

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜRKİYE'DEKİ BÖLGESEL GÜÇ SİSTEMLERİNİN EKONOMİK YÜK
DAĞITIMININ FARKLI ALGORİTMALARLA OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

Enzel AYDIN

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi A. Erhan AKKAYA

NİSAN 2025

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**TÜRKİYE’DEKİ BÖLGESEL GÜÇ SİSTEMLERİNİN EKONOMİK YÜK
DAĞITIMININ FARKLI ALGORİTMALARLA OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

Enzel AYDIN
(36213619009)

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi A. Erhan AKKAYA

NİSAN 2025

İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

**Türkiye’deki Bölgesel Güç Sistemlerinin Ekonomik Yük Dağıtımının Farklı
Algoritmalarla Optimizasyonu**

Tezi Hazırlayan: Enzel AYDIN
Yüksek Lisans Tezi

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının aşamalarında, öneri, bilgi, tecrübe ve desteklerini esirgemedi beni her konuda destekleyen danışman hocam Sayın Dr. Öğretim Üyesi A. Erhan AKKAYA'ya en içten teşekkürlerimi sunarım.

Çalışmalarında ayrıca akademik yolda benden bilgi ve tecrübesini esirgemeyen kuzenim Doç. Dr. Ömer ORUÇ'a çok teşekkür ederim.

Üzerimde büyük emekleri olan, deprem ve pandemi zamanlarında desteğini ve akademik alandaki bilgilerini benden eksik etmeyen Prof. Dr. Celaleddin YEROĞLU'na çok teşekkür ederim.

Bu süreçte beni yalnız bırakmayan ailem ve özellikle yönlendirmeleri ile yoluma devam etmemi sağlayan abim Batman Üniversitesi Yapı İşleri ve Teknik Daire Başkanı Fırat AYDIN'a teşekkür ederim.

ONUR SÖZÜ

Yüksek Lisans tezi olarak sunduğum “Türkiye’deki Bölgesel Güç Sistemlerinin Ekonomik Yük Dağıtımının Farklı Algoritmalarla Optimizasyonu” başlıklı bu çalışmanın bilimsel ahlaka ve araştırma etiğine uygun olarak, dışarıdan yardım almadan tarafımdan yazıldığını ve kullandığım kaynakların tümünün metin içinde ve kaynakçada doğru şekilde verildiğini beyan ederim.

Enzel AYDIN

İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
TABLolar DİZİNİ.....	iv
ŞEKİLLER DİZİNİ.....	v
SEMBOLLER VE KISALTMALAR DİZİNİ.....	vi
ÖZET.....	vii
ABSTRACT	viii
1. GİRİŞ	1
1.1 Tezin Amacı.....	2
1.2 Literatürde Yapılan Çalışmalar	3
2. EKONOMİK YÜK DAĞITIMI.....	7
2.1 Ekonomik Yük Dağıtımı ve Problemin Formülasyonu	7
2.2 Kayıpların ve Generatör Limitlerinin Göz Ardı Edildiği EYD	10
2.3 Kayıpların ve Generatör Limitlerinin Dahil Edildiği EYD	14
3. SIRALI EN KÜÇÜK KARELER PROGRAMLAMA ALGORİTMASI	22
3.1 SLSQP'nin Çalışma Mantığı	23
4. GENETİK ALGORİTMA	28
4.1 Genetik Algoritmanın Çalışma Prensibi	28
5. YAPAY ARI KOLONİSİ ALGORİTMASI	30
5.1 Yapay Arı Kolonisi Algoritmasının Temelleri	30
6. TABU ARAMA ALGORİTMASI	34
6.1 Tabu Arama Algoritmasının Çalışma Prensibi	34
7. TAVLAMA BENZETİMİ ALGORİTMASI	36
7.1 Tavlama Benzetimi Algoritmasının Çalışma Prensibi	36
8. UYGULAMA VE DEĞERLENDİRME	38
8.1 EYD Probleminin Uygulanacağı Güç Sistemlerinin Bilgileri.....	38
8.1.1 8 Yüksek Gerilim ve 14 Yük Barası Olmak Üzere 22 Baradan Oluşan Sistem	38
8.1.2 6 Yüksek Gerilim ve 8 Yük Barası Olmak Üzere 14 Baradan Oluşan Sistem	40
8.2 SLSQP Algoritması İle Kayıpsız Ekonomik Yük Dağıtım Sonuçları.....	41
8.3 Genetik Algoritma İle Kayıpsız Ekonomik Yük Dağıtım Sonuçlar	42
8.4 Yapay Arı Kolonisi Algoritması İle Kayıpsız Ekonomik Yük Dağıtım Sonuçları.....	43
8.5 Tabu Arama Algoritması İle Kayıpsız Ekonomik Yük Dağıtım Sonuçları.....	44
8.6 Tavlama Benzetimi Algoritması İle Kayıpsız Ekonomik Yük Dağıtım..... Sonuçları	45
9. SONUÇ VE ÖNERİLER	50
KAYNAKLAR.....	51
ÖZGEÇMİŞ	57

TABLÖLAR DİZİNİ

Tablo 8.1: 22 Baralı Sistemin Bara Numara ve Adları.....	38
Tablo 8.2: Generatör verileri	41
Tablo 8.3: Metasezgisel yöntemlerin maliyet ve süre değerlerinin bazı istatistiksel sonuçları.....	46
Tablo 8.4: Türkiye’de kullanılan 8 generatör ve 22 baralı güç sisteminde kayıpsız.... durumda GA, YAK ve SLSQP ile elde edilen maliyet sonuçları.....	47
Tablo 8.5: Türkiye’de kullanılan 8 generatör ve 22 baralı güç sisteminde kayıpsız.... durumda GA, YAK ve SLSQP ile elde edilen işlem süresi sonuçları.....	47
Tablo 8.6: Türkiye’de kullanılan 6 generatör ve 14 baralı güç sisteminde kayıpsız.... durumda GA, TB, TA ve SLSQP ile elde edilen maliyet sonuçları.....	48
Tablo 8.7: Türkiye’de kullanılan 6 generatör ve 14 baralı güç sisteminde kayıpsız.... durumda GA, TB, TA ve SLSQP ile elde edilen işlem süresi sonuçları.....	49



ŞEKİLLER DİZİNİ

Şekil 2.1 : Yakıt maliyet eğrisi.....	7
Şekil 2.2 : Artan Yakıt Maliyet Eğrisi	8
Şekil 2.3 : Ortak baraya bağlı güç ünitelerinin modeli	9
Şekil 3.1 : SQP Algoritmasının Akış Şeması.....	27
Şekil 4.1 : Genetik Algoritma Akış Şeması	29
Şekil 5.1 : YAK Algoritmasının rolleri ve Doğadaki karşılığı	32
Şekil 5.2 : YAK Algoritması Akış Şeması	33
Şekil 6.1 : Tabu Arama Algoritması Akış Şeması	35
Şekil 7.1 : Tavlama Benzetimi Algoritması akış Şeması.....	37
Şekil 8.1 : 8 Yüksek gerilim ve 14 yük barası olmak üzere 22 baradan oluşan sistem	39
Şekil 8.2 : 380 kV, 6 generatörlü ve 14 baralı sistem	40
Şekil 8.3 : 8 generatör ve 22 baralı sistemin Python programlama dili ile elde edilen SLSQP sonuçları.....	41
Şekil 8.4 : 6 generatör ve 14 baralı sistemin Python programlama dili ile elde edilen SLSQP sonuçları.....	42
Şekil 8.5 : 8 generatör ve 22 baralı sistemin Python programlama dili ile elde edilen GA sonuçları.....	42
Şekil 8.6 : 6 generatör ve 14 baralı sistemin Python programlama dili ile elde edilen GA sonuçları.....	43
Şekil 8.7 : 8 generatör ve 22 baralı sistemin Python programlama dili ile elde edilen YAK sonuçları	43
Şekil 8.8 : 6 generatör ve 14 baralı sistemin Python programlama dili ile elde edilen TA sonuçları	44
Şekil 8.9 : 6 generatör ve 14 baralı sistemin Python programlama dili ile elde edilen TB sonuçları.....	45

SEMBOLLER VE KISALTMALAR

ELD	: Economical Load Distribution (Ekonomik Yük Dağıtımı)
EYD	: Ekonomik Yük Dağıtımı
SLSQP	: Sıralı En küçük Kareler Programlama
GA	: Genetik Algoritma
TA	: Tabu Arama Algoritması
TB	: Tavlama Benzetimi
QP	: Quadratik Programing (İkinci Dereceden Programlama)
SQP	: Sequential quadratic programming (Sıralı İkinci Dereceden Programlama)
DGKÇ	: Doğalgaz Kombine Çevrim
DG	: Doğalgaz
YAK	: Yapay Arı Kolonisi
PSO	: Parçacık Sürü Optimizasyonu
SN	: Saniye
\$	: Dolar
h	: Saat

ÖZET

Yüksek Lisans Tezi

TÜRKİYE’DEKİ BÖLGESEL GÜÇ SİSTEMLERİNİN EKONOMİK YÜK DAĞITIMININ FARKLI ALGORİTMALARLA OPTİMİZASYONU

ENZEL AYDIN

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

57+VIII sayfa

2025

Danışman: Dr. Öğr. Üyesi A. Erhan AKKAYA

Ekonomik yük dağıtımı geçmişten günümüze gelene kadar önemi artarak devam eden bir problem olarak karşımıza çıkmaktadır. Günümüz şartları ve teknolojinin hızla büyümesi ile birlikte enerji talebi hızla artmaktadır. Bu talebi karşılayacak enerji kaynaklarının yetersiz kalması nedeniyle güç sistemlerinin en uygun koşullarda işletilmesi önemli hale gelmiştir. Enerjinin önemli rol oynadığı günümüzde, savaşın, ambargoların ve ekonominin ciddi etkileri de göz önüne alınarak özellikle dünyada enerjiye olan ihtiyacı karşılamak için ülkeler eski kapatılan termik santrallerini dahi yeniden devreye almak için çalışmalar yapmaktadır. Bu nedenle termik santrallerin çalışma maliyetlerinin minimize edilmesi ülkemizde de önemli hale gelmiştir. Enerji maliyetleri düşünülerek katı yakıtlı santrallerin minimum ve maksimum aktif güç değerleri arasında çalışması istenmektedir. Bu noktada ekonomik yük dağıtımı, reaktif güç optimizasyonu ve optimal güç akışı çalışmalarında etkili sonuçlar ortaya koydukça önem kazanmaya devam edecektir. Bu tez çalışmasında Türkiye’de bulunan bölgesel güç santrallerinde Ekonomik Yük Dağıtımı (EYD) probleminin Sıralı En küçük Kareler Programlama (Sequential Least Squares Programming – SLSQP) optimizasyon algoritması ile tezde kullanılan ve literatürde yer alan Genetik Algoritma (GA), Yapay Arı Kolonisi (YAK) algoritması, Tabu Arama (TA) algoritması ve Tavlama Benzetimi (TB) algoritmalarıyla çözümü yer almaktadır. İlk olarak 8 generatör ve 22 baradan oluşan enerji sistemine uygulanan GA, YAK ve SLSQP algoritmalarının 80530 \$/h olan tutarlı maliyet sonuçlarıyla beraber SLSQP algoritmasının 0.00914 saniye gibi kısa süren optimize süresi detaylandırılmıştır. Daha sonra 6 generatör ve 14 baradan oluşan enerji sistemine uygulanan GA, TA, TB ve SLSQP algoritmalarının yakın maliyetler ortaya koyduğu gösterilmektedir. Bununla beraber en uygun maliyet olan 47660 \$/h değerine SLSQP algoritmasının ulaştığı, ayrıca elde edilen bu sonuca da 0.00706 saniye gibi çok kısa bir sürede ulaştığı tablolar halinde anlatılıp detaylandırılmıştır. Kısaca algoritmalar ile hem en düşük maliyet hem de optimum sonuca ne kadar sürede ulaştıklarının sonuçları karşılaştırmalı olarak analiz edilmiştir. Analizler sonucunda generatör maliyetleri minimum olacak şekilde optimum çalışma koşulları belirlenmeye çalışılmıştır.

Anahtar Kelimeler: Ekonomik Yük Dağıtımı, Genetik Algoritma, Güç Sistemleri, Optimizasyon, SLSQP, Yapay Arı Kolonisi, Tabu Arama, Tavlama Benzetimi

ABSTRACT

Master Thesis

OPTIMIZATION OF THE ECONOMIC LOAD DISPATCH OF REGIONAL POWER SYSTEMS IN TURKEY WITH DIFFERENT ALGORITHMS

Enzel AYDIN

Inonu University
Graduate School of Nature and Applied Sciences
Department of Computer Engineering

57+VIII page

2025

Supervisor: Assistant Professor A. Erhan AKKAYA

Economic load dispatch appears as a problem whose importance has steadily increased from the past to the present. With today's conditions and the rapid growth of technology, energy demand is rising sharply. Because the available energy sources have proven insufficient to meet this demand, it has become important to operate power systems under the most favorable conditions. In an era when energy plays a critical role and given the serious impacts of war, embargoes, and economic pressures countries around the world are even working to bring previously decommissioned thermal power plants back online in order to satisfy their energy needs. For this reason, minimizing the operating costs of thermal power plants has also become important in our country. Taking energy costs into account, solid-fuel plants are required to operate between their minimum and maximum active-power limits. At this point, as economic load dispatch, reactive-power optimization, and optimal-power-flow studies continue to yield effective results, their importance will only grow. In this thesis work, the Economic Load Dispatch (ELD) problem at regional power plants in Turkey is solved using the Sequential Least Squares Programming (SLSQP) optimization algorithm together with the Genetic Algorithm (GA), Artificial Bee Colony (ABC) algorithm, Tabu Search (TS) algorithm, and Simulated Annealing (SA) algorithm both those used in this study and those reported in the literature. First, for the 8-generator, 22-bus system, the consistent cost result of \$80,530 \$/h achieved by GA, ABC, and SLSQP is presented, along with the very short optimization time of just 0.00914 seconds for SLSQP. Next, for the 6-generator, 14-bus system, it is shown that GA, TS, SA, and SLSQP produce closely comparable cost outcomes. Moreover, tables detail that SLSQP attains the best cost of \$47,660 \$/h and reaches this solution in only 0.00706 seconds. In short, the algorithms' performances both in terms of lowest cost and convergence time are analyzed comparatively. Based on these analyses, the study seeks to determine the optimal operating conditions that minimize generator costs.

Keywords: Artificial Bee Colony, Economic Load Distribution, Genetic Algorithm, Optimization, Power Systems, SLSQP, Simulated Annealing, Tabu Search

1. GİRİŞ

Enerji üretimi, modern toplumların sürdürülebilirliği ve ekonomik kalkınması için hayati bir öneme sahiptir. Artan enerji talebi ve sınırlı doğal kaynaklar, enerji üretiminde maliyet etkinliği ve verimliliğin sağlanmasını zorunlu kılmaktadır. Bu bağlamda, düşük maliyetle enerji üretiminin önemi giderek artmakta ve enerji üretiminin optimizasyonu için çeşitli yöntemler geliştirilmektedir [1]. Bu çalışmanın amacı, ekonomik yük dağıtımının algoritmalar aracılığıyla nasıl gerçekleştirildiğini açıklamak ve bu işlemin güç sistemleri üzerindeki etkilerini diğer yöntemlerle karşılaştırmalı olarak incelemektir.

Ekonomik yük dağıtımı (EYD), enerji üretim maliyetlerini minimize etmek amacıyla güç sistemlerinde generatörlerin yüklerinin optimal şekilde dağıtılması sürecidir. Bu süreçte, matematiksel modeller ve optimizasyon teknikleri kullanılarak, enerji üretim birimlerinin en verimli şekilde çalışması sağlanır [2]. Yapılan çalışmada, ekonomik yük dağıtımı problemlerinin matematiksel modelleri ve formülasyonları detaylı bir şekilde ele alınacak, bu amaçla kullanılacak optimizasyon metotlarının temel bilgileri verilecek ve çalışma prensipleri açıklanacaktır.

Bölgesel güç sistemlerinin özellikleri, ekonomik yük dağıtımı problemlerinin uygulanabilirliği açısından büyük önem taşımaktadır. Bu nedenle, çalışmada ekonomik yük dağıtımı probleminin yürütüleceği bölgesel enerji tesislerinin özellikleri hakkında bilgi verilecektir. Ayrıca, yapılacak analiz sonuçları literatürdeki diğer metotlarla yapılan EYD süreci sonucunda ulaşılan en düşük yakıt maliyetleri ve çözüm süreleri ile karşılaştırılacak ve bu tezdeki çalışmaların literatüre katkıları değerlendirilecektir.

Bu çalışmada, ekonomik yük dağıtımı problemlerinin çözümünde kullanılan optimizasyon teknikleri, Sıralı En küçük Kareler Programlama SLSQP (Sequential Least Squares Programming), Genetik Algoritma (GA), Yapay Arı Kolonisi (YAK), Tabu Arama (TA) ve Tavlama Benzetimi (TB) algoritmaları olacaktır. Çalışmada SLSQP algoritması detaylı bir şekilde ele alınacaktır. SLSQP algoritması, doğrusal olmayan kısıtlamalar altında doğrusal olmayan optimizasyon problemlerini çözmek için kullanılan etkili bir yöntemdir. Bu

algoritmanın temel prensipleri ve çalışma mekanizması detaylı bir şekilde açıklanacaktır. Ayrıca SLSQP algoritmasının diğer yöntemler ile benzer yada daha iyi sonuçlar ortaya koymasının yanı sıra karmaşık matematiksel optimizasyon problemlerinde çözüm sürelerinin de ne kadar önemli olduğunu gösterip diğer algoritmalar ile aralarındaki farka dikkat çekecek ve ekonomik yük dağıtım problemlerine uygulanabilirliği değerlendirilecektir.

Sonuç olarak bu çalışma;

- Enerji üretiminde maliyet etkinliğini artırmak ve güç sistemlerinin verimliliğini optimize etmek amacıyla ekonomik yük dağıtım problemlerinin çözümüne yönelik kapsamlı bir yaklaşım sunmayı hedeflemektedir.
- Kapsamlı yaklaşım ile SLSQP algoritması ile metasezgisel algoritmalar arasındaki maliyet sonuçları ve optimizasyon sürelerinin karşılaştırmalı analizinin yapılması hedeflenmektedir.

Bu bağlamda, ekonomik yük dağıtım problemlerinin matematiksel modelleri, optimizasyon teknikleri ve bölgesel güç sistemleri üzerindeki etkileri detaylı bir şekilde incelenecektir.

