

Çizge Boyama Problemleri İçin Evrimsel Tabu Arama Algoritması (ETA)

Evolutionary Tabu Search Algorithm for Graph Coloring Problem (ETA)

ÖZET

Bu makalede bir çok problemin indirgenebildiği klasik Çizge Boyama Problemlerinin çözümü için bir melez algoritma önerilmiştir. Evrimsel Tabu Arama Algoritması (ETA) adını verdiğimiz bu yaklaşım tabu arama yöntemi ile genetik algoritmanın birleştirilmesiyle ortaya çıkmıştır. Çizge boyama problemleri için yeni birer melezleme ve mutasyon uzmanı geliştirilmiş, DIMACS yarışmasında kullanılan pek çok deney örneği üzerinde test edilmiştir. ETA özellikle küçük ve orta boy çizge örneklerinde başarılı sonuçlar vermiştir.

ÖZGEÇMİŞ

Özgür Ülker

1980 yılında İstanbul'da doğan Özgür Ülker lisans eğitimini 2003 yılında Yeditepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nde tamamladı. 2002-03 yılları arasında aynı bölümde öğrenci asistanı olarak görev almıştır.

Ender Özcan

1991 yılında Orta Doğu Teknik Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü'nü bitirdi. Devlet bursuyla gittiği Syracuse Üniversitesi (ABD, New York), Bilgisayar ve Enformatik Bilimleri Bölümü'nde 1994 yılında yüksek lisansını, 1998 yılında da doktorasını tamamladı. Bu süre içerisinde araştırma görevliliğinin yanı sıra, BlackWatch Inc. şirketinde yazılım uzmanı olarak çalıştı. 1998 yılı güz döneminden beri Yeditepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nde öğretim üyesi olarak görevine devam etmektedir.

ABSTRACT

In this paper, a hybrid algorithm has been proposed for the Graph Coloring Problem to which many problems can be reduced. This approach, named as "Evolutionary Tabu Search Algorithm" (ETA) has emerged by combining a tabu search method with a genetic algorithm. For Graph Coloring Problem, new crossover and mutation operators are tested using many graph instances from DIMACS challenge. ETA yields successful results, especially in small and medium graph instances.

ÇİZGE BOYAMA PROBLEMLERİ İÇİN EVRİMSEL TABU ARAMA ALGORİTMASI (ETA)

GİRİŞ

Çizge Boyama Problemleri (ÇBP) en temel katışımsal eniyileme problemlerindedir. ÇBP yönsüz bir çizgenin düğümlerinin, komşu düğümler farklı renkleri alacak şekilde boyanması olarak tanımlanabilir.

ÇBP, başta kısıt sağlama ve çizelgeleme olmak üzere pek çok teorik ve pratik problemin temelini oluşturur. Bu problemin polinom zamanda çözülmemeyen tam (NP-Complete) bir problem olduğu kanıtlanmıştır [6].

ÇBP'nin çözümünde bu sebepten dolayı pek çok buluşsal yöntemden faydalanılmıştır. Davis [2] melez bir *genetik algoritma* kullanarak problemi çözmeye çalışmış, fakat elde ettiği sonuçlar tatminkar olmamıştır. Johnson [11] farklı soğutma teknikleriyle *tavlama* [12] yöntemi kullanarak küçük problem örneklerinde sonuca ulaşmıştır. Hertz ve de Werra [9] ise *tabu arama* [7][8] yöntemiyle verimli sonuçlar elde etmişlerdir. Fakat ÇBP'nin çözümünde en etkin sonuçların melez algoritmalar kullanılarak elde edildiği görülmüştür. Galinier ve Hao' [5] ile Fleurant ve Ferland [4] tabu arama ve genetik algoritmaları birleştirerek zor problem örneklerinde çok başarılı sonuçlar elde etmişlerdir.

Bu bildiri de Çizge Boyama Problemi (*Graph Coloring Problem*) için başka bir melez algoritma önerilmiştir. Bölüm 2 bu melez yöntemle kaynak oluşturan genetik ve tabu arama algoritmalarını açıklar. Bölüm 3 ise önerilen melez algoritmayı incelemektedir. Bölüm 4 test sonuçlarına ve bunlarla ilgili yorumlara ayrılmıştır. Ulaşılan sonuçlar ve ileride yapılacak çalışmalar ise Bölüm 5'te özetlenmiştir.

ÇİZGE BOYAMA PROBLEMLERİ

$G = (V, E)$ yönsüz bir çizgedir. $V = \{v_1, v_2, \dots, v_i, \dots, v_M\}$ çizgedeki düğüm kümesini, $E = \{e_1, e_2, \dots, e_j, \dots, e_N\}$ ise kenar kümesini temsil eder. $e_j = (v_p, v_q)$ ile ifade edilen ikili, V kümesine ait iki düğüm arasındaki kenardır.

Tanım 1: *Bağımsız küme* birbirleriyle komşu olmayan düğümleri içeren kümedir.

