

T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI

EVİRİMSEL ALGORİTMALAR
ve
UYGULAMALARI

HAZIRLAYANLAR:

Sedrettin ÇALIŞKAN

Ulaş DEMİRCİ

Zeki KUŞ

DANIŞMAN:

Yrd. Doç. Dr. Ebubekir KOÇ



2017 - İSTANBUL

İçindekiler Tablosu

İki Boyutlu Yerleştirme Probleminin Gazetedeiki Seri İlanlar Sayfasına Uygulanması	2
1.Giriş.....	2
2.Kullanılan Algoritmalar	2
2.1. Genetik Algoritma.....	2
2.2. Parçacık Sürü Optimizasyonu	4
2.3. Benzetilmiş Tavlama (Simulated Annealing)	6
3. Deneysel Sonuçlar	8
4. Sonuç	17
Gezgin Satıcı Probleminin Random Koordinatlar Üzerinde Uygulanması	20
1.Giriş.....	20
2.Kullanılan Algoritmalar	20
2.1. Karınca Sistem Algoritması	20
2.2. Genetik Algoritma	21
3. Deneysel Sonuçlar	23
4. Sonuç	26
Genetik Algoritmaların Farklı Çaprazlama Teknikleriyle Ders Programı Oluşturma Problemlerine Uygulanışı	28
1.Giriş.....	28
2.Genetik Algoritmalar	29
3. Çaprazlama Teknikleri ve Ders Programı Belirlenmesi	33
4.Farklı Çaprazlama Teknikleri İçin Elde Edilen Sonuçlar.....	35
5. Sonuçlar ve Öneri	38

İKİ BOYUTLU YERLEŐTİRME PROBLEMİNİN GAZETEDEKİ SERİ İLANLAR SAYFASINA UYGULANMASI

Zeki KUŐ

Mühendislik ve Fen Bilimleri Enstitüsü, Fatih Sultan Mehmet Vakıf Üniversitesi, Beyođlu, İstanbul,
zeki.kus@stu.fsm.edu.tr

ÖZET

Bu çalışmada, iki boyutlu yerleőtirme probleminin gazetelerin seri ilanlar bölümüne uygulanması anlatılmaktadır. Gazetelerin seri ilan sayfaları birden fazla ilan içermektedir. Bu ilanların sayfa içerisine en az fire ile yerleőtirilmesi ve yerleőtirilen ilanlardan maksimum kar elde etmek temel amaçtır. Probleme yönelik çözümler üretmek için bu çalışmada 3 farklı algoritma kullanılmıştır. Genetik algoritma (Genetic Algorithm) , Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) ve Benzetilmiş Tavlama (Simulated Annealing) gibi üç farklı yaklaşım problemin çözümünde kullanılmış ve sonuçlar çalışma içerisinde tablo halinde verilmiştir. Sonuçlara bakıldığında bu üç farklı yaklaşımın birbirine benzer sonuçlar verdiği ve bu problem için iyi birer alternatif oldukları görülmektedir.

Anahtar kelimeler: Genetik algoritma, benzetilmiş tavlama, parçacık sürü optimizasyonu, iki boyutlu yerleőtirme, evrimsel algoritma

1. GİRİŐ

Gazete ilanları reklam düzeni, bir dizi reklamların bir gazete sayfasına yerleőtirilmesi problemini ele alır. Hem yerleőtirilen reklam sayısını hem de sayfanın kullanılmayan kısmının faydasının optimize edilmesini amaçlar.

Bu, iki boyutlu (2D) kutu paketleme problemleriyle ilgili olan kombinatoryal bir optimizasyon problemidir. Reklamların yerleőtirileceđi sayfalar ve sayfalara yerleőtirilecek reklamlar dikdörtgen kutu biçiminde kabul edilmektedir. İki boyutlu kutu paketleme problemlerinde genel olarak genetik algoritma çözümleri uygulanmaktadır. Bu tip problemler NP-Hard problemler kümesine girmektedir. Problem temel olarak ilanların sayfa içerisine en az fire ile yerleőtirilmesini ve yerleőtirilen ilanlardan maksimum kazanç elde edilmesini amaçlamaktadır. Sayfa içerisine yerleőtirilen ilanlar bu problemde kategorik olarak gruplandırılmamaktadır. İlanlar belirtilen maksimum ve minimum değerlerde yükseklik ve genişliğe sahip olmaktadır. Daha önceki yapılan 2 boyutlu yerleőtirme problemlerinden farklı olarak bu çalışmada üç farklı algoritma farklı parametre değerleri çalıştırılmış ve sonuçlar karşılaştırmalı olarak deneysel sonuçlar bölümünde gösterilmiştir.

2. KULLANILAN ALGORİTMALAR

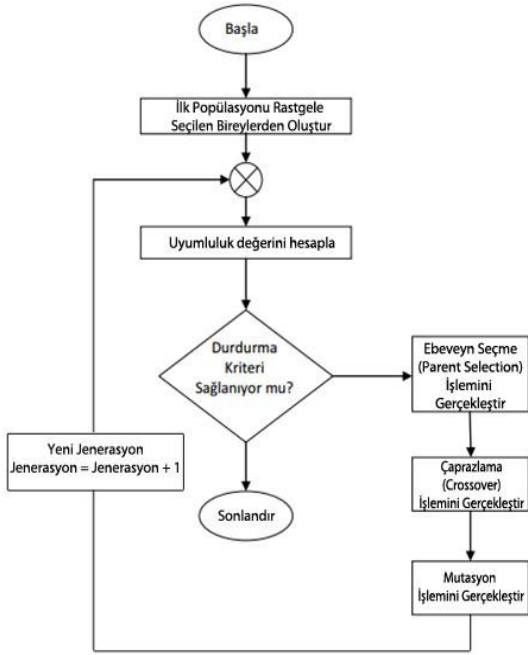
2.1. Genetik Algoritma

Genetik algoritmalar, Darwin'in doğal seçim ilkesini baz alarak geliştirilen arama, en iyileme ve optimizasyon tekniğidir. Günümüzde karşılaşılan bir çok gerçek dünya probleminde genetik algoritmalar başarıyla uygulanmaktadır. Diğer optimizasyon tekniklerine göre farklı yaklaşımlar içermektedir. Çalışma yapısı olarak deterministik yerine olasılıksal kararlar verir. Genetik algoritmalarda her zaman bir rastsallık vardır. Bu sayede çeşitlilik artırılır ve yerel optimuma takılma olasılığı, sürekli iyiye yakınsama durumu en aza indirilmeye çalışılır. [1]

Genetik algoritmalar ilk olarak Prof. John Holland tarafından ortaya atılmış, De Jong ve Goldberg tarafından genişletilmiştir. Holland'ın genetik algoritması evrimsel süreçteki aşamalara benzer durumları içerir. Bir nevi evrimsel süreç taklit edilir. Başlangıç popülasyonunun oluşturulması, yeni çocuklar üretmek üzere seçilen ebeveynler, bir çeşit doğal seleksiyon sağlamak amacıyla kullanılan çaprazlama ve mutasyon operatörleri daha iyi popülasyonlar oluşturmayı amaçlamak için kullanılan yöntemlerdir.[2]

Algoritmanın genel işleyişinde ilk olarak rastgele seçilmiş kişilerden oluşan belirli boyutta bir popülasyon oluşturulur. Daha sonra bu popülasyondaki bireyler amaç fonksiyonuna sokularak uygunluk değerleri hesaplanır. Amaç fonksiyonları problemin tanımına göre değişkenlik gösterebilmektedir. Hesaplanan uygunluk değeri algoritmanın sonlanması için gerekli değer ile eşleşirse yani problemin çözümüne ulaşırsa daha fazla yeni popülasyon(jenerasyon) oluşturmaya gerek yoktur. Fakat uygunluk değeri sağlanmazsa daha iyi popülasyonlar oluşturmak için bir sonraki adımlara geçilir. Yeni çocuklar oluşturmak amacıyla seçilme işlemi gerçekleştirilir. Mevcut popülasyon içerisinde belirli operatörlere göre seçilen bireyler belirli olasılıklara göre çaprazlanarak bir sonraki nesil için çocuklar üretilir. Üretilen çocuklar belirli olasılıklara göre mutasyona uratılır. Oluşan

çocuklar artık bir sonraki popülasyonun(jenerasyonun) bireyleridir. Bu işlemler problem çözümüne ulaşana kadar veya belirli bir adım sayısı kadar tekrarlanır. Soldaki şekilde genetik algoritmanın akış diyagramı görülmektedir.



Şekil 1. Genetik Algoritma Akış Diyagramı

çözümüne ulaşmaya çalışılır. Aynı zamanda farklı parametre kullanımı çözüm uzayının çeşitliliğini arttırmaya katkı sağlar ve yerel optimumlara takılmak yerine global optimumları yakalamayı kolaylaştırır. Aşağıda bu çalışmada kullanılan parametreler açıklanmaktadır.

Turnuva Seçimi (Tournament Selection): Turnuva seçimi genetik algoritmalarda sıklıkla kullanılan bir ebeveyn seçilim(parent selection) yöntemidir. Yöntemin çalışmasına bakıldığında en başta turnuva boyutunu belirlemek için bir k değişkeni seçilir. Bu k değişkenine göre turnuvaya sokulacak birey sayısı belirlenir. Örneğin k=3 olduğunu varsayalım. Mevcut popülasyondan üç adet birey rastgele bir şekilde seçilir. Daha sonra bu bireyler arasında turnuva gerçekleştirilir. Turnuvalar istenildiği kadar tekrarlanabilir. Bu bireylerden en yüksek uygunluk değerine sahip olan birey turnuvayı kazanır ve

çiftleşme havuzuna (mating pool) dahil edilir. Bu havuz yavru üretecek olan bireylerin ayrıldığı havuzdur. Daha sonra bu havuzdan seçilen bireylere çaprazlama ve mutasyon işlemleri uygulanarak bir sonraki nesil için yeni çocuk bireyler oluşturulur.

Düzenli Çaprazlama (Uniform Crossover): Bu yöntem genetik algoritmalarda çaprazlama işlemi için sıklıkla kullanılan bir yöntemdir. Örneğin yeni çocuklar oluşturmak üzere on gen uzunluğuna sahip iki adet ebeveyn seçilmiş olsun. Bu ebeveynlerden yine aynı gen uzunluğuna sahip iki çocuk oluşturmak istenilsin. Genetik algoritmalarda en başta sistem için bir çaprazlama olasılığı belirlenir. Aynı zamanda bu çaprazlama yönteminde ebeveynlerin gen uzunluğu ile aynı uzunlukta, 0-1 aralığında rastgele üretilmiş çaprazlama olasılıklarının tutulduğu bir dizi oluşturulur. Bireyin her bir geni için 0-1 aralığında rastgele üretilmiş olan çaprazlama olasılıkları ile sistemin çaprazlama olasılığı sırayla karşılaştırılır. Bu örnek için sistemin çaprazlama olasılığı 0.5 olsun. Eğer bireyin seçilen genine karşılık gelen çaprazlama olasılığı 0.5 ten büyük veya eşit ise çaprazlama yapılır (1.ebeveyn ve 2.ebeveyn arasında gen takası gerçekleştirilir), 0.5 ten küçük ise çaprazlama yapılmaz. Yapılan işlemler sonucunda yeni çocuk bireyler oluşturulur.

Gaussian Mutasyon: Bu yöntem genetik algoritmalarda kullanılan mutasyon yöntemidir. Başlangıçta sistem için bir mutasyon olasılığı belirlenir. Çaprazlama yönteminde olduğu gibi bireyin her bir genine karşılık gelen 0-1 aralığında rastgele üretilmiş mutasyon olasılıkları belirlenir. Bireyin genleri için belirlenen mutasyon olasılıkları sırayla sistemin mutasyon olasılığı ile karşılaştırılır. Eğer bireyin seçilen geni için belirlenmiş olan mutasyon oranı, sistemin mutasyon oranından küçük veya eşit ise mutasyon işlemi gerçekleşir. Aksi halde mutasyon işlemi gerçekleşmez. Mutasyon işlemi gerçekleşirse belirli aralıklarda üretilen rastgele değer mutasyon işlemi gerçekleşecek gene eklenir. Bu işlemler bireyin her bir geni için sırayla tekrarlanır ve sonunda mutasyona uğramış yeni bireyler üretilir.

Gezgin satıcı problemi, sırt çantası problemi, çizelgeleme problemleri, iş atama problemleri, 2 ve 3 boyutlu yerleşim problemi gibi problemlerin esas alındığı gerçek dünya problemlerinde genetik algoritmalar başarılı bir şekilde uygulanmaktadır. Yapılan çalışmanın genetik algoritmalar kısmı için yazılan uygulama kodu ekler bölümünde bulunmaktadır. (Ek- A)

2.2. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) 1995 yılında Dr. Eberhart ve Dr. Kennedy tarafından ortaya atılmış bir optimizasyon yöntemidir. Genetik algoritmalarda olduğu gibi parçacık sürü optimizasyonu da popülasyonlardan oluşur. Popülasyonların oluşumunda ve kararların verilmesinde belirli bir rastsallık vardır.

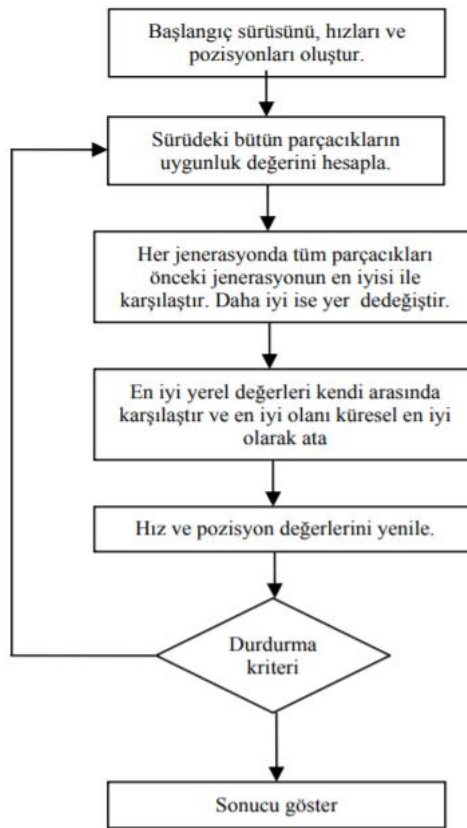
Parçacık sürü optimizasyonu kuş veya balık sürülerinin sosyal davranışlarına dayanarak geliştirilmiştir. Parçacık sürü optimizasyonu algoritması en temelde bireylerin en uzun süre hayatta kalması için yaptığı davranışları taklit eder. Her bir birey birer parçacık olarak düşünülürse, her parçacık kendisinin daha önce verdiği karar ve grubun kararları doğrultusunda stokastik hareket eder. Parçacığın aldığı kararlarda grubun en iyi kararı ve kendisinin geçmiş deneyimleri etkilidir. Kararının sonucunda oluşan durumlara göre performansını değerlendirir ve bir önceki adımıyla kıyaslayarak kendisi için en iyi kararı seçmeye çalışır. Parçacık karar verirken her zaman grubun kararına uymaz yada her zaman en iyi performans veren bir önceki kararını uygulamaz. Her zaman bir rastsallık vardır. Oluşan bu rastsallıklara göre atılacak adım büyüklükleri hesaplanır ve maksimum fayda sağlanmaya çalışılır. [3]

PSO, Genetik Algoritmalar ve evrimsel algoritmalar ile birçok benzerlik gösterir. Rastgele seçilen bireyler ile belirlenen uzunlukta bir popülasyon oluşturulur. Oluşturulan popülasyon ile başlanılır ve en iyi çözümü bulmak için popülasyon güncellenerek problemin çözümü aranır. Genetik algoritmalarından farklı olarak PSO'da mutasyon ve çaprazlama operatörleri mevcut değildir. PSO'da her bir parçacık veya birey potansiyel çözüm olarak kabul edilebilir. Bu parçacıklar, mevcut en iyi çözümleri takip ederek problem uzayında gezinir ve daha iyi çözümleri ararlar.

