



BULANIK KARAR VERME SİSTEMLERİNDE PARALEL HESAPLAMA PARALLEL COMPUTING IN FUZZY DECISION MAKING SYSTEMS

Serkan BALLI^{1*}, Bahadır KARASULU²

¹Bilişim Sistemleri Mühendisliği Bölümü, Teknoloji Fakültesi, Muğla Sıtkı Koçman Üniversitesi, 48187, Muğla.
serkan@mu.edu.tr

²Bilgisayar Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Çanakkale Onsekiz Mart Üniversitesi, 17020, Çanakkale.
bahadirkarasulu@comu.edu.tr

Geliş Tarihi/Received: 14.02.2012, Kabul Tarihi/Accepted: 07.06.2012

doi: 10.5505/pajes.2013.91300

*Yazışılan yazar/Corresponding author

Özet

Bazı karar problemlerinde, alternatif, kriter ve karar vericilerin sayısı çok yüksektir. Bu yüzden hesaplama işlemi daha zor, zaman alıcı ve karmaşık bir hal alır. Bu karmaşık işlemleri daha kısa sürede gerçekleştirilebilmek için paralel hesaplama gibi popüler teknolojiler mevcuttur. Bu çalışmada, Bulanık Analitik Hiyerarşi Süreci (BAHS) ve TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) yöntemlerinden oluşan bulanık bir karar verme sisteminde paralel hesaplamanın tasarlanması ve gerçekleştirimi incelenmiştir. Paralel hesaplama, hesaplamanın en yoğun olduğu BAHS adımında gerçekleştirilmiştir. Önerilen yöntem, homojen ve heterojen bilgisayarlarda ayrı ayrı denenmiş ve sonuçlar tartışılmıştır.

Anahtar kelimeler: Bulanık karar verme, Paralel hesaplama, Bulanık analitik hiyerarşi süreci, TOPSIS.

Abstract

In some decision problems, the numbers of alternatives, criteria and decision-makers are very high. Therefore the calculation process becomes more difficult, time consuming and complex. To achieve these complex tasks in a shorter time, popular technologies such as parallel computing are available to use. In this study, design and implementation of parallel computation in a fuzzy decision making system which consists of the methods: Fuzzy Analytic Hierarchy Process (FAHP) and TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) was investigated. Parallel computing was carried out in FAHP phase that is most intensive phase of calculation. Proposed method was tested in homogeneous and heterogeneous computers separately and results were discussed.

Keywords: Fuzzy decision making, Parallel computing, Fuzzy analytic hierarchy process, TOPSIS.

1 Giriş

Paralel hesaplama; uzun ve karmaşık bir hesaplama işleminin sonuçlarını, seri hesaba göre daha hızlı elde etmek için işlemin parçalara ayrılıp uyarlanması yolu ile çok sayıda işlemcide eş zamanlı olarak işletilmesidir. Paralel bilgisayarlar için popüler bir taksonomi ilk kez Michael Flynn, [1] tarafından tanımlanmıştır. Bu taksonominin temel hedefi, bir problemi çözülmesi için problemi, daha küçük alt görev parçalarına bölmektir. Böylece bu parçalar eş zamanlı çalışacak şekilde düzenlenebilmektedir [2]. Çok çeşitli paralel bilgisayar (işlemci) yapıları vardır. Bu yapılar, işlemciler (işleme elemanları) arasındaki veya işlemci ve bellek arasındaki bağlantıya göre belirlenirler. Genellikle, paralel bilgisayarlar komutların işlenme biçimine göre sınıflandırılırlar. Buna göre paralel bilgisayarlar; işlemcilerin aynı zamanda aynı komutları işlemelerine yani tek komut/çoklu veri (SIMD) akışına göre veya her bir işlemcinin farklı komutları işlemesi yani çoklu komut/çoklu veri (MIMD) akışına göre adlandırılırlar. Bunun yanında paralel işlemci makineleri simetrik (tüm işlemcilerin aynı seviyede olması) ve asimetrik (işlemcilerin bazı görevler için ayrılması ve önceliklerinin olması) çoklu işlemciler olmak üzere iki ana gruba ayrılırlar.

Simetrik Çoklu İşlemciler (SÇİ), iki veya daha fazla özdeş işlemeye sahip olabilen bir bilgisayar mimarisini temsil ederler. Bu işlemciler tek bir paylaşımlı ana belleğe erişmektedirler. Günümüzde birçok güncel sistem SÇİ mimarisini kullanmaktadır. Çok-çekirdekli işlemcilerde, SÇİ mimarisi çoklu çekirdeklerin her birine ayrı birer işlemciymiş gibi uygulanır. Uygun işletim sistemi desteği ile verinin bellekte nerede olduğunun pek önemli olmadığı bir biçimde

herhangi bir iş yükü için herhangi bir işlemcinin çalıştırılabilmesine olanak sağlar. Ayrıca SÇİ sistemler işlemciler arası yük dengesine de olanak tanır.

Asimetrik Çoklu İşlemciler (ASÇİ) ise, özel bir iş için özelleştirilmiş ayrı işlemcileri kullanırlar, bu durum doğal olarak karmaşıklığı da artırmaktadır. Bilgisayar kümelerinden oluşan çoklu-işlemcili sistemler (NASA'nın bir projesi olarak başlayan Beowulf projesinde olduğu gibi), tüm işlemciler için tüm bellek kapasitesi erişilebilir değildir. Bu çalışmada; bulanık karar verme sistemleri için kullanılan sistem, simetrik çoklu işlemciler sınıfına girmektedir.

Karar problemleri genel olarak birden fazla kriter içerir ve klasik çok kriterli karar verme (ÇKKV) algoritmaları kullanılarak çözülebilmektedir. Bu problemler çoğu zaman belirsiz ve doğrusal olmayan özellikler içermektedir. Bu tür belirsizlik içeren durumlarda doğrusal olmayan özelliklerin modellenmesi için klasik karar verme yöntemleri yetersiz kalmakta ve bulanık mantık, bulanık çıkarım ve yapay sinir ağları gibi yapay zeka (YZ) tekniklerine ihtiyaç duyulmaktadır.

