

Evrimsel Hesaplama Ödev

Öğr.Gör.Şevket Umut ÇAKIR
CENG 415 - Evrimsel Hesaplama

4 Aralık 2018

Amaç: 0-1 Sırt çantası(knapsack) problemini çözmek için istenilen dilde bir genetik algoritma programı oluşturmak. Programlama dili seçimi öğrenciye aittir.

Kullanıcıdan alınan aşağıdaki değerlere ve verilen algoritmaya göre genetik algoritma kodlanacaktır. Özellik ve işlemlerin detayları aşağıda verilmektedir.

BAŞLA

Başlangıç popülasyonunu rastgele oluştur (INITIALISE)

Her adayı değerlendir (EVALUATE)

Bitiş koşulu sağlanana kadar TEKRARLA(REPEAT)

1 Ebeveynleri seç (PARENT SELECT)

2 Ebeveynleri çaprazla (RECOMBINE)

3 Yavruları mutasyona tabi tut (MUTATE)

4 Yeni adayları değerlendir (EVALUATE)

5 Sonraki nesile hayatta kalacakları
seç (SURVIVOR SELECT)

BİTİR

Listing 1: Genetik Algoritma Yapay Kodu

Temsil: 0-1 sırt çantası probleminde taşıyabileceği en büyük ağırlık(W) belli olan bir sırt çantası bulunmaktadır. Farklı ağırlık(w) ve değere(v) sahip olan n tane de eleman bulunmaktadır. Sırt çantası problemindeki amaç çantaya sığacak şekilde en fazla değere sahip olan elemanları almaktır. Sırt çantası probleminde bir birey(B) bit dizisi olarak temsil edilir:

$$B = (b_1, b_2, \dots, b_n), \quad b_i \in \{0, 1\} \quad (1)$$

Bir bitin deęerinin 0 olması o elemanın antada bulunmadığını, 1 olması ise antada bulunduęunu gsterir.

Rastgele Listesi: Oluřturulan gzmlerin yeniden retilenebilmesi iin kullanılacak listedir ve kullanıcıdan alınacaktır. Her yeni rastgele sayı retiminde listede sıradaki sayı kullanılacaktır. Liste sona geldiğinde bařtan devam edilecektir. Deęerlendirme listesindeki elemanlar $[0,1]$ aralıęında gerel sayılardır.

Initialise: Poplasyonu ilk deęerlendirme iřlemidir. İstenen poplasyon boyutuna gre yeni bireyler oluřturur. Her bir birey bit dizisi řeklinde temsil edilir.(9 elemanlı anta iin '100111010' gibi). Bit dizisinin uzunluęu antaya alınabilecek eleman sayısına eřittir ve 0 deęeri ilgili elemanın antaya alınmayacağını, 1 deęeri ise elemanın alınacağını belirtir. Yeni bir birey oluřtururken her bir bit iin rastgele bir deęer retilir. Bitin deęeri rastgele deęer 0.5'den kkse 0 aksi takdirde 1 olacaktır.

Evaluate: Deęerlendirme iřlemidir, bir bireyin uygunluk(fitness) deęerini belirler. Eęer elemanlar antaya sığıyorsa ($\sum_{i=1}^n w_i \cdot b_i \leq W$ ise) deęerlendirme sonucu toplam tutar($\sum_{i=1}^n v_i \cdot b_i$), sığmıyorsa sıfır(0) olacaktır.

Parent Select: Ebeveyn seimini temsil eder. Ebeveyn seiminde turnuva seim algoritması kullanılacaktır. Poplasyon iinden elemanlar seilirken retilen rastgele deęerler kullanılır. Seme iřlemi sırasında poplasyon boyutu 5 ise $[0, 0.2]$ birinci, $(0.2, 0.4]$ ikinci, $(0.4, 0.6]$ nc, $(0.6, 0.8]$ drdnc, $(0.8, 1]$ beřinci ebeveyni temsil eder. Rastgele bir deęerin(r), n elemanlı bir dizide hangi konuma denk geleceęi(i) Denklem 2'deki gibi hesaplanabilir.

$$i = \lceil r \cdot n \rceil - 1, \quad r \in [0, 1] \quad (2)$$

Turnuva sein algoritmasında seilecek λ elemanın her biri iin k adet birey poplasyondan rastgele olarak seilir ve kendi aralarında uygunluk(fitness) deęerine sıralanarak en iyisi alınır. Bu řekilde λ adet ebeveyn seilmiş olur.

Recombine: aprazlama iřlemini temsil eder. Ebeveyn havuzundan bireyler ikiřer ikiřer alınarak kendi aralarında aprazlanır. Tek noktalı aprazlama uygulanır. 5 elemanlı bir birey iin 0.2 aprazlama noktası ikinci elemandan itibaren aprazlamayı gerektirir. Her bir aprazlama sonucunda iki adet ocuk oluřur. aprazlama iřlemleri btn ebeveynler seildikten sonra yapılmalıdır.