1.1 Tezin amacı

Günümüzün enerji ihtiyaçlarını karşılamak, giderek artan bir önem kazanmakta ve yeni enerji üretim tesislerinin kurulumu yüksek maliyetlerle birlikte gelmektedir. Bu nedenle, mevcut enerji üretim tesislerinin optimizasyonu, maliyetleri azaltmanın ve enerji verimliliğini artırmanın en etkili yollarından biridir. Bu çalışmada, güç santrallerinin ekonomik yük dağıtım (EYD) ile optimal işletilmesi amaçlanmaktadır.

Ekonomik yük dağıtım, santraller arasındaki yük paylaşımını düzenleyerek sistemlerin verimli çalışmasını sağlar. EYD'nin temel hedefi, enerji üretim birimleri arasında güç paylaştırarak toplam işletme maliyetlerini en aza indirmektir. Bu süreç, santrallerin en düşük maliyetle talep edilen enerjiyi karşılamasına olanak tanır ve enerji üretim maliyetlerinin düşürülmesine katkıda bulunur.

Bu çalışmada, Türkiye'deki bazı termik santrallerin EYD problemini çözmek için SLSQP (Sequential Least Squares Programming), Genetik Algoritma (GA), Yapay Arı Kolonisi (YAK), Tabu Arama (TA) ve Tavlama Benzetimi (TB) optimizasyon algoritmaları kullanılacaktır. Özellikle SLSQP algoritmasının diğer algoritmalar ile karşılaştırmalı

analizleri yapılarak, generatörlerin en düşük maliyetle, en kısa sürede çalışma koşulları belirlenmeye çalışılacaktır. Bu sayede, mevcut enerji üretim tesislerinden maksimum fayda sağlanması hedeflenmektedir.

Ayrıca yapılan çalışmada ekonomik yük dağıtım problemlerinin matematiksel modelleri, formülasyonları ve bundan dolayı kullanılacak olan optimizasyon metotlarının temel bilgileri verilir ve çalışma prensipleri ve yöntemleri anlatılacaktır. Ekonomik yük dağıtım problemlerinin yürütüleceği bölgesel enerji tesislerinin özellikleri hakkında bilgi verilecektir. Yapılacak olan analiz sonuçları literatürdeki yer alan diğer metotlar ile yapılan ekonomik yük dağıtım süreci sonucunda elde edilen en düşük yakıt maliyetleri geçmişte yapılan çalışmalarda bulunan sonuçlarla karşılaştırılıp ve bu tezde yapılan çalışmaların verdiği katkılardan söz edilecektir.

Çalışmanın bir diğer önemli amacı, EYD problemini en verimli şekilde çözebilecek optimizasyon algoritmasını belirlemektir. Bu ideal optimizasyon algoritması, enerji üretim maliyetlerini minimize ederken, aynı zamanda enerji kaynaklarının daha verimli kullanılmasını sağlayacaktır. Enerji üretimindeki bu verimlilik, hem ekonomik avantajlar sunacak hem de enerji kaynaklarının sürdürülebilir kullanımını destekleyecektir.

Sonuç olarak, bu çalışma, mevcut enerji üretim tesislerinin optimize edilerek maliyetlerin düşürülmesini ve enerji verimliliğinin artırılmasını amaçlamaktadır. Bu amaç doğrultusunda gerçekleştirilecek optimizasyon çalışmaları, enerji sektöründe önemli kazanımlar elde edilmesine olanak tanıyacaktır.

1.2 Literatürde yapılan çalışmalar

EYD ile ilgili literatür taramasında karşımıza, EYD konusunda farklı enerji tesislerinin kullanıldığı, enerji tesisleri için bazı kısıtlamaların dahil edildiği ve bazı kısıtlamaların dahil edilmediği çalışmaların olduğu görülmektedir.

Türkay 2002 yılında yaptığı çalışmada, jeneratörlerin minimum maliyet ile yüklenebilmesi için genetik algoritmayla Ekonomik Yük Dağıtım sorununu incelemiştir [3].

2003 yılında Sinha ve arkadaşları yaptıkları çalışmalarında, kaotik karınca sürüsü optimizasyon algoritmasını kullanarak Ekonomik Yük Dağıtım problemini incelemişlerdir [4].

Bouzeboudja ve arkadaşları yaptıkları çalışmalarında, IEEE'nin 25 baralı sisteminin üzerinde 2005 yılında genetik algoritmayı başka yöntemler kullanarak EYD sorununa çözüm önermişlerdir [5].

2008 yılında Alrashidi ve El-Hawary yaptıkları çalışmalarında, IEEE'nin alan 30 baralı enerji test sisteminde parçacık sürü optimizasyonu ve başka yöntemlerle çevresel ekonomik dağıtım sorununu çözmüşlerdir [6].

Coelho ve Mariani yaptıkları çalışmalarında, kaos teorisi ile birlikte yapay bağışıklık ağını kullanarak 13 üniteye sahip bir enerji tesisinin yakıt maliyet optimizasyonu çözmüşlerdir [7].

2009 yılında yapılan çalışmada Özyön, Genetik Algoritma metoduyla çevresel ekonomik güç dağıtım sorununun çözümü için incelemiştir. Ele aldığı problemler sadece normal termik üretim birimlerinden oluşan sistemler, kısa dönem hidrotermal koordinasyon problemleri ve ham enerji kaynağı sınırlı termik birimlerin yer aldığı kısa dönem hidrotermal koordinasyon sorunlarından oluşmaktadır [8].

Çetin, ekonomik yük dağıtım sorununu 2011 yılında yapılan çalışmada, sezgisel yöntemlerden olan PSO, YAK Optimizasyonu ve Diferansiyel Gelişim Optimizasyonu algoritmaları ile, 6 generatörlü ve 40 generatörlü test sistemlerine uygulamıştır. [9].

Yang ve ekibi konveks olmayan EYD sorunlarına çözüm getirmek için 2012 yılındaki çalışmalarında ateş böceği algoritmasına dayanan yeni bir yöntem geliştirmiştir. Bu yöntemde, valf noktası etkisi, rampa oranı sınırları ve yasak işletim bölgeleri gibi doğrusal olmayan birçok özellik göz önünde bulundurulmuştur. Çalışmada, dört farklı test sistemi kullanılmış ve yöntemin performansı literatürde yer alan çalışmalar ile birlikte değerlendirilmiştir [10].

2013 yılında yapılan bir çalışmada Özyön ve ekibi, jeneratörlerin artırma ve azaltma sınırlarını dikkate alarak yasak işletim bölgeleriyle ilgili EYD sorununu Diferansiyel Gelişim Algoritması kullanarak çözümlenmişlerdir. Bu algoritma, altı ve on beş generatörden oluşan iki ayrı test sisteminde uygulanmıştır [11].

Slimani ve Bouktir, 2013 yılında emisyon kontrollü EYD sorununu çözmek için YAK algoritmasını kullanmışlardır. Bu algoritma, IEEE 30 baralı, 6 jeneratörlü bir test sisteminde ve Cezayir'deki 59 baradan oluşan enerji tesisine uygulanmıştır [12].

2014 yılında yapılan bir çalışmada, Kaur ve ekibi, hat kayıplarını dikkate almadan termal generatörlerin EYD sorununu ele almış ve bu sorunu Genetik Algoritma yöntemi ile üç ve altı jeneratörden oluşan test sistemlerinde çözmüşlerdir [13].

Elattar, dinamik EYD sorununu çözüm getirmek amacıyla hibrit genetik algoritma ve bakteri yem arama algoritmasını kullanmıştır. 2015 yılındaki bu araştırmasında, valf noktası etkileri, rampa oranı sınırları ve hat kayıplarını dikkate almıştır [14].

2015 yılında Subathra ve arkadaşları enerji üretim sistemlerinde ekonomik yük dağıtım problemini çözmek için çapraz entropi (cross-entropy (CE)) algoritması ve sıralı kuadratik programlama (SQP) algoritmalarından faydalanarak hibrit bir yaklaşım sunmuşlardır [15].

2016 yılında Turgut ve arkadaşları EYD probleminin çözümü için yapay işbirlikçi algoritması kullanmışlardır. Yöntemin EYD problemlerinde etkinliği için iki farklı test sistemine uygulayarak göstermişlerdir [16].

2017 yılında Eminoglu ve Karahan EYD problemini diferansiyel gelişim algoritması ve parçacık sürü optimizasyonu metotları kullanarak elde edilen sonuçların Grafik Kullanıcı Arayüzü tasarımı ile daha kolay bir şekilde yapılmasına imkan sağlamışlardır [17].

2017 yılında Trived ve arkadaşları yaptıkları çalışmalarında, Balina Optimizasyon Algoritması Kullanarak emisyon kısıtlı ekonomik dağıtım problemi Çözümü için Fiyat Cezası Faktörlerine Dayalı Yaklaşım kullanmışlardır [18] .

2020 yılında soni ve arkadaşları ekonomik yük dağıtım sorununun cevabını aramak için yapay arı kolonisi algoritmasından faydalanarak, YAK algoritmasının etkinliği ve verimliliği kanıtlamak adına diferansiyel değerlendirme, yapay kolon optimizasyonu,hibrit sürü akıllı karınca optimizasyonu gibi tekniklerle kıyaslamışlardır [19].

2021 yılında Said ve arkadaşları yaptıkları çalışmalarında, Ekonomik Yük Dağıtım problemini Verimli Bir Bukalemun Sürü Algoritması ile bukalemunun ağaçlarda, çöllerde ve bataklıkların yakınında yiyecek avının dinamiklerini hesaba katan bir algoritma ile çözmüşlerdir [20] .

2022 yılında Kaur ve arkadaşları yaptıkları çalışmalarında, generatörlerin doğrusal olmayan özelliklerine sahip ekonomik yük dağıtım problemini çözmek ve iletim kayıplarını göz önünde bulundurarak Modifiye Krill Sürü Algoritması kullanmışlardır [21].

2022 yılında Tahir ve arkadaşları yaptıkları çalışmalarında, Ekonomik Yük Dağıtım Sorununu Çözmek için İyileştirilmiş Kondisyona Bağlı Optimize Edici kullanmışlardır [22].

2022 yılında Berahmandpour ve arkadaşları yaptıkları çalışmalarında, rüzgar enerjisini içeren yeni bir esnekliğe dayalı risk sınırlayıcı ile yöntemde yazarlar tarafından ortaya konulan üretim sistemi esneklik endeksi ekonomik değere dönüştürülerek toplam işletme maliyeti fonksiyonuna dâhil edilerek dinamik ekonomik yük dağıtım problemi için bir çözümü sunmuşlardır [23].

2023 yılında Ragunathan ve Ramadoss ekonomik yük dağıtımını problemini ele almak için yeni bir metaheuristik algoritma olan Altın Çakal Optimizasyon'unu kullanmışlardır. Bu yöntem altın çakalların işbirlikçi saldırı davranışlarını kullanır. Önerilen yöntemi parçacık sürüsü optimizasyonu, arı kolonisi optimizasyonu ve geri izleme arama algoritması ile karşılaştırarak EYD problemlerinde etkinliğini göstermişlerdir [24].

2023 yılında EKE ve arkadaşları Kaotik parçacık sürü optimizasyonu (CPSO) ile EYD probleminin çözümünü amaçlamışlardır. CPSO'yu 6 üretim ünitesine sahip bir test sistemine uygulamışlardır. Elde edilen sonuçları literatürde yer alan yapay arı kolonisi ve diferansiyel gelişim optimizasyonu ile karşılaştırmışlardır [25].

2024 yılında Zhang ve Li Ekonomik Yük Sevkiyatı probleminin çözümü için geliştirilmiş Pelikan Optimizasyon algoritması(IPOA) sunmaktadırlar. IPOA' yı 10, 40 ve 80 boyutlu termik santrallere uyguladıklarında sırasıyla, %0,0292, %2,7273, %3,6739 gibi kömür tüketimlerinde azalma tespit etmişlerdir. Bu da enerji santrallerinin yakıt maliyetini önemli ölçüde azaltabileceğine değinmişlerdir [26].

2024 yılında Shaban ve arkadaşları Büyüme Optimize Edici Algoritmasını kullanarak enerji sistemlerindeki en büyük sorun olan ekonomik yük dağıtımını sorununu çözmek için kullanmışlardır. Bu yeni metaheuristik yöntem ile gri kurt optimizasyonu, fil sürü optimizasyonu, tulumlular sürü optimizasyonu gibi algoritmalar ile karşılaştırmışlardır [27].

2. EKONOMİK YÜK DAĞITIMI

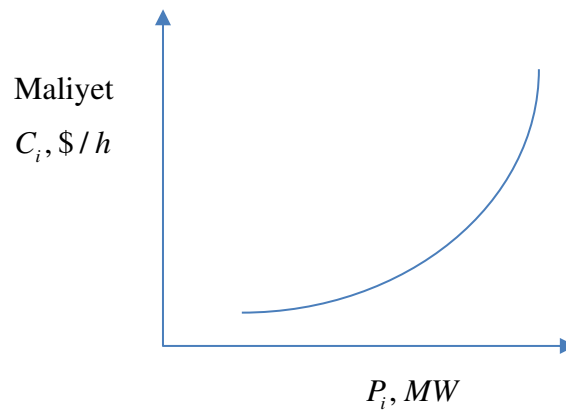
Bu bölümde, ekonomik yük dağıtımının tanımı, matematiksel formülasyonu ve iletim hattı kayıplarının dahil edildiği ile edilmediği durumlara ilişkin açıklamalar ve formüller sunulmaktadır.

2.1 Ekonomik yük dağıtımı ve problemin formülasyonu

Enerji üretim süreçlerinde, maliyet ve iletim kayıpları gibi bazı kısıtlamalar göz önünde bulundurularak sistemlerin en verimli şekilde işletilmesi büyük önem taşır. Bu bağlamda, santraller arasında yük paylaşımı yapmak, enerji üretim maliyetlerini düşürmenin etkili yollarından biridir. Ekonomik yük dağıtımı, enerji santrallerinin en düşük maliyetle ihtiyaç duyulan enerjiyi üretmesini sağlayarak bu yük paylaşımının optimize edilmesini hedefler. Yapılan analizler sonucunda elde edilen veriler, santrallerin ekonomik olarak yüklendiğinde sistem maliyetlerinde belirgin bir azalma olduğunu gösterir.

Bu süreç, enerji üretim birimleri arasında yükün dengeli bir şekilde dağıtılmasına dayanır. Ekonomik yük dağıtımının amacı, toplam işletme maliyetlerini en aza indirerek enerji üretim süreçlerini daha verimli hale getirmektir. Böylelikle, enerji santralleri sadece maliyetleri düşürmekle kalmaz, aynı zamanda enerji kaynaklarının daha etkin kullanılmasına da katkıda bulunur.

Yani ekonomik yük dağıtımının temel amacı, üretim birimleri arasında gücü paylaştırarak toplam işletme maliyetlerini en aza indirmektir. Bir üretim biriminin güç üretimi ile yakıt maliyet eğrisi arasındaki ilişki, Şekil 2.1'de gösterilmektedir. [28].



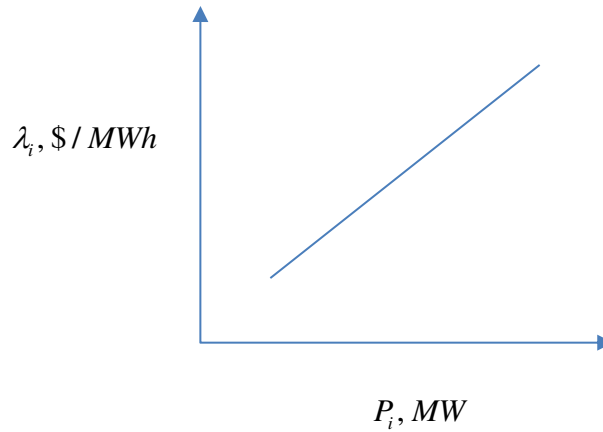
Şekil 2.1 : Yakıt maliyet eğrisi [28]

Şekil 2.1' den elden edilen veriler, bir generatörün ürettiği güçle ilişkili maliyet fonksiyonunun, ikinci dereceden bir fonksiyon olduğunu göstermektedir ve bu fonksiyon Denklem 2.1' de yer almaktadır [28].

$$C_i = a_i + b_i P_i + c_i P_i^2 \quad (2.1)$$

Denklem 2.1'de C_i , i. generatörün üretim masrafını, a , b ve c generatör maliyet katsayılarını, P_i ise i. generatörün ürettiği aktif gücü simgeler. $P_i = 0$ olduğunda, $C_i = a_i$ değerine eşittir. Bu durumda a_i santralin kurulum masraflarıyla, b_i ve c_i ise işletme sırasında ortaya çıkan masraflarla ilişkilidir.

Yakıt maliyet eğrisinin aktif güce göre türevi alındığında, yükselen yakıt maliyet eğrisi elde edilir. Bu eğri şekil 2.2'de gösterilmekte ve Denklem 2.2 'deki gibi formülize edilmektedir [28].



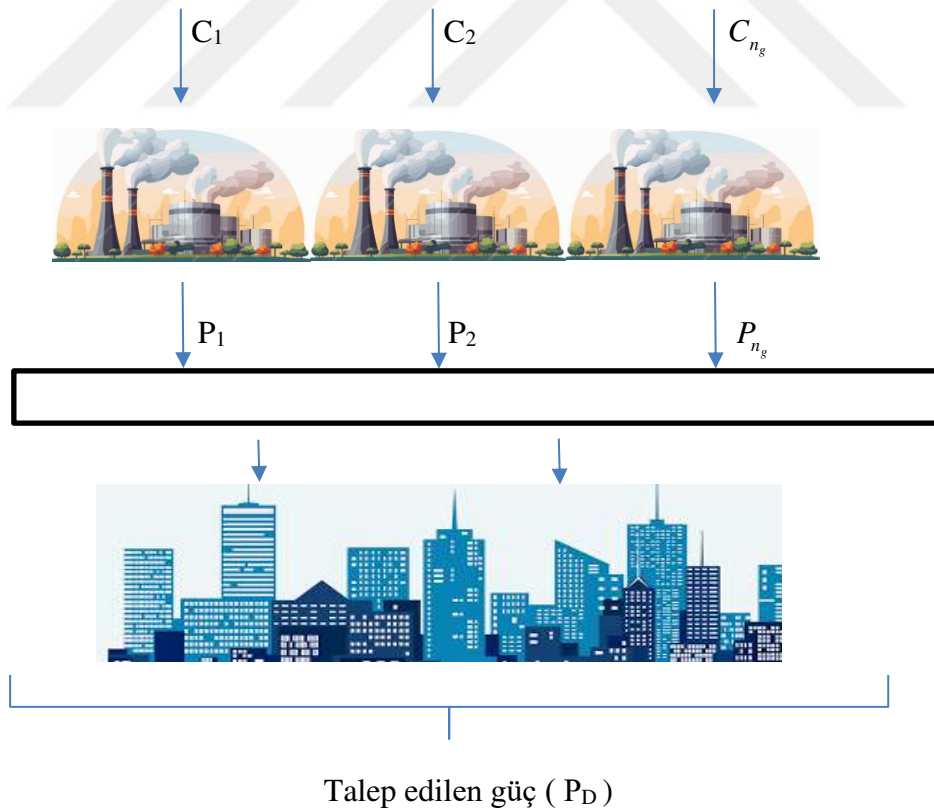
Şekil 2.2 : Artan yakıt maliyet eğrisi [28]

$$\frac{dC_i}{dP_i} = 2c_i P_i + b_i \quad (2.2)$$

Artan yakıt maliyet eğrisi, ekstra enerji üretiminin maliyetini ortaya koyar. Toplam üretim maliyetine yakıt giderleriyle beraber iş gücü, kaynaklar ve bakım masrafları da dahildir. Bu maliyetler, genellikle yakıt maliyetinin belirli bir oranı olarak kabul edilir ve artan yakıt maliyet eğrisine yansıtılır [28]. Her generatörün ürettiği güce göre hesaplanan bu masrafların toplamı, toplam üretim maliyetini oluşturur ve Denklem 2.3 'te gösterilmektedir [28].

$$C_t = \sum_{i=1}^{n_g} C_i \quad (2.3)$$

Denklem 2.3'te C_t toplam maliyeti, n_g ise güç sistemindeki generatörlerin toplam sayısını gösterir. Ekonomik dağıtımın nasıl yapılacağı, kayıpların ve generatörlerin güç sınırlarının dikkate alınıp alınmayacağına bağlı olarak farklı şekillerde formülize edilmektedir. Ekonomik yük dağıtım problemi şekil 2.3'te görselleştirilmiştir.



