

**BULANIK OPERASYON ZAMANLI GELENEKSEL MONTAJ HATTI
DENGELEME PROBLEMİ**

Murat Mustafa YILMAZ

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MAYIS 2006

ANKARA

Murat Mustafa YILMAZ tarafından hazırlanan BULANIK OPERASYON ZAMANLI GELENEKSEL MONTAJ HATTI Dengeleme Problemi adlı bu tezin yüksek lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Orhan TRKBEY

Tez Yneticisi

Bu alıřma, jrimiz tarafından Endstri Mhendislięi Anabilim Dalında yüksek lisans tezi olarak kabul edilmiřtir.

Başkan: : _____

ye : _____

ye : _____

ye : _____

ye : _____

Tarih :/...../.....

Bu tez, Gazi niversitesi Fen Bilimleri Enstits tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Murat Mustafa YILMAZ

**BULANIK OPERASYON ZAMANLI GELENEKSEL MONTAJ HATTI
DENGELEME PROBLEMİ**

(Yüksek Lisans Tezi)

Murat Mustafa YILMAZ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Mayıs 2006

ÖZET

Bu tezde, bulanık operasyon zamanlı geleneksel montaj hattı dengeleme probleminin çözümü için bir algoritma geliştirilmiştir. Montaj hattını dengelemek için, bu algoritma; deterministik operasyon sürelerini değil, uzman tarafından belirlenen bulanık operasyon sürelerini kullanmaktadır. Geliştirilen çözüm algoritması Java programlama dili ile kodlanmış ve MAN Türkiye Akyurt kablo fabrikasında test edilerek, hattın dengelenmesi sağlanmıştır.

Bilim Kodu : 906.1.141
Anahtar Kelimeler : Montaj hattı dengeleme, Bulanık küme
Sayfa Adedi : 95
Tez Yöneticisi : Prof. Dr.Orhan TÜRKBEY

**TRADITIONAL ASSEMBLY LINE BALANCING PROBLEM
WITH FUZZY OPERATION TIME**

(M. Sc. Thesis)

Murat Mustafa YILMAZ

**GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

May 2006

ABSTRACT

In this thesis, an algorithm has been developed to solve traditional assembly line balancing problem with fuzzy operation time. To balance assembly line, the algorithm uses fuzzy operation time determined by an expert, not deterministic operation time. The developed solution algorithm has been coded with Java programming language and the line balancing has been provided being tested in MAN Turkey Akyurt cable factory.

Science Code	: 906.1.141
Key Words	: Assembly line balancing, Fuzzy set
Page Number	: 95
Adviser	: Prof. Dr.Orhan TÜRKBEY

TEŞEKKÜR

Bu tezin ortaya çıkmasında yardım ve destekleri ile beni yönlendiren başta değerli hocam Prof. Dr. Orhan TÜRKBEY'e, göstermiş olduğu yakın ilgiden dolayı, Araştırma Görevlisi Selçuk Kürşat İŞLEYEN'e, değerli vaktini benimle paylaşan Bilgisayar Mühendisi İbadullah Emre ŞAHİN'e, ayrıca MAN Türkiye Akyurt fabrikasına, Endüstri Mühendisi Yenal GÜNDÜZ'e ve çalışma arkadaşlarına, bana her zaman destek olan aileme sonsuz teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ	1
2. BULANIK MANTIK.....	3
2.1 Bulanık Mantık Kavramı.....	3
2.2 Bulanık Mantığın Tarihçesi.....	5
2.3 Bulanık Küme Teorisi	6
2.3.1 Sözel değişkenler	7
2.3.2 Bulanık küme	7
2.3.3 Bulanık sayılar teorisi	11
2.4 Bulanık Mantığın Uygulama Alanları.....	14
2.5 Bulanık Mantığın Avantajları ve Dezavantajları	15
3. MONTAJ HATTI ÜRETİMİ.....	17
3.1 Montaj Hattı Üretimi Temel Tanımları.....	17
3.2 Montaj Hattı Dengeleme Probleminin (MHDP) Sınıflandırılması	20
3.2.1 Ürün sayısına göre sınıflandırma	22

Sayfa

3.2.2 Taşıma sistemlerine göre sınıflandırma	23
3.2.3 İşlem zamanlarının değişkenliğine göre sınıflandırma	24
3.3 Montaj Hatlarının Yerleşimi	25
3.4 Geleneksel Montaj Hatları ve Dengeleme	27
3.5 Montaj Hatları Literatür İncelemesi.....	28
4. GELENEKSEL MONTAJ HATTI DENGELEME PROBLEMİNE BULANIK MANTIK YAKLAŞIMI	31
4.1 Geleneksel Montaj Hattı (GMHDP – 1) Modeli.....	32
4.2 Bulanık Operasyon Zamanlı Geleneksel Montaj Hattı (GMHDP – 1) Dengeleme Modeli	33
4.3 Bulanık Operasyon Zamanlı Geleneksel Montaj Hattı (GMHDP – 1) Dengeleme Modeli Çözüm Algoritması	36
5. UYGULAMA	46
6. SONUÇ VE ÖNERİLER.....	56
KAYNAKLAR	58
EKLER.....	62
EK-1 Öncelik diyagramı	63
EK-2 Java program kodları	64
EK-3 Atama sonuçları.....	83
ÖZGEÇMİŞ	95

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. A_2 üyelik fonksiyonunun D_2 kesikli yaklaşımı	10
Çizelge 2.2. BM'nin uygulama alanları	15
Çizelge 3.1. 9 Operasyonlu bir öncelik matrisi	20
Çizelge 4.1. 4 Operasyon için atanabilirlik dereceleri	40
Çizelge 4.2. 1 İstasyon için ikilik sistemde işaretlemeler	40
Çizelge 4.3. 1 İstasyon için benzersiz alternatifler	41
Çizelge 4.4. 1 İstasyon alternatifleri için α kesmeleri toplamı	41
Çizelge 4.5. 2 İstasyon için ikilik sistemde işaretlemeler	41
Çizelge 4.6. 2 İstasyon için benzersiz alternatifler	42
Çizelge 4.7. 2 İstasyon alternatifleri için α kesmeleri toplamı	42
Çizelge 4.8. 3 İstasyon için ikilik sistemde işaretlemeler	42
Çizelge 4.9. 3 İstasyon için benzersiz alternatifler	43
Çizelge 4.10. 3 İstasyon alternatifleri için α kesmeleri toplamı	43
Çizelge 4.11. En iyi atama “12-3-4” için α değerleri	44
Çizelge 4.12. α Değerlerine göre operasyon süreleri	44
Çizelge 5.1. Gösterge panosu tesisatı operasyon tanımları ve bulanık sınır süreler	47
Çizelge 5.2. Gösterge panosu tesisatı üyelik fonksiyonlarının α kesme fonksiyonları	51
Çizelge 5.3. Minimum istasyonlu hat etkinliği en yüksek atama sonuçları	54

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. 1'e Yaklaşan sayıların üyelik fonksiyonu	8
Şekil 2.2. Genç, orta yaşlı ve yaşlı kavramlarını temsil eden üyelik fonksiyonları ve A_2 'nin kesikli yaklaşımları D_2	10
Şekil 2.3. Üçgensel bulanık sayı ($A=(a_1, a_2, a_3)$)	13
Şekil 2.4. Yamuksal bulanık sayı ($A=(a_1, a_2, a_3, a_4)$)	14
Şekil 3.1. 9 Operasyonlu öncelik diyagramı	19
Şekil 3.2. MHDP sınıflandırılması	21
Şekil 3.3. Tekli ve çoklu ürünler için montaj hatları	22
Şekil 3.4. Gecikmesiz montaj hattı yerleşimi.....	23
Şekil 3.5. Gecikmeli montaj hattı yerleşimi	23
Şekil 3.6. Düz hat	26
Şekil 3.7. Dairesel hat	26
Şekil 3.8. Zig-zag hat	26
Şekil 3.9. U - şekilli hat	27
Şekil 3.10. Değişik açılı hat	27
Şekil 4.1. Operasyon zamanı üyelik fonksiyonu	34
Şekil 4.2. 4 Operasyonlu öncelik diyagramı	38
Şekil 4.3. 4 Operasyon için üyelik fonksiyonları	39
Şekil 4.4. 4 Operasyon için dengelenmiş düz hat	44
Şekil 5.1. Gösterge panosu tesisatı üyelik fonksiyon grafikleri	48
Şekil 5.2. Çevrim zamanı txt dosyası	52
Şekil 5.3. Operasyonlara ait α kesme fonksiyonları txt dosyası	52

Şekil	Sayfa
Şekil 5.4. Operasyonlara ait öncüllük ilişkileri txt dosyası.....	53
Şekil 5.5. Atama sonuçları htm dosyası.....	54

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$A(x)$	x kümesi elemanlarına ait bulanık A kümesi
C	Çevrim zamanı
E	Hattın etkinliği
i	Operasyon
j	İstasyon
m	İstasyon sayısı
μt_i	t kümesindeki i. elemanın üyelik derecesi
n	Operasyon sayısı
SI_i	i görevinin istasyon aralığı
S_k	k istasyonunun zamanı
T	Toplam iş zamanı
$\tilde{t}_i$	i. operasyonun bulanık işlem zamanı

Kısaltmalar	Açıklama
BM	Bulanık Mantık
GMHD	Geleneksel Montaj Hattı Dengeleme
GMHDP	Geleneksel Montaj Hattı Dengeleme Problemi
MHDP	Montaj Hattı Dengeleme Problemi

1. GİRİŞ

Artan rekabet koşulları altında, karı daha yüksek veya aynı seviyede tutabilmenin şartı maliyetlerin azaltılmasıdır. Üretim maliyetlerinin düşürülmesi için daha az girdi maliyetleri ile daha çok ürün üretme isteği, işletmeleri atelye tarzı yerleşimden montaj hattı yerleşimine geçmeyi zorunlu kılmıştır. Montaj hattı yerleşimleri ile ürünler birbirlerine bağlı iş istasyonlarında üretilmektedir. Bu birbirine bağlı istasyonların herbirinde, daha önceden tanımlanan ve ürünün üretilmesi için gerekli olan operasyonlar gerçekleştirilir. Bu operasyonlar, montaj hattındaki istasyonlara öyle paylaştırılmalıdır ki, hiç bir istasyon diğerini beklemeden ilgili operasyonları yerine getirsin ve ürün tamamlanmış olarak hattan çıkartılabilsin. İşte bu istek, montaj hatlarının dengelenme problemlerini ortaya çıkarmıştır.

Montaj hattı yerleşimlerinde genelde düz, yani Geleneksel Montaj Hattı (GMH) yerleşimi kullanılmaktadır. Bu hat yerleşiminde istasyonlar birbiri ardınca düz olarak yerleştirilir. Montaj Hattı Dengeleme Problemleri (MHDP) ile ilgili yapılan çalışmaların çoğu, GMH yerleşimi üzerinedir.

MHDP'lerin çözümünde bazı kriterlere dikkat etmek gereklidir. Bunlar arasında, yerine getirilecek operasyonların birbirleri arasında belirli ilişkilerin olması, her operasyonun belirli bir süresinin olması ve bu operasyonların, istasyon içerisinde belirli bir sürede tamamlanması yer alabilir.

Montaj hatlarında önemli kriterlerden birisi, her bir operasyonun tamamlanma süresidir. Değişen çevre koşulları altında emek yoğun çalışılan işlerde operasyon tamamlanma süresini tespit etmek oldukça zor olabilmektedir. Psikolojik olgular, çevresel faktörler ve diğer insana bağlı etkenler operasyon sürelerinin değişkenlik göstermesine sebep olmaktadır. Bu nedenle montaj hattı problemlerinde deterministik olarak tek bir süre almak etkinliği yüksek hat yerleşimini gözden kaçırabilmek anlamına gelebilir. Bu nedenle MHDP'lerinde bu tarz, belirsiz operasyon süreleri ile karşılaşılan yerlerde Bulanık Mantık (BM) kavramının kullanılması avantaj sağlayabilir. BM kavramı ile, net olarak belirleyemediğimiz

operasyon sürelerini bulanık olarak belirleyerek, etkin bir montaj hattının yerleşiminin sağlanabileceği düşünülmektedir.

Bu tez çalışması kapsamında; bulanık operasyon zamanlı, tek modelli Geleneksel Montaj Hattı Dengeleme (GMHD) problemi için bir çözüm algoritması önerilmiştir. Algoritma, Java programlama dili ile kodlanmış ve bir uygulama üzerinde test edilmiştir. Sezgisel algoritmanın çözümü için hazırlanan Java programı, 3Ghz ve 1024 MB RAM donanıma sahip bir bilgisayarda çalıştırılmış ve alternatifler yaklaşık 10 saat 20 dk da üretilmiştir.

Çalışmanın ikinci bölümünde; BM kavramı açıklanmış ve bu kavram üzerindeki temel ilkelerden bahsedilmiştir.

Üçüncü bölümde; montaj hattı kavramı detaylandırılmıştır. Montaj hattı ile ilgili temel bilgiler açıklanmış, geleneksel montaj hattı incelenmiş ve bu alanda yapılmış olan çalışmalara değinilmiştir.

Dördüncü bölümde; GMHDP'ye BM yaklaşımı üzerinde durulmuştur. Operasyon sürelerinin bulanık olarak belirlendiği minumum istasyon sayısına sahip atamanın yapıldığı, model ve çözüm algoritması anlatılmıştır.

Beşinci bölümde; çözüm algoritmasının Java programlama dili ile kodlandığı program, MAN Türkiye A.Ş. Akyurt kablo fabrikasındaki uygulama ile test edilerek, minumum istasyon sayısına sahip ve hat etkinliği en yüksek olan alternatifle hat dengelemesi yapılmaya çalışılmıştır.

Sonuç ve öneriler bölümünde ise; elde edilen sonuçlar, ileride yapılabilecek çalışmalar ve öneriler ile çalışma sonlandırılmıştır.

2. BULANIK MANTIK

2.1 Bulanık Mantık Kavramı

19. yüzyılda; bilim ve matematik alanında, çok yönlü yaşanan değişimlerden birisi de “belirsizlik” kavramı üzerinde olmuştur. Bilinen klasik mantık sistemleri ile sadece belirli koşullarda oluşan, kesin doğruluk değerleri, “doğru” yada “yanlış” kavramlarına sahip önermeler oluşturulabilmekte iken; uzun-kısa, yaşlı-genç gibi belirsizlik belirten kavramlar için ise ‘bulanık’ doğruluk kavramı kullanılmaktadır. “Ayşe güzel bir kızdır”, “Ali uzun boylu bir gençtir”, “100, 1’den çok daha büyük bir sayıdır”. Bunlar, klasik mantık sistemleriyle doğruluğundan söz edilebilmesi güç cümlelerdir. Çünkü “güzel”, “uzun”, “büyük”, ve hatta “çok daha”, açık bir şekilde tanımlanmamış, belirsizlik içeren sözcüklerdir. Bu sözcükler için doğruluk değerleri verebilmek amacıyla bulanık doğruluk kavramı kullanılır. Bulanık doğruluk kavramı; sıradan doğruluk kavramıyla benzerlikler gösterir, fakat daha geneldir, ve uygulama alanı daha geniştir, belirsizliğin, doğruluk ölçütünün keskin bir şekilde tanımlanmamasından kaynaklanan durumlardaki problemlerle uğraşmak için genel bir yol sağlar.

Mantık, eski çağlardan günümüze kadar bir çok gelişme göstermiştir. Gelişimi hep zamanın gereksinimine göre değişmiştir. BM, onun gelişimindeki son aşamalardan birisidir.

BM, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematik düzen olarak tanımlanabilir.

BM, adından anlaşılacağı gibi mantık kurallarının esnek ve bulanık bir şekilde uygulanmasıdır. Klasik mantıkta, “doğru” ve “yanlış” ifadeleri vardır, oysa BM’de, ikisinin arasında bir yerde olan önermeler ve ifadelere izin verilebilir ki, gerçek hayata baktığımızda hemen hemen hiçbir şey kesinlikle doğru veya kesinlikle yanlış değildir. Gerçek hayatta önermeler genelde kısmen doğru veya belli bir olasılıkla

doğru şekilde değerlendirilir. Bulanık mantığa da zaten klasik mantığın gerçek dünya problemleri için yeterli olmadığı, durumlar dolayısıyla ihtiyaç duyulmuştur.

Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir.

BM ile matematik arasındaki temel fark, bilinen anlamda matematiğin sadece aşırı uç değerlerine izin vermesidir. Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek işte bu yüzden zordur, çünkü veriler tam olmalıdır. BM kişiyi bu zorunluluktan kurtarır ve daha niteliksel bir tanımlama olanağı sağlar. Bir kişi için 190 cm boyunda demektense sadece uzun boylu demek bir çok uygulama için yeterli bir veridir. Böylece azımsanamayacak ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir.

BM'nin sistemi şu şekilde işler: Bir ifade tamamen yanlış ise klasik mantıkta olduğu gibi 0 değerindedir, yok eğer tamamen doğru ise 1 değerindedir. Bunların dışında tüm ifadeler 0 dan büyük, 1 den küçük reel değerler alırlar. Yani değeri 0.25 olan bir ifadenin anlamı %25 doğru %75 yanlış demektir.

Peki sayılar bulanıkmıdır? Önceleri bu sorunun yanıtı değil şeklindeydi. Sayıların sınırlarının belirli olduğu düşünülüyordu. Örneğin, tek sayı, çift sayı gibi. Ancak örnek olarak 0 sayısını düşündüğümüzde bu sayının ait olduğu bir küme vardır ve bu küme sıfır kümesidir. Diğer sayılar ise bu kümenin dışındadır. Peki sıfıra yaklaşan yada sıfıra yakın sayılar hangi kümededir? İşte bu sayılarda; büyük sayılar, ufak sayılar, orta büyüklükteki sayılar gibi bulanıktır [1].

BM'nin sağladığı en büyük fayda, “insana özgü tecrübe ile öğrenme” olayının kolayca modellenebilmesi ve belirsiz kavramlarının bile matematiksel olarak ifade

edilebilmesine olanak tanınmasıdır. Bu nedenle özellikle doğrusal olmayan sistemlere yaklaşım yapabilmek için uygundur.

Geliştirilen son teoremler BM'nin ilke olarak, ister mühendislik, ister fizik, ister biyoloji ya da ekonomi olsun, her türlü konuda sürekli sistemleri modellemek üzere kullanılabileceğini göstermektedir. Çoğu alanda, BM'li sağduyu modellerinin standart matematik modellerinden daha yararlı ya da kesin sonuçlar verdiği görülmektedir. BM'nin; insan düşünme tarzına yakın olması, uygulandığının matematiksel modele ihtiyaç duymaması, yazılımın basit olması dolayısıyla ucuza mal olması gibi avantajları vardır. Bu avantajların yanı sıra; uygulamada kullanılan kuralların oluşturulmasının uzmana bağlılığı, üyelik fonksiyonlarının deneme - yanılma yolu ile bulunmasından dolayı uzun zaman alabilmesi, kararlılık analizinin yapılmasının zorluğu gibi dezavantajları bulunmaktadır.

2.2 Bulanık Mantığın Tarihçesi

BM kuramı, ilk kez. 1965 yılında Berkeley'de Californiya Üniversitesi öğretim üyelerinden. Prof. Lotfi A. Zadeh tarafından ortaya atılmış ve hızla gelişerek, modern denetim alanında bir çok bilim adamının ilgisini çeken, araştırmaya açık yeni bir dal olmuştur. Zadeh, BM teorisinin bağımsız ve tam bir teori olmaktan çok, bulanıklaştırma yönteminin herhangi bir teorisin ayrık formdan, sürekli forma dönüştürülmek suretiyle genelleştirilmesi için kullanılan bir metodoloji olarak ele alınmasını istemektedir. Londra Üniversitesinden Prof. Dr. Mamdani, kuramı bir buhar türbinin hızının denetlenmesine uygulamayı düşünmüş ve bu amaçla, insanın davranışlarını simule eden; “Eğer türbin hızı çok hızlı artıyorsa ve basınç da çok düşükse, buhar vanasını biraz aç” türünden kurallardan oluşan bir uzman sistem geliştirmiştir. BM kuramının ilk önemli endüstriyel uygulaması çimento sanayisinde olmuştur. Bir Danimarka firması BM kontrolcüsü kullanarak çimento üretimindeki oksijen ve sıcaklık dengeleri üzerine çok başarılı sonuçlar veren bir uzman sistem geliştirmiştir. Bu veya benzeri sistemler bugün bile Japonya ve Amerika'da dahil olmak üzere bir çok ülkede kullanılmaktadır.

Kronolojik sıra içerisinde bundan sonraki en önemli aşama Japonya’ da 1987 yılında görülmüştür. Hitachi firması, tren ile seyahat eden yolcuların bir yere tutunmalarına gerek kalmadan yolculuklarını gerçekleştirebildikleri, trenin hızlanma ve yavaşlama durumunu kontrol eden bir BM kontrolcüsünün bulunduğu bir sistem geliştirmişlerdir.

Yukarıda açıklanan başarılı uygulamalardan sonra bulanık denetim konusundaki çalışmalar, yeni bir ivme kazanmış ve endüstriyel uygulama alanları hızla artmıştır. Çalışmaların uluslararası alanda koordinasyonu amacı ile Japonya’da 1989 yılında LIFE (Laboratory for International Fuzzy Engineering) adlı bir laboratuvar kurulmuştur. Bu laboratuvarlarda yapılan araştırma çalışmalarına, aralarında Hitachi, Toshiba, Omron, Matsushita gibi ünlü Japon firmalarının yanı sıra IBM, NCR ve Thomson gibi Japonya dışı firmaların da bulunduğu birçok firma katılmaktadır. LIFE’in yanında FLSI (Fuzzy Logic Systems Institute) adındaki diğer bir araştırma merkezi de Bulanık Mantığın Elektronik, Otomotiv ve Üretim teknolojisi alanında yeni yeni uygulamalar kazandırmaktadır [2].

BM denetleyiciler konusundaki kuramsal çalışmaların hala sürüyor olmasına rağmen, artık bu konu endüstride kendisine önemli bir yer edinmiş durumdadır. Uygulama alanları arasında çeşitli beyaz eşya, tren, asansör, trafik kontrolü ve otomobil sanayisi sayılabilir.

2.3 Bulanık Küme Teorisi

Bulanık küme teorisi; esas olarak insan düşünce ve algılarındaki belirsizlikle ve bu belirsizliğin sayısallaştırılması ile ilgilenmektedir. Yani tam ve kesin olmayan bilgiler ışığında insanların tutarlı ve doğru kararlar vermelerini sağlayan, diğer bir deyişle BM yardımıyla düşünme ve karar verme mekanizmalarının modellenmesi ile ilgilenmektedir [3].

2.3.1 Sözel değişkenler

BM’de önemli kavramlardan biriside sözel değişken kavramıdır. Sözel değişken “sıcak” veya “soğuk” gibi kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir sözel değişkenin değerleri bulanık kümeler ile ifade edilir. Örneğin oda sıcaklığı sözel değişken için “sıcak”, “soğuk” ve “çok sıcak” ifadelerini alabilir. Bu üç ifadenin her biri ayrı ayrı bulanık kümeler ile modellenir [4].

2.3.2 Bulanık küme

En genel anlamda bulanık küme; üye olandan, üye olmayana doğru birdenbire değilde kademeli olarak geçiş yapan nesnelerin bulunduğu kümedir [4].

X boş olmayan bir küme olsun. X’deki bir Bulanık A kümesi üyelik fonksiyonu

$$A: X \rightarrow [0,1] \quad (2.1)$$

ile ifade edilmektedir. $\forall x \in X$ için; x’in üyelik derecesi $A(x)$ olarak belirtilir.

