İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BASİT U-TİPİ MONTAJ HATTI DENGELEMEDE ANALİTİK YÖNTEMLERİN KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

A. Elvan BAYRAKTAROĞLU

Anabilim Dalı: ENDÜSTRİ MÜHENDİSLİĞİ

Programı: ENDÜSTRİ MÜHENDİSLİĞİ

HAZİRAN 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BASİT U-TİPİ MONTAJ HATTI DENGELEMEDE ANALİTİK YÖNTEMLERİN KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

A. Elvan BAYRAKTAROĞLU

507031129

Tezin Enstitüye Verildiği Tarih : 7 Mayıs 2007

Tezin Savunulduğu Tarih : 11 Haziran 2007

Tez Danışmanı : Y.Doç.Dr. Murat BASKAK

Diğer Jüri Üyeleri : Prof.Dr. M. Bülent DURMUŞOĞLU (İ.T.Ü.)

Prof.Dr. Mehmet TANYAŞ (Okan Ünv.)

HAZİRAN 2007

ÖNSÖZ

20. yüzyılın başlarında, daha hızlı, ucuz ve yüksek miktarlarda üretim yapabilmek amacıyla kullanılmaya başlanan montaj hatları, bugün serî üretim sistemlerinin en temel öğesi durumuna gelmiştir. Rekâbeti sürdürebilmek açısından, kaynakların etkin kullanımının kritik önem taşıdığı günümüzde, montaj hatlarının en iyi şekilde dengelenmesi yüksek kapasite kullanımını da beraberinde getirir.

Tam Zamanında Üretim felsefesinin yaygınlaşması ile kullanım ortamı bulan U-tipi montaj hatlarının dengelenmesinde, varolan montaj hattı dengeleme problemlerinin çözümünde kullanılan yöntemler, belirli değişiklikler yapılmak kaydıyla uygulanabilmektedir.

Bu çalışmada, montaj hattı dengeleme çalışmalarının temelini oluşturan basit montaj hattı dengeleme probleminin çözümünde kullanılan analitik yöntemlerden -özellikle- dal-sınır algoritmasının, basit U-tipi montaj hattı dengeleme problemine uygulanması ele alınmıştır.

Çalışmalarım sırasında önerileri ile yol gösterip, yardımlarını esirgemeyen danışman hocam Sayın Y.Doç.Dr. Murat BASKAK'a, program yazımı sürecinde günlerini bana ayırarak çok büyük yardımda bulunan arkadaşım Murat Engin ÜNAL'a, bu süreçte manevî destekleri ile yanımda olan aileme ve bilgilerini benimle paylaşan arkadaşım Gülçin YÜCEL'e teşekkür ederim.

Mayıs 2007 A. Elvan Bayraktaroğlu

İÇİNDEKİLER

KISALTMALAR **v**
TABLO LİSTESİ **vi**
ŞEKİL LİSTESİ **vii**
SEMBOL LİSTESİ **viii**
ÖZET **ix**
SUMMARY **xi**

1. GİRİŞ **1**

2. ÜRETİM VE ÜRETİM SİSTEMLERİ **3**
2.1. Üretim 3
2.2. Üretim Sistemlerinin Sınıflandırılması 4
2.2.1. Genel Bilgi 4
2.2.2. Geleneksel Üretim Sistemleri 4
2.2.2.1. Ürüne Uygulanan Stok Politikasına Göre Sınıflandırma 4
2.2.2.2. Ürün Çeşidi ve Üretim Miktarına Göre Sınıflandırma 5
2.2.2.3. Üretim Sürecine Göre Sınıflandırma 6
2.2.2.4. Ürün Cinsine Göre Sınıflandırma 6
2.2.3. Çağdaş Üretim Sistemleri 6
2.2.3.1. Hücresel Üretim Sistemi (Grup Teknolojisi) 6
2.2.3.2. Bilgisayarla Bütünleşik Üretim Sistemleri 7
2.2.3.3. Tam Zamanında Üretim Sistemi 7
2.2.3.4. Esnek Üretim Sistemleri 7
2.2.3.5. Optimum Üretim Teknolojisi 8
2.3. Montaj Kavramı 8
2.3.1. Montaj Kavramının Tanımı 8
2.3.2. Montaj Hatlarının Üretimdeki Yeri 8

3. MONTAJ HATLARININ DENGELENMESİ **9**
3.1. Montaj Hattı ve Hat Dengeleme Kavramı 9
3.2. Montaj Hatlarının Dengelenmesinin Amaçları 10
3.3. Montaj Hatlarının Yerleşimi 11
3.4. Montaj Hatlarının Dengelenmesinde Kullanılan Temel Kavramlar 11
3.5. Montaj Hatlarının Dengelenmesini Etkileyen Temel Etmenler ve Kısıtlar 15
3.5.1. Temel Etmenler 15
3.5.2. Kısıtlar 15
3.6. Montaj Hatlarında Model Sayısı 16
3.6.1. Tek Modelli Hatlar 16
3.6.2. Karışık Modelli Hatlar 17
3.6.3. Çok Modelli Hatlar 17

4. MONTAJ HATTI DENGELEME PROBLEMİ **18**
4.1. Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması 18
4.2. Basit Montaj Hattı Dengeleme Problemi (BMHDP) 20
4.3. Çözüm Yöntemlerinin Sınıflandırılması 21
4.3.1. Sezgisel Yöntemler 21
4.3.2. Analitik Yöntemler 21

5. BMHDP'DE UYGULANAN BAŞLICA ANALİTİK YÖNTEMLER **23**
5.1. Dinamik Programlama (DP) 23
5.2. BMHDP'de Dinamik Programlama Uygulamaları Üzerine Literatür Araştırması 23
5.3. Dal-Sınır Algoritması 29
5.3.1. Dal-Sınır Algoritması Hakkında Genel Bilgi 29
5.3.2. Dallandırma 29
5.3.3. Sınırlama 30
5.3.4. Baskınlık ve Azaltma Kuralları 35
5.4. BMHDP'de Dal-Sınır Algoritması Uygulamaları Üzerine Literatür Araştırması 37

6. U-TİPİ MONTAJ HATLARI **42**
6.1. Genel Bilgi 42
6.2. Basit U-tipi Hat Dengeleme Problemi (BUHDP) 44
6.3. BUHDP'de Uygulanan Analitik Yöntemler 45
6.3.1. BUHDP'de Dinamik Programlama Uygulamaları Üzerine Literatür Araştırması 45
6.3.2. BUHDP'de Dal-Sınır Algoritması Uygulamaları Üzerine Literatür Araştırması 48

7. ETKİN DAL-SINIR ALGORİTMALARININ BİR ÖRNEK ÜZERİNDE UYGULANMASI **50**

8. UYGULAMA **51**
8.1. Firma Bilgisi 51
8.2. Uygulama Yapılan Hattın Tanıtımı 52
8.3. Problemin EUREKA, FABLE, ULINO ve U-OPT1 ile Çözümü 55
8.4. Dengeleme Çözümlerinin Karşılaştırılması 59

9. SONUÇLAR ve TARTIŞMA **61**

KAYNAKLAR **63**

EKLER **67**

ÖZGEÇMİŞ **112**

KISALTMALAR

TZÜ : Tam Zamanında Üretim
D : Denge Kaybı
Dİ : Düzgünlük İndeksi
HE : Hat Etkinliği
KE : Kuramsal Etkinlik
MHD : Montaj Hattı Dengeleme
MHDP : Montaj hattı Dengeleme Problemi
BMHDP : Basit Montaj Hattı Dengeleme Problemi
DP : Dinamik Programlama
BUHDP : Basit U-tipi Hat Dengeleme Problemi
AS : Alt-sınır
BK : Baskınlık Kuralı
GÜS : Global Üst-sınır
YÜS : Yerel Üst-sınır
GAS : Global Alt-sınır
YAS : Yerel Alt-sınır
ysiö : Yeni Seçilmiş İş Öğesi Süresi
kis : Kullanılmamış İstasyon Süresi
gdiö : Geri Dönüş İş Öğesi

TABLO LİSTESİ

Sayfa No

Tablo 3.1 11 İş Öğesi İçin Öncelik Matrisi Örneği 14
Tablo 5.1 AS4 Hesap Tablosu 33
Tablo 8.1 Uygulama Problemi İş Öğeleri 53
Tablo 8.2 Uygulama Problemi Öncelik Matrisi 54
Tablo 8.3 ULINO için iş öğeleri doğrudan öncüllük matrisi 57
Tablo 8.4 ULINO için iş öğeleri doğrudan ardıllık matrisi 57
Tablo A.1 İş Öğelerinin p_i Değerleri 68
Tablo A.2 İş Öğelerinin n_{i1} Değerleri 68
Tablo A.3 İş Öğelerinin n_{i2} Değerleri 69
Tablo A.4 İş Öğelerinin n_{i3} Değerleri 69
Tablo A.5 İş Öğelerinin n_{i4}, n_i ve Yuvarlanmış Değerleri 69

ŞEKİL LİSTESİ

Sayfa No

Şekil 3.1 : Örnek Bir Montaj Hattı Şeması 9
Şekil 3.2 : Fiziksel Montaj Hattı Tipleri..... 12
Şekil 3.3 : 10 İş Öğesi İçin Öncelik Diyagramı Örneği 13
Şekil 4.1 : MHDP'lerin ve Çözüm Yöntemlerinin Sınıflandırılması 19
Şekil 5.1 : Öncelik Diyagramı 31
Şekil 6.1 : Doğrusal ve U-tipi Hat Yerleşimi Farkı..... 44
Şekil 8.1 : Uygulama Problemi Öncelik Diyagramı..... 54
Şekil 8.2 : ULINO İçin Öncelik Diyagramı 56
Şekil A.1 : Tip-1 BMHDP Öncelik Diyagramı 67
Şekil A.2 : Yeniden Numaralandırılma ile Elde Edilen Öncelik Diyagramı 67
Şekil B.1 : Tip-1 BMHDP Öncelik Diyagramı 86
Şekil C.1 : Tip-1 BUHDP Öncelik Diyagramı..... 95
Şekil C.2 : Yeniden Numaralandırılmış Öncelik Diyagramı 96
Şekil C.3 : İndirgenmiş Problem Öncelik Diyagramı 97
Şekil C.4 : İndirgenmiş Problem Öncelik Diyagramı 2 98
Şekil C.5 : İndirgenmiş Problem Öncelik Diyagramı 3 99
Şekil C.6 : İndirgenmiş Problem Öncelik Diyagramı 4 100
Şekil C.7 : İndirgenmiş Problem Öncelik Diyagramı 5 101
Şekil C.8 : İndirgenmiş Problem Öncelik Diyagramı 6 102
Şekil C.9 : İndirgenmiş Problem Öncelik Diyagramı 7 103
Şekil C.10 : İndirgenmiş Problem Öncelik Diyagramı 8 104
Şekil C.11 : İndirgenmiş Problem Öncelik Diyagramı 9 105
Şekil D.1 : Tip-1 BUHDP'nin Öncelik Diyagramı 106

SEMBOL LİSTESİ

$\mathbf{C}$	: Çevrim süresi
$\mathbf{C^*}$	: Ortalama çevrim süresi
$\mathbf{n}$	: İş öğesi sayısı
$\mathbf{N}$	: İstasyon sayısı
$\mathbf{N_{enaz}}$	: Gerekli en az istasyon sayısı
$\mathbf{t_i}$	: i iş öğesinin işlem süresi
$\mathbf{T_i}$	: i istasyonunun işlem süresi
$\mathbf{T_{enb}}$	: En büyük istasyon süresi
$\mathbf{s}$	: Olurlu küme veya durum
$\mathbf{c_a}$	: Atanmış a dizisi için toplam istasyon süresi/mâliyeti
$\mathbf{R^f(s^f)}$	: s ileri olurlu kümesindeki elemanlardan, s olurlu kümesi içerisinde ardılı olmayan elemanların oluşturduğu küme
$\mathbf{R^b(s^b)}$	: s geri olurlu kümesindeki elemanlardan, s olurlu kümesi içerisinde öncülü olmayan elemanların oluşturduğu küme
$\tau(s)$	: s olurlu kümesindeki iş öğelerinin toplam süresi
$\mathbf{[x]^+}$	: x'e eşit veya x'ten büyük en küçük tamsayı
$\mathbf{L(s)}$	: s olurlu kümesinin etiketi
$\mathbf{L(i)}$	: i iş öğesinin etiketi
$\mathbf{l(s)}$	: s olurlu kümesinin elemanı olmayan iş öğelerinin toplam süresi
$\mathbf{F(s)}$	: s olurlu kümesinin en kısa yol çözümü
$\mathbf{p_i}$	: i iş öğesinin istasyon gereksinimi
$\mathbf{P_i}$	: i iş öğesinin bitişik öncüllerinin kümesi
$\mathbf{P_i^*}$	: i iş öğesinin bitişik ve geçişli öncüllerinin kümesi
$\mathbf{F_i}$	: i iş öğesinin bitişik ardıllarının kümesi
$\mathbf{F_i^*}$	: i iş öğesinin bitişik ve geçişli ardıllarının kümesi
$\mathbf{Sx(x)}$	: x istasyonunun (x alternatifinin) yükü
$\mathbf{A_i}$	: i iş öğesinin atandığı istasyonun numarasına eşit olan bir tamsayı değişkeni ($A_i = 0$ ise, i iş öğesi herhangi bir istasyona atanmamıştır.)
$\mathbf{W_j}$	: j istasyonuna atanmış iş öğelerinin kümesi
$\mathbf{H}$	: Bulunmuş sezgisel çözümün istasyon sayısı (Sezgisel çözümün olmadığı durumlarda H = n)
$\mathbf{E_i}$	: i iş öğesinin en erken atanabileceği istasyon numarası
$\mathbf{L_i}$	: i iş öğesinin en geç atanabileceği istasyon numarası
$\mathbf{M}$	: Tüm iş öğelerinin oluşturduğu küme

BASİT U-TİPİ MONTAJ HATTI DENGELEMEDE ANALİTİK YÖNTEMLERİN KARŞILAŞTIRILMASI

ÖZET

Endüstrileşme sürecinde daha çok miktarda, hızlı ve ucuz üretim yapabilmek için uygulanmaya başlanan montaj hatları, günümüzde serî üretim sistemlerinin en temel öğelerinden biridir. Rekâbet açısından eldeki kaynakların en iyi şekilde değerlendirilmesinin bir zorunluluk olduğu günümüzde, montaj hatlarının en iyi şekilde dengelenmesi, işletmelerin kapasitelerini etkin kullanabilmeleri açısından kritik önemdedir. Bu nedenle, basit montaj hatlarının dengelenmesi problemi için bilinen bir çevrim süresi dahilinde en iyi çözümü bulmak amacıyla kullanılan analitik yöntemler incelenmiştir. Özellikle dinamik programlama ve dal-sınır algoritması, bu anlamda en yaygın olarak kullanılan yöntemler olarak öne çıkmaktadırlar. Bu çalışmada, montaj hattı dengeleme problemi için literatürde bulunan dinamik programlama ve dal-sınır algoritmasına yönelik çalışmalar incelenmiş, etkin çözüm veren yöntemlerin hangileri olduğu araştırılmıştır. Literatür araştırması sonucunda dal-sınır algoritmasının, işlem yükü ve hesaplama süresi açısından dinamik programlamaya kıyasla daha üstün olduğu görülmüştür.

Son yıllarda Tam Zamanında Üretim felsefesinin kabul görmesi ile birlikte U-tipi montaj hatlarının kullanımı yaygınlaşmıştır. "Basit U-tipi Montaj Hattı Dengelemede Analitik Yöntemlerin Karşılaştırılması" adlı bu çalışmada, U-tipi montaj hatlarının fiziksel yapılarından ve bunların, klasik düz tip (I-tipi) montaj hatlarına olan üstünlüklerinden de sözedilmiş ve I-tipi montaj hatlarının dengelenmesinde kullanılan analitik yöntemlerin, U-tipi montaj hatlarının dengelenmesinde hangi değişiklerle kullanılabilecekleri de incelenmiştir. Hem basit montaj hattı dengeleme problemi, hem de basit U-tipi montaj hattı dengeleme problemi için en etkin çalışan algoritmalar belirlenmiş ve bu algoritmalar birer örnek problem üzerinde gösterilmiştir. Son olarak uygulama aşamasında, endüstriyel yaşamdan alınan bir problem bu algoritmalar ile çözülmüş, sonuçlar

karşılaştırılmıştır. Seçilen algoritmaların uygulama problemi için en iyi sonucu verdiği ve problem U-tipi montaj hattı olarak dengelendiğinde dengenin, I-tipi montaj hattına kıyasla, daha az sayıda istasyonla sağlandığı görülmüştür.

Anahtar kelimeler: Montaj hattı dengeleme, basit U-tipi hat dengeleme problemi, dinamik programlama, dal-sınır algoritması

COMPARING ANALYTICAL METHODS FOR SIMPLE U-LINE LINE BALANCING PROBLEM

SUMMARY

Recently, assembly lines which has been started being used for faster, cheaper and more production during industrialization process are one of the leading elements of mass production systems. Balancing assembly lines perfectly is very critical as a matter of effective capacity usage of enterprises because assessing the available resources in a best way is a requirement today. Therefore, in this study, analytical methods which are used to determine the best solution within a known cycle time has been examined for the problem of balancing simple assembly lines. Especially, dynamic programming and branch-and-bound algorithm are the most common methods in this area. First, literature studies on dynamic programming and branch-and-bound algorithm applications for balancing assembly line problem has been examined and the methods which provides the best solution has been researched. As a conclusion of literature studies, it has been determined that branch-and-bound algorithm is better than dynamic programming in terms of memory requirement and calculation time.

In recent years, U-type assembly lines usage had been increased due to the acceptance of Just-in-Time Production philosophy. Also in this study called "Comparing Analytical Methods for Simple U-Line Line Balancing Problem", U-type assembly lines' set up and its advantages versus classic I-type assembly lines has been mentioned and which alterations must be applied to the I-type assembly line balancing methods for balancing U-type assembly lines has been examined. The most effective algorithms which fit both basic assembly lines balancing problem and U-type assembly lines balancing problem have been determined and performed on one each problem. At the end, these algorithms have been performed on a problem which is taken from the industrial life and the results of the applications have been compared to each other. It has been seen that the used

algorithms helped to find the optimum solution for the problem and that solving the problem as an U-type assembly line led to a better solution in the meaning of station number compared to the classic I-type assembly line.

Keywords: Assembly line balancing, simple U-line line balancing problem, dynamic programming, branch-and-bound algorithm

1. GİRİŞ

Montaj hatları ilk defa 20. yüzyılın başlarında A.B.D.'de Ford firması tarafından kullanılmıştır. Üretim talebinin yıldan yıla artmasıyla birlikte montaj hatları, yüksek miktarlarda, ucuz, istenen kalitede ve hızlı ürün üretmesi hedeflenen serî üretim sistemlerinin vazgeçilmez bir parçası durumuna gelmiştir.

Montaj hatlarında ürünlerin montajı, malzeme ve yarı ürünlerin gerekli işlemler altında biraraya getirilmesi ile gerçekleştirilir. Bu işlemleri gerçekleştirebilmek için hat boyunca dizilmiş işçilere yâni iş istasyonlarına gereksinim vardır. Hattın bir ucundan montajı başlanan ürün, hat boyunca ilerleyerek her istasyonda çeşitli işlemler görür ve son istasyondaki işlemin bitmesi ile birlikte hattı nihaî ürün olarak terkeder.

Günümüzdeki rekâbet ortamında firmaların ayakta kalabilmesinin ilk koşulu, kaynaklarını akılcı değerlendirmeleridir. Bu nedenle firmaların varolan kapasitelerini en iyi şekilde kullanmaları gerekmektedir. Montaj hatlarının dengelenmesinin önemi de burada ortaya çıkar. İstasyonların boşta bekledikleri sürenin enazlanması, montaj hattının etkin kullanımı için esastır. İş öğelerinin bu amaçla, çeşitli kısıtlar dahilinde istasyonlara atanmasına "montaj hattı dengeleme problemi" denir.

Montaj hattı dengeleme problemleri, üretilen model sayısına, iş öğesi sürelerinin niteliğine bağlı olarak sınıflandırılabilir. Literatürde en çok işlenmiş olan, deterministik iş öğesi sürelerinin sözkonusu olduğu tek modelli montaj hattı dengeleme problemleridir.

Son yıllarda Tam Zamanında Üretim felsefesinin yaygınlaşmasıyla U-tipi montaj hatlarının kullanım oranı artmaktadır. U-tipi montaj hatlarının klasik I-tipi montaj hatlarına karşı sosyal ve ekonomik üstünlüklere sahip olması, popülerliklerini daha da arttırmaktadır.

Bu çalışmada, sonraki bölümde, üretim sistemlerinden ve montaj hatlarının üretim sistemleri içerisindeki yerinden sözedilecektir. Üçüncü bölümde, montaj hatlarının

dengelenmesindeki amaçlar, montaj hatlarında kullanılan kavramlar ve dengelemede karşılaşılan kısıtlar ele alınacaktır. Dördüncü ve beşinci bölümde, basit montaj hattı dengeleme probleminden, analitik çözüm yöntemlerinden olan dinamik programlama ve dal-sınır algoritmasının basit montaj hattı dengeleme problemine uygulanmasından ve daha önce bu konu üzerinde yapılmış çalışmalardan sözedilecektir. Altıncı bölümde, son yıllarda kullanımı yaygınlaşan U-tipi montaj hatlarının özelliklerinden, doğrusal montaj hatlarına karşı olan üstünlüklerinden ve U-tipi montaj hatlarında dengeleme için kullanılan dinamik programlama ve dal-sınır algoritmalarından sözedilecektir. Yedinci bölümde, analitik yöntemler arasında en etkin olan yaklaşımlar, birer örnek üzerinde gösterilecektir. Sekizinci ve dokuzuncu bölümlerde ise endüstriyel yaşamdan alınan bir uygulama problemi en etkin yaklaşımlarla çözülecek, elde edilen sonuçlar irdelenecektir.

2. ÜRETİM VE ÜRETİM SİSTEMLERİ

2.1. Üretim

Üretim, doğa tarafından karşılanmayan insan gereksinimlerini karşılamak üzere ekonomik değeri olan mal veya hizmetleri yaratmaktır. Başka bir tanıma göre ise üretim, hammadde ve yarı ürünlerden, kullanılabilir, topluma yararlı bir nihaî ürün yaratmak veya fiziksel bir varlık üzerinde onun ekonomik değerini arttıracak herhangi bir değişiklik yapmaktır. Bu ikinci tanım, daha çok mühendisler tarafından yapılan bir tanımdır. Ekonomistler tarafından yapılan tanımda ise "üretim, yarar yaratmaktır". Tanımlardan da anlaşılacağı üzere, üretimin temel amacı yarar ortaya koymak veya varolan olan yararı arttırmak, böylece topluma değer oluşturmaktır. Bu amaçla, yâni ekonomik bir anlam ifâde eden mal veya hizmet yaratmak amacıyla yapılan çalışmaların tümüne üretim denir **(Tanyaş ve Baskak, 2006) (Kobu, 1982) (Acar, 1989).**

Üretim, temelde dört şekilde gerçekleştirilir **(Tanyaş ve Baskak, 2006)**:

1. Şekil değişikliği ile
2. Mekân değişikliği ile
3. Zaman değişikliği ile
4. El değiştirme ile

Üretim, çok sayıda etmenden oluşan bir bileşimdir. Bu etmenler, basitçe ele alınırsa şöyle özetlenebilir: Hammadde, işgücü ve sermâye. Bu etmenler, üretim sisteminin girdilerini oluştururlar. Üretim sistemlerinin çıktıları ise ürün, yarı ürün, yan ürün ve hizmetlerdir **(Tanyaş ve Baskak, 2006) (Kobu, 1982)**.

2.2. Üretim Sistemlerinin Sınıflandırılması

2.2.1. Genel Bilgi

Üretim sistemi, girdilerin bir dönüştürme süreci sonunda çıktı durumuna geldiği bir açık sistemdir. İşletmelerde bir üretim yönetimi etkinliği olan üretim plânlama ve kontrol sisteminin yapısı, işletmedeki sözkonusu üretim sistemine göre değişir. Üretim sistemi ise fabrika ve işgücü düzenlemeleriyle belirlenir. Üretim plânlama ve kontrol faaliyetleri, üretilen ürün çeşidi azaldıkça, üretilen miktar arttıkça gittikçe basitleşir **(Tanyaş ve Baskak, 2006) (Acar, 1989)**. Üretim sistemleri, öncelikle geleneksel üretim sistemleri ve çağdaş üretim sistemleri olarak ikiye ayrılabilir.

2.2.2. Geleneksel Üretim Sistemleri

2.2.2.1. Ürüne Uygulanan Stok Politikasına Göre Sınıflandırma

Geleneksel üretim sistemleri dört şekilde sınıflandırılabilir: Ürüne uygulanan stok politikasına göre, ürün çeşidi ve üretim miktarına göre, üretim sürecine göre, ürün cinsine göre.

Firmanın uyguladığı stok politikası gözönünde bulundurulursa, üç farklı üretim tipinden sözedilir: stok için üretim, sipâriş için üretim, sipârişe göre son işlemler. Ürüne uygulanan stok politikasına göre üretimde, sipârişler doğrultusunda ürünler üzerinde yapılan özelleştirmeler ve ürünlerin teslim zamanları, değişkenleri oluşturur.

Stok İçin Üretim: Ürüne olan talebin sürekli ve düzgün olduğu sektörlerde uygulanır. Alıcının talebi, daha önce üretilmiş olan ve stokta tutulan ürünlerden karşılanır. Alıcının, ürünün üretimini beklemesi gerekmez ama bunun sonucu olarak da alıcı ürün üzerinde hiç bir özelleştirme yapılmasını isteyemez.

Sipâriş İçin Üretim: Elde hammadde stoğunun bulunduğu, alıcıdan gelen talep ve özelleştirme isteği üzerine üretimin başladığı durumdur. Bu durumda alıcı, üretim süresi boyunca bekler, bunu sonucunda da istediği tüm özelleştirmeler karşılanmış olur.

Sipârişe Göre Son İşlemler: Bu politikada yarı ürünler ve bâzı parçalar, sipâriş gelmeden üretilerek veya satın alınarak stoklanır. Daha sonra alıcıdan gelen sipâriş

üzerine hazırda bulunan parçalar kısa sürede birleştirilir ve alıcının isteği kısa sürede karşılanmış olur. Stok için üretim ve sipâriş için üretim politikalarının birleşimi gibi de düşünülebilir.

2.2.2.2. Ürün Çeşidi ve Üretim Miktarına Göre Sınıflandırma

Üretim, ürün çeşidi ve miktarına göre de sınıflandırılabilir:

Proses Tipi Üretim: Üretim sürecine tâbi tutulan malzemenin sıvı, akışkan olduğu üretim tipidir. Genelde talebin düzgün olduğu sektörlerde uygulanır. Üretim sipârişe göre değil, stok için yapılır. Ürün çeşitliliği azdır, üretim miktarları büyüktür. Bu üretim tipinde, özelleşmiş donanım kullanıldığından yatırım mâliyeti yüksektir. Proses tipi üretimin uygulandığı sektörlere örnek olarak kimya, meşrubat, çimento vb. sayılabilir.

Kitle Üretimi: Aynı çeşit ama farklı tiplerde ürünlerin büyük miktarlarda üretildiği bir sistemdir. Bu tür üretim sisteminde üretilen ürünler, genelde talebin yüksek olduğu ürünlerdir. Üretimin hızlı ve yüksek miktarlarla gerçekleştirilebilmesi için, mâliyeti yüksek özelleşmiş donanımlarla üretim hatları kurulmuştur. Bu nedenle plânlama evresi büyük önem taşır. Üretim hatlarının iyi dengelenmiş olması verimlilik açısından önemlidir. Esneklik düşüktür. Kitle üretiminin yapıldığı sektörler arasında ilk akla gelenler beyaz eşya ve otomotiv sektörüdür.

Parti Üretimi: Çok çeşidin veya aynı çeşidin farklı tiplerinin sınırlı miktarlarda üretildiği sistemdir. Her partide tek bir çeşit ürün üretilir. Ürün çeşitliliği azaldıkça ve üretim miktarları arttıkça kitle üretimine geçilir. Ürün çeşitliliği yüksek olduğundan genel amaca yönelik donanım kullanılır, bu da nitelikli işgücü gereksinimini arttırırken, donanım mâliyetini düşürür. Bu tip üretimde, üretim plânlama faaliyetleri büyük önem taşır ve oldukça karmaşıktır. Parti üretiminde, malzeme taşıma ve hazırlık sürelerinin yüksek olması gibi nedenlerden dolayı verimlilik düşüktür.

