

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ÇOK AMAÇLI OPTİMİZASYON TEMELLİ GENEL ATAMA
PROBLEMLERİNİN METASEZGİSEL YÖNTEMLERLE ÇÖZÜMÜ**

Barış SATAR

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2015**

Her hakkı saklıdır

TEZ ONAYI

Barış SATAR tarafından hazırlanan “Çok Amaçlı Optimizasyon Temelli Genel Atama Problemlerinin Metasezgisel Yöntemlerle Çözümü” adlı tez çalışması 18/02/2015 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. A. Egemen YILMAZ

Jüri Üyeleri:

Başkan: Doç. Dr. Emre AKTAŞ

Hacettepe Üniversitesi

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Üye : Doç. Dr. A. Egemen YILMAZ

Ankara Üniversitesi

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Üye : Yrd. Doç. Dr. Ahmet AKBULUT

Ankara Üniversitesi

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. İbrahim DEMİR

Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

18.02.2015

Barış SATAR

ÖZET

Yüksek Lisans Tezi

ÇOK AMAÇLI OPTİMİZASYON TEMELLİ GENEL ATAMA PROBLEMLERİNİN METASEZGİSEL YÖNTEMLERLE ÇÖZÜMÜ

Barış SATAR

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. A. Egemen YILMAZ

Gerçek dünyada karşımıza çıkan genel atama problemlerinden bir tanesi olan Frekans Atama Problemi NP (non-polynomial) zor tümleşik optimizasyon problemidir. Bu tür problemlerde çözüm zamanının problem boyutuna bağlı olarak üssel olarak artmasıyla beraber, sınırlı hesaplama süresi içerisinde çözülememesinden dolayı metasezgisel yöntemler kullanılmıştır.

Bu tezde üzerinde çalışılan Frekans Atama Probleminde birbirleri ile çelişen farklı amaçların metasezgisel yöntemler kullanarak optimize edilmesiyle probleme çözüm aranmıştır. Bu çok amaçlı optimizasyon problemini çeşitli skalarizasyon yaklaşımları ile tek amaca indirgeyerek; probleme, metasezgisel yöntemlerden genetik algoritma (GA) ve parçacık sürüsü optimizasyonu (PSO) yöntemleri uygulanmıştır. Literatürdeki genetik algoritma (GA) ve parçacık sürüsü optimizasyonu (PSO) yöntemleri üzerinde çeşitli modifikasyonlar gerçekleştirilerek, problemin çözümüne dair alternatif algoritmalar önerilmiştir. Tanımlanan problem ve senaryoları altında önerilen alternatif algoritmaların başarımlarının kıyaslanması gerçekleştirilmiş ve Frekans Atama Problemi'nde önerilen bu yöntemlerin kullanılabilirliği gösterilmiştir.

Şubat 2015, 98 sayfa

Anahtar Kelimeler: Çok Amaçlı Optimizasyon, Genel Atama Problemi, Frekans Atama Problemi, Genetik Algoritma, Parçacık Sürüsü Optimizasyonu, Pareto Optimizasyonu

ABSTRACT

Master Thesis

THE SOLUTION OF MULTI-OBJECTIVE OPTIMIZATION BASED GENERALIZED ASSIGNMENT PROBLEMS WITH METAHEURISTIC METHODS

Barış SATAR

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. A. Egemen YILMAZ

General assignment problem encountered in the real world, one of which the Frequency Assignment problem is non polynomial hard combinatorial optimization problem. These types of problems, because of solution of the time depending on the size of the problem increases exponentially with the limited time for the calculation can not be solved within, metaheuristics are used.

In this thesis the frequency assignment problem solution, different objectives conflict with each other by using metaheuristics to optimize, is searched. This multi-objective optimization problem is reduced to a single purpose, using a variety of scalarization approaches and metaheuristics methods, which are Genetic Algorithm and Particle Swarm Optimization, have been applied. Genetic algorithm (GA) and Particle Swarm Optimization which in the literature, performing various modifications on methods of problem solving that alternative applications have been produced. Under the identified problems and scenarios, comparison of the performance of alternative applications were carried out.

February 2015, 98 pages

Key Words: Multi-Objective Optimization, General Assignment Problem, Frequency Assignment Problem, Genetic Algoritmh, Particle Swarm Optimization, Pareto Optimization

TEŞEKKÜR

Tez çalışmasının her aşamasında, bilgisi ve tecrübesi ile bana yol gösteren, ortaya çıkan problemlerin aşılmasında yaptığı öneriler ile çalışmanın bu noktaya gelmesinde büyük payı olan danışmanım değerli hocam Sayın Doç. Dr. A. Egemen YILMAZ'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı) teşekkür ederim.

Tez çalışması süresince fikirleri ile katkıda bulunan Sayın Yrd. Doç. Dr. Ahmet AKBULUT'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı), Sayın Dr. Tolga NUMANOĞLU (ASELSAN A.Ş.) ve Sayın Güven YENİHAYAT'a (ASELSAN A.Ş.) teşekkür ederim.

Bu tez çalışması HBT-IA-2013-052 ve HBT-IA-2014-033 numaralı projeler kapsamında ASELSAN A.Ş. tarafından desteklenmiştir. Katkılarından dolayı teşekkür ederim.

2210 – Yurt İçi Yüksek Lisans Burs Programı çerçevesinde lisansüstü çalışmamı destekleyen TÜBİTAK'na teşekkür ederim.

Her zaman yanımda olan ve beni her durumda destekleyen canım annem Makbule SATAR ve babam Veli SATAR'a sonsuz teşekkürlerimi sunarım.

Barış SATAR
Ankara, Şubat 2015

İÇİNDEKİLER

TEZ ONAYI SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. KURAMSAL TEMELLER.....	4
3. MATERYAL ve YÖNTEM.....	6
3.1 Optimizasyon	6
3.2 Metasezgisel Yöntemler	8
3.2.1 Genetik algoritma.....	10
3.2.1.1 Seçim mekanizması	13
3.2.1.2 Çaprazlama işlemi.....	14
3.2.1.3 Mutasyon işlemi.....	15
3.2.1.4 Kontrol parametreleri	16
3.2.2 Parçacık sürüsü optimizasyonu	18
3.3 Çok Amaçlı Optimizasyon.....	23
3.3.1 Çok amaçlı optimizasyon terminolojisi.....	24
3.3.2 Çok amaçlı optimizasyon problemlerinde skalarizasyon yaklaşımları	28
3.3.2.1 Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı	29
3.3.2.2 Genelleştirilmiş Chebychev skalarizasyon yaklaşımı	33
3.3.2.3 Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımı	34
4. ARAŞTIRMA BULGULARI	39
4.1 Frekans Atama Problemi	39
4.2 Benzetim Sonuçları	45
4.2.1 Genetik algoritma.....	45
4.2.2 Parçacık sürüsü optimizasyonu	59
4.2.3 Frekans Atama Problemi'ne skalarizasyon yaklaşımları	82

5. TARTIŞMA ve SONUÇ	90
KAYNAKLAR	93
ÖZGEÇMİŞ.....	98

ŞEKİLLER DİZİNİ

Şekil 3.1 Optimizasyon yöntemlerinin sınıflandırılması	7
Şekil 3.2 Metasezgisel yöntemlerin farklı yönlerden sınıflandırılması	9
Şekil 3.3 Temel evrimsel döngü.....	12
Şekil 3.4 Örnek bir rulet çarkı.....	13
Şekil 3.5 Tek noktalı çaprazlama örneği.....	15
Şekil 3.6 Bit mutasyonu örneği.....	16
Şekil 3.7 Genetik algoritma akış diyagramı	17
Şekil 3.8 PSO konum ve hız güncellemesine ait bir örnek	20
Şekil 3.9 PSO akış diyagramı.....	22
Şekil 3.10 Pareto cephesi fiziksel anlamı.....	26
Şekil 3.11 İki amaçlı Pareto cephesi gösterimi	27
Şekil 3.12 Minimizasyon problemi için amaçlanan Pareto cephesi yayılımı	28
Şekil 3.13 a. Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı b. Konveks Pareto cephesinde elde edilen çözüm.....	31
Şekil 3.14 a. Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı b. Farklı ağırlık değeriyle konveks Pareto cephesinde elde edilen çözüm	31
Şekil 3.15 a. Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı b. Konkav Pareto cephesinde elde edilen çözüm	32
Şekil 3.16 a. Farklı ağırlık değeriyle ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı b. Farklı ağırlık değeriyle konkav Pareto cephesinde elde edilen çözüm	32
Şekil 3.17 Genelleştirilmiş Chebychev skalarizasyon yaklaşımı.....	34
Şekil 3.18 Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımı	36
Şekil 3.19 Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımında farklı ağırlık değeriyle konveks Pareto cephesinde elde edilen çözüm	38
Şekil 3.20 Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımında farklı ağırlık değeriyle konkav Pareto cephesinde elde edilen çözüm.....	38
Şekil 4.1 Frekans ataması gerçekleştirilen çevrim topolojisi.....	39
Şekil 4.2 Çevrim içi ceza fonksiyonu ceza puanı değerleri	41
Şekil 4.3 Çevrimler arası ceza fonksiyonu ceza puanı değerleri	43
Şekil 4.4 Çevrim içi, çevrimler arası ve kanal kazancı ceza fonksiyonu değerleri.....	44
Şekil 4.5 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	48
Şekil 4.6 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	48
Şekil 4.7 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	49

Şekil 4.8 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	49
Şekil 4.9 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	50
Şekil 4.10 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	50
Şekil 4.11 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	52
Şekil 4.12 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	52
Şekil 4.13 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	54
Şekil 4.14 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	54
Şekil 4.15 a. GA V1: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri.....	56
Şekil 4.16 a. GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	56
Şekil 4.17 a. GA V1 (b) GA V2 için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları.....	58
Şekil 4.18 Sınırsız YPSO algoritması yansıma geometrisi.....	62
Şekil 4.19 Sıralanmış Sınırsız YPSO algoritması sıralama geometrisi	64
Şekil 4.20 Yansımali YPSO algoritması yansıma geometrisi.....	66
Şekil 4.21 a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri.....	68
Şekil 4.22 a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	68
Şekil 4.23 a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	69
Şekil 4.24 a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	69
Şekil 4.25 a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	70
Şekil 4.26 a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	70

Şekil 4.27	a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	72
Şekil 4.28	a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	72
Şekil 4.29	a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	74
Şekil 4.30	a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	74
Şekil 4.31	a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Yansımali YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	76
Şekil 4.32	a. Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları.....	76
Şekil 4.33	a. Sınırsız YPSO (b) Yansımali YPSO için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları	78
Şekil 4.34	a. Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri b. Sıralamalı Sınırsız YPSO: Yineleme sayısı-En iyi amaç fonksiyonu değeri grafikleri	79
Şekil 4.35	a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı b.Sıralamalı Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları	80
Şekil 4.36	a.Sınırsız YPSO b. Sıralamalı Sınırsız YPSO için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları	81
Şekil 4.37	Kanal kazancı cezası ve çevrim içi ceza amaçları için oluşan Pareto cephesi.....	84
Şekil 4.38	Kanal kazancı cezası ve çevrimler arası ceza amaçları için oluşan Pareto cephesi.....	85
Şekil 4.39	Çevrim içi ceza ve çevrimler arası ceza amaçları için oluşan Pareto cephesi.....	86
Şekil 4.40	Normal ve dengelenmiş kanal kazancı cezası ve çevrim içi ceza amaçları için oluşan Pareto cepheleri.....	87
Şekil 4.41	Normal ve dengelenmiş kanal kazancı cezası ve çevrimler arası ceza amaçları için oluşan Pareto cepheleri.....	88
Şekil 4.42	Normal ve dengelenmiş çevrim içi ceza ve çevrimler arası ceza amaçları için oluşan Pareto cepheleri.....	89

ÇİZELGELER DİZİNİ

Çizelge 4.1	Çevrimler ve komşulukları.....	39
Çizelge 4.2	Genetik algoritma versiyonlarında kullanılan kontrol parametreleri	46
Çizelge 4.3	Problem tanımında ve algoritmada kullanılan sabitler.....	47
Çizelge 4.4	Kullanılmayan frekans sayısı	53
Çizelge 4.5	Çevrimlerde kullanılan frekans listesi.....	53
Çizelge 4.6	Kullanılmayan frekans sayısı	55
Çizelge 4.7	Çevrimlerde kullanılan frekans listesi.....	55
Çizelge 4.8	Kullanılmayan frekans sayısı	57
Çizelge 4.9	Çevrimlerde kullanılan frekans listesi.....	57
Çizelge 4.10	50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler.....	58
Çizelge 4.11	YPSO algoritmasına ait başlangıç parametreleri	60
Çizelge 4.12	Sınırsız YPSO algoritması	61
Çizelge 4.13	Sıralanmış Sınırsız YPSO algoritması	63
Çizelge 4.14	Yansımali YPSO algoritması	65
Çizelge 4.15	Kullanılmayan frekans sayısı	73
Çizelge 4.16	Çevrimlerde kullanılan frekans listesi.....	73
Çizelge 4.17	Kullanılmayan frekans sayısı	75
Çizelge 4.18	Çevrimlerde kullanılan frekans listesi.....	75
Çizelge 4.19	Kullanılmayan frekans sayısı	77
Çizelge 4.20	Çevrimlerde kullanılan frekans listesi.....	77
Çizelge 4.21	50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler.....	78
Çizelge 4.22	Kullanılmayan frekans sayısı	80
Çizelge 4.23	Çevrimlerde kullanılan frekans listesi.....	80
Çizelge 4.24	50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler.....	82

1. GİRİŞ

Genel atama problemi NP (Non-Polynomial) zor kombinatoryal bir optimizasyon problemidir (Fisher vd. 1986). Genel atama problemi en genel şekilde, n tane işin m tane temsilciye, temsilcilerin kapasite kısıtları göz önünde bulundurularak, minimum maliyet olacak şekilde atanması olarak tanımlanabilir.

Bu problem,

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1.1)$$

$$\text{Öyle ki } \sum_{j=1}^n a_{ij} x_{ij} \leq b_i \quad i = 1, \dots, m$$

$$\sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, n \quad (1.2)$$

$$x_{ij} \in \{0,1\} \quad \begin{array}{l} i = 1, \dots, m \\ j = 1, \dots, n \end{array}$$

Şeklinde formüle edilebilir. c_{ij} j işinin i temsilcisine atanma maliyeti, a_{ij} i temsilcisinin j işini yapabilmesi için gerekli olan kaynak değeri, b_i i temsilcisinin iş yapma kapasitesi olarak ifade edilir. x_{ij} ise i temsilcisi j işini yapıyorsa 1 diğer durumlarda 0 değerlerini almaktadır. Genel atama problemleri gerçek dünyada, araç rotalama problemleri (Fisher ve Jaikumar 1981), tesis yerleşimi (Mazzola 1989), frekans atama problemi (Hale 1980) vb. şeklinde karşımıza çıkmaktadır.

Araştırmacılar genel atama problemlerinin çözümü için 1960'ların sonundan beri çalışmakta ve pratik uygulamalar için 1970'lerin başlarından itibaren bilgisayar kodları geliştirmektedirler (Yagiura ve Ibaraki 2007).

Genel atama problemlerinin çözümünde kullanılan bazı yöntemlerin temeli Lagrangian gevşetmesine dayanmaktadır. Bu yöntemler arasında, Savelsbergh tarafından geliştirilen

“branch-and-price” algoritması (Savelsbergh 1997), optimal çözümü garanti eden Nauss tarafından sunulan “branch-and-bound” algoritması (Nauss 2003), sezgisel bir yöntem olarak verilen küme bölütleme (Cattrysse vd. 1994), tahmine veya olasılığa dayalı olmayan, belirsizlik içermeyen bir matematik ifade kullanılarak kesin sonuca ulaşan hesaplama macar algoritması (Kuhn 2005) gibi yöntemler bulunmaktadır.

Genel atama problemleri, çözüm zamanının problem boyutuna bağlı olarak üssel olarak artması, kat edilen toplam mesafe ve çalışma zamanı gibi performans ölçütlerine göre çözümü zor problemler sınıfına girmektedir. Problem boyutunun polinom fonksiyonu ile sınırlı olan hesaplama zamanı içinde çözülememesinden dolayı en iyi çözümü bulmayı garanti etmeyen, ancak çözüm zamanı, polinom sınırlar içinde kalan sezgisel optimizasyon metotlardan yararlanır (Şatır 2008). Optimizasyon, ulaşılması istenen amaç veya amaçlar için belirlenen kısıtlar doğrultusunda en iyi çözümü elde etme çalışmasıdır.

Genellikle pratik hayatta karşımıza çıkan atama problemleri çok amaçlı optimizasyon temeline dayanmaktadır. Optimizasyon işlemine bir amacın daha eklenmesi problemi daha karmaşık hale getirmektedir. Bu şekilde, çoğunlukla da birbirleriyle çelişmekte olan birden fazla amacın (bir sistemde performansın artırılması ve maliyetin azaltılması, vb.) aynı anda optimize edilmesine çok amaçlı optimizasyon (*multi-objective optimization*) adı verilir.

Bu tezde Frekans Atama Problemi’nin tüm girdi, çıktı, kısıt ve amaç fonksiyonları ile birlikte tanımlanması, literatürdeki ilgili problemin incelenmesi, problemin çözümüne dair literatürdeki algoritmalarla bakılması, belirlenen ve uyarlanan algoritmaların uygulamalarını gerçekleştirilmesi, tanımlanan senaryolar üzerinden alternatif uygulamaların başarımlarının değerlendirilmesi adımları gerçekleştirilmiştir.

Tez kapsamında çok amaçlı optimizasyon temelli olacak şekilde belirlenen bir genel atama problemi olan Frekans Atama Problemi’nin metasezgisel yöntemlerle çözümü konusunda bir çalışma yapılmıştır. Çalışmada popülasyon temelli yöntemler olan genetik algoritma (Holland 1992, Koza 1992), parçacık sürüsü optimizasyonu

(Kennedy ve Eberhart 1995, Shi ve Eberhart 1998) yöntemlerinin Frekans Atama Problemi'ne uygulamaları gerçekleştirilip, yöntemler içinde kullanılabilecek yeni operatör işlemleri geliştirilerek, atama problemi çözümündeki performansları incelenmiştir.

2. KURAMSAL TEMELLER

Frekans atama problemleri, ilk kez 1960'larda ortaya atılmıştır (Aardal vd. 2001). Kablosuz iletişim teknolojisi gelişmesi ve GSM operatörleriyle artışıyla beraber frekans atama problemleri daha bir önem kazanmıştır. Frekans atama problemi GSM operatörlerinin farklı kaygılarının giderilmesi amacıyla yola çıkılarak bugüne kadar taşınmış bir NP zor kombinatorial problemidir (Hale 1980, Smith ve Hurley 1996). Bu problem, literatürde (Hale 1980, Hurley ve Smith 1997, Koster 1999, Cabon vd. 1999, Eissenblatter 2001) tanımları farklı şekilde karşımıza çıkmaktadır.

Eldeki frekans setini, minimum girişime sebep olacak şekilde kullanıcılar arasında paylaşırması problemi, Minimum Girişimli Frekans Atama Problemi (MI-FAP) olarak adlandırılmaktadır. Kabul edilebilir girişim seviyesini aşmamak kaydıyla, minimum boyutta bir frekans seti kullanacak şekilde kullanıcı arası frekans paylaşımı yapmaktır. Kullanılan frekans sayısının göz önünde bulundurulduğu versiyona Minimum Dereceli Frekans Atama Problemi (MO-FAP) adı verilmekte; ayrılan frekans aralığının içerisinde kullanılmayan frekans olsa bile tüm aralığın ücretlendirildiğinin varsayıldığı versiyon ise Minimum Genişlikli Frekans Atama Problemi (MS-FAP) olarak isimlendirilir.

Problemdeki tüm kısıtların katı kısıt olarak ele alındığı ve geçerli bir çözüm olup olmadığının incelenmesi durumunda ise eldeki problem Fizibilite Frekans Atama Problemi olarak anılmaktadır.

Olabildiğince çok sayıda frekansın atanmasına çalışılması ise Maksimum Hizmet Frekans Atama Problemi olarak bilinmektedir. Olabildiğince az sayıda girişim bakımından problem çıkaran frekansın atanmasına çalışılması ise Minimum Bloklayan Frekans Atama Problemi olarak adlandırılmaktadır.

Frekans Atama Problemleri'nin farklı varyantlarının farklı buluşsal yöntemler ile çözümüne dair literatürde birçok çalışma bulunmaktadır.

Lokal Arama Yaklaşımına Dayalı Çalışmalar: Problemin MI-FAP versiyonunun çözümü Park ve Lee (1996), WLAN'lara yönelik dinamik MI-FAP versiyonunun çözümü ise Mishra vd. (2005) tarafından gerçekleştirilmiştir. Tsang ve Voudouris (1998), yerel optimumlara takılmayı engelleyecek bir takım önlemler aldıkları ve Yönlendirilmiş Lokal Arama adını verdikleri algoritma ile MI-FAP, MO-FAP ve MS-FAP çözümlerinin nasıl yapılabileceğini göstermişlerdir.

Tabu Arama Yaklaşımına Dayalı Çalışmalar: Tabu Arama yaklaşımı, Lokal Arama yaklaşımına benzer bir şekilde, çözüm uzayında makul ve mantıklı bir noktadan hareket ederek ve adından da anlaşılacağı üzere “Tabu” olarak adlandırılan noktalardan uzak durarak çözüm arayan bir yöntemdir. Lokal Aramadan en büyük farkı, her bir denemede bir öncekinden daha iyi çözüm bulmak zorunluluğu olmamasıdır. MS-FAP'in bu yöntemle çözümü Costa (1993) ile Hao ve Perrier (1999) tarafından gerçekleştirilmiştir. MI-FAP'in çözümü ise Castelino vd. (1996) tarafından yapılmıştır.

Genetik Algoritma Yaklaşımına Dayalı Çalışmalar: MS-FAP'in Genetik Algoritma ile çözümü Valenzuela vd. (1998) tarafından, MO-FAP'in çözümü ise Crompton vd. (1994), Kapsalis vd. (1995), Dorne ve Hao (1995) tarafından gerçekleştirilmiştir. MI-FAP'in çözümü ise Cuppini (1994), Lai ve Coghill (1996), Ngo ve Li (1998), Crisan ve Mühlenbein (1998), Jaimes-Romero vd. (1996) tarafından gerçekleştirilmiş olup; MS-FAP'in çözümü ise Beckmann ve Killat (1999) tarafından sunulmuştur. Bütün bu çalışmalar, çözüm temsil etme teknikleri ve uygulanana genetik operatörler açısından farklılıklar arz etmektedir.

Karınca Kolonisi Optimizasyonu Yaklaşımına Dayalı Çalışmalar: MI-FAP'in, daha yakın tarihte geliştirilmiş olan bu yöntemle çözümü Maniezzo ve Carbonaro (2000), tarafından gerçekleştirilmiş olup; problemin “çoklu girişim” tanımına dayalı değişik bir versiyonu ise Montemanni vd. (2002) tarafından ele alınmıştır.

3. MATERİYAL ve YÖNTEM

3.1 Optimizasyon

Optimizasyon, ulaşılması istenen amaç veya amaçlar için belirlenen kısıtlar doğrultusunda en iyi çözümü elde etme şeklinde tanımlanabilir. Genel bir optimizasyon problemi matematiksel olarak,

$$\begin{aligned} \min \quad & f(x_i) \quad i = 1, \dots, n \\ & g(x_i) \geq 0 \end{aligned} \tag{3.1}$$

şeklinde tanımlanır. Burada f fonksiyonuna amaç fonksiyonunu, x karar değişkenini, g fonksiyonu ise kısıtlayıcıları ifade etmektedir. Bir problemde birden fazla uygun çözüm olabilir, ancak bu çözüm alternatiflerini kıyaslayacak ölçütlerin olması gerekmektedir. İşte bu ölçütler amaç fonksiyonudur ve problemdeki isterlere göre maksimumu ya da minimumu (maliyetin azaltılması, performansın artırılması vb.) aranır. Kısıt fonksiyonu ise karar değişkenlerine sınır koyan fonksiyonlardır, bu fonksiyonlara göre karar değişkenlerinin alacağı değerler limitlendirilirler.

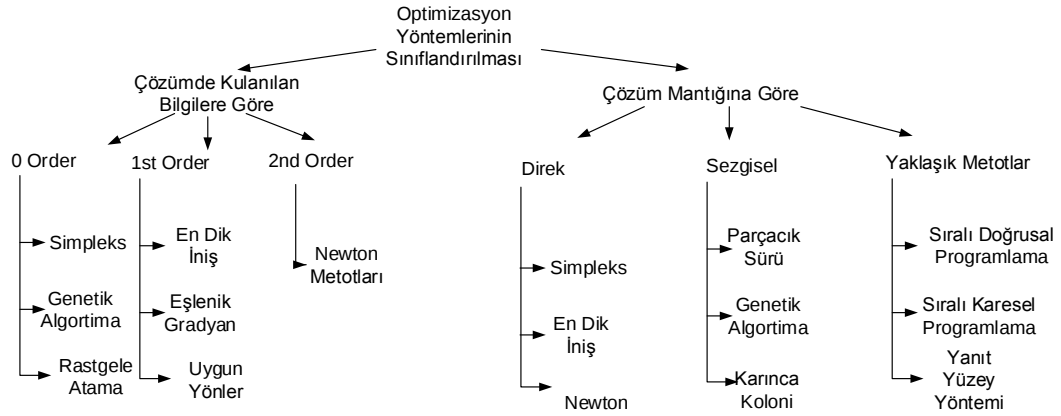
Optimizasyon kelime anlamı “En İyileme” olarak bilinmektedir. Optimizasyon ile mevcut çözümler arasından en iyi çözüm ortaya çıkarılmak istenmektedir. Çıkarılmak istenen bu çözüme, optimum çözüm adı verilmektedir ve amaç her zaman bu çözüme ulaşabilmektir.

Klasik optimizasyon teorisi Cauchy, Lagrange ve Newton tarafından geliştirilmiştir. Newton ve Leibnitz’in analiz çalışmaları, optimizasyonun diferansiyel hesaplama yöntemlerinin geliştirilmesine katkı sağlamıştır. Kısıtlı problemler için optimizasyon yöntemini, kendi adı verilen Lagrange geliştirmiştir. Kısıtsız optimizasyon problemlerini çözmek için “En Dik İniş” yönteminin ilk uygulaması da Cauchy tarafından yapılmıştır.

Optimizasyon konusundaki bu çalışmalar 20. yüzyılın ortalarına kadar çok yavaş ilerlemiştir. 1950'lerden sonra sayısal bilgisayarların icadı optimizasyonda çok büyük çalışmaları beraberinde getirerek, birçok yeni teori ve metodun ortaya çıkmasını sağlamıştır. 1980'lerde kişisel bilgisayarların yaygınlaşması ile optimizasyon metotları yaygınlaşmaya başlamış ve 2000'lere gelindiğinde optimizasyon uygulamaları çok geniş alanlarda uygulanabilirlik kazanmıştır (Rao 2009).

Optimizasyon probleminin çeşitli yönlerden sınıflandırılması gerçekleştirilebilir. Sınıflandırmalarda kullanılan karakteristikler, karar değişkenleri sayılarına göre; tek boyutlu, çok boyutlu olarak yazılır. Problemden boyut arttıkça çözümün daha zorlaşır. Karar değişkenleri türüne göre; sürekli, kesikli olarak ayrılabilir. Amaç ve kısıt fonksiyonlarına göre; doğrusal, kuadratik ve doğrusal olmayan olarak sıralanabilir. Amaç ve kısıt fonksiyonları bükümlü bir fonksiyon şeklinde de ifade edilebilir. Bu doğrusal olmayan problemin özel bir durumu gibidir. Problemin formülasyonuna göre ise; kısıtlı ve kısıtlı olmayan olarak ayrılabilir (Rao 2009).

Optimizasyon yöntemlerinin sınıflandırılması da mümkündür.



Şekil 3.1 Optimizasyon yöntemlerinin sınıflandırılması

Çözümde fonksiyonun kendi değeri kullanılıyorsa “0 Order”, fonksiyonun birinci türevi kullanılıyorsa “1st Order” ve fonksiyonun ikinci türevi kullanılıyorsa “2nd Order” yöntemlere başvurulur. Çözüm mantığına göre bakıldığında ise eğer problemi olduğu gibi çözmeye çalışıyorsa direk, problemi matematiksel temelden çok doğadan

esinlenerek çözülyorsa sezgisel, amaç ve kısıt fonksiyonları için daha basit yaklaşıklığı elde edilecekse yaklaşık metotları kullanılır (Kurtaran 2005).

Optimizasyon probleminin çözümünde ilk olarak karar değişkeninin belirlenmesi gerçekleştirilip, ardından probleme ait amaç fonksiyonu oluşturulmalıdır. Eğer varsa problemin kısıt fonksiyonu tanımlanıp, optimizasyon sınıflandırılmasına göre seçilecek olan uygun çözüm yönteminin uygulaması gerçekleştirilmelidir. Optimizasyonda yapılan çalışmalar çoğunluğu tek amaç üzerine olmasına rağmen pratikte karşımıza çıkan optimizasyon problemleri çok amaçlı optimizasyon temeline dayanmaktadır.

3.2 Metasezgisel Yöntemler

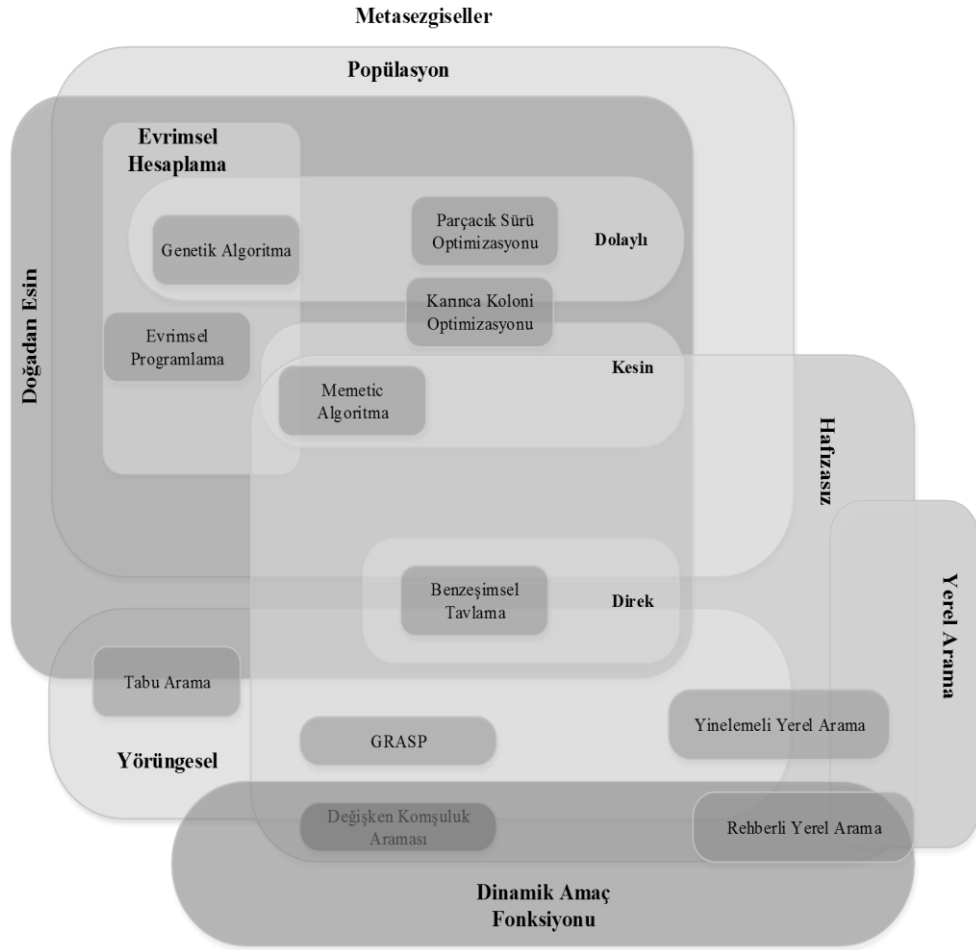
Gerçek dünya optimizasyon problemlerinin birçoğu karmaşık ve çözümü zor problemlerdir. Böyle problemlerin kabul edilebilir bir zaman diliminde kesin olarak çözümleri mümkün değildir. Bu yüzden bu problemlerde sonuca ulaşmak için yaklaşık çözüm veren algoritmalar kullanma yoluna gidilmiştir. Yaklaşık çözüm veren algoritmalar sezgisel ve metasezgisel yöntemler olarak ayrılabilir. Eğer gerçek dünyada karşılaştığımız probleme özgü, o problemin girdilerinden yararlanarak ve sadece o problemde iyi sonuçlar vermesi beklenen bir algoritma üretilmiş ise algoritma sezgisel algoritma olarak adlandırılır. Metasezgiseller, klasik optimizasyon yöntemlerinin kabul edilebilir bir çözüm üretmediği kombinatoriyal optimizasyon problemleri için sınırlı bir zaman içerisinde etkin ve uygun çözümler üretebilen yaklaşık algoritmalarıdır. Metasezgisellerde, sezgisel algoritmalar da olduğu gibi sadece probleme özgü bir başarı söz konusu değildir, problemden bağımsız olarak etkin ve uygun çözümlere ulaşabilmektedir.

