

GENETİK ALGORİTMANIN ÇİZELGELEME PROBLEMLERİNDE KULLANIMI

Â. Yurdun ORBAK

Uludağ Ün.v.

Mühendislik-Mimarlık Fak.ültesi

Endüstri Mühendisliği Bölümü

orbak@uludag.edu.tr

Aslı ERDİLLER

Uludağ Üniversitesi

Mühendislik-Mimarlık Fakültesi, Endüstri

Mühendisliği Bölümü

erdiller@uludag.edu.tr

ÖZET

Genetik algoritmaların en uygun olduğu problemler geleneksel yöntemler ile çözümü mümkün olmayan veya çözüm süresi problemin büyüklüğü ile üstel orantılı olarak artanlardır. Genetik algoritmaların çizelgeleme problemlerinde kullanımları ise son yıllarda oldukça yoğunlaşmıştır.

Bu bildiride, Basit Genetik Algoritmanın (SGA-*Simple Genetic Algorithm*) çizelgeleme problemlerine uygulanmak üzere oluşturulmuş yeni bir uyarlaması sunulmaktadır. Yapılan değişiklikler amaç fonksiyonlarının oluşturulması, amaç fonksiyon değerlerinin ve uygunluk değerlerinin hesaplanmasını içeren alt fonksiyonları kapsamaktadır. Amaç fonksiyonları olarak çizelge zamanı, geciken iş sayısı ve toplam gecikme süresi kullanılmıştır. Yapılan değişikliklerle oluşturulan geliştirilmiş genetik algoritma ile aktif ve ertelemesiz çizelgeler yakın optimal çözümlerle oldukça kısa hesaplama zamanlarında elde edilebilmektedir.

Anahtar kelimeler: Çizelgeleme, Genetik Algoritmalar, Eniyileme.

1. GİRİŞ

Tabu arama ve tavlama benzetimi yöntemlerine göre daha genel yöntemler olan genetik algoritmalar ilk olarak Holland ve diğerleri (1975) tarafından geliştirilmiştir. Genel anlamda genetik algoritmalar rassal çözümlerden oluşan bir popülasyonun bir dizi değişimden geçerek yüksek kalitede çözümlere meyilli seçim mekanizması ile yakın-eniyi çözümler üretmeyi amaçlar.

Genetik algoritmaların etkinliğini artırmak için algoritma bazı yöntemlerle zenginleştirilir. Dorndorf ve Pesch (1995) kayan darboğaz yöntemi ile genetik algoritmaları birleştirerek kayan darboğaz yönteminde karar ağacında düğüm seçimini kontrol eden bir genetik algoritma geliştirmişlerdir. Geliştirdikleri bir başka algoritmada ise genetik algoritma ile öncelik kurallarını birleştirmişlerdir.

Bir başka yaklaşım serim-çözüm gösterimine dayanan ve Aarts ve diğ. (1994) ve Nakano ve Yamada (1991) tarafından tanımlanmış olan bir yaklaşımdır. Gezgin satıcı probleminde elde edilen iyi

sonuçlara dayanarak oluşturulan bu yaklaşımlardan ilki belirli bir zaman kısıtı altında beklenen performansı gösterememiştir. Nakano ve Yamada (1991) tarafından önerilen yaklaşım ise büyük popülasyonlar ve oldukça fazla adım sayısı ile ancak yakın-eniyi çözümlerde kalmıştır. Storer ve diğ.(1992) tarafından önerilen yaklaşımda yakın-eniyi çözümler elde edilmiştir.

Gezgin satıcı problemi çaprazlama operatörlerini atölye çizelgeleme çerçevesinde kullanma fikrinden yola çıkarak oluşturulan Bierwirth (1995) gösteriminde her birey atölyedeki işlem sayısının uzunluğunun bir dizisi olarak gösterilmiştir.

