



Büyük ölçekli konveks ve konveks olmayan ekonomik güç dağıtım problemleri için artırılmış sosyal öğrenme tabanlı diferansiyel gelişim algoritmaları

Incremental social learning-based differential evolution algorithms for large-scale convex and nonconvex economic power dispatch problems

Serdar Özyön^{1,*} 

¹ Kütahya Dumlupınar Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, 43100, Kütahya Türkiye

Öz

Güç sistemleri mühendisliğinin önemli problemlerinden biri olan, ekonomik güç dağıtım problemi, mevcut sistemin optimum işletimi için hızlı bir şekilde çözülmesi gereken bir problemdir. Bu problem literatürde, sistemdeki yüklerin farklı kısıtlar altında bütün güç üretim birimleri tarafından en az maliyetle karşılanması şeklinde tanımlanır. Farklı birçok çalışmada, farklı birçok güç sistemi için probleme çözüm aranmıştır. Güç sistemleri yapı ve kısıtlar bakımından birbirlerinden farklı olduklarından problem güncelliğini korumaktadır. Bu çalışmada, modifiye edilmiş IEEE 118 bara 54 generatörlü büyük ölçekli bir güç sistemi için hem konveks yakıt maliyeti fonksiyonları için hem de konveks olmayan yakıt maliyeti fonksiyonları için probleminin çözümü yapılmıştır. Örnek güç sisteminde iletim hattı kayıpları dikkate alınmış ve bu kayıpların hesaplanabilmesi için Newton-Raphson metodu kullanılarak AC yük akışı analizi yapılmıştır. Farklı yapıdaki her iki problemde, minimum yakıt maliyeti değerlerinin hesaplanabilmesi için diferansiyel gelişim algoritmasına artırılmış sosyal öğrenme yapısı beş farklı şekilde entegre edilerek yeni artırılmış sosyal öğrenme tabanlı diferansiyel gelişim algoritmaları geliştirilmiş ve uygulanmıştır. Altı farklı algoritma ile yapılan çözümlerde elde edilen sayısal değerler karşılaştırılarak, problemlere özgü en uygun metod belirlenmeye çalışılmıştır. Bütün sonuçlar farklı tablo, grafik ve şekillerle verilmiş ve değerlendirilmiştir. Çalışmada iletim hattı kayıpları Newton-Raphson temelli AC yük akışı ile gerçekçi şekilde hesaba katılmış, ayrıca klasik diferansiyel gelişim algoritmasına artırılmış sosyal öğrenme yapısı entegre edilerek çözüm kalitesi ve algoritma kararlılığı artırılarak literatüre katkı sağlanmıştır.

Anahtar kelimeler: AC yük akışı analizi, Konveks ve konveks olmayan ekonomik güç dağıtım, Büyük ölçekli güç sistemleri, Artırılmış sosyal öğrenme, Diferansiyel gelişim algoritması.

1 Giriş

Elektrik enerjisi üretimi ve dağıtım, modern güç sistemlerinde hem ekonomik hem de teknik açıdan optimize edilmesi gereken karmaşık bir süreçtir. Bu sürecin en kritik adımlarından biri, mevcut üretim kaynaklarının sistem

Abstract

The economic power dispatch problem, which is one of the important problems of power systems engineering, is a problem that must be solved quickly for the optimum operation of the existing system. This problem is defined in the literature as the ability to meet the loads in the system at the least cost by all power generation units under different constraints. In many different studies, solutions to the problem have been sought for many different power systems. Since power systems differ from each other in terms of structure and constraints, the problem remains current. This study solves the economic power dispatch problem for both convex and non-convex fuel cost functions for a large-scale power system with modified IEEE 118 bus 54 generators. Transmission line losses were considered in the sample power system, and AC load flow analysis was performed using the Newton-Raphson method to calculate these losses. In both problems with different structures, the incremental social learning structure was integrated into the differential evolution algorithm in five different ways to calculate the minimum fuel cost values, and new incremental social learning-based differential evolution algorithms were developed and applied. By comparing the numerical values obtained in solutions made with six different algorithms, the most appropriate method specific to the problems was determined. All results are evaluated using tables, graphs, and figures. In this study, transmission line losses are realistically taken into account with Newton-Raphson based AC load flow and incremental social learning structures are integrated into the classical differential evolution algorithm to improve solution quality and algorithm stability.

Keywords: AC load flow analysis, Convex and non-convex economic power dispatch, Large-scale power systems, Incremental social learning, Differential evolution algorithm.

kısıtları altında en düşük maliyetle işletilmesini sağlayan ekonomik güç dağıtım (Economic Power Dispatch - EPD) problemidir. Özellikle büyük ölçekli güç sistemlerinde, üretim birimlerinin sayısının ve iletim hattı karmaşıklığının artması, EPD probleminin çözümünü daha da zorlaştırmakta

* Sorumlu yazar / Corresponding author, e-posta / e-mail: serdar.ozyon@dpu.edu.tr (S. Özyön)

Geliş / Received: 03.01.2025 Kabul / Accepted: 27.04.2025 Yayınlanma / Published: 15.07.2025

doi: 10.28948/ngumuh.1612958

ve geleneksel yöntemlerin yetersiz kalmasına neden olmaktadır. Ayrıca, konveks olmayan yakıt maliyeti fonksiyonları ve iletim hattı kayıplarının gerçekçi şekilde modellenmesi, problemin çözümünü daha karmaşık hale getirmektedir [1].

Literatürde elektrik enerjisi üretiminin daha verimli yapılabilmesi konusunda farklı birçok yaklaşım bulunmaktadır. Bunlardan biri de EPD yöntemidir. EPD, sistem tarafından talep edilen güç miktarının, sistem kısıtlarına uyarak, farklı üretim maliyeti değerlerine sahip birimler tarafından en az yakıt maliyetiyle üretilmesi şeklinde tanımlanabilir. EPD, güç sistemlerinde önemli bir problemdir. Probleme, literatürde farklı kısıtlar tanımlanarak ve farklı güç sistemlerine uygulanarak birçok çözüm aranmıştır. Problemde fosil yakıt maliyetinin en aza indirilmesi amaçlandığı için problem en temel optimizasyon problemlerinden biri olarak kabul edilir. EPD problemi konveks ve konveks olmayan yapılarla birçok şekilde tasarlanabilmektedir. Büyük ölçekli güç sistemlerinde EPD probleminin çözümü matematiksel metotlarla oldukça zor ve uzun süreler almaktadır. Bu nedenle problemin çözümünde optimizasyon algoritmaları yaygın olarak kullanılmaktadır [2].

Optimizasyon, çözümü makul sürelerde zor ya da imkânsız olan problemlerin çözümünde yaygın olarak kullanılan, sonucunda kabul edilebilir bir çözüm seti öneren bir metottur. Eşitlikleri, eşitsizlikleri ve amaç fonksiyonu matematiksel olarak belirlenen problemler için optimizasyon algoritmaları ile makul sürelerde kabul edilebilir çözümler alınabilir. Optimizasyon çalışmaları için çeşitli fiziksel ve biyolojik olaylardan esinlenilerek birçok algoritma geliştirilmiş ve geliştirilmeye devam etmektedir. Bunlardan en temel olanları genetik algoritma, diferansiyel gelişim algoritması (Differential Evolution - DE) ve parçacık sürü optimizasyonu olarak belirtilebilir. Ortaya atılan bu algoritmalar ilerleyen zamanlarda ortaya atılan birçok metotla da iyileştirilmeye çalışılmışlardır [1]. Algoritma iyileştirme metotlarından biri de artırılmış sosyal öğrenme (Incremental Social Learning-ISL) kavramıdır. ISL temel olarak popülasyon tabanlı meta-sezgisel algoritmalar için önerilmiş ve popülasyonda artırım yapılacak adıma kadar kazanılan öğrenmenin (toplumsal hafıza) kaybedilmemesine yönelik bir metottur.

Literatürde birçok sezgisel ve meta-sezgisel optimizasyon yöntemi EPD probleminin çözümünde uygulanmıştır. Ancak literatürde yer alan bu çalışmaların büyük bir kısmı, iletim hattı kayıplarını ya ihmal etmiş ya da yaklaşık yöntemlerle ele almıştır. Bu da gerçek sistem davranışının yeterince temsil edilmesini engellemiştir. Ayrıca çoğu çalışmada sabit boyutlu popülasyonlar ile işlem yapılmış ve bu durum çözüm uzayının etkin taranmasını sınırlamıştır. Önerilen çalışmada bu eksikliklerin giderilmesi amaçlanmıştır. İletim hattı kayıpları Newton-Raphson temelli AC yük akışı ile gerçekçi şekilde hesaba katılmış, ayrıca klasik DE algoritmasına ISL yapıları entegre edilerek çözüm kalitesinin ve algoritma kararlılığının artırılması hedeflenmiştir. Çalışma için iki farklı literatür taraması yapılmıştır. Bunlardan ilki son dönemde EPD probleminin çözümü için uygulanan sezgisel algoritmalarla alakalı diğeri

ise ISL metodunun uygulamasının yapıldığı çalışmalar özetlenmiştir. Literatürde EPD probleminin çözümü için adaptif hibrit geri izlemeli arama algoritması [3], hibrit biyocoğrafya ve diferansiyel gelişim tabanlı optimizasyon metodu [4], hibrit mutasyon stratejili karşıt-mutual öğrenme tabanlı diferansiyel gelişim algoritması [5], hibrit haris şahini optimizasyonu [6], geliştirilmiş sosyal ağ arama algoritması [7], geliştirilmiş kartal-stratejisi arz-talep optimizasyonu [8], ikili öğrenme stratejisine sahip uyarlanabilir geri izleme arama optimizasyon algoritması [9], hibrit çekirge optimizasyon algoritması ile borsa algoritması [10], hibrit çok evrenli optimizasyon metodu [11], ikili zig-zag ve genel tamsayı zig-zag metodlarıyla [12], takviyeli öğrenme tabanlı modifiye guguk kuşu arama algoritması [13], geliştirilmiş iki manta vatozu arama optimizasyonu algoritması [14,15], geliştirilmiş beluga balinası optimizasyonu [16], kısıtlı işbirlikçi uyarlanabilir çok popülasyonlu diferansiyel gelişim algoritması [17], yerçekimi karşıt tabanlı bulanık bulanık yerçekimsel arama algoritması [18], hibrit mutasyon stratejili diferansiyel gelişim algoritması [19], çok gruplu bir deniz avcısı algoritması [20], yenilikçi bir hibrit algoritması [21], karşıt güvercin optimizasyonu algoritması [22], hibrit bilgi edinme-paylaşma diferansiyel gelişim algoritması [23], gradyan arama algoritması ile hibrit bir kapuçin arama algoritması [24], kaotik balçık kalıbı algoritması [25], rulet tekerleği ve Levy uçuş yöntemleri ile bukalemun sürüsü algoritması [26] ve kısıtlı Laplacian biyocoğrafya tabanlı optimizasyon algoritması [27] kullanılmıştır. EPD probleminin çözümü için verilen çalışmalar ayrıntılı olarak incelendiğinde iki farklı durum göze çarpmaktadır. Birincisi çalışmaların birçoğu ya klasik algoritmaların geliştirilmesi üzerine ya da farklı algoritmaların birlikte kullanımlarına yönelik çalışmalardır. İkinci husus ise çalışmalarda ele alınan EPD problemlerinde ya iletim hattı kayıpları ihmal edilmiş ya da bu kayıpların hesaplanmasında yaklaşık değerler veren B kayıp matrisleri kullanılmıştır. Bu durum problemin gerçek Dünya mühendislik problemlerine yakınsamasını engellenmektedir. Bu çalışmada ise iletim hattı kayıpları için AC yük akışı metodu kullanılmış ve iletim hattı kayıpları gerçek değerlerine yakın olarak ele alınmıştır.

Yapılan ikinci literatür taraması ise çalışmada DE algoritması ile birlikte kullanılan ISL metodunun, daha önce hangi algoritmalarla birlikte kullanıldığı ve hangi problem yapılarına uygulandığı üzerinedir. Literatürde ISL metodu, test fonksiyonları için parçacık sürü optimizasyonu algoritmasına [28], yine test fonksiyonları için karınca kolonisi algoritmasına [29], farklı EPD problemleri için yapay arı kolonisi algoritmasına [30-32], test fonksiyonları ve hidrotermal koordinasyon problemi için yerçekimsel arama algoritmasına [33,34] ve test fonksiyonları için diferansiyel gelişim algoritmasına [35] başarıyla uygulanmış ve algoritma performanslarını geliştirmiştir. Bu çalışmada diğer uygulamalardan farklı olarak ISL metodu algoritmaya beş farklı şekilde uygulanmış ve büyük ölçekli EPD problemleri için en uygun artırım metodunun da belirlenmesi amaçlanmıştır.

Çalışmada DE algoritması ISL metodu ile birlikte kullanılarak geliştirilmiştir. ISL tabanlı önerilen bu

algoritma yapılarına (Incremental Social Learning-based Differential Evolution - ISLDE) adı verilmiştir. ISL yapıları beş farklı yaklaşımla algoritmaya uygulanmıştır. DE ve ISLDE algoritmaları ile EPD probleminin çözümü konveks ve konveks olmayan yakıt maliyet fonksiyonları için ilk kez bu çalışmada yapılmıştır. Örnek sistem olarak modifiye edilmiş IEEE 118 baralı 54 generatörlü güç sistemi seçilmiştir. Sistemdeki iletim hattı kayıpları AC yük akışı analizi metoduyla hesaplanmıştır. Yapılan çalışmanın öne çıkan özellikleri ve özgün değeri özetlenecek olursa; çalışmada büyük ölçekli EPD probleminin çözümü AC yük akışı metoduyla birlikte hem konveks hem de konveks olmayan yakıt maliyet fonksiyonları için yapılmış, DE algoritması ISL metoduyla beş farklı yaklaşım kullanılarak geliştirilmiştir. Her bir yaklaşım için problem 30'ar kez çözülmüş ve elde edilen sonuçlar klasik DE algoritması ile elde edilen sonuçlarla karşılaştırılarak, ISL yapılarının algoritma üzerindeki avantaj ve dezavantajları yorumlanmıştır.

2 Artırımlı sosyal öğrenme tabanlı diferansiyel gelişim algoritması (ISLDE)

2.1 Diferansiyel gelişim algoritması

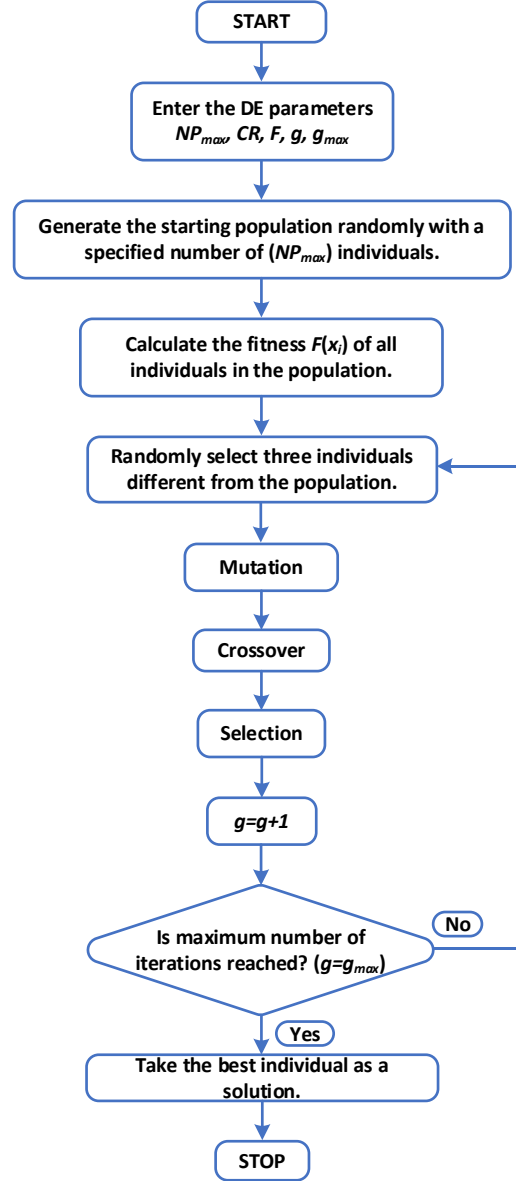
Literatürde popülasyon temelli bir optimizasyon tekniği olarak bilinen DE algoritması, Price ve Storn tarafından 1995 yılında tasarlanmıştır. DE'de probleme ait uygun bir çözümü içinde barındıran değişkenler kümesine kromozom adı verilir. Algoritmanın çalışma mantığı, her biri içinde bir çözüm barındıran, kromozom olarak adlandırılan vektörlerin, çeşitli operatörlere tabi tutularak yeni bir çözüm vektörü oluşturmalarıdır. Bu operatörler sırasıyla seçim, mutasyon ve çaprazlama operatörleridir. Oluşturulan yeni bireyin çözüme yakınlığı (uygunluğu) eskisinden daha iyi ise yeni birey, değilse eski birey bir sonraki popülasyona aktarılır, bu işlemler belirlenen adım sayısı tamamlanıncaya kadar tekrar ettirilir [4, 35, 36].

