Çok Katlı Betonarme Yapıların Optimal Ön Tasarımı için Bir Strateji

Pınar Ataş

YÜKSEK LİSANS TEZİ

İnşaat Mühendisliği Anabilim Dalı

Ekim 2020

A Strategy for Optimal Preliminary Design of Multi-storey Reinforced Concrete Structures

Pınar Ataş

MASTER OF SCIENCE THESIS

Department of Civil Engineering

October 2020

Çok Katlı Betonarme Yapıların Optimal Ön Tasarımı İçin Bir Strateji

Pınar Ataş

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
İnşaat Mühendisliği Anabilim Dalı
Mekanik Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Hakan Özbaşaran

Ekim 2020

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Hakan ÖZBAŞARAN danışmanlığında hazırlamış olduğum "Çok Katlı Betonarme Yapıların Optimal Ön Tasarımı İçin Bir Strateji" başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik, ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik, ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik, ilke ve kurallara göre sunduğumu beyan ederim. 21/10/2020

Pınar ATAŞ

İmza

ÖZET

Bu araştırmanın amacı betonarme yapıların optimal tasarımını gerçekleştirmek amacıyla bilgisayar ile otomatik tasarım yöntemlerinin uygulanıp uygulanamayacağının başlangıç testlerini yapmaktır. Günümüz bilgisayar teknolojisi göz önüne alındığında bilgisayar destekli tasarımın artık çağın taleplerine cevap vermediği açıktır. Bu amaçla mühendise ön tasarım için taşıyıcı sistem planları (yapısal analiz yapmadan) önermek adına bir strateji sunulmuştur. Burada "taşıyıcı sistem planı"ndan anlaşılacak şey taşıyıcı sistem elemanlarının plandaki yerleşimini belirlemek ve temsili bir ön boyutlandırma yapmaktır.

Sunulan strateji için ilk olarak inşaat mühendisliği alanında kapsamlı bir literatür taraması yapılarak tasarımlara yön verecek tasarım pratikleri derlenmiştir. Derlenen tasarım pratikleri optimizasyon sürecinde algoritmanın başvuracağı kısıtları oluşturmuştur. Daha sonra kaydedilen her bir tasarım pratiği için bir uygunluk değeri belirlenerek algoritma tarafından ne şekilde kullanılacağı belirlenmiştir. Bu aşamada her tasarım pratiği için bir cezalandırma sistemi oluşturulmuştur. Daha sonra "girdi biçimi" ve "tasarım stratejisi" belirleme sürecine geçilmiştir. Verilen bir mimari planın bilgisayar tarafından algılanmasını sağlayan en uygun girdi biçimi ve tasarım sürecinin nasıl gerçekleşeceğine yön veren bir tasarım stratejisi oluşturularak sayısal testler yapılmıştır.

Sunulan stratejinin test edilmesi için basit, orta ve zor seviyelerde olmak üzere toplamda dört adet mimari proje kullanılmıştır. Çıkan sonuçlar sonrası özellikle düzgün mimari plana sahip konut projeleri için sunulan stratejinin ümit verici olduğu görülmüştür.

Bu çalışmanın en büyük katkısı bilgisayar ile otomatik tasarımlar üretmenin mümkün olduğunu görmek olmuştur. Ayrıca bu çalışma otomatik tasarım süreçlerinin ileriki zamanlarda ne denli yaygın olacağının sinyallerini vermiştir.

Anahtar kelimeler: Yapay zekâ, Otomatik tasarım, Metasezgisel algoritmalar, Taşıyıcı sistem, Betonarme

SUMMARY

The aim of this research is to execute initial tests of whether computer-automated design methods can be applied in order to perform the optimal design of reinforced concrete structures. Considering today's computer technology, it is obvious that computer-aided design does not respond to the demands of our age. For this purpose, a strategy has been presented to recommend structural system plans (without structural analysis) for preliminary design to the engineer. Here, the meaning of the "structural system plan" is to determine the location of the structural system elements in the plan and to make a representative preliminary dimensioning.

First, a comprehensive literature review in the field of civil engineering was conducted and design practices that would guide the designs were collected. The collected design practices are the constraints that the algorithm will utilize in the optimization process. Then, a suitability measure for each design practice and how it would be used by the algorithm were determined. At this stage, a penalty scheme has been created for each considered design practice. Then, the process of determining "input type" and "design strategy" was started. Finally, numerical tests have been carried out after creating a strategy that guides how the design process will take place and an appropriate input form that enables the perception of a given architectural plan by the computer.

In order to test the presented strategy, a total of four architectural plans of simple, medium and hard difficulty levels were used. It has been seen from the results that the presented strategy is promising especially for residences with simple architectural plans.

The main contribution of this study is to show that it is possible to produce automatic designs with computers. In addition, this study indicates how common automated design processes will be in the future.

Keywords: Artificial intelligence, Automated design, Metaheuristic algorithms, Structural system, Reinforced concrete

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde iki yıl boyunca değerli bilgilerini benimle paylaşan, kullandığı her kelimenin hayatıma kattığı önemini asla unutmayacağım, yanında çalışmaktan onur duyduğum ve tecrübelerinden yararlanırken göstermiş olduğu hoşgörü ve sabırdan dolayı değerli danışman hocam Doç. Dr. Hakan ÖZBAŞARAN'a, desteklerinden dolayı arkadaşlarım Batuhan ŞANLI ve Yunus Eren KAYA'ya, çalışmam boyunca benden bir an olsun yardımını esirgemeyen kardeşim Deniz ATAŞ'a ve çalışma süresince tüm zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan aileme sonsuz teşekkürlerimi sunarım.

Bu çalışma Doç. Dr. Hakan ÖZBAŞARAN'ın yürütücüsü olduğu 119M848 kodlu TÜBİTAK 1002 projesi süresince ortaya çıkan bazı sorulara cevap bulmak amacıyla yapılmıştır. Desteği için TÜBİTAK'a ayrı bir paragrafta teşekkür etmek gerekir.

İÇİNDEKİLER

ÖZET vi
SUMMARY vii
TEŞEKKÜR viii
İÇİNDEKİLER ix
ŞEKİLLER DİZİNİ x
ÇİZELGELER DİZİNİ xiv
1. GİRİŞ VE AMAÇ 1
2. LİTERATÜR ARAŞTIRMASI 8
3. BETONARME TAŞIYICI SİSTEM PLANI TASARIM PRATİKLERİ 14
4. OPTİMAL TASARIM 28
4.1. Yapısal Optimizasyon 32
4.1.1. Şekil optimizasyonu 34
4.1.2. Boyut optimizasyonu 35
4.1.3. Topoloji optimizasyonu 36
4.2. Yapısalda Optimizasyonda Karşılaşılan Zorluklar 37
4.2.1. Metasezgisel algoritmalar 40
5. ÖNERİLEN TAŞIYICI SİSTEM PLANI HAZIRLAMA YÖNTEMİ 46
5.1.Optimizasyon Algoritması 54
5.2.Sunulan Strateji 56
6. BULGULAR VE TARTIŞMA 69
6.1. Örnekler 72
6.1.1. Basit bir konut projesi 72
6.1.2. Düzgün mimari plana sahip bir konut 80
6.1.3. Yamuk mimari plana sahip bir konut 88
6.1.4. Çok yamuk mimari plana sahip bir konut 95
7. SONUÇ VE ÖNERİLER 103
KAYNAKLAR DİZİNİ 104

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

1.1. Betonarme olarak tasarlanmış bir köprü.....1
1.2. Betonarme yapılar. 2
1.3. Geleneksel tasarım süreci (Coello vd., 1997). 3
1.4. Optimum tasarım süreci (Coello vd., 1997). 6
3.1. Betonarme taşıyıcı sistem elemanları.....14
3.2. Örnek bir taşıyıcı sistem planında kolon, kiriş ve döşemeler. 15
3.3. Süreksiz kirişler.....17
3.4. Eksenleri çakışık olmayan kirişler ile sürekli kiriş hatlarının oluşturulması. 19
3.5. Saplama kirişler.....20
3.6. Saplaması olan bir kiriş, başka bir kirişe saplanmamalıdır. 20
3.7. Uygulamada yapılan saplamanın saplaması kirişler 21
3.8. Kolona yakın saplanan kiriş. 21
3.9. Uygulamada yapılan ve kolona çok yakın olan saplama kiriş. 22
3.10. Bir sistemde her iki deprem yönünde yerleştirilen perdeler. 23
3.11. Dış akslara yerleştirilen perdeler. 24
3.12. Perdelerin sistem çizgilerinin bir noktada birleşmesi engellenmelidir. 24
3.13. Bir yapının her iki deprem yönünde yerleştirilen perdelere göre davranışı. 25
3.14. Bir yöne göre daha rijit davranış gösteren bir yapının deprem sırasındaki durumu...25
4.1. Yalnızca eşitsizlik kısıtlamalarının olduğu, iki boyutlu bir tasarım alanında uygulanabilir bölge 31
4.2. Optimizasyon problemlerinin sınıflandırılması. 32
4.3. Yapısal optimizasyonun faydaları. 33
4.4. Yapısal optimizasyonun alt başlıkları. 34
4.5. Yapısal optimizasyon çeşitleri. 34
4.6. Sınır düğümü kombinasyonlarının olası üç dizisi.....35
4.7. Bir kafes sistem için boyut optimizasyonu. 36
4.8. Temel yapısı yaklaşımında mevcut durumlar.....36
4.9. f(x) denkleminin global değeri. 37

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

4.10. Matematiksel olarak tanımı kolay olmayan bir problem. 38
4.11. Eyer noktası. 39
4.12. Gerçek dünya problemlerine benzer olan minimum ve maksimum noktaları. 40
4.13. Popülasyon tabanlı algoritmalar. 42
5.1. Yalnızca girdi ve çıktıları açısından görüntülenebilen kara kutu problemleri. 46
5.2. Örnek bir aks sistemi. 50
5.3. Aks sistemi üzerinde kolon yerleşimleri. 50
5.4. Ayrık optimizasyon sonucu oluşan kiriş topolojisi. 51
5.5. Basit bir planda 2-A akslarının kesişim noktasına yerleşen bir kolon. 52
5.6. Bir kolona ait tasarım değişkenleri. 52
5.7. Bir kirişe ait tasarım değişkenleri. 54
5.8. Jaya algoritmasının akış şeması. 56
5.9. Örnek olarak ele alınan mimari proje 59
5.10. Kiriş topolojisi sırasında optimizasyon algoritması tarafından optimal topolojinin belirlenmesi aşamaları 59
5.11. Kiriş topolojisi belirlenen sistem için kolon topolojisinin belirlenmesi 60
5.12. Kirişlerde kaçıklık durumları ve gösterim ifadeleri. 62
5.13. Kirişlerde kaçıklık durumları ve gösterim ifadeleri. 62
5.14. Kolon kaçıklık ifadesi "4" olan bir kolonun pozisyonu. 63
5.15. Bir kolonun iki aksa göre aldığı pozisyonlar ve gösterim ifadeleri. 64
5.16. Şekil ve boyut optimizasyonları sonrası elde edilen tasarımlar. 65
5.17. Birinci ve ikinci yinelemelere ait tasarımlar: b) birinci yineleme, c) ikinci yineleme. 66
5.18. Üçüncü ve dördüncü yinelemelere ait tasarımlar: d) üçüncü yineleme, e)dördüncü yineleme 66
5.19. Dördüncü yineleme sonrası elde edilen uygulanabilir tasarım. 67
5.20. Sunulan stratejiye ait akış şeması. 67
6.1. Basit bir konut için mimari plan ve uygun biçime dönüştürülen "DXF" çizim. 69

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

6.2. Mimari projenin DXF biçiminde düzenlenme aşamaları: a) aksların çizilmesi, b) yapı oturma alanının çizilmesi. 70
6.3. Mimari projenin DXF biçiminde düzenlenme aşamaları 2 71
6.4. Mimari projenin DXF biçiminde düzenlenme aşamaları 3 72
6.5. Basit bir konut için üretilen iyi 1. plan.....73
6.6. Basit bir konut için üretilen en iyi 1. planda kiriş ve kolonların nihai durumları. 74
6.7. Basit bir konut için üretilen en iyi 2. plan. 74
6.8. Basit bir konut için üretilen en iyi 2. planda kiriş ve kolonların nihai durumları.....75
6.9. Basit bir konut için üretilen en iyi 3. plan. 75
6.10. Basit bir konut için üretilen en iyi 3. planda kiriş ve kolonların nihai durumları.....76
6.11. Basit bir konut için üretilen en kötü 1. plan.....77
6.12. Basit bir kiriş için üretilen en kötü 1. planda kiriş ve kolonların nihai durumları. 77
6.13. Basit bir konut için üretilen en kötü 2. plan.....78
6.14. Basit bir kiriş için üretilen en kötü 2. planda kiriş ve kolonların nihai durumları.. 78
6.15. Basit bir konut için üretilen en kötü 3. plan.....79
6.16. Basit bir kiriş için üretilen en kötü 3. planda kiriş ve kolonların nihai durumları.....80
6.17. Düzgün plana sahip bir konut için mimari plan ve uygun biçime dönüştürlen DXF çizim.....81
6.18. Düzgün plana sahip bir konut projesi için üretilen en iyi 1. plan 81
6.19. Düzgün plana sahip bir konut projesi için üretilen en iyi 1. planın son hali. 82
6.20. Düzgün plana sahip bir konut projesi için üretilen en iyi 2. plan. 82
6.21. Düzgün plana sahip bir konut projesi için üretilen en iyi 2. planın son hali. 83
6.22. Düzgün plana sahip bir konut projesi için üretilen en iyi 3. plan. 83
6.23. Düzgün plana sahip bir konut projesi için üretilen en iyi 3. plan. 84
6.24. Düzgün plana sahip bir konut projesi için üretilen en kötü 1. plan.....84
6.25. Düzgün plana sahip bir konut projesi için üretilen en kötü 1. planın son hali.....85
6.26. Düzgün plana sahip bir konut projesi için üretilen en kötü 2. plan.....86
6.27. Düzgün plana sahip bir konut projesi için üretilen en kötü 2. planın son hali.....86

ŞEKİLLER DİZİNİ (devam)

Sekil **Sayfa**

6.28. Düzgün plana sahip bir konut projesi için üretilen en kötü 3. plan. 87
6.29. Düzgün plana sahip bir konut projesi için üretilen en kötü 3. planın son hali. 87
6.30. Yamuk plana sahip bir konut için mimari plan ve DXF çizim. 88
6.31. Yamuk plana sahip bir konut projesi için üretilen en iyi 1. plan 89
6.32. Yamuk plana sahip bir konut projesi için üretilen en iyi 1. planın son hali. 89
6.33. Yamuk plana sahip bir konut projesi için üretilen en iyi 2. plan. 90
6.34 Yamuk plana sahip bir konut projesi için üretilen en iyi 2. planın son hali. 90
6.35. Yamuk plana sahip bir konut projesi için üretilen en iyi 3. plan. 91
6.36. Yamuk plana sahip bir konut projesi için üretilen en iyi 3. planın son hali.. 91
6.37. Yamuk plana sahip bir konut projesi için üretilen en kötü 1. plan.. 92
6.38. Yamuk plana sahip bir konut projesi için üretilen en kötü 1. planın son hali. 93
6.39. Yamuk plana sahip bir konut projesi için üretilen en kötü 2. plan.. 93
6.40. Yamuk plana sahip bir konut projesi için üretilen en kötü 2. planın son hali.. 94
6.41. Yamuk plana sahip bir konut projesi için üretilen en kötü 3. plan. 94
6.42. Yamuk plana sahip bir konut projesi için üretilen en kötü 3. planın son hali. 95
6.43. Çok yamuk plana sahip bir konut için mimari plan ve DXF çizim 96
6.44. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 1. plan.....96
6.45. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 1. planın son hali. 97
6.46. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 2. plan. 97
6.47. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 2. planın son hali. 98
6.48. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 3. plan.....98
6.49. Çok yamuk plana sahip bir konut projesi için üretilen en iyi 3. planın son hali.....99
6.50. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 1. plan.....99
6.51. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 1. planın son hali. ...100
6.52. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 2. plan. 100
6.53. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 2. planın son hali. ...101
6.54. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 3. plan.....102
6.55. Çok yamuk plana sahip bir konut projesi için üretilen en kötü 3. planın son hali....102

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

4.1. Optimizasyon bileşenleri 30
5.1. Tasarım pratiklerinin stratejideki görevleri. 58
5.2. Boyut ve şekil optimizasyonları sırasında kullanılan vektör şemasında var olan değerlerin karşılık geldiği ifadeler. 65

1. GİRİŞ VE AMAÇ

Betonarme, Portland çimento betonunun yüksek basınç mukavemeti ve çeliğin sünekliği ve yüksek çekme mukavemeti özellikleri ile karakterize edilen, onu ideal bir yapı malzemesi haline getiren kompozit bir malzemedir (Bertolini vd., 2004). Sahip olduğu birçok avantaj ile inşaat mühendisliğinin hemen her alanında kullanılan betonarme, ilk olarak 19. yüzyılın ortalarında kullanılmıştır. Özellikle çelik yapılara kıyasla ekonomik olması, kolay uygulanabilirliği, ihmal edilebilir bakım maliyetleri, yangına, korozyona ve asitli ortamlara karşı gösterdiği dayanımı ile popülaritesi giderek artmıştır ve günümüzde de sıklıkla tercih edilmeye devam etmektedir (Şekil l.1). Ülkemizde de yapı malzemesi olarak inşaat mühendisliğinin her alanında yaygın bir şekilde kullanıldığını söyleyebiliriz (Şekil 1.2).



Şekil 1.1. Betonarme olarak tasarlanmış bir köprü.

Bilgisayarların yapısal analiz pratiğinde kullanılmasından önceki dönemlerde bina sistemleri tasarlanırken yapı ilk olarak el yöntemleri ile analiz edilebilen ve "alt sistemler" olarak adlandırılan parçalara ayrılır. Tipik bir yapı ele alındığında alt sistemlere örnek olarak "kat çerçeveleri" verilebilir. İlk olarak alt sistemler sırayla analiz edilerek donatı hesapları yapılır. Daha sonra bulunan kiriş kesme kuvvetleri alt alta toplanarak kolonların eksenel yükleri oluşturulur ve kolonlar da hazır tablolardan yararlanılarak donatılandırılır. Ardından bu aşamaya kadar yapılan işlemlere deprem, rüzgâr veya toprak itkisi gibi yatay

yüklerin etkisi de eklenir. Yatay yükler altında yapı bir bütün olarak davrandığı için bu aşamada genellikle yapının tamamını bir defada hesaba katabilen yaklaşık yöntemlere başvurulur. Çoğunluğu kat burulmalarını ihmal eden bu yöntemler ile yapı yatay yükler altında analiz edilir ve elde edilen değerler düşey yükler altında bulunan değerler ile birleştirilir. Çıkan nihai değerler ile ek donatıların gerekliliği kontrol edilerek tasarım sonuçlandırılır (Kubin, 1994).

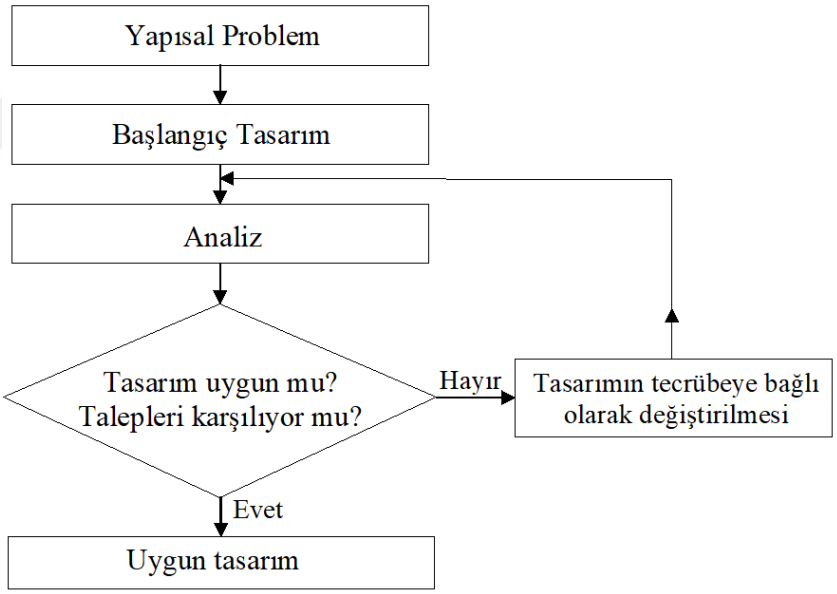


Şekil 1.2. Betonarme yapılar.

Bilgisayar kullanımının 90'lı yıllarda yaygınlaşmasıyla her alanda olduğu gibi mühendislik alanlarında da devrim niteliğinde çalışmalar başlayarak, birçok disiplinde çalışanlara büyük kolaylıklar sağlanmıştır. Özellikle "BDT: Bilgisayar Destekli Tasarım (İngilizce ismi ile CAD: Computer-Aided Design)" kavramanın hayata geçmesiyle özellikle iş gücünden büyük kazançların elde edildiği yeni bir döneme girilmiştir. Bu süreçte inşaat mühendisliği alanında ilk zamanlar matris yapısal analizinin ve daha sonra sonlu elemanlar yönteminin geliştirilmesi ile bir yapının matematiksel modelinin analizini ele alan yapısal tasarımın önemli bileşeni büyük ölçüde bilgisayar programlarına verilmiştir (Fenves, 1981). Bu yazılımlar ile çok çeşitli yapı sistemlerini modellemek, analizlerini yapmak ve tasarlayabilmek mümkün hale gelmiştir.

BDT araçları sonrası inşaat mühendisleri daha fazla alternatif keşfederek daha kapsamlı analitik modeller kullanıp, ayrıntılı hesaplamaların zamanını ve maliyetini azaltarak müşterilerinin ihtiyaçlarına daha duyarlı olmuşlardır.

Betonarme yapı tasarımı paket programlar ile gerçekleştirilirken ilk aşamada yapı geometrisi ve yükler, kullanılan yazılımın imkânları ve kullandığı yöntemler doğrultusunda sisteme tanıtılarak yapıya ait kolon, kiriş, döşeme boyutları ve yerleşimi için bir çözüm aranır. Yapının sisteme tanıtılması sonrasında analiz sürecine geçilerek, tanımlanan tasarımın uygunluk değeri ölçülür. Bu aşamada herhangi bir nedenden ötürü tasarımı yapılamayan bir yapı elemanları varsa, tasarım süreci mühendisin taşıyıcı sistemde uygun gördüğü değişiklikleri yaparak taşıyıcı sistemi tekrar analiz etmek üzere programa göndermesi ve bir çözüm buluncaya kadar bu işlemleri tekrar etmesi şeklinde ilerler (Şekil 1.3).



Şekil 1.3. Geleneksel tasarım süreci (Coello vd., 1997).

Yukarıda bahsedilen tasarım prosedürü uygulanırken aşağıda başlıklar halinde listelenen konular dikkatten kaçabilmektedir.

- **Emniyet**: Yapısal tasarım gerçekleştirilirken göz önüne alınması gereken en önemli faktörlerin başında emniyet gelir. Bir yapının hem yatay hem de düşey yüklere

karşı güvenli bir şekilde ayakta kalması, tasarım sırasında mühendisin göz önünde bulundurduğu en önemli konulardan birisidir. Özellikle deprem, sel, kasırga, tsunami gibi yer tabanlı yahut iklimsel değişiklikler olarak karşımıza çıkan doğal afetlerin yıkıcı ve ölümcül etkisini azaltmak için öncelikle yapısal alanlarda önlem almak kaçınılmazdır. Ancak özellikle rekabetçi piyasadan kaynaklı en kısa sürede tasarımın oluşturulması ve en az donatı ve beton miktarının kullanıldığı projelerin rağbet görmesi bakımından minimum şartlarda donatı ve beton miktarı kullanımı konularına odaklanan mühendis, tamamlanmayan çerçeveler, ortogonal yönlerdeki toplam ataletlerin dengesizliği gibi tasarım kusurlarını göz ardı edebilir.

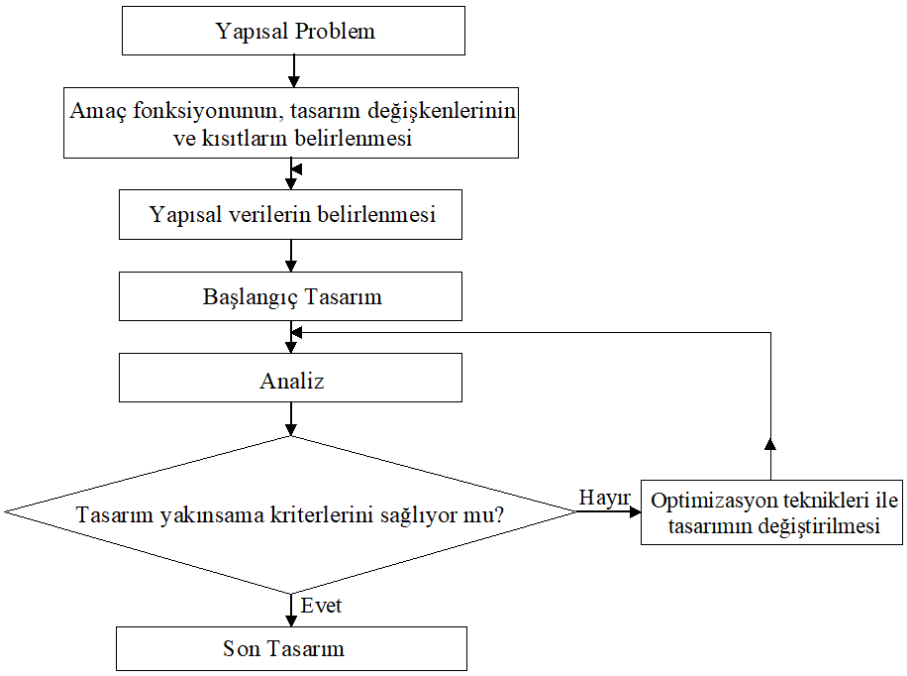
- **Zaman:** Bir projenin tasarım süresi de rekabet ortamında göz önüne alınması gereken temel konulardandır. Mühendis tasarım aşamasında güvenli ve ekonomik bir yapının tasarımını en makul sürede tamamlamalıdır. Bunun yanı sıra BDT deneme-yanılma mantığına dayanan bir süreç olarak karşımıza çıkar. Bu süreç boyunca mühendisten beklenen minimum sürelerde uygun tasarımı oluşturmasıdır. Bu şekilde gergin bir ruh hali yaşaması kaçınılmaz olan tasarımcı tarafından makul sürelerde tasarımı sonuçlandırması, ancak sınırlı sayıda tasarımı değerlendirmesiyle mümkün olacaktır.
- **Maliyet:** Küreselleşme ile birlikte gelen dinamik ve yoğun rekabet ortamında yapılacak tasarımların uygun maliyetlerde sunulması elbette ki göz önüne alınması gereken en önemli konulardandır. Beton ve donatı bir yapının temel bileşenleri oldukları için kullanılan beton ve donatı miktarı yapı maliyetini doğrudan etkileyen faktörlerdendir. Bu kapsamda mühendislik kısıtlarını sağlarken en az malzemenin kullanıldığı yapıların tasarımı, alandaki araştırmaların ana konuları arasındadır. BDT sürecinde yaşanacak zaman kayıplarının önüne geçmek amacıyla denenecek olan sınırlı sayıdaki tasarımlar ile optimum düzeydeki maliyetlerle tasarımı sonuçlandırmak belirsiz bir durum olarak karşımıza çıkar. Belki denenecek başka bir tasarım öncekilerden daha uygun maliyetlerde olacaktır. Ancak zaman kısıtlamasından dolayı başka bir adayı denemeye fırsat olmadığından üretilen tasarımın maliyet açısından optimal bir tasarım olup olmadığı belirsiz olacaktır.
- **Mimari kısıtlar:** Günümüzde taşıyıcı sistem tasarımı sürecinde ilk olarak mimari projeler inşaat mühendisine sunulur ve sonrasında mühendis mimari projeye bağlı kalarak bir taşıyıcı sistem tasarlama sürecine geçer. Taşıyıcı sistem tasarımı sürecinde mühendis belirtilen mimari kısıtlar altında tasarımını gerçekleştirmek

zorundadır. Örneğin; salon gibi kıymetli hacimlerde kiriş gibi yatay elemanların sarkmaması, kolon veya perde gibi düşey elemanların çıkıntı yapmaması en sık karşılaşılan mimari taleplerdendir. Karşılaşılan mimari taleplere bağlı kalarak yapılacak tasarım bazı zorlukları kendisiyle birlikte getirebilir. Bu zorluklar özellikle mühendisin hem gergin bir çalışma süreci yaşamasına hem de istenilen taleplerin sağlanamaması durumunda mimari projelerde değişikliklerin meydana gelmesine yol açabilir.

BDT'nin ilk dönemlerde büyük bir ilgi ile karşılanmasına rağmen hızlı değişen çevre koşulları nedeni ile belirsizliğin hâkim olduğu günümüz küresel rekabet ortamında istenilen taleplere yeteri düzeyde cevap vermediği söylenebilir.

Genel anlamda bir yapı tasarımı gerçekleştirilirken uygun maliyetli, daha sağlam, estetik ve güvenli tasarımlar geliştirmek her mühendisin temel hedefidir. Bir problemin en verimli şekilde çözüme ulaşması için çoğu disiplinde kullanılan optimizasyon kavramı, yapısal anlamda da vazgeçilmez bir bileşen olarak karşımıza çıkmaktadır. Genel olarak yapısal tasarım özellikle paket programlarda "deneme-yanılma" yöntemleriyle gerçekleştirilir (Şekil 1.3). Ancak gerçek şu ki, bu süreç tipik olarak zaman ve insan emeği açısından yüksek maliyetlerle sonuçlanır. Gerçek tasarımda zaman kavramı her zaman bir kısıt olarak karşımıza çıktığı için sorunun tüm gereksinimlerini karşılayan makul bir çözüm genellikle benimsenir ve maliyet sorunu çoğu zaman gerçek anlamda dikkate alınmaz. Bununla birlikte, bazı kısıtlamalar altında bir amaç fonksiyonunu en aza indirerek tasarımı değiştirmekten oluşan optimal tasarım prosedürü, geleneksel tasarım yöntemine bir alternatiftir (Ozturk ve Durmus, 2013) (Şekil 1.4). Özellikle son birkaç on yıldan beri bilgisayarlar mühendislerin tasarım sürecini otomatikleştirmelerine yardımcı olmak için kullanılmaktadır. Mevcut optimizasyon tekniklerinin yıllar içinde önemli ölçüde gelişmesiyle hesaplama prosedürlerinde meydana gelen kayda değer gelişmeler, optimizasyon prosedürlerinin yapısal tasarımlara sorunsuz entegrasyonuna sağlamıştır. Özellikle yapısal tasarım problemleri için optimizasyon algoritmaları, bilgisayar teknolojisi ile birleşerek gerçek anlamda karmaşık dünya problemine makul sürelerde gerçekçi çözümler üretirler.

Yapısal optimizasyon problemleri en genel anlamda topoloji, şekil ve boyut optimizasyonu olmak üzere üç başlıkta incelenir. Ayrıntılı olarak Bölüm 4.1'de anlatılan yapısal optimizasyon dalları en kaba şekilde ifade edildiğinde şekil optimizasyonu, bir yapı ve yapı elemanı için en uygun şekli arar. Boyut optimizasyonu, yapı elemanları için en uygun kesitlerin araştırıldığı alandır. Topoloji optimizasyonu ise bir yapıda elemanların varlığını veya yokluğunu, bağlantılarını ve sayısını araştırır. Bir yapının optimal tasarımı için öncelikle yapı elemanları için uygun topolojilerin elde edilmesi, sonrasında ise bu topolojiye bağlı olarak şekil ve boyut optimizasyonlarının gerçekleştirilmesi gerekir. Ancak eşzamanlı yapısal optimizasyon dallarının uygulanması pratik anlamda çok kolay bir işlem değildir. Bunun yanı sıra betonarme yapıların optimizasyonunda en düşük değeri aranan amaç fonksiyonun tanımı (en düşük değeri aranan değer genellikle yapı maliyetidir) tasarım sürecindeki en önemli kararlardan biri olarak karşımıza çıkarken, birçok parametreye bağlı bu değere ait nicel modeller elde etmek kolay değildir. Özellikle matematiksel yöntemlerle optimal tasarımlar elde etmek her açıdan pahalı bir süreç olabilir. Bu yüzden sezgisel yöntemlerin (Bölüm 4.2.1'de ayrıntılı olarak değinilmiştir) kullanılmasıyla birlikte önemli zorlukların üstesinden gelinerek arzu edilebilir tasarım süreçleri meydana gelir.



Şekil 1.4. Optimum tasarım süreci (Coello vd., 1997).

Bilgisayar ile otomatik tasarımlar elde etmenin mümkün olup olmadığının başlangıç testlerini yapmak ve sonuçların okuyucularla paylaşılması bu çalışmanın temel amacıdır. Günümüzde yapay zekânın tüm alanlarda aktif bir şekilde kullanılmasıyla birlikte inşaat mühendisliği disiplininde de yenilikçi tasarım modellerinin hayata geçmesi kaçınılmazdır. Özellikle geleneksel tasarım sürecindeki deneme-yanılma yöntemleri ile yaşanan zaman kayıpları ve yeteri denemenin yapılamaması sonucu daha uygun tasarımlara ulaşamamak gibi sorunların üstesinden gelmek için günün teknolojisinin uygulandığı modellere ihtiyaç vardır. Ayrıca çözüm üretmede aktif bir şekilde kullanılan optimizasyon tekniklerinin de tasarım süreçlerine dahil edilmesi birçok açıdan (zaman, güvenlik ve maliyet gibi) daha uygun çözümler sağlayabilir.

Bu çalışmada birçok alanda başarıyla uygulanmış, metasezgisel algoritmalar sınıfından olan Jaya algoritması ile betonarme yapı elemanlarının bir yapı içindeki yerleşimlerinin belirlenmesine yönelik bir strateji sunulmaktadır. Sunulan stratejinin mühendise ön tasarım için planlar üretmesi hedeflenmektedir. Bilgisayar tarafından otomatik olarak üretilecek çözümler arasından seçilen herhangi bir plan, mühendis tarafından alınarak yapısal analize gönderilmek üzere işlenebilecektir.

2. LİTERATÜR ARAŞTIRMASI

Haug ve Kirmser'in (1967) minimum ağırlıktaki kiriş problemlerini çözmek için yinelemeli bir yönteme dayanan çalışmaları, bilgisayarın optimal tasarım sürecinde kullanılmasına yönelik yapılan önemli girişimlerden biridir. Optimizasyon tekniklerinin kullanımına yönelik Krishnamoorthy and Munro (1981) yayımladıkları bir çalışmada, betonarme çerçevelerin optimizasyonu için doğrusal programlama tekniklerini ele almışlardır. Bu çalışmada uyumluluk, sınırlı süneklik, denge ve uygunluk gibi hususları içeren betonarme çerçevelerinin optimum tasarımını elde etmek için simpleks algoritması kullanılmışlardır. Huanchun ve Zheng (1985) çerçeve sistemlerin optimum tasarımına yönelik 1985 yılında ardışık doğrusal programlama (İngilizce ismi ile SLP: Sequential Linear Programming) tekniği ile bir çalışma yayınlamışlardır. Betonarme çerçevelerin optimizasyonunun karmaşıklığı nedeniyle problem kısıtlarının küresel ve yerel olarak iki alt gruba ayrıldığı bu çalışmada, değişken değerleri tam sayı olarak ele alınarak optimum tasarım için bir yöntem sunulmuştur. Kanagasundaram ve Karihaloo (1990) optimum maliyete sahip betonarme yapıların tasarımlarına ulaşmak için geleneksel tasarım sürecinin matematiksel olarak nasıl taklit edilebileceğini gösteren bir çalışma yayımlamışlardır. Ele alınan problemin çözümü için ardışık doğrusal programlama ve sıralı dışbükey programlama (İngilizce ismi ile SCP: Sequential Convex Programming) tekniklerini kullanmışlardır. Sonuç olarak minimum maliyetli tasarımları elde etme prosedürü matematiksel olarak geleneksel tasarım sürecini taklit eder. Bunun yanında maliyet olarak makul düzeyde olan, pratik olarak uygulanabilir tasarımlar üretilebildiğini göstermişlerdir. Hoit vd. (1991) doğrusal olmayan programlama (İngilizce ismi ile NLP: Nonlinear Programming) teknikleri kullanarak önceki çalışmalara kıyasla daha düşük ağırlıkta iki boyutlu çerçeveler tasarlamışlardır. Fadaee ve Grierson (1996) optimallik kriterlerini ve NLP tekniklerini kullanarak üç boyutlu betonarme çerçevelerde beton, çelik ve kalıp maliyetlerini optimize etmişlerdir. Adamu vd. (1994) betonarme kirişlerin maliyetini optimize etmek için bir yöntem geliştirmişlerdir. Tasarım değişkenlerinin eleman kesitleri (yükseklik ve genişlik), tasarım kısıtlarının ise maksimum yer değiştirme, eğilme ve kesme kuvvetleri olarak ele alındığı çalışmada sürekli tipte bir optimallik kriterleri ve Lagrange çarpanları yöntemi kullanılmıştır. Cohn ve MacRae (1984) NLP tekniklerini kullanarak güçlendirilmiş ve öngerilmeli kirişlerin optimum tasarımını araştırmışlardır.

Matematiksel yöntemlerin yanı sıra bilgisayar teknolojisinin gelişmesiyle artan işlem gücü sayesinde matematiksel yöntemlere göre gerçek dünya problemlerine daha makul sürelerde optimum çözümler üreten ve detaylı olarak Bölüm 4.2.1'de değinilen sezgisel ve genel arama yeteneğine sahip metasezgisel yöntemlerin gelişimi araştırmalara farklı bir boyut kazandırmıştır. Bu yöntemlerle betonarme yapıların optimal tasarımına yönelik yapılan ve dikkat çeken bazı çalışmalar şöyle sıralanabilir: Niğdeli ve Bekdaş (2016) metasezgisel algoritmalar sınıfında yer alan ve İngilizce ismi ile TLBO: Teaching Learning Based Optimization algoritmasını kullanarak eğilme momentleri altında hem betonarme kirişlerin (2015) hem de betonarme kolonların (2016) optimum tasarımlarını incelemişlerdir. Galeb (2009) sunduğu bir çalışmada Amerikan Yapı Yönetmeliği (ACI 318-05) gereklilikleri ve nihai mukavemet tasarım yöntemine dayalı olarak betonarme kirişlerin optimum tasarımını elde etmek amacıyla İngilizce ismi ile SA: Simulated Annealing algoritmasını kullanmıştır. Rajeev ve Krishnamoorthy (1998) yayımladıkları bir çalışmada Genetik Algoritma (GA) kullanarak betonarme düzlem çerçevelerin tasarım optimizasyonunu gerçekleştirmişlerdir. Öztürk ve Durmuş (2013) eksenel kuvvet ve tek eksenli eğilme momentine maruz kalan kolonların optimum maliyet tasarımını elde etmek amacıyla, İngilizce ismi ile Artificial Bee Colony (ABC) algoritmasını kullanmışlardır.

Mühendislik uygulamalarında ele alınan problemlerin çözüm kümeleri genellikle keyfi değerlere sahip olamaz. Çözüm kümelerinde herhangi bir kısıtlama gerekmeyen problemler için sürekli optimizasyon teknikleri kullanılırken özellikle mühendislik alanlarında değişkenlerin bir kısmının veya tamamının bir tamsayı veya ayrık değerler listesinden seçilmesine bağlı olarak Bölüm 4.1'de de ayrıntılı olarak değinilen ayrık optimizasyon teknikleri uygulanır. Balling ve Yao (1997) sundukları çalışmada, betonarme elemanların kesitleri ile birlikte donatı elemanlarının sayısını, çapını ve topolojisini eş zamanlı olarak ayrık değişkenlerle optimize etmek amacıyla SA algoritmasını "çok seviyeli bir yöntem"le kullanarak üç boyutlu, çok açıklıklı ve çok katlı betonarme çerçeveler üzerinde analizler yapmışlardır. Yerçekimi ve yanal yüklerin kombinasyonlarına tabi betonarme düzlem çerçevelerin ayrık optimizasyonunu gerçekleştirmek için, Lee ve Ahn (2003) GA, Kaveh ve Sabzi (2012) İngilizce ismi ile BB-BC: Big Bang-Big Crunch algoritmasını kullanmışlardır. Yapılan çalışmalar sonrasında çok açıklıklı ve çok katlı iki boyutlu betonarme çerçeveler üzerinde analizler yapmışlardır. Choi ve Kwak (1990) betonarme elemanların bireysel tasarımı ile optimal yapı tasarımının

elde edileceği düşüncesine dayanarak, iki aşamalı bir algoritma öne sürmüşlerdir. Bu algoritma öncelikle veri tabanında önceden belirlenmiş kesitlere ait değişkenlerin sürekli çözümünü, sonrasında da veri tabanı içinde doğrudan arama ile ayrık çözümü belirlemiştir. Böylece tüm yapının optimizasyonu bireysel üye optimizasyonu ile gerçekleştirilmiştir. Bu algoritma ile kesit boyutları ve donatı miktarı gibi tasarım değişkenleri tek bir tasarım değişkeni ile bağdaştırılarak değişken sayısı azaltılmış ve buna bağlı olarak matematiksel programlama yöntemlerinin kullanımında ayrık değişkenlerle ilgili yaşanan problemler çözülebilir hale gelmiştir. Algoritma iki bölmeli, on katlı bir yapı ile test edilmiştir.

Sahip olduğu avantajların yanı sıra günümüzde yaşanan küresel rekabet ortamı için yeterli olmadığı düşünülen BDT yöntemleri, bilgisayar teknolojisinde meydana gelen önemli gelişmelerle yerini "Bilgisayar ile Otomatik Tasarım (İngilizce ismi ile CAutoD: Computer-Automated Design)" kavramına bırakmaya hazırlanmaktadır. Değişim her geçen gün kaçınılmazdır ve 90'lı yıllarda ortaya çıktığı vakit herkesi heyecanlandıran paket programlar artık çağın teknolojisinin gerisinde kalmaktadır. Özellikle yapay zekânın neredeyse her alanda karşımıza çıktığı bu günlerde otomatik tasarımların gelecek için kaçınılmaz olduğunu söylemek yerinde olacaktır.

Bilgisayar destekli tasarım için CAD terimini oluşturmasıyla tanınan ve sayısal olarak kontrol edilen makine araçlarında yaygın olarak kullanılan üst düzey bir bilgisayar programlama dili olan APT'nin (ingilizce ismi ile Automatically Programmed Tool) babası olarak görülen Douglas T. Ross (1960), bilgisayar destekli tasarımın yerini almaya hazırlanan bilgisayar ile otomatik tasarım kavramı ile ilgili Türkçe çevirisi ile şu ifadeleri kullanmıştır:

> *"Otomatik tasarım ile bir bilgisayarın insan müdahalesi olmadan çalıştırabileceği bir biçimde, tamamen belirlenebilen tasarım prosedürleri kastedilmektedir. Elektrik motorları, transformatörler, elektronik devre düzenleri ve diğerlerinin tasarımı için bu tip otomatik tasarım programlarının giderek artan örnekleri vardır. Tüm bu sistemlerde esasen sabit bir tasarım veya tasarım sınıfı vardır. Bilgisayar programı tarafından çözülen sorun, genelleştirilmiş tasarımın parametrelerini, belirli bir uygulamanın getirdiği kısıtlamalara uyacak şekilde değiştirmektir. Bu tür programlar, aynı sorunun çözümüne göre çok önemli bir ekonomik avantaja sahip olabilir ve bazı durumlarda sadece tatmin edici veya yeterli olanlar yerine optimum tasarımlar elde etmek için kullanılabilirler."*

Bilgisayar teknolojisinin gelişimi ile 90'lı yılların ortalarından sonra, inşaat mühendisliği alanında da birçok araştırmacı bilgisayar ile otomatik tasarım adı altında birçok çalışmaya imza atmış ve literatüre konuya dayalı çok sayıda örnek çalışma kaydedilmiştir. Moharrami ve Grierson (1993) yayımladıkları bir çalışmada betonarme kolon, kiriş ve perdelerin optimum boyutlarının hesabını gerçekleştirmek amacıyla bilgisayar tabanlı otomatik bir yöntem geliştirmişlerdir. Eleman kesitlerinin (genişlik ve yükseklik) tasarım değişkenleri olarak ele alındığı çalışmada eş zamanlı olarak donatı çeliğinin toplam alanı da optimize edilmeye çalışılmış, mukavemet ve rijitlik kısıtlamalarına tabi beton, çelik ve kalıp maliyetlerini en aza indirmek amacıyla NLP teknikleri kullanılmıştır. Ayrıca çalışmada elemanların sürekli değişkenlerle yapılan analizi sonrasında elde edilen değerler gerçekçi bir aralıktaki değerlere dönüştürülerek ilgililere sunulmuştur. Guerra ve Kiousis (2006) betonarme yapıların optimum tasarımı için farklı açıklık ve katlara sahip betonarme yapıların boyut optimizasyonunu gerçekleştirmişlerdir. Coello vd. (1997) sundukları çalışmada betonarme kirişlerin maliyetini en aza indirirken, beton ve çelik maliyetlerini de göz önünde bulunduran ve GA'ya bağlı bir optimizasyon modeli sunmuşlardır. Niğdeli ve Bekdaş (2016) simetrik bir betonarme yapıyı ele alarak elemanlar arasında herhangi bir gruplama yapmaksızın boyut optimizasyonu gerçekleştirmişlerdir. Yayımlanan çalışmada metasezgisel algoritmalar sınıfından İngilizce ismi ile HS: Harmony Search algoritması hesaplama süresini kısaltmak amacıyla değiştirilerek kullanılmıştır. Yapılan çalışmalar sonrası elde edilen tasarımın karşılaştırmalı maliyetlerden yaklaşık %35 daha ekonomik olduğu belirtilmiştir. Kripka vd. (2016) yaptıkları çalışmada en düşük maliyete (beton, çelik ve kalıp maliyeti) karşılık gelen betonarme binaların kolon topolojisini elde etmek için bir çalışma sunmuşlardır. Çalışmada analizler sırasında gerilme değerleri belli sınırların altında kalan kolonların, sistemden kaldırılmasıyla gerçekleşen optimizasyon süreci sonrasında elde edilen kolon topoloji örneklerinin, pratikteki uygulamalarla benzerlik gösterdiği sonucu yazarlar tarafından kaydedilmiştir.