Proje Tipi Üretim: Üretim miktarı, diğer üretim tiplerine göre çok düşüktür. Üretim öncesinde talep kesin olarak bilinmektedir. Üretilen ürün genelde sabit olarak bir yerde durur, üretimde kullanılacak tüm donanım buraya getirilir. Ürün tasarımı tümüyle müşterinin istekleri doğrultusunda yapılır. Mâliyetler genelde oldukça yüksektir.

2.2.2.3. Üretim Sürecine Göre Sınıflandırma

Sipâriş Tipi Atölye: Bu tip üretim sürecinde, genel amaca yönelik tezgâhlar kullanılır. Gelen sipârişler, boşta duran tezgâhlarda, boş tezgâhın olmadığı durumlarda da tezgâhlar önünde sıraya sokulmak sûretiyle üretim sürecine alınırlar. Sipâriş tipi atölyede, makina kullanım oranı yüksek, iş akışı ise karmaşıktır.

Akış Tipi Atölye: Gelen sipârişler, ardışık olarak aynı tezgâhlardan geçerek üretilirler. Hattın başından giren malzeme veya yarı ürünler hattın sonundan nihaî ürün olarak çıkarlar. Duraklama ve ara bekleme sürelerinin en aza çekilebilmesi amacıyla, her tezgâhın üretim faaliyeti için bir süre belirlenmiştir. Üretim hattının iyi dengelenmiş olması, verimlilik açısından önemidir.

Sabit Konumlu Atölye: Bu tip üretim sisteminde, işlenen ürün sabit olarak bir yerde tutulur. Üretim süreci içerisinde, kullanılan tüm donanım bu yere getirilir ve üretim süreci burada tamamlanır. Örnek olarak yol ve köprü inşaatı verilebilir.

2.2.2.4. Ürün Cinsine Göre Sınıflandırma

Kimi durumda üretim sisteminin özellikleri, nihaî ürünün niteliklerine doğrudan bağlıdır. Üretim sisteminin yapısı tümüyle üretilen ürüne özel olabilir. Örnek olarak lastik üretimi, demir-çelik üretimi, tekstil ürünleri üretimi vb. verilebilir **(Baskak, 2005)**.

2.2.3. Çağdaş Üretim Sistemleri

2.2.3.1. Hücresel Üretim Sistemi (Grup Teknolojisi)

Bu sistemde, üretim yeri, belirlenmiş parça gruplarına yönelik oluşturulmuş makina hücreleri veya gruplarına bölünmüştür. Hücreler genelde bir veya iki makinadan oluşurlar, makina sayısı nadir olarak beşi aşar **(Askin ve Standridge, 1993)**. Bu sistem ile, küçük bir sistemin etkinlik ve denetlenebilirliği, büyük bir sisteme yansıtılabilmektedir. Sürecin işlevsel olarak düzenlenmesi ile de iş akışı basitleştirilmiştir. Hücresel üretimin üstünlükleri, benzer parçaların işlendiği tezgâhların birarada tutulması nedeniyle hazırlık sürelerinin düşmesi, malzeme taşımasının azalması, nitelikli işgücüne duyulan gereksinimin düşmesi olarak sıralanabilir **(Tanyaş ve Baskak, 2006)**.

2.2.3.2. Bilgisayarla Bütünleşik Üretim Sistemleri

Üretimde artık sayısal kontrollu tezgâhların kullanılmasıyla tüm üretim işlevlerinin bilgisayar ile bütünleştirilmesi olanaklı olmuştur. Bugün, ürün tasarımı, geliştirilmesi, üretim plânlama ve kontrol işlevi gibi üretimin pek çok aşamasında bilgisayar destekli sistemler geliştirilmiş ve bilgisayarla bütünleşik üretim sistemleri kurulmuştur. Kullanılan modüllerin başlıcaları CAD, CAM, CAE, ERP vb. olarak sayılabilir **(Tanyaş ve Baskak, 2006)**.

2.2.3.3. Tam Zamanında Üretim Sistemi

Tam zamanında üretimde (TZÜ) amaçlanan, israfı sürekli olarak azaltarak süreçleri ve yordamları (prosedürleri) en iyi hâline getirmektir. Üretimi gerçekleştirilen ürüne müşteri açısından hiç bir değer katmayan her adım ise israf olarak kabul edilir. Bu doğrultuda, malzeme taşıma, süreç içi stoklar, gecikmeler, aşırı evrak işleri israf olarak sayılır **(Everett ve Ronald, 1992)**. TZÜ'de kullanılan bazı teknikler; hazırlık sürelerinin azaltılması, kanban, hücresel üretim, hat durdurma ve grup teknolojisi olarak sayılabilir. TZÜ; tedârikçiler, nakliye, kalite, iletişim ve çizelgelemeyi de dikkate alır **(Bedworth ve Bailey, 1987)**.

TZÜ'nün önemli bir parçası olan çekme sistemine göre, her iş istasyonu gereksinim duyduğu malzemeyi bir önceki istasyondan çeker. Son istasyonun sipârişten haberdar olması ile üretim süreci başlar. Malzeme gereksinim bilgisi son istasyondan ilk istasyona doğru ilerlerken, malzemeler ilk istasyondan son istasyona doğru ilerler. Bilgi akışı için kanban kağıtları kullanılır **(Everett ve Ronald, 1992)**.

2.2.3.4. Esnek Üretim Sistemleri

Esnek üretim sistemi ile, merkezî bir bilgisayar ile kontrol edilen bir grup sayısal kontrollu makina, istasyonlar ve bunlara bağlı otomatik bir malzeme taşıma sistemi ifâde edilir **(Askin ve Standridge, 1993)**. İki temel amacı vardır:

1. Rastlantısal sipâriş üretimini sağlamak, yâni belirli bir zaman içinde parçaların istenen bir karışımını sorunsuz üretmek.
2. Tezgâh önlerindeki beklemeyi azaltarak küçük parti üretiminde tezgâhtan elde edilen yararı arttırmak (**Tanyaş ve Baskak, 2006)**.

2.2.3.5. Optimum Üretim Teknolojisi

Bu yaklaşıma göre, bir işletmedeki darboğaza neden olan tezgâhların verimliliği arttırılarak üretim miktarı arttırılabilir. Bu nedenle, çizelgeleme öncesinde tezgâhlar darboğaza neden olup olmadıklarına göre ikiye ayrılır ve bu iki grup için çizelgeleme farklı yaklaşımlarla gerçekleştirilir **(Tanyaş ve Baskak, 2006)**.

2.3. Montaj Kavramı

2.3.1. Montaj Kavramının Tanımı

Montaj, bir ürün oluşturmak amacıyla çeşitli parçaların toplanma ve birleştirilme sürecine verilen addır. Kullanılan parçalar ve bunları biraraya getirme işi üzerinden tanımlanır. Montajda kullanılan parçalar, malzeme veya yarı ürün olabilir **(Scholl, 1999)**.

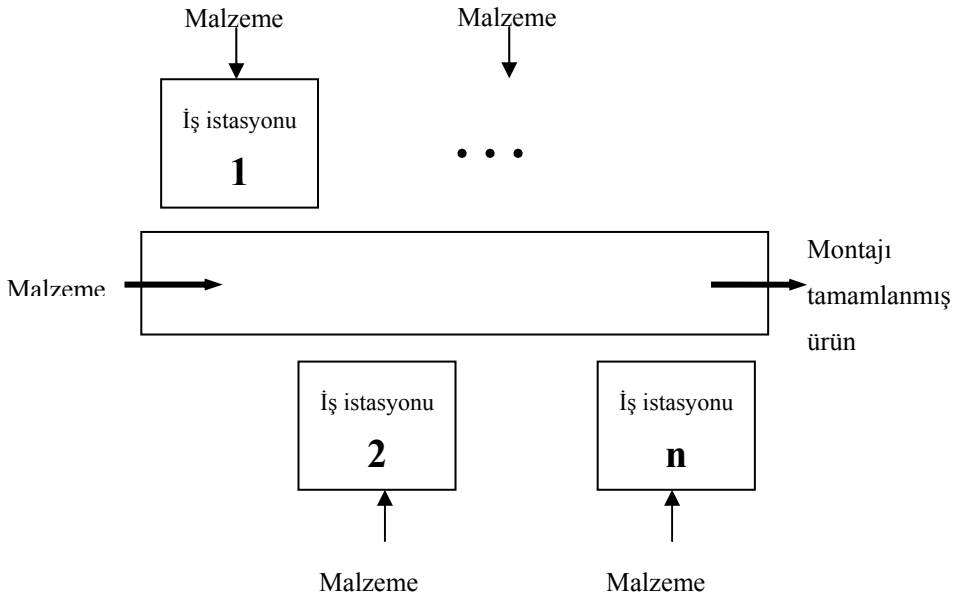
2.3.2. Montaj Hatlarının Üretimdeki Yeri

Endüstrileşme sürecinde, üretim miktarını arttırmak, yâni daha hızlı, ucuz ve serî bir üretim gerçekleştirebilmek için toplam iş parçalara bölünerek, her parçanın ayrı bir işçi tarafından işlenmesi yoluna gidilmiştir. Bunun sonucunda da malzemelerin, üzerinde istasyonlar bulunan bir hat üzerinde aktığı montaj hatları kullanılmaya başlanmıştır. Montaj hatları ilk olarak 1913'te, A.B.D.'de Ford Firması'nda kullanılmıştır. Henry Ford'un Highland ve River Rouge firmalarında, montaj hatlarının kullanılması ile birlikte oldukça önemli iyileşmeler gözlenmiştir. "İşlerin verimli duruma getirilmesi, etkin malzeme taşıma ve birbirinin yerine kullanılabilen parçaların üretimi buna örnektir" **(Baskak, 2005)**.

3. MONTAJ HATLARININ DENGELENMESİ

3.1. Montaj Hattı ve Hat Dengeleme Kavramı

Ürünler genelde birden çok parçanın biraraya gelmesi ile oluşur. Üretim sürecinde bu parçaların belli bir sıra dahilinde biraraya getirilmesi montaj işlemi olarak tanımlanır. Nihaî ürünü elde etmek için, montaj işleminin ve ilgili operasyonların gerçekleştirildiği, aralarında malzemelerin akış hattı boyunca taşındığı (genelde bir konveyör yardımıyla) ardışık iş istasyonlarından oluşan hatta ise **montaj hattı** denir. Montajı yapılan ürün, tüm istasyonları ziyâret ederek hat boyunca ilerler ve montajı tamamlanmış olarak hattı terkeder. Şekil 3.1'de örnek bir montaj hattı şeması verilmiştir.



Şekil 3.1: Örnek Bir Montaj Hattı Şeması **(Baskak, 2005)**

Montaj hatlarında malzeme genelde işgücü yardımıyla işlenir. Hat istasyonlarında çalışan işçiler sürekli olarak meşgûl olacak şekilde yerleştirilirler. Her istasyon çalışanı, yarı ürün üzerinde kendi istasyonuna atanmış işlemleri gerçekleştirir, sonra

ürün bir sonraki işlemler için izleyen istasyona geçirilir, hemen ardından da bir önceki istasyondan yeni yarı ürün gelir **(Askin ve Standridge, 1993)**.

İş istasyonlarında gerçekleştirilen en az mantıksal iş elemanları, iş öğeleri olarak adlandırılırlar. Bir iş öğesi birden fazla istasyon arasında bölünemez. Bir istasyona atanmış iş öğelerinin işlem sürelerinin toplamı, o istasyonun iş içeriğini (yükünü) belirtir. Ürünün montajı sırasında herhangi bir istasyonda işlem gördüğü en büyük süreye "çevrim süresi" denir. Ürün, her istasyonda en fazla bu süre kadar işlem görebilir. Çevrim süresi aynı zamanda hattan ne kadar sürede bir, bir nihaî ürünün çıktığını gösterir. Üretim oranı, çevrim süresinin tersi olarak ifâde edilir. Çevrim süresi genelde, talebe bağlı olarak önceden belirlenmiştir. İstasyonlarda yapılacak olan iş öğeleri, istasyonlara keyfî olarak atanamazlar. Montaj işleminde hangi sıra ile uygulandıkları dikkate alınarak oluşturulmuş olan öncelik diyagramı gözönünde tutularak atamalar gerçekleştirilir. Öncelik diyagramında öncül iş öğesi ile ardıl iş öğesi bir ok yardımıyla birbirine bağlanmış şekilde gösterilir.

İş öğelerinin iş istasyonlarına, istasyonların âtıl süreleri toplamı en az olacak şekilde atanmasına montaj hattının dengelenmesi denir. Sabit bir çevrim süresinde istasyon sayısını enazlayarak bu durum gerçekleştirilebilir. Tüm istasyonların iş yükünün süre olarak eşit olduğu bir montaj hattı mükemmel dengededir **(Baybars, 1986)**.

3.2. Montaj Hatlarının Dengelenmesinin Amaçları

Montaj hatlarının dengelenmesinde amaçlar farklılık gösterebilirler ve birbirleriyle çelişebilirler. Ama burada asıl amaçlanan, tüm istekleri dikkate alarak en uygun çözüme ulaşmaktır.

Montaj hattının kurulmasında hedeflenenler şöyle sıralanabilir:

- Malzeme akışının düzenli olması
- İnsangücü ve tezgâh kapasitelerinin en üst düzeyde kullanılması
- İşlemlerin en kısa sürede tamamlanması
- Montaj hattı üzerindeki iş istasyonu sayısının enazlanması
- Âtıl sürelerin enazlanması

- Âtıl sürelerin istasyonlar arasında düzgün şekilde dağıtılması
- Üretim mâliyetinin enküçüklenmesi

Burada mâliyet ile hem işgücü mâliyeti, hem de hattın uzunluğu ile orantılı olan kullanılan alan ve donanım mâliyeti kastedilmiştir. Her ikisi de istasyon sayısını enazlamakla enküçüklenebilir **(Baskak, 2005)**.

3.3. Montaj Hatlarının Yerleşimi

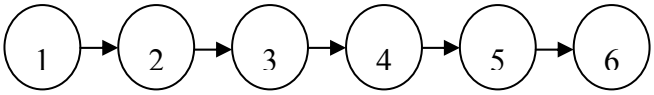
Üretilecek ürün ve hattın kurulacağı yerin fiziksel yapısı, montaj hattının şeklini belirlemede yardımcı olur. Genelde düz hatlar yeğlenir, ama kimi aygıtların birden fazla istasyonda kullanılması, hacmin fiziksel özelliklerinin düz hatta izin vermemesi gibi çeşitli nedenlerden ötürü, farklı biçimlerde hat şekilleri de yeğlenebilir. Fiziksel montaj hatları; düz, dairesel, rassal, değişik açılı, U-şekilli ve zig-zag gibi değişik biçimlerde tasarlanabilir **(Baskak, 2005)**. Fiziksel montaj hattı tipleri Şekil 3.2'de gösterilmiştir.

3.4. Montaj Hatlarının Dengelenmesinde Kullanılan Temel Kavramlar

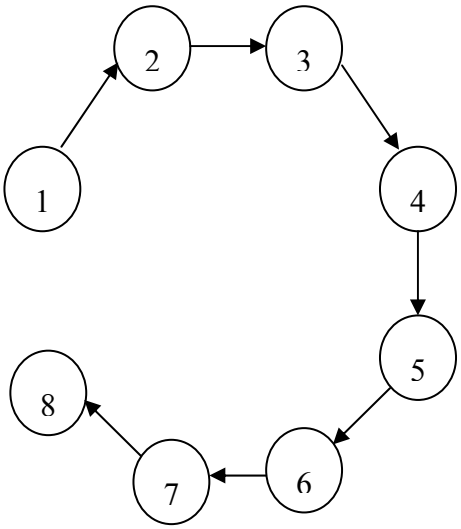
İş Öğesi: Montaj sürecinin toplam iş yükünün bölünemeyen, mantıksal olarak en küçük parçasıdır. İş öğesi, gereksiz ek iş yaratmadan daha küçük parçalara bölünemez. Bir iş öğesini gerçekleştirmek için gerek duyulan süreye **İş Öğesi İşlem Süresi** denir.

İş İstasyonu: Montaj hattı üzerinde, bir veya daha çok iş öğesinin gerçekleştirildiği hat parçasına iş istasyonu denir. Boyutları, makinalar ve ekipmanlar ve atanmış iş öğeleri üzerinden tanımlanır. İş istasyonları elle yapılan ve otomatik olarak ayrılabilir, ama genelde iş öğeleri, iş istasyonlarında işgücüne dayalı olarak gerçekleştirilir. Bir iş istasyonunun iş içeriği, o istasyonun **İş Yükü** olarak tanımlanır. İş yükü için gerekli olan süre ise **İstasyon Süresi** olarak adlandırılır **(Scholl, 1999)**.

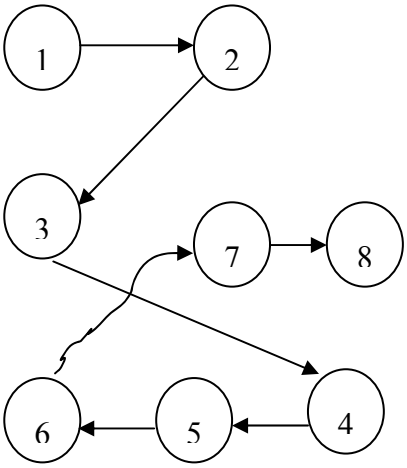
Toplam İş Süresi: Montaj hattı üzerinde montajı gerçekleştirilecek bir ürünün tüm iş öğesi sürelerinin toplamına toplam iş süresi denir.



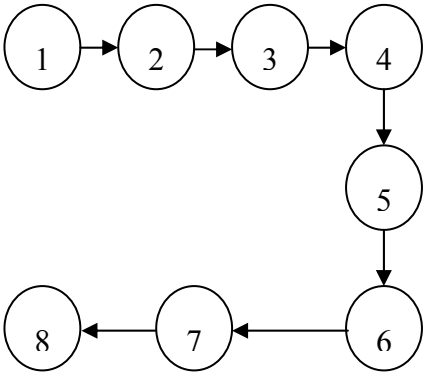
Düz hat



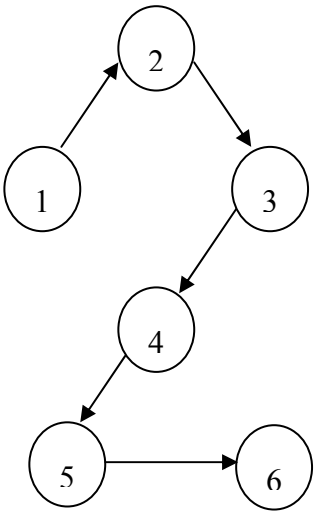
Dairesel hat



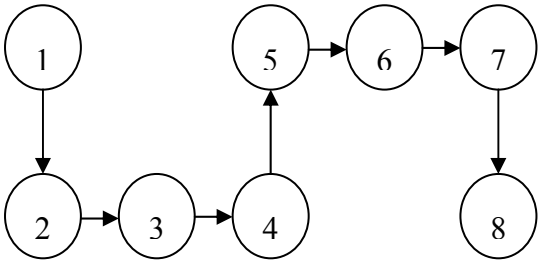
Rassal hat



U-şekilli hat



Değişik açılı

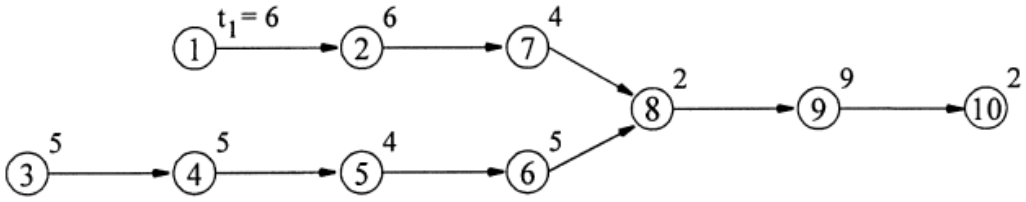


Zig-zag hat

Şekil 3.2: Fiziksel Montaj Hattı Tipleri **(Baskak, 2005)**

Çevrim Süresi: Çevrim süresi, bir iş öğesinin montaj hattı üzerindeki bir istasyonda yerine getirilebileceği en yüksek süreye denir. İş öğeleri bölünemez olduklarından çevrim süresi, en büyük iş öğesi süresinden küçük olamaz. Çevrim süresinin tersi, o hattın **Üretim Oranı** (production rate) değerini verir. Çevrim süresi ile istasyon süresi arasındaki pozitif fark, o istasyonun **Âtıl Süresi**'ni gösterir, çünkü bu süre zarfında işçi boştadır. Tüm istasyonların âtıl sürelerinin toplamı ise **Denge Gecikme Süresi** (balance delay time) olarak adlandırılır.

Öncelik Kısıtları: Teknolojik kısıtlar dahilinde (bazı iş öğelerinin birbirini takip etmesi gerekir), iş öğelerinin yaklaşık olarak hangi sıra ile uygulanması gerektiği önceden belirlenebilir. Bu kısmi sıralama **Öncelik Diyagramı** vasıtası ile gösterilir. Öncelik diyagramında her düğüm bir iş öğesini temsil eder ve her iş öğesi, ardılı ile bir ok aracılığıyla bağlanır. Okun çıktığı iş öğesi, öncül iş öğesidir. Düğümlerin sağ üst kısımlarında, o iş öğesinin işlem süresi gösterilir. Şekil 3.3.'te 10 iş öğesi için bir öncelik diyagramı örneği görülmektedir. **Öncelik Matrisi** ise, öncelik diyagramının üst üçgensel matrise dönüştürülmüş hâlidir. İş öğesinin numarasını taşıyan satırla, ardılı iş öğesinin numarasını taşıyan sütunun kesiştiği hücreye 1, diğer hücrelere 0 konur **(Scholl, 1999) (Baskak, 2003)**. 11 iş öğesi için bir öncelik matrisi örneği Tablo 3.1'de verilmiştir.



Şekil 3.3: 10 İş Öğesi İçin Öncelik Diyagramı Örneği **(Scholl ve Klein, 1999a)**

Gerekli En Az İş İstasyonu Sayısı: Tüm iş öğelerinin çevrim süresini aşmayacak şekilde istasyonlara atanması durumunda gereksinim duyulan en az istasyon sayısıdır. Hesaplamak için çeşitli yöntemler vardır. Genelde birden fazla yöntem sıra ile uygulanır, en fazla istasyon sayısını veren sonuç, gerekli en az istasyon sayısı olarak kabul edilir.

Tablo 3.1: 11 İş Öğesi İçin Öncelik Matrisi Örneği **(Baskak, 2005)**

		ARTÇIL ÖĞELER										
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
ÖNCÜL	**1**	-	1	1	1	1	1	1	1	1	1	1
	2		-	0	0	1	1	1	0	1	1	1
	3			-	0	1	1	1	0	1	1	1
	4				-	0	0	0	1	1	1	1
	5					-	1	1	0	1	1	1
ÖĞELER	**6**						-	0	0	1	1	1
	7							-	0	0	1	1
	8								-	1	1	1
	9									-	1	1
	10										-	1
	11											-

Sıralama Gücü: Öncelik diyagramı içerisindeki öncelik ilişkilerinin bağıl sayısını verir. Yüksek sıralama gücüne sahip problemlerin düşük sıralama gücü olanlara kıyasla daha karmaşık olması beklenir. Tüm öncelik ilişkileri kümesinin kardinalitesinin n*(n-1)/2 değerine bölünmesi ile elde edilir. Değeri 0-1 arasında değişir. İş öğeleri arasında tek bir sıra olanaklı ise 1, öncelik ilişkisi sözkonusu değilse 0 değerini alır.

Esneklik Oranı: Öncelik matrisindeki 0 değerli hücrelerin sayısının, toplam değer girilmiş hücre sayısına bölünmesi ile elde edilir. Bu oran;

Esneklik Oranı = 1 – Sıralama Gücü

şeklinde de hesaplanır **(Scholl, 1999)**.

Denge Kaybı: İş öğelerinin istasyonlara ne kadar dengeli dağıtıldığını gösterir. ”Denge kaybı, her istasyonda, birim üretim için ayrılan toplam süreyle gerekli süre

arasındaki farkın, ayrılan süreye oranıdır ve çöğunlukla sıfırdan büyük bir değerdir." Denge kaybının sıfır olması istenen (ideal) durumdur.

$$D(\%) = \left[\left(C - C^{*}\right)/C\right] * 100 = \left[\left(N * C - \sum_{i=1}^{n} t_i\right)/\left(N * C\right)\right] * 100 \quad \textbf{(2.1)}$$

Düzgünlük İndeksi: Montaj hattı üzerindeki iş istasyonlarının işlem sürelerinin düzgün olup olmadığını gösterir. İndeks ne kadar küçükse, düzgünlük o kadar fazladır.

$$D\dot{I}(\%) = \left[\sqrt{\left(\sum \left(T_{enb} - T_i\right)^2\right)}/\left(N * C\right)\right] * 100 \quad \textbf{(2.2)}$$

Hat Etkinliği: Dengeleme sonucunda montaj hattında montaj işlemi için çalışılan sürenin ne kadarının etkin olarak kullanıldığını gösterir.

$$HE(\%) = \left[\sum_{i=1}^{n} t_i /\left(N * C\right)\right] * 100 \quad \textbf{(2.3)}$$

Kuramsal Etkinlik: Hattın belirli bir çevrim süresi dahilinde en az sayıda istasyonla dengelenmesi durumundaki etkinliğine denir **(Tanyaş ve Baskak, 2006)**.

$$KE(\%) = \left[\sum_{i=1}^{n} t_i /\left(N_{enaz} * C\right)\right] * 100 \quad \textbf{(2.4)}$$

3.5. Montaj Hatlarının Dengelenmesini Etkileyen Temel Etmenler ve Kısıtlar

3.5.1. Temel Etmenler

Montaj hatlarının dengelenmesini etkileyen temel etmenler üçe ayrılabilir **(Baskak, 2005)**:

- Mühendislik spesifikasyonları, işlemler arası öncelikler ve gerekli kaynak gereksinimleri
- İşin yapılmasında izlenen yöntem
- Kullanılan aygıtlar ve tezgâhlar

3.5.2. Kısıtlar

Montaj hatlarının dengelenmesini etkileyen kısıtları birincil ve ikincil kısıtlar olarak ikiye ayırabiliriz.

Birincil Kısıtlar: Çevrim süresi ve öncelik ilişkileri (iş öğelerinin kendi aralarında sahip oldukları öncelik sırası).

İkincil Kısıtlar: Bu tip beş kısıt vardır:

- **Konum Kısıtı:** Montaj hattındaki nesne ile montaj işini yapan işçilerin birbirlerine göre konumları ile ilgilidir. Bu kısıtla genelde büyük boyutlu nesnelerin montajı sırasında karşılaşılır.
- **Sabit Donanım Kısıtı:** Montaj işlemi sırasında istasyonlarda kullanılan yeri değiştirilemez donanımlardan kaynaklanır. Bu kısıt, iş öğelerinin değiştirilebilirliğini azaltır.
- **İstasyon Yükü:** Özellikle hattın başındaki istasyonlarda karşılaşılabilecek sorunların tüm hattı etkilemesinin istenmediği durumlarda, bâzı istayonların yüklerinin çevrim süresinden daha az olması yeğlenebilir. Bu da bir kısıtlamayı beraberinde getirir.
- **Aynı İstasyona Atanması İstenen İş Öğeleri:** Montaj hattında gerçekleştirilen iş öğelerinden bâzılarının aynı istasyonda uygulanması istenebilir. Genelde aynı alet veya aygıtın kullanıldığı iş öğelerinde bu durum geçerli olur. Böyle bir durumda bu iş öğeleri tek bir iş öğesi olarak düşünülebilir.
- **Aynı İstasyona Atanmaması İstenen İş Öğeleri:** İşçiyi gözetmek veya işin sağlıklı yapılmasını sağlamak gbi çeşitli sebeplerden ötürü bazı iş öğelerenin aynı istasyona atanmaları istenmeyebilir. Öncelik ilişkisi açısından bir engel olmasa bile bu iş öğelerinin ayrı istasyonlara atanması gerekebilir **(Tanyaş ve Baskak, 2006)**.

3.6. Montaj Hatlarında Model Sayısı

3.6.1. Tek Modelli Hatlar

Tek bir ürün büyük miktarlarda sürekli olarak üretilir (kitle üretimi). Tüm istasyonlar aynı iş öğelerini yeniden tek tip iş parçaları üzerinde uygularlar. İstasyonların iş yükleri süreç içerisinde her zaman aynıdır. Tek modelli hatların tasarımı basittir.