DE'de optimizasyon işlemi başlamadan belirlenmesi gereken algoritma parametreleri, popülasyonun sayısı (NP), değişken (bilinmeyen) sayısı (D), maksimum iterasyon sayısı (g_{max}), çaprazlama oranı (CR) ve ölçekleme faktörü (F)'dir. Algoritmanın işlem basamakları değişkenlerin atanması ve başlangıç popülasyonunun oluşturulması, rastgele oluşturulan bireylerin uygunluklarının hesaplanması, seçim, mutasyon, çaprazlama ve algoritmanın durdurulması şeklindedir. Algoritmaya ait akış şeması Şekil 1'de verilmiştir [35].

Her adımda yeni bir kromozom oluşturmak için popülasyondan üç adet kromozom seçilmesi gerekir. Bu nedenle uygun aramalar yapabilmek için popülasyon büyüklüğünün mutlaka üçten büyük olması gerekir ($NP > 3$). NP adet D boyutlu kromozomdan (birey) meydana gelen başlangıç popülasyonunun üretimi aşağıdaki denklemden yapılır [36].

$$x_{j,i,g=0} = x_j^{(l)} + rand_j[0,1] \cdot (x_j^{(u)} - x_j^{(l)}) \quad (1)$$

Denklemden yer alan $x_{j,i,g}$, g popülasyonundaki, i kromozomunun j parametresini, $(x_j^{(l)}, x_j^{(u)})$ ise değişkenlere ait alt ve üst sınırları göstermektedir [35].



Şekil 1. DE'nin akış diyagramı

Popülasyon oluşturulduktan sonra mutasyon işlemine geçilir. Mutasyon, kromozomların belirli adımlarda birbirlerini tekrar etmemesi için seçilen herhangi bir kromozomun genleri üzerinde rastgele değişiklik yapma işlemidir. DE'nin temel yapısında mutasyon işlemi için mevcut kromozom dışında ve birbirlerinden farklı üç ana kromozom seçilir (r_1, r_2, r_3). Seçilen kromozomlardan ilk ikisinin farkı alınarak ölçekleme faktörü (F) ile çarpılıp ağırlıklandırılmış fark kromozomu oluşturulur. F , $[0-2]$ arasında değişen bir değerdir. Daha sonra ağırlıklandırılmış fark kromozomu ile üçüncü kromozom toplanarak ara kromozom oluşturulur. Literatürde birçok çalışmada farklı şekillerde mutasyon operatörü uygulamaları da bulunmaktadır. Bu çalışmada klasik DE için uygulanan

mutasyon işleminin matematiksel ifadesi **Denklem (2)**'de gösterilmiştir.

$$n_{j,i,g+1} = x_{j,r_3,g} + F.(x_{j,r_1,g} - x_{j,r_2,g}) \quad (2)$$

Denklemden $n_{j,i,g+1}$, mutasyon ve çaprazlaştırmaya tabi tutulmuş ara kromozomu, $F.(x_{j,r_1,g}-x_{j,r_2,g})$ ağırlıklandırılmış fark kromozomunu ve $x_{j,r_1,g}$, $x_{j,r_2,g}$, $x_{j,r_3,g}$ birbirinden farklı yeni kromozom üretilmesinde kullanılacak rastgele seçilmiş kromozomları göstermektedir.

Mutasyon sonucunda elde edilen fark kromozomu ve $x_{i,g}$ kromozomu kullanılarak yeni kromozom ($u_{i,g+1}$) üretilir. Deneme kromozomu için genler CR olasılıkla fark kromozomundan, $(1-CR)$ olasılıkla mevcut kromozomdan seçilir. $j=j_{rand}$ koşulu, en az bir tane genin üretilen yeni kromozomdan alınmasını garanti etmek için kullanılmaktadır. Rastgele seçilen $j=j_{rand}$ noktasındaki bir gen, CR değerine bakılmaksızın $n_{j,i,g+1}$ 'den seçilir [35, 36].

$$x_{j,u,g+1} = \begin{cases} x_{j,n,g+1} & \text{if } rand[0,1] \leq CR \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (3)$$

Mutasyon ve çaprazlama operatörleri kullanılarak hedef kromozomla birlikte üç farklı kromozom kullanılarak yeni bir kromozom (deneme kromozomu) elde edilmiştir. Yeni popülasyona ($g=g+1$) aktarılabilecek olan kromozom uygunluk değerine bakılarak belirlenir. Hedef kromozomun uygunluk değeri zaten bilinmektedir. Problemin amaç fonksiyonu değeri uygunluk fonksiyonu olarak hesaplanır. Kromozomlardan uygunluğu yüksek olan (çözüm değerine daha yakın olan) kromozom yeni popülasyona aktarılır. Döngü iterasyon sayısı belirlenen maksimum değerine ulaşıncaya ($g=g_{max}$) kadar devam eder, g_{max} olduğunda mevcut popülasyondaki en iyi birey çözüm olarak yazdırılır ve arama sonlandırılır [35].

$$x_{i,g+1} = \begin{cases} x_{u,g+1} & \text{if } f(x_{u,g+1}) \leq f(x_{i,g+1}) \\ x_{i,g} & \text{otherwise} \end{cases} \quad (4)$$

Algoritmadaki amaç her adımda daha iyi uygunluk değerine sahip kromozomlar (çözümler) elde ederek optimum değeri yakalamak ya da bu değere yaklaşımdır. Bu döngü $g=g_{max}$ olana kadar devam ettirilmektedir. Bu çalışmada algoritmanın durdurma kriteri olarak maksimum iterasyon sayısı seçilmiştir.

2.2 Artırımlı sosyal öğrenme

Sosyal öğrenme (Social Learning-SL) kavramı, toplumu oluşturan bireyler arasındaki bilgi aktarımının genetik özelliklerden bağımsız olarak yapılabilmesi olarak tanımlanmaktadır. Bu kavramın kullanımı çok bireyli sistemlerin yeniden tasarlanması ve hızlandırılması için oldukça uygundur. Bunun nedeni SL yapısının, popülasyona yeni giren bireylerin, popülasyonla ilgili bilgi edinmeleri için zaman harcamadan, daha önce oluşturulmuş diğer deneyimli bireylerden yararlanılmasına olanak tanımasıdır. Fakat yapı üzerinde yapılan teorik modeller ve çalışmalar yalnızca

sosyal olarak edinilen bilgiye dayanmanın her zaman avantajlı olmadığı sonucunu göstermişlerdir. Bireylerin toplumsal öğrenmeyle birlikte yenilik yapmak için ilave zamana gerek duydukları belirlenmiştir. ISL olarak adlandırılan bu yeni yaklaşım da, oluşturulan birey hem yenilikçi hem de bireysel olarak öğrenen artan bir popülasyonu oluşturur. ISL yapısı çoklu olarak öğrenen birey sistemlerinde, sisteme belirli aralıklarla yeni bireylerin eklenmesi temeline dayanmaktadır. Bu yapının eklendiği algoritmalar, çözüm uzayında az sayıda bireyle arama yapmaya başlarlar ve hızlı bir öğrenme süreci gerçekleştirirler. Sonraki adımlarda popülasyona mevcut bireylerle çeşitli şekillerde iletişime geçirilen (sosyal öğrenme gerçekleştiren) yeni bireyler eklenerek aramaya devam edilir. Artan nüfus stratejisi, doğada yeni doğan bireylerin, onları çevreleyen yetişkin bireylerden çok fazla beceri öğrenmeleri gözlemine dayanmaktadır. ISL yaklaşımının sözde kodu **Şekil 2**'de verilmiştir [28-35].

```
Incremental Social Learning (ISL) Algorithm
% Initialization
t=0;
%Initialize environment Et
%Initialize primogenial population of agents Xt
% Main loop
while Stopping criteria not met do
    if Agent addition Schedule or criterion is not met then
        Xt+1 ← learn (Xt, Et) %Individual learning
    else
        Create new agent anew
        slearn (anew, Xt) %Social learning
        Xt+1 ← Xt ∪ {anew}
    end if
    Et+1 ← update (Et) %Update environment
    t ← t+1
end while
```

Şekil 2. Artırımlı sosyal öğrenme yapısının sözde kodu

2.3 Artırımlı sosyal öğrenme tabanlı diferansiyel gelişim algoritma yapıları

Literatürde verilen popülasyon tabanlı meta sezgisel algoritmaların birçoğu, ilk adımda çözüm uzayında rastgele oluşturulan belirli sayıda birey kümesi ile arama yapmaya başlarlar. Bu nedenle ilk oluşturulan popülasyondaki bireylerin arama uzayındaki rastgele yerleri en iyi sonucu elde edebilmek için çok önemlidir. Rastgele oluşturulan ilk popülasyondaki bireylerin çoğunluğunun yerel minimumlar yakınında konumlanması durumunda optimum çözüme (genel minimum) yakınsama zorlaşabilir. Bu önemli sorunu ortadan kaldırmak için, bu tür algoritmalara ISL yapısı entegre edilerek, sisteme zamanla yeni bireylerin eklenmesi ile yerel minimum noktalara yakalanma riski azaltılabilmektedir. Popülasyon ilk olarak az sayıda bireyden oluşturulur bu durumda hem yerel minimumlardan kaçınılır hem de arama uzayı hakkında hızlı bir öğrenme gerçekleşir. Daha sonra popülasyona mevcut bireylerle farklı şekillerde iletişime geçirilerek oluşturulan yeni bireyler eklenir. Popülasyona ilave edilen bir birey, bir süre popülasyonda var olan diğer bireylerden sosyal öğrenmeyi gerçekleştirir. ISL'nin bu ögesi oldukça caziptir, çünkü sosyal öğrenme yoluyla yeni bireyler, bilgiyi ayrı ayrı edinmek için zaman harcamadan, daha tecrübeli bireylerden bilgiyi elde ederler. Böylece ISL yeni bireylere, yeni şeyler öğretmek için harcanacak zamandan tasarruf edilmesini sağlar.

Popülasyona yeni bir bireyin ilave edilmesinden sonra, popülasyonun yeni koşulları tekrar gözden geçirmesi gerekir, fakat onun bir parçası olan bireyler her şeyi yeni baştan öğrenmek zorunda kalmazlar [28-35].

Bu çalışmada DE algoritmasına, ISL yapısı eklenerek ISLDE algoritmaları geliştirilmiştir. ISLDE, daha önceki bölümde ayrıntılı olarak açıklandığı gibi popülasyondaki birey sayısının belirli aralıklarla artırılması prensibine dayanmaktadır. Böylece hem hızlı bir öğrenme hem de yerel minimumlardan kaçınmanın sağlanması amaçlanmıştır. Algoritmada belirli iterasyon aralıklarında (adım sayısı (step size), SS) arama uzayında yeni oluşturulan bir birey popülasyona eklenir. Bu durum belirlenen maksimum birey sayısına (NP_{max}) ulaşana kadar devam ettirilir. Maksimum sayıya ulaştıktan sonra kalan iterasyonlar tamamlanarak arama tamamlanır. Daha önce yapılan ISL çalışmalarında ya tek şekilde ya da en çok üç farklı şekilde birey ekleme yaklaşımı denenmiştir [30,32]. Bu çalışmada popülasyona birey ekleme beş farklı yaklaşım kullanılmıştır. Bunlar alt başlıklar halinde ayrıntılı şekilde aşağıda açıklanmıştır. Birey eklemede bu farklı yaklaşımların kullanılmasındaki amaç, değişik yapıdaki problemlere en uygun ISL yapısının belirlenmesidir. İlerleyen çalışmalarda, bu çalışmada kullanılanlardan başka yapılarında tasarlanabileceği görülmektedir.

2.3.1 Yapı-1: ISLDE-1 (xRand)

Birinci yapıda, çözüm uzayında aramaya belirlenen minimum birey sayısı ile başlanır. Belli aralıklarla arama uzayında rastgele oluşturulan yeni bir birey Denklem (5)'teki gibi popülasyona eklenerek birey sayısı artırılır. Bu işlem belirlenen maksimum birey sayısı tamamlanıncaya kadar devam eder.

$$x'_{j,New}(t) = x_{j,Rand}(t) \quad (5)$$

Denklemden yer alan $x'_{j,New}(t)$ j. iterasyonda popülasyona yeni eklenen bireyi, $x_{j,Rand}(t)$ j. iterasyonda arama uzayında rastgele oluşturulan bir bireyi göstermektedir [35].

2.3.2 Yapı-2: ISLDE-2 (xPop)

İkinci yapıda, popülasyona katılacak yeni birey, o anda popülasyonda yer alan bireyler arasından Denklem (6)'daki gibi rastgele seçilir.

$$x'_{j,New}(t) = x_{j,Pop}(t) \quad (6)$$

Denklemden $x_{j,Pop}(t)$ j. iterasyonda o anda popülasyonda bulunan ve rastgele seçilen bir bireyi ifade etmektedir [35].

2.3.3 Yapı-3: ISLDE-3 (xBest)

Üçüncü yapıda, o anda popülasyonda yer alan uygunluğu en yüksek olan bireyin aynısı Denklem (7)'deki gibi popülasyona eklenerek birey sayısı artırılır.

$$x'_{j,New}(t) = x_{j,Best}(t) \quad (7)$$

Denklemden $x_{j,Best}(t)$ j. iterasyonda popülasyondaki en iyi bireyi ifade etmektedir [35].

2.3.4 Yapı-4: ISLDE-4 (xPop+xBest)

Dördüncü yapıda, popülasyona eklenecek yeni birey, Denklem (8)'deki gibi, o anda popülasyondan rastgele seçilen bir birey, o anda en iyi uygunluğa sahip bireyle iletişime geçirilerek belirlenir. Bu yapı 2. ve 3. yapının birlikte kullanılması şeklinde tasarlanmıştır.

$$x'_{j,New}(t) = x_{j,Pop}(t) + rand[0,1][x_{j,Best}(t) - x_{j,Pop}(t)] \quad (8)$$

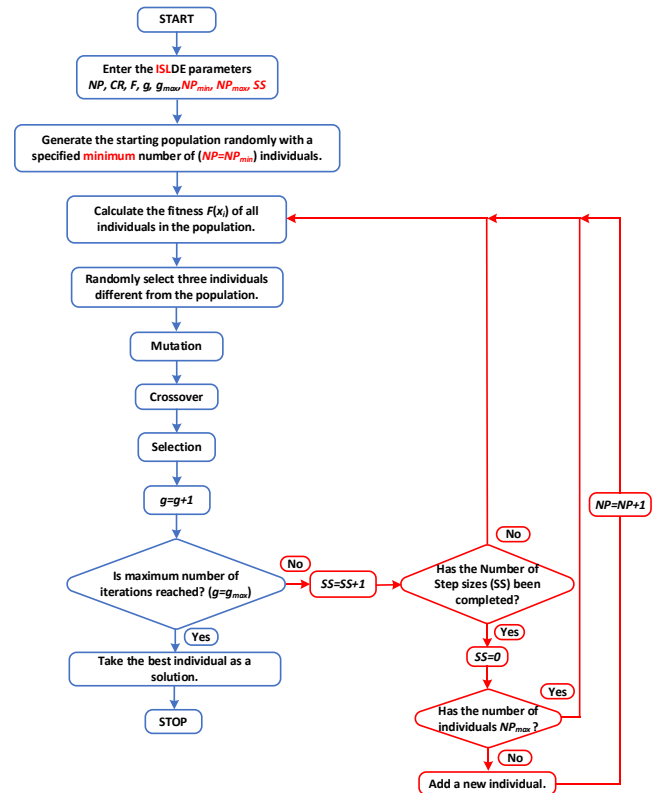
Denklemden $rand[0,1]$ 0-1 aralığında rastgele atanan bir sayıyı ifade etmektedir [35].

2.3.5 Yapı-5: ISLDE-5 (xRand+xBest)

Beşinci yapıda, popülasyondaki uygunluğu en yüksek bireyle iletişime geçecek birey, arama uzayı içinde rastgele oluşturulur ve Denklem (9)'a göre popülasyona eklenir. Bu yapı 1. ve 3. yapının birlikte kullanılması şeklinde tasarlanmıştır.

$$x'_{j,New}(t) = x_{j,Rand}(t) + rand[0,1][x_{j,Best}(t) - x_{j,Rand}(t)] \quad (9)$$

Burada verilen yapıların kullanılması ile tasarlanan ISLDE algoritması için akış diyagramı Şekil 3'te verilmiştir. ISLDE algoritmasının klasik DE algoritmasından farkları kırmızı renkte gösterilmiştir [35].



