

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRİK- ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**İLETİM ŞEBEKESİNDE DARALMA KATSAYISI PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİ
KULLANARAK TCSC İLE GÜÇ KAYBI OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

Elektrik- Elektronik Müh. Maimaijiang Aishan

**EKİM 2021
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK- ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**İLETİM ŞEBEKESİNDE DARALMA KATSAYISI PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİ
KULLANARAK TCSC İLE GÜÇ KAYBI OPTİMİZASYONU**

Elektrik- Elektronik Müh. Maimaijiang Aishan

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
" ELEKTRİK YÜKSEK MÜHENDİSİ"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 02 / 09 / 2021

Tezin Savunma Tarihi : 14 / 10 / 2021

Tez Danışmanı : Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU

Trabzon 2021

ÖNSÖZ

Elektrik enerjisine olan ihtiyacın her geçen gün artmasıyla beraber iletim şebekelerinde reaktif güç tüketiminden dolayı gereksiz yere kaybedilen enerji oldukça artmaktadır ve mevcut enerjinin verimli bir şekilde kullanılmasını zorunlu kılmaktadır. Bahsedilen problem bu tez çalışmasında ele alınarak ülke ekonomisinin önemli faktörü olan enerji kayıpları ve gerilim profilinin iyileştirilmesi amaçlanmıştır. Problemin çözümü için Standart Parçacık Sürüsü Optimizasyonu (SPSO), Atalet Ağırlığı Parçacık Sürüsü Optimizasyonu (IWPSO), Daralma Katsayısı Parçacık Sürüsü Optimizasyonu (CCPSO) kullanılarak iletim şebekesine yerleştirilmesi gereken Tristör Kontrollü Seri Kapasitör (TCSC) optimum konumunu ve değerini belirlemiştir.

Bu çalışmanın amacına ulaşması için engin bilgi ve yenilikçi düşünceleriyle bana yol gösteren, her daim beni destekleyen, ilgi ve tecrübelerini esirgemeyen değerli hocam Dr. Öğr. Üyesi Fatih M. NUROĞLU'na teşekkürlerimi saygı ve minnetle sunarım. Son olarak Türkiye'deki eğitim hayatım için bana destek olan Yurtdışı Türkler ve Akraba Topluluklar Başkanlığına ve hayatım boyunca varlıkları ve destekleriyle bana güç veren ve her zaman yanımda olan anneme ve babama şükranlarımı sunarım.

Maimaijiang Aishan

Trabzon 2021

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “İLETİM Şebekesinde Parçacık Sürüsü Optimizasyon Yöntemini Kullanarak TCSC ile Güç Kaybı Optimizasyonu” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Fatih Mehmet NUROĞLU’nun sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 14/10/2021

Maimaijiang Aishan

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	III
TEZ ETİK BEYANNAMESİ	IV
İÇİNDEKİLER.....	V
ÖZET	VI
SUMMARY	VIII
ŞEKİLLER DİZİNİ.....	IX
TABLolar DİZİNİ	X
SEMBOLLER DİZİNİ.....	XI
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Literatür Araştırması	2
1.3. Transfer Kapasitesini Artırma Yöntemleri.....	4
1.4. İletim Sistemlerinde Güç Kayıplarını Azaltma Yöntemleri.....	4
1.5. Esnek AC İletim Sistemleri.....	5
1.5.1. Seri Bağlı Kontrol Birimi.....	5
1.5.2. Şönt Bağlı Kontrol Birimi	6
1.5.3. Kombine Seri Kontrol Birimi.....	6
1.5.4. Kombine Seri Şönt Kontrol Birimi	6
1.6. ACTS Cihazlarının Uygulamaları.....	7
1.6.1. Kararlı Durum Uygulamaları	7
1.6.2. Yükleme Marjı İyileştirme Uygulaması.....	7
1.6.3. Güç Akışı ve Dengeleme Kontrolü Uygulaması.....	7
1.7. İletim Kapasitesinin Geliştirilmesi İçin FACTS Cihazlarının Konumu	8
1.8. TCSC'nin Modellenmesi.....	8
1.9. Amaç Fonksiyonu	9
1.9.1. Şebekedeki Aktif Güç Kaybı.....	9
1.9.2. Gerilim Profilini İyileştirme.....	10
1.9.3. TCSC Maliyet Fonksiyonu	10
1.10. Yük Akışı Analiz Yöntemleri	11
1.10.1. Gauss-Siedel (GS) Yük Akışı Yöntemi	11
1.10.2. Hızlı Ayrılmış Yük Akışı (FDLF) Yöntemi.....	12

1.10.3. Newton Raphson Yük Akış Yöntemi.....	14
1.11. Optimizasyon Algoritması	17
1.11.1. Optimizasyonda Sorunlar	18
1.11.2. Doğru Algoritma Seçimi	18
1.11.3. Genetik Algoritması	19
1.11.4. Parçacık Sürüsü Optimizasyonu Algoritması	19
1.12. Farklı PSO Algoritmalar	22
1.12.1. Standart Parçacık Sürüsü Optimizasyonu (SPSO)	22
1.12.2. Atalet Ağırlığı Parçacık Sürüsü Optimizasyonu (IWPSO)	22
1.12.3. Daralma Katsayısı Parçacık Sürüsü Optimizasyonu (CCPSO).....	23
1.12.4. Çok Fazlı PSO (MPPSO)	23
1.12.5. Dinamik Komşuluk PSO (DNPSO)	24
2. YAPILAN ÇALIŞMALAR	24
2.1. Sorun ve Çözüm	24
2.2. Kullanılan Metodun Test Sisteminde Ayarları.....	25
2.2.1. Standart PSO Parametre Ayarları (SPSO)	27
2.2.2. Atalet Ağırlığı PSO Parametre Ayarları (IWPSO)	28
2.2.3. Daralma Katsayısı PSO Parametre Ayarları (CCPSO).....	28
2.3. PSO Algoritmasının Uygulanması	29
2.4. Kodlama	30
2.4.1. TCSC Modeli	30
2.4.2. Amaç Fonksiyon Oluşturma.....	30
2.4.3. Newton Raphson Yük Akış Yöntemi Oluşturma.....	31
2.4.4. PSO Algoritmasını Oluşturma	31
3. ÇALIŞMA SONUÇLARI.....	27
3.1. Birinci Çalışma Senaryosu	27
3.2. İkinci Çalışma Senaryosu.....	41
3.2.1. IEEE 14 Baralı Sistemin Performans Sonuçları.....	41
3.2.2. IEEE 30 Baralı Sistemin Performans Sonuçları.....	45
3.3. IEEE 14 ve 30 Baralı Sistemin Güç Kaybı Sonuçları.....	49
3.4. IEEE 14 ve 30 Baralı Sistemin TCSC Optimal Konumu, Büyüklüğü ve Kurulum Maliyeti	50
4. TARTIŞMA VE BULGULAR	51

5.	ÖNERİLER	52
6.	KAYNAKLAR.....	53
7.	EKLER.....	54
ÖZGEÇMİŞ		



ÖZET

İLETİM ŞEBEKESİNDE DARALMA KATSAYISI PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİ KULLANARAK TCSC İLE GÜÇ KAYBI OPTİMİZASYONU

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik- Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr.Öğr.Fatih Mehmet NUROĞLU
2021, 76 Sayfa

Sosyal yaşamın gelişmesiyle birlikte yerleşik alanlarda artan reaktif güç tüketimi, hatlara yüklenilerek gerilim düşümünün ve güç kayıplarının artmasına neden olurken hattın taşıma kapasitesinin de düşmesine yol açar. Bu olumsuz etkileri ortadan kaldırmak için yaygın olarak kullanılan en etkili yöntemlerden biri iletim şebekesine TCSC yerleştirmektir.

TCSC teknolojisinin iletim şebekesinde ana rolü hat gerilim seviyesini artırmak, güç kaybını en az seviyeye indirmek ve hattın aktarma kapasitesini artırmaktır. Önceki TCSC'nin optimum konumu ve boyutu deneyimine göre farklı Parçacık Sürüsü Optimizasyonu (PSO) algoritması seçilmektedir. Bu nedenle, TCSC'nin optimum konumunu ve boyutunu bulmak için yeni bir PSO optimizasyon metodu önemlidir. Aynı zamanda, optimum konumu ve en iyi TCSC boyutu iletim şebekesinin güç kaybını en az seviyeye indirebilmektedir.

Bu çalışmada, standart PSO, atalet ağırlığı PSO ve yeni metot olarak daralma katsayısı tabanlı PSO ile güç kaybını en aza indirmek için güç sisteminde TCSC optimum konumunu ve değerini sunmaktadır. Aynı zamanda bu üç farklı algoritma üzerinde karşılaştırma yapılmıştır. Simülasyonlar, IEEE 14 bara sistemi, IEEE 30 baralı sisteminde TCSC de dahil olmak üzere Newton Raphson yük akış yöntemini kullanarak analize edilmiştir.

Anahtar Kelimeler: TCSC, CCPSO, Newton Raphson Yük Akış Yöntemi

Master Thesis

SUMMARY

POWER LOSS OPTIMIZATION WITH TCSC USING CONSTRAINT COEFFICIENT PARTICLE SWARM OPTIMIZATION METHOD IN TRANSMISSION NETWORK

Karadeniz Technical University

The Graduate School of Natural and Applied Sciences

Electrical-Electronics Engineering Graduate Program

Supervisor: Asst. Prof. Dr. Fatih Mehmet NUROĞLU

2021, 76 Pages

Development of social life has caused increment in reactive power consumption in residential areas. It is obvious that the overloading of the line due to increased reactive power should be prevented in order not to cause high power losses and voltage drops. Therefore, TCSC are prevalently used in transmission systems in order to solve these problems.

The main role of Thyristor Controlled Serial Capacitor (TCSC) technology in the transmission network is to increase the line voltage level, minimize the power loss, increase transmission capacity. In previous studies, the appropriate location and value is selected according to the optimal location and value experience of the Thyristor Controlled Series Capacitor (TCSC). Therefore, a new optimization method is important to find the most suitable location and size of the TCSC. At the same time, the best location and the best TCSC size can minimize the loss of the transmission network.

In this study, using standard PSO, inertia weight PSO and for the new method using constraint coefficient PSO to minimize power loss, it presents the optimum position and value of Thyristor Controlled Series Capacitor (TCSC) in the power system. Simulations were analyzed using the Newton Raphson load flow method, including the IEEE 14 bus system and the IEEE 30 bus system with TCSC.

KeyWords: TCSC, CCPSO, Newton Raphson Load Flow Method

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil.1. TCSC'nin değişken reaktans modeli	8
Şekil 2. PSO akış şeması.....	21
Şekil 3. IWPSO algoritması gerilim performansı, Burada (a) Bara 26'ye yük ekledikten sonraki gerilim profili, (b) Bara 29'a yük ekledikten sonraki gerilim profili	25
Şekil 4. IWPSO algoritması ile bara 26'ye yük ekledikten sonraki performansı	33
Şekil 5. IWPSO algoritması ile bara 29'a yük ekledikten sonraki performansı	34
Şekil 6. CCPSO algoritması ile bara 26'ye yük ekledikten sonraki performansı.....	35
Şekil 7. CCPSO algoritması ile bara 29'a yük ekledikten sonraki performansı.....	36
Şekil 8. Yük ekledikten sonra bara 26 üzerindeki aktif güç kaybı	38
Şekil 9. Yük ekledikten sonra bara 29 üzerindeki aktif güç kaybı	40
Şekil 10. IEEE-14 baralı sisteminin tek hat şeması	41
Şekil 11. IEEE-14 baralı sisteminde TCSC olmadan ve TCSC ile gerilimi profili, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı tabanlı PSO sonuçları ..	42
Şekil 12. IEEE 14 baralı sisteminde PSO algoritmaları performansı, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı PSO sonuçları	44
Şekil 13. IEEE-30 baralı sisteminin tek hat şeması	45
Şekil 14. IEEE-30 baralı sisteminde TCSC olmadan ve TCSC ile gerilimi profili, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı tabanlı PSO sonuçları ..	46
Şekil 15. IEEE 30 baralı sisteminde PSO algoritmaları performansı, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı PSO sonuçları	47

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Standard PSO parametreleri	27
Tablo 2. Atalet ağırlığı PSO parametreleri	28
Tablo 3. Daralma katsayısı PSO Parametreleri.....	29
Tablo 4. Bara 26 üzerindeki iletim kaybı ve TCSC yerleşimi	37
Tablo 5. Bara 29 üzerindeki iletim kaybı ve TCSC yerleşimi	39
Tablo 6. MW cinsinden aktif güç iletim kaybı	49
Tablo 7. MVAr cinsinden reaktif güç iletim kaybı.....	49
Tablo 8. TCSC'nin optimal yeri, büyüklüğü ve kurulum maliyeti.....	50

SEMBOLLER DİZİNİ

PSO	Parçacık Sürüsü Optimizasyonu
FACTS	Esnek AC İletim Sistemi
TCSC	Tristör Kontrollü Seri Kapasitör
TCSR	Tristör Kontrollü Seri Reaktör
STATCOM	Statik Kapasitör
UPFC	Birleşik Güç Akış Kontrolörü
SSSC	Statik Senkron Serisi Kapasitör
IPFC	Interline Güç Akış Kontrolörü
TCPST	Tristör Kontrollü Faz Kaydırıcı Transformatör
VAR	Volt Amper Reaktif
SVC	Statik VAR telafisi
GA	Genetik Algoritma
SA	Benzetimli tavlama Algoritma
DE	Diferansiyel Evrim
BFOA	Bakteri Toplama Optimizasyonu Algoritması
X_{kj}	“k” ve “j” baralar arasındaki iletim hattının reaktansı
X_{TCSC}	TCSC'nin Reaktansı
X_{line}	TCSC yerleştirildikten sonra hattın toplam reaktansı
Nline	Toplam iletim hattı sayısı
G _{kj}	“K” ve “j” baralar arasındaki iletkenlik
V _k	“k” Barası gerilimi
V _j	“j” Barası gerilimi
Δ_k	“k” Barası açıları
Δ_j	“J” Barası açıları
V _k	k barası gerilim büyüklüğü
V _{ref}	Bara i geriliminin referans değeri
n _k	Bara sayısı
S	TCSC'nin [MVar] çalışma aralığı
IC	TCSC cihazlarının Kurulum maliyeti [US \$]
CTCSC	[US \$ / KVar] cinsinden TCSC cihazlarının maliyeti

Q_1	TCSC kurulumundan önce reaktif güç
Q_2	TCSC kurulumundan sonra reaktif güç
P_i^G	Bara i üzerindeki Aktif güç üretimi
P_i^D	Bara i üzerindeki Aktif güç talebi
w	Ağırlık faktörü
C_1	Bilişsel sabit değeri
C_2	Sosyal ivme sabiti
$rand_1$	(0, 1) arasında eşit dağılımlı rasgele sayılar
$rand_2$	(0, 1) arasında eşit dağılımlı rasgele sayılar
X_i^k	“i” parçacığının iterasyonda mevcut konumu k
V_i^k	“i” parçacığının iterasyon k'daki mevcut hızı
X_i^{k+1}	“i” parçacığının iterasyondaki mevcut konumu k +1
V_i^{k+1}	“i” parçacığının iterasyon k'daki mevcut hızı k + 1
Q_k^G	Bara reaktif güç üretimi
Q_k^D	Bara reaktif güç ihtiyacı
P_k^c	Bara i üzerindeki aktif güç
Q_k^c	Bara i üzerindeki reaktif güç

1. GENEL BİLGİLER

1.1. Giriş

Elektrik enerjisi, günümüzde insanların temel gereksinimlerinin karşılanması, ülkelerin sosyal gelişimleri ve ekonomik olarak kalkınması için en önemli ihtiyaç maddelerinden biridir. Elektrik enerjisine olan ihtiyaç, dünya nüfusundaki hızlı artışa ve endüstriyel gelişmelere bağlı olarak gün geçtikçe artmaktadır. Artan bu enerji gereksinimini karşılamak için üretimin arttırılması kadar mevcut enerjinin yüksek verimle kullanılması oldukça önemlidir.

Elektrik enerjisinin verimliliği ve ekonomik olarak sağlanabilmesi, güç sistemlerindeki en önemli konulardan biridir. Enerji kullanımında verimliliği ve maliyeti etkileyen en önemli faktör güç iletim sistemlerindeki kayıplardır.

Sosyal yaşamın gelişmesi ile artan refah seviyesi, yerleşik alanlarda elektrik güç tüketimini artırmıştır. Elektrik güç tüketiminin artması hatları gereksiz yere yükleyerek hattaki güç kayıplarının ve gerilim düşümünün artmasına neden olurken hattın taşıma kapasitesinin düşmesine yol açar.

FACTS cihazları, güç akışını yönetebilen ve güç sisteminin kararlılığını artıran cihazlardır. Ayrıca hat empedansı, terminal gerilimleri ve gerilim açısı gibi iletim hattında tanımlanan parametreleri ve değişkenleri hızlı ve etkili bir şekilde ayarlayabilirler. Bir güç sisteminin daha esnek, güvenli ve ekonomik bir şekilde çalışmasını sağlayabilir. Hatların dayatılmasına yol açan güç üretim modelleri, daha yüksek hat kayıplarına neden olacak ve güç sistemlerinin güvenliğini ve kararlılığını zayıflatacaktır. Bu durumda, FACTS cihazları iletim hatlarındaki güç akışını kontrol ederek sistemin kabiliyetini arttırmak için kullanılabilir [1].

Bu tez çalışmasında, İletim şebekelerinde kayıpları azaltarak tüketiciye daha kaliteli bir enerjinin ekonomik olarak sunulması için TCSC optimum konumunu ve boyutunun

tespiti gerçekleştirilmiştir. Çalışma iki senaryoya bölünmüştür, Birinci senaryoda Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmada kullanılan atalet ağırlığı PSO ile bu çalışmada yeni olarak önerilen daralma katsayısı PSO'yu IEEE 30 baralı sistem üzerinde aynı şartlar altında karşılaştırılmıştır. İkinci senaryoda standart PSO, atalet ağırlığı PSO ve daralma katsayısı PSO ile IEEE 14 baralı ve IEEE 30 baralı sisteminde TCSC de dahil olmak üzere Newton Raphson yük akış yöntemi kullanılarak analiz edilmiştir. Böylelikle güç kaybını en az seviyeye indirilmesi hedeflenmiştir. Son olarak IEEE 14 baralı ve IEEE 30 baralı sistemlerinde optimum konum, optimum güç kaybı, optimum maliyeti hesaplanmıştır.

1.2. Literatür Araştırması

Son otuz yılda, FACTS kontrolörleri ve güç sistemindeki uygulamalar çok farklı yapıda tasarlanıp analiz edilmiştir. Şimdiye kadar, iletim hatlarındaki güç kayıplarını azaltmak için çeşitli FACTS cihazları kullanılmış ve çeşitli optimizasyon yöntemleri üzerinde uygulamalar yapılmıştır.

Siti Amely Jumaat ve arkadaşları [2] PSO tekniği kullanarak iletim kaybını en aza indirmek ve sistemdeki gerilim profilini artırmak için FACTS cihazlarının yerleşim ve boyutlandırma çalışmasını yapmıştır. TCSC, telafi cihazı olarak seçilir ve hatta eklenen reaktif bir kaynak olarak modellenir. Bu çalışma IEEE 30 baralı sisteminde atalet ağırlığı PSO kullanarak gerçekleştirilmiştir.

Xie da ve arkadaşları [3] TCSC kullanan büyük sistemlerde gücü kontrol etmek için bir algoritma sunar. Değiştirilmiş denklemler TCSC baraların çıkışına veya hattın ortasına mont edildiğinde Newton Raphson yük akış yöntemi kullanılarak sistemin analizi için güç akışına uygulanır. TCSC çalışırken akım akışının iyileştirilmesine yönelik Newton Raphson yük akış yöntemi önermektedir. Aynı zamanda şebekede arza olduğunda TCSC'nin güç akış kontrollüde tartışılmıştır.

Fuerte ve arkadaşları [4] TCSC modelini kullanan pratik güç şebekelerinin güç akışı çalışması hakkında bilgi vermektedir. Yazar, TCSC için modern bir güç akışı modeli sunmaktadır. Bu modelde güç akışını güvenilir bir şekilde çözmek için Newton Raphson yük akış yöntemi kullanılmıştır, Çalışmada TCSC eşdeğer reaktansının karşılaştırılması, kapasitörde depolanan yükün etkisini göstermek için yapılmaktadır.

H. Pinto ve arkadaşları [5] ULTC transformatör kademe kontrolü, gerilim kontrolü, HVDC bağlantı kontrolü ve hat akımı kontrolü için TCSC ekipmanı kullanmıştır. Bu sistemde yük akışını analize etmek için Newton-Raphson algoritması uygulamıştır, Yazar pratik test sistemindeki sonuçlara dayanarak önerilen modelin istatistiksel analizlerini sunmaktadır.

S. Mukhopadhyay ve arkadaşları [6] TCSC ve Sabit Seri Telafisinin (FSC) dinamik performansı artırmak için iletim hattındaki rolünü tartışmaktadır.

Abdel-Moamen M.A ve N.P. Padhy [7] IEEE 30 baralı sistemlerinde Newton-Raphson TCSC modelini kullanarak güç akış çözümünü tartışmıştır.

John J. Paserba [8] güç sistemlerinin kontrol edilebilirliğini ve AC iletim sistemindeki FACTS kontrolörlerinin faydalarını gösteren güç sistemlerinin kontrolü ile ilgilenmektedir. Güç akışı kontrolünü geliştirmek için geleneksel ekipman ve FACTS kontrolörlerinden örnekler almıştır. Yazar, FACTS cihazlarının iletim sistemine kurulması için güç sistemi analizinin önemli aşamalarını açıklamakta ve ayrıca FACTS kontrolör devrelerine genel bir bakış sunmaktadır.

Sıhhiyeci Hassan ve arkadaşları [9] SVC ve TCSC için kararlı bir durum modeli önermiştir. Newton-Raphson algoritması ile güç akış denklemlerini çözmek için kullanılmıştır ve performansı IEEE 9 baralı sisteminde test edilmiştir.

J. V. Parate ve A. S. Sindekar [10] statik SVC ve TCSC modelleri ile reaktif güç kontrolü ve iletim kaybının azaltılması üzerinde çalışmıştır. Algoritma Newton-Raphson tekniği kullanılarak güç akışını kontrol etmek için IEEE 9 baralı ve IEEE 14 baralı test sistemi üzerinde test edilmiştir.

Paul Clee ve arkadaşları [11] TCSC ACTS kontrolörü kullanılarak güç akışının artırılması için bir model önermiştir. TCSC kapasitansındaki değişiklik ile TCSC endüktansındaki değişim, TCSC'nin tek bir entegre modelinde Matlab / Simulink blok temsili kullanılarak tartışılmıştır.

Rahul Agrawal ve arkadaşları [12] Parçacık Sürüsü Optimizasyonu tekniği kullanılarak iletim kaybını en aza indirmek için Tristör Kontrollü Seri Kapasitörün (TCSC) güç sisteminde optimum konumunu tartışmıştır. Simülasyonlar TCSC dahil olarak Newton Raphson yük akış yöntemi IEEE 14 baralı sistemi, IEEE 30 baralı sistemi ve Hindistan 75 baralı sistemi üzerinde gerçekleştirilmektedir.

1.3. Transfer Kapasitesini Artırma Yöntemleri

İletim sisteminin kapasitesi aşağıdaki yöntemlerle geliştirilebilir:

1. Büyük boyutlu iletkeni montaj ederek ve terminal ekipmanını değiştirerek yüksek güç taşıma kapasitesini artırabilmektedir.
2. Ek hat sağlayarak yüksek akım taşıma kapasitesini artırabilmektedir
3. Seri kompanzasyonu sağlamak için uzun iletim hatlarında seri kapasitör kullanarak sisteminin iletim kapasitesini artırabilmektedir.
4. Faz açısı düzeltilmesi için regülatörleri montaj ederek sisteminin kapasitesini artırabilmektedir.
5. İletim hattına farklı FACTS cihazlarını montaj ederek sisteminin iletim kapasitesini artırabilmektedir.

1.4. İletim Sistemlerinde Güç Kayıplarını Azaltma Yöntemleri

Elektirik enerjisi, dünyadaki en yoğun tüketilen enerjidir. Bu, iklim koşullarının yanı sıra endüstri ve sosyal tüketimden kaynaklanmaktadır. Son araştırmalara göre, dünya genelinde enerji tasarrufu önemli ölçüde elektrik kullanımını yarıya düşürebilir. Bu nedenle iletim şebekelerinde optimum çalışma modunu belirlemek için gerekli stratejik yöntem ve güç kayıplarını azaltma tedbirlerin geliştirilmesi çok önemlidir. Güç kayıplarını azaltmak için farklı yöntemler vardır ve temel olarak bu yöntemler üç gruptan oluşmaktadır. Elektrik sistemini iyileştirme yöntemleri organizasyonel ve tekniksel olarak tanımlanmaktadır ve organizasyonel durumlar aşağıdakilerden oluşmaktadır:

1. Çift elektrik güç kaynağı ile güç kaybı olan konumların bulunması.
2. Dengesiz faz yükünün azaltılması.
3. Güç transformatörlerinin optimizasyonu.