Hız ve pozisyon değerleri aşağıdaki denklemlerde belirtilen şekilde yenilenir:

$$v_i^{k+1} = v_i^{kt} + c_1 \text{rand}_1(\text{pbest}_i - s_i^k) + c_2 \text{rand}_2(\text{gbest} - s_i^k) \quad [6]$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad [6]$$



Şekil 2. Parçacık Sürü Optimizasyonu akış diyagramı

PSO algoritma gerçekleştirimi bakımından genetik algoritmalara göre daha kolaydır. Genetik algoritmalara oranla daha az parametreye sahiptir. Bu durum parametre kontrolünü kolaylaştırmaktadır ve yüksek hesaplama verimi sağlamaktadır. PSO elektro manyetik cihaz tasarımı, fonksiyon optimizasyonu, bulanık sistem kontrolü gibi pek çok alanda başarılı bir şekilde uygulanmıştır.[3] Yapılan çalışmanın PSO kısmı için yazılan uygulama kodu ekler bölümünde bulunmaktadır. (Ek- B)

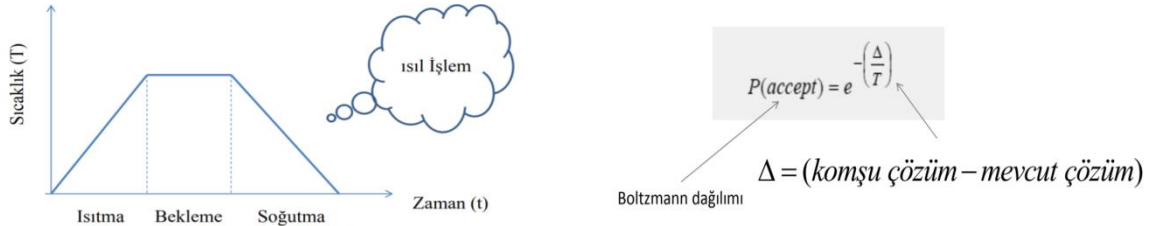
2.3. Benzetilmiş Tavlama (Simulated Annealing)

Benzetilmiş tavlamanın kökleri termodinamiğe dayanır. Benzetilmiş tavlama yönteminde sistemin termal enerjisi incelenir. Erimiş metalin soğutulması işlemi bu algoritmanın esin noktası olarak kabul edilebilir. Metalin soğutulması sırasında enerji dalgalanmaları yaşanır. Yaşanan bu dalgalanmalar çeşitliliği artırarak yerel enerji minimumuna takılmayı güçleştirir. Ancak sistemin çok hızlı bir şekilde soğutulması veya ısıtılması yerel enerji minimumuna takılmaya neden olabilir. Bu durumun sonucunda da sistem çok fazla enerji içerebilir. Benzetimli tavlama işlemi işte bu noktada enerji değeri için yerel minimum yerine küresel minimumun aranmasını amaçlar.[4]

Benzetilmiş tavlama algoritmasında da PSO ve Genetik algoritmalarda olduğu gibi stokastik karar verme yani rastsallık vardır. Aday çözümler, mevcut çözümlerden daha kötü bile olsa belirli bir olasılıkla kabul edilebilir.[5] Yani yeni çözümün kabul edilip edilmemesinde bir rastsallık vardır. Bu durum çeşitliliği artırma yolunda büyük bir öneme sahiptir. Kötü çözümlerin belirli bir olasılıkla kabul edilmesi sürekli iyiyeye yakınsamayı engeller. Bu şekilde sistemin yerel optimuma takılma olasılığı azaltılırken, çeşitlilik artırılarak küresel optimumun yakalanma şansı artırılır.

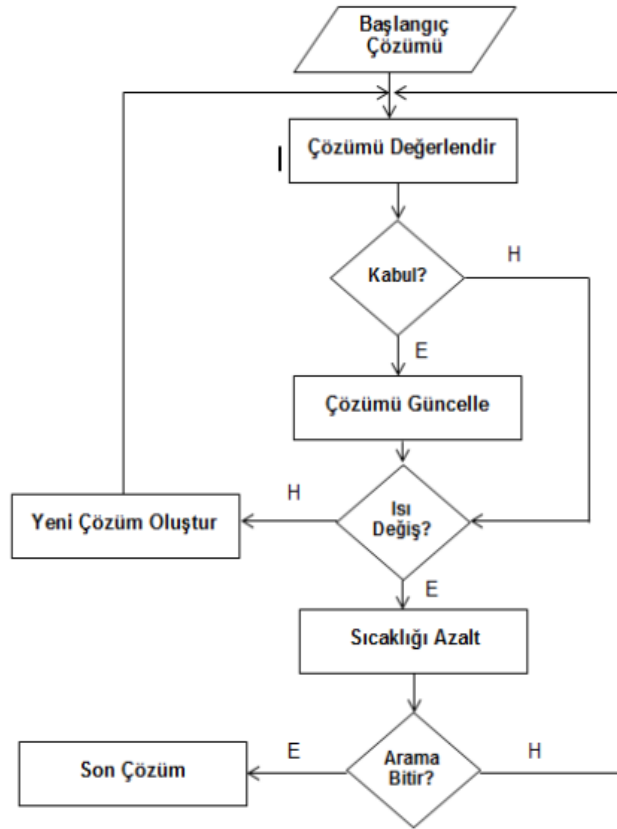
Benzetilmiş tavlama algoritması, diğer olasılıksal yaklaşımlar (genetik algoritmalar, PSO vb.) gibi en optimum çözümün en hızlı şekilde bulunmasını sağlar. Genellikle, kombinasyonel problemlerin optimize edilmesinde kullanılır. [5] Benzetilmiş tavlama algoritması; elektronik devre tasarımı, görüntü işleme, en kısa yol problemleri, gezgin seyyah satıcı tarzı seyahat problemleri, endüstriyel malzeme uygulamalarının simülasyonu, kesme ve paketleme problemleri, çizelgeleme ve graf renklendirme problemlerinin çözümlerinde başarılı sonuçlar vermiştir.[5] Yapılan çalışmanın benzetilmiş tavlama algoritması kısmı için yazılan uygulama kodu ekler bölümünde bulunmaktadır. (Ek- C)

Tavlama: Malzemeyi belirli bir süre (tavlama sıcaklığına kadar) ısıttıktan sonra, yavaş yavaş soğutmaktır. Tavlama malzemeyi rahatlatmak, yumuşatmak ve iç yapıyı daha kullanılabilir hale getirmek için yapılan ısı işlemlerinin geneline verilen addır.[8]



Şekil 3. Tavlama gösterimi (Solda) ve Boltzmann Dağılımı (Sağda) [8]

Benzetimli tavlama adımları aşağıdaki şekilde gösterilmiştir.



Şekil 4. Benzetimli tavlama akış diyagramı [7]

3. DENEYSEL SONUÇLAR

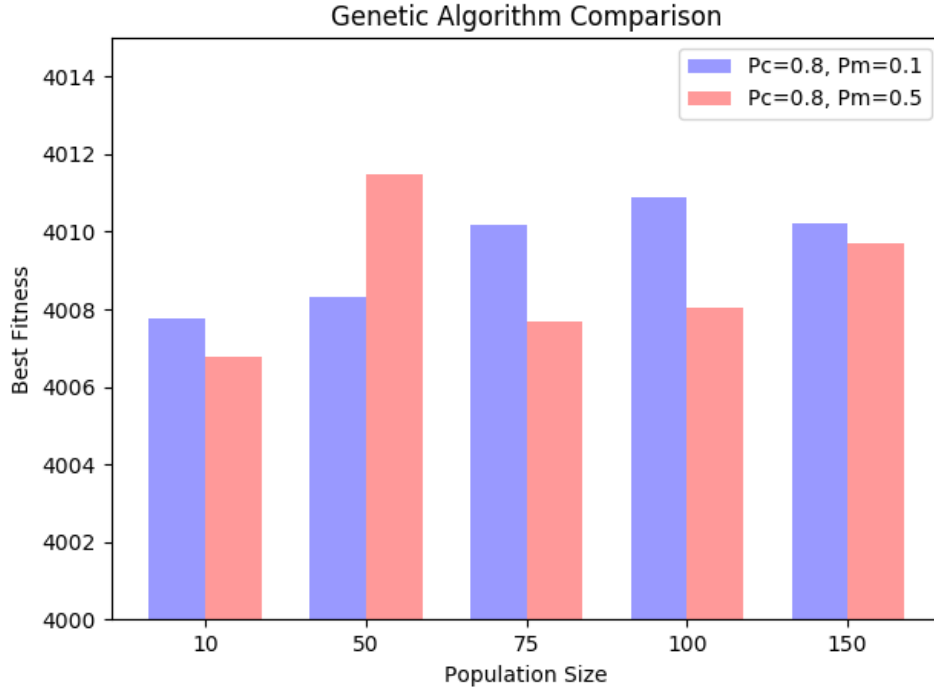
Bütün algoritmalar için gazete sayfasının boyutu 100cm * 80cm dir. Gazetenin kabul ettiği ilan boyları en az 10 cm, en fazla 50 cm olarak kabul edilmiştir. Tablolarda verilen sonuçlarda her bir satır 100 çalıştırma sonucu elde edilen en iyi değerlerin ortalamasını, standart sapmasını ve yerleştirilen en fazla ilan sayısını temsil etmektedir.

Tablo 1. Genetik Algoritma Sonuçları (Pc = 0.8, Pm = 0.1) :

Cross Over Rate: 0.8, Mutation Rate: 0.1, Tournament Selection, Gaussian Mutation				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4007.7776	3961.8720	46.6493	8
N = 50	4008.3290	4000.2601	6.8873	9
N = 75	4010.1881	4001.9863	5.8003	12
N = 100	4010.8974	4003.5111	3.4922	12
N = 150	4010.2308	4004.5471	2.3844	12
Ortalama	4009,48458	3994,435	13,0427	10,6

Tablo 2. Genetik Algoritma Sonuçları (Pc = 0.8, Pm = 0.5):

Cross Over Rate: 0.8, Mutation Rate: 0.5, Tournament Selection, Gaussian Mutation				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4006.7632	3952.0699	47.5323	11
N = 50	4011.4684	3999.6485	6.7277	12
N = 75	4007.6778	4001.9865	4.8063	11
N = 100	4008.0218	4003.2986	3.0644	12
N = 150	4009.6936	4004.8302	2.2144	10
Ortalama	4008,72496	3992,367	12,86902	11,2



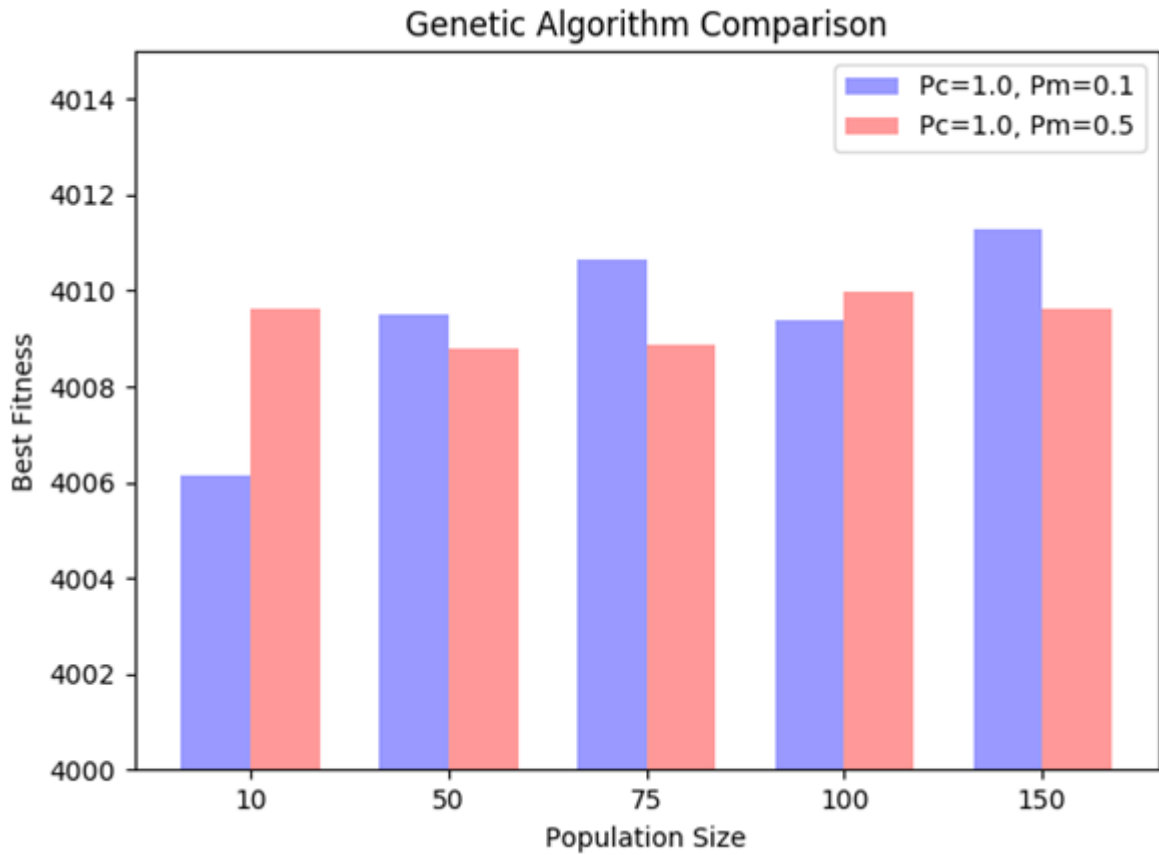
Şekil 5. Tablo 1 ve Tablo 2 Karşılaştırması

Tablo 3. Genetik Algoritma Sonuçları (Pc = 1.0, Pm = 0.1) :

Cross Over Rate: 1.0, Mutation Rate: 0.1, Tournament Selection, Gaussian Mutation				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4006.1309	3938.9961	59.6192	10
N = 50	4009.4940	3996.6610	11.3194	11
N = 75	4010.6399	3998.5952	9.0578	11
N = 100	4009.3889	4000.8092	6.5493	11
N = 150	4011.2752	4003.1724	4.9158	12
<i>Ortalama</i>	<i>4009,386</i>	<i>3987,647</i>	<i>18,2923</i>	<i>11</i>

Tablo 4. Genetik Algoritma Sonuçları (Pc = 1.0, Pm = 0.5) :

Cross Over Rate: 1.0, Mutation Rate: 0.5, Tournament Selection, Gaussian Mutation				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4009.6078	3948.6209	53.7743	11
N = 50	4008.8049	3996.4461	9.1968	11
N = 75	4008.8615	3998.7210	8.7781	11
N = 100	4009.9577	4000.6389	6.4175	11
N = 150	4009.6164	4003.1720	3.7619	12
<i>Ortalama</i>	<i>4009,36966</i>	<i>3989,52</i>	<i>16,38572</i>	<i>11,2</i>



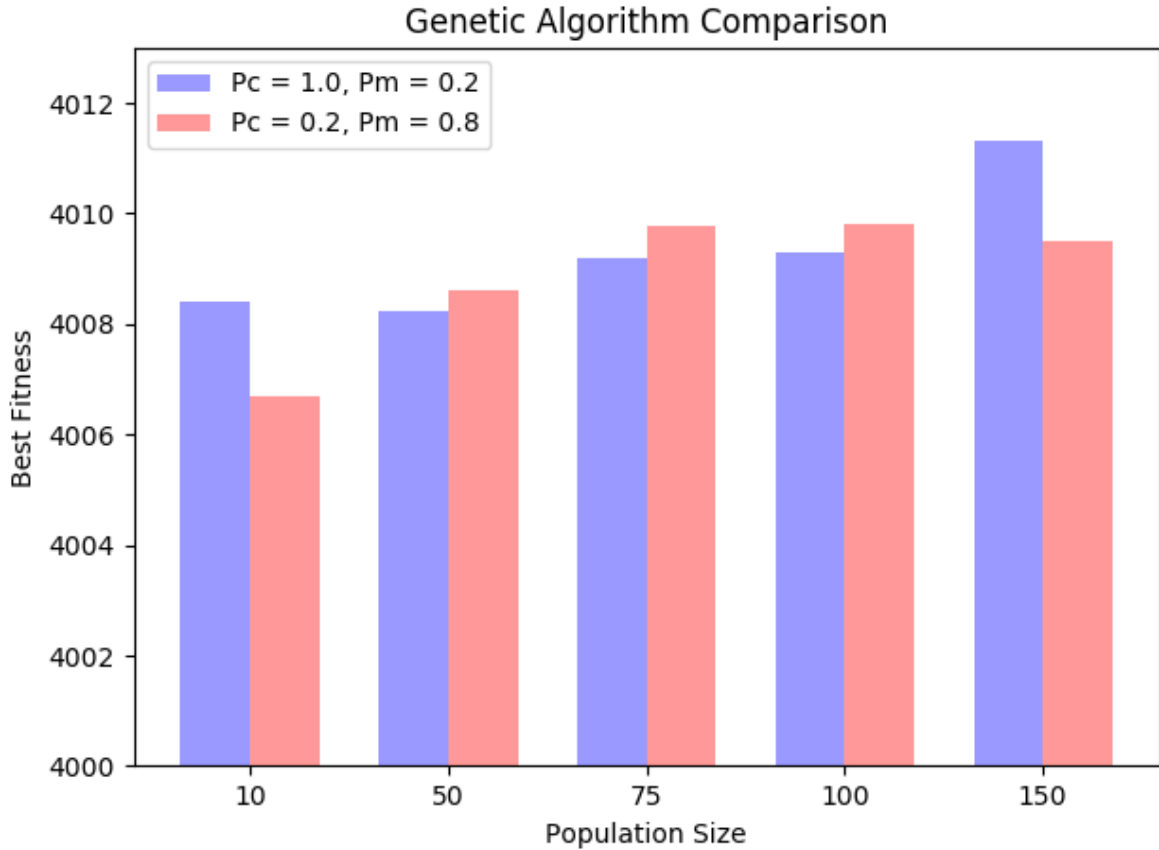
Şekil 6. Tablo 3 ve Tablo 4 Karşılaştırması

Tablo 5. Genetik Algoritma Sonuçları (Pc = 1.0, Pm = 0.2) :

	Cross Over Rate: 1.0, Mutation Rate: 0.2, Tournament Selection, Gaussian Mutation			
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4008.41876	3948.0596	48.4226	11
N = 50	4008.2496	3996.6306	9.8022	10
N = 75	4009.2047	3997.5050	9.4298	11
N = 100	4009.2900	4000.7557	6.3792	11
N = 150	4011.3313	4003.5691	3.9152	12
<i>Ortalama</i>	<i>4009,299</i>	<i>3989,304</i>	<i>15,5898</i>	<i>11</i>

Tablo 6. Genetik Algoritma Sonuçları (Pc = 0.2, Pm = 0.8) :