Paralel hesaplama yüksek hesaplama gücü gerektiren problemler için çok farklı alanlarda uygulanmaktadır. Bu çalışmada ele alınan karar verme konusundaki paralel hesaplama ile ilgili araştırmalara bakılacak olursa; Gergel ve Strongin, [3] global olarak optimal karar vermeye ait zaman- alan problemler için bilgisayar kümeleri sistemleri üzerinde paralel hesaplama için yeni bir şema önermişlerdir. Herhangi bir merkezi işlemcinin olmadığı bu tekdüze şema, çok boyutlu problem indirgeme fikrini temel alır. Yeni çoklu haritalamaların (Peano eğri tipi gibi) kullanımı ile aynı şekilde, birçok boyutlu problemin çok değişkenli problemlere indirgenmesinin paralel bir yoldan yapılabileceğini

belirtmişlerdir. Bu yolla her bir işlemci bir diğer işlemci tarafından elde edilen bilgiyi paylaşmaktadır. Rahimi v.d., [4] bulanık TOPSIS kullanan web tabanlı ve yüksek başarımlı bir tıbbi teşhis sistemi geliştirmişlerdir. Bulanık TOPSIS'e göre hesaplanan çok fazla sayıda giriş değerleri ve hesaplama olduğundan çalışma süresini kısaltmak için bu hesaplamaların eş zamanlı olarak çalışmasını öngörmüşler ve hesaplamaların paralelleştirilmesini gerçekleştirmişlerdir. ÇKKV ile ilgili tekniklerden olan Pareto yüzeyi oluşturulmasında, sonlu bir eleman analizi, benzetim veya çok kriterli yaklaşımlar içeren her bir fonksiyonun değerlendirilmesi işi hesaplama olarak oldukça karmaşık hale gelmektedir. Wuppalapati v.d., [5] hesaplama süresinin bu açıdan düşürülebilmesi için paralel işlemcilerin kullanılması tekniğini çalışmalarında tartışmışlardır. Bir test problemi olarak bir yapısal topoloji problemi seçilmiş ve Pareto yüzeyi iki farklı küme kullanılarak oluşturulmuştur. Yamamoto v.d., [6] istatistiksel verilerin hesaplanması için Jasp istatistiksel analiz sisteminde yüksek düzeyde paralel hesaplama fonksiyonları geliştirmişlerdir. Bu çalışmalardan paralel hesaplama ile iki veya daha fazla işlemci (veya bilgisayar) kullanımının seri hesaplama göre daha avantajlı yanları olduğu ortaya çıkmaktadır.

Bu çalışmada ise işletim sistemi seçim problemi için BAHS ve TOPSIS yöntemlerinden oluşan bulanık bir karar verme sisteminde paralel hesaplamaların tasarlanması ve gerçekleştirimi ele alınmıştır. Paralel hesaplama, hesaplamaların en yoğun olduğu BAHS adımında gerçekleştirilmiştir. Önerilen yöntem, homojen ve heterojen bilgisayarlarda ayrı ayrı denenmiş ve sonuçlar tartışılmıştır.

Makalenin ikinci bölümünde bulanık karar verme, BAHS ve TOPSIS yöntemlerinin detayları verilmektedir. Üçüncü bölümde geliştirilen paralel hesaplamalı karar verme algoritması anlatılmakta ve dördüncü bölümde elde edilen bulgular sunulmaktadır. Beşinci bölümde sonuçlar ve tartışmalara yer verilmektedir.

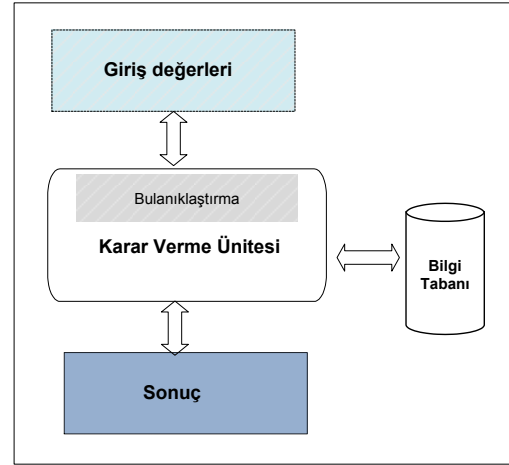
2 Bulanık Karar Verme

Genel olarak, kesin olmayan bilginin veya bir tercih yapısının gösterilmesinde kullanılan bulanık mantık, Zadeh, [7] tarafından geliştirilmiştir. Bulanık mantık ve kümelerin ana özelliği belirsizliğin modellenmesindeki yeteneğidir [8]. Endüstriyel alanda belirsiz sistemlerin modellenmesinde tam ve kesin bilgi olmadan da karar vermeyi kolaylaştırır [9]. Klasik küme teorisinde bir üye kümeye aittir ya da değildir yaklaşımı vardır. Bu teoride aitlik konusunda tam ve keskin bir ayırım söz konusudur bu yüzden üyelik sınırları da tam ve kesin olarak bellidir. Fakat gerçek hayatta karşılaşılan çoğu problemde klasik küme teorisi yetersiz kalmaktadır [10].

Bulanık küme, klasik kümenin bir uzantısıdır. Klasik kümeler sadece tam üyeliği veya üye olmamayı gösterirken, bulanık kümeler aynı zamanda kısmi üyelik de sunarlar. Bulanık veriler daha esnek ve bulanık veri kullanılması ile daha hassas sonuçlar elde edilir [11]. Bellman ve Zadeh, [12] bulanık karar verme teorisi olarak bilinen yeni bir yöntem ortaya koymuşlardır. Bulanık karar vermenin ana çalışma alanı belirsizlik altında karar vermedir. Çünkü elimizde kriterlere, alternatiflere ve sonuçlara ilişkin sayısal değerler değil sözlü ifade edilen dilsel değerler mevcuttur ve bu belirsizlik oluşturur.

Bulanık Karar Verme algoritmik olarak Şekil 1'de gösterilmiştir. Giriş değerleri karar verme ünitesine gönderilir ve burada bulanıklaştırma yapılır ve ilgili karar verme modeli gerçekleştirilerek sonuç elde edilir. Karar verme yöntemi olarak çalışmada ele alınan çok kriterli karar verme teknikleri

BAHS ve TOPSIS yöntemleri sonraki bölümlerde detaylı olarak anlatılmaktadır.