Elektrik sisteminin iyileştirmesini tekniksel durum olarak tanımlarsak bu durum özel ekipman kullanma anlamına gelir, güç kayıplarını azaltmak için iletim primer hatlarına bir şönt kapasitör bankası kurulabilir, ayrıca dengeli olmayan bir faz yükünü azaltmak için harmonik bir filtre kurulabilmektedir.

1.5. Esnek AC İletim Sistemleri

FACTS cihazı şu şekilde tanımlanabilir: “Kontrol edilebilirliği ve güç aktarma kapasitesini artırmak için bir veya daha fazla AC iletim sistemi parametresinin kontrolünü sağlayan güç elektronik tabanlı sistem ve diğer statik ekipman”.

İletim sisteminin gerilim kararlılığını, güç akış kontrolünü, iletim sistemi güvenliğini iyileştirmek için, Esnek AC İletim Sistemi (FACTS) cihazları kullanılabilir. Çünkü FACTS cihazları güç aktarma kapasitesini artırır, sistemdeki toplam güç kaybını en aza indirir ve sistemin verimliliğini artırır. Bu çalışmada, Tristör Kontrollü Seri Kapasitör (TCSC), iletim sistemindeki güç aktarma kapasitesini artırmak ve güç kaybını optimize etmek için uygulanmıştır.

FACTS kontrol birimi şu şekilde sınıflandırılabilir:

1. Seri bağlı kontrol birimi.
2. Şönt bağlı kontrol birimi.
3. Kombine seri kontrol birimi.
4. Kombine seri şönt kontrol birimi.

1.5.1. Seri Bağlı Kontrol Birimi

Seri kontrol birimi, kapasitör, reaktör vb. gibi değişken empedans veya istenen ihtiyacı karşılamak için ana frekans, alt senkron ve harmonik frekanslar gibi güç elektroniği tabanlı değişken bir kaynaktır.

- Prensipte olarak, tüm seri kontrolörler hat ile seri gerilim enjekte eder.
- Gerilim hat akımıyla aynı fazda ise, seri kontrolör yalnızca değişken reaktif güç sağlar.

Örnekler: Statik Senkron Seri Kapasitör (SSSC), Tristör kontrollü Seri Kapasitör (TCSC), Tristör Kontrollü Seri Reaktör (TCSR). Bunlar İletim sisteminde, sistemdeki akımı, güç akışını kontrol etme ve sistemi osilasyonunu iyileştirme amaçlı kullanılabilirler.

1.5.2. Şönt Bağlı Kontrol Birimi

- Kontrol cihazları değişken empedans, değişken kaynak veya kombinasyonundan ulaşabilmektedir.
- Prensipte olarak, tüm şönt kontrolörleri sistem bağlantı noktasına akım enjekte eder.
- Enjekte edilen akım hat gerilimiyle aynı fazda ise, şönt kontrol cihazı yalnızca değişken reaktif güç sağlar.
- Örnekler: Statik Senkron generatör, Statik VAR Kapasitör.

1.5.3. Kombine Seri Kontrol Birimi

- Bu çok hatlı bir iletim sisteminde ayrı seri kontrolörlerin bir kombinasyonudur.
- Seri kontrolörler, her hat için bağımsız seri reaktif kompanzasyon sağlar ancak hatlar arasında aktif gücü güç bağlantısı üzerinden aktarır.
- Seri kontrolörler her hat için bağımsız seri reaktif kompanzasyon sağlar, ancak aynı zamanda güç bağlantısı aracılığıyla hatlar arasında aktif gücü aktarır.
- Aktif güç aktarma özelliği, hatlardaki hem aktif hem de reaktif güç akışını dengelemeyi ve böylece iletim sisteminin kullanımını en üst düzeye çıkarmayı mümkün eder. Bu tip kontrolöre örnek olarak Hat İçi Güç Akışı Kontrolörü (IPFC) önerilmektedir.

1.5.4. Kombine Seri Şönt Kontrol Birimi

- Bu koordineli şekilde kontrol edilen ayrı şönt ve seri kontrolörlerin bir kombinasyonu veya seri ve şönt elemanlı birleşik bir güç akışı kontrolörüdür.
- Prensipte olarak, kombine şönt ve seri kontrolörler, kontrolörün şönt kısmı ile sisteme akım enjekte eder ve kontrolörün seri kısmı ile aynı doğrultuda seri olarak voltaj verir.
- Prensipte olarak kombine şönt ve seri kontrolörler, kontrolörün şönt kısmı ile sisteme akım ve kontrolörün seri kısmına ile gerilim enjekte eder.
- Ancak şönt ve seri kontrolör birleştirildiğinde, güç bağlantısı üzerinden seri ve şönt kontrolörler arasında aktif güç değişimi olabilir.

- Bu tür denetleyicilere örnek olarak UPFC ve Tristör Kontrollü Faz Kaydırma Transformatörü (TCPST) önerilebilir.

1.6. ACTS Cihazlarının Uygulamaları

Güç sistemindeki iletim hattı performansını artırmak için FACTS cihazları kullanılır. FACTS birden fazla kriter altında sınıflandırılabilir: uygulama, teknoloji, güç sistemine bağlantı tipi, MVA başına kurulum maliyeti, FACTS'in dinamik yanıtı ve diğer bazı hususlar vb. FACTS kontrolörleri, geleneksel cihazlarla kıyaslanarak güç sisteminin kararlı durumu, geçici durumu ve geçici sonrası durumu gibi üç durumda kullanılabilir.

1.6.1. Kararlı Durum Uygulamaları

Kararlı durum uygulamalarından bazıları, kararlı durum gerilim kontrolü, termal yükün artması, acil durum gerilim kontrolü ve güç akışı kontrolü ve döngü kontrolüdür. Genel olarak, gerilim kontrolü için SVC ve STATCOM tercih edilir, fakat döngü kontrolü ve güç akışı kontrolü için TCSC kullanılır.

1.6.2. Yükleme Marjı İyileştirme Uygulaması

Dünya genelindeki mevcut Elektrik şebekelerinde temel çıkışlarının amacı, maksimum yük altında gerilimin çökmemesidir. Bu nedenle, seri kapasitörleri, güç sistemindeki maksimum kapasitede gücü verimli aktarmak için kullanılır.

1.6.3. Güç Akışı ve Dengeleme Kontrolü Uygulaması

İletim ve dağıtım hatlarındaki güç akışını kontrol ederken, FACTS kontrolörleri çok önemli rol oynar. Bunu göz önünde bulundurarak, TCSC, SSSC, UPFC iletim hatlarındaki güç akışını maksimum seviye üzerinde sağlamak için ve son kullanıcıdaki verimli kullanım gücünde artırmak için kullanılmaktadır.

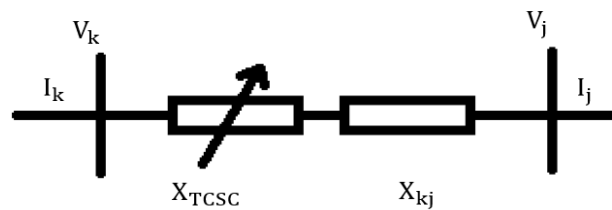
1.7. İletim Kapasitesinin Geliştirilmesi İçin FACTS Cihazlarının Konumu

FACTS cihazlarının yerleşimi çok önemlidir ve FACTS cihazının yerleştirilmesinin temel amacı güç sisteminde maksimum tıkanıklık giderilmesidir. Cihazlar, deneme yanılma yöntemiyle belirlenen konumlara yerleştirilebilir. İletim ağındaki FACTS cihazları için en uygun yeri seçmek için birçok yöntem önerilmektedir [13].

Hat kayıplarını en aza indirmek için, hassas yaklaşıma dayalı seri kapasitörler, faz kaydırıcılar ve statik VAR kapasitörleri en uygun yere yerleştirilmesi önerilmiştir [14]. Optimizasyonunda, FACTS cihazları ile optimum güç akışı için farklı Amaç fonksiyonlar uygulanmıştır [15]. FACTS cihazlarının maliyetini göz önünde bulundurarak dağıtım şebekesinde FACTS cihazlarının en uygun konumlarını bulma çalışması yapılmıştır.

1.8. TCSC'nin Modellenmesi

TCSC, seri kapasitör tarafından tristör kontrollü ve şöntlü bir indüktör ile paralel bir seri kapasitör bankından oluşan kapasitif reaktans kapasitör olarak tanımlanır. Tristör Kontrollü Seri Kapasitör (TCSC), aynı iletim hattının görünür empedansını değiştirerek iletim hattının güç aktarım seviyesini artırmak ve kontrol etmek için güçlü bir işlev sağlar. İletim sistemi kayıpları ve gerilim kontrolü için önemli bir bileşen olarak, genellikle kademe ayar transformatörünün bağlı olmadığı hatta mont edilir. Aşırı kompanzasyonu önlemek için TCSC'nin çalışma aralığı $-0.8X_{line}$ ile $0.2X_{line}$ arasında seçilir. TCSC'nin kontrol edilebilir bir reaktansı X_{TCSC} olarak modellenmesi Şekil 1'de gösterilmiştir [16].



Şekil.1 TCSC'nin değişken reaktans modeli

İletim hattının toplam reaktansı matematiksel olarak şu şekilde ifade edilebilir:

$$X_{line} = X_{TCSC} + X_{kj} \quad (1)$$

Burada,

X_{kj} : “k” ve “j” baralar arasındaki iletim hattının reaktansı.

X_{TCSC} : TCSC’nin reaktansı

X_{line} : TCSC yerleştirildikten sonra hattın toplam reaktansı.

TCSC’nin “a” hattının telafi derecesi açısından reaktansı:

$$X_{TCSC} = a \times X_{kj} \quad (2)$$

Denk. (1) şu şekilde değiştirilir:

$$X_{line} = a \times X_{kj} + X_{kj} \quad (3)$$

$$X_{line} = (a+1) \times X_{kj} \quad (4)$$

TCSC şebekede bazen endüktif bazen kapasitif çalışmaktadır bu nedenle bu çalışma sunuğunda TCSC büyüklüğü bazen pozitif bazen negatif olarak optimize edilmiştir.

1.9. Amaç Fonksiyonu

Güç akışı elektrik güç sisteminin önemli bir aracıdır. Toplam güç kaybını en aza indiren ve tüm baralarda gerilim profilini koruyan, TCSC’nin yerini ve boyutunu içeren sonucu bulmak için Newton Raphson yük akış yöntemi kullanılarak çok amaçlı bir işlev yapılmıştır.

1.9.1. Şebekedeki Aktif Güç Kaybı

İletim kayıplarının en aza indirilmesi, Şebekede ekonomik elektrik enerji gücü sağlar. Kayıpları en aza indirmek için elektrik güç sistemindeki toplam güç kaybı matematiksel olarak denklem 5’te verildiği gibi hesaplanır ve sonra PSO ile optimize edilir [17].

$$\min f_1 = P_{loss} = \sum_{k=1}^{nline} G_{kj} \left[V_k^2 + V_j^2 - 2V_k V_j \cos(\delta_k - \delta_j) \right] \quad (5)$$

$j \neq 1$

$nline$: toplam iletim hattı sayısı.

G_{kj} : “k” ve “j” bara arasındaki iletkenlik.

V_k : “k” barasının gerilimi.

V_j : “j” barasının gerilimi.

δ_k : “k” barasının gerilim açısı.

δ_j : “j” barasının gerilim açısı.

1.9.2. Gerilim Profilini İyileştirme

İkinci amaç fonksiyonu, verilen gerilim profilinin en iyi seviyede olması hedeflenerek formül 6’de verildiği gibi optimize edilmiştir.

$$\min f_2 = \sum_{(i=1)}^n \left| V_{kref} - V_k \right| \quad (6)$$

V_k : bara k daki gerilim genliği.

V_{kref} : bara k daki referans gerilim değeridir.

n : yük bara sayısıdır.

1.9.3. TCSC Maliyet Fonksiyonu

TCSC cihazlarının kurulum maliyeti, aşağıdaki denklemle matematiksel olarak formüle edilmiştir [17,18]:

$$I_C = C_{TCSC} \times S \times 1000 \quad (7)$$

I_C : TCSC cihazları kurulum maliyeti [US \$].

TCSC cihazının kurulum maliyet fonksiyonu aşağıdaki Formula kullanarak hesaplanabilir [19, 20]:

$$C_{TCSC} = 0.0015S^2 - 0.7130S + 153.7 \left[\frac{US\$}{KVA_r} \right] \quad (8)$$

$$S = |Q_2 - Q_1| \quad (9)$$

Burada:

C_{TCSC} : [US \$ / KVar] cinsinden TCSC cihazlarının maliyeti.

S : TCSC'nin çalışma aralığı [MVar].

$Q1$: TCSC kurulumundan önceki reaktif güç akışı.

$Q2$: TCSC kurulumundan sonraki reaktif güç akışı.

Matematiksel ifadeyle çok amaçlı optimizasyon problemi kondisyon fonksiyonu olarak ifade edilen tek bir amaç optimizasyon problemine dönüşür.

$$J = h_1 f_1 + h_2 f_2 + h_3 f_3 \quad (10)$$

h_1 , h_2 ve h_3 , 0 ile 1 arasında seçilen ölçeklendirme faktörüdür. f_1 , f_2 ve f_3 bu çalışmada kullanılan amaç fonksiyonlarıdır. Burada f_1 , aktif güç kaybını f_2 , gerilim profili, f_3 , maliyet fonksiyonunu ifade etmektedir.

1.10. Yük Akışı Analiz Yöntemleri

En yaygın kullanılan yük akışı yöntemleri Gauss-Siedel yöntemi, Newton-Raphson yöntemi ve Hızlı Dekuplaj yöntemi [21] olarak tanımlanmıştır.

1.10.1. Gauss-Siedel (GS) Yük Akışı Yöntemi

Gevşek bara gerilimi varsayıldığında (genellikle $V_1 = 1 \angle \theta^0 p.u$), kalan $(n-1)$ bara gerilimi yinelemeli süreçte aşağıdaki gibi bulunur [22]:

$$P_k = \sum_{j=1}^n |V_k| |V_j| |Y_{kj}| \cos(\theta_{kj} - \delta_k + \delta_j) \quad (11)$$

$$Q_k = -\sum_{j=1}^n |V_k| |V_j| |Y_{kj}| \sin(\theta_{kj} - \delta_k + \delta_j) \quad (12)$$

(11) ve (12) denkleminde statik yük akış denklemleri denir.

$$I_k = (P_k - jQ_k) / (V_k^*) \quad (13)$$

$$V_k = \frac{1}{Y_{kk}} \left(I_k - \sum_{\substack{j=1 \\ j \neq k}}^n Y_{kj} V_j \right) \quad k = 2, 3, 4, \dots, n \quad (14)$$

$(a + 1)^{th}$ iterasyon için gerilim denklemi

$$V_k^{(a+1)} = \frac{1}{Y_{kk}} \left[\frac{P_k - jQ_k}{(V_k^a)} - \sum_{j=1}^{i-1} (Y_{kj} V_j^{a+1}) - \sum_{j=i+1}^n (Y_{kj} V_j^a) \right] \quad (15)$$

1.10.2. Hızlı Ayrılmış Yük Akışı (FDLF) Yöntemi

FDLF yöntemi yakınsama geometriğidir, pratik doğruluklar için normal olarak 2 ila 5 iterasyon gereklidir, FDLF iterasyonları hızı NR yönteminin yaklaşık beş katı veya GS yönteminin yaklaşık üç katıdır [23]. Burada B_{ii} , Y_{bus} 'nin diagonal elemanlarının sanal kısmı; B'_{ii} , Y_{bus} 'nin diagonal dışı elemanlarının sanal kısmı; θ_{ii} , Y_{bus} 'un Y_{ii} elemanının açısı; B' , $(n - 1) \times (n - 1)$ düzeyindeki Y_{bus} 'un sanal kısmı; B'' , $(n - 1 - n_{pv}) \times (n - 1 - n_{pv})$ düzeyindeki Y_{bus} 'un sanal kısmı.

Diagonal elementi J_1 aşağıdaki gibi tanımlanmıştır:

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{j=1}^n |V_i| |V_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i - \delta_j) - |V_i|^2 |Y_{ii}| \sin \theta_{ii} \quad (16)$$

Yukarıdaki denklemin (16) ilk terimini $-Q_i$ ile değiştirerek aşağıdaki denklem oluşturulmuştur.

$$\frac{\partial P_i}{\partial \delta_i} = -Q_i - |V_i|^2 |Y_{ii}| \sin \theta_{ii} \quad (17)$$

$$\frac{P_i}{\delta_i} = -Q_i - |V_i|^2 B_{ii} \quad (18)$$

$$B_{ii} \gg Q_i, \text{ ile } |V_i|^2 \approx |V_{ii}| \quad (19)$$

$$\frac{\partial P_i}{\partial \delta_i} = -|V_i| B_{ii} \quad (20)$$

Benzer şekilde,

$$\frac{\partial P_i}{\partial \delta_i} = -|V_i| B_{ij} \quad (21)$$

Diagonal elementi J_4 aşağıdaki gibi tanımlanmıştır:

$$\frac{\partial Q_i}{\partial |V_i|} = \sum_{j=1}^n |V_i| \|V_j\| Y_{ij} |\sin(\theta_{ij} - \delta_i - \delta_j)| - |V_i| \|Y_{ii}\| \sin \theta_{ii} \quad (22)$$

$$\frac{\partial Q_i}{\partial |V_i|} = Q_i - |V_i| \|Y_{ii}\| \sin \theta_{ii}, \text{ nedenle } B_{ii} = Y_{ii} \sin \theta_{ii} \gg Q_i \quad (23)$$

$$\frac{\partial Q_i}{\partial |V_i|} = -|V_i| B_{ii} \quad (24)$$

Varsayarak $\theta_{ij} - \delta_i + \delta_j \approx \theta_{ij}$, verim

$$\frac{\partial Q_i}{\partial |V_i|} = -|V_i| B_{ij} \quad (25)$$

Yeni denklem aşağıdaki gibidir,

$$\frac{\Delta P}{|V_i|} = -B' \Delta \delta \quad (26)$$

$$\frac{\Delta Q}{|V_i|} = -B'' \Delta |V| \quad (27)$$

Şimdi $\Delta \delta$ ve $\Delta |V|$ tarafından elde edilmiştir.

$$\Delta \delta = -[B'] \frac{\Delta P}{|V|} \quad (28)$$

$$\Delta |V| = -[B'] \frac{\Delta Q}{|V|} \quad (29)$$

1.10.3. Newton Raphson Yük Akış Yöntemi

Newton Raphson yük akış yöntemi, Doğrusal olmayan denklemleri çözmek için yinelemeli bir tekniktir. Newton Raphson Yöntemi kullanılarak yük akışı için iki çözüm yöntemi vardır. İlk yöntemde değişkenler için dikdörtgen koordinatlar kullanılarak, ikinci yöntemde kutupsal koordinat formu kullanılarak yapılır. Kutupsal koordinat formu yaygın olarak kullanılmaktadır.

Bu yöntem aşağıdaki denklemlerle gerçekleştirilmektedir.

$$S_i = P_i + jQ_i = V_i \sum_{k=1}^n V_{ik} V_k \quad (30)$$

S_i : i barasındaki karmaşık güç

P_i, Q_i : i barasındaki aktif ve reaktif güç

V_i : i barasındaki gerilim

V_{ik} : i,k barasındaki gerilim

V_k : k barasındaki gerilim

$$S_i = \sum_{k=1}^n (V_i V_k Y_{ik} / \theta_{ik} + \delta_i - \delta_k) \quad (31)$$

Y_{ik} : i,k barasındaki admitansi

δ_i : i barasındaki gerilim açısı

δ_k : k barasındaki gerilim açısı

$$P_i = \sum_{k=1}^n (V_i V_k Y_{ik}) \cos(\theta_{ik} + \delta_i - \delta_k) \quad (32)$$

$$Q_i = \sum_{k=1}^n (V_i V_k Y_{ik}) \sin(\theta_{ik} + \delta_i - \delta_k) \quad (33)$$

P_i için ayrı olarak $i=k$ ve $i \neq k$ olarak hesaplama yapılması. Bununla Yukarıdaki denklem (32) ve (33) aşağıda gösterildiği gibi de yazılabilir.

$$P_i = V_i^2 Y_{ii} \cos \theta_{ii} + \sum_{k=1, k \neq i}^n (V_i V_k Y_{ik}) \cos(\theta_{ik} + \delta_i - \delta_k) \quad (34)$$

$$Q_i = V_i^2 Y_{ii} \sin \theta_{ii} + \sum_{k=1, k \neq i}^n (V_i V_k Y_{ik}) \sin(\theta_{ik} + \delta_i - \delta_k) \quad (35)$$

Denklem 34 ve 35, i barası üzerinde hatta giren net aktif güç P_i ve reaktif güç Q_i için hesaplanan değerleri sağlayan güç akışı denklemlerinin kutupsal biçimini oluşturur.

P_i 'nin hesaplanan değerlerinin P_{ical} ve Q_i 'nin Q_{ical} ile gösterilmesi, ΔP_i ve ΔQ_i uyumsuzluklarının tanımlanmasını sunar.

$$\Delta P_i = P_{isch} - P_{ical} \quad (36)$$

$$\Delta Q_i = Q_{isch} - Q_{ical} \quad (37)$$

$$P_{isch} = P_{gi} - P_{di} \quad (38)$$

$$Q_{isch} = Q_{gi} - Q_{di} \quad (39)$$

P_{isch} ve Q_{isch} , Bara i'de planlanan enjekte edilmiş net aktif ve reaktif güçtür.

P_{gi} ve Q_{gi} , Bara i'de planlanan üretim aktif ve reaktif gücünü ifade eder.

P_{di} ve Q_{di} , Yük barası i'de planlanmış aktif ve reaktif güç talebini belirtir.

P_{ical} ve Q_{ical} , Planlanan değerlerle çakışmadığında uyumsuzluklar meydana gelir.

Bara gerilimlerinin aktif ve reaktif güç bileşenlerindeki değişimler arasındaki ilişkiyi ifade eden lineer denklemler seti kompakt biçimde şu şekilde yazılabilir:

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{11} & J_{12} \end{bmatrix} \begin{bmatrix} \Delta \delta_i \\ \Delta V_i \end{bmatrix} \quad (40)$$

Jacobian matrisi J_{11} , J_{12} ... şeklinde konvansiyonel olarak bölünmüş bir matris kurulur.

$$\begin{bmatrix} H & N \\ M & L \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |V|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |V|} \end{bmatrix} \quad (41)$$

H, N, M ve L alt matrislerinin diyagonal dışı ve diyagonal elemanları, δ_i ve $|V_i|$ 'ye göre denklem (34) ve (35) türevleri alınarak belirlenir.

Yukarıdaki adımda hesaplanan $\Delta \delta_i$ ve $\Delta |V_i|$ değerlerinin kullanarak, aşağıda gösterilen denklemlerle tüm yük baralarındaki gerilim büyüklüğünü ve faz açısını değiştirir.

$$\left| V_i^{(r+1)} \right| = \left| V_i^{(r)} \right| + \Delta \left| V_i^{(r)} \right| \quad (42)$$

$$\delta_i^{(r+1)} = \delta_i^{(r)} + \Delta \delta_i^{(r)} \quad (43)$$

$V_i^{(r+1)}$: iterasyon r+1 üzerindeki bara i gerilim değeri

$\delta_i^{(r+1)}$: iterasyon r+1 üzerindeki bara i gerilim açısı

Bir sonraki yineleme döngüsünü $|V_i|$ ve δ_i 'nin değiştirilmiş değerleriyle başlatın.

Tüm yük baraları için planlanan hatalar belirtilen tolerans dahilinde olana kadar devam edin.

$$\Delta P_i^{(r)} < \varepsilon, \Delta Q_i^{(r)} < \varepsilon \quad (44)$$

Burada ε : yük baraları için tolerans seviyesini gösterir, r iterasyon sayısını temsil eder.

Son olarak Gauss Seidel yönteminde olduğu gibi hattın güç akışını hesaplayarak Newton Raphson yük akış yöntemini tamamlayın.

Newton Raphson yük akış yönteminin avantajları şunlardır:

- Yakınsama özelliklerine sahiptir. Bu nedenle yakınsama çok hızlıdır.
- Yineleme sayısı, sistemin boyutundan bağımsızdır. Hem küçük hem de büyük sistemler için neredeyse her zaman iki ila üç yinelemede yüksek doğrulukta çözümler elde edilir.
- Newton Raphson yük akış yönteminin yakınsaması gevşek veri yolu seçimine duyarlı değildir.
- Genel olarak, daha az sayıda yineleme gerektiğinden hesaplama süresinde bir tasarruf vardır.

Çeşitli sınırlamalar aşağıda verilmiştir.

- Bu çözüm tekniği zordur.
- Her yineleme için Jacobian öğelerinin hesaplanması daha uzun zaman alır.
- Bilgisayarın bellek gereksinimi büyük olması gerekmektedir.

