SAKARYA ÜNİVERSİTESİ

Veri Madenciliği Uygulamaları

Hafta 12

Yrd. Doç.Dr. Nilüfer YURTAY

Kümeleme-Yoğunluğa Dayalı Yöntemler

12.1 Bulanık Kümeleme

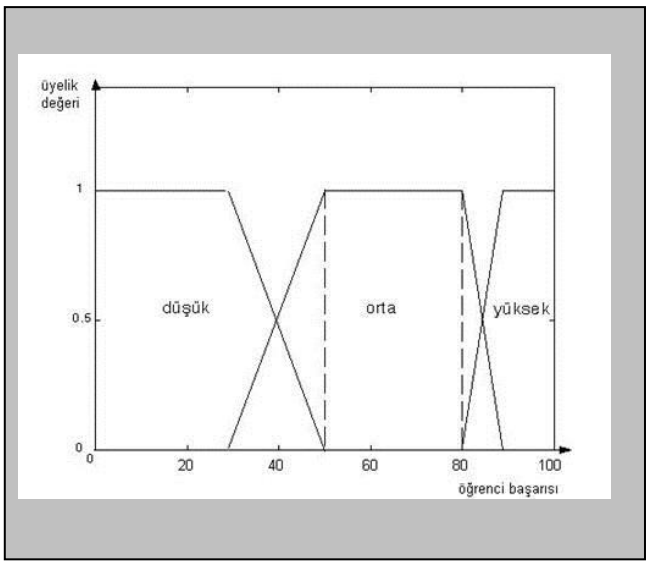
Doğadaki pek çok olgu, insan beyni yardımıyla niteliksel bir şekilde modellenebilir. Bu yaklaşım, günlük yaşamdaki dilsel (linguistik) ifadelerin sayısal modellenmesine olanak tanırken, olasılıkla (probability) ifade edilemeyen durumların olabilirlik (possibility) yardımıyla analizini de sağlar. Örneğin bir paranın atılması olayında limit sonsuza giderken yazı gelme olasılığı %50 olarak belirtilebilir. Ancak "yarın havanın yağışlı olup olmayacağı" olasılık yaklaşımıyla ortaya konulamaz. Bu durumda olabilirlik yaklaşımına ihtiyaç söz konusudur. Problem bu kez uzmanların tecrübelerinden ve yaklaşık ifade etme (approximation) tekniklerinden yararlanılarak çözüme ulaştırılmaya çalışılır[1].

Günlük yaşamda ve bilimsel çalışmalarda yaygın olarak kullanılan; "İyi insan, uzun boy, başarılı öğrenci ve ılık su" gibi tanımlardaki 'iyi', 'uzun', 'başarılı', 'ılık' ifadeleri birer dilsel tanımlama olup bulanık kavramlar olarak değerlendirilir. Belirtilen terimlerin ortak özelliği görecelik içermeleridir. Bu nedenle göreceli bir düzlem üzerinde ele alınarak sayısallaştırılmaları gerekir.

Belirli ve tanımlanmış elemanlardan oluşan topluluklar küme olarak ifade edilmektedir. Klasik küme kuramında temel mantık, ait olmadır. Bir eleman o kümenin ya elemanıdır veya değildir. Kümeye ait olduğunda 1, olmadığında 0 değerini alır. Üyelik kesin (crisp) sınırlarla ayrılmıştır ve kısmi üyelikten söz edilemez. Klasik kümelerde esneklik yoktur.