Şekil 2.3 : Ortak baraya bağlı güç ünitelerinin modeli [28]

2.2 Kayıpların ve generatör limitlerinin göz ardı edildiği ekonomik yük dağıtımı

En temel ekonomik yük dağıtım problemi, hat kayıplarının göz ardı edildiği durumlarda çözülmektedir. Bu durumda, her generatörün üreteceği aktif güç belirlenir. Hat kayıpları ihmal edildiğinde, üretilen toplam enerji, talep edilen toplam enerjiyle aynı olmalıdır. Bu durumda Denklem 2.4'teki gibi ifade edilir [28].

$$\sum_{i=1}^{n_g} P_i = P_D \quad (2.4)$$

Denklem 2.4'te P_i , i.generatörün üreteceği aktif gücü, P_D ise demand yani toplam gücü belirtmektedir.

Ekonomik Yük Dağıtımı (EYD) problemlerinde, kısıtlama fonksiyonunun amaç fonksiyonuna dahil edilmesi, Denklem 2.4'te gösterilen güç eşitliğine Lagrange çarpanının eklenmesiyle gerçekleştirilir ve bu işlem aşağıdaki gibi ifade edilir [28].

$$L = C_t + \lambda(P_D - \sum_{i=1}^{n_g} P_i) \quad (2.5)$$

Bu fonksiyonun minimum değeri, Lagrange fonksiyonunun her bir bağımsız değişkenine ait kısmi türevlerinin sıfıra eşitlendiği noktada bulunur. Bu denklemler sırasıyla Denklem 2.6 ve Denklem 2.7'de yer almaktadır [28].

$$\frac{\partial L}{\partial P_i} = 0 \quad (2.6)$$

$$\frac{\partial L}{\partial \lambda} = 0 \quad (2.7)$$

Denklem 2.6 incelendiğinde:

$$\frac{\partial C_t}{\partial P_i} + \lambda (0 - 1) = 0 \quad (2.8)$$

$$C_t = C_1 + C_2 + \dots + C_{n_g} \quad (2.9)$$

$$\frac{\partial C_t}{\partial P_i} = \frac{dC_i}{dP_i} = \lambda \quad (2.10)$$

$$\frac{dC_i}{dP_i} = \lambda \quad i = 1, \dots, n_g \quad (2.11)$$

$$b_i + 2c_i P_i = \lambda \quad (2.12)$$

elde edilir [28].

Denklem 2.7’de yer alan Lagrange fonksiyonunun Lagrange çarpanına göre kısmi türevi alındığında aşağıda yer alan Denklem 2.13 ortaya çıkmaktadır [28].

$$\sum_{i=1}^{n_g} P_i = P_D \quad (2.13)$$

Özetle, hat kayıpları ve generatör sınırları göz ardı edildiğinde, en ekonomik işletim için tüm generatörlerin Denklem 2.13’e uygun olarak aynı artan üretim maliyetiyle çalışması gerekmektedir.

Denklem 2.12’de P_i eşitliğin bir tarafına alınıp yalnız bırakıldığında aşağıda Denklem 2.14’te yer alan koordinasyon denklemi ortaya çıkar [28].

$$P_i = \frac{\lambda - b_i}{2c_i} \quad (2.14)$$

Denklem 2.13'te Denklem 2.14'te bulunan P_i eşitliği yerine yazılırsa Denklem 2.15 ve Denklem 2.16 elde edilir [28].

$$\sum_{i=1}^{n_g} \frac{\lambda - b_i}{2c_i} = b \quad (2.15)$$

$$\lambda = \frac{P_D \sum_{i=1}^{n_g} \frac{b_i}{2c_i}}{\sum_{i=1}^{n_g} \frac{1}{2c_i}} \quad (2.16)$$

Belirlenen λ değeri, Denklem 2.14'e uygulanarak generatörlerin en uygun maliyetle üretmesi gereken güç değerleri hesaplanır.

$$f(\lambda) = P_D \quad (2.17)$$

Denklem 2.17'de eşitliğin $f(\lambda)$ kısmını $\lambda^{(k)}$ çevresinde Taylor serisine açıp ve yüksek dereceli terimleri göz ardı ederek, Denklem 3.18'i buluruz [28].

$$f(\lambda)^k \left(\frac{df(\lambda)}{d\lambda} \right)^{(k)} \Delta\lambda^{(k)} = P_D \quad (2.18)$$

Denklem 2.18 aşağıda yer alan Denklem 2.19 ve Denklem 2.20 şeklinde de gösterilir [28].

$$\Delta\lambda^{(k)} = \frac{\Delta P^{(k)}}{\left(\frac{df(\lambda)}{d\lambda}\right)^{(k)}} = \frac{\Delta P^{(k)}}{\sum \left(\frac{dP_i}{d\lambda}\right)^{(k)}} \quad (2.19)$$

$$\Delta\lambda^{(k)} = \frac{\Delta P^{(k)}}{\sum \frac{1}{2c_i}} \quad (2.20)$$

İteratif olarak çözüme gidileceği için, bir sonraki yinelemede bulunan λ değeri, Denklem 2.21’de belirtildiği şekilde yer almaktadır [28].

$$\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)} \quad (2.21)$$

Talep gücü ile üretilen güç arasındaki farkı hesaplamak için Denklem 2.22 kullanılır [28].

$$\Delta P^{(k)} = P_D - \sum_{i=1}^{n_g} P_i^{(k)} \quad (2.22)$$

Denklem 2.22’de gösterilen $\Delta P^{(k)}$ eşitliği, $\Delta P^{(k)}$ ’nin belirli bir doğruluktan küçük olmasına kadar devam eder [28].

Güç sistemlerinin belirli bir alt ve üst limitler içerisinde çalıştırılması istenirse aşağıda yer alan Denklem 2.23 şartı oluşmalıdır [28].

$$P_{i(\min)} \leq P_i \leq P_{i(\max)} \quad (2.23)$$

Bu durumda, $P_{i(\min)}$ generatörün en düşük çalışma sınırını, $P_{i(\max)}$ da en yüksek çalışma sınırını temsil eder. Bu koşullarda, λ değerleri aşağıdaki şekilde belirlenmelidir [28].

$$\begin{aligned} \frac{dC_i}{dP_i} &= \lambda, & P_{i(\min)} < P_i < P_{i(\max)} \\ \frac{dC_i}{dP_i} &\leq \lambda, & P_i &= P_{i(\max)} \\ \frac{dC_i}{dP_i} &\geq \lambda, & P_i &= P_{i(\min)} \end{aligned} \quad (2.24)$$

Belirlenen λ değeri kullanılarak P_i değerleri Denklem 2.14'e göre hesaplanır ve Denklem 2.13 sağlanana kadar yineleme süreci devam eder. Eğer generatörlerden birinin güç çıkışı en düşük veya en yüksek limite ulaşırsa, geldiği noktada sabitlenir. Dolayısıyla, generatörün güç çıkışı sabitlenir ve yalnızca sınırlar arasında kalan generatörler eşit artan maliyet değerine göre çalışır [28].

2.3 Kayıpların ve generatör limitlerinin dahil edildiği ekonomik yük dağıtımı

Eğer iletim mesafesi kısa ve yük yoğunluğu yüksekse, iletim hattı kayıpları genellikle göz ardı edilebilir ve generatörlerin üretim maliyetleri eşit artışlarla optimum dağıtımı sağlayabilir. Ancak geniş ve karmaşık bağlantılı sistemlerde, enerji uzun mesafelere ve düşük yük yoğunluklu alanlara taşındığında kayıplar önemli bir hale gelir ve bu kayıplar optimum dağıtımı etkiler. Üretilen gücü kullanıcıya aktarmak için iletim hatları kullanılır, ancak bu hatlar boyunca dirençten kaynaklı olarak bir miktar güç kaybı yaşanır. Bu durumda, iletim kayıplarını hesaba katan denklemler kullanılarak toplam iletim hattı kayıpları

hesaplanır ve bu kayıplar generatör güç çıkışlarına bağlı olarak Denklem 2.25'te gösterilen ikinci dereceden bir denklemle ifade edilir [28].

$$P_L = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} P_i B_{ij} P_j \quad (2.25)$$

Hat kayıpları Kron tarafından geliştirilen ve Kirchmayer tarafından benimsenen şekilde modellenir [29]. Bu modele göre hat kayıpları matris formunda yazılmaktadır.

$$P_L = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} P_i B_{ij} P_j + \sum_{i=1}^{n_g} B_{0i} P_i + B_{00} \quad (2.26)$$

B_{ij} : NxN boyutlu kayıp katsayıları matrisi

B_{0i} : 1xN boyutlu kayıp katsayıları matrisi

B_{00} : Boyutsuz kayıp katsayıları matris

Ekonomik Yük Dağıtım sorunlarına kayıp katsayılarını katarken, sistemde üretilmesi gereken enerji miktarı, talep edilen toplam enerji ile iletim hatlarındaki enerji kayıplarının toplamı ile eşitlenmelidir [28].

Bu ifade Denklem 2.27'de gösterilmiştir.

$$\sum_{i=1}^{n_g} P_i = P_D + P_L \quad (2.27)$$

$P_D =$ Demand yani Talep Güç

$P_L =$ Toplam Hat Kaybı

Hat kayıplarının dahil edilmediği EYD probleminin denklem 2.23'deki sınırlama fonksiyonu hesaba dahil edildiğinde Langrange çarpanının fonksiyonu Denklem 2.28'de yer aldığı gibi bulunur [28].

$$L = C_t + \lambda(P_D + P_L - \sum_{i=1}^{n_g} P_i) + \sum_{i=1}^{n_g} \mu_{i(\max)} (P_i - P_{i(\max)}) + \sum_{i=1}^{n_g} \mu_{i(\min)} (P_i - P_{i(\min)}) \quad (2.28)$$

Eğer P_i , $P_{i(\max)}$ 'dan küçükse $\mu_{i(\max)} = 0$ olarak kabul edilir; P_i , $P_{i(\min)}$ 'den büyükse $\mu_{i(\min)} = 0$ olarak kabul edilir. Bu, generatörün üreteceği gücün maksimum veya minimum limit değerlerine uyduğu durumlarda, μ değerinin sıfır alınması ve buna karşılık gelen terimin denklemde yer almaması anlamına gelir. Fonksiyonun minimum noktaları, fonksiyonun kısmi türevleri alınarak ve bu türevler sıfıra eşitlenerek belirlenir [28].

Bu durumda Denklem 2.28'in minimumunu elde etmek için aşağıda yer alan denklemlere ek olarak Denklem 2.6 ve Denklem 2.7 dikkate alınır [28].

$$\frac{\partial L}{\partial \mu_{i(\max)}} = (P_i - P_{i(\max)}) = 0 \quad (2.31)$$

$$\frac{\partial L}{\partial \mu_{i(\min)}} = (P_i - P_{i(\min)}) = 0 \quad (2.32)$$

Denklem 2.8’de yer alan Lagrange çarpanının fonsiyonu denklem 2.6’a uygulanırsa aşağıda yer alan Denklemlere ulaşılır [28].

$$\frac{\partial C_t}{\partial P_i} + \lambda(0 + \frac{\partial L}{\partial P_i} - 1) = 0 \quad (2.31)$$

$$C_t = C_1 + C_2 + + C_{n_g} \quad (2.32)$$

$$\frac{\partial C_t}{\partial P_i} = \frac{dC_i}{dP_i} \quad (2.33)$$

$$\frac{dC_i}{dP_i} + \lambda \frac{\partial P_L}{\partial P_i} = \lambda \quad i = 1,, n_g \quad (2.34)$$

$\frac{\partial P_L}{\partial P_i}$ ifadesi hattın artan iletim kaybı olarak gösterilir. Denklem 2.34 düzenlenirse Denklem 2.35 ve Denklem 2.36 eşitliklerine ulaşılır [28].

$$\left(\frac{1}{1 - \frac{\partial P_L}{\partial P_i}} \right) \frac{dC_i}{dP_i} = \lambda \quad i = 1,, n_g \quad (2.35)$$

$$L_i \frac{dC_i}{dP_i} = \lambda \quad i = 1, \dots, n_g \quad (2.36)$$

Denklem 2.36’da yer alan L_i i.generatorün penalty faktörü olarak isimlendirilir ve Denklem 2.37’deki şekliyle yazılır [28].

$$L_i = \frac{1}{1 - \frac{\partial P_L}{\partial P_i}} \quad (2.37)$$

Denklem 2.2’de gösterilen artan üretim maliyet fonksiyonu, Denklem 2.26’daki fonksiyonlar bir arada kullanılırsa Denklem 2.38 ortaya çıkmış olur [28].

$$\frac{\partial P_L}{\partial P_i} = 2 \sum_{j=1}^{n_g} B_{ij} P_j + B_{0i} \quad (2.38)$$

Denklem 2.2’de ve Denklem 2.38’de yer alan fonksiyonları alıp Denklem 2.34’te yerlerine yazdığımızda aşağıda yer alan denklemler elde edilir [28].

$$\lambda = b_i + 2c_i P_i + 2\lambda \sum_{j=1}^{n_g} B_{ij} P_j + B_{0i} \lambda \quad (2.39)$$

veya

$$\left(\frac{c_i}{\lambda} + B_{ii}\right) P_i + \sum_{j=1, j \neq i}^{n_g} B_{ij} P_j = \frac{1}{2} \left(1 - B_{0i} - \frac{b_i}{\lambda}\right) \quad (2.40)$$

Sistemde yer alan tüm generatörlere Denklem 3.40'da verilen denklem uygulandığında, aşağıdaki matris denklemi ortaya çıkar [28].

$$\begin{bmatrix} \frac{c_1}{\lambda} + B_{11} & B_{12} & \dots & B_{1n_g} \\ B_{21} & \frac{c_2}{\lambda} + B_{22} & \dots & B_{2n_g} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n_g1} & B_{n_g2} & \dots & \frac{c_{n_g}}{\lambda} + B_{n_g n_g} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 - B_{01} - \frac{b_1}{\lambda} \\ 1 - B_{02} - \frac{b_2}{\lambda} \\ \vdots \\ 1 - B_{0n_g} - \frac{b_{n_g}}{\lambda} \end{bmatrix} \quad (2.41)$$

Matris denkleminin kısa hali Denklem 2.42'deki gibi gösterilir [28].

$$EP=D \quad (2.42)$$

Optimum dağıtım çözümünü bulmak için belirlenmiş bir λ^1 değerinde, Denklem 2.42'de verilen denklem aynı anda çözümlenmelidir. Çözümleri hızlı yapmak amacıyla gradient yöntemi kullanılır. Çünkü çözüm, yinelemeli bir şekilde bulunduğu için, k'ıncı yinelemedeki P_i Denklem 2.43'teki gibidir.

$$P_i^{(k)} = \frac{\lambda^{(k)}(1 - B_{0i}) - b_i - 2\lambda^{(k)} \sum_{j \neq i} B_{ij} P_j^{(k)}}{2(c_i + \lambda^{(k)} B_{ii})} \quad (2.43)$$

Denklem 2.43'teki P_i 'yi Denklem 2.37'de yerine yazdığımızda Denklem 2.44 bulunur [28].

$$P_D + P_L^{(k)} = \sum_{i=1}^{n_g} \frac{\lambda^{(k)}(1 - B_{0i}) - b_i - 2\lambda^{(k)} \sum_{j \neq i} B_{ij} P_j^{(k)}}{2(c_i + \lambda^{(k)} B_{ii})} \quad (2.44)$$

Aşağıda yer alan Denklem 2.45 şeklinde de gösterilir [28].

$$f(\lambda^{(k)}) = P_D + P_L^{(k)} \quad (2.45)$$

Denklem 2.45'in $f(\lambda^{(k)})$ kısmını $\lambda^{(k)}$ işletim noktası etrafında Taylor serisi ile genişlettiğimizde ve yüksek dereceli ifadeleri göz ardı ettiğimizde, Denklem 2.46 'ya ulaşırız.

$$f(\lambda^{(k)}) + \left(\frac{df(\lambda)}{d\lambda}\right)^{(k)} \Delta\lambda^{(k)} = P_D + P_L^{(k)} \quad (2.46)$$

$$\Delta\lambda^{(k)} = \frac{\Delta P^{(k)}}{\left(\frac{df(\lambda)}{d\lambda}\right)^{(k)}} = \frac{\Delta P^{(k)}}{\sum \left(\frac{dP_i}{d\lambda}\right)^{(k)}} \quad (2.47)$$

$$\sum_{i=1}^{n_g} \left(\frac{\partial P_i}{\partial \lambda}\right)^{(k)} = \sum_{i=1}^{n_g} \frac{c_i(1 - B_{0i}) + B_{ii}b_i - 2c_i \sum_{j \neq i} B_{ij} P_j^{(k)}}{2(c_i + \lambda^{(k)} B_{ii})^2} \quad (2.48)$$

$$\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)} \quad (2.49)$$

$$\Delta P^{(k)} = P_D + P_L^{(k)} - \sum_{i=1}^{n_g} P_i^{(k)} \quad (2.50)$$

$\Delta P(k)$ belirlenen sınırın altına düşene kadar iterasyon süreci devam eder. Kayıp formülü ise Denklem 2.51'de belirtilmiştir [28].

$$P_L = \sum_{i=1}^{n_g} B_{ii} P_i^2 \quad (2.51)$$

$B_{ij} = 0$, $B_{00} = 0$ olacak şekilde alınırsa, Denklem 2.43'ün basit şekli Denklem 2.52'de gösterildiği gibi olur [28].

$$P_i^{(k)} = \frac{\lambda^{(k)} - b_i}{2(c_i + \lambda^{(k)} B_{ii})} \quad (2.52)$$

Denklem 2.48'in basit şekli Denklem 2.53'teki gibi gösterilir [28].

$$\sum_{i=1}^{n_g} \left(\frac{\partial P_i}{\partial \lambda} \right)^{(k)} = \sum_{i=1}^{n_g} \frac{c_i + B_{ii} b_i}{2(c_i + \lambda^{(k)} B_{ii})^2} \quad (2.53)$$

3. SIRALI EN KÜÇÜK KARELER PROGRAMLAMA ALGORİTMASI

Kısa adı SLSQP (Sequential Least Squares Programming) olarak geçen Sıralı En küçük Kareler Programlama algoritmasının temel mantığı için Sıralı İkinci Dereceden programlama olan SQP'den (Sequential quadratic programming) yola çıkmamız gerekir. SQP temel prensibi, doğrusal olmayan optimizasyon probleminin gradyanını ve kısıtlamasını kullanarak ikinci dereceden bir programlama olan QP'nin (Kuadratik Programing) alt problemi oluşturmak ve bir dizi QP alt problemini kademeli olarak çözerek amaç fonksiyonunun minimumunu elde etmektir [30].

