

**DÜZLEM ÇELİK KAFES SİSTEMLERİN
KARINCA KOLONİSİ YÖNTEMİ İLE
OPTİMUM TASARIMI
Abidin Hakan YILMAZ
Yüksek Lisans Tezi
İnşaat Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Zekeriya AYDIN
2019**

T.C.
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DÜZLEM ÇELİK KAFES SİSTEMLERİN
KARINCA KOLONİSİ YÖNTEMİ İLE
OPTİMUM TASARIMI

Abidin Hakan YILMAZ

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: Doç. Dr. Zekeriya AYDIN

TEKİRDAĞ-2019

Her hakkı saklıdır

Doç Dr. Zekeriya AYDIN danışmanlığında, Abidin Hakan YILMAZ tarafından hazırlanan “Düzlem Çelik Kafes Sistemlerin Karınca Kolonisi Yöntemi ile Optimum Tasarımı” isimli bu çalışma aşağıdaki jüri tarafından İnşaat Mühendisliği Anabilim Dalı’nda Yüksek Lisans Tezi olarak oy birliği/oy çokluğu kabul edilmiştir.

Juri Başkanı: Doç.Dr. Serkan BEKİROĞLU

İmza :

Üye: Doç.Dr. Zekeriya AYDIN

İmza :

Üye: Dr.Öğr.Üyesi Aydın ÖZMUTLU

İmza :

Fen Bilimleri Enstitüsü Yönetim Kurulu adına

Doç. Dr. Bahar UYMAZ
Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

DÜZLEM ÇELİK KAFES SİSTEMLERİN KARINCA KOLONİSİ YÖNTEMİ İLE OPTİMUM TASARIMI

Abidin Hakan YILMAZ

Tekirdağ Namık Kemal Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Zekeriya AYDIN

Yapay zekaya dayalı optimizasyon teknikleri son 30 yıldır giderek artan bir hızla mühendislik problemlerinin çözümünde kullanılmaktadır. Bu algoritmaların çoğu doğadaki bir süreci taklit ederek geliştirilmiştir. Bu çalışmada karıncaların doğal yaşamını taklit ederek geliştirilmiş karınca koloni algoritması ele alınmıştır. Literatürde kullanılan farklı karınca koloni algoritmalarına alternatif yeni bir basitleştirilmiş karınca koloni algoritması önerilmiştir. Bu algoritma ile düzlem kafes sistemlerin optimum tasarımı gerçekleştirilmiştir. Çalışmada karınca sayısı, feromon güncelleme katsayısı ve ceza katsayısı gibi optimizasyon parametrelerinin uygun değerleri araştırılmıştır. Gerekli hesapları yapmak üzere “basic” kodlama dilinde bir bilgisayar programı kodlanmış ve bu program ile literatürde yer alan bazı kafes sistemlerin optimum tasarımı gerçekleştirilmiştir. Çalışma sonucunda kullanılan algoritmanın basit bir formülasyon içermesine rağmen ulaşılan sonuçların literatürdekilerden daha iyi ya da onlara yakın olduğu görülmüştür.

Anahtar Kelimeler: Yapay Zeka, Karınca Kolonisi, Optimizasyon, Kafes Sistemler

2019, 98 sayfa

ABSTRACT

MSc. Thesis

OPTIMIZATION OF PLANE STEEL TRUSS STRUCTURES USING ANT COLONY METHOD

Abidin Hakan YILMAZ

Tekirdağ Namık Kemal University
Graduate School of Natural and Applied Science
Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Zekeriya AYDIN

Optimization techniques based on artificial intelligence have been used for the solution of engineering problems for the last 30 years an increasing rapidly. Most of these algorithms have been developed by imitating a process in nature. In this study, ant colony algorithm developed by imitating the natural life of ants is considered. A new simplified ant colony algorithm has been proposed as an alternative to different ant colony algorithms used in the literature. Optimum design of plane truss structures has been realized using this algorithm. The appropriate values of optimum design parameters such as number of ants, pheromone update coefficient and penalty coefficient were investigated in this study. A computer program has been coded in "basic" coding language to make the necessary calculations, and the optimum design of some trusses from the literature has been performed. Although the algorithm used in the study has a simple formulation, it is seen that the results obtained are better than or close to the ones in the literature.

Key Words: Artificial Intelligence, Ant Colony, Optimization, Truss Structures

2019, 98 pages

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
ÇİZELGE DİZİNİ	v
ŞEKİL DİZİNİ	vi
ÖNSÖZ	vii
1. GİRİŞ	1
2. KONU İLE İLGİLİ YAPILMIŞ ÇALIŞMALAR	3
3. DÜZLEM KAFES SİSTEMLERİN OPTİMUM TASARIMI	6
3.1. Amaç Fonksiyonu.....	7
3.2. Tasarım Değişkenleri ve Parametreler	8
3.3. Sınırlayıcılar	9
3.3.1. Narinlik Sınırlayıcı	9
3.3.2. Gerilme Sınırlayıcısı.....	10
3.3.3. Deplasman Sınırlayıcıları	11
3.3.4. Ceza Fonksiyonu	12
4. DÜZLEM KAFES SİTEMLERİN SONLU ELEMANLAR YÖNTEMİ İLE ANALİZİ .	14
5. KARINCA KOLONİ ALGORİTMALARI	17
5.1. Yapay Zeka Teknikleri	17
5.1.1. Yapay sinir ağları.....	19
5.1.2. Genetik algoritmalar	20
5.1.3. Bulanık mantık	20
5.1.4. Uzman sistemler	21
5.1.5 Yapay arı kolonisi.....	21
5.2. Gerçek Karınca Kolonisi Davranışı.....	22
5.3. Karınca Kolonisi Algoritma Çeşitleri.....	24
5.3.1. Karınca Sistemi Algoritması	26
5.3.2. Max-Min Karınca Sistemi Algoritması	27
5.3.3. Karınca Koloni Optimizasyonu Algoritması	27
5.3.3.1. Ark seçme kuralı.....	28
5.3.3.2. Feromon güncelleme kuralı	28
5.3.3.3. Lokal feromon güncellemesi	28
5.3.3.4. Aday listesi kullanımı.....	29
5.3.4. Sıralı Karınca Sistemi	30
5.3.5. Çoklu Karınca Koloni Algoritmaları	30
5.3.6 Global Karınca Kolonisi Algoritması.....	31
6. KARINCA KOLONİ OPTİMİZASYON ALGORİTMASI İLE OPTİMUM TASARIM PROBLEMİNİN ÇÖZÜLMESİ	32
6.1. Karınca Koloni Algoritmasının Düzlem Kafes Sistem Optimizasyonuna Uyarlanması...	32
6.2. Bu Çalışmada Kullanılan Karınca Koloni Algoritması	36
6.3. Akış Diyagramı	38
7. UYGULAMALAR	39
7.1. Programın İrdelenmesi	39
7.1.1. Ceza katsayısının araştırılması	43
7.1.2. Feromon güncelleme katsayısının araştırılması	57
7.1.3. Tasarım değişkenleri kümesinin araştırılması	70
7.1.4. Karınca sayısının araştırılması.....	79

7.2. On Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı	81
7.3. Elliiki Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı	85
7.4. İkiyüz Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı.....	89
8. SONUÇLAR.....	93
9. KAYNAKLAR.....	95
ÖZGEÇMİŞ	98

ÇİZELGE DİZİNİ

Sayfa

Çizelge 7. 1. On çubuklu kafes sistemin çözümünde kullanılan 200 adet boru kesite ait en kesit alanları ve atalet yarıçapları.....	41
Çizelge 7. 2. Ceza katsayısı 1.0 alındığında ulaşılan sonuçlar	44
Çizelge 7. 3. Ceza katsayısının 1.5 alındığında çözüm sonuçları.....	45
Çizelge 7. 4. Ceza katsayısının 2.0 alındığında çözüm sonuçları.....	46
Çizelge 7. 5. Ceza katsayısının 2.5 alındığında çözüm sonuçları.....	47
Çizelge 7. 6. Ceza katsayısının 3.0 alındığında çözüm sonuçları.....	48
Çizelge 7. 7. Ceza katsayısının 3.5 alındığında çözüm sonuçları.....	49
Çizelge 7. 8. Ceza katsayısının 4.0 alındığında çözüm sonuçları.....	50
Çizelge 7. 9. Feromon güncelleme katsayısının 0.010 alındığında çözüm sonuçları.....	58
Çizelge 7. 10. Feromon güncelleme katsayısının 0.015 alındığında çözüm sonuçları.....	59
Çizelge 7. 11. Feromon güncelleme katsayısının 0.020 alındığında çözüm sonuçları.....	60
Çizelge 7. 12. Feromon güncelleme katsayısının 0.025 alındığında çözüm sonuçları.....	61
Çizelge 7. 13. Feromon güncelleme katsayısının 0.030 alındığında çözüm sonuçları.....	62
Çizelge 7. 14. Feromon güncelleme katsayısının 0.035 alındığında çözüm sonuçları.....	63
Çizelge 7. 15. Feromon güncelleme katsayısının 0.040 alındığında çözüm sonuçları.....	64
Çizelge 7. 16. On çubuklu kafes sistemin çözümünde kullanılan 100 tasarım değişkeninin en kesit alanları ve atalet yarıçapları	72
Çizelge 7. 17. Tasarım değişkenleri kümesinde 100 değer alındığında çözüm sonuçları.....	73
Çizelge 7. 18. On çubuklu kafes sistemin çözümünde kullanılan 50 tasarım değişkeninin enkesit alanları ve atalet yarıçapları	74
Çizelge 7. 19. Tasarım değişkenleri kümesinde 50 değer alındığında çözüm sonuçları.....	75
Çizelge 7. 20. On çubuklu kafes sistemin çözümünde kullanılan 25 tasarım değişkeninin en kesit alanları ve atalet yarıçapları	76
Çizelge 7. 21. Tasarım değişkenleri kümesinde 25 değer alındığında çözüm sonuçları.....	77
Çizelge 7. 22. On çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar ..	83
Çizelge 7. 23. Elliiki çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar ..	86
Çizelge 7. 24. İkiyüz çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar ..	91

ŞEKİL DİZİNİ

Sayfa

Şekil 4. 1. Düzlem kafes sistemde global ekseninde (a) çubuk uç kuvvetleri ve (b) çubuk uç yer değiştirmeleri	14
Şekil 4. 2. Düzlem kafes sistemde lokal ekseninde (a) çubuk uç kuvvetleri ve (b) çubuk uç yer değiştirmeleri	15
Şekil 5. 1. Karıncaların yol tercihi	24
Şekil 6. 1. Karınca Koloni Algoritmasının kafes sistem optimizasyonuna uyarlanması	35
Şekil 6. 2. Karınca koloni algoritmasına ait akış diyagramı	38
Şekil 7. 1. On çubuklu düzlem kafes sistem	40
Şekil 7. 2. Karınca sayısı ve ceza katsayısı değişimine göre cezalandırılmış amaç fonksiyonu	52
Şekil 7. 3. Karınca sayısı ve ceza katsayısı değişimine göre iterasyon sayısı.....	53
Şekil 7. 4. Karınca sayısı ve ceza katsayısı değişimine göre çözüm süresi	54
Şekil 7. 5. Ceza katsayısı değişimine göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri	55
Şekil 7. 6. Ceza katsayısı değişimine göre iterasyon sayısı ortalama değeri	56
Şekil 7. 7. Ceza katsayısı değişimine göre çözüm süresi ortalama değeri	57
Şekil 7. 8. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre cezalandırılmış amaç fonksiyonu	66
Şekil 7. 9. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre iterasyon sayısı	67
Şekil 7. 10. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre çözüm süresi.....	68
Şekil 7. 11. Feromon güncelleme katsayısına göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri	69
Şekil 7. 12. Feromon güncelleme katsayısı değişimine göre iterasyon sayısı ortalama değeri	69
Şekil 7. 13. Feromon güncelleme katsayısı değişimine göre çözüm süresi ortalama değeri.....	70
Şekil 7.14. Tasarım değişkenleri kümesi eleman sayısına göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri	78
Şekil 7. 15. Tasarım değişkenleri kümesi eleman sayısına göre iterasyon sayısı ortalama değeri.....	79
Şekil 7. 16. Tasarım değişkenleri kümesi eleman sayısına göre çözüm süresi ortalama değeri	79
Şekil 7. 17. Karınca sayısına göre cezalandırılmış amaç fonksiyonu değeri	80
Şekil 7. 18. Karınca sayısına göre iterasyon sayısı.....	81
Şekil 7. 19. Karınca sayısına göre çözüm süresi.....	81
Şekil 7. 20. On çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması	84
Şekil 7. 21. Elliiki çubuklu düzlem kafes sistem.....	87
Şekil 7. 22. Elliiki çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması	88
Şekil 7. 23. İkiyüz çubuklu düzlem kafes sistem	90
Şekil 7. 24. İkiyüz çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması	92

ÖNSÖZ

Eđitim ve öğrenim hayatımda desteđini hiçbir zaman esirgemeyen aileme, beni sabırla teşvik eden sevgili eşime ve bu çalışmanın hazırlanmasında bilgisini, tecrübesini ve en önemlisi zor zamanlarımda olumlu yaklaşımını hiçbir zaman esirgemeyen hocam Sayın Doç. Dr. Zekeriya AYDIN'a sonsuz şükranlarımı sunarım.

Mayıs 2019

Abidin Hakan YILMAZ
İnşaat Mühendisi

1. GİRİŞ

İnşaat mühendisliği çalışmalarında mimari ve statik kriterler doğrultusunda çok sayıda farklı çözümler elde edilebilir. Bunların içerisinde en uygun çözümü elde etmek inşaat mühendisliğinin başlıca amacıdır. Ancak sonsuz çözüm uzayından en uygun çözümü bulabilmek için çok sayıda tekrarlı hesaplamalar yapmak, bunun içinde çok fazla mesai harcamak gerekir. Ayrıca uygun çözümün doğruluğu ve daha kısa sürede elde edilmesi tasarımcının bilgi, birikim ve tecrübesine göre farklılıklar gösterebilir. Dünya genelinde hızla artan nüfus ve hızla büyüyen endüstri, insanoğlunun çok sayıda çeşitli yapılara ihtiyacını doğurmuştur. İnsanoğlunun yapılaşma ihtiyacının bu şekilde hızla artması, mühendislik çözümü sağlayacak insan ihtiyacını da artırmış ancak bilgili ve tecrübeli tasarımcıya ulaşmak zorlaşmıştır.

Tasarlanacak sistemde kriterlerin artması ve problemin büyüklüğü klasik optimizasyon yöntemlerini çaresiz bırakmaktadır. Bu ve bunlar gibi bir çok neden optimizasyon çalışmalarından daha hızlı ve uygun sonuçlar alabilmek için tasarımcıları farklı arayışlara ve bir çok disiplinde olduğu gibi tamamen doğal yaşamın modellenerek çözülmesine doğru götürmüştür. İnsan beyni, evrim teorisi ve koloni davranışları gibi doğal modeller matematiksel olarak ifade edilip optimizasyon problemlerinin çözümünde başarı ile uygulanmıştır.

Bu yöntemler yapay zeka yöntemleri adı altında toplanmıştır ve bu çalışmada bir yapay zeka optimizasyon yöntemi olan karınca kolonisi yöntemi incelenmiştir. Karınca kolonisi yöntemi bir çok optimizasyon probleminde başarı ile uygulanıp tasarımcıyı optimum çözüme götürmüştür. Ayrıca inşaat mühendisliği problemlerinde de başarı ile uygulandığı bu çalışmada da incelenmiştir.

Karınca kolonisi yöntemi 1991 yılından beri farklı problemlerde başarıyla uygulanmıştır. Bu çalışmada ise literatürdeki benzerlerinden biraz daha basitleştirilmiş bir karınca koloni optimizasyonu yöntemi kullanılmıştır. Bu çalışmanın amacı karınca kolonisi yönteminin mühendislik problemlerinde daha etkin kullanımına yardımcı olmaktır. Çalışma kapsamında literatürde çok sık kullanılan bir kafes sistem örneği üzerinde basitleştirilmiş karınca koloni yönteminde kullanılması gereken karınca sayısı, feromon güncelleme katsayısı ve ceza katsayısı gibi parametrelerin optimum değerleri araştırılmıştır. Amaç fonksiyonu

olarak ise yapının ağırlığı dikkate alınmıştır. Bunun yanında narinlik, gerilme sınırlayıcıları AISC-ASD (American Institute of Steel Construction-Allowable Strength Design) (1989) standardına uygun olarak hesaplanmış, ayrıca deplasman sınırlayıcısı da dikkate alınmıştır. Optimizasyon ve yapısal tasarım için gerekli hesapları yapmak üzere bir bilgisayar programı “basic” dilinde kodlanmıştır. Bu bilgisayar programı ile yine literatürde yaygın olarak yer almış bazı kafes sistem örnekleri yine literatürdeki veriler ile çözümlenerek sonuçları karşılaştırılmıştır.

2. KONU İLE İLGİLİ YAPILMIŞ ÇALIŞMALAR

Karınca Kolonisi yöntemi tüm disiplinler için alternatif bir optimizasyon yöntemidir. Çıkış noktası gezgin satıcı problemi (Dorigo 1992) olması itibari ile lojistik, endüstri mühendisliği ve ulaştırma mühendisliği dallarında daha çok gelişme kaydetmiştir. Yapı mühendisliği alanında ise bu yönteme ilgi giderek artmaktadır. Bazı çalışmalarda diğer optimizasyon yöntemlerinden daha uygun sonuçlar verdiği görülmüş olup diğer optimizasyon teknikleri ile kombine olarak kullanıldığı da görülmüştür. Bu yöntem ile yapılmış bazı çalışmalara aşağıda kısaca değinilmiştir.

Bland (2001), yapıların optimum tasarımı için Karınca Kolonisi yöntemini, ilk kullanan araştırmacılardan biridir. Bland çalışmasında 25 çubuklu bir uzay kafes sistemin optimizasyonunu, ayrık tasarım değişkenleri kullanarak yapmıştır. Bu çalışmada yöntemin geliştirilmesi için Karınca Kolonisi Optimizasyonu ile Tabu Arama (ACOTS) yöntemleri birlikte kullanılmıştır. Problemin tasarımı için, ACOTS yöntemi ile diğer sezgisel arama yöntemlerine göre daha optimum sonuç elde edilmiştir.

Camp ve Bichon (2004) ayrık tasarım değişkenleri kullanarak 10 çubuklu düzlem kafes, 25 ve 72 çubuklu uzay kafes sistemlerinin optimizasyonu üzerinde çalışmışlardır. Bu çalışmanın ardından yine Camp ve ark. (2005) ayrık tasarım değişkenleri kullanarak düzlem kafes sistemleri şekilsel optimizasyonu üzerinde çalışmışlardır.

Serra ve Venini (2006) 6 çubuklu ve 10 çubuklu düzlem kafes sistemin optimizasyonu üzerinde çalışmışlardır. Buldukları sonuçlarla, Karınca Kolonisi yönteminin tasarım problemlerinin çözümünde başarılı bir rastgelelik yöntemi olduğunu ve diğer yöntemlerden daha geliştirilebilir olduğunu belirtmişlerdir.

Kaveh ve Shahrouzi (2007) Karınca Kolonisi ve Genetik Algoritma yöntemlerini düzlem kafes sistemlerin şekil ve boyut optimizasyonu üzerinde yaptıkları çalışmalarda karşılaştırmışlardır.

Kaveh ve ark. (2008) Karınca Koloni yöntemi ile düzlem ve uzay kafes sistemlerin şekil optimizasyonu üzerinde çalışmışlardır. Amaçları gerilme kuvvetine maruz kalan en mukavim yapı şeklini tasarlamak olmuştur. Çalışma ayrık tasarım değişkenleri kullanılarak

yapılmıştır. Optimizasyon problemlerinin çözümü sonucu tasarım değişkenlerinin çok sayıda olduğu problemler Karınca Kolonisi yöntemi ile optimize edilebilir olduğu görülmüştür.

Toğan ve Daloğlu (2008) çalışmalarında Genetik Algoritma ile düzlem kafes ve uzay kafes sistemlerin optimizasyonunu incelemiştirlerdir. Bu çalışmada gruplama stratejisi ve başlangıç popülasyonunu ayarlayan iki yeni yöntemi tartışmışlardır. Sonuçta önerilen algoritma, en az sayıda olası grupla optimum ağırlığı bulma yeteneğine sahip olup, tasarım daha pratik hale gelmiştir.

Li ve ark. (2009) Höristik Parçacık Sürüsü Optimizasyonu, Parçacık Sürüsü Optimizasyonu ve Harmoni Arama yöntemlerini karşılaştırmışlardır. Bu üç algoritma arasında Höristik Parçacık Sürüsü yönteminin yakınsama hızının artırılabilirdiği ve daha hızlı yakınsayan yöntem olduğu görülmüştür.

Aydoğdu (2010) Karınca Koloni Optimizasyonu ve Harmoni Arama yöntemleri ile burulma nedeniyle oluşan çarpılma etkisi altında bulunan uzay kafes sistemlerin optimizasyonu üzerinde çalışıp, bu iki yöntemin karşılaştırmasını altı farklı optimizasyon problemi üzerinde yapmıştır. Öncelikle burulma etkisini göz ardı ederek, daha sonra burulma etkisi de ele alınıp çözüm yapılmış olup, burulma etkisi ile çözüm sonucu yapının yüksekliği ve asimetrikliği ile doğru orantılı olmak üzere, yapı ağırlığının %9 ila %29 oranında arttığı görülmüştür. Öte yandan Karınca Kolonisi ve Harmoni Arama yöntemleri ile çözülen altı problemin sonucunda yapılan karşılaştırmada, bazı problemlerde Karınca Kolonisi yönteminin bazılarında ise Harmoni Arama yönteminin daha optimum sonuç verdiği saptanmış olup, eldeki verilerle hangi yöntemin daha üstün olduğu hakkında karar verilememiştir.

Dede ve ark. (2010) kafes sistemin ağırlığının optimizasyonunu Genetik Algoritma ile yapmışlardır. Çalışmada gerilme ve deplasman sınırlayıcıları kullanılmıştır. Ayrıca çalışma ayırık tasarım değişkenleri kullanılarak yapılmıştır. Çalışmanın sonuçları literatürde sıkça kullanılan birçok problemle karşılaştırılıp etkin bir çözüm sistemi olduğu görülmüştür. Genetik Algoritma çalışmalarında parametrelerde küçük bir değişiklik gerektiğinde bir kromozom değiştirmek için çok sayıda gen gerekmektedir. Bu durum Genetik Algoritmanın verimliliğinin azalmasına neden olmaktadır. Bu çalışmada kullanılan “Hamming-cliff” yöntemi ile bunun üstesinden gelinmiştir.

Angelo ve ark. (2014) ilave kısıtlamalar ile çok amaçlı yapısal optimizasyon problemlerinin üstesinden gelmek için Çift Karınca Koloni Optimizasyonu algoritması önermişlerdir. Her hangi bir aday çözümde ortaya çıkan tasarım değişkenlerinin farklı değerlerinin sayısını sınırlamak için bir asallık kısıtlaması getirilmiştir. Hesaplamalarda elde edilen tasarımlar literatürdeki çalışmalardan elde edilen çözümlerle de karşılaştırılmıştır.

Talebpour ve ark. (2014) Karınca Kolonisi, Harmoni Arama ve Genetik Algoritma yöntemlerini melezleyerek kafes sistemleri optimizasyonu üzerinde çalışmışlardır. Karınca Kolonisi yönteminin çözüm yeteneğini artırmak için Harmoni Arama yöntemi ile birleştirilip melez bir yöntem oluşturulmuştur. Bunun yanında uygulama esnasında Genetik Algoritma yöntemi de programa kombine edilmiştir. Çalışmalarında 52 çubuklu düzlem kafes sistem, 72 çubuklu uzay kafes sistem, 200 çubuklu düzlem kafes sistem, 8 katlı tek açıklıklı ve 5 katlı 2 açıklıklı sistemlerin çözümü yapılmıştır. Bu üç yöntemin birlikte kullanıldığı yeni melez yöntem ile uygun kesitlerin kolayca belirlendiği görülmüştür.

Sönmez (2018) uzay kafes sistemlerin optimizasyonunda metasezgisel algoritmaların karşılaştırmasını yapmıştır. Bu çalışmada Genetik Algoritma, Karınca Kolonisi, Parçacık Sürüsü, Yapay Arı Kolonisi, Yerçekimi Araştırması, Ateşböceği, Gri Kurt Optimizasyonu ve Jaya gibi sekiz farklı yöntemle, sürekli tasarım değişkenleri kullanılarak optimizasyon yapılmıştır. Bu optimizasyon yöntemleri ile 10 çubuklu düzlem kafes sistem, 582 çubuklu kule kafes sistem ve 648 çubuklu çift katmanlı uzay kafes sistemlerin çözümünü yapmıştır. Bulunan sonuçlara göre bahsi geçen optimizasyon yöntemlerini karşılaştırmış ve Yapay Arı Kolonisi, Gri Kurt Optimizasyonu ve Jaya yöntemlerinin diğerlerine göre daha iyi sonuçlar verdiğini tespit etmiştir.

Kalatjari ve Talebpour (2017) çalışmalarında, geliştirilmiş bir Karınca Kolonisi optimizasyon yöntemi kullanmışlardır. Kafes ve çerçeve sistemlerin optimum tasarımını yapmışlardır. Bu çalışmada tasarım değişkenleri değer kümelerinin çok büyük olması durumunda oluşabilecek, yerel optimuma yakınsama ve optimizasyon süresinin uzaması sorunlarından kurtulabilmek için değerlerin tasarım değişkeni bazında sınıflandırılmasına gitmişlerdir. Bu nedenle her tasarım değişkenini uygun bir aralıkta aramışlardır.

3. DÜZLEM KAFES SİSTEMLERİN OPTİMUM TASARIMI

Yapıların optimizasyonu, esasında insanoğlunun yapılaşmaya başladığından beri gelişerek günümüze kadar gelmiştir. Ancak klasik bir optimizasyon tamamıyla tasarımcının bilgisine, zekasına ve tecrübesine bağlıdır. Bilgisayar algoritmalarının gelişmesi ile yapısal sistem optimizasyonu da çeşitli yöntemler ile sistematik hale getirilmiş ve insana bağlı değişkenler minimize edilmeye çalışılmıştır. Bilgisayarların gelişiminin paralelinde optimizasyon teknikleri de gelişmekte ve bu teknikler mühendislik disiplinlerinin hemen hepsinde başarıyla uygulanmaktadır.

Mühendislikte optimum tasarım, fiziksel modellerin matematiksel problemler olarak ifadesinde en iyi çözümü belirleme işlemi olarak tanımlanmaktadır. Optimizasyon alanı içerisinde, optimizasyon için kullanılan ve kullanılacak tekniklerin çeşitli problemler için çözüm algoritmalarının oluşturulması, bu tekniklerin irdelenmesi ve bilgisayar ortamında derlenmesi yer almaktadır. Yapı mühendisliği optimizasyonunda karşılaşılan yaygın problemler; yapıların statik ve dinamik davranışlarına göre optimum boyutlandırılması, yapısal sistemlerin optimum şekil ve geometrik tasarımı ve güvenilirlik bazında planlanan sistemlerin optimum kontrolü olarak sıralanabilmektedir (Aydın 2000).

Herhangi bir optimizasyon problemi genel olarak üç nesnenin tanımlanması ile belirmektedir. Bunlar amaç fonksiyonu, tasarım değişkenleri ve sınırlayıcılarıdır. Amaç fonksiyonu problemin tipine ve ihtiyaçlara göre belirlenmekte, tasarım değişkenleri, optimum çözüme ulaşmak için optimizasyon işlemi boyunca değiştirilen sistem parametreleri olarak alınmakta ve sınırlayıcılar ise tasarım ile optimizasyon probleminde istenmeyen çözümlerin oluşmasını engellemek amacıyla kullanılmaktadırlar. Optimizasyon probleminde en iyi sonuca en kısa zamanda ulaşabilmek için, amaç fonksiyonu ile sınırlayıcıların belirlenmesi ve tasarım değişkenlerinin seçimi son derece önemli olmaktadır (Aydın 2000).

Yapı mühendisliğinin ana amacı sırasıyla emniyet, ekonomi ve estetikdir. Taşıyıcı sistem tasarımında en emniyetli tasarımı üretim maliyetini en aza indirerek oluşturabilmek başlıca amaçtır. Bu amaç özellikle çelik yapıların tasarımında daha ön plana çıkmaktadır. Çelik yapıların minimum ağırlıkta olması bir tasarım kuralı halini almıştır. Ancak en az ağırlık hassasiyeti ile tasarlanan taşıyıcı sistemlerin, üzerine etkiyen yüklere karşı yeterli

dayanımda tasarlanması gerekir. İşte bu şekilde üzerine etkiyen yüklere emniyet sınırları içinde dayanabilen en hafif taşıyıcı sistem tasarımı, o yapı için optimum tasarım olur.

3.1. Amaç Fonksiyonu

Çelik yapıların tasarım problemlerinde dikkate alınan en önemli optimizasyon kriteri yapının ağırlığıdır. Doğal olarak çelik bir yapının ağırlığı maliyeti etkilemektedir. Çelik yapı maliyetine etki eden diğer faktörlere montaj işçiliği, yapının bakımı ve birleşim noktalarının teşkili eklenebilir. Yapının minimum maliyetli olacak şekilde boyutlandırılması, yapının tümünün ve yapıyı oluşturan elemanların her birinin özelliklerine bağlı bir bağıntı olarak, W amaç fonksiyonunu, p_m malzeme özelliklerini, p_c birleşim noktalarının özellikleri, p_s yapısal karakteristikleri ve p_l işçilik giderlerini göstermek üzere,

$$W = f(p_m, p_c, p_s, p_l) \quad (3.1)$$

bağıntısıyla verilebilmektedir. Bu amaç fonksiyonu ile genel anlamda bir yapısal optimum tasarım problemi, g_1, g_2, \dots, g_n sınırlayıcı fonksiyonları göstermek üzere,

$$\min W = f(p_m, p_c, p_s, p_l), \quad g_1, g_2, \dots, g_n \geq 0 \quad (3.2)$$

bağıntısıyla ifade edilebilmektedir (Aydın 2000).

Bu çalışmada çubuk elemanlardan oluşan kafes sistemlerin optimum tasarımı gerçekleştirilmiş ve amaç fonksiyonunda tek terim olarak sadece minimum ağırlık alınmıştır. Sınırlayıcılar olarak ise gerilme, narinlik ve sehim dikkate alınmıştır. Bu durumda amaç fonksiyonu W ,

$$\min W = \rho \sum_{i=1}^{nm} L_i A_i \quad (3.3)$$

şeklinde ifade edilebilmektedir. Bu bağtıda ρ malzemenin birim hacim ağırlığını, L_i i 'inci çubuğun boyunu, A_k k 'inci çubuğun enkesit alanını ve nm sistemdeki toplam eleman sayısını temsil etmektedir (Aydın 2000).

Bu çalışmada tasarım deęişkenlerinin gruplandırılması yöntemi kullanılmıştır. Bu gruplandırmanın da amaç fonksiyonu üzerinde gösterilmesi durumunda, amaç fonksiyonunun formülasyonu, A_k k 'inci gruba ait elemanların enkesit alanlarını, ng problemdeki toplam grup sayısını göstermek üzere,

$$\min W = \rho \sum_{k=1}^{ng} A_k \sum_{i=1}^{nm} L_i \quad (3.4)$$

şeklini almaktadır (Aydın 2000).

3.2. Tasarım Deęişkenleri ve Parametreler

Düzlem kafes sistemler üzerine yapılan bu çalışmada boyutsal bir optimizasyon yapılmakta olup, çubuk enkesit alanları tasarım deęişkeni olarak alınmıştır. Düzlem kafes sistemlerin optimum tasarım probleminde Karınca Koloni Algoritması uygulanmaya başlanmadan önce, o probleme ait tasarım deęişkenleri belirlenir. Kafes sistemlerin çözümünde, çözüm uzayındaki çubuklar genellikle gruplandırılır. Bu nedenle kafes sistemdeki tasarım deęişkeni sayısı grup sayısına eşit olacaktır. Bu çalışmada da ayrık tasarım deęişkenleri kullanılmaktadır. Dolayısı ile tasarım deęişkenlerinin deęerlerinin alabileceęi muhtemel deęerler (tasarım deęişkenleri deęer kümesi) optimizasyona başlamadan önce belirlenmektedir. Tasarım deęişkenleri düzlem kafes sistemde bulunan her bir çelik çubuk için ayrı ayrı seçilebilse de aynı veya birbirine yakın göreve sahip çelik çubuklar beraber kabul edilip gruplandırma da yapılabilir. Tasarım deęişkenleri sayısı ve özellikleri tamamıyla kullanıcının bilgi, tecrübe ve öngörüsüne baęlı olmakla beraber daha hassas bir çalışma için tasarım deęişkeni sayısı artırılabilir. Optimum tasarımı yapılacak olan çelik çubuklar seçildikten ve bunlara ait tasarım deęişkenleri belirlendikten sonra Karınca Koloni Algoritması yöntemi optimum tasarım problemi üzerinde uygulanabilir.

