GENETİK ALGORİTMALARLA DERS PROGRAMI HAZIRLAMA OTOMASYONU TASARIMI

Mustafa Müjdat ATANAK[1] **Fatih Onur HOCAOĞLU[2]**

[1,2]Elektrik- Elektronik Mühendisliği Bölümü
Mühendislik Fakültesi
Anadolu Üniversitesi, 26470, İki Eylül Kampusu, ESKİŞEHİR

[1]e-posta: mmatanak@anadolu.edu.tr [2]e-posta: fohocaoglu@anadolu.edu.tr

Anahtar sözcükler: Otomasyon Yazılımları, Genetik Algoritmalar

ABSTRACT

In this study, an NP-Complete problem, namely the classroom scheduling problem is heuristically solved by using the genetic algorithm techniques. In the view of some constraints such as the efficiency of classroom utilization and the time preferences of the lecturers, a program is written to find a solution that is optimized for given constraints. In this program, near optimal solution is generated using heuristic methods. The program is made flexible via some parameters that can be adjusted by the user to give some constraints more importance than the others. Consequently, the program becomes appropriate for different educational institutions with various needs.

1. GİRİŞ

Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkansız olan polinomsal zamanda çözülemeyen problemler olarak da bilinen problemlerin çözümünde kullanılmaktadır. Genetik algoritmalar birçok alanda uygulanabilmektedirler. Bunlardan bazıları deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleri olarak bilinir. Mühendislik alanında birçok problemde optimizasyon tekniği olarak kullanılan genetik algoritmalar; otomasyon sistemlerinde mekanizma tasarımında, görüntü işleme tekniklerinde, geleneksel kontrol problemlerinde, güç sistemlerinde, üretim hattı yerleşimi planlaması gibi optimizasyon gerektiren problemlerde ve daha birçok alanda kullanılmaktadır [1].

Genetik algoritma ile bir problemi çözebilmek için öncelikle rasgele başlangıç çözümleri belirlenmektedir. Bu çözüm kümesi populasyon olarak isimlendirilir. Bir populasyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir populasyon oluşturmak için kullanılır. Bu amaçla bu çözümler için performanslar hesaplanır ve çözümler birbirleriyle eşleştirilerek yeni çözümler oluşturulur. Algoritmada üretilen çok sayıda çözümden performansı yüksek olanlar aranmaktadır. Bu arama, yeterince iyi çözüm üretilinceye kadar devam etmektedir. Genetik algoritmalar ile problemlerin çözülmesinde arzu edilen sonucu üretecek özelliklerin, kalıtım yolu ile başlangıç çözümlerinden elde edilen yeni çözümlere, onlardan da daha sonraki çözümlere geçtiği kabul edilmektedir [2].

Bu çalışmada ele alınan tipteki problemler, çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte ve polinomsal zamanda çözüme ulaşmak mümkün olmamaktadır. Genetik algoritmalarla ise, kısa bir sürede kabul edilebilir bir sonuç bulunabilmektedir.

Yapılan bu çalışmada genetik algoritmalar kullanılarak sınıfların boyutunu, ders sayılarını, derse kaydolması beklenen öğrenci sayılarını ve dersi verecek olan öğretim üyesini girdi olarak alan ve verilen sınırlamalar altında dönemlik ders programı hazırlayan bir otomasyon yazılımı geliştirilmiştir. Geliştirilen yazılım öğretim üyelerinin derslerini vermek için tercih edecekleri zamanları da girdi olarak alabilmektedir. Yazılım, olası çözümler içinde en iyisini aramakta ve optimale yakın makul çözümler üretmektedir. Ayrıca geliştirilen yazılım parametreleri ve sınırlamaları değiştirmeye imkan sağlayan esnek bir yapıya sahiptir.

