Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı

Zeki Optimizasyon Teknikleri Dersi

Sevdanur GENÇ

ZEKİ OPTİMİZASYON TEKNİKLERİ NELERDİR?

Optimizasyon karar değişkenleri, amaç fonksiyonarı ve sınırlayıcılardan oluşmaktadır. İki temel basamak vardır : problem modelinin kurulması ve çözüm üretmek için modelin kullanılması. Amaç, erken yakınsamadan kaçış, bölgesel araştırmada optimuma hızlı yakınsama ve dinamik gerçek dünya optimizasyonudur.

Klasik optimizasyon algoritmaları, problem matematiksel terimlerle çok iyi tanımlanmalıdır. Özellikle gerçek dünya problemlerini çok iyi tanımlamak mümkün değildir. Küresel optimumu garanti edebilir ama aşırı derecede hesaplama zamanına ihtiyaç duyarlar. Kısa sürede bölgesel optimaya yakınsama eğilimindedirler.

Çözüm uzayını etkin bir şekilde aramayı sağlayacak temel sezgisel yöntemleri birleştirmeye çabalayan yeni yaklaşık yöntemlerdir. Bu yöntemlere örnek verecek olursak; ısıl işlem, tabu arama, değişken komşu arama, genetik algoritmalar, karınca kolonisi optimizasyonu, parçacık sürü optimizasyonu ve yapay bağışıklık sistemleridir.

Zeki optimizasyon yöntemlerinin genel özellikleri şöyledir : arama sürecine rehberlik eden stratejilerdir. Amaç, en iyi yada en iyiye yakın çözümleri bulmak için arma uzayını hızlı bir şekilde araştırmaktır. Basit yerel arama algoritmalarından karmaşık öğrenme proseslerine kadar geniş bir yelpazeyi içermektedir. Yaklaşık algoritmalardır ve genellikle deterministik değildir. Arama uzayındaki yeren en iyi tuzaklardan kurtulmak için çeşitli mekanizmaları kullanırlar. Probleme özgü değillerdir. Üst seviye stratejiler tarafından kontrol edilen sezgisellerde probleme özgü bilgi kullanımına izin verirler. İleri seviye algoritmalarda, aramaya rehberlik etmesi amacıyla arama sırasında elde edilen bilgiyi (hafızayı) kullanırlar. Kısacası, farklı metotlar ile arama uzayının araştırılması için yüksek seviye stratejilerdir.

En önemli özellikleri çeşitlendirme ve yoğunlaşma arasındaki dinamik dengeyi oluşturmasıdır. Çeşitlendirme arama uzayında araştırmayı, yoğunlaşma arama sırasında elde edilen tecrübenin (bilginin) işletilmesidir.

Zeki optimizasyon teknikleri esinlendikleri kaynaklara, aramada kullandıkları çözüm sayısına, kullanılan amaç fonksiyonuna, kullanılan komşuluk yapısına ve hafıza kullanımına göre sınıflandırılmaktadırlar.

Esinlendikleri kaynaklara göre : Genel amaçlı sezgisel yöntemler; biyolojik tabanlı, fizik tabanlı, kimya tabanlı, müzik tabanlı ve sosyal tabanlı olmak üzere beş farklı grupta değerlendirilmektedir. Ayrıca bunların birleşimi olan melez yöntemler de vardır.

Aramada kullandıkları çözüm sayısına göre : Algoritma tek bir çözümden başlayıp bunu operatörlerle ilerletiyorsa bunlara tek noktalı yöntemler denir ve tabu arama, ısıl işlem gibi bütün yerel arama tabanlı sezgisel algoritmalar bu gruba girer. Arama sırasında tek bir çözüm kullanan algoritmalar, yörünge (trajectory) metotları olarak adlandırılırlar. Bu yöntemlerde arama, arama surecinde arama uzayında tanımlanan bir yörünge üzerinde gerçekleştirilir.

Kullanılan amaç fonksiyonuna göre : Bazı sezgisel algoritmalar problemin gösterimini gerçekleştirirken amaç fonksiyonunu sabit tutar ve sabit amaç fonksiyonlu olarak adlandırılırlar. Örnek: Genetik Algoritmalar, Karınca Kolonisi Optimizasyonu, Parçacık Suru Optimizasyonu, Tabu Atama, Değişken Komsu Arama, İteratif Yerel Arama ve Tavlama Benzetimi arama. Bazıları da örneğin rehberli yerel arama algoritmasındaki gibi değiştirir ve değişen amaç fonksiyonlu olarak adlandırılırlar. Değiştirmekteki amaç yerel minimumdan kurtulmaktır. Bu mantık, yerel minimumdan kaçmak için bazen diğer sezgisel algoritmalara da uygulanabilmektedir. Örnek: Yönlendirilmiş Yerel Arama (Guided Local Search) meta sezgiseli. Amaç, arama uzayında yapılan bu tür bir modifikasyon ile yerel en iyi çözümlere yakalanmayı önlemektir. Dolayısıyla, arama sırasında elde edilen bilgi kullanılarak amaç fonksiyonu değiştirilir.

Kullanılan komşuluk yapısına göre : Çoğunlukla sezgisel yöntemler tek bir komşuluk yapısında çalışır ve tek komşuluk yapılı olarak sınıflandırılabilir. Değişken komsu arama meta sezgiseli haricindeki meta sezgiseller tek bir komşuluk yapısını kullanırlar. Bazıları da değişken komşuluk arama algoritmasında olduğu gibi arama işlemini sistematik bir şekilde değiştirerek birden fazla yerel arama yöntemiyle diğer çözüm alanlarına ulaşmaya çalışır ve değişken komşuluk yapılı şeklinde sınıflandırılabilir. Parçacık sürü optimizasyonunun iki versiyonu da bulunabilmektedir. Çeşitli komşuluk yapısını kullanmada amaç, aramada çeşitliliği sağlamaktır.

Hafıza kullanımına göre : Algoritmalar çalışırken daha önceki durumlar ya da en iyi durumlar hatırlanıyorsa hafızalı, hatırlanmıyorsa hafızasız şeklinde sınıflandırılabilir. Örneğin parçacık sürü optimizasyonu, karınca kolonisi ve tabu arama hafızalı, ısıl işlem hafızasızdır. Sınıflandırılmadaki özellik: arama sırasında aramadan elde edilen geçmiş bilgiyi kullanıp kullanmadıklarıdır. Hafıza kullanımı iki farklı şekilde gerçekleştirilebilir: Kısa donem hafızada, yeni yapılan hareketler, çözümler ya da alınan kararlar tutulur. Uzun donem hafıza ise, arama sırasında elde edilen bilgiler tutulur.