Tasarım parametreleri amaç fonksiyonun hesaplanmasında etkili olan ve tasarım değişkenlerinden farklı olarak optimizasyon süreci boyunca değeri değişmeyen yapısal özelliklerdir. Düzlem kafes sistem enkesitsel optimizasyon problemi için, kafes sistemin boyutları, topolojisi, yükler ve malzeme özellikleri birer tasarım parametresidir.

3.3. Sınırlayıcılar

Çelik yapıların tasarımı yapılırken yapının statik güvenliğini sağlamak için bazı sınırların dışına çıkılmamalıdır. Bu nedenle bu çalışmada da narinlik, gerilme ve deplasman sınırlayıcıları dikkate alınmıştır. Narinlik ve gerilme sınırlayıcılarının hesaplanması AISC-ASD (1989) standardına uygun olarak yapılmıştır.

3.3.1. Narinlik Sınırlayıcı

Bu çalışmada narinlik sınırlayıcısı (g_l) her bir çubuğun narinlik sınırının ihlal edilme derecesine bağlı olarak aşağıdaki bağıntı ile hesaplanmaktadır.

$$g_l = \sum_{i=1}^{nm} g_{l,i} \quad (3.5)$$

Bu bağıntıda, nm kafes sistemde bulunan çubuk sayısını göstermektedir; $g_{l,i}$ i'inci çubuk için narinlik sınırlayıcısının değerini ifade etmekte olup, normalize edilmiş formda aşağıdaki bağıntı ile hesaplanabilir.

$$\left. \begin{aligned} g_{l,i} &= \frac{\lambda_i}{\lambda_{lim}} - 1 & \text{if } \lambda_i &\geq \lambda_{lim} \\ g_{l,i} &= 0 & \text{if } \lambda_i &< \lambda_{lim} \end{aligned} \right\} \quad (3.6)$$

Burada, λ_i i'inci çubuğun narinlik değerini göstermekte olup AISC-ASD (1989) standardına göre,

$$\lambda_i = \frac{K_i L_i}{r_i} \quad (3.7)$$

olarak hesaplanmaktadır. Burada, K_i ve r_i i numaralı çubuk için sırasıyla burkulma katsayısını ve çubuk en kesitine ait atalet yarıçapını ifade etmektedir.

Yukarıdaki bağıntılarda, λ_{lim} narinlik sınırını temsil etmekte olup, AISC-ASD (1989) standardına göre basınç çubuklarında $\lambda_{lim} = 200$ ve çekme çubuklarında $\lambda_{lim} = 300$ alınmaktadır.

3.3.2. Gerilme Sınırlayıcısı

Kafes sistemi oluşturan çubuklarda oluşacak olan gerilmelerin belirli standartlar arasında kalmasını sağlamak için gerilme sınırlayıcısı (g_2) her bir çubuğun gerilme sınırlamasına bağlı olarak aşağıdaki şekilde hesaplanır.

$$g_2 = \sum_{i=1}^{nm} g_{2,i} \quad (3.8)$$

Burada, $g_{2,i}$ i numaralı çubuk için gerilme sınırlayıcısını ifade etmek olup normalize edilmiş formda aşağıdaki şekilde hesaplanır.

$$\left. \begin{aligned} g_{1,i} &= \frac{\sigma_i}{\sigma_{a,i}} - 1 & \text{if } \sigma_i &\geq \sigma_{a,i} \\ g_{1,i} &= 0 & \text{if } \sigma_i &< \sigma_{a,i} \end{aligned} \right\} \quad (3.9)$$

Bu bağıntıda σ_i i numaralı çubukta oluşan gerilmeyi, $\sigma_{a,i}$ i'inci çubuğa ait emniyet gerilmesi değerini göstermektedir. F_y yapısal çeliğin akma dayanımını göstermek üzere, çekmeye maruz kalan çubuklar için emniyet gerilmesi,

$$\sigma_{a,i} = 0.6 \cdot F_y \quad (3.10)$$

bağıntısı ile hesaplanmaktadır. Basınca maruz kalan çubuklar için izin verilen gerilme değeri ise,

$$\left. \begin{aligned} \sigma_{a,i} &= \frac{\left[1 - \frac{\lambda_i^2}{2 \cdot C_c^2}\right] \cdot F_y}{\frac{5}{3} + \frac{3 \cdot \lambda_i}{8 \cdot C_c} - \frac{\lambda_i^3}{8 \cdot C_c^3}} \quad \text{if } \lambda_i < C_c \\ \sigma_{a,i} &= \frac{12 \cdot \pi^2 \cdot E}{23 \cdot \lambda_i^2} \quad \text{if } \lambda_i > C_c \end{aligned} \right\} \quad (3.11)$$

bağıntısı ile hesaplanmaktadır. Burada,

$$C_c = \sqrt{\frac{2 \cdot \pi^2 \cdot E}{F_y}} \quad (3.12)$$

olup, E elastisite modülünü, C_c plastik narinlik sınırını temsil etmektedir.

3.3.3. Deplasman Sınırlayıcıları

Kafes sistem üzerinde bulunan düğüm noktalarının yapacağı deplasmanın güvenlik sınırları içerisinde kalmasını sağlamak için belirlenen sınırlayıcı (g_3), her bir noktanın deplasman sınırlayıcısına bağlı olarak aşağıdaki gibi ifade edilmektedir.

$$g_3 = \sum_{i=1}^{np} g_{3,i} \quad (3.13)$$

Bu bağıntıda, np kafes sistemde bulunan düğüm noktası sayısını göstermektedir. $g_{3,i}$ ise i numaralı nokta için deplasman sınırlayıcısını ifade etmek olup normalize edilmiş formda aşağıdaki şekilde hesaplanır.

$$\left. \begin{aligned} g_{3,i} &= \frac{f_i}{f_{a,i}} - 1 \quad \text{if } f_i \geq f_{a,i} \\ g_{3,i} &= 0 \quad \text{if } f_i < f_{a,i} \end{aligned} \right\} \quad (3.14)$$

Bu bağıntıda f_i ve $f_{a,i}$ i numaralı çubuk için sırasıyla oluşan deplasman ve izin verilebilir deplasman değerlerini göstermektedir.

3.3.4. Ceza Fonksiyonu ve Cezalandırılmış Amaç Fonksiyonu

Bu çalışmada, bu bölüme kadar tanımlanan optimum tasarım problemi bir takım sınırlayıcıları kullanmak suretiyle sonuca ulaşmaktadır. Ancak temelde Karınca Koloni Algoritması sınırlayıcısız optimizasyon problemlerinin tasarımı için geliştirilmiş bir yöntem olduğundan, sınırlayıcılara bağlı olarak belirlenen bir tasarım probleminin sınırlayıcısız bir probleme dönüştürülmesi gerekmektedir.

Bunun için sınırlayıcıların ihlal edilme derecesini hesaplayan, ceza fonksiyonu adında bir fonksiyon belirlenmektedir. Ceza fonksiyonunun amaç fonksiyonuna ilave edilmesi ile amaç fonksiyonu sınırlayıcıları da kapsayan bir forma getirilmiş olmaktadır. Sınırlayıcılara bağlı olarak ceza fonksiyonunun belirlenebilmesi için, sınırlayıcıların normalize edilmiş formları yukarıda gösterilmektedir.

Sınırlayıcılar yukarıda verilen bağıntılarla ifade edilip normalize edildikten sonra problemi sınırlayıcısız bir probleme dönüştürmek için gerekli olan C ceza fonksiyonunun hesaplanmasına geçilmektedir.

Bu çalışmada ceza fonksiyonu (C), narinlik, gerilme ve deplasman sınırlayıcılarına bağlı olarak,

$$C = g_1 + g_2 + g_3 \quad (3.15)$$

bağıntısı ile hesaplanmaktadır. . Bu şekilde hesaplanan ceza fonksiyonunun amaç fonksiyonuna eklenmesi için aşağıdaki bağıntı kullanılacaktır (Rajeev ve Krishnamoorthy 1992).

$$\Phi = W \cdot [1 + K \cdot C] \quad (3.16)$$

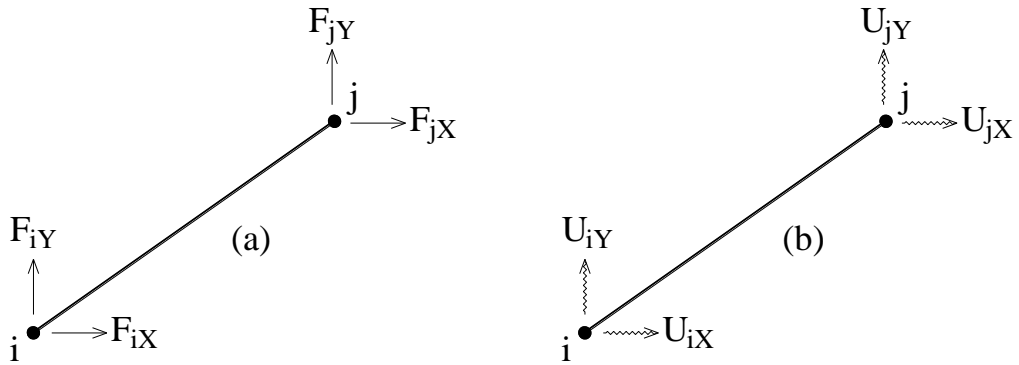
Bu bağıntıda Φ cezalandırılmış amaç fonksiyonunu, K ise ceza katsayısını ifade etmektedir. Ceza katsayısı sınırlayıcıların çözüm üzerindeki etkisini belirlemekte olup, değeri probleme göre belirlenmektedir.

4. DÜZLEM KAFES SİTEMLERİN SONLU ELEMANLAR YÖNTEMİ İLE ANALİZİ

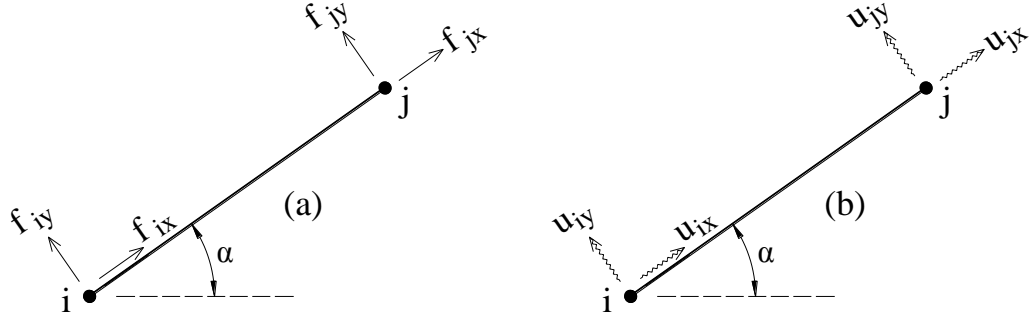
Düzlem kafes sistemler, aynı düzlemde çok sayıda çubuğun mafsallı ile uç uca bağlandığı ve yüklerin sadece kafesi oluşturan mafsallı noktalara etkidiği kabul edilen taşıyıcı sistemlerdir. Dolayısıyla, her bir doğrusal çubuk sadece aksel basınç veya çekme kuvvetinin etkisi altındadır. Bu yüzden sistemi oluşturan çubukların eğilmelerine izin verilmez.

Sonlu elemanlar yöntemi ise mühendislikte karşılaşılan birçok karmaşık problemlerde çözüm elde etmek için kullanılan sayısal bir yöntemdir. Sonlu elemanlar yönteminin temeli büyük ve karmaşık bir problemi sonlu eleman adı verilen alt bölgelere ayırıp basite indirmektir. Asıl problemin basitleştirilerek çözülmeye çalışılması kesin sonucu vermez ancak kesine çok yakın sonuç elde edilebilir.

Kafes sistemlerde düğüm noktaları mafsallı olarak kabul edildiğinden, düzlem kafes sistemlerin her bir düğüm noktasında iki adet serbestlik derecesi vardır. Bu serbestlik derecelerine karşılık gelen çubuk uç kuvvetleri ve çubuk uç yer değiştirmeleri Şekil (4.1) de global eksenler (X, Y) için, Şekil (4.2) de ise lokal eksenler (x, y) için gösterilmektedir.



Şekil 4. 1. Düzlem kafes sistemde global eksenlerde (a) çubuk uç kuvvetleri ve (b) çubuk uç yer değiştirmeleri



Şekil 4. 2. Düzlem kafes sistemde lokal eksende (a) çubuk uç kuvvetleri ve (b) çubuk uç yer değiştirmeleri

Lokal eksende çubuk uç kuvvetleri ile çubuk uç yer değiştirmeleri arasında aşağıda matris formda verilen bağıntı yazılabilir.

$$\{f_e\} = [k_e] \cdot \{u_e\} \quad (4.1)$$

$$\begin{Bmatrix} f_{ix} \\ f_{iy} \\ f_{jx} \\ f_{jy} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} u_{ix} \\ u_{iy} \\ u_{jx} \\ u_{jy} \end{Bmatrix} \quad (4.2)$$

Yukarıdaki bağıntıda, kafes sistem çubuğu için lokal eksende olmak üzere, $\{f_e\}$ çubuk uç kuvvetleri vektörünü, $[k_e]$ eleman rijitlik matrisini ve $\{u_e\}$ çubuk uç yer değiştirmeleri vektörünü ifade etmektedir. Lokal eksendeki çubuk uç yer değiştirmeleri ile global eksendeki çubuk uç yer değiştirmeleri arasında ise, çubuğun i ucunun yatay eksen ile arasındaki α açısına bağlı olarak aşağıdaki bağıntı yazılabilir.

$$\{u_e\} = [T_e] \cdot \{U_e\} \quad (4.3)$$

$$\begin{Bmatrix} u_{ix} \\ u_{iy} \\ u_{jx} \\ u_{jy} \end{Bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{Bmatrix} U_{ix} \\ U_{iy} \\ U_{jx} \\ U_{jy} \end{Bmatrix} \quad (4.4)$$

Yukarıdaki bağıntıda $[T_e]$ global ve lokal eksenler arasındaki dönüşümü sağlayan matrisi, $\{U_e\}$ global eksen çubuk uç yer değiştirmeleri vektörünü ifade etmektedir.

Yukarıdaki (4.1) ve (4.2) bağıntılarında lokal eksen çubuk uç kuvvetleri ile çubuk uç yer değiştirmeleri için verilen ilişki, dönüşüm matrisi ile gerekli işlemlerin yapılması sonucunda benzer şekilde global eksen için de aşağıdaki gibi yazılabilir.

$$\{F_e\} = [K_e] \cdot \{U_e\} \quad (4.5)$$

$$\begin{Bmatrix} F_{ix} \\ F_{iy} \\ F_{jx} \\ F_{jy} \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \cdot \begin{Bmatrix} U_{ix} \\ U_{iy} \\ U_{jx} \\ U_{jy} \end{Bmatrix} \quad (4.6)$$

Yukarıdaki bağıntıda, kafes sistem çubuğunda global eksen olmak üzere, $\{F_e\}$ çubuk uç kuvvetleri vektörünü, $[K_e]$ eleman rijitlik matrisini, c ve s ise çubuğun i ucunun yatayla yaptığı açı α olmak üzere $\cos\alpha$ ve $\sin\alpha$ değerlerini ifade etmektedir. (4.5) bağıntısında her bir çubuk için verilmiş olan ilişki kafes sistemin geneli için aşağıdaki gibi yazılabilir.

$$\{F\} = [K] \cdot \{U\} \quad (4.7)$$

Bu bağıntıda kafes sistem için $\{F\}$ dış yük vektörünü, $\{U\}$ yer değiştirme vektörünü göstermektedir. $[K]$ ise kafes sistem için sistem rijitlik matrisini ifade etmekte olup nm adet çubuğa sahip bir kafes sistem için aşağıdaki şekilde hesaplanabilir.

$$[K] = \sum_{i=1}^{nm} [K_{e,i}] \quad (4.8)$$

Yukarıda verilen (4.7) bağıntısı ile kafes sistem için global eksen çubuk uç kuvvetleri hesaplandıktan sonra, (4.5) bağıntısı ile global eksen çubuk uç kuvvetleri ve bunların da dönüşüm matrisi ile çarpılması sonucunda lokal eksen çubuk uç kuvvetleri hesaplanabilir.

5. KARINCA KOLONİ ALGORİTMALARI

Karınca Koloni Algoritması gerçek karınca kolonilerinin gıda arama tekniklerinin matematiksel modeline dayalı bir algoritmadır. İlk karınca algoritması 1991 yılında Marco Dorigo tarafından Karınca Sistemi adıyla Gezgin Satıcı Problemin'e (GSP) uygulanmış ve sonrasında doktora tezi olarak sunulmuştur (Dorigo, 1992). Dorigo Karınca Sistemi'ni değişik büyüklükte birçok GSP' ye uygulamış ancak 75 şehirden az problemlerde başarılı olurken problemin ölçeği arttıkça başarı oranının azaldığı görülmüştür. Dorigo 1992 yılından 1996 yılına kadar algoritmayı geliştirip "Elitist Strategy" olarak adlandırılan "Elitist Ant System" uygulamasını bilim dünyasına tanıtmıştır. Bu yöntemle göre karıncaların arama sırasında bulunduğu en iyi yola feromon bıraktığı kabul edilir. Daha sonra Dorigo ve Gambardella (1997) Karınca Koloni Sistem'ini ortaya koymuştur. Bu sistemde sadece en iyi karıncaların feromon bırakmasına izin verilmiş ve bir şehirden diğer bir şehre belirli oranlarla geçiş sağlayarak sözde rastgelelik oluşturulmuştur. Stützle ve Hoos (1997) feromon izini maksimum ve minimum aralıklarda sınırlayan Max-Min Ant System (MMAS)'i tanıtmışlardır. Ayrıca Bullnheimer ve ark. (1999) AS'rank (Rank Temelli Karınca Sistemi) adı altında "Elitist Strategy"nin gelişmiş bir versiyonunu sunmuşlardır.

Bu bölümde ilk olarak, literatürde kullanılan yapay zeka teknikleri belirtilip, yaygın olarak kullanılan tekniklerden kısaca bahsedilmiştir. Ardından gerçek bir karınca kolonisinin yemek arama tekniğinden bahsedilip, bir sonraki bölümde ise karıncaların bu davranışının matematiksel olarak modellenmesi incelenmiştir. Ardından bu çalışmada kullanılan Karınca Koloni Algoritması tanıtılıp, son olarak Karınca Kolonisi yöntemi ile yapı mühendisliği alanında yapılan başlıca çalışmalardan bahsedilmiştir.

5.1.Yapay Zeka Teknikleri

Endüstriyel hayatın ön plana çıktığı, insanların zamanla yarıştığı günümüzde, insanların yapabileceği birçok iş, hayatı kolaylaştırmak adına, makinelere yaptırılmaktadır. Bu da makinelerin hızla gelişmesinin hatta insandan daha verimli olmasının ve zeki makinelerin tasarımının yolunu açmıştır.

Zeka, kavramlar ve algılar yardımıyla soyut ya da somut nesnelere arasındaki ilişkiyi kavrayabilme, soyut düşünme, muhakeme etme, planlama, problemleri çözme ve bu zihinsel

işlevleri uyumlu bir şekilde kullanabilme yeteneğidir. Yapay Zeka ise; bilgisayarların öğrenme, anlama ve akıl yürütme gibi insan zekasına özgü yeteneklere sahip olmasıdır. Fikir babası, "Makineler düşünebilir mi ?" sorusunu ortaya atarak Makine Zekası'nı tartışmaya açan, bilgisayar biliminin kurucusu sayılan İngiliz matematikçi ve bilgisayar bilimcisi Alan Mathison Turing'dir. Alan M. Turing İkinci Dünya Savaşı'nda Alman ordusunun haberleşmede kullandığı Enigma şifresini kırmayı başarmıştır. Alan M. Turing (1950) savaş sonrasında kendini yapay zeka alanında çalışmalara adanmış ve "Computing Machinery and Intelligence" (Bilgisayar Mekanizması ve Zeka) adlı makalede yapay zekaya işaret etmiştir. Bu makalede ki Turing Testi'nde, birbirini görmeden iletişim kuran iki kutup vardır. Birinci kutup insandır, ikincisi bilgisayar. Turing'e göre, eğer bilgisayar, bilgi sahibi biri gibi insan ile iletişim kurabilir ise, bu iletişimin öteki ucunda bulunan insan da, işlemini tamamladıktan sonra bağlantıda bulunduğu şeyin insan mı yoksa makine mi olduğunu ayırt edemez ise, bilgisayar "zeki" olarak tanımlanmalıdır.

Yapay Zeka kavramı ise ilk olarak, 1956 yılında Dartmouth College'de düzenlenen bir konferansta doğmuştur. Konferans Marvin Minsky ve John McCarthy tarafından düzenlenmiş ve McCarthy, yeni türettiği Yapay Zeka ismini, konferansa ad olarak önermiştir. Harbert A. Simon, J.C. Shaw ve Alan Newell daha sonra "insan gibi düşünme" yaklaşımına göre üretilmiş ilk program olan Genel Sorun Çözücü'yü geliştirmişlerdir. Minsky ve McCarthy Massachusetts Institute of Technology'de, Simon ve Newell da Carnegie-Mellon Üniversitesinde, yapay zeka laboratuvarlarını kurmuşlardır. McCarthy daha sonra MIT'den Stanford Üniversitesine geçerek orada da yapay zeka laboratuvarı kurmuştur. Bu kişiler yapay zeka kavramının kurucuları olarak görülmektedir.

Yapay zeka tekniklerinin mühendislik alanında kullanımı ile ilgili çalışmalar incelendiğinde; araştırma alanı çok büyük boyutlu ve karmaşık, matematiksel analiz ile çözümü mümkün olmayan, geleneksel araştırma yöntemleri ile başarısız olunmuş ve iyi sonuç alınamamış problemlerin optimum çözümüne ulaşmada yapay zeka tekniklerinden faydalandığı görülmüştür.

Yapay zekanın ortaya çıktığı günden günümüze kadar, bir çok teknik ortaya atılmış ve bunlardan bir çoğu da literatürde yaygın bir şekilde kullanılmıştır. Aşağıda bu tekniklerden bazıları sıralanmıştır:

- Yapay sinir ađları (Artificial Neural Network)
 - Genetik Algoritmalar (Genetic Algorithm)
 - Bulanık Mantık (Fuzzy Logic)
 - Uzman Sistemler (Expert Systems)
 - Yapay Arı Kolonisi (Artificial Bee Colony)
 - Karınca Kolonisi Algoritmaları (Ant Colony Algorithm)
 - Tabu Arama (Tabu Search)
 - Harmoni Arama (Harmony Search)
 - Isıl İşlem Algoritması (Simulated Annealing)
 - Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm)
 - Yarasa Algoritması (Bath Algorithm)
 - Gri Kurt Optimizasyonu (Gray Wolf Optimizer)
- vb. gibi daha bir çok yapay zeka tekniđi sıralanabilir.

Yukarıda sıralanan yapay zeka tekniklerinden de anlaşılacağı üzere, yapay zeka teknikleri, insan beyni ve doğadaki canlıların yaşam mücadelesi verirken içgüdüsel olarak yaptıklarının matematiksel olarak modellenmesi veya bir hōristik geliştirilmesi ve bu hōristiđin bilgisayar dilinde kodlanmasıdır. Literatürde çođunlukla kullanılan yapay zeka tekniklerinin bazıları hakkında aşağıda kısaca bilgi verilmiştir.

5.1.1. Yapay sinir ađları

İnsan beyni yaklaşık 10 milyar sinir hücresinden ve 60 trilyon sinaps bağlantısından oluşmaktadır. Bu kadar karmaşık yapıya rağmen, bir sinir hücresinin tepki hızı, günümüz bilgisayarlarına göre oldukça yavaş kalmaktadır. Ancak beynin bilgisayar sistemlerine göre asıl üstünlüğü idrak etme yeteneğindedir. Ayrıca eksik bilgi ile görme, konuşma, şekil tanıma gibi konularda insan beyni bilgisayarlara karşı ezici üstünlüğe sahiptir (Baykan 2007).

Yapay Sinir Ađları (YSA), insan beyninin çalışma sisteminin yapay olarak modellenmesi sonucu ortaya çıkmıştır. YSA insan beynindeki sinir hücrelerinin, ya da yapay olarak basit işlemcilerin birbirlerine deđişik etki seviyeleri ile bağlanması sonucu oluşan karmaşık bir sistem olarak düşünülebilir.

Yapay sinir ağı uygulamaları geri yayılım yöntemine dayanmaktadır. Bir geri yayımlı sinir ağı, yapay nöronlar adı verilen, birbiriyle bağlantılı çok sayıda işlem elemanından oluşmaktadır. Elemanlar iki ya da daha fazla tabakaya mantıklı bir şekilde dizilmekte ve birbirlerini belirlenen bir ağırlıktaki işbirliğiyle etkilemektedirler. Bu ölçütsel ağırlıklar birbirleriyle bağlantılı olan elemanların arasındaki etkinin çeşidini ve kuvvetini belirlemektedir (Aydın 2000).

5.1.2. Genetik algoritmalar

Genetik algoritma, genetik evrim mekanizmasını model alan bir stratejidir. Genetik algoritmanın özü en iyinin hayatta kalması ve adaptasyonu esasına dayanmaktadır. Genetik algoritma, her ne kadar belirli kriterler sayesinde en iyiyi bulmaya yönlendirilmiş bir arama yöntemi ise de, en iyi çözümü bulması garanti olmamaktadır. Fakat yapılan çalışmalar genetik algoritmalar ile yapılan optimizasyon çalışmalarının çoğunda ulaşılan sonuçların, uygulanabilirlik açısından, diğer optimizasyon teknikleriyle elde edilenlerden daha kullanılabilir olduğunu göstermiştir (Aydın 2000).

Genetik algoritmaların çoğu Goldberg (1989) tarafından önerilen basit genetik algoritmanın değişik şekilleridir. Goldberg tarafından önerilen basit genetik algoritma üç temel genetik operatör olan kopyalama, çaprazlama ve mutasyon operatörlerini içermektedir. Basit genetik algoritmada kopyalama işlemi Darwin' in doğal seçim ve en iyilerin hayatta kalması, zayıfların varlıklarının sona ermesi prensibinin kullanımınıdır. Çaprazlama operatörü, seçilen ebeveyn çözüm dizileri arasından seçilen dizilerin belirli kısımlarının birbirleri ile yer değiştirilmesi şeklinde uygulanmaktadır. Çaprazlama operatörü ile popülasyona yeni çözüm dizisi seçenekleri eklenmiş olmaktadır. Mutasyon operatörü çözüm dizisinde gelişigüzel değişiklikler yapmaktadır. Bir genetik algoritmada mutasyon operatörü belirli bir rutine girmiş olan popülasyonu farklı yönlerde çekmek açısından faydalı olabilmektedir (Aydın 2000).

5.1.3. Bulanık mantık

Aristo mantığına göre insanlar boy bakımından uzundur veya kısadır. Oysa uzun boyluluk kavramı kişilere göre değişmektedir. Uzun boylu insanlardan bir tanesi gerçek uzun boylu olarak alınırsa, bundan biraz daha kısa veya uzun olanlarda değişik üyelik dereceleri ile uzun boylular kümesine dahil olabilirler. Yani küme dışı kalmazlar. Gerçek uzun boylu

olandan çok daha kısa olanlarda, belli bir üyelik derecesi ile yine uzun boylular kümesine dahil olabilmektedir. Bulanık kümelerdeki elamanların üyelik derecelerinin, 0 ile 1 arasında değişebileceğini ileri sürülmekle, kümeler teorisinde yeni olan ve gerçek hayatla uyumlu kümeler metodunda geniş uygulamaya sahip bulanık küme teorisi geliştirilmiştir (Şen 2001).

Bir kümenin elemanları, $[0,1]$ arasında değişen üyelik dereceleri olarak o kümeye dahil oluyorsa bu tür kümelere bulanık kümeler denir. Klasik kümeler teorisinde bir eleman herhangi bir kümeye aittir veya ait değildir. Yani üyelik dereceleri ancak sıfır veya bir olabilir. Sıfır ile bir arasında herhangi bir değer alamazlar. İşte, bulanık kümeler, klasik kümelerin genişletilmiş hali olup; üyelik dereceleri $[0,1]$ aralığında sonsuz değer alabilir. Bir elemanın bir kümeye ait üyelik derecesi "0" ise o kümeye ait değildir, "0.3" ise o kümeye %30 aittir, "0.7" ise o kümeye %70 aittir, ve "1" ise o kümeye %100 aittir, denilir. (Şen, 2001).

5.1.4. Uzman sistemler

Uzman sistemler, herhangi bir karmaşık sistemde, uzman bir kişinin yaptığı işleri yapan bir bilgisayar programı gibi düşünülebilir. Uzmanların düşünce biçimini taklit ederek, özelleşmiş bir alanda önemli problemleri çözmeye yönelik yazılımlardır. Bu yazılımlar genellikle ara yüz, veri tabanı, kural tabanı gibi birden fazla programdan oluştuklarından, program değil 'sistem' olarak adlandırılır (Emiroğlu 1999).

Uzman sistemler, algoritma kaynaklı sistemlerden ziyade bilgi sistemlerinden türetilmektedirler. Bir uzman sistem açık bir algoritmaya sahip olmayan problemlerin çözümüne ilgilendir. Özellikle denklemlerle gösterimi ve nicelik olarak ifadesi zor olan problemlerin analizinde oldukça kullanışlı olmaktadır. Uzman sistemlerle klasik programlama teknikleri arasındaki ana fark uzman sistemlerin bilgileri kullanarak sonuç çıkarma mekanizmasına sahip olmalarıdır (Emiroğlu 1999).

5.1.5 Yapay arı kolonisi

Doğada topluluklar halinde yaşayan canlıların sürü zekasındaki özellikleri temel alan ve sürünün davranışını modelleyen yapay zeka tekniklerinden bir tanesi de yapay arı koloni

algoritmasıdır. Gerçek bir arı kolonisinde, herhangi bir merkezi otorite olmadan kendi aralarında iş dağılımını organize edebilme yeteneği vardır. Yiyecek kaynağı bulan kaşif arılar kaynaktan aldıkları nektarı kovana taşırlar. Bu kaynağın yeri ve kalitesi hakkındaki bilgi paylaşımını kendi aralarında dans ederek ve antenlerini birbirlerine dokundurarak yaparlar. Yiyeceğin tatlılığı, kıvamı, kovandan uzaklığı, hava durumu ve günün hangi vaktinin olduğu kaynağın uygunluğunu etkileyen etmenlerdir. Bu bilgiler dans esnasında gözcü arılara iletilir ve gözcü arılar hangi kaynağın daha verimli olduğuna karar verirler. Buna göre kaynaktan kovana yiyecek taşırlar. Kaynakta yiyecek tükendikten sonra nektar taşıyan arılar geri bildirim yaparak artık bu kaynağın tükenmiş olduğunu o kaynağı bırakıp başka yiyecek kaynakları aramaları gerektiğini kâşif arılara bildirirler. Aynı döngü kaşif arılar vasıtasıyla eş zamanlı ve rastgele olarak tekrarlanır (Karaboğa 2004).

5.2. Gerçek Karınca Kolonisi Davranışı

Doğada topluluklar halinde yaşayan, kendi aralarında iş paylaşımı yapan ve karşılaştıkları problemleri beraberce çözen canlılara sosyal canlılar denir. Arılar, göçebe kuşlar ve karıncalar gibi sosyal canlıların problem çözme yöntemleri insanoğlunun dikkatini çekmiş ve karşılaştıkları sorunların çözümüne model olmuştur. Sürü zekası olarak adlandırılan bu davranışlarda problem çözümlerine ilham kaynağı olan asıl faktör sürünün sosyal davranışlarında bireyselliğin ön plana çıkarılmadan, herhangi bir liderliğe ihtiyaç duyulmadan sadece bireylerin kendilerine ait basit görevleri gerçekleştirmesi ile karmaşık problemlere kısa sürede çözüm bulabilmeleridir. Örneğin, karıncalar bireysel olarak yetersiz olmalarına rağmen koloni olarak hareket ederek yiyecek bulma, yiyeceği yuvaya ulaştırabilmeleri yanında, bu işlemi en kısa yoldan en kısa sürede yapabilme kabiliyetine sahiptirler. Karıncaların bunu bir davranış özelliği olarak yapmaları insanoğlunun gözünden kaçmayacak kalitede bir problem çözme modelidir.