2. GENETİK ALGORİTMALAR

Bir genetik algoritmanın temel elemanları şu şekilde özetlenebilir [3];

a) Kromozom ve Gen
Genetik algoritmanın çözmesi istenen problemin her bir çözümünü göstermektedir. Bir problem için çok sayıda çözüm olabilir. Genetik algoritmadan bunların arasındaki en iyi çözümü arayıp bulması istenir. Kromozomlar, bu çözümleri gösterirler. Başlangıçta rasgele alınan çözümler daha sonra genetik algoritmanın çalışma ilkesine göre iyileştirilmektedir. Kromozom elemanlarından her birisi çözümün bir

özelliğini göstermektedir. Bunlara da gen denilmektedir.

b) Çözüm Havuzu
Problemin en iyi çözümünü aramak için kullanılan ve rasgele belirlenmiş başlangıç çözüm setidir. Probleme göre değişen sayıda belirlenebilir. Probleme göre adapte edilebilen bu sayıya havuzun büyüklüğü denmektedir.

c) Çaprazlama
Problemin çözüm havuzunda bulunan çözümleri ikişer ikişer birleştirerek yeni çözümler üretmektir. Çaprazlamanın ne kadar sıklıkla yapılacağını belirleyen parametreye çaprazlama oranı adı verilmektedir. Optimum çaprazlama oranları, ele alınan probleme bağlı olarak genellikle %20-%70 arasında değişmektedir.

d) Mutasyon
Çaprazlama sonucunda farklı çözümlere ulaşmak bazen zor olmaktadır. Yeni çözüm aramanın kolaylaştırılması ve aramanın yönünü değiştirmek amacı ile kromozomun bir elemanının (genin) rastlantısal olarak değiştirilmesi işlemidir. Mutasyonun ne kadar sıklıkla yapılacağını belirleyen parametre mutasyon oranı olarak bilinir. Optimum mutasyon oranları, ele alınan probleme bağlı olarak genellikle %5-%50 arasında değişmektedir.

e) Uygunluk fonksiyonu
Çözüm havuzundaki kromozomların performans derecelerin ölçülmesini sağlayan bir fonksiyondur. Her problem için bir uygunluk fonksiyonunun belirlenmesi gerekmektedir. Bu fonksiyon ele alınan probleme göre değişir. Bir kromozom için fonksiyonun verdiği değer ne kadar büyük ise o kromozom o kadar sağlıklıdır ve iyi bir çözümdür.

f) Yeniden üretim
Çözüm havuzundaki kromozomlar çaprazlama ve mutasyon neticesinde üretilen yeni kromozomlarla çoğaltılacaktır. Bunların arasından havuz büyüklüğüne göre daha önceden belirlenmiş sayıda kromozom seçilerek diğerleri atılır. Seçilenler ise bir sonraki nesil çözüm adayı olarak yeniden çaprazlanıp gelecek nesil çözümleri üretirler. İyi çözümlerin bir sonraki nesil için seçilme olasılıkları daha fazladır. Ancak kötü çözümler, ileride iyi çözümlere sebep olma olasılıkları da göz önüne alınarak, daha düşük olasılık ile de olsa seçilebilirler[4].

Genetik algoritmanın aşamaları şu şekilde özetlenebilir[5]:

1. *Başlangıç*: n adet kromozom içeren populasyonun oluşturulması (Her kromozom potansiyel bir çözümdür),

2. *Uygunluk*: Her x kromozomu için uygunluğun f(x) değerlendirilmesi,

3. *Yeni populasyon*: Yeni populasyon oluşturmak için aşağıdaki adımlar tekrar edilir:

a. Seçim: İki ebeveyn kromozomun uygunluğuna göre seçimi (uygunluk ne kadar yüksek olursa seçilme şansı o oranda artar).
b. Çaprazlama: Yeni bir fert oluşturmak için ebeveynlerin bir çaprazlama sıklığı göz önüne alınarak çaprazlanması. Eğer çaprazlama yapılmazsa yeni fert anne veya babanın kopyası olacaktır.
c. Mutasyon: Yeni ferdin mutasyon olasılığına göre kromozom içinde rastsal bir değişikliğe gidilir.
ç. Ekleme: Yeni birey eğer yeteri kadar sağlıklı ise yeni populasyona eklenir.

4. *Test ve döngü*: Eğer sonuç tatmin ediyorsa algoritmanın sona erdirilmesi ve son populasyonun en iyi uyumdaki bireyi çözüm olarak sunulur. Sonuç tatmin etmiyorsa 2. adıma geri dönülür.