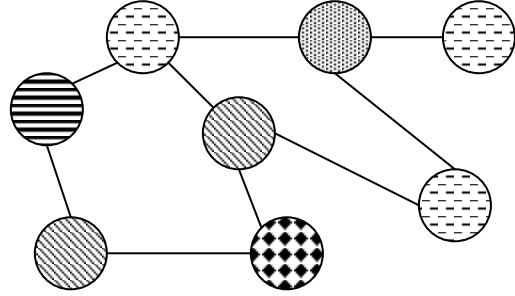
Gözlem: $V' \subseteq V, V' \neq \emptyset$ kümesinin bağımsız bir küme olabilmesi için ya $|V'|=1$ olmalı, yada $|V'|>1$ için $\forall x \in V' \wedge \forall y \in V' \Rightarrow (x, y) \notin E$ olmalıdır.

Tanım 2: $\langle G, k \rangle$ ile temsil edilen, k – renk çizge boyama problemi (*k-coloring problem*), V kümesinin birbiriyle ortak üyesi olmayan k bağımsız kümeye bölüntülenmesidir.

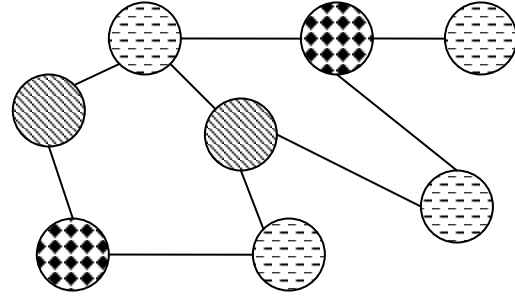
Tanım 3: *En iyi boyama*, mümkün olan en küçük k sayıdaki rengi kullanarak yapılacak olan boyamadır. Bu durumda k sayısı G çizgesinin *kromatik sayısı* ($\chi(G)$) olarak tanımlanır.

Çizim 1 ve Çizim 2'de sırasıyla k – renk çizge boyama problemi ile en iyi boyama problemine birer örnek verilmiştir. Çizim 1'de çizge $k = 5$ olacak şekilde 5 renkle

boyanmış, buna nazaran Çizim 2'de çizgenin $k = 3$ olacak şekilde en iyi boyanabildiği gösterilmiştir.



Çizim 1 : $k = 5$ olacak şekilde k - renk boyama



Çizim 2: $k = 3$ olacak şekilde en iyi boyama

GENETİK ALGORİTMALAR

Genetik Algoritmalar (GA) ilk defa J. Holland tarafından önerilmiş [10] ve pek çok zor problemin çözümünde kullanılmıştır. GA'ler evrim teorisinden esinlenilmiş, doğal seçim ve genetik biliminin temellerini kullanılarak geliştirilmiştir.

Tipik bir GA'de içindeki her birey (kromozom) in aday çözüm olduğu nüfus, sürekli bir evrim geçirir. Her nesilde belirli bir yöntemle seçilmiş olan bireylere melezleme ve mutasyon gibi çeşitli genetik uzmanlar uygulanır. Her bireyin hayatta kalma şansını bir uygunluk fonksiyonu belirler. Yapıtışı varsayımı gereği, güçlü bireyler iyi genlere sahiptir. Bu genlerin diğer nesillere geçme olasılığı seçim baskısı ile artırılarak, kötü genlere sahip güçsüz bireyler evrim esnasında kaybolması sağlanır. Evrim süreci belirlenen bitiş koşulunu sağlayıncaya kadar devam eder ve en son nüfustaki en iyi birey, ulaşılan en iyi çözümünü temsil eder.

Genetik Algoritma aşağıdaki ana bileşenlerden oluşur:

- Temsil
- Üreme için Seçim
- Melezleme (Üreme)
- Mutasyon
- Uygunluk Değeri Hesaplama
- Sonlandırma

TABU ARAMA ALGORİTMASI

Tabu Arama (TA) algoritması ilk defa F. Glover [7], [8] tarafından insan hafızasının çalışmasından esinlenilerek önerilmiş bir yerel arama yöntemidir. TA ana olarak basit tepe tırmanma (BTT) yönteminin zaafalarını gidermek için düşünülmüştür. BTT eldeki aday çözüme, bir komşuluk işleci uygulayarak, yeni adaylar üretir. Yeni adaylar bir değerlendirmeye tabi tutulur. Değerlendirme çözümün sonuca yakınlığını ölçer. Yeni adaylar ile eldeki eski aday içerisinden çözüme en yakın olan eskinin yerine geçer. BTT bu haliyle kısır bir döngüye sebep olabilir. Komşuluklar kapsamında birbirine eşit değerdeki iki yada daha fazla komşu arasında BTT takılıp kalabilir. Arama yapılan alanın özellikleri yada BTT yönteminin iyi seçilmemesi, herhangi bir denetim mekanizması kullanılmadan yapılan aramayı yerel en iyi noktalara götürebilir. Fakat gerçek hayattaki problemler yerel en iyilerin bol olduğu, fakat global iyinin az, hatta tek olabildiği problemlerdir. Kısacası kısır döngü kaçınılmazdır.