1.11. Optimizasyon Algoritması

Optimizasyon çeşitli çalışmalarda kullanılmaktadır ve bu nedenle çok çeşitli uygulamalarda önemli bir paradigmadır. Mühendislik ve endüstrideki neredeyse tüm uygulamalarda, maliyeti ve enerji kaybını en aza indirmek ve verimliliği en üst düzeye çıkarmak için her zaman bir konuyu optimize etmekteyiz. Kaynaklar, zaman ve ekonomik açısından optimizasyon çok önemlidir [24].

Yapıların ve sistemlerin doğruluğu ve giderek artan karmaşıklığı için, simülasyon sürecinin gittikçe daha fazla zaman alıcı olmasına neden olmaktadır. Birçok mühendislik alanında, tek bir tasarımın değerlendirilmesi birkaç gün hatta hafta kadar sürebilir. Simülasyon süresini ve optimizasyon sürecini hızlandırabilecek yöntem kullanarak zamandan ve ekonomiden tasarruf sağlayabilmekteyiz.

Simülasyonlar genellikle en çok zaman alan bölümdür. Birçok uygulamada, bir optimizasyon süreci genellikle amaç fonksiyonun birçok kez, hatta binlerce ve hatta milyonlarca konfigürasyonun değerlendirilmesini içerir. Bu tür değerlendirmeler genellikle bir hesaplamalı akışkanlar dinamiği simülatörü veya sonlu bir eleman çözücü gibi kapsamlı hesaplama araçlarının kullanımını içerir. Bu nedenle verimli bir optimizasyon son derece önemlidir. Optimizasyon sorunları birçok şekilde formüle edilebilir. Şimdiye kadar en yaygın doğrusal olmayan bir optimizasyon problemi aşağıdaki gibi ifade edilebilmektedir:

$$\text{minimize } f_i(x), (i = 1, 2, \dots, M) \quad (45)$$

$$h_j(x) = 0, (j = 1, 2, \dots) \quad (46)$$

$$g_k(x) \leq 0, (k = 1, 2, \dots) \quad (47)$$

Burada f_i , h_j ve g_k genel olarak doğrusal olmayan fonksiyonlardır. Burada $x = (x_1, x_2, \dots, x_n)$ tasarım vektörüdür, Ayrıca sürekli, ayık veya n boyutlu uzayda karışık olabilmektedir. $f_i(x)$ fonksiyonlarına Amaç fonksiyonları denir ve $M > 1$ olduğunda, optimizasyon çok kriterli olur [25]. Farklı hedefleri tek bir hedefte birleştirmek mümkündür, ancak çok kriterli optimizasyon soruna çok daha fazla bilgi ve fikir uygulanabilmektedir. Burada sorunu bir minimize problemi olarak yazdığımızı belirtmek gerekirse, sadece optimizasyon fonksiyonunu $-f_i(x)$ ile değiştirilerek bir maksimizasyon olarak da yazılabilir.

1.11.1. Optimizasyonda Sorunlar

Simülasyon odaklı optimizasyon ve modellemede üç ana sorun vardır ve bunlar,

1. Bir algoritmanın verimliliği.
2. Sayısal bir simülatörün verimliliği ve doğruluğu.
3. Doğru soruna doğru algoritmanın kullanılması.

Önemlerine rağmen, bu tür konular için tatmin edici bir kural veya kılavuz yoktur. Açıkçası, mevcut en verimli algoritmaları kullanmayı denemekteyiz. Ancak bir algoritmanın gerçek verimliliği, bir algoritmanın iç çalışması, ihtiyaç duyulan bilgiler (amaç fonksiyonlar ve türevleri) ve uygulama ayrıntıları gibi birçok faktöre bağlı olabilir. Bir çözücünün verimliliği, kullanılan sayısal yöntemlere ve ilgilenilen sorunun karmaşıklığına bağlı olarak daha da karmaşıktır. Doğru problemler için doğru algoritmaları seçmeye gelince, birçok ampirik gözlem vardır, ancak üzerinde bir kılavuz yoktur. Aslında, tüm problem türleri için evrensel olarak etkili bir algoritma yoktur. Bu nedenle, seçim birçok faktöre ve hedefe bağlı olmaktadır.

1.11.2. Doğru Algoritma Seçimi

Optimizasyon açısından bakıldığında, belirli bir sorun için doğru optimize edici algoritma seçimi çok önemlidir. Bir optimizasyon görevi için seçilen algoritma büyük ölçüde sorunun türüne, bir algoritmanın doğasına, istenen çözüm kalitesine, mevcut bilgi işlem kaynağına, zaman sınırına, algoritma uygulamasının kullanılabilirliğine ve kararın uzmanlığına bağlı olacaktır.

Bir algoritmanın doğası genellikle belirli bir sorun türü için uygun olup olmadığını belirler. Çözmeye çalıştığımız sorunun türü de seçtiğimiz algoritmaları belirler. Eldeki bir optimizasyon probleminin Amaç işlevi doğrusal olmayan ve çok modelli ise, klasik algoritmalar uygun değildir. Bu durumda, parçacık sürü optimizasyonu ve Genetik algoritması gibi optimizasyonlar en uygun olanlardır.

1.11.3. Genetik Algoritması

Genetik algoritma (GA), doğal seleksiyon ve genetiğin birleşimine dayanan bir arama algoritmasıdır. Genetik algoritmanın özellikleri çeşitli yönlerden diğer arama tekniklerinden farklıdır. Algoritma çok yolludur, paralel olarak birçok tepe noktası arar ve böylece yerel minimum yakalama olasılığını azaltır. Ayrıca GA parametrelerin kendileri yerine parametrelerin kodlanmasıyla çalışır. Parametre kodlaması, genetik operatörün mevcut durumu minimum hesaplamalarla bir sonraki duruma geçirmesine yardımcı olacaktır.

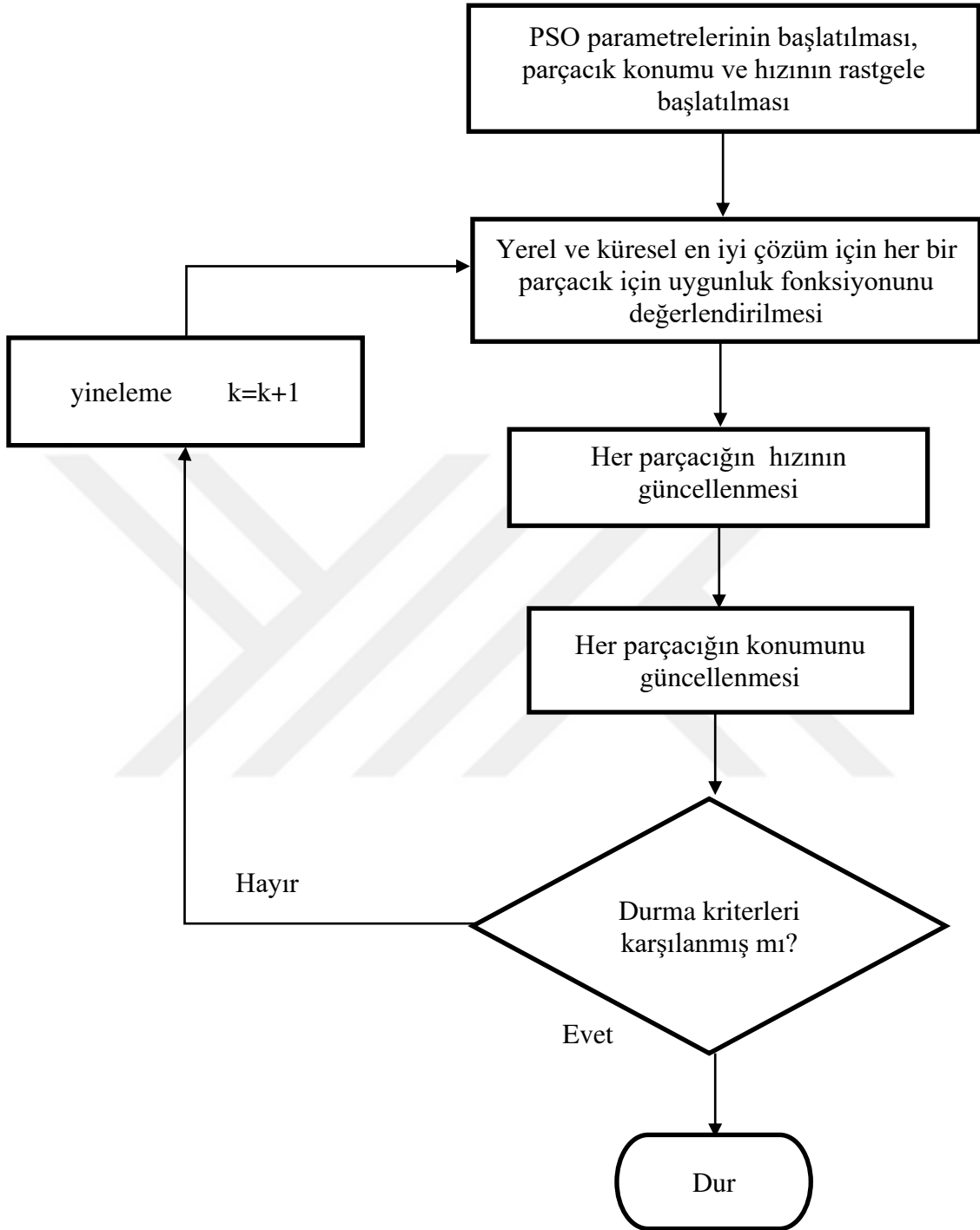
Genetik algoritma, optimizasyon işlevi yerine aramasına rehberlik etmek için her dizinin uygunluğunu değerlendirir. GA'nın araştırmasına rehberlik etmek için yalnızca nesnel işlevi (uygunluk) değerlendirmesi gerekir. Dolayısıyla, türevlerin veya diğer yardımcı fonksiyonların hesaplanmasına gerek yoktur.

Genetik algoritma optimizasyonunun başlangıcında, bir dizi karar değişkeni çözümü popülasyonun üyeleri olarak kodlanır. Çözüm öğelerini kodlamanın birden çok yolu vardır. Üremeye dayalı çaprazlama ve mutasyon operatörleri, yeni nesil popülasyon yaratılması için kullanılır. Gelecek neslin bir üyesini oluşturmak için mevcut nesildeki çözüm unsurlarını birleştirir. Mutasyon, gelecek neslin bir üyesini oluşturmak için bir çözümün öğelerini mevcut nesilden sistematik olarak değiştirir. Geçiş ve mutasyon, gelecek neslin üyelerinde çeşitlilik yaratarak arama alanını keşfetmeyi başarıyor. GA'ların geleneksel kullanımları, bu algoritmaların küresel bir minimum bulmak için arama alanının birden fazla alanını keşfetmesinden kaynaklanmaktadır. Geçiş operatörünü kullanan bu algoritmalar, tek bir global çözüm bulmak için farklı çözümlerden en iyi özellikleri birleştirme konusunda özellikle güçlüdür. Genetik algoritmalar, karmaşık, oldukça doğrusal olmayan alanları aramak için de uygundur.

1.11.4. Parçacık Sürüsü Optimizasyonu Algoritması

Parçacık sürüsü optimizasyonu, evrimsel hesaplamada, sistemin rasgele çözümlerden oluşan bir popülasyonla başlatılması bakımından biraz genetik bir algoritmaya benzeyen yeni bir metodolojidir. Parçacık sürüsü optimizasyonunun çok çeşitli mühendislik problemlerinin çözümünde son derece etkili olduğu bulunmuştur. Sorunları çok hızlı bir

şekilde uygulamak ve çözmek bu algorithmada kolaydır. PSO sisteminde, grup tüm parçacıklardan oluşan bir topluluktur ve tüm parçacıklar çok boyutlu bir arama alanında olur. Bu arama alanında, her bir parçacık kendi deneyimine ve komşu parçacıkların deneyimine göre konumunu ayarlayarak, kendisinin ve komşularının karşılaştığı en iyi konumu kullanır. Her parçacığın sürü yönü, parçacığa komşu parçacıklar kümesi ve tarihsel deneyimi ile tanımlanır. Parçacık sürüsü optimizasyonu genel olarak evrimsel hesaplama teknikleri ve GA ile birçok benzerliği paylaşmaktadır. Bu üç teknik de rastgele oluşturulmuş bir popülasyon grubuyla başlar ve hepsi popülasyonu değerlendirmek için bir uygunluk değeri kullanır. Popülasyonu günceller ve rastgele tekniklerle optimum arayışı ararlar. EC ve GA'ya kıyaslarsak PSO yaklaşımı arasındaki temel fark, PSO'nun çaprazlama ve mutasyon gibi genetik operatörlere sahip olmamasıdır. Parçacıklar kendilerini iç hız ile günceller; ayrıca algoritma için önemli bir hafızaya sahiptir. PSO algoritmasının akış şeması Şekil 2'de verilmiştir.



Şekil 2. PSO akış şeması

1.12. Farklı PSO Algoritmalar

1.12.1. Standart Parçacık Sürüsü Optimizasyonu (SPSO)

PSO algoritması sosyal psikolojik metafora dayalı uyarlanabilir bir algoritmadır; bir birey popülasyonu (parçacıklar olarak adlandırılır), daha önce başarılı olan bölgelere stokastik olarak geri dönerek uyum sağlar. Diğer evrimsel hesaplama algoritmalarında olduğu gibi, bireyleri manipüle etmek için evrimsel operatörleri kullanmak yerine, PSO'daki her bir birey, kendi deneyimine ve komşu parçacığın deneyimine göre dinamik olarak ayarlanmış bir hızla arama alanında hareket eder. Her birey, D boyutlu arama uzayında hacimsiz bir parçacık (bir nokta) olarak değerlendirilir.

Parçacığın önceki en iyi konumu (en iyi uygunluk değerini veren konum) kaydedilir ve aşağıdaki gibi temsil eder.

$$P_i (P_{i1}, P_{i2}, \dots, P_{id}) \quad (48)$$

Popülasyondaki tüm parçacıklar arasında en iyi parçacığın indeksi g sembolü ile temsil edilir. i Parçacığı için konum değişiminin (hız) oranı şu şekilde temsil edilir:

$$V_i (V_{i1}, V_{i2}, \dots, V_{id}) \quad (49)$$

Parçacıklar aşağıdaki denkleme göre manipüle edilir

$$V_i^{(k+1)} = w * V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \quad (50)$$

Burada c1 ve c2 iki pozitif sabittir. $rand_1()$ ve $rand_2()$, (0 1) aralığında iki rastgele işlemdir. w, sıfır olmayan bir atalet ağırlığıdır.

PSO'nun standart versiyonunda hız sınırlar getirilmedikçe salınım genişlenecektir bu nedenle hız sınırlama vermek gerekir.

$$\text{Eğer } V_{id} > V_{max}, \text{ ise } V_{id} = V_{max} \text{ sonuçta } V_{max} = 0.5P_{max} \quad (51)$$

$$\text{Yoksa } V_{id} < -V_{max}, \text{ ise } V_{id} = -V_{max} \text{ sonuçta } V_{max} = -0.5P_{max} \quad (52)$$

1.12.2. Atalet Ağırlığı Parçacık Sürüsü Optimizasyonu (IWPSO)

Atalet ağırlığı, parçacığın önceki yinelemede olduğu gibi aynı yönde hareket etmesi için bir tercih sağlar. Atalet ağırlığı, denklem 54'te verildiği gibi hesaplanır.

$$V_i^{(k+1)} = w * V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \quad (53)$$

$$w = \frac{w_{\max} - w_{\min}}{iter_{\max}} * iter \quad (54)$$

Burada $iter_{\max}$, maksimum yineleme sayısıdır ve $iter$ geçerli yineleme sayısıdır. Başlangıçta geliştirildiği gibi, w genellikle bir çalışma sırasında yaklaşık 0,9'dan 0,4'e doğrusal olarak azalır.

1.12.3. Daralma Katsayısı Parçacık Sürüsü Optimizasyonu (CCPSO)

Daralma Katsayısı PSO karmaşık doğrusal denklem kullanarak parçacık sürülerinin etkileşimlerini modeller. Daralma Katsayısı kullanırken aynı zamanda parçacıkların fazla yakınsaması ile sonuçlanır. Bu parçacığın salınımlarının genliği, yerel ve komşu önceki en iyi noktalara odaklandıkça azalır. Daralma katsayısı PSO aşağıdaki denklemlerle ifade edilmektedir.

$$V_i^{(k+1)} = K * \left[V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \right] \quad (55)$$

$$K = \frac{2k}{\left| 2 - \phi - \sqrt{\phi^2 - \phi^4} \right|} \quad (56)$$

$$\phi = c_1 + c_2 \quad (57)$$

Genellikle $k=1$ ve ϕ mutlaka $\phi > 4$ sonra $c_1 = 2.05, c_2 = 2.05$ olarak belirleyeceğiz.

1.12.4. Çok Fazlı PSO (MPPSO)

Çok fazlı PSO (MPPSO) yöntemi, parçacıkların ana sürülmesini alt sürülere veya alt gruplara ayırır, burada her alt sürgü farklı bir görev yapar, farklı bir davranış gösterir vb. Bir alt sürünün gerçekleştirdiği bu görev veya davranış genellikle zamanla değişir ve bilgi bu süreçte alt sürüler arasında aktarılır. MPPSO yöntemi ile ana sürüyü eşit büyüklükte iki

alt sürüye bölürler. Bu algoritmada, parçacıklar rastgele atanır ve her bir alt sürü, iki fazdan birinde olabilir. Bu aşamalar aşağıda tartışılmaktadır:

- Çekim fazı: bu aşamada, karşılık gelen alt sürünün parçacıkları küresel en iyi pozisyona doğru hareket eder.
- İtme fazı: bu aşamada, karşılık gelen alt sürünün parçacıkları küresel en iyi konumdan uzaklaşır. Bu nedenle, temel PSO'dan farklıdır.
- Parçacıklar, keşif alanının çeşitliliğini ve arama alanının genişlemesini arttırmak için birden fazla gruba ayrılır.
- Farklı yönlerine algoritma farklı aşamalar katar.

1.12.5. Dinamik Komşuluk PSO (DNPSO)

Bu algoritmada çoklu hedefler iki gruba ayrılır: f_1 ve f_2 , burada f_1 komşu hedefi olarak tanımlanır ve f_2 de optimizasyon hedefi olarak tanımlanır. Her yinelemede, her bir parçacık diğer tüm parçacıklara olan mesafeyi hesaplayarak ve en yakın komşuları seçerek dinamik olarak yeni bir parçacık toplumu belirler. Mesafe ilk Amaç fonksiyon grubu f_1 'in kondisyon değerleri arasındaki fark olarak tanımlanır. En iyi yerel değer, ikinci Amaç fonksiyon f_2 'nin uygunluk değeri açısından komşular arasından seçilir. Son olarak, küresel en iyi güncelleme sistemi sadece mevcut kişisel en iyi değere hâkim olan çözümü dikkate alır. Ana dezavantajı, sadece iki amaç için yararlı olmasıdır.

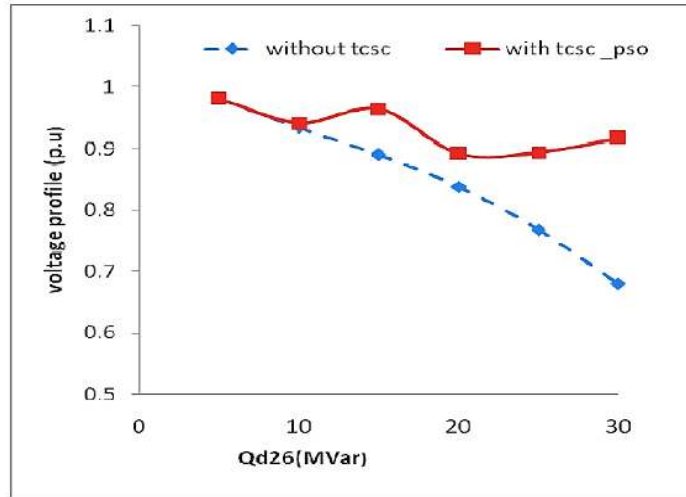
Yukarıda verilen genel PSO tanıtımından ve bu sistem özellikleri açısından Standart PSO (SPSO), Atalet Ağırlığı tabanlı PSO(IWPSO) ve Daralma katsayılı PSO deneme yapılmıştır sonuçlar tartışılmıştır.

2. YAPILAN ÇALIŞMALAR

Bu bölümde, IEEE-14 ve 30 baralı Sisteminde TCSC'nin optimum konumu için farklı PSO algoritmasının uygulaması tartışılmaktadır. Uygun görülen standart PSO, atalet ağırlığı PSO ve daralma katsayılı PSO kullanılmıştır. Özellikle literatürde kullanılmamış olan daralma katsayısı tabanlı PSO kullanılmıştır.

2.1. Sorun ve Çözüm

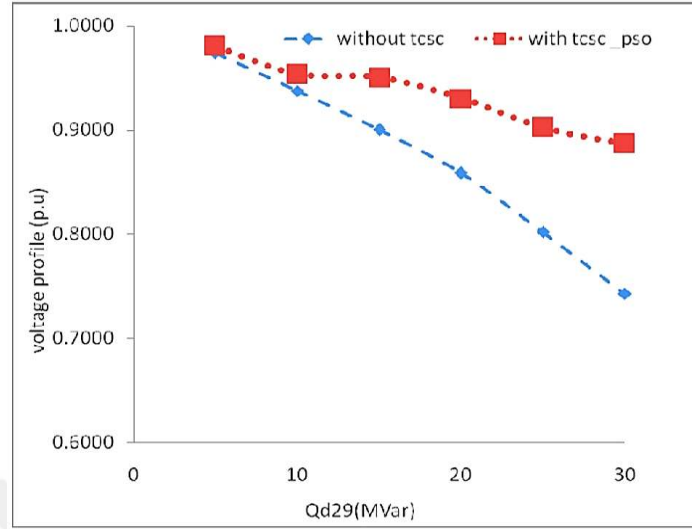
Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmasında atalet ağırlığı tabanlı PSO(IWPSO) kullanılmıştır ve IEEE 30 baralı sistem üzerinde bara 26'ye ve bara 29'a her seferinde 5 MVar'dan yük ekleyerek sistem gerilim profili oluşturmuştur. Gerilim profili şekil 4'te gösterilmiştir.



(a)

Şekil 3. IWPSO Algoritması gerilim performansı, Burada (a) Bara 26'ye yük ekledikten sonraki gerilim profili, (b) Bara 29'a yük ekledikten sonraki gerilim profili

Şekil 3'ün devamı



(b)

Şekil 3'te gerilim profiline bakarsak atalet ağırlığı PSO kullanılarak IEEE 30 baralı sistem üzerinde bara 26'ye ve 29'a yük eklediğimizde gerilimde osilasyon olduğunu görmekteyiz bu problemi çözmek için ve bu çalışmanın amacı olan güç kaybını daha iyi seviyede optimize etmek için literatürde kullanılmamış olan daralma katsayılı PSO kullanılmıştır ve atalet ağırlığı tabanlı PSO ile karşılaştırma yaparak birinci çalışma senaryosu oluşturulmuştur. İkinci çalışma senaryosunda IEEE 14 baralı ve IEEE 30 baralı sisteminde standart PSO, atalet ağırlığı PSO ve daralma katsayılı PSO'yu uygulayarak bütün sistemin gerilim performansı açısından bu çalışmada yeni olarak kullanılan daralma katsayılı PSO'nun uygunluğu denetlenmiştir.

2.2. Kullanılan Metodun Test Sisteminde Ayarları

Bu bölümde, standart PSO, atalet ağırlığı tabanlı PSO ve daralma katsayılı PSO'nun çalışmadaki parametre ayarları verilmiştir.

2.2.1. Standart PSO Parametre Ayarları (SPSO)

Standart PSO denklemi 50’de verildiği gibidir, bu fonksiyonda iterasyon sayısı, popülasyon büyüklüğü, atalet ağırlığı w , bilişsel ivme sabit değeri c_1 , sosyal ivme sabit değeri c_2 , rastgele işlev 1 ve rastgele işlev 2 sayısı verilmesi gerekmektedir.

$$V_i^{(k+1)} = w * V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \quad (58)$$

$$X_i^{(k+1)} = X_i^k + V_i^{(k+1)} \quad (59)$$

Parametrelerin anlatımı:

w = Atalet ağırlığı faktörü.

C_1 = Bilişsel ivme sabit değeri.

C_2 = Sosyal ivme sabit değeri.

$rand_1$ = (0, 1) arasında eşit dağılımlı rasgele sayılar.

$rand_2$ = (0, 1) arasında eşit dağılımlı rasgele sayılar.

X_i^k : “i” parçacığının iterasyonda mevcut konumu k .

V_i^k : “i” parçacığının iterasyon k ’daki mevcut hızı.

X_i^{k+1} : “i” parçacığının iterasyondaki mevcut konumu $k + 1$.

V_i^{k+1} : “i” parçacığının iterasyondaki mevcut hızı $k + 1$.

Yukarıda konuşulan parametrelerin ayarları Tablo 1’de verilmiştir.