Bulanık kümeler kuramının temel yapısında; belirsizlik ifade eden tanımlanması güç veya anlamı zor kavramlara üyelik derecesi atayarak onlara belirlilik getirmek vardır. Bulanık küme, değişik üyelik derecesinde öğeleri olan bir topluluktur. Klasik küme kuramındaki kesin ayrım bulanık kümelerde yer almaz. Bulanık kümelerde eleman, bir bölümüyle (örneğin: 0.3) kümeye ait iken bir bölümüyle (örneğin: 0.7) de kümenin dışındadır. Bulanık kümelerde, klasik kümelerdeki üyeliği tanımlayan karakteristik fonksiyon; , yerini üyelik fonksiyonuna; bırakır Şekil 12.1'de yamuk biçimindeki üyelik fonksiyonları kullanılarak bir uygulama gerçekleştirilmiş ve öğrenci başarısı için örnek bulanık küme gösterimi verilmiştir. Orta-düşük ve orta-yüksek geçişlerinde paylaşım bölgesi söz konusu olup katı bir ayrım geçerli değildir.

[1] http://www.egitisim.gen.tr/site/arsiv/52-18/306-bulanik-mantik-ve-egitim.html



Şekil 12.1 Öğrenci başarısı için bulanık küme gösterimi

Kümeleme Analizi X veri matrisinde yer alan ve doğal gruplamaları kesin olarak bilinmeyen birimleri, değişkenleri ya da birim ve değişkenleri birbirleri ile benzer olan alt kümelere ayırmaya yardımcı olan yöntemler topluluğudur[2].

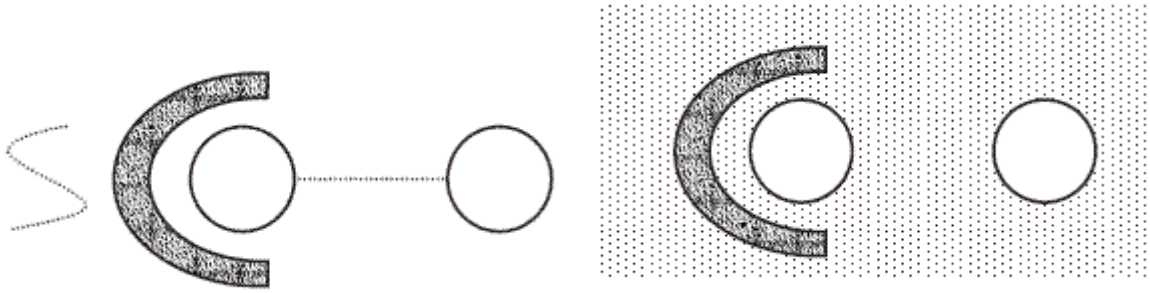
Bulanık kümeleme yöntemi, kümeler birbirinden belirgin bir şekilde ayrılamıyorsa ya da küme üyeliklerinde bazı birimler küme üyeliğinde kararsızsa uygun bir yöntem olarak ortaya çıkmaktadır. Bulanık kümeler kümedeki birimin üyeliği olarak tanımlanan 0 ile 1 arasındaki her bir birimi belirleyen fonksiyonlardır. Birbirine çok benzeyen birimler aynı kümede yüksek üyelik ilişkisine göre yer alırlar. Bundan dolayı Bulanık Kümeleme Yöntemi, birimlerin kümeye ya da kümelere ait olabilme katsayılarını hesaplar. Üyelik katsayılarının toplamı daima 1'e eşittir. Böylelikle birim en yüksek üyelik katsayısına sahip olduğu kümeye atanır. Üyelik fonksiyonları, kümedeki elemanlar sürekli veya süreksiz olsun bir bulanık kümedeki bulanıklığı karakterize eden fonksiyonlardır. Klasik kümeleme yöntemlerinde ise her bir birim sıfır olmayan sadece bir üyelik katsayısına sahiptir ve bu değer daima 1 dir. Dolayısıyla klasik kesin kümeleme yöntemleri, bulanık çözümlemenin sınırlı bir durumudur. Bulanık kümelemenin iki temel yöntemi vardır. Bunlardan c-ortalamalar kümeleme yöntemi c bölünmelere dayanır. Bulanık eşitlik ilişkisine dayalı diğer yöntemde, bulanık eşitlik 6 ilişkisine dayalı aşamalı kümeleme yöntemi olarak adlandırılır(14). Bu çalışmada G-10, Avrupa Birliği ve OECD Ülkelerinin sosyo-ekonomik göstergeleri bakımından benzerliklerinin Bulanık Kümeleme Yöntemi ile belirlenmesi amaçlanmıştır. Ülkelerin benzerlik yapıları bulanık eşitlik ilişkisine dayalı olan "Fanny Algoritması" na dayalı olarak bulunmuştur.