Bir betonarme yapı için optimal tasarım gerçekleştirilirken, problem boyutunu azaltmak amacıyla ele alınan yapısal sistem içinde üyeler arasında belli bir gruplandırma stratejisi benimsenebilir. Çerçeve elemanları arasında genellikle kirişler ve kolonlar kendi aralarında gruplandırılarak problemin değişken sayısının azaltılması hedeflenir. Bu işlem birçok çalışmada, farklı biçimlerde problemi kolaylaştırmak amacıyla sıkça başvurulan bir

teknik olarak karşımıza çıkar**.** Camp vd. (2003) GA kullanarak düşük maliyetli betonarme çerçevelerin tasarlanması için sundukları bir çalışmada betonarme elemanların boyut optimizasyonunun yanı sıra, donatının kesitler içerisindeki topolojisini de araştırmışlardır. İki açıklıklı ve altı katlı bir çerçeve sistem üzerinde elemanların kendi arasında belli gruplandırılmasıyla analizler yapılmıştır. Elde edilen sonuçlar bazı çalışmalarla kıyaslanarak sunulan yöntemin artıları ilgililere sunulmuştur. Esfandiary vd. (2016) yayımladıkları makalede çok ölçütlü karar verme mekanizması (İngilizce ismi ile DM: Decision Maker) ve Parçacık Sürüsü Optimizasyonu (İngilizce ismi ile PSO: Particle Swarm Optimization) algoritmasına dayanan ve betonarme çerçevelerin optimizasyonu için kullanacakları yeni bir algoritma (DMPSO) tanıtmışlardır. Yapılan çalışmalarda çok açıklıklı, çok katlı iki boyutlu çerçevelerin yanı sıra üç boyutlu simetrik bir yapı da (2018) ele alınmıştır. Ele alınan yapılar için betonarme elemanlar kendi içerisinde gruplandırılarak, optimum ağırlık ve maliyet için boyut optimizasyonu gerçekleştirilmiştir. Elde edilen sayısal sonuçlar, önerilen DMPSO algoritmasının sonuçların kalitesinden ödün vermeden yakınsama hızı açısından diğer evrimsel ve hibrid evrimsel algoritmalarından daha iyi performans gösterdiği belirtilmiştir. Govindaraj ve Ramasamy (2007) yayımladıkları çalışmada GA'yı kullanarak betonarme yapıların optimal çözümünü aramışlardır. Sundukları çalışmada betonarme elemanlar arasında belli gruplandırmalar yaparak boyut optimizasyonu gerçekleştiren yazarlar, önceki çalışmalarla yaptıkları kıyaslamalar sonucunda makul çözümler elde ettiklerini belirtmişlerdir.

Günümüzde dünya problemlerine gerçekçi ve makul sürelerde çözümler üreten sezgisel ve metasezgisel algoritmalara ve bilgi işlemin üst düzeyde gelişmiş olmasına rağmen çok amaçlı, çok değişkenli ve çok kısıtlı problemlere makul sürelerde çözümler üretmek hala tam anlamıyla istenen noktada değildir. Problemin zorluğuna bağlı olarak artan hesaplama gücünden dolayı, günümüzdeki bilgisayarların sahip olduğu bir mikro işlemcinin gerçekleştiremediği bazı görevler vardır. Bu tür sorunların üstesinden gelmek amacıyla günümüzde otomasyon sistemleri, sesli yanıt sistemleri, e-ticaret, çağrı merkezleri, servis robotları vb. gibi neredeyse her alanda yaygın bir şekilde kullanılan yapay zekâ kavramı yapısal analizde de ilgi görmeye başlamıştır. İnşaat mühendisliğinde yapay zekânın uygulamalarından bazıları şunlardır:

Babiker vd. (2012) yayımladıkları bir çalışmada, basit kirişlerin maliyet optimizasyonu için bir yapay sinir ağı modeli sunmuşlardır. Çalışmada tasarım değişkenleri olarak kiriş genişliği, etkili kiriş derinliği, donatı sayısı ve çapı ele alınmıştır. Yapılan analizler sonrası elde edilen değerlerin karşılaştırmalı çalışmalara göre ümit verici olduğu belirtilmiştir. Senouci ve Abdul-Salam (1998) yaptıkları çalışmada, betonarme kirişlerin derinlik tahmini için bir sinir ağı sistemi geliştirmişlerdir. Yapılan analizler sonrası sinir ağı sisteminin betonarme kirişlerin derinliğini tahmin etmede iyi performans gösterdiğini açıklamışlardır. Aga ve Adam (2015) "NeuroShell-2" yazılımı ile yapay sinir ağı hesaplama modelini kullanarak, beton ve çeliğin maliyetini en aza indirmek amacıyla betonarme düzlem çerçeve yapısı için bir tasarım optimizasyonu yapmışlardır. Çalışmada üç açıklıklı ve iki katlı betonarme bir çerçeve için farklı açıklık ve yük durumları seçilerek analizler gerçekleştirilmiştir. Elde edilen sonuçlar sonrası, çok katlı betonarme çerçeve tasarımı için yapay sinir ağı prosedürünün etkili olduğu kaydedilmiştir. Aynı şekilde Hadi (2003) yayımladığı çalışmada basit kirişlerin optimum tasarımı için bir yapay sinir ağı geliştirmiştir. Yapılan çalışma sonrası yazar, sinir ağlarının birçok yönden mevcut geleneksel yöntemlerden üstün olduğu sonucunu belirtmiştir. Yapay sinir ağlarının elverişli kullanımının başlıca nedenlerinden birisi, her ağın her türlü problemi çözmek için nispeten basit bir denklem setinin çözümünü gerektirmesi, geleneksel yöntemlerin ise daha ayrıntılı denklemler kullandığı yazar tarafından altı çizilen noktalar arasında olmuştur.

Betonarme, kalıplar kullanılarak istenilen yapısal formun oluşturulabilirliği, şartname ve yapım prosedürleri doğru olduğunda dayanıklı bir yapı malzemesi olması ile üstünlük gösterir. Betonarmenin yaygın kullanılmasına ve dolayısıyla birçok alanda araştırmaların merkezi haline gelmesine rağmen konu ile ilgili yapılan çalışmaların genelde elemanların boyut optimizasyonu ile sınırlı kaldığı ve bu tekniklerin de bazen yalnızca eleman bazında gerçekleştiği dikkat çekmiştir. Bendsoe vd. (1993) tarafından yayımlanan çalışmada yapısal optimizasyon problemlerini boyutlandırma, şekil ve topoloji optimizasyonu olarak üç ana kategoride sınıflandırmanın geleneksel olduğu ve bir yapı düzeninin yapının topoloji, şekil ve boyut bilgisini içerdiğinin altı çizilmiştir. Buna bağlı olarak boyut, şekil ve topoloji optimizasyonlarının birlikte ele alındığı tasarım süreçlerinde optimal yapı tasarımının daha etkili şekilde ortaya çıkacağı düşünülebilir. Bu nedenle tüm yapısal optimizasyon dallarının birlikte uygulandığı, daha genel bir optimizasyon sistemine ihtiyaç olduğu aşikârdır.

3. BETONARMA TAŞIYICI SİSTEM PLANI TASARIM PRATİKLERİ

Bir yapı için betonarme taşıyıcı sistem, yapının maruz kaldığı düşey (döşeme, kiriş, kolon, eşya, insan ve kar gibi) ve yatay (deprem, rüzgar kuvveti, toprak ve sıvı basıncı gibi) yükleri zemine ulaştıran ve yapıyı ayakta tutan iskelettir; döşeme, kiriş, kolon, perde ve temel elemanlarından oluşur (Şekil 3.1) (Topçu, 2019).



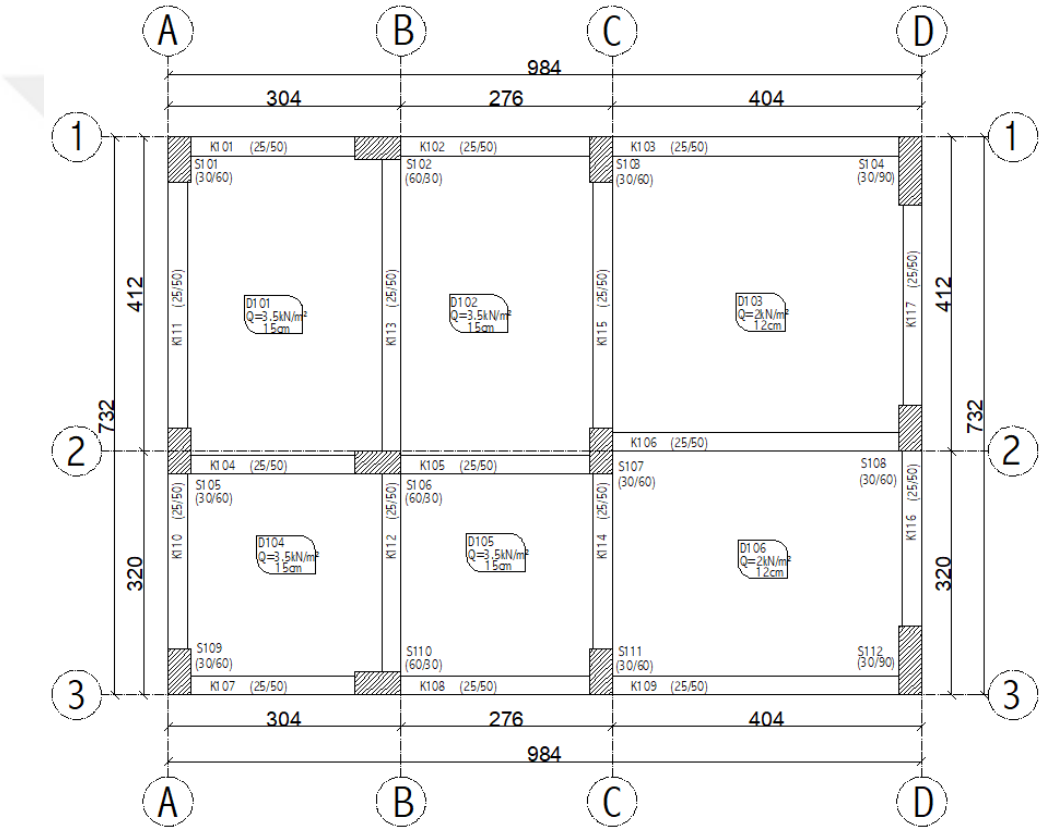
Şekil 3.1. Betonarme taşıyıcı sistem elemanları.

Taşıyıcı sistem planı, taşıyıcı sistem elemanlarının plandaki yerleşimini gösteren çizimdir; kalıp ve kolon aplikasyon planları ile karıştırılmamalıdır. Bu taslak çizimde taşıyıcı sistem elemanlarının boyutları yapısal analiz(ler) sonucu belirlenecek son boyutlardan farklı olabilir. Bu planı oluşturmadaki temel amaç taşıyıcı sistem elemanlarının uygun yerleşimini göstermektir.

Bir taşıyıcı sistem planında aynı hat üzerindeki düşey taşıyıcı elemanların (kolon ve perde) yerini göstermek üzere oluşturulan, ölçülendirme ve konumlandırma için de kullanılan yardımcı çizgilere aks denir. Aks çizgileri, çizim sırasında mühendislerin işlerini

kolaylaştıran hayali çizgilerdir. Şekil 3.2'de bir taşıyıcı sistem planı üzerinde 1, 2, A, B ve C şeklinde isimlendirilerek gösterilen çizgiler aks çizgileridir.

Döşemeler, iki boyutlu olarak değerlendirilen, kalınlığı az, yüzeyleri geniş ve genellikle düşey yüklerin doğrudan etkidiği taşıyıcı elemanlardır. Bunlar basit olarak kirişlerin iki doğrultuda yan yana gelmesi sonucu elde edilmiş gibi görülebilirler. Kendi ağırlıklarıyla birlikte üzerindeki eşya ve insan yükünü taşırlar; bu yükleri kenarlarında kendilerine mesnetlik yapan kirişlere iletirler.



Şekil 3.2. Örnek bir taşıyıcı sistem planında kolon, kiriş ve döşemeler.

Kirişler kendi ağırlıklarıyla beraber üzerlerine gelen duvar ve döşeme yüklerini taşıyan ve bu yükleri düşey taşıyıcı elemanlara (kolon ve perde) aktaran ve yapı mekaniği bakış açısı ile çubuk olarak kabul edilen yapı elemanlarıdır. Betonarme yapılarda döşeme yüklerinin öncelikle kirişlere aktarıldığı ve kesme kuvveti ile moment tesirleri taşıyan kirişlerin bu kuvvetleri kolonlara aktardığı kabul edilir.

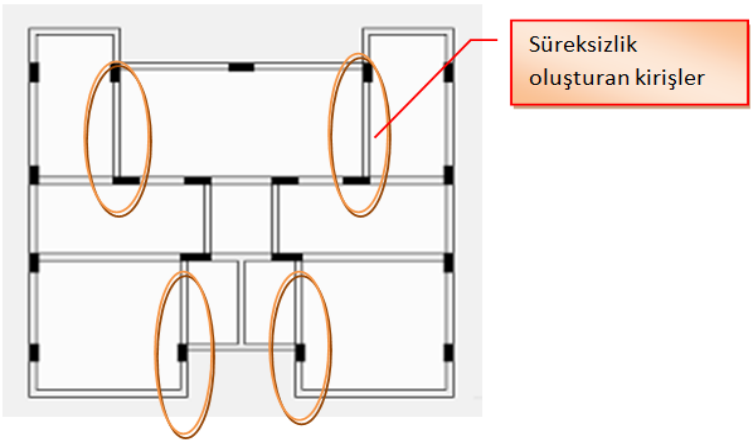
Kirişlerden gelen düşey yüklerle birlikte rüzgâr ve deprem gibi yatay yükleri taşıyan, bu yükleri kolondan-kolona ve sonuçta temele aktaran düşey taşıyıcı elemanlara kolon denir. Her kattan gelen düşey ve yatay yükleri toplayarak temele iletmeleri bakımından yapının esas taşıyıcı elemanlarıdırlar. Bu elemanlarda meydana gelecek herhangi bir hasar çoğunlukla yapının tümünü etkiler. Uygulamada genellikle dikdörtgen kesit olarak tasarlanan kolonlar bazen daire veya L, T ve U gibi poligon şekillerle de karşımıza çıkabilir.

Betonarme taşıyıcı sistemler tasarlanırken yapı fonksiyonları, önceki yapılaşmalar, parsel durumları, estetik kaygılar sebebiyle bazı düzensizlikler oluşur. Bu düzensizlikleri önlemek ve depreme dayanıklı, güvenli yapıların ortaya çıkması bakımından bir taşıyıcı sistem planının sahip olması gereken çeşitli tasarım pratiklerine değinmek gerekir. Yapılan literatür taraması sonucu inşaat mühendisliği alanında kitap, makale ve yönetmeliklerde yer alan ve bir yapının optimum topolojisini, şeklini veya yapı elemanlarının optimal kesitlerini belirlerken göz önünde bulundurulması gereken faydalı pratikler aşağıdaki gibidir ve her maddenin sonunda hangi durumlarda taşıyıcı sistem tasarımında uygun ve cezasız olduğu da açıklanmıştır.

- **Kirişlerin kullanım alanı içinde olması:** Sınırları belli bir yapının betonarme tasarımı yapılırken, mevcut bina sınırlarından herhangi bir yapı elemanının taşması söz konusu olamaz. Bu durum kiriş topolojisine karar verme aşamasında bir kısıt olarak karşımıza çıkar. Tüm kirişlerin yapı kullanım alanı içinde olduğu durumda bir sistem en uygun formdur ve bu durumda bu maddeye ait sunulan stratejideki ceza değeri 0 olacaktır.
- **Kirişlerin serbest ucunun olmaması:** Bir taşıyıcı sistem planında kirişler her iki ucundan kolonlara bağlanarak, gelen düşey yükleri kolonlara aktarırlar veya diğer kirişlere saplanarak kirişler arası yük aktarımını gerçekleştirirler Bir kirişin herhangi bir ucunun boş kalması son derece nadir bir durumdur. Bazı tasarımlar oluşturulurken çok nadir olsa da balkonlarda kirişlerin bir ucunun boş olduğu görülebilir. Ancak bu durum tercih edilmez, çünkü duvar yükü taşımayacağı düşünülen döşemenin serbest ucu, balkon kapatma gibi uygulamalarla duvar yüküne maruz kalabilmektedir; inşaat mühendisi bu tür uygulamalara hazır

olmalıdır. Bir sistemde hiçbir serbest uçlu kirişin olmaması durumunda bu maddeye ait dönecek ceza değeri 0 olacaktır.

- **Mümkün olduğunca çerçevelerin tamamlanması:** Özellikle mimarların ve müteahhitlerin estetik kaygılarından dolayı önemli hacimlerde (salon, mutfak vb.) sarkan kirişler istenmez. Bu durum kirişlerde süreksizliğe neden olabilir (Şekil 3.3). Güvenli bir yapı için yüklerin en kısa yoldan kolonlara, kolonlardan da temele aktarılması gerekir. Ancak süreksiz kirişler, yüklerin yapı içerisinde dolaşmasına neden olarak sağlıklı bir biçimde temele aktarılmalarını önler. Bu durum deprem sırasında gerilme yığılmalarına neden olur ve sonuçta yapı tehlikeye girer. Bunun yanı sıra kirişlerin kaldırılması büyük döşemelerin oluşumuna sebebiyet verebilir. Bunun sonucunda döşemeler yeterince rijit davranamaz, yatay kuvvetlerin kolon ve perdelere dağılımı düzensiz olur. Bu şekilde olumsuz durumlara yol açan süreksiz kirişlerin yapımından kaçınılmalıdır. Bir tasarımda çerçeveler ne kadar tamamlanırsa ceza değeri de o derece az çıkacaktır; tüm çerçevelerin sürekli olduğu durumda ise ceza değeri 0 olacaktır.



Şekil 3.3. Süreksiz kirişler (Topçu, 2019).

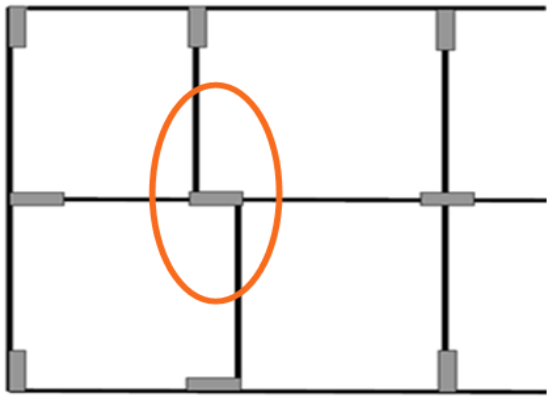
- **Tam duvarların altında kiriş bulunması:** Özellikle ıslak hacim olarak adlandırılan banyo ve tuvalet gibi yerlerde, genellikle kiriş ağının çok sık olmasından kaynaklı olarak üzerinde yarım duvar olan kirişlerden uygun görülenleri kaldırılabilir. Ancak üzerinde tam duvarların yer aldığı kirişlerin (özellikle yapı sınırlarında bulunan kirişler) kaldırılması mühendislerce büyük bir

kusur olarak değerlendirilir. Bir planda var olan tüm tam duvarların altında kiriş olması durumunda sistemin alacağı ceza 0'dır.

- **Mümkün olduğunca yarım duvarların altında da kiriş olması:** Islak hacimlerde oluşan sık kiriş ağından kaynaklı olarak, üzerinde yarım duvar bulunan kirişlerden uygun görülenlerin kaldırılması mümkün olabilir. Ancak mühendis, gerekmedikçe üzerinde yarım duvarların da bulunduğu kirişleri kaldırmayı tercih etmemelidir. Bir planda var olan tüm yarım duvarların altında kiriş olması durumunda sistemin alacağı ceza 0'dır.
- **Kirişlerin mümkünse kıymetli alanlar içerisinden geçmemesi:** Yapı tasarlanırken görece daha kıymetli hacimler meydana gelir. Örneğin; salon veya mutfak gibi alanların kıymet dereceleri, banyo veya tuvalet tarzı alanların kıymet derecelerinden çoğunlukla fazladır. Bu nedenle kiriş tasarımı gerçekleştirileceği sırada meydana gelebilecek diş oluşumlarının, kıymet derecesi daha az olan hacimlere doğru yapılması tercih edilir. Özellikle müteahhitlerin, salon gibi kıymetli alanlarda gerek kolonlar gerekse kirişler tarafından oluşabilecek diş oluşumlarını tercih etmedikleri, mühendisler tarafından bilinen önemli bir detaydır. Bir sisteme ait tüm kirişler, kıymet dereceleri az olan hacimlere doğru kaçıklık yaptıkça adayların bu maddeden alacağı ceza 0'a yakın olacaktır.
- **Komşu açıklıkların birbirinden çok farklı olmaması:** Bir kiriş ağında oluşacak açıklıkların birbirinden çok farklı değerde olmaması istenir. Komşu açıklıkların birbirinden çok farklı olması durumunda büyük miktarlarda kesme kuvveti sıçramaları ortaya çıkar. Kısa kirişlerin üzerinde büyük değerlerde kesme kuvvetleri birikir. Mümkün olduğunca komşu açıklık değerleri birbirine yakınlaştırılarak oluşabilecek kesme kuvveti yığılmalarının önüne geçilmelidir. Bir planda herhangi bir aks üzerindeki bütün kirişlerin uzunluğunun aynı olması durumunda, adayların bu maddeden alacağı ceza 0'dır.
- **Merdiven boşluğundan kiriş geçmemesi:** Bir yapı konfigürasyonunda istenilen her yere eleman yerleştirmek mümkün değildir. Örneğin bir salonun ortasına kolon yerleştirmek mühendislik anlamında mümkün olsa bile, iyi bir tasarım bakış açısına uymaz. Genellikle proje üzerindeki bazı bölgeler herhangi bir eleman koymak için yasaklı bölgelerdir. Bu şekilde bir merdiven boşluğundan bir kirişin geçmesi de hiçbir açıdan mümkün değildir ve projelendirme sürecinde kiriş koymak için

yasaklı bir bölgedir. Bir sistemde yasaklı olarak tanımlanan bölgelere herhangi bir eleman gelmediği durumda sistemin bu maddeden alacağı ceza 0'dır.

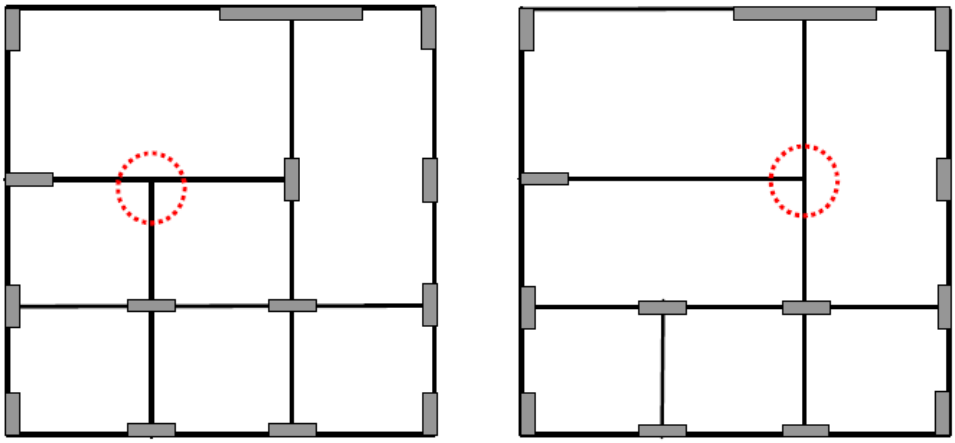
- **Çakışık eksenli kirişlerden oluşan sürekli kiriş hatları mümkün olmadığı takdirde çakışık eksenli olmayan kirişler ile sürekli kiriş hatlarının oluşturulmaya çalışılması:** Bazı projelerde çakışık eksenli kirişlerden sürekli kiriş hatları oluşturmak mümkün olmaz. Bu gibi durumlarda eksenleri çakışık olmayan kirişler arasında uygun kolon boyutlarının seçimi ile sürekli kiriş hatlarının oluşturulması gerekir (Şekil 3.4). Çakışık eksenlerin oluşturulmasıyla sistemin bu maddeden alacağı ceza değeri 0 olacaktır.



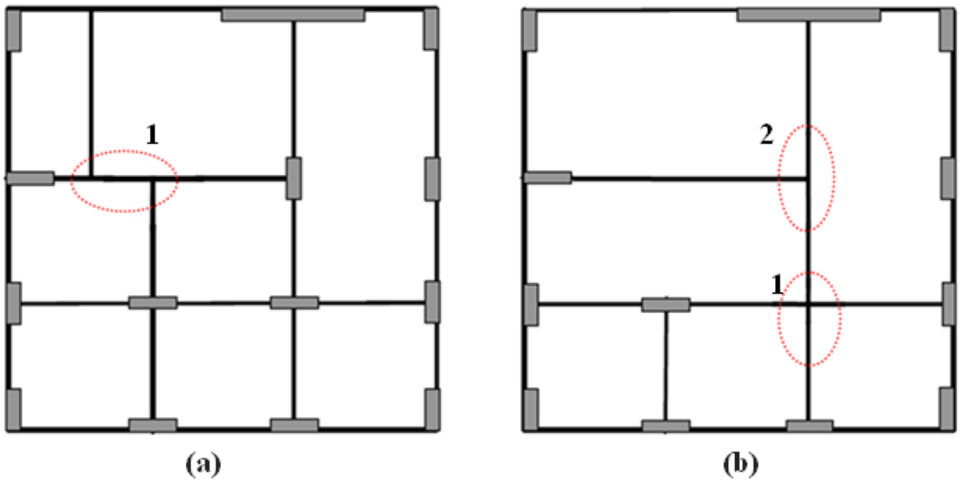
Şekil 3.4. Eksenleri çakışık olmayan kirişler ile sürekli kiriş hatlarının oluşturulması (Topçu, 2019).

- **Mümkün olduğunca saplama kiriş oluşumlarının engellenmesi:** Normal olarak kirişin her iki ucu kolona oturmalıdır, böylece kiriş yükü en kısa yoldan kolonlara aktarılmış olur. Kolonun olmaması saplama kiriş düzensizliğini oluşturur (Şekil 3.5). Kirişlerden biri diğerine oturur. Yükler doğrudan kolona aktarılamaz; kirişten kirişe aktarılmış olur. Kimin kimi taşıdığı kiriş rijitliklerine bağlıdır ve her zaman tam belirgin değildir. Bu düzensizliği çoğu zaman önlemek mümkün olmaz. Bir sistemde hiç saplama kirişin olmaması durumunda bu maddeye ait ceza değeri 0 olacaktır.
- **"Saplama kirişe saplanan kiriş" oluşumuna izin verilmemesi:** Her kiriş-kiriş birleşim noktasına kolon konulduğunda birbirine çok yakın kolonlar ve dolayısıyla çok küçük açıklıklı kirişler oluşabilir veya kolon merdiven, pencere veya kapıya denk gelebilir. Bu nedenle bazı kirişler saplama yapılmak zorunda kalınır. Ancak,

elden geldiğince saplama kirişlerden kaçınmak gerekir. En kötü ve kesinlikle yapılmaması gereken durum ise, saplaması olan bir kirişin başka bir kirişe saplanmasıdır. Şekil 3.6 (a) ve (b)'de görülen "1" numaralı kirişler saplama kiriş iken Şekil 3.6 (b)'deki "2" numaralı kiriş saplamanın saplaması kiriştir. Bu uygulama sonrası kiriş yükü dolaylı yollardan birikerek kirişten kirişe aktarılır ve taşıyan kirişte burulma momenti ve sehim oluşur. Uygulamada yapılan bu tür kirişler Şekil 3.7'de görülmektedir. Bir sistemde hiç saplama kirişe saplanan kirişin olmaması durumunda bu maddeye ait ceza değeri 0 olacaktır.



Şekil 3.5. Saplama kirişler (Topçu, 2019).

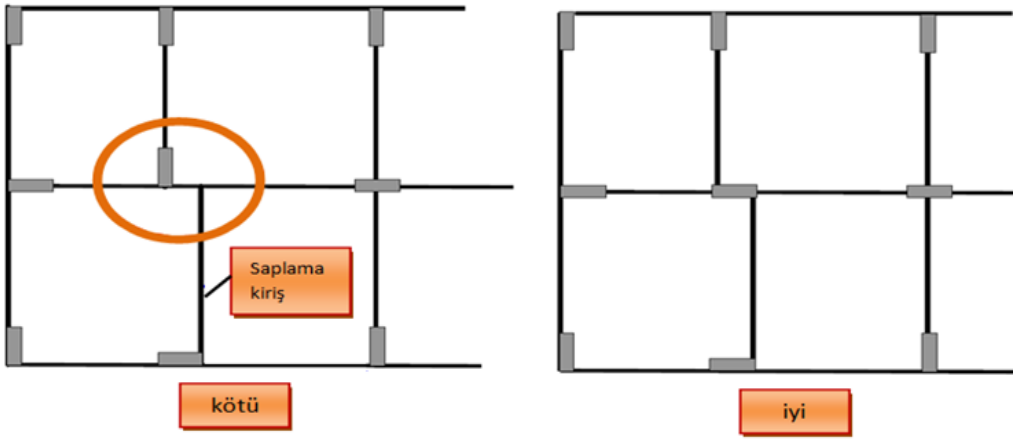


Şekil 3.6. Saplaması olan bir kiriş, başka bir kirişe saplanmamalıdır (Topçu, 2019).

- **Saplama kiriş oluşumunun engellenemediği durumlarda, saplama noktasının kirişin mesnetlendiği kolonlara çok yakın olmaması:** Kirişin kolon civarında başka bir kirişe saplanması çok kötü bir durumdur (Şekil 3.8). Bu durumun meydana gelmesi sonucu kesme ve burulma etkileri aşırı düzeylere varır (Şekil 3.9). Bir saplama kirişin kendisine en yakın kolon ile arasındaki mesafe verilen limit değerin üstünde olduğu sürece sistemin bu maddeden alacağı ceza değeri 0 olacaktır.



Şekil 3.7. Uygulamada yapılan saplamanın saplaması kirişler (Topçu, 2019).



Şekil 3.8. Kolona yakın saplanan kiriş (Topçu, 2019).

- **Birbirine çok yakın olan kolonlardan uygun görülenlerin kaldırılması:** Bir yapı tasarımında birbirine çok yakın olan kolonlardan uygun görülenleri kaldırılabilir. Ancak bu işlem çakışık eksenli olmayan kirişlere yol açarsa ilgili kolonun saplama düğümünü de içine alacak şekilde büyütülmesi, mümkünse perdeye dönüştürülmesi

fikri değerlendirilmelidir. Bir sistemde kolonlar arası mesafeler belli bir limit değerin altında değilse adayların bu maddeden aldığı ceza 0'dır.

- **Kiriş açıklığının çok büyük olmaması:** Konut tipi yapılarda oluşabilecek maksimum betonarme kiriş açıklığı yaklaşık 6~7 m'dir. Oluşan açıklık değerlerinin bu sınır değerleri geçmesi durumunda kirişli döşemelerde döşeme kalınlığı ve buna bağlı olarak kiriş derinliğinin artması gerekir. İzin verilen ve ekonomik olan kat yüksekliği değerleri ile çalışıldığında kiriş derinliğinin serbestçe artırılması mimari kısıtlar sebebiyle (kapı boşluğunun kapanması vb.) çoğu zaman mümkün olmaz. Eğer bir sistemde herhangi bir kirişin uzunluğu minimum ve maksimum olarak belirlenen limit değerler arasında ise adayların bu maddeden aldığı ceza 0'dır.

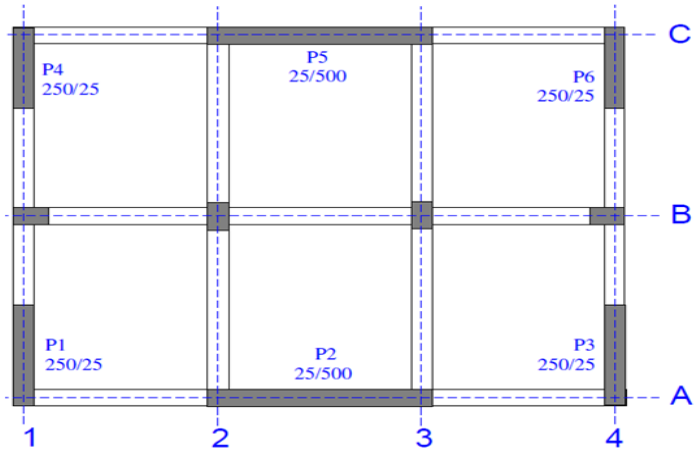


Şekil 3.9. Uygulamada yapılan ve kolona çok yakın olan saplama kiriş (Topçu, 2019).

- **Kolonların oturma alanı dışına çıkmaması:** Betonarme plan tasarımı yapılırken mevcut bina sınırlardan herhangi bir yapı elemanının taşması söz konusu olamaz. Kolon ve perdelerin temel ile birleştiği noktaların yapı oturma alanının dışına çıktığı durumlarda, ilgili idare (belediye vb) yasal işlem başlatacaktır. Bir sistemde tüm kolonların oturma alanı içerisinde olması durumunda bu maddeden dolayı oluşacak ceza değeri 0 olacaktır.
- **Kolonların kapı ve pencere boşluklarına taşmaması:** Proje tasarım sürecinde mühendisin karşılaştığı diğer bir mimari kısıt da kapı ve pencere boşluklarına kolonların denk gelmemesi durumudur. Bu durum kolon topolojisi sürecinde göz

önüne alınması gereken önemli bir kısıttır. Yasaklı olarak tanımlanan bölgelere kolonların taşmaması bu maddeden dönecek ceza değerini 0 yapar.

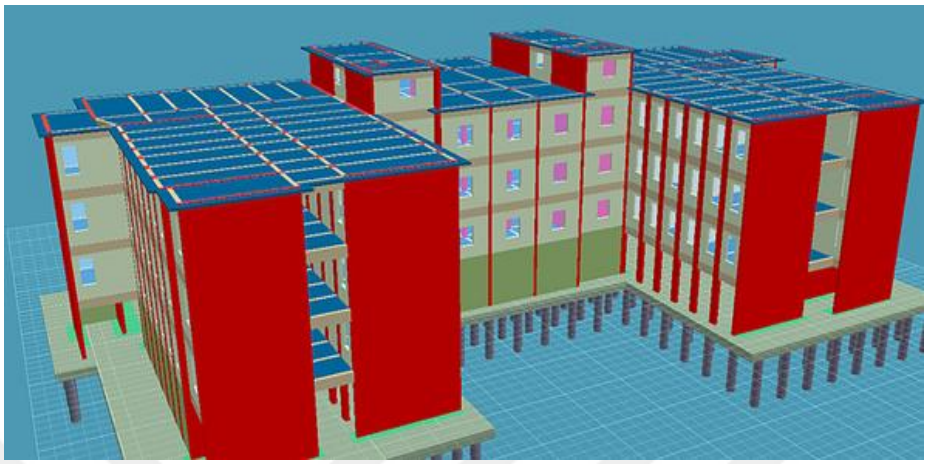
- **Kolonların mümkünse kıymetli alanlar içerisinden geçmemesi:** Kiriş topolojisi başlığı altında ayrıca değinildiği üzere yapı içerisinde birbirine kıyasla daha kıymetli hacimler vardır. Özellikle salon gibi kıymetli hacimlerde oluşabilecek diş oluşumlarının önüne geçmek estetik yapı tasarımının önemli bir ölçütüdür. Bir sisteme ait tüm kolonların komşu olduğu en önemsiz alana, olabilecek en küçük hacimde taşması durumunda adayın bu maddeden alacağı ceza 0 olacaktır.
- **Perde konfigürasyonu iki farklı yönde en az ikişer adet perde olacak şekilde düzenlenmelidir:** Yapının deprem ve rüzgâr gibi yatay kuvvetlere karşı dayanımını artırmak için her iki yönde yeterli sayıda perde düzenlemek gerekir. Perdelerin yapının çok fazla yatay yer değiştirmesini önlemek, yatay kuvvetten oluşan kesme kuvvetlerini almak, kolon ve kirişlerin rahatlamasını sağlamak, yapının burulmasını önlemek gibi çok önemli görevleri vardır. Her deprem yönünde en az 2 adet olmak üzere toplamda en az 4 perde bulunmalıdır şeklinde faydalı bir öneri vardır (Şekil 3.10). Bir sistem için bir yönde en az iki perdenin oluşması durumunda oluşacak ceza değeri 0 olacaktır.



Şekil 3.10. Bir sistemde her iki deprem yönünde yerleştirilen perdeler.

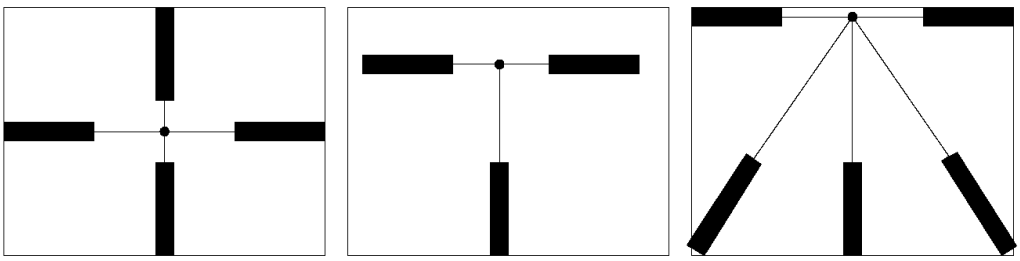
- **Perdeler mümkün olduğunca taşıyıcı sistem planında dış akslara yerleştirilmelidir:** Perdeler uzun elemanlar olduğu için rijitlikleri fazladır. Bundan dolayı mümkün mertebe binanın en dış noktalarına yerleştirilmelidir (Şekil 3.11). Bu şekilde burulma kuvvetlerinden doğan ek tesirlerin de önüne geçilir. Perdeler

merkeze yaklaştıkça yapıya olumlu etkisi azalır. Bir sistemde olan tüm perdelerin dış akslara yerleşmesi sonucunda sistem için bu kısıt 0 olarak cezalandırılacaktır.



Şekil 3.11. Dış akslara yerleştirilen perdeler.

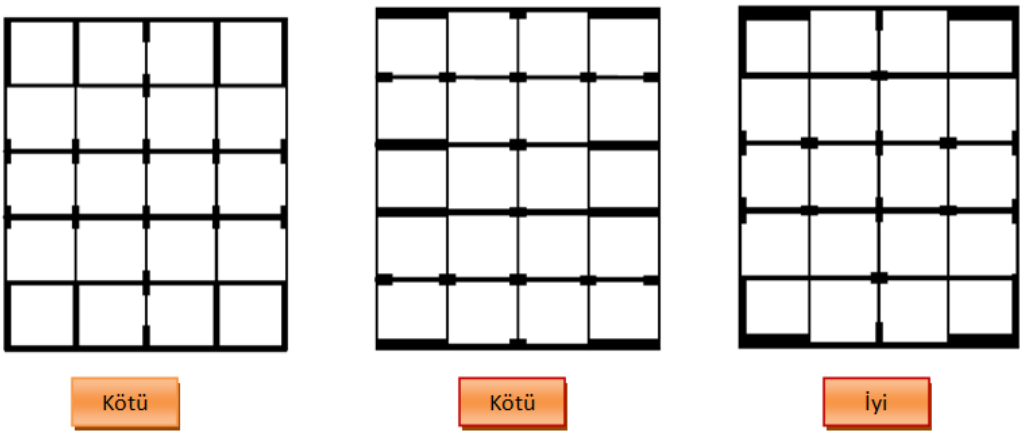
- **Planda perdelerin uzun kenarı doğrultusundaki çizgiler (etki çizgileri) bir noktada kesişmemelidir:** Yapı tasarımında yerleştirilecek perdelerin uzun kenar doğrultusundaki çizgileri bir noktada kesişirse, yapı yatay yükler altında o nokta etrafında burulmaya yatkın hale gelir ve deprem sırasında olumsuz yapı davranışları meydana gelebilir (Şekil 3.12). Bu yüzden kolon ve perde topolojisi belirlenirken perdelerin olabildiğince simetrik ve sistem çizgilerinin bir noktada çakışmasını engelleyecek şekilde yerleştirmek güvenli bir tasarımı ortaya çıkarır. Perde sistem çizgileri birden fazla noktada kesişen bir plan için ceza değeri 0 iken, sistem çizgileri tek bir noktada kesişen bir planın ceza değeri 1 olacaktır.



Şekil 3.12. Perdelerin sistem çizgilerinin bir noktada birleşmesi engellenmelidir.

- **Kiriş-kolon aksı elden geldiğince çakıştırılmalıdır:** Betonarme yapı tasarımında kirişlerin bağlandıkları kolonlarla çakışık eksenli olması beklenir. Kiriş

eksenlerinin kolon eksenleriyle çakışmaması durumunda, kirişler tam olarak kolonlara denk gelmedikleri için kolon sınırlarından taşarlar. Bu düzensizlik güvenli yapı tasarımını engelleyen olumsuz bir durumdur. Bir sistemde tüm kirişler bağlı oldukları kolonlara tam oturuyor ise adayın bu maddeden alacağı ceza 0 olacaktır.



Şekil 3.13. Bir yapının her iki deprem yönünde yerleştirilen perdelere göre davranışı (Topçu, 2019).



Şekil 3.14. Bir yöne göre daha rijit davranış gösteren bir yapının deprem sırasındaki durumu.

- **Kolonların ortogonal iki yöndeki toplam rijitlikleri eşit veya yakın olmalıdır:** Tüm kolon ve perdeleri uzun kenarları yapının sadece bir yönünde olacak şekilde yerleştirmek sık karşılaşılan bir düzensizliktir. Bu durumda yapı yatay kuvvetlere

karşı bir yönde güçlü diğer yönde ise zayıf davranır (Şekil 3.13). Deprem yer hareketinin planda tüm doğrultularda etkili olduğu dikkate alınarak taşıyıcı sistem elemanlarının tercihen birbirine dik iki asal doğrultuda düzenlenmesi ve birbirine yakın dayanım ve rijitliğe sahip olması esastır. Aksi takdirde bir yöne göre daha rijit davranış gösteren yapılar, deprem sırasında zayıf yönden gelen kuvvetler etkisinde yıkılabilirler (Şekil 3.14). Bir tasarımda ortogonal iki yöndeki toplam rijitlik oranları ne kadar küçük olursa ceza değeri de o derece az çıkacaktır ve oranın 1 olması durumunda ise oluşacak ceza miktarı 0'dır.

Yapı tasarımı sırasında göz önüne alınan bazı tasarım pratikleri yukarıdaki gibi belirtilmiştir. Bunun yanı sıra yapıların tasarımı mimari ve taşıyıcı sistem tasarımı olarak iki evreden oluşur (Doğangün 2008). Ülkemizde ilk olarak mimari proje ortaya çıktıktan sonra yapının mimari projeye bağlı betonarme projeleri oluşturulur. İnşaat mühendisliği açısından tipik bir yapısal tasarımın ilk ve en önemli adımı olan "Ön İşlem Aşaması" yapının bilgisayar modelinin hazırlanması safhasıdır. Her ne kadar yazılımların kullanıcılara sunduğu grafik ortamdaki veri girişi, kontroller ve uyarılarla donatılmış menüler ile kolaylaştırılmaya çalışılsa da bu aşamanın yükü kullanıcının yani inşaat mühendisinin omuzlarındadır. Yapı geometrisinin ve yüklerinin kullanılan yazılımın imkânları doğrultusunda sisteme tanıtıldığı bu aşama, BDT'deki en önemli hata kaynaklarından biridir. Yapısal tasarımın ikinci aşaması "Analiz Aşaması"dır. Bu safhada kullanıcının hazırlamış olduğu yapısal model bilgisayar tarafından analiz edilerek yapısal deplasmanlar ve bunlara bağlı olarak yapı elemanlarının etkileri hesaplanır. Üçüncü safha ise "Son İşlem Aşaması"dır. Analiz sonrası işlemleri içeren bu safha, analiz sonuçlarının değerlendirilmesi, eleman donatı hesapları ve pafta çizimlerinin hazırlanması gibi adımları içerir (Kubin, 1994).

Deneme-yanılma yöntemiyle her defasında analize girip, daha sonra değerlendirme aşamasıyla ilerleyen betonarme projelerin ortaya çıkma süreci, bu bakımdan birçok dezavantajı beraberinde getirir. Her defasında sistemi düzeltip analiz yaptırması sonucu mühendisin önemli zaman kayıplarına uğrayabileceği söylenebilir. Dahası tasarım kısıtlarını sağlayarak elde edilecek bir projenin maliyet açısından da optimum olup olmadığını test etmek zor olacaktır. Çünkü günümüz rekabet ortamında çizilecek projenin çok fazla zaman almaması gerektiğinden, başka alternatiflerin değerlendirilmesi çoğu

zaman mümkün olmaz. Evet, mühendis belki müteahhitlerin istediği minimum donatı oranını piyasa şartlarında yerine getirecektir. Ancak bu durumun sağlanması çoğu zaman optimum şartların sağlandığı anlamına gelmeyebilir.

Bu şekilde gergin bir süreci kapsayan, sonrasında da optimum sonucun elde edilip edilmediği kesin olarak bilinemeyen betonarme projelerin tasarım süreci, günümüz bilgisayar teknolojisi ile farklı bir boyutta, zaman kayıplarının önüne geçerek, daha uygun sonuçların ortaya çıkacağı şekilde meydana gelebilir.

