içerisinde hedefler uygun bir modele göre formüle edilir. Uygun bir model elde etmek için hesaplama bileşenleri ile meşgul oluruz. Bu anlamda hesaplama düşünme, sadece süreci kontrol eden adımları değil, modeli tasarlamayı da içerir.

Tüm görüşler ve ortak beyanlar çerçevesinde bir tanım yapılırsa; hesaplama düşünme, bir bireyin endişelerini gidermek için hedeflerini bir bilgisayarın, makinenin ya da bir insanın gerçekleştirebileceği şekilde, hesaplama adımları veya algoritma halinde ve bir hesaplama modeline göre formüle etmesidir. Bu düşünme sürecinde soyutlama, ayrıştırma, algoritmik düşünme, değerlendirme, genelleme gibi ana

bileşenlerin yanında problemin ya da hedefin kapsamına göre hata ayıklama, örüntü tanıma, yineleme, özyineleme ve daha fazlasını kullanır. Bu bileşenleri kullanmak için modellemeden, simülasyondan, görselleştirmeden, kodlamadan ve otomasyondan yararlanılabilir. Bu araçlar ve uygulamalar hesaplamalı düşünmeyi kolaylaştırır ve hesaplamalı düşünebilmek için farkındalık kazandırır ama kullanılmak zorunda değildir.

Gelinen süreçte hesaplamalı düşünmenin de anlamı değişmiştir. Hesaplamalı düşünme, daha ulaşılabilir, daha çok insana hitap eden bir anlama bürünmüştür. Programlama becerisinin yanında daha genel anlamda hesaplama adımlarını planlama ile ilişkilendirilen bu düşünme faaliyeti, hesaplamalı mimari tasarım sürecinin de özünü oluşturur. Hesaplamalı tasarım, hesaplamalı düşünme ve hesaplamalı yapmanın kesiştiği yerdir. Mimari tasarım hedefini gerçekleştirmek için hedefi gerçekleştiren yol (araç ya da süreç) tasarlanır. Bu ise hesaplamalı düşünerek yapılır.

Geçmişten bugüne değişen hesaplamalı tasarlama yöntemleri ve araçları hesaplamalı düşünme süreci açısından birçok boyutta ele alınabilir. Dijital olarak hesaplamalı mimari tasarım süreci programlanabilir ya da analog veya dijital olarak hesaplamalı mimari tasarım adımları planlanabilir. Öte yandan bir hesaplamalı tasarım (araç), bir hesaplamalı tasarım yapmak için ya da sadece tasarım yapmak için kullanılabilir. Diğer yandan bir hesaplamalı tasarım aracının içinde yeni araçlar programlanabilir veya hesaplama adımlarını veya algoritmalarını bir makinenin çalıştırabileceği şekilde programlamak yerine bir makinenin programlamasına izin verilebilir.

Günümüzde hesaplamalı mimari tasarımcılar bir yarı-programcı gibi hareket ederek Grasshopper, GenerativeComponents gibi görsel algoritmaya dayalı programlarda kendi araçlarını üretiyorlar. Performansa yönelik birçok simülasyon aracıyla yapıları analiz ediyor, optimizasyon araçlarıyla optimum sonuçlara ulaşıyorlar. En hızlı en verimli sonuçlara ulaşmak için de makine öğrenimi yaklaşımlarından yararlanabiliyorlar. Artık hiç elde olmadığı kadar veri vardır (Carpo, 2017) ve bu verilerle makinelerin insanların yerine tasarım süreçleriyle ilgilenmesi umulur.

Bu çalışmada, iç mimari tasarım sürecinde hesaplamalı yaklaşımların önemi vurgulanmaktadır. Bu yaklaşımlar altında birçok yöntem ve model bulunmakta ve her birinin avantajlarıyla birlikte tasarım hedefine ulaşmada kullanılabileceği belirtilmektedir. Yaklaşımlar arasında keskin ayrımlar olmasa da kullanılan yöntemlerle birbirlerinden ayrıldıkları ifade edilmektedir. Hesaplamalı yöntemlerin, geleneksel tasarım sürecinde ulaşılamayan karmaşıklık düzeyinde birçok tasarım alternatifini üretebileceği ve tasarımcıların en hızlı ve verimli modeli bulmaya çalıştığı belirtilmektedir. Çalışmada, hesaplamalı yaklaşımların fikirler üzerine odaklanmak gerektiği vurgulanmış ve her yöntemin avantajları olduğu ancak birden çok yöntemin bir tasarım hedefini gerçekleştirebileceği ifade edilmiştir.

İç mimari tasarım sürecinde hesaplamalı yöntemler, geleneksel tasarım sürecine göre daha hızlı, verimli, hassas ve rasyonel bir tasarım sürecini sağlar. Zamandan ve kaynaklardan optimum tasarruf sağlayan sürdürülebilir tasarımlar (aydınlatma, akustik, sirkülasyon, ergonomi, malzeme kullanımı), dekorasyon için yaratıcı desenler, otomatik mekân planlama problemleri, stil uyumlu mobilya seçimi, mekân organizasyonu, mekânsal davranışların incelenmesi, tasarımda rehberlik gibi birçok konuda hesaplamalı tasarım yapılmıştır. Tüm bu olanaklar ile birlikte iç mimari tasarım pratiği odağında hesaplamalı tasarım araştırmaları nispeten daha azdır. Literatürdeki çalışmalarda, çoğunlukla, 'iç mekân tasarımı' anahtar kelime olarak bulunmasa da iç mekân tasarımıyla ilgilendiren hesaplamalı tasarım çalışmaları bulunmaktadır. Ancak hesaplamalı tasarım hedeflerini, her alanda olduğu gibi bir iç mimarın bakış açısıyla da ele almak konuya farklı bir bakış açısı getirecektir. İç mimari tasarım sürecinde hesaplamalı mimari tasarım yöntemlerinin kullanımı daha spesifik alt konular altında incelenebilir. Bu potansiyel incelemeler ve araştırmalar, iç mekân tasarımında hesaplamalı tasarımın daha geniş bir perspektifte kullanımını destekleyebilir.

Bu çalışmada elde edilen çıkarımlar, bundan sonra yapılacak benzer akademik çalışmalarda faydalanılabilir. Bu tezde ortaya konulan bulgular, gelecekteki araştırmacıların, iç mimari tasarım sürecinde hesaplamalı mimari tasarımın etkisini daha da derinlemesine anlamalarına yardımcı olabilir. İleriki çalışmalarda, iç mekân

tasarımına ilişkin yeni perspektifler geliştirilebilir ve bu alanda daha fazla bilgi birikimi sağlanabilir.

Gelecekteki çalışmalarda, yapay zeka tabanlı görsel üretim araçlarının iç mekân tasarımında nasıl kullanılabileceği ve tasarım sürecine nasıl entegre edilebileceği konularının araştırılması önem taşımaktadır. Makine öğrenimi teknikleri ile geliştirilecek yapay zeka araçları, eğitilen görsellere ve stillere göre iç mekân çıktıları üretebilmektedir. Bu konu daha derinlemesine araştırılabilir. Bu eksik noktaların üzerinde durularak, iç mimari tasarım sürecinde makine öğrenimi tekniklerinin ve yapay zeka araçlarının kullanımının etkileri hakkında daha kapsamlı bir anlayışa ulaşılabilir. Bu çalışmalar, iç mekân tasarımında hesaplamalı mimari tasarımın yanı sıra yapay zeka tabanlı araçların da potansiyelini ortaya koyabilir.

Ayrıca, her geçen gün yenilerinin ortaya çıkması muhtemel olan makine öğrenimi teknikleri ile geliştirilebilecek pratik uygulamaların iç mekân tasarımında önemli bir rol oynaması kuvvetle muhtemeldir. Bu uygulamalar genellikle bir yazılım bilgisine sahip olmayı gerektirmez ya da kullanıcıların bilişsel yeteneklerini geliştirmeye teşvik etmez. Ancak iç mekân tasarımcılarının, bu yapay zeka temelli uygulamaların altında yatan ilkeler hakkında fikir sahip olması ve tasarım süreçlerine bu uygulamaları eklemleyebilmesi onların gelecek için daha hazırlıklı olmalarını sağlayabilir. Bu bakış açısı, iç mekan tasarımcılarını multidisipliner bir bakış açısıyla tasarım süreçlerine bakmaya teşvik edebilir.

KAYNAKLAR

- Adriaenssens, S., Block, P., Veenendaal, D. ve Williams, C. (2014). *Shell structures for architecture: form finding and optimization*. New York: Routledge.
- Aghaei-Meibodi, M. (2016). *Generative Design Exploration: Computation and Material Practice*. Stockholm.
- Agkathidis, A. (2015). *Generative design*. United Kingdom: Laurence King Publishing. London: United Kingdom: Laurence King Publishing. <https://www.bloomsburydesignlibrary.com/encyclopedia-chapter?docid=b-9781472596161&tocid=b-9781472596161-BED-G018> adresinden erişildi.
- Ahmar, S. El, Fioravanti, A. ve Hanafi, M. (2013). A methodology for computational architectural design based on biological principles. *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe, 1*, 539–548.
- Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, 55(7), 833–835. doi:10.1093/comjnl/bxs074
- Aish, R. ve Bredella, N. (2017). The evolution of architectural computing: from building modelling to design vomputation. *Architectural Research Quarterly*, 21(1), 65–73. doi:10.1017/S1359135517000185
- Aish, R. ve Woodbury, R. (2005). Multi-level interaction in parametric design. *Lecture Notes in Computer Science* içinde (C. 3638, ss. 151–162). doi:10.1007/11536482_13
- Akar, Z. ve Keskiner, C. (1978). Türk Süsleme Sanatında Desen ve Motifler. İstanbul: Tercüman Gazetesi Yayınları.
- Aksoy, M. (2001). *Varolan tasarım dilleri ve yeni tasarım dilleri bağlamında biçim gramerleri analizi*. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Alpaydın, E. (2020). *Introduction to machine learning (4th edition)*. The MIT Press.
- Amer, A. (2005). *Analytical thinking*. Pathways to Higher Education.
- Anderson, K. (2014). *Design energy simulation for architects: Guide to 3D graphics*. http://www.amazon.com/Design-Energy-Simulation-Architects-Graphics/dp/041584066X/ref=sr_1_1?ie=UTF8&qid=1460530107&sr=8-1&keywords=design+energy+simulation+for+architects adresinden erişildi.
- Anderson, L. W., Krathwohl Peter W Airasian, D. R., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J. ve Wittrock, M. C. (2000). *A taxonomy for learning, teaching, and Assessing: A revision of bloom's taxonomy of*

educational objectives, complete edition. Pearson. <https://www.uky.edu/~rsand1/china2018/texts/Anderson-Krathwohl - A taxonomy for learning teaching and assessing.pdf> adresinden erişildi.

- Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning and Teaching*, 15(3), 226–234. doi:10.1177/1475725716659252
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J. ve Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology and Society*, 19(3), 47–57.
- Anzalone, P. ve Clarke, C. (2003). Architectural applications of complex adaptive systems. *2003 Annual Conference of the Association of Computer Aided Design in Architecture (Acadia)* içinde (ss. 325–335).
- Arden, B. W. (1980). *What can be automated?: The computer science and engineering research study (COSERS)*. MIT Press.
- Asadi, S. ve Geem, Z. W. (2015). Sustainable building design: A review on recent metaheuristic methods. *Recent Advances in Swarm Intelligence and Evolutionary Computation* içinde (C. 585, ss. 203–223). doi:10.1007/978-3-319-13826-8_11
- Attia, S., Hamdy, M., O'Brien, W. ve Carlucci, S. (2013). Computational optimisation for zero energy buildings design: interviews results with twenty eight international experts. *Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association* içinde (ss. 3698–3705).
- Azadi, S. ve Nourian, P. (2021). Collective intelligence in generative design. *BouT RUMOER periodical for the Building Technologist*, No. 76: *Generative Design*, 7–16. doi:10.13140/RG.2.2.15295.84642
- Baños, R., Manzano-Agugliaro, F., Montoya, F. G., Gil, C., Alcayde, A. ve Gómez, J. (2011). Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4), 1753–1766. doi:10.1016/j.rser.2010.12.008
- Barr, V. (2016). Disciplinary thinking, computational doing: Promoting interdisciplinary computing while transforming computer science enrollments. *ACM Inroads*, 7(2), 48–57. doi:10.1145/2891414

- Barr, V. ve Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. doi:10.1145/1929887.1929905
- Başarer, D. (2017). Bir düşünme türü olarak mantıksal düşünme. *the Journal of Academic Social Sciences*, 41(41), 433–442. doi:10.16992/ASOS.12011
- Başarer, D. ve Duman, E. Z. (2018). Perceptions regarding the logical thinking of academics. *Journal of Higher Education and Science*, 8(1), 136. doi:10.5961/jhes.2018.256
- Bazafkan, E. (2018). *Assessment of usability and usefulness of new building performance simulation tools in the architectural design process*. <https://www.researchgate.net/publication/320931455> adresinden erişildi.
- Becker, R. (2008). Fundamentals of performance-based building design. *Building Simulation*, 1(4), 356–371. doi:10.1007/s12273-008-8527-8
- Beecher, K. (2017). Towards computational thinking. *Itnow*, 59(4), 6–7.
- Bentley, T. (1996). *Sharpen your team's skills in creativity*. Australia: McGraw-Hill Education.
- Benyus, J. (1997). *Biomimicry: Innovation inspired by nature*. New York, USA: New York: Morrow.
- Beyaztaş, H. S. (2012). *Mimari tasarım teknolojik bağlamda biçim ve doğa ilişkisi*. İstanbul Teknik Üniversitesi.
- Bidgoli, A. ve Veloso, P. L. A. (2018). DeepCloud: The application of a data-driven, generative model in design. *38th ACADIA Conference: Recalibration: On Imprecision and Infidelity* içinde . Mexico City.
- Blecha, P., Mikulecky, P., Tucnik, P. ve Matyska, J. (2017). Smart home environment - Agent-based models with scenarios implementation support. *2016 Global Information Infrastructure and Networking Symposium, GIIS 2016*. doi:10.1109/GIIS.2016.7814949
- Bohnacker, H., Groß, B., Laub, J. ve Lazzeroni, C. (2009). *Generative Gestaltung*. Mainz: Verlag Hermann Schmidt.
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3), 7280–7287. doi:10.1073/pnas.082080899
- Bovill, C. (1996). *Fractal geometry in architecture and design*. New York: Springer.

- Bozdoğan, A. (2007). *Fen bilgisi öğretiminde çalışma yaprakları ile öğretimin öğrencilerin fen bilgisi tutumuna ve mantıksal düşünme becerilerine etkisi*. Çukurova Üniversitesi.
- Bradner, E., Iorio, F. ve Davis, M. (2014). Parameters tell the design story: Ideation and abstraction in design optimization. *Simulation Series*, 46(7), 172–197.
- Bukhari, F. a. (2011). *A hierarchical evolutionary algorithmic design (HEAD) system for generating and evolving building design models*. Ph.D Thesis. Queensland University of Technology.
- Buna, Z., Badiu, I. ve Eles, A. (2015). On using parametric modeling in furniture design. *Acta Technica Napocensis*, 58(II), 239–244.
- Burkov, A. (2019). *Machine learning*. Amazon. doi:10.1111/j.1468-0394.1988.tb00341.x
- Burry, J. ve Burry, M. (2010). *The new mathematics of architecture*. London: Thames & Hudson.
- Burry, M. (1999). Paramorph: Anti-accident methodologies. *Hypersurface Architecture 2* içinde (Stephen Pe.). New York: John Wiley and Sons Inc.
- Burry, M. (2007). Innovative aspects of the Colònia Güell Chapel project. *Gaudi Unseen: Completing the Sagrada Família* içinde (ss. 59–61). Berlin: Jovis.
- Burry, M. (2011). *Scripting cultures: Architectural design and programming*. Wiley.
- Burry, M. ve Murray, Z. (1997). Architectural design based on parametric variation and associative geometry. *Challenges of the Future 15th eCAADe Conference* içinde (ss. 1–11). http://www.researchgate.net/publication/30868342_Architectural_Design_Based_on_Parametric_Variation_and_Associative_Geometry%0Ahttp://papers.cumincad.org/data/works/att/472d.content.pdf adresinden erişildi.
- Caetano, I. ve Leitão, A. (2019). Architecture meets computation: An overview of the evolution of computational design approaches in architecture. *Architectural Science Review*, 63(2), 165–174. doi:10.1080/00038628.2019.1680524
- Caetano, I., Santos, L. ve Leitão, A. (2020). Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, 9(2), 287–300. doi:10.1016/j.foar.2019.12.008
- Cagan, J., Campbell, M. I., Finger, S. ve Tomiyama, T. (2014). A framework for computational design synthesis: Model and applications. *Journal of Computing and Information Science in Engineering*, 5(3), 171–181. doi:10.1115/1.2013289

- Çağdaş, G. (1994). Fraktal geometri ve bilgisayar destekli mimari tasarımdaki rolü. *CAD+ Bilgisayar Destekli Tasarım ve Ötesi*, 23, 28–31.
- Canestrino, G. (2021). Considerations on optimization as an architectural design tool. *Nexus Network Journal*, 23(4), 919–931. doi:10.1007/s00004-021-00563-y
- Carmo, M. (2017). *The second digital turn: Design beyond intelligence*. Cambridge: MIT Press.
- Carta, S. (2020). Machine learning and computational mathematics. *Communications in Computational Physics*, 28(5), 1639–1670. doi:10.4208/cicp.OA-2020-0185
- Celani, G., Vaz, C. ve Pupo, R. (2013). Sistemas Generativos de Projeto: Classificação e reflexão sob o ponto de vista da representação e dos meios de produção. *Revista Brasileira de Expressão Gráfica*, 1(1), 22–39. <https://rbeg.net/artigos/artigo2.pdf> adresinden erişildi.
- Celani, G. ve Veloso, P. (2015). CAAD conferences: A brief history. *The Next City: Proceedings of the 16th CAAD Futures conference* içinde (ss. 47–58). http://papers.cumincad.org/cgi-bin/works/paper/cf2015_047 adresinden erişildi.
- Chan, C.-W. ve Chiu, M.-L. (2000). A simulation study of urban growth patterns with fractal geometry. *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia* içinde (ss. 55–64).
- Charniak, E. ve McDermott, D. (1985). *Introduction to artificial intelligence*. Addison Wesley Publishing Company.
- Chaszar, A. ve Joyce, S. C. (2016). Generating freedom: Questions of flexibility in digital design and architectural computation. *International Journal of Architectural Computing*, 14(2), 167–181. doi:10.1177/1478077116638945
- Chebiyyam, S. S. (2022). Computational vs parametric vs generative design: Understanding the similarities and differences. *Oneistox*. <https://www.oneistox.com/learning-hub/blog/computational-parametric-generative-design> adresinden erişildi.
- Church, A. (1936). A note on the entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1), 40–41.
- Cichocka, J. M., Browne, W. N. ve Rodriguez, E. (2017). Optimization in the architectural practice. *CAADRIA 2017 - 22nd International Conference on Computer-Aided Architectural Design Research in Asia: Protocols, Flows and Glitches* içinde (ss. 387–396).
- Çinici, Ş. Y. (2012). Computation|Çevirisi ve anlaması kolay olmayan dil, düşünce ve mimarlık. *Dosya*, (29), 12–19. <http://www.mimarlarodasiankara.org/dosya/dosya29.pdf> adresinden erişildi.

