



NECMETTİN ERBAKAN ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Evrimsel Hesaplamaya Giriş Dönem Projesi

Nihâl DEMİR – 19010011016

Edanur ERGENÇ – 19010011050

1. Graf Renklendirme Problemi

Graf, düğüm olarak adlandırılan noktalar (node/vertice) ve her biri bu noktaları veya sadece noktanın kendisini birleştiren ve ayır/kenar (edge) olarak adlandırılan çizgiler topluluğudur.

Graf renklendirme, graf üzerinde birbirine komşu olan düğümlere farklı renk atama işlemidir. Amaç, en az sayıda renk kullanılarak tüm düğümlere komşularından farklı birer renk vermektir. Kenar renklendirme, toplam renklendirme, etiketsiz renklendirme gibi türleri mevcuttur. Renklendirmede kullanılan toplam renk sayısı kromatik (chromatik) sayı olarak adlandırılır. Genellikle $\chi(G)$ ile gösterilir. Graf renklendirme problemi bir Constraint Satisfaction Problem ve NP-complete problemidir [1]. Kromatik sayı bulma işlemi NP-Hard problemi olarak bilinmektedir.

Graf renklendirme bilgisayar biliminde ve günlük yaşamdaki birçok problemin çözümünde kullanılan bir yaklaşımdır. Sıkça kullanıldığı bilinen alanlardan bazıları şu şekildedir [2]:

- ❖ Program / Zaman Çizelgesi Oluşturma
- ❖ Görüntü Segmentasyon Yaklaşımı
- ❖ Mobil Radyo Frekans Tahsisi
- ❖ Harita Renklendirme
- ❖ Küme Analizi
- ❖ Kayıt Tahsisi
- ❖ Sudoku

2. Graf Renklendirme Probleminin Genetik Algoritma ile Çözümü

2.1. Veri Seti

Bu çalışmada DIMACS standart biçiminde hazırlanan Myciel 3-4-5 veri setleri [3] kullanılmıştır. Veri setlerinin özellikleri tabloda verilmiştir.

Özellikler			
Veri seti	Düğüm s.	Kenar s.	Best/ $\chi(G)$
Myciel3	11	20	4
Myciel4	23	71	5
Myciel5	47	236	6

2.2. Kodlama – Temsil

Renklendirme probleminin doğası renklerin ne olduklarına değil, renklerin sayısına bağlıdır. Graf renklendirme probleminin çözümü için, matematiksel ve bilgisayar temsillerinde ‘renk’ olarak pozitif ve negatif tamsayılar kullanılmaktadır. Bu bilgilerden yola çıkılarak bireylerin temsil şekli tamsayı temsil şekli olarak seçilmiştir. Programın başında tamsayılardan oluşan bir renk listesi tanımlanmıştır. Bu listeden grafta bulunan düğümlere sayılar ile temsil edilen renkler atanarak bireyler oluşturulmuş ve 50 bireylik

başlangıç popülasyonu elde edilmiştir. 11 düğümden oluşan myciel-3 grafi için tamsayı temsil şekli ile oluşturulmuş örnek bir birey aşağıdaki şekildedir:

[6 2 10 6 9 6 10 7 3 10 3]

2.3. Fitness & Penalty

Graf renklendirme probleminin amacı, en az sayıda renk kullanılarak tüm düğümlere komşularından farklı birer renk vermektir. Problemin genetik algoritma ile çözümünde düğümler üzerinden karşılaştırma yapılarak her birey için kullanılan renk sayısı hesaplanmış ve bu değerler fitness değeri olarak kullanılmıştır. Problem aynı zamanda komşu düğümlerin aynı renkte olmaması kısıtını içerdiğinden, bu kısıtın kontrolünü sağlamak için bir penalty değeri tanımlanmıştır. Komşu düğümler kenarlar listesi üzerinden kontrol edilmiş, her bir kısıt ihlalinde penalty değeri artırılarak kısıt kontrolü sağlanmıştır.

2.4. Seçim Yöntemi

Bireylerden fitness değerleri aranan kriterlere yakın olan, yani başarılı olanlar bir seçim yöntemi ile seçilir ve bireyler ebeveyn olarak kabul edilerek aralarında üreme meydana getirilir [4]. Graf renklendirme problemi aynı zamanda bir kısıt problemi olduğundan, seçim aşamasında fitness değeri ile birlikte penalty değerleri de göz önünde bulundurularak en verimli şekilde seçim işleminin gerçekleştirilmesi amaçlanmıştır. Seçim işleminin daha kaliteli sonuçlar vermesi için öncelik penalty değeri sıfır olan bireylerin seçilmesi ve “feasible” yani kullanılabilir çözümler elde etmektir.

