Yönetim, Yıl: 19, Sayı: 59, Şubat 2008

KÜMELEME İÇİN BİR BENZETİLMİŞ TAVLAMA ALGORİTMASI YAKLAŞIMI

Dr. Tunçhan CURA
İstanbul Üniversitesi
İşletme Fakültesi
Sayısal Yöntemler Anabilim Dalı

Literatürde kümeleme için önerilmiş bir çok sayıda algoritmik yaklaşım vardır ve önemli bir kısmı da sezgisel yaklaşımlardır. Bu çalışmada bir sezgisel yaklaşım türü olan benzetilmiş tavlama yöntemiyle bir kümeleme algoritması önerilmiştir. Önerilen algoritma Gaussian Olasılık Dağılımına göre tesadüfi oluşturulan yapay problemlerde denenmiştir. Karşılaştırma yapabilmek için sıklıkla kullanılan SPSS 12.0 yazılımının "K-Means Cluster", "Hierarchical Cluster", "Twostep Cluster" araçlarından ve Lingo8' den yararlanılmıştır. Sonuçlar önerilen yöntemin sağlıklı olduğunu ortaya koymuştur.

Key Words: Clustering, simulated annealing, heuristic.

A SIMULATED ANNEALING ALGORITHM APPROACH TO CLUSTERING

As the relevant literature is surveyed, plenty of algorithmic approaches appear and many of those are heuristic techniques. The clustering algorithm that has been proposed in this study is based on simulated annealing method which is a type of heuristic algorithms. This algorithm has been tested in artificial problems which have been randomly generated with respect to the Gaussian Probability Distribution. The "K-Means Cluster", "Hierarchical Cluster", and "Twostep Cluster", which are the frequently used tools of SPSS 12.0 software, and the Lingo8 software have been exploited for comparison. The results have shown that the proposed technique is robust.

Anahtar Sözcükler: Kümeleme, benzetilmiş tavlama, sezgisel.

GİRİŞ

Kümeleme, bir veri kümesindeki herbir verinin özelliklerine bağlı olarak daha hassas tek bir alt küme içerisinde toplanmasıdır. Sözkonusu veri kümesi, özellikleri olan nesneler kümesi olarak temsil edilebilir. Buna göre herhangi bir i nesnesinin diğer bir j nesnesiyle olan zıtlığı d_{ij} ile temsil edilsin. Her bir alt küme içerisinde d_{ij} değerleri en küçük olan nesneler yeralmalıdır (bakınız Bryson ve Inniss, 2007, 3255). Görüldüğü gibi kümeleme, en uyumlu nesnelerin birarada bulunduğu alt kümelerin tespit edilmesine yönelik bir optimizasyon problemi olmaktadır.

Kümelemeye yönelik çeşitli çalışmalar vardır. Bunların başında McQueen' in (1967) önermiş olduğu ve sıklıkla kullanılan K-means algoritması gelmektedir. Bu yaklaşımla her bir kümede yeralan nesnelerin özellik ortalamaları gözönünde bulundurularak küme içerisinde yeralan her bir nesnenin ortalamadan uzaklığı dikkate alınmıştır. Başka bir yaklaşım Johnson' un (1967) önerdiği hiyerarşik kümeleme şemalarıdır. Bu yaklaşımda her nesne kendi kümesine yerleştirilir ve içlerinde birbirine en yakın iki küme birleştirilir. Böylece küme sayısı bir azaltılır. Istenen küme sayısına ulaşılana kadar işlem devam ettirilir. Buna benzer bir başka hiyerarşik yaklaşım Ward (1993) tarafından öne sürülmüştür.

Kümeleme için bunların dışında sıklıkla başvurulan çeşitli sezgisel yaklaşımlar da vardır. Örneğin Shelokar ve diğerleri (2004) tarafından önerilmiş olan karınca kolonileri optimizasyonu yaklaşımı, Liu ve diğerlerinin (2004) önerdikleri genetik algoritmalar yaklaşımı ile Güngör ve Ünler' in (2007) önermiş oldukları tabu arama yaklaşımı verilebilecek örneklerin başında yeralmaktadır.

Bu çalışmada kümeleme için herbir nesnenin bulunduğu kümenin merkezine olan uzaklıklarının toplamının en küçük olduğu çözümü araştıran bir benzetilmiş tavlama algoritması önerilmiştir. Sonuçlar SPSS 12.0 yazılımının "twostep cluster", "K-means cluster", "hierarchical cluster" araçlarının ve Lingo8' in bulduğu çözümlerle karşılaştırılmıştır. Karşılaştırılmaların yapılması için kullanılan problemler Gauss olasılık dağılımına göre tesadüfî olarak oluşturulmuştur.

