

154210

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

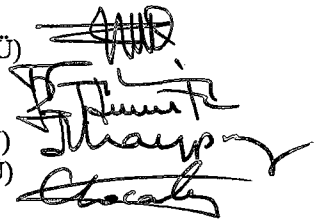
**DALLI ENERJİ DAĞITIM SİSTEMLERİNDE
YENİDEN YAPILANDIRMA PROBLEMLERİNE
GENETİK ALGORİTMA YAKLAŞIMI**

Elek. Yük. Müh. Ahmet ÖZDEŞ

F.B.E. Elektrik Mühendisliği Anabilim Dalında
Hazırlanan

DOKTORA TEZİ

Tez Savunma Tarihi : 19 Ocak 2004
Tez Danışmanı : Doç. Dr. F. Okan PEKİNER (YTÜ)
Jüri Üyeleri : Prof. Dr. Hüseyin ÇAKIR (YTÜ)
: Prof. Dr. Semra ÖZTÜRK (KOÜ)
: Prof. Dr. Adnan KAYPMAS (İTÜ)
: Prof. Dr. Celal KOCATEPE (YTÜ)



İSTANBUL, 2004

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTIMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ.....	vii
ÖNSÖZ.....	viii
ÖZET.....	ix
ABSTRACT.....	x
1. GİRİŞ.....	1
1.1 Tezin Amacı ve Konuya Ait Çözüm Yaklaşımı.....	1
1.2 Konunun Tanıtılması.....	1
1.3 Literatürde Yeniden Yapılandırma Problemi.....	3
2. GENETİK ALGORİTMALARIN TEORİ VE UYGULAMALARI.....	8
2.1 Genetik Algoritmalar.....	8
2.2 Geleneksel En İyi Arama Yöntemlerinin Dezavantajları.....	9
2.3 Genetik Algoritmalar ile Diğer Geleneksel Yöntemler Arasındaki Farklar.....	12
2.4 Basit Genetik Algoritma.....	13
2.5 Genetik Algoritma Uygulamasına Bir Örnek.....	17
3. ENERJİ DAĞITIM SİSTEMLERİNDE HATTIN AKTİF GÜÇ KAYIPLARINI MİNİMUM YAPACAK YENİDEN YAPILANDIRMA PROBLEMİNİN MATEMATİKSEL MODELİ.....	20
3.1 Giriş.....	20
3.2 Üç Fazlı Alternatif Akım Sistemlerinde Güç ve Akım İfadeleri.....	20
3.3 Üç Fazlı Alternatif Akım Sistemlerinde Aktif Güç Kaybı İfadesi.....	21
3.4 Enerji Dağıtım Sistemlerinde Hat Akış Şemalarının Çıkartılması.....	21
3.5 Enerji Dağıtım Sistemlerinde Toplam Aktif Güç Kaybının Hesaplanması.....	22
3.6 Hedef Fonksiyonunun Belirlenmesi.....	23
4. UYGULAMADA KULLANILAN ÖRNEK ENERJİ DAĞITIM SİSTEMİNE İLİŞKİN HAT VE YÜK VERİLERİ.....	24
4.1 Giriş.....	24
4.2 Örnek Enerji Dağıtım Sisteminin Tanıtılması.....	24
4.2.1 Hat Verileri.....	25
4.3 Bara Yük Verileri.....	27
5. ENERJİ DAĞITIM ŞEBEKELERİNİN YAPISI VE İŞLETME ŞARTLARININ	

	TANIMLANMASI.....	29
5.1	Enerji Dağıtım Sisteminin Dallı Yapısı.....	29
5.2	Enerji Dağıtım Sistemlerinde Yük Dengesinin Sağlanması.....	30
5.3	Yüklerin Enerjilenmesi.....	31
5.4	Gerilim Düşümü Kısıtları.....	31
5.5	Akım Taşıma Kapasiteleri.....	32
5.6	Hat Aktif ve Reaktif Güç Kayıplarının Sınırlanması	32
6.	YENİDEN YAPILANDIRMA PROBLEMİNE GENETİK ALGORİTMANIN UYGULANMASI	34
6.1	Giriş.....	34
6.2	Genetik Algoritma Metodu ile Enerji Dağıtım Sisteminin Tanımlanması ve Genel Akış Şemasının Çıkarılması	34
6.3	Enerji Dağıtım Sistemine Genetik Algoritma Metodu Uygulanırken Kısıtların Gerçekleştirilmesi.....	37
6.3.1	Dallı Yapının Gerçekleştirilmesi.....	37
6.3.2	Yüklerin Enerjilenmesi.....	39
6.4	Popülasyondaki Üyelerin Uygunluk Değerlerinin Bulunması	41
6.5	Genetik Algoritma Operatörlerinin Uygulanması.....	42
6.5.1	Üreme Operatörünün Uygulanması.....	42
6.5.2	Çaprazlama Operatörünün Uygulanması	45
6.5.3	Mutasyon Operatörünün Uygulanması	47
6.6	Enerji Dağıtım Sistem Verilerinin Genetik Algoritmada Değerlendirilmesi ve Sonuçların Elde Edilmesi	49
6.6.1	Elde Edilen Sonuçlar ve Değerlendirme	51
6.6.1.1	Çaprazlama ve Mutasyon Oranlarının Çözümüne Etkileri	55
6.6.1.2	Sonuçların Literatürdeki Çalışmalarla Karşılaştırılması	59
6.7	Sonuçlar ve Öneriler.....	60
	KAYNAKLAR.....	62
	EKLER.....	66
Ek 1	Örnek enerji dağıtım sistemine yeniden yapılandırma probleminin genetik algoritma yöntemi ile uygulandığı C++ programlama dilinde yazılan program... 67	67
Ek 2	İlk popülasyonda bile uygunluk değerlerinin birbirine ne kadar yakın olduğunu gösteren çizelge	81
Ek 3	Tek noktalı çaprazlamanın sakıncalı yanlarının şekil üzerinde görsel olarak gerçekleştirilmesi.....	82
Ek 4	MATPOWER optimizasyon programının yük akış analizi yapan “opf” dosyası . 83	83
Ek 5	MATPOWER paket programında yük akış dosyası “opf”yi çalıştıran “runopf” dosyası	87
Ek 6	MATPOWER programında yük akış analizi yapılmak amacıyla enerji dağıtım sistem verilerinin girildiği dosya.....	88
	ÖZGEÇMİŞ	92

SİMGE LİSTESİ

D_i	i hattının konumunu belirten (açık/kapalı) konum değişkeni
f	popülasyondaki her bir üyenin uygunluk değeri
I_i	i hattından çekilen akım
I_{Lwj}	j barası talep akımı aktif bileşeni
I_{Ldwj}	j barası talep akımı reaktif bileşeni
I_{Lj}	j barası talep akımı
P_i	i hattından çekilen aktif güç
P_k	Bir hat üzerindeki aktif güç kaybı
P_{lj}	j barasındaki aktif talep güç
Q_{lj}	j barasındaki reaktif talep güç
Q_i	i hattından çekilen reaktif güç
R_i	i hattının omik direnci
S_{Lj}	j barası talep görünen gücü
N_k	Toplam fider bölümleri sayısı
TP_k	Enerji dağıtım sisteminin toplam aktif güç kaybı
V_j	j barasının gerilimi
X_i	i hattının endüktif reaktansı
φ_j	j barası güç faktörü açısı

KISALTIMA LİSTESİ

AL	Yapay hayat
BM	Brownian Motion yöntemi
DISTOP	Enerji dağıtım şebekesi optimizasyon programı
MRSH	Multiple-Restart Stochastic Hill-Climbing
SA	Simulated Annealing



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Tek tepe noktalı uzay ve çok tepe noktalı uzay	9
Şekil 2.2 Pratikte karşılaşılan bir amaç fonksiyonu örneği.....	10
Şekil 2.3 Farklı arama yöntemlerinin etkinlikleri	12
Şekil 2.4 $Y=X^2$ amaç fonksiyonu ve bir kapalı kutu örneği.....	12
Şekil 2.5 Temel genetik algoritma döngüsü.....	14
Şekil 2.6 Üreme operatörü için rulet çarkı	15
Şekil 2.7 Çaprazlama işleminin şematik gösterilimi.....	16
Şekil 3.1 Bir hat ve bir yükten oluşan basit bir enerji sistemi.....	20
Şekil 3.2 Bir enerji dağıtım fideri tek hat diyagramı ve güç akış şeması.....	21
Şekil 3.3 Bir enerji dağıtım sisteminde akımların dağılımı	22
Şekil 4.1 Uygulamada kullanılacak örnek enerji dağıtım sistemi.....	25
Şekil 5.1 Tipik bir dallı şebeke.....	30
Şekil 6.1 Genetik Algoritmanın genel akış diyagramı	36
Şekil 6.2 Enerji dağıtım sisteminin kapalı şebeke formasyonu	37
Şekil 6.3 Popülasyondaki her üyeye eşdeğer enerji dağıtım şebekesinin akım dağılımını veren akış diyagramı.....	40
Şekil 6.4 Üyelerin sonraki nesle kopyalanmalarına ilişkin rulet çarkı.....	44
Şekil 6.5 Grupsal çaprazlama yapmak amacıyla rasgele seçilen iki üye ve sembolize ettikleri enerji dağıtım sistemleri	46
Şekil 6.6 A ve B üyelerinin çaprazlanması sonucu oluşan yeni C ve D bireyleri.....	47
Şekil 6.7 Uygulamada mutasyon işleminin gerçekleştirilmesine bir örnek.....	49
Şekil 6.8 Program ekranından aynen alınan optimum çözüm.....	52
Şekil 6.9 İlk bulunan en iyi çözümü başlangıç noktası kabul eden ikinci uygulama algoritmasının en iyi çözümü	54
Şekil 6.10 Sadece çaprazlama işlemi uygulanan algoritma	56
Şekil 6.11 Sadece mutasyon işlemi uygulanan algoritma.....	57
Şekil 6.12 0.10 çaprazlama ve 0.10 mutasyon oranının uygulandığı algoritmaların çözüme yakınsama dağılımı ve ortalama yakınsama eğrisi	58
Şekil 6.13 0.40 çaprazlama ve 0.40 mutasyon oranının uygulandığı algoritmaların çözüme yakınsama dağılımı ve ortalama yakınsama eğrisi	59

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Diziler, amaç fonksiyon değerleri ve kopyalanma yüzdeleri.....	15
Çizelge 2.2.a Genetik Algoritmalar.....	18
Çizelge 2.2.b Genetik Algoritmalar (devam).....	18
Çizelge 4.1 Örnek enerji dağıtım sistemi hat verileri.....	26
Çizelge 4.2 Örnek enerji dağıtım sisteminin bara talep yük verileri.....	27
Çizelge 5.1 Şekil 4.1'deki sisteme eşdeğer birey Genetik Algoritma zinciri	35
Çizelge 6.1 Üreme işlemine tabi tutulacak örnek bir popülasyon.....	43
Çizelge 6.2 Popülasyon üyelerinin bir sonraki nesilde bulunma ihtimalleri.....	44
Çizelge 6.3 Başlangıç enerji dağıtım şebekesine ilişkin yük akış analizi sonuçları	50
Çizelge 6.4 Başlangıç enerji dağıtım şebekesi sabit yük akımları	51
Çizelge 6.5 Bulunan en iyi çözüme ait yük akış analizi sonuçları.....	53
Çizelge 6.6 Bulunan en iyi çözüme ait akım değerleri	54
Çizelge 6.7 Optimum çözümü veren enerji dağıtım şebekesinde açık bırakılan anahtarlar ve toplam kayıp değeri	55
Çizelge 6.8 Literatürde aynı problemi ele almış olan çalışmalarla bir karşılaştırma çizelgesi	60

ÖNSÖZ

Doktora tez çalışmalarım boyunca maddi-manevi yardımlarını esirgemeyen başta Sayın Hocam Doç. Dr. F. Okan Pekiner olmak üzere tüm hocalarım ve kıymetli arkadaşlarıma teşekkürlerimi sunarım.



ÖZET

Enerji dağıtım sistemlerinde, üzerinde uzun yıllardır çalışılan başlıca konulardan biri, hattın aktif güç kayıplarının minimum seviyelere indirilmesi olmuştur. Bu çalışmaların geçmişi teorik olarak elektriğin icat edilmesine kadar dayanmaktadır. Son zamanlarda elektronik endüstrisindeki ve telekomünikasyon teknolojilerindeki gelişmeler ışığında enerji dağıtım sistemlerinin kontrol ve kumandası kolaylaşmış, gerçek zamanlı uzaktan kontrol sistemleri geliştirilmiştir. Mühendisler, kumanda ettikleri sistemleri daha fazla tanıma ve zorunlu haller dışında da sisteme müdahale edebilme olanağına sahip olmuşlardır. Enerji dağıtım sistemlerinde genellikle kayıpların azaltılması, yük dengelemesi veya arızalı bir bölgenin ayrılması amaçları ile yapılan yeniden yapılandırma çalışmaları da bu çerçevede gelişim göstermiştir.

Bu çalışmada, dallı bir enerji dağıtım sistemi ele alınarak sistemdeki hatlara ilişkin aktif güç kayıplarının minimum yapılması amacıyla sistemin yeniden yapılandırılması öngörülmektedir. En iyi ya da en iyiye yakın çözümleri verecek sistem yapılarının bulunması istenmektedir. Bu amaçla, C++ programlama dilinde yazılan bir programla genetik algoritma metodu, yeniden yapılandırma probleminin çözümünü arayan işlevsel bir araç olarak kullanılmıştır. Karmaşık matematiksel iterasyonları en aza indirgeyen genetik algoritma metodunun geçerliliği ve diğer yöntemlere olan üstünlükleri ortaya konulmuştur.

Genetik algoritmanın her seferinde ele alınan örnek enerji dağıtım sistemini çözüme yakınsattığı belirlenerek, yeniden yapılandırma probleminin çözümünde genetik algoritmaların başarılı olduğu kanıtlanmıştır.

Anahtar kelimeler: Enerji Dağıtım Sistemi, Yeniden Yapılandırma Problemi, Rezistif Hat Kayıpları, Genetik Algoritmalar.

ABSTRACT

One of the primary studies on energy distribution system is to minimize the resistive line losses. The theory of loss minimization problem has been carried out from the very early years of electricity. Due to recent developments in electronic industry and telecommunication technologies, control and command of energy distribution systems have become powerful. Accordingly, real time remote control systems have developed. Hence, these developments allow the engineers both to gain more knowledge about energy distribution command system and to interfere to the energy distribution system under any condition. In this respect, reconfiguration studies on energy distribution systems have been performed to provide loss minimization, load balancing and system restoration.

In this thesis, a radial distribution system is taken into consideration for system reconfiguration to provide resistive line loss minimization. Ideally, the best or nearly best solution is generally desired. For this purpose, a C++ based genetic algorithm programme has been developed and used as a tool to search reconfiguration problem solution. Validity of genetic algorithm that minimizes the complex mathematical iterations and its merits compared to other methods are presented.

In the application, presented genetic algorithm has converged to optimum solution for each trial. Hence, it is proved that genetic algorithms can be used in solving reconfiguration problem successfully.

Keywords: Energy Distribution Systems, Reconfiguration Problem, Resistive Line Losses, Genetic Algorithms.

1. GİRİŞ

1.1 Tezin Amacı ve Konuya Ait Çözüm Yaklaşımı

Enerji dağıtım sistemlerinin yeniden yapılandırılması (reconfiguration), bölümlendirme ve bağlantı anahtarlarının konumları değiştirilerek (açık-kapalı veya kapalı-açık) dağıtım fiderlerinin farklı topolojik yapılarının incelenmesi olarak tanımlanmaktadır (Chen vd., 1993; Roytelman vd., 1996).

Bu çalışmada, dallı şebeke yapısı muhafaza edilirken, aktif güç kayıpları minimum olacak şekilde enerji dağıtım sistemlerinin yeniden yapılandırılması hedeflenmiştir.

Bu amacı gerçekleştirmede yöntem olarak Genetik algoritma (GA) kullanılmıştır. Böylelikle, daha önceki çalışmalarda sunulan örneklenmiş verilere (IEEE referans çalışmaları) dayanılarak, enerji dağıtım sisteminin aktif güç kayıpları ile modellenmesi ve evrimsel yöntemler kullanılarak optimizasyonu sağlanmıştır.

Genetik algoritmalar, son yıllarda kompleks optimizasyon problemlerinin büyük bir kısmını çözmeye yeterliliği ve basitliğinden dolayı başarıyla kullanılmıştır (Man vd., 1996; Chaiyaratana ve Zalzala, 1997). Evrimsel temelli tekniklerin başarısının ana sebebi, teknikler arasındaki etkili bilgi alışverişi ve özel bir problem için aday çözümlerin bir popülasyonu ile paralel olarak çok yönlü araştırma yapabilmeleridir.

Çalışmada, enerji dağıtım sisteminin yeniden yapılandırılması probleminin optimum çözümü için C++ dilinde özel olarak yazılan genetik algoritma programı görsel olarak uygulanmıştır.

1.2 Konunun Tanıtılması

Günümüzde dallı yapıda tesis edilen enerji dağıtım sistemlerinde yeniden yapılandırma; değişen şartların algılanması (güzergahlarda, yüklerde ve yüklenme şekillerindeki değişimler vs.) ve günümüz teknolojilerinin uygulanması kapsamında gerçek zamanlı bir kontrol aracı olarak kullanılmasının yanı sıra bir planlama aracı olarak da kullanılabilir (Fawzi vd., 1983; Rosado vd., 1991; Tang 1996).

Değişik anahtarlama işlemleri ile dağıtım fiderlerinin dallı yapısı muhafaza edilerek yüklerin bir fiderden başka bir fidere transfer edilmesi tüm sistemin işletme koşullarını önemli ölçüde geliştirebilir (Peponis vd., 1996).

Dağıtım sistemindeki her bir fider ayrı ayrı ticari, konut ve endüstriyel tipte yüklerden

oluşabileceği gibi, bunların farklı kombinasyonlarından da meydana gelebilir ve bu yük tiplerinin günlük yük değişimleri birbirine benzemez. Ayrıca dağıtım transformatörlerinde, fiderlerde ya da bir fider parçasında pik yüklerin oluşumu da farklı zamanlarda meydana gelir.

Sistemin yeniden yapılandırılması aşırı yüklü fiderlerden ya da transformatörlerden daha az yüke sahip fiderlere ya da transformatörlere yüklerin transfer edilmesine izin verir. Bu tür transferler, sadece anahtarlanan fiderlerin yük seviyelerinin değiştirilmesi anlamında değil, aynı zamanda fiderler üzerindeki gerilim profilinin geliştirilmesinde ve tüm sistemdeki güç kayıplarının azaltılmasında da etkindir (Aoki vd., 1987).

Mikro ve mini bilgisayarlardaki büyük gelişme, mikroişlemciler ve telekomünikasyon teknolojilerindeki gelişmeler ve otomatik kontrol ve uzaktan kumanda araçlarındaki gelişmeler elektrik enerji sistemlerinin, özellikle de enerji dağıtım sistemlerinin otomasyonu alanında gelişmiş kontrol imkanları sunmaktadır (Lyons vd., 1981).

Gerçek zamanlı kontrol ve kumanda enerji dağıtım şebekelerinde, özellikle işletme boyutunda büyük faydalar sağlamaktadır. Enerji dağıtım sistemlerinde revizyonu zorunlu hale getiren arıza durumları da dikkate alındığında yeniden yapılandırma probleminin kapsamı genişlemektedir (Lin vd., 1998).

Kompleks mühendislik problemlerinin çözümünde optimizasyonun önemi rasgele arama algoritması ve tepe-tırmanma (hill climbing) temelli algoritmalar gibi birçok arama algoritmalarının geliştirilmesinde büyük ölçüde yardımcı olmuştur. Ancak, tepe-tırmanma gibi hesap temelli yöntemler arama uzayının süreksiz olduğu optimizasyon problemlerine uygulandığında önemli ölçüde kusurlara sahip olmaktadır. Bu teknik, eğitim bağımlı bir yöntem olduğundan yerel optimuma doğru gitme eğilimindedir.

Tepe tırmanma tekniklerinin mahsurlarının bir kısmını çözmek için farklı arama iterasyonlarında başlangıç noktasının rasgele seçimini kapsayan alternatif teknikler ileri sürülmüştür.

Bu son tekniklerden biri "Multiple-Restart Stochastic Hill-Climbing (MRSH)" (Baluja, 1995; Greene, 1997) olarak adlandırılmıştır. Bu yöntemde, bir ikili çözüm vektörü algoritmanın başında rasgele olarak üretilir, daha sonra onun bitlerinden rasgele birinin her iterasyon döngüsünde durumu değiştirilerek işlem yapılır. Bir iyileşmeye sebep olmayan tüm durum değişimlerinin bit durumları bir listede saklanır ve daha iyi bir çözüm bulunana kadar kullanılmaz. Eğer listenin büyüklüğü çözüm kodlamasının büyüklüğüne eşit olursa tek bitli

durum deęişimi çözümde iyileşmeye neden olmamıştır ve MRSH boş bir liste ile rasgele bir yerden yeniden başlatılır.

Hesap temelli metotların bazı yetersizliklerini çözmeyi amaçlayan dięer optimizasyon teknięi “Simulated Annealing”dir (Rutenbar, 1989). Hesap temelli yöntemlerin yapısından dolayı türev problemlerinin çözümü esnasında herhangi bir anda bir potansiyel çözüm göz önüne alınır ve arama işlemi süresi boyunca bu bilgiler saklanmaz.

Yapay zekayla sıkı bir şekilde ilişkili olan yeni bir araştırma alanı “Yapay Hayat” (Artificial Life (AL)) olarak adlandırılmaktadır. Yapay zeka “zeki” davranışıyla tabir edilen üretimle ilişkili iken, Yapay Hayat doğal yaşam sistemlerini taklit eden davranış üretimiyle ilgilidir. Böylece “Yapay Hayat”, her ne kadar bizim gerçek dünyamızla ilişkili olmasa da, evrimsel ilişkileri benzetmek ve bu benzetimlerden ortaya çıkan durumlarla çalışarak suni yaşam gerçeklemede bilgisayar kullanımını içermektedir (Langton, 1995). Biyolojik orijinaller karmaşık olsa da bunlara ilişkin bilgisayar benzetimlerinin karmaşık olmadığı görülmektedir. Gerçekten, karmaşık sistemin yapay davranışı çok basit üreticilerle üretilebilir. Biyolojik evrim ve doğal seleksiyon prensipleri teknięe dayalı yapay evrim gelişimine yol açmıştır. Bunlar ise; Genetik Algoritmalar (Holland, 1975; Goldberg vd., 1989), Evrimsel Stratejiler (Back, 1996), ve Genetik programlamalardır (Koza, 1992).

1.3 Literatürde Yeniden Yapılandırma Problemi

Bilgisayar teknolojilerinde ve telekominikasyon alanındaki gelişmeler özellikle dağıtım sistemi otomasyonu alanında yapılan atılımların önünü açmış, mevcut veri kazanım sistemlerini kullanan küçük ölçekli prototip dağıtım otomasyon sistemleri geliştirilmiş ve test edilmiştir (Lyons vd., 1981; Purucker vd., 1985).

Bilgisayar kontrollü dağıtım otomasyonu için bir dağıtım sistemi tanımı McCall ve arkadaşları tarafından verilmiştir (McCall vd., 1983).

Kayıpların azaltılması anlamında enerji dağıtım sistemleri üzerine uygulanan ilk yeniden yapılandırma çalışmalarından bir tanesi Merlin ve arkadaşları tarafından sunulmuştur (Merlin vd., 1975).

Bir dięer çalışmada, fiderin yeniden yapılandırması için iki farklı araştırma algoritması sunulmuştur (Ross vd., 1980). Kısıt ihlallerinin derecesini ölçmek amacıyla bazı indisler geliştirilmiş ve bunlar işletme noktası çok uygun olmadığında, uygun bir işletme noktası bulmak amacıyla kullanılmıştır. Ayrıca bu indisler, güç kaybının azaltılmasında çözümün

optimum olup olmadığını kontrol etmek amacıyla da kullanılmıştır. Buna ek olarak, arızalardan sonra servisin yeniden sağlanması konusunda detaylı bir reçete de sunulmuştur.

A. Merlin ve H. Back'in çalışmalarını temel alan bir uygulama Shirmohammadi ve arkadaşları tarafından ele alındı (Shirmohammadi vd., 1989). Bu çalışmada daha önceki çalışmada yapılan bazı kabuller ortadan kaldırılarak (örneğin; yüklerin tamamen aktif olması, gerilim açısının ihmal edilmesi gibi) bir takım sakıncalar giderilmiştir. Aynı metot bir enerji dağıtım şebekesi optimizasyon programına (DISTOP) uygulanarak, daha kısa hesaplama süreleri ve daha iyi sonuçlar elde edilmiştir.

Adı geçen çalışmalar ışığında enerji dağıtım sisteminin dallı yapısını değiştirerek sistemdeki aktif güç kayıplarını azaltmayı ve sistemi daha verimli çalıştırmayı amaçlayan yeniden yapılandırma faaliyetlerine hız verilmiştir.

Böylesi çalışmaların birinde 16 bara, 3 adet fider, 3 adet bağlantı anahtarından oluşan bir enerji dağıtım sisteminin bağlantı anahtarlarının kapatılarak yerine bölümlendirme anahtarlarının açılması mantığına dayanan bir algoritma geliştirilmiştir (Civanlar vd., 1988). Bir anahtar kapatıldığında yerine aynı kapalı çevreden başka bir anahtarın açılması gereken (dallı sistemin muhafaza edilmesi amacıyla) yöntemde, bu işlem sırasıyla tekrarlanmakta, aynı anda birden fazla anahtarlama işlemi yapılamamaktadır.

Algoritmanın esasını anahtarlar üzerinde düşen gerilimlerin farkı teşkil etmektedir. Öncelikle başlangıçta açık olan bağlantı anahtarlarından birisi kapatılarak, bu anahtarın bağlı bulunduğu baralarda anahtar açıkken oluşan gerilimlerden daha büyük değerli gerilime sahip bara tarafındaki ilk anahtarın açılması, çözüme ulaşmayı kolaylaştıracak metot olarak sunulmaktadır. Bazı basitleştirici kabullerle elde edilen bir formülasyon ile, yük transferi sonucu kayıpta oluşan azalma hesaplanabilmektedir. Bu kayıpların azaltılması işlemi en iyi çözüme ulaşıncaya kadar devam ettirilmektedir.

Mesut E. Baran ve Felix F. Wu, enerji dağıtım sistemi yeniden yapılandırma problemi üzerine yaptıkları çalışmada hem kayıpların azaltılması, hem de yük dengelemesi konularını ele almışlardır (Baran vd., 1989). S. Civanlar tarafından önerilen çözüm yaklaşımını devam ettirerek, baralar arasında yük transferi yapıldıktan sonra sistemdeki güç akışına yaklaştıracak, doğruluk derecesi değişen iki farklı metot ortaya koymuşlardır. Bu metotlarda, özellikle dallı enerji dağıtım fiderleri için geliştirilmiş bir grup güç akış eşitliği kullanılmıştır. Bu yaklaşık güç akış metotları ile hem kayıplardaki azalma, hem de sistemin yük dengesi kestirilmeye çalışılmıştır.

Bir başka çalışmada enerji dağıtım sisteminde optimal şebeke yeniden yapılandırma problemi, ikinci dereceden maliyet hesaplamalı bir konaklamalı ulaşım (transshipment) problemi olarak formüle edilmiştir (Slamocanin, 1990). Matris işlemlerine olan ihtiyaç tamamen ortadan kaldırılarak işlemler doğrudan enerji dağıtım sistemi şeması üzerinde gerçekleştirilmiştir.

Güç akış bazlı yeni bir bulgusal algoritmanın sunulduğu diğer bir çalışma da optimum akış şeması düşüncesine dayanmaktadır (Goswami vd., 1992). Bu çalışmada, dallı şebeke üzerine uygulanmış olan bir akış şeması ile işleme başlanmakta, normalde açık olan anahtarlar üzerindeki gerilim düşümleri kullanılmaktadır. Üzerindeki gerilim düşümü en düşük olan, en büyük olan ya da rasgele seçilen bir anahtar kapatılarak, dallı şebeke bir tek gözlü kapalı şebeke haline dönüştürülmektedir.

Üç farklı metotla uygulanan işlemde sadece bir tek kapalı çevre oluşturulmakta ve tekrar aynı çevreden aynı ya da farklı bir anahtar üzerinden geçen akışın minimum olması şartına bağlı olarak açılmaktadır. Bu işlem benzer şekilde tekrarlanarak minimum kayıplı optimum/optimuma yakın çözümler elde edilmektedir. Farklı metotların uygulandığı bu çalışmada, tatmin edici bir şekilde aynı sonuç elde edilmiştir.

Goswami ve Basu'nun çalışmalarına paralel olarak, tek bir kapalı çevrenin optimizasyonu hakkında sayısal analiz yolu ile analitik bir tanımlama ve sistematik bir anlayış ortaya konulmuştur (Fan vd., 1996). Şebeke yeniden yapılandırma problemi minimum kayıpların sağlanması amacıyla sayısal analiz yöntemi kullanılarak ikinci dereceden amaç fonksiyonlu bir tam sayılı optimizasyon problemi olarak formüle edilmiştir.

Bu lineer olmayan problem lineerleştirilirse, yaklaşık olarak bir tam sayılı lineer programlama problemi olarak sunulabilir. Bu düşünce, yazarları normalde lineer programlama problemlerinin çözümünde kullanılan "Simplex Metodu"nun kullanılması düşüncesine, buradan hareketle de bir tek kapalı çevre optimizasyon yaklaşımının direk olarak çıkartılması anlayışına ulaştırmıştır.

Bu çalışmada ayrıca yük değişimini ve yük akışını hesaplamak amacıyla kolay ve etkin bir şema sunulmuştur. Geliştirilen yöntem ile şebekenin ilk konfigürasyonundan optimum konfigürasyonuna ulaşmak için gerçekleştirilecek olan anahtarlama işlemlerinin minimum seviyede tutulmasına çalışılmıştır.

Song ve arkadaşları, genetik algoritma kullanılan programla tekniği ile 3 fiderden oluşan bir dağıtım sisteminde sınırlı sayıdaki anahtarlama işlemi ile yeniden yapılandırma problemine

cevap aramış ve mutasyon işleminin hangi oranda yapılacağını tespit etmek amacıyla bir fuzzy kontrolör kullanmışlardır (Song vd., 1997).

Dallı şebeke yerine ağ şebekenin dikkate alındığı bir başka çalışmada, her bir kapalı gözde daha fazla anahtarlama işlemi dikkate alınarak uygun anahtarlama kombinasyonunu daha da azaltacağına kesin olarak inanılan bir algoritma geliştirilmiştir (Lin vd., 1998). Bütün bağlantı anahtarlarının kapatılmasıyla başlanılan algoritmada üç adet anahtarlama indisi tanımlanmıştır. Amaç her ne kadar kayıpların azaltılması olsa da bu indisler, servisin tekrar sağlanması amacıyla da kullanılabilir.

Kashem ve arkadaşları tarafından, bir bağlantı anahtarı kapatılarak yerine bir bölümlendirme anahtarının açılması sonucu gerçekleştirilen dal-değişim yöntemi ile yine minimum anahtarlama işlemi gerçekleştirilerek optimum çözüme ulaşmak hedeflenmiştir (Kashem vd., 2000). Bunun için S. Civanlar'ın kayıpların azaltılması amacıyla elde ettiği basitleştirilmiş formülasyon ile Mesut E. Baran ve arkadaşlarının bir tek kapalı çevrenin analizi için çıkarttıkları formülasyonların geliştirilmesiyle elde edilen bir formülden yararlanılmıştır.

Böylece, tek bir kapalı çevredeki aktif güç azalmasını tespit edecek bu formülün tüm muhtemel kapalı çevrelere uygulanması sonucu kayıplarda azalma sağlayacak kapalı çevreler belirlenmektedir. Çözüme aday olan kapalı çevrelerdeki bağlantı anahtarlarının bağlı bulunduğu baralardan gerilimi daha düşük olan bara (ya da gerilim düşümü daha yüksek olan bara) tarafındaki anahtarlar üzerinde yapılan hesaplamalar sonucunda, bu kapalı çevreden açılması gereken anahtar belirlenmektedir. Elde edilen sonuçların karşılaştırmalı analizinden de anlaşıldığı üzere, yöntem başarılı sonuçlar vermiştir.

