İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GENERATÖR BİRİM KATKI SORUNUNUN GENETİK ALGORİTMALAR İLE ÇÖZÜLMESİ

YÜKSEK LİSANS TEZİ

Müh. Mehmet Emin SAPMAZ

Anabilim Dalı: Elektrik Mühendisliği

Programı: Elektrik Mühendisliği

Mayıs 2004

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GENERATÖR BİRİM KATKI SORUNUNUN GENETİK ALGORİTMALAR İLE ÇÖZÜLMESİ

YÜKSEK LİSANS TEZİ

Müh. Mehmet Emin SAPMAZ

(504011030)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2004
Tezin Savunulduğu Tarih : 21 Mayıs 2004

Tez Danışmanı : **Doç. Dr. Hasan DAĞ**

Jüri Üyeleri : **Prof. Dr. Celal KOCATEPE**

Doç. Dr. Mustafa Bağrıyanık

Mayıs 2004

ÖNSÖZ

Elektrik enerji sistemlerinde üretim birimlerinin çalışmasının düzenlenmesi son 40 yıldır üzerinde aktif olarak çalışılan bir problemdir. Genel olarak bir eniyileme sorunu olmasına rağmen birçok kısıtlamalarının olması ve analitik çözüm yolunun bulunmaması problemi zorlaştırmaktadır.

Günümüzde bilgisayarların hızlı işlem güçleri ve uygulama geliştirmenin kolaylaşması sayesinde, programlama algoritmaları bu tip problemlere çözüm bulmak için yaygın biçimde kullanılmaktadır. Olasılıksal çözüm yapan genetik algoritmalar bugüne kadar başarısını kanıtlamıştır. Özellikle analitik çözümü olmayan ve çözüm uzayının taranması gereken problemlerde Genetik Algoritmalar (GA) oldukça başarılı sonuçlar vermektedir.

Bu çalışmada birim katkı sorununun yapısı incelenmiş ve GA uygulanarak birim katkı sorununa çözüm bulan bir uygulama yapılmıştır.

Bana böyle bir çalışmayı yapma olanağı tanıyan, beni yönlendiren, karşılaştığım problemleri aşmamda bana yardımcı olan ve bilimsel üretim konusunda beni cesaretlendiren değerli hocam Doç. Dr. Hasan DAĞ'a en içten teşekkürlerimi sunarım.

Nisan 2004 Mehmet Emin SAPMAZ

İÇİNDEKİLER

KISALTMALAR ..... **v**

TABLO LİSTESİ ..... **vi**

ŞEKİL LİSTESİ ..... **vii**

SEMBOL LİSTESİ ..... **viii**

ÖZET ..... **ix**

SUMMARY ..... **xi**

1. GİRİŞ **1**

2. BİRİM KATKI SORUNU **4**
2.1. Giriş 4
2.2. Birim Katkı Sorununa Giriş 4
2.3. Birim Katkı Sorunu Maliyetleri 5
2.3.1. Yakıt Maliyeti 5
2.3.2. Başlangıç Maliyeti 5
2.3.3. Durdurma Maliyeti 6
2.4. Birim Katkı Sorunu Kısıtları 6
2.4.1. Yük Dengesi 6
2.4.2. Yedek Güç 7
2.4.3. Generatör Çıkış Limitleri 7
2.4.4. En düşük Devrede/Devre dışı Kalma Süresi 7
2.4.5. Eğim Oranı Limitleri 8
2.5. Amaç Formülasyonu 8
2.6. Sonuç 8

3. GENETİK ALGORİTMALAR **10**
3.1. Giriş 10
3.2. Tarihçe 10
3.3. Biyolojik Geçmiş ve Terimler 11
3.4. Genetik Algoritma İşlevleri 12
3.4.1. Kromozom Kodlaması 12
3.4.2. Seçim İşlevi 14
3.4.3. Çaprazlama (Crossover) operatörü 16
3.4.4. Mutasyon İşlevi 17
3.5. Genetik Algoritmaların İşlem Basamakları 18
3.6. Sonuç 20

4. BİRİM KATKI SORUNUNA GA UYGULANMASI ÜZERİNE YAPILAN ÇALIŞMALAR **22**

4.1. Giriş 22
4.2. Mevcut çalışmalar 22
4.3. Sonuç 30

5. BİRİM KATKI SORUNUNA GA UYGULANMASI 32
5.1. Giriş 32
5.2. Birim Katkı Modeli 32
5.3. Genetik Algoritmanın Aşamaları 34
5.3.1. Kromozom Kodlaması 34
5.3.2. Çaprazlama 35
5.3.3. Mutasyon 36
5.3.4. Elitisizm 36
5.3.5. Uygunluk Fonksiyonu 36
5.3.6. Seçim İşlevi 37
5.3.7. Kısıt Kontrol ve Düzeltme Operatörü 38
5.3.8. Penaltı Operatörü 38
5.3.9. Başlangıç Popülasyonunun Oluşturulması 38
5.3.10. Ekonomik Yük Dağılımı 39
5.4. Uygulanan Genetik Algoritma 39
5.5. Uygulama Ortamı 41
5.6. Uygulama Sonucu 41

6. SONUÇLAR 44
6.1. Sonuçlar 45
6.2. Yapılabilecek Çalışmalar ve Öneriler 46

KAYNAKLAR 48

EK A 51

EK B 53

ÖZGEÇMİŞ 55

KISALTMALAR

BKS : Birim Katkı Sorunu
GA : Genetik algoritma(lar) (Genetic Algorithms)
MDT : En az devre dışı kalma süresi (Minimum Down Time)
MF : Maliyet fonksiyonu
MUT : En az devrede kalma süresi (Minimum Up Time)
TM : Toplam Maliyet
UC : Birim katkı (Unit Commitment)
UCP : Birim katkı sorunu (Unit Commitment Problem)

TABLO LİSTESİ

Sayfa No

Tablo 3.1. Değer tabanlı kromozom kodlaması örneği 13
Tablo 4.1 Değişik seçim işlevleri için çözüm süreleri ve sonuçların kalitesi. 24
Tablo 4.2 Değişik çaprazlama-mutasyon değerleri için çözüm süreleri ve sonuçların kalitesi. 24
Tablo 4.3. Yang ve arkadaşlarının paralelleştirme yöntemlerinin diğer yöntemlerle karşılaştırılması. 30

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 Termik generatörlerin güç-maliyet eğrisi. 5
Şekil 3.1 Ağaç yapısı kodlama örneği..... 13
Şekil 3.2 Rulet tekerleği seçimi 15
Şekil 3.3 Tek noktalı çaprazlama..... 16
Şekil 3.4 İki noktalı çaprazlama..... 17
Şekil 3.5 Düzgün (Uniform) çaprazlama. 17
Şekil 4.1 Kaynak [19] tarafından önerilen kromozom kodlaması 23
Şekil 4.2 Değişik çaprazlama-mutasyon değerleri için çözümün yakınsaması. 25
Şekil 4.3 Swarup ve Yamashiro tarafından önerilen iki boyutlu kromozom kodlaması. 25
Şekil 4.4 Yang ve arkadaşlarının kullandıkları paralel işlemci yapıları. Efendi köle ve çift yönlü halka..... 29
Şekil 4.5 Yang ve arkadaşları tarafından önerilen paralel yöntemde işlemci sayısına göre işlem hızı değişimi. 29
Şekil 5.1 Günlük yük eğrisi grafiği. 32
Şekil 5.2 Gerçeklenen uygulamada kromozomların iki boyutlu gösterimi..... 35
Şekil 5.2. Uygunluk Fonksiyonunun maliyete göre değişimi 37
Şekil 5.3 Gerçeklenen uygulamada seçim işlevlerinin karşılaştırılması..... 42
Şekil 5.4 Gerçeklenen uygulamada işlem süresinin kuşak sayısına göre değişimi.... 43

SEMBOL LİSTESİ

a_i, b_i, c_i	: Generatör yakıt maliyeti katsayıları
C_f	: Generatör yakıt maliyeti
P	: Güç
C_c	: Generatörün soğuk başlatma maliyeti
$\sigma_i, \delta_i, \tau_i$	: Soğuk başlatma maliyeti katsayıları
T^{OFF}	: Generatörün devre dışı kaldığı süre
T^{ON}	: Generatörün devrede kaldığı süre
C_b	: Generatörün yatık (bunked) başlatma maliyeti
ρ	: Yatık başlatma maliyeti katsayısı
C_d	: Generatörün durdurma maliyeti
j	: saat, j=1, 2, ... T
i	: Birim numarası, i=1, 2, ...N
$P_{G,j}$	: Generatörün j saati için ürettiği güç
$P_{D,j}$	: j saati için talep edilen güç
$P_{R,j}$	: j saati için ihtiyaç duyulan yedek güç
$P_{min,i}$	: i generatörünün sağlayabileceği en düşük çıkış gücü
$P_{max,i}$	: i generatörünün sağlayabileceği en yüksek çıkış gücü
P_{UpRamp}	: Generatörün artan güç eğim oranı
$P_{DownRamp}$	: Generatörün azalan güç eğim oranı

ÖZET

GENERATÖR BİRİM KATKI SORUNUNUN GENETİK ALGORİTMALAR İLE ÇÖZÜLMESİ

Günlük yük eğrisindeki değişimler elektrik enerjisi üretiminin bir saatlik periyotlar ile gün bazında planlanmasını gerektirir. Bu planlama yapılırken sistemin en düşük maliyet ile enerji üretmesi istenir. Enerji sisteminin ekonomik işletilmesi, günlük değişken yük eğrisinde hangi santralin devreye alınıp çıkarılacağı, ne kadar kapasite ile işletileceği belirleyerek gerçeklenir. Çözüme ulaşmak için ekonomik işletmenin yanında, enerji üretim birimlerinin dinamik değişim kısıtlamaları, enerji sisteminin fiziksel kısıtlamaları gibi birçok kısıtlamadan oluşan bir problem ortaya çıkar, bu probleme "birim katkı sorunu" adı verilir ve matematiksel olarak kısıtlılık içeren bir eniyileme sorunudur.

Birim katkı sorunu analitik çözümü olmayan bir problemdir. Problemin kısıtları çözüm uzayındaki geçerli çözümler arasından çok az bir bölümünün uygun olmasına neden olmaktadır. Birim katkı sorununu üzerinde birçok çalışma yapılmasının bir nedeni problemin tasarruf potansiyelidir. Tahminlere göre 10000 MW kurulu gücü bulunan bir sistemde yapılacak günlük %1 düzeyindeki bir maliyet düşüşü, yıllık 20 milyon Amerikan doları tasarruf sağlamaktadır.

Genetik Algoritmalar doğada canlılarda görülen evrimin benzerinin problem çözümüne uygulanmasıyla oluşmuş bir tekniktir. Son 45 yılda yapılan çalışmalar, uygulanan birçok alan ve sonuçlar bu tekniğin başarısını kanıtlamaktadır. Genetik algoritmalar özellikle analitik çözümü olmayan veya çok zor yapılabilen, çözüm uzayının taranmasını gerektiren uygulamalarda başarılı sonuçlar sağlamaktadır. Birim katkı sorunu GA'nın başarılı olduğu problem tiplerinden biri olduğunda ve günümüze kadar yapılan çalışmalar bu alanda iyi sonuçlar verdiğinden bu çalışmada birim katkı sorununun çözümü için GA kullanılmıştır.

Çalışmada öncelikle birim katkı sorunu kısıtlarıyla birlikte incelenmiş ve matematiksel formülasyonu verilmiştir. Sonraki aşamada problemin çözümüne

uygulanacak olan genetik algoritmalardan bahsedilmiştir. Bugüne kadar birim katkı sorununu GA kullanarak çözüm yapmış çalışmalar incelenmiş, birbirlerine üstünlükleri ve önerdikleri yeniliklere değinilmiştir. Son olarak da çözümü yapılacak örnek sistem verilmiş ve sonuçlar listelenmiştir.

Elde edilen sonuçlar GA'ın birim katkı sorununun çözümünde başarılı olduğunu göstermektedir. Bunun yanında GA'nın parametreleri arasında mutasyon ve çaprazlama oranını çözümün yakınsamasını ve kalitesini etkilemektedir. Seçim işlevlerinin çözümün kalitesinde önemli bir etkisi olmadığı görülmüştür.

SUMMARY

A GENETIC ALGORITHMS SOLUTION TO THE UNIT COMMITMENT PROBLEM

The cyclical demand for electricity over a course of a day requires utility company to plan for power generation on an hourly basis for up to 24 or 48 hours. The main objective of this planning is to reduce the generation costs. While scheduling generators, it should be considered that electrical system has dynamic, thermal and mechanical constraints. The scheduling of generators with regard to these constraints is called unit commitment problem (UCP).

Unit Commitment Problem is a large scale constrained optimization problem. Because of the constraints, it has few feasible solutions in search space. The main aim of UCP is to minimize generation costs, since about a 1% reduces of generation costs saves 20 millions US Dollars per year. That saving potential makes this problem an intensive research area.

Genetic algorithms (GA) are problem solving techniques, which is the analogous of evolution in nature. . GA searches the solution space heuristically for a minimum or maximum value. Throughout this search it evolves a set of solutions for best value. Over 45 years of applications on various areas shows that GA is a feasible solution technique for non-analytical large scale problems

The objective of this study is to apply GA to solve the UCP and to find optimum GA parameters for this problem. Firstly, UCP problem is explained with its constraints and mathematical formulation. Then GA and its operators are mentioned in details. The research results obtained in the area of UCP using GA are compared with respect to their innovations and GA operators. Finally, the mathematical model of the application and the steps of the applied GA are explained.

The results of this work show that GA is a feasible solution technique for UCP. The mutation and crossover rate of the GA influence the solution. On the other hand the selection operators have no influence on the convergence to the optimum solution.

Also, elitism is a requirement for preserving the feasible solution over the populations.

1. GİRİŞ

Elektrik enerji sistemlerinin temelinde enerjinin üretimini sağlayan üretim birimleri bulunur. Bir veya birkaç generatörden oluşan bu birimlerce üretilen elektrik, enerji iletim ve dağıtım hatlarından geçerek tüketicilere ulaşır. Enerjinin üretilmesinden sorumlu birimlerden her birinin sistem karakteristiği ve yakıtı farklı olabilir, dolayısıyla her birimin elektriği üretim maliyeti farklıdır. Belirli bir anda sistemin sağladığı toplam enerjinin maliyetini en düşük seviyede tutmak için sistemdeki üretim birimleri arasından üretim maliyeti en az olanlar devrede olmalıdır ki günlük toplam üretim maliyeti en az seviyede kalsın.

Günlük olarak sistemden talep edilen güç sabit olmadığından, enerji üretim birimlerinin kısa bir zaman dilimi için devreye alınması ve devre dışı kalmasının düzenlenmesi gerekir. Bu düzenleme genellikle bir saatlik periyotlar ile günlük veya iki günlük olarak yapılır. Üretim birimlerinden alınan enerjinin toplam maliyetini en düşük miktarda tutmak amacıyla generatörlerin devreye alınıp çıkarılmalarının belirli bir zaman aralığı için düzenlenmesi problemine birim katkı sorunu adı verilir.

Tasarruf potansiyelinin yüksek olması nedeni ile birim katkı sorunu üzerine birçok çalışma yapılmıştır. Yapılan tahminlere göre; 10000 MW büyüklüğündeki kurulu güç için, işletme maliyetinde yapılan her %1'lik düşüş, yıllık ortalama 20 milyon Amerikan doları tasarruf sağlamaktadır [1].

Üretim birimlerinin çalışmasının planlanması, devreye girecek/çıkacak birimlerin seçilmesinin yanında devrede olan birimler arasındaki yük dağılımının belirlenmesini gerektirir. Bu nedenle birim katkı sorunu, ekonomik yük dağılımı sorununun da çözümünü gerektirmektedir. Birim katkı sorunu yük tahmininden farklı olarak 24 veya 48 saat gibi yakın gelecek için sürekli olarak yapılır.

Birim katkı sorunu kendi içerisinde birçok kısıtları bulunan bir problemdir. Sistemin yük dengesi, generatörlerin kapasite sınırları, en düşük devrede ve devre dışı kalma süreleri, izin verilen yük değişim eğimleri ve korunması gereken yedek güç en önemli kısıtlardır. Birçok kısıtlarının olması, karmaşıklığı ve analitik çözüm yolunun olmaması nedeniyle çözümü zor, karmaşık bir eniyileme problemidir. Matematiksel

olarak adlandırılırsa, birim katkı sorunu; doğrusal olmayan, geniş ölçekli, karışık tamsayı kombinasyonlu eniyileme problemidir.