Tablo 1. Standart PSO parametreleri

Parametre	Değeri
İterasyon Sayısı	30
Popülasyon Büyüklüğü	50
Atalet Ağırlığı, w	0.5
Bilişsel İvme Sabit Değeri, c_1	2
Sosyal İvme Sabit Değeri, c_2	2
Rand1	0 ile 1 arası
Rand2	0 ile 1 arası

2.2.2. Atalet Ağırlığı PSO Parametre Ayarları (IWPSO)

Atalet ağırlığı PSO denklemi 60'te verildiği gibidir, bu fonksiyonda iterasyon sayısı, popülasyon büyüklüğü, atalet ağırlığı w , bilişsel ivme sabit değeri c_1 , sosyal ivme sabit değeri c_2 , rastgele işlev 1 ve rastgele işlev 2 sayısı verilmesi gerekmektedir.

$$V_i^{(k+1)} = w * V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \quad (60)$$

Burada w , atalet ağırlığıdır, denklem 61'de hesaplanır.

$$w = \frac{w_{\max} - w_{\min}}{iter_{\max}} * iter \quad (61)$$

Burada $iter_{\max}$, maksimum iterasyon sayısıdır ve $iter$ geçerli iterasyon sayısıdır. Başlangıçta geliştirildiği gibi, w genellikle bir çalışma sırasında yaklaşık 0.9'dan 0.4'e doğrusal olarak azalır. Diğer parametre ayarlarında tablo 2'de verilmiştir.

Tablo 2. Atalet ağırlığı PSO parametreleri

Parametre	Değeri
İterasyon Sayısı	30
Popülasyon Büyüklüğü	50
Atalet Ağırlığı, w	0.9 ile 0.4 arası
Bilişsel İvme Sabit Değeri, c_1	0.1
Sosyal İvme Sabit Değeri, c_2	0.1
Rand1	0 ile 1 arası
Rand2	0 ile 1 arası

2.2.3. Daralma Katsayısı PSO Parametre Ayarları (CCPSO)

Daralma katsayısı PSO denklemi 62'de verildiği gibidir, bu fonksiyonda iterasyon sayısı, popülasyon büyüklüğü, daralma katsayısı, bilişsel ivme sabit değeri c_1 , sosyal ivme sabit değeri c_2 , rastgele işlev 1 ve rastgele işlev 2 sayısı verilmesi gerekmektedir.

$$V_i^{(k+1)} = K * \left[V_i^k + c_1 * rand_1 * (pbest_i^k - X_i^k) + c_2 * rand_2 * (gbest_i^k - X_i^k) \right] \quad (62)$$

$$K = \frac{2k}{\left| 2 - \phi - \sqrt{\phi^2 - \phi^4} \right|} \quad (63)$$

$$\phi = c_1 + c_2 \quad (64)$$

Genellikle $k=1$, ϕ mutlaka $\phi > 4$ sonra $c_1 = 2.05, c_2 = 2.05$ olarak belirlenmiştir, Bununla birlikte diğer parametreleri tablo 3'te verildiği gibi ayarlanmıştır.

Tablo 3. Daralma katsayısı PSO Parametreleri

Parametre	Değeri
İterasyon Sayısı	30
Popülasyon Büyüklüğü	50
Daralma Katsayısı	K
Bilişsel İvme Sabit Değeri, c_1	2.05
Sosyal İvme Sabit Değeri, c_2	2.05
Rand1	0 ile 1 arası
Rand1	0 ile 1 arası

2.3. PSO Algoritmasının Uygulanması

Bu çalışmada uygulanan standart PSO, atalet ağırlığı tabanlı PSO ve daralma katsayılı PSO'nun aşamaları aynıdır, ancak kullanılan PSO ayarları Tablo 1, 2 ve 3'te gösterildiği gibi değişiklik göstermektedir. Bu bölümde önerilen PSO algoritmalarının adım adım uygulanması gösterilmiştir.

Adım--1: Hat verilerini, bara verilerini, şönt verilerini ve generatör verilerini girin.

Adım--2: Tek TCSC'nin yerini ve boyutlandırmasını parçacıklar olarak ayarlayın.

Adım--3: Hat verilerini, bara verilerini ve generatör verilerini okuyun. Yük akışını çalıştırın ve tüm baralar için voltajı belirleyin.

Adım--4: TCSC yerleştirilmeden önce iletim kayıplarını değerlendirin.

Adım--5: PSO parametrelerinin Tabloda verildiği gibi başlatılması.

Adım--6: Parçacık konumunu rastgele oluşturun.

Adım--7: Parçacık hızını rastgele oluşturun.

Adım--8: Baralara bir "i" parçacığı koyarak yük akışlarını çalıştırın.

Adım--9: Kondisyon fonksiyonunu değerlendirin.

Adım--10: En iyi P değeri belirleyin ve sonra en iyi G değeri belirleyin.

Adım--11: Parçacığın hızını ve konumunu güncelleyin.

Adım--12: En iyi P ve G değerini güncelleyin.

Adım--13: Maksimum yineleme sayısına ulaşırsa, bir sonraki adıma geçin, 2. Adıma geçin.

Adım--14: TCSC yerleştirildikten sonra iletim kayıplarını değerlendirin.

Adım--15: Akış şemasını durdur.

2.4. Kodlama

2.4.1. TCSC Modeli

TCSC Genel bilgiler Kısımında verilen 1.8'deki bilgiye göre Matlab üzerinde modellenmiştir. TCSC değeri her hat için ayrıdır ve aşırı kompanzasyonu önlemek için çalışma aralığı $-0.8X_{line}$ ile $0.2X_{line}$ arasında seçilir. Çalışma aralığında farklı değerleri deneyerek optimal bir TCSC büyüklüğünü ve konumunu bulmaya çalışılır. Yazılan kod Ek 7.1'de verilmiştir.

2.4.2. Amaç Fonksiyon Oluşturma

Amaç fonksiyon optimizasyon probleminde çok önemlidir, Her çalışmalarda ihtiyaca göre farklı fonksiyonlar kullanılmaktadır. Bu çalışmada iletim hattındaki güç kaybını iyileştirme ve optimum TCSC konumunu bulma amaçlandığından dolayı aktif güç kaybını iyileştirme, gerilim profilini iyileştirme ve optimum konumunu bularak maliyeti iyileştirme hedeflenmiştir ve genel bilgilerde verilen 1.9'daki bilgiye göre Matlab üzerinden kodlamalar yapılmıştır. Yazılan kod Ek 7.2'de verilmiştir.

2.4.3. Newton Raphson Yk Akıř Yntemi Oluřturma

İletim řebekesi optimizasyon problemlerinde gç akıř fonksiyonunun seřimi optimizasyon algoritmaların sonucuna etki gsterebilmektedir bundan dolayı bu řalıřmada farklı yk akıř fonksiyonların avantajlarına ve dezavantajlarına bakarak Newton Raphson yk akıř yntemi uygulanmıřtır ve kodlamalar genel bilgilere gre yazılmıřtır. Yazılan kod Ek 7.3'te verilmiřtir.

2.4.4. PSO Algoritmasını Oluřturma

Bu řalıřmada PSO řok nemli faktrlerin biridir, řalıřma amcacı řeřitli PSO algoritmasından bu sistem iřin daha uygun daha verimli PSO'yu bulma hedeflenmiřtir. Sistem iřin en uygun grlen standart PSO, atalet ağırlığı PSO ve bu řalıřmada yeni olarak kullanılan daralma katsayılı PSO'yu karřılařtırma yaparak en uygun grlen algoritmayı sunmuřtur. Yazılan kod Ek 7.4'te verilmiřtir.

3. ÇALIŞMA SONUÇLARI

Bu bölümde, önerilen standart PSO, atalet ağırlığı PSO, daralma katsayısı PSO ile uygulama yapılan çalışmaların sonuçları gösterilmiştir. Birinci çalışma senaryosunda atalet ağırlığı PSO ve daralma katsayılı PSO'yu kullanarak Siti Amely Jumaat ve arkadaşları yaptığı çalışmasındaki aynı şartlar altında deneme yapılmıştır. İkinci çalışma senaryosundaysa IEEE 14 baralı ve IEEE 30 baralı sisteminde standart PSO (SPSO), Atalet ağırlığı tabanlı PSO(IWPSO) ve daralma katsayılı PSO (CCPSO)'yu kullanarak bütün sistemin gerilim performansı üzerinden tekrardan denetlenmiştir. Çalışma sonuçları aşağıdaki gibidir.

3.1. Birinci Çalışma Senaryosu

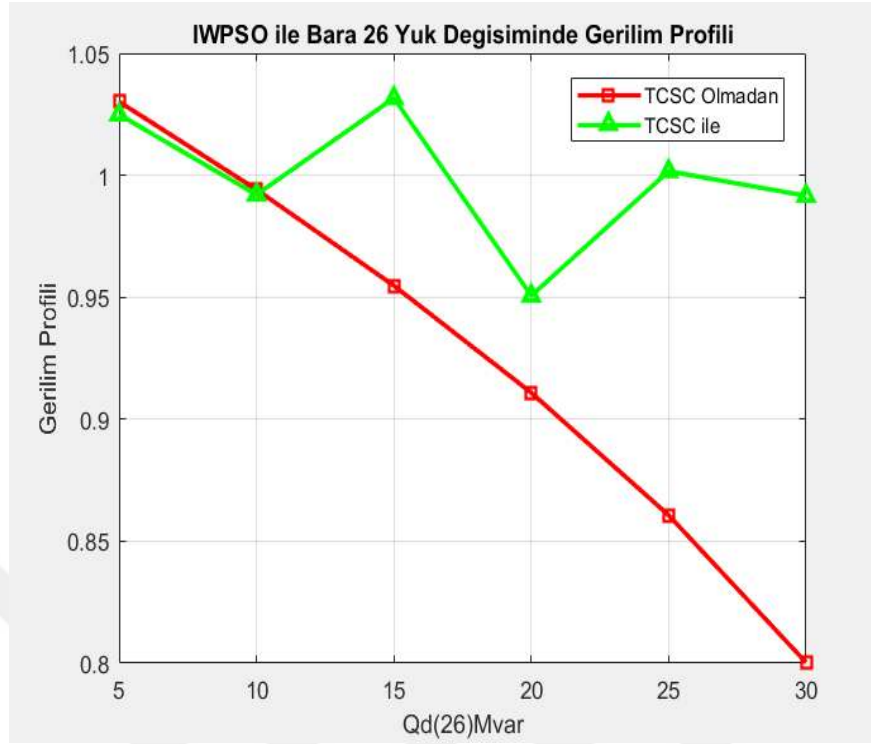
Birinci çalışma senaryosunda önerilen algoritmanın doğrudan çalışmasını tespit etmek için Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmada kullanılan atalet ağırlığı PSO metoduyla uygulama yapılmıştır sonra bu çalışmada yeni olarak önerilen daralma katsayılı PSO metoduyla aynı şartlar altında uygulama yapılmıştır ve bu iki metottu bir bireyle karşılaştırmıştır.

Şekil 4'te atalet ağırlığı PSO metoduyla bara 26'ye 5 MVar dan kademeli yük ekledikten sonraki gerilim performansını göstermiştir.

Şekil 5'te atalet ağırlığı PSO metoduyla bara 29'a 5 MVar dan kademeli yük ekledikten sonraki gerilim performansını göstermiştir.

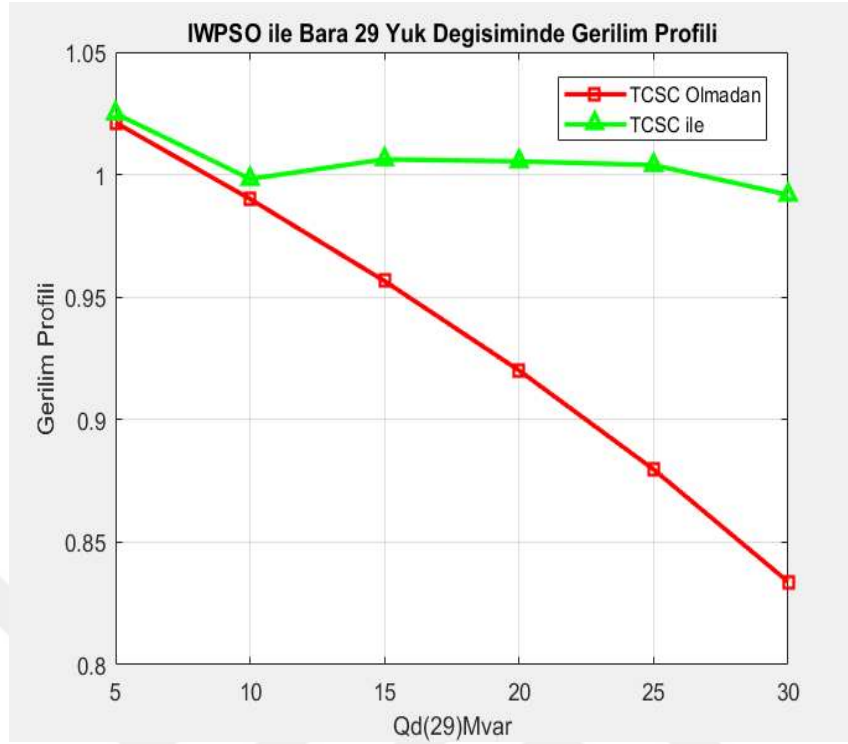
Şekil 6'de bu çalışmada yeni olarak önerilen daralma katsayılı PSO'yu kullanarak bara 26'ye 5 MVar dan kademeli yük ekledikten sonraki gerilim performansını göstermiştir.

Şekil 7'de bu çalışmada yeni olarak önerilen daralma katsayılı PSO'yu kullanarak bara 29'a 5 MVar dan kademeli yük ekledikten sonraki gerilim performansını göstermiştir.



Şekil 4. IWPSO algoritması ile bara 26'ye yük ekledikten sonraki performansı

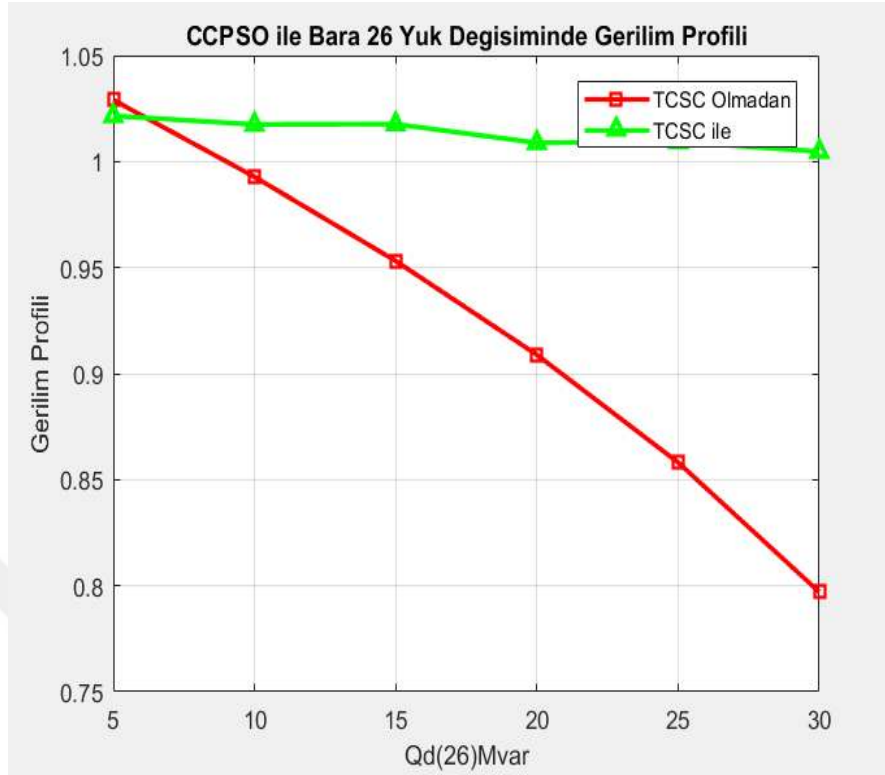
Şekil 4'te baktığımızda bara 26'ye 5 MVar yük eklediğimizde TCSC ile gerilimin normal geriliminden azaldığını görmekteyiz, 10 MVar yük eklediğimizde TCSC ile gerilimin normal gerilim ile hemen olduğunu görebiliriz, Baraya 15 MVar dan 30 MVar'a kadar yük eklediğimizdiyse Şekil 4'te gösterilen Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmadaki gibi TCSC kullandıktan sonra gerilimin normal gerilime göre osilasyon olduğunu görmekteyiz.



Şekil 5. IWPSO algoritması ile bara 29'a yük ekledikten sonraki performansı

Şekil 5'te baktığımızda bara 29'a 5 MVar'dan 30 MVar'a kadar yük eklediğimizde Şekil 4'te gösterilen çalışmadaki gibi TCSC kullandıktan sonra gerilimin normal gerilime göre iyileştiğini ancak gerilimde osilasyon oluşturduğunu görmekteyiz.

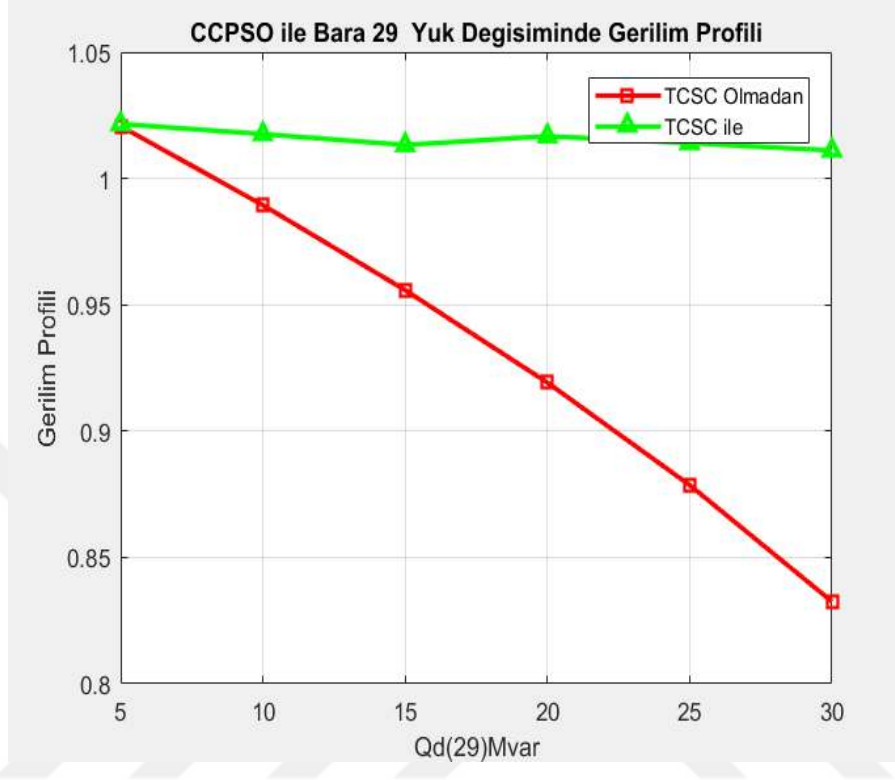
Şekil 6'de Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmadaki aynı şartlar altında bara 26'ye 5 MVar'dan yük ekleyerek ve sadece PSO fonksiyonunu bu çalışmada yeni olarak önerilen daralma katsayısı PSO algoritmasına değiştirerek yapılan çalışmanın gerilim performansı gösterilmektedir.



Şekil 6. CCPSO algoritması ile bara 26'ye yük ekledikten sonraki performansı

Şekil 6'de Siti Amely Jumaat ve arkadaşları [2] yaptığı çalışmadaki aynı şartlar altında bu tez çalışmasında yeni olarak önerilen daralma katsayısı tabanlı PSO'nu uyguladıktan sonra sistem osilasyon probleminin ve gerilimin Şekil 5'te verilen atalet ağırlığı PSO algoritmasına göre sabit şekilde iyileştiğini, Bununla birlikte daralma katsayısı tabanlı PSO'nun bu sistem için daha uygun olduğunu tespit edebilmekteyiz.

Şekil 7'de Şekil 6'de verildiği gibi aynı şartlar altında daralma katsayısı tabanlı PSO algoritmasını uygulayarak bara 29'a kademeli yük ekleyerek yapılan çalışmanın gerilim sonuçları gösterilmiştir.



Şekil 7. CCPSO algoritması ile bara 29'a yük ekledikten sonraki performansı

Şekil 7'de aynı şartlarda bu tez çalışmasında önerilen daralma katsayısı tabanlı PSO'nu uyguladıktan sonra sistem osilasyon probleminin iyileştiğini ve TCSC kullanıldıktan sonra gerilimin sabit şekilde iyileşme olduğunu görebilmekteyiz. Bununla birlikte daralma katsayısı PSO'nun bu sistem için daha uygun olduğunu tespit edebiliriz.

Birinci çalışma senaryosunda kullanılan atalet ağırlığı PSO ve daralma katsayısı PSO algoritmalarını daha iyi karşılaştırmak için ayrı olarak sistemin aktif güç kaybı, reaktif güç kaybı, TCSC yerleşim sonuçları tablo 4'te verilmiştir.

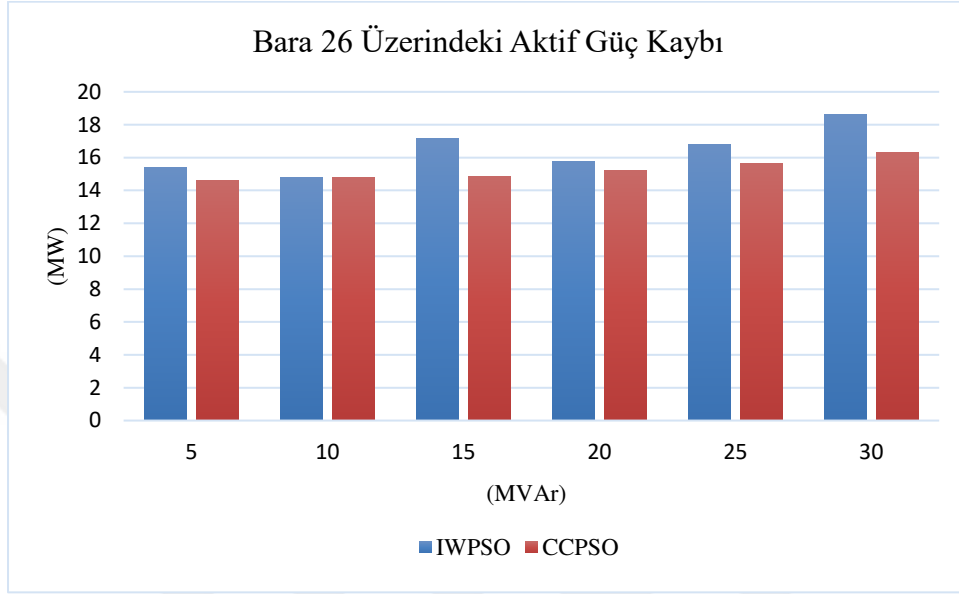
Tablo 4. Bara 26 üzerindeki iletim güç kaybı ve TCSC yerleşimi

IWPSO ile Bara 26 Üzerindeki Sonuçlar						
Yük (MVar)	TCSC Olmadan Aktif Güç Kaybı (MW)	TCSC Olmadan Reaktif Güç Kaybı (MVar)	TCSC ile Aktif Güç Kaybı (MW)	TCSC ile Reaktif Güç Kaybı (MVar)	TCSC Konumu	TCSC Büyüklüğü (p.u)
5	19.9562	75.1328	15.3744	56.2034	29(21-22)	-0.33383
10	20.1171	75.201	14.8107	55.3074	26(10-17)	0.18255
15	20.4791	75.5643	17.1689	56.575	29(21-22)	0.77078
20	21.1046	76.3135	15.7489	56.2675	26(10-17)	-0.053025
25	22.1	77.6029	16.779	57.6455	26(10-17)	-0.15544
30	23.6757	79.7386	18.615	60.3195	26(10-17)	-0.25923
CCPSO ile Bara 26 Üzerindeki Sonuçlar						
Yük (MVar)	TCSC Olmadan Aktif Güç Kaybı (MW)	TCSC Olmadan Reaktif Güç Kaybı (MVar)	TCSC ile Aktif Güç Kaybı (MW)	TCSC ile Reaktif Güç Kaybı (MVar)	TCSC Konumu	TCSC Büyüklüğü (p.u)
5	19.9562	75.1328	14.603	54.996	13(9-11)	-0.1664
10	20.1171	75.201	14.7616	55.0765	13(9-11)	-0.1664
15	20.4791	75.5643	14.8715	55.0615	34(25-26)	-0.304
20	21.1046	76.3135	15.191	55.277	34(25-26)	-0.304
25	22.1	77.6029	15.6553	55.6778	34(25-26)	-0.304
30	23.6757	79.7386	16.2919	56.2986	34(25-26)	-0.304

Tabloda gördüğümüz gibi daralma katsayısı tabanlı PSO (CCPSO) kullandıktan sonra, atalet ağırlığı PSO(IWPSO) çalışması ile karşılaştırma yaparsak aktif ve reaktif güç kayıplarının azaldığını görmekteyiz, Bara 26 üzerinde 5 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.7714 MW azalmıştır, 10 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.049 MW azalmıştır, 15 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 2.2974 MW azalmıştır, 20 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.5579 MW azalmıştır, 25 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 1.1237 MW azalmıştır, 30 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 2.3231 MW azalmıştır, Aynı zamanda CCPSO ile yapılan çalışmada reaktif güç kaybı sonuçlarında azalma görülmüştür ve TCSC yerleşiminin de farklılık göstermektedir.