4. OPTİMAL TASARIM

Mühendislik yararlı prensipler oluşturmak için doğanın prensiplerinin uygulandığı bir meslektir. Bir makine mühendisi yeni bir motor veya robot, bir inşaat mühendisi bir köprü veya bina, bir kimya mühendisi bir damıtma kulesi veya kimyasal bir ilaç tasarlar (Parkinson vd., 2013). Bunun gibi diğer mühendislik alanlarında birbirinden bağımsız birçok tasarım söz konusu olmaktadır. Mühendislikte tasarımlara rehberlik eden temel ilkeler işlevsellik, ekonomi, güvenlik ve estetiktir ve bir mühendis tasarımın tüm bu yönlerini elde etmek için elindeki kaynakları en iyi şekilde kullanmalıdır. Bu ilkeler kendi içerisinde tasarımdan tasarıma değişkenlik gösterir. Örneğin; bir makine mühendisi üreteceği bir parçanın en düşük yarıçapta olmasını hedeflerken, bir inşaat mühendisi tasarlayacağı bir köprünün en düşük maliyette olmasını baz alabilir. İşte bir üretim sürecinde belirli zaman ve kaynak kısıtlamaları altında en iyi tasarımı bulma sürecine mühendislik optimizasyonu denir. Optimizasyon karar verme süreçlerini hızlandırmak ve karar kalitesini arttırmakta kullanılarak gerçek hayatta karşılaşılan problemlerin etkin, doğru ve gerçek zamanlı çözümünü sağlar. Dahası optimizasyon, ekonomik açılardan getirdiği kazançların yanında müşteri, işveren ve çalışanların tercih ve kısıtlarının karar sürecinde yer almasında ve sistemde yer alan kaynakların kalitesinin yükseltilmesinde de aktif olarak başvurulan bir yöntem olarak kullanılmaktadır (Turkay, 2004).

Günümüz dünyasında hammadde kıtlığının artan bir şekilde gerçekleşmesi ve geleneksel enerji kaynaklarımızın hızla tükenmesi hafif, verimli ve düşük maliyetli yapılara olan talebin artmasına, dolayısıyla mühendislerin özellikle ağırlık ve maliyet optimizasyonu tekniklerini bilme ihtiyacını doğurmaktadır (Haftka ve Gürdal, 1992). Ayrıca rekabetçi ortamda olan bir mühendis, sadece nominal düzeyde çalışan bir tasarımla ilgilenmekle kalmayıp, aynı zamanda en iyi tasarımı üreteceği şekilde optimizasyon tekniklerini ele almalıdır.

Optimizasyonu problemlerde kullanabilmek için çeşitli niteliklerin karşılanması gerekir. İlk olarak, ulaşılmak istenen hedefi hesaplamak için nicel bir modelimiz olmalıdır. Isı transferini en üst düzeye çıkarmak istiyorsak, farklı tasarım konfigürasyonları için ısı transferini hesaplayabilmeliyiz veya maliyeti en aza indirmek istiyorsak, maliyeti

hesaplayabilmeliyiz. Bazen böyle nicel modeller elde etmek kolay değildir. Ancak elimizde bizi sonuca götürmesi için matematiksel bir modelin olması şarttır. Tasarım sorununun geçerli ve doğru bir modelini elde etmek, optimizasyonda en önemli adımdır. İyi bir model elde edildiğinde, optimizasyon sonuçları genellikle hızlı bir şekilde gerçekleştirilebilir (Christensen ve Klarbring, 2008). Klasik tasarım prosedürü, kabul edilebilir bir tasarımı, yani kısıtlamaları karşılayan bir tasarımı bulmayı amaçlar. Genel olarak bir problem için kabul edilebilir birkaç tasarım var olduğundan optimizasyonun amacı mümkün olan en iyi tasarımı ortaya koymaktır. Bu nedenle, farklı tasarımları karşılaştırmak için bir kriter seçilmelidir. Bu kriter tasarım değişkenlerinin bir fonksiyonu olarak ifade edildiğinde, amaç fonksiyon olarak karşımıza çıkar. Amaç fonksiyonu genel olarak fiziksel veya ekonomik kaygılarla belirlenir. Bununla birlikte bir amaç fonksiyonun seçimi asla önemsiz değildir, çünkü belirli bir kritere göre optimal tasarımın ne olduğu, başka bir kritere göre kabul edilemez olabilir. Tipik olarak performans-maliyet veya performans-güvenilirlik dengesizliği vardır, dolayısıyla amaç fonksiyonun seçimi tüm tasarım sürecinde en önemli kararlardan biridir. Birden fazla kriterin yerine getirilmesi gerekiyorsa, çok amaçlı bir optimizasyon problemimiz var demektir ve bu, birkaç amaç fonksiyonun ağırlıklı toplamı olan bir maliyet fonksiyonu dikkate alınarak yaklaşık olarak çözülebilir (Astolfi, 2006).

Bir modelin yanı sıra, düzeltilmesi serbest olan bazı değişkenler de olmalıdır. Yani tasarımı tanımlayan ve optimizasyon sırasında değiştirilebilen bir denklem veya vektör süreç süresince kaçınılmazdır. Üretilecek bir borunun kalınlığı veya uzunluğu o tasarımın değişkenleri olabilir. Tasarım modeli için karşılanması zorunlu olan sınırlar da söz konusudur. Kısıtlamalar olarak karşımıza çıkan bu sınırlar, eşitsizlik kısıtlamaları veya varsa eşitlik kısıtlamaları olarak ikiye ayrılır. Açıkçası bir problem için optimum çözümü bulmak, genellikle herhangi bir kısıtlama olmaksızın gerçekleştirilemez. Örneğin; kullanılacak malzeme miktarında herhangi bir sınırlama yoksa yapı sınırsız olarak sertleştirilebilir ve iyi tanımlanamamış bir optimizasyon problemi ile karşı kaşıya kalırız.

Bütün bu bilgiler ışığında ele alınan optimizasyon problemleri, tamsayı veya gerçek değerleri alabilen, bir veya daha fazla bağımlı değişkene dayanarak, bir veya daha fazla nesnel fonksiyonu en aza indirmeyi (minimizasyon) veya en yüksek değere ulaştırmayı (maksimizasyon) amaçlamaktadır. Dolayısıyla standart bir optimizasyon problemi

minimize veya maksimize etmeye çalıştığımız amaç fonksiyonu, amaç fonksiyonunun değerini etkileyen değişkenler kümesi ve kısıtlar olmak üzere üç temel bileşenden oluşur. Bir optimizasyon problemi Çizelge 4.1'de görüldüğü gibi ifade edilebilir:

Çizelge 4.1. Optimizasyon bileşenleri.

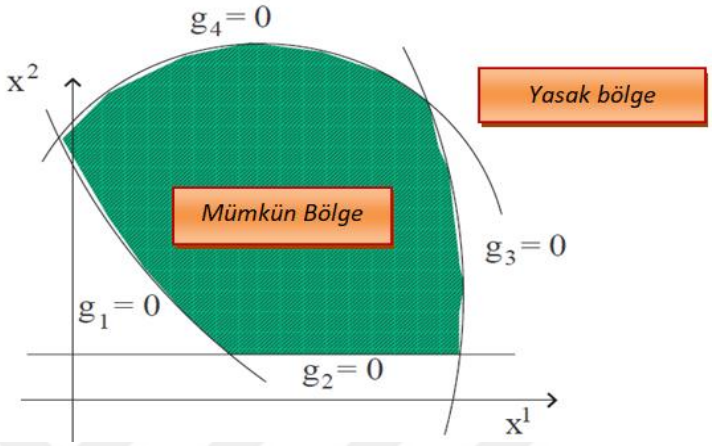
$x = (x_1, x_2, x_3, \dots, x_n)$	Tasarım Değişkenleri	(4.1)
$f(x)$	Amaç Fonksiyonu	(4.2)
$g_j(x) \leq 0$ $j = 1,2,3, \dots \dots, m$	Eşitsizlik kısıtı	(4.3)
$l_j(x) = 0$ $j = 1,2,3, \dots \dots, p$	Eşitlik kısıtı	(4.4)

Çizelge 4.1'de verilen ifadeler incelendiğinde Denklem 4.1'deki x değişkenine tasarım vektörü, Denklem 4.2'deki $f(x)$ minimize veya maksimize etmeye çalıştığımız amaç fonksiyonu, Denklem 4.3'teki $g_j(x)$ eşitsizlik ve Denklem 4.4'deki $l_j(x)$ eşitlik kısıtlarıdır. Değişkenlerin sayısı olan n ile $p + m$ kısıtlamalarının sayısının ilişkili olması gerekmez. $p + m = 0$ ise probleme kısıtsız optimizasyon problemi denir.

Bir tasarım vektörü olan x, tüm $g_j(x)$ ve $l_j(x)$ kısıtlamalarını karşılıyorsa buna uygulanabilir çözüm denir; aksi takdirde çözüm mümkün değildir. Tüm uygulanabilir noktalar veya vektörler, arama alanındaki mümkün (uygun veya olumlu da denir) bölgeleri oluşturur (Şekil 4.1).

Şekil 4.2'de optimizasyon tekniğinin birçok kola ayrıldığı görülür. Optimizasyon problemleri maksimize veya minimize etmeye çalıştığımız amaç fonksiyonuna bağlı olarak tek veya çok amaçlı olabilir. Bunun dışında hiçbir amacın olmadığı ekstrem bir durumda, sadece kısıtlamalar vardır. Böyle bir soruna fizibilite sorunu denir ve herhangi bir uygulanabilir çözüm, bu tür problemlere en uygun çözümdür.

Eğer herhangi bir kısıtlama yoksa (yani $p + m = 0$) sorun kısıtsız bir optimizasyon problemi olur; $p \geq 1$ veya $m \geq 1$ durumlarında ise sorun kısıtlı problem haline gelir.



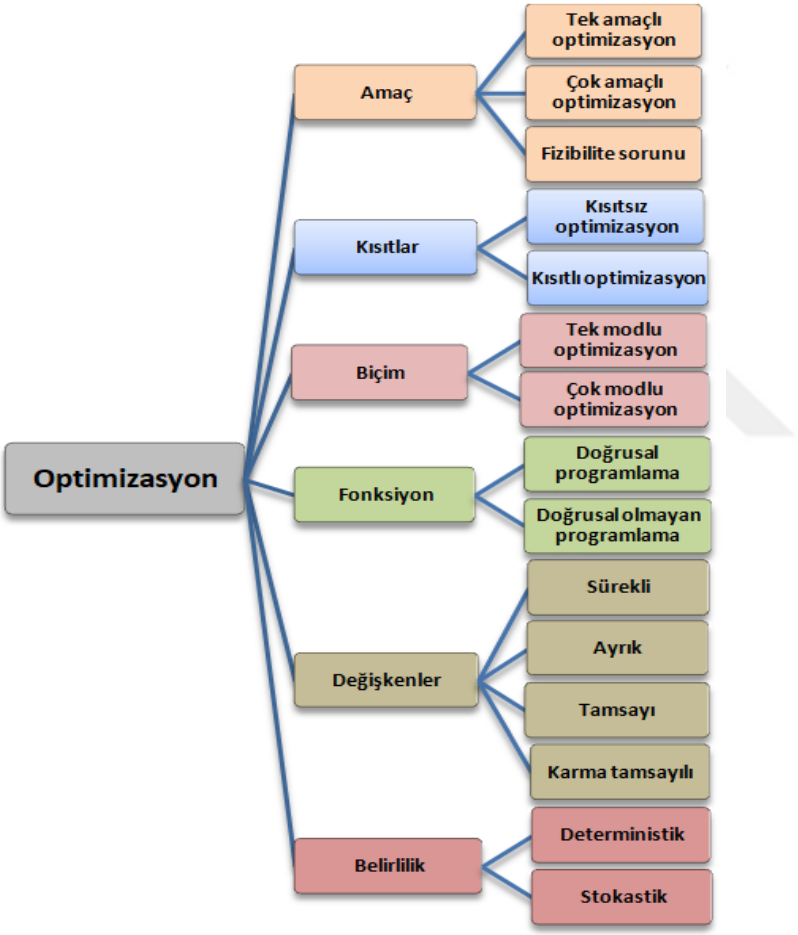
Şekil 4.1. Yalnızca eşitsizlik kısıtlamalarının olduğu, iki boyutlu bir tasarım alanında uygulanabilir bölge.

Optimizasyon problemleri doğrusal veya doğrusal olmayan problemler olarak ikiye ayrılır. Doğrusal problemlerde genellikle amaç fonksiyonu ve sistem modeli matematiksel denklemlerle gösterilir. Ancak doğrusal olmayan problemlerde bu ifadeler herhangi bir denklem ile belirlenemeyecek kadar karmaşıktır. Günümüz mühendislik problemleri doğrusal olmayan problemlere örnek olarak gösterilebilir.

Optimizasyon problemlerinde amaç fonksiyonunda kullanılan değişkenler, sürekli değişkenler ise yani aralarında boşluk olmayan bir dizi gerçek değerden oluşuyorsa sürekli optimizasyon, değişkenlerin bir kısmı veya tamamı ayrık bir değerler kümesinden seçilirse ayrık optimizasyon olur. Sürekli optimizasyon süreklilik varsayımı nedeniyle kalkülüs tekniklerinin kullanılmasına izin verir. Bu yüzden uygulanması kolaydır. Ancak ayrık optimizasyon sürekli optimizasyona göre daha karmaşık bir alandır. Burada değerlerin belli bir kümeden seçilmesi birçok zorluğu gün yüzüne çıkarır. Ele alınan mühendislik problemleri genellikle ayrık optimizasyon problemleridir. Ayrıca değişkenler aldıkları değerlere göre tamsayı veya ondalıklı olarak da değişebilir.

Başka bir sınıflandırma da, problemin belirliliğine göre yapılır. Eğer formülasyonda herhangi bir rastgelelik yoksa buna deterministik problemler denir ve amaç fonksiyonunun

eğimi veya türevi kullanılarak global optimuma ulaşılır. Bununla birlikte değişkenlerde veya fonksiyon formlarında belirsizlik varsa, optimizasyon olasılık dağılımını ve beklentisini içerir. Bu tür uygulamalara genellikle stokastik optimizasyon denir. Stokastik tabanlı yöntemler yeterli sayıda rastgele çözüm ve optimizasyon adımları ile global optimumu bulmayı amaçlar. Gerçek dünya problemlerinde var olan bir problemin deterministik yöntemlerle çözülmesi pek mümkün değildir. Çünkü problemler belirli bir formülasyonla ifade edilemeyecek kadar karmaşıktır. Bu yüzden stokastik yöntemler günümüzde sıklıkla başvurulan yöntemler olarak karşımıza çıkmaktadır.

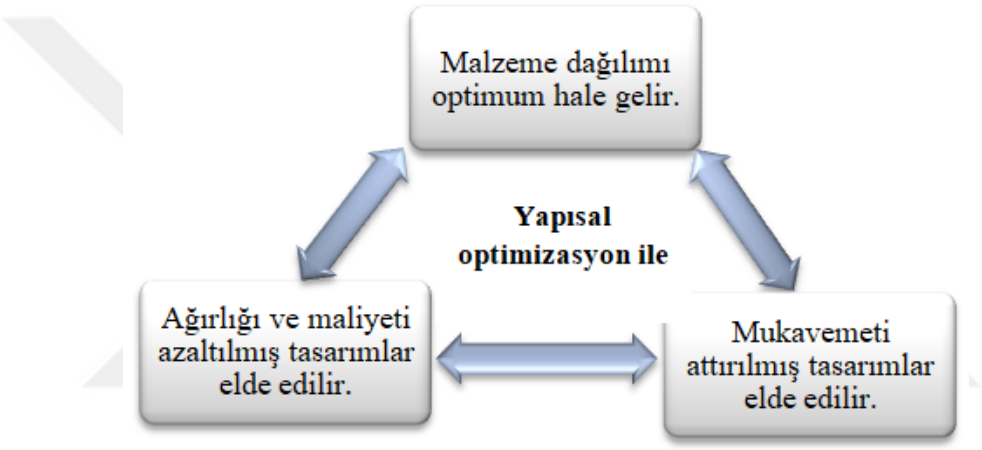


Şekil 4.2. Optimizasyon problemlerinin sınıflandırılması.

4.1. Yapısal Optimizasyon

Mekanikteki bir yapı, Gordon tarafından *"yükleri sürdürmeyi amaçlayan herhangi bir malzeme topluluğu"* olarak tanımlanır. Diğer taraftan optimizasyon, en iyi hale

getirmek anlamına gelir. Bu nedenle yapısal optimizasyon, yükleri en iyi şekilde sürdüren bir malzeme montajı yapma konusudur (Christensen ve Klarbring, 2008). Bunun yanı sıra yapısal optimizasyon işlemi belirli sınır koşulları ve kısıtlar altında verilen bir tasarımın topolojisinden başlayarak malzeme dağılımının veya sınır şekillerinin değiştirilmesiyle, optimum tasarıma ulaşılması olarak da tanımlanabilir. İnşaat mühendisliğinde en iyi çözüm, tasarım yüklerini karşılamak ve tasarımı ağırlık açısından verimli hale getirmek adına mümkün olan en iyi yapısal düzenlemeye sahip yapıyı elde etmek olarak görülür (Şekil 4.3). Bu aynı zamanda bina maliyetlerini de dolaylı olarak en aza indirir (Tsiptsis vd., 2019).

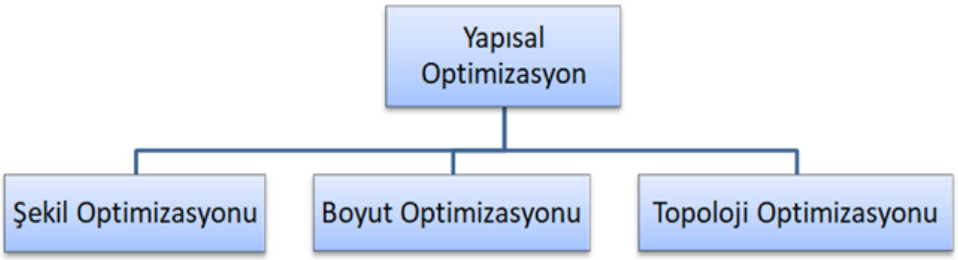


Şekil 4.3. Yapısal optimizasyonun faydaları.

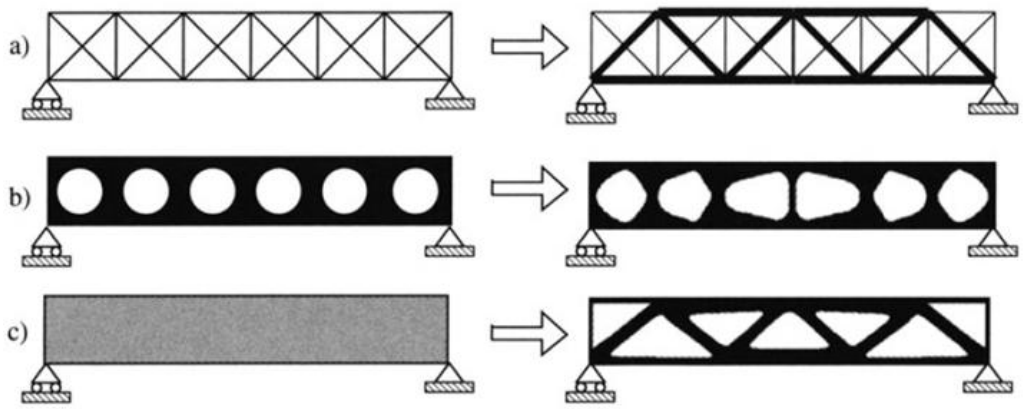
Birçok mühendislik uygulamasında, optimizasyon probleminin değişkenleri keyfi değerlere sahip olamaz. Bunun yerine değişkenlerin bir kısmı veya tamamı pratik nedenlerle bir tamsayı veya ayrık değerler listesinden seçilmelidir. Örneğin, yapısal elemanların standart ebatlarda veya eleman kalınlıklarının ticari olarak temin edilebilir olanlardan seçilmesi gerekebilir (Huang ve Arora, 1995). Bu şekilde optimizasyon sürecinde değişkenlerin bir dizi ayrık değerden seçilmesine ayrık optimizasyon denir. İlk bakışta, ayrık değişkenli bir problemi çözmek sürekli bir problemi çözmekten daha kolay gibi görünebilir. Sonuçta belirli bir aralıktaki değerler sonlu iken, sürekli değerlerin sayısı sonsuzdur. Optimum değer aranırken, sonsuz kümeden ziyade sonlu kümede arama yapmak daha kolay görünmektedir; ancak durum böyle değildir. Ayrık değerli problemleri çözmek sürekli problemlerden çok daha zordur. Bunun nedeni, en küçük problemler dışında hepsinde meydana gelen kombinasyonel büyümedir (Parkinson vd., 2013). Örneğin;

her biri 10 değer alabilen iki değişkenimiz varsa $10^2 = 100$ olasılık, her biri 10 değer alabilen 10 değişkenimiz varsa, 10^{10} olasılık söz konusudur. En hızlı bilgisayarlar için bile bunların tümünü değerlendirmenin uzun zaman alacağı açıktır.

Yapısal optimizasyon genellikle şekil, boyut ve topoloji olmak üzere üç başlık altında incelenir (Şekil 4.4). Boyut, şekil ve topoloji optimizasyon problemleri yapısal tasarım probleminin farklı yönlerini ele alırlar (Şekil 4.5).



Şekil 4.4. Yapısal optimizasyonun alt başlıkları.

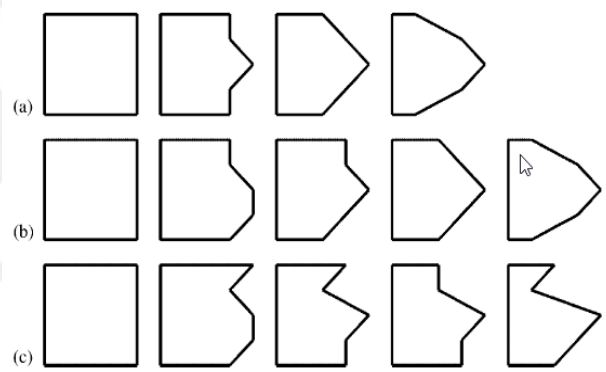


Şekil 4.5. Yapısal optimizasyon çeşitleri: a) bir kafes sistemin boyut optimizasyonu, b) şekil optimizasyonu ve c) topoloji optimizasyonu.

4.1.1. Şekil optimizasyonu

Şekil optimizasyonu bir yapı ve yapı elemanı için optimum şekli arar. Gerilme yığılmalarını, yapısal ağırlığı ve maksimum yer değiştirmeleri azaltmak gibi yapısal özellikleri iyileştirmek ve uygun tasarımlar elde etmek için şekil optimizasyonu gereklidir (Sun vd., 2018).

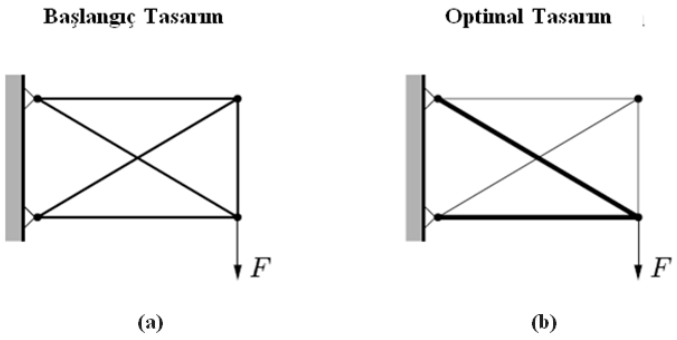
Şekil optimizasyonu yapılırken yapı bağlantısı değişmez; yeni sınırlar oluşmaz (Christensen ve Klarbring, 2008). Sınır düğümlerinin bir dizi aşamalı hareketi ile optimal bir şekle ulaşılabilir. Örneğin; Şekil 4.6'da eşit büyüklükteki bir basamak serisinde varsayımsal bir optimum şekle ulaşılmaktadır. Şekil 4.6 ayrıca optimum şekli sağlayacak birden fazla eşit boyutlu adım dizisi olduğunu gösterir. En az sayıda adım gerektiren tek bir "optimal dizi" vardır. Bunun yanında Şekil 4.6 birden fazla olası sekans olabilmesine rağmen, optimum şekle yol açmayacak sekansların mevcut olduğunu da göstermektedir. Adım hareketi göz önüne alındığında uygun ve uygun olmayan sınır düğümü hareket kombinasyonları vardır; bu nedenle hareket ettirilecek düğümlerin kombinasyonu önemlidir. Belirli bir kombinasyonun derecesi bir bütün olarak yapıya verdiği uygunluk ile belirlenir (Woon vd., 2001).



Şekil 4.6. Sınır düğümü kombinasyonlarının olası üç dizisi: a) mümkün olan en uygun kombinasyonlardan oluşan optimum dizi - elde edilen optimum, b) uygun kombinasyonlardan oluşan optimal olmayan dizi - optimum elde edildi, c) optimal olmayan dizi, uygun olmayan kombinasyonlardan oluşur - optimum elde edilemez.

4.1.2. Boyut optimizasyonu

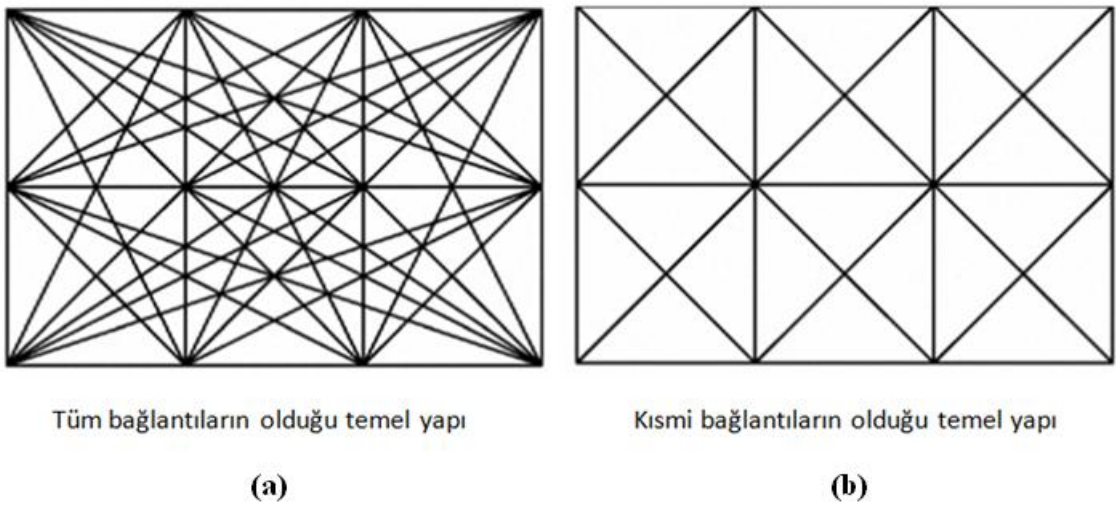
Boyut optimizasyonu yapı elemanları için en uygun kesitlerin araştırıldığı alandır (Şekil 4.7). Örneğin, bir kafes köprü tasarımında, boyutlandırma optimizasyonu, köprüyü oluşturan kiriş elemanların kesit alanlarını veya bir tabakanın kalınlık dağılımı optimize eder. Boyut değişimi sürekli veya ayrık olabilir (Steven vd., 2000). Yapısal optimizasyonun diğer bileşenleri gibi boyut optimizasyonunda da hedeflenen nokta, yapısal ağırlığı veya maliyeti belirli kısıtlar altında en aza indirmektir.



Şekil 4.7. Bir kafes sistem için boyut optimizasyonu: a) tasarım öncesi sistem elemanları, b) tasarım sonrası elde edilen yeni boyutlar.

4.1.3. Topoloji optimizasyonu

Yapısal optimizasyon yer değiştirme veya gerilme kısıtlarına uyarken, yapısal ağırlığı veya deformasyonları en aza indiren optimum bir malzeme kullanımını hedefler. Bu tarz problemlerin ele alındığı en ünlü optimizasyon dalı topoloji optimizasyonudur (Masching ve Bletzinger, 2016). Topoloji optimizasyonu yapısal optimizasyonun en genel şeklidir (Christensen ve Klarbring, 2008). Bir yapının topolojisi yapı elemanlarının varlığı, bağlantıları ve sayısı ile tanımlanabilir. Topoloji optimizasyonu daha fazla malzeme tasarrufu sağlaması, ancak daha karmaşık olması nedeniyle diğer iki optimizasyon alanından (şekil ve boyut) daha faydalıdır; ancak kurgu olarak daha zordur (Su vd., 2009).

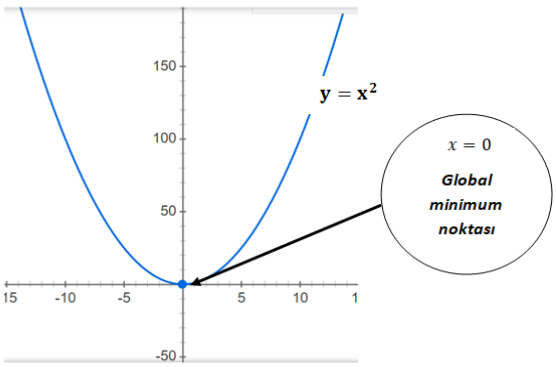


Şekil 4.8. Temel yapısı yaklaşımında mevcut durumlar: a) tüm bağlantıların olduğu temel yapısı, b) kısmi bağlantıların olduğu temel yapısı.

Topoloji optimizasyonu üzerine düşük ağırlıktaki ilk kafes çözümlerini belirleyen, topoloji optimizasyonunun teorik araştırmasında bir kilometre taşı niteliğinde olan ilk çalışma bir asır önce Michell tarafından yayınlanmıştır (Wu vd., 2017). Daha sonra kafes yapıların sürekli tasarım değişkenleri ile sayısal olarak optimizasyonu, ilk olarak 1964'te Dorn vd. tarafından ele alınmıştır (Stolpe, 2016). Modeller, düğümlerin önce tasarım alanı üzerine dağıtıldığı ve daha sonra potansiyel çubuklarla bağlandığı temel yapısı yaklaşımına dayanmaktadır. Temel yapısı yaklaşımında kesit alanını tanımlayan (sürekli) bir tasarım değişkeni, daha sonra her bir potansiyel çubukla birleştirilir (Şekil 4.8).

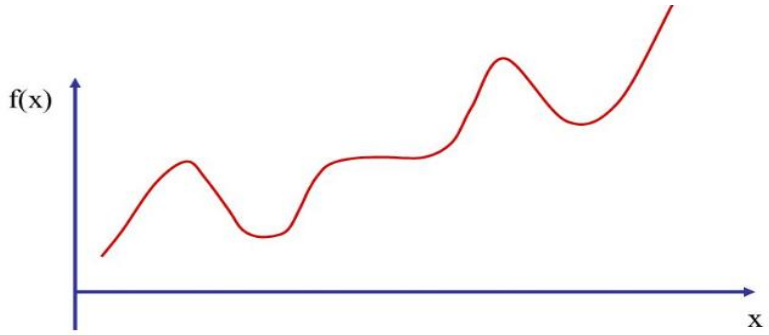
4.2. Yapısal Optimizasyonda Karşılaşılan Zorluklar

Optimizasyon en genel anlamda, bir minimizasyon veya maksimizasyon problemi olarak düşünülebilir. Elimizde $f(x) = x^2$ denklemine karşılık gelen bir problemin olduğunu ve bu problem için minimum çözümün arandığını düşünelim. Genel olarak bir fonksiyon yeterince basitse, çözüm için fonksiyonun ilk türevini $(f'(x) = 0)$ ve çözümün maksimum veya minimum olup olmadığını doğrulamak içinse ikinci türevini $(f(x)'')$ kullanabiliriz. Bu ifadeye göre $f(x) = x^2$ problemi, $-\infty < x < \infty$ arama alanında $x = 0$ noktasında minimum değerini alır $(f(x)_{min} = 0,\ x = 0\ )$ (Şekil 4.9).



Şekil 4.9. f(x) probleminin global değeri.

$f(x) = x^2$ denklemi sürekli, konveks ve tek modlu bir problemdir. Yukarıda ifade edildiği gibi türev bilgisi ile çözümü mümkündür. Bunun yanı sıra elimizde Şekil 4.10'da gösterilen grafiği temsil eden bir problem olduğunu varsayalım. Bu problemin matematiksel ifadesinin ilk denklemde belirtildiği kadar kolay olmadığı açıktır.

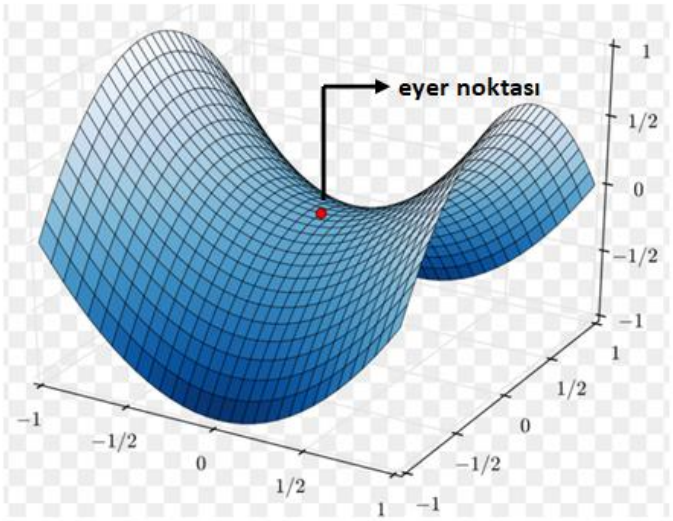


Şekil 4.10. Matematiksel olarak tanımı kolay olmayan bir problem.

Gerçek dünya problemleri Şekil 4.10'daki grafiğe benzer davranış sergilerler. Bu problemler için optimum sonucun elde edilmesi ilk problemdeki gibi hızlı ve kolay olmaz. Bu zorluğun temelinde yatan birbirinden farklı etkenler vardır.

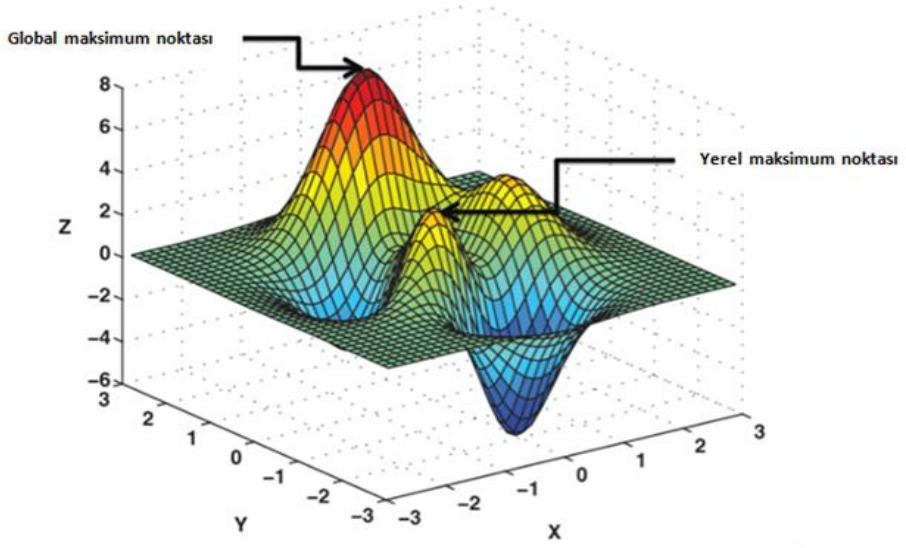
İlk olarak gerçek dünya problemlerinde amaç fonksiyonunu matematiksel olarak elde etmek basit bir süreç değildir. Örneğin, var olan bir problem için maliyet en aza indirilmeye çalışılırsa, maliyetin açık matematiksel ifadesinin olması gerekir ki maliyet fonksiyonunun birinci ve ikinci türeviyle optimum sonuç elde edilsin. Ancak birçok parametreye bağlı bu tür fonksiyonlara ait nicel modeller elde etmek her açıdan pahalı bir süreci kapsayabilir. Bunun yanı sıra bir fonksiyonun türevinin her zaman optimum bir çözümü garanti etmediğine de dikkat edilmelidir, çünkü bazen ne minimum ne de maksimum bir nokta olan, bir bükülme veya başka bir deyişle eyer noktası oluşabilir (Şekil 4.11). İkincisi, Şekil 4.12'de gösterildiği üzere bu problemler çok modlu problemlerdir. Yani birden çok lokal minimum ve maksimum noktasına sahiptir. Bu şekilde global optimuma ulaşmak $f(x) = x^2$ denkleminde olduğu gibi hızlı olmaz. Çünkü optimizasyon sürecinde global değere ulaşırken, değerin lokal noktalarda takılıp kalması muhtemeldir. Üçüncü olarak gerçek dünya problemleri çoğunlukla konveks olmayan problemlerdir. $f(x) = x^2$ denkleminde olduğu gibi gradyan bilgisiyle çözüme gidilemez. Dördüncüsü gerçek mühendislik problemleri çok fazla tasarım değişkeni bulundurur. Örneğin, minimum ağırlığını bulmak istediğimiz bir betonarme yapı için sadece kolonlara ait; kolon sayısı, kolon boyutu, kolon şekli, kolon yönü, kolon açısı vb. birçok değişken söz konusudur. Bunun gibi diğer elemanlar (perde, kiriş, döşeme, duvar vb.) ele alındığı zaman ortaya onlarca tasarım değişkeni çıkar. Tasarım değişkenlerinin sayısının artmasıyla amaç

fonksiyonlarının ifade edilmesi o derece zorlaşır. Beşincisi, tasarım değişkenlerinin çokluğunun yanı sıra gerçek dünya problemlerinde yerine getirilmesi gereken birçok kısıt söz konusudur. Özellikle yönetmelikler başta olmak üzere sağlanması istenen bu kısıtlar problemin çözümünü oldukça zorlaştırmaktadır. Örnek olarak, Bölüm 3'te belirtilen tasarım pratikleri, bir bina tasarımı için sağlanması gereken bazı kısıtlardır. Altıncı olarak mühendislik problemlerinde tasarım değişkenleri genellikle ayrık değişkenlerden oluşur. Bu yönüyle de çözüm süreci sürekli değerlere kıyasla daha zordur.



Şekil 4.11. Eyer noktası.

Günümüz teknolojisindeki son gelişmeler ile yapısal sistemlerin tasarımı daha fazla doğruluk, verimlilik ve hız gerektirir. Bu nedenle Şekil 4.10'da örnek verildiği gibi karmaşık konfigürasyonlara ve çok sayıda parametreye sahip gerçek dünya problemlerinin optimum tasarımı için yeni yöntemlerin geliştirilmesi kaçınılmazdır. Geçmiş dönemlerde optimizasyon problemlerinin çözümüne yönelik Doğrusal Programlama, Doğrusal Olmayan Programlama veya Dinamik Programlama gibi çok çeşitli geleneksel optimizasyon tekniği tasarlanmıştır (Gavrilas, 2010). Geleneksel optimizasyon yöntemleri sürekli ve türevlenebilir fonksiyonların optimizasyonunda kullanılır. Bu yöntemler analitiktir. Bu yüzden en iyi noktaların bulunmasında türev hesaplamalarına ilişkin teknikler kullanılır. Gerçek hayattaki birçok problemin amaç fonksiyonu sürekli ve türevlenebilir olamadığından, kısıtların ve tasarım değişkenlerinin çokluğundan, geleneksel yöntemlerin bu problemlere çözüm bulmada yetersiz kaldıkları açıktır.



Şekil 4.12. Gerçek dünya problemlerine benzer olan minimum ve maksimum noktaları.

Amaç fonksiyonları nicel olarak ifade edilemeyen, çok değişkenli, çok kısıtlı ve çok modlu, konveks olmayan ve çözüm kümesi çoğunlukla ayrık değerlerden oluşan problemlerin üstesinden gelmek için metasezgisel teknikler öne sürülmüştür. Bu teknikler bilgisayar teknolojisinin de ilerlemesiyle hızla gelişerek günümüz dünyasında optimizasyon sürecinin vazgeçilmez bir parçası haline gelmişlerdir.

4.2.1. Metasezgisel algoritmalar

Geçmişten günümüze kadar insanlık her dönemde sorunlarla karşılaşmış ve bu sorunlara çağının sahip olduğu koşullar altında cevaplar üretmeye çalışmıştır. Cevapların her zaman en iyi verimlilikte olması için teknikler ileri sürülmüş ve bu teknikler zaman geçtikçe yerini daha iyi olanlara devretmiştir. Özellikle 1900'lü yıllara kadar eldeki problemler için birçok geleneksel yöntem kullanılmıştır. Birçoğu deterministik olarak adlandırılan bu yöntemlerde genellikle sorunu çözmek için gereken tüm bilgiler mevcut olduğu için herhangi bir değişkenin etkisi kesin olarak hesaplanabilir. Bu yöntemler kullanılırken elde edilecek çıktı tamamen başlangıçta belirlenen girdilere göre belirlenir ve bu uygulamalarda rastgele bir değerin kullanımı gibi bir durum söz konusu değildir. Örneğin, bir problemin çözümü için malzeme özellikleri, uygulanan yükler gibi ifadeler bilinir, rastgele değildir. Deterministik modellerde, modelin çıktısı tamamen parametre değerleri ve başlangıç koşulları tarafından belirlenir. Kullanılan deterministik yöntemlerin

bazıları türev bilgisi kullanır. Bu yöntemlere gradyan tabanlı yöntemler denir. Örneğin, iyi bilinen Newton-Raphson algoritması, fonksiyon değerlerini ve türevlerini kullandığından tek modlu problemler için son derece iyi çalışır (Saka vd., 2016). Deterministik modeller belirli bir parametre seti ve başlangıç koşulları için aynı şekilde çalışıp çözümlerinin eşsiz olmasına rağmen kararsızdırlar. Yani başlangıç koşullarının küçük sapmaları veya problemi yöneten parametreler nihai çözümde büyük değişikliklere yol açar. Böylece elde edilen sonuçlar benzersiz olmasına rağmen tek bir parametreyi veya başlangıç koşulunu hafifçe bozarak önemli ölçüde farklı çözümler elde edilebilir (Renard vd., 2013). Ayrıca bu yöntemlerin başarılı bir şekilde uygulanması için iyi bir başlangıç noktası oldukça önemlidir.

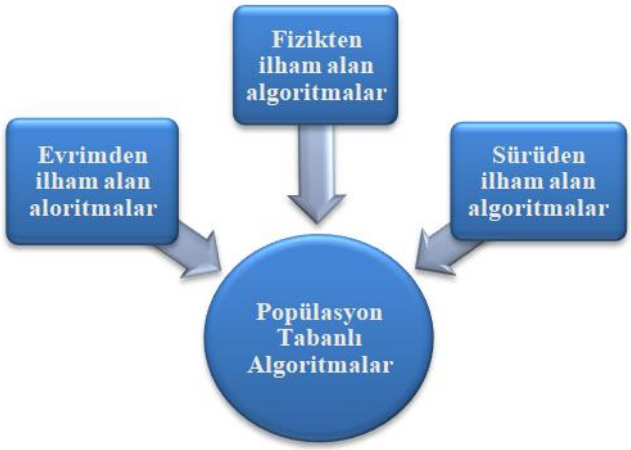
Bölüm 4.2'de de ifade edildiği üzere gerçek dünya problemlerinde çoğunlukla karmaşık, gradyan bilgisi tanımlanamayan, çoğunlukla ayrık değerlerle çalışan, çok modlu amaç fonksiyonları vardır. Bu amaçla geleneksel yöntemlerle çözümlerin elde edilmesi birçok olumsuz durumu beraberinde getirir. Bazen çözüm elde edilmesi mümkün dahi değildir.

Deterministik yöntemlerin yanı sıra karmaşık problemlere çözümler bulmak için kullanılan sezgisel ve metasezgisel algoritmalar 90'lı yılların ortalarından itibaren optimizasyon süreçlerinin vazgeçilmez birer parçası olmuşlardır. Bir hedefi deneme yanılma yoluyla bulmak mantığına dayanan sezgisel yöntemler ilk olarak 1945 yılında Macar matematikçi Polya tarafından tanıtılmıştır (Gavrilas, 2010). Daha sonra bu yöntemler 70'li yıllardan itibaren farklı disiplinlerde sıklıkla kullanılmıştır. Ardından metasezgisel kavramı ilk olarak 1986'da Glover (1986) tarafından yüksek düzeyli bir sezgisel tarama olarak tanıtılmıştır. Bu aşamadan sonra metasezgisel algoritmalar özellikle bilgisayar teknolojisinin de ilerlemesiyle hızla gelişen bir alan olmuştur.

Zhao vd. (2018) tarafından sunulan bir çalışmada metasezgisel algoritmalarla ilgili çok güzel bilgiler içeren bir açıklama Türkçe çevirisi ile aşağıdaki gibidir

> *"Metasezgisel optimizasyon algoritmaları, akıllı hesaplamada giderek daha popüler hale geliyor ve çok sayıda gerçek dünya mühendisliği problemlerine yaygın olarak uygulanıyor. Popülerlikleri aşağıdaki yönlerden kaynaklanmaktadır: İlk olarak, tüm bu optimizasyon tekniklerinin mantıklı olduğu*

kanıtlanmış, gerçek dünyadan gelen ve her tür fiziksel fenomen veya biyolojik davranıştan esinlenen bazı temel teorileri ve matematiksel modelleri vardır. Teoriler basit ve anlaşılması kolaydır. İkinci olarak, bu optimizasyon algoritmaları kara kutu olarak düşünülebilir. Bu bir dizi girdi verildiğinde, bu algoritmaların herhangi bir optimizasyon problemi için kolayca bir dizi çıktı sağlayabileceği anlamına gelir. Daha iyi çözümler elde etmek için algoritmaların yapıları ve parametreleri değiştirilebildiği için çok esnek ve çok yönlüdürler. Üçüncüsü, metasezgisel algoritmalar, birçok mühendislik problemi çok modlu fonksiyonlar olarak kabul edildiğinden, mühendislik problemlerini ele almak için çok değerli olan yerel optimadan etkili bir şekilde kaçınabilir. Ek olarak, çözümlerin doğruluğunu makul bir süre içinde iyileştirmek için diğer algoritmaların yararlarını özümseyerek varyantlarını geliştirebilirsiniz. Dördüncü olarak, metasezgisel optimizasyon algoritmaları, bunlarla sınırlı olmamak üzere, tek amaçlı ve çok amaçlı problemler, düşük boyutlu ve yüksek boyutlu problemler, tek modlu ve çok modlu problemler ve ayrık ve sürekli problemler dâhil olmak üzere farklı problem türlerinin üstesinden gelebilirler."



Şekil 4.13. Popülasyon tabanlı algoritmalar.