EYD problemi genellikle kuadratik maliyet fonksiyonlarıyla tanımlanır. Üretim ünitelerinin minimum-maksimum güç çıkışı kısıtları, yakıt kısıtları ve talep dengesi gibi çeşitli kısıtları içerir [31].

Üretim ünitelerinin sayısı arttıkça, çözüm uzayının büyümesi ve kısıtların doğrusal olmayan yapısı nedeniyle EYD problemini çözmek giderek daha karmaşık hale gelir. Bu da yüksek hesaplama gücü, uzun süreler ve etkin bir optimizasyon algoritması gerektirir [32].

SLSQP, kuadratik programlama (QP) prensibini ardışık şekilde uygulayarak, doğrusal olmayan kısıtları ve karmaşık maliyet fonksiyonlarını başarıyla yönetir [33]. SLSQP, EYD'deki toplam güç talebi ve ünite sınırlarının kısıtlarını doğrudan dikkate alır ve iterasyonlar boyunca kısıtları sürekli izler, bu da SLSQP algoritmasının kararlı şekilde yakınsamasını sağlar [34].

EYD probleminin türev bilgisi alınabilir maliyet fonksiyonları, SLSQP'nin gradyan tabanlı yöntemleriyle uyum içinde çalışarak hızlı yakınsamayı mümkün kılar [35].

Hızlı yakınsama kısa sürelerde maksimum verim elde etmeyi mümkün kılmaktadır. Ülkemizde offline olarak yapılan dağıtım planlamasının online olarak yapılmasını mümkün kılacak süreleri SLSQP algoritması ile örnek sistemler üzerinde elde edilebileceği gösterilmiştir.

3.1 SLSQP'nin çalışma prensibi

SLPSQP'nin çalışma mantığı için SQP'den hareket edersek, SQP lagrange fonsiyonun yaklaşık Hessian matrisini kullanır. SLSQP ise kademeli olarak denklemler çözülürken B matrisini yaklaşık olarak almaz B matrisini tam olarak ayrıştırıp denklemleri çözerek daha kararlı bir yakınsama elde eder [30].

Doğrusal olmayan kısıtlı optimizasyon problemi şu şekilde formüle edilebilir [30].

$$\begin{aligned} &\text{minimize } F(x) \\ &\text{subject to } c_i(x) = 0 \quad i = 1, 2, \dots, m_e \\ & \quad \quad c_i(x) \geq 0 \quad i = m_e + 1, \dots, m \end{aligned} \quad (3.1)$$

Burada $c_i(x)$ kısıtlamalarıdır. m_e ve m sırasıyla eşitlik ve eşitsizlik kısıtlamalarının miktarını temsil eder. Denklem 3.1'in optimizasyon problemi aşağıdaki QP alt problemlerine dönüştürülebilir [30].

$$\begin{aligned} &\min \frac{1}{2} d^{kT} B^k d^k + g^T d^k \\ &s.t. \quad c_i + \nabla c_i^T d^k = 0 \quad i = 1, 2, \dots, m_e \\ & \quad \quad c_i + \nabla c_i^T d^k \geq 0 \quad i = m_e + 1, \dots, m \end{aligned} \quad (3.2)$$

Burada $g = \nabla F(x)$, $c_i = c_i(x)$ ve B^k aşağıdaki gibi Lagrange fonsiyonun yaklaşık Hessian matrisidir [30].

$$L(x, \lambda) = F(x) - \sum_{i=1}^m \lambda_i c_i(x) \quad (3.3)$$

Burada λ Lagrange çarpanıdır.

QP alt problemi d^k çözümü ile x parametreleri $x^{k+1} = x^k + d^k$ olarak güncellenebilir. B^k , Hessian matrisi $\nabla_x^2 L(x^k, \lambda^k)$ 'nın iyi bir yaklaşımı ise, algoritma süper doğrusal yakınsama olacaktır. B^k , aşağıdaki şemayı kullanarak B matrisini güncelleyerek güncelleme işlemi sırasında yaklaşık bir Hessian matrisi olarak korunur [30].

$$B^{k+1} = B^k + (\tilde{y}^k \tilde{y}^{kT}) / (\tilde{y}^{kT} s^k) - (B^k s^k s^{kT} B^k) / (s^{kT} B^k s^k) \quad (3.4)$$

Burada $\tilde{y}^k = \theta y^k + (1 - \theta) B^k s^k$, $s^k = x^{k+1} - x^k$, θ ve y^k şu şekilde tanımlanır [30].

$$\theta = \begin{cases} 1 & s^{kT} y^k \geq \beta s^{kT} B^k s^k \\ (1 - \beta) s^{kT} B^k s^k / [s^{kT} B^k s^k - s^{kT} y^k] & s^{kT} y^k < \beta s^{kT} B^k s^k \end{cases} \quad (3.5)$$

ve

$$y^k = \nabla_x L(x^{k+1}, \lambda^{k+1}) - \nabla_x L(x^k, \lambda^{k+1}) \quad (3.6)$$

Yukarıda açıklanan algoritma, yalnızca yerel olarak süper doğrusal bir yakınsamaya yol açar. Global yakınsamayı sağlamak için, algoritmaya bir penalty fonksiyonu $F_r(x)$ eklenir [30].

$$F_r(x) = F(x) + r \left\{ \sum_{i=1}^{m_e} \max[0, c_i(x)] + \sum_{i=1}^{m_e} |c_i(x)| \right\} \quad (3.7)$$

Burada r penalty faktörüdür ve aşağıdaki şema kullanılarak güncellenir [30].

$$r = \begin{cases} \max(|\lambda_i^{k+1}| + \rho) & r < \max(|\lambda_i^{k+1}|) \\ r & else \end{cases} \quad (3.8)$$

Burada ρ bir pozitif sabittir.

$F_r(x)$ 'in minimumu, r yeterince büyük olduğunda Denklem 1'deki amaç fonksiyonu $F(x)$ 'e eşittir. Bu arada, x ters çözümü $x^k = x^k + a^k d^k$ olarak güncellenir. Adım boyutu a^k , aşağıdaki formül kullanılarak aranabilir [30].

$$F_r(x^{k+1}) < F_r(x^k) + \sigma a^k \bar{F}_r[(x^k, d^k) - (F_r x^k)] \quad (3.9)$$

Burada $0 < \sigma < 1$ ve $\bar{F}_r$ şu şekilde tanımlanır [30].

$$\bar{F}_r(x^k, d^k) = f(x^k) + g^{kT} d + \frac{1}{2} d^T B^k d + r \left[\sum_{i=1}^{m_e} |c_i^k + \nabla c_i^k d| + \sum_{i=m_e+1}^m |\min(0, c_i x^k + \nabla c_i x^{kT} d)| \right] \quad (3.10)$$

Süper lineer yakınsamanın kaybolmasına yol açacak olan Maratos etkisi, QP alt problemi aşağıdaki gibi değiştirilerek önlenir [30].

$$\begin{aligned} \min & \frac{1}{2} \hat{d}^{kT} B^k \hat{d}^k + p^{kT} \hat{d}^k \\ \text{s.t.} & \quad c_i + h_i^{kT} \hat{d}^k = 0 \\ & \quad c_i + h_i^{kT} \hat{d}^k \geq 0 \end{aligned} \quad (3.11)$$

Burada p ve h_i aşağıdaki denklemler ile verilir.

$$p^k = g^k + \frac{1}{2} \sum_{i=1}^m \lambda_i^{k+1} [\nabla c_i(x^k + \hat{d}^k) - \nabla c_i(x^k)] \quad (3.12)$$

$$h_i^k = \nabla c_i^k + \frac{1}{2} \hat{d}^k [\nabla c_i(x^k + \hat{d}^k) - \nabla c_i(x^k)] \quad (3.13)$$

$x + \hat{d}$ 'nin Denklem 3.1 ile ikinci derecedeki kısıtlamalarını karşılamasını sağlamak için, tek boyutlu arama aşağıdaki arama kriteri boyunca olmalıdır [30].

$$F_r \left[x^k + a^k d^k + (a^k)^2 (\hat{d}^k - d^k) \right] < F_r(x^k) + \sigma a^k \left[\bar{F}_r(x^k, d^k) - F_r(x^k) \right] \quad (3.14)$$

Denklem 3.1'in yalnızca Maratos etkisi oluştuğunda çözülmesi gerektiğine dikkat edilmelidir. SQP algoritmasının uygulanabilirliğini artırmak için Denklem 3.11, yalnızca Denklem 3.15 ve denklem 3.16 aynı anda karşılandığında çözülür [30].

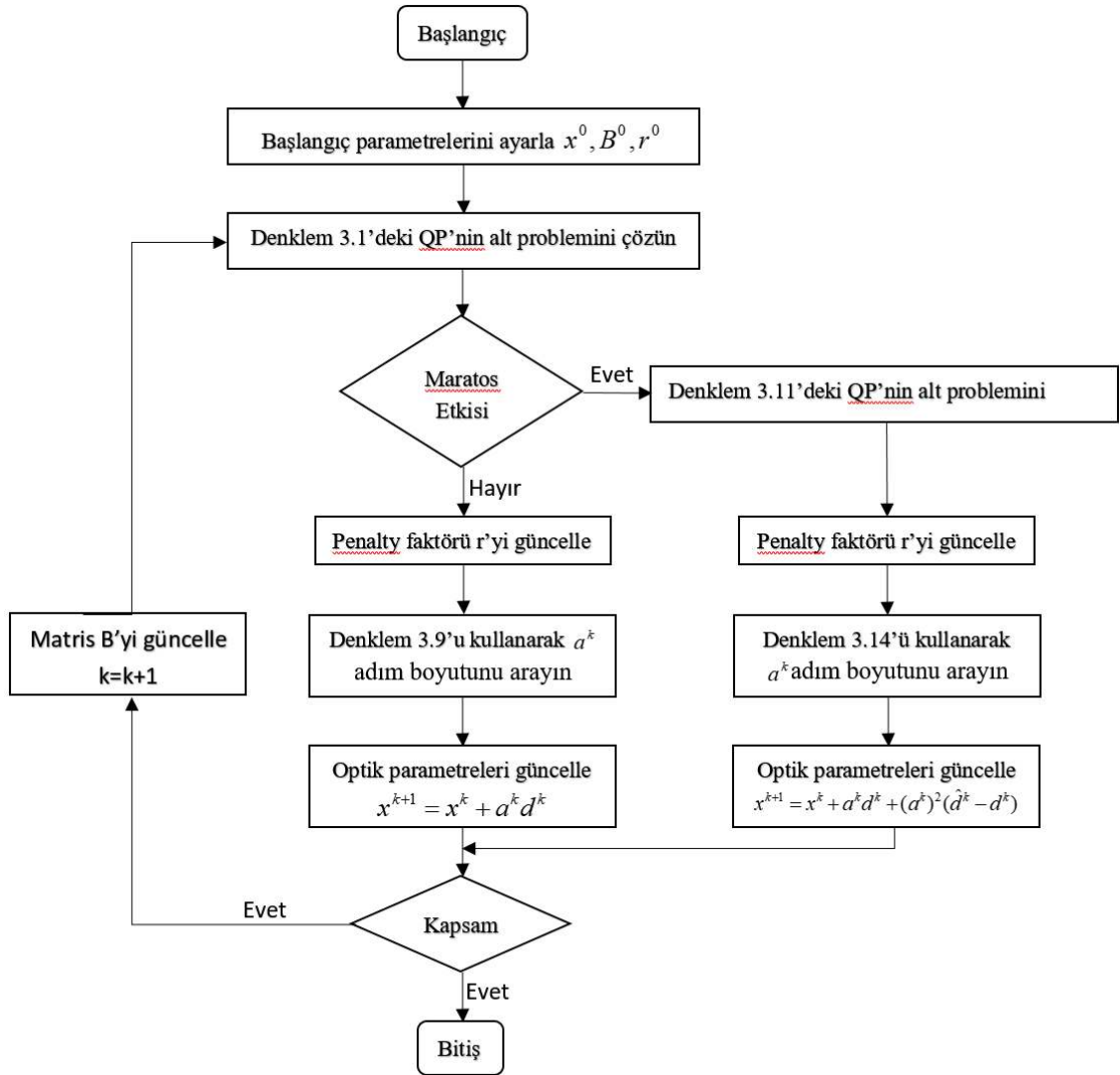
$$F_r(x^k + a^k d^k) \geq F_r(x^k) + \sigma a^k \bar{F}_r \left[(x^k, d^k) - F_r(x^k) \right] \quad (3.15)$$

$$\max \left[|c_i x^k + h_i^{kT} d^k - c_i(x^k + d^k)| \right] < \sigma \max \left[c_i x^k + \nabla c_i x^{kT} d^k - c_i(x^k + d^k) \right] \quad (3.16)$$

SQP algoritmasının yakınsaması Denklem 3.17'de aşağıdaki gibi verilir [30].

$$\left[F_r(x^k) - F_r(x^{k+1}) \right] / \left(\sigma + |F_r(x^{k+1})| \right) \leq \varepsilon \quad (3.17)$$

Bu çalışmada EYD problemi için çözüm önerisi olarak SLSQP algoritmasının akış şemasına ulaşmak istersek SQP'nin akış şemasından yola çıkmamız gerekmektedir. Akış diyagramı Şekil 3.1'de görülmektedir.



Şekil 3.1: SQP algoritmasının akış şeması [30].

Yapılan çalışmada Python programlama dili ve scipy kütüphanesinden faydalanarak SLSQP algoritmasını EYD problemlerini çözmek için kullandık. EYD problemlerinde çözüm süreleri, işlemci süreleri problemi çözmek için kullanılan sistemin özelliklerine göre değişiklik gösterebilir. Çözüm algoritmasının Python kodlarına [36]'dan ulaşılabilir. Python'daki sonuçlar ve hesaplama süreleri, i9 14900hx ve 32 gb ddr5 ram donanımına sahip bilgisayar ile elde edilmiştir.

4. GENETİK ALGORİTMA

Genetik Algoritmalar (GA), doğal seçim ve genetikten esinlenen evrimsel algoritmaların bir alt kümesi olup, karmaşık optimizasyon problemlerini çözmede güçlü araçlar olarak kullanılmaktadır [37].

4.1 Genetik algoritmanın çalışma prensibi

GA'lar, aday çözümler popülasyonunu ardışık olarak evrimleştirerek optimal çözüme giderek daha iyi yakınsamalar elde etmeyi amaçlar [38].

Tipik bir GA'da, her aday çözüm ikili (binary) dizi olarak kodlanır. Örneğin, bir aday şu şekilde temsil edilebilir:

10101011010110101100101011010110

Burada, her bit problem alanındaki bir karar değişkeni veya parametreyi ifade eder [37].

Algoritma, her adayın uygunluğunu belirli bir amaç fonksiyonu aracılığıyla değerlendirir, daha yüksek uygunluk değerine sahip bireylerin üreme şansı artar.

Üreme aşaması, iki temel genetik operatörü içerir: çaprazlama ve mutasyon. Çaprazlama aşamasında, iki ebeveynin ikili dizilerinden belirli segmentler değiştirilerek yeni yavru bireyler üretilir.

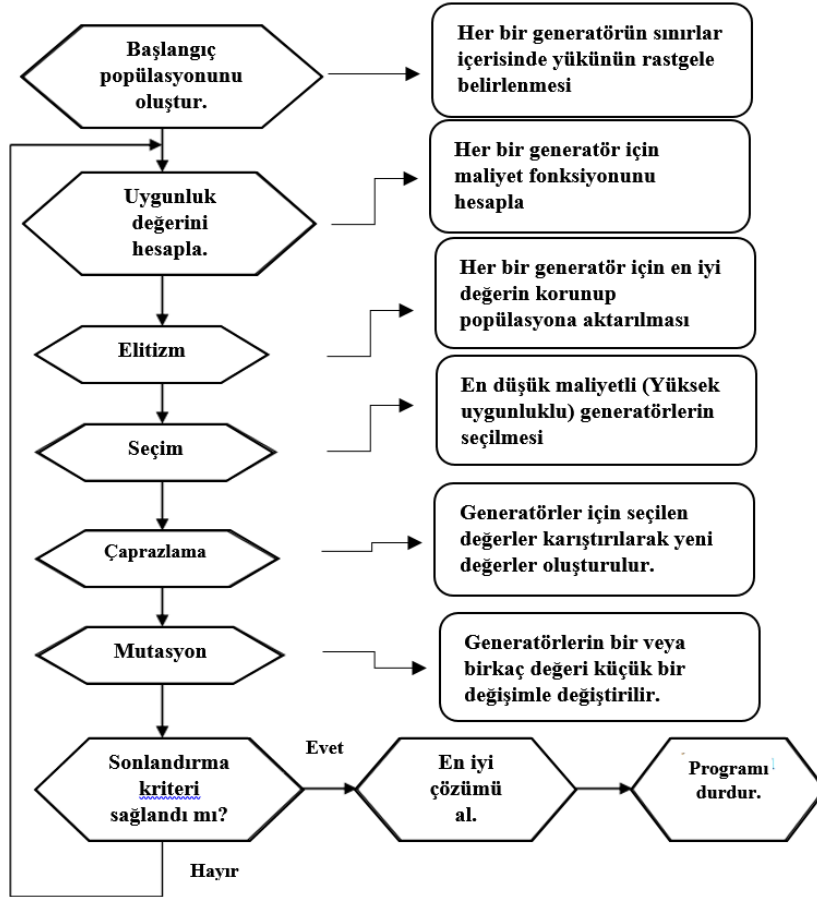
Örneğin:

Ebeveyn 1: 10101011010110101100101011010110

Ebeveyn 2: 11001100101010111100011010101011

çaprazlama işlemi sonucunda, her iki ebeveynin özelliklerini harmanlayarak örneğin 10101011001010111100011010101011 gibi bir yavru elde edilebilir [38]. Mutasyon ise, rastgele değişiklikler yaparak örneğin, belirli bir bitin değerini tersine çevirerek (0'ı 1'e veya 1'i 0'a) genetik çeşitliliği korur ve erken optimal olmayan çözümlere yakınsama riskini azaltır [38].