Daha sonra M.A. Kashem ve arkadaşları, konu ile ilgili olarak bir çalışma daha yayınlamıştır. Bu çalışmada ise, enerji dağıtım sisteminde kayıpların azaltılması problemine geometrik bir yaklaşım geliştirmiştir (Kashem vd., 2001).

Tez çalışmasının ikinci bölümünde, yeniden yapılandırma problemlerinin çözümünde kullanılan genetik algoritmalar hakkında bilgi verilmiştir. Konunun daha iyi anlaşılabilmesi amacıyla genetik algoritma örnek probleminin çözümü adım adım açıklanmıştır.

Üçüncü bölümde, enerji dağıtım sistemlerinin yapısal özellikleri incelenerek yeniden yapılandırma problemine ilişkin matematiksel modeller çıkartılmıştır. Dallı enerji dağıtım sisteminde hattın aktif güç kayıplarını minimum yapan hedef fonksiyon elde edilmiştir.

Dördüncü bölümde, genetik algoritmanın uygulanacağı örnek bir enerji dağıtım sistemi

dikkate alınarak, gerekli olan hat ve yük verileri sunulmuştur.

Beşinci bölümde, enerji dağıtım sistemlerinin özellikleri ve işletme şartları hakkında bilgi verilerek, bu tezde ele alınan örnek enerji dağıtım sisteminin kısıtları belirlenmiştir.

Altıncı bölümde, genetik algoritma örnek enerji dağıtım sistemine uygulanmış, algoritmanın tüm basamakları detaylı olarak anlatılarak örnekleme yapılmıştır.

C++ programlama dili ile yazılan program örnek enerji dağıtım sistemine uygulanarak, elde edilen sonuçların daha önceki çalışmaların sonuçları ile karşılaştırılması yapılmış, çıkarılan sonuçlar ve genel değerlendirmeler de bu bölümün sonunda verilmiştir.



2. GENETİK ALGORİTMALARIN TEORİ VE UYGULAMALARI

Bu bölümde, genetik algoritmalara ait teori ve uygulamalar ve çalışmadaki uygulama işlemleri ele alınarak, bunların diğer en iyi arama yöntemleri ile karşılaştırılması yapılacaktır. Genetik algoritmanın çalışması, bir örnek üzerinde açıklanacaktır.

2.1 Genetik Algoritmalar

Genetik algoritmalar, doğal seçme ve doğal genetik kurallardan esinlenerek geliştirilmiştir.

Doğal seçme, doğa koşullarına en fazla uyum sağlamış olan canlı neslinin devam etmesi, uyum sağlayamamış olan türlerin ise elenmesi görüşüne dayanır. Bu görüşe göre, canlıların nesilden nesile genleri aktarılırken, bu genler de doğal genetik kurallara göre başka genlerle eşleşir, değişir ve yeni genleri oluşturur.

Genetik algoritmalar ile tabiatta gözlenen genlerin nesilden nesile aktarılması ve değişime uğraması olayları birleştirilerek, en iyi çözüm aranır. Bir önceki neslin en uygun elemanlarının parçaları ve bitleri kullanılarak yeni nesillerin üyeleri oluşturulur. Bu üyelerin uygunlukları test edilerek, en iyilerinden hareketle yeniden başka elemanlar türetilir. Genetik algoritmalarda rasgeleliğin rolü olmasına karşın, olasılık bazlı bir arama metodudur.

Genetik algoritmalar Michigan Üniversitesi'nden John Holland ve arkadaşları ile öğrencileri tarafından geliştirilmiştir (Holland, 1975). Araştırmalarda iki amaç hedeflenmiştir: Birincisi, canlılardaki adaptasyonun yapısını anlamak ve kabaca açıklamak; ikincisi, bu yapıdan hareketle, insan yapısı sistemlere uygulanabilecek algoritmayı bulmak. Bu yaklaşım, önemli gelişmelere ışık tutmuştur.

Genetik algoritmalarındaki araştırmaların temel ilgi noktası değişen ortam şartlarına uyum sağlayabilen gürbüz yapıyı oluşturabilmektir. İleri derecede adaptif sistemler oluşturulabilirse pahalıya mal olan yeniden tasarım problemleri azaltılabilir, hatta ortadan kaldırılabilir. Eldeki mevcut sistemler, adaptif bir kontrol ile değişen şartlarda yeniden kullanılabilir ve görevlerini daha iyi ve daha uzun zaman yerine getirebilir.

Genetik algoritmaların esasını temsil eden biyolojik sistemler, dikkatli bir inceleme ile araştırmalara yeni ufuklar açabilecek sıklara sahiptir. Bugün için en iyi adaptasyon doğada var olan adaptasyondur. Buna dayanan genetik algoritmalar da, Holland'ın 1975'te yayınladığı "Doğal ve Yapay Sistemlerde Adaptasyon" isimli çalışmasından bu yana, başka pek çok makalede ele alınmış ve etkinliği ispatlanmıştır.

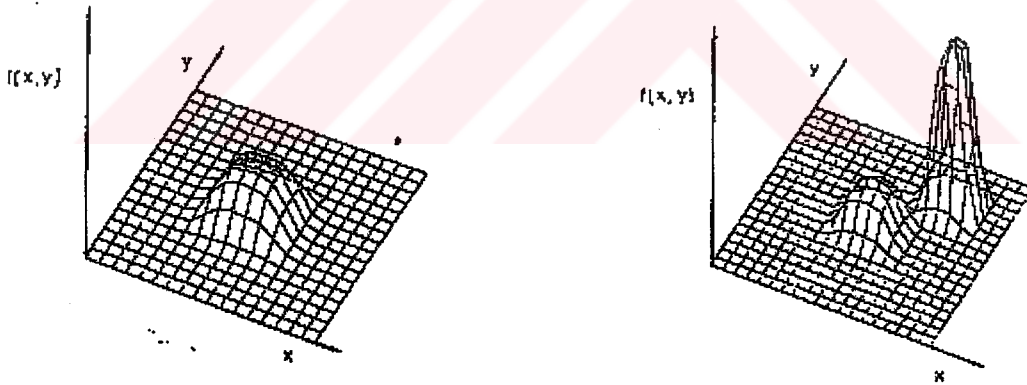
Genetik algoritmalar, basit hesaplamalar gerektirir; ancak işlemlerin basitliği, etkinliğini azaltmaz. Diğer pek çok en iyiyi arama yönteminde görülen süreklilik, türevlerin var olması, tek model olma gibi kabuller bu yöntem için geçerli değildir.

2.2 Geleneksel En İyiyi Arama Yöntemlerinin Dezavantajları

Geleneksel arama yöntemleri üç ana başlık altında toplanabilir: Hesaba dayanan, nümerik ve olasılıklı yöntemler.

Hesaba dayanan yöntemler, bugüne kadar yoğun bir biçimde incelenmiştir. Bunlar da kendi aralarında doğrudan ve dolaylı olmak üzere ikiye ayrılır.

Dolaylı metot, amaç fonksiyonunun gradyanının sifira eşit alınması sonucu çıkan lineer olmayan denklem takımının çözülmesi esasına dayanır. Bu metot, Şekil 2.1.'de tek tepe noktali yüzey üzerindeki yerel maksimumun n-boyutlu yüzeylere genellenmesi ile uygulanır. Arama, her yöndeki türevi sifir olan bu tepe veya çukur olması muhtemel noktaların civarında yapılır.



Şekil 2.1 Tek tepe noktali uzay ve çok tepe noktali uzay (Goldberg, 1989)

Diğer taraftan, doğrudan arama yönteminde, maksimum noktasının yeri hakkında bir tahminde bulunulur ve buna yakın herhangi bir noktanın gradyanı veya tersi yönde hareket edilerek arama sürdürülür. Buna literatürde tepe tırmanma (hill-climbing) adı verilir.

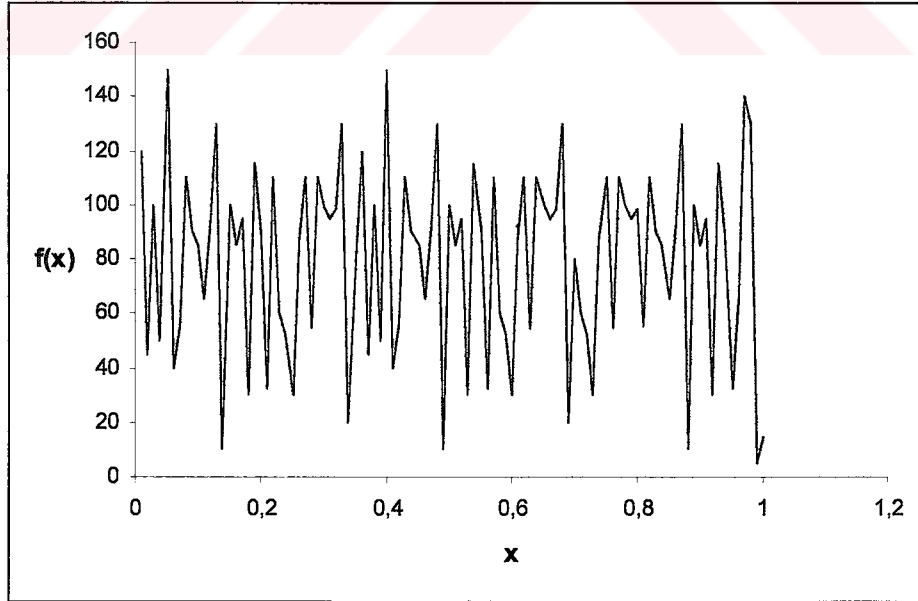
Yerel en iyi noktayı bulmak için, mümkün olan en dik eğimli yönde ilerlenir. Her ne kadar bu iki arama yöntemi pek çok çalışmada ele alınmışsa da, birkaç basit örnek, bunların zayıf yanlarını göstermeye yeter.

Bu yöntemlerin en önemli eksikliği, yerel bakış açısıyla yürütölmeleridir: Aradıkları en iyi nokta, hareket noktasının yakınındadır. Örneğin, Şekil 2.1.'de gösterilen tek tepe noktalı uzay, uzayın bir bölgesi olsun. Bu uzayın daha geniş bir görünüşü de, yine Şekil 2.1'de çok tepe noktalı uzay olarak verilmiş olsun.

Bu ikinci şekilden de görölebileceği gibi, alçak tepenin yanından aramaya başlanılırsa en iyi nokta olarak alçak tepenin üzerinde kalınacaktır. Oysa, bulunacak bu noktadan daha iyi bir nokta vardır ki, o da daha yüksek olan ikinci tepe noktasıdır.

Aramaya alçak tepenin yakınındaki bir noktadan başlanması, esas en iyi noktayı kaybettirecektir. Buna bir çözüm olarak, rasgele seçilmiş başka bir noktadan tekrar başlanabilir. Tek tepe noktalı fonksiyonlar hesaba dayalı arama yöntemleri için kolay olmasına rağmen çok tepe noktalı fonksiyonlarda ise hangi tepeye tırmanılacağı ayrı bir problemdir.

İkinci önemli eksiklik ise, amaç fonksiyonunun türevlerinin var olmasının gerekliliğidir. Teorik olarak hesabı kolay olan ikinci dereceden amaç fonksiyonlarının bulunabilmesine karşın, gerçek hayatta karşılaşılan problemler teoriden uzaktır. Parça parça süreksizlikler içerebileceği gibi, gürültülü ortamlardan elde edildikleri için, Şekil 2.2'de gösterildiği gibi, pek çok tepe noktası kapsayabilir. Bundan da anlaşılabilirceği gibi hesaba dayalı yöntemler



Şekil 2.2 Pratikte karşılaşılan bir amaç fonksiyonu örneği (Goldberg, 1989)

ancak sınırlı sayıda uygulamalar için kullanışlı olup, bütün uygulamalar için kullanışlı

değildir.

Nümerik yöntemlerin de üzerinde pek çok çalışmalar yapılmıştır. Bu tip arama yöntemlerinde temel düşünce, sınırlı bir arama uzayındaki noktaların amaç fonksiyon değerlerine bakılıp ona göre yeni noktaların bulunmasıdır. Ancak pratikte arama uzayının geniş olması ve bundan dolayı arama uzayındaki noktaların birer birer ele alınamaması, bu tip arama yöntemlerinin etkinliğini sınırlamaktadır.

Diğer arama yöntemlerinin bu zayıf yönlerinden dolayı, rasgele arama yöntemleri, giderek artan bir ilgiyle kullanılan araştırma konularından biri olmuştur. Bununla beraber, rasgele bir yönde tepe tırmanma veya rasgele noktalar seçerek bunların amaç fonksiyonlarını bulma gibi düşünceler rasgele arama yöntemlerinin dışında tutulmalıdır, çünkü etkin değildirler.

Olasılık, genetik algoritmalarda bu gibi rasgele arama yöntemlerinden farklı olarak, kodlanmış parametre uzayında işlem yapmada bir araç olarak kullanılır. Bir arama algoritmasında olasılığın kullanılması ilk önce şaşırtıcı gelebilir, ancak doğada bunun sayısız örneği mevcuttur.

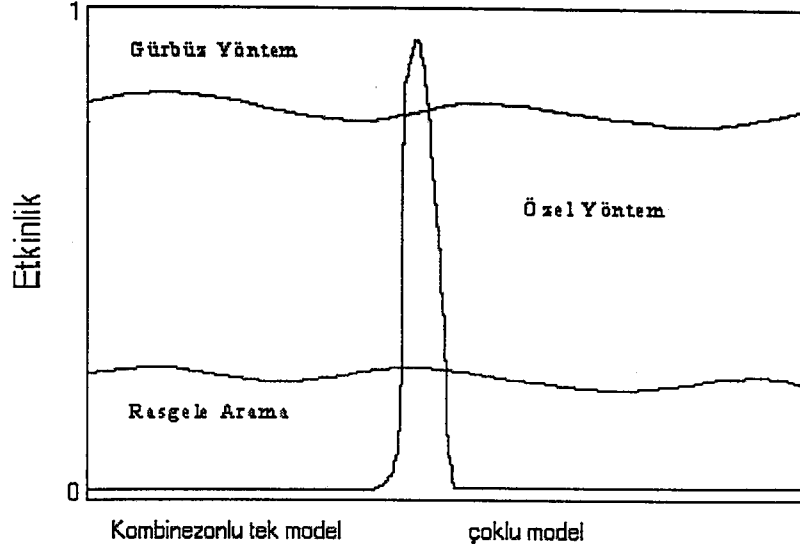
Genetik algoritmalar diğer global optimizasyon yöntemleri (Price, 1983) ile karşılaştırılırsa, bunların da olasılık hesaplarından yararlandığı ancak parametrelerin kodlanmadığı görülebilir.

Kushner'ın Brownian Motion yönteminde Gauss dağılımı kullanılmaktadır (Kushner, 1967).

Yakın zamanlarda ilgi çeken bir başka yöntem ise "Simulated Annealing" (SA) olarak adlandırılan yöntemdir. Davis'in 1987'de çıkan bir kitabında, bu yöntemle genetik algoritmalar arasındaki ilişki incelenmektedir. Bu arada rasgeleliğin kullanılmasının, belli bir mantıktan yoksun olarak arama yapmak demek olmadığı dikkate alınmalıdır.

Olasılığın kullanılmadığı global arama yöntemleri de vardır (örneğin ızgara yöntemi). Ancak bunlarda da işlem sayısının fazlalığı, n-boyutlu uzaylarda çözümü bulmayı zorlaştırmaktadır.

Şekil 2.3'de görüldüğü gibi, pek çok geleneksel arama yöntemi küçük bölgelerde iyi sonuç vermektedir. Nümerik veya rasgele arama yöntemleri bütün spektrum için yeterli etkinlikte değildir. Gürbüz yöntemler ise bütün spektrum için yeterli etkinliktedir.

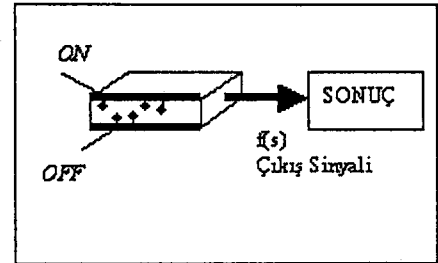
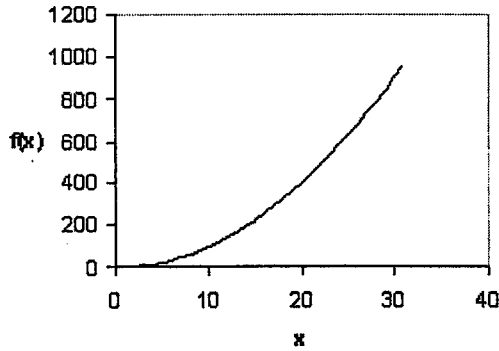


Şekil 2.3 Farklı arama yöntemlerinin etkinlikleri (Goldberg, 1989)

2.3 Genetik Algoritmalar ile Diğer Geleneksel Yöntemler Arasındaki Farklar

Genetik Algoritmaların temel özellikleri, aşağıdaki dört maddede toplanabilir:

1. GA parametrelerin kodlanmış halleriyle çalışır, parametrelerin kendileri ile değil.
2. GA tek bir nokta yerine noktalar topluluğundan hareket eder.
3. GA türev ve daha farklı bilgileri değil, sadece amaç fonksiyonu değerlerini kullanır.
4. GA deterministik değil, olasılık kurallarına dayanır.



Şekil 2.4 $Y=X^2$ amaç fonksiyonu ve bir kapalı kutu örneği

Genetik algoritmalarda parametreler sonlu uzunlukta diziler ile gösterilirler. Örnek olarak, Şekil 2.4'te gösterilen [0,31] aralığındaki optimizasyon problemi göz önüne alınırsa, ilk olarak x parametresi açık kapalı biçiminde iki farklı değer alan bir değişkenle kodlanır. Daha sonra açık-kapalı yerine “bir” ve “sıfır”lar kullanılır.

Kapalı kutu örneğinde beş adet giriş anahtarı ve bir adet f çıkış işareti bulunmaktadır. Amaç fonksiyonu matematiksel olarak, s belli bir anahtar kombinasyonu olmak üzere $f=f(s)$ (örneğin $f=x^2$) şeklinde yazılabilir.

Problemin amacı, bu beş anahtarın mümkün olan en yüksek f değerini elde edecek uygun kombinasyonu bulmaktır. Diğer arama yöntemleri anahtarların durumlarını ve bir durumdan diğerine nasıl geçileceğini araştırırken, genetik algoritmalarda önce anahtarların durumları kodlanır. Bu kodlama, beş bitlik sayı dizileriyle yapılabilir.

Tek bir noktanın ele alınıp, bu noktadan diğer bir noktaya belli bir kural kullanılarak geçilmesi, yanlış bir yerel en iyi noktaya yerleşmeye yol açabilir. GA, tek bir nokta yerine bir noktalar topluluğundan yararlandığı için, atılacak ikinci adım bu noktalar topluluğunu belirlemektir. Böylece global en iyi noktaya yakınsama şansı artacaktır.

GA'da nesil adı verilen bu noktalar topluluğundan, yeni bir nesil, yani yeni bir noktalar topluluğu elde edilir. Örnekte birey sayısı $n=4$ alınmıştır ki, bu sayı GA standartları için küçük bir sayıdır.

01101

11000

01000

10011

Bu dört noktadan hareketle GA kullanılarak yeni nesiller bulunur. Bir nokta topluluğu üzerinde paralel çalışmanın, gürbüzlüğü sağlama yolunda olumlu etkilerinin bulunacağı açıktır.

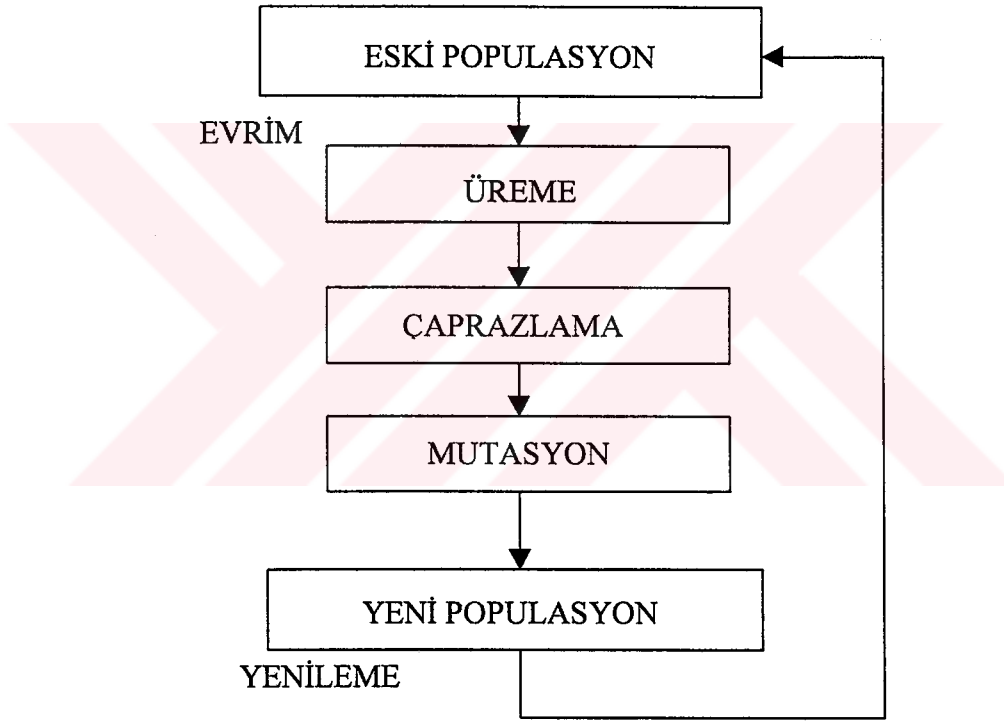
2.4 Basit Genetik Algoritma

Genetik algoritmaların mantığı, bit dizilerini kopyalamaya ve iki bit dizisinin bazı parçalarının yerlerini değiştirmeye dayanır. Uygulanmasının kolaylığı ve etkinliği, genetik algoritmalarla çözüm yaklaşımının en önemli iki üstünlüğüdür.

Bir önceki alt bölümde genetik algoritmaların, bir dizi topluluğu üzerinde nasıl işlem yaptığı kısaca ele alınmıştı. Kapalı kutu örneği tekrar dikkate alınırsa, ilk nesilde dört dizi bulunmaktaydı. Bu dört dizi de, rasgele olarak, bir paranın 20 kere atılması sonucu oluşturulmuş olsun. Bu aşamadan sonra, ilk nesillerden yeni nesilleri türetmede kullanılacak genetik operatörler tanıtılacaktır.

Pek çok uygulamada iyi sonuçlar veren basit genetik algortmada üç operatör bulunur. Bunlar; üreme, çaprazlama ve mutasyon operatörleridir.

Bu temel genetik algoritma Şekil 2.5'de gösterilmiştir (De Moura Oliveira, 1998; Goldberg, 1989).



Şekil 2.5 Temel genetik algoritma döngüsü (De Moura Oliveira, 1998)

Üreme işlemi sırasında, sayıları (noktaları) temsil eden bit dizileri, amaç fonksiyon değerlerine göre kopyalanır. Amaç fonksiyon değerleri ise, maksimize (veya minimize) edilecek fonksiyonun o noktadaki değeridir.

Maksimizasyon yaparken bir diziyi amaç fonksiyonuna göre kopyalamak demek, amaç fonksiyonunda yüksek bir uygunluk değeri alan diziyi sonraki nesillere daha yüksek bir olasılıkla kopyalamak demektir. Doğada bu operatör gerçek anlamda bulunmamakla beraber,

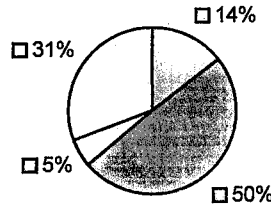
canlıların ortama uyum sağlayabilmeleri, üremeleri ve neslini devam ettirmeleri üreme operatörlerine denk olaylar olarak kabul edilmektedir. Buna göre ortama uyum sağlayanın (amaç fonksiyon değeri yüksek olanın) neslini devam ettirdiği (bir sonraki dizi topluluğuna daha yüksek bir olasılıkla kopyalanması) anlayışı uygulanmaktadır.

Bu operatör, genetik algoritmalarda pek çok değişik biçimde kullanılabilir. En basiti, bir rulet çarkı almak ve bunun üzerinde, noktalara uygunluk değerleri kadar bir dilim ayırmak olabilir. Yukarıdaki dizinin elemanlarının bu kapalı kutudan çıkan uygunluk değerleri Çizelge 2.1 ile gösterilmiştir. Uygunluk değerleri, Şekil 2.4'te verilmiş olan $f=x^2$ fonksiyonundan hareketle bulunmuştur.

Çizelge 2.1 Diziler, amaç fonksiyon değerleri ve kopyalanma yüzdeleri.

No.	Dizi	Uygunluğu	Yüzdesi
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Toplam		1170	100.0

Çizelge 2.1'den hareketle rulet çarkı oluşturulur. Bu çarkta her bir diziye, amaç fonksiyonun yüzdesiyle orantılı bir dilim ayrılır. Örnek çarkın bir görünüşü Şekil 2.6'da verilmiştir.



Şekil 2.6 Üreme operatörü için rulet çarkı

Üreme işlemini gerçekleştirmek için rulet çarkını dört kere döndürmek ve her seferinde okun

karşısına gelen diziyi yeni neslin bir üye olarak yazmak gerekir. Bu arada nesillerdeki üye veya eleman sayısının sabit kaldığı kabul edilir. Böylece yeni nesilde, örneğin birinci dizi 0.144, ikinci dizi 0.492 olasılıkla yer alacaktır.

Bu şekilde seçilen dört aday, işlem havuzunu oluşturur. İşlem havuzunda bu dizilere çaprazlama ve mutasyon işlemleri uygulanır.

Üreme işleminden sonra çaprazlama işlemi uygulanır. Bu işlem, iki kademedede gerçekleştirilir.

İlk olarak işlem havuzundaki dizilerden çaprazlama işlemine tabi tutulacak olanlar rasgele seçilir. İkinci olarak çaprazlama yapılmak üzere belirlenen bu dizilerin yine rasgele birer noktası aynı yerden olmak üzere seçilerek, o noktadan ileride kalan genler değiş tokuş edilir. Örneğin A_1 ve A_2 dizileri ele alınsın.

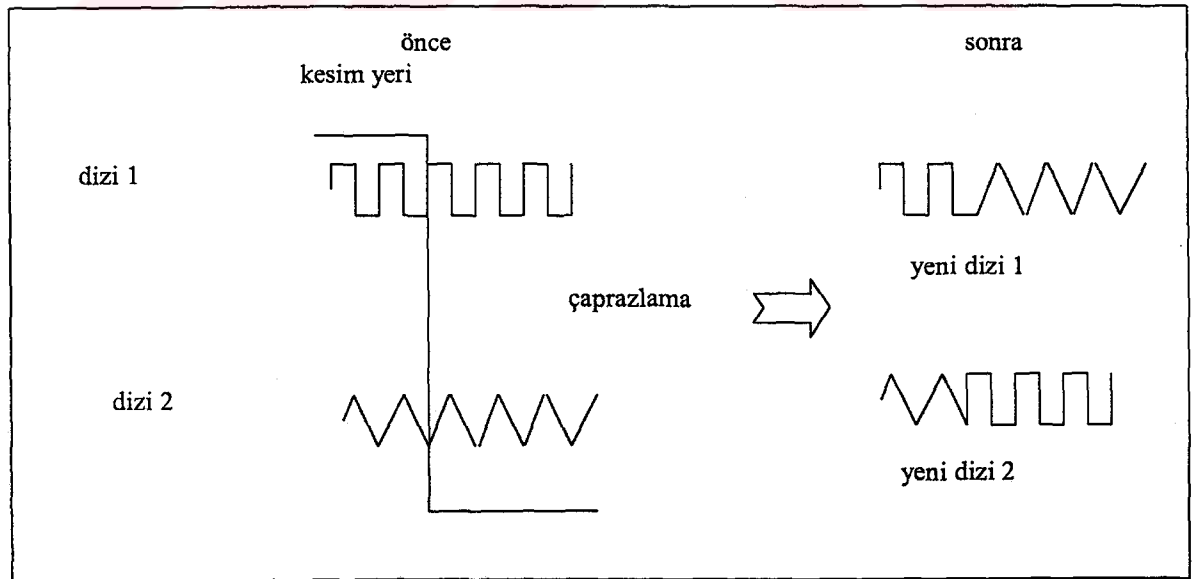
$$A_1 = 01110-1$$

$$A_2 = 11100-0$$

Bu dizilerde “-” işareti kesim yerini belirtmektedir. Kesim işlemi, bu işarettten sonraki bir bit için uygulanacaktır. Bu işlemden sonra elde edilecek yeni diziler:

$$A'_1 = 011100$$

$$A'_2 = 111001$$



Şekil 2.7 Çaprazlama işleminin şematik gösterimi

Üreme; rasgele sayı üretimi, kopyalama, çaprazlama işlemleri ve bazı alt dizilerin deęiş-

tokuşu gibi basit ancak etkili ve genetik algoritmalara işlev kazandıran unsurlardan oluşur. İyi olan dizinin daha yüksek olasılıkla kopyalanması ondan yeni nesillerde de yararlanılabilmeyi sağlarken, onun alt parçalarının başka dizilerin alt parçalarıyla değişimi ise daha iyiyi aramanın temel mantığıdır.

Çaprazlama operatörü başka şekillerde de uygulanabilir. Örneğin, tek bir kesim yeri belirleneceğine iki yer belirlenir ve sadece bu iki yerin arasında kalan bitler değiş-tokuş edilir. Bu uygulamanın iyiliği, çaprazlama operatörünün bütün bitler üzerinde homojen olarak uygulanabilmesidir.

Eğer diziler optimize edilmek istenen parametrelerden oluşturulmuşsa, yani her bir parametreyi oluşturan sayılar yan yana yazılarak bir dizi oluşturulmuşsa, o zaman parametreler arası çaprazlama uygulanabilir.

Bit bit çaprazlama yerine parametreler sekiz bit ile ifade ediliyorsa, byte byte çaprazlama uygulanabilir. Bu tarz çaprazlama işlemi robot kol yörünge optimizasyonunda kullanılmaktadır. Ancak yine de bit bit kesişimin hassasiyeti, seçiciliği daha yüksektir. Örneğin 10 üyeden oluşan bir toplumda her bir parametrenin alabileceği değer sayısı en fazla 20 olacağından ve farklı değerler alabilmesi sadece mutasyon ile sağlanacağından, yakınsama hızı düşecektir.

Bu çalışmada, bit üzerinde işlem yapan çaprazlama operatörü kullanılacaktır.

Mutasyon operatörü ise, dizilerin rasgele seçilmiş bitlerindeki “1” leri “0”, “0” ları “1” yapar ve genetik algoritmaların şans faktörüne en fazla bağlı olduğu işlemdir.

Eğer mutasyon olasılığı yüksek tutulursa, arama rasgele bir şekil alır ki, istenen bu değildir. Mutasyon uygulanmaz ise, o zaman sonuç yerel bir en iyiye takılabilir. Örneğin, en iyi netice için bir dizinin en son bitinin “1” olması gerekir ve o anda elde olan dizilerin hiç birinde son bit “1” olmaz ise en iyi sonuca ulaşmak mümkün olmaz. İşte bu son biti “0” dan “1” yapacak operatör, mutasyondur. Bu nedenle, her ne kadar aramaya rasgelelik kazandırsa da, mutasyon gereklidir. Ancak çaprazlama operatörünün olasılığına göre çok daha küçük bir olasılıkla dikkate alınır. Bu operatörün aynısı doğada da mevcuttur.

2.5 Genetik Algoritma Uygulamasına Bir Örnek

Bu kısımda basit genetik algoritma adım adım bir örnek üzerine uygulanacaktır (Goldberg, 1989). Amaç fonksiyonu olarak Şekil 2.4’te gösterilen $f(x)=x^2$ fonksiyonu kullanılacaktır.

Burada x deęişkenine 0 ile 31 arasında deęerler verilir, ikilik düzende 5 bit ile ifade edilirse ve sayıların işaretsiz ve tam sayı olduęu kabul edilirse, bu durumda $00000_2=0_{10}$ ve $11111_2=31_{10}$ olacaktır.