1. KÜMELEME PROBLEMİ

Daha önce de değinildiği gibi bu çalışmada herbir alt küme içerisinde yer alan nesnelerin, içinde bulundukları kümenin merkezine olan öklid uzaklıkları temel alınmıştır. Buna göre nesne sayısı, nesnelerin özellik sayısı, i nesnesinin özelliklerinin değerleri, küme sayısı ve herbir kümenin merkezi sırasıyla N, n, x_{il} (l = 1, ..., n), K ve m_{kl} (k = 1, ..., K) ile temsil edilirse, herhangi bir i nesnesinin (i = 1, ..., N) k kümesinin merkezine uzaklığı d_{ik} ile temsil edilir ve aşağıdaki gibi hesaplanır:

$$d_{ik} = \sum_{l=1}^{n} \left(x_{il} - m_{kl}\right)^2 \qquad i = 1, ..., N,\ k = 1, ..., K \tag{1}$$

Görüldüğü gibi (1) de nesnelerin küme merkezlerine olan uzaklıkları hesaplanmaktadır. Ancak küme merkezinin nasıl hesaplandığı bu noktada daha önemli bir sorun olmaktadır. Bir kümenin merkezi, içinde yeralan nesnelerin özelliklerinin ortalamasıdır. Buna göre i nesnesinin k kümesinin içinde olup olmadığını temsil eden bir W_{ik} bir-sıfır değişkeni kullanılabilir. Böylece W_{ik} = 1 ise i nesnesi k kümesinde yeralacaktır, W_{ik} = 0 ise yeralmayacaktır. Bir kümenin merkezi aşağıdaki gibi hesaplanır:

$$m_{kl} = \frac{\sum_{i=1}^{N} W_{ik} x_{il}}{\sum_{i=1}^{N} W_{ik}} \tag{2}$$

Buna göre, kümeleme probleminin bu çalışmada kullanılan matematik modeli (1) ve (2)' nin yardımıyla aşağıdaki gibi olur:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{l=1}^{n} W_{ik} (x_{il} - m_{kl})^2 \\
s.t \quad & \sum_{k=1}^{K} W_{ik} = 1, i = 1, \ldots, N, \\
& \sum_{i=1}^{N} W_{ik} \geq 1, k = 1, \ldots, K, \\
& m_{kl} = \frac{\sum_{i=1}^{N} W_{ik} x_{il}}{\sum_{i=1}^{N} W_{ik}}, k = 1, \ldots, K, l = 1, \ldots, n, \\
& W_{ik} \in (0,1), i = 1, \ldots, N, k = 1, \ldots, K.
\end{aligned}$$

2. BENZETİLMİŞ TAVLAMA ALGORİTMASI

Benzetilmiş Tavlama (BT) algoritması Metropolis ve diğerlerinin (1958) çalışmasını temel alan Kirkpatrick ve diğerleri (1983) tarafından ortaya atılmıştır. BT eşik algoritmalarının bir türüdür (Bakınız

Aarts E. ve Lenstra J. K., 2003, 92 - 94). Adından da anlaşılacağı gibi BT, bir maddenin ısıtılıp yavaşça soğumaya bırakılması durumunda çevresiyle termal eşitliğe ulaşana kadar yavaşça soğuması sürecini taklit eder. Sözkonusu süreçte ısı seviyesinin yüksek olduğu durumlarda maddenin enerjisi azami düzeydedir ve parçacıkları tesadüfi olarak dağılmaktadır. Ancak ısı seviyesi termal eşitlik noktasına ulaştığında ise maddenin enerjisi asgari düzeyde olur ve parçacıklar kuvvetli yapısal bir bütünlüğe ulaşmış olurlar.

Fiziksel tavlama süreci bir optimizasyon problemi çözümünün aranması sürecine benzetilebilir. Buna göre yüksek ısı seviyesine karşılık gelen bir tesadüfi başlangıç çözümünün enerjisi amaç fonksiyonunun değerine karşılık gelmektedir. Böylece çözüm aranırken, t ısı seviyesinde i çözümünden j komşu çözümüne geçme olasılığı aşağıdaki gibi hesaplanır:

$$P_t\{j' \, yi \; kabul \; et\} = \min\left(e^{\frac{f(i)-f(j)}{t}}, 1 \right) \tag{3}$$

Görüldüğü gibi BT algoritmasına göre yüksek ısı seviyeli durumlarda daha uzaktaki komşu çözümlerin kabul edilme olasılığı daha yüksek olmakta, böylece bu sıçrama hareketleri sayesinde lokal en küçük noktaya takılma sorunu engellenebilmektedir.

3. KÜMELEME için BT TASARIMI

BT algoritmarının en önemli unsurlarından birisi yukarıda da kısaca girişi yapılmış olan komşuların araştırılmasıdır. Adından da açıkça görüldüğü gibi en küçük enerjili durum araştırılırken, tekrarla bir pozisyonun komşuları araştırılır ve (3)' teki fonksiyona göre bulunan komşu çözüm kabul edilir veya edilmez. Böylece söz konusu yeni çözüm en düşük enerjinin aranmasında yeni odak noktası olmaktadır.

Bu çalışmada K adet kümeye N adet nesne başlangıçta tesadüfi olarak dağıtılmaktadır. Bu uygulama BT algoritmalarının en temel başlangıç çözümü belirleme biçimidir. Söz konusu durum en yüksek ısı seviyesinin olduğu durumdur. Bu ilk çözüm yukarıdaki gibi i çözümü olarak nitelenecek olursa, j komşu çözümü i' de tesadüfi olarak iki kümenin seçilmesi ve bu iki küme içerisinde tesadüfi iki pozisyondaki elemanların yerdeğiştirilmesi ile belirlenir.

Matematiksel olarak daha rahat izah edebilmek için çözüm s matrisiyle temsil edilmektedir. Sözkonusu matrisin boyutu K x N' dir. Başka bir ifadeyle satırlarda kümeler kolonlarda da nesneler bulunmaktadır. Ancak herhangi bir kolonda sıfır değeri bulunabilir, bunun anlamı ilgili pozisyonda herhangi bir nesne bulunmamakta olduğudur. Buna göre s matrisinin her satırının bir kümeyi temsil ettiği açıkça görülmektedir.