Metasezgisel algoritmalar genel olarak tek çözümlü ve popülasyon tabanlı algoritmalar olmak üzere iki kategoriye ayrılmıştır. Tek çözüm tabanlı algoritmalar bir çözümün rastgele üretildiği ve optimum sonuç elde edilene kadar geliştirildiği algoritmalardır; oysa popülasyon tabanlı algoritmalar belirli bir arama alanında bir dizi çözümün rastgele oluşturulduğu ve çözüm değerlerinin en iyi çözüme kadar tekrarlamalar sırasında yükseltildiği algoritmalardır. Bununla birlikte tek çözüm tabanlı algoritmalar, belirli bir sorun için rastgele üretilen tek bir çözümü yeniden biçimlendirdiğinden küresel optimum değerine ulaşmamızı engelleyebilecek yerel optimum noktasına sıkışabilirler (Dhiman ve Kaur, 2018). Bunun yanı sıra popülasyon temelli algoritmalar yerel optimumdan kaçma konusunda çok yüksek bir performans seviyesine sahiptirler. Bu

nedenle günümüzde popülasyon tabanlı algoritmalar birçok zorlu problemin çözünde aktif olarak kullanılır.

Popülasyon tabanlı metasezgisel algoritmalar; evrimden, fizikten ve sürüden ilham alan yöntemler olarak kabaca üç sınıfa ayrılır (Şekil 4.13). Evrimsel algoritmalar, türlerin evriminden ilham alan algoritmalardır. Genel olarak Charles Darwin'in evrim teorisine dayanarak doğal seçilim, yani en uygun olanın hayatta kalmasına ve dolayısıyla nüfusun ortalama uygunluğunun artmasına yol açan çevresel baskıya maruz kalan bir birey popülasyonunun varlığı ile karakterize edilir. Uygunluk bir canlının çevresine adaptasyon derecesinin ölçüsüdür. Evrimi etkileyen ana yöntemler üreme, mutasyon, rekombinasyon, doğal seçilimdir. Holland tarafından 1975'te sunulan GA, iyi bilinen klasik bir evrim algoritmasıdır. GA genellikle mutasyon, çapraz geçiş ve seçim adımlarını kullanarak yüksek kaliteli çözümler elde edebildiğinden, orijinal versiyon ve varyantları birçok gerçek dünya problemine yaygın olarak uygulanır (Zhao vd., 2019). Alanlarında popüler bazı genetik algoritmalar İngilizce isimleri ile Evolution Strategies (ES) (Beyer ve Schwefel, 2002), Genetic Programming (GP) (Koza, 1994), Differential Evolution (DE) (Storn ve Price, 1997), Biogeography-Based Optimization (BBO) (Simon, 2008) şeklinde sıralanabilir.

İkinci kategori fizik tabanlı algoritmalardır. Bu algoritmalarda arama ajanları yerçekimi kuvveti, elektromanyetik kuvvet vb. gibi fizik kurallarına göre arama alanı etrafında hareket edebilir. Fizikten ilham alan algoritmalar evrendeki fiziksel yasaları taklit eder. SA (Kirkpatrick vd., 1983) en iyi bilinen fizik esinli algoritmalardandır. SA, ısıtılmış bir metalin minimum enerji ile kristal bir dokuya soğuduğu ve donduğu fiziksel malzemede kullanılan tavlama işleminden esinlenmiştir. Alanlarında popüler bazı fizik esinli algoritmalar BB-BC (Erol ve Eksin, 2006), Gravitational Search Algorithm (GSA) (Rashedi vd., 2009), Chemical Reaction Optimization (CRO) (Nayak vd., 2019), Galaxy-based Search Algorithm (GbSA) (Shah-Hosseini, 2011), Ray Optimization (RO) (A. Kaveh ve Khayatazad, 2012), Magnetic Optimization Algorithms (MOA) (Tayarani-N. ve Akbarzadeh-T., 2014), Light Ray Optimization (LRO) (Shen ve Li, 2009), Gravitation Field Algorithm (GFA) (Zheng vd., 2010), Water Evaporation Optimization (WEO) (Ali Kaveh ve Bakhshpoori, 2019) şeklinde sıralanabilir.

Doğadaki sürü davranışına yönelik çalışmalar sonrası çok verimli optimizasyon algoritmalarının ortaya çıkışı mümkün olmuştur. Sürü esinli metasezgisel algoritmalar olarak karşımıza çıkan bu dalın doğal örnekleri arasında; karınca kolonileri, hayvan sürüsü, kuş akınları, bakteri üremesi ve balık eğitimi yer alır (Roy vd., 2014). Sürü esinli algoritmalar sınıfında yer alan ve kuş sürüsü davranışlarını taklit eden en popüler algoritmalardan birisi PSO algoritmasıdır (Kennedy ve Eberhart, 1995). PSO, belirli bir kalite ölçüsüne göre aday bir çözümü yinelemeli geliştirmeye çalışarak problemi optimize eden bir hesaplama yöntemidir. Alanlarında popüler bazı sürü esinli algoritmalar İngilizce isimleri ile Fish Swarm Optimization (FSO) (Neshat vd., 2012), Tree-seed Algorithm (TSA) (Kiran, 2015), Dolphin Echolocation (DE) (A. Kaveh ve Farhoudi, 2013), Hunting Search (HS) (Oftadeh vd., 2010), Bird Mating Optimizer (BMO) (Askarzadeh, 2014), Squirrel Search Algorithm (SSA) (Jain vd., 2019), Shark Smell Optimization (SSO) (Mohammad-Azari vd., 2018), Ant Colony Optimization (ACO) (Dorigo vd., 2006), Salp Swarm Algorithm (SSA) (Mirjalili vd., 2017), Sine Cosine Algorithm (CSO) (Mirjalili, 2016) olarak örneklendirebilir.

Literatürde farklı prensiplerden esinlenen metasezgisel algoritmaların olduğu da belirtilmelidir. Bu alanda popüler olmuş bazı algoritmalar İngilizce isimleri ile TLBO (Rao vd., 2011), HS (Askarzadeh ve Rashedi, 2017), Group Search Optimizer (GSO) (Tan ve Zhu, 2010), Interior Search Algorithm (ISA) (Gandomi, 2014), Social-Based Algorithm (Ramezani ve Lotfi, 2013), Exchange Market Algorithm (EMA) (Ghorbani ve Babaei, 2014) olarak örneklendirebilir.

Herhangi bir optimizasyon algoritması için keşif ve sömürü kavramları, çok temel iki bileşendir (Emary ve Zawbaa, 2016). Keşif aşamasının ana rolü küresel alanı olabildiğince rastgele keşfetmek iken, sömürü aşaması araştırma aşamasının sonuçlarını takip eder ve uzayda gelecek vaat eden alanları ayrıntılı olarak inceler (Wu vd., 2019). Keşif ve sömürü arasında uygun bir denge bulmak optimizasyon sürecinin sezgisel doğasından ötürü herhangi bir metasezgisel algoritmanın geliştirilmesinde en zorlayıcı görevdir (Mirjalili ve Lewis, 2016). Bir algoritmanın genel verimliliği esas olarak bu iki bileşen arasındaki ince dengeden etkilenir. Keşif çok az ve sömürü çok fazla ise sistem yerel optimumda sıkışabilir. Bu durumda küresel optimum noktasını bulmak çok zor hatta imkansız olur. Öte yandan çok fazla keşif ancak çok az sömürü varsa sistemin birleşmesi

zor olabilir. Bu durumda genel arama performansı yavaşlar. Bu iki bileşenin dengelenmesi başlı başına zorlu bir problemdir. Arama sırasında en iyi çözümleri seçmek için uygun bir mekanizma veya kriter göz önünde bulundurulmalıdır. Optimizasyon süreçlerinde en iyi olanın hayatta kalması mantığı, sürecin temel prensibini oluşturur. Dahası her defasında bulunacak çözüm bir önceki çözüm ile kıyaslanarak aralarında gereken kıyaslamalar yapılır ve sonuçta daha iyi olan çözüm güncel çözüm olarak saklanır. Bu işlem tüm optimizasyon sürecinde tekrarlanarak optimum çözüm aranır.

5. ÖNERİLEN TAŞIYICI SİSTEM PLANINI HAZIRLAMA YÖNTEMİ

Bölüm 4'de değinildiği gibi optimizasyon problemleri değişkenler, kısıtlar ve amaç fonksiyonu olmak üzere üç temel bileşenden oluşur. Optimizasyon sürecinde problemlerin amaç fonksiyonu maksimize veya minimize edilerek optimum sonuca gidilmeye çalışıldığı için elimizde amaç fonksiyonuna ait nicel bir model olmak zorundadır. Optimizasyon sürecinin temel zorluğu, optimize edilecek problemde amaç fonksiyonunun matematiksel bir modelinin yokluğunda nasıl optimize edileceğidir. Matematiksel modellerin amaç fonksiyonunun tasarım değişkenlerine nasıl bağlı olduğunun açıkça sunulması halinde çalışan optimizasyon yazılımlarına karşı inatçı olduğu veya tamamen mevcut olmadığı durumlarda, bu problemlerin "kara kutu problemleri" olduğunu belirtmekte fayda vardır. Kara kutu problemleri amaç fonksiyonu hakkında bilgi sahibi olmadan sadece girdi ve çıktıları açısından görüntülenebilen bir sistem olarak düşünülebilir (Şekil 5.1).



Şekil 5.1. Yalnızca girdi ve çıktıları açısından görüntülenebilen kara kutu problemleri.

Bu noktaya kadar özellikle vurgulanmak istenen nokta gerçek mühendislik problemlerinin matematiksel ifadesi mümkün olmayan ve kara kutu problemleri olarak karşımıza çıkan amaç fonksiyonlarından oluşmasıdır. Birçok değişken ve kısıtla içinden çıkılması neredeyse imkânsız hale gelen amaç fonksiyonlarının üstesinden gelmek için 90'lı yılların ortalarından itibaren sezgisel yöntemler geliştirilmiş ve günümüzde bilgisayar teknolojisinin gelişmesiyle daha genel bir arama yeteneğine sahip metasezgisel algoritmaların kullanımına geçilmiştir.

Yapısal optimizasyon algoritmaları optimizasyon sorununun rastgele seçilmiş bir grup aday çözümü ile başlar ve maliyet fonksiyonu temelinde grubun kalitesini artırmaya çalışır. Çoğunlukla doğadan ilham alan bir dizi basit kurala göre maliyet fonksiyonu değerlerini geliştirmek için mevcut çözümler yinelemeli olarak uygulanır. Çözüm arama alanını verimli bir şekilde arayarak ve boyutunu azaltarak metasezgisel algoritmalar tarafından bulunur (Gharehchopogh ve Gholizadeh, 2019). Buradan da anlaşılır ki sezgisel yöntemler genellikle yinelemeli algoritmalar olarak kabul edilir; burada her yineleme, daha önce bulunan en iyi çözümden daha iyi olabilecek yeni bir çözüm için arama yapılmasını içerir. Algoritmanın sonlandırıldığı makul bir süreden sonra sağladığı çözüm, herhangi bir yineleme sırasında bulunan çözümün en iyisidir (Kaveh, 2016).

Bu noktada söylemek gerekir ki her yineleme bir aday tasarımın çözülmesi demektir. Her yinelemeyle işlemleri tekrarlama sürecinde çok fazla işlem gücü harcanır. Bu noktada kullanılan algoritmanın başarısına bağlı olarak daha üst seviyede verimlilik elde edilebilir. Başarılı bir algoritma ile gerek zaman konusunda tasarruf, gerek problemin sunulması açısından kolaylıklar ve gerekse elde edilecek sonuçların daha kaliteli olması bakımından birçok avantaj elde edilebilir. Bu şekilde bir algoritmanın başarısı yadsınamayacak bir durum olarak karşımıza çıkar ve herhangi bir metasezgisel algoritmanın yapısal optimizasyondaki başarısı üç ana unsurla test edilebilir.

I. **Kesinlik:** Bir algoritmanın başarısı ürettiği çözümlerle belirlenebilir. Buradaki kesinlik kavramıyla vurgulanmak istenen, her bir analiz sonrası ortaya çıkan sonuçların birbirinden çok uzak olmamasıdır. Üretilen çözümler arasındaki standart sapma ne kadar düşükse algoritma o kadar sağlam ve güvenilirdir diyebiliriz. Düşük standart sapma değeri algoritmanın farklı çalışmalarda neredeyse benzer bir cevap vermesi anlamına gelebilir.
II. **Çalışma zamanı:** Sezgisel yöntemler genellikle yinelemeli algoritmalar olarak kabul edilir. Yineleme veya iterasyon, bir bilgisayar yazılımı aracılığı ile problemdeki işlemlerin tekrarıdır. Sezgisel yöntemlerde yaygın olarak kullanılan iterasyon yönteminde birden fazla veri bloğu sırayla işlenir. Bu tekrarlar genellikle yazılan kodlar içerisindeki "while" veya "for" döngüsü ile gerçekleştirilir. Bu döngüler belirli bir sayıya veya çözümde istenilen duruma ulaşılana kadar tekrarlanır. Çalışma zaman yapılan iterasyonların sayısına bağlı olarak değişir.

Başarılı bir algoritmada yapılacak toplam iterasyon sayısını belirleyen temel bir kriter yoktur. Ancak günümüz rekabetçi ortam göz önüne alındığında zaman konusu çok önemli bir faktör olarak karşımıza çıktığından, makul düzeylerde kalabilmek önemlidir.

III. **Hassasiyet:** Başarılı bir algoritmanın diğer bir özelliği ise diğer algoritmalarla kıyaslandığında makul çözümler üretmesidir. Günümüzde farklı esinlerle birçok algoritma geliştirilmekte ve birçoğu gerçek dünya problemlerine uygulanmaktadır. Bu algoritmalar birbirleriyle kıyaslandığı vakit elde edilen çözümler konusunda aralarında bir paralellik görülür. Bu kriter özellikle problemin çözümüyle doğrudan bağlantılı olduğu için en belirleyici özelliklerden sayılabilir.

Bu şekilde bir algoritmanın başarısını belirleyen en temel özelliklerden birkaçına değinilmiştir. Burada karşımıza çıkan ve bu tezin de kapsamı içerisine giren esas konu çalışma zamanıdır. Çalışma zamanı daha önce de belirtildiği üzere rekabetçi piyasada sadece uygun çözümlerin üretilmesi değil, aynı zamanda bu çözümlerin makul sürelerde üretilmesi bağlamında belirleyici bir unsur olarak karşımıza çıkar. Özellikle birçok değişken ve kısıttan oluşan gerçek dünya problemleri ele alındığı vakit, zaman konusu çok daha önemli bir noktaya gelmektedir. Çünkü bilgisayar yazılımlarıyla geliştirilen kodlar büyük veri kümeleriyle çalıştırıldığı zaman, çalışma süresi sorun haline gelir. Bu kapsamda günümüzde makul sürelerde çözümler üretmek adına çok çeşitli çalışmalar yapılmaktadır.

Çalışma zamanını kullanılan bilgisayar, programlama dili, algoritma, girdi biçimi vb. gibi unsurlar belirler. Burada altı çizilmek istenen unsur algoritma girdi biçimidir. Bir algoritmadaki girdi biçimini tasarım değişkenlerinin sayısı ve biçimi oluşturur. Aynı problem için birbirinden farklı tasarım vektör biçimleri kullanılarak, her birinden farklı sürelerde çözümler alınabilir. Buradan anlaşılacağı üzere bir problemin çözüm süresi tasarım vektör biçimlerine göre farklılıklar gösterir.

Tasarım vektörü problemin tüm tasarım değişkenlerini içinde bulunduran bir vektördür. Bu vektörün ne kadar tasarım değişkeni içerdiği, tasarım değişkenlerinin düzeni problemin çözümünü belirleyen en önemli unsurlardan biridir. Bir betonarme projesi ele aldığımızı düşünürsek birçok tasarım değişkeni ortaya çıkar. Örneğin; kiriş elemanları için, ilk bakışta kirişin bir noktadaki varlığı, kiriş boyutu, kiriş kaçıklığı gibi değişkenlerle karşı

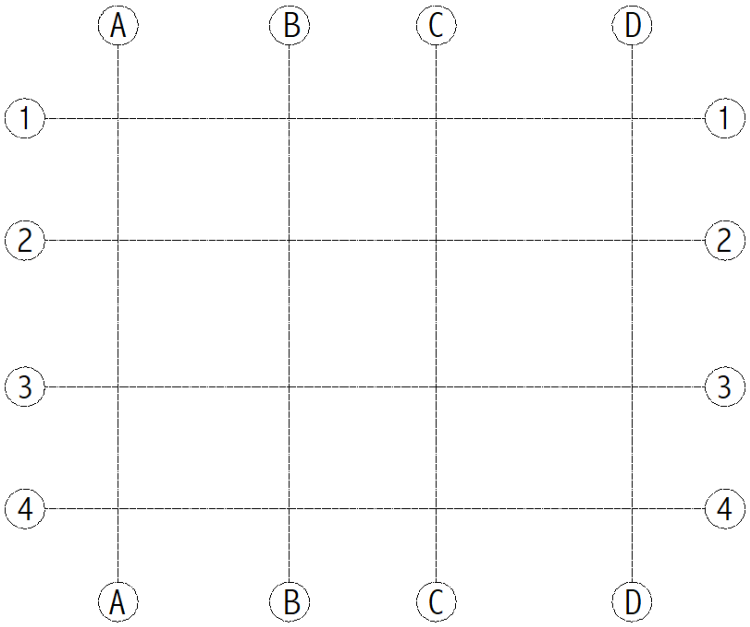
karşıya kalırız. Bu değişkenlerin sayısının artması ya da azalması söz konusu olabilir ve bu da farklı tasarım vektörlerinin ortaya çıkmasını sağlar. Benzer şekilde tasarım vektöründeki değişkenlerin biçimleri yani hangi sırayla dizildikleri, hangi ifadelerle gösterildikleri gibi unsurlar da tasarım vektörlerinde farklılıkların oluşmasının bir diğer nedenidir.

Bu noktada bir kez daha önemle vurgulamak gerekir ki, bir yapının optimal tasarım problemi tahmin edilemeyecek kadar zor bir konudur. Mimari açıdan basit bir yapı dahi ele alındığında yapıyı tarif edecek onlarca tasarım değişkeni ve optimal tasarıma yön vermek amacıyla ele alınması gereken onlarca kısıt ortaya çıkabilir. Bunun yanı sıra bir yapıya yapısal optimizasyonun tüm alanlarını (topoloji, boyut ve şekil) uygulamak da çok zor bir işlemdir. Bir yapı için sadece elemanların topolojilerine karar vermek dahi başlı başına bir çalışmanın konusunu oluşturabilecek niteliktedir.

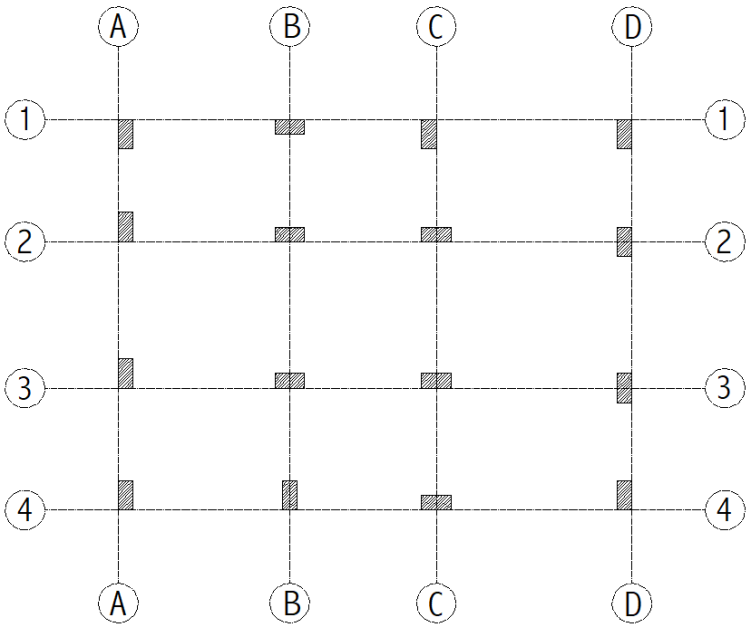
Diğer yapısal optimizasyon dalları gibi sürekli veya ayrık olarak yapılabilen boyut optimizasyonu, farklı yapısal elemanların kesitlerinin daha iyi yapısal performans sağlayacak şekilde ayarlandığı en erken yapısal optimizasyon şeklidir. Birçok mühendislik alanında olduğu gibi inşaat mühendisliği problemlerinin çoğunda tasarım değişkenleri ayrıktır. İnşaat mühendisliğinde tasarlanacak bir betonarme yapı için optimum kolon boyutlarının arandığı bir problemi ele aldığımızda, boyutların uygulamada sıklıkla tercih edilen kesitler arasından seçilmesinde fayda vardır. Örneğin, optimizasyon sonrası $33{,}8x52{,}4\ cm^2$ veya $30{,}9x60{,}4\ cm^2$ gibi kolon kesitleriyle karşılaşmak istenen bir durum değildir. Piyasa şartları ele alındığında uygulamada sıklıkla tercih edilen, şantiye ortamında uygulama açısından kolaylıklar sağlayan bazı kesitlerle karşı karşıya kalırız (örneğin; $30x50, 30x60, 40x40, 40x60\ cm^2$ vs.). Bu şekilde ayrık değerlerle çalışan boyut optimizasyonuna "Ayrık boyut optimizasyonu" denir.

Topoloji optimizasyonu ise bir yapı için elemanların (betonarme yapılar için kolon, kiriş, perde ve döşeme) varlıkları, bağlantıları ve sayısı ile tanımlanabilir. Şekil 5.2'de bir aks sistemi gösterilmiştir. Bu aks sistemi üzerine düzenlenecek bir taşıyıcı sistem planı için kolonların tasarımını ele alalım. İnşaat mühendisliğinde "iki aksın kesişim noktasına kolon yerleştirilir" ilkesine göre kolonlar yerleştirilmeye başlanır. Şekil 5.3'te de gösterildiği üzere kolonların gelebileceği muhtemel yerlere kolonlar yerleştirilmiş ve başlangıç için bir

ön tasarım elde edilmiştir. Bu noktada dikkat çekilmek istenen nokta, kolonlar için başlangıçta konumların belli olması ve başlangıç tasarımının belirli olan bu noktalar ile oluşturularak optimizasyon sürecine dâhil edilmesidir. İşte bu şekilde bir alan içerisinde belirli noktalara göre başlayan topoloji optimizasyonuna "Ayrık topoloji optimizasyonu" denir. Bu aşamadan sonra gerekli görülen kolonlar topoloji optimizasyonu ile kaldırılır.

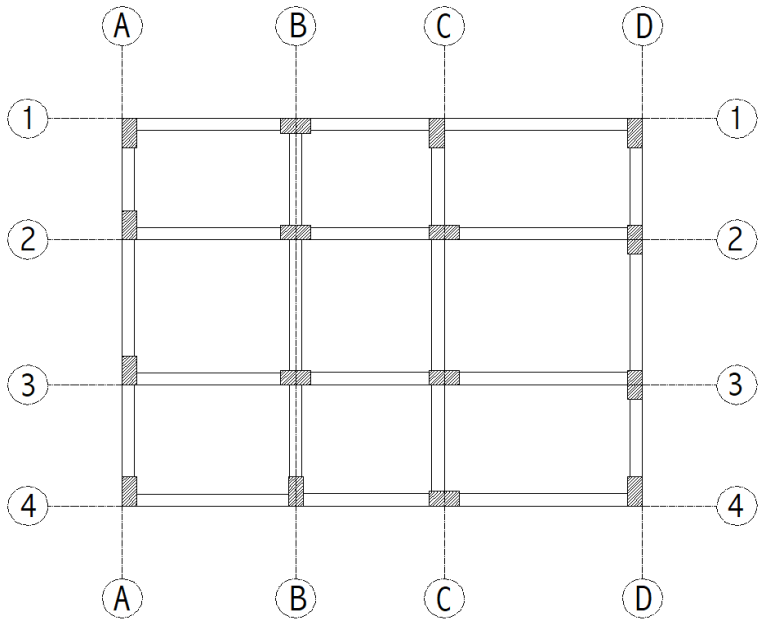


Şekil 5.2. Örnek bir aks sistemi.



Şekil 5.3. Aks sistemi üzerinde kolon yerleşimleri.

Benzer şekilde "kirişler kolonları birbirine bağlar" bilgisiyle kirişleri yerleştirelim. Ancak Şekil 5.4'de görüleceği gibi kirişler sadece ortogonal şekilde birleşmiş, her kolon yalnızca kendisine komşu olan kolonlarla bağlanmıştır. Dolayısıyla kirişler içinde belli bir yerleşim durumu olduğundan ayrık optimizasyon kuralları geçerlidir.

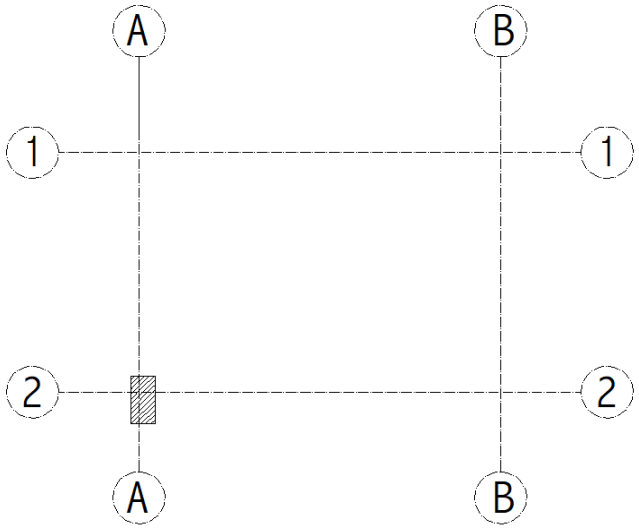


Şekil 5.4. Ayrık optimizasyon sonucu oluşan kiriş topolojisi.

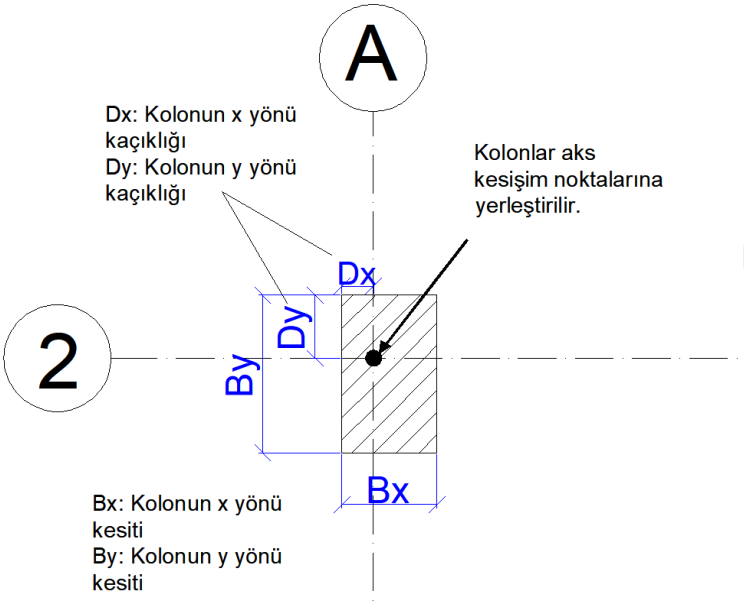
İnşaat mühendisliği uygulamalarında sürekli optimizasyon sonucunda estetik, güvenlik vb. durumlar açısından optimal yapıların oluşumu mümkün değildir. Bu yüzden genellikle ayrık optimizasyon teknikleri ile uygulanabilir sonuçlara ulaşılmaya çalışılır. Bu tez kapsamında da ayrık optimizasyon teknikleri uygulanacaktır.

Betonarme yapıların optimal tasarımına yönelik sunulacak stratejinin daha iyi anlaşılabilmesi için optimizasyon sürecinin en önemli bileşenleri arasında olan tasarım vektörü konusu üzerinde durmanın önemi büyüktür. Tasarım vektörü bir problemdeki tüm tasarım değişkenlerini barındıran girdi biçimidir. Bir sistemde var olan kolonlara ve kirişlere ait tasarım vektörünü meydana getirecek değişkenlere sırasıyla göz atalım.

Şekil 5.5'te basit bir aks sistemi üzerinde 2 ve A akslarının kesişim noktasına yerleştirilen bir kolon görülmektedir. Buna göre yerleştirilen kolonu biraz yakından inceleyelim.



Şekil 5.5. Basit bir planda 2 ve A akslarının kesişim noktasına yerleşen bir kolon.



Şekil 5.6. Bir kolona ait tasarım değişkenleri.

2 ve A akslarının kesişim noktasına yerleştirilen bu kolon için elimizde olması gereken tasarım değişkenlerin aşağıdaki gibi sıralanabilir (Şekil 5.6).

I. **Kolonun varlığı veya yokluğu:** Bir yapıyı sisteme tanıtmak için öncelikle sınırları belli bir alanda mümkün noktalara gelebilecek elemanların varlığının belirlenmesi gerekir. Bir kolonun gelebileceği muhtemel yerler proje üzerinde bellidir. Ancak herhangi bir noktaya kolon gelip gelmeyeceği konusu optimizasyon algoritması tarafından karar verilmesi gereken bir durumdur. Bundan dolayı kolonun bir

noktadaki varlığı bir tasarım değişkeni olarak karşımıza çıkar.

II. **Kolon kesit türü:** Yapı tasarım sürecinde farklı tür kesitlerle çalışmak söz konusu olabilir. Kolonlar için kullanılan dikdörtgen, daire veya T, L ve U tarzında poligon kesitler sıklıkla kullanılan eleman kesit türleridir. Bir noktaya hangi kolon tipinin geleceğinin doğru kararı da yine optimizasyon algoritması tarafından belirlenmesi gereken bir durum olduğu için, kesit türü tasarım vektöründe yer alması gereken bir tasarım değişkenidir. Ancak bu tez kapsamında tüm kesit türleri ile çalışılmamış uygulamada sıklıkla tercih edilen dikdörtgen ve kare kesit türleri ele alınmıştır.

III. **Kolon boyutu:** Optimizasyon algoritması tarafından belirlenmesi gereken diğer bir konu ise kolon boyutudur. Seçilen kesit türüne bağlı olarak dikdörtgen kesit için kısa ve uzun kenar bilgisi, daire kesit için çapın veya yarıçapın belirlenmesi gerekir. Dolayısıyla kolon boyutları da bir tasarım değişkeni olarak ele alınmalıdır.

IV. **Kolon kaçıklığı:** Bir kolon bir noktaya yerleştikten sonra proje sınırları içinde kalmak şartıyla hareket etme hakkına sahiptir. Bu hareketin optimizasyon algoritması tarafından belirlenmesi, optimum sonuçların ortaya çıkmasını sağlar. Bu yüzden kolon kaçıklık bilgisi de bir tasarım değişkenidir.

V. **Kolonun aksa göre pozisyonu:** Bir kolon bir noktaya farklı şekillerde yerleşebilir. Başka bir deyişle yerleştiği noktadaki iki akstan herhangi birine paralel veya dik konumlanabilir. Bu bilgi de bir tasarım değişkeni olarak karşımıza çıkar.

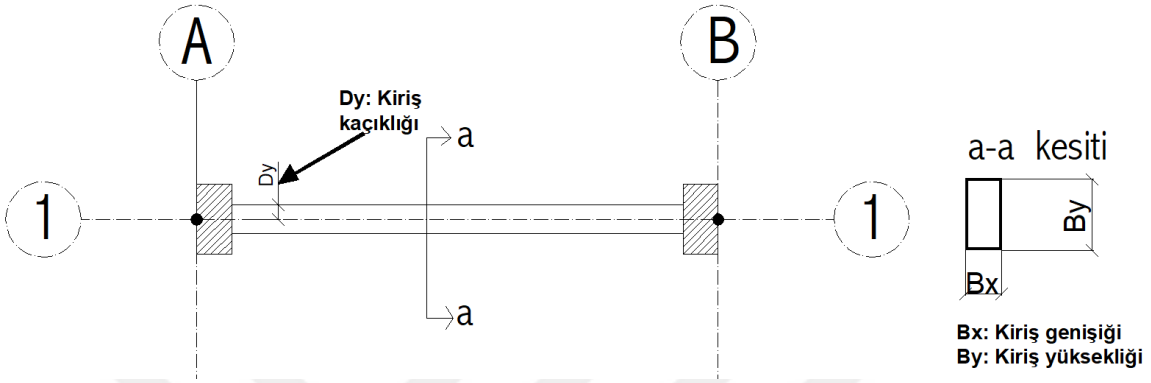
Bu noktada vurgulamak gerekir ki bir tasarım sürecinde sistemdeki düşey taşıyıcı elemanları tanıtmak için yukarıdaki bilgilere (kolonun varlığı-yokluğu, bir noktaya nasıl yerleşeceği, hangi tür kesitte ve boyutta olacağı bilgileri) ihtiyaç vardır.

Kolon için tasarım değişkenleri belirlendikten sonra şimdi bir kiriş için tasarım değişkenlerine göz atalım. Şekil 5.7'de gösterilen bir kirişin bir noktaya yerleşmesi için gereken bilgiler aşağıdaki gibidir:

I. **Kirişin varlığı veya yokluğu**
II. **Kiriş boyutu**
III. **Kiriş kaçıklığı**

Kirişler kolonlara göre daha az değişkenle ifade edilirler. Bunun nedenlerinden ilki, kirişler kolonlar gibi farklı türlerde olmazlar. Uygulamada genellikle dikdörtgen kesit

olarak tercih edilirler. Dahası kirişler iki kolonu birbirine bağladıkları için bir sistemde hem hareket etme şansları sınırlıdır hem de kolonları birbirine bağlayan doğrultu üzerinde başka bir açı yapma seçenekleri yoktur. Bu yüzden yalnızca yerleştikleri aksa dik olarak hareket ederler.



Şekil 5.7. Bir kirişe ait tasarım değişkenleri (Bx, By, Dy).

Son olarak döşemeleri ele alırsak, döşemeler kolonlar veya kirişler gibi birbirinden farklı değişkenler içermezler. Döşemelerin plandaki yerleri ve boyutları (en ve boy) mimari projede belirtildiği üzere sabittir. Bir döşeme için sadece kalınlık değişkeni vardır. Dolayısıyla döşemenin kalınlığına bağlı olarak yalnızca boyut optimizasyonu yapılır. Ancak bu tez kapsamında döşemeler göz ardı edilerek yalnızca kolon ve kiriş elemanlar üzerinde durulmuştur.

Bir betonarme projesinin tasarım sürecinde kolonlar ve kirişler için gerekli değişkenler yukarıda anlatıldığı gibidir. Bu noktada tekrar değinmek gerekir ki bu bilgiler tasarımı gerçekleştirilecek eleman için gereklidir. Yani bu bilgilerden herhangi birinin eksikliği sonucunda yapılması istenen tasarım ya mümkün olmaz ya da mümkün olsa dahi eksik bir şekilde gerçekleşir. Ancak bu değişkenlerin ne biçimde dizileceğinin tek bir yolu yoktur.

5.1. Optimizasyon Algoritması

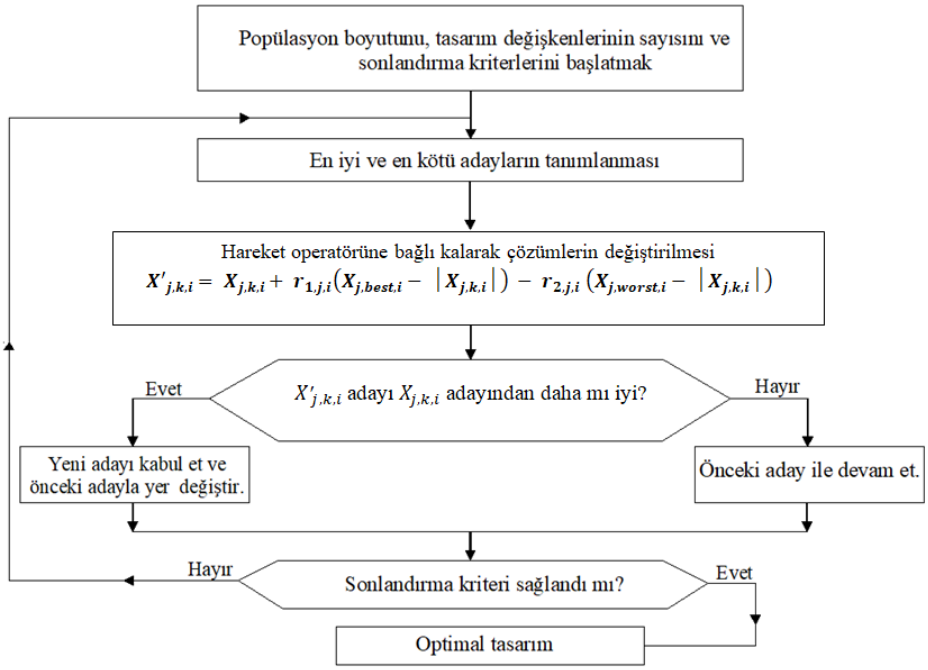
Bu tez kapsamında Rao (2016) tarafından önerilen, uygulanması rahat, gerçek dünya problemlerine uyarlanabilir, üst seviyede sezgisel arama yeteneğine sahip "Jaya" algoritması kullanılmıştır. Jaya algoritması kısıtlı, kısıtsız optimizasyon problemleri için

tatmin edici performansı ile güçlü bir potansiyele sahip, uygulaması oldukça kolay, parametresiz ve nispeten daha az sayıda fonksiyon değerlendirmesi gibi birçok avantaja sahiptir. Jaya algoritması, belirli bir problem için elde edilen çözümün en iyi çözüme doğru hareket etmesi ve en kötü çözümden kaçınması konseptine dayanır. Bu algoritma sahip olduğu birçok avantaj ile alandaki araştırmacılar tarafından yoğun bir ilgiyle karşılanmıştır. Jaya algoritmasının popülerliği aşağıdaki özelliklerden kaynaklanmaktadır:

İlk olarak, Jaya algoritması matematiksel olarak ifadesi zor, kara kutu problemleri olarak değerlendirilen gerçek dünya problemlerine makul sürelerde, gerçekçi çözümler bulan metasezgisel algoritmalar sınıfında yer alır. İkincisi, metasezgisel algoritmalar geleneksel metotlara kıyasla zorlu problemlere makul sürelerde uygulanabilir çözümler üretmelerine rağmen, alanında başarılı birçok algoritmanın kendine özgü parametreleri bulunur. Örneğin; GA (Holland, 1984) seçim operatörünün ayarlanmasını, mutasyon ve çaprazlama olasılığı gibi parametrelere ihtiyaç duyarken, PSO (Kennedy ve Eberhart, 1995) atalet ağırlığının ve sosyal ve bilişsel parametrelerin ayarlanmasını gerektirir. Herhangi bir algoritmanın kontrol parametreleri yanlış ayarlanırsa algoritma yerel bir optimum noktasında sıkışabilir veya algoritmanın yakınsama hızı yavaşlayabilir. Algoritmaya özgü parametrelerin ayarlanması popülasyon büyüklüğü gibi ortak kontrol parametrelerine uygulanır ve optimizasyon problemini çözmeden önce kullanıcı bunları ayarlamalıdır. Bundan dolayı algoritmaya özgü parametrelerin çeşitli kombinasyonları ile bir dizi çalışma yapılmalıdır. Kullanıcının çabasının, ayarlanabilir parametrelerin sayısındaki artışla birlikte artacağı açıktır (Ravipudi ve Neebha, 2018). Bu şekilde bir algoritmanın parametreli olması çok tercih edilen bir durum değildir. Jaya algoritması parametresiz olması bakımından, kullanıcıya büyük kolaylıklar sağlayan ve parametrelerin yanlış ayarlanmasından kaynaklanan yanlış hesaplamalardan bu şekilde korunan bir algoritmadır.

Üçüncüsü, Jaya algoritması sahip olduğu sade hareket operatörü ile (Denklem 5.1) uygulanması oldukça kolaydır. Dördüncüsü, güçlü bir arama motoruna sahip bu algoritmanın etkinliği matematiksel optimizasyon problemleri, mekanik tasarım problemleri, çok amaçlı optimizasyon problemleri ve kafes yapıların sürekli optimizasyonu gibi birbirinden farklı alanlarda gösterilmiştir.

$$\boldsymbol{X'_{j,k,i}} = \boldsymbol{X_{j,k,i}} + \boldsymbol{r_{1,j,i}}\left(\boldsymbol{X_{j,best,i}} - \left|\boldsymbol{X_{j,k,i}}\right|\right) - \boldsymbol{r_{2,j,i}}\left(\boldsymbol{X_{j,worst,i}} - \left|\boldsymbol{X_{j,k,i}}\right|\right) \qquad (5.1)$$



Şekil 5.8. Jaya algoritmasının akış şeması.

Denklem 5.1.'de Jaya algoritmasının hareket operatörü gösterilmiştir. Burada $X_{j,best,i}$, en iyi ve $X_{j,worst,i}$, en kötü aday için j değişkeninin değeridir. $X'_{j,k,i}$, $X_{j,k,i}$ değerinin güncellenmiş halidir ve $r_{1,j,i}$ ve $r_{2,j,i}$ iterasyon boyunca değerleri $[0, 1]$ arasında güncellenen rastgele değerlerdir. $r_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|)$ ifadesi çözümün en iyi çözüme yaklaşma eğilimini belirtirken, $r_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|)$ ifadesi ise çözümün en kötü çözümden kaçınma eğilimini belirtir. İterasyon süresince eğer $X'_{j,k,i}$ değeri daha iyi bir değere ulaşırsa kabul edilir. Şekil 5.8'de algoritmaya ait akış şeması gösterilmiştir.

5.2. Sunulan Strateji

Belirtmek gerekir ki çalışma konusu olan bir problemi çözebilmek için ideal prosedür eşzamanlı şekil, boyut ve topoloji optimizasyonudur. Ancak yapılar için topoloji, boyut ve şekil optimizasyonunu birlikte uygulamak zor bir işlemdir. Hesaplama gücü açısından çok güçlü bilgisayarların varlığını gerektirir. Ancak bu çalışma yapılacak işlemlerin daha sonradan tüm kullanıcılara ulaşmasını hedeflediği için, normal bir bilgisayarın dahi altından kalkabileceği performansı hedefler.

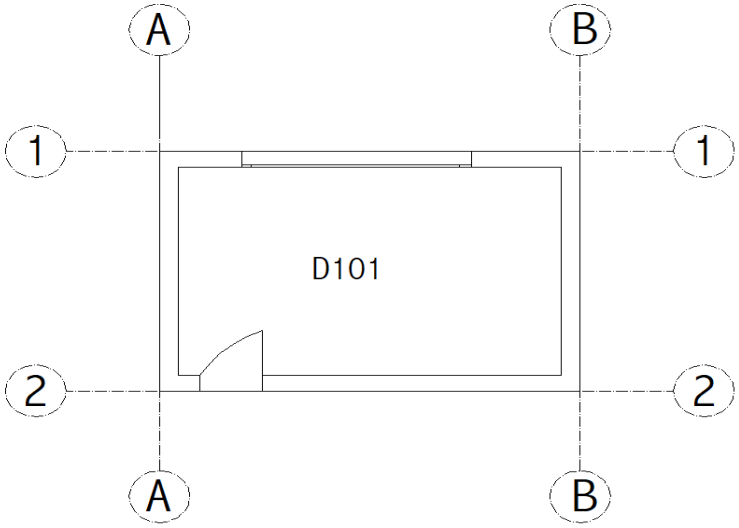
Bu projede sunulan strateji üç aşamalı bir prosedürden oluşmaktadır. Tasarım için verilen bir plan için ilk olarak kiriş topolojisi oluşturulmuş sonrasında ise kiriş topolojisine bağlı olarak kolon topolojisi belirlenmiştir. Yapıya ait taşıyıcı sistem elemanlarının topolojileri belirlendikten sonra ise elemanların boyut ve yerleşimlerinin uygunluğu ile süreç devam etmiştir. Buna göre tasarımı yapılacak planın uygunluk derecesinin ölçülmesini sağlamak üzere belirlenen üç aşamalı prosedür aşağıdaki gibi sıralanabilir:

1. Kiriş topolojisi için amaç değeri, kiriş ağının toplam uzunluğu olarak belirlenmiştir. Kısıtları sağlayan kirişler ne kadar kısa bir kiriş ağı oluşturuyorsa, sistem o kadar iyi anlamına gelmektedir. Tasarım adayının amaç değeri, Bölüm 3'te değinilen tasarım pratikleri kurallarına göre değerlendirilerek aday cezalandırılır ve uygunsuz bir tasarım olması halinde algoritma tarafından algılanması sağlanır.
2. Kolon topolojisi için amaç değeri, sistemdeki var olan kolonların toplam sayısıdır. Kısıtları sağlayan kolonların sayısı ne kadar az ise, sistem o kadar iyi anlamına gelmektedir. Tasarım adayının amaç değeri, Bölüm 3'te değinilen tasarım pratikleri kurallarına göre değerlendirilerek aday cezalandırılır ve uygunsuz bir tasarımın oluşması halinde algoritma tarafından algılanması sağlanır.
3. Elemanların boyutu ve yerleşimi için amaç değeri, ceza değeri ile aynıdır. Buna göre en küçük amaç değerine (cezaya) sahip tasarım en iyi tasarım olarak değerlendirilebilir. Elemanların boyutları yapısal analiz yapılmadan da söylenebilecek kurallara göre belirlenmiştir (örneğin; bir yapıda her iki ortogonal yönde en az iki perde olması uygundur).

Bu noktada "Sunulan strateji ile kolon ve kiriş boyutlarına nasıl karar veriliyor?" gibi bir soruyla karşılaşmak ihtimal dâhilinde olabilir. Önemle vurgulamak gerekir ki bu projede sunulan strateji kolon ve kirişleri yapısal analize gönderecek "ön boyutları" belirler. Elemanlar için nihai boyutlara defalarca yapısal analiz yapıp kesitler üzerinde değişikler yapılarak ulaşılır. Bu bakış açısıyla bu tezde sunulan prosedürün (yapısal analiz yapılmadığından) bir boyutlandırma prosedürü olduğu düşünülmemelidir. Sunulan strateji bilgisayar ile otomatik tasarım konusunda bir ön çalışma olarak ele alınmalıdır.

Çizelge 5.1. Tasarım pratiklerinin stratejideki görevleri.