Şekil 1: Bulanık karar verme.

2.1 Bulanık Analitik Hiyerarşi Süreci

Analitik Hiyerarşi Süreci Saaty, [13] tarafından geliştirilen bir ÇKKV yöntemidir. Bu çalışmada, BAHS için üçgen bulanık sayılar ve ikili karşılaştırmalardaki sentetik mertebeler için merkezi analiz kullanılmıştır.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ nesne kümemiz ve $G = \{g_1, g_2, g_3, \dots, g_n\}$ ise hedef kümemiz olsun.

Chang, [14]-[15]'in merkezi analiz yöntemine göre her bir nesne alınmakta ve her bir g_i hedefi için sırasıyla merkezi analiz yapılmaktadır. Böylece her bir nesne için aşağıda gösterildiği gibi m merkezi analiz değeri bulunmaktadır:

$$M^1_{gi}, M^2_{gi}, \dots, M^m_{gi}, \quad i = 1, 2, \dots, n,$$

Burada, M^j_{gi} ($j = 1, 2, \dots, m$) değerlerinin hepsi üçgen bulanık sayıdır. Bundan sonra gerçekleştirilecek olan Chang'ın merkezi analiz aşağıdaki adımlardan oluşmaktadır.

Adım 1: i . nesneye ait bulanık sentetik merkezi değeri şu şekilde bulunur:

$$S_i = \sum_{j=1}^m M^j_{gi} \otimes \left[\sum_{i=1}^n \sum_{j=1}^m M^j_{gi} \right]^{-1} \quad (1)$$

$\sum_{j=1}^m M^j_{gi}$ elde etmek için, aşağıdaki gibi m merkezi değerleri

için bulanık toplama işlemi gerçekleştirilir:

$$\sum_{j=1}^m M^j_{gi} = \left(\sum_{j=1}^m l_j, \sum_{j=1}^m m_j, \sum_{j=1}^m u_j \right) \quad (2)$$

ve $\left[\sum_{j=1}^m M^j_{gi} \right]^{-1}$ elde etmek için ise M^j_{gi} ($j = 1, 2, \dots, m$)

değerleri için aşağıdaki gibi bulanık toplama işlemi gerçekleştirilir:

$$\sum_{i=1}^n \sum_{j=1}^m M^j_{gi} = \left(\sum_{i=1}^n l_i, \sum_{i=1}^n m_i, \sum_{i=1}^n u_i \right) \quad (3)$$

ve sonra vektörün tersi aşağıdaki formülle bulunur:

$$\left[\sum_{i=1}^n \sum_{j=1}^m M_{gi}^j \right]^{-1} = \left(\frac{1}{\sum_{i=1}^n u_i}, \frac{1}{\sum_{i=1}^n m_i}, \frac{1}{\sum_{i=1}^n l_i} \right) \quad (4)$$

Adım 2: $\tilde{M}_1 = (l_1, m_1, u_1)$ ve $\tilde{M}_2 = (l_2, m_2, u_2)$, iki üçgen bulanık sayı olmak üzere, $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ gerçekleşme olasılığı şu şekilde hesaplanır:

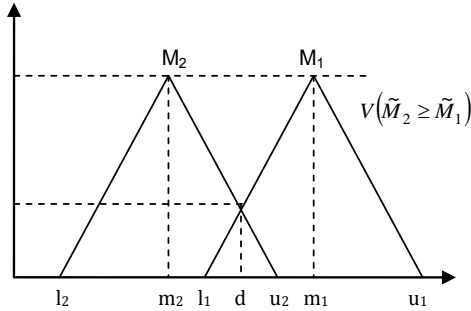
$$V(\tilde{M}_2 \geq \tilde{M}_1) = \sup_{y \geq x} \left[\min(\mu_{\tilde{M}_1}(x), \mu_{\tilde{M}_2}(y)) \right] \quad (5)$$

ve bu aşağıdaki eşitliğe eşittir:

$$V(\tilde{M}_2 \geq \tilde{M}_1) = \text{hgt}(\tilde{M}_1 \cap \tilde{M}_2) = \mu_{M_2}(d) \quad (6)$$

$$= \begin{cases} 1, & \text{eger } m_2 \geq m_1 \\ 0, & \text{eger } l_1 \geq u_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & \text{diğer durumlarda} \end{cases} \quad (7)$$

Şekil 2'de eşitlik 11'e göre d noktası μ_{M_1} ve μ_{M_2} arasındaki en büyük kesişme noktası D'nin ordinatıdır. M_1 ve M_2 'yi karşılaştırmak için $V(M_1 \geq M_2)$ ve $V(M_2 \geq M_1)$ değerlerine ihtiyacımız vardır.



Şekil 2: M_1 ve M_2 'nin kesişimi.

Adım 3: Bir konveks bulanık sayının k konveks bulanık $M_i (i=1, 2, k)$ sayıdan büyük olma olasılığı şu şekilde tanımlanmaktadır:

$$\begin{aligned} V(M \geq M_1, M_2, \dots, M_k) \\ = V \left[(M \geq M_1) \text{ and } (M \geq M_2) \text{ and } \dots \text{ and } (M \geq M_k) \right] \\ = \min V(M \geq M_i), \quad i = 1, 2, 3, \dots, k \end{aligned} \quad (8)$$

her bir $k = 1, 2, \dots, n; k \neq i$ için $d(A_i) = \min V(S_i \geq S_k)$ olduğunu varsayalım. Bu durumda ağırlık vektörü aşağıdaki gibidir:

$$W' = (d'(A_1), d'(A_2), \dots, d'(A_n))^T \quad (9)$$

Burada, $A_i = (i = 1, 2, \dots, n)$ n alternatiflerdir.

Adım 4: Normalizasyon yoluyla elde edilen normalize edilmiş ağırlık vektörü aşağıdaki gibidir:

$$W = (d(A_1), d(A_2), \dots, d(A_n))^T \quad (10)$$

Burada W bulanık değildir.