Metasezgisel yöntemler kuvvetli algoritmalarıdır ve çeşitlilik gösterirler. Çeşitlilik göstermesi, arama uzayının farklı noktalarında çözümü araştırabilmesi ve etkin, uygun çözümler bulunmasını sağlar. Kuvvetli algoritmalar olması ise bu arama uzayında yerel aramada yapabilmesidir. Bu yöntemler, her zaman global en iyi çözümün bulunmasını garantileyemeseler de, büyük ölçekli ve karmaşık problemlere etkin ve uygun çözümler üretebildikleri için oldukça elverişli olmaktadır (Talbi 2009, Önder 2011).

Metasezgisel algoritma akışı genel olarak dört adımda özetlenebilir:

- Rastgele seçilen bir çözümden algoritmanın başlatılması.
- Algoritma kontrol parametreleri altında birçok komşu çözümlerin araştırılması.
- Kabul edilmiş kurallara göre anlık durumun yenilenmesi.
- Kontrol parametrelerinin ayarlanması ve kabul edilmiş kurallara uyana kadar akışın tekrar edilmesi (Eker 2005).

Metasezgisel yöntemler, farklı yönler göz önünde bulundurularak çeşitli şekillerde sınıflandırılabilir.



Şekil 3.2 Metasezgisel yöntemlerin farklı yönlerden sınıflandırılması (Dreo 2011)

Şekil 3.2’den de görüldüğü üzere sınıflandırma, doğadan esinlenen ve esinlenmeyen metasezgisel yöntemler; popülasyon temelli ve yörüngesel metasezgisel yöntemler;

dinamik ve sabit amaç fonksiyonlu yöntemler; hafızalı ve hafızasız metasezgiseller olarak yapılabilmektedir.

3.2.1 Genetik algoritma

Genetik algoritmalar doğal seçim ve doğal genetik mekanizmasını temel alan bir arama algoritmasıdır (Goldberg 1989). Genetik algoritmanın doğru anlaşılabilmesi için Darwin tarafından üretilen evrimsel teorinin incelenmesi gerekmektedir. Doğal seçim, bir evrimsel mekanizma olup dış çevreyle daha uyumlu özellikleri olan organizmaların diğer organizmalara göre doğada yaşamlarını sürdürme ve üreme olasılıklarının daha fazla olduğunu anlatmaktadır. Böylece dış çevreyle daha uyumlu olan bireyin özellikleri yeni kuşaklara aktarılmakta ve uyum açısından zorluk yaşayan bireylerin popülasyondan çıkarılması sağlanmaktadır. Süreç içerisinde organizmaların çevreyle uyumsuz olan özellikleri ayıklanarak çevreyle uyum sağlamakta güçlü bireylerden oluşan bir popülasyona ulaşılabilmektedir.

Genetik algoritma, belirli bir alandaki sorunlara çözüm adayları taşıyan popülasyon yapıları sağlayan yinelemeli bir işlemler bütünüdür. Evrim mekanizmasına yani en iyinin hayatta kalması ilkesine dayanmaktadır.

Genetik algoritmalar gibi temeli evrimsel mekanizmasına dayanan algoritmalar ile zamanla beraber daha da zorlaşan ve kısıtları daha da artan optimizasyon problemlerinin çözümünün basitleştirilmesi ve hızlandırılması istenmiştir. Genetik algoritmalarla ilgili 1950 ve 1960'larda bilim insanlarının yaptıkları çalışmalar olsa da genetik algoritmaların temel prensiplerinin ortaya konulması Holland tarafından 1975 yılında gerçekleştirilmiştir.

Genetik algoritmaların büyük ölçekli kompleks problemlerin çözümünde tercih edilmesindeki nedenler aşağıdaki gibi sıralanabilir.

- Çevre şartlarından az etkilenen arama tekniğidir.

- Genetik algoritmalar makul süre içerisinde optimale yakın sonuçlar üretmektedir.
- Genetik algoritmalar paralel işleme için uygundur.
- Genetik algoritmalar, amaç fonksiyonlarına bağımlıdır, amaç fonksiyonları popülasyondan daha iyi sonuçların üretilmesinde katkı sağlarlar ancak gürültülü de olsa amaç fonksiyonlarını kullanabilirler.
- Genetik algoritmaların geliştirilmesi kolaydır.
- Problem uzayı için herhangi bir varsayıma ihtiyaç duymaz (Goldberg 1989, Wainright 1993).

Genetik algoritmalar, çok büyük kompleks problemlerin çözümünde diğer klasik yöntemlere göre başarısını kanıtlamıştır. Çözülen bu problemler büyük ölçekli, kompleks arama uzayında gerçek değerli parametre kestirimi gerçekleştirilen ve yerel minimum noktalarıyla dolu olabilmektedir. Genetik algoritmaların bu tür problemlerin üstesinden gelebildiğinin bilinmesi, bilim insanlarını çok çeşitli alanlarda bu algoritmayı kullanmasına yöneltmiştir. Bu alanlar görüntü işleme, lazer teknolojileri, yarı iletken fiziği, robotik, yapay sinir ağları, kontrol vb. olarak yazılabilir (Coley 1991).

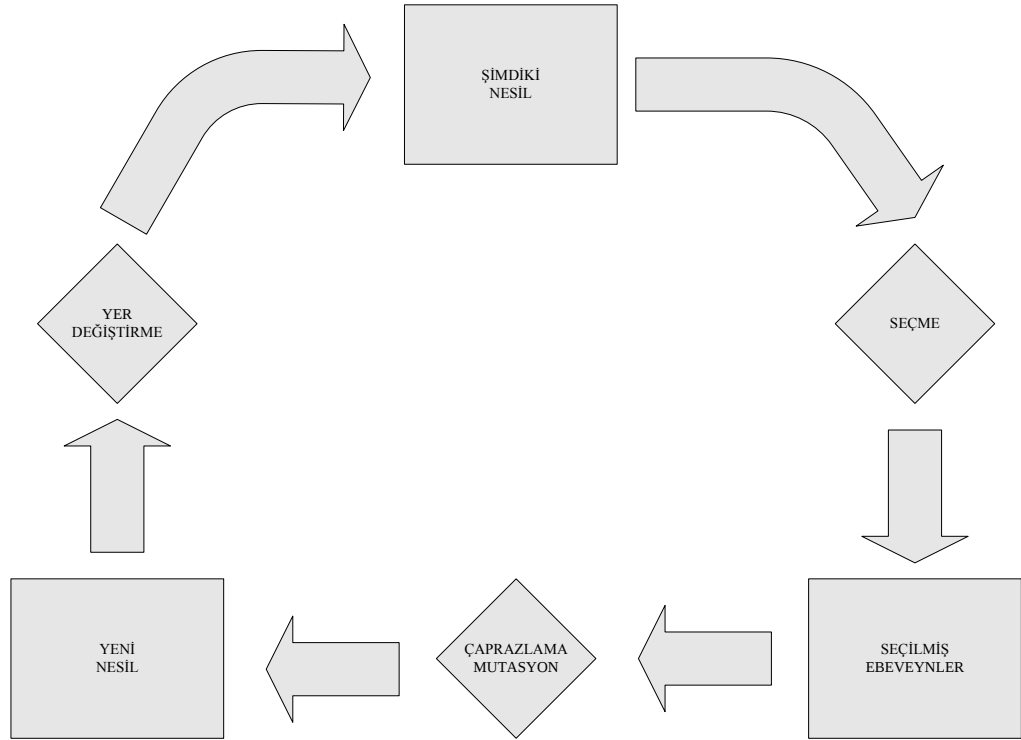
Genetik algoritmalarının klasik optimizasyon yöntemlerine göre farklılaştığı noktalar vardır.

- Genetik algoritmalar gerçek parametreler üzerinden çok kodlanmış parametreler üzerinde çalışırlar.
- Genetik algoritmalar tek bir nokta yerine popülasyon dizilerinden arama işleminin gerçekleştirirler.
- Genetik algoritmalar yalnızca amaç fonksiyonu bilgilerine ihtiyaç duyarlar.
- Genetik algoritmalar deterministik kurallar yerine olasılıksal geçiş kurallarını kullanırlar.
- Genetik algoritma çözüm uzayının belirli bir kısmını tararlar. Bu sayede, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar (Goldberg, 1989).

Genetik algoritmalarla biyolojideki karşılıkları ile bir analogi kurulacak olunursa; ele alınan her bir çözüm parametresi geni, tüm bu çözüm parametreler bütününe oluşturduğu küme ise kromozomu ifade etmektedir. Oluşturulan her bir kromozom,

özellikli bir alandaki probleme ait çözüm adaylarıdır. Bu çözüm adayları, kromozomlar, ise çözümler kümesini yani popülasyonu ortaya çıkarmaktadırlar.

Genetik algorithmadaki her bir nesildeki popülasyona ait her bir kromozomun, problemin çözümü için sağladığı uygunluk hesaplanmakta ve buradan çıkacak olan sonuç yardımıyla yeni popülasyon için aday çözümler; çoğalma, çaprazlama, mutasyona uğrama vb. genetik operatörler yardımıyla düzenlenir (Grefenstette 1986).



Şekil 3.3 Temel evrimsel döngü

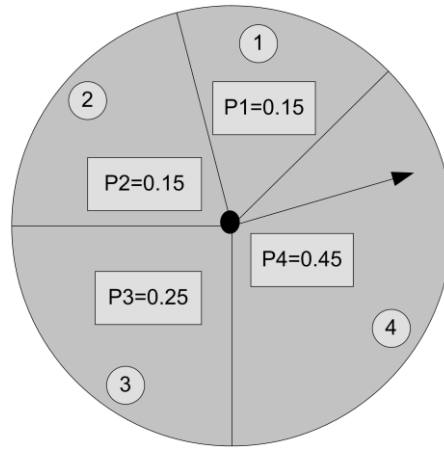
Şekil 3.3’de temel bir evrimsel döngü görülmektedir. İlk nesil için rastgele kromozomlar oluşturulmakta, bu rastgele oluşturulan kromozomlar içerisinde problemin çözümüne ait uygunluk dereceleri dikkate alınarak uygunluk derecesi yüksek olan ebeveynler seçilir. Seçilmiş ebeveynlerden çoğalma, çaprazlama, mutasyon gibi çeşitli genetik varyasyonlarla yeni bir nesil oluşturulmakta ve bu oluşturulan yeni nesil aynı işlemlere maruz bırakılmak üzere eski nesil ile yer değiştirmektedir.

3.2.1.1 Seçim mekanizması

Seçim mekanizması her bir bireye ait uygunluk derecesi göz önüne alınmasıyla beraber başlatılmaktadır. Kromozomun uygunluk derecesi ile seçim mekanizması, popülasyon içerisinde hangi kromozomların üreme havuzunda daha yüksek olasılıkla bulunacağını göstermektedir. Birden fazla seçim mekanizması bulunmaktadır, çalışmalarda çok sık olarak kullanılan rulet çarkı mekanizması ve turnuva seçimi mekanizmalarından bahsedilecektir.

Rulet çarkı mekanizması uygunluk fonksiyonu orantılı bir seçme mekanizmasıdır ve uygunluk fonksiyonu değerlerine bağlı bir olasılık dağılımına göre ebeveyn seçme işlemini gerçekleştirmektedir.

Rulet çarkı mekanizmasında öncelikle popülasyon içerisindeki her bir kromozom için uygunluk fonksiyonu değeri hesaplanır. Ardından bu hesaplanan bütün uygunluk fonksiyonu değerleri toplanarak popülasyonun toplam uygunluk fonksiyonu değerine ulaşılır. Bu aşamadan sonraki adım ise uygunluk fonksiyonlarına göre bir olasılık dağılımı çıkarmaktır. Bu dağılım, her bir kromozomun uygunluk fonksiyonu değerinin popülasyonun toplam uygunluk fonksiyonu değerlerine bölünmesiyle elde edilir. Olasılıksal dağılımın hesaplanmasındaki son evre kromozomların kümülatif olasılıklarının hesaplanmasıdır.



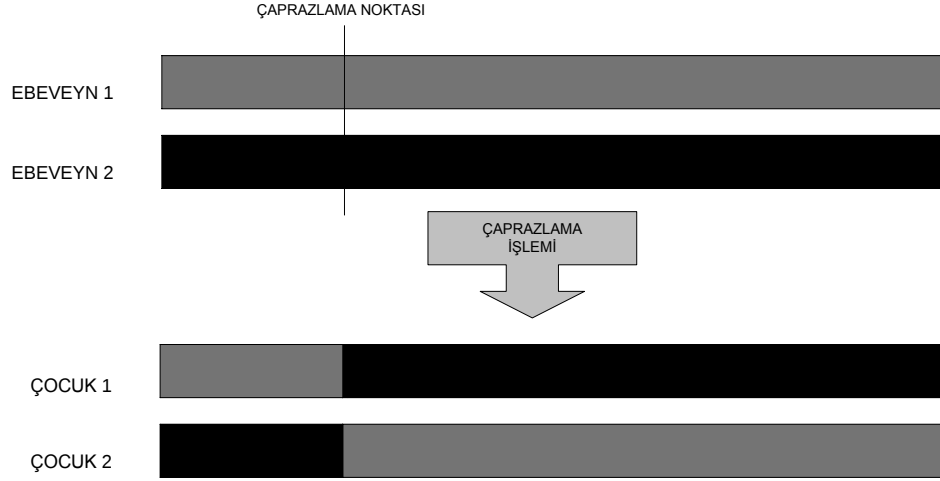
Şekil 3.4 Örnek bir rulet çarkı

Şekil 3.4’de olasılıksal dağılımı hesaplanan dört kromozoma sahip bir popülasyon için rulet çarkı örneği verilmiştir. Görüldüğü gibi aralıkların toplam değeri 1’e eşittir. Seçme mekanizmasında şekil 3.4’deki kadran, kromozom sayısı kadar döndürülür ve her bir döngüde 0 ile 1 aralığına denk düşen bir sayı üretilir. Her bir sayı aralığına denk düşen kromozom, ebeveyn olarak üreme havuzuna alınır.

Turnuva seçimi mekanizması, verimliliği ve kolay uygulanır olması sebebiyle genetik algoritmada kullanılan en popüler seçme mekanizmalarından biridir (Goldberg ve Deb 1991). Mekanizma, ilk olarak popülasyon içerisindeki her bir kromozom için uygunluk fonksiyonu değerinin hesaplanmasıyla başlatılır. Ardından popülasyon içerisinde bulunan kromozom sayısı kadar popülasyon içerisinden rastgele kromozomlar seçilerek, gruplar oluşturulur. Gruplar içerisinden rastgele seçilecek olan birkaç adet kromozom, uygunluk fonksiyonu değerlerine göre yarıştırlarak seçilmiş ebeveyn olup olmadıklarına karar verilmektedir. Her bir grup içerisinde kazanan kromozomlar seçilmiş ebeveynler olarak genetik operasyonlara hazır hale getirilirler.

3.2.1.2 Çaprazlama işlemi

Çaprazlama, seçilmiş ebeveynler arasında gen değişimi işlemini gerçekleştirmektedir. Çaprazlama operatörü ile kromozomlar arasında bilgi alışverişinde bulunarak arama uzayı içerisinde fakat daha önce çözüm aranmamış bölgelere de ulaşp yeni çözümler üretilmesi amaçlanmaktadır (Booker 1987). Çaprazlama mekanizmasında, öncelikle üreme havuzundaki seçilmiş ebeveynler arasından rastgele iki kromozom seçilir ve ardından kromozomların rastgele seçilmiş bir bölümünden kesilip, kromozomlar arası gen alışverişinde bulunularak yeni kromozomlar oluşturulur.



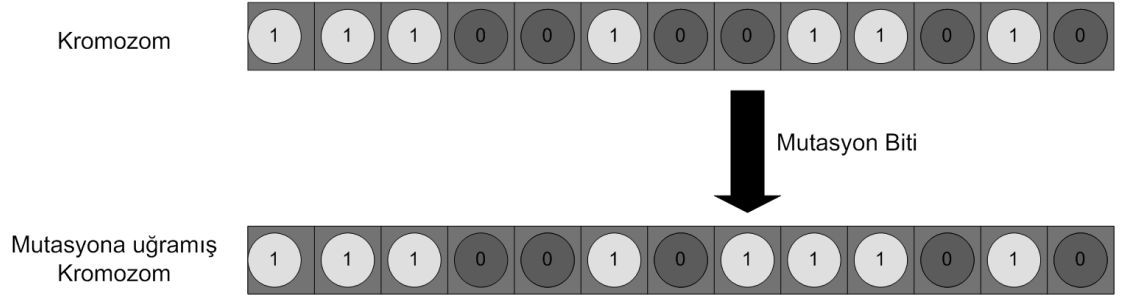
Şekil 3.5 Tek noktalı çaprazlama örneği

Şekil 3.5’de tek noktalı çaprazlama örneği verilmiştir. Çaprazlama işlemi sonucunda iki kromozomdan, seçilen kromozomlardan daha iyi özelliklerde olmaları beklenen iki yeni kromozom üretilmiştir.

Çaprazlama işlemi aynı zamanda çift noktalı olacak şekilde de yapılabilir. Bu durumda yapılacak işlemin tek noktalı çaprazlamadan tek farkı kromozomların rastgele seçilen iki noktadan kesilerek gen yer değişiminin gerçekleştirilmesinin sağlanmasıdır.

3.2.1.3 Mutasyon işlemi

Mutasyon, genetik bir operatör olup, yapılan rastgele değişimlerle arama bölgesinde yeni çözüm noktalarına ulaşılması konusunda katkı sağlamaktadır (Man vd. 1996). Mutasyon işlemi, genetik algorithmadaki kontrol parametrelerinden mutasyon oranı değerine göre belirlenmektedir. Seçilen kromozomda rastgele seçilen bir gen için değişiklik yapıp yapılmayacağına mutasyon oranına bakılarak karar verilir. 0 ile 1 arasında rastgele belirlenen sayı, mutasyon oranı değerinden küçükse o gen mutasyona uğrar.



Şekil 3.6 Bit mutasyonu örneği

Mutasyon işlemi, kaybedilen bilgi yerine koyulması ya da kromozoma yeni bilgi eklenmesi gibi durumları sağlayabilir. Çeşitliliğin azaltılmaması adına daha az sıklıkta yapılmaktadır.

Mutasyon, genetik algoritmalarda son yinelemelere gelindiğinde etkinliği artmaktadır. Çünkü yinelemelerde artık sona gelindiğinde iyi kromozomlar birbirlerine benzemeye başlamakta, böylelikle çaprazlama işleminin arama bölgesi kısıtlanmaktadır. Mutasyon işlemi ile beraber gen değişikliğiyle arama bölgesinin farklı noktalarında da çözüme bakılması sağlanmaktadır (Altıparmak 1996).

3.2.1.4 Kontrol parametreleri

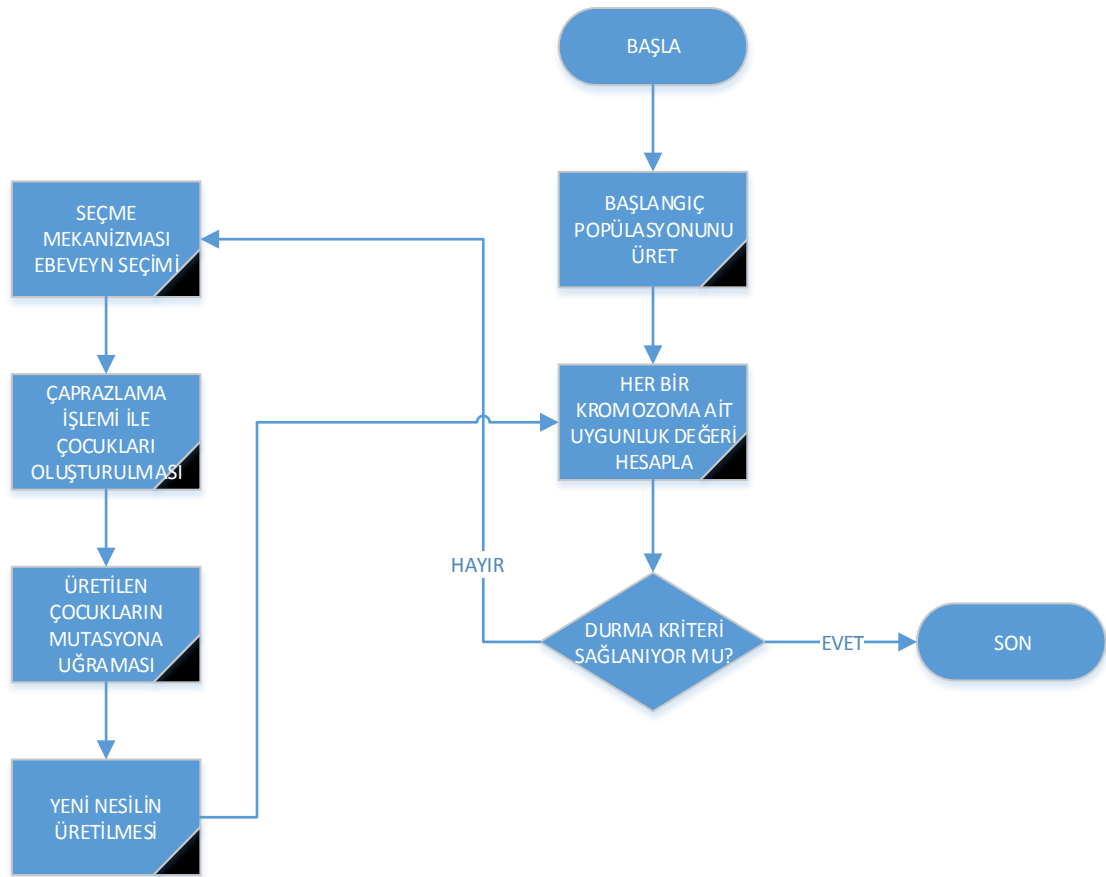
Genetik algoritma fonksiyonu, arama bölgesi içerisinde yeni bölgelerin keşfedilmesi ile hali hazırda örneklenmiş arama bölgesinde çalışılması arasında bir denge kurmaktadır. Genetik algoritmanın performansında önemli bir yer tutan bu denge, kontrol parametreleri olarak bilinen popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranı değerlerinin doğru seçilmesi ile sağlanmaktadır (Srivinas ve Patnaik 1994).

Popülasyon büyüklüğü, doğru belirlenmesi problemin çözümüne ulaşılması için kritiktir. Eğer popülasyon büyüklüğü olması gerektiğinden küçük seçildiği takdirde çözüm uzayında arama yapılırken lokal optimumlara takılınması kaçınılmaz olabilir. Popülasyon büyüklüğü çok yüksek seçildiğinde ise çözüm uzayı yeterince örneklendiği

için daha iyi bir arama gerçekleştirilir ancak bu durum çözüme ulaşma süresinin artmasına yol açabilmektedir.

Çaprazlama oranı, seçilmiş ebeveynlerin çaprazlama işlemine hangi sıklıkla maruz kalacağını belirlemektedir. Çaprazlama oranının çok yüksek seçilmesi iyi çözüm dizilerinin bozulmasına neden olmaktadır.

Mutasyon işlemi, genetik algoritmadaki kontrol parametrelerinden mutasyon oranı değerine göre uygulanmaktadır. Mutasyon olasılığı çok yüksek seçilmesi durumunda genetik arama rastsal bir aramaya dönüşür. Fakat bu durum, kaybedilen bilginin tekrar yerine konulmasında yardımcı olmaktadır.



Şekil 3.7 Genetik algoritma akış diyagramı

Şekil 3.7’de Genetik Algoritmaya ait basit bir akış diyagramı verilmektedir. Algoritmayı çalıştırmaya başlarken, genetik algoritmaya ait kontrol parametrelerinin

probleme özgü olarak belirlenmesi gerekmektedir. Bu parametreler belirlendikten sonra popülasyon büyüklüğü kadar çözümü içerisinde barındıran başlangıç neslinin rastsal olarak üretilmesi gerçekleştirilecektir. Algoritmada bir sonraki aşama başlangıç popülasyonuna ait her bir kromozomun uygunluk fonksiyonlarının hesaplanmasıdır. Hesaplanan uygunluk fonksiyonu değerleri, belirlenen durma kriterini sağlamıyorsa seçme mekanizması çalıştırılarak çaprazlama ve mutasyon işlemlerinin uygulanacağı kromozomlar seçilmektedir. Ardından genetik çeşitliliğin oluşmasını sağlayan sırasıyla çaprazlama ve mutasyon mekanizmaları çalıştırılır ve yeni nesil üretilir. Algoritma bu adımdan sonra üçüncü adıma, uygunluk fonksiyonlarının hesaplanması adımı geri döner. Döngü uygunluk fonksiyonları durma kriterini sağlayıncaya kadar devam eder.

3.2.2 Parçacık sürüsü optimizasyonu

Parçacık sürüsü optimizasyonu, doğadaki balık ve kuş sürülerinden esinlenerek 1995 yılında Kennedy ve Eberhart tarafından geliştirilen sürü tabanlı bir algoritmadır (Kennedy ve Eberhart, 1995). Algoritmanın temelini çeşitli hayvanların sürü davranışları oluşturmaktadır. Sürü hayvanlarının yaşamsal ihtiyaçlarını gidermek için ortaya koydukları davranışsal hareketler üzerine algoritma inşa edilmiştir. Sürüde bulunan hiçbir bireyin en iyi kaynağın yerini bilmediği durumda bile sürünün başarılı bir şekilde kaynağa ulaşabilmesinden yola çıkarak geliştirilen bu yöntem; her bir bireyin kendi hatıralarında yer etmiş olan iyi konuma gitme eğilimi olarak tanımlanabilecek bilişsel davranış biçimi, her bir bireyin iyi konumlarda bulunan diğer bireyleri takip etme eğilimi olarak tanımlanabilecek sosyal davranış biçimi, her bir bireyin rastgele olarak arama yapma eğilimi olarak tanımlanabilecek keşifsel davranış biçimleri arasında bir denge olduğu varsayımına dayanmaktadır (Eberhart ve Kennedy 1995, Altınöz ve Yılmaz 2009).

PSO algoritmasında kullanılan parçacık kavramı, sürüde bulunan her bir bireyi temsil etmektedir. Sürüdeki her bir birey, d boyutlu çözüm uzayında bulunduğu noktayı belirten pozisyon ve d boyutlu çözüm uzayındaki hareketini gösteren hız bilgilerine sahiptir. PSO yinelemeli yapıda olan bir algoritmadır. Dolayısıyla her bir yinelemede parçacığa ait hız ve pozisyon bilgileri güncellenmektedir. Parçacığın hız güncellemesi

gerçekleştirilirken, bir önceki adımdaki keşifsel hız bilgisinin yanı sıra bilişsel ve sosyal tecrübelerinden yararlanılmaktadır. Parçacığın güncel pozisyon bilgisi için de parçacığın bir önceki adımındaki pozisyon bilgisi ile güncellenmiş hız bilgisi kullanılmaktadır.

$$v_i^d(k+1) = w v_i^d(k) + c_1 r_1 [x_{pbest_i}^d - x_i^d(k)] + c_2 r_2 [x_{gbest}^d - x_i^d(k)] \quad (3.2)$$

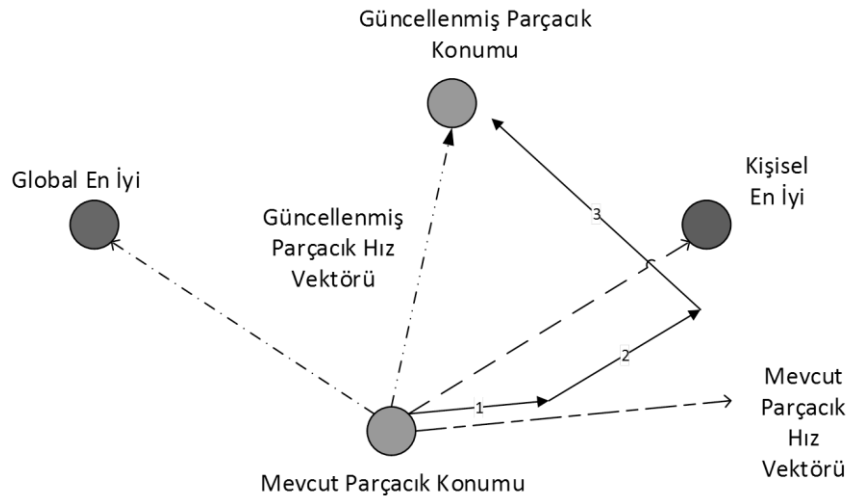
$$x_i^{(d)}(k+1) = x_i^{(d)}(k) + v_i^{(d)}(k) \Delta t \quad (3.3)$$

(3.2) ve (3.3) denklemlerinde i parçacık indeksini, d parçacık boyutunu, k yineleme sayısını, x parçacık pozisyonunu, v parçacık hızını, Δt benzetim adımını belirtmektedir. Hız güncellemesinde kullanılan r_1 ve r_2 değerleri $[0,1]$ arasında rastgele seçilen sayılardır. Öğrenme faktörleri olarak bilinen c_1 ve c_2 katsayıları parçacıklara ait bilişsel ve sosyal tecrübelerin ağırlıklandırılmasında kullanılmaktadır. c_1 ve c_2 katsayıları için en çok tercih edilen değer, Clerc ve Kennedy'nin önerisiyle, 1.494 değeri olarak gözükmektedir (Clerc ve Kennedy 2002, Altınöz ve Yılmaz 2011). Atalet değeri w ise Kennedy ve Eberhart tarafından yapılan ilk çalışmada yer almamaktadır. Daha sonra Shi ve Eberhart tarafından yapılan çalışmada performans artışı sağlanması için bir önceki adımdaki hız değerine katsayı çarpımı olarak eklenmiştir. Bu işlem ile performans artışının sağlanması, sürünün başlangıçta geniş bir alanı taraması ve sürü optimuma yaklaştıkça taranan alan küçülmesi ve bu sayede optimum değer elde edilebilmesi şeklinde özetlenebilir.

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k \quad (3.4)$$

Eşitlik (3.4)'de kullanılan $k_{\max}$, algoritma döngüsü maksimum yineleme sayısı, k değeri ise algoritmanın o anda bulunduğu yineleme değeri olarak belirtilmektedir. $w_{\max}$ ve $w_{\min}$ değerleri, deneysel olarak 0.9 ve 0.4 olarak seçilmektedir (Shi ve Eberhart 1999, Altınöz ve Yılmaz 2011).

PSO algoritmasında hem parçacığa ait kişisel bellek hem de tüm parçacıkları içeren sürüye ait sosyal bir bellek bulunmaktadır. Algoritmada hız ve pozisyon güncellemesi kişisel ve sosyal bellekte tutulan en iyi değerlere göre gerçekleştirilmektedir. Kişisel en iyi $x_{pbest_i}^{(d)}$, parçacığın o ana kadar gerçekleştirilen yinelemede elde ettiği en uygun çözüm değeri olarak kaydedilmektedir. Global en iyi $x_{gbest}^{(d)}$ ise sürüdeki tüm parçacıkların yineleme boyunca ulaşabildiği en iyi çözüm değeri olarak saklanmaktadır.