	Cross Over Rate: 0.2, Mutation Rate: 0.8, Tournament Selection, Gaussian Mutation			
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4006.6994	3961.3328	40.6447	12
N = 50	4008.6040	3998.9637	7.7693	13
N = 75	4009.7816	4001.5978	5.2989	10
N = 100	4009.8018	4003.1330	3.3970	12
N = 150	4009.5184	4004.4295	2.7607	12
<i>Ortalama</i>	<i>4008,88104</i>	<i>3993,891</i>	<i>11,97412</i>	<i>11,8</i>

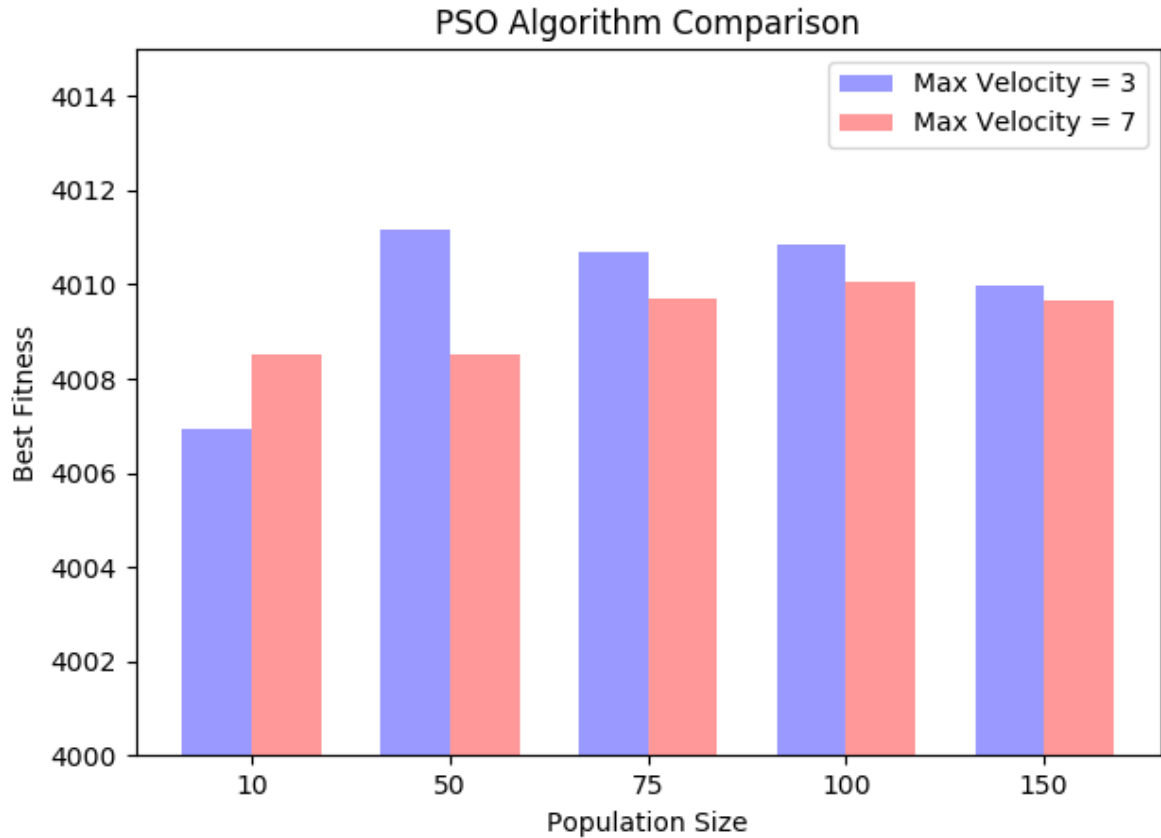
**Şekil 7. Tablo 5 ve Tablo 6 Karşılaştırması**

Tablo 7. PSO Sonuçları (Max Velocity = 3, Generation Size = 1000) :

Max Velocity = 3, Generation Size = 1000				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4006.9301	3976.4424	38.7776	11
N = 50	4011.1528	4001.4075	7.1043	13
N = 75	4010.6775	4002.7621	4.5769	11
N = 100	4010.8492	4004.4484	4.0565	12
N = 150	4009.9640	4005.8758	2.4063	12
<i>Ortalama</i>	<i>4009,91472</i>	<i>3998,187</i>	<i>11,38432</i>	<i>11,8</i>

Tablo 8. PSO Sonuçları (Max Velocity = 7, Generation Size = 1000) :

Max Velocity = 7, Generation Size = 1000				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4008.5336	3959.2199	55.3578	11
N = 50	4008.5020	3999.7489	7.3209	11
N = 75	4009.6960	4001.0951	6.6789	12
N = 100	4010.0507	4004.0942	3.7218	12
N = 150	4009.6632	4004.8951	2.7428	11
<i>Ortalama</i>	<i>4009,2891</i>	<i>3993,811</i>	<i>15,16444</i>	<i>11,4</i>



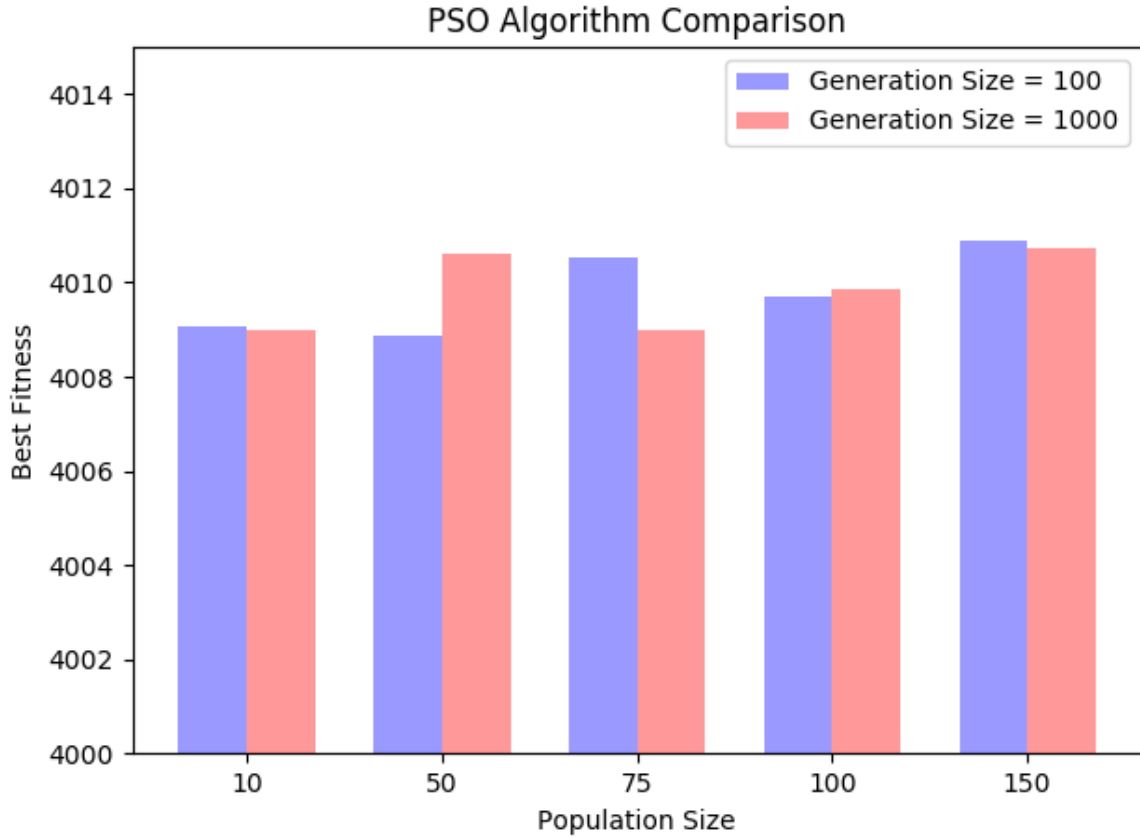
Şekil 8. Tablo 7 ve Tablo 8 Karşılaştırması

Tablo 9. PSO Sonuçları (Max Velocity = 2, Generation Size = 100) :

Max Velocity = 2, Generation Size = 100				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4009.0554	3981.0672	27.4005	12
N = 50	4008.8729	4003.6989	4.7667	11
N = 75	4010.5385	4004.0565	4.4593	11
N = 100	4009.6889	4005.2483	3.2059	13
N = 150	4010.8802	4006.1299	2.4922	12
<i>Ortalama</i>	<i>4009,80718</i>	<i>4000,04</i>	<i>8,46492</i>	<i>11,8</i>

Tablo 10. PSO Sonuçları (Max Velocity = 2, Generation Size = 1000) :

Max Velocity = 2, Generation Size = 1000				
	Best Value	Mean	Standart Dev.	Max Advert Size
N = 10	4008.9919	3973.4600	39.2276	11
N = 50	4010.6083	4003.1013	5.2689	11
N = 75	4008.9993	4004.2011	3.8169	12
N = 100	4009.8571	4005.7501	3.3309	10
N = 150	4010.7374	4006.4741	2.9697	12
<i>Ortalama</i>	<i>4009,8388</i>	<i>3998,597</i>	<i>10,9228</i>	<i>11,2</i>



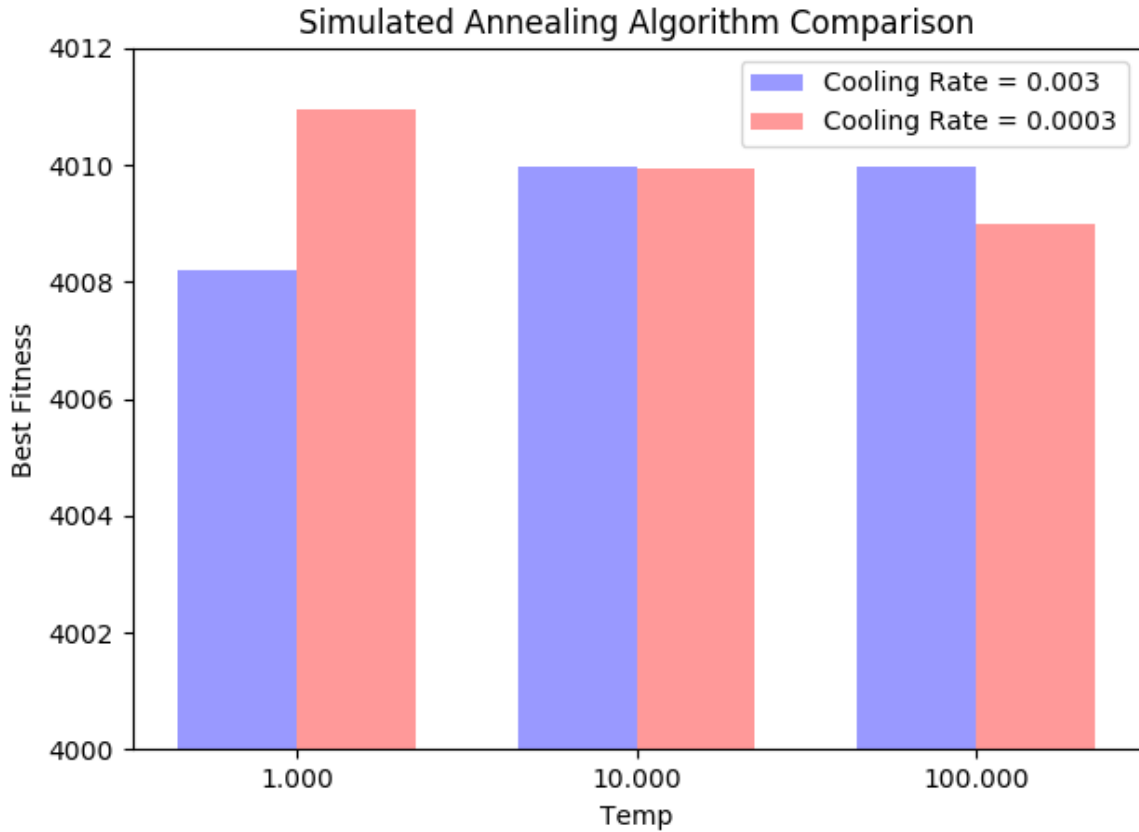
Şekil 9. Tablo 9 ve Tablo 10 Karşılaştırması

Tablo 11. Simulating Annealing Sonuçları (Cooling Rate: 0.003) :

Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 1,000, Cooling Rate = 0.003	4008.2062	4004.0541	5.2875	10
Temp 10,000 , Cooling Rate = 0.003	4009.9770	4004.6716	2.9513	11
Temp 100,000, Cooling Rate = 0.003	4009.98371	4004.9990	3.5459	11
<i>Ortalama</i>	<i>4009,389</i>	<i>4004,575</i>	<i>3,928233</i>	<i>10,66667</i>

Tablo 12. Simulating Annealing Sonuçları (Cooling Rate: 0.0003):

Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 1,000, Cooling Rate = 0.0003	4010.9579	4001.4547	46.8465	11
Temp 10,000 , Cooling Rate = 0.0003	4009.9454	4006.2101	1.7652	10
Temp 100,000, Cooling Rate = 0.0003	4008.9961	4005.9594	1.4988	9
<i>Ortalama</i>	<i>4009,966</i>	<i>4004,541</i>	<i>16,7035</i>	<i>10</i>



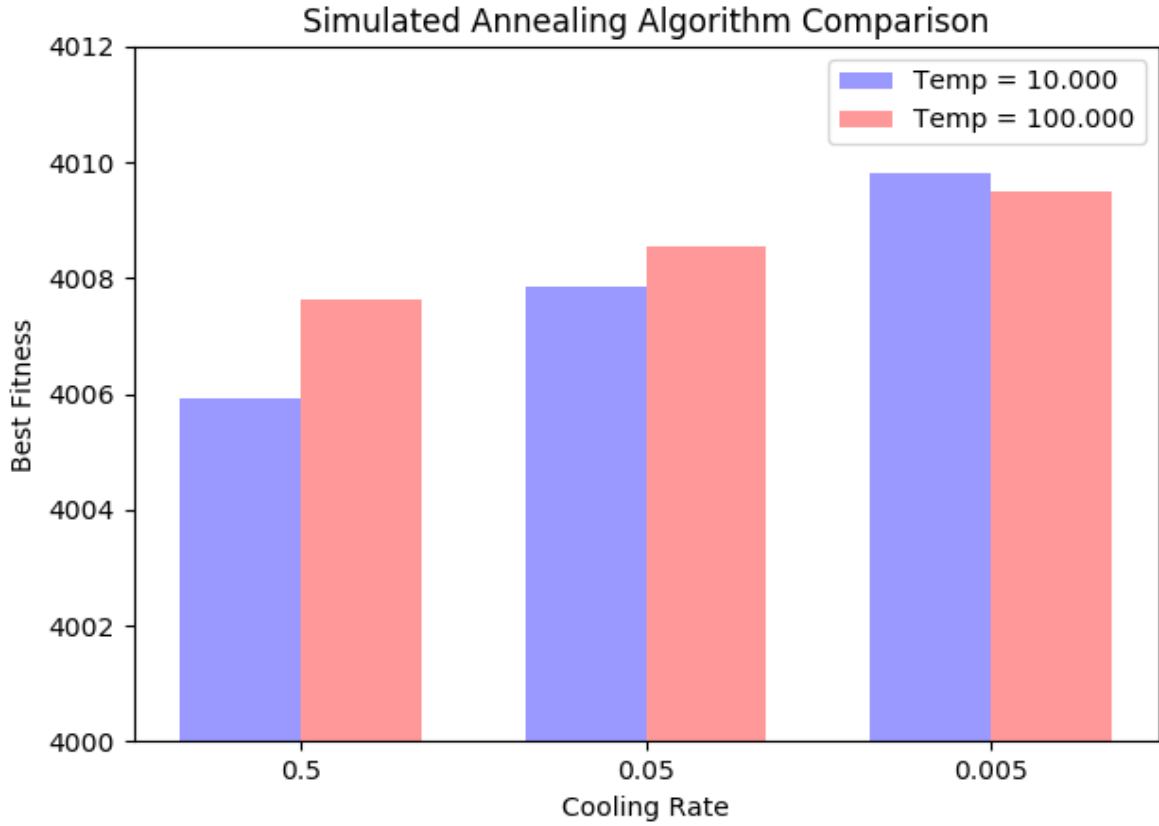
Şekil 10. Tablo 11 ve Tablo 12 Karşılaştırması

Tablo 13. Simulating Annealing Sonuçları (Temp: 10.000) :

Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 10,000, Cooling Rate = 0.5	4005.9339	3828.7259	175.5746	10
Temp 10,000 , Cooling Rate = 0.05	4007.8449	3977.7098	45.8330	11
Temp 10,000, Cooling Rate = 0.005	4009.8177	4003.3258	5.3756	11
<i>Ortalama</i>	<i>4007,866</i>	<i>3936,587</i>	<i>75,5944</i>	<i>10,66667</i>

Tablo 14. Simulating Annealing Sonuçları (Temp: 100.000):

Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 100,000, Cooling Rate = 0.5	4007.6321	3837.7352	182.6294	11
Temp 100,000 , Cooling Rate = 0.05	4008.5648	3984.5078	37.4099	9
Temp 100,000, Cooling Rate = 0.005	4009.4926	4004.0231	3.5735	9
<i>Ortalama</i>	<i>4008,563</i>	<i>3942,089</i>	<i>74,5376</i>	<i>9,666667</i>