2.2 TOPSIS Metodu

İlk olarak Hwang ve Yoon, [16] tarafından ortaya konulan TOPSIS, uygulanması daha basit yaklaşımlardan biridir. Bu tekniğe göre, en iyi alternatif, pozitif ideal çözüme en yakın ve negatif ideal çözüme en uzak alternatiftir [17]. Metot aşağıdaki adımlardan oluşmaktadır:

Adım 1. Karar matrisi eşitlik 15'deki gibi normalize edilir:

$$r_{ij} = \frac{w_{ij}}{\sqrt{\sum_{j=1}^J w_{ij}^2}} \quad j = 1, 2, 3, \dots, J \quad i = 1, 2, 3, \dots, n \quad (11)$$

Adım 2. Ağırlıklandırılmış normalize matris aşağıdaki gibi elde edilir:

$$v_{ij} = w_{ij} * r_{ij}, \quad j = 1, 2, 3, \dots, J, \quad i = 1, 2, 3, \dots, n \quad (12)$$

Adım 3. Pozitif ideal çözüm ve negatif ideal çözüm aşağıdaki gibi bulunur:

$$A^* = \{v_1^*, v_2^*, \dots, v_n^*\} \quad \text{Maksimum değerler} \quad (13)$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\} \quad \text{Minimum değerler} \quad (14)$$

Adım 4. Her bir alternatifin pozitif ve negatif ideal çözümden uzaklığı hesaplanır:

$$d_i^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2}, \quad j = 1, 2, \dots, J \quad (15)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \quad i = 1, 2, \dots, J \quad (16)$$

Adım 5. Her bir alternatifin yakınlık katsayıları aşağıdaki formülle hesaplanır:

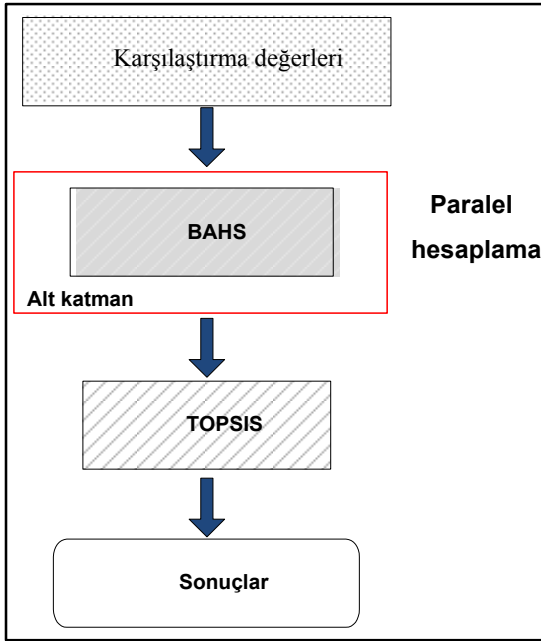
$$CC_i = \frac{d_i^-}{d_i^* + d_i^-}, \quad i = 1, 2, \dots, J \quad (17)$$

Adım 6. CC_i yakınlık katsayı değerlerinin birbiriyle karşılaştırılması ile alternatiflerin sıralaması elde edilmiş olur.

3 Paralel Hesaplamalı Karar Verme Algoritması

Ballı ve Korukoğlu, [18] çalışmasında ele alınan işletim sistemi seçimi problemi için karar algoritması Şekil 3'teki gibidir. Bu algoritmaya ek olarak BAHS hesaplamaları için paralel hesaplama yöntemi geliştirilmiştir. Seri algoritmada BAHS için karşılaştırma değerleri yapıldıktan sonra, 30 ayrı karar matrisi için değerler BAHS adımında tek tek hesaplanarak TOPSIS adımına aktarılmaktadır. Burada 30 matrisin BAHS bölümünde seri olarak hesaplanması yerine paralel hesaplama tekniği kullanılarak çözülmesi önerilmiştir.

Geliştirilen seri algoritmanın BAHS bölümüne ek olarak paralel hesaplamanın bir alt katman biçiminde eklenmesi sayesinde, hesaplamanın çekirdek kısmı olan matris hesapları paralel olarak yapılabilir hale gelmiştir. Bu yolla süreçler kendilerine ait veri kısmını yöneterek bunlardan üretilen kendilerine ait sonuçları elde edebilmişlerdir. Bu bölümden sonra, bu sonuçlar bir araya getirilerek sonraki seri çalışan TOPSIS adımına girdi olarak verilmektedirler. Bu işlem tamamen daha üst katmanlardan bağımsız olarak geliştirilen paralel hesaplama altyapısı tarafından yapılmaktadır.



Şekil 3: Karar verme algoritması.

Dağıtık ve dinamik bellekli ve bir çok işlemciye (süreç) destek verebilecek bir algoritmanın geliştirilmesi, verimi ve başarımı arttıracaktır. Seri algoritmanın geliştirilmesinden sonra, böyle bir sistemin programlanmasında mesaj geçme arayüzü (Message Passing Interface-MPI) ile yapısal ANSI C dilinin (GNU C) olanaklarının kullanılması tercih edilmiştir [19]. Program geliştirilmesinde MPI paralel kütüphanesi olan MPI'nin geniş çapta taşınabilirlik ve yüksek başarıma sahip bir gerçekleştirimi olan MPICH2 (<http://www.mcs.anl.gov/research/projects/mpich2>) kullanımı tercih edilmiştir. MPICH2 hem MPI-1, hem de MPI-2 ye ait özellikleri içerir. MPICH2'nin amaçları arasında bilgisayar kümelerini (masaüstü sistemler, paylaşımlı-bellekli sistemler, çok çekirdekli mimariler) de içeren birçok farklı hesaplama ve iletişim platformunun etkin bir şekilde desteklenmesi, genişletilmeye uygun bir çatı oluşturulması gelmektedir. MPICH2 tamamen ücretsiz olarak dağıtılır. MPI standardı paralel uygulamalar için endüstride standart haline gelmiştir.