Aşağıdaki grafikte görsel olarak aktif güç kaybının iyileştiğini görebiliyoruz.

Şekil 8’de Tablo 4’te verilen sistemin aktif güç kaybını göstermektedir, burada reaktif güç kaybı ve aktif güç kaybı grafiklerinde her şey aynı olduğundan sadece aktif güç kaybı sonuçlarını gösterilmektedir.



Şekil 8. Yük ekledikten sonra bara 26 üzerindeki aktif güç kaybı

Bara 26 üzerinde 5 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 15.3744 MW CCPSO ile yapılan çalışmada 14.603 MW olmuştur, 10 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 14.8107 MW CCPSO ile yapılan çalışmada 14.7616 MW olmuştur, 15 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 17.1689 MW CCPSO ile yapılan çalışmada 14.8715 MW olmuştur, 20 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 15.7489 MW CCPSO ile yapılan çalışmada 15.191 MW olmuştur, 25 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 16.779 MW CCPSO ile yapılan çalışmada 15.6553 MW olmuştur, 30 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 18.615 MW CCPSO ile yapılan çalışmada 16.2919 MW olmuştur. Şekil 8’de görsel olarak CCPSO ile yapılan çalışmada aktif güç kaybının azaldığını sistem performansının iyileştiğini görebilmekteyiz.

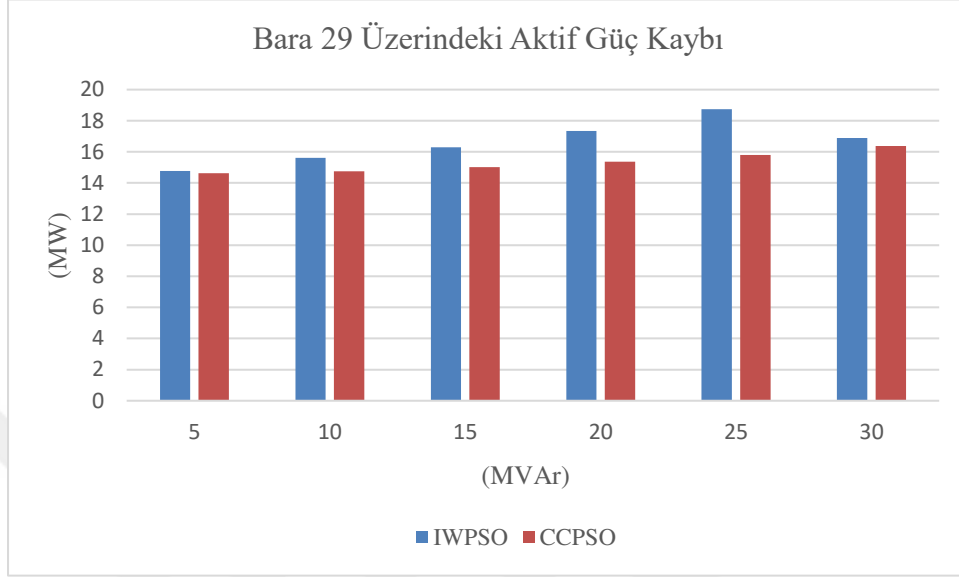
IEEE 30 baralı sistemde bara 29 üzerine yük ekledikten sonraki aktif güç kaybı, reaktif güç kaybı, TCSC yerleşimi aşağıdaki tabloda verilmiştir.

Tablo 5. Bara 29 üzerindeki iletim güç kaybı ve TCSC yerleşimi

IWPSO ile Bara 29 Üzerindeki Sonuçlar						
Yük (MVar)	TCSC Olmadan Aktif Güç Kaybı (MW)	TCSC Olmadan Reaktif Güç Kaybı (MVar)	TCSC ile Aktif Güç Kaybı (MW)	TCSC ile Reaktif Güç Kaybı (MVar)	TCSC Konumu (Baralar arası)	TCSC Büyüklüğü (p.u)
5	19.9891	75.2078	14.7725	55.0824	16(12-13)	-0.16183
10	20.1222	75.3612	15.6073	56.2424	29(21-22)	-0.49379
15	20.3764	75.8136	16.2904	56.7171	29(21-22)	-0.65584
20	20.7826	76.6407	17.338	57.6201	29(21-22)	-0.75607
25	21.3876	77.9574	18.7444	59.1355	29(21-22)	-0.8
30	22.2698	79.9561	16.8935	59.8448	26(10-17)	-0.099294
CCPSO ile Bara 29 Üzerindeki Sonuçlar						
Yük (MVar)	TCSC Olmadan Aktif Güç Kaybı (MW)	TCSC Olmadan Reaktif Güç Kaybı (MVar)	TCSC ile Aktif Güç Kaybı (MW)	TCSC ile Reaktif Güç Kaybı (MVar)	TCSC Konumu (Baralar arası)	TCSC Büyüklüğü (p.u)
5	19.9891	75.2078	14.6334	55.0624	13(9-11)	-0.1664
10	20.1222	75.3612	14.759	55.2129	13(9-11)	-0.1664
15	20.3764	75.8136	15.0081	55.67	13(9-11)	-0.1664
20	20.7826	76.6407	15.3621	54.9303	36(28-27)	-0.29595
25	21.3876	77.9574	15.7957	55.605	36(28-27)	-0.3168
30	22.2698	79.9561	16.38	56.654	36(28-27)	-0.3168

Tabloda gördüğümüz gibi CCPSO kullandıktan sonra, IWPSO çalışması ile karşılaştırma yaparsak aktif ve reaktif güç kayıplarının azaldığını görmekteyiz, Bara 29 üzerinde 5 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.14 MW azalmıştır, 10 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.85 MW azalmıştır, 15 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 1.2823 MW azalmıştır, 20 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 1.9759 MW azalmıştır, 25 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 2.9487 MW azalmıştır, 30 MVar yük eklediğimizde CCPSO ile yapılan çalışmada IWPSO çalışmasına göre aktif güç kaybı 0.5135 MW azalmıştır, Aynı zamanda CCPSO ile yapılan çalışmada reaktif güç kaybı sonuçları üzerinde azalma görülmüştür ve yükten dolayı TCSC yerleşiminde farklılık göstermektedir.

Şekil 9'da görsel olarak bara 29'za yük ekledikten sonraki aktif güç kaybı performansı gösterilmektedir.



Şekil 9. Yük ekledikten sonra bara 29 üzerindeki aktif güç kaybı

Bara 29 üzerinde 5 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 14.7725 MW CCPSO ile yapılan çalışmada 14.6334 MW olmuştur, 10 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 15.6073 MW CCPSO ile yapılan çalışmada 14.759 MW olmuştur, 15 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 16.2905 MW CCPSO ile yapılan çalışmada 15.0081 MW olmuştur, 20 MVar yük eklediğimizde IWPSO ile yapılan çalışmada Aktif güç kaybı 17.338 MW CCPSO ile yapılan çalışmada 15.3621 MW olmuştur, 25 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 18.7444 MW CCPSO ile yapılan çalışmada 15.7957 MW olmuştur, 30 MVar yük eklediğimizde IWPSO ile yapılan çalışmada aktif güç kaybı 16.8935 MW CCPSO ile yapılan çalışmada 16.38 MW olmuştur. Şekil 9'da görsel olarak CCPSO ile yapılan çalışmada aktif güç kaybının azaldığını sistem performansının iyileştiğini görebilmekteyiz.

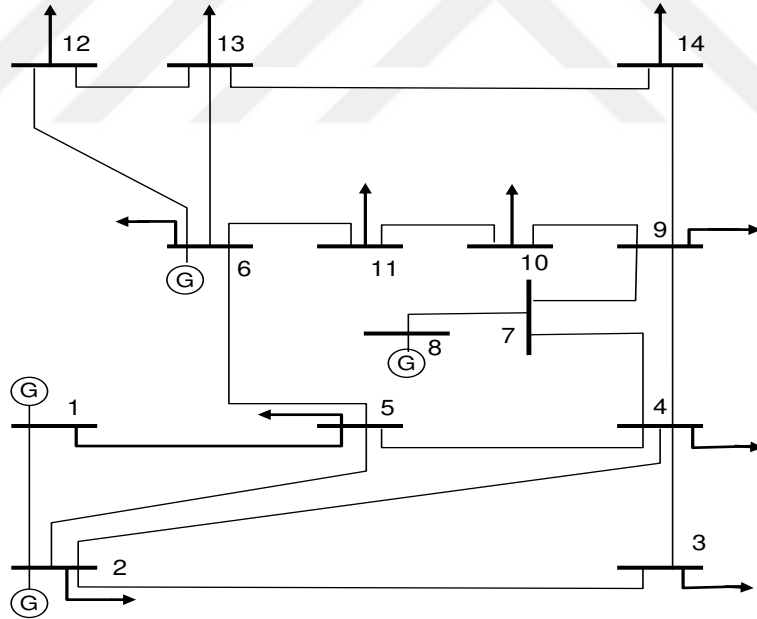
Birinci çalışma senaryosunda IWPSO ve CCPSO sonuçlarına karşılaştırma yaparken CCPSO'nun yük eklenmiş sistem için iyi performans gösterdiğini görmekteyiz. Bununla birlikte algoritmanın bütün sistem üzerinden tespit etmek için aşağıdaki çalışma yapılmıştır.

3.2. İkinci Çalışma Senaryosu

İkinci çalışma senaryosunda önerilen üç farklı algoritma standart PSO, atalet ağırlığı PSO, daralma katsayısı PSO algoritmaları ile IEEE-14 baralı ve IEEE-30 baralı sistemde analize edilen çalışmaların sonucu gösterilmiştir.

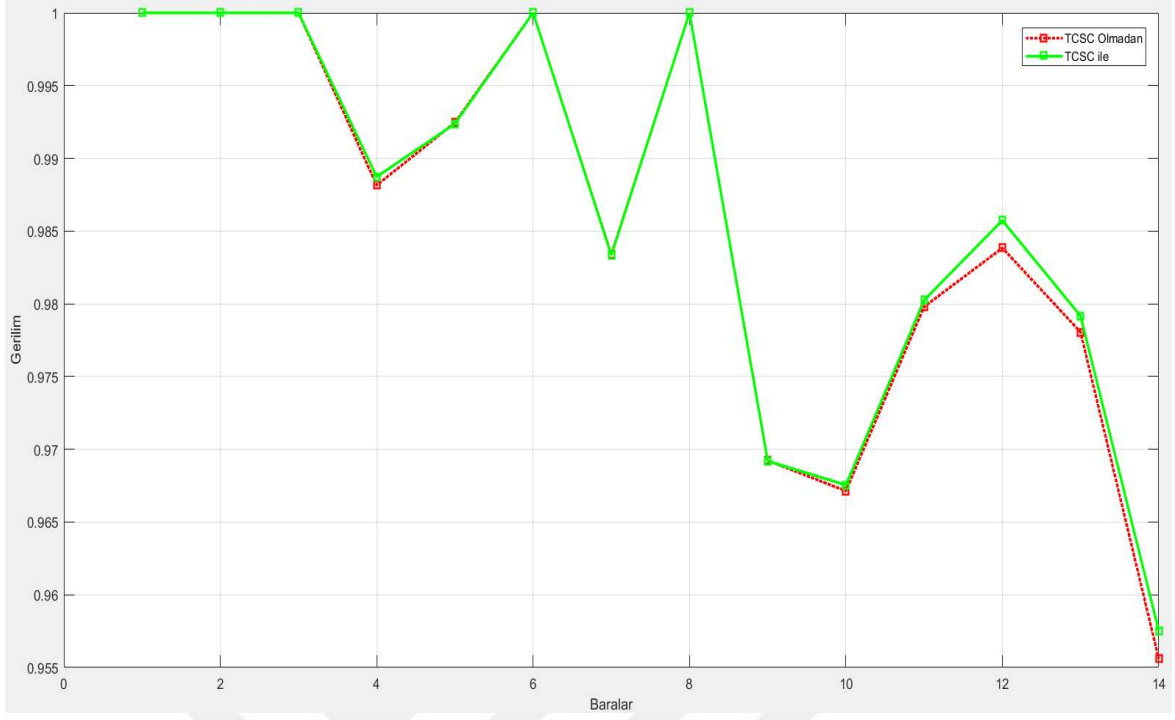
3.2.1. IEEE 14 Baralı Sistemin Performans Sonuçları

IEEE 14 baralı test sisteminin tek hat şeması Şekil 10'da gösterilmiştir. IEEE14 Baralı Sistemi 4 senkron makineden oluşur, bunlardan ikisi sadece reaktif güç desteği için kullanılan senkron kapasitörlerdir. Bu test sistemi, bir elektrik şebekesini simüle etmek için yeterli üretim gücüne ve yüke sahiptir bu nedenle, önerilen tekniklerle anlamlı sonuçlar üretebilmektedir.

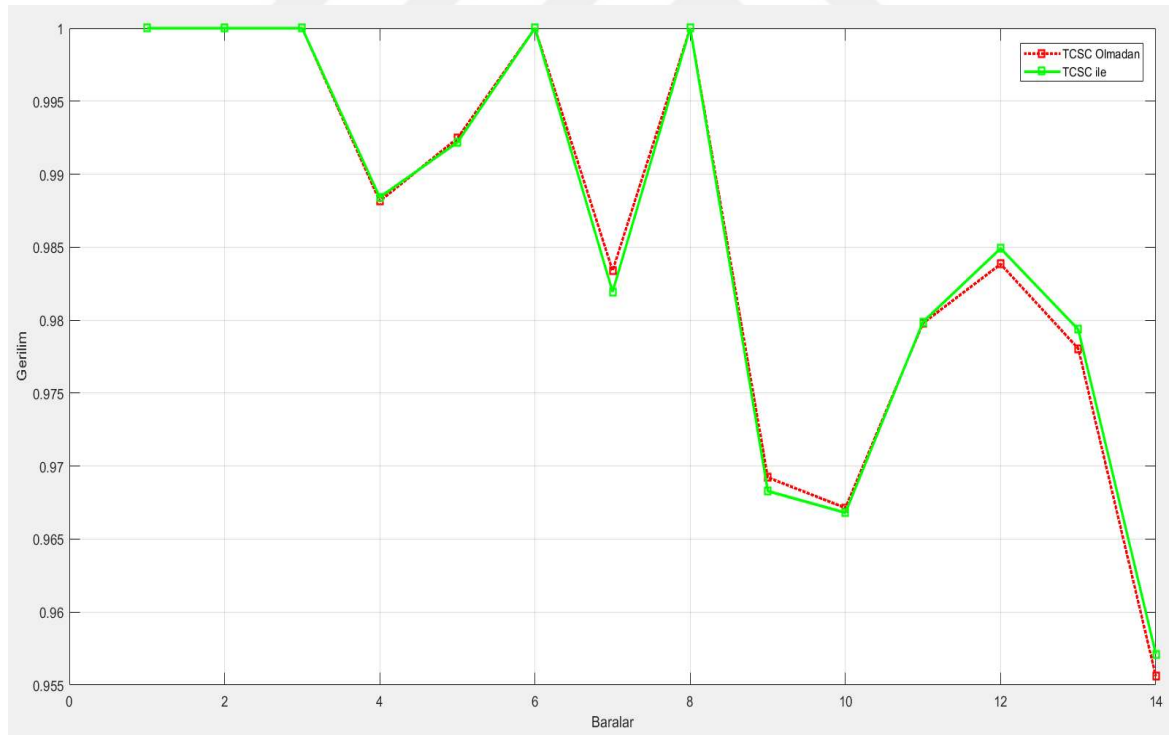


Şekil 10. IEEE-14 baralı sisteminin tek hat şeması

IEEE 14 baralı test sisteminde Standart PSO, Atalet ağırlığı PSO, Daralma katsayısı tabanlı PSO algoritmaları ile aynı sistem üzerinden aynı amaç fonksiyonu ve aynı güç akış analizi kullanılarak Matlab üzerinde çalışma yapılmıştır kullanılan metotlar ve algoritma ayarları genel bilgilerde ve yapılan çalışmalarda detaylı anlatılmıştır.

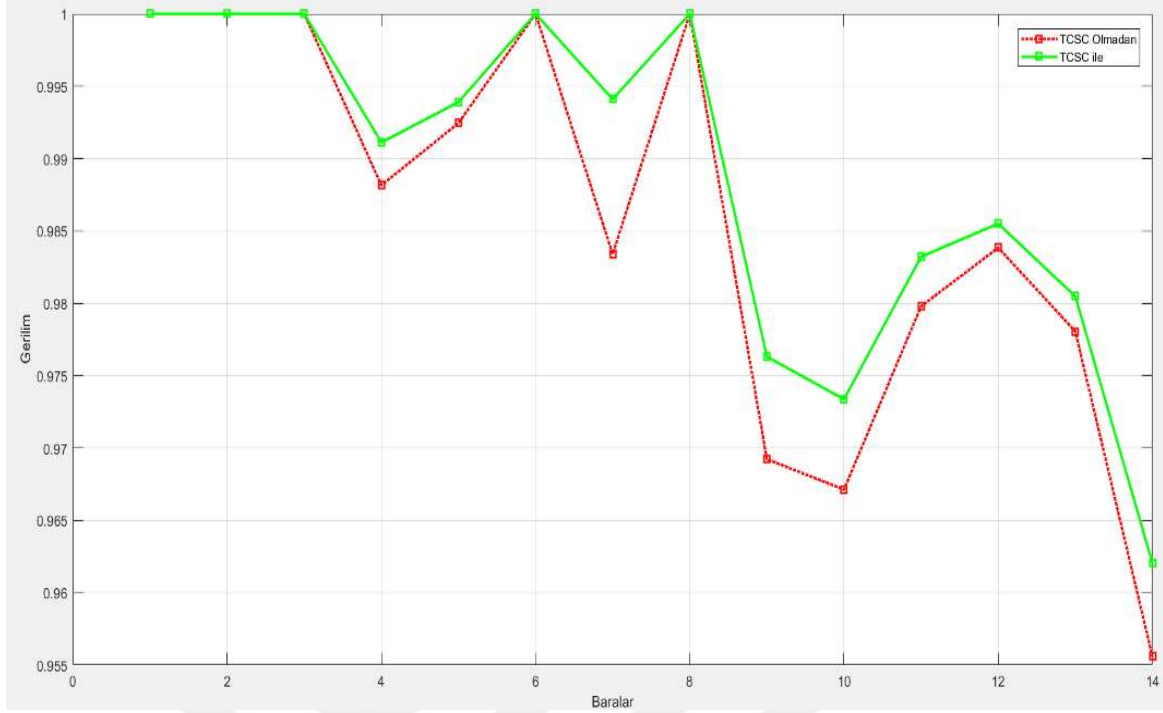


(a)



(b)

Şekil 11. IEEE-14 baralı sisteminde TCSC olmadan ve TCSC ile gerilim profili,
 (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı tabanlı
 PSO sonuçları

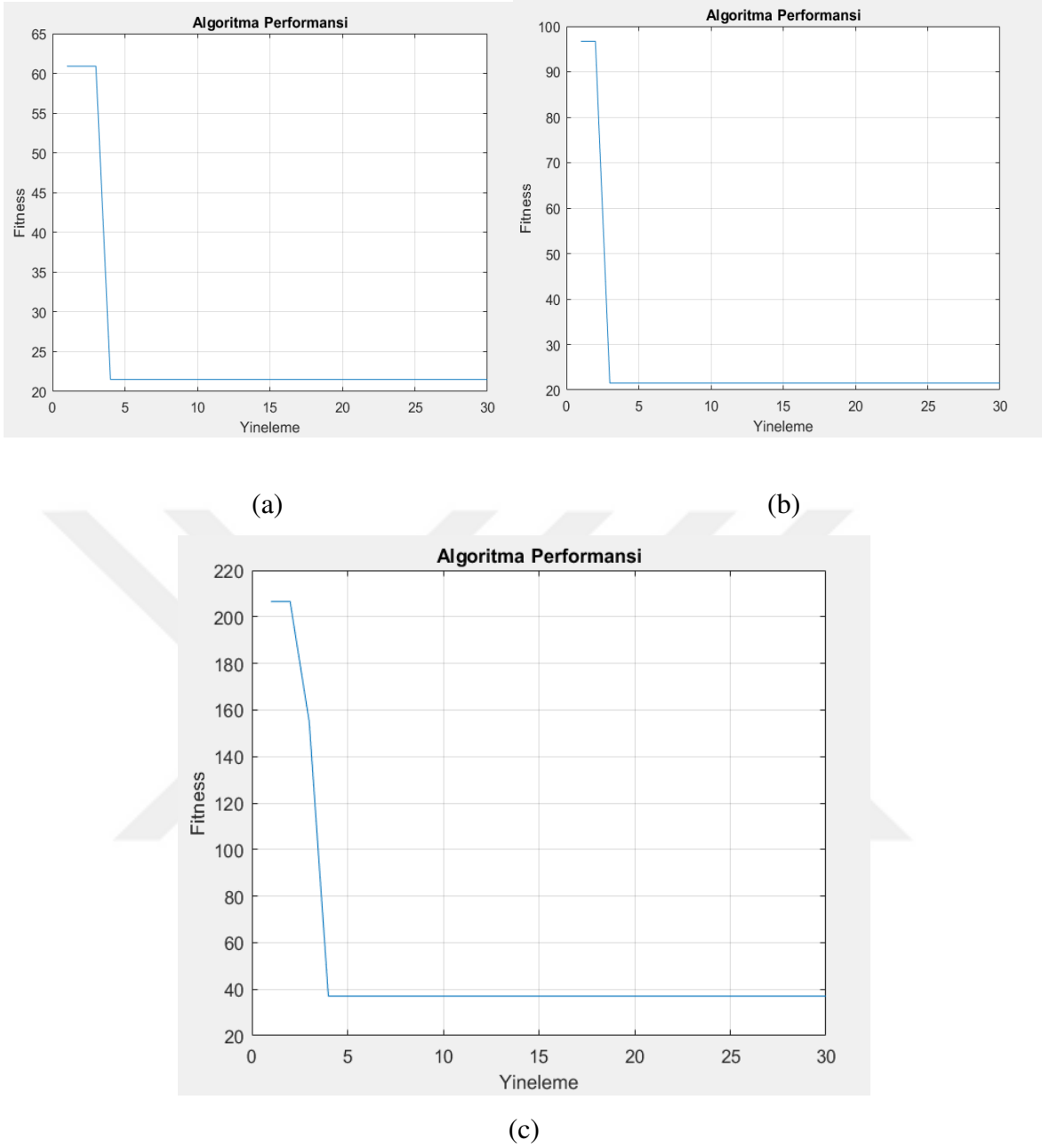


(c)

Şekil 11’de IEEE 14 baralı sisteminde yapılan çalışmanın TCSC olmadan önceki gerilim değerleri aynıdır sonra bu çalışmada önerilen üç farklı PSO üzerinden deneme yaparak en iyi gerilim performansı almaya çalışılmıştır. Gerilim profilinin iyileşmesi güç kayıplarının azaldığından olmaktadır bu nedenle gerilim performansı daha iyi olan çalışma en optimal sonuç veren algoritmadır.

Gerilim performansına bakarsak en iyi olarak CCPSO üzerinde görülmüştür. Gerilimin bu algortmada ilk baralarda değeri TCSC olmadan normal gerilim sonucu ile aynıdır ve daha sonra gerilim bara 3 ila 14 arasında çok farklılık göstermektedir. TCSC ile en yüksek gerilim değeri bara 1’de, en düşük gerilim değeri bara 10’dadır. Benzeri şekilde, TCSC olmadan en yüksek gerilim değeri bara-1’dedir ve en düşük gerilim değeri bara 26’tıdır. Bu üç farklı PSO algoritmasını karşılaştırsak daralma katsayısı tabanlı PSO’nun daha iyi performans verdiğini görebilmekteyiz.

Şekil 12’de amaç fonksiyonunun farklı algortmada çalışma performansını sunmaktadır.