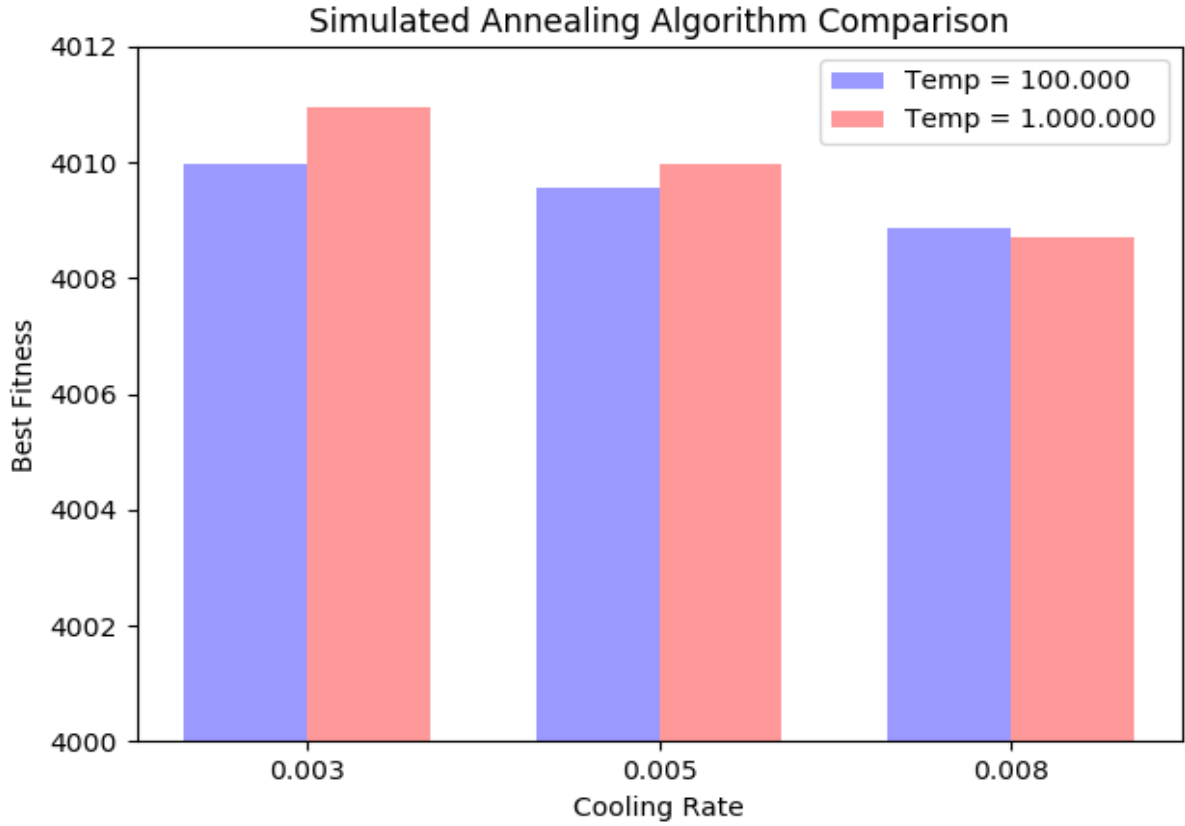
Şekil 11. Tablo 13 ve Tablo 14 Karşılaştırması

Tablo 15. Simulating Annealing Sonuçları (Temp: 100.000) :

Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 100,000, Cooling Rate = 0.003	4009.9853	4004.3069	4.3711	11
Temp 100,000 , Cooling Rate = 0.005	4009.5662	4004.4713	2.8028	10
Temp 100,000, Cooling Rate = 0.008	4008.8565	3998.5777	47.8083	10
<i>Ortalama</i>	<i>4009,469</i>	<i>4002,452</i>	<i>18,3274</i>	<i>10,33333</i>

Tablo 16. Simulating Annealing Sonuçları (Temp: 1.000.000):

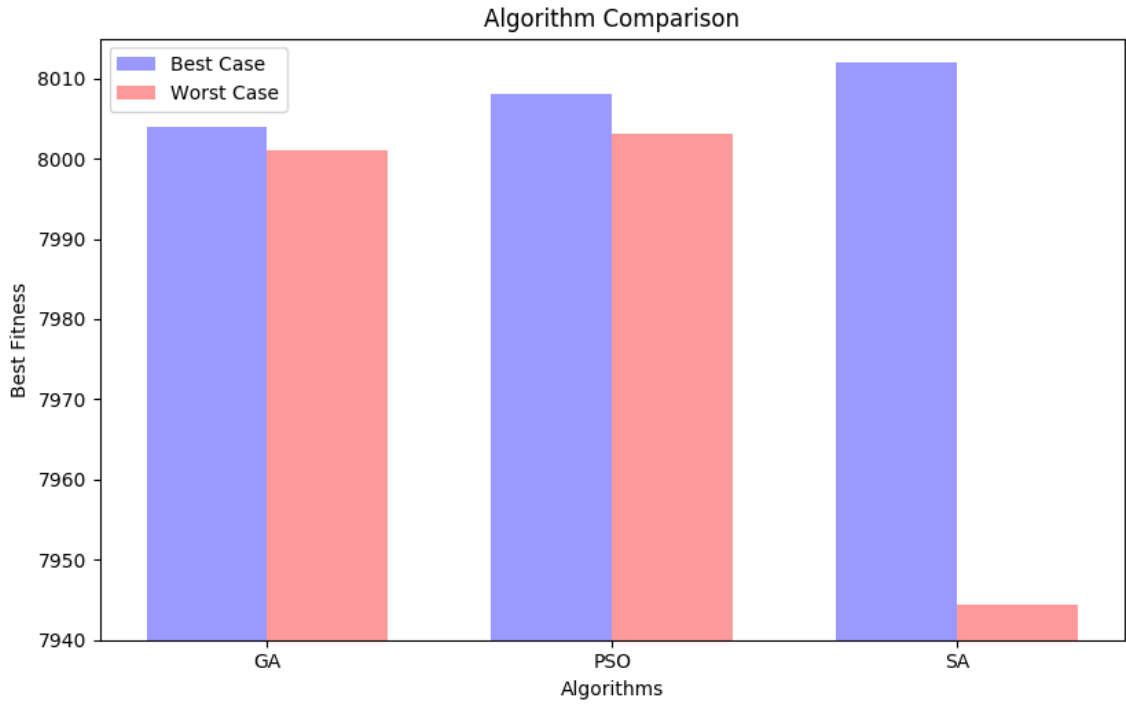
Simulated Annealing				
	Best Value	Mean	Standart Dev.	Max Advert Size
Temp = 1,000,000, Cooling Rate = 0.003	4010.9476	4005.1952	2.1842	11
Temp 1,000,000 , Cooling Rate = 0.005	4009.9714	4003.5057	5.4696	9
Temp 1,000,000, Cooling Rate = 0.008	4008.7214	4004.1096	3.5226	9
<i>Ortalama</i>	<i>4009,88</i>	<i>4004,27</i>	<i>3,725467</i>	<i>9,666667</i>



Şekil 12. Tablo 15 ve Tablo 16 Karşılaştırması

Tablo 17. Üç Farklı Algoritmanın En İyi ve En Kötü Durumlarının Karşılaştırılması:

	Genetik Algoritma	PSO	Simulated Annealing
Best Case: Best Value Mean + Run Mean	8003,915	8008,1012	8011,948
Worst Case: Best Value Mean + Run Mean	8001,091	8003,100	7944,453
Ortalama	8002,503	8005,601	7978,201



Şekil 13. Algoritmaların En İyi ve En Kötü Durumlarının Karşılaştırılması

4. SONUÇ

Bu çalışma ile gazetelerde bulunan ilan sayfalarının yerleşim probleminin çözülmesi amaçlanmıştır. Günlük yayınlanan gazetelerde talep doğrultusunda ilan sayfaları oluşturulmaktadır. Bu ilanların yerleştirilmesi sırasında minimum yer israfı ve maksimum kazanç sağlamak temel amaçtır. Bu problemin çözümünde gazetenin yükseklik ve genişlik gibi temel değerleri ve aynı zamanda yerleştirilecek olan ilanların yükseklik ve genişlik değerleri problem çözümü için gereken kritik parametrelerdendir.

Problem çözümü için 3 adet farklı yöntem denenmiştir. Ulaşılan sonuçlar arasında genel olarak büyük bir fark görünmemekle beraber, birbirine yakın sonuçlar elde edilmiştir. Farklı yöntemler için denen farklı parametre değerleri sonuçları doğrudan etkilemiştir. Bu nedenden dolayı algoritmalar için farklı parametre değerleri karşılaştırmalı olarak gösterilmiştir. Bu sonuçlardan yola çıkarak her algoritma için optimum değerler yakalanabilir.

Genetik Algoritma, Parçacık Sürü Optimizasyonu ve Benzetilmiş Tavlama algoritmaları en kötü ve en iyi durumlardaki değerleri üzerinden karşılaştırıldı. Bu sonuçlar yorumlanmak istenirse en iyi ve en kötü durumların ortalamasına bakıldığında Parçacık Sürü Optimizasyon algoritması en iyi sonucu veren algoritma olmuştur. Daha sonra Genetik Algoritma ve arkasından Benzetilmiş Tavlama Algoritması gelmiştir. Benzetilmiş Tavlama algoritması en iyi senaryoda en yüksek değeri vermesine rağmen ortalama olarak kötü sonuç vermiştir.

Bu çalışma ile farklı algoritmaların seçilen gerçek dünya problemi üzerine uygulanması örneklenmiştir. Bir algoritma her durum için her zaman en iyi sonuçları vermez. Bu problem için en iyi sonucu veren algoritma Parçacık Sürü Optimizasyonu Algoritması olmuştur. Aynı zamanda algoritmalar için seçilen parametre değerleri sonuç ile doğrudan ilişkilidir. Probleme uygun algoritmanın seçilmesi ve bu algoritma için optimum parametrelerin belirlenmesi problemin çözümü için büyük bir öneme sahiptir. Bu çalışma ile bu sorunlar ortaya konulmuş ve çözüm geliştirilmeye çalışılmıştır.

Referanslar:

- [1] Emel, Gül Gökay, and Ç. Taşkın. "Genetik Algoritmalar ve Uygulama Alanları." *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* 21.1 (2002): 130.
- [2] Saruhan, Hamit. "Genetic algorithms: an optimization technique." *Technology* 7.1 (2004): 105.
- [3] Lazinica, Aleksandar, ed. *Particle swarm optimization*. Kirchengasse: InTech, 2009: 2-3.
- [4] Goffe, William L., Gary D. Ferrier, and John Rogers. "Global optimization of statistical functions with simulated annealing." *Journal of econometrics* 60.1-2 (1994): 68-69.
- [5] Aarts, Emile, Jan Korst, and Wil Michiels. "Simulated annealing." *Search methodologies* (2005): 91 - 120.
- [6] Parçacık Sürü Optimizasyonu, Fatih Sultan Mehmet Vakıf Üniversitesi, Yapay Zeka Uygulamaları Ders Slaytları, Yrd. Doç. Dr. Ebubekir KOÇ
- [7] <http://hasansubasi.blogspot.com.tr/2015/06/tavlama-benzetimi-simulated-annealing.html>
- [8] Tavlama Benzetimi Algoritması (Simulated Annealing), Fırat Üniversitesi, BMÜ-579 Benzetim ve Modelleme Ders Slaytı, Yrd. Doç. Dr. İlhan Aydın

Ekler:

Ek – A : https://github.com/zekikus/GA_2DLayout

Ek – B: https://github.com/zekikus/PSO_2DLayout

Ek – C: https://github.com/zekikus/SimulatedAnnealing_2DLayout

Gezgin Satıcı Probleminin Random Koordinatlar Üzerinde Uygulanması

Ulaş Demirci

Mühendislik ve Fen Bilimleri Enstitüsü, Fatih Sultan Mehmet Vakıf Üniversitesi, Beyoğlu, İstanbul,

ulas.demirci@stu.fsm.edu.tr

Bu çalışmada servis elamanlarının en kısa mesafelere giderek katedilecek yolun minimize edilmesi ele alınmaktadır. Gelen taleplerle oluşmuş servis listesindeki yerlere başlangıç noktasından başlayarak servis noktalarına tekrar uğramadan başlangıç noktasına en kısa güzergahı kullanarak ulaşma problemidir. Burada en çok kullanılan sürü algoritması olan Karınca Algoritması (Ant Algorithm) ve doğadaki evrimi kullanan Genetik Algoritması (Genetic Algorithm) ile problemimizi çözmeye çalışacağız. Buradaki amaç iki algoritmanın benzer durumlardaki problemleri çözerken ortaya koyduğu tepkiyi görmektir.

Giriş

İnsanlık Tarihinde ulaşım problemi her zaman sorun olmuştur. Bu problem temel ihtiyaçların kısıtlı olarak barındıkları yerlerden (Kaynaktan) yiyecek bulmak yada avlanmak için(hedef) gidecekleri noktalara taşıma probleminden dolayı en kısa güzergahları kullanmaları gerekmektedir. Buda en kısayol problemini oluşturmuş. Kısayollar bizim dilimizde patika olarak adlandırılmıştır. Patikalar bir güzergaha giderken en kısa yolun kullanılması ile oluşmuştur. Patikalar; ilk kullananın tecrübesine göre rasgele yada iç güdülerine güvenerek yol alması ve daha sonra sık sık kullanarak yada başkasına aktarması yoluyla oluşmuştur.

Patikalar çok uzun zamanlarda oluşmaktadır. Günümüzde hızla gelişen teknoloji ile birlikte en kısa yol problemi gerçek bir problem olan, Gezgin Satıcı Problemi (GSP) ye dönüştürülerek çözülmeye çalışılmıştır.

İnsanoğlu karşısına çıkan problemleri çözmek için etrafındaki yaşanan olayları inceleyerek, problemlerini çözmeye çalışmıştır. Bizde burada Karınca Kolonisi ve Genetik Algoritma ile problemimizi çözmeye çalışacağız ve bu iki algoritmayı bir biri ile karşılaştıracğız.

2. KULLANILAN ALGORİTMALAR

2.1 KARINCA SİSTEM ALGORİTMASI

Yazımızda daha önce değindiğimiz gibi insanlar etrafındaki problemleri çözmek için etrafında gerçekte var olan şeyleri izleyerek kendi problemine çözüm aramıştır. Örneğin; uçma problemi için kuşların taklit edilmesi en bilinen örneklerdendir. Bizim ele aldığımız Gezgin Satıcı Problemini bir taşıma problemi olarak görüyoruz. Etrafımızdaki canlılara baktığımızda, canlıların tek tek yada sürü ile birlikte yiyecek arama ve yuvalarına taşımalarını birlikte nasıl yaptıkları görülmüştür. Karınca kolonilerinde bireylerin tek tek değil hareketlerini birbiri ile uyum içinde yaptıkları görülmüştür. Sürüdeki özerk elemanların birleşerek geliştirdikleri zeka “Sürü Zekasıdır” (Swarm Intelligence). Sürü zekası Stigmergy” ve “Self-Organization” (Kendinden organizasyon) denen iki mekanizma üzerine kuruludur.[1] Stigmergy olaylar yada algılanan veriler arasındaki iletişime yada ortama müdahale ederek üyelerin kendi kendine sonuç üretebilmeleridir.

Sürülerin bu hareketleri klasik yapay zeka kavramına yeni bir yaklaşım getirmiştir. Klasik yapay zeka kavramında bulunan insan zekası modelleme odaklı, karmaşık, merkezî, planlı yaklaşımların aksine, sürü zekası basit yapıllı, özerk, önceden planlama yapmayan dağınık ajanların kompleks problemlerin çözümünde başarılı olduklarını göstermiştir.[1]

Sürü algoritmaları arasında en başarılı olanları Karınca Kolonisi Algoritmasıdır.

Karınca Kolonisi Optimizasyonu (ACO) bir çok kombinasyonel optimizasyon problemlerinde iyi sonuçlar veren bir meta-heuristic tekniktir .[2][3]

Karınca algoritması Dorigo tarafından 1991 yılında doktora tezi olarak uygulanmıştır. Karıncalar, yiyecek kaynaklarından yuvalarına en kısa yolu görme duyularını kullanmadan bulma yeteneğine sahiptirler. Aynı zamanda, çevredeki değişime adapte olma yetenekleri vardır. Dış etkenler sonucu takip ettikleri mevcut yol artık en kısa yol değilse, yeni en kısa yolu bulabilmektedirler [4]

Karıncaların feromon bırakma ve takip etme mantığı üzerine kurulu olan bir algoritmadır. [1]

Algoritmanın Çalışma Prensibi

- 1 .Adım: Feromon değerleri belirle
- 2.Adım: Karıncalar her şehire random olarak yerleştirilir
3. Adım : Karıncalar sonraki şehire lokal arama

$$p(ij) \begin{cases} \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{t \in j_k(i)} [\tau(i,t)]^\alpha [\eta(i,t)]^\beta} & \text{eğer } j \in j_k(i) \\ 0 & \text{diğer durumlarda} \end{cases}$$

Local Feromon Güncellemesi

$\eta(ij)$: iki şehir arasındaki uzaklığın ters orantısı ($1/d(i,j)$) arama işleminin sezgisel yapılması içindir

$\tau(i,j)$ iki şehir arasındaki feromon değeri α ve β değerleri katılım değeridir feromene göre mi uzaklığa göre mi yapıcağını belirler. 0 dan farklı bir değer olarak algoritmaya fonksiyonun yeterli ağırlık katmasını sağlar.