Ardışık nesiller boyunca GA, popülasyondaki çözümleri iyileştirir ve genellikle belirli bir maksimum nesil sayısına ulaşılan veya tatmin edici bir uygunluk seviyesi elde edilene kadar işlem devam eder [37].



Şekil 4.1 : Genetik Algoritma Akış Şeması.

5. YAPAY ARI KOLONİSİ ALGORİTMASI

Yapay Arı Kolonisi (YAK) algoritması, doğada arıların kolektif besin arama davranışlarından esinlenerek geliştirilmiş bir meta-sezgisel optimizasyon yöntemidir. 2005 yılında Derviş Karaboğa tarafından önerilen bu algoritma, özellikle karmaşık ve doğrusal olmayan problemlerde etkili sonuçlar sunar [40].

Algoritma, arıların yiyecek kaynaklarını keşfetme, paylaşma ve sürdürme stratejilerini matematiksel bir modelle taklit eder [41].

5.1 Yapay arı kolonisi algoritmanın temelleri

Reaktif difüzyon denklemleri temel alınarak geliştirilen ve Tereshko tarafından öne sürülen, ortak zekanın ortaya çıkmasına imkan veren arama modelinde; besin kaynakları, aktif görev yapan işçi arılar ve görev dışı kalan işçi arılar olmak üzere üç ana unsur yer almaktadır. [41].

1. **Yiyecek kaynakları:** Üretim maliyetini temsil eden ve optimizasyon uzayındaki potansiyel çözümleri ifade eden noktalar. Kaynağın kalitesi, nektar miktarı (maliyet fonksiyonu değeri) ve kovana yakınlık (kısıt uyumluluğu) ile ilişkilidir [41].
2. **Görevli işçi arılar:** Önceden keşfedilen kaynakları işleyen ve bu kaynaklarla ilgili bilgiyi dans yoluyla paylaşan arılar.
3. **Görevsiz işçi arılar:**
 - **Kaşif arılar:** Rastgele yeni kaynaklar keşfeder.
 - **Gözcü arılar:** Görevli arıların verdiği bilgiyi kullanarak en umut verici kaynaklara yönelir [42,43].

Arılar arası iletişim, kovanda gerçekleşen üç tür dans ile sağlanır:

- **Daire dansı:** Yakın mesafeli (<100 m) kaynaklar için yön bilgisi içermez.
- **Titreme dansı:** Kaynak kapasitesinin aşıldığını bildirir.
- **Kuyruk dansı:** Uzak kaynakların yön ve mesafe bilgisini 8 rakamı şeklindeki hareketlerle aktarır [43].

Adım 1: Başlangıç hazırlıkları – Kovandaki plan

Arılar ve kaynaklar:

Arıların keşfedeceği "yiyecek kaynakları" (çözüm adayları), rastgele belirlenir. Her kaynak, bir üreticinin güç çıkışı gibi bir parametreyi temsil eder [40].

Adım 2: Kaynakların kalite kontrolü – Nektar hesaplama

Çiçeklerin değerlendirilmesi:

Her kaynağın "nektar değeri" (maliyet fonksiyonu) hesaplanır. Örneğin, bir üreticinin güç çıkışı ne kadar ekonomikse, nektar o kadar kalitelidir. [40,44].

Adım 3: İşçi arıların keşfi – Yakın çevrede arama

Yerel iyileştirme:

İşçi arılar, mevcut kaynakların etrafında küçük değişiklikler yaparak yeni çözümler arar. Yeni çözüm daha iyiye, kovan hafızasına kaydedilir [40,44].

Adım 4: Gözcü arıların seçimi – Dansla bilgi paylaşımı

Dans ile bilgi:

Gözcü arılar, işçi arıların "danslarından" (uygunluk değerleri) etkilenir. Daha kaliteli kaynaklara daha fazla arı gönderilir [40].

Adım 5: Kaşif arılar

Yenilikçi çözümler:

Bir kaynak belirli bir süre (limit) iyileşmezse, kaşif arılar tamamen yeni bir bölge keşfeder. Bu, yerel optimumda takılmayı önler [40,45].

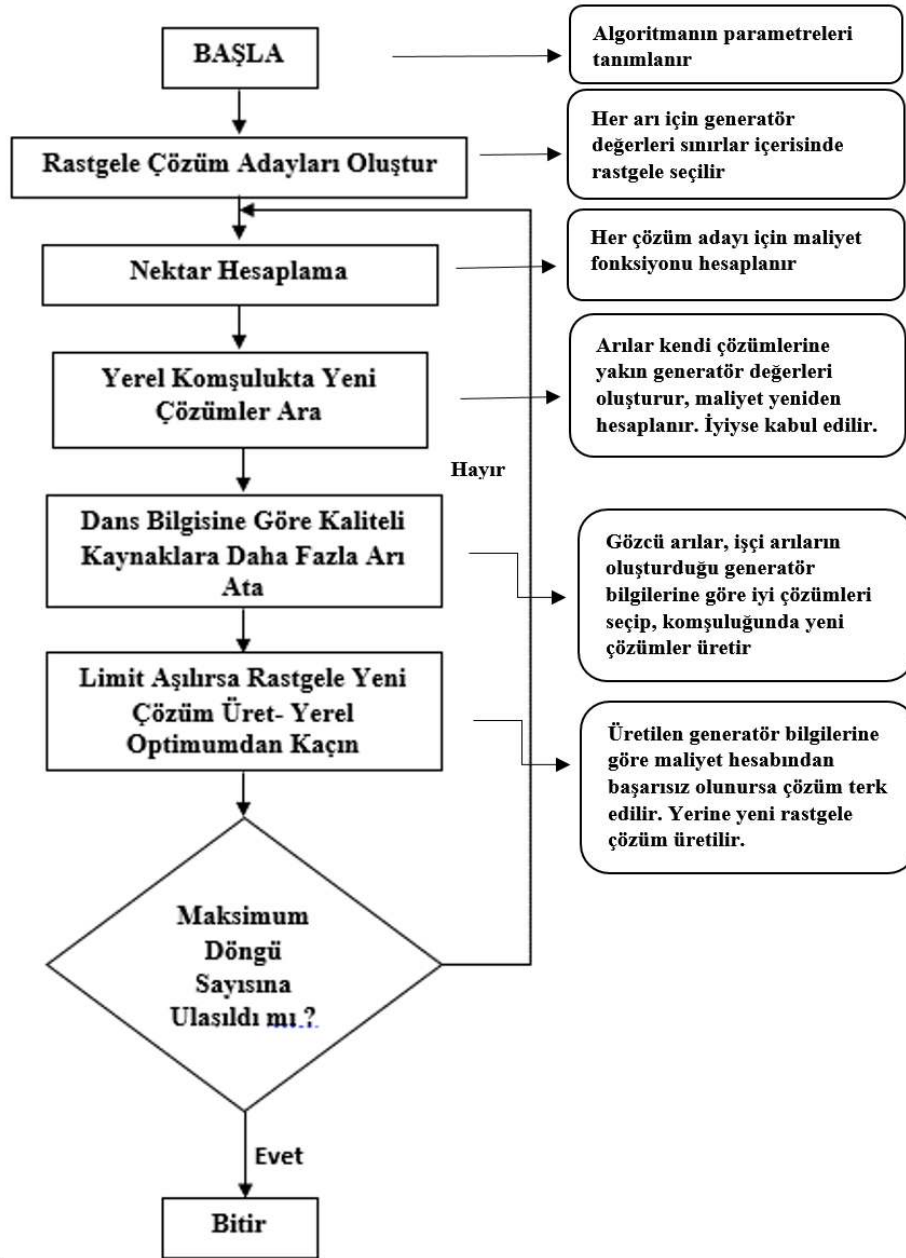
Adım 6: Sonuç – En iyi çözümün ilanı

Kovanın kararı:

Algoritma, belirlenen maksimum döngü sayısına ulaştığında durur. En düşük maliyetli çözüm (en bol nektarlı çiçek) sonuç olarak ilan edilir [44].

Adım	Doğadaki karşılığı	Algoritmadaki rolü	Kaynak
Başlangıç	Çiçek tarlasını keşfetme	Rastgele çözüm üretme	[40,44]
Kalite Kontrol	Nektar ölçümü	Maliyet fonksiyonu hesaplama	[40]
İşçi Arılar	Yakın çiçekleri tarama	Yerel arama ve iyileştirme	[40]
Gözcü Arılar	Dansla bilgi paylaşımı	Olasılıklı seçim ve global arama	[40,44]
Kâşif Arılar	Yeni bölge keşfi	Yerel optimumdan kaçınma	[40,45]
Sonuç	En iyi çiçeğin seçilmesi	En iyi çözümün raporlanması	[44]

Şekil 5.1 : YAK algoritmasının rolleri ve Doğadaki karşılığı



Şekil 5.2 : YAK algoritması akış şeması.

6. TABU ARAMA ALGORİTMASI

Tabu Arama (TA), yerel optimumlardan kaçınmak için geçmiş hareketleri "yasaklayan" (tabu listesi) ve uzun vadeli bellek stratejileri kullanan bir meta-sezgisel optimizasyon algoritmasıdır. Doğadaki keşifçi davranışlardan esinlenerek, Glover tarafından 1986'da geliştirilmiştir [47].

TA algoritmasında, tabu listesi ile oluşturulan ilk aday çözümler ve değişken komşu çözümlerin sayısı, bir çeşit engelleme işlevi görür. TA sürecinde, en iyi komşu tasarımların bulunmasını sağlayan hareketler yapay bir bellekte tutulur. Bu bellek, kısa süreli bellek olarak adlandırılır. Bu yapay bellek aracılığıyla, arama sırasında daha önceki tasarımların tekrar elde edilmesini sağlayacak hareketler engellenir [48].

6.1 Tabu arama algoritmasının çalışma prensibi

Adım 1 Başlangıç :

- **Rastgele bir çözüm seç:**

Bilinmeyen rastgele bir başlangıç noktası belirlenir. Örneğin Üretici güç çıkışlarının rastgele atanır [47].

Adım 2 Komşuluk arama :

- **Küçük değişiklikler :**

Mevcut çözümün yakınında yeni adaylar üretilir örneğin, bir üreticinin gücü $\pm\Delta P$ kadar değiştirilir [49].

Adım 3 Tabu listesi :

- **Yasaklı hareketleri kaydet :**

Son 5-10 iterasyonda yapılan değişiklikler listelenir ve tekrarlanması engellenir [47,50].

Adım 4 Aspirasyon kriteri :

- **İstisnalara izin vermek :**

Tabu listesindeki bir hareket, gerçekten en iyi çözümünü bulursa kabul edilir [51].

Adım 5 Uzun vadeli bellek :

- **Bellek stratejisi :**

Seyrek ziyaret edilen çözüm bölgelerine öncelik verilerek çeşitlilik sağlanır [52].

Adım 6 Durdurma :

- **Maksimum iterasyon veya çözüm kalitesi :**

Algoritma, belirlenen şart sağlandığında durur ve en iyi çözümü raporlar [52].



Şekil 6.1 : Tabu Arama Algoritması Akış Şeması.

7. TAVLAMA BENZETİMİ ALGORİTMASI

Tavlama benzetimi (TB) algoritması; karmaşık optimizasyon problemlerinde küresel optimum değerine ulaşmak amacıyla kullanılan meta-sezgisel bir yöntemdir. Bu yöntem, metalleri yüksek sıcaklıkta ısıtıp sonrasında kademeli olarak soğutarak daha kararlı ve homojen hale getirme prensibinden esinlenmiştir [53].

Algoritma, başlangıçta geniş arama alanı sağlamanın yanında, soğuma süreciyle arama stratejisini daraltarak kötü çözümlerin de belirli bir olasılıkla kabul edilmesi sayesinde yerel minimumlardan kaçınmayı mümkün kılar [53].

7.1 Tavlama benzetimi algoritmasının çalışma prensibi

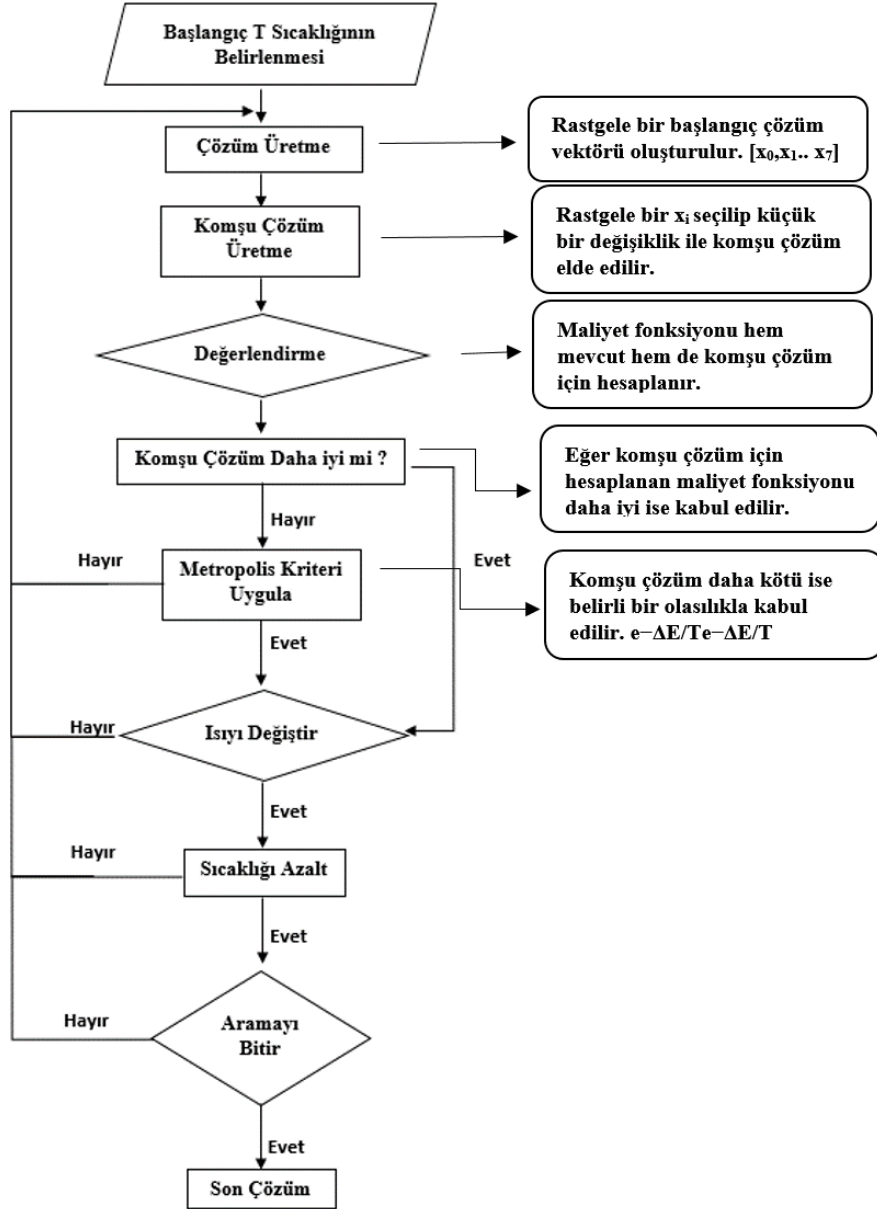
TB, Kirkpatrick ve arkadaşları tarafından 1983'te önerilmiştir. Temel prensibi, metallerin kontrollü soğutularak enerji minimizasyonu sağlanmasıdır [53].

Algoritmanın adımları şunlardır:

- 1. Başlangıç sıcaklığı (T) ve sıcaklık ataması :** Algoritma, rastgele ya da belirlenmiş bir yöntemle elde edilen bir başlangıç çözümü ve yüksek bir başlangıç sıcaklığı (T) ile çalışmaya başlar. Bu, geniş bir çözüm alanının keşfedilebilmesi için gereklidir. Yüksek sıcaklıkta kötü çözümlerin kabul olasılığı yüksektir [53].
- 2. Komşu çözüm üretme :** Mevcut çözüm küçük pertürbasyonlarla değiştirilir. Yeni komşu bir çözüm üretilir. Bu adım, çözüm uzayındaki lokal varyasyonları keşfetmek için temel bir mekanizmadır [53].
- 3. Değerlendirme ve metropolis kriteri :** Eğer üretilen yeni çözüm, mevcut çözümden daha iyi (daha düşük maliyet/enerji) ise doğrudan kabul edilir. Eğer yeni çözüm daha kötü ise, aşağıdaki Metropolis kriteri uygulanır :

Burada, ΔE yeni ve mevcut çözüm arasındaki enerji farkıdır. Yüksek sıcaklıkta, daha kötü çözümlerin de belli bir olasılıkla maliyeti azaltıyorsa kabul edilir. Maliyet artıyorsa, $e^{-\Delta E/T}$ olasılığıyla kabul edilir. Bu da algoritmanın yerel minimuma takılmasını engeller [53].
- 4. Soğuma programı :** Sıcaklık iterasyonla azaltılarak arama daraltılır. Sıcaklığın düşmesiyle birlikte, algoritma daha az "riskli" adımlar atar ve daha yerel aramaya yönelir. (örneğin, $T_{k+1} = \alpha \cdot T_k$, $\alpha = 0.85 - 0.99$) [53].

5. Durduma kriteri : Algoritma, belirlenen durdurma kriterlerinden biri gerçekleştiğinde sonlanır; bu kriterler arasında sıcaklığın belirli bir eşik değerin altına düşmesi veya önceden tanımlanmış iterasyon sayısına ulaşılması yer alır [53].



Şekil 7.1: Tavlama Benzetimi Algoritması Akış Şeması

8. UYGULAMA VE DEĞERLENDİRME

Bu tez kapsamında, ekonomik yük dağıtım problemiyle ilgili çözüm, belirli güç sistemleri üzerinde incelenmiştir. İlk olarak, Türkiye’de yer alan 380 kV ve 22 baraya sahip ve 8 generatörden oluşan bağlantılı bir güç sistemi üzerinde EYD sorunu ele alınmıştır. Ardından, Türkiye’de yer alan 380kV ve 6 generatörden oluşan 14 baralı bir test sistemi üzerinde benzer çalışmalar gerçekleştirilmiştir. Her iki problemin çözümü sırasında hat kayıpları göz ardı edilmiştir.