---

[2] [2] G10 - AVRUPA BİRLİĞİ VE OECD ÜLKELERİNİN SOSYO-EKONOMİK BENZERLİKLERİNİN FUZZY KÜMELEME ANALİZİ İLE BELİRLENMESİ
Yard. Doç. Dr. Mehmet ŞAHİN
Öğr. Gör. Bahattin HAMARAT
Çanakkale Onsekiz Mart Üniv. Turizm İşletmecilik ve Otelcilik
Y.O.http://docs.google.com/viewer?a=v&q=cache:ZcBQl6SOpCkJ:www.econ.utah.edu/~ehrbar/erc2002/pdf/P397.pdf+Bulan%C4%B1k+k%C3%BCmeleme&hl=tr&gl=tr&pid=bl&srcid=ADGEESjEOo6GKgdWiJFL9XTQQLRtbbLqLYarmQtPx4xaNY-wW9rT-5UJ1zWZEfIE9nftFqstHfR8fbvY5AQ9VKIlqNrEgI2Jc58ipNEnNpZTzX_zNbguLyVUNxCAQymMnNx_I0t7XSkv&sig=AHIEtbTfFcwkvIPI8n3SdezjcZJjN_uCDQ

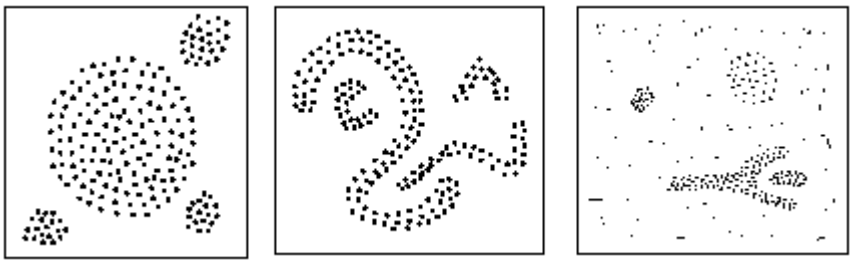
12.2 Yoğunlğa Dayalı Kümeleme

Yoğunluk Tabanlı Bir küme düşük yoğunluğa sahip bir bölge tarafından sarılmış ve yüksek nesne yoğunluğuna sahip bir bölgedir. Şekil 12.2 (a-b) deki veriye gürültü eklenerek elde edilmiş veri için çeşitli yoğunluk tabanlı kümeleri göstermektedir. Bir kümenin yoğunluk tabanlı tanımlaması daha çok kümeler düzensiz yada birbirlerine geçmiş iken ve aynı zamanda gürültü ve dışsallar (outlier) var iken kullanılır.



Şekil 12.2 Komşuluk ve yoğunluk tabanlı kümeler

12.3 DBSCAN Algoritması



Yoğunluk tabanlı kümelemede, alanlar veri yoğunluğunun fazla ve az olmasına göre belirlenir. DBSCAN yoğunluk tabanlı kümeleme yapan basit ve etkin bir algoritmadır[3].

DBSCAN algoritması

1-Tüm noktaları çekirdek,kenar ya da gürültü noktalar olarak işaretle.
2-Gürültü noktaları çıkar.
3-Birbiriyle Eps çapı içersindeki tüm çekirdek noktalar arasına bir kenar çiz.
4-Her bağlı çekirdek nokta grubu için bir küme oluştur.
5-Çekirdek noktalara göre her bir kenar noktayı bir kümeye dahil et.