4.Adım: her karıncının tur uzunluğu hesaplanır, en iyi tur bulunur, lokal ve global feromon yenilenmesi yapılır.

$$\Delta\tau_{i,j}^k \begin{cases} \frac{1}{L_k} & \text{eğer } k \text{ arıncası } (i,j) \text{ yolu kullanmışsa} \\ 0 & \text{diğer durumlarda} \end{cases}$$

$$\Delta\tau_{ij}: \Delta\tau_{ij} + \Delta\tau_{ij}^k$$

Global Feromon Güncellemesi

5.Adım: Eğer maksimum iterasyon sayısına erişmiş ise en kısa turu bastır eğer erişmemişse Adım 2 ye giderek işlemleri tekrar et.

2.2 GENETİK ALGORİTMASI

Problemimizi bu sefer doğadaki evrim sürecini kullanarak optimizasyon problemimize uygulayacağımız Genetik Algoritmayı (GA) inceleyeceğiz.

Michigan Üniversitesin'de yer alan John Holland GA'yı Darwinin evrimin doğal seçilimi teorisinden etkilenerek 1970 li yıllarda ortaya çıkardığı 1985'te Holland'ın öğrencisi olarak doktorasını veren David E. Goldberg adlı inşaat mühendisi 1989'da konusunda bir klasik sayılan kitabını yayınladığı dek genetik algoritmaların pek yarar olmayan bir araştırma konusu olduğu düşünülüyordu.

Burada canlıların doğal seçim (1), gen transferi ve (2)mutasyonu gibi biyolojik değerlendirmeden esinlenerek evrimsel hesaplama olarak adlandırılan bir şemsiye altında gruplandırılmıştır. Evrimsel hesaplamanın ana elemanları aşağıda tanımlanmaktadır; [5]

- (1) Değerlendirme stratejileri
- (2) Evrimsel programlama
- (3) Genetik algoritmalar

Amaç daha iyi özelliğe sahip yeni nesiller üretmek ve arama algoritmasının alanını genişletmektir. Farklı uygulamalarda farklı operatörler kullanılmakla birlikte genetik algoritmada 3 standart operatör kullanılır.[6] Ve amaç değeri yada iterasyon sonundaki en iyi değeri verir.

Bu operatörler:

- Yeniden Üretim (Reproduction)
- Çaprazlama (Crossover)
- Mutasyon (Mutation)

Genetik Algoritmanın Çalışma Prensibi

(i) Popülasyonu oluştur

1>2>3>4>5>6>7>8>9>10>1

1>2>4>5>6>3>7>9>8>10>1

2>10>3>4>7>1>8>3>6>5>2

.

.

.

5>6>7>6>3>10>9>1>4>2>5

(ii) Uygunluk değerinin hesaplanması

Yukarıda oluşturduğumuz dizide şehirlerin mesafeleri toplanır, her dizinin toplamı bütün toplama bölünerek seçilme olasılıkları bulunur. Seçilme olasılığı; rasgele seçilen bireylerin toplamı seçilme oranına eşitse ve küçükse seçilir.

Örneğin 1 dizi 2000 >0.1069
2 dizi 4800 >0.2566
3 dizi 6200 >0.3315
4 dizi 5700 >0.3048
Toplam 18700

(iii) Çaprazlama (Crossover)

Alana birinci dizi Paren1 ikinci dizide parent 2 olur ve altlarına rasgele (0,1) yazılır başlangıç sabit kalacak şekilde

Birey 1
5>6>7>8>3>10>9>1>4>2>5
Birey 2 5>7>8>3>4>9>10>1>2>1>5
Üretim değerleri 1 0 1 0 1 1 0 0 1 0 1

Çocuk
5>6>8>7>3>9>10>1>4>2>5

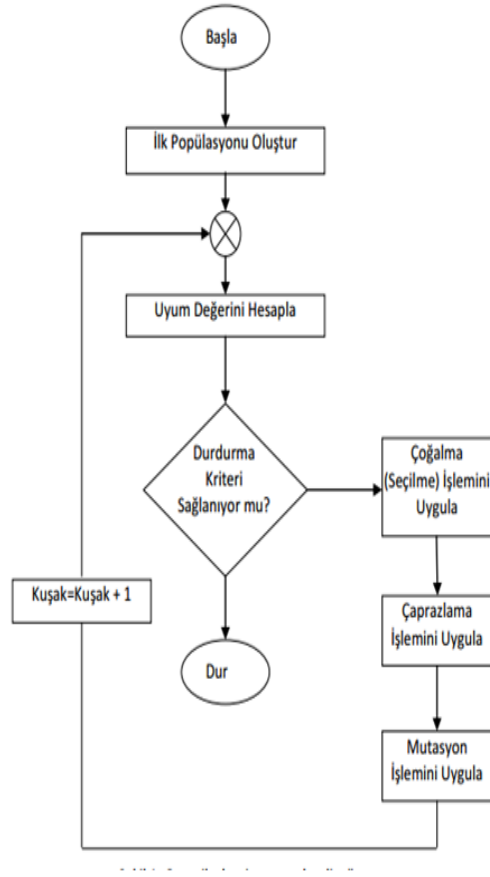
(iv) Mutasyon (Mutation)

GA algoritma gen dizilimin çeşitliliğini arttırmak için kullanılır.

5>6>8>7>3>9>10>1>4>2>5
5>6>8>7>3>9>10>1>4>2>5
5>6>4>7>3>9>10>1>8>2>5

(iv) Durdurma

Genetik algoritma işlemini sonlandırmak için çeşitli yöntemler bulunmaktadır. Bu yöntemler; algoritmanın çalışması esnasında istenen çözüm bulunduğu, GA'nın başlangıcında tanımlanan toplam iterasyon sayısına ulaşıldığında veya uygunluk değeri sürekli olarak sabit kaldığında, bulunan en iyi bireyin temsil ettiği çözüm, problem için bulunmuş en uygun çözüm olarak sunulur.



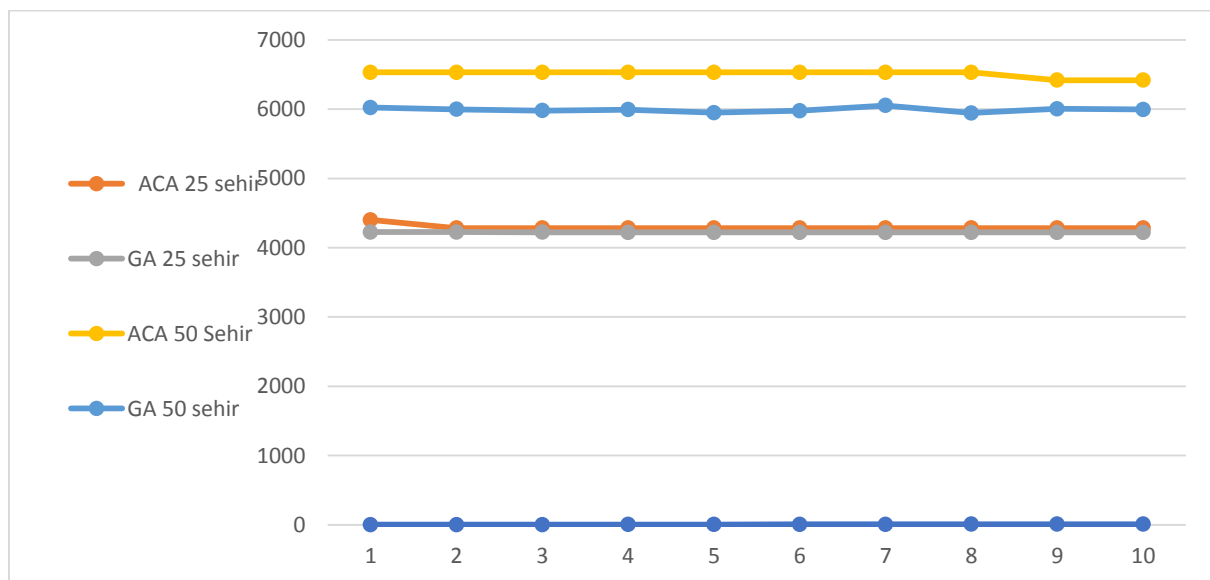
Genetik Algoritma akış diyagramı

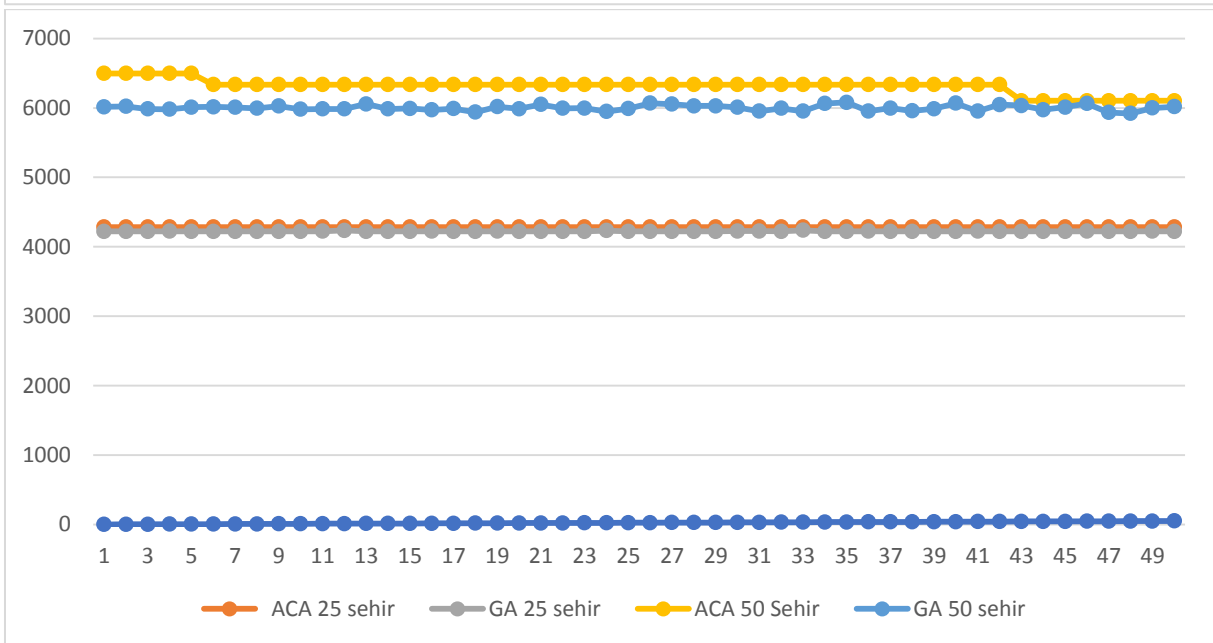
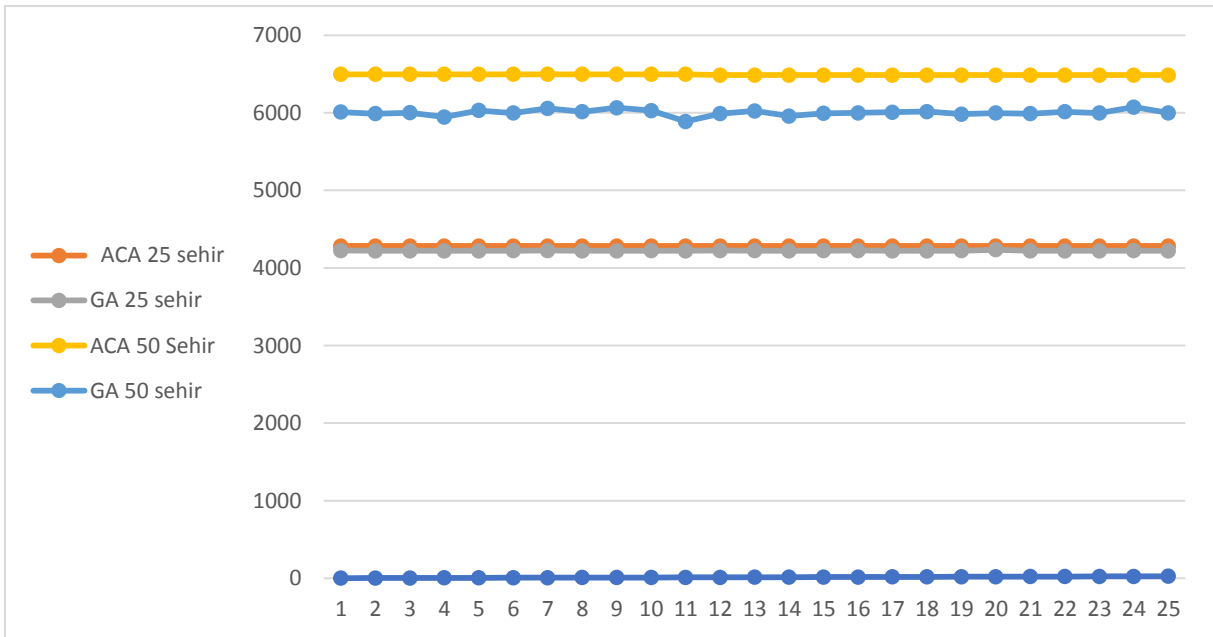
Karıncı Koloni			
Alfa=1 Beta=5 Buharlaştırma Faktörü=0,5 Rassalılık faktörü=0.01 Şehir Sayısı=25			
Deney sayısı	Best Value	Mean	Standart Dev.
10	4282,986990154700	4294,826240989630	35,517752504790
25	4282,986990154700	4282,986990154700	0,00000000000272848410531878
50	4282,986990154700	4282,986990154700	0,00000000000090949470177293
100	4247,922649081180	4260,896455278380	16,929214933891300000000000

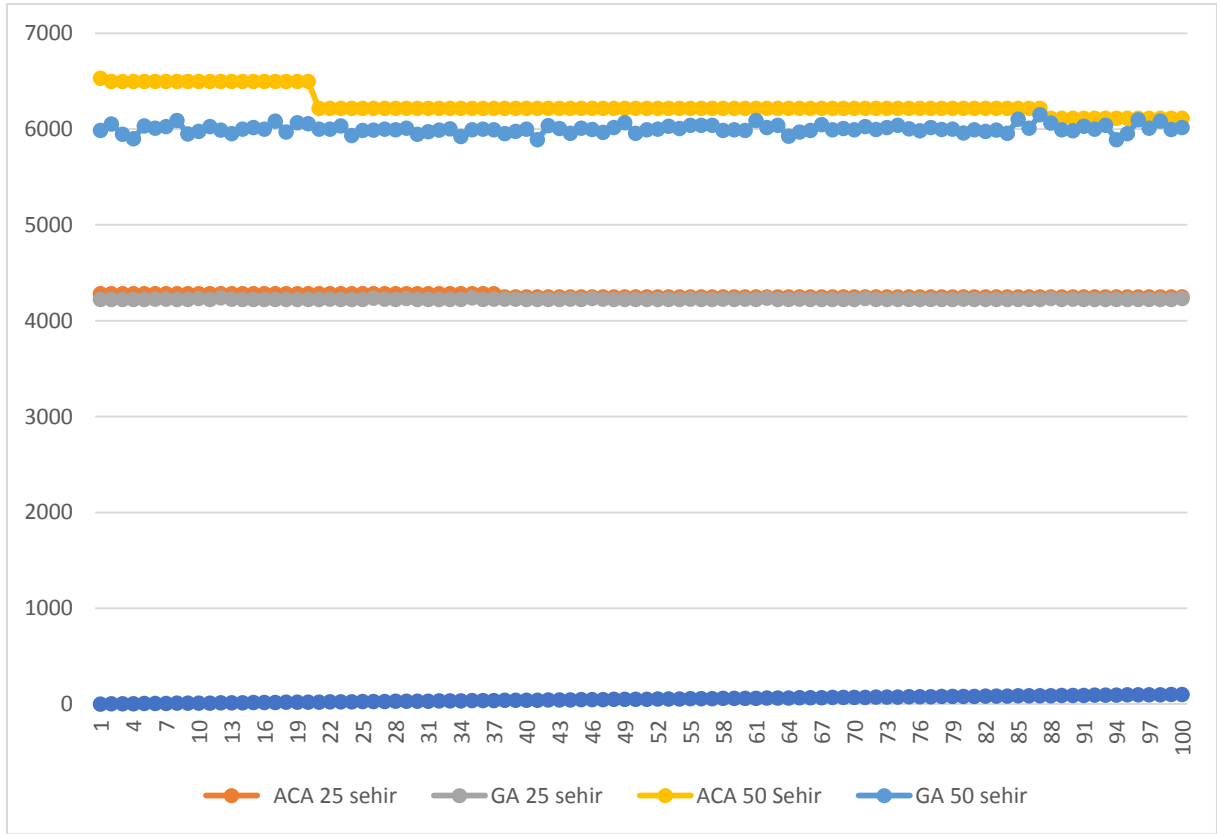
Genetik Algoritma			
Şehir Sayısı=25 mutationRate=0,5 crassoverRate=0,5			
Deney sayısı	Best Value	Mean	Standart Dev.
10	4220,6740694300700	4221,698960665230	35,517752504790
25	4220,6740694300700	4222,263165006780	2,9081542397824200000000000000
50	4282,986990154700	4220,674069430070	3,4957788364845200000000000000
100	4220,674069430070	4223,158009598360	3,9630392262082900000000000000

Karıncı Kolonisi			
Alfa=1 Beta=5 Buharlaştırma Faktörü=0,5 Rassalılık faktörü=0.01 Şehir Sayısı=50			
Deney sayısı	Best Value	Mean	Standart Dev.
10	6417,145942302210	6508,391320395420	45,6241889777662000
25	6485,241503519470	6490,473025699280	5,901952993
50	6101,874592822100	6314,705221543730	104,466356
100	6110,733391735080	6257,666628999670	125,34848214195500