Çalışılan X evreni kesin ve sınırlı olduğu zaman A kümesi sembolik olarak aşağıdaki gibi gösterilir:

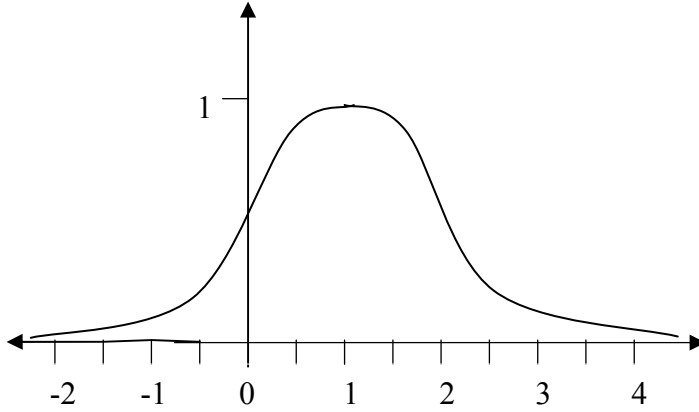
$$A = \{ \mu_A(x_1) / x_1 + \mu_A(x_2) / x_2 + \dots \} = \{ \sum_i \mu_A(x_i) / x_i \mid i = (1, \dots) \} \quad (2.2)$$

X evreni sürekli ve sınırsız ise A kümesi

$$A: \{ \int \mu_A(x) / x \} \quad (2.3)$$

şeklinde gösterilir. Bu gösterimdeki cebirsel semboller cebirsel anlamlarıyla kullanılmazlar. Örneğin “+” toplam anlamında değil teorik olarak birleşme anlamındadır [5].

Örneğin; “1’e yaklaşan” reel sayıların bulanık kümesinin üyelik fonksiyonunu aşağıdaki gibi tanımlanabilir:



Şekil 2.1. 1’e Yaklaşan sayıların üyelik fonksiyonu

Yukardaki önerme için uygun fonksiyonlardan biri çan eğrisidir:

$$\mu_{a,m}(x) = e^{-a(x-m)^2} \quad a>0, m \in \mathbb{R} \quad (2.4)$$

Bu örnek için $m=1$ ’dir. Eğer özel olarak “1’e yaklaşan doğal sayılar” için bir küme tanımlamak istersek, bunu aşağıdaki şekilde ifade edebiliriz

$$A = \{ 0,0/-2 + 0,3/-1 + 0,6/0 + 1,0/1 + 0,6/2 + 0,3/3 + 0,0/4 \}$$

Bulanık kümenin desteği: A X’in bir bulanık kümesi olsun. A’nın desteği, $\text{Supp}(A)$, X’in elemanları sıfır olmayan bir alt kümesidir [6].

$$\text{Supp}(A) : \{x \in X, A(x) > 0\} \quad (2.5)$$

Bulanık kümenin normalliği: Eğer herhangi bir $x \in X$ için ;

$$A(x)=1 \quad (2.6)$$

ise A bulanık kümesine normal denir [6].

Bulanık kümenin yüksekliği: A bulanık kümesinin en büyük üyelik derecesine o kümenin yüksekliği denir [6].

$$h(A) = \sup_{x \in X} A(x) \quad (2.7)$$

Bulanık kümenin α kesmesi: X de tanımlı bir A bulanık kümesi ve $\alpha \in [0,1]$ verilsin. α -kesmesi, ${}^\alpha A$, ve güçlü α -kesmesi, ${}^{\alpha+} A$, aşağıdaki gibi tanımlanmış kesme kümeleridir:

$${}^\alpha A = \{ x \mid A(x) \geq \alpha \} \quad (2.8)$$

$${}^{\alpha+} A = \{ x \mid A(x) > \alpha \} \quad (2.9)$$

α kesmelerini daha iyi anlamak için aşağıdaki örneği inceleyebiliriz:

Genç, orta yaşlı ve yaşlı insan kavramını temsil eden $[0,80]$ aralığında tanımlı üç bulanık küme göz önüne alalım: sırasıyla A_1 , A_2 ve A_3 .

$$A_1(x) = \begin{cases} 1 & x \leq 20 \\ (35-x)/15 & 20 < x \leq 35 \\ 0 & x \geq 35 \end{cases}$$

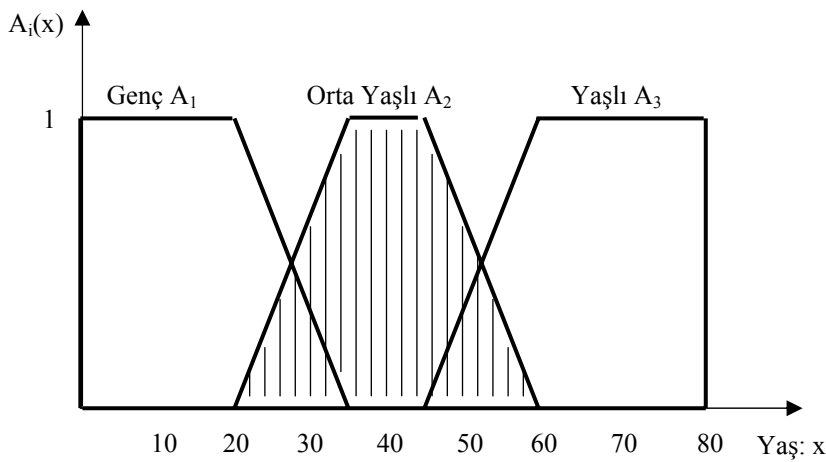
$$A_2(x) = \begin{cases} 0 & x \leq 20 \text{ veya } x \geq 60 \\ (x-20)/15 & 20 < x < 35 \\ (60-x)/15 & 45 < x < 60 \\ 1 & 35 \leq x \leq 45 \end{cases}$$

$$A_3(x) = \begin{cases} 1 & x \leq 20 \\ (35-x)/15 & 20 < x \leq 35 \\ 0 & x \geq 35 \end{cases}$$

A_2 fonksiyonunun mümkün bir kesikli yaklaşımı, D_2 , Şekil 2.2.'de ve daha açıklayıcı sayısal değerleri Çizelge 2.1.'de görülmektedir. Bu yaklaşımlar bulanık kümelerinin bilgisayar gösterimlerinde sıkça kullanılır.

Bulanık kümelerin en önemli kavramlarından biri α -kesmesi ve güçlü α -kesmesidir.

X 'e ait verilen bir bulanık A kümesi ve $\alpha \in [0,1]$ için ${}^\alpha A$, ${}^{\alpha+} A$ kesin kümelerdir [6].



Şekil 2.2. Genç, orta yaşlı ve yaşlı kavramlarını temsil eden üyelik fonksiyonları ve A_2 'nin kesikli yaklaşımları D_2

Çizelge 2.1. A_2 üyelik fonksiyonunun D_2 kesikli yaklaşımı

x	$D_2(x)$
$x \notin \{22, 24, \dots, 58\}$	0.0
$x \in \{22, 58\}$	0.13
$x \in \{24, 56\}$	0.27
$x \in \{26, 54\}$	0.40
$x \in \{28, 52\}$	0.53
$x \in \{30, 50\}$	0.67
$x \in \{32, 48\}$	0.80
$x \in \{34, 46\}$	0.93
$x \in \{36, 38, \dots, 44\}$	1.00

$${}^0 A_1 = {}^0 A_2 = {}^0 A_3 = [0, 80] = X$$

$${}^\alpha A_1 = [0.35 - 15\alpha]$$

$$\text{tüm } \alpha \in (0, 1]$$

$${}^\alpha A_2 = [15\alpha + 20, 60 - 15\alpha]$$

$$\text{tüm } \alpha \in (0, 1]$$

$${}^\alpha A_3 = [15\alpha + 45, 80]$$

$$\text{tüm } \alpha \in (0, 1]$$

için ;

$${}^{\alpha+}\mathbf{A}_1 = (0.35 - 15\alpha) \quad \text{tüm } \alpha \in (0,1]$$

$${}^{\alpha+}\mathbf{A}_2 = (15\alpha + 20, 60 - 15\alpha) \quad \text{tüm } \alpha \in (0,1]$$

$${}^{\alpha+}\mathbf{A}_3 = (15\alpha + 45, 80) \quad \text{tüm } \alpha \in (0,1]$$

için ;

$${}^{1+}\mathbf{A}_1 = {}^{1+}\mathbf{A}_2 = {}^{1+}\mathbf{A}_3 = \emptyset$$

Bulanık kümenin konveksliği: Bulanık kümelerin önemli özelliklerinden biriside onların konveksliğidir.

$x_1, x_2 \in R; \lambda \in [0,1]$ olmak üzere R üzerinde tanımlı bir A bulanık kümesi,

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \min[A(x_1), A(x_2)] \quad (2.10)$$

şartını sağlıyorsa konvektir [6].

2.3.3 Bulanık sayılar teorisi

Birçok alanda geniş kullanım alanı bulunan bulanık sayıların bazı temel özellikleri şu şekildedir.

Normal ve konveks bir bulanık kümenin, zayıf bir α -kesmesi kapalı bir küme ise, bulanık sayı olarak adlandırılır.

Bulanık sayılar, gerçel sayıların bir alt kümesini oluşturur. Bu sayıların kullanıldığı fonksiyonda, en büyük değer 1 olmaktadır. Bu değer dışındaki her x değeri için

elde edilen $\mu(x)$ değeri, azalan bir özellik taşır. Burada kullanılan $\mu(x)$, bir dışbükey fonksiyondur. Bu fonksiyon, en az bir x değeri için, $\mu(x) = 1$ değerini alır [7].

Üçgensel ve yamuksal bulanık sayılar: İlgilenilen bulanık sayılar kesikli özellik gösteriyorsa, daha önceki bölümlerde de anlatıldığı gibi tek tek üyelik derecelerine göre küme şeklinde gösterilebilirler. Ancak her zaman bulanık sayıların kümesi bu kadar dar olmayabilir. Bu durumda bulanık sayılar fonksiyonlar ile ifade edilebilir. Oldukça değişik bulanık sayı biçimleri bulunur. Bunlar arasında kullanımı en yaygın olan bulanık sayı biçimleri, üçgensel ve yamuksal bulanık sayılardır. Bu sayılar üzerinden α kesmeleri yöntemi kullanılarak kesin kümelerle geçiş yapılabilir ve bu sayede modelleme ve aritmetik işlemlerde büyük kolaylık sağlanır. Bu tezde de α kesmeleri yöntemiyle bulanık sayılardan kesin kümelerle geçiş yapılarak ilgili modelleme ve aritmetik işlemler gerçekleştirilecektir.

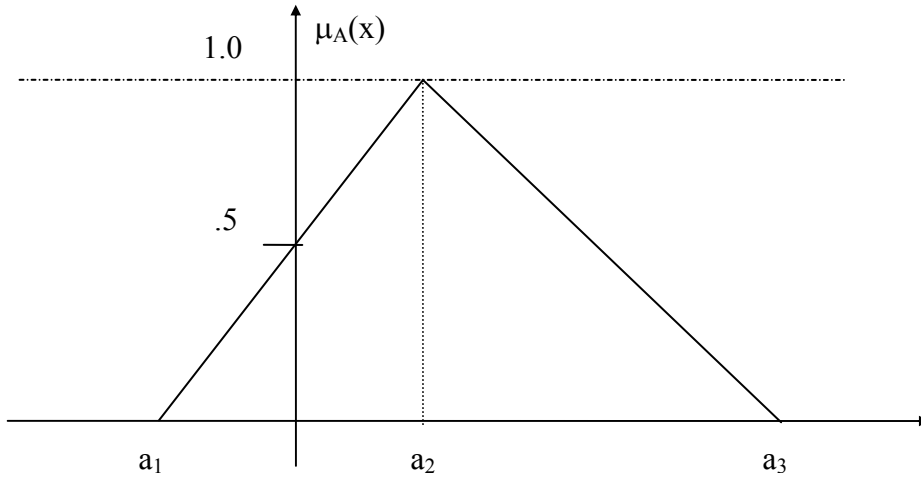
Üçgensel bulanık sayılar: Üçgensel bulanık sayılar, özellikle sistem modellemede çok sık kullanılmaktadır. Üçgensel bulanık sayılar, üç elemandan oluşan, (a_1, a_2, a_3) takımı şeklinde tanımlanabilir. Şekil 2.3.'de gösterilen üçgensel bulanık sayı için, üyelik fonksiyonu şu şekilde tanımlanabilir;

$$\mu_A(x) = \begin{cases} 0 & x < a_1 \text{ ise} \\ (x-a_1)/(a_2-a_1) & a_1 \leq x \leq a_2 \text{ ise} \\ (a_3-x)/(a_3-a_2) & a_2 \leq x \leq a_3 \text{ ise} \\ 0 & x > a_3 \text{ ise} \end{cases} \quad (2.11)$$

Alternatif olarak, α seviyesinde güven aralığı tanımı, üçgensel bulanık sayı olarak aşağıdaki şekilde nitelendirilebilir [2];

$$\forall \alpha \in [0,1] :$$

$$A_\alpha = [a_1(\alpha), a_3(\alpha)] = [(a_2-a_1)\alpha + a_1, -(a_3-a_2)\alpha + a_3] \quad (2.12)$$



Şekil 2.3. Üçgensel bulanık sayı ($A=(a_1, a_2, a_3)$)

Yamuksal bulanık sayılar: Bulanık sayıların diğer önemli bir bölümünü yamuksal bulanık sayılar oluşturmaktadır. Bu durumda, $\alpha = 1$ için bir nokta değil, Şekil 2.4.'de görüldüğü gibi, (a_2, a_3) aralığında tanımlı, bir doğru söz konusudur. Bu, bulanık sayıların özel bir sınıfı olan, yamuksal bulanık sayıları oluşturur.

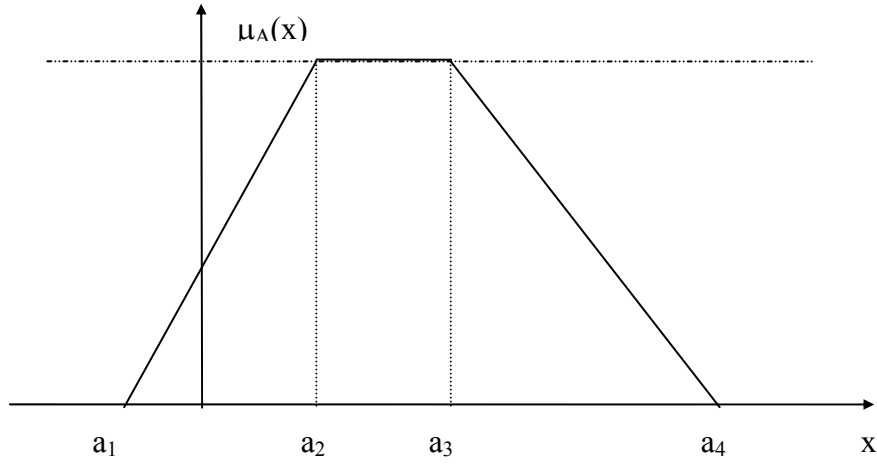
Yamuksal bulanık sayıların özellikleri, üçgensel bulanık sayılarınki ile benzerdir. Yamuksal bulanık sayı, tamamıyla $A = (a_1, a_2, a_3, a_4)$ şeklindeki bir dörtlüyle temsil edilebilir.

Yamuksal bulanık bir sayı, güven aralığı ile birlikte α -kesme düzeyinde nitelendirilebilir;

$$A_\alpha = (a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_4 \quad \forall \alpha \in [0,1] \quad (2.13)$$

olur. Yamuksal bir sayının üyelik fonksiyonu ise aşağıdaki gibi tanımlanabilir [2];

$$\mu_A(x) = \begin{cases} 0 & x < a_1 \text{ ise} \\ (x - a_1)/(a_2 - a_1) & a_1 \leq x \leq a_2 \text{ ise} \\ 1 & a_2 \leq x \leq a_3 \text{ ise} \\ (a_4 - x)/(a_4 - a_3) & a_3 \leq x \leq a_4 \text{ ise} \\ 0 & x > a_4 \text{ ise} \end{cases} \quad (2.14)$$



Şekil 2.4. Yamuksal bulanık sayı ($A=(a_1, a_2, a_3, a_4)$)

2.4 Bulanık Mantığın Uygulama Alanları

Bulanık mantığın ve bu mantık kurallarını kullanan bulanık küme teorisinin ilk olarak Lotfi A. Zadeh tarafından geliştirilerek 1965 yılında yayınlanmasından sonra ancak 1970’li yılların ikinci yarısından sonra kullanılmaya başlanmıştır. Zadeh’in kazandırdığı bu kavram, ilk olarak Danimarka’da bir şirketin geliştirdiği teknolojilerde kullanılmış olduğunu, daha sonrada Japonya’da bulanık mantık teknolojisinin benimsenmesiyle titreşimsiz video alıcısı, otomobil, tren gibi çeşitli ürünlerinde; Çin’de ise başlangıçta denizcilikte daha sonra savaş teknolojisinde jet uçakların üretiminde; günümüzde ise ABD’de dünyanın bir çok yerinde hızla genişleyerek farklı kullanım alanlarına yayılmış olduğunu görüyoruz. Günümüzde hemen her alanda BM uygulamalarına rastlamak mümkündür.

BM’nin uygulama alanlarından bazıları; otomatik kontrol sistemleri (robotik, otomasyon, akıllı denetim, izleme sistemleri, vb.), bilgi sistemleri (bilgi depolama ve yeniden çağırma, uzman sistemler, bilgi tabanlı sistemler, vb.), Optimizasyon (fonksiyon optimizasyonu, süzgeçleme, vb.).

Çizelge 2.2. BM'nin uygulama alanları

<i>Uygulama</i>	<i>Şirket</i>	<i>Bulanık mantığın rolü</i>
Araba Motoru	NOK/ Nissan	Benzin Pombası, oksijen miktarı, su sıcaklığı, devir sayısı, benzin miktarı, krank açısı, tekleme ve manifold basıncına göre benzin enjeksiyonunu ve patlamayı kontrol etme
Asansör Denetimi	Fujitec, Mitshubishi Elektrik, Toshiba	Taşınilan insan sayısına göre bekleme süresini azaltmak
Avuçiçi Bilgisayar	Sony	Elle yazılmış karakterleri tanımak
Bulaşık Makinesi	Matsushita	Bulaşıkların sayısına ve kirlilik miktarına göre yıkama ve parlatma stratejilerini belirlemek
Buzdolabı	Sharp	Kullanıma göre soğutma ve dondurma sürelerini belirlemek
Elektrikli süpürge	Hitachi, Matsushita, Toshiba	Toz miktarı ve zemin türüne göre motorun emme stratejisini tespit etmek
Fabrika Denetim	Omron	İş bölümü ve seri üretim stratejilerini belirlemek
Fren sistemi	Nissan	Arabanın hızlanmasıyla ortaya çıkan tehlikeli durumlarda frenleri kontrol etmek
Hisse senedi alım satımı	Yamaichi	Makro ve mikro-ekonomik verilere göre Japon hisse senedi piyasasını idare etmek
Kamera	Matsushita, Panasonic	Elin titremesinden kaynaklanan görüntü bozukluklarını gidermek ve oto-fokusu temin etmek
Klima	Hitachi, Matsushita, Mitsubishi, Sharp	İstenmeyen sıcaklık oksilasyonunu önlemek ve açma-kapamada daha az enerji sarf etmek

BM'nin uygulama alanı bulduğu ülkelerin başında Japonya gelmektedir. Japonya'da birçok şirket geliştirdikleri ürünler ve kendi organizasyonları içinde BM ilkelerini yoğun bir şekilde kullanmaktadırlar. Çizelge 2.2.'de BM'nin en çok kullanıldığı uygulamalar görülmektedir. Çizelden de anlaşılabileceği gibi BM, insan düşünce ve zekasının bulunduğu her alanda kendini göstermektedir [8].

2.5 Bulanık Mantığın Avantajları ve Dezavantajları

BM, klasik mantık yöntemlerinde karmaşıklığı daha basite indirgeyerek pratiğe uygun çözümler üretmede oldukça etkin bir araçtır. Bulanık mantığın sağladığı en büyük yararlardan biri “insana özgü tecrübe ile öğrenme” olayının kolaylıkla modellenebilmesidir. İnsanın kontrol stratejisini taklit ettiği için anlaşılması kolaydır. BM, tasarımın geliştirme sürecini azaltır. Çevresel değişiklikler, gürültüler, parametre değişiklikleri ve hatalarda sadece basit kural değişiklikleri yapılarak

sistem deęiřtirilebilir. Giriř ve ıkıřlar arasındaki iliřkileri tanımlayan kompleks matematiksel modellere veya farklı sistem yöntemlerine ihtiya duyulmaz. BM yaklařımında iřaretlerin bir n iřlemeye tabi tutulmaları ve geniř bir alana yayılmıř deęerlerin az sayıda yelik fonksiyonlarına indirgenmeleri, uygulamaların daha hızlı bir řekilde sonuca ulařmasını saęlar. Ayrıca BM ile yapılan uygulamaların yazılımlarındaki basitlikten dolayı maliyet aısından da büyük kazan saęlanır [9].

Bulanık mantıęın bu avantajlarının yanında bazı dezavantajları da vardır. BM uygulamalarında mutlaka kuralların uzman deneyimlerine dayanarak tanımlanması gerekir. yelik fonksiyonlarını ve BM kurallarını tanımlamak her zaman kolay deęildir. yelik fonksiyon deęiřkenlerinin belirlenmesinde kesin sonu veren belirli bir yöntem ve ęrenme yeteneęi yoktur. En uygun yöntem deneme yanılma yöntemidir, bu da ok uzun zaman alabilir. Sistemlerin kararlılık, gözlemlenebilirlik ve denetlenebilirlik analizlerinin yapılmasında ispatlanmış kesin bir yöntemin olmayıřı BM'nin temel sorunudur.

3. MONTAJ HATTI ÜRETİMİ

Montaj hattı; malzemelerin bir hat boyunca işgücü yardımıyla ya da otomatik olarak transfer edilmeleri ve parça üzerindeki işlemlerin de bir hat boyunca sıralı iş istasyonlarında yapılması olarak tanımlanabilir. Bir üretim montaj hattı, seri durumda iş istasyonlarından oluşur. Bu istasyonlar bir yada daha fazla makine ve işçiden oluşur.

Montaj hatlarının tasarımıındaki ana amaçlardan biri, her iş istasyonuna eşit miktarda iş dağıtımını yapabilmektir. Dengenin sağlanamadığı bir durumda bazı istasyonlarda diğerlerinden daha fazla iş yükü olacağı için, verimlilikte düşüşlerin olması ve bir takım kayıpların ortaya çıkması kaçınılmazdır.

Montaj hatları, günümüz endüstrisinde önemli bir yer tutan kitle üretim yöntemlerinden birisidir. Değişik tipteki montaj hatlarının kullanılmasının önemli amaçlarından birisi üretilecek olan ürüne olan kitle talebini karşılamaktır.

3.1 Montaj Hattı Üretimi Temel Tanımları

Montaj : Tamamlanmış bir ürün oluşturmak amacıyla değişik parçaların bir araya getirilip birleştirilme işlemidir. Parçalar komponent ve alt montaj parçaları olarak iki kısımda adlandırılabilir. Komponentler genel anlamda satın alma olarak tedarik edilen, belirli sayıda parçadan oluşan parça gruplarıdır. Alt montaj parçası ise üretimde ön montaj işlemi ile montaja hazır hale getirilen birleştirilmiş parça grubudur. Parçaların arasındaki ilişki ve parçaların akışı montaj şemalarıyla görselleştirilir.

Operasyon/Görev : Operasyon, bir montaj prosesinde toplam iş içeriğinin bir parçasıdır. Operasyonlar i indisi ile ifade edilir, operasyon sayısı toplamı ise N ile gösterilir. Operasyonu gerçekleştirmek için gerekli zaman, operasyon (görev) zamanı olarak adlandırılır ve i görevine ait operasyon zamanı t_i ile ifade edilir. Operasyonlar toplam işin bölünemez en küçük parçaları olarak düşünülürler.

İş İstasyonu : Montaj hattı üzerinde bir operasyonun yapıldığı alandır. Montaj hattı iş istasyonları genellikle bir kişi tarafından çalıştırılır. Birden fazla kişinin çalışması da mümkündür.