Diğer eniyileme problemlerinde kullanılan bütün yöntemler birim katkı sorununun çözümü için kullanılmıştır. Bunlar arasında dinamik programlama, karışık tamsayı programlama, doğrusal programlama, öncelik listesi, Lagrange rahatlatması gibi klasik eniyileme yöntemleri yanında, yapay sinir ağları, tabu taraması, simulated annealing ve genetik algoritmalar gibi heuristik yöntemler de sayılabilir [2-14].

Birim katkı sorununda çözüm uzayı çok büyüktür ve analitik çözüm yoktur. Bu da çözüm uzayında olasılıksal olarak gezinen yöntemlerin başarısını arttırmaktadır. Genetik algoritmaların olasılıksal oluşu ve doğrusal olmayan programlamadaki başarıları nedeniyle bu tip uygulamalardaki başarısını kanıtlamıştır. Günümüzde bilgisayarların belleklerinin büyümesi ve işlem hızlarının artması genetik algoritmaların daha kısa sürede kaliteli sonuçlar üretmesine imkan sağlamaktadır.

Genetik algoritmalar en genel haliyle doğada canlılarda meydana gelen evrimin problem çözümüne aktarılmasıdır. Birçok bireyi barındıran ve rasgele oluşturulan popülasyondan seçilen ebeveynlerden türetilen bireyler ile yeni bir popülasyon oluşturulur. Oluşturulan her popülasyonda bulunan çözümlerin kalitesi daha da artar. Önceden belirlenen kuşak sayısına ulaşıldığında veya çözüm belirli bir değere ulaştığında yeni popülasyon üretimi durdurulur [14-15].

Bu çalışmadaki amaç birim katkı sorununun incelenmesi, genetik algoritmalar uygulayarak problemin çözümünün yapılmasıdır. Bunun için yapılmış mevcut çalışmalar incelenecek, bunlar arasından bazılarının uygulanabilirliği sınanacaktır. Yapılan birim katkı sorunu çözüm uygulaması GA'nın parametreleri değiştirilerek bulunan sonuçlar karşılaştırılacaktır.

Çalışma altı bölümden oluşmaktadır. İlk bölümde çalışmanın içeriği ve bölümleri anlatılmış, ikinci bölümde birim katkı sorunu, kısıtlar, formülasyonu ve problemin detayları açıklanmıştır. Üçüncü bölümde genetik algoritmalar anlatılmıştır. Genetik algoritmaların bölümleri ve her bölümde kullanılan yöntemlerden bahsedilmiştir. Dördüncü bölümde birim katkı sorununa genetik algoritmaların uygulanması ile ilgili günümüze kadar yapılmış çalışmalardan bazıları incelenmiştir. Yapılmış olan çalışmalar önerdikleri yöntemler ve kullandıkları GA operatörleri ile farklılık yaratmaktadır. Beşinci bölümde birim katkı sorununa GA uygulamasının yapılması

ve seçim işlevi, mutasyon oranı gibi GA parametrelerinin değiştirilerek karşılaştırılması anlatılmaktadır. Son olarak elde edilen sonuçların genel bir değerlendirmesi ve ilerde yapılabilecek çalışmalar için öneriler tezin altıncı bölümünde sunulmuştur.

2. BİRİM KATKI SORUNU

2.1. Giriş

Günlük yük (elektrik enerjisi) talebindeki (eğrisindeki) değişimler elektrik enerjisi üretiminin saat bazında, bir veya iki gün için planlanmasını gerektirir. Bu planlama yapılırken sistemin en düşük maliyet ile enerji üretmesi istenir. Enerji sisteminin ekonomik işletilmesi, günlük değişken yük eğrisinde hangi santralin devreye alınıp çıkarılacağı, ne kadar kapasite ile işletileceği belirleyerek gerçeklenir. Çözüme ulaşmak için ekonomik işletmenin yanında, enerji üretim birimlerinin dinamik değişim kısıtlamaları, enerji sisteminin fiziksel kısıtlamaları gibi birçok kısıtlamadan oluşan bir problem ortaya çıkar, bu probleme "birim katkı sorunu" adı verilir ve gerçekte bir eniyileme sorunudur.

Bu bölümde birim katkı sorununa giriş yapılacak ve problemi matematiksel olarak tanımlanacaktır. Daha sonra birim katkı sorununun karmaşıklaşmasına neden olan kısıtları detaylı olarak incelenecek ve en sonunda problemin genel formülasyonunu verilecektir.

2.2. Birim Katkı Sorununa Giriş

Birim katkı sorunu, bir enerji sisteminde 24 veya 48 saatlik gelecek için sistemin durumu ve maliyet göz önüne alınarak en uygun generatör gruplarının devreye alınıp çıkarılmasının belirlenmesidir. Genel olarak birim katkı sorunu maliyetlerden ve sistem kısıtlamalarından oluşmaktadır.

Maliyetler generatörler devrede iken oluşan yakıt maliyetleri, generatörlerin devreye girmesi sırasındaki başlatma maliyetleri ve generatörlerin durdurulması sırasındaki durdurma maliyetleridir [16].

Birim katkı sorununun kısıtları sistemin fiziksel özelliklerinden kaynaklanmaktadır. Generatörlere güç sağlayan türbinlerin ısıtılması ihtiyacı, dönmeye başlamasının ve durdurulmasının belirli süre alması, karşılanması gereken talep güç ve yedek güç kısıtların nedenleridir.

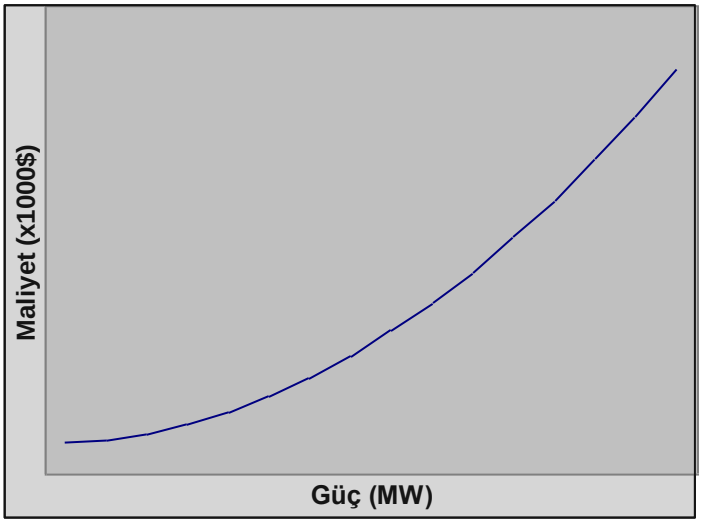
2.3. Birim Katkı Sorunu Maliyetleri

2.3.1. Yakıt Maliyeti

Yakıt maliyeti generatörler devrede iken oluşan ve generatörden çekilen güce bağlı olan bir fonksiyondur. Termik üretim birimlerinde genel olarak iki tip türbinden oluşmaktadır, buhar türbinleri ve yanmalı türbinler. Buhar türbinleri fosil yakıtların yanması veya nükleer enerjiden elde edilen buhar ile döndürülür. Yanmalı türbinlerde ise doğal gaz veya distile edilmiş petrolün yanması ile hareket elde edilir.

Aralarındaki farklara rağmen her iki tipteki üretim birimleri benzer şekilde modellenebilir. Termik üretim birimlerinin yakıt maliyeti (2.1) deki gibi formüle edilebilir. Üretim birimlerinin çıkış güçlerine bağlı maliyetleri genel olarak şekil 2.1 ile gösterilebilir [16].

$$C_f = a_i + b_i P + c_i P^2 \tag{2.1}$$



Şekil 2.1 Termal generatörlerin güç-maliyet eğrisi.

2.3.2. Başlangıç Maliyeti

Generatörlerin devre dışı durumdan devreye alınırken oluşan maliyettir. Termik üretim birimleri iki türlü devre dışı bırakılabilir, bunlardan birincisi generatörün şebeke bağlantısını kestikten sonra türbinlerin soğumaya bırakılmasıdır. Eğer birim ortam sıcaklığına kadar soğuduktan sonra türbinler tekrar ısıtılıp birim devreye alınırsa bu duruma soğuk başlatma adı verilir. Soğuk başlatma maliyeti (2.2) ile verilmiştir [16].

$$C_c = \sigma_i + \delta_i \left[1 - e^{(-T_{i,j}^{OFF} / \tau_i)}\right] \quad (2.2)$$

Termik birimleri devre dışı bırakmanın bir diğer yolu türbinleri yatık (banked) konumda bırakmaktır. Yatık konumda iken generatörün şebeke bağlantısı kesilir ve generatör elektrik üretmez, fakat türbinlerin soğumasına da izin verilmez, bu sayede generatörün devreye alınması kolaylaşır. Üretim birimleri kısa süreli olarak devre dışı kalacakları zaman yatık konumda bırakılırlar. Yatık konumdaki üretim biriminin başlatma maliyeti (2.3) ile verilmiştir.

$$C_b = \rho T^{OFF} \quad (2.3)$$

Burada ρ, yatık konum maliyet sabiti ve T^{OFF}, yatık konumda kalma süresidir [16].

2.3.3. Durdurma Maliyeti

Generatörlerin devrede iken devre dışına alınması sırasında oluşan maliyettir, bu maliyet yakıt veya başka bir sınırlamadan gelmemektedir, genellikle durdurmak için gerekli çalışan maliyeti olarak düşünülür. Birimleri durdurmak için birim çalışanları hariç fazladan işgücüne gerek duyulmadığından genellikle durdurma maliyeti sıfır kabul edilir [17].

$$C_d = 0 \quad (2.4)$$

2.4. Birim Katkı Sorunu Kısıtları

Birim katkı sorununu zor bir eniyileme problemi haline getiren neden kısıtlarıdır. Birim katkı sorunu kısıtları iki ana başlık altında toplanabilir. Bunlar sistem kısıtları ve birim kısıtlarıdır. Sistem kısıtları, sistemin yük dengesinin sağlanması ve yedek yük ihtiyacıdır. Birim kısıtları ise generatörlerin fiziksel ve karakteristik özelliklerinden kaynaklanan kısıtlardır [16,17].

2.4.1. Yük Dengesi

Üretim birimleri ve talep yükten oluşan bir sistemde üretim birimlerinin her t anı için ürettiği güç, talep edilen güce eşit veya fazla olmalıdır. Talep edilen güç sadece tüketim birimlerinin harcadığı güç değildir, üretim birimleri açısından bakıldığından sistemdeki bütün kayıplar da talep gücüne dahildir. Yük dengesi (2.5) deki gibi formüle edilebilir [17,18].

$$P_{G,j} - P_{D,j} \geq 0 \tag{2.5}$$

Burada P_G, generatörlerin toplam gücü, P_D sistemin çektiği toplam güç ve j=1,2....T saatleridir.

2.4.2. Yedek Güç

Elektrik enerji sisteminde üretimin talep edilen yükü karşılamasının yanında sistemin güvenilirliği için belirli bir orandaki gücü istenildiğinde karşılaması istenir. Üretim birimlerinden bir veya birkaçının herhangi bir nedenle devre dışı kalması durumunda kullanılacak olan bu güce yedek güç (spinning reserve) denir. Yedek güç genellikle generatörlerden bir kısmı en üst güç limitinin altında çalıştırılarak sağlanır. İhtiyaç duyulduğunda ilgili generatörlerin sağladığı güç arttırılarak devreye sokulur. Yedek güç ifadesi (2.6) da verilmiştir [17,18].

$$P_{G,j}^{\max} \geq P_{D,j} + P_{R,j} \tag{2.6}$$

Burada $P_{R,j}$, j. saat verilmesi gereken yedek gücü göstermektedir.

2.4.3. Generatör Çıkış Limitleri

Her bir generatörden çekilen güç (2.7) de formüle edildiği gibi ilgili generatörün sağlayabileceği en düşük ve en yüksek güç değerleri arasında olmalıdır [16-18].

$$P_{\min i,j} \leq P_{i,j} \leq P_{\max i,j} \tag{2.7}$$

Burada $P_{i,j}$, i. generatörün j. saatteki çıkış gücünü, min ve max indisleri ise ilgili generatörün sağlayabileceği en düşük ve en yüksek güç miktarını belirtmektedir.

2.4.4. En düşük Devrede/Devre dışı Kalma Süresi

Generatör türbinleri ani sıcaklık değişiminden zarar görebilir, ayrıca generatörlerin başlatılması ve durdurulmasının maliyeti olduğundan bu maliyeti düşük tutmak için başlama ve durdurmadan sonra belirli bir süre geçmesi istenir. Bütün bunlar birim katkı sorunu için kısıtlama oluşturur.

Generatörlerin en az devrede kalma süresi (2.8) ile gösterilmiştir [17,18].

$$\sum_{j=1}^{T} \left| T_j^{ON} - MUT \right| \geq 0 \tag{2.8}$$

Burada MUT, ilgili generatörün en az devrede kalma süresidir.

Generatörlerin en düşük devre dışı kalma süresi (2.9) daki gibi formüle edilebilir [17,18]. (2.9) da bulunan MDT, ilgili generatörün en az devre dışı kalma süresidir.

$$\sum_{j=1}^{T} \left| T_j^{OFF} - MDT \right| \geq 0 \tag{2.9}$$

2.4.5. Eğim Oranı Limitleri

Generatörlerin termik ve mekanik sınırları generatörden çekilen yükte ani değişimlere izin vermez. Bunun sonucu olarak generatörden çekilen gücün saatlik olarak değişimi belirli bir değeri aşmamalıdır. Bazı generatörler için çekilen gücün artması ve azalması farklı oranlarda olmalıdır, buna göre en genel hali ile birimlerin eğim oranı kısıtları (2.10) daki şekilde ifade edilebilir.

$$\begin{aligned} P_j - P_{j-1} &\leq P_{UpRamp} \\ P_{j-1} - P_j &\leq P_{DownRamp} \end{aligned} \tag{2.10}$$

Burada P_{UpRamp}, ilgili generatörün bir saat içinde gücünde meydana gelebilecek en fazla artış miktarıdır, $P_{DownRamp}$ ise ilgili generatörün bir saat içinde gücünde meydana gelebilecek en fazla azalma miktarıdır [19].

2.5. Amaç Formülasyonu

Birim katkı sorununun amacı en düşük toplam üretim maliyetini, kısıtları da ihlal etmeden elde etmektir. Buna göre (2.11) de bir arada verilen kısıtlar göz önüne alınarak (2.12) deki toplam üretim maliyeti en düşük olmalıdır.

$$\text{AMAÇ}=\left| \sum_{j=1}^{T} \sum_{i=1}^{n} (a_i + b_i P_{i,j} + c_i P_{i,j}^2) + \sum_{j=1}^{T} \sum_{i=1}^{n} \sigma_i + \delta_i \left[1 - e^{(-T_{i,j}^{OFF} / \tau_i)} \right] \right|_{\min} \tag{2.11}$$

2.6. Sonuç

Birim katkı sorunu bir eniyileme problemidir, fakat kısıtları ve analitik çözüm yöntemi bulunmaması nedeniyle çözümü zorlaşmaktadır. Çözüm için amaçlanan en düşük toplam yakıt maliyetini bulmaktır. Birim katkı sorununda yakıt maliyeti, üretim yakıt maliyeti yanında generatörlerin başlatma ve durdurma maliyetlerini de içermektedir.

Birim katkı sorununun kısıtları generatörlerin fiziksel sınırlarından kaynaklanmaktadır. Generatörlerin hemen devreye alınması ve belirli bir süre geçmeden devre dışına çıkarılması mümkün olmamaktadır. Generatörlerden çekilecek güç de generatörün üretim sınırları içinde kalmalı ve generatörün karakteristiğinde verilen güç değişimleri aşılmamalıdır. Bunların yanında sistemde üretilen toplam güç talep gücünü (yük + kaçak) sağlamalı ve generatörler herhangi bir arıza durumunda önceden belirlenen oranda güç vermeye hazır olmalıdır.

3. GENETİK ALGORİTMALAR

3.1. Giriş

Genetik algoritmaların geçmişi atası olan evrimsel stratejiler ile birlikte 45 yıla dayanmaktadır. Bu bölümde öncelikle genetik algoritmaların tarihçesinden bahsedilecek, biyolojik temellerine ve genetik algoritmalar ile ilgili kullanılan terimlere değinilecektir. Daha sonra genetik algoritmaların çalışma ilkesi, aşamaları ve her bir aşamada kullanılan yöntemler anlatılacaktır.