Tabu arama yöntemi kısır döngüden kurtulmak için hafıza kullanılmasını, hatırlamayı önerir. Daha önce ziyaret edilmiş ya da herhangi bir nedenle ziyaret edilmesi istenilmeyen aday çözümlerle ilgili özellikler, *tabu listesi* adı verilen, kısa dönem hafızaya benzer bir yapıda tutulur. Yöntem bu listedeki hamlelerin belirli bir süre yapılmasını yasaklar. Böylece arama yerel bir en iyi noktadan kurtularak asıl sonuca yakınsayabilir. Bu temel altyapının yanısıra, tabu arama yöntemlerinde aramayı belirli bir noktaya yönlendirebilecek uzun dönem hafıza yapıları da kullanılmıştır [16].

ÇİZGE BOYAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN MELEZ EVRİMSEL TABU ARAMA ALGORİTMASI

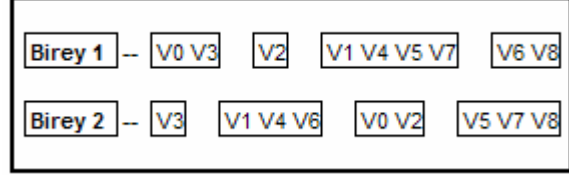
ÇBP'nin çözümü için melez evrimsel bir tabu arama algoritması önerilmektedir. Bu algoritma ana olarak yukarıda sözü geçen iki farklı algoritmanın birleştirilmesinden oluşmuştur. Bir nüfusun bireyleri genetik uzmanlar (melezleme ve mutasyon) ile tabu arama yöntemiyle sürekli bir evrim geçirirler. Bu süreç Çizim 3'de gösterildiği gibidir.

- Nüfustaki bireylerin oluşturulması
- Sonuç bulunana kadar yada nesil üst sınırına kadar
 - Melezleme için ebeveynler seç
 - Melezlemeyi uygula
 - Çocuk(lara) mutasyon uygula
 - Tabu Arama (uygula)
 - Yeni birey(lerin) uygunluk hesaplamasını yap
 - Yeni nesli oluştur

Çizim 3 : ETA algoritmasının bileşenleri

TEMSİL

Bu algortmada her bir birey, bağımsız renk kümelerinin birleşiminden oluşmuştur. Her bir renk kümesi o renge boyanacak olan düğümleri içermektedir. Aynı şekilde nüfusta belli sayıda bireyin toplamından oluşmaktadır. Çizim 4'te örnek iki birey gözükmektedir.



Çizim 4 : Bağımsız renk kümelerinden oluşmuş iki birey

NÜFUSTAKİ BİREYLERİN OLUŞTURULMASI

Her bireyin oluşturulması çizgedeki düğümlerin bağımsız renk kümelerine atanmasını gerektirir. Bunun için öncelikle hiç bir kümenin boş kalmamasını sağlamak için her kümeye bir düğüm rasgele atanmıştır. Diğer düğümlerin bir kısmı ise o an için en az hangi kümede kenar çakışımına sebep oluyorsa o kümeye atanmıştır. Geri kalan diğer düğümler ise tamamen rasgele atanmıştır. Bu ikisi arasındaki denge ise her bir birey için farklıdır. Örneğin bazı bireylere düğümlerin çok büyük bir kısmı rasgele atanabilirken, diğer bireylere ise kenar çakışımını azaltan bir atanma yapılabilir. Algoritmamızda bulunan oranlar aşağıdaki gibidir.

$$\text{rasgele atanan düğüm} = \text{birey no} / \text{toplam birey sayısı}$$

$$\text{sistemik atanma} = \text{toplam birey sayısı} - \text{rasgele atanan düğüm}$$

Birey no bütün bireylerin sıralı bir şekilde numaralandırılmasından sonra kaçınıcı olduğunu gösterir. Böylece son yaratılan bireyler sonuca daha uzak, ilk yaratılanlar ise daha az kenar çakışımı sağlanacak şekilde ayarlanmış olur.

Kullandığımız bu yaklaşımın amacı ilk nüfusu oluşturan bireylerin kenar çakışım sayısı ve düğümlerin kümelere yerleşimi açısından mümkün olduğunca birbirinden farklı olmasıdır. Aksi takdirde oluşturulacak çeşitliliği az olan bir nüfus sağlıklı bir şekilde evrim geçirme şansı yoktur. Bu sebepten dolayı diğer pek çok benzer algortmada kullanılan RLF [15] ya da DSATUR [1] gibi açgözlü yöntemler çeşitliliği azaltacağı için kullanılmamıştır.

EBEVEYN SEÇİMİ

Melezleme işleminin yapılabilmesi için öncelikle iki adet bireyin seçilmesi gerekmektedir. Bu algortmamızda bunun için *sıra bazlı ebeveyn seçimi* uygulanmıştır. Bu yöntemin en önemli avantajı, zayıf bireylere de seçilme şansı tanınmasıdır. Böylece güçlü bireylerin nüfusa erkenden hakim olması da engellenmiş olmaktadır.