Çevrim Zamanı : Çevrim zamanı, montaj hattında, ürünün bir istasyonda kalabileceği en büyük süre veya bir iş istasyonundaki çalışanın o istasyonda yapılması gerekli işleri tamamlaması için gerekli süre olarak tanımlanabilir. Çevrim süresini seçmekteki ana düşünce, gerek duyulan üretim hızıdır. Çevrim zamanı C ile gösterilir. Çevrim zamanı aynı zamanda bir parçanın bir istasyondan diğerine geçiş süresi olarak ifade edilebilir.

İstasyon Zamanı : İstasyona gelen bir parça üzerinde o istasyonda yapılması gereken ilk operasyonun başlangıç anı ile son operasyonun bitiş anı arasındaki süre farkıdır. k istasyonuna ait zaman S_k ile gösterilir ve montaj hattındaki bir istasyon zamanı, çevrim zamanından küçük veya eşit olmalıdır.

$$Max t_i \leq Max S_k \leq C \quad (3.1)$$

Toplam İş Zamanı : Montaj hattı üzerinde bir ürünün montajı için gerekli olan sürelerin veya işi oluşturan tüm operasyonların standart sürelerinin toplamıdır. T ile gösterilir.

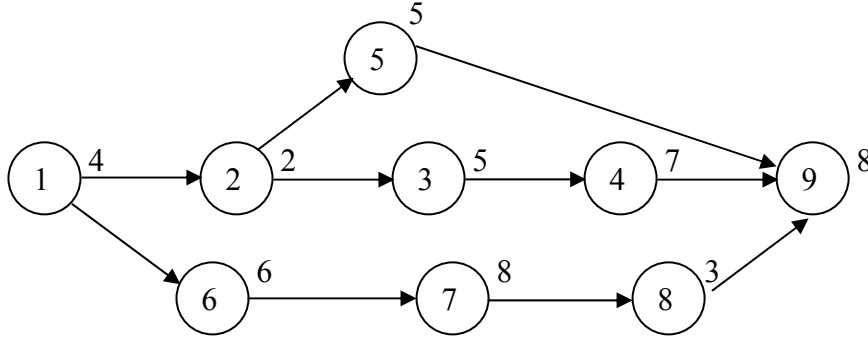
$$T = \sum t_i \quad (3.2)$$

İstasyon Gecikme Zamanı/Boş zaman : Çevrim süresi ile iş istasyonu zamanı arasındaki fark iş istasyonunun boş zamanını göstermektedir. İstasyondaki boş zaman $C-S_k$ olarak hesaplanır. Toplam boş zaman (TBZ) ise hat üzerindeki tüm istasyonların boş zamanlarının toplamıdır.

$$TBZ = \sum_{k=1}^K C - S_k \quad (3.3)$$

Öncelik Diyagramı : Montajın teknik özelliklerinden dolayı, bazı operasyonların zorunlu olarak birbirini izlemesi gerekir. Bu özelliklerin tümü öncelik ilişkileri adı altında toplanır. Bu ilişkiler genellikle bir grafik ile gösterilir. Bu grafik gösterim, hat dengeleme sistemlerinde öncelik ilişkilerinin belirtilmesinde çok yaygın olarak kullanılan öncelik diyagramıdır.

Bu diyagramı bir okla birbirine bağlanmış iki iş ögesinden okun çıktığı yönde bulunanın, okun ucunda bulunan iş ögesinden daha önce işleme alınacağını gösterir. İçindeki numaralar iş ögesi numaralarını, dairelerin sağ üstündeki sayılar ise iş ögesi sürelerini gösterir. Şekil 3.1.'de bir montaj hattına ilişkin öncelik diyagramı görülmektedir.



Şekil 3.1. 9 Operasyonlu öncelik diyagramı

Öncelik Matrisi : Öncelik diyagramının matris durumuna dönüştürülmüş şeklidir. Bu matriste aralarında doğrudan doğruya öncelik ilişkileri bulunan iş ögeleri için, matriste önde gelen öge numaralı satırla; izleyen öge numaralı sütunun kesiştiği noktaya 1, diğer noktalara 0 değeri konur. Çizelge 3.1.'de, Şekil 3.1.'de öncelik diyagramı verilen operasyonlar için hazırlanmış öncelik matrisi görülmektedir.

Çizelge 3.1. 9 Operasyonlu bir öncelik matrisi

	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0

Hat Dengesi : Hat dengeleme problemlerinin uygun bir çözümünü temsil eder. Hat dengesinden söz edebilmek için aşağıdaki noktaların sağlanması gerekmektedir:

- Bütün operasyonların tamamlanması gerekir.
- Bir operasyon birden fazla istasyona atanamaz.
- Herhangi bir istasyonun istasyon zamanı, çevrim zamanını aşamaz.
- Eğer öncelik diyagramında i operasyonu j operasyonundan önce geliyorsa, i operasyonu bir istasyona atanmadan j operasyonu atanamaz.

Hattın Etkinliği : Verimlilik ölçütü olarak kullanılır ve E ile ifade edilir. Hat etkinliği değeri ne kadar 1'e yakınsa hat o kadar etkindir [10]

$$E = \frac{\sum_{k=1}^K S_k}{K.C} \quad (3.4)$$

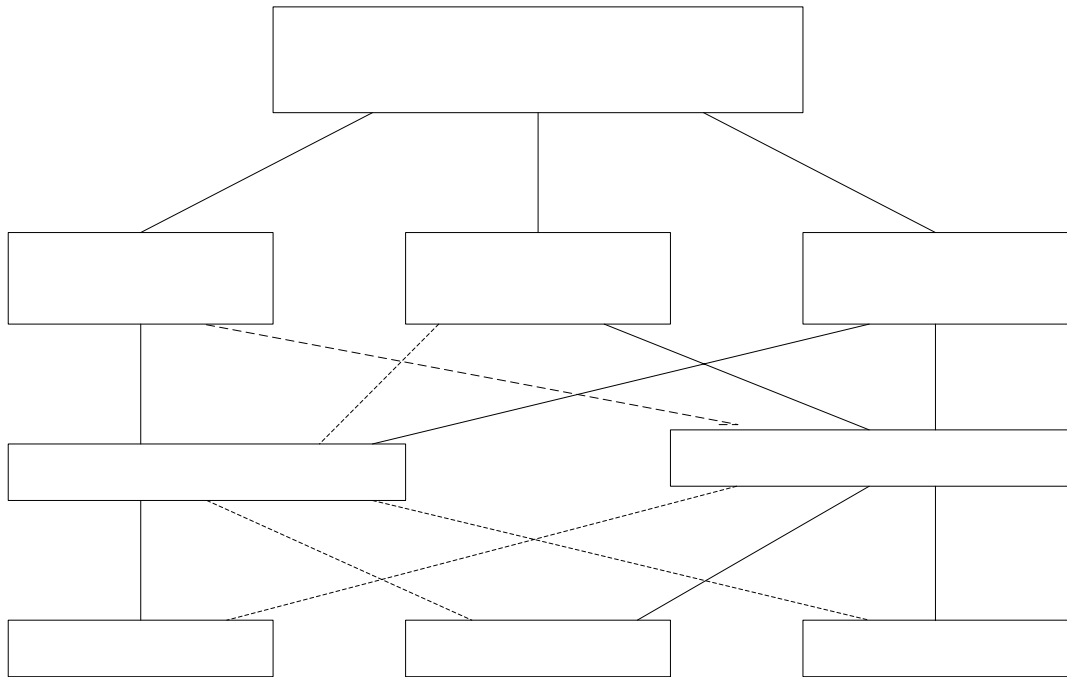
3.2 Montaj Hattı Dengeleme Probleminin (MHDP) Sınıflandırılması

Montaj hattı üretim sistemleri çeşitli endüstriyel alanlara hitap eder ve çok değişik sayıda ürünün üretildiği alanlarda fayda sağlar. Özellikle otomobil, motor,

televizyon, bilgisayar ve diğer elektrikli aletler gibi tüketim mallarının çok sayıda üretildiği sistemler için tasarlanırlar.

Montaj hatlarının tasarlanmasının genel amacı, bir ürünü üretbilmek için yapılması gerekli operasyonları, belirli kısıtlar altında en az maliyetle, planlanan ürün çıktısını alabilmektir. Temel olarak montaj hatları sınırlı sayıda görev kümesi ve her göreve ait bir işlem zamanı ve öncelik ilişkileri kümesinden oluşur. MHDP ise bu operasyonların çeşitli verimlilik ölçütlerine göre (örneğin; minimum istasyon, minimum çevrim zamanı vb.), sıralı istasyonlara atanmasıdır.

Literatürde bulunan MHDP'nin temel özellikleri ve ilişkileri Şekil 3.2.'de gösterilmiştir. İlgili şekilde sürekli çizgiler çok rastlanan ilişkileri göstermekte, kesikli çizgiler ise nadir olarak karşılaşılan ilişkileri göstermektedir.



Şekil 3.2. MHDP Sınıflandırılması [10]

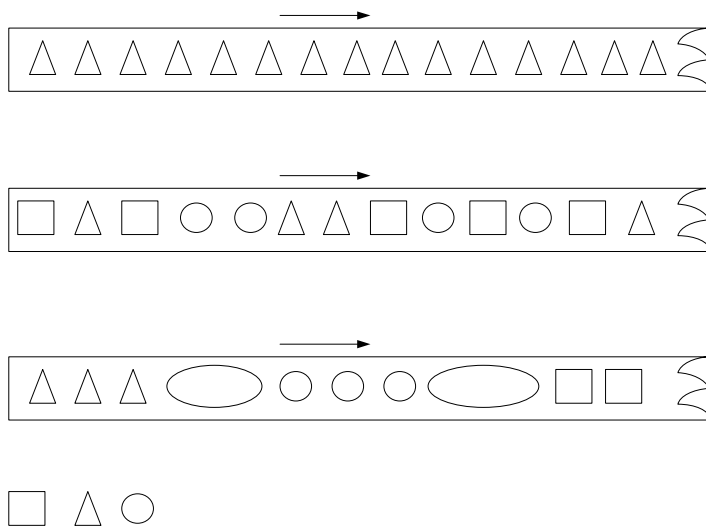
Şekil 3.2.'den de anlaşılacağı gibi MHDP'ler aynı hatta üretilecek ürün model sayısına, taşıma sistemlerine ve işlem zamanlarındaki değişkenliklere göre sınıflara ayrılabilir.

3.2.1 Ürün sayısına göre sınıflandırma

Tek modelli hatlar : Tek bir ürünün sürekli olarak ve yüksek miktarlarda üretildiği hatlardır. Tüm istasyonlarda, eş iş parçalarında aynı görevler tekrar edilir. Zaman boyunca tüm istasyonların iş yükü sabittir. Bu çalışmada geliştirilen yöntemde de tek modelli hat kullanılacaktır.

Karışık Modelli Hatlar : Bir ana ürünün çeşitli modellerinin aynı hat üzerinde üretilmesiyle oluşurlar. Ana işlemler, her bir model için ortak olduğundan model değişiminde hazırlık zamanı ya yoktur veya çok azdır. Taleplerdeki değişimi karşılayabilmek için, modeller hat üzerinde karışık olarak eş zamanlı üretilirler. Bu durumda dengeleme problemi, en etkin model sırasının belirlenmesi problemiyle ilişkilendirilir.

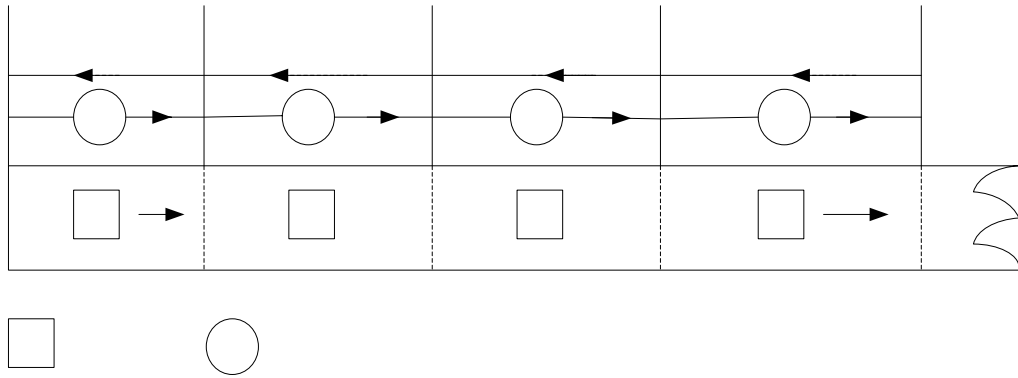
Çok Modelli Hatlar : Değişik model yada ürünlerin bir veya birden fazla montaj hattında üretilmesiyle oluşturulurlar. Üretim işlemindeki önemli farklılıklar, ürün değişiminde hattın yeniden düzenlenmesini gerektirir. Hazırlık zamanlarını azaltmak amacıyla üretim ayrı kafieler şeklinde gerçekleştirilir. Ancak kafiye büyüklükleri arttıkça hazırlık maliyetleri düşerken stok maliyetleri artmaktadır. Şekil 3.3.'de ürün sayısına göre montaj hatlarının sınıflandırılması görülmektedir.



Şekil 3.3. Tekli ve çoklu ürünler için montaj hatları [10]

3.2.2 Taşıma sistemlerine göre sınıflandırma

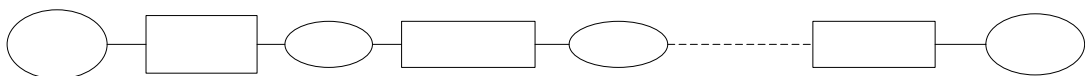
Gecikmesiz hatlar : Konveyörler, hareketli bantlar gibi malzeme taşıma ekipmanları, istasyonları esnek olmayan bir biçimde birbirine bağlar (Şekil 3.4.). İş parçaları hareketli bantlar üzerinde bir istasyondan diğerine sabit hızda hareket ederler veya işlendikten sonra kesik kesik transfer edilirler. Her iki durumda da her bir istasyona, görevlerin yerine getirilmesinde aynı zaman miktarı verilir.



Şekil 3.4. Gecikmesiz montaj hattı yerleşimi [10]

Şekil 3.4.'de sürekli hareket eden bir bant görülmektedir. İş parçası istasyondan geçerken operatörde onunla birlikte hareket etmekte, işlemi gerçekleştirmekte, ve istasyonun başlangıç noktasına geri dönmektedir. Kesikli transferde ise parça istasyonda beklemekte ve işlem biter bitmez bir sonraki istasyona sevk edilmektedir. Transferdeki gecikmeyi önlemek için, malzemenin hareketinin çok hızlı olması gerekmektedir.

Gecikmeli Hatlar : Bu tip hatlarda istasyonlar arasında tampon bulunmaktadır. Tampon, takip eden istasyonda bir önceki işlem devam ederken iş parçasının tutulduğu yer olarak tanımlanır.



Şekil 3.5. Gecikmeli montaj hattı yerleşimi [10]

Tampon kapasiteleri kısıtlı olduklarından, eğer takip eden tampon tam doluyorsa istasyon tıkanır. Bu durumda istasyon, tampon da alan müsait olana kadar boş bekleyecektir. Bir başka olumsuz durum ise istasyon açlığıdır. İstasyon açlığı istasyonun uygun olması durumunda işlem yapacak parçanın olmamasıyla ortaya çıkar bu durumda tampona parça girene kadar istasyon boş bekleyecektir. İstasyon açlığının sebepleri düşük üretim oranları ve bir önceki istasyonda meydana gelen arızalar olabilmektedir. İstasyon açlığı, eğer iş yükleri eşit olarak dağıtılmazsa, gecikmesiz hatlarda kaçınılmazdır.

3.2.3 İşlem zamanlarının değişkenliğine göre sınıflandırma

Deterministik İşlem Zamanları : Montaj hattı dengeleme modellerinin çoğu sabit deterministik işlem zamanlarını temel alırlar. Bu kabul, beklenen görev zamanlarının değişimi oldukça küçük olduğunda doğru olur. Montaj hatlarında genelde insan faktörüyle gerçekleştirilen görevlerin zamanını sabit tutabilmek için çalışanların kalifiyeli ve yüksek motivasyona sahip olmaları gerekmektedir. Gelişmiş üretim teknolojileri otomasyonu arttırarak görev zamanlarındaki değişimi azaltmışlardır.

Stokastik İşlem Zamanları : İnsan faktörü görev sürelerinin değişken olmasına yol açmaktadır bu da sistem performansını önemli ölçüde etkilemektedir. İşlem zamanları çalışanların becerisine ve motivasyonuna bağlıdır. Söz konusu değişkenliğin sebepleri arasında yorulma, dikkatin dağılması, yetersiz nitelikteki işgücü, iş tatminsizliği, hatalı girdiler, araç-gereç bozulmaları gibi unsurlar sayılabilir. Bu durum, istasyonlara atanan işlerin aldıkları toplam sürenin istasyona verilen süreyi aşmasına ve dolayısıyla bazı işlerin bitirilememesine sebep olabilmektedir. Özellikle işler arasındaki öncelik ilişkileri gözönüne alındığında, bazı işlere hiç başlanamamaktadır. Bu sebeple hattın çıktı oranı görev zamanlarının istatistiksel dağılımına bağlı olmaktadır.

Dinamik İşlem Zamanları : İşlem zamanlarındaki stokastik değişimin yanında, üretim sisteminin geliştirilmesi ve öğrenmenin etkisiyle görev zamanlarında

sistematik bir azalma olması mümkündür. Genellikle yeni bir montaj hattı sistemi kurulduğunda, operatörün işe alışincaya kadar harcadığı zaman fazlayken, kalifiyesi arttıkça ve öğrenme derecesine bağlı olarak görev zamanı azalır.(10)

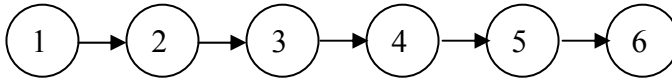
Son yıllarda; yukarıda belirtilen çevresel, makina ve insana bağlı değişkenlikleri ve uzman deneyimini göz önünde bulunduran; ayrıca deterministik, stokastik ve dinamik işlem zamanlarının belirlenmesinde karşılaşılan olumsuz yönleri elimine edecek BM kavramı ortaya çıkmıştır. Bu kavram ile operasyon süreleri *bulanık işlem zamanları* olarak belirlenebilmektedir. Ayrıca deterministik işlem zamanlarındaki; gerçeği tam olarak yansıtmayan, sabit bir sürenin operasyon zamanını temsil etmesi ve stokastik işlem zamanlarının hesaplanmasındaki; yeterli sayıda düzgün veri elde edememe ihtimali, dağılım fonksiyonlarının tam olarak sistemi yansıtmaması durumu, ve yüksek maliyet gibi olumsuz etkiler bu kavramla ortadan kaldırılabilmektedir. Bu çalışmada uzman kişi tarafından bulanık operasyon zamanları tespit edilecek ve bu bulanık operasyon zamanları ilgili operasyonunun süresini temsil edecektir.

3.3 Montaj Hatlarının Yerleşimi

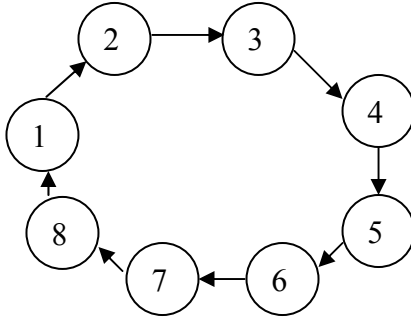
Donanımın ve iş istasyonlarının yerleşim biçimi, hat tipindeki üretimleri etkileyen önemli bir etmendir. Hattın bulunduğu yer ve üretilecek ürünün özellikleri, hattın alacağı şekli belirler. Fiziksel montaj hatları; düz, dairesel, değişik açılı, U-şekilli ve zig-zag gibi değişik biçimlerde tasarlanabilir.

Basit ve sistematik olması, kolayca yerleşim yapılabilmesi, servis verme olanaklarının kolaylıkla sağlanabilmesi, konveyör sistemlerinin uygulanabilirliğinin artması ve maliyetlerinin düşmesi, ayrıca köşelerde meydana gelebilecek transfer zorlukların ortadan kalkması gibi nedenlerle, montaj hatlarının yerleşiminde düz hatlar yeğlenir. Ama aşağıda belirtilen bazı özel durumlarda, değişik şekillerde montaj hatlarının kullanımı söz konusu olabilir.

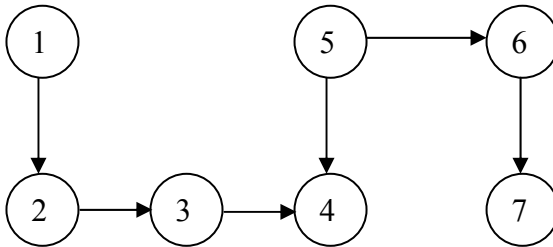
- Hattın boyu uzun ise, varolan yere sığmıyorsa veya düz hat yerleştirme yapıldığında boşta kalacak alanın boyutları büyükse, U-şekilli veya dairesel hat kullanılır.
- Elektrik ve basınçlı hava gibi tesisat bağlantıları, birden fazla istasyona aynı kaynaktan yapılıyorsa U-şekilli hatlar kullanılır.
- Varolan alan uygun değilse ve çok bitişik düzenlemeler gerektiriyorsa, düz hat dışında kalan uygun bir hat şekli tercih edilir.
- Maliyeti yüksek olan bir makineye, birbirinden farklı iki etkinlik yaptırılması gerekiyorsa, U-şekilli hatlar tercih edilir.
- Bir kişi, değişik işlem sıralarındaki ve aynı hat üzerindeki makinalardan birden çoğu ile çalışıyorsa U-şekilli hat kullanılır.



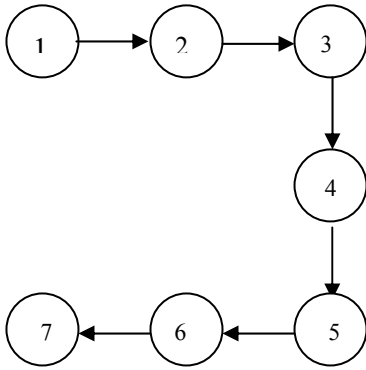
Şekil 3.6. Düz hat



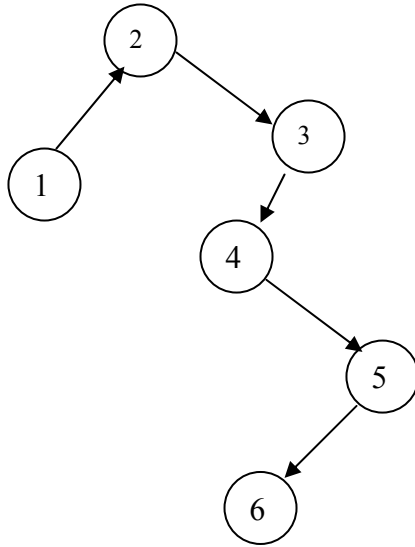
Şekil 3.7. Dairesel hat



Şekil 3.8. Zig-zag hat



Şekil 3.9. U- Şekilli hat



Şekil 3.10. Değişik açılı hat

3.4 Geleneksel Montaj Hatları ve Dengeleme

MHDP sınıflandırması dikkate alındığında GMHDP’de tek model üretilmekte ve operasyon süreleri deterministik kabul edilmektedir. Bu problemdeki diğer varsayımlar ise; öncelik diyagramının bilindiği, bir operasyonun iki veya daha fazla istasyona bölünmesine izin verilmemesi ve her bir operasyonun kendinden önce gelen operasyonlar tamamlanmadan başlayamamasıdır. Bu modeldeki kısıtlara bakarsak; hiç bir operasyon birden fazla istasyona atanamaz, istasyon zamanı çevrim zamanından büyük olamaz ve ilgili atamalar öncelik diyagramındaki ilişkiler göz önünde bulundurularak yapılır. Geleneksel hat dengeleme probleminde modellenen üretim hattı “düz” olarak organize edilmiştir. Öncelik diyagramındaki ilk

görev(ler)den başlamak ve diyagram boyunca görevleri istasyonlarda gruplamak suretiyle denge oluşturulmaktadır.