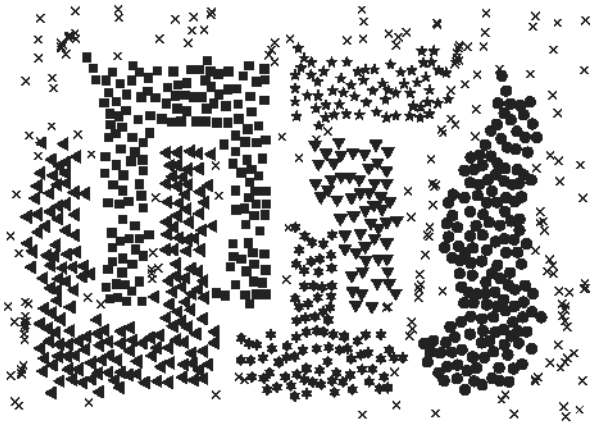
Birbirine yeteri kadar yakınlıkta (en fazla Eps) olan iki çekirdek nokta aynı kümeye konur. Aynı şekilde çekirdek noktaya yeteri kadar yakınlıkta olan bir kenar nokta çekirdek noktayla aynı kümeye yerleştirilir.(Bir kenar noktanın başka bir kümedeki çekirdek noktaya olan uzaklığı da dikkate alınmalıdır.) Gürültü noktalar çıkartılır.

DBSCAN algoritması bir kümeyi belirlemek için önce keyfi olarak seçilen bir p noktasıyla işe başlar. Minpts ve Eps koşulları sağlanacak şekilde p noktası üzerinden yoğunluğa erişebilecek tüm noktaları

[3] www.bilmuh.gyte.edu.tr/~htakci/vm/**veri**madenciligi.doc

toplar. P bir iç nokta ise, buraya kadar yapılan işlemler bir küme oluşturmuş olur. P bir dış nokta ise ya da p üzerinden hiçbir noktaya erişilemiyorsa, bu p noktası bırakılır ve yeni bir nokta seçilerek işlemlere devam edilir.

DBSCAN kümenin yoğunluk-tabanlı tanımını kullandığından gürültüye dayanıklıdır. Farklı şekilde ve büyüklükte kümeleri ele alabilir. Bu yüzden, DBSCAN algoritması K-mean ile bulunamayan birçok kümeyi bulabilir. Önceden bahsedildiği gibi kümeler arasındaki yoğunluk farkı büyükse DBSCAN algoritması problem yaşayacaktır. Ayrıca yüksek boyutta verilerde de problem yaşayacaktır çünkü böyle veriler için yoğunluk belirlemek oldukça zordur. DBSCAN algoritmasında yakın komşulukların aranması tüm çiftlerin yakınlıklarının hesaplanmasını gerektiriyorsa (genelde yüksek boyutlu verilerde oluşan bir durumdur) maliyeti yüksek olacaktır.



Şekil 12.3 DBSCAN ile oluşmuş küme örneği

12.4 Karınca Koloni Optimizasyonu

Günümüzde biyolojik yapılardan esinlenerek oluşturulan sistemler önem kazanmakta ve yapay zeka araştırmacılarının ilgisini çekmektedir[4].