Şekil 3.8 PSO konum ve hız güncellemesine ait bir örnek

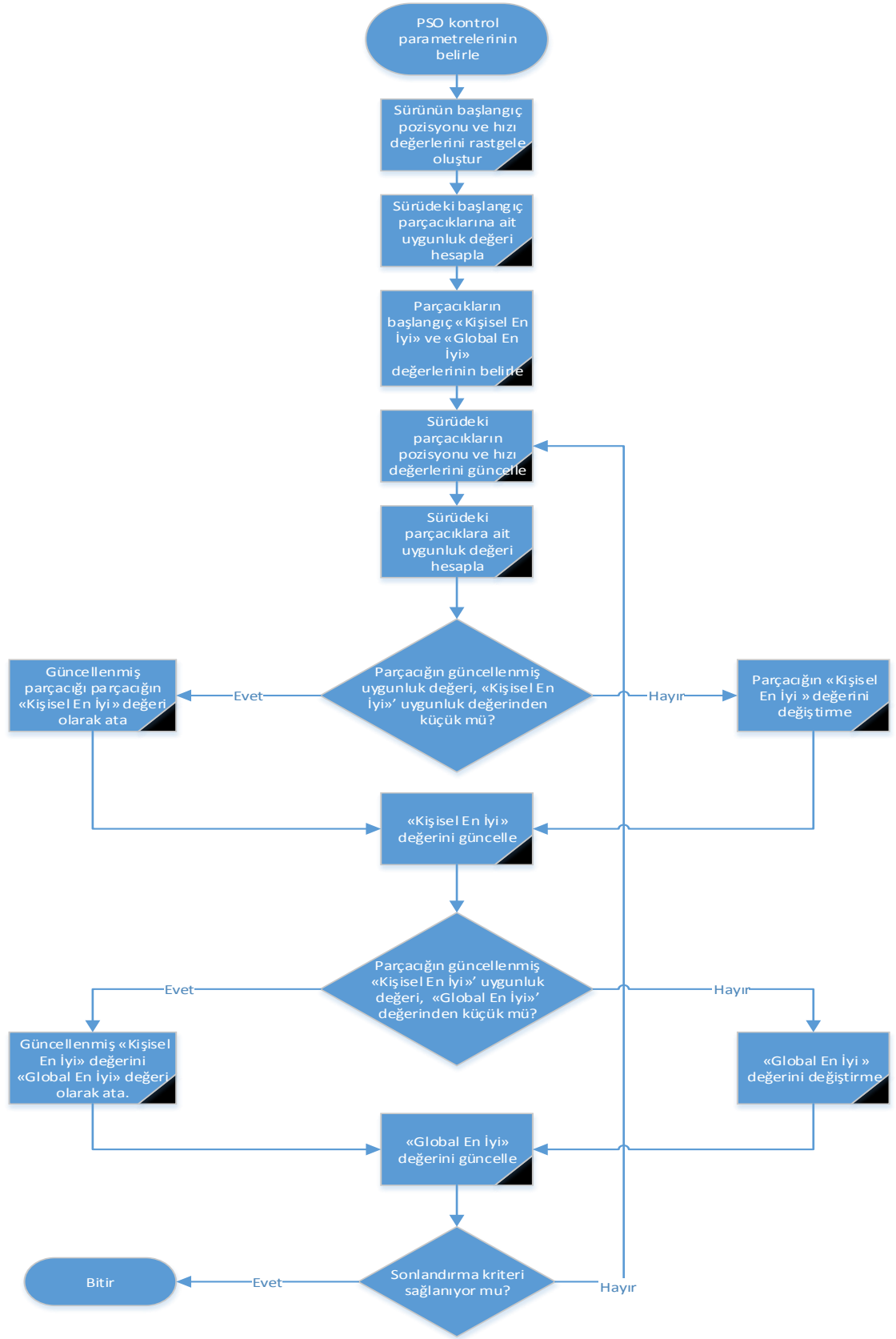
Şekil 3.8’de PSO konum ve hız güncellemesine ait bir örnek verilmiştir. Şekildeki birinci hız vektörü bileşeni bir önceki hızın “eylemsizliğini” göstermektedir. Hız birden bire değişemez, maddelerin bir eylemsizliği vardır. Mevcut hızdan değişim olmaktadır. İkinci hız vektörü bileşeni parçacığın kendi kendine düşünmesini ve kendi geçmiş tecrübesini gösteren “ıdrak” kısmı olarak adlandırılabilir. Üçüncü hız vektör bileşeni ise parçacıklar arasındaki işbirliğini temsil eden “sosyalleşme” kısmıdır. Bu kısım ile parçacıklar sürünün uçuş tecrübesinden yararlanırlar (Önder 2011). Bu üç hız vektör bileşeninden yararlanarak, güncellenmiş parçacık hız vektörü ve güncellenmiş parçacık pozisyonuna erişilmektedir.

PSO algoritmasının GA’ya kıyasla basit bir yapısı vardır. Bu algoritmada da GA’da olduğu gibi çeşitli parametrelere sahiptir. Bunlar parçacık sayısı, parçacık boyutu,

parçacık maksimum hızı vb. olarak yazılabilir. Parçacık sayısı ve parçacık boyutu değerleri üzerinde çalışılan probleme göre seçilen parametrelerdir. Maksimum hız parametresi her bir parçacığın hız değerinin sınırlayan, problemdeki parçacık aralığına göre belirlenen bir parametredir.

$$v_{\max} = \frac{\text{Arama Uzayı Büyüklüğü}}{\text{Yineleme Sayısı}} \quad (3.5)$$

Eşitlik (3.5) yardımıyla her bir parçacığın alabileceği maksimum hız parametresi belirlenebilmektedir.



Şekil 3.9 PSO akış diyagramı

Şekil 3.9’da PSO algoritması akış diyagramı verilmiştir. PSO algoritmasını adımlar halinde aşağıdaki şekilde sıralayabiliriz:

Adım 1: PSO algoritması ile probleme uygun parçacık sayısı, parçacık boyutu, parçacık maksimum hızı vb. değerlerin seçilmesidir.

Adım 2: PSO kontrol parametrelerine göre belirlenen boyut ve sayıda, parçacıklara rastgele olacak şekilde hız ve pozisyon ataması yapılmasıdır.

Adım 3: Başlangıç pozisyon değerlerine göre her bir parçacık için uygunluk değerlerinin hesaplanmasıdır.

Adım 4: Hesaplanan uygunluk değerlerine göre “Kişisel En İyi” ve “Global En İyi” parçacıkların belirlenmesidir.

Adım 5: “Kişisel En İyi” ve “Global En İyi” parçacıkları ile güncellenmiş hız ve konum değerlerinin hesaplanmasıdır.

Adım 6: Güncellenmiş pozisyon değerlerine göre her bir parçacık için uygunluk değerlerinin bulunmasıdır.

Adım 7: Her parçacığın uygunluk değerleri ile kendi geçmiş en iyi uygunluk değerlerinin karşılaştırılarak eğer mevcut değer “Kişisel En İyi” den daha iyi ise mevcut değer “Kişisel En İyi” olarak belirlenmesidir.

Adım 8: Güncellenmiş tüm “Kişisel En İyi” uygunluk değerleri ile geçmiş en iyi uygunluk değeri karşılaştırılarak eğer mevcut değer “Global En İyi” den daha iyi ise mevcut değer “Global En İyi” olarak atanmasıdır.

Adım 9: Adım 5-8 sonlandırma kriteri sağlanana kadar tekrar edilir. Sonlandırma kriteri olarak belirlenen yineleme sayısı ya da yeterli görülen bir uygunluk değeri seçilir.

3.3 Çok Amaçlı Optimizasyon

Genelde gerçek hayatta karşımıza çıkan problemlerin çözümü için birden fazla amacın optimize edilmesi gerekmektedir. Çok amaçlı optimizasyon problemlerinin çözümü,

genelde amaçların birbirleri ile çelişmesi sonucu tek amaçlı optimizasyon problemlerine göre çok daha zordur. Çok amaçlı optimizasyon problemlerinde her zaman tek amaçlı optimizasyon problemlerinde olduğu gibi tek bir optimum çözüm üretmek mümkün olmamaktadır. Bunun yerine, istenilen bütün amaçların üzerinde mutabık kaldığı sonlu uzunluktaki bir kümeden çözüm seçmesi mümkün olabilmektedir.

Genel bir çok amaçlı optimizasyon problemi, n adet parametre (karar değişkeni), k tane amaç fonksiyonu ve m adet bir grup kısıt içermektedir. Amaç fonksiyonları ve kısıtlar karar değişkenlerinin fonksiyonu şeklinde yazılırlar. Optimizasyon işlemi,

$$\begin{aligned} \min \quad & y = f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{Koşul:} \quad & e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \leq 0 \\ & x = (x_1, x_2, \dots, x_n) \in X \\ & y = (y_1, y_2, \dots, y_k) \in Y \end{aligned} \tag{3.6}$$

olarak formüle edilebilir. Eşitlik (3.6)'da kullanılan x karar vektörü, y amaç vektörü, X karar uzay ve Y ise amaç uzayı olarak adlandırılmaktadır. Kısıtlar $e(x) \leq 0$ tanımlanan problem için erişilebilir çözümler kümesini göstermektedir (Zitzler 1999).

3.3.1 Çok amaçlı optimizasyon terminolojisi

Tanım 1: (Erişilebilir Küme) Erişilebilir küme X_f , x karar vektörü kümesi olup $e(x)$ kısıtları sağlamaktadır.

$$X_f = \{x \in X \mid e(x) \leq 0\}$$

şeklinde tanımlanabilir. X_f 'in imajı yani amaç uzayındaki erişilebilir bölge ise,

$$Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\}$$

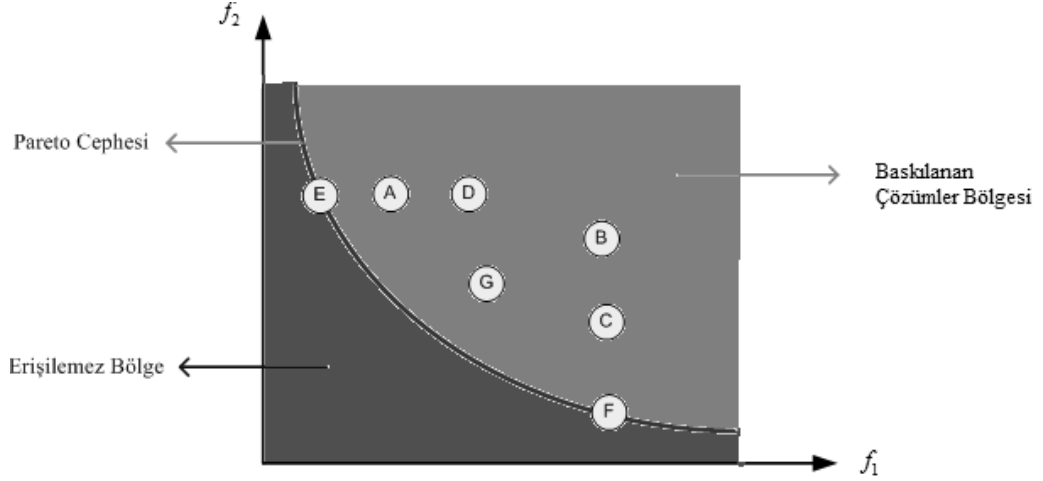
Olarak tanımlanmaktadır.

Tanım 2: (Dominantlık) Herhangi iki karar vektörü (Örn: x_1, x_2) için dominantlık kavramı aşağıdaki gibi verilmektedir:

- $x_1 \prec x_2$ x_1, x_2 'ye göre dominanttır.
Yalnız ve yalnız: $f(x_1) < f(x_2)$
- $x_1 \preceq x_2$ x_1, x_2 'nin zayıf dominantıdır
Yalnız ve yalnız $f(x_1) \leq f(x_2)$
- $x_1 \cong x_2$ x_1, x_2 'ye eşittir.
Yalnız ve yalnız $f_1(x_1) \leq f_2(x_2)$ ve $f_1(x_2) \leq f_2(x_1)$

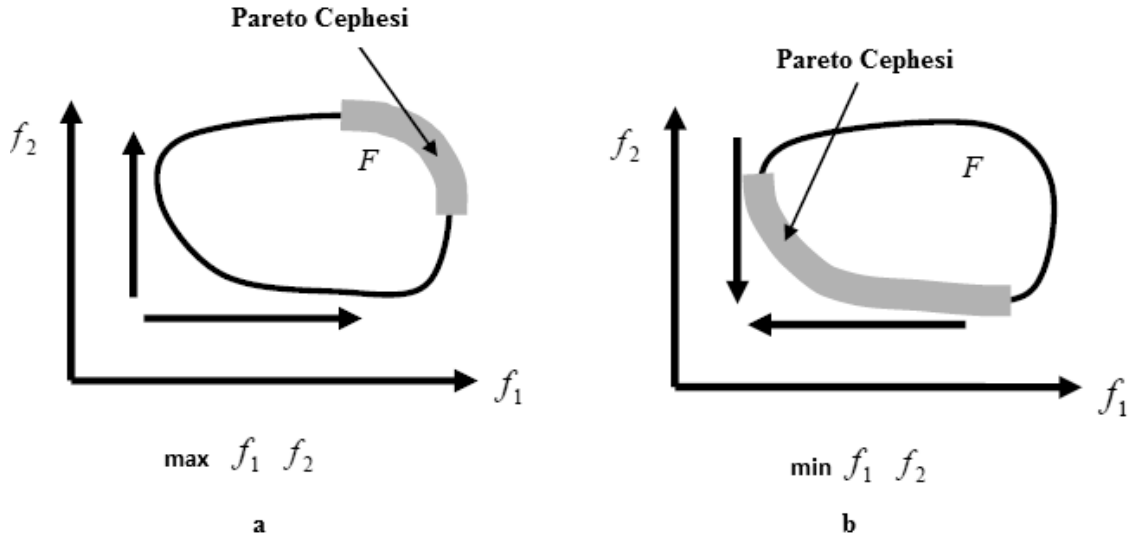
Tanım 3: (Pareto-Optimal) Pareto-Optimal kavramı 1900'lü yılların başlarında, esasen İtalyan bir iktisatçı ve sosyolog olan Vilfredo Pareto tarafından bulunmuştur. Pareto-Optimal kavramı, evrimsel algoritmalara uyarlanmasıyla, çok amaçlı evrimsel optimizasyon algoritmaları geliştirilmiştir. Buna göre, bir çözüm amaç değerlerine göre, en iyi, en kötü ve diğer çözümlere eşit olabilir. En iyi çözüm, amaçların herhangi biri içinde en kötü olmayan ve en azından bir amaç içinde diğerlerinden daha iyi olan çözüm anlamındadır. Pareto-Optimal çözüm, arama uzayında, herhangi bir diğer çözüm tarafından domine edilmeyen çözümdür (Deb 2001).

Tanım 4: (Pareto cephesi) Tüm amaç fonksiyonları göz önünde bulundurularak, domine eden tüm çözümlerin kümesi, Pareto Ön Yüzü veya daha yaygın kullanılan tabirle Pareto Cephesi olarak adlandırılır.



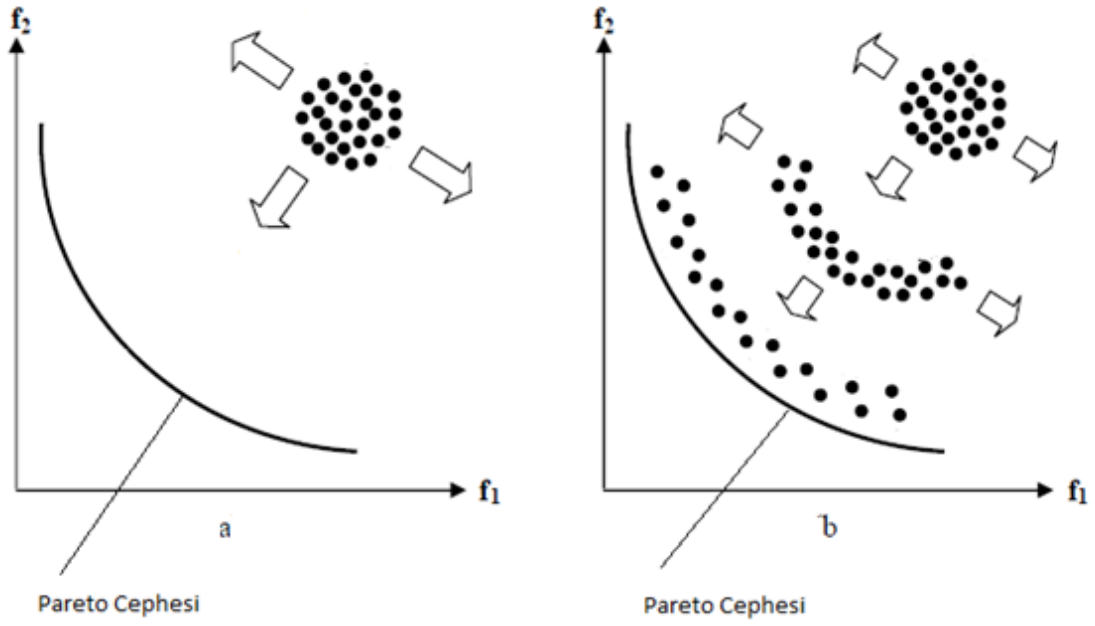
Şekil 3.10 Pareto cephesi fiziksel anlamı

Şekil 3.10’da iki amaçlı örnek bir minimize etme problemine ait bir şekil görülmektedir. Bu şekilde ok ile belirtilen çizgi Pareto cephesidir. Bu cephe çok amaçlı bir optimizasyon probleminde ulaşılabilecek çözümlerin sınırını belirler Şekildeki E ve F çözümleri Pareto cephesi üzerinde olup, diğer A, B, C, D ve G çözümleri ise baskılanan çözümler bölgesinde yer almaktadırlar. Bir çözümün daha iyi olabilmesi için çözümün hiçbir amaç değerinin diğer çözümlerin amaç değerlerinden düşük olmaması ve en az bir amaç değerinde daha küçük olması demektir. Örneğin C çözümü, B çözümüne göre daha iyi bir çözümdür. Ancak F çözümü hem C çözümünü hem de B çözümünü domine etmektedir. E ve F çözümleri amaç değerlerinden birinin bir çözümde, diğerinin ise diğer çözümde yüksek olmasından dolayı bu çözümler eşittir. E ve F çözümlerini amaç uzayı içerisinde baskın gelecek başka çözümler olmadığı için bu çözümler Pareto-optimal çözüm kümesi içerisinde yer almaktadırlar. Pareto Cephesi üzerinde gezinerek, farklı amaçlardaki farklı tolerans seviyelerine ilişkin çözümleri bulmak mümkündür.



Şekil 3.11 İki amaçlı Pareto cephesi gösterimi (Ngatchou vd. 2005)

Şekil 3.11’de iki amaç fonksiyonuna sahip maksimizasyon ve minimizasyon problemleri için ortaya çıkan Pareto cephesi örneği görülmektedir. F bölgesi problemler için amaç uzayı bölgesini göstermektedir. Her iki grafik incelendiğinde hem maksimizasyon problemi hem de minimizasyon problemi için ok yönlerine doğru gidildikçe, daha iyi çözümleri içinde barındıran Pareto cephesine ulaşılabilindiği gözlemlenmektedir. Şekil 3.11.b’de görülmekte olan okların kesişecekleri nokta “karşıt nokta”, bu noktanın sol altında kalan noktalar ise “ütopik nokta” olarak adlandırılır.



Şekil 3.12 Minimizasyon problemi için amaçlanan Pareto cephesi yayılımı

Çok amaçlı optimizasyonun asıl hedefi, Pareto cephesini bulmak veya ona yaklaşmak ve bu cephe üzerinde düzgün bir dağılım sağlamaktır. Şekil 3.12’de bir minimizasyon problemi için istenilen Pareto cephesi yayılımı görülmektedir. Düzgün dağılımın artısının sağlanması için başlangıçtaki topluluk diyagonal oklar yönünde genişlemesi gerekirken, aynı zamanda topluluğun Pareto cephesine yakınsaması için de orijine doğru çizilen ok yönünde ilerlemesi gerekmektedir. Düzgün dağılım – yakınsama balansı doğru ayarlandığında, şekil 3.12-b’de gösterildiği gibi probleme ait iyi bir çözüm kümesi bulunabilir (Ergül 2010).

3.3.2 Çok amaçlı optimizasyon problemlerinde skalarizasyon yaklaşımları

Gerçek dünyada karşımıza çıkan optimizasyon problemlerindeki amaçların genel olarak birbirleriyle çelişmesi neticesinde matematiksel olarak tek bir optimal çözüm bulmak pek mümkün olamamaktadır ancak üzerinde uzlaşılan çözümler yani Pareto optimal kümesi elde edilebilmektedir. Skalarizasyon işleminin çok amaçlı optimizasyon problemlerinin çözümünde anahtar bir rolü vardır ve literatürde farklı yaklaşımlara dayanan bir çok birleştirme yöntemlerinden bahsedilmektedir. Çok amaçlı optimizasyon problemini tek amaçlı hale dönüştürmek için kullanılan yöntemler, tüm

amaç fonksiyonlarının ağırlıklandırma fonksiyonu yardımıyla birden fazla amacın aritmetik operatörler ile birleştirilmesi, bir amaç fonksiyonunu diğer amaç fonksiyonlarını kısıt olarak kabul ederek optimize edilecek tek bir amaç fonksiyonunun tanımlanması, uzaklık ölçümü gibi çözümleri içermektedir. Skalarizasyon olarak adlandırılan bu dönüşüm işlemi literatürde birleştirme, tek amaca indirgeme gibi farklı isimlerle de anılmaktadır (Miettinen 1999, Zitzler 1999, Miettinen ve Makela 2002, Lin 2003, Ngatchou vd. 2005, Ehrgott 2006, Klamroth ve Tind 2006, Eichfelder 2008).

Bu tez kapsamında ağırlıklandırma fonksiyonu kullanılarak gerçekleştirilen ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı ve uzaklık ölçüm temelli Chebychev skalarizasyon yaklaşımı incelenecektir.

3.3.2.1 Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı

Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı Gaas ve Saaty (1955) tarafından önerilmiştir. Birden fazla amacın ağırlıklandırılarak amaç fonksiyonların lineer kombinasyonları şeklinde yazılıp tek amaca dönüştürülmesi yaklaşımıdır.

$$\min \sum_{i=1}^k w_i f_i(x) \quad (3.7)$$

Koşul: $x \in X$

Eşitlik (3.7)'de Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı verilmektedir. Bu yaklaşım, farklı w_i değerleri için $0 < w_i < 1$ elde edilecek ağırlıklandırılmış toplam fonksiyonlarının ayrı ayrı minimize edilmesi esasına dayanır. k tane $i = 1, 2, \dots, k$ amaç fonksiyonu için bu ağırlıklar;

$$\sum_{i=1}^k w_i = 1 \quad \text{ve} \quad w_i > 0 \quad (3.8)$$

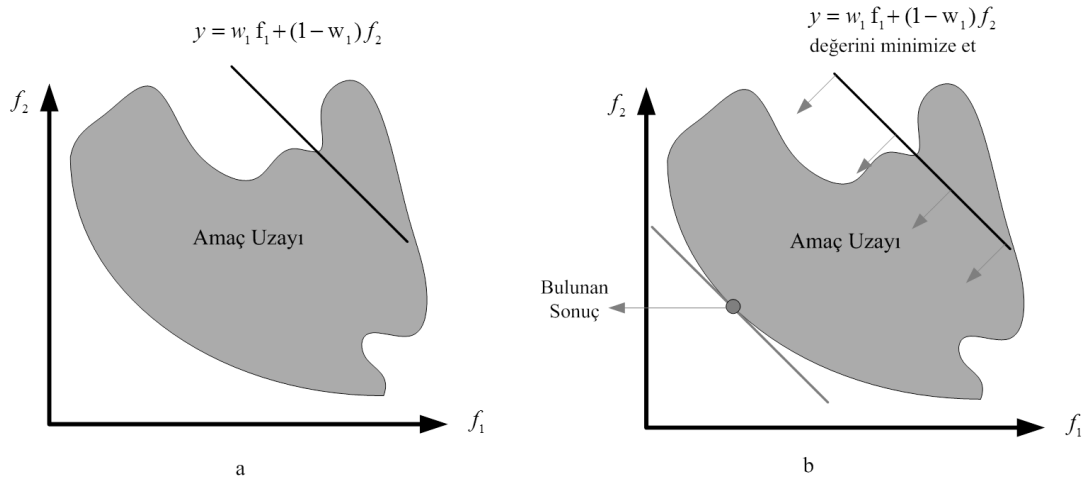
(3.8) koşulu dikkate alınarak belirlenmektedir.

Teorem: Eşitlik (3.7)'deki ağırlıklandırılmış problem çözümü, eşitlik (3.8) koşuluna göre Pareto optimaldir.

İspat: (Olmayana ergi yöntemiyle) $x^* \in X$ problemin çözümü olsun ve çözümün Pareto optimal olmadığı varsayımı yapılsın. Bu durumda tüm $i = 1, 2, \dots, k$ $f_i(x) \leq f_i(x^*)$ ve en az bir j değeri için $f_j(x) < f_j(x^*)$ eşitsizliğini sağlayacak $x \in X$ çözümü ortaya çıkacaktır. $w_i > 0$ $i = 1, 2, \dots, k$ olduğundan $\sum_{i=1}^k w_i f_i(x) < \sum_{i=1}^k w_i f_i(x^*)$ elde edilir. Bu durum problem çözümü konusunda yapılan varsayımla çeliştiğinden dolayı x^* Pareto optimal olduğu kanıtlanmıştır (Miettinen 1999).

Ağırlık değeri her değiştirildiğinde yeni bir tek amaçlı optimizasyon problemi oluşturulur; farklı ağırlıklar, farklı Pareto çözümlerinin elde edilmesini sağlar. Farklı Pareto-optimal çözümlerin bulunabilmesi için optimizasyon algoritmasının farklı ağırlıklarla çalıştırılması gerekmektedir.

Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı'nın çalışma prensibi, iki amaç fonksiyonlu minimizasyon problemi için şekil 3.13.a'da resmedilmiştir. Minimize edilen $y = w_1 f_1(x) + (1 - w_1) f_2(x)$ fonksiyonu, Amaç Uzayı'nda belirli bir eğime sahip doğruya karşılık gelmektedir. Problemdeki Ağırlıklandırılmış Toplam Temelli Skalarizasyon ile elde edilen amaç fonksiyonun minimize edilmesi, ilgili doğrunun eğiminin korunarak şekil 3.13.b'de görüldüğü gibi kaydırılması anlamına gelmektedir.



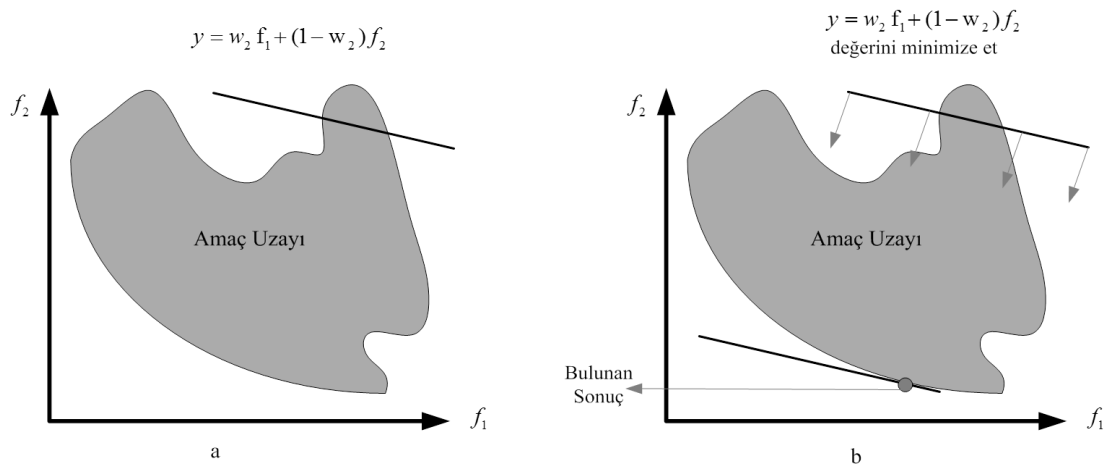
Şekil 3.13.a.Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı, b. Konveks Pareto cephesinde elde edilen çözüm

Belirli bir eğime sahip doğruya karşılık gelen $y = w_1 f_1(x) + (1 - w_1) f_2(x)$ fonksiyonu;

$$y = w_1 f_1(x) + (1 - w_1) f_2(x)$$

$$f_2(x) = -\frac{w_1}{(1 - w_1)} f_1(x) + \frac{y}{(1 - w_1)} \quad (3.9)$$

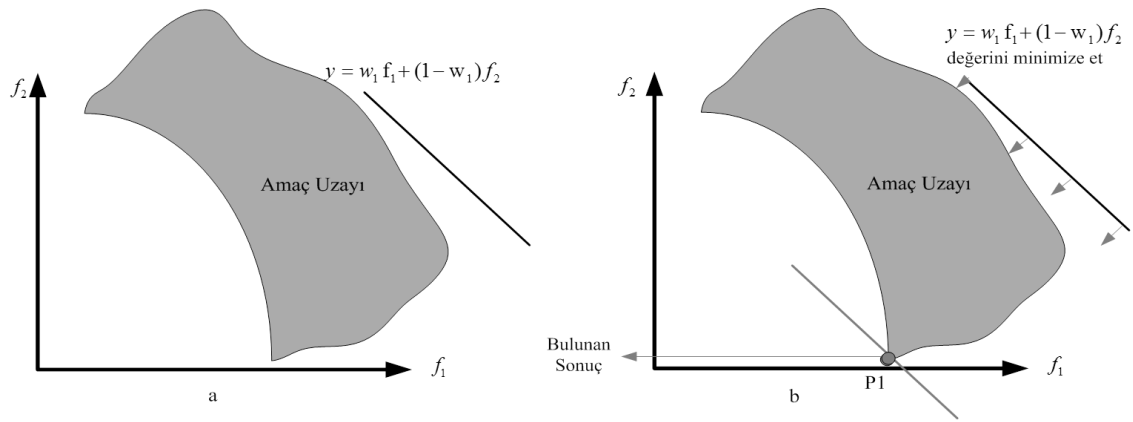
şeklinde yeniden düzenlendiğinde bu doğrunun eğimi $m = -\frac{w_1}{(1 - w_1)}$ olarak hesap edilmektedir. Bulunan minimum değer, yani sonuç da şekil 3.13.b’de görülen kırmızı noktaya karşılık gelir.



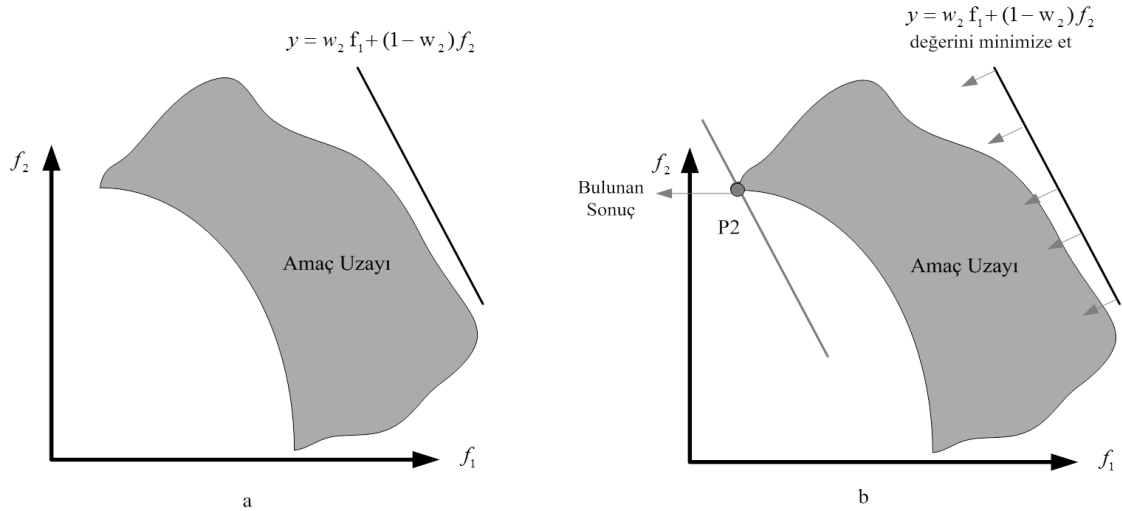
Şekil 3.14.a.Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı, b. Farklı ağırlık değeriyle konveks Pareto cephesinde elde edilen çözüm

Doğrunun eğimi w tarafından belirlendiği için, farklı ağırlık değerleri sayesinde farklı eğimli doğrular bulunur. Böylelikle şekil 3.14'da görüldüğü gibi Pareto Cephesi üzerinde farklı noktalar elde etmek mümkündür.

Pareto cephesinin konveks olduğu durumlarda Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı'nın iyi sonuç verdiği şekil 3.13 - 3.14 grafiklerinden görülmektedir. Ancak bu yaklaşımın dezavantajı Pareto cephesinin konkav olduğu durumlarda ortaya çıkmaktadır.