Çok noktalı algoritmalar : Evrimsel (Gelişimsel) Hesaplama, Genetik Algoritmalar, Evrimsel Algoritmalar, Diferansiyel Gelişim Algoritması, Evrimsel Programlama, Evrim Stratejileri, Sınıflayıcı Sistemleri, Genetik Programlama, Karınca Koloni Algoritmaları, Arı Koloni Algoritmaları, Yapay Bağışıklık Sistemleri, Parçacık Sürü Optimizasyonu.

Bu çalışmada, zeki optimizasyon tekniklerinden karınca kolonisi algoritması Matlab programlama dili ile kullanılarak görsellerde kenar tespiti yapılmıştır. Çalışmanın sonunda orijinal görüntünün diğer kenar tespit algoritmaları olan Robert Cross Filtresi, Prewitt Filtresi, Sobel Filtresi, LoG Filtresi ve Canny Filtreleri ile karşılaştırılması yapılmıştır.

GİRİS

Karincalar, yasadiklari populasyonlari icerisinde tek baslarina kucuk bir canli olsa dahi topluluk olarak incelenecek olursa buyuk bir karmasanin bireyleri olabilyorlar. Yasamlarini surdurebilmeleri icin kendi boyut ve agirliklarindan daha fazla yukleri tasiyabiliyorlar. Kendi iclerinde sezgisel bir sekilde iletisim agi kurup, kopru misali yollar tasarlayip yuklerini bu yollar uzerinden tasiyarak yuvalarina goturebiliyorlar.

Karinca Algoritmasi, gercek karinca populasyonundan esinlenerek tasarlanmis olup bilim dunyasinda bu algoritma ile bir cok cozumde kullanilmistir.

Karinca kolonisi algoritmasi ve diger tum sezgisel algoritmalarin kullanim alanlari en kisa mesafe problemleriyle sinirli degildir. En kisa zaman, en az kaynak, en hizli sonuclar gibi bir cok cozum yolu icin kullanilabilmektedir. Ornegin, bir arama motorunda karinca kolonisi optimizasyonu kullanilarak amac edilen sonuca en kisa zamanda ve en hizli sekilde ulasilabilir.

Bu calismada, oncelikle karinca algoritmasindan bahsedilecektir. Bununla birlikte karincalarin sezgisel davranislarini matematigin olasilik formulleri ile incelenip, ilgili terimlerden bahsedilecektir. Sonrasinda Karinca algoritmasi ile goruntuler uzerinde kenar tespit islemleri yapilacaktir.

Anahtar Kelimeler; Karinca kolonisi algoritmasi, Feromon guncellemesi, Kenar tespit, Karinca algoritmasi, Karinca kolonisi optimizasyonu.

KARİNCA KOLONİ ALGORİTMASİ NEDİR?

Karinca kolonisi algoritmasi (Ant Colony Algorithm - ACA), 1991 ylinda Marco Dorigo tarafindan tasarlanmis bir algoritmadir.

Algoritmanin genel bakis acisi; gercek karinca kolonilerinde yasamlarini surduren karincilarin kendi sistemlerini surdurebilmeleri icin yiyeceklerini arastirip bulmalari gerekmektedir. Bunun icin yiyecekleri ile yuvalari arasindaki mesafeyi her zaman icin en kisa surede katetmeleri gerekmektedir. Bunun icinde yiyecekleri ve yuvalari arasindaki en kisa yolu secmek zorundadirlar ve zamanla bu sezgisel davranislarini bir kabiliyet haline getirmislerdir. Bu kabiliyet sayesinde zaman icerisinde surekli kullandiklari en kisa yolun cevresinde fiziksel, kimyasal veya cevresel herhangi bir degisim oldugunda, en kisa yol artik onlar icin iyi bir tercih olmayabilir. Bunun icinde yeni en kisa yollar bulmaya baslarlar.

Tum bu sezgisel davranislarinin yani sira, karincalarin onemli olan diger bir ozellikleri ise cok iyi bir gorme kabiliyetine sahip olmamalaridir. Bu en kisa yolu secmek icin tabikide etrafi tam olarak gorebilecekleri anlamina gelmiyor. Tam olarak kor de degillerdir. Bu sebepten, karincalar birbirleri ile haberlesebilmeleri icin kendileri tarafindan urettikleri bir kimyasal maddeyi kullanmaktadirlar.

Feromon:

Karincalar kendi aralarindaki iletisimi saglayabilmeleri icin Feromon isimli bir kimyasal madde kullanmaktadirlar. Karincalar yiyecek bulduklari hedeflerine yada yuvalarina ulasabilecekleri yollari katederken, yollari uzerine kendilerinin salgilamis oldugu bir miktar Feromon kimyasal sivisini ve ya kokusunu birakirlar. Karincilar yonlerini tespit ederken feromon sivisinin miktarina onem vermektedirler. Eger ki ulasilacak hedefe ait tum yonlerin feromon sivi miktarlari birbirine esit ise, tum karincalarin bu yonleri secebilme ihtimalleri yani olasiliklari ayni olacaktir. Karincalarinin hepsinin ayni hizlara veya birakmis olduklari sivi miktarlarinin ayni olmasi, daha kisa yollar birim zamanda daha cok feromon maddesi alacagini belirler. Bu sebepten, karincilar hizli bir sekilde en kisa yolu bulurlar. Karincalarin tum bu ugraslari tamamen dogal bir optimizasyon islemine ornek olabilecek seviyede bir davranistir. Tum bu optimizasyon islemleri icinde bir karinca algoritmasi olusturulmustur.