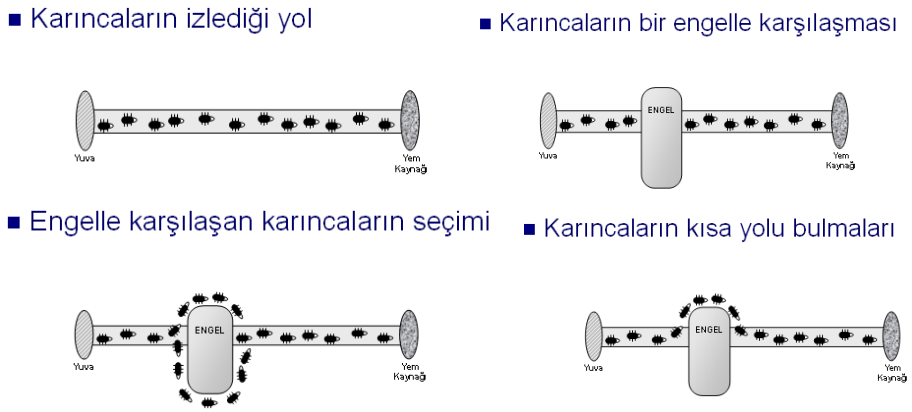
Doğadaki bazı sosyal sistemler (örneğin arılar, karıncalar ve hatta bakteriler), sınırlı yetenekli basit bireyler tarafından oluşturulmalarına rağmen Kolektif Zekâ (Swarm Intelligence – SI) davranışı sergilerler. Problemlere üretilen zeki çözümler, bu bireylerin kendi içerisindeki organizasyonları ve dolaylı iletişimlerinden ortaya çıkar.

Karıncaların yiyecek ile yuva arasındaki yolu nasıl buldukları ilk bakışta anlaşılması zor bir problemdir. Bu sorunun cevabı geçtikleri yolda bıraktıkları feromon (pheromone) izlerinde gizlidir.

Her karınca yiyecek ararken geçtiği yerlere yani rotasına değişik miktar ve yoğunluklarda, karın bölgesinde yer alan Dufour bezlerinde salgıladığı feromon adlı özel bir sıvı bırakır. Bu koku karıncaya, yuvasına dönmesini sağladığı gibi, diğer karıncalara da yiyeceğe giden yolu göstermeye yarayan bir izdir.

Karınca ancak feromon kokularını takip ederek doğru yolu bulabilmektedir, çünkü karıncalar neredeyse kör varlıklardır.

Araştırmacılar karıncaların davranışlarını taklit ederek geliştirilen bu yeni problem çözme metodunu Karınca Koloni Optimizasyonu – KKO (Ant Colony Optimization – ACO) olarak adlandırmaktadırlar. Karınca tabanlı algoritmalarda temel fikir, basit iletişim mekanizmalarını kullanan yapay akıllı etmenlerin (agent), birçok karmaşık problem için çözümler üretebilmesidir (Şekil 12.4).



Şekil 12.4 Karıncalar yuvalarından yem kaynağına giderler ve geri dönerler, Bir süre sonra, karıncalar yem kaynağına giden en kısa yolutercih etmeye başlarlar. Karıncalar feromon izleri yolu ile iletişim kurarlar ve yüksek feromona sahip yolu takip ederler

[4] Gülizar Çit, Yüksek Lisans Tezi,SAÜ Fen Bilimleri Enstitüsü, 2005

Karıncaları model alarak geliştirilen KKO yönteminde karıncalardan aynen alınan özellikler şunlardır:

- Karıncalar arasında feromon aracılığı ile kurulan iletişim
- Feromon miktarının fazla olduğu yolların öncelikle tercih edilmesi,
- Kısa yollar üzerinde feromon miktarının daha hızlı artması

KKO yöntemine gerçek karıncalardan farklı olarak eklenen yeni özellikler de şu şekidedir :

- Zamanın ayrık olarak hesaplandığı bir ortamda yaşarlar.
- Tamamen kör olmayıp, problem ile ilgili detaylara erişebilirler.
- Belli bir miktar hafıza ile problemin çözümü için oluşturdukları bilgileri tutabilirler.

