

GENETİK ALGORİTMA İLE KAYNAK KISITLI PROJE ÇİZELGELEME

Öğr. Gör. Dr. Semin PAKSOY
Çukurova Üniversitesi
İktisadi ve İdari Bilimler Fakültesi
Ekonometri Bölümü
spaksoy@cu.edu.tr

Yrd. Doç. Dr. Arzu UZUN
Çukurova Üniversitesi
İktisadi ve İdari Bilimler Fakültesi
İşletme Bölümü
garzu@cu.edu.tr

ÖZET

Bu çalışmada kaynak kısıtlı proje çizelgeleme problemlerinin genetik algoritma yaklaşımı ile çözümü ele alınmıştır. Başlangıçta genetik algoritmanın temel kavramlarına yer verilerek, proje çizelgeleme şemaları ve çizelgelemede göz önünde bulundurulması gereken unsurlar özetlenmiştir. Daha sonra, Delphi 6.0 da geliştirilen genetik algoritmanın, otuz faaliyetli ve dört kaynak kullanan standart test problemlerindeki sonuçlarına yer verilmiştir. Geliştirilen algoritmanın, iterasyon sayısı, çaprazlama oranı ve öncelik kuralları açısından davranışları test edilmeye çalışılmıştır. Son olarak, çizelgeleme problemlerinin faaliyet sayıları, basit ya da karmaşık olma özelliklerinin algoritma üzerinde etkileri araştırılmaya çalışılmıştır. Çalışma sonucunda, geliştirilen genetik algoritma ile elde edilen çözümlerin, genel olarak optimuma yakın çözümler olduğu görülmektedir.

Anahtar Kelimeler: Genetik algoritma, Proje çizelgeleme, Çizelge oluşturma şemaları, Öncelik kuralları

ABSTRACT

In this paper, we consider genetic algorithm approach for the resource constrained project scheduling problems. Starting with the summarizing the basic components of genetic algorithm approaches, we describe the generating project schemes and its necessities. Subsequently, we present the results of our computational study which is depend on the standart set of test instances. These instances have only four reources and thirty activities. The results are obtained from a computer program that is developed by Delphi 6.0. Moreover, the behaviour of the developed genetic algorithm is analyzed with respect to its main components such as iteration number, crossover rate and priority rules. Finally the influence of the problem characteristics, like that simplicity, complexity and size, is examined on the performance of the genetic algorithm. At the end of the study, the results show that the solutions produced from the developed genetic algorithm is generally near the optimum solution.

Keywords: Genetic algorithm, Project scheduling, Schedule generation schemes, Priority rules

1. Giriş

Günümüzde karmaşık problemlerin çözümünde geleneksel yöntemlerin yetersiz kalması, yeni çözüm teknikleri arayışlarını gündeme getirmiştir. Evrimsel hesaplama teknikleri, bu arayışların sonucu olarak ortaya çıkan tekniklerdir ve 19. yüzyılda Charles Darwin tarafından ileri sürülen evrim teorisinden esinlenerek geliştirilmiştir.

Evrimsel hesaplama teknikleri, doğadaki evrimsel süreci temel alan bilgisayara dayalı problem çözme teknikleridir. Bilgisayarların doğuşundan günümüze değin yapılan çalışmalarda, insan ve davranışlarının en önemli model olarak benimsendiği görülmektedir. İnsanın davranış ve tutumuna benzetim yapılarak geliştirilen programlardan sonra, insanın geçirdiği evrimsel sürecin problemlerin çözümünde kullanılması gündeme gelmiştir. Geleneksel programlama teknikleriyle çözülmesi güç olan, özellikle sınıflandırma, çizelgeleme ve çok boyutlu optimizasyon problemleri üzerinde yapılan çalışmalar, probleme ait uygun çözümlerin evrimsel hesaplama teknikleri ile daha kolay ve hızlı olarak elde edilebildiğini göstermektedir. Ancak karmaşık problemlerin optimum çözümlerine yönelik yapılan araştırmalar halen devam etmektedir.

Evrimsel kavramsal olarak kontrollü bir süreç değildir. Farklı genetik bilgilere sahip canlılar, buldukları ortamda, yaşamak için yarışır. Güçlü ve uyumlu olanlar hayatta kalıp çoğalarak (üreyerek) genetik bilgilerini bir sonraki kuşağa geçirme şansını yakalarlar. Evrimin devamını sağlayan en önemli unsur üreme sonucu oluşan farklılıktır.

Evrimsel farklılık, üreme sırasında gerçekleşen yeniden birleşimler (recombination) ve mutasyonlar sonucunda oluşmaktadır. İki canlının genetik olarak farklı, ama işlevsel olarak eş kromozomları karşılıklı olarak yer değiştirmek suretiyle bilgi değişimi olmaktadır. Üreme sonucunda oluşan farklı canlıların bir bölümü, önceki kuşağa göre değişen çevreye daha iyi uyum gösterip daha güçlü olabilmektedir. Bunların yaşaması ve üreyerek soyunu devam ettirmesine doğal seleksiyon denilmektedir.

Evrimsel hesaplama tekniği kullanılan algoritmaların temelini biyolojik evrimsel süreç oluşturmaktadır. Evrimsel algoritmalar, bulunduğu çevrede en fazla güçlü olanları seçme, yeniden birleşme, üreme ve mutasyon gibi evrimsel süreçlerden yararlanarak bir sistem oluşturmaktadır.

2. Genetik Algoritmalar

En yaygın kullanılan evrimsel algoritma, genetik algoritmadır (Mitchell ve Taylor,1999,s.594). Genetik algoritma, evrim sürecini kullanarak bir bilgisayara anlayamadığı çözüm yöntemlerinin öğretilbileceği düşüncesi ile ortaya çıkmıştır.

Evrimsel hesaplama psikolojisinden etkilenen John Holland, genetik algoritmayı bilgisayar ortamında uygulayan ilk araştırmacı olmuştur; 1975 yılında yayınladığı “Doğal ve Yapay Sistemlerde Uyumlaştırma” isimli kitabında yapay problemlerin çözümünde evrimsel süreçlerin uygulanabileceğini açıklamıştır (Koza,1995,s.287). Evrimsel hesaplama tekniğinde kullanılan problemler, çözüm için oldukça fazla olasılık ya da kurallar dizisi içeren problemlerdir.

Evrimsel algoritmaların amacı, salt rastgele çözümler aramak değildir. Asıl amacı, olası hata ve belirsizlik payını da dahil edecek süreçler kullanarak rastgele sonuçlardan çok daha iyi çözümler üretmektir.

Darwin’in evrim teorisinden etkilenen genetik algoritma ile bir problemi çözmek, problemi sanal evrimden geçirmek anlamına gelmektedir.

Genetik algoritma, karmaşık problemlerin çözüm alternatifleri arasından optimum çözümü bulmak için kullanılan bir arama yöntemidir (Mori ve Tseng,1997,s.135).

Genetik algoritmalar stokastik yöntemlerden ya da programlardan oluştuğu için genellikle optimum (global) çözüme yakın çözümler üretmektedirler (Brochmann,2005,s.1). Genetik algoritma ile üretilen çözüm, global olmayan alternatif çözüm ise bu çözümün optimuma yakın (lokal) çözüm olduğu söylenebilmektedir (Yeo ve Agyei,1998,s.268).

Genetik algoritmalar, geleneksel yöntemlerle çözümü zor veya hemen hemen imkansız olan problemlerin çözümünde kullanılan sayısal optimizasyon yöntemlerinden biridir. Dolayısı ile karmaşık, kısıt sayısı fazla, amaç fonksiyonu kurulamayan ve kesin çözüm yöntemi olmayan optimizasyon problemlerinin çözümünde kullanılabilir.