Şekil 3. ISLDE akış diyagramı

3 AC yük akış analizi ve ekonomik güç dağıtım problemi

3.1 AC yük akış analizi

Bir güç sisteminde en verimli çalışma şeklinin belirlenmesi için, bu güç sistemindeki her bir baranın gerilim genliğinin ve faz açısının hesaplanması yük akışı olarak tanımlanır. Her bir baraya ait bu bilgilerle güç sisteminde yer alan her bir iletim hattının taşıdığı aktif ve reaktif güçler ve iletim hattı kayıpları hesaplanabilir. Yük akışındaki bir diğer amaç ise salınım barası olarak belirlenen bara dışındaki tüm baraların aktif ve reaktif güçlerinin de belirlenebilmesidir. Mevcut sistem içinde eğer gerilim kontrollü bara mevcutsa, bu bara için de sabit tutulacak bara gerilim genliği belirlenir. Bu barada hesaplanması gereken reaktif güç üretimi ve bara gerilim açısı da çözümlerden elde edilir [37-40].

Sistemde yer alan bir k barasının geriliminin kutupsal koordinatlardaki ifadesi Denklem (10)'da verilmiştir [37-40].

$$V_k = |V_k| \angle \delta_k = |V_k| (\cos \delta_k + j \sin \delta_k) \quad (10)$$

Sistemin Y_{kn} bara matrisinin kutupsal koordinatlardaki ifadesi aşağıdaki gibi yazılır.

$$Y_{kn} = |Y_{kn}| \angle \theta_{kn} = |Y_{kn}| \cos \theta_{kn} + j |Y_{kn}| \sin \theta_{kn} = G_{kn} + jB_{kn} \quad (11)$$

Y_{bara} matrisi kullanılarak sistemdeki her bir k barası için enjekte edilen net akımlar Denklem (12)'deki gibi yazılır.

$$I_k = Y_{k1}V_1 + Y_{k2}V_2 + \dots + Y_{kN}V_N = \sum_{n=1}^N Y_{kn}V_n \quad (12)$$

P_k ve Q_k , k barasından sisteme giren net aktif ve reaktif gücü gösterirken, k barasından sisteme enjekte edilen net aktif ve net reaktif güçler Denklem (13) ve (14)'ten hesaplanır [37].

$$P_k = |V_k| \left| \sum_{n=1}^N |V_n| (G_{kn} \cos \delta_{kn} + B_{kn} \sin \delta_{kn}) \right| \quad (13)$$

$$Q_k = |V_k| \left| \sum_{n=1}^N |V_n| (G_{kn} \sin \delta_{kn} - B_{kn} \cos \delta_{kn}) \right| \quad (14)$$

Denklem (13) ve (14)'teki P_k ve Q_k durum değişkenleri, $|V_k|$ ve δ_k 'nin lineer olmayan fonksiyonlarıdır. Bu nedenle, güç akışı hesaplamaları lineer olmayan denklem sistemleri için iteratif çözüm yöntemlerinin kullanılmasını gerektirir. Dolayısıyla çalışmada Newton-Raphson metodunun kullanılması tercih edilmiştir. Newton-Raphson metodu güç akışı problemlerinin kutupsal formunu, her baradaki aktif ve reaktif güç farkları belli bir tolerans içinde kalana kadar çözümlemeye kullanılır. Yöntemde devre teoremlerinden biri olan düğüm gerilimleri yöntemi esas alınır.

AC yük akışı analizinde her k barası için potansiyel olarak bilinmeyen değerler aktif güç P_k , reaktif güç Q_k , gerilim açısı δ_k ve gerilim genliği $|V_k|$ 'dir. Her k barasının türüne göre P_k , Q_k , δ_k ve $|V_k|$ değerlerinden ikisi tanımlanır ve

geri kalan diğer ikisi hesaplanır. Güç akışı problemlerinin çözümünde amaç, P_k ve Q_k değerleri ile hesaplanmış değerler arasındaki fark sıfıra yaklaştırılırken bilinmeyen iki değer bulunmasıdır [37-40].

Bara gerilim büyüklükleri $|V_k|$ ve açıları δ_k güç akışı çalışmalarında giriş bilinmeyenleridir, sahip oldukları değerlere göre sistemin durumlarını belirledikleri için durum ya da matematiksel açıdan bakılırsa bağımsız değişkenlerdir. Bu durumda güç akışı problemi, giriş verilerini esas alarak eşit miktarda güç akışı denklemi kullanarak tüm durum değişkenlerinin değerlerinin bulunması olarak da tanımlanabilir. Eğer N_g tane gerilim kontrollü bara (salınım barası dışında) varsa $(2N - N_g - 2)$ adet durum değişkeni için çözülmesi gereken $(2N - N_g - 2)$ adet denklem bulunmaktadır. Durum değişkenlerinin değerleri hesaplandıktan sonra sistemin tüm durumu bilinmektedir ve durum değişkenlerine bağlı tüm değerler hesaplanabilir.

Sistemde yer alan k . barayı n . baraya bağlayan iletim hattındaki aktif güç (P_{kn}), reaktif güç (Q_{kn}), baraların gerilim genlikleri ($|V_k|$, $|V_n|$) ve bara açılarının (δ_k , δ_n), Newton-Raphson yöntemi sonucunda elde edilen değerleri Denklem (15) ve (16)'da yer alan formüllerle hesaplanır [37-40].

$$P_{kn} = G_{kn} |V_k|^2 - |V_k| |V_n| (G_{kn} \cos \delta_{kn} + B_{kn} \sin \delta_{kn}) \quad (15)$$

$$Q_{kn} = |V_k|^2 \left(\frac{B_{kn}}{2} - B_{kn} \right) + |V_k| |V_n| (B_{kn} \cos \delta_{kn} - G_{kn} \sin \delta_{kn}) \quad (16)$$

Burada P_{kn} ve Q_{kn} , n . baraya k . baradan gönderilen sırasıyla aktif ve reaktif güçlerin k . bara sınırındaki değerleridir. Bir güç sistemi için bu değerler hesaplandıktan sonra ekonomik güç dağıtımı yapılabilir.

3.2 Konveks ve konveks olmayan ekonomik güç dağıtım problemi

EPD problemi, bir güç sisteminin en ekonomik şekilde işletilebilmesinin temelini oluşturmaktadır. Güç sistemi işletiminin temelini ise sistemin güvenilirliğini korumanın yanında, talep edilen gücü en düşük maliyetle üretebilmek oluşturmaktadır. EPD probleminin çözümü, sistemdeki bütün üretim birimleri için belirlenen yakıt maliyeti fonksiyonlarına bağlıdır [41]. Bu yakıt maliyetleri üretim birimlerinin kullandığı fosil kaynaklı yakıtlara bağlı olarak değişmektedir. Literatürde yakıt maliyeti fonksiyonları için iki farklı yapı sıklıkla kullanılmaktadır. Bunlardan ilki sistemdeki üretim birimlerinin valf noktası etkilerinin ihmal edildiği durumlarda kullanılan ve problemin konveks yapıda olduğu yakıt maliyeti fonksiyonlarıdır. Bu fonksiyon her bir üretim birimi için aktif güç çıkışlarının 2. derece fonksiyonu olarak Denklem (17)'deki gibi alınmıştır. Bu durumda çözümü yapılan problem konveks ekonomik güç dağıtımı (Convex EPD - cEPD) olarak adlandırılmıştır.

$$F_n(P_{G,n}) = a_n + b_n P_{G,n} + c_n P_{G,n}^2, \quad (\$/h) \quad (17)$$

$F_n(P_{G,n})$ sistemdeki n . üretim birimine ait yakıt maliyeti değeridir.

Problemin çözümü yapılırken, güç sisteminde yer alan üretim birimlerinin valf noktası etkileri de dikkate alınacak

ise, fonksiyon konveks olmayan bir hale gelir ve [Denklem \(18\)](#)'deki gibi kullanılmalıdır. Bu durumda çözümü yapılan problem konveks olmayan ekonomik güç dağıtımı (Non-convex EPD - ncEPD) halini alır. Optimizasyon algoritmaları ile ncEPD probleminin optimal sonucuna ulaşmak, cEPD problemine göre daha zordur [\[2, 37\]](#).

$$F_n(P_{G,n}) = a_n + b_n P_{G,n} + c_n P_{G,n}^2 + \left| e_n \cdot \sin \left(f_n (P_{G,n}^{\min} - P_{G,n}) \right) \right|, (\$/h) \quad (18)$$

[Denklem \(17\)](#) ve [\(18\)](#)'de verilen değişkenlerden ($P_{G,n}$) her bir n . üretim biriminin aktif güç çıkış değeri olmakla birlikte birimi MW olarak alınmaktadır. Hesaplama kolaylığı için pu dönüşümler, çözüm aşamasında yapılmaktadır. a_n , b_n , c_n , e_n ve f_n ise n . üretim birimine ait sabit katsayılardır.

cEPD ve ncEPD problemlerinin çözümü, sistemin belirlenen bütün kısıtları altında belirlenen amaç fonksiyonunun minimize edilmesi şeklindedir. Problemlerin çözümünde amaç fonksiyonu olarak sistemdeki bütün üretim birimlerinin minimum toplam yakıt maliyeti değerlerinin toplamını ifade eden (Total Fuel Cost – TFC) alınmış ve [Denklem \(19\)](#)'da verilmiştir.

$$TFC = \min \sum_{n=1}^{N_G} F_n(P_{G,n}) \quad (n \in N_G), (\$/h) \quad (19)$$

N_G sistemde yer alan üretim birimi sayısını göstermektedir. Literatürde EPD problemlerinin çözümünde genellikle güç dağıtımı sırasında iletim hatlarında meydana gelen kayıplar ya ihmal edilmekte ya da B kayıp matrisleri ile yaklaşık olarak hesaplanarak probleme dahil edilmektedir. Bu durumda elde edilen sonuçlar da yaklaşık olmaktadır. Bu çalışmada iletim hattı kayıpları AC yük akışı yapılarak hesaplanmış ve çözüme dahil edilmiştir. Sistemde sağlanması gereken aktif ve reaktif güç denge kısıtları [Denklem \(20\)](#) ve [\(21\)](#)'deki gibi alınmıştır.

$$P_{load} + P_{loss} - \sum_{n=1}^{N_G} (P_{G,n}) = 0, \quad (n \in N_G) \quad (20)$$

$$Q_{load} + Q_{loss} - \sum_{n=1}^{N_G} (Q_{G,n}) = 0, \quad (n \in N_G) \quad (21)$$

[Denklemde](#) P_{load} ve Q_{load} sistemde talep edilen toplam aktif ve reaktif yükü, P_{loss} ve Q_{loss} güç dağıtımında iletim hatlarında meydana gelen toplam aktif ve reaktif güç kayıplarını göstermektedir. ($Q_{G,n}$) ise her bir n . üretim biriminin reaktif güç çıkış değeri olmakla birlikte birimi MVar olarak alınmaktadır.

Sistemde yer alan güç üretim birimlerinin üretebileceği minimum ve maksimum aktif ve reaktif güç üretimi değerleri ise [Denklem \(22\)](#) ve [\(23\)](#)'de verilmiştir.

$$P_{G,n}^{\min} \leq P_{G,n} \leq P_{G,n}^{\max}, \quad (n \in N_G) \quad (22)$$

$$Q_{G,n}^{\min} \leq P_{G,n} \leq Q_{G,n}^{\max}, \quad (n \in N_G) \quad (23)$$

4 Problem çözümleri

Beş farklı ISLDE yapısı, cEPD ve ncEPD probleminin çözümü için modifiye edilmiş IEEE 118 baralı test sistemine ayrı ayrı uygulanmıştır. [Şekil A1](#)'de tek hat diyagramı verilen modifiye edilmiş sistem 118 bara, 179 iletim hattı, 54 termik üretim birimi ve 91 yük barasından oluşmaktadır. Sistemin yük baralarında toplam 42.42 pu aktif yük (P_{load}) ve 14.38 pu reaktif yük (Q_{load}) bulunmaktadır. Bütün veriler pu olarak verilmiş ve baz değerleri $S_{base}=100$ MVA, $U_{base}=230$ kV ve $Z_{base}=529 \Omega$ olarak alınmıştır. Sistemde 69 nolu bara salınım barası olarak belirlenmiş ve bara gerilimi $1.035 \angle 0^\circ$ V olarak alınmıştır [\[40, 42\]](#).

Örnek test sistemi literatürde büyük boyutlu test sistemleri arasında yer almakla birlikte, birçok farklı çalışmada farklı değişken ve kısıtlar altında çözülmüştür. Bu çalışmalardan çoğu ya kayıplar ihmal edilerek ya da B kayıp matrisleri kullanılarak çözülmüştür. Bu çalışmadaki farklı iki durum için sistemin iletim hattı kayıpları dikkate alınarak yapılan çözümlerde P_{loss} bölüm 3.1'de ayrıntıları verilen yük akışı analizi sonucunda hesaplanmıştır. AC yük akışı metoduyla yapılacak çözümler için sistem modifiye edilmiş ve sistemin çözümü için gerekli bütün veriler çalışmanın ek kısmında verilmiştir. Bunlardan sistemin nominal π eşdeğer devresine ait R , X , B değerleri [Tablo A1](#)'de, aktif ve reaktif yük değerleri [Tablo A2](#)'de, AC yük akışı başlangıç değerleri [Tablo A3](#)'de, üretim birimlerine ait işletim limitleri [Tablo A4](#)'de ve konveks ve konveks olmayan yakıt maliyeti fonksiyonlarına ait katsayılar [Tablo A5](#)'de verilmiştir. cEPD ve ncEPD problemlerinin klasik DE, geliştirilen ISLDE-1, ISLDE-2, ISLDE-3, ISLDE-4 ve ISLDE-5 algoritmalarıyla çözümü, MATLAB R2022a'da geliştirilen programla 30 kez yapılmıştır. Programlar 2xIntel Xeon E5-2637 v4 3.50 GHZ x64 işlemci ve 512 GB ram bellekli iş istasyonunda koşturulmuştur.

Çözümler yapılırken kullanılan algoritmalara ve problemlere ait parametre değerleri [Tablo 1](#)'de verilmiştir.

[Tablo 1](#)'de yer alan CR çaprazlama oranı, F ölçekleme faktörü, $PopSize$ sabit popülasyondaki birey sayısı, $PopMin$ ve $PopMax$ artırımlı öğrenme kavramı eklenen algoritmalar için aramaya başlanan popülasyondaki minimum ve maksimum birey sayıları, SS popülasyona yeni birey eklenecek adım sayısı, $fcall$ her algoritma için fonksiyon çağırımı sayısı, $IteN$ iterasyon sayısı, Run problemin kaç kez çalıştırıldığını gösteren algoritma parametreleridir. CPF_{slack} salınım barası ceza katsayısını ve ε ise iteratif işlemler için kabul edilebilir hata oranını göstermektedir.

Tablo 1. Algoritma ve problemlere ait parametre değerleri

DE	CR	F	PopSize	fcall
	0.9	0.5	50	
	PopMin	PopMax	SS	25000 (IteN: 500)
ISLDE	30	59	2	
cEPD	CPF _{slack}	ε	Run	
ncEPD	1e5	1e-5	30	
$P_{yük} (pu)$	42,42	$Q_{yük} (pu)$	14,38	
S_{baz}	100 MVA	U_{baz}	230 kV	Z_{baz} 529 Ω

4.1 Durum-1: Konveks ekonomik güç dağıtım problemi (cEPD)

İlk olarak, örnek güç sisteminin, [Denklem \(17\)](#)'de verilen konveks yakıt maliyeti fonksiyonu ile çözümü yapılmıştır.

Örnek problem 6 farklı algoritma ile 30'ar kez çözülmüştür. Bu çalışmalardan eniyi yakıt maliyetinin elde edildiği çözüme ait aktif güç üretim değerleri [Tablo 2](#)'de, reaktif güç üretim değerleri [Tablo 3](#)'de verilmiştir.