Genetik algoritmalar kullanılarak bir problem çözülecekse algoritmanın ne zaman sonlanacağına kullanıcı karar vermektedir. Genetik algoritmaların belli bir sonlanma kriteri yoktur. Sonucun yeterince iyi olması veya yakınsamanın sağlanması algoritmanın durması için kriter olarak kullanılabilir.

3. DERS PROGRAMI HAZIRLAMA PROBLEMİNİN BELİRLENMESİ

Geliştirilen yazılımda çözülmesi istenilen probleme ilişkin sınır değerleri ve bazı tanımlamalar şu şekildedir:

Yazılıma maksimum 100 adet farklı boyutlarda sınıf, bir dönemde verilmesi düşünülen 1000 adet ayrı ders, 300 Adet ders verecek öğretim üyesi, 20 Adet birbiriyle çakışmayan zaman dilimi, girdi olarak verilebilmektedir.

Tanımlanan problemde öğretim üyelerinden derslerini vermek için tercih edecekleri iki zaman dilimi ve derslerini vermek istemedikleri bir zaman dilimi sorulur ve bu parametrelerde yazılımda kullanıcıdan girdi olarak istenmektedir.

Bunlarla birlikte çözüm uzayının büyüklüğü ve programın kaç defa çalışacağı da parametre olarak kullanıcı tarafından ayarlanabilmektedir. İçinde arama yapacağımız uzayın büyük olması daha iyi bir çözüm bulabilmemizi sağlamakla beraber programın çalışma süresini arttırmaktadır. Gerçekleştirilen yazılımda çözüm uzayımızın büyüklüğü 20, programın çalışmasını istediğimiz döngü sayısı 1000, çaprazlama oranı yüzde 30 ve mutasyon oranı da yüzde 5 olarak belirlenmiştir.

Geliştirilen yazılıma verilen bir girdi dosyası için mutlaka sağlanması gereken kısıtlarımız şunlardır;

1-) Her ders için bir öğretim üyesi atanmış olmalı
2-) Her öğretim üyesi en az bir ders veriyor olmalı

Geliştirdiğimiz yazılımdaki çözüm uzayı elemanları aşağıdaki olmazsa olmaz kısıtları mutlaka sağlayacak şekilde tasarlanmıştır;

1-) Her bir ders için sadece bir derslik atanmış olmalı,
2-) Her bir ders için bir ders saati ayarlanmış olmalı,
3-) Bir derslikte herhangi bir zaman diliminde en fazla bir ders olmalı,
4-) Bir öğretim üyesi bir zaman diliminde en fazla bir ders vermeli,
5-) Çakışması istenmeyen dersler aynı zaman dilimine konmamalı.

Geliştirilen yazılım aynı derslikte aynı saatte birden fazla ders olmayacak şekilde öğretim üyelerinin derslerini vermek istedikleri zaman dilimleri için iki tercih ve derslerini vermek istemedikleri zaman dilimi için bir tercih ve olası sınıf mevcutlarını göz önüne alarak optimale yakın şekilde hangi derslikte hangi zaman diliminde hangi derslerin verileceğini belirlemektedir. Ayrıca derse kaydolması beklenen öğrenci sayısı-sınıf boyutu farkının minimum olması da hedeflenmektedir.

Bulunan çözümler, olması zorunlu kısıtları yerine getirmekle beraber, optimize etmek için kullandığımız girdilerinde birçoğunu sağlayan çözümler üretmektedir.

4. GELİŞTİRİLEN YAZILIM

Geliştirilen programla, çözüm uzayının büyüklüğüne rağmen iyi bir çözüme kısa zamanda ulaşılmaktadır. Programda üç temel işlem meydana gelmektedir: (1) Seçim, (2) Çaprazlama ve (3) Mutasyon.
Bu işlemlerde nelerin yapıldığı kısaca şu şekilde açıklanabilir;

1) Seçim
Programda her çözüm bir dizi (birey) olarak kodlanmakta ve iki matris olarak tanımlanmaktadır. Bu matrislerden ilki her bir ders ile o dersin işleneceği zamanın eşleşmesinden, ikincisi ise her bir ders ile o dersin işleneceği sınıfın eşleşmesinden oluşmaktadır. Bu bireyler değerlendirme aşamasında deşifre edilerek belirli amaç fonksiyonu ya da fonksiyonlarında gösterdikleri performanslarına (uygunluklarına) göre değerlendirilmektedir. Bu değerlendirme işlemi bazı parametreler kullanılarak yapılmaktadır.