3.2. Tarihçe

Genetik algoritmalar Darwin'in evrim teorisinden ilham alınarak geliştirilmiş olan evrimsel hesaplamanın bir koludur. Evrimsel hesaplama fikri ilk olarak 1960 yılında I. Rechenberg'in "Evolutionsstrategie" (evrimsel stratejiler) çalışmasıyla ortaya atılmıştır [20]. Temel olarak evrim stratejileri tekniği, olası çözüm adayı olarak, ana birey ve ana bireyden mutasyon yoluyla türetilmiş olan oğul bireyleri içermekteydi.

Bu tekniğin bir sonraki evresi olan evrimsel programlama (Evolutionary programming) günümüzde kullanılan Genetik Algoritmalara benzeyen bir yöntem olup Fogel, Owens ve Walsh tarafından geliştirilmiştir. Burada bir önceki aşamada tanıtılan özelliklere ek olarak yeni üretilen çözüm adaylarının belirli bir uygunluk fonksiyonu altında değerlendirilmesi eklenmiştir [15].

Genetik algoritma kelimesini ilk kullanan ve çalışmalarını 1967 yılında yayımlayan Bagel olmuştur, aynı tarihte bağımsız olarak Rossenberg de biyolojik ve benzetim esaslı bir çalışma yapmıştır [15].

Makina öğrenmesi (machine learning) konusunda çalışan John Holland 1975 yılında "Adaptation in Natural and Artificial Systems" adlı kitabını yayımlamıştır [21]. Bu çalışmada Holland tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılardan oluşan bir topluluğun çoğalma, çiftleşme, mutasyon vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturduğunu gözlemlemiş, doğadaki evrimsel işlemin benzeri bir yapı elde etmiştir. Holland'ın bu

çalışmasından sonra bu yöntemin adı *genetik algoritmalar* (kısaca GA) olarak yerleşmiş ve birçok çalışmanın önünü açmıştır.

3.3. Biyolojik Geçmiş ve Terimler

Yaşayan bütün organizmalar hücrelerden, her bir hücrede ise organizmanın bütün bilgilerini barındıran, DNA dizilerinden oluşan *kromozom*lar vardır. Kromozomlar organizmanın belirli bir bölümünün kodlamasını yapan *gen*lerden oluşur. Mesela göz rengini kodlayan bir gen gözün mavi, kahverengi vs. renklerden hangisini alacağının bilgisini içerir. Genler kodladıkları özelliklere göre değişik değerler alabilirler, genin alabileceği her bir değere *alel* (allele) adı verilir. Kromozom içerisindeki her bir genin yeri bellidir. Genetik malzemenin tamamını oluşturan kromozomların hepsine *genome* adı verilir [22].

Genetik algoritmalarda çözüm kümesini taramak için birçok kromozomdan oluşan bir yapı kullanılır. Bu yapıya *popülasyon adı verilir*. Popülasyondaki birey sayısı (kromozom) genelde sabit tutulur. GA'da popülasyondaki birey sayısı ile ilgili genel bir kural yoktur. Popülasyondaki kromozom sayısı arttıkça çözüme ulaşma süresi (yineleme sayısı) azalır fakat her bir basamaktaki işlem süresi artar [23].

İki kromozomun bir araya getirilerek genetik bilgilerinin değiştirilmesine *çaprazlama* adı verilir. Çaprazlama için birey seçilmesine ise *eşleme* (mating) denir. Çaprazlama genetik algoritmaların temelini oluşturmaktadır. Genellikle çaprazlama oranı %100 seçilerek yeni kuşağın oluşturulmasında bütün bireylerin çaprazlamadan geçmesi sağlanır. Çaprazlama sayesinde çözüm uzayında değişik noktalara ulaşılabilir. Bir kromozomum taşıdığı genetik bilgide bir nedene bağlı olmaksızın (rasgele) değişme olmasına *mutasyon* adı verilir. Mutasyon operatörü de çaprazlama kadar önemlidir. Mutasyon oranı %0 yapılır ise çözüm yerel minimumlara takılabilir. Mutasyon sayesinde çözüm uzayında daha fazla alan taranabilir. Çok büyük olan bir mutasyon değeri ise evrimin minimuma gidişini yavaşlatır ve işlem süresini arttırır [22, 23].

Temel olarak çaprazlama ve mutasyona uğrayacak bireylerin popülasyon içinden seçilmesi farklı yöntemlerle yapılabilir, bu yöntemlere *seçim operatörleri* denir. Seçim, algoritmanın çalışma hızını dolaylı olarak etkiler. Seçim yönteminin uygun olması, algoritmanın iyiye gidişini hızlandırır [22].

Üreme, çaprazlama ve mutasyon işlemleri sonrasında kuşakta bulunan en iyi uygunluk değerine sahip birey, sonraki kuşağa aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra oluşan yeni kuşağa, bir önceki kuşağın en iyilerinden (elit) belirli sayıdaki birey doğrudan aktarılır. Bu işleme *elitizm* adı verilir [22, 24].

Popülasyon içindeki bireylerin, diğer bir ifadeyle her biri bir çözüm grubunu ifade eden kromozomların ne kadar iyi bir çözüm olduğunu yani uygunluğunu değerlendirmekte kullanılan fonksiyona *uygunluk fonksiyonu* (fitness function) adı verilir. Bireyler seçim için değerlendirilirken ilgili bireye ait uygunluk fonksiyonu değerinden faydalanılır [22-24].

3.4. Genetik Algoritma İşlevleri

Genetik algoritmalarda kullanılan 4 temel işlev vardır, bunlar [3];

- Kromozom kodlaması
- Seçim İşlevi
- Çaprazlama
- Mutasyon

3.4.1. Kromozom Kodlaması

Kromozomlar GA'ın uygulandığı problemin parametrik bileşenleridir. Kromozomların uygunluk fonksiyonu hesaplanarak çözüm veya çözümü gösteren değer elde edilir. Kromozom problemin parametrelerini içerdiğinden uygulanan probleme göre kromozom yapısı ve kodlanması değişmektedir. Kromozomlar genellikle vektör şeklindedir, diğer taraftan pek yaygın olmasa da iki boyutlu kromozom da tanımlanabilir.

En genel kromozom kodlaması ikilik tabanda yapılan kodlamadır. John Holland"ın çalışmalarında genelde ikili tabanda, bit dizisi şeklinde kodlama sistemi kullanılmıştır. Bu kodlamada kromozomdaki genler 0 veya 1 sayılarından birini alabilir. Yalnızca 0 ve 1'in kullanılmasına rağmen, birçok olası farklı kromozom yapısı üretebilir, ama birçok problem için kullanılabilecek yeter düzeyde değildir, bazen çaprazlama veya mutasyon aşamalarından sonra düzeltme yapmak gerekebilir. İkili düzenin yeğlenmesinin nedeni, basit olması ve bilgisayar tarafından daha kolay

ve hızlı bir biçimde işlenebilmesidir. Tamsayılar veya ondalık sayılar ikili dizi şeklinde gösterildiğinde bu düzeltmelere ihtiyaç duyulur.

Bit dizisi kodlamadan sonra en çok kullanılan kodlama biçimi değer tabanlı kodlamadır. Değer tabanlı kodlamada her bir gen probleme göre tamsayı, ondalık sayı, kesirli sayı, harf vb. değerleri alır.

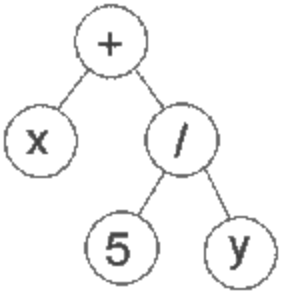
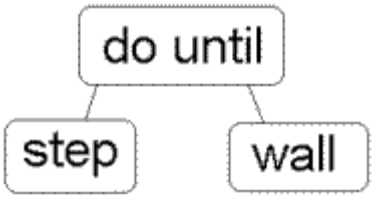
Tablo 3.1. Değer tabanlı kromozom kodlaması örneği

Kromozom A	1,2324 5,3243 0,4556 2,3293 2,4545
Kromozom B	A B D J E I F J D H D I E R J F D L D
Kromozom C	(arka), (arka), (sağ), (ön), (sol), (sağ)

Değer tabanlı kodlamada genlerin alacağı değerlerin bilgisayar hafızasında kapladığı yer programın hızını etkiler. Örneğin her bir geni 32 bitlik bir reel (float) sayıdan oluşan kromozomda gen sayısının artması programın bellek ihtiyacını dramatik bir biçimde arttırır.

Üçüncü tür kodlama şekli ağaç yapısı kodlamadır. Bu sistem esas olarak genetik programlamada programların ve ifadelerin değerlendirilmesinde kullanılır.

Ağaç yapısı kodlama sisteminde her kromozom bazı nesnelerin ağacıdır. Bunu, programlama dilindeki komutlar ve fonksiyonlar gibi düşünebiliriz (Bkz. Şekil 3.1).

Kromozom A	Kromozom B
	
(+ x (/ 5 y)) işlemler	(do_until step wall) komutlar

Şekil 3.1 Ağaç yapısı kodlama örneği

Ağaç yapısı kodlaması programların değerlendirilmesinde kullanılır, örneğin LISP programlama dili bu yapıya sıklıkla başvurur. Giriş ve çıkış değerlerinden ilgili fonksiyonun bulunması probleminde bu kodlama sistemi kullanılır [22].

3.4.2. Seçim İşlevi

Bu prosedürde; bireysel diziler, amaç fonksiyonuna göre kopyalanır, gelecek kuşakta daha iyi oğul oluşturabilecek bireyler seçilir. Üreme operatörü, yapay bir seleksiyondur.

Burada bir kromozom havuzundaki kromozomlardan hangilerinin yeni yaratılacak havuza aktarılacağının, kromozomların başarı değerlerine bağlı olarak belirlenmesi esastır.

Yeni popülasyondaki ebeveynleri oluşturmak üzere bazı bireylerin seçilmesi gerekir. Kurama göre iyi olan bireyler yaşamını sürdürmeli ve bu bireylerden yeni bireyler oluşmalıdır. Bu seçim çeşitli ölçütlere göre yapılabilir. Bu yöntemlerden bazıları şunlardır:

- Rulet Seçimi
- Boltzman Seçimi
- Turnuva Seçimi
- Sıralı Seçim

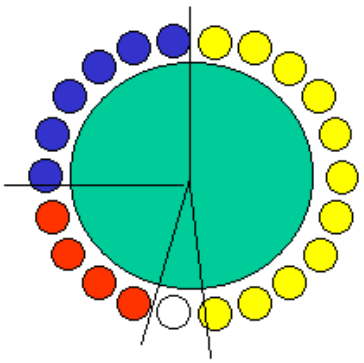
Bunların yanısıra problem içeriği ve yapısına göre daha karmaşık, algoritmanın işleyişi sırasında seçim parametrelerini değiştiren seçim yöntemleri de vardır. Temel olarak bu yöntemler benzetim-tavlama yöntemi gibi çalışır. Elitizm denilen seçicilik yöntemi de her kuşaktaki belli sayıdaki en iyi sonucun diğer popülasyonlarda varlığını sürdürmesi demek olup isteğe göre kullanılabilir. Örnek olarak rulet tekerleği seçimi aşağıda açıklanmıştır:

1. Tüm bireylerin uygunluk değerleri bir tabloya yazılır.
2. Bu değerler toplanır.
3. Tüm bireylerin uygunluk değerleri toplama bölünerek [0,1] aralığında sayılar elde edilir. Bu sayılar bireylerin seçilme olasılıklarıdır. Sayıların hepsi bir tabloda tutulur.

4. Seçilme olasılıklarını tuttuğumuz tablodaki sayılar birbirine eklenerek rasgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde, son eklenen sayının ait olduğu çözüm seçilmiş olur.

Rulet seçiminde kromozomlar uygunluk fonksiyonuna göre bir rulet çevresine gruplanır. Uygunluk fonksiyonu, herhangi bir ölçüte uyan bireylerin seçilmesi için kullanılır. Bu rulet üzerinden rasgele bir birey seçilir.

Bu yönteme rulet tekerleği seçimi adı; bir daireyi, çözümlerin uygunluklarına göre dilimleyip çevirdiğimizde olacakların benzeşimi olduğu için verilmiştir (Bkz. Şekil 3.2.).



Şekil 3.2 Rulet tekerleği seçimi

Görüldüğü üzere daha büyük alana sahip bireyin seçilme şansı daha fazla olacaktır.

Rulet tekerleği seçimi, eğer uygunluk fonksiyonu değerleri çok fazla değişiyorsa sorun çıkartabilir. Örneğin en iyi kromozomun uygunluk değeri % 90 ise, diğer kromozomların seçilme şansı azalacaktır.

Bunu önlemek için sıralı seçim kullanılabilir. Sıralı seçimde en kötü uygunluk değerine sahip olan kromozoma 1 değeri, sonrakine 2 değeri verilir ve böylelikle seçilmede bunlara öncelik tanınmış olur. Bu şekilde onların da seçilme şansı artar ama bu, çözümün daha geç yakınsamasına neden olabilir.

Üreme yönteminde genellikle rulet çemberi yöntemi kullanılır. Varolan popülasyondaki en uygun kromozomlar rulet çemberi üzerine yerleştirilir, çember popülasyondaki dizi sayısı kadar döndürülerek yeni kuşak elde edilir.

Bir diğer seçim yöntemi olan turnuva seçimi ise hızlı olması nedeni ile tercih edilmektedir. Turnuva seçiminde eşleme yapılacak bireyler seçilirken popülasyonun belli bir oranı kadar birey seçilir ve bunların arasından uygunluk fonksiyonu en iyi

olan eşlemeye alınır. Örneğin 100 bireyden oluşan bir popülasyonda rasgele 10 birey seçilir ve bu 10 bireyden uygunluk fonksiyonu en iyi olan alınıp eşlemeye sokulur.

3.4.3. Çaprazlama (Crossover) operatörü

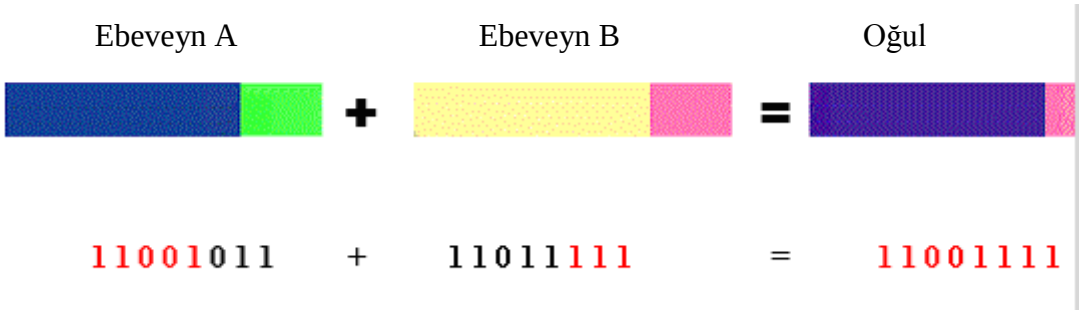
GA'nın performansını etkileyen parametrelerden biri de çaprazlama işlemidir. Doğal popülasyonlarda çaprazlama, organizmalar arasında yapılan döllenme (hayvanlarda çiftleşme, bitkilerde tozlaşma) çalışmaları olarak bilinir.

Çaprazlama olarak adlandırılan bu operatör iki çözümün yapı taşları kullanılarak yeni bir çözüm oluşturulması temeline dayanır. Çaprazlama işlemi genel olarak ikili dizilerin parçalarının değiş tokuşu şeklinde gerçekleştirilir. Çaprazlama, ebeveynlerden bazı genleri alarak yeni bireyler oluşturma işlemidir. Farklı uygulamalarda, farklı kodlama yöntemlerinden yararlanıldığı için farklı çaprazlama yöntemleri kullanılır.