- Çıltık, A. (2008). *Sayısal tasarım kavramları ve algoritmik düşüncenin mimari tasarıma etkileri*. Yıldız Teknik Üniversitesi.
- Clayton, M. J. (2005). How i stopped worrying and learned to love AutoCAD. *ACADIA 2005 Conference: Smart Architecture* içinde (ss. 94–103). Savannah, Georgia, USA.
- Clayton, M. J., March, L., Mitchell, B. ve Stiny, G. (2005). Computational design and Autocad: Reading software as oral history. *Sigradi 2005* içinde (s. 5).
- Çolakoğlu, B. (2002). An informal shape grammars for interpolations of traditional Bosnian Hayat Houses in a contemporary context, 15.1-15.9.
- Collins, H. (2010). *Tacit and Explicit Knowledge*. University of Chicago Press.
- Council, N. R. (2010). *Report of a workshop on the scope and nature of computational thinking*. *The National Academies Press*. Washington, D.C.: National Academies Press. doi:10.17226/12840
- Council, N. R. (2011). *Report of a workshop of pedagogical aspects of computational thinking*. *The National Academies Press*. Washington. doi:10.1128/AAC.03728-14
- Crooks, A. T. ve Heppenstall, A. J. J. (2012). Introduction to agent-based modelling. A. J. Heppenstall, A. T. Crooks, L. M. See ve M. Batty (Ed.), *Agent-based models of geographical systems* içinde (ss. 85–105). Dordrecht: Springer Netherlands. doi:10.1007/978-90-481-8927-4_5
- Cross, N. (1977). *The automated architect*. London: Piion Limited.
- Cross, N. ve Maver, T. W. (1973). Computer aids for design participation. *Architectural Design*, 53(5).
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. ve Woollard, J. (2015). *Computational thinking a guide for teachers*. *Computing At School*.
- CSTA ve ISTE. (2011). *Operational definition of computational thinking for K–12 education*. *National Science Foundation*. <http://www.iste.org/docs/pdfs/Operational-Definition-of-Computational-Thinking.pdf> adresinden erişildi.

- da Graça, V. A. C., Kowaltowski, D. C. C. K. ve Petreche, J. R. D. (2007). An evaluation method for school building design at the preliminary phase with optimisation of aspects of environmental comfort for the school system of the State São Paulo in Brazil. *Building and Environment*, 42(2), 984–999. doi:10.1016/j.buildenv.2005.10.020
- Dana, J. (1837). On the Drawing of Figures of Crystals. *American Journal of Science and Arts*, 32, 30–50.
- Darwin, C. (1859). *The Origin of Species*. London: John Murray.
- Davis, D. (2013a). A history of parametric. www.danieldavis.com/a-history-of-parametric adresinden erişildi.
- Davis, D. (2013b). *Modelled on software engineering: flexible parametric models in the practice of architecture*. RMIT University.
- Davis, D., Burry, J. ve Burry, M. (2011). Untangling parametric schemata: Enhancing collaboration through modular programming. *Designing Together: CAAD Futures 2011 - Proceedings of the 14th International conference on Computer Aided Architectural Design* içinde (ss. 55–68).
- Demirel, Ö. (2003). *Planlamadan değerlendirmeye öğretim sanatı*. Ankara: Pegem Yayıncılık.
- Denner, J., Werner, L. ve Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers and Education*, 58(1), 240–249. doi:10.1016/j.compedu.2011.08.006
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13–18. doi:10.1145/1272516.1272529
- Denning, P. J. (2009a). The profession of IT: Beyond computational thinking. *Communications of the ACM*, 52(6), 28–30. doi:10.1145/1516046.1516054
- Denning, P. J. (2009b). The computing field: Structure. *Wiley Encyclopedia of Computer Science and Engineering*, 2008(April 2008). doi:10.1002/9780470050118.ecse962
- Denning, P. J. (2010). Ubiquity symposium “what is computation?” *Ubiquity*, 2010(November), 1–11. doi:10.1145/1880066.1880067
- Denning, P. J. (2017a). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. doi:10.1145/2998438
- Denning, P. J. (2017b). Computational thinking is not necessarily computational. *Communications of the ACM*, 60(9), 8. doi:10.1145/3128899

- Denning, P. J. (2017c). Computational design. *Ubiquity*, 2017(August), 1–9. doi:10.1145/3132087
- Denning, P. J. ve Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(1), 361–390. doi:10.15388/infedu.2021.21
- Denning, P. J., Tedre, M. ve Yongpradit, P. (2017). Misconceptions about computer science. *Communications of the ACM*, 60(3), 31–33. doi:10.1145/3041047
- Dey, L. (1993). *Qualitative data analysis: A user-friendly guide for social scientists*. New York, USA: Routledge.
- Dijkstra, E. W. (1974). Programming as a discipline of mathematical nature. *The American Mathematical Monthly*, 81(6), 608–612. doi:10.1080/00029890.1974.11993624
- Dijkstra, Edsger W. (1967). The structure of the “The” - Multiprogramming system. *Proceedings of the 1st ACM Symposium on Operating Systems Principles, SOSp 1967*, 7.1-7.5. doi:10.1145/800001.811672
- Dino, İ. G. (2012). Creative design exploration by parametric generative systems In architecture. *METU Journal of the Faculty of Architecture*, 29(1), 207–224. doi:10.4305/METU.JFA.2012.1.12
- Driscoll, J. C. (2019). *Fractals as basis for design and critique. ProQuest Dissertations and Theses*. <http://search.proquest.com.ezaccess.library.uitm.edu.my/dissertations-theses/fractals-as-basis-design-critique/docview/2306306963/se-2?accountid=42518> adresinden erişildi.
- Durand, J. (1813) *Marche à suivre dans la composition d’un projet quelconque in Le Précis d’architecture*, 1813.
- Ediz, Ö. (2003). *Mimari tasarımda fraktal kurguya dayalı üretken bir yaklaşım*. İstanbul Teknik Üniversitesi.
- Ei-Khaldi, M. (2007). *Mapping boundaries of generative systems for design synthesis*. Massachusetts Institute of Technology.
- Eiben, A. E. ve Smith, J. E. (2015). *Introduction to evolutionary computing second edition*. New York: Springer Heidelberg. papers2://publication/uuid/F189C4FB-EB51-43FE-B6FC-17BAF1BF36C8 adresinden erişildi.
- Eilouti, B. H. ve Hamamieh Al Shaar, M. J. (2012). Shape grammars of traditional Damascene houses. *International Journal of Architectural Heritage*, 6(4), 415–435. doi:10.1080/15583058.2011.575530

- Ekici, B., Çubukçuoğlu, C., Turrin, M. ve Sarıyıldız, I. S. (2019). Performative computational architecture using swarm and evolutionary optimisation: A review. *Building and Environment*, 147(October 2018), 356–371. doi:10.1016/j.buildenv.2018.10.023
- El-Zeiny, R. M. A. (2012). Biomimicry as a problem solving methodology in interior architecture. *Procedia - Social and Behavioral Sciences*, 50(July), 502–512. doi:10.1016/j.sbspro.2012.08.054
- Eltaweel, A. ve Su, Y. (2017). Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews*, 73, 1086–1103. doi:10.1016/j.rser.2017.02.011
- Emil Racec, Stefania Budulan, A. V. (2016). Computational intelligence in architectural and interior design: A state-of-the-art and outlook on the field. *Proceedings of the 19th Catalan Conference on Artificial Intelligence* içinde (ss. 108–113).
- Erbaş, S. K. (2015). The impact of computational design approach on computer-aided design education. *Journal of Research in Education and Teaching*, 4(3), 17–23.
- Ercan, B. ve Elias-Ozkan, S. T. (2015). Performance-based parametric design explorations: A method for generating appropriate building components. *Design Studies*, 38, 33–53. doi:10.1016/j.destud.2015.01.001
- Erdoğan, E. ve Gönenç Sorguç, A. (2011). Hesaplamalı modeller aracılığıyla mimari ve doğal biçim türetim ilkelerini ilişkilendirmek. *METU Journal of the Faculty of Architecture*, 28(2), 269–281. doi:10.4305/METU.JFA.2011.2.15
- Estrada, R. D., Wyller, M. ve Dahy, H. (2019). Aerochair: Integrative design methodologies for lightweight carbon fiber furniture design. *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe* içinde (C. 1, ss. 691–700). doi:10.5151/proceedings-ecaadesigradi2019_665
- Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews*, 22, 230–245. doi:10.1016/j.rser.2013.02.004
- Farzaneh, A. (2017). *Computational Morphogenesis of Architectural Objects*. The Open University.
- Feldhusen, J. (1995). Creativity: A knowledge base, metacognitive skills, and personality factors. *The Journal of Creative Behavior*, 29(4), 255–268. doi:10.1002/j.2162-6057.1995.tb01399.x
- Felek, S. Ö. (2022). Parametric modelling in furniture design a case study : Two door wardrobe. *The European Journal of Research and Development*, 2(2), 62–74.