2.1. Genetik Algoritmanın Özellikleri

Genetik algoritma, doğal seleksiyon ve doğal genetik aşamalara benzetim yapılmış stokastik bir arama metodudur. Metot, kromozom adı verilen ve rastgele çözüm kümelerinden oluşan bir toplum ile başlar.

Başlangıç toplumu, sonraki üremelerde elde edilen kromozomlar ile sürekli değişime (mutasyona) uğratarak yeni toplumlar oluşturulmaktadır. Her bir toplum, değişikliğe uğratılmış yeni çözüm kümeleridir.

Genetik algoritmalar, algoritmanın başlangıcında oluşturulan çözüm kümesinden başlayarak, sonraki aşamalarda bu çözümlere rastgele değişim uygulayarak oluşturulan yeni çözüm kümeleri içerisinde amaç fonksiyonunu en iyileyen çözümlere erişmeye çalışmaktadır. Başlangıç toplumu hariç tüm toplumlar yani farklı çözüm kümeleri, uzun hesaplama işlemlerine ihtiyaç duyulmadan evrimsel operatörle hızlı bir şekilde hesaplanmaktadır.

Algoritma, her bir toplum içerisindeki iyi çözümleri aradığından, diğer optimizasyon yazılımlarından yararlanamayacak büyüklük ve karmaşıklık içeren problemlerde rahatlıkla kullanılabilir.

Algoritmanın bu özelliği, başlangıç toplumu kromozomlarının oluşturulmasının gerekliliğini vurgulamaktadır. Özellikle karmaşık problemlerde hesaplanması güç olan ve her bir elemanı problemin uygun (feasible) çözümünden birini temsil eden toplum, genetik algoritmalarda bir kereye mahsus olmak üzere oluşturulması gibi bir güçlüğü sahiptir. Uygun çözümlerin hesaplanmasında, her hangi bir yöntem kullanılabilir. Ancak dikkat edilmesi gereken, başlangıç toplumundan sonraki toplumlara müdahale edilmeden, salt genetik işlemlerle sürecin tamamlanmasıdır.

2.2. Temel Kavramlar

Bir problemi genetik algoritma ile çözümlenebilmek için, öncelikle problemin algoritma mantığına uyarlanması gerekmektedir. Bu nedenle genetik algoritmanın temel kavramları, geliştirilecek algoritmanın başarısı açısından daha da önem kazanmaktadır.

Kromozom: Genetik algoritma bir başlangıç toplumu ile başlamaktadır. Toplum, kromozom (birey, çözüm) olarak adlandırılan elemanlardan oluşmaktadır. Her kromozom, temsil ettiği çözüme ait bilgileri içermektedir ve bilgiler bir dizi halinde kodlanmaktadır.

Bilgilerin anlam bütünlüğü açısından, kodlamanın önceden belirlenen bir sisteme göre yapılması gerekmektedir. Kromozom kodlamasının, genetik algoritmaların

ilk uygulamalarında, ikili sayı sistemine göre yapıldığı gözlemlenmektedir. İkili sayı sisteminde hazırlanan bir kromozomda, her bir bit/bit kombinasyonu çözümün bir karakteristiğini temsil etmekte (Chan vd.,2005,s.345) ve bilgisayar tarafından daha rahat ve hızlı bir şekilde işleme tabii tutulmaktadır (Kurt ve Semetay,2001,s.3).

Kromozomların kodlanması, genellikle ikili sayı sistemi temel alınarak kodlansa da, tamsayı ve reel sayılar da kullanılmaktadır (Mitchell,1999,s.67). Özellikle, karar değişkeni sayısı fazla ve değişkenin ikili sistemdeki karşılığı uzun olan problemlerde onlu sistemde yapılan kodlama tercih edilmektedir.

Gen: Bir canlının (bireyin) kalıtsal özelliklerinden herhangi birini taşıyan parçaya gen adı verilmektedir. Bu nedenle kalıtımın fiziksel ve işlevsel birimi olmaktadır. Her gen özel bir işlev taşıyan kromozomların belli konumunda bulunan dizilerden oluşmaktadır.

Kromozom üzerinde, belli bir konumda bulunan genler, temsil edilen değişkenin değerini ifade etmektedir (Koza,1995,s.287). Her bir karar değişkeninin sayısal değeri, bir geni temsil eder. Bir problemde kaç tane karar değişkeni varsa o kadar da gen vardır.

Genlerin bir dizi halinde sıralanması ile ortaya çıkan genler dizisine “kromozom” adı verilmektedir. Böylece bir kromozomda, problemdeki karar değişkenlerinin her birinin bir arada bir dizi halinde bulunduğu anlaşılmaktadır. Algoritmanın başlangıcında, kromozomlara yerleştirilmesi karar verilen genlerin sırasının genetik işlemler esnasında asla değiştirilmemesi gerekmektedir.

Toplum Büyüklüğü: Toplum büyüklüğü başka bir deyişle kromozom (birey) sayısı genetik algoritmanın başarısını ya da optimum sonuca erişim süresini etkileyen önemli unsurlardan biridir. Toplum büyüklüğünün ne olacağını karar vermek, tüm genetik algoritma kullanıcıları açısından oldukça önemli bir aşamadır.

Toplum büyüklüğünün gereğinden küçük alınması, algoritmanın dolayısı ile arama tekniğinin dar bir çözüm uzayında uygulanması anlamını taşımakta ve caprazlama için seçenek sayısını daraltmaktadır.

Gereğinden büyük alınması ise uygulamanın anlamlı sonuçlara geç ulaşmasına veya yavaş çalışmasına neden olacaktır (Goldberg,1992, s.20; Kurt ve Semetay,2001,s.5). Büyük toplumlarda çözüm uzayı daha iyi örneklenebileceğinden arama daha etkin yapılırken, arama süresi artacaktır.

Uyumluluk: Mevcut toplumda bulunan iyi özelliklere sahip (uyumlu) kromozomların bir sonraki aşama için kullanılacak yeni topluma (kuşağa) aktarılması, belirlenen kriterler dâhilinde değerlendirilerek yapılmaktadır. Kromozomların uyumluluk değerlendirmesi ya da kromozom kalitesi, bir uygunluk fonksiyonu değeri ile belirlenmektedir.

Uygunluk fonksiyonları en iyi olan kromozomlar ise optimum ya da optimuma yakın çözümü temsil eden kromozomlardır (Mori ve Tseng,1997,s.137).

Uyumluların Seçilmesi: Seçme işlemi, “yetenekli olanın yaşaması” ilkesine uygun olarak çalışmaktadır. Başlangıç toplumundaki kromozomlar arasından, en uyumlu olanlardan bazıları yeni topluma aktarılmaktadır. Yeni toplumun diğer kromozomları

da, genetik işlemler ile oluşturulmaktadır. Uygunluk değeri yüksek olanların seçilme ve yeni topluma aktarılma şansı daha yüksektir.

Böylece bir yandan yüksek uyumluluğa sahip bireyin bir sonraki kuşağa aktarılamama olasılığı ortadan kaldırılırken, diğer yandan da yeni toplumun maksimum veya ortalama uygunluk değerinde beklenen artışın optimum çözümün uygunluk değerine de yansımaları söz konusu olacaktır.

Uygun olarak seçilen bir kromozom, yeni toplumun bir veya daha fazla kromozom oluşumuna katkıda bulunabilmektedir. Böylece toplumdan topluma kalıtsal özellikler aktarılabilecektir.