Tablo 2. Eniyi yakıt maliyetlerinin elde edildiği aktif güç değerleri (cEPD)

Gen. Units (54)		DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
		Run:3	Run:21	Run:8	Run:19	Run:18	Run:21
1	$P_{G,1}$	0.003030	0.008282	0.005805	0.004438	0.002147	0.025486
4	$P_{G,4}$	0.581459	0.622219	0.603355	0.607409	0.609370	0.612427
6	$P_{G,6}$	0.390629	0.351176	0.345975	0.362774	0.363878	0.372219
8	$P_{G,8}$	0.501430	0.519644	0.540376	0.542814	0.514790	0.540330
10	$P_{G,10}$	1.000000	0.999975	0.999993	0.999985	0.999907	0.999977
12	$P_{G,12}$	0.857229	0.857174	0.861018	0.865030	0.865754	0.862109
15	$P_{G,15}$	0.748305	0.723548	0.724239	0.735142	0.741274	0.712131
18	$P_{G,18}$	0.551277	0.561287	0.546651	0.534305	0.545235	0.532864
19	$P_{G,19}$	0.417273	0.416510	0.439446	0.419550	0.402181	0.437486
24	$P_{G,24}$	0.472998	0.462366	0.476447	0.445641	0.477755	0.449151
25	$P_{G,25}$	1.000000	0.999934	1.000000	0.999997	0.999813	0.999806
26	$P_{G,26}$	2.972836	2.981312	2.977438	2.965153	2.984790	2.966378
27	$P_{G,27}$	0.000109	0.000143	0.000289	0.000162	0.000243	0.000733
31	$P_{G,31}$	0.072779	0.076555	0.071105	0.075233	0.071461	0.074839
32	$P_{G,32}$	0.472468	0.484184	0.479749	0.474550	0.444985	0.478037
34	$P_{G,34}$	0.344265	0.371901	0.335040	0.352503	0.380900	0.380685
36	$P_{G,36}$	0.428633	0.395242	0.424485	0.400064	0.398648	0.384742
40	$P_{G,40}$	0.408765	0.425493	0.395039	0.435988	0.434025	0.423415
42	$P_{G,42}$	0.850157	0.858011	0.852499	0.861182	0.839418	0.872137
46	$P_{G,46}$	0.210651	0.210323	0.209282	0.211846	0.213369	0.212290
49	$P_{G,49}$	1.189907	1.189960	1.189944	1.189958	1.190000	1.189897
54	$P_{G,54}$	0.486494	0.489234	0.491058	0.484825	0.492310	0.488354
55	$P_{G,4}$	0.562414	0.554305	0.583081	0.575392	0.562946	0.564219
56	$P_{G,4}$	0.535583	0.539536	0.527837	0.530950	0.543672	0.517357
59	$P_{G,59}$	0.999825	0.999848	0.999911	1.000000	0.999941	0.999752
61	$P_{G,61}$	1.632419	1.646580	1.634710	1.651765	1.626452	1.645002
62	$P_{G,62}$	0.000077	0.005356	0.017421	0.001672	0.027311	0.034983
65	$P_{G,65}$	0.999983	0.999875	0.999940	0.999912	0.999879	0.999870
66	$P_{G,66}$	3.824834	3.814763	3.806596	3.821532	3.794852	3.798397
69	$P_{G,69,slack}$	4.909827	4.911128	4.913114	4.904699	4.912257	4.886089
	$Q_{G,69,slack}$	1.980540	1.981212	1.981939	1.981768	1.981876	1.982492
70	$P_{G,70}$	0.218192	0.187581	0.188470	0.185856	0.203004	0.200736
72	$P_{G,72}$	0.000648	0.009652	0.019424	0.004077	0.008521	0.028720
73	$P_{G,73}$	0.116756	0.131840	0.127842	0.155259	0.131349	0.124730
74	$P_{G,74}$	0.442848	0.470533	0.450851	0.446960	0.443004	0.475703
76	$P_{G,76}$	0.483133	0.436416	0.480129	0.493066	0.470313	0.457764
77	$P_{G,4}$	0.080597	0.100616	0.067606	0.056958	0.087241	0.086732
80	$P_{G,80}$	4.603383	4.618659	4.621976	4.594417	4.620694	4.618903
85	$P_{G,85}$	0.000232	0.001037	0.000139	0.000519	0.000013	0.000821
87	$P_{G,87}$	0.037648	0.036391	0.037757	0.037649	0.037253	0.038102
89	$P_{G,89}$	5.102846	5.090106	5.096419	5.101871	5.111316	5.080062
90	$P_{G,90}$	0.000150	0.000125	0.000344	0.000000	0.001304	0.000052
91	$P_{G,91}$	0.000362	0.000015	0.000890	0.000576	0.000368	0.000752
92	$P_{G,92}$	0.000457	0.000193	0.000139	0.000000	0.000982	0.000195
99	$P_{G,99}$	0.001054	0.000000	0.001056	0.006041	0.000685	0.001900
100	$P_{G,100}$	2.396569	2.377707	2.380920	2.378519	2.375649	2.381040
103	$P_{G,103}$	0.388724	0.387338	0.390146	0.390592	0.391860	0.392927
104	$P_{G,104}$	0.045111	0.066262	0.084715	0.068761	0.030990	0.048936
105	$P_{G,105}$	0.133498	0.137232	0.100926	0.141670	0.153197	0.145238
107	$P_{G,107}$	0.357374	0.349866	0.348173	0.335778	0.339044	0.350364
110	$P_{G,110}$	0.121018	0.108595	0.125774	0.134593	0.100725	0.115357
111	$P_{G,111}$	0.356306	0.354574	0.351418	0.353746	0.353218	0.350840
112	$P_{G,112}$	0.445679	0.442686	0.432902	0.412021	0.468177	0.432534
113	$P_{G,113}$	0.288035	0.259964	0.285679	0.287117	0.275761	0.249758
116	$P_{G,116}$	0.000112	0.002065	0.000311	0.004477	0.002984	0.000232
Toplam P_{gen} (pu)		43.046389	43.045320	43.045850	43.044995	43.047216	43.043557
P_{loss} (pu)		0.626389	0.625321	0.625850	0.624995	0.627216	0.623557
TFC (\$/h)		135193.313325	135193.671100	135193.225528	135193.697742	135194.224591	135194.795624
CPU Time (s)		1486.87	1481.79	1554.13	1620.17	1590.11	1419.18

Tablo 3. Eniye yakıt maliyetlerinin elde edildiği reaktif güç değerleri (cEPD)

Gen. Units (54)		DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
		Run:3	Run:21	Run:8	Run:19	Run:18	Run:21
1	$Q_{G,1}$	1.514461	1.512738	1.513071	1.513246	1.514635	1.512119
4	$Q_{G,4}$	-0.195786	-0.207230	-0.201957	-0.203040	-0.203623	-0.204516
6	$Q_{G,6}$	0.030837	0.039191	0.040206	0.037797	0.036653	0.035081
8	$Q_{G,8}$	0.069556	0.065188	0.060365	0.060028	0.066372	0.060608
10	$Q_{G,10}$	-1.400456	-1.399515	-1.399901	-1.401045	-1.399423	-1.400136
12	$Q_{G,12}$	-0.782597	-0.782595	-0.782716	-0.782840	-0.782863	-0.782750
15	$Q_{G,15}$	0.962938	0.970112	0.969316	0.965941	0.964733	0.972923
18	$Q_{G,18}$	0.016632	0.013669	0.018519	0.020981	0.017823	0.021594
19	$Q_{G,19}$	0.264615	0.264977	0.259490	0.263983	0.267894	0.259500
24	$Q_{G,24}$	-0.284179	-0.281122	-0.285791	-0.276735	-0.285048	-0.278037
25	$Q_{G,25}$	-0.340234	-0.338296	-0.339041	-0.339643	-0.339216	-0.339470
26	$Q_{G,26}$	1.741104	1.742009	1.741207	1.741645	1.741492	1.743680
27	$Q_{G,27}$	-1.435481	-1.434525	-1.435370	-1.435533	-1.434692	-1.434612
31	$Q_{G,31}$	-0.041650	-0.043065	-0.041179	-0.042541	-0.040780	-0.042603
32	$Q_{G,32}$	0.234729	0.231404	0.232604	0.234022	0.243017	0.233127
34	$Q_{G,34}$	-0.291062	-0.299385	-0.288253	-0.293522	-0.301231	-0.301283
36	$Q_{G,36}$	0.154513	0.163827	0.154840	0.162955	0.162821	0.166587
40	$Q_{G,40}$	0.008558	0.003367	0.012412	0.000407	0.000920	0.003835
42	$Q_{G,42}$	0.024517	0.021770	0.023594	0.020763	0.027740	0.016871
46	$Q_{G,46}$	0.285586	0.285658	0.286061	0.285111	0.284821	0.284940
49	$Q_{G,49}$	-0.155953	-0.155891	-0.155854	-0.155974	-0.155796	-0.155770
54	$Q_{G,54}$	1.515017	1.513876	1.513424	1.514127	1.513757	1.513555
55	$Q_{G,4}$	-0.009379	-0.007185	-0.015273	-0.013185	-0.009618	-0.009915
56	$Q_{G,4}$	-0.104196	-0.105440	-0.101867	-0.102859	-0.106567	-0.098957
59	$Q_{G,59}$	-0.383861	-0.383626	-0.384549	-0.384344	-0.383989	-0.383754
61	$Q_{G,61}$	1.201969	1.199033	1.201870	1.198452	1.203267	1.198621
62	$Q_{G,62}$	-0.136804	-0.137731	-0.139118	-0.137591	-0.140171	-0.141608
65	$Q_{G,65}$	-0.198396	-0.198604	-0.198760	-0.198735	-0.198834	-0.199320
66	$Q_{G,66}$	-2.832757	-2.832891	-2.833108	-2.833774	-2.832301	-2.832745
69	$P_{G,69,slack}$	4.909827	4.911128	4.913114	4.904699	4.912257	4.886089
	$Q_{G,69,slack}$	1.980540	1.981212	1.981939	1.981768	1.981876	1.982492
70	$Q_{G,70}$	-0.093181	-0.086213	-0.086160	-0.086179	-0.090046	-0.088969
72	$Q_{G,72}$	-0.101474	-0.103492	-0.106089	-0.102713	-0.103474	-0.108638
73	$Q_{G,73}$	0.072178	0.069199	0.070096	0.064248	0.069243	0.070793
74	$Q_{G,74}$	-0.126676	-0.134961	-0.129149	-0.128059	-0.126590	-0.136887
76	$Q_{G,76}$	-0.128133	-0.113559	-0.127347	-0.131536	-0.124113	-0.120625
77	$Q_{G,4}$	0.277415	0.271324	0.280597	0.283182	0.275375	0.273755
80	$Q_{G,80}$	2.258230	2.252616	2.250818	2.254689	2.253926	2.252668
85	$Q_{G,85}$	0.033076	0.032697	0.032805	0.032809	0.033077	0.032438
87	$Q_{G,87}$	0.110681	0.110931	0.110660	0.110681	0.110759	0.110591
89	$Q_{G,89}$	0.040170	0.041295	0.040576	0.039998	0.039066	0.042227
90	$Q_{G,90}$	0.587706	0.587685	0.587620	0.587770	0.587354	0.587662
91	$Q_{G,91}$	-0.129090	-0.128957	-0.129254	-0.129165	-0.129136	-0.129168
92	$Q_{G,92}$	-0.218742	-0.219542	-0.218992	-0.218616	-0.218133	-0.220052
99	$Q_{G,99}$	-0.175502	-0.175366	-0.175643	-0.176831	-0.175520	-0.175870
100	$Q_{G,100}$	0.922698	0.925806	0.925463	0.926123	0.927850	0.924772
103	$Q_{G,103}$	0.651543	0.652939	0.652483	0.652711	0.651000	0.652137
104	$Q_{G,104}$	-0.010593	-0.016394	-0.020954	-0.016904	-0.006641	-0.011593
105	$Q_{G,105}$	-0.060596	-0.062487	-0.051461	-0.063426	-0.065498	-0.064694
107	$Q_{G,107}$	-0.001453	0.000823	0.001403	0.005149	0.004145	0.000696
110	$Q_{G,110}$	0.029295	0.032958	0.026759	0.023111	0.036715	0.030164
111	$Q_{G,111}$	-0.017475	-0.017024	-0.016199	-0.016807	-0.016670	-0.016048
112	$Q_{G,112}$	0.224116	0.225337	0.229336	0.237899	0.214960	0.229486
113	$Q_{G,113}$	0.148658	0.157596	0.149749	0.148373	0.151950	0.159753
116	$Q_{G,116}$	-1.593612	-1.593169	-1.592542	-1.592532	-1.593842	-1.595205
Toplam Q_{gen} (pu)		4.112020	4.110974	4.110756	4.103838	4.119428	4.099454
Q_{loss} (pu)		-10.267980	-10.269026	-10.269244	-10.276162	-10.260572	-10.280546

Tablo 1 incelendiğinde bütün algoritmaların 30 çalışma için eniye çözüm bakımından benzer yakıt maliyeti değerlerini en az bir koşturmada yakaladığı görülmektedir. Eniye maliyet değeri 135193.225528 \$/h olarak ISLDE-3 algoritması ile elde edilirken, süre bakımından en hızlı algoritma 1433.288900 s ile ISLDE-5 algoritması olmuştur. Algoritmaların birbirlerine olan üstünlükleri ve kararlılıklarının daha belirgin incelenebilmesi için, 30 çalışmanın tamamında elde edilen istatistiki değerler Tablo 4'de verilmiştir.

Tablo 4. 30 çalışmaya ait istatistiki değerler (cEPD)

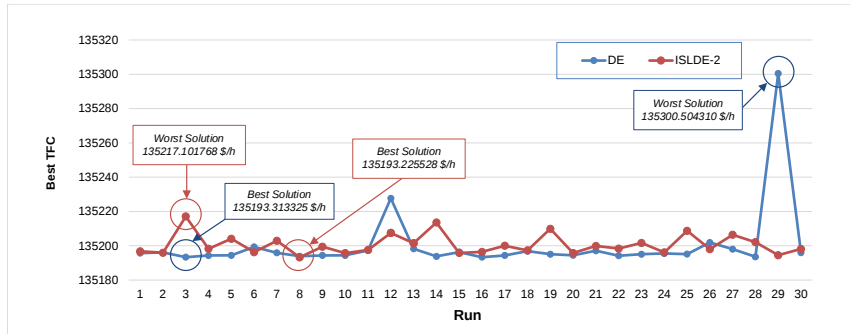
	DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
Eniyi	135193.31332 (Run: 3)	135193.671100 (Run: 21)	135193.225528 (Run: 8)	135193.697742 (Run: 19)	135194.224591 (Run: 18)	135194.795624 (Run: 21)
Ortalama	135200.186155	135207.725988	135200.609906	135200.635062	135205.258781	135204.016180
Enkötü	135300.504310 (Run: 29)	135304.164358 (Run: 26)	135217.101768 (Run: 3)	135252.224546 (Run: 16)	135262.641324 (Run: 21)	135250.448091 (Run: 24)
StdSapma	19.919008	24.208178	5.863443	11.738140	16.479825	13.726839
Toplam süre (s)	44606.1	44453.7	46623.9	48605.1	47703.3	42575.4
Ortalama süre (s)	1486.87	1481.79	1554.13	1620.17	1590.11	1419.18

Tablo 4 incelendiğinde bütün algoritmaların 30 çalışma için eniyi çözüm bakımından benzer yakıt maliyeti değerlerini yakalamasına rağmen, enkötü yakıt maliyeti değerleri farklılıklar göstermektedir. 30 çalışmada elde edilen standart sapma değerlerine bakıldığında en kararlı algoritmanın ISLDE-2 olduğu görülmüştür. En kararsız yapı ise ISLDE-1 olmuştur. Diğer bütün yapılar klasik DE'ye göre daha kararlı hale gelmesine rağmen ISLDE-1'de kararlılığın klasik algoritmaya göre iyileştirilemediği görülmüştür.

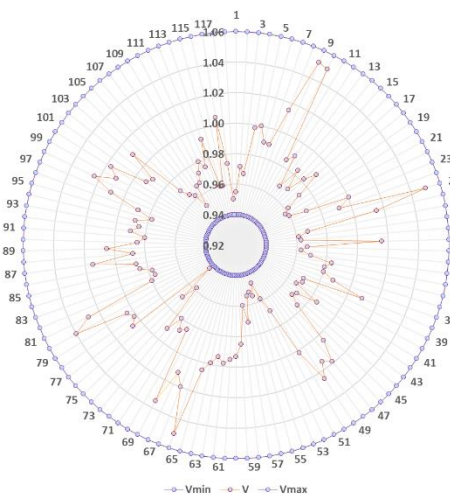
Klasik algoritma ile 30 çalışmada elde edilen eniyi yakıt maliyeti değerleri ile artırılmış sosyal öğrenme yapılarından

eniyi yakıt maliyetini elde eden ISLDE-2 algoritmasının 30 çalışmada elde ettiği değerler Şekil 4'te verilmiştir.

Bütün sonuçlar arasında eniyi yakıt maliyeti değerinin bulunduğu durum için (ISLDE-2, Run:8) sistemdeki bara gerilim değerlerinin grafiği ise Şekil 5'te verilmiştir. Şekilden de görüldüğü üzere bütün bara gerilimleri verilen alt (0.94 pu) ve üst (1.06 pu) gerilim değerleri arasında yer almaktadır. Örnek istemde yer alan 54 adet üretim birimi bulunan baraların tamamı gerilim kontrollü baralar olduğundan, bara gerilim değerleri yük akışı süresince değişmemiştir.