3.6.2. Karışık Modelli Hatlar

Temel ürünün çeşitli modelleri aynı hat üzerinde aynı anda üretilir. Modellerin üretim süreci ana hatlarıyla benzerdir, salt çeşitli özelliklerle birbirlerinden ayrılırlar. Üretim sürecinde tüm modeller için yaklaşık olarak aynı temel işlemler uygulanır. Farklı modellerde kimi iş öğeleri eksik veya fazla iken, kimi iş öğeleri de farklı işlem sürelerine sahip olabilir. Tüm modeller için benzer iş öğeleri sözkonusu olduğundan hazırlık süreleri çok azdır. Karışık modelli hatların dengelenmesinde model sırasının belirlenmesi de önemli rol oynar.

3.6.3. Çok Modelli Hatlar

Çeşitli ürünler aynı hat üzerinde partiler hâlinde üretilirler. Ürünlerin üretim süreçlerindeki belirgin farklılıklardan dolayı, üretilen model değiştiğinde hattın yeniden hazırlanması gerekir. Bu nedenle hazırlık mâliyetini düşürmek amacıyla ürünler büyük partiler hâlinde üretilirler, bu da stok mâliyetlerini arttırır. Çok modelli hatlar, ekonomik parti büyüklüğü ve modeller için üretim çevrimini belirleme problemini de beraberinde getirir **(Scholl, 1999)**.

4. MONTAJ HATTI DENGELEME PROBLEMİ

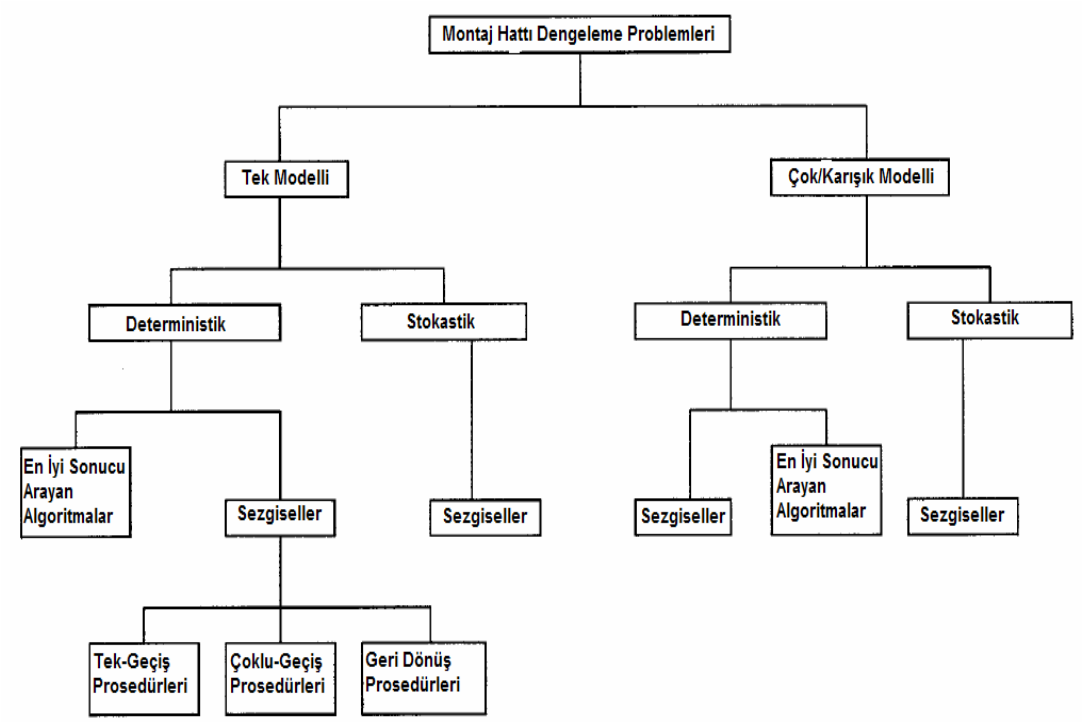
4.1. Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması

Montaj Hattı Dengeleme Problemlerinin (MHDP) tümünde, her iş öğesinin bir tek istasyona atandığı, öncelik kısıtlarının ve başka varolan kısıtların dikkate alındığı olurlu bir hat dengesine ulaşmak amaçtır **(Becker ve Scholl, 2006)**.

MHD problemlerini, hat üzerinde üretilen model sayısı ve iş öğelerinin işlem sürelerinin deterministik veya stokastik olmalarına göre farklı kategorilere ayırabiliriz. Deterministik işlem sürelerinin sözkonusu olduğu durumda, iş öğeleri sürelerinin belirli ve her birim ürün için aynı olduğu kabul edilir. Stokastik işlem sürelerinin sözkonusu olduğu durumda ise iş öğesi işlem sürelerinin belirli olmadığı, her birim ürünün işlem görmesi sırasında farklılık gösterdiği düşünülür. İşlem süreleri herhangi bir dağılıma uygun olabilir veya olmayabilir.

Hat dengelenirken farklı amaçlar gözetilebilir. Bunlar deterministik durumda genelde belirli bir çevrim süresi için istasyon sayısının enküçüklenmesi veya belirli sayıda istasyona sahip olunan durumda çevrim süresinin enküçüklenmesidir. Stokastik durumda en sık amaçlanan ise toplam sistem mâliyetinin enküçüklenmesidir.

Bu sınıflandırma ve kullanılan çözüm yöntemleri Şekil 4.1'de görülebilir.



Şekil 4.1: MHDP'lerin ve Çözüm Yöntemlerinin Sınıflandırılması **(Erel ve Sarin, 1998)**

Tek-Geçiş Prosedürleri: Tek bir özelliğe göre atanacak iş öğesi seçilir

Çoklu-Geçiş Prosedürleri: Tek-Geçiş karar kurallarının bileşimi kullanılarak atanacak iş öğesi seçilir

Geri Dönüş Prosedürleri: Varolan bir sonuç üzerinden daha iyi bir sonuca ulaşılmaya çalışılır **(Talbot ve diğ., 1986)**

Montaj hattı dengeleme problemleri, yukarıdaki sınıflandırmadan farklı olarak, Basit Montaj Hattı Dengeleme Problemi ve Genel Montaj Hattı Dengeleme Problemi olarak da ikiye ayrılarak incelenmişlerdir. Basit Montaj Hattı Dengeleme Problemi tek modelli deterministik montaj hattı dengeleme problemi ile aynı niteliklere sahiptir. Çevrim süresinin veya istasyon sayısının enküçüklenmesine göre tip-1 ve tip-2 olarak adlandırılır. Genel Montaj Hattı Dengeleme Problemi ise, tek modelli deterministik montaj hattı dengeleme problemi dışında kalan tüm problemleri içerir. Yâni Basit Montaj Hattı Dengeleme Problemi'nin karakteristiklerinin bir veya daha fazlasının gevşetilmiş hâlidir **(Baybars, 1986)**.

Bu çalışmada basit (tek modelli deterministik) montaj hattı dengeleme problemi ele alınacaktır.

4.2. Basit Montaj Hattı Dengeleme Problemi (BMHDP)

Basit MHD Problemi'nin ana karakteristikleri şunlardır:

- Tüm girdi parametreleri kesinlikle bilinmektedir.
- Bir iş öğesi iki veya daha fazla iş istasyonu arasında paylaştırılamaz.
- Teknolojik öncelik kısıtlarından dolayı iş öğeleri rastlantısal olarak istasyonlara atanamazlar.
- Tüm iş öğeleri gerçekleştirilmelidir.
- Montaj hattı tek bir model için tasarlanmıştır.
- Hat boyunca herhangi bir besleyici, alt-montaj hattı yoktur.
- Tüm istasyonlar, her iş öğesi için uygun donanıma sahiptir.
- İş öğesi süreleri, işin yapıldığı istasyondan ve önceki veya sonraki iş öğelerinden bağımsızdır.
- Herhangi bir iş öğesi, herhangi bir istasyonda uygulanabilir.

Tip-1 BMHDP'de yukarıdakilere ek olarak

- çevrim süresi verilmiştir ve sabittir

kabulü geçerlidir.

Tip-2 BMHDP'de ise

- istasyon sayısı verilmiş ve sabittir

kabulü geçerlidir.

Tip-1 BMHDP'de amaç, istasyon sayısını enküçüklemek iken, Tip-2 BMHDP'de çevrim süresini enküçüklemektir **(Baybars, 1986) (Scholl ve Becker, 2006)**.

Bu çalışmada Tip-1, BMHDP ele alınacaktır.

Tip-1 BMHDP'nin matematiksel modeli **(Talbot ve Patterson, 1984)**:

$$\min A_n$$

$$\sum_{i \in W_j} t_i \leq C \qquad i = 1,\ldots, n \qquad j = 1,\ldots, H$$

$$A_m \leq A_k \qquad m \in P_k \qquad k = 1,\ldots, n$$

$E_i \leq A_i \leq L_i$(ÜS)

A_i bir tamsayı ve $\forall i \in M$

E_i ve L_i ise şöyle bulunur;

$$E_i = \left[\left(t_i + \sum_{m \in P_i^*} t_m \right) / C \right]^+$$

$$L_i\,(ÜS) = ÜS + 1 - \left[\left(t_i + \sum_{m \in F_i^*} t_m \right) / C \right]^+$$

4.3. Çözüm Yöntemlerinin Sınıflandırılması

4.3.1. Sezgisel Yöntemler

Sezgisel yöntemlerde problemin çözümü, belli kabuller altında çeşitli prosedürlerin (yordamların) uygulanmasıyla elde edilir. Elde edilen çözüm olurlu bir çözümdür ama en iyi çözüm olmayabilir. Varolan sezgisel problemlerin çoğu Tip-1 BMHDP'nin çözümüne yöneliktir. Tip-2 BMHDP'ye yönelik sezgiseller de vardır. Sezgisel yöntemler kullanılarak, analitik yöntemlere kıyasla daha kısa sürede çözüme ulaşılır, ama bulunan çözümün genelde en iyi çözüm olmadığı unutulmamalıdır. Son yıllarda genetik algoritma gibi meta-sezgisel yöntemler de dikkat çekmektedir. Literatürde sözedilen bâzı sezgisel yöntemler şunlardır **(Baskak, 2005)**:

- Konum ağırlıklı dengeleme tekniği
- COMSOAL tekniği
- İki aşamalı dengeleme tekniği (Moddie-Young)
- Gruplama Yöntemi
- Aday matris ile çözüm

4.3.2. Analitik Yöntemler

Analitik yöntemler sâyesinde MHD problemlerinin en iyi çözümünü bulmak olanaklıdır. Literatürde, MHD problemlerinin çözümü için analitik yöntem olarak 0-1 tam sayılı programlama, tam sayılı programlama, dal-sınır algoritması, dinamik

programlama kullanılmıştır **(Baybars, 1986)**. Ağırlıklı olarak ise dinamik programlama ve dal-sınır algoritması tercih edilmiştir. Analitik yöntemler kullanıldığında, özellikle işlem sayısının arttığı ve esneklik oranının yüksek olduğu durumlarda, en iyi çözüme ulaşmak uzun sürebilir. Bu gibi durumlarda özellikle dal-sınır algoritmalarındaki eleme kuralları, gereksinim duyulan zamanın azaltılmasında büyük yarar sağlamaktadır. Bu çalışmada analitik yöntemlerden dinamik programlama ve dal-sınır algoritması ele alınacaktır.

5. BMHDP'DE UYGULANAN BAŞLICA ANALİTİK YÖNTEMLER

5.1. Dinamik Programlama (DP)

"Dinamik programlama, n değişkenli bir problemin optimum çözümünü, problemi n aşamaya ayrıştırarak ve her aşamada tek değişkenli bir alt-problemi çözerek belirler. Bunun hesaplama üstünlüğü, n-değişkenli alt-problemler yerine tek değişkenli alt-problemleri optimum kılacak olmamızdır." Dinamik programlamada hesaplamaların, her alt-problemde yinelenmesi ile elde edilen sonuç, bir sonraki alt-problemde girdi olarak kullanılır **(Taha, 2003)**.

Dinamik programlama probleminin en iyi çözümüne, alt-problemlerin en iyi çözümlerinden hareketle ulaşılabileceğine dayanır. BMHDP'de dinamik programlama uygulamalarında, genelde ileriye doğru yineleme kullanılır. Başlangıç durumundan başlanarak, aşamalar arttırılarak son duruma ulaşılır.

BMHDP'de dinamik programlamanın kullanıldığı durumlarda, en iyi sonucu veren istasyon yüklerinin belirlenebilmesi için tüm olurlu istasyon atamaları seçeneklerinin oluşturulması gerekir. İş öğesi sayısının fazla, esneklik oranının yüksek olduğu problemlerde bu durum çok büyük işlem yüküne ve uzun işlem sürelerine yol açar.

5.2. BMHDP'de Dinamik Programlama Uygulamaları Üzerine Literatür Araştırması

Jackson'ın (1956) makalesinde 'montaj hattı dengeleme problemi' ismi kullanılmış, teknolojik öncelik diyagramının nasıl oluşturulacağı anlatılmış ve montaj hattı dengeleme problemi ilk kez dinamik programlama mantığı ile çözülmüştür. Jackson makalesinde, ilk seviye için olası istasyon atama dizilerinin tümünü oluşturmuş ve sonraki tüm seviyelerde bu işlemi tekrar etmiştir. Hesaplama boyunca aynı iş öğelerini içeren dizilerden sadece biri elde tutulmuş, diğerleri elenmiştir.

Held, Karp ve Shareshian'ın (1963) makalesinde DP, dinamik programlama adı ilk kez kullanılarak, basit montaj hattı dengeleme problemine uygulanmıştır. Bu modelde amaçlanan, belirli bir çevrim süresi için iş istasyonu sayısını enküçüklemektir. Modelde, Held ve diğerleri, "olurlu küme" leri ve "olurlu dizi" leri tanımlayarak işe başlamışlardır. Olurlu küme, öncelik ilişkileri dikkate alındığında, öncesinde herhangi bir j iş öğesinin uygulanması gerekmeden, birden fazla sıra ile uygulanabilen iş öğelerinden oluşur. Olurlu dizi ise, öncelik ilişkileri dikkate alındığında, öncesinde herhangi bir j iş öğesinin uygulanması gerekmeden, belirli bir sıra ile uygulanan iş öğelerinden oluşur. Her olurlu dizi ile bağlantılı olarak, dizideki iş öğelerinin belirlenmiş sıra dahilinde en az sayıdaki iş istasyonuna atanmasını sağlayan özel atamaya da "dizinin onaylanan ataması" denir. Bu atamada dizideki iş öğeleri, belirtilen sıra dahilinde birinci istasyondan başlanarak ve çevrim süresi kısıtı dikkate alınarak iş istasyonlarına atanır. Herhangi bir a dizisinin onaylanan atamasında r iş istasyonuna gereksinim duyulmuşsa ve son iş istasyonuna atanan iş öğelerinin toplam süresi $\tau^{(r)}$ olarak ifâde edilirse, olurlu dizinin atanmasının toplam süresi, yâni "mâliyeti" $c_a = (r-1)C + \tau^r$ olarak tanımlanır. Eğer varolan a olurlu dizisinin sonuna bir j iş öğesi ekleyerek yeni bir b olurlu dizisi elde edilmiş ise, b dizisinin mâliyeti $c_b = c_a + \Delta(c_a, t_j)$ olur.

(i) Eğer $[x/C]^+ = [(x+y)/C]^+$ veya $[(x+y)/C]^+ = (x+y)/C$ ise, o zaman $\Delta(x,y) = y$'dir.

(ii) Eğer $[x/C]^+ < [(x+y)/C]^+$ veya ise, $[(x+y)/C]^+ < (x+y)/C$ o zaman $\Delta(x,y) = C[(x+y)/C]^+ + y - x$ olur.

Yukarıdaki ifâdede $\lceil x \rceil^+$, x'e eşit veya x'ten küçük en büyük tamsayı demektir. (i), son istasyonda eklenen iş öğesinin atanabileceği boş sürenin olduğu durumu, (ii) ise olmadığı durumu gösterir. Böylece olurlu kümelerin mâliyeti de, kümelerin olurlu dizilerinin en az mâliyetlisinin mâliyeti olarak belirlenir. Herhangi bir s olurlu kümesi için $c(s) = \min c_a$ dır. n elemanlı bir olurlu kümenin mâliyeti ise 1, 2, 3, ..., (n-1) elemanlı alt kümelerin mâliyetlerinin yinelemeli hesaplanması ile bulunur.

$$c(s) = \min_{\substack{j_l \in s \\ s - j_l olurlu}} [c(s - j_l) + \Delta[c(s - j_l), t_l]] \qquad \textbf{(5.1)}$$

Bu şekilde, tüm olurlu kümelerin mâliyeti bulunur ve buradan yararlanarak tüm iş öğelerinin atanması durumundaki optimum çözüm bulunur.

Gutjahr ve Nemhauser (1964), Held ve diğerlerinden bir yıl sonra, en kısa yol modelini montaj hattı dengeleme problemine uyarlamışlar. Burada amaç, toplam boş süreyi enküçükleyecek istasyon sayısını bulmaktır. Makalede tüm hat dengeleme probleminin tek bir en kısa yol problemi olarak nasıl formüle edileceği anlatılmıştır. Oluşturulan en kısa yol modelinde düğümler s olurlu kümeleri, yâni durumları ifâde etmektedir. Başlangıç düğümü boş küme, bitiş düğümü ise tüm iş öğelerini içeren kümedir. Gutjahr ve Nemhauser'e göre, ağ içerisindeki en kısa yolu bulmak için, başlangıç düğümünden bitiş düğümüne giden herhangi bir minimal düğüm içeren yolu bulmak yeterlidir. Bu yolu bulmak için de başlangıç düğümünden başlanarak düğümler üzerinden olası bağlar ile, öncelik ilişkileri ve çevrim süresi gözönünde tutularak, son düğüme doğru ilerlenir. Erişilen düğümler, ilk düğümler olarak adlandırılır. Son düğüme ilk erişildiği andan geriye doğru gidilerek, istasyonlara yapılacak atamaların sırası bulunur. Bu işlemlerin yapılabilmesi için de öncelikle olurlu kümelerin yâni düğümlerin oluşturulması gerekir. Oluşturma aşamasında da Held ve diğerlerinin kullandığı yönteme benzer bir yöntem kullanılmıştır. Öncülü olmayan iş öğelerinin atanması ile başlanır ve bu iş öğelerinin ardılları, o aşama için "işaretlenmemiş" olarak belirlenir. Sonraki aşamada "işaretlenmemiş" olan iş öğeleri, "işaretlenmiş" kabul edilir ve tüm alt-kümelerin önceki aşamadaki olurlu kümelerle bileşimi oluşturularak o aşamanın olurlu kümeleri elde edilir. Tüm iş öğelerini içeren durum elde edilene kadar işlem sürdürülür. Böylece tüm olurlu kümeler oluşturulmuş olur.

Kao ve Queyranne (1982), montaj hattı dengeleme problemlerinde DP yöntemlerini karşılaştırdıkları makalelerinde, DP yineleyen fonksiyonunu şu şekilde ifâde etmişlerdir:

$$c(s) = \min_{j \in Q(s)} \{c(s), c(s-j) + \Delta(c(s-j), t_j)\} \qquad \textbf{(5.2)}$$

Burada $Q(s)$, s olurlu kümesi için tanımlanan bir karar kümesidir ve $Q(s) = \{j \in s \mid s - j\}$ şeklinde tanımlanmıştır. Eşitliğin sağ tarafındaki $c(s)$ terimi kaldırılırsa, formülasyon Held ve diğerlerinin önerdikleri formülasyonla aynı duruma gelir.

Kao ve Queyranne, makalelerinde DP formülasyonunda yararlanılacak olurlu kümelerin ve kümeler hakkındaki bilgiler için bir adres yerine geçecek etiketlerin oluşturulmasına yönelik Schrage-Baker (1978) ve Lawler (1979) yaklaşımlarını karşılaştırmışlardır.

Schrage ve Baker, bu yaklaşımı basit montaj hattı dengelemenin özel bir durumu olan iş sıralama problemi için önermişlerdir. Olurlu kümeler iş öğesi numarası sıra düzeniyle oluşturulmuştur. Olurlu kümeler için etiketler ise, kümelerin içerdiği iş öğelerinin etiketleri yardımıyla oluşturulmaktadır ($L(s)$ olurlu kümesinin etiketi, $L(j)$ iş öğesinin etiketidir). $L(s) = \sum_{j \in s} L(j)$. Böylece $L(s)$, s kümesi için özgün olmaktadır (Baybars 1986). Bu etiketleme yöntemi için bellek gereksinimi, ortalama problemler için bile büyük olduğundan, Kao ve Quayranne bu yöntemin bir varyasyonunu geliştirmişlerdir. Makalelerinde Schrage-Baker yöntemini, kendi geliştirdikleri varyasyonu ve Lawler yaklaşımını bellek gereksinimi ve işlem süresi açısından belirli test problemleri üzerinden karşılaştırdıklarında, Lawler yaklaşımının diğer iki yönteme göre, işlem süresi ve bellek gereksinimi açılarından daha verimli olduğunu görmüşlerdir.

Henig (1986) ise makalesinde, üretim mâliyetlerini enküçükleme sorununu iki ölçüt açısından ele almıştır: Çevrim süresi ve iş istasyonu sayısı. Henig, Gutjahr ve Nemhauser'in oluşturduğu DP yöntemini genişleterek, verilen istasyon sayısına göre en küçük çevrim süresini bulmayı sağlayan bir yöntem geliştirmiştir. Böylece her istasyon sayısı için en küçük çevrim süresi bilineceğinden, üretim mâliyetini enküçükleyen çevrim süresi-iş istasyonu sayısı çifti rahatlıkla seçilebilecektir. Örneğin n sayıda iş öğesinin olduğu bir hatta, en fazla n sayıda çevrim süresi-iş istasyonu çifti olur ve bu çiftler birbirleriyle karşılaştırılarak en ekonomik çift belirlenir.

Algoritma gereği bir büyük istasyon sayısına yapılan atamalar sonucu elde edilen en küçük çevrim sürelerinin hesaplanabilmesi için, bir küçük istasyon sayısının çevrim sürelerinden yararlanılmaktadır. Bu da, DP'nın en büyük sorunu olan hesaplama ve bellek yükünü bir kez daha karşımıza çıkartmaktadır.

Henig, yukarıdaki çalışmasına ek olarak, işlem sürelerinin stokastik olduğu durumda, DP modelinin istasyon sayısını ve kritik çevrim süresini optimize etmedeki yeterliliğini de göstermiştir.

Bard (1989) ise makalesinde, paralel istasyonlu montaj hattı dengeleme probleminde dinamik programlama uygulamasını ele almıştır. Sunulan algoritmada hem iş öğelerinin hem de istasyonların paralel yapıldığı durum ve istasyonlardaki verimsiz (ölü) zamanlar hesaba katılmıştır. Burada ölü zaman ile kastedilen, istasyonun üretim için kullanılmadığı süredir (Örneğin robotun montajı yapılan parçaya doğru yaklaşırken harcadığı süre).

Bard'a göre, paralelleme durumunda, serî hat durumundaki kuramsal en az istasyon sayısından daha az sayıda istasyonla dengeleme gerçekleştirilebilir.

Bard, makalesinde, Held ve diğerleri tarafından önerilen DP formülasyonunu ve olurlu kümelerin oluşturulması ve etiketlenmesi için de Schrage-Baker yaklaşımını kullanmıştır. Bard, önerdiği formülasyonda, 2-düzeyli parallelleme durumunda oluşan ek mâliyetleri hesaba katmıştır.

Easton (1990), makalesinde, montaj hattı dengelemede DP'nin getirdiği işlem yükünü hafifletmek, kullanılmayacak kısmî sonuçları elemek için diğer araştırmacılar tarafından daha önceden öne sürülmüş olan eleme yöntemini geliştirerek, statik yerine dinamik bir üst-sınırdan yararlanmayı önermiştir.

Easton, makalesinde Lawler yaklaşımının kullanıldığı, Kao ve Queyranne'ın önerdiği DP yöntemi ile ilgilenmiştir.

$$c(s) = \min_{j \in Q(s)} \{c(s), c(s-j) + \Delta(c(s-j), t_j)\}$$

Easton geliştirdiği prosedürü, Morin ve Marsten (1976)'in, DP hakkında gezgin satıcı ve doğrusal olmayan sırt çantası problemlerine benzer bir gözlem yaptıkları "Branch and Bound Strategies for Dynamic Programming" adlı makalelerine dayandırmıştır. Morin ve Marsten'in önerdiği yöntem, montaj hattı dengeleme probleminde şu şekilde işlemektedir: Yöntem, problemin olurlu bir çözümünün bilinmesine dayanmaktadır. Montaj hattı dengelemede mâliyetler, istasyon sayısı (N) ile orantılı olduğundan, istasyon sayısı olarak ifâde edilmektedir. Bilinen bir çevrim süresi ve olurlu çözümün verdiği istasyon sayısı durumunda, olurlu kümeler ağının en kısa yol çözümü

$F(s_f) \leq CN$ olarak gösterilmektedir.

Montajı s durumunda (olurlu kümesinde) tamamlamak için gereken ek süre $(l(s))$ için alt-sınır, s 'nin elemanı olmayan iş öğelerinin toplam süresi olarak gösterilir.

$$l(s) = \sum_{j \notin \{s\}} t_j \qquad \textbf{(5.3)}$$

Buradan hareketle en kısa yol üzerinde olan bir durumun şu koşulu yerine getireceği kesindir: $F(s) + l(s) \leq CN$

Morin ve Marsten, bu koşulu sağlamayan bir s durumunun eleneceğini ve daha sonra da dikkate alınmayacağını belirtmişlerdir. Ama zâten N sayıda iş istasyonu veren bir sonucun elde olduğu, asıl istenenin N'den daha az sayıda istasyona gereksinim duyulan bir çözüm olduğu düşünülünce, yukarıdaki eşitlik koşulunu sağlayan durumların da elenebileceği söylenmiş, yâni yeni koşul

$$[(F(s) + l(s))/C]^+ < N \qquad \textbf{(5.4)}$$

olmuştur.

Morin ve Marsten, ayrıca DP'nın herhangi bir K aşamasında, $S'(K)$ (yeni koşulu sağlayan K elemanlı olurlu kümelerin oluşturduğu küme) boş küme ise, başlangıçta sezgisel olarak bulunan olurlu çözümün optimal çözüm olduğunu söylemişlerdir.

$l(s)$ sınırı da, son istasyonun olası en az âtıl süresini ekleyerek iyileştirilmiştir:

$l(s) \leftarrow l(s) + SLK(s),$

$$SLK(s) = \begin{bmatrix} C - F(s) \bmod C & F(s) \bmod C + \min\limits_{\substack{j \notin s \\ s-j \text{ olurlu}}} \{t_j\} > C \\ 0 & diger \end{bmatrix} \qquad \textbf{(5.5)}$$

DP'nin montaj hattı dengeleme probleminde eleme ve statik üst-sınır ile uygulanması, işlem yükünü azaltmak, sonuca daha çabuk ulaşmak için etkili bir yöntemdir. Ama Easton, başlangıç olurlu çözümünün optimal olmaması durumunda, eleme yönteminden istenen verimin alınamayacağını, işlem yükünü azaltmak isterken, aksine arttıracağını söylemektedir. Bunun önüne geçmek için de üst-sınırın statik değil dinamik olmasını savunmuştur. Easton'ın önerdiği yöntemde, eleme durumlarının bağıl başarısı, başlangıç çözümünün kalitesi için gösterge olarak kullanılmaktadır. Kontrol edilen durumlardan çok azı eleniyorsa ve bu durumda üst-sınırın iyiliğinden kuşku duyuluyorsa, bulunan elenmemiş durumdan çözümü sürdürmek üzere bir sezgisel daha uygulanabiliyor, böylece üst-sınır geliştiriliyor. Eğer geliştirme sezgiselinin uygulandığı s durumu en kısa yol üzerinde ise uygulanan sezgiselin optimal çözümü verme olasılığı da artmış oluyor.

Burada dikkat edilmesi gereken nokta, bu üst-sınır geliştirme prosedürünün, işlem yükünü arttırmamak için ne sıklıkla yineleneceği ve hangi sezgisellerin kullanılacağıdır. Easton başlangıçta ve geliştirme amacıyla kullanılacak sezgisellerin ikisinin de birbirinden farklı olmasını ve ikisinin de hızlı ve iyi sonuç veren sezgiseller olması gerektiğini söylemiştir. Eleme başarı oranı eşiği olarak da uygun bir sayının seçilmesi gerektiğini belirtmiştir.