Kullandığımız bu seçim yönteminde bütün bireyler ilk önce uygunluk derecelerine göre sıralanmış, daha sonra da her bireye bir derece atanmıştır. Algoritmamızda kullanılan derecelendirme sistemine göre ($n-j$)'nci bireyin seçilme olasılığı Denklem 1'deki gibidir.

$$P_{n-j} = \frac{j}{\sum_{k=1}^n k}$$

Denklem 1 – Bir bireyin sıra bazlı seçimdeki seçim baskısı

MELEZLEME UZMANI

Bu algoritmada melezleme uzmanı olarak Galinier ve Hao'nun önerdiği açgözlü bölüntüleme melezlemesi [5] yönteminin geliştirilmiş bir hali kullanılmıştır. Açgözlü bölüntüleme yöntemindeki ana hedef büyük bağımsız renk kümelerinin sonraki nesillere aktarılmasıdır. Bu yöntem iki adet ebeveyn bireyin en büyük renk kümelerinin sırayla çocuk bireye aktarılmasından oluşur. Yöntem kısaca aşağıdaki şekilde açıklanabilir:

```

Veri: Bağımsız kümelerden oluşmuş bireyler
P1, P2
Sonuç: Renk kümelerinden oluşan oğul birey C1

Başla

  Bütün renk kümeleri için
  P1'in en büyük kümesini C1'in kümesi
  olarak seç
  P1'in bu en büyük kümesini P1'den çıkar
  P1'in en büyük kümesinin elemanlarını
  P2'den sil
  P1 ile P2'yi yer değiştir

Bitir

Eğer atanmayan düğümler varsa
Her düğüm için rasgele bir küme seç
Bu düğümü seçilen kümeye ata

```

Çizim 5 : Açgözlü melezleme uzmanı

Bu yöntem başarılı bir şekilde tabu arama yöntemiyle beraber kullanılmıştır. Fakat bu melezleme uzmanının da yetersiz olduğu bazı noktalar bulunmaktadır. Bu uzman sadece uzun bağımsız renk kümelerinin gelecek nesillere iletilmesini sağlamakta, fakat bu kümelerin çakışım değerlerini gözardı etmektedir. Bu yöntemle uzun fakat çok fazla kenar çakışımına sahip kümeler de gelecek nesillere iletilmektedir. Bu sebepten dolayı, renk kümelerine bu şartı da değerlendirecek şekilde bir ağırlığa dayalı yöntem geliştirilmiştir. Bu yöntemde kümeler hem uzunlukları hem de çakışan kenarlarının sayısı göz önünde bulundurularak aşağıdaki formülle derecelendirilmişlerdir.

$$Derece_i = \alpha Uzunluk_i - \beta Ceza_i$$

Denklem 2 – Küme derecelendirme yöntemi

Bu formülde i kümesinin uzunluğu $Uzunluk_i$, aynı kümedeki kenar çakışım sayısı $Ceza_i$ dir. Bu özelliklerin α ve β katsayıları kullanılarak ağırlıklı toplamları alınarak ortaya çıkan sonuç, i kümesinin derecesi olarak belirlenmiştir. Deneylerimizde α ve β katsayıları 1 olarak kullanılmıştır.

Açgözlü Bölüntüleme yönteminin diğer bir zaafı ise melezleme sırasında bir renge boyanamayan düğümlerin rasgele olarak atanmasıyla ilgilidir. Deneylerimiz sırasında düğümlerin yaklaşık olarak %10 ile %40 olan kısmının herhangi bir yön belirlenmeden tamamen rasgele atıldığı görülmüştür. Bu rasgelelik gen havuzuna bir nevi çeşitlilik sağlasada bu oran çözüme ulaşmayı engelleyici olabilir. O

yüzden bu algoritmada bu düğümlerin belirli bir kısmı rasgele, diğer kısmı ise en az kenar çakışımına sebep olacak şekilde atanmıştır. Bu dağılımın nasıl yapılacağı ise yazı tura atılarak belirlenmiştir.

MUTASYON UZMANI

Algoritmada kullanılan mutasyon uzmanı renk kümelerindeki elemanların karıştırılmasından ibarettir. Öncelikle karıştırılacak iki tane renk kümesi rasgele seçilmiş, daha sonra da bu kümelerin elemanlarından rasgele bir permütasyon oluşturulmuştur. Bu permütasyonların elemanları daha sonra tekrar renk kümelerine dağıtılmıştır. Her kümenin kaç düğüm alacağı ise gene rasgele belirlenmiştir. Bu mutasyon vasıtasıyla nüfusa özellikle bireyler arasındaki farklılık çok azaldığında biraz daha çeşitlilik kazanması amaçlanmıştır.

UYGUNLUK DEĞERİ

Algoritmada kullanılan uzmanlarla elde edilen sonuçların istenen sonuca uzaklığını hesaplamak için bir ceza puanı uygulaması yapılmıştır. Bu ceza puanı aynı renge boyanmış düğümlerin arasındaki kenarların toplam sayısıdır.