Önce, rasgele oluşturulmuş bir dizi topluluęu olarak daha önce kapalı kutu örneğinde ele alınan topluluk dikkate alınacaktır.

Bu topluluęun elemanları Çizelge 2.2. de sol tarafta gösterilmiştir. Çizelgede soldan saęa doğru x 'in onluk tabandaki deęeri, amaç fonksiyon deęeri, seçilme olasılıęı, buna göre bir sonraki nesildeki tahmini sayısı ve rulet çarkı döndürüldükten sonraki sayı verilmiştir.

Çizelge 2.2.a Genetik Algoritmalar

Dizi No.	İlk nesil	x	$f(x)=x^2$	$\frac{f_i}{\sum f}$	$\frac{f_i}{f}$	Sayı
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1

İlk nesildeki dizilerin ortalama uygunluk deęeri 293 ve en iyi uygunluk deęeri ise 576'dır. Bir sonraki adım, bu dizilerden işlem havuzunun oluşturulması ve çaprazlama işleminin uygulanmasıdır. Bunlar da Çizelge 2.2.b'de gösterilmiştir.

Çizelge 2.2.b Genetik Algoritmalar (devam)

İşlem	Eşi	Yer	Yeni nesil	x	$F(x)$
0110-1	2	4	01100	12	0.58
1100-0	1	4	11001	25	1.97
11-000	4	2	11011	27	0.22
10-011	3	2	10000	16	1.23
Toplam					1754
Ortalama					439
En iyi					729

Çizelge 2.2’de üreme ve çaprazlama işlemleri uygulanmıştır. Mutasyon olasılığı 0.001 gibi küçük bir değer olduğundan, yukarıdaki dizilerin $5 \times 4 = 20$ bitinde görülmemiştir. Sonuçta hiçbir bit mutasyon sonucu değişmemiştir.

Çizelge 2.2.a ve çizelge 2.2.b’deki diziler (ve dolayısıyla temsil ettikleri onluk düzendeki sayılar) karşılaştırıldığında en iyi sonuca yaklaşıldığı görülmektedir. Ortalama ve en yüksek uygunluk değeri yükselmiştir. Örneğin ortalama, 293’ten 439’a yükselmiştir. Ancak genetik algoritmalar olasılığa da dayalı olduğu için bir nesilden diğerine geçerken iyileşme bu kadar hızlı olmayabilir. Hatta yeni neslin ortalaması bir öncekinden daha düşük olabilir. Fakat uzun vadede bulunacak sayılar mutlaka ilk nesli oluşturanlardan daha iyi amaç fonksiyon değerleri vereceklerdir.

Genetik algoritmalarındaki en son adım, bulunan yeni nesildeki elemanları, ilk nesil gibi ele alarak işlemleri bunların üzerinde yeniden uygulamaktır. Bu işlemler tekrar ve tekrar, belli bir durma kriteri sağlanana kadar yürütülür. En son bulunan sayı tekrar onluk düzene çevrilerek kullanılır.

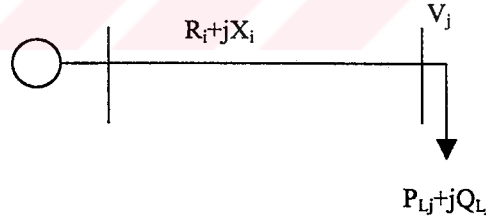
3. ENERJİ DAĞITIM SİSTEMLERİNDE HATTIN AKTİF GÜÇ KAYIPLARINI MİNİMUM YAPACAK YENİDEN YAPILANDIRMA PROBLEMİNİN MATEMATİKSEL MODELİ

3.1 Giriş

Enerji dağıtım sistemleri çoğunlukla dallı olarak düzenlenir. Dağıtım fiderleri belirli sayıda normalde kapalı ve normalde açık anahtarla kumanda edilen fider bölümlerinden oluşur. Bağlantı anahtarları normalde açık, bölümlendirme anahtarları ise normalde kapalı olarak tasarlanır. Normal işletme koşullarında, sistem mühendisleri dağıtım sistemlerinde güvenilirliği arttırmak ve hat kayıplarını azaltmak amacıyla yeniden yapılandırma işlemini gerçekleştirirler.

Bir enerji dağıtım sistemi çok sayıda anahtardan oluşur ve bu anahtarlar üzerinde uygulanabilecek anahtarlama konfigürasyonları sayısı ise çok daha fazla sayıda olmaktadır. Bu sebepten dolayı, yeniden yapılandırma problemi kompleks bir optimizasyon problemi haline dönüşmektedir. Tüm bu karmaşık sistemi çözebilmek amacıyla problemin adım adım matematiksel modelinin çıkartılması gerekmektedir.

3.2 Üç Fazlı Alternatif Akım Sistemlerinde Güç ve Akım İfadeleri



Şekil 3.1 Bir hat ve bir yükten oluşan basit bir enerji sistemi

Şekil 3.1'de gösterilen üç fazlı enerji sisteminde güç ifadeleri: j barası için

$$P_{Lj} = 3 \cdot V_j \cdot I_{Lwj} \quad [\text{Watt}] \quad (3.1)$$

$$Q_{Lj} = 3 \cdot V_j \cdot I_{Ldj} \quad [\text{Var}] \quad (3.2)$$

$$S_{Lj} = 3 \cdot V_j \cdot I_{Lj} \quad [\text{Va}] \quad (3.3)$$

olarak verilir ve sırasıyla aktif, reaktif ve görünür güç olarak isimlendirilir. Akımlar da akım aktif bileşeni ve akım reaktif bileşeni olmak üzere;

$$I_{Lwj} = I_{Lj} \cdot \cos \varphi_j \quad (3.4)$$

$$I_{Ldj} = I_{Lj} \cdot \sin \varphi_j \quad (3.5)$$

iki bileşenle ifade edilir.

3.3 Üç Fazlı Alternatif Akım Sistemlerinde Aktif Güç Kaybı İfadesi

Alternatif akım sistemlerinde fiderlerin direnci sebebiyle oluşan aktif güç kayıpları;

$$P_k = 3 \cdot R_i \cdot |I_{Lj}|^2 \quad (3.6)$$

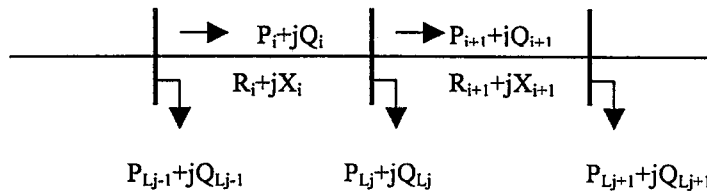
veya

$$P_k = 3 \cdot R_i \cdot \left(\frac{P_{Lj}^2 + Q_{Lj}^2}{V_j^2} \right) \quad (3.7)$$

olarak ifade edilir.

3.4 Enerji Dağıtım Sistemlerinde Hat Akış Şemalarının Çıkartılması

Enerji dağıtım sisteminde her bir fider bileşeni üzerinden geçen akımları/güçleri hesaplayabilmek için basit olarak güç akış analizlerinin yapılması gerekir. Şekil 3.2'de, bir dağıtım fiderine ilişkin güç akışı gösterilmiştir.



Şekil 3.2 Bir enerji dağıtım fideri tek hat diyagramı ve güç akış şeması

“i” fider parçasındaki güç akışı;

$$P_i = P_{i+1} + P_{Lj} + 3 \cdot R_{i+1} \frac{(P_{i+1} + P_{Lj})^2 + (Q_{i+1} + Q_{Lj})^2}{V_j^2} \quad (3.8)$$

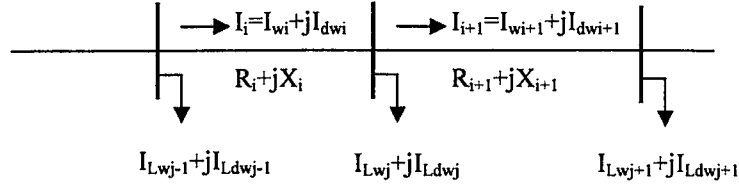
şeklinde ifade edilir.

Enerji dağıtım sisteminde talep yüklerine göre hat kayıplarının çok düşük olduğu göz önünde

bulundurulduğunda, formüldeki ikinci dereceli bileşen ihmal edilerek bir basitleştirmeye gidilebilir. Böylece fider güç akışları, sadece sabit talep yüklerin toplamıyla elde edilebilir.

$$P_i = P_{i+1} + P_{Lj} \quad (3.9)$$

Talep güçler yerine akımlar kullanılabilir. Bu durum, Şekil-3.3'de gösterilmiştir.



Şekil 3.3 Bir enerji dağıtım sisteminde akımların dağılımı

Eşitlikler ise;

$$I_{wi} = I_{wi+1} + I_{Lwj} \quad \text{ve}$$

$$I_{dwi} = I_{dwi+1} + I_{Ldwj} \quad (3.10)$$

şeklini alır. Sonuç olarak hat akımı;

$$I_i = I_{wi} + j \cdot I_{dwi} \quad (3.11)$$

ya da

$$|I_i| = \sqrt{(I_{wi})^2 + (I_{dwi})^2} \quad (3.12)$$

şeklinde ifade edilir.

3.5 Enerji Dağıtım Sistemlerinde Toplam Aktif Güç Kaybının Hesaplanması

Şekil 3.3'de verilen fider kesiti için güç kaybı ifadesi;

$$P_k = 3 \cdot R_i \cdot (I_i)^2 \quad (3.13)$$

ya da,

$$P_k = 3 \cdot R_i \cdot [(I_{wi})^2 + (I_{dwi})^2] \quad (3.14)$$

şeklinde yazılabilir. Benzer şekilde, enerji dağıtım sisteminin bütün parçalarına bu işlem uygulanarak toplam aktif güç kaybı hesaplanmış olur. Bu ise; N_k , fider parçası sayısını

göstermek üzere,

$$TP_k = \sum_{i=1}^{N_k} 3 \cdot R_i \cdot [(I_{wi})^2 + (I_{dwi})^2] \cdot |D_i - 1| \quad (3.15)$$

şeklinde ifade edilebilir. Burada D_i , her bir fider parçasındaki anahtarların kapalı ya da açık olduklarını belirten konum matrisidir.

3.6 Hedef Fonksiyonunun Belirlenmesi

Elektrik enerji sistemlerinin her kademesinde, enerjinin mümkün olan en az kayıp ve minimum distorsiyonla tüketicilere ulaştırılması hedeflenir. Özellikle fiderlere ilişkin aktif kayıpların düşük seviyelerde tutulması, enerji sistem tasarlama sürecinin birincil hedefleri arasındadır. Fiderlere ilişkin toplam aktif kayıpların minimuma yakınsamasını sağlayacak olan ifade hedef fonksiyonu olarak tanımlanır ve;

$$\min TP_k = \sum_{i=1}^{N_k} 3 \cdot R_i \cdot [(I_{wi})^2 + (I_{dwi})^2] \cdot |D_i - 1| \quad (3.16)$$

şeklinde ifade edilir.

4. UYGULAMADA KULLANILAN ÖRNEK ENERJİ DAĞITIM SİSTEMİNE İLİŞKİN HAT VE YÜK VERİLERİ

4.1 Giriş

Elektrik enerji sistemleri üzerine yapılan çalışmalarda genellikle sistemi temsil etmesi düşünülen basitleştirilmiş, gerçek sisteme göre çok daha küçük boyutlarda sistemler veya bunların bilgisayar simülasyonları kullanılır. Çoğu zaman bunun yapılması bir zorunluluk olarak ortaya çıkmakta, mevcut enerji sistemleri üzerinde gerçek değerlere dayalı çalışmalar yapmak, özellikle ülkemizde, ancak bir takım projeler çerçevesinde gerçekleştirilmektedir.

Bu sebeple, çalışmada, gerçek boyutlu enerji dağıtım sistemlerini temsil edebilmesi ve gerçek sistemlere uygulanabilir olması bakımından Mesut E. Baran ve Felix F. Wu'nun (1989) örnek enerji dağıtım sistemi ele alınmıştır. Bu örneğin seçilmesinin diğer önemli sebebi ise, önerilen enerji dağıtım sistemine ait verilerin eksiksiz mevcut olması ve ayrıca bu örnek sistem üzerine yapılmış farklı çalışmaların bulunmasıdır.

4.2 Örnek Enerji Dağıtım Sisteminin Tanıtılması

Örnek sistem; 12,66 kV'luk dağıtım transformatör merkezinden beslenen, 32 adet yük barası ve 37 adet hat parçasından oluşan bir enerji dağıtım sistemidir. 37 hat parçasının 32'sinde normalde kapalı olarak çalışan bölümlendirme anahtarları mevcuttur. Geri kalan 5'inde ise normalde açık bağlantı anahtarları ile donatılmıştır.

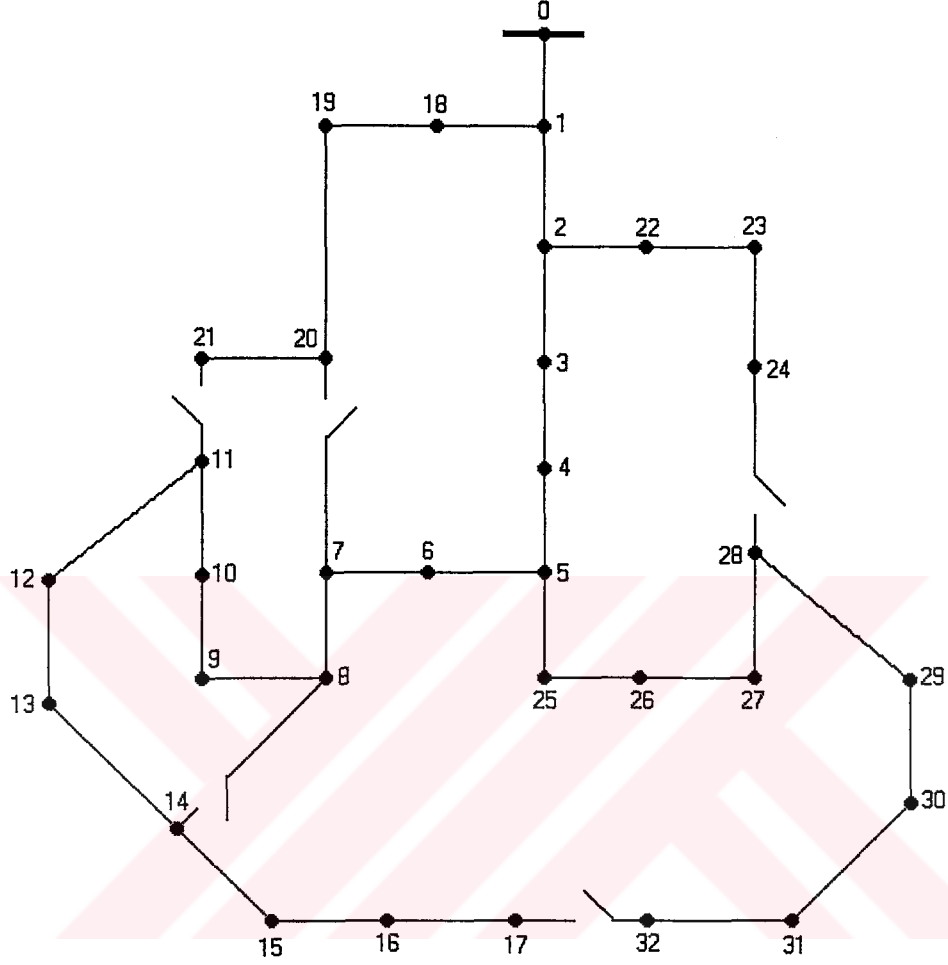
Bölümlendirme anahtarları; koruma, bir arızanın açılması ya da yeniden yapılandırma çalışmalarının gerçekleştirilebilmesi amacıyla fider bölümlerinde kullanılır.

Bağlantı anahtarları ise, gerektiğinde dağıtım sistemini kapalı bir şebeke haline dönüştürebilecek ya da bir fiderden başka bir fidere yük transferini gerçekleştirebilecek noktalarda yerleştirilir.

Uygulamada kullanılacak olan enerji dağıtım şebekesi Şekil 4.1'de gösterilmiştir (Fan vd., 1996).

Sıfır numaralı nokta, referans barasını (besleme noktası), diğer noktalar üzerindeki numaralar ise ilgili yük baralarını göstermektedir. Noktalar arasındaki düz çizgiler ise normalde kapalı anahtarlarla donatılmış her bir hat parçasını göstermektedir. Bağlantı anahtarları ise dallı şebeke yapısının tesis edilmesi amacıyla açık bırakılmıştır. Hatların konumları ve empedansları Çizelge 4.1'de verilmektedir. Uygulama ile ilgili olarak gerçekleştirilecek bütün

işlemlerde, çizilecek şekillerde ve yapılacak atıflarda Şekil 4.1 ve buna bağlı olarak Çizelge 4.1'deki notasyonlar kullanılacaktır.



Şekil 4.1 Uygulamada kullanılacak örnek enerji dağıtım sistemi

4.2.1 Hat Verileri

Şekil 4.1'de gösterilen enerji dağıtım sistemine ilişkin hat verileri Çizelge 4.1'de verilmektedir. Burada ilk sütunda hatların numaraları tanımlanmakta, ikinci ve üçüncü sütunlarda hatların hangi baralar arasında bağlı oldukları gösterilmektedir.

Bu amaçla ilk bara ve son bara adı altında numarası küçük olan bara önce yazılacak şekilde iki sütun oluşturulmuştur.

Hatların karakteristiklerini gösteren veriler, dördüncü ve beşinci sütunlarda sırasıyla direnç ve

Çizelge 4.1 Örnek enerji dağıtım sistemi hat verileri

Hat Numarası	İlk Bara	Son Bara	Hat Direnci R (ohm)	Hat Reaktansı X (ohm)
1	0	1	0.0922	0.0470
2	1	2	0.4930	0.2511
3	2	3	0.3660	0.1864
4	3	4	0.3811	0.1941
5	4	5	0.8190	0.7070
6	5	6	0.1872	0.6188
7	6	7	0.7114	0.2351
8	7	8	1.0300	0.7400
9	8	9	1.0440	0.7400
10	9	10	0.1966	0.0650
11	10	11	0.3744	0.1238
12	11	12	1.4680	1.1550
13	12	13	0.5416	0.7129
14	13	14	0.5910	0.5260
15	14	15	0.7463	0.5450
16	15	16	1.2890	1.7210
17	16	17	0.7320	0.5740
18	1	18	0.1640	0.1565
19	18	19	1.5042	1.3554
20	19	20	0.4095	0.4784
21	20	21	0.7089	0.9373
22	2	22	0.4512	0.3083
23	22	23	0.8980	0.7091
24	23	24	0.8960	0.7011
25	5	25	0.2030	0.1034
26	25	26	0.2842	0.1447
27	26	27	1.0590	0.9337
28	27	28	0.8042	0.7006
29	28	29	0.5075	0.2585
30	29	30	0.9744	0.9630
31	30	31	0.3105	0.3619
32	31	32	0.3410	0.5302
33	7	20	2.0000	2.0000
34	8	14	2.0000	2.0000
35	11	21	2.0000	2.0000
36	17	32	0.5000	0.5000
37	24	28	0.5000	0.5000

endüktif reaktans olmak üzere gösterilmiştir. Hatların dirençleri ve endüktif reaktansları

“ohm” cinsinden gerçek değerler olarak verilmiştir.

4.3 Bara Yük Verileri

Şekil 4.1’de gösterilen orta gerilimli enerji dağıtım sistemine ilişkin yük verileri Çizelge 4.2’de listelenmiştir.

Çizelge 4.2 Örnek enerji dağıtım sisteminin bara talep yük verileri

Bara Numarası	Aktif Güç (kW)	Reaktif Güç (kvar)
1	100	60
2	90	40
3	120	80
4	60	30
5	60	20
6	200	100
7	200	100
8	60	20
9	60	20
10	45	30
11	60	35
12	60	35
13	120	80
14	60	10
15	60	20
16	60	20
17	90	40
18	90	40
19	90	40
20	90	40
21	90	40
22	90	50
23	420	200
24	420	200
25	60	25
26	60	25
27	60	20
28	120	70
29	200	600
30	150	70
31	210	100
32	60	40

Burada ilk stun bara numaralarını, ikinci ve nc stunlar ise sırasıyla ilgili baranın talep aktif ve reaktif glerini gstermektedir. Sıfır numaralı bara bir yk barası olmadığından bu izelgeye dahil edilmemiřtir.



5. ENERJİ DAĞITIM ŞEBEKELERİNİN YAPISI VE İŞLETME ŞARTLARININ TANIMLANMASI

Elektrik enerjisinin üretilmesinden tüketicilere ulaştırılmasına kadar geçen süreçte üzerinde en fazla durulması gereken kısım, enerjinin üretim ya da iletim tesislerinden alınarak tüketicilere ulaştırıldığı dağıtım tesisleridir. Enerjinin tüketicilere kaliteli ve sürekli olarak sağlanması, üretildiği sırada tüketilmesi bir zorunluluktur. Bu anlamda enerji dağıtım sistemlerinin tasarlanmasında, teknik olarak geleceğe dair yapılan öngörülerin iyi değerlendirilmesi gerekmektedir. Gün geçtikçe artan talebin önceden tahmin edilmesindeki belirsizlikler eskiye göre daha fazladır (Sun vd., 1982; Ponnaivaikko vd., 1982).

Karmaşıklığı gün geçtikçe artan enerji dağıtım sistemlerinin işletilmesi ise sadece teknik şartların karşılanması şeklinde değil, aynı zamanda talep yönetimi ve güvenilirlik analizleri yapılarak gerekli tedbirlerin alınmasıyla optimum bir şekilde gerçekleştirilir.

Enerji dağıtım şebekesi, tüm enerji sisteminin son kademesini oluşturmaktadır. Burada yapılacak bütün düzenlemeler öncelikle dağıtım sistemini ilgilendirse de, aynı zamanda iletim ve üretim tesislerinin performanslarını da doğrudan etkiler. Bu bakımdan, dağıtım şebekelerinin işletilmesi ayrı bir önem taşımaktadır.

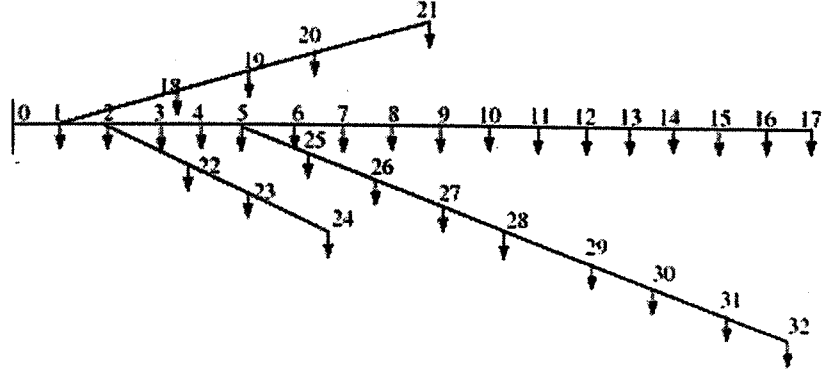
Tüm enerji sistemleri bilinen belirli ortak kurallar çerçevesinde işler. Fakat her bir sistemin kendine has yapısı ve farklı karakteristikleri vardır. Bu durum, aynı sınıftan sistemler için de geçerlidir. Farklı sistemlerin yapılarının ve işletme koşullarının farklı olması ise oldukça normal bir durumdur. Dağıtım sistemleri de kendine has bir takım uyulması zorunlu kriterleri ve aşılmaması gereken kısıtlamaları yerine getirmelidir.

Enerji dağıtım sisteminin yapısı ve işletme şartları ne kadar iyi bilinirse taşıdığı problemlerin çözümü de o ölçüde çabuk ve hatasız olacaktır.

Bu bölümde genel olarak enerji dağıtım sistemlerinin yapısı ve işletme koşulları incelenirken, yapılacak uygulama ve uygulanacak kurallar hakkında bilgi verilecektir.

5.1 Enerji Dağıtım Sisteminin Dallı Yapısı

Enerji dağıtım sistemleri çoğunlukla dallı olarak işletilir. Dallı şebekeler güç kayıpları ve güvenilirlik açısından kapalı şebekelere göre daha elverişsiz görünse de, daha basit işletilebilmeleri ve tesis maliyetleri açısından tercih sebebidir. Şekil 5.1’de dallı bir şebeke örneği görülmektedir.



Şekil 5.1 Tipik bir dallı şebeke (M. A. Kashem, 2000)

Tasarım mühendisleri kısıtları dikkate alarak mevcut şartlar içerisinde en iyi çözümü ararlar (Jonnavithula vd., 1996). Enerji dağıtım sistemini dallı olarak tasarladıkları gibi, güvenilirlik şartlarını da dikkate almak görevleri arasındadır. Tüketicinin memnuniyeti, işletmecinin başarısı anlamını taşır. Bu amaçla gerek olumsuz şartların olabileceği düşüncesi gerekse gelecekle ilgili bir takım tahminler veya beklentiler, dallı dağıtım sisteminde gerektiğinde fiderleri birbirleri ile ilişkilendirebilecek bir takım tedbirler almaya götürür.

Bu tedbirlerin başında normalde açık olarak tasarlanan bağlantı anahtarları gelir. Bağlantı anahtarları öyle noktalara yerleştirilir ki, bunlar sayesinde şebeke kapalı şebekeye dönüştürülebilir. Amaç kapalı şebeke oluşturmak olmasa bile bütün yüklerin farklı noktalardan beslenebilmesi imkanı sağlanır.

Dallı yapı, öncelikle güvenlik gereksinimlerinin daha basit karşılanması amacıyla istenir. Selektif koruma, bu yapının vazgeçilmezidir ve yapısı çok karmaşık değildir. En önemli yanı ise, basit şebeke yapısı ve buna bağlı olarak işletme kolaylığıdır.

Bu çalışmada, enerji dağıtım sisteminin yeniden yapılandırılmasında dallı yapının korunması öncelikli hedefler arasındadır.

5.2 Enerji Dağıtım Sistemlerinde Yük Dengesinin Sağlanması

Yüklerin dengeli olarak dağıtılması, enerji dağıtım tesislerinde yüklerin fazlara eşit seviyelerde dağıtılması demektir. İletim tesislerinde de fazlara yüklerin dengeli olarak paylaşılması gereklidir. Enerji sistemlerinin her aşamasında dengeli yük dağılımı esastır (Baran vd., 1989).

Dengeli yük dağılımı, her bir fazdaki gerilimlerin benzer değerler almasına ve bunun sonucu olarak da işletme veriminin artmasına yol açar. Dengesiz yüklenme, enerji sisteminin

tamamında sistem bileşenlerinin optimum olmayan çalışma noktalarında çalışmasına ve cihazlarda erken yaşlanma problemlerine sebep olmaktadır. Optimum olmayan güç akışı ile sonuçlanmakta ve enerji sistemi kayıp maliyetlerini arttırmaktadır.

Dallı enerji dağıtım sistemlerinde bağlantı anahtarları gerektiğinde yüklerin dengeli dağılımı için de kullanılabilir. Yeniden yapılandırma işlemi yük dengelemesi problemine de çözüm olmaktadır.

Bu çalışmanın uygulama aşamasında yüklerin fiderler üzerinde dengeli dağılımını sağlayacak herhangi bir hedef fonksiyonu geliştirilmemiştir.

5.3 Yüklerin Enerjilenmesi

Enerji dağıtım sistemlerinde normal işletme şartlarında bütün yüklerin sürekli olarak devrede olması sağlanır. Başka bir deyişle, yükler devre dışı bırakılmaz. Çünkü bu yüklerin tamamının tüketicilerden oluştuğu ve her an enerji talebinde bulunduğu varsayılır. Her ne kadar yüklerin karakteristikleri yük tipine ve zamana bağlı olarak değişse de amaç, tüketicilere en az kesintiyle servis sağlamaktır.

Yine bu kapsamda bir fiderin belirli bir bölümünde arıza meydana geldiğinde, dallı yapıdaki enerji dağıtım sistemlerinde uygun bağlantı anahtarları, arızalı fider bölümü devre dışı edildikten sonra yüklerin enerjisiz kalmaması amacıyla kapatılarak, servisin sürekliliği sağlanır ve sistemin güvenilirliği artırılır.

Enerji dağıtım sisteminde, sistemi yeniden yapılandırırken dallı şebeke formasyonunun sağlanabilmesi amacıyla bir takım yüklerin devre dışı bırakılması söz konusu olamaz. Yeniden yapılandırma probleminin çözümünde de şebekede açık bırakılacak anahtarlar öyle seçilmelidir ki, hem dallı yapı tesis edilsin hem de hiçbir yük devre dışı kalmasın. Bu kısıt, tüm süreç boyunca kontrol edilmek zorundadır.

5.4 Gerilim Düşümü Kısıtları

Elektrik enerji sistemlerinin tasarımında ve işletilmesinde uyulması gerekli kriterlerden birisi de gerilim düşümü kısıtlarıdır. Gerilim düşümü, enerji sistem bileşenlerinin empedansları neticesinde oluşur. Gerilim düşümü kısıtı sayesinde, gerilimin çok fazla kayba uğramadan tüketicilere ulaştırılması amaçlanır.

Enerji dağıtım sistemlerinde gerilim düşümü sınırlarının ihlal edilmesi, alıcı cihazların verimli çalışmamasına veya hasar görmesine sebep olabilir. Bütün alıcıların çalışabildikleri gerilim

bandı yaklaşık olarak belirlidir ve aşılması, sakıncalı durumların ortaya çıkmasına neden olur.

Enerji sistemlerinde gerilim düşümü sınırları standartlarda belirlenmiştir. Bu oranların aşılması tüketiciler açısından olumsuzluklara neden olurken, aynı zamanda sistemlerin optimum olmayan işletme noktalarında çalışmalarına sebep olur.

Uygulama çalışmalarının gerçekleştirildiği örnek enerji dağıtım sisteminde gerilim düşümü sınırları ± 0.1 pu yani $\pm \%10$ olarak belirlenmiştir. Bu sınırların aşılmaması hedeflenmekle birlikte, Genetik Algoritmanın araştırma uzayının daraltılmaması amacıyla yazılan programda bu kısıta yer verilmemiştir. Gerilim düşümü kısıtı sadece bulunan en iyi çözüm için kontrol edilmektedir.

5.5 Akım Taşıma Kapasiteleri

Enerji sistemlerinin her aşamasında iletkenlerin optimum şartları sağlayacak kapasitede seçilmeleri bir zorunluluktur. İletkenlerin akım taşıma kapasiteleri bu şartların sağlanmasında bir gösterge niteliğindedir. Her iletkenin malzeme yapısına, boyutlarına ve imalat şartlarına göre taşıyabilecekleri akımlar belirlidir. Uygun iletkenlerin seçimi, iletkenlere ait ilgili tablolar vasıtasıyla gerçekleştirilir. Tasarım ve işletme mühendislerini zorlayan sınırlamalardan bir tanesi de iletkenlerin akım taşıma kapasiteleridir (Tram vd., 1988; Wang vd., 2000).

Bu uygulamada incelenen sistem, mevcut bir enerji dağıtım sistemidir ve iletkenlere ait sadece direnç ve endüktif reaktans bilgileri verilmektedir. İletkenlerin cinsi, kesiti veya uzunlukları hakkında bilgi verilmemektedir. Daha önce yapılan çalışmalardan iletkenlerin akım taşıma sınırlarına dair bir problemin oluşmadığı sonucu çıkarılmaktadır. Bu nedenle, akım taşıma ile ilgili bir sınırlama getirilmemekte, bütün iletkenlerin uygun boyutlarda olduğu düşünülmektedir.

5.6 Hat Aktif ve Reaktif Güç Kayıplarının Sınırlanması

Üretilen enerjinin çok fazla kayba uğramadan tüketicilere ulaştırılması esastır ve bu kapsamda tarih boyunca çalışmalar sürdürülmektedir. Enerji dağıtım sistemlerinde de hat kayıplarının mümkün olan en düşük seviyelerde tutulması istenir (Curcic vd., 2001). Bu çerçevede, hattaki aktif ve reaktif güç kayıplarını sınırlandırmak için tedbirler alınır. Bu uygulamalardan biri de enerji dağıtım sistemlerinin yeniden yapılandırılmasıdır.