Easton, DP tekniğini, statik elemeli DP tekniğini ve önerdiği dinamik elemeli DP tekniğini, Talbot ve Patterson'un problem kümesindeki 60 probleme uyguladığında, deney için belirlenen parametreler içinde tüm problemleri çözen tek yöntemin dinamik elemeli DP olduğunu görmüştür.

5.3. Dal-Sınır Algoritması

5.3.1. Dal-Sınır Algoritması Hakkında Genel Bilgi

Dal-sınır algoritmaları, tüm olurlu çözümleri değerlendirerek en iyi çözüme ulaşırlar **(Aase ve diğ., 2003)**. Dal-sınır algoritmaları iki temel bileşenden oluşurlar: dallandırma (numaralandırma, sayma) ve sınırlama. Yöntemin performansını arttırmak için baskınlık ve azaltma kurallarından da yararlanılır. Sınırlama, baskınlık ve azaltma kuralları, hep birlikte eleme kuralları olarak da adlandırılabilir. Bu kuralların uygulanması sonucu, üzerinden dallandırmanın sürdürülmesinde yarar görülmeyen düğümler elenir **(Scholl, 1999)**.

5.3.2. Dallandırma

Dallandırma aşamasında başlangıç düğümünden başlanarak her adımda olası istasyon yükleri oluşturulur. Başlangıç düğümü 0 düzeyinde kabul edilir. Ağacın herhangi bir k düzeyinde ilk k istasyona atama yapılmış olur. Buna kısmî çözüm denir. Henüz ataması yapılmamış iş öğelerinin öncelik diyagramı da indirgenmiş problemi gösterir. Başlangıç düğümünden başlayıp süren her yola "dal" denir **(Scholl, 1999) (Scholl ve Klein, 1999a)**.

Dal-sınır algoritmaları, dallandırmanın hangi düğümden süreceği konusunda farklılık gösterirler. Bu konuda üç farklı ağaç oluşturma stratejisi vardır: En iyi ilk, lazer tipi, derinlik önce arama.

En iyi ilk stratejisinde, oluşturulan tüm olurlu düğümler bir aday listesine eklenir. Bu listedeki en yüksek önceliğe sahip olan düğüm bir sonraki dallandırma düğümü olarak seçilir. Seçilmeyen düğümler aday listesinde tutulurlar. Aday listesinde ağacın farklı düzeylerinden düğümler olabilir.

Lazer tipi dallandırma stratejisinde, her düzeydeki dallandırma, son oluşturulan düğümden itibaren sürer. Böylece olurlu bir çözüme hemen ulaşılır. Bulunan çözüm en iyi çözüm değilse, varolan düğümlerden geriye doğru gidildikçe düzeylerdeki diğer seçenek düğümler oluşturulur. Böylece en iyi çözüme ulaşılmaya çalışılır.

Derinlik önce arama stratejisinde, dallandırma için seçilen son düğümden bir sonraki düzeydeki tüm olası düğümler oluşturulur. Bu aynı düzeydeki aday düğümlerden en yüksek önceliğe sahip olan düğüm bir sonraki düzeye, dallandırma işlemi için seçilir. Diğer adaylar bir listede tutulurlar ve düğüme her geri dönüldüğünde öncelik sırasına göre oluşturulurlar **(Aase ve diğ., 2003)**.

5.3.3. Sınırlama

Sınırlama, dal-sınır algortimasında ağacın boyunu küçültmek, olası istasyon atama küme sayılarını azaltmak için kullanılır. Bu amaçla, problem için alt ve üst-sınır bulunmaya çalışılır. Üst-sınır olarak 0 düzeyinde iş öğesi sayısı veya sezgisel bir yöntemle bulunmuş gereksinim duyulan istasyon sayısı kullanılır. Problemin düzeylerinde ilerlendikçe, o an geçerli en iyi çözümün istasyon sayısı, üst-sınır olarak kabul edilir.

Alt-sınır ise problemin olurlu bir çözüm vermesi için gereksinim duyulan en az istasyon sayısını ifâde eder. Bu işlem 0 düzeyinde yapıldığında bulunan alt-sınır, "global alt-sınır" olarak adlandırılır. Her düğümde yinelendiğinde ise bulunan alt-sınır "yerel alt-sınır" adını alır. Alt-sınırın bulunması için çeşitli yaklaşımlar vardır. En sağlıklı alt-sınır değeri, birkaç yaklaşımın kullanılması sonucu elde edilen değerlerden en büyüğünün alınması ile elde edilir. Bir problemin en iyi çözümünün istasyon sayısı global alt-sınırdan daha büyük olabilir.

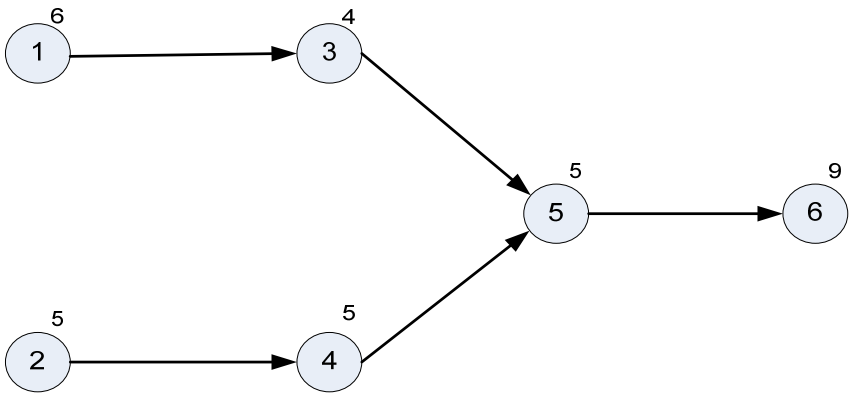
BMHDP'de en çok kullanılan alt-sınır yaklaşımları şunlardır:

- **AS1:** Kuramsal en az istasyon sayısı olarak da adlandırılan AS1, ($p_i = t_i / C$) iken şu şekilde hesaplanır:

$$AS1 = \left[\sum_{i=1}^{n} p_i \right]^{+} \quad \textbf{(5.6)}$$

Şekil 5.1'deki örnek problem için AS1 şöyle hesaplanır:

$$AS1 = \left[\sum_{i=1}^{6} p_i \right]^{+} = [0,6 + 0,5 + 0,4 + 0,5 + 0,5 + 0,9]^{+} = 4$$



Şekil 5.1: Öncelik Diyagramı (C = 10 dk/adet)

- **AS2:** İşlem süreleri C/2'den büyük olan iş öğelerinin aynı istasyona atanamayacağı gerçeğine dayanır. Çevrim süresinin yarısına eşit işlem süresine sahip iş öğelerinin sayısını da dikkate alır.

J tüm iş öğelerinin oluşturduğu kümenin alt kümesi,

$J(x, y) = \{i \in J \mid x < p_i < y\}$

$J(x, y] = \{i \in J \mid x < p_i \leq y\}$

$J[x, y] = \{i \in J \mid x \leq p_i \leq y\}$ iken,

$$AS2 = \left| J\left(\frac{1}{2},1\right] \right| + \left[\frac{1}{2} * \left| J\left[\frac{1}{2},\frac{1}{2}\right] \right| \right]^{+} \quad \textbf{(5.7)}$$

olarak hesaplanır.

Şekil 5.1'deki örnek problem için AS2 şöyle hesaplanır:

$$AS2 = 2 + [3/2]^{+} = 4.$$

- **AS3:** AS2'dekine benzer bir düşünceyle ama bu sefer çevrim süresinin 1/3'ü ve katlarının aralığında kalan işlem süresine sahip iş öğelerinin sayısı kullanılarak hesaplanır. Bu aralıklarda kalan iş öğelerinin en fazla kaçının bir istasyona atanabileceği düşünülerek çarpanlar belirlenmiştir.

$$AS3 = \left[\left| J\left(\frac{2}{3},1\right] \right| + \frac{2}{3} * \left| J\left[\frac{2}{3},\frac{2}{3}\right] \right| + \frac{1}{2} * \left| J\left(\frac{1}{3},\frac{2}{3}\right) \right| + \frac{1}{3} * \left| J\left[\frac{1}{3},\frac{1}{3}\right] \right| \right]^{+} \quad \textbf{(5.8)}$$

olarak hesaplanır.

Şekil 5.1'deki örnek problem için AS3 şöyle hesaplanır:

$AS3 = [1 + 0 + (5/2) + 0]^{+} = 4$ olarak bulunur.

- **AS4:** AS4'te dengeleme problemi, tek-makina sıralama problemi olarak düşünülür. İş öğeleri p_i sürelerine sahip işler olarak düşünülür. AS4 hesaplanırken yukarıda sözedilen üç sınırlama yaklaşımından da yararlanılır.

$$N_{i1} = AS1(F_i^*) \quad \textbf{(5.9)}$$

Burada $AS1 = \sum_{i=1}^{n} p_i$ olarak hesaplanır.

$$N_{i2} = \begin{cases} LB2(F_i^*) & eger\ p_i \geq \frac{1}{2}\ veya\ LB2(F_i^*) \notin N \\ LB2(F_i^*) - \frac{1}{2} & diger\ durumda \end{cases} \quad \textbf{(5.10)}$$

Burada $AS2 = \left| J\left(\frac{1}{2},1\right] \right| + \frac{1}{2}\left| J\left[\frac{1}{2},\frac{1}{2}\right] \right|$ olarak hesaplanır.

$$N_{i3} = \begin{cases} LB3(F_i^*) & eger\ p_i \geq \frac{2}{3} \\ LB3(F_i^*) - \frac{1}{3} & diger\ durumda \end{cases} \quad \textbf{(5.11)}$$

Burada $AS3 = \left| J\left(\frac{2}{3},1\right] \right| + \frac{2}{3}\left| J\left[\frac{2}{3},\frac{2}{3}\right] \right| + \frac{1}{2}\left| J\left(\frac{1}{3},\frac{2}{3}\right) \right| + \frac{1}{3}\left| J\left[\frac{1}{3},\frac{1}{3}\right] \right|$ olarak hesaplanır.

$[h_1, h_2, ..., h_r]$ iş öğelerinin yuvarlanmış değerlerine göre azalan şekilde sıralanmasıyla elde edilen sırayı belirtir.

$$N_{i4} = Enb\left\{ p_{h_1} + n_{h_1}, p_{h_1} + p_{h_2} + n_{h_2}, ..., p_{h_1} + p_{h_2} + ... + p_{h_r} + n_{h_r} \right\} \quad \textbf{(5.12)}$$

$N_i = Enb \{N_{i1}, N_{i2}, N_{i3}, N_{i4}\}$ **(5.13)**

Eğer $p_i + N_i > [N_i]^+$ ise, N_i en yakın büyük tamsayıya yuvarlanır ve bu hâli kullanılır.

1. iş öğesinin yuvarlanmış değeri AS4'ü verir.

Şekil 5.1'deki örnek problem için AS4 değeri şöyle hesaplanır (Tablo 5.1.'de AS4 hesap tablosu görülmektedir):

Tablo 5.1: AS4 Hesap Tablosu

	6	5	4	3	2	1	0
p_i	0,9	0,5	0,5	0,4	0,5	0,6	0
N_{i1}	0	0,9	1,4	1,4	1,9	1,8	3,4
N_{i2}	0	1	1,5	1,5	2	1,5	3,5
N_{i3}	0	0,67	1,17	1,17	1,67	1,67	3,5
N_{i4}	0	0,9	1,5	1,5	2	1,9	3,5
N_i	0	0,9	1,5	1,5	2	1,9	3,5
Yuvarlanmış değer	0	1	1,5	1,5	2	2	4

$N_{64} = 0$

$N_{54} = Enb \{0,9 + 0\} = 0,9$

$N_{44} = Enb \{0,5 + 1 ; 0,5 + 0,9 + 0\} = 1,5$

$N_{34} = Enb \{0,5 + 1 ; 0,5 + 0,9 + 0\} = 1,5$

$N_{24} = Enb \{0,5 + 1,5 ; 0,5 + 0,5 + 1 ; 0,5 + 0,5 + 0,9 + 0\} = 2$

$N_{14} = Enb \{0,4 + 1,5 ; 0,4 + 0,5 + 1 ; 0,4 + 0,5 + 0,9 + 0\} = 1,9$

$N_{04} = Enb \{0,6 + 2 ; 0,6 + 0,5 + 2 ; 0,6 + 0,5 + 0,4 + 1,5 ; 0,6 + 0,5 + 0,4 + 0,5 + 1,5 ; 0,6 + 0,5 + 0,4 + 0,5 + 0,5 + 1 ; 0,6 + 0,5 + 0,4 + 0,5 + 0,5 + 0,9 + 0\} = 3,5$

Buradan AS4 = 4 olarak hesaplanır.

- **AS6:** AS2 ve AS3'ün mantıkları kullanılır ve genişletilir. 4 adım izlenerek bulunur.

Adım 1. $J\left(\frac{1}{2},1\right]$'de yer alan iş öğelerini artmayan işlem süresi sırası ile 1.'den $d_1 = \left|J\left(\frac{1}{2},1\right]\right|$ inci istasyona kadar ata.

Adım 2. $J\left(\frac{1}{3},\frac{1}{2}\right]$'de yer alan iş öğelerini azalmayan işlem süresi sırası ile, uygun âtıl süreye sahip ilk istasyondan başlayarak birer birer, iş öğeleri veya d_1 istasyon bitene kadar ata.

Adım 3. 2. Adımda istasyonlara atanamayan iş öğesi sayısı d ise $d_2 = [d/2]^+$ olur.

Adım 4. $J\left[0,\frac{1}{3}\right]$'de yer alan iş öğeleri için d_3 sınırı hesaplanır.

$$d_3 = maks\left\{0, maks\left\{d_3(q) \middle| q \in \left\{p_i \middle| i \in J\left[0,\frac{1}{3}\right]\right\}\right\}\right\} \quad \textbf{(5.14)}$$

$AS6 = [d_1 + d_2 + d_3]^+$ olarak hesaplanır **(Scholl, 1999)**.

Şekil 5.1'deki örnek problem için AS6 değeri şöyle hesaplanır:

Adım 1. A1 = {6} , A2 = {1} ➔ $d_1 = 2$

Adım 2. A1 = {6} , A2 = {1, 3} ➔ 2, 4 ve 5 no.lu iş öğeleri açıkta kalır. d = 3

Adım 3. $d_2 = [3/2]^+ = 2$

Adım 4. $d_3 = 0$

$AS6 = [2 + 2 + 0]^+ = 4$ olarak hesaplanır.

- **Çiftlenmiş İş Öğeleri Alt-sınırı (AS5):** Bu alt-sınır, istasyon başına ortalama bir veya iki iş öğesinin düştüğü problemler içindir. Beş adım ile bulunur:

Adım 1. Atanmamış tüm iş öğeleri artan işlem sürelerine göre dizilirler.

Adım 2. Listedeki ilk ve son iş öğesinin işlem süresi toplanır. Eğer toplam süre çevrim süresinden büyükse veya listede salt tek bir atanmamış iş öğesi kalmışsa Adım 3'e geçilir. Toplam süre çevrim süresine eşit ise Adım 4'e geçilir. Diğer durumda Adım 5'e geçilir.

Adım 3. Listedeki son iş öğesi tek başına bir sonraki istasyona atanır ve listeden çıkartılır. Tüm iş öğeleri atanmışsa durulur. Diğer durumda Adım 2'ye dönülür.

Adım 4. Listedeki hem ilk hem de son iş öğesi birlikte, sıradaki istasyona atanır. Tüm iş öğeleri atanmışsa durulur. Diğer durumda Adım 2'ye dönülür.

Adım 5. Listedeki ilk, ikinci ve son iş öğesinin işlem süreleri toplamı belirlenir. Eğer toplam, çevrim süresine eşit veya çevrim süresinden daha azsa Adım 6'ya geçilir. Diğer durumda, listedeki son iş öğesi ile bu iş öğesi ile aynı istasyona atandığında çevrim süresini aşmayacak olup, listedeki en büyük işlem süreli iş öğesi aynı istasyona atanır. Adım 2'ye dönülür.

Adım 6. Adım 5, üç iş öğeli olurlu bir istasyon atamasını oluşturur. Prosedür durdurulur ve kalan atanmamış iş öğeleri için AS1 kullanılarak alt-sınır belirlenir **(Aase ve diğ., 2003)**.

Şekil 5.1'deki örnek problem için AS5 değeri şöyle hesaplanır:

Adım 1. 3-2-4-5-1-6

Adım 2. $t_3 + t_6 = 13 > C$

Adım 3. A1 = {6} ➔ 3-2-4-5-1

Adım 2. $t_3 + t_1 = 10 = C$

Adım 4. A2 = {1, 3} ➔ 2-4-5

Adım 2. $t_2 + t_5 = 10 = C$

Adım 4. A3 = {2, 5} ➔ 4

Adım 3. A4 = {4}

Buradan AS5 = 4 olarak bulunur.

5.3.4. Baskınlık ve Azaltma Kuralları

Baskınlık ve azaltma kuralları, ağacın boyutlarını küçültmek, seçenek istasyon atama kümelerinin sayısını azaltmak için kullanılan mantıksal testlerdir. Bu kurallar yardımıyla problem basitleştirilir, varolan çözümden daha iyi bir çözüme, yâni en iyi çözüme götürmeyecek olan düğümler elenir. Başlıca baskınlık ve azaltma kuralları şunlardır:

- **İş Öğesi Süresi Arttırma Kuralı:** Herhangi bir iş öğesinin işlem süresi, bu iş öğesinin, öncelik ilişkilerinden bağımsız olarak, en küçük işlem süreli iş öğesi ile aynı istasyona atanmasına izin vermeyecek kadar büyükse, yâni bu iş öğesi,

öncelik ilişkilerinden bağımsız olarak, bir istasyona tek başına atanmak zorunda ise, bu iş öğesinin işlem süresi, çevrim süresi olarak değiştirilir. Şekil 5.1'deki örnekte yer alan 6 numaralı iş öğesinin süresi, bu kural dahilinde (C = 10) olarak kabul edilebilir.

- **En Büyük Yük Kuralı:** Bir veya daha fazla tamamlanmış ama en fazla şekilde yüklenmemiş kısmî sonuçları eler, çünkü aynı istasyon sayısına sahip ama daha fazla iş öğesi içeren en az bir kısmî sonuç kesinlikle vardır. Tüm istasyonların oluşturulurken en üst düzeyde yükle yüklenmesine dikkat edilir.

- **Jackson Baskınlık Kuralı:** Eğer $F_i \subseteq F_h$ ve $t_i \leq t_h$ ise, h. iş öğesi i. iş öğesini baskılar. Eğer iki iş öğesinin işlem süreleri ve izleyen ardılları aynıysa, küçük numaralı iş öğesi, büyük numaralı iş öğesini baskılar. Jackson baskınlık kuralına göre, eğer k. iş öğesi m. iş öğesini baskılıyorsa, m. iş öğesi bir en üst düzeyde istasyon yükü S_b kümesine atanmışsa ve k. iş öğesi henüz hiç bir istasyona atanmamışsa, S_b kümesinden m. iş öğesi çıkartılıp, yerine k. iş öğesi atandığı varsayıldığında istasyon süresi çevrim süresini aşmıyorsa, S_b olası istasyon yükü elenebilir. Bu kural, k. ve m. iş öğelerinin ardıllarının yapılabilmesi için k. ve m. iş öğelerinin her ikisinin de yapılması gerektiği ama yapılış sırasının önemli olmadığı gerçeğine dayanır **(Scholl, 1999)**. Şekil 5.1'deki örnek problemde, 3 ve 4 no.'lu iş öğelerinin ardılları aynı ve ($t_4 > t_3$) olduğundan, 4 no.'lu iş öğesi 3 no.'lu iş öğesini baskılar.

- **İlk İstasyon Baskınlık Kuralı:** 2 ve daha büyük numaralı istasyonların herhangi bir olası yükü, eğer daha önceden değerlendirilmiş ilk istasyon olası yüklerinin herhangi birinin alt kümesi ise değerlendirmeye alınmaz, elenir. Bu kural aynı istasyon yüklerine farklı sıralarla sahip olan çözümlerin eşit olduğuna dayanır.

- **Etiketleme Kuralı / Ağaç Baskınlık Kuralı:** Etiketleme kuralı salt iş öğesi sırasında farklılık gösteren aynı kısmî sonuçların yeniden bulunmasının önüne geçmek için kullanılır. İstasyonlara yapılmış tüm olurlu atama kümeleri için gereken istasyon sayısı bellekte tutulur. Daha önce ele alınmış bir olurlu atama kümesi için daha iyi bir sonuç bulunduğunda, yeni bulunan sonuç, daha da iyi bir sonuç bulunana kadar bu küme için bellekte tutulan sonuç olur. Ağaç baskınlık kuralı da aynı amacı güder ama değerlendirilmiş kümelerin sonuçlarını bellekte

tutmak için ağaç yapısından yararlanır. Bu da daha az oranda bellek kullanımı sağlar **(Scholl ve Klein, 1999a)**.

Şekil 5.1'deki örnek problem için etiketleme kuralı şöyle uygulanır:

Etiketleme baskınlık kuralı için iş öğelerinin etiketleri belirlenir.

i = 1, 2, …, n için

$L(i) = \sum_{h \in p_i'} L(h) + 1$ $\qquad p_i' = \{h \mid h < i \ ve \ h \notin P_i^*\}$

L(1) = 0 + 1 = 1

L(2) = L(1) + 1 = 2

L(3) = L(2) + 1 = 3

L(4) = L(1) + L(3) + 1 = 5

L(5) = 0 + 1 = 1

L(6) = 0 + 1 = 1

Bu etiketlerden yararlanarak, tüm olurlu atama kümeleri için, o kümeye özgü etiket değerleri belirlenir. Tüm olurlu atama kümelerinin etiket değerleri şu şekilde bulunur:

L(1, 2) = 3

L(1, 3) = 4

L(2, 4) = 7

L(1, 2, 3) = 6

L(1, 2, 4) = 8

L(1, 2, 3, 4) = 11

L(1, 2, 3, 4, 5) = 12

L(1, 2, 3, 4, 5, 6) = 13

- **Âcil İş Öğesi Atanması Baskınlık Kuralı:** Eğer bir iş öğesi atanmaya hazırsa ama atanmamış iş öğelerinden, öncelik ilişkilerine bakılmaksızın, herhangi biri ile çevrim süresi aşılmadan aynı istasyona atanamıyorsa, bir sonraki istasyona âcil olarak bu iş öğesi atanır, diğer olasılıklar dikkate alınmaz **(Aase ve diğ., 2003)**.

5.4. BMHDP'de Dal-Sınır Algoritması Uygulamaları Üzerine Literatür Araştırması

Van Assche ve Herroelen (1978)'in önerdiği dal-sınır algoritması, minimal alt-sınır stratejisini içeren bir algoritmadır. Prosedür, boş kümeye atanabilir iş

öğelerinin atanması ile başlar. Her düğüm, "varolan bir çözüm verecek mi ?"diye kontrol edilir (bir çözüm elde etmek için salt iki istasyon kalmışsa o düğüm elde varolan bir çözüm verecek düğümdür). Elde varolan bir çözümün olmadığı belli olduğunda bu düğümden dallanma, yeni olası iş istasyonlarının oluşturulması sürdürülür. Yeni istasyonlar (düğümler) oluşturulurken, Jackson Baskınlık Kuralı uygulanır. Her düğüm için bir istasyon alt-sınırı hesaplanır. En düşük alt-sınıra sahip düğüm bir sonraki iterasyon için seçilir. Aynı en düşük alt-sınıra sahip düğümler arasında bir eleme yapabilmek için ayrıca her düğüm için bir "ceza" hesaplanır. "Ceza", geriye kalan iş istasyonlarının ortalama zamanını süresini verir. Bir sonraki iterasyon için en düşük alt-sınıra sahip düğümler arasından en küçük "ceza"ya sahip olan düğüm seçilir. Aynı alt-sınır ve "ceza"ya sahip düğümler arasında bir seçim yapabilmek için ise farklı, eşitlik durumunda geçerli öncelik kuralları uygulanmıştır. Ağacın herhangi bir düğümü için varolan bir çözüm bulunursa, düğümün tüm öncül (ata) düğümleri için bir çözüm garantilidir.

Johnson (1981), bu makalesinde basit montaj hattı dengeleme problemlerinin çözümünde kullanılan optimizasyon algoritmalarını karşılaştırmıştır. Algoritmalar 20 iş öğesi ve 40 iş öğesinin sözkonusu olduğu iki farklı durumda, farklı çevrim süreleri ve sıralama güçleri açısından incelenmiştir. En iyi çözüm veren algoritmaların istasyon başına düşen iş öğesi sayısına göre değiştiği gözlenmiştir. Karşılaştırılan altı algoritma şunlardır: Johnson'ın kendi önerdiği algoritma, Held ve diğerlerinin önerdiği, Gutjahr ve Nemhauser'in önerdiği, Jackson'ın önerdiği ve bunun yeniden düzenlenmiş hâli, Charlton ve Death'in önerdiği dinamik programlama ve dal-sınır algoritmaları. Johnson'ın önerdiği algoritma, en yeni düğüm araştırması olarak nitelenen bir dal-sınır algoritması. Düğümleri bağlayan kanallar (arc) olurlu istasyonları simgeler. Her yol bir olurlu çözümü simgeler. Ağaç oluşturulurken salt maksimum yüklenmiş istasyonlar dikkate alınır. Her düğümden çıkan kanallar, olası maksimum yüklenmiş istasyonları simgeler. Bir sonraki düğüm, en az âtıl süreye sahip istasyon (ark) üzerinden gidilerek oluşturulur. Ulaşılan çözüm, geçerli en iyi çözüm olarak kabul edilir ve bu çözüm üzerinden her adımda bir düğüm olmak üzere geri gidilerek ve oluşturulmuş diğer olurlu maksimum yüklenmiş istasyonlar kullanılarak en iyi çözüm bulunmaya çalışılır.

Johnson (1983), makalesinde orijinal montaj hattı dengeleme probleminin iki değişik durumu için bir dal-sınır algoritması önermiştir. Bu iki değişik durumdan birincisi, istasyon sürelerinin plânlanmış dengesizliği, ikincisi ise iş öğelerinin özel bir tip istasyona atanma zorunluluğudur. Makalede bu iki değişik durum için uygun olan algoritmanın, başka yedi değişik durum için de kullanılabileceği öne sürülmektedir. Bu yedi değişik durum ise, sürekli hatlardaki stokastik iş öğesi süreleri, belirli bir istasyona belirli bir iş öğesinin atanması, nitelik derecesi, hattın-solu ve hattın-sağı iş öğeleri, iş öğelerinin farklı istasyonlara atanma zorunluluğu, karma modelli montaj hatları ve paralel istasyonlarda bulunan iş öğelerinin sözkonusu olduğu durumdur.

Saltzman ve Baybars (1987)'ın önerdiği algoritma bir derinlik önce arama algoritmasıdır. Başlangıç üst-sınırı bir sezgisel ile belirlenir (Wee ve Magazine'in IUFFD yöntemi), başlangıç alt-sınır ise AS1 ve AS5 ile. En büyük istasyon yükünün bulunmasından sonra sezgisel ile tamamlanan problem yardımıyla daha uygun bir üst-sınır bulunmaya çalışılır. Son maksimum yüklenmiş istasyon atamasının yerel alt-sınırının veya global alt-sınırın, varolan üst-sınırı aştığı, son maksimum yüklenmiş istasyon atamasının âtıl sürenin varolan âtıl süre sınırını aştığı, düğümün daha iyi sonuç veren bir istasyona atanamadığı durumlarda sözkonusu düğüm elenmektedir. Bu algoritma probleme aynı anda hem ileri yönde hem de geri yönde uygulanır. Her süreç sırası ile bir düğümü inceler. Problem ile ilgili öncelik diyagramı gibi ortak bilgiler ve sürecin ilerlemesi ile değişebilen varolan en iyi üst ve alt-sınırlar gibi ortak bilgiler paylaşılan bir bellekte tutulur ve buradaki bilgiler her iki süreç tarafından güncellenir.