Tasarım Pratikleri	Tasarım Pratiğinin Stratejideki Rolü
Kirişlerin kullanım alanı içinde olması	Kiriş kaçıklığı
Kirişlerin serbest ucunun olmaması	Kiriş topolojisi, Kolon topolojisi
Mümkün olduğunca çerçevelerin tamamlanması	Kiriş topolojisi
Tam duvarların altında kiriş bulunması	Kiriş topolojisi
Mümkün olduğunca yarım duvarların altında da kiriş olması	Kiriş topolojisi
Kirişlerin mümkünse kıymetli alanlar içerisinden geçmemesi	Kiriş topolojisi, Kiriş kaçıklığı
Komşu açıklıkların birbirinden çok farklı olmaması	Kiriş topolojisi, Kolon topolojisi
Mümkün olduğunca saplama kiriş oluşumlarının engellenmesi	Kolon topolojisi
Çakışık eksenli kirişlerden oluşan sürekli kiriş hatları mümkün olmadığı takdirde çakışık eksenli olmayan kirişler ile sürekli kiriş hatlarının oluşturulmaya çalışılması	Kiriş topolojisi
"Saplama kirişe saplanan kiriş" oluşumuna izin verilmemesi	Kiriş topolojisi, Kolon topolojisi
Saplama kiriş oluşumunun engellenemediği durumlarda, saplama noktasının, kirişin mesnetlendiği kolonlara çok yakın olmaması	Kiriş topolojisi, Kolon topolojisi
Kiriş açıklığının çok büyük olmaması	Kiriş topolojisi, Kolon topolojisi
Birbirine çok yakın olan kolonlardan uygun görülenlerin kaldırılması	Kolon topolojisi
Kolonların oturma alanı dışına çıkmaması	Kolon topolojisi, Kolon kaçıklığı
Kolonların kapı ve pencere boşluklarına taşmaması	Kolon topolojisi, Kolon kaçıklığı
Kolonların mümkünse kıymetli alanlar içerisinden geçmemesi	Kolon topolojisi, Kolon kaçıklığı
Perde konfigürasyonu iki farklı yönde en az ikişer adet perde olacak şekilde düzenlenmelidir.	Kolon topolojisi
Perdeler mümkün olduğunca taşıyıcı sistem planında dış akslara yerleştirilmelidir.	Kolon topolojisi
Planda perdelerin uzun kenarı doğrultusundaki çizgiler bir noktada kesişmemelidir.	Kolon topolojisi
Kolonların her iki deprem yönünde yaklaşık aynı rijitliği göstermesi sağlanmalıdır.	Ön boyutlandırma
Kiriş-kolon aksı elden geldiğince çakıştırılmalıdır.	Kiriş kaçıklığı, kolon kaçıklığı

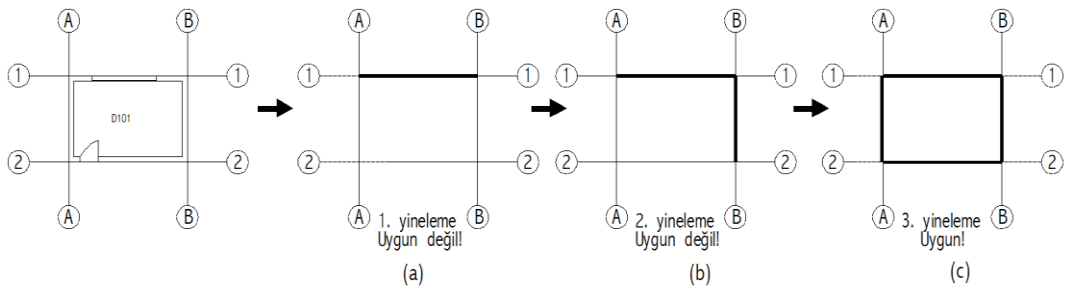


Şekil 5.9. Örnek olarak ele alınan mimari proje.

Sunulan stratejide bir ön taşıyıcı sistem planının nasıl oluştuğunu daha iyi anlatmak adına işlem adımları basit bir mimari proje (Şekil 5.9) ele alınarak anlatılmıştır. Buna göre işlem adımları aşağıdaki gibi sıralanabilir:

I. Öncelikle mimari proje sisteme tanıtılır. Bu noktada mimari proje üzerinde gerekli değişiklikler (bu değişiklikler Bölüm 6'da ayrıntılı olarak anlatılacaktır) yapılarak proje analize hazır hale getirilir.

II. Üç aşamalı tasarım prosedürünün ilk aşaması olan kiriş topolojisi belirlenir. Çizelge 5.1'de kiriş topolojisi ile ilgili tasarım pratikleri bu noktada devreye girer ve optimizasyon algoritması yineleme metoduyla defalarca atamalar yaparak plana ait en uygun (cezası en düşük) kiriş topolojisine karar verir.



Şekil 5.10. Kiriş topolojisi sırasında optimizasyon algoritması tarafından optimal topolojinin belirlenmesi aşamaları: a) birinci yineleme sonrası kiriş topoloji planı, b) ikinci yineleme sonrası kiriş topoloji planı, c) üçüncü. yineleme sonrası kiriş topoloji planı.

Kirişler için ayrık topoloji optimizasyonu yapılırken kirişlerin varlığı veya yokluğu araştırılır. Verilen bir plan için kiriş gelebileceği yerler tanıtılır. Bölüm 3'te ve Çizelge 5.1'de verilen tasarım pratikleri kısıt olarak belirlenir ve uygun kiriş topolojisi optimizasyon algoritması tarafından aranır. Bunu vektör biçiminde gösterecek olursak kiriş varlığı 1, yokluğu ise 0 ile ifade edilmiştir. Buna göre sırasıyla 1., 2. ve 3. yinelemelere karşılık gelen kiriş topolojileri için vektör biçimleri şöyledir:

1.*Yineleme*: [1, 0, 0, 0], (Şekil 5.10 (a))

2.*Yineleme*: [1, 0, 0, 1], (Şekil 5.10 (b))

3.*Yineleme*: [1, 1, 1, 1], (Şekil 5.10 (c))

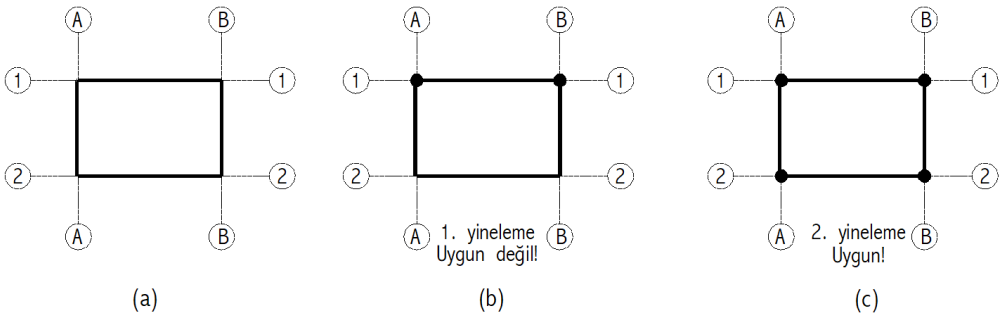
Böylece optimizasyon algoritmasının yinelemeler sonrası bulacağı uygun tasarımda (3. yineleme) tüm duvarların altına kiriş gelmiş ve vektör biçiminde de tüm adayların ifadeleri 1 olarak belirlenmiştir.

III. Sunulan prosedüre ait ikinci aşamada, sistemin kolon topolojisi belirlenir. Bu aşamada kiriş topolojisi belirlenen sistemin (Şekil 5.10 (c)), kolonlara ait kısıt fonksiyonları (Çizelge 5.1'de belirtilen) devreye girerek optimizasyon algoritması ile en uygun kolon topoloji konfigürasyonu seçilir (Şekil 5.11 (c)). Buna göre Şekil 5.11'de verilen durumlar için kolon topolojisi vektör biçimi aşağıdaki gibidir:

1.*Yineleme*: [1, 1, 0, 0], (Şekil 5.11 (b))

2.*Yineleme*: [1, 1, 1, 1], (Şekil 5.11 (c))

Optimizasyon algoritması ilk yinelemede yalnızca iki aks kesişimine kolon koymuştur. Ancak 2. denemede yaptığı atama ile daha uygun bir tasarıma ulaşmıştır.



Şekil 5.11. Kiriş topolojisi belirlenen sistem için kolon topolojisinin belirlenmesi: a) birinci yineleme sonrası kolon topoloji planı, b) ikinci yineleme sonrası kolon topoloji planı.

IV. Eldeki sistem için hangi noktalara kolonların ve kirişlerin geleceği belirlendikten sonra tasarım prosedürünün üçüncü ve son aşamasına geçilir. Bu kısımda eleman topoloji planı belirlenen proje için boyut ve yerleşim durumları belirlenir. Belirtmek gerekir ki boyut ve kaçıklık için kesin değerler yapısal analiz sonucunda belirlenir. Ancak bu çalışma yapısal analiz modülü olan bir ürünün ön çalışması olarak düşünüldüğünden bu kısımda yapılacak boyutlandırma veya kaçıklık değerlerinin belirlenmesi gibi işlemler yapısal analiz aşamasından önceki ön aşamayı teşkil etmektedir. Dolayısı ile değerler nihai değerler değildir. Şekil 5.10 (c) ve Şekil 5.11 (c)'de görüleceği üzere "Uygun" olarak oluşan tasarımlar için herhangi bir kiriş veya kolon boyutu veya bu elemanların hangi kaçıklık ile konumlanacağı belli değildir. İşte bu noktadan sonra boyut ve şekil optimizasyonları devreye girerek elemanların sistem içindeki nihai durumları belirlenir. Elemanlara ait yerleşim ve boyut ifadeleri, topolojileri belli olan kolon ve kirişlere ait boyut, kaçıklık ve pozisyon (kolonlar için) atamalarının yapıldığı aşamadır. Bu işlemler eşzamanlı olarak hem kirişler hem de kolonlar üzerinde yapılır. İşlem basamakları şöyledir:

a. Boyut optimizasyonu için kiriş ve kolonlara ait ayrı kesit listeleri tanımlanır. Burada örnek verecek olursak;

   $$\boldsymbol{Kiriş\ Kesit\ Listesi} = [[25,50],[25,60],[30,60]]$$

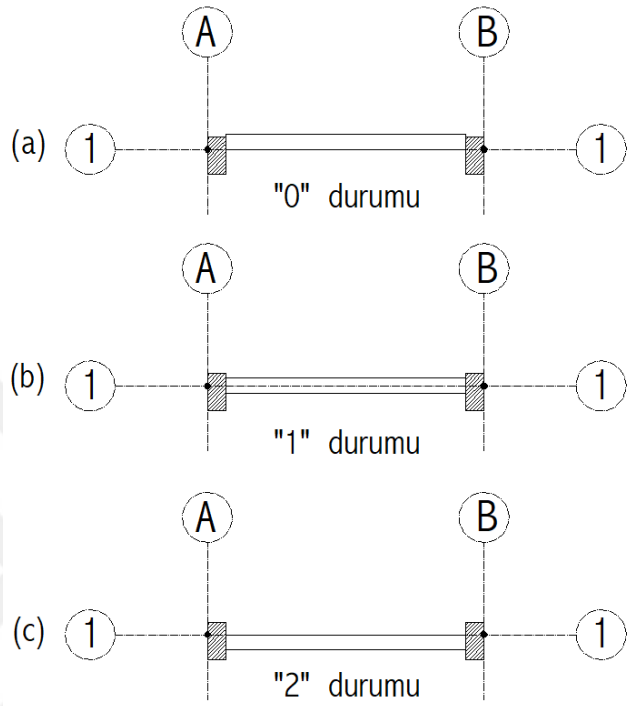
   $$\boldsymbol{Kolon\ Kesit\ Listesi} = [[30,50],[30,60],[30,80],[40,40],[25,150]]$$

   şeklinde ifade edebiliriz. Burada kesit sayısı isteğe bağlı olarak arttırılıp, azaltılabilir. Bu elemanlar vektör şemasında gösterilirken liste içerisinde bulundukları indeks numaraları ile ifade edilmiştir. Örneğin; herhangi bir kolon için $[25,150]$ kesit ataması yapıldığında bu kesit listede 4. eleman olduğu için (saymaya sıfırdan başladık) vektör şemasında 4 olarak ifade edilecektir.

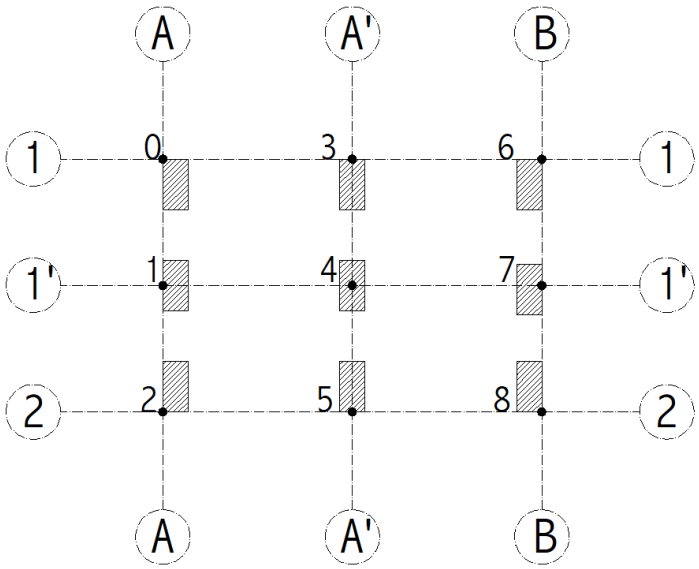
b. Şekil optimizasyonu kolonlara ve kirişlere ait kaçıklıkların atamasıyla yapılır. Kiriş ve kolon kaçıklıkları için listeler aşağıdaki gibidir:

   $\boldsymbol{Kiriş\ Kaçıklık\ Listesi} = [0,1,2]$, (Şekil 5.12)

Kiriş kaçıklık listesinde belirtilen 0 değeri, Şekil 5.12 (a)'da gösterilen durumu gösterirken, 1 değeri Şekil 5.12 (b)'deki, 2 değeri ise Şekil 5.12 (c)'deki durumu ifade eder.



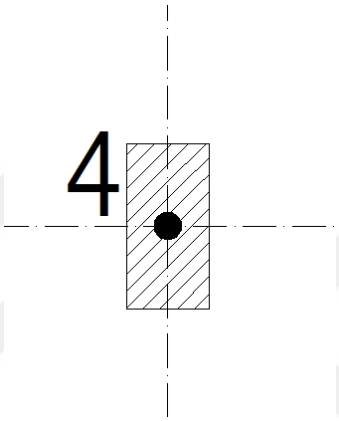
Şekil 5.12. Kirişlerde kaçıklık durumları ve gösterim ifadeleri: a) kirişin aksa üstten yerleşmesi durumu, b) kirişin aksı ortalaması durumu, c) kirişin aksa alttan bağlanması durumu.



Şekil 5.13. Kolon kaçıklık durumları ve gösterim ifadeleri.

$\boldsymbol{Kolon\ Kaçıklık\ Listesi} = \boldsymbol{[0, 1, 2, 3, 4, 5, 6, 7, 8]}$, (Şekil 5.13).

Kolon kaçıklık listesinde verilen rakamsal ifadeler Şekil 5.13'te verilen durumlara karşılık gelir. Vektör listesindeki her rakam şekil üzerindeki değerini karşılıktır. Örneğin, vektör şemasında herhangi bir kolon için kaçıklık ifadesi değeri 4 olarak görüldüğünde aklımıza Şekil 5.14'de görülen kolon durumu gelecektir.



Şekil 5.14. Kolon kaçıklık ifadesi 4 olan bir kolonun pozisyonu.

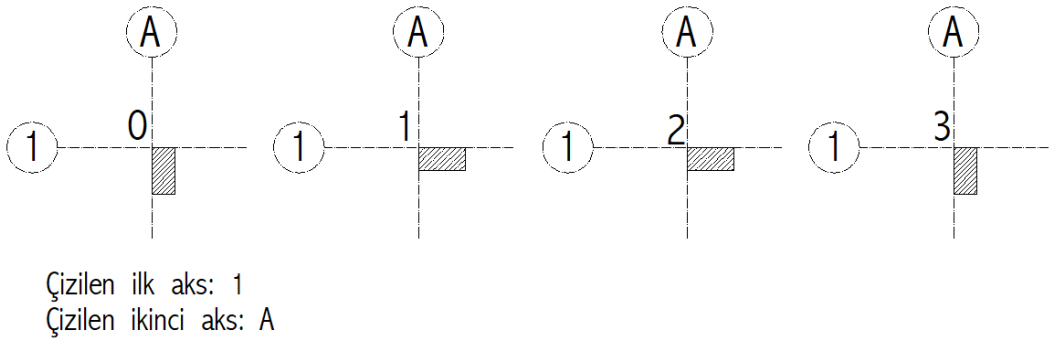
c. Son madde olarak ifade edilecek bu durum, yalnızca kolonlar için geçerli olan ve aksa göre alınan pozisyonların vektör şemasındaki gösterimidir. Bu madde kirişleri etkilemez. Çünkü kirişler iki kolonu bir doğrultu üzerinde birbirine bağlarlar ve alacakları konum zaten üzerinde bulundukları aksa göredir. Ancak bir kolon iki ayrı aks kesişiminde yer aldığı için, iki aksa göre farklı şekillerde durabilir. Bu noktada çizilen iki akstan hangisinin ilk ve hangisinin ikinci olarak çizildiği önemlidir. Aksların çizildiği sıraya göre kolon duruşları Şekil 5.15'da gösterilirken, bu bilginin vektör şemasındaki ifade şekli de "Kolon Yön Listesi"nde ifade edilmiştir.

$\boldsymbol{Kolon\ Yön\ Listesi} = \boldsymbol{[0, 1, 2, 3]}$, (Şekil 5.15).

Herhangi bir [x,y] kesitine sahip kolon için "Kolon Yön Listesi"nde belirtilen rakamlar aşağıdaki gibi ifade edilir:

- 0: Çizilen ilk aksa paralel olan kenar "x" kenarıdır (Şekil 5.15 (a)).
- 1: Çizilen ikinci aksa paralel olan kenar "x" kenarıdır (Şekil 5.15 (b)).
- 2: Çizilen ilk aksa paralel olan kenar "y" kenarıdır (Şekil 5.15 (c)).

- 3: Çizilen ikinci aksa paralel olan kenar "y" kenarıdır (Şekil 5.15 (d)).

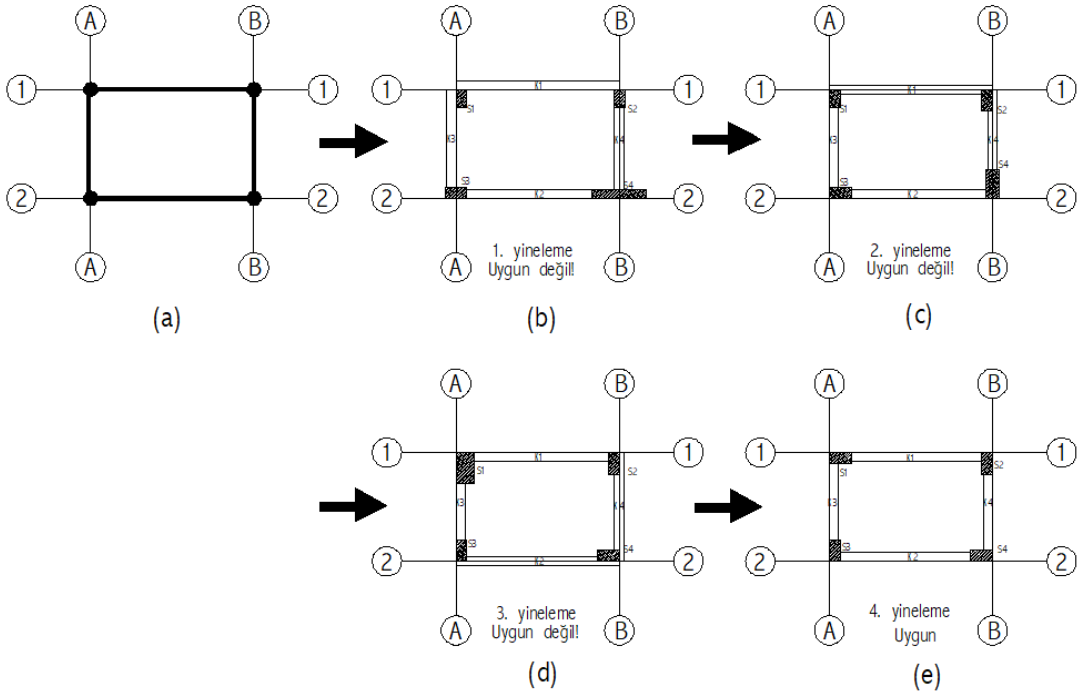


Şekil 5.15. Bir kolonun iki aksa göre aldığı pozisyonlar ve gösterim ifadeleri.

Şekil ve boyut optimizasyonları süresince kiriş ve kolon elemanların alabileceği durumlar ve bu durumlara bağlı vektör şemasında gösterilecek ifadeler ana hatlarıyla yukarıda açıklanmıştır. Bu noktada şunu belirtmek gerekir ki şekil ve boyut optimizasyonları kiriş ve kolonlar için eşzamanlı gerçekleşeceği için oluşacak tasarım vektörü aynı anda kiriş ve kolon bilgilerini içerecektir. Oluşacak vektör tasarımı;

$$[\textit{Kiriş kaçıklığı}, \textit{Kiriş boyutu}, \textit{Kolon kaçıklığı}, \textit{Kolon boyutu}, \textit{Kolon yönü}]$$

şeklinde olacaktır. Buna göre vektör şemasında ilk olarak sistemdeki tüm kirişlere ait sırasıyla kaçıklıklar ve boyutlar yazılıp, ardından tüm kolonlara ait sırasıyla kaçıklıklar, boyutlar ve duruşlar yazılarak vektör şemasının son hali elde edilir. Bundan sonra daha önceden verilen (bkz. Şekil 5.9) ve ilk olarak kiriş sonrasında da kolon topolojisi belirlenen projenin (bkz. Şekil 5.16 (a)), Çizelge 5.1'de verilen kısıtlar ile şekil ve boyut optimizasyon süreçlerinde meydana gelebilecek olası durumlar ve bu durumlara karşılık gelen vektör şemalarına göz atalım.



Şekil 5.16. Şekil ve boyut optimizasyonları sonrası elde edilen tasarımlar.

Çizelge 5.2. Boyut ve şekil optimizasyonları sırasında kullanılan vektör şemasında var olan değerlerin karşılık geldiği ifadeler.

$\left[\underbrace{\textit{kiriş kaçıklık değerleri}}_{\widehat{0,2,0,1}} \quad \underbrace{\textit{kiriş boyut değerleri}}_{\widehat{0,1,1,0}} \;,\; \underbrace{\textit{kolon kaçıklık değerleri}}_{\widehat{0,3,5,5}} \;,\; \underbrace{\textit{kolon boyut değerleri}}_{\widehat{0,0,0,4}} \;,\; \underbrace{\textit{kolon yön değerleri}}_{\widehat{0,0,2,2}}\right]$

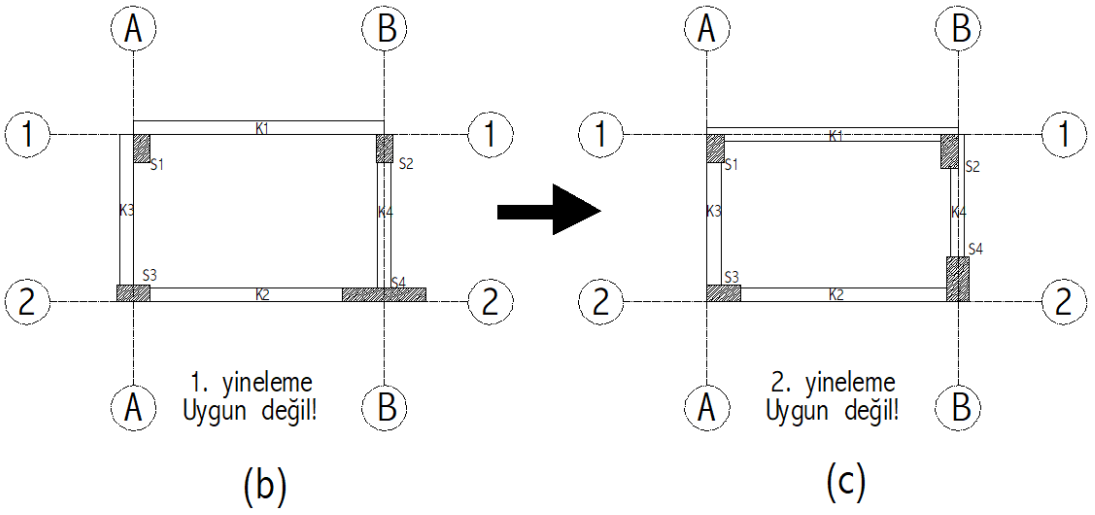
Çizelge 5.2'de kiriş ve kolonlar için boyut ve şekil optimizasyonları sırasında vektör şemalarında oluşan değerlere ait ifadeler gösterilmektedir. Aşağıda Şekil 5.16'da verilen tasarımlar, Şekil 5.17 ve 5.18'de büyütülerek gösterilmiştir ve yinelemelere ait vektör şemaları aşağıdaki gibidir.

1.*Yineleme*: **[0, 2, 0, 1, 0, 1, 1, 0, 0, 3, 5, 5, 0, 0, 0, 4, 0, 0, 2, 2]**, (Şekil 5.17 (b)).

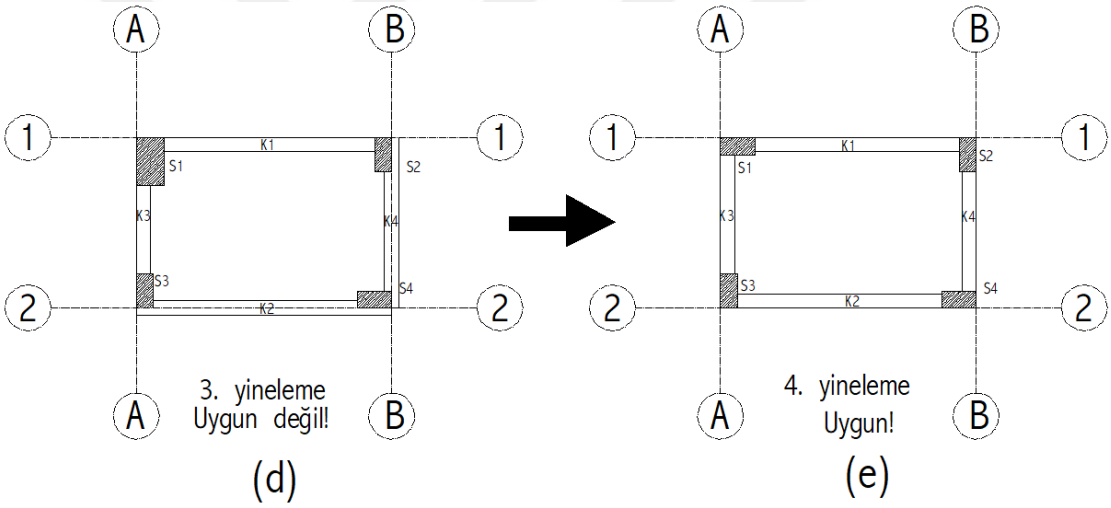
2.*Yineleme*: **[1, 2, 2, 1, 0, 0, 0, 0, 6, 2, 5, 0, 0, 0, 0, 1, 0, 2, 3, 2]**, (Şekil 5.17 (c)).

3.*Yineleme*: **[2, 2, 1, 1, 1, 1, 0, 1, 0, 6, 2, 8, 2, 0, 0, 0, 3, 3, 0, 1]**, (Şekil 5.18 (d)).

4.*Yineleme*: **[2, 2, 2, 2, 0, 0, 1, 1, 0, 6, 2, 8, 0, 0, 0, 0, 2, 0, 0, 1]**, (Şekil 5.18 (e)).

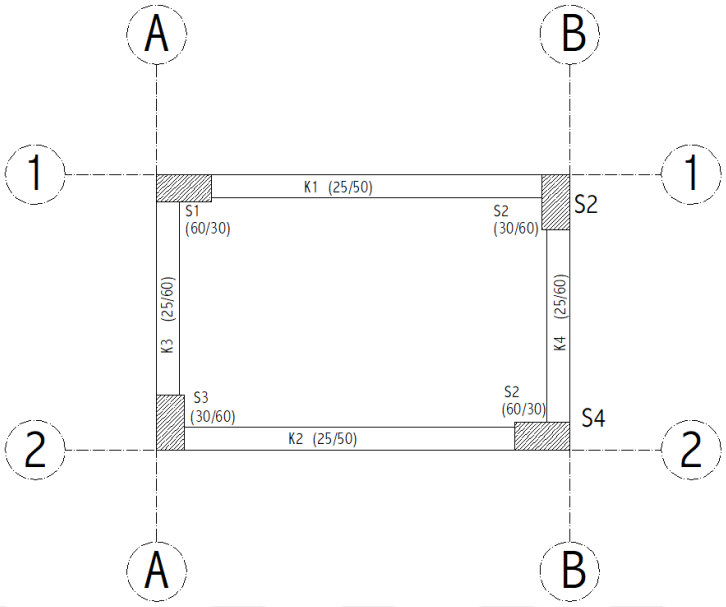


Şekil 5.17. Birinci ve ikinci yinelemelere ait tasarımlar: b) birinci yineleme, c) ikinci yineleme.

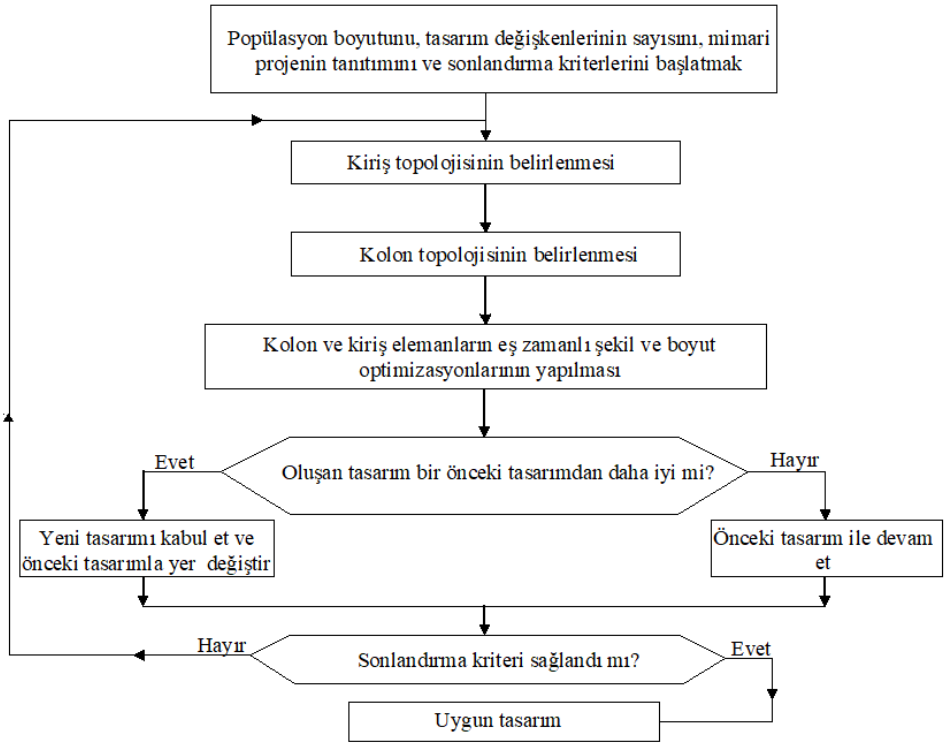


Şekil 5.18. Üçüncü ve dördüncü yinelemelere ait tasarımlar: d) üçüncü yineleme, e) dördüncü yineleme.

Betonarme yapıların optimal tasarımına yönelik sunulan strateji ile örnek olarak verilen mimari tasarıma (bkz. Şekil 5.9) ait sistemin topoloji, şekil ve boyut optimizasyonları sonrasında elde edilen uygun tasarım Şekil 5.19'da gösterilmiştir.



Şekil 5.19. Dördüncü yineleme sonrası elde edilen uygulanabilir tasarım.



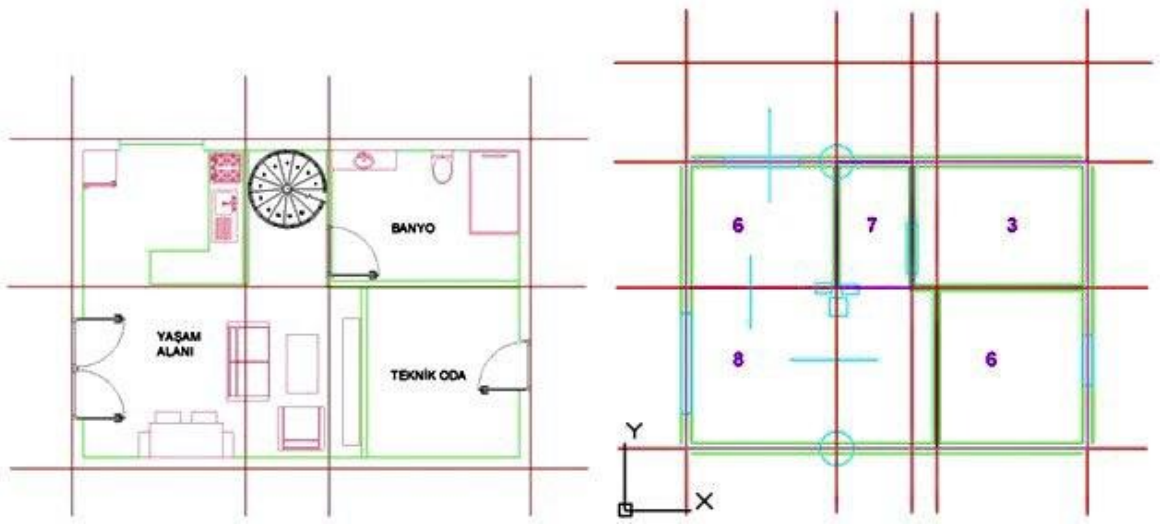
Şekil 5.20. Sunulan stratejiye ait akış şeması.

Şekil 5.20'de verilen sunulan stratejiye ait akış şemasında, sonlandırma kriterini açıklamakta fayda vardır. Bu çalışmada sonlandırma kriteri çalıştırma miktarı olarak belirlenmiştir. Örneğin; çalıştırma miktarı 10 olarak belirlenen bir sistem için sunulan

stratejiye bağlı olarak önce kiriş ve kolon topolojileri oluşturulur. Daha sonra boyut ve şekil optimizasyonları ile sistemin nihai durumu belirlenir ve oluşan tasarım bir önceki tasarım ile kıyaslanır. Kıyaslama sonucunda daha iyi olan aday seçilir ve bu işlem on kez yapılarak adaylar değerlendirilir. Çalıştırma miktarı sağlanınca durdurma kriteri de sağlandığı için o ana kadar üretilmiş en iyi aday algoritma tarafından seçilerek sunulur.

6. BULGULAR VE TARTIŞMA

Bahsedilen taşıyıcı sistem planı stratejisinin test edilmesi için Doç. Dr. Hakan ÖZBAŞARAN (Özbaşaran, 2019) yürütücülüğündeki 119M848 kodlu, "Konut Yapıları İçin Betonarme Taşıyıcı Sistem Planı Tasarımını Kolaylaştıracak Bir Uzman Sistem Geliştirilmesi" başlıklı TÜBİTAK projesi kapsamında geliştirilen yazılımın bir modülü kullanılmıştır.



(a) (b)

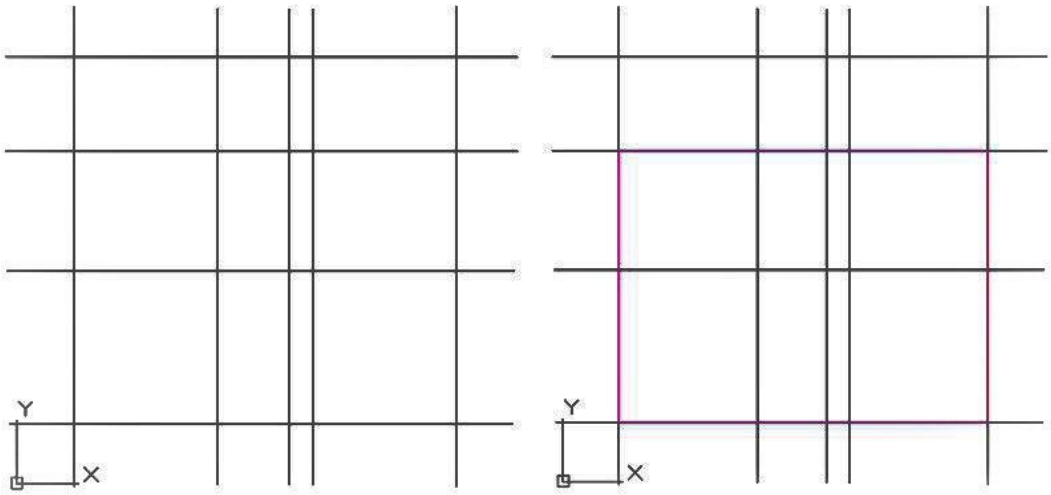
Şekil 6.1. Basit bir konut için mimari plan ve uygun biçime dönüştürülen "DXF" çizim: a) mimari proje, b) DXF çizim.

Kullanılan modülün mimari projeleri düzgün okuyabilmesi için mimari projelerin DXF biçimine dönüştürülmeleri ve DXF dosyalarının da aşağıdaki kurallara uygun olarak hazırlanmış olması gerekmektedir.

- Mimari projede belirlenen tüm akslar tek bir katmanda "line" komutu kullanılarak çizilmiş olmalıdır.
- Tüm duvarlar tek bir katmanda "line" komutu kullanılarak çizilmiş olmalıdır.
- Yapının oturma alanının ayrı bir katmanda "polyline" komutu ile çizilmiş olması gerekmektedir. Yapı kullanım alanının (açık ve kapalı çıkmalar dâhil) ayrı bir katmanda "polyline" komutu ile çizilmiş olması gerekmektedir.

- Taşıyıcı sistem elemanlarının taşmaması istenen bölgelerin ayrı bir katmanda "circle", "line" ve "polyline" komutları ile işaretlenmiş olması gerekmektedir.
- Her bağımsız bölümün içine ayrı bir katmanda ilgili bölümün ne kadar kıymetli olduğu "mtext" komutu kullanılarak yazılmalıdır. Burada "1" sayısı en düşük kıymet derecesidir.

Şekil 6.1'de örnek bir mimari plan ve ilgili DXF girdisi gösterilmiştir. Bu mimari plan (Şekil 6.1 (a)), basit bir konut projesi olarak düşünülebilir. Şekil 6.1 (b)'de görüleceği üzere yapının oturma ve kat alanı sınırları dışında faydasız bir aks daha konmuştur. Bu aks önerilen algoritmayı zorlamak içindir. Söz edilen DXF çizim dokuz adet aks ile tanımlanmış, yirmi adet aks kesişimi bulunmaktadır. Mümkün olan yirmi iki adet kiriş ve on beş adet kolon (faydasız aks üzerine gelecek elemanlar sayılmadı) vardır. Bu mimari planın düzenlenmiş ve Şekil 6.1 (b)'de verilen DXF çizimde akslar, duvarlar, taşıyıcı sistem elemanlarının taşmasını istemediğimiz bölgeler, döşemeler ve bu döşemelere ait kıymet dereceleri, yapının kat ve oturma alanları çizilmiştir. Aşağıdaki şekillerde sırasıyla çizilen elemanlar gösterilmiştir.

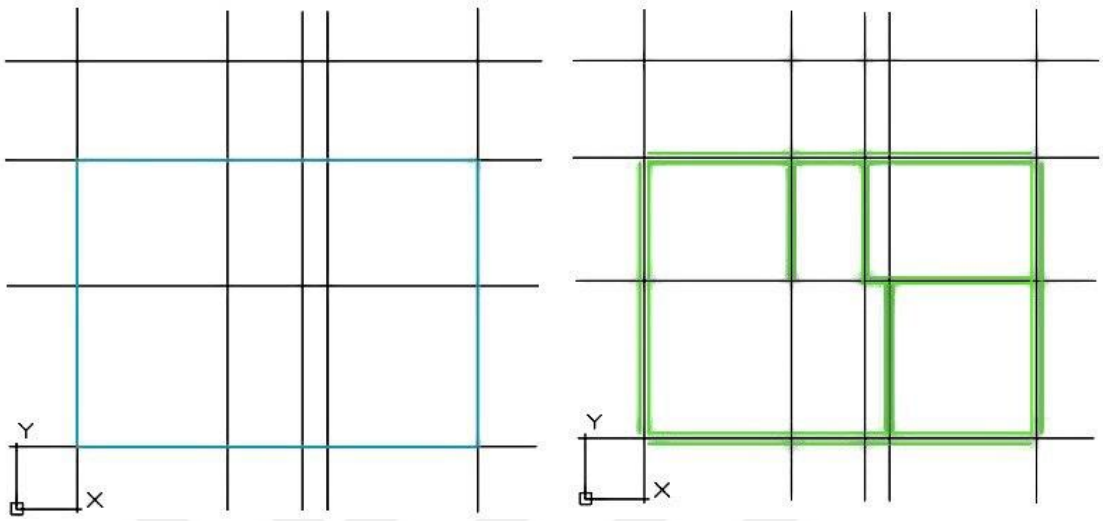


(a) (b)

Şekil 6.2. Mimari projenin DXF biçiminde düzenlenme aşamaları: a) aksların çizilmesi, b) yapı oturma alanının çizilmesi.

Şekil 6.2 (a)'da yapıya ait aks sistemi görülmektedir. Aks sistemi mimari plandan alınarak üzerinde gerekli ekleme ve çıkarmalar yapılır ve mimari çizim yazılımın

anlayacağı biçime getirilir. İlk olarak Şekil 6.2 (b)'de de görüldüğü üzere aks sistemi üzerine yapı oturma alanı çizilmiş böylece yapının sınırları belirtilmiştir.

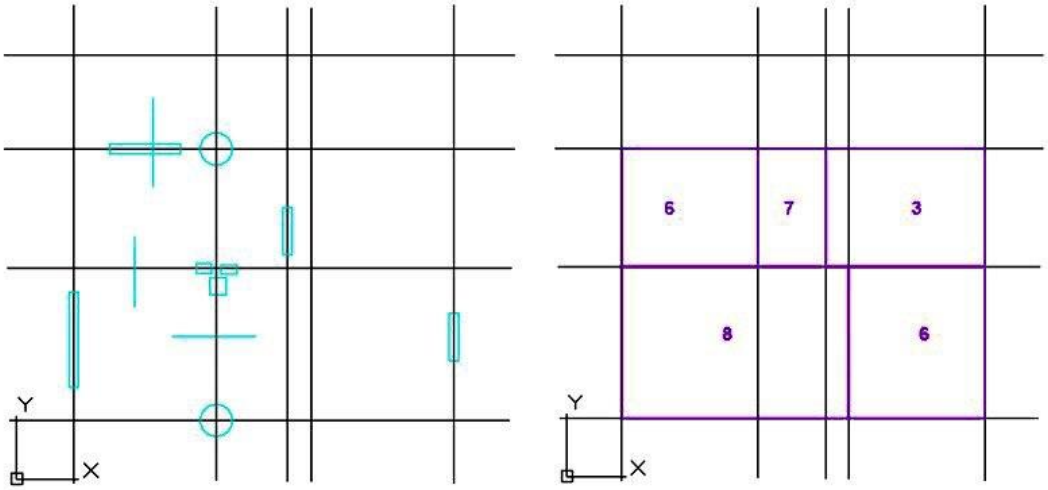


(a) (b)

Şekil 6.3. Mimari projenin DXF biçiminde düzenlenme aşamaları 2: a) yapıya ait kat alanının çizilmesi, b) yapıya ait duvarların çizilmesi.

Şekil 6.3'te mimari projenin uygun biçime getirilme işlemlerinin devam etiğini görebiliriz. Daha önce Şekil 6.2'de çizilen aks sistemi ve oturma alanına ek olarak yapıya ait kullanım alanı (Şekil 6.3 (a)) ve duvarlar (Şekil 6.3 (b)) çizilir. Yapı kullanım alanı oturma alanından farklı olarak açık ve kapalı çıkmalara sahip olabilir. Bu yüzden kullanım ve oturma alanlarının farklı katmanlarda çizilmesine dikkat edilmelidir.

Şekil 6.4'de yapılacak son çizimler görülmektedir. Şekil 6.4 (a)'da "daire", "çizgi" ve "düzgün poligon" şekillerinin olduğu görülmektedir. Daire şekli "kolon", çizgiler ise "kiriş" elemanlarının gelmesinin istenmediği yerler için kullanılmıştır. Düzgün poligon şekiller ise kapı ve pencereleri temsil eder. Mimari projede kapı ve pencerelerin olduğu yerler DXF çizim dosyası üzerinde "polyline" komutu ile çizilerek belirtilmelidir. Bununla birlikte Şekil 6.4 (b)'de döşemeleri ve bu döşemelere ait kıymet derecelerini görebiliriz. Herhangi bir alana döşeme çizmek için "polyline" komutunun kullanılarak alanın ve bu alana ait kıymet derecesinin tanımlanması gerekir. Kıymet derecesi, en kıymetli hacim için en yüksek değerde olmaldır. Örneğin; salon için kıymet derecesi "7" alınırken, oda için "5", banyo için "2" alınabilir.



(a) (b)

Şekil 6.4. Mimari projenin DXF biçiminde düzenlenme aşamaları 3: a) yapıda taşıyıcı sistem elemanlarının taşmasının istenmediği noktaların çizilmesi, b) yapıya ait döşemelerin ve bu döşemelere ait kıymet derecelerinin belirlenmesi.