Tek Noktalı Çaprazlama: Rulet çemberi yöntemi ile yapay seçim sonucunda elde edilen yeni popülasyon dizisinde rastsal olarak iki kromozom seçilir ve karşılıklı çaprazlama işlemine tabi tutulur. Çaprazlama işleminde, L dizi uzunluğu olmak üzere,

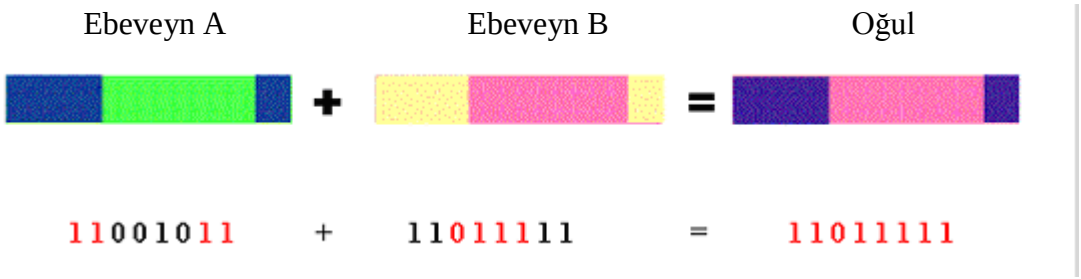
$\{1 \le k \le L-1\}$ aralığında düzgün (uniform) olarak k tamsayısı seçilir. Bu tamsayı değerine göre dizi, çaprazlamaya tabi tutulur.

En basit çaprazlama yöntemi, Şekil 3.3.'de verildiği gibi bir noktalı çaprazlama yöntemidir. Bir noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir.



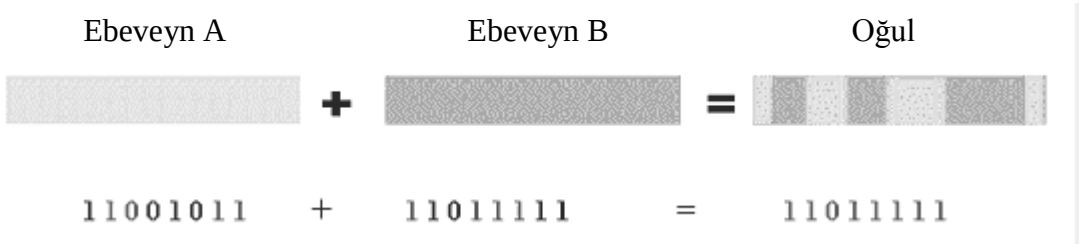
Şekil 3.3 Tek noktalı çaprazlama

İki Noktalı Çaprazlama: İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir (Bkz. Şekil 3.4).



Şekil 3.4 İki noktalı çaprazlama

Düzgün (Uniform) Çaprazlama: Kromozom üzerindeki bit değerleri rassal olarak belirli noktalardan bir diğer ebeveyn üzerine kopyalanır (Bkz. Şekil 3.5).



Şekil 3.5 Düzgün (Uniform) çaprazlama.

3.4.4. Mutasyon İşlevi

Genetik algoritmalarda önemli rol oynayan proseslerden biri de mutasyon operatörüdür. Canlılarda gen rekombinasyonlarının dışındaki diğer nedenlerle ve ani olarak oluşan kalıtsal değişimlere *mutasyon* denir. Doğal popülasyonlarda mutasyon işlemi ile aşağıdaki değişimler gerçekleşir.

1. Kromozom yapısındaki değişmeler
2. Kromozom sayısındaki değişmeler
3. Gen yapısındaki fiziksel ve kimyasal değişmeler

Mutasyon, çaprazlama operatörü gerçekleştikten sonra yapılır. Mutasyon, oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek ve sonuca daha hızlı ulaşmak amacıyla yapılır. Mutasyon, oluşan yeni bireyin bir bitini (eğer ikili düzende ifade edilmiş ise) rasgele değiştirir.

Yapay sistemlerde mutasyon işlemi sırasında kromozomdaki gen sayısı değişmez sabit kalır. Doğal popülasyonlarda mutasyon oranı oldukça düşüktür. Mutasyon frekansının büyüklüğü genetik algoritmanın performansını etkilemektedir.

Mutasyon işlemi bir tek kromozom üzerinde yapılır. Mutasyon frekansına göre, mutasyona uğratılacak sayıdaki diziler, popülasyondan rassal olarak seçilir ve belirlenen mutasyon yöntemine göre değişime uğratılır. Bu değişim işlemi çok değişik şekillerde yapılabilir. Problemin yapısı bu aşamada çok önemlidir. Örneğin sıralama problemlerinde sıralamayı değiştirmek, ikilik dizi gösteriminde bitleri değiştirmek, çözüm ağaçlarında parçaları değiştirmek, işlem olarak tanımlanır [23].

Mutasyonun en önemli faydası çözümün yerel minimumlara takılmasını engeller. Eğer mutasyon oranı çok büyük seçilir ise bu defa da çözüm evrimleşmez, rasgele değerler alır, bu da genetik algoritmaların mantığına terstir.

Aşağıda bit dizisi bir kromozom için mutasyon işlemi gösterilmiştir.

1 **1** 0 0 1 0 0 1 => 1 **0** 0 0 1 0 0 1

3.5. Genetik Algoritmaların İşlem Basamakları

GA'nın işleyişi çok basittir ama bu kadar basit olan yöntemin, en zor problemleri nasıl çözdüğünün anlaşılması da o kadar zordur. Bu da GA'nın en karmaşık ve bilim adamlarının yıllardır çözmeye çalıştıkları en önemli sorularından biridir [22].

GA'nın bu yönünü biraz açıklayalım; GA, çözüm(ler) bulmak için taranması gereken parametre uzayının çok büyük olduğu durumlarda bu arama işlemi için en akılcı yöntemdir. Evrimin her sürecinde edinilen bilgi, sonraki kuşaklara aktarılarak taramanın daha uygun bölgelerde gezmesi sağlandığı gibi değişim işlemi yardımıyla yerel çözüm noktalarına sıkışıp kalma olasılığı da azaltılmaktadır. Ayrıca GA'nın paralel işlem yapılan bilgisayarlarda kullanılmaya elverişli yapısı da, zaman alıcı problemlerin çözümü için çekici bir seçenek olmasını sağlamaktadır.

Genetik algoritmaların genel haliyle takma kodu (pseudo code) şu şekildedir;

```
1 Başlangıç popülasyonunu rasgele oluştur
2 Kuşak sayısı=0
3 (Kuşak sayısı < istenilen kuşak sayısı) olduğu sürece tekrarla
4    Yeni boş popülasyon oluştur
5    En iyi belirli orandaki bireyleri yeni popülasyona kopyala
(elitizm)
6    Yeni Popülasyon doluncaya kadar tekrarla
7       Seçme operatörü ile iki birey al
8       Çaprazlama uygula
9       Mutasyon uygula
10      Uygunluk değerini bul
11      Yeni popülasyona ekle
12   Eski popülasyonu yok et
13   Kuşak sayısı=Kuşak sayısı + 1
14 En iyi bireyi bul ve yaz (çözüm)
```

İşlemleri adım adım açıklarsak;

1. Bu adıma toplumda bulunacak birey sayısını belirleyerek başlanmaktadır. Kullanılacak sayı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Başlangıç popülasyonu, sonuçlardaki düzgün (uniform) yapının korunması için genellikle rasgele oluşturulur.

2. Kuşak sayısı değeri sıfırlanır, genetik algoritmalar genellikle büyük çözüm uzayları olan problemlere uygulandığından sonuç için amaçlanan belirli bir değer yoktur. Bunun yerine yapılan yinelemeler belirli bir değere yakınsadığında sonuç bulunmuş sayılır. Bu yinelemelerin sayısını belirlemek için kuşak sayısı değişkeni kullanılır.

3. Kuşak sayısı değeri istenilen kuşak sayısına ulaşılıncaya kadar evrimsel işlemlerin yapıldığı yinelemeler devam eder.

4. Her yinelemede yeni kuşaklar evrimleşip yeni popülasyon oluşturulacağından evrim sonucu aktarılacak popülasyon için yer ayrılır.

5. Yapılan elitizm oranına göre eski popülasyondaki uygunluk değeri en iyi bireylerin bir kısmı yeni popülasyona evrimleşmeden doğrudan atılır. Bu sayede ataların bilgileri oğullara geçirilmiş olur ve en iyi sonuç sonraki kuşaklarda kaybedilmez.

6. Yeni popülasyon doluncaya kadar türetme işlemleri devam ettirilir. Genellikle popülasyon sayısı sabit tutulur.

7. Kullanılan seçim işlevine göre iki birey seçilir. Seçim işlevi doğadaki ortama en çok uyum sağlayan hayatta kalır kuralına göre yapılır. Seçim operatörlerinin hepsinde uygunluk fonksiyonu değeri iyi olanların seçilme olasılığı daha fazladır.
8. Seçilen bireylere çaprazlama oranına göre çaprazlama uygulanır. Çözüm uzayında homojen bir tarama yapabilmek için çaprazlama oranı yüksek tutulur.
9. Çaprazlama sonunda elde edilen bireyler mutasyona oranına bağlı olarak mutasyona tabii tutulur. Mutasyon çözümlerin yerel minimum değerlerinden çıkmasını sağlar. Mutasyon oranı çok yüksek seçilir ise çözümler evrimi takip etmek yerine çözüm uzayında rasgele dolaşmaya başlar.
10. Elde edilen bireylerin uygunluk değeri bulunarak bireyin bilgileri arasına yazılır
11. Oluşturulan yeni oğul bireyler yeni popülasyona eklenirler. Yeni popülasyon dolmaz ise 7. adıma geri dönülür.
12. Yeni popülasyon tamamen dolduğunda işi biten eski popülasyon yok edilerek hafıza boşaltılır.
13. Kuşak sayısı bir arttırılarak 3. basamağa dönülür. Eğer istenilen kuşak sayısına ulaşılmadıysa yineleme 3-13 arasındaki adımlarda tekrar eder. İstenilen kuşak sayısına ulaşılır ise 14. adımdan devam edilir.
14. İstenilen kuşak sayısına ulaşıldığından mevcut popülasyondaki en iyi birey bulunan en iyi çözümdür, sonuç bulunmuştur.

3.6. Sonuç

Genetik algoritmalar problem çözümü yapmak için doğadaki evrimin bilgisayar ortamında benzetimidir. Genel olarak Evrimsel hesaplamanın bir parçası olan GA yapılan sayısız çalışma ile birçok alanda uygulama görmüş ve başarısını kanıtlamış bir problem çözme tekniğidir.

GA sezgisel ve rastsal işleyişi ile analitik çözümü olmayan veya analitik çözümü çok zor yapılabilen problemlerin çözümünde daha sık kullanılmaktadır. Rastsal yapısı ile çözüm uzayını tarayıp, popülasyondaki bireylerin evrimleşmesiyle kaliteli çözüme yönelmektedir.

GA nın çalışması sırasında sonuçların kalitesini doğrudan etkileyen operatörler vardır, bunlar; Mutasyon, çaprazlama ve seçim operatörüdür. Bunların yanında işlem hızını doğrudan etkileyen penaltı ve düzeltme operatörleri de GA nın önemli değişkenleridir. Bütün bu operatörler için yaklaşık bir oran verilse de uygulanan probleme özel olarak bu operatörler için uygun oranlar ve yöntemler denenerek bulunmalıdır.

4. BİRİM KATKI SORUNUNA GA UYGULANMASI ÜZERİNE YAPILAN ÇALIŞMALAR

4.1. Giriş

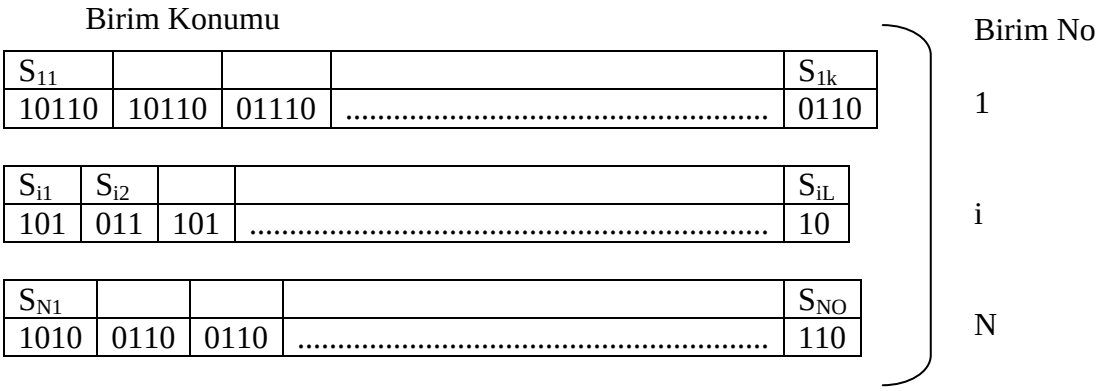
Generatör birim katkı sorunu analitik olarak çözülemeyen, oldukça karmaşık bir eniyileme sorunudur. Bilgisayarların işlem gücünün artması ile oluşturulan algoritmalar sayesinde çözüme daha hızlı biçimde ulaşılmaktadır. Çözüm uzayının genişliği nedeni ile sezgisel (heuristic) çözüm yapan algoritmalar daha başarılı olmaktadır.

Bu bölümde generatör birim katkı sorununa genetik algoritmalar uygulanarak çözüm yapılmış çalışmalardan bazıları incelenmiştir. Çalışmalar arasındaki farklılıklar genellikle kromozom kodlaması ile olmaktadır. Bunun yanında kullanılan genetik operatörlerde de önerilen kromozom kodlamasına göre farklılıklar olmaktadır.

4.2. Mevcut çalışmalar

Rudolf ve Bayrleithner birim katkı sorunu için çalışmalarında iki aşamalı bir çözüm önermektedirler. Önerilen yöntemde birimlerin katkısına karar verilirken genetik algoritmalar kullanılmakta, ekonomik yük dağılımı için ise Lagrange rahatlatması kullanılmaktadır [19].

Çalışmanın en özgün yanı birim katkı sorununun en zor iki kısıtını olan en düşük devrede kalma ve en düşük devre dışı kalma sürelerini kromozomların kodlamasına dahil etmesidir (Bkz. Şekil 4.1). Bu yönteme göre kodlamada birimlerin açık kapalı olduğunu bir bit ile gösterdikten sonra sonraki 3 bit ile generatörlerin ne kadar saat daha açık veya kapalı olması gerektiği gösterilmektedir. Örneğin bir generatör için belirli bir saatte “1011” kodlaması yapıldı ise ilk bit generatörün devrede olduğunu göstermekte, sonraki 3 bit ise generatörün en düşük devrede kalma süresinin ihlal edilmemesi için 3 saat daha devrede kalması gerektiği belirtilmektedir.



Şekil 4.1 Kaynak [19] tarafından önerilen kromozom kodlaması

Generatörün çalışma/durma süresini gösteren bitlerin hepsi 0 olduğunda ilgili generatör bir sonraki saatte konum değiştirebilir. Doğal olarak yöntemdeki çalışma/durma süresini gösteren bölümün kaç bit olacağı sistemdeki generatörün devrede/devre dışı kalma süresi belirleyecektir. Devrede/devre dışı kalma sürelerinden en büyük olanı ikilik tabanda gösterecek kadar bit bu işlem için ayrılmalıdır. Üç bitlik bir bölüm ile en fazla 8 saatlik devrede/devre dışı kalma konumu gösterilebilir.

Rudolf ve Bayrleithner'ın çalışmalarında önerdikleri kodlama çaprazlama ve mutasyon işlemlerinde gösterimin bozulmaması için özel operatörler gerektirmektedir. Çalışmada oluşan geçersiz çözümleri devre dışı bırakmak amacı ile penaltı işlevleri kullanılmıştır.

Önerdikleri çözüm yöntemini bir saatlik periyodlar ile 24 saatlik bir süre için test etmişlerdir. Sonuçta çözümler uygulamanın kolay olduğunu ve en uygun sonuçlara ulaşılabileceğini göstermiştir.

Rudolf ve Bayrleithner'ın önerdikleri kodlama yöntemini kullanan bir diğer çalışma Yiying ve arkadaşları tarafından yapılmıştır [25]. Bu yöntemde de devrede/devre dışı kalma süresi kısıtları kodlama ile çözüldükten sonra diğer kısıtlar için penaltı operatörleri kullanılmıştır.

Yiying ve arkadaşları çalışmalarını 10 ve 110 generatörlü iki farklı sistem için test etmişlerdir. Testlerinde değişik seçim işlevleri kullanarak süre ve sonuçların kalitesini göstermişlerdir (Bkz. Tablo 4.1).

Tablo 4.1 Değişik seçim işlevleri için çözüm süreleri ve sonuçların kalitesi.