YEREL ARAMA YÖNTEMİ

Algoritmamızda yerel arama metodu olarak Hertz ve de Werra'nın [9] önerdiği tabu arama yönteminin gelişmiş bir hali kullanılmıştır. Hamle olarak ise literatürde çokca kullanılan basit taşıma kullanılmıştır. Basit taşıma bir düğümün bir renk kümesinden diğer renk kümesine taşınmasından ibarettir. Bu yöntem vasıtasıyla komşuluk yapılabilecek bütün basit taşıma hamleleri olarak ortaya çıkar. Özellikle büyük çizgelerde bu komşuluğun boyutu çok yüksek mertebelere çıkabilir. Dolayısıyla komşuluğun boyunu daraltmak için iki adet seyrek örnekleme yapılmıştır. İlk olarak eğer bir düğüm bulunduğu kümede başka hiç bir düğümle çakışmıyorsa bu düğümle ilgili herhangi bir taşıma hareketi yapılmamıştır. İkinci önemli örnekleme metodu ise komşuluğun boyunun dinamik olarak değişmesiyle ilgilidir. Algoritmamızda komşuluk sonuçtan olan uzaklığa göre değişmektedir. Algoritmanın özellikle sonuçtan uzak olunan ilk aşamalarında, bütün komşuluğun taranmasına gerek görülmemiştir. Bunun tersine sonuca yaklaşılan bölümlerde komşuluğun boyunun genişlemesi sağlanmıştır. Sözü edilen bu parametre aşağıdaki şu formülle gösterilmektedir:

$$\%Boy = (1 - \alpha \frac{ToplamCeza}{ToplamKenar})^p * 100\%$$

Denklem 3 – Komşuluk boyunun hesaplanması

Bu formülde $ToplamCeza$, bütün bireydeki çakışan kenarların, $ToplamKenar$ ise çizgedeki bütün kenarların sayısına denk gelmektedir. α katsayısı ise bu ikisinin oranının (hedef çözümden uzaklık oranının) ne kadar önemli olduğunu ayarlar. p katsayısı komşuluğun ne kadarlık bir kısmının taranacağını belirler. Bu sayı arttıkça komşuluğun boyu hızlı bir şekilde azalır.

Tabu arama algoritması oluşturulan bu komşuluk içinde o an için en iyi hamleyi seçer. En iyi hamle ceza puanını en çok düşüren, bu mümkün değilse de en az yükselten

hamledir. Bu basit taşıma hamlesi yapıldıktan sonra bu hamlenin tersi tabu listesine girilir. Örneğin i -kümesindeki m elemanı j -kümesine taşınmışsa (hamle $\langle j, m \rangle$) m elemanı belirli bir süre için tekrar i kümesine dönemez (hamle $\langle i, m \rangle$). Bu süreyi belirlemek için aşağıdaki formül kullanılmıştır:

$$\text{ZamanAşımı} = \mu + \nu \text{ToplamCeza}$$

Denklem 4 – Tabu listesinde bulunma zamanının hesaplanması

Formülden de açıkça görülmektedir ki, tabu uzunluğu toplam cezayla olan ilişkisi μ ve ν katsayılarına bağlıdır.

Bu uzunluk sistemi büyük ve zor çizge örneklerinde olumlu sonuçlar verse de küçük çizgelerde yukarıdaki formülle elde edilen zaman aşımı kısa kalabilmektedir. Bu yüzden küçük çizgelerde tabu listesi zaman aşımı o andaki tabu döngü sayısına eşit tutulmuştur.

Bütün bu işlemlerden sonra tabu aramayla oluşturulan bireyin tekrar nüfusa konulması gerekmektedir. Elde edilen birey sonuç olarak kötü olan ebeveynin yerini yalnızca daha iyiyse ya da ona eşitse alabilir. Bunun sebebi algoritmanın ileriki aşamalarında elde edilen iyi bireyleri korumaktır.

ALGORİTMANIN SONLANDIRILMASI

Algoritmanın sonlandırılması iki şarta bağlıdır. Bunlar nüfustaki en iyi bireyin kenar çakışım sayısının sıfıra inmesi ya da üst sınır olarak belirlenen nesil sayısına ulaşılmasıdır. Bu üst sınır farklı çizgeler için farklı şekilde belirlenmiştir.

GENETİK VE TABU ARAMANIN AĞIRLIĞI

Pek çok melez algoritmadaki en önemli hususlardan biri bileşenlerin hangi ağırlıkta aramayı oluşturacağıdır. Bu algoritma genetik ve tabu arama ağırlığının yaklaşık olarak eşit önemde olmasını hedeflemiştir. Algoritmanın sonuçtan uzak olunan ilk aşamalarında genetik uzmanlardan sonra gerçekleştirilen tabu aramanın uzunluğu kısa tutulmuş, buna nazaran sonuca yakınsanan aşamalarda tabu uzunluğu giderek arttırılmıştır. Bu ayarlama yerel arama yöntemi bölümündeki anlatılan komşuluk uzunluğuna benzer bir şekilde aşağıdaki formülle gerçekleştirilmiştir:

$$\text{TabuDöngüSayısı} = k + l \left(1 - \alpha \frac{\text{ToplamCeza}}{\text{ToplamKenar}}\right)^n$$

Denklem 5 – Tabu arama yineleme hesaplanması

Bu formülde k en düşük tabu uzunluğunu, l ise hedef alınan en yüksek tabu döngü uzunluğunu belirtir. Algoritma sonuçtan uzakken, ikinci terim sıfıra yaklaşmakta, dolayısıyla gayet kısa bir tabu döngüsü sağlanmaktadır. Sonuca yaklaşıırken ikinci terim 1'e yakınsayarak, ihmal edilebilir düzeyde olan k ile birlikte l adet tabu döngüsünü sağlamış olur.