Uyumlu kromozomların seçilmesi amacıyla geliştirilmiş birçok yöntem bulunmaktadır. Ancak elitist strateji, rulet çarkı ve turnuva seçim yöntemleri en yaygın kullanılanlardır. Bunların dışında da bazı pratik yöntemler kullanılabilmektedir.

Elitist stratejide uygunluk değeri en iyi olan birey, en kötü birey ile yer değiştirerek yeni toplumun daha iyi bireylerden oluşması sağlanır. Böylece bir yandan en yüksek uygunluk değerine sahip bireyin bir sonraki kuşağa aktarılamama olasılığı ortadan kaldırılmaktadır.

Rulet çarkı yönteminde, toplumdaki tüm bireylerin uygunluk değerleri hesaplanır. Bireylerin uygunluk değerleri toplanarak toplumun uygunluk değeri elde edilir. Her bireyin uygunluk değerleri, toplumun uygunluk değerine bölünerek, bireyin seçilme olasılıkları elde edilir.

Bireyler, seçilme olasılıkları ölçüsünde rulet çarkında yer alırlar. Böylece rulet çarkı üzerinde yüksek olasılık değeri ile temsil edilen bireylerin seçilme olasılıkları artmaktadır.

Ebeveynler, rulet çarkı üzerinden farklı yöntemlerle seçilebilmektedir. Seçim, rastgele (Buckles ve Petry,1992,s.2), en yüksek olasılık değerine göre ya da eklemeli oranlardan yararlanılarak yapılabilmektedir. Eklemeli oranlar, seçilme olasılıklarının toplanması ile elde edilmektedir. Uygulamada, rastgele bir sayıya erişinceye kadar eklemeli oranlar elde edilir ve son eklenen sayının ait olduğu çözüm seçilmek suretiyle de işlem tamamlanabilir.

Turnuva seçim yöntemi, kolaylığı ve uygunluğu açısından en yaygın olanıdır. Turnuva seçim yönteminde, önce toplumu oluşturan kromozomlardan bazıları, belli kriterlere göre seçilmektedir. Daha sonra, bu kromozomlar kıyaslanarak aralarında uygunluk değeri yüksek olan bir kromozom yeni topluma aktarılacak üzere seçilmektedir. Böylece oluşturulan yeni toplum, bir önceki toplumun kötü bireylerinden arındırılmaya çalışılmaktadır. Optimum çözüme hızlı erişim sağlanması açısından önemli bir yöntemdir.

Turnuva seçim yönteminde, yerine koyarak ya da koymayarak rastgele seçilen t adet bireyden oluşturulan gruba turnuva genişliği denir (Bolat vd.,2004,s.267). Bu grupta bulunan en iyi birey yeni topluma aktarılır. Bu işlem kullanıcının önceden belirlediği işlem sayısı kadar tekrarlanır.

Turnuva seçim yönteminde uyumluların seçilmesi (yeteneklilerin yaşaması) ve uyumsuzların yaşamaması ilkesi temel alınmıştır ve değişik şekillerde seçimler söz konusudur. Tablo 1’ de verilen örnekte de görüldüğü gibi, sıra numarası 6 olan bireyin maliyeti en fazla olduğundan en uyumsuz bireydir. Sıra no’ su 3 olan birey ise en uyumlu olanıdır. 6 numaralı kromozom sadece kendisi ile kıyaslandığı durumda yaşama şansına sahip olacaktır. Benzer şekilde, 3 numaralı birey sadece kendisi ile

kıyaslansaydı, sağ kalanlar arasından tekrar seçilemeyecekti. Buna rağmen 3 numaralı birey, turnuva seçim yöntemine göre yaşamını garantilemektedir. İkinci en uyumlu birey 1 numaralı kromozomdur. 1 numaralı birey, 3 numaralı ile kıyaslandığında daha az uyumlu olduğundan seçilemeyecektir. 2 no' lu birey oldukça uyumsuz olmasına rağmen, kendisinden daha uyumsuz olan 6 no' lu bireyle kıyaslandığından yaşama şansına sahip olmaktadır.

Tablo 1: Turnuva Seçim Yöntemine Göre Uyumlu Kromozomların Seçilmesi

Sıra No	Rastgele Sayılar	Uygunluk Değeri (f_i - Maliyet)	Seçilenler
1	6	10222 \$	1
2	3	13667 \$	3
3	1	9886 \$	3
4	8	12443 \$	4
5	7	11563 \$	7
6	2	14788 \$	2
7	4	10587 \$	7
8	5	13213 \$	5

Kaynak: Yeo ve Agyei,1998,s.271

Yukarıda bahsedilen ve genetik algoritma uygulamalarında yaygın olarak kullanılan seçim yöntemlerinden hiçbiri, henüz literatürde en iyi seçim stratejisi olarak belirlenememiştir. Bu nedenle, son yıllarda yapılan çalışmalarda bazı araştırmacıların bu seçim stratejilerinden bir karma oluşturdukları gözlenmektedir. Örneğin, çözümü zor optimizasyon problemlerinde, rulet çarkı ve elitist stratejinin birlikte uygulandığı bir genetik algoritma çalışması gerçekleştirilmiştir (Nakamura vd.,2005,s.924).

Bir sonraki toplumun daha da iyileştirilmesi için, topluma ait bireylerin kendi aralarında eşlenerek genetik algoritma işlemlerinin rastgele uygulanması gerekmektedir. Bu aşamada uygulanabilecek değişik seçim yöntemlerinin kullanılması da mümkündür. Ancak burada dikkat edilecek unsur, kullanılacak seçim yönteminin genetik algoritma mantığına uygun olarak seçilmesi ve uygulanmasıdır. Bu yöntemlerden bazıları aşağıdadır:

- N_{iyi} adet uygunluk değeri yüksek kromozomlar içerisinde sırası ile en iyisini seçmek ve diğer eşi de $N_{kötü}$ adet uygunluk değeri düşük kromozomlar arasından rastgele seçimle eşleşme yapmak (Mori ve Tseng,1997,s.138).
- N_{iyi} adet iyi kromozomları olduğu gibi bırakmak ve $N_{kötü}$ adet kötü kromozomlar arasından rastgele seçimle eşleşme yapmak.
- Böyle bir ayırma gitmeden kromozomlar arasında sıralı eşleşme yapmaktır.
- Burada sıralanmayan ve uygulamaya yönelik geliştirilmiş olan, daha farklı sezgisel yöntemler kullanılabilir.

2.3. Genetik İşlemler

Bir sonraki toplumu iyileştirmek amacıyla yapılacak genetik işlemler, toplumdaki tüm bireylere ya da içlerinden seçilen bazı bireylere uygulanabilmektedir.

Toplumdaki en iyi çözümü temsil eden bireyleri yaşatmak ya da yeni oluşturulacak kuşağa aktarmak amacıyla, en uyumlu olan bireylerden bazıları bir sonraki topluma doğrudan aktarılabilir. Ebeveyn olarak seçilen kromozomlara ise genetik işlemler uygulanarak daha farklı çözümlere erişilmeye çalışılmaktadır.

Toplum büyüklüğü sabit tutulan genetik algoritmalarda, genetik işlemlerden yararlanılarak oluşturulacak birey sayısı, k , bir sonraki topluma doğrudan aktarılan en uyumlu birey sayısı ;

$k = \text{Toplum büyüklüğü}(n) - \text{Aktarılan uyumlu birey sayısı}$
kadarır.

Genetik işlemler, genetik işlemlerde çok önemli yeri olan genetik parametrelerden çok fazla etkilenmektedir. Bu parametreler toplum büyüklüğü, çaprazlama oranı ve mutasyon oranı şeklinde sayılabilir.