Mutasyon: ocuklar zerinde bit evirme(bit flipping) mutasyonu uygulanır. Bakılan bit iin retilen rastgele deęer mutasyon oranından kkse 0 bit tersine evrilir. Mutasyon iřlemleri btn aprazlama iřlemleri bittikten

Ebeveyn 1	0	0	0	0	0	0	1	1	1	1	1	Yavru 1
X												
Ebeveyn 2	1	1	1	1	1	1	0	0	0	0	0	Yavru 2

Tablo 1: 0.2 rastgele değeri için örnek çaprazlama

sonra yapılmalıdır.

Survival Select: Hayatta kalanların seçilmesini temsil eder. Popülasyon sayısı(μ)'nın sabit kalması istendiği için yeni üretilen yavru(λ) popülasyona eklenir ve tüm popülasyondan en iyi μ tanesi seçilir. Hayatta kalanların seçimi için turnuva seçim algoritması kullanılmayacaktır.

Bitiş koşulu: Bitiş koşulu belirtilen maksimum iterasyon sayısına(nesile) ulaşmaktır.

Test Senaryoları: Yukarıdaki işlemleri belirtilen şekilde yapmanız durumunda algoritmanızın doğruluğunu test etmeniz için 5 farklı test senaryosu verilmiştir. Her bir senaryo için `testi.txt` ve `outi.txt` dosyaları bulunmaktadır. `testi.txt` dosyalarının içeriği aşağıdaki gibidir:

```
Rastgele listesi(virgülle ayrılmış)
Popülasyon boyutu
Turnuva eleman sayısı(k)
Mutasyon olasılığı[0,1]
İterasyon sayısı
Çanta boyutu
Elemanların ağırlıkları(virgülle ayrılmış)
Elemanların değerleri(virgülle ayrılmış)
```

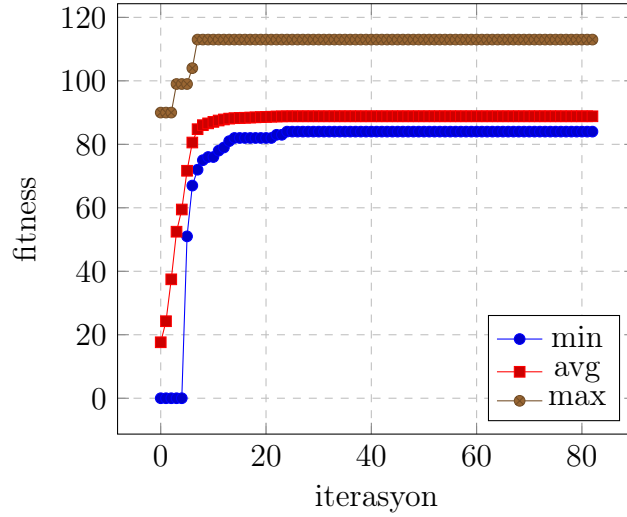
Listing 2: test.txt içeriği

Her bir test senaryosunun çıktısı `outi.txt` dosyalarında bulunmaktadır. `testi.txt` dosyalarındaki değerler aşağıdaki şekilde programınıza girdi olarak verilebilir. Programınız girdileri bu sırayla alabilmelidir.

```
$ uygulama < test1.txt
```

Uygunluk değeri grafikleri: Rapor içerisinde her bir test senaryosu için verilen girdilere göre uygunluk(fitness) değerinin nesiller(iterasyon) boyunca değişimini gösteren Şekil 1-5'teki gibi grafikler eklenmelidir. Grafikleri çizmek için Microsoft Excel ve LibreOffice gibi programları kullanabilirsiniz. Alternatif olarak <https://plot.ly> gibi çevrimiçi grafik çizme platformları da bulunmaktadır.

Şekil 1-5’de test senaryoları için beklenen uygunluk değeri grafikleri verilmektedir. Seçim algoritmasındaki uygunluk değeri eşit olan bireyler arasında gerçekleştirilen rastgele seçim her program için farklı olabilmektedir. Grafik değerleri birebir aynı olmayabilir fakat benzer olmalıdır.



Şekil 1: test1.txt için fitness değerleri

Rapor içeriği: Raporda kullanılan programlama dili, geliştirme ortamı(IDE) ve 5 farklı test senaryosu için eğitim süresi boyunca değişen uygunluk(fitness) değerlerinin grafikleri bulunmalıdır.