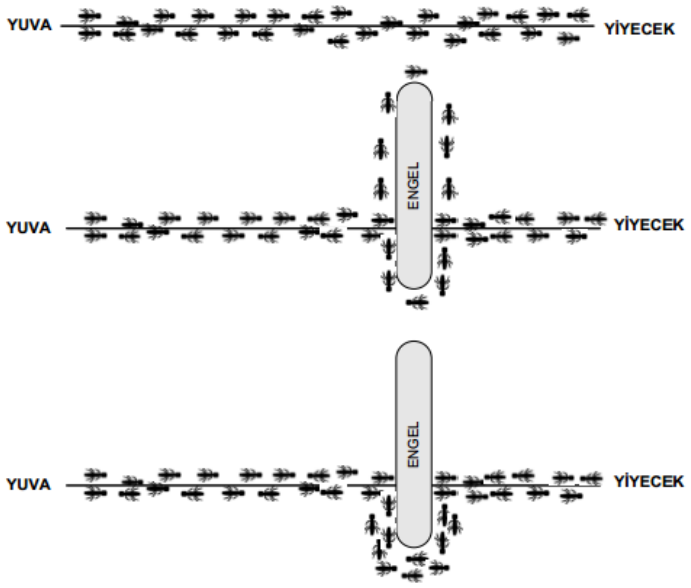
Karinca algoritmalari genetik algoritma gibi bir populasyon sistemini yaklasim olarak benimsemistir. Karinca populasyonu icerisinde bulunan tum karincalar bir cozum yolunu temsil etmektedir. Tum bu cozum yollari sayesinde populasyon icerisindeki tum karincilarin hareketlerini belirlemelerinde referans olabilmektedirler.

Algoritma, karinca kolonilerinden esinlenerek tasarlandigindan olusan sisteme; karinca sistemi (KS), algoritmaya ise karinca kolonileri algoritmasi (KKA) ismi verilmektedir.

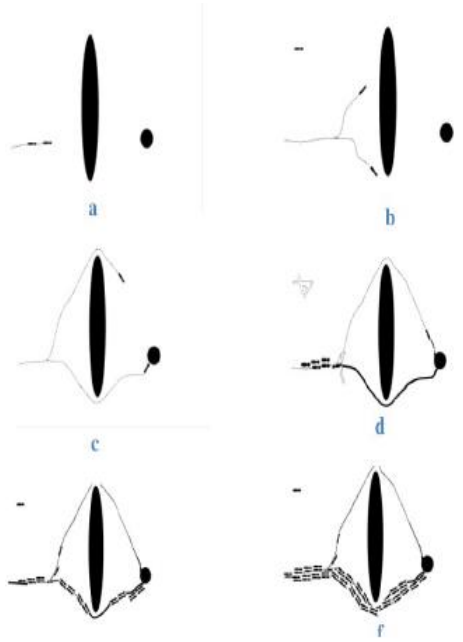
```
BEGIN
  REPEAT
       Bütün yapay karıncalar için yolların üretilmesi
       Bütün yapay yolların uzunluğunun hesaplanması
       Yapay yollar üzerinde bulunan feromon maddesi miktarının güncellenmesi
       Şu ana kadar bulunan en kısa yapay yolun hafızada tutulması
    UNTIL ( iterasyon = maksimum iterasyon yada yeterlilik kriteri )
END.
```

Karinca Koloni Algoritmasinin Adimlari:

1. Adim: Karincalarin yonlerini bulabilmesi icin kullandiklari feromon sivilarina ait baslangic feromon sivi degerleri belirlenir.
2. Adim: Karincalar her farkli noktaya rastgele yerlestirilir.
3. Adim: Karincalar, sonraki hedeflerine olasilik denklemlerine bagli olacak sekilde turlarini tamamlarlar.
4. Adim: Karincalarin katettikleri yollar ve buna ait olan feromon sivi miktarlari hesaplarinir, sonrasinda yeni lokal feromon sivi miktarlari olusturularak ilgili bilgi guncellenir.
5. Adim: En iyi cozume ait yol hesaplanir ve global feromon guncellenmesinde kullanilir.
6. Adim: Iterasyon sayilari tamamlandiktan sonra 2. Adim'a gidilir.



Karincalarin Sezgisel ve Feromon Davranislari:



Karincalar yasamlarini surdurebilmeleri icin yiyeceklerini cevreden arastirmalari gerekir. Bunun icin oncelikle, karinca kolonisinden oncu karincalar arastirma yapmak icin yuvalarindan cikarlar.

Oncu karincalar cevrede sezgisel olarak bulmus olduklari herhangi bir yiyecek kaynaginin konumunu tespit edip hafizalarinda bunu hedef olarak belirlerler. Sonrasinda oncu karincalar hafizalarindaki bu bilgi ile yuvalarina geri donerler.

Geri donus esnasinda vucutlarindan salgiladiklari kimyasal sivi olan feromon yardimiyla, gectikleri yollara koku veya sivi olacak sekilde izler birakirlar. Oncu karincalar yuvalarina vardiklarinda karinca kolonisindeki diger karincalara hedef hakkinda hafizasindaki bilgiyi aktarir ve populasyondaki belirli sayidaki karincalar bilgiyi alarak hedefe varmak icin yola cikarlar. Karincalar hedefe ulasana kadar oncu karincalarin yollara onceden birakmis olduklari izleri referans alarak rastal bir sekilde ilerlerler. Boylelikle iz bulunan tum yollarda farkli sayilarda karincalar bulunmaktadir. Bu yollardan kisa yollari tercih eden karincalar hedefle yuva arasinda daha sik bir sekilde gidip gelecekleri icin artik o yoldaki feromon kokusu yada sivisinin miktari daha fazla olacaktir. Az kullanilmaya baslanan yollardaki feromon sivi yada kokusu zamanla buharlasmaya baslayacaktir.

Daha onceden kisa yolu secmeyen ve diger yollara rastsal bir sekilde dagilan karincalar yuvalarindan ayrilirken yol uzerindeki daha yogun olan feromon kokusuna yada sivisina yonelme olasiliklari cok daha fazla artacaktir. Zaman icerisinde bu kisa yolu tercih eden diger tum karincalarin sayilari artacaktir. Boylelikle karincalar yuvalari ve hedefleri arasinda duzenli bir kopru kurmus olacaklardir.

Karincalarin Feromon ve Sezgisel Davranislarinin Algoritmadaki Yeri:

Karincalarin bir problem uzerindeki cozumlerinin yaklasiminda kullanacaklari yontemleri oncesinde sezgisel sonrasinda ise feromon kimyasal maddesi olarak siniflandirabiliriz. Bunun icinde algoritma icerisinde adim adim ilerleyebilmemiz icin belirlenen karincalarin sayilari ile birlikte matrisler olusturuyoruz. Bu matrisleri sezgisel yaklasimlari ifade edebilmesi sezgisel matris, feromon kimyasalini ifade edebilmesi icinde feromon matrisi olarak isimlendiriyoruz. Bu matrisler olasilik formullerinde kullanilarak iyi sonuclara ulasmamizi saglayacaktir.