Genetik Algoritma			
Şehir Sayısı=50 mutationRate=0,5 crassoverRate=0,5			
Deney sayısı	Best Value	Mean	Standart Dev.
10	5944,5614088190100	5991,283293166410	30,33486187
25	5888,3268371584500	6002,119633024080	36,68317303
50	5920,5337762446400	6001,6267780477800	37,5765541567369
100	5888,4836134565700	6001,5803599964000	43,79268626249840

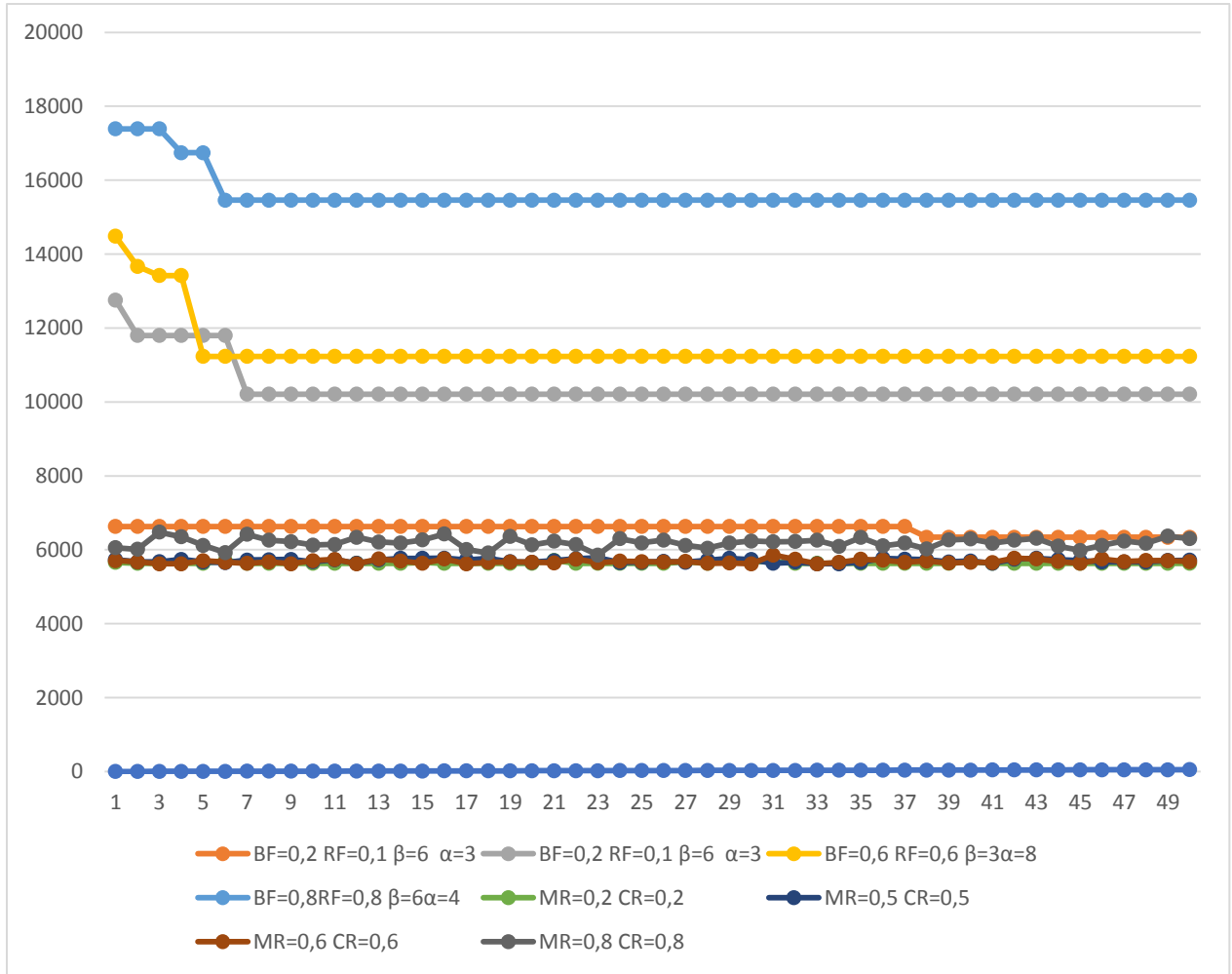






Karıncı Kolonisi				
Şehir Sayısı=50 Deney Sayısı =50				
Deney sayısı	Best Value	Mean	Standart Dev.	
BF=0,2 RF=0,1 $\beta=6$ $\alpha=3$	6340,670922884150	6553,825753491210	126,3473080756530	
BF=0,5 RF=0,5 $\beta=5$ $\alpha=8$	10212,63776156380	10422,27634282940	580,9555427880330	
BF=0,6 RF=0,6 $\beta=3$ $\alpha=8$	11232,14847511960	11433,36868880450	693,4904430586420	
BF=0,8 RF=0,8 $\beta=6$ $\alpha=4$	15461,40493393310	15628,51061905030	511,2519281172580	

Genetik Algoritma				
Şehir Sayısı=50				
Deney sayısı	Best Value	Mean	Standart Dev.	
MR=0,2 CR=0,2	5632,8666446243700	5641,123948942770	19,2972026128848000	
MR=0,5 CR=0,5	5888,3268371584500	6002,119633024080	36,68317303	
MR=0,6 CR=0,6	5622,6424498048800	5704,603224899830	45,30200618855160	
MR=0,8 CR=0,8	5856,4799176579700	6190,910002527340	134,0626131980750	



MR Mutasyon oranı

CR Corosover oranı

BF Buharlařma oranı

RF Rasgelelik oranı

Sonuç

Karıncı Algoritması az řehirlerde GA'ya göre daha hızlı sonuç verir. Fakat řehir sayısı artıkca local optimuma takılması sebebi ile iyi sonuçlar vermeyebilir . Feromen ve rasgelelik özelliđi büyütülmesi alfa betanın küçük tutulması negatif yönde etki eder ve iyi sonuç vermez. GA daha determanistik olarak ilerlemekte itarasyon sayısının fazla olmasından daha çok mutasyon yada seçilme olasılıđının normal deđerler

üzerinden seçilmesi durumunda daha kararlı ve iyi sonuç verir.

Kaynaklar

- [1] Ugur Aybars, Aydın Dogan “Ant System Algoritmasının Java İle Görselleştirilmesi”
- [2] I Osman and J Kelly. Meta-Heuristics: Theory & Applications. Kluwer Academic Publishers, Boston, 1996.
- [3] Marco Dorigo, Thomas Sttzle, Ant Colony Optimization. Bradford Books. July 1st 2004.
- [4] Dorigo, M., Gambardella, L.M., “Ant Colonies for the Travelling Salesman Problem”, Biosystems, 43 (2), 73–81, 1997
- [5] KARR, L. C., FREEMAN L. M. (1999), *Industrial Applications of Genetic Algorithms*, CRC Prees. 350 p.
- [6] Dr. İŞÇİ Öznur, Prof. Dr. KORUKOĞLU Serdar “Genetik Algoritma Yaklaşım ve Yöneylem Araştırmasında Bir Uygulama” *YÖNETİM VE EKONOMİ* Yıl:2003 Cilt:10 Say :2
Uygulamanın kaynak kodu: <https://github.com/ulasdemirci/KarincaveGenetikAlgoritmasi>

GENETİK ALGORİTMALARIN FARKLI ÇAPRAZLAMA TEKNİKLERİYLE

DERS PROGRAMI OLUŞTURMA PROBLEMLERİNE UYGULANIŞI

Sedrettin Çalışkan¹

ÖZET

Bu çalışmada, farklı çaprazlama teknikleri kullanan genetik algoritmalar (GA) ile en optimize ders programı oluşturma problemine Java ortamında çözüm geliştirilmiştir. Sınıf kullanımının etkinliği ve öğretim görevlilerinin zaman tercihleri gibi bazı kısıtlamalar göz önüne alınarak, verilen kısıtlamalar için optimize edilmiş bir çözüm bulmaya çalışılmıştır. Çalışma sonucunda aynı problem için, farklı çaprazlama tekniklerinin birbirinden çok farklı sonuçlar verdiği görülmüştür. Tüm nesil boyunca her çaprazlama tekniği için elde edilmiş uygunluk değerlerinin aritmetik ortalamalarının ve standart sapmalarının frekansları, en iyi sonucun tek noktadan çaprazlama yöntemine dayalı çaprazlama tekniği ile, en kötü sonucun ise çift noktalı çaprazlama tekniği ile elde edildiğini göstermektedir.

Anahtar kelimeler : Genetik algoritmalar, Çaprazlama teknikleri, Uniform çaprazlama, Tek noktadan çaprazlama, Çift noktalı çaprazlama.

1.GİRİŞ

Genetik algoritma klasik metodlarla çözülmesi güç hatta imkansız olan polinomsal zamanda çözülemeyen sorunların çözümünde kullanılır. Genetik algoritmalar birçok sahada uygulanabilmektedirler. Bunlardan bazıları deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleri olarak bilinir. Mühendislik alanında birçok problemde optimizasyon tekniği olarak kullanılan genetik algoritmalar; otomasyon sistemlerinde mekanizma tasarımında, geleneksel kontrol problemlerinde, güç sistemlerinde, görüntü işleme tekniklerinde, üretim hattı yerleşimi planlaması gibi optimizasyon gerektiren problemlerde ve daha birçok alanda kullanılmaktadır. [1]

Genetik algoritma ile bir problemi çözebilmek için ilk olarak rastgele başlangıç çözümleri belirlenmektedir. Bu çözüm kümesi popülasyon olarak adlandırılır. Bir öncekinden daha doğru olacağı beklenen yeni bir popülasyon meydana getirmek için bir popülasyondan alınan sonuçlar kullanılır. Bu amaçla bu çözümler için performanslar hesaplanır ve çözümler

¹ Mühendislik ve Fen Bilimleri Enstitüsü, Fatih Sultan Mehmet Vakıf Üniversitesi, Beyoğlu, İstanbul, sdrtnclskn@gmail.com

birbirleriyle eşleştirilerek yeni çözümler oluşturulur. Algoritmada üretilen çok sayıda çözümden performansı yüksek olanlar aranmaktadır. Bu arama, yeterince iyi çözüm üretilinceye kadar devam etmektedir. Genetik algoritmalar ile problemlerin çözülmesinde tasarlanan sonucu üretecek niteliklerin, kalıtım yolu ile başlangıç çözümlerinden elde edilen yeni çözümlere, onlardan da daha sonraki çözümlere geçtiği kabul edilmektedir. [2]

Bu çalışmada ele alınan tipteki problemler, çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte ve polinomsal zamanda çözüme ulaşmak mümkün olmamaktadır. Genetik algoritmalarla ise, kısa bir sürede kabul edilebilir bir sonuç bulunabilmektedir.

Yapılan bu çalışmada genetik algoritmalar kullanılarak sınıfların boyutunu, ders sayılarını, derse kaydolması beklenen öğrenci sayılarını ve dersi verecek olan öğretim üyesini girdi olarak alan ve verilen sınırlamalar altında dönemlik ders programı hazırlayan bir optimizasyon süreci geliştirilmiştir. Genetik Algoritma, olası çözümler içinde en iyisini aramakta ve optimele yakın makul çözümler üretmektedir.

Bölüm 2'de GA (Genetik Algoritma) tanımı, parametreler ve akış diyagramı tanımları yapılmıştır. **Bölüm 3**'de iyi ders programı planı oluşturulması ve permütasyona dayalı problemlerde kullanılan farklı çaprazlama teknikleri anlatılmaktadır. **Bölüm 4**'de farklı çaprazlama teknikleri için elde edilen sonuçlar verilmiştir. Son olarak da **Bölüm 5**'de çalışmaya dair sonuçlar tartışılmıştır.

2. GENETİK ALGORİTMALAR

Genetik Algoritma (GA) ilk defa Holland tarafından 1975'te kullanılmıştır ve canlı sistemlerdeki genetik şifre mantığı kullanılarak sezgisel olarak en iyi çözümü veya en iyi çözüme yakın olabilecek bir sonuç bulmayı hedefler. Bu mantık doğal seçim yani güçlü bireyin hayatta kalma olasılığının yüksek olmasıdır. Bu yöntemle evrim sonucu hayatta kalan birey en iyi sonuç olarak alınır.[3]

Genetik algoritmalar kromozom, gen, popülasyon, uygunluk oranı, seçim, çaprazlama ve mutasyon gibi bir dizi parametreler ile karakterize edilmektedir.

a) *Kromozom ve Gen*

Genetik algoritmanın çözümü istenen problemin her bir çözümünü göstermektedir. Kromozomlar, bu çözümleri gösterir. Bir problem için çok sayıda çözüm olabilir. Genetik

algoritmada bunların arasındaki en iyi çözümü arayıp bulması istenir. Kromozom elemanlarından her birisi çözümün bir özeliğini göstermektedir. Bunlara da gen denilmektedir.

b) Popülasyon

GA(Genetik Algoritma) da işleme koyulacak tüm kromozomlar birleşerek popülasyonu oluşturmaktadır. Toplum büyüklüğünü belirlemek, tüm GA kullanıcıları açısından oldukça önemli bir aşamadır. Toplum büyüklüğü ya da birey sayısı, GA'nın başarısını ya da optimum sonuca erişim süresini belirler.

c) Uygunluk Fonksiyonu

Popülasyondaki kromozomların performans derecelerini ölçülmesini sağlayan bir fonksiyondur. Her problem için bir uygunluk fonksiyonunun belirlenmesi gerekmektedir. Bir kromozom(birey) için fonksiyonun verdiği değer ne kadar büyük ise o kromozom o kadar sağlıklıdır ve iyi bir çözümdür.

d) Seçim

Seçim işlemin amacı, uygunluk değerleri yüksek bireylerin nesiller aracılığıyla daha çok üretilmelerine imkan vermektir. Bu çalışmada kullanılan Turnuva yöntemidir. Turnuva, yığından rastsal olarak bir grup dizi seçilir. Bu grup içindeki en iyi uygunluk değerine sahip dizi yeni yığına kopyalanır. Bu işlem kullanıcı tarafından önceden kararlaştırılan çevrim sayısı kadar tekrarlanır. [4]

e) Çaprazlama

Problemin çözüm havuzunda bulunan çözümleri ikişer ikişer birleştirerek yeni çözümler üretmektedir. Bu problemde üç çaprazlama metodu kullanılmıştır. Tek noktalı çaprazlama, İki noktalı çaprazlama, Tekdüze (Uniform) çaprazlama teknikleri kullanılmıştır.

f) Mutasyon

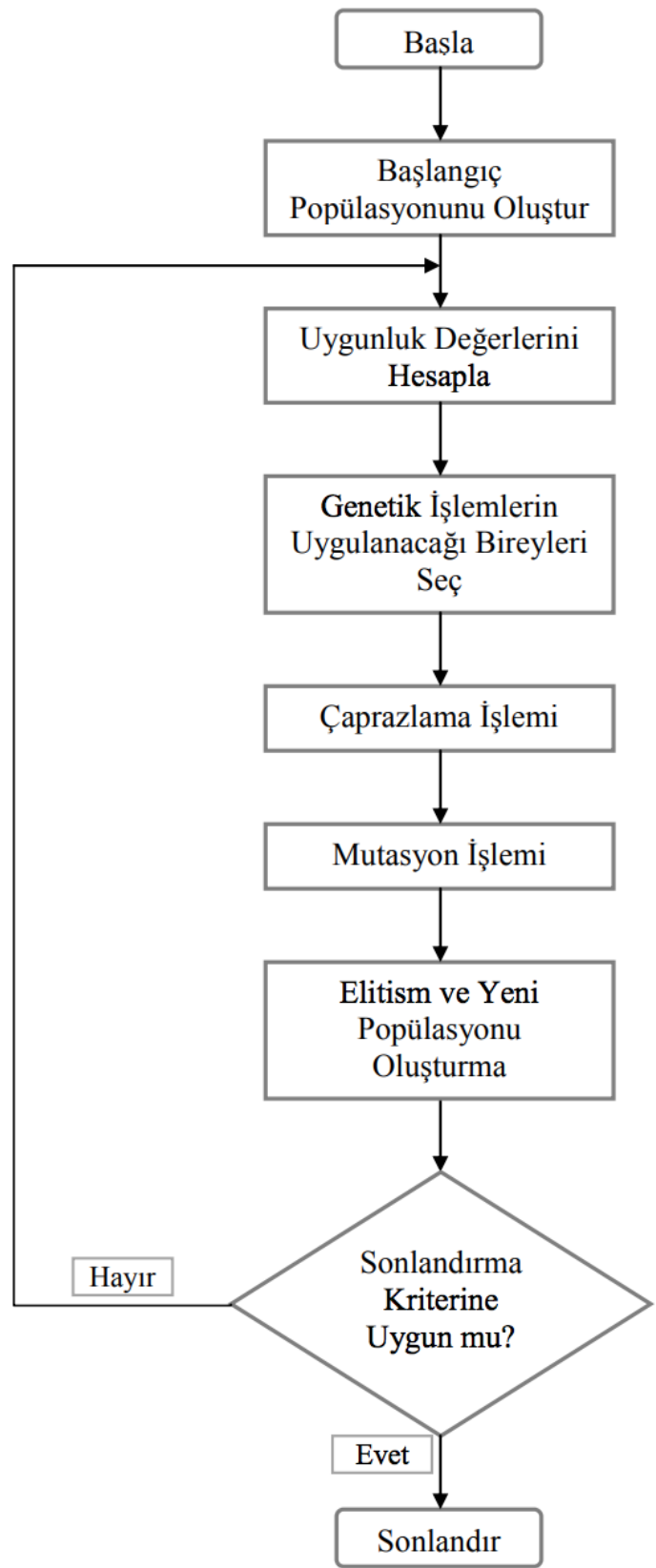
Çaprazlama sonucunda farklı çözümlere ulaşmak bazen zor olmaktadır. Yeni çözüm aramanın kolaylaştırılması ve aramanın yönünü değiştirmek amacı ile kromozomun bir elemanın (genin) rastlantısal olarak değiştirilmesi işlemidir.