Johnson (1988) makalesinde, toplamda 1.000 civarı ve istasyon başına yaklaşık 6 iş öğesinin bulunduğu Tip-1 BMHDP montaj hatlarının optimal dengelenmesi için geliştirdiği "Fable" isimli algoritmayı açıklamıştır. "Fable" lazer tipi, bir derinlik önce dal-sınır algoritmasıdır. "Fable" 'ın özelliği, optimum sonucu kısa süre içinde düşük bellek gereksinimi ile bulmasıdır. Bu da her seferde, çözüm ağacının tek dalının yaratılmasından kaynaklanmaktadır. Bir sonraki dalın oluşturulması için, önceki dalın düğümleri "yok edilmektedir". Dal oluşturma sırasında görece daha uygun iş öğelerinin seçilmesi için iş öğelerinin yeniden nasıl numaralandığı önemlidir. Eğer bir iş öğesi süresi nedeniyle istasyona atanmış tek iş öğesi olacaksa,

bu iş öğesinin süresi çevrim süresi olarak alınabilir. Düğümlerin elenmesinde ise 8 yöntemden sınır olarak yararlanılmıştır.

Hackman, Magazine ve Wee (1989), makalelerinde, Tip-1 BMHDP için sezgisel bir dal-sınır algoritması önermişlerdir. Dal-sınır algoritmasında en az alt-sınır dallandırma stratejisi izlenmiştir. Sezgisel ise önerilen yöntemin iki aşamasında kullanılmış, ilki sınırlama, ikincisi ise elemedir (ikincisi sâyesinde dal-sınır ağacının boyutu önemli oranda azalmış.). Sezgisel olarak, yapılan karşılaştırma sonucunda kabul edilebilir sonuç verenlerden olduğundan, IUFF6 yöntemi kullanılmıştır. Sezgiselden, sınırlama aşamasında istasyon sayısı üst-sınırı belirlenirken, eleme aşamasında ise iki baskınlık kuralı yanında düğümlerin sayısının fazla olması durumunda kimi düğümün sezgisel olarak elenmesinde yararlanılmıştır. Yapılan testlerde yöntemin iyi sonuçlar verdiği görülmüştür. Tip-1 BMHDP için kullanılan yöntemin Tip-2 BMHDP için nasıl uyumlandırıldığından da sözedilmiştir.

Berger, Bourjolly ve Laporte (1992)'nin makalesinde bir derinlik önce arama dal-sınır algoritması ele alınmıştır. Amaç, gerekli istasyon sayısını enküçüklemektir. Berger ve diğerleri, geliştirdikleri algoritmayı Hackman ve diğerlerinin (1989) çalışması üzerine kurmuşlardır. Farklı olarak daha güçlü bir alt-sınır ve dal-sınır araştırması için daha güçlü bir ayırma yordamı kullanmışlardır. Alt-sınır için AS1 veya AS6 kullanılırken üst-sınır 'immediate update first-fit (IUFF)' sezgiseli ile hesaplanmıştır (s_i olarak t_i alınmış). Çok fazla miktarda olurlu atama olasılığının olduğu durumlarda ise ayırma yordamı olarak (incelenecek olurlu atama olasılığını azaltmak için) Hackman, Wee ve Magazine'in yönteminin üzerine İlk Uyan Arttırma Kuralı, Berger vd.'nin düğüm oluşturma prosedürü ile tüm düğümleri oluşturma prosedürü uygulanmıştır. Atanma olasılığı olan iş öğeleri kümelerinin oluşturulmasında Schrage ve Baker (1978) yöntemi kullanılmıştır. Makalede ayırma kuralları ve alt-sınır kuralları birleştirilerek beş çeşit dal-sınır algoritması oluşturulmuştur.

Hoffmann (1992)'ın makalesinde Tip-1 BMHDP için, Hoffmann'ın önceden geliştirdiği sezgisel yöntemin de bir aşamasında kullanılan, bir dal-sınır optimizasyon algoritması (EUREKA) ele alınmıştır. Bu bir derinlik önce, lazer tipi algoritmadır. Algoritmada optimal çözüme götürmeyen düğümlerin elenmesinde sınır olarak 'kuramsal en az toplam atıl süre'den yararlanılmıştır. Oluşturulan her

düğüm için birikimli istasyon âtıl süreleri hesaplanır ve kuramsal en az toplam âtıl süre ile karşılaştırılır ve daha büyük olması durumunda düğüm elenir. Kuramsal en az istasyon sayısı için geçerli bir olurlu çözümün olmadığı durumda, en az istasyon sayısı bir arttırılarak işlemler yinelenir. Algoritmanın tümünde, yöntemin belirlenmiş bir süre içinde, öncelik diyagramında ileriye ve geriye doğru sıra ile uygulanmasına rağmen optimum sonucun bulunamadığı durumda, son olarak herhangi bir süre kısıtı olmadan Hoffmann'ın sezgisel yöntemi uygulanır.

Scholl ve Klein (1999a) ise makalelerinde en etkin dal-sınır algoritmaları olan FABLE, OptPack, Eureka ve SALOME'yi basit montaj hattı dengeleme problemi Tip-1 için kıyaslamışlar. Kıyaslama için tek bir hesaplama ortamı ve genelde kullanılan veri kümelerine kıyasla daha zor bir veri kümesi kullanmışlar. Sayısal kıyaslamadan önce tüm prosedürler dallandırma, sınırlama ve uyguladıklar mantıksal testler açısından karşılaştırılmışlardır. Sayısal karşılaştırma için yeni oluşturulan daha zor birleştirilmiş veri kümesinden başka bilinen veri kümeleri de kullanılmıştır. Basit problemlerde en hızlı çözüm veren yöntemin OptPack olduğu görülürken, genelde bakıldığında, özellikle zor problemlerde, SALOME'nin performansının en iyi olduğu ortaya çıkmıştır. SALOME'ye, ağaç baskınlık kuralı, Jackson baskınlık kuralının genişletilmiş hâli, düğümlerin dinamik numaralandırılması eklendiğinde, performansının daha da arttığı görülmüştür.

6. U-TİPİ MONTAJ HATLARI

6.1. Genel Bilgi

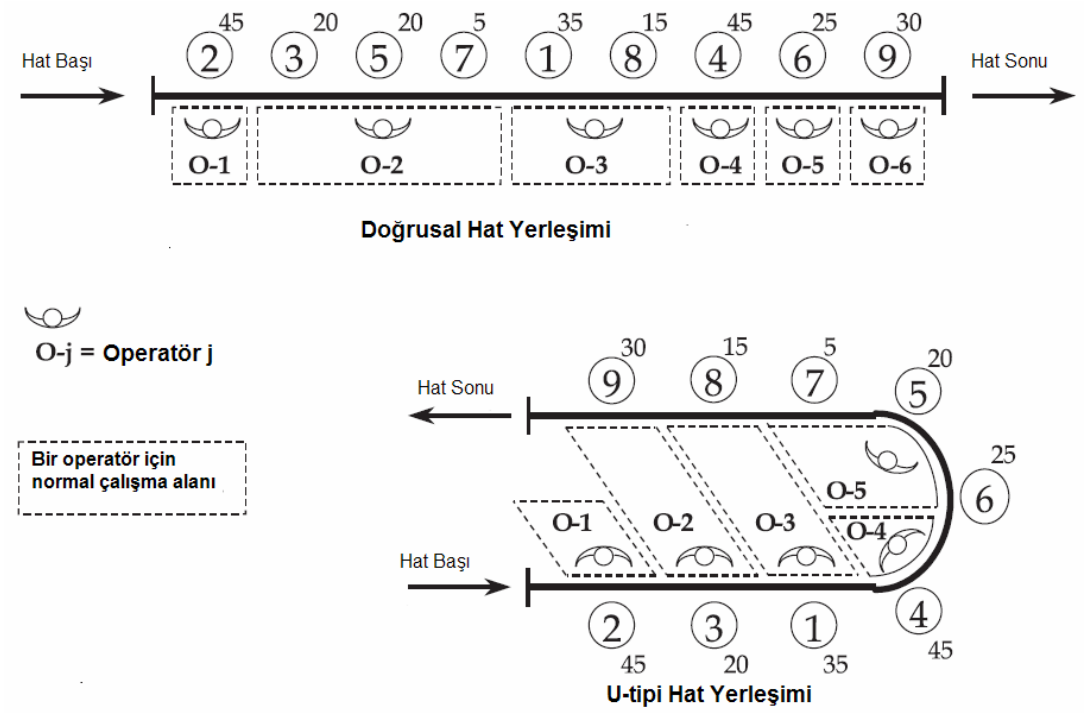
TZÜ sistemleri, daha önce de tanımladığımız gibi, ürün kalitesini arttırmak ve mâliyetleri düşürmek için üretim sürecindeki varolan israfları ortadan kaldırmayı amaçlar. U-tipi montaj hatları, günümüzde tam zamanında üretim sistemlerinin vazgeçilmez bir parçasıdır. U-tipi hatta, makinalar hattın çevresine konuşlandırılır. İşçiler ise hattın iç tarafında çalışırlar. U-tipi hatlarda çalışan işçilerin yüksek nitelikli olmaları, yâni birden çok daha fazla süreç içerisinde çalışabilecek durumda olmaları, U-tipi montaj hatlarının esnekliğinin dayandığı noktadır. Genelde hattın girişi ve çıkışından bir işçi sorumludur. Bu sâyede hattan bir ürün çıkmadan, yeni bir ürünün hatta girmesine izin verilmez. Bu sâyede hatta bir sorun olduğunda, hattın hemen durdurulması olanaklı olur. U-tipi hatlarda, ürünler ve işçiler saat yönünde veya tersi yönde hareket edebilir. U-tipi hatlarda çalışan işçi sayısı, hattaki istasyon sayısına bağlı olarak değişebilir. Tek işçinin çalıştığı U-tipi hatlar vardır. Birden fazla işçinin bir hatta birlikte çalıştığı durumlarda, işçilerin her biri baştan sona hat boyunca bir ürünün montajını gerçekleştirebilirler veya işçiler salt hat üzerindeki belli işlerden sorumlu olurlar. İkinci durum, U-tipi hatların uygulanmasında daha yaygındır **(Miltenburg, 2001)**.

U-tipi hatların doğrusal hatlara kıyasla çeşitli üstünlükleri vardır. Doğrusal hatlarda bir veya birkaç iş öğesinden sorumlu olan işçi, hattın belirli bir kısmında çalışırken, U-tipi hatlarda doğrusal hatlardan farklı olarak hattın farklı kısımlarında çalışabilmektedir, yâni hat üzerinde ardışık olarak bulunmayan iş öğelerinden sorumlu olabilirler. İşçilerin birbirleriyle yakın konumlanmış olması, işçiler arasındaki iletişim düzeyini yükseltir. İşçiler sorun yaşandığında iletişime geçebilir, birbirlerine yardım edebilir ve böylece sorunu kısa sürede çözebilirler. Ayrıca U-tipi hatlarda, hat işçiler tarafından daha rahat gözlemlenebilir, bu da kalite sorunları gibi sorunların çabuk saptanabilmesine ve anında müdahaleye olanak tanır.

U-tipi hatlarda işçilerin hat üzerindeki tüm işleri yapabilecek kadar nitelikli olmaları, doğrusal hatlar karşısındaki başka bir üstünlüktür. Bu sâyede işçiler talebin artıp azalmasına göre hatlara rahatlıkla atanabilir, bu da beraberinde yüksek oranda esnekliği getirir. Artan talebin, düşük nitelikli işgücü nedeniyle çevrim süresini değiştiremeyip, üretim saatini arttırarak karşılandığı doğrusal hatlara kıyasla, U-tipi hatlardaki bu esneklik sâyesinde talep değişiklikleri, hat üzerindeki çalışan sayısının değiştirilmesi ile kolayca çözülür. Ayrıca işçilerin bir bütün olarak yapılan iş hakkında bilgi sahibi olmaları, sürecin geliştirilmesine yardımcı olabilmelerini sağlar.

Ayrıca U-tipi hatların diğer üstünlükleri arasında; daha kolay malzeme taşıma, düşük stok, daha kolay üretim plânlama ve kontrol, kalite kontrol ve takım çalışması da sayılabilir **(Miltenburg ve Wijngaard, 1994)**.

U-tipi montaj hatları, tesislerde tek başlarına kullanılabildikleri gibi, esnekliği arttırmak ve işgücünden en üst düzeyde yararlanabilmek için, birkaç U-tipi hat bir arada veya daha karmaşık şekilde de kullanılabilir. Özellikle birden çok ürünün üretilmesi için bu tür hatlar yeğlenir. Karmaşık yapıdaki U-tipi hatlar, çok sayıda hattın oluşturduğu tek U-tipi hat, ikili bağımlı U-tipi hat, iç içe U-tipi hat gibi farklı yapılarda olabilirler. Çoklu U-tipi hat tesisi ise, bu anlamda ulaşılabilecek en karmaşık durumdur. Burada amaçlanan, her hatta olan, çevrim süresinden daha az istasyon süresine sahip istasyonları, U-tipi hattın ilk istasyonu yaparak ve birden çok U-tipi hattın bu düşük süreli istasyonlarını tek bir nitelikli işçiye vererek işgücünden en yüksek düzeyde yararlanmaktır. Böyle bir tesiste sözkonusu işçinin katedeceği mesafeler büyüdüğünden, mesafeleri alırken harcanan süreler de dikkate alınır **(Miltenburg, 2001)**. U-tipi montaj hatları ile doğrusal montaj hatları arasındaki yerleşim farkı, Şekil 6.1.'de gösterilmiştir.



Şekil 6.1: Doğrusal ve U-tipi Hat Yerleşimi Farkı **(Aase ve diğ., 2003)**

6.2. Basit U-tipi Hat Dengeleme Problemi (BUHDP)

Basit U-tipi hat dengeleme problemi, Basit Montaj Hattı Dengeleme Problemi'nden, iş öğelerinin istasyonlara atanması açısından farklılık gösterir. Basit montaj hattı dengelemede, istasyonlara atanacak iş öğeleri, öncülleri atanmış iş öğeleri arasından seçilir. Oysa U-tipi montaj hattı dengeleme probleminde ataması yapılacak iş öğeleri, öncülleri veya ardılları atanmış iş öğeleri arasından seçilir.

Basit U-tipi montaj hatları dengelenirken amaçlar farklı olabilir. Aynı BMHDP'de olduğu gibi, sabit bir çevrim süresi dahilinde istasyon sayısının enküçüklenmesinin amaçlandığı duruma Tip-1 BUHDP, sabit istasyon sayısı sözkonusu iken çevrim süresinin enküçüklenmesinin amaçlandığı duruma Tip-2 BUHDP denir.

Tip-1 BUHDP'de gereksinim duyulan istasyon sayısı, U-tipi hatta iş öğelerinin oluşturduğu olurlu küme olasılığı daha fazla olduğundan, hiç bir zaman hattın doğrusal hâlinde gereksinim duyulan istasyon sayısından fazla olamaz. Bu nedenle doğrusal hattın istasyon sayısı, U-tipi hatların istasyon sayısı için bir üst-sınır olarak alınabilir. Bu çalışmada Tip-1 BUHDP'de en iyi çözüm veren yöntemler ele alınacaktır.

Tip-1 BUHDP'nin matematiksel modeli **(Miltenburg ve Wijngaard, 1994)**:

$$\text{Enk}\left[N * C - \sum_{j=1}^{N} \sum_{k \in W_j} t_k \right]$$

$$\bigcap_{j=1}^{N} W_j = M$$

$$W_i \bigcap_{i \neq j} W_j = \phi$$

$$\sum_{k \in Wj} t_k \leq C \qquad \text{j} = 1, \ldots, \text{N} \qquad\qquad 1 \leq k \leq \text{n}$$

Her k iş öğesi için;

$$A_j \leq A_k \qquad \text{j} \in \text{P}_\text{k}$$

$$A_i \leq A_k \qquad \text{j} \in \text{F}_\text{k}$$

$$1 \leq A_i \leq \ddot{U}S$$

6.3. BUHDP'de Uygulanan Analitik Yöntemler

6.3.1. BUHDP'de Dinamik Programlama Uygulamaları Üzerine Literatür Araştırması

Miltenburg ve Wijngaard (1994) makalelerinde, DP'yi U-tipi montaj hatlarını dengelemek için kullanmışlardır. Tam zamanlı üretim ilkesinin bir uzantısı olan U-tipi hatların dengelenmesi, iş öğeleri öncelik ilişkisi diyagramı içerisinde ileriye, geriye veya aynı anda her iki yöne doğru gruplanabileceğinden, geleneksel montaj hatlarının dengelenmesinden daha zordur. Geleneksel montaj hattı dengeleme probleminde montaj hattı doğrusal hat olarak düşünülmüştür. Doğrusal hatlarda, iş öğelerinin iş istasyonlarına atanmasına ilk iş öğesi ile başlanır ve öncelik ilişkileri diyagramı içerisinde ileriye doğru işlem sürdürülür. U-tipi hatlarda ise istasyonlara atanan iş öğeleri, hattın farklı yerlerinden olabilirler.

Geleneksel hat dengeleme problemi ile U-tipi hat dengeleme problemi arasındaki temel fark şudur: Geleneksel hat dengeleme probleminde istasyonlara atanacak iş öğeleri, öncülleri daha önceden bir istasyona atanmış olan ve atanabilecek durumdaki iş öğeleri kümesinden seçilir. U-tipi hat dengeleme probleminde ise atanacak iş öğeleri, öncülleri daha önceden bir istasyona atanmış olan ve

atanabilecek durumdaki iş öğelerinin bulunduğu küme ile ardılları daha önceden bir istasyona atanmış olan ve atanabilecek durumdaki iş öğelerinin bulunduğu kümenin bileşiminden oluşan iş öğeleri kümesinden seçilir. Bu nedenle U-tipi montaj hattı dengelemede iki farklı tip olurlu küme sözkonusudur: İleriye olurlu kümede bulunan her iş öğesinin öncülü olan iş öğesi/öğeleri de kümenin elemanıdır, geriye olurlu kümede bulunan her iş öğesinin ise ardılı olan iş öğesi/öğeleri de bu kümenin elemanıdır. U-tipi montaj hattı dengelemede gözönünde tutulan olurlu kümeler de, ileriye ve geriye olurlu kümelerin bileşiminden oluşur. Buradan da olurlu dizilerin, ileriye olurlu diziler ile geriye olurlu dizilerin birleşimi olduğu söylenebilir. İleriye ve geriye dizilerin iş öğelerinin kendi aralarındaki sıraları değişmediği sürece, olurlu dizinin farklı permütasyonları olabilir.

Miltenburg ve Wijngaard, U-tipi hatların dengelenmesine yönelik geliştirdikleri prosedürde Held ve diğerlerinin DP yönteminden ve Schrage ve Baker'in olurlu küme oluşturma ve etiketleme prosedüründen yararlanmışlardır. Miltenburg ve Wijngaard'ın olurlu kümeleri oluşturmak için geliştirdikleri algoritmada önce, tıpkı Schrage-Baker'da olduğu gibi, ileriye olurlu kümeler oluşturulur. Daha sonra oluşturulan her geriye olurlu küme ile birlikte, ileri ve geriye olurlu kümelerin bileşimi olan nihaî olurlu kümeler oluşturulur. Burada da kullanılan DP yineleyen fonksiyonu, Held vd.'nin önerdiği ile aynıdır.

$$c(s) = \min\{c(s-j) + \Delta(j \mid s-j) \mid j \in R^f(s^f) \text{ veya } j \in R^b(s^b), s = s^f \cup s^b\}$$

Miltenburg ve Wijngaard, bu yöntemi 21 küçük boyutlu probleme uyguladıklarında, 19 problemde geleneksel montaj hattıyla aynı sayıda iş istasyonu veren çözüme ulaşmışlardır. İki problemde ise geleneksel hatta gereksinim duyulan istasyon sayısından 1 tane az sayıda istasyon sayısı veren çözüm bulunmuştur. Miltenburg ve Wijngaard, DP yöntemini daha büyük boyutlu problemler için uygulamamışlardır.

Miltenburg (1998), makalesinde çoklu U-tipi hatların bulunduğu tesislerde U-tipi hatların dengelenmesi problemini, birden fazla U-tipi hattan oluşan ve komşu U-tipi hatların kimi istasyonları paylaştığı durumda iş öğelerinin en az sayıdaki iş istasyonuna atanması problemi olarak ele almıştır.

Miltenburg, çoklu U-tipi hatların bulunduğu tesisler için üç tip iş istasyonu tanımlamıştır: Normal istasyon (geleneksel hatlardaki tüm istasyonlar normaldir. U-

tipi hatlarda büyük normal istasyonlar problem giderme ve iletişim sorunu yaratacağından istenmez, 5 veya 6 iş öğesi ile sınırlanırlar.), geçişli istasyon (hattın iki tarafında bulunan iş öğesi gruplarının birlikte atandığı istasyon. Operatör bu iki grup iş öğesi arasında gidip gelirken belirli bir mesafeyi kateder.), çoklu hat istasyonu (iki komşu hattan da iş öğeleri içeren iş istasyonu. Bu istasyonda ek olarak iki hat arasındaki mesafeler de katedilmektedir.). Bu makalede yapılan kabule göre bitişik hatlar arasında birden fazla çoklu istasyona izin verilmemiştir ve bu istasyon, ancak hatların başlangıç ve bitiş kısımlarında bulunabilir. Yâni bir hattın son istasyonundaki âtıl süre, bir sonraki hattın ilk istasyonundaki atama yapılabilen süre olur.

Miltenburg'un makalesinde belirttiğine göre, çoklu U-tipi hatların bulunduğu tesislerde U-tipi hatların dengelenmesi probleminde amaç, çevrim süresi, öncelik, iş öğeleri için mevkî ve istasyon tipi kısıtlarını dikkate alarak, en küçük "mâliyeti" veren olurlu iş öğesi dizisini bulmaktır. Çevrim süresinin tüm U-tipi hatlar için aynı olduğu kabul edilmiştir. U-tipi hatlar arasındaki çoklu istasyonlar sâyesinde DP algoritması ile her seferinde bir hat dengelenmektedir. Bir U-tipi hattın son istasyonunun, bir sonraki U-tipi hattın ilk istasyonu olması ise, hattın "son"undaki iş öğelerine rastlantısal bir yüksek mâliyet atanması, örneğin en büyük iş öğesi tamamlanma süresi ile sağlanmaktadır. Böylece bu iş öğelerinin atandığı normal veya geçişli istasyonda, bitişik hattın "başlangıç"ındaki iş öğelerinin atanması için yer kalmış olmaktadır.

DP'nin uygulanması için olurlu kümelerin oluşturulmasında Miltenburg, Lawler yaklaşımını yeğlemiştir. Lawler yaklaşımında, olurlu kümeler artan eleman sayısı sırası dahilinde oluşturulmaktadır. Burada U-tipi hatlar sözkonusu olduğundan, Lawler'ın doğrusal hat için geliştirdiği algoritma U-tipi hatta uygulanmış, yâni geriye olurlu kümeler ve ileriye ve geriye olurlu kümelerin bileşimi olan olurlu kümeler oluşturulmuştur. Olurlu kümelerin oluşturulmasından sonra kullanılan DP fonksiyonu ise Held vd.'nin fonksiyonu olarak gösterilmiştir.

$$c(s) = \min_{j \in s} \{c(s-j) + \Delta(c(s-j), t_j)\} \qquad \textbf{(6.1)}$$

Miltenburg, yöntemi yeni denge kurma durumunda bâzı işlerin veya tüm işlerin yerlerinin değiştirilmesi durumları için bir örnek tesis üzerinde ve bilinen test problemlerinden yaratılmış bir farazî çoklu U-tipi hatların bulunduğu tesiste

uygulamıştır. Yöntemin, istenen sayıda, iş öğesi sayısı 22'yi aşmayan ve öncelik diyagramı çok geniş olmayan çoklu U-tipi hat içeren tesislere uygulanabileceği görülmüştür.

6.3.2. BUHDP'de Dal-Sınır Algoritması Uygulamaları Üzerine Literatür Araştırması

BUHDP'deki öncelik ilişkilerindeki farklılık nedeniyle BMHDP'de kullanılan AS4 ve Jackson baskınlık kuralı etkin şekilde kullanılamaz **(Aase ve diğ., 2003)**.

Scholl ve Klein (1999b), bu makalelerinde BMHDP için geliştirdikleri ve SALOME adını verdikleri dal-sınır algoritmasını Basit U-tipi Montaj Hattı Dengeleme Problemi için geliştirmişler ve yeni hâlini ULINO olarak adlandırmışlardır. ULINO'da atamalar iş öğesi numaraları dikkate alınarak yapıldığından, iş öğelerinin uygun şekilde numaralandırılmış olması önemlidir. Bu nedenle problemin başında iş öğelerini yeniden numaralandırmak gereklidir. Bu numaralandırma sâyesinde iş öğeleri ardıllarından ve/veya öncüllerinden daha küçük numaraya sahip olurlar. ULINO'da dallandırma aşamasında derinlik önce arama stratejisinden yararlanılmıştır. Dallandırma aşamasında seçenek istasyon atamaları (düğümler), âtıl sürelerine göre iki gruba ayrılmaktadır: Varolan alt-sınır dâhilinde gerçekleşen atama seçenekleri ve âtıl süresi nedeniyle alt-sınır değeri bir arttırılarak gerçekleşen atama seçenekleri. Öncelikli olarak birinci gruptaki seçenekler üzerinden ilerleniyor. Düğüme her geri dönüşte bir diğer seçenek üzerinden ilerlenmekte. Sınırlama yaklaşımı olarak ULINO'da AS1, AS2, AS3 ve AS6 kullanılmakta. ULINO'da uygulanan baskınlık ve azaltma kuralları ise en fazla yük kuralı, Jackson baskınlık kuralı ve ağaç baskınlık kuralıdır. ULINO, makalede Tip-1 BUHDP, Tip-2 BUHDP ve genel BUHDP için çeşitli veri kümelerine uygulanmıştır. Hesaplamalar sonucunda ULINO'nun kısıtlı süre içerisinde Tip-1 BUHDP ve Tip-2 BUHDP'de çok iyi sonuçlar verdiği ortaya çıkmıştır. Özellikle Tip-1 BUHDP'de ULINO'nun 297 iş öğesine kadar olan problemleri kısa sürede çözdüğü görülmüştür.

Aase, Schniederjans ve Olson (2003), makalelerinde basit U-tipi montaj hattı dengeleme problemi için geliştirilen ve U-OPT adı altında toplanan dal-sınır prosedürlerini ele almıştır. U-OPT1 dallanma stratejisi olarak derinlik önce aramayı

kullanırken, U-OPT2 lazer tipi, U-OPT3 ise en iyi ilk'i kullanmıştır. Bu yordamların ULINO'dan temel farkı, AS5 ve BK6'un geliştirilmiş ve uygulamaya konmuş olmasıdır. Yordamların tanıtılmasından başka, makalede iki soruya yanıt aranmıştır:

1. U-OPT'un basit U-tipi montaj hattı dengeleme problemini çözme performansı üzerinde en büyük etkiye sahip tasarım elemanları nelerdir ? (Burada tasarım elemanı olarak ele alınanlar: AS1, AS2, AS3 ve AS5 alt-sınırları, BK1, BK3, BK4, BK5 ve BK6 baskınlık kuralları, her düğümde yerel üst-sınırın aranması, bir başlangıç çözümünün belirlenmesi ve basit montaj hattı dengeleme probleminin çözümünün başlangıç çözümü olarak kullanılmasıdır. AS4 ve BK2 eleme kuralları, U-tipi montaj hattı dengeleme problemine uyarlanırken güçlerini önemli ölçüde yitirdiklerinden burada ele alınmamışlardır.)
2. Basit U-tipi montaj hattı dengeleme probleminin çözümünde U-OPT diğer bilinen yöntemlere göre ne kadar etkindir ? İlk sorunun yanıtı olarak, en iyi sonuçlar U-OPT1 ile alınmış. Genel olarak bakıldığında BK1, BK3, BK4 ve BK5'in yordam performansı üzerinde önemli bir etkiye sahip olmadığı görülmüştür. U-OPT'un ULINO ile karşılaştırılmasında ise, U-OPT'un belirgin olarak daha iyi performans sergilediği görülmüştür.

7. ETKİN DAL-SINIR ALGORİTMALARININ BİR ÖRNEK ÜZERİNDE UYGULANMASI

Literatürde, Tip-1 BMHDP için etkin oldukları belirtilen FABLE ve EUREKA algoritmaları bir örnek problem üzerinde, Tip-1 BUHDP için etkin oldukları belirtilen ULINO ve U-OPT1 algoritmaları da başka bir problem üzerinde uygulanmıştır.

FABLE ve EUREKA ile çözülen BMHDP'de, iki yöntemin uygulandığı durumda da 6 istasyon ile denge sağlanmıştır. Her iki çözümde de denge kaybı % 20 olarak bulunmuştur.