Günün şartlarına göre tasarlanan sistemlerin değişen şartlar çerçevesinde revize edilmesi bir

zorunluluktur. Sistemde bir takım yapısal deęişiklikler ve ilave düzenlemeler yapılabileceęi gibi, basit müdahalelerle de sistemin performansı arttırılabilir. Mevcut bir sistemin bir takım anahtarlama işlemleri neticesinde yeni bir yapı kazandırılarak kayıplarının azaltılması da mümkündür.

Çalışmanın daha önceki bölümlerinde yeniden yapılandırma işleminin farklı amaçlar için yapılabileceęi tanımlanmıştı. Bu bağlamda bazı çalışmalarda birden fazla hedef fonksiyonu aynı anda incelenmiştir (Roytelman vd., 1996; Hsiao vd., 2001). Bu çalışmada ise, enerji dağıtım sistemlerinde yeniden yapılandırma, hattın aktif güç kayıplarını azaltmak amacıyla yapılmaktadır.



6. YENİDEN YAPILANDIRMA PROBLEMİNE GENETİK ALGORİTMANIN UYGULANMASI

6.1 Giriş

Yeniden yapılandırma, isminden de anlaşılacağı gibi mevcut durumun değiştirilmesi anlamını taşır. Enerji dağıtım sisteminde bunu başarmak için en az iki adet anahtarın konumlarını değiştirmek gerekir. Basit olarak yapılacak bu iki anahtarlı işlemde bile dikkat edilecek pek çok durum vardır. Yüzlerce anahtardan oluşabilecek gerçek bir sistemde ise karşılaşılan güçlükler katlanarak artmaktadır.

Enerji dağıtım sistemlerinde yeniden yapılandırma çalışmalarında genel olarak hesap temelli yöntemlerin kullanıldığı görülmektedir. Lineer olmayan fonksiyonların çözüldüğü bu süreçte hesaplama süreleri de uzun olabilmektedir. Farklı bir yaklaşım olarak Genetik Algoritma metodu, karmaşık işlemsel yükün azaltılması ve bilgisayar hesaplama zamanları açısından daha iyi sonuçların alınması amacıyla Bölüm 4'te tanımlanan enerji dağıtım sistemine uygulanmıştır. Bu uygulama için C++ programlama dilinde yazılan program metni Ek 1'de verilmiştir.

6.2 Genetik Algoritma Metodu ile Enerji Dağıtım Sisteminin Tanımlanması ve Genel Akış Şemasının Çıkarılması

Optimizasyon işleminde kullanılacak Genetik Algoritmaların genel yapısı ve işlem basamakları Bölüm 2'de detaylı olarak verilmiştir.

Bu çalışmada geliştirilen genetik algoritma, enerji dağıtım sisteminin yapısal bileşenleri/parametreleri olan anahtarlar üzerinde bit seviyesinde üç operatör ile işlem yaparak en iyi çözümü veren sistem yapısının bulunmasında kullanılmıştır.

Enerji dağıtım sistemini temsil eden 37 adet anahtar, 37 bitten oluşan bir zincir şeklinde her biri bir aday çözümü gösterecek şekilde sembolize edilmiştir. 37 bitten oluşan bu zincirlerin her birine üye ya da birey denilecektir.

Her bir bireydeki lojik "0" ve "1"ler anahtarların konumlarını gösterir. Anahtar kapalı ise "0", açık ise "1" sayısı görevlendirilir. Aynı zamanda anahtarların konumlarını gösteren "0" ve "1"ler, hedef fonksiyonun hesaplanmasında D_i konum matrisinin bileşenlerini de oluşturmaktadır. Şekil 4.1'deki enerji dağıtım şebekesine ilişkin eşdeğer Genetik Algoritma birey çözümü, Çizelge 5.1'de gösterilmiştir.

Çizelge 5.1 Şekil 4.1'deki sisteme eşdeğer birey Genetik Algoritma zinciri

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30	S31	S32	S33	S34	S35	S36	S37	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Şekil 4.1'de 5 adet bağlantı anahtarı açık olarak çizilmiş ve buna karşılık olarak da S33, S34, S35, S36, S37 anahtarları Genetik Algoritmada "1" değerini almışlardır. Diğer anahtarlar ise fider bölümlendirme anahtarlarıdır ve hepsi kapalı konumdadır. Buna karşılık olarak "0" değerini almışlardır.

Genetik Algoritma, başlangıçta 15 adet bireyi rasgele oluşturarak bir popülasyon ortaya çıkartmaktadır. Yazılan programda popülasyondaki üye sayısı ayarlanabilmektedir; ancak üye sayısının düşük olması popülasyonun erken yakınsamasına, yüksek olması ise bilgisayar hesaplama zamanlarının artmasına neden olmaktadır. Burada önemli olan yeterli büyüklükte bir üye sayısının belirlenmesidir.

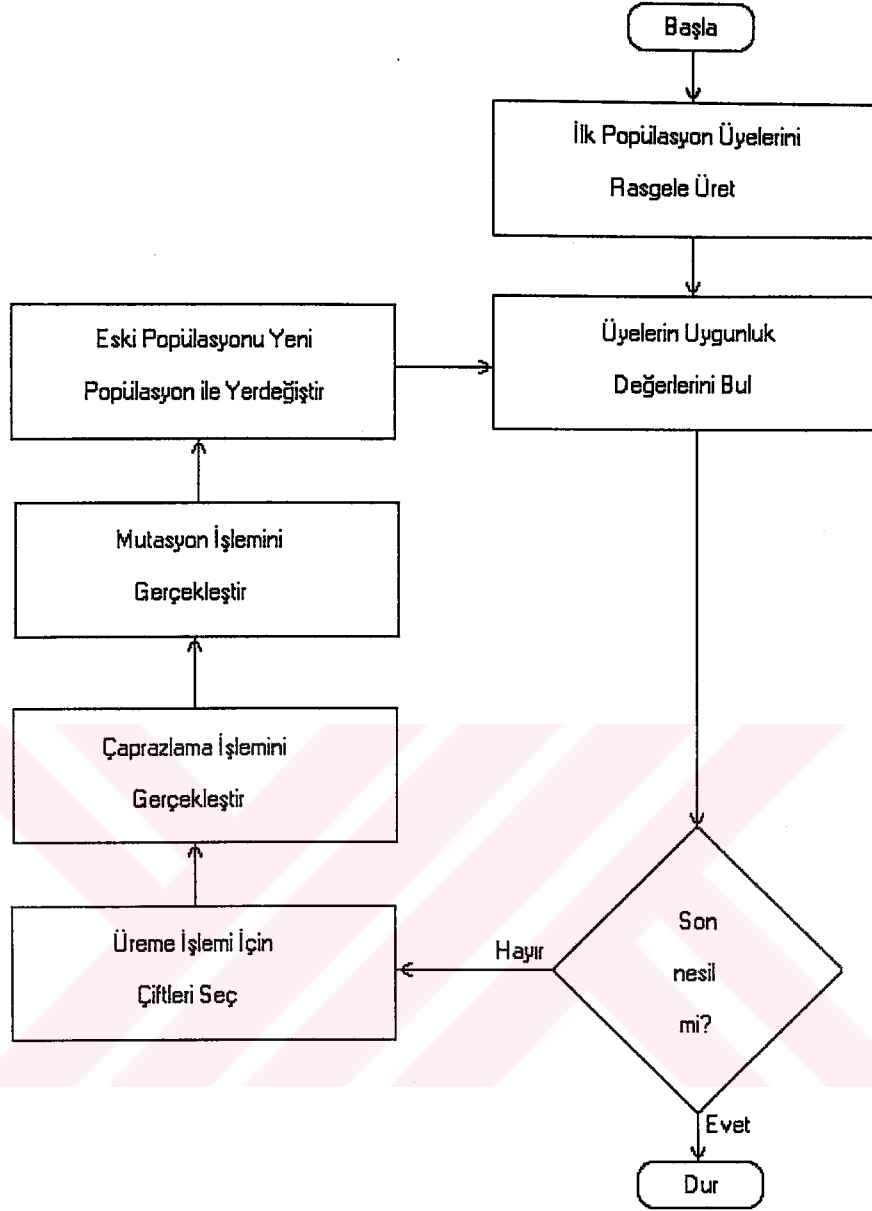
İlk popülasyonun rasgele oluşturulmasının ardından her birey hedef fonksiyonu ile değerlendirilir. Hedef fonksiyonu değerlerine göre bireylerin yeni popülasyona kopyalanıp kopyalanmayacağına karar verilir ve kopyalama işlemi, üreme operatörü ile gerçekleştirilir.

Hedef fonksiyonuna uygunluk fonksiyonu (fitness function) da denilmektedir. Uygunluk değeri yüksek olan birey, yeni popülasyona yüksek bir olasılıkla kopyalanır.

Üreme işlemi Bölüm 2'de detaylı bir şekilde anlatılan rulet çarkı yöntemi ile gerçekleştirilmiştir. Çaprazlama operatörü olarak grupsal çaprazlama yöntemleri kullanılmıştır. Mutasyon işleminde ise enerji dağıtım sisteminde bir anahtara tekabül eden bir tek bitin rasgele değiştirilmesi söz konusudur.

Bu değiştirme işlemi: Genetik zincirde bir adet lojik "1" lojik "0" yapılırken; bir adet lojik "0"ında lojik "1" yapılması ile gerçekleştirilmektedir. Genetik Algoritmanın performansını yükseltmek amacıyla çaprazlama ve mutasyon operatörleri üzerinde yapılabilecek pek çok alternatif işlem mevcut olmakla birlikte uygulamada güçlükler yaşanabilmektedir.

İlk popülasyondan sonra yapılan üreme, çaprazlama ve mutasyon işlemleri ile büyük çoğunlukla atalarının genlerini taşıyan yeni bir popülasyon elde edilmiş ve yeni bir nesil oluşturulmuş bulunmaktadır. Genetik Algoritmanın genel akış şeması olarak da adlandırılabilen bu durum, Şekil 6.1'de özetlenmektedir. Genetik Algoritmada bu süreç, belirli bir nesil sayısına kadar ya da istenilen yakınsama sağlanana kadar devam ettirilmektedir.



Şekil 6.1 Genetik Algoritmanın genel akış diyagramı

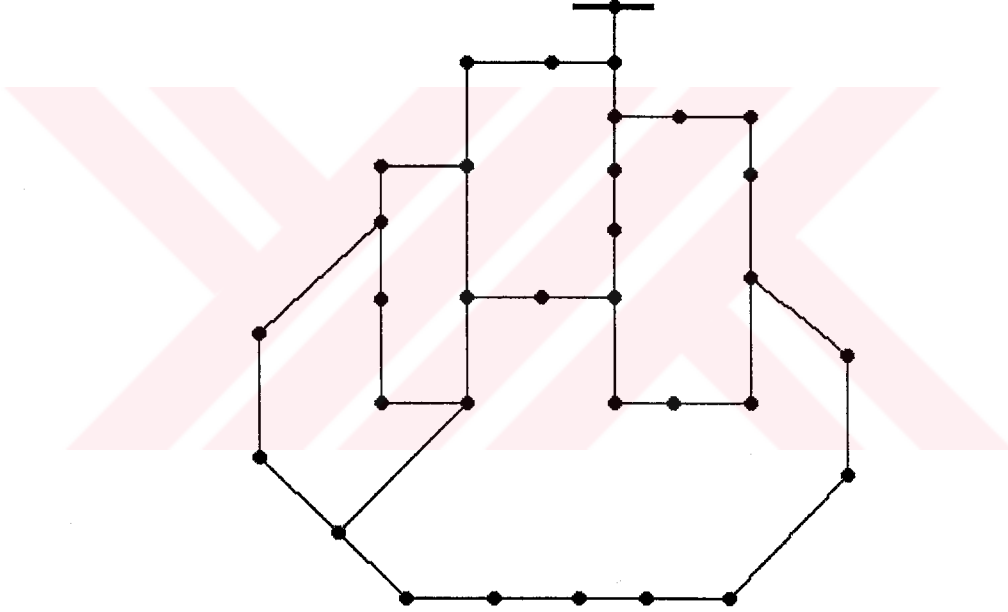
Akış diyagramından da anlaşılacağı gibi ilk popülasyonun rasgele oluşturulması ilk basamağı teşkil etmektedir. Daha sonra üreme, çaprazlama ve mutasyon operatörleri ile yeni bir nesle geçiş yapılmaktadır. İster ilk popülasyonun rasgele oluşturulması sırasında, isterse diğer operatörlerin uygulanması sırasında olsun, Bölüm 5’te tanımlanan enerji dağıtım sistemine ait kısıtların yerine getirilmesi zorunludur. Aksi halde, enerji dağıtım sistemi eksik tanımlanmış ve etkin bir çözüm süreci geliştirilmemiş olacaktır.

6.3 Enerji Dağıtım Sistemine Genetik Algoritma Metodu Uygulanırken Kısıtların Gerçekleştirilmesi

Enerji dağıtım sisteminin tam olarak incelenebilmesi için sistem işletme şartlarının ve kısıtlarının tam olarak yansıtılması gerekmektedir. Yapılan uygulamada, enerji dağıtım sisteminin dallı olarak işletilmesi öncelikli olarak dikkate alınmıştır.

6.3.1 Dallı Yapının Gerçekleştirilmesi

37 adet anahtardan oluşan enerji dağıtım sisteminde 5 adet anahtarın açılmasıyla oluşacak şebeke alternatifleri oldukça fazladır. İlk popülasyonun rasgele üretilmesi işleminden başlamak üzere bütün operatörlerin gerçekleştirilmesi dallı şebeke yapısının sağlanması ile mümkün olacaktır. Bütün anahtarlar kapatıldığında 5 adet bağımsız çevreden oluşan kapalı şebeke şeması Şekil 6.2’de verilmiştir.



Şekil 6.2 Enerji dağıtım sisteminin kapalı şebeke formasyonu

Uygulamada kullanılan enerji dağıtım şebekesi, Şekil 6.2’de görüldüğü gibi bütün anahtarları kapatılarak kapalı şebeke şekline dönüştürülmüştür.

Beş adet bağımsız çevreden oluşan bu sistem, ancak beş adet anahtarın açılması ile dallı şebeke şekline dönüştürülebilir. Hiçbir kapalı göz kalmaması gereken sistemde, her bağımsız çevreden bir anahtar açılmak zorundadır. Bu ise aşağıdaki eşitliklerin kontrol edilmesi ile sağlanabilir.

$$D_3 + D_4 + D_5 + D_{22} + D_{23} + D_{24} + D_{25} + D_{26} + D_{27} + D_{28} + D_{37} = 1 \quad (6.1)$$

$$D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_{18} + D_{19} + D_{20} + D_{33} = 1 \quad (6.2)$$

$$D_8 + D_9 + D_{10} + D_{11} + D_{21} + D_{33} + D_{35} = 1 \quad (6.3)$$

$$D_9 + D_{10} + D_{11} + D_{12} + D_{13} + D_{14} + D_{34} = 1 \quad (6.4)$$

$$D_6 + D_7 + D_8 + D_{15} + D_{16} + D_{17} + D_{25} + D_{26} + D_{27} + D_{28} + D_{29} + D_{30} + D_{31} + D_{32} + D_{34} + D_{36} = 1 \quad (6.5)$$

(6.1), (6.2), (6.3), (6.4) ve (6.5) eşitlikleri kapalı çevreleri ve her kapalı çevreden bir anahtarın açık olması gerektiğini tanımlayan eşitliklerdir. Kapalı anahtarlar D_i konum vektörü üzerinde "0" ile, açık anahtarlar ise "1" ile gösterilmiştir. Eşitliklerin "1"e eşitlenmesi, her çevreden bir adet anahtarın açık bırakılacağı anlamını taşımaktadır.

İlk popülasyonun oluşturulması işlemi de, bu eşitliklere paralel olarak her kapalı çevreden bir anahtarın rasgele açılması ile sağlanmaktadır. Yani 15 üyeli popülasyonda her üye, kapalı çevrelerden birer anahtarın rasgele açılması ile üretilmektedir.

Beş adet anahtarın açık bırakılması dallı şebeke yapısı için gerekli, fakat yeterli değildir. Dallı yapının tesisi için bir takım koşulların daha sağlanması gerekmektedir. Bunlar, sistemde bağımsız kapalı çevreler dışında oluşabilecek kapalı çevreleri tanımlayan eşitliklerdir.

$$D_2 + D_6 + D_7 + D_{18} + D_{19} + D_{20} + D_{22} + D_{23} + D_{24} + D_{25} + D_{26} + D_{27} + D_{28} + D_{33} + D_{37} = 1 \quad (6.6)$$

$$D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_9 + D_{10} + D_{11} + D_{18} + D_{19} + D_{20} + D_{21} + D_{35} = 1 \quad (6.7)$$

$$D_8 + D_{12} + D_{13} + D_{14} + D_{21} + D_{33} + D_{34} + D_{35} = 1 \quad (6.8)$$

$$D_2 + D_6 + D_7 + D_8 + D_9 + D_{10} + D_{11} + D_{18} + D_{19} + D_{20} +$$

$$D_{21} + D_{22} + D_{23} + D_{24} + D_{25} + D_{26} + D_{27} + D_{28} + D_{35} + D_{37} = 1 \quad (6.9)$$

$$D_2 + D_3 + D_4 + D_5 + D_6 + D_7 + D_8 + D_{12} + D_{13} + D_{14} + D_{18} + D_{19} + D_{20} + D_{21} + D_{34} + D_{35} = 1 \quad (6.10)$$

$$D_2 + D_6 + D_7 + D_8 + D_{12} + D_{13} + D_{14} + D_{18} + D_{19} + D_{20} + D_{21} +$$

$$D_{22} + D_{23} + D_{24} + D_{25} + D_{26} + D_{27} + D_{28} + D_{34} + D_{35} + D_{37} = 1 \quad (6.11)$$

$$D_{15} + D_{16} + D_{17} + D_{29} + D_{30} + D_{31} + D_{32} + D_{36} = 1 \quad (6.12)$$

Rasgele üretilen üyeler için, (6.6), (6.7), (6.8), (6.9), (6.10), (6.11) ve (6.12) eşitlikleri de dikkate alınarak, ilgili çevrelerde en az bir anahtarın açık bırakıldığı kontrol edilir. Eğer üyelerden bir tanesi bu eşitliklerden tek bir tanesini bile karşılamaz ise, o üye dikkate

alınmayarak yerine rasgele yeni bir üye üretilir. Bu sayede enerji dağıtım sisteminde kapalı çevreler ortadan kaldırılıp dallı yapı tesis edilmiş olur.

Bu işlemden sonra uygunluk değerlerinin hesaplanabilmesi amacıyla yüklerin hangi güzergahlardan enerjilendiği bilgisi sorgulanır.

6.3.2 Yüklerin Enerjilenmesi

Rasgele oluşturulan ilk popülasyonda, çaprazlama ve mutasyon operatörleri sonucunda oluşan yeni nesillerde yüklerin hangi güzergahlar üzerinden enerjilendiğini bilmek gerekmektedir. Oluşturulan üyelerin şebeke şekillerinin algılanması anlamına gelen bu işlem sırasında hatlar üzerinden geçen akımlar tespit edilir.

Hatlardan geçen toplam akımların hesaplanması işlemine, açık anahtarların bağlı bulunduğu baraların tespit edilmesiyle başlanır. Dallı şebekede bu baralar en son/uç noktadaki baralardır ve bu işlem sırasında “son bara” olarak isimlendirileceklerdir.

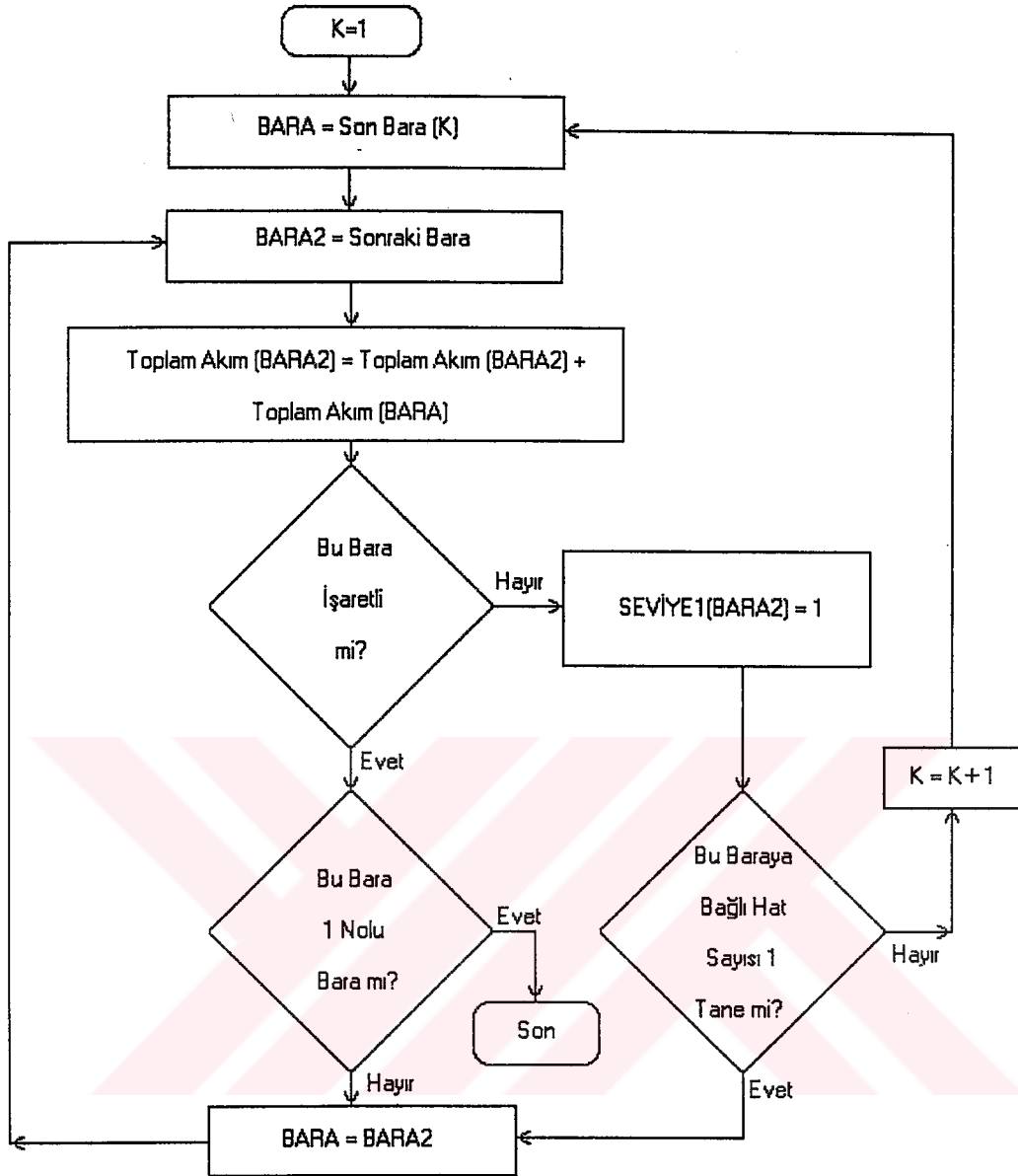
Son baradan geriye doğru ilerleyerek besleme noktasına kadar ulaşılmaya çalışılır. Son baradan geriye doğru hareketle ulaşılabacak ilk baraya BARA2 denir ve bu baraya daha önce gelinip gelinmediğini göstermek amacıyla bir işaret (SEVİYE1) konur.

Bütün baralara başlangıçta henüz ulaşılmadığını göstermek amacıyla SEVİYE1=0 olarak ve ilk defa ulaşılan barada ise bu işaret “1” yapılmak suretiyle işlem devam ettirilir.

Eğer ulaşılan barada bir işaret var ise (daha önceden ulaşılmış ise; SEVİYE1=1), örnek sistemimizde bir bara en fazla iki barayı besleyebildiğinden, akımların toplanması işlemine geriye doğru devam edilir. Eğer ulaşılan barada bir işaret yok ise (SEVİYE1=0), bu durumda bu baranın bağlı bulunduğu hat sayısı tespit edilir (O baraya ulaşmakta kullanılan hat hariç).

Ulaşılan işaretsiz bara bir adet hatla bağlıysa akımların toplanması işlemi geriye doğru devam ettirilir. Eğer işaretsiz baraya iki adet hat bağlı ise bu demektir ki; o bara iki adet kolu beslemektedir ve kollardan birindeki (üzerinde ilerlenen hattaki) akımlar toplanmıştır. İşaretsiz bu bara işaretlenerek işlem durdurulur ve yeni bir son baradan işleme başlanır.

Bu sayede işlem 1 nolu baraya kadar devam ettirilerek hatlardan geçen toplam akımlar gerçek hat numaraları üzerine kaydedilir. Açık anahtarların bulunduğu beş adet hat dışında bütün hatların enerjilendiği görülür. Bu da boşta kalan yükün bulunmadığı anlamına gelir. Bu süreci özetleyen akış diyagramı Şekil 6.3’te gösterilmiştir.



Şekil 6.3 Popülasyondaki her üyeye eşdeğer enerji dağıtım şebekesinin akım dağılımını veren akış diyagramı

Enerji dağıtım sistemine 15 adet aday çözüm üreten ilk popülasyon, rasgele dallı şebeke yapısını oluşturacak şekilde üretildikten sonra her bir üyenin şebeke yapısı ve akım dağılımlarını veren süreç, Şekil 6.3'te verilen akış diyagramı tarzında uygulanır. Bütün hatlardan geçen akımlar bu sayede belirlenir. Bundan sonra her bir üyenin, popülasyondaki diğer üyelere nazaran durumlarını ortaya koyacak olan uygunluk değerlerinin hesaplanması işlemine geçilir.

6.4 Popülasyondaki Üyelerin Uygunluk Değerlerinin Bulunması

Genetik Algoritmada uygunluk fonksiyonu, diğer sayısal algoritmalarındaki amaç fonksiyonunun yerini tutar. Uygunluk denmesindeki maksat, üyelerin problemin çözümüne ne kadar yakın ya da uzak olduklarını göstermek içindir. Uygunluk derecesi yüksek olan üye, problemin çözümünde diğer üyelere nazaran daha fazla rol oynayacaktır.

Enerji dağıtım sisteminde yeniden yapılandırma işlemi, hattın aktif kayıplarını sınırlamak maksadıyla yapıldığına göre, amaç fonksiyonunda toplam aktif güç kayıplarını veren ifade kullanılmalıdır. Sabit akımlardan oluşan enerji dağıtım sisteminde toplam aktif güç kaybı

$$TP_k = \sum_{i=1}^{N_k} 3 \cdot R_i \cdot [(I_{wi})^2 + (I_{dwi})^2] \cdot |D_i - 1|$$

şeklinde ifade edilmişti (ifade (3.15)). İster ilk popülasyonda, isterse daha sonraki popülasyonlarda Bölüm (6.3.2)'deki sürecin uygulanmasıyla tespit edilen hat akımları yardımıyla her bir üyenin toplam güç kaybı bu ifade yardımı ile bulunur.

Toplam kayıp ifadesi Genetik Algoritmada uygunluk değerlerini veren amaç fonksiyonunda

$$f = \frac{1}{\frac{TP_k}{3} - 40} \quad (6.13)$$

şeklinde kullanılmıştır. Enerji dağıtım sisteminde aktif hat kayıplarının minimum yapılması istenildiği için toplam kayıp ifadesi paydada kullanılmıştır. Kayıp güçler üzerinde bir maksimizasyon işlemi söz konusu olsaydı, o zaman direkt olarak amaç fonksiyonu olarak kullanılabilirdi.

Toplam kayıp ifadesi uygunluk değeriyle ters orantılı olmak zorundadır. Yani toplam kayıp ne kadar düşük değer alırsa uygunluk değeri o kadar büyük değer almalıdır. Böylece en düşük toplam kayıp değerini veren üye en yüksek uygunluk değerine sahip olarak çözüm aşamasında etkin bir rol oynayacaktır.

Bölüm 2'de genel olarak belirtildiği gibi

$$f = \frac{1}{TP_k} \quad (6.14)$$

olarak niçin alınmadığı merak edilebilir. Genetik Algoritma, amaç fonksiyonu olarak (6.14) ifadesinin kullanılması durumunda da istenilen sonuçları bulabilmektedir. Ancak bu süreç çok

daha uzun olmaktadır. Bunun sebebi ise, (6.14) ifadesi ile bulunan uygunluk değerlerinin birbirine çok yakın olmasıdır.

Uygunluk değerlerinin birbirine çok yakın olması neticesinde üreme operatörü ile elde edilen yeni neslin üyeleri neredeyse eski neslin üyeleriyle tamamen benzerlik arzietmekte ve yakınsama daha uzun sürelerde gerçekleşmektedir. Ayrıca Genetik Algoritma böyle bir sınırlama getirmemekte, tam aksine performansı yükseltecek değişiklikleri teşvik etmektedir.

(6.14) ifadesi kullanılarak elde edilen bir popülasyonda ilk nesil üyeleri ve bunların birbirine yakın uygunlukları Ek 2’de gösterilmiştir.

(6.13) ifadesi kullanılarak uygunluk değerleri hesaplanan üyeler istenilen sonuca ilk nesilde bile yakınsama şansına sahip olabilir. Genetik Algoritmanın rastsal araştırma yapısı bunun için uygundur. Fakat bu çok düşük bir olasılıktır. İlk neslin değerlendirilmesinin ardından en uygun çözümü bulma çalışması üreme, çaprazlama ve mutasyon operatörlerinin uygulanmasıyla nesilden nesile aktarılır.

6.5 Genetik Algoritma Operatörlerinin Uygulanması

Genetik Algoritmada yeni nesillerin oluşturulması üreme, çaprazlama ve mutasyon operatörlerinin gerçekleştirilmesi ile sağlanır. Her nesil bir sonraki nesle dönüşürken sırasıyla bu operatörlere maruz kalır. Doğadan esinlenerek oluşturulan bu operatörler, yeni nesillerin atalarının özelliklerini taşımalarını sağlamanın yanında, atalarından farklılaşmalarını sağlayacak özellikler de taşır.

6.5.1 Üreme Operatörünün Uygulanması

Bir nesildeki üyelerin yeni nesle kopyalanmaları işlemi üreme operatörü ile gerçekleşir. Eski popülasyondaki üye sayısı kadar üye, yeni nesle kopyalanır.

Genetik Algoritmanın temel varsayımlarından bir tanesi; gürbüz bireylere genetik yapı olarak üstün bireylerin kullanılması ile ulaşılacağı varsayımdır. Buradan hareketle, üyelerin uygunluk değerlerinin kullanıldığı üreme işleminde yüksek uygunluk değerli üyelerin yeni nesilde büyük olasılıkla bulunmalarını sağlamak hedeflenmektedir. Ayrıca üreme işlemi nesiller arasındaki genetik bağın sürekliliğini sağlamaktadır.

Üreme işlemi için literatürde pek çok yöntem tanımlanmıştır. Uygulamada, Bölüm 2’de verilen rulet çarkı tekniği kullanılmaktadır. Bu teknik en etkin yöntem olmamakla beraber yapısının çok karmaşık olmaması genel tercih sebebi sayılmaktadır. Rulet çarkı tekniği

uygulamada üreme işlemini başarıyla gerçekleştirmektedir.

Üreme işlemini gerçekleştirmek amacıyla Çizelge 6.1’de örnek bir popülasyon verilmiştir.