	1	2	3	4	5	6
s_1	0	1	5	0	0	2
s_2	0	0	3	0	0	0
s_3	4	0	0	6	0	0

Şekil 1. Herhangi bir çözümü temsil eden s matrisi

Her satır bir vektör olduğundan kümeler bu çalışmada vektör olarak temsil edilmişlerdir. Şekil 1' de 3 kümenin ve 6 nesnenin olduğu örnek bir s çözüm matrisi gösterilmektedir. Buna göre s_1 vektörü incelendiğinde birinci kümeye 1 inci, 5 inci ve 2 inci nesnelerin yerleştirilmiş olduğu görülmektedir.

4. ENERJİ FONKSİYONU

```
g(s_k)
s_k: s çözüm matrisinin k ıncı satırı
begin
  o_l = 0, l = 1, ..., n
  sonuç = 0
  for l = 1 to n
    eleman_sayısı = 0
    for i = 1 to N
      if s_ki > 0 then
        o_l = o_l + x_(s_ki)l
        eleman_sayısı = eleman_sayısı + 1
      End if
    End for
    if eleman_sayısı = 0 then
      sonuç = 10^50          // Her bir kümede en az bir nesne bulunmalı
      Algortimayı sonlandır
    End if
    o_l = o_l / eleman_sayısı
  End for
  for l = 1 to n
    for i = 1 to N
      if s_ki > 0 then
        sonuç = sonuç + (o_l - x_(s_ki)l)^2
      End if
    End for
  End for
  return sonuç
End
```

Şekil 2. s_k vektörünün enerji fonksiyonu.

Bu çalışmada önerilen BT algoritması çözüm matrisi yaklaşımında her bir satır (vektör) bir kümeyi temsil etmektedir. Buna göre herhangi bir vektörün enerji fonksiyonu kod taslağı şekil 2' de verilen $g(s_k)$ fonksiyonuyla temsil edilecek olursa, mevcut çözümün toplam enerjisi (4)' deki gibi hesaplanır.

$$f_i(s) = \sum_{k=1}^{K} g(s_k) \tag{4}$$

5. İYİLEŞTİRME OPERATÖRÜ

Bilindiği gibi BT algortiması komşu çözümleri tesadüfi olarak belirlemektedir. Aslında, sezgisel yaklaşımların hemen hepsinin mantığı kısmen de olsa tesadüflüğe dayanmaktadır. Bazı durumlarda mevcut çözüm en iyi çözüme çok yakın olmasına rağmen komşu çözümün tesadüfen oluşturulması sabebiyle en iyiye ulaşması çok güç olabilir. Örneğin üç kümenin ve 150 nesnenin olduğu bir durum düşünülebilir. Böyle bir durumda öyle bir çözüme ulaşılmış olsun ki yalnız 100 numaralı nesne 2. kümeden alınıp 3. kümeye konulsa en iyi çözüme ulaşılacak olsun. İşte böyle bir komşunun seçilebilme olasılığı çok düşüktür. Bu durumda şekil 3' te kod taslağı verilmiş olan iyileştirme operatörü bu amaç için kullanılabilir. Görüldüğü gibi bu operatör iç içe girmiş döngüler yardımıyla ikili olarak mevcut çözüm matrisinin hücrelerindeki değerleri yer değiştirmektedir. Böylece örnekteki 100 numaralı nesne 2. kümeden alınıp 3. kümeye kesinlikle yerleştirilecektir.

6. KÜMELEME için BT ALGORİTMASI

Buraya kadar anlatılanlar biraraya getirildiği taktirde şekil 4' te gösterilen Kümeleme BT Algoritması

```
gk = g(sk)  k = 1, ..., K.
for k = 1 to K
   for k* = k + 1 to K
      for i = 1 to N
         for i* = 1 to N
            if ski ≠ sk*i* then
               ski ile sk*i* ' ı yerdeğiştir
               if    gk + gk* ≤ g(sk) + g(sk*)
then
                  ski ile sk*i* ' ı yerdeğiştir
               else
                  gk = g(sk)
                  gk* = g(sk*)
               End if
            End if
         End for
      End for
   End for
End for
F = Σ_{k=1}^{K} g(sk)
```

Şekil 3. *s* çözüm matrisi için iyileştirme operatörü

elde edilir. Görüldüğü gibi algoritmada bazı sabit değerler kullanılmıştır. Bu değerlerin hepsi ampirik olarak belirlenmiştir. Örneğin *iyileştirme operatörü*' nün çağırılma olasılığı 0.001, başlangıç ısı seviyesi 1000, son ısı seviyesi 16' dan büyük ve soğuma programı $t_d = t_{d-1} \times 0.92$ (d: döngü numarası) olarak belirlenmiştir.

```
t = 1000
tesadüfi olarak başlangıç s matrisini oluştur
F = f(s)
F^eniyi = F
s^eniyi = s
fk = g(sk), k = 1, ..., K
while t > 16 do
   for nrep = 1 to 500
      k = round(random(0, 1)×(K - 1)) +1
      do
         k* = round(random(0, 1)×(K-1))+1
      untill k ≠ k*
      i = round(random(0, 1) × (N - 1)) + 1
      do
         i* = round(random(0,1)×(N - 1))+1
      untill ski ≠ sk*i*
      ski ile sk*i* ' ı yerdeğiştir
      Δ = fk + fk* − g(sk) − g(sk*)
      if random(0, 1) < min( e^(Δ/t) , 1) then
         fk = g(sk)
         fk* = g(sk*)
         F = F - Δ
      Else
         ski ile sk*i* ' ı yerdeğiştir
      End if
      if random(0, 1) < 0.001 then
         s matrisi için iyileştirme
            operatörünü çağır
      End if
      if F < F^eniyi then
         s^eniyi = s
         F^eniyi = F
      else if random(0, 1) < 0.001 then
         F = F^eniyi
         s = s^eniyi
      End if
   End for
   t = t × 0.92
End while
```