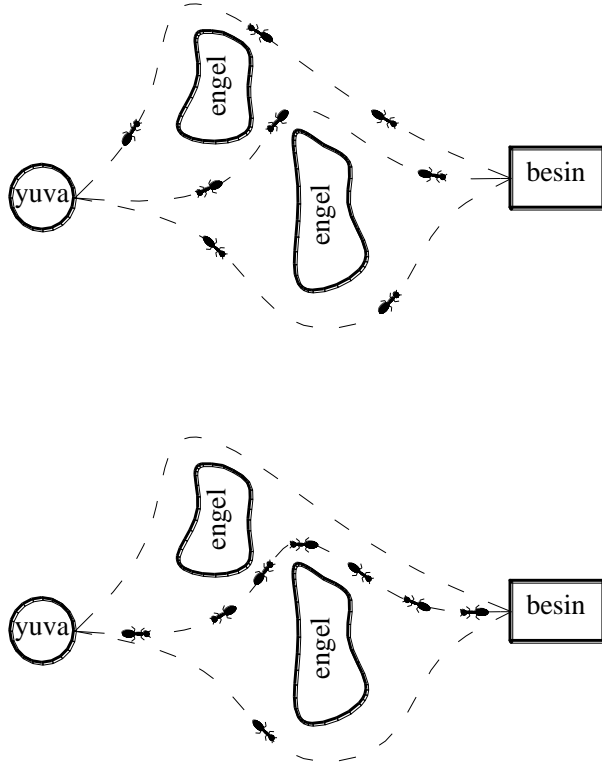
Karıncaların yuva-yiyecek-yuva döngüsü, öncü karıncaların yiyecek aramasıyla başlar. Yuvaya en yakın yiyeceği bulan karınca, beklenildiği gibi, en çabuk geri dönen karınca olacaktır. İşte burada karıncaların birbirlerini takip etme özelliğini ortaya çıkaran kimyasal feromon maddesi devreye girmektedir. Feromon, karıncaların hareket halinde iken gittikleri yola bıraktıkları kokudur. Bu koku karıncaların izini gösteren bir feromon yolu olur. Diğer karıncaların feromon maddesinin yoğun olduğu yolu takip etme olasılığı, daha az feromon yoğunluğu olan yolu takip etme olasılığından daha fazladır. Ancak diğer karıncalar

tarafından feromon yoğunluğunun az olduğu yolun veya hiç feromon bulunmayan bir yolun seçilmesi de ihtimal dahilindedir.

Karıncaların, değişen zaman aralıklarında, farklı miktardaki ve yoğunluktaki feromonlara gösterdikleri davranışlar gözlenmiş ve feromonların buharlaşma oranı, emilim oranı, difüzyon sabiti gibi faktörlere bağlı oldukları gözlenmiştir. Feromonların etki süreleri, karınca türlerine, koloni büyüklüğüne, enzim yapısına bağlı olarak birkaç saatten birkaç aya kadar devam edebilmektedir.

Karıncaların hızlarının hemen hemen aynı olduğunu varsayarsak, birim zamanda kısa yol üzerinden geçen karıncaların sayısı uzun yoldan geçenlerin sayısından daha fazla olacaktır. Kolonideki karıncaların yaklaşık olarak geçtikleri yola eşit seviyede feromon maddesi bıraktıklarını kabul edersek, birim zamanda kısa yola bırakılan koku miktarı daha fazla olacaktır. Yukarıda belirtildiği gibi yön tercihinin koku miktarı ile bağlantılı olması, yeni gelen karıncaların kısa yolu daha yüksek ihtimalle seçmelerine ve bu yola daha fazla kokunun depolanmasına sebep olmaktadır. Böylece, kısa sürede kolonideki çoğu karınca yeni kısa yolu seçerek değişmiş olan çevreye adapte olmaktadır. Bu olaydan anlaşılacağı üzere pozitif geri besleme söz konusudur. Bir yoldan geçen karınca sayısı arttıkça yola yapıştırılan koku miktarı artmakta ve koku miktarı arttıkça da yolu tercih eden karınca sayısı artmaktadır. Bu olay oto-katalitik işlem olarak da adlandırılır. Pozitif geri besleme olayından dolayı, karıncaların yoğunluğunun kısa süre içerisinde daha kısa yolu seçmesi gerçekleşmektedir (Karaboğa 2004).

Bundan sonraki sayfalarda bulunan şekillerde de görüldüğü gibi, öncü karıncalar yuvalarından yemek aramaya farklı yolları rastgele tercih ederek başlarlar. Yol üzerinde karşılaştıkları engelleri tamamen rastgele yaptıkları yön tercihleri sonucu aşarlar. Yemeğe ulaşan karıncalar bir miktar yemek alıp tekrar yuvaya dönerler. Bu esnada yola kendi salgıladığı feromon maddesini bırakırlar. Yeterli yemek depolamak için, bu gidip gelme işini birçok kez tekrarlamaları gerekir. Yola bıraktıkları feromon izlerini takip etme olasılığı daha çok olmak üzere yine rastgele yol tercihleri yaparak yola çıkarlar. Bu döngü böyle devam ettikçe, yol seçenekleri arasından en kısa olanda daha fazla feromon birikir ve her bir karıncanın bu yolu tercih etme olasılığı daha da artar. Nihayetinde görülür ki tüm karıncalar en çok feromon yoğunluğunun olduğu yani kısa yolu kullanmaktadırlar (Şekil 5.1).



Şekil 5. 1. Karıncaların yol tercihi

5.3.Karıncı Kolonisi Algoritma Çeşitleri

Karıncı Koloni Algoritmalarının hepsi temelde gerçek karıncı kolonisi davranışı model alınmakla birlikte, farklı matematiksel formülasyonlara sahip çeşitli algoritmalarda geliştirilmiştir. Bu algoritmaların hemen hepsinde doğal karıncaların yemek arama yöntemleri örnek alınmasına rağmen yapay karıncalara, doğal karıncaların yeteneklerine ilave yetenekler verilmiştir.

Karıncı Kolonisi Algoritmaları önceki bölümlerde bahsedilen gerçek bir karıncı kolonisinin yiyecek arama sırasında yaptığı optimizasyonun yapay bir versiyonudur. Gerçek karıncalar yemek arama işlemi sırasında, yollarını feromon yoğunluğuna göre belirlerler ve karıncı geçiş sıklığına göre feromon yoğunluğu sürekli değişir. Yapay karıncalar ise yönedikleri problem bölgesinde sınırlı biçimde depolanmış bazı sayısal bilgileri değiştirirler. Bu bilgi karıncaların güncel geçmişini/performansını da hesaba katmakta ve bölgeye giriş yapan herhangi bir karıncı tarafından okunabilmekte/yazılabilmektedir. Karıncı Kolonisi

Optimizasyonu Algoritmalarında genellikle, gerçek feromon buharlaşmasına benzer bir buharlaşma mekanizması feromon bilgisini zaman içinde değişikliğe uğratar. Feromon buharlaşması karınca kolonisinin geçmiş tercihlerini yavaşça unutmasını sağlar. Böylece koloni, geçmiş kararlar tarafından etkilenmeksizin arayışını yeni yollara kaydırabilir (Urgan 2011).

Yapay ve gerçek karıncaların temel amacı, bir kaynak (yuva) ile hedefi (yiyecek) birbirine bağlayan en kısa (en az maliyetli) yolu bulmaktır. Yapay karıncalar da gerçek karıncalar gibi, birbirine yakın durumlarda hareket etmek için olasılıklı bir karar politikası uygulayarak çözümler üretirler. Yapay karıncalar bir taraftan, gerçek karınca kolonilerinde olduğu gibi en kısa yolu bulma amacının merkezinde yer alan davranış karakterlerinin bir soyutlamasıdır. Diğer taraftan da, doğal karşılığı bulunmayan bazı yeteneklerle zenginleştirilmişlerdir. Yapay karıncalara, gerçek karıncaların davranışlarına hiçbir şekilde uymayan bazı yetenekler eklenmesi, onları daha etkili ve randımanlı hale getirmiştir. (Urgan 2011).

Gerçek karıncaların yeteneklerine ilaveten, yapay karıncalara verilen bazı yetenekler şunlardır:

- Zamanı kesikli bir değişken olarak kullanırlar.
- Ziyaret ettikleri noktaları belleklerinde tutarlar.
- Elde edilen çözümün kalitesi ile orantılı olarak çözümü oluşturan bağlar üzerine iz bırakırlar. Yapay karıncalar bulunan çözümün niteliğine ait işlev yerine getirecek miktarda feromon bırakırlar.

-Yapay karıncaların feromon bırakma zamanlamaları probleme bağlıdır ve genellikle gerçek karıncaların davranışlarını yansıtmaz. Örneğin, birçok durumda yapay karıncalar feromon izlerini sadece yeni bir çözüm ürettikten sonra tazelerler.

- Tamamen kör değildirlir, aday noktalar hakkında sezgisel bilgiye sahiptirler.

Karıncalar arasındaki kolektif iletişim ve sezgisel bilginin kullanımı ile karıncalar, arama uzayının umut vaat eden bölgelerine yönlendirilmektedir. Karıncalar, arama uzayı hakkındaki bilgiyi çözüme ulaşma aşamasında yollar üzerine bırakılan iz miktarı ile depolamaktadırlar. Karıncalar çözüm için, bırakıldıkları ağ üzerindeki düğümlerden hangi tarafa gideceklerini yollar üzerindeki iz miktarlarını ve yolların uzunluklarını değerlendirerek karar vermektedirler. Kısa olan yolda bir süre sonra feromon miktarı uzun yollara göre daha

fazla biriktiğinden, yani kısa yoldan geçiş daha hızlı gerçekleşeceğinden, birim zamanda geçiş yapan karınca sayısı uzun yola göre daha fazla olacaktır. Dolayısıyla herhangi iki düğüm arasındaki yol üzerinde bulunan feromon miktarı, yolun uzunluğuyla ters orantılı olacaktır. Bu karar aşamasında kullanılan kural, kısa ve iz yoğunluğu fazla olan yolun seçilme olasılığı daha fazladır şeklinde açıklanabilir. Dolayısıyla iyi çözümler için daha fazla iz bırakılmaktadır. Ayrıca yollar üzerinde iz yoğunluğu oluşmasını engellemek için izlerin buharlaşması sağlanmaktadır. Böylece aramanın erken safhalarında alt-optimal bir çözüme yakınsama engellenmekte, arama uzayının yeni bölgelerinin keşfedilmesine olanak sağlamaktadır (Urgan 2011).

5.3.1. Karınca Sistemi Algoritması

Karınca sistem algoritması; karınca-döngü (Ant-Cycle), karınca-yoğunluk (Ant-Density) ve karınca-miktar (Ant-Quantity) olmak üzere üç farklı algoritma kümesinden oluşmaktadır. Bunların arasındaki fark ise feromonun depolanmasında yatmaktadır. Karınca-döngü algoritmasında feromon sadece her karınca turunun sonunda o karıncanın turu üzerindeki her kenar, uzunluklarıyla ters orantılı olarak feromon depolanmaktadır. Bununla birlikte her sanal karınca gerçeğinin aksine geçtiği yolları hatırlamak için bir hafızaya sahiptir. Karınca-yoğunluk ve karınca-miktar algoritmalarında ise karıncalar iki düğüm noktası arasında yaptıkları her hareketin sonunda iz bırakırlar. Feromon yolu, karınca bir noktadan diğer bir noktaya geçerken güncellenmektedir. Yapılan testler sonunda karınca döngü algoritmasının daha iyi sonuçlar verdiği ortaya çıkmış ve genel bir yapı kazanmıştır. (Uğur ve Aydın 2006).

Bu algorithmada yapay karıncalar, gezgin satıcı probleminin ağ yapısı üzerindeki yollarda bir düğümden (şehir) başka bir düğüme hareket etmek için olasılıklı bir geçiş kuralı kullanmaktadırlar. Başlangıçta her karınca ağ üzerinde rastgele bir düğüm üzerine yerleştirilir. Daha sonra her karınca daha önce ziyaret etmediği düğümlere hareket ederek bütün düğümleri içeren bir çözüm oluşturur. Sistemin temel özelliği, her yinelemede feromon değerlerinin, yinelemenin kendi içinde bir çözüm üretmiş olan tüm karıncalar tarafından güncellenmesidir (Urgan 2011).

5.3.2.Max-Min Karınca Sistemi Algoritması

Max-Min karınca sistemi algoritması ilk kez Stützle ve Hoese (1997) tarafından ortaya koyulmuştur. Bu sistem klasik karınca sisteminden dört noktada ayrılır (Serin ve Terzi 2008).

Birincisi bu algoritma çok iyi bir başarıyla en iyi turları bulur. O turdaki veya o zamana kadar gerçekleşen turlar arasındaki en iyi karıncanın feromon bırakmasına izin verilir. Ancak, böyle bir yöntem, bütün karıncaların aynı yolu izlediği bir kısır döngü durumuna da yol açabilir. Bunun nedeni, iyi fakat optimum olmayan güzergahtaki feromon izlerinin aşırı yükselmesidir. Bu durumu önlemek için, Max-Min Karınca Sisteminin getirdiği ikinci farklılık, ihtimal dahilindeki feromon izi değerlerinin, belirli bir aralıkta sınırlandırılmasıdır. Üçüncüsü, başlangıçta yollarda bulunan feromon miktarının, öngörülen maksimum değere getirilmesidir. Bu şekilde feromon buharlaşma oranıyla birlikte araştırmanın başlangıcında turların keşfedilme ihtimali artırılmış olur. Dördüncü olarak, sistem durgunluğa veya duraklamaya her yaklaştığında veya kısır döngüye girip tekrarlar boyunca geliştirilmiş tur bulamazsa, feromon izleri yeniden başlangıç seviyelerine getirilir (Urgan 2011).

5.3.3. Karınca Koloni Optimizasyonu Algoritması

Karınca Koloni Optimizasyonu algoritmasında önemli oranda Karınca Sistemi Algoritması'ndan esinlenilmiştir. Fakat orijinal Karınca Sistemi Algoritması'nda bulunmayan yeni mekanizmalar ve yaklaşımlar dâhil edilerek Karınca Koloni Optimizasyonu oluşturulmuştur. Bu şekilde daha yüksek optimizasyon başarısı elde edilmektedir (Serin ve Terzi 2008).

Karınca Koloni Optimizasyonu'nun, Karınca Sistemi Algoritması'ndan

- 1- Ark seçme kuralı
- 2- Feromon güncelleme kuralı
- 3- Lokal feromon güncellemesi
- 4- Aday listesi kullanımı

olmak üzere dört ana noktada farklılığı bulunmaktadır.

Birincisi Karınca Koloni Optimizasyonu'nda, Karınca Sistemi Algoritması'na göre, daha atılgan bir seçme kuralı uygulanmasıyla, Karınca Sistemi'ndeki karıncaların birikimi olağan üstü bir başarıya dönüştürülmüştür İkincisi feromon buharlaşması ve feromon

bırakımı, sadece o ana kadarki en iyi tura ait arklarda geçerlidir. Bir tur sonunda bütün karıncalar bir çözüm oluşturduklarında, feromon izi tekrarın başlangıcından o ana kadarki en iyi turu bulan karıncanın kullandığı arklara ilave edilir Üçüncüsü, karıncaların i şehrinde j şehrine hareket ederken kullandıkları bir (i,j) arkındaki feromonun bir kısmı, her defasında, alternatif yolların keşfini arttırmak için, azalır. Bu şekilde farklı bir lokal feromon güncellenmesi gerçekleştirilir Dördüncüsü ziyaret edilecek bir sonraki şehrin seçimini sınırlamak için aday listeleri kullanılır (Serin ve Terzi 2008).

5.3.3.1. Ark seçme kuralı

Karıncı Koloni Optimizasyonunda bir noktadaki karınca diğer noktaya geçmeden önce nokta seçimini belirli bir kural ile yapar. Burada bütün karıncalar hüristik bilginin ve öğrenilmiş feromon izlerinin gösterdiği sonuca göre ihtimal dahilindeki en iyi hareketi yapar. Bu durumda karıncalar da öğrenilmiş bilgiye önem vermiş, bu bilgiye göre hareketlerini düzenlemiş olurlar. Yine rastgele yolların keşfini yaparlar. Belirli parametreler ile ayarlama, keşfetme derecesinin değişim imkânını sağlar ve araştırmanın o ana kadarki en iyi çözüm etrafına mı yoksa turların keşfi etrafına mı odaklanmasını seçmemize imkan sağlarlar (Serin ve Terzi 2008).

5.3.3.2.Feromon güncelleme kuralı

Karıncı Kolonisi Optimizasyonu'nda, programın her tekrarından sonra, sadece o ana kadarki en iyi olan karıncanın feromon eklemesine izin verilir. Karıncı Kolonisi Optimizasyonu'ndaki feromon izi güncellemesi, hem buharlaşma hem de yeni feromon bırakımı, sadece en iyi karıncanın turundaki yollara uygulanır. Karıncı Sistemi Algoritması'nda ise bütün yollara uygulanılmaktadır. Ayrıca Karıncı Kolonisi Optimizasyonu'nda Karıncı Sistemi Algoritması'ndan farklı olarak, bırakılan feromon miktarı belli miktar kadar azaltılarak bırakılır. (Serin ve Terzi 2008).

5.3.3.3. Lokal feromon güncellemesi

Karıncı Koloni Optimizasyonu'nda karıncalar, tur oluşturma esnasında, global feromon izi güncellemesine ek olarak, herhangi iki nokta arasındaki yolu geçtikten hemen sonra lokal bir feromon güncelleme kuralını uygular. Lokal güncelleme kuralının etkisi, her

seferinde bir karıncanın kullandığı bir yoldaki feromon miktarının azaltılmasıdır. Böylece bu yol gelecek olan diğer karıncalar için daha az isteklilik arz edecektir. Bu şekilde henüz ziyaret edilmemiş yolların keşfedilme ihtimali arttırılarak uygulamada algoritmanın bir durgunluk davranışı gösterme etkisi giderilmiş olur ve karıncaların çoğunlukla ortak bir yola yönelmesi önlenir. Başka bir ifadeyle, Karınca Koloni Optimizasyonu'ndaki lokal güncelleme kuralının rolü, turları karıştırmaktır. Böylece bir karıncanın turunda bulunan öncelikle uğranılmış bulunan noktalar, daha sonradan diğer karıncaların turlarında keşfedilebilir. Bu yüzden lokal güncelleme kuralının etkisi, yolların öğrenilmiş istekliliğinin dinamik olarak değiştirilmesidir. Bir karınca aynı yolu her kullandığında, bu yol karınca için daha az isteklilik arz edecektir. Bu şekilde karıncalar feromon bilgisini daha iyi kullanmış olacaklardır. Lokal güncelleme olmasaydı bütün karıncalar önceki en iyi turun yakınlarındaki dar bir alanda araştırma yapmış olacaklardı. Oysa lokal güncelleme ile daha iyi tura yönlendirme sağlanacaktır (Serin ve Terzi 2008).

Diğer bir önemli nokta da Karınca Sistemi Algoritması'nda, karıncalar turlarını oluştururken, paralel veya ardışık çalışma önemli değildir. Fakat Karınca Kolonisi Optimizasyonu'nda, lokal feromon güncelleme kuralından dolayı bir farklılık oluşmaktadır. Gerçekleştirilen çoğu Karınca Kolonisi Optimizasyonu uygulamasında deneysel olarak hangisinin seçiminin daha uygun olduğuyula ilgili bir kanıt bulunmamasına rağmen, genellikle bütün karıncaların paralel hareket ettirilmesi tercih edilmektedir (Serin ve Terzi 2008).

5.3.3.4. Aday listesi kullanımı

Karıınca Koloni Optimizasyonu, bir aday listesi yapısını kullanır. Bu genellikle büyük gezgin satıcı problemi örneklerini çözmeye kullanılan bir veri yapısıdır. Aday listesi, bir şehirden sonra öncelikle veya en yakın ziyaret edilmesi istenen şehirlerin listesidir. Verilen herhangi bir şehirden ihtimal dahilinde gidilecek bütün şehirleri incelemek yerine öncelikle aday listesindeki ziyaret edilmeyen şehirler incelenir. Aday listesindeki şehirler artan uzaklıklarına göre sıraya konulmuşlardır ve liste düzenli olarak taranır. Aday listesi yapısına sahip Karınca Koloni Optimizasyonu modeli ile çözülen gezgin satıcı problemi şu şekilde çalışmaktadır:

Bir karınca öncelikle bir sonraki şehir seçimini listedekilerle sınırlar. Diğer şehirleri, bu listedeki şehirlerin tamamı ziyaret edilmiş ise hesaba katar. Eğer aday listesinde hala

şehirler var ise, bir sonraki şehir seçilir. Aksi halde henüz ziyaret edilmemiş liste dışı şehirlerden en yakın olanı seçilir (Serin ve Terzi 2008).

5.3.4. Sıralı Karınca Sistemi

Sıralı Karınca Sisteminde, her karınca rütbesine göre artan bir miktarda feromon bırakır. Aynı zamanda, o ana kadar ki en iyi karınca her tekrarda daima en yüksek feromonu bırakır. Bu sistemde, geçerli iterasyondaki en iyi belli sayıda karıncanın feromon güncellemesine izin verilir. Bu amaçla karıncalar tur uzunluklarına göre belirli formülasyonlara göre sıralanır ve bir karıncanın feromon düzeyi karıncanın sıra değerine bağlıdır. Böylece mevcut en iyi çözümün feromon güncelleme düzeyinin en yüksek olması sağlanmış olur (Urgan 2011).

5.3.5. Çoklu Karınca Koloni Algoritmaları

Çoklu Karınca Koloni yaklaşımı, iki veya daha fazla koloninin çözümünün birleştirilmesi esasına dayanır. Bu konudaki temel yaklaşımdan ilki olan heterojen yaklaşım tüm kolonilerin birbirinden farklı davranış sergilemesi esasına dayanmaktadır ve çok kriterli optimizasyon problemlerinin çözümünde kullanılmıştır. İkinci yaklaşımda tüm koloniler birbirine paralel olarak çalışırlar ve her bir jenerasyon sonunda koloniler arasında feromonlar vasıtası ile karşılıklı bilgi alışverişi olur. Sözü edilen diğer yaklaşım, bilgi alışverişinin sınırlı olduğu ve çok kısa olmayan aralıklarla yapıldığı takdirde başarılı sonuçlar vermektedir (Urgan 2011).

Bilgi alışverişi için önerilen dört yöntem vardır.

- Her bilgi alışverişinde global en iyi çözümler hesaplanır ve tüm kolonilere gönderilir. Böylece her koloni yeni bir lokal en iyi çözüme sahip olur.
- Koloniler arasında bir köprü kurulur ve her bilgi alışverişi adımında, her koloni lokal en iyi çözümleri en yakınındaki bir ya da daha fazla koloniye gönderir.
- İkinci yöntemdeki bilgiler kullanılarak her bilgi alışverişi adımında, bir kolonideki en iyi karınca, en yakınındaki en iyi karınca ile karşılaştırılır ve karıncanın en iyi olanları alınarak feromon güncellemesinde kullanılır.
- Bu seçenek ikinci ve üçüncü yöntemlerin birleştirilmesinden oluşur.

5.3.6 Global Karınca Kolonisi Algoritması

Global Karınca Kolonisi Optimizasyonu, tüm düğümlere uğrama sınırlaması ihtiyacı olmayan problemlerin çözümüne yönelik geliştirilmiş olan bir Karınca Koloni Optimizasyon Algoritmasıdır. En kısa yol problemi bunlara örnek gösterilebilir. Bu algoritma, en kısa yol problemi olarak modellenen gezgin satıcı problemine uygulanmıştır. Geliştirilmiş diğer Karınca Koloni Optimizasyonu sistemlerinden farklı olarak, karıncaların tam bir tur yapma zorunlulukları yoktur. Herhangi bir düğümden başlayarak ve herhangi bir ya da birden fazla düğüme uğrayarak bir çözüm alternatifi geliştirmeleri mümkündür. Sıra bozulmamak üzere turlarda herhangi bir kısıtlama yoktur. Tek başına bir ya da daha fazlası, bir tur sayılabileceği gibi tüm düğümlere uğrayarak da tur tamamlanabilir (Keskintürk Söyler 2006).

6. KARINCA KOLONİ OPTİMİZASYON ALGORİTMASI İLE OPTİMUM TASARIM PROBLEMİNİN ÇÖZÜLMESİ

Literatürde farklı Karınca Kolonisi Optimizasyonu yöntemleri kullanılmakla beraber bu yöntemlerin optimizasyon problemine uygulanmasında da farklılıklar olabilir. Bu çalışmada kullanılan yöntemde sanal karıncanın yol aldığı güzergâhın uzunluğu veya kısalığından ziyade o güzergâh üzerinde bulunan durakların yani tasarım değişkenlerinin önemi vardır. Problemin başlangıç aşamasında belirlenen bu duraklara yüklenen değerler, optimum tasarım probleminin çözümünü vermektedir.

6.1. Karınca Koloni Algoritmasının Düzlem Kafes Sistem Optimizasyonuna Uyarlanması

Bu bölümde doğal karınca kolonilerinin davranışı, basitleştirilmiş bir Karınca Koloni Algoritması ile bir yapısal optimizasyon probleminin çözümüne uyarlanacaktır. Bu uyarlama neticesinde doğal karıncalar için yuva ve yem kavramları Karınca Koloni Algoritmasında sırasıyla aramanın başlangıç noktası ve amaç fonksiyonuna benzetilebilir. Yapay kolonideki her bir karıncanın başlangıç noktasından çıkarak amaç fonksiyonuna ulaşmak için takip ettiği güzergah yapısal problemin muhtemel bir çözümünü ifade etmektedir. Dolayısıyla her bir yapay karınca başlangıç noktasından çıkarak bir amaç fonksiyonuna ulaşabilmek için probleme ait her bir tasarım değişkeninin önceden belirlenmiş değerlerinden birine uğramalıdır. Bu yaklaşımla her bir tasarım değişkeni başlangıçtan amaç fonksiyonuna giden yoldaki bir istasyona ve tasarım değişkenleri için önceden belirlenen olası değerler ise istasyonlardaki peronlara benzetilebilir. Bu benzetmeye bağlı olarak da her yapay karıncanın her istasyonun sadece bir peronunda durmak suretiyle amaç fonksiyonuna ulaşacağı söylenebilir.

Gerçek karıncalar için takip ettiği yolun ne kadar iyi olduğu bu yolun ne kadar kısa olduğu ile ölçülürken, yapay karıncaların takip ettiği yolun iyilik derecesi uğradığı tasarım değişkeni değerleri ile hesaplanacak cezalandırılmış amaç fonksiyonu değerine bağlı olarak karar verilir. Bazı yapısal problemlerde cezalandırılmış amaç fonksiyonun maksimize edilmeye çalışılırken, bazı yapısal problemlerde bu değer minimize edilmeye çalışılır. Bu çalışmada cezalandırılmış amaç fonksiyonu minimize edilmeye çalışıldığından, ulaşılan cezalandırılmış amaç fonksiyonu değeri ne kadar küçük ise yapay karıncanın takip ettiği yol da o kadar iyi kabul edilecektir.

Optimizasyonun her bir iterasyonunda, yapay kolonideki her karınca rastgele belirlenecek yolları takip ederek bir cezalandırılmış amaç fonksiyonu değerine ulaşacak ve kolonideki tüm karıncaların ulaştığı cezalandırılmış amaç fonksiyonu değerleri karşılaştırılarak hangisinin daha iyi olduğuna karar verilecektir.

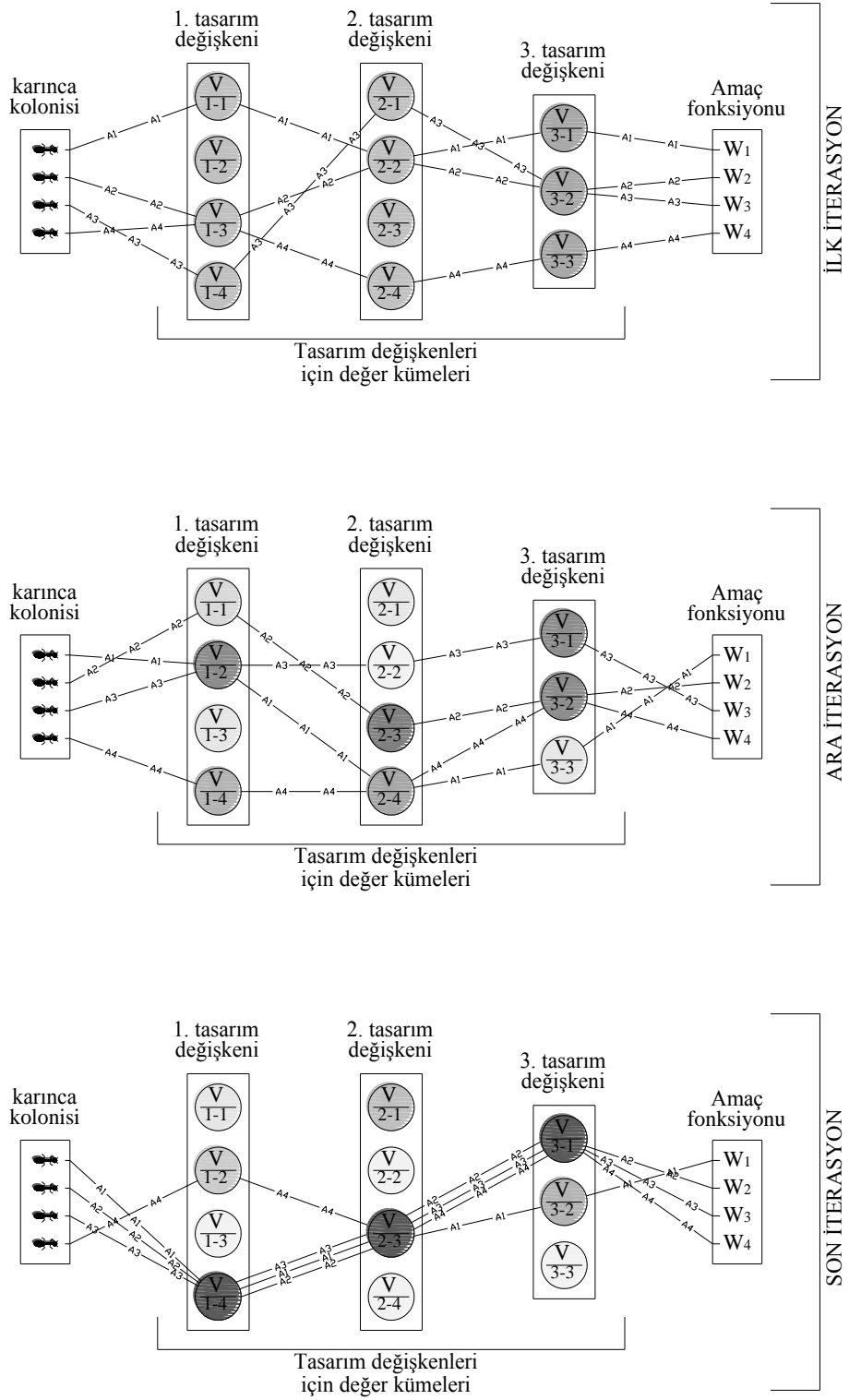
Doğal karınca davranışından farklılık olarak bu çalışmada kullanılan Karınca Koloni Algoritması'nda sadece en iyi karıncanın izlediği yoldaki feromon miktarları artırılır. Böylece en iyi karıncanın uğradığı tasarım değişkeni değerlerindeki feromon miktarları artırılırken diğer tüm tasarım değişkeni değerleri eşlenikli olarak azaltılır. Bu durumda en iyi karıncanın takip ettiği tasarım değişkeni değerlerinin bir sonraki iterasyonda seçilme olasılıkları daha yüksek olacaktır. Böylece devam eden iterasyonlar sonucunda bazı tasarım değişkeni değerlerindeki feromon miktarları oldukça artacak ve artık sonraki iterasyonlardaki karıncaların çok büyük bir bölümü bu tasarım değişkeni değerlerini takip ederek amaç fonksiyonuna ulaşacaklardır. Bu şekilde başlangıç noktasından amaç fonksiyonuna giden en kısa yol bir başka değişle yapısal optimizasyon probleminin optimum çözümü bulunmuş olacaktır.