Bu tez çalışmasındaki bütün hesaplamalar, Python dilinde yazılan kodlar ile i9 14900hx işlemci ve 32gb ddr5 ram donanımına sahip bir bilgisayarda gerçekleştirilmiştir. Tezdeki algoritmaların python kodları [36]’ da verilmiştir.

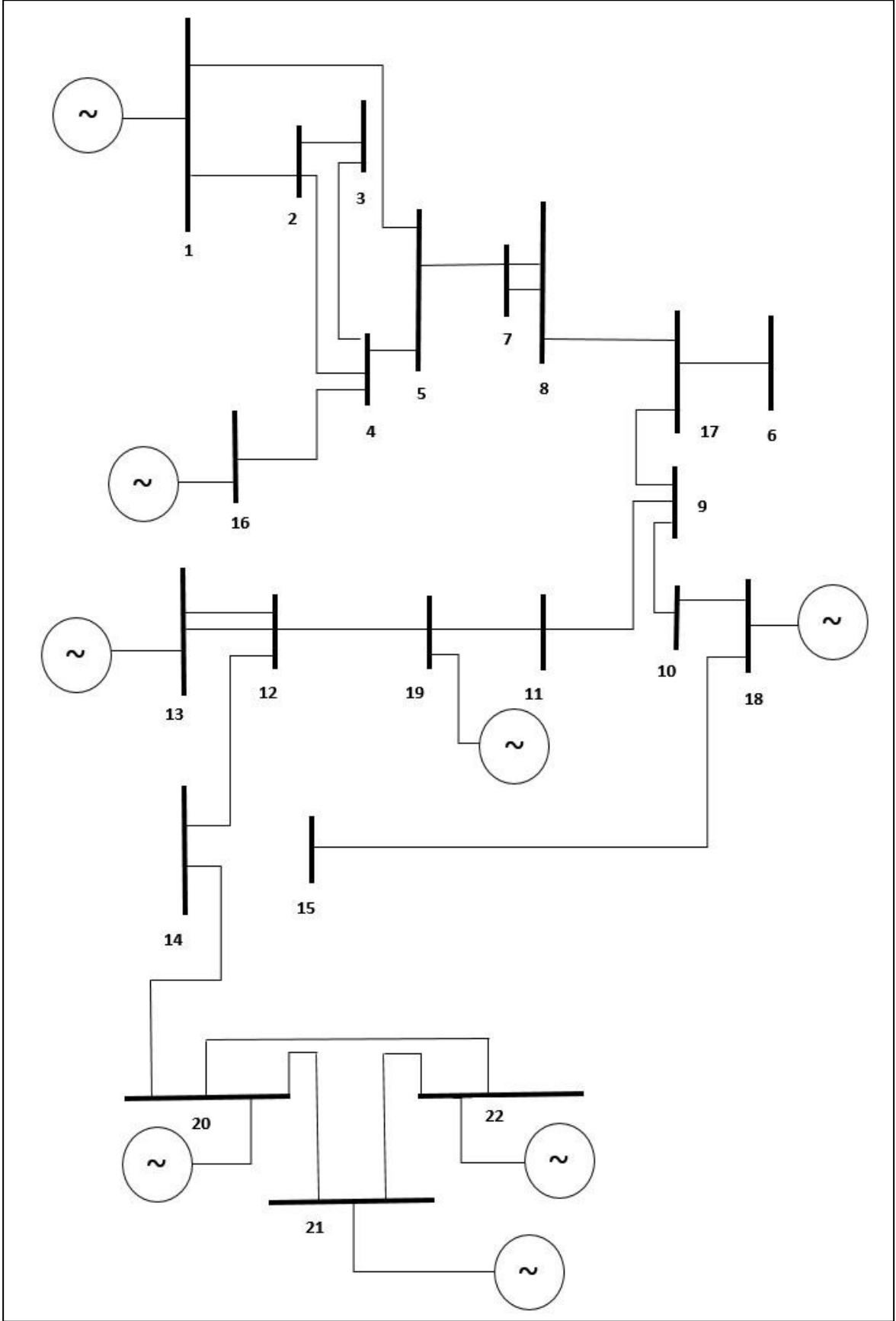
8.1 EYD probleminin uygulanacağı güç sistemlerinin bilgileri

8.1.1 8 Yüksek gerilim ve 14 yük barası olmak üzere 22 baradan oluşan sistem

Türkiye’deki bu güç sistemi, toplamda 22 baradan oluşan sistem, 8 yüksek gerilim barası ve 14 yük barasından meydana gelmektedir. Sistem, Bursa Doğalgaz, SomaB, Hamitabat, Seyitömer, Ambarlı, Yeniköy, Kemerköy ve Yatağan termik santrallerinin birbirine entegre şekilde bağlanmasıyla oluşturulmuş. Sistemde, 4000 MW’lık bir yük için EYD gerçekleştirilecektir. Sisteme ait baraların isim ve bara numaralarının yer aldığı Tablo 8.1 ve sistem bara şeması da Şekil 8.1’de yer almaktadır [1].

Bara Numarası	Bara Adı	Bara Numarası	Bara Adı
1	Hamitabat	12	Aliağa II
2	Unimar	13	İzmir DGKÇ
3	Habibler	14	Uzundere
4	İkitelli	15	Işıklar
5	Alibeyköy	16	Ambarlı
6	Karabiga	17	Bursa D.Gaz
7	Ümraniye	18	Seyit Ömer
8	T.Ören	19	Soma B
9	Bursa San	20	Yeniköy
10	T.Şalt	21	Kemerköy
11	Balıkesir II	22	Yatağan

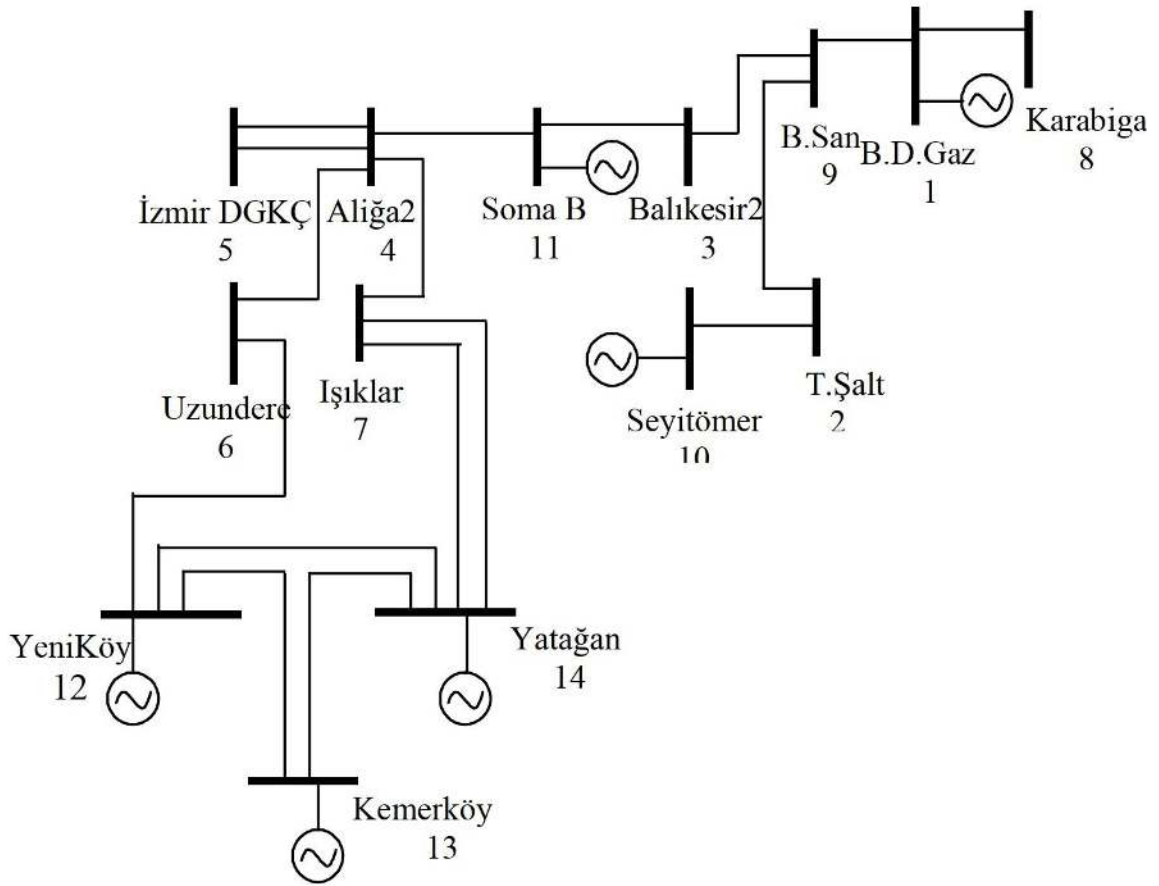
Tablo 8.1: 22 Baralı Sistemin Bara Numara ve Adları [1].



Şekil 8.1: 8 Yüksek gerilim ve 14 yük barası olmak üzere 22 baradan oluşan sistem [1].

8.1.2 6 Yüksek gerilim ve 8 yük barası olmak üzere 14 baradan oluşan sistem

Türkiye'deki bu güç sistemi, toplamda 14 baradan oluşan sistem, 6 yüksek gerilim barası ve 8 yük barası içermektedir. Sistem, Bursa DGZ, SomaB, Seyitömer, Yeniköy, Kemerköy ve Yatağan termik santrallerinin birbirine entegre şekilde bağlanmasıyla oluşturulmuş. Sistemde, 2734.9 MW'lık bir yük durumu için ekonomik dağıtım gerçekleştirilecektir. Sisteme ait baraların isim ve bara numaralarının yer aldığı sisteme ait bara şeması Şekil 8.2'de yer almaktadır [55].



Şekil 8.2: 380 kV, 6 generatörlü ve 14 baralı sistem [55].

8.2 SLSQP algoritması ile kayıpsız ekonomik yük dağıtım sonuçları

SLSQP algoritmasının uygulanacağı Şekil 8.1 ve Şekil 8.2’de yer alan güç sistemlerine ait maliyet fonksiyonları ve sınır değerleri aşağıda bulunan Tablo 8.2’de yer almaktadır.

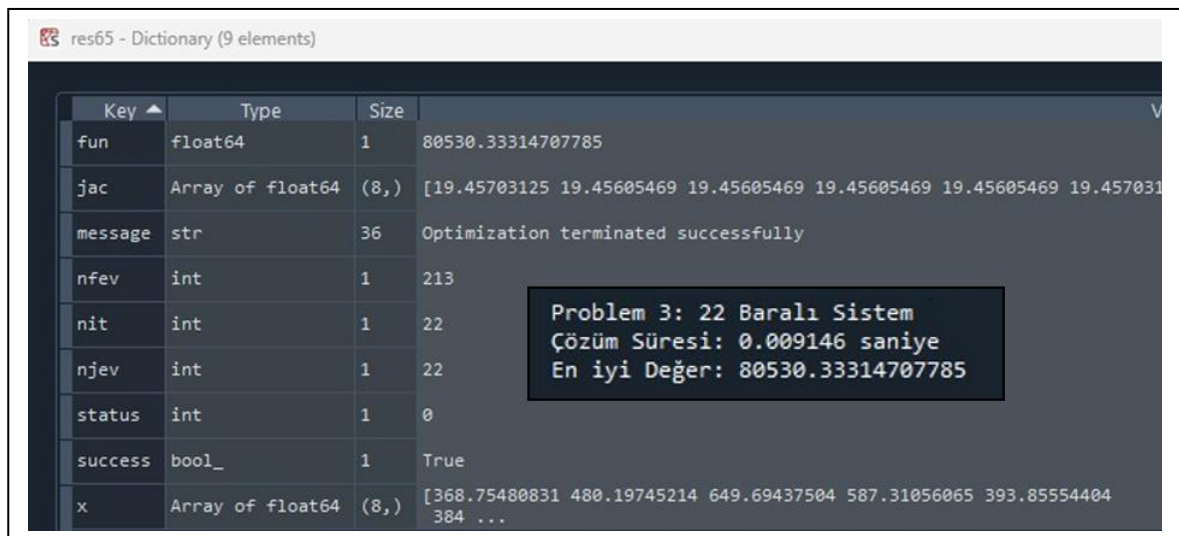
Santraller	$P_{i,min}(MW)$	$P_{i,max}(MW)$	a	b	c
Bursa D.gaz	318	1432	6780.5	5.682	0,0106
Seyitömer	150	600	1564.4	3.1288	0.0139
SomaB	210	990	5134.1	6.2232	0,0168
Yeniköy	110	420	1159.5	3.3128	0.021
Kemerköy	140	630	1697	3.2324	0.0137
Yatağan	140	630	1822.8	3.472	0.0147
Hamitabat	190	1120	6595.5	7.0663	0.0168
Ambarlı	245	1350	7290.6	7.2592	0.0127

Tablo 8.2: Generatör verileri [1].

Ekonomik yük dağıtım problemlerinin SLSQP ile çözümü için Python’da bulunan Scipy kütüphanesinden faydalanılmıştır.

Python’daki SciPy kütüphanesi, `scipy.optimize.minimize()` fonksiyonu SLSQP algoritmasını desteklemektedir [56].

Şekil 8.1’de yer alan 22 baralı güç sistemine ait verileri SLSQP algoritması ile çözdüğümüzde bize Şekil 8.3’te görünen sonuçları vermektedir.



Şekil 8.3: 8 generatör ve 22 baralı sistemin python programa dili ile elde SLSQP sonuçları

Şekil 8.2’de yer alan 14 baralı güç sistemine ait verileri SLSQP algoritması ile çözdüğümüzde bize Şekil 8.4’te görünen sonuçları vermektedir.

Key	Type	Size	Value
fun	float64	1	47660.88380750896
jac	Array of float64	(6,)	[17.44824219 17.44726562 17.44726562 17.44824219 17.44824219 17.448242 ...]
message	str	36	Optimization terminated successfully
nfev	int	1	145
nit	int	1	20
njev	int	1	20
status	int	1	0
success	bool_	1	True
x	Array of float64	(6,)	[555.00270889 515.0724106 334.07362919 336.55939739 518.81947281 475 ...]

Problem 3: 14 Baralı Sistem
Çözüm Süresi: 0.007069 saniye
En iyi Değer: 47660.88380750896

Şekil 8.4 : 6 generatör ve 14 baralı sistemin python programa dili ile elde SLSQP sonuçları

8.3 Genetik algoritma ile kayıpsız ekonomik yük dağıtım sonuçları

GA’nın uygulanacağı Şekil 8.1 ve Şekil 8.2’de yer alan güç sistemlerine ait maliyet fonksiyonları ve sınır değerleri Tablo 8.2’de yer almaktadır.

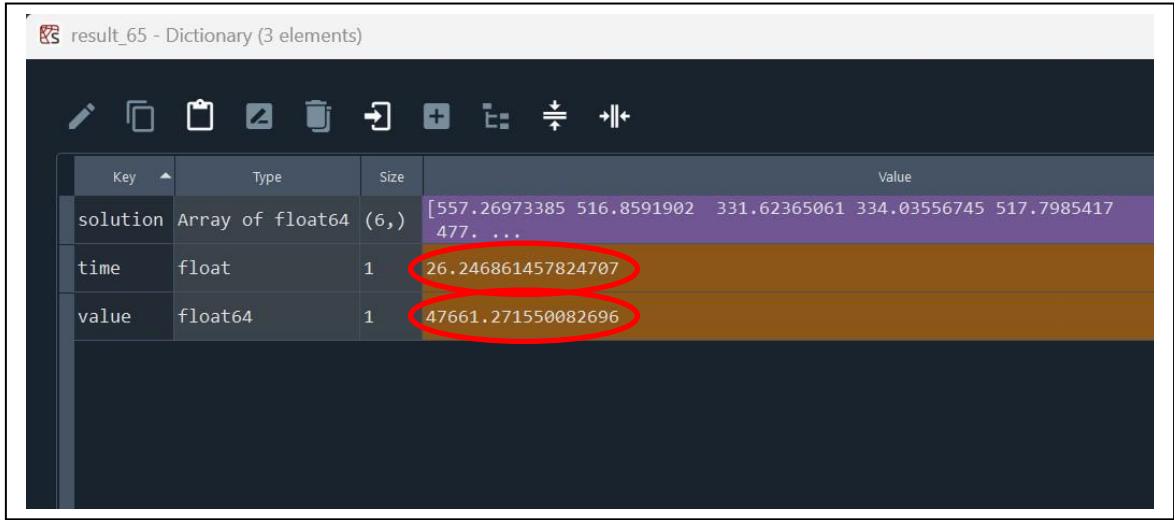
Şekil 8.1’de yer alan 22 baralı güç sistemine ait verileri GA ile çözdüğümüzde bize

Şekil 8.5’te görünen sonuçları vermektedir.

Key	Type	Size	Value
solution	Array of float64	(8,)	[366.58335338 479.37087451 651.50406418 587.30803907 394.13157475 387 ...]
time	float	1	112.90871405601501
value	float64	1	80530.72473229577

Şekil 8.5 : 8 generatör ve 22 baralı sistemin python programa dili ile elde GA sonuçları

Şekil 8.2’de yer alan 14 baralı güç sistemine ait verileri GA ile çözdüğümüzde bize Şekil 8.6’da görünen sonuçları vermektedir.



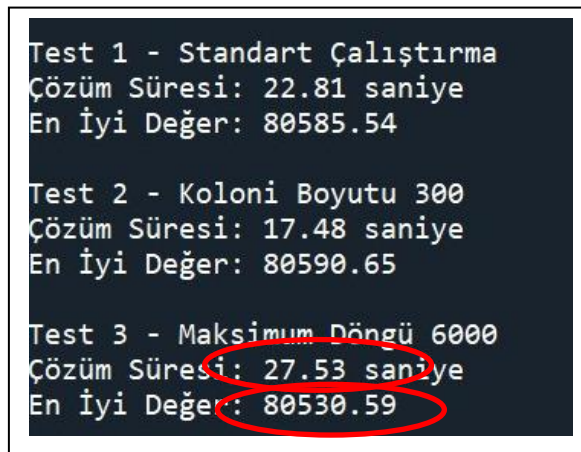
Key	Type	Size	Value
solution	Array of float64	(6,)	[557.26973385 516.8591902 331.62365061 334.03556745 517.7985417 477. ...]
time	float	1	26.246861457824707
value	float64	1	47661.271550082696

Şekil 8.6 : 6 generatör ve 14 baralı sistemin python programa dili ile elde GA sonuçları

8.4 Yapay arı kolonisi algoritması ile kayıpsız ekonomik yük dağıtım sonuçları

YAK’nın uygulanacağı Şekil 8.1’de yer alan güç sistemine ait maliyet fonksiyonları ve sınır değerleri Tablo 8.2’de yer almaktadır.