Şekil 3.15.a.Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı, b. Konkav Pareto cephesinde elde edilen çözüm



Şekil 3.16.a.Farklı ağırlık değeriyle Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı, b. Farklı ağırlık değeriyle konkav Pareto cephesinde elde edilen çözüm

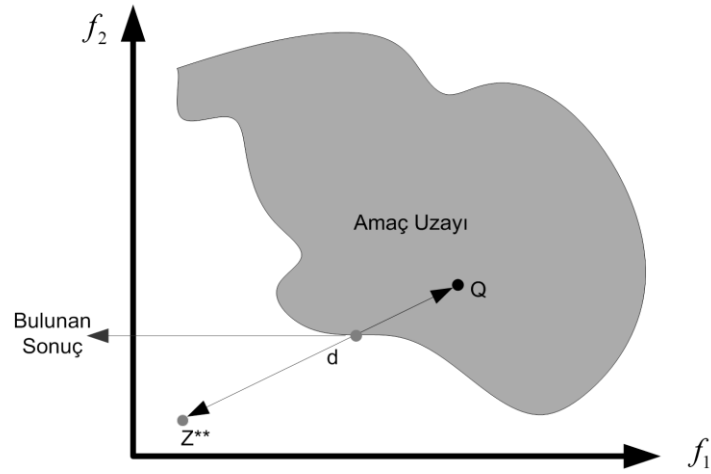
Şekil 3.15 - 3.16 grafiklerinden de görüldüğü üzere Pareto cephesinin konkav olduğu problemler için Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı pek kullanışlı değildir. Farklı ağırlık değerleri dolayısıyla farklı eğimlerdeki doğrular minimizasyon için kaydırıldığında Pareto cephesi üzerinde ulaşılabilecek iki nokta P1 ve P2 noktaları olacaktır; ağırlıklar her ne seçilirse seçilsin P1 ve P2 noktaları dışında herhangi bir sonuç elde edilemeyecektir.

3.3.2.2 Genelleştirilmiş Chebychev skalarizasyon yaklaşımı

Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı doğrusal bir skalarizasyon yaklaşımı olup Pareto cephesinin konveks olmadığı durumlarda uygulanabilirliği konusunda dezavantaja sahip olduğu şekil 3.15 - 3.16 grafiklerinden anlaşılmaktadır. Bu durumların üstesinden gelebilmek için doğrusal olmayan bir skalarizasyon yöntemi olan Genelleştirilmiş Chebychev Skalarizasyon Yaklaşımı kullanılabilmektedir. Genelleştirilmiş Chebychev Skalarizasyon Yaklaşımı Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımına göre üç açıdan çözümde bir ilerleme sağlamaktadır:

- Pareto cephesinin şekline, konveks veya konveks olmamasına, bakmaksızın Pareto optimal çözümleri bulabilmesi.
- Pareto optimal çözümler arasında daha iyi yayılmış bir Pareto cephesi elde edebilmesi.
- Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımına göre kullanılan amaç fonksiyonu ağırlıklarına daha az duyarlılık göstermesi (Moffaert vd. 2013)

Genelleştirilmiş Chebychev Skalarizasyon Yaklaşımı, farklı w_i değerleri için ($0 < w_i < 1$) elde edilecek $d_i = \min_{i=1, \dots, k} \text{dist}(z_i^{**}, (w_i f_i(x)))$ değerlerinin ayrı ayrı minimize edilmesi esasına dayanmaktadır.



Şekil 3.17 Genelleştirilmiş Chebychev Skalarizasyon Yaklaşımı

Şekil 3.17’de görülen z_i^{**} noktası problem için belirlenen ütöpik nokta, $Q_i = w_i f_i$ noktası değerleridir. $dist$ fonksiyonu ise mesafe normu özelliklerini sağlamak zorunda olan bir fonksiyon olup; L_1 (Manhattan veya Taxi-Cab Normu), L_2 (Öklid Normu), L_{inf} (Maksimum Normu) gibi literatürde geçen normlardan herhangi biri olarak seçilebilir.

3.3.2.3 Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımı

Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı, Genelleştirilmiş Chebychev Skalarizasyon Yaklaşımı’nda ilgili mesafe normunun L_{inf} (Max-Norm) normu olarak seçildiği duruma karşılık gelmektedir. Bu yaklaşım Zeleny (1973) tarafından önerilen bir yaklaşımdır.

$$\min \max_{i=1, \dots, k} [(w_i f_i(x) - z^{**})] \quad (3.10)$$

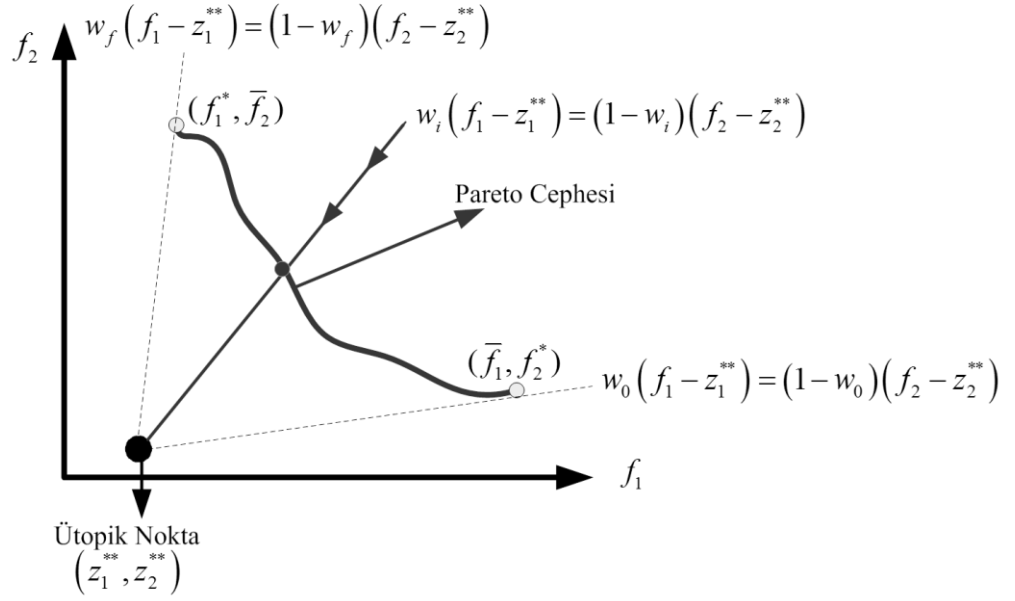
Matematiksel olarak eşitlik (3.10)’daki gibi ifade edilmektedir (Miettinen 1999, Marler ve Arora 2004, Kalmroth ve Tind 2006, Eichfelder 2008, Dutta ve Kaya 2011, Weerasena 2014). Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı’nda birden

fazla fonksiyonun ağırlık değeri w_i ($0 < w_i < 1$) değiştirilip maksimumu alınarak tek amaçlı optimizasyon problemine çevrilme işlemi gerçekleştirilmektedir.

Teorem: Eşitlik (3.10)'da ağırlıklandırılmış problem çözümü, eşitlik (3.8) koşuluna göre Pareto optimaldir.

İspat: (Olmayana ergi yöntemiyle) $x^* \in X$ problemin çözümü olsun ve çözümün Pareto optimal olmadığı varsayımı yapılsın. Bu durumda tüm $i=1,2,...,k$ $f_i(x) \leq f_i(x^*)$ ve en az bir j değeri için $f_j(x) \leq f_j(x^*)$ eşitsizliğini sağlayacak $x \in X$ çözümü ortaya çıkacaktır. $w_i > 0$ $i=1,2,...,k$ için $w_i(f_i(x) - z_i^{**}) \leq w_i(f_i(x^*) - z_i^{**})$ olduğundan dolayı $\max(w_i(f_i(x) - z_i^{**})) \leq \max(w_i(f_i(x^*) - z_i^{**}))$ eşitsizliğini sağlamaktadır. Öte yandan $f_i(x) - z_i^{**} \leq f_i(x^*) - z_i^{**}$ $i=1,2,...,k$ ve en az bir eşitsizlik kesindir. Bu nedenle $\sum_{i=1}^k f_i(x) - z_i^{**} < \sum_{i=1}^k f_i(x^*) - z_i^{**}$ sağlanır. Bu durum problem çözümü konusunda yapılan varsayımla çeliştiğinden dolayı x^* Pareto optimal olduğu kanıtlanmıştır (Miettinen 1999).

Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı'nda Pareto cephesi oluşturması önem arz etmektedir. Bu yaklaşımda w ağırlıklandırma değerleri $[0,1]$ aralığında seçilmektedir. Bu aralık daha ayrıntılı belirtilmek gerekirse, w ağırlıklandırma değerlerinin $0 < w_0 < w < w_f < 1$ şeklinde seçilmesi genellikle Pareto cephesinin oluşturulması için yeterlidir. Şekil 3.18'de Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı geometrisi gösterilmektedir. Şekilden de görüldüğü üzere sınır ışınları arasında kalacak şekilde ağırlıklandırma değerleri seçilerek farklı noktalar elde edildikten sonra probleme ait Pareto cephesi meydana getirilmektedir (Dutta ve Kaya 2011, Kaya ve Maurer 2013).



Şekil 3.18 Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı

Şekil 3.18’de kullanılan sınır değerleri ütopik noktalardan geçen ve belirli bir eğime sahip ışınlarla saptanmaktadır. Burada kullanılan maksimum ve minimum ışın eğimleri,

$$\alpha_{\max} = \arctan \left(\frac{(\bar{f}_2 - z_2^{**})}{(\bar{f}_1 - z_1^{**})} \right)$$

$$\alpha_{\min} = \arctan \left(\frac{(f_2^* - z_2^{**})}{(\bar{f}_1 - z_1^{**})} \right)$$
(3.11)

olarak hesaplanmaktadır.

Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı’nda Pareto cephesinin elde edilmesi konusu Dutta ve Kaya (2011) ve Kaya ve Maurer (2013) yaptığı çalışmalarda bir sistematığe oturulmuştur.

Adım 1: Ütopya parametreleri $\varepsilon_1, \varepsilon_2 > 0$ seçilmesi ve Pareto cephesini oluşturacak $N+1$ nokta sayısının belirlenmesi.

Adım 2: Pareto cephesi sınır koşullarının elde edilmesi için P1 probleminin çözüm $\bar{x}$ değerinin bulunması.

$f_1^* \triangleq f_1(\bar{x})$ ve $\bar{f}_2 \triangleq f_2(\bar{x})$ olmak üzere Pareto cephesinde $(f_1^*, \bar{f}_2)$ gibi bir sınır noktası elde edilmesi.

Pareto cephesi sınır koşullarının elde edilmesi için P2 probleminin çözüm $\bar{x}$ değerinin bulunması.

$\bar{f}_1 \triangleq f_1(\bar{x})$ ve $f_2^* \triangleq f_2(\bar{x})$ olmak üzere Pareto cephesinde $(\bar{f}_1, f_2^*)$ gibi bir sınır noktası elde edilmesi.

Adım 3: $z_i^{**} = f_i^* - \varepsilon_i$ $i=1,2$. sağlayacak (z_1^{**}, z_2^{**}) ütöpik noktasına ulaşılması.

Adım 4: Eşitlik (3.11)'den $\alpha_{\min}$ ve $\alpha_{\max}$ değerlerinin elde edilmesi ve artırım miktarı

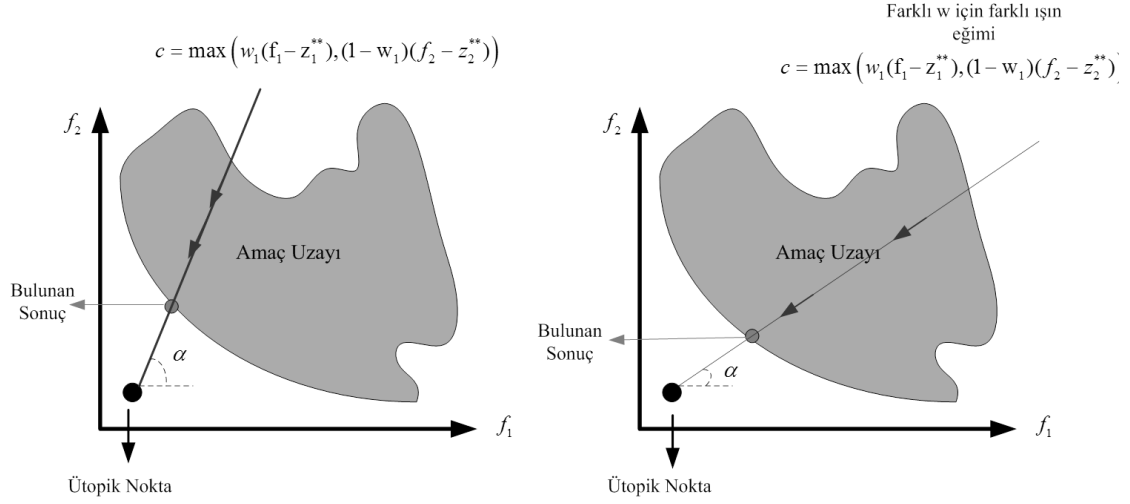
$$\Delta\alpha = \left(\frac{(\alpha_{\max} - \alpha_{\min})}{N} \right) \text{ hesaplanması.}$$

Adım 5: $\alpha = \alpha_{\min} + \Delta\alpha$ değeri bulunarak $w_1 = \sin(\alpha)$ ve $w_2 = \cos(\alpha)$ ağırlıklarının hesaplanması.

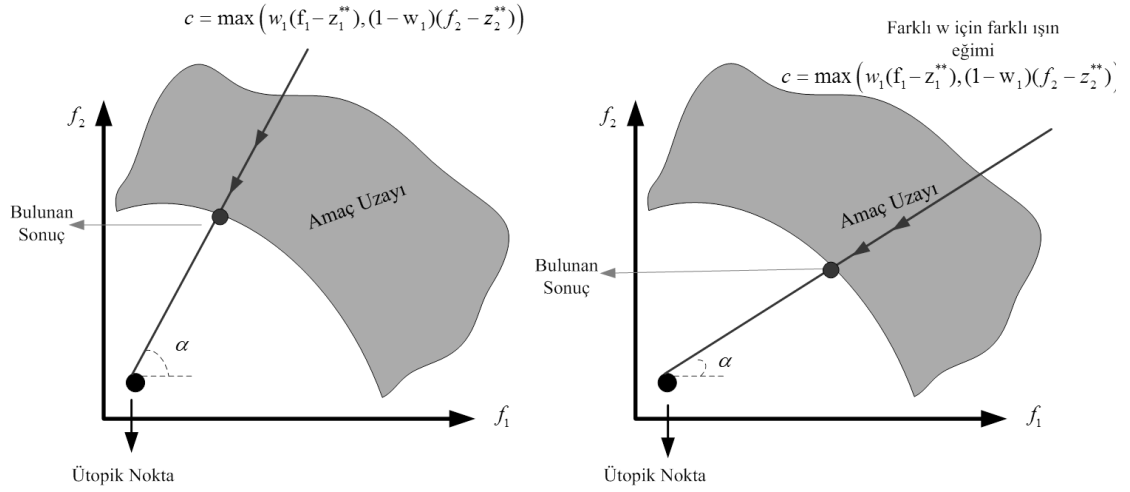
Adım 6: Ağırlıklara göre problem çözümü $\bar{x}$ Pareto minimum değerine ulaşarak Pareto cephesi üzerinde bir $\bar{f}^k = (f_1(\bar{x}), f_2(\bar{x}))$ noktasının elde edilmesi.

Adım 7: 4-6 adımlarının Pareto cephesi üzerinde $N+1$ adet nokta bulununcaya kadar devam ettirilmesi.

Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı'nın konveks ve konkav Pareto cephelerinde ürettiği sonuçların geometrisi şekil 3.19 - 3.20'deki grafiklerde verilmektedir.



Şekil 3.19 Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı'nda farklı ağırlık değeriyle konveks Pareto cephesinde elde edilen çözüm



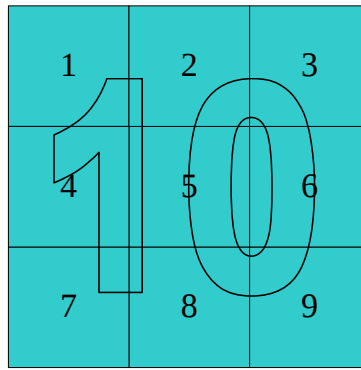
Şekil 3.20 Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı'nda farklı ağırlık değeriyle konkav Pareto cephesinde elde edilen çözüm

Şekil 3.17 ile 3.22 grafikleri karşılaştırıldığında, Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı'nın Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı'na göre konkav Pareto cephelerinde daha iyi sonuç verdiği görülmektedir.

4. ARAŞTIRMA BULGULARI

4.1 Frekans Atama Problemi

Bu tezde çalışılan Frekans Atama Problemi, girişim, sönümlleme vb. bozucu etkilerin önlenmesine yönelik tedbirler gözetilerek en etkin şekilde frekans planlaması yapılması problemidir. Bu problemin tanımlanması ve bir optimizasyon problemi şeklinde matematiksel olarak ifade edilmesi gerçekleştirilecektir.



Şekil 4.1 Frekans ataması gerçekleştirilen çevrim topolojisi

Şekil 4.1’de frekans atama probleminde kullanılan çevrimlerin topolojisi görülmektedir. Problemden frekans planlanmasının gerçekleştirildiği 10 tane çevrim vardır ve bu çevrimlerin komşuluk ilişkileri çizelge 4.1’de verilmiştir.

Çizelge 4.1 Çevrimler ve komşulukları

Çevrim Numarası										
Komşu Oldukları Çevrim Numaraları	1	2	3	4	5	6	7	8	9	10
	2 - 4	3 - 5	2 - 6	1- 5 - 7	2-4- 6- 8	3-5- 9	4 -8	5-7- 9	6 - 8	Tüm Çevrimler

Şekil 4.1 göz önüne alındığında 1’den 10’a kadar ızgara şeklinde bir topoloji gösteren her bir çevrim içerisinde, frekans atama işleminin gerçekleştirileceği 10 tane kullanıcının olduğu varsayımı yapılmaktadır. Buradan hareketle toplamda kullanıma

uygun ve geçerli olan $F_{\text{Frekans K\u00fcmesi}} = \{f_1, f_2, \dots, f_N\}$ N tane frekans de\u011ferine sahip k\u00fcmeye i\u00e7erisinden, 100 tane frekans de\u011feri \u00e7evrimlerdeki kullan\u0131cılara atanacaktır. $F^{(j)} = \{f_1^{(j)}, f_2^{(j)}, \dots, f_{10}^{(j)}\}$ k\u00fcmesi, j 'inci \u00e7evrimin frekans seti, $F^{(k)} = \{f_1^{(k)}, f_2^{(k)}, \dots, f_{10}^{(k)}\}$ k\u00fcmesi ise, k 'inci \u00e7evrimin frekans seti olarak g\u00f6sterilmektedir.

Bu tezde \u00fczerine \u00e7a\u015f\u0131lan Frekans Atama Problemi bir optimizasyon problemi olup, bu problemin sahip oldu\u011fu bir kısıt bulunmaktadır. Bu kısıt matematiksel olarak;

$$(f_a^{(j)}, f_b^{(j)} \in F^{(j)}) \wedge (a \neq b) \Rightarrow f_a^{(j)} \neq f_b^{(j)} \quad j = 1, 2, \dots, 10 \quad (4.1)$$

\u015feklinde ifade edilir. E\u015fitlik (4.1)'de ifade edilen kısıtta, frekans atamas\u0131 ger\u00e7ekle\u015ftirilirken, her bir \u00e7evrim i\u00e7indeki a ve b kullan\u0131c\u0131larına aynı frekans de\u011ferinin atanmasının engellenmesi gerekti\u011fi verilmektedir.

Frekans Atama Problemi i\u00e7in \u00fc\u00e7 ayrı ceza fonksiyonu tanımı yapılmaktadır. Bunlar:

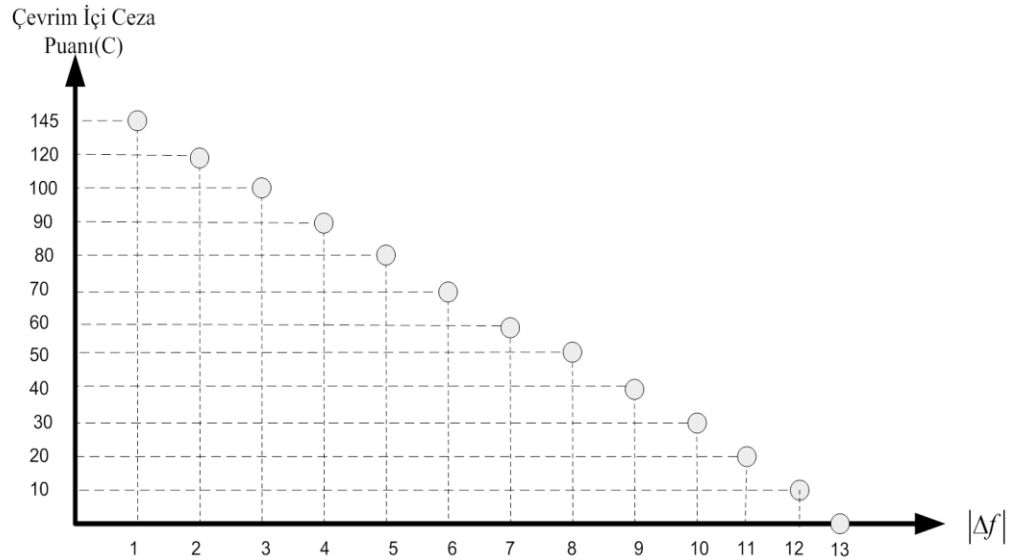
- **\u00c7evrim i\u00e7i ceza fonksiyonu:** Her bir \u00e7evrim i\u00e7erisindeki kullan\u0131cılara frekans atamas\u0131 ger\u00e7ekle\u015ftirilirken belirli ko\u015fulların g\u00f6z \u00f6n\u00fcnde bulundurulması gerekmektedir. Bu ko\u015ful, \u00e7evrim i\u00e7i ceza olarak ifade edilen ve herhangi bir \u00e7evrim i\u00e7erisindeki kullan\u0131cılara atanan frekans de\u011ferlerinin, birbirlerine olan yak\u0131nl\u0131k uzaklık ili\u015fkisine g\u00f6re belirli bir ceza puan\u0131 veren bir fonksiyondur. Bu fonksiyon matematiksel olarak;

$$P^{(j)} = \sum_{a=1}^{10} \sum_{b=a+1}^{10} \exp(s(d_f(f_a^{(j)}, f_b^{(j)}))) \quad j = 1, 2, \dots, 10 \quad (4.2)$$

\u015feklinde form\u00fclize edilir. E\u015fitlik (4.2)'de kullanılan $d_f()$ fonksiyonu, herhangi bir \u00e7evrim i\u00e7erisindeki kullan\u0131cılara atanan iki frekans de\u011feri arasındaki mesafeyi hesaplayan bir fonksiyondur. \u00c7evrim i\u00e7i frekans atamas\u0131 ko\u015fulunun sa\u011flanması i\u00e7in frekans mesafesine ba\u011f\u0131 olarak verilen ceza puanları,

$$d_f(f_a^{(j)}, f_b^{(j)}) = \begin{cases} C_1 & |f_a^{(j)} - f_b^{(j)}| = 1 \\ C_2 & |f_a^{(j)} - f_b^{(j)}| = 2 \\ \vdots & \vdots \\ C_{12} & |f_a^{(j)} - f_b^{(j)}| = 12 \\ 0 & |f_a^{(j)} - f_b^{(j)}| > 12 \end{cases} \quad (4.3)$$

olarak ifade edilmektedir. Çevrim içi ceza puanı verilirken iki frekans değeri arasındaki mesafenin 0 olduğu durum hali hazırda optimizasyon problemi için bir kısıt olarak tanımlanmış olup, bu mesafenin 12'nin üzerinde olduğu bütün durumlar için ise 0 ceza puanı uygulanmaktadır.



Şekil 4.2 Çevrim içi ceza fonksiyonu ceza puanı değerleri

- **Çevrimler arası ceza fonksiyonu:** Her bir çevrim içerisindeki kullanıcılara frekans ataması gerçekleştirilirken ikinci olarak göz önünde bulundurulacak koşul, çevrimler arası ceza fonksiyonudur. Bu koşulun ne ölçüde sağlandığını belirten çevrim içi ceza fonksiyonu eşitlik (4.4)'deki gibi ifade edilmektedir.

$$R^{(i,j)} = \alpha_{i,j} \cdot \frac{1}{2} \cdot \left(\sum_{a=1}^{10} \sum_{b=1}^{10} \exp(s(m_f(f_a^{(i)}, f_b^{(j)}))) \right)$$

$$i = 1, \dots, 10$$

$$j = 1, \dots, 10$$
(4.4)

Eşitlikte kullanılan $m_f()$ fonksiyonu i ve j çevrimleri içerisindeki a ve b kullanıcılarına atanan iki frekans değeri arasındaki mesafeyi hesaplamaktadır.

Çevrimler arası ceza fonksiyonu da çevrim içi ceza fonksiyonunda olduğu gibi atanan frekans değerlerinin birbirlerine olan uzaklığına göre bir ceza puanı atamaktadır. Bu ceza puanı;

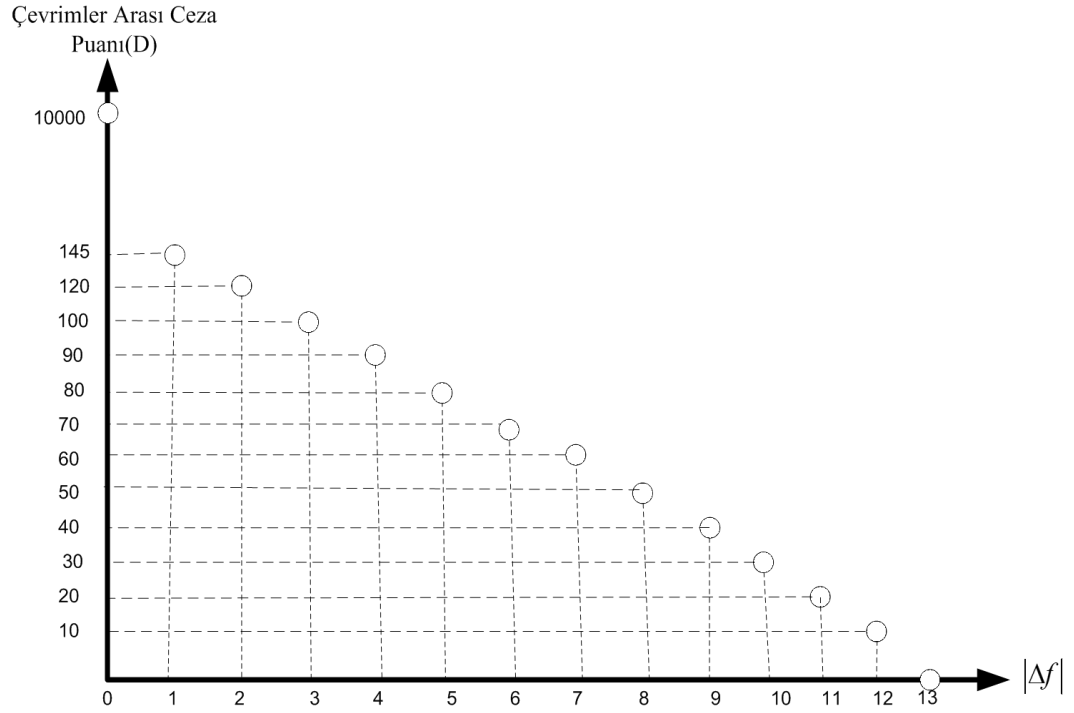
$$m_f(f_a^{(i)}, f_b^{(j)}) = \begin{cases} D_1 & f_a^{(i)} = f_b^{(j)} \\ D_2 & |f_a^{(i)} - f_b^{(j)}| = 1 \\ \vdots & \vdots \\ D_{12} & |f_a^{(i)} - f_b^{(j)}| = 12 \\ 0 & |f_a^{(i)} - f_b^{(j)}| > 12 \end{cases}$$
(4.5)

olarak verilmektedir. Çevrim içi ceza fonksiyonunda olduğu gibi iki frekans arası mesafe 12'nin üzerine çıktığı anda 0 ceza puanı uygulanmaktadır. Eşitlik(4.4)'de verilen çevrimler arası ceza fonksiyonunda kullanılan $\alpha_{i,j}$ ağırlıklandırma katsayısıdır. Bu katsayı değeri;

$$\alpha_{i,j} = \begin{cases} 1, & i, j \text{ nin komşusu ise} \\ 0, & i, j \text{ nin komşusu değilse} \end{cases}$$
(4.6)

olarak yazılır. Çizelge 4.1'deki çevrimler ve komşulukları bilgilerine göre katsayı 1 veya 0 değerini almaktadır. Birbirlerine komşu olmayan çevrimler arası ceza fonksiyonu değeri sıfırdır. Yani komşu olmayan çevrimlerin çevrimler arası ceza fonksiyonu değerine katkısı bulunmamaktadır.

Faklı çevrimler içerisindeki kullanıcıların aynı frekansı almaması konusunda bir kısıt koyulmamaktadır; ancak aynı frekansları almaları yine de çok istenmeyen bir durum olmasından dolayı yüksek bir ceza puanı verilmektedir.



Şekil 4.3 Çevrimler arası ceza fonksiyonu ceza puanı değerleri

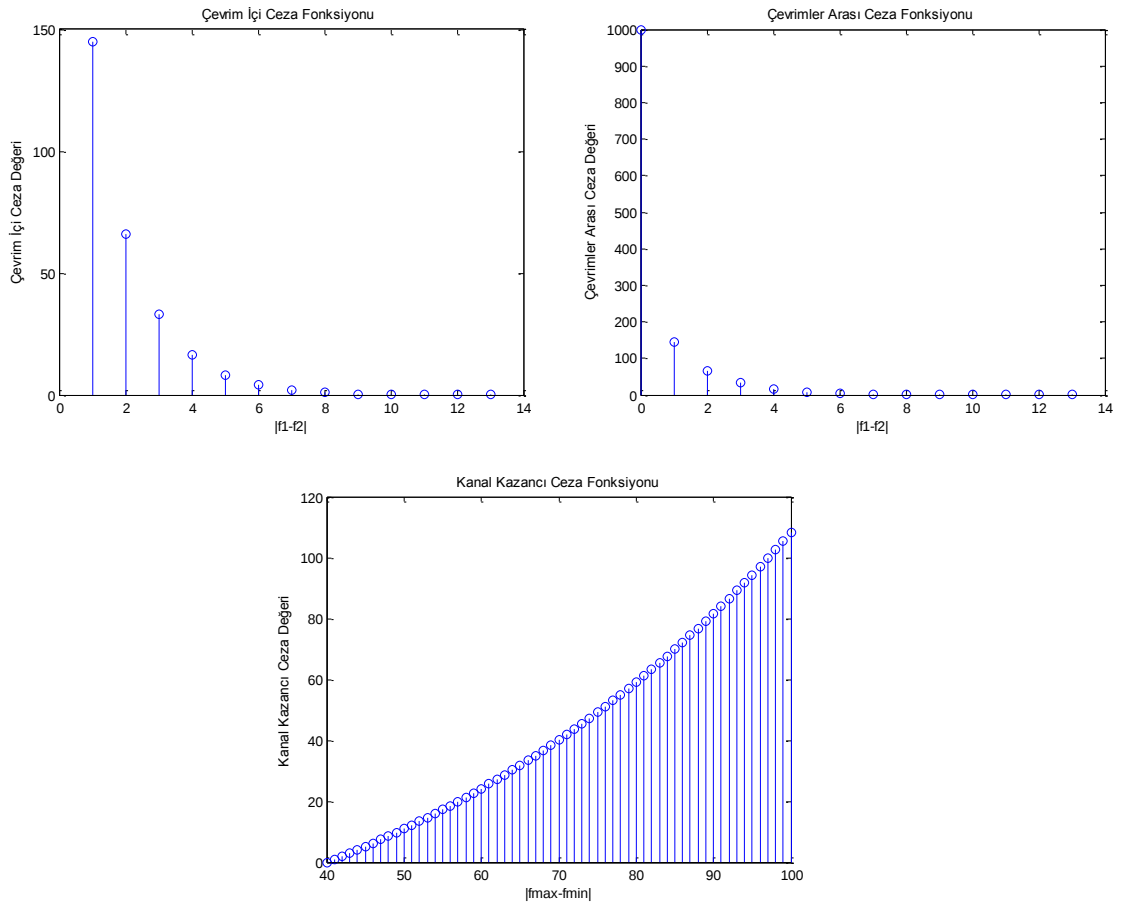
- **Kanal kazancı ceza fonksiyonu:** Her bir çevrim içerisinde kullanıcılara atanacak olan frekans setinin minimum ve maksimum frekans değerleri arasındaki farkın büyüklüğü, sönümlenme gibi bir takım bozucu etkilerin ortaya çıkmasına yol açmaktadır. Bu yüzden her bir çevrime atanan frekans setinin maksimum ve minimum değerlerinin belli bir eşik değerinin altında olması istenmektedir. Bu durumdan yola çıkılarak oluşturulan kanal kazancı ceza fonksiyonu;

$$K^{(j)} = \begin{cases} 0 & (f_{a\max}^{(j)} - f_{a\min}^{(j)} < \delta) \\ \exp(r(f_{a\max}^{(j)} - f_{a\min}^{(j)} - \delta)) & (f_{a\max}^{(j)} - f_{a\min}^{(j)} \geq \delta) \end{cases} \quad (4.7)$$

şeklinde verilmektedir. Tezde üzerinde çalışılan problemde bu eşik değeri $\delta = 40$ olarak alınmaktadır. Eşitlik (4.7)'de çevrim içi maksimum ve minimum frekans

değerleri arasındaki fark belirlenen eşik değerinden küçük olması durumunda kanal kazancı ceza fonksiyonu değerine etki etmezken, bu fark kabul edilebilir eşik değerinin üzerine çıkması durumu, eksponansiyel artan bir şekilde cezalandırılmaktadır.