KKO Algoritması

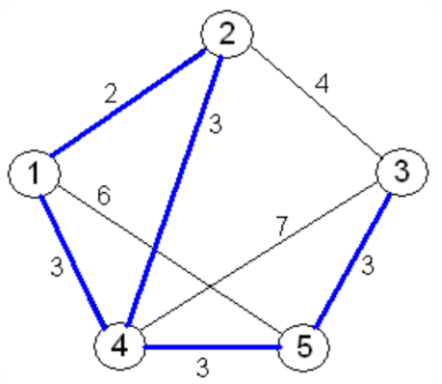
```
Feromon İzini Belirleyen Parametrelerin Başlatılması

while (Sonuç şartlar sağlanana kadar) do
    Çözümlerin Oluşturulması
    Lokal Aramanın Uygulanması
    İzin Güncelleştirilmesi
end
end
```

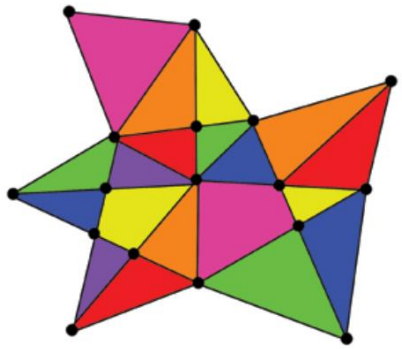
Karınca koloni optimizasyonu algoritmalarının uygulamaları ilk olarak gezgin satıcı problemi olarak adlandırılan, bir satıcının kendi şehrinden başlayarak, malını satacağı şehirleri, mümkün olan en kısa yolu kullanarak dolaşıp, tekrar kendi şehrine dönmesi problemi üzerine uygulanmıştır

Gezgin Satıcı Problemi ilk bakışta basit gibi görünen kombinasyonel bir problemdir:

Bir satıcı N adet şehri (veya düğümü) dairesel olarak (yani işi bittiğinde başladığı noktaya geri gelecek şekilde) gezerken minimum mesafe kat etmelidir. Bu şehirleri gezerken tüm şehirlere uğramak zorunda olduğu gibi bir şehirden de sadece bir kere geçebilir.



Grafik Renklendirmede kullanılmaktadır:



Rota belirleme, Çizelgeleme, İletişim ağı tasarımı, Kümeleme ve Sınıflandırma alanlarında da kullanımlaktadır.

Haberleşme ağlarında kullanılan yönlendirici sinyallerin en kısa rotadan gönderilmesi, trafik sıkışıklığının önlenmesi gibi problemlerin de bu yöntemle kolayca çözülebileceği düşünülmektedir. Karınca Kolonisi Yönlendirmesinin son derece esnek olması ve ağa yeni kanalların eklenmesi veya çıkarılması gibi değişikliklerin kolayca adapte edilebilmesi de önemli avantajları arasında sayılmaktadır.

Karıncaların bu önemli özellikleri Hewlett-Packard ve British Telecom'daki araştırmacılar tarafından iletişim ağlarının dengelenmesi ve mesaj rotalaması problemlerinde kullanılmıştır
Son yıllarda ise, veri madenciliğinde de kullanılan kümeleme ve sınıflandırma problemlerinin çözümü için geliştirilen karınca koloni optimizasyonu algoritmaları mevcuttur

T : İterasyon Sayısı
K : Küme sayısı
N : Nesne sayısı
n : Nitelik sayısı
R : Ajan (yazılım karıncası) sayısı
L : Lokal arama yapılacak ajan sayısı (L=%20 R)
S : Çözüm kataı (S_1,S_2,S_3,…,S_R,)
LS : Lokal arama çözüm kataı (LS_1,LS_2,…,LS_L)

τ : Feromon matrisi (NxK boyutlu)
τ_0 : Feromon başlangıç matrisi
τ_{ij} : *j*. kümeye ait *i*. örneğin feromon yoğunluğu
x_{iv} : *i*. örneğin *v*. niteliğinin değeri
q_0 : (0,1) aralığında eşik olasılığı
p_{ls} : (0,1) aralığında lokal arama eşik olasılığı