Feromon matrislerinin olasilik formullerindeki onemi; karincalarin hedefle yuva arasindaki secmis olduklari yollarda bulunan feromon sivi miktarlarinin duzeyi yer almaktadir.

Sezgisel matrislerinin olasilik formullerindeki onemi; karincalarin vardiklari bir dugumden varacaklari baska bir dugume kadar olan mesafenin uzakligiyla ters orantili bir baslangic degeridir.

Algoritma cozumunde kullanilacak karinca sayisinin artirilmasi cozum icin iyi sonuclar verecegi gibi bu matrislere degerlerin atanmasi ve olasiligin cok fazla artmasi ile cozume ulasmak icin tahmin edilen surenin cok fazla artmasina sebebiyet vereceginden zaman gittikce artacaktir. Karincalarin sayisi, cozulecek problemlerin buyuklugune ve algoritmanin kullanilacagi alana bakilarak veriler degistirilebilir.

Karinca Tur Kurallari:

Karinca Koloni Algoritmasinda bir tur esnasinda, i noktasinda yer alan k karincasi, j noktasina yonelirken iki alternatif yol secmek zorundadir. Bu secimde Feromon degerinin en yuksek olmasina dikkat edecektir. Genellikle bu secim ilk alternatifte olasilik degeri q0 = %90 seviyesinde belirlenmektedir. Ikinci alternatifte ise, olasilik dagilimina bagli olarak yollar secilecektir. Bu tur esnasinda i noktasindaki k karincasinin u adet alternatif yoluna ait cozum yapilacak formul soyledir;

$$j = \max_{u \in J_k(i)} \left\{ \left[\tau(i,u) \right]^{\alpha} \times \left[\eta(i,u) \right]^{\beta} \right\} \quad \text{eğer} \quad q \le q_0$$

Burada ilk parantez icerisindeki degerler feromon izleridir. Ikinci parantez arasindaki degerler ise i noktasindan u noktasina ait uzakligin tersi bulunmaktadir. Jk(i) ise, i noktasinda bulunan k karincasinin henuz gitmedigi noktalari temsil eder. Beta (B>0) feromon guncellemesinde, uzakligin goreli onemliligini belirleyen bir degerdir. q0(0<q0<1) cozum uzayini belirleyen bir degerdir.

q<=q0 durumu gerceklestiginde, gidilmesi gereken diger hedef secim degerlerine bagli olarak rastsal bir sekilde secilmektedir. Boylelikle, feromon miktari yogun olan yollarin secilme olasiligi daha fazla olacaktir.

Gidilecek olan yollarin secilme olasigina ait formul ise soyledir;

$$p_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha} \times [\eta(i,j)]^{\beta}}{\sum_{u \in j_k(i)} [\tau(i,u)]^{\alpha} \times [\eta(i,u)]^{\beta}} & \text{eğer } j \in J_k(i) \\ 0 & \text{Diğer durumlarda} \end{cases}$$

Pk(i,j) : k karincasinin i noktasindan j noktasina gecme olasiligi
r(i,j) : i ve j noktalari arasindaki feromon matris degeri
n(i,j) : i ve j noktalari arasindaki sezgisel matris degeri
a : feromon katsayisi
B : sezgisel katsayisi
Jk : yollara ait noktalarin tamamdir.

Feromon Guncellemesi :

Populasyondaki karincalarin tamami yuva - hedef arasindaki turlarini tamamladiktan sonra feromon sivi miktarlari zaman icerisinde yenilenmekte yani guncellenmektedir. Ilk islem, noktalarin etrafindaki tum yollarda feromonlar bellirli oranlarda sirasiyla buharlasmaktadir. Buharlastirma islemi az tercih edilen tum yollardaki feromonlara uygulanmaktadir. Buharlastirma icin 0 ve 1 arasinda sabit bir degere sahiptir. Karincalarin turlamis olduklari yollardaki feromon miktarlari, o yolu onceden kullanan karincanin toplam yol uzunluguyla ters orantili olarak artis miktarinda degisim gosterecektir. Boylelikle, kisa yollardan gecmis olan karincalarin feromon miktarlari daha fazla artisa sebep olacaktir.

Feromon guncellemeleri iki sekilde yapilmaktadir. Bunlar; Local ve Global feromon guncellemeleridir.

Local Feromon Guncellemesi;

Feromon degerleri bir matris olarak dusunelecek olursa; matris degeri t iterasyonuna kadar ilerleyen feromon degerlerine ait bir vektoru olusacaktir. t iterasyonundaki feromon duzeyi ve 0<p<1 araligindaki buharlastirma sabiti uzerine ilgili local feromon guncelleme formulu soyle olacaktir;

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t+1)$$

$$\Delta\tau_{ij}^{k}(t+1) = \begin{cases} 1/L^{k}(t+1) & \text{k karıncası (i, j)} \\ & \text{yolunu kullanmışsa,} \\ 0 & \text{diğer durumlarda} \end{cases}$$

Lk(t+1), k karincasinin t+1 iterasyonundaki toplam tur miktaridir. Karincalar degisen feromon miktarlari ile birlikte her iterasyonda turlarinida degistirmektedirler. Amac kisa yola ait turlari tespit edebilmekdir.

Global Feromon Guncellemesi;

Karinca koloni algoritmasinda gecerli yollara ait en iyi sonuca sahip olan k karincasinin izledigi yolun feromon duzeyinin arttirilmasi saglanir. Bununla birlikte iterasyonlarda bulunan en iyi sonuclarin belli bir oranda ileriki iterasyonlara aktarilmasi gerceklestirilir. Lbest(t+1), gecerli iterasyonda bulunan en iyi yola ait olan turun uzunluk miktadir.

Local feromon guncellemesine benzeyen bu olayin formulu soyle olacaktir;

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}^{k}(t+1)$$

$$\Delta\tau_{ij}(t+1) = \begin{cases} \dfrac{1}{L_{best}(t+1)} & (i, j) \text{ en iyi tura ait ise} \\ 0 & \text{diğer durumlarda} \end{cases}$$

KENAR TESPİTİ NEDİR?