DENEY SONUÇLARI

Bu bölüm ETA algoritmasının deney sonuçlarına ayrılmıştır. Öncelikle kullanılan deney örnekleri hakkında bilgi verilmiş daha sonra parametrelerin nasıl ayarlandığı anlatılmıştır. Deney sonuçları verildikten sonra ETA'nın performansı hakkında yorumlar yapılmıştır.

DENEY ÖRNEKLERİ

Algoritmaların karşılaştırılması için 1993 yılında düzenlenen 2. DIMACS yarışmasında kullanılan deney örnekleri kullanılmıştır. Bu deney örneklerinin en zor olanlarının kenar yoğunluğu 0.5 ile 0.7 arasında değişen çizgeler olduğu görülmüştür. Dolayısıyla deneyler aşağıdaki çizge örnekleri üzerinde yoğunlaştırılmıştır.

DSJC Çizgeleri: Bu çizgeler Johnson [11] tarafından benzetimli tavlama yönteminin çizge boyamaya uygulanması sırasında yaratılmış rasgele çizgelerdir. DSJC örnekleri 125, 250, 500 ya da 1000 düğümlü, kenar yoğunlukları ise 0.1, 0.5 ve 0.9 olan çizgelerden oluşmaktadır. Bu çizgelerin kromatik sayıları bilinmemektedir.

LEIGHTON Çizgeleri: Leighton [15] tarafından kromatik sayısı belli olacak şekilde sistematik bir şekilde yaratılan bu çizgeler 450 düğümden oluşmaktadır. Kromatik sayıları sırasıyla 5, 15 ve 25'tir.

Bu örneklerin dışında daha küçük ya da en iyi şekilde boyanması daha kolay olan diğer çizge örnekleri de test edilmiştir.

ETA PARAMETRELERİ

ETA algoritmasının en önemli parametreleri aşağıda sıralanmıştır:

- Nüfustaki bireylerin sayısı
- Her melezlemeden sonra gerçekleştirilen Tabu Aramanın en yüksek döngü sayısı

Genelde küçük nüfusların ve yüksek tabu döngü sayısının kullanıldığı diğer melez algoritmalara göre bu çalışmada daha büyük bir nüfus ve daha kısa tabu döngü sayısının etkileri üzerinde durulmuştur. Bu yüzden özellikle kolay örnekler için birey sayısı en az 5 veya 10'a, daha zorları için sırasıyla 25'e ve 50'ye çıkarılmıştır.

Küçük tabu döngü sayılarının nüfustaki çeşitliliği bir nebze düşürdüğü bildirilmiştir [5]. Buna rağmen tabu döngü sayısı için küçük değerler seçilerek, daha büyük nüfuslar kullanılmış ve çeşitlilik bu şekilde sağlanmaya çalışılmıştır. ETA'da tabu döngü sayıları olarak sırasıyla 100, 250 ve 500 değerleri kullanılmıştır.

Diğer önemli bir nokta ise tabu arama ile genetik algoritmaya hangi oranlarda ağırlık verileceğidir. Bunun için en az yineleme sayısı k , 10 olarak, ceza ağırlığı olarak kullanılan α katsayısı ise 5 olarak belirlenmiştir. Gerçek tabu döngü sayısının değişim miktarı ise yukarıda anlatılan en yüksek tabu döngü sayısına, kısacası l değerine bağlıdır. Denklem 5'te bahsedilen n değeri için en iyi 3 değerinin iyi sonuçlar verdiği görülmüştür.

Diğer önemli parametreler ise tabu arama sırasında kullanılacak olan komşuluğun genişliğiyle ilgilidir. Tabu uzunluğuna benzer bir şekilde ceza ağırlık katsayısı α için 3 değeri kullanılmış, komşuluğun daraltılması için, Denklem 3' te bahsedilen p değerleri olarak 4 yada 5 kullanılmıştır.

Son olarak tabu listesine yerleştirilen hamlelerin ne kadar bir süre için yasak olduğunu belirlemek için Denklem 4'te kullanılan μ ve ν parametreleri sırasıyla 10 ve 0.6 olarak ayarlanmıştır. Daha küçük örnekler içinse bu parametre tabu döngü sayısına eşit tutulmuştur.

Çizge Adı	Düğüm Sayısı	Kenar Sayısı	Kromatik Sayı	Elde Edilen Min. Renk	t (sn.)	Std.	Genetik Yineleme Sayısı	Std.	Nüfus Boyu	Max. Tabu Döngü
dsjc125.1	125	1472	?	5	0.31	0.13	29.5	11.28	5	1000
dsjc125.5	125	7782	?	18	0.90	0.41	62.8	30.68	25	250
dsjc125.9	125	13922	?	44	3.51	1.58	196.6	104.74	25	250
dsjc250.1	250	6436	?	9	0.48	0.21	23.7	10.36	25	250
dsjc250.5	250	31336	?	29	42.11	10.2	716	245	50	250
dsjc250.9	250	55974	?	77	4.13	1.54	51.6	25.27	25	250
dsjc500.1	500	24916	?	14	21.47	7.01	100.7	33.94	50	500
le450_5a	450	5714	5	5	20.75	5.89	219.6	42.74	50	1000
le450_5b	450	5734	5	5	24.44	6.67	230.8	44.58	50	1000
le450_5c	450	9803	5	5	4.84	2.10	64	19	50	1000
Le450_5d	450	9757	5	5	5.58	2.19	75.8	25.01	50	1000
Le450_15a	450	8168	15	16	17.94	10.22	249.4	147.34	5	500
Le450_15b	450	8169	15	16	13.39	4.82	139.6	50.82	5	500
Le450_25a	450	8260	25	25	0.74	0.49	10.9	7.2	10	500
Le450_25b	450	8263	25	25	0.18	0.15	3.1	2.58	5	500