Geleneksel montaj hatları dengeleme problemlerinin iki tipi mevcuttur:

GMHDP-1: Bu problem tipinde amaç çevrim zamanı verildiğinde, istasyon sayısını minimize etmektir. Genellikle yeni bir hat tasarlamak amacıyla kullanılır.

GMHDP-2: Bu problem tipinin amacı ise istasyon sayısı verildiğinde çevrim zamanını minimize etmektir. Bu problem tipi, genellikle ürün talep değişiklikleri ve üretim proseslerinde değişiklik söz konusu olduğunda geçerlidir.

Bu çalışmada GMHDP-1 ile ilgili bir yöntem sunulacaktır.

3.5 Montaj Hatları Literatür İncelemesi

Montaj hatları ile ilgili eski yıllardan günümüze kadar bir çok çalışma yapılmıştır. Bu bölümde bu geniş literatür içerisinde özellikle tek modelli GMHDP’lerdeki en bilinen çalışmalar ve sınırlı sayıdaki bulanık montaj hattı dengeleme çalışmaları üzerinde kısaca durulacaktır.

GMHDP’lerdeki yaklaşım tarzlarına bakarsak bunların iki kısımda yoğunlaştığını görürüz. Bunlardan birincisi en iyi çözümü bulan yöntemler, ikincisi ise sezgisel yöntemlerdir.

GMHDP’lerde en iyi çözüme dayalı yöntemlerin büyük bir çoğunluğunda dal ve sınır algoritmaları kullanılmıştır. Tahngavelu ve Shetty [11], Patterson ve Albracht [12], Talbot ve Patterson [13] tamsayılı programlamaya dayalı dal ve sınır algoritmaları geliştirmişlerdir. Dal ve sınır algoritmalarına dayalı yapılan diğer çalışmalar ise; Jackson [14], Van Assche ve Herroelen [15], Wee ve Magazin [16]. Geliştirilen dal sınır algoritmalarının en bilinenleri ise; Johnson’ın [17] FABLE, Hoffmann’ın [18] EURAKA, Scholl ve Klein’in [19] SALOME isimli

algoritmalarıdır. Klein [20], Gökçen ve Erel [21] GMHDP üzerine en kısa yol metodunu uygulamışlardır. Bunların dışında GMHDP’de en iyi çözümü bulan dinamik programlamanın kullanıldığı çeşitli çalışmalar; Held, Karp ve Saresian [22], Schrage ve Baker [23] tarafından yapılmıştır.

GMHDP’de, problem boyutu büyüdükçe en iyi çözümü bulan yöntemlerin yerini sezgisel yöntemler almaya başlamış ve araştırmalar bu yöne kaymıştır. Sezgisel yöntemlerde ise iki tür yaklaşım söz konusudur. İlk yaklaşımda çözüme ulaşmak için bir kurallar dizisi belirlenir ve bu kurallar dizisine göre problem çözülür ve atamalar yapılır. İkinci yaklaşımda ise; bir başlangıç çözümü alınır ve belirlenen sezgisel kurallar ile daha iyi başka bir çözüm elde edilmeye çalışılır. En son bulunan çözümden daha iyi bir çözüm elde edilemiyorsa, en son bulunan iyi çözüm, en iyi çözüm olarak kabul edilir. İlk yaklaşımla ilgili literatürde; Tonge’nin [24], Kilbridge ve Wester’in [25], Helgeson ve Birnie [26], Moodie Young [27], Arcus’un [28] COMSOAL yöntemi, Dar-El’in [29] MALB yöntemi çalışmaları vardır. Çözüm iyileştirici ikinci yaklaşımda ise çözüm metodu olarak kullanılan yöntemler genetik algoritmalar, tabu arama, tavlama benzetimidir. Anderson ve Ferris [30], Leu ve arkadaşları [31], Kim, Kim, Kim [32], Suresh ve Sahu [33], Sabuncuoğlu ve arkadaşları [34], Ponnambalam ve arkadaşları [35] genetik algoritmaya dayalı çalışmalar yapmışlardır. Scholl ve Vob [36], Chiang [37] tarafından tabu arama yönteminin kullanıldığı; Suresh ve Sahu [38] tarafından ise tavlama benzetiminin kullanıldığı çalışmalar yapılmıştır.

BM yaklaşımını MHDP’de kullanan çalışmaların sayısı yok denecek kadar azdır. Bu alanda literatürde karşımıza çıkan ilk çalışma 1995 yılında Tsujimura, Gen ve Kubota [39] tarafından yapılmıştır. Çalışmada operasyon zamanları ve çevrim zamanı bulanık olarak düşünülmüş ve üçgensel bulanık operasyon zamanlarının kullanıldığı bir genetik algoritma geliştirilmiştir. Bundan bir yıl sonra Gen, Tsujimura ve Li [40] tarafından yapılan diğer bir çalışmada ise genetik algoritmanın bulanık MHDP’lerde kullanımı detaylandırılmış ve örneklerle desteklenmiştir. Ağpak ve Gökçen [41] ilk defa basit U tipi MHDP için bulanık mantığı kullanmışlardır. Yapılan çalışmada; çevrim zamanı, istasyon sayısı ve istasyon iş

yükü bulanık olarak düşünülerek bir tamsayılı programlama geliştirilmiştir. Burada, tahmini zor olan bu parametrelerin belirsiz olabileceği düşünülmüş ve karar vericiye bir çözüm sunmak amaçlanmıştır.

4. GELENEKSEL MONTAJ HATTI Dengeleme Problemine Bulanık Mantık Yaklaşımı

GMHDP'leri için günümüze kadar birçok yaklaşım ve çözüm yöntemi geliştirilmiştir. Montaj hattı dengelemek amacıyla tasarlanacak sistemlerdeki parametrelerin sabit değerli olamaması, diğer; çevresel, insan ve makine gibi modele büyük etkisi olabilecek parametrelerin göz ardı edilmesi gibi problemlerden dolayı bazı deterministik çözüm yöntemleri GMHDP'leri için yeterli etkiyi gösterememektedir. Bu nedenle kontrol altına alınamayan parametrelerin etkisini azaltabilmek için sezgisel yöntemlere başvurulmuş, dolayısıyla sistemler daha iyi modellenmeye başlanmıştır. Ayrıca deterministik yöntemlerdeki; veri toplama, veri hesaplama, model kurma ve modeli test etme çalışmaları oldukça uzun zaman ve maliyet getirmektedir. Bu yöntemde uzman görüşünden gerektiği gibi yararlanılamayıp, matematiksel ve istatistiksel yöntemler ön plana çıkartılmaktadır. Fakat; sistemi modellerken alınacak verilerin uzman görüşünden de yararlanılarak alınması, model dışında yer alan parametrelerin etkisini daha da azaltmakta büyük fayda sağlayacaktır. Bunun yanında; deneyimle gelen bilgi birikimi ile, ilgili girdi verilerinin belirlenmesinde uzman görüşünün, büyük zaman ve maliyet kazancı sağlayacağı açıktır. Ayrıca matematiksel olarak kesin bir şekilde belirlenemeyen parametre değerlerinin, uzman görüşünden yararlanıp, bulanık olarak belirlenmesi şartıyla; karar vericiye, dengelenmiş montaj hattı alternatifleri üretilebilecektir. Dolayısıyla uzman, sistem dışı parametreleri de düşünerek hangi montaj hattı yerleşimini uygulaması gerektiğine karar verebilecektir. GMHDP – 1 tipi problemlerde, tüm veriler deterministiktir. Özellikle gerçek uygulamalara bakıldığında tüm verilerin kesin bir şekilde belirlenmesi mümkün değildir. Mutlaka matematiksel olarak belirleyemediğimiz veya belirlese bile model içerisine aldığımızda çözüme ulaşamadığımız durumlarla karşılaşabiliriz. Dolayısıyla problemin model yapısını basit tutup, matematiksel olarak kesinliğine karar veremediğimiz parametrelerin değerlerini bulanık olarak belirleyerek, daha uygulanabilir çözüm alternatifleri geliştirebiliriz. GMHDP- 1 tipi problemde, çevrim zamanı, ilgili ürün talep değerine göre deterministik olarak belirlenebilir. Buna karşın operasyon zamanları; özellikle, emek yoğun olarak çalışılan sektörlerde

oldukça değişkenlik göstermektedir. Çünkü insan ve çevreye bağlı nedenlerden dolayı kesin bir operasyon zamanı belirlemek mümkün olamayabilmektedir yada standart zamanların hesaplanması için oldukça fazla deneme yaparak istatistik ve olasılık yardımıyla standart zamanları hesaplamak gerekebilmektedir. Ancak değişen üretim teknolojilerine ve rakabet şartlarına anında yanıt verebilmek, daha esnek ve daha kısa zamanda çözüm üretebilecek çalışmaların yapılmasını zorunlu kılmaktadır. Bu nedenlerden dolayı, bu çalışmada GMHDP - 1 problemi için; operasyon zamanları, uzman görüşü ile bulanık olarak, çevrim zamanı da sabit olarak belirlenerek karar vericiye montaj hattı yerleşimi için alternatifler sunulacaktır.

4.1 Geleneksel Montaj Hattı (GMHDP – 1) Modeli

GMHDP – 1 için literatürde Salveson [42], Bowman [43], White [44], Thangavelu ve Shetty [11], Patterson ve Albracht [12] tarafından geliştirilen modeller bulunmaktadır. Aşağıda Patterson ve Albracht tarafından kurulan deterministik model verilmiştir.

$$x_{ij} = \begin{cases} 1 & \text{eğer görev } i, j \text{ istasyonuna atanmışsa} \\ 0 & \text{diğer durumda} \end{cases} \quad \text{tüm } i=1,...,n \text{ ve } j \in SI_i \quad (4.1)$$

$$MinZ = \sum_{j \in SI_n} j \cdot x_{nj} \quad (4.2)$$

$$\sum_{j \in SI_i} x_{ij} = 1 \quad i=1,...,n \quad (4.3)$$

$$\sum_{j \in B_j} t_i \cdot x_{ij} \leq c \quad j=1,...,m \quad (4.4)$$

$$\sum_{j \in SI_k} j \cdot x_{kj} \leq \sum_{j \in SI_i} j \cdot x_{ij} \quad (k,i) \in A \text{ ve } L_k \geq E_i \quad (4.5)$$

$$x_{ij} \in \{0,1\} \quad i=1,...,n \text{ ve } j \in SI_i \quad (4.6)$$

İstasyon sayısını minimum yapacak, Eş. 4.2 amaç fonksiyonuna sahip bu modelde Eş. 4.3 atama kısıtını göstermektedir. Eş. 4.6 ile birlikte bu kısıt, her bir görevin

yanlız tek bir istasyona atanmasını sağlar. Çevrim zamanı kısıtı Eş. 4.4 ile istasyon zamanının çevrim zamanını aşmaması garanti altına alınır. Öncüllük kısıtı Eş. 4.5 ile de; bir görevin, öncülünden önce atanması engellenir.

4.2 Bulanık Operasyon Zamanlı Geleneksel Montaj Hattı (GMHDP – 1) Dengeleme Modeli

Yukarıda matematiksel modeli verilen GMHDP – 1 tipi problemlerde ve diğer montaj hattı problemlerinde kesin olarak belirlenemeyen parametreler için BM yaklaşımını kullanmak oldukça yararlı olabilmektedir. Montaj hattı problemlerinde; çevrim zamanı, istasyon sayısı, istasyon iş yükü gibi sağ taraf değişkenlerinin belirsiz olduğu durumlarda bu parametreler bulanık olarak tespit edilip modeller geliştirilebilir. Ayrıca amaç fonksiyonunun da bulanık olabileceği modeller kurmakta mümkün olabilir. Bulanık olarak düşünülebilecek bir diğer parametre ise operasyon zamanlarıdır. Bu bölümde operasyon zamanları dışındaki parametrelerin deterministik olduğu, operasyon zamanlarının ise bulanık olarak tespit edildiği model sunulacaktır.

Yukarıda belirtilen GMHDP - 1 için sadece operasyon zamanlarını gösteren t_i , bulanık olarak gösterileceği için Eş. 4.4'te değişiklik olacaktır ve ilgili kısıt;

$$\sum_{j \in B_j} \tilde{t}_i . x_{ij} \leq c \quad j=1,...,m \quad (4.7)$$

şekline dönüşecektir.

Söz konusu model toplu halde aşağıdaki gibi ifade edilebilir.

$$x_{ij} = \begin{cases} 1 & \text{eğer görev } i, j \text{ istasyonuna atanmışsa} \\ 0 & \text{diğer durumda} \end{cases} \quad \text{tüm } i=1,...,n \text{ ve } j \in SI_i \quad (4.1)$$

$$MinZ = \sum_{j \in SI_n} j.x_{nj} \quad (4.2)$$

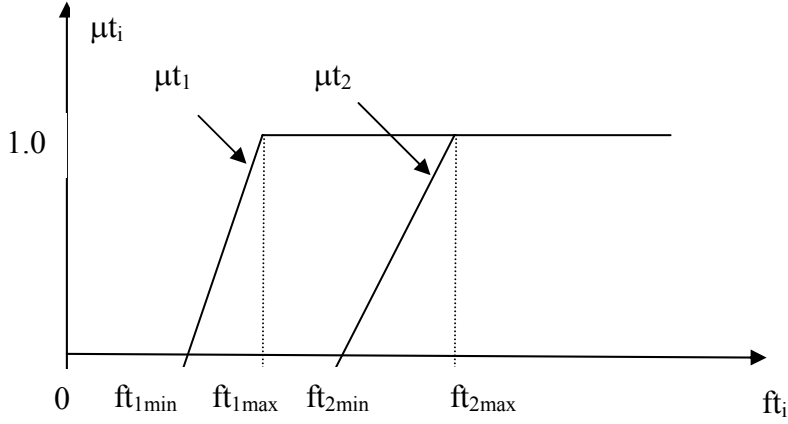
$$\sum_{j \in SI_i} x_{ij} = 1 \quad i=1,...,n \quad (4.3)$$

$$\sum_{j \in B_j} \tilde{t}_i.x_{ij} \leq c \quad j=1,...,m \quad (4.7)$$

$$\sum_{j \in SI_k} j.x_{kj} \leq \sum_{j \in SI_i} j.x_{ij} \quad (k,i) \in A \text{ ve } L_k \geq E_i \quad (4.5)$$

$$x_{ij} \in \{0,1\} \quad i=1,...,n \text{ ve } j \in SI_i \quad (4.6)$$

Deterministik olan t_i parametresinin bulanık olduğunu ifade etmek için $\tilde{t}_i$ ifadesi kullanılmıştır. Burada $\tilde{t}_i$ i operasyonuna ait bulanık operasyon zamanını temsil etmektedir. Bulanık operasyon zamanını temsil etmek amacıyla her bir operasyona ait bir üyelik fonksiyonu kullanılabilir, yada ilgili değerler kesikli olarak verilebilir. Bu çalışmada bulanık operasyon zamanlarını temsil etmesi amacıyla Şekil 4.1.'deki gibi üyelik fonksiyonları kullanılacaktır.



Şekil 4.1. Operasyon zamanı üyelik fonksiyonu

Şekil 4.1.'de iki operasyon için bulanık operasyon zamanlarına ait üyelik fonksiyonu grafik ile gösterilmiştir. Şekil 4.1.'deki üyelik fonksiyonları matematiksel olarak;

$$\mu_{t_1}(ft_1) = \begin{cases} 0 & ft_1 \leq ft_{1min} \\ \frac{ft_1 - ft_{1min}}{ft_{1max} - ft_{1min}} & ft_{1min} < ft_1 < ft_{1max} \\ 1 & ft_1 \geq ft_{1max} \end{cases} \quad (4.8)$$

$$\mu_{t_2}(ft_2) = \begin{cases} 0 & ft_2 \leq ft_{2min} \\ \frac{ft_2 - ft_{2min}}{ft_{2max} - ft_{2min}} & ft_{2min} < ft_2 < ft_{2max} \\ 1 & ft_2 \geq ft_{2max} \end{cases} \quad (4.9)$$

şeklinde ifade edilebilir. Şekil 4.1.'de, ft_{1max} 1. operasyonu tamamlamak için gerekli olan en iyimser zamanı, ft_{1min} ise 1. operasyonu tamamlamak için gerekli olan en kötümser zamanı belirtmektedir. Burada i operasyonları için μ_{t_i} değerlerinin sıfır olması demek, t_i sürelerine göre dengeleme yapmak demek olduğundan, ilgili çevrim zamanı içerisinde operasyonların tamamlanmasının çok güç olacağı anlamına gelmektedir. Dolayısıyla üretilmesi gereken ilgili ürün sayısına, zor ulaşılabilecektir. Tam tersini düşünürsek, t_{max} sürelerine göre hat dengelenirse, ilgili çevrim zamanında; operasyonlar, rahatlıkla tamamlanabilecek ve ilgili ürün sayısına, rahatlıkla ulaşılabilecektir. Burada açıklananların ışığı altında; amaç, minimum istasyon sayısına sahip dizilimler arasında hat etkinliği en yüksek olan atamayı bulmak olacaktır. Dolayısıyla hat etkinliğinin yüksek olması demek üyelik derecelerinin de yüksek olması anlamına gelmektedir. Burada dikkat edilmesi gereken bir nokta ise, her bir istasyon için farklı üyelik derecesi oluşmasıdır. Bu nedenle; karar vericiye ilgili atamanın homojen dağılımı ile ilgili bilgi vermek amacıyla, istasyonlardaki üyelik derecelerinin ortalaması ve istasyonlardaki üyelik derecelerinin varyansıda hesaplanacaktır.

Bu çalışmada bulanık sayılardan kesin sayılara geçiş ve en etkin atamayı hesaplamak için kullanılacak yöntem α kesmeleridir. Şekil 4.1.'de t_1 ve t_2 bulanık operasyon zamanlarına ait α kesmelerini;

$$ft_1 = (ft_{1max} - ft_{1min}) \alpha + ft_{1min} \quad 0 \leq \alpha \leq 1 \quad (4.10)$$

$$ft_2 = (ft_{2max} - ft_{2min}) \alpha + ft_{1min} \quad 0 \leq \alpha \leq 1 \quad (4.11)$$

şeklinde ifade edebiliriz.

α kesmeleri ile tanımlanan fonksiyonlar, model içerisinde kullanılarak, diğer kısıtların da sağlandığı uygun çözüm alternatifleri oluşturulabilecektir. Bu çözümü geliştirebilmek için hazırlanan algoritma bir sonraki aşamada anlatılacaktır.

4.3 Bulanık Operasyon Zamanlı Geleneksel Montaj Hattı (GMHDP – 1) Dengeleme Modeli Çözüm Algoritması

Üretim sistemlerinin dinamik olmasından dolayı, anlık değişimlere hızlı yanıt verebilmek oldukça önem arz etmektedir. Bu nedenle, zaman kısıtı ve maliyet kısıtı olan bu sistemlerde hızlı ve maliyeti düşük çözümler üretmek oldukça önemlidir. Örneğin tek ürünün üretildiği düz bir montaj hattı düşünelim. Bu hatta üretilen ürünle ilgili müşteri talepleri ve teknik nedenlerden dolayı sürekli olarak değişiklikler yapılması doğal bir durumdur. Dolayısıyla dengede olan bir montaj hattı bu değişimlerden etkilenerek ve dengesini kaybedecektir. Değişen montajlar ile ilgili standart zamanın tespit edilememesinden dolayı da; yeni operasyon, değişen operasyonun yerine atanacak, üretim esnasında ilgili operasyon gözlemlenerek ilgili standart zamanlar tespit edilecek ve hat dengeleme çalışması yeniden üretime başladıktan sonra yapılacaktır. Bu durumun sürekli olarak tekrar ettiğini düşünürseniz, bu değişiklikleri yapabilmek, oldukça fazla zaman alacak, sağlıklı veri alabilmek için ilgili operasyonun uzun süre gözlemlenmesi gerekecek ve gelen değişikliklere zamanında cevap verilemeyecektir. Buda darboğazlara yada boş zamanların fazla olmasına neden olacaktır. Bunun neticesinde; işletme, fazladan maliyet yüklenecektir. Bu gibi durumlarda, operasyon zamanlarının uzman tarafından bulanık olarak belirlenmesi, ve bu bulanık operasyon zamanlarına göre hat dengeleme alternatiflerinin oluşturulması, uzmana daha hızlı karar verme şansı tanıyacaktır. Bu tezde anlatılan bulanık operasyon zamanlı GMHDP – 1 çözümünde aşağıdaki adımlar uzman kişi tarafından gerçekleştirilir:

- a. Ürünü oluşturan operasyonlar tanımlanır.
- b. Operasyonlara ait öncüllük ilişkileri belirlenir.
- c. Çevrim zamanı belirlenir.
- d. Herbir operasyonun bulanık süresi en erken ve en geç tamamlanma zamanı olarak belirlenir.
- e. Bulanık süreler göre üyelik fonksiyonları tanımlanır.
- f. Üyelik fonksiyonlarına ait α kesmeleri tanımlanır.
- g. Öncüllük ilişkileri, çevrim zamanı ve α kesme fonksiyonları, ilgili bilgisayar programına girilir.
- h. Program tarafından üretilen alternatifler, hattın etkinliği, ortalama α ve α değerleri arasındaki varyansa göre değerlendirilir.

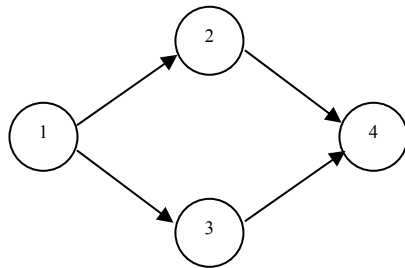
Yukarıdaki adımlar, uzman tarafından rahatlıkla gerçekleştirilebilecektir. Uzmanın belirlemiş olduğu verilerin, ilgili bilgisayar programı tarafından işlenmesi ise aşağıdaki algoritma ile gerçekleştirilir:

1. Herbir operasyonun, bulanık operasyon zamanının çevrim zamanından küçük olduğunu kontrol et. $\alpha \geq 0$ olduğunu doğrula. $\alpha < 0$ ise ilgili üyelik fonksiyonuna ait bulanık işlem zamanının çevrim zamanından büyük olduğunu hata olarak bildir.
2. Operasyonların öncüllük ilişkilerine göre atanabilirlik seviyelerini belirle.
3. Herbir atanabilirlik seviyesi içerisindeki ilgili operasyonların dizilim alternatiflerini hesaplayarak atanabilirlik derecelerine göre sıralamaları yap.
4. Herbir atanabilirlik derecesine ait sıralamanın herbirini, diğer atanabilirlik derecelerindeki sıralamalarının herbiri ile birleştir.
5. Minimum istasyon sayısını 0 olarak belirle.
6. Minimum istasyon sayısını bir artır. Minimum istasyon sayısına göre, operasyon sıralamalarının herbir elemanını ikilik sistemde (0-1); sıralama içerisinde, minimum istasyon sayısından bir eksik kadar 1 olacak şekilde işaret.

7. Minumum istasyon sayısına ait sıralamaları ikilik sistemdeki işaretlemelere göre istasyonlara ayır. İstasyonlara göre, aynı olan alternatifleri çıkararak benzersiz alternatifleri belirle.
8. Herbir alternatif için istasyondaki α kesme fonksiyonlarını toplayıp çevrim zamanına eşitle ve $1 \geq \alpha \geq 0$ olduğunu kontrol et.
9. Tüm istasyonlarda sağlanmak üzere; eğer istasyon için $1 \geq \alpha \geq 0$ veya $\alpha > 1$ ise ilgili istasyon sayısı için mümkün çözüm elde edilir. $\alpha > 1$ için, α maksimum değerini alır, $\alpha = 1$ yapılır. Eğer herhangi bir istasyonda $\alpha < 0$ ise ilgili istasyon için mümkün çözüm yoktur, 6. adıma geri dön.
10. Mümkün operasyon atamalarının gerçekleştirilebildiği, minumum istasyona sahip dizilimler için, istasyon α değerlerini yaz, hattın etkinliğini, ortalama α değerlerini, α değerlerine ait varyansı hesapla ve yaz. Maksimum etkinliğe sahip alternatifi işaretle.