Kenar belirleme (Edge Detection) olarak bilinen en onemli goruntu isleme teknigine ait bir cok algoritma bulunmaktadir. Bu algoritmalarin araciligiyla var olan goruntumuze ait kenarlari tespit edebiliriz.

Kenar olarak bildigimiz tum tanimlar image'lar yani goruntuler uzerinde farkli bir boyut almaktadir. Image'larda kenarin tek bir anlami vardir o da, goruntuler arasindaki renk degisimleridir. Fakat her renk degisimide kenar olarak tespit edilemez.

Kenar tespit algoritmalarinda calisirken oncelikle uzerinde calistigimiz goruntunun siyah - beyaza donusturulmesi gerekiyor. Kenar tespit edilmeden once belli bir thershold yani esik deger belirlenilmelidir. Esik degerinden sonra goruntu uzerindeki satir ve sutunlarda bulunan tum pixel'ler arasindaki renk tonlarinin degisimi bu esik degerini referans alarak kenarlar tespit edilebilir. Eger goruntu uzerindeki renk degisimi esik degerin altinda bir sonuc ise, kenar olarak isimlendirilemez.

Kenar tespit yontemlerinde en cok kullanilan algoritmalara ait filtreler ise soyledir;

Sobel Filtresi; Sobel filtresine ait algoritmada Iki tane konvulasyon kerneli bulunmaktadir. Birisi yatay kenarlari tespit etmek icin kullanilirken digeri ise dikey kenarlari tespit etmek icin kullanir. Eksenler uzerindeki piksellere daha cok agirlik verir.

Prewitt Filtresi; Goruntuler uzerinde yatay ve dikey yonlere ait olan kernellerle birlikte egimlere odaklanarak sonuclar vermektedir.

Canny Filtresi; Kenar tespit yontemlerinde en basarili sonucu veren bir kernel yapisina sahiptir. Goruntu turevi alinmadan once yumusatma filtresi uygulanmaktadir. Tek piksel kalinliginda kenarlar uretir ve kirik cizgilerle birlikte pikselleri birlestirir.

Kenar bulma islemlerinde genel amac; goruntudeki gurultelere karsi dusuk duyarliligi bulabilmektedir. Sinirlarin iyi belirlenebilmesi ve geri kalan tum kenarlardaki karisikliklari elemektir. Bu sebepten, kerneller sayesinde goruntu icerisindeki isik yogunlukluklarina ait degisikliklerin ani oldugu yerleri yakalayabiliriz.

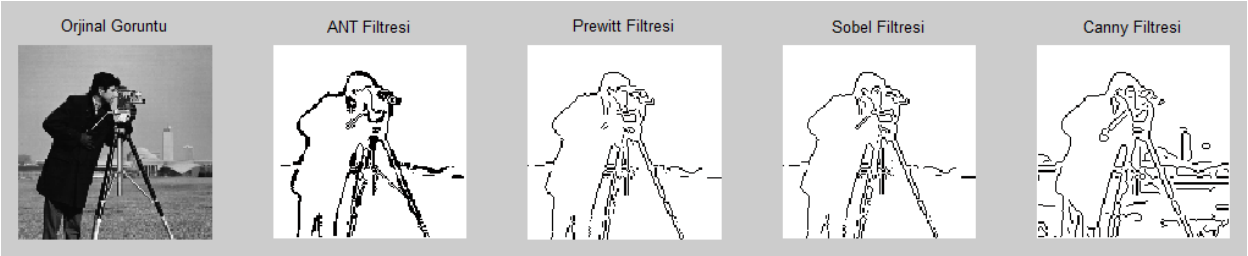
KARİNCA KOLONİ ALGORİTMASİ YARDİMİYLA GORUNTU ISLEMEDE KENAR TESPİT YAKLASİMİ

Kenarlari tespit edilecek goruntulerdeki her bir kenar aslinda karincalar icin bir beslenme yeri, hedef nokta olarak belirlenmektedir.

Goruntu uzerinde, goruntunun boyutlarina gore karincalar rastgele olacak sekilde pikseller uzerinde farkli konumlara dagitiliyor. Bilimsel makalelerde calisilan goruntu boyutlari 128X128 ve 8 bit'lik ozelliklerdedir. Yine bu makalelerde kenar tespit yonteminde oldukca iyi

sonuclar alinabildigi gibi goruntunun piksellerine gore olumsuz sonuclarda alindigi gorulmektedir.

Kenar tespit yonteminde Canny algoritmasinin basarisiyla kiyaslandiginda ise malesef Prewitt ve Sobel kenar tespit yontemleri ile hemen hemen ayni seviye bir basari sergiledigi gorulmektedir.



Algoritmanin pixel sayisi arttikca calismasi oldukca uzun zaman alabiliyor, bunun icin uygulamanin hiz bakimindan performansini arttirabilmek adina matlab'in **Mex File Function** 'larini kullanmak cok fazla faydali olunabilecegini dusunuyor ve ileride bununla ilgili uygulamayi daha iyi bir sekilde iyilestirmeyi planliyorum.

KARINCA KOLONİ ALGORİTMASİ KULLANİLARAK GORUNTULERDE KENAR TESPİT YONTEMİ KULLANİMİ

Bu calismamizda, goruntu uzerindeki kenarlarin tespit edilmesinde cok saglikli sonuclar alinabildigi gibi nadir de olsa basarisiz sonuclarda alabiliyoruz.

Calismada zamandan tasarruf edebilmek adina, kullanilan goruntuler 128X128 boyutlarinda ve 8 bitlik olacak sekilde hazirlanmistir. Bu istege bagli olarak baska calismalarda degistirilebilir.