Uygun genetik parametreleri seçilerek yapılan genetik işlemler sonucunda, global sonuca ulaşma olasılığı artmaktadır. Ancak algoritma, genetik işlemlerin rastgele ya da sezgisel özellikleri nedeniyle uygun olan parametre değerleri ile çalıştırıldığında da, global sonuca erişemeyebilmektedir. Birçok problemin çözümünde iyi sonuçlar üretebilen genetik algoritma, üç temel genetik işlemin uygulanması ile sonuçlandırılmaktadır. Bunlar üreme, çaprazlama ve mutasyon işlemleridir.

Üreme: Üreme, uygunluk kriterlerine uyan bir kromozoma ait özelliklerin yeni topluma aktarılmasını sağlayan bir işlemdir. Uygun olarak seçilen kromozom çifti, yeni toplumun bir veya daha fazla kromozomunun oluşumuna katkıda bulunabilmektedir. Ebeveyn olarak adlandırılan bu kromozom çiftinin özellikleri, üreme sonucunda yeni topluma aktarılmaya çalışılmaktadır.

Seçilen uyumlu kromozomlardan daha uyumlu bireyler elde etmek için çaprazlama ve mutasyon işlemlerinin uygulanması gerekmektedir.

Çaprazlama: Seçilen ebeveynlerden yeni çözümler elde etmek amacıyla çaprazlama işlemi yapılmaktadır. Çaprazlama işlemi; seçilen ebeveyn kromozomlarında, seçilen çaprazlama konumu temel alınarak, ebeveyn kromozomlarının çapraz bir şekilde yer değiştirmesi işlemidir. Böylece ebeveyn kromozomlarındaki gen/genler yeni bireylere(çocuklara) kalıtsallaştırılmakta, kromozomlar arasındaki genetik bilgi sürekli olarak değişmekte ve toplumun başarısı artmaktadır.

Mutasyon: Genetik algoritmada kullanılan mutasyon (değişim), doğal genetik mutasyon fikrinden ortaya çıkmıştır. Kromozomdaki bir ya da birden fazla genin rastgele değişimidir. Mutasyon, kromozom başkalaştırılması veya farklılaştırılması için kullanılan bir operatördür.

3. Genetik Algoritma ve Proje Çizelgeleme

Kaynak kısıtlı proje çizelgeleme problemleri, kısıtlı kaynak kullanılarak bir projeyi oluşturan tüm faaliyetlerin, öncüllük ilişkilerini dikkate alarak amaç fonksiyonunu en iyileyecek şekilde çizelgelenmesidir (Ulusoy,2006,s.4). n faaliyet sayısına sahip bir kaynak kısıtlı proje çizelgeleme problemlerinde tüm faaliyetler iki türlü kısıta sahiptir. Birincisi öncüllük ilişkilerinden kaynaklanan kısıttır. Bu kısıt, her bir j faaliyetinin kendinden önce gelen faaliyetleri tamamlamadan başlatılamaması

zorunluluğunu getirmektedir. İkincisi ise, kaynak kısıtıdır. Projedeki faaliyetler sınırlı kapasiteye sahip kaynak kullanmaktadır.

Klasik kaynak kısıtlı proje çizelgelerde öncelik kısıtına sahip olan faaliyetler başlatıldıktan sonra kesintiye uğratılmazlar (Mori ve Tseng,1997,s.134). Proje seriminde yer alan başlangıç ve bitiş faaliyetleri kukla faaliyetlerdir; kaynak kullanımına yönelik talepleri olmayan ve faaliyet süreleri sıfır olan faaliyetlerdir. Proje çizelgeleme problemlerinin tanımında yer alan varsayımları;

- faaliyet sürelerinin deterministik olması,
- faaliyetlerin birim zaman kaynak kullanımının sabit olması,
- bir faaliyete atanan kaynağın, faaliyet süresince o faaliyet tarafından kullanılması,
- bir faaliyetin kullandığı kaynak miktarının pozitif bir sayı ile gösterilmesi,
- başlatılan faaliyetlerin kesintiye uğramadan tamamlanması,
- projedeki tüm faaliyetlerin gerçekleştirilmesi

şeklinde sıralamak mümkündür (Artigues vd.,2003,250; Ulusoy,2006,s.4).

Özellikle kaynak kısıtlı proje çizelgeleme gibi problemlerde, faaliyet sayıları açısından herhangi bir kısıt ile karşılaşılması arzu edilmektedir. Karmaşık ve çok faaliyetli kaynak kısıtlı çizelgeleme problemlerde, araştırma alanının geniş olması, matematiksel ve geleneksel yöntemlerle başarılı sonuçların elde edilmesi güçleşmektedir.

Çizelgeleme problemlerinin çözümünü güçleştiren unsurlar problemin büyüklüğü, karmaşıklığı, karmaşık problemlerde uygun çözümlerin elde edilmesinde karşılaşılan güçlükler ve belirsizliklerin tamamen kaldırılamaması şeklinde sıralanabilir.

Problemin büyüklüğü, başka bir deyişle projedeki faaliyet sayıları, projede kullanılan kaynakların kapasitesi, sayısı ve tipleri karmaşıklığı etkileyen önemli unsurlardır. Projede kullanılan kaynakların, kaynak faktörü ve kaynak gücü, kaynak kısıtlı çizelgeleme problemlerinin çözümüne doğrudan etkisi olan özelliklerdir.

Projede pek çok faaliyet aynı kaynakları kullanıyorsa, kaynak faktörü ya da kaynak kullanım durumu oldukça yüksek düzeydedir sonucu çıkarılmaktadır (Hindi vd.,2002,s.516). Kaynakların bol ya da kıt olma durumunu belirlemek amacıyla kullanılan kaynak gücü yüksek ise, projenin tamamlanması açısından kaynakların gereken en üst kaynak talebini karşılayabilecek seviyede olduğu anlamını taşımaktadır (Kolisch vd.,1995,s.1696-1698). Yüksek kaynak faktörü ve/ya da düşük kaynak gücü, çizelgelenen problemin “zor” problem olduğunu gösterirken, düşük kaynak faktörü ve/ya da yüksek kaynak gücü, çizelgelenen problemin “zor olmayan” bir problem olduğunu göstermektedir (Kolisch,1996,s.331).

Problemden faaliyet sayısı arttıkça, faaliyetlere yönelik 3N olarak isimlendirilebilecek Ne (hangi faaliyet), Nereye (hangi kaynakta) ve Ne zaman (kaynağı meşgul tutma süresi) sorularına tutarlı cevap verilmesi güçleşmektedir (Wall,1996,s.16)

Çözümü zor problemlerde, olası çözümler kümesini hesaplamak zor ve zaman alıcıdır. Zor problemlerde, en iyi çözüm arayışları için tekrarlı hesaplama işlemlerinin yapılması güçleşirken, çoğu zamanda hesaplama güçlükleri nedeniyle çözüm arayışları tamamlanamamaktadır.

Tüm güçlükler rağmen en iyi çözümü hesaplanabilen karmaşık problemlerde, dönem içinde bulunan koşulların belirsizliği, faaliyetlere ve kısıtlarına yansımakta, değişen şartlar altında problemin tekrar çözülmesi gerekmektedir. Dolayısı ile deterministik yöntemlerle çözümü zor olan bu tür çizelgeleme problemlerinde, genetik algoritma gibi alternatif yöntemlerle çözüm arayışlarına yönelik ihtiyaç giderek artmaktadır.