Şekil 4. 30 çalışmada elde edilen TFC değerleri (cEPD)



Şekil 5. ISLDE-2, Run:8 için bara gerilim değerleri (cEPD)

Tablo 5. Eniye yakıt maliyetlerinin elde edildiği aktif güç değerleri (ncEPD)

Gen. Units (54)		DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
		Run:3	Run:21	Run:8	Run:19	Run:18	Run:21
1	$P_{G,1}$	0.020233	0.013516	0.053304	0.019987	0.015033	0.008603
4	$P_{G,4}$	0.602811	0.611273	0.595530	0.594973	0.614478	0.601310
6	$P_{G,6}$	0.343537	0.327649	0.360615	0.367387	0.353431	0.368240
8	$P_{G,8}$	0.555812	0.536222	0.524947	0.540550	0.516331	0.544016
10	$P_{G,10}$	0.999975	0.999997	0.999841	0.999790	0.999953	0.999969
12	$P_{G,12}$	0.865406	0.866332	0.864918	0.859925	0.856872	0.867119
15	$P_{G,15}$	0.713991	0.742797	0.725200	0.748896	0.747963	0.695074
18	$P_{G,18}$	0.552194	0.564791	0.579745	0.536473	0.522264	0.579057
19	$P_{G,19}$	0.466842	0.448201	0.420486	0.427790	0.453354	0.449672
24	$P_{G,24}$	0.373946	0.374040	0.374209	0.374099	0.374143	0.374047
25	$P_{G,25}$	0.999866	0.999873	0.999924	0.999820	0.999534	0.999883
26	$P_{G,26}$	2.981134	2.979839	2.984140	2.997292	2.966015	2.993874
27	$P_{G,27}$	0.000092	0.000470	0.000464	0.006169	0.000186	0.001206
31	$P_{G,31}$	0.075855	0.073152	0.075071	0.074839	0.075290	0.072265
32	$P_{G,32}$	0.489654	0.486884	0.484027	0.478186	0.485066	0.490434
34	$P_{G,34}$	0.349158	0.356726	0.397371	0.345004	0.365889	0.346106
36	$P_{G,36}$	0.429752	0.438333	0.386438	0.414613	0.401957	0.415974
40	$P_{G,40}$	0.426511	0.384459	0.458007	0.458699	0.395568	0.431026
42	$P_{G,42}$	0.851579	0.861067	0.822191	0.911805	0.855469	0.851213
46	$P_{G,46}$	0.211175	0.215633	0.204962	0.213853	0.215595	0.211330
49	$P_{G,49}$	1.189946	1.189866	1.189840	1.189399	1.190000	1.189843
54	$P_{G,54}$	0.490074	0.487379	0.491729	0.485259	0.496491	0.495052
55	$P_{G,4}$	0.588829	0.607269	0.574389	0.550112	0.581012	0.569814
56	$P_{G,4}$	0.541914	0.500998	0.535061	0.537363	0.536147	0.542884
59	$P_{G,59}$	0.999890	0.999830	0.999746	0.999909	0.999819	0.999859
61	$P_{G,61}$	1.637129	1.649309	1.641796	1.642601	1.659123	1.648520
62	$P_{G,62}$	0.020210	0.027374	0.050164	0.014838	0.045341	0.013844
65	$P_{G,65}$	0.999872	0.999975	0.999171	0.999983	1.000000	0.999960
66	$P_{G,66}$	3.820232	3.824222	3.825081	3.793955	3.795937	3.817253
69	$P_{G,69,slack}$	4.909298	4.918238	4.885018	4.915800	4.919740	4.920000
	$Q_{G,69,slack}$	1.982924	1.982504	1.983084	1.982553	1.983903	1.982534
70	$P_{G,70}$	0.237508	0.229323	0.190460	0.199557	0.242158	0.205405
72	$P_{G,72}$	0.001259	0.005183	0.015419	0.008855	0.020676	0.002722
73	$P_{G,73}$	0.092138	0.103192	0.140774	0.109927	0.137868	0.144895
74	$P_{G,74}$	0.466639	0.463560	0.435085	0.486981	0.426663	0.460495
76	$P_{G,76}$	0.505403	0.486815	0.484697	0.483494	0.502348	0.480199
77	$P_{G,4}$	0.000065	0.000000	0.000223	0.000059	0.000018	0.000352
80	$P_{G,80}$	4.622009	4.629688	4.641244	4.622224	4.626210	4.641566
85	$P_{G,85}$	0.000175	0.000598	0.000000	0.000789	0.000433	0.000310
87	$P_{G,87}$	0.037936	0.038412	0.037706	0.037832	0.036840	0.038018
89	$P_{G,89}$	5.141390	5.112189	5.131071	5.135713	5.128482	5.124341
90	$P_{G,90}$	0.000404	0.000110	0.000179	0.000035	0.001564	0.000483
91	$P_{G,91}$	0.000322	0.000253	0.002223	0.000973	0.001226	0.000503
92	$P_{G,92}$	0.000119	0.000017	0.000212	0.000048	0.000081	0.000168
99	$P_{G,99}$	0.000006	0.000000	0.000364	0.000330	0.000095	0.000025
100	$P_{G,100}$	2.244007	2.243954	2.244764	2.244184	2.244373	2.244052
103	$P_{G,103}$	0.387889	0.392992	0.396861	0.391625	0.389291	0.393248
104	$P_{G,104}$	0.072895	0.097434	0.052191	0.080951	0.062409	0.072383
105	$P_{G,105}$	0.167477	0.173017	0.143703	0.137138	0.165172	0.168852
107	$P_{G,107}$	0.347557	0.354454	0.359927	0.383654	0.365555	0.355235
110	$P_{G,110}$	0.139730	0.134696	0.169385	0.130369	0.157757	0.125287
111	$P_{G,111}$	0.354380	0.354228	0.352247	0.355275	0.354487	0.350973
112	$P_{G,112}$	0.437964	0.433284	0.423591	0.438320	0.422125	0.439725
113	$P_{G,113}$	0.277797	0.303456	0.310493	0.291246	0.318184	0.297107
116	$P_{G,116}$	0.002422	0.000056	0.010937	0.004171	0.002316	0.001348
Toplam P_{gen} (pu)		43.046389	43.044406	43.042624	43.047144	43.043123	43.044329
P_{loss} (pu)		0.626389	0.624406	0.622624	0.627144	0.623123	0.624329
TFC (\$/h)		135193.313325	135411.228474	135411.362343	135415.637539	135414.267529	135413.386380
CPU Time (s)		1486.87	1478.361762	1511.226445	1574.009588	1417.238985	1669.336020

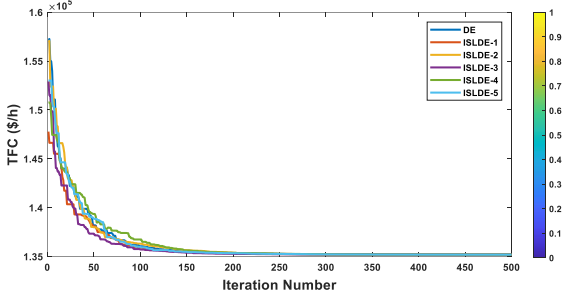
Tablo 6. Eniye yakıt maliyetlerinin elde edildiği reaktif güç değerleri (ncEPD)

Gen. Units (54)		DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
		Run:3	Run:21	Run:8	Run:19	Run:18	Run:21
1	$Q_{G,1}$	1.509595	1.509610	1.500998	1.509593	1.511095	1.510656
4	$Q_{G,4}$	-0.201871	-0.204160	-0.199834	-0.199554	-0.205044	-0.201506
6	$Q_{G,6}$	0.039930	0.043827	0.035919	0.036561	0.038767	0.034266
8	$Q_{G,8}$	0.056746	0.061175	0.063867	0.060525	0.065943	0.059571
10	$Q_{G,10}$	-1.398828	-1.398571	-1.397704	-1.400936	-1.398958	-1.398713
12	$Q_{G,12}$	-0.782852	-0.782880	-0.782837	-0.782681	-0.782586	-0.782905
15	$Q_{G,15}$	0.971787	0.963751	0.969391	0.961633	0.962742	0.977943
18	$Q_{G,18}$	0.015819	0.012754	0.009240	0.019536	0.026181	0.008460
19	$Q_{G,19}$	0.253150	0.258450	0.265945	0.262055	0.257586	0.257935
24	$Q_{G,24}$	-0.257769	-0.257458	-0.256812	-0.256756	-0.257845	-0.257366
25	$Q_{G,25}$	-0.339550	-0.338823	-0.336662	-0.337307	-0.340399	-0.338072
26	$Q_{G,26}$	1.741262	1.741467	1.742809	1.739498	1.744206	1.740741
27	$Q_{G,27}$	-1.434980	-1.434839	-1.433847	-1.435158	-1.434287	-1.434698
31	$Q_{G,31}$	-0.042763	-0.042111	-0.042940	-0.041866	-0.042875	-0.041594
32	$Q_{G,32}$	0.229519	0.230633	0.231957	0.232721	0.231330	0.229490
34	$Q_{G,34}$	-0.292623	-0.294968	-0.307017	-0.291220	-0.296953	-0.291710
36	$Q_{G,36}$	0.154919	0.150709	0.165193	0.162567	0.160056	0.158156
40	$Q_{G,40}$	0.003472	0.015538	-0.006359	-0.005604	0.011858	0.001931
42	$Q_{G,42}$	0.023858	0.020356	0.033797	0.004847	0.021431	0.024121
46	$Q_{G,46}$	0.285472	0.283955	0.287702	0.283992	0.284230	0.285348
49	$Q_{G,49}$	-0.155793	-0.155690	-0.155826	-0.156463	-0.155075	-0.155969
54	$Q_{G,54}$	1.513185	1.513561	1.515005	1.509829	1.512731	1.511451
55	$Q_{G,4}$	-0.017055	-0.022107	-0.013081	-0.005827	-0.014997	-0.011648
56	$Q_{G,4}$	-0.105972	-0.093991	-0.104156	-0.104744	-0.104453	-0.106351
59	$Q_{G,59}$	-0.384861	-0.385141	-0.384272	-0.383467	-0.384536	-0.384201
61	$Q_{G,61}$	1.202072	1.198757	1.199866	1.199881	1.197028	1.199354
62	$Q_{G,62}$	-0.139798	-0.140583	-0.143909	-0.138571	-0.143153	-0.138882
65	$Q_{G,65}$	-0.198976	-0.199071	-0.199652	-0.198558	-0.199590	-0.198786
66	$Q_{G,66}$	-2.834082	-2.833887	-2.834151	-2.835200	-2.832442	-2.834491
69	$P_{G,69,slack}$	4.909298	4.918238	4.885018	4.915800	4.919740	4.920000
	$Q_{G,69,slack}$	1.982924	1.982504	1.983084	1.982553	1.983903	1.982534
70	$Q_{G,70}$	-0.097430	-0.096093	-0.088154	-0.088696	-0.100379	-0.091612
72	$Q_{G,72}$	-0.101291	-0.102230	-0.104763	-0.102921	-0.107160	-0.101954
73	$Q_{G,73}$	0.077356	0.075098	0.067459	0.073776	0.067917	0.066419
74	$Q_{G,74}$	-0.134369	-0.133198	-0.123995	-0.140591	-0.121640	-0.132168
76	$Q_{G,76}$	-0.135769	-0.129893	-0.129005	-0.129038	-0.134689	-0.127904
77	$Q_{G,4}$	0.300387	0.301238	0.301226	0.301004	0.300542	0.300770
80	$Q_{G,80}$	2.244157	2.242265	2.238142	2.243779	2.243247	2.238449
85	$Q_{G,85}$	0.032898	0.032589	0.032785	0.032751	0.032871	0.032625
87	$Q_{G,87}$	0.110624	0.110530	0.110670	0.110645	0.110841	0.110608
89	$Q_{G,89}$	0.035344	0.038647	0.036407	0.036023	0.036871	0.037105
90	$Q_{G,90}$	0.587836	0.587800	0.587831	0.587921	0.587344	0.587735
91	$Q_{G,91}$	-0.129212	-0.129103	-0.129782	-0.129395	-0.129469	-0.129228
92	$Q_{G,92}$	-0.215743	-0.217884	-0.216381	-0.216135	-0.216572	-0.216939
99	$Q_{G,99}$	-0.175508	-0.175459	-0.175643	-0.175564	-0.175522	-0.175552
100	$Q_{G,100}$	0.958210	0.956892	0.958006	0.957811	0.957673	0.957248
103	$Q_{G,103}$	0.649023	0.647009	0.646192	0.648180	0.648223	0.648470
104	$Q_{G,104}$	-0.018858	-0.025822	-0.012894	-0.021099	-0.016144	-0.018738
105	$Q_{G,105}$	-0.071154	-0.074031	-0.063328	-0.063704	-0.070870	-0.072325
107	$Q_{G,107}$	0.001368	-0.000803	-0.002338	-0.009544	-0.004112	-0.000962
110	$Q_{G,110}$	0.023666	0.025203	0.013454	0.026606	0.017339	0.028134
111	$Q_{G,111}$	-0.016973	-0.016933	-0.016416	-0.017206	-0.017001	-0.016083
112	$Q_{G,112}$	0.227266	0.229179	0.233149	0.227120	0.233750	0.226546
113	$Q_{G,113}$	0.151335	0.144945	0.144509	0.146311	0.141069	0.146633
116	$Q_{G,116}$	-1.591606	-1.590561	-1.592593	-1.591750	-1.592117	-1.590389
Toplam Q_{gen} (pu)		4.112020	4.107496	4.102154	4.120250	4.098161	4.107908
Q_{loss} (pu)		-10.267980	-10.272504	-10.277846	-10.259750	-10.281838	-10.272092

Tablo 7. 30 çalışmaya ait istatistiki değerler (ncEPD)

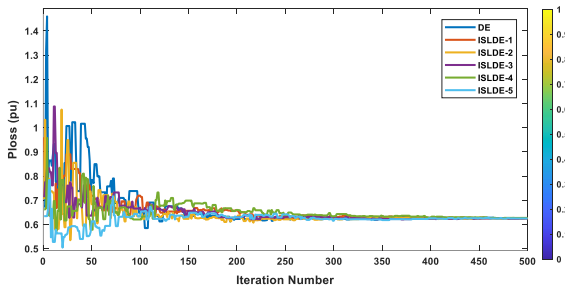
	DE	ISLDE-1	ISLDE-2	ISLDE-3	ISLDE-4	ISLDE-5
Eniye	135411.228474 (Run: 12)	135411.362343 (Run: 17)	135415.637539 (Run: 12)	135414.267529 (Run: 10)	135413.386380 (Run: 10)	135411.175400 (Run: 17)
Ortalama	135430.557018	135524.451218	135491.816324	135531.629631	135501.751667	135489.683171
Enkötü	135487.556184 (Run: 23)	135840.561056 (Run: 23)	135759.4555 (Run: 13)	135972.070818 (Run: 24)	135813.009314 (Run: 2)	135727.341142 (Run: 11)
StdSapma	19.606033	118.094651	97.524139	132.384474	104.725298	94.427856
Toplam süre (s)	44797.5	46073.7	45614.7	44600.7	45062.1	44917.2
Ortalama süre (s)	1493.25	1535.79	1520.49	1486.69	1502.07	1497.24

Eniyi yakıt maliyetinin elde edildiği çalışmaya ait yakınsama eğrileri TFC için Şekil 6'da, P_{loss} için Şekil 7'de verilmiştir. 30 çalışmaya ait kutu grafikleri ise Şekil 8'de verilmiştir.



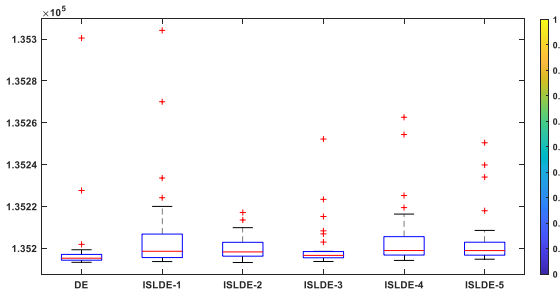
Şekil 6. Toplam yakıt maliyeti değerlerinin iterasyon sayısına göre değişimleri (cEPD)

Grafik incelendiğinde, ISLDE tabanlı algoritmaların başlangıç iterasyonlarından itibaren daha hızlı bir iyileşme sağladığı görülmektedir. Özellikle ISLDE-3 ve ISLDE-5 algoritmaları, hem daha düşük nihai maliyet değerlerine ulaşmakta hem de konverjans sürecinde daha kararlı bir performans sergilemektedir. Bu sonuçlar, sosyal öğrenme yapısının klasik DE algoritmasına entegrasyonunun, arama sürecini daha etkili hale getirdiğini göstermektedir.



Şekil 7. İletim hattı kayıplarının iterasyon sayısına göre değişimleri (cEPD)

Şekil 7'de verilen iletim hatlarında meydana gelen toplam aktif güç kaybı grafikleri incelendiğinde ise ilk 100 iterasyonda özellikle klasik DE algoritmasında yüksek dalgalanmalar mevcuttur, ISLDE algoritmaları ise daha kontrollü ve düşük dalgalı bir seyir izlemiştir.



Şekil 8. 30 çalışma için kutu grafikleri (cEPD)

Şekil 8'de verilen 30 çalışma için kutu grafikleri incelendiğinde ise bütün algoritmalarda sapan değerler mevcuttur. Enaz sapan değer 2 değerle ISLDE-2'de iken, en fazla sapan değer 6 tane ile ISLDE-3'dedir. ISLDE-1 yapısının diğer artırılmış öğrenme yapılarına göre daha zayıf olduğu bu grafikte de görülmektedir.

Verilen bütün tablo ve grafiklere genel olarak bakıldığında, konveks yapıda olan fonksiyonlar için artırılmış sosyal öğrenme kavramının kullanılmasının, klasik algoritma üzerinde kararlılık, arama aralığı ve süre bakımından olumlu etkiler gösterdiği söylenebilir.