Şekil 8.1’de yer alan 22 baralı güç sistemine ait verileri YAK ile çözdüğümüzde bize Şekil 8.7’de görünen sonuçları vermektedir.



```
Test 1 - Standart Çalıştırma
Çözüm Süresi: 22.81 saniye
En İyi Değer: 80585.54

Test 2 - Koloni Boyutu 300
Çözüm Süresi: 17.48 saniye
En İyi Değer: 80590.65

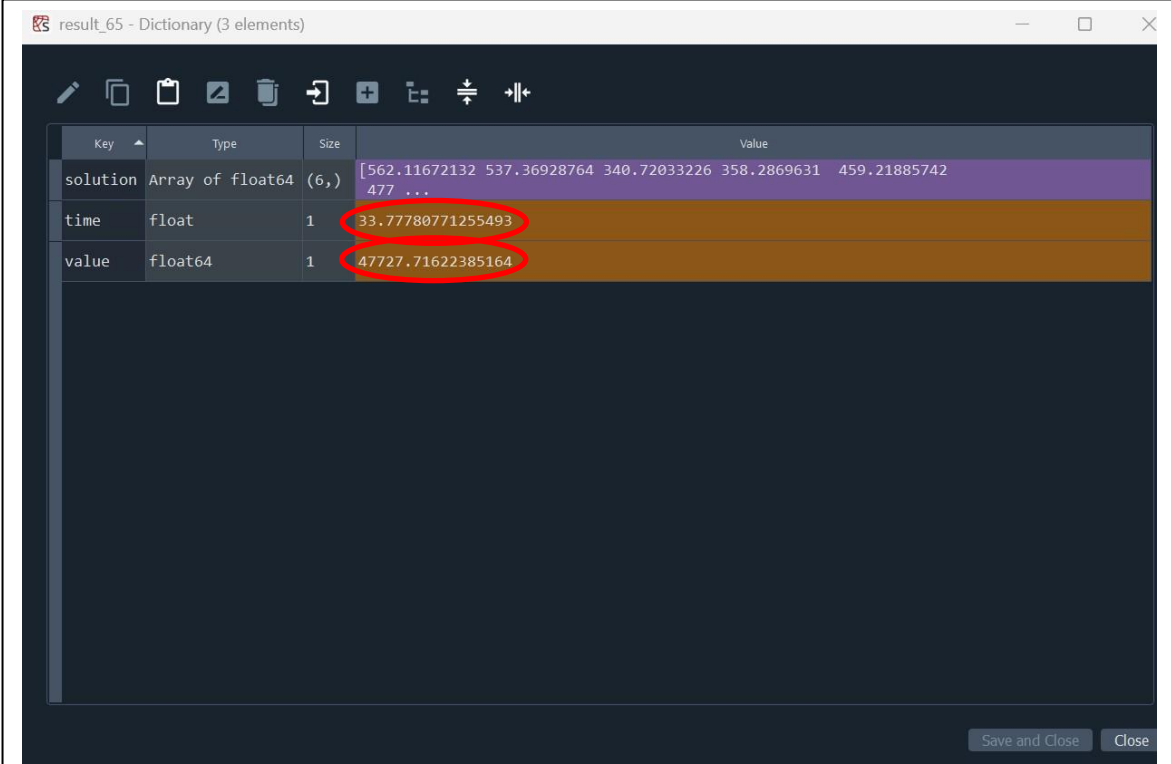
Test 3 - Maksimum Döngü 6000
Çözüm Süresi: 27.53 saniye
En İyi Değer: 80530.59
```

Şekil 8.7 : 8 generatör ve 22 baralı sistemin python programa dili ile elde YAK sonuçları

8.5 Tabu arama algoritması ile kayıpsız ekonomik yük dağıtım sonuçları

TA algoritmasının uygulanacağı Şekil 8.2’de yer alan güç sistemine ait maliyet fonksiyonları ve sınır değerleri Tablo 8.2’de yer almaktadır.

Şekil 8.2’de yer alan 14 baralı güç sistemine ait verileri TA ile çözdüğümüzde bize Şekil 8.8’de görünen sonuçları vermektedir.



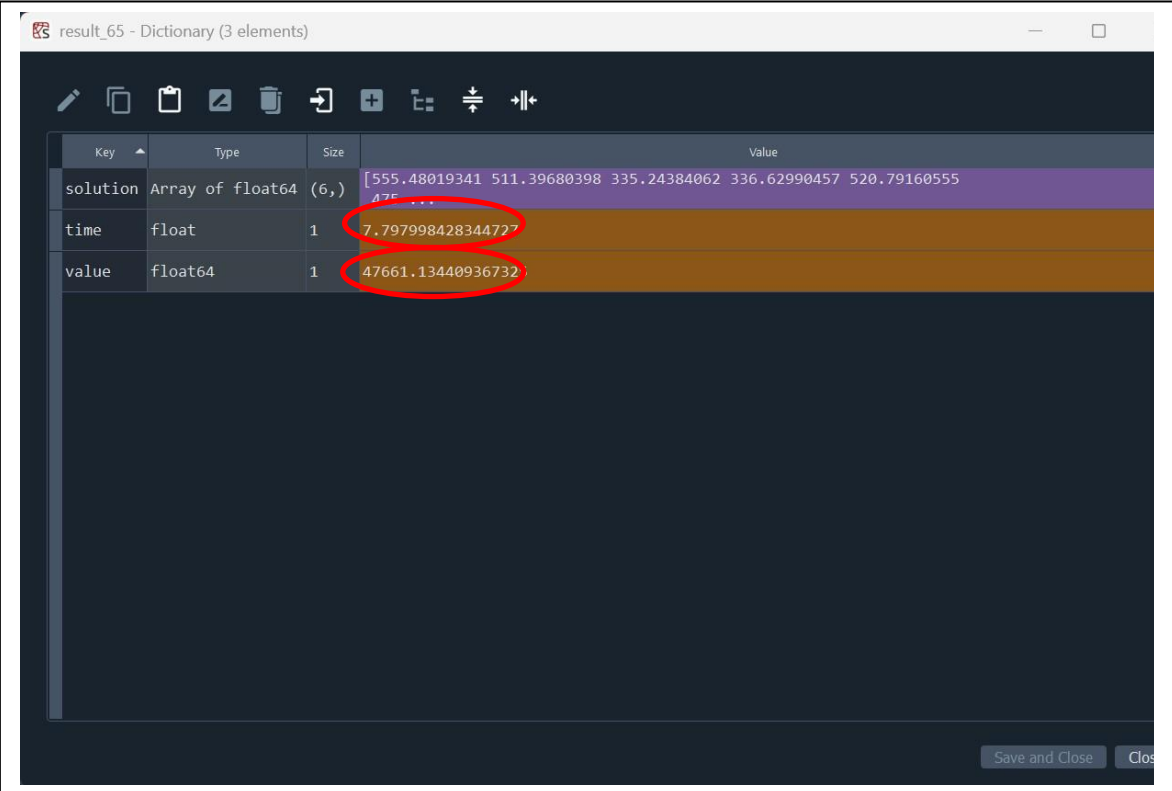
Key	Type	Size	Value
solution	Array of float64	(6,)	[562.11672132 537.36928764 340.72033226 358.2869631 459.21885742 477 ...]
time	float	1	33.77780771255493
value	float64	1	47727.71622385164

Şekil 8.8 : 6 generatör ve 14 baralı sistemin python programa dili ile elde TA sonuçları

8.6 Tavlama benzetimi algoritması ile kayıpsız ekonomik yük dağıtım sonuçları

TB algoritmasının uygulanacağı Şekil 8.2’de yer alan güç sistemlerine ait maliyet fonksiyonları ve sınır değerleri Tablo 8.2’de yer almaktadır.

Şekil 8.2’de yer alan 14 baralı güç sistemine ait verileri TB ile çözdüğümüzde bize Şekil 8.9’da görünen sonuçları vermektedir.



Key	Type	Size	Value
solution	Array of float64	(6,)	[555.48019341 511.39680398 335.24384062 336.62990457 520.79160555 475. ...]
time	float	1	7.797998428344727
value	float64	1	47661.13440936732

Şekil 8.9 : 6 generatör ve 14 baralı sistemin python programa dili ile elde TB sonuçları

Türkiye’de yer alan Şekil 8.1’deki 8 generatör 22 baralı ve Şekil 8.2’deki 6 generatör ve 14 baralı sistemlerin maliyet ve süre hesaplamalarında kullanılan metasezgisel yöntemler olan GA, TB, TA ve YAK algoritmaları her hesaplama sonunda algoritmaların doğası gereği farklı maliyet ve süreler ortaya koymaktadır. Bu doğrultuda elde ettiğimiz sonuçlar bize bazı istatistiksel değerler sunmaktadır. Metasezgisel yöntemlerin maliyet ve süre değerlerinin bazı istatistiksel sonuçları Tablo 8.3’te yer almaktadır.

Tabloyu incelediğimiz zaman standart sapmalardan da anlaşılacağı üzere GA ile elde edilen sonuçların diğer metasezgisel algoritmalarla göre daha tutarlı olduğu görülmektedir. Burada oluşan farkın GA'nın popülasyon tabanlı bir algoritma olmasından kaynaklandığını düşünebiliriz. Diğer algoritmalarda oluşan rastgeleliğin fazla oluşuda etken olarak değerlendirilebilir. Ayrıca algoritmanın çözüm kalitesini, süresini ve stabilitesini etkileyen parametrelerin ideal şekilde ayarlanmamış olmasıda algoritmalar arasında standart sapma farkını ortaya çıkarmaktadır. Örneğin GA global optimuma yakınsama eğilimindeyken, TA yerel aramalarda daha fazla sapma gösterebilir. Bu da bize iterasyonlar arasında oluşan farkın neden oluştuğunu göstermektedir.

Metasezgisel yöntemlerin maliyet ve süre değerlerinin bazı istatistiksel sonuçları										
	GA (14 Baralı test sistemi için)		TB (14 Baralı test sistemi için)		TA(14 Baralı test sistemi için)		GA (22 Baralı test sistemi için)		YAK (22 Baralı test sistemi için)	
	Maliyet (\$/h)	Süre (sn)	Maliyet (\$/h)	Süre (sn)	Maliyet (\$/h)	Süre (sn)	Maliyet (\$/h)	Süre (sn)	Maliyet (\$/h)	Süre (sn)
1	47661.83	24.57	47909.98	2.03	48174.79	33.13	80532.51	91.89	80563.52	27.76
2	47662.02	21.69	47663.29	1.99	47970.10	33.48	80532.37	118.03	80536.68	27.61
3	47662.54	26.48	47661.40	1.97	48130.39	34.49	80533.78	95.04	80586.49	27.58
4	47662.82	19.67	48219.19	1.97	47991.57	33.82	80533.68	107.19	80550.45	27.56
5	47662.97	22.79	47661.27	1.97	47814.92	33.94	80533.39	98.27	80545.62	27.71
6	47663.94	23.02	47661.66	2.07	47895.92	33.55	80538.77	107.72	80583.94	27.51
7	47666.55	23.63	47662.13	2.02	47820.12	34.69	80531.36	105.65	80552.22	27.55
8	47661.95	24.22	47799.44	1.99	48455.64	33.41	80535.94	103.35	80557.44	27.73
9	47661.73	20.42	47661.74	2.03	48455.64	33.42	80532.91	106.03	80530.59	27.59
10	47662.03	19.91	47661.68	2.04	48301.19	33.30	80533.60	99.23	80541.96	27.59
11	47661.27	26.24	47662.32	1.98	48165.21	33.21	80537.06	100.33	80535.23	27.52
12	47664.52	22.65	47661.80	2.04	48220.89	33.55	80532.05	106.99	80541.38	27.58
13	47662.06	20.34	47662.03	1.97	48571.77	33.04	80533.71	113.48	80543.45	27.56
14	47663.48	21.26	47662.49	2.00	48607.79	33.19	80531.90	119.84	80591.85	27.68
15	47663.66	22.02	47662.29	1.99	48114.63	33.19	80532.86	113.10	80549.13	27.74
16	47661.94	20.52	47663.23	1.97	47811.52	33.13	80534.80	98.73	80530.66	28.42
17	47662.21	19.98	47661.69	2.00	48068.50	33.16	80535.94	100.66	80573.47	28.69
18	47662.76	23.52	47662.72	2.03	48095.63	33.14	80536.09	92.62	80562.15	28.98
19	47665.80	22.14	47663.42	2.02	48210.65	33.19	80535.50	100.91	80554.43	28.42
20	47663.34	21.86	47663.07	2.01	48099.30	33.18	80530.72	112.91	80550.26	28.38
Ortalama	47662.97	22.34	47709.34	2.00	48148.80	33.46	80533.94	104.59	80554.04	27.85
Standart Sapma	1.38	2.00	134.95	0.029	238.21	0.45	2.06	7.96	18.08	0.44
Varyans	1.81	3.81	17301.39	0.0008	53908.59	0.19	4.04	60.34	310.85	0.19

Tablo 8.3: Metasezgisel yöntemlerin maliyet ve süre değerlerinin bazı istatistiksel sonuçları

Türkiye’de yer alan Şekil 8.1’deki 8 generatör ve 22 baralı sistem için Tablo 8.3’te yer alan verilerden yola çıkarak; GA, YAK ve SLSQP algoritmaları ile yapılan EYD çözümlerinin sonuçları Tablo 8.4’te yer almaktadır. Elde edilen sonuçlar literatürde yer alan GA ve YAK algoritma sonuçları ile de karşılaştırılmıştır. Ayrıca Python ile yapılan optimizasyon çözümlerinde elde edilen çözüm süreleri için de Tablo 8.5 oluşturulmuştur.

Santraller	Ekonomik Yük Dağıtım Sonuçları ve Maliyet				
	GA [39]	YAK [39]	GA	YAK	SLSQP
Hamitabat	368.771	368.5353	366.583	367.435	368.7548
Ambarlı	480.198	480.0591	479.371	480.295	480.1974
Bursa DG	649.699	649.0000	651.504	650.617	649.6943
Seyitömer	587.312	586.1135	587.308	587.724	587.3105
SomaB	393.893	393.7376	394.131	393.592	393.8555
Yeniköy	384.342	384.0000	387.252	385.278	384.375
Kemerköy	592.043	592.0738	591.856	592.367	592.111
Yatağan	543.742	544.4815	542.267	542.784	543.701
MALİYET	80530 \$/h	80530 \$/h	80530 \$/h	80530.59 \$/h	80530.333 \$/h

Tablo 8.4: Türkiye’de kullanılan 8 generatör ve 22 baralı güç sisteminde kayıpsız durumda GA, YAK ve SLSQP ile elde edilen maliyet sonuçları

	22 Baralı Güç Sistemine Ait Optimizasyon Süreleri		
	GA	YAK	SLSQP
Süre	112.908 sn	27.53 sn	0.00914 sn

Tablo 8.5: Türkiye’de kullanılan 8 generatör ve 22 baralı güç sistemine ait kayıpsız durumda GA, YAK ve SLSQP ile elde edilen işlem süresi sonuçları

SLSQP algoritması ile 8 generatör ve 22 baralı sistemden elde ettiğimiz sonuçları, GA ve YAK algoritmalarıyla hem elde ettiğimiz hemde literatürde yer alan sonuçlar ile karşılaştırdığımızda gayet başarılı bir sonuç elde ettiğimiz Tablo 8.4’te bulunan maliyet sonuçlarından anlaşılmaktadır.

SLSQP algoritması ile elde ettiğimiz verilere baktığımız zaman maliyet sonucu diğer algoritmalar ile tutarlı iken 22 baralı güç sisteminde SLSQP algoritmasının avantajı Tablo 8.5'teki işlemci sürelerini incelediğimizde ortaya çıkmaktadır.

Tablo 8.5'te yer alan işlemci sürelerini incelediğimiz SLSQP algoritmasının avantajı çok net bir şekilde görülmektedir. Buradaki GA ile SLSQP arasında yaklaşık 12000 kat süre farkı, YAK ile SLSQP arasında ise yaklaşık 3000 kat süre farkının ortaya çıktığı görülmektedir.

Tablo 8.5'te yer alan bu süre farklarına bakarak Türkiye'de bölgesel yük dağıtımını, online olarak da verileri toplayıp optimize etme avantajını sağlayacağını görebiliriz.

Türkiye'de yer alan Şekil 8.2'deki 6 generatör ve 14 baralı sistem için Tablo 8.3'te yer alan verilerden yola çıkarak; GA, TA, TB ve SLSQP algoritmaları ile yapılan EYD çözümlerinin sonuçları aşağıda yer alan Tablo 8.6'te yer almaktadır.

Santraller	Ekonomik Yük Dağıtım Sonuçları						
	GA [55]	TB [55]	TA [55]	GA	TB	TA	SLSQP
Bursa DG	554.0455	483.5981	553.6206	557.269	555.480	562.116	555.002
Seyitömer	496.9588	478.4897	503.5243	516.859	511.396	537.369	515.072
SomaB	320.8097	294.8322	358.3317	331.623	335.243	340.720	334.073
Yeniköy	357.1712	374.3331	321.1134	334.035	336.629	358.286	336.559
Kemerköy	519.2502	582.2331	519.5332	517.798	520.791	459.218	518.819
Yatağan	486.6204	521.4198	478.7977	477.564	475.387	477.245	475.372
MALİYET	47679 \$/h	47876 \$/h	47678 \$/h	47661 \$/h	47661 \$/h	47727 \$/h	47660 \$/h

Tablo 8.6: Türkiye'deki 6 generatörlü 14 baralı enerji sisteminde kayıpsız durumda GA, TB, TA ve SLSQP ile elde edilen maliyet sonuçları

SLSQP algoritması ile 6 generatör ve 14 baralı sistemden elde edilen sonuçları GA, TA ve TB algoritmalarıyla elde edilen sonuçlar ile karşılaştırdığımızda gayet başarılı sonuçlar elde ettiğimiz Tablo 8.6'da yer alan maliyet sonuçlarından anlaşılmaktadır.

SLSQP algoritması ile elde ettiğimiz verilere baktığımız zaman maliyet sonucu diğer algoritmalarından daha iyi olduğu ayrıca 14 baralı güç sisteminde SLSQP algoritmasının başka bir avantajının aşağıda yer alan Tablo 8.7’teki işlemci sürelerini incelediğimizde ortaya çıkmaktadır.

	14 Baralı Güç Sistemine Ait Optimizasyon Süreleri			
	GA	TB	TA	SLSQP
Süre	26.24 sn	7.79 sn	33.77 sn	0.00706 sn

Tablo 8.7: Türkiye’de kullanılan 6 generatör ve 14 baralı güç sistemine ait kayıpsız durumda GA, TB, TA ve SLSQP ile elde edilen işlem süresi sonuçları

Tablo 8.7’de yer alan işlemci sürelerini incelediğimiz SLSQP algoritmasının avantajı çok net bir şekilde görülmektedir. Buradaki GA ile SLSQP arasında yaklaşık 3700 kat süre farkı, TB ile SLSQP arasında yaklaşık 1100 kat süre farkı, TA ile SLSQP karşılaştırdığımızda ise 4700 kat gibi bir süre farkı ortaya çıktığı görülmektedir.