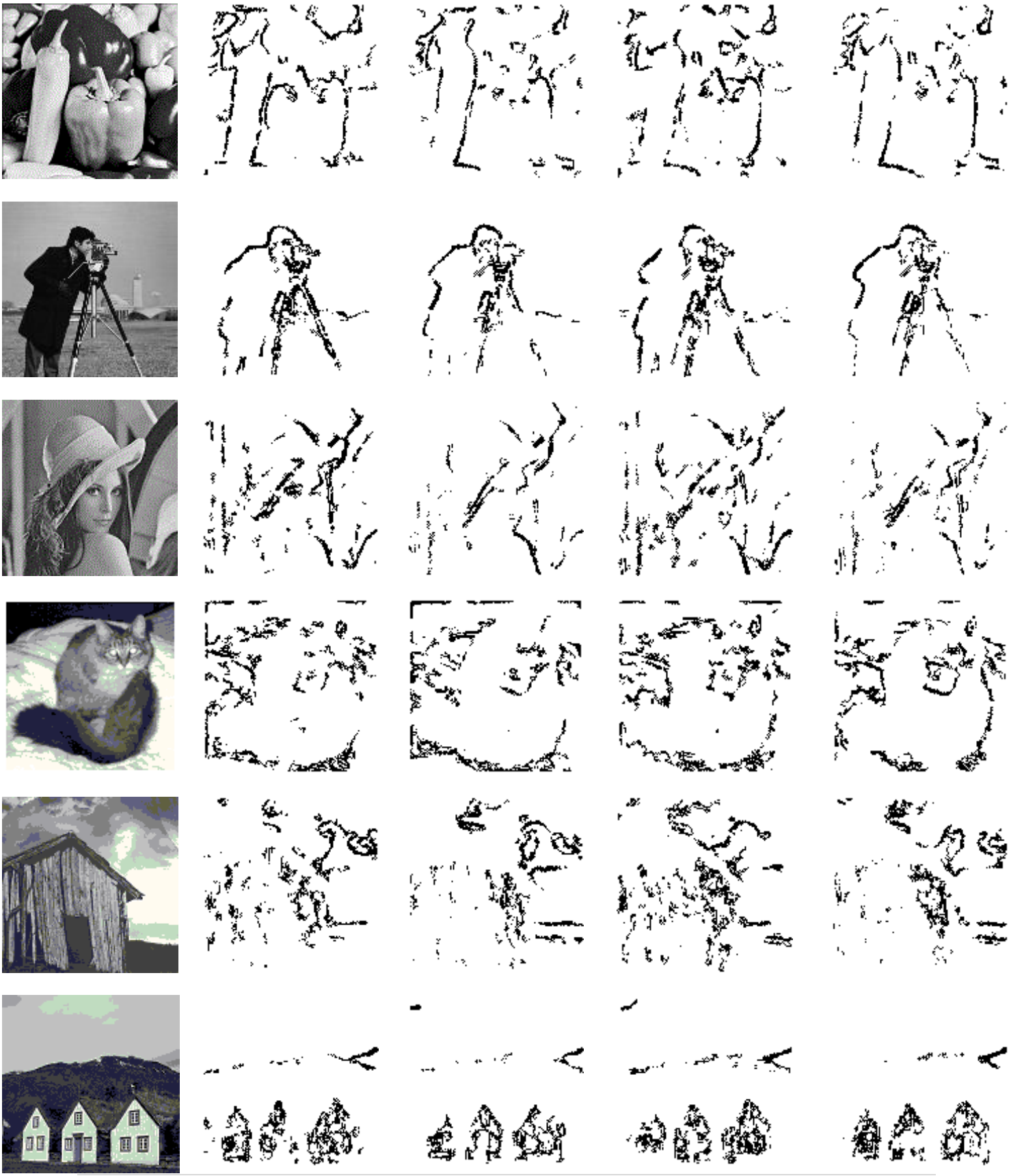
Calismada goruntuler uzerinde islemleri en hizli sekilde gerceklestirebilmek adina Matlab programinin R2017a versiyonu kullanilmistir.

Asagidaki goruntulerde iterasyon sayisi 3 olarak kullanilarak kenar tespit algoritmasi ile almis oldugumuz goruntuler bulunmaktadir.

Kullanilan toplam karinca sayilarini goruntunun satir ve sutun pixel degerlerine gore belirliyoruz. Burada 128 X 128 piksel degerlerine uygun yaklasik olarak 300 degeri verilmektedir.

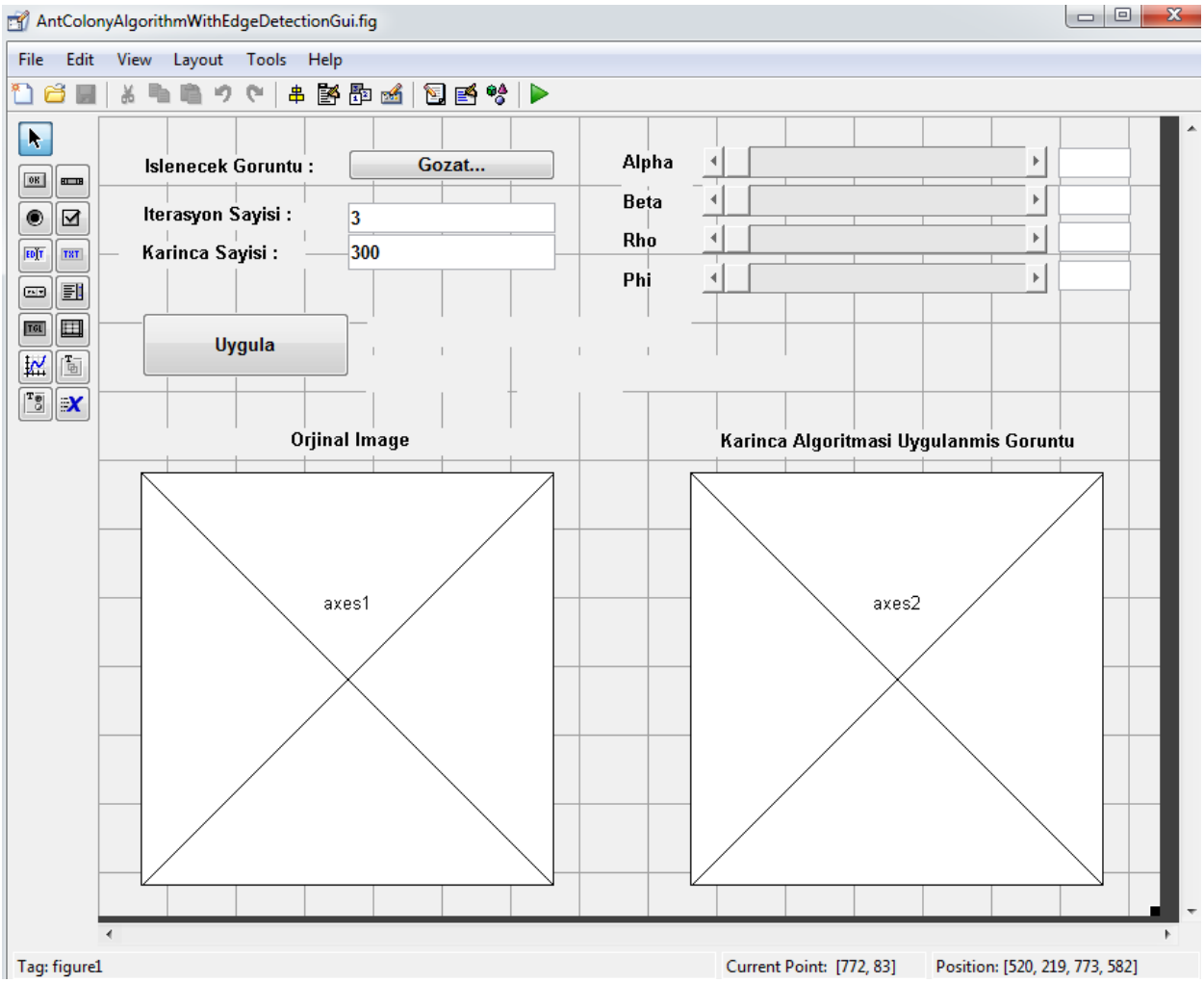
Varsayilan olarak alpha ve beta degerlerini 1 verilir.