2. GENETİK ALGORİTMALAR

Genetik algoritmalar ile ilgili ilk çalışmaları Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland ve öğrencileri yapmıştır. Mekanik öğrenme (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramından etkilenecek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini görmüş; çalışmalarının sonucunu açıkladığı kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar (veya kısaca GA) olarak literatüre geçmiştir (Goldberg 1989). Ancak Holland'ın öğrencisi olan Goldberg bu konuda bir klasik sayılan kitabını yayınlamaya dek genetik algoritmaların pek pratik yararı olmayan bir araştırma konusu olduğu düşünülmüştür. Ancak Goldberg kitabında genetik algoritmalarla dayalı tam 83 uygulamaya yer vererek genetik algoritmaların dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu göstermiştir.

Genetik algoritma, doğadaki evrim mekanizmasını örnek alan bir arama yöntemidir (Goldberg 1989). Genetik algoritmalar, doğada geçerli olan en iyinin yaşaması kuralını rassal bilgi değişimi ile birleştirerek bir arama algoritması oluştururlar. Her nesilde bir önceki neslin eniyileri kullanılarak yeni bireyler elde edilmeye çalışılır. Bunun için "iyi"nin ne olduğunu belirleyen bir *uygunluk* (fitness) fonksiyonu ve yeni çözümler üretmek için *yeniden üreme* (reproduction), *çaprazlama* (crossover) ve *değişim* (mutation) gibi operatörleri kullanır. Genetik algoritmaların bir diğer önemli özelliği de bir grup çözümle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötülerini elenebilir (Goldberg 1989, Pirlot 1996).

Genetik algoritmalarda kullanılan bazı kavram ve terimler aşağıda açıklanmıştır (Goldberg 1989, Deb 2001).

Evrim (Evolution): Genetik bilgi taşıyan bir topluluk üzerinde genetik işlemlerin uygulanması süreci olarak tanımlanabilir.

Evrimsel Algoritma (Evolutionary Algorithm EA): Genetik algoritmaları da içine alan algoritmik yöntemlerdir.

Genetik Programlama (Genetic Programming GP): Genlerinde program parçacıklarının kodlandığı diziler üzerinde çalışan bir genetik algoritma yolu ile istenilen işi yapan bir programın oluşturulması sürecidir.

Dizi (string): Genetik algortimada kullanılacak olan parametrelerin kodlanması ile oluşan doğadaki kromozomlara benzer sıralardır. Dizilerin her elemanına bit adı verilir. Kullanılan kodlama sistemine göre bitlerin alacağı değerler değişmektedir. Örneğin ikili kodlama kullanıldığında her bit "0" ve "1" değerlerini alabilirken, gerçek parametrelili genetik algoritmalarda bitler farklı değerler alabilmektedir.

Nüfus (Population): Genetik algoritmanın bulacağı veya belirleyeceği parametrelerin kodlanması ile oluşan dizilerin oluşturduğu gruptur.

Nüfus boyutu (Population size): Nüfusu oluşturan dizilerin sayısıdır ve işlem boyunca sabit kalır.

Allel (Allele): Bir özelliği temsil eden bir genin alabileceği değişik değerlerdir.

Gen (Gene): Kendi başına anlamlı genetik bilgi taşıyan en küçük genetik yapıdır. Örneğin 101 bit dizisi bir noktanın x- koordinatının ikilik düzende kodlandığı bir gen olabilir.

Seçme (Selection): Bir dizi havuzundaki dizilerden hangilerinin başarı değerlerine bağlı olarak yeni yaratılacak havuza aktarılacağını belirlenmesi işlemidir.

Eşleme (Mating): İki dizinin yeni bir bireyi oluşturmak üzere çaprazlanması amacıyla seçilmesi işlemidir.

Genetik algoritmalarla sıklıkla kullanılan operatörler üreme, çaprazlama, değişim ve tersinmedir.

Üreme (Reproduction): Üreme işlemi daha iyi uygunluk değerine sahip dizilerin daha yüksek olasılık ile yeni nesilde geniş sayıda kopyalarını üretebilen işlemidir. Bu işlem yapılırken uygunluk değeri ortalama değer ile normalize edilir ve diziler yeni değere göre seçilir. Bu durumda ortalama uygunluk değerinin üzerindeki diziler, ortalama uygunluk değerinin altındaki dizilerden daha fazla üretilmeye sahip olurlar. Uygunluk değerine göre dizileri kopyalamak bir sonraki nesilde daha fazla ürün oluşma olasılığının yüksek olması demektir. Örneğin 100111011 dizisi çoğaltıldığında bu dizinin aynısı olan 100111011 ve 100111011 dizileri elde edilir.