Çizelge 6.1 Üreme işlemine tabi tutulacak örnek bir popülasyon

Üye Sayısı : 15		
Çaprazlama Oranı : 0.30		
Mutasyon Oranı : 0.30		
ÜyeNo	Güç	Uygunluk
1	144.665	0.122
2	162.530	0.071
3	167.957	0.063
4	180.965	0.049
5	199.274	0.038
6	227.078	0.028
7	231.778	0.027
8	246.185	0.024
9	328.544	0.014
10	370.365	0.012
11	375.074	0.012
12	443.570	0.009
13	473.093	0.008
14	528.222	0.007
15	956.293	0.004

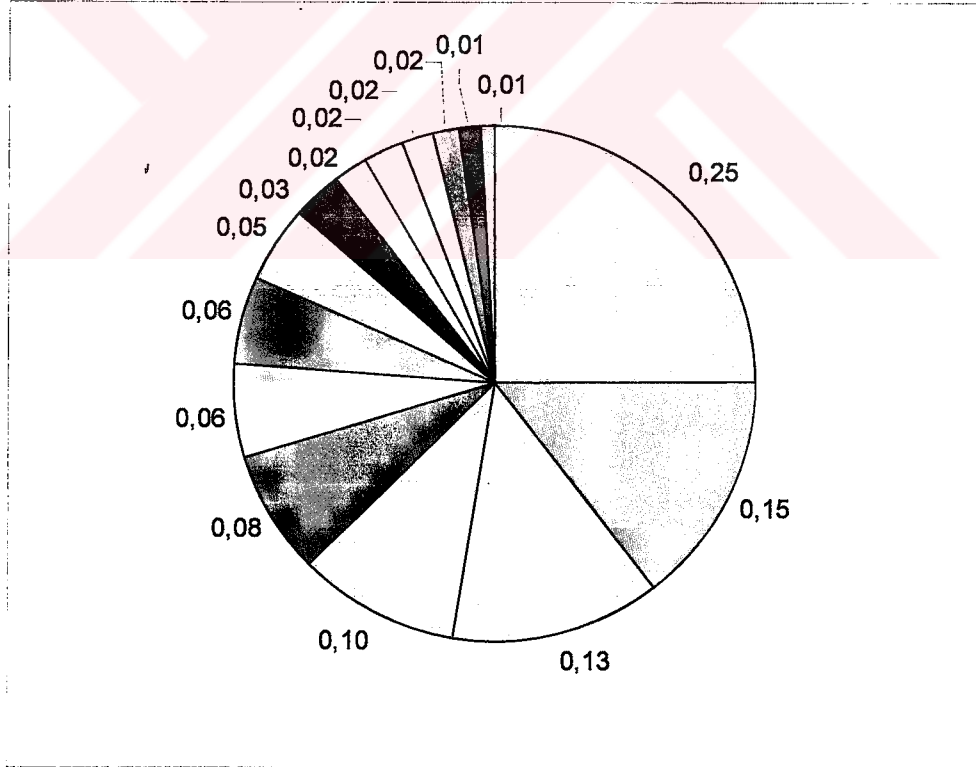
Çizelge 6.1’de 15 üyeli bir popülasyon uygunluk değerleri ile birlikte görülmektedir. Uygunluk değerleri son sütunda büyükten küçüğe doğru sıralanmış olarak verilmiştir. Üreme işleminde, uygunluk değerleri bilinen her üyenin sonraki nesilde bulunma ihtimalini veren değerlerin hesaplanması gerekmektedir. Bölüm 2’de belirtildiği gibi, üyelerin bir sonraki nesilde bulunma oranları, her üyenin uygunluk değerinin uygunluk değerleri toplamına oranı ile elde edilmektedir. Üyelerin bir sonraki nesilde bulunma ihtimallerini veren değerler Çizelge 6.2’de verilmiştir.

Bir rulet çarkı üzerine üyelerin bir sonraki nesilde bulunma yüzdeleri Şekil 6.4’te görüldüğü gibi yerleştirilir.

Üzerine üyelerin bir sonraki nesle aktarılma yüzdeleri orantısal olarak işaretlenmiş olan rulet çarkı, bir başlangıç noktasından 0 – 1 aralığında üretilen bir rasgele sayı kadar döndürülür. Üzerinde durulan üye bir sonraki nesle kopyalanır.

Çizelge 6.2 Popülasyon üyelerinin bir sonraki nesilde bulunma ihtimalleri

Üye No	Uygunluk	Yüzdesi
1	0,122	0,25
2	0,071	0,15
3	0,063	0,13
4	0,049	0,10
5	0,038	0,08
6	0,028	0,06
7	0,027	0,06
8	0,024	0,05
9	0,014	0,03
10	0,012	0,02
11	0,012	0,02
12	0,009	0,02
13	0,008	0,02
14	0,007	0,01
15	0,004	0,01
Toplam	0,488	1



Şekil 6.4 Üyelerin sonraki nesle kopyalanmalarına ilişkin rulet çarkı

Daha sonra rulet çarkı sıfırlanmak suretiyle ikinci kez döndürme işlemine geçilir. Yine aynı rassal sayı kadar döndürme işlemi gerçekleştirilir.

Döndürme işlemi, popülasyondaki üye sayısı kadar tekrarlanır. Her döndürme sonucunda bir üye yeni nesle kopyalanır. Böylece uygunluk değeri yüksek olan bireylerin daha fazla sayıda kopyasının bulunduğu yeni bir popülasyon üreme işlemi ile elde edilmiş olur.

Üreme işlemi sırasında bir nesilde elde edilen en yüksek uygunluk değerine sahip üyenin sonraki nesle direkt olarak kopyalanması, kimi literatürde “elitizm” olarak tanımlanmaktadır. Rulet çarkı, bu işlemi büyük bir olasılıkla üreme işlemi esnasında zaten gerçekleştirecektir. Fakat en iyi bireyin yeni nesle direkt kopyalanması, o ana kadar ulaşılan en iyi noktanın kaybedilme riskini ortadan kaldırır. Eğer en iyi birey yeni nesle aktarılmazsa, Genetik Algoritmanın rasgele arama özelliği daha ağır basarak, yakınsamanın daha geç gerçekleşmesi ihtimali doğar.

Uygulamada elitizm işlemi, uygunluk değeri en yüksek olan üyenin üreme işlemi sonucunda oluşan yeni popülasyon üyeleri arasında uygunluk değeri en düşük olan üye üzerine kopyalanması ile gerçekleşir. Böylece, her nesilde elde edilen en iyi üyenin bir sonraki nesle aktarılması kesin olarak sağlanmış olur.

6.5.2 Çaprazlama Operatörünün Uygulanması

Bir nesilden yeni bir nesle geçiş işleminin ikinci basamağını çaprazlama işlemi teşkil eder. Çaprazlama, üreme operatörü ile yeni nesle kopyalanmış üyelerden rasgele seçilen iki üye arasında gen takası yapılarak ortaya iki yeni üye çıkarılması işlemidir.

Genellikle, genetik zincir üzerinde rasgele seçilen bir noktadan sonraki bitlerin değişiminin gerçekleşmesi söz konusudur. Birden fazla nokta (örneğin seçilen iki nokta arasındaki genlerin takası) seçilerek genlerin takas edilmesi de mümkün olabilir, fakat bu işlemin gerçekleştirilmesi oldukça güçtür.

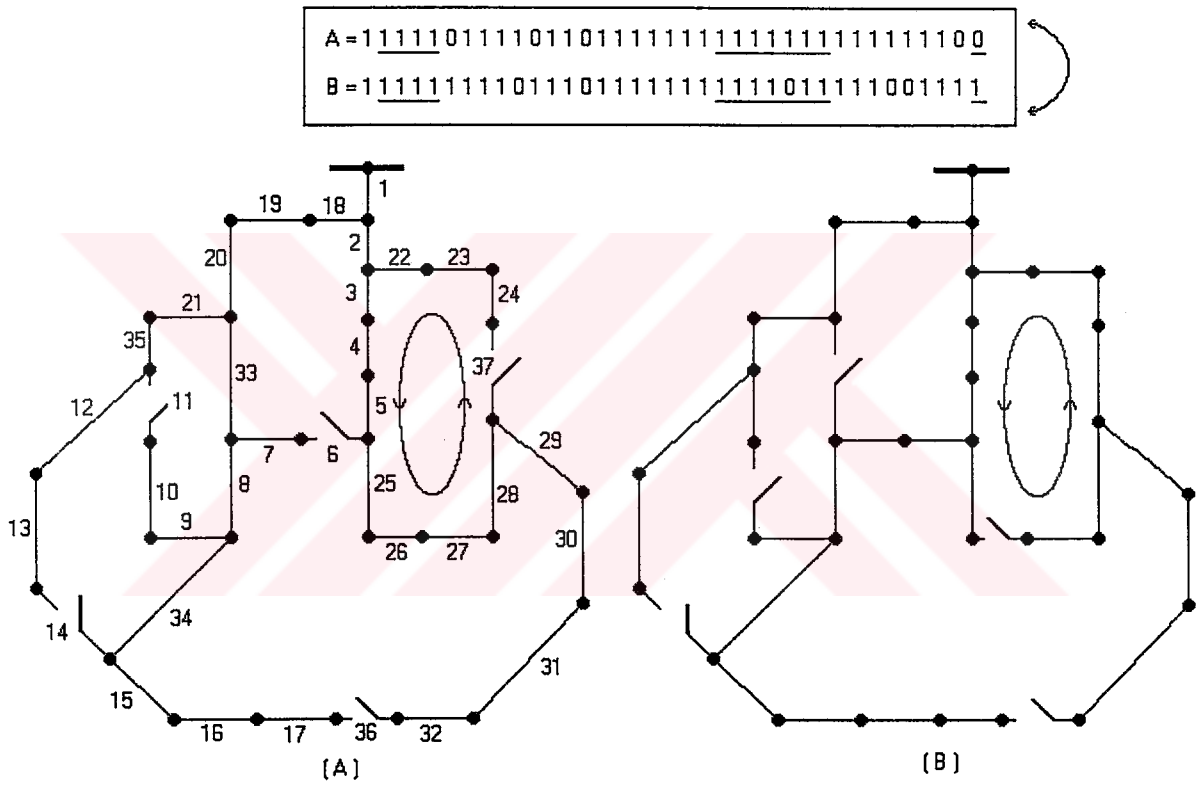
Her biri 37 bitlik bir gen zincirinden oluşan bireylerin bir noktalarından kesilerek çaprazlanmaları pek mümkün değildir. Çünkü 5 adet bağımsız çevre oluşturabilen enerji dağıtım sisteminde dallı yapının büyük olasılıkla kaybolması ya da bazı yüklerin enerjisi kalması söz konusu olabilir. Bu ise, hesaplama sürelerinin artmasına ve belki de istenilen sayıda çaprazlama işleminin gerçekleştirilememesine neden olabilir. Bu durum, Ek 3'ten detaylı olarak takip edilebilir.

Tek noktalı çaprazlama yerine uygulamada, enerji dağıtım sisteminin bağımsız kapalı

çevrelerinin kendi aralarında yer değiştirmelerini mümkün kılan, ilgili genlerin grupsal çaprazlanmaları mantıklı bir seçenektir.

Daha önce tanımlanan bağımsız kapalı çevrelerden bir tanesi rasgele seçilerek, iki üye üzerinde bu kapalı çevreye ait genler yer değiştirir. Grupsal çaprazlama işleminin sonucunda da bir geçerlilik testinin yapılması zorunludur. Eğer oluşan yeni üyelerden en az bir tanesi kısıtları sağlamaz ise bu çaprazlama işlemi tekrarlanır.

Grupsal çaprazlama işlemine örnek teşkil etmesi açısından, popülasyondan rasgele seçilen iki üye Şekil 6.5'te gösterilmiştir.



Şekil 6.5 Grupsal çaprazlama yapmak amacıyla rasgele seçilen iki üye ve sembollendikleri enerji dağıtım sistemleri

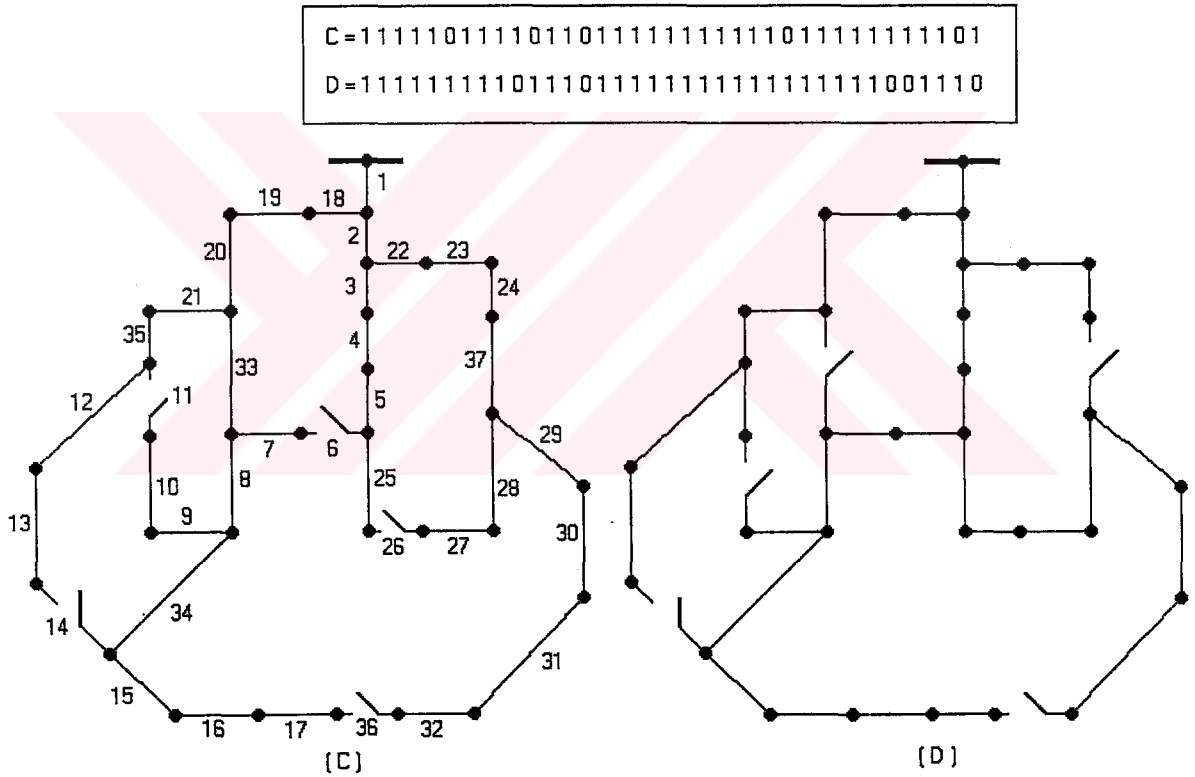
Çaprazlama yapmak amacıyla rasgele seçilen iki üye Şekil 6.5'te A ve B olarak isimlendirilmiştir. Bu üyelerin, ifade (6.1)'de tanımlanan birinci bağımsız kapalı çevreleri çaprazlama işlemi için rasgele seçilmiş ve eşdeğer dağıtım sistemleri üzerinde bu çevreler işaretlenmiştir. Aynı zamanda bu üyelere ait genetik zincirler üzerinde bu kapalı çevreyi oluşturan üyelerin altları çizilmiştir.

Çaprazlama, sadece altı çizili üyelerin grup olarak yer değiştirmeleri ile gerçekleşir. Böylece

bu iki ebeveyn üyenin genlerini taşıyan fakat onlarla bire bir benzerlik göstermeyen tamamen farklı iki yeni üye üretilmiş olur. Çaprazlama sonucu oluşan yeni üyeler C ve D olarak isimlendirilirse, onlara ait gen haritaları ve eşdeğer dağıtım şemaları Şekil 6.6'da gösterildiği gibi olur.

Çaprazlama işlemi iki üye arasında gerçekleşir, fakat bir popülasyonda kaç adet çaprazlama yapılacağı ya da kaç çiftin birbirleriyle çaprazlanacağını çaprazlama oranı belirler. Örneğin 0.30 olarak verilen çaprazlama oranı, popülasyondaki üyelerin genlerinin yüzde otuzunun çaprazlamaya maruz kalacağını ifade eder.

Uygulamada çaprazlama oranı değiştirilebilmektedir. Böylece popülasyonun daha hızlı yakınsamasını sağlayacak olan çaprazlama oranı tespit edilebilir.



Şekil 6.6 A ve B üyelerinin çaprazlanması sonucu oluşan yeni C ve D bireyleri

6.5.3 Mutasyon Operatörünün Uygulanması

Genetik Algoritmada yeni nesillerin oluşturulmasına çaprazlama operatöründen sonra mutasyon operatörü ile devam edilir.

Mutasyon operatörü, doğada genlerin mutasyona uğrayarak canlılarda farklılaşmalara ya da

atalarından çok farklı canlıların ortaya çıkmasına neden olmaları görüşünden esinlenilerek üretilmiştir. Popülasyonda rasgele seçilen bir üyenin rasgele seçilen bir geninin değiştirilmesi yani mutasyona uğratılması ile oluşur. Pratikte, genetik zincirde bir genin değerinin 0'dan 1'e ya da 1'den 0'a dönüştürülmesi işlemidir.

Bu çalışmada, çaprazlama operatöründe olduğu gibi mutasyon operatöründe de genetik zincir üzerinden rasgele seçilen bir genin değiştirilmesi sakıncalıdır. Çünkü popülasyondaki bütün üyeler 5 adet açık anahtara sahiptir ve genetik zincirleri buna göre düzenlenmiştir.

Her hangi bir genin rasgele değiştirilmesi, popülasyondaki açık anahtar sayısını değiştirir. Anahtar sayısı ya 4 ya da 6'ya değiştirilmiş olur. Her iki durumda da enerji dağıtım sistemi işletme şartları sağlanamaz.

Bundan sakınmak için mutasyon operatörü popülasyonda rasgele seçilen bir üyenin rasgele seçilen bir muhtemel bağımsız kapalı çevresine uygulanır. Üyeler için tanımlanan beş adet bağımsız kapalı çevreden bir tanesi rasgele seçilerek, bu çevredeki açık olan anahtar ya da anahtarlardan biri kapatılır ve yine aynı çevreden kapalı bir anahtarın açılması sağlanır. Böylece mutasyon işlemi uygulanmış olur. Ancak üyenin kısıtlara uyup uymadığı tespit edilerek gerektiğinde bu işlem yenilenir.

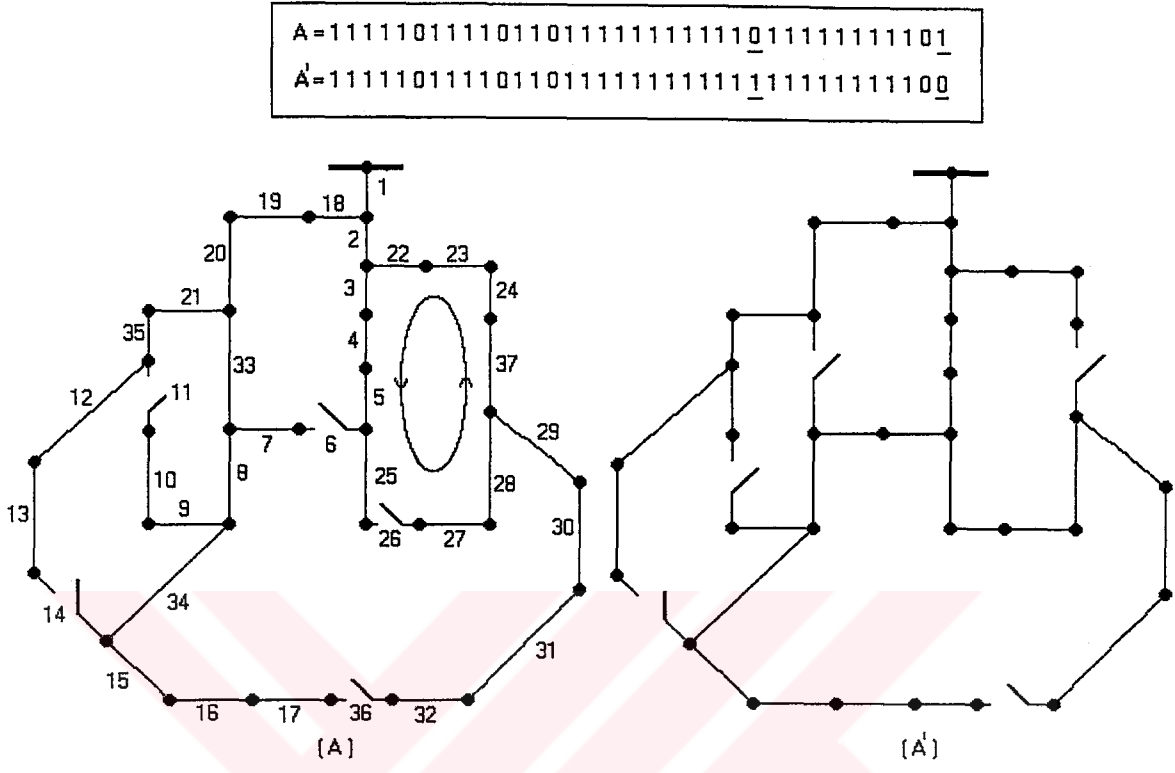
Şekil 6.7'de, verilen bir A üyesine mutasyon işleminin uygulanması gösterilmiştir. Bunun için rasgele seçilen A üyesinin rasgele seçilen bir bağımsız kapalı çevresinde açık konumda bulunan anahtar kapatılırken (0-1'e dönüştürülürken) aynı çevreden bir kapalı anahtar açılmaktadır (1-0'a dönüştürülür). A üyesi mutasyondan sonra A' üyesine dönüşmektedir. Böylece, enerji dağıtım sistemi işletme şartları gerçekleştirilirken yeni bir aday çözüm üretilmiş olur.

Mutasyon operatörünün bir popülasyona ne ölçüde uygulanacağını mutasyon oranı tayin eder. Yazılan programda mutasyon oranı değiştirilebilmektedir. Bu sayede en uygun mutasyon oranı ayarlanabilmektedir.

Gerçekte işlem, çaprazlama ve mutasyon oranlarının bir kombinasyonu üzerinde yapılır. Tek başına çaprazlama oranının yüksek tutulması popülasyonda erken yakınsama (pramature convergence) yani istenmeyen bir alt optimum noktaya ulaşmaya sebep olurken, tek başına mutasyon oranının yüksek bir değere ayarlanması ise popülasyonun rasgele arama özelliğini arttırarak yakınsamanın gecikmesine sebep olur.

Popülasyonun en kısa zamanda optimum noktaya ulaşmasını sağlamak için en uygun

çaprazlama-mutasyon kombinasyonu ile çalışmak gerekir.



Şekil 6.7 Uygulamada mutasyon işleminin gerçekleştirilmesine bir örnek

6.6 Enerji Dağıtım Sistem Verilerinin Genetik Algoritmada Değerlendirilmesi ve Sonuçların Elde Edilmesi

Uygulamada kullanılacak enerji dağıtım sistemine ait veriler Bölüm 4'te listelenmiştir. Fakat bu verileri kullanabilmek için şebekenin başlangıç şartlarının tanımlanması gerekmektedir.

Enerji dağıtım sisteminin başlangıç şebeke topolojisini teşkil edecek sistem olarak, Baran ve Wu'nun sadece bağlantı anahtarlarını açık bırakarak elde ettikleri ve Şekil 4.1'de gösterilen sistem seçilmiştir.

Başlangıç noktası olarak bu şebeke şeklinin seçilmesi, önceki çalışmalarla paralellik arz etmek amacıyla. Başlangıç noktası, pek çok çalışmada olduğu gibi bu çalışmada da çok büyük önem taşımaktadır. İyi bir başlangıç noktası, algoritmanın daha kısa sürede yakınsamasına sebep olabilir.

Tüm algoritma boyunca yük akış analizi yapmak oldukça külfetli bir işlemdir. Örneğin 100

nesil sürecek bir algoritmada her nesilde 15 üye üzerinde işlem yapıldığı varsayılırsa, bir algoritmada 1500 defa yük akış analizi yapmak gerekecektir. Bu ise hesaplama süreleri açısından etkin bir çözüm ortaya koymayı imkansız hale getirir.

Her üyeye yük akışı uygulamanın zorluğunu ortadan kaldırmak için, sadece ilk şebeke formasyonuna ve algoritma sonunda elde edilecek olan son şebeke formasyonuna yük akış analizi uygulanacak ve gerekli olması halinde (istenen yakınsama sağlanamaz ise) bir algoritma sonunda bulunan en iyi üye, şebekenin başlangıç şeklini oluşturacak şekilde algoritma tekrar başlatılacaktır.

Sabit yüklerden oluşan enerji dağıtım şebekesinde, yükler için eşdeğer sabit akım değerleri dikkate alınarak işlem yapılmıştır. Başlangıç şebekesinde sabit akım verilerini elde edebilmek amacıyla sistem yük akış analizine tabi tutularak bara gerilim değerleri elde edilmiş ve akım değerleri bulunmuştur. Yük akış analizini elde edebilmek amacıyla, Matlab tabanlı MATPOWER yük akış programı kullanılmıştır. Bu programın yük akış optimizasyonunu yapan “opf” dosyası Ek 4’te, “opf” dosyasını çalıştıran “runopf” dosyası Ek 5’te ve enerji dağıtım sistemi verilerinin girildiği “case33” dosyası Ek 6’da verilmiştir.

Çizelge 6.3 Başlangıç enerji dağıtım şebekesine ilişkin yük akış analizi sonuçları

Bara No	Bara Gerilimi (pu)	Bara Gerilimi (kV)	Bara No	Bara Gerilimi (pu)	Bara Gerilimi (kV)
1	0.997	12.62202	17	0.913	11.55858
2	0.983	12.44478	18	0.996	12.60936
3	0.975	12.3435	19	0.993	12.57138
4	0.968	12.25488	20	0.992	12.55872
5	0.95	12.027	21	0.992	12.55872
6	0.946	11.97636	22	0.979	12.39414
7	0.941	11.91306	23	0.973	12.31818
8	0.935	11.8371	24	0.969	12.26754
9	0.929	11.76114	25	0.948	12.00168
10	0.928	11.74848	26	0.945	11.9637
11	0.927	11.73582	27	0.934	11.82444
12	0.921	11.65986	28	0.925	11.7105
13	0.918	11.62188	29	0.922	11.67252
14	0.917	11.60922	30	0.918	11.62188
15	0.916	11.59656	31	0.917	11.60922
16	0.914	11.57124	32	0.917	11.60922

MATPOWER programıyla değerlendirilen başlangıç sistemine ilişkin yük akış analizi sonuçları Çizelge 6.3'te özetlenmiştir. Bu çizelgedeki bara gerilimleri yardımıyla hesaplanan başlangıç sistemi sabit yük akımları, Çizelge 6.4'te verilmiştir.

Çizelge 6.4 Başlangıç enerji dağıtım şebekesi sabit yük akımları

Bara No	Bara Akımı Aktif Bileşeni	Bara Akımı Reaktif Bileşeni	Bara No	Bara Akımı Aktif Bileşeni	Bara Akımı Reaktif Bileşeni
1	4.574151	2.744491	17	4.495494	1.997997
2	4.175367	1.855719	18	4.120869	1.831497
3	5.612835	3.74189	19	4.133319	1.837031
4	2.826712	1.413356	20	4.137486	1.838883
5	2.880271	0.96009	21	4.137486	1.838883
6	9.641498	4.820749	22	4.192427	2.329126
7	9.692728	4.846364	23	19.6853	9.373954
8	2.926478	0.975493	24	19.76656	9.412649
9	2.945379	0.981793	25	2.886347	1.202645
10	2.211415	1.474277	26	2.89551	1.206463
11	2.951734	1.721845	27	2.929612	0.976537
12	2.970963	1.733062	28	5.916232	3.451135
13	5.961345	3.97423	29	9.89247	29.67741
14	2.983923	0.49732	30	7.451681	3.477451
15	2.98718	0.995727	31	10.44373	4.973205
16	2.993717	0.997906	32	2.983923	1.989282

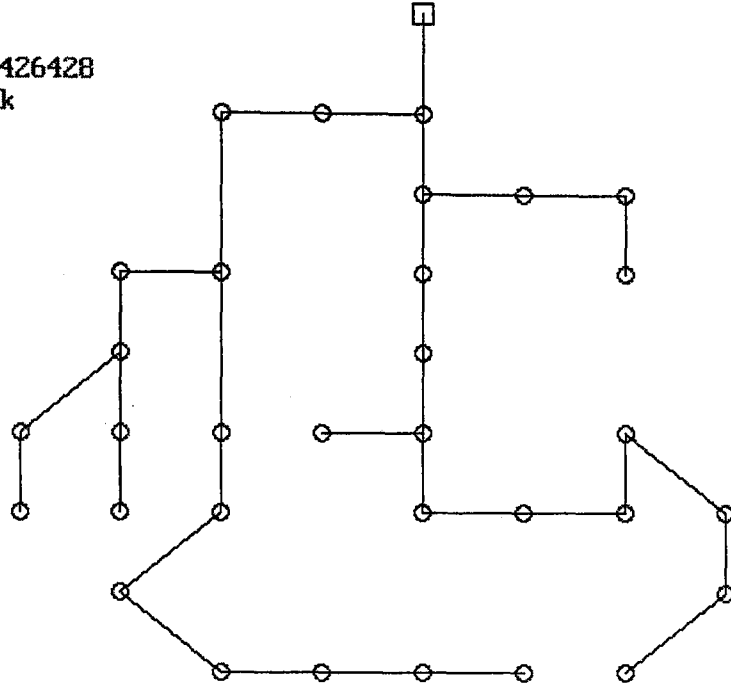
Enerji dağıtım sisteminin başlangıç şebeke formuna ait sabit akım verileri programa girilerek uygulama başlatılmıştır. Bu bölümde detaylı olarak anlatılan Genetik Algoritma operatörleri nesiller boyunca verilere uygulanarak, optimum çözüme yakınsama sağlanmıştır.

6.6.1 Elde Edilen Sonuçlar ve Değerlendirme

Genetik Algoritma yönteminin enerji dağıtım sistemine uygulanması esnasında çeşitli çaprazlama ve mutasyon oranları kullanılmıştır. Makul çaprazlama ve mutasyon oranlarının kullanıldığı bütün uygulamalarda algoritma optimum çözüme başarıyla yakınsamıştır. Bulunan en iyi çözüm, program ekranından aynen alınarak Şekil 6.8'de gösterilmiştir.

Nesil=56 Uygunluk=0.426428

ÜyeNo	Güç	Uygunluk
1	127.035	0.426
2	127.035	0.426
3	127.035	0.426
4	127.035	0.426
5	127.035	0.426
6	127.035	0.426
7	127.035	0.426
8	127.035	0.426
9	129.848	0.305
10	129.848	0.305
11	129.848	0.305
12	129.848	0.305
13	131.172	0.269
14	148.664	0.105
15	156.744	0.082



Şekil 6.8 Program ekranından aynen alınan optimum çözüm

Programda, üye numaraları birinci sütunda, bulunan toplam aktif hat kayıpları ikinci sütunda, buna bağlı olarak elde edilen uygunluk değerleri de üçüncü sütunda sıralı olarak gösterilmiş ve en iyi çözümü veren birinci üyeye ait enerji dağıtım sistemi şeması yan tarafa çizdirilmiştir.

Program bu sonuca ulaştıktan sonra, binlerce nesil daha ilerletildiği halde daha iyi bir çözüme ulaşamamıştır. Pek çok defalar program tekrar tekrar başlatılmış, fakat her birinde de en iyi çözüm olarak bu şebeke yapısına ulaşılmıştır.

Hesaplanan toplam kayıp güç değerleri güncellenmek zorundadır. Çünkü hesaplama, başlangıç sistemi verilerine göre yapılmıştır. Elde edilen bu en iyi çözüme yük akış analizi uygulanarak bu sisteme ait akım verileri hesaplanmalı ve toplam kayıp güç değeri bu akım değerlerine göre bulunmalıdır.

En iyi çözüme ait yük akış analizi sonuçları Çizelge 6.5'te ve bu yük akış analizinden elde edilen akım değerleri ise Çizelge 6.6'da gösterilmiştir.

Çizelge 6.5 Bulunan en iyi çözüme ait yük akış analizi sonuçları

Bara No	Bara Gerilimi (pu)	Bara Gerilimi (kV)	Bara No	Bara Gerilimi (pu)	Bara Gerilimi (kV)
1	0.997	12.622	17	0.947	11.989
2	0.987	12.4954	18	0.995	12.5967
3	0.982	12.4321	19	0.978	12.3815
4	0.978	12.3815	20	0.974	12.3308
5	0.967	12.2422	21	0.97	12.2802
6	0.967	12.2422	22	0.983	12.4448
7	0.963	12.1916	23	0.977	12.3688
8	0.959	12.1409	24	0.973	12.3182
9	0.963	12.1916	25	0.966	12.2296
10	0.963	12.1916	26	0.963	12.1916
11	0.963	12.1916	27	0.953	12.065
12	0.96	12.1536	28	0.945	11.9637
13	0.96	12.1536	29	0.942	11.9257
14	0.953	12.065	30	0.938	11.8751
15	0.951	12.0397	31	0.938	11.8751
16	0.949	12.0143	32	0.947	11.989

Bu en iyi çözüme ait akım değerleri Genetik Algoritmaya başlangıç değerleri olarak yeniden girilerek daha iyi bir çözümün olup olmadığı araştırılmalıdır. Daha iyi başka bir çözümün bulunmaması ve yine aynı çözüme yakınsamanın sağlanması halinde, bu çözüme ait toplam kayıp değeri hesaplanmış olacaktır. Eğer daha iyi bir çözüm mevcut ise, bu işlemler baştan tekrar edilmelidir.