Doğal karıncalar için yuva ile yem arasında sonsuz sayıda güzergah varken, yukarıda anlatılanlardan anlaşılacağı üzere, yapay karıncalar için tasarım değişkeni sayısına ve bu tasarım değişkenleri için seçilen değer kümelerindeki değer sayısına bağlı olarak sınırlı sayıda güzergah seçeneği (problemin muhtemel çözümü) bulunmaktadır. Doğal olarak problem için belirlenen tasarım değişkeni sayısı ve bu tasarım değişkenleri için belirlenen değer kümelerindeki değer sayısı arttıkça, karıncaların amaç fonksiyonuna ulaşmak için takip edecekleri güzergah sayısı artacaktır. Bu durumda bir yandan optimum yolun bulunma süreci artarken diğer yandan da daha fazla yol sınanacağı için daha iyi bir yol (daha iyi bir çözüm) bulma olasılığı artacaktır. Bu bakımdan tasarım değişkenleri için belirlenen değer kümelerindeki değer sayısının tespiti önemlidir.

Bu çalışmada düzlem kafes sistemlerin optimum tasarımı gerçekleştirilmektedir ve doğal olarak kafes sistemdeki çubuk sayısı kadar da tasarım değişkeni olacaktır. Ancak bu çubuklar belli bir sistematik içinde gruplandırılacak olursa, bu durumda tasarım değişkeni sayısı çubuk sayısından daha az olarak grup sayısı kadar olacaktır. Uygulamada da kafes sistemlerin projelendirilmesinde çubukların gruplandırılması yaygın bir yöntemdir.

Aşağıda Şekil 6.1’de bir düzlem kafes boyutsal optimizasyon probleminin Karınca Kolonisi Algoritması ile çözümüne ilişkin bir örnek gösterilmiştir. Bu örnekte 3 adet tasarım değişkeni bulunmaktadır. Bu durumda örnekteki kafes sistemin 3 adet çubuktan oluştuğu veya kafes sistemi oluşturan çubukların 3 grupta toplandığı düşünülebilir. Sırasıyla birinci, ikinci ve üçüncü tasarım değişkeni için 4, 4 ve 3 adet değer içeren tasarım değişkeni değer kümeleri belirlenmiştir. Şekil 6.1’de tasarım değişkeni değerlerindeki feromon miktarları renklerle temsil edilmekte olup daha koyu renkler daha çok feromonu daha açık renkler daha az feromonu ifade etmektedir. Bu durumda son iterasyondaki en koyu renklere sahip tasarım değişkeni değerleri optimum çözüme ait değerleri ifade etmektedir.

Bu örnekten de görüleceği üzere başlangıçta bütün olası güzergahlarda eşit miktarda feromon bulunmakta ve kolonideki karıncaların izledikleri yollar birbirinden farklı olmaktadır. İterasyonlar ilerledikçe bazı güzergahlar üzerindeki feromonlar artarken bir çoğunda azalmaktadır. Feromon miktarlarının artmış olduğu güzergahlar ise sonraki aşamalarda daha fazla karınca tarafından takip edilmektedir. En son aşamada ise sadece bir güzergah üzerindeki feromon miktarları diğerlerinden belirgin şekilde fazla olmakta ve kolonideki karıncaların çok büyük bir çoğunluğu bu güzergahı takip etmektedir. Feromon miktarlarının en fazla olduğu güzergah ise problemin en iyi çözümünü temsil etmektedir.



Şekil 6. 1. Karınca Koloni Algoritmasının kafes sistem optimizasyonuna uyarlanması

6.2. Bu Çalışmada Kullanılan Karınca Koloni Algoritması

Bu çalışmada literatürdekilere oranla basitleştirilmiş bir Karınca Koloni Optimizasyonu kullanılmıştır. Kullanılan algoritmanın basitleştirilmiş olarak nitelendirilmesinin sebebi, literatürde kullanılan diğer Karınca Koloni Algoritmalarına oranla, aşağıda açıklandığı şekilde, çok daha kısa ve az sayıda işlem ile sonuca gidiliyor olmasıdır.

Doğadaki karınca davranışında, bir karınca kolonisi yem aramaya başladığında yuva ile yem arasındaki güzergahta başlangıçta hiç feromon bulunmamaktadır. Bu durum literatürde kullanılan birçok karınca koloni algoritmasında da bu şekildedir. Ancak bu çalışmada kullanılacak olan “ Karınca Koloni Algoritmasında” araştırmanın başlangıcında, problemin muhtemel tüm çözümlerini temsil eden yollarda, bir miktar feromon bulunduğu kabul edilecektir. Başlangıçta optimum çözüm hakkında herhangi bir bilgi sahibi olunmadığından, her bir tasarım değişkeni için, seçilen değerlerin her birinde eşit miktarda feromon olduğu düşünülecektir. Bu ilk feromon miktarları i 'inci tasarım değişkeninin j 'inci değeri için aşağıda verilen bağıntı ile hesaplanabilir.

$$P_{ij}^1 = \frac{1}{nv_i} \quad (6.1)$$

Bu bağıntıda nv_i i 'inci tasarım değişkeni için seçilen değerler kümesindeki tasarım değişkeni sayısını ifade etmektedir. Böylece her bir tasarım değişkenine ait değerler için feromon miktarlarının toplamı “1” e eşit olacaktır ve bu toplam değer optimizasyon süreci boyunca değişmeyecektir.

Yukarıdaki bağıntı ile hesaplanan feromon miktarı her iterasyondaki en iyi çözümün takip ettiği değerler için artırılırken diğer değerler için azaltılmalıdır. Daha önce de söz edildiği gibi bu işleme feromon güncelleme adı verilmektedir. Karınca Koloni Algoritması'nda feromon güncellemesi sonucu yeni feromon miktarları k 'inci iterasyonda i 'inci tasarım değişkeninin j 'inci değeri için aşağıdaki bağıntılar ile hesaplanacaktır.

$$\left. \begin{aligned} P_{ij}^k &= P_{ij}^{k-1} \cdot \left(1 - \frac{F \cdot nv_i}{nv_i - 1}\right) & \text{if } j \neq V_i^k \\ P_{ij}^k &= \frac{1 - P_{ij}^{k-1}}{P_{ij}^{k-1}} \cdot \frac{F \cdot nv_i}{nv_i - 1} & \text{if } j = V_i^k \end{aligned} \right\} \quad (6.2)$$

Yukarıdaki bağıntıların ikincisi ile en iyi çözümün kullanmış olduğu değerin feromon miktarı artırılırken, birincisi ile diğerlerin feromon miktarı azaltılmaktadır. Bu bağıntılarda V_i^k en iyi çözüm için i 'inci tasarım değişkenin aldığı değerin sıra numarasının k 'ıncı iterasyondaki değerini ifade etmektedir. F en iyi çözüme ait tasarım değişkeni değerlerine ne oranda feromon bırakılacağını belirleyen bir katsayı olup, feromon güncelleme katsayısı olarak isimlendirilmektedir. Feromon güncelleme katsayısının en iyi değeri problemde problemde değişebilmekte olup, bu değer deneyimlerden yola çıkarak deneme yanılmayla bulunabilir. Feromon güncelleme katsayısının belirlenmesi için bu çalışmada bir inceleme sunulmaktadır.

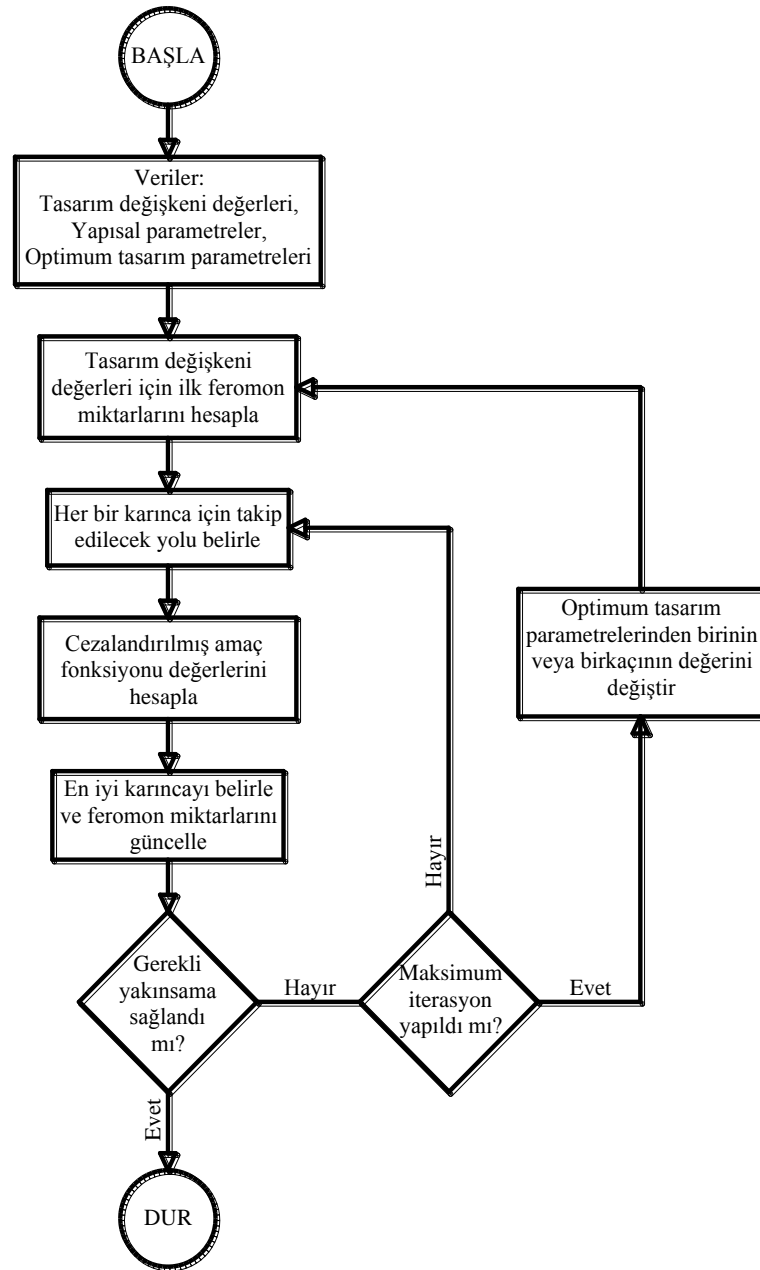
Karınca Koloni Algoritması'nda en iyi çözümün takip etmediği tasarım değişkeni değerlerinin feromon miktarlarındaki azalmaktadır. Bu durum doğal karınca kolonisi davranışında, yuva ile yem arasındaki daha az tercih edilen güzergahlardaki feromon miktarının buharlaşmasına benzetilebilir.

Karınca Koloni Algoritması'na ait iterasyonların sonlanması için iki ölçütten biri gerçekleşmelidir. Bu ölçütlerin birincisi ve esas olanı her bir tasarım değişkeni için herhangi bir değerin feromon miktarının önceden belirlenen bir orandan fazla olması koşulunun sağlanmasıdır. Önceden belirlenen bu oran ise "0.50" den daha küçük alınmamalıdır. İkinci ölçüt ise yine iterasyonlara başlamadan önce belirlenen maksimum iterasyon sayısına ulaşılma durumudur.

Bu ölçütlerden birincisinin gerçekleşmesi durumunda optimum çözüme ulaşılmış sayılarak, her bir tasarım değişkeni için en büyük feromon miktarına sahip değerler ile optimum çözüm ifade edilir. İkinci ölçütün birinciden önce gerçekleşmesi durumunda optimum tasarım parametrelerinden (karınca sayısı, feromon güncelleme katsayısı, ceza katsayısı, maksimum iterasyon sayısı) bazılarının değeri değiştirilerek optimizasyon süreci yeniden gerçekleştirilir.

6.3. Akış Diyagramı

Çalışmada anlatılan basitleştirilmiş karınca koloni algoritması ile düzlem kafes sitemlerin optimum tasarımının gerçekleştirilmesi için bir bilgisayar yazılımı kodlanmıştır. Kodlanan bu yazılımın akış diyagramı aşağıda Şekil 6.2’de verilmektedir.



Şekil 6. 2. Karınca koloni algoritmasına ait akış diyagramı

7. UYGULAMALAR

Bu bölümde düzlem kafes sistemlerin minimum ağırlıklı olarak boyutlandırılması amacıyla geliştirilmiş olan bilgisayar programı kullanılarak akademik literatürde çokça rastlanan üç adet düzlem kafes sistemin optimizasyonu yapılmıştır. Üzerinde çalışılan tüm örnekler literatürden seçilip, daha önce yapılan çalışmalarla karşılaştırma yapılmıştır. Örneklerin ağırlık birimleri ton, uzunluk birimleri metre olarak alınıp hesap yapılmıştır. Ancak tasarım verileri belirtilirken literatürdeki birim sistemleri ile yazılmıştır. Visual Basic kodlama dilinde geliştirilen program, 2.0 GB RAM'li ve 64 bit işletim sistemli bir bilgisayar ile koşturulmuştur.

7.1. Programın İrdelenmesi

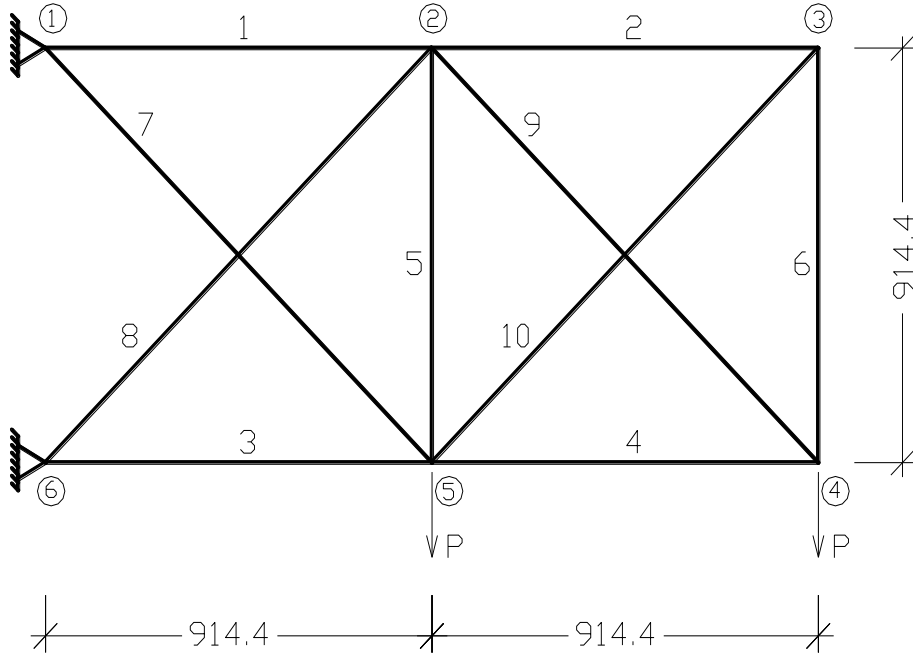
Buradaki amaç, geliştirilen programın düzlem kafes sistemlerin optimizasyon probleminin çözümündeki yeteneğini denetlemektir. Bunu yaparken seçilen bir düzlem kafes sistemin farklı parametreler ile defalarca optimum tasarımı gerçekleştirilmiştir. Böylece bazı tasarım parametrelerinin ulaşılan optimum sonucu ve çözüm süresini ne kadar etkilediği belirlenmektedir.

Bu kapsamda cezalandırılmış amaç fonksiyonundaki ceza katsayısı ve yeni feromon miktarlarının belirlenmesinde kullanılan feromon güncelleme katsayısı için bu probleme ait optimum değerler araştırılmıştır. Bu araştırmalar yapılırken 5 ile 5000 arasında değişen farklı sayıda karınca içeren kümeler kullanılarak optimizasyonlar gerçekleştirilmiş, böylece hem yukarıda bahsedilen katsayılar için daha detaylı bir inceleme yapılmış hem de kullanılması gereken optimum karınca sayısı araştırılmıştır.

Bu çalışmada ayrık tasarım değişkenleri kullanılarak bir optimizasyon gerçekleştirilmektedir. Bu ayrık tasarım değişkenlerinin değerlerinin alabilecekleri muhtemel değerleri içeren kümeler önceden belirlenmektedir. Bu kümelerdeki değer sayısının gerçekleştirilen optimizasyonu nasıl etkilediğini görmek için ise 200, 100, 50, 25 değer içeren kümeler kullanılarak optimizasyon gerçekleştirilmiş ve ulaşılan sonuçlar karşılaştırılmıştır.

Yukarıda bahsedilen incelemeleri yapmak üzere Şekil 7.1'de verilen 10 çubuklu ve 12 düğüm noktalı düzlem kafes sistem seçilmiştir. Şekil üzerindeki ölçüler santimetre

cinsindedir. Bu kafes sistem literatürde sıkça karşılaşılan popüler bir uygulamadır. Seçilen bu örnek üzerinde yaklaşık 360 adet çözüm yapılmış ve kullanılan parametreler ile ulaşılan sonuçlar irdelenip tablo ve çizelgeler halinde gösterilmiştir.



Şekil 7. 1. On çubuklu düzlem kafes sistem

Bu örnekte kullanılan malzemenin birim hacim ağırlığı $\rho = 7.85 \text{ ton/m}^3$, elastisite modülü $E = 20 \cdot 10^6 \text{ ton/m}^2$, akma dayanımı $F_y = 24.000 \text{ ton/m}^2$ alınmıştır. Tüm noktaların deplasman sınırı iki doğrultuda da $\pm 0.0508 \text{ m}$ 'dir. Deplasman sınırlayıcısının yanı sıra gerilme ve narinlik sınırlayıcıları da bu örnekte Bölüm 3.3'te tarif edildiği şekliyle AISC-ASD (1989) standardına uygun olarak dikkate alınmıştır. 4 ve 5 numaralı düğüm noktalarından $P = 44.5 \text{ ton}$ yük uygulanmıştır. Bu örnekte çubuklar gruplandırılmamış olup toplam on adet tasarım değişkeni vardır. Bu tasarım değişkenleri için, DIN 2448 standardından, değerlerini alabilecekleri 200 adet boru kesit seçilmiş olup bunlar Çizelge 7.1'de görülmektedir.

Çizelge 7. 1. On çubuklu kafes sistemin çözümünde kullanılan 200 adet boru kesite ait en kesit alanları ve atalet yarıçapları

Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı
1	11.088	3.467	21	22.515	3.576	41	44.673	3.575	61	28.094	4.458	81	62.203	4.410
2	12.270	3.454	22	25.143	3.547	42	49.431	3.521	62	31.429	4.428	82	20.917	5.231
3	13.733	3.437	23	27.436	3.521	43	15.463	4.351	63	34.350	4.402	83	23.163	5.214
4	15.180	3.420	24	30.800	3.483	44	17.325	4.334	64	38.657	4.363	84	25.837	5.194
5	16.896	3.400	25	33.534	3.451	45	19.171	4.317	65	42.177	4.331	85	28.928	5.170
6	18.869	3.377	26	37.518	3.405	46	21.366	4.297	66	47.339	4.283	86	32.423	5.143
7	21.087	3.350	27	41.862	3.354	47	23.899	4.273	67	53.019	4.230	87	36.306	5.113
8	23.534	3.321	28	46.263	3.302	48	26.755	4.247	68	58.834	4.175	88	39.716	5.087
9	25.666	3.296	29	12.525	3.916	49	29.920	4.217	69	17.059	4.800	89	44.754	5.047
10	28.789	3.258	30	13.866	3.902	50	32.691	4.191	70	19.121	4.783	90	48.884	5.014
11	31.322	3.227	31	15.529	3.885	51	36.771	4.152	71	21.167	4.766	91	54.961	4.966
12	35.004	3.181	32	17.176	3.868	52	40.103	4.120	72	23.602	4.745	92	61.677	4.912
13	39.005	3.131	33	19.131	3.848	53	44.982	4.072	73	26.413	4.722	93	68.590	4.856
14	43.045	3.079	34	21.384	3.825	54	50.341	4.020	74	29.589	4.695	94	21.851	5.465
15	11.812	3.693	35	23.921	3.798	55	55.817	3.965	75	33.113	4.665	95	24.200	5.448
16	13.074	3.680	36	26.727	3.769	56	16.217	4.563	76	36.203	4.638	96	26.998	5.427
17	14.638	3.663	37	29.178	3.743	57	18.174	4.546	77	40.763	4.599	97	30.235	5.403
18	16.186	3.646	38	32.780	3.704	58	20.114	4.529	78	44.493	4.567	98	33.895	5.376
19	18.022	3.626	39	35.712	3.673	59	22.422	4.509	79	49.971	4.519	99	37.966	5.346
20	20.137	3.603	40	39.993	3.626	60	25.087	4.485	80	56.009	4.465	100	41.541	5.319

Çizelge 7.1. On çubuklu kafes sistemin çözümünde kullanılan 200 adet boru kesite ait en kesit alanları ve atalet yarıçapları (Devam)

Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı	Sıra No	Enkesit Alanı	Atalet Yarıçapı
101	46.829	5.280	121	38.090	6.040	141	53.077	7.469	161	82.657	9.305	181	109.965	14.061
102	51.166	5.247	122	42.693	6.010	142	58.163	7.442	162	90.577	9.271	182	124.583	14.019
103	57.554	5.198	123	46.741	5.983	143	65.717	7.401	163	102.339	9.221	183	136.695	13.985
104	64.622	5.144	124	52.737	5.943	144	71.943	7.368	164	115.499	9.164	184	154.746	13.933
105	71.909	5.087	125	57.665	5.910	145	81.164	7.318	165	129.234	9.104	185	175.033	13.875
106	23.166	5.793	126	64.939	5.861	146	91.444	7.262	166	70.692	11.203	186	196.315	13.814
107	25.661	5.776	127	73.012	5.806	147	102.130	7.203	167	79.426	11.172	187	140.486	15.808
108	28.635	5.756	128	81.362	5.748	148	47.164	8.425	168	87.148	11.145	188	154.189	15.773
109	32.076	5.732	129	33.106	6.653	149	52.974	8.397	169	98.654	11.104	189	174.625	15.722
110	35.971	5.705	130	37.105	6.629	150	59.463	8.366	170	108.174	11.070	190	197.615	15.663
111	40.304	5.675	131	41.638	6.602	151	65.188	8.339	171	122.336	11.019	191	221.760	15.602
112	44.113	5.648	132	46.690	6.572	152	73.700	8.298	172	138.215	10.961	192	171.820	17.576
113	49.751	5.608	133	51.138	6.545	153	80.724	8.265	173	154.830	10.901	193	194.661	17.524
114	54.381	5.575	134	57.734	6.504	154	91.143	8.214	174	87.397	12.293	194	220.376	17.466
115	61.207	5.526	135	63.162	6.471	155	102.780	8.158	175	95.915	12.265	195	247.406	17.404
116	68.773	5.471	136	71.186	6.422	156	114.903	8.098	176	108.617	12.224	196	214.696	19.327
117	76.585	5.414	137	80.108	6.366	157	52.807	9.432	177	119.133	12.190	197	243.136	19.268
118	27.154	6.112	138	89.358	6.308	158	59.334	9.404	178	134.789	12.138	198	273.051	19.206
119	30.307	6.091	139	42.134	7.527	159	66.629	9.373	179	152.362	12.081	199	265.897	21.071
120	33.957	6.068	140	47.306	7.500	160	73.070	9.346	180	170.770	12.020	200	298.697	21.009

7.1.1.Ceza katsayısının araştırılması

Daha öncede bahsedildiği gibi problemi sınırlayıcısız hale getirmek için amaç fonksiyonu değerine ceza fonksiyonu, belli bir katsayı ile çarpılarak, eklenmektedir. Bu katsayı ceza katsayısı olarak adlandırılmakta olup, sınırlayıcıların amaç fonksiyonun hesaplanmasında ne derece etkili olacağını belirlemektedir. Bu nedenle optimum sonuca kısa sürede ulaşılabilmesi için ceza katsayısının etkili bir şekilde belirlenmesi önemlidir.

Burada ceza katsayısının optimum değerinin belirlenebilmesi için, ceza katsayısına ait yedi farklı değer (1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0) kullanılarak on çubuklu kafes sistemin optimizasyonu yapılmıştır. Her bir ceza katsayısı için 5 ile 5000 arasında değişen karınca sayılarına sahip kümeler kullanılarak 21 farklı optimizasyon gerçekleştirilmiştir. Ceza katsayısının belirlenmesi için yapılan optimizasyonlarda feromon güncelleme katsayısı 0.025 alınmıştır. Aşağıdaki çizelge ve şekillerde bu optimizasyon çalışmalarına ait sonuçlar sunulmaktadır (Çizelge 7.2, Çizelge 7.3, Çizelge 7.4, Çizelge 7.5, Çizelge 7.6, Çizelge 7.7, Çizelge 7.8).

Çizelge 7. 2. Ceza katsayısı 1.0 alındığında ulaşılan sonuçlar

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.5633	4.5633	417	3.254
2	10	-	4.3497	4.3497	293	3.709
3	20	-	4.2569	4.2569	281	6.160
4	30	-	4.0760	4.0760	320	9.373
5	40	0.002436	4.0882	4.0982	228	9.516
6	50	-	4.0250	4.0250	201	9.999
7	100	-	4.1926	4.1926	199	18.975
8	200	-	3.9461	3.9461	181	33.576
9	300	0.000517	4.0336	4.0357	197	54.069
10	400	-	3.9601	3.9601	208	75.627
11	500	0.000051	3.9828	3.9830	170	77.138
12	600	-	3.9131	3.9131	180	97.458
13	700	0.003889	3.9022	3.9174	178	112.151
14	800	-	3.8411	3.8411	204	146.546
15	900	-	3.8931	3.8931	185	150.629
16	1000	0.002379	3.8801	3.8893	142	128.422
17	1500	-	3.9005	3.9005	152	208.229
18	2000	0.001418	3.9225	3.9281	138	249.668
19	3000	-	3.8429	3.8429	157	430.649
20	4000	0.002537	3.8565	3.8663	137	494.791
21	5000	-	3.8673	3.8673	145	649.719

Çizelge 7.2’de ceza katsayısı 1.0 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısı arttıkça beklendiği gibi iterasyon sayısının azaldığı ancak çözüm süresinin arttığı görülmüştür. Karınca sayısının 40, 300, 500, 700, 1000, 2000 ve 4000 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. En iyi çözümler ise 800 karınca sayısı ile bulunan 3.8411 ton ve 3000 karınca sayısı ile bulunan 3.8429 ton değerleri olmuştur.

Çizelge 7. 3. Ceza katsayısının 1.5 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4695	4.4695	433	3.970
2	10	-	4.7848	4.7848	316	4.258
3	20	-	4.3835	4.3835	260	5.743
4	30	-	4.2655	4.2655	285	8.443
5	40	-	4.1486	4.1486	232	9.399
6	50	-	4.3284	4.3284	269	13.448
7	100	-	4.3656	4.3656	251	24.326
8	200	0.000260	3.9009	3.9024	233	43.963
9	300	-	3.9760	3.9760	212	58.738
10	400	-	3.9407	3.9407	182	68.271
11	500	-	4.0172	4.0172	194	89.243
12	600	-	4.0140	4.0140	147	81.241
13	700	-	3.9489	3.9489	168	105.175
14	800	0.000662	3.9061	3.9100	172	125.299
15	900	-	3.8950	3.8950	151	124.010
16	1000	-	3.9051	3.9051	147	134.132
17	1500	0.001085	4.0777	4.0843	128	176.249
18	2000	-	3.8593	3.8593	129	235.298
19	3000	-	3.8520	3.8520	151	426.396
20	4000	0.002083	3.8679	3.8800	146	524.541
21	5000	-	3.9753	3.9753	174	781.646

Çizelge 7.3'te ceza katsayısının 1.5 alındığında elde edilen çözüm sonuçları bulunmaktadır. Yine karınca sayısı arttıkça iterasyon sayısının azaldığı ancak çözüm süresinin arttığı görülmüştür. Karınca sayısının 200, 800, 1500 ve 4000 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. En iyi çözümler ise 3000 karınca sayısı ile bulunan 3.8520 ton ve 2000 karınca sayısı ile bulunan 3.8593 ton değerleri olmuştur.

Çizelge 7. 4. Ceza katsayısının 2.0 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.3829	4.3829	487	3.758
2	10	0.010377	4.5730	4.6679	404	5.035
3	20	0.000712	4.1708	4.1767	296	6.981
4	30	-	4.5382	4.5382	214	7.627
5	40	-	4.5052	4.5052	206	8.606
6	50	-	4.2383	4.2383	216	10.936
7	100	-	3.9588	3.9588	246	23.383
8	200	-	4.0756	4.0756	191	35.775
9	300	0.001207	3.9651	3.9747	211	58.531
10	400	0.011522	4.0123	4.1048	154	56.365
11	500	-	3.9903	3.9903	189	86.848
12	600	-	3.8969	3.8969	173	94.868
13	700	0.000190	3.9567	3.9582	140	89.620
14	800	-	3.8753	3.8753	192	140.807
15	900	-	3.9282	3.9282	192	157.336
16	1000	0.000099	3.8410	3.8418	178	161.012
17	1500	-	3.9582	3.9582	143	196.253
18	2000	0.001287	3.9078	3.9179	145	262.501
19	3000	0.001338	3.8887	3.8991	101	310.693
20	4000	-	3.9122	3.9122	111	402.206
21	5000	-	3.8997	3.8997	136	612.606

Çizelge 7.4'te ceza katsayısı 2.0 alındığında elde edilen çözüm sonuçları bulunmaktadır. Yine karınca sayısı arttıkça iterasyon sayısı azalır, çözüm süresi ise artmıştır. Karınca sayısının 10, 20, 300, 400, 700, 1000, 2000 ve 3000 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. Ancak 1000 karınca ile bulunan 3.8418 ton değeri sınırlayıcı ihlali yapsa da, ihlal değeri çok küçük olduğu için en iyi cezalandırılmış amaç fonksiyonu değeri olmuştur.

Çizelge 7. 5. Ceza katsayısının 2.5 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4354	4.4354	454	3.617
2	10	-	4.7962	4.7962	326	4.144
3	20	-	4.2705	4.2705	325	6.860
4	30	-	4.6993	4.6993	242	7.784
5	40	0.004745	4.0434	4.0914	269	10.938
6	50	-	4.1441	4.1441	286	12.077
7	100	-	4.1803	4.1803	253	24.255
8	200	-	4.1874	4.1874	181	34.058
9	300	-	3.9691	3.9691	198	54.343
10	400	-	4.0244	4.0244	183	67.624
11	500	0.000660	3.9793	3.9859	186	85.439
12	600	-	3.8717	3.8717	219	119.370
13	700	-	3.9024	3.9024	166	107.540
14	800	-	3.9511	3.9511	158	115.964
15	900	-	3.9025	3.9025	172	141.686
16	1000	0.000150	3.8894	3.8909	173	157.694
17	1500	-	3.8850	3.8850	171	230.833
18	2000	-	3.8533	3.8533	149	255.612
19	3000	-	3.8476	3.8476	159	431.536
20	4000	-	3.8835	3.8835	140	507.848
21	5000	-	3.8713	3.8713	156	709.889

Çizelge 7.5'te ceza katsayısının 2.5 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısı artıkça yine beklendiği gibi iterasyon sayısının azaldığı ancak çözüm süresinin arttığı görülmüştür. Karınca sayısının 40, 500, ve 1000 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. En iyi çözümler ise 3000 karınca sayısı ile bulunan 3.8476 ton ve 2000 karınca sayısı ile bulunan 3.8533 ton değerleri olmuştur.