Genetik algoritmanın akış diyagramı (şekil 2.2) de gösterildiği gibidir.

Genetik Algoritmanın Çalışma Prensipleri

Tipik bir GA aşağıdaki aşamaları gerçekleştirir :

1. N kromozomdan oluşan başlangıç toplumunun rastgele oluşturulması.
2. Toplumdaki her bir kromozoma ait $f(i)$ uygunluk değerinin hesaplanması.
3. Yeni toplum oluşuncaya kadar (n-birey oluşuncaya kadar) aşağıdaki adımları izleyerek çözümlerin yapısının genetik işlemlerle değiştirilmesi gerekmektedir.
 - i) Bir ebeveyn çiftinin mevcut toplumdan seçimidir. Seçilme olasılığı, uygunluk fonksiyonuna bağlı olarak artar ya da azalır.
 - ii) Yeni bir bireyi oluşturmak için, bir ebeveynin bir çaprazlama olasılığı ile çaprazlanması. Eğer çaprazlanma olmazsa, yeni ürün anne ya da babanın bir kopyası olacaktır.
 - iii) Yeni ürünün mutasyon olasılığına göre kromozom içindeki konumu (lokus) değiştirilir ve bu yeni ürün, yeni topluma dâhil edilir. Örnek olarak ikili sayı sisteminde kodlanan kromozomda mutasyon, mevcut 0 değerini 1 olarak değiştirmek demektir.
4. Yeni popülasyona (toplumu) oluştururken mevcut toplumun en iyi birkaç kromozomunu (bireyini) çaprazlama ya da mutasyon işlemine sokmadan (hiçbir değişime uğratmadan) yeni popülasyona (topluma) eklemek.
5. Mevcut toplumun, oluşturulan yeni toplum ile değiştirilmesi.
6. İkinci adıma gidilmesi. İkinci ve altıncı adımlar arasında gerçekleştirilen her bir yineleme, bir jenerasyon ya da yeni toplumu oluşturan bir üreme sürecidir. Her bir üreme sürecinde bir ya da daha fazla uygun kromozom oluşacaktır. Bu oluşum ise, her yeni toplumun bir öncekinden daha iyi olacağını göstermektedir [5]



Şekil 2.2. Genetik Algoritma Akış Diyagramı

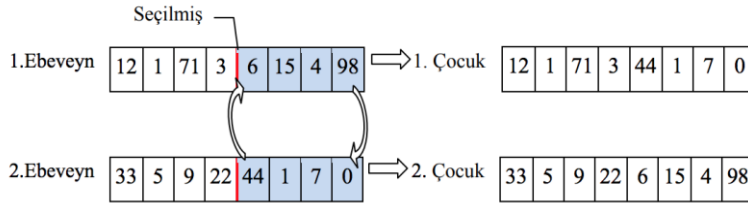
3. ÇAPRAZLAMA TEKNİKLERİ VE DERS PROGRAMI BELİRLENMESİ

Çaprazlama Teknikleri

Mevcut gen havuzunun potansiyelini araştırmak üzere, bir önceki kuşaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Ders programı oluşturma probleminde üç tip farklı çaprazlama operatörü kullanılmaktadır. Bu çaprazlama teknikleri: Tek noktalı çaprazlama, İki noktalı çaprazlama, Tekdüze (Uniform) çaprazlama şeklindedir.

a) Tek noktalı çaprazlama

Tek noktalı çaprazlamada, rastgele seçilen kromozom çiftinde, çaprazlama yapılacak bölge rastgele seçilerek çaprazlama yapılmaktadır. Şekil 3.1’de böyle bir tek noktalı çaprazlama işlemi gösterilmiştir.[4]

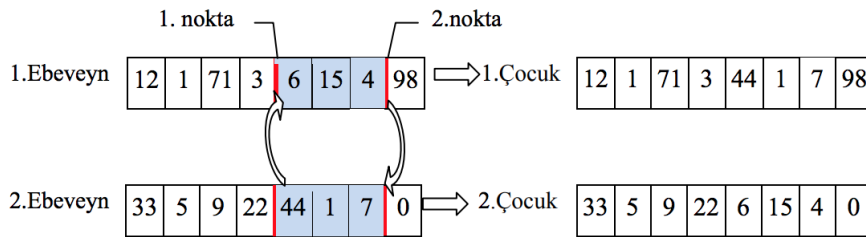


Şekil 3.1 Tek noktalı çaprazlama

b) İki noktalı çaprazlam

İki noktalı çaprazlama da, kromozom eşleri iki farklı yerden kesilerek üç parçaya ayrılmaktadır.

Parçalar karşılıklı olarak yer değiştirilerek çaprazlama yapılmaktadır. Çaprazlama tek bir parçaya uygulanırsa iki yeni kromozom elde edilmektedir. Rastgele seçilen iki parçaya uygulanırsa, dört yeni kromozom elde edilmektedir. Şekil 3.2’ de böyle bir İki noktalı çaprazlama işlemi gösterilmiştir. [5]

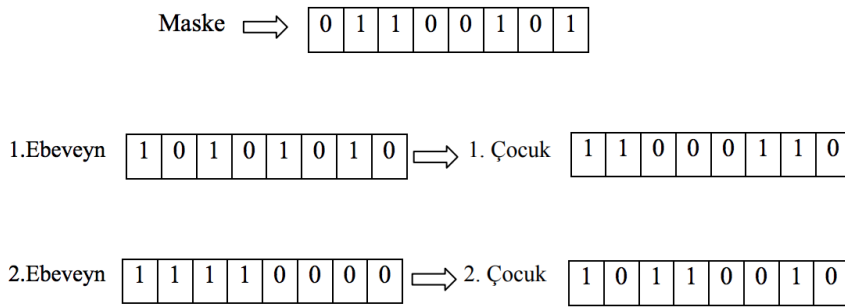


Şekil 3.2 İki noktalı çaprazlama

c) Tekdüze (Uniform) çaprazlama

Tekdüze çaprazlama yöntemi, yeni oluşturulan kromozomlar üzerindeki bütün genlerin, ebeveyn kromozomlardan taşınma olasılıklarının eşitliği ilkesine dayanmaktadır. İkili sayı sisteminde ve tekdüze çaprazlama yönteminde oluşturulan maske yerine, 0–1 arasında rastgele olasılık değerlerinden oluşan diziler kullanılabilir. Çaprazlama ile oluşturulan kromozomun geni, olasılık değeri 0,5'in altında ise anneden, 0,5'in üstünde ise babadan taşınmaktadır.

Şekil 3.3'de tekdüze çaprazlama modeli görülmektedir.[5]



Şekil 3.3 Tekdüze çaprazlama

Ders Programı Belirlenmesi

Ders programı problemine getirilen, GA'da farklı çaprazlama teknikleri ile çözüm arayışında istenilen probleme ilişkin sınır değerleri ve bazı tanımlamalar şu şekildedir:

Farklı dersliklerdeki zaman boşluklarının, öğrenci şubelerinin, öğretim elemanlarının ve derslerin koordinasyonunu içermektedir. Derslerin çizelgeye yerleştirilmesi ve problemin çözümü, şube sayısına, öğretim elemanı sayısına, derslik sayısına, her bir öğretim elamanının özel kısıtlarına, gün sayısına ve her bir gündeki saat sayısına bağlı olarak şekillenmektedir.

Bu çalışmada ele alınan ders programı özelde Fatih Sultan Mehmet Vakıf Üniversitesi Fen Bilimler Enstitüsü , Bilgisayar Mühendisliği, Elektrik-Elektronik Mühendisliği ve Biyomedikal Mühendisliği bölümü için geliştirilmiştir. Ancak kolaylıkla diğer bölüm veya fakülteler için de uygulanabilecek özellikte geliştirilmiştir. Test amaçlı gerçekleştirilen ders programında Bilgisayar, Elektrik-Elektronik, Biyomedikal bölümüne ait toplamda 7 dersliğe,

6 adet akademik personele , 3 departman ve hafta içi belli saatlere göre boş dersliklere ait ders yerleştirilmesi sağlanmıştır.

Temel olarak üniversite ders çizelgeleme programı için 3 ana kısıt bulunmaktadır. Bu kısıtlar şöyle sıralanabilir:

- Şubelerin Çakışması
- Dersliklerin Çakışması
- Öğretim Elemanlarının Çakışması

Uygunluk değer fonksiyonu içinde bu ana kısıtlar ve özel kısıtlarının her biri için öncelik sırasına göre puanlama sistemi bulunmaktadır. Uygunluk değeri(Fitness) çakışma sayısına göre değeri değişmektedir. Çakışma sayısının olmadığı değer bizim Fitness değerimizin aranan çözüme olduğudur. Bulunan çözümler, olması zorunlu kısıtları yerine getirmekle beraber, optimize etmek için kullandığımız girdilerinde birçoğunu sağlayan çözümler üretmektedir. [4]

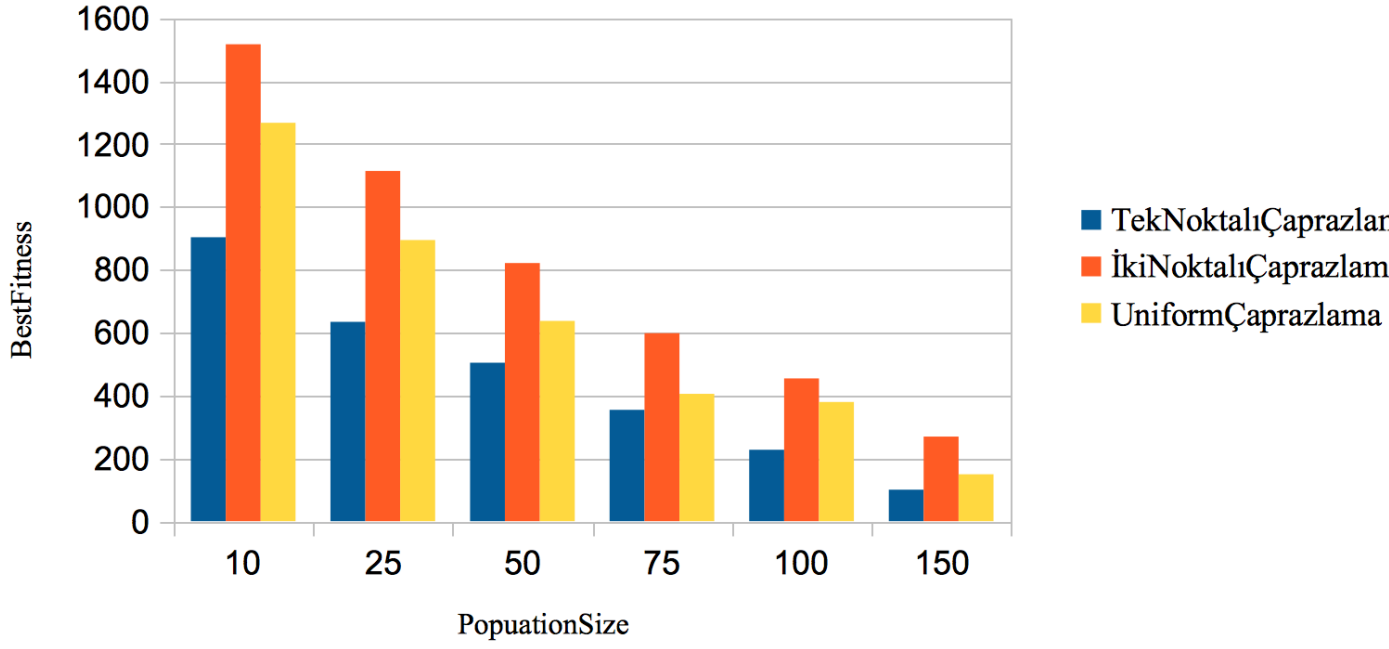
4. FARKLI ÇAPRAZLAMA TEKNİKLERİ İÇİN ELDE EDİLEN SONUÇLAR

Tek noktadan çaprazlama yöntemi ortalama 453.67 sonucunda Best Fitness değerine ulaşırken Uniform 621.83, İki noktalı çaprazlama ise 795.5 ortalama sonucunda Best Fitness değerine ulaşmıştır. Bu ortalama değerlerden varılan sonuç, tek noktadan çaprazlama yöntemi ile daha iyi sonuç alınmıştır.

Çaprazlama Teknikleri	Ortalama	Standart Sapma	En İyi
TekNoktalıÇaprazlama	453,67	290,98	354
İkiNoktalıÇaprazlama	795,5	459,45	821
<u>UniformÇaprazlama</u>	621,83	404,99	637

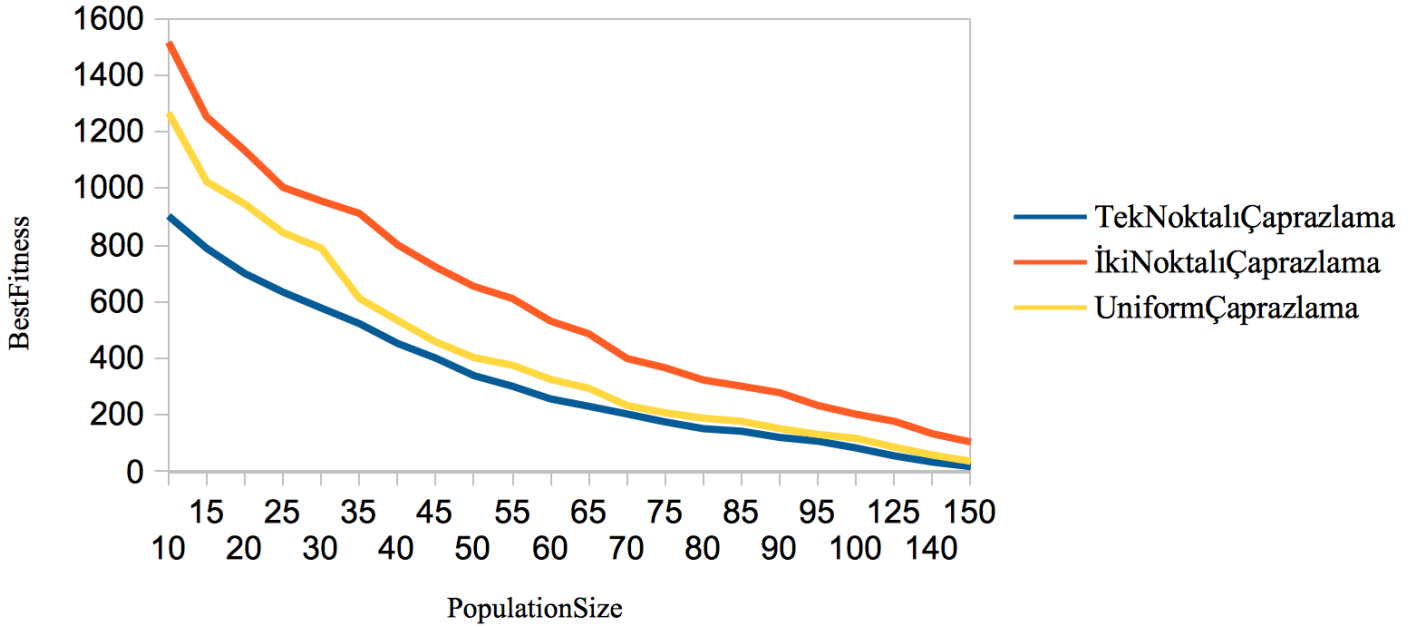
Şekil 4.1. Her çaprazlama tekniği için ortalama değer, standart sapma ve en iyi değerlerin ortalamasından oluşmaktadır.

Karşılaştırma Tabloları



Şekil 4.2. Population Size göre her çaprazlama tekniğinin karşılaştırması.