Şekil 6.8'de gösterilen en iyi çözüme ait veriler Genetik Algoritmaya başlangıç değerleri olarak girilmiş ve elde edilen en iyi çözüm Şekil 6.9'da gösterilmiştir.

Şekil 6.9'da görüldüğü gibi, ilk algoritma sonucunda bulunan şebeke yine en iyi çözüm olarak elde edilmiştir.

Çok farklı başlangıç noktalarından hareket edilerek gerçekleştirilen uygulamaların hepsinde ulaşılabilen en iyi çözüm, Şekil 6.9'da gösterilen şebeke formasyonu olmuştur.

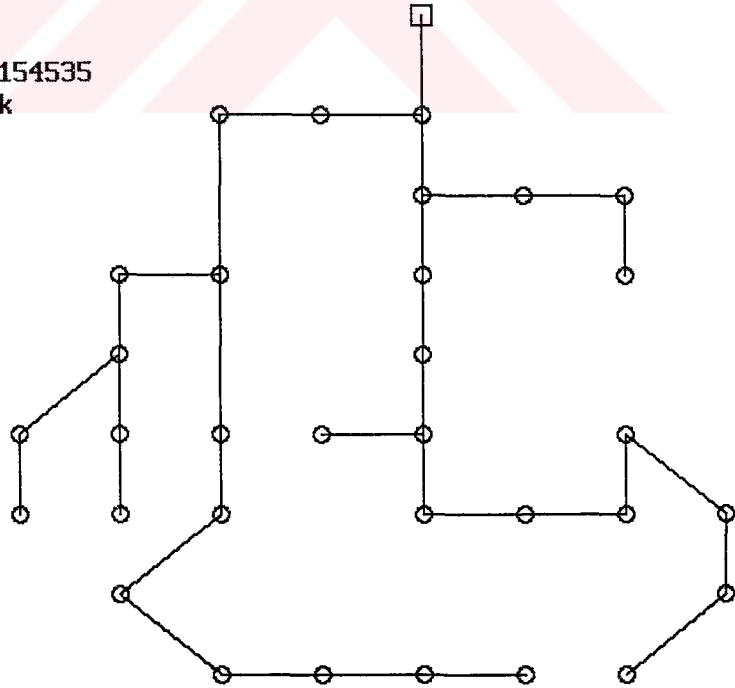
Genetik Algoritma ile enerji dağıtım sistemi üzerine yapılan yeniden yapılandırma çalışmasının en iyi çözümü olarak, daha önceki çalışmalarda da optimum çözüm olduğu ileri sürülen bu çözüm elde edilmiştir.

Çizelge 6.6 Bulunan en iyi çözüme ait akım değerleri

Bara No	Bara Akımı Aktif Bileşeni	Bara Akımı Reaktif Bileşeni	Bara No	Bara Akımı Aktif Bileşeni	Bara Akımı Reaktif Bileşeni
1	4.57415	2.74449	17	4.33409	1.92626
2	4.15845	1.8482	18	4.12501	1.83334
3	5.57283	3.71522	19	4.19671	1.86521
4	2.79781	1.3989	20	4.21395	1.87287
5	2.82964	0.94321	21	4.23133	1.88059
6	9.43212	4.71606	22	4.17537	2.31965
7	9.4713	4.73565	23	19.6047	9.33558
8	2.85324	0.95108	24	19.6853	9.37395
9	2.84139	0.94713	25	2.83256	1.18024
10	2.13104	1.42069	26	2.84139	1.18391
11	2.84139	1.65748	27	2.8712	0.95707
12	2.85027	1.66266	28	5.79102	3.3781
13	5.70054	3.80036	29	9.68244	29.0473
14	2.8712	0.47853	30	7.2928	3.4033
15	2.87724	0.95908	31	10.2099	4.86186
16	2.88331	0.9611	32	2.8894	1.92626

Nesil=51 Uygunluk=0.154535

ÜyeNo	Güç	Uygunluk
1	139.413	0.155
2	142.464	0.134
3	142.464	0.134
4	142.464	0.134
5	142.464	0.134
6	142.464	0.134
7	142.464	0.134
8	142.464	0.134
9	145.863	0.116
10	148.183	0.106
11	148.288	0.106
12	151.604	0.095
13	152.118	0.093
14	156.976	0.081
15	177.705	0.052



Şekil 6.9 İlk bulunan en iyi çözümü başlangıç noktası kabul eden ikinci uygulama algoritmasının en iyi çözümü

Enerji dağıtım sisteminde en iyi çözümün ne kadarlık bir aktif güç kaybıyla gerçekleştirildiği, hangi anahtarların açık bırakıldığı ve çözüme ilişkin Genetik Algoritma genetik zinciri Çizelge 6.7’de gösterilmiştir.

Çizelge 6.7 Optimum çözümü veren enerji dağıtım şebekesinde açık bırakılan anahtarlar ve toplam kayıp değeri

En İyi Çözümde Açık Kalan Anahtarlar	S7 S9 S14 S32 S37
İlk Şebeke Toplam Aktif Güç Kaybı	205,81 kW
Optimum Çözüm Toplam Aktif Güç Kaybı	139,41 kW
En iyi çözümü veren enerji dağıtım şebekesi genetik zinciri	
00000010100001000000000000000000100001	

6.6.1.1 Çaprazlama ve Mutasyon Oranlarının Çözüme Etkileri

Genellikle çaprazlama operatörü önce ve mutasyon operatörleri daha sonra olmak üzere sırasıyla genetik algorithmada yer alırlar. Sadece çaprazlama veya sadece mutasyonun etkilerini kavrayabilmek için Şekil 6.10 ve Şekil 6.11’e bakmak yeterli olacaktır.

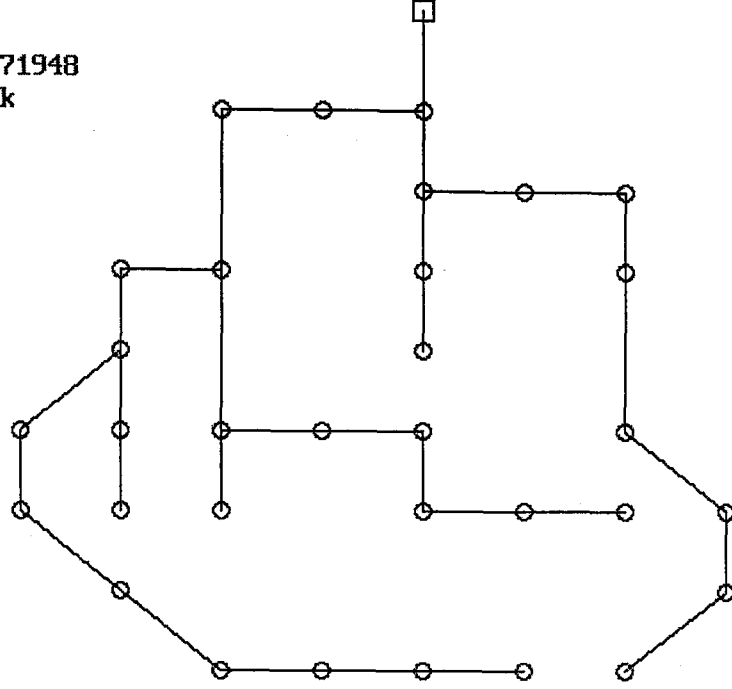
Şekil 6.10’da mutasyon oranı sifira eşitlenmiş ve çaprazlama oranı 0.50 olarak alınmış bir uygulama algoritmasının 1. ve 54. nesil sonundaki durumları görülmektedir. 1. nesil sonunda olması gerektiği gibi rasgele atanmış 15 üye, homojen bir uygunluk dağılımı göstermektedir.

54. nesil sonunda ise çaprazlama işlemleri ile elde edilebilen en iyi çözüme yakınsama sağlanmıştır. Bu çözümün ancak bir alt iyi çözüm olduğu ve popülasyondaki bütün üyelerin aynı değere yakınsadıkları görülmektedir. Bu algoritma isterse binlerce nesil devam ettirilsin, sadece çaprazlama işlemi ile bu noktadan öteye gidemeyeceği açıktır.

Sadece çaprazlama işleminin uygulanması ile Genetik Algoritmanın vaktinden önce bir alt çözüme yakınsaması kaçınılmazdır. Gerçek en iyi çözüme bu şartlar altında ulaşmak çok küçük bir olasılıktır.

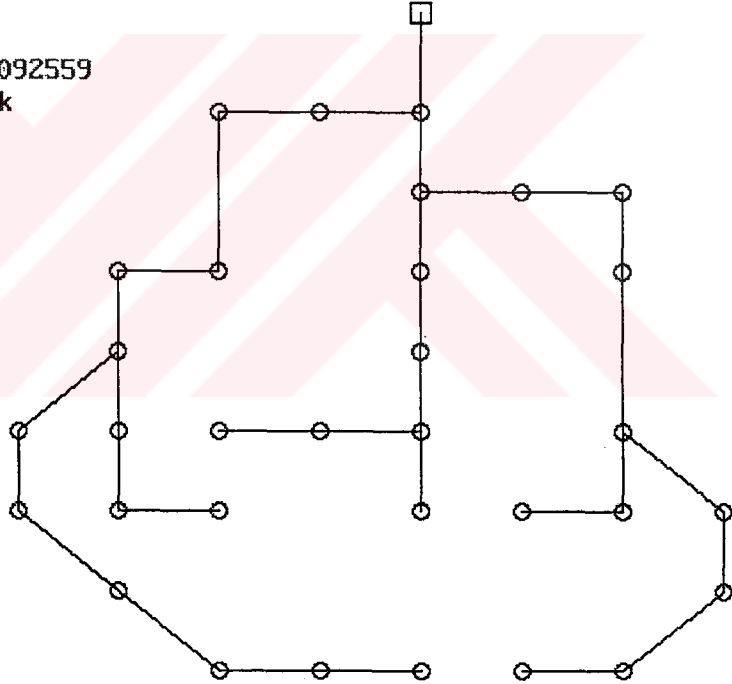
Nesil=1 Uygunluk=0.071948

ÜyeNo	Güç	Uygunluk
1	161.697	0.072
2	168.370	0.062
3	185.047	0.046
4	185.047	0.046
5	187.631	0.044
6	187.631	0.044
7	188.621	0.044
8	192.960	0.041
9	192.960	0.041
10	198.436	0.038
11	198.787	0.038
12	204.209	0.036
13	235.671	0.026
14	261.767	0.021
15	342.312	0.013



Nesil=54 Uygunluk=0.092559

ÜyeNo	Güç	Uygunluk
1	152.412	0.093
2	152.412	0.093
3	152.412	0.093
4	152.412	0.093
5	152.412	0.093
6	152.412	0.093
7	152.412	0.093
8	152.412	0.093
9	152.412	0.093
10	152.412	0.093
11	152.412	0.093
12	152.412	0.093
13	152.412	0.093
14	152.412	0.093
15	152.412	0.093



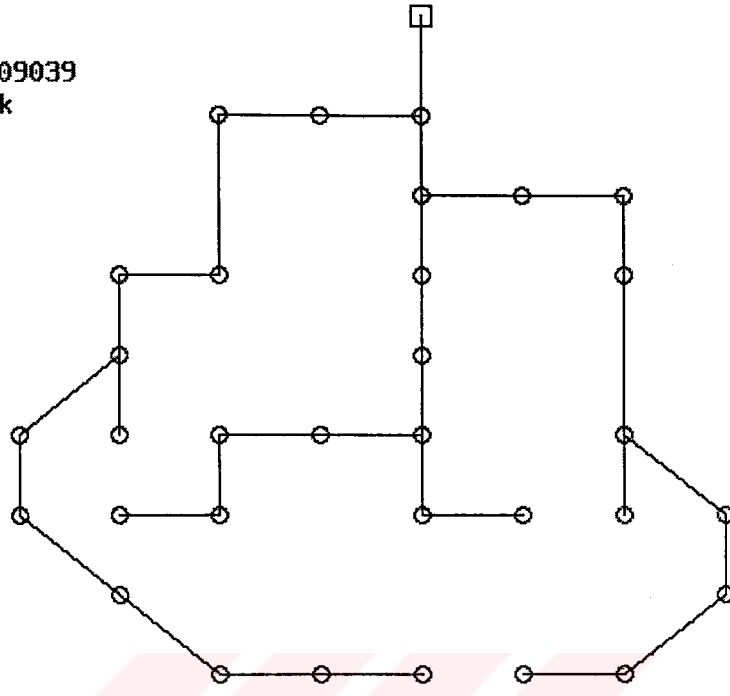
Şekil 6.10 Sadece çaprazlama işlemi uygulanan algoritma

Şekil 6.11'de çaprazlama oranı sıfır ve mutasyon oranı 0.50 yapılarak uygulanan bir algoritmanın 1. ve 1095. nesilleri örnek olarak gösterilmiştir. Birinci nesilde üyeler rasgele üretilerek birbirinden farklı 15 üye elde edilmiştir. Çaprazlama işlemi yapılmayıp sadece mutasyon işlemi uygulanmasından dolayı 1095. nesilde hala çözüme ulaşamamıştır. Bu algoritmanın gerçek en iyi çözüme yakınsaması muhtemeldir, fakat bu biraz daha zaman

alacaktır.

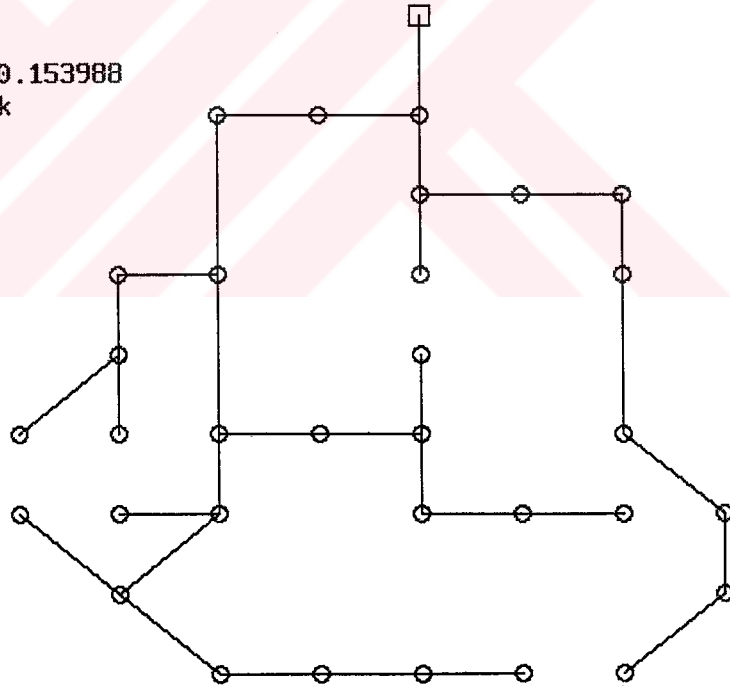
Nesil=1 Uygunluk=0.109039

ÜyeNo	Güç	Uygunluk
1	147.513	0.109
2	159.525	0.076
3	165.090	0.067
4	210.222	0.033
5	219.724	0.030
6	221.330	0.030
7	221.330	0.030
8	241.300	0.025
9	264.779	0.021
10	282.361	0.018
11	300.812	0.017
12	300.812	0.017
13	347.948	0.013
14	365.644	0.012
15	451.988	0.009



Nesil=1095 Uygunluk=0.153988

ÜyeNo	Güç	Uygunluk
1	139.482	0.154
2	165.648	0.066
3	176.253	0.053
4	177.705	0.052
5	177.705	0.052
6	177.705	0.052
7	177.705	0.052
8	182.170	0.048
9	182.170	0.048
10	182.170	0.048
11	182.170	0.048
12	182.170	0.048
13	182.170	0.048
14	182.170	0.048
15	292.236	0.017

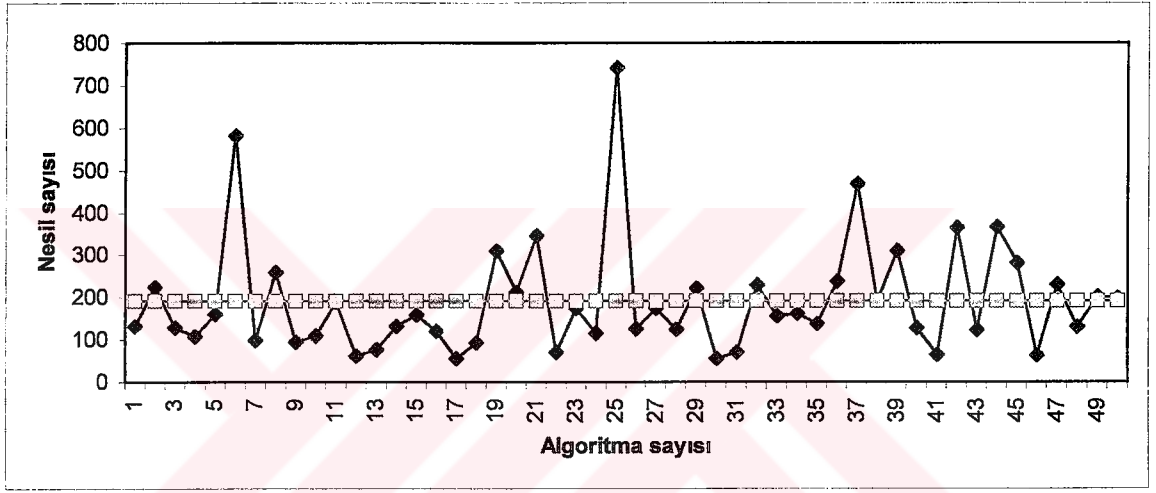


Şekil 6.11 Sadece mutasyon işlemi uygulanan algoritma

Tek başına mutasyon operatörü algoritmaya rasgele arama özelliği katmaktadır. Eğer kopyalama da yapılmamış olsa algoritmanın sadece mutasyon operatörüyle çözüme ulaşması tamamen tesadüf eseri olacaktır. Sadece mutasyon operatörü ile algoritmanın en iyi çözüme

ulaşması olasıdır, fakat bu işlem muhtemelen çok nesillerden sonra gerçekleşir.

Çaprazlama ve mutasyon operatörleri algoritmaya birlikte uygulanmalıdır. Genetik Algoritmanın en hızlı yakınsaması için gerekli olan çaprazlama ve mutasyon oranı kombinasyonunun belirlenmesi bile başlı başına bir istatistiksel optimizasyon işlemidir. Hangi oranlarda çaprazlama ve mutasyon uygulandığında en iyi çözüme ulaşılabileceği yaklaşık olarak tahmin edilebilir, fakat kesin olarak önceden bilinemez. En iyi çözümü farklı iki çaprazlama ve mutasyon oranı kombinasyonları ile gerçekleştiren Genetik Algoritmalarda çözüme yakınsama dağılımı ve ortalama yakınsama eğrilerine iki örnek, Şekil 6.12 ve Şekil 6.13'te verilmiştir.

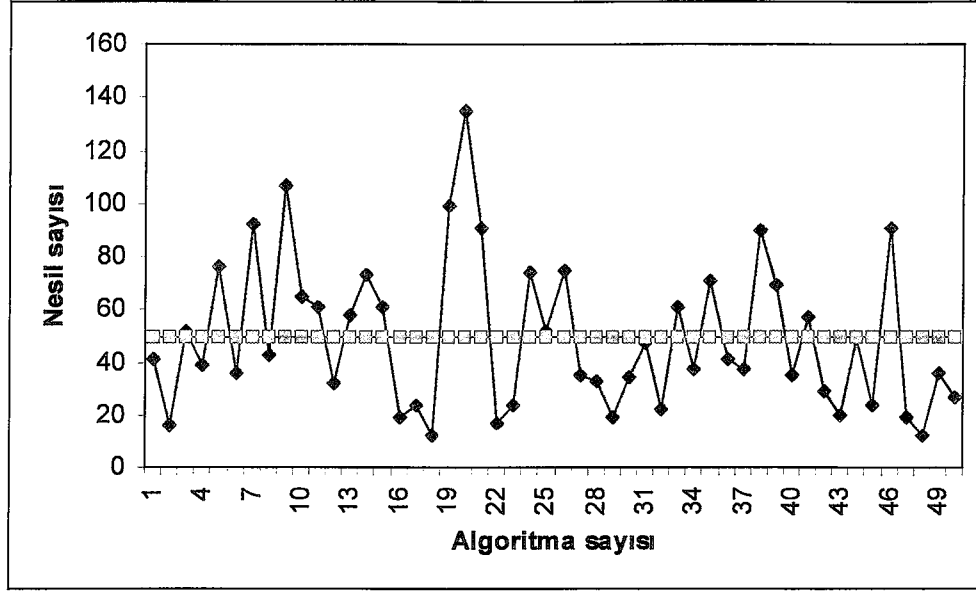


Şekil 6.12 0.10 çaprazlama ve 0.10 mutasyon oranının uygulandığı algoritmaların çözüme yakınsama dağılımı ve ortalama yakınsama eğrisi

Şekil 6.12'de, 0.10 çaprazlama ve 0.10 mutasyon oranı kullanılarak 50 farklı algoritmanın çözüme kaçınıcı nesilde yakınsadıkları gösterilmiştir.

Düz bir hattan oluşan ikinci grafik ise 50 adet algoritmanın çözüme yakınsama ortalama nesil sayısını göstermek amacıyla çizdirilmiştir. Düşük çaprazlama ve mutasyon oranının kullanıldığı bu algoritmalarda en iyi çözüme yakınsama ortalama nesil sayısı yaklaşık olarak 191 nesil olarak elde edilmiştir.

Şekil 6.13'de ise, 0.40 çaprazlama ve 0.40 mutasyon oranının kullanıldığı algoritmalar ele alınmıştır. Çözüme yakınsama ortalama nesil sayısının yaklaşık 49 nesil olarak hesaplandığı algoritmalar ile probleme daha hızlı çözüm bulmak mümkün olmuştur.



Şekil 6.13 0.40 çaprazlama ve 0.40 mutasyon oranının uygulandığı algoritmaların çözüme yakınsama dağılımı ve ortalama yakınsama eğrisi

Şekil 6.12 ve Şekil 6.13'den görüldüğü üzere, çözüme yakınsama ortalamaları olarak iki durum arasında yaklaşık 4 kat fark vardır. Doğrusal olarak çözüme yakınsama süreleri arasında da yaklaşık 4 kat fark olacaktır.

Ayrıca çaprazlama ve mutasyon oranlarının 0.10 olarak alındığı birinci durumda çözüme 56.-742. nesiller aralığında yakınsama sağlanırken, ikinci durumda ise bu aralık 12.-135. nesiller aralığında gerçekleşmiştir.

Birinci durumda çözüme yakınsama çok geniş bir aralıkta, ikinci durumda ise çok daha dar bir aralıkta ancak daha kararlı olarak gerçekleşmiştir. Bu iki örnek, farklı çaprazlama ve mutasyon oranları kullanılması halinde algoritmaların çözüme yakınsamalarındaki farklılıkların anlaşılması açısından önemlidir. Genel olarak çaprazlama oranı için 0.40-0.60 aralığı uygun görülürken, mutasyon oranı için ise 0.30-0.50 aralığında değerlerin seçilmesi uygun bulunmaktadır.

6.6.1.2 Sonuçların Literatürdeki Çalışmalarla Karşılaştırılması

Daha önceki çalışmalarda elde edilen en iyi çözümlerle yapılan karşılaştırma Çizelge 6.8'de verilmiştir.

Çizelge 6.8'de Baran tarafından önerilen üç metot B1, B2, B3 ve Goswami tarafından önerilen üç farklı metot da G1, G2, G3 olarak gösterilmiştir. Üçüncü satırda, Optimum

çözümü veren şebeke şekillerine ilişkin açık anahtar numaraları ve dördüncü satırda ise optimum çözümü veren şebekeye ulaşıncaya kadar tasarruf edilen enerji kaybının yüzdelik değeri verilmiştir.

Bu tezde önerilen Genetik Algoritma metodunun optimum çözümü bulmada oldukça etkin bir yöntem olduğu Çizelge 6.8'den anlaşılmaktadır. Şu ana kadar yapılan bütün çalışmalarda bulunan en iyi çözüm, bu çalışmada da ulaşılan S7, S9, S14, S32 ve S37 anahtarlarının açık konumda kaldığı duruma ilişkin olan dallı şebeke yapısıdır.

Çizelge 6.8 Literatürde aynı problemi ele almış olan çalışmalarla bir karşılaştırma çizelgesi

Baran Metodu			Goswami Metodu			Fan Metodu	Shirmohammadi Metodu	Kashem Metodu	Bu Tezde Önerilen Metod
B1	B2	B3	G1	G2	G3				
6	6	6	7	7	7	7	7	6	7
11	11	11	9	9	9	9	10	9	9
28	31	31	14	14	14	14	14	14	14
31	34	34	32	32	32	32	32	32	32
33	37	37	37	37	37	37	37	37	37
27,8	23,8	23,8	32,2	32,2	32,2	32,2	31,6	30,8	32,2

6.7 Sonuçlar ve Öneriler

Çalışmada, dallı enerji dağıtım sisteminin lineer olmayan aktif güç kayıplarının modellenmesi gerçekleştirilerek enerji dağıtım sisteminin toplam aktif güç kayıplarını minimum yapacak yeni şebeke şekillerinin bulunması sağlanmıştır. Genetik algoritma metodu, bu işlevi gerçekleştirecek araç olarak kullanılmış, C++ programlama dilinde geliştirilen yazılım ile bir IEEE referans çalışmasına ilişkin enerji dağıtım sistemi ve bu sisteme ait veriler test edilmiştir. Gerektiğinde güç akış analizi yapmak üzere MATPOWER güç akış programı kullanılmıştır.

Genetik algoritma yönteminin kullanıldığı süreçte, aktif hat kayıpları minimum yapılarak enerji dağıtım sisteminin yeniden yapılandırılması gerçekleştirilmiş, dallı enerji dağıtım

sisteminin parametreleri kullanılarak bu güne kadar yapılan çalışmalarda elde edilen en iyi çözümlere ulaşılmıştır.

Genetik algoritma yönteminin optimizasyon problemlerini çözmekteki başarısı ve geçerliliği sonuçların karşılaştırılmasıyla bir kez daha kanıtlanmıştır. Önerilen yöntemin güvenilir ve etkin sonuçları, sadece enerji dağıtım sistemlerinin işletilmesi aşamasında değil, aynı zamanda planlama çalışmalarında da genetik algoritmaların kullanılabileceğini göstermektedir.

Değişkenler yerine parametrelerle çalışan yapısı, genetik algoritmalara diğer pek çok sayısal yöntemle kıyasla matematiksel işlemlerin azaltılmasını sağlarken, diğer yandan, genellikle daha geniş bir çözüm uzayında araştırma yapmayı sonuç vermektedir.

Ayrıca, genetik algoritmaların olasılık temelli yapısı yanında rasgele arama özelliği, Baran ve Shirmohammadi'nin ileri sürdükleri metotlarda olduğu gibi, diğer bir kısım çalışmalarda sergilenen alt optimum noktalara takılıp optimum çözümlerin kaçırılması riskini de ortadan kaldırmıştır.

Genetik algoritmalar bir tek noktadan hareket ederek çözüme ulaşmak yerine pek çok aday çözümle aynı anda işlem yapmaktadır. Böylece enerji dağıtım sistemi yeniden yapılandırılırken, anahtarların ya da kapalı çevrelerin teker teker dikkate alınması gibi bir zorunlulukları yoktur.

Genetik algoritmalar, enerji dağıtım sistemlerinin işletmesinde yüksek çalışma frekanslı bilgisayarlar, veri kazanım sistemleri ve uzaktan kontrollü anahtarlarla birlikte kullanılarak güç kaybı tasarrufu veya arızalardan sonra sistemin tekrar enerjilenmesi gibi konulara gerçek zamanlı çözümler üretmekte kullanılabilir.

KAYNAKLAR

- Anthony, B.M., Mareels, I.M.Y., (2000), "An Efficient Brute-Force Solution to the Network Reconfiguration Problem", *IEEE Transactions on Power Delivery*, Vol. 15, No. 3, July, pp. 996-1000.
- Aoki, K., Ichimori, T., Kanezashi, M., (1987), "Normal State Optimal Load Allocation in Distribution Systems", *IEEE Trans. On PAS*, Vol. PWRD-2, January, p.p. 147-155.
- Back, T., (1996), "Evolutionary Algorithms in Theory and Practice", Oxford University Press, ISBN 0-19-509971-0.
- Baluja, S., (1995), "An Empirical Comparison of Seven Iterative and Evolutionary Function Optimization Heuristics, Technical Report CMU-CS-95-193, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.
- Baran, M.E., Wu, F.F., (1989), "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing", *IEEE Transactions on Power Delivery*, Volume 4, No. 2, April, pp. 1401-1407.
- Castro, C.H., Bunch, J.B., Topka, T.M., (1980), "Generalized Algorithms for Distribution Feeder Deployment and Sectionalizing", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS 99, March, p.p. 549-557.
- Chaiyaratana, N. and Zalzal, A.M.S., (1997), "Recent Developments in Evolutionary and Genetic Algorithms: Theory and Applications", 2nd IEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, No. 446, pp. 270-277, September, Glasgow.
- Chen, C.S., Cho, M.Y., (1992), "Determination of Critical Switches in Distribution Systems", *IEEE Transactions on Power Delivery*, Vol. 7, No 3, pp. 1443-1448.
- Chen, C.S., Cho, M.Y., (1993), "Energy Loss Reduction by Critical Switches", *IEEE Transactions on Power Delivery*, Vol. 8, No 3, July, p.p. 1246-1253.
- Civanlar S., Grainger, J.J., Yin, H., Lee, S.S.H., (1988), "Distribution Feeder Reconfiguration for Loss Reduction", *IEEE Transactions on Power Delivery*, Volume 3, No. 3, July, pp. 1217-1223.
- Curcic, S., Strbac, G., Zhang, X.P., (2001), "Effect of Losses in Design of Distribution Circuits", *IEE Proc.-Gener. Transm. Distrib.*, Vol.148, No. 4, July, p.p. 343-349.
- Davis, R., (1982), "Expert systems: where are we? And where do we go from here?" , *Artificial Intelligence*, pp. 3-22, Spring.
- De Moura Oliveira, P.B., (1998), "Evolutionary Design of Process Control Systems", Ph. D. Thesis, Chapter 1, University of Salford, Department of Aeronautical and Mechanical Engineering.
- Fan, J.Y., Zhang, L., McDonald, J.D., (1996), "Distribution Network Reconfiguration: Single Loop Optimization", *IEEE Transactions on Power Systems*, Volume 11, No. 3, August, p.p. 1643-1647.
- Fawzi, T.H., Ali, K.F., El-Sobki, S.M., (1983), "A New Planning Model for Distribution Systems", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 9, September, pp. 3010-3017.