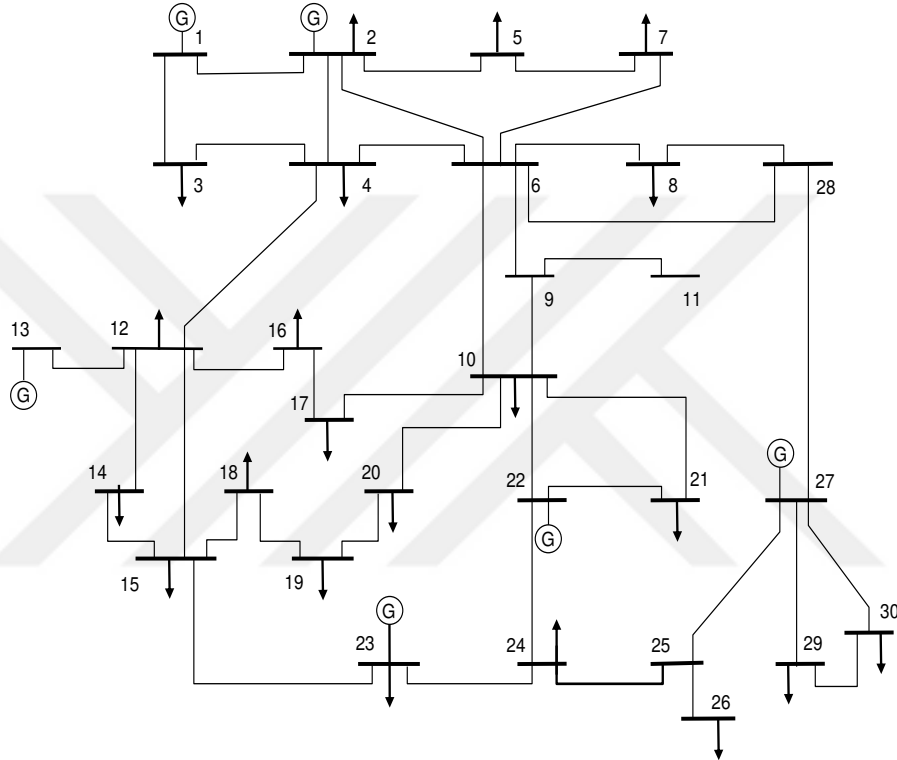


Şekil 12. IEEE 14 baralı sisteminde PSO algoritmaları performansı, Burada
 (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı
 PSO sonuçları

PSO'da 4-iterasyonda sabit değere ulaştığını, Atalet ağırlığı PSO'da 3-iterasyonda sabit değer verdiğini, Daralma katsayısı tabanlı PSO'da 4-iterasyonda sabit değer verdiğini görebilmekteyiz. Amma fitness fonksiyonunun sabitleşmiş son değerinde farklılık göstermektedir bu da TCSC konum değişikliğinden dolayı maliyet fonksiyonunda farklı değer verdiğinden olmaktadır.

3.2.2. IEEE 30 Baralı Sistemin Performans Sonuçları

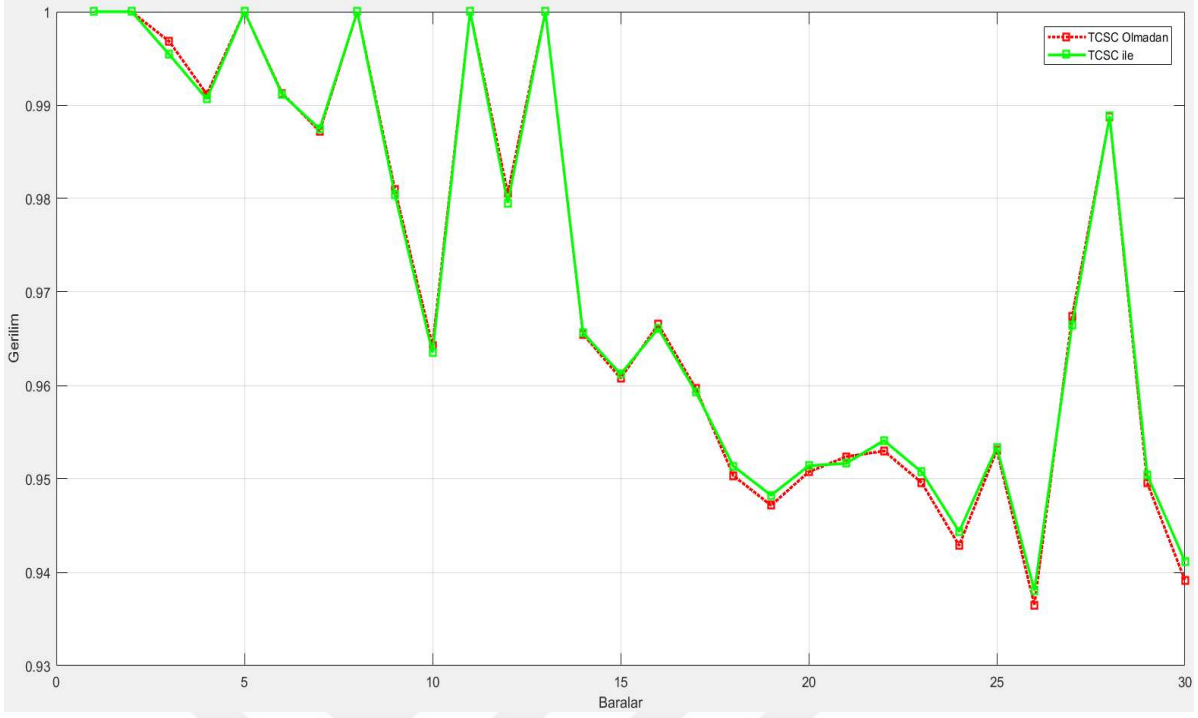
Şekil 13'te gösterilen IEEE-30 baralı Sistemin çalışma şemasıdır bu sistemde 6 jeneratör ve 24 yük barası vardır. Bu test sistemi, bir elektrik şebekesini simüle etmek için yeterli üretim gücüne ve yüke sahiptir bu nedenle, önerilen tekniklerle anlamlı sonuçlar üretir.



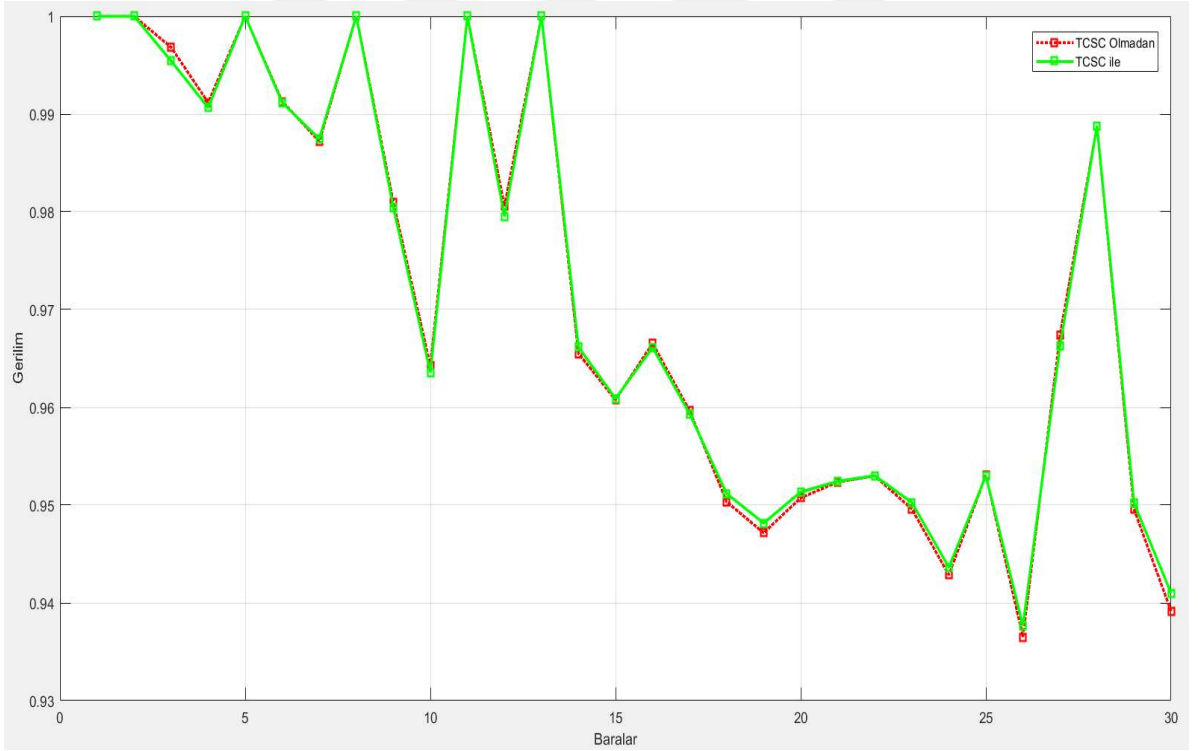
Şekil 13. IEEE-30 baralı sistemin tek hat şeması

IEEE 30 baralı test sisteminde çalışmalar standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO ile aynı sistem üzerinden aynı amaç fonksiyonu ve aynı güç akış analizi kullanılarak yapılmıştır kullanılan metotlar ve algoritma ayarları genel bilgiler ve yapılan Kısımında çalışmalarda detaylı anlatılmıştır.

Şekil 14'te TCSC olmadan ve TCSC ile ayrı olarak, standart PSO, atalet ağırlığı PSO, daralma katsayısı PSO kullanarak IEEE-30 baralı sistemin gerilim profili sonuçları verilmiştir.



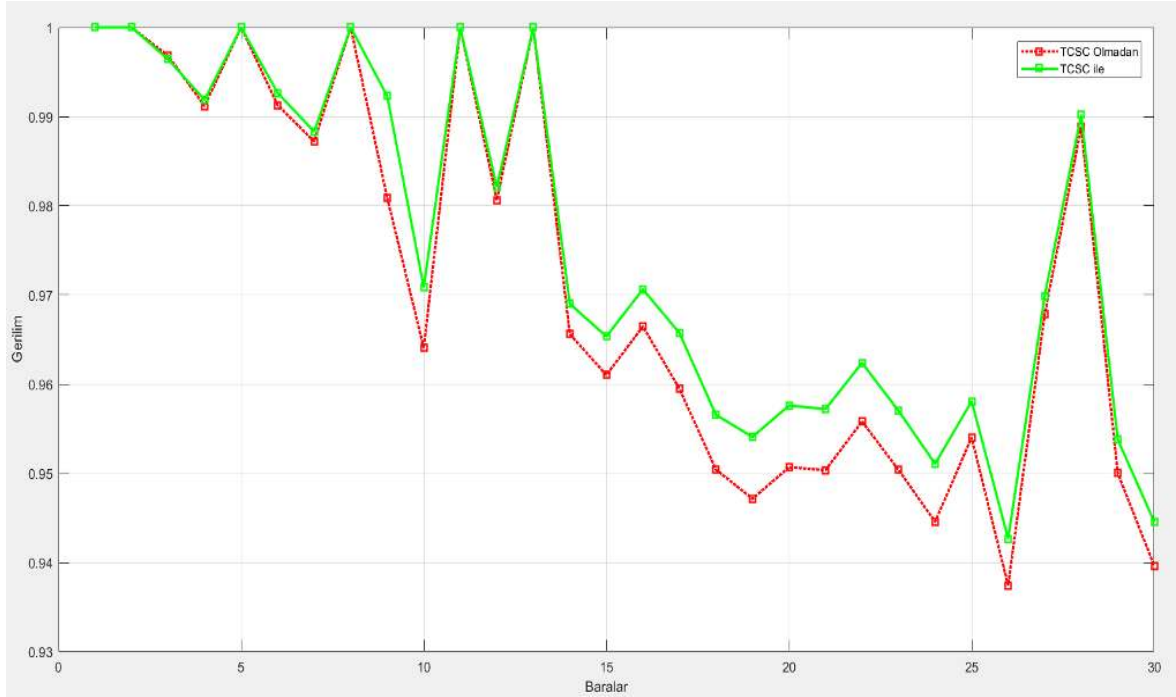
(a)



(b)

Şekil 14. IEEE-30 baralı sisteminde TCSC olmadan ve TCSC ile gerilim profili, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı tabanlı PSO sonuçları

Şekil 14'ün devamı

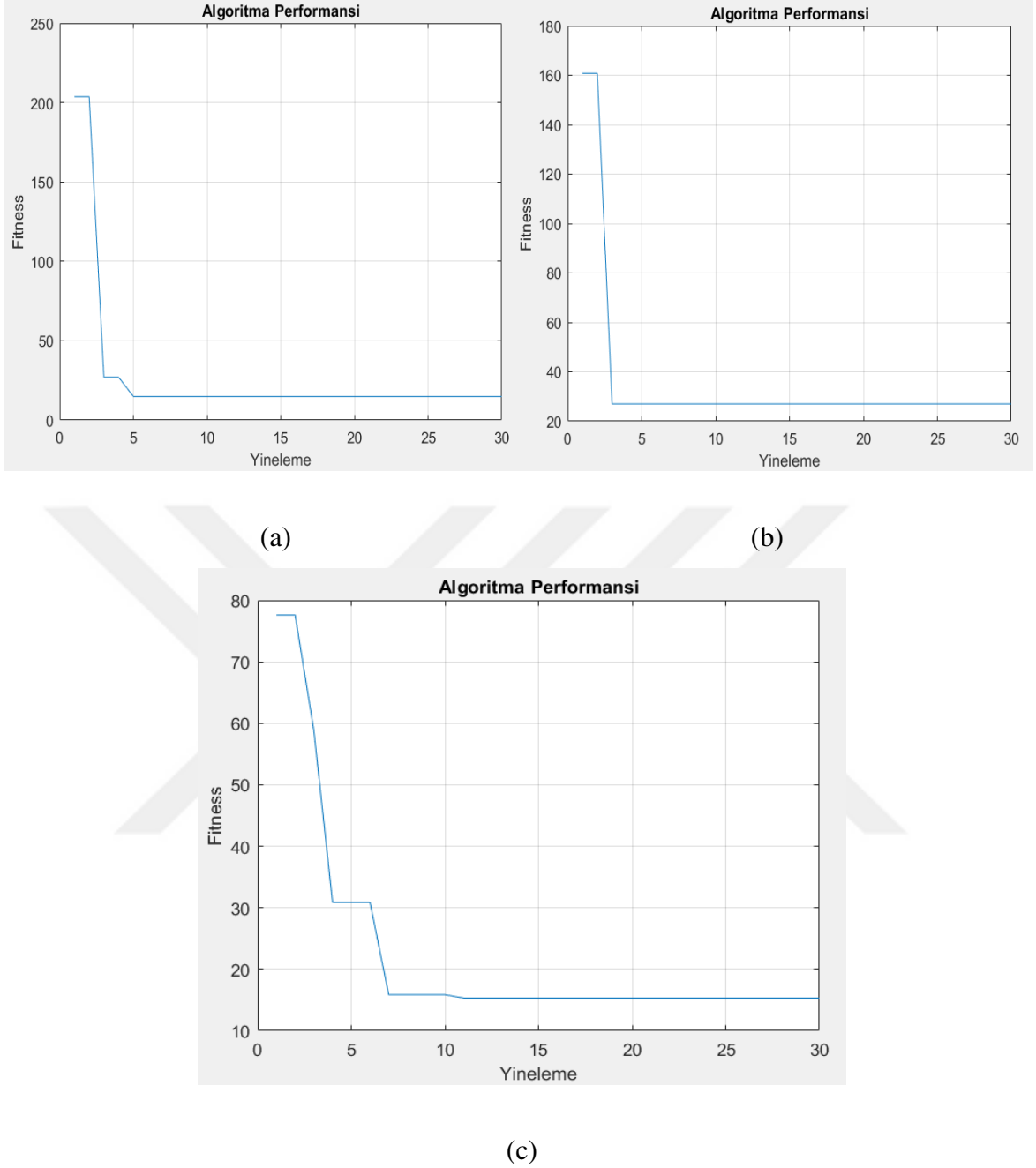


(c)

Şekil 14'te IEEE 30 baralı sistem üzerinde yapılan çalışmanın gerilim performansını göstermektedir. TCSC olmadığında önceki gerilim değerleri aynıdır. Sonra bu çalışmada önerilen üç farklı PSO üzerinden deneme yaparak en iyi gerilim performansı almaya çalışılmıştır gerilim profilinin iyileşmesi güç kayıplarının azaldığından olmaktadır bu nedenle gerilim performansı daha iyi olan çalışma en optimal sonuç veren algoritmadır.

Gerilim performansına bakarsak en iyi olarak daralma katsayısı PSO üzerinde görülmüştür. Gerilimin bu algortmada ilk baralarda değeri aynıdır ve daha sonra gerilim bara 9 ila 30 arasında çok farklılık göstermektedir. TCSC ile en yüksek gerilim değeri bara 1'de, en düşük gerilim değeri bara 26'tıdır. Benzeri şekilde, TCSC olmadan en yüksek gerilim değeri bara 1'dedir ve en düşük gerilim değeri bara 26'tıdır. Bu üç farklı PSO algortmasını karşılaştırsak daralma katsayısı PSO algortmasının daha iyi performans verdiğini görebilmekteyiz.

Şekil 15'te maksimum yinleme sayısı 30 olan üç farklı PSO algortmalarının performansını göstermektedir.



Şekil 15. IEEE 30 baralı sisteminde PSO algoritmaları performansı, Burada (a) Standart PSO, (b) Atalet ağırlığı PSO, (c) Daralma katsayısı PSO sonuçları

Şekil 15'te maksimum yineleme sayısı 30 olan üç farklı PSO algoritmalarının performansını göstermektedir. Bu üç farklı algoritma performansına bakarsak standart PSO'da 5-iterasyonda sabit değere ulaştığını, atalet ağırlığı PSO'da 4-iterasyonda sabit değer verdiğini, daralma katsayısı tabanlı PSO'da 6-iterasyonda sabit değer verdiğini görebilmekteyiz.

3.3. IEEE 14 ve 30 Baralı Sistemin Güç Kaybı Sonuçları

Tablo 6’de standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO olarak IEEE-14 baralı ve 30 baralı sistem üzerinde aktif ve reaktif güç kayıp sonuçları gösterilmiştir.

Tablo 6. MW Cinsinden Aktif Güç Kaybı

Standart PSO (SPSO)		
IEEE SİSTEMİ	TCSC olmadan (MW)	TCSC ile (MW)
14	15.9817	11.6878
30	19.9558	14.6283
Atalet Ağırlığı PSO (IWPSO)		
14	15.9817	11.6878
30	19.9558	14.6283
Daralma Katsayısı tabanlı PSO (CCPSO)		
14	15.9817	11.6625
30	19.9595	14.6105

Bu tabloda standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO olarak aktif güç kayıp sonuçları gösterilmiştir, IEEE 14 baralı sistemde aktif güç kaybı en az 11.6625 MW olarak, IEEE 30 baralı sistemde 14.6105 MW olarak daralma katsayısı tabanlı PSO sisteminde görülmüştür.

Tablo 7. MVAr cinsinden Reaktif Güç İletim kaybı

Standart PSO (SPSO)		
IEEE SİSTEMİ	TCSC olmadan (MVAr)	TCSC ile (MVAr)
14	63.0669	46.0951
30	75.3010	55.3604
Atalet Ağırlığı PSO (IWPSO)		
14	63.0669	46.0951
30	75.3010	55.3604
Daralma Katsayısı tabanlı PSO (CCPSO)		
14	63.0669	46.0621
30	75.3131	55.1771

Bu tabloda standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO olarak sonuçlar ayrı şekilde gösterilmiştir, IEEE 14 baralı sistemde reaktif güç kaybı en az 46.0621 MVar olarak, IEEE 30 baralı sistemde 55.1771 MVar olarak daralma katsayısı tabanlı PSO sisteminde görülmüştür.

3.4. IEEE 14 ve 30 Baralı Sistemin TCSC Optimal Konumu, Büyüklüğü Ve Kurulum Maliyeti

Tablo 8’de standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO olarak IEEE-14 baralı ve 30 baralı sisteminde TCSC’nin optimal konumu, büyüklüğü ve kurulum maliyeti sonuçları ayrı şekilde gösterilmiştir

Tablo 8. TCSC’nin optimal konumu, büyüklüğü ve kurulum maliyeti

Standart PSO (SPSO)			
IEEE SİSTEMİ	TCSC Maliyeti	TCSC Konumu (Baralar arası)	TCSC Büyüklüğü
14	403.8409	19(12-13)	0.2000
30	4155.1	29(21-22)	0.2000
Atalet Ağırlığı PSO (IWPSO)			
14	403.8409	19(12-13)	0.2000
30	4155.1	29(21-22)	0.2000
Daralma Katsayısı tabanlı PSO (CCPSO)			
14	20530	14(7-8)	-0.1409
30	21768	13(9-11)	-0.1664

Bu tabloda TCSC’ın farklı algoritmada optimal maliyeti ve yeri belirlenmiştir. TCSC maliyet birimi [US\$/KVar] olarak ve TCSC büyüklüğü p.u (0.2000, 0.2000, -0.1409, -0.1664) olarak belirlenmiştir. 14 baralı sistem için TCSC konumu standart PSO ve atalet ağırlığı PSO’da bara 12 ve 13 arasındaki hat 19 üzerinde seçilmiştir ve TCSC boyutu 0.2 p.u olarak her iki algoritmada aynıdır amma daralma katsayısı PSO’da konumu bara 7 ve 8 arasındaki hat 14 üzerinde TCSC boyutu -0.1409 p.u olarak seçilmiştir. 30 baralı sistem için TCSC konumu standart PSO ve atalet ağırlığı PSO’da bara 21 ve 22 arasındaki hat 29 üzerinde seçilmiştir ve TCSC boyutu 0.2 p.u olarak aynıdır amma daralma katsayısı PSO’da konumu bara 9 ve 11 arasındaki hat 13 üzerinde seçilmiştir TCSC boyutu -0.1664 p.u olarak seçilmiştir.

4. TARTIŞMA VE BULGULAR

IEEE 14 ve 30 baralı sistemlerin Simülasyonları Matlab üzerinde gerçekleştirilmiştir. Sonuçlardan aşağıdakiler ortaya çıkmıştır:

1. Birinci çalışma senaryosunda sonuçları karşılaştırma yaptığımız makaleye göre atalet ağırlığı PSO kullanılarak 30 baralı sistem üzerinde uygulanmıştır. Sonra sisteme daralma katsayısı PSO'yu ekleyerek aynı şartlarda analiz yapılmıştır ve daralma katsayısı tabanlı PSO'nun sistem için daha uygun olduğunu ve sistem performansının iyileştiğini güç kayıplarının atalet ağırlığı PSO'ya göre daha azaldığını görmekteyiz.

2. İkinci çalışma senaryosunda standart PSO, atalet ağırlığı PSO, daralma katsayısı tabanlı PSO kullanarak yapıldığı çalışmada aktif güç iletim kaybını önemli faktör olarak göz atarsak IEEE 14 baralı sisteminde aktif güç kaybı en az 11.6625 MW ve IEEE 30 baralı sistemin 14.6105 MW olarak daralma katsayısı PSO algoritmasında göstermiştir ve diğer standart PSO'da ve atalet ağırlığı PSO'da aktif güç kaybının farklı olmadığını görebilmekteyiz. Bu nedenle bu çalışmada uygulanan daralma katsayısı tabanlı PSO'nun daha iyi performans gösterdiğini tespit edebilmekteyiz.

3. Bu üç farklı algoritmaya reaktif güç iletim kaybı bazından bakarsak IEEE 14 baralı sisteminde reaktif güç kaybı en az 46.0621 MVar ve IEEE 30 sisteminde 55.1771 MVar olarak daralma katsayısı tabanlı PSO sisteminde göstermiştir ve diğer Standart PSO'da ve atalet ağırlığı PSO'da reaktif güç kaybının farklı olmadığını görebilmekteyiz. Bu nedenle bu çalışmada uygulanan daralma katsayısı tabanlı PSO'nun daha iyi performans gösterdiğini tekrardan tespit edebilmekteyiz.

4. Bu üç farklı algoritmaya kurulum maliyeti açısından bakarsak sistemde kurulum maliyeti standart PSO'da ve atalet ağırlığı PSO'da en düşük olarak sonuç vermiştir, daralma katsayısı tabanlı PSO'da kurulum maliyeti en yüksek sonuç vermiştir. Bu nedenle maliyet açısından çalışma yapılırsa standart PSO ve atalet ağırlığı PSO daha iyi performans gösterir.

5. ÖNERİLER

Ekonomik, çevresel ve politik nedenlerden dolayı yeni iletim hatları inşa etmek çok zordur. Dolayısıyla, TCSC gibi yeni FACTS cihazını kurarak mevcut iletim sistemlerinin optimum kullanımına ilgi duyulmaktadır. TCSC, birçok avantaj sunan popüler olarak kullanılan FACT cihazıdır. Ancak birçok nedenlerden dolayı tam faydalarını elde etmek için en uygun yerleşimini bulmak çok önemlidir. Bu tez temel olarak TCSC'nin optimal yerleşimini, güç kaybı ve maliyeti gibi konularını en optimal duruma getirmek için farklı PSO çalışmalarından en iyi performans veren algoritmayı bulmuştur.

Literatür araştırması bölümünde FACTS cihazlarının optimal konumuyla ilgili çalışmalar üzerine kapsamlı bir inceleme yapılmıştır ve farklı optimizasyon algoritması türleri tartışılmıştır. Sonunda standart PSO, atalet ağırlığı PSO, daralma katsayısı PSO seçilerek çalışma yapılmıştır.

TCSC'nin optimum yerini bulmak ve güç kaybını en aza indirmek için standart PSO, atalet ağırlığı PSO, daralma katsayısı PSO algoritmaları kullanarak IEEE 14 ve 30 baralı sistemde test edilmiştir. Aynı anda Optimal güç akışı gibi çeşitli güç sistemi optimizasyon problemlerini çözmek için güç sistemi kontrolörü tasarımı, maliyet tahmini, yük tahmini ve güç sisteminin diğer alanları tartışılmıştır.