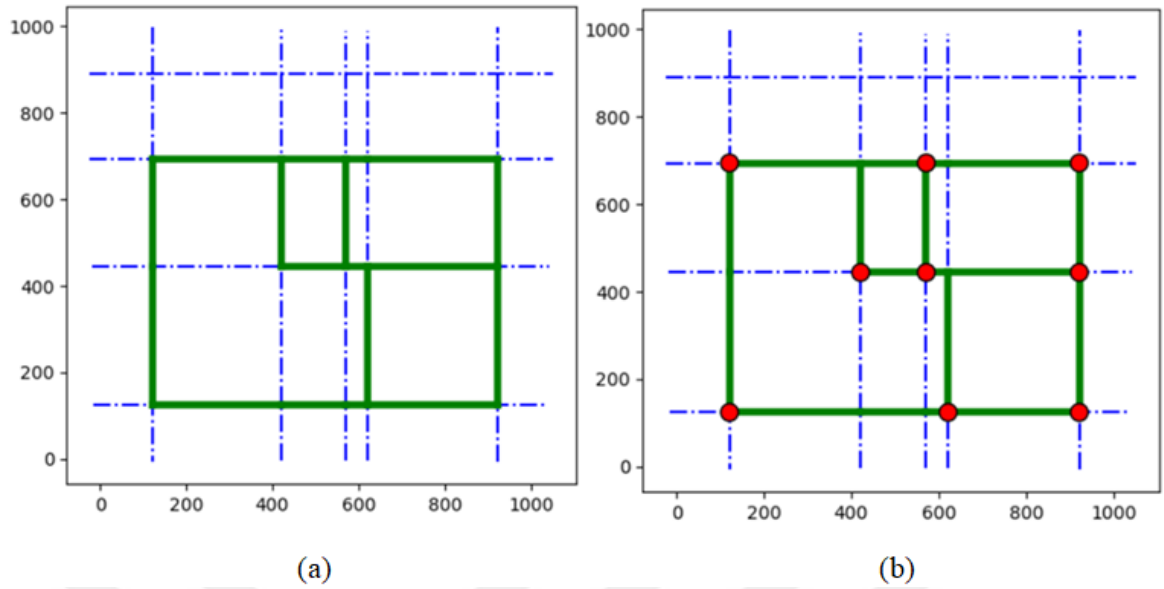
Şekil 6.2'den Şekil 6.4'e kadar belirtilen çizimler bir DXF dosyası üzerinde tamamlandıktan sonra mimari proje kullanılacak biçime getirilmiş olur. Bu aşamadan sonra analiz sürecine geçilir ve mimari projeye uygun taşıyıcı sistem planları üretilir. Daha önce Bölüm 3'te verilen tüm tasarım pratikleri kısıt olarak ele alınmıştır.

Sonraki bölümlerde Bölüm 6'nın başında tanıtılan verilen mimari proje (bkz. Şekil 6.1) de dâhil olmak üzere 4 adet örnek ile bu örnekler için üretilen taşıyıcı sistem planları sunulacaktır. Her örnek için 30 adet taşıyıcı sistem planı üretilmiştir ve bu planlar kullanılan yöntemin doğası gereği uygunluk bakımından birbirinin aynısı değillerdir.

6.1. Örnekler

6.1.1. Basit bir konut projesi

İlk örnek Bölüm 6'nın başında da tanıtılan (bkz. Şekil 6.1) basit bir konut projesidir. Bu proje için analiz sürecinde popülasyon büyüklüğü 9 olarak seçilmiş ve her aşamada 9000 adayın değerlendirilmesine izin verilmiştir. Her aşamada izin verilen toplam iterasyon sayısı 1000 iken bir tasarımın oluşturulma süresi ortalama 32 dakika olarak kaydedilmiştir. Bu durumda üretilen en iyi çözümler aşağıdaki gibi sıralanmıştır.

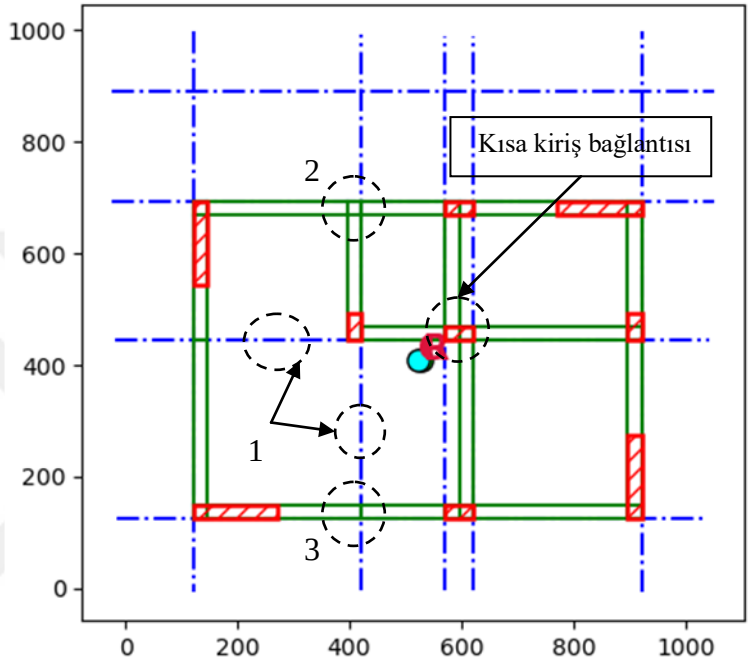


Şekil 6.5. Basit bir konut için üretilen iyi 1. plan: a) yapıya ait kiriş topolojisi, b) yapıya ait kolon topolojisi.

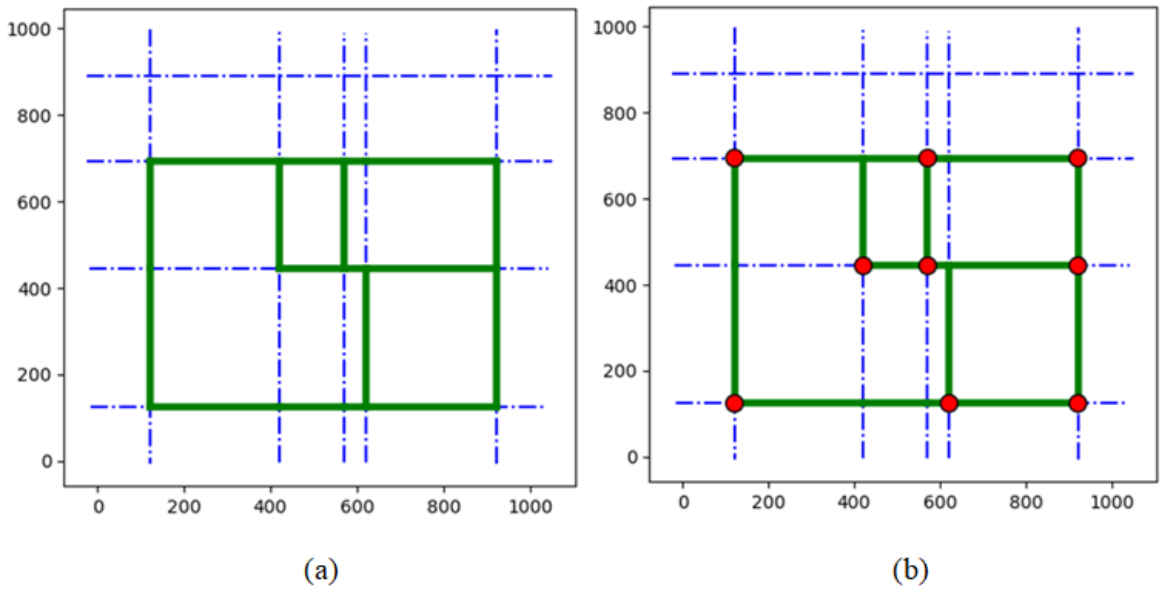
Sunulan strateji daha önce Bölüm 5.2'de de değinildiği üzere öncelikle yapıya ait kiriş topolojisini belirler. Kiriş topolojisi belirlenen sistem için daha sonra kolon topololjisi belirlenir. Bu şekilde taşıyıcı sistem elemanlarının yapı üzerindeki varlıkları tespit edilir. Bu aşamadan sonra belirlenen kiriş ve kolonların boyut ve yerleşim durumlarına geçilir. Optimizasyon algoritması birçok deneme ile amaç ve ceza değerlerine göre en iyi sisteme karar verir ve sonuçta en iyi uygunluk derecesine sahip plan ortaya çıkar. Şekil 6.5'te basit konut projesi için üretilen en iyi 1. tasarım için kiriş ve kolon topoloji planları verilirken, Şekil 6.6'da planın son hali gösterilmiştir.

Şekil 6.6'yı yakından inceleyelim. Üretilen bu sistemin uygulanabilir bir sistem olduğu açıktır. İlk olarak değinmek gerekir ki basit konut projesinde "1" olarak işaretlenen doğrultularda kirişlerin ve "2" ve "3" olarak gösterilen kesişim noktalarında ise kolonların eksik olduğu görülür. Bu eksikler mimari planın uygun biçime getirilme aşamasında algoritmayı zorlamak adına tercihen tanımlanmış yasaklı bölgelerden dolayı oluşmuştur. Bu noktalar Şekil 6.4 (a)'da görülen doğru parçaları (kirişler için) ve daireler (kolonlar için) ile yasaklı bölgeler olarak tanımlanarak o doğrultu ve kesişim noktalarında herhangi bir elemanın gelmesi engellenmiştir. Bunun yanı sıra Şekil 6.6'da görülen planda perdeler simetrik yerleştirilmiş, rijitlik (R) ve kütle (G) merkezleri neredeyse çakışıktır. Dahası her iki yönde minimum iki perde ve sistemdeki tüm perdelerin dış asklara yerleştirilmesi

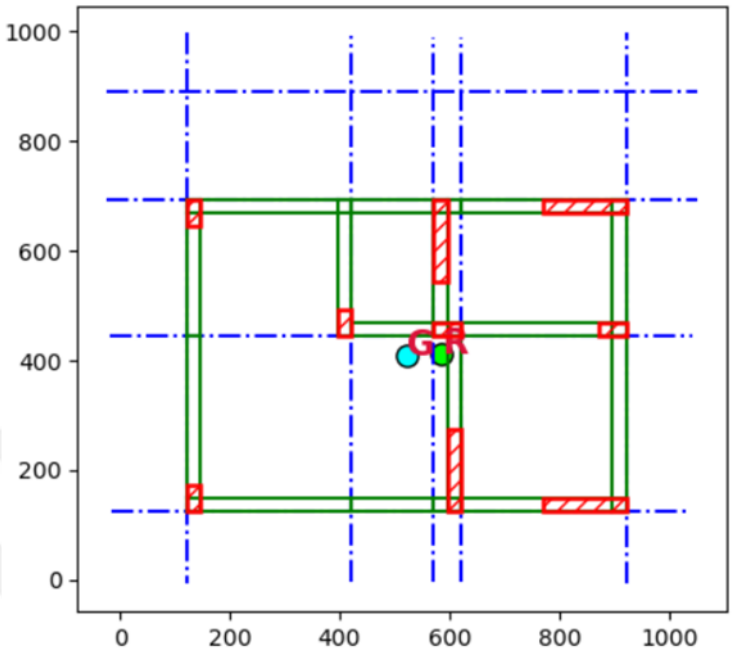
kuralları da uygulanarak daha güvenli bir taşıyıcı sistem tasarımı ortaya çıkmıştır. Şekilde görülen kısa kiriş bağlantısı optimizasyon algoritması tarafından kolona dönüştürülerek uygulama projelerinde çok sık rastlanılan bir özellik de sağlanmıştır. Bu şekilde uygulamada birebir uygulanabilir bir projenin otomatik tasarım sonucu ortaya çıkması gerçekten umut vericidir.



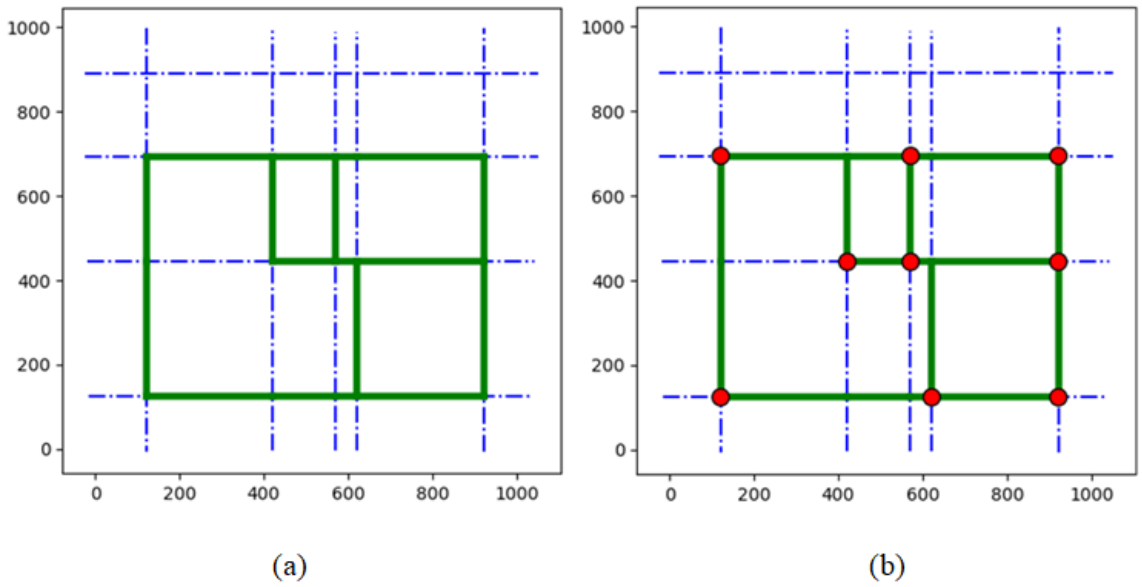
Şekil 6.6. Basit bir konut için üretilen en iyi 1. planda kiriş ve kolonların nihai durumları.



Şekil 6.7. Basit bir konut için üretilen en iyi 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.

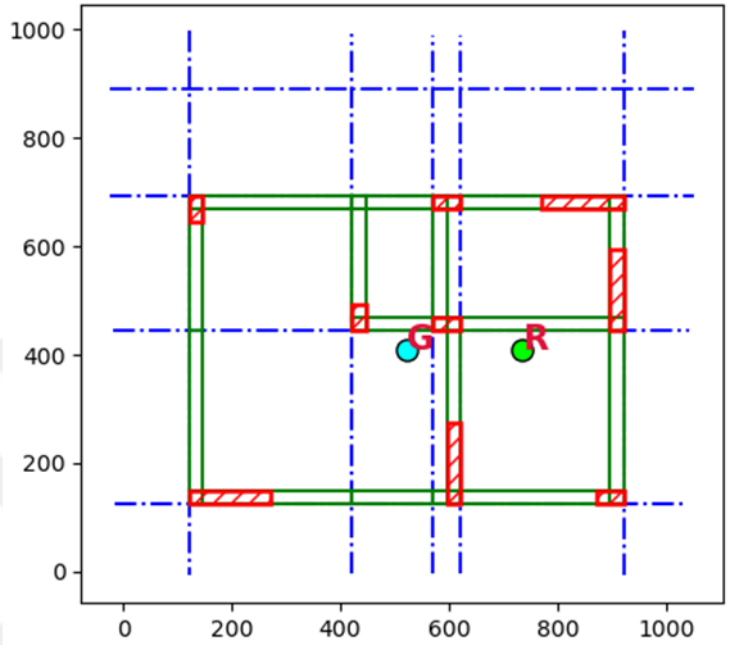


Şekil 6.8. Basit bir konut için üretilen en iyi 2. planda kiriş ve kolonların nihai durumları.



Şekil 6.9. Basit bir konut için üretilen en iyi 3. plan: a) 3.plana ait kiriş topolojisi, b) 3. plana ait kolon topolojisi.

Benzer şekilde basit bir konut için üretilen diğer en iyi iki çözüme ait kiriş ve kolon topolojileri sırasıyla Şekil 6.7 ve Şekil 6.9'da, planların son durumları ise sırasıyla Şekil 6.8 ve Şekil 6.10'da verilmiştir.

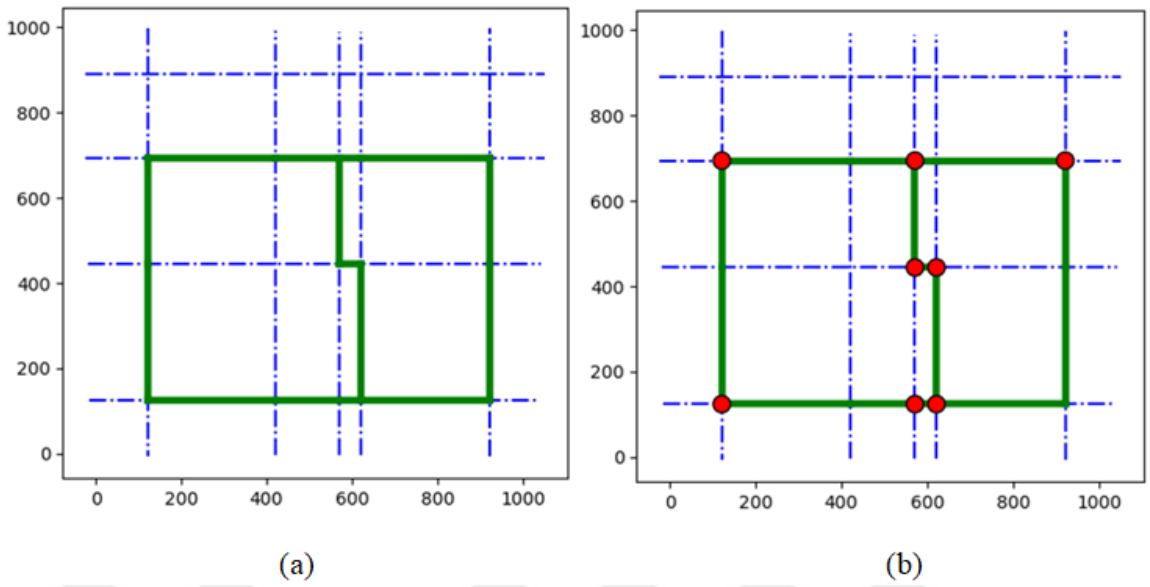


Şekil 6.10. Basit bir konut için üretilen en iyi 3. planda kiriş ve kolonların nihai durumları.

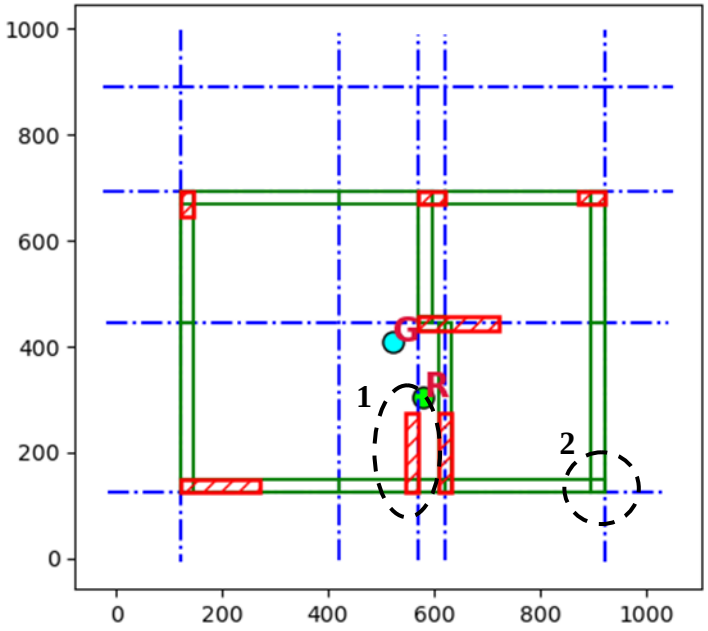
Bu noktada ifade etmek gerekir ki sunulan en iyi üç plan için de kiriş ve kolon topoloji planları aynıdır. Ancak boyut ve yerleşim durumları incelendiğinde nihai durumda planların birbirinden farklı olduğu görülmektedir. Bu durum optimizasyon algoritması tarafından yapılan çok sayıda denemenin bir sonucudur. Algoritma veri tabanındaki bilgilere bağlı her seferinde farklı çözümleri deneyerek en iyi amaç ve ceza değerine sahip sistemi seçer. Bu şekilde mühendise birbirinden farklı tasarımlar sunularak istenilen doğrultudaki planın seçilmesinin hedeflendiği söylenebilir.

Basit bit konut projesi için üretilen en iyi planların hepsi uygulanabilir planlardır. Bu yapıya ait en iyi planlar verildikten sonra şimdi de üretilen en kötü planlara göz atalım. Aşağıda basit konut projesine ait üretilen en kötü planlar sıralanacaktır.

Şekil 6.11'de üretilen en kötü 1. plana ait kiriş ve kolon topolojisi gösterilmiştir. İlk olarak verilen kiriş ve kolon topolojileri en iyi planlarda verilen topolojilerden farklıdır.



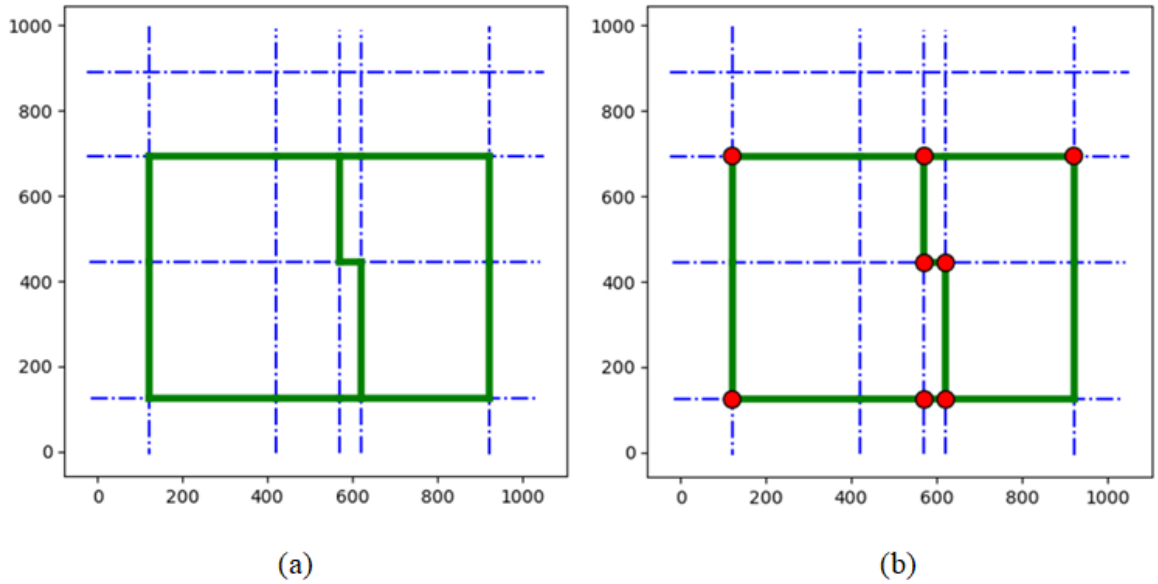
Şekil 6.11. Basit bir konut için üretilen en kötü 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi.



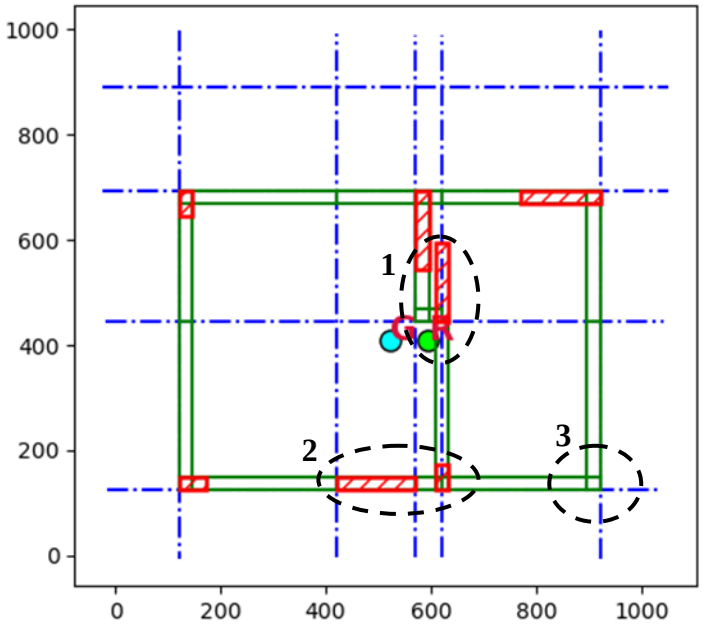
Şekil 6.12. Basit bir konut projesi için üretilen en kötü 1. planda kiriş ve kolonların nihai durumları.

Şekil 6.12'de son durumlar incelendiğinde ise kusurlu bir taşıyıcı sistem planının oluştuğunu söyleyebiliriz. Planda 1 olarak işaretlenen duruma bakıldığı zaman salon gibi kıymetli bir alan içerisine bir perdenin taştığı görülmektedir. Uygulamada kesin olarak istenmeyen bu durum bu planda ortaya çıkmıştır. İkincisi, 2 numaralı durumda köşe kolonun eksik olduğunu görmekteyiz. Bu şekilde bir uygulamanın da gerçek projelerde

olmayacağı açıktır. Bu plan sahip olduğu olumsuz durumlardan kaynaklı olarak uygulanabilir değildir.



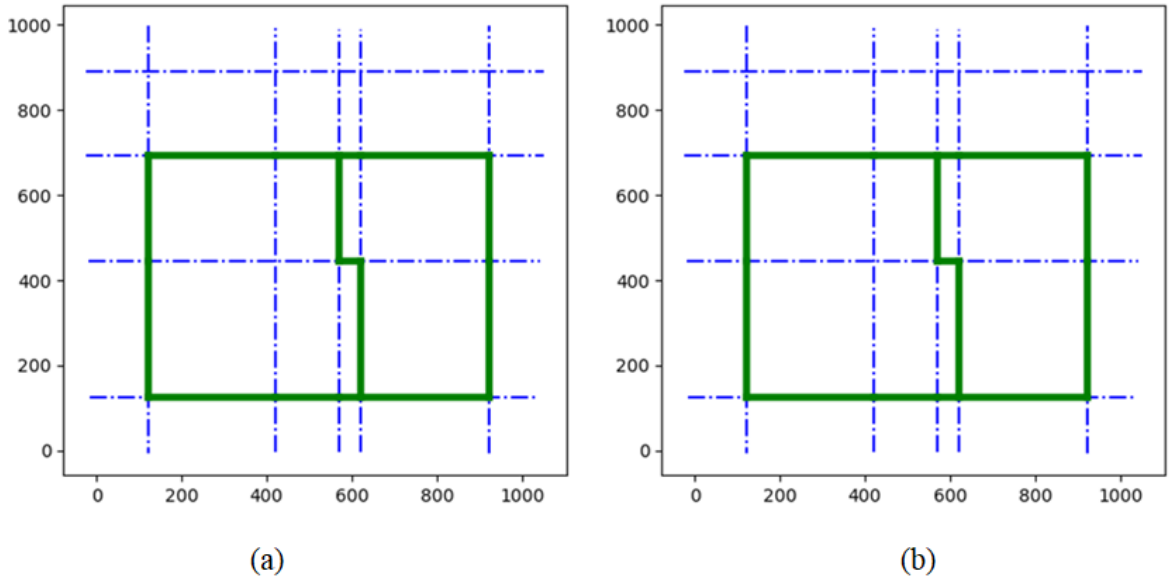
Şekil 6.13. Basit bir konut için üretilen en kötü 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.



Şekil 6.14. Basit bir konut projesi için üretilen en kötü 2. planda kiriş ve kolonların nihai durumları.

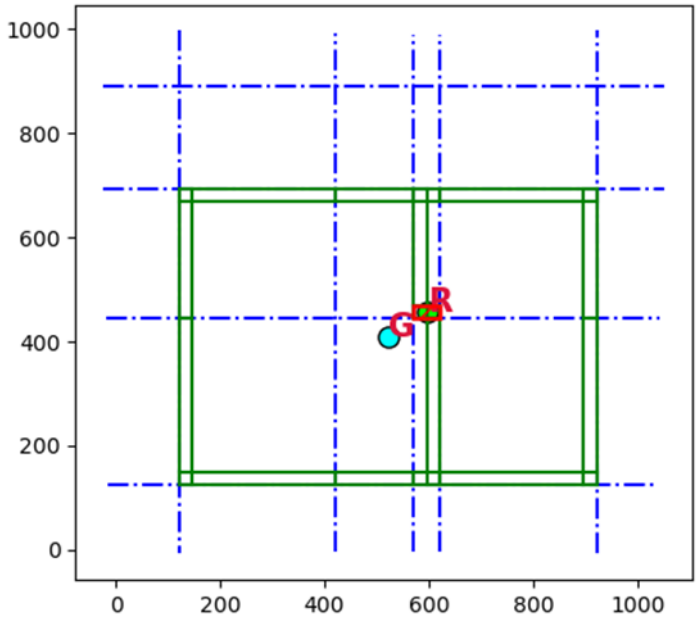
Şekil 6.13'te üretilen en kötü 2. tasarıma ait kiriş ve kolon topolojileri ve Şekil 6.14'de ise tasarımın son hali görülmektedir. Bu tasarıma ait olumsuz durumlar şöyledir:

1. Birbirine çok yakın iki perde oluşmuştur. İkincisi, bir perde mimari planda "banyo" olarak belirtilen alanın içerisine taşmıştır.
2. Bu noktada perde ile kolonun birbirine çok yakın olduğunu görebiliriz. Uygulamada iki düşey taşıyıcı elemanın birbirine çok yakın yerleşmesi tercih edilmez.
3. Köşe kolon eksiktir. Bundan dolayı iki kiriş havada asılı durmaktadır.



Şekil 6.15. Basit bir konut için üretilen en kötü 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.

Şekil 6.15'te ise üretilen en kötü 3. plana ait kiriş ve kolon topolojisi görülmektedir. Şekil 6.15 (b)'de kolon topolojisi incelendiği zaman sistemde herhangi bir elemanın olmadığı görülür. Buna bağlı olarak Şekil 6.16'da de herhangi bir düşey taşıyıcı eleman olmadan proje son halini almıştır. Bu şekilde bir tasarım mümkün olmadığı için yazılım tarafından sunulan en kötü plandır.

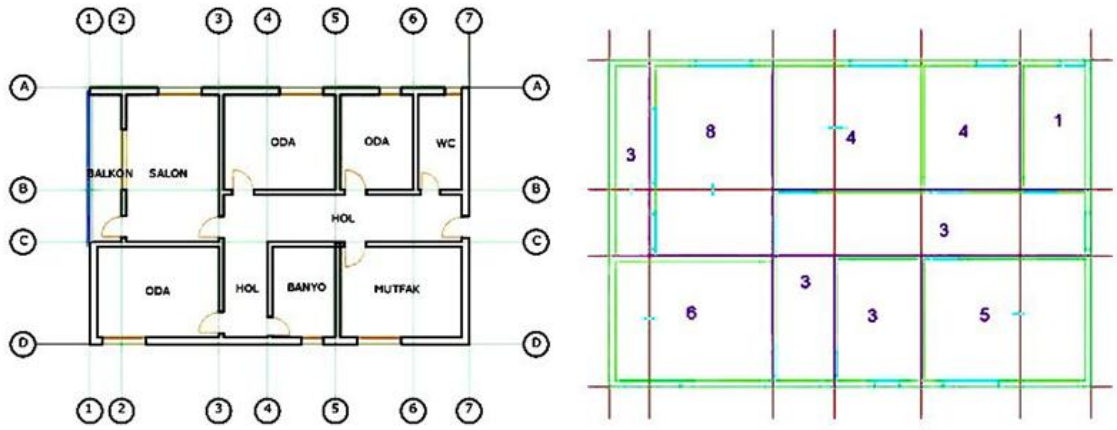


Şekil 6.16. Basit bir konut için üretilen en kötü 3. planda sistemin son hali.

6.1.2. Düzgün mimari projeye sahip bir konut

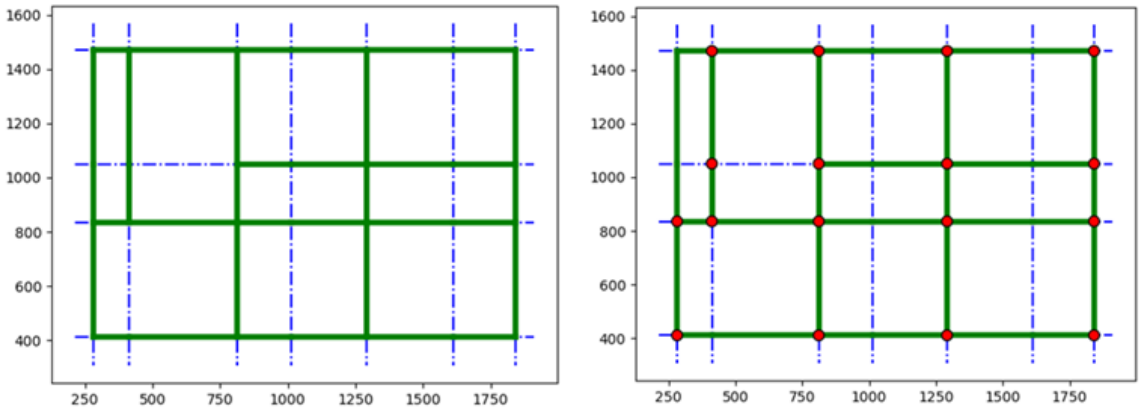
Basit bir konuta ait tasarımlar incelendikten sonra şimdi de analiz edilen diğer bir mimari örnek sunulacaktır. Şekil 6.17 (a)'da verilen mimari proje düzgün bir mimari plana sahip bir konuttur. Bu projede toplamda 11 adet aks çizgisi, mümkün 45 adet kiriş ve 28 adet kolon vardır. Bu mimari planın DXF biçimine getirilmiş hali ise Şekil 6.17 (b)'de verilmiştir. Bu çizimde akslar, duvarlar, taşıyıcı sistem elemanlarının taşmasını istemediğimiz bölgeler, döşemeler ve bu döşemelere ait kıymet dereceleri, yapının kat ve oturma alanları çizilmiştir. Uygun biçime getirilen proje daha sonra analiz edilerek taşıyıcı sistem planları üretilir. Bu proje için analiz sürecinde popülasyon büyüklüğü 40 olarak belirlenmiş ve her aşamada 80000 adayın değerlendirilmesine izin verilmiştir. Her aşamada izin verilen toplam iterasyon sayısı 2000 ve bir tasarımın oluşturulma süresi ortalama 6 saat 57 dakika olarak kaydedilerek toplamda 30 plan üretilmiştir. Önceki örnekte olduğu gibi üretilen taşıyıcı sistem planları arasında en iyi ve en kötü 3'er tasarım seçilmiştir. Üretilen en iyi üç tasarım da uygulanabilir çözümlerdir. Şekil 6.18'de üretilen en iyi 1.plana ait kiriş ve kolon topolojileri görülebilir. Topolojileri belirlenen elemanlar için boyutlar ve şekil düzenleri de ayarlandıktan sonra sistem son haline ulaşır. Buna göre sistemin aldığı nihai durum Şekil 6.19'da gösterilmiştir. Şekil 6.19 incelenirse her iki deprem yönünde de 2 adet olmak üzere toplamda 4 adet perde mevcut olup tüm perdeler

dış akslara yerleştirilmiştir. Planda, rijitlik ve kütle merkezleri arasındaki mesafenin istenildiği düzeyde olmadığı durumu hariç herhangi bir olumsuz durum söz konusu değildir.



(a) (b)

Şekil 6.17. Düzgün plana sahip bir konut için mimari plan ve uygun biçime dönüştürülen "DXF" çizim: a) mimari proje, b) DXF çizim.

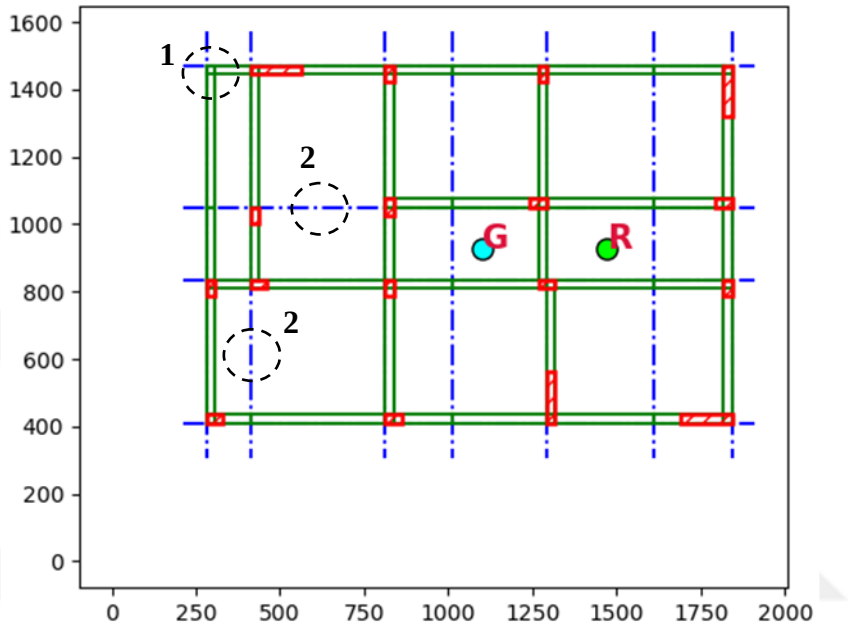


(a) (b)

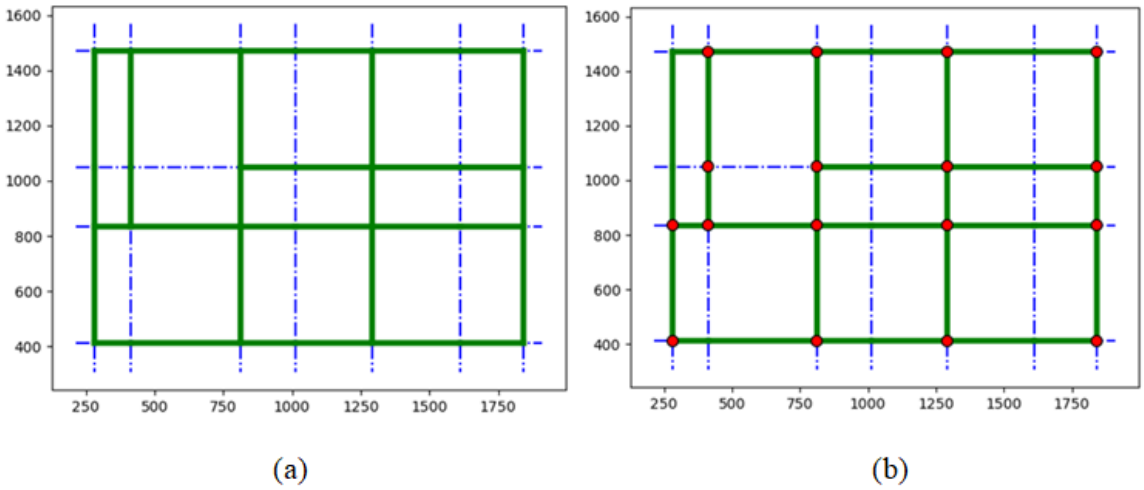
Şekil 6.18. Düzgün plana sahip bir konut projesi için üretilen en iyi 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi

Altı çizilecek diğer bir konu da Şekil 6.19 üzerinde "1" ve "2" olarak işaretlenmiş durumlardır. Öncelikle "1" olarak işaretlenen noktada köşe kolonun eksik olduğu görülmektedir. Ancak Şekil 6.17 (a) incelendiği zaman o alanın "balkon" olarak tanımlandığı görülür. Balkon bina oturma alanı dışında kalan bir bölge olduğu için o noktada kolon olması kolonun yapı sınırları dışına taşmasına ve yetkili kurumlar tarafından

yasal işlemlerin uygulanmasına sebebiyet verir. DXF planda çizilen yapı oturma alanı ile bu tür bir sorunun oluşması engellenmiştir. Bunun yanında "2" olarak işaretlenen noktalarda kirişlerin olmadığı ve çerçevelerin tamamlanmadığı görülür. Buna yol açan durum ise Şekil 6.17 (b)'de görüleceği üzere o noktalarda kiriş gelmesini engellemek adına (tercihen) çizilmiş doğru parçalarıdır.

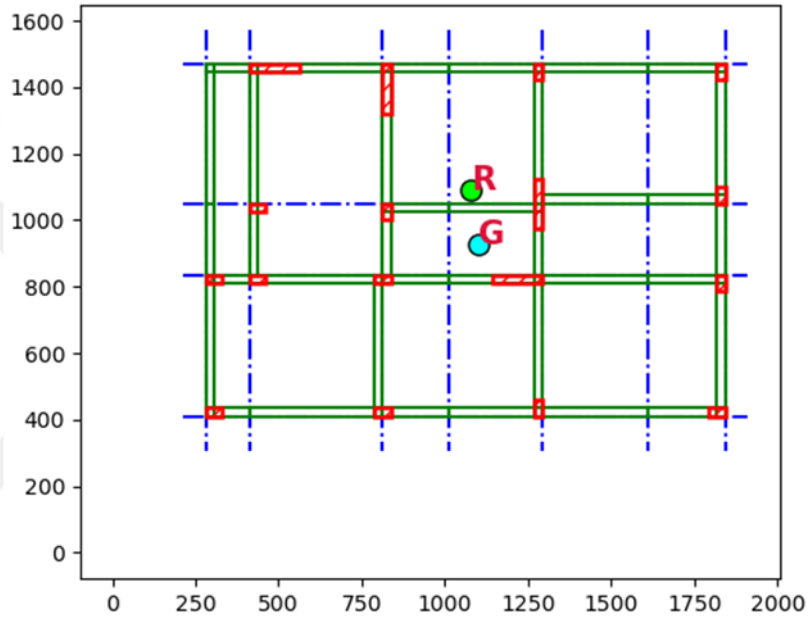


Şekil 6.19. Düzgün plana sahip konut projesi için üretilen en iyi 1. planın son hali.

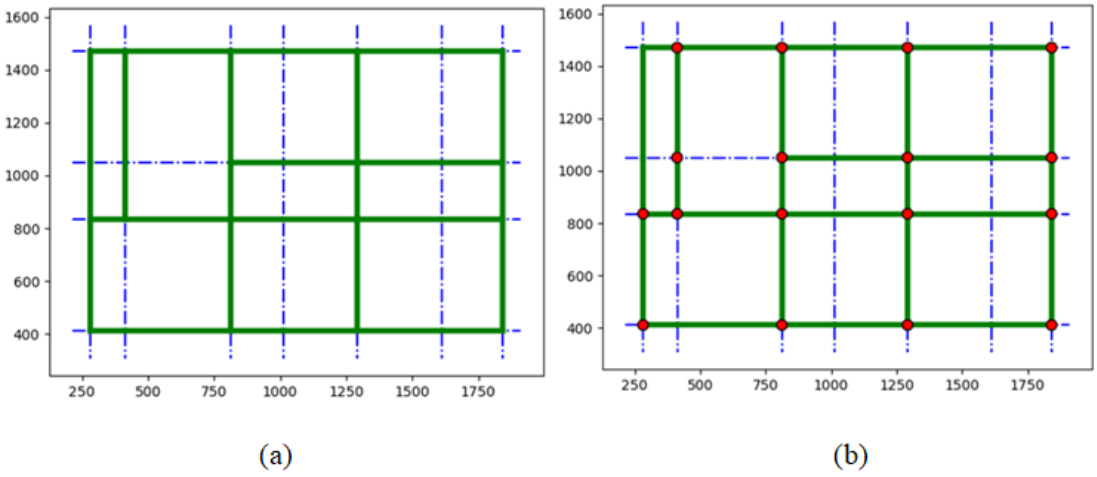


Şekil 6.20. Düzgün plana sahip konut projesi için üretilen en iyi 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.

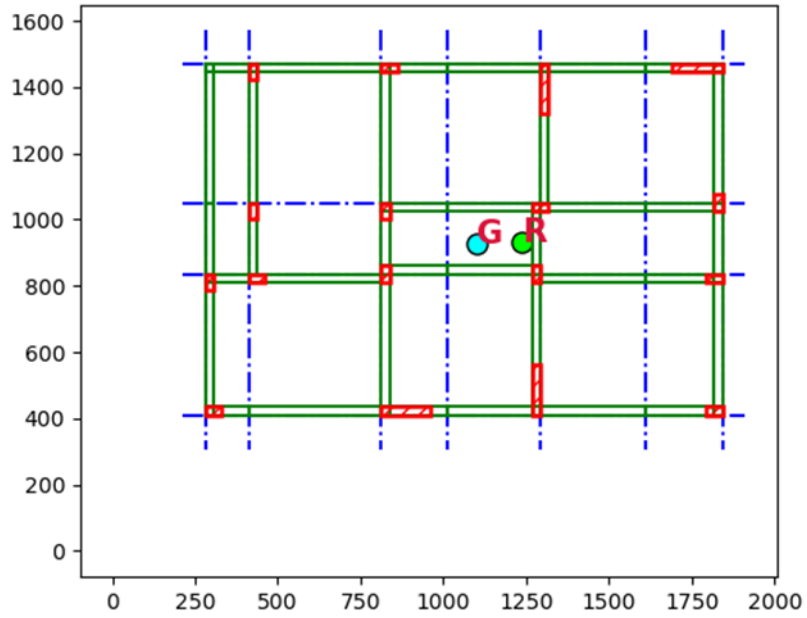
Düzgün plana sahip konut projesi için en iyi 2. plana ait kiriş ve kolon topolojileri Şekil 6.20'de verilmiştir. Şekil 6.18'de birinci tasarıma ait kiriş ve kolon topolojileri ile Şekil 6.20'de ikinci plana ait topolojiler incelediği zaman sistemlerin aynı olduğu görülür. Ancak tasarımlara ait son durumlar (Şekil 6.19 ve Şekil 6.21) farklıdır. Bu durum daha öce de değinildiği üzere optimizasyon algoritmasının doğasından kaynaklı birçok denemenin sonucudur. Şekil 6.21'de verilen plan incelendiği zaman uygulanabilir bir proje olduğu görülür. Kütle ve rijitlik merkezleri arasındaki mesafe ilk tasarıma göre de daha azdır.