Çizelge 7. 6. Ceza katsayısının 3.0 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4064	4.4064	373	2.852
2	10	-	4.3759	4.3759	388	4.750
3	20	-	4.8253	4.8253	247	5.522
4	30	-	4.3082	4.3082	276	8.291
5	40	-	4.2602	4.2602	240	9.106
6	50	-	4.2508	4.2508	223	10.601
7	100	-	4.4447	4.4447	242	22.764
8	200	-	4.1283	4.1283	152	28.380
9	300	-	3.9691	3.9691	207	56.679
10	400	0.001269	4.0134	4.0287	191	69.291
11	500	-	3.9623	3.9623	196	89.289
12	600	0.002846	4.0873	4.1222	173	95.069
13	700	-	3.9654	3.9654	168	106.722
14	800	-	3.9888	3.9888	150	110.030
15	900	-	3.9596	3.9596	164	136.195
16	1000	-	3.8582	3.8582	119	105.874
17	1500	-	3.8338	3.8338	167	226.030
18	2000	-	3.9081	3.9081	167	301.290
19	3000	-	3.8357	3.8357	150	423.045
20	4000	-	3.8579	3.8579	151	546.623
21	5000	-	3.8865	3.8865	132	596.678

Çizelge 7.6'da ceza katsayısının 3.0 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısı artıkça yine iterasyon sayısının azaldığı ancak çözüm süresinin arttığı görülmektedir. Karınca sayısının 400 ve 600 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. En iyi çözümler ise 1500 karınca sayısı ile bulunan 3.8338 ton ve 3000 karınca sayısı ile bulunan 3.8357 ton değerleri olmuştur.

Çizelge 7. 7. Ceza katsayısının 3.5 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4941	4.4941	485	3.694
2	10	-	4.4118	4.4118	372	4.588
3	20	-	4.6202	4.6202	364	7.900
4	30	-	4.5765	4.5765	278	8.222
5	40	-	4.6195	4.6195	214	9.133
6	50	-	4.3584	4.3584	240	11.887
7	100	-	4.3661	4.3661	252	23.932
8	200	-	4.1283	4.1283	152	28.697
9	300	-	3.9417	3.9417	237	64.970
10	400	0.000604	4.2618	4.2708	179	65.715
11	500	-	3.9966	3.9966	187	91.741
12	600	-	4.0305	4.0305	164	94.121
13	700	0.000544	3.8867	3.8941	176	117.277
14	800	-	3.9046	3.9046	167	126.735
15	900	-	3.9596	3.9596	165	139.365
16	1000	-	3.9190	3.9190	152	139.908
17	1500	-	3.8550	3.8550	177	248.859
18	2000	-	3.8554	3.8554	189	342.274
19	3000	0.000293	3.9472	3.9512	109	313.769
20	4000	-	3.8573	3.8573	152	547.284
21	5000	-	3.8778	3.8778	141	633.406

Çizelge 7.7’de ceza katsayısının 3.5 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısı artıkça yine iterasyon sayısı azalır, çözüm süresi artmaktadır. Karınca sayısının 400, 700 ve 3000 alındığı çözüm sonuçları sınırlayıcılarımızı ihlal etmiştir. En iyi çözümler ise 1500 karınca sayısı ile bulunan 3.8550 ton ve 2000 karınca sayısı ile bulunan 3.8554 ton değerleri olmuştur.

Çizelge 7. 8. Ceza katsayısının 4.0 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4941	4.4941	485	3.614
2	10	-	4.2577	4.2577	379	4.650
3	20	0.000524	4.1599	4.1686	363	7.880
4	30	0.000245	4.4502	4.4546	298	9.287
5	40	-	4.1470	4.1470	260	9.973
6	50	-	4.2717	4.2717	240	12.456
7	100	-	4.3335	4.3335	299	28.077
8	200	-	3.8950	3.8950	176	32.924
9	300	-	4.0309	4.0309	186	51.281
10	400	-	3.9333	3.9333	175	64.592
11	500	-	3.9966	3.9966	187	85.583
12	600	-	3.9111	3.9111	180	98.825
13	700	-	3.9763	3.9763	168	107.862
14	800	-	3.9354	3.9354	181	132.422
15	900	-	3.9603	3.9603	167	136.641
16	1000	-	3.9190	3.9190	152	138.367
17	1500	-	3.9388	3.9388	164	223.660
18	2000	-	3.8893	3.8893	147	268.691
19	3000	0.000186	3.8473	3.8502	164	457.028
20	4000	-	3.8457	3.8457	128	460.035
21	5000	-	3.8687	3.8687	155	695.654

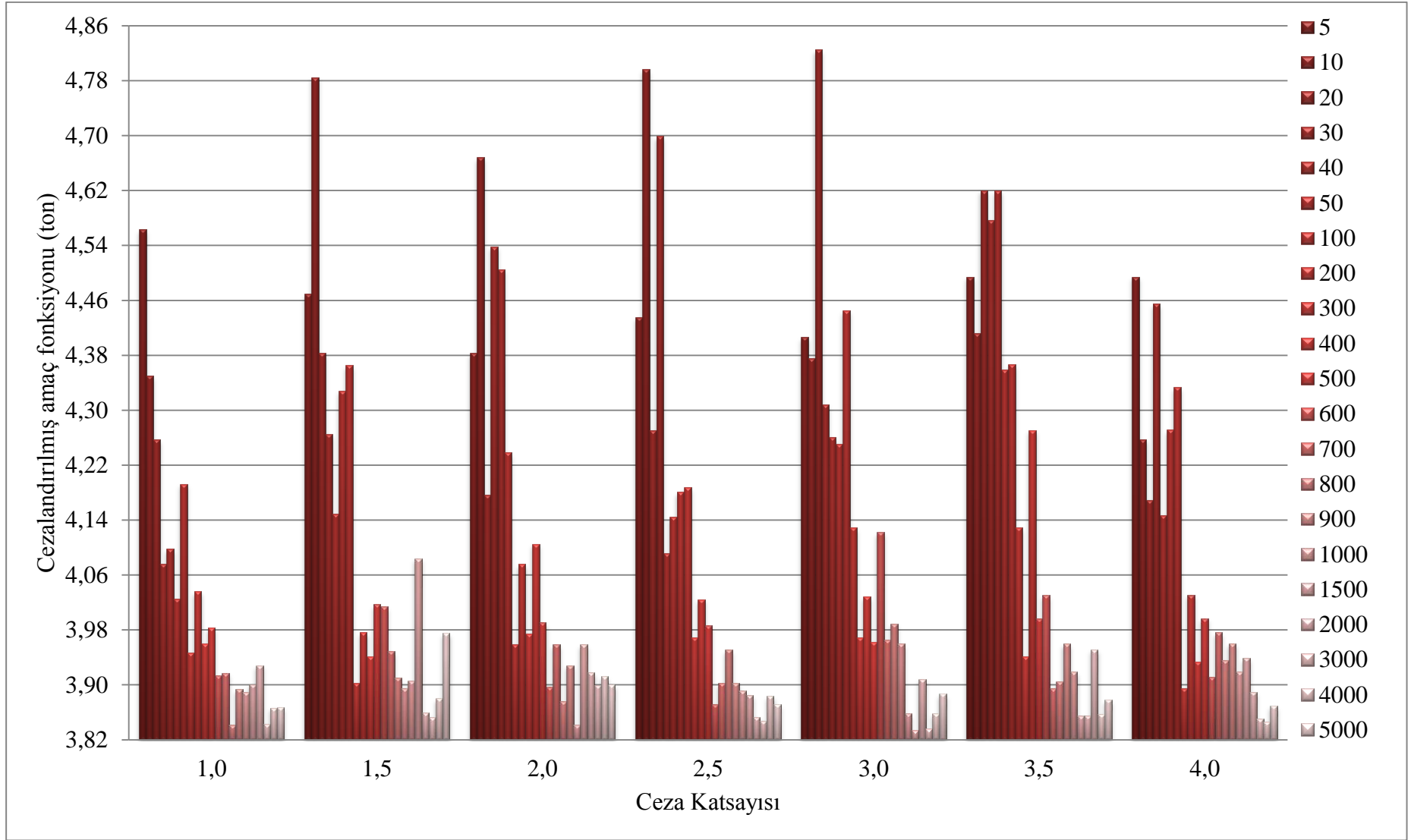
Çizelge 7.8’de ceza katsayısının 4.0 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısı artıkça yine iterasyon sayısı azalır, çözüm süresi artmaktadır. Karınca sayısının 20, 30 ve 3000 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözümler ise 4000 karınca sayısı ile bulunan 3.8457 ton ve sınırlayıcı ihlali yapmasına rağmen 3000 karınca sayısı ile bulunan 3.8554 ton cezalandırılmış amaç fonksiyonu değerleri olmuştur.

Şekil 7.2’de ceza katsayısının ve karınca sayısının değişimine göre cezalandırılmış amaç fonksiyonu değerinin değişimi farklı ceza katsayıları için aynı grafik üzerinde gösterilmiştir. Karınca sayısı artıkça sürekli olmamakla birlikte cezalandırılmış amaç

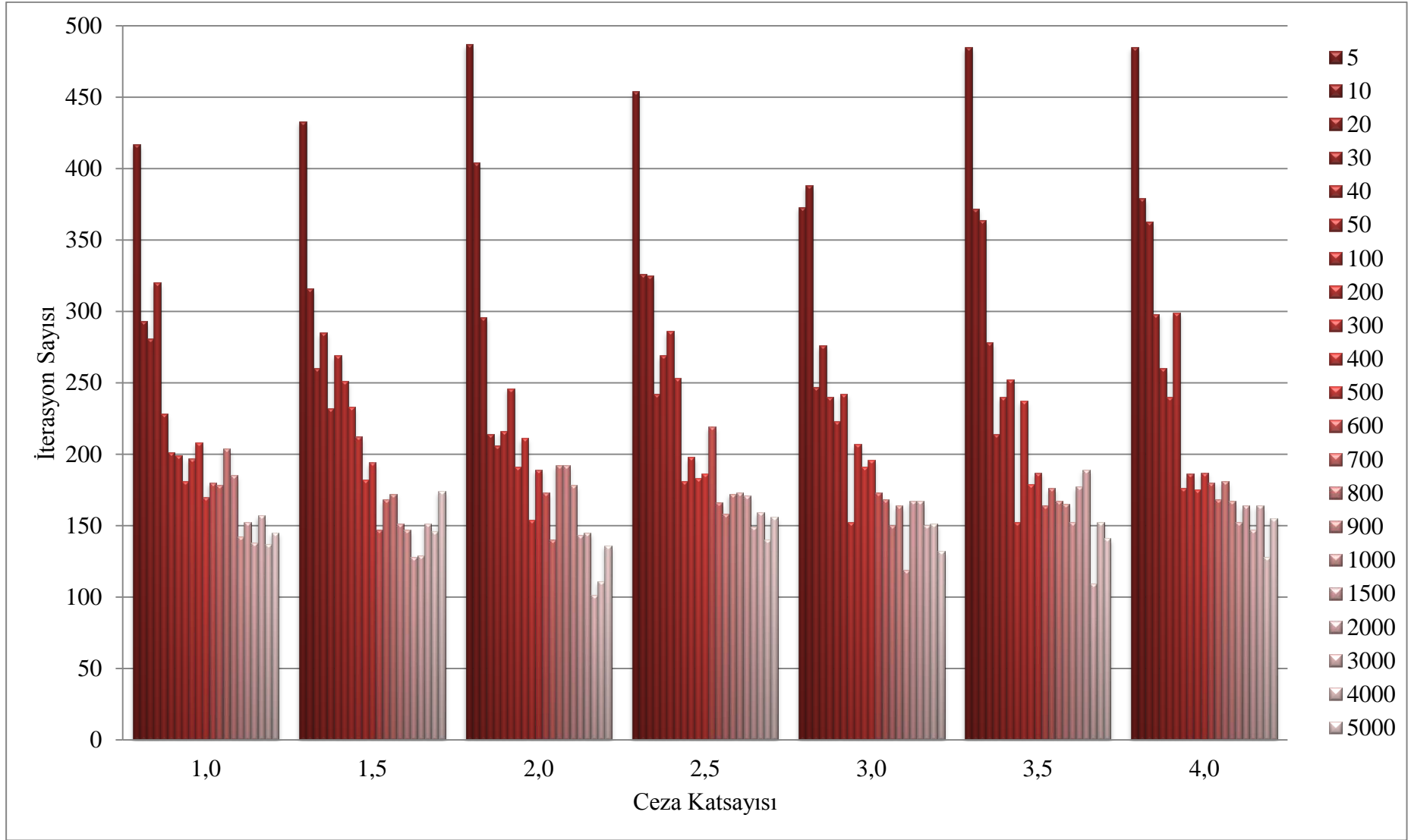
fonksiyonu deęerinin genellikle azalmakta olduęu grlmektedir. Her ceza katsayısı deęeri iin en iyi cezalandırılmıř ama fonksiyonu deęeri farklı karınca sayısında bulunmuřtur.

řekil 7.3'te ceza katsayısının ve karınca sayısının deęişimine gre iterasyon sayısının deęişimi gsterilmiřtir. Karınca sayısı artıka iterasyon sayısında genel bir azalma olmaktadır.

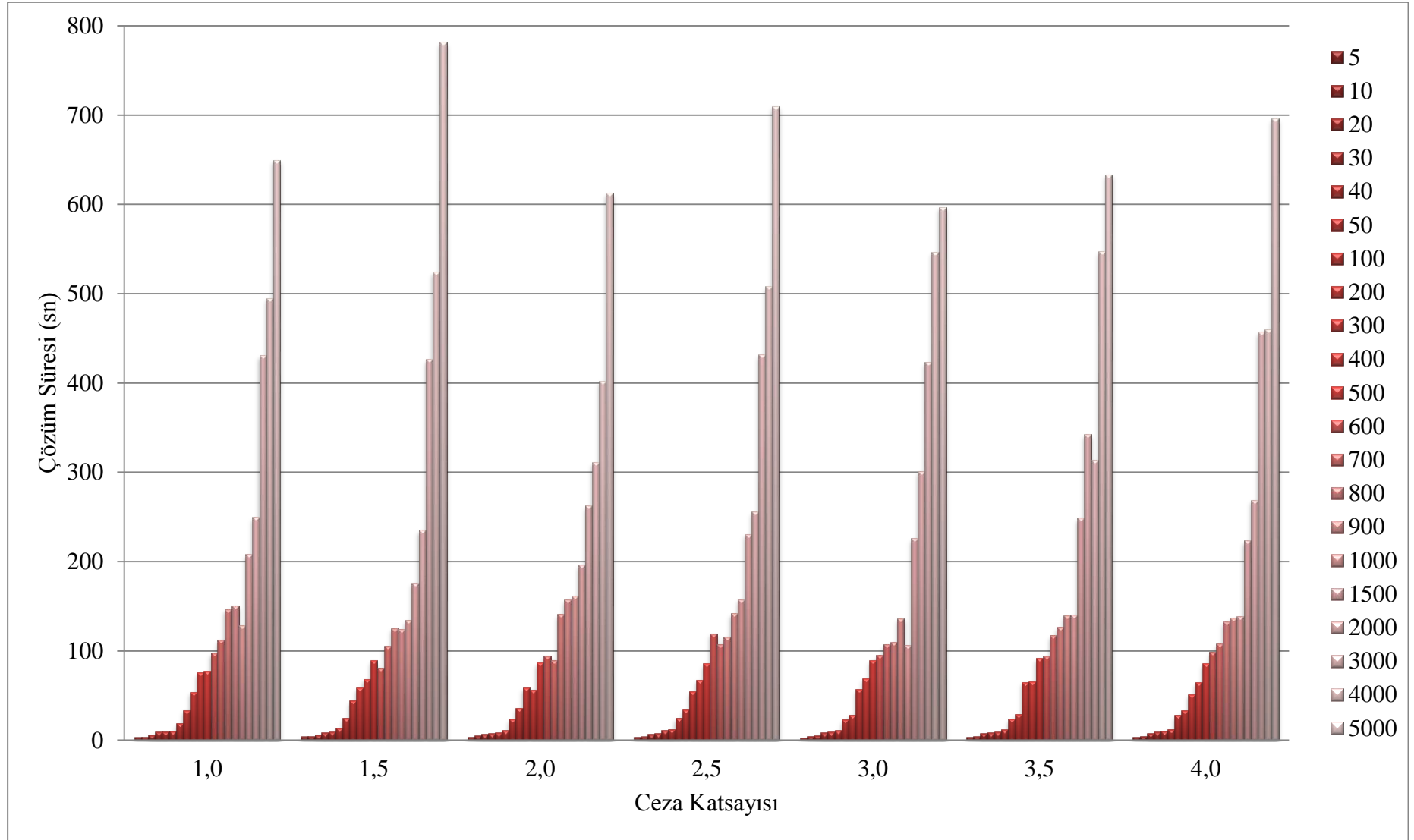
řekil 7.4'te ise ceza katsayısının ve karınca sayısının deęişimine gre zm sresinin deęişimi gsterilmiřtir. Yine karınca sayısı arttıka zm sresinin de arttıęı grlmektedir.



Şekil 7. 2. Karınca sayısı ve ceza katsayısı değişimine göre cezalandırılmış amaç fonksiyonu

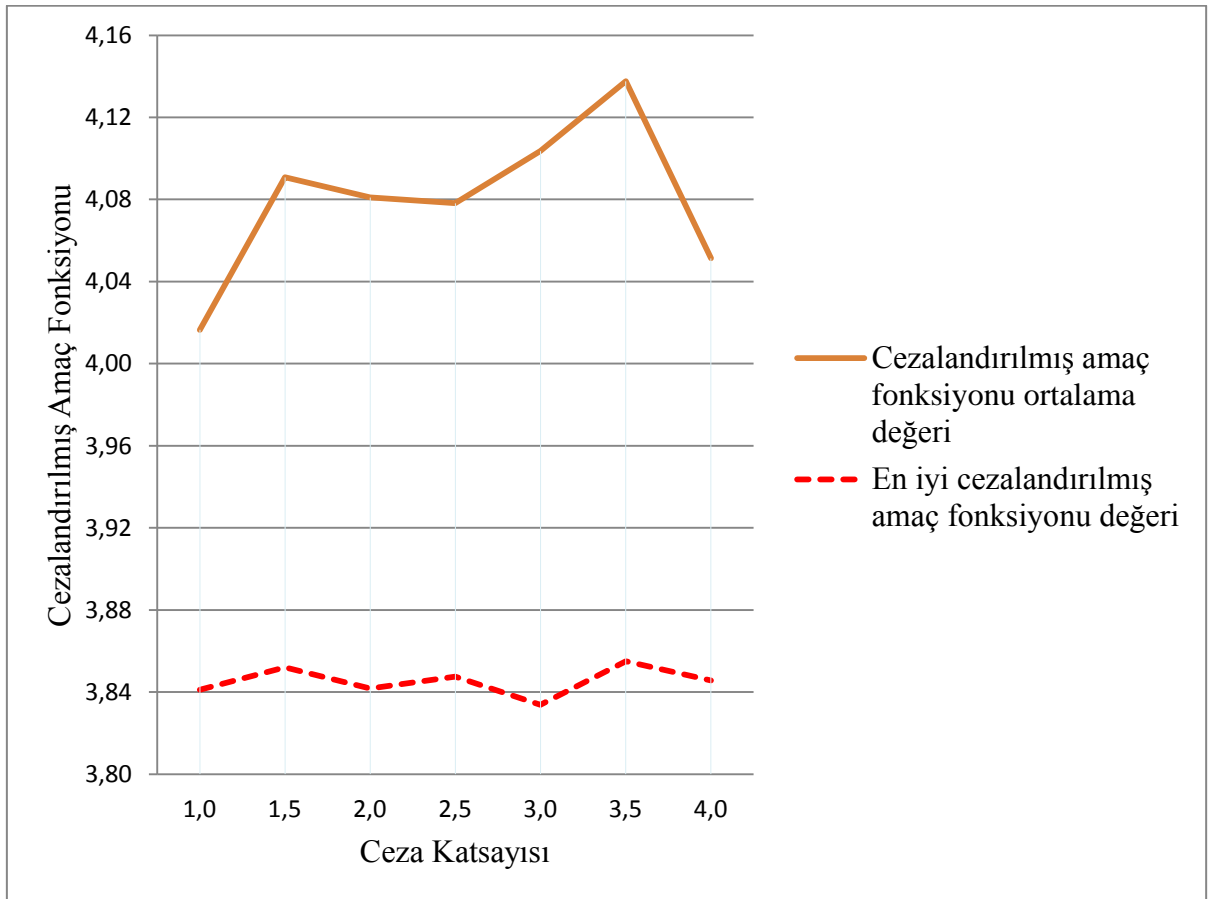


Şekil 7. 3. Karınca sayısı ve ceza katsayısı değişimine göre iterasyon sayısı



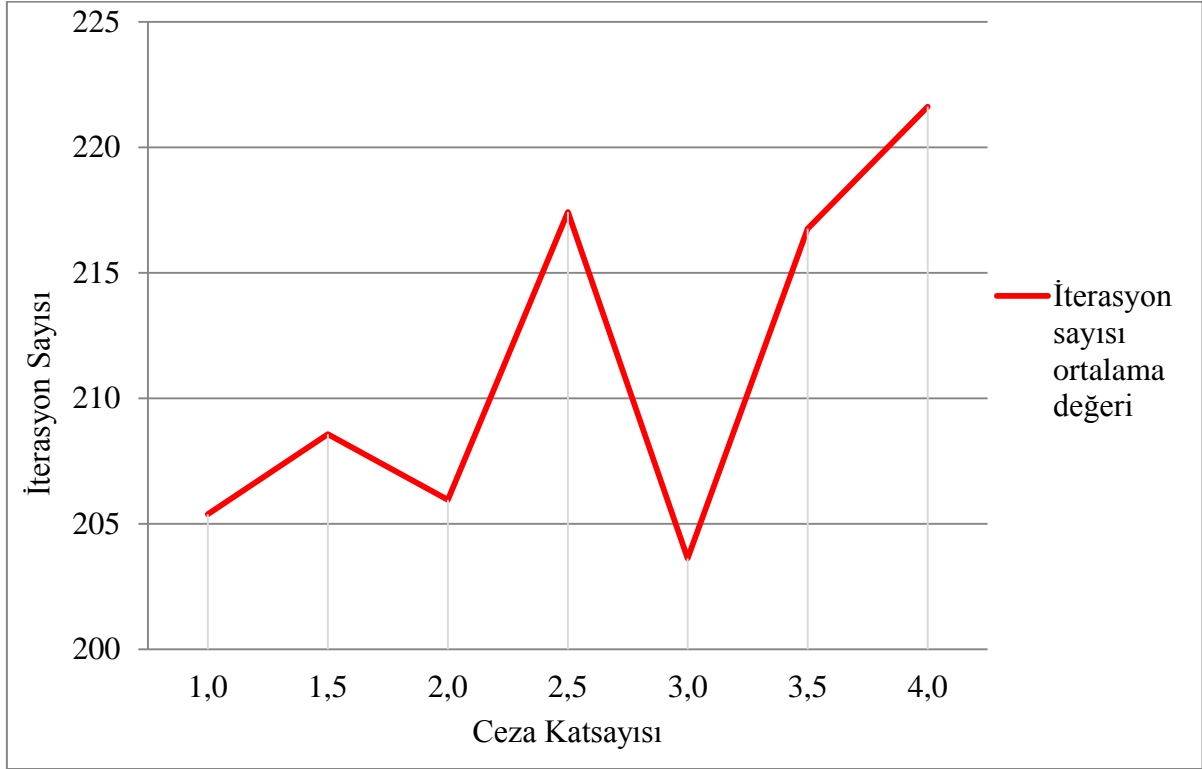
Şekil 7. 4. Karınca sayısı ve ceza katsayısı değişimine göre çözüm süresi

Şekil 7.5'te ceza katsayıları değişimi ile bulunan en iyi cezalandırılmış amaç fonksiyonu ve cezalandırılmış amaç fonksiyonu ortalama değerleri verilmiştir. Daha önce bahsedildiği gibi her ceza katsayısı değerinde bulunan en iyi çözümler farklı karınca değerleri ile bulunmuştur. Bu çizelgeye bakıldığında ceza katsayısının artışı veya azalışı yönünde iyi çözüme giden bir eğilim olmamaktadır. Ortalama değerler için her ne kadar en iyi ceza katsayısı 1.0 olarak görünse de, optimizasyonda amaç en iyi çözümü bulmak olduğundan bu sonuca ceza katsayısı 3.0 alınarak ulaşıldığı en iyi değerler eğrisinden görülmektedir. Bu doğrultuda bir sonraki bölümde feromon güncelleme katsayısı araştırılırken ceza katsayısı tüm çözümlerde 3.0 alınacaktır.



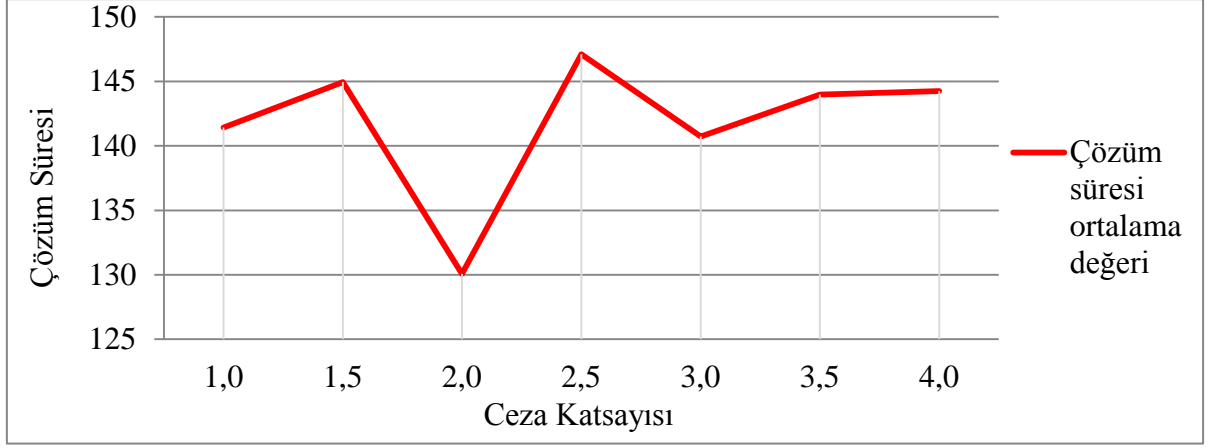
Şekil 7. 5. Ceza katsayısı değişimine göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri

Şekil 7.6'da ceza katsayısının değişimine göre iterasyon sayısı ortalaması verilmiştir. Bu çizelgede ceza katsayı arttıkça iterasyon sayısı ortalaması lineer olmasa da artma eğilimi göstermektedir. Ancak en az iterasyon sayısı ortalama değeri ceza katsayısı 3.0 alındığında görülmüştür.



Şekil 7. 6. Ceza katsayısı değişimine göre iterasyon sayısı ortalama değeri

Şekil 7.7'de ceza katsayısının değişimine göre çözüm süresi ortalaması verilmiştir. Bu çizelgeden ceza katsayısına bağlı olarak çözüm süresinin % 13 oranında değişebildiği görülmektedir. Buradan hareketle de ceza katsayısının belli sınırlar içinde seçildiği takdirde çözüm süresini çok fazla etkilemediği söylenebilir.



Şekil 7. 7. Ceza katsayısı değişimine göre çözüm süresi ortalama değeri

7.1.2. Feromon güncelleme katsayısının araştırılması

Daha öncede bahsedildiği gibi karıncaların, tasarım değişkenlerini seçme olasılığını etkileyen feromon miktarının belirlenmesi için seçilen katsayıya feromon güncelleme katsayısı denilmektedir. Feromon güncelleme katsayısı seçildikten sonra feromon miktarını hesaplayan bağıntı ile bu değer bulunup en iyi karıncanın tercih ettiği yola eklenmekte ve bu yolun bir sonraki iterasyonda seçilme olasılığı artmaktadır. Diğer yolların seçilme olasılığı ise en iyi yola eklenen miktar kadar azalmaktadır. Sonuç olarak toplam feromon miktarı hiçbir zaman değişmeyip, tüm yolların tercih edilme olasılığına göre paylaştırılmaktadır.

On çubuklu kafes sistemin optimum tasarım problemi için, uygun feromon güncelleme katsayısı belirlenirken yedi farklı değer kullanılmıştır (0.010, 0.015, 0.020, 0.025, 0.030, 0.035, 0.040).

Her bir ceza katsayısı için 5 ile 5000 arasında değişen karınca sayılarına sahip kümeler kullanılarak 21 farklı optimizasyon gerçekleştirilmiştir. Feromon güncelleme katsayısının belirlenmesi için yapılan optimizasyonlarda ceza katsayısı 3.0 alınmıştır. Aşağıdaki çizelge ve şekillerde bu optimizasyon çalışmalarına ait sonuçlar sunulmaktadır (Çizelge 7.9, Çizelge 7.10, Çizelge 7.11, Çizelge 7.12, Çizelge 7.13, Çizelge 7.14, Çizelge 7.15).

Çizelge 7. 9. Feromon güncelleme katsayısının 0.010 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.1633	4.1633	1465	10.047
2	10	-	4.2882	4.2882	1023	12.212
3	20	-	4.1510	4.1510	881	18.675
4	30	-	4.0751	4.0751	823	24.592
5	40	-	3.9931	3.9931	692	27.499
6	50	-	3.8839	3.8839	730	35.068
7	100	-	3.9803	3.9803	644	60.208
8	200	-	3.9074	3.9074	631	116.195
9	300	-	3.8922	3.8922	576	159.570
10	400	-	3.8913	3.8913	488	178.493
11	500	-	3.8395	3.8395	422	192.448
12	600	-	3.8935	3.8935	490	269.631
13	700	-	3.8419	3.8419	554	353.054
14	800	-	3.8387	3.8387	495	364.351
15	900	-	3.8315	3.8315	463	377.996
16	1000	-	3.8717	3.8717	545	492.256
17	1500	-	3.7964	3.7964	416	565.259
18	2000	-	3.8785	3.8785	393	712.451
19	3000	-	3.8389	3.8389	428	1164.370
20	4000	-	3.9395	3.9395	391	1409.272
21	5000	-	3.8400	3.8400	353	1584.273

Çizelge 7.9'da feromon güncelleme katsayısının 0.010 alındığında elde edilen çözüm sonuçları bulunmaktadır. Hiçbir çözümde sınırlayıcılar ihlal edilmemiştir. En iyi çözümler ise 1500 karınca sayısı ile 565.259 saniyede bulunan 3.7964 ton, 900 karınca sayısı ile 377.996 saniyede bulunan 3.8315 ton değerleri olmuştur. İki sonuç arasında 0.0351 ton gibi az bir fark varken çözüm süresi ise % 49 daha fazladır. İyi sonuçlardan bir tanesi olan, 500 karınca ile 192.448 saniyede bulunan 3.8395 ton değeri en iyi çözümden 0.0431 ton fazladır. Ancak en iyi sonucun çözüm süresi % 193 daha fazladır.

Çizelge 7. 10. Feromon güncelleme katsayısının 0.015 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.7665	4.7665	657	4.873
2	10	-	4.1526	4.1526	773	9.338
3	20	-	3.9883	3.9883	601	12.402
4	30	-	4.1152	4.1152	553	16.685
5	40	-	3.9178	3.9178	461	17.688
6	50	0.004503	3.9131	3.9660	394	19.326
7	100	-	3.8970	3.8970	418	39.302
8	200	0.000052	3.9542	3.9548	324	59.741
9	300	-	4.0564	4.0564	349	96.143
10	400	-	3.9734	3.9734	324	119.339
11	500	0.009570	4.0183	4.1337	427	195.870
12	600	-	3.8552	3.8552	356	195.491
13	700	-	3.8662	3.8662	324	204.639
14	800	-	3.8568	3.8568	267	195.745
15	900	-	3.8773	3.8773	335	271.892
16	1000	-	3.8343	3.8343	292	263.880
17	1500	-	3.7986	3.7986	257	348.026
18	2000	-	3.8819	3.8819	239	429.832
19	3000	0.000565	3.8513	3.8578	286	777.695
20	4000	-	3.8419	3.8419	252	908.840
21	5000	-	3.8399	3.8399	265	1193.295

Çizelge 7.10’da feromon güncelleme katsayısının 0.015 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısının 50, 200, 500 ve 3000 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözüm 1500 karınca sayısı ile 348.026 saniyede bulunan 3.7986 ton değeri olmuştur. Kabul edilebilir çözümlerden bir tanesi de 600 karınca ile 195.491 saniyede bulunan 3.8552 ton değeridir. İki sonuç arasında 0.0566 ton gibi küçük bir fark varken çözüm süresi ise % 78 daha fazladır.