Phi degerleri 0 - 0.1 deger araliklarindayken, Rho degerleri 0 - 1 deger araliklarinda gore alinan sonuclara ait ornek olabilecek ekran goruntuleri soyledir;



EDGE DETECTION BY ANT COLONY ALGORITHM UYGULAMASININ EKRAN GORUNTULERI;

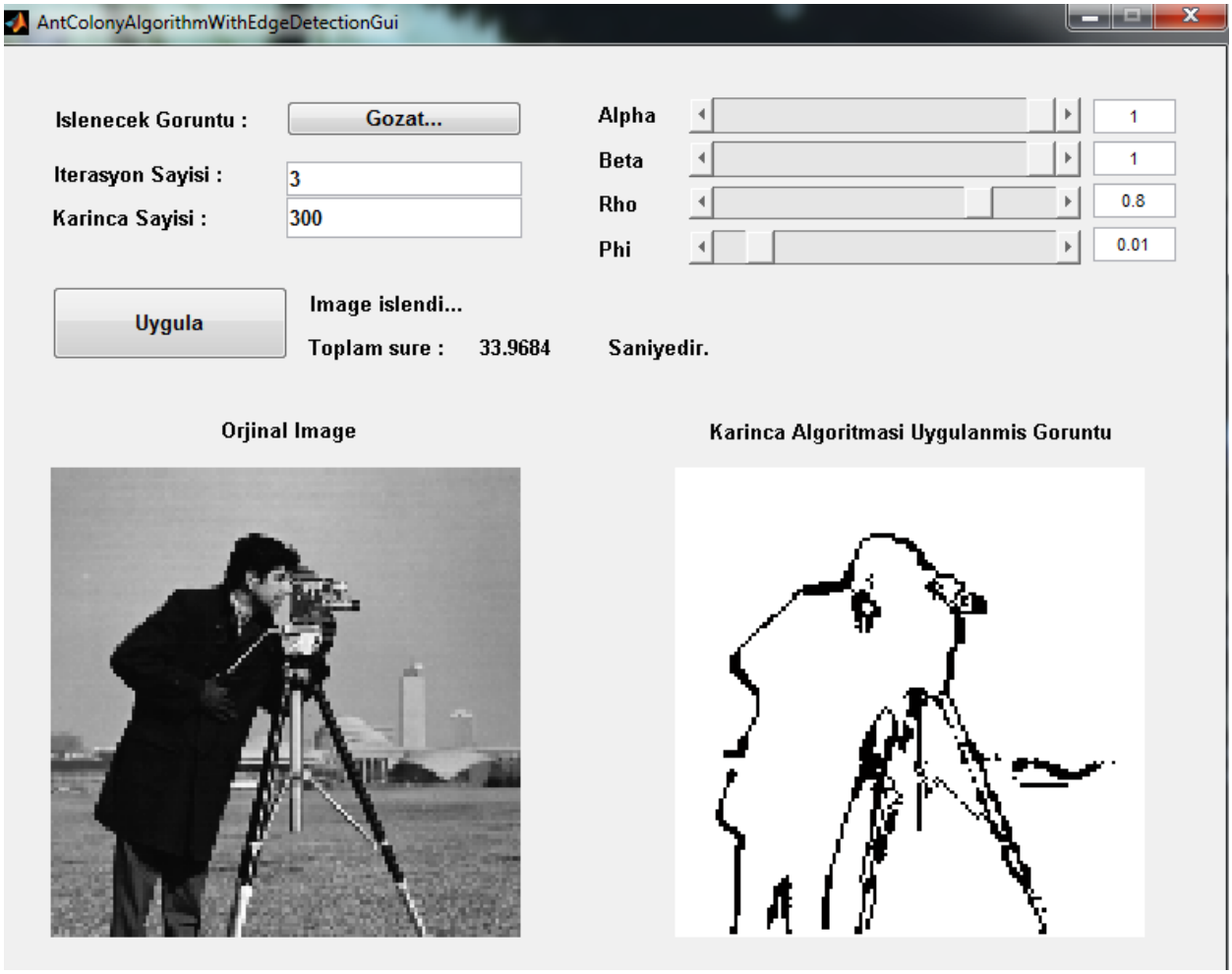
MATLAB GUIDE TASARIMI



Bu calismada, Matlab'in gui tool'u kullanilarak kullanici arayuzu ile algoritmanin onemli verilerini kullanicidan isteniyor.

Formun uzerinde push button yardimiyla kullanicidan algoritmasi uygulanacak image isteniyor. Secilen image, Axes1 icerisinde gozukmesi icin ilgili kodlar arka planda yaziliyor. Gui tool'una ait slider'lar yardimiyla kullanicidan olasilik formullerinde kullanilacak olan alpha, beta, rho ve phi sayilari isteniyor. Edit text'ler yardimiyla algoritmanin iterasyon sayisini ve yine image'in boyutlarina uygun olacak sekilde algoritmada kullanilacak karinca sayisinin girisleri yapiliyor. Uygulamanin calistirilabilmesi icin, Push button yardimiyla secilen image uzerinde karinca algoritmasi uygulaniyor. Ilgili image'in algoritma uygulanmis hali hemen Axes1'in yaninda bulunan Axes2 icerisinde kullaniciya sonucu sunuluyor. Algoritmanin calisma suresi ise static text'lerle kullaniciya raporlamis oluyoruz.