4.2 Durum-2: Konveks olmayan ekonomik güç dağıtım problemi (ncEPD)

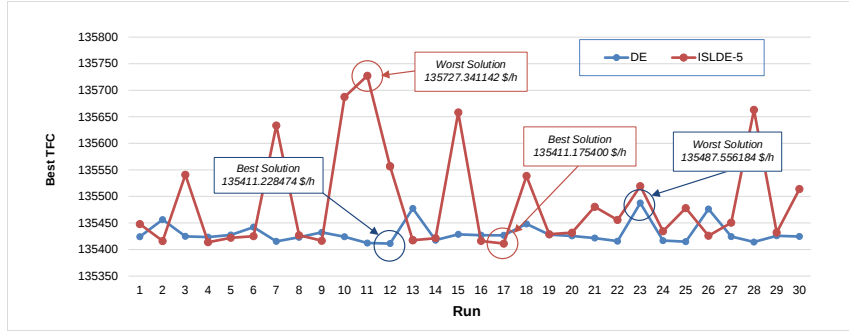
İkinci olarak, örnek güç sisteminin, Denklem (18)'de verilen konveks olmayan yakıt maliyeti fonksiyonu ile çözümü yapılmıştır. Örnek problem klasik algoritma ve geliştirilen diğer beş algoritma ile 30'ar kez çözülmüştür. Bu 30 çalışmadan eniyi yakıt maliyetinin elde edildiği çözüme ait aktif güç üretim değerleri Tablo 5'de, reaktif güç üretim değerleri Tablo 6'da verilmiştir.

Tablo 5 incelendiğinde 30 çalışma için eniyi yakıt maliyeti değerlerini, farklı aktif güç üretimi ve farklı iletim hattı kayıpları ile üç algoritma (DE, ISLDE-1 ve ISLDE-5) yakalamıştır. Süre bakımından en hızlı algoritma 1417.238985 s ile ISLDE-3 algoritması olmuştur.

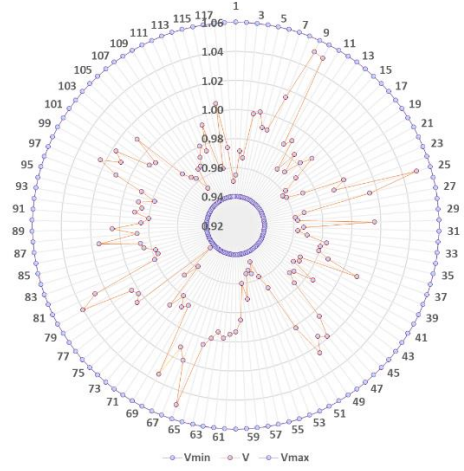
Konveks olmayan maliyet fonksiyonu ile çözümü yapılan sistemde, algoritmaların birbirlerine olan üstünlükleri ve kararlılıklarının değerlendirilebilmesi için, 30 çalışmanın tamamında elde edilen istatistiki değerler Tablo 7'de verilmiştir.

Tablo 7 incelendiğinde üç algoritma (DE, ISLDE-1 ve ISLDE-5), 30 çalışma için eniyi çözüm bakımından benzer yakıt maliyeti değerlerini yakalamış, diğer algoritmalar bu maliyet değerinin bir miktar uzağında kalmışlardır. Eniyi yakıt maliyeti değeri ISLDE-5 algoritmasıyla 135411.175400 \$/h olarak 17. çalışmada elde edilmiştir. 30 çalışmada elde edilen standart sapma değerlerine göre konveks olmayan fonksiyonlar için en kararlı algoritmanın klasik algoritma olduğu görülmüştür. En kararsız yapı ise ISLDE-3 olmuştur. Diğer bütün yapılar klasik DE'ye göre daha kararlı hale gelmesine rağmen ISLDE-1'de kararlılığın klasik algoritmaya göre iyileştirilemediği görülmüştür. Klasik algoritma ile 30 çalışmada elde edilen eniyi yakıt maliyeti değerleri ile artırılmış sosyal öğrenme yapılarından eniyi yakıt maliyetini elde eden ISLDE-5 algoritmasının 30 çalışmada elde ettiği değerler Şekil 9'da verilmiştir.

Bütün sonuçlar arasında eniyi yakıt maliyeti değerinin bulunduğu durum için (ISLDE-5, Run:17) sistemdeki bara gerilim değerlerinin grafiği ise Şekil 10'da verilmiştir. Şekilden de görüldüğü üzere bütün bara gerilimleri verilen alt ve üst gerilim değerleri arasında yer almaktadır. Örnek istemde yer alan 54 adet üretim birimi bulunan baraların tamamı gerilim kontrollü baralar olduğundan, bara gerilim değerleri yük akışı süresince değişmemiştir.

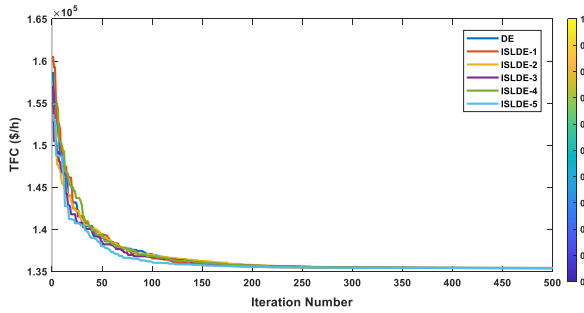


Şekil 9. 30 çalışmada elde edilen TFC değerleri (ncEPD)



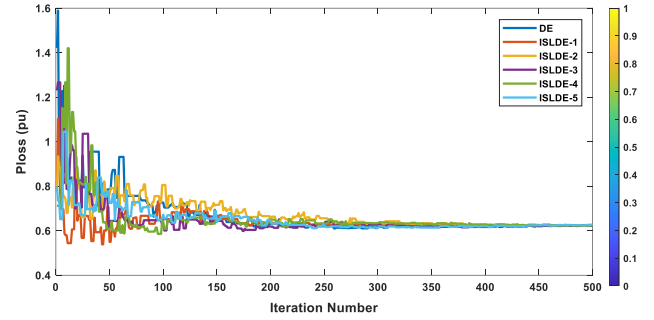
Şekil 10. ISLDE-5, Run:17 için bara gerilim değerleri (ncEPD)

ncEPD probleminin 30 kez çözümünde eniyi yakıt maliyetinin elde edildiği çalışmaya ait yakınsama eğrileri TFC için Şekil 11'de, P_{loss} için Şekil 12'de verilmiştir. 30 çalışmaya ait kutu grafikleri ise Şekil 13'de verilmiştir.

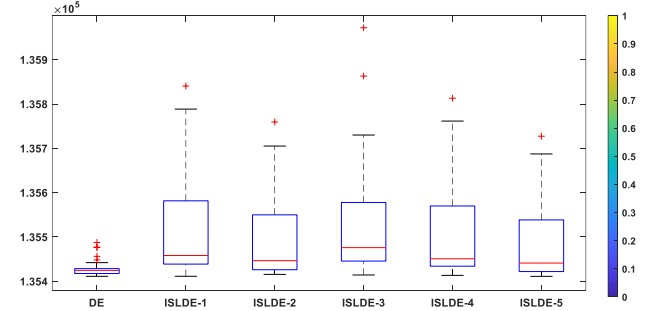


Şekil 11. Toplam yakıt maliyeti değerlerinin iterasyon sayısına göre değişimleri (ncEPD)

TFC için yakınsama eğrilerine bakıldığında ISLDE-5 yapısının diğer algoritmalarla göre daha hızlı yakınsadığı görülmektedir. En yavaş yakınsama ISLDE-2'de olmuştur. 30 çalışma için kutu grafikleri incelendiğinde ise bütün algoritmalarda sapan değerler mevcuttur. En çok sapan değer 4 değerle DE algoritması olmasına rağmen arama aralığı en küçük olan algoritma yine DE algoritmasıdır. Diğer algoritmaların konveks olmayan fonksiyonlar için kararlılıkları, klasik algoritmaya göre daha zayıftır.



Şekil 12. İletim hattı kayıplarının iterasyon sayısına göre değişimleri (ncEPD)



Şekil 13. 30 çalışma için kutu grafikleri (ncEPD)

Konveks olmayan maliyet fonksiyonları için verilen bütün tablo ve grafiklere genel olarak bakıldığında, artırımı sosyal öğrenme kavramının kullanılmasının, klasik algoritma üzerinde işlem hızı ya da en iyi sonucu bulma konusunda olumlu etkileri görünmesine rağmen kararlılık ve arama aralığı bakımından olumsuz etkiler gösterebileceği söylenebilir. Daha uygun artırımı öğrenme yapılarının geliştirilmesi, algoritma parametrelerinin probleme özgü olarak optimize edilmesi ile daha olumlu etkilerin elde edilebileceği görülmektedir.

5 İstatistiksel analizler

Bu çalışmada, önerilen artırımı sosyal öğrenme tabanlı diferansiyel gelişim algoritmalarının (ISLDE-1, ISLDE-2, ISLDE-3, ISLDE-4 ve ISLDE-5) performanslarının klasik diferansiyel gelişim algoritması (DE) ile karşılaştırılması amacıyla her algoritma için toplam 30 tekrar yapılmıştır. Elde edilen toplam yakıt maliyeti değerleri üzerinden istatistiksel anlamlılık testleri bu bölümde gerçekleştirilmiştir. Optimizasyon sonuçlarının normal dağılım göstermemesi nedeniyle non-parametrik testler tercih edilmiştir. İkili karşılaştırmalar için Wilcoxon işaretli sıra testi (*signrank*) ve Mann-Whitney U testi (*ranksum*) uygulanmıştır [43-46]. Analizler ayrı ayrı hem konveks hem de konveks olmayan yakıt maliyeti fonksiyonları için gerçekleştirilmiştir. Konveks ve konveks olmayan fonksiyonlar için elde edilen sonuçlar Tablo 8’de verilmiştir.

Konveks olmayan fonksiyonlarla yapılan çalışmalarda da benzer şekilde her algoritma için 30 tekrar yapılmış ve “ranksum” ile “signrank” testleri uygulanmıştır. Tablo Y’de özetlenen test sonuçları, klasik DE algoritması ile tüm ISLDE algoritmaları arasında anlamlı farklar olduğunu ortaya koymuştur ($p < 0.05$). ISLDE yapılarının kendi aralarındaki farkların ise çoğunlukla anlamsız olduğu gözlemlenmiştir. Ayrıca altı farklı algoritmanın performansları arasındaki farkın istatistiksel olarak anlamlı olup olmadığını değerlendirmek amacıyla, elde edilen toplam yakıt maliyeti değerlerine Kruskal-Wallis testi de uygulanmıştır. Test sonucunda algoritmalar arasında %95 güven aralığında anlamlı farklılıklar gözlenmiştir ($p < 0.05$).

Yapılan istatistiksel analizler, önerilen ISLDE yapıların hem konveks hem de konveks olmayan EPD problemleri için klasik algoritmaya göre daha düşük toplam yakıt maliyeti sonuçları ürettiğini ve bu farkların istatistiksel olarak anlamlı olduğunu göstermektedir.

6 Sonuçlar

Çalışmada, DE algoritmasına artırımı sosyal öğrenme yapısı entegre edilerek geliştirilen ISLDE algoritmalarıyla, konveks ve konveks olmayan EPD probleminin çözümü başarıyla yapılmıştır. İki farklı durum için ele alınan EPD probleminin çözümü için, örnek sistem olarak yüksek boyutlu modifiye edilmiş IEEE 118 baralı test sistemi seçilmiştir. Örnek sistemin çözümleri yapılırken iletim hattı kayıpları da dikkate alınmış ve bu kayıplar Newton-Raphson metoduyla AC güç akışı yapılarak hesaplanmıştır.

Çalışmada iki farklı durum için örnek sistem çözülmüştür. İlk durumda, 30 çalışma için eniyi yakıt maliyeti değeri 135193.225528 \$/h olarak ISLDE-3 algoritması ile hesaplanırken, süre bakımından en hızlı algoritma 1433.288900 s ile ISLDE-5 algoritması olmuştur. İkinci durumda ise 30 çalışma için eniyi yakıt maliyeti değeri ISLDE-5 algoritmasıyla 135411.175400 \$/h olarak 17. çalışmada elde edilmiştir. Çözüm süresi bakımından ise en hızlı algoritma 1417.238985 s ile ISLDE-3 algoritması olmuştur. Bu sonuçlara bakıldığında klasik DE algoritmasına artırımı sosyal öğrenme yapısının entegre edilmesi ile hem eniyi sonuca ulaşma hem de çözüm süresi bakımından algoritmada iyileştirme sağlandığı görülmüştür. Bunun yanında farklı sosyal öğrenme yapılarının da algoritma geliştirme de kullanılabileceği önerilmiştir.

Çalışmanın önemli amaçlarından biri karmaşık, doğrusal olmayan ve çok sayıda kısıtı bir arada bulunduran optimizasyon problemlerinden olan, güç mühendisliğinde büyük bir öneme sahip ve yapılan literatür çalışmalarında geniş bir yer tutan EPD probleminin ISLDE algoritmaları ile de çözülebilirliğini göstermektir. Bir diğer amaç ise literatürde çok az sayıda olan AC yük akışı ile çözümü yapılabilecek yüksek boyutlu bir örnek sistemin literatüre kazandırılmasıdır.

Tablo 8. Algoritmalar arası istatistiksel test sonuçları

	cEPD		ncEPD	
	signrank	ranksum	signrank	ranksum
DE - ISLDE-1	0.013974564	0.001679756	0.000114992	0.000035923
DE - ISLDE-2	0.004991554	0.000253058	0.000019729	0.000587373
DE - ISLDE-3	0.158855499	0.013271805	0.000012381	0.000008883
DE - ISLDE-4	0.005319684	0.000110577	0.000261343	0.000283887
DE - ISLDE-5	0.004389618	0.000038307	0.008216736	0.009883401
ISLDE-1 - ISLDE-2	0.440521591	0.888302842	0.280214408	0.251880825
ISLDE-1 - ISLDE-3	0.205888223	0.195790758	0.861212520	0.795845542
ISLDE-1 - ISLDE-4	0.599935928	0.599689485	0.404834722	0.395267011
ISLDE-1 - ISLDE-5	0.975387164	0.539510317	0.360038829	0.153667235
ISLDE-2 - ISLDE-3	0.205888223	0.063532651	0.253644098	0.125970193
ISLDE-2 - ISLDE-4	0.628843002	0.630876292	0.452806505	0.739398819
ISLDE-2 - ISLDE-5	0.958990172	0.662734758	0.877402728	0.630876292
ISLDE-3 - ISLDE-4	0.135907785	0.027086318	0.530440091	0.233988916
ISLDE-3 - ISLDE-5	0.115608188	0.017649028	0.152860694	0.074827008
ISLDE-4 - ISLDE-5	0.813016851	1.000000000	0.718887562	0.411910837

Teşekkür

Kütahya Dumlupınar Üniversitesi, Akıllı Sistemler Tasarımı Uygulama ve Araştırma Merkezi (ASTAM)'a bu çalışma için gerekli bazı temel araştırma olanaklarını sağladığı için teşekkür ederim.

Çıkar çatışması

Yazar çıkar çatışması olmadığını beyan etmektedir.