Mesaj-Geçme programlama paradigması, en eski ve en geniş paralel bilgisayar programlama yaklaşımıdır. Paralel hesaplamada işlemciler için verinin çeşidine göre bölümlenme ve haritalama yapılmaktadır [20]. Bölümlenme, işlemcilere (süreçlere) dağıtılmak üzere veri yapılarını kişisel modüllere bölerek oluşturulan bir yol iken; haritalama, bu modüllerin veya ilgili fonksiyonların ilgili işlemcilere (süreçlere) ilgili modül gelecek şekilde atanmasıdır [2]. Çalışmamızda kullanılan paralel hesaplama tekniği, ilgili veri hesabını yapan fonksiyonların birçok süreç makinesi (hesaplama kümesine dahil edilmiş işlemciler/süreçler) arasında eşit miktarlarda bölüştürülmesi ile paylaşılması ilkesine dayanmaktadır. Bu mantıkla veri bazlı haritalama yapılmıştır.

Yazılımın geliştirilmesinden sonra yazılımın yeterince güçlü bir sistemde denenmesi aşamasına gelinmiştir. Bu aşamada 2 GB bellekli Intel® Core 2 Duo isimli iki çekirdekli işlemci'den oluşan (her biri 2.40 GHz) sekiz adet homojen donanım özelliklerine ve Windows XP 32 bit işletim sistemine sahip, birbirlerine 100 Mbps hızındaki ethernet ile ağ'a bağlı bilgisayarlar test işlemleri için seçilmiştir.

Sistem ayrıca farklı donanım özelliklerine sahip heterojen yapıdaki bilgisayarlar üzerinde de denenmiş ve sonuçlar karşılaştırılmıştır. Kullanılan heterojen bilgisayarların özellikleri Tablo 1'de verilmiştir.

Tablo 1: Heterojen bilgisayarların özellikleri.

İşletim Sistemi	İşlemci	Bilgisayar sayısı
Windows XP 32 bit	Pentium 4	3
Windows Vista 32 bit	Pentium 4	2
Windows XP 32 bit	Core 2 Duo	2
Windows XP 64 bit	Core 2 Duo	1

Yazılımın bu bilgisayarlardan biri üzerinde önce seri olarak, sonrasında çeşitli bilgisayar (işlemci) sayıları baz alınarak paralel çalıştırılması sağlanmış ve böylece deney sonuçları elde edilmiştir. Sırasıyla 1, 2, 3, 4, 6, 8 işlemci sayısı denenmiş ve başarımlar optimizasyonu dikkate alınarak yazılımda buna uygun değişiklikler de yapılmıştır. Geliştirilen yöntemde seri ve paralel çalışacak şekilde harmanlanmış algoritmaya ait algoritmik akış aşağıdaki gibidir:

1. Başla,
2. Karşılaştırma değerlerini dosyadan oku,
3. İlgili dizi ve değişkenleri tanımla,
4. Gerekli dizi elemanları ve/veya değişkenlerin değerlerini sıfırla,
5. MPI'ı başlat,
6. İşlemci sayısı ve kimliklerini tespit et,
7. Hesaplama geçen sürenin tespiti için saat tutmaya başla,
8. Kaç adet işlemci ile çalışılacağını ilgili parametreye göre belirle,
9. Giriş değerleri üzerinde BAHS yöntemini gerçekleştir,
10. Elde edilen yerel sonuçları işlemci sırası (rank) sıfır olan (yani rank numarası sıfır olan işlemci veya yönetici işlemci olarak adlandırılan süreç düğümü) üzerinde toplama,
11. Her bir adımda elde edilmiş olan BAHS sonuçlarını ilgili dosyaya yazdır,
12. Saat tutmayı bitir,
13. MPI'ı bitir,
14. BAHS'den gelen değerleri üzerinde TOPSIS yöntemini uygula,
15. TOPSIS ile elde edilen sıralama sonuçlarını dosyaya yazdır,
16. Dur.

Algoritma seri olarak başlamakta, daha sonra BAHS yöntemi paralel çalıştırılıp sonuçlar TOPSIS yöntemi ile birleştirilmektedir. Hesaplanan süreler tüm algoritma için değil paralel çalıştırılan BAHS yöntemi içindir. Yukarıdaki algoritmadaki sekizinci adımda belirtilen parametreye göre hesaplanması istenilen matrisler ilgili işlemci sayısına bölünerek işlemciler arasında dağıtılmak suretiyle (işlemci haritalama) hesaplanmaktadır. Örneğin 6 işlemcili durumda BAHS bölümündeki 30 adet matris her biri beşerlik gruplar halinde işlemcilere dağıtılmıştır. Matrisler 3x9 boyutlarındadır. Toplam 270 bulanık değer kullanılmıştır. Her bir bulanık değer 3 sayısal değerle ifade edildiği için toplamda 810 adet sayısal değer vardır.

4 Bulgular

Çalışmamızda deneyler sırasında elde edilen hızlanma oranları (başarım) ve verim yüzdeleri aşağıdaki tablo ve grafiklerde verilmektedir. Gene Amdahl, [21] tarafından ortaya konulan ve Amdahl Kanunu olarak da anılan genel bir model paralel işlemdeki hızlanma oranı için uygulanmıştır [22]. Hızlanma deyimini (S_p) için (en basit biçimde) Amdahl Kanunu formülü,

$$S_p(n_p) = \frac{T(1)}{T(n_p)} \quad (18)$$

şeklinde. Burada, n_p , işlemci sayısıdır, $T(1)$, tek işlemci çalıştırıldığında geçen süre, $T(n_p)$ ise n_p adet işlemci çalıştırıldığında geçen süredir. Verim için (en basit biçimde) kullanılan formül ise,

$$\varepsilon(n_p) = \frac{T(1)}{n_p T(n_p)} \quad (19)$$

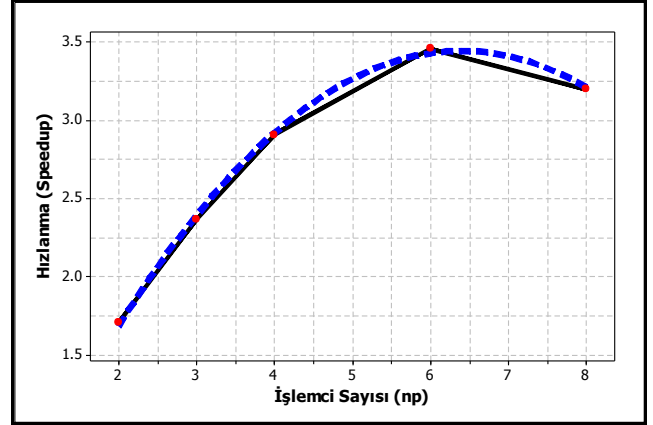
şeklinde. Ölçümler BAHS yöntemi için programın önce seri sonra da paralel çalıştırılması yoluyla elde edilmiştir. Homojen tek ve çok işlemcili program için seri hesap (yani $n_p=1$) iken ve paralel hesapta ($n_p=2, n_p=3, n_p=4, n_p=6$ veya $n_p=8$) iken geçen süreler, bu süreler için hızlanma değerleri ve bu hızlanma değerlerine ait verim değerleri Tablo 2’de gösterilmektedir.