Çizelge 7. 11. Feromon güncelleme katsayısının 0.020 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	0.000262	4.1389	4.1422	722	5.283
2	10	-	4.2540	4.2540	525	6.481
3	20	-	4.3722	4.3722	401	8.052
4	30	-	4.2692	4.2692	327	10.027
5	40	-	4.1912	4.1912	385	15.364
6	50	-	4.0584	4.0584	265	14.100
7	100	-	4.1903	4.1903	227	21.437
8	200	0.000417	3.9528	3.9577	248	45.641
9	300	-	3.9880	3.9880	298	81.287
10	400	0.000049	3.9458	3.9464	281	102.358
11	500	-	4.0594	4.0594	179	81.567
12	600	-	3.9571	3.9571	218	118.651
13	700	0.019294	3.9142	4.1408	259	164.101
14	800	0.001086	4.0081	4.0212	187	137.158
15	900	-	3.9926	3.9926	193	157.446
16	1000	-	3.8515	3.8515	185	167.023
17	1500	-	3.8582	3.8582	220	296.774
18	2000	-	3.9287	3.9287	174	315.040
19	3000	-	3.8637	3.8637	216	581.324
20	4000	-	3.9486	3.9486	202	730.089
21	5000	-	3.8399	3.8399	212	963.352

Çizelge 7.11’de feromon güncelleme katsayısının 0.020 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısının 5, 200, 400, 700 ve 800 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözüm 5000 karınca sayısı ile 963.352 saniyede bulunan 3.8399 ton değeri olmuştur. İyi sonuç veren çözümlerden bir tanesi de 1000 karınca ile 167.023 saniyede bulunan 3.8515 ton değeridir. İki sonuç arasında 0.0116 ton gibi küçük bir fark varken çözüm süresi ise % 477 daha fazladır. Bu sonuçlarda ağırlık açısından çok fazla bir fark olmamasına rağmen, çözüm süresinin çok fazla arttığı görülmektedir ve en iyi çözüm yerine, daha kısa sürede en iyiye yakın bulunan çözüm kabul edilebilir sonuç olarak değerlendirilebilir.

Çizelge 7. 12. Feromon güncelleme katsayısının 0.025 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.4064	4.4064	373	2.852
2	10	-	4.3759	4.3759	388	4.750
3	20	-	4.8253	4.8253	247	5.522
4	30	-	4.3082	4.3082	276	8.291
5	40	-	4.2602	4.2602	240	9.106
6	50	-	4.2508	4.2508	223	10.601
7	100	-	4.4447	4.4447	242	22.764
8	200	-	4.1283	4.1283	152	28.380
9	300	-	3.9691	3.9691	207	56.679
10	400	0.001269	4.0134	4.0287	191	69.291
11	500	-	3.9623	3.9623	196	89.289
12	600	0.002846	4.0873	4.1222	173	95.069
13	700	-	3.9654	3.9654	168	106.722
14	800	-	3.9888	3.9888	150	110.030
15	900	-	3.9596	3.9596	164	136.195
16	1000	-	3.8582	3.8582	119	105.874
17	1500	-	3.8338	3.8338	167	226.030
18	2000	-	3.9081	3.9081	167	301.290
19	3000	-	3.8357	3.8357	150	423.045
20	4000	-	3.8579	3.8579	151	546.623
21	5000	-	3.8865	3.8865	132	596.678

Çizelge 7.12’de feromon güncelleme katsayısının 0.025 alındığındaki çözüm sonuçları bulunmaktadır. Karınca sayısının 400 ve 600 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözüm 1500 karınca sayısı ile 226.030 saniyede bulunan 3.8338 ton değeri olmuştur.

Çizelge 7. 13. Feromon güncelleme katsayısının 0.030 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	5.2134	5.2134	349	3.279
2	10	-	5.1597	5.1597	271	3.577
3	20	-	4.3666	4.3666	283	6.311
4	30	-	4.4121	4.4121	186	5.915
5	40	-	4.2856	4.2856	218	8.397
6	50	-	4.2576	4.2576	202	10.271
7	100	-	4.1379	4.1379	174	17.363
8	200	-	3.9840	3.9840	164	30.639
9	300	-	3.9826	3.9826	159	44.418
10	400	-	3.9369	3.9369	135	49.936
11	500	0.000037	3.9526	3.9530	158	72.693
12	600	0.000062	3.9931	3.9938	135	73.509
13	700	-	4.0589	4.0589	142	91.161
14	800	0.001483	3.9945	4.0123	148	108.128
15	900	-	4.0928	4.0928	151	123.819
16	1000	-	3.9577	3.9577	148	133.518
17	1500	-	3.9413	3.9413	131	179.517
18	2000	-	3.8722	3.8722	146	264.619
19	3000	-	3.8447	3.8447	163	440.809
20	4000	0.000389	3.9452	3.9498	140	504.678
21	5000	-	3.8957	3.8957	104	470.588

Çizelge 7.13'te feromon güncelleme katsayısının 0.030 alındığında elde edilen çözüm sonuçları bulunmaktadır. Karınca sayısının 500, 600, 800 ve 4000 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözüm 3000 karınca sayısı ile 440.809 saniyede bulunan 3.8447 ton değeri olmuştur.

Çizelge 7. 14. Feromon güncelleme katsayısının 0.035 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	5.0528	5.0528	283	2.482
2	10	-	4.5093	4.5093	249	3.316
3	20	-	4.4479	4.4479	204	4.611
4	30	-	4.7680	4.7680	197	6.422
5	40	-	4.5837	4.5837	157	6.354
6	50	-	4.5363	4.5363	175	8.865
7	100	-	4.0826	4.0826	127	12.188
8	200	-	4.1303	4.1303	138	25.750
9	300	-	3.9943	3.9943	124	34.992
10	400	-	4.0823	4.0823	114	42.071
11	500	-	4.2577	4.2577	124	56.837
12	600	-	3.8940	3.8940	114	62.717
13	700	-	3.9541	3.9541	136	86.978
14	800	-	3.9665	3.9665	113	82.183
15	900	0.000007	3.9995	3.9996	114	93.376
16	1000	-	4.0480	4.0480	130	118.336
17	1500	-	3.8761	3.8761	112	251.803
18	2000	-	3.9574	3.9574	125	225.374
19	3000	-	4.0008	4.0008	95	256.569
20	4000	-	3.9596	3.9596	94	337.355
21	5000	-	3.9307	3.9307	85	382.709

Çizelge 7.14'te feromon güncelleme katsayısının 0.035 alındığındaki çözüm sonuçları bulunmaktadır. Karınca sayısının 900 alındığı çözüm sonucu sınırlayıcıları ihlal etmiştir. En iyi çözüm 1500 karınca sayısı ile 251.803 saniyede bulunan 3.8761 ton değeridir.

Çizelge 7. 15. Feromon güncelleme katsayısının 0.040 alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.7960	4.7960	265	2.248
2	10	-	5.9030	5.9030	156	2.312
3	20	-	4.7460	4.7460	172	3.942
4	30	-	5.3637	5.3637	146	4.885
5	40	-	4.7729	4.7729	109	4.756
6	50	-	4.2537	4.2537	169	8.587
7	100	-	4.0831	4.0831	136	13.071
8	200	-	4.0313	4.0313	123	22.320
9	300	0.000254	4.1033	4.1064	117	32.250
10	400	-	4.1442	4.1442	97	35.794
11	500	-	4.0082	4.0082	89	41.423
12	600	-	3.9602	3.9602	73	40.258
13	700	-	3.9622	3.9622	101	64.360
14	800	-	4.1752	4.1752	92	66.765
15	900	-	3.9108	3.9108	110	89.532
16	1000	-	4.0300	4.0300	83	76.180
17	1500	-	4.1344	4.1344	106	144.504
18	2000	-	3.8493	3.8493	106	191.184
19	3000	0.000005	4.0876	4.0877	79	214.850
20	4000	0.011674	3.8614	3.9966	117	426.512
21	5000	-	3.8928	3.8928	90	402.767

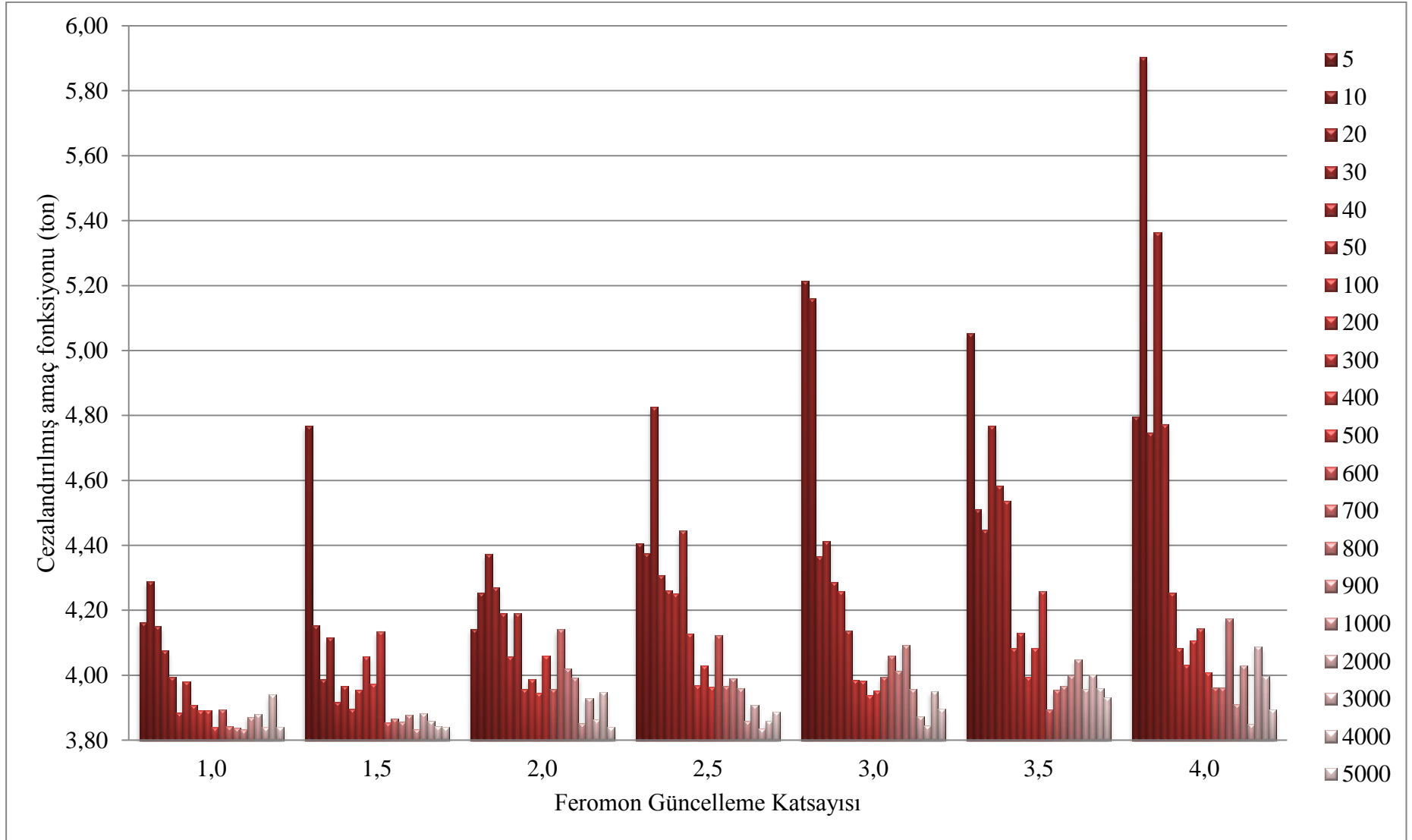
Çizelge 7.15'te feromon güncelleme katsayısının 0.040 alındığındaki çözüm sonuçları bulunmaktadır. Karınca sayısının 300, 3000 ve 4000 alındığı çözüm sonuçları sınırlayıcıları ihlal etmiştir. En iyi çözüm 2000 karınca sayısı ile 191.184 saniyede bulunan 3.8493 ton değeri olmuştur.

Şekil 7.8'de karınca sayısı ve feromon güncelleme katsayısı değişimine göre cezalandırılmış amaç fonksiyonu değişimi gösterilmiştir. Yine karınca sayısının artmasıyla beraber cezalandırılmış amaç fonksiyonu değerinde bir iyileşme olduğu görülmektedir. Ancak karınca sayısı arttıkça cezalandırılmış amaç fonksiyonu değeri kesinlikle azalır denemez.

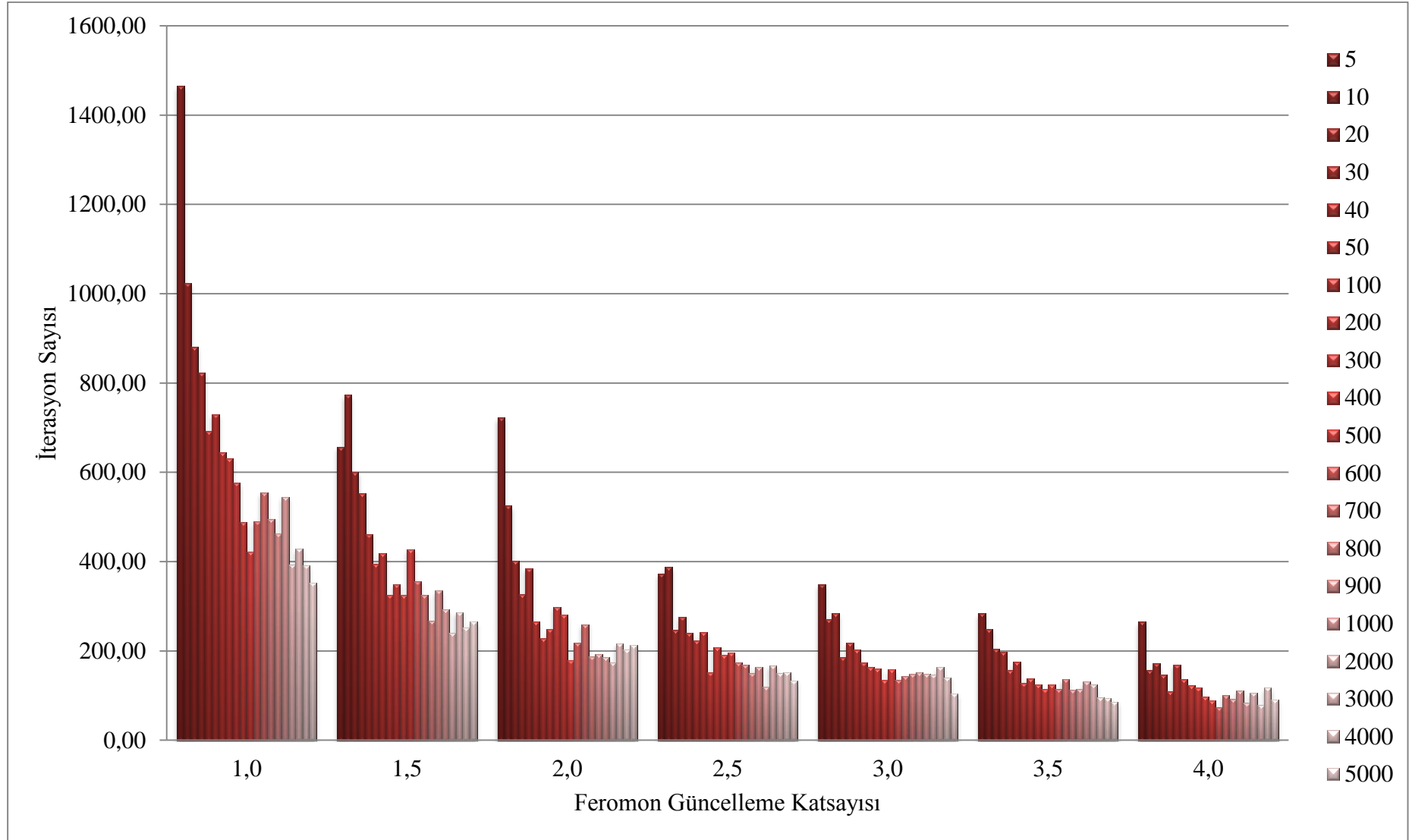
Şekil 7.11’de daha açıkça görüldüğü üzere, en optimum cezalandırılmış amaç fonksiyonu değeri, feromon güncelleme katsayısının 0.010 alındığı çözümler arasındadır. Genel olarak, feromon güncelleme katsayısı azaldıkça cezalandırılmış amaç fonksiyonu değeri daha optimum çözüme ulaşmaktadır.

Şekil 7.9’da karınca sayısı ve feromon güncelleme katsayısı değişimine iterasyon sayısının değişimi gösterilmiştir. Karınca sayısı artıkça iterasyon sayısının azalmaktadır. Açıkça görüldüğü gibi feromon güncelleme katsayısının artmasıyla da iterasyon sayılarının ortalamasında da azalma olmaktadır.

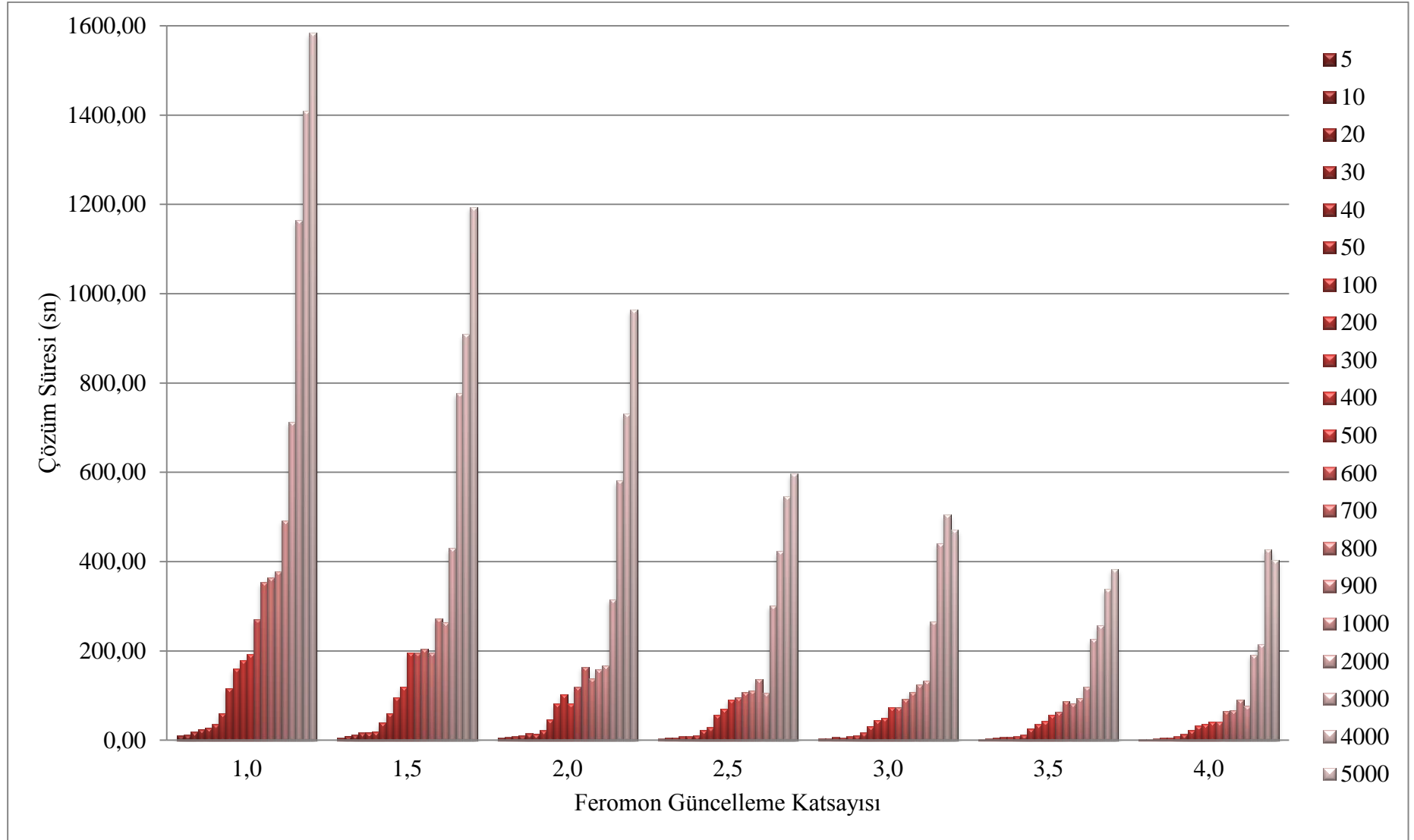
Genel olarak iterasyon sayısı, karınca sayısı ve feromon güncelleme katsayısı ile ters orantılı değişirken, Şekil 7.10’da görüldüğü gibi karınca sayısı artıkça çözüm süresi artmakta, feromon güncelleme katsayısı artıkça ise ortalama çözüm süresinde azalma olmaktadır.



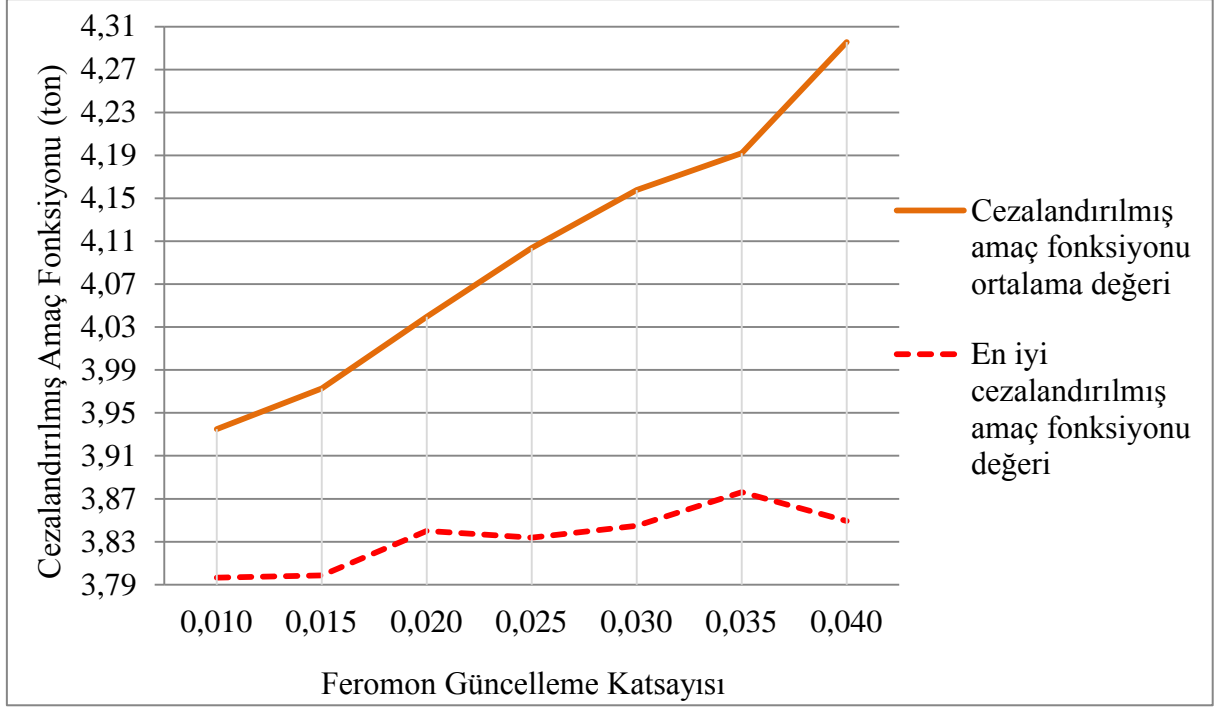
Şekil 7. 8. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre cezalandırılmış amaç fonksiyonu



Şekil 7. 9. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre iterasyon sayısı

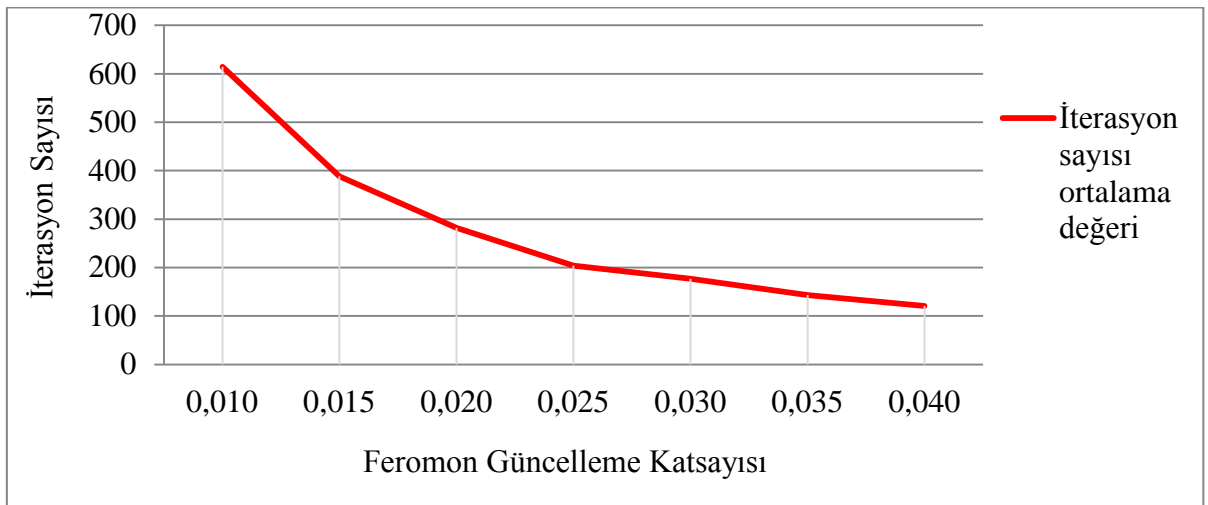


Şekil 7. 10. Karınca sayısı ve feromon güncelleme katsayısı değişimine göre çözüm süresi



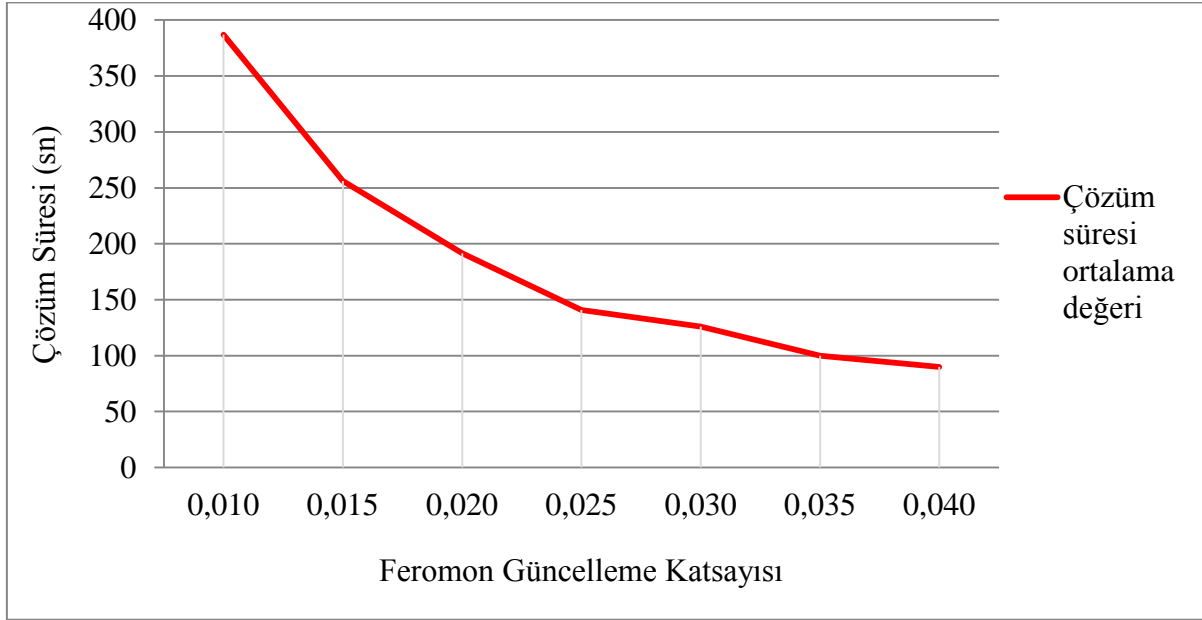
Şekil 7. 11. Feromon güncelleme katsayısına göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri

Şekil 7.11’de feromon güncelleme katsayısı değişimine göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri gösterilmiştir. Açıkça görülüyor ki, feromon güncelleme katsayısı azaldıkça en iyi cezalandırılmış amaç fonksiyonu değeri ve cezalandırılmış amaç fonksiyonu ortalama değeri de azalma eğilimi göstermektedir. En optimum çözümden feromon güncelleme katsayısının 0.010 alındığı değerdir.



Şekil 7. 12. Feromon güncelleme katsayısına göre iterasyon sayısı ortalama değeri

Şekil 7.12’de feromon güncelleme katsayısı değişimine göre iterasyon sayısı ortalama değerlerinin değişimi gösterilmiştir. Bu şekilden, feromon güncelleme katsayısı arttıkça, iterasyon sayısının azaldığı açıkça görülmektedir.



Şekil 7. 13. Feromon güncelleme katsayısı değişimine göre çözüm süresi ortalama değeri

Şekil 7.13’te feromon güncelleme katsayısı değişimine göre çözüm süresi ortalama değerlerinin değişimi gösterilmiştir. Feromon güncelleme katsayısı arttıkça çözüm süresi ortalama değeri bariz bir şekilde azalmaktadır.

7.1.3. Tasarım değişkenleri kümesinin araştırılması

Optimum tasarım için seçilecek olan çelik çubukların belirlenmesi bir problemin en önemli kısmıdır. Bu yöntemde tasarımda kullanılacak çelik çubuklar önceden belirlenmiş bir küme içerisinde seçilmektedir. Bu kümede bulunan çelik çubuklar problemin ihtiyaçlarına uygun olmalıdır. Bu nedenle, kendi bölgesi için en uygun olan çelik eleman, tasarım değişkenleri kümesi içerisinde olmadıkça, optimum çözüme ulaşmakta mümkün olmayacaktır.

Daha önceki bölümlerde 200 adetlik tasarım değişkeni kümesi kullanılmıştı. Bu bölümde, tasarım değişkenleri kümesinin optimum çözüme etkisini görebilmek için, bu küme sadeleştirilip 100, 50 ve 25 adetlik yeni kümeler ile optimum tasarım yapılmıştır. Bu kümeler

Çizelge 7.16, Çizelge 7.18 ve Çizelge 7.20’de verilmiştir. Çözüm sonuçları ise Çizelge 7.17, Çizelge 7.19 ve Çizelge 7.21’de verilmiştir. Çözümlerin tamamında ceza katsayısı 3.0 ve feromon güncelleme katsayısı 0.010 olarak alınmıştır.