Şekil 4. Kümeleme BT Algoritması

7. KARŞILAŞTIRMALI PROBLEMLER

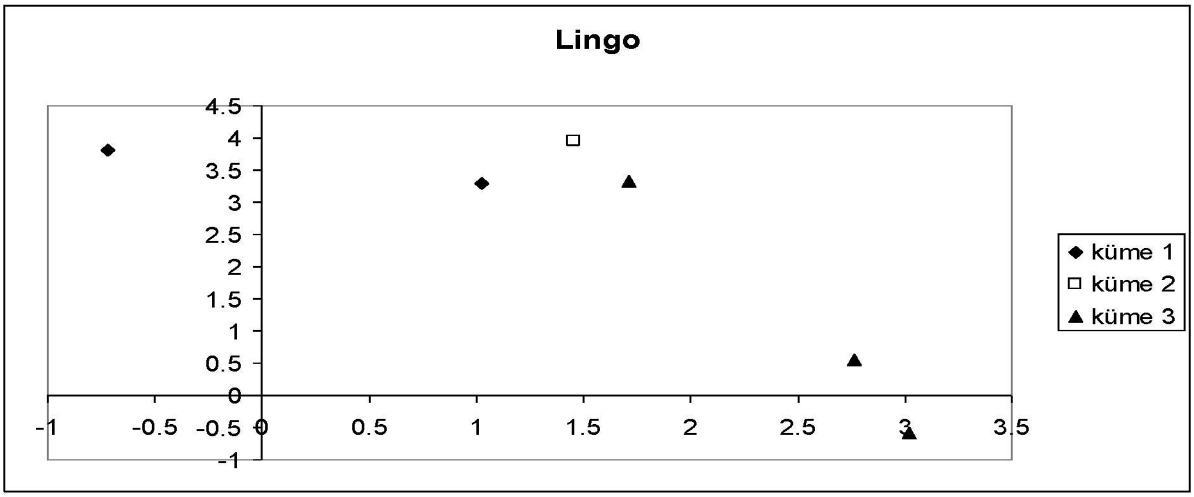
Bu kısımda bazı kümeleme problemleri önerilen BT algoritmasını test etmek için kullanılmaktadır. Bunun için dört farklı kümeleme problemi Gaussian dağılımına göre tesadüfî olarak oluşturulmuştur. Tüm problemlerde sonuç grafiklerinin rahat gösterilebilmesi için $n = 2$ alınmıştır. Problemler Pentium M 2.13 GHz kişisel bilgisayarda denenmiştir.