Yakınsama Grafiği



Şekil 4.3. Population Size göre her çaprazlama tekniğinin yakınsama grafiği.

```

> Generation # 145
Schedule # |
Classes [dept,class,room,instructor,meeting-time]
| Fitness | Conflicts
1305 | [ BLM,C1 , R2,I1, MT4],[ BLM,C4 , R3,I3, MT4],[ EEM,C2 , R3,I4, MT1],[ EEM,C4 , R2,I3, MT3],[ EEM,C5 , R3,I4, MT2],[ BMD,C6 , R3,I1, MT3],[ BMD,C7 , R2,I2, MT2] | 1,00000 | 0
1306 | [ BLM,C1 , R2,I4, MT3],[ BLM,C4 , R1,I4, MT4],[ EEM,C2 , R3,I2, MT2],[ EEM,C4 , R2,I3, MT2],[ EEM,C5 , R3,I4, MT3],[ BMD,C6 , R3,I1, MT1],[ BMD,C7 , R1,I2, MT3] | 0,33333 | 2
1307 | [ BLM,C1 , R3,I1, MT1],[ BLM,C4 , R2,I3, MT1],[ EEM,C2 , R3,I2, MT2],[ EEM,C4 , R2,I4, MT1],[ EEM,C5 , R2,I4, MT2],[ BMD,C6 , R2,I1, MT3],[ BMD,C7 , R1,I4, MT3] | 0,25000 | 3
1308 | [ BLM,C1 , R1,I1, MT3],[ BLM,C4 , R1,I3, MT2],[ EEM,C2 , R3,I3, MT4],[ EEM,C4 , R2,I3, MT3],[ EEM,C5 , R3,I4, MT1],[ BMD,C6 , R3,I3, MT4],[ BMD,C7 , R3,I4, MT2] | 0,20000 | 4
1309 | [ BLM,C1 , R3,I2, MT4],[ BLM,C4 , R3,I4, MT2],[ EEM,C2 , R2,I2, MT4],[ EEM,C4 , R2,I4, MT3],[ EEM,C5 , R3,I4, MT3],[ BMD,C6 , R2,I3, MT2],[ BMD,C7 , R2,I4, MT1] | 0,20000 | 4
1310 | [ BLM,C1 , R2,I2, MT3],[ BLM,C4 , R3,I4, MT2],[ EEM,C2 , R2,I2, MT1],[ EEM,C4 , R1,I3, MT4],[ EEM,C5 , R2,I4, MT3],[ BMD,C6 , R2,I1, MT2],[ BMD,C7 , R3,I4, MT2] | 0,12500 | 7
1311 | [ BLM,C1 , R2,I1, MT2],[ BLM,C4 , R2,I3, MT4],[ EEM,C2 , R2,I2, MT1],[ EEM,C4 , R1,I4, MT4],[ EEM,C5 , R2,I4, MT4],[ BMD,C6 , R2,I1, MT3],[ BMD,C7 , R1,I4, MT3] | 0,12500 | 7
1312 | [ BLM,C1 , R3,I2, MT3],[ BLM,C4 , R3,I3, MT4],[ EEM,C2 , R2,I4, MT3],[ EEM,C4 , R1,I3, MT1],[ EEM,C5 , R1,I4, MT4],[ BMD,C6 , R1,I3, MT1],[ BMD,C7 , R3,I4, MT3] | 0,11111 | 8
1313 | [ BLM,C1 , R1,I2, MT3],[ BLM,C4 , R1,I4, MT1],[ EEM,C2 , R1,I3, MT1],[ EEM,C4 , R3,I4, MT1],[ EEM,C5 , R3,I4, MT1],[ BMD,C6 , R1,I1, MT3],[ BMD,C7 , R3,I2, MT2] | 0,09091 | 10

Class # | Dept | Course (number, max # of students ) | Room (Capacity) | Instructor (Id) | Meeting Time (Id)
01 | BLM | 325K(C1, 25 ) | R2(25 ) | Dr.Ebubekir Koç ( I1 ) | SPH 10:30 - 12:00(MT4)
02 | BLM | 464K(C4, 25 ) | R3(35 ) | Prf.Dr.Fevzi Yılmaz ( I3 ) | SPH 10:30 - 12:00(MT4)
03 | EEM | 319K(C2, 35 ) | R3(35 ) | Dr.Berna Kiraz ( I4 ) | PÇC 09:00 - 10:00(MT1)
04 | EEM | 464K(C4, 25 ) | R2(25 ) | Prf.Dr.Fevzi Yılmaz ( I3 ) | SPH 09:00 - 10:30(MT3)
05 | EEM | 360K(C5, 35 ) | R3(35 ) | Dr.Berna Kiraz ( I4 ) | PÇC 10:00 - 11:00(MT2)
06 | BMD | 303K(C6, 35 ) | R3(35 ) | Dr.Ebubekir Koç ( I1 ) | SPH 09:00 - 10:30(MT3)
07 | BMD | 303LK(C7, 20 ) | R2(25 ) | Prf.Dr.Burhannetin Can ( I2 ) | PÇC 10:00 - 11:00(MT2)
> Solution Found in 146 generations

```

Şekil 4.4. Tek Noktalı Çaprazlama Metodu için oluşan örnek ders programı.

En iyi fitness değerini **146 nesilde bulmuş**. *Fitness değeri*= 1 / (Çatışma Sayısı)

```

> Generation # 411
Schedule # |
Classes [dept,class,room,instructor,meeting-time]
| Fitness | Conflicts
3699 | [ BLM,C1 , R2,I4, MT4],[ BLM,C4 , R2,I3, MT3],[ EEM,C2 , R3,I2, MT1],[ EEM,C4 , R3,I3, MT4],[ EEM,C5 , R3,I4, MT2],[ BMD,C6 , R3,I1, MT3],[ BMD,C7 , R1,I2, MT2] | 1,00000 | 0
3700 | [ BLM,C1 , R3,I1, MT4],[ BLM,C4 , R3,I3, MT2],[ EEM,C2 , R3,I4, MT3],[ EEM,C4 , R2,I3, MT3],[ EEM,C5 , R1,I4, MT4],[ BMD,C6 , R2,I1, MT2],[ BMD,C7 , R3,I4, MT1] | 0,33333 | 2
3701 | [ BLM,C1 , R1,I2, MT1],[ BLM,C4 , R3,I4, MT3],[ EEM,C2 , R3,I4, MT2],[ EEM,C4 , R1,I3, MT2],[ EEM,C5 , R2,I4, MT4],[ BMD,C6 , R3,I1, MT4],[ BMD,C7 , R2,I2, MT2] | 0,25000 | 3
3702 | [ BLM,C1 , R3,I2, MT1],[ BLM,C4 , R3,I4, MT3],[ EEM,C2 , R3,I2, MT4],[ EEM,C4 , R2,I4, MT2],[ EEM,C5 , R2,I4, MT4],[ BMD,C6 , R1,I3, MT2],[ BMD,C7 , R2,I2, MT2] | 0,25000 | 3
3703 | [ BLM,C1 , R2,I1, MT3],[ BLM,C4 , R2,I3, MT1],[ EEM,C2 , R1,I2, MT4],[ EEM,C4 , R3,I4, MT4],[ EEM,C5 , R2,I4, MT4],[ BMD,C6 , R3,I1, MT1],[ BMD,C7 , R2,I4, MT1] | 0,20000 | 4
3704 | [ BLM,C1 , R3,I1, MT1],[ BLM,C4 , R1,I3, MT3],[ EEM,C2 , R3,I3, MT3],[ EEM,C4 , R1,I4, MT2],[ EEM,C5 , R3,I4, MT2],[ BMD,C6 , R3,I1, MT3],[ BMD,C7 , R1,I2, MT1] | 0,16667 | 5
3705 | [ BLM,C1 , R1,I2, MT2],[ BLM,C4 , R3,I4, MT1],[ EEM,C2 , R2,I3, MT3],[ EEM,C4 , R1,I4, MT3],[ EEM,C5 , R1,I4, MT4],[ BMD,C6 , R3,I3, MT2],[ BMD,C7 , R1,I4, MT3] | 0,14286 | 6
3706 | [ BLM,C1 , R2,I4, MT3],[ BLM,C4 , R1,I3, MT3],[ EEM,C2 , R1,I4, MT2],[ EEM,C4 , R3,I4, MT4],[ EEM,C5 , R3,I4, MT1],[ BMD,C6 , R2,I3, MT3],[ BMD,C7 , R3,I4, MT3] | 0,14286 | 6
3707 | [ BLM,C1 , R2,I1, MT2],[ BLM,C4 , R1,I3, MT3],[ EEM,C2 , R2,I4, MT3],[ EEM,C4 , R1,I4, MT3],[ EEM,C5 , R3,I4, MT3],[ BMD,C6 , R3,I3, MT4],[ BMD,C7 , R1,I2, MT4] | 0,12500 | 7

Class # | Dept | Course (number, max # of students ) | Room (Capacity) | Instructor (Id) | Meeting Time (Id)
01 | BLM | 325K(C1, 25 ) | R2(25 ) | Dr.Berna Kiraz ( I4 ) | SPH 10:30 - 12:00(MT4)
02 | BLM | 464K(C4, 25 ) | R2(25 ) | Prf.Dr.Fevzi Yılmaz ( I3 ) | SPH 09:00 - 10:30(MT3)
03 | EEM | 319K(C2, 35 ) | R3(35 ) | Prf.Dr.Burhannetin Can ( I2 ) | PÇC 09:00 - 10:00(MT1)
04 | EEM | 464K(C4, 25 ) | R3(35 ) | Prf.Dr.Fevzi Yılmaz ( I3 ) | SPH 10:30 - 12:00(MT4)
05 | EEM | 360K(C5, 35 ) | R3(35 ) | Dr.Berna Kiraz ( I4 ) | PÇC 10:00 - 11:00(MT2)
06 | BMD | 303K(C6, 35 ) | R3(35 ) | Dr.Ebubekir Koç ( I1 ) | SPH 09:00 - 10:30(MT3)
07 | BMD | 303LK(C7, 20 ) | R1(20 ) | Prf.Dr.Burhannetin Can ( I2 ) | PÇC 10:00 - 11:00(MT2)
> Solution Found in 412 generations

```

Şekil 4.5. Uniform Çaprazlama Metodu için örnek ders programı.

En iyi fitness değerini **412 nesilde bulmuş**. *Fitness değeri*= 1 / (Çatışma Sayısı)


```

> Generation # 1364
Schedule # |
Classes [dept,class,room,instructor,meeting-time]
| Fitness | Conflic
12276 | [ BLM,C1 , R2,I1, MT3],[ BLM,C4 , R2,I3, MT4],[ EEM,C2 , R3,I3, MT2],[ EEM,C4 , R2,I3, MT1],[ EEM,C5 , R3,I4, MT1],[ BMD,C6 , R3,I1, MT4],[ BMD,C7 , R2,I2, MT2] | 1,00000 | 0
12277 | [ BLM,C1 , R1,I2, MT4],[ BLM,C4 , R2,I4, MT1],[ EEM,C2 , R3,I2, MT2],[ EEM,C4 , R1,I4, MT2],[ EEM,C5 , R1,I4, MT1],[ BMD,C6 , R3,I3, MT2],[ BMD,C7 , R3,I4, MT3] | 0,16667 | 5
12278 | [ BLM,C1 , R1,I4, MT4],[ BLM,C4 , R3,I4, MT2],[ EEM,C2 , R1,I3, MT2],[ EEM,C4 , R1,I3, MT3],[ EEM,C5 , R2,I4, MT3],[ BMD,C6 , R3,I3, MT4],[ BMD,C7 , R2,I4, MT4] | 0,16667 | 5
12279 | [ BLM,C1 , R3,I2, MT2],[ BLM,C4 , R2,I3, MT1],[ EEM,C2 , R1,I4, MT3],[ EEM,C4 , R1,I3, MT1],[ EEM,C5 , R1,I4, MT4],[ BMD,C6 , R3,I1, MT3],[ BMD,C7 , R1,I2, MT2] | 0,16667 | 5
12280 | [ BLM,C1 , R2,I2, MT3],[ BLM,C4 , R1,I4, MT3],[ EEM,C2 , R1,I4, MT2],[ EEM,C4 , R2,I4, MT2],[ EEM,C5 , R3,I4, MT4],[ BMD,C6 , R2,I1, MT4],[ BMD,C7 , R2,I2, MT3] | 0,14286 | 6
12281 | [ BLM,C1 , R2,I1, MT1],[ BLM,C4 , R2,I3, MT1],[ EEM,C2 , R1,I2, MT1],[ EEM,C4 , R1,I4, MT4],[ EEM,C5 , R2,I4, MT4],[ BMD,C6 , R1,I1, MT2],[ BMD,C7 , R3,I4, MT3] | 0,14286 | 6
12282 | [ BLM,C1 , R1,I2, MT3],[ BLM,C4 , R3,I3, MT4],[ EEM,C2 , R2,I4, MT4],[ EEM,C4 , R1,I3, MT2],[ EEM,C5 , R1,I4, MT2],[ BMD,C6 , R1,I3, MT1],[ BMD,C7 , R2,I4, MT4] | 0,11111 | 8
12283 | [ BLM,C1 , R1,I4, MT4],[ BLM,C4 , R3,I3, MT2],[ EEM,C2 , R3,I4, MT1],[ EEM,C4 , R1,I4, MT2],[ EEM,C5 , R1,I4, MT2],[ BMD,C6 , R1,I3, MT1],[ BMD,C7 , R1,I4, MT1] | 0,11111 | 8
12284 | [ BLM,C1 , R2,I4, MT2],[ BLM,C4 , R2,I4, MT2],[ EEM,C2 , R1,I4, MT4],[ EEM,C4 , R1,I3, MT4],[ EEM,C5 , R1,I4, MT4],[ BMD,C6 , R2,I3, MT3],[ BMD,C7 , R1,I2, MT4] | 0,07143 | 13

Class # | Dept | Course (number, max # of students ) | Room (Capacity) | Instructor (Id) | Meeting Time (Id)
-----|-----|-----|-----|-----|-----
01 | BLM | 325K(C1, 25 ) | R2(25) | Dr.Ebubekir Koç ( I1 ) | SPH 09:00 - 10:30(MT3)
02 | BLM | 464K(C4, 25 ) | R2(25) | Prf.Dr.Fevzi Yılmaz ( I3 ) | SPH 10:30 - 12:00(MT4)
03 | EEM | 319K(C2, 35 ) | R3(35) | Prf.Dr.Fevzi Yılmaz ( I3 ) | PÇÇ 10:00 - 11:00(MT2)
04 | EEM | 464K(C4, 25 ) | R2(25) | Prf.Dr.Fevzi Yılmaz ( I3 ) | PÇÇ 09:00 - 10:00(MT1)
05 | EEM | 360C(C5, 35 ) | R3(35) | Dr.Berna Kiraz ( I4 ) | PÇÇ 09:00 - 10:00(MT1)
06 | BMD | 303K(C6, 35 ) | R3(35) | Dr.Ebubekir Koç ( I1 ) | SPH 10:30 - 12:00(MT4)
07 | BMD | 303LK(C7, 20 ) | R2(25) | Prf.Dr.Burhanettin Can ( I2 ) | PÇÇ 10:00 - 11:00(MT2)
> Solution Found in 1365 generations

```

Şekil 4.6. İki Noktadan Çaprazlama metodu için örnek ders programı.

En iyi fitness değerini **1365 nesilde bulmuş**. $Fitness\ deęeri = 1 / (\text{Çatışma Sayısı})$

5. SONUÇLAR VE ÖNERİ

Bu çalışmada, bir yapay zeka algoritması olan GA kullanılarak, çok kriterli üniversite ders programı tasarımı ve optimizasyonunu sağlayan bir program geliştirilmiştir.

Bir hafta içerisinde ders programına yerleştirilecek birçok farklı ders ve bu derslerin yerleştirileceği olası birçok zaman boşluğu vardır. GA, yapacağı araştırma ile olası çözümlerden, öğretim elemanı verimliliği ve eğitim esaslara dayalı öğrenci verimliliği kriterlerini sağlayan en uygun sistemi belirler. En uygun çözümü bulmak için programı pek çok defa çalıştırabilirsiniz GA’da farklı çaprazlam teknikleri kullanarak hazırlanan ders programı üzerinde, yapılan deneylerin sonuçlarına göre tüm nesil boyunca her çaprazlama tekniği için elde edilmiş uygunluk değerlerinin, en iyi sonucun tek noktadan çaprazlama yöntemine dayalı çaprazlama tekniği ile, en kötü sonucun ise çift noktalı çaprazlama tekniği ile elde edildiğini göstermektedir. Bu durum GA’nın oldukça verimli çalıştığını göstermiştir. [6]

İyi bir ders programında amaç, kaynakları en iyi şekilde kullanmak ve verimliliği arttırmak olduğu için geliştirilen programda, eğitim esaslara dayalı haftanın günlerine göre verimlilik ve gün içi çalışma saatlerine göre verimlilik değişimleri ile ders ağırlıklarının dikkate alan, eğitim-öğretim verimliliğini arttırmaya yönelik kısıtlar kullanılmıştır. Program için kullanılan örneğin tüm bölümlere ve fakültelere yönelik olarak genişletilmesi mümkündür. Ayrıca, çalışmaya bakılarak probleme özgü yeni genetik operatörler geliştirilebilir.

KAYNAKÇA

- [1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, 1997.
- [2] Reeves, Colin R. and Jonathan E. Rowe, “Genetic algorithms : principles and perspectives : a guide to GA theory,” Kluwer Academic, Boston, 2003.
- [3] Man, K.F., Tang, K.S. ve Kwong, S., *Genetic Algorithms*, Springer Publishing, 1999.
- [4] Bolat, B., Erol, K.O. ve İmrak, C.E., “*Mühendislik Uygulamalarında Genetik Algoritma ve Operatörlerin İşlevleri*”, Mühendislik ve Fen Bilimleri Dergisi, Sigma 2004/4.
- [5] Paksoy, S. , “*Genetik Algoritma ile Proje Çizelgeleme*”, ÇUKUROVA ÜNİVERSİTESİ Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Yayınlanmış Doktora Tezi, 2007.
- [6] Bağış, A. , (1996) “*Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi*”, ERCİYES ÜNİVERSİTESİ Fen Bilimleri Enstitüsü Elektronik Mühendisliği Bölümü Yayınlanmış Yüksek Lisans Tezi, 1996.

EK

Proje Link: <https://github.com/sdrtnclskn/GeneticAlgorithm-FsmvuSchedule>