4. Genetik Algoritma Uygulaması

Örnek problemde her bir faaliyetin gerçekleştirilebilmesi için, d_i – faaliyet süresine ve r_j – j. kaynaktan r birime ihtiyaç duyulmaktadır. Kullanılan yenilenebilir kaynaklar; faaliyetlerin kullanım süresi ile kısıtlı olup, faaliyetlerin bitiminde tüm kapasitesi ile serbest kalan kaynaklardır. Kullanılan kaynak sayısı 4 adettir. Geliştirilen algoritmada proje tamamlama süresini minimize etmek amaçlanmıştır.

4.1. Kromozom Temsili (Chromozom Representation)

Bu problemin GA ile çözülmesi için önce kromozomların nasıl kodlanacağına karar verilmelidir. Kromozomların faaliyet listesi şeklinde kodlanması kararı verilmiştir. Ancak faaliyet listesi şeklinde tanımlanan kromozomlara çaprazlama ve mutasyon işlemleri uygulandığında, faaliyet öncelik sıralarında bozulmaların olabileceği bilinmektedir. Bu nedenle, iki toplumlu genetik algoritma kullanılmaktadır.

Birinci toplumda, proje çizelgelerini temsil eden başlangıç toplumunun oluşturulması amacıyla, proje faaliyet sayısı kadar rastgele sayılar üretilmektedir. Toplumdaki bu rastgele sayılardan yararlanılarak, çizelgeleme probleminin olası çözümlerini yansıtan kromozomlar, seri çizelgeleme (seri SGS) kullanılarak oluşturulmaktadır. Böylece birinci toplumdaki her bir elemana karşılık bir kromozom oluşturularak ikinci başlangıç toplumu hazırlanmaktadır.

İkinci başlangıç toplumundaki kromozomlarda yer alan temel bilgiler faaliyet etiketleri ve süreleridir. Bu bilgilerden yararlanılarak proje tamamlama süreleri belirlenir. Dolayısı ile toplumdaki kromozomların her biri, projeye ait çözüm alternatifleri olduğundan, ebeveyn kromozomların seçiminde doğrudan doğruya kullanılabilir. Uyumlu kromozomların ebeveyn olarak seçiminde herhangi bir fonksiyon kullanılmayarak, doğrudan kromozomun temsil ettiği çözüm alternatifinin proje tamamlama süresi dikkate alınmaktadır. Proje tamamlama süresi en az olan kromozom, en uyumlu kromozom olarak belirlenmektedir.

4.2. Çizelge Oluşturma Şemaları (Schedule Generation Schemes-SGS)

Özellikle kapsamlı proje çizelgeleme problemlerinde, genetik algoritma ile olanaklı çizelge hazırlamak için, literatürde konu edilen iki farklı çizelgeleme türü mevcuttur. Bunlar seri çizelgeleme (seri SGS) ve paralel çizelgeleme (paralel SGS) olarak adlandırılmaktadırlar (Debels ve Vanhoucke,2005,s.5). Çizelge hazırlamak için, çizelge türlerinden biri ya da her ikisi birlikte kullanılmaktadır (Kolisch ve Hartmann,1999,s.149). Çizelgeleme, projenin başlangıç konumundan başlayarak, çizelgenin parçalarını adım adım genişletme yoluyla çizelge üretmektedir. Çizelge parçası ise $(n+2)$ faaliyetten oluşan projenin faaliyetler alt kümesidir.

Konu edilen çizelge türleri, faaliyet ve zaman artışına göre birbirinden farklılık göstermektedirler (Hartmann ve Kolisch,2000,s.395). Seri çizelgeleme faaliyet artışına dayanırken, paralel çizelgeleme zaman artışına dayanır. Her iki çizelgeleme yöntemi de,

aday faaliyetler kümesinde bir faaliyetin seçilmesi aşamasında; seçilebilir faaliyetleri derecelendirme ve sıralama açısından, bir ya da daha fazla öncelik kuralı kullanılmaktadır.

Geliştirilen uygulamada, seri çizelgeleme yöntemi kullanılmaktadır. Seri çizelgelemeler, faaliyet adedi kadar aşamadan oluşurlar ve her bir aşamada, seçilebilecek tamamlama süresi ve kaynak kısıtına uyan faaliyetler kümesinden bir faaliyet seçilmektedir. Seçilen faaliyetin aynı zamanda kukla olmaması gerekmektedir (Kolisch ve Hartmann,1999,s.149). Bu nedenle, her bir aşamada seçilebilecek faaliyetler kümesi (karar kümesi), öncül faaliyetleri tamamlanan ve kaynak kısıtına uyan faaliyetlerden oluşturulmaktadır.

Seri çizelgeleme, algoritmanın her aşamasında güncelleştirilmesi gereken üç kümeyi temel alarak çalışmaktadır. Bunlar:

Karar kümesi (Aday faaliyetler kümesi): Bu küme, öncülleri bitmiş, seçilebilmeye aday olan faaliyetlerden oluşmaktadır.

Tamamlanmış faaliyetler kümesi: Daha önce seçilerek çizelgelenmiş faaliyetler kümesidir.

Kalan faaliyetler kümesi: Karar kümesi ve tamamlanmış faaliyetler kümesi dışında kalan faaliyetlerden oluşmaktadır.

Başlangıçta; çizelge, karar kümesi ve tamamlanmış faaliyetler kümesi boştur. Faaliyet sayısına eşit olan aşama boyunca her bir aşamada, karar kümesinden bir faaliyet öncelik kuralı uygulanarak seçilmekte ve çizelgeye (kısmi çizelgeye) ilave edilmektedir. Daha sonra karar kümesi, tamamlanmış faaliyetler kümesi ve kalan faaliyetler kümesi güncellenmektedir (Tormos ve Lova,2003,s.1073).

Çizelgeleme, çizelgelenecek faaliyet kalmayınca kadar devam etmektedir. Seri çizelgelemelerin her aşamasında seçilebilir aday faaliyet kümesi hazırlanması, yöntemde her zaman uygun ve etkin çizelge elde edilmesini mümkün kılmaktadır.

4.3. Öncelik Kuralları (Priority Rules)

Öncelik kuralları, kaynak kısıtlı proje çizelgeleme problemlerinde, kısıtları karşılayan birden fazla faaliyet içerisinden hangisinin seçilerek çizelgeleneceğini belirlemede yaygın olarak kullanılan kurallardır.

Genetik algoritmaların başlangıç uygulamalarında, sezgisel olmayan uygulamalarda da kullanılan öncelik kuralları, kullanım kolaylığı ve sezgisel olma özelliğinden dolayı önemini korumaktadır (Kolisch,1996a,s.322). Ancak öncelik kurallarından hangisinin kullanılacağı kararının, optimum amaç fonksiyonu değerinden ortalama sapma değeri açısından problemin çözümüne büyük bir üstünlük getirmediği de görülmektedir (Brucker vd.,1999,s.11).

Öncelik kurallarından hangisinin kullanılacağı kararı, projeyi çizelgeleme süresi ve algoritmanın yinelenme sayısı gibi önemli unsurları etkilemektedir. Çizelgedeki faaliyetlerin öncelikleri, öncelik kuralları kullanılarak belirlenmektedir. Geliştirilen algoritmada kullanılan öncelik kuralları (Cheng ve Gen,1994,s.740; Özdamar,1999,s.57):

1. En Küçük Bolluk Değeri (Minimum Total Slack - MINSLACK):

Bu kuralda amaç, en az bolluk süresine (slack time) sahip kritik faaliyeti belirlemektir.

$$P_j = LFT_j - EST_j - d_{jm}$$

Burada;

P_j : j faaliyetinin öncelik değeri

LFT_j : En geç bitirme zamanı

EST_j : En erken başlama zamanı

d_{jm} : j faaliyetinin m modundaki faaliyet süresi

dir.