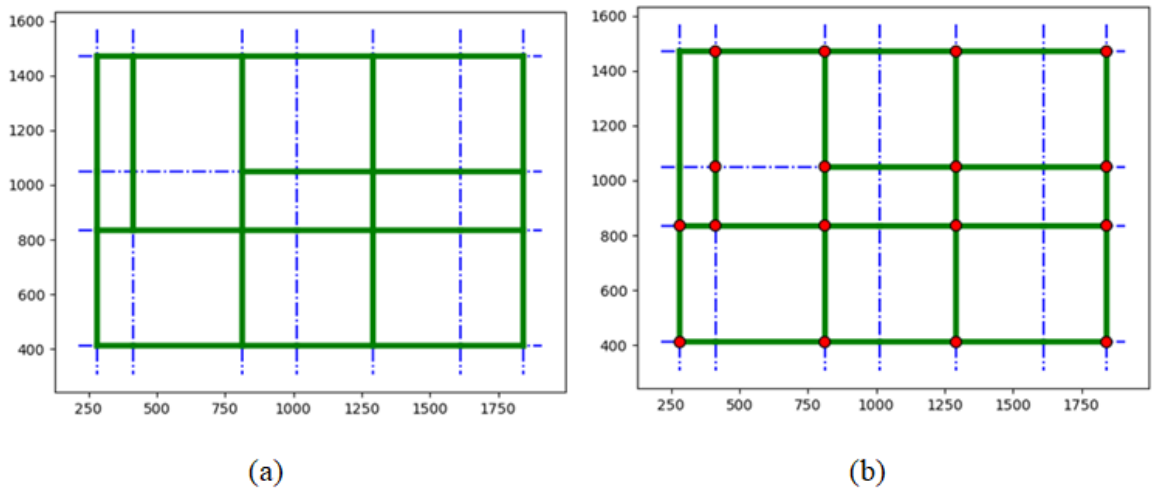
Şekil 6.21. Düzgün plana sahip konut projesi için üretilen en iyi 2. planın son hali.



Şekil 6.22. Düzgün plana sahip konut projesi için üretilen en iyi 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.



Şekil 6.23. Düzgün plana sahip konut projesi için üretilen en iyi 3. planın son hali.

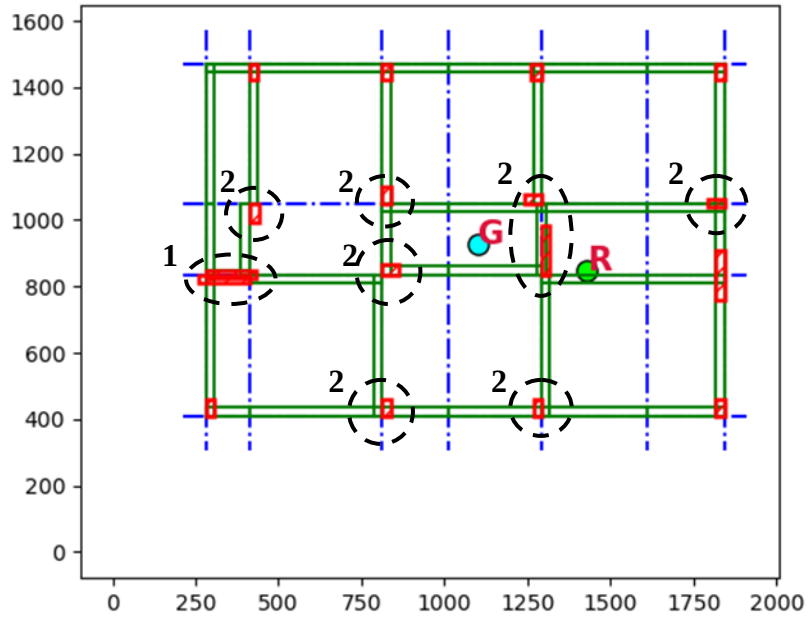


Şekil 6.24. Düzgün plana sahip konut projesi için en kötü 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi.

Düzgün plana sahip konut projesi için üretilen en iyi planlar incelendikten sonra şimdi de en kötü üç tasarım sıralanacaktır. Buna göre Şekil 6.25'te üretilen en kötü 1. plan verilmiştir. Şekil 6.24'de bu plana ait kiriş ve kolon topoloji planı gösterilmiştir. Verilen topoloji planları incelendiği zaman, en iyi planlarla aynı olduğu görülür. Ancak boyut ve

yerleşim durumlarının da belirlenmesiyle oluşan planın son hali farklı bir tasarımı ortaya çıkarmıştır. Şekil 6.25'te verilen plan incelendiğinde var olan olumsuz durumlar aşağıdaki gibi sıralanabilir:

1. Şekilde görülen ve "1" numara olarak işaretlenen durumda iki perde çakışıktır. Bu şekilde bir durum uygulamada mümkün değildir.
2. Diğer bir olumsuz durum ise "2" numarayla belirtildiği üzere taşan kirişlerdir. Bölüm 2'de değinildiği üzere betonarme yapı tasarımında kirişlerin bağlandıkları kolonlarla çakışık eksenli olması beklenir. Kiriş eksenlerinin kolon eksenleriyle çakışmaması durumunda kirişler tam olarak kolonlara denk gelmedikleri için kolon sınırlarından taşarlar. Bu düzensizlik güvenli yapı tasarımını engelleyen olumsuz bir durumdur.

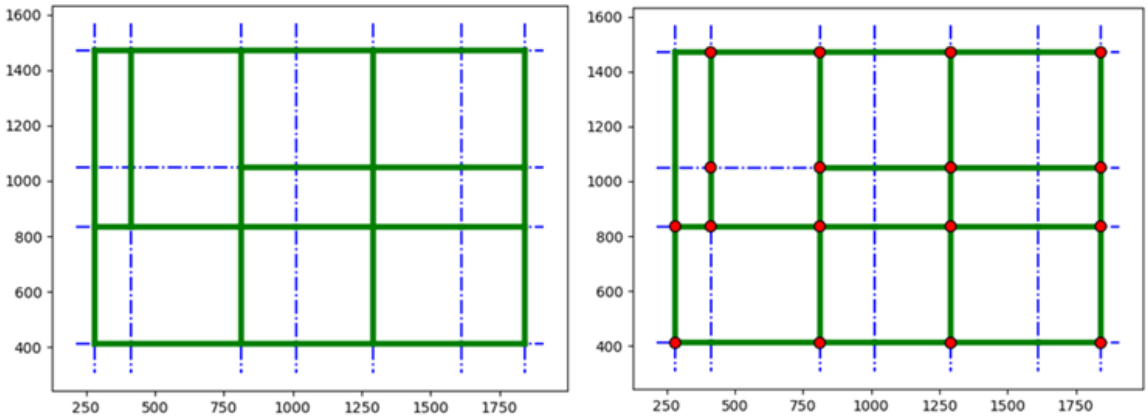


Şekil 6.25. Düzgün plana sahip konut projesi için üretilen en kötü 1. planın son hali.

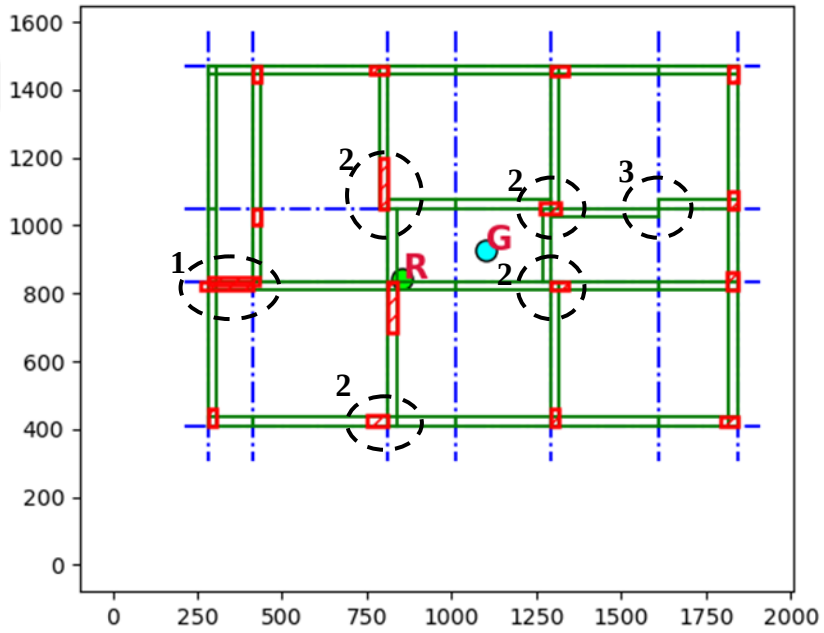
Şekil 6.26'da en kötü 2. plana ait topolojiler görülmektedir. Şekil 6.27'de ise son hali verilen sisteme ait olumsuz durumlar aşağıdaki gibi sıralanabilir:

1. Şekilde görülen ve "1" numara olarak işaretlenen durumda iki perde çakışıktır. Bu şekilde bir durum uygulamada mümkün değildir.
2. Diğer bir olumsuz durum ise "2" numarayla belirtildiği üzere taşan kirişlerdir.

3. Şekilde görülen ve "3" numara olarak belirtilen durum ise kiriş eksenlerinin çakışık olmamasıdır.



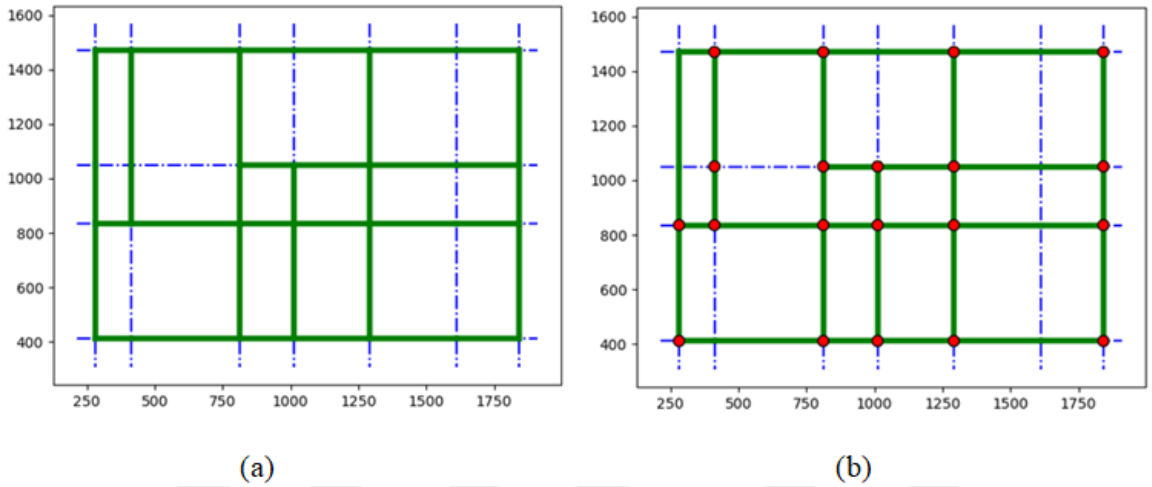
Şekil 6.26. Düzgün plana sahip konut projesi için üretilen en kötü 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.



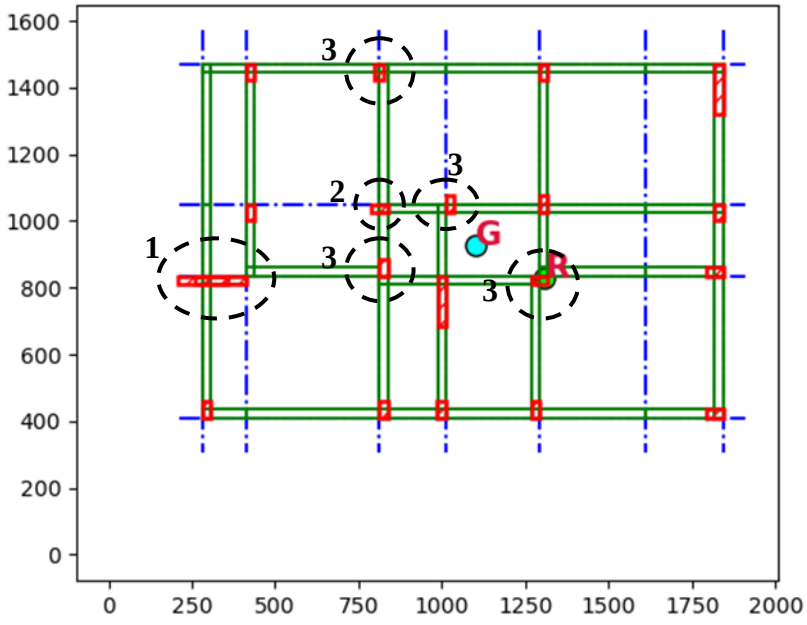
Şekil 6.27. Düzgün plana sahip konut projesi için üretilen en kötü 2. planın son hali.

Şekil 6.28'de en kötü 3. plana ait topolojiler görülmektedir. Şekil 6.29'da verilen sistem için olumsuz durumlar aşağıdaki gibi sıralanabilir:

1. Şekilde görülen ve "1" numara olarak işaretlenen durumda perde yapının oturma alanından taşmıştır. Mimari sınırları belli bir yapı için bir betonarme plan tasarımı yapılırken, mevcut bina sınırlardan herhangi bir yapı elemanının taşması söz konusu olamaz.
2. Şekilde "2" numara olarak görülen durumda kolon kullanım alanı içine taşmıştır.
3. Diğer bir olumsuz durum ise "3" numarayla belirtildiği üzere taşan kirişlerdir.



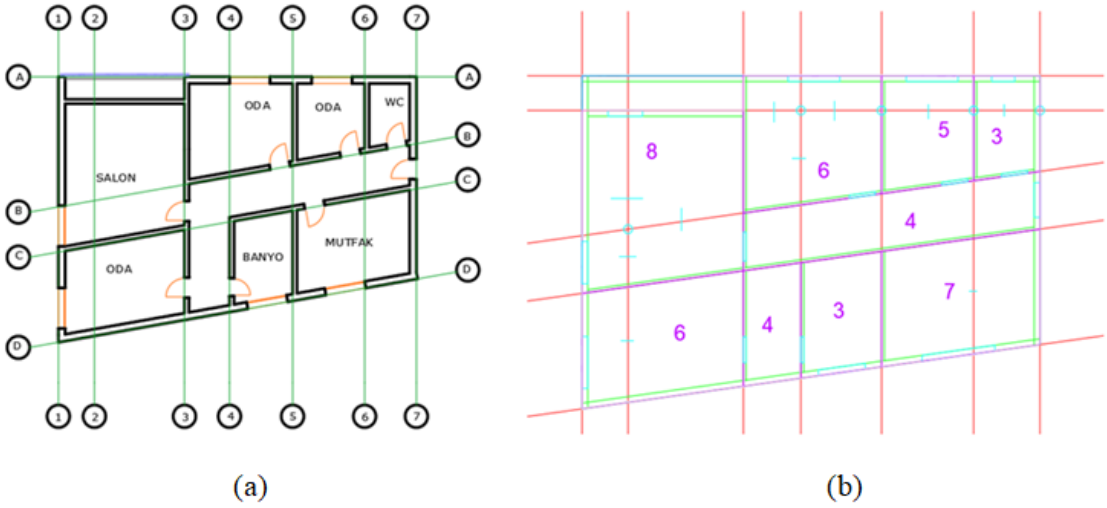
Şekil 6.28. Düzgün plana sahip konut projesi için üretilen en kötü 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.



Şekil 6.29. Düzgün plana sahip konut projesi için üretilen en kötü 3. planın son hali.

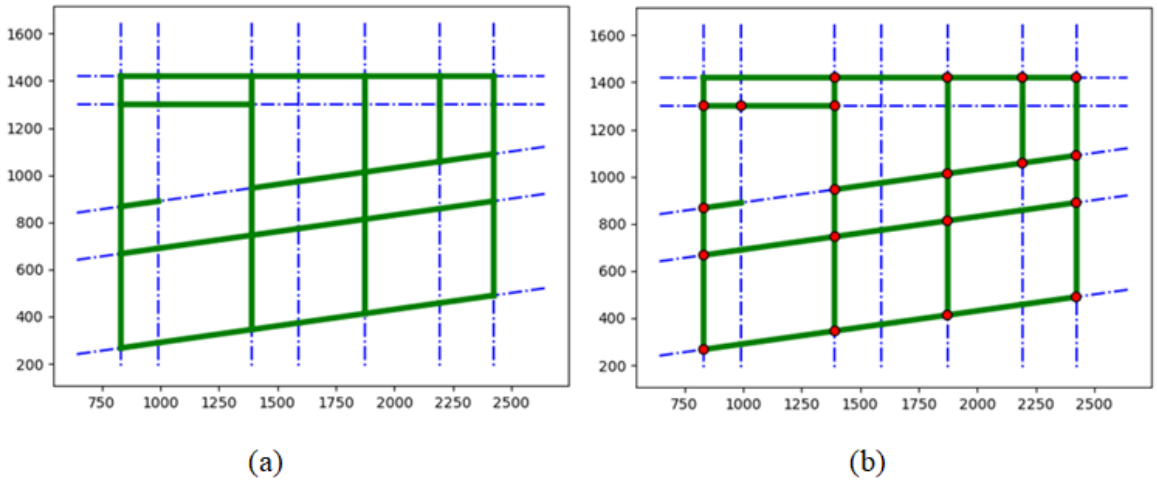
6.1.3. Yamuk mimari projeye sahip bir konut

Analiz edilen bir diğer proje mimarisi yamuk olan bir konut projesidir. Bu proje için uygulanabilir bir sistem elde edilememiştir. Ancak yapılacak ufak değişiklikler ile var olan olumsuz durumların ortadan kalkacağı düşünülmektedir. Bu proje için analiz sürecinde popülasyon büyüklüğü 46 olarak belirlenmiş ve her aşamada 92000 adayın değerlendirilmesine izin verilmiştir. Her aşamada izin verilen toplam iterasyon sayısı 2000 ve bir tasarımın oluşturulma süresi ortalama 8 saat 4 dakika olarak kaydedilerek toplamda 30 plan üretilmiştir.

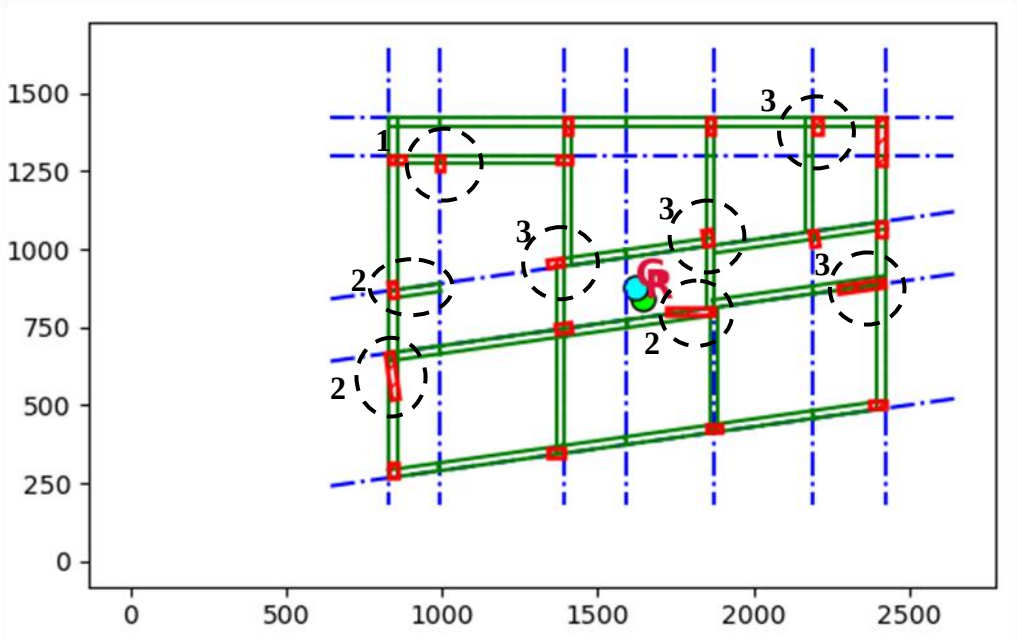


Şekil 6.30. Yamuk plana sahip konuta için mimari plan ve DXF çizim: a) mimari proje, b) DXF çizim.

Şekil 6.30'da yamuk projeye sahip konut için mimari ve DXF çizim görülmektedir. Bu proje bir önceki örnekteki verilere sahiptir ancak algoritmayı zorlamak amacıyla bazı akslar belli açılar kullanılarak çizilmiştir. Buna göre ilk olarak üretilen en iyi planlara göz atalım. Tekrardan ifade etmek gerekir ki bu örnekte uygulanabilir bir sistem ortaya çıkmamıştır. Şekil 6.31'de en iyi 1. plana ait topoloji planları verilirken, Şekil 6.32'de en iyi 1. plana ait son durum görülür.



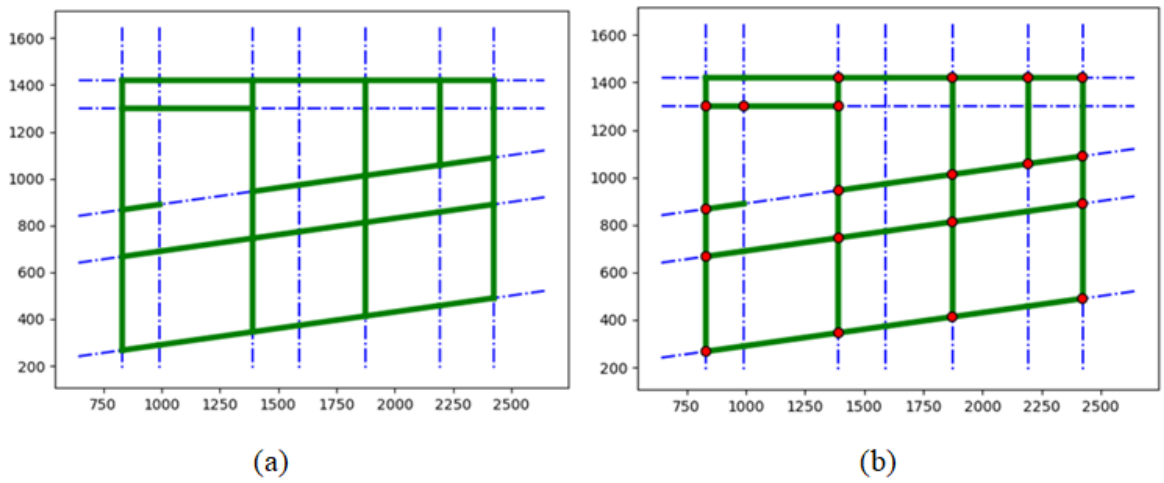
Şekil 6.31. Yamuk plana sahip konut projesi için üretilen en iyi 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi.



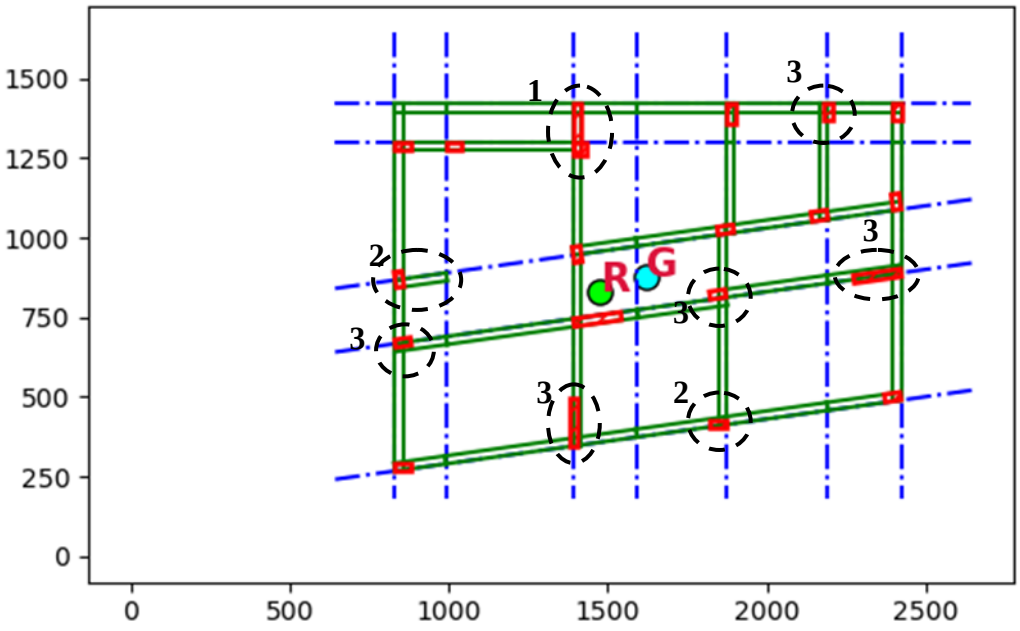
Şekil 6.32. Yamuk plana sahip konut projesi için üretilen en iyi 1. planın son hali.

Şekil 6.32'de görülen olumsuz durumlar aşağıdaki gibi sıralanabilir:

1. Şekilde görülen ve "1" numara olarak işaretlenen durumda kolon salon gibi önemli bir hacim içerisine taşmıştır.
2. Şekilde "2" numara olarak görülen durumda perde aksa düzgün yerleşmemiştir.
3. Diğer bir olumsuz durum ise "3" numarayla belirtildiği üzere taşan kirişlerdir.



Şekil 6.33. Yamuk plana sahip konut projesi için üretilen en iyi 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.

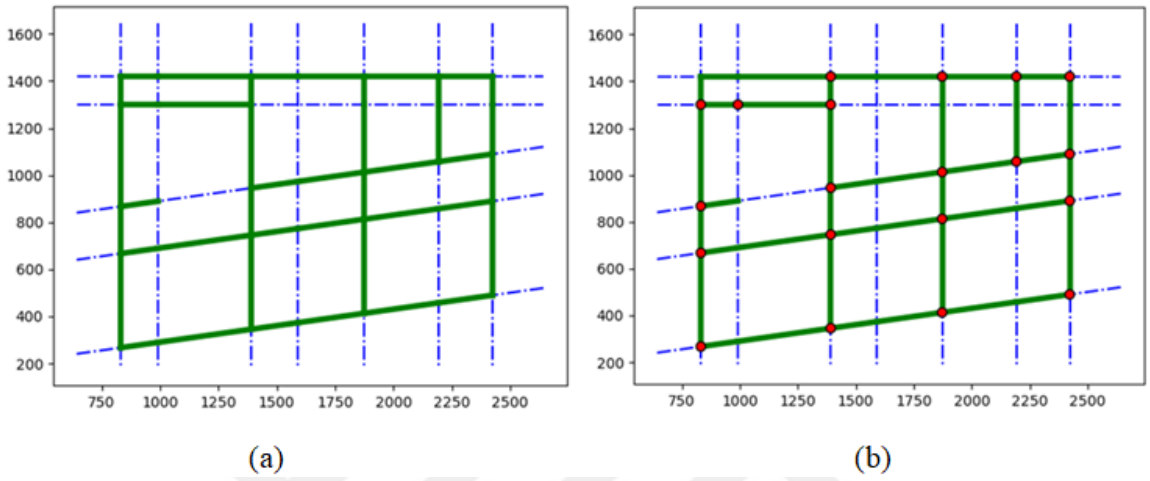


Şekil 6.34. Yamuk plana sahip konut projesi için üretilen en iyi 2. planın son hali.

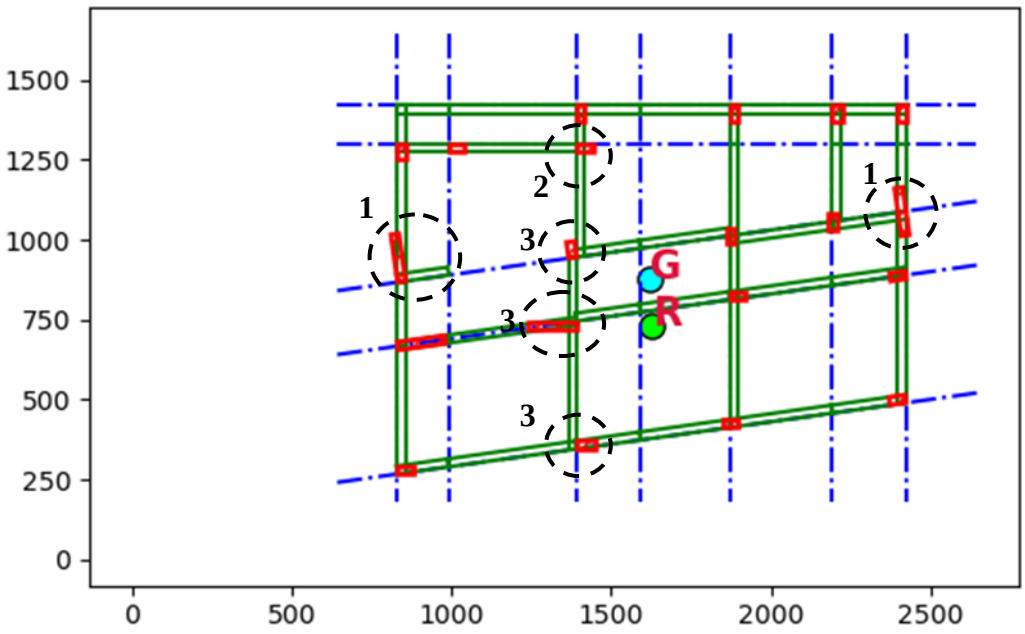
Şekil 6.33'te yamuk plana sahip konut projesi için üretilen en iyi 2. plana ait topoloji planları, Şekil 6.34'de planın son hali verilmiştir. Buna göre en iyi 2. planda (Şekil 6.34) görülen olumsuz durumlar şöyledir:

1. Düşey taşıyıcı elemanlar (perde ve kolon) çakışmıştır.

2. Şekil üzerinde "2" numaralı olarak işaretlenen durumda öncelikle kolonun aksa düzgün bir şekilde yerleşmediğini görürüz. Bu noktada var olan diğer bir durum ise kirişin bir ucunda kolon olmamasından kaynaklı olarak havada asılı kalmasıdır.
3. Diğer bir olumsuz durum ise "3" numarayla belirtildiği üzere taşan kirişlerdir.



Şekil 6.35. Yamuk plana sahip konut projesi için üretilen en iyi 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.



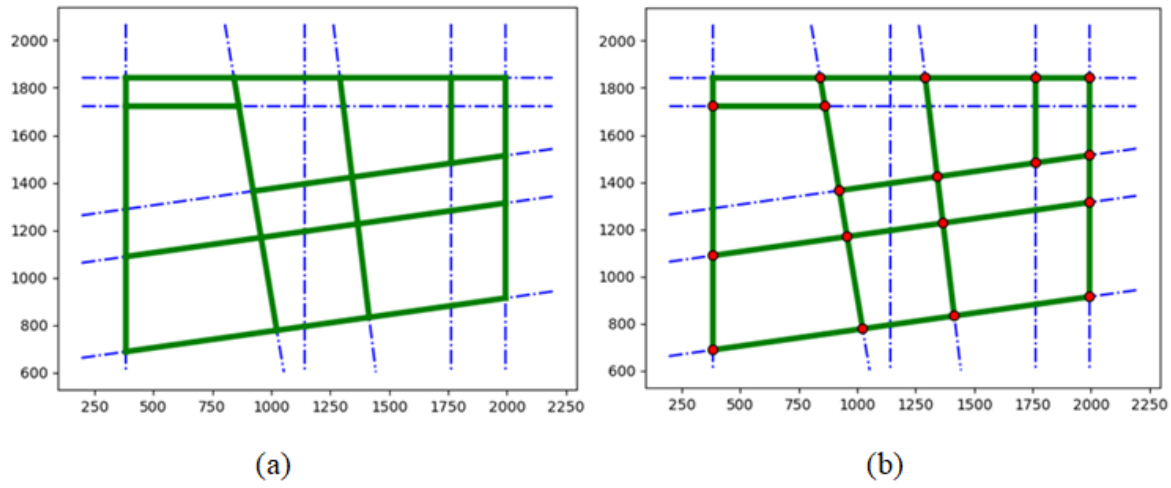
Şekil 6.36. Yamuk plana sahip konut projesi için üretilen en iyi 3. planın son hali.

Şekil 6.35'te yamuk plana sahip konut projesi için üretilen en iyi 3. plana ait topoloji planları, Şekil 6.36'da sistemin son hali verilmiştir. Buna göre 3. plana ait olumsuz durumlar şöyledir:

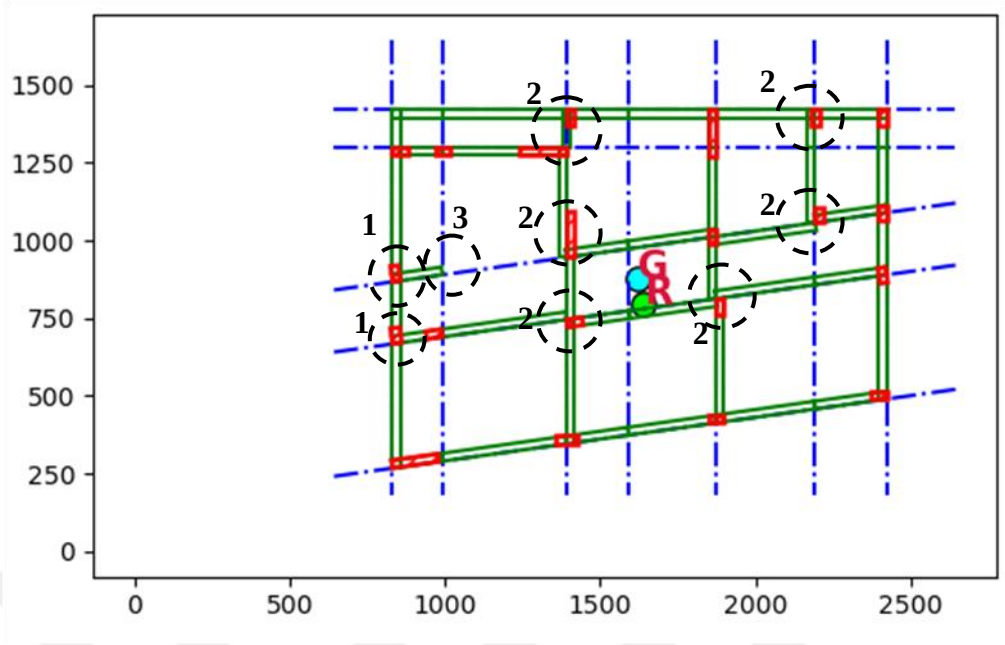
1. Şekil üzerinde "1" numaralı olarak işaretlenen durumda kolonun aksa düzgün bir şekilde yerleşmediğini görürüz.
2. Şekilde görülen ve "2" numara olarak işaretlenen durumda kolon odanın içerisine taşmıştır.
3. Diğer bir olumsuz durum ise "3" numarayla belirtildiği üzere taşan kirişlerdir.

Şekil 6.37'de yamuk plana sahip konut projesi için üretilen en kötü 1. plana ait topoloji planları, Şekil 6.38'de bu planın son hali verilmiştir. Buna göre en kötü 1. plana ait olumsuz durumlar şöyledir:

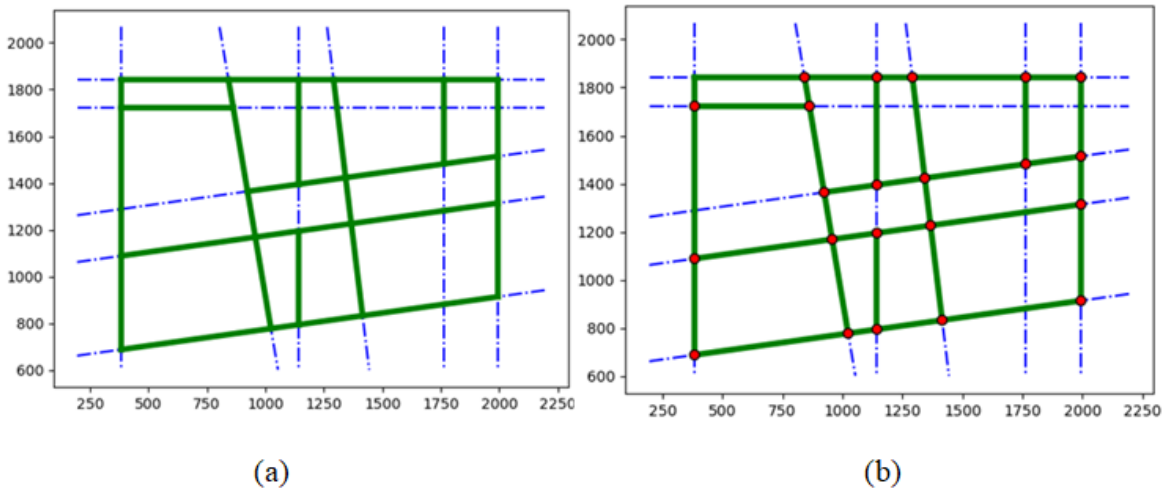
1. Kolon aksa düzgün bir şekilde yerleşmemiştir.
2. Çok sayıda taşan kiriş mevcuttur.
3. Kirişin ucu boş kalmıştır.



Şekil 6.37. Yamuk plana sahip konut projesi için üretilen en kötü 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi.



Şekil 6.38. Yamuk plana sahip konut projesi için üretilen en kötü 1. planın son hali.

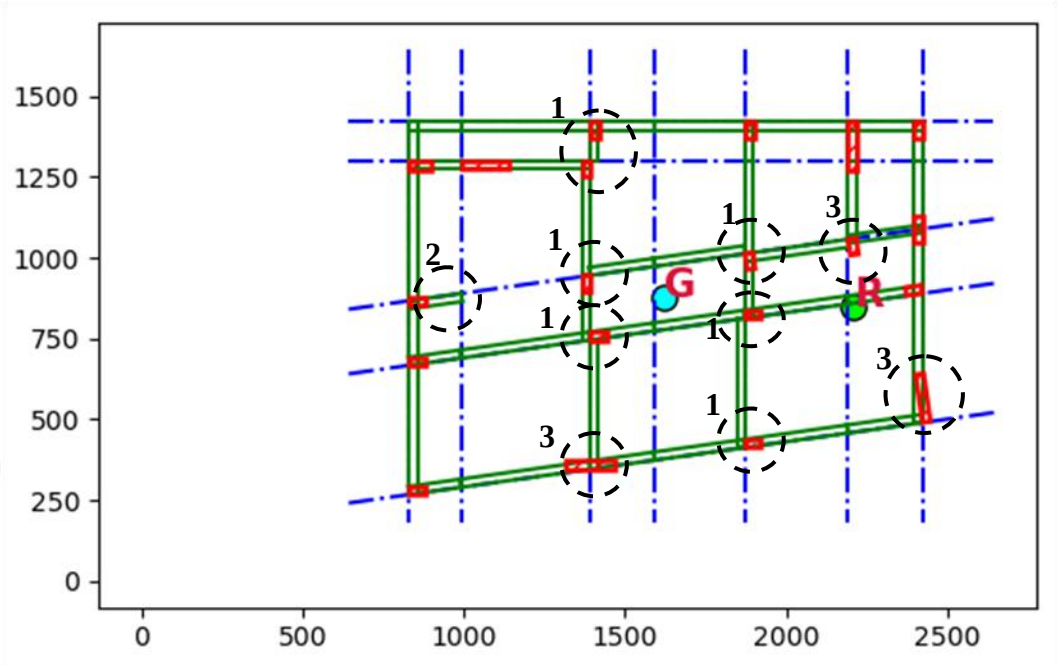


Şekil 6.39. Yamuk plana sahip konut projesi için üretilen en kötü 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.

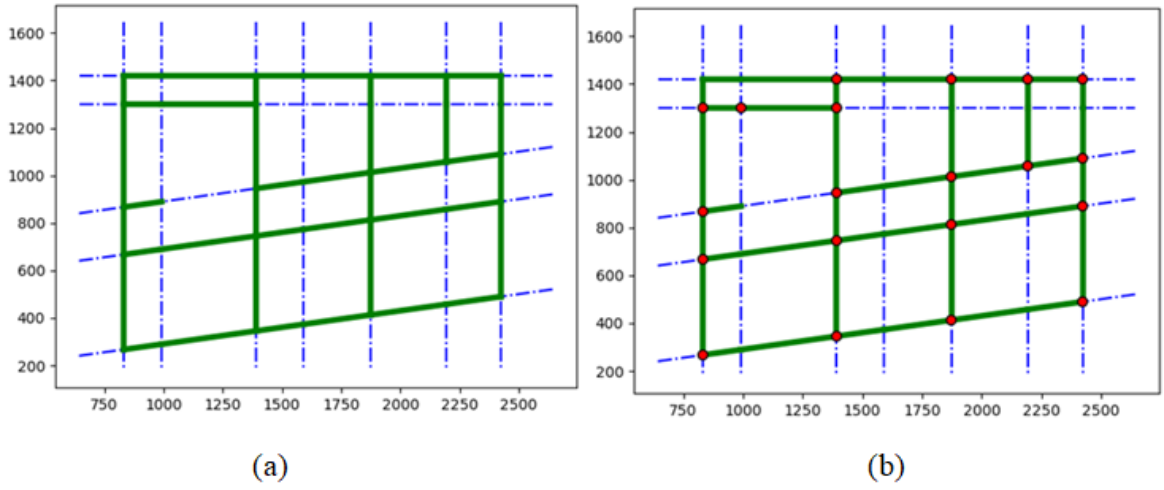
Şekil 6.39'da yamuk plana sahip konut projesi için üretilen en kötü 2. plana ait topoloji planları, Şekil 6.40'ta sistemin son hali verilmiştir. Buna göre 2. plana ait olumsuz durumlar şöyledir:

1. Çok sayıda taşan kiriş mevcuttur.
2. Kirişin ucu boş kalmıştır.
3. Düşey taşıyıcı elemanlar (kolon ve perde) aksa düzgün yerleşmemiştir.

4. Sistemde rijitlik (R)ve kütle (G) merkezleri arasındaki mesafe oldukça fazladır.



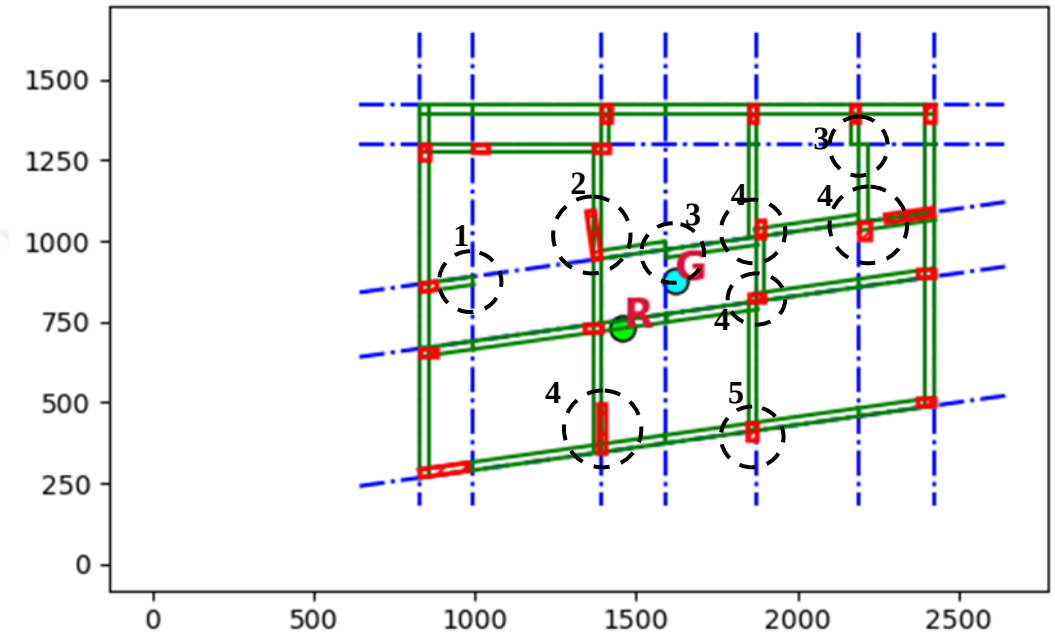
Şekil 6.40. Yamuk plana sahip konut projesi için üretilen en kötü 2. planın son hali.



Şekil 6.41. Yamuk plana sahip konut projesi için üretilen en kötü 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.

Şekil 6.41'de yamuk plana sahip konut projesi için üretilen en kötü 3. plana ait topoloji planları, Şekil 6.42'de sistemin son hali verilmiştir. Buna göre 3. plana ait olumsuz durumlar şöyledir:

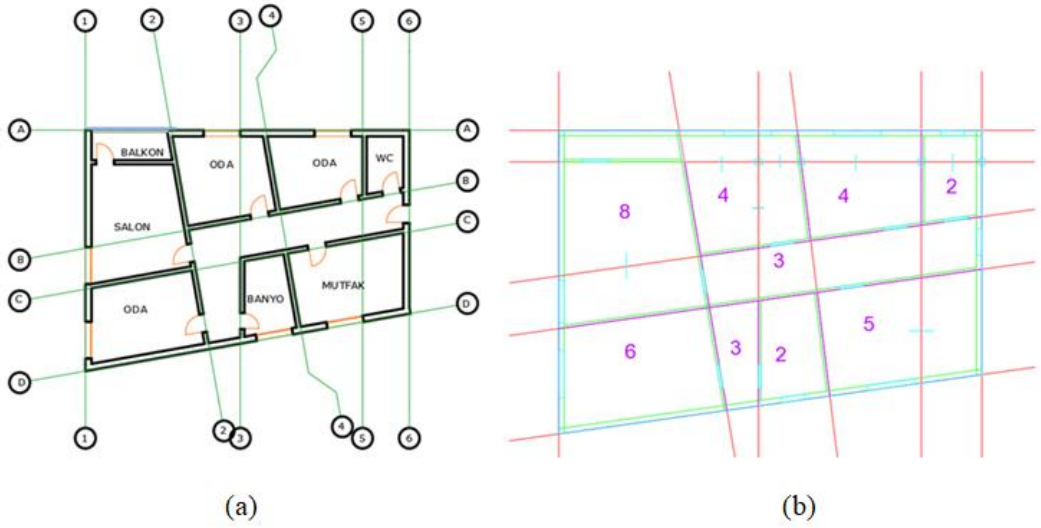
1. Kirişin ucu boş kalmıştır.
2. Düşey taşıyıcı elemanlar (kolon ve perde) aksa düzgün yerleşmemiştir.
3. Kiriş eksenleri çakışık değildir.
4. Çok sayıda taşan kiriş mevcuttur.
5. Kolon yapı sınırları dışına taşmıştır.