Çizelge 7. 16. On çubuklu kafes sistemin çözümünde kullanılan 100 tasarım değişkeninin en kesit alanları ve atalet yarıçapları

No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı
1	11.088	3.467	21	29.920	4.217	41	40.304	5.675	61	73.700	8.298	81	134.789	12.138
2	12.270	3.454	22	17.059	4.800	42	44.113	5.648	62	80.724	8.265	82	152.362	12.081
3	13.733	3.437	23	19.121	4.783	43	33.106	6.653	63	91.143	8.214	83	109.965	14.061
4	15.180	3.420	24	21.167	4.766	44	37.105	6.629	64	52.807	9.432	84	124.583	14.019
5	16.896	3.400	25	23.602	4.745	45	41.638	6.602	65	59.334	9.404	85	136.695	13.985
6	18.869	3.377	26	26.413	4.722	46	46.690	6.572	66	66.629	9.373	86	154.746	13.933
7	21.087	3.350	27	29.589	4.695	47	51.138	6.545	67	73.070	9.346	87	175.033	13.875
8	12.525	3.916	28	33.113	4.665	48	57.734	6.504	68	82.657	9.305	88	196.315	13.814
9	13.866	3.902	29	20.917	5.231	49	63.162	6.471	69	90.577	9.271	89	140.486	15.808
10	15.529	3.885	30	23.163	5.214	50	42.134	7.527	70	102.339	9.221	90	154.189	15.773
11	17.176	3.868	31	25.837	5.194	51	47.306	7.500	71	70.692	11.203	91	174.625	15.722
12	19.131	3.848	32	28.928	5.170	52	53.077	7.469	72	79.426	11.172	92	197.615	15.663
13	21.384	3.825	33	32.423	5.143	53	58.163	7.442	73	87.148	11.145	93	221.760	15.602
14	23.921	3.798	34	36.306	5.113	54	65.717	7.401	74	98.654	11.104	94	171.820	17.576
15	15.463	4.351	35	39.716	5.087	55	71.943	7.368	75	108.174	11.070	95	194.661	17.524
16	17.325	4.334	36	23.166	5.793	56	81.164	7.318	76	122.336	11.019	96	220.376	17.466
17	19.171	4.317	37	25.661	5.776	57	47.164	8.425	77	87.397	12.293	97	247.406	17.404
18	21.366	4.297	38	28.635	5.756	58	52.974	8.397	78	95.915	12.265	98	214.696	19.327
19	23.899	4.273	39	32.076	5.732	59	59.463	8.366	79	108.617	12.224	99	243.136	19.268
20	26.755	4.247	40	35.971	5.705	60	65.188	8.339	80	119.133	12.190	100	273.051	19.206

Çizelge 7. 17. Tasarım değişkenleri kümesinde 100 değer alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.1022	4.1022	1842	10.834
2	10	-	4.0454	4.0454	886	9.276
3	20	-	4.0277	4.0277	710	13.508
4	30	-	3.8720	3.8720	826	22.501
5	40	-	3.9673	3.9673	782	27.959
6	50	-	3.9092	3.9092	739	32.918
7	100	-	3.9672	3.9672	649	56.219
8	200	-	3.8883	3.8883	548	93.351
9	300	-	3.8460	3.8460	513	130.483
10	400	-	3.8461	3.8461	433	147.047
11	500	-	3.8478	3.8478	490	206.496
12	600	-	3.8497	3.8497	376	189.982
13	700	-	3.9516	3.9516	403	237.574
14	800	-	3.8505	3.8505	372	250.858
15	900	-	3.8399	3.8399	355	269.394
16	1000	-	3.8430	3.8430	459	385.851
17	1500	-	3.8431	3.8431	389	489.856
18	2000	-	3.8425	3.8425	444	743.254
19	3000	-	3.8430	3.8430	344	865.403
20	4000	-	3.8425	3.8425	300	1005.580
21	5000	-	3.8644	3.8644	291	1224.782

Çizelge 7.17’de, daha önceki çözümlerde kullandığımız 200 adetlik tasarım değişkenleri kümesinin uygun şekilde sadeleştirilmesi sonucu elde edilen 100 adetlik bir tasarım değişkenleri kümesi araştırılarak bulunan amaç fonksiyonu, cezalandırılmış amaç fonksiyonu, iterasyon sayısı ve çözüm süresi değerleri bulunmaktadır.

Çizelge 7. 18. On çubuklu kafes sistemin çözümünde kullanılan 50 tasarım değişkeninin enkesit alanları ve atalet yarıçapları

No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı
1	11.088	3.467	26	52.974	8.397
2	12.270	3.454	27	59.463	8.366
3	15.180	3.420	28	52.807	9.432
4	15.463	4.351	29	59.334	9.404
5	19.171	4.317	30	66.629	9.373
6	17.059	4.800	31	73.070	9.346
7	19.121	4.783	32	70.692	11.203
8	21.167	4.766	33	79.426	11.172
9	23.602	4.745	34	87.148	11.145
10	20.917	5.231	35	98.654	11.104
11	23.163	5.214	36	87.397	12.293
12	25.837	5.194	37	95.915	12.265
13	23.166	5.793	38	108.617	12.224
14	25.661	5.776	39	119.133	12.190
15	28.635	5.756	40	124.583	14.019
16	32.076	5.732	41	136.695	13.985
17	33.106	6.653	42	154.746	13.933
18	37.105	6.629	43	140.486	15.808
19	41.638	6.602	44	154.189	15.773
20	46.690	6.572	45	174.625	15.722
21	42.134	7.527	46	197.615	15.663
22	47.306	7.500	47	194.661	17.524
23	53.077	7.469	48	220.376	17.466
24	58.163	7.442	49	243.136	19.268
25	47.164	8.425	50	273.051	19.206

Çizelge 7. 19. Tasarım değişkenleri kümesinde 50 değer alındığında çözüm sonuçları

Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	3.9687	3.9687	1550	8.625
2	10	-	3.8942	3.8942	1070	10.445
3	20	-	3.8807	3.8807	634	12.026
4	30	-	3.9269	3.9269	599	17.452
5	40	-	3.9495	3.9495	672	22.970
6	50	-	3.9282	3.9282	637	26.941
7	100	0.002046	3.8688	3.8925	509	42.367
8	200	-	3.8572	3.8572	416	68.704
9	300	-	3.8555	3.8555	440	107.893
10	400	-	3.8487	3.8487	361	118.087
11	500	-	3.8487	3.8487	378	155.038
12	600	-	3.8487	3.8487	335	164.354
13	700	-	3.8487	3.8487	358	203.952
14	800	-	3.8505	3.8505	386	251.438
15	900	-	3.8487	3.8487	376	275.596
16	1000	-	3.8537	3.8537	357	334.071
17	1500	-	3.8285	3.8285	305	363.487
18	2000	-	3.8505	3.8505	342	555.242
19	3000	-	3.8487	3.8487	289	709.680
20	4000	-	3.8505	3.8505	262	853.692
21	5000	-	3.8505	3.8505	223	908.766

Çizelge 7.19’da, daha önceki çözümlerde kullandığımız 200 ve 100 adetlik tasarım değişkenleri kümesinin uygun şekilde sadeleştirilmesi sonucu elde edilen 50 adetlik bir tasarım değişkenleri kümesi araştırılarak bulunan amaç fonksiyonu, cezalandırılmış amaç fonksiyonu, iterasyon sayısı ve çözüm süresi değerleri bulunmaktadır.

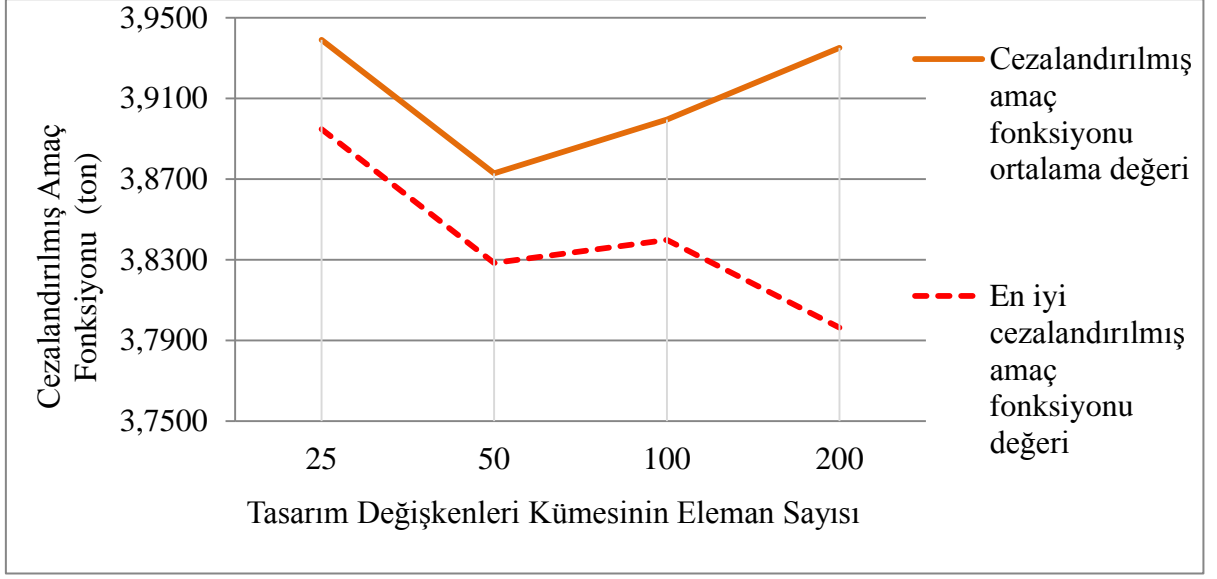
Çizelge 7. 20. On çubuklu kafes sistemin çözümünde kullanılan 25 tasarım değişkeninin en kesit alanları ve atalet yarıçapları

No	Enkesit Alanı	Atalet Yarıçapı	No	Enkesit Alanı	Atalet Yarıçapı
1	12.270	3.454	14	66.629	9.373
2	15.180	3.420	15	79.426	11.172
3	15.463	4.351	16	87.397	12.293
4	19.171	4.317	17	108.617	12.224
5	23.163	5.214	18	124.583	14.019
6	25.661	5.776	19	154.746	13.933
7	32.076	5.732	20	140.486	15.808
8	37.105	6.629	21	174.625	15.722
9	46.690	6.572	22	197.615	15.663
10	42.134	7.527	23	194.661	17.524
11	53.077	7.469	24	220.376	17.466
12	47.164	8.425	25	243.136	19.268
13	52.807	9.432			

Çizelge 7. 21. Tasarım değişkenleri kümesinde 25 değer alındığında çözüm sonuçları

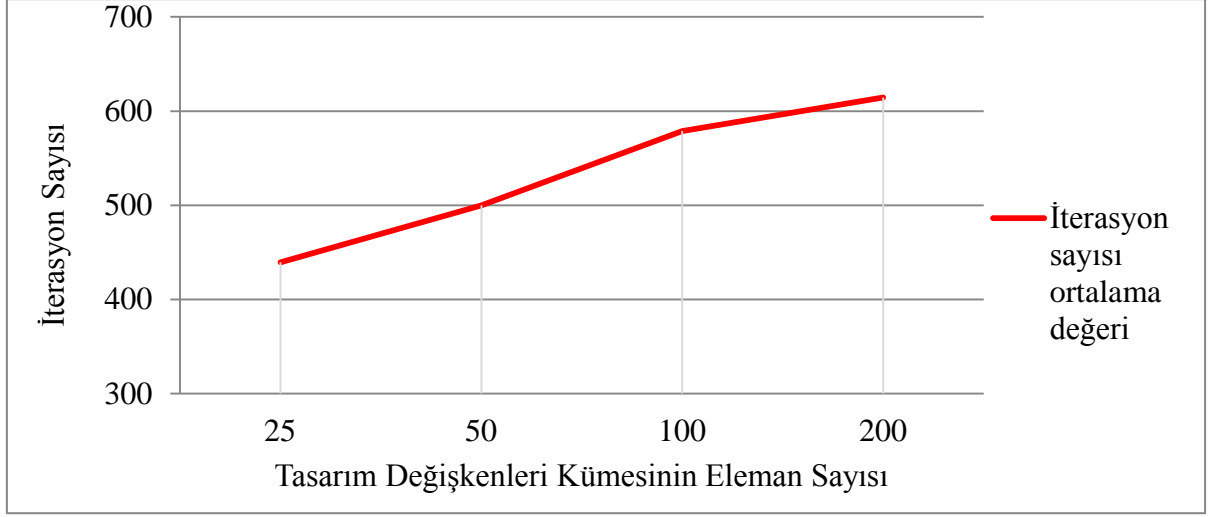
Çözüm No	Karınca Sayısı	Ceza Fonksiyonu	Amaç Fonksiyonu (ton)	Cezalandırılmış Amaç Fonksiyonu (ton)	İterasyon Sayısı	Çözüm Süresi (sn)
1	5	-	4.2012	4.2012	1412	7.160
2	10	-	3.9225	3.9225	1025	9.758
3	20	-	4.0123	4.0123	723	13.034
4	30	-	4.1621	4.1621	541	14.009
5	40	-	3.9521	3.9521	561	19.219
6	50	-	3.8948	3.8948	682	29.107
7	100	-	3.8948	3.8948	426	35.107
8	200	-	3.8948	3.8948	328	53.306
9	300	0.014671	3.8739	4.0444	312	75.648
10	400	-	3.8948	3.8948	303	97.975
11	500	-	3.8948	3.8948	332	137.658
12	600	0.001625	3.8772	3.8961	290	140.138
13	700	0.001625	3.8772	3.8961	286	169.969
14	800	-	3.8948	3.8948	281	181.780
15	900	-	3.8948	3.8948	306	221.174
16	1000	-	3.8948	3.8948	256	206.225
17	1500	-	3.8948	3.8948	281	337.786
18	2000	-	3.8948	3.8948	244	387.028
19	3000	-	3.8948	3.8948	217	526.715
20	4000	-	3.8948	3.8948	200	646.937
21	5000	-	3.8948	3.8948	218	873.365

Çizelge 7.21’de, daha önceki çözümlerde kullandığımız 200, 100 ve 50 adetlik tasarım değişkenleri kümesinin uygun şekilde sadeleştirilmesi sonucu elde edilen 25 adetlik bir tasarım değişkenleri kümesi araştırılarak bulunan amaç fonksiyonu, cezalandırılmış amaç fonksiyonu, iterasyon sayısı ve çözüm süresi değerleri bulunmaktadır.

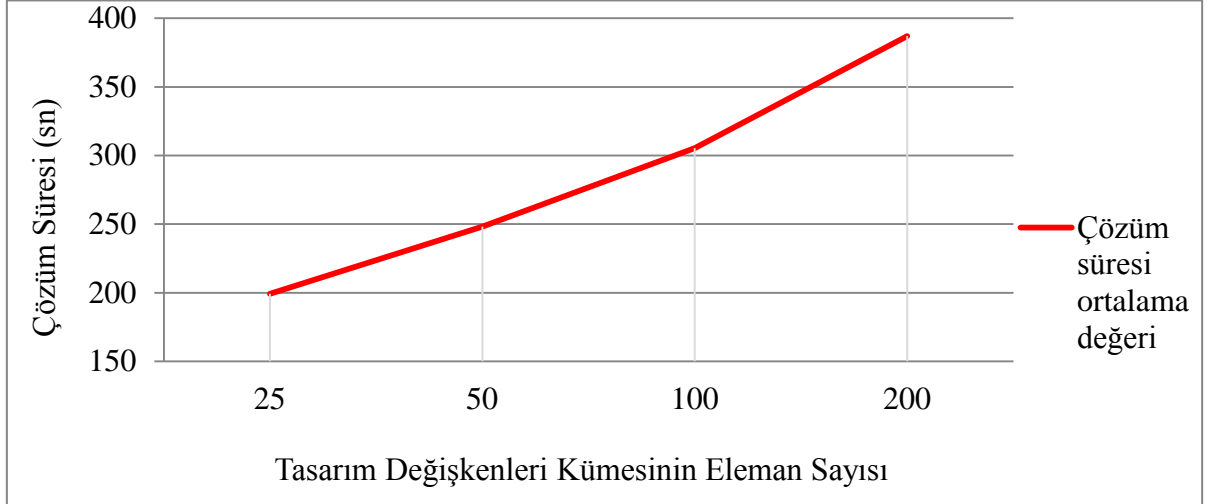


Şekil 7. 14. Tasarım değişkenleri kümesi eleman sayısına göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değeri

Şekil 7.14'te tasarım değişkenleri kümesi eleman sayısına göre cezalandırılmış amaç fonksiyonu ortalama değeri ve en iyi cezalandırılmış amaç fonksiyonu değerleri verilmiştir. Tasarım değişkeni kümesi büyüdükçe en iyi cezalandırılmış amaç fonksiyonu değeri azalmaktadır. Bu durum karıncaların seçeneğinin fazla olması ve bunların arasında daha uygun çelik çubuk elemanların olmasından kaynaklanmaktadır. Tasarım değişkenleri kümesinin geniş tutulmasının optimum tasarımı bulma açısından avantajı olmakla beraber karıncaların araştırma alanı büyük olduğu için iterasyon sayısı ortalama değeri de artmaktadır (Şekil 7.15). Aynı nedenden dolayı çözüm süresi de bariz şekilde artmaktadır (Şekil 7.16). Esasında dikkate alınması gereken asıl parametre iterasyon sayısı değil çözüm süresidir. İterasyon sayısının çok olduğu durumlarda çözüm süresi az olabilirken, iterasyon sayısı az iken çözüm süresi çok fazla artabilmektedir.



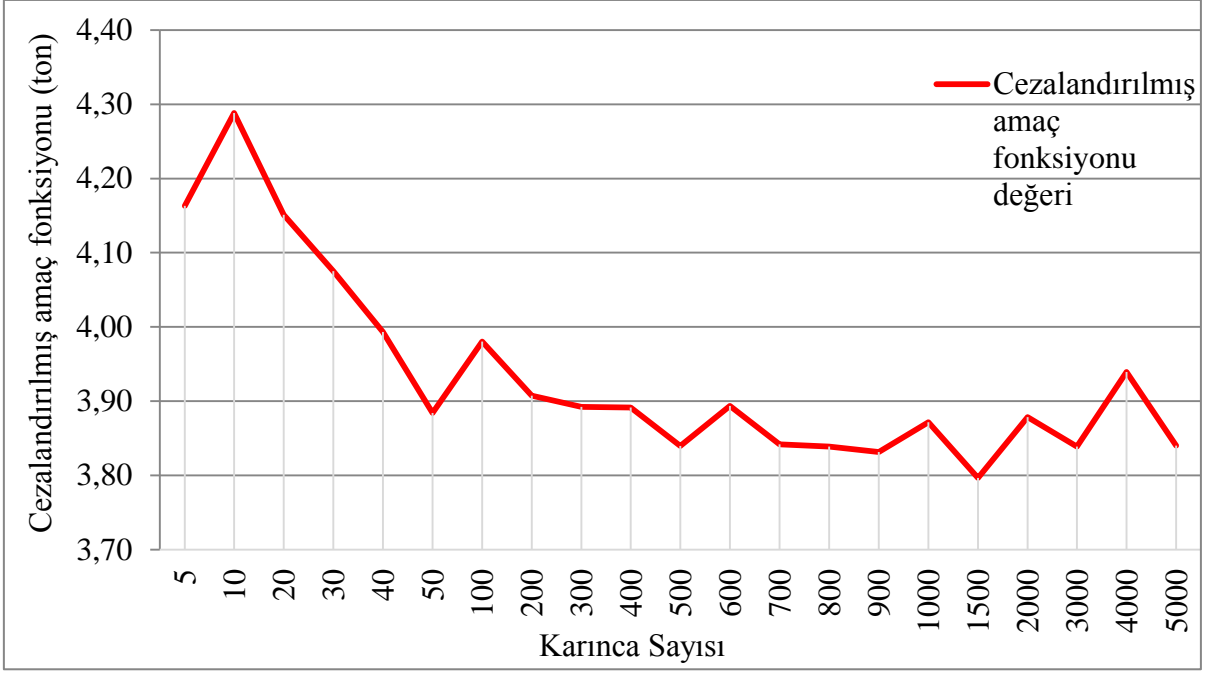
Şekil 7. 15. Tasarım değişkenleri kümesi eleman sayısına göre iterasyon sayısı ortalama değeri



Şekil 7. 16. Tasarım değişkenleri kümesi eleman sayısına göre çözüm süresi ortalama değeri

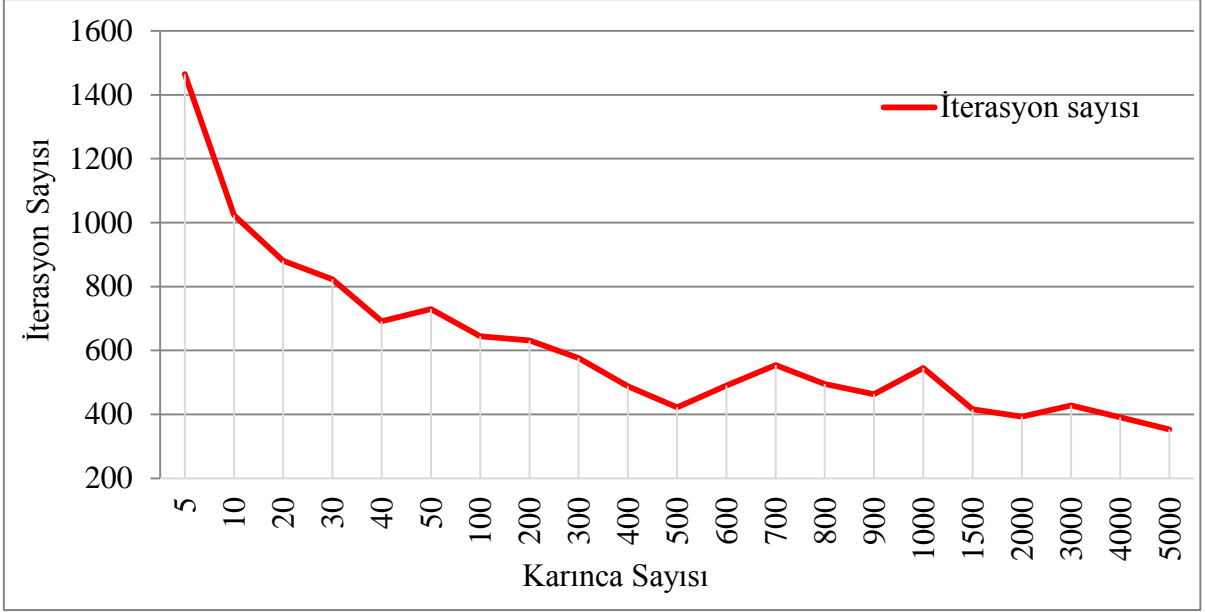
7.1.4. Karınca sayısının araştırılması

Karınca sayısının çözüm sonuçlarına nasıl etkilediğinden önceki bölümlerde çokça bahsedilmişti. En iyi çözüm sonuçlarına sahip olan set, tasarım değişkenleri sayısı 200, ceza katsayısı 3.0 ve feromon güncelleme katsayısı 0.010 olan değerlerle bulunmuştu (Çizelge 7.9). Bu çizelge üzerinden değerlendirme yapacak olursak, karınca sayısının çok düşük seçildiği çözümlerin cezalandırılmış amaç fonksiyonu değerleri yüksek çıkmıştır (Şekil 7.17). Karınca sayısı arttıkça cezalandırılmış amaç fonksiyonu düşüş göstermekle beraber, belirli bir noktadan sonra çok fazla bir değişiklik olmamış, eğri çizgisi yatay seyretmektedir.

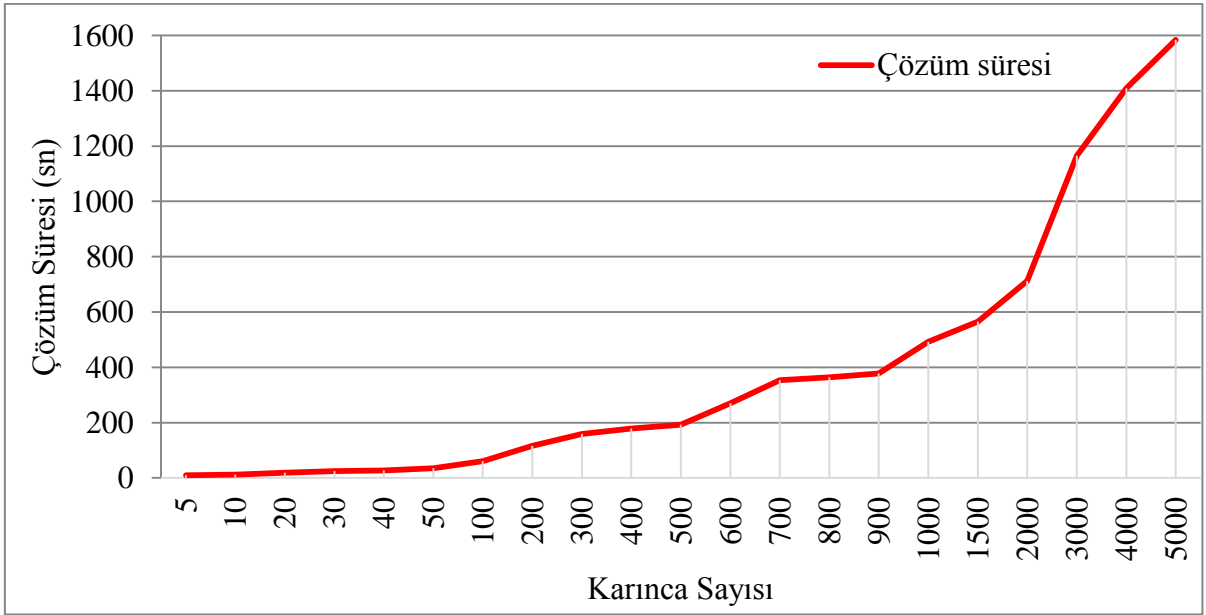


Şekil 7. 17. Karınca sayısına göre cezalandırılmış amaç fonksiyonu değeri

İterasyon sayısı ise daha önceki bölümlerde bahsedildiği gibi karınca sayısı artıkça azalma eğilimi göstermektedir (Şekil 7.18). Bununla beraber çözüm süresi, karınca sayısının artmasıyla daha çok işlem yapılacağından hızla artmıştır (Şekil 7.19).



Şekil 7. 18. Karınca sayısına göre iterasyon sayısı



Şekil 7. 19. Karınca sayısına göre çözüm süresi

7.2. On Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı

Bu bölümde on çubuklu düzlem kafes sistem (Şekil 7.1), ulaşılan sonuçların karşılaştırılabilmesi için, veriler literatürdeki örneklerle aynı olacak şekilde seçilerek yeniden optimize edilmiştir. Yukarıda da bahsedildiği gibi on çubuklu düzlem kafes sistem, literatürde en çok karşılaşılan optimum tasarım problemlerinden bir tanesidir. Diğer birçok çalışmadaki

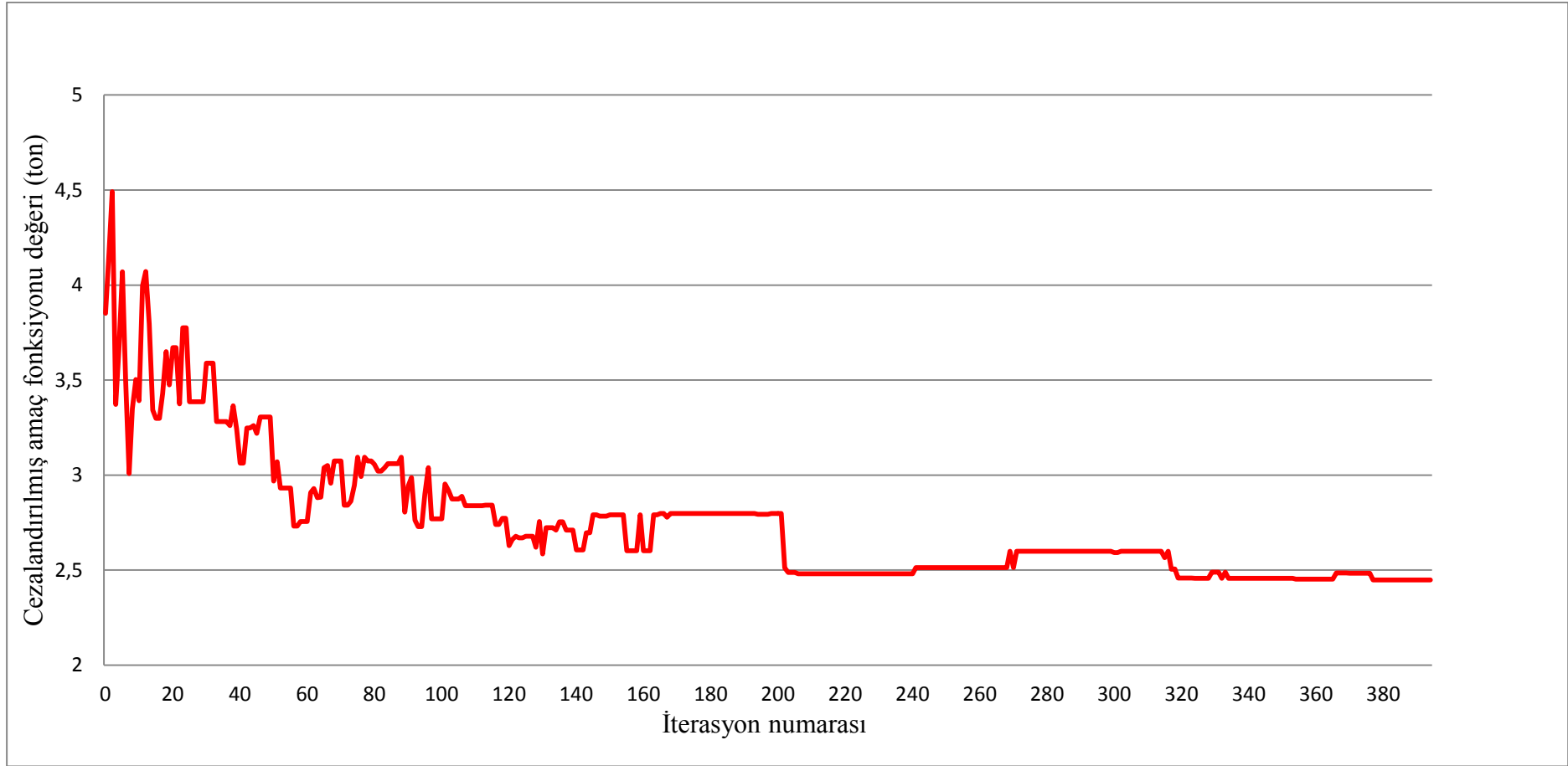
gibi bu çalışmada da çubukların birim hacim ağırlığı $\rho = 0.11\text{lb/in}^3$ (2.768 t/m^3), malzemenin elastisite modülü $E=10000\text{ ksi}$ (68947.59MPa), gerilme sınırlayıcısı olarak malzemenin emniyet gerilmesi $\sigma_{em} = \pm 25\text{ ksi}$ ($172,37\text{ MPa}$) alınmıştır. Deplasman sınırlayıcısı olarak tüm noktaların yerdeğiştirme sınırı iki doğrultuda da $\pm 2.0\text{ in}$ (5.08 cm)'dir. Bu örnekte narinlik sınırlayıcısı dikkate alınmamıştır. 4 ve 5 numaralı düğüm noktalarından $P= 100\text{ kips}$ (444.82kN) yük uygulanmıştır. Bu örneğin optimum tasarımında gruplandırma yapılmayıp on çubukta ayrı ayrı boyutlandırılmıştır. Tasarım değişkenleri değer kümesi ise yine literatürde olduğu şekilde, $A = \{1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50\}$ (in^2) 42 adet değer kullanılmıştır.

Önceki bölümlerde yapılan detaylı araştırmaların neticesinde, kullanılan optimizasyon yöntemi için gerekli parametrelerden ceza katsayısı 3.0, feromon güncelleme katsayısı 0.010 ve karınca sayısı 500 alınmıştır. Bu veriler doğrultusunda problem 396 iterasyonda çözülmüş ve 2448.11 kg ağırlık bulunmuştur.

On çubuklu düzlem kafes sistem için bu çalışmada yapılan optimum tasarıma ait sonuçlar ve aynı problemin literatürdeki diğer çözümleri ile karşılaştırmalı olarak sunumu Çizelge 7.22'de verilmektedir. Bu çizelgeden görüleceği üzere, bu çalışmada ulaşılan sonuç, literatürdeki diğer çalışmalarla birlikte en iyi sonuçlardan biridir.