Problem 1

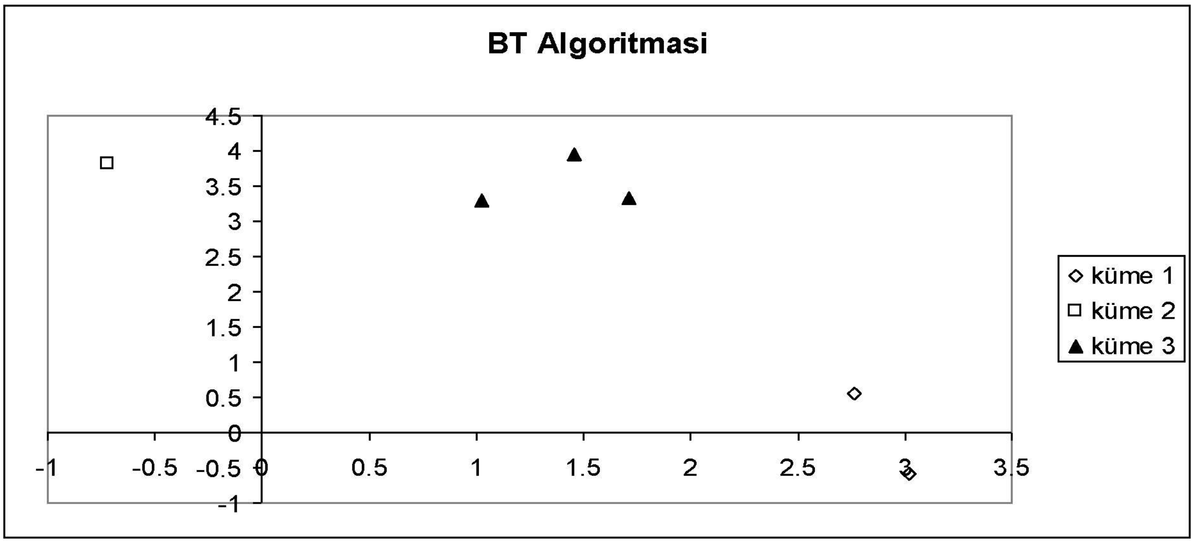
İlk problem birinci kısımda verilmiş olan matematiksel optimizasyon modelinin bir yazılımla (Lingo8) incelenmesi amacıyla oluşturulmuştur. Model son derece basit görünmektedir. Ancak küme ortalamalarının dinamik oluşu ve optimize edilecek değişkenlere bağlı olarak değişmesi uygun (feasible) bir çözümün dahi bulunabilmesini çok güçleştirmektedir. Buna göre çok basit bir problem olarak $N = 6$ ve $K = 3$ olduğu durum incelenmiştir. Problem ortalamaların $\mu_1 = \{3, 0\}$, $\mu_2 = \{0, 3\}$ ve $\mu_3 = \{1.5, 2.5\}$ standart sapmaların ise $\sigma_1 = \{0.3, 1\}$, $\sigma_2 = \{1, 0.5\}$ ve $\sigma_3 = \{2, 1\}$ olduğu Gaussian dağılımından tesadüfi olarak örneklerin seçilmesiyle oluşturulmuştur. Böylece Lingo8 modeli şekil 5' teki gibidir. Şekil 6' da Lingo8' in bulduğu sonuç, şekil 7' de ise BT algoritmasının bulduğu sonuç gösterilmektedir. Görüldüğü gibi Lingo8 ancak uygun bir çözüm bulabilmiştir. Amaç fonksiyon değeri D ile temsil edilecek olursa $D_{Lingo8} = 10.719$ iken $D_{BT} = 1.182$ dir.

```
MODEL:
sets:
  K/1..3/;
  N/1..6/;
  p/1..2/;
  Nxp(N, p): x;
  KxN(K, N): W;
  Kxp(K, p): m;
endsets
data:
x = 3.183821505591241,1.5159106555167672, ...;
enddata
  min = @sum(K(k): @sum(N(i): @sum(p(l): W(i, k) * (x(i, l) - m(k, l)) * (x(i,
l) - m(k, l)))));
  @for(N(i): @sum(K(k): W(i, k)) = 1);
  @for(K(k): @sum(N(i): W(i, k)) >= 1);
  @for(KxN: @bin(W));
  @for(K(k): @for(p(l): m(k, l) = @sum(N(i): W(i, k) * x(i, l)) / @sum(N(i):
W(i, k))));
END
```

Şekil 5. Kümeleme için Lingo8 modeli



Şekil 6. Lingo8' in bulduğu sonuç.



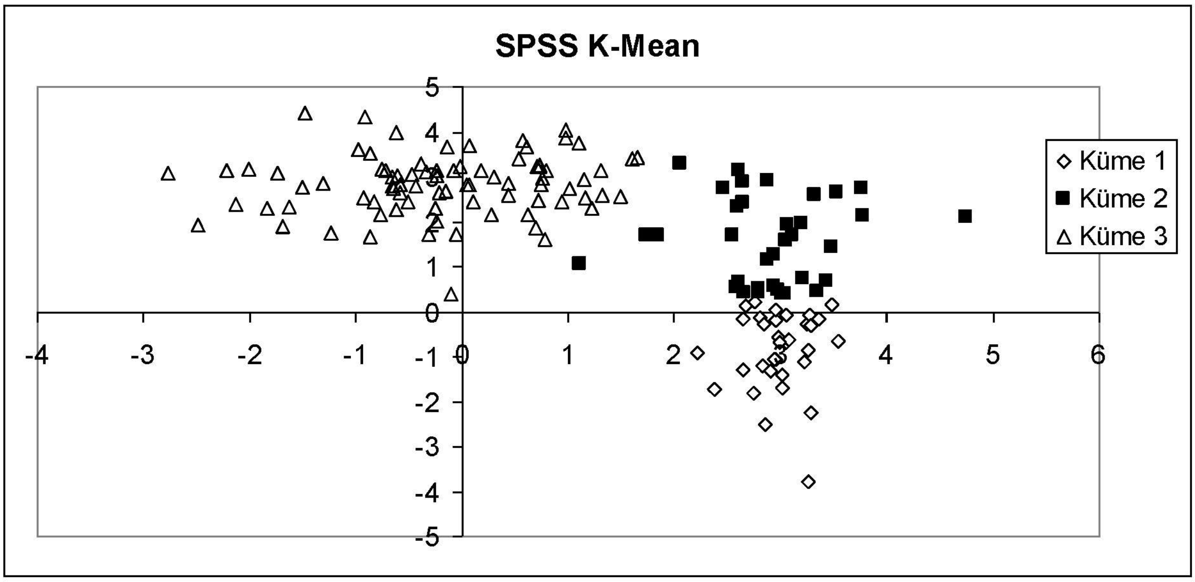
Şekil 7. BT algoritmasının bulduğu sonuç