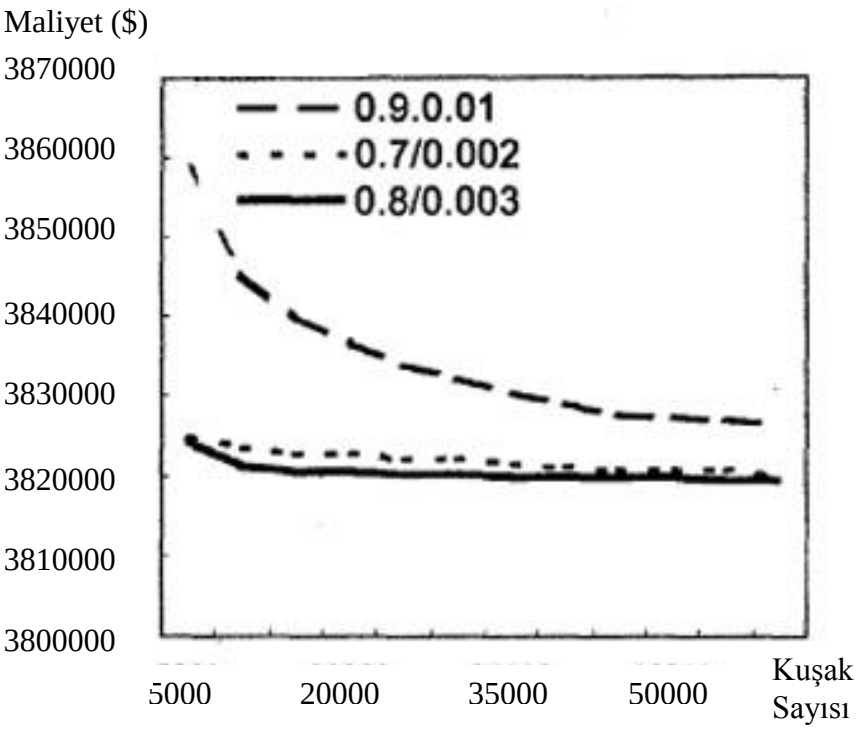
Seçim Yöntemleri	Rank	Rulet Tekeri	Turnuva	Deterministik örnekleme	Stokastik örnekleme	Stokastik uniform örnekleme
Süre	9”	14”	12”	31”	16”	22”
Yakınsayan Kuşak Sayısı	1000	1600	1100	3500	1700	2200
Yakınsadığı Değer	79023	79024	78971	79016	79038	78974

Seçim işlevi yanında değişik çaprazlama ve mutasyon oranları ile de test edilen çözümler Tablo 4.2 de gösterilmiştir [25].

Tablo 4.2 Değişik çaprazlama-mutasyon değerleri için çözüm süreleri ve sonuçların kalitesi.

Çaprazlama/ Mutasyon	0.85/0.005	0.85/0.015	0.95/0.005	0.95/0.015	0.9/0.01
Süre	9”	14”	12”	31”	16”
Yakınsayan Kuşak Sayısı	1000	1600	1100	3500	1700
Yakınsadığı Değer	79023	79024	78971	79016	79038

Yiying ve arkadaşlarının elde ettikleri sonuçlara göre değişik seçim işlevleri çözümde önemli değişikliklere neden olmaz iken çaprazlama ve mutasyon oranının değişimi çözüm kalitesini ve süresini daha çok etkilemektedir (Bkz. Şekil 4.2) [8]. GA da çaprazlama oranı genellikle 0,8 ve üzeri alındığında çözüm uzayı daha iyi taranmaktadır. Mutasyon oranı olarak 0,01 seçildiğinde ise sonuçların çözüme yakınsaması gecikmektedir, bu da 0,01 mutasyon oranının bu sistem için büyük olduğunu göstermektedir.



Şekil 4.2 Değişik çaprazlama-mutasyon değerleri için çözümün yakınsaması.

Swarup ve Yamashiro birbirinin aynı iyi yayını ile genetik algoritmada kromozom kodlamasında değişiklik öneren başka bir çalışma yapmışlardır [17, 18]. Önerilen kodlamada birimler açık/kapalı durumları bir bit ile gösterilmekte fakat vektör şeklindeki kromozom çok uzun olduğundan yerine iki boyutlu matris biçiminde kromozom önerilmektedir (Bkz. Şekil 4.3).

Saat / Birim	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Şekil 4.3 Swarup ve Yamashiro tarafından önerilen iki boyutlu kromozom kodlaması.

Önerdiği kromozom kodlaması oldukça yenilikçi olmasına rağmen kaynaklar [17, 18] de, mutasyon, çaprazlama gibi genetik işlevlerin nasıl yapıldığı belirtilmemiştir. Önerilen kodlama yapısı alışılagelmiş bit dizisinden oldukça farklı olduğundan kendine özel çaprazlama işlevi gerektirmektedir, çalışmada bundan bahsedilmemesi büyük bir eksikliktir. Bunun yanında 2003 yılında yayımlanan çalışmada (kaynak [18]) test için seçildiği söylenen günlük yük eğrisi 2002 yılındaki çalışmadakinden (kaynak [17]) tamamen farklı iken generatörlere ait yükler aynı çıkmıştır!

Juste ve arkadaşlarının 1999 yılında yayımladıkları çalışma [26] diğerlerinden kromozom kodlamasındaki özgünlük ile ayrılmaktadır. Bu çalışmada kullanılan yöntem gerçekte genetik algoritma değil evrimsel programlamadır, iki çözüm yöntemi arasındaki tek fark ise evrimsel programlamada kromozom kodlaması için bit dizisi yerine tamsayı veya ondalıklı sayı kullanılmasıdır [26].

Çalışma, önerilen kodlama kısıtların içinde barındırmaktadır. Kodlama yenilikçi olması yanında mutasyon ve çaprazlama işlevlerinin en az devrede/devre dışı kalma kısıtları için uygun olmayan çözümler üretmesini imkansız hale getirdiğinden oldukça başarılıdır. Kullanılan kodlamanın olumsuz yanı ise sistemde en az devrede kalma süresi (MUT) ve en az boşta kalma süresi (MDT) çok düşük olan generatörler bulunduğunda işlem süresinin diğer yöntemlere göre oldukça yavaşlayacak olmasıdır.

Kodlama mantığı şu şekildedir [26];

- Birim katkı sorununun uygulanacağı zaman aralığı (24 saat) p saatlik birimlere bölünerek gösterilir. Örneğin p=4 olduğu durum aşağıda gösterilmiştir.

Birim i	0000	0001	1111	1111	1110	0000
Birim i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}

- p değeri belirlenirken sistemdeki generatörlerin en düşük devrede kalma süresi veya en düşük devre dışı kalma süresi değerlerinden en az olanından küçük veya eşit olan ve problemin çalıştırılacağı zaman aralığının bölen olan sayı alınır. Örneğin sistemdeki generatörlerden MUT değeri en az olan 7, MDT değeri en az olan 5 ise p=4 alınır.

$$p \leq \min(MUT,MDT) \text{ ve}$$

$$p \in \{1,2,3,4,6,8,12,24\}$$ (24 saatlik problem için)

- k, p'nin bit sayısı olmak üzere, p sayısı -k≤p≤k arasında değerler alabilir, p'nin aldığı değerlerin açılımı şu şekildedir;

1111	4
1110	3
1100	2
1000	1
0000	0
0001	-1
0011	-2
0111	-3

Burada 1 generatörün devrede olduğunu, 0 generatörün devre dışı olduğunu gösterir.

Yukarıda anlatılan kodlama sayesinde sisteme uygulanan mutasyon ve çaprazlama işevlerinden sonra geçersiz bireylerin kontrol edilmesi, gerekirse düzeltilmesi kolaylaşmaktadır. Kodlamanın kötü yanı ise bütün işlevler için kodlamaya özel fonksiyonlar yazılmasını gerektirmekte ve işlem süresini uzatmaktadır.

Birim katkı sorununun çözümüne paralel GA uygulayan yöntemlerde bulunmaktadır [27, 28]. GA genellikle iki şekilde paralelleştirilir, birincisi her bir paralel makinada veya işlemcide birbirinden bağımsız popülasyonlar oluşturmak ve elitisizm veya seçim işlevi gibi bütün popülasyon hakkında bilgiye ihtiyaç duyan işlevleri ana merkezde yapmaktır. İkinci yöntem ise popülasyonunu tek merkezde tutarak, bireyler için yapılan mutasyon, çaprazlama, evrimleştirme ve uygunluk fonksiyonunun hesaplanması işlemlerini köle bilgisayarlara dağıtmaktır.

Yang ve arkadaşları kullandıkları 8 işlemcili paralel bilgisayarda iki yöntemle paralelleştirme yapmışlardır, birincisi efendi-köle paralel GA yöntemidir. Bu yöntemde tek bir efendi işlemci k adet köle işlemciyi koordine etmektedir. İşlemin basamakları şu şekildedir [28];

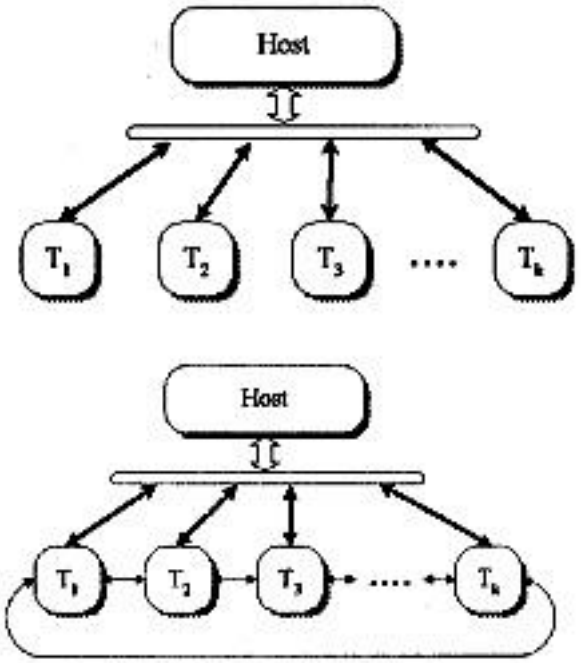
- Efendi işlemci bütün popülasyonu rasgele üretir
- Popülasyondan belirli sayıda bireyi uygunluk fonksiyonlarını dikkate alarak rasgele seçer ve alt-popülasyon olarak kölelere gönderir. Burada rulet tekeri seçimi kullanmaktadır, bu sayede uygunluk fonksiyonu değeri daha iyi olan bireylerin seçilmesi ihtimali artmaktadır.

- Alt-popülasyonu alan köleler bu popülasyon için çaprazlama, mutasyon ve uygunluk fonksiyonu değerini hesaplar ve popülasyonu ana işlemciye gönderirler.
- Ana işlemci durma koşulu sağlananıncaya kadar ikinci adıma dönerek işlemleri tekrarlar.

Yang ve arkadaşları kullandıkları ikinci paralelleştirme yöntemine ise çift yönlü halka paralelleştirmesi adını vermektedirler (Bkz. Şekil 4.4). Kullandıkları bu yöntemde her işlemci alt-popülasyonu kendisi evrimleştirmekte fakat belirli zamanlarda popülasyonundaki en iyi bireyleri sağındaki ve solundaki işlemcilere göndermektedir, bu işlemciler ise aldıkları en iyi bireyleri kendi popülasyonlarındaki en kötü bireylerle değiştirmektedirler. Kullanılan yöntemin basamakları şunlardır;

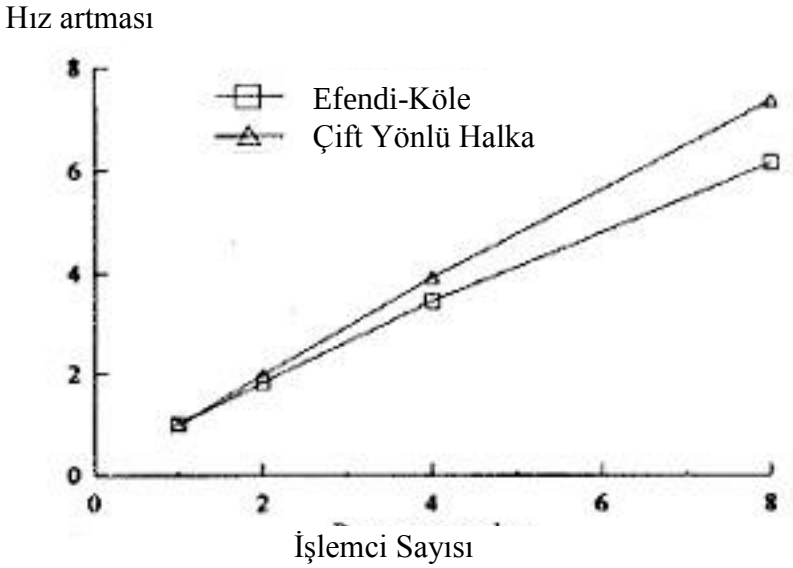
- Her işlemci kendi popülasyonunu rasgele üretir ve bunlar alt-popülasyon olur.
- Her işlemci GA yinelemesini kendi alt-popülasyonu ile üretir.
- Her işlemci popülasyonlarındaki en iyi bireyleri sağındaki ve solundaki işlemcilere gönderir ve kendisine gelen en iyi bireyleri alt-popülasyonlarındaki en kötü bireylerin yerine koyar.
- Durdurma şartı sağlanmadıysa ikinci adımdan işlemlere devam edilir.

Yang ve arkadaşları önerdikleri paralelleştirme yöntemlerini 48 birimlik bir sistem için test etmişler ve işlem süresi ile sonuçları "simulated annealing", Lagrange rahatlatması ve tek işlemcili GA sonuçlarıyla karşılaştırmışlardır. Elde edilen veriler sonuç kalitesinde önemli bir değişiklik olmaz iken işlem süresinin oldukça azaldığını göstermektedir (Bkz. Tablo 4.3) [28].



Şekil 4.4 Yang ve arkadaşlarının kullandıkları paralel işlemci yapıları. Efendi köle ve çift yönlü halka.

Kaynak [28] tarafından önerilen paralelleştirmede, paralel işlemci sayısı arttıkça işlem süresi beklendiği şekilde azalmaktadır, işlemci sayısına göre işlem süresindeki artışın grafiği şekil 4.5 de verilmiştir [28].



Şekil 4.5 Yang ve arkadaşları tarafından önerilen paralel yöntemde işlemci sayısına göre işlem hızı değişimi.

Tablo 4.3. Yang ve arkadaşlarının paralelleştirme yöntemlerinin diğer yöntemlerle karşılaştırılması.

Yöntem	İşlemci Sayısı	Çalışma Süresi(sn)	Hız Katlanması	İşlemci Başına Verim (%)	Toplam Maliyet
SA	1	3924	1	100	195,296
LR	1	25,2	1	100	195,801
Sıralı GA	1	3377	1	100	195,218
Efendi-köle Paralel GA	2	1830	1,845	92,27	195,219
	4	974	3,467	86,68	195,221
	8	549	6,151	76,89	195,218
Çift yönlü halka paralel GA	2	1696	1,991	99,56	195,222
	4	860	3,927	98,17	195,218
	8	460	7,341	91,76	195,219

4.3. Sonuç

Birim katkı sorununu GA yöntemini uygulayan çalışmaların algoritma akışları hemen hemen aynıdır. GA'nın özelliğinden dolayı bu beklenen bir sonuçtur. Algoritmalardaki fark ekonomik yük dağılımının yapıldığı yerdir. Bazı uygulamalarda ekonomik yük dağılımı her yinelemede birkaç kere yapılırken [17, 18], çoğu uygulamada yoğun işlem zamanı aldığından tek yerde yapılmaktadır.

Mevcut çalışmalar arasındaki en önemli fark kromozomların kodlanmasıdır. Birim katkı sorunu kısıtları nedeniyle karmaşık bir problem olduğundan önerilen kodlamalar genellikle kısıtları da kapsayacak şekilde olmakta, bu sayede mutasyon ve çaprazlama operatörlerinden sonra uygun olmayan çözümlerin ortaya çıkma ihtimali azaltılmakta veya bu çözümlerin kontrolü ile düzeltilmesi kolaylaştırılmaktadır.

Kullanılan mutasyon ve çaprazlama işlevleri kodlamaya özel olmalıdır ve bu şekilde de yapılmaktadır. Mutasyon ve çaprazlama oranları hakkında herhangi bir tartışma yoktur. Çaprazlama oranları için 0,8-1,0 arası değerler kabul görürken, mutasyon oranları için 1/L-2/L oranı (L=kromozom bit genişliği) kabul görmektedir.