ρ : [0,1] aralığında izin devam etmesi
(1-ρ) : buharlaşma oranı
U : Uygunluk Değeri

w : NxK boyutlu merkezin ağırlığı
w_{ij} : Küme *j* ile ilişkili x_i nesnesinin ağırlığı

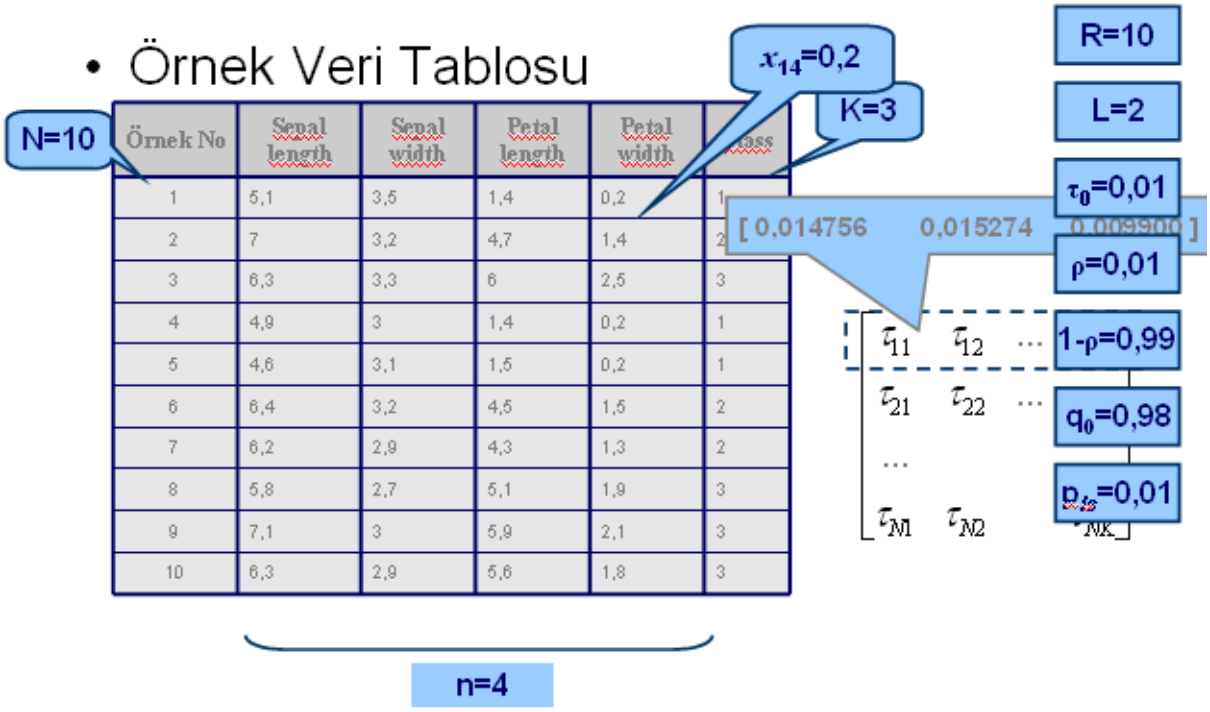
$$w_{ij} = \begin{cases} 1 & \text{eğer } i \text{ nesnesi küme } j\text{'de yer alıyorsa} \\ 0 & \text{diğer} \end{cases}$$

m : Kxn boyutlu matrisin küme merkezidir
m*jv* : Küme *j*'deki bütün örneklerin *v*. nitelik değerlerinin ortalaması

$$m_{jv} = \frac{\sum_{i=1}^{N} w_{ij} . x_{iv}}{\sum_{i=1}^{N} w_{ij}}$$

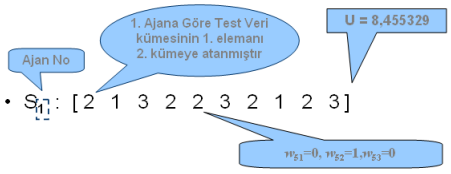
Her bir ajan için "Uygunluk (Fitness)" değerini hesaplanır

$$F(w,m) = \sum_{j=1}^{K} \sum_{i=1}^{N} \sum_{v=1}^{n} w_{ij} \left\| x_{iv} - m_{jv} \right\|^2$$