Problem 2

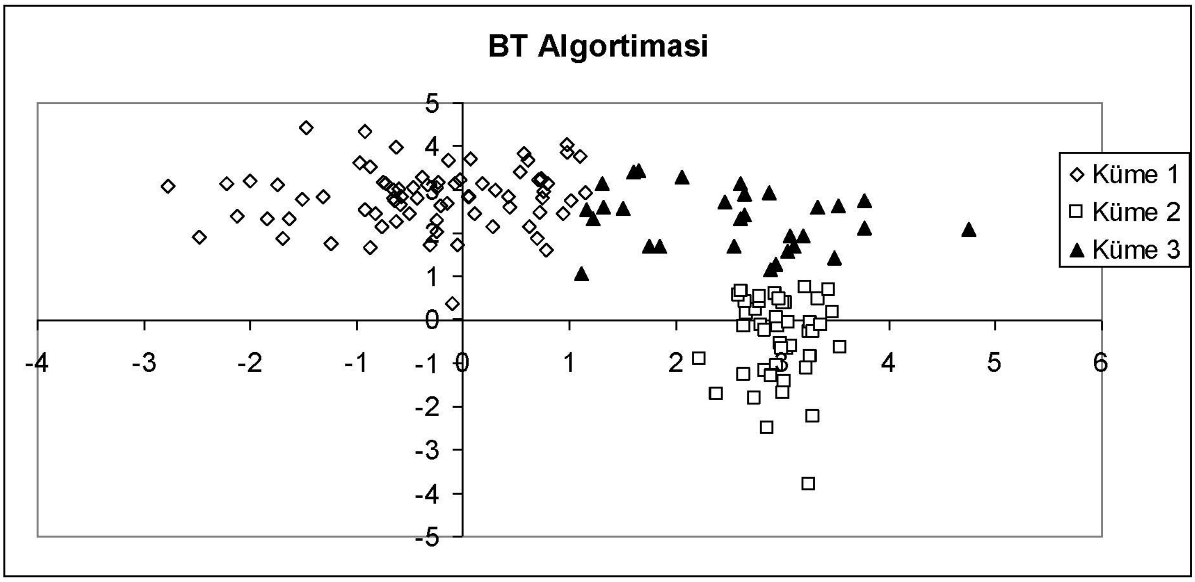
İkinci problemde N = 150 ve K = 3 tür. Problem ortalamaların $\mu_1 = \{3, 0\}$, $\mu_2 = \{0, 3\}$ ve $\mu_3 = \{1.5, 2.5\}$ standart sapmaların ise $\sigma_1 = \{0.3, 1\}$, $\sigma_2 = \{1, 0.5\}$ ve $\sigma_3 = \{2, 1\}$ olduğu Gaussian dağılımından tesadüfi olarak örneklerin seçilmesiyle oluşturulmuştur. BT algoritması 9.203 saniyede sonucu bulmuştur. Diğer teknikler ile karşılaştırmalı sonuçlar tablo 1' de verilmiştir. En iyi iki çözüm olan SPSS 12.0 K-Means tekniği ile BT' nın sonuç grafikleri sırasıyla şekil 8 ve şekil 9' da gösterilmiştir.

Tablo 1. Problem 2' nin tüm tekniklere göre amaç fonksiyonu değerleri

Kullanılan Teknik	***D***
BT Algortiması	183.0604
SPSS K-Means	191.3374
SPSS Hierarchical	270.59
SPSS Twostep	823.9007



Şekil 8. SPSS 12.0 K-Means sonucu.



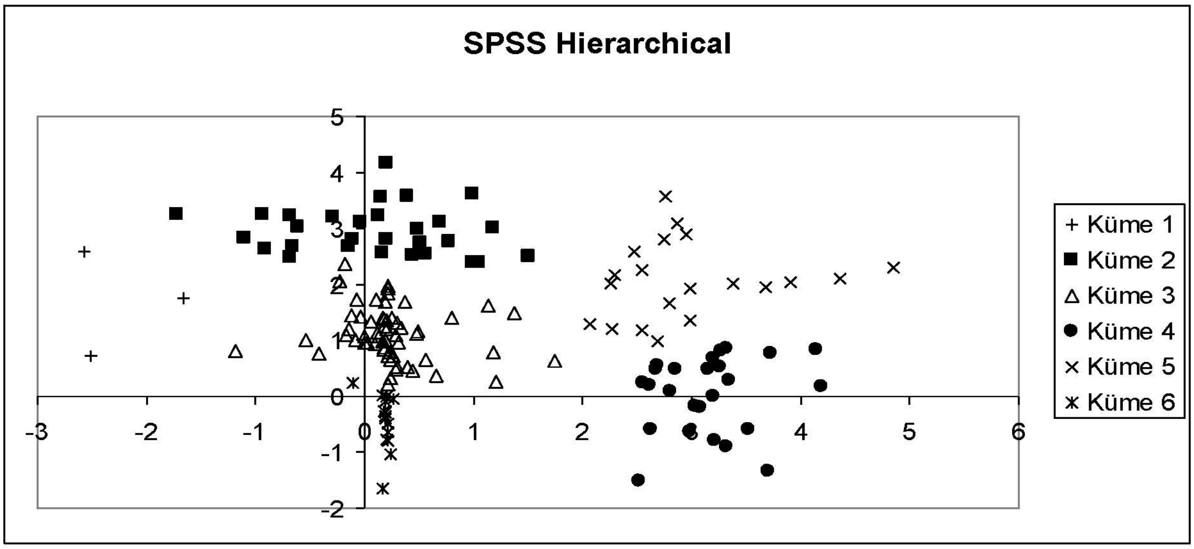
Şekil 9. BT algoritmasının bulduğu sonuç