MS Project 2000 programı, kaynak kısıtlı proje çizelgeleme problemlerinde sadece MINSLACK öncelik kuralını kullanmaktadır (Wang ve Lu,2002. s.1979).

2. En Kısa En Geç Bitirme Zamanı (Minimum LFT):

LFT kuralının amacı, proje bitirme zamanının artma riskini azaltacak, küçük LFT'lere sahip faaliyetleri çizelgelemektir.

$P_j=LFT_j$

Burada;

P_j : j faaliyetinin öncelik değeri

LFT_j : En geç bitirme zamanı

dir.

3. En Kısa İşlem Süresi (SPT):

SPT'nin amacı, en kısa faaliyet süreli faaliyeti göndermektir.

$P_j= d_{jm}$

Burada;

P_j : j faaliyetinin öncelik değeri

d_{jm} : j faaliyetinin m modundaki faaliyet süresi

dir.

4. Rastgele (Random):

Bu kural, seçilebilme koşullarını sağlayan faaliyetler arasından bir faaliyetin rastgele seçilmesini sağlamaktadır.

$P_j=RANDOM(ELIGSET)$

Burada;

P_j : j faaliyetinin öncelik değeri

ELIGSET: Seçilebilir faaliyetler kümesidir

RANDOM : Bu fonksiyon, aday çözümler içeren faaliyetler kümesinden bir faaliyeti rastgele seçmektedir.

5. En Kısa En Geç Başlama Zamanı (Minimum LST):

Bu kural da amaç, LST' leri erken olan faaliyetleri belirlemek ve böylece kritik faaliyetlerin ertelenmesinden doğacak proje gecikme sürelerini elimine etmektir.

$P_j=LFT_j - d_{jm}$

Burada;

P_j : j faaliyetinin öncelik değeri

LST_j : En geç başlama süresi

dir.

6. En Kısa En Erken Başlama Süresi (Minimum EST):

Bu kural da amaç, EST' leri erken olan faaliyetleri belirlemek ve böylece kritik faaliyetlerin ertelenmesinden doğacak proje gecikme sürelerini elimine etmektir.

$$P_j = EST_j$$

Burada;

P_j : j faaliyetinin öncelik değeri

EST_j : En erken başlama süresi

dir.

7. En Kısa En Erken Bitirme Süresi (Minimum EFT):

Bu kural da amaç, EFT' leri erken olan faaliyetleri belirlemektir.

$$P_j = EST_j + d_{jm}$$

Burada;

EST_j : En erken başlama zamanı

d_{jm} : j faaliyetinin m modundaki faaliyet süresi

dir.

8. En Büyük Kaynak Talebi (Greatest Resource Demand - GRD) :

Bu kuralda amaç; seçilebilir faaliyetler arasından, en fazla kaynak kullanacak faaliyete öncelik vermektir. Kaynak kısıtlı proje çizelgelerde kullanılmaktadır.

$$d_j \cdot \sum_{k=1}^K r_{jk} \cdot$$

şeklindeki formül, bir faaliyetin faaliyet süresi boyunca gerekli kaynak miktarını temsil etmektedir. Bu değeri, en fazla olan faaliyet seçilmektedir (Simpson ve Patterson,1996,s.532).

Bunlara ek olarak, literatürde çok kullanılan diğer kurallar ise (Boctor,1996,s.351);

MAX RWK: Faaliyetin süresi ile kendinden sonra gelen faaliyetlerin sürelerinin toplamı en büyük olan faaliyetlerin seçilmesi kuralıdır.

MAX NIS: Kendinden önce gelen faaliyet sayıları toplamı en büyük olan faaliyetin seçilmesi kuralıdır.

MAX PTM: En fazla faaliyet süresine sahip olan faaliyetin seçilmesi kuralıdır.

MIN PTM: En az faaliyet süresine sahip olan faaliyetin seçilmesi kuralıdır.

MAX CAN: Faaliyetin seçiminden sonra, seçilebilir faaliyet sayısı en fazla olan faaliyetin seçilmesi kuralıdır.

Öncelik kuralları; çizelgeleme aşamalarında değişik değer almaları (devingen-dinamik) ya da çizelgeleme boyunca sabit değer almalarına (durağan-statik) göre farklılık göstermektedirler. Örneğin SPT durağan bir kural iken, MAX CAN devingen bir kuraldır (Ulusoy,2006,s.9). Faaliyetlerin önceliklerini belirleme aşamasında kullanılan bu kuralların bazılarını deterministik bazılarını da olasılığa dayalı kurallar olarak da tanımlamak mümkündür.

Öncelik kurallarında kullanılan EST, LST, EFT, LFT ve Slack (bolluk) değerleri; kritik yol metodu ile hesaplanan değerlerdir (Simpson ve Patterson,1996,s.532). Bu nedenle de durağan öncelik kurallarıdır. Durağan öncelik kuralı, her bir kaynak kısıtlı proje çizelgeleme probleminde, bir kereye mahsus olarak hesaplandığından, algoritmanın işlem süresini artırmayan kurallardır. Ancak devingen

kurallar, her bir faaliyetin seçim aşamasında yeniden hesaplanarak dikkate alınacağından, algoritmanın işlem süresini etkilemektedirler.

Literatürde kaynak kısıtlı proje çizelgeleme problemlerinde, hangi kuralın daha iyi olduğu konusunda kesin bir yaklaşıma rastlanmamaktadır. Bu nedenle genetik algoritma uygulamalarında farklı tercihlerin yapıldığı görülmektedir. Bazı genetik algoritma uygulamaları, özellikle eski uygulamalar, tek bir öncelik kuralını kullanırken, bazı uygulamalarda birden fazlası rastgele bir şekilde seçilerek uygulanmaktadır.

Geliştirilen algoritmada, yukarıda açıklanan ve kullanılmasına karar verilen öncelik kuralları, çizelgeleme esnasında birinci toplumda oluşturulan rastgele sayıların kullanılması ile belirlenmektedir. Böylece algoritmanın her bir iterasyon ve toplumda oluşturacağı çizelge şemalarına etki edilmemektedir.

4.4. Algoritmanın Değerlendirilmesi

Geliştirilen algoritmanın üstünlük ya da eksikliklerinin saptanmasının objektif olarak yapılabilmesi için literatürde araştırmacılar değişik yöntemler ve test problemleri kullanmışlardır. Sezgisel yöntemlerle geliştirilen genetik algoritmaların değerlendirilmesinde söz konusu olan asıl unsur, algoritmanın genel olarak nasıl davranış gösterdiğiidir.

Literatür incelendiğinde, geliştirilen algoritmaların çözümü bilinen örneklerle test edildiğine sıkça rastlanmaktadır. Ancak, çalışmamız kapsamında da görülmüştür ki, bir problemde en iyi çözüm bulan bir algoritma, diğer problemlerde en iyi çözümü bulamayabilmektedir. Bu nedenle, algoritmanın performansının başarılı bir şekilde değerlendirilmesinde yeterli olamamaktadır.

Optimum çözümü bilinen karmaşık problemlerin çözümünü üreten tek ve kesin bir algoritma henüz mevcut olmadığından, yöntemin karmaşık problemlerde uygulanabilmesine yönelik çalışmalar devam etmektedir.