ULINO ve U-OPT1 ile çözülen BUHDP'de, iki yöntemin uygulandığı durumda da 6 istasyon ile denge sağlanmıştır. Her iki çözümde de denge kaybı % 0 olarak bulunmuştur. İdeal durum elde edilmiştir.

U-OPT1 uygulamasında gereksinim duyulan başlangıç çözümü, problemin BMHDP olarak EUREKA ile çözülmesi ile elde edilmiştir. EUREKA ile hat doğrusal olarak 7 istasyon ile dengelenmiştir. Bu durumda denge kaybı %17 olarak bulunmuştur. Bu da U-tipi hatların doğrusal hatlara göre üstünlüğünün bir göstergesidir.

Algoritmaların örnekler üzerindeki ayrıntılı uygulamaları Ek A, Ek B, Ek C ve Ek D'de bulunmaktadır.

8. UYGULAMA

8.1. Firma Bilgisi

İncelenen dal-sınır algoritmalarının endüstriyel yaşamdan bir problem üzerinde uygulanması hedeflenmiş, bu amaçla Mercedes-Benz Türk firmasındaki 'MS Tipi Koltuk Kumaş Dikim Bölümü'ndeki "deri başlıklı koltuk kumaş dikim hattı" incelenmiştir.

Mercedes-Benz Türk, 1967 yılında İstanbul'da Otomarsan adı altında kurulmuştur. Şirket 1968 yılından beri İstanbul'da otobüs, 1986 yılından itibaren de Aksaray'da kamyon üretmektedir. Şirketin adı 1980 yılında Mercedes-Benz Türk A.Ş. olarak değiştirilmiştir. Şirketin ortakları Daimler Chrysler AG, Overseas Lending Corporation, Koluman Holding A.Ş., Türk Silahlı Kuvvetlerini Güçlendirme Vakfı ve MKE'dir.

Bugün, Mercedes-Benz Türk'ün Aksaray fabrikasında hafif ve ağır sınıf kamyonlar ve çekiciler, İstanbul'daki Hoşdere ve Davutpaşa fabrikalarında ise şehirlerarası ve belediye tiplerinde otobüsler üretilmektedir. Mercedes-Benz Türk A.Ş.'de bugün yaklaşık olarak 2.800 personel istihdam edilmektedir.

Otobüs üretiminde kapasitenin, verimliliğin, kalitenin arttırılması ve ürün yelpazesinin genişletilmesi amacıyla kurulmuş olan Hoşdere fabrikasında, Davutpaşa'dan gelen şehirlerarası ve belediye tipi otobüslerin karoserlerinin boya, montaj ve finiş işlemleri gerçekleştirilmektedir.

Yıllık üretim kapasitesi 3.000 otobüs olan Hoşdere fabrikasının kapasitesinin 4.000 otobüse çıkarılması hedeflenmektedir. Hoşdere ve Davutpaşa Otobüs fabrikalarında bugün yaklaşık olarak 2.200 personel çalışmaktadır.

Montaj ve transfer hatlarının ve sabit çalışma istasyonları kombinasyonunun uygulandığı Hoşdere fabrikasında, aynı anda farklı modellerin üretilmesinde sağlanan esnekliğin yanısıra, üretimde bekleme süreleri de en aza çekilebilmektedir.

Karoserlerin montaj istasyonları arasında taşınması ve başka montaj işleri için hava yastığı teknolojisinin kullanıldığı Hoşdere fabrikasında, bu sâyede, ergonomik açıdan çalışma koşulları iyileştirilmiştir. Hoşdere fabrikasında ambar ve malzeme birimleri, en iyi malzeme akışını sağlamak amacı ile îmalat holleriyle bütünleşik duruma getirilmiştir.

8.2. Uygulama Yapılan Hattın Tanıtımı

Uygulama Mercedes-Benz Türk firmasının Hoşdere Otobüs Fabrikası'ndaki "Koltuk Kumaş Dikim Bölümü"nde, "deri başlıklı koltuk kumaş dikim hattı"nda gerçekleştirilmiştir.

Hatta 07:30-15:30 ve 15:30-23:30 olmak üzere iki vardiya hâlinde çalışılmaktadır. Her vardiyada 2 adet 10 dakikalık çay molası ile 30 dakikalık yemek molası verilmektedir. Bu molalar çıkartıldığında hattın iki vardiyadaki net çalışma süresi 860 dakika olmaktadır.

Varolan durumda günlük araç üretimi 8,2 olduğundan ve her araçta 50 koltuk bulunduğundan, çevrim süresi 126 saniye olarak belirlenmiştir. Ancak hat gerçek durumda basit bir montaj hattı değildir. Belirli dikişlerin belirli makinalarda yapılmasının zorunlu olması ve farklı ayak ayarı gerektiren dikişlerin aynı makinada yapılmasının ek hazırlık süresi gerektirmesi nedeniyle ortaya çıkan sabit donanım kısıtı, hattı basit montaj hattı olmaktan çıkarmıştır.

Uygulama sırasında deri başlıklı koltuk kumaş dikim hattının basit bir montaj hattı olduğu kabul edilmiş, sabit donanım kısıtı gözönüne alınmamıştır.

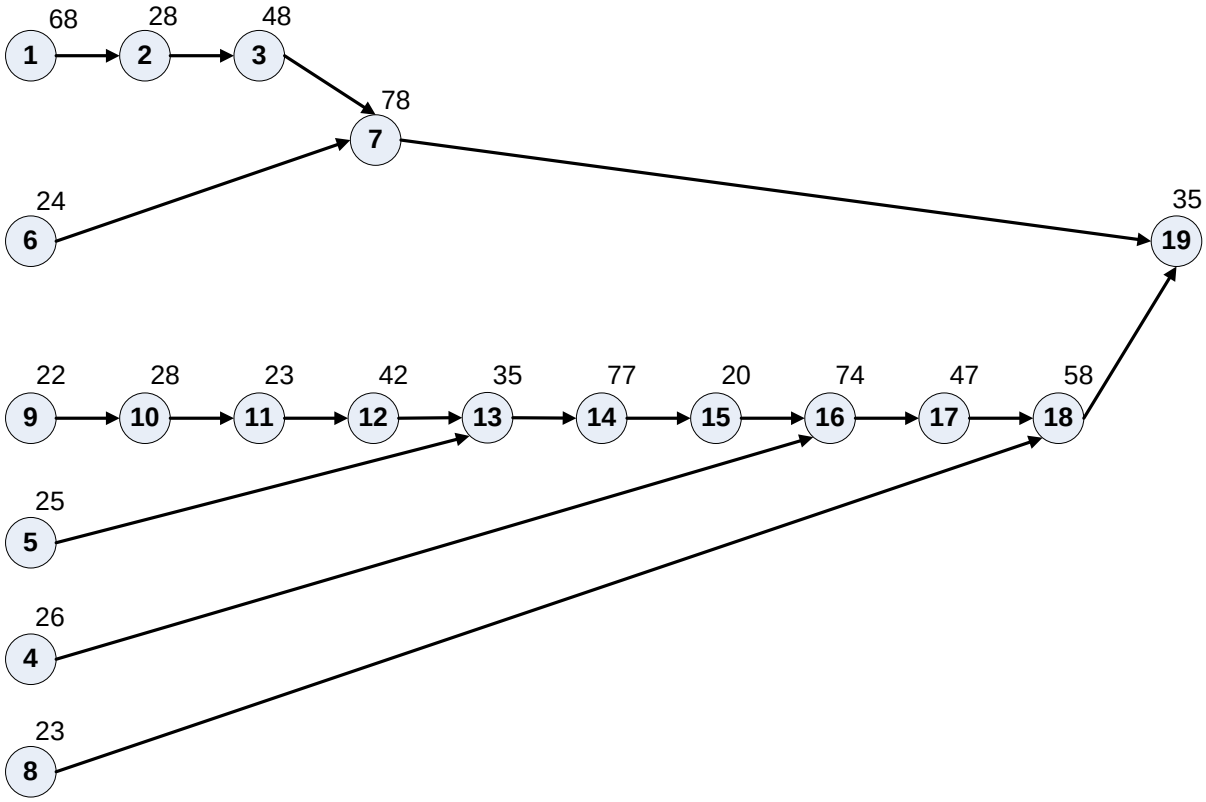
Firmada günlük araç üretiminin 8,6'ya çıkartılması hedeflenmiştir. Buradan hareketle, uygulamada hat için dikkate alınacak çevrim süresi hesaplanırken bu hedef gözönünde tutulmuş, bunun sonucunda da uygulama için çevrim süresi 120 saniye olarak belirlenmiştir.

Deri başlıklı koltuk kumaş dikim hattı 19 iş öğesinden oluşmaktadır. Hatta ait iş öğeleri ve işlem süreleri Tablo 8.1'de gösterilmiştir.

Tablo 8.1: Uygulama Problemi İş Öğeleri

İş Öğesi No	İş Öğesi	Süre (sn.)
1	Minder kumaş overlok	68
2	Minder arka vinleks dikimi	28
3	Minder fitil dikimi	48
4	Sırtlık façası uç dikim	26
5	Çektirme bezinin sırtlığa dikimi	25
6	Minder çektirme bezi dikimi	24
7	Minder faça dikimi	78
8	Gri çektirme bezinin dikimi	23
9	Arka deri başlık çentik atma	22
10	Arka deri başlık dikimi	28
11	Arka deri başlık çentik	23
12	Başlık çevre dikimi	42
13	Başlığın ana kumaşa dikimi	35
14	Sırtlık fitil dikim	77
15	Sırtlık vinleks dikim	20
16	Sırtlık faça dikim	74
17	Arka kapama kumaşının sırtlık ile birleştirme dikimi	47
18	Yan kapama overlok	58
19	Minder sırtlık birleştirme	35

Hatta ait iş öğelerinin teknolojik öncelik diyagramı Şekil 8.1'de, teknolojik öncelik matrisi ise Tablo 8.2'de, gösterilmiştir. Yapılan hesaplama sonucunda hattın esneklik oranının 0,53 olduğu bulunmuştur. Bu da, oldukça esnek öncelik ilişkisi yapısına sahip bir problem ile karşı karşıya olunduğunu göstermektedir.



Şekil 8.1: Uygulama Problemi Öncelik Diyagramı

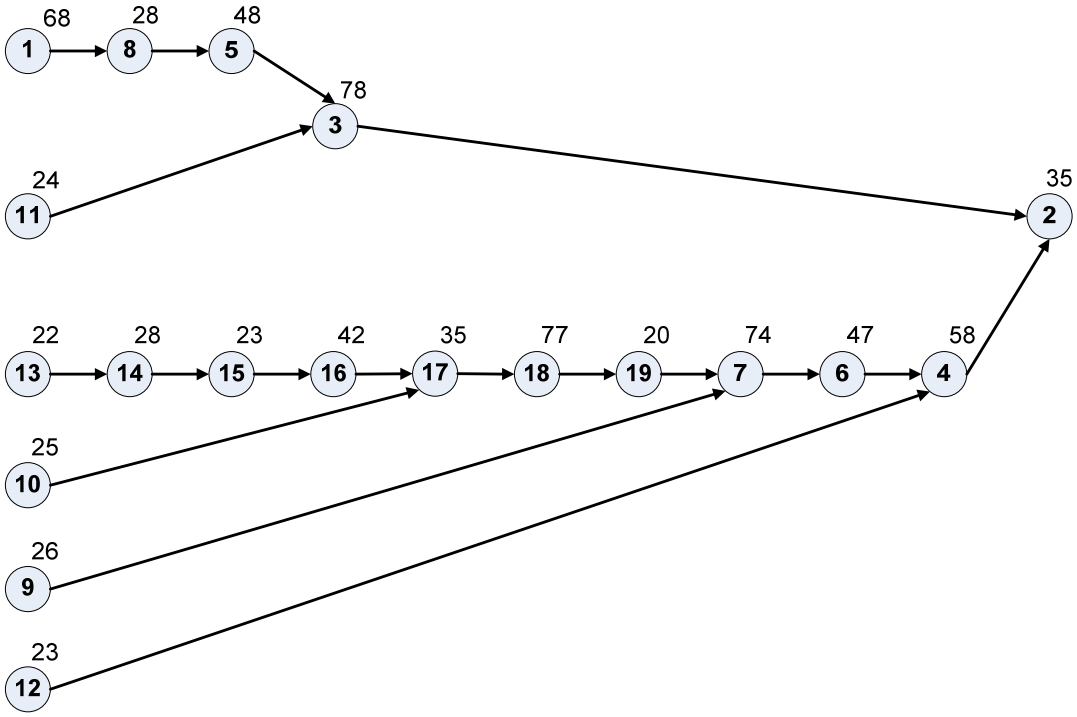
Tablo 8.2: Uygulama Problemi Öncelik Matrisi

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
1	-	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2		-	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
3			-	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
4				-	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
5					-	0	0	0	0	0	0	0	1	1	1	1	1	1	1
6						-	1	0	0	0	0	0	0	0	0	0	0	0	1
7							-	0	0	0	0	0	0	0	0	0	0	0	1
8								-	0	0	0	0	0	0	0	0	0	1	1
9									-	1	1	1	1	1	1	1	1	1	1
10										-	1	1	1	1	1	1	1	1	1
11											-	1	1	1	1	1	1	1	1
12												-	1	1	1	1	1	1	1
13													-	1	1	1	1	1	1
14														-	1	1	1	1	1
15															-	1	1	1	1
16																-	1	1	1
17																	-	1	1
18																		-	1
19																			-

8.3. Problemin EUREKA, FABLE, ULINO ve U-OPT1 ile Çözümü

Uygulama için seçilen problem basit I-tipi montaj hattı olarak EUREKA ve FABLE algoritmaları ile, daha sonra da basit U-tipi montaj hattı olarak ULINO ve U-OPT1 algoritmaları ile çözülmüştür. EUREKA ve U-OPT1 algoritmalarının uygulanmasında MATLAB7.3 kullanılmıştır. EUREKA ve ULINO algoritmalarının uygulanması sırasında, istasyonlara olurlu iş öğesi atamaları, **Hoffmann**'ın **(1963)** geliştirdiği öncelik matrisi yaklaşımından yararlanılarak oluşturulmuştur. UOPT1 algoritmasında her düzeydeki istasyon atama seçeneklerinin oluşturulmasında da, yine Hoffmann'ın yaklaşımı temel alınmıştır.

Hoffmann'ın Olurlu İstasyon Atama Yaklaşımı: Hoffmann (1963) hangi aşamada hangi iş öğelerinin atanabilir olduğunu anlamak için öncelik matrisinden yararlanmıştır. Hoffmann'ın yaklaşımında kullandığı öncelik matrisinde salt doğrudan ardıllık ilişkisi 1 ile gösterilir. Dolaylı öncelik ilişkilerinde hücrelere 0 yazılır. Öncelik matrisinin her sütunu toplanarak matrisin altında yeni bir satır yaratılır. Bu satırda 0 değerine sahip iş öğeleri, o aşamada atanmaya hazır iş öğeleridir, çünkü 0 değeri bu iş öğelerinin o aşamada atanmamış herhangi bir öncülünün olmadığını göstermektedir. Bulunulan aşamada istasyona, boş kalan istasyon süresine uygun süreli her iş öğesinin atanmasından sonra, atanan iş öğesine ait satır ve sütun öncelik matrisinden çıkartılır, böylece öncelik matrisi güncellenir. Güncellenmiş öncelik matrisi için her seferinde yeni bir sütun toplamı satırı oluşturulur ve atanmaya hazır yeni iş öğeleri belirlenmiş olur. Bu yaklaşımı U-tipi hatların dengelenmesinde kullanabilmek için öncelik matrisinin yanında ardıllık matrisinin de oluşturulması gerektiği görülmüştür. Bu amaçla öncelik matrisini oluştururken kullanılan mantık ışığında bir ardıllık matrisi oluşturulmuştur. Öncelik matrisinin sütunlarının toplanması ile oluşturulan ek satır, ardıllık matrisi için de oluşturulur. Bu satırdaki 0 değerleri, U-tipi hat dengeleme problemi çözülürken ardılı atandığı için, o aşamada atanmaya hazır iş öğelerinin hangileri olduğunu gösterir. Uygulama probleminin ULINO ile çözülmesi sırasında oluşturulan, yeniden numaralandırılmış iş öğelerine ilişkin öncelik diyagramı Şekil 8.2'de, doğrudan öncelik ve doğrudan ardıllık matrisleri ise Tablo 8.3 ve Tablo 8.4'te görülmektedir.



Şekil 8.2: ULINO İçin Öncelik Diyagramı

Uygulama probleminin, 120 saniyelik çevrim süresi için, basit I-tipi montaj hattı olarak **EUREKA** ile dengelenmesi için öncelikle Hoffmann'ın öncelik matrisi yaklaşımından yararlanılmıştır. Uygun iş öğeleri, EUREKA algoritması gereğince istasyon âtıl süreleri gözönünde tutularak istasyonlara atandığında, toplam 19 iş öğesi, 8 istasyon ile dengelenmiştir. Hat, **FABLE** ile dengelendiğinde de EUREKA ile aynı sonuç bulunmuştur. Her iki dengeleme sonucunda iş öğelerinin istasyonlara atanmaları şu şekildedir:

S_1 = {1-2-6}

S_2 = {3-4-5}

S_3 = {7-8}

S_4 = {9-10-11-12}

S_5 = {13-14}

S_6 = {15-16}

S_7 = {17-18}

S_8 = {19}

Tablo 8.3: ULINO için iş öğeleri doğrudan öncüllük matrisi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
H0	0	2	2	2	1	1	2	1	0	0	0	0	0	1	1	1	2	1	1

Tablo 8.4: ULINO için iş öğeleri doğrudan ardıllık matrisi

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

Uygulama problemi aynı çevrim süresi dahilinde, bu sefer basit U-tipi montaj hattı olarak dengelenmiştir. Problem, U-tipi montaj hattı olarak, öncelikle **U-OPT1** algoritması ile çözülmüştür. U-OPT1 ile çözerken Hoffmann'ın öncelik matrisi yaklaşımından ve problemin global üst sınırı olarak da EUREKA algoritması ile bulduğumuz 8 istasyonlu dengeden yararlanılmıştır. U-OPT1 ile amaçlanan, problemin kuramsal en az istasyon sayısı olan 7 istasyon ile dengelenmiş bir çözümünü bulmaktır. U-OPT1 uygulaması sonucu, problem 7 istasyon ile şu şekilde dengelenmiştir:

S_1 = {1-2-6}

S_2 = {3-4-5}

S_3 = {7-19}

S_4 = {9-10-11-12}

S_5 = {13-14}

S_6 = {18-17}

S_7 = {8-15-16}

Uygulama problemi, 120 saniyelik çevrim süresi için, basit U-tipi montaj hattı dengeleme problemi olarak U-OPT1 algoritması ile dengelendikten sonra, bu kez de yine basit U-tipi montaj hatları için en iyi çözümü veren **ULINO** algoritması ile dengelenmiştir. ULINO algoritmasında başlangıçta ve her aşamada atanmamış iş öğelerinin belirli kurallar dahilinde yeniden numaralandırılması gerekmektedir. ULINO ile hat, U-OPT1'de olduğu gibi, uygulama problemi için kuramsal en az istasyon sayısı olan 7 istasyon ile dengelenmiştir. ULINO için yeniden numaralandırılmış iş öğeleri, atandıkları istasyonlarda başlangıçtaki özgün numaraları ile aşağıda gösterilmiştir:

S_1 = {1-2-6}

S_2 = {3-4-5}

S_3 = {7-19}

S_4 = {9-10-11-12}

S_5 = {13-14}

S_6 = {8-15-16}

S_7 = {17-18}

8.4. Dengeleme Çözümlerinin Karşılaştırılması

Görüldüğü üzere, uygulama problemini oluşturan hat basit I-tipi bir montaj hattı olarak düşünüldüğünde 8 istasyonla, basit U-tipi montaj hattı olarak düşünüldüğünde ise gerekli en az istasyon sayısı olan 7 istasyon ile dengelenmiştir. U-tipi montaj hattı sâyesinde, işin istenen çevrim süresi dahilinde yapılması için gereksinim duyulan istasyon sayısı 1 azalmış ve böylece U-tipi hattın I-tipi hatta olan en görünür üstünlüğü gözlemlenmiştir. Hattın U-OPT1 ve ULINO ile dengelenmesi sonucunda elde edilen 7 istasyonlu çözümlerin ilk 5 istasyonu bütünüyle aynıdır, son iki istasyona atanan iş öğelerinin istasyon içi yapılış sıraları aynı ama istasyon sıraları farklıdır.

Bulunan her dört çözüm de, iş öğelerinin varolan istasyonlara ne ölçüde dengeli dağıldığının görülmesi amacıyla, denge kaybı açısından karşılaştırıldığında;

EUREKA ve FABLE için denge kaybı %18,6

ve

U-OPT1 ve ULINO için denge kaybı %7

olarak bulunmuştur.

Her iki denge kaybı da ideal değildir ama U-tipi hat olarak bulunan çözümün daha küçük bir denge kaybı sağladığı açıktır.

İşlerin istasyonlar arasında orantılı dağılıp dağılmadığını görmek için dört çözümü düzgünlük indeksi açısından karşılaştırdığımızda;

EUREKA ve FABLE için düzgünlük indeksi %10

ve

U-OPT1 ve ULINO için düzgünlük indeksi %3,4

olarak bulunmuştur.

Burada da U-tipi hat dengeleme yapılması durumunda istasyonların çok daha düzgün yükleneceği görülmektedir.

Uygulama problemi için, hattın kuramsal en az istasyonla dengelenmesi durumunu gösteren kuramsal hat etkinliği %93'tür. Oysa hattın basit I-tipi montaj hattı olarak dengelenmesi durumunda hat etkinliği %81 olarak bulunmaktadır. Hat U-OPT1 veya ULINO kullanılarak basit U-tipi montaj hattı olarak dengelendiğinde ise hat etkinliği %93 olarak bulunmuştur. Bu da kuramsal etkinliğe eşittir.

9. SONUÇLAR ve TARTIŞMA

Bu çalışmada öncelikle, basit montaj hattı dengeleme problemi için en iyi çözüm veren yöntemler arasında etkinlik açısından öne çıkan dinamik programlama ve dal-sınır algoritması incelenmiştir. Bu yöntemler üzerine literatürde yapılan inceleme sonucunda, dal-sınır algoritmasının hesaplama süresi ve gereksinim duyulan bellek boyutu açısından dinamik programlamaya kıyasla daha etkin çalıştığı görülmüştür. Dal-sınır algoritmalarının araştırılması sonucunda FABLE ve EUREKA algoritmalarının en iyi çözüme en kısa sürede ulaşmak açısından etkin algoritmalar olduğu belirlenmiştir.

Dinamik programlama ve dal-sınır algoritmalarının U-tipi hatlara uygulandığı çalışmaların sonuçlarının da benzer olduğu görülmüştür. Dal-sınır algoritmaları arasında U-tipi montaj hatlarının dengelenmesinde öne çıkan yaklaşımlar olarak ise ULINO ve U-OPT1 algoritmaları belirlenmiştir.

Endüstriyel yaşamdan alınan 19 iş öğeli bir montaj hattı dengeleme probleminin I-tipi montaj hattı olarak çözümü için EUREKA ve FABLE, U-tipi montaj hattı olarak çözümü için ise ULINO ve U-OPT1 algoritmaları kullanılmıştır.

Problemin U-tipi montaj hattı olarak dengelendiği durumlarda, beklendiği gibi, I-tipi hatta kıyasla daha az sayıda istasyonla denge sağlanmıştır. U-tipi montaj hattı olarak dengelemede bulunan istasyon sayısı, kuramsal en az istasyon sayısına eşittir. Denge kaybı, düzgünlük indeksi ve hat etkinliği açısından da, problemin U-tipi montaj hattı olarak dengelenmesi durumunda daha iyi sonuçlar elde edilmiştir.

ULINO ve U-OPT1 algoritmalarının uygulama problemine uygulanması sonucunda elde edilen istasyonlardan ilk beşi bütünüyle aynı iş öğelerinden oluşmaktadır ve iş öğelerinin sıraları aynıdır. Son iki istasyon ise her iki durumda da aynı iş öğeleriyle yüklenmiştir ama istasyon sıraları farklıdır.

Yapılan çalışma sonucunda dal-sınır algoritmaları ile, montaj hattı dengeleme problemi için en iyi çözümün bulunduğu ve buna ek olarak, problemin U-tipi

montaj hattı olarak dengelenmesinin, I-tipi montaj hattı olarak dengelenmesine kıyasla çok daha üstün olduğu görülmüştür.

KAYNAKLAR

Aase, G.R., Schniederjans, M.J., Olson, J.R., 2003. U-OPT: An Analysis of Exact U-shaped Line Balancing Procedures, *International Journal of Production Research*, **41 (17)**, 4185-4210.

Acar, N., 1989. Üretim Plânlaması Yöntem ve Uygulamaları, Milli Prodüktivite Merkezi Yayınları, Ankara.

Askin, R.G., Standridge, C.R., 1993. Modelling and Analysis of Manufacturing Systems, John Wiley and Sons Inc., Singapore.

Bard, J. F., 1989. Assembly Line Balancing with Parallel Workstations and Dead Time, *International Journal of Production Research*, **27**, 1005-1018.

Baskak, M., 2005. Üretim Hatlarında Modelleme Ders Notları, İstanbul.

Baybars, İ., 1986. A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem, *Management Science,* **32 (8)**, 909-932.

Becker, C., Scholl, A., 2006. A Survey on Problems and Methods in Generalized Assembly Line Balancing, *European Journal of Operational Research*, **168**, 694-715.

Bedworth, D.D., Bailey, J.E., 1987. Integrated Production Control Systems, John Wiley and Sons Inc., Singapore.

Berger I., Bourjolly J.M., Laporte G., 1992. Branch-and-Bound Algorithms for the Multi-Product Assembly Line Balancing Problem, *European Journal of Operational Research*, **58**, 215-222.

Easton, F. F., 1990. A Dynamic Program with Fathoming and Dynamic Upper Bounds for the Assembly Line Balancing Problem, *Computers and Oper. Res.*, **17**, 163-175.

Erel, E., Sarin, S. C., 1998. A Survey of the Assembly Line Balancing Procedures, *Production Planning and Control*, **9 (5)**, 414-434.

Everett, E.A., Ronald J.E., 1992. Production and Operations Management, Prentice Hall Inc., New Jersey.

Gutjahr, A. L. ve Nemhauser, G. L., 1964. An Algorithm for the Line Balancing Problem, *Management Science*, **11**, 308-315.

Hackman, S.T., Magazine, M. J., Wee, T.S., 1989. Fast, Effective Algorithms For Simple Assembly Line Balancing Problems, *Operations Research*, **37 (6)**, 916-924.

Held, M., Karp, R. M. ve Shareshian, R., 1963. Assembly Line Balancing – Dynamic Programming with Precedence Constraints, *Operations Research*, **11**, 442-459.

Henig, M. I., 1986. Extensions of the Dynamic Programming Method in the Deterministic and Stochastic Assembly Line Balancing Problems, *Computers and Oper. Res.*, **13**, 443-449.

Hoffmann, T.R., 1963. Assembly Line Balancing with a Precedence Matrix, *Management Science,* **9 (4)**, 551-562.

Hoffmann, T.R., 1992. EUREKA: A Hybrid System for Assembly Line Balancing, *Management Science*, **38 (1)**, 39-47.

Jackson, J.R., 1956. A Computing Procedure for a Line Balancing Problem, *Management Science,* **2 (3)**, 261-271.

Johnson, R.V., 1981. Assembly Line Balancing Algorithms: Computation Comparisons, *International Journal of Production Research*, **19 (3)**, 277-287.

Johnson, R.V., 1983. A Branch and Bound Algorithm for Assembly Line Balancing Problems with Formulation Irregularities, *Management Science,* **29 (11)**, 1309-1324.

Johnson, R.V., 1988. Optimally Balancing Large Assembly Lines With 'Fable', *Management Science*, **34 (2)**, 240-253.

Kao, E. P. C. ve Queyranne, M., 1982. On Dynamic Programming Methods for Assembly Line Balancing, *Operations Research*, **30**, 375-390.

Kobu, B., 1982, Üretim Yönetimi, İstanbul Üniversitesi Yayınları, İstanbul.

Miltenburg, J., 1998. Balancing U-lines in a Multiple U-line Facility, *European Journal of Operational Research*, **109**, 1-23.