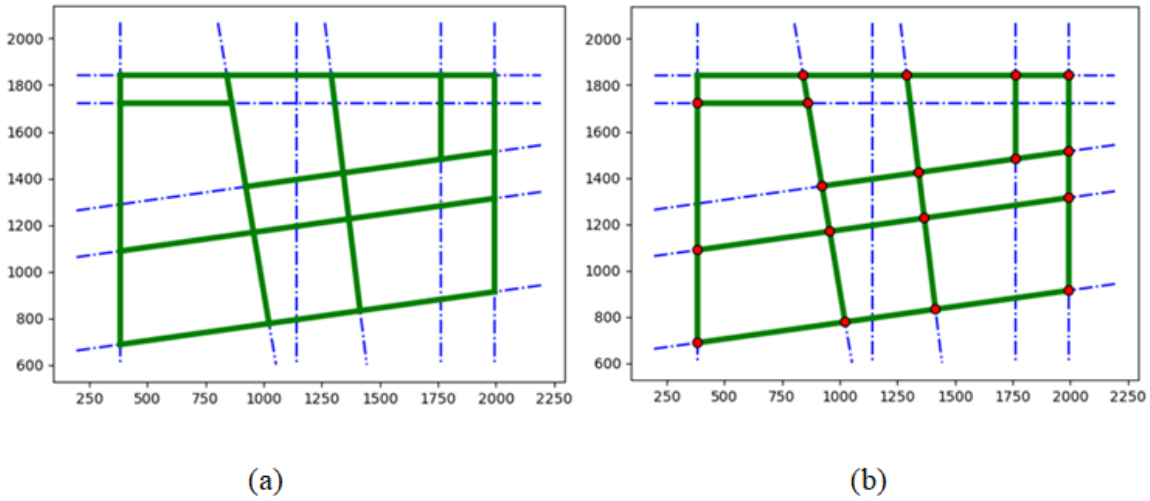
Şekil 6.42. Yamuk plana sahip konut projesi için üretilen en kötü 3. planın son hali.

6.1.4. Çok yamuk mimari projeye sahip bir konut

Analiz edilen son proje ise mimarisi çok yamuk olan bir konut projesidir. Bu mimari proje için de bir önceki yamuk proje örneğinde olduğu gibi uygulanabilir bir sistem elde edilememiştir. Bu proje için analiz sürecinde popülasyon büyüklüğü 42 olarak belirlenmiş ve her aşamada 84000 adayın değerlendirilmesine izin verilmiştir. Her aşamada izin verilen toplam iterasyon sayısı 2000 ve bir tasarımın oluşturulma süresi ortalama 7 saat 25 dakika olarak kaydedilerek toplamda 30 plan üretilmiştir.



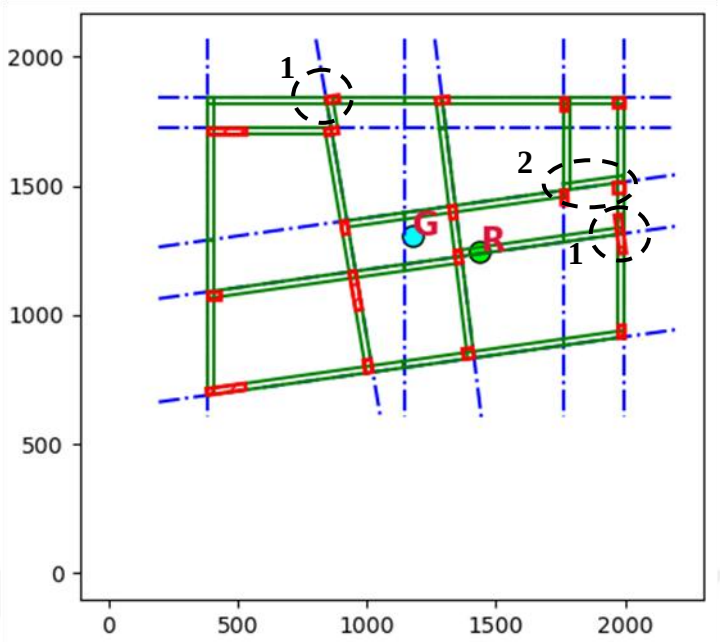
Şekil 6.43. Çok yamuk plana sahip konut için mimari plan ve DXF çizim: a) mimari proje, b) DXF çizim.



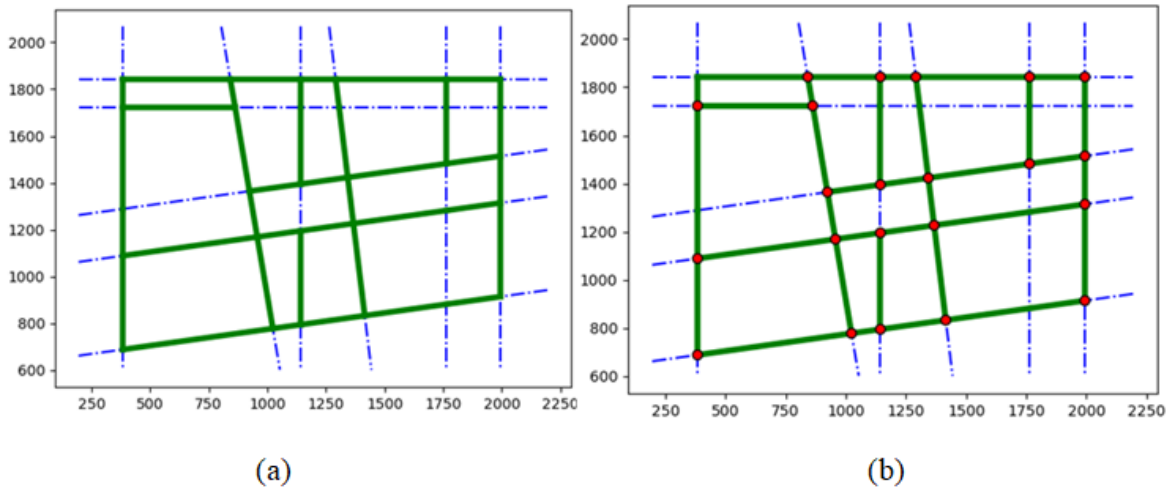
Şekil 6.44. Çok yamuk plana sahip konut projesi için üretilen en iyi 1. plan: a) birinci plana ait kiriş topolojisi, b) 1. plana ait kolon topolojisi.

Şekil 6.43'te çok yamuk projeye ait mimari ve DXF çizim görülmektedir. Bu proje de bir önceki örnekteki verilere sahiptir ancak algoritmayı zorlamak amacıyla çok sayıda aks belli açılar kullanılarak çizilmiştir. Buna göre ilk olarak üretilen en iyi planlara göz atalım. Şekil 6.44'de elde edilen en iyi 1. plana ait topoloji planları verilirken Şekil 6.45'te ise 1. plana ait son durum görülür. Şekil 6.45'te verilen en iyi 1. plana ait olumsuz durumlar şöyledir:

1. Kolon ve perde aksa düzgün yerleşmemiştir.
2. Kiriş kolondan taşmıştır.



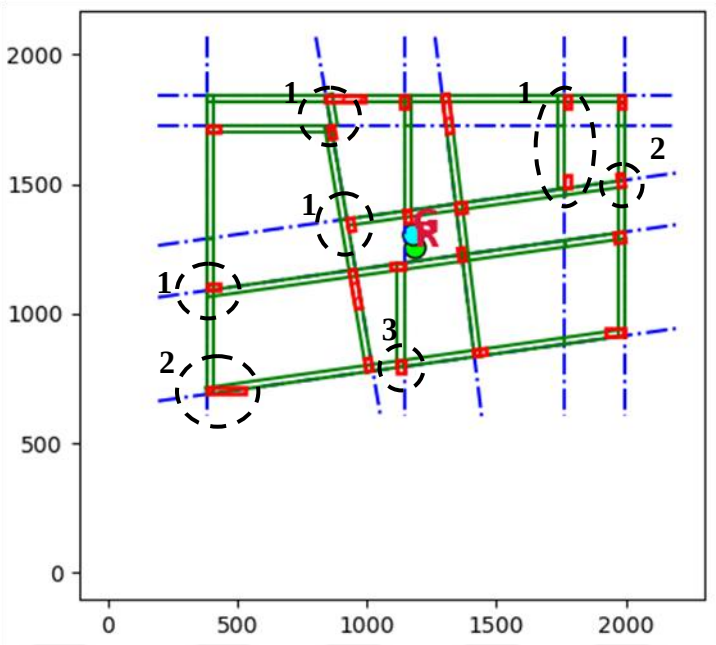
Şekil 6.45. Çok yamuk plana sahip konut projesi için üretilen en iyi 1. planın son hali.



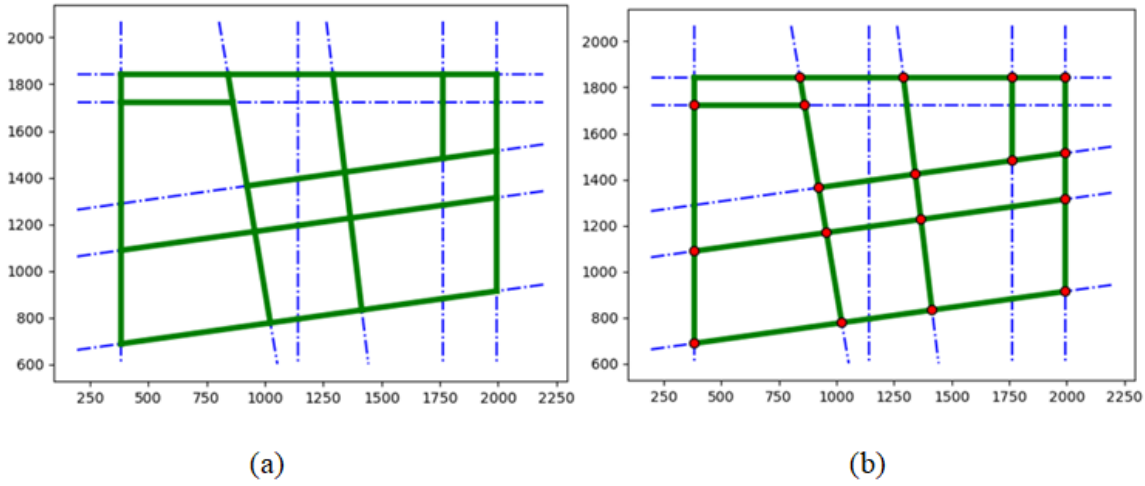
Şekil 6.46. Çok yamuk plana sahip konut projesi için üretilen en iyi 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi

Şekil 6.46'da çok yamuk plana sahip konut projesi için üretilen en iyi 2. plana ait topoloji planları, Şekil 6.47'de son hali verilen 2. plana ait olumsuz durumlar şöyledir:

1. Kiriş kolona tam oturmayarak taşmıştır.
2. Düşey taşıyıcı elemanlar (kolon ve perde) aksa düzgün yerleşmemiştir.
3. Kolon yapı sınırları dışına taşmıştır.



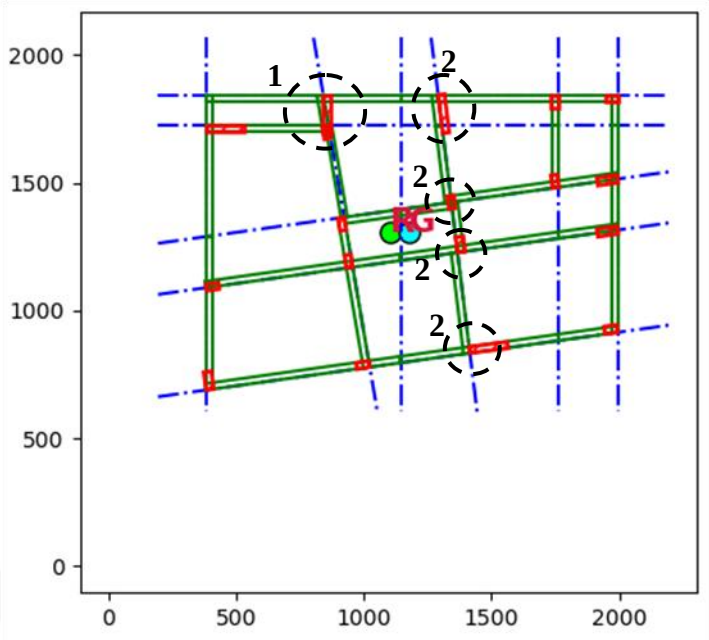
Şekil 6.47. Çok yamuk plana sahip konut projesi için üretilen en iyi 2. planın son hali.



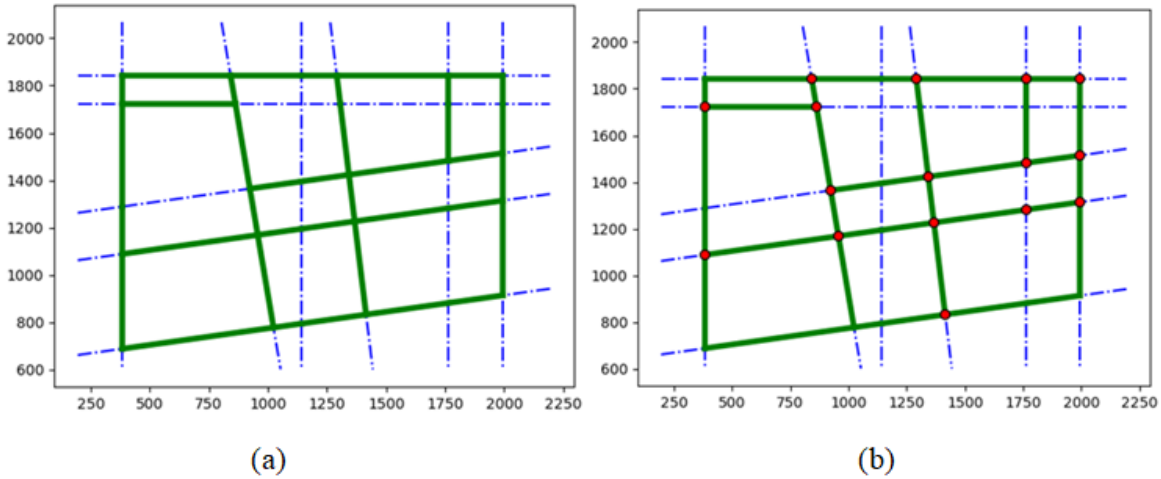
Şekil 6.48. Çok yamuk plana sahip konut projesi için üretilen en iyi 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.

Şekil 6.48'de çok yamuk plana sahip konut projesi için üretilen en iyi 3. plana ait topoloji planları, Şekil 6.49'da sistemin son hali verilmiştir. Buna göre 3. plana ait olumsuz durumlar şöyledir:

1. Perde ve kolon çakışıktır.
2. Kiriş kolona tam oturmayarak taşmıştır.



Şekil 6.49. Çok yamuk plana sahip konut projesi için üretilen en iyi 3. planın son hali.

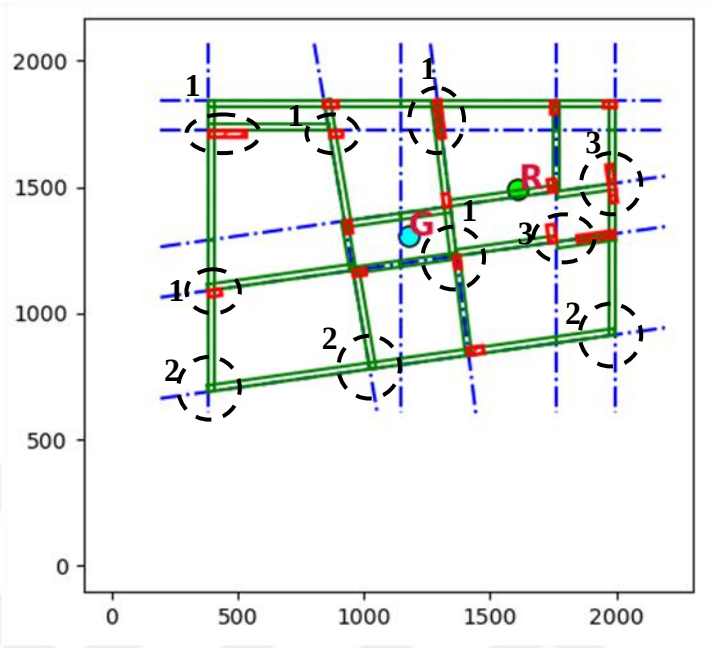


Şekil 6.50. Çok yamuk plana sahip konut projesi için üretilen en kötü 1. plan: a) birinci plana ait kiriş topolojisi, b) birinci plana ait kolon topolojisi.

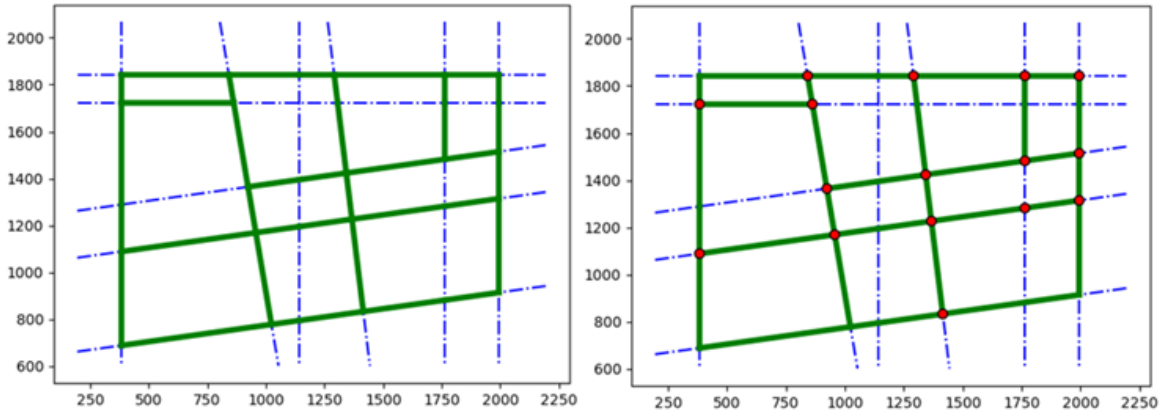
Çok yamuk projeye ait en iyi tasarımlar incelendiktan sonra üretilen en kötü çözümler incelenecektir.Buna göre Şekil 6.50'de çok yamuk plana sahip konut projesi için üretilen en kötü 1. plana ait topoloji planları, Şekil 6.51'de sistemin son hali verilmiştir. Buna göre en kötü 1. plana ait olumsuz durumlar şöyledir:

1. Kirişler kolondan taşmıştır.
2. Köşe noktalarda kolon yoktur.
3. Kolon ve perde aksa düzgün bir şekilde yerleşmemiştir.

4. Sistemde rijitlik (R)ve kütle (G) merkezleri arasındaki mesafe oldukça fazladır.



Şekil 6.51. Çok yamuk plana sahip konut projesi için üretilen en kötü 1. plana ait sistemin son hali.

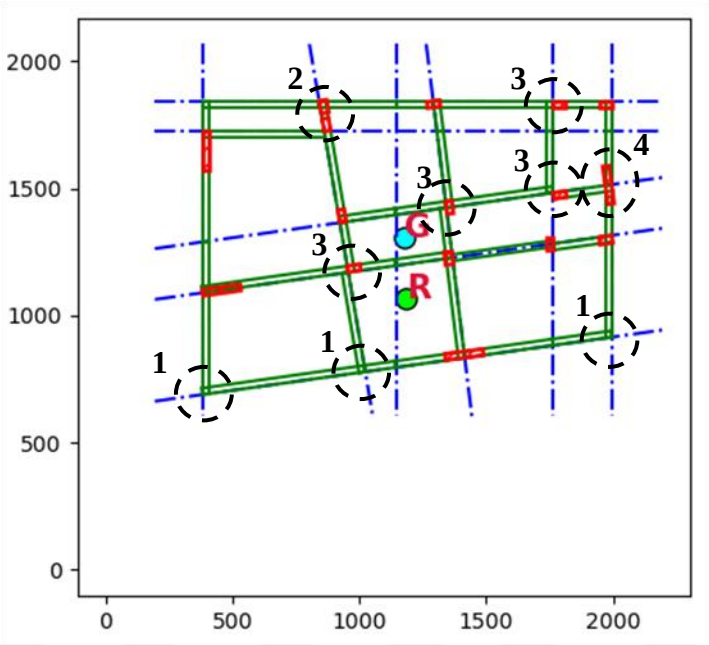


Şekil 6.52. Çok yamuk plana sahip konut projesi için üretilen en kötü 2. plan: a) ikinci plana ait kiriş topolojisi, b) ikinci plana ait kolon topolojisi.

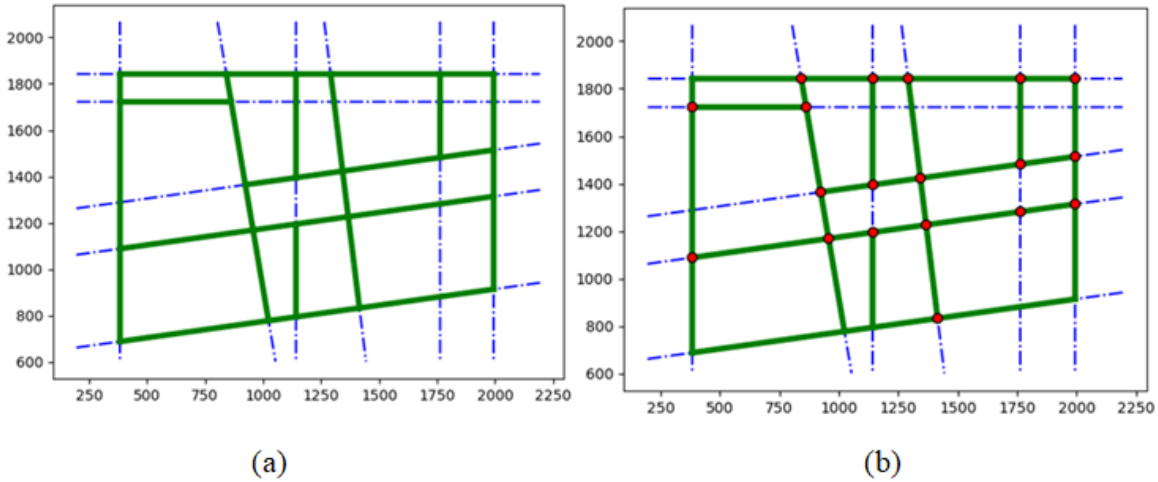
Şekil 6.52'de çok yamuk plana sahip konut projesi için üretilen en kötü 2. plana ait topoloji planları, Şekil 6.53'te sistemin son hali verilen 2. plana ait olumsuz durumlar şöyledir:

1. Köşe noktalarda kolon yoktur.

2. Kolonlar biribirine çok yakın yerleşmiştir.
3. Kirişler kolona tam oturmayarak taşmıştır.
4. Kolon ve perde aksa düzgün bir şekilde yerleşmemiştir.



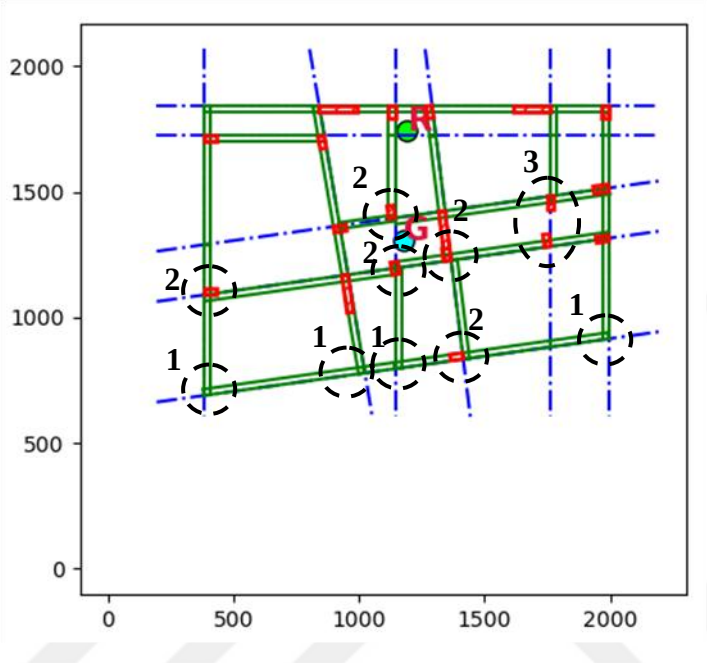
Şekil 6.53. Çok yamuk plana sahip konut projesi için üretilen en kötü 2. planın son hali.



Şekil 6.54. Çok yamuk plana sahip konut projesi için üretilen en kötü 3. plan: a) üçüncü plana ait kiriş topolojisi, b) üçüncü plana ait kolon topolojisi.

Şekil 6.54'de çok yamuk plana sahip konut projesi için üretilen en kötü 3. plana ait topoloji planları, Şekil 6.55'te sistemin son hali verilmiştir. Buna göre 3. plana ait olumsuz durumlar şöyledir:

1. Köşe noktalarda kolon yoktur.
2. Kirişler kolona tam oturmayarak taşmıştır.
3. Kolonlar kullanım alanına taşmıştır.
4. Sistemde rijitlik (R)ve kütle (G) merkezleri arasındaki mesafe oldukça fazladır.



Şekil 6.55. Çok yamuk plana sahip konut projesi için üretilen en kötü 3. planın son hali.

7. SONUÇ VE ÖNERİLER

Bu çalışma kapsamında bilgisayar ile taşıyıcı sistem planları üretebilmek için önerilen tasarım vektörü ve strateji, bilgisayar ile otomatik tasarım bağlamında ümit vericidir. Bölüm 6.1.1 ve 6.1.2'de verilen en iyi tasarımlar (bkz. Şekil 6.6, 6.8, 6.10, 6.19, 6.21 ve 6.23) incelendiğinde kabul edilebilir ve birkaç küçük değişiklik ile yapısal analizlere başlanabilir olduğu düşünülmektedir. Ancak üretilen en kötü tasarımların yapısal analiz aşamasına geçebilmesi için üzerilerinde çok fazla değişiklik yapılması gerekmekte, bu da şimdilik bir yapı için uygun bir taşıyıcı sistem planı elde edebilmek için önerilen strateji ile birden fazla taşıyıcı sistem planı üretilmesi gerektiğini ortaya koymaktadır. Bu çalışmanın önemli sonuçları aşağıda maddeler halinde verilmiştir.

- Öncelikle belirtmek gerekir ki ele alınan problemin çözümü sahip olduğu karmaşık yapısından kaynaklı olarak sonuç üretmede çok sayıda analize ihtiyaç duymaktadır. En basit bir yapı için dahi onlarca tasarım değişkeninin varlığı ve ele alınan çok sayıda kısıt sonrası ortaya çıkan problemi değerlendirme süresi şu anki ev bilgisayarlarının hesaplama gücü dikkate alındığında oldukça zaman alıcı olmaktadır.
- Bölüm 6'da verilen örneklerden görüleceği üzere üretilen bazı taşıyıcı sistem planları (yamuk ve çok yamuk mimari plana sahip konutlar) uygulanabilir nitelikte değildir. Önceki maddede de söylendiği gibi problemin artan tasarım değişkenleri sebebiyle zorlaşması sonucu, sunulan strateji bazı durumlarda uygun tasarımlar önerememektedir.
- Özellikle düzgün planlara sahip konut yapıları (bkz. Bölüm 6.1.1 ve Bölüm 6.1.2) için çok sayıda uygulanabilir çözümün elde edilmesi, önerilen stratejinin bilgisayar ile otomatik tasarım fikri için iyi bir başlangıç noktası olduğunu göstermektedir. Verilen mimari planlar sonucunda herhangi bir insan müdahalesi olmadan uygulanabilir taşıyıcı sistem tasarımlarının elde edilmesi bu düşüncenin en büyük destekleyici unsurudur.
- Sunulan stratejinin eksik yönlerinin olduğu açıktır. Bu yüzden başarısız olunan noktalar, bu konu üzerinde yapılacak çok fazla şeyin olduğunu göstermektedir.

KAYNAKLAR DİZİNİ

Adamu, A., Karihaloo, B. L., & Rozvany, G. I. N. (1994). Minimum cost design of reinforced concrete beams using continuum-type optimality criteria. *Structural Optimization*, *7*(1–2), 91–102. https://doi.org/10.1007/BF01742512

Aga, A. A. A., & Adam, F. M. (2015). Design Optimization of Reinforced Concrete Frames. *Open Journal of Civil Engineering*, *05*(01), 74–83. https://doi.org/10.4236/ojce.2015.51008

Askarzadeh, A. (2014). Bird mating optimizer: An optimization algorithm inspired by bird mating strategies. *Communications in Nonlinear Science and Numerical Simulation*, *19*(4), 1213–1228. https://doi.org/10.1016/j.cnsns.2013.08.027

Askarzadeh, A., & Rashedi, E. (2017). Harmony Search Algorithm. In *Recent Developments in Intelligent Nature-Inspired Computing* (pp. 1–36). https://doi.org/10.4018/978-1-5225-2322-2.ch001

Astolfi, A. (2006). *Optimization An introduction*. (September).

Babiker, S. A., Adam, F. M., & Mohamed, A. E. (2012). Design Optimization of Reinforced Concrete Beams Using Concrete Beams Using Artificial Neural Network. *Green Bookee*, *1*(8), 7–13. https://doi.org/10.1016/S0169-1317(03)00099-1

Balling, R. J., & Yao, X. (1997). Optimization of reinforced concrete frames. *Journal of Structural Engineering*, *123*(2), 193–202. https://doi.org/10.1061/(ASCE)0733-9445(1997)123:2(193)

Bekdas, G., & Niğdeli, S. M. (2015). Optimum Design of Reinforced Concrete Beams using Teaching- Learning-Based Optimization. *Mathematics and Materials*.

Bekdaş, G., & Niğdeli, S. M. (2016). Optimum design of reinforced concrete columns employing teaching-learning based optimization. *Challenge Journal of Structural Mechanics*, *2*(4), 216–219. https://doi.org/10.20528/cjsmec.2016.11.030

Bendsøe, M. P., Díaz, A., & Kikuchi, N. (1993). Topology and Generalized Layout Optimization of Elastic Structures. In *Topology Design of Structures* (pp. 159–205). https://doi.org/10.1007/978-94-011-1804-0_13

Bertolini, L., Elsener, B., Pedeferri, P., & Polder, R. P. (2004). *Concrete Technology for Corrosion Prevention*.

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution Strategies. *Natural Computing*, 3–52. https://doi.org/10.1023/A:1015059928466

KAYNAKLAR DİZİNİ (devam)

Camp, C. V., Pezeshk, S., & Hansson, H. (2003). Flexural Design of Reinforced Concrete Frames Using a Genetic Algorithm. *Journal of Structural Engineering*, *129*(1), 105–115. https://doi.org/10.1061/(ASCE)0733-9445(2003)129:1(105)

Choi, C., & Kwak, H. (1990). Optimum RC Member Design With Predetermined Discrete Sections. *Journal of Structural Engineering*, *116*(10), 2634–2655. https://doi.org/10.1061/(ASCE)0733-9445(1990)116:10(2634)

Christensen, P. W., & Klarbring, A. (2008). *An Introduction to Structural Optimization*. https://doi.org/10.1007/978-1-4020-8666-3

Coello Coello, C. A., Christiansen, A. D., & Hernandez, F. S. (1997). A simple genetic algorithm for the design of reinforced concrete beams. *Engineering with Computers*, *13*(4), 185–196. https://doi.org/10.1007/BF01200046

Cohn, M. Z., & MacRae, A. J. (1984). Optimization of Structural Concrete Beams. *Journal of Structural Engineering*, *110*(7), 1573–1588. https://doi.org/10.1061/(ASCE)0733-9445(1984)110:7(1573)

Dhiman, G., & Kaur, A. (2018). Optimizing the Design of Airfoil and Optical Buffer Problems Using Spotted Hyena Optimizer. *Designs*, *2*(3), 28. https://doi.org/10.3390/designs2030028

Doğangün, A. (2008). *Betonarme Yapıların Hesap Ve Tasarımı*.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

Douglas, T. R. (1960). *Computer-Aided Design: A Statement Of Objectives*.

Emary, E., & Zawbaa, H. M. (2016). Impact of Chaos Functions on Modern Swarm Optimizers. *PLOS ONE*, *11*(7), e0158738. https://doi.org/10.1371/journal.pone.0158738

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software*, *37*(2), 106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005

Esfandiari, M. J., Urgessa, G. S., Sheikholarefin, S., & Manshadi, S. H. D. (2018). Optimum design of 3D reinforced concrete frames using DMPSO algorithm. *Advances in Engineering Software*, *115*(September), 149–160. https://doi.org/10.1016/j.advengsoft.2017.09.007

KAYNAKLAR DİZİNİ (devam)

Esfandiary, M. J., Sheikholarefin, S., & Bondarabadi, H. A. R. (2016). A Combination Of Particle Swarm Optimization And Multi-Criterion Decision-Making For Opiımum Design Of Reinforced Concrete Frames. *INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING Int . J . Optim . Civil Eng*, *6*(2), 245–268.

Fadaee, M. J., & Grierson, D. E. (1996). Design optimization of 3D reinforced concrete structures. *Structural Optimization*, *12*(2–3), 127–134. https://doi.org/10.1007/BF01196945

Fenves, S. J. (1981). Computer-aided design in civil engineering. *Proceedings of the IEEE*, *69*(10), 1240–1248. https://doi.org/10.1109/PROC.1981.12164

Galeb, A. (2009). *Optimum Design of Reinforced Concrete Rectangular Beams Using Simulated Annealing*.

Gandomi, A. H. (2014). Interior search algorithm (ISA): A novel approach for global optimization. *ISA Transactions*, *53*(4), 1168–1183. https://doi.org/10.1016/j.isatra.2014.03.018

Gavrillas, M. (2010). Applications of Heuristics and Metaheuristics in Power Systems. *Journal of Electrical and Computer Engineering*, *2010*, 1–2.

Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, *48*(March), 1–24. https://doi.org/10.1016/j.swevo.2019.03.004

Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied Soft Computing*, *19*, 177–187. https://doi.org/10.1016/j.asoc.2014.02.006

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1

Govindaraj, V., & Ramasamy, J. V. (2007). Optimum detailed design of reinforced concrete frames using genetic algorithms. *Engineering Optimization*, *39*(4), 471–494. https://doi.org/10.1080/03052150601180767

Guerra, A., & Kiousis, P. D. (2006). Design Optimization of Reinforced Concrete Beams. *Computers and Concrete*, *3*(5), 313–334.

Hadi, M. N. S. (2003). Neural networks applications in concrete structures. *Computers and Structures*, *81*(6), 373–381. https://doi.org/10.1016/S0045-7949(02)00451-0

Haftka, R. T., & Gürdal, Z. (1992). *Elements of Structural Optimization*. https://doi.org/10.1007/978-94-011-2550-5

KAYNAKLAR DİZİNİ (devam)

Haug, E. J., & Kirmser, P. G. (1967). Minimum Weight Design of Beams With Inequality Constraints on Stress and Deflection. *Journal of Applied Mechanics*, *34*(4), 999–1004. https://doi.org/10.1115/1.3607869

Hoit, M., Soeiro, A., & Fagundo, F. (1991). Probabilistic Design And Optimization Of Reinforced Concrete Frames. *Engineering Optimization*, *17*(3), 229–235. https://doi.org/10.1080/03052159108941072

Holland, J. H. (1984). Genetic Algorithms and Adaptation. In *Adaptive Control of Ill-Defined Systems* (Vol. 16, pp. 317–333). https://doi.org/10.1007/978-1-4684-8941-5_21

Huanchun, S., & Zheng, C. (1985). Two-Level Optimum Design Of Reinforced Concrete Frames With Integer Variables. *Engineering Optimization*, *9*(3), 219–232. https://doi.org/10.1080/03052158508902515

Huang, M., & Arora, J. (1995). Engineering optimization with discrete variables. *36th Structures, Structural Dynamics and Materials Conference*. https://doi.org/10.2514/6.1995-1333

Jain, M., Singh, V., & Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*, *44*(November 2017), 148–175. https://doi.org/10.1016/j.swevo.2018.02.013

Kanagasundaram, S., & Karihaloo, B. L. (1990). Minimum cost design of reinforced concrete structures. *Structural Optimization*, *2*(3), 173–184. https://doi.org/10.1007/BF01836566

Kaveh, A. (2016). Advances in metaheuristic algorithms for optimal design of structures, second edition. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures, Second Edition*. https://doi.org/10.1007/978-3-319-46173-1

Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, *59*, 53–70. https://doi.org/10.1016/j.advengsoft.2013.03.004

Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray Optimization. *Computers & Structures*, *112–113*, 283–294. https://doi.org/10.1016/j.compstruc.2012.09.003

Kaveh, A., & Sabzi, O. (2012). Optimal design of reinforced concrete frames Using big bang-big crunch algorithm. *International Journal of Civil Engineering*, *10*(3), 189–200.

Kaveh, Ali, & Bakhshpoori, T. (2019). Water Evaporation Optimization Algorithm. In *Metaheuristics: Outlines, MATLAB Codes and Examples* (pp. 137–152). https://doi.org/10.1007/978-3-030-04067-3_12

KAYNAKLAR DİZİNİ (devam)

Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kiran, M. S. (2015). TSA: Tree-seed algorithm for continuous optimization. *Expert Systems with Applications*, *42*(19), 6686–6698. https://doi.org/10.1016/j.eswa.2015.04.055

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Koza, J. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, *4*(2), 87–112. https://doi.org/10.1007/BF00175355

Kripka, M., Boscardin, J. T., & Casteli, S. D. (2016). Topology Optimization of Reinforced Concrete Plane Frames. *Revista Sul-Americana de Engenharia Estrutural*, *13*(2). https://doi.org/10.5335/rsaee.v13i2.6343

Krishnamoorthy, C. S., & Mosi, D. R. (1981). Optimal Design Of Reinforced Concrete Frames Based On Inelastic Analysis. *Engineering Optimization*, *5*(3), 151–167. https://doi.org/10.1080/03052158108902441

Kubin, J. (1994). *İnşaat Mühendisliğinde Bilgisayar Destekli Tasarım ve Projelendirme*.

Lee, C., & Ahn, J. (2003). Flexural Design of Reinforced Concrete Frames by Genetic Algorithm. *Journal of Structural Engineering*, *129*(6), 762–774. https://doi.org/10.1061/(ASCE)0733-9445(2003)129:6(762)

Masching, H., & Bletzinger, K. (2016). Parameter free structural optimization applied to the shape optimization of smart structures. *Finite Elements in Analysis and Design*, *111*, 33–45. https://doi.org/10.1016/j.finel.2015.12.008

Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

KAYNAKLAR DİZİNİ (devam)

Mohammad-Azari, S., Bozorg-Haddad, O., & Chu, X. (2018). Shark Smell Optimization (SSO) Algorithm. In *Studies in Computational Intelligence* (pp. 93–103). https://doi.org/10.1007/978-981-10-5221-7_10

Moharrami, H., & Grierson, D. E. (1993). Computer-Automated Design of Reinforced Concrete Frameworks. *Journal of Structural Engineering*, *119*(7), 2036–2058. https://doi.org/10.1061/(ASCE)0733-9445(1993)119:7(2036)

Nayak, J., Paparao, S., Naik, B., Seetayya, N., Pradeep, P., Behera, H. S., & Pelusi, D. (2019). Chemical Reaction Optimization: A Survey with Application and Challenges. In *Information Sciences* (Vol. 179, pp. 507–524). https://doi.org/10.1007/978-981-13-0514-6_50

Neshat, M., Adeli, A., Sepidnam, G., Sargolzaei, M., & Najaran Toosi, A. (2012). A Review Of Artificial Fish Swarm Optimization Methods And Applications. *International Journal on Smart Sensing and Intelligent Systems*, *5*(1), 108–148. https://doi.org/10.21307/ijssis-2017-474

Nigdeli, S. M., & Bekdas, G. (2016). *Detailed optimum design of reinforced concrete frame structures*. *11*(6).

Oftadeh, R., Mahjoob, M. J., & Shariatpanahi, M. (2010). A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. *Computers & Mathematics with Applications*, *60*(7), 2087–2098. https://doi.org/10.1016/j.camwa.2010.07.049

Özbaşaran, H. (2019). *Konut Yapıları İçin Betonarme Taşıyıcı Sistem Planı Tasarımını Kolaylaştıracak Bir Uzman Sistem Geliştirilmesi*.

Ozturk, H. T., & Durmus, A. (2013). Optimum cost design of RC columns using artificial bee colony algorithm. *Structural Engineering and Mechanics*, *45*(5), 643–654. https://doi.org/10.12989/sem.2013.45.5.643

Parkinson, A. R., Balling, R. J., & Hedengren, J. D. (2013). Optimization Methods for Engineering Design Applications and Theory.

Rajeev, S., & Krishnamoorthy, C. S. (1998). Genetic Algorithm-Based Methodology for Design Optimization of Reinforced Concrete Frames. *Computer-Aided Civil and Infrastructure Engineering*, *13*(1), 63–74. https://doi.org/10.1111/0885-9507.00086

Ramezani, F., & Lotfi, S. (2013). Social-Based Algorithm (SBA). *Applied Soft Computing*, *13*(5), 2837–2856. https://doi.org/10.1016/j.asoc.2012.05.018

KAYNAKLAR DİZİNİ (devam)

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315. https://doi.org/10.1016/j.cad.2010.12.015

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, *179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Ravipudi, J. L., & Neebha, M. (2018). Synthesis of linear antenna arrays using Jaya, self-adaptive Jaya and chaotic Jaya algorithms. *AEU - International Journal of Electronics and Communications*, *92*(3), 54–63. https://doi.org/10.1016/j.aeue.2018.05.022

Renard, P., Alcolea, A., & Ginsbourger, D. (2013). Stochastic versus Deterministic Approaches. In *Environmental Modelling* (pp. 133–149). https://doi.org/10.1002/9781118351475.ch8

Roy, S., Biswas, S., & Sinha Chaudhuri, S. (2014). Nature-Inspired Swarm Intelligence and Its Applications. *International Journal of Modern Education and Computer Science*, *6*(12), 55–65. https://doi.org/10.5815/ijmecs.2014.12.08

Saka, M. P., Hasançebi, O., & Geem, Z. W. (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation*, *28*, 88–97. https://doi.org/10.1016/j.swevo.2016.01.005

Senouci, A. B., & Abdul-Salam, M. A. (1998). Prediction of Reinforced Concrete Beam Depth Using Neural Networks. *Engineering Journal of the University of Qatar*, *11*, 117–132.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering*, *6*(1–2), 132–140. https://doi.org/10.1504/IJCSE.2011.0412021

Shen, J., & Li, Y. (2009). Light Ray Optimization and Its Parameter Analysis. *2009 International Joint Conference on Computational Sciences and Optimization*, *2*, 918–922. https://doi.org/10.1109/CSO.2009.485

Simon, D. (2008). Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

Steven, G., Querin, O., & Xie, M. (2000). Evolutionary structural optimisation (ESO) for combined topology and size optimisation of discrete structures. *Computer Methods in Applied Mechanics and Engineering*, *188*(4), 743–754. https://doi.org/10.1016/S0045-7825(99)00359-X

KAYNAKLAR DİZİNİ (devam)

Stolpe, M. (2016). Truss optimization with discrete design variables: a critical review. *Structural and Multidisciplinary Optimization*, *53*(2), 349–374. https://doi.org/10.1007/s00158-015-1333-x

Storn, R., & Price, K. (1997). Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Su, R., Gui, L., & Fan, Z. (2009). Topology and Sizing Optimization of Truss Structures Using Adaptive Genetic Algorithm with Node Matrix Encoding. *2009 Fifth International Conference on Natural Computation*, *4*, 485–491. https://doi.org/10.1109/ICNC.2009.300

Sun, S. H., Yu, T. T., Nguyen, T. T., Atroshchenko, E., & Bui, T. Q. (2018). Structural shape optimization by IGABEM and particle swarm optimization algorithm. *Engineering Analysis with Boundary Elements*, *88*(May 2017), 26–40. https://doi.org/10.1016/j.enganabound.2017.12.007

Tan, Y., & Zhu, Y. (2010). Fireworks Algorithm for Optimization. In *IEEE Transactions on Evolutionary Computation* (Vol. 13, pp. 355–364). https://doi.org/10.1007/978-3-642-13495-1_44

Tayarani-N., M.-H., & Akbarzadeh-T., M.-R. (2014). Magnetic-inspired optimization algorithms: Operators and structures. *Swarm and Evolutionary Computation*, *19*, 82–101. https://doi.org/10.1016/j.swevo.2014.06.004

Topçu, A. (2019). *Betonarme 2_2- Bir yapının doğuşu*.

Tsiptsis, I. N., Liimatainen, L., Kotnik, T., & Niiranen, J. (2019). Structural optimization employing isogeometric tools in Particle Swarm Optimizer. *Journal of Building Engineering*, *24*(April), 100761. https://doi.org/10.1016/j.jobe.2019.100761

Turkay, M. (2004). *Optimizasyon Modelleri ve Çözüm Metodları*.

Venkata Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19–34. https://doi.org/10.5267/j.ijiec.2015.8.004

Woon, S. Y., Querin, O. M., & Steven, G. P. (2001). *Structural application of a shape optimization method based on a genetic algorithm*. (1975), 57–64. https://doi.org/10.1007/s001580100124

Wu, X., Zhang, S., Xiao, W., & Yin, Y. (2019). The Exploration/Exploitation Tradeoff in Whale Optimization Algorithm. *IEEE Access*, *7*, 125919–125928. https://doi.org/10.1109/ACCESS.2019.2938857

KAYNAKLAR DİZİNİ (devam)

Wu, Y., Li, Q., Hu, Q., & Borgart, A. (2017). Size and Topology Optimization for Trusses with Discrete Design Variables by Improved Firefly Algorithm. *Mathematical Problems in Engineering*, *2017*(June), 1–12. https://doi.org/10.1155/2017/1457297

Zhao, W., Wang, L., & Zhang, Z. (2018). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2018.08.030

Zhao, W., Wang, L., & Zhang, Z. (2019). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*, *163*, 283–304. https://doi.org/10.1016/j.knosys.2018.08.030

Zheng, M., Liu, G., Zhou, C., Liang, Y., & Wang, Y. (2010). Gravitation field algorithm and its application in gene cluster. *Algorithms for Molecular Biology*, *5*(1), 32. https://doi.org/10.1186/1748-7188-5-32