Tablo 2: Homojen bilgisayarlar için süreler, hızlanma ve verim tablosu.

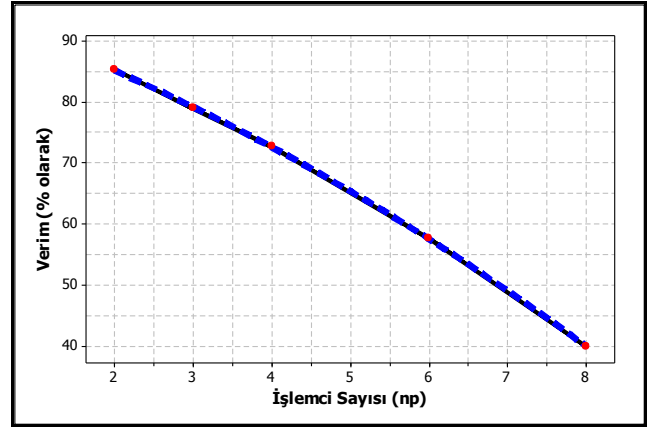
İşlemci Sayısı	Geçen Süre (saniye)	Hız Oranları	Verim (% olarak)
1	128x10 ⁻⁶	1	100
2	75x10 ⁻⁶	1.70667	85.3333
3	54x10 ⁻⁶	2.37037	79.0123
4	44x10 ⁻⁶	2.90909	72.7273
6	37x10 ⁻⁶	3.45946	57.6577
8	40x10 ⁻⁶	3.20000	40.0000

Hızlanma için bulunan regresyon fonksiyonu Hızlanma= - 0,03177 + 1,059 n_p - 0,08142 n_p^2 şeklindedir. $s=0,0663220$ ve $R^2 =99,7\%$ olarak bulunmuştur. Burada s , regresyon standart hatasını göstermektedir. R^2 ise [0,100] aralığında değer alır ve regresyon eğrisinin gerçek veriye ne kadar yaklaştığını göstermektedir. Şekil 4’teki düz çizgi ile gösterilen eğri gerçek hızlanma değerlerini, kesikli çizgilerle gösterilen eğri ise ‘karesel eğri uydurma’ ile elde edilen hızlanma grafiğini gösterir. Şekil 4’te homojen 8 işlemci ile çalıştırılan paralel programın, hızlanma grafiğinde tepe performansına ($S_p=3.45$) yani doyum noktasına, aynı anda çalışan 6 işlemcinin bulunduğu bir deney çalıştırması sırasında ulaştığı görülmektedir.

Verim için bulunan regresyon fonksiyonu Verim 104,7-8,070 n_p şeklindedir. $s=2,54739$ ve $R^2 =98,8\%$ olarak bulunmuştur. Şekil 5’ten görüleceği gibi, çalışmada 8 işlemcili benzetimdeki tepe verim değeri % 85.33’dür. Bu verim değerine ancak 2 işlemcili deneyde ulaşılması ve daha sonraki işlemci sayılarında verimin düşmesinin başlıca nedeni, problemin kendi doğası gereği işlem yapan fonksiyonların karmaşıklığı ile doğru orantılıdır. Verimin en optimal değeri yine en yüksek hızlanmanın da elde edildiği 6 işlemcili deney sırasında elde edilmiştir. Heterojen tek ve çok işlemcili bilgisayarlar için de aynı işlemler gerçekleştirilmiş ve geçen süreler, bu süreler için hızlanma değerleri ve bu hızlanma değerlerine ait verim değerleri Tablo 3’te verilmiştir.



Şekil 4: Homojen bilgisayarlar ile elde edilen hızlanma grafiği.



Şekil 5: Homojen bilgisayarlar ile elde edilen verim grafiği.

Tablo 3: Heterojen bilgisayarlar için süre değerleri, hızlanma ve verim tablosu.

İşlemci Sayısı	Geçen Süre (saniye)	Hız Oranları	Verim (% olarak)
1	140 x10 ⁻⁶	1	100
2	135 x10 ⁻⁶	1.03704	51.8519
3	97 x10 ⁻⁶	1.44330	48.1100
4	74 x10 ⁻⁶	1.89189	47.2973
6	54 x10 ⁻⁶	2.59259	43.2099
8	56 x10 ⁻⁶	2.50000	31.2500

Hızlanma için bulunan regresyon fonksiyonu Hızlanma= 0,3564+0,4838 n_p - 0,02526 n_p^2 şeklindedir. $s=0,230006$ ve $R^2 =93,5\%$ olarak bulunmuştur. Şekil 6’da heterojen 8 işlemci ile çalıştırılan paralel programın, hızlanma grafiğinde tepe performansına ($S_p=2.59$) yani doyum noktasına, aynı anda çalışan 6 işlemcinin bulunduğu bir deney çalıştırması sırasında ulaştığı görülmektedir.

Verim için bulunan regresyon fonksiyonu Verim =109,1-23,98 n_p +1,866 n_p^2 şeklindedir. $s=14,7057$ ve $R^2 =77,1\%$ olarak bulunmuştur. Şekil 7’den görüleceği gibi, çalışmada heterojen 8 işlemcili benzetimdeki tepe verim değeri % 51.85’dir. Verimin en optimal değeri yine en yüksek hızlanmanın da elde edildiği 6 işlemcili deney sırasında elde edilmiştir. Bu değer verimin alt ve üst sınırı arasındaki farkın ortalamasına oldukça yakın bir değerdir.