Ceza fonksiyonlarındaki s ve r parametreleri yerine sırasıyla -0.75 ve 0.01 değerleri konulduğunda elde edilen gerçek ceza fonksiyonları değerleri aşağıdaki grafiklerde verilmektedir.



Şekil 4.4 Çevrim içi, çevrimler arası ve kanal kazancı ceza fonksiyonu değerleri

Frekans atama işlemi yukarıda belirtilen üç amaç fonksiyonu dikkate alınarak gerçekleştirilmektedir. Bu amaç fonksiyonlarından çevrim içi ve çevrimler arası ceza fonksiyonları, her bir çevrim içerisindeki kullanıcılara atanan frekans değerlerinin birbirine yakın değerler olarak seçilmesini isterken; kanal kazancı ceza fonksiyonu ise

bu durumun tam tersi olarak çevrim içi kullanıcılara atanan frekans değerlerinin birbirinden uzak değerler çıkmasını amaçlamaktadır. Böylelikle üzerinde çalışılan problemi birbirleri ile çelişen amaçları olan çok amaçlı bir optimizasyon problemi olarak tanımlayabiliriz.

4.2 Benzetim Sonuçları

Bu bölümde, verilen amaç fonksiyonları ve kısıtlar doğrultusunda, Frekans Atama Problemi'nin Bölüm 3.2'de anlatılan metasezgisel yöntemlerle çözüm performansları karşılaştırılacaktır. Algoritmaların tek amaç olduğu durumda gösterdiği başarımlar verilecek ve algoritmalarda kullanılan diğer parametreler sabit tutularak, popülasyondaki kromozom ve sürüdeki parçacık sayısının 50, 100 ve 150 olacak şekilde değiştirilmesiyle, 50 bağımsız Monte Carlo koşusu sonrasında elde edilen sonuçlar gösterilecektir. Her bir algoritma, farklı nesil büyüklükleri için ilk nesilleri aynı olacak şekilde başlatılarak performansları kıyaslanmıştır. Son olarak seçilen bir algoritma yardımıyla Bölüm 3.3.2'de verilen skalarizasyon yaklaşımlarının etkisinin gözlemlenmesine yönelik benzetim sonuçları verilecektir. Benzetim sonuçları Intel Core i3 4150 CPU 3.5GHz 4GB bir masaüstü bilgisayar ile MATLAB2010a ortamında elde edilmiştir.

4.2.1 Genetik algoritma

Bölüm 3.2.1'de anlatılan metasezgisel bir yöntem olan Genetik Algoritma ile Frekans Atama Problemi'nin çözümü için iki algoritma versiyonu kullanılmıştır. Bu iki versiyona ait kontrol parametresi bilgileri çizelge 4.2 verilmektedir.

Çizelge 4.2 Genetik algoritma versiyonlarında kullanılan kontrol parametreleri

Kontrol Parametreleri	Genetik Algoritma Versiyon 1 (GA V1)	Genetik Algoritma Versiyon 2 (GA V2)
Popülasyon Büyüklüğü	Sabit	Sabit
Bitirme Kriteri	Yineleme Sayısı	Yineleme Sayısı
Başlatma	Rastgele	Rastgele
Komşuluk	Karesel Izgara	Karesel Izgara
Seçme Mekanizması	Rulet Çarkı	Rulet Çarkı
Elitizm	Yok	Yok
Çaprazlama	Tek Nokta	Tek Nokta
Çaprazlama Oranı	Sabit	Sabit
Mutasyon	Çözüm başına	Her bir çevrim içi frekans başına
Mutasyon Oranı	Sabit	Sabit
Özel Operatörler	Yok	Yok
Amaçlar	Çevrim içi ceza	Çevrim içi ceza
	Çevrimler arası ceza	Çevrimler arası ceza
	Kanal kazancı cezası	Kanal kazancı cezası
Skalarizasyon Yaklaşımı	Ağırlıklandırılmış Toplam	Ağırlıklandırılmış Toplam
	Temelli Yaklaşım	Temelli Yaklaşım

GA V1 ile GA V2 arasındaki temel fark mutasyon parametresidir. GA V1’de, her bir çözüm başına mutasyon işlemi uygulanıp uygulanmayacağı kontrol edildikten sonra hangi frekans değerinin mutasyona uğrayacağı seçilirken; bu işlem GA V2’de ise her bir çevrimdeki çözüm için ataması gerçekleştirilen her bir frekans değerinin mutasyona maruz kalıp kalmayacağının kontrol edilmesi esasına dayanmaktadır.

Çizelge 4.3 Problem tanımında ve algorithmada kullanılan sabitler

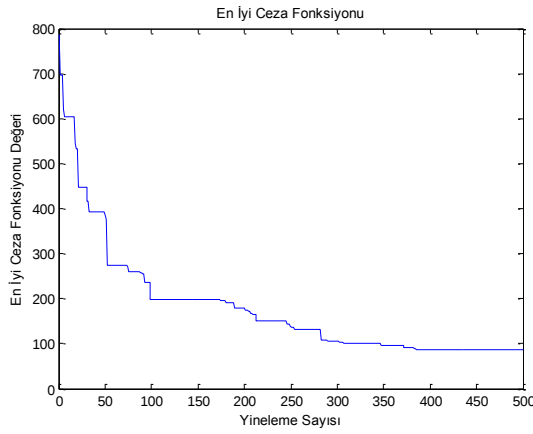
GA V1		GA V2	
Problem Tanımı ve Sabit Parametreler		Problem Tanımı ve Sabit Parametreler	
Kullanılabilir Frekans Sayısı	80	Kullanılabilir Frekans Sayısı	80
Kullanılabilir Frekans Seti	{225-304 MHz} kümesindeki tüm frekanslar	Kullanılabilir Frekans Seti	{225-304 MHz} kümesindeki tüm frekanslar
Çevrim Sayısı	10	Çevrim Sayısı	10
Her Bir Çevrimdeki Kullanıcı Sayısı	10	Her Bir Çevrimdeki Kullanıcı Sayısı	10
Çaprazlama Oranı	0.4	Çaprazlama Oranı	0.4
Mutasyon Oranı	0.6	Mutasyon Oranı	0.006
Yineleme Sayısı	500	Yineleme Sayısı	500

Problem tanımında verilen kullanılabilir frekans sayısı değeri 80 olarak alındığında, diğer tüm algoritma ve problem parametreleri sabit tutularak 50 Monte Carlo sonucunda ulaşılan sonuçlar aşağıda verilmektedir.

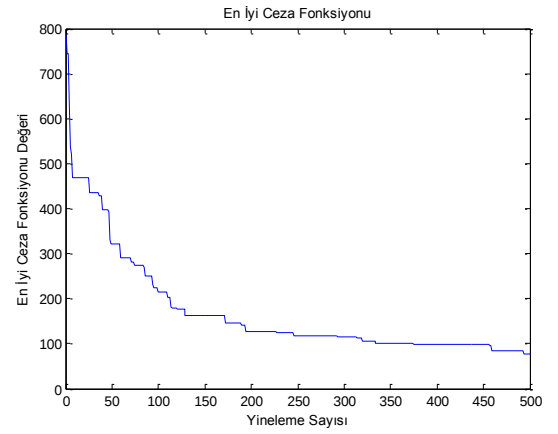
- **Amaç fonksiyonu ağırlıklarının etkisi**

Popülasyondaki kromozom sayısı 50 olacak şekilde amaç fonksiyonlarının birinin 1, diğer ikisinin ise 0 olarak ağırlıklandırıldığı, yani tek bir amaç için optimizasyon gerçekleştirildiğinde 50 bağımsız Monte Carlo koşusu sonrasında çıkan sonuçlar sunulacaktır.

Popülasyondaki Kromozom Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	1	0	0

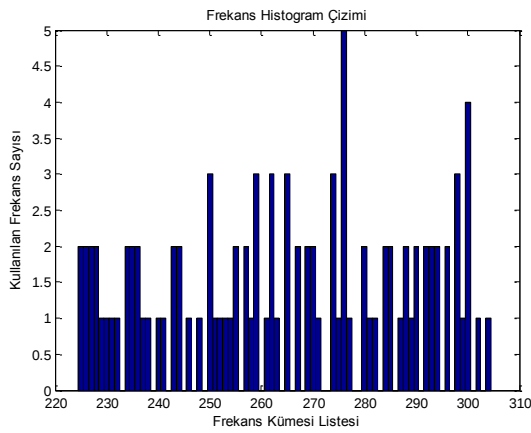


a

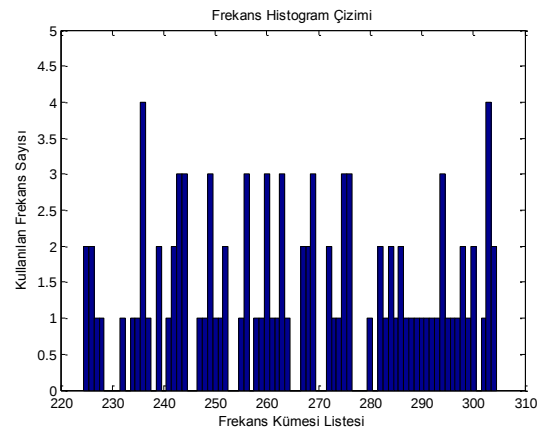


b

Şekil 4.5.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



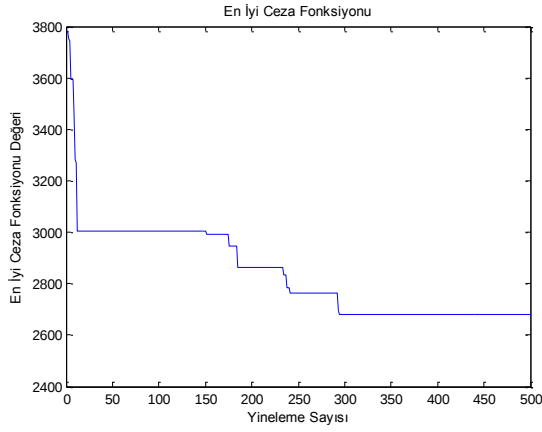
a



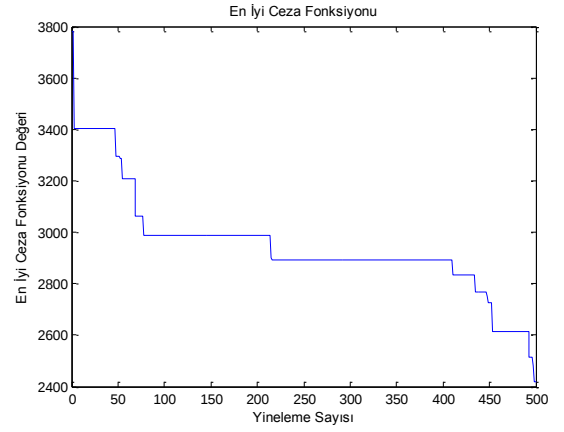
b

Şekil 4.6.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı, b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Popülasyondaki Kromozom Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0	1	0

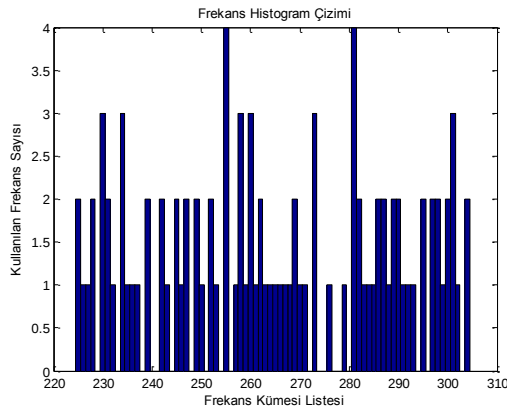


a

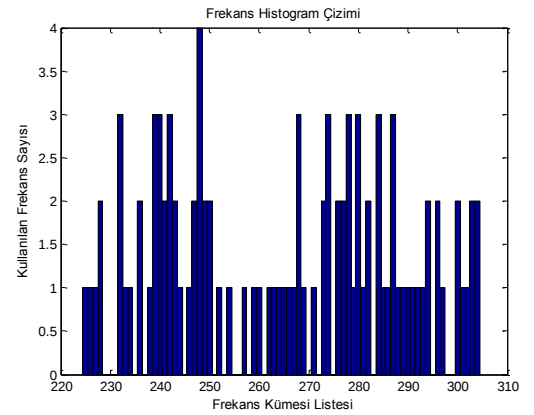


b

Şekil 4.7.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



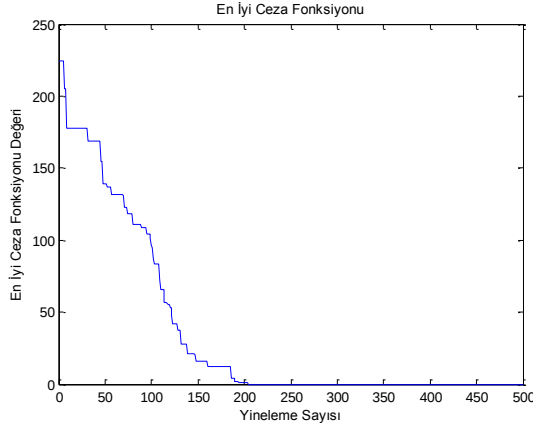
a



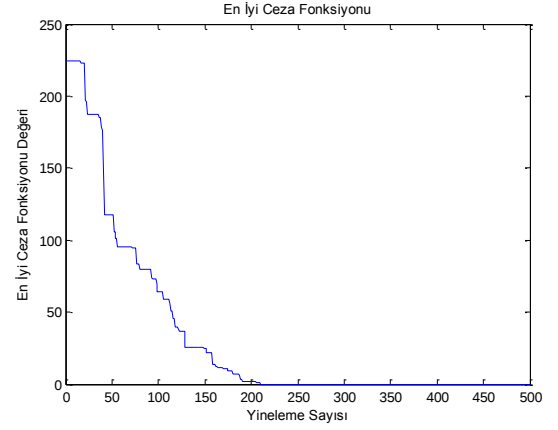
b

Şekil 4.8.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı, b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Popülasyondaki Kromozom Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0	0	1

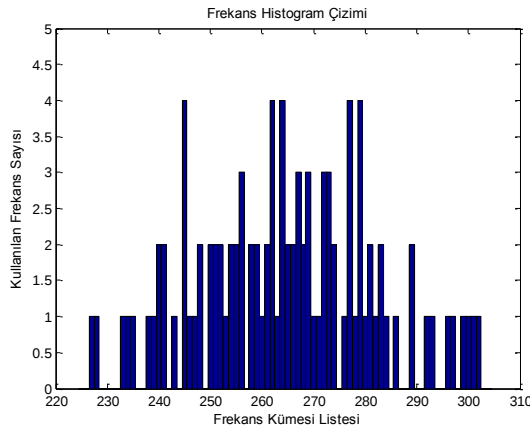


a

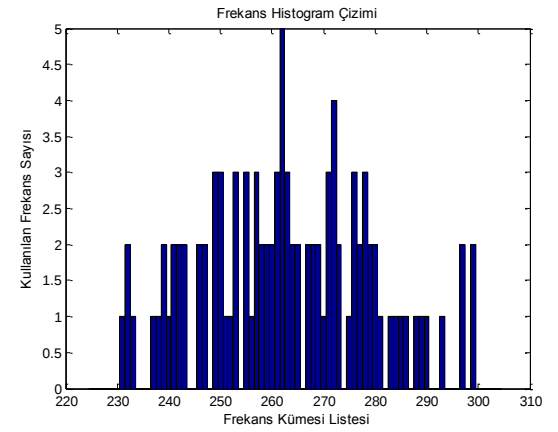


b

Şekil 4.9.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.10.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

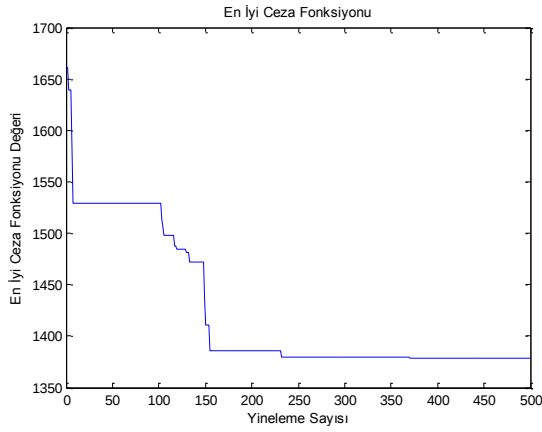
GA V1 ile GA V2 algoritmalarının frekans seti içerisinde kullanılabilir 80 tane frekans değerine sahip olduğu senaryodaki performansları kıyaslanmıştır.

İlk olarak GA V1 ile GA V2 algoritmalarının tek amaç için gösterdikleri performanslar incelenmiştir. Şekil 4.5 ile 4.9'a bakıldığında her iki algoritma versiyonunun çevrim içi ceza için yaklaşık %90 ve kanal kazancı ceza amacı için %100 iyileştirme sağladığı görülmektedir. Şekil 4.7'den görüldüğü üzere çevrimler arası ceza amacı için ise GA V1 %30 iyileştirme sağlarken, GA V2 % 36'lık bir iyileştirme gerçekleştirmektedir.

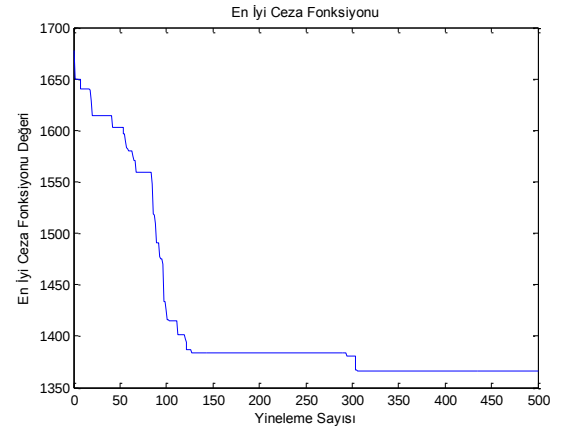
- **Popülasyondaki kromozom sayısı etkisi**

Diğer parametreler sabit tutularak popülasyondaki kromozom sayısının 50, 100 ve 150 olacak şekilde değiştirilmesiyle, 50 bağımsız Monte Carlo koşusu sonrasında elde edilen sonuçlar aşağıdaki gibidir.

Popülasyondaki Kromozom Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

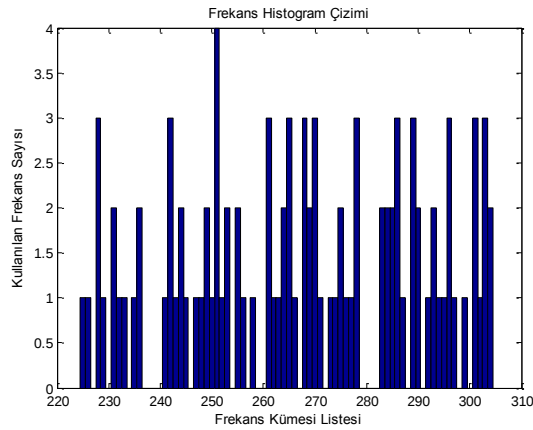


a

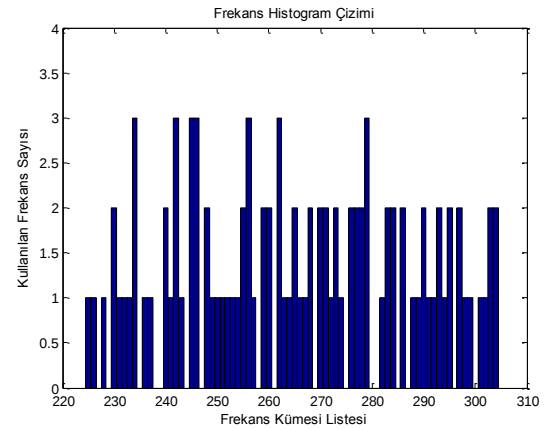


b

Şekil 4.11.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.12.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

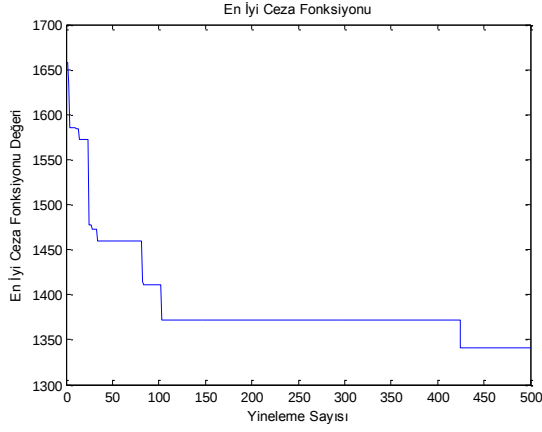
Çizelge 4.4 Kullanılmayan frekans sayısı

GA V1	GA V2
22	17

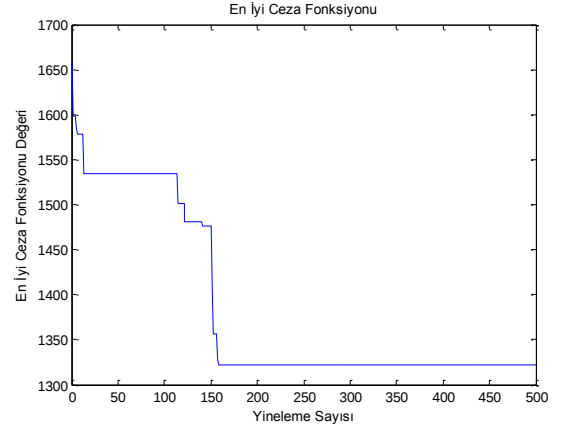
Çizelge 4.5 Çevrimlerde kullanılan frekans listesi

GA V1				GA V2			
Her Bir Çevrim İçin Frekans Listesi(MHz)				Her Bir Çevrim İçin Frekans Listesi(MHz)			
226	247	227	231	233	234	230	242
252	253	232	234	236	256	240	242
256	261	238	241	261	263	245	272
263	290	257	277	265	270	276	282
295	304	279	281	274	295	283	290
228	234	225	251	231	232	293	295
242	257	261	264	243	258	225	234
265	276	265	269	269	271	230	253
281	285	275	280	273	280	236	245
294	300	287	292	293	299	255	260
229	233	226	253	228	230	265	273
246	264	256	258	236	239	279	286
269	279	259	262	246	247	271	297
289	291	266	281	251	265	228	246
297	301	283	293	269	301	234	246
229	236	244	254	255	261	249	262
	295	296	302		285	277	279
						284	293
						294	304
						232	246
						248	257
						266	270
						273	278
						299	303
						226	231
						234	240
						241	250
						254	254
						268	286
						302	

Popülasyondaki Kromozom Sayısı	100		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

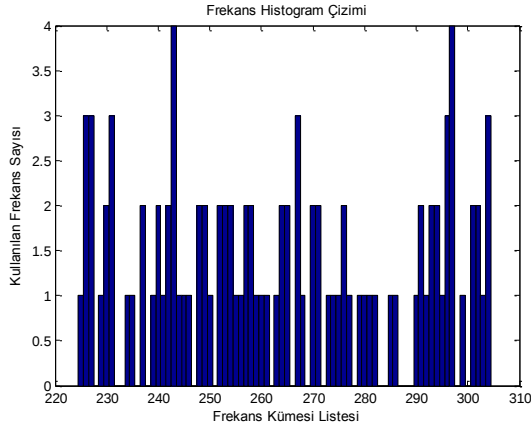


a

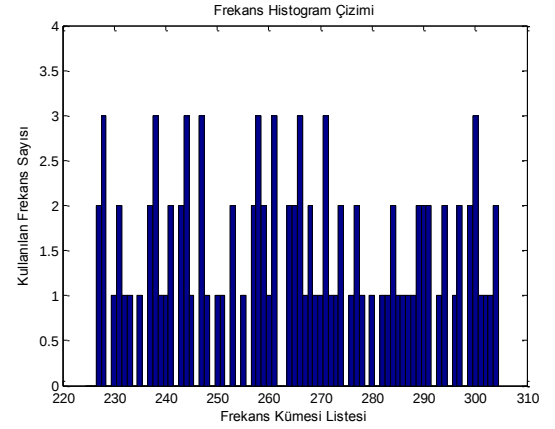


b

Şekil 4.13.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.14.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

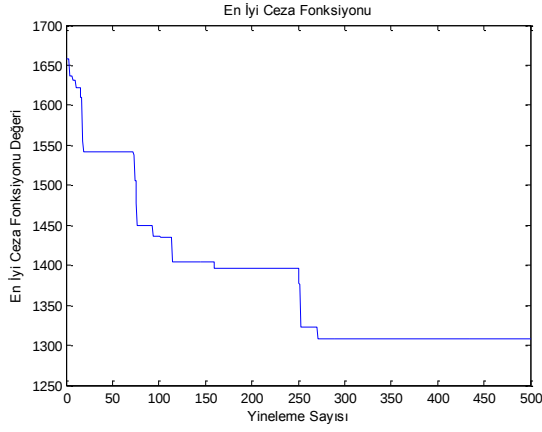
Çizelge 4.6 Kullanılmayan frekans sayısı

GA V1	GA V2
19	19

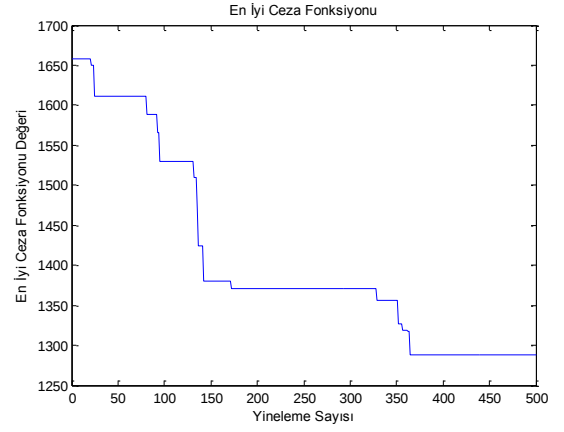
Çizelge 4.7 Çevrimlerde kullanılan frekans listesi

GA V1						GA V2					
Her Bir Çevrim İçin Frekans Listesi(MHz)						Her Bir Çevrim İçin Frekans Listesi(MHz)					
229	230	226	235	243	249	227	228	230	231	240	243
241	243	239	252	255	257	241	257	233	237	253	255
248	258	271	275	261	270	258	259	238	260	257	266
264	267	276	293	286	294	266	293	261	266	288	289
281	296	296	304	297	301	294	304	271	304	294	297
227	231	226	227	226	234	228	237	238	244	235	241
243	245	230	237	242	246	239	244	247	261	247	250
253	254	256	265	267	276	248	258	265	272	259	264
260	265	280	293	279	294	271	274	283	284	268	270
268	302	297	301	299	302	282	287	290	291	300	303
231	237	225	240	227	231	231	238	228	243	227	265
242	244	243	253	240	248	244	261	245	251	274	277
263	270	254	282	249	257	268	269	253	264	278	280
277	292	290	291	264	271	290	297	267	271	284	289
297	304	295	296	291	304	299	300	299	302	291	300
250	252	258	259	267	273	274	232	247	258	273	276
		285	297	303					286	296	301

Popülasyondaki Kromozom Sayısı	150		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

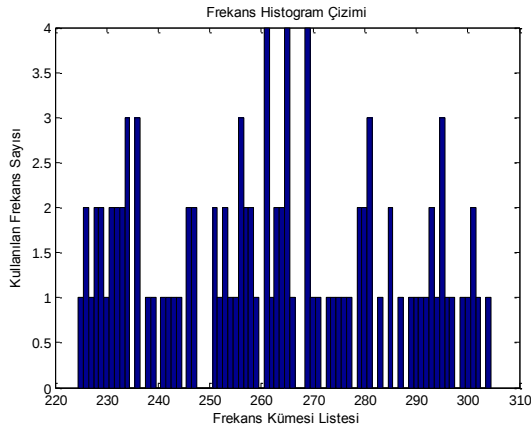


a

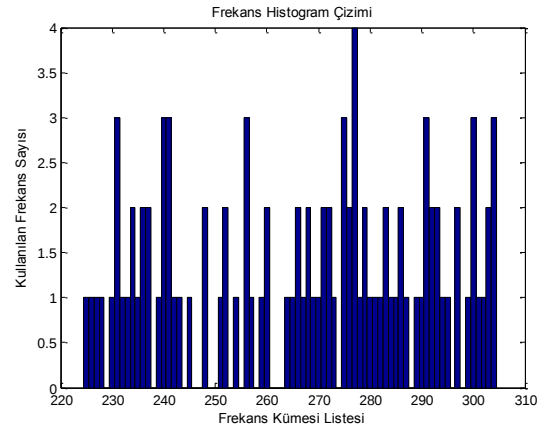


b

Şekil 4.15.a.GA V1: Yineleme sayısı - En iyi amaç fonksiyonu değeri b. GA V2: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.16.a.GA V1: Frekans listesi - Atanan frekans kullanım sayısı b. GA V2: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Çizelge 4.8 Kullanılmayan frekans sayısı

GA V1	GA V2
18	17

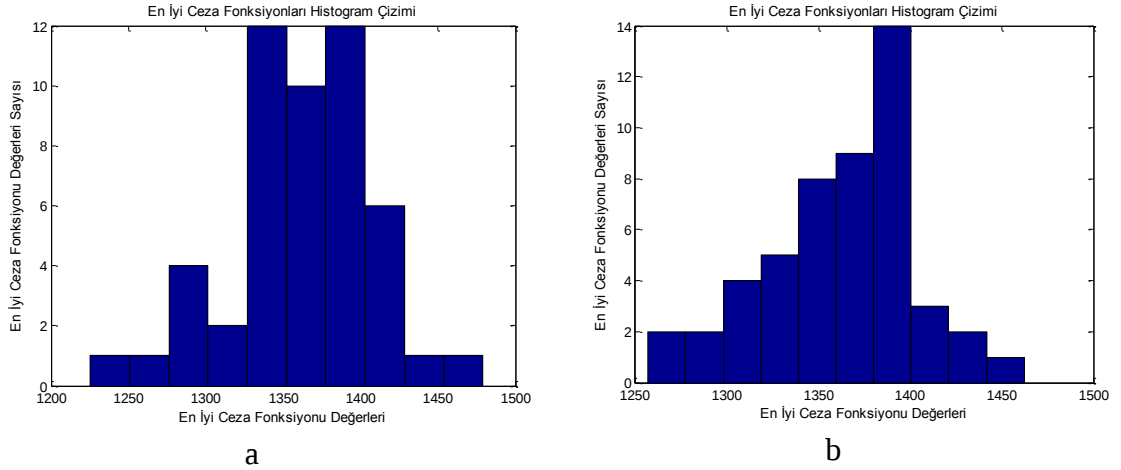
Çizelge 4.9 Çevrimlerde kullanılan frekans listesi

GA V1						GA V2					
Her Bir Çevrim İçin Frekans Listesi(MHz)						Her Bir Çevrim İçin Frekans Listesi(MHz)					
226	247	227	231	233	234	234	242	231	240	233	241
252	253	232	234	236	256	252	256	252	254	251	257
256	261	238	241	261	263	265	277	277	279	266	271
263	290	257	277	265	270	286	290	280	283	286	291
295	304	279	281	274	295	292	301	294	300	292	299
228	234	225	251	231	232	228	241	226	232	225	230
242	257	261	264	243	258	248	267	234	236	236	248
265	276	265	269	269	271	270	272	240	256	269	272
281	285	275	280	273	280	276	279	264	289	283	297
294	300	287	292	293	299	285	300	291	303	302	304
229	233	226	253	228	230	235	237	227	231	237	241
246	264	256	258	236	239	239	260	240	243	256	260
269	279	259	262	246	247	268	271	259	268	266	273
289	291	266	281	251	265	275	281	275	277	277	278
297	301	283	293	269	301	293	304	293	300	291	303
229	236	244	254	255	261	231	245	275	276	282	284
		295	296	302	285			295	297	304	287

GA V1 ile GA V2 algoritmalarının kullanılabilir 80 tane frekans değerine sahip olduğu senaryodaki performansları şekil 4.11, şekil 4.13 ve şekil 4.15’de verilmiştir. Popülasyondaki kromozom sayısı değerleri 50, 100 ve 150 olacak şekilde artırılarak, GA V1 ile GA V2 algoritma performansları incelendiğinde, GA V1 için sırasıyla %18.2, %19 ve %21.2; GA V2 için %19, %20, %22.3’lük iyileştirmeler gerçekleşmiştir.

