

Gezgin Satıcı Probleminin İkili
Kodlanmış Genetik
Algoritmalarla Çözümünde Yeni
Bir Yaklaşım

Mehmet Ali Aytekin
Tahir Emre Kalaycı

Gündem

- Gezgin Satıcı Problemi
- GSP'yi Çözen Algoritmalar
- Genetik Algoritmalar
- GSP'nin GA ile Çözülmesi
 - Permütasyon Kodlama
 - İkili Kodlama
- Geliştirilen Yeni Yöntem
 - Geliştirilen Yeni Yöntemin Avantajları ve Dezavantajları
- Deneysel Sonuçlar
- Sonraki Çalışmalar

Gezgin Satıcı Problemi

- Verilen n tane şehir için
 - Her şehre bir kez uğramak
 - Ve başlanan noktaya geri dönmek
- Şartlarıyla en az maliyeti içeren turu bulma
- Tanımlanması kolay, çözümü zor bir problemdir.
 - GSP'yi çözen algoritmalar:
 - Sezgisel Algoritmalar
 - Kesin Algoritmalar

GSP'yi Çözen Algoritmalar

- Kesin Algoritmalar
 - Genellikle tamsayı doğrusal programlamaya dayanır.
 - Hesaplanabilirlik açısından pahalıdır.
 - Branch&Bound
 - Dinamik Programlama

GSP' yi Çözen Algoritmalar

- Sezgisel Algoritmalar
 - İdeale yakın bir çözüm
 - 3 ana gruba ayrılır:
 - Tur Oluşturan Sezgiseller: Greedy, Ekleme Sezgiseli,...
 - Turu Geliştiren Sezgiseller: 2-opt, Lin-Kernighan, GA,...
 - Melez Yöntemler: Iterated Lin-Kernighan, Hibrid GA,.....

Genetik Algoritmalar

- Arama ve en iyileme problemlerini çözmek için kullanılır.
- “Doğal Seçilim ve Uyarlanım” prensibi.
- Rasgele çözümler üretilir ve her bir çözümün problemi çözebilme yeteneği ölçülür.
- Problemi en iyi çözen “çözümler”den “Genetik İşleçler” yardımı ile yeni çözümler üretilir.

Genetik Algoritmalar

BEGIN GA

t:=0

populasyon P(t)' yi ilkle

P(t)'deki her bireyin uygunluk deęerini hesapla

WHILE (NOT durma-koşulu) DO

BEGIN WHILE

t :=t+1

P(t-1) den P(t) 'yi oluřtur

P(t)'ye GA uygula.

P(t)'deki her bireyin uygunluk deęerini hesapla

END WHILE

END GA

Genetik Algoritmalar

- Genetik Algoritmalar için üç önemli karar noktası vardır:
 - Kodlama (aday çözümün temsili)
 - Çözümleri Değerlendirme (Uygunluk Fonksiyonu)
 - Birey Üretimi (Çaprazlama, Mutasyon, Seçilim)

GSP'nin GA İle Çözülmesi – Permütasyon Kodlama

- Permütasyon Kodlama
 - Verilen şehir sayısı (n) kadar “gen”
 - Her gen bir şehri temsil eder
 - Genlerin dizilimi (permütasyon) o anki “turu “ temsil eder.
 - $n=5$ olsun. Örnek bir populasyon:
 - 1. kromozom: (2 3 0 1 4)
 -
 - x. kromozom: (0 2 3 1 4)
 - n. kromozom: (4 3 1 0 2)

GSP'nin GA İle Çözülmesi – Permütasyon Kodlama

- En iyi kromozomlar seçilerek Genetik İşleçlere (çaprazlamaya/mutasyona) tabii tutulur.
- (2 3 0 1 4) ve (4 3 1 0 2) kromozomlarını çaprazlayalım :
 - çaprazla[(2 3 0 1 4) , (4 3 1 0 2)] -> (2 3 1 0 2)
 - Klasik çaprazlama yöntemi
 - Hatalı kromozom oluşabilir
 - Tamir edilmeli!

GSP'nin GA İle Çözülmesi – İkili Kodlama (Lidd'in yöntemi)

- Her bir “gen” onluk değeri ile değil de ikili (binary) değeri ile temsil edilir.
- $n=5$ olsun. Her şehri ikili olarak temsil etmek için 3 bite ihtiyaç var.
- Bir önceki (2 3 0 1 4) ve (4 3 1 0 2) turlarını ikili olarak temsil edelim:
 - (010 011 000 001 100) \rightarrow (2 3 0 1 4 5)
 - (100 101 001 000 010) \rightarrow (4 3 1 0 2)

GSP'nin GA İle Çözülmesi – İkili Kodlama (Lidd'in yöntemi)