Miltenburg, J., 2001. U-shaped Production Lines: A Review of Theory and Practice, *International Journal of Production Economics*, **70**, 201-214.

Miltenburg, J. ve Wijngaard J., 1994. The U-line Line Balancing Problem, *Management Science*, **40**, 1378-1388.

Saltzman M.J., Baybars İ., 1987. A Two-process Implicit Enumeration Algorithm for the Simple Assembly Line Balancing Problem, *European Journal of Operational Research*, **32**, 118-129.

Scholl, A., 1999. Balancing and Sequencing of Assembly Lines, Physica-Verlag, Heidelberg.

Scholl, A., Becker, C., 2006. State-of-the art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing, *European Journal of Operational Research*, **168**, 666-693.

Scholl, A., Klein, R., 1999a. Balancing Assembly Lines Effectively – A Computational Comparison, *European Journal of Operational Research*, **114**, 50-58.

Scholl, A., Klein, R., 1999b. ULINO: Optimally Balancing U-shaped JIT Lines, *International Journal of Production Research*, **37 (4)**, 721-736.

Schrage, L. ve Baker, K. R., 1978. Dynamic Programming Solution of Sequencing Problems with Precedence Constraints, *Operations Research*, **26**, 444-449.

Taha, H.A., 2003. Yöneylem Araştırması, Literatür Yayıncılık, İstanbul.

Talbot, F.B., Patterson, J.H., 1984. An Integer Programming Algorithm with Network Cuts for Solving the Assembly Line Balancing Problem, *Management Science,* **30 (1)**, 85-99.

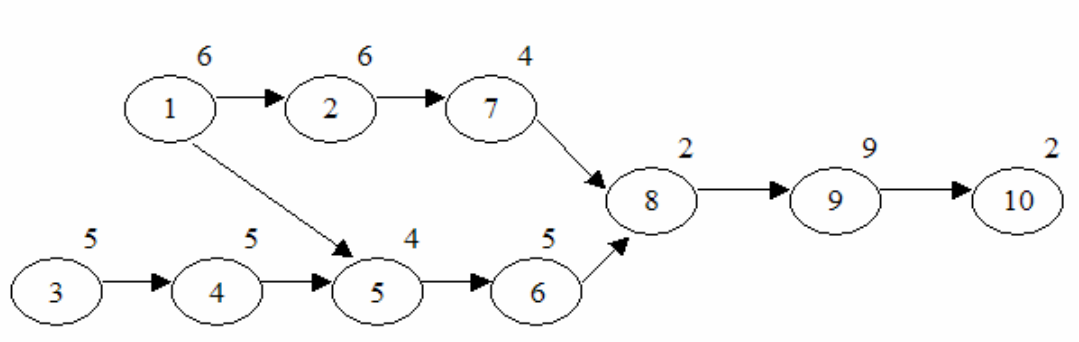
Talbot, F.B., Patterson, J.H., Gehrlein, W.V., 1986. A Comparative Evaluation of Heuristic Line Balancing Techniques, *Management Science*, **32 (4)**, 430-454.

Tanyaş M., Baskak, M., 2006. Üretim Plânlama ve Kontrol, İrfan Yayımcılık, İstanbul.

Van Assche F., Herroelen W.S., 1978. An Optimal Procedure for the Single-Model Deterministic Assembly Line Balancing Problem, *European Journal of Operations Research*, **3**, 142-149.

EKLER

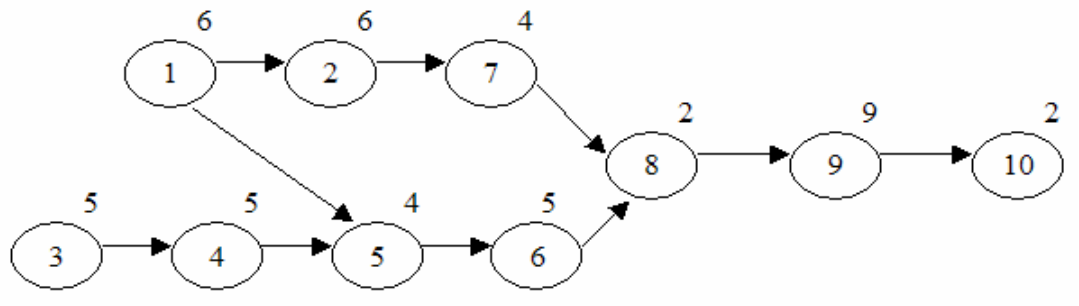
EK A: FABLE Uygulaması



Şekil A.1: Tip-1 BMHDP Öncelik Diyagramı **(Scholl, 1999)**

Şekil A.1.'deki öncelik diyagramı için C = 10'dur. Olanak varsa, iş öğesi süresi arttırma kuralı uygulanır. Bu örnekte başka hiç bir iş öğesi ile birlikte bir istasyona atanamayacak salt tek bir iş öğesi var: 9 numaralı iş öğesi. 9 numaralı iş öğesinin işlem süresi 10, yâni çevrim süresi olarak kabul edilir.

FABLE'da kümeler iş öğesi numaraları dikkate alınarak oluşturulacağından, iş öğeleri yeniden numaralandırılır. Numaralandırılma sonucu Şekil A.2.'deki öncelik diyagramı elde edilir.



Şekil A.2: Yeniden Numaralandırılma ile Elde Edilen Öncelik Diyagramı

Jackson'ın baskınlık kuralına uyan iş öğesi çiftlerinin varolup olmadığı, belirlenir.

(a, b) ifâdesi 'a, b'ye göre baskındır' demektir.

Bu örnekte böyle iki çift var; (1, 4) ve (6, 7)

AS4 (tek makina çizelgeleme sınırı) kullanılarak problem için alt-sınır hesaplanır.

$p_i = t_i / C$

İş öğelerinin p_i değerleri Tablo A.1'de görülebilir.

Tablo A.1: İş Öğelerinin p_i Değerleri

	10	9	8	7	6	5	4	3	2	1	0
p_i	0,2	1	0,2	0,4	0,5	0,4	0,5	0,5	0,6	0,6	0

$n_{i1} = AS1(F_i^*)$ F_i^* = i işinin doğrudan ve geçişli ardıllarının kümesi

burada AS1 = $\sum_{i=1}^{n} p_i$ olarak hesaplanır.

İş öğelerinin n_{i1} değerleri Tablo A.2'de gösterilmiştir.

Tablo A.2: İş Öğelerinin n_{i1} Değerleri

	10	9	8	7	6	5	4	3	2	1	0
n_{i1}	0	0,2	1,2	1,4	1,4	1,9	2,3	2,8	1,8	3,3	4,9

$$n_{i2} = \begin{cases} AS2(F_i^*) & eger\ p_i \geq \frac{1}{2}\ veya\ AS2(F_i^*) \notin N \\ AS2(F_i^*) - \frac{1}{2} & diger\ durumda \end{cases}$$

burada AS2 = $\left|J\left(\frac{1}{2},1\right]\right| + \frac{1}{2}\left|J\left[\frac{1}{2},\frac{1}{2}\right]\right|$ olarak hesaplanır.

|A|, A kümesindeki eleman sayısını belirtir.

İş öğelerinin n_{i2} değerleri Tablo A.3'de gösterilmiştir.

Tablo A.3: İş Öğelerinin n_{i2} Değerleri

	10	9	8	7	6	5	4	3	2	1	0
n_{i2}	0	0	0,5	0,5	1	1,5	1,5	2	1	2,5	4,5

$$n_{i3} = \begin{cases} AS3(F_i^*) & eger\ \ p_i \geq \dfrac{2}{3} \\ AS3(F_i^*) - \dfrac{1}{3} & diger\ durumda \end{cases}$$

burada $AS3 = \left| J\left(\frac{2}{3},1\right]\right| + \frac{2}{3}\left| J\left[\frac{2}{3},\frac{2}{3}\right]\right| + \frac{1}{2}\left| J\left(\frac{1}{3},\frac{2}{3}\right)\right| + \frac{1}{3}\left| J\left[\frac{1}{3},\frac{1}{3}\right]\right|$ olarak hesaplanır.

İş öğelerinin n_{i3} değerleri Tablo A.4'de gösterilmiştir.

Tablo A.4: İş Öğelerinin n_{i3} Değerleri

	10	9	8	7	6	5	4	3	2	1	0
n_{i3}	0	0	0,67	0,67	0,67	1,17	1,67	2,17	1,17	2,67	4,5

$$n_{i4} = \text{Enb}\ \left\{p_{h_1} + n_{h_1}, p_{h_1} + p_{h_2} + n_{h_2}, ..., p_{h_1} + p_{h_2} + ... + p_{h_r} + n_{h_r}\right\}$$

$n_i = \text{Enb}\ \{n_{i1}, n_{i2}, n_{i3}, n_{i4}\}$

Eğer $p_i + n_i > [n_i]^+$ ise, n_i en yakın büyük tamsayıya yuvarlanır ve bu hâli kullanılır.

Enb değerinin hesaplanması sırasında iş öğelerinin yazılış sırası, iş öğelerinin yuvarlanmış değerlerine göre azalan şekilde sıralanmasıyla elde edilir. İş öğelerinin n_{i4}, n_i ve yuvarlanmış değerleri Tablo A.5'de gösterilmiştir.

Tablo A.5. İş Öğelerinin n_{i4}, n_i ve Yuvarlanmış Değerleri

	10	9	8	7	6	5	4	3	2	1	0
n_{i4}	0	0,2	2	2,2	2,2	2,7	3,4	3,9	2,6	4,1	5,7
n_i	0	0,2	2	2,2	2,2	2,7	3,4	3,9	2,6	4,1	5,7
yuvarlanmış	0	1	2	2,2	2,2	3	3,4	4	3	4,1	6

$n_{104} = 0$

$n_{94} = \text{Enb}\ \{p_{10} + n_{10}\} = \text{Enb}\ \{0,2 + 0\} = 0,2$

$n_{84} = \text{Enb}\ \{p_9 + n_9, p_9 + p_{10} + n_{10}\} = \text{Enb}\ \{1 + 1, 1 + 0,2 + 0\} = 2$

$n_{74} = \text{Enb}\ \{0,2 + 2; 0,2 + 1 + 1; 0,2 + 1 + 0,2 + 0\} = 2,2$

$n_{64} = \text{Enb}\ \{2,2; 2,1; 1,3\} = 2,2$

$n_{54} = \text{Enb}\ \{2,7; 2,7; 2,7; 1,9\} = 2,7$

$n_{44} = \text{Enb}\ \{3,4; 3,1; 3,1; 3,1; 2,3\} = 3,4$

$n_{34} = \text{Enb}\ \{3,9; 3,9; 3,6; 3,6; 3,6; 2,8\} = 3,9$

$n_{24} = \text{Enb}\ \{2,6; 2,6; 2,6; 1,8\} = 2,6$

$n_{14} = \text{Enb}\ \{3,6; 4; 3,7; 3,7; 4,1; 4,1; 3,3\} = 4,1$

$n_{04} = \text{Enb}\ \{4,7; 5,1; 5; 5,2; 5,6; 5,3; 5,7; 5,7;5,7; 4,9\} = 5,7$

$AS4 = n_0 = 6$ olur

Etiketleme baskınlık kuralı için iş öğelerinin etiketleri belirlenir.

i = 1, 2, …, n için

$L(i) = \sum_{h \in p_i'} L(h) + 1 \qquad p_i' = \{h \mid h < i \ ve \ h \notin P_i^*\}$

P_i^* = i işinin doğrudan ve geçişli öncüllerinin kümesi

$L(1) = 0+1 = 1$

$L(2) = 0+1 = 1$

$L(3) = L(1) + L(2) + 1 = 3$

$L(4) = L(1) + L(2) + 1 = 3$

$L(5) = L(2) + 1 = 2$

$L(6) = L(2) + 1 = 2$

$L(7) = L(3) + L(4) + L(5) + L(6) + 1 = 11$

$L(8) = 0 + 1 = 1$

$L(9) = 0 + 1 = 1$

$L(10) = 0 + 1 = 1$

Ağaç oluşturulur.

Adım 1 Varolan istasyon numarası k = 1

Kullanılmamış k. istasyon süresi = C = 10

Varolan atanmış iş öğesi sayısı p = 0

Yeni seçilmiş iş öğesi i = 1

Adım 2 Yeni seçilmiş iş öğesi 1. istasyona atanır $S_1 = \{1\}$

Kullanılmamış istasyon süresi = $C - t_1 = 10 - 6 = 4$

Varolan atanmış iş öğesi sayısı p = 1 {1}

Geri dönüş iş öğesi = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 (Yeni seçilmiş iş öğesi > 0) ama böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 Yeni bir istasyon başlat (k = 2)

Kullanılmamış istasyon süresi = (C = 10)

Adım 5 Yeni seçilmiş iş öğesi i = 2

Adım 2 Yeni seçilmiş iş öğesi 2. istasyona atanır $S_2 = \{2\}$

Kullanılmamış istasyon süresi = $C - t_2 = 10 - 6 = 4$

Varolan atanmış iş öğesi sayısı p = 2 {1, 2}

Geri dönüş iş öğesi = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 Yeni seçilmiş iş öğesi (ysiö) i = 7

Adım 2 Yeni seçilmiş iş öğesi 2. istasyona atanır $S_2 = \{2, 7\}$

Kullanılmamış istasyon süresi (kis) = 0

Varolan atanmış iş öğesi sayısı p = 3 {1, 2, 7}

Geri dönüş iş öğesi (gdiö) = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö > 0 fakat böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 k = 3

kis = C = 10

Adım 5 ysiö i = 3

Adım 2 S_3 = {3}

kis = 10 – 5 = 5

p = 4 {1, 2, 7, 3}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö i = 4

Adım 2 S_3 = {3, 4}

kis = 0

p = 5 {1, 2, 7, 3, 4}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 (ysiö > 0) ama böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 k = 4

kis = 10

Adım 5 ysiö i = 5

Adım 2 S_4 = {5}

kis = 6

p = 6 {1, 2, 7, 3, 4, 5}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö i = 6

Adım 2 $S_4 = \{5, 6\}$

kis = 1

p = 7 {1, 2, 7, 3, 4, 5, 6}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö > 0 ama böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 k = 5

kis = 10

Adım 5 ysiö i = 8

Adım 2 $S_5 = \{8\}$

kis = 8

p = 8 {1, 2, 7, 3, 4, 5, 6, 8}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 (ysiö > 0) ama böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 k = 6

kis = 10

Adım 5 ysiö i = 9

Adım 2 $S_6 = \{9\}$

kis = 0

p = 9 {1, 2, 7, 3, 4, 5, 6, 8, 9}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 (ysiö > 0) ama böyle bir iş öğesi yok, Adım 4'e geç

Adım 4 k = 7

kis = 10

Adım 5 ysiö i = 10

Adım 2 $S_7 = \{10\}$

kis = 8

p = 10 {1, 2, 7, 3, 4, 5, 6, 8, 9, 10}

gdiö = 0

p = n olduğundan Adım 6'ya geç

Adım 6 Varolan geçerli çözümde k = 7 istasyonla denge sağlanmış.

{1; 2, 7; 3, 4; 5, 6; 8; 9; 10} olurlu ama optimal olmayan bir sonuç, çünkü AS = 6.

Adım 7 Geri dönüş

gdiö = en son atamış iş öğesi i =10

$S_7 - 10 \rightarrow S_7 = \Phi$

S_7'nin kis = 10

p = 9 {1, 2, 7, 3, 4, 5, 6, 8, 9}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 İş öğesi varolan değil ve p > 0 , Adım 7'ye geç

Adım 7 S_7'nin kis = C olduğundan, k – 1 ➔ k = 6

gdiö = 9

$S_6 - 9 \rightarrow S_6 = \Phi$

S_6'nın kis = 10

p = 8 {1, 2, 7, 3, 4, 5, 6, 8}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 İş öğesi varolan değil ve p > 0, Adım 7'ye geç

Adım 7 S_6'nın kis = C olduğundan, k -1 ➔ k = 5

gdiö = 8

$S_5 - 8 \rightarrow S_5 = \Phi$

S_5'in kis = 10

p = 7 {1, 2, 7, 3, 4, 5, 6}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 İş öğesi varolan değil ve p > 0, Adım 7'ye geç

Adım 7 S_5'in kis = C olduğundan, k – 1 ➔ k = 4

gdiö = 6

$S_4 - 6 \rightarrow S_4 = \{5\}$

S_4'ün kis = 6

p = 6 {1, 2, 7, 3, 4, 5}

ysiö = 0

kis < C olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve gdiö > 0, Adım 7'ye geç

Adım 7 gdiö = 5

$(S_4 - 6) - 5 \rightarrow S_4 = \Phi$

S_4'ün kis = 10

p = 5 {1, 2, 7, 3, 4}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 İş öğesi varolan değil ve p > 0, Adım 7'ye geç

Adım 7 S_4'ün kis = C olduğundan, k – 1 ➔ k = 3

gdiö = 4

$S_3 - 4 \rightarrow S_3 = \{3\}$

S_3'ün kis = 5

p = 4 {1, 2, 7, 3}

ysiö = 0

kis < C olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve gdiö > 0, Adım 7'ye geç

Adım 7 gdiö = 3

$(S_3 - 4) - 3 \rightarrow S_3 = \Phi$

S_3'ün kis = 10

p = 3 {1, 2, 7}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 İş öğesi varolan değil ve p > 0, Adım 7'ye geç

Adım 7 S_3'ün kis = C olduğundan, k – 1 ➔ k = 2

gdiö = 7

$S_2 - 7 \rightarrow S_2 = \{2\}$

S_2'nin kis = 4

p = 2 {1, 2}

ysiö = 0

kis < C olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve gdiö > 0, Adım 7'ye geç

Adım 7 gdiö = 2

$(S_2 - 7) - 2 \rightarrow S_2 = \Phi$

S_2'nin kis = 10

p = 1 {1}

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 ysiö = 3

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN DEĞİL

AS4 ile olurlu çözüm için istasyon sayısı 7 bulundu. Bu çözüm varolan geçerli çözümden daha iyi değil. Bu nedenle (i = 3) yeni seçilmiş iş öğesi olarak alınmaz.

İş öğesi varolan değil ve p > 0, Adım 7'ye geç

Adım 7 S_2'nin kis =C olduğundan, k -1 ➔ k = 1

gdiö = 1

$S_1 - 1$ ➔ $S_1 = \Phi$

S_1'in kis = 10

p = 0

ysiö = 0

kis = C olduğundan Adım 5'e geç

Adım 5 ysiö = 3

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 3) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_1 = \{3\}$

kis = 5

p = 1 {3}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö = 4

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 4) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_1 = \{3, 4\}$

kis = 0

p = 2 {3, 4}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve ysiö > 0, Adım 4'e geç

Öncesinde eleme kuralları uygulanır:

- En fazla yük kuralı UYGUN
- Jackson'ın baskınlık kuralı UYGUN
- İlk istasyon baskınlık kuralı UYGUN
- Etiketleme baskınlık kuralı UYGUN
- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

Adım 4 k = 2

kis = 10

Adım 5 ysiö = 1

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 1) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_2 = \{1\}$

kis = 4

p = 3 {3, 4, 1}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 ysiö = 5

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 5) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_2 = \{1, 5\}$

kis = 0

p = 4 {3, 4, 1, 5}

gdiö = 0

p ≠ n olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve ysiö > 0, Adım 4'e geç

Öncesinde eleme kuralları uygulanır:

- En fazla yük kuralı UYGUN
- Jackson'ın baskınlık kuralı UYGUN
- İlk istasyon baskınlık kuralı UYGUN
- Etiketleme baskınlık kuralı UYGUN
- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

Adım 4 k = 3

kis = 10

Adım 5 ysiö = 2

Eleme kuralları uygulanır:
- Etiketleme baskınlık kuralı UYGUN
- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 2) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_3 = \{2\}$

kis = 4

p = 5 {3, 4, 1, 5, 2}

gdiö = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 ysiö = 7

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 7) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_3 = \{2, 7\}$

kis = 0

p = 6 {3, 4, 1, 5, 2, 7}

gdiö = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve ysiö > 0, Adım 4'e geç

Öncesinde eleme kuralları uygulanır:

- En fazla yük kuralı UYGUN
- Jackson'ın baskınlık kuralı UYGUN
- İlk istasyon baskınlık kuralı UYGUN
- Etiketleme baskınlık kuralı UYGUN
- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

Adım 4 k = 4

kis = 10

Adım 5 ysiö = 6

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 6) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_4 = \{6\}$

kis = 5

p = 7 {3, 4, 1, 5, 2, 7, 6}

gdiö = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 ysiö = 8

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, i = 8 yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_3 = \{6, 8\}$

kis = 3

p = 8 {3, 4, 1, 5, 2, 7, 6, 8}

gdiö = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve ysiö > 0, Adım 4'e geç

Öncesinde eleme kuralları uygulanır:

- En fazla yük kuralı UYGUN

- Jackson'ın baskınlık kuralı UYGUN

- İlk istasyon baskınlık kuralı UYGUN

- Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN

- AS2 UYGUN

- AS3 UYGUN

- AS4 UYGUN

Adım 4 k = 5

kis = 10

Adım 5 ysiö = 9

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 9) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_5 = \{9\}$

kis = 0

p = 9 {3, 4, 1, 5, 2, 7, 6, 8, 9}

gdiö = 0

$p \neq n$ olduğundan Adım 3'e geç

Adım 3 Böyle bir iş öğesi yok ve ysiö > 0, Adım 4'e geç

Öncesinde eleme kuralları uygulanır:

- En fazla yük kuralı UYGUN

- Jackson'ın baskınlık kuralı UYGUN

- İlk istasyon baskınlık kuralı UYGUN

- Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN

- AS2 UYGUN

- AS3 UYGUN

- AS4 UYGUN

Adım 4 k = 6

kis = 10

Adım 5 ysiö = 10

Eleme kuralları uygulanır: - Etiketleme baskınlık kuralı UYGUN

- AS1 UYGUN
- AS2 UYGUN
- AS3 UYGUN
- AS4 UYGUN

AS4 = 6 istasyon olarak bulunduğundan, (i = 10) yeni seçilmiş iş öğesi olarak alınabilir, Adım 2'ye geç

Adım 2 $S_6 = \{10\}$

kis = 8

p = 10 {3, 4, 1, 5, 2, 7, 6, 8, 9, 10}

gdiö = 0

p = n olduğundan Adım 6'ya geç

Adım 6 Yeni çözüm 6 istasyonla dengelemeyi sağladığı için yeni varolan geçerli çözüm olur. Ayrıca (AS = 6) koşulunu sağladığından en iyi çözümdür.

FABLE ile bulunan en iyi çözüm:

1. İstasyon: 3 – 4

2. İstasyon: 1 – 5

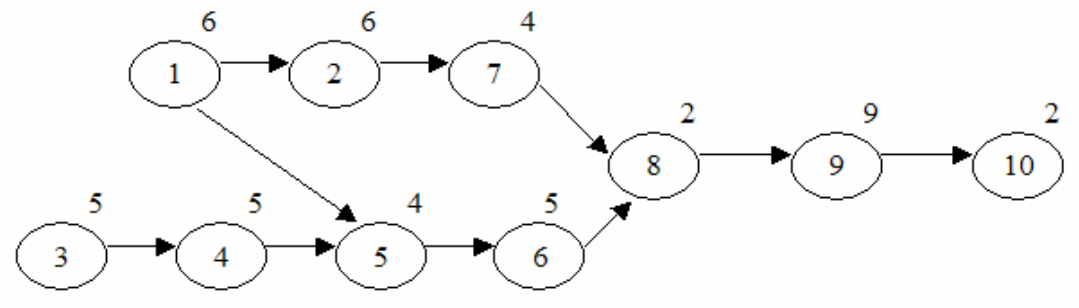
3. İstasyon: 2 – 7

4. İstasyon: 6 – 8

5. İstasyon: 9

6. İstasyon: 10

EK B: EUREKA Uygulaması



Şekil B.1: Tip-1 BMHDP Öncelik Diyagramı **(Scholl, 1999)**

Şekil B.1'teki öncelik diyagramı için C = 10'dur. Eureka'nın ilk defa sunulduğu Hoffmann'ın makalesinde istasyon atamaları için iş öğesi kümeleri "Hoffmann süreci" ile oluşturulmuştur. Bizim örneğimizde öncelik diyagramında yer alan iş öğeleri öncelik sırası gözetilerek numaralandırılmış olduğundan, istasyon atamaları için iş öğesi kümelerini bu numaraları takip ederek oluşturacağız.

TI(a): Montaj hattının a istasyonla dengelenmesi durumunda istasyonların toplam âtıl süresi.

$RI(S_1)$: Birinci istasyona olası iş öğesi kümelerinden birinin atanması durumunda, atama yapılmış tüm istasyonların toplam âtıl süresi.

C = 10

$AS1 = [48/10]^{+} = 5$ istasyon AS = 5

TI(5) = (5*10) – 48 = 2

AS = 5 için

$S_1 = \{3\}$	$RI(S_1) > 2$	olurlu değil
$S_1 = \{3, 4\}$	$RI(S_1) < 2$	olurlu
$S_2 = \{1\}$	$RI(S_2) > 2$	olurlu değil

$S_2 = \{1, 5\}$ $RI(S_2) < 2$ olurlu

$S_3 = \{2\}$ $RI(S_3) > 2$ olurlu değil

$S_3 = \{2, 7\}$ $RI(S_3) < 2$ olurlu

$S_4 = \{6\}$ $RI(S_4) > 2$ olurlu değil

$S_4 = \{6, 8\}$ $RI(S_4) > 2$ olurlu değil

AS = 5 için çözüm yok

AS değeri 1 arttırılarak elde edilen yeni AS değeri için çözüm aranır.

AS = 6

TI(6) = (6*10) – 48 = 12

AS = 6 için

$S_1 = \{1\}$ $RI(S_1) < 12$ olurlu

$S_2 = \{2\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{3\}$ $RI(S_3) > 12$ olurlu değil

$S_3 = \{3, 4\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{5\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{5, 6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_5) > 12$ olurlu değil

5. istasyon için seçenek iş öğesi kümesi kalmadığından sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Devam edilemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{3, 7\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{4\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{4, 5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

S_3 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_2 = \{2, 7\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{3\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{4\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{4, 5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{3, 4\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

S_3 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_2 = \{3\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{2\}$ $RI(S_3) > 12$ olurlu değil

$S_3 = \{2, 7\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{4\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{4, 5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{4\}$ $RI(S_3) > 12$ olurlu değil

$S_3 = \{4, 5\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{2\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{2, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

S_3 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_2 = \{3, 4\}$ $RI(S_{2)} < 12$ olurlu

$S_3 = \{2\}$ $RI(S_{3)} < 12$ olurlu

$S_4 = \{5\}$ $RI(S_{4)} > 12$ olurlu değil

$S_4 = \{5, 6\}$ $RI(S_{4)} < 12$ olurlu

$S_5 = \{7\}$ $RI(S_{5)} > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_{5)} > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{2, 5\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{2, 7\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{5\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{2\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{2, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6\}$ $RI(S_4) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_3 = \{5, 6\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{2\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_3) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_3) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{2, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

S_3 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

S_2 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_1 = \{3\}$ $RI(S_1) < 12$ olurlu

$S_2 = \{1\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{2\}$ $RI(S_3) > 12$ olurlu değil

$S_3 = \{4\}$ $RI(S_3) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_2 = \{4\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{1\}$ $RI(S_3) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_2 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_1 = \{3, 4\}$ $RI(S_1) < 12$ olurlu

$S_2 = \{1\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{2\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{5\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{5, 6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{2, 5\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{2, 7\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{5\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{5, 6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{5\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{2\}$ $RI(S_4) > 12$ olurlu değil

$S_4 = \{2, 7\}$ $RI(S_4) < 12$ olurlu55

$S_5 = \{6\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{6, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6\}$ $RI(S_4) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_3 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_2 = \{1, 5\}$ $RI(S_2) < 12$ olurlu

$S_3 = \{2\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{7\}$ $RI(S_5) > 12$ olurlu değil

$S_5 = \{7, 8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6, 7\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

S_4 ataması için olası iş öğesi kümesi kalmadı, bir düğüm geriye dönülür.

$S_3 = \{2, 7\}$ $RI(S_3) < 12$ olurlu

$S_4 = \{6\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{8\}$ $RI(S_5) > 12$ olurlu değil

Sürdürülemez, bir düğüm geriye dönülür.