- Fischer, T. ve Herr, C. (2001). Teaching generative design. *Proceedings of the 4th International Conference on Generative Art* içinde (ss. 1–14). Milan. http://generativedesign.eu/on/cic/ga2001_PDF/fischer.pdf adresinden erişildi.
- Ford, R. C. (2013). *Think like ants, not like gods: A study of cellular automata and its validity within the architectural design process*. Unitec Institute of Technology.
- Forsythe, G. E. (1968). *What to do till the computer scientist comes*. *The American Mathematical Monthly* (C. 75). doi:10.2307/2314698
- Foster, D. (2019). *Generative deep learning: Teaching machines to paint, write, compose and play*. O'Reilly.
- Fouad, A. A. ve El Antably, A. (2022). Agent-based modeling and simulation of pandemic propagation in a school environment. *International Journal of Architectural Computing*, 0(0), 1–16. doi:10.1177/14780771221089886
- Frazer, J. (1995). *An evolutionary architecture*. London: Architectural Association.
- Futrell, B. J., Ozelkan, E. C. ve Brentrup, D. (2015). Bi-objective optimization of building enclosure design for thermal and lighting performance. *Building and Environment*, 92, 591–602. doi:10.1016/j.buildenv.2015.03.039
- Gane, V. (2004). *Parametric design - A paradigm shift?* MIT, Cambridge.
- Gane, V. ve Haymaker, J. (2007). Conceptual design of high-rises with parametric methods. *Predicting the Future, 25th eCAADe Conference Proceedings* içinde (ss. 970–978).
- Gaudillière-Jami, N. (2020). Mirroring the development of the computational field in architecture 1965–2020. *AD Magazine*, 150–159.
- Genesereth, M. R. ve Ketchpel, S. P. (1994). Software Agents. *Communications of the ACM*, 37(7), 48–53.
- Gero, J. S. (1975). Architectural optimization a review. *Engineering Optimization*, 1(3), 189–199. doi:10.1080/03052157508960586
- Gilbert, N. ve Terna, P. (2000). How to build and use agent-based models in social science. *Mind & Society*, 1(1), 57–72. doi:10.1007/bf02512229
- Glymph, J. (2003). Evolution of the digital design process. *Architecture in the digital age design and manufacturing* içinde (ss. 149–176). New York: Spon Press.

- Gödel, K. (1934). *On undecidable propositions of formal mathematics*. Princeton: Institute for Advanced Study.
- Gökmeral, B. (2014). *Sürdürülebilir ve bütünleşik bina tasarım süreçlerinde iç mimarlık*. <http://dx.doi.org/10.1016/j.biochi.2015.03.025><http://dx.doi.org/10.1038/nature10402><http://dx.doi.org/10.1038/nature21059><http://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127><http://dx.doi.org/10.1038/nrmicro2577> adresinden erişildi.
- Goodfellow, I., Bengio, Y. ve Courville, A. (2016). *Deep Learning*. Cambridge: The MIT Pres. www.deeplearningbook.org adresinden erişildi.
- Gorn, S. (1967). The computer and information sciences and the community of disciplines. *Behavioral science*, 12(6), 433–452. doi:10.1002/bs.3830120602
- Goyal, K. (2019). Deep learning vs neural networks: difference between deep learning and neural networks. *upGrad*. <https://www.upgrad.com/blog/deep-learning-vs-neural-networks-difference-between-deep-learning-and-neural-networks/> adresinden erişildi.
- Graziosi, S., Scurati, G. W., Parmose, R., Lecchi, A., Bordegoni, M. ve Ferrise, F. (2021). Bioinspired computational design: A case study on a 3D-printed lamp based on the physalis alkekengi. *Proceedings of the Design Society*, 1(August), 561–570. doi:10.1017/pds.2021.56
- Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 122–128. doi:10.1109/TSMC.1986.289288
- Grossfeld, B. ve Grieve, P. (2021). Deep learning vs. machine learning: What's the difference? *Zendesk Blog*. <https://www.zendesk.com/blog/machine-learning-and-deep-learning/> adresinden erişildi.
- Grover, S. ve Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. doi:10.3102/0013189X12463051
- Grover, S. ve Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education* içinde. Bloomsbury Academic. doi:10.5040/9781350057142.ch-003
- Gürkaynak, İ., Üstel, F. ve Gülgöz, S. (2003). *Eleştirel düşünme*. Eğitim Reformu Girişimi.
- Guzdial, M. (2015). *Learner-centered design of computing education: Research on computing for everyone*. (John M. Carroll, Ed.) *Synthesis Lectures on Human-Centered Informatics*.

- Güzelci, O. Z. (2012). *Amasya Yalıboyu evleri üzerine bir biçim grameri çalışması*. İstanbul Teknik Üniversitesi.
- Hamad, S. H. ve Husein, H. A. (2020). The influence of parametric design tools on Increasing creativity in the furniture design process. *Eurasian Journal of Science and Engineering*, 6(1), 199–211. doi:10.23918/eajse.v6i1p199
- Hamdy, M., Nguyen, A. T. ve Hensen, J. L. M. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, 121, 57–71. doi:10.1016/j.enbuild.2016.03.035
- Harel, I. ve Papert, S. (1991). Situating Constructionism. *Constructionism*, 36(2), 1–16.
- Harvard Business Review Analytic Services. (2018). *The Next Wave of Intelligent Design Automation*. Harvard Business Review Analytic Services. <https://www.autodesk.com/solutions/generative-design/manufacturing%0A> <https://www.autodesk.com/industry/manufacturing/resources/mechanical-engineer/generative-design-white-paper> adresinden erişildi.
- Hastie, T., Tibshirani, R. ve Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.
- Haupt, S. E. (2003). *Practical genetic algorithms* (Second Edi.). Wiley-Interscience.
- Hayes, C. C. (1999). Agents in A Nutshell A Very Brief Introduction. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 127–132. doi:10.1109/69.755621
- He, Y., Cai, Y., Guo, Y.-C., Liu, Z.-N., Zhang, S.-K., Zhang, S.-H., ... Chen, S.-Y. (2020). Style-compatible object recommendation for multi-room indoor scene synthesis. *arXiv preprint arXiv:2003.04187*.
- Heiselberg, P., Brohus, H., Hesselholt, A., Rasmussen, H., Seinre, E. ve Thomas, S. (2009). Application of sensitivity analysis in design of sustainable buildings. *Renewable Energy*, 34(9), 2030–2036. doi:10.1016/j.renene.2009.02.016
- Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads*, 1(2), 4–7. doi:10.1145/1805724.1805725
- Hensel, M., Menges, A. ve Weinstock, M. (2006). *Techniques and technologies in morphogenetic design*. *Architectural Design* (C. 76). Wiley-Academy.
- Hensel, M. U. (2010). Performance-oriented architecture: Towards a biological paradigm for architectural design and the built environment. *FormAkademisk - forskningstidsskrift for design og designdidaktikk*, 3(1). doi:10.7577/formakademisk.138

- Herr, C. M. ve Ford, R. C. (2015). Adapting cellular automata as architectural design tools. *CAADRIA 2015 - 20th International Conference on Computer-Aided Architectural Design Research in Asia: Emerging Experiences in the Past, Present and Future of Digital Architecture* içinde (ss. 169–178).
- Herr, C. M. ve Ford, R. C. (2016). Cellular automata in architectural design: From generic systems to specific design tools. *Automation in Construction*, 72, 39–45. doi:10.1016/j.autcon.2016.07.005
- Herr, C. M. ve Kvan, T. (2007). Adapting cellular automata to support the architectural design process. *Automation in Construction*, 16(1), 61–69. doi:10.1016/j.autcon.2005.10.005
- Hu, C. (2011). Computational thinking. *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education - ITiCSE '11* içinde (s. 223). New York, New York, USA: ACM Press. doi:10.1145/1999747.1999811
- Hudson, R. (2010). *Strategies for Parametric Design in Architecture: An application of practice led research*. <http://opus.bath.ac.uk/20947/> adresinden erişildi.
- Humppi, H. ve Österlund, T. (2016). Algorithm-Aided BIM. *Complexity & Simplicity - Proceedings of the 34th eCAADe Conference* içinde (C. 2, ss. 601–609).
- Ibrahim, O., Kamel, A. ve Khamis, E. (2021). Fractal geometry as a source of innovative formations in interior design. *Journal of Design Sciences and Applied Arts*, 2(2), 67–87. doi:10.21608/jdsaa.2021.42275.1075
- Ilachinski, A. (2001). *Cellular automata: A discrete universe*. Singapore; River Edge: World Scientific Publishing Company.
- Indolia, S., Goswami, A. K., Mishra, S. P. ve Asopa, P. (2018). Conceptual understanding of convolutional neural network- A deep learning approach. *Procedia Computer Science*, 132, 679–688. doi:10.1016/j.procs.2018.05.069
- ISTE. (2016). *Iste Standarts for Students*. iste.org.
- Jabi, W. (2013). *Parametric design for architecture*. London: Laurence King Publishing Ltd.
- Jabi, W., Soe, S., Theobald, P., Aish, R. ve Lannon, S. (2017). Enhancing parametric design through non-manifold topology. *Design Studies*, 52, 96–114. doi:10.1016/j.destud.2017.04.003
- Jason Gerber, D. (2007). *Models for design exploration in architecture*. University of Southern California.
- Jennings, N. R. ve Sycara, K. (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 38(1.1), 7–38.