Ayrıca, bu tez çalışmasında yeni olarak kullanılan daralma katsayısı PSO algoritmasının çözülmesi gereken bazı sorunları vardır. Bu nedenle, daralma katsayısı PSO algoritması üzerinde gelecekteki çalışmalar muhtemelen aşağıdakilere odaklanabilir:

1. Daralma katsayısını etkileyecek faktörler üzerinde çalışma yaparak yeni bir algoritmanın sunulması.
2. Doğruluğu artırmak için daralma katsayısı PSO algoritmasının farklı optimizasyon yöntemleriyle birleştirilmesi.
3. Daralma katsayısı PSO algoritmasını yeni algoritmalar ile karşılaştırma yapılması.

6. KAYNAKLAR

1. Narain Hingorani., Laszlo Gyugyi ve Mohamed El-Hawary., Understanding FACTS Concepts and Technology of Flexible AC Transmission Systems, International Journal of Engineering Science, 20,6 (2000) 423-431.
2. Xe, Da., Niu, H., Chen, C. ve Wu, J., An Algorithm to Control the Power Flow in Large Systems Based On TCSC, Proceedings of Power System Technology, 8,3 (1998) 344-348.
3. Fuerte, E., Acha, H. ve Ambriz, P., A Thyristor Controlled Series Capacitor Model for the Power Flow Solution of Practical Power Networks, IEEE Transactions on power systems, 15,1 (2000) 58-64.
4. Pinto, H., Pereira, J., Martins, N., Filho, J., Junior, S., Alves, F., Ferraz, J., Henriques, R. ve Costa, V., Needs and Improvements in Power Flow Analysis, VII Symposium of Specialists in Electric Operational and Expansion Planning, Mayıs 2000, Brasilia, Brasil.
5. Mukhopadhyay, S., Tripathi, V. ve Prasher, V., Application of FACTS in Indian Power System, Transmission and Transmission Conference and Exhibition Asia Pacific, 1, 5 (2002) 237-242.
6. Abdel, M. ve Padhy, A., Newton-Raphson TCSC Model for Power Flow Solution of Practical Power Networks, IEEE Power Engineering Society Summer Meeting, Temmuz 2002, NYC, USA.
7. John, J. ve Paserba, A., How FACTS Controllers Benefit AC Transmission System, IEEE Proceedings of the Power Engineering Society General Meeting, Haziran 2004, USA.
8. Hassan, S., Cheng, J., ve Zakaria, Z., Steady-state Modeling of SVC and TCSC for Power Flow Analysis, International MultiConference of Engineers and Computer Scientists, Mart 2009, Hong Kong.
9. Parate, J. ve Sindekar, P., Reactive Power Control and Transmission Loss Reduction with Realization of SVC and TCSC, International Journal of Engineering Science and Technology , 4, 7 (2012) 0975-5462.
10. Paul, C. ve Narasimham, S., Enhancement of Power Flow Using TCSC Controller, International Journal of Electrical and Electronics Engineering (IJEET), 1, 1 (2012) 75-82.

11. Acharya, N. ve Mithulananthan, N., Locating Series FACTS devices for Congestion Management in Deregulated Electricity Markets, *Electric Power Systems Research*, 77,2 (2001) 352–360.
12. Siti, J., Ismail, M., Muhammad, M. ve Hazlie M., Placement and Sizing of Thyristor Controlled Series Capacitor Using PSO Based Technique for Loss Minimization IEEE International Power Engineering and Optimization Conference, Haziran 2012, Melaka, Malezya.
13. Acharya, N., Sode-Yome, A. ve Mithulananthan, A., Facts about flexible AC transmission systems (FACTS) controllers: practical installations and benefit, *Proceedings of Australian Universities Power Engineering Conference*, Eylül 2005, Avustralya.
14. Sreejith, S., Simon, P. ve Selvan, M., Power Flow Analysis Incorporating Firing Angle Model Based TCSC, IEEE 5th International conference on industrial and information systems, Temmuz 2010, Mangalore, Hindistan.
15. Kumar, N., Venkatesh, P. ve Sangamewara, R., Modeling and Analysis of SVC, TCSC, TCPAR in Power Flow Studies, *International Journal of Emerging Technology and Advanced Engineering*, 3, 1 (2013) 203-211.
16. Siti, J., Ismail, M., Muhammad, M. ve Hazlie M., Placement and Sizing of Thyristor Controlled Series Capacitor Using PSO Based Technique for Loss Minimization IEEE International Power Engineering and Optimization Conference, Haziran 2012, Melaka, Malezya.
17. Purwo, H., Muhammad, A., Ontoseno. P. ve Adi S., Optimal Placement and Sizing of Thyristor-controlled series-capacitor using Gravitational Search Algorithm TELKOMNIKA Indonesian Journal of Electrical Engineering Vol.10, 10, 5 (2012) 891-904.
18. Ali, M., Eltamaly, A., Yehia, S., Abou, H. ve Amer, A., Optimum Power Flow Analysis by Newton Raphson Method, A Case Study, *International Journal of Engineering*, 6, 4 (2018) 91-104.
19. Shaheen, H., Rashed, A. ve Cheng, S., Optimal location and parameters setting of Unified Power Flow Controller based on evolutionary optimization techniques IEEE PES General Meeting, Haziran. 2004, FL, USA.
20. Yaakov, W., Sergey, F. ve Nikolai A., Methods of reducing power losses in transmission systems MATEC Web of Conferences 1,50 (2017) 141-150.
21. Singh, J., Singh, N. ve Srivastava, H., Enhancement of Power System Security through Optimal Placement of TCSC and UPFC, IEEE Power Engineering Society General Meeting, 10,5 (2017) 1-6.

22. Srinivasa, P. ve Srivastava, R., Optimal Placement of TCSC Based on A Sensitivity Approach for Congestion Management Fifteenth National Power Systems Conference (NPSC), 15,2 (2008) 558-563.
23. Baldick, R. ve Kahn, E., Contract paths, phase shifters and efficient electricity trade, IEEE Transactions on Power Systems, 2,2 (1998) 749-755.
24. Bladow, J. ve Montoya, A., Experiences with parallel EHV phase shifting transformers, IEEE Transactions on Power Delivery Meeting, 2,2 (1991) 1096-1100.
25. Galiana, D., Assessment and control of the impact of FACTS devices on power system performance, IEEE Transactions on Power System, 11,4 (1996) 1931-1936.



7. EKLER

7.1. TCSC Modellenmesi

```
if(~isempty(tcsc_loc))  
for km=1:length(tcsc_loc)  
TCSC=R_size*reactance_value(tcsc_loc);  
reactance_value(tcsc_loc)=reactance_value(tcsc_loc)+TCSC;  
end
```

7.2. Amaç Fonksiyon

```
% objective  
% Active Power Loss  
ACTIVE_POWER_LOSS_WITH_TCSC=tcsc_case_result{1};  
f1=ACTIVE_POWER_LOSS_WITH_TCSC;  
  
% Voltage Profile Improvement  
volt_tcsc=tcsc_case_result{3};  
f2=sum(abs(1-volt_tcsc));  
  
% cost of installation of TCSC  
diffQ=abs(REACTIVE_POWER_LOSS_BASECASE(tcsc_loc)-  
REACTIVE_POWER_LOSS_WITH_TCSC(tcsc_loc));  
CTCSC=0.001*(diffQ.^2)-0.0713*diffQ+153.75;  
TCSC_COST=CTCSC*diffQ*1000;  
f3=TCSC_COST;  
  
h1=1;h2=0.5;h3=0;  
final_fitness=h1*f1+h2*f2+h3*f3;  
finalout{9}=final_fitness;
```

7.3. Newton Raphson Yük Akış Yöntemi Oluşturma

```
admitt_matrix_data=abmitt_imp_bus(bus_number,basevalue_inkv,basevalue_inmva,tcsc_lo  
c,R_size);  
global TCSC;  
[bus_data_infor,~]=bus_data_line_data_file(bus_number);  
base_mvain=basevalue_inmva*1000;  
bus=bus_data_infor(:,1);type_bus_in=bus_data_infor(:,2);  
voltage_bus=bus_data_infor(:,3);
```

```

volt_ang=bus_data_infor(:,4);

pload_value=bus_data_infor(:,5)/base_mvain;
qload_value=bus_data_infor(:,6)/base_mvain;
pload_value2=bus_data_infor(:,7)/base_mvain;
qload_value2=bus_data_infor(:,8)/base_mvain;

if(~isempty(qloaddata))

qloadloc=qloaddata(1);
qloadsize=qloaddata(2);

for km=1:length(qloadloc)

qload_value2(qloadloc(km))=qload_value2(qloadloc(km))+(qloadsize*1e3)/base_mvain;
end
end

final_pload=0.85*(pload_value-pload_value2);
final_qload=(qload_value-qload_value2);
final_pload_data=final_pload;
final_qload_data=final_qload;
admitt_gmat=real(admitt_matrix_data);
admitt_bmat=imag(admitt_matrix_data);
pvbus_loc=find(type_bus_in == 2 | type_bus_in == 1);
pqbus_loc= find(type_bus_in == 3);

npv=length(pvbus_loc);
number_of_pqbus=length(pqbus_loc);

toler_range = 1;
no_of_iter = 1;
while (toler_range > 1e-5 || no_of_iter<10 ) % Iteration starting..

    final_pload = zeros(bus_number,1);
    final_qload = zeros(bus_number,1);

    for loop1=1:bus_number
        for loop2=1:bus_number
            final_pload(loop1) = final_pload(loop1) + voltage_bus(loop1)*...
                voltage_bus(loop2)*(admitt_gmat(loop1,loop2)*cos(volt_ang(loop1)-
volt_ang(loop2)) +...
                admitt_bmat(loop1,loop2)*sin(volt_ang(loop1)-volt_ang(loop2)));
            final_qload(loop1) = final_qload(loop1) + voltage_bus(loop1)*...

```

```

        voltage_bus(loop2)*(admitt_gmat(loop1,loop2)*sin(volt_ang(loop1)-
voltage_ang(loop2)) -...
        admitt_bmat(loop1,loop2)*cos(volt_ang(loop1)-voltage_ang(loop2)));
    end
end

diff_ploadin=final_pload_data-final_pload;
diff_qloadin=final_qload_data-final_qload;
loop2 = 1;
qvald=zeros(number_of_pqbus,1);
for loop1 = 1:bus_number
    if type_bus_in(loop1) == 3
        qvald(loop2,1) = diff_qloadin(loop1);
        loop2 = loop2+1;
    end
end
diff_pdatapro=diff_ploadin(2:bus_number);
final_mmatrix=[diff_pdatapro; qvald];

jacob_mat1=zeros(bus_number-1,bus_number-1);
for loop1=1:(bus_number-1)
    m =loop1+1;
    for loop2 = 1:(bus_number-1)
        n = loop2+1;
        if n == m
            for n = 1:bus_number
                jacob_mat1(loop1,loop2) = jacob_mat1(loop1,loop2) + ...
                    voltage_bus(m)* voltage_bus(n)*(-admitt_gmat(m,n)*...
                    sin(voltage_ang(m)-voltage_ang(n)) + admitt_bmat(m,n)*cos(voltage_ang(m)-
voltage_ang(n)));
            end
            jacob_mat1(loop1,loop2) = jacob_mat1(loop1,loop2) -
voltage_bus(m)^2*admitt_bmat(m,m);
        else
            jacob_mat1(loop1,loop2) = voltage_bus(m)* voltage_bus(n)*...
                (admitt_gmat(m,n)*sin(voltage_ang(m)-voltage_ang(n)) - ...
                admitt_bmat(m,n)*cos(voltage_ang(m)-voltage_ang(n)));
        end
    end
end
end

jacob_mat2=zeros(bus_number-1,number_of_pqbus);
for loop1 = 1:(bus_number-1)
    m = loop1+1;
    for loop2 = 1:number_of_pqbus
        n=pqbus_loc(loop2);

```

```

if n == m
    for n = 1:bus_number
        jacob_mat2(loop1,loop2) = jacob_mat2(loop1,loop2) + ...
            voltage_bus(n)*(admitt_gmat(m,n)*cos(volt_ang(m)-volt_ang(n))...
            + admitt_bmat(m,n)*sin(volt_ang(m)-volt_ang(n)));
    end
    jacob_mat2(loop1,loop2) = jacob_mat2(loop1,loop2) +
voltage_bus(m)*admitt_gmat(m,m);
else
    jacob_mat2(loop1,loop2) = voltage_bus(m)*(admitt_gmat(m,n)*...
        cos(volt_ang(m)-volt_ang(n)) + admitt_bmat(m,n)*sin(volt_ang(m)-
voltage_ang(n)));
    end
end
end

jacob_mat3=zeros(number_of_pqbus,bus_number-1);
for loop1 = 1:number_of_pqbus
    m = pqbus_loc(loop1);
    for loop2 = 1:(bus_number-1)
        n = loop2+1;
        if n == m
            for n = 1:bus_number
                jacob_mat3(loop1,loop2)=jacob_mat3(loop1,loop2) + ...
                    voltage_bus(m)* voltage_bus(n)*(admitt_gmat(m,n)*...
                    cos(volt_ang(m)-volt_ang(n)) + admitt_bmat(m,n)*sin(volt_ang(m)-
voltage_ang(n)));
            end
            jacob_mat3(loop1,loop2) = jacob_mat3(loop1,loop2) -
voltage_bus(m)^2*admitt_gmat(m,m);
        else
            jacob_mat3(loop1,loop2) = voltage_bus(m)* voltage_bus(n)*...
                (-admitt_gmat(m,n)*cos(volt_ang(m)-volt_ang(n)) - ...
                admitt_bmat(m,n)*sin(volt_ang(m)-volt_ang(n)));
        end
    end
end
end

jacob_mat4=zeros(number_of_pqbus,number_of_pqbus);
for loop1 = 1:number_of_pqbus
    m = pqbus_loc(loop1);
    for loop2 = 1:number_of_pqbus
        n = pqbus_loc(loop2);
        if n == m
            for n = 1:bus_number
                jacob_mat4(loop1,loop2) = jacob_mat4(loop1,loop2) + ...
                    voltage_bus(n)*(admitt_gmat(m,n)*sin(volt_ang(m)-volt_ang(n)) - ...
                    admitt_bmat(m,n)*cos(volt_ang(m)-volt_ang(n)));
            end
        else
            jacob_mat4(loop1,loop2) = voltage_bus(m)*voltage_bus(n)*...
                (-admitt_gmat(m,n)*sin(volt_ang(m)-volt_ang(n)) - ...
                admitt_bmat(m,n)*cos(volt_ang(m)-volt_ang(n)));
        end
    end
end

```

```

        end
        jacob_mat4(loop1,loop2) = jacob_mat4(loop1,loop2) -
voltage_bus(m)*admitt_bmat(m,m);
    else
        jacob_mat4(loop1,loop2) = voltage_bus(m)*(admitt_gmat(m,n)*...
            sin(volt_ang(m)-volt_ang(n)) - admitt_bmat(m,n)*cos(volt_ang(m)-
volt_ang(n)));
    end
end
end
final_jacob_matrix=[jacob_mat1 jacob_mat2; jacob_mat3 jacob_mat4];
tmp_volt_th=(inv(final_jacob_matrix))*final_mmatrix;
thersh_dev_value=tmp_volt_th(1:bus_number-1);
volt_dev_value=tmp_volt_th(bus_number:end);
volt_ang(2:bus_number) = thersh_dev_value + volt_ang(2:bus_number);

% voltage_bus=volt_dev_value;
loop2 = 1;
for loop1 = 2:bus_number
    if type_bus_in(loop1) == 3
        voltage_bus(loop1) = volt_dev_value(loop2) + voltage_bus(loop1);
        loop2 = loop2+1;
    end
end
if(no_of_iter>10)
    break;
end

no_of_iter=no_of_iter+1;
toler_range=max(abs(final_mmatrix));
end
REACTIVE_POWER_LOSS_BASECASE=base_case_result{5};

finalout=flow_calculation_process(bus_number,voltage_bus,volt_ang,base_mvain,...
    basevalue_inkv,basevalue_inmva,tcsc_loc,R_size);

```

7.4. PSO Algoritmasını Oluştur

```

function [psoutput, optivoltprof]=PSO_PROCESS_FUNC(bus_number,D,mv,VR,...
    initial_population_size,...
    maxiter,psousedval,base_case_result,qloaddata)
psopara_val(2)=maxiter;% Maximum number of iterations
psopara_val(3)=initial_population_size; %%population size

kappa = 1;
phi1 = 2.05;
phi2 = 2.05;
phi = phi1 + phi2;

```

```

chi = 2*kappa/abs(2-phi-sqrt(phi^2-4*phi));

psopara_val(4)=phi1*chi ; % acceleration const 1
psopara_val(5)=phi2*chi ; % acceleration const 2
psopara_val(8)=1500; % Epoch when Ataletl weight at final value
psopara_val(9)=1e-10; % minimum global error gradient
psopara_val(10)=25000;%epochs before error gradient criterion terminates run
psopara_val(13)=0;

maxiter= psopara_val(2); % Maximum number of iterations
ps = psopara_val(3); %population size

ac1 = psopara_val(4); %acceleration const 1
ac2 = psopara_val(5); %acceleration const 2
iw1 = chi; % Initial Atalet weight

tr_gbestval = ones(1,maxiter)*NaN;
% setting max velocity and position params here
if length(mv)==1
    velmaskmin = -mv*ones(ps,D); % min vel, psXD matrix
    velmaskmax = mv*ones(ps,D); % max vel
elseif length(mv)==D
    velmaskmin = repmat(forcerow(-mv),ps,1); % min vel
    velmaskmax = repmat(forcerow( mv),ps,1); % max vel
end
posmaskmin = repmat(VR(1:D,1)',ps,1); % min pos, psXD matrix
posmaskmax = repmat(VR(1:D,2)',ps,1); % max pos
pos(1:ps,1:D) = normmat(rand([ps,D]),VR',1);
% construct initial random velocities between -mv,mv
vel(1:ps,1:D)=normmat(rand([ps,D]),[col_trans(-mv),col_trans(mv)]',1);
pbest=pos;

for km=1:initial_population_size
    tcsc_loc=round(pbest(km,1));
    R_size=(pbest(km,2));

    [finalout]=load_flow_newton_rapshon_withtcsc(bus_number,base_case_result,...
        tcsc_loc,R_size,qloaddata);

    out(km)=finalout{9};

end

pbestval=out;
[gbestval,idx1] = min(pbestval);
bestpos = zeros(maxiter,D+1)*NaN;
gbest = pbest(idx1,:);
bestpos(1,1:D) = gbest;
iwt(1) = iw1;

```

```

for iter_loop=1:maxiter % start epoch loop (iterations)

    resoutdata=[pos;gbest];

    [rsize,csize]=size(resoutdata);

    for km=1:rsize
        tcsc_loc=round(resoutdata(km,1));
        R_size=(resoutdata(km,2));

        [finalout]=load_flow_newton_rapshon_withtcsc(bus_number,base_case_result,...
            tcsc_loc,R_size,qloaddata);

        out(km)=finalout{9};
    end

    out = out(1:end-1);
    tr_gbestval(iter_loop+1) = gbestval; % keep track of global best val
    bestpos(iter_loop,1:D+1) = [gbest,gbestval];

    rstflg = 0;
    if rstflg == 0
        [tempi] = find(pbestval>=out); % new min pbestvals
        pbestval(tempi) = out(tempi); % update pbestvals
        pbest(tempi,:) = pos(tempi,:); % update pbest positions

        [iterbestval,idx1] = min(pbestval);
        if gbestval >= iterbestval
            gbestval = iterbestval;
            gbest = pbest(idx1(1),:);
        end
    end

    % get new velocities, positions
    % each epoch get new set of random numbers
    rannum1 = rand([ps,D]);
    rannum2 = rand([ps,D]);

    vel = iw1*vel+ac1.*(pbest-pos)... % independent
        +ac2.*(repmat(gbest,ps,1)-pos);

    % limit velocities here using masking
    vel = ( (vel <= velmaskmin).*velmaskmin ) + ( (vel > velmaskmin).*vel );
    vel = ( (vel >= velmaskmax).*velmaskmax ) + ( (vel < velmaskmax).*vel );

```

```

% update new position (PSO algo)
pos = pos + vel;

minposmask_throwaway = pos <= posmaskmin; % these are psXD matrices
minposmask_keep = pos > posmaskmin;
maxposmask_throwaway = pos >= posmaskmax;
maxposmask_keep = pos < posmaskmax;
pos = ( minposmask_throwaway.*posmaskmin ) + ( minposmask_keep.*pos );
pos = ( maxposmask_throwaway.*posmaskmax ) + ( maxposmask_keep.*pos );
vel = (vel.*minposmask_keep) + (-vel.*minposmask_throwaway);
vel = (vel.*maxposmask_keep) + (-vel.*maxposmask_throwaway);

end
psoutput=[gbest';gbestval; tr_gbestval'];

```

7.5. Ana Fonksiyon

```

clc
clear all
close all
warning off;
%%
bus_number=30;
[bus_data_in,line_data_in]=bus_data_line_data_file(bus_number);
FROM_BUS=line_data_in(:,1);
TO_BUS=line_data_in(:,2);
Line_reactance=line_data_in(:,4);
%% PSO algorithm
maxiter=30; % set number of iteration
initial_population_size=20; % set population size
%%
no_of_variables=2;
min_max_value_range=ones(no_of_variables,2); %set limits for random particle value
%% set limits of tcsc
max_line_number=length(FROM_BUS);
tcsc_loc_min=1;
tcsc_loc_max=max_line_number;
R_min=-0.8;
R_max=0.2;

%%
qloaddata=[];
[base_case_result]=load_flow_newton_rapshon_basecase(bus_number,qloaddata);
ACTIVE_POWER_LOSS_BASECASE=base_case_result{1}
REACTIVE_POWER_LOSS_BASECASE=base_case_result{2}
VOLTAGE_BASECASE=base_case_result{3};

```

```

LINE_LOSS_MVA=base_case_result{4};
LINE_LOSS_MVAr=base_case_result{5};
BASECASE_RESULT=table(FROM_BUS,TO_BUS,LINE_LOSS_MVA,LINE_LOSS_
MVAr)

% generate random value
for kr=1:1
    min_max_value_range(kr,1)=tcsc_loc_min; %set lower value
    min_max_value_range(kr,2)=tcsc_loc_max; %set upper value
end
for kr=2:no_of_variables
    min_max_value_range(kr,1)=R_min; %set lower value
    min_max_value_range(kr,2)=R_max; %set upper value
end
initial_pso_seed=zeros(initial_population_size,no_of_variables); % create zero matrix
% randmoly generate initial value
for kr=1:initial_population_size
    for kc=1:1
        initial_pso_seed(kr,kc)=tcsc_loc_min+((tcsc_loc_max-tcsc_loc_min)*rand(1,1));
    end
    for kc=2:2
        initial_pso_seed(kr,kc)=R_min+((R_max-R_min)*rand(1,1));
    end
end

% call pso function
qloaddata=[];

[psoutput]=PSO_PROCESS_FUNC(bus_number,no_of_variables,...
    2,min_max_value_range,...
    initial_population_size,maxiter,initial_pso_seed,base_case_result,qloaddata);
finalsolution_data=psoutput(1:no_of_variables);

tcsc_loc=round(finalsolution_data(1));
R_size=(finalsolution_data(2));
TCSC_size=R_size;%*Line_reactance(tcsc_loc);
[tcsc_case_result]=newton_rapshon_withtcscfinal(bus_number,base_case_result,...
    tcsc_loc,TCSC_size,qloaddata);% after find out TCSC
and calculate
ACTIVE_POWER_LOSS_WITH_TCSC=tcsc_case_result{1}
REACTIVE_POWER_LOSS_WITH_TCSC=tcsc_case_result{2}
VOLTAGE_WITH_TCSC=tcsc_case_result{3};
TCSC_COST=tcsc_case_result{7};
TCSC_LOCATION=tcsc_loc;
TCSC_size;
LINE_LOSS_MVA=tcsc_case_result{4};
LINE_LOSS_MVAr=tcsc_case_result{5};
NRcalculation01=0; NRcalculation11=0;
[finalout]=NRvaluecalculation(NRcalculation01,NRcalculation11);