Geliştirilen algoritmanın başlangıç testlerinde, çözümü bilinen problemler üzerinde test edilmesi halen kullanılan bir yöntemdir. Ancak algoritmayı değerlendirmek açısından yeterli bir yöntem değildir. Yapılan çalışmalar; basit ve bilinen problemlerde optimum sonuç veren algoritmaların, karmaşık problemlerde optimum çözümden oldukça uzak değerler verebildiğini göstermektedir.

Sonuç olarak, algoritmanın testi aşamasında; öncelikle bilinen problemlerle test edilmesi, algoritmanın başarılı bir şekilde izlenmesi açısından bir gereklilik olmaktadır. Ancak yeterli olduğu söylenememektedir. Dolayısıyla genetik algoritmaların performans değerlendirilmesinde tek başına kullanılması yanıltıcı sonuçlar verebilecektir.

Uygulamada test amaçlı olarak kullanılan J30 veri kümesinin özelliği, 30 faaliyetli ve yenilenebilir 4 kaynak kullanan çizelgeleme problemlerini içermesidir. İnternet ortamında kullanıma sunulan ve Kiel Üniversitesi'nin PSPLIB'inden, <ftp://ftp.bwl.uni-kiel.de/pub/operations-research/psplib> adresinden alınan bu problemlerin yine aynı ortamda optimum proje tamamlama süreleri de mevcuttur (Kolisch, ve Sprecher,1997,ss.207–210). Dolayısıyla bu kümede bulunan toplam 480 adet örnek çizelgeleme problemleri kullanılarak geliştirilen algoritmanın optimal çözümden sapmaları analiz edilmektedir. Veri kümesindeki her bir proje çizelgeleme probleminin optimal çözüm değerleri ve geliştirilen algoritma ile 500 ve 1000 iterasyon sonucunda elde edilen değerler karşılaştırılmaktadır.

J30 veri kümesinde bulunan problemlere uygulanan testler sonucunda geliştirilen genetik algoritma ile aşağıda belirtilen sonuçlara ulaşılmaktadır:

1. İterasyon sayısını artırmak, bazen oldukça iyi çözümler elde edilmesini sağlamaktadır. Ancak sistematik bir şekilde ve her zaman iyi çözümlere ulaşılacağı anlamına gelmemektedir. Aşağıda j305-3 probleminin 500 ve 1000 iterasyon sonucunda elde edilen çözümleri gösterilmektedir. Algoritmada kullanılan genetik parametre değerleri Tablo 2' de verilmektedir. 500 iterasyonda 75 çözümü optimum olarak bulunurken, 1000 iterasyonda 92 olarak bulunmuştur.

Tablo 2: J305-3 Çizelgeleme Probleminde Kullanılan Genetik Parametreler

Faaliyet sayısı	32
Toplum büyüklüğü	20
Toplum sayısı (iterasyon)	500
Çaprazlama oranı	0.500
Mutasyon oranı	0.500
Çok noktalı çaprazlama	Evet / 2 noktalı çaprazlama

Problemin 500 iterasyon ile test edilmesi sonucunda aşağıdaki bilgiler elde edilmektedir.

Son Toplumun Değerlendirilmesi:

En kısa proje tamamlama süresi: 75

En iyi çözümün faaliyet sıraları: 1-2,3,4-10,5,6,18,16-7,12,15,13-17,26,9,19-11,14,29-8,21,20-25,23-22,24-28,27-31,30-32

Projenin Değerlendirilmesi:

En kısa proje tamamlama süresi : 75

En uzun proje tamamlama süresi: 100

En kısa proje tamamlama süresi(KYM) : 54

Algoritma işlem süresi(CPU time) : 00:00:27

Aynı problemin 1000 iterasyondaki sonuçları ise aşağıda verilmektedir.

Son Toplumun Değerlendirilmesi:

En kısa proje tamamlama süresi: 92

En iyi çözümün faaliyet sıraları: 1-2-3,4,16-6,18,5,12-7,10,15-13,11,14,19-20,8,26-22,9-17,29-21,23,27-25,24-28-31-30-32

Projenin Değerlendirilmesi:

En kısa proje tamamlama süresi : 92

En uzun proje tamamlama süresi: 111

En kısa proje tamamlama süresi(KYM) : 54

Algoritma işlem süresi(CPU time) : 00:01:07

Dolayısıyla aynı veri kümesi için iterasyon sayısını artırma yolu ile çözüm uzayını genişletme yöntemi, daha iyi çözümleri elde edebilmek açısından yeterli olamamaktadır.

2. Yine çalışmada özellikle optimum çözümden büyük sapmalar gösteren örnek problemlerin çözümünde çaprazlama noktası sayısını artırarak yapılan denemelerde de sistematik bir iyileşmeye rastlanamamış, rastsal olarak daha iyi ya da daha kötü çözümlere rastlanmıştır.

3. Geliştirilen algoritmada, farklı parametrelerle yapılacak testlerle elde edilen çözümlerden en iyisini seçmek gerekmektedir. Çizelgeleme probleminin en iyi çözümünün elde edilmesi arayışında; farklı çaprazlama noktası, artırılan iterasyon ve toplum sayıları ile yapılan denemeler sonucunda elde edilen çözümler içerisinde en iyisini seçmek gerekmektedir. Bu amaçla, parametrelerin farklı kombinasyonları ile denemelerin yapılması tavsiye edilebilmektedir.

4. Çok faaliyetli ve çözümü zor çizelgeleme problemlerinden olan j30 veri kümesinde, 500 iterasyon için ortalama 29 saniye, 1000 iterasyon için ortalama 6:24 (6 dakika) gibi kısa işlem süreleri sonucunda çözüm alternatiflerinin elde edilmesi, GA kullanımının önemini artırmaktadır. Geliştirdiğimiz algoritma ile veri kümesindeki her bir örneğin optimum sonucuna erişilememektedir. Veri kümesinde bulunan toplam 480 adet örnek problemde elde edilen çözümlerin optimum çözümden sapmaları ortalama 0,20' dir. Genel olarak optimuma çok yakın sonuçlar vermese bile bu kadar kısa bir sürede, çizelgeleme projesinin tamamlanma süresi hakkında global bir bilgiye erişim söz konusu olabilmektedir. Dileğimiz, daha sonraki çalışmalarımızda algoritmaya, optimum çözüme daha fazla yakınsama yapacak işlemlerin ilave edilmesi ile bu ve benzeri sürede daha etkin çözümler üretebilmektir.

5. Sonuç

Yapılan testler, algoritmanın bulduğu en iyi çizelge alternatifinin, yöneticiler için en iyi proje tamamlanma süresi olarak en azından bir fikir verebileceği sonucunu doğurmaktadır. Çünkü bir problemde, GA ile üretilen en iyi çözümlerin statik olmadığı, her denemede değişebileceği görülmektedir.

Daha iyi çözümler elde etmek amacıyla çözüm uzayını genişletmek, yani iterasyon sayısını artırmak yeterli bir şart olamamakla birlikte, gerekli bir unsur olarak değerlendirilebilmektedir. Böylece rastgele genişletilen çözüm uzayında en iyiye rastlama olasılığı değerlendirilerek, etkin çözüm arayışlarına imkân tanınabilmektedir. Bu nedenle, çizelgeleme problemlerine çözüm arayışları esnasında kullanılması gerekmektedir.