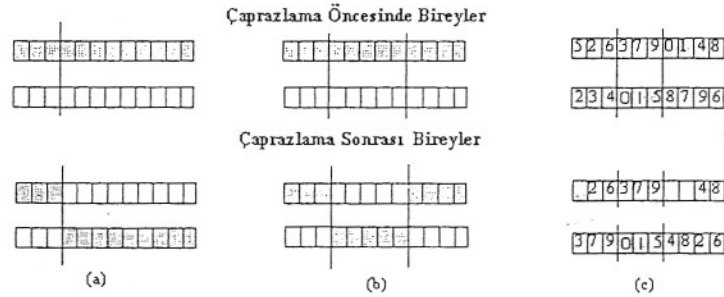
Üreme işlemi birçok şekilde algoritmik olarak elde edilebilmektedir. Bu işlem için en basit yöntem rulet tekerleği yöntemidir. Bu teknikte nüfustaki her dizi için uygunluk değeri ile orantılı olarak tekerlek üzerinde hisse verilir. Uygunluk değerine bağlı olarak her dizi tekerlek üzerinde belli bir yüzdeye sahip olur. Dizilerin yeterli sayıda üreme işlemleri için rulet tekerleği çevrilir. Diziler uygunluk değerlerine göre hesaplanan olasılıklarla kopyalanır. Kopyası üretilen diziler eşleştirme havuzunda (mating pool) toplanarak diğer operatörlerin uygulanması için hazırlanırlar.

Çaprazlama (Crossover): Çaprazlama işlemi, bilgilerin iki dizi arasındaki değişimi ile ilgilidir. Çaprazlama işleminde, yeni üretilmiş iki dizi eşleştirme havuzundan seçilerek rassal olarak belirlenmiş çaprazlama noktasına göre diziler arasında bilgi değişimi gerçekleştirilir. Bu işlem ile tercih edilmiş iyi dizilerden daha iyi özellikleri olan diziler elde edilebilir.

Çaprazlama işlemini gerçekleştirmek için ilk olarak üreme işlemi ile oluşturulmuş ve eşleşme havuzunda toplanmış olan diziler rassal olarak eşleştirilir. Daha sonra seçilen dizilerin bitleri rassal olarak seçilmiş çaprazlama noktasından itibaren karşılıklı olarak değiştirilir.

Çaprazlama işleminin uygulanacağı dizinin " l " uzunluğunda olduğu kabul edilirse çaprazlama noktası dizi boyunca 1 ile $(l-1)$ arasında rassal olarak seçilir. Seçilen bu nokta " k " ile ifade edilirse iki yeni dizi " $k+1$ " ve " l " noktaları arasındaki bütün bitlerin karşılıklı değişmesi ile oluşmaktadır.

Örnekte kullanılan çaprazlama yöntemi tek noktalı çaprazlamadır (Şekil 1-a). Çaprazlama işlemi farklı şekillerde de gerçekleştirilebilir. İki noktalı çaprazlama (Şekil 1-b) ve sıralı çaprazlama (Şekil 1-c) en sık kullanılan çaprazlama yöntemleridir. İki noktalı çaprazlamada rassal olarak belirlenen iki çaprazlama noktası arasında kalan bölge iki birey arasında değiştirilir. Sıralı çaprazlama gerçek parametrelili genetik algoritmalarla kullanılan bir çaprazlama yöntemidir. Sıralı çaprazlama yöntemi ile birey içersinde her bir bit değerinden bir adet olması sağlanır.