- Gen, M., Cheng, R., (1997), "Genetic Algorithms and Engineering Design", John Wiley and Sons Inc.
- Goldberg, D.E., (1989), "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley P.C., ISBN 0-201-15767-5.
- Goldberg, D.E., Korb, B. and Deb, K., (1989), "Messy Genetic Algorithms: Motivation, Analysis and First Results", Complex Systems, Vol. 3, pp. 493-530.
- Goswami, S.K., Basu, S.K., (1992), "A New Algorithm for The Reconfiguration of Distribution Feeders for Loss Minimization", IEEE Transactions on Power Delivery, Volume 7, No. 3, July, p.p. 1484-1491.
- Greene, J.R., (1997), "A Role for Simple, Robust 'Black-Box' Optimisers in the Evolution of Engineering Systems and Artefacts", 2nd IEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, No. 446, pp. 427-432, September, Glasgow.
- Holland, J.H., (1975), "Adaptation in Natural and Artificial Systems", 1st MIT Press edition, ISBN: 0-262-58111-6.
- Hsiao, Y.T., Chien, C.Y., (2001), "Multiobjective Optimal Feeder Reconfiguration", IEE Proc.-Gener. Transm. Distrib., Vol. 148, No. 4, July, p.p. 333-3336.
- IEEE Transactions on Power Apparatus and Systems, (1984), "Bibliography on Distribution Automation", June, p.p. 1176-1182.
- Jonnavithula, S., Billington, R., (1996), "Minimum Cost Analysis of Feeder Routing in Distribution System Planning", IEEE Transactions on Power Delivery, Volume 11, No. 4, October, pp. 1935-1940.
- Kashem, M.A., Jasmon, G.B., Ganapathy, V., (2000), "A New Approach of Distribution System Reconfiguration for Loss Minimization", International Journal of Electric Power and Energy Systems, Volume 22, Issue 4, May, p.p. 269-276.
- Kashem, M.A., Ganapathy, V., Jasmon, G.B., (2001), "A Geometrical Approach for Network Reconfiguration Based Loss Minimization in Distribution Systems", International Journal of Electric Power and Energy Systems, Volume 23, Issue 4, May, p.p. 295-304.
- Koza, J.R., (1992), "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press, A Bradford book, ISBN: 0-262-11170-5, pp. 429-437.
- Kushner, H.J., (1967), "Stochastic Stability and Control", Academic Press.
- Langton, C.G., (1995), "Artificial Life, The Philosophy of Artificial Life", Boden M A editor, Chapter 1, pp. 34-94, pp. ISBN: 0-19-875154-0.
- Lin, W.M., Chin, H.C., (1998), "A New Approach for Distribution Feeder Reconfiguration for Loss Reduction and Service Restoration", IEEE Transactions on Power Delivery, Volume 13, No. 3, July, p.p. 870-875.
- Liu, C.C., Lee S.J., Venkata, S.S., (1988), "Loss Minimization of Distribution Feeders: Optimality and Algorithms", IEEE Transactions on Power Delivery, Volume 3, pp. 619-625.
- Liu, C.C., Lee S.J., Vu, K., (1989), "Loss Minimization of Distribution Feeders: Optimality and Algorithms", IEEE Transactions on Power Delivery, Volume 4, No 2, April, pp. 1281-

1289.

Lyons, P.C., Thomas, S.A., (1981), "Microprocessor Based Control of Distribution Systems", IEEE Transactions on Power Apparatus and Systems, December, pp. 4893-4899.

Man, K.F., Tang, K.S. and Kwong, S., (1996), "Genetic Algorithms: Concepts and Applications", IEEE Trans. Of Industrial Electronics, Vol. 43, No. 5, pp. 519-533.

McCall, L.V., Chamber, B.J., (1983), "Defining a Distribution System for Computer-Controlled Distribution Automation", IEEE Transactions on Power Apparatus and Systems, August, pp. 2665-2669.

Merlin, A., Back, H., (1975), "Search for a Minimal-Loss Operating Spanning Tree Configuration in Urban Power Distribution Systems", Proc. Of 5 th Power System Comp. Con., Cambridge, U.K., September 1-5, pp. 2-6.

Morton, A.B., Mareels, M.Y., (2000), "An Efficient Brute-Force Solution to the Network Reconfiguration Problem", IEEE Transactions on Power Delivery, Volume 15, No. 3, July, p.p. 996-1000.

Overbye, T.J., Klump, R.P., (1996), "Effective Calculation of Power System Low-Voltage Solutions", IEEE Transactions on Power Systems, Volume 11, No. 1, February, pp. 75-82.

Peponis, G.J., Papadopoulos, M.P., Hatziargyriou, N.D., (1996), "Optimal Operation of Distribution Networks", IEEE Transactions on Power Systems, Volume 11, No. 1, February, pp. 59-67.

Ponnaivaikko, M., Prakasa Rao, K.S., (1982), "An Approach to Optimal Distribution System Planning Through Conductor Gradation", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 6, June, pp. 1735-1742.

Price, W.L., (1983), "Global Optimization by Controlled Random Search", Journal of Optimization Theory and Applications, Vol. 40, No. 3.

Prucker, S.L., Reddoch, T.W., Detwiler, J.S., Monteen, L.D., (1985), "The Design of an Integrated Distribution Control System", IEEE Transactions on Power Apparatus and Systems, March, pp. 745-752.

Rosado, I.J.R., Gönen, T., (1991), "Pseudodynamic Planning for Expansion of Power Distribution Systems", IEEE Transactions on Power Systems, Volume 6, No. 1, February, pp. 245-254.

Ross, D.W., Carson, M., Cohen, A., (1980), "Development of Advanced Methods for Planning Electric Energy Distribution Systems", DEO final report no SCI-5263, February 1980.

Roytelman, I., Melnik, V., Lee, S.S.H., Lugtu, R.L., (1996), "Multi-Objective Reconfiguration by Distribution Management System", IEEE Transactions on Power Systems, Volume 11, No. 2, May, pp. 661-667.

Rutenbar, R.A., (1989), "Simulated Annealing An Overview", IEEE Circuits and Devices, January, pp. 19-26.

Sarfi, R.J., Salama, M.M., Chikhani, A.Y., (1994), "A Survey of the State of the Art in Distribution System Reconfiguration for System Loss Reduction", Electr. Power Syst. Res., pp. 61-70.

Shirmohammadi, D., Hong, H.W., (1989), "Reconfiguration of Electric Distribution Network for Resistive Line Losses Reduction", IEEE Transactions on Power Delivery, Volume 4, No. 2, April, p.p. 1492-1498.

Slamocanin V., (1990), "Optimal Loss Reduction of Distribution Networks", IEEE Transactions on Power Systems, Volume 5, No. 3, August, pp. 774-780.

Song, Y.H., Wang, G.S., Johns, A.T., Wang, P.Y., (1997), "Distribution Network Reconfiguration for Loss Reduction Using Fuzzy Controlled Evolutionary Programming", IEE Proc.-Gener. Transm. Distrib., Vol. 144, No. 4, July, p.p. 345-350.

Spears, W.M., (1992), "Crossover or Mutation?", 2nd Workshop on Foundations of Genetic Algorithms (FOGA), Morgan Kaufmann Publication, pp. 221-237, ISBN 1-55860-263-1.

Sun, D.I., Farris, D.R., Cote, P.J., Shoults, R.R., Chen, M.S. (1982), "Optimal Distribution Substation and Primary Feeder Planning via the Fixed Charge Network Formulation", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, March, p.p. 602-609.

Tang, Y., (1996), "Power Distribution System Planning with Reliability Modeling and Optimization", IEEE Transactions on Power Systems, Vol. 11, No. 1, February, pp. 181-187.

Taylor, T., Lubkeman, D., (1990), "Implementation of Heuristic Search Strategies for Distribution Feeder Reconfiguration", IEEE Transactions on Power Delivery, Volume 5, p.p. 239-246.

Tram, H.N., Wall, D.L., (1988), "Optimal Conductor Selection in Planning Radial Distribution Systems", IEEE Transactions on Power Systems, Vol. 3, No. 1, February, pp. 200-206.

Wagner, T.P., Chikhani, A.Y., Hackam, R., (1991), "Feeder Reconfiguration for Loss Reduction: An Application of Distribution Automation", IEEE Transactions on Power Delivery, Volume 6, No. 4, April, pp. 1922-1931.

Wang, Z., Liu, H., Yu, D.C., Wang, X., Song, H., (2000), "A Practical Approach to the Conductor Size Selection in Planning Radial Distribution Systems", IEEE Transactions on Power Delivery, Volume 15, No. 1, January, pp. 350-354.

INTERNET KAYNAKLARI

[1] www.pserc.cornell.edu/matpower/matpower.html

[2] www.aic.nrl.navy.mil/galist/

[3] www-illigal.ge.uiuc.edu/index.php3

[4] <http://lancet.mit.edu/ga/>

[5] <http://garage.cps.msu.edu/>

[6] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html

[7] <http://www.cs.colostate.edu/~genitor/>

EKLER

- Ek 1 Örnek enerji dağıtım sistemine yeniden yapılandırma probleminin genetik algoritma yöntemi ile uygulandığı C++ programlama dilinde yazılan program
- Ek 2 İlk popülasyonda bile uygunluk değerlerinin birbirine ne kadar yakın olduğunu gösteren çizelge
- Ek 3 Tek noktalı çaprazlamanın sakıncalı yanlarının şekil üzerinde görsel olarak gerçekleştirilmesi
- Ek 4 MATPOWER optimizasyon programının yük akış analizi yapan “opf” dosyası.
- Ek 5 MATPOWER paket programında yük akış dosyası “opf”yi çalıştıran “runopf” dosyası
- Ek 6 MATPOWER programında yük akış analizi yapılmak amacıyla enerji dağıtım sistem verilerinin girildiği dosya



Ek 1 Örnek enerji dağıtım sistemine yeniden yapılandırma probleminin genetik algoritma yöntemi ile uygulandığı C++ programlama dilinde yazılan program

```

#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define uye_max 25
int cross=20; //Çaprazlama Oranı
int mutat=10; //Mutasyon Oranı
int uye_say=15;

int sekil[33][2],i,l,h,m,tanim[33][7],kaynak[30];
int j,k,ii,haf[38][4],nesil,temp;
float hatm[38][5],akim[33],akimw[33],akimdw[33],uye_u[uye_max+1],top_u[uye_max+1];
float top_uyum,rnnd[21],ftemp,en_ iyi=0.0,en_ iyi1[4],tt[uye_max+1];
int gdriver = DETECT, gmode, errorcode;
int left, top, right, bottom,rnd,rnd1;
int rnd2,temp1,temp2,temp3,tempe,tempe1,tempe2,tempe3;
int uye[uye_max+1][4],yeni_ uye[uye_max+1][4],cev[6][12];
int teem[3][4],z,r1[3],ka_ ce[uye_max+2][4],kac,tamam,h_ acik,h_ buldu;
int seviye1[150],tem,seviye,uyet[4],a_ hat;
void kontrol1();

/***/Başlangıç Verilerinin Atanması *****/
Bas_Dur_At()
{
//Ekran çizdirilen şekille ilgili değişken ve değerleri
sekil[0][1]=370;sekil[0][2]=60;
sekil[1][1]=370;sekil[1][2]=100;
sekil[2][1]=370;sekil[2][2]=140;
sekil[3][1]=370; sekil[3][2]=180;
sekil[4][1]=370;sekil[4][2]=220;
sekil[5][1]=320;sekil[5][2]=220;
sekil[6][1]=270;sekil[6][2]=220;
sekil[7][1]=270; sekil[7][2]=260;
sekil[8][1]=220; sekil[8][2]=260;
sekil[9][1]=220;sekil[9][2]=220;
sekil[10][1]=220;sekil[10][2]=180;
sekil[11][1]=170;sekil[11][2]=220;
sekil[12][1]=170;sekil[12][2]=260;
sekil[13][1]=220;sekil[13][2]=300;
sekil[14][1]=270;sekil[14][2]=340;
sekil[15][1]=320; sekil[15][2]=340;
sekil[16][1]=370; sekil[16][2]=340;
sekil[17][1]=320;sekil[17][2]=60;
sekil[18][1]=270;sekil[18][2]=60;
sekil[19][1]=270;sekil[19][2]=140;
sekil[20][1]=220;sekil[20][2]=140;
sekil[21][1]=420;sekil[21][2]=100;

```

```

sekil[22][1]=470;sekil[22][2]=100;
sekil[23][1]=470;sekil[23][2]=140;
sekil[24][1]=370;sekil[24][2]=260;
sekil[25][1]=420;sekil[25][2]=260;
sekil[26][1]=470;sekil[26][2]=260;
sekil[27][1]=470;sekil[27][2]=220;
sekil[28][1]=520;sekil[28][2]=260;
sekil[29][1]=520;sekil[29][2]=300;
sekil[30][1]=470;sekil[30][2]=340;
sekil[31][1]=420;sekil[31][2]=340;

```

//Yükler arasında var olan bağlantılar

```

tanım[1][1]=3;tanım[1][2]=0;tanım[1][3]=2;tanım[1][4]=18;tanım[1][5]=0;
tanım[2][1]=3;tanım[2][2]=1;tanım[2][3]=3;tanım[2][4]=22;
tanım[3][1]=2;tanım[3][2]=2;tanım[3][3]=4;
tanım[4][1]=2;tanım[4][2]=3;tanım[4][3]=5;
tanım[5][1]=3;tanım[5][2]=4;tanım[5][3]=6;tanım[5][4]=25;
tanım[6][1]=2;tanım[6][2]=5;tanım[6][3]=7;
tanım[7][1]=3;tanım[7][2]=6;tanım[7][3]=8;tanım[7][4]=20;
tanım[8][1]=3;tanım[8][2]=7;tanım[8][3]=9;tanım[8][4]=14;
tanım[9][1]=2;tanım[9][2]=8;tanım[9][3]=10;
tanım[10][1]=2;tanım[10][2]=9;tanım[10][3]=11;
tanım[11][1]=3;tanım[11][2]=10;tanım[11][3]=12;tanım[11][4]=21;
tanım[12][1]=2;tanım[12][2]=11;tanım[12][3]=13;
tanım[13][1]=2;tanım[13][2]=12;tanım[13][3]=14;
tanım[14][1]=3;tanım[14][2]=8;tanım[14][3]=13;tanım[14][4]=15;
tanım[15][1]=2;tanım[15][2]=14;tanım[15][3]=16;
tanım[16][1]=2;tanım[16][2]=15;tanım[16][3]=17;
tanım[17][1]=2;tanım[17][2]=16;tanım[17][3]=32;
tanım[18][1]=2;tanım[18][2]=1;tanım[18][3]=19;
tanım[19][1]=2;tanım[19][2]=18;tanım[19][3]=20;
tanım[20][1]=3;tanım[20][2]=7;tanım[20][3]=19;tanım[20][4]=21;
tanım[21][1]=2;tanım[21][2]=11;tanım[21][3]=20;
tanım[22][1]=2;tanım[22][2]=2;tanım[22][3]=23;
tanım[23][1]=2;tanım[23][2]=22;tanım[23][3]=24;
tanım[24][1]=2;tanım[24][2]=23;tanım[24][3]=28;
tanım[25][1]=2;tanım[25][2]=5;tanım[25][3]=26;
tanım[26][1]=2;tanım[26][2]=25;tanım[26][3]=27;
tanım[27][1]=2;tanım[27][2]=26;tanım[27][3]=28;
tanım[28][1]=3;tanım[28][2]=24;tanım[28][3]=27;tanım[28][4]=29;
tanım[29][1]=2;tanım[29][2]=28;tanım[29][3]=30;
tanım[30][1]=2;tanım[30][2]=29;tanım[30][3]=31;
tanım[31][1]=2;tanım[31][2]=30;tanım[31][3]=32;
tanım[32][1]=2;tanım[32][2]=17;tanım[32][3]=31;

```

//Kısıtların ikili tabanda belirlenmesi

```

ka_ce[1][1]=pow(2,(3-1))+pow(2,(4-1))+pow(2,(5-1));
ka_ce[1][2]=pow(2,(22-13))+pow(2,(23-13))+pow(2,(24-13));
ka_ce[1][3]=pow(2,(25-25))+pow(2,(26-25))+pow(2,(27-25))+pow(2,(28-25))
+pow(2,(37-25));
ka_ce[2][1]=pow(2,(2-1))+pow(2,(3-1))+pow(2,(4-1))+pow(2,(5-1))

```

```

+pow(2,(6-1))+pow(2,(7-1));
ka_ce[2][2]=pow(2,(18-13))+pow(2,(19-13))+pow(2,(20-13));
ka_ce[2][3]=pow(2,(33-25));
ka_ce[3][1]=pow(2,(2-1))+pow(2,(6-1))+pow(2,(7-1));
ka_ce[3][2]=pow(2,(18-13))+pow(2,(19-13))+pow(2,(20-13))
+pow(2,(22-13))+pow(2,(23-13))+pow(2,(24-13));
ka_ce[3][3]=pow(2,(33-25))+pow(2,(37-25))+pow(2,(25-25))
+pow(2,(26-25))+pow(2,(27-25))+pow(2,(28-25));
ka_ce[4][1]=pow(2,(8-1))+pow(2,(9-1))+pow(2,(10-1))+pow(2,(11-1));
ka_ce[4][2]=pow(2,(21-13));
ka_ce[4][3]=pow(2,(33-25))+pow(2,(35-25));
ka_ce[5][1]=pow(2,(2-1))+pow(2,(3-1))+pow(2,(4-1))+pow(2,(5-1))
+pow(2,(6-1))+pow(2,(7-1))+pow(2,(8-1))+pow(2,(9-1))
+pow(2,(10-1))+pow(2,(11-1));
ka_ce[5][2]=pow(2,(18-13))+pow(2,(19-13))+pow(2,(20-13))+pow(2,(21-13));
ka_ce[5][3]=pow(2,(35-25));
ka_ce[6][1]=pow(2,(2-1))+pow(2,(6-1))+pow(2,(7-1))+pow(2,(8-1))
+pow(2,(9-1))+pow(2,(10-1))+pow(2,(11-1));
ka_ce[6][2]=pow(2,(18-13))+pow(2,(19-13))+pow(2,(20-13))+pow(2,(21-13))
+pow(2,(24-13))+pow(2,(23-13))+pow(2,(22-13));
ka_ce[6][3]=pow(2,(35-25))+pow(2,(25-25))+pow(2,(26-25))+pow(2,(27-25))
+pow(2,(28-25))+pow(2,(37-25));
ka_ce[7][1]=pow(2,(9-1))+pow(2,(10-1))+pow(2,(11-1))+pow(2,(12-1));
ka_ce[7][2]=pow(2,(13-13))+pow(2,(14-13));
ka_ce[7][3]=pow(2,(34-25));
ka_ce[8][1]=pow(2,(8-1))+pow(2,(12-1));
ka_ce[8][2]=pow(2,(13-13))+pow(2,(14-13))+pow(2,(21-13))+pow(2,(33-13))
+pow(2,(35-13));
ka_ce[8][3]=pow(2,(34-25))+pow(2,(33-25))+pow(2,(35-25));
ka_ce[9][1]=pow(2,(2-1))+pow(2,(3-1))+pow(2,(4-1))+pow(2,(5-1))
+pow(2,(6-1))+pow(2,(7-1))+pow(2,(8-1))+pow(2,(12-1));
ka_ce[9][2]=pow(2,(13-13))+pow(2,(14-13))+pow(2,(18-13))+pow(2,(19-13))
+pow(2,(20-13))+pow(2,(21-13));
ka_ce[9][3]=pow(2,(35-25))+pow(2,(34-25));
ka_ce[10][1]=pow(2,(2-1))+pow(2,(6-1))+pow(2,(7-1))+pow(2,(8-1))
+pow(2,(12-1));
ka_ce[10][2]=pow(2,(13-13))+pow(2,(14-13))+pow(2,(18-13))+pow(2,(19-13))
+pow(2,(20-13))+pow(2,(21-13))+pow(2,(24-13))
+pow(2,(23-13))+pow(2,(22-13));
ka_ce[10][3]=pow(2,(35-25))+pow(2,(25-25))+pow(2,(26-25))
+pow(2,(27-25))+pow(2,(28-25))+pow(2,(37-25))+pow(2,(34-25));
//Başlangıçta rasgele atanan üyeler için her kapalı cevreden bir anahtar açılıyor.
//Bu 5 çevre ile ilgili bilgiler:
cev[1][0]=8;cev[1][1]=15;cev[1][2]=16;cev[1][3]=17;cev[1][4]=36;
cev[1][5]=32;cev[1][6]=31;cev[1][7]=30;cev[1][8]=29;
cev[2][0]=11;cev[2][1]=3;cev[2][2]=4;cev[2][3]=5;cev[2][4]=25;cev[2][5]=26;
cev[2][6]=27;cev[2][7]=28;cev[2][8]=37;cev[2][9]=24;cev[2][10]=23;
cev[2][11]=22;
cev[3][0]=10;cev[3][1]=2;cev[3][2]=3;cev[3][3]=4;cev[3][4]=5;
cev[3][5]=6;cev[3][6]=7;cev[3][7]=33;cev[3][8]=20;cev[3][9]=19;
cev[3][10]=18;

```

```
cev[4][0]=7;cev[4][1]=8;cev[4][2]=9;cev[4][3]=10;cev[4][4]=11;
cev[4][5]=35;cev[4][6]=21;cev[4][7]=33;
cev[5][0]=7;cev[5][1]=9;cev[5][2]=10;cev[5][3]=11;cev[5][4]=12;
cev[5][5]=13;cev[5][6]=14;cev[5][7]=34;
```

```
//Hatların hangi noktalar arasında tanımlı olduğu bilgisi
```

```
// Ve her hattın direnç ve endüktif reaktans değerleri
```

```
hat[1][1]=1;hat[1][2]=0;hatm[1][1]=0.0922;hatm[1][2]=0.0470;
hat[2][1]=2;hat[2][2]=1;hatm[2][1]=0.4930;hatm[2][2]=0.2511;
hat[3][1]=3;hat[3][2]=2;hatm[3][1]=0.3660;hatm[3][2]=0.1864;
hat[4][1]=4;hat[4][2]=3;hatm[4][1]=0.3811;hatm[4][2]=0.1941;
hat[5][1]=5;hat[5][2]=4;hatm[5][1]=0.8190;hatm[5][2]=0.7070;
hat[6][1]=6;hat[6][2]=5;hatm[6][1]=0.1872;hatm[6][2]=0.6188;
hat[7][1]=7;hat[7][2]=6;hatm[7][1]=0.7114;hatm[7][2]=0.2351;
hat[8][1]=8;hat[8][2]=7;hatm[8][1]=1.0300;hatm[8][2]=0.7400;
hat[9][1]=9;hat[9][2]=8;hatm[9][1]=1.0440;hatm[9][2]=0.7400;
hat[10][1]=10;hat[10][2]=9;hatm[10][1]=0.1966;hatm[10][2]=0.0650;
hat[11][1]=11;hat[11][2]=10;hatm[11][1]=0.3744;hatm[11][2]=0.1238;
hat[12][1]=12;hat[12][2]=11;hatm[12][1]=1.4680;hatm[12][2]=1.1550;
hat[13][1]=13;hat[13][2]=12;hatm[13][1]=0.5416;hatm[13][2]=0.7129;
hat[14][1]=14;hat[14][2]=13;hatm[14][1]=0.5910;hatm[14][2]=0.5260;
hat[15][1]=15;hat[15][2]=14;hatm[15][1]=0.7463;hatm[15][2]=0.5450;
hat[16][1]=16;hat[16][2]=15;hatm[16][1]=1.2890;hatm[16][2]=1.7210;
hat[17][1]=17;hat[17][2]=16;hatm[17][1]=0.7320;hatm[17][2]=0.5740;
hat[18][1]=18;hat[18][2]=17;hatm[18][1]=0.1640;hatm[18][2]=0.1565;
hat[19][1]=19;hat[19][2]=18;hatm[19][1]=1.5042;hatm[19][2]=1.3554;
hat[20][1]=20;hat[20][2]=19;hatm[20][1]=0.4095;hatm[20][2]=0.4784;
hat[21][1]=21;hat[21][2]=20;hatm[21][1]=0.7089;hatm[21][2]=0.9373;
hat[22][1]=22;hat[22][2]=21;hatm[22][1]=0.4512;hatm[22][2]=0.3083;
hat[23][1]=23;hat[23][2]=22;hatm[23][1]=0.8980;hatm[23][2]=0.7091;
hat[24][1]=24;hat[24][2]=23;hatm[24][1]=0.8960;hatm[24][2]=0.7011;
hat[25][1]=25;hat[25][2]=24;hatm[25][1]=0.2030;hatm[25][2]=0.1034;
hat[26][1]=26;hat[26][2]=25;hatm[26][1]=0.2842;hatm[26][2]=0.1447;
hat[27][1]=27;hat[27][2]=26;hatm[27][1]=1.0590;hatm[27][2]=0.9337;
hat[28][1]=28;hat[28][2]=27;hatm[28][1]=0.8042;hatm[28][2]=0.7006;
hat[29][1]=29;hat[29][2]=28;hatm[29][1]=0.5075;hatm[29][2]=0.2585;
hat[30][1]=30;hat[30][2]=29;hatm[30][1]=0.9744;hatm[30][2]=0.9630;
hat[31][1]=31;hat[31][2]=30;hatm[31][1]=0.3105;hatm[31][2]=0.3619;
hat[32][1]=32;hat[32][2]=31;hatm[32][1]=0.3410;hatm[32][2]=0.5302;
hat[33][1]=20;hat[33][2]=7;hatm[33][1]=2.0000;hatm[33][2]=2.0000;
hat[34][1]=14;hat[34][2]=8;hatm[34][1]=2.0000;hatm[34][2]=2.0000;
hat[35][1]=21;hat[35][2]=11;hatm[35][1]=2.0000;hatm[35][2]=2.0000;
hat[36][1]=32;hat[36][2]=17;hatm[36][1]=0.5000;hatm[36][2]=0.5000;
hat[37][1]=28;hat[37][2]=24;hatm[37][1]=0.5000;hatm[37][2]=0.5000;
```

```
//Her yükün çektiği aktif ve reaktif akım bileşenleri
```

```
akimw[1]=4.5604;akimdw[1]=2.7363;
akimw[2]=4.1044;akimdw[2]=1.8242;
akimw[3]=5.4725;akimdw[3]=3.6483;
akimw[4]=2.7363;akimdw[4]=1.3681;
akimw[5]=2.7363;akimdw[5]=0.9121;
akimw[6]=2.7363;akimdw[6]=0.9121;
```



```

akimw[6]=9.1209;akimdw[6]=4.5604;
akimw[7]=9.1209;akimdw[7]=4.5604;
akimw[8]=2.7363;akimdw[8]=0.9121;
akimw[9]=2.7363;akimdw[9]=0.9121;
akimw[10]=2.0522;akimdw[10]=1.3681;
akimw[11]=2.7363;akimdw[11]=1.5962;
akimw[12]=2.7363;akimdw[12]=1.5962;
akimw[13]=5.4725;akimdw[13]=3.6483;
akimw[14]=2.7363;akimdw[14]=0.4560;
akimw[15]=2.7363;akimdw[15]=0.9121;
akimw[16]=2.7363;akimdw[16]=0.9121;
akimw[17]=4.1044;akimdw[17]=1.8242;
akimw[18]=4.1044;akimdw[18]=1.8242;
akimw[19]=4.1044;akimdw[19]=1.8242;
akimw[20]=4.1044;akimdw[20]=1.8242;
akimw[21]=4.1044;akimdw[21]=1.8242;
akimw[22]=4.1044;akimdw[22]=2.2802;
akimw[23]=19.1538;akimdw[23]=9.1209;
akimw[24]=19.1538;akimdw[24]=9.1209;
akimw[25]=2.7363;akimdw[25]=1.1401;
akimw[26]=2.7363;akimdw[26]=1.1401;
akimw[27]=2.7363;akimdw[27]=0.9121;
akimw[28]=5.4725;akimdw[28]=3.1923;
akimw[29]=9.1209;akimdw[29]=27.3626;
akimw[30]=6.8406;akimdw[30]=3.1923;
akimw[31]=9.5769;akimdw[31]=4.5604;
akimw[32]=2.7363;akimdw[32]=1.8242;

```

```

//*****Başlangıçta üyeler rasgele fakat kısıtlara uygun şekilde belirleniyor *****

```

```

kaynak[1]=1;
kaynak[2]=0;
randomize();
for(ii=1;ii<=uye_say;ii++)
{
if(ii==1) printf(" \n");
uye[ii][1]=0;uye[ii][2]=0; uye[ii][3]=0;
for(k=1;k<=5;k++)
{
i=random(cev[k][0])+1;
if(cev[k][i]<13)uye[ii][1]+=pow(2,(cev[k][i]-1));
else if(cev[k][i]>24)uye[ii][3]+=pow(2,(cev[k][i]-25));
else uye[ii][2]+=pow(2,(cev[k][i]-13));
}
kac=ii;
kontrol1(); //Çevre oluşturup oluşturmadığını kontrol ediyor.
if(tamam==0)ii--;
}
}
//Kapalı çevre oluşturup oluşturmadığı kontrol ediliyor

```

```

void kontroll()
{
    for(i=1;i<=10;i++)
    {
        if(((ka_ce[i][1]&uye[kac][1])||((ka_ce[i][2]&uye[kac][2])||((ka_ce[i][3]&uye[kac][3]
        ))!=0)tamam=1;
        else {tamam=0;goto atla;}
    }
    atla:
}
//*****
//Havuzdaki üyeler uygunluk değerlerine göre sıralanıyor
sort_population()

{
    for(i=1;i<=(uye_say-1);i++){
        for(j=i+1;j<=uye_say;j++)
            if(uye_u[i]<uye_u[j]){for(k=1; k<=3;k++)
            {
                temp=uye[j][k];
                uye[j][k]=uye[i][k];
                uye[i][k]=temp;
            }
            ftemp=uye_u[i];
            uye_u[i]=uye_u[j];
            uye_u[j]=ftemp;
        }
    }
}
//*****
//Her üyenin sırayla enerji akışı belirleniyor
//Ve ne kadar enerji kaybı olduğu hesaplanıp uygunluk değerleri bulunuyor
maliet()
{
    //k Üyenin indeksi
    for(k=1;k<(uye_say+1);k++)
    {
        // *** Her üyenin enerjilendirilmesi *****
        seviye1[0]=1;
        seviye1[1]=1;
        seviye=1;
    ddder:
        //her düğümün mümkün bağlanma yerlerini teker teker kontrol için
        for(i=2;i<=(tanım[seviye1[seviye]][1]+1);i++)
        {

            if(tanım[seviye1[seviye]][i]==tanım[seviye1[seviye]][5])
                goto atla; //eğer enerji aldığı yer ise oraya bakma
            // eldeki düğümün bağlı olduğu düğümlerle arasındaki hattı bul

```

```

for(ii=1;ii<38;ii++)
if((((tanım[seviye1[seviye]][i]==hat[ii][1])
    &&(seviye1[seviye]==hat[ii][2]))||
    ((tanım[seviye1[seviye]][i]==hat[ii][2])&&
    (seviye1[seviye]==hat[ii][1])))
{
    h_buldu=ii;
    goto cik;
}
cik:
// Bulunan hattın açık mı yoksa kapalı mı olduğunu kontrol et
h_acik=0;
if((h_buldu>12)&&(h_buldu<25))
{
    tem=pow(2,(h_buldu-13));
    if((tem&uye[k][2])!=0)h_acik=1;
}
if(h_buldu>24)
{
    tem=pow(2,(h_buldu-25));
    if((tem&uye[k][3])!=0)h_acik=1;
}
if(h_buldu<13)
{
    tem=pow(2,(h_buldu-1));
    if((tem&uye[k][1])!=0)h_acik=1;
}
// hat açık ise bir işlem yapma
// kapalı ise bu düğümden enerji alınıp alınmadığını kontrol
// etmek için seviye1'e bunu kaydet...
if(h_acik==0)
{
    tanım[tanım[seviye1[seviye]][i]][5]=seviye1[seviye];
    seviye1[0]++;
    seviye1[seviye1[0]]=tanım[seviye1[seviye]][i];
}
atla:
}
seviye++;
if(seviye>seviye1[0]);
else goto ddder;
/****Enerjilendirme işleminin sonu****
/****Kayıpların hesaplanması****
for(i=1;i<38;i++){hatm[i][3]=0;hatm[i][4]=0;}
for(i=32;i>1;i--)
{ j=i;
devamet:
for(ii=1;ii<38;ii++)
    if(((j==hat[ii][1])&&
        (tanım[j][5]==hat[ii][2]))||((j==hat[ii][2])&&
        (tanım[j][5]==hat[ii][1])))