Çizelge 7. 22. On çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar

Çubuk No	Bu Çalışma	Li ve ark. (2009)	Toğan ve Daloğlu (2008)	Camp ve ark. (1998)
1	30.00	30.00	33.50	30.00
2	1.62	1.62	1.62	1.62
3	22.00	22.90	22.90	26.50
4	16.00	13.50	16.00	13.50
5	1.62	1.62	1.62	1.62
6	1.62	1.62	1.62	1.62
7	7.97	7.97	7.97	7.22
8	22.90	26.50	22.90	22.90
9	22.00	22.00	19.90	22.00
10	1.62	1.80	1.62	1.62
Ağırlık (lb)	5397.17	5531.98	5448.62	5556.90
Ağırlık (kg)	2448.11	2509.26	2471.45	2520.57



Şekil 7. 20. On çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması

7.3. Elliiki Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı

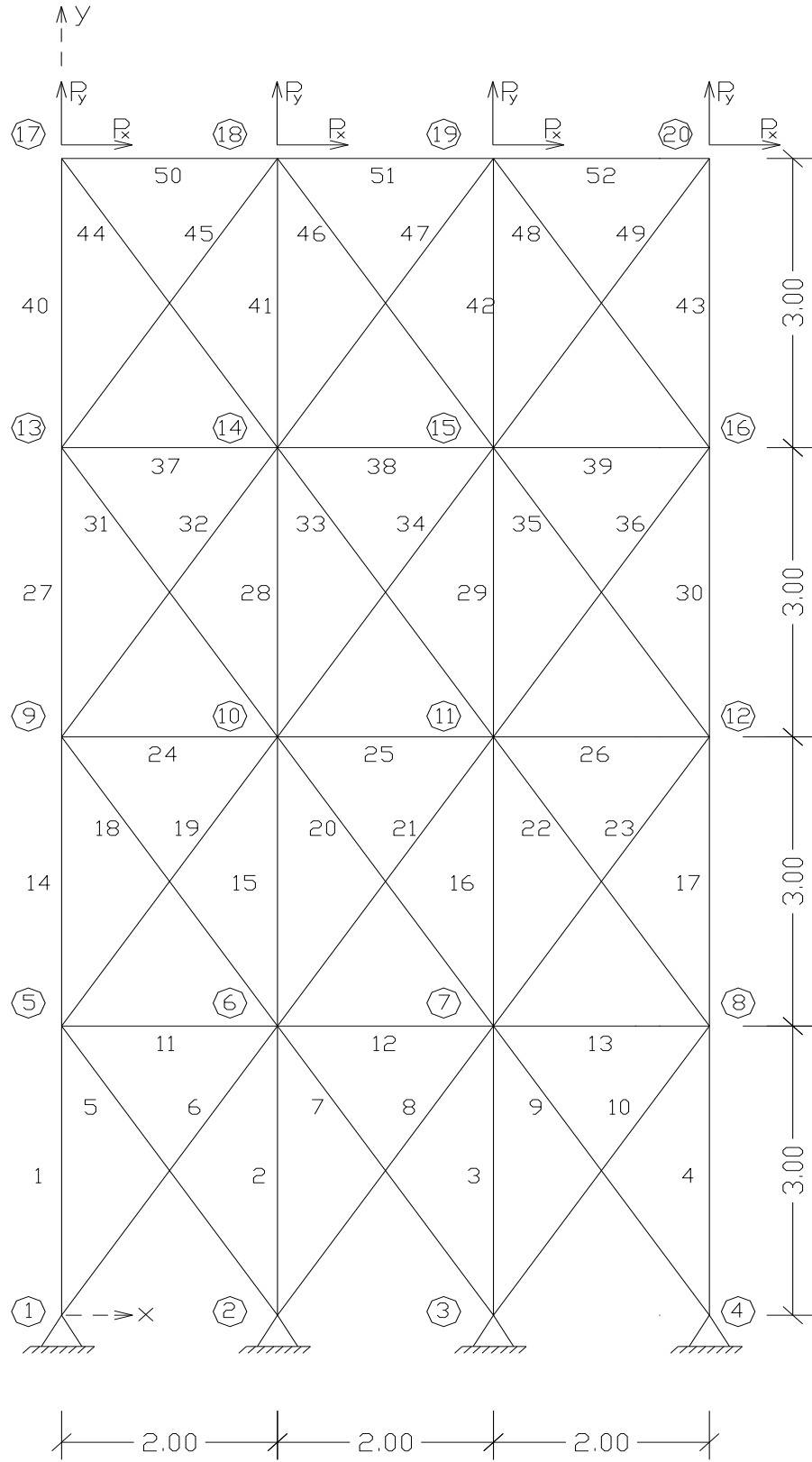
Literatürde yine çokça örneği bulunan bir diğer kafes sistem ise elliiki çubuklu düzlem kafes sistemidir. Şekil 7.20’de gösterilen elliiki çubuklu düzlem kafesin ölçüleri metre cinsindedir. Optimum tasarımı için çubukların birim hacim ağırlığı $\rho = 7860 \text{ kg/m}^3$, malzemenin elastisite modülü $E = 207 \times 10^3 \text{ MPa}$, gerilme sınırlayıcısı olarak malzemenin emniyet gerilmesi $\sigma_{em} = \pm 180 \text{ MPa}$ alınmıştır. Bu örnekte deplasman ve narinlik sınırlayıcıları dikkate alınmamıştır. Sisteme etki eden dış yükler 17, 18, 19 ve 20 numaralı noktalardan x yönünde $P_x = 100 \text{ kN}$ ve y yönünde $P_y = 200 \text{ kN}$ olarak uygulanmıştır. Bu örneğin optimum tasarımı gruplandırma yöntemi ile yapılmış olup çubuklar 12 gruba ayrılmıştır. Tasarım değişkenleri değer kümesi ise yine literatürde olduğu şekilde, $A = \{ 0.11, 0.14, 0.20, 0.25, 0.31, 0.39, 0.44, 0.56, 0.60, 0.77, 0.79, 0.99, 1.00, 1.23, 1.27, 1.46, 1.56, 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 8.53, 9.30, 10.85, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 24.50, 26.50, 28.00, 30.00, 33.50 \}$ (in²) olmak üzere 64 adet değer kullanılmıştır.

Kullanılan optimizasyon yöntemi için gerekli parametrelerden ceza katsayısı 5.0, feromon güncelleme katsayısı 0.010 ve karınca sayısı 500 alınmıştır. Bu veriler doğrultusunda problem 505 iterasyonda çözülmüş ve 1902.61 kg ağırlık bulunmuştur.

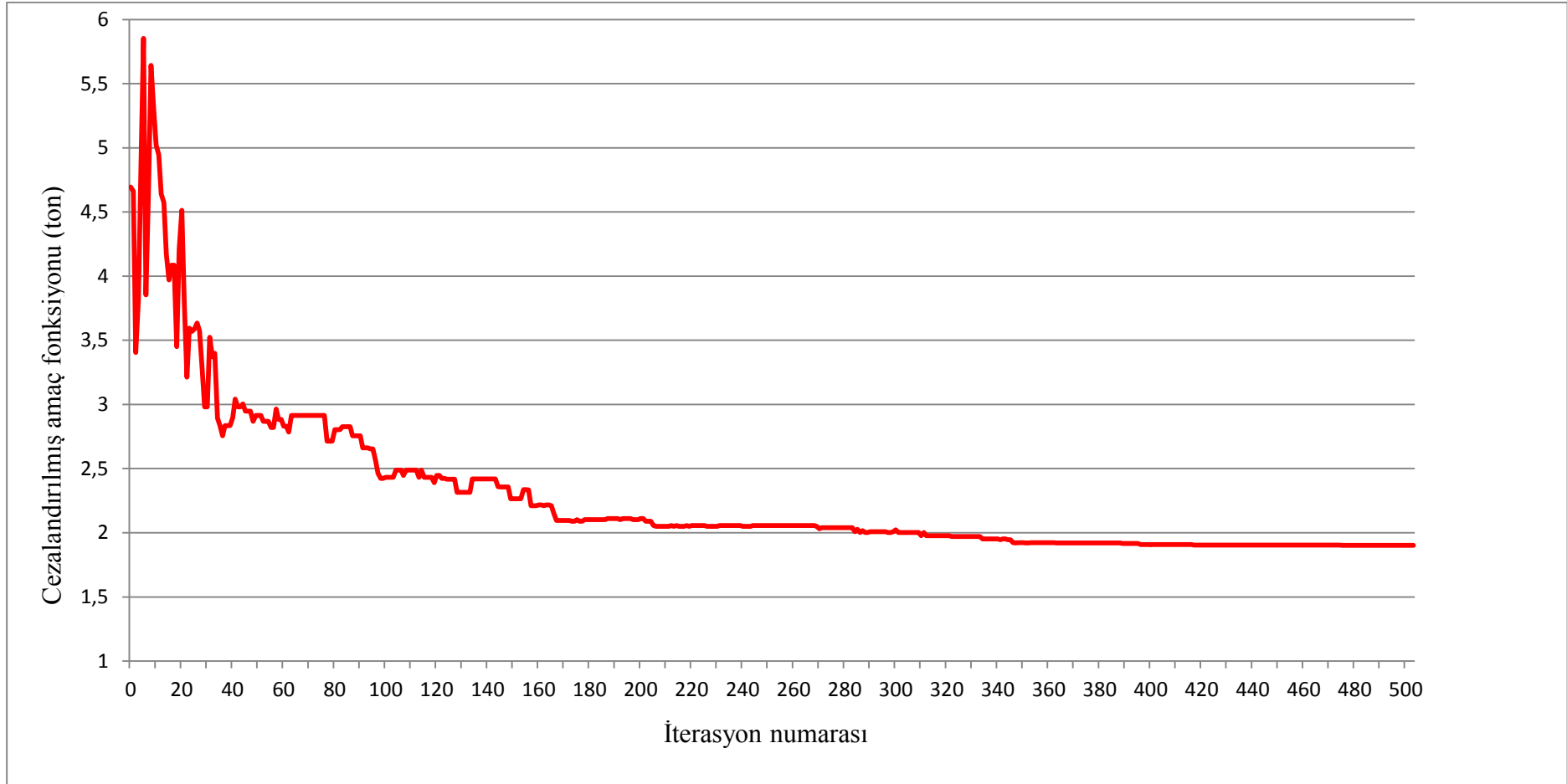
Elliiki çubuklu düzlem kafes sistem için bu çalışmada yapılan optimum tasarıma ait sonuçlar ve aynı problemin literatürdeki diğer çözümleri ile karşılaştırmalı olarak sunumu Çizelge 7.23’te verilmektedir. Bu çizelgeden görüleceği üzere bu çalışmada ulaşılan sonuç, literatürdeki diğer çalışmaların sonuçları ile kıyaslandığında, en iyi sonuca ulaşılamamış olsa da ulaşılan sonuç en iyi çözüme oldukça yakındır.

Çizelge 7. 23. Elliiki çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar

Grup No	Çubuk No	Bu Çalışma	Wu ve Chow (1995)	Li ve ark. (2009)	Capriles ve ark. (2007)	Kaveh ve Talatahari (2009)
1	1,2,3,4	7.220	7.220	7.220	7.220	7.220
2	5,6,7,8,9,10	1.800	1.800	1.800	1.800	1.800
3	11,12,13	0.766	1.000	0.563	0.785	0.766
4	14,15,16,17	5.120	5.120	5.120	5.120	5.120
5	18,19,20,21,22,23	1.457	1.620	1.457	1.457	1.563
6	24,25,26	0.766	0.766	0.766	0.785	0.442
7	27,28,29,30	3.470	3.840	3.470	3.470	3.550
8	31,32,33,34,35,36	1.563	1.620	1.563	1.563	1.563
9	37,38,39	0.602	0.442	0.602	0.602	0.602
10	40,41,42,43	1.990	2.630	1.990	1.990	1.990
11	44,45,46,47,48,49	1.800	1.620	1.800	1.800	1.800
12	50,51,52	0.766	0.994	1.228	0.785	0.785
	Ağırlık (lb)	4194.53	4343.42	4200.90	4187.35	4199.43
	Ağırlık (kg)	1902.61	1970.14	1905.50	1899.35	1904.83



Şekil 7. 21. Elliiki çubuklu düzlem kafes sistem



Şekil 7. 22. Elliiki çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması

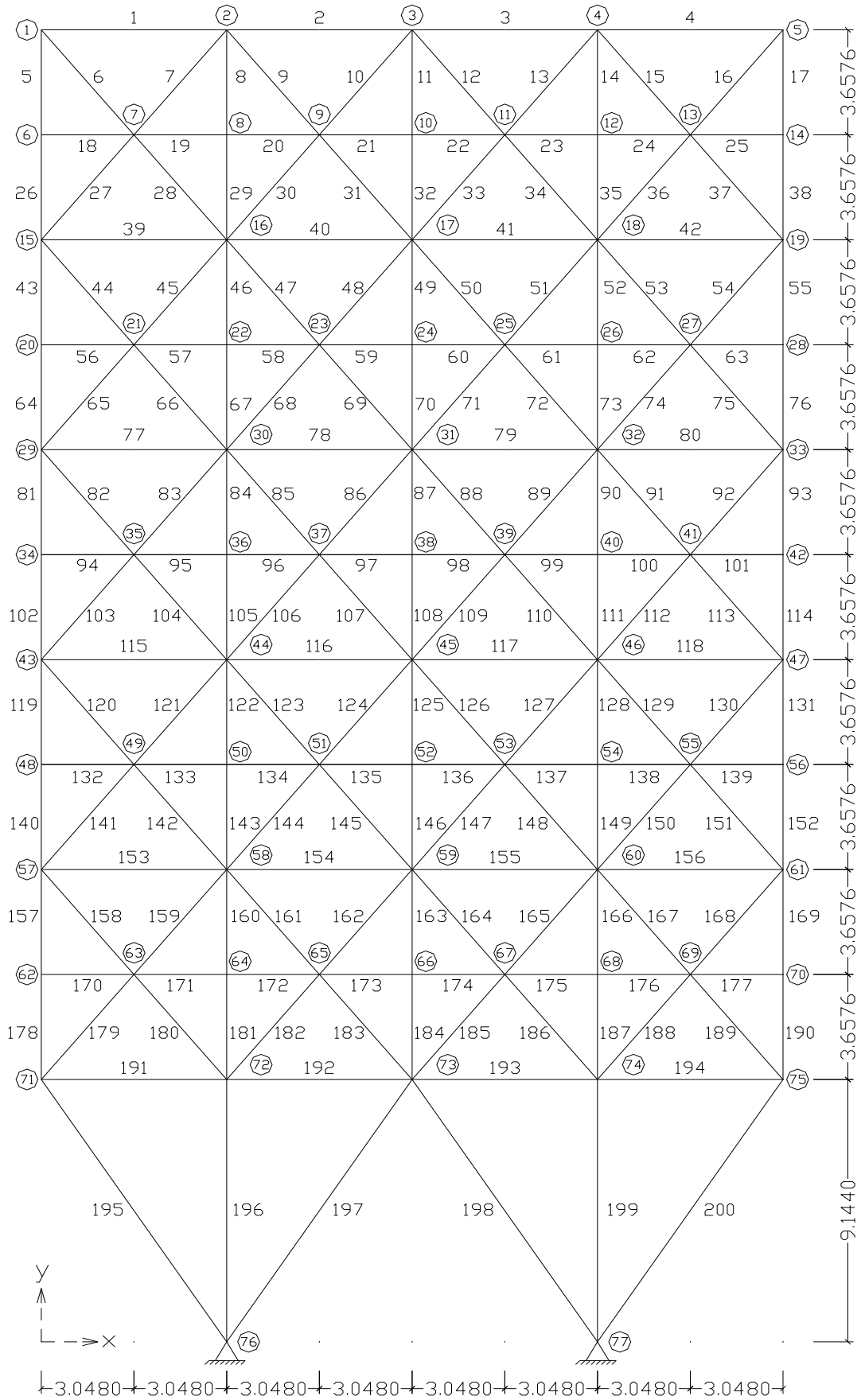
7.4. İkiyüz Çubuklu Düzlem Kafes Sistemin Optimum Tasarımı

Bu çalışmada son olarak ikiyüz çubuklu düzlem kafes sistemin optimum tasarımı gerçekleştirilmiştir. Şekil 7.21’de gösterilen ikiyüz çubuklu düzlem kafesin ölçüleri metre cinsindedir. İkiyüz çubuklu düzlem kafes sistemin optimum tasarımı için çubukların birim hacim ağırlığı $\rho = 0.283 \text{ lb/in}^3$, malzemenin elastisite modülü $E = 30000 \text{ ksi}$, gerilme sınırlayıcısı olarak malzemenin emniyet gerilmesi $\sigma_{em} = \pm 10 \text{ ksi}$ alınmıştır. Bu örnekte deplasman ve narinlik sınırlayıcıları dikkate alınmamıştır. Sisteme 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, ve 71 numaralı noktalardan x yönünde $P_x = 1$ kip yük, bununla beraber y yönünde 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74 ve 75 numaralı noktalardan $P_y = -10$ kips yük uygulanmıştır. Bu örneğin optimum tasarımı gruplandırma yöntemi ile yapılmış olup çubuklar 29 gruba ayrılmıştır. Tasarım değişkenleri değer kümesi ise yine literatürde olduğu şekilde, $A = \{ 0.100, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180, 23.680, 28.080, 33.700 \}$ (in^2) olmak üzere 30 adet değer kullanılmıştır.

Kullanılan optimizasyon yöntemi için gerekli parametrelerden ceza katsayısı 3.0, feromon güncelleme katsayısı 0.010 ve karınca sayısı 200 alınmıştır. Bu veriler doğrultusunda problem 1115 iterasyonda çözülmüş ve 13015.44 kg ağırlık bulunmuştur.

İkiyüz çubuklu düzlem kafes sistem için bu çalışmada yapılan optimum tasarıma ait sonuçlar ile aynı problemin literatürdeki diğer çözümleri ile karşılaştırmalı olarak sunumu Çizelge 7.24’te verilmektedir. Bu çizelgeden görüleceği üzere bu çalışma ulaşılan sonuç, literatürdeki diğer çalışmaların sonuçları ile kıyaslandığında, bir önceki örnekte olduğu gibi, ulaşılan sonuç diğer çözümlerle kıyaslanabilecek bir çözümdür.

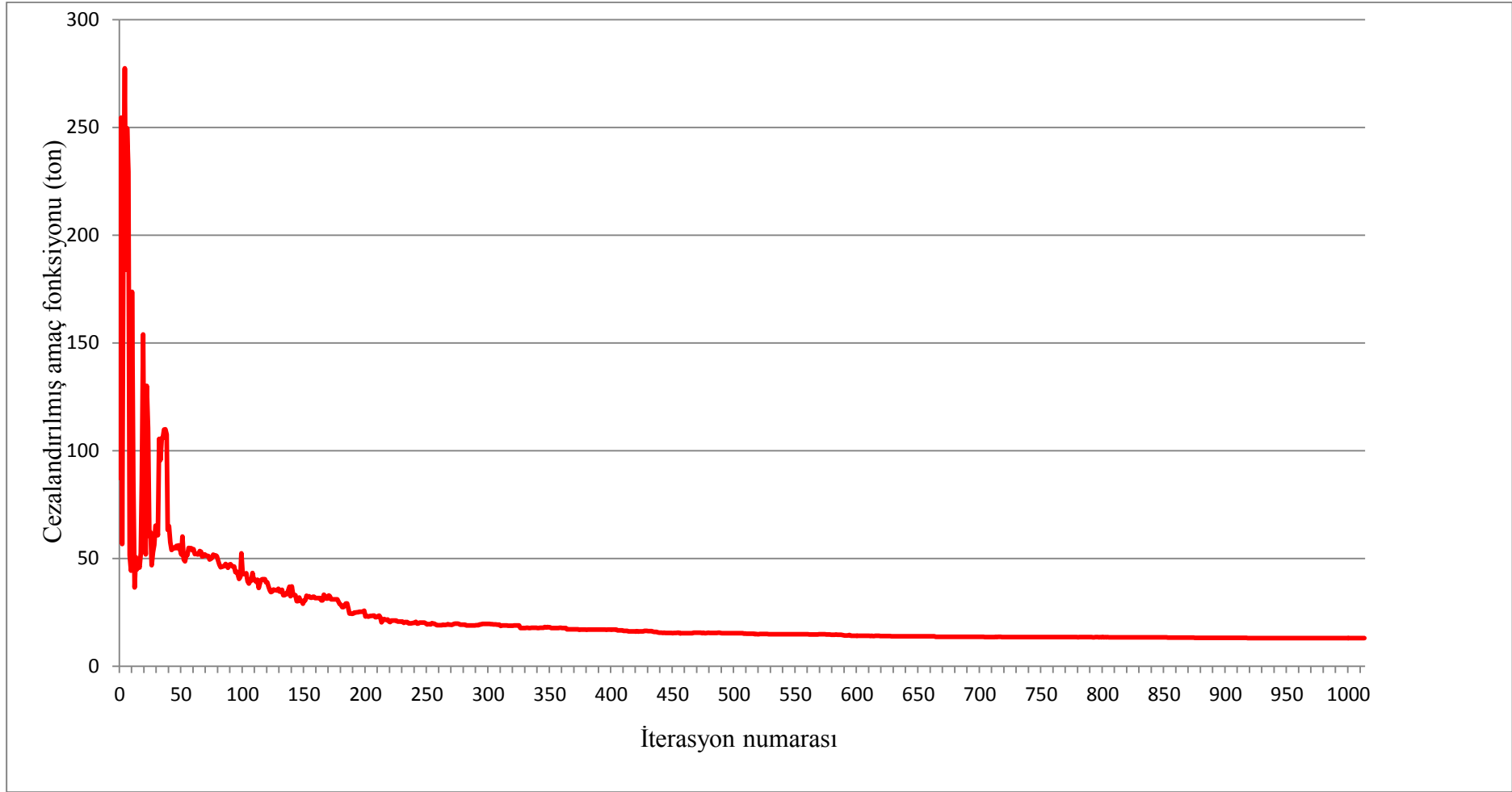
Bu örnek literatürde çok araştırmacı tarafından çözülmüş ve çok daha iyi sonuçların elde edildiği görülmüş ancak tasarım değişkenleri kümesi onlarda sürekli kabul edildiği için onlarla bir karşılaştırma yapılamamıştır.



Şekil 7. 23. İkiyüz çubuklu düzlem kafes sistem

Çizelge 7. 24. İkiyüz çubuklu kafes sistem için bu çalışmaya ve diğer çalışmalara ait sonuçlar

Grup No	Çubuk No	Bu Çalışma	Toğan ve Daloğlu (2008)	Talebpour ve ark. (2014)
1	1,2,3,4	0.100	0.347	0.100
2	5,8,11,14,17	0.954	1.081	1.081
3	19,20,21,22,23,24	0.100	0.100	0.347
4	18,25,56,63,94,101,132,139,170,177	0.347	0.100	0.100
5	26,29,32,35,38	2.142	2.142	2.142
6	6,7,9,10,12,13,15,16,27,28,30,31,33,34,36,37	0.347	0.347	0.347
7	39,40,41,42	0.347	0.100	0.100
8	43,46,49,52,55	2.800	3.565	3.131
9	57,58,59,60,61,62	0.347	0.347	0.100
10	64,67,70,73,76	3.813	4.805	4.805
11	44,45,47,48,50,51,53,54,65,66,68,69,71,72,74,75	1.081	0.440	0.440
12	77,78,79,80	0.347	0.440	0.100
13	81,84,87,90,93	5.952	5.952	5.952
14	95,96,97,98,99,100	0.539	0.347	0.100
15	102,105,108,111,114	6.572	6.572	6.572
16	82,83,85,86,88,89,91,92,103,104,106,107,09,110,112,113	1.081	0.954	0.539
17	115,116,117,118	0.347	0.347	1.174
18	119,122,125,128,131	8.525	8.525	8.525
19	133,134,135,136,137,138	0.100	0.100	0.100
20	140,143,146,149,152	9.300	9.300	9.300
21	120,121,123,124,126,127,129,130,141,142,144,145,147,148,150,151	1.081	0.954	1.333
22	153,154,155,156	1.333	1.764	0.539
23	157,160,163,166,169	13.330	13.330	13.330
24	171,172,173,174,175,176	0.539	0.347	1.174
25	178,181,184,187,190	13.330	13.330	13.330
26	158,159,161,162,164,165,167,168,179,180,182,183,185,186,188,189	2.697	2.142	2.697
27	191,192,193,194	3.565	4.805	3.565
28	195,197,198,200	8.525	9.300	8.525
29	196,199	17.170	17.170	17.170
Ağırlık (lb)		28694.13	28544.01	28030.20
Ağırlık (kg)		13015.44	12947.35	12714.29



Şekil 7. 24. İkiyüz çubuklu kafes sistem için cezalandırılmış amaç fonksiyonu yakınsaması

8. SONUÇLAR

Bu çalışmanın amacı düzlem kafes sistemlerin Karınca Kolonisi yöntemi ile optimum tasarımını yapmakla beraber, kullanılan optimizasyon parametrelerinin çözümü nasıl etkidiğini ve sürecin nasıl ilerlediğini göstermektir. Ele alınan optimizasyon probleminin amaç fonksiyonu olarak kafes sistemin ağırlığı seçilmiştir. Sınırlayıcılar olarak ise narinlik, gerilme ve deplasman dikkate alınmıştır. Yapısal analiz ve optimizasyon süreci için gerekli hesaplamaları yapmak üzere bir bilgisayar programı kodlanmıştır. Bu bilgisayar programı ile öncelikle literatürde çok sık kullanılan 10 çubuklu düzlem kafes sistem örneği üzerinden ceza katsayısı, feromon güncelleme katsayısı, tasarım değişkeni kümeleri ve karınca sayıları parametrelerinin optimizasyon sürecine etkisi araştırılmıştır. Ayrıca kodlanan bilgisayar programı ile yine literatürde sık karşılaşılan 52 çubuklu ve 200 çubuklu kafes sistemlerde çözülmüş ve sonuçlar diğer çalışmalar ile karşılaştırılmıştır. Yapılan çalışma neticesinde aşağıdaki sonuçlara ulaşılmıştır:

- Bu çalışmada benzerlerine oranla daha basit bir algoritma kullanılmış olmasına rağmen, literatürdeki çalışmalardan daha iyi veya onlara çok yakın sonuçlara ulaşılmıştır.
- Ceza katsayısı değerinin değişimi, ulaşılan çözümün optimum değere yakınlığı, iterasyon sayısı ve çözüm süresi üzerinde düzenli bir etki göstermeyip, problem özelinde değişkenlik göstermektedir. Ancak çözümü yapılan örneklerden yola çıkarak, optimum çözüme ulaşılabilmesi için ceza katsayısı değerinin önemli olduğu görülmektedir.
- Feromon güncelleme katsayısının azalması sonucunda iterasyon sayısında ve çözüme ulaşma süresinde lineer olmayan bir artma gerçekleşirken, cezalandırılmış amaç fonksiyonunun ortalama değerinde ve en iyi çözüme ait değerinde yaklaşık olarak doğrusal bir azalma görülmektedir.
- Tasarım değişkeni değer kümesi uzunluğu arttıkça iterasyon sayısı ve çözüme ulaşma süresi doğrusal olarak artmakta, cezalandırılmış amaç fonksiyonunun en iyi çözüm için değeri genel olarak azalmaktadır. Ancak cezalandırılmış amaç fonksiyonunun ortalama değeri önce azalmasına rağmen daha sonrasında artmaktadır. Buradan hareketle yeterli sayıda tasarım değişkeni değerinin

kullanılması önemli olmakla birlikte gereğinden fazla deęerin kullanılmasının sonuca olumsuz etkileri olabileceęi görölmektedir.

- Karınca sayısının artmasına baęlı olarak cezalandırılmıř amaç fonksiyonu deęeri önce hızlı bir řekilde azalmıř daha sonra ise yatay bir seyir izlemiř, iterasyon sayısı azalırken çözüme ulařma süresi ise doęrusal olmayan bir řekilde artmıřtır. Bu nedenle en iyi çözüme kısa sürede ulařılabilmesi için karınca sayısının optimum bir deęerde olması önemlidir.

9. KAYNAKLAR

- AISC-ASD (1989), Manual of Steel Construction-Allowable Stress Design, American Institute of Steel Construction, Chicago, IL, USA.
- Angelo J S, Bernardino H S, Barbosa H J C (2014). Ant Colony Approaches for Multiobjective Structural Optimization Problems with a Cardinality Constraint. Science Direct, 80:101-115.
- Aydın Z (2000). Uzak Kafes Sistemlerin Genetik Algoritma ile Optimum Tasarımı. Y.Lisans Tezi, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Trabzon.
- Aydoğdu İ (2010). Optimum Design of 3-D Irregular Steel Frames Using Ant Colony Optimization and Harmony Search Algorithms. Ph.D Thesis, The Graduate School of Natural and Applied Science, Ankara.
- Baykan U N (2007). İnşaat Projelerinde Kaynak İhtiyacının Yapay Sinir Ağları Yaklaşımı ile Tahmini. Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Bland J A (2001). Optimal Structural Design by Ant Colony Optimization. Taylor&Francis, 33:425-443.
- Bullnheimer B, Hartl R F, Strauss C (1999). A New Rank Based Version of the Ant System- A Computational Study. Central European Journal of Operations Research, 7:25-38.
- Camp C V, Pezeshk S, Cao G (1998). Optimized design of two-dimensional structure using genetic algorithm. J Struct Eng.,124:551-9
- Camp C V, Bichon B J (2004). Design of Space Trusses Using Ant Colony Optimization. Journal of Structural Engineering, 130:741-751.
- Camp C V, Bichon B J, Stovall S P (2005). Design of Steel Frames Using Ant Colony Optimization. Journal of Structural Engineering, 131:369-379.
- Capriles P V S Z, Fonseca L G, Barbosa H J C, Lemonge A C C (2007). Rank-Based Ant Colony Algorithms for Truss Weight Minimization with Discrete Variables, 23:553-575.
- Dede T, Bekiroğlu S, Ayvaz Y (2010). Weight Minimization of Trusses with Genetic Algorithm. Science Direct, 11:2565-2575.
- Dorigo M (1992). Optimization, Learning and Natural Algorithms. PhD Thesis, Politecnico di Milano, Italy.

- Dorigo M, Gambardella L M (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1:53-56.
- Emirođlu M E (1999). Barajların Tip Seçiminin Belirlenmesinde Uzman Sistem Kullanımı. Doktora tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Goldberg D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-esley Publishing Company, Inc., New York, NY, USA.
- Kalatjari V R, Talebpour M H (2017). An Improved Ant Colony Algorithm for the Optimization of Skeletal Structures by the Proposed Sampling Search Space Method. *Periodica Polytechnica Civil Engineering*, 61:232-243.
- Karabođa D (2004). *Yapay Zeka Optimizasyon Algoritmaları*. Nobel Yayın Dağıtım, 225, Ankara.
- Kaveh A, Shahrouzi M (2007). Optimal Structural Design Family by Genetic Search and Ant Colony Approach. *Emerald*, 25:268-288.
- Kaveh A, Hassani B, Shojaee S, Tavakkoli S M (2008). Structural Topology Optimization Using Ant Colony Methodology. *Science Direct*, 30:2559-2565.
- Kaveh A, Talatahari S (2009). Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures*, 87:267-283.
- Keskintürk T, Söyler H (2006). Global Karınca Kolonisi Optimizasyonu, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 21:689-698.
- Li L J, Huang Z B, Liu F (2009). A Heuristic Particle Swarm Optimization Method for Truss Structures with Discrete Variables. *Science Direct*, 87:435-443.
- Rajeev S, Krishnamoorthy C S (1997), Genetic algorithms-based methodologies for design optimization of trusses, *J. Struct. Eng.*, 123(3);350-358.
- Serin S, Terzi S (2008). Ulaştırma Mühendisliğinde Karınca Kolonisinin Kullanılması. *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, 167-173, Isparta
- Serra M, Venini P (2006). On Some Applications of Ant Colony Optimization Metaheuristic to Plane Truss Optimization. *Springer*, 32:499-506.

- Stützle T, Hoos H (1997). The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. IEEE International Conference on Evolutionary Computation, 1:309-314.
- Sönmez M (2018). Performance Comparison of Metaheuristic Algorithms for The Optimal Design of Space Trusses. Arabian Journal for Science and Engineering, 43:5265-5281.
- Şen, Z. (2001). Bulanık Mantık ve Modelleme İlkeleri. Su, Bilge Kültür Sanat Yayınları, 176s, İstanbul.
- Talebpour M H, Kaveh A, Kalatjari V R (2014). Optimization of Skeletal Structures Using a Hybridized Ant Colony-Harmony Search-Genetic Algorithm. IJST, 38:1-20.
- Toğan V, Daloğlu A T (2008). An Improved Genetic Algorithm with Initial Population Strategy and Self-Adaptive Member Grouping. Science Direct, 86:1204-1218.
- Turing A M (1950). Computing Machinery and Intelligence. Mind, 49:433-460.
- Uğur A, Aydın D (2006). Ant System Algoritmasının Java ile Görselleştirilmesi. Akademik Bilişim 2006, 9-11, Pamukkale Üniversitesi, Denizli.
- Urgan B (2011). Karınca Koloni Algoritmaları ve Bir Uygulama. Y.Lisans Tezi, Mimar Sinan Güzel Sanatlar Üniversitesi, İstanbul.
- Wu S J, Chow P T (1995). Steady-State Genetic Algorithms for Discrete Optimization of Trusses. Computer and Structures, 56:979-991.

ÖZGEÇMİŞ

Abidin Hakan Yılmaz 1986 yılında Kırklareli'nin Babaeski ilçesinde doğdu. İlkokul, ortaokul ve liseyi Babaeski'de tamamladı. 2009 yılında Atatürk Üniversitesi İnşaat Mühendisliği bölümünden mezun oldu. Halen Tekirdağ Namık Kemal Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimine devam etmektedir.