Genetik Algoritmalar, birim katkı sorununa uygulandığında diğer sezgisel yöntemlerden daha hızlı ve kaliteli sonuçlar vermektedir [28]. Lagrange rahatlatması

sezgisel yöntemlere karşı çok daha hızlı sonuç vermektedir, diğer taraftan Lagrange rahatlatması her zaman sonuç sağlamamakta ve sağladığı sonuçların kalitesi sezgisel yöntemlere göre daha kötü olmaktadır [28].

5. BİRİM KATKI SORUNUNA GA UYGULANMASI

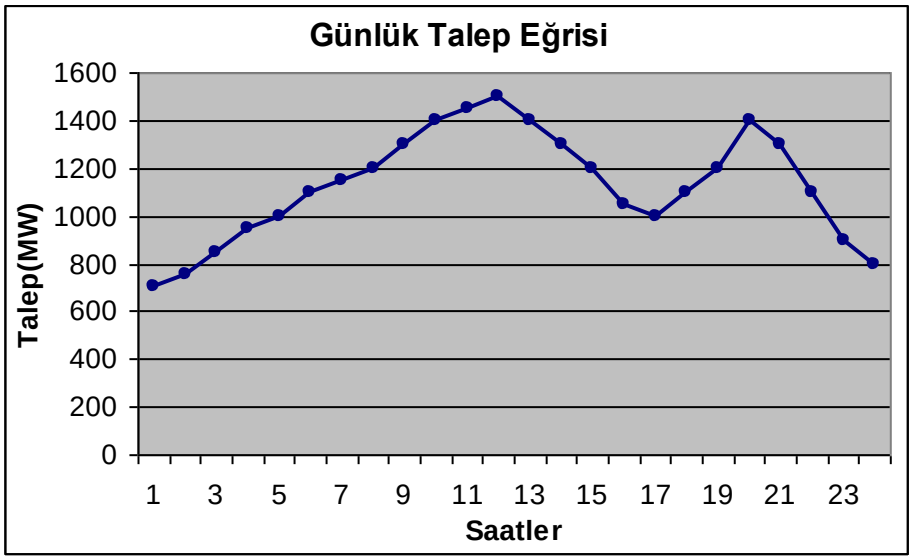
5.1. Giriş

İlk dört bölümde birim katkı sorunu, genetik algoritmalar ve birim katkı sorununa GA uygulayan mevcut çalışmalara değinilmiştir. Bu bölümde ise çalışmanın amacı olan örnek uygulama hakkında bilgi verilecektir.

Öncelikle birim katkı sorunu uygulamasının yapılacağı örnek generatör ve yük bilgileri verilecek, daha sonra kullanılan matematiksel model açıklanacaktır. İkinci aşama olarak uygulamada kullanılan genetik işlevler ve yöntemler detaylı olarak ele alınacaktır. Programın takma kodu verildikten sonra elde edilen sonuçlar listelenecektir.

5.2. Birim Katkı Modeli

Genetik algoritma ile çözüm uygulanacak örnek sistem 10 generatörlü olarak alınmıştır. Test sisteminde generatörlere ait karakteristik katsayıları ile günlük yük eğrisi tabloları Ek A'da verilmiştir [18]. Günlük yük eğrisinin grafiği şekil 5.1 de verilmiştir.



Şekil 5.1 Günlük yük eğrisi grafiği.

Birim katkı sorunu (BKS) kısıtları nedeni ile çözümü zor karmaşık bir eniyileme problemidir. Kısıtlar aynı zamanda uygulanan algoritmaların yavaş çalışmasına da neden olmaktadır. BKS'nun GA ile çözümünün uygulanmasında dikkate alınan kısıtlar ve matematiksel model aşağıdaki gibidir.

1. Yük Dengesi: Generatörlerin her saat için ürettiği toplam güç ilgili saat için talep edilen güce (yük) eşit veya daha fazla olmalıdır. Burada P_i, i. generatörün gücü, P_D ise toplam talep gücüdür (yük+kayıp).

$$\sum_{i=1}^{n} P_{i,j} - P_{D,j} \geq 0 \tag{5.1}$$

2. Yedek Güç (P_R): Her saat için sistemden talep edilen gücün %10 'u oranında yedek güç (spinning reserve) bulunmalıdır.

$$\sum_{i=1}^{n} P_{i,j}^{\max} \geq P_{D,j} + P_{R,j} \tag{5.2}$$

3. En az Devrede Kalma Süresi (MUT): Generatörler karakteristiklerine göre devreye alındıktan sonra devre dışı kalmadan önce belirli bir süre çalıştırılmalıdır. Burada T^{ON} ilgili generatörün devrede olduğu saatleri göstermektedir.

$$\sum_{j=1}^{T} \left| T_{i,j}^{ON} - MUT_i \right| \geq 0 \tag{5.3}$$

4. En az Boşta Kalma Süresi (MDT): Generatörler karakteristiklerine göre devre dışı kaldıktan sonra devreye alınmadan önce belirli bir süre boşta kalmalıdır. Burada T^{OFF} ilgili generatörün boşta olduğu saatleri göstermektedir.

$$\sum_{j=1}^{T} \left| T_{i,j}^{OFF} - MDT_i \right| \geq 0 \tag{5.4}$$

5. Generatör Limitleri: Her bir saat için generatörlerden çekilen güç ilgili generatörün sağlayabileceği en düşük ve en yüksek güç değerleri arasında olmalıdır.

$$P_{\min i,j} \leq P_{i,j} \leq P_{\max i,j} \tag{5.5}$$

Birim katkı sorununda amaç generatörlerin günlük toplam maliyeti en düşük seviyeye getirecek generatör çalışma planının bulunmasıdır. Sistem kısıtlarının

yanında maliyet hesabının yapılması için yakıt maliyeti ile başlatma ve durdurma maliyeti de göz önüne alınmalıdır.

Birim katkı sorununda maliyeti oluşturan fonksiyonlar ve matematiksel gösterimleri aşağıda verilmiştir.

1. Yakıt Maliyeti: Generatörler devrede iken generatörden çekilen güce bağlı bir fonksiyondur.

$$\sum_{j=1}^{T}\sum_{i=1}^{n}(a_i + b_i P_{i,j} + c_i P_{i,j}^2) \tag{5.6}$$

2. Başlangıç Maliyeti: Generatörlerin devre dışı durumdan devreye alınırken oluşan maliyettir.

$$\sum_{j=1}^{T}\sum_{i=1}^{n}\sigma_i + \delta_i\left[1 - e^{(-T_{i,j}^{OFF}/\tau_i)}\right] \tag{5.7}$$

3. Durdurma Maliyeti: Generatörlerin devrede iken devre dışına alınmasıyla oluşan maliyettir. Uygulama yapılan sistemde generatörlerin durdurma maliyeti çok düşük olduğundan ve kullanılan algoritmanın karmaşıklığını arttırmamak için durdurma maliyeti sıfır alınmıştır.

$$\text{SDC}=0 \tag{5.8}$$

Buna göre (5.5)-(5.7) eşitliklerinden toplam maliyet (TM);

TM = Yakıt Maliyeti + Başlangıç Maliyeti + Durdurma Maliyeti

$$\text{TM} = \sum_{j=1}^{T}\sum_{i=1}^{n}(a_i + b_i P_{i,j} + c_i P_{i,j}^2) \quad + \quad \sum_{j=1}^{T}\sum_{i=1}^{n}\sigma_i + \delta_i\left[1 - e^{(-T_{i,j}^{OFF}/\tau_i)}\right] \tag{5.9}$$

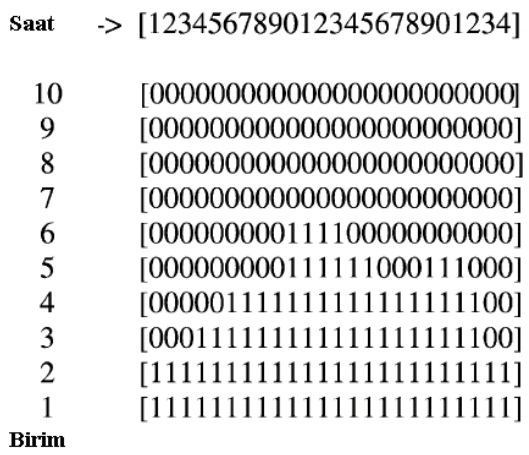
şeklinde bulunur. Amaç, uygulamanın yapılacağı 24 saatlik sürede toplam maliyeti en düşük yapmaktır.

5.3. Genetik Algoritmanın Aşamaları

5.3.1. Kromozom Kodlaması

Bundan önceki çalışmalarda kromozomların kodlanması için çeşitli yöntemler önerilmiştir. Basitliği ve genetik işlevlerde daha hızlı işlem yapmaya olanak sağlaması nedeni ile bit dizisi kodlaması seçilmiştir. Gösterim kolaylığı ve

anlaşılmasının kolaylaştırılması için kodlama iki boyutlu düşünülürse şekil 5.2 deki gibi olacaktır.



Şekil 5.2 Gerçeklenen uygulamada kromozomların iki boyutlu gösterimi

Şekil 5.2 deki gösterim doğrudan programın kodlanmasında kullanılırsa mutasyon ve çaprazlama işlemleri daha karmaşık olmakta ve işlem süresi de artmaktadır, bu nedenle işlem sırasında kolaylık sağlaması açısından yukarıdaki iki boyutlu gösterim aşağıdaki gibi tek boyutlu bir diziye indirgenir.

$[G_{1,1}\ G_{1,2}\ G_{1,3}\ \ldots\ldots\ G_{1,24}\ G_{2,1}\ G_{2,2}\ \ldots\ldots\ G_{2,24}\ \ldots\ldots\ \ldots\ldots\ldots G_{10,1}\ G_{10,2}\ \ldots\ldots\ G_{10,24}]$

Tek boyutlu dizi biçimindeki kromozom kodlaması 10 ila 50 generatörlük bir sistem için uygundur, fakat generatör sayısı arttıkça dizi çok genişleyeceğinden genetik operatörler hem yavaşlayacak hem de tek veya iki noktalı çaprazlama işlevi çözüm uzayını daha büyük bir alanının taranmasında yetersiz kalacaktır. Böyle bir durumda ikiden çok noktadan çaprazlama yapmak veya düzgün (uniform) çaprazlama kullanmak çözümlerin daha düzenli olmasını sağlar.

5.3.2. Çaprazlama

Çaprazlama operatörü iki ebeveyn kromozomdan bir çift oğul kromozom (çözüm) oluşturulmasını sağlar. Genetik algoritmalarda kullanılan çaprazlamalar çoğunlukla tek ve iki noktadan çaprazlamadır. Kromozom bit dizisi büyük olduğundan bu çalışmada iki noktadan çaprazlama seçilmiştir. Buna göre ebeveynler rasgele iki noktadan bölünecek ve arada kalan parçaları değiştirilerek yeni iki birey oluşturulacaktır.

5.3.3. Mutasyon

Mutasyon genetik algoritmalarda çözümün yerel minimumlara takılmasını engellemek, çözüm uzayı üzerinde daha farklı noktalarda gezinmek için kullanılır. Önerilen mutasyon oranı L kromozom bit dizisi genişliği olmak üzere 1/L ile 2/L değerleri arasında olmasıdır.

5.3.4. Elitisizm

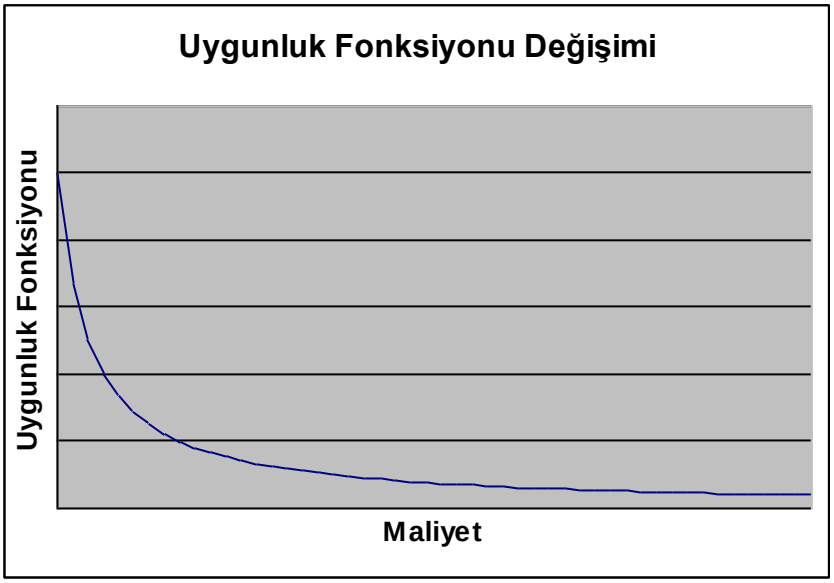
Elitisizm her bir yinelemede elde edilen en iyi uygunluk değerine sahip bireylerin bir sonraki kuşaklara aktarılarak en uygun sonucun arada kaybolmasını engeller ve sonuca yansımasını sağlar. Elitisizm oranı popülasyon içinden kaç bireyin bir sonraki yinelemeye gireceğini belirler. Bu oranın büyük olması yinelemelerin yerel maksimumlara takılmasına neden olabilir. Bu nedenle hem en iyi sonuçların kaybolmasını engelleyecek, hem de GA nın evrimini olumsuz etkilemeyecek bir oran seçilmelidir. Önerilen %1-3 arasında bir oran seçilmesidir [24]. Çalışmada bu nedenle elitisizm oranı %2 alınmıştır, popülasyon sayısı 100 olduğundan bu her yinelemede popülasyondaki en iyi iki bireyin bir sonraki yinelemeye doğrudan aktarılacağı anlamına gelir.

5.3.5. Uygunluk Fonksiyonu

Uygunluk fonksiyonu popülasyondaki her bir bireyin ne kadar iyi bir çözüm olduğunu göstermek için kullanılır. Uygunluk fonksiyonu genellikle doğrusal alınmak ile birlikte kullanılan çözüme göre logaritmik uygunluk fonksiyonları da kullanılabilir. Uygunluk fonksiyonu çözüme bağlı olduğundan (bazen çözüm değeri de uygunluk fonksiyonu olarak seçilebilir) GA'nın uygulandığı probleme göre değişiklik gösterir.

Bu çalışmada büyük maliyet değerlerini istenilmediğinden dolayı maliyet arttıkça daha küçük değer alıp seçme işlevinde devre dışı kalması için uygunluk fonksiyonu (5.10)'daki gibi alınmıştır. Amaçlanan en düşük maliyet değerini bulmak olduğundan uygunluk fonksiyonu toplam maliyeti içermektedir. Uygunluk fonksiyonunun maliyete göre değişimi Şekil 5.2 'de gösterilmiştir.

$$UF = \frac{1}{1 + ToplamMaliyet} \tag{5.10}$$



Şekil 5.2. Uygunluk Fonksiyonunun maliyete göre değişimi

Eğer uygunluk fonksiyonu 1/TM şeklinde alınır ise toplam maliyet düştükçe uygunluk fonksiyonu değeri önerilen yönteme göre daha az azalır, bu da kalitesiz çözüm üreten kromozomların popülasyonlar boyunca aktarılma ihtimalini arttırır. Düşük maliyetlerin yakınsamadaki etkisini arttırmak için uygunluk fonksiyonu (5.10) daki gibi önerilmiştir.

5.3.6. Seçim İşlevi

Seçim operatörleri çaprazlama için eşlemeye girecek bireylerin seçiminde kullanılır. Bu çalışmada iki tip seçim operatörü kullanılmıştır, bunlar turnuva seçimi ve rulet tekeri seçimi.

Turnuva seçimi için popülasyondan rasgele seçilen 6 birey arasından uygunluk fonksiyonu en iyi olan iki tanesi eşlemeye alınmıştır. Turnuva seçiminde kullanılan örneklenecek birey sayısını çok arttırmak çözümün belirli bir gruba kaymasına neden olur, bu durumda çözüm istenmeyen yerel minimumlara takılabilir ve en uygun sonuca ulaşılması zorlaşır. Bunu engellemek için bu sayının popülasyona oranı düşük olmalıdır.

Rulet tekeri seçimi ise popülasyondaki her bireyin uygunluk fonksiyonu değerini kullanarak sıralama yaptığından daha fazla hesaplama zamanı almaktadır. Diğer taraftan bu yöntemde iyi uygunluk değerine sahip bireylerin sonraki kuşaklara aktarılma ihtimali turnuva seçiminde olduğundan daha fazladır.