UYGULAMANİN CALİSTİRİLMİS HALİ

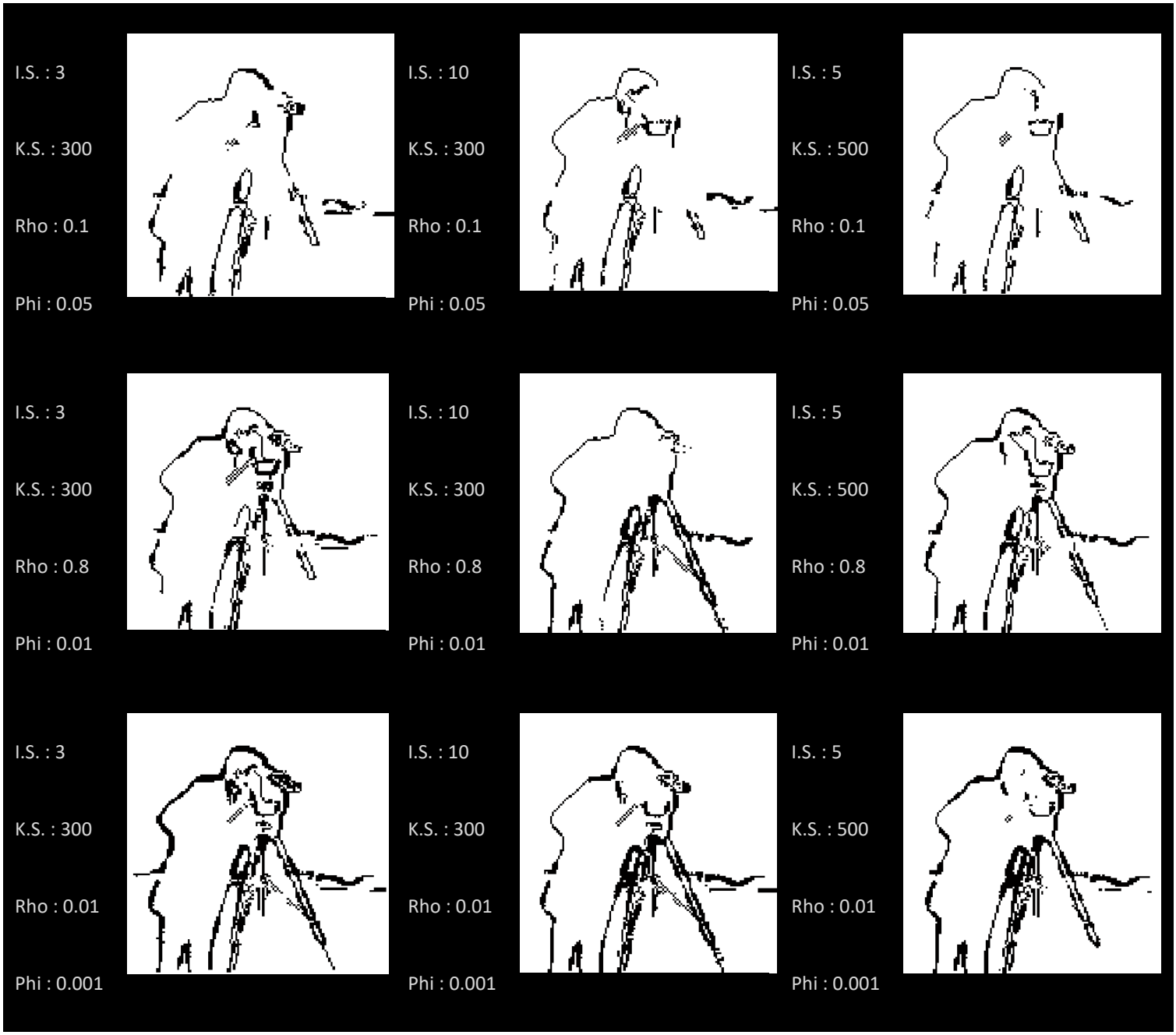


TESTLERDE KULLANİLAN DEGERLER

İterasyon sayisi	**3**
Karinca sayisi (image satir piksel X image sutun piksel)	**300**
Alpha degeri (Feromon matris varsayilan katsayisi)	**1**
Beta degeri (Sezgisel matris varsayilan katsayisi)	**1**
Rho degeri (Feromon buharlasma katsayisi)	**0.1**
Phi degeri (Feromon yok olma katsayisi)	**0.05**
Feromon degeri	**0.0001**
Karinca adim sayisi (128X128 / 256X256 / 512X512)	**40 / 30 / 20**

Bu calismada, kullanilan goruntuler 128X128 boyutlarinda olup; karinca sayisi, iterasyon sayisi, feromon buharlasma kat sayisi rho degeri ile feromon yok olma kat sayisi phi degerlerinin degisimleri ile goruntu uzerinde farkli kenar tespit sonuclari tespit edebiliyoruz.

TEST SONUCLARİ



* I.S. : Iterasyon Sayisi
* K.S. : Karinca Sayisi
* Rho : Feromon buharlasma katsayisi olan Rho deger araligi
* Phi : Feromon yok olma katsayisi olan Phi deger araligi

DİGER KENAR TESPİT ALGORİTMALARİNDAKİ GORUNTU SONUCLARİ İLE KİYASLAMA

```
goruntu = imread('KameraliAdam.bmp');
robertFiltresi = edge(goruntu,'roberts');
prewittFiltresi = edge(goruntu,'prewitt');
sobelFiltresi = edge(goruntu,'sobel');
LoGFiltresi = edge(goruntu,'log');
cannygoruntu = edge(goruntu,'canny');
figure
subplot(231), imshow(goruntu), title('Orjinal Goruntu')
subplot(232), imshow(robertFiltresi) , title('Robert Cross Filtresi')
subplot(233), imshow(prewittFiltresi), title('Prewitt Filtresi')
subplot(234), imshow(sobelFiltresi), title('Sobel Filtresi')
subplot(235), imshow(LoGFiltresi), title('LoG Filtresi')
subplot(236), imshow(cannygoruntu), title('Canny Filtresi')
```