Yukarıda; adımları verilen çözüm yaklaşımını, bir örnek üzerinde açıklayalım. Uzman tarafından verilen bilgilerin aşağıdaki gibi olduğunu düşünelim:

- a. Ürün 4 operasyonda tamamlanmaktadır. $n = 4$.
- b. Ürüne ilişkin öncelik diyagramı Şekil 4.2.'de görülmektedir.

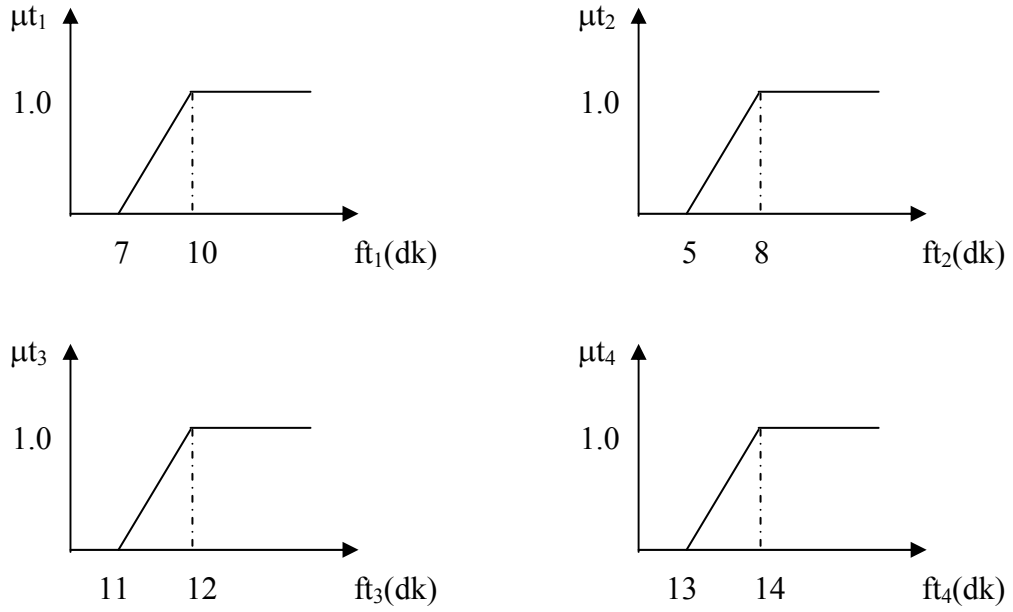


Şekil 4.2. 4 Operasyonlu öncelik diyagramı

- c. $C = 15$ dk.
- d. 1 nolu operasyon en erken 7 dk, en geç 10 dk'da tamalanabilir.
2 nolu operasyon en erken 5 dk, en geç 8 dk'da tamamlanabilir.
3 nolu operasyon en erken 11 dk, en geç 12 dk'da tamamlanabilir.

4 nolu operasyon en erken 13 dk, en geç 14 dk'da tamamlanabilir.

e. Bulanık süreler için üyelik fonksiyonları Şekil 4.3.'te görülmektedir.



Şekil 4.3. 4 Operasyon için üyelik fonksiyonları

f. Üyelik fonksiyonlarına ait α kesmeleri;

$$ft_1 = 3\alpha + 7$$

$$ft_2 = 3\alpha + 5$$

$$ft_3 = \alpha + 11$$

$$ft_4 = \alpha + 13$$

Uzman tarafından belirlenen değerlere göre program algoritması aşağıdaki şekilde işleyecektir.

1. α kesme fonksiyonlarında, α değeri, $\alpha \geq 0$ eşitliğini sağlıyorsa ilgili fonksiyon, çevrim zamanından küçük veya çevrim zamanına eşit değer alabilecektir.

1. operasyon için;

$$3\alpha + 7 = 15 \text{ olduğundan } \alpha > 0$$

2. operasyon için;

$$3\alpha + 5 = 15 \text{ olduğundan } \alpha > 0$$

3. operasyon için;

$$\alpha + 11 = 15 \text{ olduğundan } \alpha > 0$$

4. operasyon için;

$$\alpha + 13 = 15 \text{ olduğundan } \alpha > 0$$

2. Operasyonların öncüllük ilişkilerine göre atanabilirlik dereceleri belirlenir;

Çizelge 4.1. 4 Operasyon için atanabilirlik dereceleri

Operasyon No	Atanabilirlik Derecesi
1	1
2	2
3	2
4	3

Atanabilirlik derecelerine göre; ilk önce 1 nolu operasyon atanmalıdır, daha sonra 2 veya 3 nolu operasyon atanabilir. 2 ve 3 nolu operasyonların atanması tamamlandıktan sonra da 4 nolu operasyon atanabilir.

3. 1. atanabilirlik derecesi için dizilim alternatifi “1” olarak belirlenir. 2. atanabilirlik derecesi için dizilim alternatifleri “2,3” ve “3,2” dir. 3. atanabilirlik derecesi için dizilim alternatifi “4” şeklindedir.

4. Herbir atanabilirlik derecesi alternatifleri, diğer atanabilirlik dereceleri alternatifleri ile birleştirildiğinde; “1,2,3,4” alternatifi ve “1,3,2,4” alternatifi elde edilir.

5. Minimum istasyon sayısı 0 yapılır.

6. Minimum istasyon sayısı 1 artırılır. “1,2,3,4” ve “1,3,2,4” alternatifleri için, ikilik sistemde, sıfır tane 1 içeren işaretlemeler yapılır.

Çizelge 4.2. 1 İstasyon için ikilik sistemde işaretlemeler

Alternatif Sıralama	1234	1324
İkilik Sistemde İşaretleme	0000	0000

7. Program ikilik sistemdeki sayının sağından başlayarak 1 gördüğü yerin soluna yaptığı işaretleme ile ilgili sıralamaları istasyonlara ayırır ve minumum istasyon sayısına ait olan benzersiz alternatifleri süzer. Burada, dizilimde 1 olmadığı için “1,2,3,4” ve “1,3,2,4” sıralama alternatifleri tek istasyona atanır. Bunlar tek istasyon için aynı atamalar olduğundan bir tanesi alınır.

Çizelge 4.3. 1 İstasyon için benzersiz alternatifler

İstasyon Sayısı	Benzersiz Alternatifler
1	1234

8. Benzersiz alternatifler için α kesmeleri toplamı Çizelge 4.4.’te görülmektedir.

Çizelge 4.4. 1 İstasyon alternatifleri için α kesmeleri toplamı

İstasyon Sayısı	Alternatifler	α Kesmeleri Toplamı
1	1234	$8\alpha+36$

İlgili alternatif için; $8\alpha+36 = 15$ olduğundan $\alpha = -1,4$ ’tür.

9. $\alpha < 0$ olduğundan 1 istasyonlu mümkün çözüm yoktur. Algoritmada adım 6’ya geri dönlür.

10. Minumum istasyon sayısı 1 artırılır. “1,2,3,4” ve “1,3,2,4” alternatifleri için, ikilik sistemde, bir tane 1 içeren işaretleme yapılır.

Çizelge 4.5. 2 İstasyon için ikilik sistemde işaretleme

Alternatif Sıralama	1234	1324
İkili Sistemde İşaretleme	1000	1000
	0100	0100
	0010	0010
	0001	0001

11. Program ikilik sistemdeki sayının sağından başlayarak 1 gördüğü yerin soluna yaptığı işaretleme ile ilgili sıralamaları istasyonlara ayırır ve minumum istasyon sayısına ait olan benzersiz alternatifleri süzer. Benzersiz alternatifler Çizelge 4.6.’da verilmiştir.

Çizelge 4.6. 2 İstasyon için benzersiz alternatifler

İstasyon Sayısı	Benzersiz Alternatifler
2	1-234
2	12-34
2	123-4
2	13-24

12. Benzersiz alternatifler için α kesmeleri toplamı Çizelge 4.7.'de görülmektedir.

Çizelge 4.7. 2 İstasyon alternatifleri için α kesmeleri toplamı

İstasyon Sayısı	Alternatifler	α Kesmeleri Toplamı
2	1-234	$(3\alpha+7) (5\alpha+29)$
2	12-34	$(6\alpha+12) (2\alpha+24)$
2	123-4	$(7\alpha+23) (\alpha+13)$
2	13-24	$(4\alpha+18) (4\alpha+18)$

13. Çizelge 4.7.'ye göre; ilk alternatifte, ikinci istasyona atanan operasyonlar için $\alpha < 0$ olmaktadır. İkinci alternatifte, yine ikinci istasyona atanan operasyonlar için $\alpha < 0$ olmaktadır. Üçüncü alternatifte, ilk istasyona atanan operasyonlar için $\alpha < 0$ olmaktadır. Son alternatifte ise her iki istasyona atanan değerler için $\alpha < 0$ olmaktadır. Dolayısıyla; tüm alternatiflerde $\alpha < 0$ olduğundan 2 istasyona ait mümkün bir atama gerçekleştirilemez.. Algoritmada adım 6'ya geri dönülür.

14. Minumum istasyon sayısı 1 artırılır. “1,2,3,4” ve “1,3,2,4” alternatifleri için, ikilik sistemde, iki tane 1 içeren işaretlemeler yapılır.

Çizelge 4.8. 3 İstasyon için ikilik sistemde işaretlemeler

Alternatif Sıralama	1234	1324
İkili Sistemde İşaretleme	1100	1100
	1010	1010
	1001	1001
	0110	0110
	0101	0101
	0011	0011

15. Benzersiz alternatifler Çizelge 4.9.'da verilmiştir.

Çizelge 4.9. 3 İstasyon için benzersiz alternatifler

İstasyon Sayısı	Benzersiz Alternatifler
3	1-2-34
3	1-23-4
3	12-3-4
3	1-3-24
3	13-2-4

16. Benzersiz alternatifler için α kesmeleri toplamı Çizelge 4.10.'da görülmektedir.

Çizelge 4.10. 3 İstasyon alternatifleri için α kesmeleri toplamı

İstasyon Sayısı	Alternatifler	α Kesmeleri Toplamı
3	1-2-34	$(3\alpha+7) (3\alpha+5) (2\alpha+24)$
3	1-23-4	$(3\alpha+7) (4\alpha+16) (\alpha+13)$
3	12-3-4	$(6\alpha+12) (\alpha+11) (\alpha+13)$
3	1-3-24	$(3\alpha+7) (\alpha+11) (4\alpha+18)$
3	13-2-4	$(4\alpha+18) (3\alpha+5) (\alpha+13)$

17. Çizelge 4.10.'a göre; bir,iki,dört ve beşinci alternatiflerde $\alpha < 0$ olduğundan, bu alternatifler için mümkün atama yoktur. Üçüncü alternatifte ise;

$$6\alpha+12 = 15$$

$$\alpha+11 = 15$$

$$\alpha+13 = 15$$

eşitliklerinden sırasıyla;

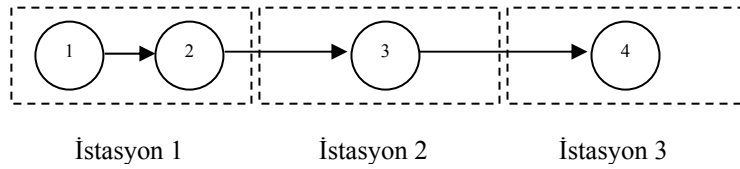
$$1. \text{ İstasyon için } \alpha=0,5$$

$$2. \text{ İstasyon için } \alpha=4$$

$$3. \text{ İstasyon için } \alpha=2$$

sonuçları elde edilir. İkinci ve üçüncü istasyonlar için α maksimum 1 olabileceğinden dolayı 2. ve 3. istasyonlar için α değerleri 1 yapılır; minimum istasyona ve maksimum α değerine sahip atama gerçekleştirilmiş olur.

Bu örnek için mümkün olan tek bir atama olduğundan dolayı, bu atama aynı zamanda en iyi atamadır. Birden fazla alternatif bulunduğunda ise; en uygun atamayı bulmak için hat etkinliğine, hat için ortalama α değerine ve istasyonlardaki α değerleri için varyansa bakılarak karar verilir. Bu, en iyi atama için hat yerleşimi Şekil 4.4.'te gösterilmiştir.



Şekil 4.4. 4 Operasyon için dengelenmiş düz hat

18. Mümkün atama için α değerleri Çizelge 4.11.'de verilmiştir.

Çizelge 4.11. En iyi atama “12-3-4” için α değerleri

İstasyon	α
1	0,5
2	1
3	1

Hattın etkinliğini hesaplamak için, istasyonlara ait ilgili α değerleri ile operasyon zamanlarının belirlenmesi gerekmektedir. Çizelge 4.11.'de “12-3-4” atamasına göre, operasyonlara ait tamamlanma süreleri görülmektedir.

Çizelge 4.12. α Değerlerine göre operasyon süreleri

Operasyon	α Kesme Foksiyonu	α	Operasyon Süresi
1	$3\alpha+7$	0,5	8,5
2	$3\alpha+5$	0,5	6,5
3	$\alpha+11$	1	12
4	$\alpha+13$	1	14

İlgili değerlere göre hattın etkinliği, ortalama α değeri ve istasyonlar arası α değerleri için varyans değerleri şu şekilde olacaktır:

$$E = (8,5+6,5+11+13)/(3*15)=\%91$$

$$\alpha_{\text{ort}} = (0,5+1+1)/3=0,83$$

$$\text{Var } \alpha = ((3 * ((0,5)^2+1^2+1^2)) - (0,5+1+1)^2)/(3*(3-1))=0,083$$

Bir sonraki bölümde MAN Türkiye A.Ş. Akyurt kablo fabrikasında yapılmış olan uygulama anlatılacaktır.

5. UYGULAMA

Ankara’da faaliyet gösteren MAN Türkiye A.Ş., Türkiye ve dünyanın çeşitli ülkelerine, şehirler arası ve şehir içi otobüs üretmektedir. Ayrıca Ankara’da ve dünyanın çeşitli noktalarında üretilen MAN araçları için MAN Türkiye, Akyurt kablo fabrikasında elektrik tesisatlarını üretmektedir.

MAN Türkiye Akyurt fabrikası, günde 8 saat olmak üzere haftanın 6 günü faaliyet göstermektedir. Bu bölümde, tek bir kamyon modeline ait gösterge panosu tesisatının üretilmesine ilişkin uygulama anlatılacaktır. İlgili gösterge panosu tesisatı tek bir hat üzerinde üretilmekte olup talep 8 adet/gün’dür.

Operasyon zamanlarının bulanık olarak belirleneceği GMHDP için gerekli bilgiler; operasyonlar ile ilgili yeterli deneyime sahip olan bir uzman tarafından verilmiştir. Belirlenen bu veriler bir önceki bölümde algoritması anlatılan ve Java programlama dili kullanılarak hazırlanan atama programına verilerek en iyi atama alternatifleri elde edilmiştir.

Hazırlanan atama programında, girdi verileri ilgili txt dosyalarına girilmektedir. Çevrim zamanı, cycle.txt dosyasına, ardıllık ilişkileri, arcs.txt dosyasına, operasyonlar için α kesme fonksiyonlarının parametreleri ise nodes.txt dosyalarına girilmektedir.

Gösterge panosu tesisatının üretimi için yapılması gereken operasyon adımları Çizelge 5.1.’de gösterilmektedir. İkinci aşamada bu operasyonlar için öncelik diyagramı oluşturulmuştur (Bkz. EK-1 Öncelik Diyagramı).

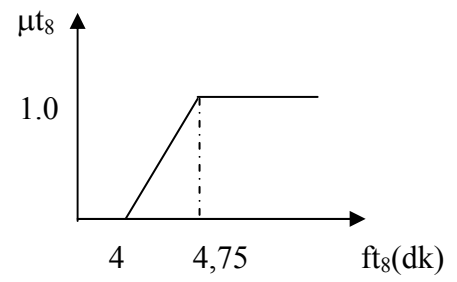
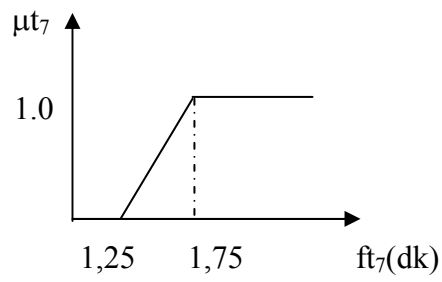
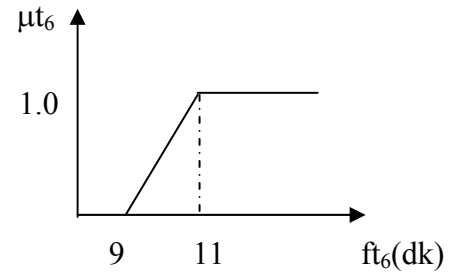
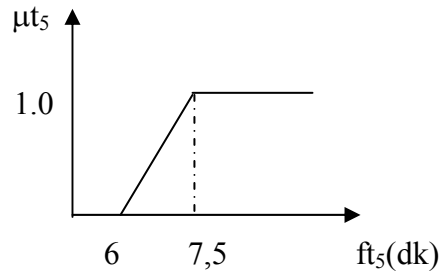
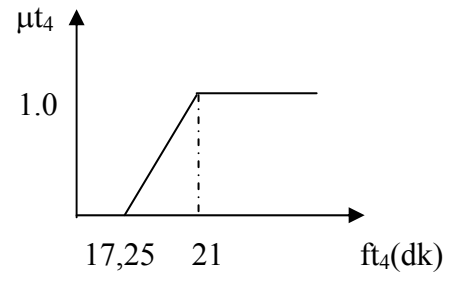
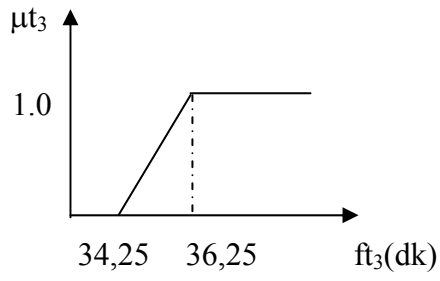
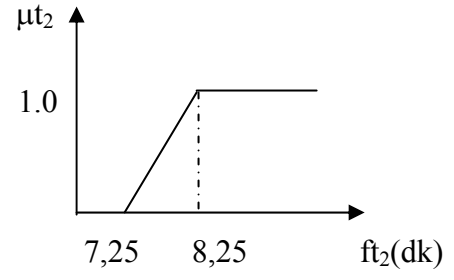
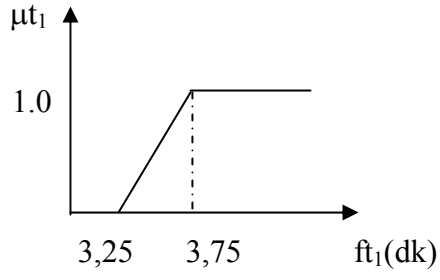
İşletme 08:00 – 16:00 saatleri arasında çalışmakta olup, gün içerisinde 30 dk yemek arası ve 10 dk’lık iki çay molası verilmektedir. Gün içerisinde kullanılan toplam verimli zaman; $8*60dk - (30dk+10dk+10dk) = 430dk/gün$ ’dür.

Günde hedeflenen üretim sayısı 8 adet olduğundan; çevrim zamanı (C) = $430/8 \approx 53$ olarak hesaplanır

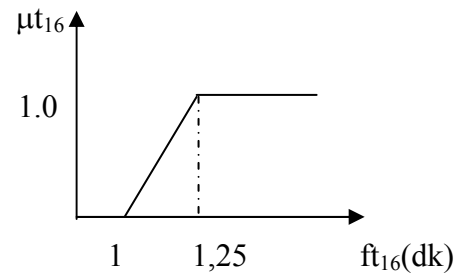
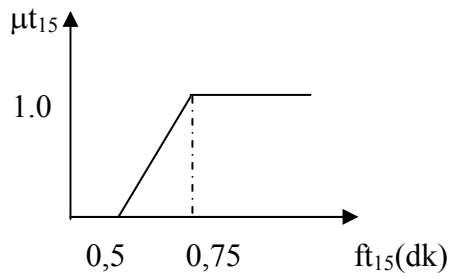
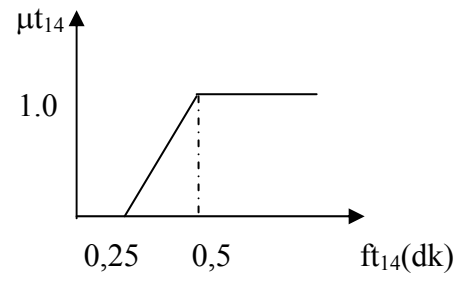
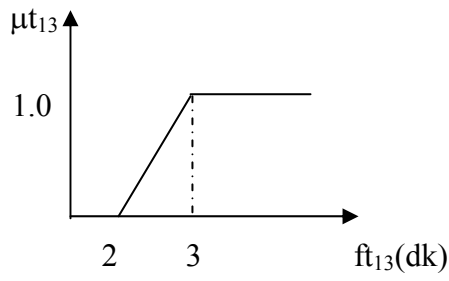
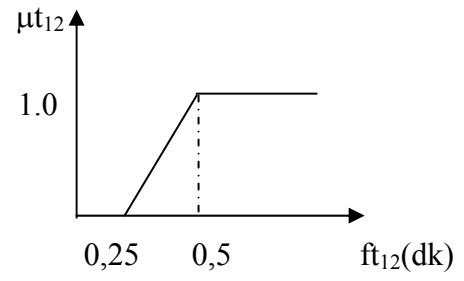
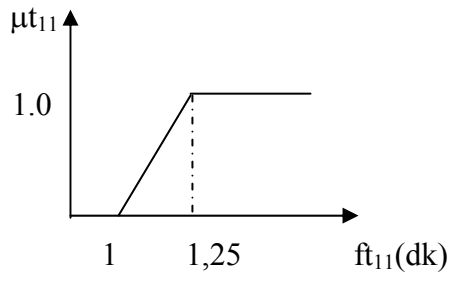
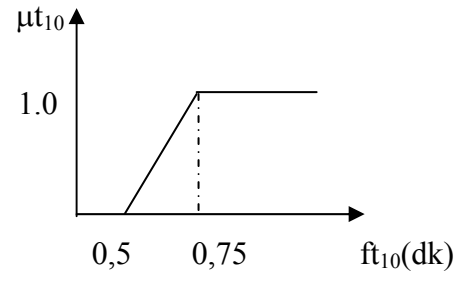
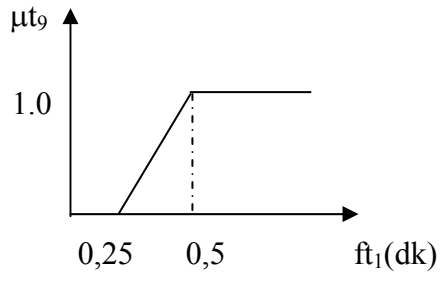
Çizelge 5.1. Gösterge panosu tesisatı operasyon tanımları ve bulanık sınır süreler