Problem 3

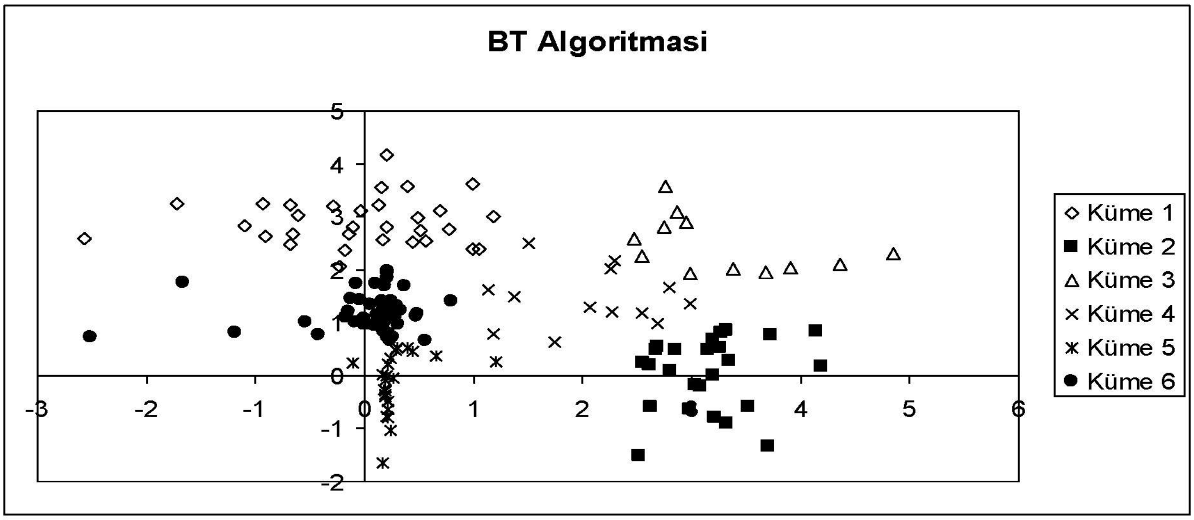
Üçüncü problemde N = 150 ve K = 6 dır. Problem ortalamaların $\mu_1 = \{3, 0\}$, $\mu_2 = \{0, 3\}$, $\mu_3 = \{1.5, 2.5\}$, $\mu_4 = \{0.2, 0.1\}$, $\mu_5 = \{1.2, 0.8\}$ ve $\mu_6 = \{0.1, 1.1\}$ standart sapmaların ise $\sigma_1 = \{0.3, 1\}$, $\sigma_2 = \{1, 0.5\}$, $\sigma_3 = \{2, 1\}$, $\sigma_4 = \{0.03, 1\}$, $\sigma_5 = \{2, 0.5\}$ ve $\sigma_6 = \{0.2, 0.4\}$ olduğu Gaussian dağılımından tesadüfi olarak örneklerin seçilmesiyle oluşturulmuştur. BT algoritması 16.453 saniyede sonucu bulmuştur. Diğer teknikler ile karşılaştırmalı sonuçlar tablo 2' de verilmiştir. En iyi iki çözüm olan SPSS 12.0 Hierarchical tekniği ile BT' nın sonuç grafikleri sırasıyla şekil 10 ve şekil 11' de gösterilmiştir.

Tablo 2. Problem 3' ün tüm tekniklere göre amaç fonksiyonu değerleri

Kullanılan Teknik	***D***
BT Algortiması	88.7992
SPSS K-Means	100.9936
SPSS Hierarchical	87.01999
SPSS Twostep	527.228