Geliştirilen yazılımda kullanılan genetik algoritma için uygunluk derecelerinin belirlenmesinde aşağıda verilen dört faktör göz önünde bulundurulmuştur:

a. Öğretim elemanlarının önceden tercih ettikleri zaman dilimlerine derslerin atanması,
b. Toplam olarak kaç tane sınıfta sınıf boyutunun beklenen öğrenci sayısından fazla olduğu ve öğrenci sayısının fazla olduğu durumda bu fazlalığın miktarı
c. Çakışması istenmeyen kaç dersin çakıştığı,
ç. Öğretim üyelerinin birbirini takip eden zaman dilimlerinde ders verip vermedikleri (Bu durumda öğretim elemanlarının performanslarının düşeceği öngörülmüştür).

Mutasyon ve çaprazlama işlemleri için değerlendirme neticesinde yüksek uygunluktaki bireylerin seçilme olasılıkları daha fazladır.

2) Mutasyon
Seçilen bireyler mutasyona girmeye hak kazanırlar. Seçilen bireyi mutasyona uğratmak için elemanları ikilik sistemdeki sayılardan oluşan ders-zaman matrisinin rasgele iki satırı yer değiştirilmekte ve aynı işlem ikinci eşleme matrisi olan ders-sınıf matrisi içinde yapılmaktadır. Mutasyon sonucunda oluşan yeni bireyin performansı hesaplanmaktadır.

Bu hesaplama sonucunda oluşan yeni bireylerin uygunluk değerleri bu bireyleri oluşturan bireylerin uygunluk değerlerinden büyükse yeni bireyler çözüm için kabul edilmekte, küçükse kabul edilmemektedir.

3) Çaprazlama
Çaprazlama işlemi için ele alınan iki bireyin ders-zaman eşlemesi matrislerinin satırları yer değiştirilmektedir. Aynı şekilde o bireylerin ders-sınıf matrislerinin satırları da yer değiştirilerek toplam dört adet yeni birey elde edilmektedir. Elde edilen bu bireylerin uygunlukları hesaplanmakta ve bu bireylerden iki tanesi seçilmektedir.

Algoritmada çaprazlama ve mutasyon işlemlerinin sonucunda oluşan yeni bireylerin ilk olarak olmazsa olmaz koşulları sağlayıp sağlamadıklarına bakılmakta eğer bu koşullar sağlanıyorsa uygunluk hesabı yapılmaktadır.

Geliştirilen yazılıma verilecek girdi dosyasının ilk satırında sırasıyla ders sayısı, öğretim üyesi sayısı, birbiriyle çakışmayan zaman dilimi sayısı, derslik sayısı verilmektedir. Ders sayısınca satırda; o dersin kodu, kaydolması beklenen öğrenci sayısı, hangi öğretim üyesinin o dersi vereceği, öğretim üyesinin dersini vermek için tercih ettiği ilk zaman dilimi, ikinci zaman dilimi ve tercih etmediği zaman dilimi girilmelidir. Sonraki satırdan itibaren derslik sayısınca satırda; sırasıyla dersliğin kodu ile dersliğin kapasitesi girilmelidir.

Örnek olarak bir girdi dosyası Tablo-1 de verilmiştir.

Programın çalışması sonucunda Tablo-2 de verilen çıktı dosyası elde edilmiştir.

Tablo-2'de bir sınıfta bir zaman diliminde sırasıyla hangi dersin hangi öğretim üyesi tarafından verildiği, öğretim üyesinin zaman tercihinin öngörülen zaman dilimi ile ne kadar uygunluk gösterdiği (1: 1. tercih; 2: 2. tercih; -1: istenmeyen zaman; 0:diğer), çakışması istenmeyen derslerin çakışıp çakışmadığı (ç: çakışma var), sınıfların kapasitesini aşıp aşmadığı (k: kapasite aşılmış) yazılım tarafından belirtilmektedir.