- Jos, T. (1989). *The model: Antoni Gaudi's Hanging Model and its reconstruction*. Universität Stuttgart, Stuttgart.
- Kalay, Y. (1989). *Modeling objects and environments (principles of computer aided design)*. New York: Wiley-Academy.
- Kalay, Y. E. (1999). Performance-based design. *SFPE Handbook of Fire Protection Engineering* içinde (ss. 1233–1261). New York, NY: Springer New York. doi:10.1007/978-1-4939-2565-0_37
- Kalelioglu, F., Gulbahar, Y. ve Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583.
- Kán, P. ve Kaufmann, H. (2017). Automated interior design using a genetic algorithm. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST, November 8*. doi:10.1145/3139131.3139135
- Karplus, R. (2003). Science teaching and the development of reasoning. *Journal of Research in Science Teaching*, 40(SUPPL.), 169–175. doi:10.1002/tea.10091
- Kasabov, N. K. (1996). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering (Computational Intelligence)*. London, England: The MIT Press.
- Katz, D. (1960). The use of computers in engineering classroom instruction (ss. 522–527).
- Kaznatcheev, A. (2017). Antoni Gaudi and learning algorithms from nature. <https://egtheory.wordpress.com/2016/12/25/antoni-gaudi/amp/> adresinden erişildi.
- Keller, S. (2005). 10 2 83EFE0F0 2 or the Seagram Building. *The Art of Architecture/The Science of Architecture*, 298–309.
- Keough, I. ve Benjamin, D. (2010). Multi-objective optimization in architectural design. *Spring Simulation Multiconference 2010, SpringSim'10*, 1–8. doi:10.1145/1878537.1878736
- Kevin P. Murphy. (2012). *Machine learning a probabilistic perspective*. London, England: The MIT Press.
- Khalili Araghi, S. ve Stouffs, R. (2015). Exploring cellular automata for high density residential building form generation. *Automation in Construction*, 49(PA), 152–162. doi:10.1016/j.autcon.2014.10.007
- Kim, B., Kim, T. ve Kim, J. (2014). Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, 49(4), 437–459. doi:10.2190/EC.49.4.b

- Kılıç, D. ve Sağlam, N. (2009). Öğrencilerin mantıksal düşünme yeteneklerinin bazı değişkenler açısından incelenmesi. *Ege Eğitim Dergisi*, 2(2), 23–38.
- Klemmt, C. ve Sugihara, S. (2020). Architectural design by cellular growth algorithm. *ALIFE 2018 - 2018 Conference on Artificial Life: Beyond AI* içinde (ss. 548–549). doi:10.1162/isal_a_00102
- Klügl, F. ve Bazzan, A. L. C. (2012). Agent-based modeling and simulation. *AI Magazine*, 33(3), 29–40. doi:10.1609/aimag.v33i3.2425
- Knight, T. (1999). *Applications in architectural design and education and practice*.
- Knight, T. W. (1994). *Transformations in design: A formal approach to stylistic change and innovation in the visual arts 1st edition*. Cambridge: Cambridge University Press.
- Knuth, D. (1974). Computer science and its relation to mathematics. *American Mathematical Monthly*, 81, 323–343.
- Knuth, D. (1981). Algorithms in modern mathematics and computer science. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 122 LNCS(January), 82–99. doi:10.1007/3-540-11157-3_26
- Knuth, D. (1985). Algorithmic thinking and mathematical thinking. *Mathematical Association of America*, 92(3), 170–181.
- Kolarevic, B. (1993). *Geometric relations as a framework for design conceptualization*. Harvard University.
- Kolarevic, B. (2003). Computing the performative in architecture. In *Proceedings of the 21th eCAADe Conference: Digital Design. Graz, Austria* içinde (ss. 457–464). http://cumincad.architecturez.net/system/files/pdf/ecaade03_457_104_kolarevic.content.pdf adresinden erişildi.
- Kolarevic, B. (2004). Back to the future: Performative architecture. *International Journal of Architectural Computing*, 2(1), 43–50. doi:10.1260/1478077041220205
- Kolarevic, B. (2005a). Computing the performative. *Performative architecture beyond instrumentality* içinde . Spon Press.
- Kolarevic, B. (2005b). Towards the performative in architecture. *Performative Architecture: Beyond Instrumentality* içinde (ss. 203–213). New York: Spon Press.
- Kolarevic, B. (2005c). Digital morphogenesis. *Architecture in the Digital Age: Design and Manufacturing* içinde (ss. 17–45). New York: Spon Press.

- Kolarevic, B. (2013). Parametric evolution. *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design* içinde (C. 9781118522, ss. 50–59). John Wiley & Sons Ltd. doi:10.1002/9781118653074.ch3
- Koo, B., Hergel, J., Lefebvre, S. ve Mitra, N. J. (2017). Towards zero-waste furniture design. *IEEE Transactions on Visualization and Computer Graphics*, 23(12), 2627–2640. doi:10.1109/TVCG.2016.2633519
- Koopmans, T. C. (1951). An analysis of production as an efficient combination of activities. in *Koopmans, T. C. (Ed.): Activity Analysis of Production and Allocation* içinde (C. 13, ss. 33–97). New York: John Wiley and Sons.
- Korkmaz, Ö., Çakir, R. ve Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. doi:10.1016/j.chb.2017.01.005
- Kotnik, T. (2006). *Algorithmic extension of architecture*. Zurich MAS ETH ARCH/CAAD.
- Kotnik, T. (2010). Digital architectural design as exploration of computable functions. *International Journal of Architectural Computing*, 8(1), 1–16. doi:10.1260/1478-0771.8.1.1
- Koutamanis, A. (2005). A biased history of CAAD the bibliographic version. *Digital Design: The Quest for New Paradigms: 23rd eCAADe Conference Proceedings* içinde (C. June, ss. 629–637). https://cumincad.architexturez.net/system/files/pdf/2005_629.content.pdf adresinden erişildi.
- Koziel, S., Ciaurri, D. E. ve Leifsson, L. (2011). Surrogate-based methods. *Computational optimization, methods and algorithms* içinde (ss. 33–59). doi:10.1007/978-3-642-20859-1_3
- Kramer, O. (2017). *Genetic algorithm essentials*. Studies in Computational Intelligence (C. 679). Cham: Springer International Publishing. doi:10.1007/978-3-319-52156-5
- Krawczyk, R. J. (2002). Experiments in architectural form generation using cellular automata. *Proceedings of the 20th eCAADe Conference* içinde (ss. 552–555).
- Kvochick, T. (2020). Fundamental aspects of pattern recognition in architectural drawing. *The routledge companion to artificial intelligence in architecture* içinde . New York: Routledge.

- L'Heureux, J., Cohen, R., Boisvert, D. ve Sanghera, K. (2012). IT problem solving: An implementation of computational thinking in information technology. *SIGITE'12 - Proceedings of the ACM Special Interest Group for Information Technology Education Conference* içinde (ss. 183–188). doi:10.1145/2380552.2380606
- Lantz, B. (2019). *Expert Techniques for Predictive Modeling, 3rd Edition*. Birmingham Uk: Packt Publishing.
- Lawson, A. E. (1982). Formal reasoning, achievement, and intelligence: An issue of importance. *Science Education*, 66(1), 77–83. doi:10.1002/sce.3730660110
- Lawson, A. E. ve Thompson, L. D. (1988). Formal reasoning ability and misconceptions concerning genetics and natural selection. *Journal of Research in Science Teaching*, 25(9), 733–746. doi:10.1002/tea.3660250904
- Lawson, B. (2005). *How designers think?* Architectural Press.
- Lazaer, D. (2000). *The intelligent curriculum*. USA: Zephyr Press.
- Leach, N. (2009). Digital morphogenesis. *Architectural Design*, 79(1), 32–37. doi:10.1002/ad.806
- Leatherbarrow, D. (2005). Architecture's unscripted performance. *Performative Architecture Beyond Instrumentality* içinde (ss. 7–19). New York: Spon Press.
- Lecun, Y., Bengio, Y. ve Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. doi:10.1038/nature14539
- Lee, I. ve Malyn-Smith, J. (2020). Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *Journal of Science Education and Technology*, 29(1), 9–18. doi:10.1007/s10956-019-09802-x
- Leslie, J. (1821). *Geometrical analysis and geometry of curve lines second edition*. Edinburgh: W. & C. Tait.
- Lindenmayer, A. (1968). Mathematical models for cellular interactions in development I. filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3), 280–299. doi:10.1016/0022-5193(68)90079-9
- Liu, B. ve He, J. (2014). Teaching mode reform and exploration on the university computer basic based on computational thinking training in network environment. *Proceedings of the 9th International Conference on Computer Science and Education, ICCSE 2014* içinde (ss. 59–62). doi:10.1109/ICCSE.2014.6926430
- Lockwood, J. ve Mooney, A. (2017). *Computational thinking in education: Where does it fit? a systematic literary review*.