Bu doğrultuda, ebeveyn seçimi aşamasında rastgele belirlenen 2 aday arasında aşağıdaki karşılaştırmalar yapılarak seçim işlemi gerçekleştirilmiştir:

- ❖ Bireylerin her ikisinin de penalty değeri sıfır ise, fitness değerleri hesaplanır ve fitness değeri minimum olan birey seçilir.
- ❖ Bireylerden yalnızca birinin penalty değeri sıfır ise, penalty değeri sıfır olan birey seçilir.
- ❖ Bireylerden her ikisinin de penalty değeri sıfırdan büyük ise, penalty değerleri karşılaştırılarak daha küçük olan birey seçilir.

2.5. Çaprazlama

Çaprazlama, ebeveynlerin bazı genlerini yeni bireyler üzerine kopyalamaları işlemidir [4]. Çaprazlama operatörü, mevcut gen havuzunun potansiyelini artırmak üzere kullanılmaktadır. Bu çalışmada çaprazlama yöntemi olarak tek nokta çaprazlama tercih edilmiştir. Tek nokta çaprazlama yöntemine göre bir önceki aşamada seçilen ebeveynler

üzerinden rastgele bir gen belirlenerek yeni bireyin bu genden önceki genlerini birinci ebeveynden, kalan genlerini ise ikinci ebeveynden alması sağlanmıştır.

```
Parent1: [ 1  2  8  9  5  8 10  1  4  5  7]
Parent2: [ 2  8  2  2  9  3  6  5 10  8  1]
cross_over point: 5
child1: [ 1  2  8  9 | 9  3  6  5 10  8  1]
child2: [ 2  8  2  2 | 5  8 10  1  4  5  7]
```

2.6. Mutasyon

Çaprazlama işleminden sonra oluşan yeni bireylerin ebeveynlerinin birebir kopyası olması olasılığını önlemek ve türdeki çeşitliliği artırmak amacıyla bireyler mutasyona uğratılır [4]. Bu aşamada mutasyon oranı 0,3 olarak seçilmiştir. Seçilen mutasyon oranına göre bireyin her bir geni için 0-1 aralığında rastgele bir sayı üretilerek mutasyon oranından küçük olup olmadığı kontrol edilmiş, mutasyon oranından küçük bir değer üretilen genler renk havuzundan farklı bir renk ile değiştirilerek mutasyona uğratılmıştır.

```
mutatedGen: 5
ind_mutated: 8
randColor: 2
mutatedGen: 7
ind_mutated: 1
randColor: 7
[10  8  7  5 | 8  5 | 1  1  6  9  5]
[10  8  7  5 | 2  5 | 7  1  6  9  5]
```

2.7. Yeni Neslin Oluşturulması

Başlangıç popülasyonu ve aynı sayıda üretilen yeni bireyler fitness ve penalty değerleri hesaplanarak sıralanmıştır. Yeni neslin seçiminde iki farklı yöntemden faydalanılmıştır:

- ❖ Birinci yöntemde her bir birey için hesaplanan fitness ve penalty değerleri 0-1 aralığında normalize edilmiş ve 40'a 60 oranında ağırlıklandırılmıştır. Elde edilen değerler toplanarak yeni bir uygunluk değeri gibi küçükten büyüğe sıralanarak bir önceki popülasyon ve yeni popülasyondan oluşan 100 bireylik listeden en iyi ilk 50 birey seçilmiştir.
- ❖ İkinci yöntemde 2.4'te ebeveyn seçimi için gerçekleştirilen seçim yöntemi kullanılarak sıralanan 100 bireylik popülasyondan en uygun 50 birey seçilmiştir.

2.8. Sonlandırma Kriteri

Sonlandırma kriteri kromatik sayıya ulaşma durumu olarak belirlenmiştir. Sonlandırma kriterine ulaşamadığı durumlarda 10.000 fitness değerine ulaşıldığında program sonlanmaktadır.

3. Sonuç

Bu projede graf renklendirme problemi genetik algoritma kullanılarak çözülmüştür. Proje sonunda 2 farklı nesil oluşturma yöntemi için her veri seti 15 kez çalıştırılmış ve sonuçları kaydedilmiştir.

- ❖ Her bir birey için hesaplanan fitness ve penaltı değerlerinin 0-1 aralığında normalize edilip ağırlıklandırılarak sıralandığı birinci yöntem için sonuçlar aşağıdaki gibidir:

MYCIEL 3 ($\chi(G) = 4$)				
Çalıştırma	Birey	Fitness	Penaltı	İterasyon
1	[9 5 2 5 9 3 5 3 1 9 2]	5	0	180
2	[10 8 2 4 10 10 8 4 4 8 1]	5	0	109
3	[3 1 3 8 1 3 9 3 8 1 5]	5	0	75
4	[1 7 10 7 1 6 4 4 10 1 7]	5	0	153
5	[2 5 2 7 1 2 5 2 7 5 1]	4	0	61
6	[10 9 10 9 8 10 8 10 9 8 6]	4	0	197
7	[6 10 8 8 10 6 2 6 8 6 10]	4	0	9
8	[3 10 5 8 2 5 2 5 10 3 8]	5	0	103
9	[10 3 10 3 2 10 6 6 4 2 3]	5	0	99
10	[6 8 4 1 5 6 8 6 1 8 5]	5	0	0
11	[4 5 1 1 4 4 8 8 8 4 5]	4	0	60
12	[8 6 8 6 9 5 5 8 5 5 6]	4	0	141
13	[2 9 1 9 2 1 8 8 1 8 5]	5	0	90
14	[1 5 2 2 3 1 5 2 2 3 7]	5	0	178
15	[5 7 5 10 1 1 1 10 10 7 5]	4	0	50

MYCIEL 4 ($\chi(G) = 5$)				MYCIEL 5 ($\chi(G) = 6$)			
Çalıştırma	Fitness	Penaltı	İterasyon	Çalıştırma	Fitness	Penaltı	İterasyon
1	8	0	147	1	10	7	104
2	9	0	82	2	9	12	54
3	9	0	19	3	10	9	156
4	9	0	184	4	9	13	129
5	9	0	26	5	9	13	23
6	9	0	40	6	9	13	199
7	8	0	73	7	10	8	182
8	8	0	32	8	9	10	49
9	8	1	75	9	10	10	79
10	9	0	81	10	10	9	170
11	9	0	45	11	9	11	178
12	8	1	3	12	10	11	29
13	9	0	21	13	10	11	89
15	8	0	39	14	9	12	98
15	8	0	61	15	9	12	193

- ❖ 2.4'te gerçekleştirilen seçim yöntemi kullanılarak sıralanan popülasyondan en uygun bireylerin sıralanarak seçildiği ikinci yöntem için sonuçlar aşağıdaki gibidir:

MYCIEL 3 ($\chi(G) = 4$)				
Çalıştırma	Birey	Fitness	Penaltı	İterasyon
1	[5 9 5 9 10 5 9 5 9 10 6]	4	0	24
2	[7 6 7 8 9 7 2 7 8 6 9]	5	0	154
3	[10 4 10 8 4 2 4 2 8 2 10]	4	0	97
4	[5 3 9 4 3 5 3 7 4 5 9]	5	0	199
5	[7 3 10 10 7 9 9 4 4 3 7]	5	0	20
6	[9 5 1 10 9 1 10 8 8 8 5]	5	0	92
7	[1 7 9 5 1 1 5 5 5 1 9]	4	0	175
8	[3 2 6 6 3 3 9 10 10 9 2]	5	0	62
9	[8 3 8 2 3 5 2 5 2 5 8]	4	0	61
10	[8 7 3 3 8 8 5 10 10 7 3]	5	0	21
11	[1 8 7 7 1 1 10 10 10 8 7]	4	0	130
12	[2 7 2 8 3 3 7 9 8 7 2]	5	0	1
13	[2 3 10 10 6 2 3 10 10 2 6]	4	0	63
14	[3 8 6 7 8 3 7 5 5 5 8]	5	0	104
15	[3 9 8 8 3 3 1 8 1 1 9]	4	0	74

MYCIEL 4 ($\chi(G) = 5$)				MYCIEL 5 ($\chi(G) = 6$)			
Çalıştırma	Fitness	Penaltı	İterasyon	Çalıştırma	Fitness	Penaltı	İterasyon
1	9	1	50	1	10	7	98
2	9	1	179	2	10	8	137
3	8	1	99	3	10	9	123
4	9	1	87	4	10	8	113
5	8	1	196	5	10	7	51
6	9	1	198	6	10	9	21
7	9	1	158	7	10	8	48
8	9	1	199	8	10	6	80
9	8	1	102	9	10	8	134
10	8	1	7	10	10	6	199
11	9	1	84	11	10	8	17
12	8	1	3	12	10	9	173
13	9	1	32	13	10	3	42
15	9	1	61	14	10	7	189
15	8	1	177	15	10	8	85

KAYNAKÇA

- [1] Hindi, M., & Yampolskiy, R.V. (2012). Genetic Algorithm Applied to the Graph Coloring Problem. Midwest Artificial Intelligence and Cognitive Science Conference.
- [2] MORA-GUTIERREZ, Roman Anselmo et al. A hybrid algorithm for the robust graph coloring problem. Rev. Mat [online]. 2016, vol.23, n.2, pp.421-444. ISSN 1409-2433.
- [3] <https://mat.tepper.cmu.edu/COLOR/instances.html>
- [4] Yaman, F., & Yılmaz, A.E. Genetik Algoritmada Tek ve Çok Noktalı Çaprazlama Tekniklerinin Doğrusal Anten Dizisi Sentez Probleminde İncelenmesi.