Şekil 1 Çaprazlama Yöntemleri (a) Tek noktalı (b) İki noktalı (c) Sıralı çaprazlama

Değişim (Mutation): Değişim, üreme ve çaprazlama işlemlerinin tamamlayıcı bir işlemidir. Değişim işlemi basit olarak bit değerlerini tersine çevirmektir. Rassal olarak seçilmiş bir noktadaki değerin değişikliği ile eniyiye ulaşma olasılığı artırılmış olur. Örneğin; 10101110000 dizisinin 3. bitinin değişmeye uğraması sonucunda 10111110000 dizisi ortaya çıkar. Değişim işlemi " p_m " olasılığı ile tek bir bitin rassal olarak değişimi olup bu işlem oluşturulmuş neslin elverişli durumunu birden bozabilir. Bu yüzden p_m olasılığı oldukça küçük olarak alınır.

Tersinme (*Inversion*): Bir diziyi oluşturan genlerden ardışık bir grubun kendi içerisinde birbirleri ile yer değiştirerek ters dizilmeleridir. Örneğin; 0111101011 dizisi (her genin bir konum olduğu varsayımı ile) 5. ve 8. gen konumları arasında tersindiğinde ortaya 0111101011 dizisi ortaya çıkar.

2.1. Genetik Algoritmaların Diğer Eniyileme Yöntemlerinden Farkları

Genetik algoritmalar diğer eniyileme yöntemlerinden dört açıdan farklılık gösterir. Bu farklılıklar aşağıda verilmektedir (Goldberg 1989).

1. Genetik algoritmalar parametrelerin kendileriyle değil parametre dizilerinin kodlanması ile çalışır.
2. Genetik algoritmalar bir tek noktada değil, noktalardan oluşan nüfus içinde arama yapar.
3. Genetik algoritmalar, verilen fonksiyonun türevi veya yardımcı fonksiyonları değil amaç fonksiyonunu kullanır.
4. Genetik algoritmalar olasılık kurallarını kullanırlar. Genetik algoritmalarda sınırlayıcı kurallar yoktur.

Bir çok eniyileme yönteminde tek bir noktadan çözüme başlanarak bir sonrakine değişim kuralı ile çözümden çözüme hareket edilir. Genetik algoritmalar ise bu yöntemlerin tersine, geniş veri tabanlı noktalara eşzamanlı olarak çalışırlar. Bu yaklaşım nedeniyle genetik algoritmalar hem genel eniyiye daha kolay ulaşırlar, hem de yerel eniyilerde durma riskini azaltırlar.

Genetik algoritmalar doğrudan kodlamanın kullanılması, nüfusu araştırma yaklaşımı, yardımcı bilgiye ihtiyaç duyulmaması ve kullanılan operatörlerin rassal olması ile oluşan farklılık kararlılığa katkıda bulunmaktadır (Goldberg 1989).

2.2. Genetik Algoritmaların Çalışma Prensipleri

Tüm genetik algoritmaların çalışma prensibi genellikle aynıdır. Klasik bir genetik algoritmanın içerdiği adımlar aşağıdaki gibi açıklanabilir:

Adım 1: Bu adıma nüfusta 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. Nüfus bu işlem后会 sonra rassal olarak oluşturulur.

Adım 2: Nüfusu oluşturan her dizinin çözüme uygunluğunu belirleyen uygunluk fonksiyonu vardır. Bu fonksiyon işletilerek dizilerin uygunluk fonksiyon değerlerinin bulunmasına ise *hesaplama* (evaluation) adı verilir. Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır. Genetik algoritmada probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu dizileri problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte (decoding), sonra bu parametrelere göre hesaplamayı yaparak dizilerin uygunluğunu bulmaktadır. Çoğu zaman genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır.

Adım 3: Dizilerin eşleşmesi dizilerin uygunluk değerlerine göre yapılır. Bu seçimi yapmak için *rulet tekerleği seçimi* (roulette wheel selection), *turnuva yöntemi* (tournament selection) veya *sıralamalı seçim yöntemi* (ranking selection) kullanılabilir. En sık kullanılan seçim yöntemi olan rulet tekerleği seçimi yönteminin adımları aşağıdaki gibidir (Golberg 1989):

- 1- Tüm bireylerin uygunluk değerleri bir tabloya yazılır.
- 2- Bu değerler toplanır.
- 3- Tüm bireylerin uygunluk değerleri, toplama bölünerek [0,1] aralığında normalize edilmiş uygunluk fonksiyon değerleri elde edilir. Bu değerler bireylerin seçilme olasılıklarıdır. Değerlerin hepsi bir tabloda tutulur.
- 4- Seçilme olasılıklarının bulunduğu tablodaki değerler birbirine eklenerek rassal olarak bir değere kadar ilerlenir. Bu değere ulaşıldığında ya da geçildiğinde son eklenen değerin ait olduğu çözüm seçilmiş olur.