Sırasıyla GA V1 için 22, 19, 18 ve GA V2 için 17, 19, 17 tane frekans seti içerisinde kullanılmayan frekans değeri vardır.

GA V1 ile GA V2 algoritmalarının gürbüzlüğünün karşılaştırılması için popülasyonda 50 kromozom kullanarak, 50 Monte Carlo sonucunda elde edilen amaç fonksiyonu minimum değerleri tutulmuştur. Tutulan bu değerlerin histogramları çizdirildiğinde ortaya çıkan grafikler aşağıda verilmektedir.



Şekil 4.17.a.GA V1, b. GA V2 için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları

Çizelge 4.10 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler

	GA V1	GA V2
Ortalama Değer	1373.7	1359.3
Standart Sapma Değeri	45.52	42.2
Minimum Değer	1225.3	1256.9
Maksimum Değer	1479.2	1462.2

Çizelge 4.10'daki minimum amaç fonksiyonları ortalama değerlerinden görüldüğü üzere GA V1 ve GA V2 algoritmaları neredeyse birbirlerine yakın performans sergilemektedirler. Algoritmaların gürbüzlüğünü incelemek için standart sapma değerlerini incelendiğinde, performanslarında olduğu gibi birbirlerine çok yakın sonuçlar elde edilmiştir.

4.2.2 Parçacık sürüsü optimizasyonu

Literatürde yapılan çalışmalar incelendiğinde genel olarak kesikli karar değişkenlerine sahip optimizasyon problemleri için metasezgisel yöntemler arasında genetik algoritmanın doğasının bu tür problemler için daha uygun olmasından dolayı tercih edildiği; sürekli karar değişkenlerine sahip optimizasyon problemlerinde ise parçacık sürüsü optimizasyonu algoritma yapısından dolayı, parçacık sürüsü optimizasyonunun daha uygun olduğu görülmektedir.

Frekans Atama Problemi karar değişkenine göre kesikli bir optimizasyon problemidir. Bu yüzden PSO algoritmasının bu problemde kullanılabilmesi için Bölüm 3.2.2’de anlatılan algoritmanın genel yapısı korunarak bir takım eklemeler gerçekleştirilmiştir. Frekans Ataması Problemi çözümünde kullanılacak olan Yuvarlatılmış Parçacık Sürüsü Optimizasyonu (YPSO) yönteminde şekil 3.9’da verilen PSO akış diyagramı üzerinde herhangi bir değişikliğe gidilmeden, algoritma sonlandırma kriteri sağlandığında ortaya çıkan sonuçların yuvarlanmasıyla “En İyi Çözüm” değerlerine ulaşılmaktadır. Uygulanan bu değişikliklerle beraber elde edilen sonuçlarda bazı frekans değerlerinin yuvarlama neticesinde aynı değeri aldığı görülmüş ve bu sorun için aynı çıkan frekans değerlerinden birisinin yuvarlama üst limitine, diğerinin ise yuvarlama alt limitine yuvarlanması sağlanarak sorun giderilmiştir.

Çizelge 4.11’de YPSO algoritmasına ait çeşitli parametre bilgileri verilmektedir.

Çizelge 4.11 YPSO algoritmasına ait başlangıç parametreleri

Başlangıç Parametreleri	YPSO
Parçacık Sayısı	Sabit
Bitirme Kriteri	Yineleme Sayısı
Başlatma	Rastgele
Komşuluk	Karesel Izgara
Arama Bölgesi Büyüklüğü	Sabit
Maksimum Hız	Sabit (Eşitlik (3.5))
Atalet Ağırlığı	Lineer Azalan (Eşitlik (3.4))
Kişisel Öğrenme Faktörü	1.491
Sosyal Öğrenme Faktörü	1.491
Amaçlar	Çevrim içi ceza
	Çevrimler arası ceza
	Kanal kazancı cezası
Skalarizasyon Yaklaşımı	Ağırlıklandırılmış Toplam Temelli Yaklaşım

Frekans Atama Problemi için kullanılan YPSO algoritmasında performans iyileştirilmesi sağlama adına üç farklı uygulama geliştirilmiştir:

- Sınırsız Yuvarlatılmış Parçacık sürüsü Optimizasyonu
- Yansımali Yuvarlatılmış Parçacık sürüsü Optimizasyonu
- Sıralamalı Sınırsız Yuvarlatılmış Parçacık sürüsü Optimizasyonu

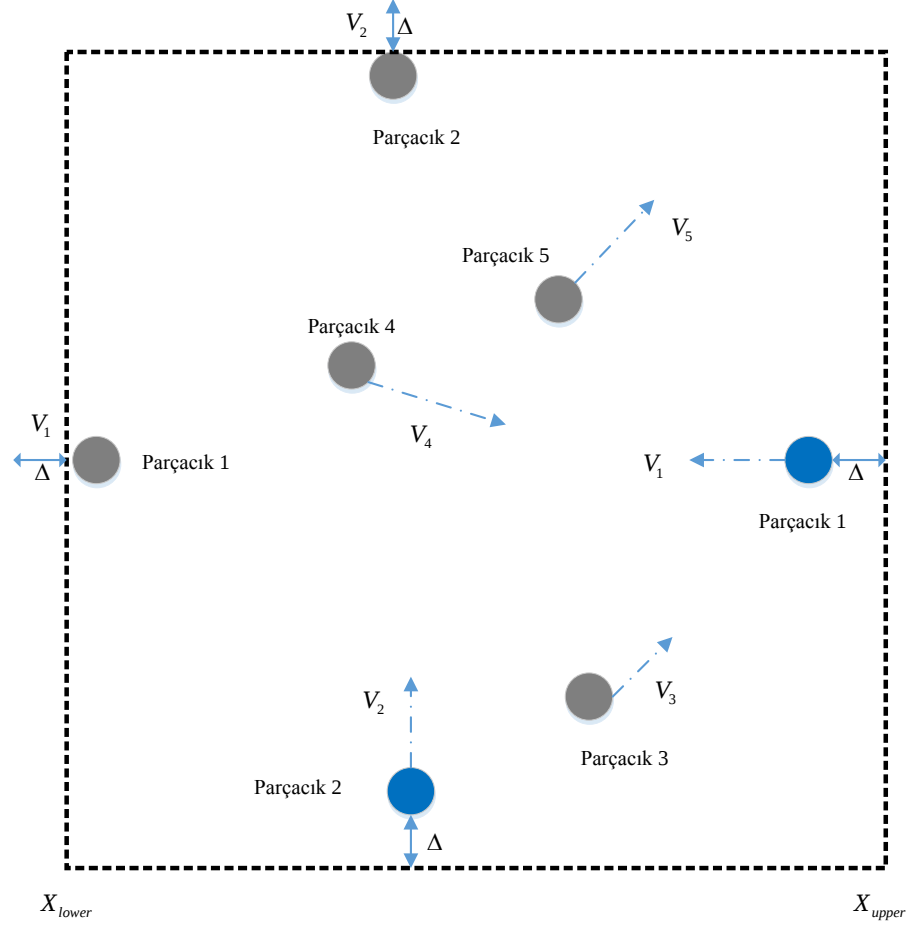
- Sınırsız yuvarlatılmış parçacık sürüsü optimizasyonu

Çizelge 4.12 Sınırsız YPSO Algoritması

Do For <i>başlangıç parametreleri belirle ve her parçacık için rastgele hız-pozisyon ataması</i> End Do For <i>her parçacık için uygunluk değerini hesapla</i> If (uygunluk değeri < "Kişisel En İyi"($x_{pbest_i}^{(d)}$) uygunluk değeri) <i>şimdiki değeri yeni , "Kişisel En İyi" olarak ata</i> End If End <i>Tüm parçacıkların bulunduğu , "Kişisel En İyi" değerlerinden</i> <i>En iyisini, tüm parçacıkların "Global En İyi"($x_{gbest}^{(d)}$) olarak ayarla</i> For her parçacık için <i>"Kişisel En İyi" ve "Global En İyi" parçacıkları ile güncellenmiş hız $V_{particle}$ ve</i> <i>konum $X_{particle}$ değerlerinin hesapla.</i> End	1
If (Parçacık Konumu($X_{particle}$) > Uzak Üst Sınırı(X_{Upper})) $\Delta = X_{particle} - X_{upper}$ $X_{particle} = X_{lower} + \Delta$ Else If (Parçacık Konumu($X_{particle}$) < Uzak Alt Sınırı(X_{lower})) $\Delta = X_{lower} - X_{particle}$ $X_{particle} = X_{upper} - \Delta$ End If	2
While (maksimum yineleme sayısına kadar devam et) Elde edilen "Global En İyi"($x_{gbest}^{(d)}$) değerini yuvarla	3

Çizelge 4.12’de Sınırsız YPSO algoritmasına ait akış verilmiştir. Yansımali YPSO algoritması için YPSO algoritması akışına çizelge 4.12’deki 2 numaralı parça eklenmiştir. Parçacık konumları, parçacık hız değerlerinden dolayı arama uzayının alt ve üst limitleri dışına çıkabilmektedir. Bu gerçekleşen dezavantajı avantaja dönüştürmek için algoritmaya sınırsızlık operatörü eklenmiştir. Böylelikle GA’daki

mutasyon gibi arama uzayının başka uzak noktalarında da çözüm araştırılmasını sağlamaktadır.



Şekil 4.18 Sınırsız YPSO algoritması yansıma geometrisi

- Sıralanmış sınırsız yuvarlatılmış parçacık sürüsü optimizasyonu

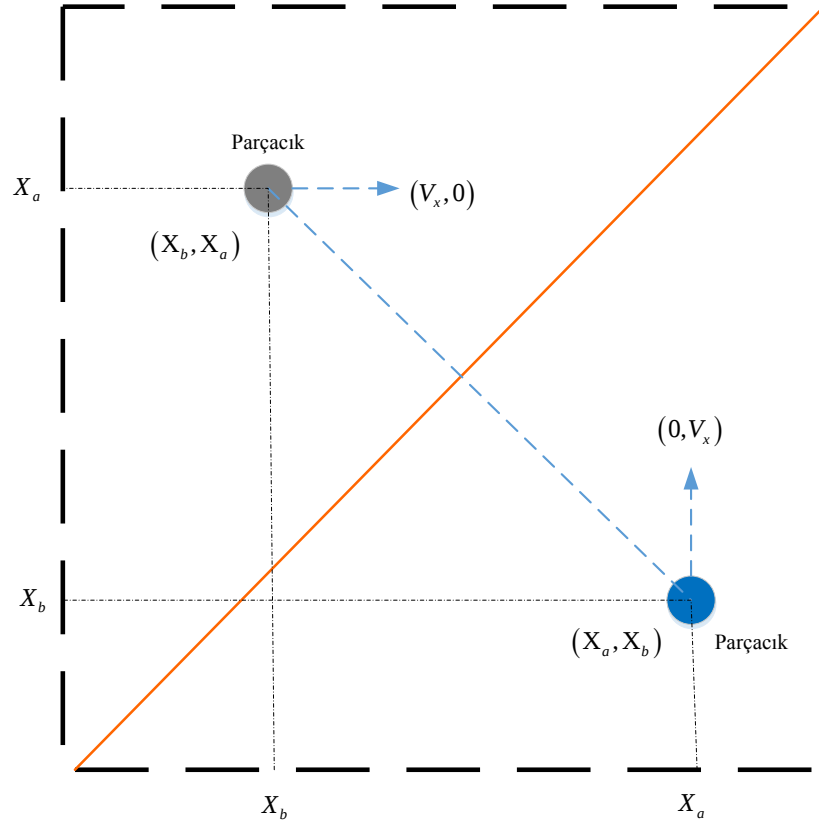
Çizelge 4.13 Sıralanmış Sınırsız YPSO algoritması

Do For <i>başlangıç parametreleri belirle ve her parçacık için rastgele hız-pozisyon ataması</i> End Do For <i>her parçacık için uygunluk değerini hesapla</i> If (uygunluk değeri < "Kişisel En İyi"($x_{pbest_i}^{(d)}$) uygunluk değeri) <i>şimdiki değeri yeni , "Kişisel En İyi" olarak ata</i> End If End <i>Tüm parçacıkların bulunduğu , "Kişisel En İyi" değerlerinden</i> <i>En iyisini, tüm parçacıkların "Global En İyi"($x_{gbest}^{(d)}$) olarak ayarla</i> For her parçacık için <i>"Kişisel En İyi" ve "Global En İyi" parçacıkları ile güncellenmiş hız $V_{particle}$ ve</i> <i>konum $X_{particle}$ değerlerinin hesapla.</i> End	1
Sort <i>Konum değeri $X_{particle}$</i> <i>Sıralanmış konum değerlerine göre,</i> <i>Hız değerlerini ata</i>	2
If (Parçacık Konumu($X_{particle}$)>Uzay Üst Sınırı(X_{Upper})) $\Delta = X_{particle} - X_{upper}$ $X_{particle} = X_{lower} + \Delta$ Else If (Parçacık Konumu($X_{particle}$)<Uzay Alt Sınırı(X_{lower})) $\Delta = X_{lower} - X_{particle}$ $X_{particle} = X_{upper} - \Delta$ End If	3
While (maksimum yineleme sayısına kadar devam et) <i>Elde edilen "Global En İyi"($x_{gbest}^{(d)}$) değerini yuvarla</i>	4

Çizelge 4.13’de Sıralanmış Sınırsız YPSO algoritmasına ait akış verilmiştir. Sınırsız YPSO algoritması için Sınırsız YPSO algoritması akışına çizelge 4.13’deki 2 numaralı parça eklenmiştir. Frekans Atama Problemi için,

$$\begin{aligned}
 X^i &= [f_1^i, f_2^i, \dots, f_{10}^i] \quad i=1,2,\dots,10 \\
 \text{sort}(X^i) &\text{ öyle ki } f_7^i < f_8^i < \dots < f_1^i \\
 X^i &= [f_7^i, f_8^i, \dots, f_1^i] \\
 V^i &= [v_7^i, v_8^i, \dots, v_1^i]
 \end{aligned} \tag{4.8}$$

Şeklinde uygulanmaktadır.



Şekil 4.19 Sıralanmış Sınırsız YPSO algoritması sıralama geometrisi

Algoritmaya şekil 4.19’da geometrisi verilen sıralama operatörü eklenerek, parçacıkların yönlendiği “En İyi” değerler konusunda yaşadığı belirsizliğin giderilmesinin sağlanması ve ayrıca problemin çözüm uzayı içerisinde bulunan gereksiz

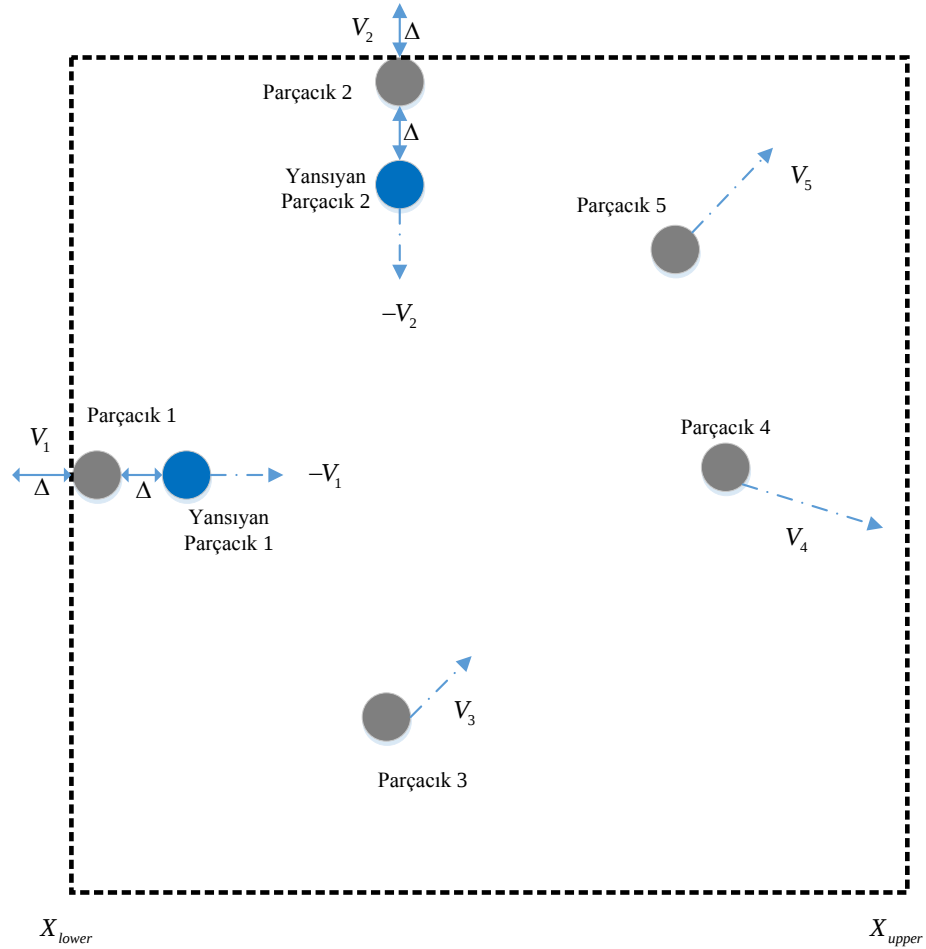
bölgelerin atılmasıyla beraber çözümün yakınsayacağı noktaya daha hızlı bir şekilde ulaşması amaçlanmıştır.

- **Yansımali yuvarlatılmış parçacık sürüsü optimizasyonu**

Çizelge 4.14 Yansımali YPSO Algoritması

Do For <i>başlangıç parametreleri belirle ve her parçacık için rastgele hız-pozisyon ataması</i> End Do For <i>her parçacık için uygunluk değerini hesapla</i> If (uygunluk değeri < "Kişisel En İyi"($x_{pbest_i}^{(d)}$) uygunluk değeri) <i>şimdiki değeri yeni , "Kişisel En İyi" olarak ata</i> End If End <i>Tüm parçacıkların bulunduğu , "Kişisel En İyi" değerlerinden</i> <i>En iyisini, tüm parçacıkların "Global En İyi"($x_{gbest}^{(d)}$) olarak ayarla</i> For her parçacık için <i>"Kişisel En İyi" ve "Global En İyi" parçacıkları ile güncellenmiş hız $V_{particle}$ ve</i> <i>konum $X_{particle}$ değerlerinin hesapla.</i> End	1
If (Parçacık Konumu($X_{particle}$)>Uzay Üst Sınırı(X_{Upper})) $\Delta = X_{particle} - X_{upper}$ $X_{particle} = X_{upper} - \Delta$ $V_{particle} = -V_{particle}$ Else If (Parçacık Konumu($X_{particle}$)<Uzay Alt Sınırı(X_{lower})) $\Delta = X_{lower} - X_{particle}$ $X_{particle} = X_{lower} + \Delta$ $V_{particle} = -V_{particle}$ End If	2
While (maksimum yineleme sayısına kadar devam et) <i>Elde edilen "Global En İyi"($x_{gbest}^{(d)}$)değerini yuvarla</i>	3

Çizelge 4.14’de Yansımali YPSO algoritmasına ait akış verilmiştir. Yansımali YPSO algoritması için YPSO algoritması akışına çizelge 4.12’deki 2 numaralı parça eklenmiştir. Sınırsız YPSO’da olduğu gibi parçacık konumları, parçacık hız değerlerinden dolayı arama uzayının alt ve üst limitleri dışına çıkabilmektedir. Bu durumu çözüme ulaşmada avantaj sağlamak için algoritmaya yansıma operatörü eklenmiştir.



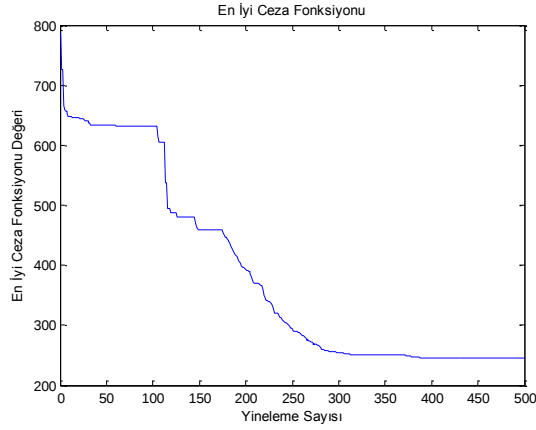
Şekil 4.20 Yansımali YPSO algoritması yansıma geometrisi

Sınırsız YPSO		Yansımali YPSO	
Problem Tanımı ve Sabit Parametreler		Problem Tanımı ve Sabit Parametreler	
Kullanılabilir Frekans Sayısı	80	Kullanılabilir Frekans Sayısı	80
Kullanılabilir Frekans Seti	{225-304 MHz} kümesindeki tüm frekanslar	Kullanılabilir Frekans Seti	{225-304 MHz} kümesindeki tüm frekanslar
Çevrim Sayısı	10	Çevrim Sayısı	10
Her Bir Çevrimdeki Kullanıcı Sayısı	10	Her Bir Çevrimdeki Kullanıcı Sayısı	10
Atalet Ağırlığı	0.9'dan 0.4'e kadar lineer azalan	Atalet Ağırlığı	0.9'dan 0.4'e kadar lineer azalan
Kişisel Öğrenme Faktörü (c1)	1.41	Kişisel Öğrenme Faktörü (c1)	1.41
Sosyal Öğrenme Faktörü (c2)	1.41	Sosyal Öğrenme Faktörü (c2)	1.41
Yineleme Sayısı	500	Yineleme Sayısı	500

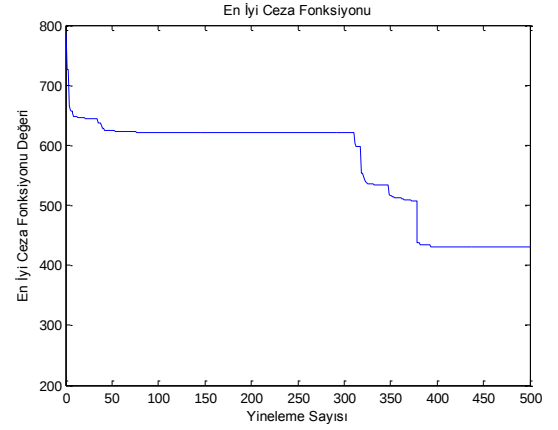
- Amaç fonksiyonu ağırlıklarının etkisi

Sürü parçacık sayısı 50 olacak şekilde, amaç fonksiyonlarının birini 1 olarak ağırlıklandırılırken, diğerlerini ise 0 olarak ağırlıklandırarak, yani tek bir amaç için optimizasyon gerçekleştirildiğinde 50 bağımsız Monte Carlo koşusu sonrasında çıkan sonuçlar aşağıdaki verilmiştir.

Sürüdeki Parçacık Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	1	0	0

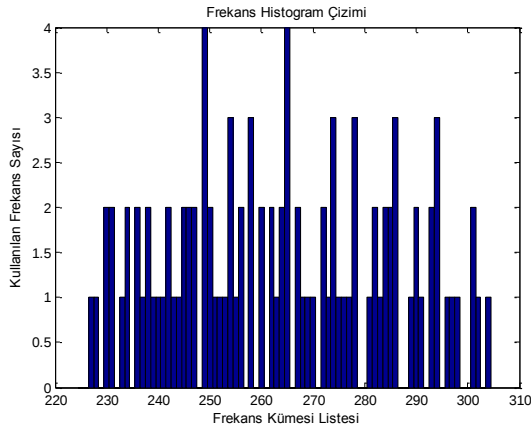


a

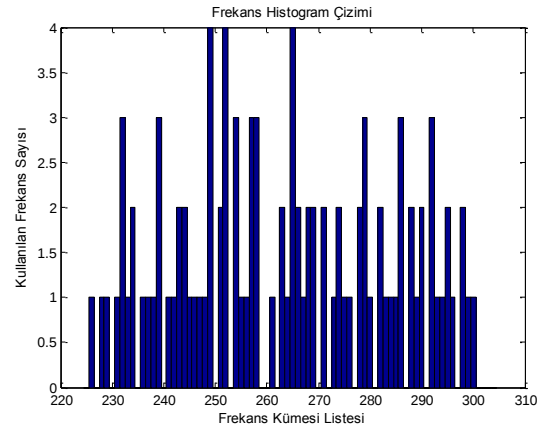


b

Şekil 4.21.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



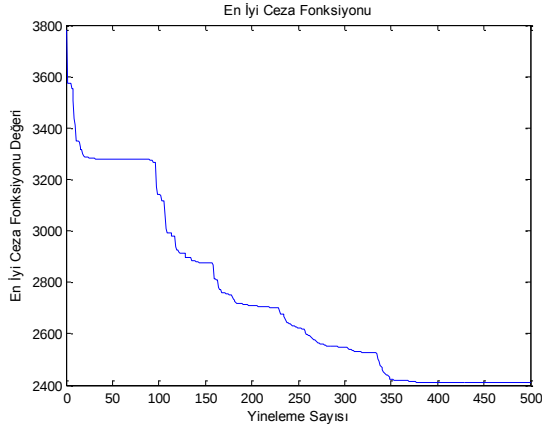
a



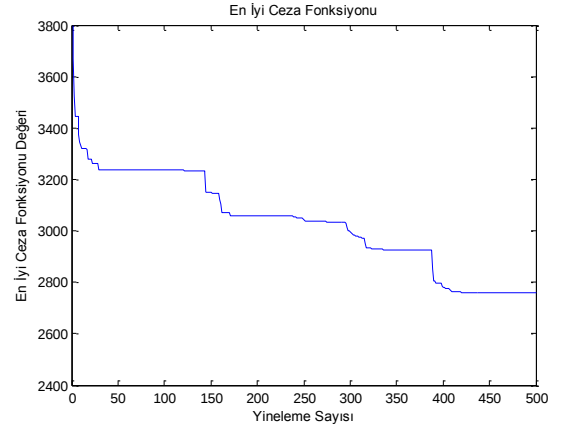
b

Şekil 4.22.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b.Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Sürüdeki Parçacık Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0	1	0

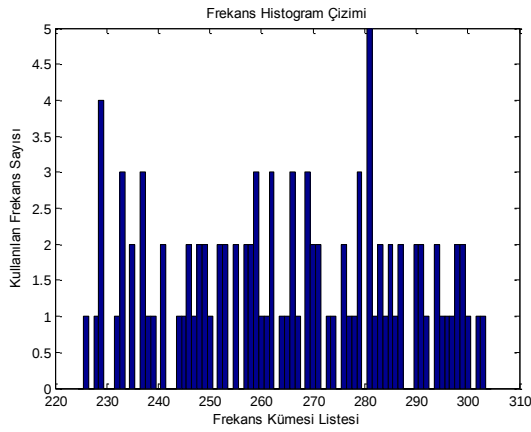


a

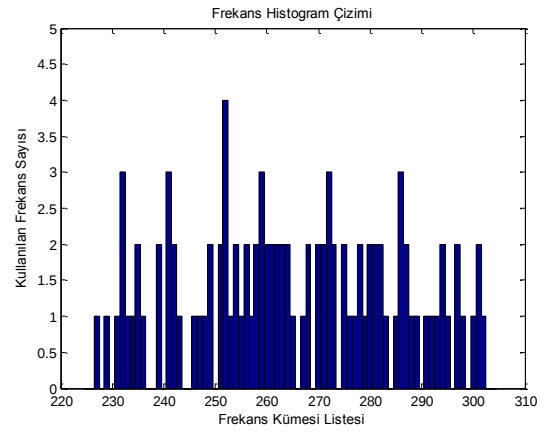


b

Şekil 4.23.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



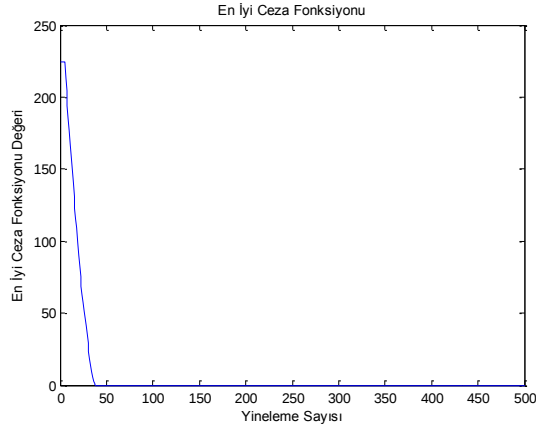
a



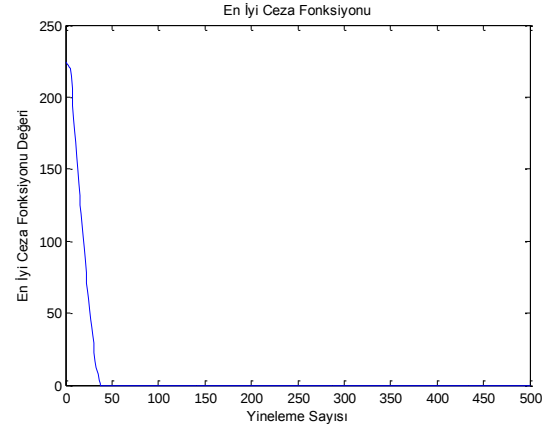
b

Şekil 4.24.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Sürüdeki Parçacık Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0	0	1

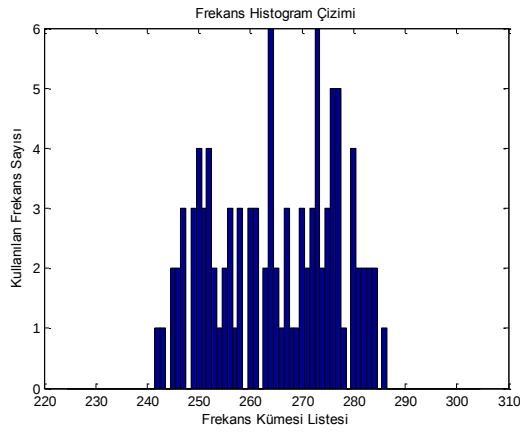


a

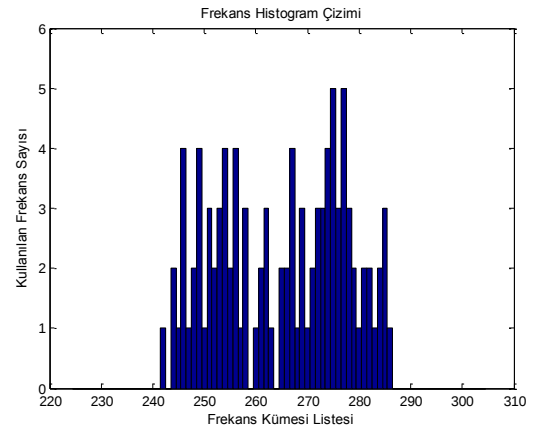


b

Şekil 4.25.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.26.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

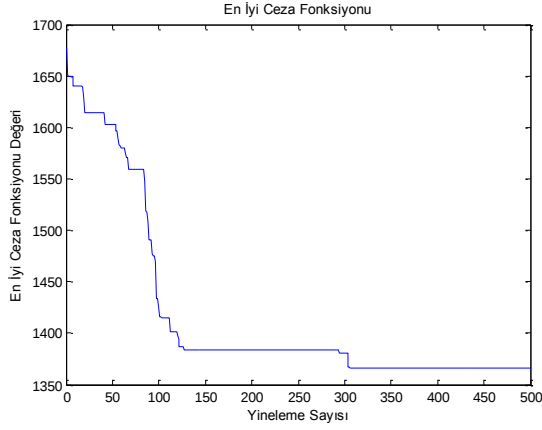
Sınırsız YPSO ile Yansımali YPSO algoritmalarının kullanılabilir 80 tane frekans değerine sahip olduğu senaryodaki performansları kıyaslanmıştır.

Sınırsız YPSO ile Yansımali YPSO algoritmalarının tek amaç için gösterdikleri performanslar incelenmiştir. Şekil 4.21 incelendiğinde Sınırsız YPSO çevrim içi ceza fonksiyonu değerinde %68’lik bir iyileştirme yaparken, Yansımali YPSO %45’lik bir iyileştirme gerçekleştirmektedir. Şekil 4.23’te çevrimler arası ceza amacı için algoritmalar kıyaslanmış, Sınırsız YPSO %37, Yansımali YPSO %28’lik iyileştirme yapmıştır. Şekil 4.25’e bakıldığında her iki algoritma versiyonunun da kanal kazancı ceza amacı için %100 iyileştirme sağladığı görülmektedir.