- Lorenzo-Eiroa, P. ve Sprecher, A. (2013). *Architecture in formation: On the nature of information in digital architecture 1st edition*. Routledge.
- Lye, S. Y. ve Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. doi:10.1016/j.chb.2014.09.012
- Machairas, V., Tsangrassoulis, A. ve Axarli, K. (2014). Algorithms for optimization of building design: A review. *Renewable and Sustainable Energy Reviews*, 31(1364), 101–112. doi:10.1016/j.rser.2013.11.036
- Madrazo, L. (1994). Durand and the science of architecture. *Journal of Architectural Education*, 48(1), 12–24. doi:10.1080/10464883.1994.10734619
- Mahmoud, F. A. M. H. M. M. (2019). Towards innovative formation in interior design. According to the application of Chaos Theory. *Journal of architecture arts and humanistic science*, 12–26. doi:10.12816/mjaf.2019.11805.1126
- Maini, V. ve Sabri, S. (2017). *Machine learning for humans*. Self-Published. https://www.dropbox.com/s/e38nil1dnl7481q/machine_learning.pdf?dl=0 adresinden erişildi.
- Malkawi, A. M. ve Augenbroe, G. (2004). *Advanced building simulation. Advanced Building Simulation*. New York: Spon Press. doi:10.4324/9780203684009
- Malyn-Smith, J., Lee, I. A., Martin, F., Grover, S., Evans, M. A. ve Pillai, S. (2018). Developing a framework for computational thinking from a disciplinary perspective. *Proceedings of International Conference on Computational Thinking Education* içinde (ss. 182–186).
- Manavis, A., Minaoglou, P., Tzetzis, D., Efkolidis, N. ve Kyratsis, P. (2021). Computational design technologies for interior designers: A case study. *IOP Conference Series: Materials Science and Engineering*, 1009(1). doi:10.1088/1757-899X/1009/1/012037
- Mandelbrot, B. (1982). *The fractal geometry of nature*. New York: W.H. Freeman and Company.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. ve Settle, A. (2014). Computational thinking in K-9 education. *ITiCSE-WGR 2014 - Working Group Reports of the 2014 Innovation and Technology in Computer Science Education Conference* içinde (ss. 1–29). doi:10.1145/2713609.2713610
- Martin, L. (1967). Architects' Approach to Architecture. *Journal of the Royal Institute of British Architects*, 74, 191–200.
- McKnight, M. (2017). Generative design: What it is? How is it being used? Why it's a game changer. *KnE Engineering*, 2(2), 176–181. doi:10.18502/keg.v2i2.612

- Menges, A. (2007). Computational morphogenesis: Integral form generation and materialization processes. *3rd Int'l ASCAAD Conference on Em'body'ing Virtual Architecture* içinde (ss. 725–744).
- Menges, A. (2012). Biomimetic design processes in architecture: Morphogenetic and evolutionary computational design. *Bioinspiration and Biomimetics*, 7(1). doi:10.1088/1748-3182/7/1/015003
- Middleton, M. (2021). Deep Learning vs. Machine Learning — What's The Difference? *Flatiron School*. www.flatironschool.com/blog/deep-learning-vs-machine-learning/ adresinden erişildi.
- Mike, K., Ragonis, N., Rosenberg-Kima, R. B. ve Hazzan, O. (2022). Computational thinking in the era of data science. *Communications of the ACM*, 65(8), 33–35. doi:10.1145/3545109
- Mitchell, M. (2009). *Complexity: A guided tour*. New York: Oxford University Press.
- Mitchell, W J. (1986). Formal representations: A foundation for computer-aided architectural design. *Environment and Planning B: Planning and Design*, 13(2), 133–162. doi:10.1068/b130133
- Mitchell, William J. (1977). *Computer-aided architectural design*. New York: Van Nostrand Reinhold.
- Mohri, M., Rostamizadeh, A. ve Talwalkar, A. (2018). *Foundations of machine learning*. *Syria Studies* (second edi., C. 7). London, England: The MIT Press. https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~renyal/Civilwars_12December2010.pdf%0Ahttps://think-asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625 adresinden erişildi.
- Monedero, J. (2000). Parametric design: A review and some experiences. *Automation in Construction*, 9(4), 369–377. doi:10.1016/S0926-5805(99)00020-5
- Moretti, L. (1971). Ricerca matematica in architettura e urbanistica. *Moebius*, IV(1), 30–53.
- Mukkavaara, J. (2021). *Exploration and optimization of building design solutions using computational design*. Luleå University of Technology.
- Mukkavaara, J. ve Sandberg, M. (2020). Architectural design exploration using generative design: framework development and case study of a residential block. *Buildings*, 10(11), 1–17. doi:10.3390/buildings10110201
- Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. New York: MIT Press.
- Nabiyev, V. (2012). *Yapay zeka*. Seçkin Yayıncılık.

- Nardelli, E. (2019). Viewpoint do we really need computational thinking? *Communications of the ACM*, 62(2), 32–35. doi:10.1145/3231587
- Negendahl, K. (2015). Building performance simulation in the early design stage: An introduction to integrated dynamic models. *Automation in Construction*, 54, 39–53. doi:10.1016/j.autcon.2015.03.002
- Nguyen, A. T. ve Reiter, S. (2015). A performance comparison of sensitivity analysis methods for building energy models. *Building Simulation*, 8(6), 651–664. doi:10.1007/s12273-015-0245-4
- Nguyen, A. T., Reiter, S. ve Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113, 1043–1058. doi:10.1016/j.apenergy.2013.08.061
- Nilsson, N. J. (2005). *Introduction to machine learning: An early draft of a proposed textbook*. <http://www.ncbi.nlm.nih.gov/pubmed/21172442> adresinden erişildi.
- Oğuzkan, F. (1974). *Eğitim terimleri sözlüğü*. Ankara: Türk Dil Kurumu Yayınları.
- Öksüz, E. B. (2020). *Tasarımcı bakış açısıyla hesaplamalı düşünmede prosedürel soyutlamanın değerlendirilmesi için bir yöntem*. İstanbul Teknik Üniversitesi.
- Oktan, S. ve Vural, S. (2021). A teaching strategies model experiment for computational design thinking. *TECHNE: Journal of Technology for Architecture & Environment*, 2, 154–158. doi:10.13128/techne-10701
- Østergård, T., Jensen, R. L. ve Maagaard, S. E. (2016). Building simulations supporting decision making in early design - A review. *Renewable and Sustainable Energy Reviews*, 61, 187–201. doi:10.1016/j.rser.2016.03.045
- Ostwald, M. J. ve Vaughan, J. (2016). The fractal dimension of architecture. *The Fractal Dimension of Architecture* içinde (ss. 1–18). doi:10.1007/978-3-319-32426-5
- Otto, F. ve Rasch, B. (1996). *Finding Form: Towards an Architecture of the Minimal*. Stuttgart: Axel Menges.
- Oxman, N. (2010). *Material-based design computation*. Massachusetts Institute of Technology.
- Oxman, R. (2006). Theory and design in the first digital age. *Design Studies*, 27(3), 229–265. doi:10.1016/j.destud.2005.11.002
- Oxman, R. (2008a). Performance-based design: Current practices and research issues. *International Journal of Architectural Computing*, 6(1), 1–17. doi:10.1260/147807708784640090

- Oxman, R. (2008b). Digital architecture as a challenge for design pedagogy: theory, knowledge, models and medium. *Design Studies*, 29(2), 99–120. doi:10.1016/j.destud.2007.12.003
- Oxman, R. (2008c). Performance-based design generation. *Proceedings of the 9th Biennial ASME Conference on Engineering Systems Design and Analysis ESDA08* içinde (ss. 1–5). Haifa.
- Oxman, R. (2009). Performative design: A performance-based model of digital architectural design. *Environment and Planning B: Planning and Design*, 36(6), 1026–1037. doi:10.1068/b34149
- Oxman, R. (2011). Novel concepts in digital design. *Computational Design Methods and Technologies* içinde . Information Science Reference. doi:10.4018/978-1-61350-180-1
- Oxman, R. (2017a). Thinking difference: Theories and models of parametric design thinking. *Design Studies*, 52, 4–39. doi:10.1016/j.destud.2017.06.001
- Oxman, R. (2017b). Parametric design thinking. *Design Studies*, 52, 1–3. doi:10.1016/j.destud.2017.07.001
- Özbek, H. (2004). *Gelenekselden türeyen çağdaş Mardin konut yerleşimi*. Yıldız Teknik Üniversitesi.
- Özülkü, Ö. (2010). *Modern mimarlıkta doğadan etkilenen form ve geleceğe yönelik yaklaşımlar*. Mimar Sinan Güzel Sanatlar Üniversitesi.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Library of Congress Cataloging in Publication Data.
- Papert, S. (1996). An exploration in the space Of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95–123. doi:https://doi.org/ 10.1007/BF00191473
- Pereira, I., Belém, C. ve Leitão, A. (2020). Optimizing exhibition spaces a multi-objective approach, (September), 53–62. doi:10.5151/proceedings-eaadesigradi2019_456
- Peter Bentley. (1999). *Evolutionary design by computes*. London, UK: Morgan Kaufmann Publishers.
- Peters, B. ve Peters, T. (2013). *Inside SmartGeometry expanding the architectural possibilities of computational design*. Wiley; 1st edition.
- Post, E. (1936). Finite combinatory processes. *The Journal of Symbolic Logic*, 1(3), 103–105.

- Prusinkiewicz, P. ve Hanan, J. (1989). Lindermayer Systems, fractals, and plants. *Lecture Notes in Biomathematics*, 120.
- Prusinkiewicz, P. ve Lindenmayer, A. (1996). *The algorithmic beauty of plants* (2nd bs.). New York: Springer.
- Radford, A. D. ve Gero, J. S. (1980). On optimization in computer aided architectural design. *Building and Environment*, 15(2), 73–80. doi:10.1016/0360-1323(80)90011-6
- Raschka, S. (2016). How to explain machine learning to a software engineer. *KD nuggets*. <https://www.kdnuggets.com/2016/05/explain-machine-learning-software-engineer.html> adresinden erişildi.
- Raschka, S. ve Mirjalili, V. (2017). *Python machine learning second edition*. Birmingham: Packt Publishing.
- Reyling, C. G. (2019). *Pursuing A More Habitable Space Module : Using Impression Based Generative Design*. University of Washington.
- Rhino Grasshopper, (2019). Islamic Geometric Pattern (Grasshopper Tutorial) [Video]. [youtube.com/watch?v=SbJRhcgC7_k](https://www.youtube.com/watch?v=SbJRhcgC7_k), Erişim Tarihi: 15.12.2022
- Riley, D. ve Hunt, K. A. (2014). *Computational thinking for the modern problem solver. Computational Thinking for the Modern Problem Solver*. Chapman and Hall/CRC. doi:10.1201/b16688
- Roadrangka, V. (1983). The construction of a group assessment of logical thinking (GALT). *The annual meeting of the National Association for Research in Science Teaching*. [http://www.thaiscience.info/journals/Article/The construction of a group assessment of logical thinking %28galt%29.pdf](http://www.thaiscience.info/journals/Article/The%20construction%20of%20a%20group%20assessment%20of%20logical%20thinking%20galt.pdf) adresinden erişildi.
- Rocha, A. J. M. (2004). *Architecture theory, 1960-1980: Emergence of a computational perspective*. Massachusetts Institute of Technology.
- Rocker, I. (2006). When code matters. *Architectural Design*, 76(4), 16–25. doi:10.1002/ad.289
- Román-González, M. (2015). Computational thinking test: design guidelines and content validation. *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)* içinde (ss. 2436–2444). doi:10.13140/RG.2.1.4203.4329
- Román-González, M., Pérez-González, J. C., Moreno-León, J. ve Robles, G. (2018). Can computational talent be detected? predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction*, 18, 47–58. doi:10.1016/j.ijcci.2018.06.004