5.3.7. Kısıt Kontrol ve Düzeltme Operatörü

Bu operatör bir tür düzeltme operatörüdür. Popülasyon çaprazlama ve mutasyon işlevlerinden geçtikten sonra kısıtlara uymayan bireyler oluşması ihtimali yüksektir. Bu durumda ya penaltı operatörü ile bu bireylerin devre dışı bırakılması gerekir ya da kısıtları sağlayacak şekilde düzeltilmesi gerekir.

Kısıtların olmadığı bir birim katkı sorununun çözüm uzayı kısıtların olduğu durumdaki çözüm uzayından çok daha büyüktür. Bu durumda eğer penaltı operatörü kullanılır ise popülasyondaki bireylerin büyük bir oranı penaltıya düşeceğinden geçerli birey oranını arttırmak için popülasyon boyutunu oldukça büyük tutmak gerekir. Bu da doğal olarak her bir basamaktaki işlem süresini arttırır. Bunun yerine kısıtları ihlal eden bireyler için, düzeltme operatörü ile, en az devrede kalma/boşta kalma süreleri düzeltilerek geçerli bireyler oluşması sağlanır.

5.3.8. Penaltı Operatörü

Genetik algoritmalarda kısıtlar olduğunda penaltı veya düzeltme operatörlerinden sadece bir tanesi algoritmaya eklenir. Geçersiz bireyler ya düzeltilir ya da düşük uygunluk değeri verilir. Birim katkı sorununun ise karmaşıklığı nedeni ile uygulanan algoritmada hem düzeltme hem de penaltı operatörü kullanılmıştır, tabii ki bu iki operatör aynı adımda değil farklı yerlerde kullanılmıştır.

Kısıtların kontrolü için düzeltme operatöründen geçen bireylere ekonomik yük dağılımı uygulanmadan önce bireyin sağlayabileceği en fazla gücün ilgili saat için talep edilen güce eşit veya fazla olması kontrol edilir. Kromozomun kodladığı güç talep gücünden az ise bireyin uygunluk fonksiyonu değeri sıfır yapılır ve ekonomik yük dağılımına girmez. Aksi halde birey yük dengesi kısıtını sağladığından ekonomik yük dağılımına girer.

5.3.9. Başlangıç Popülasyonunun Oluşturulması

Başlangıç popülasyonu rasgele olarak üretilen, 1 ve 0'lardan oluşan bit dizisi şeklindeki çözümlerden (bireylerden) oluşur. Program her çalıştırıldığında farklı bir popülasyon oluşturur.

Popülasyon büyüklüğü olarak 100 birey seçilmiştir. Popülasyon büyüklüğü fazla tutulmamış, bunun yerine evrimleşen kuşak sayısı arttırılmıştır. Bu sayede genetik algoritmaların mantığına daha uygun, daha çok evrimleşen bir yapı kurulmuştur.

Bazı uygulamalarda başlangıç popülasyonu tamamen rassal olarak oluşturulmayıp probleme göre akıllı bazı algoritmalarla yönlendirilerek oluşturulmaktadır. GA yapısı gereği olasılıksal olarak işlem yapıp çözüme ulaştığından bu uygulamada başlangıç popülasyonu tamamen rassal olarak oluşturulmuştur. Oluşturulan popülasyon daha sonra kısıtların kontrolü ve düzeltilmesi işleminden geçmiştir.

5.3.10. Ekonomik Yük Dağılımı

Yük dağılımı kendi başına bir problem olmasına rağmen birim katkı sorunu yapısı gereği bu problemini içinde barındırmaktadır. Ekonomik yük dağılımı hesaplamanın yoğun olduğu bir basamak olduğundan her yinelemede popülasyona bir defa uygulanmaktadır. Bunun yanında penaltı fonksiyonu ile yük dengesini sağlamayan bireylerin ekonomik yük dağılımı adımına girmesi engellenmekte ve işlem süresinden tasarruf edilmektedir.

Ekonomik yük dağılımı aşamasında generatörlerin en düşük/en yüksek güç sınırlarını aşmamak için öncelikle bütün generatörlere izin verilen en az yükleri eklenmekte daha sonra ilgili saat için talep edilen güce ulaşana kadar yineleme yapılarak ile generatörlere talep gücü dağıtılmaktadır. İterasyonda herhangi bir generatör en fazla gücüne ulaştığında ilgili generatör yineleme dışı bırakılmaktadır. Ekonomikliği sağlamak için generatörler yakıt maliyet katsayılarına bakılarak yüklenmektedir.

5.4. Uygulanan Genetik Algoritma

Uygulanan algoritma GA'da kullanılan genel yineleme yöntemidir. Problemden dolayı bazı özel basamaklar bulunmaktadır.

Bölüm 5.3 de aşamaları detaylı anlatılan genetik algoritmanın takma kodu aşağıdaki gibidir.

```
Başlangıç popülasyonunu oluştur
Popülasyondaki her birey için
   Kısıt kontrol ve düzeltme işlevini uygula
   Penaltı işlevini uygula
   Ekonomik yük dağılımını uygula
   Uygunluk değerini bul

Kuşak sayısı=0
(Kuşak sayısı < istenilen kuşak sayısı) olduğu sürece tekrarla
   Yeni boş popülasyon oluştur
   En iyi iki bireyi yeni popülasyona kopyala
   Yeni Popülasyon doluncaya kadar tekrarla
      Seçme operatörü ile iki birey al
      Çaprazlama uygula
      Mutasyon uygula
      Yeni popülasyona ekle
   Eski popülasyonu yok et
   Popülasyondaki her birey için
      Kısıt kontrol ve düzeltme işlevini uygula
      Penaltı işlevini uygula
      Ekonomik yük dağılımını uygula
      Uygunluk değerini bul
   Kuşak sayısı=Kuşak sayısı + 1
En iyi bireyi bul (çözüm)
```

İlk başlangıçta oluşturulan popülasyon büyük oranda kısıtlara uymayan, geçersiz bireyleri barındırmaktadır. Bu nedenle her yinelemede popülasyona uygulanan kısıt kontrolü ve düzeltmesi ile penaltı işlevi uygulanmakta, daha sonra bireylerin uygunluk değerlerini bulabilmek için ekonomik yük dağılımı uygulanmaktadır.

Test sisteminde yinelemenin yapılacağı kuşak sayısı değeri 2500 olarak belirlenmiştir. Bu değerin belirlenmesi için öncelikle program beş kez 3500 yinelemeye kadar çalıştırılmıştır, daha sonra çözümlerin hangi kuşak sayısından sonra çok ufak değişimler gösterdiği incelenmiş ve 2500. kuşaktan sonra toplam maliyetin 3500. kuşağa oranının %99'un üzerinde olduğu görülmüştür. İşlem süresini arttırmamak ve kalite/çalışma süresi oranını yüksek tutmak için kuşak sayısı değeri 2500 alınmıştır. Birim katkı sorunu kısıtları ve karmaşıklığı nedeni ile bu kadar yüksek bir yinelemeye ihtiyaç duymaktadır.

Algoritmada bulunan en iyi çözümlerin kaybolmayıp, sonraki kuşaklara da aktarılması amacıyla elitizm uygulanmış ve her yinelemede en iyi iki birey hiçbir işleme tabii tutulmadan yeni popülasyona kopyalanmıştır.

5.5. Uygulama Ortamı

Uygulama programı yazılırken java programlama dili kullanılmıştır. Java programlama dili; nesne yönelimli olması, platformdan bağımsız çalışması, 10 yıllık geçmişi boyunca güvenilirliğini kanıtlaması ve sürümlerinin kararlı olması nedeniyle seçilmiştir. Uygulama java dilinin son alfa sürümü olan j2sdk 1.4.03 ile derlenip çalıştırılmıştır.

Uygulamanın çalıştığı bilgisayar Pentium IV, 2.4 GHz işlemci ve 1GB RAM özelliklerine sahiptir. Uygulama Windows 2003 Server işletim sistemi üzerinde çalıştırılmış ve program çalışırken bütün kaynaklar bu uygulamaya aktarılmıştır.

5.6. Uygulama Sonucu

Yapılan çalışmada bulunan sonuçlar şu şekildedir.

- Bulunan en iyi günlük maliyet değeri $576996 olmuştur.

  Bulunan maliyet değeri test verilerinin alındığı makalenin maliyet değerine yakındır [26]. Buna dayanarak uygulamanın başarılı olduğuna kanaat getirebiliriz.

- Yaklaşık 2500 kuşaktan sonra çözüm en iyi sonuca yakınsamaktadır.

  Birim katkı sorunu çözüm uzayı oldukça geniş olan bir problemdir. Bu nedenle optimum sonuçlara yüksek yineleme sayısıyla ulaşılması beklenen bir sonuçtur. Sistemde düzeltme işlevlerinin kullanılması bu sayının artmasını engellemiştir. Düzeltme işlevi yerine devreye girme, devreden çıkma kısıtları için de penaltı operatörleri kullanılsa idi oluşturulan popülasyonlardaki geçersiz birey sayısı artacak, çözüm uzayında daha az nokta taranacaktı. Düzeltme işlevi kullanarak daha fazla geçerli bireyin evrime katılması sağlanmış oldu.

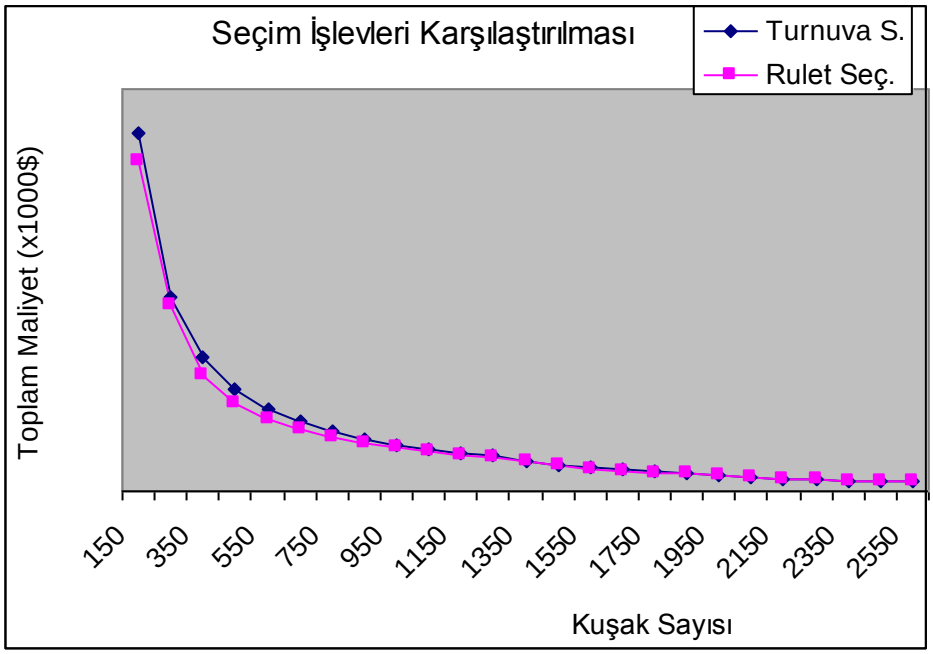
- Turnuva seçimi ile rulet tekeri seçimi işlevlerinden önemli bir üstünlüğü olan seçim işlevi çıkmamıştır (Bkz. Şekil 5.3).

  Genetik algoritmalarda kullanılan parametreler problemlere özel davranacağından bu sonucu bütün problemler için genellemeyip sadece birim katkı sorununa genelleme yapabiliriz.

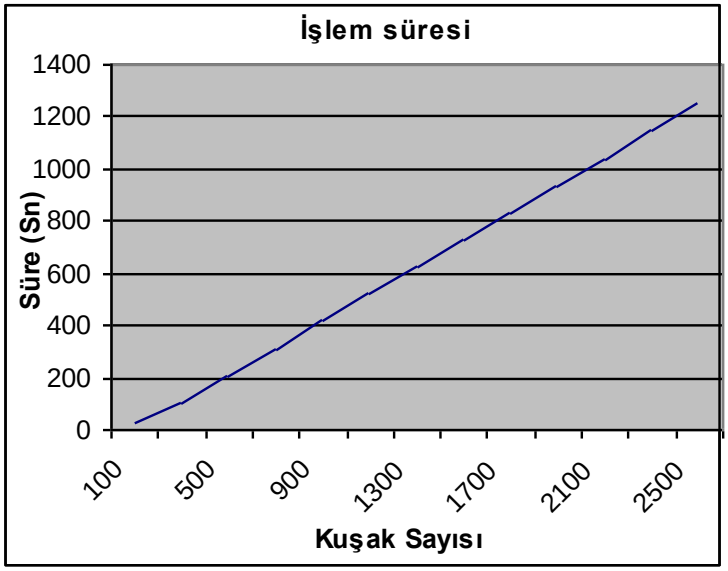
- Yapılan hesaplamalarda işlem süresi kuşak sayısıyla orantılı olarak artmaktadır (Bkz. Şekil 5.4).

  GA'nın yapısı gereği her yinelemede yapılan işlemler aynı olduğundan bu beklenen bir sonuçtur. Yineleme sayısı arttıkça algoritma en iyi çözüme yakınsamaktadır, fakat çözümdeki değişmenin süreye oranı da oldukça azalmaktadır. Kabul edilebilir sonuç değerlerinden sonra programın çalıştırılması fazla bir etki sağlamadığından daha fazla kuşak üretilmesi anlamsızdır.

Yapılan yineleme sonucunda 24 saatlik süre için bulunan generatör açık/kapalı değerleri ile ekonomik yük paylaşımı sonucu ortaya çıkan, generatörlerin saatlik yük dağılımları EK B'de verilmiştir.



Şekil 5.3 Gerçeklenen uygulamada seçim işlevlerinin karşılaştırılması



Şekil 5.4 Gerçeklenen uygulamada işlem süresinin kuşak sayısına göre değişimi.

6. SONUÇLAR

Elektrik enerji sistemlerinin temelinde enerjinin üretimini sağlayan üretim birimleri bulunur. Enerjinin üretilmesinden sorumlu birimlerden her birinin sistem karakteristiği ve yakıtı farklı olabilir, dolayısıyla her birimin elektriği üretim maliyeti farklıdır. Belirli bir anda sistemin sağladığı toplam enerjinin maliyetini en düşük seviyede tutmak için sistemdeki üretim birimleri arasından üretim maliyeti en az olanlar devrede olmalıdır. Diğer taraftan üretim birimleri mekanik ve termik özelliklerinden dolayı bazı kısıtlamalar getirmektedir. Bu kısıtlamalar generatörlerin istenildiği zaman devreye girip çıkarılmasını, gücün istenilen oranda değişmesini engeller. Üretim birimlerinden alınan enerjinin toplam maliyetini en düşük miktarda tutmak amacıyla generatörlerin devreye alınıp çıkarılmalarının, kısıtlar da göz önüne alınarak, belirli bir zaman aralığı için düzenlenmesi problemine birim katkı sorunu adı verilir.

Birim katkı sorununda çözüm uzayı çok büyüktür ve analitik çözümü yoktur. Bu da çözüm uzayında olasılıksal olarak gezinen yöntemlerin başarısını arttırmaktadır. Genetik algoritmalar olasılıksal oluşu ve doğrusal olmayan programlamadaki başarıları nedeniyle bu tip uygulamalardaki başarısını kanıtlamıştır. GA'nın başarısı ve BKS'ye GA uygulayarak çözüm arayan çalışmalar göz önüne alınarak bu çalışmada BKS'nun GA ile çözümü uygulanmıştır.

Genetik algoritmalar hızlı ve eniyilemeli bir yöntem olması nedeni ile performansın ve işlem hızının önemli olduğu problemlerde çok kullanılan bir yöntemdir. Özellikle elektrik enerji sistemleri gibi saatlik çalışmanın ön planda olduğu ve sonuçların kısa zamanda bulunup uygulamaya geçilmesi gereken sistemlerde etkili bir yöntem olarak kullanılabilir.

Çalışmada öncelikle BKS'nun, maliyet nedenleri ve maliyetlerin formülasyonları verilmiştir. Kısıtların nedenleri ve matematiksel gösterimi yapıldıktan sonra genel amaç formülasyonu gösterilmiştir.

Üçüncü bölümde genetik algoritmaların işlevleri, adımları ve parametreleri anlatılmıştır. İşlevler genetik algoritmanın yinelemesi sırasındaki adımlardır ve her

adımda kullanılabilecek yöntemlerden bahsedilmiştir. GA da kullanılan parametreler mutasyon oranı, çaprazlama oranı, elitizm oranı ve popülasyondaki birey sayısıdır.