Tablo 1 : DSJC ve Leighton Çizge Boyama Sonuçları

DENEYLER

Deneyler için C++ diliyle yazılan algoritma Tablo 1 'deki çizge örnekleri için Pentium 4 2Ghz işlemciye ve 256Mb belleğe sahip 4 farklı kişisel bilgisayarda test edilmiştir. Tablo 2'deki çizgeler ise AMD Duron 1300 Mhz işlemcili ve 256Mb belleğe sahip başka bir kişisel bilgisayarda test edilmiştir. Tablo 1 Leighton ve DSJC çizge sonuçlarını, Tablo 2 ise diğer çizgelerin sonuçlarını göstermektedir. Tablo1'deki her çizge için 10 deneme yapılmış bunların ortalaması alınmıştır. Tablo 2'deki her çizge için deneme sayısı 50'dir.

Tablolarda öncelikle her çizge ile ilgili bilgiler (düğüm, kenar ve kromatik sayıları) verilmiştir. Daha sonra elde çizgenin boyanabildiği minimum renk sayısı verilmiştir. Algoritmanın bitmesi için geçen süre, bu sürenin standart sapması ise saniye cinsinden verilmiştir. Bu süreye çizgenin diskten okunması dahil değildir. Daha sonra gerçekleşen genetik yineleme sayısı (melezleme sayısı) ve bunun standart sapması verilmiştir. Tablo 1'de daha sonra nüfusun büyüklüğü verilmiştir. Tablo 2'deki çizgelerin hepsi için nüfusun büyüklüğü 5 olarak belirlenmiştir. Daha sonraki sütunda (melezlemeden sonra yapılabilecek) en yüksek (Max.) tabu döngü sayısı verilmiştir. Tablo 1'deki deneylerde tabu zaman aşımı süresi dinamik olarak hesaplanmıştır.

DENEY YORUMLARI

Özellikle zor çizgeler üzerinde yapılan deneylerin sonucunda (Bkz. Tablo 1), ETA'nın gayet verimli bir şekilde çalıştığı görülmektedir. Bu örneklerde yaklaşık olarak literatürde bulunan en iyi renk değerlerinden (kromatik sayılardan) en fazla 1 yada 2 renk uzak kalınmıştır. ETA, daha küçük ve kolay deney örneklerinin

(bkz Tablo 2) hemen hemen hepsinde rahat bir şekilde doğru kromatik sayı değerlerine ulaşarak, bütün denemelerden 100% başarıyla çıkmıştır.

Deneylerde özellikle üzerinde durulan noktalardan biri genetik algoritma ve tabu arama yöntemlerine verilen ağırlıktır. Deneyler sonucunda belli seviyedeki tabu arama döngüsünden sonra sonucun pek fazla değişmediği, buna nazaran düşük döngü nüfustaki çeşitliliği bir nebze düşürdüğü gözlenmiştir. Özellikle zor çizgelerde nüfus çeşitliliğinin azalması aramanın yerel bir en iyi noktada takılıp kalmasına yol açmıştır. Birey sayısının artırılmasının ise buna çözüm sağladığı görülmüştür. Birey sayısının artırılması aynı zamanda algoritmanın kararlılığını da artırmaktadır. Özellikle küçük nüfuslarda elde edilen yineleme ve zaman sürelerinin standart sapmalarının yüksek olduğu gözlenmiştir. Nüfusun büyütülmesi bu süreleri düzenli bir şekilde azaltmıştır. Buna nazaran özellikle küçük örneklerde (düğüm sayısı < 100) küçük nüfusların kullanılıp (2 < nüfus sayısı < 5) tabu aramanın artırılmasının en hızlı sonuçları verdiği görülmüştür.

Büyük deney kümelerinde ayrıca tabu arama ağırlıklı deneyler de yapılmıştır. (Çok büyük tabu uzunlukları, çok küçük nüfuslar, vs). Elde edilen sonuçlar tabu arama yönteminin belli bir seviyeden sonra daha fazla gelişim gösteremediğinden bir yerel noktada tıkanıp kaldığını göstermektedir. Melezleme ve mutasyon uzmanları eklendiğinde ise bu noktalardan kurtulma olasılığının arttığı gözlenmektedir. Bunun sebebi melezleme ve mutasyon uzmanlarının aramanın belli bir yere takılmasını engelleyip çözüme yakın başka bir yerden devam etmesini sağlamasıdır.

Elde ettiğimiz bu sonuçlar melez algoritmaların Çizge Boyama Problemi için son derece başarılı olabileceğini göstermiştir.