- Rossi, M. (2019). Organic reference in design. The shape between invention and imitation. P. Magnaghi-Delfin, G. Mele ve T. Norando (Ed.), *Lecture notes in networks and systems* içinde (ss. 177–185). Switzerland: Springer.
- Roudavski, S. (2009). Towards morphogenesis in architecture. *International Journal of Architectural Computing*, 7(3), 345–374. doi:10.1260/147807709789621266
- Runco, M. A. (1996). Personal creativity: Definition and developmental issues. *New Directions for Child and Adolescent Development*, (72), 3–30. doi:10.1002/cd.23219967203
- Rusk, N. (2015). Deep learning. *Nature Methods*, 13(1), 35. doi:10.1038/nmeth.3707
- Saban, D. (2018). Early Houses designed by Farkas Molnár and Semih Rüstem an examination in accordance with Bauhaus and KURI principles. *MEGARON / Yıldız Technical University, Faculty of Architecture E-Journal*, 13(1), 117–131. doi:10.5505/megaron.2017.10437
- Sabetfard, M. ve Nadimi, H. (2020). Generating Square Kufic patterns using cellular automata. *Nexus Network Journal*, 22(2), 275–290. doi:10.1007/s00004-019-00454-3
- Sahab, M. G., Toropov, V. V. ve Gandomi, A. H. (2013). *A review on traditional and modern structural optimization: problems and techniques. Metaheuristic Applications in Structures and Infrastructures* (First Edit.). Elsevier Inc. doi:10.1016/B978-0-12-398364-0.00002-4
- Samuel, A. L. (1959). Some studies in machine learning. *IBM Journal of Research and Development*, 3(3), 210–229. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5392560> adresinden erişildi.
- Şapcı, B. ve Pektaş, Ş. T. (2021). Integrating user experience knowledge into early architectural design processes through machine learning. *JCoDe: Journal of Computational Design*, 2(1), 67–94.
- Sanal 1: <https://www.zaha-hadid.com/archive> (Erişim tarihi: 24.02.2022).
- Sarıyıldız, S. (1991). *Conceptual design by means of Islamic-geometric-patterns within a CAAD environment*. İstanbul: Yıldız Teknik Üniversitesi.
- Sarkar, P. (2000). A brief history of cellular automata. *ACM Computing Surveys*, 32(1), 80–107. doi:10.1145/349194.349202
- Saygıner, Ş. (2017). *Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin erişimi, mantıksal düşünme ve motivasyona etkileri*. Hacettepe Üniversitesi.
- Scheutz, M. ve Mayer, T. (2016). Combining agent-based modeling with big data methods to support architectural and urban design. *Understanding complex urban systems* içinde (ss. 15–31). doi:10.1007/978-3-319-30178-5_2

- Schumacher, P. (2008). Parametricism as style - Parametricist manifesto. *Darkside Club, 11th Venice Architecture Biennale*, 1–5. [http://www.patrikschumacher.com/Texts/Parametricism as Style.htm](http://www.patrikschumacher.com/Texts/Parametricism%20as%20Style.htm) adresinden erişildi.
- Sebestyen, A. (2021). Machine learning for architects and designers: Implementing machine learning into the digital design process. *School of Architecture Symposia* içinde (ss. 295–306).
- Selby, C. (2013). Computational thinking: The developing definition. *ITiCSE Conference 2013* içinde (ss. 5–8).
- Selby, C. C. ve Woollard, J. (2014). Refining an understanding of computational thinking. *Author's original*, (2006), 1–23.
- Senge, P. (1996). *The fifth discipline: The art and practice of the learning organization: second edition*. New York: Random House Business.
- Shi, X. (2010). Performance-based and performance-driven architectural design and optimization. *Frontiers of Architecture and Civil Engineering in China*, 4(4), 512–518. doi:10.1007/s11709-010-0090-6
- Shi, X., Tian, Z., Chen, W., Si, B. ve Jin, X. (2016). A review on building energy efficient design optimization from the perspective of architects. *Renewable and Sustainable Energy Reviews*, 65, 872–884. doi:10.1016/j.rser.2016.07.050
- Shute, V. J., Sun, C. ve Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. doi:10.1016/j.edurev.2017.09.003
- Silver, M. (2006). Building without drawings: Automason Ver 1.0. *Architectural Design*, 76(4), 46–51. doi:10.1002/ad.293
- Singh, V. ve Gu, N. (2012). Towards an integrated generative design framework. *Design Studies*, 33(2), 185–207. doi:10.1016/j.destud.2011.06.001
- Soleimani, A. (2019). Computational design thinking and thinking design computing. *Reynolds Symposium: Education by Design* içinde (ss. 18–19). Portland: PubPub. doi:10.21428/f7d9ca02.b7daadcc
- Soltani, S., Guimaraes, G. D., Liao, P., Calixto, V., Gu, N. ve Creative, U. (2020). Computational design sustainability: A conceptual framework for built environment research. *eCAADe 38, 1*, 219–228.
- Sorguç, A. G. ve Selçuk, S. A. (2013a). Computational models in srchitecture: understanding multi-dimensionality and mapping. *Nexus Network Journal*, 15(2), 349–362. doi:10.1007/s00004-013-0150-z

- Sorguç, A. G. ve Selçuk, S. A. (2013b). Computational models in Architecture: Understanding multi-dimensionality and mapping. *Nexus Network Journal*, 15(2), 349–362. doi:10.1007/s00004-013-0150-z
- Stave, K. ve Hopper, M. (2007). What constitutes systems thinking? A proposed taxonomy. *Proceedings of the 25th International Conference of the System Dynamics Society. Boston, MA, July 29-August 3, 2007* içinde (ss. 1–24). ftp://www.clexchange.org/conference/CLE_2010/CO2010-06Session5MeasureST.pdf%5Cnhttp://www.systemdynamics.org/conferences/2007/proceed/index.htm adresinden erişildi.
- Sternberg, R. J. (2002). Raising the achievement of all students: Teaching for successful intelligence. *Educational Psychology Review*, 14(4), 383–393. doi:10.1023/A:1020601027773
- Sternberg, R. J. (2006). The rainbow project: Enhancing the SAT through assessments of analytical, practical, and creative skills. *Intelligence*, 34(4), 321–350. doi:10.1016/j.intell.2006.01.002
- Steve Furber. (2012). *Shut down or restart?* London. <https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-summary.pdf%0Ahttp://dx.doi.org.ezproxy.elib10.ub.unimaas.nl/10.1111/bjet.12453>. adresinden erişildi.
- Stiny, G ve Mitchell, W. J. (1980). The grammar of paradise: on the generation of Mughul gardens. *Environment and Planning B: Planning and Design*, 7(2), 209–226. doi:10.1068/b070209
- Stiny, George. ve Gips, J. (1978). *Algorithmic aesthetics: Computer models for criticism and design in the arts*. University of California Press. Los Angeles: University of California Press.
- Stiny, George. (1977). Ice-Ray: A note on the generation of Chinese lattice Designs. *Environment and Planning B: Planning and Design*, 4(1), 89–98. doi:10.1068/b040089
- Stiny, George. (1980). Introduction to shape and shape grammars. *Environment and Planning B*, 7(November), 343–351.
- Stiny, George ve Gips, J. (1972). Shape grammars and the generative specification of painting and sculpture. *CV Freiman (ed.) Information Processing 71* içinde (ss. 1460–1465). Amsterdam: North-Holland.
- Stinyf, G. (1984). Spatial systems in architecture and design: some history and logic. *Environment and Planning B: Planning and Design*, 12, 31–53.
- Stouffs, R. ve Rafiq, Y. (2015). Generative and evolutionary design exploration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 29(4), 329–331. doi:10.1017/S0890060415000360

- Sutherland, I. E. (1963). *Sketchpad, a man-machine graphical communication system*. Lincoln Laboratory.
- Sydorenko, I. (2021). What is a dataset in machine learning: Sources, features, analysis. *Labelyourdata*. <https://labelyourdata.com/articles/what-is-dataset-in-machine-learning> adresinden erişildi.
- Symeonidou, I. (2019). Epidermis: Algorithmic design based on biomimetic morphology. *Nexus Network Journal*, 21(1), 161–174. doi:10.1007/s00004-018-0412-x
- Syswerda, G. (1993). Simulated crossover in genetic algorithms (ss. 239–255). doi:10.1016/B978-0-08-094832-4.50021-0
- Szalapaj, P. (2001). *CAD principles for architectural design*. New York: Architectural Press.
- Şimşir, Z. (1990). Konya Selçuklu Çinilerinde Kullanılan Motifler. Yayınlanmamış Yüksek Lisans Tezi. Konya: Selçuk Üniversitesi.
- Tamke, M., Nicholas, P. ve Zwierzycki, M. (2018). Machine learning for architectural design: Practices and infrastructure. *International Journal of Architectural Computing*, 16(2), 123–143. doi:10.1177/1478077118778580
- Tan, M. (1989). Saying what it is by what it is like describing shapes using line relationships. *CAAD futures Digital Proceedings* içinde (ss. 201–214).
- Tang, X., Yin, Y., Lin, Q., Hadad, R. ve Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148(May 2019), 103798. doi:10.1016/j.compedu.2019.103798
- Tecim, V. (2004). Sistem yaklaşımı ve soft sistem düşüncesi. *D.E.Ü. İ.İ.B.F Dergisi*, 19(2), 75–100.
- Tedeschi, A. (2014). *AAD algorithms-aided design: Parametric strategies using Grasshopper*. (A. Tedeschi, Ed.) *Le Penseur Publisher*. Brienza, Italy: Le Penseur Publisher.
- Tedre, M. ve Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* içinde (ss. 120–129). New York, NY, USA: ACM. doi:10.1145/2999541.2999542
- Temizyürek, K. (2003). *Fen öğretimi ve uygulamaları*. Ankara: Nobel Akademik Yayıncılık.
- Terzidis, K. (2004). Algorithmic design: A paradigm shift in architecture? *eCAADe* içinde (ss. 201–207). Copenhagen. <http://www.bol.ucla.edu/~kostas/> adresinden erişildi.