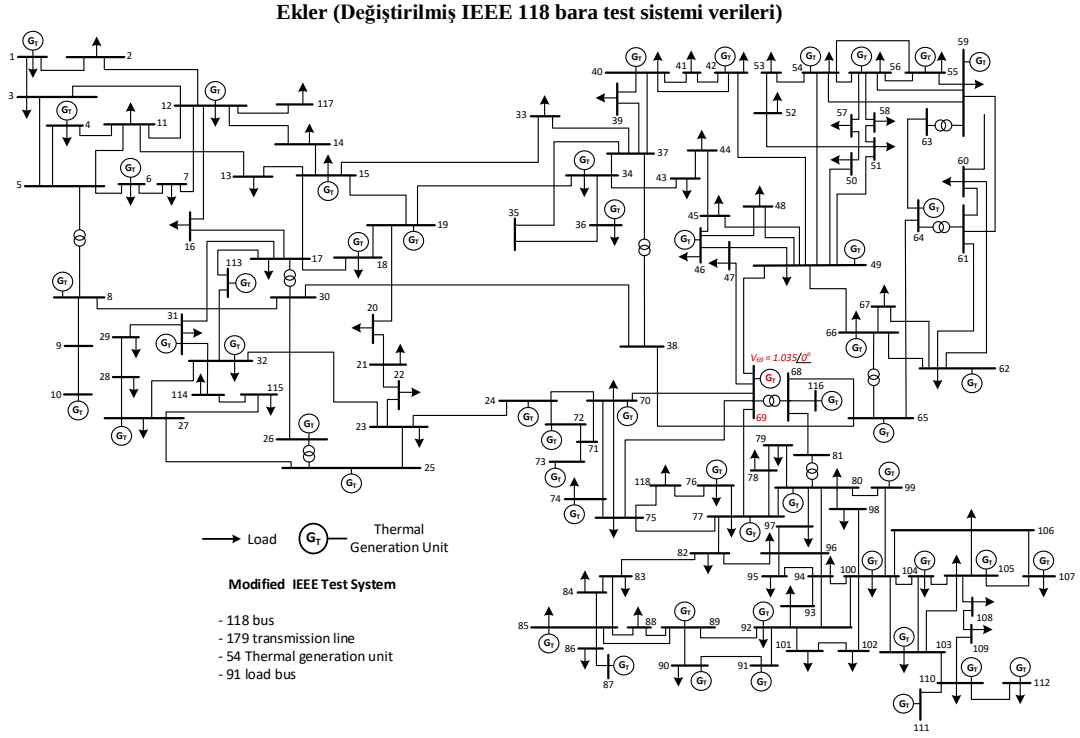
Benzerlik oranı (iThenticate): %15

Kaynaklar

- [1] S. Özyön, C. Yaşar, H. Temurtaş, A novel hybrid algorithm for nonconvex economic power dispatch problems. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 11(6), 92-101, 2016. <https://doi.org/10.9790/1676.11060392101>
- [2] C. Yaşar, S. Özyön, A new hybrid approach for nonconvex economic dispatch problem with valve point effect. *Energy*, 36(10), 5838-45, 2011. <https://doi.org/10.1016/j.energy.2011.08.041>
- [3] C. Dai, Z. Hu, Q. Su, An adaptive hybrid backtracking search optimization algorithm for dynamic economic dispatch with valve-point effects. *Energy*, 239(122461), 1-18, 2022. <https://doi.org/10.1016/j.energy.2021.122461>
- [4] A. Bhattacharya, P. K. Chattopadhyay, Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Trans Power Syst*, 25(4), 1955-64, 2010. <https://doi.org/10.1109/TPWRS.2010.2043270>
- [5] T. Liu, G. Xiong, A.W. Mohamed, P. N. Suganthan, Opposition-mutual learning differential evolution with hybrid mutation strategy for large-scale economic load dispatch problems with valve-point effects and multi-fuel options. *Inf Sci*, 609, 1721-45, 2022. <https://doi.org/10.1016/j.ins.2022.07.148>
- [6] M. A. Al-Betar, M.A. Awadallah, S.N. Makhadmeh, I.A. Doush, R.A. Zitar, S. Alshathri, M. Abd Elaziz, A hybrid Harris Hawks optimizer for economic load dispatch problems. *Alexandria Eng J*, 64, 365-89, 2023. <https://doi.org/10.1016/j.aej.2022.09.010>
- [7] M. H. Hassan, S. Kamel, F. Jurado, U. Desideri, Global optimization of economic load dispatch in large scale power systems using an enhanced social network search algorithm. *Int J Electr Power Energy Syst*, 156(109719), 1-30, 2024. <https://doi.org/10.1016/j.ijepes.2023.109719>
- [8] M. H. Hassan, S. Kamel, A. Eid, L. Nasrat, F. Jurado, M.F. Elnaggar, A developed eagle-strategy supply-demand optimizer for solving economic load dispatch problems. *Ain Shams Eng J*, 14(102083), 1-25, 2023. <https://doi.org/10.1016/j.asej.2022.102083>
- [9] Z. Hu, C. Dai, Q. Su, Adaptive backtracking search optimization algorithm with a dual-learning strategy for dynamic economic dispatch with valve-point effects. *Energy*, 248(123558), 1-16, 2022. <https://doi.org/10.1016/j.energy.2022.123558>
- [10] Y. Sharifian, H. Abdi, Solving multi-area economic dispatch problem using hybrid exchange market algorithm with grasshopper optimization algorithm. *Energy*, 267(126550), 1-12, 2023. <https://doi.org/10.1016/j.energy.2022.126550>
- [11] M. N. Iqbal, A. R. Bhatti, A. D. Butt, Y. A. Sheikh, K. N. Paracha, R. H. Ashique, Solution of economic dispatch problem using hybrid multi-verse optimizer. *Electr Power Syst Res*, 208(107912), 1-8, 2022. <https://doi.org/10.1016/j.epsr.2022.107912>
- [12] H. Sharifzadeh, Two efficient logarithmic formulations to solve nonconvex economic dispatch. *Electr Power Syst Res*, 229(110123), 1-10, 2024. <https://doi.org/10.1016/j.epsr.2024.110123>
- [13] W. Luo, X. Yu, Reinforcement learning-based modified cuckoo search algorithm for economic dispatch problems. *Knowledge-Based Syst*, 257(109844), 1-17, 2022. <https://doi.org/10.1016/j.knsys.2022.109844>
- [14] C. Chen, L. Qu, M. L. Tseng, L. Li, C. C. Chen, M. K. Lim, Reducing fuel cost and enhancing the resource utilization rate in energy economic load dispatch problem. *J Cleaner Prod*, 364(132709), 1-13, 2022. <https://doi.org/10.1016/j.jclepro.2022.132709>
- [15] X. Y. Zhang, W. K. Hao, J. S. Wang, J. H. Zhu, X. R. Zhao, Y. Zheng, Manta ray foraging optimization algorithm with mathematical spiral foraging strategies for solving economic load dispatching problems in power systems. *Alexandria Eng J*, 70, 613-40, 2023. <https://doi.org/10.1016/j.aej.2023.03.017>
- [16] M. H. Hassan, S. Kamel, F. Jurado, M. Ebeed, M. F. Elnaggar, Economic load dispatch solution of large-scale power systems using an enhanced beluga whale optimizer. *Alexandria Eng J*, 72, 573-91, 2023. <https://doi.org/10.1016/j.aej.2023.04.002>
- [17] L. Fu, H. Ouyang, C. Zhang, S. Li, A. W. Mohamed, A constrained cooperative adaptive multi-population differential evolutionary algorithm for economic load dispatch problems. *Appl Soft Comput*, 121(108719), 1-18, 2022. <https://doi.org/10.1016/j.asoc.2022.108719>
- [18] X. Yu, Q. Zhao, T. Wang, Y. Li, An antigravity-based fuzzy Gravitational Search Algorithm for economic dispatch problems. *Appl Soft Comput*, 145(110630), 1-23, 2023. <https://doi.org/10.1016/j.asoc.2023.110630>
- [19] T. Visutarnom, T. C. Chiang, Economic dispatch using metaheuristics: Algorithms, problems, and solutions. *Appl Soft Comput*, 150(110891), 1-43, 2024. <https://doi.org/10.1016/j.asoc.2023.110891>
- [20] J. S. Pan, J. Shan, S. C. Chu, S. J. Jiang, S. G. Zheng, L. Liao, A multigroup marine predator algorithm and its application for the power system economic load dispatch. *Energy Sci Eng*, 10(6), 1840-54, 2022. <https://doi.org/10.1002/ese3.957>
- [21] R. P. Parouha, P. Verma, An innovative hybrid algorithm to solve nonconvex economic load dispatch problem with or without valve point effects. *Int Trans*

- Electr Energy Syst, 31(1), 1-67, 2021. <https://doi.org/10.1002/2050-7038.12682>
- [22] R. Ramalingam, D. Karunanidhy, S. S. Alshamrani, M. Rashid, S. Mathumohan, A. Dumka, Oppositional pigeon-inspired optimizer for solving the non-convex economic load dispatch problem in power systems. *Mathematics*, 10(18)(3315), 1-24, 2022. <https://doi.org/10.3390/math10183315>
- [23] Q. Liu, G. Xiong, X. Fu, A. W. Mohamed, J. Zhang, M. A. Al-Betar, S. Xu, Hybridizing gaining-sharing knowledge and differential evolution for large-scale power system economic dispatch problems. *J Comput Des Eng*, 10(2), 615-31, 2023. <https://doi.org/10.1093/jcde/qwad008>
- [24] M. Braik, M. A. Awadallah, M. A. Al-Betar, A. I. Hammouri, A hybrid capuchin search algorithm with gradient search algorithm for economic dispatch problem. *Soft Comput*, 27(22), 16809-41, 2023. <https://doi.org/10.1007/s00500-023-09019-6>
- [25] T. Singh, Chaotic slime mould algorithm for economic load dispatch problems. *Appl Intell*, 52(13), 15325-44, 2023. <https://doi.org/10.1007/s10489-022-03179-y>
- [26] M. S. Braik, M. A. Awadallah, M. A. Al-Betar, A. I. Hammouri, R. A. Zitar, A non-convex economic load dispatch problem using chameleon swarm algorithm with roulette wheel and Levy flight methods. *Appl Intell*, 53, 17508-47, 2023. <https://doi.org/10.1007/s10489-022-04363-w>
- [27] V. Garg, K. Deep, N. P. Padhee, Constrained laplacian biogeography-based optimization for economic load dispatch problems. *Process Integr Optim Sustainability*, 6(2), 483-96, 2022. <https://doi.org/10.1007/s41660-022-00227-5>
- [28] M. A. Montes de Oca, T. Stützle, K. Van den Enden, M. Dorigo, Incremental social learning in particle swarms. *IEEE Trans. Syst. Man Cybern. Part B Cybern*, 41(2), 368-84, 2011. <https://doi.org/10.1109/TSMCB.2010.2055848>
- [29] T. Liao, M. A. Montes de Oca, D. Aydın, T. Stützle, M. Dorigo, An incremental ant colony algorithm with local search for continuous optimization problems. In: *Proceeding of genetic and evolutionary computation conference (GECCO 2011)*, 125-32. <https://doi.org/10.1145/2001576.2001594>
- [30] D. Aydın, S. Özyön, Solution to non-convex economic dispatch problem with valve point effects by incremental artificial bee colony with local search. *Appl. Soft Comput*, 13(5), 2456-66, 2013. <https://doi.org/10.1016/j.asoc.2012.12.002>
- [31] S. Özyön, D. Aydın, Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones. *Energy Convers. Manage*, 65, 397-407, 2013. <https://doi.org/10.1016/j.enconman.2012.07.005>
- [32] D. Aydın, S. Özyön, C. Yaşar, L. Tianjun, Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem. *Int. J. Electr. Power Energy Syst*, 54, 144-153, 2014. <https://doi.org/10.1016/j.ijepes.2013.06.020>
- [33] S. Özyön, C. Yaşar, H. Temurtaş, Incremental gravitational search algorithm for high-dimensional benchmark functions. *Neural Comput. Appl*, 31(8), 3779-3803, 2019. <https://doi.org/10.1007/s00521-017-3334-8>
- [34] C. Yaşar, S. Özyön, A modified incremental gravitational search algorithm for short-term hydrothermal scheduling with variable head. *Eng. Appl. Artif. Intell*, 95(103845), 1-17, 2020. <https://doi.org/10.1016/j.engappai.2020.103845>
- [35] S. Özyön, Differential evolution algorithm with incremental social learning. *Bilecik Şeyh Edebali University J. Sci. Technol*, 7, 133-63, 2020. <https://doi.org/10.35193/bseufbd.666626>
- [36] D. Karaboğa, *Artificial Intelligence Optimization Algorithms (Turkish)*, Atlas Publishing Distribution, First Edition, Turkey, 2004.
- [37] A. J. Wood, B. F. Wollenberg, G. B. Sheble, *Power Generation Operation and Control*, IEEE & Wiley, Third Edition, USA, 2013.
- [38] D. P. Kothari, J. S. Dhillon. *Power System Optimization*, PHI, New Delhi, 2007.
- [39] P. Rossoni, W. M. Rosa, E. A. Belati, Linearized AC load flow applied to analysis in electric power systems, *IEEE Lat. Am. Trans*, 14(9), 4048-53, 2016. <https://doi.org/10.1109/TLA.2016.7785932>
- [40] C. Yaşar, S. Özyön, H. Temurtaş, A new program design developed for AC load flow analysis problems. *Kırıkkale University Faculty of Engineering Int. J. Eng. Res. Dev*, 9(3), 207-22, 2017. <https://doi.org/10.29137/umagd.372979>
- [41] J. Lin, F. H. Magnago, *Power System Economic Dispatch*, in *Electricity Markets: Theories and Applications*, Wiley-IEEE Press, 119-46, 2017. <https://doi.org/10.1002/9781119179382.ch5>
- [42] Al-roomi Power. <https://www.al-roomi.org/power-flow/118-bus-system> . Erişim tarihi 02 Ocak 2025.
- [43] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, 15, 617-44, 2019.
- [44] H. Zhang, Z. Wang, D. Liu, A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7), 1229-62, 2014.
- [45] Matlab Help Center. <https://www.mathworks.com/help/stats/ranksum.html>. Erişim tarihi 02 Ocak 2025.
- [46] Matlab Help Center. <https://www.mathworks.com/help/stats/signrank.html>. Erişim tarihi 02 Ocak 2025.





Şekil A1. Örnek sisteme ait tek hat diyagramı

Tablo A1. Nominal π eşdeğer devresine ait iletim hattı değerleri

Line no	From bus	To bus	R (pu)	X (pu)	B (pu)
1	1	2	0.03030	0.09990	0.02540
2	1	3	0.01290	0.04240	0.01082
3	4	5	0.00176	0.00798	0.00210
4	3	5	0.02410	0.10800	0.02840
5	5	6	0.01190	0.05400	0.01426
6	6	7	0.00459	0.02080	0.00550
7	8	9	0.00244	0.03050	1.16200
8	8	5	-	0.02670	-
9	9	10	0.00258	0.03220	1.23000
10	4	11	0.02090	0.06880	0.01748
11	5	11	0.02030	0.06820	0.01738
12	11	12	0.00595	0.01960	0.00502
13	2	12	0.01870	0.06160	0.01572
14	3	12	0.04840	0.16000	0.04060
15	7	12	0.00862	0.03400	0.00874
16	11	13	0.02225	0.07310	0.01876
17	12	14	0.02150	0.07070	0.01816
18	13	15	0.07440	0.24440	0.06268
19	14	15	0.05950	0.19500	0.05020
20	12	16	0.02120	0.08340	0.02140
21	15	17	0.01320	0.04370	0.04440
22	16	17	0.04540	0.18010	0.04660
23	17	18	0.01230	0.05050	0.01298
24	18	19	0.01119	0.04930	0.01142
25	19	20	0.02520	0.11700	0.02980
26	15	19	0.01200	0.03940	0.01010
27	20	21	0.01830	0.08490	0.02160
28	21	22	0.02090	0.09700	0.02460

29	22	23	0.03420	0.15900	0.04040
30	23	24	0.01350	0.04920	0.04980
31	23	25	0.01560	0.08000	0.08640
32	26	25	-	0.03820	-
33	25	27	0.03180	0.16300	0.17640
34	27	28	0.01913	0.08550	0.02160
35	28	29	0.02370	0.09430	0.02380
36	30	17	-	0.03880	-
37	8	30	0.00431	0.05040	0.51400
38	26	30	0.00799	0.08600	0.90800
39	17	31	0.04740	0.15630	0.03990
40	29	31	0.01080	0.03310	0.00830
41	23	32	0.03170	0.11530	0.11730
42	31	32	0.02980	0.09850	0.02510
43	27	32	0.02290	0.07550	0.01926
44	15	33	0.03800	0.12440	0.03194
45	19	34	0.07520	0.24700	0.06320
46	35	36	0.00224	0.01020	0.00268
47	35	37	0.01100	0.04970	0.01318
48	33	37	0.04150	0.14200	0.03660
49	34	36	0.00871	0.02680	0.00568
50	34	37	0.00256	0.00940	0.00984
51	38	37	-	0.03750	-
52	37	39	0.03210	0.10600	0.02700
53	37	40	0.05930	0.16800	0.04200
54	30	38	0.00464	0.05400	0.42200
55	39	40	0.01840	0.06050	0.01552
56	40	41	0.01450	0.04870	0.01222
57	40	42	0.05550	0.18300	0.04660
58	41	42	0.04100	0.13500	0.03440
59	43	44	0.06080	0.24540	0.06068
60	34	43	0.04130	0.16810	0.04226
61	44	45	0.02240	0.09010	0.02240
62	45	46	0.04000	0.13560	0.03320
63	46	47	0.03800	0.12700	0.03160
64	46	48	0.06010	0.18900	0.04720
65	47	49	0.01910	0.06250	0.01604
66	42	49	0.03575	0.16150	0.17200
67	45	49	0.06840	0.18600	0.04440
68	48	49	0.01790	0.05050	0.01258
69	49	50	0.02670	0.07520	0.01874
70	49	51	0.04860	0.13700	0.03420
71	51	52	0.02030	0.05880	0.01396
72	52	53	0.04050	0.16350	0.04058
73	53	54	0.02630	0.12200	0.03100
74	49	54	0.03993	0.14507	0.14680
75	54	55	0.01690	0.07070	0.02020
76	54	56	0.00275	0.00955	0.00732
77	55	56	0.00488	0.01510	0.00374
78	56	57	0.03430	0.09660	0.02420
79	50	57	0.04740	0.13400	0.03320
80	56	58	0.03430	0.09660	0.02420
81	51	58	0.02550	0.07190	0.01788
82	54	59	0.05030	0.22930	0.05980
83	56	59	0.04070	0.12243	0.11050
84	55	59	0.04739	0.21580	0.05646
85	59	60	0.03170	0.14500	0.03760