Rulet tekerleği seçimi çözümlerin uygunluk değerlerinin negatif olmamasını gerektirir. Çünkü olasılıklar negatif olursa bu çözümlerin seçilme şansı yoktur. Çoğunluğunun uygunluk değeri negatif olan bir nüfusta yeni nesiller belli noktalara takılıp kalabilir.

Çaprazlama nüfusta çeşitliliği sağlar. İyi özelliklerin bir araya gelmesini kolaylaştırarak en iyiye yaklaşmayı sağlar. Değişim (mutasyon) dizinin bir bitinin rassal olarak dışarıdan değiştirilmesi şeklinde tanımlanır. Değişim görünüşte genetik algoritmanın dayanak noktasıdır, ancak etkisi bir çözüm

üzerindedir. Bu da yalnız başına başarılı olmasını zorlaştırır. Çok düşük bir değiştirme olasılığı nüfusta bazı özelliklerin kaybolmasına neden olabilir. Bu da en iyi sonuçların bulunmasına engeldir. Ancak yüksek bir değiştirme olasılığı da eldeki çözümleri bozarak sonuca ulaşmayı zorlaştırır. Çaprazlama ve değişimin olasılıkları için kesin bir sayı yoktur. Değişim (mutasyon) olasılığı 0.01-0.001, çaprazlama (crossover) olasılığı 0.5-1.0 aralığında tavsiye edilir (Goldberg 1989).

Adım 4: Eski diziler nüfustan çıkartılarak sabit büyüklükte bir nüfus sağlanır.

Adım 5: Tüm dizilerin uygunluk fonksiyon değerleri yeniden hesaplanarak yeni nüfusunun başarıları bulunur.

Adım 6: Genetik algoritma belirlenen nesil sayısı kadar çalıştırılarak çok sayıda nüfus oluşturulup hesaplanır.

Adım 7: Nüfusların hesaplanması sırasında en iyi bireyler saklandığı için o ana kadar bulunmuş en iyi çözüm çözümdür.

3. ÇOK AMAÇLI GENETİK ALGORİTMANIN BİLGİSAYAR UYGULAMASI

Bu çalışmada Golberg (1989) tarafından hazırlanmış ve Pascal dilinde kodlanmış olan Basit Genetik Algoritma (SGA- Simple Genetic Algorithm) gerekli değişiklikler yapılarak kullanılmıştır. Algoritma yedi alt fonksiyon içermektedir. Bu alt fonksiyonlardan ilki başlangıç nüfusunun oluşturularak gerekli hesaplamaların yapıldığı *initialize* alt fonksiyonudur. *Generation* alt fonksiyonu ile her bir neslin seçme, çaprazlama ve değişim kullanılarak oluşturulması sağlanır. Seçme, çaprazlama ve değişim operatörlerinin bulunduğu alt fonksiyon *triops* alt fonksiyonudur. En önemli alt fonksiyonlardan birisi olan *interface* alt fonksiyonu amaç fonksiyon değerleri olan çizelge zamanını, toplam pozitif geç kalmayı ve geç kalan iş sayısı ile uygunluk fonksiyonu değerini hesaplamaktadır. *stats* alt fonksiyonu her nesil için gerekli istatistikleri hesaplamakta, hesaplamaların sonuçları *report* alt fonksiyonu ile kullanıcıya sunulmaktadır. Genetik algoritmanın yapısındaki rassallığın sağlanması için *random* alt fonksiyonu kullanılmaktadır.