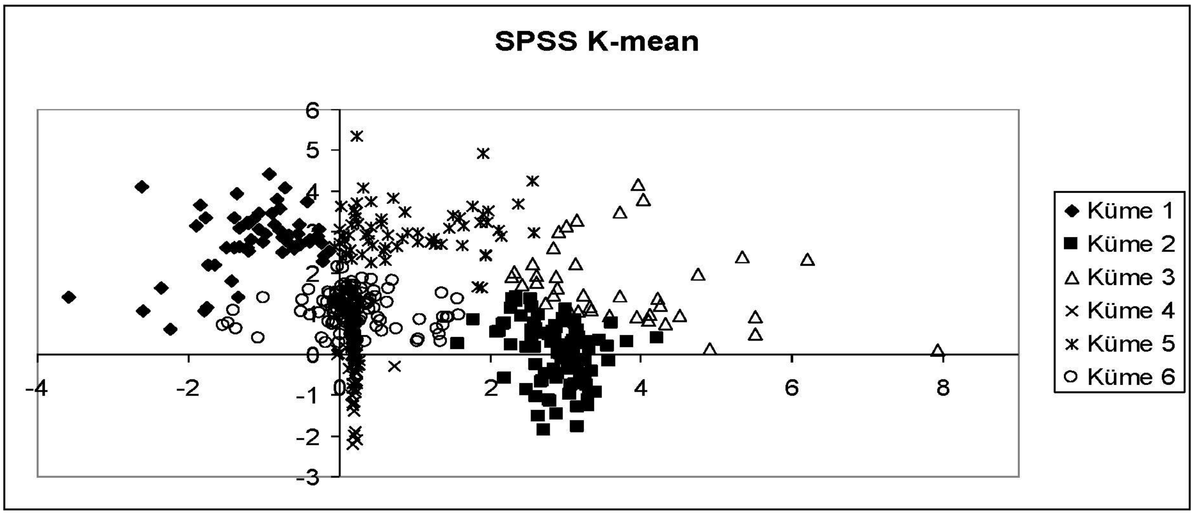
Şekil 10. SPSS 12.0 Hierarchical sonucu



Şekil 11. BT algoritmasının bulduğu sonuç

Problem 4

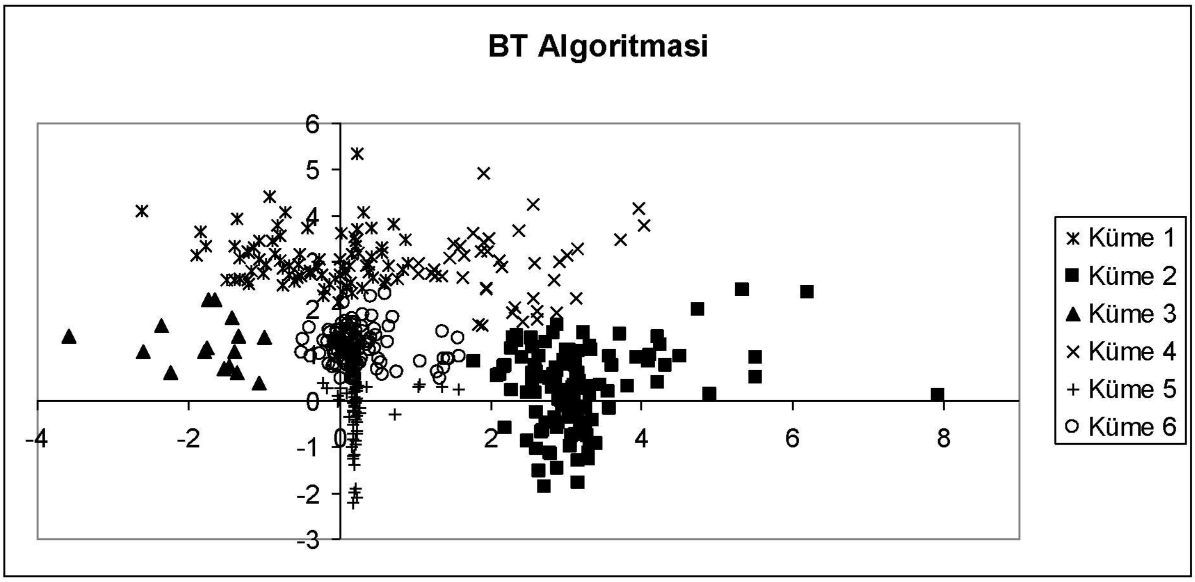
Dördüncü problemde N = 450 olmak kaydıyla diğer parametereler üçüncü problem için kullanılanla aynı olarak alınmıştır. BT algoritması 325.578 saniyede sonucu bulmuştur. Diğer teknikler ile karşılaştırmalı sonuçlar tablo 3' te verilmiştir. En iyi iki çözüm olan SPSS 12.0 K-Mean tekniği ile BT' nın sonuç grafikleri sırasıyla şekil 12 ve şekil 13' te gösterilmiştir.



Şekil 12. SPSS 12.0 K-Mean sonucu

Tablo 3. Problem 4' ün tüm tekniklere göre amaç fonksiyonu değerleri

Kullanılan Teknik	D
BT Algortiması	367.3931
SPSS K-Mean	369.9486
SPSS Hierarchical	876.0893
SPSS Twostep	1996.158



Şekil 13. BT algoritmasının bulduğu sonuç

SONUÇ

Bu çalışmada önerilen yöntem, en sık kullanılan yüntemlerden bazıları ile kıyaslanmıştır. Sonuçlar incelendiğinde bir problem hariç hepsinde önerilen BT algoritması daha başarılı olmuştur. Söz konusu problemde ise en iyi çözüme oldukça yakın bir çözüm elde edilmiştir. Çalışma süreleri incelendiğinde algoritmanın problem boyutuna bağlı olarak makul sürelerde çözümü bulduğu söylenebilmektedir. Böylece sonuç olarak BT yaklaşımının kümeleme için başarıyla kullanılabileceği görülmektedir.

KAYNAKÇA

Aarts E., Lenstra J. K., 2003, **Local Search in Combinatorial Optimization**, Princeton University Press, USA.

Bryson K. M. O., Inniss T. R., 2007, *"A hybrid clustering algorithm"*, **Computers & Operations Research**, 34, 3255 – 3269.

Güngör Z., Ünler A., (Baskıdaki Makale), *"K-Harmonic Means Data Clustering with Tabu-Search Method"*, **Applied Mathematical Modeling.**

Johnson S., 1967, *"Hierarchical clustering schemes"*, **Psychometrika**, 2, 241–254.

Kirkpatrick S., Gerlatt C. D. Jr., Vecchi M.P., 1983, *"Optimization by Simulated Annealing"*, **Science,** 220, 671-680.

Liu Y., ChenK., Liao X., Zhang W., 2004, *"A genetic clustering method for intrusion detection"*, Pattern Recognition, 37, 927 – 947

McQueen J., 1967, *"Some methods for classification and analysis of multivariate observations"*, **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**, 281–297.

Metropolis N., Rosenbluth A.W., Rosenbluth M. N., Teller A.H., Teller E., 1958, *"Equations of State Calculations by Fast Computing Machines"*, **J. Chem. Phys.**, 21, 1087- 1092.

Shelokar P.S., Jayaraman V.K., Kulkarni B.D., 2004, *"An ant colony approach for clustering"*, **Analytica Chimica Acta**, 509, 187 – 195

Ward J., 1993, *"Hierarchical grouping to optimize an objective function"*, **Journal of American Statistical Association**, 58, 236–244.