- Terzidis, K. (2006). *Algorithmic Architecture*. Oxford: Architectural Press.
- Terzidis, K. (2007). Digital design: Ideological gap or paradigm shift? *SIGraDI Proceedings*, 220–224.
- Thomas, W. (2001). Logic for computer science: The engineering challenge, (August 2000), 257–267. doi:10.1007/3-540-44577-3_18
- Tobin, K. G. ve Capie, W. (1981). The development and validation of a group test of logical thinking. *Educational and Psychological Measurement*, 41(2), 413–423. doi:10.1177/001316448104100220
- Torrance, E. P. (1968). *Education and creative potential: modern school practices series* (5. bs.). Minneapolis: University of Minnesota Press.
- Torus, B. (2008). *Konut ön tasarım sürecinde kural tabanlı bir yazılım modeli: Mardin örneği*. Yıldız Teknik Üniversitesi.
- Torus, B. (2011). Learning from vernacular Turkish house: Designing mass-customized houses in Mardin. *Intercultural Understanding*, 1(1984), 105–112.
- Touloupaki, E. ve Theodosiou, T. (2017). Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization - A review. *Energies*, 10(5). doi:10.3390/en10050637
- Toussi, H. (2020). The application of evolutionary, generative, and hybrid approaches in architecture design optimization. *Journal of Faculty of Architecture*, 2(2), 1–20.
- Tseytin, G. S. (1981). *From logicism to proceduralism (an autobiographical account)*.
- Turhan, K. (2018). *Fraktal geometrinin iç mimari kurguda kullanımına yönelik bir araştırma*. Hacettepe Üniversitesi.
- Turing, A. M. (1936). On computable numbers with an application to the entscheidungs problem. *Proc. of the London Mathematical Society*, 2(42), 181–203. doi:10.1007/s10506-017-9200-2
- Turrin, M. (2014). *Performance assessment strategies: A computational framework for conceptual design of large roofs* (C. 4). TU Delft. http://abe.tudelft.nl/index.php/faculty-architecture/article/view/turrin/remote_turrin adresinden erişildi.
- Ubiquity staff. (2007). An interview with Peter Denning on the great principles of computing. *Ubiquity*, 2007(June), 1–1. doi:10.1145/1276162.1276163
- Uzun, C. (2021). The network of interactions for an artificial architectural intelligence. *The Routledge Companion to Artificial Intelligence in Architecture* içinde . New York: Routledge.

- Valanides, N. C. (1996). Formal reasoning and science. *Ethnographies of Reason*, 96(2), 99–107. doi:10.4324/9781315580555-2
- Varinlioglu, G., Halici, S. M. ve Alacam, S. (2016). Computational thinking and the architectural curriculum: simple to complex or complex to simple? *eCAADe 34 / CAAD EDUCATION / Design Tools*, 1(August), 253–259.
- Veloso, P., Celani, G. ve Scheeren, R. (2018). From the generation of layouts to the production of construction documents: An application in the customization of apartment plans. *Automation in Construction*, 96(May), 224–235. doi:10.1016/j.autcon.2018.09.013
- Veloso, P. ve Krishnamurti, R. (2021). Mapping generative models for architectural design. *The Routledge Companion to Artificial Intelligence in Architecture* içinde (ss. 29–58). Abington, Oxon; New York: Routledge, 2021.: Routledge. doi:10.4324/9780367824259-4
- Von Neumann, J. (1951). The general and logical theory of automata. *Cerebral mechanisms in behavior-the Hixon symposium* içinde (ss. 1–41). New York: John Wiley.
- Voogt, J., Fisser, P., Good, J., Mishra, P. ve Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. doi:10.1007/s10639-015-9412-6
- Vukorep, I. ve Kotov, A. (2021). Machine learning in architecture an overview of existing tools. *The Routledge Companion to Artificial Intelligence in Architecture* içinde . New York: Routledge.
- Wagdy, A. ve Fathy, F. (2015). A parametric approach for achieving optimum daylighting performance through solar screens in desert climates. *Journal of Building Engineering*, 3, 155–170. doi:10.1016/j.jobbe.2015.07.007
- Walden, J., Doyle, M., Garns, R. ve Hart, Z. (2013). An informatics perspective on computational thinking. *Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13* içinde (s. 4). New York, New York, USA: ACM Press. doi:10.1145/2462476.2483797
- Wang, K., Savva, M., Chang, A. X. ve Ritchie, D. (2018). Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics*, 37(4). doi:10.1145/3197517.3201362
- Wang, S.-C. (2003). Artificial neural network. *Interdisciplinary Computing in Java Programming* içinde (C. 725, ss. 81–100). Boston, MA: Springer US. doi:10.1007/978-1-4615-0377-4_5

- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. ve Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. doi:10.1007/s10956-015-9581-5
- Weisberg, D. (2008). The engineering design revolution: The people, companies and computer systems that changed forever the practice of engineering. www.cadhistory.net adresinden erişildi.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49, 33–35. doi:10.1201/b16812-43
- Wing, J. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. doi:10.1098/rsta.2008.0118
- Wing, J. M. (2010). Computational thinking: What and why? *The Magazine of the Varnege Mellon University School of Computer Science*, (March 2006), 1–6. <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why> adresinden erişildi.
- Wolfewicz, A. (2021). Deep learning vs. machine learning — What’s the difference? *Levity*. <https://levity.ai/blog/difference-machine-learning-deep-learning> adresinden erişildi.
- Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, 311(5985), 419–424.
- Wolfram, S. (2002). *A new kind of science*. Champaign: IL: Wolfram Media.
- Woodbury, R. (2010). *Elements of parametric design*. New York: Routledge.
- Woodbury, R. F. ve Burrow, A. L. (2006). Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 20(2), 63–82. doi:10.1017/S0890060406060057
- Wortmann, T., Cichocka, J. ve Waibel, C. (2022). Simulation-based optimization in architecture and building engineering—Results from an international user survey in practice and research. *Energy and Buildings*, 259(January). doi:10.1016/j.enbuild.2022.111863
- Wortmann, T., Costa, A., Nannicini, G. ve Schroepfer, T. (2015). Advantages of surrogate models for architectural design optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 29(4), 471–481. doi:10.1017/S0890060415000451

- Wortmann, T. ve Nannicini, G. (2016). Black-box optimisation methods for architectural design. *Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia* içinde (ss. 177–186).
- Wortmann, T. ve Nannicini, G. (2017). Introduction to architectural design optimization. *Springer Optimization and Its Applications*, 128(December), 259–278. doi:10.1007/978-3-319-65338-9_14
- Wright, J. A., Loosemore, H. A. ve Farmani, R. (2002). Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings*, 34(9), 959–972. doi:10.1016/S0378-7788(02)00071-3
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S. ve Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1). doi:10.1145/2576872
- Yağcı, M. (2019). A valid and reliable tool for examining computational thinking skills. *Education and Information Technologies*, 24(1), 929–951. doi:10.1007/s10639-018-9801-8
- Yang, X.-S. (2021). Genetic Algorithms. *Nature-Inspired Optimization Algorithms* içinde (ss. 91–100). USA: Elsevier. doi:10.1016/B978-0-12-821986-7.00013-5
- Yang, X. S. (2010). *Engineering optimization: An introduction with metaheuristic applications*. Chichester: John Wiley & Sons.
- Yang, X. S. ve Koziel, S. (2011). Computational optimization: An overview. *Studies in Computational Intelligence*, 356, 1–11. doi:10.1007/978-3-642-20859-1_1
- Yazıcı, S. (2013). *Mimarlıkta malzeme tabanlı bütünleşik hesaplamalı tasarım modeli*. İstanbul Teknik Üniversitesi.
- Yazıcı, S. (2020). Computational design informed by natural systems. *Journal of Computational Design and Engineering*, 1(3), 1–16.
- Yu, X. ve Su, Y. (2015). Daylight availability assessment and its potential energy saving estimation -A literature review. *Renewable and Sustainable Energy Reviews*, 52, 494–503. doi:10.1016/j.rser.2015.07.142
- Zhang, S. H., Zhang, S. K., Liang, Y. ve Hall, P. (2019). A survey of 3d indoor scene synthesis. *Journal of Computer Science and Technology*, 34(3), 594–608. doi:10.1007/s11390-019-1929-5
- Zimmerman, A. (2006). *Integrated design process*. Canada: Canada Mortgage and Housing Corporation.