Op No	Operasyon Tanımı	Operasyon En Erken Bitiş Süresi (dk)	Operasyon En Geç Bitiş Süresi (dk)
1	Otomatik Kablo Kesim, Çift Taraf Uçlama, Numaralama (Set 1)	3,25	3,75
2	Bantlama - Barkotlama (Set 1)	7,25	8,25
3	Otomatik Kablo Kesim, Seri Uçlama, Numaralama (Set 2)	34,5	36,5
4	Bantlama - Barkotlama (Set 2)	17,25	21
5	Yarı Otomatik Kablo Kesim, Tek Taraf Uçlama, Numaralama (Set 3)	6	7,5
6	Bantlama - Barkotlama (Set 3)	9	11
7	Set 1 Hat Dizimi İçin Açma	1,25	1,75
8	Set 2 Hat Dizimi İçin Açma	4	4,75
9	Spiral Ölçüm -Kesim	0,25	0,5
10	Spiral Kaplama	0,5	0,75
11	Hat 1 Bantlama-Barkod	1	1,25
12	Hat 1 Soketleme	0,25	0,5
13	Set 3 Hat Dizimi İçin Açma	2	3
14	Spiral Ölçüm -Kesim	0,25	0,5
15	Spiral Kaplama	0,5	0,75
16	Hat 2 Bantlama-Barkod	1	1,25
17	Hat 2 Soketleme	0,5	0,75
18	Hat 1 - Hat 2 Temini ve Seçimi	3,5	4,5
19	Hat 1 - Hat 2 'nin Kontrol Panosu Üzerine Yerleştirilmesi	42	49
20	İlave Soketlerin Pano Üzerine Yerleştirilmesi	2	2,5
21	Tesisat Devre Kontrolü ve Onay Alınması	0,25	0,5
22	Klips İle Bağlama	11,75	14,25
23	Klips Kesme	1,5	2
24	Tesisat Paketleme	0,75	1,5

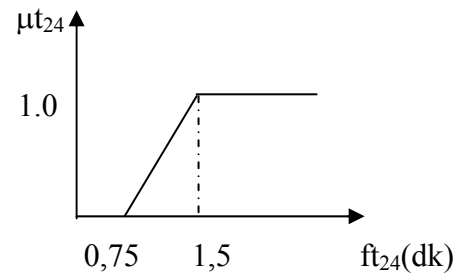
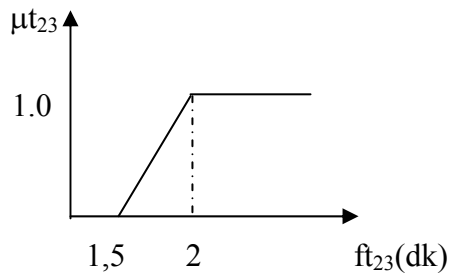
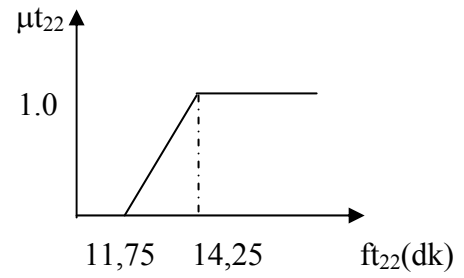
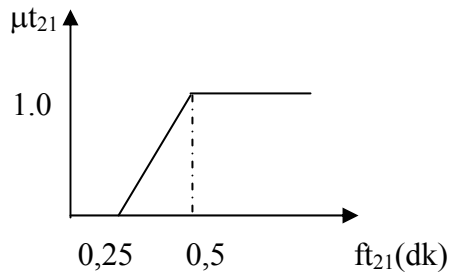
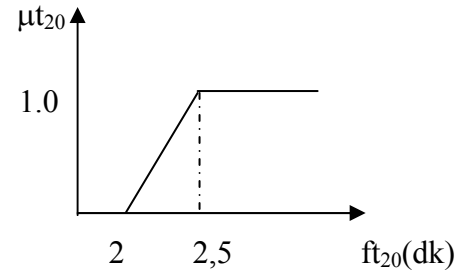
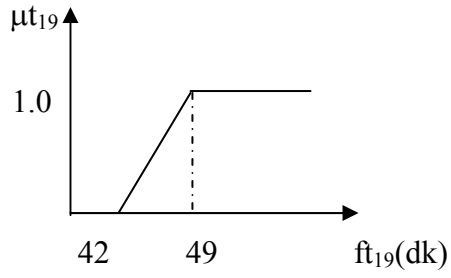
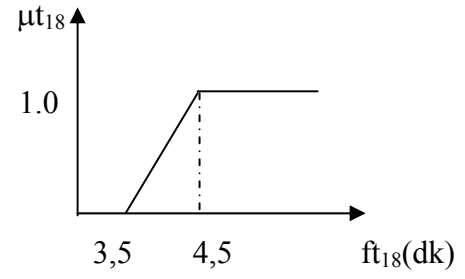
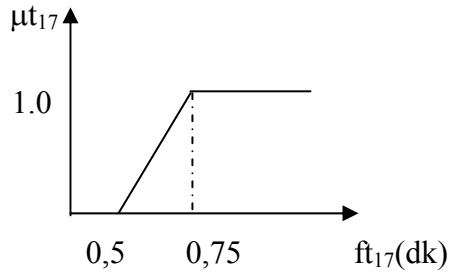
Belirlenen en erken bitiş ve en geç bitiş sürelerine göre üyelik fonksiyonlarının doğrusal olarak artış göstereceği düşünülmüştür. Buna göre ilgili üyelik fonksiyonlarının grafikleri Şekil 5.1.'de verilmiştir.



Şekil 5.1. Gösterge panosu tesisatı üyelik fonksiyon grafikleri



Şekil 5.1. (Devamı) Gösterge panosu tesisatı üyelik fonksiyon grafikleri



Şekil 5.1. (Devamı) Gösterge panosu tesisatı üyelik fonksiyon grafikleri

İlgili üyelik fonksiyonlarına ait α kesme fonksiyonları ise hesaplanarak Çizelge 5.2.'de verilmiştir.

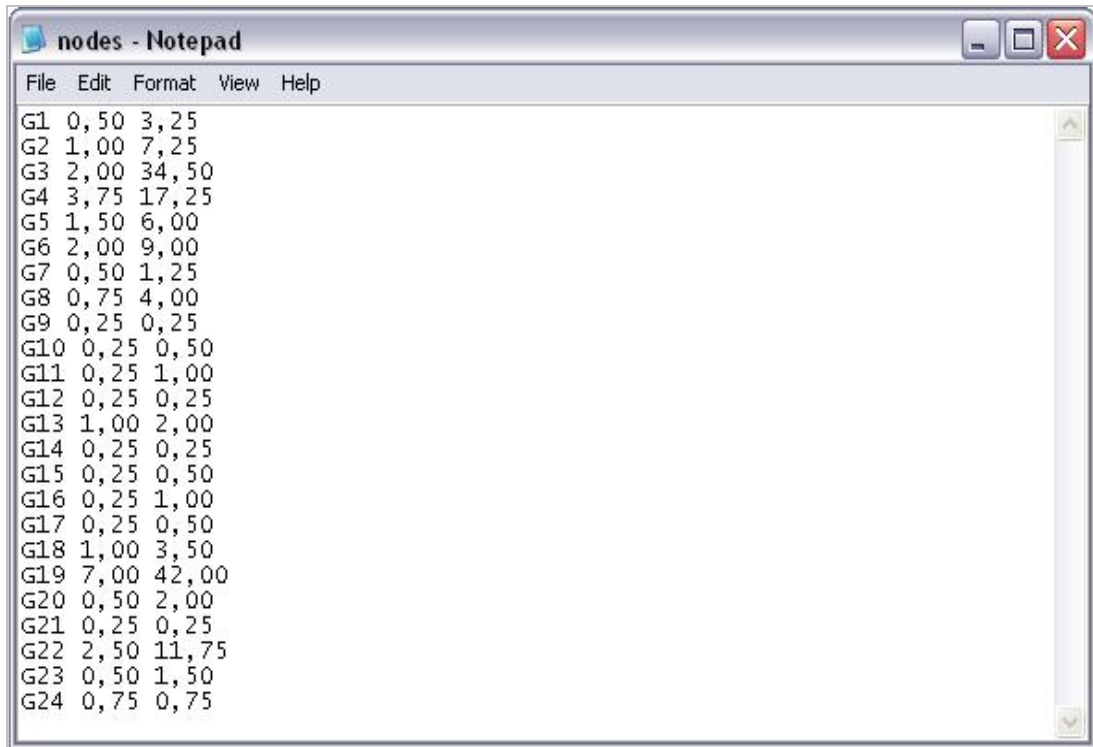
Çizelge 5.2. Gösterge panosu tesisatı üyelik fonksiyonlarının α kesme fonksiyonları

Op No	Operasyon Tanımı	α Kesme Fonksiyonları
1	Otomatik Kablo Kesim, Çift Taraf Uçlama, Numaralama (Set 1)	$0,5\alpha+3,25$
2	Bantlama - Barkotlama (Set 1)	$1,0\alpha+7,25$
3	Otomatik Kablo Kesim, Seri Uçlama, Numaralama (Set 2)	$2,0\alpha+34,5$
4	Bantlama - Barkotlama (Set 2)	$3,75\alpha+17,25$
5	Yarı Otomatik Kablo Kesim, Tek Taraf Uçlama, Numaralama (Set 3)	$1,5\alpha+6,0$
6	Bantlama - Barkotlama (Set 3)	$2,0\alpha+9,0$
7	Set 1 Hat Dizimi İçin Açma	$0,5\alpha+1,25$
8	Set 2 Hat Dizimi İçin Açma	$0,75\alpha+4,0$
9	Spiral Ölçüm -Kesim	$0,25\alpha+0,25$
10	Spiral Kaplama	$0,25\alpha+0,5$
11	Hat 1 Bantlama-Barkod	$0,25\alpha+1,0$
12	Hat 1 Soketleme	$0,25\alpha+0,25$
13	Set 3 Hat Dizimi İçin Açma	$1,0\alpha+2,0$
14	Spiral Ölçüm -Kesim	$0,25\alpha+0,25$
15	Spiral Kaplama	$0,25\alpha+0,5$
16	Hat 2 Bantlama-Barkod	$0,25\alpha+1,0$
17	Hat 2 Soketleme	$0,25\alpha+0,5$
18	Hat 1 - Hat 2 Temini ve Seçimi	$1,0\alpha+3,5$
19	Hat 1 - Hat 2 'nin Kontrol Panosu Üzerine Yerleştirilmesi	$7,0\alpha+42,0$
20	İlave Soketlerin Pano Üzerine Yerleştirilmesi	$0,5\alpha+2,0$
21	Tesisat Devre Kontrolü ve Onay Alınması	$0,25\alpha+0,25$
22	Klips İle Bağlama	$2,5\alpha+11,75$
23	Klips Kesme	$0,5\alpha+1,5$
24	Tesisat Paketleme	$0,75\alpha+0,75$

Uzman kişi tarafından belirlenen bu girdi verileri, ilgili txt dosyalarına girilerek, Java programlama dili ile kodlanan program çalışır hale getirilir. Girdi verilerinin ilgili txt dosyasına girilmesi ile ilgili arayüzler Resim 5.1, Resim 5.2 ve Resim 5.3'te verilmiştir.



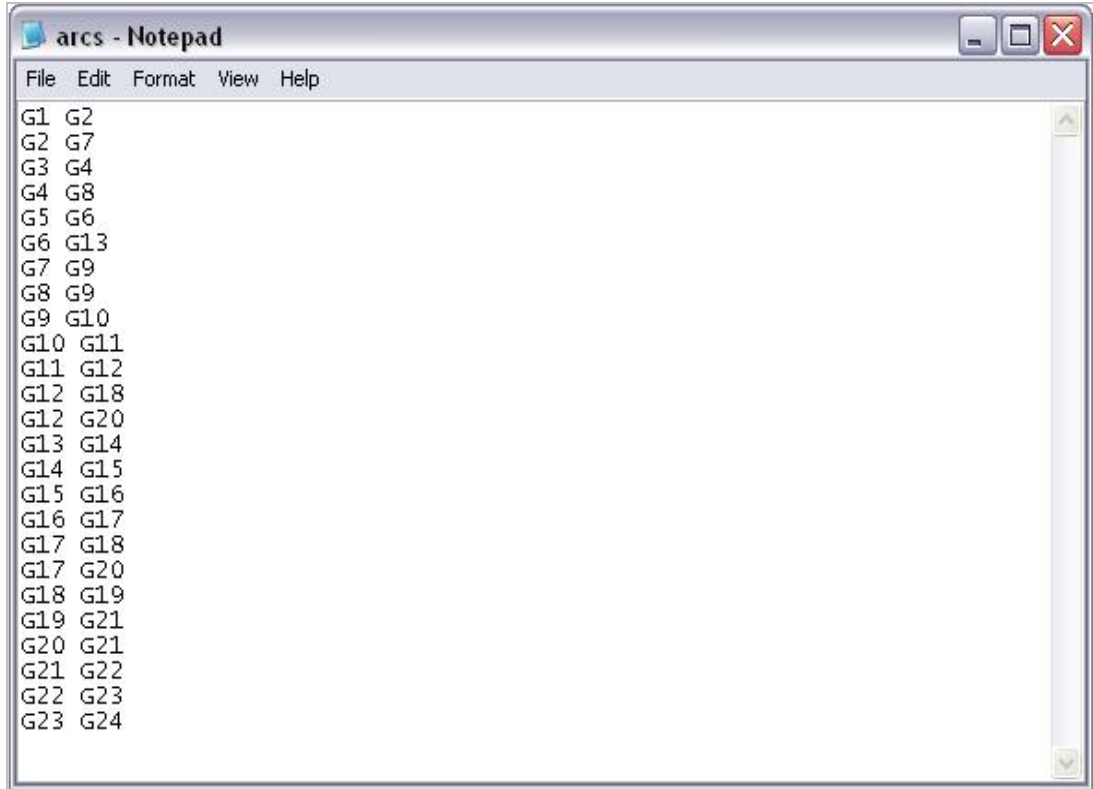
Şekil 5.2. Çevrim zamanı txt dosyası



Şekil 5.3. Operasyonlara ait α kesme fonksiyonları txt dosyası

Üyelik fonksiyonlarına ait α kesme fonksiyonları katsayıları Resim 5.2’de gösterildiği gibi ilgili txt dosyasına girilir.

Operasyonlara ait öncüllük ilişkileri ilgili txt dosyasına girilirken; öncül operasyon önce, ardıl operasyon ise, araya bir boşluk bırakılarak sonra yazılır. Bu şekilde öncelik diyagramında belirtilen tüm ilişkiler txt dosyasına tanımlanır.



Şekil 5.4. Operasyonlara ait öncüllük ilişkileri txt dosyası

Java ile kodlanan program 5 bölümden oluşmaktadır. Programa ait kodlama satırları bölümler bazında gösterilmiştir (Bkz. EK-2 Java program kodları).

Program daha önceki bölümlerde de açıklanan algoritmayı kullanarak minimum istasyona atanan görevlerin atama alternatifleri arasında, hat etkinliği (E) en yüksek olan alternatifi (yeşil işaretli alternatif) belirlemektedir. Bunun yanında uzman kişiye bilgi sunmak amacıyla tüm alternatifler verilmekte ayrıca istasyonlar arasındaki ortalama α değerinin en yüksek olduğu (mavi işaretli alternatif) ve α varyanslarının en küçük olduğu (sarı işaretli alternatif) alternatiflerde gösterilmektedir. Resim 5.4'te ilgili program tarafından üretilen atama sonuçlarına ait fuzzy.htm dosyasının arayüz görüntüsü görünmektedir. Tüm liste ise Ek 3. Atama Sonuçları bölümünde verilmiştir.

Atama Sonuçları - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Gorevler

#	Görev	Bulanık İşlem Fonksiyonu	Ardıllar
1	G1	$0.5a+3.25$	G2
2	G2	$1.0a+7.25$	G7
3	G3	$2.0a+34.5$	G4
4	G4	$3.75a+17.25$	G8
5	G5	$1.5a+6.0$	G6
6	G6	$2.0a+9.0$	G13
7	G7	$0.5a+1.25$	G9
8	G8	$0.75a+4.0$	G9
9	G9	$0.25a+0.25$	G10
10	G10	$0.25a+0.5$	G11
11	G11	$0.25a+1.0$	G12
12	G12	$0.25a+0.25$	G18 G20
13	G13	$1.0a+2.0$	G14
14	G14	$0.25a+0.25$	G15
15	G15	$0.25a+0.5$	G16
16	G16	$0.25a+1.0$	G17
17	G17	$0.25a+0.5$	G18 G20
18	G18	$1.0a+3.5$	G19
19	G19	$7.0a+42.0$	G21
20	G20	$0.5a+2.0$	G21
21	G21	$0.25a+0.25$	G22
22	G22	$2.5a+11.75$	G23
23	G23	$0.5a+1.5$	G24
24	G24	$0.75a+0.75$	

Atama Sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
1	G5 G3	1	G1 G6 G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,25	0,75	0,79
2	G5 G3 G1	1	G6 G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,25	0,75	0,79
3	G5 G3 G1 G6	0,04	G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,32	0,51	0,76
4	G5 G3	1	G1 G6 G4 G2 G13 G8 G7	0,95	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,19	0,77	0,79
5	G5 G3 G1	1	G6 G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,19	0,78	0,8
6	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,28	0,54	0,77

Şekil 5.5. Atama sonuçları htm dosyası

Program tarafından yapılan atama sonucunda minimum istasyon sayısı 4 olarak bulunmuştur ve bu istasyon sayısına ait 178 tane alternatif çıkartılmıştır. Bu alternatifler arasında 121 ve 122. sıradaki alternatifler için hat etkinliği en yüksek değerini almıştır. Hat etkinliği en yüksek olan bu iki atamaya ilişkin, operasyonların istasyonlara dağılımı Çizelge 5.3.'te verilmiştir.

Çizelge 5.3. Minimum istasyonlu hat etkinliği en yüksek atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
121	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19	1	G21 G22 G23 G24	1	0,09	0,85	0,83
122	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19 G21	1	G22 G23 G24	1	0,09	0,85	0,83

İlgili tabloyu incelersek 2,3 ve 4. istasyonlar için α , en yüksek değerini alabilmekte ve üretilen bulanık süreye göre ilgili operasyonlar rahatlıkla yapılabilmektedir. 1.

istasyonda ise α ; 0,4 değerini almakta dolayısıyla üretilen bulanık zaman ise uzman tarafından verilen, bulanık süre aralığının alt limitine yaklaşmaktadır. Bu atamalar için hat etkinliği $E=0,83$ olarak bulunmuştur. Bu değere bakarak; 2,3 ve 4. istasyonlarda α , en yüksek değerini almasına rağmen, bu istasyonlarda operasyon dışı boş zaman olduğu sonucuna varılabilir. Ayrıca α değerlerine ait varyans ve ortalama, karar vericiye diğer alternatifler arasında karşılaştırma yapma imkanı sağlar. EK-3 Atama sonuçları bölümünde, en yüksek α ortalamasına sahip alternatifler ve α değerleri arasındaki varyansın minimum olduğu alternatifler de gösterilmiştir.

6. SONUÇ VE ÖNERİLER

Bulanık mantık kavramının yaygınlaşması ile üretimde daha fazla parametrenin kontrol edilmesi gerektiği durumlarda, bulanık verilerin modelleme çalışmalarında kullanımı zorunlu hale gelebilecek ve daha etkin modelleme yaklaşımları geliştirilebilecektir. Literatürde sadece operasyon zamanlarını bulanık olarak belirleyen bir montaj hattı dengeleme çalışması bulunmamaktadır. Doğrusal programlamayı kullanan ve sağ taraf değişkenlerinin bulanık olarak belirlendiği çalışmalar ise oldukça sınırlıdır. Ayrıca, ilgili literatürde genetik algoritmanın kullanıldığı bir iki adet montaj hattı dengeleme çalışmasına rastlanmıştır.

Günümüzde, dinamik ortamdaki sistemlerle çalışırken, deneyime sahip beyin güçlerini kullanmak, değişimlere hızlı cevap vermek açısından oldukça büyük önem arz etmektedir. Bu yaklaşım ise bulanık mantık kavramının en önemli avantajlarından birisidir. Bu çalışmada da bulanık operasyon süreleri uzman görüşü ile belirlenmektedir.

Bu tezde yapılan bulanık operasyon zamanlı geleneksel montaj hattı dengeleme çalışmasında, belirsizlik durumlarında da etkin bir montaj hattı yerleşiminin tasarlanabileceği bir sezgisel yöntem gösterilmiştir. Tezde geliştirilen sezgisel yöntem ile ilgili algoritma Java programlama dili ile kodlanmış ve sonuçların daha hızlı alınması sağlanmıştır.

Bilindiği gibi operasyon sayılarının artması ve operasyonlar arasındaki ilişkilerin karmaşıklaşması sonucunda bu tip problem türü, NP-Hard problem olarak adlandırılmaktadır. Dolayısıyla daha yüksek operasyonlar ve operasyonlar arası ilişki olan modellemelerde algoritmaya farklı sezgisellerin ilave edilmesi gerekir.

Bu tezde yapılan uygulama ile MAN Türkiye A.Ş. Akyurt kablo fabrikası üretim hattında üretilen bir ürüne ait teknik değişikliklerden kaynaklanan, operasyon değişiklikleri ve bu operasyonların standart sürelerinin hesaplanması için yeterli sürenin olmamasından, ayrıca; bu operasyonlar için belirlenecek standart sürelerin

emek yoğun işlerde sabit bir değerden çok, değişkenlik göstermesinden dolayı uzman görüşünden yararlanılarak bulanık süreler ile etkin bir montaj hattı dengelemesi yapılmıştır. MAN Türkiye A.Ş Akyurt kablo fabrikasında yapılan çalışmada, 24 operasyona sahip gösterge panosu tesisatının geleneksel montaj hattı üzerindeki üretimi için dengeleme yapılmış olup, bu operasyonlara ait öncüllük ilişkileri çıkartılmış ve operasyon süreleri uzman görüşünden faydalanılarak belirlenip üyelik fonksiyonları α kesmeleri ile ifade edilmiştir. Belirlenen bu veriler, geliştirilen sezgisel algoritma için hazırlanan Java programı tarafından işlenerek ilgili sezgisel yönteme ait en iyi atama alternatifleri çıkartılmıştır. Üretim hattı 4 istasyonda dengelenmiş ve farklı α değerlerine sahip toplam 178 adet alternatif üretilmiştir. Bu 178 adet alternatif arasından en yüksek hat etkinliği %83 olan iki alternatif en iyi sonuç olarak seçilmiş ve ilk alternatif montaj hattına uygulanarak hat dengelemesi yapılmıştır. Uygulanan atama ile 1. istasyona 4 operasyon atanmış ve $\alpha=0,4$ değerini almıştır. 2. istasyona 15 operasyon atanmış $\alpha=1$ değerini almıştır. 3. istasyona 2 operasyon atanmış $\alpha=1$ değerini almış ve son istasyona 4 operasyon atanmış $\alpha=1$ değerini almıştır. İfade edilen α değeri 1'e yaklaştıkça, istasyondaki operasyonların süreleri de uzman tarafından belirlenen bulanık sürelerin üst limitine yaklaşmaktadır.

İlgili atamaya ait denge kaybı oranı, $1-E=1-0,83=0,17$ olarak hesaplanır. Buna göre toplam hat üzerindeki denge kaybı; $0,17*53*4=36,04$ dk olmaktadır.

Sezgisel algoritmanın çözümü için hazırlanan Java programı, 3Ghz ve 1024 MB RAM donanıma sahip bir bilgisayarda çalıştırılmış ve alternatifler yaklaşık 10 saat 20 dk da üretilmiştir.

Diğer montaj hattı tipleri için, operasyon zamanlarının bulanık olduğu çalışmalar, gelecek dönemlerde yapılabilir. Ayrıca, bulanık operasyon zamanları dışında, diğer değişkenlerin de bulanık olarak belirlendiği çalışmalar da yapılabilir. Böyle problemlerde karmaşıklığın artması nedeniyle bunların çözümlerinde de bazı sezgisel metotların kullanılması gerekecektir.