Homojen ve heterojen sistemlerin karşılaştırması Şekil 8 ve Şekil 9’da verilmektedir. Elde edilen değerleri tek bir grafik

üzerinde karşılaştırdığımızda, homojen sistemin farklı işlemci sayıları ile her bir deney çalıştırması için en etkin sonuçları verdiği gözlenmektedir. Heterojen sistemlerde farklı işlemci hızları, paralel hesaplama için hızlanma ve verimi düşürücü yönde etki etmiştir. Her iki sistemde de aynı algoritma aynı işletim sistemi üzerinden çalıştırılmış olmasına rağmen sadece homojen sistemde daha iyi sonuçlar elde edilmesi bu fikri desteklemektedir.

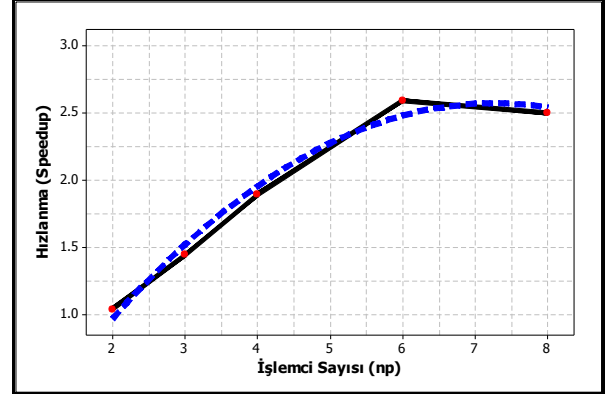
Şekil 8'de de görülebileceği gibi, homojen işlemcili sistem ile elde edilen hızlanma sonuçları, heterojen sistem ile elde edilenlere göre daha iyidir. Şekil 9'da ise verim değerleri için, homojen işlemcili sistem ile elde edilen sonuçlar heterojen sisteme göre daha fazla çıkmaktadır. Fakat 6 işlemcili optimal verim değeri geçildikten sonra iki sistem arasındaki fark gittikçe kapanmaktadır. Bunun başlıca nedeni; geliştirilen algoritma aynı kalmakla beraber, problemin kendi doğası gereği işlem yapan fonksiyonların karmaşıklığı ile verim düşüşünün doğru orantılı olmasıdır.

5 Sonuç ve Tartışma

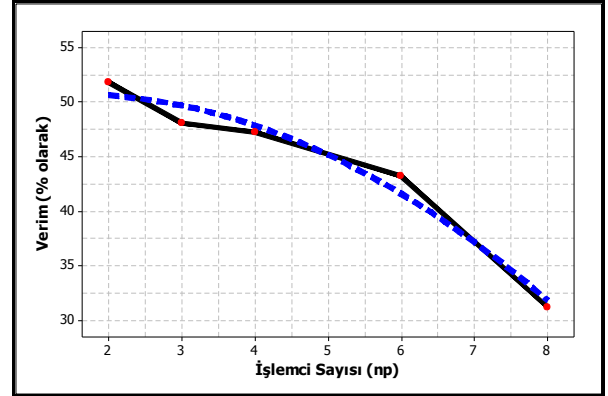
Gerçek hayatın karmaşıklığından ve insanların algılama kapasitesinin sınırlı olmasından dolayı, kesin olarak kavrayamadığımız çeşitli durumlar vardır ki bunlar ancak sözel olarak ifade edilebilir ve bu durum belirsizlik oluşturur. Böyle durumlarda uygun bir şekilde karar verilmesi oldukça zordur ve iyi bir seçme yeteneği gerektirir. Karar probleminde uzman sistemler, bulanık mantık, yapay sinir ağları gibi YZ teknikleri kullanılarak uzman bilgisi, eksik ve belirsiz bilgi ve durumlar modellenilebilir. Bulanık kümeler, karar vericiye daha geniş bir hareket alanı sağlayarak, uygulamaların gerçek dünyayı yansıtmaya becerisine büyük katkı sağlamaktadır. Bu bakış açısıyla, ele alınan problemde yer alan çok sayıda ve yoğun hesaplama gerektiren BAHS işlemleri için paralel hesaplama yöntemi geliştirilerek sistemin daha hızlı çalışması sağlanmış, verim ve başarıyı artırılmıştır. TOPSIS, BAHS'den gelen sonuçları birleştirdiği ve tek bir işlem yaptığı için paralelleştirmeye uygun değildir.

Geliştirilen paralel hesaplamalı karar verme algoritması homojen ve heterojen yapıdaki bilgisayarlar için ayrı ayrı denenmiş ve BAHS hesaplamalarının paralelleştirme için uygun olduğu görülmüştür. Amdahl Kanunu'nun doğal bir sonucu olarak yapılan deneyler sonucunda, sistemin başarı oranının işlemci sayısı arttırıldıkça arttığı fakat bir noktadan sonra işlemci artımının başarıyı etkilemediği görülmüştür. Bu durum belirli bir işlemci sayısında doyum noktasına ulaşıldığını göstermektedir. Buna ait grafiklerden görülebileceği gibi, tepe başarımına erişildiği noktada en verimli hesaplamaların da geliştirilen algoritma tarafından gerçekleştirildiği gözlemlenmiştir. Homojen sistemin, heterojen sisteme göre avantajı, hızlanma ve verim değerlerinin daha yüksek olarak elde edilmiş olması haricinde hesaplama karmaşıklığının değişmemesine rağmen ölçeklenebilirliğin önemini göstermesidir. Heterojen sistemde daha hızlı veya daha yavaş işlemcinin sisteme dahil edilmesi veya çıkarılması, toplam başarıyı önemli ölçüde etkilememektedir. Bu yüzden heterojen sistemde ölçeklenebilirlik bir sorun teşkil edebilecekken, homojen sistemde işlemciler aynı hızlara sahip oldukları için başarı, ölçekli olarak azalır arttırılabilecek yapıdadır.