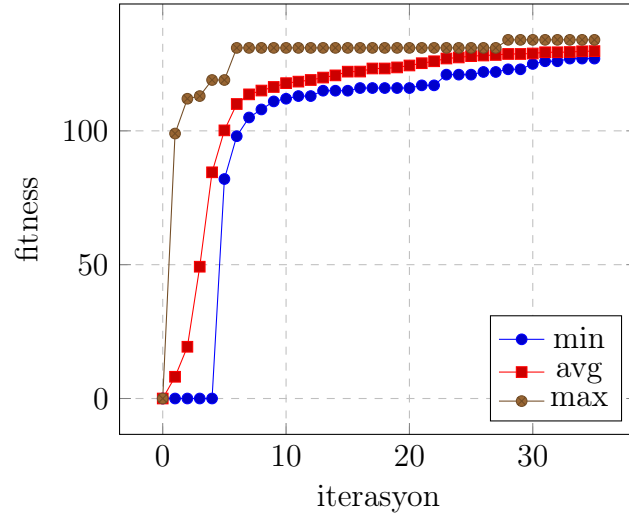
Önemli Tarihler :

Tablo 2: Önemli Tarihler

Olay	Tarih	Konum	Biçim
Ödev Teslimi	28.12.2018	EDS	kaynak kod dosyaları ve çıktıları gösteren rapor

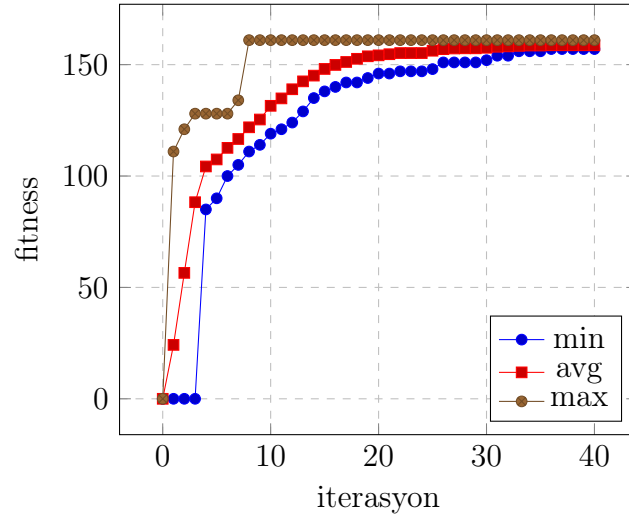
Ödev Teslimi ile İlgili Açıklamalar

- Ödevler Eğitim Destek Sistemi(<http://eds.pau.edu.tr/>) üzerinden sıkıştırılmış bir dosya olarak yüklenecektir.
- Projenin ve sıkıştırılmış dosyanın adı HWEH<öğrenci numarası> şeklinde olacaktır. Örneğin öğrenci numarası 11253699 olan bir öğrenci ödevini teslim ederken dosya ismi **HWEH11253699.zip** olacaktır.

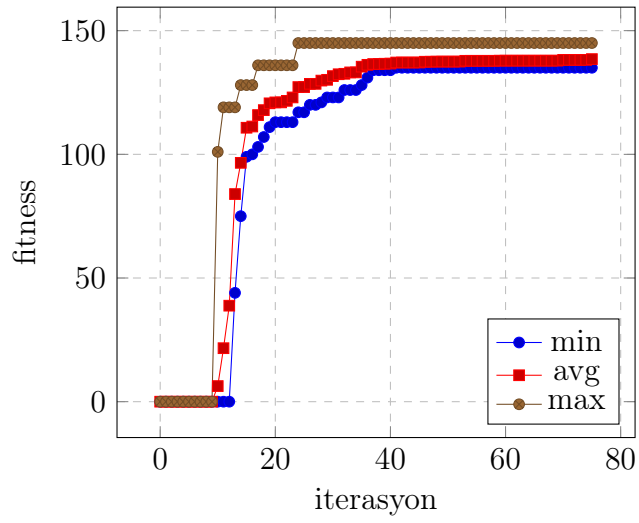


Şekil 2: test2.txt için fitness değerleri

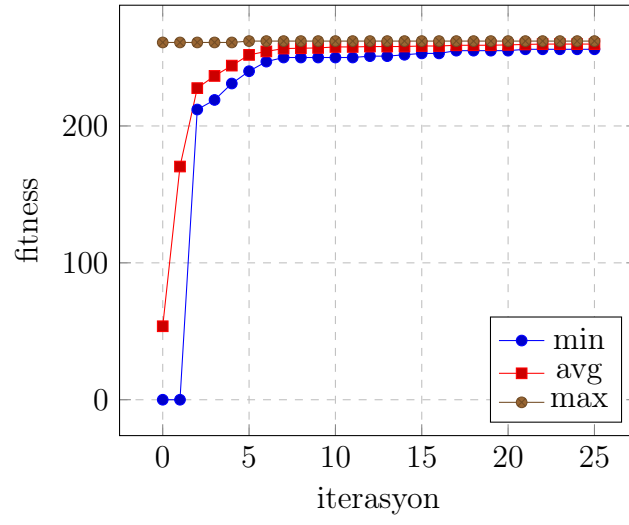
- Kaynak kod dosyasının en üstüne öğrenci numarası ve ad soyad açıklama satırı olarak eklenmek zorundadır.
- Ödev bireysel olarak cevaplanacaktır.
- Rapor dosyası PDF biçiminde olmalıdır.
- Sıkıştırılmış dosya biçimi zip veya tar.gz olmalıdır.



Şekil 3: test3.txt için fitness değerleri



Şekil 4: test4.txt için fitness değerleri



Şekil 5: test5.txt için fitness değerleri