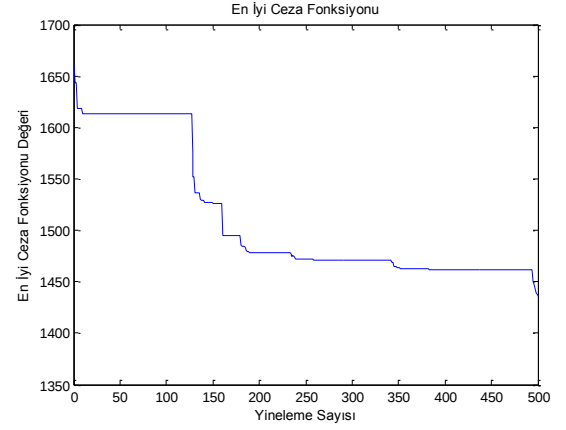
- **Sürüdeki parçacık sayısı etkisi**

Diğer parametreler sabit tutularak sürüdeki parçacık sayısının 50, 100 ve 150 olacak şekilde ayarlanarak, 50 bağımsız Monte Carlo koşusu sonrasında elde edilen sonuçlar aşağıdaki gibidir.

Sürüdeki Parçacık Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

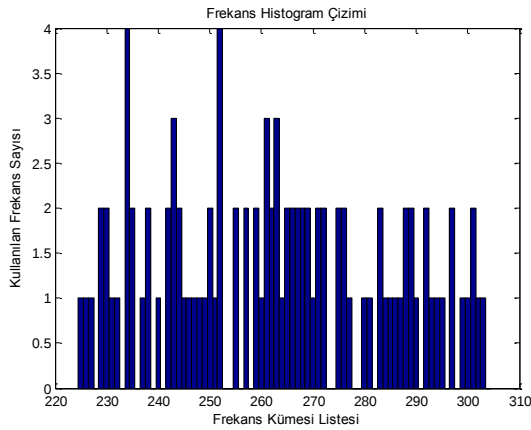


a

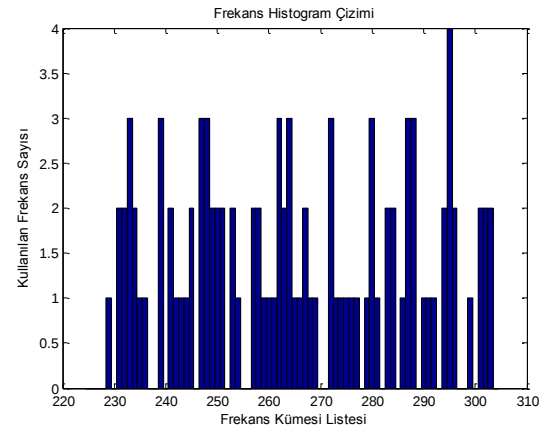


b

Şekil 4.27.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b.Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.28.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

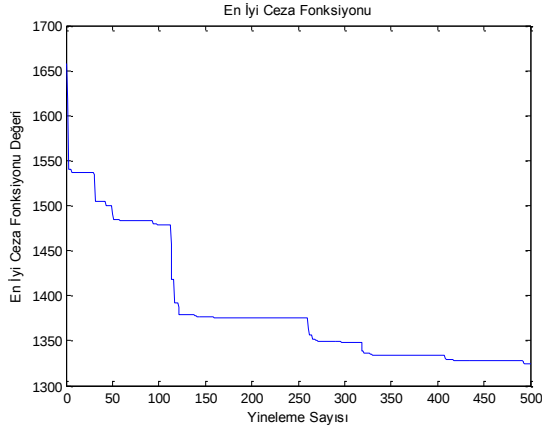
Çizelge 4.15 Kullanılmayan frekans sayısı

Sınırsız YPSO	Yansımali YPSO
18	23

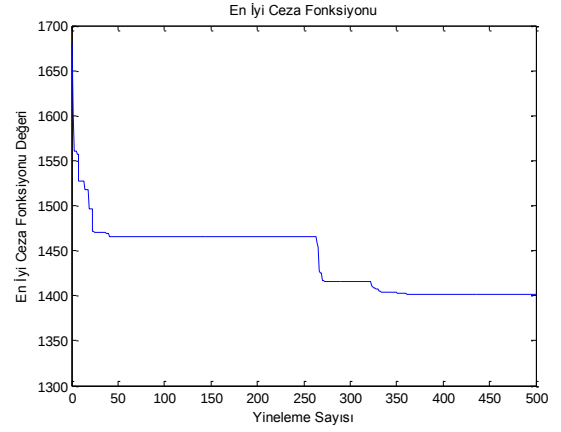
Çizelge 4.16 Çevrimlerde kullanılan frekans listesi

Sınırsız YPSO				Yansımali YPSO			
Her Bir Çevrim İçin Frekans Listesi(MHz)				Her Bir Çevrim İçin Frekans Listesi(MHz)			
230	243	234	243	226	231	248	249
259	267	249	250	238	245	254	257
270	283	252	260	257	263	268	272
285	288	263	268	272	288	273	280
294	301	275	297	292	303	284	287
235	252	235	240	242	251	232	233
257	261	242	244	252	263	239	247
262	263	247	264	265	276	276	277
264	265	276	277	286	289	284	288
272	292	289	302	297	300	295	296
225	230	248	252	227	229	235	239
232	234	259	261	234	235	241	253
243	255	266	267	238	246	260	263
256	269	271	280	250	268	283	301
295	301	281	284	275	290	302	303
229	237	244	266	269	271	229	242
		287	293	299		243	245
						259	263
						274	279
							290

Sürüdeki Parçacık Sayısı	100		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

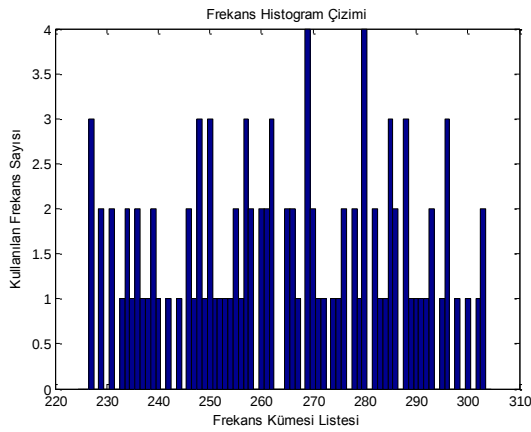


a

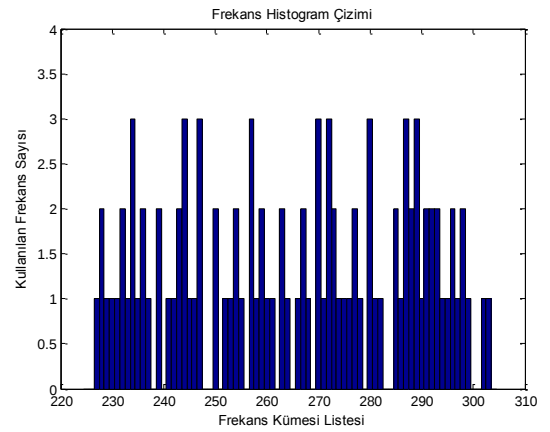


b

Şekil 4.29.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.30.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

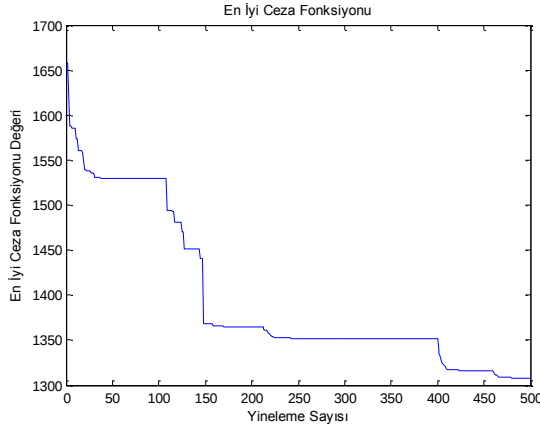
Çizelge 4.17 Kullanılmayan frekans sayısı

Sınırsız YPSO	Yansımali YPSO
20	17

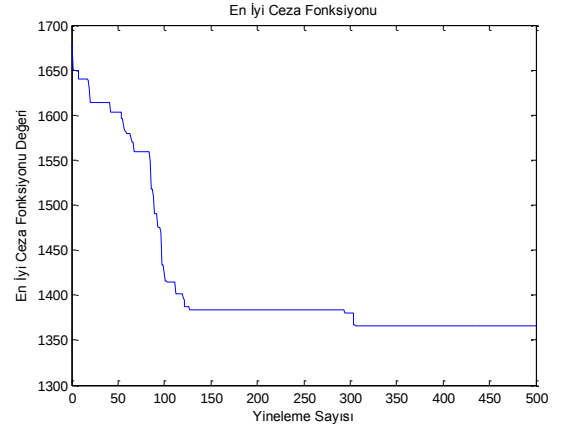
Çizelge 4.18 Çevrimlerde kullanılan frekans listesi

Sınırsız YPSO				Yansımali YPSO			
Her Bir Çevrim İçin Frekans Listesi(MHz)				Her Bir Çevrim İçin Frekans Listesi(MHz)			
236	249	229	253	234	236	234	250
250	252	258	282	239	247	228	230
255	262	285	286	248	257	235	239
263	269	288	296	269	270	247	248
280	292	297	300	284	289	264	273
					293	273	277
					298	287	296
						280	293
227	240	231	248	234	257	227	244
251	255	250	251	261	265	239	242
260	270	261	266	271	276	243	246
280	283	269	279	278	282	260	266
285	302	293	295	286	291	252	257
						272	276
						270	280
						281	288
						289	290
229	233	227	228	231	248	232	237
242	244	235	239	254	258	228	229
256	265	246	257	260	262	234	244
266	280	276	285	269	275	257	267
296	303	289	298	288	303	285	267
						287	292
						268	278
						299	302
						289	292
						296	298
237	238	246	267	270	272	273	277
		278	288	290		287	294
						297	

Sürüdeki Parçacık Sayısı	150		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

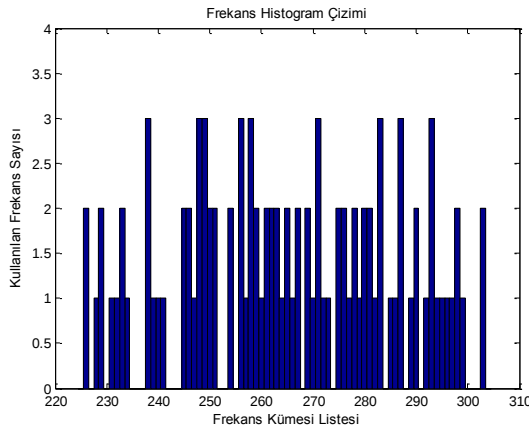


a

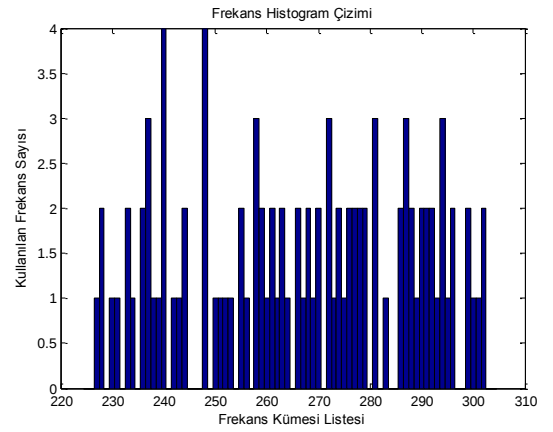


b

Şekil 4.31.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b. Yansımali YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.32.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Yansımali YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Çizelge 4.19 Kullanılmayan frekans sayısı

Sınırsız YPSO	Yansımali YPSO
23	20

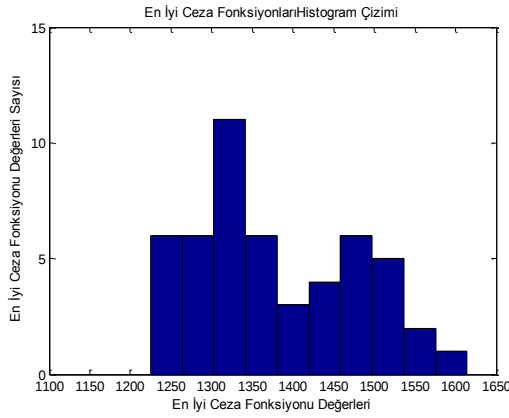
Çizelge 4.20 Çevrimlerde kullanılan frekans listesi

Sınırsız YPSO				Yansımali YPSO			
Her Bir Çevrim İçin Frekans Listesi(MHz)				Her Bir Çevrim İçin Frekans Listesi(MHz)			
231	246	235	241	240	242	234	236
251	259	248	249	247	255	237	243
269	271	265	275	267	269	251	256
279	281	286	290	280	293	273	277
288	291	297	300	295	301	275	279
234	243	230	240	234	240	263	266
250	256	242	245	248	252	270	281
274	276	255	256	254	257	282	287
277	280	266	267	271	289	228	237
281	287	270	271	294	295	255	272
242	246	230	235	232	251	277	279
255	257	240	265	257	258	291	294
261	280	271	274	259	271	296	299
287	291	277	281	276	284	248	259
293	297	289	294	287	292	260	266
226	231	239	253	263	265	268	276
		274	282	286		286	289
						293	

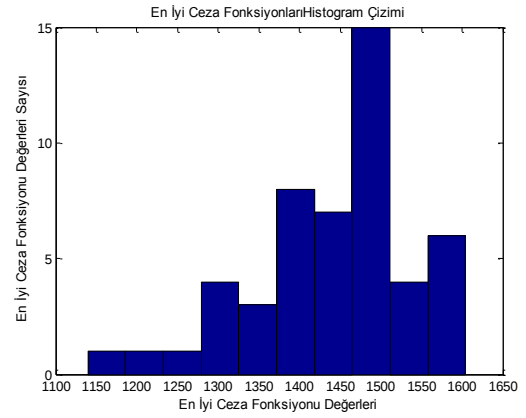
Sınırsız YPSO ile Yansımali YPSO algoritmalarının frekans kümesi içerisinde kullanılabilir 80 tane frekans değerine sahip olduğu senaryodaki performansları şekil 4.27, şekil 4.29 ve şekil 4.31’de verilmiştir. Sürüdeki parçacık sayısı değerleri 50, 100 ve 150 olacak şekilde artırılarak, Sınırsız YPSO ile Yansımali YPSO algoritma performansları incelendiğinde; Sınırsız YPSO için sırasıyla %16, %21 ve %22.1; Yansımali YPSO için %14.3, %15.2, %18.2’lik iyileştirmeler gerçekleşmiştir.

Sınırsız YPSO için sırasıyla 18, 20, 23 ve Yansımali YPSO için 23, 17, 20 tane frekans seti içerisinde kullanılmayan frekans değeri vardır.

Sınırsız YPSO ile Yansımali YPSO algoritmalarının gürbüzlüğünün kıyaslanması için sürüde 50 parçacık ile 50 Monte Carlo sonucunda elde edilen amaç fonksiyonu minimum değerlerinin histogramları çizdirildiğinde ortaya çıkan grafikler aşağıda gösterilmektedir.



a



b

Şekil 4.33.a.Sınırsız YPSO b. Yansımali YPSO için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları

Çizelge 4.21 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler

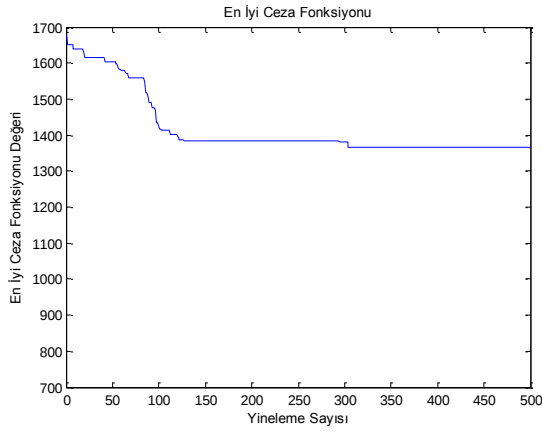
	Sınırsız YPSO	Yansımali YPSO
Ortalama Değer	1387.6	1437.1
Standart Sapma Değeri	100.4	105.3
Minimum Değer	1224.7	1160
Maksimum Değer	1614.4	1580

Çizelge 4.21'deki minimum amaç fonksiyonları ortalama değerlerinden görüldüğü üzere Sınırsız YPSO algoritması Yansımali YPSO algoritmasına göre daha iyi bir performans sergilemektedir. Ayrıca algoritmaların gürbüzlüğünü incelemek için standart sapma değerlerini karşılaştırdığımızda Sınırsız YPSO algoritması Yansımali YPSO algoritmasına karşı yine bir üstünlük sağlamaktadır.

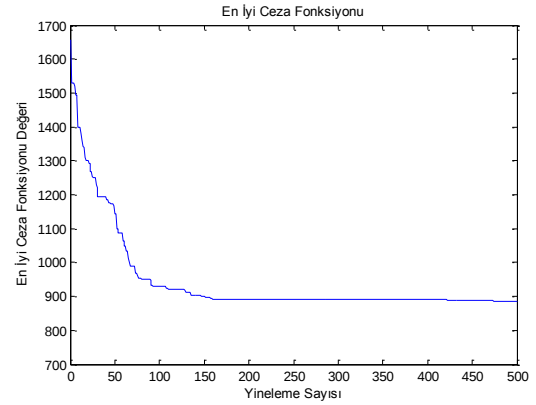
Sıralamalı Sınırsız YPSO algoritmasında parçacıkların yönlendiği “En İyi” değerler konusunda oluşabilecek karışıklığın önüne geçilmesi ve ayrıca problemin çözüm uzayı

içerisinde bulunan gereksiz bölgelerin çıkarılması ile çözümün yakınsayacağı noktaya daha hızlı bir şekilde ulaşması istenmektedir. Sıralamalı Sınırsız YPSO ve Sınırsız YPSO algoritmalarının parçacık sayısı ve Monte Carlo 50 olacak şekilde çalıştırılması sonucunda elde edilen grafikler aşağıda verilmektedir.

Sürüdeki Parçacık Sayısı	50		
Amaç Fonksiyonu Ağırlıkları	Çevrim İçi Ceza	Çevrimler Arası Ceza	Kanal Kazancı Ceza
	0.33	0.33	0.33

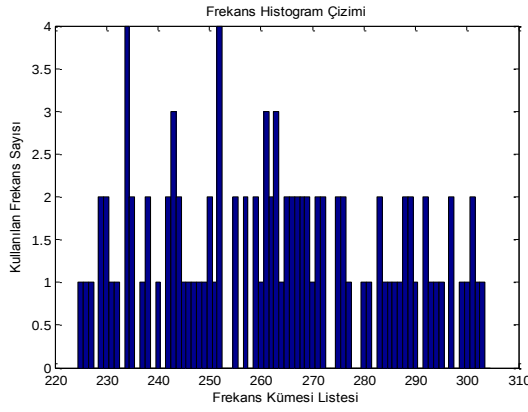


a

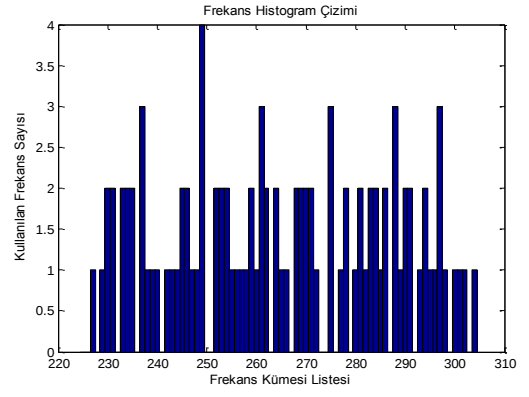


b

Şekil 4.34.a.Sınırsız YPSO: Yineleme sayısı - En iyi amaç fonksiyonu değeri, b.Sıralamalı Sınırsız YPSO Yineleme sayısı - En iyi amaç fonksiyonu değeri grafikleri



a



b

Şekil 4.35.a.Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı, b. Sıralamalı Sınırsız YPSO: Frekans listesi - Atanan frekans kullanım sayısı histogramları

Çizelge 4.22 Kullanılmayan frekans sayısı

Sınırsız YPSO	Sıralamalı Sınırsız YPSO
18	18

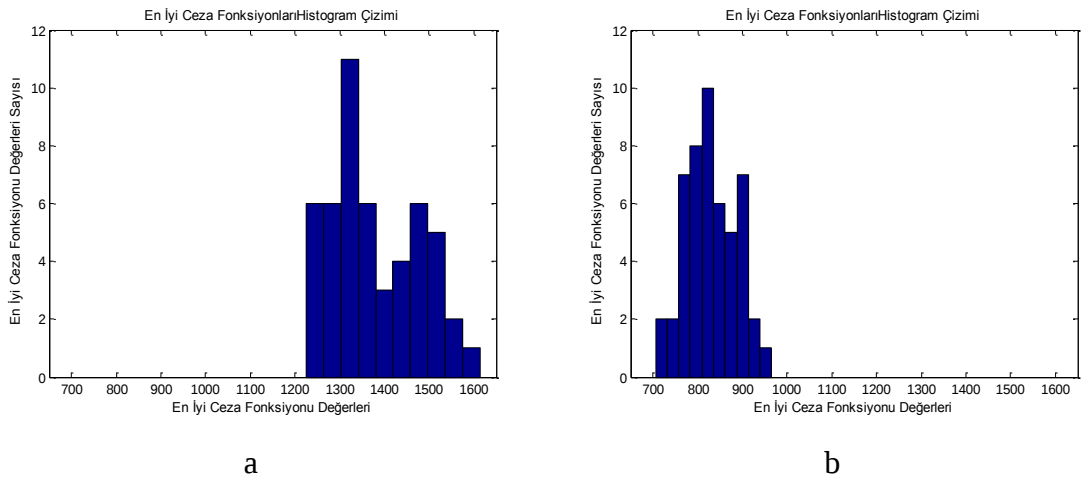
Çizelge 4.23 Çevrimlerde kullanılan frekans listesi

Sınırsız YPSO					Sıralamalı Sınırsız YPSO				
Her Bir Çevrim İçin Frekans Listesi(MHz)					Her Bir Çevrim İçin Frekans Listesi(MHz)				
230	243	234	243	226	231	248	249	231	241
259	267	249	250	238	245	254	257	247	248
270	283	252	260	257	263	268	272	264	265
285	288	263	268	272	288	273	280	266	267
294	301	275	297	292	303	284	287	272	294
235	252	235	240	242	251	232	233	232	234
257	261	242	244	252	263	239	247	244	248
262	263	247	264	265	276	276	277	257	262
264	265	276	277	286	289	284	288	267	286
272	292	289	302	297	300	295	296	287	294
225	230	248	252	227	229	235	239	233	234
232	234	259	261	234	235	241	253	239	250
243	255	266	267	238	246	260	263	251	258
256	269	271	280	250	268	283	301	265	288
295	301	281	284	275	290	302	303	295	296
229	237	244	266	269	271	283	229	242	243
		287	293	299				274	279
								295	290

Sınırsız YPSO ile Sıralamalı Sınırsız YPSO algoritmalarının frekans kümesi içerisinde kullanılabilir 80 tane frekans değerine sahip olduğu senaryodaki performansları Şekil 4.34’de verilmiştir. Sürüdeki parçacık sayısı değeri 50 olacak şekilde, Sınırsız YPSO ile Sıralamalı Sınırsız YPSO algoritma performansları incelendiğinde; Sınırsız YPSO için %16’lık bir iyileşme söz konusu iken; aynı başlangıç değerleriyle Sıralamalı Sınırsız YPSO algoritması için %51’lık bir iyileştirme gerçekleşmiştir.

Sınırsız YPSO için 18 ve Sıralamalı Sınırsız YPSO için 18 tane frekans seti içerisinde kullanılmayan frekans değeri vardır.

Sınırsız YPSO ile Sıralamalı Sınırsız YPSO algoritmalarının gürbüzlüğünün kıyaslanması için sürüde 50 parçacık ile 50 Monte Carlo sonucunda elde edilen amaç fonksiyonu minimum değerlerinin histogramları çizdirildiğinde ortaya çıkan grafikler aşağıda gösterilmektedir.



Şekil 4.36.a.Sınırsız YPSO, b. Sıralamalı Sınırsız YPSO için 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri histogramları

Çizelge 4.24 50 Monte Carlo sonucunda bulunan minimum amaç fonksiyonu değerleri için istatistiksel ölçüler

	Sınırsız YPSO	Sıralamalı Sınırsız YPSO
Ortalama Değer	1387.6	830.2
Standart Sapma Değeri	100.4	57
Minimum Değer	1224.7	705.5
Maksimum Değer	1614.4	965.2

Çizelge 4.24'deki minimum amaç fonksiyonları ortalama değerlerinden görüldüğü üzere Sıralamalı Sınırsız YPSO algoritması Sınırsız YPSO algoritmasına göre çok daha iyi bir performans sergilemektedir. Ayrıca algoritmaların gürbüzlüğünü incelemek için standart sapma değerlerini kıyasladığımızda Sıralamalı Sınırsız YPSO algoritması Sınırsız YPSO algoritmasına karşı yine bir üstünlük göstermektedir.

4.2.3 Frekans Atama Problemi'ne skalarizasyon yaklaşımları

- **Ağırlıklandırılmış toplam temelli skalarizasyon yaklaşımı**

Yukarıda anlatılan amaç fonksiyonları ve kısıtlar ışığında Frekans Atama Problemi'nin Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı ile modellenmesi gerçekleştirilecektir. Bölüm 3.3.2.1'de anlatıldığı üzere Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı birden fazla amaç fonksiyonunun lineer kombinasyon şeklinde ağırlıklandırılması sonucu çok amaçlı problemi tek amaçlı bir optimizasyon problemine dönüştürmektedir. Bu temelde Frekans Atama Problemi'ni matematiksel olarak eşitlik (4.9)'deki gibi ifade edebiliriz.

$$A = w_{intra} \sum_{i=1}^{10} P^{(i)} + w_{inter} \sum_{i=1}^{10} \sum_{j=i+1}^{10} R^{(i,j)} + w_{channel} \sum_{i=1}^{10} K^{(i)} \quad (4.9)$$

A fonksiyonuna, her bir çevrim için hesaplanmış çevrim içi ceza fonksiyonlarının toplamı; farklı bütün çevrim ikilileri arası hesaplanmış çevrimler arası ceza fonksiyonlarının toplamı ve her bir çevrim için hesaplanmış kanal kazancı ceza fonksiyonlarının toplamı, belirlenecek olan w_{intra} , w_{inter} ve $w_{channel}$ oranında katkı

sağlayacaktır. Frekans planlaması, eldeki girdiler ve kısıtlar ile A fonksiyonunun minimize edilmesi amaçlanarak gerçekleştirilecektir.

- **Ağırlıklandırılmış Chebychev skalarizasyon yaklaşımı**

Frekans planlaması için kullanılan amaç fonksiyonları ve tanımlanan kısıt göz önüne alınarak; Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı ile problemin modellemesi yapılacaktır. Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı doğrusal olmayan bir skalarizasyon yaklaşımı olup; yine çok amaçlı optimizasyon probleminin tek amaçlı olacak şekilde tanımlanmasını sağlamaktadır. Bu yaklaşımla Frekans Atama Problemi'ni matematiksel olarak,

$$\begin{aligned} A_{intra} &= \sum_{i=1}^{10} P^{(i)} \\ A_{inter} &= \sum_{i=1}^{10} \sum_{j=a+1}^{10} R^{(i,j)} \\ A_{channel} &= \sum_{i=1}^{10} K^{(i)} \end{aligned} \quad (4.10)$$

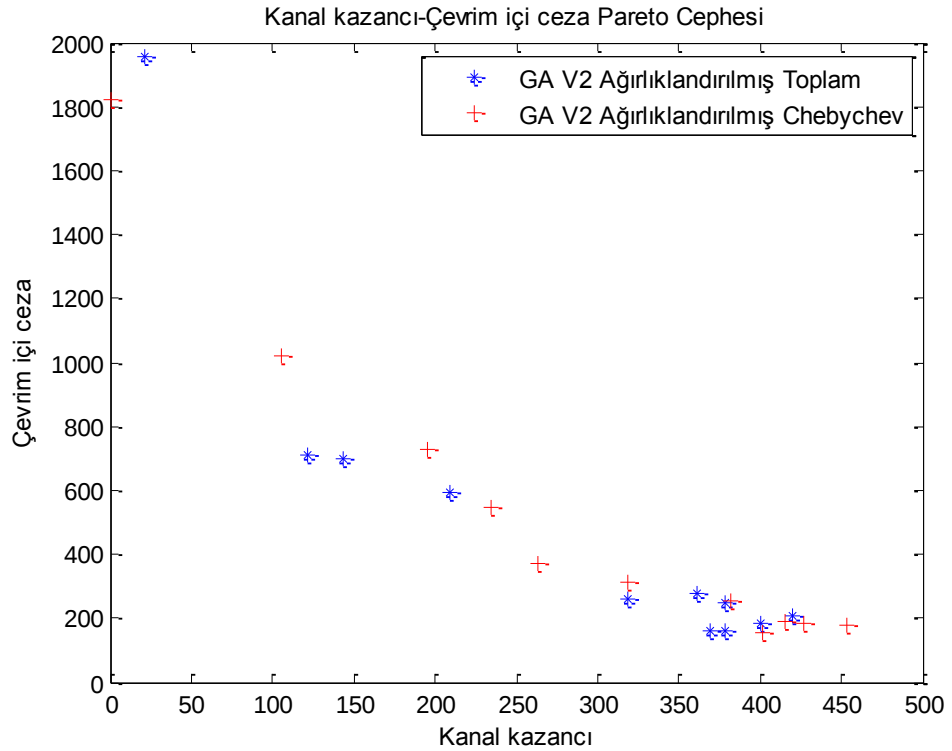
Şeklinde ifade edilmektedir. A_{intra} fonksiyonu, her bir çevrim için hesaplanmış çevrim içi ceza fonksiyonlarının toplamı; A_{inter} fonksiyonu farklı bütün çevrim ikilileri arası hesaplanmış çevrimler arası ceza fonksiyonlarının toplamı ve $A_{channel}$ fonksiyonu her bir çevrim için hesaplanmış kanal kazancı ceza fonksiyonlarının toplamını vermektedir. Buradan frekans atama işlemi için tek amaca indirgenen ve minimize edilmesi istenen A fonksiyonu eşitlik (4.11)'de sunulmaktadır.

$$A = \max \left[w_{intra} (A_{intra} - z_1^{**}), w_{inter} (A_{inter} - z_2^{**}), w_{channel} (A_{channel} - z_3^{**}) \right] \quad (4.11)$$

z_1^{**} , z_2^{**} ve z_3^{**} değerleri tanımlanan problem için ütöpik noktalar. Bu değerler amaç uzayının negatif bölgede olamayacağı bilindiğinden dolayı bu problem için 0 olarak seçilmiştir.

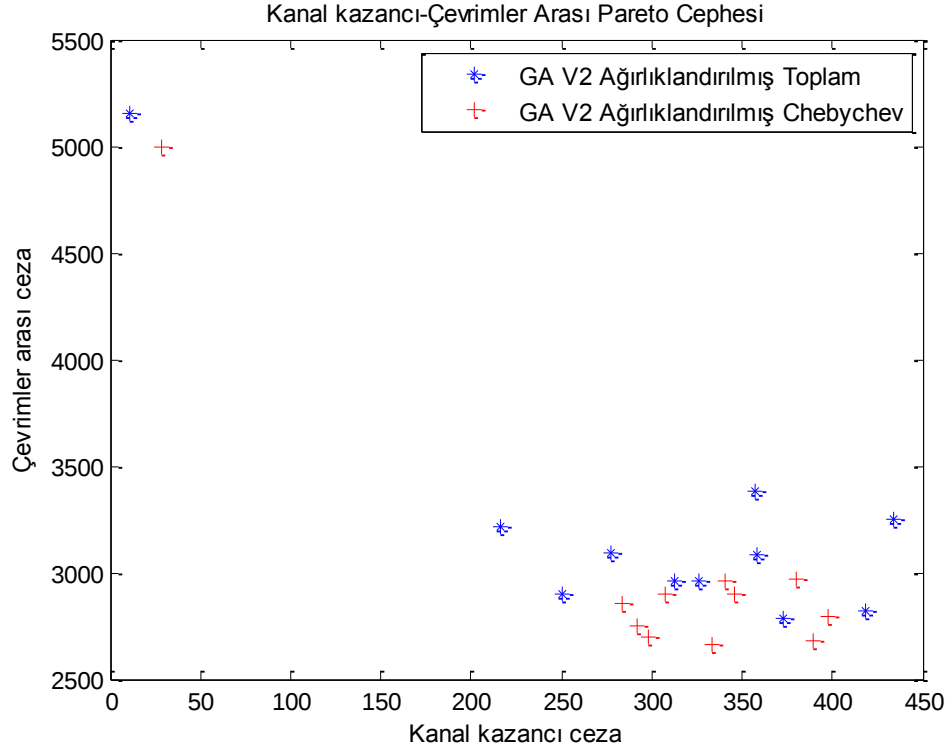
Üzerinde çalışılan Frekans Atama Problemi'nde farklı skalarizasyon yaklaşımları kullanarak elde edilen Pareto cephelerini inceleyelim. Ağırlıklandırılmış Toplam Temelli Skalarizasyon ve Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımları etkileri gözlemlenmesi için GA V2 algoritması seçilmiştir. Etkilerin görsel olarak daha iyi anlatılması adına amaç fonksiyonlarının ikili gruplar halinde, yani kanal kazancı – çevrim içi ceza, kanal kazancı – çevrimler arası ceza ve çevrim içi – çevrimler arası ceza şeklinde alınarak Pareto cepheleri çizdirilmiştir.