```

```

        {
            h_buldu=ii;
            goto cik1;
        }

cik1:
    hatm[h_buldu][3]=hatm[h_buldu][3]+akimw[i];
    hatm[h_buldu][4]=hatm[h_buldu][4]+akimdw[i];
    if(tanim[j][5]==1)goto b_tamam;
    j=tanim[j][5];
    goto devamet;
b_tamam:
}
hatm[1][3]=hatm[2][3]+hatm[18][3]+akimw[1];
hatm[1][4]=hatm[2][4]+hatm[18][4]+akimdw[1];
ftemp=0.0;
for(i=1;i<38;i++)ftemp+=(hatm[i][3]*hatm[i][3]+hatm[i][4]*hatm[i][4])*hatm[i][1]/1000.0;
uye_u[k]=1.0/(ftemp-40.0); //Bulunan kayiptan uygunluk deęerleri hesaplanıyor
}

    sort_population();           //Havuzdaki üyeler uygunluk deęerlerine göre sıralanıyor
}
//*****
//Üyelerin uygunluk deęerleri kullanılarak yeni havuza kopyalama yapılıyor
select_parents()

{
    // En iyi üye hafızada tutuluyor ve havuza en
    // kötü üyenin üzerine kopyalanıyor
    if(en_ iyi>uye_u[1])
    {
        for (m=1; m<=3;m++)uye[uye_say][m]=en_ iyi1[m];
        uye_u[uye_say]=en_ iyi;
    }
    else
    {
        for(m=1; m<=3;m++) en_ iyi1[m]=uye[1][m];
        en_ iyi=uye_u[1];
    }
}
sort_population(); //yeniden popülasyonu sırala
printf("\n ÜyeNo Güç Uygunluk");
for(k=1;k<(uye_say+1);k++)
printf("\n%2d %8.3f %3.3f",k,(1.0/uye_u[k]+40.0)*3,uye_u[k]);
//*****Yeni popülasyonu eski popülasyonun uyumluluklarına bakarak üret *****
top_uyum=0;
for(i=1;i<=uye_say;i++)
{
    top_uyum=top_uyum+uye_u[i];
    top_u[i]=top_uyum;
}

```

```

}
for(i=1;i<=uye_say;i++)
{
  rndd[i]=(rand()%100)/100.0*top_uyum;
  for(m=1;top_u[m]<=rndd[i];m++);
  for(j=1;j<=3;j++)yeni_uye[i][j]=uye[m][j];
  tt[i]=uye_u[m];
}
for(i=1;i<=uye_say;i++)
{

  uye_u[i]=tt[i];
}
getch();
}
//*****
//Mutasyon ve çaprazlamanın yapıldığı bölüm
replace_population()
{
//*****çaprazlama yapılıyor*****
  for(j=1;j<=(cross/100.0*uye_say);j++)
  {
//Hangi üye
    r1[1]=rand()%uye_say+1;
//Hangi üye ile
    r1[2]=rand()%uye_say+1;
// Seçilen iki üye enerjilendirilip çaprazlama yapılabilirliği
// kontrol ediliyor
    for(z=1;z<3;z++)
    {
      seviye1[0]=1;
      seviye1[1]=1;
      seviye=1;
      k=r1[z];
dder2:
      for(i=2;i<=(tanim[seviye1[seviye]][1]+1);i++)
      {
        if(tanim[seviye1[seviye]][i]==tanim[seviye1[seviye]][5]) goto atla2;
        for(ii=1;ii<38;ii++)if(((tanim[seviye1[seviye]][i]==hat[ii][1])
          &&(seviye1[seviye]==hat[ii][2]))||
          ((tanim[seviye1[seviye]][i]==hat[ii][2])&&
          (seviye1[seviye]==hat[ii][1])))
          {
            h_buldu=ii;
            goto cik2;
          }
      }
      cik2:
        h_acik=0;
        if((h_buldu>12)&&(h_buldu<25))
        {
          tem=pow(2,(h_buldu-13));

```

```

        if((tem&uye[k][2])!=0)h_acik=1;
    }
    if(h_buldu>24)
    {
        tem=pow(2,(h_buldu-25));
        if((tem&uye[k][3])!=0)h_acik=1;
    }
    if(h_buldu<13)
    {
        tem=pow(2,(h_buldu-1));
        if((tem&uye[k][1])!=0)h_acik=1;
    }
    if(h_acik==0)
    {
        tanim[tanim[seviye1[seviye]][i]][5]=seviye1[seviye];seviye1[0]++;
        seviye1[seviye1[0]]=tanim[seviye1[seviye]][i];
    }
atla2:
}
seviye++;
if(seviye>seviye1[0]);
else goto ddder2;
teem[z][1]=tanim[18][5];
teem[z][2]=tanim[6][5];
teem[z][3]=tanim[17][5];
}
// Eğer bu iki üye çaprazlanabilir ise çaprazlama yap
if((teem[1][1]==teem[2][1])&&
(teem[1][2]==teem[2][2])&&(teem[1][3]==teem[2][3]))
{
    temp2=pow(2,(18-13))+pow(2,(22-13))+pow(2,(23-13))+pow(2,(24-13));
    temp3=pow(2,(25-25))+pow(2,(26-25))+pow(2,(27-25))+pow(2,(28-25))
        +pow(2,(29-25))+pow(2,(30-25))+pow(2,(31-25))+pow(2,(32-25))
        +pow(2,(36-25))+pow(2,(37-25));

    tempe1=temp1&uye[r1[1]][1];
    tempe2=temp1&uye[r1[2]][1];
    uye[r1[1]][1]=((temp1|uye[r1[1]][1])^temp1)|tempe2;
    uye[r1[2]][1]=((temp1|uye[r1[2]][1])^temp1)|tempe1;

    tempe1=temp2&uye[r1[1]][2];
    tempe2=temp2&uye[r1[2]][2];
    uye[r1[1]][2]=((temp2|uye[r1[1]][2])^temp2)|tempe2;
    uye[r1[2]][2]=((temp2|uye[r1[2]][2])^temp2)|tempe1;

    tempe1=temp3&uye[r1[1]][3];
    tempe2=temp3&uye[r1[2]][3];
    uye[r1[1]][3]=((temp3|uye[r1[1]][3])^temp3)|tempe2;
    uye[r1[2]][3]=((temp3|uye[r1[2]][3])^temp3)|tempe1;

```

```

    }
    else j--;
}
//*****
// Mutasyon yapıyor
for(ii=1;ii<=(mutat/100.0*uye_say);ii++)
{
    rnd=rand()%uye_say+1;        //Hangi üyenin
    rnd1=rand()%5+1;            //Hangi çevresinin
    rnd2=rand()%cev[rnd1][0]+1; //Hangi elemanı
    temp1=0;temp2=0;temp3=0;
    for(k=1;k<(cev[rnd1][0]+1);k++)

    {
        if(cev[rnd1][k]<13)
        {
            temp=pow(2,(cev[rnd1][k]-1));
            if((temp&uye[rnd][1])!=0)a_hat=cev[rnd1][k];
        }
        if((cev[rnd1][k]>12)&&(cev[rnd1][k]<25))
        {
            temp=pow(2,(cev[rnd1][k]-13));
            if((temp&uye[rnd][2])!=0)a_hat=cev[rnd1][k];
        }
        if(cev[rnd1][k]>24)
        {
            temp=pow(2,(cev[rnd1][k]-25));
            if((temp&uye[rnd][3])!=0)a_hat=cev[rnd1][k];
        }
    }
    if(a_hat<13)
    {
        temp=pow(2,(a_hat-1));
        uye[rnd][1]=uye[rnd][1]^temp;
    }
    if((a_hat>12)&&(a_hat<25))
    {
        temp=pow(2,(a_hat-13));
        uye[rnd][2]=uye[rnd][2]^temp;
    }
    if(a_hat>24)
    {
        temp=pow(2,(a_hat-25));
        uye[rnd][3]=uye[rnd][3]^temp;
    }
}
tekrar:
    if(cev[rnd1][rnd2]<13)
    {
        temp1=pow(2,(cev[rnd1][rnd2]-1));
        if((temp1&uye[rnd][1])!=0)
        {

```

```

        rnd2=rand()%cev[rnd1][0]+1;
        goto tekrar;
    }
    uye[rnd][1]=tempe1|uye[rnd][1];
}
if((cev[rnd1][rnd2]>12)&&(cev[rnd1][rnd2]<25))
{
    tempe2=pow(2,(cev[rnd1][rnd2]-13));
    if((tempe2&uye[rnd][2])!=0)
    {
        rnd2=rand()%cev[rnd1][0]+1;
        goto tekrar;
    }
    uye[rnd][2]=tempe2|uye[rnd][2];
}
if(cev[rnd1][rnd2]>24)
{
    tempe3=pow(2,(cev[rnd1][rnd2]-25));
    if((tempe3&uye[rnd][3])!=0)
    {
        rnd2=rand()%cev[rnd1][0]+1;
        goto tekrar;
    }
    uye[rnd][3]=tempe3|uye[rnd][3];
}
kac=rnd;
kontrol1();
if(tamam==0)
{
    if(rnd2<13)uye[rnd][1]=tempe1^uye[rnd][1];
    if((rnd2>12)&&(rnd2<25))uye[rnd][2]=tempe2^uye[rnd][2];
    if(rnd2>24)uye[rnd][3]=tempe3^uye[rnd][3];
    rnd2=rand()%cev[rnd1][0]+1;
    goto tekrar;
}
}
}
}
}
}

/***** ANA PROGRAM *****/
int main(void)
{
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics hatasi: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}

Bas_Dur_At();

```



```

maliyet();
sort_population();
dd:
select_parents();

```

```

maliyet();
// En iyi üye çizdirilmek üzere enerjilendiriliyor
if(en_iyi>uye_u[1])
{
    for (m=1; m<=3;m++) uye[uye_say][m]=en_iyi1[m];
    uye_u[uye_say]=en_iyi;
}
else
{
    for(m=1; m<=3;m++) en_iyi1[m]=uye[1][m];
    en_iyi=uye_u[1];
}
sort_population(); //yeniden populasyonu sırala

```

```

seviye1[0]=1;
seviye1[1]=1;
seviye=1;
k=1;
ddder1:
for(i=2;i<=(tanim[seviye1[seviye]][1]+1);i++)
{
    if(tanim[seviye1[seviye]][i]==tanim[seviye1[seviye]][5]) goto atla1;
    for(ii=1;ii<38;ii++)if(((tanim[seviye1[seviye]][i]==hat[ii][1])
    &&(seviye1[seviye]==hat[ii][2]))||
    ((tanim[seviye1[seviye]][i]==hat[ii][2])&&(seviye1[seviye]==hat[ii][1])))
    {
        h_buldu=ii;
        goto cik1;
    }
    cik1:
    h_acik=0;
    if((h_buldu>12)&&(h_buldu<25))
    {
        tem=pow(2,(h_buldu-13));
        if((tem&uye[k][2])!=0)h_acik=1;
    }
    if(h_buldu>24)
    {
        if((tem&uye[k][3])!=0)h_acik=1;
    }
    if(h_buldu<13)
    {
        tem=pow(2,(h_buldu-1));
        if((tem&uye[k][1])!=0)h_acik=1;
    }
}

```

```

if(h_acik==0)
{
    tanim[tanim[seviye1[seviye]][i]][5]=seviye1[seviye];seviye1[0]++;
    seviye1[seviye1[0]]=tanim[seviye1[seviye]][i];
}
atla1:
}
seviye++;
if(seviye>seviye1[0]);
else goto ddder1;
cleardevice();
rectangle(365,5,375,15);
for(i=0;i<32;i++)
circle(sekil[i][1],sekil[i][2],4);
line(sekil[0][1],sekil[0][2],370,10);
for(i=2;i<33;i++)
{
    line(sekil[i-1][1],sekil[i-1][2],sekil[tanim[i][5]-1][1],sekil[tanim[i][5]-1][2]);
}

nesil++;
gotoxy(1,2);
printf(" \nNesil=%d Uygunluk=%f",nesil,uye_u[1]);

if(kbhit())
{
    if(getch()=='q')exit(0);}
else goto bekle;
goto dd;
closegraph();
return 0;
}

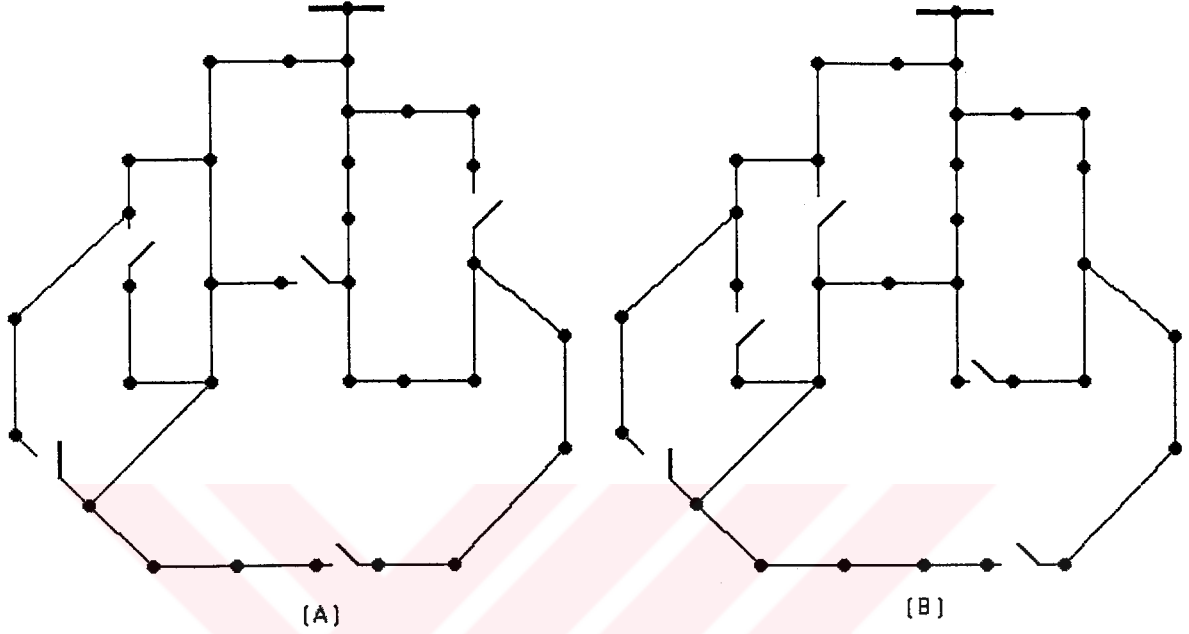
```

Ek 2 İlk popülasyonda bile uygunluk değerlerinin birbirine ne kadar yakın olduğunu gösteren çizelge

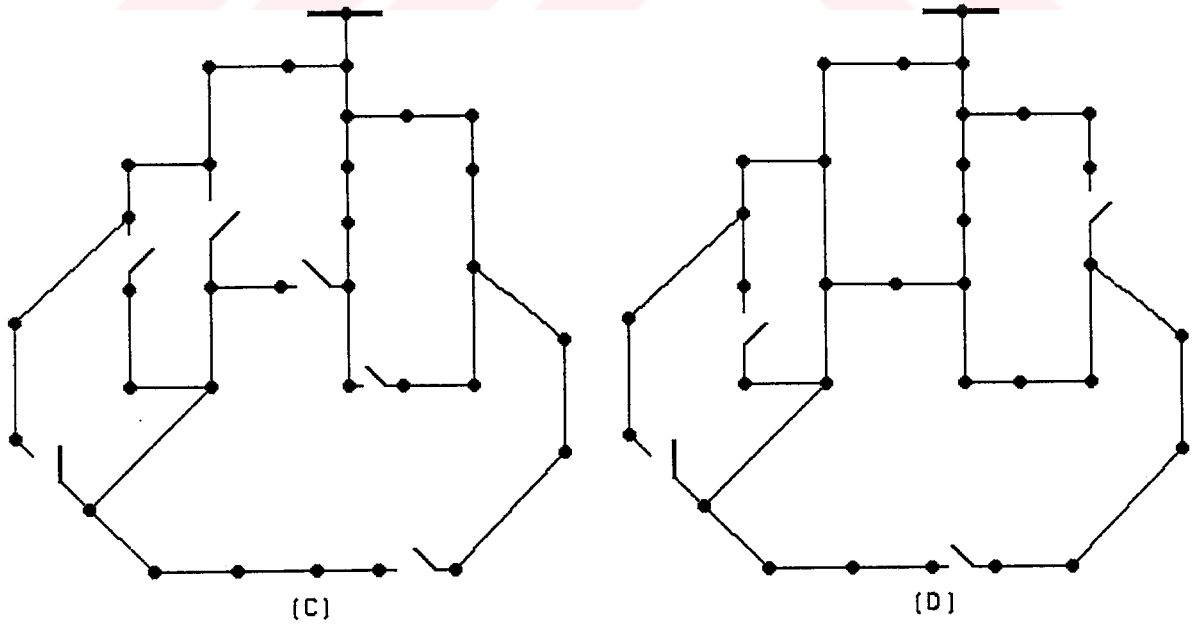
Üye Sayısı : 15		
Çaprazlama Oranı : 0.30		
Mutasyon Oranı : 0.30		
Nesil=1 Uygunluk=0.007308		
ÜyeNo	Güç	Uygunluk
1	136.832	0.007
2	155.005	0.006
3	157.618	0.006
4	161.653	0.006
5	177.125	0.006
6	177.125	0.006
7	188.801	0.005
8	188.801	0.005
9	203.722	0.005
10	214.040	0.005
11	227.702	0.004
12	268.254	0.004
13	370.757	0.003
14	509.495	0.002
15	552.142	0.002

Ek 3 Tek noktalı çaprazlamamın sakıncalı yanlarının şekil üzerinde görsel olarak gerçekleştirilmesi

A=1111101111011011111 | 111111111111111100
 B=1111111111011011111 | 111111011111001111



C=1111101111011011111 | 111110111111001111
 D=1111111111011011111 | 111111111111111100



Ek 4 MATPOWER optimizasyon programının yük akış analizi yapan “opf” dosyası

```

function [bus, gen, branch, f, success, et] = opf(baseMVA, bus, gen, gencost, ...

if nargin < 12
    mpopt = mpopt;
end

verbose    = mpopt(31);
npts = mpopt(14);
j = sqrt(-1);

[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX,
    MU_VMIN] = idx_bus;
[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, ...
    GEN_STATUS, PMAX, PMIN, MU_PMAX, MU_PMIN, MU_QMAX,
    MU_QMIN] = idx_gen;
[PW_LINEAR, POLYNOMIAL, MODEL, STARTUP, SHUTDOWN, N, COST] = idx_cost;

model = gencost(:, MODEL);
i_pwln = find(model == PW_LINEAR);
i_poly = find(model == POLYNOMIAL);
if any(i_pwln) & any(i_poly) & verbose
    fprintf('not all generators use same cost model, all will be converted to piece-wise
    linear\n');
end

if mpopt(11) == 0
    if find(model ~= PW_LINEAR & model ~= POLYNOMIAL)
        error('unknown generator cost model!');
    elseif any(i_pwln)
        mpopt(11) = mpopt(13);
    else
        mpopt(11) = mpopt(12);
    end
end

alg = mpopt(11);
formulation = opf_form(alg);

if any(i_pwln)
    ng = size(gen, 1);
    gen(:, PMIN) = gen(:, PMIN) - 1e-6 * ones(ng, 1);
    gen(:, PMAX) = gen(:, PMAX) + 1e-6 * ones(ng, 1);
end

if any(i_pwln) & formulation == 1
    error(sprintf('algorithm %d does not handle piece-wise linear cost functions',
    alg));
end

if any(i_poly) & formulation == 2

```

```

if verbose
    fprintf('converting from polynomial to piece-wise linear cost model\n');
end
[pcost, qcost] = pqcost(gencost, size(gen, 1));
i_poly = find(pcost(:, MODEL) == POLYNOMIAL);
pcost = poly2pwl(pcost(i_poly, :), gen(i_poly, PMIN), gen(i_poly, PMAX), npts);
if ~isempty(qcost)
    i_poly = find(qcost(:, MODEL) == POLYNOMIAL);
    qcost = poly2pwl(qcost(i_poly, :), gen(i_poly, QMIN), gen(i_poly, QMAX), npts);
end
gencost = [pcost; qcost];
end

end

t0 = clock;

nb = size(bus, 1);
nl = size(branch, 1);
npv = length(pv);
npq = length(pq);
ng = npv + 1;

on = find(gen(:, GEN_STATUS));
V = bus(:, VM) .* exp(sqrt(-1) * pi/180 * bus(:, VA));
V(gen(on, GEN_BUS)) = gen(on, VG) ./ abs(V(gen(on, GEN_BUS))).* V(gen(on,
EN_BUS));
Pg = gen(on, PG) / baseMVA;
Qg = gen(on, QG) / baseMVA;

[pcost, qcost] = pqcost(gencost, size(gen, 1), on);

j1 = 1; j2 = npv;
j3 = j2 + 1; j4 = j2 + npq;
j5 = j4 + 1; j6 = j4 + nb;
j7 = j6 + 1; j8 = j6 + npv + 1;
j9 = j8 + 1; j10 = j8 + npv + 1;

if formulation == 1
    Cp = [];
    Cq = [];
else
    Cp = totcost(pcost, Pg * baseMVA);
    Cq = totcost(qcost, Qg * baseMVA);
end
end
x = [angle(V([pv; pq])); abs(V); Pg; Qg; Cp; Cq];

[fun, grad] = fg_names(alg);
mpopt(15) = 2 * nb;

if opf_slvr(alg) == 0

    if mpopt(19) == 0

```

```

mpopt(19) = 2 * nb + 150;
end

otopt = foptions;
otopt(1) = (verbose > 0);

otopt(13) = mpopt(15);
otopt(14) = mpopt(19);

[x, otopt, lambda] = constr(fun, x, otopt, [], [], grad, ...

[f, g] = feval(fun, x, baseMVA, bus, gen, gencost, branch,
Ybus, Yf, Yt, V, ref, pv, pq, mpopt);

if otopt(10) >= otopt(14) | max(abs(g(1:otopt(13)))) > otopt(4) ...

success = 0;
else
success = 1;
end

else
[x, success_if] = LPeqslvr(x, baseMVA, bus, gen, gencost, branch, Ybus, Yf, Yt,
V, ref, pv, pq, mpopt);
if success_if ~= 1
error('Sorry, cannot find a starting point using power flow, please check data!');
end

cstep = 0;
if ~isempty(Cp) cstep = max(abs(Cp));
if cstep < 1.0e6, cstep = 1.0e6; end
end
step0=[2*ones(nb-1,1);
ones(nb,1);
0.6*ones(ng,1);
0.3*ones(ng,1);
cstep*ones(length(Cp),1);
cstep*ones(length(Cq),1)];
idx_xi = [];

[x, lambda, success] = LPconstr(fun, x, idx_xi,
mpopt, step0, [], [], grad, 'LPeqslvr', ...

f = feval(fun, x, baseMVA, bus, gen, gencost, branch,
Ybus, Yf, Yt, V, ref, pv, pq, mpopt);

end

Va = zeros(nb, 1);
Va([ref; pv; pq]) = [angle(V(ref)); x(j1:j2); x(j3:j4)];
Vm = x(j5:j6);
V = Vm .* exp(j * Va);
Sg = x(j7:j8) + j * x(j9:j10);

```

```
[bus, gen, branch] = opfsoln(baseMVA, bus, gen, branch, ...
```

```
et = etime(clock, t0);
```

```
return;
```



Ek 5 MATPOWER paket programında yük akış dosyası “opf”yi çalıştıran “runopf” dosyası

```

function [MVABase, bus, gen, gencost, branch, f, success, et] = ...
    runopf(casename, mpopt, fname)
%RUNOPF Runs an optimal power flow.
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX,
    MU_VMIN] = idx_bus;
[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, ...
    GEN_STATUS, PMAX, PMIN, MU_PMAX, MU_PMIN, MU_QMAX,
    MU_QMIN] = idx_gen;

if nargin < 3
    fname = "";
    if nargin < 2
        mpopt = mpoption;
        if nargin < 1
            casename = 'case';
        end
    end
end

[baseMVA, bus, gen, branch, area, gencost] = feval(casename);
[i2e, bus, gen, branch, area] = ext2int(bus, gen, branch, area);

[ref, pv, pq] = bustypes(bus, gen);

[Ybus, Yf, Yt] = makeYbus(baseMVA, bus, branch);

[bus, gen, branch, f, success, et] = opf(baseMVA, bus, gen, gencost, branch, ...

[bus, gen, branch, area] = int2ext(i2e, bus, gen, branch, area);
if fname
    [fd, msg] = fopen(fname, 'at');
    if fd == -1
        error(msg);
    else
        fprintf(baseMVA, bus, gen, branch, f, success, et, fd, mpopt);
        fclose(fd);
    end
end
fprintf(baseMVA, bus, gen, branch, f, success, et, 1, mpopt);

if narginout, MVABase = baseMVA; end

return;

```

Ek 6 MATPOWER programında yük akış analizi yapılmak amacıyla enerji dağıtım sistem verilerinin girildiği dosya

```

function [baseMVA, bus, gen, branch, area, gencost] = case
%CASE Defines the power flow data in a format similar to PTI.
% [baseMVA, bus, gen, branch, area, gencost] = case
% The format for the data is similar to PTI format except where noted.
% An item marked with (+) indicates that it is included in this data
% but is not part of the PTI format. An item marked with (-) is one that
% is in the PTI format but is not included here.
%
% Bus Data Format
% 1 bus number (1 to 29997)
% 2 bus type
%   PQ bus      = 1
%   PV bus      = 2
%   reference bus = 3
%   isolated bus = 4
% 3 Pd, real power demand (MW)
% 4 Qd, reactive power demand (MVAR)
% 5 Gs, shunt conductance (MW (demanded?) at V = 1.0 p.u.)
% 6 Bs, shunt susceptance (MVAR (injected?) at V = 1.0 p.u.)
% 7 area number, 1-100
% 8 Vm, voltage magnitude (p.u.)
% 9 Va, voltage angle (degrees)
% (-) (bus name)
% 10 baseKV, base voltage (kV)
% 11 zone, loss zone (1-999)
% (+) 12 maxVm, maximum voltage magnitude (p.u.)
% (+) 13 minVm, minimum voltage magnitude (p.u.)
%
% Generator Data Format
% 1 bus number
% (-) (machine identifier, 0-9, A-Z)
% 2 Pg, real power output (MW)
% 3 Qg, reactive power output (MVAR)
% 4 Qmax, maximum reactive power output (MVAR)
% 5 Qmin, minimum reactive power output (MVAR)
% 6 Vg, voltage magnitude setpoint (p.u.)
% (-) (remote controlled bus index)
% 7 mBase, total MVA base of this machine, defaults to baseMVA
% (-) (machine impedance, p.u. on mBase)
% (-) (step up transformer impedance, p.u. on mBase)
% (-) (step up transformer off nominal turns ratio)
% 8 status, 1 - machine in service, 0 - machine out of service
% (-) (% of total VARS to come from this gen in order to hold V at
%      remote bus controlled by several generators)
% 9 Pmax, maximum real power output (MW)
% 10 Pmin, minimum real power output (MW)
%
% Branch Data Format

```

```

% 1 f, from bus number
% 2 t, to bus number
% (-) (circuit identifier)
% 3 r, resistance (p.u.)
% 4 x, reactance (p.u.)
% 5 b, total line charging susceptance (p.u.)
% 6 rateA, MVA rating A (long term rating)
% 7 rateB, MVA rating B (short term rating)
% 8 rateC, MVA rating C (emergency rating)
% 9 ratio, transformer off nominal turns ratio (= 0 for lines )
% (taps at 'from' bus, impedance at 'to' bus, i.e. ratio = Vf / Vt)
% 10 angle, transformer phase shift angle (degrees)
% (-) (Gf, shunt conductance at from bus p.u.)
% (-) (Bf, shunt susceptance at from bus p.u.)
% (-) (Gt, shunt conductance at to bus p.u.)
% (-) (Bt, shunt susceptance at to bus p.u.)
% 11 initial branch status, 1 - in service, 0 - out of service
%
% (+) Area Data Format
% 1 i, area number
% 2 price_ref_bus, reference bus for that area
%
% (+) Generator Cost Data Format
% NOTE: If gen has n rows, then the first n rows of gencost contain
% the cost for active power produced by the corresponding generators.
% If gencost has 2*n rows then rows n+1 to 2*n contain the reactive
% power costs in the same format.
% 1 model, 1 - piecewise linear, 2 - polynomial
% 2 startup, startup cost in US dollars
% 3 shutdown, shutdown cost in US dollars
% 4 n, number of cost coefficients to follow for polynomial
% (or data points for piecewise linear) total cost function
% 5 and following, cost data, piecewise linear data as:
% x0, y0, x1, y1, x2, y2, ...
% and polynomial data as, e.g.:
% c2, c1, c0
% where the polynomial is  $c0 + c1 * P + c2 * P^2$ 
%
% << this file created [97-Aug-26 12:29:15] by PB::System version 1.3 >>

%%----- Power Flow Data -----%%
%% system MVA base
baseMVA = 10.0000;

%% bus data
bus = [
    1 3 0.0 0.0 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.0000 0.8000;
    2 1 0.1000 0.0600 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
    3 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
    4 1 0.1200 0.0800 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
    5 1 0.0600 0.0300 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;

```

```

6 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
7 1 0.2000 0.1000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
8 1 0.2000 0.1000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
9 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
10 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
11 1 0.0450 0.0300 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
12 1 0.0600 0.0350 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
13 1 0.0600 0.0350 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
14 1 0.1200 0.0800 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
15 1 0.0600 0.0100 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
16 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
17 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
18 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
19 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
20 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
21 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
22 1 0.0900 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
23 1 0.0900 0.0500 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
24 1 0.4200 0.2000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
25 1 0.4200 0.2000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
26 1 0.0600 0.0250 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
27 1 0.0600 0.0250 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
28 1 0.0600 0.0200 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
29 1 0.1200 0.0700 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
30 1 0.2000 0.6000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
31 1 0.1500 0.0700 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
32 1 0.2100 0.1000 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
33 1 0.0600 0.0400 0.0 0.0 1 1.0000 0.0000 12.6600 1 1.1000 0.8000;
];

%% generator data
gen = [
    1 4.0000 3.0000 3.5000 0.0000 1.0000 10.0000 1 5.0000 0.0000;
];

%% branch data
branch = [
    1 2 0.0058 0.0029 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    2 3 0.0308 0.0157 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    3 4 0.0228 0.0116 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    4 5 0.0238 0.0121 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    5 6 0.0511 0.0441 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    6 7 0.0117 0.0386 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    7 8 0.0444 0.0147 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 0;
    8 9 0.0643 0.0462 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    9 10 0.0651 0.0462 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 0;
    10 11 0.0123 0.0041 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    11 12 0.0234 0.0077 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    12 13 0.0916 0.0721 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    13 14 0.0338 0.0445 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
    14 15 0.0369 0.0328 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 0;
];

```

```

15 16 0.0466 0.0340 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
16 17 0.0804 0.1074 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
17 18 0.0457 0.0358 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
2 19 0.0102 0.0098 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
19 20 0.0939 0.0846 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
20 21 0.0255 0.0298 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
21 22 0.0442 0.0585 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
3 23 0.0282 0.0192 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
23 24 0.0560 0.0442 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
24 25 0.0559 0.0437 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
6 26 0.0127 0.0065 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
26 27 0.0177 0.0090 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
27 28 0.0661 0.0583 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
28 29 0.0502 0.0437 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
29 30 0.0317 0.0161 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
30 31 0.0608 0.0601 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
31 32 0.0194 0.0226 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
32 33 0.0213 0.0331 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 0;
8 21 0.1248 0.1248 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
9 15 0.1248 0.1248 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
12 22 0.1248 0.1248 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
18 33 0.0312 0.0312 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 1;
25 29 0.0312 0.0312 0.0000 16.0000 16.0000 16.0000 0.0000 0.0000 0;

```

```
];
```

```
%%----- OPF Data -----%%
```

```
%% area data
```

```
area = [
```

```
1 1;
```

```
];
```

```
%% generator cost data
```

```
gencost = [
```

```
2 0.00 0.00 3 0.02 2 0;
```

```
];
```

```
return;
```

ÖZGEÇMİŞ

Doğum tarihi 24.09.1971

Doğum yeri Kayseri

Lise 1985-1989 Kayseri Merkez Teknik Lisesi

Lisans 1990-1994 Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak.
Elektrik Mühendisliği Bölümü

Yüksek Lisans 1994-1997 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Müh. Anabilim Dalı

Doktora 1997-2003 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Müh. Anabilim Dalı

Çalıştığı kurum(lar)

1995-Devam ediyor YTÜ Elektrik Müh. Bölümü Araştırma Görevlisi