$S_4 = \{6, 8\}$ $RI(S_4) < 12$ olurlu

$S_5 = \{9\}$ $RI(S_5) < 12$ olurlu

$S_6 = \{10\}$ $RI(S_6) = 12$ olurlu

Tüm iş öğeleri atanmıştır. Problem için optimal sonuç bulunmuştur: Montaj hattı 6 istasyon ile dengelenmiştir.

EUREKA ile bulunan en iyi çözüm:

1. İstasyon: 3 – 4

2. İstasyon: 1 – 5

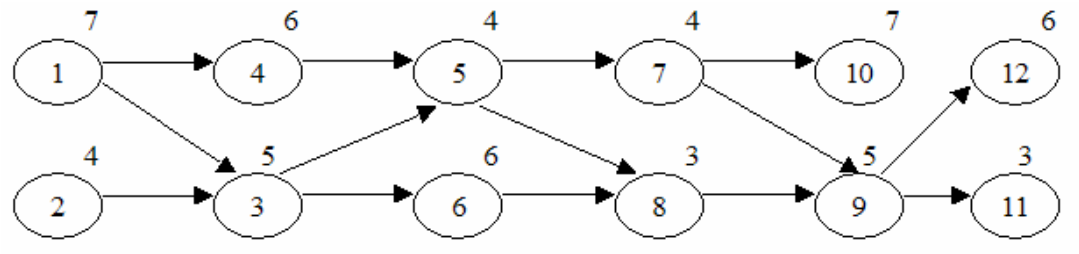
3. İstasyon: 2 – 7

4. İstasyon: 6 – 8

5. İstasyon: 9

6. İstasyon: 10

EK C: ULINO Uygulaması



Şekil C.1: Tip-1 BUHDP Öncelik Diyagramı **(Scholl, 1999)**

Şekil C.1'teki öncelik diyagramı için C = 10'dur.

AS1 = 6

AS2 = 6

AS3 = 6

AS6 = 6

AS6: $J\left(\frac{1}{2},1\right] = \{1, 4, 6, 10, 12\}$ ➔ $d_1 = 5$

A1 = {1} A2 = {10} A3 = {4} A4 = {6} A5 = {12}

$J\left(\frac{1}{3},\frac{1}{2}\right] = \{2, 3, 5, 7, 9\}$

A3 = {4, 2} A4 = {6, 5} A5 = {12, 7} Atanmayan iş öğesi sayısı: 2 ➔

$d_2 = [2/2]^+ = 1$

$J\left[0,\frac{1}{3}\right] = \{8, 11\}$ için $d_3 = Enb\left\{0, Enb\left\{d_3(q) \middle| q \in \left\{p_i \middle| i \in J\left[0,\frac{1}{3}\right]\right\}\right\}\right\}$

$d_3(q) = \sum_{i \in J[q,1-q]} p_i - \left|J\left(\frac{1}{2},1-q\right]\right| - d_2$ ➔ $d_3(0,3) = 6 - 5 - 1 = 0$

$d_3 = Enb\ \{0, Enb\ \{d_3(0,3), d_3(0,3)\}\} = 0$

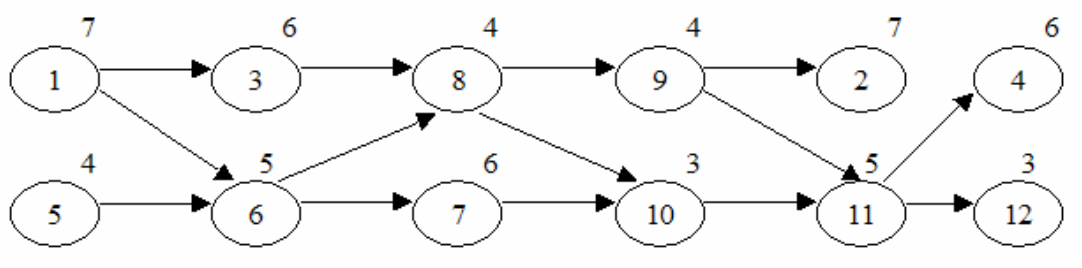
$AS6 = [d_1 + d_2 + d_3]^+ = [5 + 1 + 0]^+ = 6$

ÜS = 12 alındı.

ÜS > AS olduğundan işlem sürdürülür.

İş öğelerinin yeniden numaralandırılması: En küçük numaranın başlangıç veya bitiş iş öğesine verilmesi ile başlanır. Büyük işlem süresine sahip olan iş öğesi daha küçük numarayı alır. Eşit işlem süreli iş öğeleri arasında öncelik a) doğrudan ardıl, öncül sayısı yüksek olana, b) küçük orijinal iş öğesi numarası olana verilir.

Yeniden numaralandırılmış öncelik diyagramı Şekil C.2'de gösterilmiştir.



Şekil C.2: Yeniden Numaralandırılmış Öncelik Diyagramı

RI_0 (Kalan âtıl süre) = TI (6) (6 istasyon için toplam âtıl süre) = 0

Jackson baskınlık kuralına göre iş öğesi çiftleri:

İleri doğru baskınlık: (1, 5), (3, 7), (9, 10), (2, 12), (4, 12)

Geriye doğru baskınlık: (4, 12), (2, 7), (7, 3), (3, 5), (1, 5)

Dallandırma (k = 0):

S_{11}: birinci istasyon için birinci olasılık kümesi

I (S_{11}): S_{11} ataması sonrasında istasyonun âtıl süresi

S_{11} = {1, 12} ➔ $I(S_{11}) \leq RI_0$ (1. grup)

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{11} elenmez ➔ k+1 Orijinal numara ile S_{11} = {1, 11}

Dallandırma (1)

Önceki düzeyden AS = 6

AS1 = 5 + 1 = 6

AS2 = 5 + 1 = 6

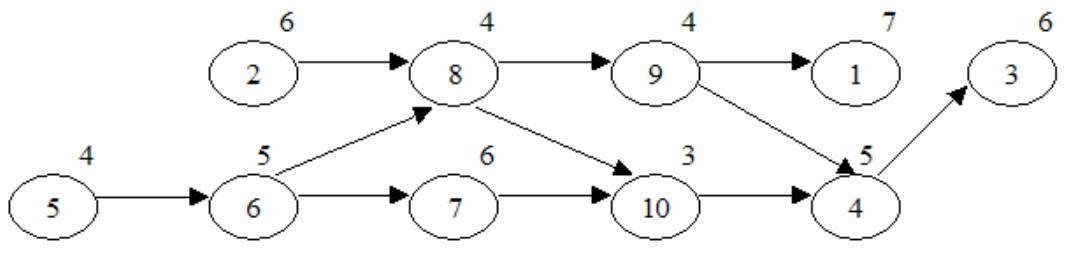
AS3 = 5 + 1 = 6

AS6 = 5 + 1 = 6

AS = 6

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.3'te gösterilmiştir.



Şekil C.3: İndirgenmiş Problem Öncelik Diyagramı 1

$RI_1 = TI(6) = 0$

$S_{21} = \{2, 5\}$ ➔ $I(S_{21}) \leq RI_1$ (1. grup)

Mantıksal testler uygulanır:	- iş öğesi süresi arttırma kuralı	UYGUN
	- ağaç baskınlık kuralı	UYGUN
	- Jackson baskınlık kuralı	UYGUN

S_{21} elenmez ➔ k+1 Orijinal numara ile $S_{21} = \{2, 4\}$

Dallandırma (2)

Önceki düzeyden AS = 6

AS1 = 4 + 2 = 6

AS2 = 4 + 2 = 6

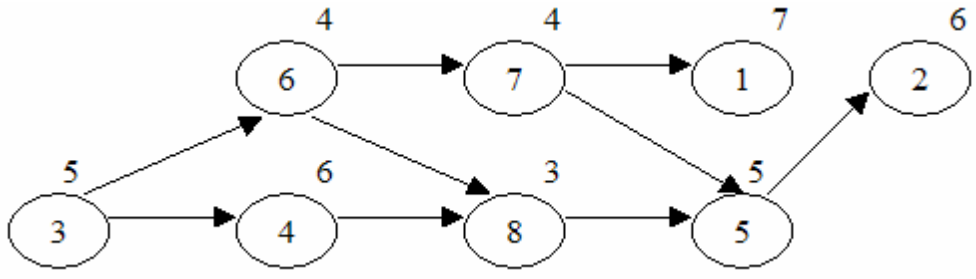
AS3 = 4 + 2 = 6

AS6 = 4 + 2 = 6

AS = 6

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.4'te gösterilmiştir.



Şekil C.4: İndirgenmiş Problem Öncelik Diyagramı 2

$RI_2 = TI(6) = 0$

$S_{31} = \{1\}$ ➔ $I(S_{31}) > RI_2$ (2. grup)

$I(S_3) \leq RI_2$'yi sağlayan bir atama olasılığı olmadığından ➔ AS = AS + 1 = 7

AS < ÜS

$RI_2 = TI(7) = 10$

$S_{31} = \{1\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{31} elenmez ➔ k+1 Orijinal numara ile $S_{31} = \{10\}$

Dallandırma (3)

Önceki düzeyden AS = 7

AS1 = 3,3 + 3 = 7

AS2 = 3 + 3 = 6

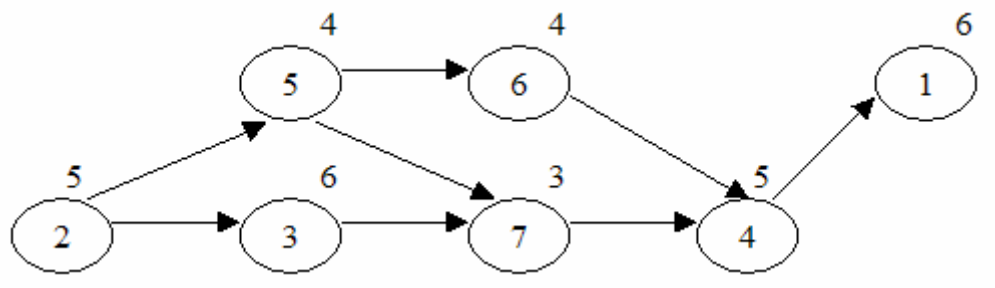
AS3 = 3 + 3 = 6

AS6 = 4 + 3 = 7

AS = 7

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.5'de gösterilmiştir.



Şekil C.5: İndirgenmiş Problem Öncelik Diyagramı 3

$RI_3 = TI(7) - (I(S_1) + I(S_2) + I(S_3)) = 10 - 3 = 7$

$S_{41} = \{1\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{41} elenmez ➔ k+1 Orijinal numara ile $S_{41} = \{12\}$

Dallandırma (4)

Önceki düzeyden AS = 7

AS1 = 2,4 + 4 = 7

AS2 = 2 + 4 = 6

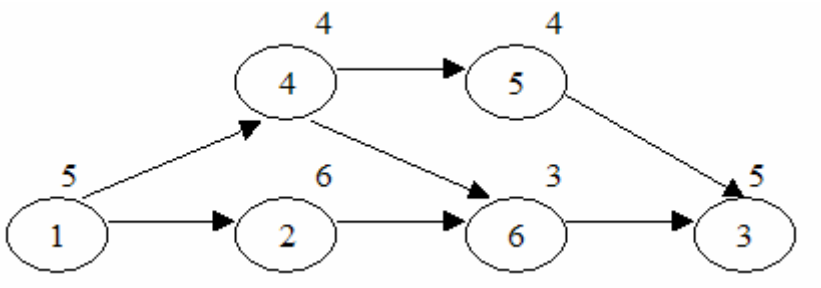
AS3 = 3 + 4 = 7

AS6 = 3 + 4 = 7

AS = 7

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.6'da gösterilmiştir.



Şekil C.6: İndirgenmiş Problem Öncelik Diyagramı 4

$RI_4 = TI\ (7) - (I(S_1) + I(S_2) + I(S_3) + I(S_4)) = 10 - 7 = 3$

$S_{51} = \{1, 3\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{51} elenmez ➔ k+1 Orijinal numara ile $S_{51} = \{3, 9\}$

Dallandırma (5)

Önceki düzeyden AS = 7

AS1 = 1,7 + 5 = 7

AS2 = 1 + 5 = 6

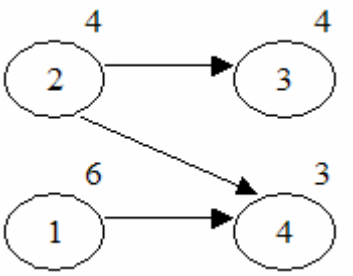
AS3 = 1,5 + 5 = 7

AS6 = 2 + 5 = 7

AS = 7

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.7'de gösterilmiştir.



Şekil C.7: İndirgenmiş Problem Öncelik Diyagramı 5

$RI_5 = TI\ (7) - (I(S_1) + I(S_2) + I(S_3) + I(S_4) + I(S_5)) = 10 - 7 = 3$

$S_{61} = \{1, 2\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{61} elenmez ➔ k+1 Orijinal numara ile $S_{61} = \{6, 5\}$

Dallandırma (6)

AS = 7

AS < ÜS

Yeniden numaralandırma sonucu kalan iki iş öğesi 1 ve 2 numaralarını alır.

$S_{71} = \{1, 2\}$ Orijinal numara ile $S_{71} = \{7, 8\}$

Tüm iş öğeleri atanmıştır. Artık ÜS = 7

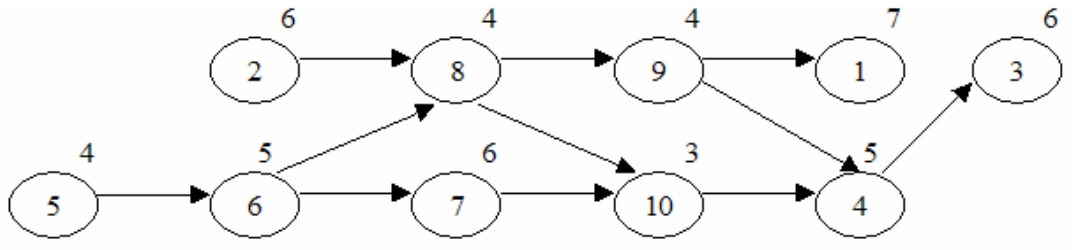
Varolan geçerli çözüm (orijinal numaralarla) {1, 11; 2, 4; 10; 12; 3, 9; 5, 6; 7, 8}

Daha iyi bir çözüm aramak için AS = 6 olan son istasyona dönülür;

$S_1 = \{1, 11\}$ ➔ S_2 için AS = 6

$S_2 = \{12, 2\}$ ➔ S_3 için AS = 7 olduğundan S_2 için AS = 6 olan başka bir atama olasılığı olup olmadığı kontrol edilir.

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.8'de gösterilmiştir.



Şekil C.8: İndirgenmiş Problem Öncelik Diyagramı 6

$S_{22} = \{3, 5\}$ $I(S_{22}) \leq RI_1$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{22} elenmez ➔ k+1 Orijinal numara ile $S_{22} = \{2, 12\}$

Dallandırma (2)

Önceki düzeyden AS = 6

AS1 = 4 + 2 = 6

AS2 = 4 + 2 = 6

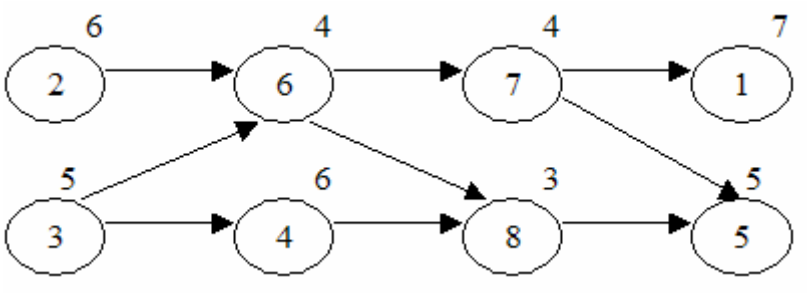
AS3 = 4 + 2 = 6

AS6 = 4 + 2 = 6

AS = 6

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.9'da gösterilmiştir.



Şekil C.9: İndirgenmiş Problem Öncelik Diyagramı 7

$RI_2 = TI(6) = 0$

$S_{32} = \{3, 5\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{32} elenmez ➔ k+1 Orijinal numara ile $S_{32} = \{3, 9\}$

Dallandırma (3)

Önceki düzeyden AS = 6

AS1 = 3 + 3 = 6

AS2 = 3 + 3 = 6

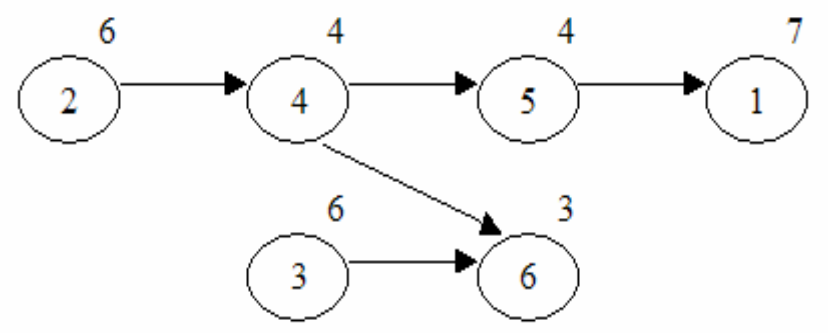
AS3 = 3 + 3 = 6

AS6 = 3 + 3 = 6

AS = 6

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.10'da gösterilmiştir.



Şekil C.10: İndirgenmiş Problem Öncelik Diyagramı 8

$RI_3 = TI(6) = 0$

$S_{42} = \{1, 6\}$

Mantıksal testler uygulanır:	- iş öğesi süresi arttırma kuralı	UYGUN
	- ağaç baskınlık kuralı	UYGUN
	- Jackson baskınlık kuralı	UYGUN
S_{42} elenmez ➔ k+1	Orijinal numara ile $S_{42} = \{10, 8\}$	

Dallandırma (4)

Önceki düzeyden AS = 6

AS1 = 2 + 4 = 6

AS2 = 2 + 4 = 6

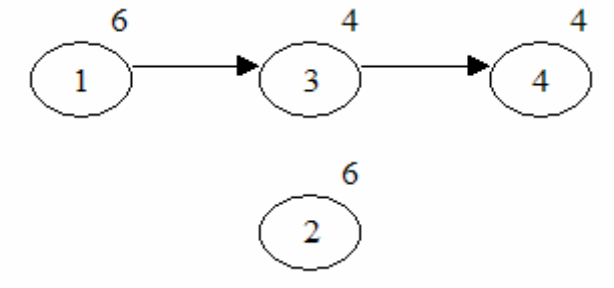
AS3 = 2 + 4 = 6

AS6 = 2 + 4 = 6

AS = 6

AS < ÜS

Yeniden numaralandırma ile elde edilen öncelik diyagramı Şekil C.11'de gösterilmiştir.



Şekil C.11: İndirgenmiş Problem Öncelik Diyagramı 9

$RI_4 = TI\ (6) = 0$

$S_{52} = \{1, 3\}$

Mantıksal testler uygulanır: - iş öğesi süresi arttırma kuralı UYGUN

- ağaç baskınlık kuralı UYGUN

- Jackson baskınlık kuralı UYGUN

S_{52} elenmez ➔ k+1 Orijinal numara ile $S_{52} = \{4, 5\}$

Dallandırma (5)

AS = 6

AS < ÜS

Yeniden numaralandırma sonucunda kalan iş öğeleri 1 ve 2 numaralarını alır.

$S_{62} = \{1, 2\}$ Orijinal numara ile $S_{62} = \{6, 7\}$

Tüm iş öğeleri atanmıştır. AS = 6 ile en iyi çözüm bulunmuştur.

En iyi çözüm (orijinal numaralarla) {1, 11; 12, 2; 3, 9; 10, 8; 4, 5; 6, 7}

ULINO ile bulunan en iyi çözüm:

1. İstasyon: 1 – 11

2. İstasyon: 12 – 2

3. İstasyon: 3 – 9

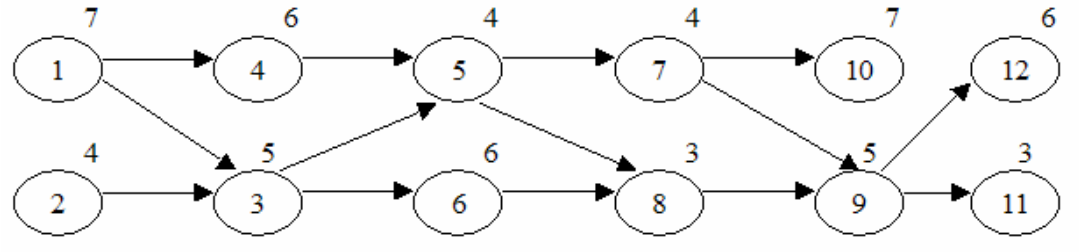
4. İstasyon: 10 – 8

5. İstasyon: 4 - 5

6. İstasyon: 6 – 7

EK D: U-OPT1 Uygulaması

Üst-sınır (ÜS) olarak problemin basit montaj hattı dengeleme problemi çözümünü kullanalım. Bu amaçla problem önce EUREKA ile çözülecek. Problemin öncelik diyagramı Şekil D.1'de gösterilmiştir.



Şekil D.1: Tip-1 BUHDP'nin Öncelik Diyagramı **(Scholl, 1999)**

C = 10'dur.

AS1 = 60 / 10 = 6 istasyon TI(6) = 0

$S_1 = \{1\}$ $I(S_1) = 3 > 0$

$S_1 = \{2\}$ $I(S_1) = 6 > 0$

Süremez, AS = 7 ile denenir.

TI(7) = 10

$S_1 = \{1\}$ $I(S_1) = 3 < 10$

$S_2 = \{2, 4\}$ $I(S_2) = 3 < 10$

$S_3 = \{3, 5\}$ $I(S_3) = 4 < 10$

$S_4 = \{6, 7\}$ $I(S_4) = 4 < 10$

$S_5 = \{8, 10\}$ $I(S_5) = 4 < 10$

$S_6 = \{9, 11\}$ $I(S_6) = 6 < 10$

$S_7 = \{12\}$ $I(S_7) = 10 = 10$

Problem Tip-1 BMHDP olarak düşünüldüğünde EUREKA ile istasyon sayısı 7 bulunur, bu da U – OPT1 için, aynı problemin BMHDP çözümü olduğundan, ÜS olarak kullanılır.

U – OPT1 Uygulaması

GÜS (Global üst-sınır) = 7 istasyon (BMHDP sonucu)

- Düzey 0

AS1 = 60/10 = 6

AS2= 5 + 1 = 6

AS3 = 2 + 4 = 6

AS5 = 6

GAS (Global alt-sınır) = 6 istasyon

AS5'in hesaplanması:

8 – 11 – 2 – 5 – 7 – 3 – 9 – 4 – 6 – 12 – 1 – 10

Adım1: $t_8 + t_{10} = 3 + 7 = 10 = C$

Adım4: S1 = {8, 10}

Adım2: $t_{11} + t_1 = 3 + 7 = 10 = C$

Adım4: S2 = {11, 1}

Adım2: $t_2 + t_{12} = 4 + 6 = 10 = C$

Adım4: S3 = {2, 12}

...

S4 = {5, 6}, S5 = {7, 4}, S6 = {3, 9}

buradan AS5 = 6 olarak bulunur.

- Düzey 1

S_{11} = {1} AS1 = 6

S_{12} = {1, 11} AS1 = 5 AS2 = 5 AS3 = 5 AS5 = 5 AS = 5

S_{13} = {2} AS1 = 6

S_{14} = {2, 11} AS1 = 6

S_{15} = {2, 12} AS1 = 5 AS2 = 5 AS3 = 5 AS5 = 5 AS = 5

S_{16} = {10} AS1 = 6

S_{17} = {10, 11} AS1 = 5 AS2 = 5 AS3 = 5 AS5 = 5 AS = 5

S_{18} = {11} AS1 = 6

S_{19}= {11, 12} AS1 = 6

En küçük AS'ye sahip ilk yapılan küme seçilir: S_{12}

BK6 sözkonusu değil.

En iyi düğüm ➔ geçerli düğüm S_{12} = {1, 11} olur.

S_{12} için yerel alt-sınır (YAS) = 5 + 1 = 6

YAS ≥ GÜS değil

- Düzey 2

S_{21} = {2} AS1 = 5

S_{22} = {2, 3} AS1 = 5

S_{23} = {2, 4} AS1 = 4 AS2= 4 AS3 = 4 AS5 = 4 AS = 4

S_{24} = {2, 12} AS1 = 4 AS2= 4 AS3 = 4 AS5 = 4 AS = 4

S_{25} = {4} AS1 = 5

S_{26} = {10} AS1 = 5

S_{27} = {12} AS1 = 5

BK6 sözkonusu değil.

Geçerli düğüm S_{23} = {2, 4} olur

YAS = 4 + 2 = 6

YAS ≥ GÜS değil

- Düzey 3

BK6 sözkonusu değil.

S_{31} = {3} AS1 = 4

S_{32} = {3, 5} AS1 = 4

S_{33} = {10} AS1 = 4

S_{34} = {12} AS1 = 4

Biri en iyi olarak seçilse bile hepsi için en iyi olasılık ile YAS = 4 + 3 = 7 olur.

Bu durumda YAS ≥ GÜS olur.

Düzey = Düzey – 1 = 2

- Düzey 2

BK6 sözkonusu değil.

Bu düzeyde tek olasılık kaldı: S_{24} = {2, 12}

Geçerli düğüm S_{24} = {2, 12}

YAS = 4 + 2 = 6

YAS ≥ GÜS değil

- Düzey 3

BK6 sözkonusu değil.

S_{31} = {3} AS1 = 4

S_{32} = {3, 9} AS1 = 3 AS2 = 3 AS3 = 3 AS5 = 3 AS = 3

S_{33} = {4} AS1 = 4

S_{34} = {9} AS1 = 4

S_{35} = {9, 8} AS1 = 4

S_{36} = {10} AS1 = 4

Geçerli düğüm S_{32} = {3, 9}

YAS = 3 + 3 = 6

YAS ≥ GÜS değil

- Düzey 4

BK6 sözkonusu değil.

S_{41} = {4} AS1 = 3

S_{42} = {4, 5} AS1 = 2 AS2= 2 AS3 = 2 AS5 = 2 AS = 2

S_{43} = {6} AS1 = 3

S_{44} = {8, 6} AS1 = 3

S_{45} = {4, 8} AS1 = 3

S_{46} = {8} AS1 = 3

S_{47} = {8, 10} AS1 = 2 AS2= 2 AS3 = 2 AS5 = 2 AS = 2

Geçerli düğüm S_{42} = {4, 5} olur

YAS = 4 + 2 = 6

YAS ≥ GÜS değil

- Düzey 5

BK6 sözkonusu değil.

S_{51} = {6} AS1 = 2

S_{52} = {6, 7} AS1 = 1 AS2= 1 AS3 = 1 AS5 = 1 AS = 1

S_{53} = {6, 8} AS1 = 2

S_{54} = {7} AS1 = 2

S_{55} = {7, 8} AS1 = 2

S_{56} = {8} AS1 = 2

S_{57} = {8, 10} AS1 = 1 AS2= 1 AS3 = 1 AS5 = 1 AS = 1

S_{58} = {10} AS1 = 2

Geçerli düğüm S_{52} = {6, 7} olur

YAS = 5 + 1 = 6

YAS ≥ GÜS değil

- Düzey 6

BK6 sözkonusu değil.

S_{61} = {8} AS1 = 1

S_{62} = {8, 10} AS1 = 0

S_{63} = {10} AS1 = 1

Geçerli düğüm S_{62} = {8, 10}

Tüm iş öğeleri atandı. Varolan istasyon sayısı 6 ➔ yeni yerel üst-sınır (YÜS) = 6

YÜS = GAS = 6 ➔ En iyi çözüm bulundu.

U-OPT1 ile bulunan en iyi çözüm:

1. İstasyon: 1 – 11

2. İstasyon: 2 – 12

3. İstasyon: 3 – 9

4. İstasyon: 4 - 5

5. İstasyon: 6 - 7

6. İstasyon: 8 – 10

ÖZGEÇMİŞ

Ayşe Elvan Bayraktaroğlu 20 Nisan 1980 İstanbul doğumludur. Lise eğitimini aldığı Sankt Georg Avusturya Lisesi'nden 1999 yılında mezun olmuştur. 2003 yılında İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi Elektrik Mühendisliği Bölümü'nü bitirmiştir. Aynı yıl İstanbul Teknik Üniversitesi İşletme Fakültesi Endüstri Mühendisliği Bölümü'nde yüksek lisans eğitimine başlamıştır. Aralık 2005'ten beri Endüstri Mühendisliği Yöneylem Araştırması Anabilim Dalı'nda araştırma görevlisi olarak görev yapmaktadır.