Amaç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	0'dan 1'e kadar 0.1 aralıklarla artıyor.
Çevrimler arası ceza	0
Kanal kazancı ceza	1'den 0'a kadar 0.1 aralıklarla azalıyor.



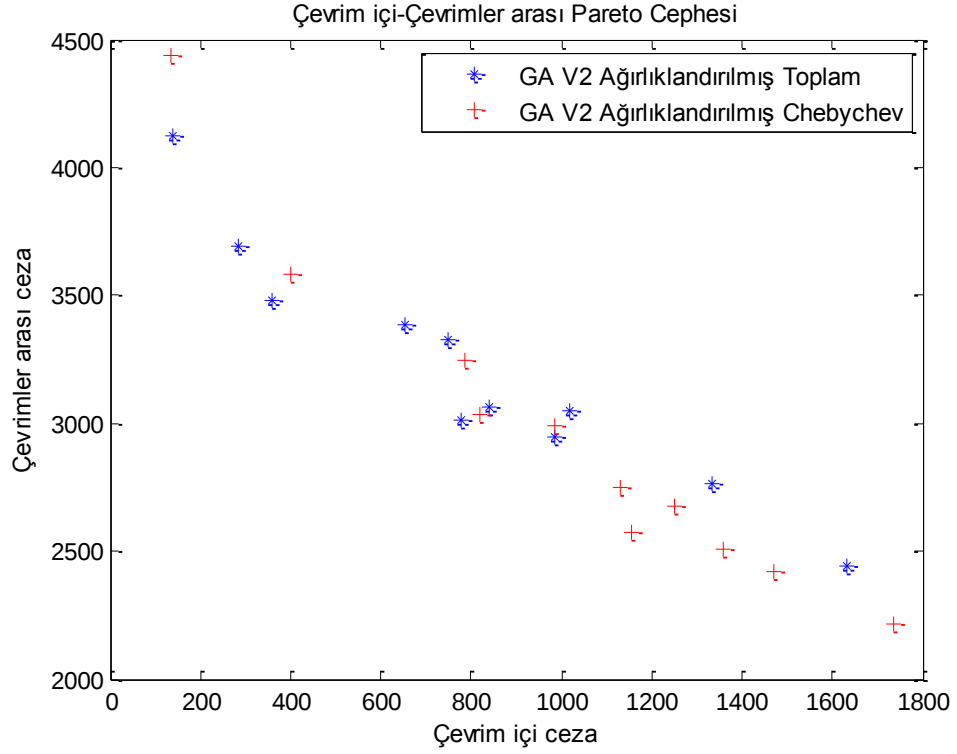
Şekil 4.37 Kanal kazancı cezası ve çevrim içi ceza amaçları için oluşan Pareto cephesi

Amaç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	0
Çevrimler arası ceza	0'dan 1'e kadar 0.1 aralıkla artıyor.
Kanal kazancı ceza	1'den 0'a kadar 0.1 aralıkla azalıyor.



Şekil 4.38 Kanal kazancı cezası ve çevrimler arası ceza amaçları için oluşan Pareto cephesi

Amaç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	1'den 0'a kadar 0.1 aralıkla azalıyor.
Çevrimler arası ceza	0'dan 1'e kadar 0.1 aralıkla artıyor.
Kanal kazancı ceza	0



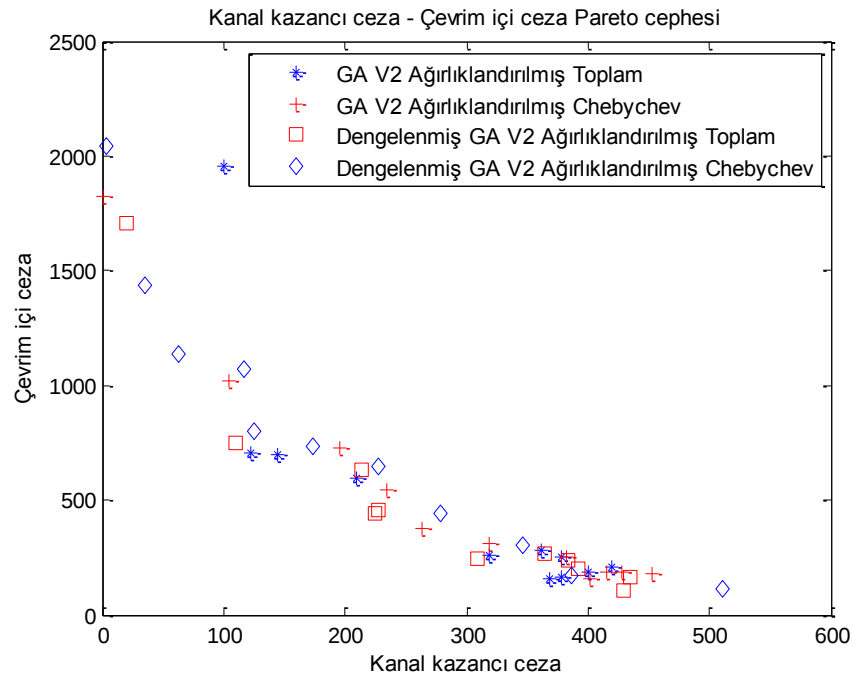
Şekil 4.39 Çevrim içi ceza ve çevrimler arası ceza amaçları için oluşan Pareto cephesi

Özellikle Şekil 4.37 - 4.38 incelendiğinde Ağırlıklandırılmış Chebychev Skalarizasyon yaklaşımı için beklenen düzgün dağılım gösteren Pareto cephelerine ulaşamadığı gözlemlenmektedir. Düzgün gerçekleşmeyen bu dağılımın sebebinin, amaç fonksiyonları değerlerinin birbirlerine göre büyüklükleri arasında çok yüksek farkların olmasından kaynaklandığı düşünülmüştür. Örneğin, çevrimler arası ceza fonksiyonu 3500 civarı değerler alırken, kanal kazancı ceza fonksiyonu ise 250 civarı değer almaktadır. Bu durumun fiziksel olarak, Pareto cephesi saptamakta kullanılan belirli eğimlere sahip ışınların düzgün bir dağılımla cepheyi kesememesine yol açtığı görülmüştür.

Ortaya çıkan amaç fonksiyonlar arasındaki farkların giderilmesi için amaç fonksiyonları değerlerini dengeleyecek katsayılar hesaplanmıştır. Bu katsayılar kullanılarak elde edilen Pareto cephesi grafikleri sunulmaktadır.

Amaç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	0'dan 1'e kadar 0.1 aralıkla artıyor.
Çevrimler arası ceza	0
Kanal kazancı ceza	1'den 0'a kadar 0.1 aralıkla azalıyor.

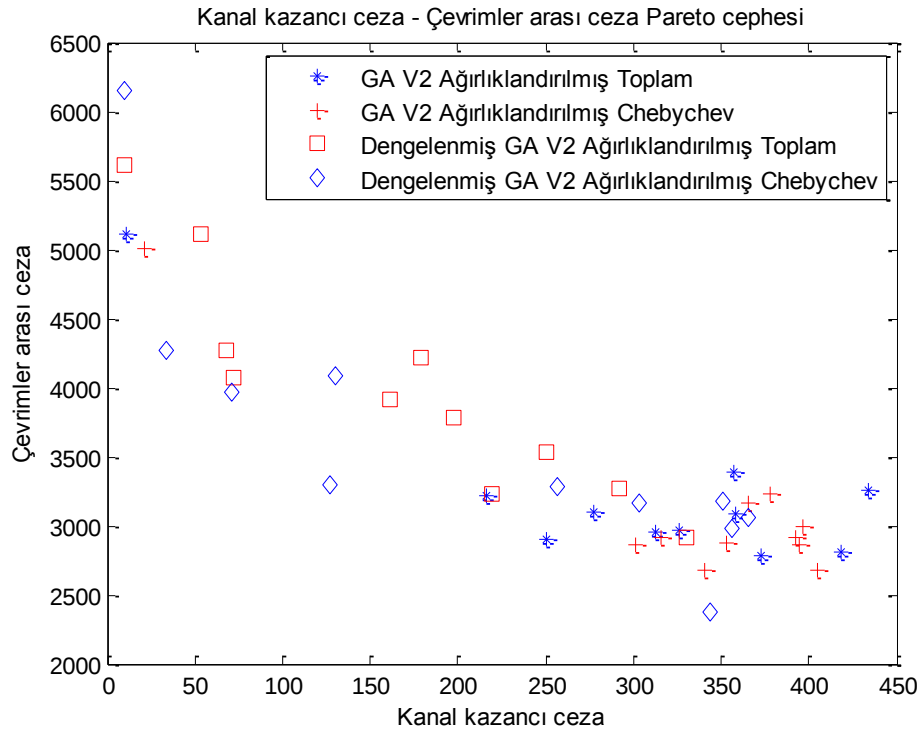
Amaç Fonksiyonları	Katsayı Değeri
Kanal kazancı ceza - Çevrim içi ceza değerleri dengelemesi	0.321



Şekil 4.40 Normal ve dengelenmiş kanal kazancı cezası ve çevrim içi ceza amaçları için oluşan Pareto cepheleri

maç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	0
Çevrimler arası ceza	0'dan 1'e kadar 0.1 aralıkla artıyor.
Kanal kazancı ceza	1'den 0'a kadar 0.1 aralıkla azalıyor.

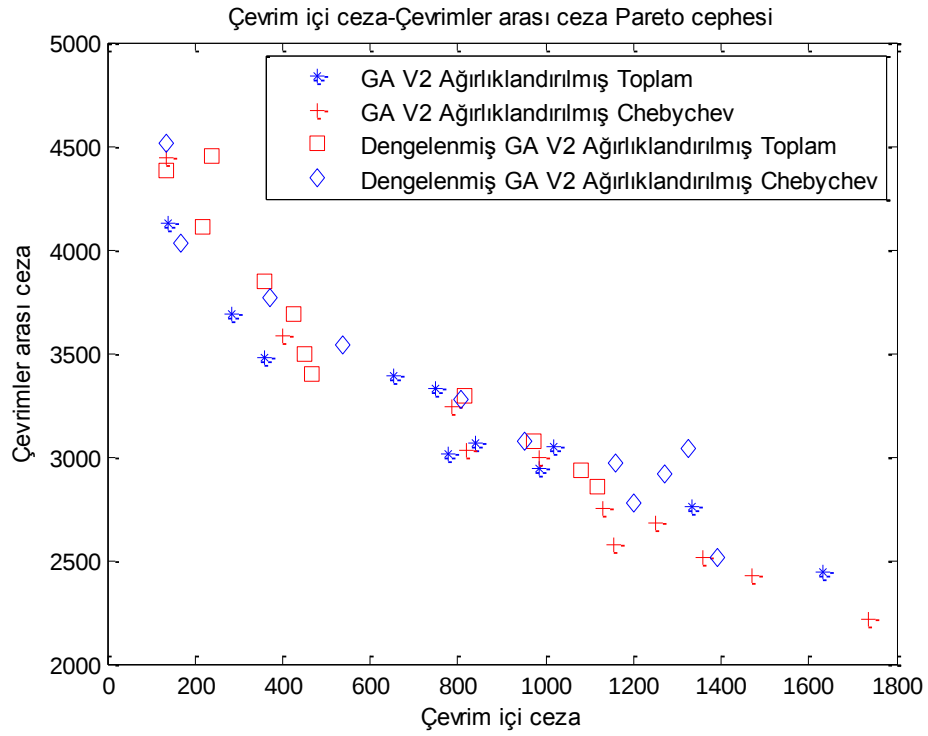
Amaç Fonksiyonları	Katsayı Değeri
Kanal kazancı ceza - Çevrimler arası ceza değerleri dengelemesi	0.077



Şekil 4.41 Normal ve dengelenmiş kanal kazancı cezası ve çevrimler arası ceza amaçları için oluşan Pareto cepheleri

Amaç Fonksiyonları	Ağırlık Değerleri
Çevrim içi ceza	1'den 0'a kadar 0.1 aralıkla azalıyor.
Çevrimler arası ceza	0'dan 1'e kadar 0.1 aralıkla artıyor.
Kanal kazancı ceza	0

Amaç Fonksiyonları	Katsayı Değeri
Çevrim içi ceza – Çevrimler arası ceza değerleri dengelemesi	0.2403



Şekil 4.42 Normal ve dengelenmiş çevrim içi ceza ve çevrimler arası ceza amaçları için oluşan Pareto cepheleri

Özellikle birbirleri ile çelişen ceza fonksiyonları arasında büyüklük farklarının çok yüksek olduğu, şekil 4.40 - 4.41'de verilen Pareto cepheleri incelendiğinde, dengeleme işleminin skalarizasyon yaklaşımları sonucu elde edilen Pareto cephelerinin düzgün dağılım sağlayabilmesi konusunda gösterdiği iyileştirme çok açık bir şekilde anlaşılmaktadır.

5. TARTIŞMA ve SONUÇ

Bu tezde üzerinde çalışılan Frekans Atama Problemi, her bir çevrim içindeki frekans değerlerinin birbirleriyle yaptığı girişimin minimize edilmesi, komşu çevrimlerdeki frekans değerlerinin birbirleriyle yaptığı girişimlerin minimize edilmesi ve her bir çevrimde kullanılan frekansların kanal kazanç değerlerinin birbirlerine yakın seçilmesi gibi birbirleri ile çelişen amaçları içeren çok amaçlı optimizasyon problemidir. Literatür araştırmalarında görüldüğü üzere, bu problemle karşılaştırılabilecek birebir bir çalışma bulunmamakla beraber, Frekans Atama Problemi Bölüm 2’de verilen Minimum Girişimli Frekans Atama ve Minimum Genişlikli Frekans Atama problemlerinin füzyonu şeklinde olan bir yapıdadır.

Tezde gerçekleştirilen çalışmada, tanımı verilen Frekans Atama Problemi için problem uzayının çeşitli yerlerinde bir takım sanal temsilcilerin oluşturulması, bu temsilcilerin işaret ettiği noktalarda amaç fonksiyonlarının hesaplanması, ardından ilgili temsilcilerin güncellenerek bu işlemin yinelemeli bir döngü içerisinde gerçekleştirilmesi şeklinde sistematik deneme-yanılma esasına dayanan metasezgisel yöntemlere başvurulmuş ve bu yöntemlerin performansı, amaç fonksiyonu değeri azaltma yüzdeleri temel alınarak kıyaslanmıştır. Problemin çözümünde, modifiye edilerek üretilen GA ve PSO algoritmalarının çeşitli versiyonları kullanılmıştır. Özellikle PSO yapısından yararlanılarak oluşturulan YPSO algoritması, literatürdeki frekans atama problemi çözümü için gerçekleştirilen çalışmalara yeni bir katkı sağlamaktadır.

Verilen senaryolar ve tanımlanan problem için algoritmaların başarımlarına bakıldığında GA ve PSO yapılarını temel alan bir üstünlük karşılaştırılması gerçekleştirilememektedir. Çünkü GA temelli GA V2 ve PSO temelli Sınırsız YPSO algoritma versiyonları, GA temelli GA V1 ve PSO temelli Yansımali YPSO’ya göre problemin çözümünde daha iyi performans göstermektedir. Buradan da anlaşılacağı üzere yöntemlerin yapılarından çok, üzerinde gerçekleştirilen modifikasyonlar algoritmaların çözüm performansına etki etmektedir. Böylece, yapısı gereği sürekli problemlerin çözümü için daha kullanışlı olan PSO algoritmasının kesikli bir problem olan frekans atama problemi için kullanılmasından kaynaklı oluşabilecek dezavantajlar,

PSO algoritması üzerinde yapılan değişikliklerle ortadan kaldırılabildiği anlaşılmaktadır. Özellikle Sınırsız YPSO algoritması üzerinde gerçekleştirilen değişikliklerle ortaya çıkarılan Sıralamalı Sınırsız YPSO algoritması, parçacıkları doğru noktalara yönlendirerek çözüm performansı üzerinde çok büyük bir pozitif etki yarattığı ve amaç uzayındaki çözüm için kullanışsız bölgelerin atılmasıyla bu yüksek iyileştirme performansına çok kısa yineleme adımında erişilmesini sağladığı sonucuna varılmıştır.

Problemin çözümünde kullanılan sanal temsilcilerin artırılması, başlangıçta verilen alternatif çözümlerin sayısını artırırken daha önce çözüm aranmamış farklı noktalarda da çözüm arayışlarının gerçekleştirmesini sağlamaktadır. Böylece, sanal temsilci sayısını artışı daha uygun çözümlere ulaşma konusunda katkı vermektedir. Ancak parçacık ve kromozom sayısındaki büyüme hesaplama zamanı maliyeti artışı da beraberinde getirmesinden dolayı temsilci sayısı, performansa vereceği katkı ile hesaplama zamanı maliyeti konusundaki denge göz önünde bulundurularak seçilmelidir.

Problemde atama işlemi gerçekleştirildikten sonra frekans kümesi içerisinde atanamayan kalan frekans sayıları incelendiğinde, bu sayı değerlerinin algoritmaların optimizasyon performansı ile orantılı olarak değişmediği, bu problem için algoritmaların başarımlarına direk olarak etki etmediği anlaşılmıştır.

Üzerinde çalışılan Frekans Atama Problemi tek amaçlı olacak şekilde ele alındığında, sadece kanal kazancı ceza fonksiyonunun minimize edilmesi istenirse, her bir algoritmanın mükemmel optimum çözümlere ulaştığı; sadece çevrim içi ceza fonksiyonunun minimize edilmesi arzulandığında GA V1 ve GA V2 algoritmalarının neredeyse mükemmel uygun çözümler elde ederek diğer iki algoritmayı geride bıraktığı ve sadece çevrimler arası ceza fonksiyonunun minimize edilmesi durumunda tüm algoritmaların, diğer ceza fonksiyonlarını indirmedeki başarımlarına kıyasla, çok yüksek performans gösteremediği görülmüştür.

Frekans Atama Problemi'nin çözümünde kullanılan algoritmaların gürbüzlüğü konusunda bir karşılaştırma gerçekleştirildiğinde GA V1 ve GA V2 algoritmalarının Sınırsız YPSO ve Yansımali YPSO algoritmalarına göre çok daha sağlam sonuçlar

ürettiği çok net bir şekilde görülmektedir. PSO tabanlı üretilen algoritmaların gürbüzlük konusunda yaşadığı dezavantaj Sıralamalı Sınırsız YPSO algoritması ile giderilse de bu algoritmanın da GA tabanlı üretilen algoritmaların gerisinde kaldığı gözlemlenmiştir.

Tezde verilen sonuçlardan biri de farklı skalarizasyon yöntemlerinin problemin amaç uzayında oluşturdukları Pareto cepheleridir. Oluşan cephelerin kıyaslanması, GA V2 yöntemi yardımıyla, birden fazla amacın ağırlıklandırılarak amaç fonksiyonların doğrusal birleştirilmesi şeklinde yazıldığı Ağırlıklandırılmış Toplam Temelli Skalarizasyon Yaklaşımı ve ilgili mesafe normunun L_{inf} normu olarak seçildiği duruma karşılık gelen Ağırlıklandırılmış Chebychev Skalarizasyon Yaklaşımı arasında gerçekleştirilmiştir. Skalarizasyon yaklaşımlarının düzgün yayımlı bir Pareto cephesi oluşturması beklenirken; tanımlanan Frekans Atama Problemi için bu sonucun elde edilemediği görülmüştür. Ancak bu durumun elde edilememesinde ceza fonksiyon değerleri arasındaki büyük farkların oluşması olduğu tespit edilmiş ve ceza fonksiyonları değerleri arasında dengeleme gerçekleştirdikten sonra cephe üzerindeki bulunan noktaların normal aralığa eşleştirilmesiyle daha düzgün dağılımlı cepheler elde edilebileceği sonuçlarına ulaşılmış ve bu açıdan literatüre katkı sağlanmıştır.

Gelecek çalışmalarda, dinamik olarak değişebilen çevrim topolojisi ve dolayısıyla dinamik olarak değişen kısıt ve amaç fonksiyonlarının olduğu durumlarda üretilen algoritmalarının başarımlarının incelenmesi ve bu duruma uygun olarak algoritma modifikasyonları gerçekleştirilebilir. Merkezi olarak çözülen Frekans Atama Problemi'ni dağıtık olarak çözebilen algoritmaların geliştirilmesi açısından ilerleme kaydedilebilir. Ayrıca doğrudan çok amaçlı optimizasyon yapan Srivinas ve Deb (1994) tarafından önerilen Bastırılmamış Sınıflandırılmalı Genetik Algoritma ve Schaffer (1985) tarafından ilk çalışmaları gerçekleştirilen Vektör Değerlendirmeli Genetik Algoritma vb. yöntemlerin Frekans Atama Problemi çözümündeki performansları değerlendirilebilir.

KAYNAKLAR

- Altıparmak, F. 1996. Genetik Algoritma ile Haberleşme Şebekelerinin Topolojik Optimizasyonu. Doktora Tezi. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara
- Altınöz, Ö. T. and Yılmaz, A. E. 2009. Parçacık sürüsü Optimizasyonunda Yeni Bir Birey Davranış Biçimi Önerisi, EMO 13. Ulusal Kongre Bildiriler Kitabı, 23-26 Aralık 2009, Ankara
- Altınöz, Ö. T. ve Yılmaz, A. E. 2011. Particle Swarm Optimization with Parameter Dependency Walls and its Sample Application to the Microstrip-Like Interconnect Line Design, International Journal of Electronics and Communications, 66(2); 107-114
- Beckmann, D. and Killat, U. 1999. A new strategy for the application of genetic algorithms to the channel assignment problem, IEEE Transactions on Vehicular Technology, 48(4); 1261–1269
- Booker, L. 1987. Improving search in genetic algorithms, Genetic Algorithms and Simulating Annealing Edition, 5(2); 61-73
- Castelino, D. J., Hurley, S. and Stephens, N. M. 1996. A tabu search algorithm for frequency assignment, Annals of Operations Research, 63(1); 301–319
- Cattrysse, D.G., Salomon, M. and Van Wassenhove, L. 1994. A set partitioning heuristic for the generalized assignment problem, European Journal of Operational Research, 72(1); 167-174
- Clerc, M. and Kennedy, J. 2002. The particle swarm: explosion, stability and convergence in a multi-dimensional complex space, IEEE Transactions on Evolutionary Computation, 6(1); s. 58–73.
- Coley, D.A. 1991. An introduction to genetic algorithms for scientists and engineers, World Scientific, 244, New Jersey
- Costa, D. 1993. On the use of some known methods for t-colourings of graphs, Annals of Operations Research, 41(1); 343–358.
- Crisan, C. and Mühlenbein, H. 1998. The breeder genetic algorithm for frequency assignment, Lecture Notes in Computer Science, 1498(1); 897–906.
- Crompton, W., Hurley, S. and Stephens, N. M. 1994. A parallel genetic algorithm for frequency assignment problems, Proceedings IMACS/IEEE International Symposium on Signal Processing, Robotics and Neural Networks, 25-27 Nisan 1994, 81–84, Lille

- Cuppini, M. 1994. A genetic algorithm for channel assignment problems, *European Transactions on Telecommunications and Related Technologies*, 5(1); 285–294
- Deb, K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization, 954, Chichester.
- Dorne, R. and Hao, J.-K. 1995. An evolutionary approach for frequency assignment in cellular radio networks, *IEEE International Conference on Evolutionary Computing*, 2(1); 539-544
- Dreou, J. and Candan, C. 2011. Web Sitesi: <http://www>. <http://en.wikipedia.org>, Erişim Tarihi: 5.12.2014.
- Dutta, J. ve Kaya, C. Y. 2011. A new scalarization and numerical method for constructing weak Pareto front of multi-objective optimization problems *Optimization*, 60(8-9); 1091-1104
- Ehrgott, M. 2006. A discussion of scalarization techniques for multiple objective integer programming, *Ann Oper Res*, 147(1); 343–360
- Eichfelder, G. 2008. *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer-Verlag, 244, Berlin
- Eker, İ. 2005. Particle swarm optimization and differential evolution algorithms for continuous function optimization problems. Yüksek Lisans Tezi. İstanbul Kültür Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul
- Ergül, E.U. 2010. Çok amaçlı genetik algoritmalar: temelleri ve uygulamaları. Doktora Tezi. Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun
- Fisher, M.L. and Jaikumar, J. 1981. A generalized assignment heuristic for vehicle routing, *Wiley Periodicals Inc.* 11(2); 109–124
- Fisher, M.L., Jaikumar, J. and Van Wassenhove, L.N. 1986. A multiplier adjustment method for the generalized assignment problem, *Institute for Operations Research and the Management Sciences*, 32(9); 1095 - 1103
- Grefenstette, J.J. 1986. Optimization of control parameters for genetic algorithm, *IEEE Trans Systems, Man and Cybernetics*, 16(1); 122-128
- Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 432, New York
- Goldberg, D.E. and Deb, K. 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, *Foundations of Genetic Algorithms*, Morgan Kaufmann, 69-93, California

- Hao, J.-K. and Perrier, L. 1999. Tabu search for the frequency assignment problem in cellular radio networks, Teknik Rapor, LGI2P EMA-EERIE, Parc Scientifique Georges Besse, Nimes
- Hale, W.K. 1980. Frequency Assignment: Theory and Applications, Proceedings of IEEE International Conference, 68(12); 1497–1514.
- Holland, J. 1992. Adaptation in Natural and Artificial Systems, Cambridge, MA: MIT Press, 211, Cambridge
- Jaimes-Romero, F.J., Munoz-Rodriguez, D. and Tekinay, S. 1996. Channel assignment in cellular systems using genetic algorithms, Proceedings of the 46th IEEE Vehicular Technology Conference, 28 Nisan-1 Mayıs 1996, Atlanta
- Kapsalis, A., Chardaire, P., Rayward-Smith, V.J. and Smith, G.D. 1995. The Radio Link Frequency Assignment Problem: A Case Study Using Genetic Algorithms, Bilgisayar Bilimi Ders Notları, 117-131
- Kaya, C. Y. and Maurer, H. 2013. A numerical method for nonconvex multi-objective optimal control problems Computational Optimization and Applications, 57(3); 685-702
- Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks IV, 4(1), 1942–1948
- Klamroth, K. and Tind, J. 2006. Constrained optimization using multiple objective programming, J Global Optimization, 37(1); 325–355
- Koza, J. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection, MA: MIT Press, 840, Cambridge
- Kurtaran, H. 2005. Optimizasyon Teorisi ve Uygulamaları Ders Notları, Gebze Yüksek Teknoloji Enstitüsü Tasarım ve İmalat Mühendisliği Bölümü, Gebze
- Lai, W.K. and Coghill, G.G. 1996. Channel assignment through evolutionary optimization, IEEE Transactions on Vehicular Technology, 45(1); 91–95
- Lin, J. G. 2003. On min-norm and min-max methods of multi-objective optimization, Springer-Verlag Mathematic Programming, 103(1); 1–33
- Maniezzo, V. and Carbonaro, A. 2000. An ants heuristic for the frequency assignment problem, Future Generation Computer Systems, 16(1); 927–935
- Marler, R.T. and Arora, J. S. 2004. Survey of multi-objective optimization methods for engineering, Struct Multidisc Optim 26(1); 369–395

- Mazzola, J.B. 1989. Generalized assignment with nonlinear capacity interaction, Institute for Operations Research and the Management Sciences, 35(8); 923-941
- Miettinen, K.M. 1999. Nonlinear Multiobjective Optimization, Kluwer Academic Publisher, 296, Norwell
- Miettinen, K.M. and Makela, M. 2002. On scalarizing functions in multiobjective optimization, OR Spectrum, 24(1); 193–213
- Mishra, A., Banerjee, S. and Arbaugh, W. 2005. Weighted coloring based channel assignment for WLANs, ACM SIGMOBILE Mobile Computing and Communications Review, 9(3); 19–31
- Montemanni, R., Smith, D. H. and Allen, S. M. 2002. An ANTS algorithm for the minimum-span frequency-assignment problem with multiple interference, IEEE Transactions on Vehicular Technology, 15(5); 949–953.
- Nauss, R.M. 2003. Solving the generalized assignment problem: an optimizing and heuristic approach, INFORMS Journal on Computing, 15(3); 249-266
- Ngatchou, P., Zarei, A. and El-Sharkawi, M. A. 2005. Pareto Multi Objective Optimization, Intelligent Systems Application to Power Systems,. Proceedings of the 13th International Conference, 6-10 Kasım 2005, 84-91, Arlington
- Ngo, C. Y. and Li, V. O. K. 1998. Fixed channel assignment in cellular radio networks using a modified genetic algorithm, IEEE Transactions on Vehicular Technology, 47(1); 163–171
- Önder, E. 2011. Araç Rotalama Problemlerinin Parçacık sürüsü ve Genetik Algoritma ile Optimizasyonu. Doktora Tezi. İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul
- Park, T. and Lee, C.Y. 1996. Application of the graph coloring algorithm to the frequency assignment problem, Journal of the Operations Research Society of Japan, 39(1); 258–265
- Rao, S.S. 2009. Engineering Optimization: Theory and Practice, John Wiley & Sons, Inc., Hoboken, 828, New Jersey
- Savelsbergh, M.W.P. 1997. A branch-and-price algorithm for the generalized assignment problem, Annals of Operations Research, 45(6); 831–841
- Schafer, J.D. 1985. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Proceedings of the 1st International Conference on Genetic Algorithms, 93-100, Hillsdale, New Jersey

- Shi, Y. and Eberhart, R.C. 1998. A modified particle swarm optimizer, Proceedings of IEEE International Conference on Evolutionary Computation, 4-9 Mayıs 1998, 69–73, Anchorage
- Shi, Y. and Eberhart, R. C. 1999. Empirical study of Particle Swarm Optimization, Proceedings of congress on evolutionary computation, 6-9 Temmuz 1999, 1945–1950, Washington
- Srinivas, M. and Patnaik, L. M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, IEEE Transactions on Systems, Man and Cybernetics, 24(4); 656- 667
- Srinivas, N. and Deb, K. 1994. Multi-objective Function Optimization Using Non-dominated Sorting Genetic Algorithms. Evolutionary Computation Journal, 2(3), 221-248.
- Şatır, M. 2008. Kuadratik atama problemlerinin çözümünde ayrık birey-koloni optimizasyon modeli. Doktora Tezi. İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul
- Talbi, E.G. 2009. Metaheuristic: from Design to Implementation, Wiley, 624, Lille
- Tsang, E. and Voudouris, C. 1998. Solving the radio link frequency assignment problem using guided local search, NATO Symposium on Radio Length Frequency Assignment, Ekim 1998, 14(b), Aalborg
- Valenzuela, C.L.A. and Hurley, S. 1998. Breeding Permutations for Minimum Span Frequency Assignment, Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference, 308-311, Norwich
- Van Moffaert, K., Drugan, M.M., and Nowe, A. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques, Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), IEEE Symposium, 16-19 Nisan 2013, 191-199, Singapore
- Weerasena, L. 2014. Approximation in Multiobjective Optimization with Applications. Doktora Tezi. Clemson Üniversitesi, Matematik Bilimleri, South Carolina
- Yagiura, M. and Ibaraki, T. 1989. The Generalized Assignment Problem and Its Generalizations, <http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.417.8967>, Colorado
- Zitzler, E. 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Doktora Tezi. İsviçre Federal Zürih Teknik Bilimler Enstitüsü, Zürih

ÖZGEÇMİŞ

Adı Soyadı : Barış SATAR

Doğum Yeri : Anamur

Doğum Tarihi : 26.09.1987

Medeni Hali : Bekâr

Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Anamur Anadolu Lisesi (2005)

Lisans : Ankara Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği
Bölümü (2011)

Yüksek Lisans: Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektirik-Elektronik
Mühendisliği Anabilim Dalı (Şubat 2015)

Çalıştığı Kurum

Ankara Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü
Araştırma Görevlisi (2013-...)