22 baralı ve 14 baralı güç sistemlerindeki çözüm sürelerini karşılaştırdığımız zaman güç sistemlerinin büyümesiyle SLSQP ile diğer algoritmalar arasındaki süre farkının arttığı görülmektedir. Bu da bize daha büyük sistemlerde belkide online olarak SLSQP algoritmasının kullanılabileceğini göstermektedir.

9. SONUÇ VE ÖNERİLER

Ülkemizin dışa bağımlılığını azaltmada enerjinin ne kadar önemli olduğu herkes tarafından bilinmektedir. Günümüzde artan nüfus ile beraber, teknolojinin hızla büyümesi dünya genelinde kaynak tüketimine ivme kazandırmış ve enerjiye olan talep devasa boyutlara gelmiştir. Bu talebi karşılayacak enerji kaynaklarının yetersizliği sebebiyle, güç sistemlerinin en verimli şekilde işletilmesi büyük önem taşımaktadır. Enerjinin önemli rol oynadığı günümüzde, savaşın, ambargoların ve ekonominin ciddi etkileri de göz önüne alınarak özellikle dünyada enerjiye olan ihtiyacı karşılamak için ülkeler eski kapatılan termik santrallerini dahi yeniden devreye almak için çalışmalar yapmaktadır. Bu nedenle güç santrallerinin çalışma maliyetlerinin minimize edilmesi ülkemizde de önemli hale gelmiştir. Bu konuda Ekonomik Yük Dağıtımı (EYD) geçmiş zamanlardan daha önemli ve kıymetli bir hale gelmiştir. Ekonomik Yük Dağıtımı, reaktif güç iyileştirmesi ve en uygun güç iletimi araştırmaları gittikçe önem kazanmaktadır. Bu tez çalışmasında kullanılan SLSQP algoritması ile 8 generatörlü enerji sisteminde 80530.333 \$/h maliyet değeri ve 0.00914 sn'lik bir optimizasyon süresi elde edilmiş, 6 generatörlü enerji sisteminde ise 47660 \$/h maliyet değeri ve 0.00706 sn'lik bir optimizasyon süresi elde edilmiştir. Bu çalışmada elde edilen sonuç ve değerler ile Türkiye'de bulunan bölgesel güç santrallerinde EYD probleminin farklı optimizasyon algoritmaları ile karşılaştırmalı analizleri yapıldığında SLSQP algoritmasının generatör maliyetlerinin minimum olacak şekilde optimum çalışma koşullarının belirlenmesine yardımcı olduğu belirlenmiştir. Türkiye'deki bölgesel güç sistemleri gibi gerçek sistemler, çok sayıda generatör, karmaşık iletim kayıpları, generatör limitleri, doğrusal olmayan maliyet fonksiyonları ve birçok kısıt içerir. Yapılan çalışmada Türkiye'deki bölgesel güç sistemlerinin gerçekçi yapıları bütün kısıtlar göz ardı edilerek modellenmiştir. Bu nedenle yapılan çalışmada SLSQP ve metasezgisel algoritma karşılaştırmaları tercih edilmiştir. Bu çalışmada özellikle SLSQP ve farklı algoritmalar ile yaptığımız süre karşılaştırmaları bize minimum maliyet ile beraber işlemci sürelerinde ne kadar önemli olduğunu göstermiştir. Elde edilen ve gösterilen süre farkları bize SLSQP algoritmasının bölgesel sistemlerde hatta mümkün olduğu takdirde online sistemlerde kullanabilme avantajını kazandırabilir. Yapılan karşılaştırmalarda SLSQP algoritmasının geçerliliği ve avantajı görülmekte olup EYD problemine yeni bir çözüm önerisi getirebileceği söylenebilir.

KAYNAKÇA

- [1] **Ü. Başaran**, “Türkiye’deki 380 kV’luk Enterkonnekte Güç Sisteminde Çeşitli Güç Akışı ve Ekonomik Dağıtım Analizleri.” p. 117, 2004.
- [2] **T. Y. T. Y. H. ALTUN**, “TABU ARAŞTIRMASI UYGULANARAK EKONOMİK YÜK DAĞITIMI PROBLEMİNİN ÇÖZÜMÜ,” no. 1, 2011.
- [3] **B. Türkay**, “ECONOMIC DISPATCH AT THE AMBARLI POWER PLANT USING GENETIC ALGORITHM,” no. 1, 2002.
- [4] **N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay**, “Evolutionary programming techniques for economic load dispatch,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 1, pp. 83–94, 2003, doi: 10.1109/TEVC.2002.806788.
- [5] **E. Depatment**, “Economic dispatch solution using a real-coded genetic algorithm *,” vol. 5, no. 4, pp. 2–6, 2005.
- [6] **M. R. Alrashidi**, “Impact of Loading Conditions on the Emission- Economic Dispatch,” pp. 148–151, 2008.
- [7] **S. Coelho and V. C. Mariani**, “Chaotic artificial immune approach applied to economic dispatch of electric energy using thermal units,” *Chaos, Solitons and Fractals*, vol. 40, no. 5, pp. 2376–2383, 2009, doi: 10.1016/j.chaos.2007.10.032.
- [8] **Serdar ÖZYÖN**, “Genetik Algoritmanın Bazı Çevresel Ekonomik Güç Dağıtım Problemlerine Uygulanması,” 2009, [Online]. Available: https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=DCxE5NsKUFfmxHt1Js92Qw&no=tXlms5wILmxMsaU-wC_iqg.
- [9] **Emrah ÇETİN**, “Rüzgar Eenerjisi Dahil Olan Güç Sistemlerinde Ekonomik Yük Dağıtımı Probleminin Çözümü,” 2011.
- [10] **X. Yang, S. Soheil, S. Hosseini, and A. Hossein**, “Firefly Algorithm for solving non-convex economic dispatch problems with valve loading effect,” *Appl. Soft Comput. J.*, vol. 12, no. 3, pp. 1180–1186, 2012, doi: 10.1016/j.asoc.2011.09.017.
- [11] **S. Özyön et al.**, “Diferansiyel Gelişim Algoritmasının Yasak İşletim Bölgesi Ekonomik Güç Dağıtım Problemine Uygulanması The Application of Differential Evolution Algorithm for the Economic Power Dispatch with Prohibited Operating

- Zone,” *Erciyes Üniversitesi Fen Bilim. Enstitüsü Derg.*, vol. 29, no. 1, pp. 61–69, 2013.
- [12] **L. Slimani**, “Turkish Journal of Electrical Engineering and Computer Sciences Economic power dispatch of power systems with pollution control using artificial bee colony optimization,” vol. 21, no. 6, 2013, doi: 10.3906/elk-1106-10.
- [13] **B. H.**, “Analysis of Economic Load Dispatch Using Fuzzified Pso,” *Int. J. Res. Eng. Technol.*, vol. 03, no. 15, pp. 237–241, 2014, doi: 10.15623/ijret.2014.0315046.
- [14] **E. E. Elattar**, “Electrical Power and Energy Systems A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem,” *Int. J. Electr. Power Energy Syst.*, vol. 69, pp. 18–26, 2015, doi: 10.1016/j.ijepes.2014.12.091.
- [15] **M. S. P. Subathra, S. Easter Selvan, T. Aruldoss Albert Victoire, A. Hepzibah Christinal, and U. Amato**, “A hybrid with cross-entropy method and sequential quadratic programming to solve economic load dispatch problem,” *IEEE Syst. J.*, vol. 9, no. 3, pp. 1031–1044, 2015, doi: 10.1109/JSYST.2013.2297471.
- [16] **M. S. TURGUT and G. K. DEMİR**, “Ekonomik Yük Dağıtımı Probleminin Yapay İşbirlikçi Algoritması ile Çözümü Mert,” *Dokuz Eylul Univ. Eng. J. Sci. Eng.*, vol. 20, no. 59, pp. 201–211, 2018, doi: 10.21205/deufmd.
- [17] **U. Eminoğlu and O. Karahan**, “Metasezgisel Metotlar Kullanılarak Ekonomik Yük Dağıtımı Probleminin Çözümü: Grafiksel Kullanıcı Arayüzü Uygulaması,” *J. Polytech.*, vol. 0900, no. 4, pp. 827–835, 2017, doi: 10.2339/politeknik.369019.
- [18] **I. N. Trivedi, N. Jangir, P. Jangir, M. H. Pandya, R. H. Bhesdadiya, and A. Kumar**, “Price penalty factors based approach for emission constrained economic dispatch problem solution using whale optimization algorithm,” *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst. ICPEICES 2016*, pp. 1–5, 2017, doi: 10.1109/ICPEICES.2016.7853284.
- [19] **R. K. Abhishek Soni, Hariom Mishra, Vikas Kumar Koli**, “Economic Load Dispatch using Artificial Bee Colony Optimization,” *Int. J. Adv. Electron. Eng.*, vol. 1, no. 1, pp. 119–124, 2011.
- [20] **M. Said, A. M. El-Rifaie, M. A. Tolba, E. H. Houssein, and S. Deb**, “An efficient chameleon swarm algorithm for economic load dispatch problem,” *Mathematics*,

vol. 9, no. 21, pp. 1–14, 2021, doi: 10.3390/math9212770.

- [21] **A. Kaur, L. Singh, and J. S. Dhillon**, “Modified Krill Herd Algorithm for constrained economic load dispatch problem,” *Int. J. Ambient Energy*, vol. 43, no. 1, pp. 4332–4342, 2022, doi: 10.1080/01430750.2021.1888798.
- [22] **B. H. Tahir et al.**, “Improved Fitness-Dependent Optimizer for Solving Economic Load Dispatch Problem,” vol. 2022, 2022.
- [23] **H. Berahmandpour, S. M. Kouhsari, and H. Rastegar**, “International Journal of Electrical Power and Energy Systems A new flexibility based probabilistic economic load dispatch solution incorporating wind power,” *Int. J. Electr. Power Energy Syst.*, vol. 135, no. September 2021, p. 107546, 2022, doi: 10.1016/j.ijepes.2021.107546.
- [24] **R. Ragunathan and B. Ramadoss**, “Golden jackal optimization for economic load dispatch problems with complex constraints,” *Bull. Electr. Eng. Informatics*, vol. 13, no. 2, pp. 781–793, 2024, doi: 10.11591/eei.v13i2.6572.
- [25] **İ. EKE, M. SAKA, and S. S. TEZCAN**, “KAOTİK Parçacık SürOptimizasyonu KullanarakEkonomik Yük DağıtımProbleminin Çözümü,” *Mühendislik Bilim. ve Tasarım Derg.*, vol. 11, no. 3, pp. 957–965, 2023, doi: 10.21923/jesd.1293964.
- [26] **Y. Zhang and H. Li**, “Research on Economic Load Dispatch Problem of Microgrid Based on an Improved Pelican Optimization Algorithm,” *Biomimetics*, vol. 9, no. 5, 2024, doi: 10.3390/biomimetics9050277.
- [27] **A. E. Shaban, A. A. K. Ismaeel, A. Farhan, M. Said, and A. M. El-Rifaie**, “Growth Optimizer Algorithm for Economic Load Dispatch Problem: Analysis and Evaluation,” *Processes*, vol. 12, no. 11, pp. 1–23, 2024, doi: 10.3390/pr12112593.
- [28] “**Saadat H.**, Power Systems Analysis, McGraw Hill, Boston, 1999.” .
- [29] **A. Zaraki and M. F. Bin Othman**, “Implementing particle swarm optimization to solve economic load dispatch problem,” *SoCPaR 2009 - Soft Comput. Pattern Recognit.*, pp. 60–65, 2009, doi: 10.1109/SoCPaR.2009.24.
- [30] **Y. Bin Qiao, H. Qi, Y. T. Ren, J. P. Sun, and L. M. Ruan**, “Application of SQP algorithm for fluorescence tomography with the time-domain equation of radiative transfer,” *J. Quant. Spectrosc. Radiat. Transf.*, vol. 193, no. January, pp. 21–30, 2017, doi: 10.1016/j.jqsrt.2017.03.007.

- [31] **Xin-SheYang**, *Engineering Optimization An Introduction with Metaheuristic Applications*. 2018.
- [32] **M. Basu**, “Dynamic economic emission dispatch using evolutionary programming and fuzzy satisfying,” *Int. J. Emerg. Electr. Power Syst.*, vol. 2, p. 1065, 2005.
- [33] **J. Nocedal** and **S. J. Wright**, *Numerical Optimization*. 2006.
- [34] **L. LJUNG**, “System Identification,” *Prentice Hall Inf. Syst. Sci. Ser.*, vol. 2, 1999, [Online]. Available: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBERTUNGAN_TERPUSAT_STRATEGI_MELESTARI.
- [35] **C. T. Kelley**, “1. Basic Concepts and Stationary Iterative Methods,” *Iterative Methods for Linear and Nonlinear Equations*, pp. 3–10, 1995, doi: 10.1137/1.9781611970944.ch1.
- [36] **E. AYDIN**, “ELD optimization with SLSQP,” <https://github.com/EnzelAydin/ELD-optimization-with-SLSQP>, 2025. <https://github.com/EnzelAydin/ELD-optimization-with-SLSQP>.
- [37] **S. Katoch**, **S. S. Chauhan**, and **V. Kumar**, *Katoch2021_Article_AReviewOnGeneticAlgorithmPastP.pdf*, vol. 80. Multimedia Tools and Applications, 2021.
- [38] **B. Alhijawi** and **A. Awajan**, “Genetic algorithms: theory, genetic operators, solutions, and applications,” *Evol. Intell.*, vol. 17, no. 3, pp. 1245–1256, 2024, doi: 10.1007/s12065-023-00822-6.
- [39] **Ö. P. ARSLAN**, “Güç sistemlerinin ekonomik ve çevresel ekonomik yük dağıtımında yeni optimizasyon tekniklerinin kullanımı,” vol. 9, p. 102, 2015.
- [40] **D. Karaboga**, “AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION,” *Tech. REPORT-TR06*, no. 2, pp. 1–7, 2005.
- [41] **V. Tereshko**, “Reaction-Diffusion Model of a Honeybee Colony’s Foraging Behaviour,” 2000.
- [42] **D. Karaboga** and **B. Basturk**, “On the performance of artificial bee colony (ABC)

- algorithm,” *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 687–697, 2008, doi: 10.1016/j.asoc.2007.05.007.
- [43] **T. D. Seeley**, *Honeybee Democracy (ミツバチの会議)*. 2010.
- [44] **B. Akay** and **D. Karaboga**, “A modified Artificial Bee Colony algorithm for real-parameter optimization,” *Inf. Sci. (Ny)*, vol. 192, pp. 120–142, 2012, doi: 10.1016/j.ins.2010.07.015.
- [45] **D. Karaboga** and **B. Gorkemli**, “A quick artificial bee colony -qABC- algorithm for optimization problems,” *INISTA 2012 - Int. Symp. Innov. Intell. Syst. Appl.*, pp. 1–5, 2012, doi: 10.1109/INISTA.2012.6247010.
- [46] **V. Yamaçlı**, **K. Abacı**, ve **E. M. Bölümü**, “Güç Sistemlerinde Aktif Güç Kaybının Optimizasyonu -Optimization of the Active Power Loss in Power Systems,” pp. 27–29, 2014.
- [47] **F. Glover**, “Future paths for integer programming and links to artificial intelligence,” *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986, doi: 10.1016/0305-0548(86)90048-1.
- [48] **S. O. Değertekin**, **M. Ülker**, and **M. S. Hayalioğlu**, “Uzay çelik çerçevelerin tabu arama ve genetik algoritma yöntemleriyle optimum tasarımı,” *Tek. Dergi/Technical J. Turkish Chamb. Civ. Eng.*, vol. 17, no. 3, pp. 3917–3934, 2006.
- [49] **T. Search**, *Fred Glover, Manuel Laguna (auth.)-Tabu Search-Springer US (1997)*.
.
- [50] **M. Gendreau** and **J.-Y. Potvin**, *Handbook of Metaheuristics Third Edition*. 2019.
- [51] **El-Ghazali Talbi**, *METAHEURISTICS FROM DESIGN TO IMPLEMENTATION*. 2009.
- [52] **Y. Rochat** and **É. D. Taillard**, “Probabilistic diversification and intensification in local search for vehicle routing,” *J. Heuristics*, vol. 1, no. 1, pp. 147–167, 1995, doi: 10.1007/BF02430370.
- [53] **A. S. Kirkpatrick**, **C. D. Gelatt**, **M. P. Vecchi**, **S. Science**, **N. Series**, and **N. May**, “Optimization by Simulated Annealing Published by : American Association for the Advancement of Science Stable URL : <http://www.jstor.com/stable/1690046>,” *Science (80-.)*, vol. 220, no. 4598, pp. 671–680, 1983.

- [54] **Y. GÖZÜDELİ** and **M. al. AKCAYOL**, “XML Veritabanı için Tavlama Benzetimi ile Sorgu Optimizasyonu,” *Bilişim Teknol. Derg.*, vol. 1, no. 1, 2010, [Online]. Available: <https://dergipark.org.tr/en/pub/gazibtd/issue/6612/87861>.
- [55] **A. Ö. Ö. Duman, Serhat, M. Kenan Döşoğlu**, “TÜRKİYEDEKİ GÜÇ SİSTEMİNDE TAVLAMA BENZETİMİ, GENETİK ALGORİTMA ve TABU ARAŞTIRMA ALGORİTMALARI KULLANILARAK EKONOMİK DAĞITIM,” no. March 2009, pp. 1–23, 2010.
- [56] “**Python scipy**,” [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>.



ÖZGEÇMİŞ

Ad-Soyad : Enzel AYDIN

ÖĞRENİM DURUMU:

- **Lisans:** 2005-2011, Mersin Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği

MESLEKİ DENEYİM

- **12.2013-08.2024** tarihleri arasında Afşin-Elbistan B Termik Santrali bünyesinde Ölçü-Kontrol Mühendisi ve Bilgi İşlem Mühendisi olarak görev aldım.
- **08.2024-** Ağustos 2024 tarihi itibariyle Dicle ve Yöresi HES İşletme Müdürlüğü bünyesinde Diyarbakır'da bulunan barajlara ait otomasyon sistemlerinin kontrolü için görevlendirildim.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **AYDIN, E. & AKKAYA, A.E.** (2025). An Analysis Of Economic Load Dispatch Of Power Systems. 7th INTERNATIONAL Mediterranean Scientific Research Congress 330. https://izdas.org/_files/ugd/614b1f_7ec08aeb3cc241dfa572261d352447ae.pdf