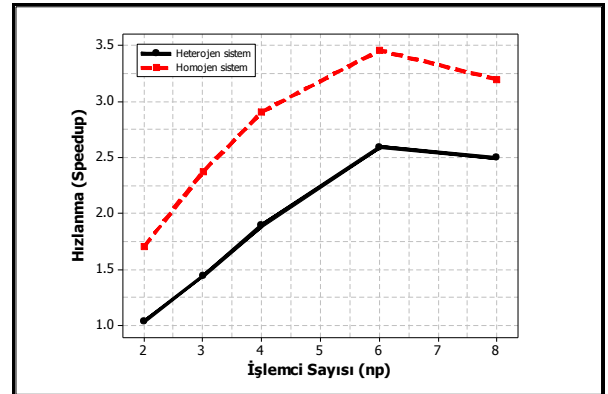
Bundan sonraki çalışmalarda, çok kişiden oluşan grup karar verme sırasında paralel hesaplamaların uygulanabilirliği, karmaşık problemler için uzun hesaplamalar gerektiren ÇKKV yöntemleri ve YZ tekniklerinin melez kullanımlarının paralel olarak gerçekleştirimleri araştırılacaktır.



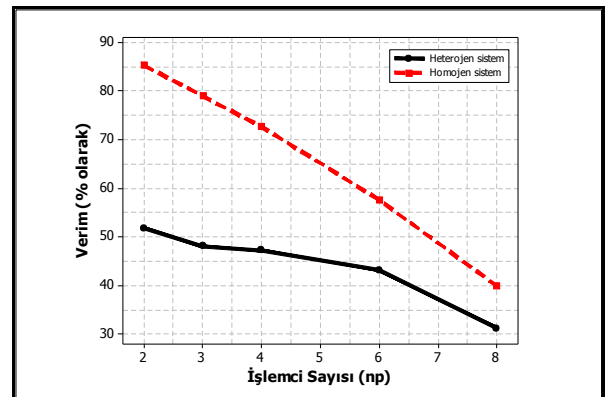
Şekil 6: Heterojen bilgisayarlar ile elde edilen hızlanma grafiği.



Şekil 7: Heterojen bilgisayarlar ile elde edilen verim grafiği.



Şekil 8: Homojen ve heterojen sistemlerin hızlanma açısından karşılaştırması.



Şekil 9: Homojen ve heterojen sistemlerin verim açısından karşılaştırması.

6 Kaynaklar

- [1] Flynn, M., "Some computer organizations and their effectiveness", *IEEE Transactions on Computers*, 21 (9), 948-960, 1972.
- [2] Grama A., Gupta A., Karypis G. ve Kumar V., *Introduction to Parallel Computing*, Addison Wesley Publishing, Newyork, A.B.D., 2003.
- [3] Gergel, V.P. ve Strongin, R.G., "Parallel computing for globally optimal decision making on cluster systems", *Future Generation Computer Systems*, 21 (5), 673-678, 2005.
- [4] Rahimi, S., Gandy, L. ve Mogharreban, N., "A web-based high-performance multicriteria decision support system for medical diagnosis", *International Journal of Intelligent Systems*, 22 (10), 1083-1099, 2007.
- [5] Wuppalapati, S., Belegundu, A.D., Aziz, A. ve Agarwala, V., "Multicriteria decision making with parallel clusters in structural topology optimization", *Advances in Engineering Software*, 39 (5), 416-421, 2008.
- [6] Yamamoto, Y., Nakano, J. ve Fujiwara, T., "Parallel computing in the statistical system Jasp", *Computational Statistics*, 25 (2), 291-298, 2009.
- [7] Zadeh, L.A., "Fuzzy sets", *Information Control*, 8, 338-353, 1965.
- [8] Kahraman, C., Cebeci, U. ve Ruan, D., "Multi-attribute comparison of catering service companies using fuzzy AHP: the case of Turkey", *International Journal of Production Economics*, 87, 171-184, 2004.
- [9] Ertuğrul, İ. ve Karakaşoğlu, N., "Performance evaluation of Turkish cement firms with fuzzy analytic hierarchy process and TOPSIS methods", *Expert Systems with Applications*, 36 (1), 702-715, 2009.
- [10] Chen, G. ve Pham, T.T., *Introduction to fuzzy sets, fuzzy logic and fuzzy control systems*, CRC Press, New York, A.B.D., 2001.
- [11] Lin, H.Y., Hsu P.Y. ve Sheen, G.J., "A Fuzzy-Based Decision-Making Procedure for Data Warehouse System Selection", *Expert Systems with Applications*, 32 (3), 939-953, 2007.
- [12] Bellman, R.E. ve Zadeh, L.A., "Decision-making in a fuzzy environment", *Management Science*, 17 (4), 141-164, 1970.
- [13] Saaty, T.L., *The analytic hierarchy process*, McGraw-Hill, A.B.D., 1980.
- [14] Chang, D.Y., *Extent Analysis and Synthetic Decision Optimization Techniques and Applications*, World Scientific, Singapore, 1992.
- [15] Chang, D.Y., "Applications of the extent analysis method on fuzzy AHP", *European Journal of Operational Research*, 95, 649-655, 1996.
- [16] Hwang, C.L. ve Yoon, K., *Multiple attributes decision making methods and applications*, Springer-Verlag, New York, A.B.D., 1981.
- [17] Benitez, J.M., Martin, J.C. ve Roman, C., "Using fuzzy number for measuring quality of service in the hotel industry", *Tourism Management*, 28 (2), 544-555, 2007.
- [18] Ballı S. ve Korukoğlu, S., "Operating System Selection Using Fuzzy AHP and TOPSIS Methods", *Mathematical and Computational Applications*, 14 (2), 119-130, 2009.
- [19] Ballı S., Melez Zeki Karar Destek Sistemlerinin Tasarımı ve Gerçekleştirimi, Doktora Tezi, Ege Üniversitesi, İzmir, 2010.
- [20] Karasulu, B., Ballı, S., Korukoğlu, S. ve Uğur, A., "Kutup Dengeleme Problemi İçin Yüksek Başarılı Bir Optimizasyon Tekniği", *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 14 (2), 175-183, 2008.
- [21] Amdahl, G.M., "Validity of single-processor approach to achieving large-scale computing capability", *Proceedings of AFIPS Conference*, Reston, VA., 1967, 483-485.
- [22] El-Rewini H. ve Abd-El-Barr M., *Advanced Computer Architecture and Parallel Processing*, JohnWiley and Sons, New York, A.B.D., 2005.