Çizge Adı	Düğüm Sayısı	Kenar Sayısı	Kromatik Sayı	Elde Edilen Min. Renk	t (s)	Std.	Genetik Yineleme Sayısı	Std.	Max. Tabu Döngüsü
miles250	128	774	8	8	0.038	0.004	2.26	0.95	50
miles500	128	2340	20	20	0.081	0.082	3.02	2.25	100
miles750	128	4226	31	31	0.093	0.082	7.82	6.60	250
miles1000	128	6432	42	42	0.377	0.615	9.88	18.50	500
miles1500	128	10396	73	73	0.209	0.153	4.04	4.59	250
mulsol.i.1	197	3925	49	49	0.29	0.528	8.96	21.57	250
mulsol.i.2	188	3885	31	31	0.337	0.417	17.94	22.27	250
mulsol.i.3	184	3916	31	31	0.571	0.824	31.3	44.46	250
mulsol.i.4	185	3946	31	31	0.504	0.703	26.74	36.24	250
mulsol.i.5	196	3973	31	31	0.427	0.777	23.14	43.21	250
anna	138	986	11	11	0.003	0.005	1.30	0	50
david	87	812	11	11	0.003	0.004	2.06	1.83	50
huck	74	602	11	11	0.001	0.003	1	0	50
homer	561	3258	13	13	0.086	0.065	2.56	1.43	50

Tablo 2 : Diğer DIMACS Çizge Boyama Sonuçları

SONUÇLAR VE İLERİDE YAPILACAK ÇALIŞMALAR

Bu bildiriye çizge boyama için yeni bir melez algoritma önerilmiştir. Probleme uygun şekilde hazırlanan bir melezleme ve mutasyon uzmanı denenmiştir. Ayrıca tabu arama yöntemi için dinamik olarak değişen yineleme ve komşuluk daraltma sistemleri geliştirilmiştir. Çizge boyama için önerilen diğer melez algoritmalarda kullanılan küçük nüfusların yanısıra büyük nüfuslarda kullanılmıştır. Özellikle küçük ve orta boy çizge örneklerinde başarılı sonuçlar elde edilmiştir.

Yine de algoritmanın geliştirilmesi için daha yapılabilecek çalışmalar mümkündür. Algoritmanın özellikle daha büyük deney örnekleri üzerinde nasıl bir davranış göstereceği incelenecektir. Bu özellikle pek çok parametrenin en iyi bir şekilde ayarlanmasına yardımcı olacaktır. Önerilen algoritmada ki parametrelerin sayısını düşürmek için bunlardan bazılarının arama sırasındaki durumlara göre uyum sağlayıp kendiliğinden değişmesi de amaçlanmaktadır. Son olarak tabu arama yönteminde daha değişik hamleler kullanılması hedeflenmektedir.

KAYNAKÇA:

- [1] Brelaz, D., 1979. "New methods to color vertices of a graph", Communications of ACM 22, 251-256
- [2] Davis, L., 1991. "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New York.
- [3] Falkenauer, E., 1996. "A Hybrid Grouping Genetic Algorithm for Bin Packing", Journal of Heuristics 1996;2(1):5-30.
- [4] Fleurant, C., and Ferland, J.A., 1996. "Object oriented implementation of heuristic search methods for graph coloring, maximum clique, and satisfiability", Proceedings of the 2nd DIMACS implementation challenge", Volume 26 of DIMACS Series.
- [5] Galinier, P. and Hao, J.K., 1999. "Hybrid evolutionary algorithms for graph coloring.", Journal of Combinatorial Optimization. 3(4): 379-397.
- [6] Garey, M.R, Johnson, D.S., 1979. "Computers and Intractability", Freeman, San Francisco.
- [7] Glover, F., 1989. "Tabu Search – Part I", ORSA Journal of Computing Vol. 1, No. 3.

- [8] Glover, F., 1990. "Tabu Search – Part II", ORSA Journal of Computing Vol. 2, No. 1.
- [9] Hertz, A., de Werra, D., 1987, "Using tabu search techniques for graph coloring". Computing 39: 345-351.
- [10] Holland, J. H., 1975. Adaptation in Natural and Artificial Systems, Univ. Mich. Press.
- [11] Johnson D.S., Aragon C.R, McGeoch L.A, and Schevon C., 1991. "Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning.", Operations Research, 39(2):378-406, 1991.
- [12] Kirkpatrick S., Gelatt C.D., Jr., Vecchi M.P., 1983. "Optimization by Simulated Annealing", Science, Number 4598.
- [13] Kirovski, D., Potkonjak, M., 1998. "Efficient Coloring of a Large Spectrum of Graphs", Proc. of ACM/IEEE 35th Design Automation Conference, pp. 427-432, San Francisco, USA.
- [14] Klotz, W., 2002. "Graph coloring algorithms", Mathematik-Bericht 2002/5, TU Clausthal.
- [15] Leighton, F. T., 1979. A graph coloring algorithm for large scheduling problems, Journal of Research of the National Bureau of Standards, 84:489-506,
- [16] González-Velarde J.L., and Laguna M., "Tabu Search with Simple Ejection Chains for Coloring Graphs", To appear in the Annals of Operations Research.