KAYNAKLAR

1. Kosko, B., "Fuzzy Thinking: The New Science of Fuzzy Logic", *New York*, 123-125 (1993).
2. Yılmaz, M., "Amaç Fonksiyonu Kısıtlı Ulaştırma Problemlerinde Tamsayılı Bulanık Optimizasyon", Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, 4-21 (2003).
3. Turkbey, O., "Çok Amaçlı Makina Sıralama Problemi İçin Bir Bulanık Güçlü Method", *D.E.Ü. Müh. Fak. Fen ve Mühendislik Dergisi*, 5 (3) : 81-98 (2003).
4. Evans, G. W., Karwowski, W., Wilhelm, M. R., "Applications of Fuzzy Sets Methodologies in Industrial Engineering", *New York*, 3-6 (1989).
5. Yager, R. R., Zadeh, L. A., "An Intoduction to Fuzzy Logic Applications in Intelligent Systems", *Kluwer Academic Publishers*, 69-73 (1992).
6. Klir, G. J.,Yuan, B., "Fuzzy Sets and Fuzzy Logic: Theory and Applications", *Prentice-Hall*, 5-33 (1995).
7. Lai, Y., Hwang, C., "Fuzzy Mathematical Programming, Methods and Applications", *Springer-Verlag*, 301 (1992).
8. Ibrahim, A.M, "Fuzzy Logic for Embedded Systems Applications", *Newnes*, 1-13 (2004).
9. Gündoğan, Ç., Arifoğlu, B., Erfidan, T., Çetinkaya, H. B., "Bulanık Mantık ve Evrik Sarkaç Uygulaması", *Otomasyon Dergisi*, 164 (2005).
10. Scholl, A., "Balancing and Sequencing of Assembly Lines", *Physica-Verlag*, 4-14 (1999).
11. Thangavelu, S. R., Shetty, C. M., "Assembly Line Balancing by Zero-One Integer Programming", *AIIE Transactions*, 3 (1) : 61-68 (1971).
12. Patterson, J. H., Albracht, J. J., "Assembly Line Balancing: Zero-One Programming with Fibonacci Search", *Operations Research*, 23 (1) : 166-172 (1975).
13. Talbot, F. B., Patterson, J. H., "An Integer Programming Algorithm with Network Cuts for Solving the Assembly Line Balancing Problem", *Management Science*, 30 (1) : 85-99 (1984).
14. Jackson, J. R., "A Computing Procedure for a Line Balancing Problem", *Management Science*, 2 (3) : 261-271 (1956).

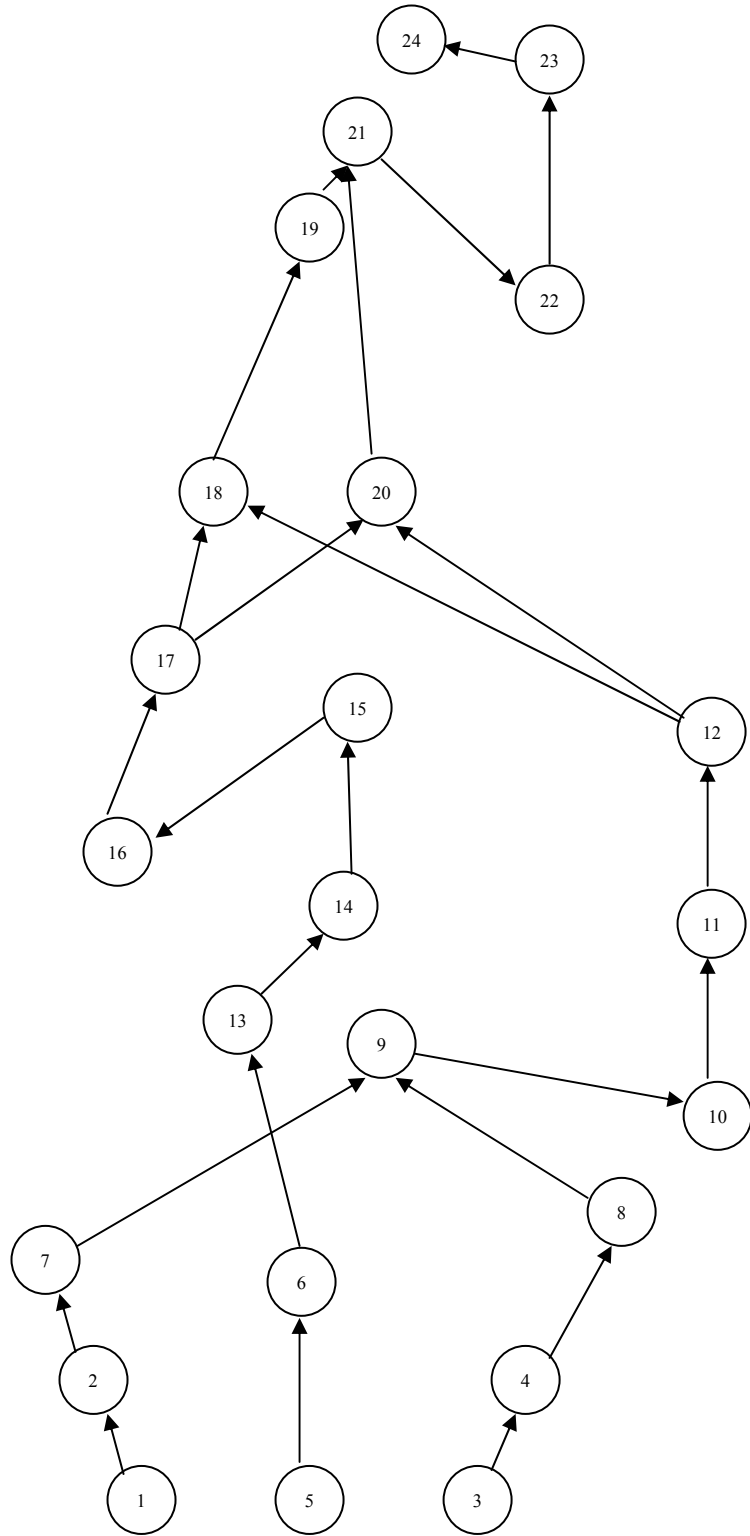
15. Van Assche, F., Herroelen, W. S., "An Optimal Procedure for the Single Model Deterministic Assembly Line Balancing Problem", ***European Journal of Operational Research***, 3 (2) : 142-149 (1979).
16. Wee, T. S., Magazine, M. J., "An Efficient Branch and Bound Algorithm for an Assembly Line Balancing Problem - Part II: Minimize The Number of Work Stations", ***University of Waterloo***, 6 : 151 (1981).
17. Johnson, R. V., "Optimally Balancing Large Assembly Lines with 'Fable'", ***Management Science***, 34 (1) : 240-253 (1988).
18. Hoffmann, T. R., "EUREKA: A Hybrid System for Assembly Line Balancing", ***Management Science***, 38 : 39-47 (1992).
19. Scholl, A., Klein, R., "SALOME: A Bidirectional Branch and Bound Procedure for Assembly Line Balancing", ***Journal on Computing***, 9 : 319-334 (1997).
20. Klein, M., "On Assembly Line Balancing", ***Operations Research***, 11(2) : 55-62 (1963).
21. Gökçen, H., Erel, E., "Shortest-Route Formulation of Mixed Model Assembly Line Balancing Problem", ***European Journal of Operations Research***, 116 : 194-204 (1999).
22. Held, M., Karp, R.M., Sharesian, R., "Assembly Line Balancing Dynamic Programming with Precedence Constraints", ***Operation Research***, 11(3) : 442-459 (1963).
23. Schrage, L., Baker, K. R., "Dynamic Programming Solution of Sequencing Problems with Precedence Constraints", ***Operations Research***, 26 : 444-459 (1978).
24. Tonge, F. M., "A Heuristic Program for Assembly Line Balancing", ***Prentice Hall Inc.***, New York, 15-24 (1961).
25. Kilbridge, M. D., Wester, L., "A Heuristic Method of Assembly Line Balancing", ***Journal of Industrial Engineering***, 12 (4) : 292-298 (1961).
26. Helgeson, W. B., Birnie, D. P., "Assembly Line Balancing Using Ranked Positional Weight Technique", ***Journal of Industrial Engineering***, 12 (6) : 394-398 (1961).
27. Moodie, C. L., Young H. H., "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times", ***Journal of Industrial Engineering***, 16 (1) : 23-29 (1965).

28. Arcus, A. L., "COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines", *International Journal of Production Research*, 4 (4) : 259-277 (1966).
29. Dar-El, E. M., "MALB-A Heuristic Technique for Balancing Large Single Model Assembly Lines", *AIIE Transactions*, 5 (4) : 343-356 (1973).
30. Anderson, E. J., Ferris, M. C., "Genetic Algorithms for Combinatorial Optimization: The Assembly Line Balancing Problem", *ORSA Journal of Computing*, 6 : 161-173 (1994).
31. Leu, Y. Y., Matheson, L. A., Rees, L. P., "Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria", *Decision Sciences*, 25 (4) : 581-606 (1994).
32. Kim, Y. K., Kim, Y. H., Kim, Y. J., "Two-Sided Assembly Line Balancing: A Genetic Algorithm Approach", *Production Planning and Control*, 11 (1) : 44-53 (2000).
33. Suresh, G., Vinod, V. V., Sahu, S., "A Genetic Algorithm for Assembly Line Balancing", *Production Planning and Control*, 7 (1) : 38-46 (1996).
34. Sabuncuoğlu, I., Erel, E., Tanyer, M., "Assembly Line Balancing Using Genetic Algorithms", *Journal of Intelligent Manufacturing*, 11 (3) : 295-310 (2000).
35. Ponnambalam, S. G., Aravindan, P., Naidu, G. M., "Multi-Objective Genetic Algorithm for Solving ALB Problem", *International Journal of Advanced Manufacturing Technology*, 16 (5) : 341-352 (2000).
36. Scholl, A., Vob, "Simple Assembly Line Balancing – Heuristic Approaches", *Journal of Heuristic*, 2 : 217-244 (1996).
37. Chiang, W. C., "The Application of a Tabu Search Metaheuristic to the Assembly Line Balancing Problem", *Annals of Operations Research*, 77 : 209-227 (1998).
38. Suresh, G., Sahu, S., "Stochastic Assembly Line Balancing Using Simulated Annealing", *Int. J. Prod. Res.*, 32 (8) : 1801-1810 (1994).
39. Tsujimura, Y., Gen, M., Kubota, E., "Solving Fuzzy Assembly Line Balancing Problem with Genetic Algorithms", *Computers and Industrial Engineering*, 29 (1-4) : 543-547 (1995).
40. Gen, M., Tsujimura, Y., Li, Y. X., "Fuzzy Assembly Line Balancing Using Genetic Algorithms", *Computers and Industrial Engineering*, 31 (3-4) : 631-634 (1996).

41. Ağpak, K., Gökçen, H., “Basit U Tipi Montaj Hattı Dengeleme Problemine Bulanık Programlama Yaklaşımı”, **D.E.Ü. Müh. Fak. Fen ve Mühendislik Dergisi**, 4 (2) : 29-40 (2002).
42. Salveson, M. E., “The Assembly Line Balancing Problem”, **Journal of Industrial Engineering**, 6 (3) : 18-25 (1955).
43. Bowman, E. H., “Assembly Line Balancing by Linear Programming”, **Operations Research**, 8 (3) : 385-389 (1960).
44. White, W. W., “Comment on a Paper by Bowman”, **Operations Research**, 9 (2) : 274-276 (1961).

EKLER

EK-1 Öncelik diyagramı



EK-2 Java program kodları

Main.java

```

/*
 * Main.java
 *
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication1;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.text.DecimalFormat;
import java.util.Scanner;
import java.io.File;
import java.util.Vector;
import java.util.Hashtable;
/**
 *
 * @author iesahin
 */
public class Main {

    private static final String nodesFile = "nodes.txt";
    private static final String arcsFile = "arcs.txt";
    private static final String outputFile = "output.txt";
    private static final String cycleFile = "cycle.txt";

    private static double CYCLETIME = 30;

    private Vector<Node> nodes = new Vector();
    private Vector<PermutableGroup> groups = new Vector();
    private Vector<Vector> alternatives = new Vector();
    private Vector<Vector> feasibles = new Vector();

    /** Creates a new instance of Main */
    public Main()
    {

        // read

        // create nodes

        // loop on
        // create stations

        // find alphas

        // write

    }

```

EK-2 (Devam) Java program kodları

```

public void createNodes()
{
    try
    {
        Scanner sc = new Scanner(new File(nodesFile));
        while(sc.hasNextLine())
        {
            String nodeName = sc.next();
            double alphaCoeff = sc.nextDouble();
            double constant = sc.nextDouble();

            // check if we have a node with identical name

            for (Node n : nodes)
            {
                if (n.name.equals(nodeName))
                {
                    System.err.println(nodesFile + " contains multiple nodes with same
name!");
                    System.exit(99);
                }
            }
            Node n = new Node();
            n.name = nodeName;
            n.alpha = alphaCoeff;
            n.constant = constant;
            nodes.add(n);
        }
        sc.close();
    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
}

public void loadCycleTime()
{
    try
    {
        Scanner sc = new Scanner(new File(cycleFile));
        CYCLETIME = sc.nextDouble();
        sc.close();
    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
}

public void createArcs()
{
    try

```

EK-2 (Devam) Java program kodları

```

{
    Scanner sc = new Scanner(new File(arcsFile));
    while(sc.hasNextLine())
    {
        String predName = sc.next();
        String succName = sc.next();

        Node succ = null;
        Node pred = null;

        // find the nodes
        for (Node n : nodes)
        {
            if (n.name.equals(predName))
            {
                pred = n;
            }
            if (n.name.equals(succName))
            {
                succ = n;
            }
        }

        if (pred == succ)
        {
            System.err.println("Predecessor and Successor cannot be identical in "
+ arcsFile);
            System.exit(99);
        }

        if (pred == null || succ == null)
        {
            System.err.println("Predecessor or Successor in " + arcsFile + "
cannot be found in " + nodesFile);
            System.exit(99);
        }

        pred.successors.add(succ);
        succ.predecessors.add(pred);
    }

    sc.close();
}
catch (Exception ex)
{
    ex.printStackTrace();
}

}

public void buildPermutableGroups()
{
    // copy nodes

```

EK-2 (Devam) Java program kodları

```

Vector<Node> nodesReplica = this.buildNodesReplica();
while (!nodesReplica.isEmpty())
{
    //find nodes without predecessor
    PermutableGroup pg = new PermutableGroup();

    for (Node nr : nodesReplica)
    {
        if (nr.predecessors.isEmpty())
        {
            pg.nodes.add(nr);
        }
    }

    for (Node nr : pg.nodes)
    {
        nodesReplica.remove(nr);
    }

    //remove group nodes from predecessor lists
    for (Node n : pg.nodes)
    {
        for (Node nr : nodesReplica)
        {
            if (nr.predecessors.contains(n))
            {
                nr.predecessors.remove(n);
            }
        }
    }

    this.groups.add(pg);
}
}

public void buildAllAlternatives()
{
    Vector<Vector> nextAlt = new Vector();

    alternatives = this.groups.firstElement().getPermutations();

    int crossingGroupIndex = 1;

    for (; crossingGroupIndex < groups.size(); crossingGroupIndex++)
    {
        Vector<Vector> crossingGroupPerms =
groups.elementAt(crossingGroupIndex).getPermutations();

        for (Vector vva : alternatives)
        {

```


EK-2 (Devam) Java program kodları

```

for (Vector vvn : crossingGroupPerms)
    {
        Vector nv = new Vector();
        nv.addAll(vva);
        nv.addAll(vvn);
        nextAlt.add(nv);
    }

    alternatives = nextAlt;
    nextAlt = new Vector();

}

}

public Vector<Station> numString2Station(long bitStr, Vector<Node> vn)
{
    int elementIndex;

    Vector<Station> stations = new Vector();

    Station s = new Station();

    int nodesize = vn.size();
    for (elementIndex = 0; elementIndex < nodesize; elementIndex++)
    {
        Node n = vn.elementAt(elementIndex);
        s.nodes.add(n);
        if (((bitStr >> elementIndex) & 1L) > 0)
        {
            stations.add(s);
            s = new Station();
        }
    }

    stations.add(s);
    return stations;
}

public boolean isFeasible(Vector<Station> stations)
{
    for (Station s : stations)
    {
        if (s.getTotalConstant() > this.CYCLETIME)
        {
            return false;
        }
    }

    return true;
}

```

EK-2 (Devam) Java program kodları

```

public Vector<Vector> checkStationsFor(int n)
{
    Vector<Long> stationConfs = this.getStationDelimiters(this.nodes.size(), n);
    Vector<Vector> feasibles = new Vector();
    for (Vector alt : alternatives)
    {
        for (Long ll : stationConfs)
        {
            Vector<Station> stations = this.numString2Station(ll, alt);
            if (this.isFeasible(stations))
            {
                feasibles.add(stations);
            }
        }
    }

    return feasibles;
}

public void findBestConf()
{
    Vector<Vector> vv = new Vector();
    for (int i = 0; i < nodes.size(); ++i)
    {
        vv = checkStationsFor(i);
        if (vv.size() > 0 )
        {
            this.feasibles = vv;
            break;
        }
    }
}

public Vector<Node> buildNodesReplica()
{
    //contains <original, replica> object reference pairs
    Hashtable<Node, Node> ht = new Hashtable();

    //contains new objects
    Vector<Node> replicas = new Vector();

    //add all elements of nodes to replica
    for (Node n : nodes)
    {
        Node nr = new Node();
        nr.name = n.name;
        nr.alpha = n.alpha;
        nr.constant = n.constant;
        nr.predecessors = (Vector<Node>) n.predecessors.clone();
        nr.successors = (Vector<Node>) n.successors.clone();
        replicas.add(nr);
        ht.put(n, nr);
    }
}

```

EK-2 (Devam) Java program kodları

```
//correct references in predecessors and successors by ht
```

```

    for (Node nr : replicas)
    {
        Vector<Node> succsReplica = new Vector();
        for (Node s : nr.successors)
        {
            Node sr = ht.get(s);
            succsReplica.add(sr);
        }
        nr.successors = succsReplica;

        Vector<Node> predsReplica = new Vector();
        for (Node p : nr.predecessors)
        {
            Node pr = ht.get(p);
            predsReplica.add(pr);
        }
        nr.predecessors = predsReplica;
    }

    return replicas;
}

public String toString()
{
    DecimalFormat df = new DecimalFormat();
    df.setMaximumFractionDigits(2);

    String str = "";
    for (PermutableGroup pg : groups)
    {
        str += pg.toString();
    }

    for (Vector v : alternatives)
    {
        str += "\n\nALTERNATIVE\n";
        for (Object o : v)
        {
            str += ((Node)o).toString() + "\n";
        }
    }

    str += "\n\nFEASIBLES:\n";
    str += "TOTAL: " + feasibles.size() + "\n";
    str += "STATION COUNT: " + feasibles.elementAt(0).size() + "\n";
    int i = 1;

```

EK-2 (Devam) Java program kodları

```

for (Vector v : feasibles)
{
    str += "" + (i++) + " ) -----\n";
    str += "EFFICIENCY: " + df.format(stationLineEfficiency(v)) + "\n";
    str += "AVG(ALPHA): " + df.format(stationAvgAlpha(v)) + "\n";
    str += "VAR ALPHA: " + df.format(stationAlphaVariance(v)) + "\n";
    for (Object o : v)
    {
        ((Station) o).CYCLETIME = this.CYCLETIME;
        str += ((Station) o).toString() + "\n";
    }
}

return str;
}

public Vector<Long> getStationDelimiters(int numTasks, int numStations)
{
    long max1 = (long) Math.pow(2, numTasks);

    Vector<Long> v1 = new Vector();

    for (long l = 0; l < max1; ++l)
    {
        if (Long.bitCount(l) == numStations)
        {
            v1.add(l);
        }
    }

    return v1;
}

public boolean isStationsEqual(Vector<Station> vs1, Vector<Station> vs2)
{
    if (vs1.size() != vs2.size())
        return false;

    for (int i = 0; i < vs1.size(); i++)
    {
        if (! vs1.elementAt(i).equals(vs2.elementAt(i)))
            return false;
    }

    return true;
}

```

EK-2 (Devam) Java program kodları

```

    }

    public double stationAvgAlpha(Vector<Station> vs)
    {
        double totalAlpha = 0;

        for (Station s : vs)
        {
            totalAlpha += s.alpha(CYCLETIME);
        }

        return totalAlpha / vs.size();
    }

    public double stationLineEfficiency(Vector<Station> vs)
    {
        double accumulatedTime = CYCLETIME * vs.size();

        double employedTime = 0;

        for (Station s : vs)
        {
            employedTime += s.alpha(CYCLETIME) * s.getTotalAlpha() +
s.getTotalConstant();
        }

        return employedTime / accumulatedTime;
    }

    public double stationAlphaVariance(Vector<Station> vs)
    {
        double mu = 0;

        for (Station s : vs )
        {
            mu += s.alpha(CYCLETIME);
        }

        mu = mu / vs.size();

        double var = 0;

        for (Station s : vs)
        {
            var += (mu - s.alpha(CYCLETIME)) * (mu - s.alpha(CYCLETIME));
        }

        var /= (vs.size()-1);

        return var;
    }

```

EK-2 (Devam) Java program kodları

```

public void filterFeasibles()
{
    Vector<Vector> newFeasibles = new Vector();

    for (int i = 0; i < feasibles.size(); ++i)
    {
        Vector<Station> elementToCheck = feasibles.elementAt(i);
        boolean found = false;

        for (Vector<Station> nf : newFeasibles)
        {
            if (this.isStationsEqual(elementToCheck, nf))
            {
                found = true;
            }
        }
        if (! found)
        {
            newFeasibles.add(elementToCheck);
        }
    }

    feasibles = newFeasibles;
}

public void fileOutput()
{
    FileOutputStream out; // declare a file output object
    PrintStream p; // declare a print stream object

    try
    {
        out = new FileOutputStream(outputFile);

        // Connect print stream to the output stream
        p = new PrintStream( out );

        p.println(this);

        p.close();
    }
    catch (Exception e)
    {
        System.err.println ("Error writing to file");
    }
}

public void htmlOutput()
{
    FileOutputStream out; // declare a file output object
    PrintStream p; // declare a print stream object
    DecimalFormat df = new DecimalFormat();

```

EK-2 (Devam) Java program kodları

```

df.setMaximumFractionDigits(2);

try
{
    //find minimum variance stations
    double minVariance = Double.MAX_VALUE;

    for (Vector v : feasibles)
    {
        if (stationAlphaVariance(v) < minVariance)
        {
            minVariance = stationAlphaVariance(v);
        }
    }

    minVariance += 0.001;

    double maxEff = Double.MIN_VALUE;

    for (Vector v : feasibles)
    {
        if (stationLineEfficiency(v) > maxEff)
        {
            maxEff = stationLineEfficiency(v);
        }
    }

    maxEff -= 0.001;

    double maxAvg = Double.MIN_VALUE;

    for (Vector v : feasibles)
    {
        if (this.stationAvgAlpha(v) > maxAvg)
        {
            maxAvg = stationAvgAlpha(v);
        }
    }

    maxAvg -= 0.001;

    out = new FileOutputStream("fuzzy.html");

    // Connect print stream to the output stream
    p = new PrintStream( out );

    p.println("<html>");
    p.println("<head><title>Atama Sonuclari</title></head>");
    p.println("<body>");
    p.println("<center>");

    p.println("<h2>Gorevler</h2>");
    p.println("<table border=\"1\">");