3.1. Amaç Fonksiyonu ve Uygunluk Fonksiyonu Değerlerinin Hesaplanması

Üretim sistemi için tanımlanmış olan amaçlar çizelge zamanının, geç kalan iş sayısının ve toplam pozitif geç kalmanın enküçüklenmesiydi. Bu amaçlardan sadece birisi uygunluk fonksiyonuna dahil edilebileceği gibi tamamı da uygunluk fonksiyonunda kullanılabilir.

Genetik algoritmada amaç fonksiyonu ve uygunluk değerlerinin hesaplanması *interfac* alt programında *objective* prosedürü ile gerçekleştirilmektedir. Amaç fonksiyonlarının değerlerinin hesaplanmasında temel kriter çizelge zamanıdır. Çizelge zamanının hesaplanması bir matris (*schedule*) yardımıyla yapılmaktadır. Oluşturulan matriste her makine için bir satır bulunmaktadır. İşlemlerin makinelere ataması yapılırken, gerekli makinede gerçekleştirilecek işlem süresini (P_{ij}) karşılayacak sürenin olup olmadığı kontrol edilir. Eğer gerekli boşluk (*space*) varsa, işlem bu boşluğa atanır. Eğer gerekli boşluk bulunamazsa, matrisin bu satırı taranarak gerekli boşluk bulunur. İşlemlerin çizelgeye atamaları yapılırken birbirlerine bağlı olmaları dikkate alınır. Bu işlemlere ait kodlar aşağıda verilmektedir.

```
if schedule[g,n]<>0 then begin
  repeat
    n:=n+1;
  until schedule[g,n]=0;
  t:=n;
  end;
  for n:=t to ty do begin
    if schedule[g,n]=0 then space := space + 1;
  end;
  if space>=Pij then
    for n:=t to ty-1 do
      begin schedule[g,n]:=s; end;
```

Bir dizideki tüm işlemler için bu işlemler tekrarlanarak çizelge tamamlanır ve o andaki en büyük zaman indisi değeri (T_{max}) çizelge zamanı (*makespan*) olarak atanır.

if $t > T_{max}$ *then* $T_{max} := t$;

Makespan := T_{max} ;

Her siparişin teslim tarihi (*DueDate*) ile tamamlanma zamanı karşılaştırılarak bu işin geç kalıp kalmadığı, kaldı ise pozitif geç kalması hesaplanır. Geç kalan tüm işlerin pozitif geç kalmaları toplanarak toplam pozitif geç kalma (*TotTardi*) hesaplanır. Geç kalan her iş için geç kalan iş sayısı sayacı (*NTar*) artırılır.

$dt := t - \text{DueDate}$;

if $dt > 0$ *then begin*

$TotTardi := TotTardi + dt$;

$NTar := NTar + 1$;

end;

Her dizi için hesaplanan değerler ilgili değişkenlere atanırlar.

Makespan := T_{max} ;

TTard := $TotTardi$;

NTardy := $NTar$;

Tümü amaç fonksiyon değerleri hesaplandıktan sonra dizinin uygunluk fonksiyonu değeri hesaplanır. Uygunluk fonksiyonu olarak amaç fonksiyonlarında birisi alınabileceği gibi tüm amaç fonksiyonları da uygunluk fonksiyonunda kullanılabilir. Uygunluk fonksiyonunda her amaç için ağırlıklar tanımlamak da mümkündür.

$fitness := 1 - (\text{Makespan} / \text{maksimum})$;

$fitness := 1 - (0.2 * TTard / \text{maksimum} + 0.3 * NTardy / ss + 0.5 * \text{Makespan} / \text{maksimum})$;

Bu algoritma ile gecikmesiz ve aktif iş sıraları tek aşamada ve tüm kriterler göz önüne alınarak üretilmektedir. Daha önce yapılmış olan çalışmalarda gecikmesiz ve aktif iş sıraları yaratabilmek için iki veya daha çok aşamalı yaklaşımlar kullanılmıştır. Bu tür yaklaşımlar ile sonuç elde etme süresi artmaktadır.