86	59	61	0.03280	0.15000	0.03880
87	60	61	0.00264	0.01350	0.01456
88	60	62	0.01230	0.05610	0.01468
89	61	62	0.00824	0.03760	0.00980
90	63	59	-	0.03860	-
91	63	64	0.00172	0.02000	0.21600
92	64	61	-	0.02680	-
93	38	65	0.00901	0.09860	1.04600
94	64	65	0.00269	0.03020	0.38000
95	49	66	0.00900	0.04595	0.04960
96	62	66	0.04820	0.21800	0.05780
97	62	67	0.02580	0.11700	0.03100
98	65	66	-	0.03700	-
99	66	67	0.02240	0.10150	0.02682
100	65	68	0.00138	0.01600	0.63800
101	47	69	0.08440	0.27780	0.07092
102	49	69	0.09850	0.32400	0.08280
103	68	69	-	0.03700	-
104	69	70	0.03000	0.12700	0.12200
105	24	70	0.00221	0.41150	0.10198
106	70	71	0.00882	0.03550	0.00878
107	24	72	0.04880	0.19600	0.04880
108	71	72	0.04460	0.18000	0.04444
109	71	73	0.00866	0.04540	0.01178
110	70	74	0.04010	0.13230	0.03368
111	70	75	0.04280	0.14100	0.03600
112	69	75	0.04050	0.12200	0.12400
113	74	75	0.01230	0.04060	0.01034
114	76	77	0.04440	0.14800	0.03680
115	69	77	0.03090	0.10100	0.10380
116	75	77	0.06010	0.19990	0.04978
117	77	78	0.00376	0.01240	0.01264
118	78	79	0.00546	0.02440	0.00648
119	77	80	0.01088	0.03321	0.07000
120	79	80	0.01560	0.07040	0.01870
121	68	81	0.00175	0.02020	0.80800
122	81	80	-	0.03700	-
123	77	82	0.02980	0.08530	0.08174
124	82	83	0.01120	0.03665	0.03796
125	83	84	0.06250	0.13200	0.02580
126	83	85	0.04300	0.14800	0.03480
127	84	85	0.03020	0.06410	0.01234
128	85	86	0.03500	0.12300	0.02760
129	86	87	0.02828	0.20740	0.04450
130	85	88	0.02000	0.10200	0.02760
131	85	89	0.02390	0.17300	0.04700
132	88	89	0.01390	0.07120	0.01934
133	89	90	0.01638	0.06517	0.15880
134	90	91	0.02540	0.08360	0.02140
135	89	92	0.00799	0.03829	0.09620
136	91	92	0.03870	0.12720	0.03268
137	92	93	0.02580	0.08480	0.02180
138	92	94	0.04810	0.15800	0.04060
139	93	94	0.02230	0.07320	0.01876
140	94	95	0.01320	0.04340	0.01110
141	80	96	0.03560	0.18200	0.04940
142	82	96	0.01620	0.05300	0.05440

143	94	96	0.02690	0.08690	0.02300
144	80	97	0.01830	0.09340	0.02540
145	80	98	0.02380	0.10800	0.02860
146	80	99	0.04540	0.20600	0.05460
147	92	100	0.06480	0.29500	0.04720
148	94	100	0.01780	0.05800	0.06040
149	95	96	0.01710	0.05470	0.01474
150	96	97	0.01730	0.08850	0.02400
151	98	100	0.03970	0.17900	0.04760
152	99	100	0.01800	0.08130	0.02160
153	100	101	0.02770	0.12620	0.03280
154	92	102	0.01230	0.05590	0.01464
155	101	102	0.02460	0.11200	0.02940
156	100	103	0.01600	0.05250	0.05360
157	100	104	0.04510	0.20400	0.05410
158	103	104	0.04660	0.15840	0.04070
159	103	105	0.05350	0.16250	0.04080
160	100	106	0.06050	0.22900	0.06200
161	104	105	0.00994	0.03780	0.00986
162	105	106	0.01400	0.05470	0.01434
163	105	107	0.05300	0.18300	0.04720
164	105	108	0.02610	0.07030	0.01844
165	106	107	0.05300	0.18300	0.04720
166	108	109	0.01050	0.02880	0.00760
167	103	110	0.03906	0.18130	0.04610
168	109	110	0.02780	0.07620	0.02020
169	110	111	0.02200	0.07550	0.02000
170	110	112	0.02470	0.06400	0.06200
171	17	113	0.00913	0.03010	0.00768
172	32	113	0.06150	0.20300	0.05180
173	32	114	0.01350	0.06120	0.01628
174	27	115	0.01640	0.07410	0.01972
175	114	115	0.00230	0.01040	0.00276
176	68	116	0.00034	0.00405	0.16400
177	12	117	0.03290	0.14000	0.03580
178	75	118	0.01450	0.04810	0.01198
179	76	118	0.01640	0.05440	0.01356

Tablo A2. Aktif ve reaktif yük değerleri

Bus No	$P_{load} (pu)$	$Q_{load} (pu)$	Bus No	$P_{load} (pu)$	$Q_{load} (pu)$
1	0.5100	0.2700	60	0.7800	0.0300
2	0.2000	0.0900	61	-	-
3	0.3900	0.1000	62	0.7700	0.1400
4	0.3900	0.1200	63	-	-
5	-	-	64	-	-
6	0.5200	0.2200	65	-	-
7	0.1900	0.0200	66	0.3900	0.1800
8	0.2800	0.0000	67	0.2800	0.0700
9	-	-	68	-	-
10	-	-	69	-	-
11	0.7000	0.2300	70	0.6600	0.2000
12	0.4700	0.1000	71	-	-
13	0.3400	0.1600	72	0.1200	-
14	0.1400	0.0100	73	0.0600	-
15	0.9000	0.3000	74	0.6800	0.2700
16	0.2500	0.1000	75	0.4700	0.1100

17	0.1100	0.0300	76	0.6800	0.3600
18	0.6000	0.3400	77	0.6100	0.2800
19	0.4500	0.2500	78	0.7100	0.2600
20	0.1800	0.0300	79	0.3900	0.3200
21	0.1400	0.0800	80	1.3000	0.2600
22	0.1000	0.0500	81	-	-
23	0.0700	0.0300	82	0.5400	0.2700
24	0.1300	-	83	0.2000	0.1000
25	-	-	84	0.1100	0.0700
26	-	-	85	0.2400	0.1500
27	0.7100	0.1300	86	0.2100	0.1000
28	0.1700	0.0700	87	-	-
29	0.2400	0.0400	88	0.4800	0.1000
30	-	-	89	-	-
31	0.4300	0.2700	90	1.6300	0.4200
32	0.5900	0.2300	91	0.1000	-
33	0.2300	0.0900	92	0.6500	0.1000
34	0.5900	0.2600	93	0.1200	0.0700
35	0.3300	0.0900	94	0.3000	0.1600
36	0.3100	0.1700	95	0.4200	0.3100
37	-	-	96	0.3800	0.1500
38	-	-	97	0.1500	0.0900
39	0.2700	0.1100	98	0.3400	0.0800
40	0.6600	0.2300	99	0.4200	-
41	0.3700	0.1000	100	0.3700	0.1800
42	0.9600	0.2300	101	0.2200	0.1500
43	0.1800	0.0700	102	0.0500	0.0300
44	0.1600	0.0800	103	0.2300	0.1600
45	0.5300	0.2200	104	0.3800	0.2500
46	0.2800	0.1000	105	0.3100	0.2600
47	0.3400	-	106	0.4300	0.1600
48	0.2000	0.1100	107	0.5000	0.1200
49	0.8700	0.3000	108	0.0200	0.0100
50	0.1700	0.0400	109	0.0800	0.0300
51	0.1700	0.0800	110	0.3900	0.3000
52	0.1800	0.0500	111	-	-
53	0.2300	0.1100	112	0.6800	0.1300
54	1.1300	0.3200	113	0.0600	-
55	0.6300	0.2200	114	0.0800	0.0300
56	0.8400	0.1800	115	0.2200	0.0700
57	0.1200	0.0300	116	1.8400	-
58	0.1200	0.0300	117	0.2000	0.0800
59	2.7700	1.1300	118	0.3300	0.1500
Toplam yük		P_{load}: 42.4200 pu		Q_{load}: 14.3800 pu	

Tablo A3. AC yük akış analizi başlangıç verileri

Gen. Unit	Bus No	P _i (pu)	Q _i (pu)	V (pu)	Gen. Unit	Bus No	P _i (pu)	Q _i (pu)	V (pu)
1	1	0.500	0.075	0.955	28	65	2.000	1.000	1.005
2	4	0.500	1.500	0.998	29	66	2.000	1.000	1.050
3	6	0.500	0.250	0.990	30	69	-	-	1.035
4	8	0.500	1.500	1.015	31	70	0.500	0.150	0.984
5	10	2.000	1.000	1.050	32	72	0.500	0.500	0.980
6	12	1.000	0.600	0.990	33	73	0.500	0.500	0.991
7	15	0.500	0.150	0.970	34	74	0.500	0.045	0.958
8	18	0.500	0.250	0.973	35	76	0.500	0.120	0.943

9	19	0.500	0.120	0.962	36	77	0.500	0.350	1.006
10	24	0.500	1.500	0.992	37	80	2.000	1.400	1.040
11	25	1.000	0.700	1.050	38	85	0.500	0.120	0.985
12	26	2.000	5.000	1.015	39	87	0.500	5.000	1.015
13	27	0.500	1.500	0.968	40	89	2.000	1.500	1.005
14	31	0.500	1.500	0.967	41	90	0.500	1500	0.985
15	32	0.500	0.210	0.963	42	91	0.500	0.500	0.980
16	34	0.500	0.120	0.984	43	92	0.500	0.045	0.990
17	36	0.500	0.120	0.980	44	99	0.500	0.500	1.010
18	40	0.500	1.500	0.970	45	100	1.000	0.750	1.017
19	42	0.500	1.500	0.985	46	103	0.500	0.200	1.010
20	46	0.500	0.500	1.005	47	104	0.500	0.120	0.971
21	49	1.000	1.000	1.025	48	105	0.500	0.120	0.965
22	54	0.500	1.500	0.955	49	107	0.500	1.000	0.952
23	55	0.500	0.120	0.952	50	110	0.500	0.120	0.973
24	56	0.500	0.075	0.954	51	111	0.500	5.000	0.980
25	59	1.000	0.900	0.985	52	112	0.500	5.000	0.975
26	61	1.000	1.500	0.995	53	113	0.500	1.000	0.993
27	62	0.500	0.100	0.998	54	116	0.500	5.000	1.005

Tablo A4. Üretim birimlerine ait işletim limitleri

Gen. Unit	Bus	P_{min} (pu)	P_{max} (pu)	Q_{min} (pu)	Q_{max} (pu)	V_{min} (pu)	V_{max} (pu)
1	1	0.0000	1.0000	-0.0500	0.1500	0.9400	1.0600
2	4	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
3	6	0.0000	1.0000	-0.1300	0.5000	0.9400	1.0600
4	8	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
5	10	0.0000	5.5000	-1.4700	2.0000	0.9400	1.0600
6	12	0.0000	1.8500	-0.3500	1.2000	0.9400	1.0600
7	15	0.0000	1.0000	-0.1000	0.3000	0.9400	1.0600
8	18	0.0000	1.0000	-0.1600	0.5000	0.9400	1.0600
9	19	0.0000	1.0000	-0.0800	0.2400	0.9400	1.0600
10	24	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
11	25	0.0000	3.2000	-0.4700	1.4000	0.9400	1.0600
12	26	0.0000	4.1400	-10.0000	10.0000	0.9400	1.0600
13	27	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
14	31	0.0000	1.0700	-3.0000	3.0000	0.9400	1.0600
15	32	0.0000	1.0000	-0.1400	0.4200	0.9400	1.0600
16	34	0.0000	1.0000	-0.0800	0.2400	0.9400	1.0600
17	36	0.0000	1.0000	-0.0800	0.2400	0.9400	1.0600
18	40	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
19	42	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
20	46	0.0000	1.1900	-1.0000	1.0000	0.9400	1.0600
21	49	0.0000	3.0400	-0.8500	2.1000	0.9400	1.0600
22	54	0.0000	1.4800	-3.0000	3.0000	0.9400	1.0600
23	55	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600
24	56	0.0000	1.0000	-0.0800	0.1500	0.9400	1.0600
25	59	0.0000	2.5500	-0.6000	1.8000	0.9400	1.0600
26	61	0.0000	2.6000	-1.0000	3.0000	0.9400	1.0600
27	62	0.0000	1.0000	-0.2000	0.2000	0.9400	1.0600
28	65	0.0000	4.9100	-0.6700	2.0000	0.9400	1.0600
29	66	0.0000	4.9200	-0.6700	2.0000	0.9400	1.0600
30	69	0.0000	8.0520	-3.0000	3.0000	0.9400	1.0600
31	70	0.0000	1.0000	-0.1000	0.3200	0.9400	1.0600
32	72	0.0000	1.0000	-1.0000	1.0000	0.9400	1.0600
33	73	0.0000	1.0000	-1.0000	1.0000	0.9400	1.0600
34	74	0.0000	1.0000	-0.0600	0.0900	0.9400	1.0600
35	76	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600

36	77	0.0000	1.0000	-0.2000	0.7000	0.9400	1.0600
37	80	0.0000	5.7700	-1.6500	2.8000	0.9400	1.0600
38	85	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600
39	87	0.0000	1.0400	-1.0000	10.0000	0.9400	1.0600
40	89	0.0000	7.0700	-2.1000	3.0000	0.9400	1.0600
41	90	0.0000	1.0000	-3.0000	3.0000	0.9400	1.0600
42	91	0.0000	1.0000	-1.0000	1.0000	0.9400	1.0600
43	92	0.0000	1.0000	-0.0300	0.0900	0.9400	1.0600
44	99	0.0000	1.0000	-1.0000	1.0000	0.9400	1.0600
45	100	0.0000	3.5200	-0.5000	1.5500	0.9400	1.0600
46	103	0.0000	1.4000	-0.1500	0.4000	0.9400	1.0600
47	104	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600
48	105	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600
49	107	0.0000	1.0000	-2.0000	2.0000	0.9400	1.0600
50	110	0.0000	1.0000	-0.0800	0.2300	0.9400	1.0600
51	111	0.0000	1.3600	-1.0000	10.0000	0.9400	1.0600
52	112	0.0000	1.0000	-1.0000	10.0000	0.9400	1.0600
53	113	0.0000	1.0000	-1.0000	2.0000	0.9400	1.0600
54	116	0.0000	1.0000	-10.0000	10.0000	0.9400	1.0600

Tablo A5. Üretim birimlerine ait konvex ve konveks olmayan maliyet katsayıları

Gen. Unit	Bus No.	a	b	c	e	f
1	1	0	4000	100	0	0
2	4	0	4000	100	0	0
3	6	0	4000	100	0	0
4	8	0	4000	100	0	0
5	10	0	2000	222.222222	100	8.4
6	12	0	2000	1176.47059		
7	15	0	4000	100	0	0
8	18	0	4000	100	0	0
9	19	0	4000	100	0	0
10	24	0	4000	100	100	8.4
11	25	0	2000	454.545455	120	7.7
12	26	0	2000	318.471338	0	0
13	27	0	4000	100	0	0
14	31	0	2000	14285.7143	0	0
15	32	0	4000	100	0	0
16	34	0	4000	100	0	0
17	36	0	4000	100	0	0
18	40	0	4000	100	0	0
19	42	0	4000	100	0	0
20	46	0	2000	5263.15786	0	0
21	49	0	2000	490.196078	0	0
22	54	0	2000	2083.33333	0	0
23	55	0	4000	100	0	0
24	56	0	4000	100	0	0
25	59	0	2000	645.16129	0	0
26	61	0	2000	625	0	0
27	62	0	4000	100	0	0
28	65	0	2000	255.754476	150	6.3
29	66	0	2000	255.102041	0	0
30	69	0	2000	193.648335	0	0
31	70	0	4000	100	0	0
32	72	0	4000	100	0	0
33	73	0	4000	100	0	0
34	74	0	4000	100	0	0
35	76	0	4000	100	0	0
36	77	0	4000	100	105	8.1

37	80	0	2000	209.643606	0	0
38	85	0	4000	100	0	0
39	87	0	2000	25000	0	0
40	89	0	2000	164.744646	0	0
41	90	0	4000	100	0	0
42	91	0	4000	100	0	0
43	92	0	4000	100	200	4.2
44	99	0	4000	100	200	4.2
45	100	0	2000	396.825397	200	4.2
46	103	0	2000	2500	0	0
47	104	0	4000	100	0	0
48	105	0	4000	100	0	0
49	107	0	4000	100	0	0
50	110	0	4000	100	0	0
51	111	0	2000	2777.77778	0	0
52	112	0	4000	100	0	0
53	113	0	4000	100	0	0
54	116	0	4000	100	0	0