- Yukarıdaki iki kromozomu çaprazlama işlemine tabi tutalım:
 - çaprazla[(010 011 000 001 100),
(100 101 001 000 010)] - > (010 011 001 000 010)
 - Klasik Çaprazlama
 - Hatalı kromozom üretimi!
 - mutasyon[010 011 000 001 100] ->
(010 111 000 001 100)
 - Klasik Mutasyon (Rasgele bit seç, deęilini al)
 - Hatalı kromozom üretimi! (111 şehri tanımlı deęil)

GSP'nin GA İle Çözülmesi – İkili Kodlama (Lidd'in yöntemi)

- Lidd'in yöntemi 100 şehirlik örneklere kadar iyi sonuç üretmektedir.
- Fakat
 - Lidd'in yöntemiyle de sıra bağımlılığından kurtulamıyoruz.
 - Klasik GA işleçleri hatalı üretime neden oluyor.
 - Eksik şehirleri içeren kromozomlar üretilebiliyor.
 - İdeal üretim teknikleri (mutasyon, çaprazlama) atalardan bilgiyi bozmadan alabilmelidir.

Geliştirilen Yeni Yöntem

- Her şehir için ikili değerde bir puan tanımlanır.
- Kromozom bu puanlardan oluşur.
- Bu puan değerlerinin sıralanması, bizim için bir tur oluşturur.
- $n=5$ olsun.
 - Rasgele ikili dizi oluşturalım.
 - Her bir şehir için en az 3 bitlik bir puan değeri olmalı.
 - İkili dizinin toplam boyutu = $3*5$ dir.

Geliştirilen Yeni Yöntem

- Rasgele üretilen bir kromozom:
 - (000010001110111) olsun.
 - Peki biz bu kromozomdan anlamlı bir turu nasıl oluşturacağız?
- Kromozomu 3'erli gruplara ayırıyoruz:
 - (000110001010111)
 - Her bir grubun onluk sistemdeki değeri puandır.
 - O halde (0 3 1 2 7) bizim puan dizimiz olacaktır.
- Puan dizisinden geçerli bir turu iki şekilde oluşturabiliriz:

Geliştirilen Yeni Yöntem

1. puan dizisinin bir kopyasını (kopya1) oluştur ve sırala.
 - (0 3 1 2 7) -> puan dizisi
 - (0 1 2 3 7) -> kopya1 dizisi
 - kopya1'deki her bir elemanın puan dizisindeki konumuyla tur oluşturulur.
 - kopya1'deki 0 elemanı puan dizisinde 0. sırada
 - ...
 - kopya1'deki 2 elemanı puan dizisinde 3.sırada
 - O halde geçerli tur -> (0 2 3 1 4)

Geliştirilen Yeni Yöntem

2. puan dizisinin bir kopyasını (kopya2) oluştur ve sırala.

- (0 3 1 2 7) -> puan dizisi
- (0 1 2 3 7) -> kopya2 dizisi
- puan dizisindeki her bir elemanın kopya2'deki konumuyla geçerli bir tur oluştur.
 - puan dizisindeki 0 elemanın kopya2 dizisinde 0. sırada
 -
 - puan dizisindeki 2 elemanı kopya2 dizisinde 2. sırada
 - O halde geçerli tur -> (0 3 1 2 4)

Geliştirilen Yeni Yöntem

- Geliştirilen yöntemle bir kromozomdan geçerli bir tur üretilirken
 - Hatalı gen üretimi
 - Eksik şehir üretimi
 - Tur olmayan kromozom üretimi

gibi herhangi bir sorunla karşılaşmadık.

Geliştirilen Yeni Yöntemin Avantajları ve Dezavantajları

- Avantajları

- GSP sıra değeri bağımlılığından kurtuluyor.
- Klasik GA işlemleri kullanılıyor.
- Gerek permütasyon kodlamada gerekse de Lidd'in yönteminde karşılaşılan hatalı üretimler (eksik şehir ya da tanımlanmamış şehirlerin üretilmesi gibi) ortadan kalkıyor.

- Dezavantajları

- Şehir sayısına bağlı olarak kromozom boyutu artmaktadır. Karmaşıklık ve çalışma süresi artmaktadır.

Deneyisel Sonular

- TSPLIB' ten gr24, gr48, berlin52 ve kroA100 test verilerini kullandık.
- Her bir kromozomdan yukarıda bahsettiğimiz Yöntem-1'e göre geçerli tur oluşturduk.
- Genetik algoritma parametreleri olarak aprazlama oranı olarak 0.8, mutasyon oranı olarak da 0.03 belirledik.
- Her test verisi için algoritmayı 10 defa alıřtırdık.

Deneysel Sonular

TSPLIB Problemi	Populasyon Boyutu	Nesil Sayısı	Yaklaşık Hata Oranı
gr24	500	300	%8
gr48	2000	400	%11
berlin52	6000	600	%13
kroA100	10000	800	%25

İleri Çalışmalar

- Daha az ikili değer kullanarak bu problem çözülebilir mi?
- Bu yöntemeye özgü yerel en iyileme algoritmaları geliştirilebilir mi?
- Diğer temsil yöntemlerine göre nasıl çalışmaktadır?

Teşekkürler!
Sorular?