Toplumu oluşturan bireylerde ikiz ya da çoğul oluşumlardan kaçınarak, toplumu farklılaştırmak çözüme yakınsama sağlayabilecek bir unsur olabilir. İterasyon sayısının ve toplum büyüklüğünün yüksek alınması gereken karmaşık problemlerde; çalışmamızda da yapıldığı gibi, rastgele oluşumlara yer verilmesi nedeniyle benzer çözümlerin tekrarı söz konusu olabilmektedir. Alternatif çözümlerdeki bu tekrarlı oluşumlar, arama işleminde, aramayı yavaşlatıcı ya da etkin çözüme yakınsamayı geciktirici/engelleyici etkiye sahip olabilmektedir.

Geliştirilen algoritmada, farklı parametrelerle yapılacak testlerle elde edilen çözümlerden en iyisini seçmek gerekmektedir. Çalışmamızda kısmen yapılan bu çeşitliliğin daha da artırılması fayda sağlayabilir. Çizelgeleme probleminin en iyi çözümünün elde edilmesi arayışında; farklı çaprazlama noktası, artırılan iterasyon ve

toplum sayıları ile yapılan denemeler sonucunda elde edilen çözümler içerisinde en iyisini seçmek gerekmektedir.

Genetik algoritma sonucunda elde edilen çözümün, çizelgeleme problemlerinde anlamlı bir fikir oluşturması açısından kullanılan algoritmanın farklı genetik oranlarının yanı sıra, farklı genetik işlemler ile denenmesi yararlıdır sonucu çıkarılmaktadır. Bu nedenle bundan sonraki çalışmalarda, gerek çizelgeleme gerekse genetik işlem ve oranlar açısından, sezgisel modellerin birçok kombinasyonunu, kullanıcının tercih edebilmesini sağlayan algoritmaların geliştirilmesi daha yararlı olabilecektir.

KAYNAKÇA

- Artigues, Christian, Philippe Michelon ve Stephane Reusser (2003), "Insertion techniques for static and dynamic resource-constrained project scheduling", European Journal of Operational Research, Volume 149, No. 2, 1 September, ss.249-267.
- Boctor, Fayez F.(1996), "A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes", European Journal of Operational Research, Volume 90, No. 2, 19 April, ss.349-361.
- Bolat, Berna, K.Erol Osman ve Erdem C.İmsak (2004), "Mühendislik uygulamalarında genetik algoritma ve operatörlerin işlevleri", Sigma 2004/4, Mühendislik ve Fen Bilimleri Dergisi, ss.264-271.
- Brochmann, Harold (2005), "A genetic algorithm",
<http://www.salts-brucker-brucker-pring.com/brochmann/math/GA/GA-1.00.html> [3.5.2005].
- Brucker, Peter, Andreas Drexl, Rolf Möhring, Klaus Neumann ve Erwin Pesch (1999), "Resource-constrained project scheduling: Notation, classification, models, and methods", European Journal of Operational Research, Volume 112, No. 1, 1 January 1999, ss.3-41.
- Buckles, Bill P. ve Frederick E. Petry (Derl.) (1992), Genetic Algorithms, Washington: IEEE Computer Society Press, Technology Series.
- Chan, Felix T.S., S.H Chung ve Subhash Wadhwa (2005), "A hybrid genetic algorithm for production and distribution", The International Journal of Management Science, Omega 33, ss.345-555.
- Cheng, Runwei. ve Mitsuo Gen (1994); "Evolution program for resource constrained project scheduling problem", Evolutionary Computation, IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference ,27-29 June 1994, Vol.2, ss.736 – 741.

- Debels, Dieter ve Mario Vanhoucke (2005), "A bi-population based genetic algorithm for the resource constrained project scheduling problem", Vierick Leuven Gent Working Paper, (Seri No: 8).
- Goldberg, David E. (1992), "Sizing populations for serial and parallel genetic algorithms", Derl.: Bill P.Buckles and Frederick E. Petry, Washington: IEEE Computer Society Press, Technology Series, ss.20-29.
- Hartmann, Sönke ve Rainer Kolisch (2000), "Experimental evaluation of state-of-art heuristics for the resource constrained project scheduling problem", European Journal of Operational Research, 127, ss.394-407.
- Hindi, S.Khalil, Hongbo Yang ve Krzysztof Fleszar (2002), " An evolutionary algorithm for resource constrained project scheduling", IEEE Transactions on Evolutionary Computation, Vol.6, No.5, October.
- Kolisch, Rainer, Arno Sprecher ve Andreas Drexler (1995), "Characterization and generation of a general class of resource constrained project scheduling problems", Management Science, Volume 41, No. 10, October, ss.1693-1703.
- Kolisch, Rainer (1996), "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation", European Journal of Operational Research, Volume 90, No. 2, 19 April, ss.320-333.
- Kolisch, Rainer ve Arno Sprecher (1997), "PSPLIB - A project scheduling problem library : OR software - ORSEP operations research software exchange program", European Journal of Operational Research, Volume 96, No. 1, 10 January, ss.205-216.
- Kolisch, Rainer ve Sönke Hartmann (1999), "Heuristics algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis", Kluwer Academic Publishers In: Weglarz. J.(ed.).Project Scheduling-Recent Models, Algorithms and Applications, ss.147-178, Boston.
- Koza, John R. (1995), "Two ways of discovering the size and shape of a computer program to solve a problem", Proceedings of the Sixth International Conference on Genetic Algorithm, ss.287-294.
- Kurt, Mustafa ve Cumali Semetay (2001), "Genetik algoritma ve uygulama alanları", http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm, [12.4.2005].
- Mitchell, Melanie ve Charles E. Taylor (1999), "Evolutionary computations: an overview", Annual Review of Ecology and Systematics, 30: ss.593-616.
- Mitchell, Melanie (1999), An Introduction to Genetic Algorithms, USA: Massachusetts Institute of Technology.

- Mori, Masao ve Ching Chih Tseng (1997), "A genetic algorithm for multi-mode resource constrained project scheduling problem", *European Journal of Operational Research*, Volume 100, No. 1, 1 July, ss.134-141.
- Nakamura, Morikazu., N.Yamashiro, Y.Gong, T.Matsumura ve K.Onaga (2005), "Iterative parallel genetic algorithms based on biased initial population", *IEICE Trans. Fundamentals*, (April), Vol.E88-A, No.4, ss.923-929.
- Özdamar, Linet (1999), "A genetic algorithm approach to a general category project scheduling problem", *Systems, Man and Cybernetics, Part C, IEEE Transactions*, Vol. 29, No 1, February, ss.44 – 59.
- Simpson, Wendell P.III, ve James H. Patterson (1996), "A multiple-tree search procedure for the resource-constrained project scheduling problem", *European Journal of Operational Research*, Volume 89, No. 3, 22 March, ss.525-542.
- Tormos, P. ve Antonio Lova (2003), "An efficient multi-pass heuristic for project scheduling with constrained resources", *International Journal of Production Research*, Vol.41, No.5, ss.1071-1086.
- Ulusoy, Gündüz (2006), "Proje planlamada kaynak kısıtlı çizelgeleme", Sabancı Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, [http://people.sabanciuniv.edu/gunduz/HalimDogrusoz .pdf](http://people.sabanciuniv.edu/gunduz/HalimDogrusoz.pdf) , [8.7.2006].
- Wall, Mathew Bartschi (1996), "A genetic algorithm for resource constrained scheduling", Doktora Tezi, Massachusetts Institute of Technology, Cambridge.
- Wang, Peng-Yi ve Ming Lu (2002), "Genetic algorithm optimized resource activity critical path method", *Proceedings of the First Conference on Machine Learning and Cybernetics*, Vol.. 4, 4-5 November, ss.1978-1982.
- Yeo, M. F. ve E. O. Agyei (1998), "Optimising engineering problems using genetic algorithms", *Engineering Computations*, Vol.15, No.2, ss.268-280.