Tablo-1. Örnek Girdi Dosyası

```
20 6 6 4
1 100 2 3 2 1
2 80 1 2 3 1
3 60 3 1 2 3
4 80 2 2 1 3
5 80 1 1 2 3
6 60 3 2 3 1
7 100 1 1 2 3
8 100 2 2 3 1
9 80 4 3 6 2
10 80 6 4 2 6
11 100 5 4 3 2
12 80 4 3 4 5
13 80 6 4 2 1
14 100 3 3 2 4
15 100 4 3 4 2
16 80 5 2 4 5
17 80 2 4 2 3
18 100 6 6 4 5
19 100 1 2 3 1
20 80 5 4 3 2
1 80
2 100
3 100
4 80
```

Tablo-2. Örnek girdi dosyasına karşılık yazılımın oluşturduğu ders programı

	Period1	Period2	Period3	Period4	Period5	Period6
Sınıf1	17,2(0, ,)	16,5(1, ,)	Boş	Boş	10,6(0, ,)	6,3(0, ,)
Sınıf2	Boş	2,1(1, ,)	11,5(2, ,)	Boş	5,1(0, ,)	4,2(0, ,)
Sınıf3	7,1(1, ,)	1,2(2, ,)	18,6(0, ,)	15,4(2, ,)	8,2(0, ,)	19,1(0, ,)
Sınıf4	9,4(0, ,)	13,6(2, ,)	14,3(1, ,k)	3,3(0, ,)	12,4(-1, ,)	20,5(0, ,)

5. SONUÇ

Genetik algoritmalar çözüm uzayının çok büyük olduğu problemlerde hızlı bir şekilde kabul edilebilir bir sonuç bulmak için kullanılmaktadır. Geliştirilen yazılımla, doğrusal programlama teknikleri genetik algoritmanın çabuk çözüme varma gücü ile birleştirilerek; ders programı hazırlamak gibi günümüz bilgisayarları tarafından en iyi çözümü bulmanın haftalar alabileceği bir problemi, dakikalarla ölçülen kısa bir zaman dilimi içinde bulunabilmektedir. 100 profesör, 300 farklı ders, 15 çakışmayan zaman dilimi, 20 derslikten oluşan orta büyüklükte örnek bir problemi, program Pentium 4 1600 Mhz işlemci altında çalışan 256 MB belleğe sahip bir bilgisayar tarafından 1 dakikadan kısa zamanda, kabul edilebilir bir sonuca ulaşabilmektedir.

Geliştirilen otomasyon yazılımında sınıfların boyutunu, ders sayılarını, derse kaydolması beklenen öğrenci sayılarını ve dersi verecek olan öğretim üyeleri girdi olarak alınmakta ve verilen sınırlamalar göz önünde bulundurularak dönemlik ders programı hazırlanabilmektedir.

Geliştirilen yazılım öğretim üyelerinin derslerini vermek için tercih edecekleri zamanları da girdi olarak alabilmektedir. Geliştirilen yazılım parametreleri ve sınırlamaları değiştirmeye imkan sağlayan esnek bir yapıya sahiptir.

KAYNAKLAR

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1997.

[2] Reeves, Colin R. and Jonathan E. Rowe, "Genetic algorithms : principles and perspectives : a guide to GA theory," Kluwer Academic, Boston, 2003.

[3] D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms: Part 1, fundamentals," University Computing, vol. 15, pp. 58-69, Feb. 1993. AI 2001, Las Vegas, Nevada, 2001.

[4] Öztemel E. *Yapay Sinir Ağları* Papatya Yayıncılık 2003.

[5] Areibi S., Moussa M., Abdullah, H., A "Comparison of Genetic/Memetic Algorithms and Other Heuristic Search Techniques," 2001 International Conference on Artificial Intelligence IC-

[6] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Mass, 1989.

[7] Ceylan, H, Öztürk, H. K. Ve Karahan, H. (2003) "Türkiye Enerji Talebinin Genetik Algoritma Yaklaşımı ile Modellenmesi" I. Ege Enerji Sempozyumu ve Sergisi, Pamukkale Üniversitesi, Mühendislik Fakültesi, Denizli, Mayıs, pp. 266-37