```

EK-2 (Devam) Java program kodları

```

p.println("<tr
bgcolor='snow'><td>#</td><td>Gorev</td><td>Bulanik Islem
Fonksiyonu</td><td>Ardillar</td></tr>");
int i = 1;
for (Node n : this.nodes)
{
    p.println("<tr><td>"+(i++)+"</td>");
    p.println("<td>" + n.name + "</td>");
    p.println("<td>" + n.alpha + "&alpha;+" + n.constant + "</td>");
    p.println("<td>");
    for (Node s : n.successors)
    {
        p.println(s.name);
    }
    p.println("</td></tr>");
}
p.println("</table>");

p.println("<h2>Atama Sonuclari</h2>");
p.println("<table border='1'>");

p.println("<tr><td>#</td>");

i = 1;
int snum = feasibles.elementAt(0).size();

for (i = 0; i < snum; i++)
{
    p.println("<td><span style='color:blue'> Istasyon " + (i+1) + "
</span></td>");
    p.println("<td><span style='color:red'> &alpha; </span></td>");
}

p.println("<td>Var</td>");

p.println("<td>Ort(&alpha;)</td>");
p.println("<td>E</td>");
p.println("</tr>");

i = 1;
for (Vector v : feasibles)
{
    if (stationLineEfficiency(v) > maxEff)
    {
        p.println("<tr bgcolor='greenyellow'>");
    }
    else
    {
        if (stationAvgAlpha(v) > maxAvg)
        {
            p.println("<tr bgcolor='lightblue'>");
        }
        else
        {
            if (stationAlphaVariance(v) < minVariance)

```


EK-2 (Devam) Java program kodları

```

        {
            p.println("<tr bgcolor='yellow'>");
        }
        else
        {
            p.println("<tr>");
        }

p.println("<td>" + (i++) + "</td>");

for (Object o : v)
{
    Station s = (Station) o;
    p.println("<td><span style='color:blue'> ");
    for (Node n : s.nodes)
    {
        p.println(n.name + " ");
    }
    p.println(" </span> </td>");
    p.println("<td><span style='color:red'> " +
df.format(s.alpha(CYCLETIME)) + "</span></td>");

    }
    p.println("<td>" + df.format(stationAlphaVariance(v)) + "</td>");

    p.println("<td>" + df.format(stationAvgAlpha(v)) + "</td>");
    p.println("<td>" + df.format(stationLineEfficiency(v)) + "</td>");
    p.println(" </tr> ");

}
p.println(" </table></center></body></html>");

p.close();
}
catch (Exception e)
{
    System.err.println ("Error writing to file");
}

}

/**
 * @param args the command line arguments
 */
public static void main(String[] args)
{
    long beginTime = System.currentTimeMillis();
    Main m = new Main();
    m.createNodes();
    m.createArcs();
    m.loadCycleTime();
    m.buildPermutableGroups();
    m.buildAllAlternatives();
    m.findBestConf();
}

```

EK-2 (Devam) Java program kodları

```

//System.out.print(m);
m.filterFeasibles();
m.fileOutput();
m.htmlOutput();
System.out.print(m);
long endTime = System.currentTimeMillis();

System.out.println("Total Millis: " + (endTime - beginTime));

/*Permuter p = new Permuter();
Vector<Vector> vall = p.getAllPermutations(m.nodes);

for (Vector v : vall )
{
    System.out.println("\n\nNEW PERMUTATION\n");
    for (Object n : v)
    {
        System.out.println(((Node) n));
    }
}
*/

/* Vector<Long> vl = m.getStationDelimiters(30, 3);

for (Long l : vl)
{
    System.out.println(l);
}
*/

}

}
Node.java
/*
 * Node.java
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication1;

/**
 *
 * @author iesahin
 */

import java.util.Vector;

public class Node
{

```

EK-2 (Devam) Java program kodları

```

    public String name;
    public Vector<Node> successors = new Vector();
    public Vector<Node> predecessors = new Vector();
    public double alpha;
    public double constant;
    /** Creates a new instance of Node */
    public Node()
    {

    }

    public String toString()
    {
        return "Task: " + name + " (" + alpha + "A+" + constant + ") ";
    }

    public String relations()
    {
        String str = this.toString();
        str += "\nPREDECESSORS\n";

        for (Node n : this.predecessors)
        {
            str += "\t" + n.toString();
        }

        str += "\nSUCCESSORS\n";
        for (Node n : this.successors)
        {
            str += "\t" + n.toString();
        }

        return str;
    }
}

PermutableGroup.Java
/*
 * PermutableGroup.java
 *
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication1;

import java.util.Vector;

/**
 *
 * @author iesahin

```

EK-2 (Devam) Java program kodları

```

*/
public class PermutableGroup
{
    public Vector<Node> nodes = new Vector();

    /** Creates a new instance of PermutableGroup */
    public PermutableGroup()
    {

    }

    public Vector<Vector> getPermutations()
    {
        Permuter p = new Permuter();
        return p.getAllPermutations(this.nodes);
    }

    public String toString()
    {
        String str = "PERMUTABLE GROUP\n";

        for (Node n : nodes)
        {
            str += n.toString() + "\n";
        }

        return str;
    }
}

```

Permuter.java

```

/*
 * Permuter.java
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package javaapplication1;

import java.util.Vector;
/**
 *
 * @author iesahin
 */
public class Permuter
{

    /** Creates a new instance of Permuter */
    public Permuter()

```

EK-2 (Devam) Java program kodları

```

{
    }

    public Vector<Vector> getAllPermutations(Vector v)
    {
        int maxn = v.size();

        Vector<Vector> permutations = new Vector();
        //initial permutation has single element
        Vector vinit = new Vector();
        vinit.add(v.firstElement());
        permutations.add(vinit);
        Vector<Vector> nextPerm = new Vector();

        for (int i = 1; i < maxn; ++i)
        {
            Object elementToAdd = v.elementAt(i);

            //build i+1 copies of current permutations vector
            for (Vector nextp : permutations)
            {
                for (int k = 0; k < nextp.size() + 1; k++)
                {
                    Vector nextpClone = (Vector) nextp.clone();
                    nextpClone.add(k, elementToAdd);
                    nextPerm.add(nextpClone);
                }
            }

            permutations = nextPerm;
            nextPerm = new Vector();
        }

        return permutations;
    }
}

```

Station.java

```

/*
 * Station.java
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

```

```

package javaapplication1;

```

```

/**
 *
 * @author iesahin
 */

```

EK-2 (Devam) Java program kodları

```

import java.text.DecimalFormat;
import java.util.Vector;

public class Station
{
    Vector<Node> nodes = new Vector();
    double totalAlpha;
    double CYCLETIME = 0; // used only in toString() Method

    /** Creates a new instance of Station */
    public Station()
    {

    }

    public double getTotalConstant()
    {
        double sum = 0;
        for (Node n : nodes)
        {
            sum += n.constant;
        }

        return sum;
    }

    public double getTotalAlpha()
    {
        double sum = 0;
        for (Node n : nodes)
        {
            sum += n.alpha;
        }

        return sum;
    }

    public boolean equals(Object o)
    {
        if (o.getClass() != this.getClass())
            return false;

        Station s = (Station) o;

        if ((s.nodes.size() == nodes.size()) && s.nodes.containsAll(this.nodes))
            return true;

        return false;
    }

    public double alpha(double cycleTime)
    {

```

EK-2 (Devam) Java program kodları

```

        this.CYCLETIME = cycleTime;

        double rawAlpha = (cycleTime - this.getTotalConstant()) / this.getTotalAlpha();

        if (rawAlpha > 1)
        {
            return 1;
        }
        return rawAlpha;
    }

    public String toString()
    {
        DecimalFormat df = new DecimalFormat();
        df.setMaximumFractionDigits(2);
        String str = "STATION:\n";
        str += "TCONST:" + this.getTotalConstant() + "\n";
        str += "ALPHA=" + df.format(this.alpha(CYCLETIME)) + "\n";
        for (Node n: nodes)
        {
            str += n.toString() + "\n";
        }
        return str;
    }
}

```

EK-3 Atama sonuçları

Görevler

#	Görev	Bulanık İşlem Fonksiyonu	Ardıllar
1	G1	$0.5\alpha+3.25$	G2
2	G2	$1.0\alpha+7.25$	G7
3	G3	$2.0\alpha+34.5$	G4
4	G4	$3.75\alpha+17.25$	G8
5	G5	$1.5\alpha+6.0$	G6
6	G6	$2.0\alpha+9.0$	G13
7	G7	$0.5\alpha+1.25$	G9
8	G8	$0.75\alpha+4.0$	G9
9	G9	$0.25\alpha+0.25$	G10
10	G10	$0.25\alpha+0.5$	G11
11	G11	$0.25\alpha+1.0$	G12
12	G12	$0.25\alpha+0.25$	G18 G20
13	G13	$1.0\alpha+2.0$	G14
14	G14	$0.25\alpha+0.25$	G15
15	G15	$0.25\alpha+0.5$	G16
16	G16	$0.25\alpha+1.0$	G17
17	G17	$0.25\alpha+0.5$	G18 G20
18	G18	$1.0\alpha+3.5$	G19
19	G19	$7.0\alpha+42.0$	G21
20	G20	$0.5\alpha+2.0$	G21
21	G21	$0.25\alpha+0.25$	G22
22	G22	$2.5\alpha+11.75$	G23
23	G23	$0.5\alpha+1.5$	G24
24	G24	$0.75\alpha+0.75$	

EK-3 (Devam) Atama sonuçları

Atama Sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
1	G5 G3	1	G1 G6 G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,25	0,75	0,79
2	G5 G3 G1	1	G6 G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,25	0,75	0,79
3	G5 G3 G1 G6	0,04	G4 G2 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,32	0,51	0,76
4	G5 G3	1	G1 G6 G4 G2 G13 G8 G7	0,95	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,19	0,77	0,79
5	G5 G3 G1	1	G6 G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,19	0,78	0,8
6	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,28	0,54	0,77
7	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14	0,9	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,17	0,76	0,79
8	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,18	0,79	0,8
9	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,28	0,55	0,77
10	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9	0,85	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,16	0,76	0,79
11	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,17	0,79	0,8
12	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,27	0,55	0,77
13	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,78	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,13	0,75	0,79
14	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,15	0,81	0,8
15	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,25	0,57	0,78
16	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,71	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,11	0,75	0,79
17	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,13	0,82	0,81

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
18	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,24	0,58	0,78
19	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,6	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,09	0,75	0,79
20	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,95	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,08	0,84	0,81
21	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	1	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,22	0,61	0,79
22	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,5	G17 G12 G20 G18 G19	0,53	G21 G22 G23 G24	1	0,08	0,76	0,79
23	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,83	G17 G12 G20 G18 G19	0,53	G21 G22 G23 G24	1	0,05	0,84	0,81
24	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	1	G17 G12 G20 G18 G19	0,53	G21 G22 G23 G24	1	0,21	0,64	0,79
25	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,44	G12 G20 G18 G19	0,6	G21 G22 G23 G24	1	0,08	0,76	0,79
26	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,77	G12 G20 G18 G19	0,6	G21 G22 G23 G24	1	0,04	0,84	0,81
27	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	1	G12 G20 G18 G19	0,6	G21 G22 G23 G24	1	0,21	0,66	0,8
28	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,41	G20 G18 G19	0,65	G21 G22 G23 G24	1	0,08	0,77	0,79
29	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,73	G20 G18 G19	0,65	G21 G22 G23 G24	1	0,03	0,84	0,81
30	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	1	G20 G18 G19	0,65	G21 G22 G23 G24	1	0,2	0,67	0,8
31	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0,23	G18 G19	0,94	G21 G22 G23 G24	1	0,14	0,79	0,79
32	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0,52	G18 G19	0,94	G21 G22 G23 G24	1	0,05	0,86	0,81
33	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	1	G18 G19	0,94	G21 G22 G23 G24	1	0,22	0,74	0,81
34	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	0,2	G19	1	G21 G22 G23 G24	1	0,16	0,8	0,79

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
35	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	1	G19	1	G21 G22 G23 G24	1	0,23	0,76	0,81
36	G5 G3	1	G1 G6 G4 G2 G13 G8 G7	0,95	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,2	0,76	0,79
37	G5 G3 G1	1	G6 G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,21	0,77	0,79
38	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,29	0,53	0,77
39	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14	0,9	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,18	0,75	0,79
40	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,19	0,78	0,8
41	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,28	0,54	0,77
42	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9	0,85	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,17	0,75	0,79
43	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,18	0,79	0,8
44	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,28	0,55	0,77
45	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,78	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,14	0,75	0,79
46	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,16	0,8	0,8
47	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,26	0,56	0,77
48	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,71	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,12	0,74	0,79
49	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,14	0,81	0,81
50	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,25	0,57	0,78
51	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,6	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,1	0,74	0,79
52	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,95	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,1	0,83	0,81

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
53	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	1	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,23	0,6	0,78
54	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,5	G17 G12 G20 G18 G19 G21	0,49	G22 G23 G24	1	0,09	0,75	0,79
55	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,83	G17 G12 G20 G18 G19 G21	0,49	G22 G23 G24	1	0,06	0,83	0,81
56	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	1	G17 G12 G20 G18 G19 G21	0,49	G22 G23 G24	1	0,21	0,63	0,79
57	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,44	G12 G20 G18 G19 G21	0,56	G22 G23 G24	1	0,09	0,75	0,79
58	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,77	G12 G20 G18 G19 G21	0,56	G22 G23 G24	1	0,05	0,83	0,81
59	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	1	G12 G20 G18 G19 G21	0,56	G22 G23 G24	1	0,21	0,65	0,79
60	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,41	G20 G18 G19 G21	0,6	G22 G23 G24	1	0,09	0,75	0,79
61	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,73	G20 G18 G19 G21	0,6	G22 G23 G24	1	0,04	0,83	0,81
62	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	1	G20 G18 G19 G21	0,6	G22 G23 G24	1	0,21	0,66	0,8
63	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0,23	G18 G19 G21	0,88	G22 G23 G24	1	0,14	0,78	0,79
64	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0,52	G18 G19 G21	0,88	G22 G23 G24	1	0,05	0,85	0,81
65	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	1	G18 G19 G21	0,88	G22 G23 G24	1	0,21	0,73	0,81
66	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	0,2	G19 G21	1	G22 G23 G24	1	0,16	0,8	0,79
67	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	1	G19 G21	1	G22 G23 G24	1	0,23	0,76	0,81
68	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	0,1	G20 G19	1	G21 G22 G23 G24	1	0,2	0,78	0,79
69	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	0,38	G20 G19	1	G21 G22 G23 G24	1	0,1	0,84	0,8

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
70	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19	1	G21 G22 G23 G24	1	0,23	0,76	0,81
71	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	0,1	G20 G19 G21	1	G22 G23 G24	1	0,2	0,78	0,79
72	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	0,38	G20 G19 G21	1	G22 G23 G24	1	0,1	0,84	0,8
73	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19 G21	1	G22 G23 G24	1	0,23	0,76	0,81
74	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,47	G17 G20 G18 G19	0,57	G21 G22 G23 G24	1	0,08	0,76	0,79
75	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,79	G17 G20 G18 G19	0,57	G21 G22 G23 G24	1	0,04	0,84	0,81
76	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	1	G17 G20 G18 G19	0,57	G21 G22 G23 G24	1	0,21	0,65	0,79
77	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,47	G17 G20 G18 G19 G21	0,53	G22 G23 G24	1	0,08	0,75	0,79
78	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,79	G17 G20 G18 G19 G21	0,53	G22 G23 G24	1	0,05	0,83	0,81
79	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	1	G17 G20 G18 G19 G21	0,53	G22 G23 G24	1	0,21	0,64	0,79
80	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,6	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,09	0,75	0,79
81	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,95	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,08	0,84	0,81
82	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	1	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,22	0,61	0,79
83	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,6	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,1	0,74	0,79
84	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,95	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,1	0,83	0,81
85	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	1	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,23	0,6	0,78
86	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,78	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,13	0,75	0,79
87	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,15	0,81	0,8
88	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,25	0,57	0,78

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
89	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,78	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,14	0,75	0,79
90	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,16	0,8	0,8
91	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,26	0,56	0,77
92	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G9	0,9	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,17	0,76	0,79
93	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,18	0,79	0,8
94	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,28	0,55	0,77
95	G5 G3	1	G1 G6 G4 G2 G13 G8 G7 G9	0,9	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,18	0,75	0,79
96	G5 G3 G1	1	G6 G4 G2 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,19	0,78	0,8
97	G5 G3 G1 G6	0,04	G4 G2 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,28	0,54	0,77
98	G5 G3 G1 G2	0,4	G6 G4 G13 G8	1	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,24	0,6	0,77
99	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,2	0,63	0,78
100	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,19	0,64	0,78
101	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,18	0,64	0,79
102	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,16	0,66	0,79
103	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,15	0,67	0,79
104	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16	1	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,12	0,7	0,8
105	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11	1	G17 G12 G20 G18 G19	0,53	G21 G22 G23 G24	1	0,1	0,73	0,81

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
106	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	1	G12 G20 G18 G19	0,6	G21 G22 G23 G24	1	0,09	0,75	0,81
107	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	1	G20 G18 G19	0,65	G21 G22 G23 G24	1	0,09	0,76	0,81
108	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	1	G18 G19	0,94	G21 G22 G23 G24	1	0,08	0,83	0,82
109	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	0,85	G19	1	G21 G22 G23 G24	1	0,08	0,81	0,82
110	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7	1	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,2	0,62	0,78
111	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14	1	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,2	0,63	0,78
112	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9	1	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,19	0,64	0,78
113	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15	1	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,17	0,65	0,79
114	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10	1	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,15	0,66	0,79
115	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16	1	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,13	0,69	0,8
116	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11	1	G17 G12 G20 G18 G19 G21	0,49	G22 G23 G24	1	0,1	0,72	0,8
117	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	1	G12 G20 G18 G19 G21	0,56	G22 G23 G24	1	0,09	0,74	0,81
118	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	1	G20 G18 G19 G21	0,6	G22 G23 G24	1	0,09	0,75	0,81
119	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	1	G18 G19 G21	0,88	G22 G23 G24	1	0,08	0,82	0,82
120	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18	0,85	G19 G21	1	G22 G23 G24	1	0,08	0,81	0,82
121	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19	1	G21 G22 G23 G24	1	0,09	0,85	0,83

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
122	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G18	1	G20 G19 G21	1	G22 G23 G24	1	0,09	0,85	0,83
123	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	1	G17 G20 G18 G19	0,57	G21 G22 G23 G24	1	0,09	0,74	0,81
124	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	1	G17 G20 G18 G19 G21	0,53	G22 G23 G24	1	0,1	0,73	0,81
125	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G11	1	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,12	0,7	0,8
126	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G15 G10 G11	1	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,13	0,69	0,8
127	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,16	0,66	0,79
128	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G14 G9 G10	1	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,17	0,65	0,79
129	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,19	0,64	0,78
130	G5 G3 G1 G2	0,4	G6 G4 G13 G8 G7 G9	1	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,2	0,63	0,78
131	G3	1	G5 G1 G6 G4 G2 G13 G8	0,4	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,24	0,6	0,76
132	G3	1	G5 G1 G6 G4 G2 G13 G8 G7	0,27	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,22	0,6	0,76
133	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14	0,24	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,22	0,6	0,76
134	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9	0,22	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,22	0,6	0,76
135	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,17	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,21	0,6	0,76
136	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,12	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,21	0,6	0,76
137	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,04	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,22	0,61	0,76
138	G3	1	G5 G1 G6 G4 G2 G13 G8 G7	0,27	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,23	0,59	0,76

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
139	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14	0,24	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,23	0,59	0,76
140	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9	0,22	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,22	0,59	0,76
141	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,17	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,22	0,59	0,76
142	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,12	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,22	0,6	0,76
143	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,04	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,23	0,6	0,76
144	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,04	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,22	0,61	0,76
145	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,04	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,23	0,6	0,76
146	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,17	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,21	0,6	0,76
147	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,17	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,22	0,59	0,76
148	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G9	0,24	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,22	0,6	0,76
149	G3	1	G5 G1 G6 G4 G2 G13 G8 G7 G9	0,24	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,23	0,59	0,76
150	G3 G1	1	G5 G6 G4 G2 G13 G8	0,75	G7 G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0	G21 G22 G23 G24	1	0,22	0,69	0,78
151	G3 G1	1	G5 G6 G4 G2 G13 G8 G7	0,6	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,12	G21 G22 G23 G24	1	0,18	0,68	0,78
152	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14	0,56	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,17	0,68	0,78
153	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9	0,52	G15 G10 G16 G11 G17 G12 G20 G18 G19	0,18	G21 G22 G23 G24	1	0,16	0,67	0,78
154	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,47	G10 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,15	0,67	0,78
155	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,41	G16 G11 G17 G12 G20 G18 G19	0,29	G21 G22 G23 G24	1	0,14	0,68	0,78

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
156	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,32	G11 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,14	0,68	0,78
157	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,23	G17 G12 G20 G18 G19	0,53	G21 G22 G23 G24	1	0,14	0,69	0,78
158	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,18	G12 G20 G18 G19	0,6	G21 G22 G23 G24	1	0,15	0,7	0,78
159	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,16	G20 G18 G19	0,65	G21 G22 G23 G24	1	0,16	0,7	0,78
160	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0	G18 G19	0,94	G21 G22 G23 G24	1	0,24	0,73	0,78
161	G3 G1	1	G5 G6 G4 G2 G13 G8 G7	0,6	G14 G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,09	G22 G23 G24	1	0,19	0,67	0,77
162	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14	0,56	G9 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,18	0,67	0,77
163	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9	0,52	G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,15	G22 G23 G24	1	0,17	0,67	0,77
164	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15	0,47	G10 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,16	0,67	0,77
165	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10	0,41	G16 G11 G17 G12 G20 G18 G19 G21	0,26	G22 G23 G24	1	0,15	0,67	0,77
166	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16	0,32	G11 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,14	0,67	0,77
167	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11	0,23	G17 G12 G20 G18 G19 G21	0,49	G22 G23 G24	1	0,15	0,68	0,77
168	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17	0,18	G12 G20 G18 G19 G21	0,56	G22 G23 G24	1	0,16	0,68	0,77
169	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12	0,16	G20 G18 G19 G21	0,6	G22 G23 G24	1	0,16	0,69	0,77
170	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G17 G12 G20	0	G18 G19 G21	0,88	G22 G23 G24	1	0,23	0,72	0,77
171	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,2	G17 G20 G18 G19	0,57	G21 G22 G23 G24	1	0,15	0,69	0,78

EK-3 (Devam) Atama sonuçları

#	Istasyon 1	α	Istasyon 2	α	Istasyon 3	α	Istasyon 4	α	Var	Ort(α)	E
172	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G16 G11 G12	0,2	G17 G20 G18 G19 G21	0,53	G22 G23 G24	1	0,15	0,68	0,77
173	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,32	G16 G17 G12 G20 G18 G19	0,41	G21 G22 G23 G24	1	0,14	0,68	0,78
174	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G15 G10 G11	0,32	G16 G17 G12 G20 G18 G19 G21	0,37	G22 G23 G24	1	0,14	0,67	0,77
175	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,47	G15 G16 G11 G17 G12 G20 G18 G19	0,23	G21 G22 G23 G24	1	0,15	0,67	0,78
176	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G14 G9 G10	0,47	G15 G16 G11 G17 G12 G20 G18 G19 G21	0,2	G22 G23 G24	1	0,16	0,67	0,77
177	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G9	0,56	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19	0,15	G21 G22 G23 G24	1	0,17	0,68	0,78
178	G3 G1	1	G5 G6 G4 G2 G13 G8 G7 G9	0,56	G14 G15 G10 G16 G11 G17 G12 G20 G18 G19 G21	0,12	G22 G23 G24	1	0,18	0,67	0,77

Not : Yeşil işaretli alternatifler, en yüksek etkinlik değerine sahip alternatifleri, mavi olan alternatifler, ortalama α değeri en yüksek olan alternatifleri, sarı işaretli alternatifler ise istasyonlar arasındaki α değerlerinin varyansı en küçük olan alternatifleri göstermektedir.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : YILMAZ, Murat Mustafa
Uyruğu : T.C.
Doğum tarihi ve yeri : 13.05.1978 Ankara
Medeni hali : Bekar
Telefon : 0 (555) 566 24 03
e-mail : muratmyilmaz@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi/ Endüstri Müh.	2001
Lise	Mimar Kemal Lisesi	1995

İş Deneyimi

Yıl	Yer	Görev
2003-2006	MAN Türkiye A.Ş.	İş Hazırlama Mühendisi

Yabancı Dil

İngilizce

Hobiler

Futbol, Bilgisayar programlama