4. SONUÇLAR

Önerilen yaklaşımda çok amaçlı bir genetik algoritma kullanılmıştır. Genetik algoritmaların tek bir birey yerine bir nüfusla çalışma özelliği diğer yaklaşımlara göre önerilen yaklaşıma hız avantajı sağlamaktadır. Algoritmada kullanılan seçme, çaprazlama ve değişim işlemleri yöntemlerinin basit olması da algoritma hızını olumlu yönde etkilemektedir. Daha önce yapılmış olan çalışmalarda kullanılan eniyilik kriterleri çoğu zaman çizelge zamanı ile sınırlı kalmakta, bazı durumlarda makine yüklerinin düzgün dağılımı kriteri dikkate alınmaktadır.

Genetik algoritmaların etkinlikleri algoritmada kullanılan seçim, çaprazlama, değişim yöntemleri ile yakından ilişkilidir. Önerilen algoritmada seçim yöntemi olarak literatürde en sık kullanılan rulet tekerleği yöntemi, çaprazlama yöntemi olarak tek noktalı çaprazlama, değişim yöntemi olarak tek bir bitin değer değişimi seçilmiştir. Kullanılan tüm yöntemler Goldberg (1989)'in basit genetik algoritmasında kullanılan yöntemlerdir. Bu yöntemler oldukça basit yöntemlerdir. Bu yöntemlerin değiştirilmesi ile genetik algoritmanın etkinliği ve çözümü bulma hızı değişiklik gösterecektir.

Önerilen algoritma ile modellerin çözümünde kullanılmış olan çaprazlama ve değişim olasılıkları literatürde en sık kullanılan değerlerdir. Her ne kadar bu değerler ile elde edilen çözümler yeterli olsa da değerler değiştirilerek farklı denemeler yapılmalıdır. Son yıllarda genetik algoritmalar ile ilgili yapılan bazı çalışmalar parametre seçimi ve etkinlikleri üzerinedir. Dolayısıyla bu problem için de değerlerin değiştirilerek çözüm üzerindeki etkileri incelenebilir. Değerlerin değiştirilmesi ile yöntem seçiminde olduğu gibi algoritmanın etkinliği ve hızı değişiklik gösterecektir.

KAYNAKLAR

- BAKER, K.R.1994. Elements of Sequencing and Scheduling. Amos Tuck School of Business Administration. Hanover.
- BIEGEL, J.E., J.J. DAVERN. 1990. Genetic algorithms and job shop scheduling. Computers and Industrial Engineering. 19(1-4):81-91.
- CHOI, I., D. CHOI. 2002. A local search algorithm for job shop scheduling problems with alternative operations and sequence-dependent setups. Computers and Industrial Engineering. 42:43-58.
- COCHRAN, J.K., S.M. HORNG, J.W. FOWLER. Baskıdaki Makale (2002). A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. Computers and Operations Research.
- DEB, K. 2001. Multi-Objective Optimization using Evolutionary Algorithms, John Wiley and Sons, NJ, p 77-224.
- DENGİZ, B. <http://www.mmf.gazi.edu.tr/~berna/turkce/index.html>.
- GOLDBERG, D.E. 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Pub. USA. p 1-87.
- KAN, R.A.H.G. 1976. Machine Scheduling Problems: Classification, Complexity and Computation. Martinus Nijhoff. The Hague. 145 p.
- KUMAR, N.S.H., G. SRINAVASAN. 1996. A genetic algorithm for job shop scheduling-A case study. Computers in Industry. 31:155-160.
- MURATA, T., H. ISHIBUCHI, H. TANAKA. 1996. Multi-objective genetic algorithms and its application to flow shop scheduling. Computers and Industrial Engineering. 30(4): 957-968.
- PINEDO, M. 1995. Scheduling: Theory, Algorithms and Systems. Prentice Hall. NJ.
- PIRLOT, M.1996. General local search methods. European Journal of Operational Research. 92: 493-511.
- PONGCHAROEN, P., C. HICKS, P.M. BRAIDEN, D.J. STEWARDSON. 2002. Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. International Journal of Production Economics. 78:311-322.
- ZHOU, H., Y.FENG, L.HAN. 2001. The hybrid heuristic genetic algorithm for job shop scheduling. Computers and Industrial Engineering. 40:191-200.