```

```

%% fitness graph
fitness_powerloss=psooutput(end-(maxiter-1):end);
figure,plot(1:maxiter,fitness_powerloss);
xlabel('iteration');ylabel('fitness');grid on;
title('pso algorithm performance');
%% Q load CASE at 26
qvar_data=[5:5:30];
qcase=[ones(length(qvar_data),1)*26 qvar_data.'];
for kx=1:length(qvar_data)
    qloaddata=qcase(kx,:);
    [base_case_result1]=load_flow_newton_rapshon_basecase(bus_number,qloaddata);
    ACTIVE_POWER_LOSS_BASECASE1{kx,1}=num2str(base_case_result1{1});
    REACTIVE_POWER_LOSS_BASECASE1{kx,1}=num2str(base_case_result1{2});
    VOLTAGE_BASECASE1{kx}=base_case_result1{3};
    VOLTAGE_BASECASE11(kx)=base_case_result1{8};
    LINE_LOSS_MVA=base_case_result1{4};
    LINE_LOSS_MVAr=base_case_result1{5};
    BASECASE_RESULT=table(FROM_BUS,TO_BUS,LINE_LOSS_MVA,LINE_LOSS_MVAr);
    % generate random value
    for kr=1:1
        min_max_value_range(kr,1)=tcsc_loc_min; %set lower value
        min_max_value_range(kr,2)=tcsc_loc_max; %set upper value
    end
    for kr=2:no_of_variables
        min_max_value_range(kr,1)=R_min; %set lower value
        min_max_value_range(kr,2)=R_max; %set upper value
    end
    initial_pso_seed=zeros(initial_population_size,no_of_variables); % create zero matrix
    % randmoly generate initial value
    for kr=1:initial_population_size
        for kc=1:1
            initial_pso_seed(kr,kc)=tcsc_loc_min+((tcsc_loc_max-tcsc_loc_min)*rand(1,1));
        end
        for kc=2:2
            initial_pso_seed(kr,kc)=R_min+((R_max-R_min)*rand(1,1));
        end
    end
    % call pso function
    [psooutput]=PSO_PROCESS_FUNC(bus_number,no_of_variables,...
        2,min_max_value_range,...
        initial_population_size,maxiter,initial_pso_seed,base_case_result,qloaddata);
    finalsolution_data=psooutput(1:no_of_variables);
    tcsc_loc=round(finalsolution_data(1));
    R_size=(finalsolution_data(2));
    TCSC_size=R_size;%*Line_reactance(tcsc_loc);
    [tcsc_case_result]=newton_rapshon_withtcscfinal(bus_number,base_case_result,...
        tcsc_loc,TCSC_size,qloaddata);% after find out TCSC

```

```

ACTIVE_POWER_LOSS_WITH_TCSC1{kx,1}=num2str(tcsc_case_result{1});
REACTIVE_POWER_LOSS_WITH_TCSC1{kx,1}=num2str(tcsc_case_result{2});
VOLTAGE_WITH_TCSC1{kx}=tcsc_case_result{3};
TCSC_COST1(kx)=tcsc_case_result{7};
TCSC_LOCATION1{kx,1}=num2str(tcsc_loc);
TCSC_size1{kx,1}=num2str(TCSC_size);

VOLTAGE_WITH_TCSC01(kx)=tcsc_case_result{8};
LOAD_CONDITION_Qd26{kx,1}=num2str(qvar_data(kx));
FROM_BUS=line_data_in(:,1);
TO_BUS=line_data_in(:,2);
LINE_LOSS_MVA=tcsc_case_result{4};
LINE_LOSS_MVAr=tcsc_case_result{5};
end
VOLTAGE_WITH_TCSC01=finalout{1};
%%
ACTIVE_POWER_LOSS_WITHOUT_TCSC=ACTIVE_POWER_LOSS_BASECASE1;
REACTIVE_POWER_LOSS_WITHOUT_TCSC=REACTIVE_POWER_LOSS_BASECASE1;
ACTIVE_POWER_LOSS_WITH_TCSC=ACTIVE_POWER_LOSS_WITH_TCSC1;
REACTIVE_POWER_LOSS_WITH_TCSC=REACTIVE_POWER_LOSS_WITH_TCSC1;
TCSC_COST=TCSC_COST1;
TCSC_LOCATION=TCSC_LOCATION1;
TCSC_size=TCSC_size1;
figure,plot(qvar_data,VOLTAGE_BASECASE11,'r-s','linewidth',2);
hold on,plot(qvar_data,VOLTAGE_WITH_TCSC01,'g-^','linewidth',2);
xlabel('Qd(26)MVAr');ylabel('Voltage Profile');grid on;
title('IWPSO_Results for Voltage Profile When Load Variation at Bus 26');
legend('without TCSC','with TCSC PSO');
TRANSMISSION_LOSS_AT_BUS_26_USING_PSO_TECHNIQUE=...
table(LOAD_CONDITION_Qd26,ACTIVE_POWER_LOSS_WITHOUT_TCSC,REACTIVE_POWER_LOSS_WITHOUT_TCSC,...
ACTIVE_POWER_LOSS_WITH_TCSC,REACTIVE_POWER_LOSS_WITH_TCSC,...
TCSC_LOCATION,TCSC_size)

%%
%% Q load CASE at 29

qvar_data=[5:5:30];
qcase=[ones(length(qvar_data),1)*29 qvar_data.'];

for kx=1:length(qvar_data)

qloaddata=qcase(kx,:);
[base_case_result1]=load_flow_newton_rapshon_basecase(bus_number,qloaddata);

```

```

ACTIVE_POWER_LOSS_BASECASE1{kx,1}=num2str(base_case_result1{1});
REACTIVE_POWER_LOSS_BASECASE1{kx,1}=num2str(base_case_result1{2});
VOLTAGE_BASECASE1{kx}=base_case_result1{3};
VOLTAGE_BASECASE11(kx)=base_case_result1{8};

LINE_LOSS_MVA=base_case_result1{4};
LINE_LOSS_MVA_r=base_case_result1{5};
BASECASE_RESULT=table(FROM_BUS,TO_BUS,LINE_LOSS_MVA,LINE_LOSS_
MVA_r);

% generate random value
for kr=1:1
    min_max_value_range(kr,1)=tcsc_loc_min; %set lower value
    min_max_value_range(kr,2)=tcsc_loc_max; %set upper value
end
for kr=2:no_of_variables
    min_max_value_range(kr,1)=R_min; %set lower value
    min_max_value_range(kr,2)=R_max; %set upper value
end
initial_pso_seed=zeros(initial_population_size,no_of_variables); % create zero matrix
% randmoly generate initial value
for kr=1:initial_population_size
    for kc=1:1
        initial_pso_seed(kr,kc)=tcsc_loc_min+((tcsc_loc_max-tcsc_loc_min)*rand(1,1));
    end
    for kc=2:2
        initial_pso_seed(kr,kc)=R_min+((R_max-R_min)*rand(1,1));
    end
end

% call pso function
[psooutput]=PSO_PROCESS_FUNC(bus_number,no_of_variables,...
    2,min_max_value_range,...
    initial_population_size,maxiter,initial_pso_seed,base_case_result,qloaddata);
finalsolution_data=psooutput(1:no_of_variables);

tcsc_loc=round(finalsolution_data(1));
R_size=(finalsolution_data(2));
TCSC_size=R_size;%*Line_reactance(tcsc_loc);
[tcsc_case_result]=newton_rapshon_withtcscfinal(bus_number,base_case_result,...
    tcsc_loc,TCSC_size,qloaddata);% after find out TCSC
ACTIVE_POWER_LOSS_WITH_TCSC1{kx,1}=num2str(tcsc_case_result{1});
REACTIVE_POWER_LOSS_WITH_TCSC1{kx,1}=num2str(tcsc_case_result{2});
VOLTAGE_WITH_TCSC1{kx}=tcsc_case_result{3};
TCSC_COST1(kx)=tcsc_case_result{7};
TCSC_LOCATION1{kx,1}=num2str(tcsc_loc);
TCSC1{kx,1}=num2str(TCSC_size);
VOLTAGE_WITH_TCSC11(kx)=tcsc_case_result{8};
LOAD_CONDITION_Qd29{kx,1}=num2str(qvar_data(kx));

```

```

FROM_BUS=line_data_in(:,1);
TO_BUS=line_data_in(:,2);
LINE_LOSS_MVA=tcsc_case_result{4};
LINE_LOSS_MVA_r=tcsc_case_result{5};
end
VOLTAGE_WITH_TCSC11=finalout{2};
%%
ACTIVE_POWER_LOSS_WITHOUT_TCSC=ACTIVE_POWER_LOSS_BASECASE1;
REACTIVE_POWER_LOSS_WITHOUT_TCSC=REACTIVE_POWER_LOSS_BASECASE1;
ACTIVE_POWER_LOSS_WITH_TCSC=ACTIVE_POWER_LOSS_WITH_TCSC1;
REACTIVE_POWER_LOSS_WITH_TCSC=REACTIVE_POWER_LOSS_WITH_TCSC1;
TCSC_COST=TCSC_COST1;
TCSC_LOCATION=TCSC_LOCATION1;
TCSC_size=TCSC1;

figure,plot(qvar_data,VOLTAGE_BASECASE11,'r-s','linewidth',2);
hold on,plot(qvar_data,VOLTAGE_WITH_TCSC11,'g-^','linewidth',2);
xlabel('Qd(29)MVA_r');ylabel('Voltage Profile');grid on;
title('IWPSO_Results for Voltage Profile When Load Variation at Bus 29');
legend('without TCSC','with TCSC PSO');
TRANSMISSION_LOSS_AT_BUS_29_USING_PSO_TECHNIQUE=...
table(Load_CONDITION_Qd29,ACTIVE_POWER_LOSS_WITHOUT_TCSC,REACTIVE_POWER_LOSS_WITHOUT_TCSC,...

ACTIVE_POWER_LOSS_WITH_TCSC,REACTIVE_POWER_LOSS_WITH_TCSC,...
TCSC_LOCATION,TCSC_size)

Newton Rapshon fuction:

function [finalout]=load_flow_newton_rapshon_withtcsc(bus_number,base_case_result,...
tcsc_loc,R_size,qloaddata)

basevalue_inkv=12.66;
basevalue_inmva=100;

admitt_matrix_data=abmitt_imp_bus(bus_number,basevalue_inkv,basevalue_inmva,tcsc_loc,R_size);
global TCSC;
[bus_data_infor,~]=bus_data_line_data_file(bus_number);
base_mvain=basevalue_inmva*1000;
bus=bus_data_infor(:,1);type_bus_in=bus_data_infor(:,2);
voltage_bus=bus_data_infor(:,3);
volt_ang=bus_data_infor(:,4);

pload_value=bus_data_infor(:,5)/base_mvain;
qload_value=bus_data_infor(:,6)/base_mvain;
pload_value2=bus_data_infor(:,7)/base_mvain;

```

```

qload_value2=bus_data_infor(:,8)/base_mvain;

if(~isempty(qloaddata))

qloadloc=qloaddata(1);
qloadsize=qloaddata(2);

for km=1:length(qloadloc)

qload_value2(qloadloc(km))=qload_value2(qloadloc(km))+(qloadsize*1e3)/base_mvain;
end
end

final_pload=0.85*(pload_value-pload_value2);
final_qload=(qload_value-qload_value2);
final_pload_data=final_pload;
final_qload_data=final_qload;
admitt_gmat=real(admitt_matrix_data);
admitt_bmat=imag(admitt_matrix_data);
pvbus_loc=find(type_bus_in == 2 | type_bus_in == 1);
pqbus_loc= find(type_bus_in == 3);

npv=length(pvbus_loc);
number_of_pqbus=length(pqbus_loc);

toler_range = 1;
no_of_iter = 1;
while (toler_range > 1e-5 || no_of_iter<10 ) % Iteration starting..

    final_pload = zeros(bus_number,1);
    final_qload = zeros(bus_number,1);

    for loop1=1:bus_number
        for loop2=1:bus_number
            final_pload(loop1) = final_pload(loop1) + voltage_bus(loop1)*...
                voltage_bus(loop2)*(admitt_gmat(loop1,loop2)*cos(volt_ang(loop1)-
volt_ang(loop2)) +...
                admitt_bmat(loop1,loop2)*sin(volt_ang(loop1)-volt_ang(loop2)));
            final_qload(loop1) = final_qload(loop1) + voltage_bus(loop1)*...
                voltage_bus(loop2)*(admitt_gmat(loop1,loop2)*sin(volt_ang(loop1)-
volt_ang(loop2)) -...
                admitt_bmat(loop1,loop2)*cos(volt_ang(loop1)-volt_ang(loop2)));
        end
    end
end

```

```

diff_ploadin=final_pload_data-final_pload;
diff_qloadin=final_qload_data-final_qload;
loop2 = 1;
qvald=zeros(number_of_pqbus,1);
for loop1 = 1:bus_number
    if type_bus_in(loop1) == 3
        qvald(loop2,1) = diff_qloadin(loop1);
        loop2 = loop2+1;
    end
end
diff_pdatapro=diff_ploadin(2:bus_number);
final_mmatrix=[diff_pdatapro; qvald];

```

```

jacob_mat1=zeros(bus_number-1,bus_number-1);
for loop1=1:(bus_number-1)
    m =loop1+1;
    for loop2 = 1:(bus_number-1)
        n = loop2+1;
        if n == m
            for n = 1:bus_number
                jacob_mat1(loop1,loop2) = jacob_mat1(loop1,loop2) + ...
                    voltage_bus(m)* voltage_bus(n)*(-admitt_gmat(m,n)*...
                        sin(volt_ang(m)-volt_ang(n)) + admitt_bmat(m,n)*cos(volt_ang(m)-
voltage_ang(n)));
            end
            jacob_mat1(loop1,loop2) = jacob_mat1(loop1,loop2) -
voltage_bus(m)^2*admitt_bmat(m,m);
        else
            jacob_mat1(loop1,loop2) = voltage_bus(m)* voltage_bus(n)*...
                (admitt_gmat(m,n)*sin(volt_ang(m)-volt_ang(n)) - ...
                    admitt_bmat(m,n)*cos(volt_ang(m)-volt_ang(n)));
        end
    end
end
end

```

```

jacob_mat2=zeros(bus_number-1,number_of_pqbus);
for loop1 = 1:(bus_number-1)
    m = loop1+1;
    for loop2 = 1:number_of_pqbus
        n=pqbus_loc(loop2);
        if n == m
            for n = 1:bus_number
                jacob_mat2(loop1,loop2) = jacob_mat2(loop1,loop2) + ...
                    voltage_bus(n)*(admitt_gmat(m,n)*cos(volt_ang(m)-volt_ang(n))...
                        + admitt_bmat(m,n)*sin(volt_ang(m)-volt_ang(n)));
            end
        end
    end
end

```

```

        jacob_mat2(loop1,loop2) = jacob_mat2(loop1,loop2) +
voltage_bus(m)*admitt_gmat(m,m);
    else
        jacob_mat2(loop1,loop2) = voltage_bus(m)*(admitt_gmat(m,n)*...
            cos(volt_ang(m)-volt_ang(n)) + admitt_bmat(m,n)*sin(volt_ang(m)-
voltage_ang(n)));
    end
end
end

jacob_mat3=zeros(number_of_pqbus,bus_number-1);
for loop1 = 1:number_of_pqbus
    m = pqbus_loc(loop1);
    for loop2 = 1:(bus_number-1)
        n = loop2+1;
        if n == m
            for n = 1:bus_number
                jacob_mat3(loop1,loop2)=jacob_mat3(loop1,loop2) + ...
                    voltage_bus(m)* voltage_bus(n)*(admitt_gmat(m,n)*...
                        cos(volt_ang(m)-volt_ang(n)) + admitt_bmat(m,n)*sin(volt_ang(m)-
voltage_ang(n)));
            end
            jacob_mat3(loop1,loop2) = jacob_mat3(loop1,loop2) -
voltage_bus(m)^2*admitt_gmat(m,m);
        else
            jacob_mat3(loop1,loop2) = voltage_bus(m)* voltage_bus(n)*...
                (-admitt_gmat(m,n)*cos(volt_ang(m)-volt_ang(n)) - ...
                    admitt_bmat(m,n)*sin(volt_ang(m)-volt_ang(n)));
        end
    end
end
end

jacob_mat4=zeros(number_of_pqbus,number_of_pqbus);
for loop1 = 1:number_of_pqbus
    m = pqbus_loc(loop1);
    for loop2 = 1:number_of_pqbus
        n = pqbus_loc(loop2);
        if n == m
            for n = 1:bus_number
                jacob_mat4(loop1,loop2) = jacob_mat4(loop1,loop2) + ...
                    voltage_bus(n)*(admitt_gmat(m,n)*sin(volt_ang(m)-volt_ang(n)) - ...
                        admitt_bmat(m,n)*cos(volt_ang(m)-volt_ang(n)));
            end
            jacob_mat4(loop1,loop2) = jacob_mat4(loop1,loop2) -
voltage_bus(m)*admitt_bmat(m,m);
        else
            jacob_mat4(loop1,loop2) = voltage_bus(m)*(admitt_gmat(m,n)*...

```

```

        sin(volt_ang(m)-volt_ang(n)) - admitt_bmat(m,n)*cos(volt_ang(m)-
volt_ang(n)));
    end
    end
    end
    final_jacob_matrix=[jacob_mat1 jacob_mat2; jacob_mat3 jacob_mat4];
    tmp_volt_th=(inv(final_jacob_matrix))*final_mmatrix;
    thersh_dev_value=tmp_volt_th(1:bus_number-1);
    volt_dev_value=tmp_volt_th(bus_number:end);
    volt_ang(2:bus_number) = thersh_dev_value + volt_ang(2:bus_number);

%   voltage_bus=volt_dev_value;
    loop2 = 1;
    for loop1 = 2:bus_number
        if type_bus_in(loop1) == 3
            voltage_bus(loop1) = volt_dev_value(loop2) + voltage_bus(loop1);
            loop2 = loop2+1;
        end
    end
    if(no_of_iter>10)
        break;
    end

    no_of_iter=no_of_iter+1;
    toler_range=max(abs(final_mmatrix));
end

```

```

REACTIVE_POWER_LOSS_BASECASE=base_case_result{5};

```

```

finalout=flow_calculation_process(bus_number,voltage_bus,volt_ang,base_mvain,...
    basevalue_inkv,basevalue_inmva,tcsc_loc,R_size);

```

```

tcsc_case_result=finalout;
REACTIVE_POWER_LOSS_WITH_TCSC=tcsc_case_result{5};
finalout{8}=mean(2-finalout{3});
%% objective
% Active Power Loss
ACTIVE_POWER_LOSS_WITH_TCSC=tcsc_case_result{1};
f1=ACTIVE_POWER_LOSS_WITH_TCSC;

```

```

% Voltage Profile Improvement
volt_tcsc=tcsc_case_result{3};
f2=sum(abs(1-volt_tcsc));

```

```

% cost of installation of TCSC
diffQ=abs(REACTIVE_POWER_LOSS_BASECASE(tcsc_loc)-
    REACTIVE_POWER_LOSS_WITH_TCSC(tcsc_loc));

```

```
CTCSC=0.001*(diffQ.^2)-0.0713*diffQ+153.75;
TCSC_COST=CTCSC*diffQ*1000;
f3=TCSC_COST;
```

```
h1=0.5;h2=0.3;h3=0.5;
final_fitness=h1*f1+h2*f2+h3*f3;
finalout{9}=final_fitness;
finalout{7}=TCSC_COST;
```

PSO Fuction:

```
function [psoutput, optivoltprof]=PSO_PROCESS_FUNC(bus_number,D,mv,VR,...
    initial_population_size,...
    maxiter,psseedval,base_case_result,qloaddata)
```

```
psopara_val(2)=maxiter;% Maximum number of iterations
psopara_val(3)=initial_population_size; %%population size
psopara_val(4)=.1; % acceleration const 1
psopara_val(5)=.1; % acceleration const 2
psopara_val(6)=.9; % Initial Atalet weight
psopara_val(7)=.4; % Final Atalet weight
psopara_val(8)=1500; % Epoch when Ataletl weight at final value
psopara_val(9)=1e-10; % minimum global error gradient
psopara_val(10)=25000;%epochs before error gradient criterion terminates run
psopara_val(13)=0;
```

```
maxiter= psopara_val(2); % Maximum number of iterations
ps = psopara_val(3); %population size
ac1 = psopara_val(4); %acceleration const 1
ac2 = psopara_val(5); %acceleration const 2
iw1 = psopara_val(6); % Initial Atalet weight
iw2 = psopara_val(7); % Final Atalet weight
iwe = psopara_val(8); % Epoch when Ataletl weight at final value
tr_gbestval = ones(1,maxiter)*NaN;
```

% setting max velocity and position params here

```
if length(mv)==1
```

```
    velmaskmin = -mv*ones(ps,D); % min vel, psXD matrix
```

```
    velmaskmax = mv*ones(ps,D); % max vel
```

```
elseif length(mv)==D
```

```
    velmaskmin = repmat(forcerow(-mv),ps,1); % min vel
```

```
    velmaskmax = repmat(forcerow( mv),ps,1); % max vel
```

```
end
```

```
posmaskmin = repmat(VR(1:D,1)',ps,1); % min pos, psXD matrix
```

```
posmaskmax = repmat(VR(1:D,2)',ps,1); % max pos
```

```
pos(1:ps,1:D) = normmat(rand([ps,D]),VR',1);
```

% construct initial random velocities between -mv,mv

```
vel(1:ps,1:D)=normmat(rand([ps,D]),[col_trans(-mv),col_trans(mv)]',1);
```

```
pbest=pos;
```

```
for km=1:initial_population_size
```

```

    tcsc_loc=round(pbest(km,1));
    R_size=(pbest(km,2));

[finalout]=load_flow_newton_rapshon_withtcsc(bus_number,base_case_result,...
    tcsc_loc,R_size,qloaddata);

out(km)=finalout{9};

end

pbestval=out;
[gbestval,idx1] = min(pbestval);
bestpos = zeros(maxiter,D+1)*NaN;
gbest = pbest(idx1,:);
bestpos(1,1:D) = gbest;
iwt(1) = iwt1;

for iter_loop=1:maxiter % start epoch loop (iterations)

    resoutdata=[pos;gbest];

    [rsize,csz]=size(resoutdata);

    for km=1:rsize
        tcsc_loc=round(resoutdata(km,1));
        R_size=(resoutdata(km,2));

        [finalout]=load_flow_newton_rapshon_withtcsc(bus_number,base_case_result,...
            tcsc_loc,R_size,qloaddata);

        out(km)=finalout{9};
    end

    out = out(1:end-1);
    tr_gbestval(iter_loop+1) = gbestval; % keep track of global best val
    bestpos(iter_loop,1:D+1) = [gbest,gbestval];

    rstflg = 0;
    if rstflg == 0
        [tempi] = find(pbestval>=out); % new min pbestvals
        pbestval(tempi) = out(tempi); % update pbestvals
        pbest(tempi,:) = pos(tempi,:); % update pbest positions

        [iterbestval,idx1] = min(pbestval);
        if gbestval >= iterbestval
            gbestval = iterbestval;

```

```

        gbest = pbest(idx1(1),:);
    end
end
% get new velocities, positions
% each epoch get new set of random numbers
rannum1 = rand([ps,D]);
rannum2 = rand([ps,D]);

% get Atalet weight
if iter_loop<=iwe
    iwt(iter_loop) = ((iw2-iw1)/(iwe-1))*(iter_loop-1)+iw1;
else
    iwt(iter_loop) = iw2;
end

% random number including acceleration constants
ac11 = rannum1.*ac1;
ac22 = rannum2.*ac2;

vel = iwt(iter_loop)*vel+ac11.*(pbest-pos)... % independent
    +ac22.*( repmat(gbest,ps,1)-pos);

% limit velocities here using masking
vel = ( (vel <= velmaskmin).*velmaskmin ) + ( (vel > velmaskmin).*vel );
vel = ( (vel >= velmaskmax).*velmaskmax ) + ( (vel < velmaskmax).*vel );

% update new position (PSO algo)
pos = pos + vel;

minposmask_throwaway = pos <= posmaskmin; % these are psXD matrices
minposmask_keep = pos > posmaskmin;
maxposmask_throwaway = pos >= posmaskmax;
maxposmask_keep = pos < posmaskmax;
pos = ( minposmask_throwaway.*posmaskmin ) + ( minposmask_keep.*pos );
pos = ( maxposmask_throwaway.*posmaskmax ) + ( maxposmask_keep.*pos );
vel = (vel.*minposmask_keep) + (-vel.*minposmask_throwaway);
vel = (vel.*maxposmask_keep) + (-vel.*maxposmask_throwaway);

end

```

ÖZGEÇMİŞ

2011 yılında Çinin Şendong eyaletinde bulunan Muping lisessini tamamladıktan sonra aynı yılda Chunchun üniversitesini kazandı, 2015 yılında ChangChun Üniversitesi Ölçüm ve Kontrol Teknolojisi ve enstrümanları bölümünden mezun oldu, 2016 yılında Pekin Ormancılık Üniversitesi Kontrol Mühendisliği bölümünde yüksek lisans kazandı aynı zamanda lisans çalışmalarından dolayı Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı (YTB) tarafından Karadeniz Teknik Üniversitesinde Yüksek lisans eğitimi alma fırsatı buldu ve Türkiye'ye gelmeye karar verdi ,2017 haziranda Türkçe eğitimini tamamladı,2018 haziranda yüksek lisans zorunla derslerini tamamladı,2018 temmuzdan 2018 Aralığa kadar Altinova Elektronik firmasında çalıştı,2019 ocaktan mayısa kadar Sieyuan Elektrik ile EMTA Şirketi beraber yaptığı Kayseri Seri Kapasitör projesinde yardımcı proje sorumlusu olarak çalıştı,2019 Mayıstan temmuza kadar ODTÜ Tekno kente yerleşimi olan Nanomanyatik Bilimsel Cihazları Şirketinde Test Mühendisi olarak çalıştı,2019 eylülünden şüana kadar Singapur TDCX şirketinde çalışmaktadır, kendisi Türkçe, Uygurca, İngilizce ve Çince konuşabilmektedir.