Dördüncü bölümde bugüne kadar yapılan çalışmalar incelenmiş, yeniliklerine, birbirlerinden farklarına ve olumsuzluklarına değinilmiştir. Çalışmalar genel olarak kromozom kodlamasında öneriler sunmaktadır, bu sayede kısıtların bir kısmı kodlamaya eklenerek düzeltme ve penaltı işlevlerinin yükü ve karmaşıklığı azaltılmaya çalışılmıştır. BKS'nin çözümünde paralel yöntemler de kullanılmıştır. Paralelleştirme biçimlerini birim katkı sorunundan çok çözüm yöntemi olan GA belirlemektedir.

Beşinci bölümde yapılan uygulama anlatılmıştır. Uygulama Java programlama dili kullanılarak yazılmıştır. Platformdan bağımsız olması, nesne yönelimli olması ve kararlı sürümleri nedeniyle bu dil seçilmiştir. Test sisteminde 10 generatörlük bir sistem alınmış, sistemin MUT, MDT, yük dengesi, yedek güç ve generatör limitleri kısıtları dikkate alınarak çözüm yapılmıştır. Amaç fonksiyonuna yakıt maliyeti ve başlatma maliyeti dahil edilmiştir.

6.1. Sonuçlar

Yapılan çalışmada bulunan sonuçlar şu şekildedir.

- Uygulanan örnek sistem için bulunan en iyi günlük maliyet değeri $576996 olmuştur. Bu değer test verilerinin alındığı makalenin maliyet değerine yakın fakat daha kötüdür. Yineleme sayısı daha fazla tutulur ise sonuçların kalitesi az da olsa artar diğer taraftan yineleme sayısının artışı çalışma süresini doğru orantılı olarak arttırması bir dezavantajdır. Bunun yanında kullanılan ekonomik yük dağılımı (EYD) sonuçların kalitesini doğrudan etkilemektedir. Örnek alınan makalelerde kullanılan EYD çok daha karmaşıktır ve sonuçların kalitesini arttırmaktadır. Bu çalışmada ise EYD birinci öncelik olmadığından daha basit bir yöntem kullanılmıştır. EYD ve yineleme sayısının daha fazla arttırılması gereği bulunan sonucun referans makalelerden daha kötü olmasına neden olmuştur.
- Programın çalıştırılmasında kuşak sayısı sınırı olarak 2500 alınmıştır. Birim katkı sorunu çözüm uzayı oldukça geniş olan bir problemdir. Bu nedenle

optimum sonuçlara yüksek yineleme sayısıyla ulaşılması beklenen bir sonuçtur. Sonuç kalitesinin artması için bu sayı daha da arttırılabilir.

- Turnuva seçimi ile rulet tekeri seçimi işlevlerinden önemli bir üstünlüğü olan seçim işlevi çıkmamıştır. Ara yinelemelerde rulet tekeri seçimi daha kaliteli sonuçlar vermesine rağmen kuşak sayısı sınırına yaklaştıkça elde edilen sonuçlar neredeyse aynı olmuştur, bu da iki yöntemin birbirine önemli üstünlüğü olmadığını göstermektedir.
- Yapılan hesaplamalarda işlem süresi kuşak sayısıyla doğru orantılı olarak doğrusal biçimde artmaktadır. GA nın yapısı gereği her yinelemede yapılan işlemler aynı olduğundan doğrusal orantı beklenen bir sonuçtur, çünkü ilk başta yeni popülasyon oluşturulup kısıtlarının kontrol edilmesi düşünülmez ise yapılan her yinelemedeki işlemler birebir aynıdır.

6.2. Yapılabilecek Çalışmalar ve Öneriler

Daha ileri bir çalışma olarak, birim katkı sorununun çözüm aşamalarından biri olan ekonomik yük dağılımı (EYD) yapılır iken bulanık karar verme (fuzzy decision making) kullanılarak GA'nın optimum şekilde katkısına karar verdiği üretim birimleri arasındaki yük dağılımı daha iyi yapılarak maliyet azaltılabilir ve sonuçların kalitesi arttırılabilir. EYD nin işlem süresi olarak programda önemli bir etkisi vardır. Daha kaliteli sonuçları daha hızlı üreten EYD programın geliştirilmesine önemli katkılar sağlar.

Programın geliştirilmesinde kullanılabilecek bir diğer yöntem ise başlangıç popülasyonunun rasgele üretilmesi yerine akıllı yöntemler ile üretilmesidir. Basit iyileştirmeler ile oluşturulan başlangıç popülasyonu daha az kuşak sayısı ile (ve daha az işlem süresi ile) aynı sonuçlara ulaşılmasını sağlar. Diğer taraftan başlangıç popülasyonunda rasgele üretimi tamamen ortadan kaldırmak genetik algoritmanın evrimleşmesine olumsuz etki eder ve sonuçların yerel maksimumlarda takılmasına neden olabilir. Önerilen akıllı yöntemler kullanan fakat rasgeleliği tamamen ortadan kaldırmayan bir yöntemdir.

Birim katkı sorunu sağladığı ekonomik avantajlar nedeni ile bu konuda daha birçok çalışmanın yapılmasına neden olacaktır. Bilgisayarların her geçen gün artan işlemci

gücü sayesinde aynı sürede işlem zamanı ile programların sonuç kalitesi ise daha da artacaktır.

KAYNAKLAR

[1]. **Kazarlis, S. A., Bakirtzis, A. G. and Petridis V.,** 1996. A Genetic Algorithm Solution to the Unit Commitment Problem, *IEEE Transactions on Power Systems*, **11(1)**, 83-92.

[2]. **Rajan, C. A. and Mohan, M. R.**, 2004. An evolutionary programming-based tabu search method for solving the unit commitment problem, *IEEE Transactions on Power Systems*, **19(1)**, 1165-1172.

[3]. **Ongsakul, W., Petcharaks, N.**, 2004. Unit commitment by enhanced adaptive Lagrangian relaxation, *IEEE Transactions on Power Systems*, **19(1)**, 620-628.

[4]. **Senjyu, T., Shimabukuro, K., Uezato, K. and Funabashi, T.**, 2003. A fast technique for unit commitment problem by extended priority list, *IEEE Transactions on Power Systems*, **18(2)**, 882-888.

[5]. **Purushothama, G. K., Jenkins, L.**, 2003. Simulated annealing with local search-a hybrid algorithm for unit commitment, *IEEE Transactions on Power Systems*, **18(1)**, 273-278.

[6]. **Borghetti, A., Frangioni, A., Lacalandra, F. and Nucci, C. A.**, 2003. Lagrangian heuristics based on disaggregated Bundle methods for hydrothermal unit commitment, *IEEE Transactions on Power Systems*, **18(1)**, 313-323.

[7]. **Qiaozhu Z., Xiaohong G. and Jian C.**, 2002. Unit commitment with identical units successive subproblem solving method based on Lagrangian relaxation, *IEEE Transactions on Power Systems*, **17(4)**, 1250-1257.

[8]. **Takriti, S. and Birge, J. R.**, 2000. Using integer programming to refine Lagrangian-based unit commitment solutions, *IEEE Transactions on Power Systems*, **15(1)**, 151-156.

[9]. **Cheng, C. P., Liu, C. W. and Liu, C. C.**, 2000. Unit commitment by Lagrangian relaxation and genetic algorithms, *IEEE Transactions on Power Systems*, **15(2)**, 707-714.

[10]. **Mantawy, A. H., Abdel-Magid, Y. L. and Selim, S. Z.**, 1999. Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem, *IEEE Transactions on Power Systems*, **14(3)**, 829-836.

[11]. **Mantawy, A. H., Abdel-Magid, Y. L. and Selim, S. Z.**, 1998. A simulated annealing algorithm for unit commitment, *IEEE Transactions on Power Systems*, **13(1)**, 197-204.

[12]. **Saneifard, S., Prasad, N. R. and Smolleck, H. A.**, 1997. A fuzzy logic approach to unit commitment, *IEEE Transactions on Power Systems*, **12(2)**, 988-995.

[13]. **Huang, S. J. and Huang, C. L.**, 1997. Application of genetic-based neural networks to thermal unit commitment, *IEEE Transactions on Power Systems*, **12(2)**, 654-660.

[14]. **Koza, J. R.**, 1992. Genetic Programming, Bradford Book, The MIT Press, Massachusetts.

[15]. **Goldberg, D. E.,** 1989. Genetic Algorithms in Search Optimization and Machine Learning, Addion Wesley Publishing Company, USA.

[16]. **Momoh, J. A.,** 2001. Electric Power System Applications of Optimization, Marcel Dekker Inc., New York.

[17]. **Swarup, K. S. and Yamashiro, S.,** 2002. Unit Commitment Solution Methodology Using Genetic Algorithm, *IEEE Transactions on Power Systems*, **17(1)**, 87-91.

[18]. **Swarup, K. S. and Yamashiro, S.,** 2003. A genetic algorithm approach to generator unit commitment, *International Journal of Electrical Power & Energy Systems*, **25(9)**, 679-687.

[19]. **Rudolf, A. and Bayrleithner, R.,** 1999. A genetic algorithm for solving the unit commitment problem of a hydrothermal power system, *IEEE Transactions on Power Systems*, **14(4)**, 1460-8.

[20]. **Rechenberg I**., 1973. Evolutionstrategie: Optimierung technisher Systeme nach Prinzipien der biologischen Evolution, Fromman-Holzboog Verlag, Sttutgart.

[21]. **Holland, J. H.,** 1975. Adaptation in Natural and Artifical System, Ann Arbor, The University of Michigan Press, USA.

[22]. **"Introduction to genetic algorithms with Java applets",** Marek Obitko, 1998, http://cs.felk.cvut.cz/~xobitko/ga/.

[23]. **Michalewicz, Z., 1996**. Genetic algorithms + data structures = evolution programs, Springer-Verlag, New York.

[24]. **Coley, D. A.**, 1999. An introduction to genetic algorithms for scientists and engineers, River Edge, NJ, World Scientific.

[25]. **Yiying, Z., Wang F. and Xiaomin B.,** 2002. Unit Commitment Problems Using GA, *PowerCon 2002 International Conference on Power System Technology*, 1, 625-631, October 13-17.

[26]. **Juste K. A., Kita, H., Tanaka, E. And Hasegawa J.,** 1999. An Evolutionary Programming Solution to the Unit Commitment Problem, *IEEE Transactions on Power Systems*, **14(4)**, 1452-9.

[27]. **Arroyo, J.M. and Conejo, A.J.,** 2002. A Paralel Repair Genetic Algorithm to Solve the Unit Commitment Problem, IEEE Transactions on Power Systems, 17(4), 1216-24.

[28]. **Yang, H.T., Yang, P.C. and Huang, C.L.,** 1997. A Parallel Genetic Algorithm Approach to Solving the Unit Commitment Problem Implementation on the Transputer Networks, IEEE Transactions on Power Systems, 12(2), 661-8.

EK A

TEST GENERATÖRLERİNE AİT KARAKTERİSTİK TABLOLARI VE GÜNLÜK YÜK EĞRİSİ TABLOSU

Birim No	Üretim Limitleri		Yakıt Maliyeti Katsayıları		
	P_{min}	P_{max}	a	b	c
1	250.00	520.00	105.00	1.395	0.00130
2	120.00	320.00	49.00	1.264	0.00290
3	75.00	200.00	82.00	1.214	0.00150
4	75.00	280.00	72.00	1.350	0.00260
5	50.00	150.00	29.00	1.540	0.00210
6	50.00	150.00	100.00	1.329	0.00140
7	25.00	120.00	32.00	1.500	0.00380
8	30.00	100.00	40.00	1.350	0.00390
9	20.00	80.00	25.00	1.500	0.00400
10	15.00	60.00	15.00	1.400	0.00510

Boşta Kalma / Çalışma Süreleri		Başlatma Maliyeti Katsayıları		
MUT	MDT	α	δ	τ
10	4	60.00	207.00	11.00
7	5	50.00	137.00	7.00
6	6	70.00	157.00	9.00
6	3	30.00	146.00	6.00
1	3	30.00	130.00	5.00
3	2	80.00	202.00	11.00
3	3	12.00	100.00	5.00
4	4	25.00	110.00	5.00
3	4	15.00	123.00	5.00
3	2	15.00	123.00	5.00

Saatlere Göre Günlük Talep Yük Miktarı (MW)

Saat	Yük	Saat	Yük	Saat	Yük
1	700	9	1300	17	1000
2	750	10	1400	18	1100
3	850	11	1450	19	1200
4	950	12	1500	20	1400
5	1000	13	1400	21	1300
6	1100	14	1300	22	1100
7	1150	15	1200	23	900
8	1200	16	1050	24	800

EK B

UYGULAMA SONUCU BULUNAN SAATLİK GENERATÖR DURUMLARI VE YÜK DAĞILIMLARI

Saat		Generatör Yükleri (MW)										Yük (MW)	$P_{G,Max}$ (MW)	Yedek Yük (MW)
	1234567890	PI	P2	P3	P4	P5	P6	P7	P8	P9	P10			
1	1100000000	455	245	0	0	0	0	0	0	0	0	700.00	910.00	210.00
2	1100000000	455	295	0	0	0	0	0	0	0	0	750.00	910.00	160.00
3	1100000000	455	395	0	0	0	0	0	0	0	0	850.00	910.00	60.00
4	1110000000	455	365	130	0	0	0	0	0	0	0	950.00	1040.00	90.00
5	1110000000	455	415	130	0	0	0	0	0	0	0	1000.00	1040.00	40.00
6	1111000000	455	385	130	130	0	0	0	0	0	0	1100.00	1170.00	70.00
7	1111100000	455	435	130	130	0	0	0	0	0	0	1150.00	1332.00	182.00
8	1111100000	455	455	130	130	30	0	0	0	0	0	1200.00	1332.00	132.00
9	1111100000	455	455	130	130	130	0	0	0	0	0	1300.00	1332.00	32.00
10	1111111100	455	455	130	130	162	68	0	0	0	0	1400.00	1555.00	155.00
11	1111111100	455	455	130	130	162	80	38	0	0	0	1450.00	1555.00	105.00
12	1111111110	455	455	130	130	162	80	88	0	0	0	1500.00	1610.00	110.00
13	1111110000	455	455	130	130	162	68	0	0	0	0	1400.00	1412.00	12.00
14	1111110000	455	455	130	130	130	0	0	0	0	0	1300.00	1412.00	112.00
15	1111110000	455	455	130	130	30	0	0	0	0	0	1200.00	1412.00	212.00
16	1111100000	455	335	130	130	0	0	0	0	0	0	1050.00	1332.00	282.00
17	1111100000	455	415	130	0	0	0	0	0	0	0	1000.00	1332.00	332.00
18	1111100000	455	385	130	130	0	0	0	0	0	0	1100.00	1332.00	232.00
19	1111110000	455	455	130	130	30	0	0	0	0	0	1200.00	1412.00	212.00
20	1111111000	455	455	130	130	162	68	0	0	0	0	1400.00	1500.00	100.00
21	1111110000	455	455	130	130	130	0	0	0	0	0	1300.00	1412.00	112.00
22	1110000000	455	455	130	0	0	0	0	0	0	0	1100.00	1040.00	- 60.00

23	1100000000	455	445	0	0	0	0	0	0	0	0	900.00	910.00	10.00
24	1100000000	455	345	0	0	0	0	0	0	0	0	800.00	910.00	110.00
Toplam		7869.00												

ÖZGEÇMİŞ

Mehmet Emin SAPMAZ 1979 yılında İzmir'de doğdu. İlk ve orta öğrenimini sırasıyla İnkılap İlkokulu ve Alsancak Ortaokulunda tamamladı, daha sonra İzmir Çınarlı Teknik Lisesi Bilgisayar bölümünde 4 yıl öğrenim gördü. 1996 yılında İstanbul Teknik Üniversitesinde İngilizce hazırlığı tamamladıktan sonra 1997 – 2001 yılları arasında İstanbul Teknik Üniversitesi, Elektrik – Elektronik Fakültesi, Elektrik Mühendisliği Bölümünde eğitimini tamamlayarak Elektrik Mühendisi unvanını aldı. 2001 yılında İTÜ Fen Bilimleri Ensitüsü Elektrik Mühendisliği Programında yüksek lisans eğitimine başladı, halen yüksek lisan eğitimine devam etmekte ve Fen Bilimleri Enstitüsünde araştırma görevlisi kadrosu ile bilgi işlem sorumlusu olarak görev yapmaktadır.