Örnek No	Sepal length	Sepal width	Petal length	Petal width	Class
1	5,1	3,5	1,4	0,2	1
2	7	3,2	4,7	1,4	2
3	6,3	3,3	6	2,5	3
4	4,9	3	1,4	0,2	1
5	4,6	3,1	1,5	0,2	1
6	6,4	3,2	4,5	1,5	2
7	6,2	2,9	4,3	1,3	2
8	5,8	2,7	5,1	1,9	3
9	7,1	3	5,9	2,1	3
10	6,3	2,9	5,6	1,8	3

- S_1 : [K_1 K_2 K_3 ... K_N]



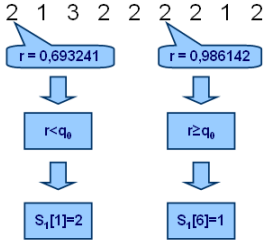
- Her etmen için S çözümlerini yeniden oluşturma

(i) k=1
(ii) S_k'nın her bir elemanı için (0,1) aralığında r gibi bir random sayı üretilir
(iii) Eğer r ≥ q_0 ise, (1,K) aralığında bir j tamsayısı random olarak seçilir ve $S_k(i)=j$ olarak atanır, i = 1,...,N.
(iv) k = k + 1; eğer k ≤ R ise Adım (ii)'ye git
değilse dur.

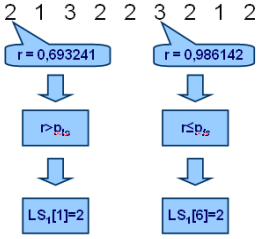
- Lokal arama algoritması

(i) k = 1
(ii) S_t, geçici bir çözüm olsun ve $S_t(i) = S_k(i)$ olsun, i = 1,...,N.
(iii) S_t'nin her bir i elemanı için, (0,1) aralığında r gibi bir random sayı üretilir.
Eğer r ≤ p_{ls} ise, (1,K) aralığında bir j tamsayısı, $S_k(i)\neq j$ olmak kaydıyla, random olarak seçilir ve $S_t(i)=j$ olarak atanır.
(iv) Çözüm kataı S_t ile ilgili küme merkezlerini ve ağırlıkları hesaplanır be U_t amaç fonksiyon değerini hesapla.
Eğer Ut, Uk'dan daha küçükse, $S_k = S_t$ ve $U_k = U_t$ olarak atanır.
(v) k = k + 1; eğer k ≤ L ise Adım (ii)'ye git
değilse dur.

- LS_1: [2 1 3 2 2 2 2 1 2 3]



- S_1 : [2 1 3 2 2 3 2 1 2 3]



Feromon güncelleme

- $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{l=1}^{L} \Delta\tau_{ij}^{l}$

- $\Delta\tau_{ij}^{l} = \begin{cases} 1/U_l & \text{eğer } l. \text{ Karınca tarafından üretilen çözümün } i. \\ & \text{elemanının değeri küme } j\text{'ye eşitse} \\ 0 & \text{diğer} \end{cases}$

Genel olarak, herhangi bir iterasyon seviyesinde, algoritma temel olarak üç adım gerçekleştirir:

- Önceki iterasyondan elde edilen feromon iz bilgisini kullanarak yazılım karıncaları tarafından yeni R çözümlerin elde edilmesi
- Yeniden üretilmiş çözümler üzerinde lokal arama işleminin gerçekleştirilmesi
- Feromon iz matrisinin güncellenmesi.

Algoritma adımları aşağıdaki gibi özetlenebilir.